

**BULANIK KONTROLÖR İLE YENİ BİR GLOBAL
ENİYİLEME YÖNTEMİ**

100797

DOKTORA TEZİ
Y. Müh. Berk ÜSTÜNDAĞ
504940003012

100797

Tezin Enstitüye Verildiği Tarih : 21 Eylül 1999

Tezin Savunulduğu Tarih : 23 Şubat 2000

Tez Danışmanı :

Prof.Dr.Atilla BİR

Diğer Jüri Üyeleri

Prof.Dr. Eşref ADALI (İ.T.Ü.)

Prof.Dr. Ferit KONAR (S.Ü.)

Doç.Dr. İbrahim EKSİN(İ.T.Ü.)

Doç.Dr. Levent AKIN (B.Ü.)

[Handwritten signatures of Prof. Dr. Atilla Bir, Prof. Dr. Eşref Adalı, Prof. Dr. Ferit Konar, Doç. Dr. İbrahim Eksin, and Doç. Dr. Levent Akın]

ŞUBAT 2000

ÖNSÖZ

Burada sunulan çalışmanın, sayısal eniyileme problemleri ile kapalı çevrim kontrol kuramı arasında karşılıklı bir ilişki kurulması bakımından yeni bir bakış açısı getirdiğini düşünüyorum, ve konu ile ilgili kişilere faydalı olmasını diliyorum.

Bu doktora tezinin hazırlanmasında ve yöntemin geliştirilmesinde yardımcı olan tez danışmanım sayın Prof.Dr.Atilla BİR'e ve sayın Doç.Dr.İbrahim EKSİN'e teşekkür ederim.

Eylül 1999

Berk ÜSTÜNDAĞ

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	x
ÖZET	xii
SUMMARY	xvi
1. GİRİŞ	1
1.1. Global Eniyileme Problemi	1
1.2. Global Eniyilemenin Önemi	2
1.3. Bölümlerin Açıklaması	2
2. GLOBAL ENİYİLEME PROBLEMİ VE ÇÖZÜM YÖNTEMLERİ	4
2.1. Global Eniyileme Probleminin Tanımı	4
2.2. Global Eniyileme Problemlerinin Sınıflandırılması	5
2.3. Global Eniyileme Yöntemleri	6
2.3.1. Yerel arama üzerine kurulu yöntemler	6
2.3.2. Arama bölgesini örten yöntemler (set covering methods)	8
2.3.3. İndirgeme ve alan daraltma kullanan yöntemler	10
2.3.4. Stokastik modellemeye dayalı yöntemler.	11
2.3.5. Raslantısal örneklemeye dayalı yöntemler	13
2.3.6. Evrimsel arama yöntemleri	14
2.4. Global Eniyileme Yöntemlerinin Değerlendirilmesi	15
2.4.1. Önbilgi gereksinimi	15
2.4.2. Çözüm aralığı ve kısıtlamalar	16
2.4.3. Çözünürlük	16
2.4.4. Etkinlik	18
2.4.5. Kararlılık	18
2.4.6. Sonlandırma kriteri	19
3. KONTROL KURAMININ ENİYİLEMeye UYGULANMASI	20
3.1 Kapalı Çevrim Kontrol Tekniği ile Kök Arama	21
3.1.1. Tek değişkenli bir fonksiyonun kökünün bulunması	25
3.1.2. İstenilen bir doğrultuda kök aranması	31
3.2. Kapalı Çevrim Kontrol Tekniği ile Yerel Eniyileme	33
3.2.1. Tek değişkenli fonksiyonların yerel eniyilenmesi	33

3.2.2. Çok deęişkenli fonksiyonların yerel eniyilenmesi	40
4. PI TİPİ KONTROLÖR İLE SAYISAL ARAMADA KARŞILAŞILAN SORUNLAR VE BULANIK KONTROLÖR KULLANIMI	46
4.1. Sayısal Aramada PI Tipi Kontrolör Kullanımında Karşılaşılan Sorunlar	46
4.2. Sayısal Aramada Lineer Kontrolör Kullanımında Çıkan Sorunların Sebebi ve Öneriler	48
4.3. Sayısal Aramada Bulanık Kontrolör Kullanımı	53
4.3.1. Bulanık kontrolör ile kök aranması	60
5. BULANIK KONTROLÖR İLE GLOBAL ENİYİLEME	66
5.1. Bulanık Kontrolörle Arama Yapılarak Global Eniyi Noktanın Bulunması	66
5.2. Çok Deęişkenli Fonksiyonların Global Eniyilenmesi	71
5.3. Standart Test Fonksiyonları	79
5.4 Paralel İşletime Uygunluk	81
6. SONUÇLAR VE TARTIŞMA	84
KAYNAKLAR	86
EKLER	89
ÖZGEÇMİŞ	98

KISALTMALAR

AÖF	: Amaç ölçüt fonksiyonu
AS	: Alt sınır
NB	: Negatif-Büyük
NO	: Negatif-Orta
NK	: Negatif-Küçük
PB	: Pozitif-Büyük
PI	: Oransal-İntegral (Proportional-Integral)
PO	: Pozitif-Orta
PK	: Pozitif-Küçük
REF	: Referans
US	: Üst sınır



TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. IEEE754'e göre 32bit ve 64bit sayı tanımları.....	17
Tablo 3.1. PI kontrolörle $f(x)=x^2-4=r=0$ çözümünde hata işaretine göre kontrolörün aldığı çıkış değerleri.....	28
Tablo 3.2. $x_0=0.8$ için $x^2-4=4$ çözümünde hata işareti ve PI kontrolör çıkışı.....	29
Tablo 3.3. $x_0=0$, $f(x)=\sin(x)+0.3e^{-x}$ için $f'(x)=0$ 'ın (+) işaret yönü çözümünde hata işareti ve PI tipi kontrolör çıkışı.....	37
Tablo 3.4. $x_0=0$, $f(x)=\sin(x)+0.3e^{-x}$ için $f'(x)=0$ 'ın (-) işaret yönü çözümünde hata işareti ve PI tipi kontrolör çıkışı.....	39
Tablo 3.5. $f(x)=-(x_1-3)^2-(x_2+1)^2+4$, $K_i=0.4$, $K_p=0.2$ (x_{10},x_{20})=(0,0) için $f'(x_1,x_2)=0$ çözümünde kontrolör çıkışının ve fonsiyonun aldığı değerler.....	45
Tablo 4.1. $K_p=0.05$, $3x^2-9=0$ denkleminin PI tipi kontrolörle çözümünde çeşitli K_i integral kazanç değerleri için sonuca erişimdeki döngü sayıları.....	47
Tablo 5.1. Global enküçük araması yapılan bazı test fonksiyonları ve literatürde yer alan bazı algoritmaların sonuca erişimdeki değerlendirme sayıları.....	79
Tablo 5.2. Literatüdeki bazı global arama algoritmalarının iki değişkenli test fonksiyonlarında $E_m=0.00001$ hassasiyetle yapılan arama sayılarının karşılaştırılmaları.....	80

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 3.1 : Sayısal arama probleminin bir otomatik kontrol sistemi olarak ğerlendirilmesi.....	21
Şekil 3.2 : Zamandan ayırık bir kontrol sisteminde poligonal integrasyon.....	23
Şekil 3.3 : PI tipi bir kontrolörle $f(\cdot)$ fonksiyonunun köklerinin bulunması...	24
Şekil 3.4 : $f(x)=x^2-4$ fonksiyonunun $(-5,+5)$ aralığında değişimi.....	26
Şekil 3.5 : $f(x)=x^2-4$, $r=0$, $x_0=0.8$ için a) $f(x)=r$ çözümünde kontrol işaretinin (çözüm) değişimi b) fonksiyon çıkışı c) hata işaretinin değişimi.....	27
Şekil 3.6 : $f(x)=x^2-4$, $r=4$, $x_0=0.8$ için a) $f(x)=r$ çözümünde kontrol işaretinin (çözüm) değişimi b) fonksiyon çıkışı c) hata işaretinin değişimi.....	29
Şekil 3.7 : $f(x)=-0.2x^2+4+4\sin(x)$ fonksiyonunun $[-6,+6]$ aralığında değişimi.....	30
Şekil 3.8 : $f(x)=-0.2x^2+4+\sin(4x)$, $r=0$, $x_0=0$ için a) $f(x)=r$ çözümünde kontrol işaretinin (çözüm) değişimi b) fonksiyon çıkışı c) hata işaretinin değişimi.....	30
Şekil 3.9 : İstenilen işaret yönünde arama için $d(k)$ doğrultu çarpanı eklenmesi.....	31
Şekil 3.10 : $f(x)=0.2x^2-4-4\sin(4x)$ fonksiyonunun $(-6,+6)$ aralığında değişimi	32
Şekil 3.11 : $f(x)=0.2x^2-4-\sin(4x)$, $r=0$, $x_0=0$ için a) $f(x)=r$ çözümünde kontrol işaretinin (çözüm) değişimi b) fonksiyon çıkışı c) hata işaretinin değişimi.....	33
Şekil 3.12 : Yerel eniyileme için geri besleme yolunda iyileştirme yapılması	34
Şekil 3.13 : Tek değişkenli fonksiyonların yerel eniyileme için önerilen kapalı çevrim kontrol düzeni.....	35
Şekil 3.14 : $f(x)=4\sin(x)+0.3e^{-x}$ fonksiyonunun $(-5,+5)$ aralığında değişimi...	36
Şekil 3.15 : $f(x)=4\sin(x)+0.3e^{-x}$, $K_i=0.21$, $K_p=0.1$, $x_0=0$ için $f'(x)=0$ çözümünde a) Kontrol işareti b) $f'(x)$ aldığı değerler c) Hata işareti	37
Şekil 3.16 : $f(x)=4\sin(x)+0.3e^{-x}$, $K_i=0.21$, $K_p=0.1$, $x_0=0$ için $f'(x)=0$ 'ın (-) işaret yönündeki çözümünde a) Kontrol işareti b) $f'(x)$ aldığı değerler c) Hata işareti.....	39
Şekil 3.17 : Kapalı çevrim kontrol kuramı yaklaşımı ile iki değişkenli bir fonksiyonun yerel ekstremum değerlerinin bulunması.....	42
Şekil 3.18 : $f(x)=-(x_1-3)^2-(x_2+1)^2+4$ fonksiyonunun $[(-2,-19),(8,5)]$ aralığında değişimi	43
Şekil 3.19 : $f(x)=-(x_1-3)^2-(x_2+1)^2+4$, $K_i=0.4$, $K_p=0.2$ (x_{10},x_{20})=(0,0) için $f'(x_1,x_2)=0$ çözümünde a) Kontrolör çıkışı u_1 bileşeni b) Kontrolör çıkışı u_2 bileşeni c) Fonksiyonun arama sırasında aldığı değerler $f(u_1(k),u_2(k))$	44

Şekil 4.1	: $f_1(x)=3x^2-9$ ve $f_2(x)=x^2-3$ fonksiyonlarının kök civarında doğrusallaştırılması : $f_1'(1.732)=10.39$, $f_2'(1.732)=3.46$	47
Şekil 4.2	: Tablo 4.1'de görülen parametrelere bağlı olarak, $f(x)=3x^2-9$, $r=0$ için $f(x)=r$ denkleminin çözümünde kontrolör çıkışının değişimi....	48
Şekil 4.3	: Lipchitz değeri bilinen bir fonksiyonun köklerinin integral kontrolörle aranması.....	49
Şekil 4.4	: Ayrık poligonal integral kontrolör ile kök aranması.....	51
Şekil 4.5	: Bulanık kontrolörün sayısal aramaya uyarlanması.....	53
Şekil 4.6	: Bulanık kontrolör blok diyagramı.....	55
Şekil 4.7	: Hata , hatanın değişimi ve çıkış işareti için üyelik fonksiyonları..	56
Şekil 4.8	: Bulanık kontrolör çıkışına ilişkin kontrol yüzeyi.....	58
Şekil 4.9	: a) $f(x)=-x^2+3$ b) $f(x)=-1.5x^2+4.5$ c) $f(x)=-3x^2+9$ fonksiyonlarının $(-5,+5)$ aralığında değişimi.....	61
Şekil 4.10	: a) $-x^2+3=0$ b) $-1.5x^2+4.5=0$ c) $-3x^2+9=0$ çözümlerinde kontrol işaretinin değişimi ($Sce=0.9$, $Scce=0.05$, $Scu=0.3$).....	62
Şekil 4.11	: $f(x)=-x^2+2+3\sin(11x)+2\cos(5x)$ fonksiyonunun $(-5,+5)$ aralığında değişimi.....	63
Şekil 4.12	: $-x^2+2+3\sin(11x)+2\cos(5x)=0$ denkleminin çözümünde a) $x_0=u(0)=0$, b) $x_0=u(0)=0.37$, c) $x_0=u(0)=0.6$, d) $x_0=u(0)=0.9$, e) $x_0=u(0)=1.1$, f) $x_0=u(0)=1.48$, g) $x_0=u(0)=2.39$ için kontrol işaretinin değişimi ($Sce=0.3$, $Scce=0.05$, $Scu=0.3$).....	64
Şekil 5.1	: Tek değişkenli amaç ölçüt fonksiyonları için global eniyileme algoritması.....	67
Şekil 5.2	: Tek değişken için global enbüyük noktanın bulunması.....	68
Şekil 5.3	: $(-10,10)$ aralığında $f(x)=-0.2x^2+5+2\sin(2x)$ fonksiyonu.....	69
Şekil 5.4	: $(-10,10)$ aralığında $f(x)=-0.2x^2+5+2\sin(2x)$ fonksiyonunun global enbüyüğünün hesaplanmasında kontrolör çıkışının değişimi.....	69
Şekil 5.5	: $(-10,10)$ aralığında $f(x)=-0.5x^2+5+2\sin(7x)$ fonksiyonu.....	70
Şekil 5.6	: $f(x)=-0.5x^2+5+2\sin(7x)$ fonksiyonunun $(-4,10)$ aralığında global enbüyüğünün hesaplanmasında kontrolör çıkışının değişimi.....	71
Şekil 5.7	: İki değişkenli $f(x_1,x_2)$ fonksiyonu için bulanık mantıkla global enbüyük arama algoritması.....	72
Şekil 5.8	: a) $f(x,y) = x e^{-x^2-y^2}$ ifadesinin $[(0,+2),(-2,+2)]$ aralığında değişimi b) y yönünde tek değişkenli arama yapılırken x yönünde atılan adımların canlandırılması.....	74
Şekil 5.9	: $f(x,y)=-0.2x^2+2\sin(2x)-y^2+5$ fonksiyonunun $(-10,+10),(-5,5)]$ aralığında değişimi	76
Şekil 5.10	: a) $y=-2$ için x yönünde global enbüyüğün aranması b) (a)'daki arama sonucu referans alınarak $y=-2+h=1.9999$ için x yönünde arama.....	76
Şekil 5.11	: $f(x,y)=-0.2x^2+2\sin(2x)-0.2y^2+2\sin(2y)+5$ fonksiyonunun $(-10,-10),(10,10)]$ aralığında değişimi.....	77
Şekil 5.12	: $f(x,y)=-0.2x^2+2\sin(2x)-0.2y^2+2\sin(2y)+5$ fonksiyonunun eşyükselti çizimi.....	78
Şekil 5.13	: Çoklu işletimde kölelerin ana makinaya yıldız bağlantı durumu..	83
Şekil C.1	: $2.7 \leq x \leq 7.5$ aralığında $f(x) = \sin x + \sin(10x/3) + \ln x - 0.84x$ fonksiyonu.....	92
Şekil C.2	: $3.1 \leq x \leq 20.4$ aralığında $f(x) = \sin x + \sin(2x/3)$ fonksiyonu	93

Şekil C.3	: $-10 \leq x \leq 10$ aralığında $f(x) = -\sum_{i=1}^5 \sin((i+1)x + i)$ fonksiyonu.....	94
Şekil C.4	: a) $-5 \leq x \leq 5, -5 \leq y \leq 5$ aralığında $f(x,y) = 4x^2 - 2.1x^4 + x^6/3 + xy - 4y^2 + 4y^4$	95
Şekil C.5	: a) $-5 \leq x \leq 5, -5 \leq y \leq 5$ aralığında Goldstein-Price test fonksiyonu b) Eşyükselti eğrisi.....	96
Şekil C.6	: a) $-5 \leq x \leq 10, 0 \leq y \leq 15$ aralığında iki değişkenli Branin test fonksiyonu b) Eşyükselti eğrisi	97



SEMBOL LİSTESİ

a	: Çözüm aralığı alt sınırı
b	: Çözüm aralığı üst sınırı
C	: Sürekli fonksiyonlar kümesi
C^1	: Birinci türevi sürekli fonksiyonlar kümesi
C^2	: İkinci türevi sürekli fonksiyonlar kümesi
C^k	: k . türevi sürekli fonksiyonlar kümesi
ce	: Hatanın değişimi
CE	: Ölçeklendirilmiş hata değişimi
d	: Doğrultu çarpanı
du	: Bulanık kontrolör çıkış işareti (değişim miktarı)
e	: Hata işareti
E	: Ölçeklendirilmiş hata
E_m	: Hata sınırı
f_m	: Değerlendirilen fonksiyon değerlerinden en büyüğü
f_l	: Değerlendirilen fonksiyon değerlerinden en küçüğü
h	: Sayısal yaklaşık türev alma işleminde küçük değerli bir sabit sayı
H	: Hessian
k	: Özyineleme sayısı
K_i	: İntegral katsayısı
K_p	: Oransal kontrol katsayısı
L	: Lipchitz katsayısı
n	: Fonksiyonun derecesi
N	: Hesaplama yapılan nokta sayısı
P	: Olasılık
r	: Referans işaret
S_{ce}	: Hata ölçeklendirme katsayısı
S_{cec}	: Hatanın değişimini ölçeklendirme katsayısı
S_{cu}	: Çıkış ölçeklendirme katsayısı
s_k	: k .özyinelemedeki arama yönü
u	: Kontrolör çıkış işareti
u_{sat+}	: Kontrolör çıkışının üst doyma sınırı
u_{sat-}	: Kontrolör çıkışının alt doyma sınırı
x^*	: AÖF'nin bir ekstremumunu veren nokta
X	: Çalışmaya uygun bölge (Feasible region)
X^*	: AÖF'yi en küçük yapan noktalar kümesi
$x(k)$	: k .özyinelemede kontrolör iç değişken değeri
y	: Sistem çıkış işareti
δ	: Çözüm hassasiyeti
Δ	: Sayısal hesaplamada öteleme miktarı
ε	: Sıfırdan büyük, sıfıra yakın, çok pozitif reel sayı
$\in$	: Eleman
$\notin$	: Eleman değil

$\phi(\mathbf{x}, \omega)$	: Rastlantı fonksiyonu
γ_k	: k.özyinelemedeki adım uzunluğu
η	: Etkinlik
μ	: Üyelik derecesi
∇	: Birinci derece vektörel türev alıcı (gradyen)
∇^2	: İkinci derece vektörel türev alıcı
Ω	: Olasılık uzayı
ω	: Açısal frekans
Σ_k	: Rastlantı fonksiyonu
$\forall$	: Her
$\wedge$	: Mantıksal ve bağlacı
$\cup$	: Birleşim
$\cap$	: Kesişim
$\neg$	: Tümleme



BULANIK KONTROLÖR İLE YENİ BİR GLOBAL ENİYİLEME YÖNTEMİ

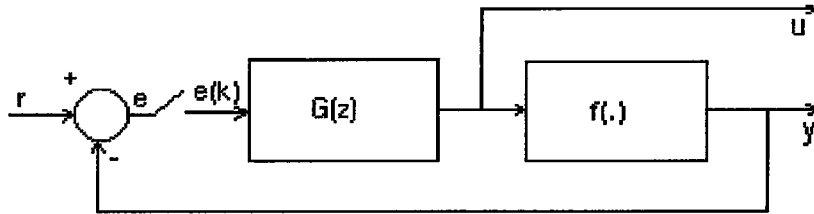
ÖZET

Global enküçültme , $X \subset \mathbb{R}^n$ kompakt bir küme olmak üzere, $f: \mathbb{R}^n \rightarrow \mathbb{R}^1$ amaç ölçüt fonksiyonu ,

$$f^* = \min f(x) \quad , \quad x \in X \quad (1)$$

şeklinde tanımlanan bir f^* değerini bulmayı amaçlar . $f(x)$ fonksiyonunu enküçüten x değeri, ayrıca $-f(x)$ fonksiyonunu enbüyüteceğinden, bu tanım global enbüyük için de geçerlidir ve genel olarak global eniyileme olarak adlandırılır. Amaç ölçüt fonksiyonu f 'nin süreksizliliği, birden fazla ekstremumunun bulunması, kullanılacak eniyileme tekniğini kısıtlar. Örneğin, süreksiz bir amaç ölçüt fonksiyonunun eniyilenmesinde, türev ifadesini kullanan yöntemlerde sorunlarla karşılaşılır.

Şekil 1'de birim geribeslemeli bir ayrık kontrol sisteminin blok yapısı görülmektedir. $f(.)$ diferansiyel denklem takımı yerine bir $f(x)$ fonksiyonu yerleştirilmesi halinde, kontrol kuramı kullanılarak, (1) ile tanımlanan global eniyileme probleminin çözümüne, aynı blok diyagramı ile yaklaşılabılır.



Şekil 1 Sayısal arama probleminin bir otomatik kontrol sistemi olarak görülmesi.

Kontrol kuramı gereği uygun seçilmiş bir G kontrolörü, sürekli halde e hata işaretini sıfıra götürür. Diğer bir bakış açısıyla, k adım sayısı olmak üzere,

$$k \rightarrow \infty \text{ için } e(k) \rightarrow 0, f(u(k)) \rightarrow r \quad (2)$$

ya da, $*$ eniyi çözümü ifade etmek üzere,

$$\left| f(u) - r \right| \xrightarrow{k \rightarrow \infty} 0 \Rightarrow u^* = x^*, f(x^*) = r \quad (3)$$

kullanılan kapalı çevrimli sistemdeki tek bellek elemanı bulanık kontrolör çıkışındaki integratördür. Sayısal çözüm amaçlı bu sistem, sınırlı giriş için sınırlı çıkış üreteceğinden, hatalı kontrolör parametresi seçimi, yalnızca aranan kök civarında sınırlı genlikli salınıma yol açmaktadır. Bulanık kontrolör karar tablosu doğru seçilmiş ise, giriş kazanç parametrelerinde (Sce, Scce) ideal değerden sapma olması durumunda bile sistem, en fazla, aranan kök civarında kararlı bir salınıma girer. Bulanık kontrolör çıkışı u , her iki işaret yönünde de doymaya girdiğinden sistem en azından Lyapunov anlamında kararlıdır.

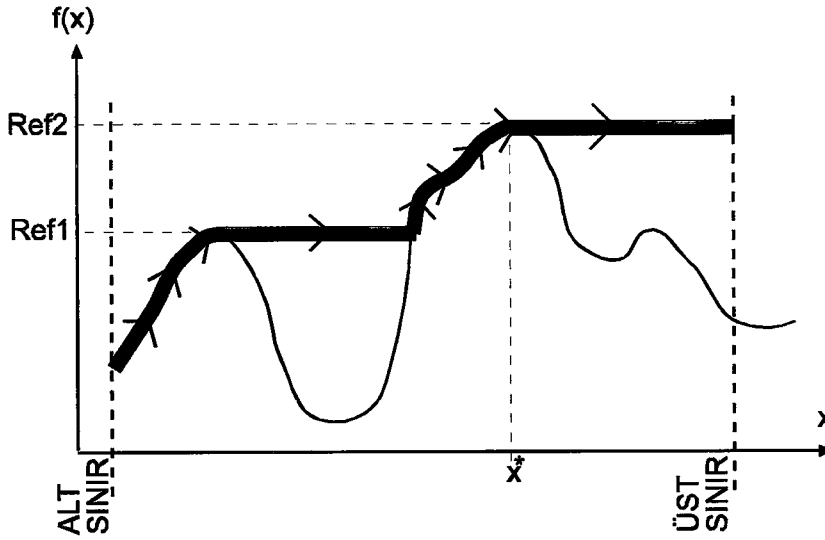
Arama yapılacak üst sınır US ; başlama noktası AS olsun . (AS,US) aralığında arama yapabilmek için $u(0)=AS$ olmalıdır. Yapılan bu atama sayısal arama işlemine istenen bir noktadan başlanmasını sağlar.

Unimodal bir fonksiyonun eniyilenmesi ya da köklerinin bulunmasına ilişkin çok sayıda yöntem bulunmaktadır. Bilindiği üzere yerel kök arama veya eniyileme yöntemleri unimodal olmayan fonksiyonlarda başarısız olabilmektedir. Bulanık kontrolör kullanılan bu yeni yöntemde, verilen bir başlangıç noktasından itibaren istenilen doğrultudaki bir köke erişilirken, sadece daha önceki sayısal denemeler ve bulanık karar tablosu referans alındığından, fonksiyonun karmaşıklığı çözüme ulaşma başarısını etkilemez. Bu özellik global eniyileme probleminin çözümünde kullanılmaktadır.

Bulanık kontrolör ile global eniyileme için çeşitli yaklaşımlar mümkündür. Burada önerilen eniyileme algoritması, bulanık kontrolörlü kapalı çevrim kontrol düzeni ile çözüm istenen aralığın taranması üzerinedir. Tarama işlemi, alt sınırdan üst sınıra, üst sınırdan alt sınıra, ya da çok başlangıç noktalı yapılabilir. Eniyileme algoritması, alt sınırı başlangıç noktası kabul etmekte ve verilen bir $f(.)$ fonksiyonunun enbüyük değerini aramaktadır. Aramaya başlanan noktada fonksiyon artış yönünde ise $r=0$ için $f'(u)=0$ çözümü sağa doğru aranır ve bulunan ilk kök yeni global enbüyük noktası olarak kabul edilir. Aramaya başlanan noktada fonksiyon azalma yönünde ise, $r=f(u)$ için sağa doğru bulanık kontrolör ile kök aranır ve bulunan ilk kökten itibaren $r=0$ için $f'(u)=0$ çözümü elde edilerek varılan yeni nokta global enbüyük kabul edilir. Her iki halde de r =yeni global enbüyük, alınarak $f(u)=r$ çözümü ile tekrarlı olarak işleme devam edilir. Arama sırasında kontrol işareti $u>US$ koşulu oluşursa varılmış olan en son $f'(u)=0$ çözümü global enbüyüğü verecektir. Bu durum şekil 3'deki örnek bir fonksiyon üzerinde arama algoritmasının aldığı yol üzerinde (kalın çizgili) görülmektedir. $f'(u)=0$ çözümünde türev alma işlemini gerçekleştirmek için geri besleme yoluna z^{-1} öteleyici ve fark alıcı konularak $f(k)-f(k-1)$ değeri kullanılabilir. Fakat adım boyu farklı olduğundan doğru sonuca ulaşmaktaki başarımlar düşmektedir. İşlem bilgisayar ortamında yapıldığında tercih edilmesi gereken, $f(x)$ yerine sayısal türev (örneğin $(f(x+h)-f(x))/h$) kullanmaktır.

$f(x)=0$ alt işletiminde, sonlandırma kriteri olarak global aramada istenen duyarlılığı almak gerekmez. Çünkü, global noktalara $f'(x)=0$ çözümünün sonrasında ulaşılmaktadır. Örneğin, $E_m=0.000001$ hata üst sınırı ile çözüm isteniyorsa; kök arama çevrimlerinde $E_m=0.01$; türev arama çevrimlerinde ise $E_m=0.000001$ seçmek aynı doğrulukta, daha kısa sürede çözüme ulaşmayı sağlar.

Amaç ölçüt fonksiyonu değişkenleri üzerine getirilebilecek kısıtlamalara uymak için, amaç ölçüt fonksiyonuna ceza bileşeni eklenebilir [29].



Şekil 3 Tek değişkenli bir fonksiyonda global enbüyük noktanın aranması.

Burada önerilen global eniyileme yönteminde, $f(x)$ için bir Lipchitz değeri verilebilmesi durumunda çözümün doğruluğu garanti edilir. Fonksiyonun süreksiz olması durumunda, kuramsal olarak doğru çözümü hiç bir yöntem garanti etmez. Açıklanan yöntemin standart test fonksiyonlarında başarıya ulaştığı görülmüştür. Birden çok değişkenli eniyileme problemlerinde bir değişken için arama yapıldıktan sonra, diğer değişkenlerin yeni değerleri için aramanın tekrarı sırasında bir önceki global eniyi nokta referans olarak alınmaktadır. Bu nedenle sonuca erişmedeki çevrim sayısı azalır. Benzer şekilde birden fazla bilgisayarın aynı problemi farklı aralıklarda çözmesi sırasında, arama yapan bilgisayarların birbirine gönderdikleri yeni global noktalar içerisinde en büyük olanı (enbüyük için) daha sonraki çevrimde tüm diğer makinalarda yeni referans nokta olarak alınmaktadır. Böylelikle aramaya kendi bölünmüş değişken aralığında katılan bilgisayarlar, gerçek global noktadan uzak yerleri büyük adım aralıklarıyla, daha az sayıda değerlendirme ile aşar.

A NEW GLOBAL OPTIMIZATION METHOD BY USING FUZZY LOGIC CONTROLLER

SUMMARY

Let X be a compact set called feasible region and f be an objective function such that $X \subset \mathbb{R}^n$, and $f: \mathbb{R}^n \rightarrow \mathbb{R}^1$. The global minimization problem is defined as,

$$f^* = \min f(x) \quad , \quad x \in X \quad (1)$$

where f^* is intended to be found as the minimum value of $f(x)$. Since the value x which minimizes the function $f(x)$ will maximize $-f(x)$ the above definition is valid for global maximum problem too and it is called as global optimization problem in general case. Discontinuity and multi-extremality of the objective function f restricts the optimization technique to be used. For example, some problems may occur in the optimization of discontinuous objective functions if the method needs the derivative values of the function f .

Block diagram of a discrete unit feedback control system is shown in figure 1. In case of replacement of the set of differential equations by the function $f(x)$, solution of the global optimization problem that was defined in (1) can be approached with the same block diagram.

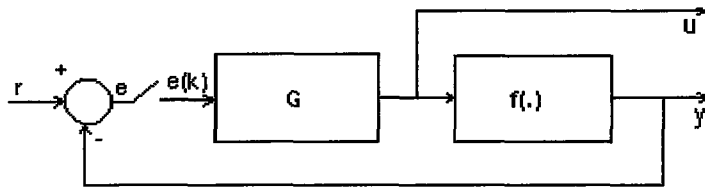


Figure 1 Numerical analysis problem with the automatic control system point of view.

According to closed loop control theory, a properly designed controller G makes the error sign e be zero in steady state. Equivalently, let k be the number of function evaluations then

$$k \rightarrow \infty \text{ for } e(k) \rightarrow 0, \quad f(u(k)) \rightarrow r \quad (2)$$

or

$$\lim_{k \rightarrow \infty} |f(u) - r| \rightarrow 0 \Rightarrow u^* = x^*, f(x^*) = r \quad (3)$$

where * represents the optimal solution and r is the reference value.

The output of the controller u yields the solution of the equation $f(x)-r=0$ as it can be seen from the equation (3). If reference sign r is particularly set to zero then the steady state value of u will be equal to one of the roots of $f(x)$. This control theory approach was previously used for the calculation of the step length in numerical solution of differential equations [2]. Roots of a function which will be evaluated in obtaining solution of the global optimization problem is determined by the same approach.

A single or multi-variable objective function is placed instead of $f(\cdot)$ and a fuzzy rule based controller takes the place of G in the proposed method.

The closed loop control system shown in figure 2 finds the solution of the equation $f(x)-r=0$ where k represents the number of steps and controller output is considered as the solution. Some problems arise in application of classical linear controllers with control theory approach for the solution of equations instead of well known iterative numerical methods. For example, when a PID type controller is used for root seeking, stability problems are met in reaching the solution. Large valued derivatives of the function around the roots to be solved in an equation, not only causes oscillations, and it may also be the reason of missing the root.

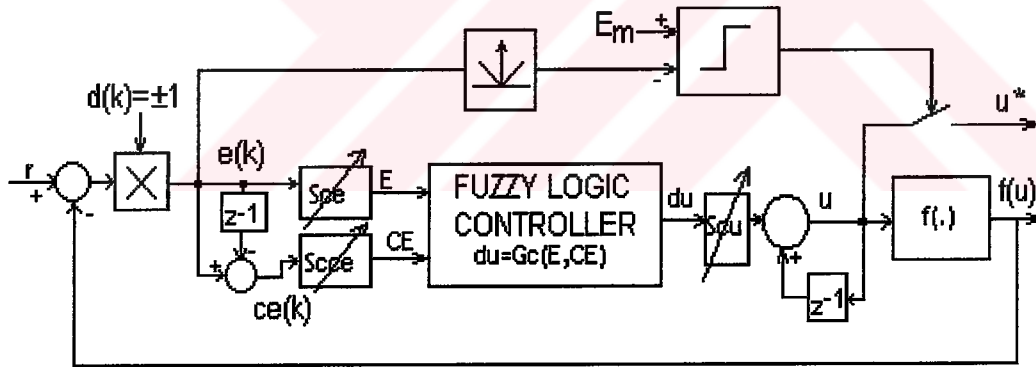


Figure 2 Block diagram of the system used for solution of the single variable equation $f(x)-r=0$.

In this new method, the fuzzy logic controller structure offered by Mamdani is used in the block diagram given in figure 2 [24]. e is actual error, ce is change of actual error, E is scaled error, CE is scaled change of error, S_{ce} is scaling factor of the error, S_{ce} is scaling factor of the change of error, du is output of the fuzzy logic controller, u is discretely integrated controller output and u* represents the value of calculated root in figure 2.

Output of this kind of fuzzy controller saturates over a certain limit depending on the error and change of the error. The integrator at the output of the fuzzy controller is only the memory element in the closed loop system used for finding of the roots.

Unsuitable selection of the parameters (Sce, Scce) causes only limited amplitude oscillation around the root since new method generates bounded output for bounded input if at least fuzzy decision table has been chosen properly.

