

134580

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(YÜKSEK LİSANS TEZİ)

**YAPAY SİNİR AĞLARI TEMELLİ
MODEL ÖNGÖRÜLÜ KONTROL**

Savaş ŞAHİN

Elektrik – Elektronik Mühendisliği Anabilim Dalı

Bilim Dalı Kodu : 609.01.00.

Sunuş Tarihi : 13.08.2003

Tez Danışmanı : Yrd. Doç. Dr. Aydoğan SAVRAN

134580

Bornova-İZMİR

**T.C. YÜKSEKÖĞRETİM KURULU
BÖKÜMANTASYON MERKEZİ**

Savaş ŞAHİN tarafından YÜKSEK LİSANS TEZİ olarak sunulan “Yapay Sinir Ağları Temelli Model Öngörülü Kontrol” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 13 Ağustos 2003 tarihinde yapılan tez savunma sınavında aday oybirliği ile başarılı bulunmuştur.

13 Ağustos 2003

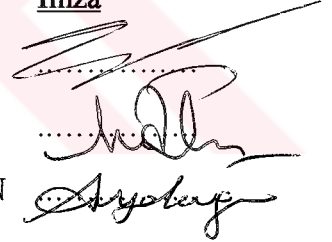
Jüri Üyeleri :

Jüri Başkanı : Prof.Dr. Erol UYAR

Raportör Üye : Yrd.Doç.Dr. Musa ALCI

Üye : Yrd.Doç.Dr. Aydoğan SAVRAN

İmza



ÖZET
YAPAY SINIR AĞLARI TEMELLİ
MODEL ÖNGÖRÜLÜ KONTROL

ŞAHİN, Savaş

Yüksek Lisans Tezi, Elektrik – Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Aydoğan SAVRAN

13 Ağustos 2003, 103 sayfa

Bu tez çalışmasında, bir yapay sinir ağları (YSA) temelli doğrusal olmayan öngörülü kontrol yapısı geliştirilmiştir.

Bu yapıda, dinamik sistemlerin giriş – çıkış ilişkisi YSA tanılama modeli kullanılarak, gösterilmiştir.

YSA ağırlıkları güncellenmesi en dik iniş yöntemi ile gerçekleştirilmiştir. Daha sonra, YSA tanılama modeli temel alınarak doğrusal olmayan model öngörülü kontrol yapısı oluşturulmuştur.

Kontrol sinyali, YSA modeli ile öngörülen sistem çıkışı ve referans arasındaki karesel hatayı içeren bir ölçütü minimize edecek şekilde kestirilir.

Sonuç olarak, YSA temelli model öngörülü kontrol yapısının, doğrusal olmayan karmaşık sistemlerin kontrolü uygulamalarında etkin bir alternatif olduğu gösterilmiştir.

Anahtar sözcükler: Optimal kontrol, yapay sinir ağları (YSA), model öngörülü kontrol (MÖK), doğrusal olmayan dinamik sistemler.

ABSTRACT
NEURAL NETWORK BASED
MODEL PREDICTIVE CONTROL

ŞAHİN, Savaş

MSc in Electrical – Electronics Eng.

Supervisor: Assist of Prof. Aydoğan SAVRAN

13 August 2003, 103 pages

In this thesis, it is developed a neural network based nonlinear model predictive control scheme.

Neural network (NN) identification models are used to represent the input-output relation of dynamical systems.

The steepest descent method is used to update NN parameters. Then, a nonlinear model predictive controller scheme is designed on the basis of a neural network plant model.

Based on the neural network model, the control is estimated by minimizing a projected cost function that penalizes future tracking errors.

Consequently, NN based model predictive control scheme is an effective alternative for controlling the nonlinear complex systems.

Keywords: Optimal control, neural network (NN), model predictive control (MPC), nonlinear dynamical system.

TEŐEKKÜR

Bütün tez alıřmalarım boyunca bilgi ve deneyimleriyle beni ynlendiren ve yreklendiren tez danıřmanım Yrd. Do. Dr. Aydoėan SAVRAN bařta olmak zere, Prof. Dr. Erol UYAR, Yrd. Do. Dr. Musa ALCI ve yksek lisans eėitimim boyunca katkılarını esirgemeyen E.. Elektrik – Elektronik Mh. blm ėretim yeleri ve personeline, maddi ve manevi destekleriyle yanımda olan deėerli dostlarıma, E.. Ege Meslek Yksekokulu ėretim elemanları ve personeline teőekkr bir bor bilirim.

Ayrıca benim bu gnlere ulařmamda en fazla emeėi geen ve destekleyen, annem Yařar ŐAHİN, babam nal ŐAHİN ve kardeřim Burcu ŐAHİN'e sonsuz teőekkr ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET.....	V
ABSTRACT.....	VII
TEŞEKKÜR.....	IX
ŞEKİLLER DİZİNİ.....	XIV
ÇİZELGELER DİZİNİ.....	XVII
1. GİRİŞ.....	1
2. OPTİMAL KONTROL TEKNİKLERİ.....	8
2.1 Giriş.....	8
2.2 Modern Kontrolün Felsefesi.....	8
2.3 Doğrusal Sistemler.....	9
2.4 Doğrusal Olmayan Sistemler.....	10
2.4.1 İki Nokta Sınır Değer Problemi.....	15
2.4.2 Optimal Kontrol Uygulamaları.....	15
3. MODEL ÖNGÖRÜLÜ KONTROL (MÖK) TEKNİĞİ... 27	27
3.1 Giriş.....	27
3.2 Yapılan Çalışmalar ve Gelişimi.....	30
3.3 Temel Model Öngörülü Kontrol Kavramı	32

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
3.4 Tek Girişli – Tek Çıkışlı (TGTÇ) MÖK Algoritması....	34
3.4.1. Model Algoritmik Kontrol (MAK).....	35
3.4.2 Dinamik Matris Modeli (DMM)	38
3.4.3 Dinamik Matris Algoritması (DMA).....	39
3.4.4 Genelleştirilmiş Öngörülü Kontrol (GÖK) Metodu... ..	45
3.5 Öngörülü Kontrol Kanunu ve PÖ Tanımı.....	47
3.5.1 GÖK’de Kontrol Sinyali Tanımlanması.....	50
3.5.2 Kontrol Ufkunun Etkisi.....	52
3.5.3 Çıkış ve Kontrol Ufuklarının Belirlenmesi.....	53
3.5.4 GÖK’ü Oluşturan Yapı Taşları.....	54
3.5.5 GÖK Algoritması.....	55
3.6 TGTÇ Model Öngörülü Kontrol Uygulaması.....	56
3.7 Çok Girişli – Çok Çıkışlı (ÇGÇÇ) Sistemler için MÖK algoritması.....	63
3.7.1 ÇGÇÇ Model Öngörülü Kontrol Uygulaması.....	66
4. YAPAY SİNİR AĞLARI.....	70
4.1 Giriş.....	70
4.2 YSA Temel Özellikleri.....	70
4.2.1 Yapay Sinir Hücre Yapısı.....	71
4.2.2 Yapay Sinir Ağ Yapıları.....	72
4.3 Yapay Sinir Ağında Öğrenme.....	73
4.4 Hata Küçültülmesi ve En Dik İniş Algoritması.	74

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
4.5 En Dik İniş Yöntemi (EDİY) Tanımı.....	75
5. YSA TEMELLİ MODEL ÖNGÖRÜLÜ KONTROL....	77
5.1 Giriş.....	77
5.2 Doğrusal Olmayan Tanılama Modelleri.....	78
5.3 YSA ile Sistem Tanılama.....	78
5.3.1 Seri – Paralel Tanılama Yapısı.....	79
5.3.2 Paralel Tanımlama Yapısı.....	80
5.4 YSA Temelli Model Öngörülü Kontrol Yapısı.....	81
5.5 YSA ile Süreç Modelinin Eğitimi.....	82
5.6 YSA Temelli MÖK Benzetim Çalışmaları.....	85
5.6.1 Doğrusal Süreç	86
5.6.2 Doğrusal Olmayan Süreç	88
5.6.3 Doğrusal Olmayan CSTR Uygulaması.....	91
6.SONUÇ.....	98
KAYNAKLAR DİZİNİ.....	100
ÖZGEÇMİŞ.....	103

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 Optimal kontrolör Hamiltonian algoritması	14
2.2 Örnek 2.1 için DC motor durum denklemleri.....	19
2.3 Örnek 2.1 için yardımcı durum denklemleri.....	20
2.4 Örnek 2.1 için $u(t)$ ve $PÖ$	20
2.5 Örnek 2.2 için (CSTR) durum değişkenleri.....	25
2.6 Örnek 2.2 için (CSTR) yardımcı durum değişkenleri	25
2.7 Örnek 2.2 için (CSTR) kontrol sinyali “u”	26
2.8 Örnek 2.2 için (CSTR) performans ölçütü.....	26
3.1 Model öngörülü kontrol için giriş-çıkış değerleri	29
3.2 Tek-Amaçlı Model-Öngörülü Kontrol algoritması	33
3.3 TAMÖK algoritması blok diyagramı	39
3.4 Referans, kontrol ve çıkış yörüngeleri.....	47
3.5 Genelleştirilmiş öngörülü kontrol algoritması	55
3.6 Doğrusal sistem örnek 3.1 için blok diyagram	56
3.7 Örnek 3.1 için $N_u=1$, $k_e=-0.5$	57
3.8 Örnek 3.1 için $N_u=1$, (a) $k_e=0$, (b) $k_e=0.5$	58
3.9 Örnek 3.1 için $N_u=2$, $k_e = -0.5$	59
3.10 Örnek 3.1 için $N_u=2$, (a) $k_e=0$, (b) $k_e=0.5$	60
3.11 Örnek 3.1 için $N_u=3$, (a) $k_e=-0.5$, (b) $k_e=0$	61
3.12 Örnek 3.1 için $N_u = 3$, $k_e = 0.5$	62
3.13 Çok Girişli – Çok Çıkışlı sistem modeli.....	64

ŞEKİLLER DİZİNİ (devam)

<u>Şekil</u>	<u>Sayfa</u>
3.14 Çok Girişli – Çok Çıkışlı süreç blok diyagramı.....	67
3.15 Örnek 3.2 için çıkış y_1 ve kontrol sinyali u_1	68
3.16 Örnek 3.2 için çıkış y_2 ve kontrol sinyali u_2	68
4.1 Yapay sinir ağının fiziksel bağlantısı	71
4.2 Yapay sinir hücresi yapısı.....	72
4.3 Hatanın küçültülmesi ve en dik iniş yöntemi	74
4.4 Performans yüzeyi ve gradyan.....	75
5.1 Seri – paralel tanılama, (z^{-1} , geriye kaydırma operatörü).....	79
5.2 Paralel tanılama, (z^{-1} , geriye kaydırma operatörü).....	80
5.3 YSA temelli model öngörülü kontrol yapısı.....	82
5.4 Süreç modeli için YSA temelli tasarım	83
5.5 CSTR süreci için eğitim veri seti	84
5.6 CSTR süreci için test veri seti.....	84
5.7 YSA için en küçük kareler toplamı hatası	85
5.8 Örnek 5.1 için doğrusal süreç modeli.....	86
5.9 Örnek 5.1 için çıkış “y”, giriş “u” ve $N_u = 2$	87
5.10 Örnek 5.1 için çıkış “y”, giriş “u” ve $N_u = 6$	88
5.11 Örnek 5.2 için “y” ve “u” ($N_2 = 10, N_u = 1$).....	89
5.12 Örnek 5.2 için “y” ve “u” ($N_2 = 10, N_u = 2$).....	89
5.13 Örnek 5.2 için “y” ve “u” ($N_2 = 10, N_u = 3$).....	90

ŞEKİLLER DİZİNİ (devam)

<u>Sekil</u>	<u>Sayfa</u>
5.14 Örnek 5.2 için “y” ve “u” ($N_2 = 5, N_u = 2$).....	90
5.15 Örnek 5.2 için “y” ve “u” ($N_2 = 20, N_u = 2$).....	91
5.16 CSTR için YSA model ve süreç sonuçları “ x_1 ve x_2 ”	93
5.17 CSTR için durum değişkenleri “ x_1 ve x_2 ” ve “u” ($N_2=3$).....	94
5.18 CSTR için durum değişkenleri “ x_1 ve x_2 ” ve “u” ($N_2=6$).....	95
5.19 CSTR için durum değişkenleri “ x_1 ve x_2 ” ve “u” ($N_2=8$).....	96
5.20 CSTR, PÖ’de $(25-u)^2$ yok iken “ x_1 ve x_2 ” ve “u” ($N_2=6$).....	97

ÇİZELGELER DİZİNİ

Çizelge

Sayfa

2.1 Doğrusal olmayan optimal kontrolör tasarım tablosu.....13



1. GİRİŞ

Bu tez çalışmasında, YSA (Yapay Sinir Ağları) temelli öngörülü kontrol yapısı geliştirilerek, doğrusal olmayan süreçlerin kontrolünde kullanılması, YSA'nın doğrusal olmayan eşleme, paralellik, öğrenme ve uyarlanma gibi özelliklerinden faydalanılarak, mevcut öngörülü kontrol yapılarının performanslarının iyileştirmesi amaçlanmaktadır.

Öngörülü kontrol yapıları, çok değişkenli kontrol uygulamalarında bir endüstri standardı haline gelmiştir. Kimyasal süreçler, otomotiv, uçak-uzay ve kağıt endüstrisi yoğun olarak uygulandığı alanlardandır. Uygulamada kontrol edilen süreçler çoğunlukla doğrusal olmamasına rağmen kontrolör doğrusal dinamik modeller kullanılarak tasarlanmaktadır. Her ne kadar bu modeller süreç tek bir denetim noktasında işletildiği durumlarda kabul edilebilir performans gösterebilir de, farklı denetim noktalarında çalıştırılmaları durumunda kötü kontrol performansına sebep olmaktadır. Bu noktadan başlangıç alarak, bu tez çalışmasında geliştirilen YSA temelli yaklaşım ile doğrusallaştırma gibi sınırlamaları ortadan kaldıracak, uyarlanma ve öğrenme özelliklerine sahip öngörülü kontrol yapısı oluşturulmuştur. Sonuçta geliştirilen algoritma ile daha az adımda çözüme ulaşılabilecektir.

Genel anlamda kontrol, sistemlerin fiziksel davranışlarının etkileriyle ilgilenen bir bilim dalıdır. Kontrol, günlük hayatın her sahasında karşımıza çıkar. Mekanik, elektrik, elektro-mekaniksel v.b. bir çok araçlar insan denetimi olmadan otomatik olarak çalışmaktadırlar. Bu süreç kontrolü otomatik kontrol sayesinde olmaktadır. Kontrol sistemleri genellikle geri-beslemeli kontrolörler kullanır. Geri-beslemeli kontrol ve hata devamlı olarak gözlenir. İnsan da ise kendi kontrol mekanizmasıyla otomatik olarak kontrol etmektedir ki bu da kontrol mekanizmalarının hem insan hem de fiziksel dış dünyada ne kadar önemli olduğunu gösterir.

Kontrol ve otomatik kontrol ifadeleri birbirini tamamlayan ifadelerdir. Kontrolün tanımı yapılırsa; incelenmiş davranışların istenilen değerde tutulması veya arzulanan değişimleri yapabilmesi için yapılan eylemlerin hepsidir. Otomatik kontrol ise, kontrol işleminin, kurulan bir fiziksel bir düzenek tarafından insanın direkt müdahalesi olmaksızın gerçekleştirilmesidir.

Son yıllarda, otomatik kontrol teorisinde bir çok gelişmeler olmuştur. Kontrol teorisinin önemli yeri olan geri beslemeli (GB) kontrol teorisinin gelişimi incelendiğinde, bazı temel gelişimleri gözlemek mümkündür. Günümüzde GB kontrol, bir mühendislik disiplini olarak yerini almıştır. Geri beslemeli kontrol sistemlerinin gelişimini etkileyen tarihteki önemli gelişmeler şöyle özetlenebilir (Lewis, 1992).

1. M.Ö. 300 ile M.S. 1200 yılları arasında Arapların ve Yunanlıların zamanı doğru olarak tespit ederek takip edebilmeye yönelik çalışmaları.
2. 18. yüzyılda Avrupa'daki endüstriyel devrim olsa da, kökü 1600'lere dayanır.
3. 1910 ile 1945 yılları arasında, toplu haberleşmenin başlaması ve Birinci ve İkinci Dünya Savaşları.
4. Uzay ve bilgisayar çağının 1957'de başlaması.

Endüstriyel devrim ve dünya savaşları arasındaki zamanlarda oldukça önemli gelişmeler olmuştur. 1868'de, J.C. Maxel bir geri besleme kontrol sisteminin ilk matematiksel analizini yaptı. Savaş yıllarındaki gelişmeler, matematiksel analizlerin uygulanmasını sağladı. 1868'den 1900'lü yılların başları arasındaki zaman, otomatik kontrolün ilkel zamanı olarak adlandırılır. 1960'a kadar olan kısma klasik zaman ve 1960'dan günümüze kadar ki zamana ise modern zaman ismi verilir.

Endüstri devrimine kadar, GB kontrol sistemlerinin tasarımı büyük bir mühendislik sezgisiyle beraber deneme yanılmaya dayalıydı ve bilimden çok bir sanattı. 1800'ün ortalarında, matematik ilk kez, geri-beslemeli kontrol sistemlerinin kararlılığını analiz etmek için kullanıldı. 1840'da, Greenwich'te İngiliz kraliyet gök bilimci G.B. Airy, bir teleskopu kontrol etmek için geri-beslemeli bir araç geliştirdi. Bu araç dünyanın dönüşüne uygun olarak otomatik kontrolle teleskopu ayarlayan bir hız kontrol sistemiydi. Fakat Airy, sisteme kötü osilasyonların ve bozucu etkilerin girdiğini ve geri-beslemeli kontrol çevriminin uygun olmayan yönünü keşfetti. Bu kapalı çevrim sistemlerinin kararlılığını tartışan ve analizinde diferansiyel denklemleri kullanan ilk kişiydi. Daha sonraları diferansiyel denklem teorisi iyi bir şekilde geliştirildi. Bu, I. Newton (1642-1727), G.W. Leibniz (1646-1716), J.F. Riccati (1676-1754), Bernoulli kardeşlerin ve diğerleri tarafından sonsuz küçük hesabının keşfi sayesinde oldu. Dinamik sistemlerin hareketinin analiz edilmesinde diferansiyel denklemlerin kullanımı, J.L. Lagrange (1736-1813) ve W.R. Hamilton (1805-1865) tarafından yapıldı.

Tam olarak Kararlılık teorileri de, 1800 yılının ortalarından itibaren geliştirilmiştir. Bu konuda, ilk olarak J.C Maxwell'in 1868'deki çalışması olmuştur. Bu çalışmada, doğrusallaştırılmış diferansiyel denklemi kullanarak sistemin karakteristik denklemini incelemiştir. Karakteristik denklemin kararlılığa etkisini incelemiştir. 1877'de E.J. Routli, nümerik bir teknik kullanarak kararlılığı analiz etmiştir. 1877'de Rus I.I. Vyshnegradsky, Maxwell'den bağımsız olarak diferansiyel denklemleri kullanarak regülatörlerin kararlılığını analizini yaptı. A.M. Lyapunov'un çalışması, kontrol teorisinde büyük bir kaynaktır. 1892'de genelleştirilmiş enerji fikrini kullanarak doğrusal olmayan diferansiyel denklemlerin kararlılığı incelenmiştir.

Haberleşme sistemlerindeki gelişmeler 20. yüzyılın başlarından itibaren tüm alanlarda etkisini göstermiştir. Zaman düzleminde diferansiyel denklemler kullanılarak, kontrol sistemlerinin matematiksel analizleri çıkarılmıştır. P.S. de Laplace (1749-1827), J. Fourier (1768-1830), A.L. Cauchy (1789-1857) ve diğerleri tarafından geliştirilen frekans düzlemi yaklaşımları, 1920 ve 1930 yılları arasında Bell Telefon laboratuvarlarında keşfedilip, haberleşme sistemlerinde kullanılmıştır. Uzaklara sesi göndermek için kullanılan kuvvetlendirici gereksinimi idi. Ancak, uygulamada ses ile beraber gürültü de kuvvetlendirilerek gönderiliyordu. Böylece, uygun tekrarlayıcı kuvvetlendirici tasarımı başlıca öneme sahip oldu. 1927'de, H. S. Black kuvvetlendiricideki bozulmayı azaltmak için negatif geri beslemenin kullanımını gösterdi. 1932'de H.Nyquist, kararlık kuvvetlendirici tasarımı için tekrar elde etme teorisini geliştirdi. 1938'de H.W. Bode, kompleks bir fonksiyonun genlik ve frekans cevabı çizimlerini kullandı. Genlik ve faz fikrini kullanarak kapalı çevrim kararlılığını araştırmıştır (Becerikli, 1998).

Dünya savaşları sırasında geri beslemeli kontrol teknikleri gelişerek savaş sistemlerinde kullanıldı. O sıralarda önemli askeri problem, gemilerin kontrolü ve seyriydi. 1910'da E.A. Sperry jireskopu keşfetti ve gemilerin kullanımı ve kararlılığında ve sonraları uçak kontrolünde kullandı. 1922'de N. Minorsky, gemilerin kullanımı için üç-parçalı Oransal İntegral Türev (PID) kontrolcüyü ilk olarak kullandı. Kapalı çevrimdeki doğrusal olmayan etkileri düşünmüştü. Dünya savaşları sırasında karşılaşılan önemli sorunlardan biri de, gemi ve uçaklardan atılan silah ve bombaların doğru olarak hedeflere atılmasıdır. 1934'de H.L. Hazen servo mekanizmaları teorisi konusunda yaptığı araştırma bu şekildeki problemlere ilk olarak mekanik kontrol teorisinin kullanılması olmuştur.

1940'da MIT'de radyasyon laboratuvarları kuruldu. 1941'de MIT/Sperry Kurumu'nda katıldığı proje üzerinde çalışırken, A.C. Hall, kontrol sistemi tasarımında gürültüyü göz önüne almanın kötü etkilerini keşfetti. Transfer fonksiyonu blok diyagramı ve frekans düzlemi metotlarına dayalı tasarım yaklaşımları kullanarak, Radyasyon Laboratuvarlarında kontrol tasarımlarında büyük başarılar yakaladı. 1948'de Kuzey Amerika Havacılık'ta çalışan W.R. Evans, köklerin yer eğrileri tekniğini sundu. Bu teknik "s" ortamında kapalı çevrim kutupları yerleştirmeye imkan veriyordu. Uzay çağının gelmesiyle ABD'de kontrol tasarımı tekrar diferansiyel denklem tekniklerine geri döndü ve teknikler zaman ortamındaydı.

Gelişmelerin ilerlemesiyle Kalman teorisi ile kontrol sistemlerinin performansını garantileyen optimal bir çözüm sağlandı. Doğrusal olmayan kontrol teorisi konusunda çok önemli çalışmalar yapılmıştır (Lewis, 1992; Leung and Chan, 2003).

Kontrol, otomatik kontrol, optimal kontrol, uyarlanabilir kontrol, öngörülü kontrol, geri besleme, kararlılık ve olasılıkla kontrol konularında tarihsel gelişimleriyle beraber daha detaylı bilgiler literatürde mevcuttur (Clarke et al., 1987; Henson and Seborg, 1997; Sarıoğlu, 1985).

Kontrol teorisinde son yıllarda yeni teknikler girmiştir. Bu yeni tekniklerden en önemlilerinden birisi yapay sinir ağları (YSA) teknolojisidir. Daha önceleri temeli atılmış olsa bile, YSA 1985'den sonra uygulama alanlarına girmiştir. YSA insan beyni çalışmasından esinlenerek çok çeşitli YSA yapısı geliştirilmiştir. Bunların hepsinde işlem elemanları olan nöronlar ve bunların ara bağlantı parametreleri olan ağırlıklar mevcut durumdadır. Bir çok amaç için kullanılabilir. Örneğin; sınıflama, ayırma, tanıma, kontrol, kestirim, görüntü işleme, ses işleme vb.

Yapısal olarak sinir ağıları iki tipte incelenebilir. Bunlardan birincisi ileri beslemeli (İB) sinir ağılarıdır. Burada nöronlar gizli katmanlardadır. Bir katmandan sonraki katmana ağırlıklarla bağlantı yapılıır. İB sinir ağıları birkaç türe sahiptir. Bunlardan birisi çok katmanlı algılayıcı (ÇKA) ağı, diğeri öğrenme vektörü kuvantizeli ağı gibi. İkinci tip YSA ise, geri beslemeli (GB) veya dinamik sinir ağıları (DSA)'dır .En popüler olanı Hopfield ağıdır. Burada DSA dinamik hafızaya sahiptir.

Bu algoritmalar da kullanılan çok katlı ağılar tipi YSA'ların eğitiminde en çok kullanılan yöntem Geriye Yayınım (GY) algoritmasıdır (Henson and Seborg, 1997).

Öngörülü kontrol yapısında YSA süreç modeli olarak kullanılmaktadır. Bu amaçla, seri-paralel tanımlama modeli yapısı kullanılarak, YSA, GY algoritması ile eğitilmiştir. Eğitim işlemi çevrim dışı (off-line) olduğundan, daha hızlı eğitim algoritmalarına ihtiyaç duyulmamıştır.

Yapılan bu tez çalışmasının organizasyonu aşağıdaki gibidir;

2. Bölümde optimal kontrol yapısı ve kavramı anlatılmıştır. Bu kısımda Hamiltonian sistemi ve çözümü, doğrusal ve doğrusal olmayan uygulamalar bulunmaktadır.

3. Bölümde öngörülü kontrol yöntemi tanıtılmış temel prensipleri verilmiştir. Optimal kontrol ile ilişkisine değinilmiştir. Çeşitli model öngörülü algoritma yöntemlerinin analizi, bunlar sırasıyla Model Algoritmik Kontrol (MAK), Dinamik Matris Kontrol (DMK) ve Genelleştirilmiş Öngörülü Kontrol (GÖK) yapıları incelenerek çeşitli uygulamalar yapılmıştır.

4. Bölümde, YSA konusu ele alınmıştır. Süreç kontrolünde önemli yeri olan YSA'larının eşleme, paralellik, öğrenme ve uyarlanma gibi

özelliklerini içeren bir ağ kurulması, aynı zamanda GY'la yapılmış olan eğitim yönteminden bahsedilmiştir.

5. Bölümde ise bu çalışmanın adını veren yapay sinir ağları temelli model öngörülü kontrol yapısı verilmiştir. Dinamik sistemlerin YSA ile modellenmesi gösterilmiştir. Model Öngörülü Kontrol (MÖK) için kullanılan performans ölçütü tanıtılmış, öngörü ve kontrol ufku nasıl seçileceği açıklamıştır. YSA temelli MÖK yapısı kullanılarak yapılan doğrusal ve doğrusal olmayan uygulamaların sonuçları verilmiştir.

6. Bölümde ise çalışmanın sonuçları ve öneriler verilmiştir.



2. OPTİMAL KONTROL TEKNİKLERİ

2.1 Giriş

Optimal kontrol algoritmaları modern kontrol tekniğinin temel yapısı olarak adlandırılır ve zaman ortamında yapılan bir kontrol tekniğidir. Bu konuda günümüze kadar bir çok çalışma yapılmış ve devam etmektedir (Lewis, 1992).

Doğrusal kuadratik regülatör “DKR” (Linear Kuadratik Regulator, LQR) problemini çözebilmek için Lagrange katsayılarına bağlı bir çözüm geliştirilmiştir. Doğrusal olmayan optimal kontrol problemi hesaplanması konusunda Euler-Lagrange denkleminin genelleştirilmesi yoluyla çözülebilmektedir. Doğrusal veya doğrusal olmayan optimal kontrol problemlerinin bir başka çözümü Riccati diferansiyel denklemleri ile çözülebilmektedir (Sarıoğlu, 1985; Chang and Krogh, 2001).

Optimal kontrol problemi performans ölçütünün belirlenerek, kontrol sinyalinin bir sonraki adımda alacağı değerin, maliyet fonksiyonunu en küçük yapacak şekilde hesaplamasına dayanır. Genel optimizasyon yapılarından başlanıp, doğrusal ve doğrusal olmayan optimal kontrol problemleri incelenmiştir. Hamiltonian prensibine dayalı olarak çözülen bu problemler, literatürde Hamiltonian prensibine dayalı yöntemler olarak geçmektedir.

Hamiltonian sisteminin yazılabilmesi için elimizde durum denklemlerinin, yardımcı durum denklemlerinin ve buna bağlı olarak da Lagrange katsayılarının bilinmesi gerekmektedir. Bu temel yapılar hesaplandıktan sonra algoritmanın oluşturulması gerekmektedir.

2.2 Modern Kontrolün Felsefesi

Modern kontrol teknikleri zaman ortamında tasarlanmaktadır. Dolayısıyla sistemin durum uzayı modelinde kontrol edilebilirliğini

bilmek şarttır. Durum uzayında dinamik bir sistem ifadesi aşağıdaki gibi ise;

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\quad (2.1)$$

Eşitlik 2.1 için $x(t)$ durum vektörü, $u(t)$ kontrol giriş vektörü, $y(t)$ çıkış vektörü, olarak zaman düzleminde açıklanabilir. Çoklu girişli – çoklu çıkışlı “ÇGÇÇ” (Multi Input – Multi Output, MIMO) sistemler ile tek girişli – tek çıkışlı “TGTÇ” (Single Input – Single Output) sistemler temelde $u(t)$ ve $y(t)$ genel olarak vektörleridir. Eşitlik 2.1’deki A, B ve C matris elemanları sistemin dinamik olarak birbirine bağımlı elemanlarıdır.

2.3 Doğrusal Sistemler

Modern kontrol teknikleri ilk olarak doğrusal sistemler için yapılmıştır. Doğrusal olmayan sistemlerde dinamik programlama kullanılmaktadır. Doğrusal olmayan sistemler için açık-çevrim optimal kontrol tasarımı doğrusal olmayan iki nokta-sınır değer problemi ile çözülmektedir.

Geri besleme kontrol formu Eşitlik 2.2’de verilmiştir.

$$u = -Kx \quad (2.2)$$

Eşitlik 2.2’de, geri besleme kazancı K matrisidir ve elemanları sistemin kendine özgü kontrol kazançlarıdır. Bütün durumlar geri besleme için kullanılmaktadır ve durum geri beslemesi olarak adlandırılır. Eğer n tane durum ve m tane skaler kontrol değeri varsa, K matrisi $m \times n$ boyutundadır.

Standart bir doğrusal kuadratik regülatörde, geri besleme kazanç matrisi K, Eşitlik 2.3’de verilen Performans Ölçütünü “PÖ”

(Performance Index – PI) en küçük yapacak şekilde belirlenir (Kajima and Ishijm, 1999).

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (2.3)$$

Eşitlik 2.3’de Q ve R ağırlık matrisi dizayn parametreleridir. Bu elemanlar performansa uygun seçilmelidir. Doğrusal kuadratik regülatör dizaynında en önemli nokta K matrisinin iyi seçimi ve performans ölçütünü sonlu bir değere götürmesi olacaktır. Eşitlik 2.3’de u(t) ve x(t) performans ölçütünün sınırlarını belirler. Eğer bu değerler zamanla sifıra gidiyorsa, kapalı çevrim kararlılığı garantilenir.

K’nın değeri performans ölçütünün en küçük değer yapacak olan, verilen değerler ışığında Eşitlik 2.4 yazılabilir.

$$K = R^{-1} B^T S \quad (2.4)$$

Riccati eşitliğinden elde edilen çözümde S matrisinin boyutu n x n olarak gösterilmiştir. Burada denge durumunda olması beklenen Q ve R matrislerinin değerlerini bulmak için Eşitlik 2.5’den faydalanılır.

$$0 = A^T S + SA - SBR^{-1}B^T S + Q \quad (2.5)$$

Eşitlik 2.5’deki doğrusal kuadratik çözüm çerçevesinde, Q ve R nin kontrol edilebilir seçimleri ile kapalı çevrim kazancı Eşitlik 2.5’deki gibi olur. Bundan sonra K katsayısının etkisi durum denkleminde Eşitlik 2.6’deki gibi gösterilebilir.

$$\dot{x} = (A - BK)x + Bu \quad (2.6)$$

2.4 Doğrusal Olmayan Sistemler

Doğrusal olmayan dinamik bir sistem için durum uzayı modeli Eşitlik 2.7’deki gibi verilsin.

$$\dot{x} = f(x, u, t), \quad t \geq t_0 \quad t_0 \text{ sabit} \quad (2.7)$$

Burada $x(t) \in \mathcal{R}^n$ iç durum vektörü, $u(t) \in \mathcal{R}^m$ kontrol girişlerinin vektörüdür ve bu saha kontrol edilebilir saha olması gerekir. Performansı belirleyen $u(t)$ kontrol objesi performans ölçütün (PÖ) veya verilen maliyet fonksiyonu değerini minimuma götürmesi gerekir.

$$J(t_0) = \phi(x(T), T) + \int_{t_0}^T L(x(t), u(t), t) dt \quad (2.8)$$

Eşitlik 2.8'de t_0 başlangıç zamanı T ise alınacağı son zaman değeridir. Son durum ağırlık fonksiyonu $\phi(x(T), T)$ ve ağırlık fonksiyonu $L(x, u, t)$ değerleri seçilen performans objesine bağımlıdır.

Optimal kontrol problemini çözerken girişe uygulanan kontrol vektörü $u(t)$ sistemin PÖ değerini minimuma götürürken son durum sınır değerini göstermektedir.

$$\psi(x(T), T) = 0 \quad (2.9)$$

Eşitlik 2.9'daki $\psi(t) \in \mathcal{R}^p$ zamanın bir fonksiyonu olarak tanımlanır. “ ϕ ” ve “ ψ ” değerleri birbirine karıştırılmamalıdır. Birincisi çok küçük bir değer olması istenir, ikincisinin ise son değerinde sıfır olması istenir.

Optimal kontrol problemini çözebilmek için Lagrange katsayılarını belirlemek gerekir. Bunun için Eşitlik 2.7 ve 2.9, performans ölçütünün Eşitlik 2.8'deli denklemlerinden faydalanılır. İlk olarak belirlenecek bir katsayıya ihtiyaç vardır. Bu katsayı $\lambda(t) \in \mathcal{R}^n$ zamanın bir fonksiyonudur. Son durum sınırını belirlemesi amacıyla ise yeni bir katsayı belirlenir bu katsayı $v(t) \in \mathcal{R}^p$ dir. Bu durumdan dolayı PÖ Eşitlik 2.10'daki gibi olur.

$$J = \phi(\mathbf{x}(T), T) + \mathbf{v}^T \psi(\mathbf{x}(T), T) + \int_{t_0}^T [\mathbf{L}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T(t)(\mathbf{f}(\mathbf{x}, \mathbf{u}, t) - \dot{\mathbf{x}})] dt \quad (2.10)$$

Bu durum Hamiltonian fonksiyonu ile açıklanabilir, Eşitlik 2.11 deki gibi düzenlenebilir.

$$H(\mathbf{x}, \mathbf{u}, t) = \mathbf{L}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T(t)\mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (2.11)$$

$$J = \phi(\mathbf{x}(T), T) + \mathbf{v}^T \psi(\mathbf{x}(T), T) + \int_{t_0}^T [\mathbf{H}(\mathbf{x}, \mathbf{u}, t) - \boldsymbol{\lambda}^T(t)\dot{\mathbf{x}}] dt \quad (2.12)$$

Lagrange katsayıları teorisine göre kontrol fonksiyonu $\mathbf{u}(t)$, PÖ Eşitlik 2.8 denklemini minimuma götürür.

Doğrusal olmayan sistemler için Hamiltonian denklemlerinin hesaplanması sırasıyla yapılabilir. İlk olarak, gereken durum denklemini yazmaktır, Eşitlik 2.13.

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \boldsymbol{\lambda}} = \mathbf{f}, \quad t \geq t_0 \quad (2.13)$$

Bundan sonraki adımda yardımcı durum denklemlerini yazmak gerekmektedir, Eşitlik 2.14.

$$-\dot{\boldsymbol{\lambda}} = \frac{\partial H}{\partial \mathbf{x}} = \frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \boldsymbol{\lambda} + \frac{\partial \mathbf{L}}{\partial \mathbf{x}}, \quad t \leq T \quad (2.14)$$

Bir sonraki adımda durağan durum denkleminde elde edilecek olan Eşitlik 2.15 bulunur.

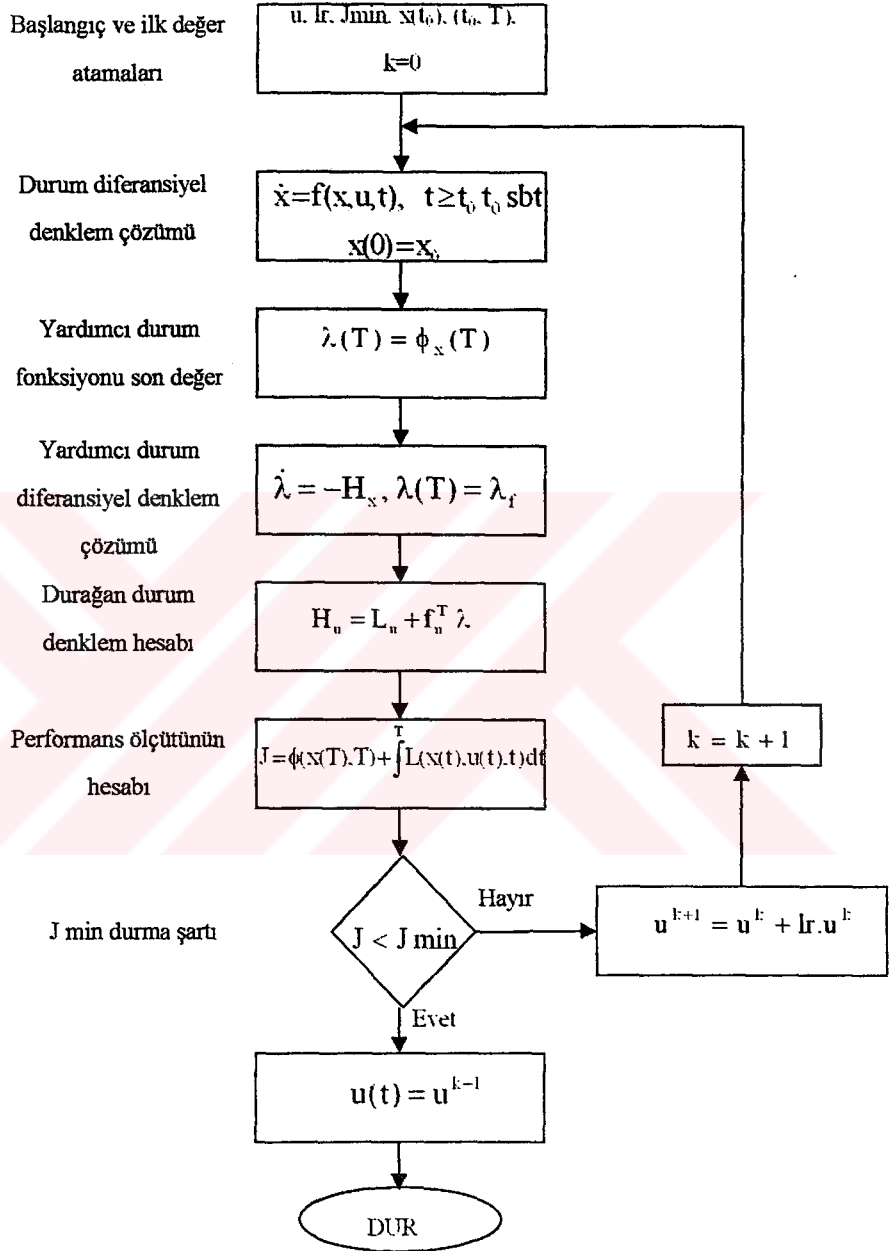
$$0 = \frac{\partial H}{\partial \mathbf{u}} = \frac{\partial \mathbf{L}}{\partial \mathbf{u}} + \frac{\partial \mathbf{f}^T}{\partial \mathbf{u}} \boldsymbol{\lambda} \quad (2.15)$$

En son adım olarak da sınır koşullarının belirlenmesi gerekiyor, bu şartta Eşitlik 2.16'da verilmiştir.

Çizelge 2.1 Doğrusal olmayan optimal kontrolör tasarım tablosu

Sistem modeli	$\dot{x} = f(x, u, t), \quad t \geq t_0 \quad t_0 \text{ sabit}$
Performans ölçütü	$J(t_0) = \phi(x(T), T) + \int_{t_0}^T L(x(t), u(t), t) dt$
Son durum sınırı	$\psi(x(T), T) = 0$
Optimal kont. Hamiltonian	$H(x, u, t) = L(x, u, t) + \lambda^T(t) f(x, u, t)$
Durum denklemi	$\dot{x} = \frac{\partial H}{\partial \lambda} = f, \quad t \geq t_0$
Yardımcı durum denklemi	$-\dot{\lambda} = \frac{\partial H}{\partial x} = \frac{\partial f^T}{\partial x} \lambda + \frac{\partial L}{\partial x}, \quad t \leq T$
Durağan durum denklemi	$0 = \frac{\partial H}{\partial u} = \frac{\partial L}{\partial u} + \frac{\partial f^T}{\partial u} \lambda$
Sınır koşulu	$(\phi_x + \psi(v - \lambda))^T \Big _T dx(T) + (\phi_x + \psi(v + H))^T \Big _T dT = 0 \quad x(t_0) \text{ belli}$

$$\begin{aligned}
 & (\phi_x + \psi(v - \lambda))^T \Big|_T dx(T) \\
 & + (\phi_x + \psi(v + H))^T \Big|_T dT = 0 \quad x(t_0) \text{ belli}
 \end{aligned} \tag{2.16}$$



Şekil 2.1 Optimal kontrolör Hamiltonian algoritması

Bu eşitlikte sınır şartları belirlenebilir. Bütün gereken hesaplamalar ve izlenecek yol Çizelge 2.1'de sırasıyla verilmiştir. Bu problemin çözümü için gerekli algoritma Şekil 2.1'de verilmiştir.

2.4.1 İki Nokta Sınır Değer Problemi

Çizelge 1.1'deki $u(t)$ probleminin çözümü iki çift diferansiyel denkleme yani durum ve yardımcı durum denklemlerine bağlıdır. Bu dinamik denklem takımlarından oluşturulmuştur.

$$\dot{x} = f(x, u, t), \quad t \geq t_0 \quad x(0) = x_0 \quad (2.17)$$

$$-\dot{\lambda} = \frac{\partial H}{\partial x} = \frac{\partial f^T}{\partial x} \lambda + \frac{\partial L}{\partial x}, \quad t \leq T \quad \lambda(T) = \lambda_f \quad (2.18)$$

Hamilton sistemi ilk önce durağan durum denklemindeki $u(t)$ çözülür (Lewis, 1992). Burada sınır koşulları

$$n \text{ durum: } x(t_0) \text{ belli} \quad (2.19)$$

$$p \text{ durum: } \psi(x(T), T) = 0 \quad (2.20)$$

$$n-p \text{ durum: } (\phi_x + \psi_x^T v - \lambda)^T \Big|_T dx(T) = 0 \quad (2.21)$$

Son zaman değeri "T" değişmediğine göre $dT=0$ olarak Eşitlik 2.16'dan alınabilir. Burada "n" sınır durumu son değerindeki T'nin, t_0 başlangıç zamanı ve n durumunun oluşturduğu probleme iki nokta sınır değer problemi olarak bilinir. Bu problemin aşımı için bir çok çözüm yöntemi vardır. Bunlardan birisi olan, atış nokta metodu (shooting - point method) uygun çözüm yöntemi olarak kullanılabilir, Örnek 2.1'de gösterilmektedir.

2.4.2 Optimal Kontrol Uygulamaları

Optimal kontrolör tasarımına ilişkin iki örnek verilmiştir. Birincisi, armatür kontrollü doğru akım motoru, ikincisi ise eş ısı CSTR'dir.

2.4.2.1 Armatür Kontrollü Doğru Akım Motoru

Örnek 2.1: Armatür Kontrollü DC Motor için Hamiltonian Sisteminin Nümerik Çözümü yapılacaktır. İlk olarak durum denklemleri yazılır, Eşitlik 2.22.

$$\begin{aligned} L\dot{i} &= -Ri - vb + u \\ J_m\dot{\omega} &= T_m - T_L - b_m\omega \end{aligned} \quad (2.22)$$

Motorun dönme momenti ve ters elektro motor kuvveti aşağıdaki gibidir.

$$\begin{aligned} T_m &= k_m i \\ vb &= k'_m \omega \end{aligned} \quad (2.23)$$

Burada $i(t)$ armatür akımı, $\omega(t)$ motorun hızı ve $u(t)$ kontrol girişi, $T_L=0$ olarak alınacaktır. Uygun bir şekilde problemi tekrar yazarsak elde edeceğimiz durum denklemleri Eşitlik 2.24'de verilmiştir.

$$\begin{aligned} \dot{i} &= -ai - k'\omega + bu \\ \dot{\omega} &= -\alpha\omega + ki \end{aligned} \quad (2.24)$$

Verilen değerler $a = R/L$ elektriksel zaman sabiti ve $\alpha = b_m / J_m$ mekaniksel zaman sabitidir. Durum vektörü $x = [i \ \omega]^T \in \mathbb{R}^n$ ve $n=2$ şeklinde yazılabilir. Kontrol objesi için uygun olan PÖ Eşitlik 2.25'deki şekilde verilmiştir.

$$J = \frac{1}{2}s(\omega(T) - \omega_T)^2 + \frac{1}{2}\int_0^T u^2 dt \quad (2.25)$$

Burada "s" sabit dizayn parametresidir ve u^2 kontrol enerjisinin minimum en küçük enerjisi ile $\omega(T)$ değerini ω_T 'ye yaklaştırması için gerekli değerdir. ω_T motorun hızının istenilen son değeridir.

Bu kısımda, atış nokta metodunu uygularken bize Hamiltonian ve sınır şartları yeterli olacaktır. Burada Hamiltonian denklem takımı Eşitlik 2.26'da olduğu gibidir.

$$H = \frac{1}{2}u^2 + [\lambda_i \quad \lambda_w] \begin{bmatrix} -ai & -k'\omega & +bu \\ -\alpha\omega & +ki \end{bmatrix} \quad (2.26)$$

Eşitlik 2.26'da hesaplanacak olan Lagrange katsayıları vektör olarak $\lambda = [\lambda_i \quad \lambda_w]^T \in \mathfrak{R}^2$ şekilde gösterilebilir. Yardımcı durum denklemleri, Eşitlik 2.27 gibi yazılabilir.

$$\begin{aligned} \dot{\lambda}_i &= \frac{-\partial H}{\partial i} = a\lambda_i - k\lambda_w \\ \dot{\lambda}_w &= \frac{-\partial H}{\partial \omega} = a\lambda_w - k'\lambda_i \end{aligned} \quad (2.27)$$

Bundan sonraki adımda durağan durum denklemi yazılarak “u” kontrol sinyali için Eşitlik 2.28 elde edilir.

$$\begin{aligned} 0 &= \frac{\partial H}{\partial u} = u + b\lambda_i \\ u &= -b\lambda_i \end{aligned} \quad (2.28)$$

Eşitlik 2.24 ve 2.28 denklemlerinden yararlanılarak “u” değerini yerine yazarsak Hamiltonian sistemini elde etmiş oluruz.

$$\begin{aligned} \dot{i} &= -ai - k'\omega + bu \\ \dot{\omega} &= -\alpha\omega + ki \end{aligned} \quad (2.29)$$

Denklemleri tekrar düzenleyecek olursak Eşitlik 2.29 ve 2.30 oluşturulur.

$$\begin{aligned}\dot{\lambda}_i &= \frac{-\partial H}{\partial i} = a\lambda_i - k\lambda_\omega \\ \dot{\lambda}_\omega &= \frac{-\partial H}{\partial \omega} = a\lambda_\omega - k'\lambda_i\end{aligned}\quad (2.30)$$

Başlangıç koşulları Eşitlik 2.31'de verilmiştir.

$$\begin{aligned}i(0) &= 0 \\ \omega(0) &= 0\end{aligned}\quad (2.31)$$

Yardımcı durum denklemlerinin son değerlerinin sıfır olması gereklidir.

$$\begin{aligned}\lambda_i(T) &= \frac{\partial \phi}{\partial i} \Big|_T = 0 \\ \lambda_\omega(T) &= \frac{\partial \phi}{\partial \omega} \Big|_T = s(\omega(T) - \omega_T)\end{aligned}\quad (2.32)$$

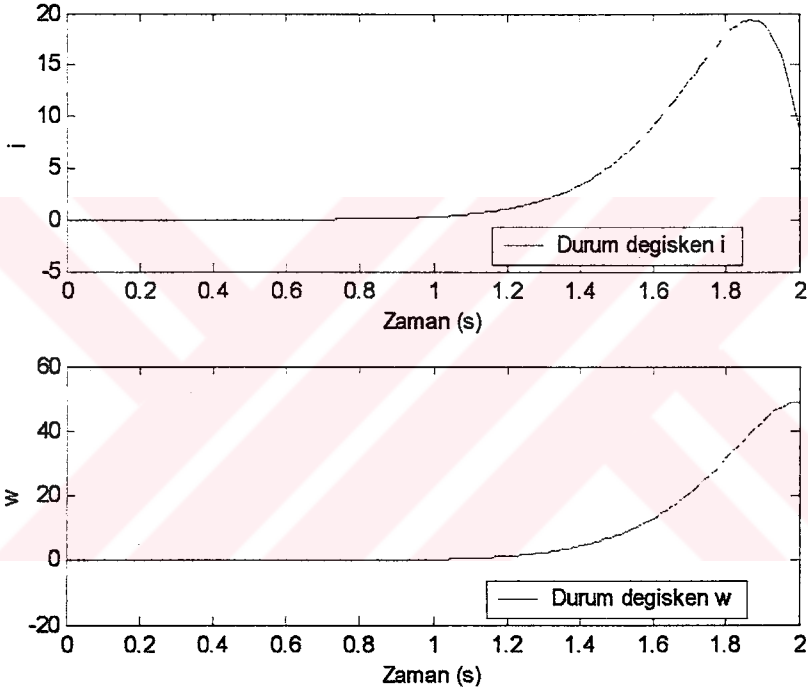
Hamiltonian sisteminin toplam durum gösterimi $X = [i \ w \ \lambda_i \ \lambda_\omega]^T$ şeklindedir ve $N=2n$ bu sistemin boyutudur.

\dot{X} bulunurken $N \times N$ boyutunda Hamiltonian sisteminin Jakobyeni Eşitlik 2.33'deki gibi hesaplanmış olur.

$$\dot{X} = \begin{bmatrix} -a & -k' & -b^2 & 0 \\ k & -\alpha & 0 & 0 \\ 0 & 0 & a & -k \\ 0 & 0 & k' & \alpha \end{bmatrix} X = JAC \cdot X \quad (2.33)$$

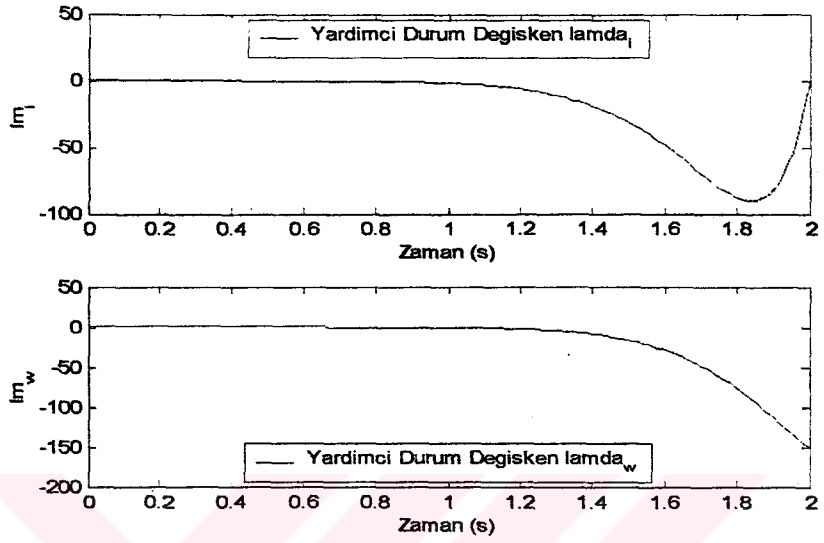
Motorun parametreleri, $j_m = 0.1 \text{ kg.m}^2$, $k_m = k'_m = 1 \text{ V.s}$, $L = 0.5 \text{ H}$, $b_m = 0.2 \text{ N.m.rad/s}$ ve $R=5\Omega$. Son istenen hız değeri $\omega(T) = 50 \text{ rad/s}$ ve simülasyon zamanı $T = 2\text{s}$. Dizayn parametresi olan "s" değerinin en iyi sonucu 100 olarak verilmiştir ve $\omega(T) = 48.499 \text{ rad/s}$ olarak bulunmuştur.

Dizayn parametresi “s” büyütülmesi istenen $\omega(T)$ 'ye yaklaştırır fakat performans ölçütündeki kontrol enerjisi u^2 nin çok büyümesine sebep olur. Bu verilen nümerik değerler kullanılarak gerçekleştirilen benzeşim çalışması sonucu durum değişkenlerinin değişimi Şekil 2.2’de gösterilmiştir.

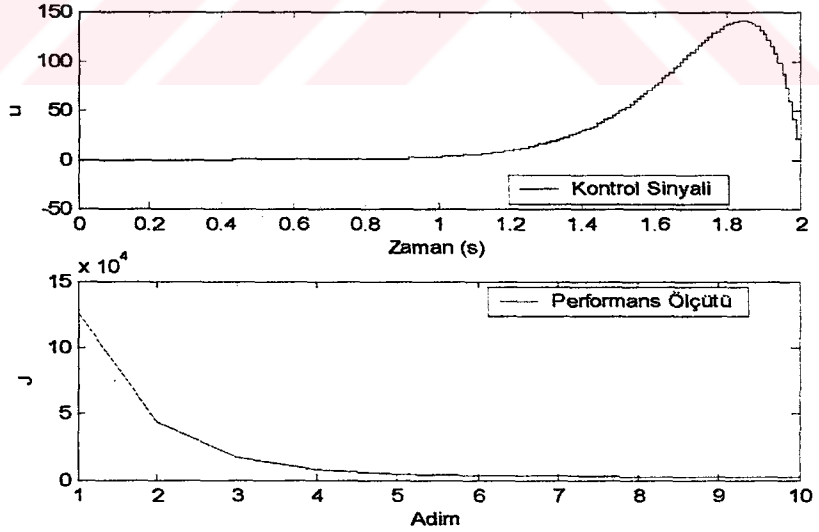


Şekil 2.2 Örnek 2.1 için DC motor durum denklemleri

Yardımcı durum denklemleri için ise simülasyon sonuçları Şekil 2.3’de verilmiştir. Burada $s=100$ alınması ile Lagrange katsayıları $\lambda_i = 0$, $\lambda_w = -150$ değerine ulaşmaktadır.



Şekil 2.3 Örnek 2.1 için yardımcı durum denklemleri



Şekil 2.4 Örnek 2.1 için $u(t)$ ve PÖ

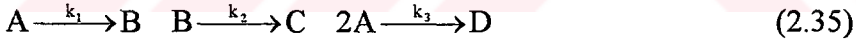
Örnek 2.1 için Performans ölçütü PÖ ve kontrol sinyali u 'nun değişimi Şekil 2.4'de gösterilmiştir. Görüldüğü üzere performans ölçütü ilk birkaç adım içerisinde en küçük değerine ulaşmıştır. Kontrol sinyali u 'nun değişimi DC motorun uyarım sargılarına uygulanan gerilim temsil etmektedir.

2.4.2.2 Eşışıl Reaktor (CSTR)

Örnek 2.2: Eş ısııl reaktörün (Isothermal Continuous Stirred Tank Reactor, CSTR) Hamiltonian sistemi ile nümerik çözümü yapılacak olursa. İlk olarak reaktörün verilen durum denklemleri ve başlangıç şartları $x_1(0) = 2.9997, x_2(0) = 1.11697$ yazılır (Henson and Seborg, 1997).

$$\begin{aligned}\dot{x}_1 &= -k_1 x_1 - k_3 x_1^2 + (x_F - x_1)u \\ \dot{x}_2 &= k_1 x_1 - k_2 x_2 + x_2 u\end{aligned}\quad (2.34)$$

Bu durum denklemleri Eşitlik 2.35'deki reaksiyon denklemlerinin çözümünden çıkarılmış modelleridir.



$x = [x_1 \ x_2]^T \in \mathfrak{R}^n$ olduğuna göre $n=2$ olarak alınacaktır. Durum

değişkenleri x_1 ve x_2 (Eşitlik 2.34), A ve B karışımlarının iki türünü ifade eder (Eşitlik 2.35). x_F , A konsantrasyonuna yapılan besleme girişi, u ise besleme oranını gösterir. Sırasıyla $\{k_1, k_2, k_3, x_F\}$ parametrelerinin nominal değerleri $\{50 \text{ hr}^{-1}, 100 \text{ hr}^{-1}, 10 \text{ liter}/(\text{mol hr}), 10 \text{ mol/liter}\}$ olarak verilmiştir. Öncelikli olarak kontrol objesi olarak çıkış değerinin sabit tutulması ve değerinin $y = x_2 = 1.0$ 'de tutulması istenmiştir.

Eşitlik 2.34'de verilen durum denklemleri takımlarının denge durumu çözümü yapıldığında bulunacak olan $\{x_1, x_2, u\}$ değerleri ile

istenilen nümerik çözüm takımına ulaşılmış olur. Bu çözüm takımlarından iki çözüm sonucuna ulaşılmaktadır. Bunlardan birincisi {2.5, 1.0, 25} ve bir diğeri {6.6667, 1.0, 233.33} sonuçlarıdır. Fakat bizden istenilen nesnel fonksiyon ağırlığı, en az belseme oranı “u” ile yapılan çözüm olduğuna göre seçilecek takım, birinci denklem çözüm takımı olan {2.5, 1.0, 25}dir.

Durum denklem modelinin çözümü istenen $\{x_1, x_2, u\}$ kararlı noktaları bulunmuştur. Buna göre performans ölçütü PÖ yazılacak olursa (Eşitlik 2.36).

$$J = \int_t^{t+T} \gamma_1 [x_1(\tau) - 2.5]^2 + \gamma_2 [x_2(\tau) - 1.0]^2 + \gamma_3 [u(\tau) - 25]^2 d\tau \quad (2.36)$$

Eşitlik 2.36’da “ x_1 ve x_2 ” durum değişkenleri ve kontrol sinyali olan “u” kontrolördür. Bu denklemdeki $\{\gamma_1, \gamma_2, \gamma_3\}$ katsayıları PÖ’nün ölçeklendirilmesi ve değişkenlerin ağırlıklarının belirlenmesi sebebiyle, her birinin katkısı önemlidir. “T” ulaşılacak olan son zaman değeri, “t” ise başlangıç zaman değeridir.

Doğrusal olmayan bu denklem sistemi için, PÖ ağırlık değerleri Eşitlik 2.37’deki gibi verilmiştir.

$$\gamma_1 = \gamma_2 = 1000, \gamma_3 = 1 \quad (2.37)$$

Durum denklemlerinin optimal kontrolör Hamiltonian sistemi (Eşitlik 2.11) çözümüne göre Eşitlik 2.38’deki gibidir.

$$H = \gamma_1 [x_1 - 2.5]^2 + \gamma_2 [x_2 - 1.0]^2 + \gamma_3 [u - 25]^2 + \begin{bmatrix} \lambda_{x_1} & \lambda_{x_2} \end{bmatrix} \begin{bmatrix} -k_1 x_1 & -k_3 x_1^2 & +(x_F - x_1)u \\ k_1 x_1 & -k_2 x_2 & -x_2 u \end{bmatrix} \quad (2.38)$$

Burada Hamiltonian sistemindeki hesaplanacak Lagrange katsayıları $\lambda = [\lambda_{x_1} \quad \lambda_{x_2}]^T \in \mathfrak{R}^2$ olarak gösterilir.