Let the upper bound of the search region be US (upper bound) and starting point of search be AS (lower bound) of the region then $u(0)$ must be set to AS for searching in the interval (AS,US). This numerical assignment provides search to be started from a predetermined point.

There exist many methods for optimizing or finding the roots of a unimodal function. All of these methods may fail in non-unimodal functions. Complexity of the function does not affect the achievement in reaching the solution, since only the fuzzy decision table and previously trial points are evaluated starting from a given point. This property will be used in solution of global optimization problem.

There are several possible approaches for global optimization with fuzzy logic controller. The proposed optimization algorithm is based on scanning of the feasible region in a closed loop control system with fuzzy logic controller. Search direction may be from the upper bound to lower bound or from lower bound to upper bound or it even can be applied in sub-intervals covering the feasible region. Optimization algorithm accepts the lower bound as the starting point and finds the maximum value of a given function $f(\cdot)$. Roots of the equation $f'(u)=0$ is searched through running towards the right hand side for $r=0$ and the first found root is considered to be new global maximum value if the slope of the function is positive in the starting point of search. Otherwise root of f is searched through running towards the right hand side for $r=f(u)$ and the new reached point determined by the solution of the equation $f'(u)=0$ is considered to be a new global maximum value.

This procedure is repeated with solution of $f(u)=r$ by taking r is equal to the global maximum value in both cases. The lastly determined root of $f'(u)=0$ will be the global maximum value, if controller output (u) exceeds the upper bound US (saturation). This search procedure is illustrated with an example function shown in figure 3. The bold line represents the route of the search algorithm in finding the solution.

$f(k)-f(k-1)$ value can be evaluated by adding a shift operator (z^{-1}) and a subtracter on the feedback line for evaluating the derivative $f'(u)$ however this operator decreases the performance in finding the correct solution since step lengths are different in each iteration. The numerical equivalent of derivative such as $(f(x+h)-f(x))/h$ has to be preferred because these operations are handled in computer environment.

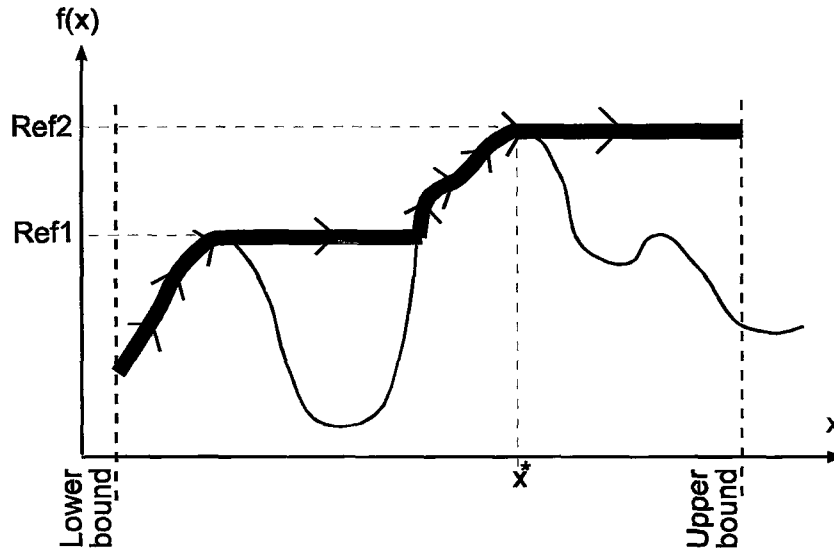


Figure 3 Search of global maximum value of a single variable function

There is no need target resolution to be a precise value of global optimization as the ending criteria in the sub operation $f(x)=0$ because global extremal points are only reached after the solution of $f'(x)=0$. For example, if the demanded upper limit of the error is $E_m=0.000001$, the ending criteria for root search is set as $E_m=0.01$ and stopping criteria of the derivative loops is set as $E_m=0.000001$ then the same correct global maximum value will be found by lesser number of evaluations.

In order to fit in restrictions on the function optimization parameters penalty components can be added to the objective function.

Finding the correct solution is guaranteed if a Lipchitz value can be given for $f(x)$ in the proposed method of global optimization. Although theoretically none of the solution methods can guarantee finding the correct global maximum value, the proposed method is successful in solution of standard test functions. Previously found global maxima is set as new reference for other variables after a search for a variable in multi variable optimization problems Otherwise, the number of evaluations would be increased exponentially in proportion to the number of variables. When the problem is solved by proposed method on a network, each computer shares a sub interval and new global maximum points are sent to each other after all of the derivative loops. This feature provides decreasing the amount of iterations for the machines those evaluate some intervals far from real global maxima region.

1. GİRİŞ

Global eniyileme problemi, mühendislikten, ekonomik uygulamalara kadar geniş bir alandaki eniyileme problemlerinin en gerçekçi ve genelleştirilmiş halidir. Bu çalışmada, otomatik kontrol kuramı yaklaşımı ile ardışıl arama yöntemleri arasında bir ilişki kurulmuştur. Bu ilişkiden faydalanılarak, amaç ölçüt fonksiyonunun kontrol edilecek sistem, bulanık kontrolörün ise özyineleyici olarak kullanıldığı, yeni bir global eniyileme yöntemi önerilmiştir.

1.1 Global Eniyileme

Tüm eniyileme problemleri bir Amaç Ölçüt Fonksiyonu (AÖF) üzerine kurulur. AÖF, tasarım ya da karar aşamasındaki bir işlemde, değişkenlere bağlı olarak toplam getiri ya da götürünün bir ölçüsüdür. Eniyilemenin amacı, toplam getiriyi temsil eden AÖF'yi en büyük yapan değişken değerlerini bulmaktır. Eğer AÖF'götürüyü temsil ediyorsa, bu durumda enküçülten değişken değerleri belirlenmek istenir.

Yerel eniyileme, AÖF'nin türevinin bir defa işaret değiştirdiği problemlerin çözülmesi işlemidir. Diğer bir deyişle yerel eniyileme, arama yapılan aralıkta yalnızca bir tane enbüyük ya da bir tane en küçük içeren AÖF'lere uygulanabilir. Bu tür eniyileme problemlerinin çözümü, fonksiyon gradiyeninden faydalanılabildiği durumlarda (AÖF sürekli bir fonksiyon ise) oldukça kolaydır. Ancak birçok eniyileme probleminde AÖF, yüksek dereceden polinomlar ya da periyodik bileşenler içerdiğinden birden çok sayıda yerel enbüyük ya da enküçük vardır. Birden çok sayıda yerel enbüyük ya da yerel enküçük içeren AÖF'lerin eniyilenmesine global eniyileme adı verilir.

1.2 Global Eniyilemenin Önemi

Global eniyileme, eniyileme problemlerinin büyük bir kısmını oluşturmakla birlikte yerel eniyilemeye göre çözümü oldukça karmaşıktır. Hisseseledi portföy yönetiminden cihaz tasarımına kadar değişkenlere bağlı olarak karar vermeyi gerektiren işlerde global eniyileme gereksinimi ortaya çıkmaktadır.

Global eniyileme algoritmasının amacı verilen koşullarda (aralık, sınırlamalar, çözünürlük vb.) en az sayıda değerlendirme ile en hızlı şekilde çözüme ulaşmaktır. %100 olasılıkla doğru çözümü bulan yöntemler genellikle yüksek sayıda değerlendirme yapmakta ya da AÖF'nin yapısı hakkında ön bilgiye ihtiyaç duyulmaktadır.

1.3 Bölümlerin Açıklaması

Bu tezin ikinci bölümünde global eniyileme problemi tanımlanmış ve literatürde sıkça karşılaşılan mevcut eniyileme teknikleri kısaca anlatılmıştır. Burada değinilmeyen çok sayıda global eniyileme yöntemi de bulunmaktadır. Yer alan açıklamanın amacı kullanılagelmiş yöntemlerin arama stratejileri hakkında fikir vermek ve kabaca bir sınıflandırma yapmaktır.

Üçüncü bölümde, bu çalışmada önerilen yöntemin temelini teşkil eden otomatik kontrol kuramı yaklaşımı verilmiştir. Bu yaklaşımda kontrol edilecek sistem yerine eniyilenecek fonksiyon yer almakta ve kontrolör çıkışı, sayısal çözüm olarak değerlendirilmektedir. Önerilen yöntem, üçüncü bölümde açıklandığı şekli ile yerel eniyileme ve kök arama problemleri için geçerlidir. Burada karşılaşılan salınım, kök atlama vb. sorunlar dördüncü bölümde ele alınmıştır. Bu bölümde lineer bir kontrolör kullanımı ile ortaya çıkan sorunların aşılması için bulanık kontrolör kullanımı önerilmiştir. Beşinci bölümde, kontrol kuramı yaklaşımının kullanıldığı yeni bir global eniyileme yöntemi önerilmektedir. Global eniyi noktaya erişebilmek için geribeslemeli kontrol çevrimine bir algoritma dahilinde değişken referans işaretleri uygulanmaktadır. Altıncı bölümde ifade edildiği gibi benzer bir yaklaşımla, mevcut sayısal arama algoritmalarının, bazı iyileştirmelerle gerçek zamanlı kontrol sistemlerinde kullanılmasını mümkün kılar.

Bu çalışma global eniyileme problemine yeni bir çözüm yöntemi getirmekte, ayrıca kontrol kuramı ile sayısal arama yöntemleri arasında bir ilişki kurulmasını sağlamaktadır. Bu yaklaşım çalışmaya özgünlük kazandırmaktadır.



2. GLOBAL ENİYİLEME PROBLEMİ VE ÇÖZÜM YÖNTEMLERİ

Eniyileme problemlerinde atılması gereken ilk adım, Amaç Ölçüt Fonksiyonunu (AÖF) belirlemektir. AÖF’de yer alan ifadeler, enbüyütmede her bir değişkenin sağladığı kazancı, enküçültmede ise kaybı ifade eder. Eniyilemenin hedefi, amaç ölçüt fonksiyonunu enbüyük ya da en küçük yapan değişken kümesini belirlemektir. Eniyileme probleminin genelleştirilmiş bir hali olan global eniyilemede birden çok ekstremumu olan amaç ölçüt fonksiyonlarını en büyük ya da en küçük yapan noktalar belirlenmek istenir.

2.1 Global Eniyileme Probleminin Tanımı

Global enküçültme problemi, $f: R^n \rightarrow R^1$ amaç ölçüt fonksiyonu ve $X \subset R^n$ kompakt bir küme olmak üzere,

$$f^* = \min_{x \in X} f(x) \quad (2.1)$$

şeklinde ifade edilen f^* değerini belirlemek şeklinde tanımlanır. $f(x)$ fonksiyonunu enküçülten x değeri $-f(x)$ fonksiyonunu enbüyüteceğinden, bu tanım global enbüyük için de geçerlidir ve genel olarak global eniyileme olarak adlandırılır.

AÖF’yi global enküçük yapan noktaların kümesi X^* olarak adlandırılırsa,

$$X^* = \{x \in X \mid f(x) = f^*\} \quad (2.2)$$

şeklinde tanımlanan her bir nokta, global eniyiyi veren noktalar olacaktır. Global enküçük bulma işlemi, $f(x^*) = f^*$ olmak üzere

$$x^* = \arg \min_{x \in X} f(x) \quad (2.3)$$

şeklinde ifade edilebilir [1].

X içerisinde seçilen $\{x_k\}$ noktalarını, $f(x_k)$ 'yi f^* 'a yaklaştıracak ya da eşitleyecek şekilde belirleyen işletim düzenine global eniyileme algoritması denir.

2.2 Global Eniyileme Probleminin Sınıflandırılması

Global eniyileme problemleri, genel olarak kısıtlamalı ve kısıtlamasız olmak üzere iki temel gruba ayrılırlar. Kısıtlamasız problemler, kesin bir tanımlama ile, uygulanabilir çözüm kümesi (feasible region) $X=\mathbb{R}^n$ olan kümelerdir. İlgili duyulan bir aralıkta global eniyi aranırken kısıtlama sadece aralığı tasvir ediyorsa çözüm yine kısıtlamasız sınıfına sokulmalıdır. Global aramada, yerel arama problemlerinden farklı olarak, X'e alttan ve üstten sınırlama getirilmediği durumda genel bir çözüm şekli yoktur.

X içerisindeki yerel ekstremumların sayısı probleme göre çok değişkenlik gösterir. Göreceli olarak yerel ekstremumların sayısı az problemlerin çözümü daha kolay olacaktır. Yerel ekstremumların dağılımı da çözüme ulaşmada etkilidir.

Bir çok uygulamada, eniyileyen fonksiyon değerlerinin aranmasında kullanılabilecek ön bilgiler mevcuttur. Bu ön bilgiler amaç ölçüt fonksiyonunun matematiksel özellikleri ile ilgili veriler içerir. Çözüm algoritmasında ön bilgi kullanımı çözüme ulaşma hızını ve güvenilirliğini artırır.

Amaç ölçüt fonksiyonu f ve uygulanabilir çözüm kümesi $X \subset \mathbb{R}^n$ hakkında ön bilgi bulunması uygun çözüm algoritmasının seçimi ve çözüme ulaşmadaki başarısı üzerinde etkilidir. Amaç ölçüt fonksiyonu f hakkında sıkça karşılaşılan ön bilgi türleri şöyle sıralanabilir,

- a) $F \subset C(X)$
- b) $F \subset C^1(X)$
- c) $F \subset C^2(X)$
- d) $F \subset C^2(X)$ ve Hessian hesaplanabiliyor, $f \in F$
- e) $F \subset C^k(X)$
- f) $F \subset \text{Lip}(X, L)$, burada L bir sabit
- g) $F \subset \text{Lip}(X, L, \rho)$, burada L bir sabit, ρ ise X üzerinde metrik
- h) $F \subset \{f \in C^1(X) : \forall f \in \text{Lip}(X, L)\}$ bazı sabit L 'ler için

- i) $F \subset \{f \in C^2(X) : \|\nabla^2 f\| \leq M\}$ bazı sabit M 'ler için
- j) f , derecesi p 'yi geçmeyen bir polinom
- k) f , karesel
- l) f rasyonel bir fonksiyon
- m) f için dış bükey bir zarf tanımlanabiliyor (ya da iç bükey)
- n) $f = f_1 - f_2$ şeklinde iki dış bükey fonksiyon cinsinden tanımlanabiliyor
- o) f 'nin l tane yerel en küçük noktası var
- p) f 'nin yerel en küçük noktası l taneden az
- r) Global ekstremumu (f^*) veren bir tane x^* var

2.3 Global Eniyileme Yöntemleri

Bazı global arama yöntemleri, yerel arama yöntemlerinin iyileştirilmesi ile ortaya çıkmıştır. Yerel arama üzerine kurulu bu yöntemler 2.3.1 ayrıtında yer almaktadır. 2.3.2'de yer alan, arama alanını örten yöntemler üzerine çok sayıda kuramsal çalışma bulunmasına karşın uygulamada aynı oranda yaygın değildirler. 2.3.3 ayrıtında amaç ölçüt fonksiyonunun tek değişkene indirgenmesine dayalı yöntemlere değinilmiştir. 2.3.4 de amaç ölçüt fonksiyonunun stokastik modeline dayalı yöntemler açıklanmıştır. Son zamanlarda rastlantısal örneklemeye dayalı yöntemler yaygınlık kazanmaktadır. Önceden verilen doğru çözüme erişme olasılığını hedef alan bu yöntemlere 2.3.5'de değinilmiştir. Evrimleşme sonucu genlerin, kendilerini nesilden nesile iyileştirerek aktarmalarını örnek alan genetik eniyileme algoritmaları üzerine çalışmalar sürmektedir. 2.3.6'daki evrimsel arama yöntemleri kapsamında genetik eniyileme yöntemi anlatılmıştır.

2.3.1 Yerel Arama Üzerine Kurulu Yöntemler

$f: R^n \rightarrow R^1$ amaç ölçüt fonksiyonu, $X \subset R^n$ kompakt bir küme, $|x - x^*| < \epsilon$ olacak şekilde bir $\epsilon > 0$ sayısı varsa ve tüm $x \in X$ 'ler için $f(x) \geq f(x^*)$ ilişkisi geçerliyse x^* 'a bir yerel en küçük denir.

Yerel arama yöntemleri (2.4)'deki ardışıl adımlama üzerine kurulur.

$$x_{k+1} = x_k + \gamma_k s_k \quad k=1,2,\dots \quad (2.4)$$

Burada, $x_1 \in X$ başlangıç noktası olmak üzere x_{k+1} ile $k+1$ 'inci değer belirtilmektedir. s_k , k 'inci adımdaki arama yönü ve $\gamma_k \geq 0$ ise adım uzunluğudur. Yerel enküçültme algoritmaları $f(x_{k+1}) \leq f(x_k)$ olacak şekilde $\{\gamma_k\}$ ve $\{s_k\}$ 'nin seçiminde farklılık göstermektedir.

Yerel aramalar ile global en iyi değerleri arayan yöntemlerin ilki ve en yaygını “çok noktadan başlatım”dır (multistart). Bu yöntem farklı başlangıç noktaları ile çok sayıda yerel arama yapılmasına dayanır. Genellikle başlangıç noktaları, AÖF içindeki değişkenler arama aralığına düzgün yayılmış bir ızgara üzerinden seçilir. Güvenli bir şekilde global eniyi noktaya erişebilmek için, yerel eniyi noktaların sayısından çok daha fazla başlangıç noktası kullanmak gerekmektedir. Çoğunlukla yerel eniyi noktaların sayısı arama öncesi bilinmediğinden ızgara aralığının seçiminde zorluklarla karşılaşılır. Izgara aralığı, fonksiyonun değişimlerine göre çok sık ise hesaplama sayısı çok artmakta; az ise gerçek global eniyi ile çakışan yerel eniyi atlanabilmektedir. Çok noktadan başlatma ile global eniyiyi arayan algoritmalarından en basit ve yaygın olanı “en yakın komşular” yöntemidir (nearest neighbour method) [1].

Yerel aramalar üzerine kurulmuş diğer bir yöntem ise “tünel açma”dır (tunneling method). Bu yöntem iki aşamadan oluşmaktadır. Birincisi verilen bir $x(0)$ başlangıç noktasından itibaren yerel enküçük x^* ’in bulunmasıdır. İkinci aşama ise tünel açma fonksiyonu $T(x)$ ’in belirlenmesidir. $T(x)$ her noktada türevi alınabilen, sürekli ve $f(x)$ ’e bağlı bir fonksiyondur. Levy ve Montalvo 1985 yılında aşağıdaki $T(x)$ tanımını önermiştir.

$$T(x) = T(x, x^*, \alpha) = ((f(x) - f(x^*)) / \|x - x^*\|)^\alpha \quad (2.5)$$

burada $\alpha > 0$ bir sabittir [2]. Tünel açma algoritmalarında $T(x)$ ’e bağlı olarak $x \in X$ olacak şekilde yeni bir nokta belirlenir ve yerel arama ile devam edilir. Önceden belirlenmiş bir döngü sayısına ulaşıldığında, ya da yeni nokta $x \notin X$ kaldığında bu çevrim sonlandırılır.

Yerel eniyi noktalardan global noktaya geçiş için ortaya atılan fikirlerden biri de global noktaya erişim sürecini bir diferansiyel denklem ile ifade etmektir [3]. Sürekli ikinci derece türevi bulunabilen bir $f(x)$ fonksiyonunun gradiyenini $g(x) = \nabla f(x)$ ve

Hessian'i $H(x)=\nabla^2 f(x)$ olsun. s , $+1$ ya da -1 değerini alabilen bir sabit olmak üzere başlangıç koşulu $g(x(0))=g_0$ olan,

$$\frac{d}{dt}g(x(t)) = sg(x(t)) \quad (2.6)$$

diferansiyel denkleminin çözüm kümesi $g(x(t))= g_0 e^{st}$ yapısındadır. Eğer $s=1$ ve $t \rightarrow -\infty$ ise ya da $s=-1$ ve $t \rightarrow +\infty$ ise çözüm sıfıra yakınsayacaktır [4]. Bu, çözüm eğrisinin, f 'nin kararlı bir noktasına yakınsama eğilimi içinde olduğunu göstermektedir. Branin yöntemi, f 'nin kararlı noktalarına ulaşabilmek ya da bir kararlı noktadan bir diğerine geçebilmek için (2.6)'daki diferansiyel denklemin sıra ile $s=+1$ ve $s=-1$ alınarak çözülmesine dayanır.

2.3.2. Arama Bölgesini Örtme Yöntemleri (Set Covering Methods)

Uygulamada global arama yapılan amaç ölçüt fonksiyonunun değişim hızı genellikle sınırlıdır. ρ , X üzerinde tanımlı bir metrik ve L ise bu değişimin üst sınırını ifade eden bir sabit olmak üzere,

$$|f(x_1)-f(x_2)| \leq L\rho(x_1,x_2) \quad (2.7)$$

sağlayan fonksiyonlara Lipschitz denir ve fonksiyonların değişim mertebelerini gösteren en yaygın ifadelerden biridir. Amaç ölçüt fonksiyonunun değişim mertebesi biliniyor olsun. Eğer fonksiyonun bir $x_i \in X$ noktasındaki değeri $f(x_i)$ biliniyorsa,

$$X_i \subset \{x \in X : f(x) \geq f(x_i) - \delta\} \quad (2.8)$$

kümesi belirlenebilir. Burada $\delta > 0$ bir sabittir. Eğer $x_1, \dots, x_N \in X$ izin verilen çözüm bölgesinden $X_1, \dots, X_N$ tüm X 'i örtecek şekilde seçilirse, diğer bir gösterimle,

$$X \subset \bigcup_{i=1}^N X_i \quad (2.9)$$

sağlanıyorsa global enküçültme problemi δ hassasiyetiyle çözülebilir. Bu durumda,

$$f_N^* = \min_{1 \leq i \leq N} f(x_i) \quad , \quad f^* = \min f \quad (2.10)$$

olmak üzere,

$$f_N^* - \delta \leq f^* \quad (2.11)$$

sağlayacak şekilde $x_1 \dots x_N$ noktalarının seçimini tanımlayan arama yöntemlerine “arama bölgesini örten yöntemler” (set covering methods) adı verilir. Arama bölgesinden değişkenlerin eşit aralıklarla seçildiği ızgara yöntemleri bu sınıftaki çoğu arama yöntemlerinin çıkış noktasıdır. Amaç ölçüt fonksiyonunun bir değişkeni x , $[a,b]$ aralığında bulunsun. x 'in N tane noktada değerlendirilmesi halinde (2.1)'i sağlayan x^* 'in bulunmasında yapılacak hatanın üst sınırı,

$$e_{\max} \leq \frac{L(b-a)}{2N} \quad (2.12)$$

olacaktır [4]. Pijavskij-Shubert algoritması [5] örtünmenin en temel halidir. Bu yöntemde (2.12)'yi sağlayan N tane nokta arama aralığında homojen olarak seçilir. Bu yöntem kuramsal açıdan uygun gözükse de uygulamada sorunlar çıkmaktadır. Özellikle çok değişkenli fonksiyonlarda istenen hassasiyetle çözüme ulaşmak için çok sayıda noktada değerlendirme yapmak gerekmekte ve büyük bir belleğe ihtiyaç duyulmaktadır. Bu soruna çözüm olarak seçilen noktaların değerlendirilmesi sırasında bazı aday alt çözüm kümelerini eleyen ve “arama bölgesinin aktif örtülmesi” adı verilen yöntemler (active covering methods) geliştirilmiştir. Evtushenko'nun önerdiği yöntem (1971,1985) [6],[7] bunlardan biridir. X , $[a,b]$ aralığında çözüm bölgesi olsun. k tane değerlendirme sonunda,

$$f_k^* = \min\{f(x_1), f(x_2), \dots, f(x_k)\} \quad (2.13)$$

olacaktır. Evtushenko, yeni noktaların eşit aralıkla seçilmesi yerine

$$x_1 = a, \quad x_{k+1} = x_k + (f(x_k) - f_k^* + 2\delta)/L \quad (2.14)$$

ardışıl bağıntısıyla seçimini önermiştir. (2.14) işlemi $x_k > b$ durumunda sonlandırılacaktır.

Örtünme için benzeri bir yöntem de Brent (1973) [8] tarafından önerilmiştir. Brent'in geliştirdiği algoritma benzerlerinden farklı olarak Lipchitz özelliği yerine arama yapılan fonksiyonun ikinci türevinin üst sınırının belirli olması şartını kullanmaktadır. Ancak uygulamada çoğu kez bu şartı sağlamak mümkün olmamaktadır.

2.3.3 İndirgeme ve Alan Daraltma Kullanan Yöntemler

Tek değişkenli fonksiyonları eniyilemek için bir çok etkin algoritma mevcuttur. Bu algoritmaların çok değişkenli eniyileme problemlerine uyarlanması için değişik teknikler geliştirilmiştir. Bunların en yaygın olanı "çok adımlı boyut indirgemesi" yöntemidir. n değişkenli bir problemde arama alanı,

$$X = \{x | a_i \leq x_i \leq b_i, i=1,2,...,n\} \quad (2.15)$$

olarak tanımlanırsa,

$$\min_{x \in X} f(x) = \min_{a_1 \leq x_1 \leq b_1} \dots \min_{a_n \leq x_n \leq b_n} f(x_1, x_2, \dots, x_n) \quad (2.16)$$

olacaktır. Tüm arama alanında, sırayla, her bir değişkenin örneklenmiş tüm diğer değişkenler cinsinden değerlendirilmesi gereksinimi, değişken sayısı ile değerlendirme sayısının üstel olarak artmasına sebep olmaktadır. Bu sebeple uygulamada $n=5$, bu yöntem için bir üst sınır olarak görülmektedir.

Bazı çok değişkenli global eniyileme algoritmaları raslantısal seçilen doğrultularda tek değişkenli arama yapmaktadır. Bremermann tarafından önerilen algoritma bu tekniğe örnek olarak gösterilebilir [9].

Alan daraltma yöntemleri, boyut indirgmeden farklı olarak, arama aralığının var olan tüm değişkenler dahilinde, küçültülmesini ve işlem sayısının azaltılmasını amaçlar. Bu işlem iki şekilde gerçekleştirilmektedir. Arama alanı boyunca yapılan örneklemeler sonucu oluşturulan, (2.8),(2.9) ile ifade edilen X_i alt alanlarından global ekstremum içermesi mümkün olmayanların belirlenerek arama dışı tutulmasıdır. Ardarda gelen komşu alt alanlardan birinin global ekstremum

içermediğinin belirlenmesi ancak fonksiyon hakkındaki önbilgiler değerlendirilerek kesinlik kazanabilir.

Alan daraltma üzerine ikinci bir yöntem ise “aralık aritmetiği” dir (interval arithmetics). Bu yöntemde fonksiyon topolojik olarak değerlendirilmektedir. $f(x_1, x_2)$, iki değişkenli bir amaç ölçüt fonksiyonu olsun. x_1, x_2 değişkenlerinin alt ve üst sınırları $[a, b] = \{x_1 : a \leq x_1 \leq b\}$ ve $[c, d] = \{x_2 : c \leq x_2 \leq d\}$ şeklinde ifade edilirse aralık aritmetiğinin aksiyomları şu şekilde sıralanabilir,

$$[a, b] + [c, d] = [a + c, b + d] \quad (2.17)$$

$$[a, b] - [c, d] = [a - d, b - c] \quad (2.18)$$

$$[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)] \quad (2.19)$$

$$[a, b] \div [c, d] = [a, b] \times [1/d, 1/c] \text{ , } 0 \notin [c, d] \quad (2.20)$$

Böylelikle genel arama öncesi ya da yöresel alt aramalarda fonksiyonun alacağı alt ya da üst sınır için bir sınırlama getirmek mümkün olmaktadır. Bir çok algoritmanın aralık aritmetiğinden faydalanan versiyonları geliştirilmiştir [10]. Örneğin Shen ve Zhu 1987’de Pijavskij-Shubert algoritması üzerine bu tür bir iyileştirme yapmışlardır [11].

Aralıkları dışlamakta seçilen bir başka yol Newton algoritmasının aralıkları değerlendiren halidir [4]. X_1 değerlendirilmeye başlayan ve orta noktası x_1 olan bir alt arama alanı olsun. Ardışıl işlem şu şekilde tanımlanmaktadır,

$$N(X_k) = x_k - f'(x_k)/f''(x_k) \quad (2.21)$$

$$X_{k+1} = X_k \cap N(X_k)$$

Burada x_k , değerlendirme yapılan k . X_k bölgesinin orta noktasıdır. Hansen [12] $0 \in f'(X_1)$ durumunu da içeren bir Newton aralık daraltma yöntemi [10] geliştirmiştir.

2.3.4 Stokastik Modellemeye Dayalı Yöntemler

Ω , olasılık uzayı, X , uygulanabilir bölge olmak üzere $x \in X$, $\omega \in \Omega$ ve $\varphi(x, \omega)$ bir rastlantı fonksiyonu olsun. f ile ilgili ön bilgi $f \in F = \{\varphi: X \times \Omega \rightarrow \mathbb{R}^1\}$ ise $f(x) = \varphi(x, \omega)$ sağlayan φ ve ω bulmak mümkündür. Diğer bir deyişle f amaç ölçüt fonksiyonu, rastlantı değişkenleri uygun seçilmiş stokastik bir süreç ile modellenenbilmektedir [1]. Stokastik model üzerine kurulu global eniyileme algoritması ilk defa 1964 yılında Kushner [13] tarafından önerilmiştir. Kushner, çözümünde amaç ölçüt fonksiyonunun Wiener sürecine benzer davranış gösterdiğini varsaymıştır. $X=[a,b]$ aralığında uygun çözüm kümesi ve $\varphi(x, \omega)$ ise ortalaması $\mu(x) = \mu = \text{sabit}$, varyansı $\eta(x, z) = \sigma^2$, olan bir Wiener süreci olsun. σ^2 başlangıçta bilinmiyorsa, m tane eş uzaklıklı noktada f değerlendirilerek enbüyük benzerlik kestirimi yöntemi ile (maximum likelihood estimator) bir değer belirlenir [1]. Ayrıca algoritma işletimi sırasında da σ^2 üzerinde düzeltme yapmak fayda sağlamaktadır. $k \geq m$ noktada f değerlendirilir ve $a = x_1 < \dots < x_k = b$ noktalarında elde edilen $y_i = f(x_i)$ değerlerinden koşullu ortalama $\mu(x|\Sigma_k)$ ve varyans $\sigma^2(x|\Sigma_k)$ değerleri hesaplanır. $i=1, \dots, k-1$ tüm aralıklarda $E(\varphi_i|\Sigma_k)$ beklenen değerleri hesaplanarak $\varphi_i = \min \varphi(x)$ olan aralık bulunur. Daha sonra elde edilen aralık tekrar alt aralıklara bölünerek aynı işleme devam edilir. Aralık genişliği önceden belirli bir değerin altına düştüğünde işlem sonlandırılır

Stokastik modellemeye dayalı bir başka global eniyileme yöntemi de tavlama benzetişimidir (simulated annealing). Bu yöntem, donan ve kristal yapıya geçen bir sıvıda ya da ergidikten sonra soğutulan ve tavlanan metallerde görülen termodinamik olayların benzetimine dayanmaktadır. Yüksek sıcaklıklarda moleküller birbirlerine göre serbestçe hareket ederler. Eğer sıvı yavaşça soğutulursa atomlar düzgün, saf bir kristal yapı oluşturacak şekilde dizilirler. Bu kristal yapı maddenin en düşük enerjili durumudur. Eğer soğutma işlemi hızlı yapılırsa atomlar bu dizilime ulaşamazlar ve daha yüksek enerjili amorf bir yapıda kalırlar. Sistemin E_0 enerjili bir durumdan E_1 enerjili bir duruma geçmesi olasılığını gösteren dağılım fonksiyonu,

$$P(E) = e^{(-\frac{E_1 - E_0}{kT})} \quad (2.22)$$

olarak ifade edilir. Burada T [K°] denge sıcaklığı, k ise Boltzman sabitidir. Termodinamik olaylar ile eniyileme arasındaki analogi ilk defa 1953 yılında Metropolis [14] tarafından kurulmuştur. Bu amaçla E yerine amaç ölçüt fonksiyonu konulmakta dolayısıyla minimize edilmeye çalışılmaktadır. 1983’de Kirkpatrick tarafından global eniyileme için aynı temele dayalı bir yöntem geliştirilmiştir [15]. Bu yöntemde başlangıç sıcaklığı, $P(E) \geq 0.999$ olacak şekilde büyük bir değer seçilir. $E=f$ içinden rasgele seçilmiş bir sistem parametresi değiştirilir ve enerji fonksiyonunun yeni bir global enküçük değere ulaşp ulaşmadığı kontrol edilir. Yeni bir değere ulaşılmışsa, o parametre, yeni değeri ile sabit tutulur. T , %0.01...%2 arasında düşürülerek işleme devam edilir. Enerjide artış görülmüşse sıcaklık düşürülmez ve eski değer geri alınır ve $P(E)$ dağılımına göre rasgele yeni bir parametre değişikliği yapılır. Bu işlem son k tane iterasyonda (k bir sabit) $E=f$ ’nin değeri önceden belirli bir ϵ sayısından küçük kalana kadar devam ettirilir. Böylelikle enküçük enerjili atom dizilimine benzeştirilerek amaç ölçüt fonksiyonunu enküçük yapan değişken değerleri belirlenmiş olur.

2.3.5 Raslantısal Örneklemeye Dayalı Yöntemler

Raslantısal arama yöntemleri önceden belirli bir olasılık değeri içinde kalacak şekilde örneklem yapılmasına dayanır. Bunlardan ilkel olanları ham örneklem (crude sampling), tek noktadan başlatım (single start) ve çok noktadan başlatımdır (multistart). Ham örneklemde yerel arama yapılmaksızın belirli sayıda rasgele örnek alınır ve en iyi değeri sağlayan nokta civarında örneklemeye devam edilir. Tek noktadan başlatımda rasgele örneklenen noktalar içerisinde en uygunu seçilerek yerel arama yapılır. Çok noktadan başlatımda ise rasgele seçilen tüm noktalar civarında yerel arama yapılır. Bu üç yöntemde de örneklem sayısı sonsuza giderken doğru sonuca erişme olasılığı $P \rightarrow 1$ olacaktır. Bu yöntemler diğer raslantısal arama yöntemlerinden basit olmasına rağmen yapılan değerlendirme sayısına göre doğru sonuca erişme olasılığı bakımından verimsizdirler.