Yardımcı durum denklemlerini yazacak olursak Eşitlik 2.39'daki denklem takımı elde edilir.

$$\begin{aligned}\dot{\lambda}_{x_1} &= \frac{-\partial H}{\partial x_1} = -2\gamma_1(x_1 - 2.5) + k_1\lambda_{x_1} + 2\lambda_{x_1}k_3x_1 + u\lambda_{x_1} - \lambda_{x_2}k_1 \\ \dot{\lambda}_{x_2} &= \frac{-\partial H}{\partial x_2} = -2\gamma_2(x_2 - 1) + k_2\lambda_{x_2} + \lambda_{x_2}u\end{aligned}\quad (2.39)$$

Nümerik değerler $k_1 = 50$, $k_2 = 100$, $k_3 = 10$, $x_F = 10$ yerine Eşitlik 2.39'da yazılır. Bun dan sonra durağan durum denklemleri yazılarak kontrol sinyali “u” için (Eşitlik 2.15) denklemden Eşitlik 2.40'daki değer elde edilir.

$$0 = \frac{\partial H}{\partial u} = 2\gamma_3(u - 25) + (x_F - x_1)\lambda_{x_1} - x_2\lambda_{x_2}.\quad (2.40)$$

$$u = \frac{-50\gamma_3 + (x_F - x_1)\lambda_{x_1} - x_2\lambda_{x_2}}{2\gamma_3}$$

Hamiltonian sistemi için elde edilen denklemleri sırasıyla yazacak olursak Eşitlik 2.41 elde edilir.

$$\begin{aligned}\dot{x}_1 &= -k_1x_1 - k_3x_1^2 + (x_F - x_1)u \\ \dot{x}_2 &= k_1x_1 - k_2x_2 + x_2u \\ \dot{\lambda}_{x_1} &= \frac{-\partial H}{\partial x_1} = -2\gamma_1(x_1 - 2.5) + k_1\lambda_{x_1} + 2\lambda_{x_1}k_3x_1 + u\lambda_{x_1} - \lambda_{x_2}k_1 \\ \dot{\lambda}_{x_2} &= \frac{-\partial H}{\partial x_2} = -2\gamma_2(x_2 - 1) + k_2\lambda_{x_2} + \lambda_{x_2}u\end{aligned}\quad (2.41)$$

Bu çözüm sistemi için verilen başlangıç koşulları Eşitlik 2.42'deki gibidir.

$$\begin{aligned} x_1(0) &= 2.9997 \\ x_2(0) &= 1.11697 \end{aligned} \quad (2.42)$$

Yardımcı durum denklemlerinin son değerlerinin sıfır olması gerekli olduğuna göre Eşitlik 2.43'deki denklem takımı yazılır.

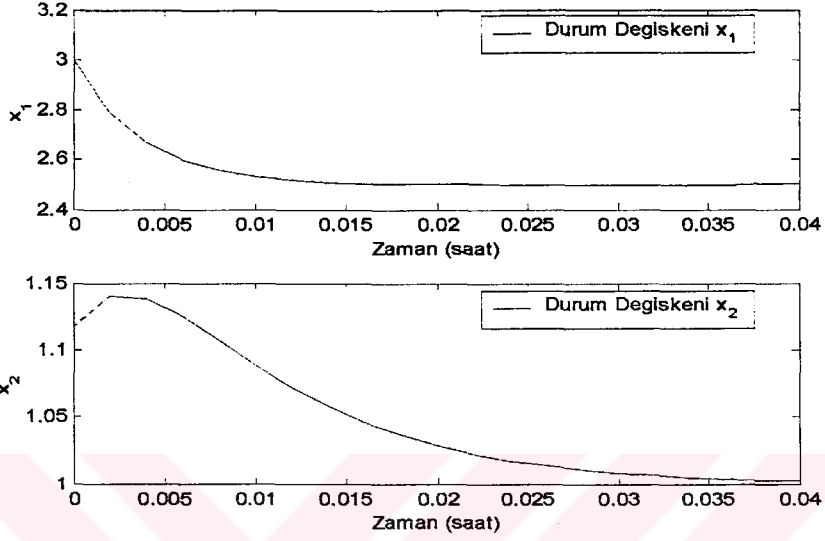
$$\begin{aligned} \lambda_{x_1}(T) &= \frac{\partial \phi}{\partial x_1} \Big|_T = 0 \\ \lambda_{x_2}(T) &= \frac{\partial \phi}{\partial x_2} \Big|_T = 0 \end{aligned} \quad (2.43)$$

Hamiltonian sistemini kullanarak elde edilen sonuçların Runge-Kutta algoritması ile nümerik çözümleri yapılmıştır. Kullandığımız, Runge-Kutta metodu zamanda ileri ve geriye doğru çözme yöntemiyle durum ve yardımcı durum denklemlerinin çözümü yapılmıştır. Burada "u" için yapılan seçim ilk adımda bir olarak tayin edilmiştir. Bundan sonraki aşamalarda "u" değeri güncellenerek her adımda yeni kontrol sinyali giriş değeri hesaplanmıştır. Şekil 2.5'de durum değişkenlerinin aldığı değerler gösterilmiştir.

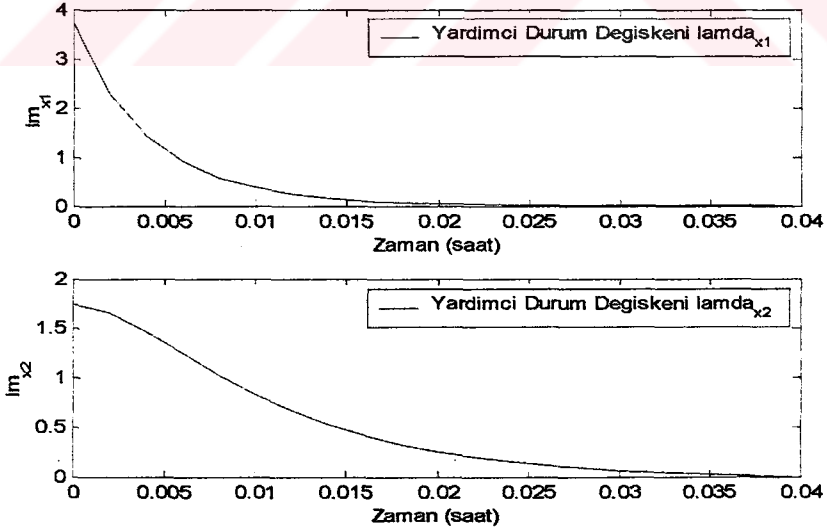
Bu sistemin çözülmesinde kullanılan yardımcı durum denklemlerinin elde edilen simülasyon sonuçları Şekil 2.6'da verilmiştir.

Elimizdeki doğrusal olmayan durum denklemlerinin Hamiltonian sistemi çözümünden elde edilen kontrol sinyali u'nun değişimi Şekil 2.7'de verilmiştir.

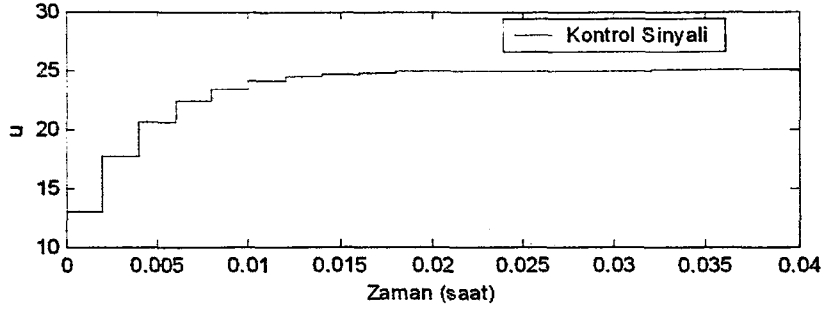
Aynı zamanda verilen PÖ'nün her adımdaki aldığı değerler görülmektedir (Şekil 2.8). Bu sonuçlar ışığında elde ettiğimiz çözüm doğrulanmaktadır (Henson et al., 1997).



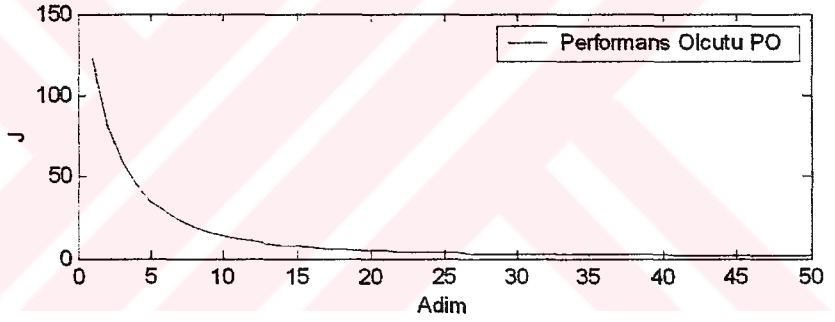
Şekil 2.5 Örnek 2.2 için (CSTR) durum değişkenleri



Şekil 2.6 Örnek 2.2 için (CSTR) yardımcı durum değişkenleri



Şekil 2.7 Örnek 2.2 için (CSTR) kontrol sinyali "u"



Şekil 2.8 Örnek 2.2 için (CSTR) performans ölçütü

3. MODEL ÖNGÖRÜLÜ KONTROL (MÖK) TEKNİĞİ

3.1 Giriş

Öngörülü kontrol algoritmalarında, değişik bozucuların da etkisinde minimum olmayan faz ve/veya değişen zaman gecikmeli süreçlere uygulanarak iyi sonuçlar almak mümkündür. Bu tip yapılarda ölü zaman kompanzasyonu mümkündür. Ölçülen bozucuların etkisini dengeleyebilmek, için ileri besleme kontrol işlemi yapılabilmektedir. Öngörülü algoritmaların kontrol problemlerine uygulanması, ilerisini görme özelliğinden ötürü çok başarılı olmaktadır (Rawlings, 2000).

Öngörülü kontrol algoritmaları minimum-varyans kontrolörden türetilmektedir. Tasarlanan kontrolörler de, $y(t+d)$ çıkışı için sistem gecikmesi kadar eklenerek, bununla birlikte gelecekteki maliyet fonksiyonunu en küçük değeri bulunmaktadır. Performans ölçütü (PÖ), tek çıkışlı büyüklüğü içermesi yüzünden bu algoritmalar tek adımlı (single-step) veya tek adım sonrasını (one-step-ahead) öngörülü algoritmalar olarak adlandırılır.

Bu algoritmalar, sistemlerin minimum faz olduğu ve sistemin zaman gecikmelerinin tam olarak değerleri bilindiğin de kararlı kontrol sağlamaktadır. Bunun aksine durumlarda, sistemin dinamiği iyi olarak bilinmezse (sistem minimum olmayan fazsa), bir adım sonrasını öngörmek yeterli değildir (Clarke at al., 1987).

Öngörülü kontrol algoritmaları minimum-varyans kontrolörlerin tek adım sonrasını öngörmesi genişletilerek yapılmışlardır. Burada temel alınan teori, süreç çıkışının uzun menzilli (long-range) olarak öngörülmesidir. Uzun menzilin anlamı, öngörülen ufkun o andaki kontrol sinyali için süreç cevabının önemli bir kısmını kapsayacak kadar büyük olmasıdır. Öngörülü kontrol algoritmaları şöyle izah edilebilir.

MÖK'ün temel prensibi Şekil 3.1'de gösterilmiştir. En bilinen MÖK ise Genelleştirilmiş Öngörülü Kontrol "GÖK" (Generalized Predictive Controller, GPC) olarak tanımlanır (Clarke et al., 1987). GÖK, doğrusal ve ayrık zaman temelli transfer fonksiyonu modeli temellidir. Doğrusal zamanla değişmeyen süreçlerde tercih edilir. MÖK ise öngörülü kontrol algoritmalarının doğrusal ve doğrusal olmayan bütün süreç modellerine uygulanabilen genel ismi olarak tanımlanabilir (Rau and Schröder, 2002).

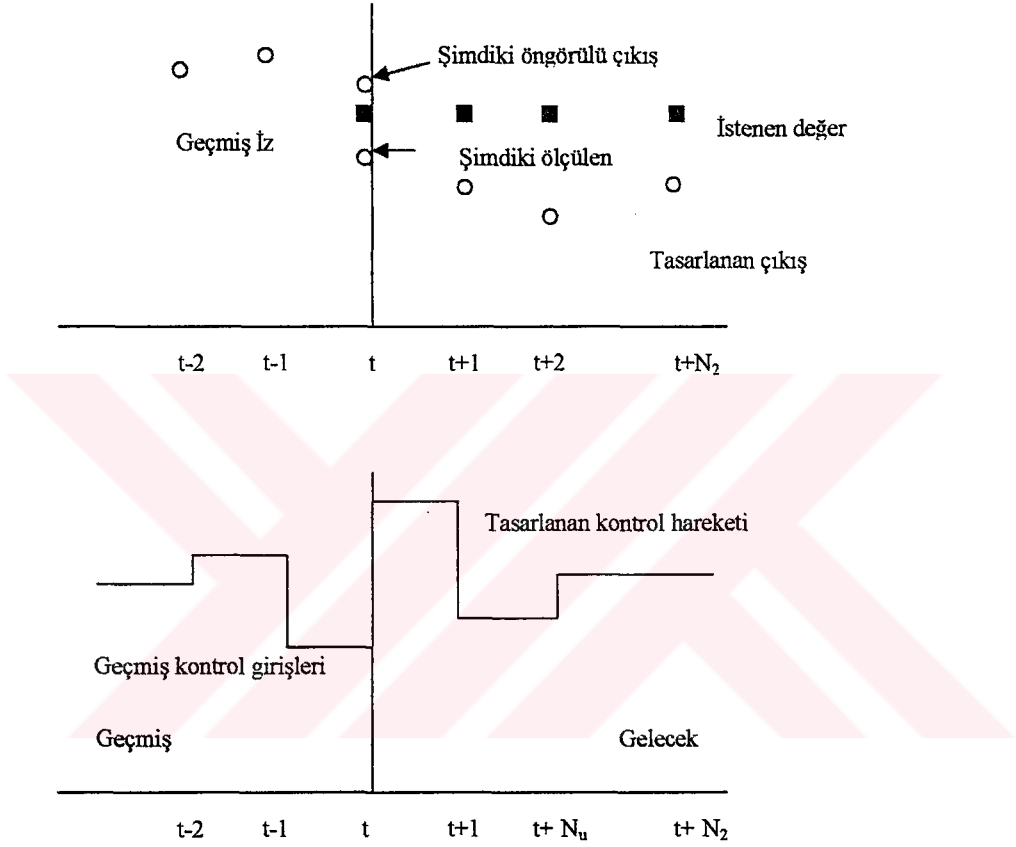
Her bir "t" anında süreç çıkışı $y(t+N_2)$ kadar öngörüle bulunulur. Burada ufuk N_2 üzerinden gösterilmektedir. Burada öngörü işlemi, süreç dinamiği için seçilen bir modele göre yapılır. O anda ki "t" değerinden başlayarak gelecekteki kontrol değerine $u(t+ N_2/t)$ bağlıdır. Burada, $N_2>0$ için $u(t+ N_2/t) \neq u(t+N_2)$ olur. Gelecekteki kontrol sinyali değeri $u(t+N_2)$ ise gelecekteki kontrol değerinin gerçek değeri olan "t" anındaki değeri bilinmemektedir.

Süreç çıkışının referans yörüngesi, $y_t^f, y_{t+1}^f, y_{t+2}^f, \dots, y_{t+N_2}^f$ olarak belirlenirse, gelecekteki kontrol vektörü $u(t+ N_2/t)$ için öngörü hataları vektör şeklinde gösterilirse $[y_t^f - y_t]$, $t=1, \dots, N_2$ 'dir. PÖ en küçük değeri hesaplanır.

MÖK özellikle son zamanlarda endüstride süreç kontrolünde kullanılmaktadır. MÖK'ün temeli sürecin modelidir. Tüm süreçlerin giriş sinyalini, başlangıç koşullarını sağladıktan sonra çıkış sinyalini öngörme işlemi yapılabilir. Sürecin modeli ile yapılan gelecekteki davranışını tahmin etmektir. Bunun için kullanılan öngörü ufku " N_2 " olarak gösterilmiştir.

Bu öngörülü çıkış değerleriyle açık çevrim performans kriterini en küçük değerine ulaştırmak (öngörülü ufukta kontrol hatalarının kareleri

toplamı) ve giriş sinyali olan “u” değerini kontrol ufkunda “ N_u ” yu hesaplamak mümkündür.



Şekil 3.1 Model öngörülü kontrol için giriş-çıkış değerleri

Kontrol ufkundan N_2 'ye kadar geçen kısımda kontrol sinyali sabit tutulur ta ki süreçten yeni değer elde edilinceye kadar. Bu işlem yeni öngörü ve kontrol ufkı ile tekrarlanarak devam eder. Bu yaklaşım gerileyen ufuk kontrol (receding horizon control) olarak adlandırılmaktadır (Primbs and Nevistic, 2000; Low and Zhuang, 2000).

Bir başka anlatımla hesaplanan optimal kontrol vektörünün sadece ilk elemanı $u(t/t)$ sisteme uygulanır. Diğer elemanları bir sonraki örnekleme anında kullanılmak için geriye kaydırılır. Bunun sonucunda bir sonraki öngörülen çıkış $y(t+1)$ elde edilir. Böylece, genel olarak önceden hesaplanan u' dan farklı yeni bir kontrol vektörü $u(t+1/t+1)$ elde edilir.

Öngörülü kontrol algoritmalarının dezavantajları da mevcuttur. Algoritmaları hesaplama yönünden karışıktır. Bu durum hesaplama gücü yüksek bilgisayarlar ve özellikle yavaş davranan süreçlerin kontrolünde bir sorun olmaz. Hangi tip algoritma kullanılırsa kullanılsın modelleme veya sistem tanımlaması gereklidir.

3.2 Yapılan Çalışmalar ve Gelişimi

Çok aşamalı (multi-stage) performans ölçütünün en küçük değer yapan “ u ” için ilk olarak Peterka ve Aström tarafından irdelenmiştir. Burada minimum varyans kavramı üzerine minimum faz olmayan sifirleri ile ilgili sorunlar çözülmüştür.

Doğrusal Kuadratik Gaussian “DKG” (Linear Quadratic Gaussian, LQG), metodun da, modelin eşdeğer durum uzayı gösterimine çevrilmesi sırasında aşırı parametre ve hesapsal yük oluşmaktadır. Bu sorunun çözümü için, LKG yaklaşımını geliştirmiş. Riccati denklemini çevrim başına bir kez kullanılacak, yeni bir metot geliştirmiştir. Clarke ise, Lam'ın yaklaşımını, genelleştirmiş ve minimum-varyans kontrolöre benzeyen özellikler ilave edilmiştir. Clarke'in algoritması, CARIMA modeli (Controlled Autoregressive Integrated Moving Average) kullanılmaktadır. Bu modelde kontrol yasasındaki integralin etkisi mevcuttur. Aynı zamanda, geliştirilen algoritma değişken sistem zaman gecikmesi ile minimum faz olmayan ve açık çevrimli kararsız sistemlere de uygundur (Clarke et al., 1987).

Uzun menzilli öngörü (long range prediction), ilk olarak Richalet ve arkadaşları tarafından IDCOM (Identification and Command) algoritması ile irdelenmiştir. Bu algoritmasında ofsetlerden sakınmak için geliştirdikleri “ad hoc” metodunu kullanılmış ve model olarak dürtü cevabı seçilmiştir.

Cutler ve Ramaker’a ait Dinamik Matris Kontrol “DMK” (Dynamic Matrix Control, DMC) metodu ise sınırlı kontrol ufku yardımı ile ortaya çıkmış ve bu kontrol ufku dışında kontrol artımlarının olmadığı düşünülmüştür. Model olarak birim basamak cevabı modeliyle incelenmiştir. Bu yeni yöntem minimum olmayan faz sistemleri ile kontrol edilebilir. Yalnız basamak cevabı modeline yüzünden uygun olmayan parametre sıkıntısından ötürü açık çevrimli kararsız sistemler kontrol edilememektedir (Maciejowski, 2002).

Richalet tarafından geliştirilen Model Algoritmik Kontrol “MAK” (Model Algorithmic Control, MAC) algoritmasında doğrusal zamanla değişmeyen süreçlerin ayrık zaman ifadelerindeki çıkışı, yine ayrık dürtü cevaplarıyla tanımlanır.

De Keyser'e göre EPSAC (Extended Prediction Self Adaptive Control) metodu ise gelecekteki kontrol değerlerine kısıtlama getirmektedir. Öngörülü çıkışların sadece $u(t/t)$ 'ye bağımlı hesabıyla yapılır.

Clarke ve arkadaşları tarafından geliştirilmiş ve öngörülü kontrol algoritmalarının bir sentezi olan, Genelleştirilmiş Öngörülü Kontrol (Generalized Predictive Control, GPC) algoritması ise gerileyen ufuk metodu olarak da adlandırılır. Referans yörüngenin önceden belirlenen durumları için kullanılır.

3.3 Temel Model Öngörülü Kontrol Kavramı

Ayrık zamanlı sistemlerin veya süreçlerin kontrol problemi, performans ölçütünün en aza indirilmesi problemi olarak kısaca tanımlanabilir. Temel model öngörülü kontrol yönteminde modelin gelecek değerlerinin istenen anda öngörülmesi, kontrol mantığımızın temelini oluşturmaktadır. Buradaki kontrol hareketlerimiz, kontrol objemizin değerlerini yakınsamalıdır. Her örnekleme zamanı için kontrol problemi yeniden çözülmeli ve bu istenen değere yaklaşma olmalıdır. Bu tip çözüme Tek-Amaçlı Model-Öngörülü Kontrol “TAMÖK” (Single-Objective Model-Predictive Control, SOMPC) denir (Brosilow and Joseph, 2002).

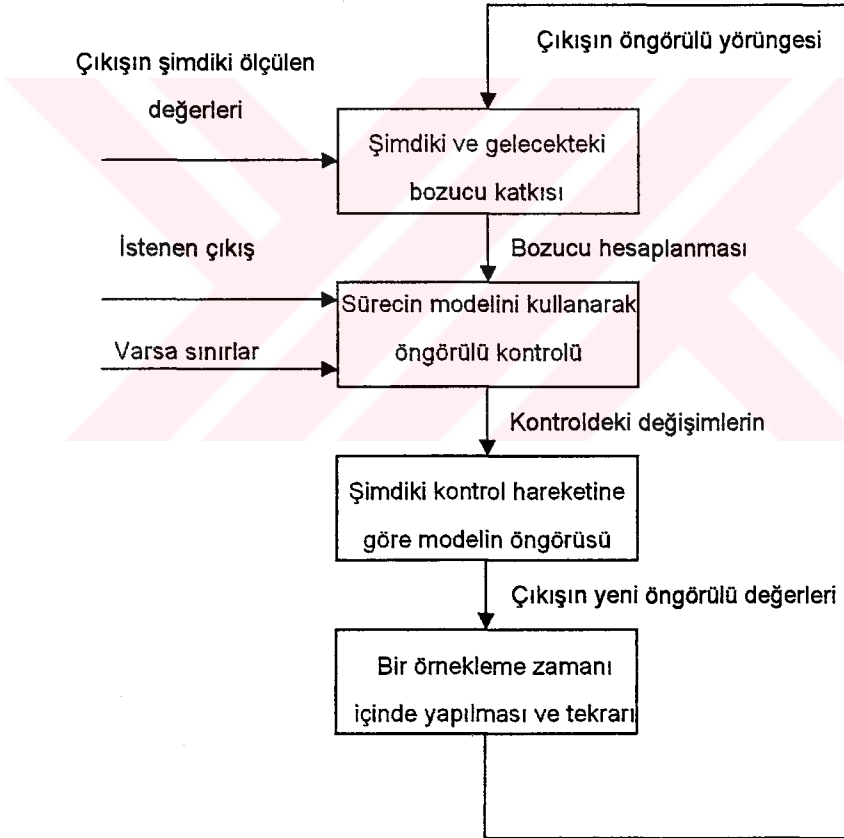
Tek-Amaçlı Model-Öngörülü Kontrol yaklaşımında en büyük avantaj tek-girişli tek-çıkışlı “TGTC” (Single-Input Single-Output, SISO) sistemlere uygun olmasıdır. Şekil 3.2’de TAMÖK’in genel algoritması gösterilmiştir. TAMÖK algoritması TGTC sistemlerin sınırları verilmeyen çözümleri içinde kullanılabilir. Bu algorithmada ilk olarak bozucuların etkisinin hesaplanması gerekmektedir. Burada elde ettiğimiz bozucu değeri ile istenen çıkış yörüngesinde süreç modelinin öngörülü çıkış kontrol miktarına eklenmektedir ve performans ölçütünün en az seviyeye indirilebilmesi için gerekli kontrol hareketinin öngörülmesi için yapılır.

Bundan sonraki adımda kontrol problemi nesnel bir fonksiyon ile açıklanabilir ve optimizasyon problemine çevrilebilir. Böylelikle kontrol objesinin karesel hatası şeklinde tanımlanır (çıkış ve istenen değer arasındaki). Bu durum sürecin optimal kontrolör ile aynı temele bağlı olduğunu da izah etmektedir. Çıkışın sonlu bir ufuk üzerinde öngörülmesi ve kontrol sinyalinin dizi halinde kuadratik bir performans ölçütünün en küçük değerinin hesaplanması, optimal kontrolör temeline

dayandığını izah eder (Clarke et al., 1987; Normey-Rico and Camacho, 2000).

$$\text{Min}_{\Delta u} \{ \|y - y_r\|^2 \} \quad (3.1)$$

Burada y çıkışın gelecekteki öngörülü değerlerinin vektörü, y_r ise çıkışın istenen değerlerinin vektörü. Kontrol hareketiyle zaman düzlemindeki problem her ayrık zaman noktasında çözülür.



Şekil 3.2 Tek-Amaçlı Model-Öngörülü Kontrol algoritması

Bu algoritmanın geliştirilmesi ile birlikte Dinamik Matris Kontrol (DMK), Cutler ve Ramaker'ın geliştirdikleri algoritması ve yine benzer yaygın bir algoritma olan Richalet'in Model Algoritmik Kontrol (MAK) algoritması kullanılmaktadır. DMK algoritması sürecin basamak cevabını temel alır. MAK ise dürtü cevabını baz almaktadır. Her iki algoritmanın avantajı kontrol hareketinin birbirleriyle koordinasyon içinde olmasıdır. Dezavantajları ise birbirlerinden farklı önceliklere sahip kontrol nesnelere sahip olmaları sebebiyle performans ölçütlerinin çıkarılması çok zor olabilmektedir.

Daha sonraki gelişmelerde ise Garcia and Morari'nin Dahili Model Kontrol "DAMK" (Internal Model Control, IMC) vardır. Bu modeli kısaca izah edersek sürecin modeli kontrolörün bir parçasıdır ve modelin çıkış değerleri ölçülen çıkış değerlerinden çıkartılır. Bu gelişmeden sonra Ricker, Garcia ve Morshedi'nin geliştirdiği DMK algoritması geliştirilerek Kuadratik Dinamik Matris Algoritması "KDMK" (Quadratic Dynamic Matrix Algorithm, QDMA) yapılmıştır.

Genellikle sürecin belli sınırlar içinde sınırlandırılması maliyet fonksiyonu açısından olumlu olarak düşünülür. Bu düşünce tarzı Model Öngörülü Kontrol "MÖK" (Model Predictive Control, MPC) algoritmasının ortaya çıkmasına vesile olmuştur. MÖK kendi arasında doğrusal program DP-MÖK veya kuadratik program KP-MÖK olarak iki tipte kullanılabilir (Brosilow and Joseph, 2002).

3.4 Tek Girişli – Tek Çıkışlı (TGTC) MÖK Algoritması

TGTC sistemlerde sınır koşulu olmadığına problem en basit hali ile gösterilir. Daha sonra çok değişkenli ve doğrusal olmayan kontrol problemleri ile devam eder.

3.4.1. Model Algoritmik Kontrol (MAK)

Doğrusal ve zamanla değişmeyen bir sistemin ayrık zamandaki $y(t)$ çıkışı için dürtü cevabı $h(j)$ yardımı ile aşağıdaki gibi gösterilebilir.

$$y(t) = \sum_{j=-\infty}^{\infty} h(j)u(t-j) \quad (3.2)$$

Burada;

$y(t)$: t anındaki sistem çıkışı,

$u(t)$: t anındaki sistem girişi,

$h(j)$: dürtü cevabının katsayılarıdır.

MAK algoritması, sistem dürtü cevabı $h(j)$ 'nin sonlu N adet terimden meydana gelen bir çözüm kullanılırsa, bir iç (internal) model kullanılarak, N_2 adım sonrasını öngören $y_{im}(t+N_2)$ öngörü değeri ile

$$y_{im}(t+N_2) = \sum_{j=1}^N h(j)u(t+N_2-j) \quad (3.3)$$

veya,

$$y_{im}(t+N_2) = h^T u(t+N_2) = u^T(t+N_2).h \quad (3.4)$$

olarak yazılabilir. Burada, $h^T = [h(1), h(2), \dots, h(N)]$ ise yeni u giriş vektörü düzenlenecek olursa,

$$u^T(t+N_2) = [u(t+N_2-1), u(t+N_2-2), \dots, u(t+N_2-N)] \quad (3.5)$$

şeklinde düzenlenir. "im" alt indisi iç modeli simgeler. Her bir "t" örnekleme anında aşağıdaki işlemler tekrarlanır.