Geliştirilen yöntemlerden biri Price [16] tarafından önerilen “Kontrollü raslantısal arama”dır (controlled random search). Bu yöntemde X içerisinde N tane nokta $x_1, x_2, \dots, x_N$ raslantısal olarak seçilir. Bunların en yüksek değere karşılık düşeni f_M , en düşük değere karşılık düşeni f_L belirlenir. Global enbüyük arama probleminde, her

yeni döngüde rasgele bir P sayısı alınır ve f_L ile karşılaştırılır. $f_p > f_L$ ise f_L çıkartılarak yerine f_p saklanır. Bir tabloda tutulacak N tane sayı,

$$N=10(n+1) \quad (2.23)$$

bağıntısından hesaplanır. Burada n, amaç ölçüt fonksiyonunun boyutudur. Her döngüde bu N tane biriktirilen noktadan (n+1) tane farklı nokta seçilir. Bu (n+1) tane noktadan keyfi seçilen bir tanesi, R_{n+1} , simpleks kutbu olarak alınır. Geri kalan n tane noktanın ağırlık merkezi, G'ye göre, R_{n+1} 'in görüntüsü seçilerek bir sonraki deneme noktası bulunur.

$$P=2G-R_{n+1} \quad (2.24)$$

Simpleks noktası $R_1, R_2, \dots, R_{n+1}$ 'den farklı olarak, rasgele seçilmez. R_1 , biriktirilen N tane noktadan en düşük olanına eşittir. Fonksiyonun bu noktadaki değeri f_L olarak kaydedilir. P noktasındaki fonksiyon değeri, f_p , f_L ile karşılaştırılır ve daha büyükse tablodaki eski en küçük çıkartılarak tabloya konur. T önceden belirlenmiş bir hata toleransı olsun. $f_M - f_L < T$ olduğunda rasgele nokta seçimi ve simpleks doğrultusunda ilerlemeden oluşan bu döngü sonlandırılır ve en büyük değer olan f_M ve bu değeri veren M noktası global enbüyük olarak değerlendirilir.

2.3.6 Evrimsel Arama Yöntemleri

Genetik eniyileme algoritmaları doğadaki canlı türlerinin evrimleşmesi örnek alınarak ortaya atılmıştır. Bu nedenle evrimsel arama yöntemleri grubuna girerler. Tek bir başlangıç noktası yerine nesil adı verilen ve başlangıçta rasgele belirlenen bir bireyler kümesi ile çalışılır. Aslında her bir nesil optimize edilecek parametre kümesini temsil etmektedir. Bu arama yöntemi üç temel işlem içermektedir. Bunlardan birincisi çoğullamadır. Çoğullama, mevcut nesil özelliklerine bağlı olarak yeni nesil oluşturma işlemidir. Çoğullama sırasında bireyler amaç ölçüt

fazla aktarılması olasılığı yüksek olacaktır. Genetik algoritmanın ikinci aşaması çaprazlamadır. Çaprazlama sırasında n tane bireyin parametrik özellikleri yine rasgele seçilmiş n tane birey ile yer değiştirilir. Örneğin 5 tane parametreden oluşan özellik grubu var ise 2. bireyin ilk 3 parametresi ile 3. bireyin son iki parametresi yer değiştirilerek yeni nesil yaratılır. Genetik algoritmanın üçüncü aşaması olan mutasyon her zaman gerçekleşmeyebilir. Mutasyon, rasgele bir anda birey özelliklerinden en az birinin rasgele değişime uğramasıdır. Diğer bir deyişle, mutasyon gerçekleşmemişse, çaprazlama sonrası tekrar çoğullama işlemine dönülerek elde edilen nesil doğal seleksiyona tabi tutulur. Mutasyon gerçekleşiyorsa, çoğullamadan önce bireylerin özelliklerinden en az biri rasgele değiştirilir. Mutasyona uğramış nesillerden amaç ölçütüne göre kötü sonuç verenleri doğal seleksiyonlar sırasında kaybolacaktır. İyi sonuç veren özellikler ise yine belirli olasılıklar dahilinde yeni nesillere aktarılacaktır. Matematiksel yönden inceleme yapılırsa, çoğullama ve çaprazlamadan oluşan adımların, bireyleri, yerel eniyi noktalara ulaştırdıkları görülmektedir. Mutasyonlar sonucu daha iyi özelliklere ulaşabilen bireyler, yeni yerel eniyi noktalara geçişi sağlamaktadırlar. Böylelikle genetik algoritma ile yapılan eniyileme, arama aralığında global geçerli olacaktır. Genetik algoritma işletiminde ayrıca bir ön bilgi ihtiyacı olmayışı bir üstünlüktür. Ancak yeteri doğrulukla çözüme ulaşılması için çok sayıda birey üzerinde işlem yapılması gerekliliği, algoritmanın işletim süresinin artmasına neden olmaktadır.

2.4 Global Eniyileme Yöntemlerinin Değerlendirilmesi

2.3 ayrıtında açıklanan yöntemlerin bir ya da birden fazlasını kullanan çok sayıda global eniyileme algoritması geliştirilmiştir. Bu yöntemlerden probleme uygun olanını seçerken ya da başarılarını karşılaştırırken bazı belirli özellikleri göz önünde tutmak gerekir. Bu özellikler, ön bilgi gereksinimi, çözüm aralığı, kısıtlamalar, etkinlik, kararlılık ve sonlandırma kriterleri olarak sıralanabilir.

2.4.1. Ön Bilgi Gereksinimi

Genellikle aynı türden ön bilgi kullanan algoritmalar beraber değerlendirilebilir. Örneğin bir Lipschitz değerinin var olması ve bilinmesinin gerek koşul olduğu bir algoritma, bu tür bir ön bilgi kullanmayan algoritmadan daha hızlı ya da daha güvenli olarak doğru çözüme ulaşabilir, fakat çoğu amaç ölçüt fonksiyonu için bu koşulu

arama öncesi vermek mümkün olmaz. Aynı önbilgi gereksinimleri ve daha az sayıda algoritma çevrimi ile aynı sonuca erişen, ya da daha az ön bilgi ve yakın hesaplama sayıları ile aynı sonuca erişen sayısal hesaplama algoritmaları daha başarılı bulunur.

2.4.2 Çözüm Aralığı ve Kısıtlamalar

Yerel eniyilerin sayısı bilinmediği durumda, $X=[a,b]=[-\infty,+\infty]$ aralığında çözümü garanti eden global eniyileme yöntemi yoktur. Genellikle yerel ekstremumların sayısı önceden bilinmediğinden, algoritmalar, önceden belirli ve sabit $[a,b]$ sınırı için tasarlanmıştır. n değişkenli bir amaç ölçüt fonksiyonunda değişken sınırları,

$$\begin{aligned} a_1 &\leq x_1 \leq b_1 \\ a_2 &\leq x_2 \leq b_2 \\ &\dots\dots\dots \\ a_n &\leq x_n \leq b_n \end{aligned} \quad (2.25)$$

şeklinde verilir. X , uygun çözüm kümesi değişken aralık değerleri ile sınırlanan bölgenin bir alt kümesidir. $f(x_1, x_2, \dots, x_n)$ için, eşitlik ya da eşitsizlik türünden ek kısıtlamalar getirilmediği durumda X , (2.25) ile ifade edilen alan tarafından birebir örtülecektir. Fonksiyonel kısıtlamalar n adet eşitsizlik ve $(m-n)$ adet eşitlik için,

$$\begin{aligned} g_1(x_1, x_2, \dots, x_n) &\leq 0 \\ g_2(x_1, x_2, \dots, x_n) &\leq 0 \\ &\dots\dots\dots \\ g_n(x_1, x_2, \dots, x_n) &\leq 0 \\ g_{n+1}(x_1, x_2, \dots, x_n) &= 0 \\ &\dots\dots\dots \\ g_m(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad (2.26)$$

şeklinde ifade edilir. Global eniyileme yöntemlerinden birçoğu fonksiyonel kısıtlamalar için ayrıca işlem yapmazlar. Bu durumda kısıtlamalar ceza fonksiyonu yöntemi kullanılarak $f(x_1, x_2, \dots, x_n)$ amaç ölçüt fonksiyonunun içine katılır.

2.4.3 Çözünürlük

X uygun çözüm kümesi $[a,b]$ aralığında tanımlanmış olsun. Hassasiyetin tam ondalık katlar olması halinde (0.0001, 0.0000001 vb), hesaplanan eniyi noktanın ($x^* \in X$) hane sayısı cinsinden doğruluğunu, r , çözünürlüğü (resolution) vermektedir. n , x^* sayısında noktanın sağ tarafında kalan, anlamlı ondalık hanelerin sayısını gösterebilir. Bu durumda çözünürlük,

$$r=h \cdot (b-a) \cdot 10^n \quad (2.27)$$

olacaktır. Benzer şekilde hassasiyet yarım ondalıklar (0.005, 0.000005 vb) şeklinde olması halinde (2.27)'deki $h=0.5$ alınmalıdır. Buradaki n hane sayısı, toplam hata üst sınırı ile ters orantılıdır. Çözünürlük, bu şekliyle daha sonra etkinlik tanımında kullanılmak üzere tarafımızca tanımlanmıştır. Yöntem hataları ve yuvarlatma hataları hata üst sınırını belirlemektedir. Yöntem hatalarından gelen hata bileşeni özellikle raslantısal arama yöntemlerinde doğruluk olasılığı ile birlikte verilir. Kesinlik söz konusu değildir. Yuvarlatma hataları ise işlemlerin yapıldığı elektronik hesaplayıcı ortamda (bilgisayar vb.), sayıların ifade hassasiyeti ile orantılıdır. İşlem yapılan rakamlar, işlemin yapıldığı makinede belirli bir bit sayısı ile ifade edilmektedir. Bu nedenle aritmetik dört işlem ve karşılaştırma gibi temel işlemlerde, rakamlar ifade edildikleri hassasiyet kadar işlem görürler. Tanımlanmayan küsürat, tanımlı en küçük değerli hane içine yuvarlatılır. Standartlara uygun bir elektronik hesaplayıcı, IEEE754'te belirtilen sayı hassasiyetlerine uygunluk gösterir [17]. IEEE754'e göre, sayısal hesaplamada sıkça kullanılan 32 ve 64 bit sayılar tablo 1'de görüldüğü şekilde tanımlanmıştır. Her iki sayı ifadesinde de 1 bit işaret için (-/+) kullanılmaktadır.

Tablo 2.1 : IEEE754'e göre 32 ve 64 bit sayı ifadeleri

Sayı tipi	Genişlik	Mantis	Üstel	Birim yuvarlatma
Tek hassasiyet	32bit	23+1 bit	8bit	$2^{-24} \approx 5.96 \times 10^{-8}$
Çift hassasiyet	64bit	52+1 bit	11bit	$2^{-53} \approx 1.11 \times 10^{-16}$

Hata üst sınırı, duyarlılık ve işletim hızı gereksinimlerine bağlı olarak kullanıcı tarafından belirlenerek algoritma işletimi öncesi, parametrik olarak da girilebilir. Makinanın hesap hassasiyetinden daha kaba olan bu rakam hata üst sınırını belirler.

2.4.4 Etkinlik

Yerel eniyileme algoritmalarında bir yakınsama hızı vermek mümkün olmasına rağmen, global eniyileme problemlerinde genel olarak düzgün bir yakınsama söz konusu değildir. Bunun yerine belirli bir çözünürlük için sonuca erişim süresinden ya da işlem sayısından söz etmek mümkündür. k çevrimde r çözünürlükle sonuca erişen bir algoritmanın etkinliğini,

$$\eta = \frac{r}{k} \quad (2.28)$$

olarak ifade etmek mümkündür. Ancak algoritmaların bir çevrimi sırasındaki işlem sayıları birbirinden çok farklıdır. Bu sebeple algoritmaları karşılaştırırken (2.28)'deki k değerini normalize etmekte fayda vardır. Algoritmanın çözüme erişim hızı sonlandırma kriterine bağlı olduğundan ilk defa bu tez çalışmasında bir etkinlik ifadesi tanımlanmıştır.

2.4.5 Kararlılık

Algoritmalar, kapalı çevrimli kontrol sistemlerine benzer dinamik davranış gösterirler. Çözüm noktası civarında, ya da bazı ara değerler civarında aynı ya da artan genlikli, ileri/geri adımlamalar oluşabilmektedir. Salınımların olduğu durumda ortalama ilerleme sıfır civarındadır. Bu tür kararsızlıklar algoritmanın işlem sayısında gereksiz artışlara sebep olabildiği gibi doğru sonuca erişmeyi de engellemektedir. Adımlar halinde ilerlenen algoritmalarda, fonksiyonun gradiyentinin büyük olduğu bölgelerde, atılan adımın genişliği büyük olduğu takdirde hedef noktanın ilerisine düşülecektir. Geri adım atıldığı sırada da aynı olay ters yönde yaşanabilmektedir. Bu da belirli bir nokta civarında lineer olmayan sistemlerde bilinegelen sınır çevrimine (limit cycle) sebep olmaktadır. Çözüm noktasına erişebilmek için algoritmanın parametreleri el ile (manuel) ya da kendiliğinden (otomatik) ayarlanmalıdır. Raslantısal arama yöntemlerinde genellikle bu tür sorunlarla karşılaşılmaz.

2.4.6 Sonlandırma Kriteri

Global eniyilemede önemli sorunlardan biri de ulaşılan noktanın global eniyi olup olmadığının sınılanmasıdır. Özellikle rasgele örnekleme üzerine kurulu yöntemler için belirli bir sınıma tekniği de mevcut değildir. Bu tür yöntemler önceden belirli bir değerlendirme sayısına ulaşıldığında elde edilen sonucun aranılan nokta olduğunu varsayarlar. Alan daraltma üzerine kurulu yöntemlerde, inceleme aralığı istenilen çözüm hassasiyetinin altında kalıncaya kadar işleme devam edilir. X 'in tanım aralığını örterek alt parçalarda arama yapan yöntemlerde, tüm tanım aralığının değerlendirilmesi durumunda işletim durdurulur. Yerel ekstremumların sayısının belirliliğini kullanan yöntemlerde, tüm ekstremumlara ulaşıldığında aramaya son verilir. Adımlayarak $[a,b]$ aralığını tarayan yöntemlerde başlanılan noktaya göre ters tarafta kalan sınıra ulaşıldığında aramaya son verilir ya da arama aralığını daraltarak işleme devam edilir.

Bu bölümde global eniyileme problemi ana hatlarıyla açıklanmıştır. Kullanılabilmiş yöntemler, arama stratejileri bakımından kabaca sınıflandırılmaya çalışılmış ve yöntemlerin değerlendirilmesinde göz önünde tutulacak etkenler ele alınmıştır. Beşinci bölümde açıklanan global eniyileme yöntemi sayısal arama tekniği bakımından, arama aralığını örten yöntemler sınıfına girmektedir. 2.3.2 ayrıtında anlatılan Evtushenko'nun yönteminden farkı yeni değerlendirme noktasının hesaplanmasında kontrol kuramı yaklaşımının kullanılmasıdır.

3. KONTROL KURAMININ ENİYİLEMeye UYGULANMASI

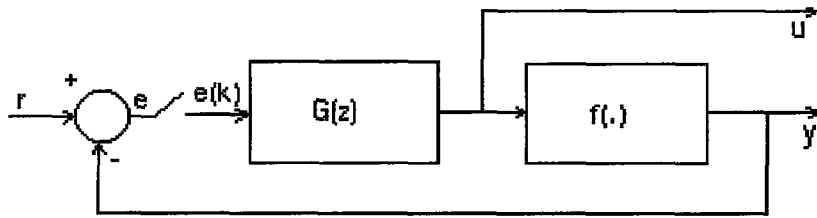
Kök arama ve eniyileme en çok karşılaşılan iki çeşit sayısal arama problemidir. Ardışıl bir arama mekanizması ile belirli bir fonksiyonu sıfıra götüren değişkenlerin bulunması işlemine sayısal kök arama denir. Sonuca erişimdeki işlem sürecini belirleyen ardışıl arama mekanizması sayısal arama algoritması olarak adlandırılır. Kök arama algoritmaları genellikle bir önceki denemede elde edilen sonuçlara bağlı olarak değişken değerlerindeki değişim miktarlarını üretirler. Değişken değerlerindeki artma ya da azalma miktarı literatürde adımlama olarak bilinir. 2.bölümde değinildiği üzere adım boyunun belirlenme şekli algoritmalar arasında farklılık göstermektedir. Ancak hesaplanan adıma göre yeni deneme değerlerinin belirlenmesi genellikle bir önceki değişken değerlerine değişim miktarının eklenmesi şeklinde olur (örneğin, gradyent yönünde yaklaşma, Newton-Raphson vb. yöntemler). Bu ekleme işlemini ayırık integrasyonla ifade etmek mümkündür. Kapalı çevrim kontrol kuramı ile bir benzetmeye gidilirse, fonksiyonun bir önceki denemedeki çıkış değerine göre değişkenlerde yapılması gereken düzeltme miktarını (adımlama) belirleyen ayırık matematiksel işlemi ve integrasyonu bir kontrolörle, fonksiyonu değişken kazanç elemanına eşdeğer bir sistem ile, fonksiyon çıkışını ise hata işareti ile ifade etmek gerekir. Negatif birim geribeslemeli kapalı çevrimli bir kontrol düzeninde referans işaret olarak sıfır alınırsa, uygun tasarlanmış bir kontrolör, sistem çıkışını sıfıra götürecektir. Sayısal bir kök arama yöntemi, yukarıda açıklanan benzeşim yolu ile kapalı çevrimli bir kontrol düzeni şeklinde modellenirse, sıfır referans için, kontrolör (adım üretici ve integratör) fonksiyon çıkışını sıfır yapan değişken değerlerini verir. Bu benzeşim bakımından yerel eniyileme probleminin sayısal kök aranmasından farkı, sistem bloğu yerine fonksiyonun türevinin (ya da eşdeğeri bir fark ifadesinin) bulunmasıdır. Bu durumda kontrolör, fonksiyonun türevini sıfır yapan değişken değerlerini üretecektir. Erişilen uç değer (ekstremumun) enbüyük ya da enküçük olması o noktadaki ikinci türevin işaretine bağlıdır.

Sayısal arama yöntemlerinin kapalı çevrimli kontrol düzenleri şeklinde ifade edilebilmesinden iki şekilde yararlanılabilir. Birincisi, sistemlerin denetiminde kullanılan ayırık kontrolörlerden yeni kök arama ya da yerel eniyileme algoritmaları türetilir. İkinci olarak, kapalı çevrimli sayısal kontrol sistemleri için geliştirilmiş yöntemler, mevcut arama algoritmalarının kararlılık analizinde kullanılabilir.

Bu bölümde PI tipi bir kontrolör kullanılarak sayısal kök arama ve yerel eniyileme problemlerinin çözümü anlatılmıştır. Dördüncü bölümde kararlılık sorunlarına değinilmiştir. Beşinci bölümde ise kapalı çevrimli bir kontrol düzeni ile kök arama ve yerel eniyileme işlemlerinden faydalanan yeni bir global eniyileme yöntemi açıklanmıştır.

3.1. Kapalı Çevrim Kontrol Yöntemi İle Kök Arama

Şekil 3.1’de birim geribeslemeli bir ayırık kontrol sisteminin blok yapısı görülmektedir. Burada r , referans işaret, $e(k)$, k . değerlendirmedeki hata işareti, G , kontrolör transfer fonksiyonu, $f(.)$ ise kontrol edilmek istenen sistemi temsil eden ayırık diferansiyel denklem takımıdır (ya da tek denklem). $f(.)$ diferansiyel denklem takımı yerine bir $f(x)$ fonksiyonu konulması halinde, kontrol kuramı kullanılarak, (2.1) ile tanımlanan sayısal eniyileme problemine aynı blok diyagramı ile yaklaşılabilir. Bu blok diyagramda, u , fonksiyonun giriş değeri, y ise çıkış değeridir.



Şekil 3.1 Sayısal arama probleminin, sayısal bir otomatik kontrol sistemi olarak değerlendirilmesi.

Kontrol kuramı gereği, uygun seçilmiş bir G kontrolörü, sürekli halde e hata işaretini sıfıra götürür. Diğer bir bakış açısıyla, k , adım sayısı, $r(k)=r$ sabit olmak üzere,

$$k \rightarrow \infty \text{ için } e(k) \rightarrow 0, f(u(k)) \rightarrow r \quad (3.1)$$

ya da, $*$ sürekli hal çözümünü ifade etmek üzere,

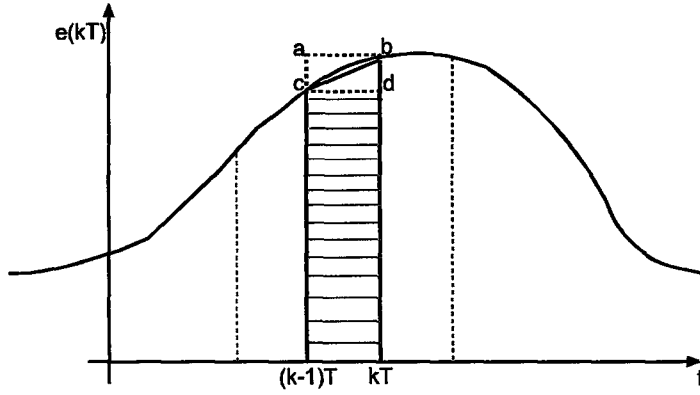
$$(f(u) - r) \Big|_{k \rightarrow \infty} \rightarrow 0 \Rightarrow u^* = x^*, f(x^*) = r \quad (3.2)$$

ifadeleri geçerlidir.

Buna göre, u kontrolör çıkışı, $f(x) - r = 0$ denkleminin çözümünü verir. Özel olarak referans işaretinin $r = 0$ olarak verilmesi durumunda, sürekli halde u 'nun son değeri, $f(x)$ fonksiyonunun bir köküne karşı düşer. Bu kontrol kuramı yaklaşımı, Gustafson ve Astrom tarafından diferansiyel denklemlerin sayısal çözümlerinde adım boyunun hesaplanmasında kullanılmıştır [18]. Çalışmamızda ise bu yaklaşım, ilk kez global eniyileme probleminin çözümünde bir fonksiyonun köklerinin belirlenmesinde ve yerel eniyilemede kullanılmıştır.

Burada bu amaca yönelik olarak, $f(.)$ yerine, herhangi bir tek veya çok değişkenli fonksiyon, G yerine ise uygun seçilmiş bir kontrolör yer almaktadır.

Basit yapısı ve integrasyon içermesi bakımından ilk olarak PI (oransal integral) tipi bir kontrolör ile uygulama yapılmıştır. PI tipi kontrolör sayısal kök arama ve yerel eniyileme problemlerine uyarlanmaktadır. PI tipi bir kontrolör hata işaretini iki paralel kol üzerinden değerlendirir. Bunlardan birinde hata işareti bir K_p katsayısıyla çarpılır. Diğerinde ise K_i kazancıyla orantılı olarak integre edilir. Bu iki kolun çıkış işaretleri toplanarak kontrolör çıkışı olarak sisteme uygulanır. Kendi içerisinde integrasyon olmayan sistemlerin denetiminde, kontrolördeki integrasyon işlemi sürekli hal hatasının giderilmesini sağlar. Oransal kontrol bileşeni, hataya bağlı olarak gecikmesiz cevap verdiği için yerleşme süresini kısaltmaktadır. Yerleşme süresini küçültmek için K_i 'yi arttırmak çoğu zaman salınımaya yol açar. Yalnızca oransal kontrol kullanarak sürekli hal hatasını gidermek mümkün olmaz.



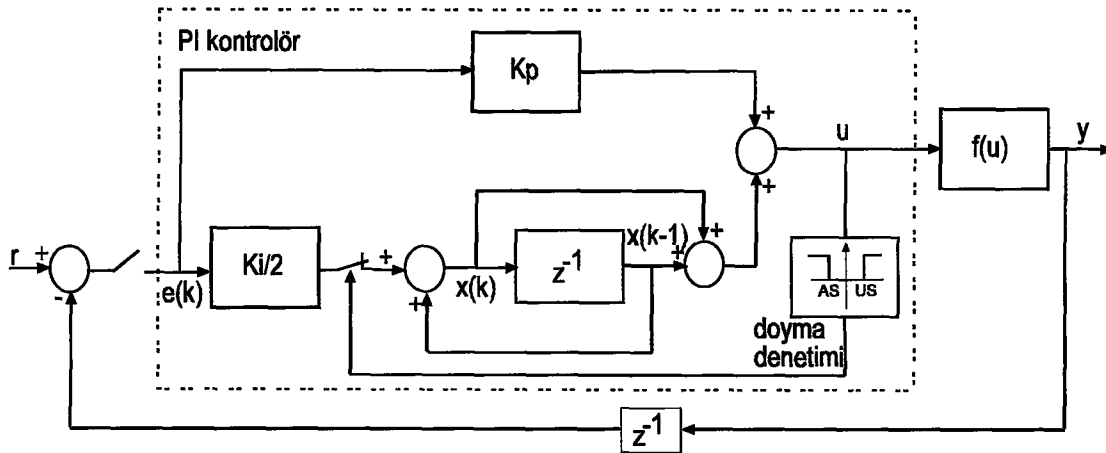
Şekil 3.2 Ayırık zamanlı bir kontrol sisteminde yamuksal integrasyon.

Ayrık zamanlı bir kontrol düzeninde yaygın olarak kullanılan bir yaklaşık integrasyon yöntemi yamuksal hesaplamadır. Bu çalışmada sayısal hesaplama hatası dürtgensel integrasyona göre az olduğu için yamuk yaklaşıklığı tercih edilmiştir. Bu yöntemde iki ardışıl örnekleme değerinin bir doğru ile birleştiği varsayılmaktadır. Şekil 3.2’de T periyodu ile örneklenmiş bir hata işaretinin yamuksal integrasyonu görülmektedir. $\overline{cb}$ doğru parçasının altında kalan alan z tanım bölgesinde yamuksal olarak hesaplanmak istenirse $\overline{ab}$ ve $\overline{cd}$ doğru parçalarının altında kalan alanın dürtgenselliğinden faydalanılabilir. Bu iki integralin farkının yarısına $\overline{cd}$ ’nin altında kalan dürtgenin alanı eklenirse $\overline{cb}$ ’nin altında kalan alan yamuksal yaklaşıklıkla hesaplanmış olur. Buna göre örneklenmiş hata işaretinin integrali z tanım bölgesinde,

$$\begin{aligned} Z\left\{K_i \int e(t)dt\right\} &= \frac{K_i T}{z-1} E(z) + \left(\frac{K_i T z}{z-1} - \frac{K_i T}{z-1}\right) \frac{E(z)}{2} \\ &= \frac{K_i T (z+1)}{2 (z-1)} \end{aligned} \quad (3.3)$$

olarak hesaplanır [19].

(3.3)'de hesaplanan integrasyon işlemi ile birlikte PI tipi sayısal bir kontrolör, $T=1s$ için şekil 3.3'de görüldüğü şekilde programlanır. Kontrolör çıkış işareti doyma sınırına ulaştığı andan itibaren hata işaretinin integre edilmemesi için işaretin integratöre girişi engellenir (anti-wind up). Böylece, u kontrolör çıkışının değişmediği sınır durumunda integral iç değişkeninin artması engellenmiş olur. Bu önlemin alınmaması halinde hata işareti doymaya sebep olmayacak bir değere geri geldiğinde integral iç değişkeni daha geç azalacağından gecikme ve kararlılık sorunlarıyla karşılaşmaktadır. Bu çalışmada $T=1s$ alınmasının tek nedeni hesaplama kolaylığıdır. Tanımı gereği, sistemlerden farklı olarak, matematiksel bir fonksiyon belirli bir giriş değeri için daima aynı çıkışı verir. Oysa sistemlerin çıkış değerleri sadece anlık girişe değil aynı zamanda kendi iç durumlarına da bağlıdır. Diğer bir deyişle belirli bir süre ile sabit bir giriş uygulanan (örneğin sıfırcı dereceden tutucu çıkışı) bir sistemin çıkışı, kendi iç durumlarına bağlı olarak değişebilir. Burada, sistem yerine matematiksel bir fonksiyon yerleştirildiğinden, yazılım ile programlanmış eşdeğer PI tipi kontrolörde, T örnekleme periyodu ile integrasyon değeri değişmeyecektir. Bir algoritma çevrimi tamamlandığında bir sonraki çevrim için örnekleme periyodunun beklenmesine gerek yoktur. Bilgisayar algoritmayı ne kadar hızlı işletiyorsa sonuca o oranda çabuk erişilir. Eğer kontrolör analog ortamda, fonksiyon ise sayısal ortamda (girişine A/D ve çıkışına D/A dönüştürücü konularak) gerçekleştirilseydi, kontrolörün içerisindeki integratör sebebiyle T örnekleme periyodunun doğru seçilmesi sonuca erişimde önem kazanacaktı.



Şekil 3.3 PI tipi bir kontrolörle $f(\cdot)$ fonksiyonunun köklerinin bulunması.

Şekil 3.3’de önerilen kapalı çevrim kontrol düzeni ile $f(x)-r=0$ denkleminin çözümü elde edilebilir. Bu çevrimde örnekleme periyodu k adım sayısına karşı düşmekte ve çözüm olarak kontrolör çıkışı değerlendirilmektedir. Kontrol sistemlerinin sayısal eniyileme problemine uyarlanması geribesleme işareti en az bir çevrim gecikmeyle uygulanabilir. Bu nedenle şekil 3.3’deki blok diyagramda geribesleme yoluna z^{-1} konulmuştur. Burada PI tipi (oransal-integral) kontrolör kullanılmıştır. İntegratör, yamuksal integrasyon yöntemi ile gerçekleştirilmektedir. Buna göre bir kontrol çevriminde yapılan işlemler,

$$e(k)=r-y(k-1) \quad (3.4)$$

$$x(k) = \begin{cases} |u| < US \Rightarrow 0.5e(k)K_i + x(k-1) \\ |u| \geq AS \Rightarrow x(k-1) \quad (\text{Doyma denetimi}) \end{cases} \quad (3.5)$$

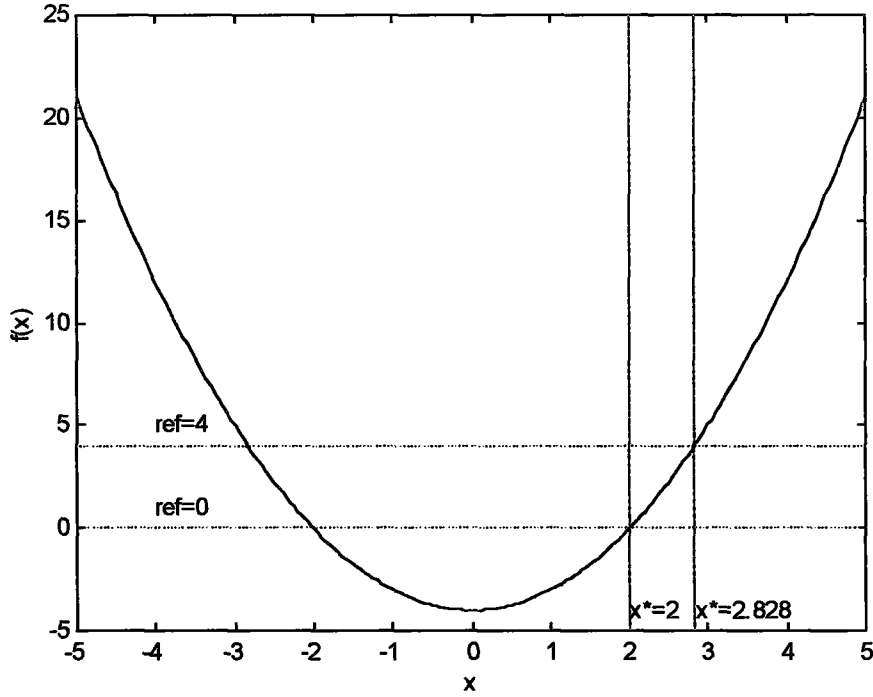
$$u(k)=x(k)+x(k-1)+ K_p e(k) \quad (3.6)$$

$$y = f(u) \quad (3.7)$$

şeklinde ayrı olarak programlanabilmektedir. Yapılan işlemin süresi algoritmanın hesap süresine eşittir. Sistemlerin integral kontrolünde çarpan olarak gelen $1/(T_i T)$ oranı yerine sadece K_i integral kazancı kullanılmıştır. Bu ifadedeki T_i integral zaman sabitini, T ise örnekleme periyodunu göstermektedir. Açıklanan yöntemde belirli bir örnekleme periyoduna gereksinim olmamasına rağmen aradaki benzeşimin görülebilmesi için $K_i/2$ integral kazancındaki 2’ye bölme işlemi matematiksel karşılığında olduğu gibi bırakılmıştır. PI kontrolöre ilişkin K_p ve K_i parametreleri, fonksiyonun değişimine yaklaşık bir üst sınır getirilerek belirlenebilir. Parametre seçimi ile ilgili tartışmalar 4.bölümde verilmiştir.

3.1.1. Tek Değişkenli Bir Fonksiyonun Kökünün Bulunması

Yapılan denemelerde (3.4)-(3.7) ile ifade edilen fark denklemleri IBM-PC tabanlı bir bilgisayarda programlanarak koşturulmuştur. Şekil 3.4’de görülen $f(x)=x^2-4$ fonksiyonunda referans $r=0$ alındığında, öngörülen sayısal arama yöntemi $f(x)=x^2-4=0$ denkleminin bir kökünü hesaplamaktadır.



Şekil 3.4 $f(x)=x^2-4$ fonksiyonunun $(-5,+5)$ aralığında değişimi.

Belirli bir x_0 noktasından aramaya başlamak için $u(0) = x_0$ olacak şekilde integral değişkeninin ilk değerini belirlemek gerekir. Buna göre (3.5) ve (3.6)'dan,

$$k=0 \Rightarrow x(0) = 0.5e(0)K_i + x(-1) \quad (3.8)$$

$$u(0) = x(0) + x(-1) + K_p e(0) \quad (3.9)$$

$e(0)=0$ kabul edilirse, integral iç değişkenine bağlı olarak çıkışın ilk değeri,

$$u(0) = x(-1) + x(-1) = 2x(-1) \quad (3.10)$$

olarak hesaplanır. (3.10) ilişkisi gereği belirli bir $u(0)$ değerinden başlamak için integral değişkeni başlangıç koşulu olarak,

$$x(-1) = 0.5u(0) = 0.5 \cdot x_0 \quad (3.11)$$

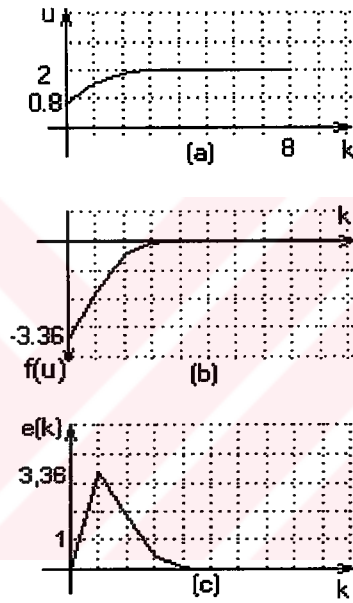
alınmalıdır.

Şekil 3.5-a'da $x_0=0.8$, $r=0$ için kontrol işareti u 'nun değişimi görülmektedir. Şekil3.5-b'deki $f(u)$ 'nun istenen hata sınırı dahilinde referansa eşit olduğu adımdaki u , $f(u)=r$ denkleminin bir köküne karşı düşer. Burada $K_i=0.21$, $K_p=0.1$ alınmıştır.

Bu çözümde sonlandırma kriteri olarak,

$$|e(k)| < E_m \quad (3.12)$$

kullanılmaktadır. Buradaki E_m , çözümde izin verilen hatanın üst sınırıdır.



Şekil 3.5 $f(x)=x^2-4$, $r=0$, $x_0=0.8$ için a) $f(x)=r$ çözümünde kontrol işaretinin (çözüm) değişimi b) Fonksiyon çıkışı c) Hata işaretinin değişimi.

Tablo 3.1'de görüldüğü üzere $E_m=0.001$ için 5 özyinelemede sonuca erişilirken $E_m=0.000001$ için 8 özyinelemede işlem bitirilmekte ve $x=2.000000$ 'daki köke ulaşılmaktadır $E_m=0.000001$ seçilmesi, çözüm hatasının $-0.0000009 \leq e \leq +0.0000009$ olması anlamına gelmektedir.

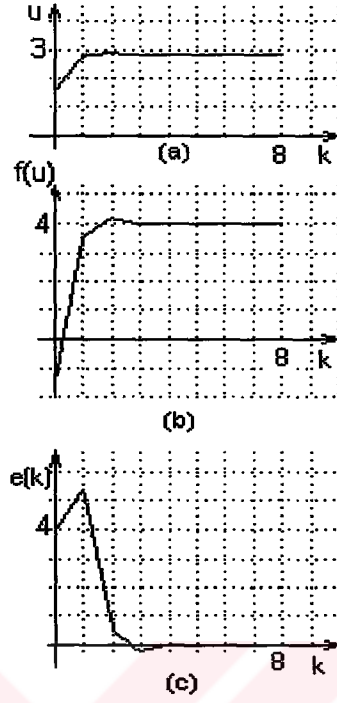
Tablo 3.1- PI kontrolörle $f(x)=x^2-4=r=0$ çözümünde hata işaretine göre kontrolörün aldığı çıkış değerleri

Değerlendirme k	Kontrolör çıkışı u(k)	Değerlendirilen hata işareti e(k)
0	0.8	0
1	1.4888	3.36
2	1.871212	1.783475
3	1.982335	0.4985643
4	1.999249	0.0070334
5	2.000216	0.000300208
6	2.000054	-0.000086598
7	2.000005	-0.000021648
8	2.000000	-0.000002008

Kullanılan çözüm tekniğinde referans işaretin değiştirilerek sıfır dışında bir başka sabit değere getirilmesi durumunda, yeni r değeri için $f(x)-r=0$ denkleminin çözümü elde edilecektir.

$r=4$ alınarak $f(x)=x^2-4=4$ denkleminin $x_0=0.8$, $K_i=0.21$, $K_p=0.1$ için çözdürülmüş ve şekil 3.6'da görüldüğü üzere 8 özyinelemede $x=\sqrt{8}=2.828428$ 'deki köke ulaşılmıştır. Burada (3.12)'deki sonlandırma kriteri uygulanarak $E_m=0.00001$ seçilmiştir.

Şekil3.6-b'de görüldüğü üzere fonksiyon çıkışı kontrol işaretine bağlı olarak, kontrol sistemlerindeki benzer bir davranış göstermektedir. Şekil3.6-a'da görülen kontrolör çıkışı $x_0=0.8$ değerinden başlayarak $x=2.828428$ 'deki köke 8.özyinelemede erişmektedir. Benzer şekilde şekil 3.6-c'de $e(k)$ değişimi ve tablo 3.2'de arama sırasında hesaplanan değerler görülmektedir.

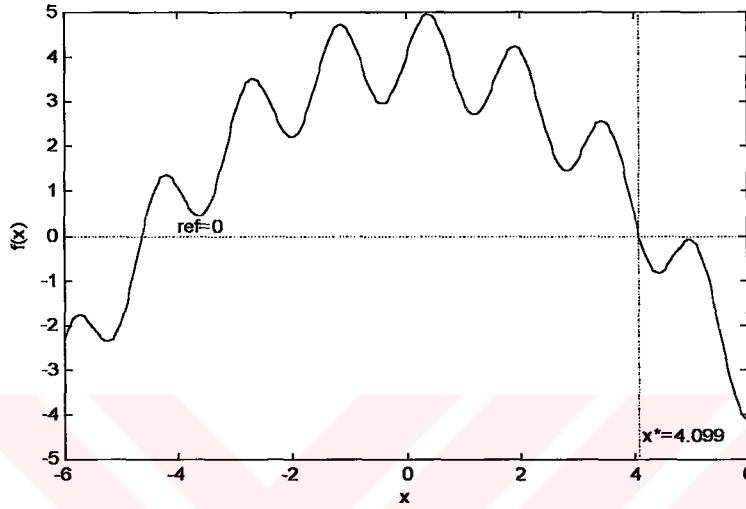


Şekil 3.6 $f(x)=x^2-4$, $r=4$, $x_0=0.8$ için a) $f(x)=r$ çözümünde kontrol işaretinin (çözüm) değişimi b) Fonksiyon çıkışı c) Hata işaretinin değişimi.

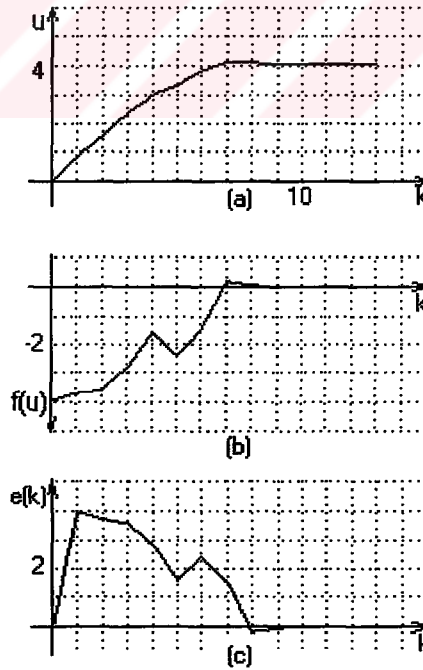
Tablo 3.2 - $x_0=0.8$ için $x^2-4=4$ çözümünde hata işareti ve PI tipi kontrolör çıkışı

Değerlendirme	Kontrolör çıkışı	Değerlendirilen hata işareti
0	1.62	4
1	2.741998	5.3756
2	2.867473	0.4814472
3	2.82427	-0.2229738
4	2.827972	0.002349615
5	2.828617	0.000257349
6	2.82841	-0.000107622
7	2.828424	0.0000098705
8	2.828428	0.0000016451

Kapalı çevrim kontrol tekniğinin sayısal analizde kullanımında söz konusu fonksiyon lineer olmayan bir kazanç bloğu olarak değerlendirilmektedir. Fonksiyonun yapısal karmaşıklığı yöntemi etkilememektedir. Şekil 3.7’de, üzerine sinüsoidal bileşen bindirilmiş bir parabol olan $f(x)=-0.2x^2+4+4\sin(x)$ fonksiyonu görülmektedir. $x_0=0$ ’dan başlanarak $K_i=0.21$, $K_p=0.1$, $r=0$ için $f(x)=0$ denklemi çözdürülmüştür.



Şekil 3.7 $f(x)=-0.2x^2+4+4\sin(x)$ fonksiyonunun $[-6,+6]$ aralığında değişimi.

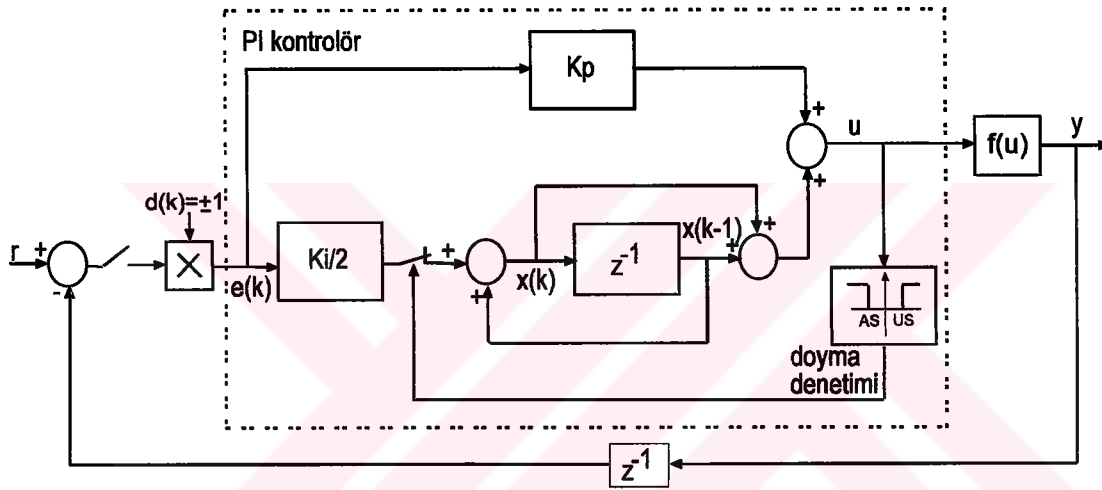


Şekil 3.8 $f(x)=-0.2x^2+4+\sin(4x)$, $r=0$, $x_0=0$ için a) $f(x)=r$ çözümünde kontrol işaretinin (çözüm) değişimi b) Fonksiyon çıkışı c) Hata işaretinin değişimi.

Şekil 3.8-a' da görüldüğü gibi kontrolör çıkışı u , 18.özyinelemede $E_m=0.00001$ kriterine göre $x=4.099979$ 'daki köke ulaşmaktadır.

3.1.2 İstenilen Bir Doğrultuda Kök Aranması

Burada açıklanan yöntemin bir algoritma içerisinde kullanılabilmesi için, kök arama yapılan doğrultunun, kontrol edilebilmesi gerekmektedir. Aksi durumda kontrolör çıkışı ilk özyinelemedeki hata yönüne bağlı olarak kök aramaktadır. İstenilen bir yöndeki kökün bulunabilmesi amacı ile kontrolör girişinde bir yönlendirme yapılmıştır.



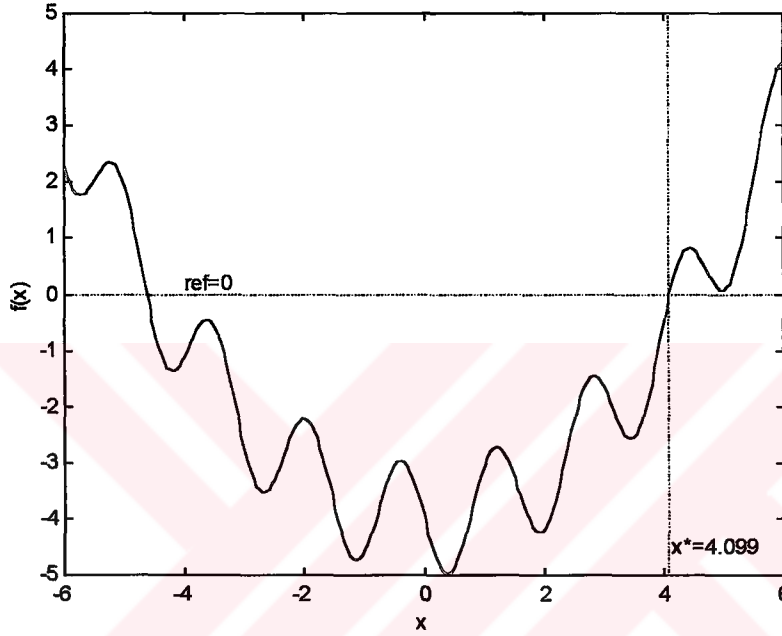
Şekil 3.9 İstenilen işaret yönünde arama için sisteme $d(k)$ doğrultu çarpanı eklenmesi.

Şekil 3.9'da da görüldüğü üzere kontrolör girişi bir d sayısı ile çarpılarak hata işareti aynalanmaktadır. $d(k)$ sayısı,

$$d(k) = \begin{cases} d(0) = +1, +e(0) \text{ yönünde kök arama} \Rightarrow d(k > 0) = +1 \\ d(0) = -1, -e(0) \text{ yönünde kök arama} \Rightarrow d(k > 0) = -1 \end{cases} \quad (3.13)$$

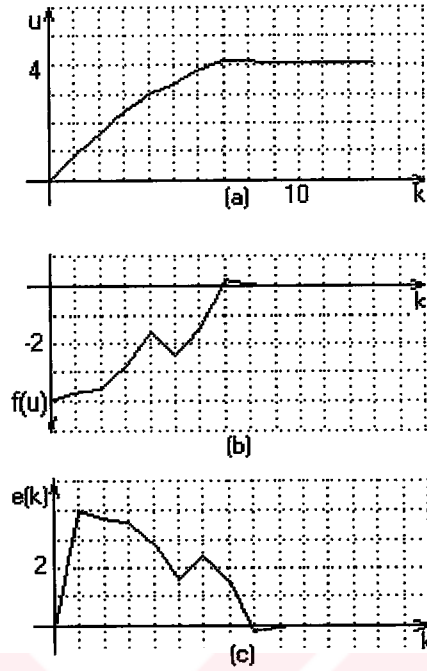
şeklinde belirlenir. Şekil 3.7'deki fonksiyonun ters işaretlisi olan $f(x)=0.2x^2-4-4\sin 4x$ fonksiyonunun $(-6,+6)$ aralığındaki değişimi şekil 3.10'da gösterilmiştir. $r=0$ referansına göre ilk iterasyon sonundaki hata işareti şekil 3.10'daki fonksiyonun tersi yönde olacaktır. Bu durumda kontrolör çıkışı azalan yönde kök arayacaktır. Diğer bir deyişle, $(-)$ işaret yönündeki köke ulaşılacaktır. Oysa $e(0)$ 'ın tersi yönünde, arama

yapabilmek için (3.13)'de tanımlanan d ile çarpılarak şekil 3.7'deki fonksiyon ile aynı yönde arama yapılır hale gelmektedir. Böylelikle arama yapılan doğrultu ilk özyinelemede bir defaya mahsus olarak değiştirilmektedir. Daha sonra böyle bir işlemin yapılması yanlış sonuca varılmasına sebep olabilir. Nitekim kök arama amacı ile oluşturulan kontrol düzenindeki hata işareti, fiziksel kontrol düzenlerindeki benzer şekilde sıfır civarında salınım yaparak da oturabilmektedir.



Şekil 3.10 $f(x)=0.2x^2-4-4\sin 4x$ fonksiyonun $(-6,+6)$ aralığında değişimi.

$f(x)=0.2x^2-4-4\sin 4x$ fonksiyonunun kökleri, $K_i=0.21$, $K_p=0.1$, $x_0=0$, $r=0$ alınarak aratılmış ve $E_m=0.00001$ sonlandırma kriteri ile 13.özyinelemede $x=4.099979$ 'daki köke ulaşılmıştır. Başlangıç özyinelemesinden sonra hata işareti $d=-1$ sabiti ile çarpılmasaydı (-) işaret yönündeki köke ulaşılacaktı. Şekil 3.11'de görüldüğü üzere $d=-1$ için şekil 3.10'daki fonksiyon ile $d=1$ için şekil 3.7'deki fonksiyonun çıkış değişimi ve ulaşılan kökü aynıdır.



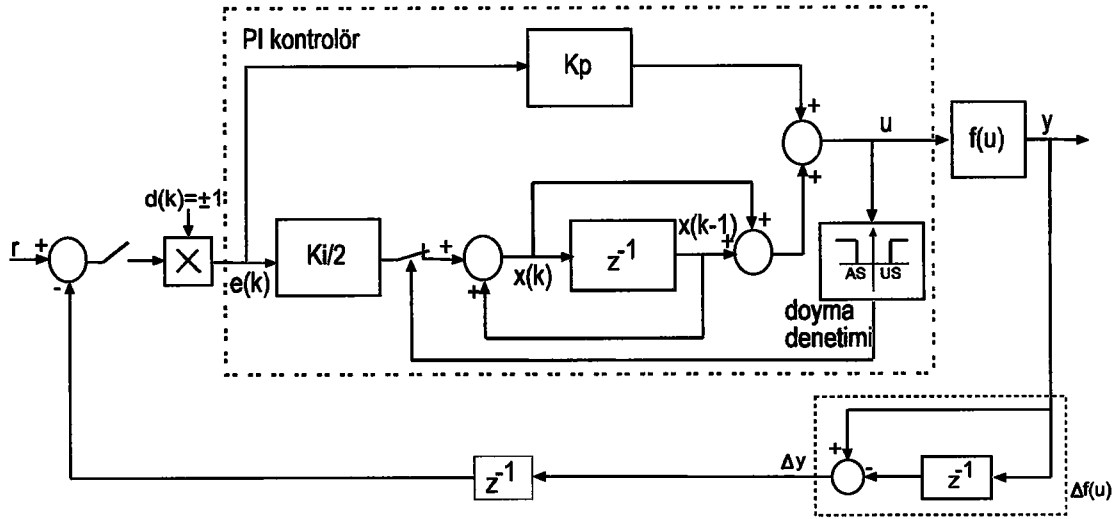
Şekil 3.11 $f(x)=0.2x^2-4\sin(4x)$, $r=0$, $x_0=0$ için a) $f(x)=r$ çözümünde kontrol işaretinin değişimi b) Fonksiyon çıkışı c) Hata işaretinin değişimi.

3.2 Kapalı Çevrim Kontrol Tekniği İle Yerel Eniyileme

3.2.1 Tek Değişkenli Fonksiyonların Yerel Eniyilenmesi

3.1 ayrıtında yer alan örneklerde, fonksiyon, kapalı çevrim kontrol kuramındaki sistem bloğu yerine konularak kök arama işlemi yapılmaktaydı. $f(x)$ fonksiyonun ekstremumlarının aranması ile $f'(x)=0$ denkleminin çözüm noktalarının bulunması aynı anlamı taşımaktadır.

Bulunan bir x^* noktasındaki kökün enbüyük ya da enküçük olması $f''(x^*)$ 'ın işaretine işaretine bağlıdır. Türev alma işlemi sayısal olarak $T=1$ için



Şekil 3.12 Yerel eniyileme için geribesleme yolunda iyileştirme yapılması.

$\Delta f(u) = f(u(k)) - f(u(k-1))$ şeklinde gerçekleşir. (3.1)'deki hata işareti bağıntısına benzer şekilde, kontrol kuramı yaklaşımı ile,

$$\Delta f(u) - r = 0 \quad (3.14)$$

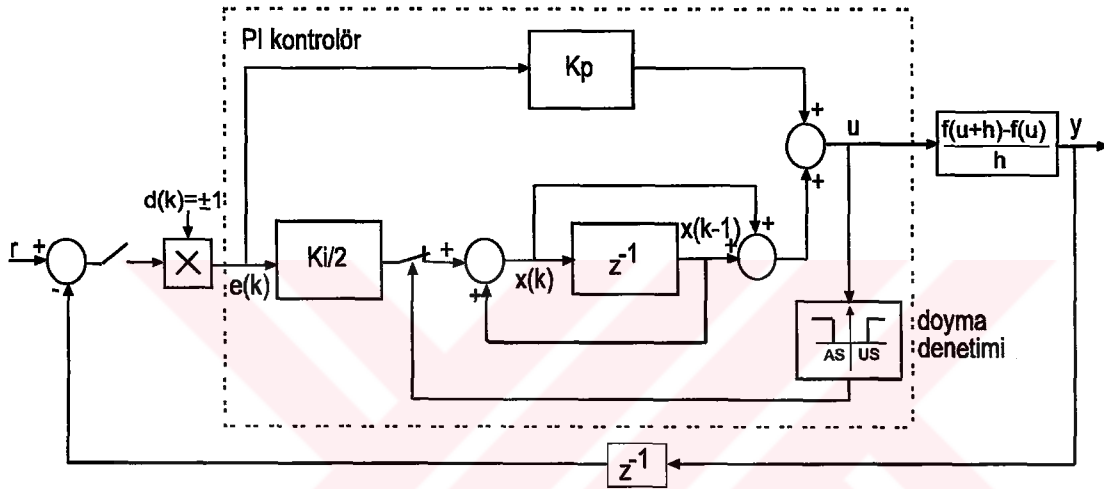
denklemini çözmek üzere geribesleme yoluna geriye fark alma bloğu eklenerek (Şekil 3.12),

$$f(u(k)) - f(u(k-1)) - r = 0 \quad (3.15)$$

fark denklemi çözülebilmektedir. x_1 ve x_2 , $x_1, x_2 \in X$, $f'(x_1) = f'(x_2)$, $x_1 \neq x_2$ olacak şekilde seçilmiş iki tane nokta olsun. İki özyineleme arasındaki adım boyu değişken olduğundan, $u(k) = x_1$ ve $u(k) = x_2$ için, (3.15) de yer alan fark ifadesi değişik değerler verebilmektedir. Kontrol kuramından farklı olarak, diferansiyel bir denklem ile ifade edilen sistem yerine, fonksiyon bloğu yerleştirilmiş olduğundan $\Delta f(u)$ geriye fark ifadesi yerine doğrudan türev ifadesini de kullanmak mümkündür. Yapılan denemelerin tümünde geriye fark ifadesi yerine kontrol işareti civarında sayısal türev alınması özyineleme sayısı ve sonuca erişimdeki başarı bakımından daha iyi sonuç vermiştir. Bu durumda şekil 3.9 daki $f(u)$ yerine,

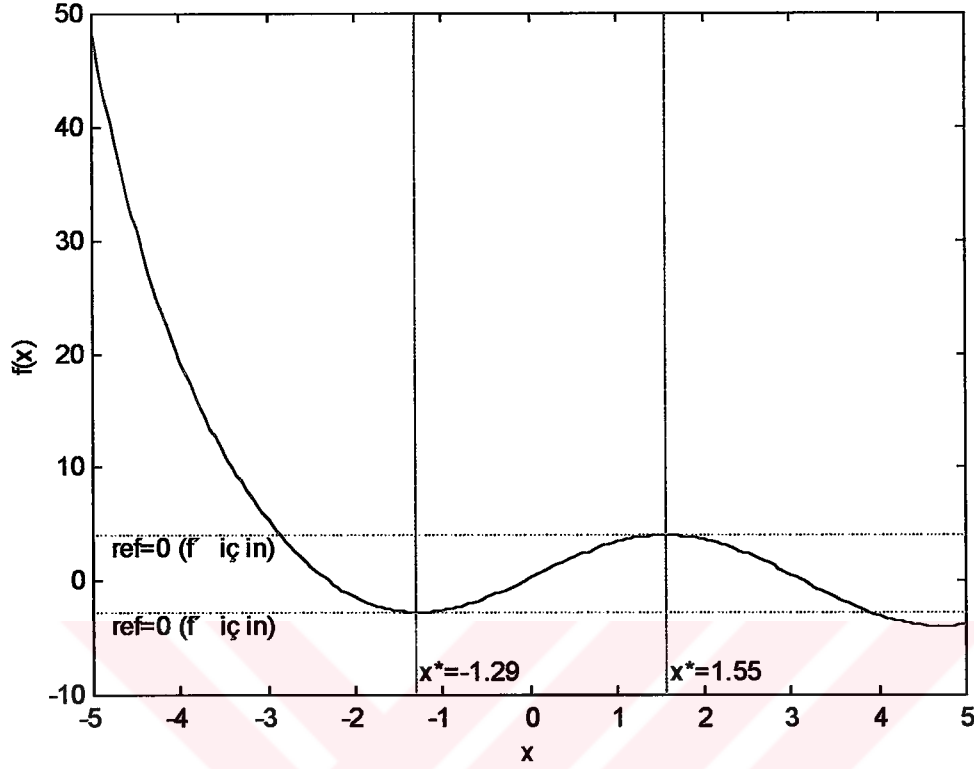
$$f'(u) = \frac{f(u+h) - f(u)}{h} \quad (3.16)$$

kullanılmaktadır. Burada h yeterince küçük seçilmiş bir sayıdır. Yapılan denemelerde $h=0.001$ alınmıştır. $f(u)$ yerine $f'(u)$ kullanmakla $f'(u)-r=0$ denkleminin çözümü elde edilmekte ve r referans işareti eğimi göstermektedir. $r=0$ alınması, arama doğrultusunda $f'(u)=0$ yapan noktanın kontrolör tarafından hesaplanması anlamına gelmektedir.



Şekil 3.13 Tek değişkenli fonksiyonların yerel eniyilenmesi için önerilen kapalı çevrim kontrol düzeni.

$x_0=0$ 'dan başlayarak ve şekil 3.13'deki arama düzeni kullanılarak Şekil3.14'de görülen $f(x)=4\sin(x)+0.3e^{-x}$ fonksiyonunun ekstremumları özyinelemeli olarak hesaplanmıştır. $d(k)$, Doğrultu çarpanının çalıştırıldığı ilk aramada, $f'(x)=0$ denkleminin 0'dan itibaren (+) işaret yönündeki ilk köküne ulaşarak $x^*=1.554448$ noktasına gelinmiştir.

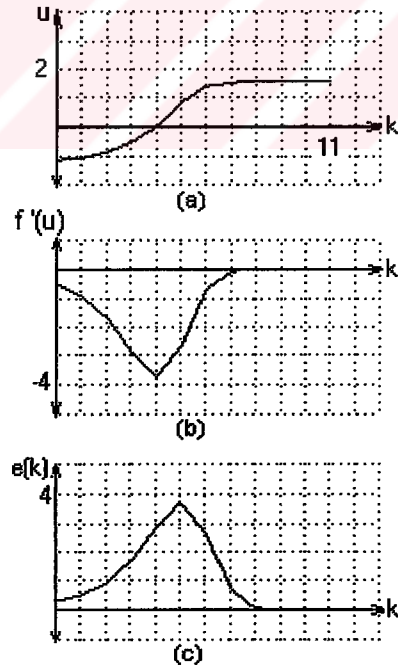


Şekil 3.14 $f(x)=4\sin(x)+0.3e^{-x}$ fonksiyonunun $(-5,+5)$ aralığında değişimi.

İlk koşul işlemlerinin yapıldığı sıfıncı adımdan sonra Tablo 3.3’de görüldüğü gibi 11.adımda oluşan hata, $|e(11)| < E_m = 0.00001$ olduğundan algoritma sonlandırılmıştır. Tablodaki hata değerlerinin aynı adımdaki kontrolör girişi olduğuna dikkat edilmelidir. Bu sebeple 11.adım sonunda oluşan hata tabloda yer almamıştır. (+) işaret yönünde $x^* = 1.554448$ den sonra gelen ekstremuma ulaşmak için $x_0 = 1.554448 + \Delta$ almak gerekir. Buradaki Δ ’nın $f'(x) = 0$ ’ın en yakın iki kökünden daha küçük olması gerekmektedir. Aksi halde arama başlangıç noktası ve yönüne bağlı olarak daha sonraki bir kök hesaplanabilecektir. Kontrolör çıkış işaretinin doyma sınırı, arama sınırına eş olarak seçilirse, kontrolör çıkışının doymaya girmesi, arama yönünde kök bulunamadığı anlamına gelecektir.

Tablo 3.3 - $x_0=0$, $f(x)=\sin(x)+0.3e^{-x}$ için $f'(x)=0$ 'ın (+) işaret yönü çözümünde hata işareti ve PI tipi kontrolör çıkışı

Değerlendirme k	Kontrolör çıkışı u	Değerlendirilen hata işareti e(k)
0	-1.199495	0.2589226
1	-1.10431	0.4580021
2	-0.9182954	0.8962154
3	-0.569582	1.679182
4	0.0209003	2.839446
5	0.7947131	3.705442
6	1.359574	2.665043
7	1.528618	0.7596015
8	1.553237	0.1015663
9	1.554722	0.00476837
10	1.55455	-0.00095367
11	1.554448	-.0000476834



Şekil 3.15 $f(x)=4\sin(x)+0.3e^{-x}$, $K_I=0.21$, $K_P=0.1$, $x_0=0$ için $f'(x)=0$ çözümünde

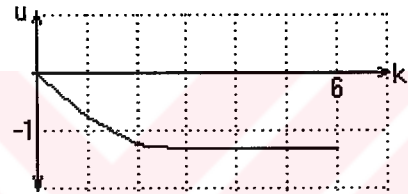
a) Kontrol işareti b) $f'(x)$ aldığı değerler c) Hata işareti.

(3.13)'deki tanımı ile $d(k)$ (+) işaret yönünde kök aranmasına neden olmaktadır. $x_0=0$ 'dan başlanarak (-) işaret yönünde kök aranması istenirse, doğrultu çarpanı $d(k)=1 \forall k$ alınmalıdır. Kapalı çevrim kontrol sistemlerindeki benzer şekilde aşım sonucu hata işareti daha sonra tekrar yön değiştirebilir. Fakat $d(k)$ tanımı (3.13) sadece sıfırcı adımda etkili olduğu için daha sonra, kök civarındaki işaret değişikliklerini etkilememektedir 4.bölümde açıklanacağı üzere global en iyi noktanın bulunmasında bu özellikten yararlanılmaktadır. (-) işaret yönünde kök aranmasında elde edilen hata değerleri ve kontrolör çıkışları Tablo 3.4'de yer almaktadır. Burada görüldüğü üzere, ilk koşul verileri ile atılan sıfırcı adımdan sonraki 6.adımda $E_m=0.00001$ kriterine göre $x^*=-1,294273$ 'deki köke ulaşılmıştır. Toplam 7 çevrimde sonuç elde edilmiştir. Hata kriterini sağlayan e değerinin tabloda gözükmediğine dikkat edilmelidir.

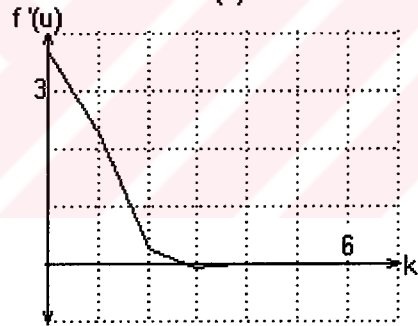
Ekstremum hesabında sayısal türev ifadesi kullanılmasında dikkat edilmesi gereken ikinci bir nokta, h sonlu fark büyüklüğünün değeridir. $E_m=0.00001$ sonlandırma kriterinin belirlediği doğruluk verilen belirli bir h değeri için geçerlidir. Aynı sonlandırma kriteri ile $h=0.1$, $h=0.001$ ya da $h=0.00001$ seçilerek yapılan çözümler farklı sonuçlar elde edilmesine neden olacaktır. h sayısının değişmesi çözümü aranan $f'(x)$ 'in sayısal karşılığının da değişmesi anlamına gelmektedir. Bu sebeple h değeri E_m 'ye yakın seçilmelidir. E_m ve h değeri seçilirken rakam duyarlılığını kaybetmemek bakımından 32bit ya da 64bit sayı ifadelerinden hangisi kullanıldığı dikkate alınmalıdır. Örneğin 32bit işlemlerde, E_m ve h 'nin çok küçük seçilmesi, en küçük anlamlı hanedeki yuvarlatma hatalarından etkilenmeye sebep olmaktadır.

Tablo 3.4 - $x_0=0$, $f(x)=\sin(x)+0.3e^{-x}$ için $f'(x)=0$ 'ın (-) işaret yönündeki çözümünde hata işareti ve PI tipi kontrolör çıkışı

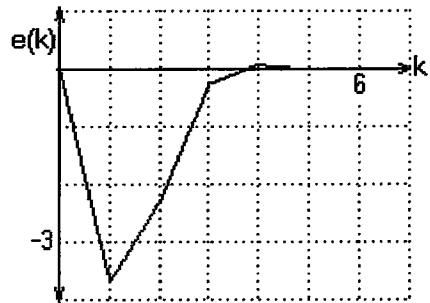
Değerlendirme k	Kontrolör çıkışı u	Değerlendirilen hata işareti e(k)
0	0	0
1	-0.7585281	-3.700137
2	-1.241251	-2.2645
3	-1.305653	-0.2589226
4	-1.295364	0.0565052
5	-1.294006	0.0052452
6	-1.294273	0.0014305



(a)



(b)



(c)

Şekil 3.16 $f(x)=4\sin(x)+0.3e^{-x}$, $K_i=0.21$, $K_p=0.1$, $x_0=0$ için $f'(x)=0$ 'ın (-) işaret yönündeki çözümünde a) Kontrol işareti b) $f'(x)$ aldığı değerler c) Hata işareti.