İlk olarak bir model referans yörüngesi $y_r(t)$, öngörü ufku " t " boyunca hesaplanır. Birinci mertebeden " τ " zaman sabitli bir modelin, gerçek $y(t)$ çıkışından başlayarak, $y_r(t)$ referansına doğru değişim, Şekil 3.1'deki gibi gösterilir.

$$y_r(t+1) = \alpha y_r(t+i-1) + (i-\alpha)y_r \quad (3.6)$$

Burada $\alpha = e^{-vT_s}$ olarak yazılır. T_s ise örnekleme periyodunu, "r" model referansı gösterir.

Öngörülen ufuk boyunca gelecekteki giriş değerlerinde değişim olmadığı kabul edilir. Çıkışın öngörülü değeri hesaplanırsa,

$$\Delta u(t+i-1) = u(t+i-1) - u(t+i-2) = 0 \quad i=1,2,\dots,N \quad (3.7)$$

$$y_{im}(t) = h^T u(t) \quad (3.8)$$

$$y_{im} = \begin{bmatrix} y_{im}(t+1) \\ y_{im}(t+2) \\ \vdots \\ y_{im}(t+i) \end{bmatrix} = \begin{bmatrix} u(t) & u(t-1) & \dots & u(t+1-N) \\ u(t+1) & u(t) & \dots & u(t+2-N) \\ \vdots & \vdots & \ddots & \vdots \\ u(t+i-1) & u(t+i-2) & \dots & u(t+i-N) \end{bmatrix} \begin{bmatrix} h(1) \\ h(2) \\ \vdots \\ h(N) \end{bmatrix} \quad (3.9)$$

$i=1, 2, \dots, N$ için, $\Delta u(t+i-1)=0$ kabul edildiği için yukarıdaki eşitlik aşağıdaki şekilde tekrar formüle edilir.

$$u(t-1) = u(t) = u(t+1) = \dots = u(t+i-1) \quad (3.10)$$

Ayrıca bilinmeyen yük bozucuları ile baş edebilmek için öngörülen değerler,

$$y_{im}(t+i) = y_{im}(t+i) + (y(t) - y_{im}(t)) \quad (3.11)$$

Formül 3.11'de olduğu gibi sistemin gerçek $y(t)$ çıkışına göre düzeltilir. Model referans yörüngesi $y_r(t+i)$ ve öngörülü (adjusted prediction) $y_{im}(t+i)$ arasındaki fark alınarak bir hata vektörü oluşturulur.

$$e = \begin{bmatrix} e(t+1) \\ e(t+2) \\ \cdot \\ \cdot \\ e(t+i) \end{bmatrix} = \begin{bmatrix} y_r(t+1) \\ y_r(t+2) \\ \cdot \\ \cdot \\ y_r(t+i) \end{bmatrix} - \begin{bmatrix} y_{im}(t+1) \\ y_{im}(t+2) \\ \cdot \\ \cdot \\ y_{im}(t+i) \end{bmatrix} = y_r - y_{im} \quad (3.12)$$

Öngörü ufku boyunca giriş sinyalindeki değişimler dikkate alınır, $u(t+i-1) = u(t+i-2) + \Delta u(t+i-1)$ halini alır. Hata vektörü ise,

$$e = y_r - y_{im} - H\delta u \quad (3.13)$$

formunda yazılabilir.

Burada, $\Delta u^T = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+i-1)]$ ifadesi açık şekilde yazılırsa H matris formu aşağıdaki gibi olur.

$$H = \begin{bmatrix} h(1) & 0 & \cdot & 0 \\ h(2) & h(1) & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ h(i) & h(i-1) & \cdot & h(1) \end{bmatrix} \quad (3.14)$$

Hatayı en küçük yapacak kontrol değerlerinin optimal dizi şeklindeki hali Eşitlik 3.15'de kuadratik performans ölçütünün en küçük değeri için u kullanılır.

$$J = e^T e + \beta^2 \Delta u^T \Delta u \quad (3.15)$$

Bu eşitlikte, β artımlar cinsinden kontrol değerleri için bastırma faktörüdür. En küçük kareler yöntemi ile Δu aşağıdaki çözümlerle bulunabilir.

$$\Delta u = (H^T H + \beta^2 I)^{-1} H^T (y_r - y_{im}) \quad (3.16)$$

Sürece ilk kontrol değeri $u(t)=u(t-1)+\Delta u(t)$ uygulanır. Dürtü cevabına göre katkısı ilave edilirse, MAK algoritmasının uyarlanabilir hali elde edilir. Bu, her örnekleme anında dürtü cevabı katsayılarının çevrim-içi öngörülmesi, kontrol değeri hesabından önce yapılmalıdır.

3.4.2 Dinamik Matris Modeli (DMM)

Bir süreç incelendiğinde, örnekleme zamanı T_s dir. Çıkış $y(t)$ 'nin gelecek öngörülü değerleri $y^{N_2}(t+1), y^{N_2}(t+2), \dots, y^{N_2}(t+N_2)$ olarak gösterilir. Bu değerler geçmişteki bütün kontrol hareketlerini içerirler. Sıfır dereceden tutucu "SDT" (Zero Order Hold, ZOH) sayesinde zamanda ayırık sinyaller basamak fonksiyonu haline getirilir (Bkz. Şekil 3.1).

Birim basamak cevabı katsayılarını kullanarak y nin öngörülü değişimlerini gelecek kontrol hareketleri ile bulunabilir.

$$\Delta y_{t+1} = a_1 \Delta u_t \quad (3.17)$$

$$\Delta y_{t+2} = a_2 \Delta u_t + a_1 \Delta u_{t+1} \quad (3.18)$$

$\Delta u_t, \Delta u_{t+1}, \dots, \Delta u_{t+N_2}$ kontrol hareketleridir. Sürecin bütün cevabı yazılırsa,

$$\Delta y_{t+3} = a_3 \Delta u_t + a_2 \Delta u_{t+1} + a_1 \Delta u_{t+2} \quad (3.19)$$

...

$$\Delta y_{t+N_2} = a_{N_2} \Delta u_t + a_{N_2-1} \Delta u_{t+1} + \dots + a_{N_2-N_2} \Delta u_{t+N_2} \quad (3.20)$$

Eşitlik 3.17, 3.18, 3.19 ve 3.20'den elde edilecek matris yapısı Eşitlik 3.21'deki gibi olur.

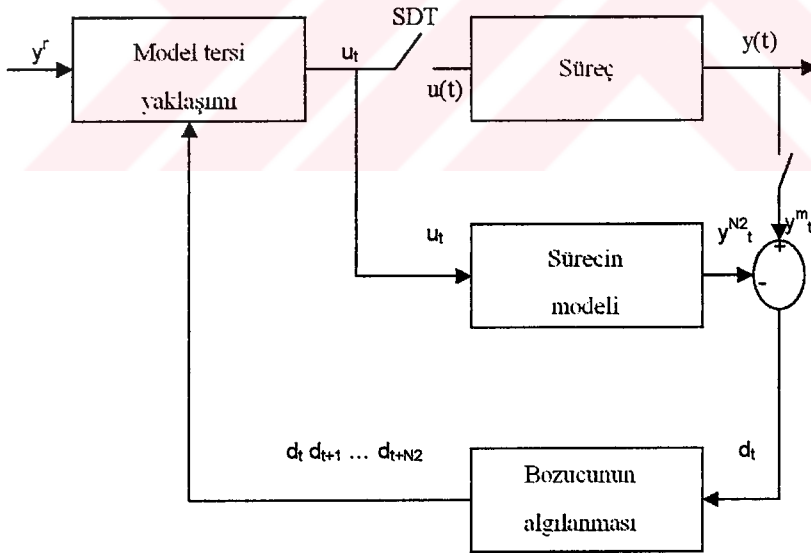
$$\begin{bmatrix} \Delta y_{t+1} \\ \Delta y_{t+2} \\ \vdots \\ \Delta y_{t+N_2} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & 0 & \dots & 0 \\ \mathbf{a}_2 & \mathbf{a}_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{N_2} & \mathbf{a}_{N_2-1} & \dots & \mathbf{a}_{N_2-N_0} \end{bmatrix} \begin{bmatrix} \Delta u_t \\ \Delta u_{t+1} \\ \vdots \\ \Delta u_{t+N_2} \end{bmatrix} \quad (3.21)$$

Sürecin dinamik matris modeli Eşitlik 3.22’de verilmiştir.

$$\Delta y = A\Delta u \quad (3.22)$$

3.4.3 Dinamik Matris Algoritması (DMA)

Dinamik algoritması açıklanabilmesi için Şekil 3.3’deki TAMÖK algoritması blok diyagramı verilmiştir (Brosilow and Joseph, 2002).



Şekil 3.3 TAMÖK algoritması blok diyagramı

Bu algoritma için verilenler;

Çıkışın öngörülü yörüngesinde $y^{N_2}(t+1), y^{N_2}(t+2), \dots, y^{N_2}(t+N_2)$ “t” zaman, “ N_2 ” öngörülü çıkış ufkunun uzunluğu

- İstenen çıkış yörüngesi $y^r(t+1), y^r(t+2), \dots, y^r(t+N_2)$
- Çıkışın o andaki ölçülen değeri y_t^m

Algoritmaya göre hesaplanacak değerler,

- Bozucunun hesaplanması $d(t), d(t+1), \dots, d(t+N_2)$
- Yeni istenen çıkışı yörüngesinin hesaplanması
- Çıkışı istenen yörüngeye götürecekt olan kontrol hareketleri

$\Delta u_t, \Delta u_{t+1}, \dots, \Delta u_{t+N_u}, N_u =$ kontrol hareketi ufkü

Açıklamalar,

- Çıkış ufkü uzunluğu N_2 , süreç modelini kullanarak çıkış değerlerinin tasarlanması için gerekli örnek sayısıdır.
- Kontrol hareketi uzunluğu N_u , örnekleme zamanları arasında kontrol girişinin ince ayarının yapılması içindir. Genel olarak kontrol hareketi daima çıkış ufkü N_2 den küçük olmalıdır ve seçilecek N_2 ufkün sonunda çıkışı durağan-duruma (steady-state) götürmelidir.
- y^{N_2} , N_2 ufkünde y nin aldığı değerleri gösterir. Algoritmanın başlangıcında çıkış değerinin o anki değeriyle başlar. Geçmiş kontrol hareketlerini barındırarak gelecek kontrol hareketlerini bulmak için güncelleştirilir.

Bozucu ifadesi Eşitlik 3.23’de gösterilmiştir.

$$d_{yt} = y_t^m - y_t^{N_2} \quad (3.23)$$

Gelecek değerlerin bütün bozucu etkisinin hesaplanması yerine basitleştirerek bu değeri çıkış ufkunun sonuna kadar aynı alabiliriz.

$$d_{y_{t+1}} = d_{y_{t+2}} = \dots = d_{y_{t+N_2}} \quad (3.24)$$

Daha sonraki adımda yeni istenen çıkış yörüngesinin hesaplanması gerekecektir. Bu formda istenen yörünge olan y^{set} yada diğer bir deyişle referans çıkışı eşit alınacaktır.

$$y_{t+1}^r = y_{t+2}^r = \dots = y_{t+N_2}^r = y^{set} \quad (3.25)$$

Son aşama olarak da kontrol hareketinin hesaplanması gerekmektedir. Burada Dinamik matris modeli kullanılacaktır.

$$\begin{aligned} y_{t+1} &= y_{t+1}^{N_2} + a_1 \Delta u_t && + d_{y_{t+1}} \\ y_{t+2} &= y_{t+2}^{N_2} + a_2 \Delta u_t + a_1 \Delta u_{t+1} && + d_{y_{t+2}} \\ y_{t+3} &= y_{t+3}^{N_2} + a_3 \Delta u_t + a_2 \Delta u_{t+1} + a_1 \Delta u_{t+2} && + d_{y_{t+3}} \\ &\dots && \\ y_{t+N_2} &= y_{t+N_2}^{N_2} + a_{N_2} \Delta u_t + a_{N_2-1} \Delta u_{t+1} + \dots + a_{N_2-N_u} \Delta u_{t+N_u} + d_{y_{t+N_2}} \end{aligned} \quad (3.26)$$

Eşitlik 3.26'da a_1, a_2, \dots, a_{N_2} birim basamak cevabı katsayılarıdır ve

$\Delta u_t, \Delta u_{t+1}, \dots, \Delta u_{t+N_u}$ her adımdaki kontrolör girişindeki değişimlerdir.

Matris formuna sokacak olursak bu değerleri toplu halde görmemiz mümkün olacaktır. Eşitlik 3.27 ve 3.28'de görüldüğü gibidir.

y^{N_2} ve d_y bilindiğine göre istenen Δu ile y nin değerini y^r 'ye yaklaştırmak mümkündür.

$$\begin{bmatrix} y_{t+1} \\ y_{t+2} \\ \cdot \\ \cdot \\ y_{t+N_2} \end{bmatrix} = \begin{bmatrix} y_{t+1}^{N_2} \\ y_{t+2}^{N_2} \\ \cdot \\ \cdot \\ y_{t+N_2}^{N_2} \end{bmatrix} + \begin{bmatrix} a_1 & 0 & \cdot & 0 \\ a_2 & a_1 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{N_2} & a_{N_2-1} & \cdot & a_{N_2-N_u} \end{bmatrix} \begin{bmatrix} \Delta u_t \\ \Delta u_{t+1} \\ \cdot \\ \cdot \\ \Delta u_{t+N_u} \end{bmatrix} + \begin{bmatrix} d_{y_{t+1}} \\ d_{y_{t+2}} \\ \cdot \\ \cdot \\ d_{y_{t+N_2}} \end{bmatrix} \quad (3.27)$$

$$y = y^{N_2} + A\Delta u + d_y \quad (3.28)$$

Bu durumda optimizasyon problemini tanımlamış oluruz. Burada amaç fonksiyon ile açıklanabilir (objective function).

$$\text{Min}_{\Delta u} \{ \|e\|^2 \} \quad (3.29)$$

Eşitlik 3.29'da,

$$\begin{aligned} e &= y - y^r \\ &= y^{N_2} + A\Delta u + d_y - y^r \end{aligned} \quad (3.30)$$

şeklinde düzenlenir. Bu problem en küçük kareler (least-square) çözüm yöntemiyle çözülebilir.

$$A\Delta u = y^r - d_y - y^{N_2} \quad (3.31)$$

$N_2 \gg N_u$ olması sebebiyle ve N_u+1 ($\Delta u_t, \Delta u_{t+1}, \dots, \Delta u_{t+N_u}$)

bilinmeyen de N_2 kadar denklem vardır. Bu da bilinmeyen denklem sayısını artırmaktadır. En küçük kareler yöntemi ile yapılan çözümde Eşitlik 3.32 deki gibi olacaktır.

$$\begin{aligned}\Delta u &= (A^T A)^{-1} A^T (y^r - d_y - y^{N_2}) \\ &= B(y^r - d_y - y^{N_2})\end{aligned}\quad (3.32)$$

Burada, A bilinmeyen matristir ve hesaplanacaktır. B ise süreç çalışırken saklanacak matristir. İlerleyen zaman da Δu_t 'ye ihtiyacımız vardır. Bunun içinde B matrisinin ilk satırını almak yeterli olacaktır.

$b^T = B$ matrisinin ilk satırındır.

Yeni Δu_t Eşitlik 3.33'de gösterilmiştir.

$$\Delta u_t = b_1^T (y^r - d_y - y) \quad (3.33)$$

Bir sonraki örnekleme zamanına geçildiğinde güncellenmesi gereken y nin öngörülü değeri hesaplanacaktır.

$$\begin{aligned}y_{t+1}^{N_2} &= y_{t+1}^{N_2} + a_1 \Delta u_t + d_{yt+1} \\ y_{t+2}^{N_2} &= y_{t+2}^{N_2} + a_2 \Delta u_t + d_{yt+2} \\ &\vdots \\ y_{t+N_2}^{N_2} &= y_{t+N_2}^{N_2} + a_{N_2} \Delta u_t + d_{yt+N_2}\end{aligned}\quad (3.34)$$

Eşitlik 3.34'de görülmektedir ki o anki kontrol hareketi ve bozucunun etkisi eklenmiştir.

Dinamik matris algoritması için yapılması gereken parametre ayarları,

Örnekleme zamanı T_s ve öngörülü ufku N_2 , y nin ufuk zamanını verir. $N_2 T_s$ durağan duruma ulaşmak için gereken zamandır. Birinci dereceden bir sistem 4τ zamanda istenen değerine ulaştığına göre

$$N_2 T_s = 4\tau$$

bulunur. Burada τ sürecin baskın zaman sabitidir. Çıkış y 'nin örnekleme sayısı kadar A matrisinin büyüklüğü hesaplanabilir. Büyük matris boyutları hesaplanma açısından uygun değildir ve tercih edilmez. T_s değerinin de çok büyük olma ihtimali N_2 'yi düşürür ve uygun cevap elde edilemez. Bunun anlamı uygun bir N_2 seçilmesidir ve bu değer yaklaşık olarak $N_2 = 20 - 30$ arasında seçilmelidir.

Kontrol ufku N_u olduğuna göre, kontrol ufkunun artması y 'nin alacağı değerlerin istenen değere daha çabuk ulaşmasını sağlar. Burada kontrolörün istenen değere keskin bir şekilde yaklaşması istenmez.. Kontrol sinyali olan u 'nun hareketine bir sınırlama getirilmemesi ufuk boyunca $y = y^r$ değerlerinin bilinmesi anlamına gelir ve pratikte istenmez. Bu istenmeyen durumda, performansa katkısı sebebiyle N_u için uygulamada küçük değer seçilir. $A^T A$, $N_u \times N_u$ matris olmalıdır. Bu da yaklaşık olarak $N_u = 1 - 4$ arasındadır.

Sistem kararlılığı için $N_2 \gg N_u$ olmalıdır.

Bastırma faktörü β , azalan kareler çözüm yönteminde karşımıza çıkacak olan Eşitlik 3.29 ve 3.31'den

$$\text{Min}_{\Delta u} \left\| y^r - d_y - y^{N_2} - A\Delta u \right\|^2 \quad (3.35)$$

hatayı en küçük değer yaparken Δu değişimi de hesaba katılırsa.

$$\text{Min}_{\Delta u} \left\| y^r - d_y - y^{N_2} - A\Delta u \right\|^2 + \beta^2 \|\Delta u\|^2 \quad (3.36)$$

Eşitlik 3.36'daki gibi olacaktır. Burada β bastırma faktörüdür. Çözümde Δu 'nun etkisiyle azalır. Denklemlerin yeni değerlerine eklenir.

$$\begin{aligned}
\beta \Delta u_t &= 0 \\
\beta \Delta u_{t+1} &= 0 \\
&\vdots \\
\beta \Delta u_{t+N_a} &= 0
\end{aligned} \tag{3.37}$$

Eşitlik 3.32' nin yeni çözüm denklemi Eşitlik 3.38'deki gibi olur.

$$\Delta u = (A^T A + D)^{-1} A^T (y^r - d_y - y^{N_2}) \tag{3.38}$$

D matrisi $n \times n$ boyutundadır.

$$D = \text{diag} [\beta^2 \ \beta^2 \ \beta^2 \ \dots \ \beta^2] \tag{3.39}$$

Eşitlik 3.39'da, D matrisinin avantajı sadece bir parametre eklenerek, Δu 'nun Eşitlik 3.38'de görüldüğü üzere tekil olma sorunu çözülür. β genelde çevrimi-içi çalışma parametresi (online parameter) olarak kullanılır.

3.4.4 Genelleştirilmiş Öngörülü Kontrol (GÖK) Yöntemi

Genelleştirilmiş öngörülü kontrol "GÖK" (Generalized Predictive Control, GPC), süreçlerin kontrol işlemlerini parametreleriyle kararlı halde yapabilme özelliğine sahiptir. Burada ilave olarak, ölü zaman ve giriş – çıkış arasındaki rasgele değişimlerin etkisi de etkin olarak kontrol altına alınabilmektedir.

Genelleştirilmiş öngörülü kontrol algoritması CARIMA formunda bir süreç modeli kullanır (Clarke et al., 1987). Bu model, belli sayıdaki parametrelerden oluşan az sönümlenmiş veya kararsız sistemler için doğru çıkış öngörülleri elde etmesinden ötürü kullanılır. GÖK, kararlı bir sistem için diğer yöntemlerdeki kontrol kalitesini sağladığı gibi daha da karmaşık dinamik süreçleri kontrol edebilecek şekilde genişletilmiştir.

GÖK, uygulama da $y(t)$ çıkışının sonlu bir ufuk üzerinden öngörüldüğü ve kontrol sinyallerini dizi halinde kuadratik bir performans ölçütünün en küçük değeri için hesaplayan bir algoritmadır. İstenilen cevabın elde edilebilmesi için kontrol sinyali değerinin beklenen değerlerinde bazı öngörüler yapılmaktadır: Bu öngörüler GÖK yaklaşımının püf noktasını teşkil eder.

GÖK algoritmasının avantajlarından biri, parametre, öngörü ufku ve kontrol ufkunun uygun şekilde seçilmeleri sayesinde de diğer model öngörülü kontrol algoritmalarını ve kendini ayarlayabilen kontrol algoritmalarını türetmek mümkündür (Maciejowski, 2002).

GÖK algoritmasında hesaplanacak parametre sayısı diğer metotlar olan MAK ve DMA'ye göre daha azdır. Böylelikle hesapsal yükü azalır. Dolayısıyla da daha hızlıdır. Yapısında integratör bulunması nedeniyle kararsız ve titreşimli sistemlerde iyi performans göstermektedir.

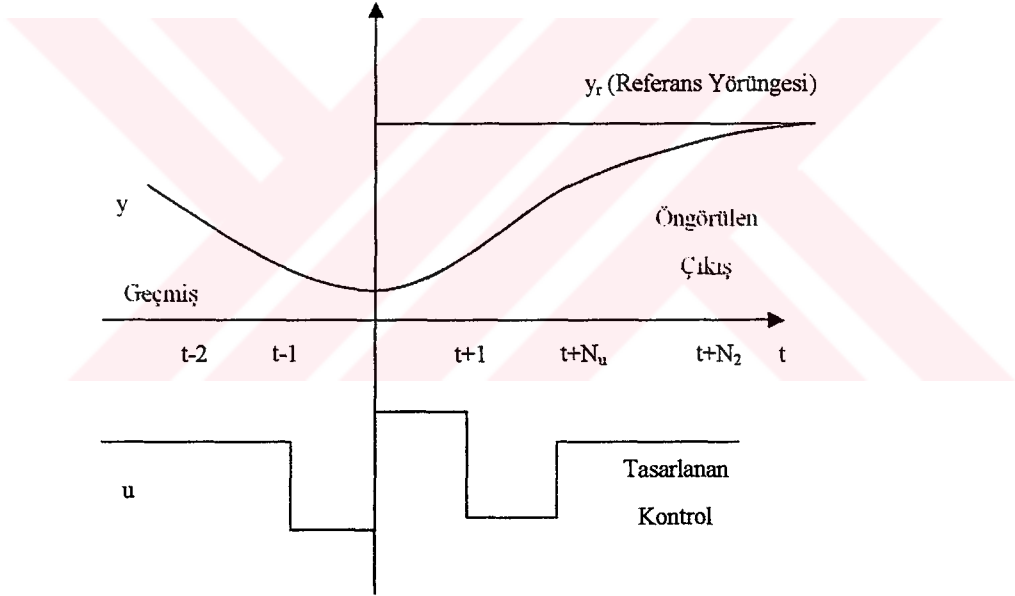
Bu üstünlüklerinden dolayı yapay sinir ağlarının "YSA" (Artificial Neural Network, ANN) eğitiminde kullanılacak olan eğitim ve test dataları için kullanılabilir (Song and Koivo, 1999; Nguyen and Widrow, 1990).

Genelleştirilmiş öngörülü kontrol GÖK, sadece bir öngörü yerine sonlu öngörü ufku boyunca oluşturulan, daha genel bir maliyet fonksiyonuna sahip olması ile kendinden önce yapılan bir adım sonrasını öngörebilen algoritmalarından farklıdır.

Öngörülü bir kontrol algoritması esas olarak, maliyet fonksiyonu en küçük olacak, bir kontrol dizisini seçme algoritmasından meydana gelmekte ve uyarlanabilir (adaptive) kontrol uygulamaların da önemli bir yer tutmaktadır.

3.5 Öngörülü Kontrol Kanunu ve PÖ Tanımı

Öngörülü kontrolün temel tanımlamaları için süreç modelin doğrusal, maliyet fonksiyonunun da kuadratik ve varsa sınırların doğrusal eşitsizlikler olduğu kabul edilir (Maciejowski, 2002). Öngörülü kontrol kanunun amacı, gelecek zaman dilimlerindeki $y(t+i)$ sistem çıkışlarının referans değeri $y_r(t+i)$ 'yi mümkün olduğunca yakın olarak takip etmesini sağlamaktır. Bu konuyu daha iyi izah edebilmek için, Şekil 3.4'deki genelleştirilmiş öngörülü kontrol değişkenleri ve gerileyen ufuk yaklaşımı kullanılmaktadır (Clarke et al., 1987).



Şekil 3.4 Referans, kontrol ve çıkış yörüngeleri

Şekil 3.4'de "t" anında $y(t)$ çıkışı hedeflenen referans değerine Eşitlik 3.40'daki formda ulaşabilmektedir.

$$y_r(t+i) = \alpha y_r(t+i-1) + (1-\alpha)y_r \quad i=1, 2, \dots \quad (3.40)$$

Burada “ α ” bir ayar parametresi olup, 0 ila 1 arasında bir değerdir. Şayet $\alpha \rightarrow 0$ ise hızlı kontrol, aksine $\alpha \rightarrow 1$ yaklaşırsa da yavaş kontrol sağlamaktadır. GÖK öngörülü algoritmalar içinde değişen referans yörüngesine uyum göstermesi sebebiyle tercih edilir.

Gerileyen ufuk yaklaşımı ile şu şekilde açıklanabilir. Her bir örnekleme anında, gelecekteki referans değerleri hesaplanır. Bir başka anlatımla, gelecekteki sistem çıkışları $y(t+i)$ öngörülür. Burada $y(t+i)$ 'nin kısmen hesaplanması gereken gelecekteki kontrol büyüklükleri $u(t+i)$ 'ye bağlı olacaktır. Bir sonraki adımda, belirli bir kontrol ufkundan sonra kontrol büyüklükleri artımı sıfır kabul edilerek, kuadratik maliyet fonksiyonu en küçük değerini bulmak için çalışılır. Hesaplanan kontrol değeri $u(t)$ 'nin ilk elemanı olarak sisteme uygulanır ve gelecek örnekleme anında tekrarlanmak üzere veri vektörleri bir geriye kaydırılır bu işleme gerileyen ufuk yaklaşımı (GUY) adı verilir (Kothare and Morari, 2000).

Performans ölçütü “ J ”, öngörülü kontrol çıkışlarındaki $y(t+i)$ sapmalarını, referans yörüngesi olan $y_r(t+i)$ 'ye götürecektir. Parametreler için kullanılan işaretlemeler gösterir ki, referans yörüngesi “ t ” zamanına kadar olan ölçümlere bağlı olabilir. Özellikle başlangıç noktası çıkış ölçümü olan $y(t)$ olabilir. Fakat aynı zamanda sabit bir nokta veya bazı diğer önceden belirlenmiş yörünge olabilir. Maliyet fonksiyonu Eşitlik 3.40'daki gibi tanımlanır.

$$J(t) = \sum_{i=N_1}^{N_2} [y(t+i) - y_r(t+i)]^2 + \sum_{i=1}^{N_u} \lambda(i) [\Delta u(t+i-1)]^2 \quad (3.41)$$

N_1 : En küçük maliyet ufku

N_2 : En büyük maliyet ufku (Öngörü ufku)

N_u : Kontrol ufku

$\lambda(i)$: Kontrol ağırlıkları dizisi

Eşitlik 3.41’de not edilecek çeşitli noktalar vardır. Öngörü ufku N_2 uzunluğuna sahiptir fakat y ’nin sapmalarının y_r ’ye hemen ulaşmasına ihtiyaç duyulmaz. Bu arada $N_1 > 1$ olmalıdır. Çünkü girişin uygulanmasıyla, herhangi bir etkinin görülmesi arasında biraz zaman gecikmesi olabilir. N_u kontrol ufkudur ve her zaman $N_u \leq N_2$ kabul edilir, bunun sebebi kontrol hareketinin değişimi öngörü ufkundan önce bitmelidir ki o anki kontrol sinyali için öngörülü çıkış alınabilsin.

Maliyet fonksiyonunun denkleminde, hata vektörü $(y(t+i) - y_r(t+i))$ ’nin öngörü ufku içinde $N_1 \leq i \leq N_2$ sınırlarında her noktada referansa ulaşabileceğini gösterir. Bu gerçekten öngörülü kontrol içinde oldukça önemli bir durumdur.

$$\sum_{i=1}^{N_u} \lambda(i) [\Delta u(t+i-1)]^2 \quad (3.42)$$

Maliyet fonksiyonu sadece giriş vektöründe içindeki değişiklikleri sınırlandırır. Bunun aksine değerlerini sınırlandırmaz.

Maliyet fonksiyonun ikinci kısımdaki ifadesi, kontrol sinyalinin değişimlerini içerir. Eşitlik 3.42’deki denklemde kontrol ufku boyunca yapılacak kontrol aksiyonları arasındaki fark hesaplanacaktır. Bu işlemde kontrol ufkundan sonra değişim olmayacağı için kontrol ufkundan sonraki değişim sıfır olacaktır. Burada, kontrol ağırlığı $\lambda(i)$, kontrol aksiyonunun PÖ’ye olan katkısını belirler.

Öngörü ve kontrol ufukları N_2 ve N_u , sınırlama parametresi N_1 , kontrol ağırlığı $\lambda(i)$ ve referans yörüngesi $y_r(t+i)$ öngörülü kontroller ve sürecin kapalı çevrim davranışını etkilerler. Bu parametrelerden bazıları, kontrol sistemlerinin maliyet fonksiyonları tarafından direkt etkilenirler. Genelde de bunlar, dinamik performans doyumu vermek için ayarlanan ince ayar parametreleridir (Maciejowski, 2002).