3.2.2 Çok Değişkenli Fonksiyonların Yerel Eniyilenmesi

$F(x_1, x_2, \dots, x_n)$, n değişkenli bir amaç ölçüt fonksiyonu olsun. Enbüyük ya da enküçük şartının sağlandığı bir noktada,

$$\frac{\partial F}{\partial x_1} = 0, \frac{\partial F}{\partial x_2} = 0, \dots, \frac{\partial F}{\partial x_n} = 0 \quad (3.17)$$

koşulu sağlanacaktır [20]. (3.17) şartını sağlayan bir $(x_1, x_2, \dots, x_n)$ noktasının enbüyük ya da enküçük olması,

$$H = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \dots & \dots & \dots \\ \vdots & & & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \dots & \dots & \frac{\partial^2 F}{\partial x_n^2} \end{bmatrix} \quad (3.18)$$

şeklinde tanımlanan Hessian matrisinin pozitif ya da negatif tanımlı olmasına bağlıdır. Hessian matrisinin pozitif tanımlı olması $(x_1, x_2, \dots, x_n)$ noktasındaki ekstremumun yerel bir enküçük olduğunu gösterir. (3.17)'e bağlı olarak, (3.16)'daki sayısal türev alma işlemi n tane değişken için,

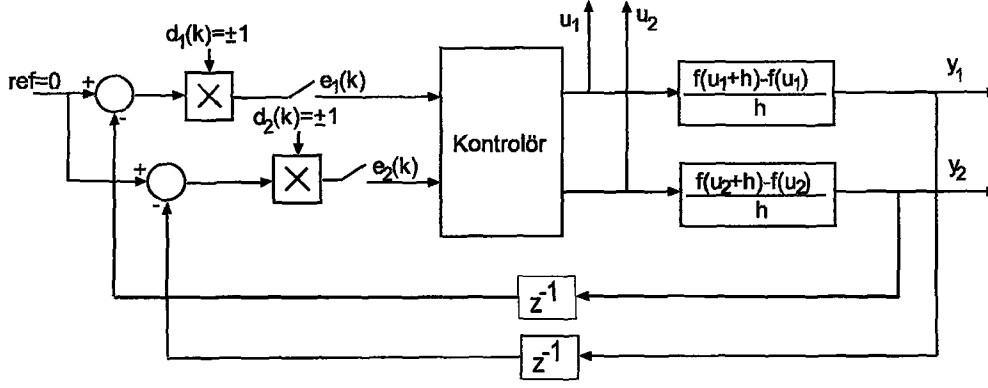
$$\begin{aligned} \frac{\partial F}{\partial x_1} &\cong \frac{f(x_1 + h, x_2, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{h} \\ \frac{\partial F}{\partial x_2} &\cong \frac{f(x_1, x_2 + h, \dots, x_n) - f(x_1, x_2, \dots, x_n)}{h} \\ &\dots \dots \dots \\ \frac{\partial F}{\partial x_n} &\cong \frac{f(x_1, x_2, \dots, x_n + h) - f(x_1, x_2, \dots, x_n)}{h} \end{aligned} \quad (3.19)$$

olarak hesaplanabilir [21]. Buradaki yaklaşık sayısal türev alma işleminde ileri fark yöntemi kullanılmıştır. Aynı işlemi geri fark ya da merkeze eş uzaklıklı fark yöntemi ile gerçekleştirmek de mümkündür. n değişken için kontrolör çıkışı $u(u_1, u_2, \dots, u_n)$ olacağından şekil (3.3)'deki $f(u)$ yerine, $\text{grad}(f(u_1, u_2, \dots, u_n)) = (Y_1, Y_2, \dots, Y_n)$ 'e karşılık olarak,

$$\begin{aligned} y_1 &= \frac{f(u_1 + h, u_2, \dots, u_n) - f(u_1, u_2, \dots, u_n)}{h} \\ y_2 &= \frac{f(u_1, u_2 + h, \dots, u_n) - f(u_1, u_2, \dots, u_n)}{h} \\ &\dots\dots\dots \\ y_n &= \frac{f(u_1, u_2, \dots, u_n + h) - f(u_1, u_2, \dots, u_n)}{h} \end{aligned} \quad (3.20)$$

yazılırsa, sistem $r=0$ için bir ekstremuma erişmeye çalışacaktır. Bu durumda her bir değişken için ayrı bir hata işareti üretileceğinden $d(k)$ yön çarpanı yerine $\mathbf{d}(k)=[d_1(k),d_2(k),...,d_n(k)]$ yön çarpan vektörü gelecektir. Yön çarpan vektörü (3.13)'deki tanımı ile daima $+\text{grad}(f(u_1,u_2,...u_n))$ doğrultusundaki ekstremum olan, enbüyüğe ulaşmayı sağlayacaktır. Böylelikle kontrol çevriminde Hessian matrisinin negatif tanımlılığını denetlemeye gerek kalmamaktadır.

İki değişkenli bir $f(x_1, x_2)$ fonksiyonu için belirli bir (x_{10}, x_{20}) başlangıç noktasından itibaren yerel enbüyük nokta olan (x_1^*, x_2^*) hesaplamak için şekil(3.17)'de görülen kontrol düzenini kullanmak uygun olacaktır. Bu kontrol düzeni ayırık olarak programlanırsa,



Şekil 3.17 Kapalı çevrim kontrol kuramı yaklaşımı ile iki değişkenli bir fonksiyonun yerel ekstremum değerlerinin bulunması.

$$e_1(k) = r - y_1(k-1)$$

$$e_2(k) = r - y_2(k-1)$$

$$x_1(k) = \begin{cases} |u_1| < US \Rightarrow 0.5e_1(k)K_I + x_1(k-1) \\ |u_1| > AS \Rightarrow x_1(k-1) \text{ (Doyma denetimi)} \end{cases}$$

$$x_2(k) = \begin{cases} |u_2| < US \Rightarrow 0.5e_2(k)K_I + x_2(k-1) \\ |u_2| > AS \Rightarrow x_2(k-1) \text{ (Doyma denetimi)} \end{cases}$$

$$u_1(k) = x_1(k) + x_1(k-1) + K_p e_1(k)$$

$$u_2(k) = x_2(k) + x_2(k-1) + K_p e_2(k)$$

$$x_1(k-1) = x_1(k)$$

$$x_2(k-1) = x_2(k)$$

$$y_1(k) = [f(u_1(k) + h, u_2(k)) - f(u_1(k), u_2(k))] / h$$

$$y_2(k) = [f(u_1(k), u_2(k) + h) - f(u_1(k), u_2(k))] / h \quad (3.21)$$

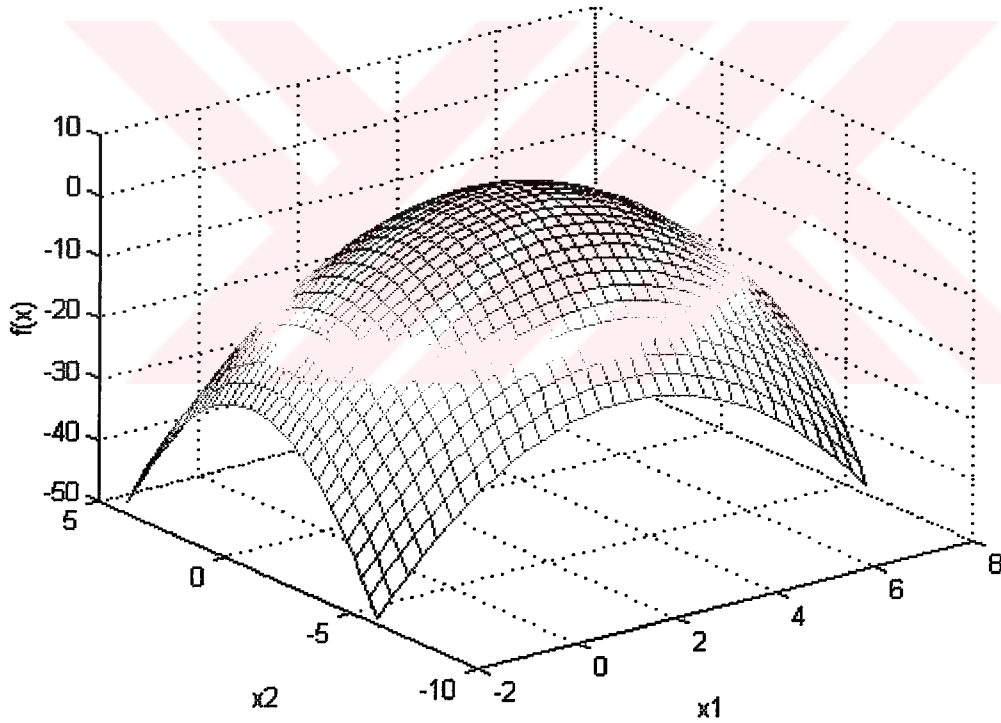
elde edilir. Adım sayısını ifade eden k her özyineleme sonunda bir artırılarak algoritmanın başına döndülür. Yapılan denemelerde iki değişken için sonlandırma kriteri olarak,

$$|e_1(k)| < E_m \wedge |e_2(k)| < E_m \quad (3.22)$$

şartı uygulanmıştır. Buradaki E_m mutlak hatanın üst sınırını ifade etmektedir. Belirli bir (x_{10}, x_{20}) noktasından aramaya başlamak için,

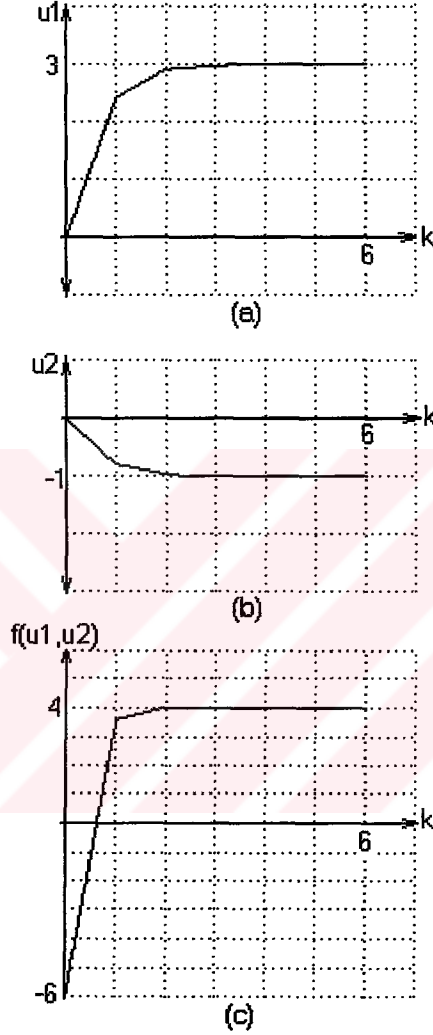
$$\begin{aligned} x_1(k-1) &= x_1(-1) = 0.5 \cdot x_{10} \\ x_2(k-1) &= x_2(-1) = 0.5 \cdot x_{20} \end{aligned} \quad (3.23)$$

olarak seçilmelidir. Şekil 3.18’de görülen $f(x) = -(x_1-3)^2 - (x_2+1)^2 + 4$ fonksiyonunun $(x_1^*, x_2^*) = (3, -1)$ noktasında yer alan bir tane enbüyüğü vardır. $(x_{10}, x_{20}) = (0, 0)$ başlangıç koşulu ile $f'(x_1, x_2) = 0$ çözümü için (3.21) özyinelemeli olarak uygulanmıştır.



Şekil 3.18 $f(x) = -(x_1-3)^2 - (x_2+1)^2 + 4$ fonksiyonunun $[(-2, -8), (-10, 5)]$ aralığında değişimi.

İlk koşul işlemlerinin yapıldığı sıfırıncı adımdan sonraki 6.adımda (Toplam 7 özyineleme) $E_m=0.000001$ sonlandırma kriteri ile $(2.999306,-1.000404)$ 'deki enbüyüğe erişilmiştir. Sonlandırma kriteri 0.000001 olmasına rağmen sonuçta $(-0.0007,+0.0004)$ hata oluşmasının nedeni sayısal türev işleminde $h=0.0001$ alınmasıdır. Seçilen h için oluşan hata yine $\pm E_m$ 'den küçük kalmaktadır.



Şekil 3.19 $f(x)=-(x_1-3)^2-(x_2+1)^2+4$, $K_i=0.4$, $K_p=0.2$, $(x_{10},x_{20})=(0,0)$ için $f'(x_1,x_2)=0$ çözümünde a) Kontrolör çıkışı u_1 bileşeni b) Kontrolör çıkışı u_2 bileşeni c)Fonksiyonun arama sırasında aldığı değerler $f(u_1(k),u_2(k))$.

Tablo 3.5 $f(x)=-(x_1-3)^2-(x_2+1)^2+4$, $K_i=0.4$, $K_p=0.2$ $(x_{10},x_{20})=(0,0)$ için $f'(x_1,x_2)=0$ çözümünde kontrolör çıkışının ve fonksiyonun aldığı değerler.

Değerlendirme k	Kontrolör çıkışı		Fonksiyon değeri $f(u_1,u_2)$
	$u_1(k)$	$u_2(k)$	
0	0	0	0
1	2.39445	-0.7991791	3.599004
2	2.880096	-0.9593964	3.983974
3	2.976418	-0.9918213	3.999377
4	2.995491	-0.998497	3.999977
5	2.999306	-0.9994507	3.999999
6	2.999306	-1.000404	3.999999

Tablo 3.5’de görüldüğü üzere 6.özyinelemede $f(x_1^*,x_2^*)=3.999999$ olarak hesaplanmıştır. Toplam 7 adımda 0.000001 hata ile enbüyük değerini hesaplanması yordamın etkinliği bakımından oldukça iyi bir sonuçtur. Ancak etkinlik katsayısının yüksek olması fonksiyona uygun kontrolör parametrelerinin seçimine de bağlıdır. Önbilgi bulunmayan bir fonksiyon için PI tipi kontrolör parametrelerinin yaklaşık olarak seçimi 4.bölümde açıklanan kararlılık ve kaçma (kök atlama) sorunlarına neden olmaktadır. Bu sorunlara karşı uygun bir kontrolör tasarımı da 4.bölümde incelenecektir.

4. PI TİPİ KONTROLÖR İLE SAYISAL ARAMADA KARŞILAŞILAN SORUNLAR VE BULANIK KONTROLÖR KULLANIMI

3.Bölümde kapalı çevrim kontrol düzeni yaklaşımı ile denklem çözümü ve yerel eniyileme açıklanmıştır. Söz konusu çözüm tekniği için verilen örneklerde lineer kontrolör (PI) kullanılmıştır. Bu bölümde lineer kontrolör kullanımında çıkan sorunlara değinilmekte ve çözüm olarak bulanık kontrolör yapısı önerilmektedir.

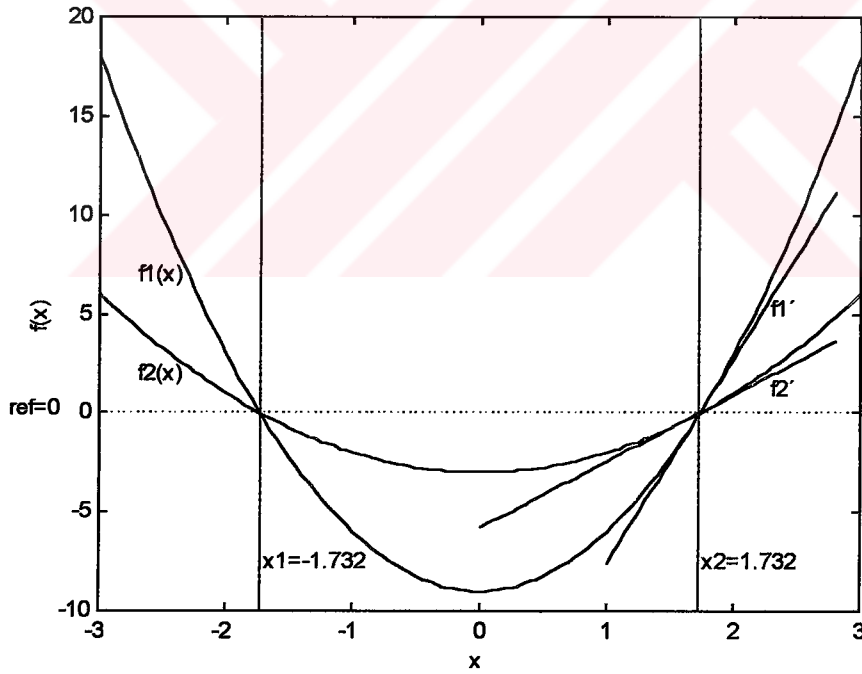
4.1 Sayısal Aramada PI Tipi Kontrolör Kullanımında Karşılaşılan Sorunlar

Kapalı çevrim kontrol kuramı yaklaşımı ile sayısal kök aramada PID vb lineer kontrolörlerin kullanımında bazı sorunlarla karşılaşılmaktadır. Sayısal arama yapılan fonksiyon, arama aralığında lineer olmayan bir kazanç elemanı gibi davranmaktadır. Fiziksel sistemlerin kapalı çevrimli kontrolüne benzer şekilde, sayısal arama probleminin çözümünde de parametrelerin yanlış seçilmesi, salınım, yerleşme süresinin artmasına ve kontrol işaretinin kararlı noktayı geçerek doymaya girmesine neden olmaktadır.

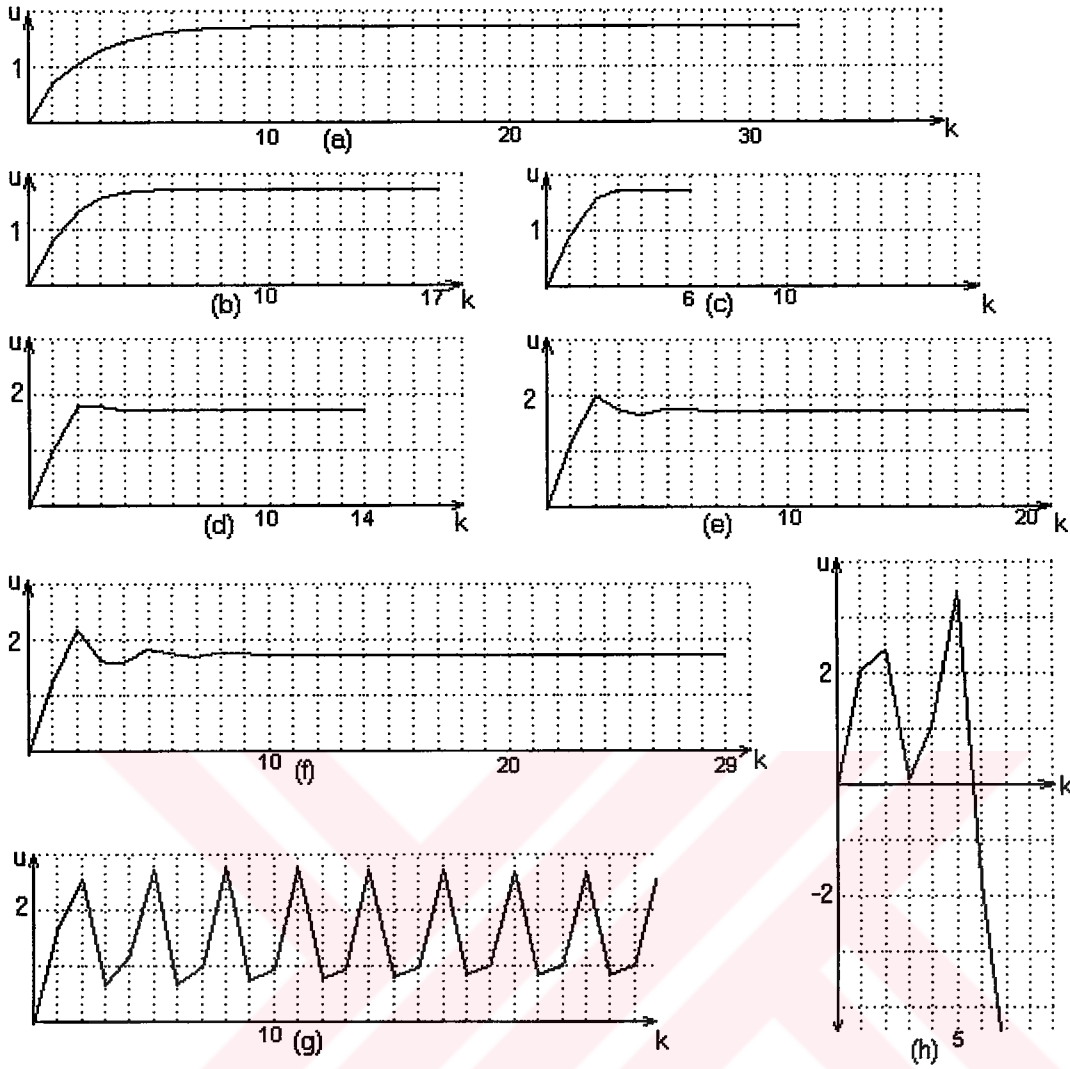
Örnek olarak $r=0$, $K_p=0.05$ alınarak çeşitli K_i değerleri için $f_1(x)=3x^2-9$ fonksiyonun kökleri aranmıştır. Tablo 4.1’de görüldüğü üzere $K_i=0.05$ için $x=1.732051$ ’deki köke $E_m=0.00001$ sonlandırma kriteri ile 32 özyinelemede erişilmektedir. $K_i=0.1$ ’e kadar algoritmanın etkinliği artmaktadır. K_i ’nin arttırılmaya devam edilmesi ile aşım başlamakta ve sonuca erişmedeki değerledirme sayısı artmaktadır. Şekil 4.2-g’de görüldüğü gibi $K_i=0.28$ için salınım oluşmakta ve sonlandırma şartı sağlanamamaktadır. Burada dikkat edilmesi gereken bir nokta salınımın ağırlık merkezinin (ortalamasının) kök civarında olmasıdır. Ayrık bir filtrenin geribesleme yoluna konulması önerilebilir. Fakat bu tür bir önlem sonuca erişme hızını düşürür. $K_i=0.35$ için (Şekil 4.1-h) kontrolör çıkışı genliği artan bir salınım yapmakta ve doyma sınırına ulaşmaktadır.

Tablo 4.1- $K_p=0.05$, $3x^2-9=0$ denkleminin PI tipi kontrolörle çözümünde çeşitli K_i integral kazanç değerleri için sonuca erişimdeki döngü sayıları

Şekil 4.1	K_i	Döngü sayısı (Özyineleme)
(a)	0.05	32
(b)	0.075	17
(c)	0.1	6
(d)	0.125	14
(e)	0.15	20
(f)	0.175	29
(g)	0.28	sürekli salınım
(h)	0.35	kararlı noktadan uzaklaşarak doyma (kararsız)



Şekil 4.1 $f_1(x)=3x^2-9$ ve $f_2(x)=x^2-3$ fonksiyonlarının kök civarında doğrusallaştırılması : $f_1'(1.732)=10.39$, $f_2'(1.732)=3.46$



Şekil 4.2 Tablo 4.1’de görülen parametrelere bağlı olarak, $f(x)=3x^2-9, r=0$ için $f(x)=r$ denkleminin çözümünde kontrolör çıkışının değişimi : a) $K_i=0.05$ b) $K_i=0.075$ c) $K_i=0.1$ d) $K_i=0.125$ e) $K_i=0.15$ f) $K_i=0.175$ g) $K_i=0.28$ h) $K_i=0.35$

4.2 Sayısal Aramada Lineer Kontrolör Kullanımında Çıkan Sorunların Nedeni ve Öneriler

$f_1(x)=3x^2-9$ ve $f_2(x)=x^2-3$ fonksiyonlarının her ikisinde $f(x)=0$ çözümleri aynı olmasına rağmen aynı K_p , K_i değerleri için algoritmanın etkinliği değişmektedir. Bunun sebebi ise kök civarındaki kazanç değerlerinin (değişim oranının) farklı olmasıdır. Yukarıdaki örnek için değişim oranı $f_2'(\sqrt{3})=3.464102$,

$f_1'(\sqrt{3})=10.392304$ 'dir. Buna göre, kök civarında kontrolörün çıkışındaki $u(k)-u(k-1)=\Delta u$ değişimi $f_1(x)$ çıkışında $f_2(x)$ çıkışındakinin 3 katı etkili olacaktır. Aynı algoritma etkinliğinin sağlanması bakımından $K_{i2}=3K_{i1}$ olarak seçilmelidir. Ancak burada ikinci önemli bir nokta da kritik sönüm şartının sağlanmasıdır. Arama yapılan aralığın sınırları $[AS,US]$ ile gösterilmiş olsun. Lipchitz şartı sağlanıyorsa [4], $[AS,US]$ aralığındaki herhangi x_1, x_2 için,

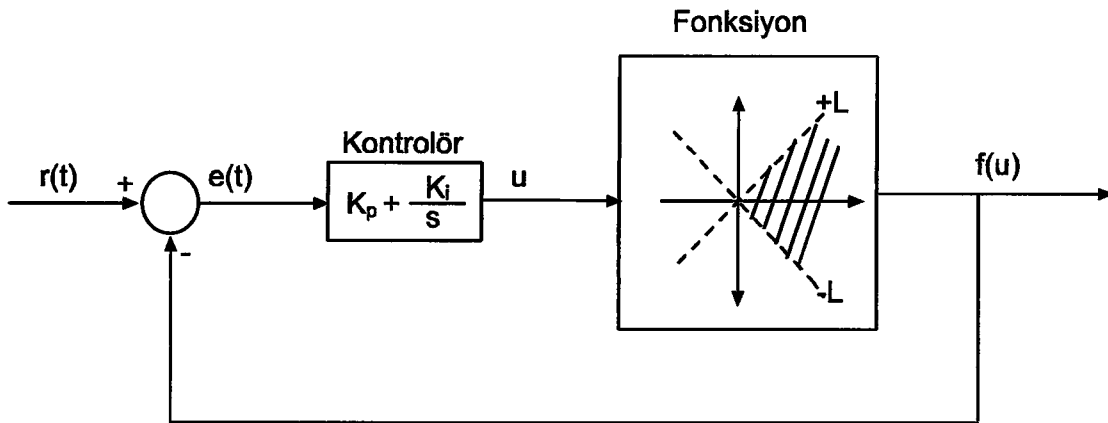
$$L \geq |f(x_1)-f(x_2)| / |x_1-x_2| \text{ ve } |x_1-x_2| \geq E_m \quad (4.1)$$

sağlayan L bulunabilir. Bu değer bir ön bilgi olarak verilmiş olması $[AS,US]$ aralığında, $|\Delta f(u)| = |f(x_1)-f(x_2)|$ olarak tanımlanırsa,

$$\Delta f(u) \leq \pm L \quad (4.2)$$

olmasını garanti etmektedir.

Burada açıklanan sayısal arama düzeninin gerçek zamanlı bir eşdeğeri, doğrusal bir kontrolör ve lineersizliği $\pm L$ ile sınırlı bir sektör fonksiyonu ile modellenirse şekil 4.3'deki kapalı çevrimli kontrol düzeni elde edilir. Burada sektör fonksiyonu $|\Delta f(u)| \leq L$ ilişkisini sağlayan fonksiyonlar kümesini oluşturur. Blok diyagramda PI kontrolörün transfer fonksiyonu $G(s)=K_p+K_i/s$ alınmıştır.



Şekil 4.3 Lipchitz değeri bilinen bir fonksiyonun köklerinin integral kontrolörle aranması.

Lineer olmayan sektör fonksiyonlarının kararlılık sınırını veren daire kriterini matematiksel ifadesine göre[22],[23],

$$\Delta K \cdot \sup_{\omega \in \mathbb{R}} |G_{km}(j\omega)| < 1 \quad (4.3)$$

eşitsizliğinin sağlandığı bölgede şekil4.3'deki kontrol düzeni kararlıdır. Daire kriterinde yer alan lineersizliğin alt ve üst sınırları K_1 ve K_2 burada $+L$ ve $-L$ 'ye karşı düşmektedir. Buna göre 4.3'de yer alan ΔK ,

$$\Delta K = \frac{+L - (-L)}{2} = L \quad (4.4)$$

ve lineersizliğin ortalaması K_m ,

$$K_m = \frac{+L + (-L)}{2} = 0 \quad (4.5)$$

olarak elde edilir. (4.3)'e göre $G_{km}(j\omega)$ eğrisinin $\frac{1}{\Delta K}$ yarıçaplı dairenin içinde kaldığı bölgede lineer olmayan kontrol sistemi kararlıdır. Buradaki $G_{km}(s)$, $G(s)$ cinsinden,

$$G_{km}(s) = \frac{G(s)}{1 + K_m G(s)} = G(s) \quad (4.6)$$

şeklinde hesaplanır. Şekil 4.3'deki kontrol düzeninde $G(s) = \frac{K_p(s + K_i / K_p)}{s}$

olduğundan kararlılık koşulu, $s \rightarrow 0$ için $|G_{km}(j\omega)| = \left| \frac{K_i}{j\omega} \right|$ ilişkisinden,

$$\frac{K_i^2}{\omega^2} < \frac{1}{L^2} \quad (4.7)$$

ya da,

$$L \cdot K_i < \omega \quad (\omega > 0) \quad (4.8)$$

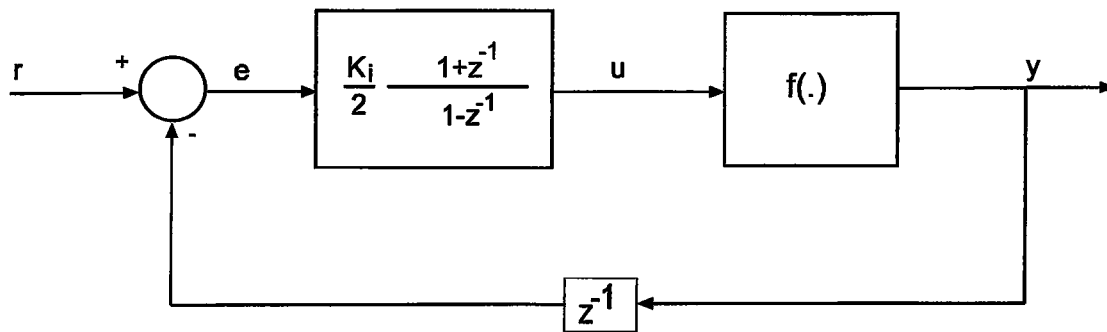
olarak hesaplanır. Diğer bir bakış açısıyla değişimi $\pm L$ aralığında sınırlı, rasgele değerler alabilen bir fonksiyon üzerinde, PI kontrolör ile yapılan sayısal aramada, kararlı bölgede kalınabilmesi için, verilen L sınırı ne kadar büyükse K_i kazancı aynı oranda küçük olmalıdır. Ancak ω 'nın alt sınırı hakkında kesin bir bilgi yoksa önceden kararlılığı garanti etmek mümkün olmaz. Buradaki ω , açısal hız olup özel olarak $K_i \cdot L$ 'ye eşit olduğu durumda kapalı çevrimli sistemin salınım frekansına eşittir. Benzer bir durum sayısal arama amaçlı ayrık modellenen sistem için de geçerlidir (şekil 4.2).

Eğer $f(\cdot)$ fonksiyonu için bir L değeri verilebiliyorsa, kök civarında bir doğrusallaştırma yapılabildiği kabulü ile değişimin üst sınırı L olacağı sonucu ortaya çıkar. Fonksiyonun arama noktası civarındaki kökü x^* ise,

$$|f(x^*)| < L \quad (4.9)$$

yazılabilir.

Şekil 3.2'deki kontrol düzeni, $K_p \approx 0$ için şekil 4.3'de görülen kapalı çevrimli kontrol düzenine eşdeğerdir.



Şekil 4.4 Ayrık poligonal integral kontrolör ile kök aranması

Şekil 4.4'deki $f(\cdot)$ fonksiyonu için bir L değeri verilmiş olsun . Arama algoritmasının kök civarında salınım girmemesi için gerekli koşul, kapalı çevrim kutubunun birim daire içerisinde kalması olacaktır. Sayısal kontrol sistemlerinde bu koşulun belirlenmesi için karakteristik denkleme Jury [19] kararlılık testi uygulanır.

$$q(z) = 1 + \frac{K_i L}{2} \frac{z+1}{(z-1)z} = 0 \quad (4.10)$$

düzenlenirse,

$$q(z) = 2z^2 - (K_i L - 2)z + K_i L = 0 \quad (4.11)$$

karakteristik denklemi elde edilir. Jury kararlılık testine göre (4.11)'deki polinom,

$$q(-1) > 0 \quad \Rightarrow \quad K_i L > 0 \quad (4.12)$$

($q(1) > 0$ koşulundan $K_i L$ 'ye ilişkin bir koşul çıkmaz)

$$|a_0| < a_n \quad \Rightarrow \quad K_i L < 2 \quad (4.13)$$

koşullarını sağladığı sürece arama algoritması kararlı olacaktır. Nitekim, tablo 4.1'de görülen örnekte, $f'(x^*) = f'_1(\sqrt{3}) = 10.39 \cong L$ olarak kabul edilirse algoritmayı kararlı kılan K_i değer aralığı,

$$0 < K_i < \frac{2}{L} \cong 0.19 \quad (4.14)$$

olarak hesaplanır. Tablo 4.1'de görüldüğü üzere 0.19'dan küçük K_i değerleri için sistem kararlı kalmaktadır. Burada $K_i \cdot L$ değerine getirilen üst sınır, analog sistemlerdekine benzer şekilde sistemin salınım frekansı ile orantılıdır (şekil 4.2 – 9).

K_i 'nin çok küçük seçilmesi sönüm katsayısını ve dolayısıyla döngü sayısını arttırmakta, $2/L$ 'ye yakın seçilmesi ise aşım sebep olmaktadır. Bu sebepler göz önüne alındığında K_i integral katsayısının fonksiyonun getirdiği çevrim kazancına bağlı olarak, değişken alınması gerekliliği ortaya çıkar. Böylelikle hem salınım girme engellenecek, hem de az bir döngü sayısı ile sonuca erişilecektir.