Eşitlik 3.41'deki maliyet fonksiyonunda N_2 öngörü ufku genellikle değer olarak, süreç cevabının tamamını içerecek şekilde seçilir. Başka bir deyişle, tipik olarak sistemin yükselme zamanına (rise-time) yakın olarak seçilir. N_1 genellikle "1" olarak seçilmektedir. Eğer sistemin ölü zamanı "d" biliniyorsa hesaplamaları küçültmek için d'ye eşit veya daha büyük " λ " ise $0 < \lambda < 1$ arasında seçilebilir. Sistem modellerinin büyük bir kısmında $N_1=1$ ve $N_2=10$ değerleri ile karalı hale getirilebileceği gösterilmiştir.

3.5.1 GÖK'de Kontrol Sinyali Tanımlanması

Kontrol sinyalinin hesaplanabilmesi için öncelikli olarak kolaylık olabilmesi için $N_1 = 1$, $N_2 = N$, $N_u = N$ ve $\lambda(i) = \lambda$ olarak gösterilecektir. Eşitlik 3.43'deki CARIMA modeli ifadesinden faydalanılarak gelecekteki öngörülü çıkış ifadesi Eşitlik 3.44'deki gibi düzenlenebilir.

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + x(t) \quad (3.43)$$

$$y(t+i) = E_i B \Delta u(t+i-1) + F_i y(t) + E_i \xi(t+i) \quad (3.44)$$

Eşitlik 3.44'deki ifadede $G_i(q^{-1}) = E_i B$ tanımlanır. Burada E_i gelecekteki etkiyen tüm bozucu etkileri simgeler.

Bu hatırlatıcı bilgilerin ışığında çıkış ifadeleri tekrar yazılacak olursa Eşitlik 3.45'deki denklem takımı elde edilir. Bu takım üç terimden oluşmaktadır.

Bunlardan birincisi, gelecekteki kontrol aksiyonu. İkincisi, geçmişteki kontrol aksiyonu ile ölçülen değişkenler. Üçüncüsü ise, gelecekteki gürültü sinyalidir. Burada gürültü dizisi $\xi(t+i)$ öngörü hatası olarak kabul edilmektedir.

$$\begin{aligned}
y(t+1) &= G_1 \Delta u(t) + F_1 y(t) + E_1 \xi(t+1) \\
y(t+2) &= G_1 \Delta u(t+1) + F_2 y(t) + E_2 \xi(t+2) \\
&\dots \\
&\dots \\
y(t+N) &= G_N \Delta u(t+N-1) + F_N y(t) + E_N \xi(t+N)
\end{aligned} \tag{3.45}$$

$y(t+i)$ 'nin t anındaki bilinen değerlerden oluşan bileşenleri $f(t+i)$ olarak gösterilirse (Eşitlik 3.46).

$$\begin{aligned}
f(t+1) &= [G_1(q^{-1}) - g_{10}] \Delta u(t) + F_1 y(t) \\
f(t+2) &= q[G_2(q^{-1}) - q^{-1} - g_{21} - g_{20}] \Delta u(t) + F_2 y(t) \\
&\dots \\
&\dots
\end{aligned} \tag{3.46}$$

Eşitlik 3.46'da $G_i(q^{-1}) = g_{i0} + g_{i1}q^{-1} + \dots$ şeklinde yazılır. $F(t+i)$ ifadesi sistemin serbest çıkış cevabına karşılık gelmektedir. Genel formda ifade edilecek olursa Eşitlik 3.47'deki gibi.

$$\hat{y} = G\hat{u} + f \tag{3.47}$$

Eşitlik 3.47'deki değişkenlerin açık hali şöyledir;

$$\begin{aligned}
\hat{y} &= [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N)]^T \\
\hat{u} &= [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N-1)]^T \\
f &= [f(t+1), f(t+2), \dots, f(t+N)]^T
\end{aligned} \tag{3.48}$$

Eşitlik 3.48'deki vektörler $N \times 1$, G matrisi ise $N \times N$ boyutundadır. $G_i(q^{-1})$ 'deki ilk i terim basamak cevabı katsayılarıdır. $i = 0, 1, 2, \dots < i$ için $g_{ji} = g_i$ dir. Yani $g_{10} = g_0$, $g_{21} = g_1$ dir. G matrisi açık ifadesi (Eşitlik 3.49).

Eğer sistemde ölü zaman var ise ve $d > 1$ ise G matrisinin ilk satırı $d-1$ satırı sıfır alınacaktır.

$$G = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \cdot & \cdot & & \\ g_{N-1} & g_{N-2} & \dots & g_0 \end{bmatrix} \quad (3.49)$$

Burada N_1 , “d” değerine eşit alınarak ilk satırın sıfır alınması önlenmiş olur.

Verilen referans yörüngesine göre PÖ tekrar yazılacak olursa Eşitlik 3.51’deki gibi olur.

$$y_r = [y_r(t+1), y_r(t+2), \dots, y_r(t+N)]^T \quad (3.50)$$

$$J = \{(G\hat{u} + f - y_r)^T (G\hat{u} + f - y_r) + \lambda \hat{u}^T \hat{u}\} \quad (3.51)$$

Burada Eşitlik 5.14’deki ifadeyi en küçük değerine götürecek olan kontrol vektörü PÖ’nün türevinin sıfıra eşitlenmesi ile bulunur (Eşitlik 3.52).

$$\hat{u} = (G^T G + \lambda I)^{-1} G^T (y_r - f) \quad (3.52)$$

Eşitlik 5.2’den, \hat{u} ’deki ilk elemanı olan $\Delta u(t)$ ’den $u(t)$ ’nin değeri bulunabilir.

$$u(t) = u(t-1) + (G^T G + \lambda I)^{-1} G^T (y_r - f) \quad (3.53)$$

3.5.2 Kontrol Ufkunun Etkisi

Kontrol sinyali ifadesindeki (Eşitlik 3.53) G matrisi boyutu $N \times N$ olarak bulunmuştu. Kontrol aksiyonunun hesaplanmasında, her bir örnekleme anında $(G^T G + \lambda I)$ ’nin ters alma işleminin yapılması hesapsal açıdan büyük bir yük getirecektir. Buna ek olarak ölü zaman için hatalı değer kabul edilirse, $(G^T G)$ tekil olacaktır. Tekilliği ortadan kaldıracak olan λ ’nın ağırlığı önemlidir ve gerçek değerini önceden bilmek mümkün olmadığı için bu değerini seçimi kullanıcıya problem teşkil edebilir.

Hesaplama yükünü azaltmak için gelecekteki kontrol büyüklükleri ile ilgili olarak DMK metodunda da kullanılan bir kabul yapılmaktadır. Bu kabule göre, $N_u < N_2$ olacak şekilde seçilen N_u kontrol ufkundan sonra kontrol artımlarının olmadığı varsayılmıştır.

Burada N_u "Kontrol ufkı" olarak adlandırılır. N_u kontrol ufkunun gösterildiği maliyet fonksiyonu, yani PÖ'deki Eşitlik 3.41'de verilmiştir.

Bu maliyet fonksiyonunda $N_u=1$ ise sadece tek bir kontrol değişimi $\Delta u(t)$ göz önüne alınarak, sonraki $u(t+1)$ kontrol büyüklüklerinin hepsi $u(t)$ 'ye eşit alınacaktır. $N_u=1$ olduğunda kontrol aksiyonu genelde düzgün ve ağır hareketler sağlar. N_u 'nun değeri arttıkça daha keskin kontrol aksiyonu elde edilir.

$N_u < N_2$ seçimi, matris hesabından kaynaklanan yükü azaltır. Kontrol sinyalinin vektör boyutu $N_u \times 1$ olur.

$$G_1 = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \cdot & \cdot & & \\ g_{N_2-1} & g_{N_2-2} & \dots & g_{N_2-N_u} \end{bmatrix}_{N_2 \times N_u} \quad (3.54)$$

Bu bilgilerin ışığı altında kontrol sinyalinin değeri Eşitlik 3.55'deki gibi olacaktır.

$$\hat{u} = (G_1^T G_1 + \lambda I)^{-1} G_1^T (y_r - f) \quad (3.55)$$

Bu düzenleme sayesinde tersi alınacak matrisin boyutu $N_u \times N_u$ olarak indirgenmiştir.

3.5.3 Çıkış ve Kontrol Ufuklarının Belirlenmesi

En küçük öngörü ufkı - N_1 : Eğer ölü zaman "d" biliniyorsa, gerek olmayan hesabı önlemek için N_1 ufkı d'den küçük seçilir. Aksine d'nin

değeri bilinmiyorsa veya zaman içerisinde değişiyorsa, $N_1=1$ alınır. Bu durumda, Eşitlik 3.43'deki $B(q^{-1})$ 'in mertebesi d 'nin tüm değerlerini kapsayacak şekilde artırılmış olur.

En büyük öngörü ufku - N_2 : Genellikle $B(q^{-1})$ 'in derecesinden büyük seçilir. Daha yalın halde söylenirse, tipik olarak sistemin yükselme zamanına (rise-time) denk alınmalıdır.

Kontrol ufku - N_u : Öngörülü kontrolde önemli bir tasarım parametresidir. Basit bir sistem için $N_u = 1$ değeri genellikle kabul edilir kontrolü sağlar. Kompleks sistemler için N_u 'nun değerini büyük almak uygun olur. Bu tip sistemler için N_u için en küçük değer kararsız veya kötü sönümlenmiş kutupların sayısına eşit alınır. $N_u = N_2$ seçildiğinde GÖK kontrolör minimum-varyans bir kontrolöre dönüşür. Bu yüzden N_u 'nun değeri için dikkatli seçim gerekir.

Sonuç olarak, GÖK iki ayrı şekilde kullanılabilir. $N_1=1$ ve N_2 sürecin yükselme zamanına eşit, aynı zamanda $N_u=1$ ise sürecin kontrolü için uygun performans demektir. Yüksek performanslı uygulamalar için ise N_u 'nun değerini arttırmak daha uygun olur.

3.5.4 GÖK'ü Oluşturan Yapı Taşları

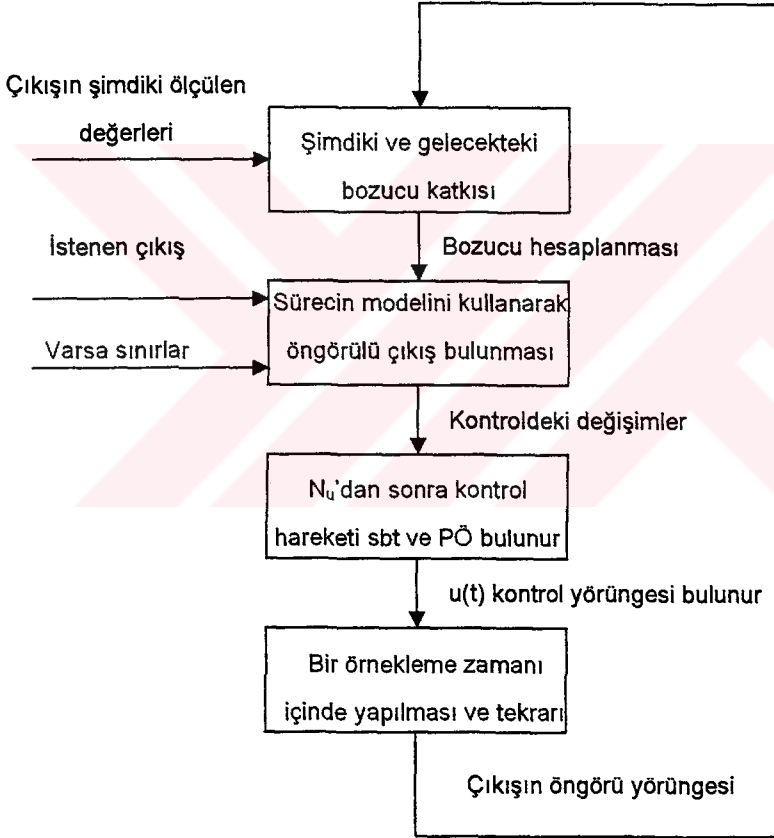
GÖK'ü oluşturan beş temel yapı taşı vardır. Bunlar aşağıdaki gibidir;

- CARMA sistem modeli yerine, CARIMA modeli kullanılması.
- Sistemin ölü zamanından büyük ve en küçük değer olarak model mertebesine eşit ufuk üzerinden öngörünün yapılması.
- Diophantine denkleminin ardışık olarak kullanılmasıdır (Bu denklem takımı E_{i+1} , F_{i+1} değerleri E_i , F_i değerlerinden elde edilmesidir).
- PÖ'de kontrol aksiyonlarına bir ağırlığın uygulanması.

- N_u 'dan sonra kontrol artımlarının sıfır olarak alındığı bir kontrol ufku seçilmesi.

3.5.5 GÖK Algoritması

Genelleştirilmiş öngörülü kontrol algoritmasında her bir örnekleme zamanındaki işlem algoritması.



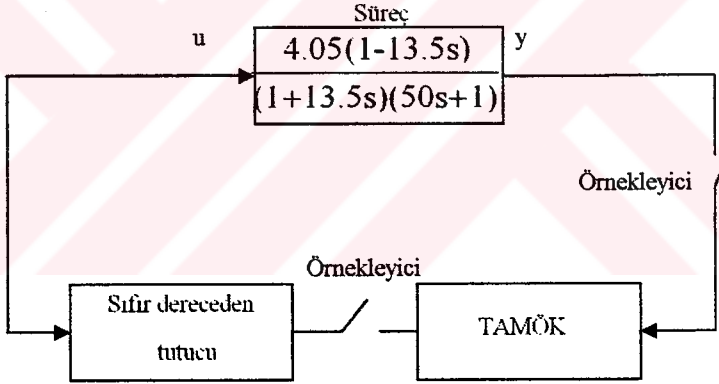
Şekil 3.5 Genelleştirilmiş Öngörülü Kontrol algoritması

3.6 TGTC Model Öngörülü Kontrol Uygulaması

Örnek 3.1: Tek girişli – tek çıkışlı bir sürecin model öngörülü kontrol algoritması ile çözümlenmesi için kullanılacak olan DMA algoritması ile çözüme başlanır. İlk olarak, transfer fonksiyonu verilen bir sürecin denklemi Eşitlik 3.56'daki gibi olsun.

$$y(s) = 4.05(1 + ke) \frac{(1 - 13.5s)}{(1 + 13.5s)(50s + 1)} u(s) \quad (3.56)$$

Bu eşitlikte verilen ke değeri için sırasıyla (-0.5, 0, 0.5), ve verilen diğer parametreler; bastırma faktörü $\beta=1$, kontrol ufku için de $N_u=1, 2, 3$ ise SOMPC algoritmasından faydalanarak DMA yöntemi ile çözümü yapılacaktır (Brosilow and Joseph, 2002).



Şekil 3.6 Doğrusal sistem örnek 3.1 için blok diyagram

Burada simülasyon için gerekli parametrelerin tanımı sırasıyla;

$N_2 \rightarrow$ öngörü ufku uzunluğu,

$t_s \rightarrow$ örnekleme zamanı,

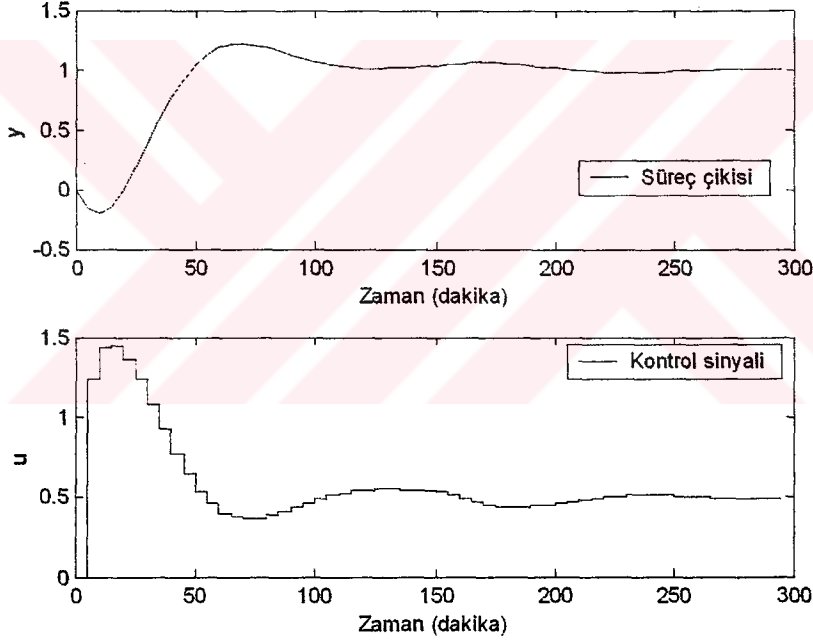
N_u → kontrol hareketi sayısı veya kontrol ufku,

β → bastırma katsayısı,

ke → model için bozucu etki,

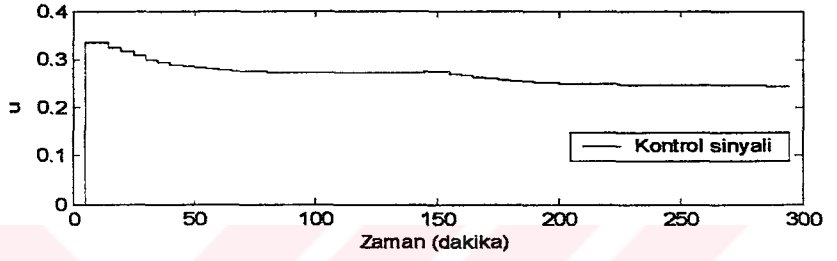
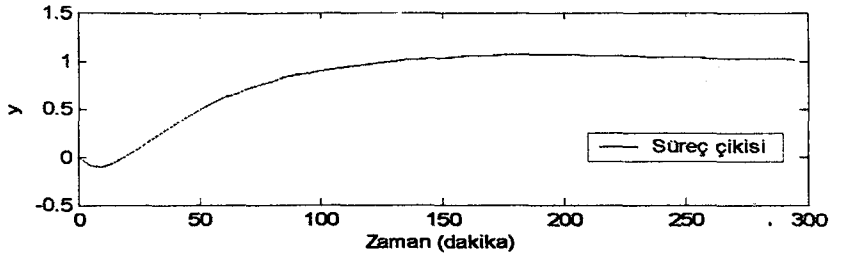
nt → zamanda simülasyon için 300 dakika verilmiştir.

Algoritmada kullanılacak nümerik değerler $N_2=30$, $ts=5$, $N_u=1$, $\beta=1$ ve $nt=60$ olarak verilmiştir. Buna göre model bozucu etkisi istene değere göre -0.5 alınmış ve simülasyon sonucu Şekil 3.7’de gösterilmiştir.

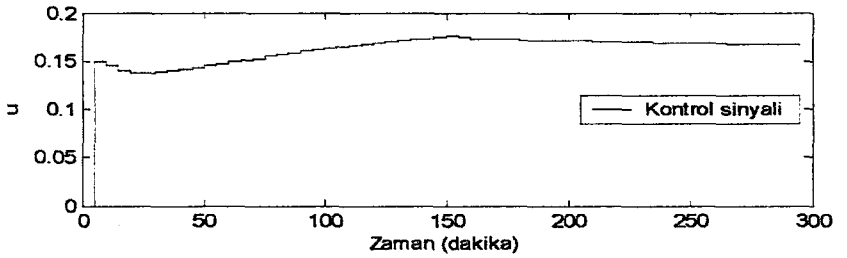
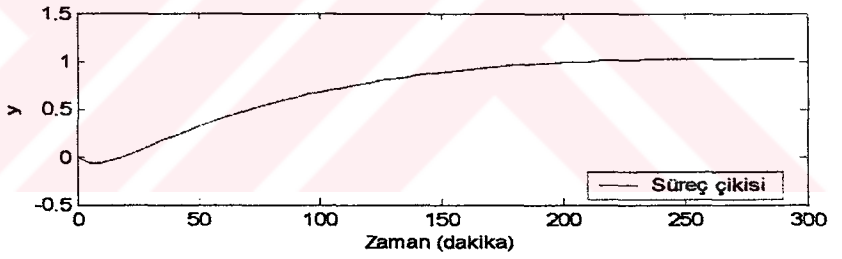


Şekil 3.7 Örnek 3.1 için $N_u=1$. $ke=-0.5$

Bu adımdan sonra kullanılacak nümerik değerlerden $ke=0$ hariç hepsinin sabit kalması koşulu ile benzetim çalışması sonucu Şekil 3.8 (a)'da verilmiştir.



(a)

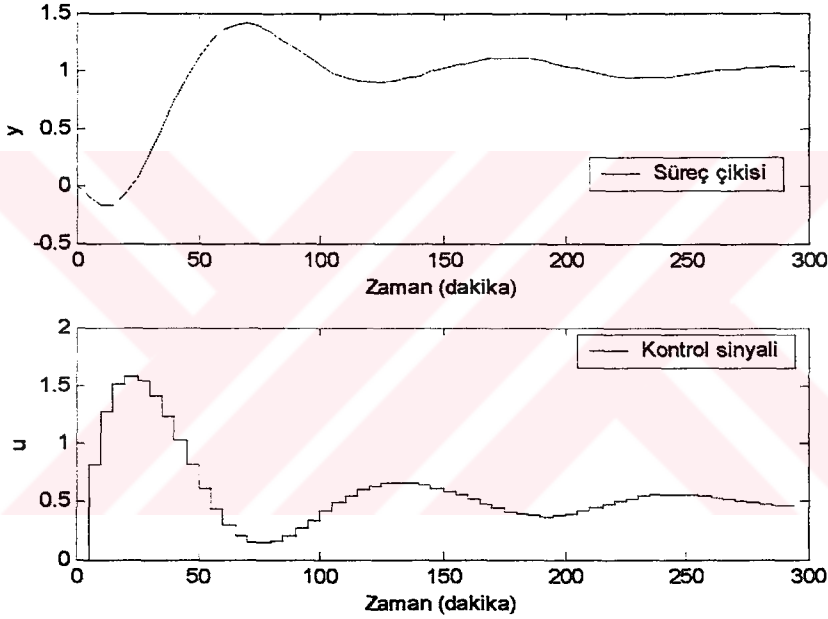


(b)

Şekil 3.8 Örnek 3.1 için $N_d=1$, (a) $k_e=0$, (b) $k_e=0.5$

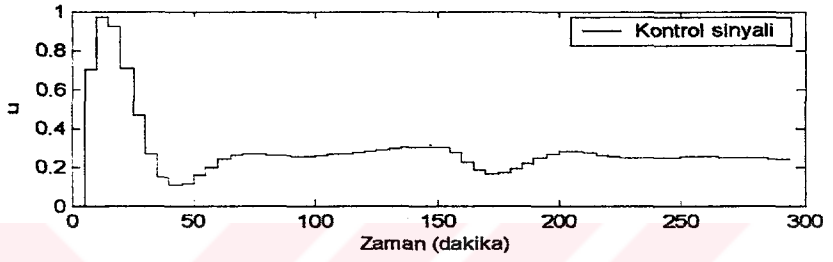
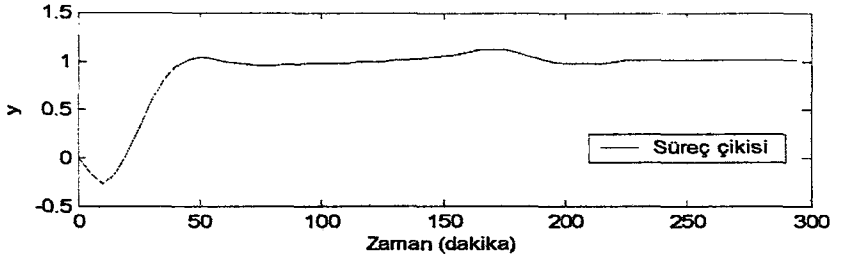
Şekil 3.8 (b)'de $ke=0.5$ için sonucu görülmektedir. Sonuç olarak bakıldığında bozucu etkinin her üç hali içinde sistem istenilen değer olan 1'e ulaşmaktadır.

Aynı örnek için ikinci koşul olan $N_u=2$ yaparak, diğer değerlerin aynı kalması koşulu ile tekrar sırasıyla üç bozucu etki içinde benzetim çalışması gerçekleştirilmiştir.

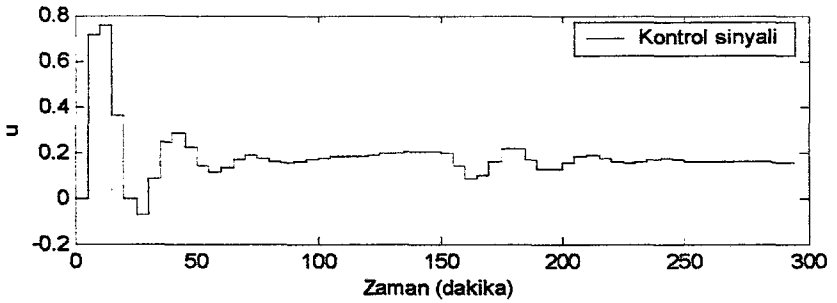
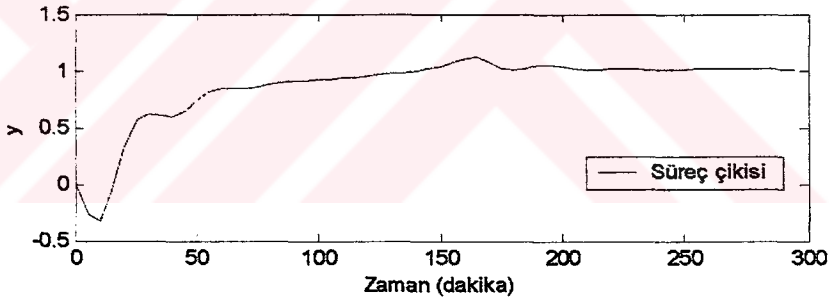


Şekil 3.9 Örnek 3.1 için $N_u = 2$. $ke = -0.5$

Şekil 3.9 ve Şekil 3.10 (a) ve (b)'de kontrol ufku sayısının artması sürecin kontrol sinyalinin ani hareket yapmasına sebep olduğu görülmektedir. Fakat sürecin istenilen değere ulaşma hızı artmıştır.

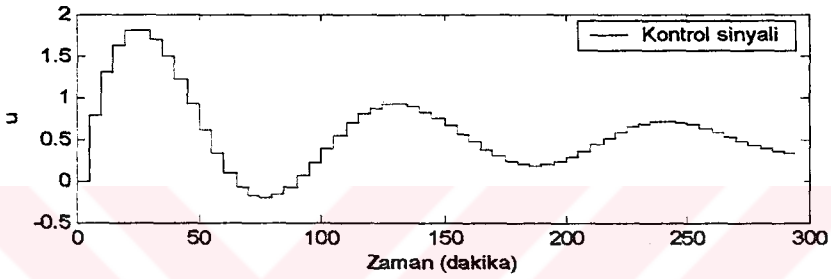
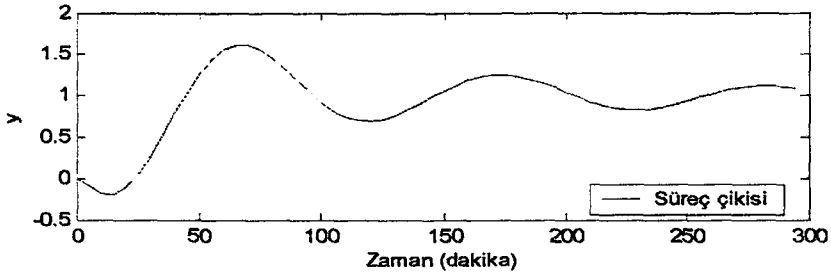


(a)

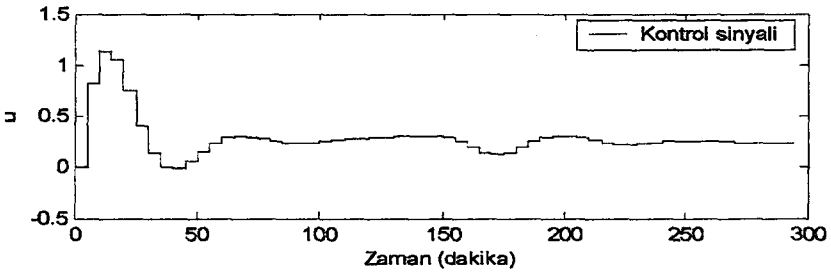
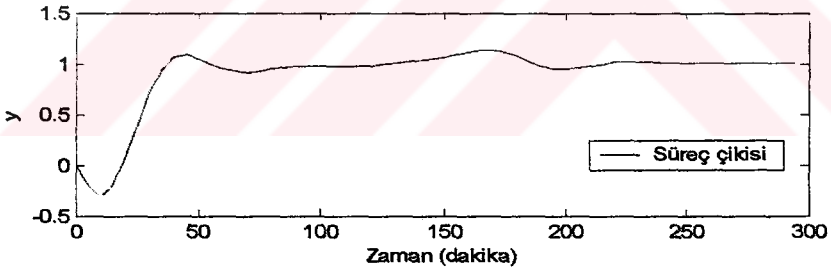


(b)

Şekil 3.10 Örnek 3.1 için $N_u=2$, (a) $k_e=0$, (b) $k_e=0.5$

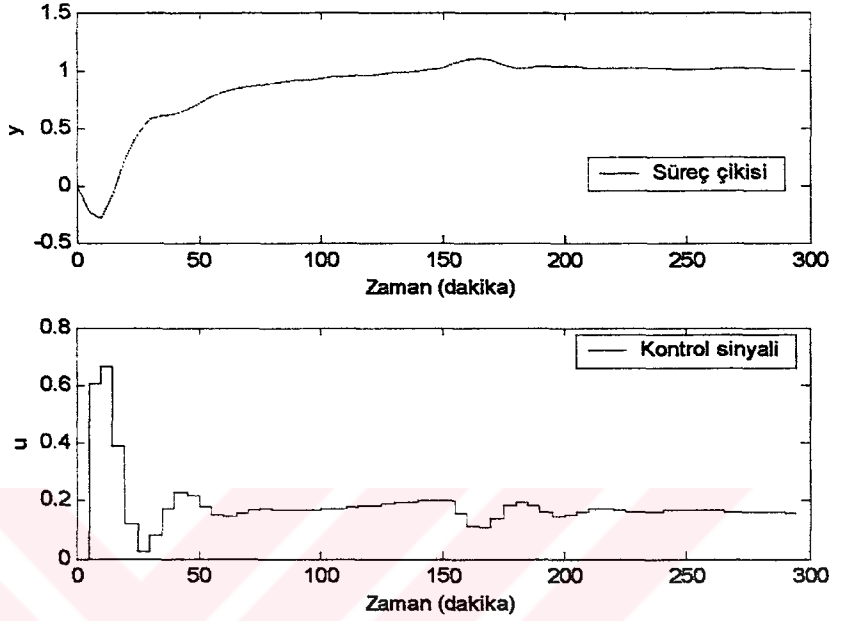


(a)



(b)

Şekil 3.11 Örnek 3.1 için $N_u=3$, (a) $k_e=-0.5$, (b) $k_e=0$



Şekil 3.12 Örnek 3.1 için $N_u = 3$, $k_e = 0.5$

Sonuç olarak Şekil 3.11 (a) ve (b), Şekil 12’de ve örnek 3.1 sonuçlarını gösteren bütün şekillerde görülmektedir ki artan kontrol hareketi sürecin istenilen değere ulaşma süresini daha azaltmıştır. Kontrol hareketinin artması kontrol sinyalinin aldığı değerlerini yükseltmiştir. Bu açıdan istenmeyen bir durum söz konusudur. Diğer bir anlatımla, kontrol sinyalinin çok keskin cevaplar verdiği görülür bu da istenilen bir durum değildir. O yüzden her ikisi arasında optimum seviyede seçilecek değerler sürecin daha kararlı ve düzgün çalışmasını aynı zaman da en az kontrol sinyali değeri ile daha düşük maliyetle çalışabileceğini göstermektedir. Bütün bozucu etkiler ve değişen kontrol hareketleri sonucunda görülmektedir ki süreç istenilen değere ulaşmaktadır.

3.7 Çok Girişli – Çok Çıkışlı (ÇGÇÇ) Sistemler için MÖK Algoritması

Tek amaçlı yaklaşım konusu olan TAMÖK geliştirilerek çok değişkenli sistemlere uygulanmıştır. Gerçekte algoritma yapısı olarak, tek girişli – tek çıkışlı “TGTÇ” ve çok girişli – çok çıkışlı “ÇGÇÇ” (Multi Input – Multi Output, MIMO) arasında temelde herhangi bir fark bulunmamaktadır (Brosilow and Joseph, 2002; Mollow et al., 2002).

Şekil 3.13’deki çok giriş – çok çıkış bir sistem düşünüldüğünde y_j ve u_i arasındaki ilişki Tek Girişli – Tek Çıkışlı sistemde olduğu gibi modellenebilmektedir.

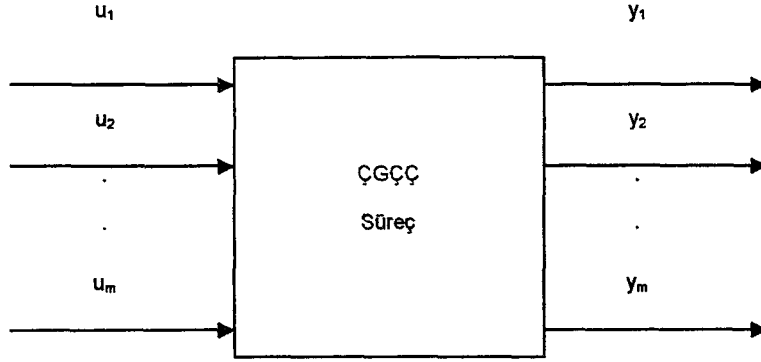
$$\Delta y_j = A_{ij} \Delta u_i \quad (3.57)$$

Burada A_{ij} dinamik matrisleri, i . girişin j . çıkış olarak değişimini gösteren bir ilişkiye sahiptir. Elde edilen bu dinamik matrisi kullanarak, çıkışı öngörmek mümkün olacaktır.

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_m \end{bmatrix} = \begin{bmatrix} y_1^{N_2} \\ y_2^{N_2} \\ \cdot \\ \cdot \\ y_m^{N_2} \end{bmatrix} + \begin{bmatrix} A_{11} & A_{12} & \cdot & \cdot & A_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{m1} & A_{m2} & \cdot & \cdot & A_{mm} \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \cdot \\ \cdot \\ \Delta u_m \end{bmatrix} + \begin{bmatrix} d_{y1} \\ d_{y2} \\ \cdot \\ \cdot \\ d_{ym} \end{bmatrix} \quad (3.58)$$

Eşitlik 3.58’de görüldüğü gibi rakamlar ve işaretler sistemi TGTÇ’de olduğu gibi yapılmaktadır. $y_i^{N_2}$, gelecekte öngörülen çıkış i değerlerinin öngörülü vektörünü simgelemektedir. Eğer model düzenlenerek tekrar yazılırsa Eşitlik 3.59 elde edilir.

$$y = y^{N_2} + A\Delta u + d_y \quad (3.59)$$



Şekil 3.13 Çok Girişli – Çok Çıkışlı sistem modeli

ÇGÇÇ algoritması için ilk olarak bozucu etkisini gösteren vektörün hesaplanması gerekir. Bunun için aynı TGTÇ sistemlerde olduğu gibi bir metot izlenir ve Eşitlik 3.60'daki formda ifade edilir.

$$d_{y_i} = \begin{bmatrix} d_{y_i,t} \\ d_{y_i,t+1} \\ \vdots \\ d_{y_i,t+N_2} \end{bmatrix} \quad (3.60)$$

Burada ifade daha belirgin halde yazılacak olursa Eşitlik 3.61'deki form oluşur.

$$d_{y_i,t} = y_{i,t}^m - y_{i,t}^{N_2} \quad (3.61)$$

Burada görülmektedir ki, aynı zamanda oluşacak olan bozucu etkiler birbirine eşit alınır. Böylelikle sistem çözümü algoritması daha çözülebilir forma da sokulmuş olur.

$$d_{y_i,t+1} = d_{y_i,t+2} = \dots = d_{y_i,t+N_2} = d_{y_i,t} \quad (3.62)$$

Bundan sonraki adımda, yapılması gereken referans yörüngesi olarak adlandırılan yörüngenin hesaplanması olacaktır. Basit formda yazılacak olursa Eşitlik 3.63'deki denklem elde edilir.

$$y_{i,t+1}^r = y_{i,t+2}^r = \dots = y_{i,t+N_2}^r = y_i^{set} \quad (3.63)$$

Burada referans yörüngenin hesaplanmasında TGTC sistemlerdeki uygulanan metot kullanılır. Bir sonraki basamakta, kontrol hareketinin hesaplanması gerekmektedir. Bu Eşitlik 3.64'de verilmiştir.

$$\Delta u = (A^T A + D)A^T (y^r - d_y - y^{N_2}) \quad (3.64)$$

Burada, D bastırma faktörünün oluşturduğu matris, bir köşegen matrisidir. Oluşabilecek tekilliği (singular) engellemektedir.

Elde edilen değerler saklanarak bir sonraki işlem basamağına geçilir. Burada y^{N_2} 'nin yeni değerinin hesaplanması gerekmektedir ki bu değer sayesinde öngörüle bulunabilmek mümkün olsun.

$$y^{N_2} = y^{N_2} + A\Delta u^1 + d_y - y^r \quad (3.65)$$

Eşitlik 3.65'de kontrol sinyali için kullanılacak vektör Δu^1 , sadece t_t anındaki kontrol hareketinin vektörünü içermektedir. Bu da Eşitlik 3.66'de verilmiştir.

$$\Delta u_i^1 = \begin{bmatrix} \Delta u_{i,1} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.66)$$

Son olarak da bir sonraki örnekleme zamanındaki t_{t+1} hareketin hesaplanması olacaktır. Bu noktada y^{N_2} 'nin değerleri aynı TGTC'de olduğu gibi güncellenmiştir.

ÇGÇÇ sistemler için geliştirilen bu algoritmada, bütün çıkışlar kontrol objesinin formüle edilmiş çıkışlarındaki ağırlıklara eşit olarak kabul edilir ve Eşitlik 3.67'de gösterilmiştir.

$$\text{Min}_{\Delta u} (y^r - y)^T Q (y^r - y) + \Delta u^T R \Delta u \quad (3.67)$$

Burada "Q" ve "R" giriş ve çıkışların oluşturduğu ağırlıkların matrisleridir.

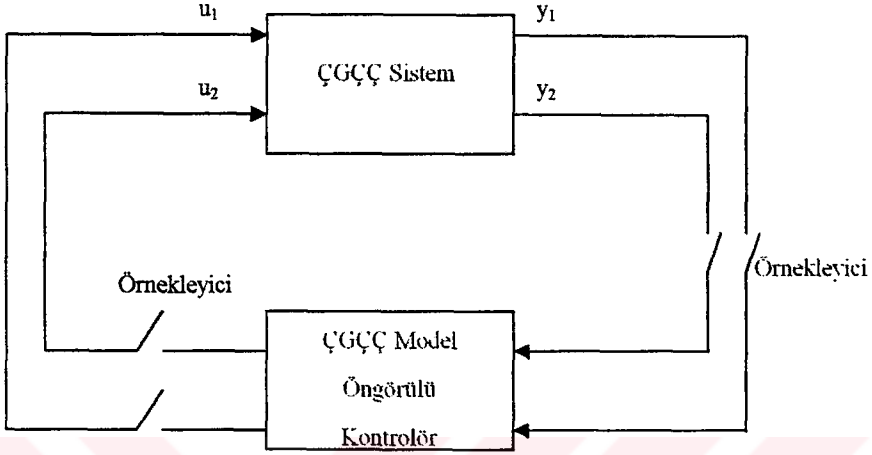
ÇGÇÇ algoritmasının parametrelerinin ayarlanmasında TGTC'deki yöntemin aynısı kullanılmaktadır. Çıkış ufku N_2 ve kontrol ufku " N_u " olarak adlandırılır.

3.7.1 ÇGÇÇ Model Öngörülü Kontrol Uygulaması

Örnek 3.2: Çok girişli – çok çıkışlı bir sürecin model öngörülü kontrol algoritması ile çözümlenmesi amaçlanmıştır. Kullanılacak olan algoritma TGTC sisteminde olduğu gibidir. Transfer fonksiyonu verilen bir sürecin denklemi Eşitlik 3.68'deki gibi olsun.

$$\begin{aligned} y_1(s) &= \frac{4.05(1-13.5s)}{(1+13.5s)(50s+1)} u_1(s) + \frac{1.77(1-14s)}{(14s+1)(60s+1)} u_2(s), \\ y_2(s) &= \frac{5.39(1-9s)}{(1+9s)(50s+1)} u_1(s) + \frac{5.72(1-7s)}{(1+7s)(60s+1)} u_2(s). \end{aligned} \quad (3.68)$$

Örnek için verilen diğer parametreler kullanılarak, ÇGÇÇ algoritması için Şekil 3.14'de blok diyagramı verilen sürecin çözümü yapılacak olursa (Brosilow and Joseph, 2002),



Şekil 3.14 Çok Girişli – Çok Çıkışlı süreç blok diyagramı

Örnek 3.2 için verilen parametrelerin açıklaması aşağıdaki gibidir.

$N_2 \rightarrow$ öngörü ufku uzunluğu,

$t_s \rightarrow$ örnekleme zamanı,

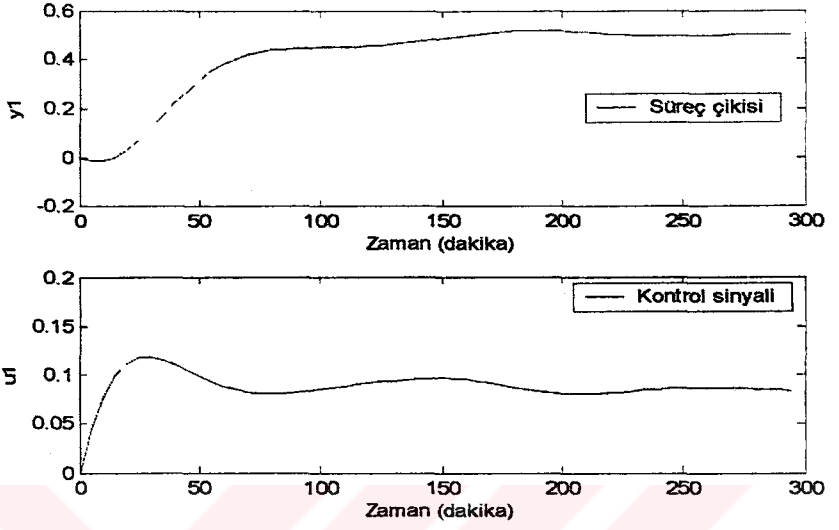
$N_u \rightarrow$ kontrol hareket sayısı,

$\beta \rightarrow$ bastırma katsayısı,

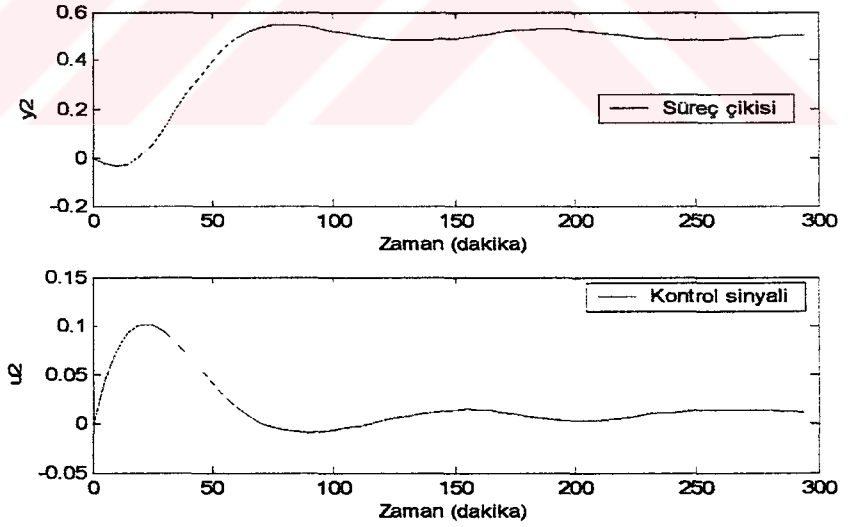
$k_e \rightarrow$ model için bozucu etki,

$n_t \rightarrow$ zaman da simülasyon (300 dk.)

Algoritmada kullanılacak nümerik değerler $N_2=30$, $t_s=5$, $N_u=10$, $\beta=1$ ve $n_t=60$ olarak verilmiştir. Buna göre model bozucu etkisi istenen değere göre "0" alınmış ve benzetim çalışması sonucu Şekil 3.15 ve 3.16'da gösterilmiştir.



Şekil 3.15 Örnek 3.2 için çıkış y_1 ve kontrol sinyali u_1



Şekil 3.16 Örnek 3.2 için çıkış y_2 ve kontrol sinyali u_2

Bu örnekte y_1 ve y_2 'nin değeri "0.5" olması istenilmektedir. Burada, modelleme de herhangi bir bozucu etki kullanılmadığı için sonuç ve kontrolör cevapları iyidir. Örnek için alınan zaman cevabı sürecin açık çevrim zaman cevabına yakındır. Aradaki önemli farklılık, hesaplanan her iki kontrol girişi başlangıçta oldukça büyüktür. Bu da göstermektedir ki sürecin kontrolörü durağan durum cevabına etkin bir şekilde ulaşmak istemiştir. Daha fazla öngörü ufku kullanılması, kontrolörün ani hareket göstermesine ve değerinin büyümesine yani daha baskın bir durağan durum cevabı göstermesine sebep olmaktadır. Bu yüzden optimum bir değer seçilerek bu maliyet fonksiyonun değerini arttırabilecek hareketlerden kaçınmak gerekir. Bununla birlikte algoritma da daha etkin ve hızlı kullanılabilir.

4. YAPAY SİNİR AĞLARI

4.1 Giriş

Yapay Sinir Ağları (YSA), doğrusal olmayan eşleme, paralellik, öğrenme ve uyarlanma gibi özelliklere sahiptir. YSA paralellik özelliğine sahip, algoritmik ve sayısal olmayan bir bilgi işleme sistemidir. Basitçe çok sayıda birbiri ile bağlantılı sinir hücresi (nöron) olarak isimlendirilen işlem birimlerinden oluşmaktadır.

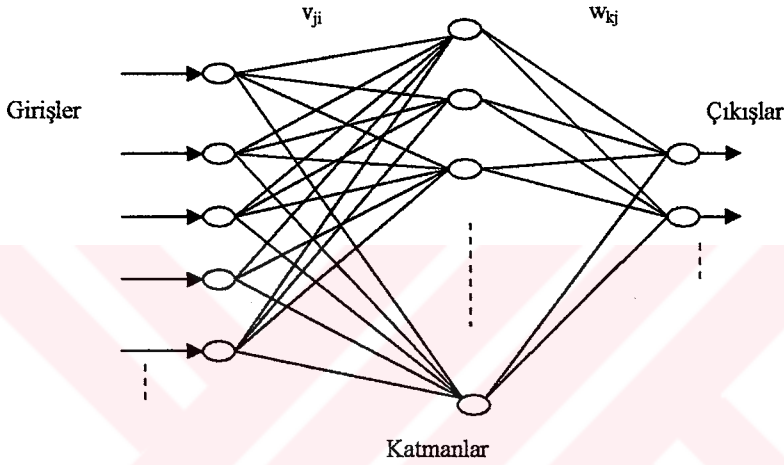
YSA'nın bu sayılan özelliklerinden ötürü son yıllarda duyulan ilgi büyük bir artış göstermiştir. Endüstri, sanayi ve akademik araştırmalar gibi sektörlerde çalışanlar, yapay sinir ağlarını kendi uzmanlık alanlarında, problem çözümünde bir araç olarak kullanılmaktadır. İnsan zekasına has olan kavramların sayısal olarak ifade edildiği ve işlendiği, böylece de öğrenme gibi işlevlerin yerine getirilebildiği görülmüştür (Principe et al., 2000; Zurada 1992).

4.2 YSA Temel Özellikleri

Yapay sinir ağları, insan sinir hücrelerinin ve sinir ağının biyolojik yapısından esinlenerek yapılmaya çalışılmıştır. Diğer bir deyişle, YSA insan sinir sisteminin en küçük birimi olan nöronların yani sinir hücresinin fonksiyonunu gören temel elemanlardan oluşmaktadır. Bu nedenle, insan beyninin bazı özelliklerini taşırlar. Mesela, tecrübe ederek öğrenebilirler, önceden öğrenilmiş bilgilerden tümevarım yapabilirler ve yeni sonuçlar çıkarabilirler.

İnsan beyinde yaklaşık 10^{10} sinir hücresi ve bir hücre başına bağlantı sayısı 10^4 civarında olduğu düşünülürse sistemin ne kadar büyük olduğu anlaşılır. Her nöron kendi bağlantılarından, diğer nöronlardan veya dış dünyadan bilgi alabilir.

Nöronlar birden fazla çıkış işareti verebilir ve hat boyunca bütün her yerinde aynı bilgi mevcuttur. İşaret kollar arasında bölünmez, ağdaki diğer nöronlardan gelen bazı kollarda son bulabilir. YSA'nın fiziksel yapısı basitçe Şekil 4.1'de gösterilmiştir (Song and Koivo, 1999).

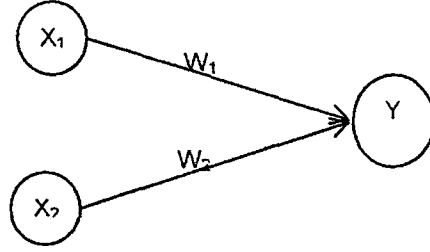


Şekil 4.1 Yapay sinir ağının fiziksel bağlantısı

4.2.1 Yapay Sinir Hücresi Yapısı

Yapay sinir hücresi, bir "t" anında yalnızca bir işaret gönderebilir. Bu işlem esnasında nöronun girişine gelen işaretler ağırlık katsayıları ile çarpılıp, daha sonra da toplanırlar. Bu durum yapay bir nöronda Şekil 4.2'de görüldüğü gibi izah edilebilir.

Şekil 4.2'de görüldüğü gibi x_1 , x_2 nöronlarından girişler alan "Y" nöronu ağırlık katsayıları ile birlikte hesaplanacak olursa Eşitlik 4.1'deki gibi olur. Burada "y_g" nöronun girişini kast etmektedir.



Şekil 4.2 Yapay sinir hücresi yapısı

$$y_g = w_1 x_1 + w_2 x_2 \quad (4.1)$$

Bu giriş bilgisi ile nöronun aktivasyon fonksiyonuna uygulanarak çıkış değeri elde edilir. Bu fonksiyon doğrusal olmayan bir fonksiyondur. Aynı zamanda eşik değeri seçimi “b” ile birlikte bu fonksiyon seçimi yapay nöronun modelini belirler. Genelde sigmoid fonksiyonu kullanılır. Sonuçta çıkış “y” olur (Eşitlik 4.2).

$$y = f(y_g) \quad (4.2)$$

4.2.2 Yapay Sinir Ağ Yapıları

Yapay sinir hücresinin matematiksel ifadesi hatırlanacak olursa (Bkz. Eşitlik 4.1) gelen giriş değerleri ile ağırlıkların çarpımlarının toplamına eşittir. Bunu daha açıkça matematiksel olarak yazacak olursak Eşitlik 4.3 olur.

$$\text{Net} = \sum_{i=1}^n x_i w_i \quad (4.3)$$

Eşitlik 4.3’de “Net” adı verilen ifade ağırlıklı girdilerin cebirsel toplamıdır. “x” ise giriş vektörü, “w” ise ağırlık vektörü olarak adlandırılır.

Yapay sinir ağı katmanlar olarak adlandırılan bölümlerden oluşmaktadır (Bkz. Şekil 4.1). Eğer katman sayısı tek ise bunlara tek katmanlı sinir ağı olarak adlandırılır. Şayet katman sayısı birden fazla ise bu tipteki sinir ağlarına çok katmanlı sinir ağları denir. Çok katmanlı sinir ağlarda her katmandaki nöron sayısı farklı olabilir (Hines, 1997).

4.3 Yapay Sinir Ağında Öğrenme

Yapay sinir ağı bir problemi çözmeyi öğrenebilir. Eğitim ve / veya öğrenme sinir ağların tercih edilmesindeki en büyük sebeptir. Öğrenme, ağda bulunan nöronlar arasındaki bağlantı ağırlıklarının değiştirilmesi ile olur. Tek nöronun çıkışı, transfer fonksiyonun aynı olması koşulu ile çıkışını yalnızca; gelen işaret ve nörona bağlı giriş ağırlıkları belirler. Nöronun gelen bilgiye doğru cevap vermesi ve performansı bağlantı ağırlıklarına bağlıdır. Eğitim ve öğrenme aynı anlama gelmezler. Eğitim, ağın öğrenmesi işlemidir. Öğrenme ise eğitim işleminin sonucudur. Eğitim ağa dışardan yapılan müdahaledir. Öğrenme ise ağın içerisinde oluşan olaydır. Yapay nöron ağların eğitilmesinde üç tip vardır.

Bunlardan ilki, öğreticili eğitmedir (supervised training); Bu yöntemde ağ giriş ve çıkış verileri mevcuttur. Her adımdan sonra ağ kendi çıkışını hedefle karşılaştırır, çıkış hatası azaltılacak şekilde ağırlıklar ayarlanır (Zurada, 1992).

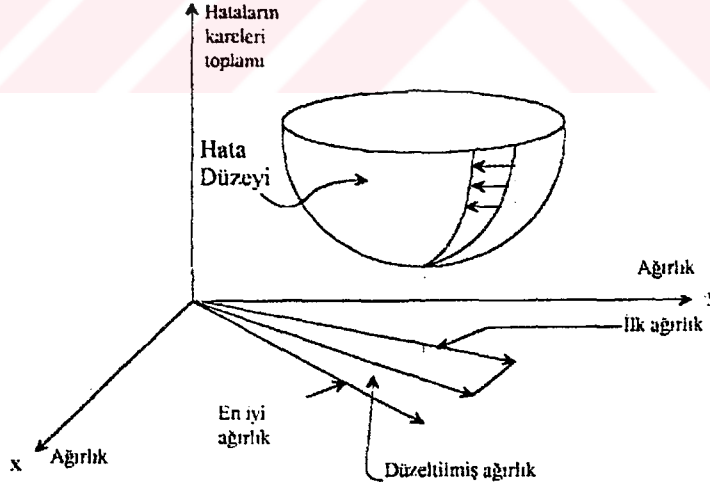
İkinci yöntem ise, derecelendirmeli eğitmedir (graded training). Öğreticili eğitmenin benzeri fakat giriş işaretine karşılık gelen çıkış işareti "başarılısın" "başarısızsın" veya "çok yüksek", "çok düşük" türünde değerlendirmeler yapılır.

Son yöntemde ise kendini düzenleme ile eğitim (self - organization training) olarak bilinir. Bu yöntemde ağa giriş örnekleri verilir. Performans seviyesi bilgisi geri besleme ile verilmez Ağ giriş işaretine göre kendisini düzenler.

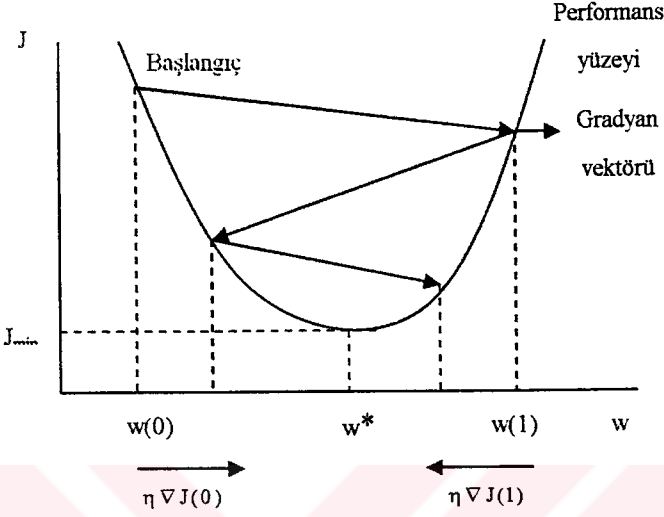
4.4 Hata Küçültülmesi ve En Dik İniş Yöntemi

Hatanın küçültülmesi, tanım olarak ölçülen çıkış değeri ile istenen çıkış değeri arasındaki farkın kareleri toplamının azaltılması olarak ifade edilebilir. Bu tanıma geometrik yolla izah etmek mümkündür. Burada, ağırlıkları bütün değerlerin hatalarının kareleri toplamına uygun şekilde yazılır “x-y-z” koordinat sisteminde çizilirse, hata yüzeyi Şekil 4.3’deki gibi olur. Bu yüzeyin en alt kısmı hataların kareleri toplamının en küçük değeridir. Eğitim sırasında ağırlıklar kümesinin değeri en iyi olması için en altta bulunmalıdır (Principe et al., 2000).

Bu çalışmada ağırlıkların güncellenmesi en dik iniş yöntemi ile yapılmıştır. Şekil 4.3’de ağırlıklar ile değişimi görülmektedir. Burada, iniş yönü ölçütün ağırlıklara göre gradyanın tersi yönüdür. Bu şekilde, gradyan tersi yönünde adım büyüklüğü kadar ilerlenerek, en küçük değere doğru ilerleme esasına dayanır.



Şekil 4.3 Hatanın küçültülmesi ve en dik iniş yöntemi



Şekil 4.4 Performans yüzeyi ve gradyan

Bu yöntemde, ağırlıklara başlangıç rasgele küçük değerler atanmaktadır. Daha sonra, giriş verileri YSA'ya uygulanıp, çıkışlar elde edilmektedir. Elde edilen çıkışlar ile istenilen değerler arasında karesel hata azaltılacak şekilde en dik iniş yöntemi ile ağırlıklar güncellenmektedir.

4.5 En Dik İniş Yöntemi (EDİY) Tanımı

EDİY (steepest descent) yöntemi basit olmasına karşın etkin bir yöntemdir. YSA literatüründe geriye-yayılım (GY) ismiyle kullanılır.

$\frac{\partial J}{\partial w} = 0$ olması maliyet fonksiyonunu en küçük değerine götürecektir.

GY'nin matematiksel formu Eşitlik 4.4 ve Eşitlik 4.5'de verilmiştir.

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t) \quad (4.4)$$

$$\Delta w_{kj}(t) = -\eta \left(\frac{\partial J(t)}{\partial w} \right) = -\eta \nabla J_w(t) \quad (4.5)$$

Eşitlik 4.5'de t adımı, η ise t anındaki adım büyüklüğüdür. Bundan sonra yenilenecek olan bu değer bir sonraki adımda Eşitlik 4.4 deki ifadeye uygulanarak yeni değerler bulunacaktır. Burada η için bulunabilecek minimum adım değeri Eşitlik 4.6'da verilmiştir.

$$\min_{\eta} J(w - \eta \frac{\partial J}{\partial w}) \quad (4.6)$$

Burada η yeterince küçük olması gerekir. Doğrusal arama yöntemleri kullanılarak her adımda η için en uygun değerler belirlenebilir. Ancak, bu durumda işlem külfeti artacaktır (Scales, 1985; Principe et al., 2000).