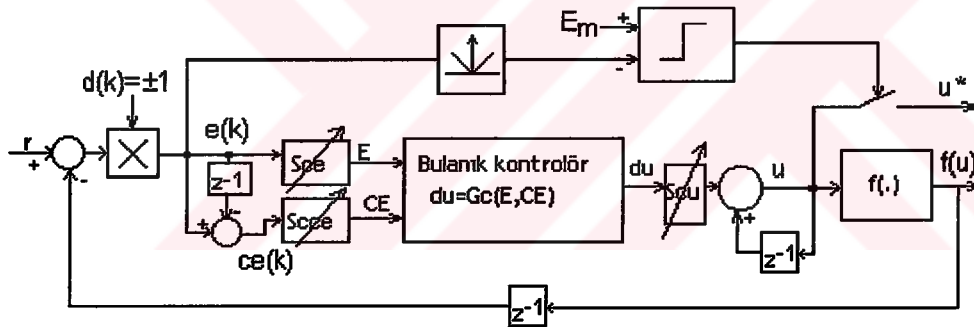
Arama öncesi belirli bir L 'değerinin verilemediği durumda $r=0$ için hata işaretinin değişimi,

$$\Delta e(k) = e(k) - e(k-1) \cong \Delta f(uc) \quad (4.15)$$

olacağından, $\Delta e(k)$ arama yapılan nokta civarında fonksiyon kazancının bir ölçütü olarak kullanılabilir.

4.3 Sayısal Aramada Bulanık Kontrolör Kullanımı

(4.8) ve (4.14)'de de görüldüğü üzere PI tipi kontrolörün kararlılığı ve çözüme ulaşma hızı, arama yapılan fonksiyonun yapısına göre değişkenlik göstermektedir. Geniş bir fonksiyon ailesinde özel bir hazırlığa gereksinim olmadan sonuca erişim için kontrolör parametrelerinin değişken olması gereksinimi ortaya çıkar. Burada, bu gereksinimi karşılamak amacıyla, hata işaretinin genliği ve değişim miktarına bağlı olarak adım uzunluğunu belirleyecek bir bulanık kontrolör kullanımı önerilmektedir. 3.Bölümde önerilen çözüm yönteminde kontrolör bloğu yerine bulanık kontrolör yerleştirilirse şekil 4.5'de görülen kontrol sistemi elde edilir.



Şekil 4.5 Bulanık kontrolörün sayısal aramaya uyarlanması

Burada açıklanan yöntemde Mamdani'nin [24] önerdiği bulanık kontrolör yapısı kullanılmıştır. Şekil 4.5'de görülen blok yapıda e hatayı, ce hata değişimini, Scke hata ölçeklendirilme katsayısını, Scce hata değişiminin ölçeklendirilme katsayısını, E ölçeklendirilmiş hatayı, CE ölçeklendirilmiş hata değişimini, du bulanık kontrolör çıkışını, u ayrık integre edilmiş kontrolör çıkışını ve u^* ulaşılan kök değerinin göstermektedir.

Blok diyagramda görüldüğü gibi k'ncı adımda değerlendirilen hata işareti,

$$e(k)=[r-f(u(k-1))]d(k) \quad (4.16)$$

3.Bölümde de açıklandığı üzere başlangıç noktasına göre istenen yönde kök arayabilmek için bir defaya mahsus olmak üzere 1.adım sonunda d(k) değeri,

$$d(k) = \begin{cases} d(0) = +1, + e(0) \text{ yönünde kök arama} \Rightarrow d(k > 0) = +1 \\ d(0) = +1, - e(0) \text{ yönünde kök arama} \Rightarrow d(k > 0) = -1 \end{cases} \quad (4.17)$$

şeklinde alınmıştır. e(k) Hata işareti, (4.17) ile tanımlanan d(k) değeri ile çarpılarak istenen yönde kök aramaya devam edilir.

Bulanık kontrolör karar mekanizması için gerekli olan hatanın değişimi,

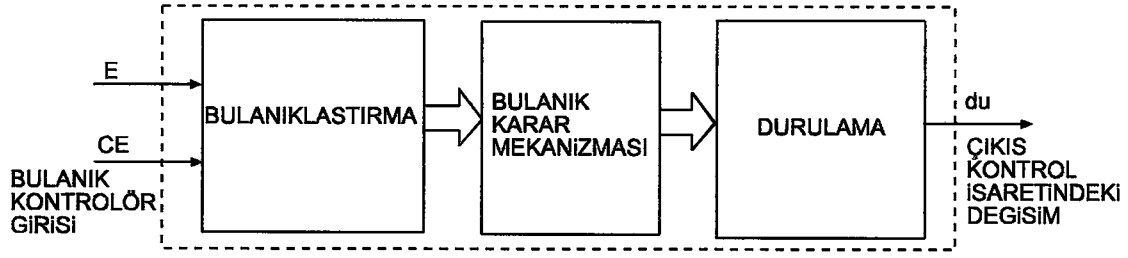
$$ce(k) = (1 - z^{-1})e(k) = e(k) - e(k-1) \quad (4.18)$$

fark ifadesi ile elde edilir. Ölçeklendirilmiş hata E ve hatanın değişimi CE

$$E(k) = S_{ce} \cdot e(k) \quad (4.19)$$

$$CE(k) = S_{cce} \cdot ce(k) \quad (4.20)$$

şeklinde hesaplanır. Ölçeklendirmenin amacı, giriş işaretinin değişim aralığını bulanık kontrolörün kural tabanının tanımladığı aralığa yaklaştırmaktır. Örneğin e, 0..±10 aralığında , ce ise 0..±20 aralığında değişiyor olsun. Eğer bulanık kontrolör içerisindeki üyelik tanımları 0..±1 aralığındaki girişlere göre yapılmışsa, $S_{ce}=0.1$, $S_{cce}=0.05$ seçmek uygun olacaktır. Üyelik tanımlarında doyma olduğundan e ve ce için verilen aralığın kesin olması gerekmez. Çoğunlukla çalışılan değer aralığını seçmek yeterlidir. Şekil 4.5’de de görüldüğü gibi alt sınır ve üst sınırdaki doyma olması çalışma sınırını esnek kılmaktadır. E ve CE’nin değişim aralığının, üyeliklerin tanım aralığına yakın olacak şekilde ölçeklendirilmiş olması performansın iyileştirilmesi bakımından önemlidir. e ve ce’nin çok geniş bir aralıkta değişebilmesi durumunda, birden fazla ölçeklendirme katsayısı kullanılarak mevcut kural tabanının etkinliği artırılabilir. “Pencereleme” (windowing) olarak adlandırılan bu teknik ile üyelik fonksiyon sayısını arttırmaksızın geniş bir işaret değişim aralığında çalışmak mümkün olmaktadır. Buradaki tanımlar öncelikle sabit ölçeklendirme katsayıları için yapılacaktır.



Şekil 4.6 Bulanık kontrolör blok diyagramı

Ölçeklendirilmiş hata (E) ve hata değişim işareti (CE) bulanıklaştırılarak, bulanık çıkartım mekanizmasına uygun girişli hale getirilir. Bulanıklaştırma işlemi, E ve CE'nin büyüklüğüne göre üyelik derecelerinin belirlenmesi işlemidir. Yapılan uygulamada temel üyelikler,

NB: Negatif Büyük,

NO: Negatif Orta,

NK: Negatif Küçük,

S : Sıfır,

PK: Pozitif Küçük,

PO: Pozitif Orta,

PB: Pozitif Büyük

olarak seçilmiştir. Geleneksel küme kuramında bir a elemanının A kümesinin elemanı olması μ üyelik derecesi bakımından,

$$a \in A \Rightarrow \mu_A(a) = 1$$

$$a \notin A \Rightarrow \mu_A(a) = 0$$

olarak değerlendirilir. μ üyelik derecesi 0 ya da 1 dışında bir değer alamaz. Oysa bulanık mantık kuramında bir büyüklüğün belirli bir kümeye aidiyeti 0 ya da 1 dışında değerler alabilir. Bulanık bir değişkenin hangi oranlarda hangi kümeye ait

olduğunu üyelik fonksiyonları belirler. A ve B , X örnek uzayında tanımlı iki bulanık küme olsun. Herhangi bir $x \in X$ elemanı için,

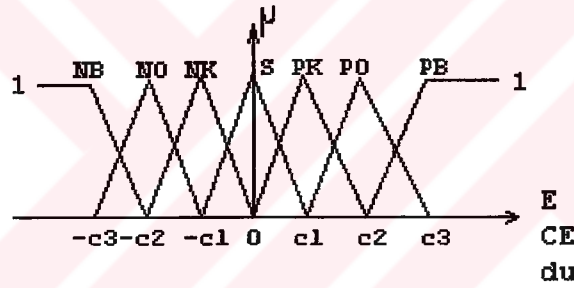
$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (\text{Birleşme}) \quad (4.21)$$

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad (\text{Kesişme}) \quad (4.22)$$

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \quad (\text{Tümleme}) \quad (4.23)$$

bulanık ilişkileri geçerli olur [25].

Yapılan uygulamada giriş ve çıkış değişkenlerinin üyeliklerinin belirlenmesinde Şekil 4.7’de görülen üyelik fonksiyonları kullanılmıştır. Buna göre, örneğin $\mu_{PK}(E=c1)=1$, $\mu_{NK}(E=-c1/2)=0.5$, $\mu_S(E=-c1/2)=0.5$ ya da $[0, c1]$ aralığında $\mu_S(E) + \mu_{PK}(E)=1$ olacaktır.



Şekil 4.7 Hata , hatanın değişimi ve çıkış işareti için üyelik fonksiyonları

Şekil 4.6’daki blok diyagramda görülen bulanık karar mekanizmasının işlevi bulanık giriş büyüklükleri ve kural tabanına bağlı olarak bulanık sonuç çıkartımıdır. Kural tabanı, sistemin kontrolü hakkındaki bilgi ve tecrübenin bir düzen içerisindeki ifadesidir. Sayısal arama probleminde kullanılan çıkartımların BASIC dili karşılığı ek A’da verildiği gibidir.

Bulanıklaştırma sonucu, E ve CE’nin üyelik derecelerine göre kontrolörün vermesi istenen bulanık cevaplar tablo 4.2’de toplu olarak görülmektedir.

Tablo 4.2 Hata (E) ve Hatanın değişimine (CE) göre bulanık karar tablosu

E CE	NB	NO	NK	S	PK	PO	PB
NB	NB	NB	NB	NB	NO	NK	S
NO	NB	NB	NB	NO	NK	S	PK
NK	NB	NB	NO	NK	S	PK	PO
S	NB	NO	NK	S	PK	PO	PB
PK	NO	NK	S	PK	PO	PB	PB
PO	NK	S	PK	PO	PB	PB	PB
PB	S	PK	PO	PB	PB	PB	PB

Şekil 4.7'deki üyelik fonksiyonlarına göre E ve CE ($\pm c1, \pm c2, \pm c3$) değerleri dışında aynı anda sıfırdan farklı iki üyeliğe sahiptirler [26]. Bu sebeple durulama bloğuna tablo 1'e göre 2'nin (E ve CE) 2'li kombinasyonu (iki ayrı üyelik derecesi) olan 4 tane sonuç değişik üyelik dereceleri ile iletilir. B1 ve B2 E'nin bulanık ifadesi, B3 ve B4 CE'nin bulanık ifadesi olsun. B1, B2, B3 ve B4 , NB, NO, NK, S, PK, PO, PB değerlerinden birini ilgili üyelik derecesinde alacaktır. (4.23)'ye göre E için, simetri nedeniyle,

$$\mu_{B1}(E) = 1 - \mu_{B2}(E) \quad (4.24)$$

olacaktır. Benzer şekilde CE için,

$$\mu_{B3}(CE) = 1 - \mu_{B4}(CE) \quad (4.25)$$

geçerlidir. E(B1, B2) ve CE(B3, B4)'ün sonucu elde edilen 4 kuralın üyelik dereceleri (4.21) ve (4.22)'ye bağlı olarak,

$$\mu_{K1} = \min(\mu_{B1}(E), \mu_{B3}(CE)) \quad (4.26)$$

$$\mu_{K2} = \min(\mu_{B1}(E), \mu_{B4}(CE)) \quad (4.27)$$

$$\mu_{K3} = \min(\mu_{B2}(E), \mu_{B3}(CE)) \quad (4.28)$$

$$\mu_{K4} = \min(\mu_{B2}(E), \mu_{B4}(CE)) \quad (4.29)$$

ilişkilerinden hesaplanır.

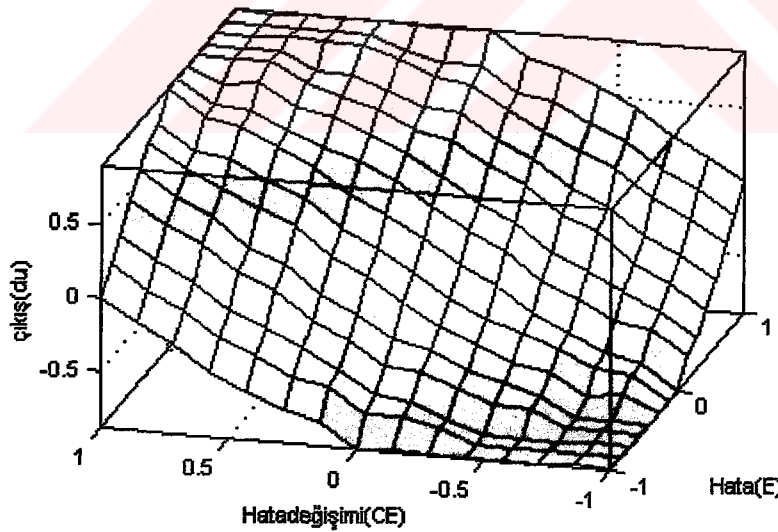
Elde edilen bulanık kararlar ve üyelik dereceleri, ağırlık merkezi yöntemi ile durulanarak du çıkış değeri elde edilmiştir. Ağırlık merkezi yöntemine göre durulanmış çıkış işareti,

$$du = \frac{\sum_{i=1}^4 u_i \mu(u_i)}{\sum_{i=1}^4 \mu(u_i)} \quad (4.30)$$

ile ayrık olarak ifade edilir. Tablo 1'e göre K1,K2,K3 ve K4 kurallarına karşı düşen çıkışların kesin (crisp) değerleri u1,u2,u3 ve u4 olsun. Buna göre (4.30) gereği durulanmış çıkış,

$$du = (\mu_{K1} \cdot u1 + \mu_{K2} \cdot u2 + \mu_{K3} \cdot u3 + \mu_{K4} \cdot u4) / (\mu_{K1} + \mu_{K2} + \mu_{K3} + \mu_{K4}) \quad (4.31)$$

şeklinde hesaplanacaktır. Hata (E) ve hata değişimine(CE) bağlı olarak du çıkışı bir yüzey oluşturur. Şekil 4.8'de görülen bu grafiğe bulanık kontrol yüzeyi adı verilir.



Şekil 4.8 Bulanık kontrolör çıkışına ilişkin kontrol yüzeyi

Neticede du durulanmış bulanık kontrolör çıkışı, kontrol edilen sisteme (sayısal aramada f(x) fonksiyonu) uygulanan işaretin değişim miktarını belirler. Buna göre k. Adımda f(x) fonksiyonuna uygulanacak x=uc işareti, ayrık integrasyon sonucu,

$$u(k)=u(k-1)+Scu \cdot du(k) \quad (4.32)$$

olacaktır. Bu tür bir bulanık kontrolörün çıkışı du , hata e ve hatanın değişimi ce 'ye göre belirli bir karar değerinden sonra doymaya girmektedir. Arama yapılan bölgede kök yoksa ya da ilk koşul olarak bulanık kontrolör çıkış kazancı, fonksiyonun Lipchitz değerine göre çok büyük ise, u kontrol işareti arama sınırının dışına çıkacaktır. Bu durumda arama yapılan bölgede bir kök bulunsa bile, bölgede bir kökün bulunmadığı sonucuna varılır.

Arama yapılacak üst sınır US ; başlama noktası AS olsun. Uygulama örneklerinde,

$$c_3 = \lfloor US/2 \rfloor, \quad c_2 = (2/3) \cdot c_3, \quad c_1 = (1/3) \cdot c_3 \quad (4.33)$$

alınmıştır. (AS, US) aralığında arama yapabilmek için, $u(0)=AS$ olmalıdır. Yapılan bu atama nümerik arama işlemine istenen bir noktadan başlanmasını sağlamaktadır.

Fonksiyonun bulunulan bir $u(k-1)$ noktasındaki değeri $f(u(k-1)) > 0$ ise bir sonraki adımda kök değerinin aşılmaması için fonksiyonun $-L$ Lipchitz değerinden daha hızlı sıfıra yaklaşamayacağı koşulundan faydalanılabilir. Buna göre (4.1) ifadesinde, $[u(k)-u(k-1)]$ hep pozitif olduğundan $u(k)$ 'nın kök noktasını aşmaması için,

$$u(k) < \lfloor f(u(k-1)) \rfloor / L + u(k-1) \quad (4.34)$$

ilişkisi sağlanmalıdır. Bu ifadeye göre bulunulan $u(k-1)$ noktasında fonksiyon değeri negatifse,

$$f(u(k-1)) < 0 \Rightarrow \max(f(u(k-1))) = 0$$

pozitifse,

$$f(u(k-1)) > 0 \Rightarrow \min(f(u(k-1))) = 0 \quad (4.35)$$

koşulu altında bir sonraki $u(k)$ adımında kök değeri aşılmamış olur. (4.29) ve (4.32)'ye göre çıkış işaretinin aşırı olarak köke ulaşmasını garanti etmek için,

$$Scu \cdot du(k) < \lfloor f(u(k-1)) \rfloor / L \quad (4.36)$$

sınırlaması elde edilir. Kök arama probleminde $r=0$ alındığından $e(k)=-f(u(k))$ ve $du=Gc(E,CE)$ olduğundan, çözümü garanti eden du 'nun üretimi, sabit c_1, c_2, c_3 için, sadece Scu seçimine bağlıdır. L değeri bilinmeyen bir fonksiyon için adım boyu ve genlik değişimine bağlı ardışıl bir üst sınır değeri (4.1) ilişkisinden,

$$L(k)=\max(L(k-1), |f(u(k))-f(u(k-1))|/|u(k)-u(k-1)|) \quad (4.37)$$

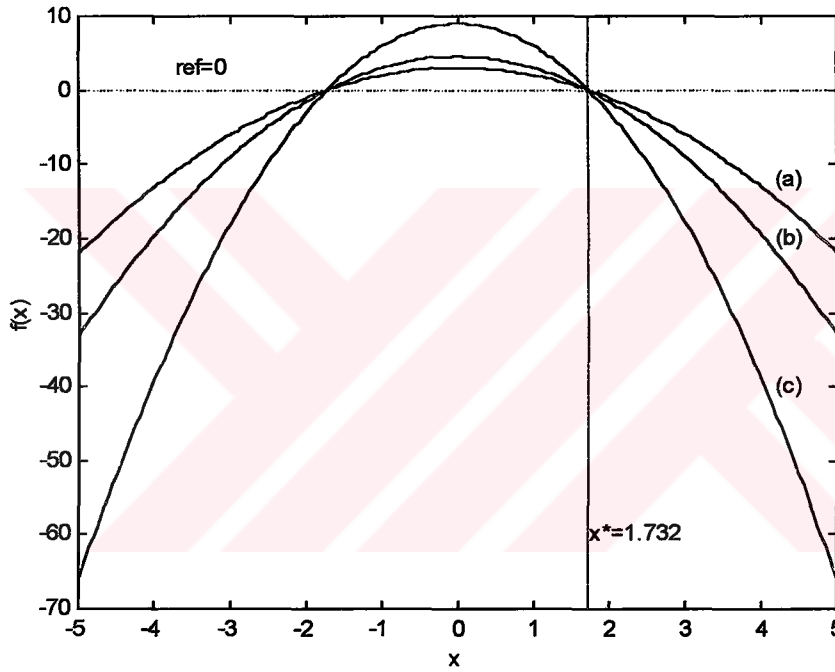
ifadesi ile türetilir. Burada elde edilen $L(k)$ değeri adım duyarlılığına karşılık düşmektedir. $L(k)$, L değerinin bilinmediği durumda Scu 'nun ayarlanmasında kullanılabilir fakat kuramsal olarak sonuca erişmeyi garanti etmez.

4.3.1 Bulanık Kontrolör İle Kök Aranması

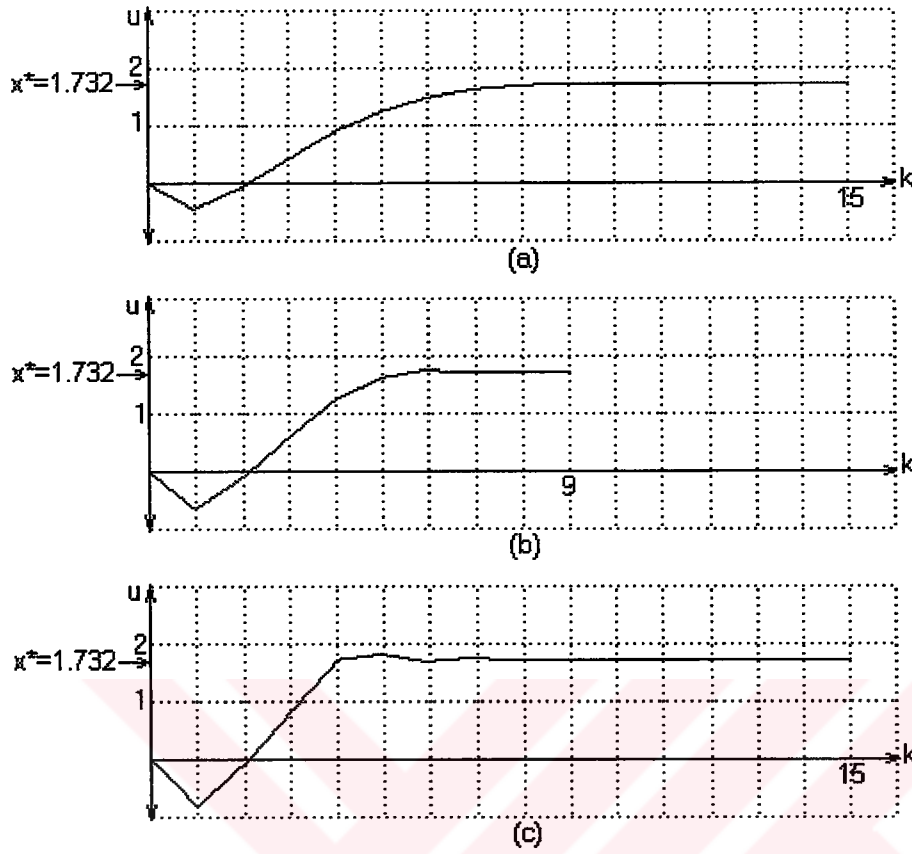
Şekil 4.5'de blok diyagramı görülen kontrol düzeni yazılım ile gerçekleştirir (Ek B) ve $r=0$ alınırsa, $f(.)$ fonksiyonunun yerine konulacak fonksiyon için $f(x)=0$ çözümü elde edilir. Üçüncü bölümde anlatıldığı gibi önerilen düzen, $[AS,US]$ aralığındaki sayısal aramaya alt sınırdan başlar ve kök bulunana ya da $u \geq UB$ (doyma) olana kadar devam edilir. Şekil 4.9'da görülen üç fonksiyonun ($f(x)=-x^2+3$, $f(x)=-1.5x^2+4.5$, $f(x)=-3x^2+9$) kökleri birbirine eşit fakat kök civarındaki Lipchitz değerleri birbirinden farklı olmasına rağmen sabit $Sc_e=0.9$, $Sc_{ce}=0.05$, $Sc_u=0.3$ ölçeklendirme katsayıları için başarı ile çözüme erişildiği görülmüştür.

Şekil 4.10'da (a) $-x^2+3=0$, (b) $-1.5x^2+4.5=0$, (c) $-3x^2+9=0$ denklemlerinin çözümü sırasında kontrol işaretinin değişimi görülmektedir. Burada başlangıç noktası $AS=u(0)=0$ alınmıştır. $E_m=0.00001$ sonlandırma kriteri ile sırasıyla (a)16, (b)10 ve (c)16 çevrimde $x^*=1.73206$ 'daki köke erişilmiştir. Açıklanan sayısal kök arama yönteminde doğru sonuca hızlı erişmedeki en önemli etken, fonksiyonun kök civarındaki türevidir. Aynı ölçeklendirme katsayıları kullanılırsa, Lipchitz değeri küçük fonksiyonlarda adımlamalar küçük olacağından sonuca erişmedeki özyineleme sayısı artacaktır. Çıkış ölçeklendirme katsayısı Sc_u , arama sırasındaki adımlamalara bir üst sınır getirmektedir. Kök aramada ve fonksiyon türevinin kökünü aramadaki adım üst sınırı birbirinden farklı seçilmelidir. 5.Bölümde açıklanan global eniyileme yönteminde, türev çevrimlerinde eniyi değer bulunmakta, kök çevrimlerinde ise sadece yeni başlangıç noktası yaratılmaktadır. Bu durumda türev çevrimleri sırasındaki du 'nun üst sınırı arama duyarlılığı ile orantılıdır. Sonuca erişmedeki hata

üst sınırı daha küçük olan türev çevrimlerindeki Scu , kök çevrimlerindeki göre daha küçük seçilmiştir. Seçilen bir du üst sınırına göre salınım girmeme şartını sağlayan Scu değeri (4.36) eşitsizliğinden hesaplanabilir. Salınım sınırına yaklaşıldıkça çözüm hızının artacağı açıktır. Ancak eniyi değerin ne olacağı ayrı bir araştırma konusudur. Yapılan uygulamalarda Scu , sistemi salınım sokağından başlatılarak küçültülmüş ve salınım sokağına sokan değerin %75..90'ı aralığında en hızlı çözümü veren Scu değeri elde edilmiştir.

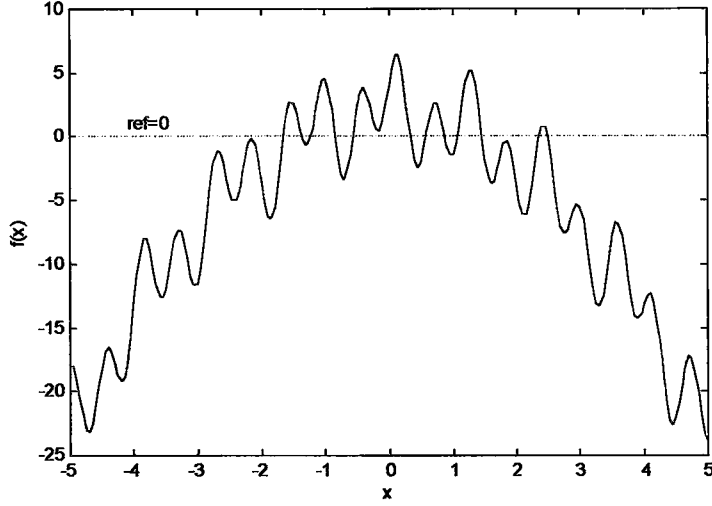


Şekil 4.9 a) $f(x)=-x^2+3$ b) $f(x)=-1.5x^2+4.5$ c) $f(x)=-3x^2+9$ fonksiyonlarının $(-5,+5)$ aralığında değişimi



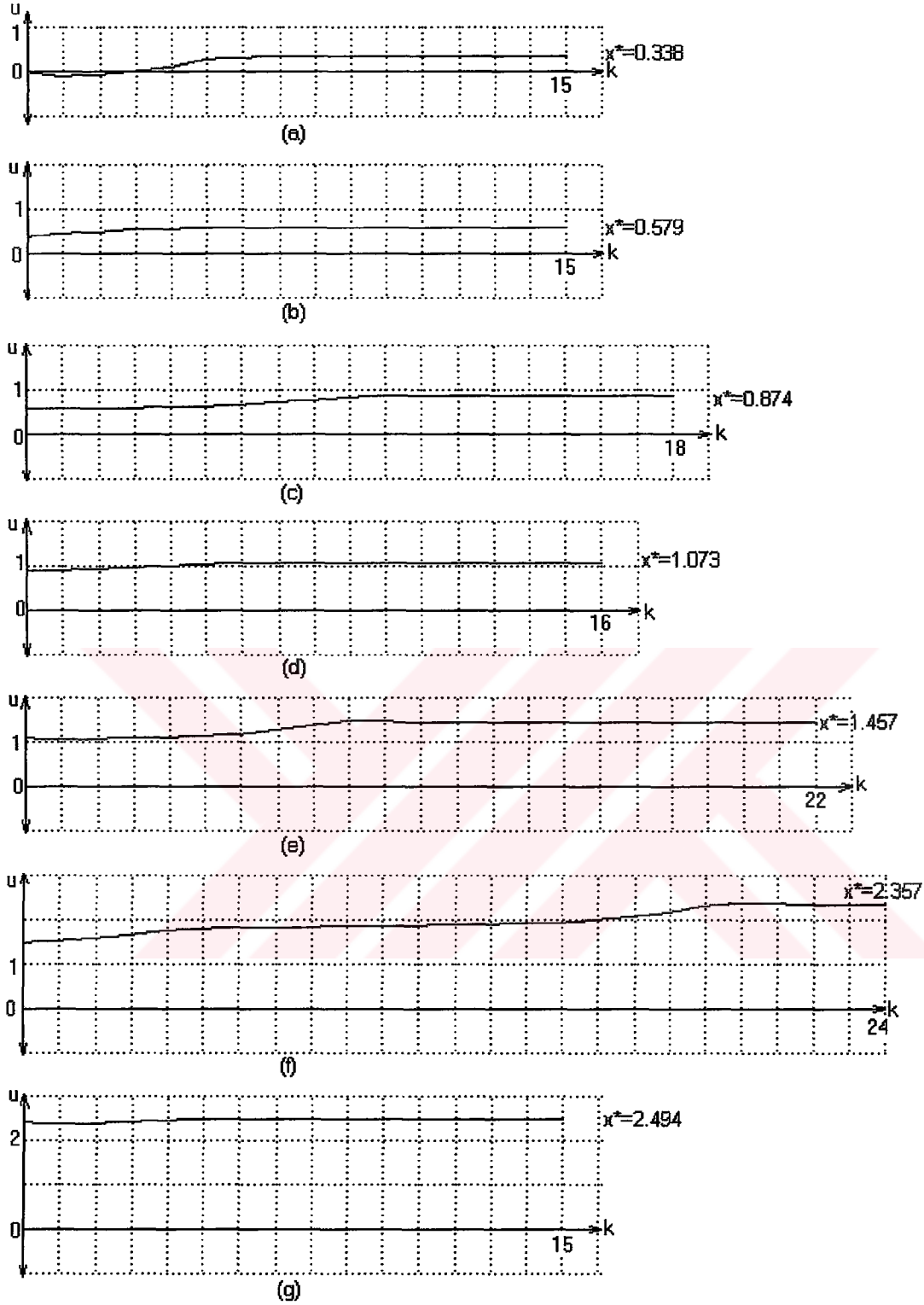
Şekil 4.10 a) $-x^2+3=0$ b) $-1.5x^2+4.5=0$ c) $-3x^2+9=0$ çözümlerinde kontrol işaretinin değişimi ($S_{ce}=0.9$, $S_{cde}=0.05$, $S_{cu}=0.3$).

Bulanık kontrolör ile kök aramaya ikinci örnek olarak $(-5,5)$ aralığında değişimi şekil 4.11’de görülen $f(x)=-x^2+2+3\sin(11x)+2\cos(5x)$ verilmiştir. $r=0$ için $f(x)-r=0$ çözümünde birden çok kök bulunmaktadır. Şekil 4.12’de görülen çevrimlerde arama aralığı $(0,+10)$ alınmış ve $x_0=uc(0)=0$ alınarak soldan sağa doğru (+ işaret yönünde) algoritma işletilmiştir. İlk köke $x^*=0.338335$ ’de ulaşılmıştır. Her ulaşılan kökten sonra yeni başlangıç koşulu olarak $x_0=x^*+\Delta$ alınarak algoritma tekrar işletilmiştir.



Şekil 4.11 $f(x)=-x^2+2+3\sin(11x)+2\cos(5x)$ fonksiyonunun $(-5,+5)$ aralığında değişimi

Pozitif yöndeki 7 adet kök sırasıyla $x_0=u(0)=0$ için $x^*=0.338335$, $x_0=u(0)=0.37$ için $x^*=0.5796005$, $x_0=u(0)=0.6$ için $x^*=0.8740202$, $x_0=u(0)=0.9$ için $x^*=1.073365$, $x_0=u(0)=1.1$ için $x^*=1.457320$, $x_0=u(0)=1.48$ için $x^*=2.356822$, $x_0=u(0)=2.39$ için $x^*=2.494206$ noktalarında köke ulaşılmıştır. Her çevrim sonunda yeni başlangıç koşulu belirlenirken $\Delta=0.03$ alınmıştır. En yakın iki kök arası uzaklık Δ 'dan küçük ise arada kalan köke erişilemez. Yapılan aramaların tümünde $S_{ce}=0.3$, $S_{cce}=0.05$, $S_{cu}=0.3$ seçilmiştir.



Şekil 4.12 $-x^2+2+3\sin(11x)+2\cos(5x)=0$ denkleminin çözümünde a) $x_0=uc(0)=0$, b) $x_0=uc(0)=0.37$, c) $x_0=uc(0)=0.6$, d) $x_0=uc(0)=0.9$, e) $x_0=uc(0)=1.1$, f) $x_0=uc(0)=1.48$, g) $x_0=uc(0)=2.39$ için kontrol işaretinin değişimi ($S_{ce}=0.3$, $S_{cde}=0.05$, $S_{cu}=0.3$, $\Delta=0.03$)

3.2 ayrıtında PI kontrolör kullanarak yerel ekstremumların bulunması anlatılmıştı. $f(x)$ yerine yaklaşık sayısal türev ifadesi konularak kök aramayı temel alan bu yöntem, şekil 4.5’de görülen bulanık kontrolörlü arama düzeni için de geçerlidir. $f(uc)$ yerine (3.12)’deki fark ifadesi kullanılarak $f'(x)=0$ denkleminin yaklaşık çözümü elde edilmektedir. Diğer bir deyişle başlangıç noktasından itibaren (+) kontrol işareti yönündeki ilk ekstremum sayısal olarak hesaplanabilmektedir. Bulanık kontrolör ile $f(x)=r$ ve $f'(x)=0$ denklemlerinin çözülmesinden faydalanılarak global optimum nokta hesaplanabilmektedir. 5.Bölümde tek ve çok değişkenli global eniyileme problemlerinde bu yöntemin kullanımı ele alınmaktadır.