5. YSA TEMELLİ MODEL ÖNGÖRÜLÜ KONTROL

5.1 Giriş

Bu bölümde, YSA temelli MÖK yapısı geliştirilmiştir. YSA bu yapıda dinamik sistemlerin tanılama modeli olarak kullanılmıştır. YSA tanılama modelleri sistemlere ait giriş – çıkış verileri kullanılarak, geriye yayılma algoritması ile eğitilmiştir. Elde edilen tanılama modelleri MÖK yapısında sistemin çıkışlarının öngörülmesinde kullanılmıştır. Bu yöntemle gerçekleştirilen benzeşim çalışmaları ile performansı test edilmiştir.

Günümüzde YSA dinamik sistemlerin modellenmesi ve kontrolündeki önemli bir araç haline gelmiştir.(Nguyen and Widrow, 1990; Sjoberg et al., 1995; Piche et al., 2000).

Son yıllarda Model Öngörülü Kontrol (MÖK) geniş bir kullanım alanı bulmakta ve özellikle endüstriyel uygulama alanlarında bir standart haline gelmeye başlamıştır (Clarke et al., 1987; Rau and Schröder, 2002). MÖK'ün temel yapı taşı, süreç modelidir. Süreç modeli sayesinde gelecekteki kontrol hareketi hakkında öngörülecek ve böylelikle istenilen çıkış yörüngesine ulaşmak mümkün olacaktır. Dolayısı ile süreç modelinin çok iyi tanımlanması gerekmektedir.

Bir çok endüstriyel uygulamada doğrusal modellerin, doğrusal olmayan modellere göre çözümünün kolay olması yüzünden tercih edilmekte yada doğrusal olmayan süreçlerin belli sınırlar içinde doğrusallaştırılarak çözüme ulaşılmaya çalışılmaktadır. Bu tez çalışmasında, bu sorunun ortadan kaldırılması ve özellikle doğrusal olmayan süreçler için kullanılacak, model öngörülü kontrolör tasarımı için yapay sinir ağları temelli model öngörülü kontrolör yapılması hedeflenmiştir.

5.2 Doğrusal Olmayan Tanılama Modelleri

Ayrık zamanda doğrusal olmayan dinamik bir sitemin ifadesi Eşitlik 5.1 şeklindedir.

$$\begin{aligned} x(t+1) &= \phi[x(t), u(t)] \\ y(t) &= \psi[x(t)] \end{aligned} \quad (5.1)$$

Burada “x” durum vektörünü, “u” giriş vektörünü, “y” çıkış vektörünü göstermektedir. Burada ϕ ve ψ doğrusal olmayan fonksiyonlardır. ϕ ve ψ bilinmiyorsa tanılama problemi ortaya çıkar. Bu çalışmada tanılama için YSA kullanılmıştır. YSA kullanılarak elde edilen modeller sürecin giriş – çıkış davranışını temsil eden, kara kutu (black box) olarak adlandırılmış modellerdir (Maciejowski, 2002; Sjoberg et al., 1995).

Öngörülü kontrol hesaplamaları genelde “u” değerini hesaplamak içindir ve sürece uygulanmadan önce başlangıç değeri olan “ u_0 ” genelde bu değere eklenir. Fakat öngörülü kontrolde esas olarak, kontrol sinyali içindeki bir zaman basamağından diğerine olan gerekli değişimler hesaplanır (Eşitlik 5.2).

$$\Delta u(t) = u(t) - u(t-1) \quad (5.2)$$

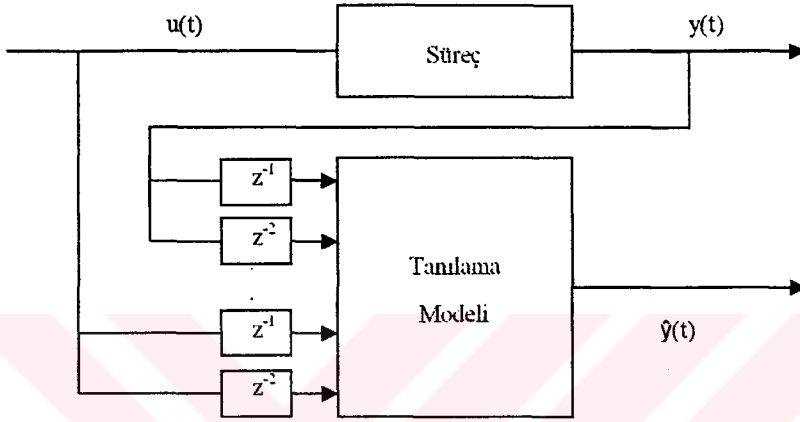
Eğer sınır şartları varsa bunlara doğru yaklaşım için u_0 'ın dikkate alınması gerekmektedir.

5.3 YSA ile Sistem Tanılama

YSA'ların doğrusal olmayan fonksiyonları benzeştirebilme özelliği tanılama çalışmalarında kullanılmasına imkan tanımaktadır. YSA ile iki temel tanılama yapısı vardır. Bunlar paralel ve seri – paralel tanılama yapılarıdır (Henson and Seborg, 1997).

5.3.1 Seri – Paralel Tanılama Yapısı

Şekil 5.1’de görüldüğü gibi, bu tanılama yapısında, sistem çıkışının geçmiş değerleri tanılama modeli girişine uygulanmaktadır.



Şekil 5.1 Seri – paralel tanılama, (z^{-1} – geriye kaydırma operatörü)

Bir örnekle izah edilecek olursa, bir dinamik süreç için örneklenmiş giriş –çıkış veri seti Eşitlik 5.3’deki gibi olsun.

$$y(t) = F(y(t-1), \dots, y(t-K_y), u(t-1), \dots, u(t-K_u)) + d \quad (5.3)$$

Eşitlik 5.3’deki “y” çıkışı, “u” kontrol sinyalini ve “d” ise gürültü faktörünü simgelemektedir. Bu tip denklem modeline, “Dışardan Girişle Lineer Olmayan Öz-yayınım” modeli (Nonlinear Autoregressive with Exogenous Input – NARX) denir.

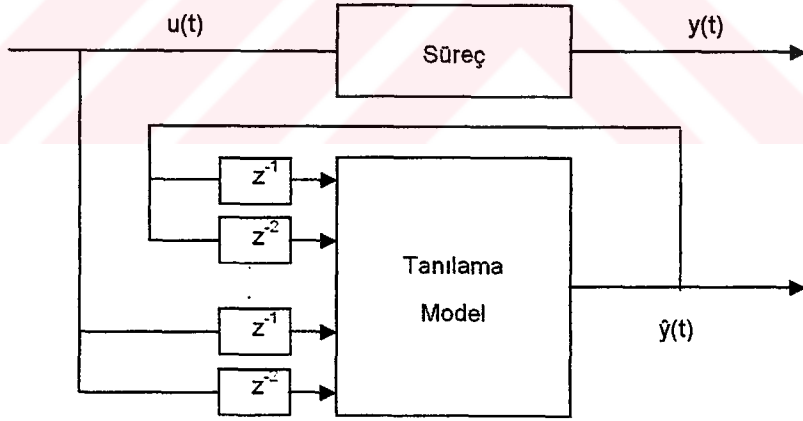
$$\hat{y}(t) = F(y(t-1), \dots, y(t-K_y), u(t-1), \dots, u(t-K_u)) \quad (5.4)$$

Şekil 5.1'deki seri – paralel tanılama modeli gösterimi Eşitlik 5.4'deki gibi ifade edilebilir. Burada süreç çıkışı “y” direk olarak modelin girişine uygulanmıştır. Aynı zamanda bu tanımlama formu, hata modeli tanımlaması olarak da ifade edilir. Sürecin gürültüsünü daha iyi öngörmektedir bu yüzden tercih edilir. Burada YSA olarak çok katmanlı ileri beslemeli bir ağ kullanılabilir (Hines, 1997).

5.3.2 Paralel Tanılama Yapısı

Şekil 5.2'de paralel tanılama yapısı geliştirilmektedir. Bu yapıda tanılama modelinin girişine kendi çıkışının değerleri uygulanmıştır.

Eşitlik 5.3'deki aynı örnek için tanımlarsak, dinamik süreç için örneklenmiş giriş – çıkış veri seti aynen alınır. Eşitlik 5.3'deki “y” çıkışı, “u” kontrol sinyalini ve “d” ise gürültü faktörünü simgelemektedir.



Şekil 5.2 Paralel tanılama, (z^{-1} – geriye kaydırma operatörü)

Şekil 5.2'deki paralel tanımlama modeline göre cevabı Eşitlik 5.5'de verilmiştir. Burada model çıkışının geçmiş değerleri model girişine uygulanmıştır.

$$\hat{y}(t) = F(\hat{y}(t-1), \dots, \hat{y}(t-K_y), u(t-1), \dots, u(t-K_u)) \quad (5.5)$$

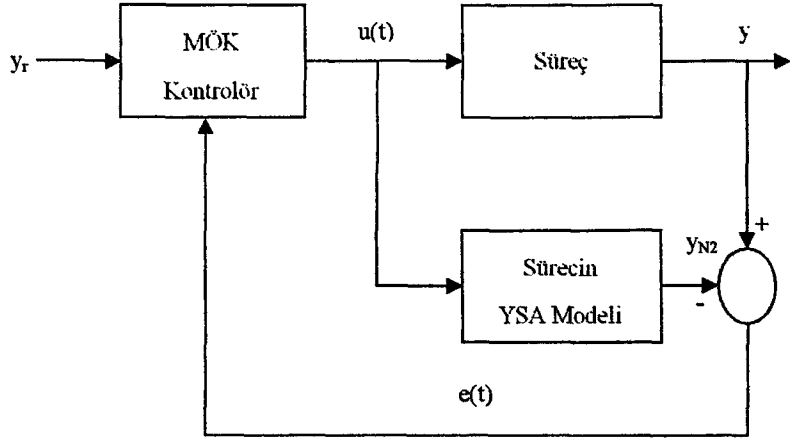
Bu metotta sistem çıkışı kullanılmadığı için bu özeliğe uygun uygulamalarda tercih edilebilir.

5.4 YSA Temelli Model Öngörülü Kontrol Yapısı

Model öngörülü kontrol konusu, sürecin kontrolü için referans alınan modeli tanımlayan genel bir ifadedir. Öngörülü kontrol bölümünde de anlatıldığı gibi birden fazla öngörülü kontrol metodu mevcuttur (Henson and Seborg, 1997). Şekil 5.3'de sürecin modeli, süreç ve MÖK kontrolöründen oluşan MÖK blok diyagramı görülmektedir.

Burada süreç modeli çıkışı ile sürecin çıkışı arasındaki fark yani öngörülen çıkış ile ölçülen çıkış arasındaki hata MÖK kontrolöre iletilmektedir. MÖK kontrolör kuvadratik PÖ'yü en küçük değere götürecek olan kontrol sinyalini üreterek sürece tekrar uygular. Burada referans yörünge MÖK kontrolöre uygulanır. Sürecin öngörülen çıkışlarının elde edilebilmesi için YSA tanımlama modeli kullanılır.

Sonuçta buradaki YSA kullanıcı tarafından seçilecek yöntemi öğrenecektir fakat algoritması ağır ve tasarımı zor olan modellerin aksine direk olarak sürecin giriş çıkış değerlerinin öğrenilmesi tercih edilmektedir. Çünkü gerçek uygulamalarda en büyük sorun hız olacaktır genelde hızlı yavaş olan uygulamalarda kullanılmasının sebebi de bu yüzdendir (Piche et al., 2000).



Şekil 5.3 YSA temelli model öngörülü kontrol yapısı

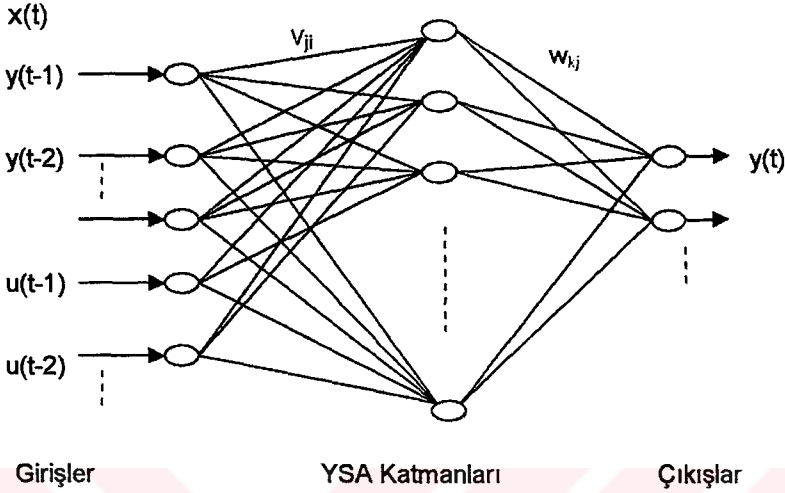
Bu elimizdeki verilerle Eşitlik 3.41'deki genel olarak tanımlanmış PÖ'nün çözümünü yapmak yani mümkün olan en küçük değerine ulaştıracak kontrol sinyalini üretmek mümkündür. Bu kontrol sinyalinin güncellenmesini en dik iniş yöntemi ile yapılabilir.

5.5 YSA ile Süreç Modelinin Eğitimi

Bu çalışmada, kullanılan YSA yapısı Şekil 5.4'de gösterilen çok katmanlı ileri beslemeli ağ yapısındadır. Süreçlerin modellenmesi seri – paralel tanımlama yapısında, bu ağ kullanılarak yapılmaktadır (Song and Koivo, 1999).

Şekil 5.4'de verilen giriş vektörü $x(t)$ Eşitlik 5.6'da tanımlanmıştır. Kontrol sinyali girişi $u(t)$, süreç model çıkışı $y(t)$ 'dir.

$$\begin{bmatrix} x(t) \\ u(t-1) \end{bmatrix} = \begin{bmatrix} y(t-1) & y(t-2) & \dots & y(t-q) \\ u(t-1) & u(t-2) & \dots & u(t-q) \end{bmatrix}^T \quad (5.6)$$

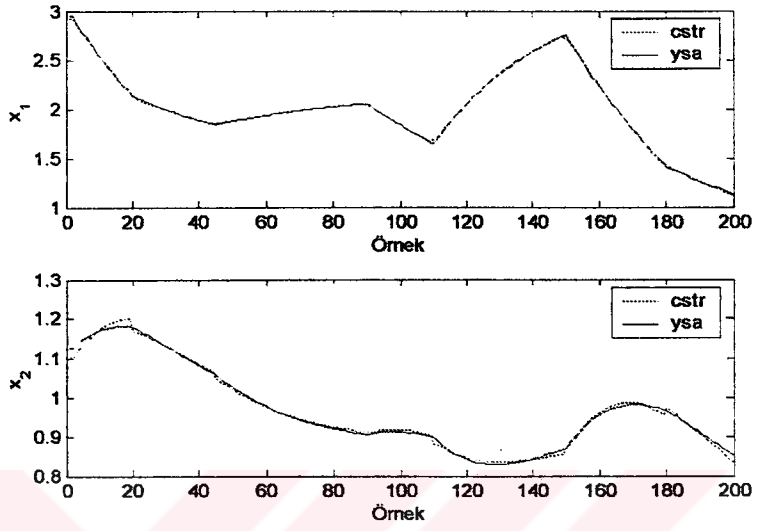


Şekil 5.4 Süreç modeli için YSA temelli tasarım

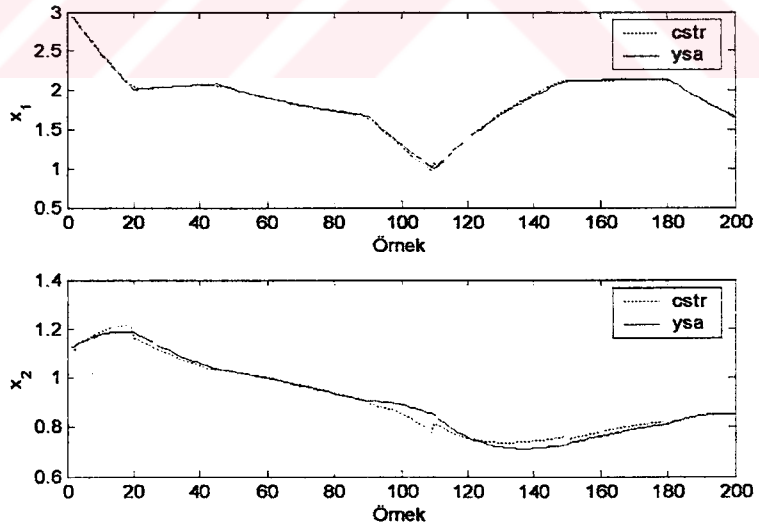
YSA yapının eğitimi en dik iniş yöntemi ile yapılmıştır. Bu konudaki ayrıntılı bilgi yapay sinir ağları bölümünde anlatılmıştır (Bkz. Bölüm 4).

Oluşturulan YSA tanılama modeli CSTR modeli için elde edilmiş olan eğitim veri seti cevabı Şekil 5.5'de gösterilmiştir. Burada x_1 ve x_2 durum değişkenleridir. Kesikli çizgilerle gösterilmiş olan sürecin çıkışıdır. Sürekli çizgi ile gösterilense süreç modeli olan yapay sinir ağı çıkışıdır. Örnek sayısı görüldüğü gibi 200 olarak alınmıştır.

YSA tanılama modeli test veri seti cevabı Şekil 5.6'da gösterilmektedir. Görüldüğü gibi model çıkışı süreç cevabını iyi bir şekilde takip etmektedir.



Şekil 5.5 CSTR süreci için eğitim veri seti

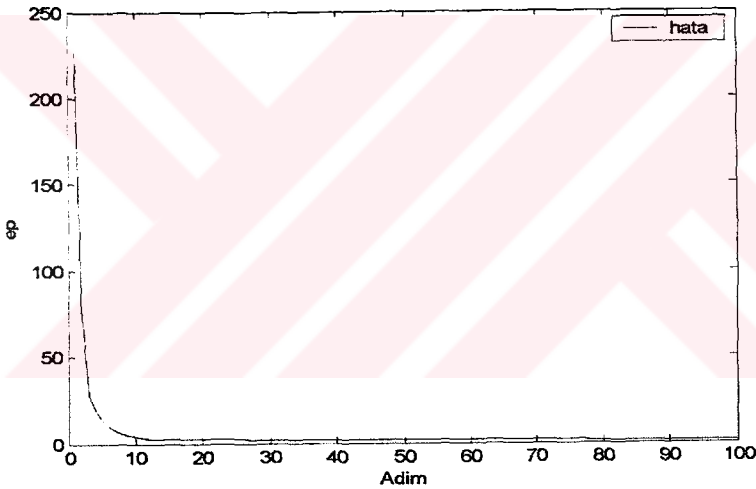


Şekil 5.6 CSTR süreci için test veri seti

YSA eğitiminde kullanılan ölçüt fonksiyonu Eşitlik 5.7'de verilmiştir. Burada model çıkışı ile sistemin çıkışı arasındaki karesel hata kullanılmaktadır.

$$j_M(w) = \frac{1}{2} \sum_{t=1}^N \|\varepsilon(t)\|^2 = \frac{1}{2} \sum_{t=1}^N \|\hat{y}(t) - y(t)\|^2 \quad (5.7)$$

Şekil 5.7'de karesel hatanın her adımdaki değişimi gösterilmektedir. Görüldüğü gibi 10. adımda önemli ölçüde azalma gerçekleşmektedir.



Şekil 5.7 YSA için en küçük kareler toplamı hatası

5.6 YSA Temelli MÖK Benzetim Çalışmaları

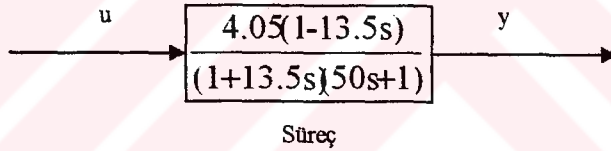
Doğrusal ve doğrusal olmayan süreçler için yapay sinir ağı temelli model öngörülmesi kontrol yapısı kullanılarak benzetim çalışmaları gerçekleştirilmiştir. Kullanılan genel blok diyagram Şekil 5.3'de verilmiştir.

5.6.1 Doğrusal Süreç

Örnek 5.1: Tek girişli – Tek Çıkışlı bir sürecin YSA temelli model öngörülü kontrol yapısı ile gerçekleştirilen benzetim çalışması için kullanılacak olan transfer fonksiyonu Eşitlik 5.8'deki gibi ve blok diyagram gösterimi de Şekil 5.8'deki gibidir.

$$y(s) = \frac{4.05(1-13.5s)}{(1+13.5s)(50s+1)}u(s) \quad (5.8)$$

Tek girişli – tek çıkışlı olan bu süreç için verilen parametreler; öngörülü çıkış ufku $N_2 = 30$, kontrol ufku $N_u = 2$, örnekleme aralığı 5 dakika ve örnek sayısı 60'dır. Aynı örneğin DMA yapısı ile çözümü Örnek 3.1'dedir. Geliştirilen YSA temelli model öngörülü kontrol çözümü aşağıda verilmektedir (Brosilow and Joseph, 2002).



Şekil 5.8 Örnek 5.1 için doğrusal süreç modeli

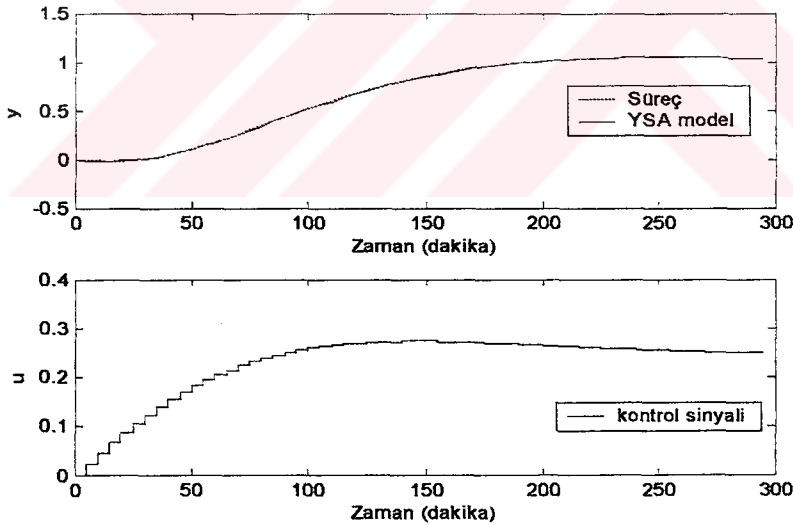
Verilen çıkışın referans değeri 1 olarak belirlenmiştir. Örnek 3.1 ile karşılaştırılması açısından kontrol ufku ve diğer parametreleri yukarıda belirtildiği gibi alınmıştır. Şekil 5.9'da görülmektedir ki verilen parametrelerle yapılan çözümde referans yörüngeye ulaşılmaktadır. DMA ile olan çözümde kontrol sinyalinin bu yeni çözüme göre değer ve salınım miktarı çok yüksektir (Bkz. Şekil 3.10 (a)).

Ani değişim ve salınım istenmeyen bir durumdur. DMA çözüm yönteminde sürecin yükselme zamanı çok küçük olmuştur, fakat kontrol

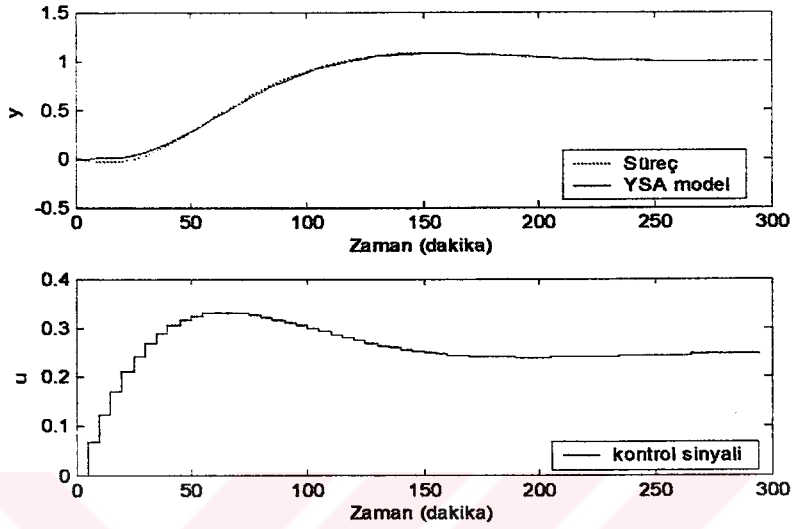
ufkunu hesapsal yükün artması sebebiyle fazla yükseltememekteyiz. YSA temelli öngörülü kontrol algoritmasında ise kontrol ufkunu belli sınıra (öngörü ufkundan uzun olmamalı Şekil 3.4’de görülmektedir) kadar arttırmak hesapsal yük açısından büyük bir sıkıntı getirmemektedir.

Bu bilgiler ışığında eğer kontrol ufku arttırılırsa yükselme zamanı Şekil 5.9’daki değerinin altına rahatlıkla düşürülebilir ve kontrol ufkunun artmasına rağmen DMA algoritmasına göre, algoritma hızı daha iyidir.

Bu yeni kontrol ufku için seçilen değer 6 olarak alınacak olursa yeni durumun gösterimi Şekil 5.10’da gösterilmiştir. Şekil 5.9 ve 5.10’daki karşılaştırma da rahatlıkla kontrol ufkunun süreç ve süreç modeli üzerindeki etkisini göstermektedir.



Şekil 5.9 Örnek 5.1 için çıkış “y”, giriş “u” ve $N_u = 2$



Şekil 5.10 Örnek 5.1 için çıkış “y”, giriş “u” ve $N_u = 6$

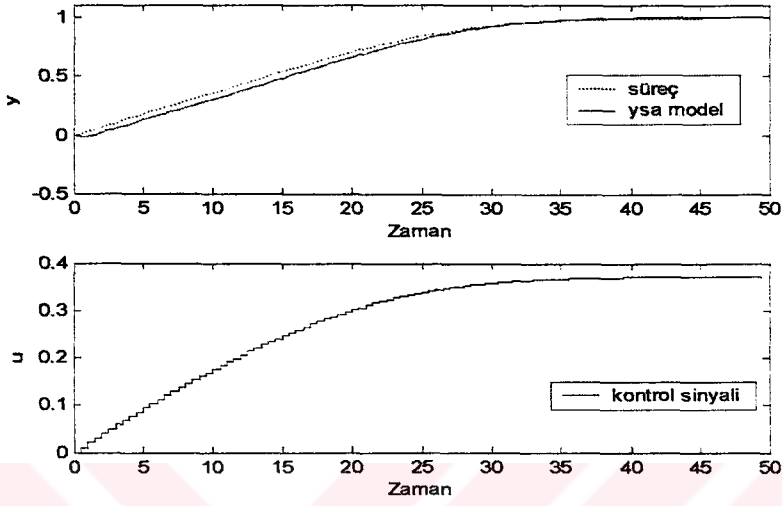
5.6.2 Doğrusal Olmayan Süreç

Örnek 5.2: İkinci bir örnek olarak Eşitlik 5.9’deki verilen ifade doğrusal olmayan sistem için benzetim çalışması gerçekleştirilmiştir (Leung and Chan, 2003). Çıkış değerini “1” olacak şekilde en uygun öngörü ufku ve kontrol ufku kullanılacaktır.

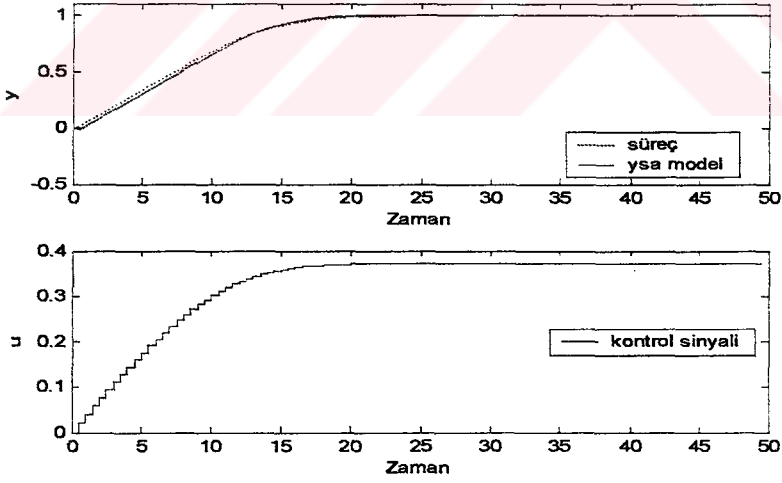
$$y(t) = 0.5y(t-1) + u(t)(1 + 0.2y^2(t-1)) + u^3(t) \quad (5.9)$$

Başlangıç koşulları “0” olarak verilmiştir. $N_1 = 1$, $N_2 = 10$ ve $N_u = 1, 2, 3$ değerleri için çıkış sonuçları değerlendirilirse aşağıdaki süreç çıkışları elde edilir (Şekil 5.11, Şekil 5.12, Şekil 5.13).

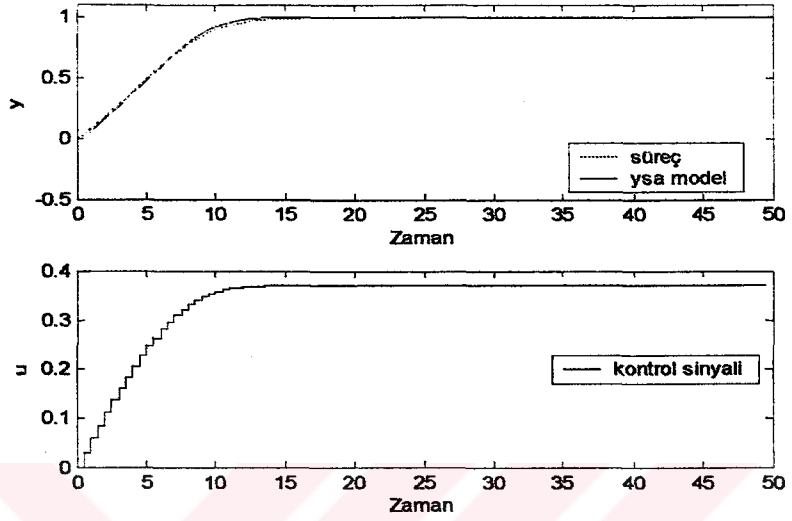
Örnek 5.2 için her üç sonuçta karşılaştırıldığında kontrol ufkunun açık etkisi görülmektedir.