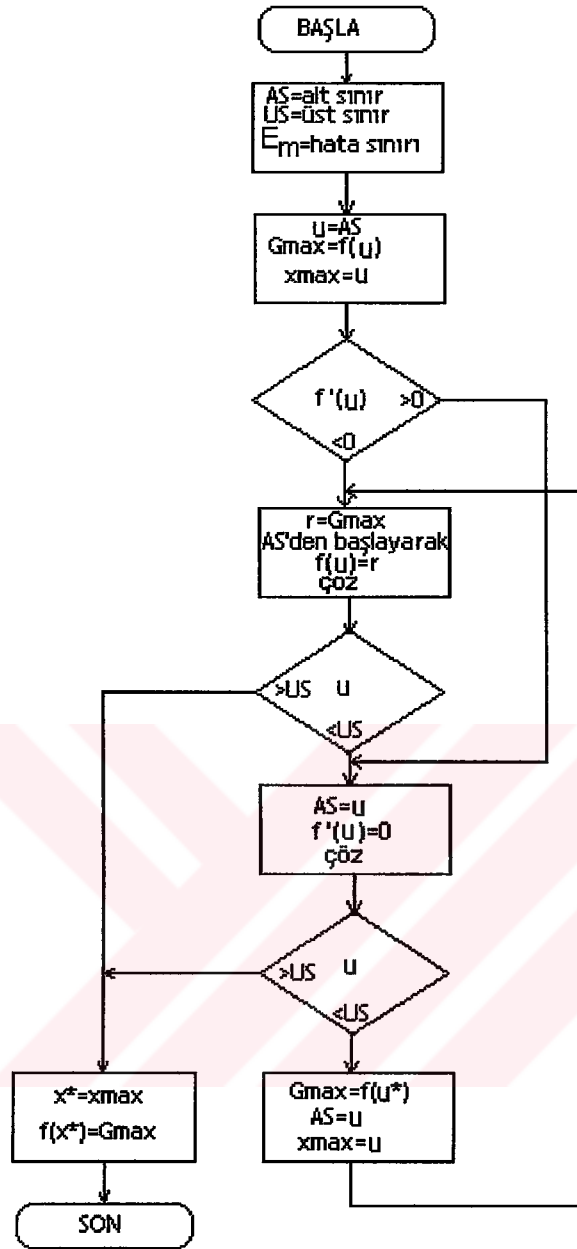


5. BULANIK KONTROLÖR İLE GLOBAL ENİYİLEME

3.Bölümde kapalı çevrim kontrol kuramının sayısal eniyileme problemine uygulanması ve 4.Bölümde ise sayısal aramada bulanık kontrolör kullanımı anlatılmıştı. Bu bölümde, kapalı çevrim kontrol tekniğinden faydalanılarak global eniyileme problemine yeni bir yaklaşım getirilecektir. Bu amaçla bulanık kontrolör ile kök arama ve ekstremum bulma alt işlemleri kullanılacaktır. Bir ve iki değişkenli amaç ölçüt fonksiyonları için, önerilen çözüm yönteminden yararlanarak çeşitli uygulama örnekleri verilecektir.

5.1 Bulanık Kontrolörle Arama Yapılarak Global Eniyi Noktanın Bulunması

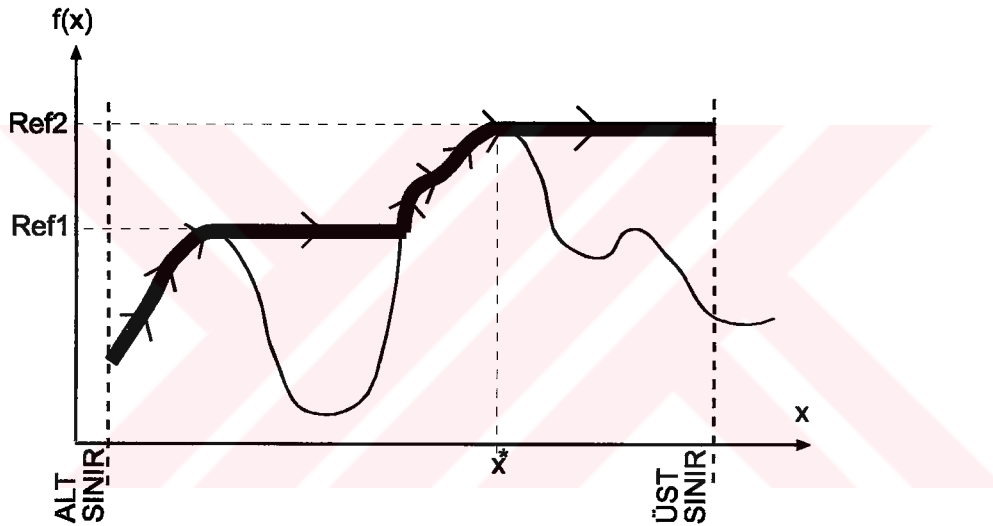
Bulanık kontrolör ile global eniyileme için çeşitli yaklaşımlar mümkündür. Burada önerilen eniyileme algoritması, bulanık kontrolörlü kapalı çevrim kontrol düzeni ile çözüm istenen aralığın taranması üzerinedir. Tarama işlemi, alt sınırdan üst sınıra, üst sınırdan alt sınıra, ya da çok başlangıç noktalı yapılabilir. Şekil 5.1’de görülen eniyileme algoritması alt sınırı başlangıç noktası kabul etmekte ve verilen bir $f(.)$ fonksiyonunun enbüyük değerini aramaktadır. Aramaya başlanan noktada fonksiyon artış yönünde ise $r=0$ için $f'(u)=0$ çözümü sağa doğru aranır ve bulunan ilk kök yeni global enbüyük noktası olarak kabul edilir. Aramaya başlanan noktada fonksiyon azalma yönünde ise, $r=f(u)$ için sağa doğru bulanık kontrolör ile kök aranır ve bulunan ilk kökten itibaren $r=0$ için $f'(u)=0$ çözümü elde edilerek varılan yeni nokta global enbüyük kabul edilir. Her iki halde de referans r =yeni global enbüyük, alınarak $f(u)=r$ çözümü ile tekrarlı olarak işleme devam edilir. Arama sırasında kontrol işareti $u>U_S$ koşulu oluşursa varılmış olan en son $f'(u)=0$ çözümü global enbüyüğü verecektir [27].



Şekil 5.1 Tek değişkenli amaç ölçüt fonksiyonları için global eniyileme algoritması.

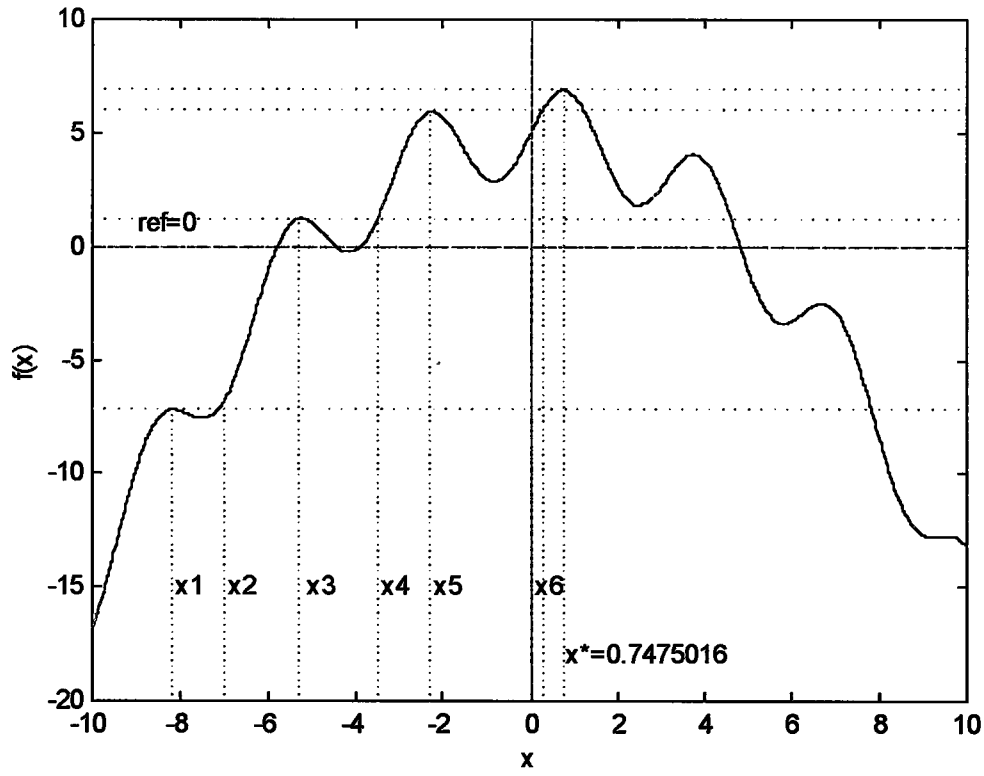
$f(x)=0$ alt işletiminde, sonlandırma kriteri olarak global aramada istenen duyarlılığı almak gerekmez. Çünkü, global noktalara $f'(x)=0$ çözümünün sonrasında ulaşılmaktadır. Örneğin, $E_m=0.000001$ hata üst sınırı ile çözüm isteniyorsa; kök arama çevrimlerinde $E_m=0.01$; türev arama çevrimlerinde ise $E_m=0.000001$ seçmek aynı doğrulukta daha kısa sürede çözüme ulaşmayı sağlayacaktır.

Önerilen çözüm tekniğinin bir fonksiyonun global enbüyük noktasının bulunması sırasında izleyeceği yol Şekil 5.2’de canlandırılmıştır. Burada fonksiyonun arama alt sınırındaki eğimi pozitif olduğundan $f'(x)=0$ çözümü aranarak işleme başlanmış ve ulaşılan değer (Ref1) referans alınmıştır. $f(x)=\text{Ref1}$ çözülmüş ve çözüm noktası başlangıç alınarak tekrar $f'(x)=0$ çözümüne geçilmiştir. Yeni ulaşılan nokta referans (Ref2) alınarak $f(x)=\text{Ref2}$ denklemi çözdürülür. Bu aramada kontrol işareti köke ulaşmadan arama üst sınırına erişeceğinden döngü sonlandırılır. Verilen aralıktaki global eniyi değer olarak Ref2 ve bunun karşılığı kontrol değişkeni olarak da x^* bulunmuş olur.

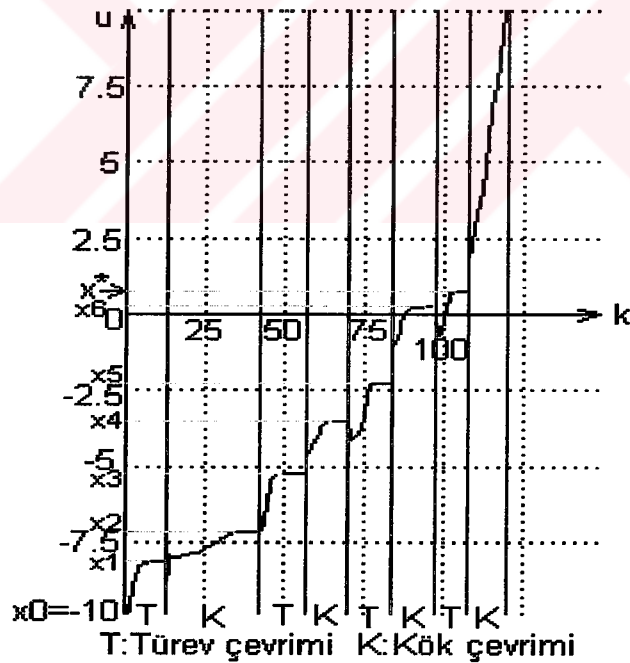


Şekil 5.2- Tek değişken için global enbüyük noktanın bulunması.

Şekil 5.3’de değişimi verilen $f(x)=-0.2x^2+5+2\sin(2x)$ fonksiyonu için $(-10,10)$ aralığında global enbüyük aranmasında -10 noktası başlangıç noktası alınmıştır. $f'(-10)>0$ olduğundan aramaya türev çevrimi ile başlanmıştır. Şekil 5.4’de görüleceği üzere dördüncü türev çevrimi sonunda $x^*=0.7475016$ da global enbüyük olan $f(x^*)=6.882506$ değerine $E_m=0.00001$ hata üst sınırı ile ulaşılmıştır. $f(x^*)$ referans alınarak yapılan bir sonraki kök çevriminde, toplam 120 iterasyon sonunda $u>+10$ değeri ile doyma olduğundan işlem sonlandırılmıştır (Şekil 5.4). Buradaki kök arama çevrimlerinde $S_{ce}=0.6$, $S_{cce}=0.2$, $S_{cu}=0.3$; türev çevrimlerinde ise $S_{ce}=0.4$, $S_{cce}=0.1$, $S_{cu}=0.2$ alınarak işlem yapılmıştır.



Şekil 5.3 $(-10,10)$ aralığında $f(x)=-0.2x^2+5+2\sin(2x)$ fonksiyonu.

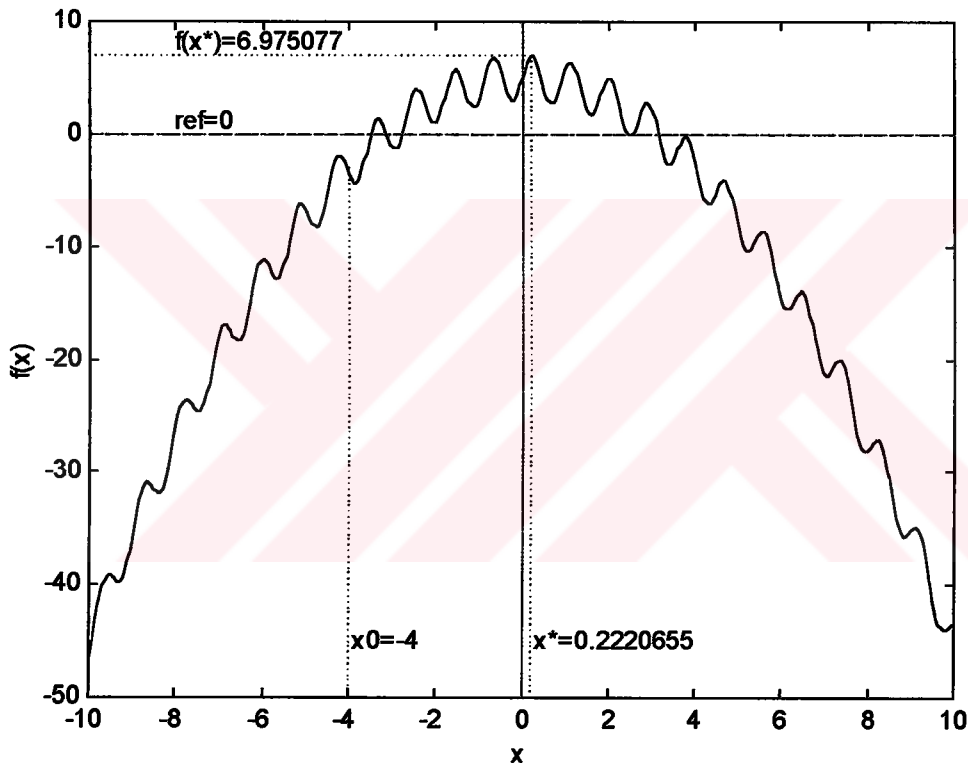


Şekil 5.4 $(-10,10)$ aralığında $f(x)=-0.2x^2+5+2\sin(2x)$ fonksiyonunun global enbüyüğünün hesaplanmasında u kontrolör çıkışının değişimi.

Diğer bir bakış açısıyla, kontrol işareti doymaya ulaşmadan önceki son türev çevriminde ulaşılan nokta $x^*=0.7475016$ (şekil 5.4), global enbüyük değeri

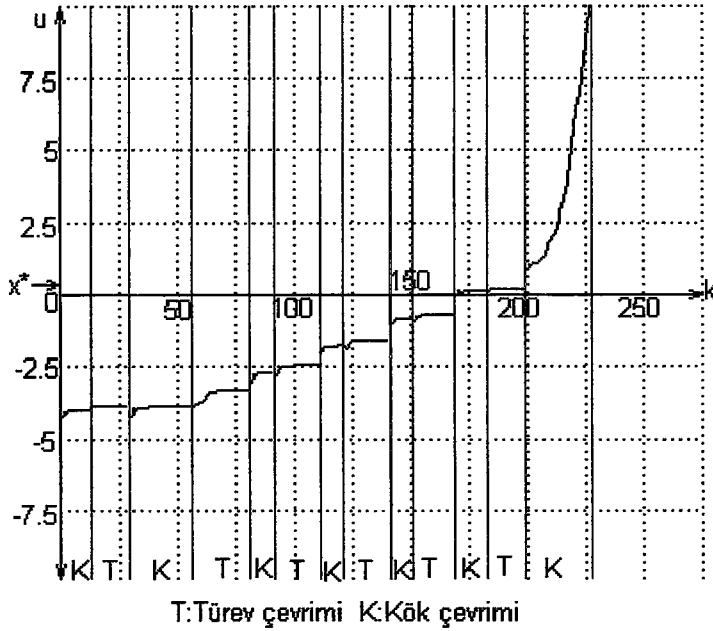
vermektedir. Kontrol işareti, arama sınırına (doyma) türev çevrimi sırasında ulaşırsa, arama sınırınının aynı zamanda global enbüyük nokta olduğu sonucu ortaya çıkar.

İkinci örnek olarak ele alınan $f(x)=0.5x^2+5+2\sin 7x$ (Şekil 5.5), Şekil 5.3'deki fonksiyonun frekansı arttırılmış bir benzeridir. $x=0$ civarında birbirine yakın enbüyükler içerdiğinden daha düşük ölçekleme katsayıları ile doğru sonuca erişilmektedir. Şekil 5.6'da görüleceği üzere global enbüyük değeri olan $x^*=0.2220655$ 'deki $f(x^*)=6.975077$ değerine, $x_0=-4$ başlangıç koşulu ile, toplam 228 iterasyonda ulaşılmıştır. Burada, $E_m=0.00001$, kök çevrimlerinde $Sce=0.5$, $Scce=0.2$, $Scu=0.2$; türev çevrimlerinde $Sce=0.4$, $Scce=0.1$, $Scu=0.05$ değerleri kullanılmıştır



Şekil 5.5 (-10,10) aralığında $f(x)=0.5x^2+5+2\sin(7x)$ fonksiyonu.

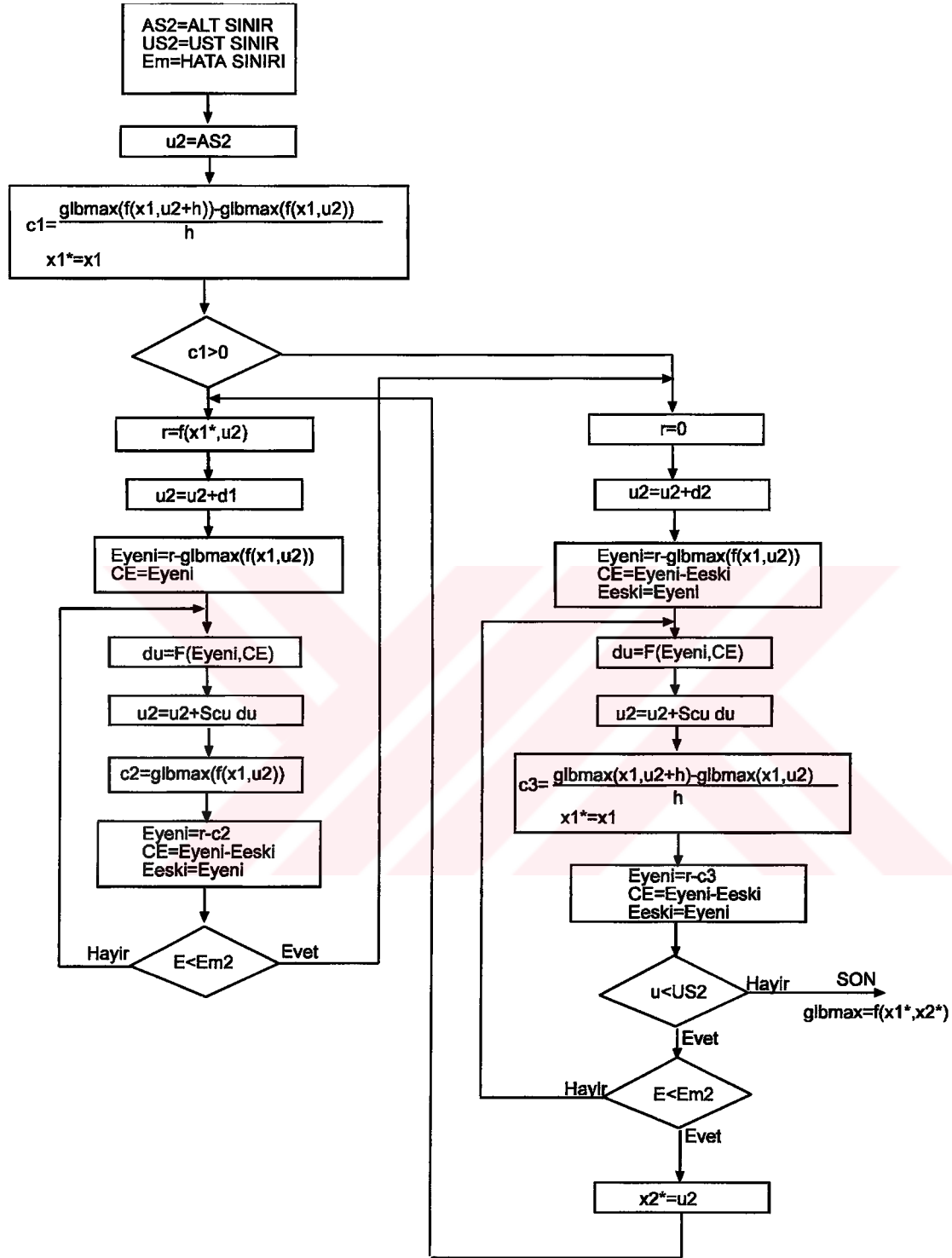
Yüksek çıkış ölçeklendirme katsayısı (Scu) kullanılması durumunda sonuca erişmedeki döngü sayısı düşmekte fakat birbirine yakın enbüyüklerden global olanı atlanabilmektedir.



Şekil 5.6 $f(x)=-0.5x^2+5+2\sin(7x)$ fonksiyonunun $(-4,10)$ aralığında global enbüyüğünün hesaplanmasında u kontrolör çıkışının değişimi.

5.2 Çok Değişkenli Fonksiyonların Global Eniyilenmesi

Çok değişkenli fonksiyonların bulanık kontrolörle global eniyi noktasının bulunmasında (2.15) ve (2.16)'daki boyut indirgemesinden faydalanılmaktadır. Buna göre, n değişkenli bir fonksiyonun global enbüyük noktasının bulunmasında, sırasıyla her değişken için, diğer tüm değişkenler sabit tutularak, arama yapılır. Değişken sayısı ile değerlendirme sayısı üstel artacak gibi gözükse de her değişken için ilk kök arama referansı bir önceki değişken adımına kadar bulunmuş eniyi nokta olarak değerlendirildiğinden, erişilmiştense daha düşük genlikli noktalar daha hızlı taranır. Şekil 5.7'de iki değişken için arama algoritması verilmiştir. Buradaki bloklarda yer alan $glbmax(f(x_1, u_2))$, x_1 değişken ve $x_2=u_2=sabit$ için $f(x_1, x_2)$ fonksiyonunun enbüyük değerini tek değişkenli arama yöntemi ile hesaplar. Aramaya u_2 , x_2 'nin alt sınırı alınarak başlanır ve u_2 'nin artışı ile x_1 yönünde hesaplanan global enbüyükün artış gösterip göstermediğine bakılır. Eğer azalma yönünde ise $f(x_1^*, u_2)$ referans alınarak u_2 yönünde kök aranır. Daha sonra sıfır referans için $f'(x_1, x_2)=0$ çözülür. Bu çözüm sırasında, hataya (Eyen) ve hatanın değişimine bağlı olarak $x_2=u_2$ 'nin artışı belirlenir ve en son erişilen u_2 değeri x_2^* olarak kaydedilir. Çözüme ulaşmadan önce x_2 değişkeni için üst sınır olan US_2 'ye



Şekil 5.7 İki değişkenli $f(x1, x2)$ fonksiyonu için bulanık mantıkla global enbüyük arama algoritması.

gelinmişse $x_2^*=US_2$ ve global enbüyük $glbmax(f(x_1,US_2))$ olacaktır. x_2 yönündeki $f(x_1,x_2)=r$ çözümlerinde sonlandırma için kullanılan hata üst sınırı E_{m1} 'dir. x_2 yönündeki $f'(x_1,x_2)=0$ aramasında ise sonlandırma koşulu olarak hatanın E_{m2} 'den küçük olmasına bakılır. Belirli bir E_m hata üst sınırı için kök aranıyorsa, sadece türev çevrimleri sonunda yeni referans noktalar elde edildiğinden $E_{m2}=E_m$, $E_{m1}>E_m$ alınması aramayı hızlandırır. E_{m1} 'in çok büyük seçilmesi (Örneğin 1000 E_m) doğru çözüme ulaşma güvenilirliğini etkileyebilir. Yapılan örneklerde verilen bir E_m 'ye bağlı olarak, çeşitli deneyler sonucunda

$$E_{m2}=E_m \quad (5.1)$$

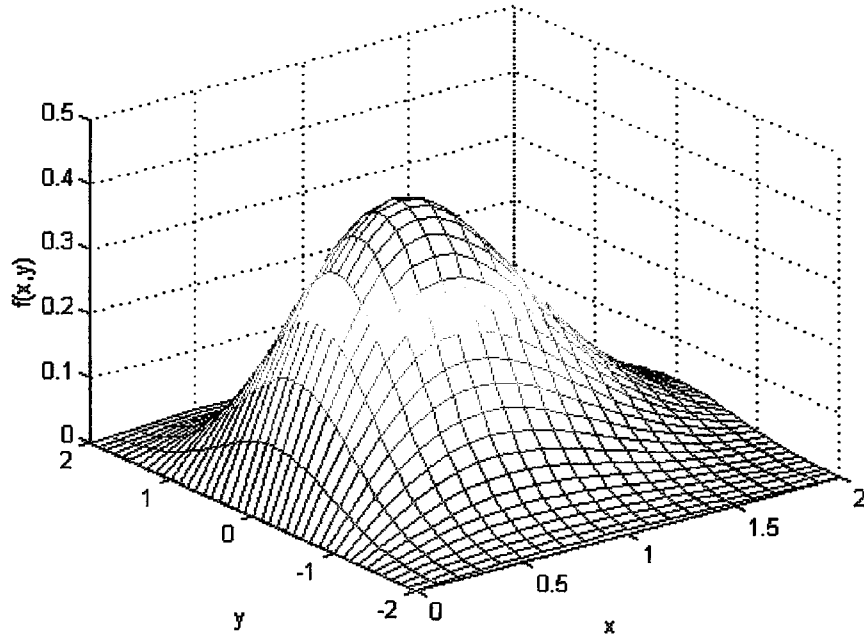
$$E_{m1}=5(US-AS) E_m, (US-AS)>1 \quad (5.2)$$

olarak seçilmiştir. (US-AS) değeri birden küçükse (5.2) kriterindeki hata sınırını düşürmek gerekir. Alt çevrimler kapalı çevrim kontrol kuramı yaklaşımı ile yapıldığından, kök aramada ilk çevrimde hata sıfır olursa çevrim sonlanacaktır. Amaç bir sonraki kökü bulmak olduğundan ilk değerlendirmede sonlandırma koşulunun sağlanmaması için kontrol işareti d_1 kadar arttırılarak işleme başlanır. Arama güvenliği bakımından, d_1 , en yakın iki kök arası olabilecek mesafeden küçük seçilir. Fonksiyonun topolojisi hakkında bir ön bilgi yoksa,

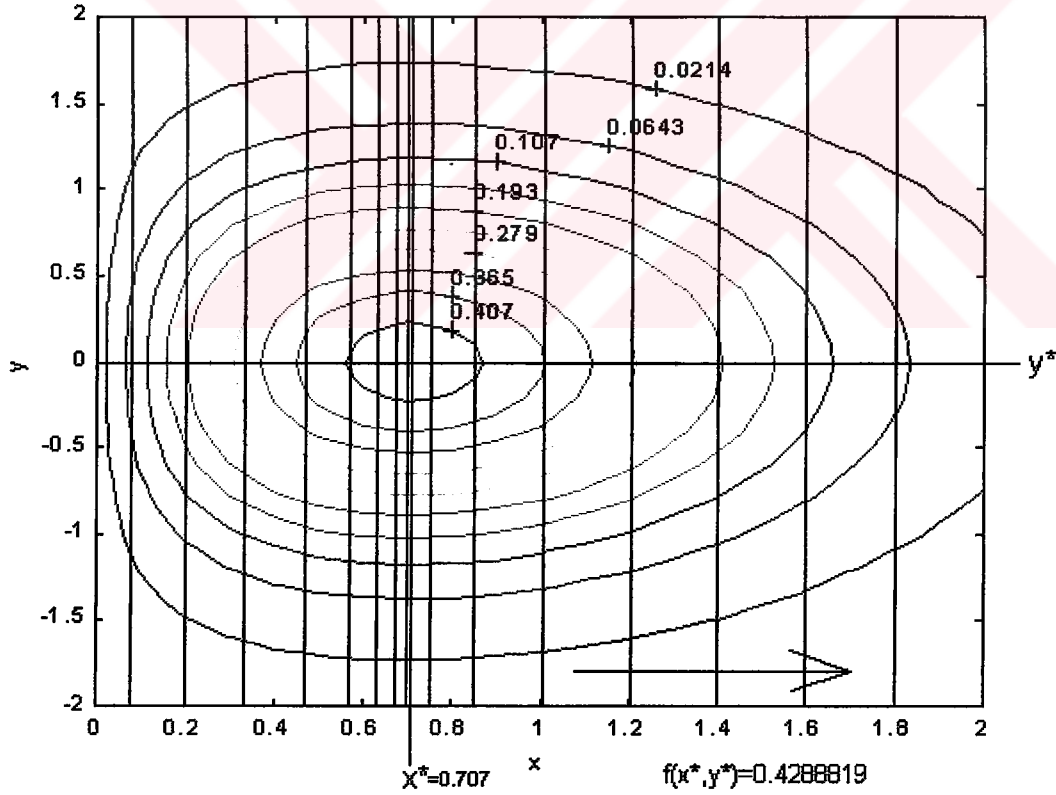
$$d_1=2E_m \quad (5.3)$$

seçilebilir. d_1 'in (5.3)'e göre seçilmesi başlangıç hatasının küçüklüğü sebebiyle işlem sayısını arttıracaktır. Diğer bir bakış açısıyla, önerilen yöntem, bulanık kontrolörle aralıkların değiştirildiği ızgara aramasına benzetilebilir. Şekil 5.8-a'da $f(x,y) = x e^{-x^2-y^2}$ fonksiyonunun $[(0,2),(-2,2)]$ aralığında değişimi görülmektedir. Verilen aralıkta tek ekstremum olduğundan bu fonksiyonun yerel ekstremumu ile global ekstremumu çakışıktır. Bu sebeple yerel arama yöntemleri daha az sayıda işlemde sonuca erişebilir. Canlandırmanın basitliği bakımından global aramada bu örnek seçilmiştir. $(x,y)=(0,-2)$ noktasından başlayarak arama yapılırsa

$f(x+h,y)>f(x,y)$ olduğundan $\frac{\partial f(x,y)}{\partial x} < E_m$ çözümü aranır. Bu amaçla $x=\text{sabit}$ için y



(a)



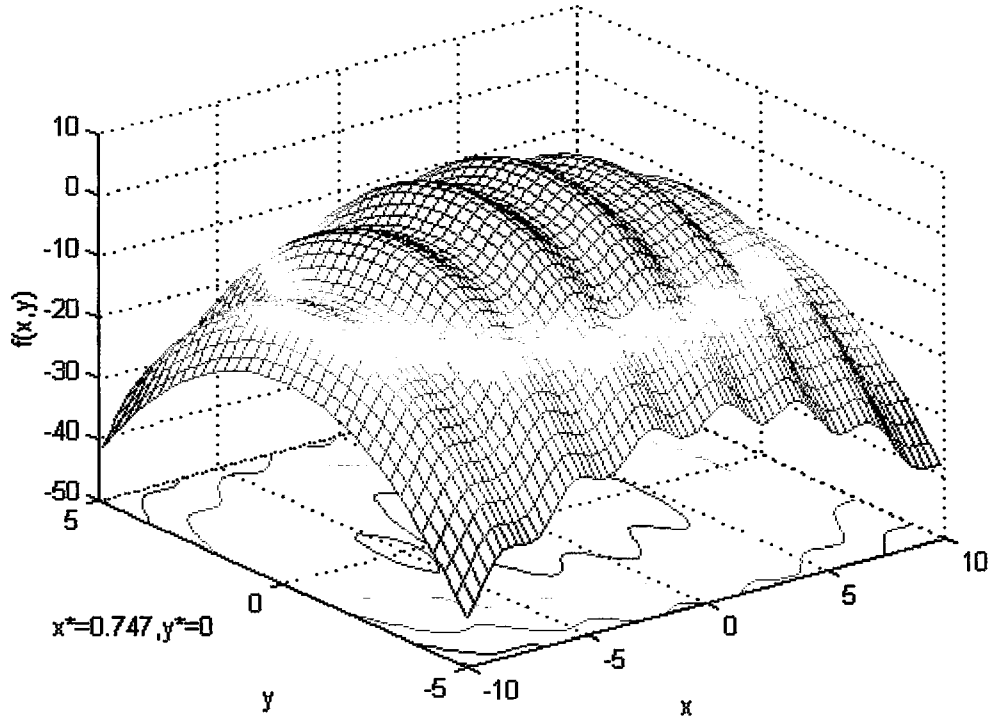
(b)

Şekil 5.8 a) $f(x, y) = x e^{-x^2-y^2}$ ifadesinin $[(0,+2),(-2,+2)]$ aralığında değişimi b) y yönünde tek değişkenli arama yapılırken x yönünde atılan adımların canlandırılması.