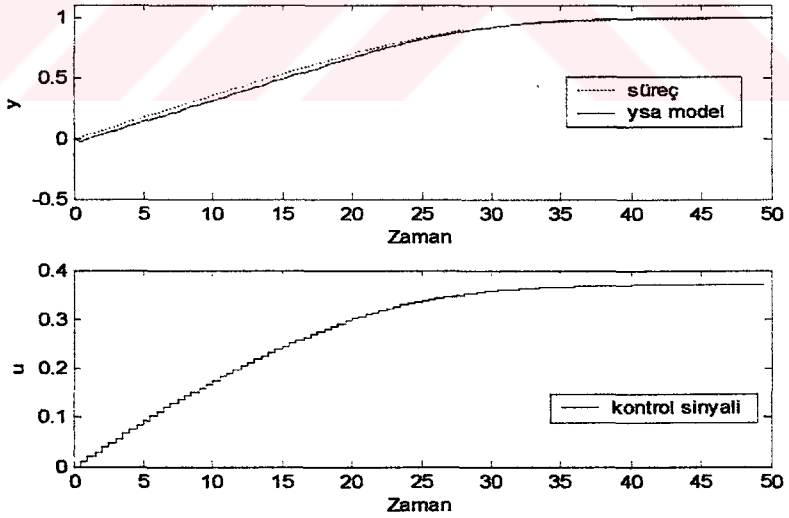
Şekil 5.11 Örnek 5.2 için “y” ve “u” ($N_2 = 10, N_u = 1$)



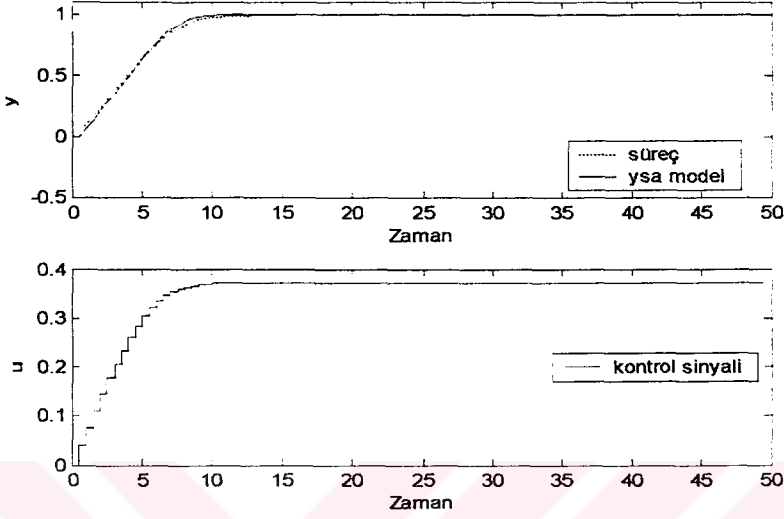
Şekil 5.12 Örnek 5.2 için “y” ve “u” ($N_2 = 10, N_u = 2$)



Şekil 5.13 Örnek 5.2 için “y” ve “u” ($N_2 = 10$, $N_u = 3$)



Şekil 5.14 Örnek 5.2 için “y” ve “u” ($N_2 = 5$, $N_u = 2$)



Şekil 5.15 Örnek 5.2 için “y” ve “u” ($N_2 = 20$, $N_u = 2$)

Bu örnek de verilen süreç çıkışının $N_2=10$ için “1” olduğu görülmektedir. Aynı uygulamada öngörü ufkunun değişiminin nasıl sonuçlara sebep olduğu sorgulanması için $N_u=2$ iken $N_2 = 5$ ve 20 olarak iki farklı durumda ele alınarak, sonuçlar karşılaştırıldığında tüm durumlarda istenilen çıkışa ulaşmaktadır. Öngörü ufkunun artırılması yükselme zamanını azaltmakta olduğu görülmektedir (Şekil 5.12, Şekil 5.14, Şekil 5.15).

5.6.3 Doğrusal Olmayan CSTR Uygulaması

Örnek 5.3: Eş ısıtıl reaktör (Isothermal Continuous Stirred Tank Reactor, CSTR) için YSA temelli model öngörülü kontrol yapısı ile benzetim çalışması gerçekleştirilmiştir. Reaktörün durum uzayı gösterimi Eşitlik 2.34’de verilmiştir (Henson and Seborg, 1997).

Eşitlik 2.34 ve 2.35'deki verilen parametrelerin nümerik değerleri hatırlatılacak olursa durum değişkenleri x_1 ve x_2 (Eşitlik 2.34), A ve B karışımlarının iki türüdür (Eşitlik 2.35). x_F , A konsantrasyonundaki besleme girişi, u besleme oranıdır. Sırasıyla $\{k_1, k_2, k_3, x_F\}$ parametrelerin değerleri $\{50 \text{ hr}^{-1}, 100 \text{ hr}^{-1}, 10 \text{ liter}/(\text{mol hr}), 10 \text{ mol/liter}\}$ olarak verilmiştir. Çıkış değeri sabit ve $y = x_2 = 1.0$ 'de olacaktır.

Yapılan nümerik çözümle en az besleme oranı ile sağlanacak denge durumu değerleri $\{x_1 = 2.5, x_2 = 1.0, u = 25\}$ bulunmuştur. Bu bilgiler ışığı altında YSA temelli model öngörülü kontrol PÖ'süne uygun şekilde Eşitlik 2.36'yı düzenlersek yeni PÖ bulunur (Eşitlik 5.10).

$$J = \sum_{i=N_1}^{N_2} \gamma_1 [2.5 - x_1(t+i)]^2 + \gamma_2 [1 - x_2(t+i)]^2 + \gamma_3 [25 - u(t+i)]^2 + \lambda \sum_{i=0}^{N_u} (\Delta u(t+i))^2 \quad (5.10)$$

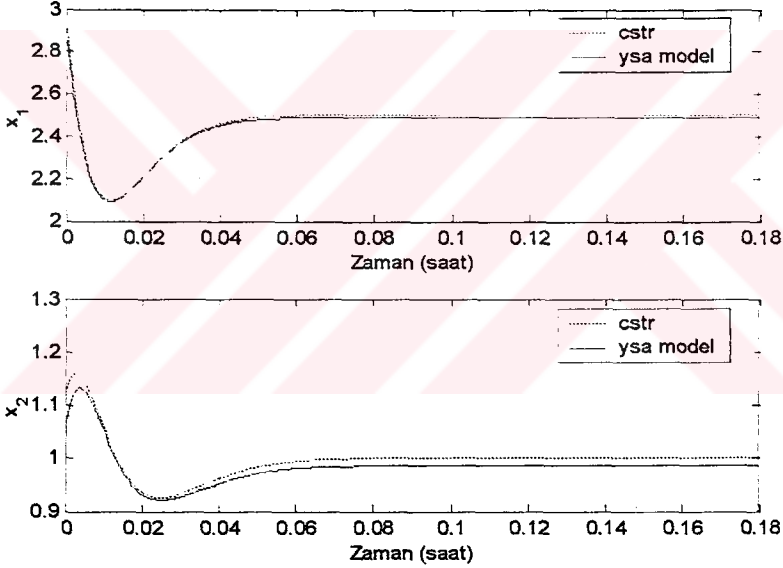
Maliyet fonksiyonu için gereken sabit parametreler, $0 < \lambda < 1$ ve $\gamma_1 = \gamma_2 = 1000, \gamma_3 = 1$ olarak alınacaktır (Bkz. Eşitlik 2.37, Eşitlik 3.42). Başlangıç koşulları Eşitlik 2.42'de verilmiştir.

Doğrusal olmayan özelliği hassas olan bu örnekte öngörü ufku çok uzun tutmak iyi olamayan neticelere götürebilir. Bu yüzden öngörü ufku N_2 'nin farklı değerleri, kontrol ufku $N_u = 1$ minimum maliyet ufku ise herhangi bir fazladan belirtilen ölü zaman olmadığı için $N_1 = 1$ olarak alınmıştır. Bu şartlar altında elde edilen sonuçlar aşağıdaki şekillerde verilmiştir (Şekil 5.16, Şekil 5.17, Şekil 5.18, Şekil 5.19, Şekil 5.20).

Sonuçlar kendi aralarında karşılaştırıldığında doğrusal olmayan CSTR uygulamasının öngörü seçiminde ne kadar hassas olunması gerektiği görülmektedir. Öngörü ufkunun gereğinden fazla arttırılması süreç çıkışında iraksamaya sebep olmaktadır. Ufuk seçiminin hassas olduğu bu durumda kontrol ufkunun çok artması da durum değişkenleri

ve kontrol sinyalini denge durumundan uzaklaştırır. CSTR uygulamasının literatürdeki Hamiltonian çözümü ile (Bkz. Şekil 2.5) olan sonuçları karşılaştırıldığında durum değişkenleri ve kontrol sinyalinin değişimi benzer olduğu görülmektedir.

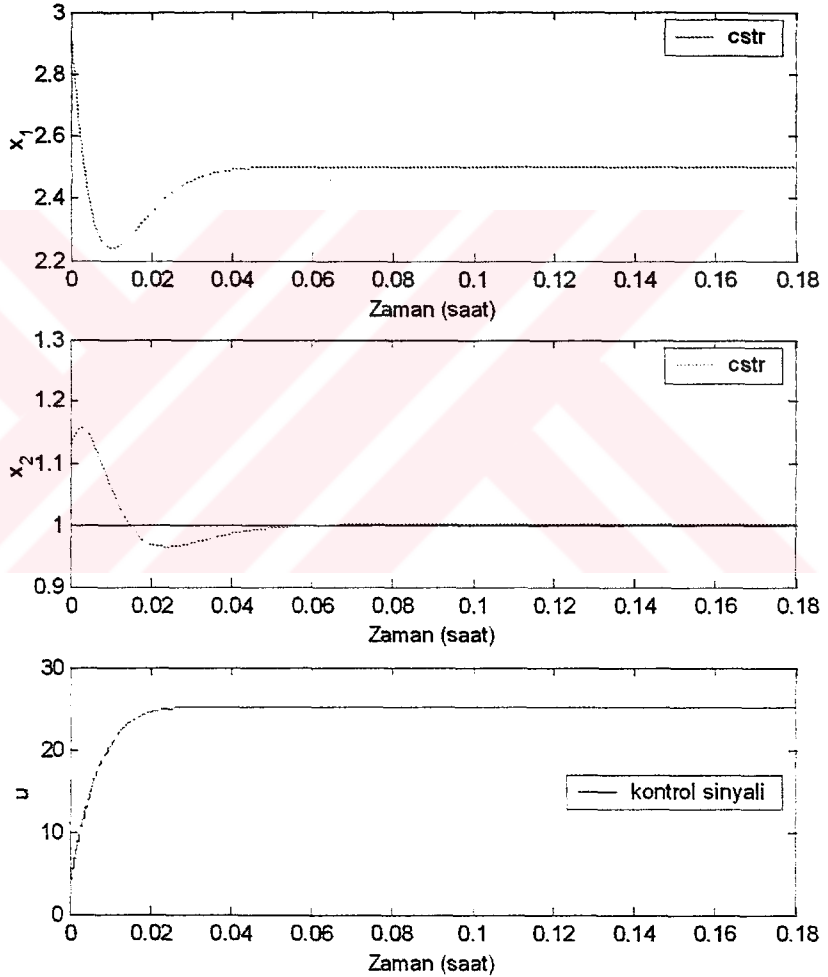
Yapılan tanılamada, test ver seti için “ x_1 ve x_2 ” süreç ve süreç modeli arasındaki fark görüldüğü üzere çok küçüktür. Burada YSA'nın eğitiminde kullanılan örnek sayısı ve uygun algoritma seçimi ile daha iyi sonuçlara ulaşılabilir (Şekil 5.16).



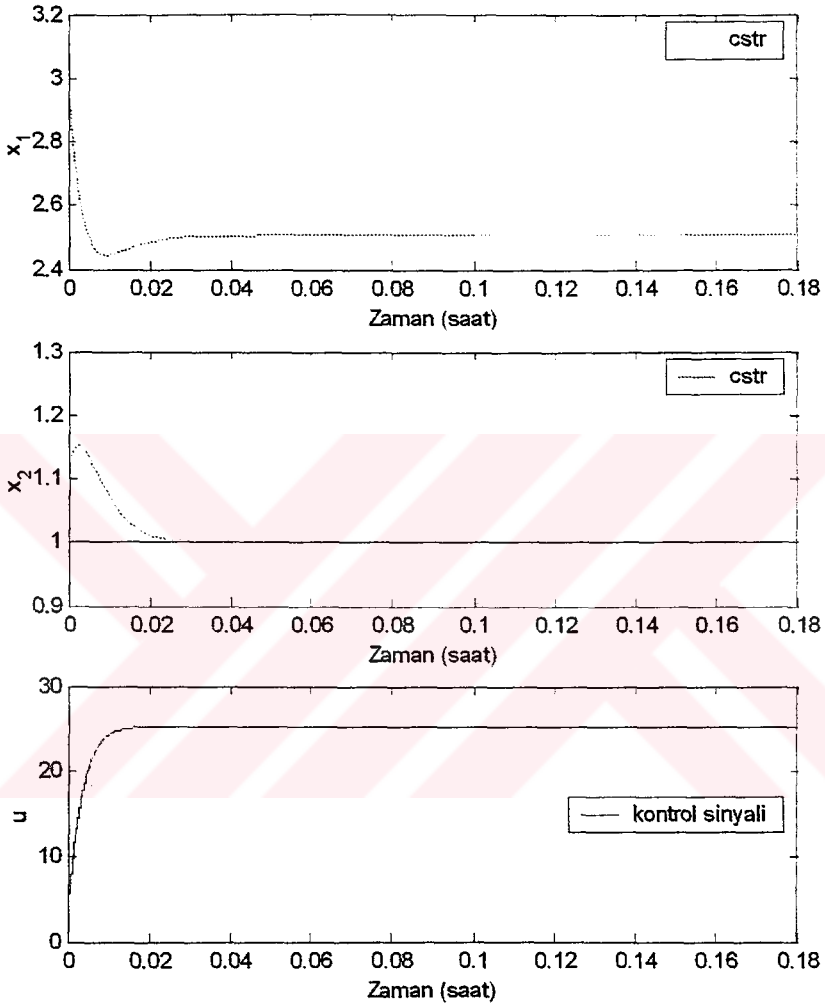
Şekil 5.16 CSTR için YSA model ve süreç sonuçları “ x_1 ve x_2 ”

Bazı durumlarda PÖ'de ek bir ifade olan, giriş vektörünün sapmalarını ideal değerine sınırlandıran $\sum \|u(t+i/t) - u_0\|_s^2$ terimi eklenebilir. Bu işlem genellikle kontrol edilecek değişkenlerden daha

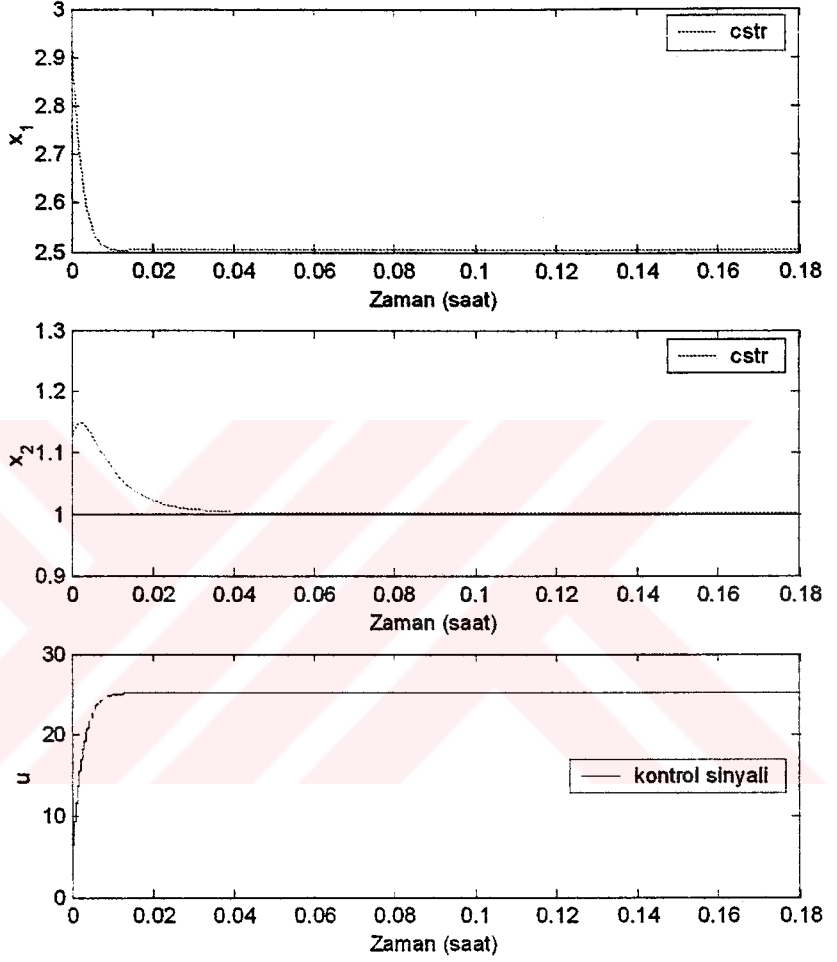
fazla giriş olduğunda yapılır. Basit PÖ formunda bu terimi içermez (Maciejowski, 2002). Eşitlik 5.10'daki $\gamma_3[25 - u(t+i)]^2$ ifadesi kaldırılarak, sadece durum değişkenlerinden oluşan PÖ ile istenilen değerler elde edilebilecektir (Şekil 5.20).



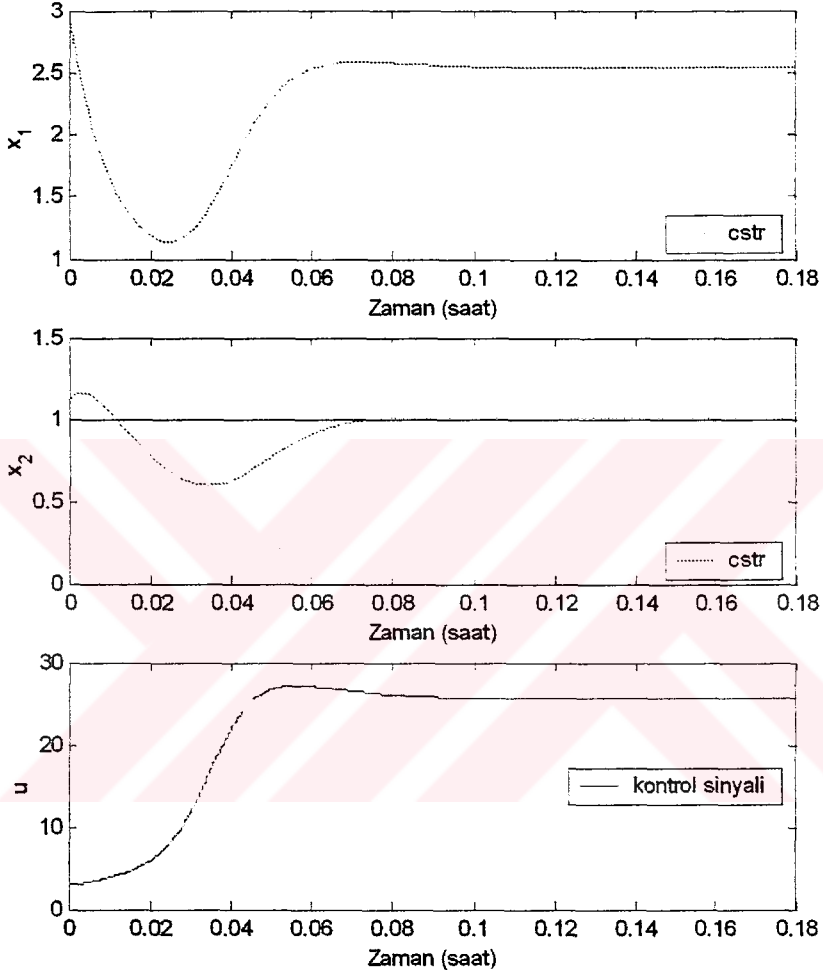
Şekil 5.17 CSTR için durum değişkenleri " x_1 ve x_2 " ve " u " ($N_2 = 3$)



Şekil 5.18 CSTR için durum değişkenleri " x_1 ve x_2 " ve " u " ($N_2 = 6$)



Şekil 5.19 CSTR için durum değişkenleri " x_1 ve x_2 " ve " u " ($N_2 = 8$)



Şekil 5.20 CSTR, PÖ'de $(25-u)^2$ yok iken " x_1 ve x_2 " ve " u " ($N_2 = 6$)

6. SONUÇ

Bu tez çalışmasında, YSA temelli model öngörülü kontrol (YSAMÖK) yapısı geliştirilerek, doğrusal ve doğrusal olmayan sistemlerin kontrolünde kullanılmıştır.

Son yıllarda, öngörülü kontrol yapıları, çok değişkenli kontrol uygulamalarında bir endüstri standardı haline gelmiştir. Ancak, doğrusal olmayan süreçler için kontrolör, doğrusal dinamik modeller kullanılarak tasarlanmaktadır. Bu modeller sürecin tek bir denetim noktasında işletildiği durumlar için kabul edilir performans göstermelerine rağmen farklı denetim noktalarında, kontrol performansı bozulmaktadır. Literatürde yapılan çalışmalarda modellerin doğrusallaştırıldığı görülmektedir (Clarke et al., 1987; Piche et al., 2000; Rau and Schröder, 2002).

Bu tez çalışmasında geliştirilen YSAMÖK yaklaşımı ile doğrusallaştırma gibi sınırlamaları ortadan kaldıran, uyarlanma ve öğrenme özelliklerine sahip model öngörülü kontrol yapısı oluşturulmuştur.

YSA'nın doğrusal ve özellikle doğrusal olmayan süreçlerde eşleme, paralellik, öğrenme ve uyarlanma gibi özellikleri vurgulanmış. Geliştirilen YSAMÖK'ün süreç modellemesi seri-paralel tanılama modeli ile yapılmıştır.

YSA temelli MÖK'de, YSA'ların doğrusal olmayan eşleme, paralellik, öğrenme ve uyarlanma gibi özellikleri model öngörülü kontrol yapısına katılmıştır.

Bu çalışmada ilk aşamada model öngörülü kontrol tasarımına alt yapı oluşturması amacıyla DC armatürlü motor ve eş ısıt reaktör (CSTR) sistemleri için Hamiltonian yaklaşımı ile optimal kontrol sistemleri tasarlanmıştır. Optimal kontrolör performansları, Henson et al. (1997)'de

verilen model öngörülü kontrol yöntemleri ile elde edilmiş olan cevaplar ile karşılaştırılmıştır. Sonuçta, benzer sonuçlar elde edilmiştir (Lewis, 1992; Henson and Seborg, 1997).

Çalışmanın ikinci aşamasında çeşitli öngörülü kontrol algoritmaları analiz ederek, uygulamaları gerçekleştirilmiştir (Bkz. Bölüm 3). Bu algoritmalarda genelde doğrusal modeller kullanılmıştır. Doğrusal olmayan modellerin doğrusallaştırıldığı ve kontrol sinyalini üretmek için kullanılan matris boyutlarının büyüdüğü görülmüştür. Matris boyutları öngörü ufku ve kontrol ufkuna bağlı olduğundan, ufukların artması algoritmanın performansını olumsuz yönde etkilediği görülmüştür (Clarke et al., 1987; Brosilow and Joseph, 2002).

Üçüncü aşamada YSA temelli MÖK yapısı geliştirilmiştir. Geliştirilen YSAMÖK yapısında sürecin yapay sinir ağı ile modeli oluşturulmuştur. Giriş sinyali için kullanılan kuadratik maliyet fonksiyonu, durum değişkenlerini istenilen yörüngeye ulaştıracak kontrol sinyalini üretme performansı hız yönünden iyidir. YSAMÖK algoritması, modelden elde edilen veriler sayesinde klasik öngörülü kontroldeki matris bağımlılığını ortadan kaldırmaktadır. Algoritma hızı açısından literatürde geçen DMA algoritması ile YSAMÖK karşılaştırıldığında performansının daha iyi olduğu görülmüştür.

Çalışmanın son kısmında, optimal kontrol ve klasik öngörülü kontrol yöntemleri ile yapılan karşılaştırmalar ile tasarlanan kontrolörün etkinliği gösterilmiştir.

Bundan sonraki aşamada, geliştirilen YSA temelli MÖK yapısı FPGA ve DSP gibi tasarım araçları kullanılıp, gerçekleştirilerek endüstriyel süreçlerin kontrolünde kullanımı amaçlanmaktadır.

KAYNAKLAR DİZİNİ

Becerikli Y., 1998, Nöro – Optimal Kontrol, Doktora Tezi, Sakarya Üniversitesi FBE, 196s (yayınlanmamış).

Brosilow C. and Joseph B., 2002, Techniques of Model – based Control, Prentice Hall, 680.

Camacho, E. F., Bordons, C., 1999, Model Predictive Control, Springer Verlag, London Berlin New York.

Chang X. and Krogh B.H., 2001, Stability – Constrained Model Predictive Control, IEEE Transactions on Automatic Control, Vol.146, No.11, November, 1816–1821.

Clarke D.W., Mohtadi, C., Tuffs, P.S., 1987, Generalized Predictive Control – Part I Basic Algorithm, Automatica Vol. in GB, 23, No.2, 137-148.

Henson M.A. and Seborg D.E., 1997, Nonlinear Process Control, Prentice Hall PTR, New Jersey, 423.

Hines J.W., 1997, MATLAB Supplement to Fuzzy and Neural Approaches in Engineering, John Wiley & Sons, Inc., 206.

Kajima A. and Ishijima S., 1999, LQ Preview Synthesis: Optimal Control and Worst Case Analysis, IEEE Transactions on Automatic Control Vol. 44 No.2 February, 352-357.

Kothare S.L. de O. and Morari M., 2000, Contractive Model Predictive Control for Constrained Nonlinear Systems, IEEE Transactions On Automatic Control, June, Vol. 45, No.6, 1053-1071.

Leung C.S. and Chan L.W., 2003, Dual extended Kalman filtering in recurrent neural networks, Elsevier science, Neural Networks 16 (2003), 223-239.

Lewis, F., 1992, *Applied Optimal Control & Estimation*, Prentice-Hall, Inc., 624.

Low K.S. and Zhuang H., 2000, Robust Model Predictive Control and Observer for Direct Drive Applications, *IEEE Transactions on Power Electronics*, Vol.15, No.6, November, 1018–1029.

Maciejowski J.M., 2002, *Predictive Control with Constrained*, Prentice Hall, 332.

Mollow S., Boom T.D., Cuesta F., Ollero A. and Babuska R., 2002, Robust Stability Constraints for Fuzzy Model Predictive Control, *IEEE Transactions on Fuzzy Systems*, Vol.10, No.1, February, 50–65.

Nguyen D.H. and Widrow B., 1990, Neural networks for self-learning control systems, *IEEE Control System Magazine*, Vol.10, April, 18-23.

Normey-Rico J.E. and Camacho, E.F., 2000, Multivariable Generalised predictive Controller Based on The Smith Predictor, *IEE Proc. Control Theory Appl.* Vol.147 No.5, September, 538-547.

Piche S., Sayyar-Rodsari B., Johson D. and Gerules M., 2000, Nonlinear Model predictive Control Using Neural Networks, *IEEE Control System Magazine*, June, 53–62.

Primbs J.A. and Nevistic V., 2000, A New Approach to Stability Analysis for Constrained Finite Receding Horizon Control without End Constrained, *IEEE Transactions on Automatic Control* Vol. 45 No.8 August, 1507-1512.

Principe J.C., Euliano N.R. and Lefebvre W.C., 2000, *Neural and Adaptive Systems*, John Wiley & Sons, Inc., 656.

Rau M. and Scröder D., 2002, Model Predictive Controller with Nonlinear State Space Models, IEEE, AMC 2002 - Maribar, Slovenia, 136-141.

Rawlings J., 2000, Tutorial Overview of Predictive Control IEEE Control System Magazine, June, 38-52.

Sarioğlu K., 1985, Optimal Control Theory, ITU yayınları, İstanbul., 174.

Scales L.E., 1985, Introduction to Non-Linear Optimization, Springer-Verlag New York Inc.

Sjoberg J., Zhang Q., Ljung L., Benveniste A., Delyon B., Glorennec P.Y., Hjalmarsson H. and Juditsky A., 1995, Nonlinear Black-box Modeling in System Identification, a Unified Overview, Automatica, Vol. 31, No.12, 1691–1724.

Song B.J. and Koivo A.J., 1999, Nonlinear Predictive Control with Application to Manipulator with Flexible Forearm, IEEE Transactions on Industrial Electronics, October Vol. 46, No.5, 923-932.

Zurada J.M., 1992, Artificial Neural Systems, West Publishing Company, 675.

ÖZGEÇMİŞ

Savaş Şahin, Türkiye Cumhuriyeti vatandaşı olup, 21 Kasım 1974 tarihinde Çorum'da doğdu. İlk ve orta öğrenimini İzmir'de tamamladı. Lise öğrenimini İzmir Çınarlı Teknik Lisesi Elektronik Bölümünde 1992 yılında bitirdi. Lisans eğitimi; Kocaeli Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Bölümünden 1996 yılında mezun oldu. Askerlik görevini asteğmen olarak 3. Hava İkmal Bakım Merkezi Komutanlığında 1998 yılında tamamladı. Halen, Ege Üniversitesi FBE'de Mühendislik Fakültesi Elektronik ana bilim dalında yüksek lisans çalışması yapmaktadır. Aynı zamanda, Ege Üniversitesi Ege Meslek Yüksekokulunda öğretim görevlisi olarak çalışmaktadır.