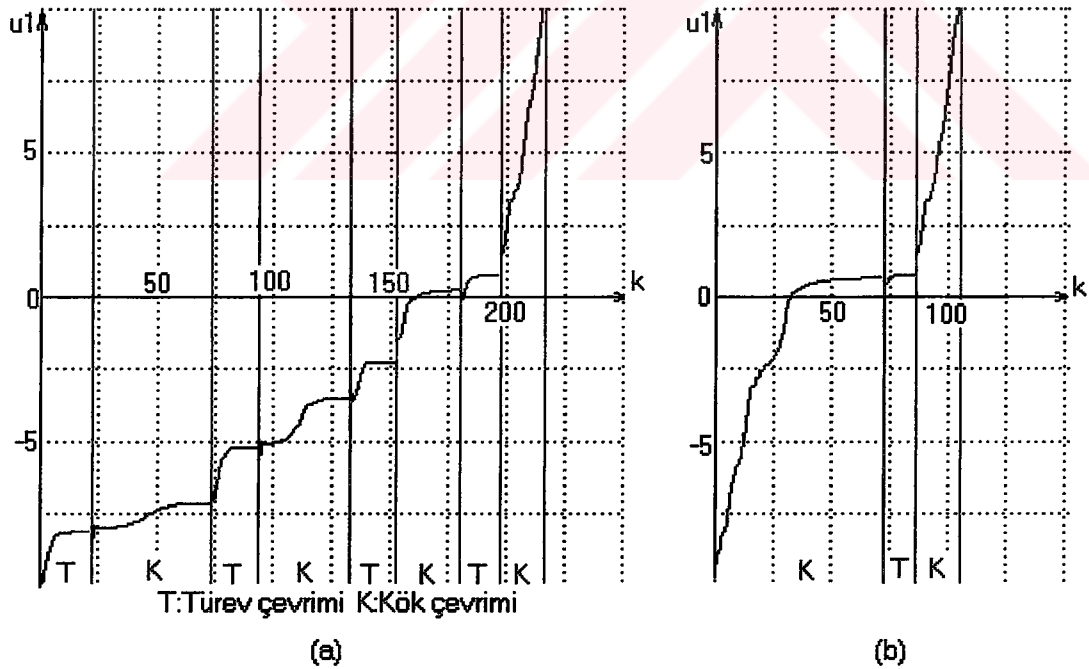
yönünde tek değişkenli global arama yapılır ve $x+h$ için tekrarlanarak x yönündeki artış belirlenir. Sonlandırma koşulu sağlanıncaya kadar yeni x değerleri ile işleme devam edilir. Şekil (5.8)'deki örnekte $\frac{\partial f(x, y)}{\partial x} < E_m$ şartı sağlandıktan sonra $r=f(x^*, y^*)$ alınarak kök aramasına geçilmiştir. x yönünde yeni kök bulunamadan üst sınıra ulaşıldığından son ulaşılan (x^*, y^*) değerleri global enbüyüğü veren nokta olarak kaydedilir. Şekil 5.8-b'de y eksenine paralel çizgiler sabit x değerleri için y yönünde global aramayı temsil etmektedir. $x < x^*$ için x yönünde global enbüyükün değişimi incelendiğinden her arama $f(x+h, y)$ için tekrarlanır. Sabit bir x için arama yapıldıktan sonra elde edilen global enbüyük, $(x+h)$ için yapılan aramada başlangıç referansı olarak kullanıldığından, bu arama, işlem sayısı bakımından çabuk bitecektir. Şekil 5.8-b'deki canlandırmada x eksenini yönünde ekstremuma yaklaşıldıkça y eksenini yönündeki aramaları temsil eden çizgilerin sıklaştığı görülmektedir.

Şekil 5.9'da görülen $f(x, y) = -0.2x^2 + 2\sin(2x) - y^2 + 5$ fonksiyonu y değişkenine göre türevi bir defa işaret değiştirmektedir. Yapılan uygulamada x doğrultusunda tek değişkenli global arama yapılırken şekil 5.7'deki algoritmaya bağlı olarak y doğrultusunda atılan adımlar hesaplatılmıştır. Bu aramalarda sonlandırma kriteri olarak $E_{m1} = 0.001$, $E_{m2} = 0.00001$ alınmıştır. Toplam 2688 değerlendirmede $(x_0, y_0) = (-10, -5)$ başlangıç koşulu ile $(x^*, y^*) = (0.747565, 0)$ noktasına erişilmiştir.

$\frac{\partial f(x, y)}{\partial y} = 0 \pm E_{m2}$ elde edilinceye kadar belirli bir y değeri için x doğrultusunda yapılan aramalar $y+h$ için tekrarlanmıştır. Şekil 5.10'da $y = -2$ ve $y = -2+h = -1.9999$ için bu arama çiftlerinden biri örnek olarak verilmiştir. $y = -2$ için toplam 215 çevrimde $x^* = 0.747565$ 'deki (doyma öncesi son türev çevriminin kökü) enbüyük elde edilmiştir. $r = f(0.747565, 2)$ ve $y = -2+h$ alınarak yapılan aramada yeni global enbüyük değer referansa yakın olduğundan 85 çevrimde sonuca erişilmiştir. Burada u_1 , $+x$ yönündeki aramada kontrolör çıkış işaretidir. n tane değişken için eniyileme yapılmak istenirse $u_1, u_2, \dots, u_n$ olmak üzere n tane kontrolör çıkışı kullanmak gerekecektir. Bir değişken yönünde elde edilen ekstremum diğer değişken yönündeki ilerlemede referans alındığından değişken sayısı ile değerlendirme sayısı üstel olarak artmamıştır.



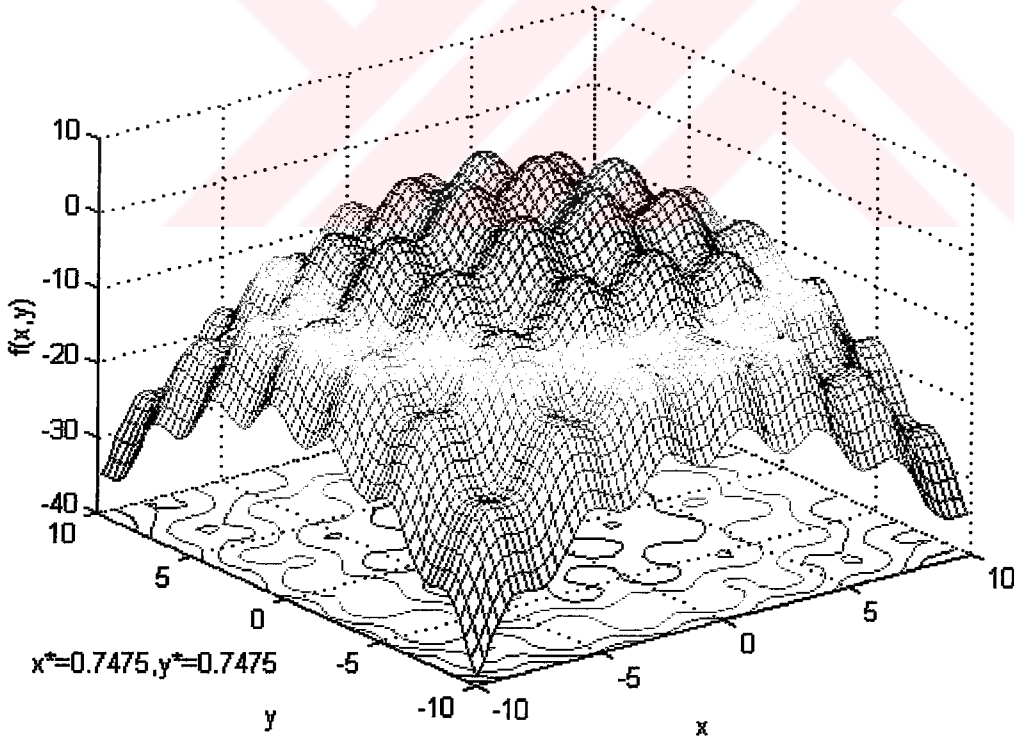
Şekil 5.9 $f(x,y)=-0.2x^2+2\sin(2x)-y^2+5$ fonksiyonunun $[(-10,+10),(-5,5)]$ aralığında değişimi.



Şekil 5.10 a) $y=-2$ için x yönünde global enbüyüğün aranması b) (a)'daki arama sonucu referans alınarak $y=-2+h=1.9999$ için x yönünde arama.

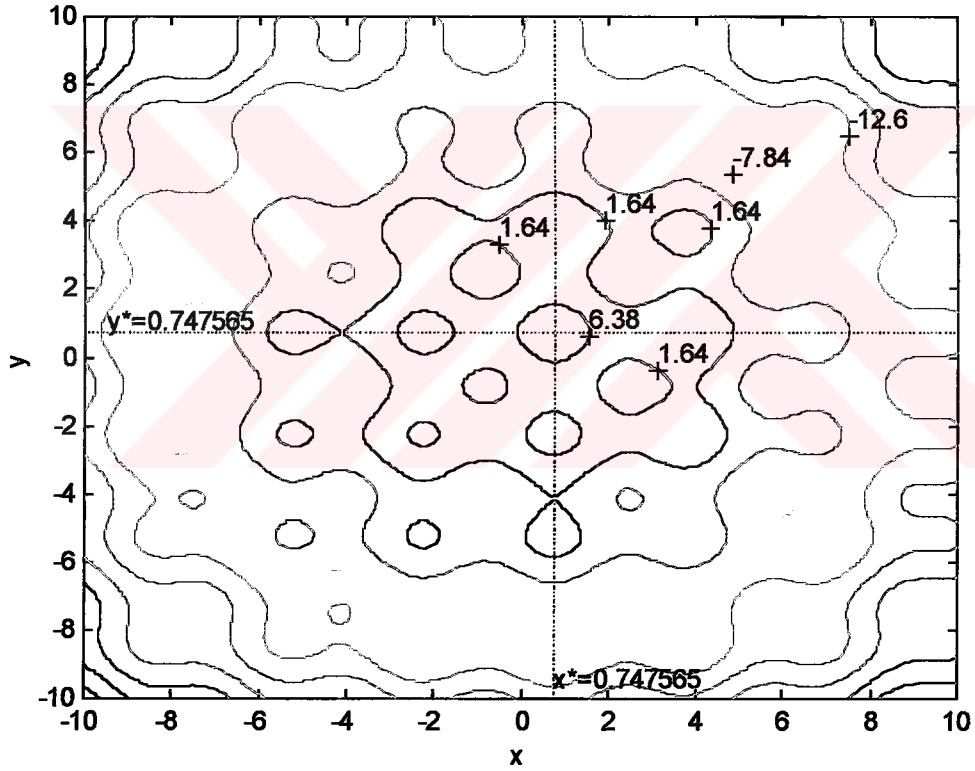
Yapılan aramalarda, türev çevrimlerindeki ölçeklendirme değerleri $S_{ce}=0.7$, $S_{cce}=0.05$, $S_{cu}=0.15$, kök çevrimlerindeki ise $S_{ce}=0.9$, $S_{cce}=1.5$, $S_{cu}=0.2$ olarak alınmıştır. Bu ölçeklendirme değerleri denemeler sonucu elde edilmiş değerlerdir. 3.Bölümde de açıklandığı gibi belirli bir fonksiyon için sonuca daha az işlem sayısında ulaşılabilecek başka ölçeklendirme değerleri mevcuttur. Global enbüyük/enküçük araması yapılacak fonksiyon için bir Lipchitz değeri verilebiliyorsa sonuca erişmeyi garanti edecek ölçeklendirme değerleri hesaplanabilir. Ancak bu değerlerin kullanılması işlem sayısını da arttıracaktır. Diğer bir deyişle uygulamada ihmal edilebilecek bir risk ile fakat çok daha az işlem sayısında aynı sonuca erişmek mümkündür.

Şekil 5.11'de görülen $f(x,y)=-0.2x^2+2\sin(2x)-0.2y^2+2\sin(2y)+5$ fonksiyonunun türevi, her iki değişken için $[(-10,+10),(-10,+10)]$ aralığında 6'defa işaret değiştirmektedir. $(x_0,y_0)=(-10,-10)$ başlangıç koşulu ile $(x^*,y^*)=(0.747565,0.747565)$ noktası 3995 çevrimde elde edilerek sınırlara ulaşılmış ve arama sonlandırılmıştır.



Şekil 5.11 $f(x,y)=-0.2x^2+2\sin(2x)-0.2y^2+2\sin(2y)+5$ fonksiyonunun $[(-10,-10),(10,10)]$ aralığında değişimi.

Bu aramada herbir deęişken doęrultusunda dördüncü türev ve üçer defa kök arama işlemi yapılmıştır. y yönündeki dördüncü $\frac{\partial f(x,y)}{\partial y}=0 \pm E_{m2}$ çözümü ile y^* elde edilmiş; $y=y^*=sabit$ için dördüncü $\frac{\partial f(x,y)}{\partial x}=0 \pm E_{m2}$ elde edildiğinde x^* 'a ulaşılmıştır. $r=f(x^*,y^*)$ alınarak $f(x,y)-r=0$ (kök arama) çevrimlerine devam edilmiş ancak yeni bir köke rastlanmadan (10,10) üst sınırına ulaşılmıştır. Buna baęlı olarak, üst sınıra erişmeden önceki türev çevrimlerinde elde edilen $(x^*,y^*)=(0.747565,0.747565)$ noktası global enbüyük olduęu sonucu ortaya çıkmıştır. Bu durumda $f^*(x^*,y^*)\cong 8.76501$ olarak hesaplanır.



Şekil 5.12 $f(x,y)=-0.2x^2+2\sin(2x)-0.2y^2+2\sin(2y)+5$ fonksiyonunun eşyükselti çizimi.

5.3 Standart Test Fonksiyonları

Global eniyileme problemlerinde yineleme sayısı, kullanılan yönteme ve eniyilenen fonksiyona bağlı olarak değişkenlik göstermektedir. Tablo 5.1’de karşılaştırılması yapılan tek değişkenli fonksiyonlar diğer yöntemlerin başarılı oldukları arasından seçilmiştir.

Tablo 5.1 – Global enküçük araması yapılan bazı test fonksiyonları ve literatürde yer alan bazı algoritmaların sonuca erişimdeki değerlendirme sayıları

f(x)	Aralık	Bulanık kontrolör	CRS2 (N=15)	Genetik Algoritma	Pijav	Batish
$f_1(x)$	(2.7 , 7.5)	57	85	2000	462	120
$f_2(x)$	(3.1 , 20)	68	90	1800	448	158
$f_3(x)$	(-10 , 10)	159	95x3	2500	3817	816

Tablo 5.1’de yer alan fonksiyonlar,

$$f_1(x) = \sin x + \sin(10x/3) + \ln x - 0.84x$$

$$f_2(x) = \sin x + \sin(2x/3)$$

$$f_3(x) = - \sum_{i=1}^5 \sin((i+1)x + i)$$

Bu test fonksiyonları hakkında ayrıntılı bilgi Ek C’de mevcuttur. Algoritmaların bir çevrimlerindeki işlem sayıları farklı olduğundan tablo 5.1’de görülen değerler sonuca erişme süreleri bakımından doğrudan bir karşılaştırma yapmaya uygun değildir. Zamana bağlı karşılaştırmalarda çalışılan bilgisayarın donanım özellikleri, işletim sistemi, programın derleyicisi ve kod uzunluğuna bağlı derleme parametreleri de göz önünde tutulmalıdır. Pentium II 366Mhz mikroişlemcili bir makinada, MS-DOS işletim sisteminde $f_1(x)$ ’in bulanık kontrolörle çözümü 1.1s tutarken CRS2 ile çözümü 850ms almıştır. Aynı koşullarda $f_2(x)$ olarak ifade edilen test fonksiyonunun eniyilenmesi bulanık kontrolörle 1.2s, alırken CRS2 ile 950ms’de çözüme erişilmiştir. Derleme işlemi QuickBasic 7.0 ile yapılmıştır.

Optimizasyon deęiřkenleri üzerine getirilebilecek kısıtlamalara uymak için, amaç ölçüt fonksiyonuna ceza bileřeni eklenebildięinden (penalty function) [29] burada ayrıca ele alınmamıřtır.

Tablo 5.2’de iki deęiřkenli bazı test fonksiyonları için karřılařtırma yer almaktadır [28].

Tablo 5.2 Literatüdeki bazı global arama algoritmalarının iki deęiřkenli test fonksiyonlarında $E_m=0.00001$ hassasiyetle yapılan arama sayılarının karřılařtırmaları

$f(x,y)$	Aralık	Bulanık kontrolör	Kontrollü raslantısal arama (CRS)	Genetik algoritma
$f_4(x,y)$	$-5 \leq x \leq 5, -5 \leq y \leq 5$	3981	2000	3000
$f_5(x,y)$	$-2 \leq x \leq 2, 2 \leq y \leq 2$	1970	2500 (CRS2’de 914)	2500
$f_6(x,y)$	$-5 \leq x \leq 10, 0 \leq y \leq 15$	2400	2500 (CRS2’de 670)	3000

$f_4(x,y)=4x^2-2.1x^4+x^6/3+xy-4y^2+4y^4$ (altı kamburlu deve sırtı fonksiyonu) [30]

$f_5(x,y)=[1+(x+y+1)^2(19-14x+3x^2+14y+6xy+3y^2)] \times$

$[30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2)]$ (Goldstein-Price test fonksiyonu) [31]

$f_6(x,y) = (y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x + 10$ (Branin test fonksiyonu)

5.4 Paralel İşletime Uygunluk

Günümüzde bilgisayarların donanım maliyeti düştükçe, bir yapı içerisinde birden fazla işlemci kullanımı ya da programların şebeke üzerinden yük paylaşımı çalıştırılması gündeme gelmiştir. Böylelikle donanım maliyetine katlanıldığı oranda hızlı işletim sağlanabilmektedir. Özellikle çözüm için gerekli işlem sayısı çok yüksek olan problemlerde, sayısal hesaplama algoritmasının paralel işletime uygunluğu önem kazanır. Burada önerilen algoritmayı bir şebek üzerinden birden fazla bilgisayarda işletmek mümkündür. Kullanıcı terminalinin bağlı bulunduğu ana makinaya eniyileme yapılacak fonksiyon ve arama aralığı girilir. İşletimin başlatılması komutu ile ana makina sistem üzerindeki köle makinalara fonksiyonu ve $a_i, b_i (i=1,2,...,m)$ arama aralıklarını yükler. $f(x_1, x_2, ..., x_m)$ fonksiyonundaki değişkenler,

$$a_1 \leq x_1 \leq b_1 \quad (5.4)$$

$$a_2 \leq x_2 \leq b_2$$

.....

$$a_m \leq x_m \leq b_m$$

aralıklarında değişiyor olsun. Eğer n adet makina (bilgisayar ya işlemci kartı) kullanılırsa ve şebeke üzerindeki makinaların işletim hızları birbirine eşitse, birinci makinanın payına düşen arama aralığı,

$$a_1 \leq x_1 < \frac{b_1 - a_1}{n} + a_1 \quad (5.5)$$

$$a_2 \leq x_2 < \frac{b_2 - a_2}{n} + a_2$$

.....

$$a_m \leq x_m < \frac{b_m - a_m}{n} + a_m$$

ikinci makinanın payına düşen arama aralığı,

$$\frac{b_1 - a_1}{n} + a_1 \leq x_1 < 2 \frac{b_1 - a_1}{n} + a_1 \quad (5.6)$$

$$\frac{b_2 - a_2}{n} + a_2 \leq x_2 < 2 \frac{b_2 - a_2}{n} + a_2$$

.....

$$\frac{b_m - a_m}{n} + a_m \leq x_m < 2 \frac{b_m - a_m}{n} + a_m$$

n.makinanın payına düşen arama aralığı,

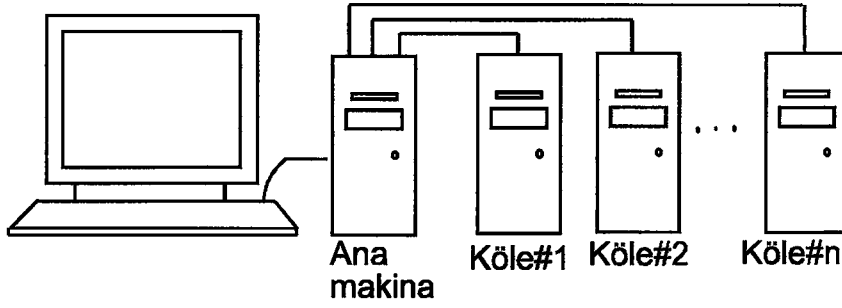
$$(n-1) \frac{b_1 - a_1}{n} + a_1 \leq x_1 \leq b_1 \quad (5.7)$$

$$(n-1) \frac{b_2 - a_2}{n} + a_2 \leq x_2 \leq b_2$$

.....

$$(n-1) \frac{b_m - a_m}{n} + a_m \leq x_m \leq b_m$$

olarak seçilmelidir. Şebeke üzerindeki makinaların birbirinden farklı hızlara sahip olması durumunda, (5.5),(5.6) ve (5.7)'deki homojen aralık seçimi sonucu hızlı makinalar kendi işlerini bitirecek işlemin sonuçlanması için en yavaş makinanın işini bitirmesini bekleyeceklerdir. Böyle bir durumda arama aralıklarını işletimdeki makinaların hızları ile orantılı dağıtmak uygun olacaktır.



Şekil 5.13 Çoklu işletimde kölelerin ana makinaya yıldız bağlantı durumu.

Her türev çevrimi sonunda, köle makinalar, elde ettikleri yeni global ekstremumu ana makinaya gönderirler. Her kök çevrimi başında, köle makinalar, ana makineden gönderilmiş en büyük (enbüyük arama için) değeri alırlar. Eğer bir köle makinanın kendi kök arama referans seviyesi ana makineden gelenden daha küçük ise, yeni gelen değer ile işleme devam eder. Böylelikle, algoritmanın sonuca erişme süresi, fonksiyonun yapısına da bağlı olarak köle makina sayısından daha büyük bir oranda kısalır. Şekil 5.13'te n tane köle makinada işlemi sağlayacak bir yapı görülmektedir. Makinalara paylaşımlı algoritma işletimi, internet üzerinden ya da ethernet hattı ile bağlı yerel şebeke üzerinden gerçekleştirilebilir.

Bu bölümde önerilen global eniyileme yönteminde, $f(x)$ için bir Lipchitz değeri verilebilmesi durumunda algoritmanın kararlılığı ve çözümün doğruluğu garanti edilir. Fonksiyonun süreksiz olması durumunda, kuramsal olarak doğru çözümü hiç bir yöntem garanti etmez. Açıklanan yöntemin standart test fonksiyonlarında başarıya ulaştığı görülmektedir. Standart test fonksiyonlarında değerlendirme sayısı bakımından, önerilen yöntemin mevcut yöntemlere göre etkinliğinin düştüğü görülür. Ancak karşılaştırma yapılan diğer arama yöntemleri raslantısallık içermekte ve sonlandırma kriteri olarak işlem sayısını kullanmaktadır. Oysa bulanık kontrolör ile çözümde istenilen duyarlılık sınırında arama yapmak mümkündür. Diğer taraftan değişken sayısının artışı ile algoritmanın karmaşıklaşması sorunu ile karşılaşılmaktadır. Paralel işletim ile arama yükü işlemciler arasında dağıtılarak sonuca erişme süresi kısaltılabilir.

6. SONUÇLAR VE TARTIŞMA

Bu tezde global eniyileme problemine yeni bir bakış açısı ve çözüm yöntemi getirilmiştir. 3.Bölümde sayısal kök arama ve yerel eniyileme problemlerinin kapalı çevrim kontrol kuramı yaklaşımı ile nasıl çözülebileceği gösterilmiştir. Mevcut sayısal arama yöntemlerinin bir çoğu bir kontrol düzeni ile modellenebilmektedir. Bu yaklaşımla değişik fonksiyonlar için çeşitli algoritmaların kararlılıklarını incelemek mümkündür. Böylelikle geleneksel algoritmaların performansını iyileştirmek üzere parametrik ayarlamalar yapılabilir. Bu tezde, ilk defa bir hesaplama algoritmasının, Lipchitz değeri bilinen bir fonksiyonun kök ya da yerel ekstremumunun aramasında, kontrol kuramı bakış açısıyla, bir kararlılık kriteri verilmiştir. Kontrol kuramı yaklaşımında, fonksiyon, kapalı bir kutu gibi değerlendirilmekte ve fonksiyon çıkış işaretinin davranışına göre yeni giriş işaretleri üretilmektedir. Bu sebeple fonksiyonun matematiksel yapısındaki karmaşıklık, çözümü ve algoritmanın işletimini etkilemez.

Uygulamalarda çok değişkenli yerel eniyilemede de önerilen yöntemin başarılı olduğu görülmüştür. Tek değişkenli global aramada elde edilen sonuçlar da en az eşdeğer yöntemler kadar iyidir. Global arama için önerilmiş ve az sayıda değerlendirme ile sonuca erişen yöntemler raslantısallık içermektedir. Önerilen bulanık kontrolör ile arama yönteminde AÖF (amaç ölçüt fonksiyonu) için bir Lipchitz değeri (L) verilebiliyorsa doğru sonuca erişim garanti edilir. Ancak bu yaklaşım işlem sayısının çok artmasına neden olmaktadır. Bulanık kontrolörün karar mekanizması fonksiyonun referansa göre değişim hızını da değerlendirdiğinden adım uzunluğunun $\pm 2L$ değerine kadar başarılı olur. Raslantısal arama yapan yöntemler sonlandırma kriteri olarak önceden belirlenen bir çevrim sayısını kabul eder. Bulanık kontrolörle aramada ise, sonuca erişimde hata işareti değerlendirilmektedir.

Çok değişkenli global eniyilemede önerilen yöntemin kullanılması, algoritmayı karmaşıktırmaktadır. İki değişkenli arama algoritmasının bile diğer yöntemlere göre oldukça karmaşık olduğu şekil 5.7’de görülmektedir. Her değişken için, arama yönünde diğer değişkenler için yapılan aramada elde edilen değerler kullanıldığından, sonuca erişmedeki işlem sayısı değişken sayısının kuvveti ile orantılı artmaz. Bu nedenle algoritmanın yazılım uzunluğu kabul edilebiliyor ise bulanık kontrolör de raslantısal aramaya yakın değerlendirme sayıları ile sonuca erişir.

Önerilen yöntem paralel bilgi işleme uygun bir yapıdadır. Aynı şebeke üzerinde çalışan bilgisayarlara çözüm yükü paylaştırılabilir. Çözümüne erişme hızındaki artış, kullanılan bilgisayar sayısındaki artıştan daha büyüktür.

Tüm adımlayarak arama yapan algoritmaları kapalı çevrimli bir kontrol düzeni olarak modellenenebilir. Fiziksel sistemlerin sayısal kontrolünde, türev bilgisi olarak bir önceki örnekleme sırasında elde edilen değer ile mevcut değer arasındaki fark kullanılmaktadır. Algoritmalar ise türev değerini ‘h’ kadar öteleme yaklaşıklığı ile hesaplar. Sayısal arama algoritmaları h kadar öteleme yerine geriye fark ifadesini kullandığında sonuca erişmedeki performansları düşer. Diğer yandan böyle bir model ile denklem çözdürmek yerine sistem kontrol etmenin de mümkün olduğu düşünülürse bu tezdeki yaklaşım ters yönde de kullanılabilecektir. Örneğin Newton-Raphson kök arama yöntemi, bir kontrol sisteminde lineer olmayan bir kontrolör olarak kullanılabilir.

Yukarıda belirtilen özellikleri ile bu tez eniyileme problemine yeni bir bakış açısı ve çözüm yöntemi getirmektedir. Ayrıca kontrol kuramı ile sayısal arama yöntemleri arasında bir ilişki kurulması yönünden benzerlerinden farklılık göstermektedir.

KAYNAKLAR

- [1] **Zhigljavsky,A.A.**, 1991, “Theory of Global Random Search”, Kluwer Academic Publishers, Dordrecht.
- [2] **Levy,A.V., Montalvo A.** 1985, “The tunneling algorithm for the global minimization of the functions”, *SIAM Journal of Sci.Stat.Comp.*, 6, 15-29.
- [3] **Branin,F., Hoo**, 1972, “A method for finding multiple extrema of a function of n variables”, *Numerical methods of nonlinear optimization*, Academic Press, London.
- [4] **Törn,A. and Zilinskas,A.**, 1988, “Global Optimization – Lecture Notes in Computer Science.”, Spriner-Verlag , Berlin.
- [5] **Pijavskij, S.A.**, 1972, “An algorithm for finding the absolute extremum of a function”, *USSR Comput. Math. and Phys.* ,2, 57-67.
- [6] **Evtushenko,Y.G.**, 1971, “Numerical methods for finding global extrema (case of a non-uniform mesh)”, *USSR Comp.Math. and Math.Phys.* ,11, 1390-1403.
- [7] **Evtushenko,Y.G.**, 1985, “Numerical Optimization Techniques”, Optimization Software Inc, Newyork.
- [8] **Brent, R.P.**, 1973, “Algorithms for minimization without derivatives”, Prentice-Hall, New Jersey.
- [9] **Bremermann,H.A.**, 1970, “A method of unconstrained global optimization”, *Mathematical Biosciences* ,9, 1-15.
- [10] **Leclerc,A.P.**, 1992, “Efficient and reliable global optimization”, *Doktora Tezi*, Ohio State University, Ohio.
- [11] **Shen,Z. and Zhu Y.**, 1987, “An interval version of Shubert’s method for localization of the global maximum”, *Computing* ,38, 275-280.

- [12] **Hansen,E.**, 1979, "Global optimization using interval analysis: the one dimensional case", *Journal of Optimization Theory and Applications*, **29**, 331-344.
- [13] **Kushner,M.J.**, 1964, "A new method for locating the maximum point of an arbitrary multipeak curve in the presence of noise", *Journal of Basic Engineering*, **86**, 97-106.
- [14] **Metropolis,N.,Rosenbluth ,A.W.,Rosenbluth,M.,Teller,A.H.,Teller,M.** 1953, "Equaiton of state calculation by fast computing machines", *Journal of Chem.Phys.*, **21**, 1087-1092.
- [15] **Kirkpatrick,S.,Gelatt,C.D.,Vecchi,M.P.** 1983, "Optimization by simulated annealing", *Science*, **220**, 671-680.
- [16] **Price,W.L.**, 1983, "Global optimization by controlled random search", *Journal of Optimization Theory and Applications*, **40**, 333-348.
- [17] **Hingham,N.J.**, 1996, "Accuracy and Stability of Numerical Algorithms", *Society for Industrial and Applied Mathematics*, Philedeliphia .
- [18] **Gustaffsson, K.**, 1988, "Step size control in ordinary differential equations", *Lund Institute of Technology* , Sweden.
- [19] **Sarioğlu,M.K.**, 1992, "Dijital kontrol sistemleri", Sistem Yayıncılık, İstanbul.
- [20] **Shoup,T.E.,Mistree,F.**, 1991, "Optimization methods with applications for personal computers", Prentice Hall, Boca Raton, Florida.
- [21] **Nakamura,S.**, 1996, "Numerical analysis and graphic visualization with Matlab", Prentice-Hall Inc., New Jersey.
- [22] **Vidyasagar,M.**, 1978, "Nonlinear Systems Analysis", Prentice-Hall Inc., New Jersey.
- [23] **Bir,A.**, 1995, "Lineer Olmayan Kontrol Sistemleri", İ.T.Ü.Elektrik-Elektronik Fakültesi, Yüksek lisans ders notları
- [24] **Sousa,C.D., Bose B.K.**, 1994, "A fuzzy set theory based control of a Phase-controlled converter DC machine drive", *IEEE Transactions on Industry Applications*, **30**, 34-44.

- [25] **Pedrycz,W.**, 1989, “Fuzzy control and fuzzy systems”, Research Studies Ltd.,England.
- [26] **Qin,S.J., Borders,G.**, 1994, “A multiregion fuzzy logic controller for nonlinear process control”, *IEEE Transactions on Fuzzy Systems*, **2**, 74-81.
- [27] **Üstündağ,B.,Bir,A.,Eksin,İ.**, 1998, “Bulanık kontrolör ile yeni bir global optimizasyon yöntemi”, *TOK98 Otomatik Kontrol Ulusal Toplantısı*, İ.T.Ü., İstanbul, 15-16 Ekim, s.109-114.
- [28] **Goertzel,B.**, 1992, “Global optimization by multilevel search”, *Journal of Optimization Theory and Applications*, **75**, 423-432.
- [29] **Luenberger,D.G.**, 1984, “Linear and nonlinear programming”, Addison-Wesley Publishing Company., California.
- [30] **Baba,N.**, 1983, “A hybrid algorithm for finding a global”, *Int J. Control*, **37**, 929-942.
- [31] **Solomatine,D.P.**, 1995, “Application of global optimization to models calibration”, *Netherlands Interdisciplinary Demographic Institute*, The Hauge, July 28.

EK A

Tablo 4.2’deki Bulanık Karar Tablosunun BASIC Dili Karşılığı

If E=NB and CE=NB Then DU=NB

If E=NB and CE=NO Then DU=NB

If E=NB and CE=NK Then DU=NB

If E=NB and CE=S Then DU=NB

If E=NB and CE=PK Then DU=NO

If E=NB and CE=PO Then DU=NK

If E=NB and CE=PB Then DU=S

If E=NO and CE=NB Then DU=NB

If E=NO and CE=NO Then DU=NB

If E=NO and CE=NK Then DU=NB

If E=NO and CE=S Then DU=NO

If E=NO and CE=PK Then DU=NK

If E=NO and CE=PO Then DU=S

If E=NO and CE=PB Then DU=PK

If E=NK and CE=NB Then DU=NB

If E=NK and CE=NO Then DU=NB

If E=NK and CE=NK Then DU=NO

If E=NK and CE=S Then DU=NK

If E=NK and CE=PK Then DU=S

If E=NK and CE=PO Then DU=PK

If E=NK and CE=PB Then DU=PO

If E=S and CE=NB Then DU=NB

If E=S and CE=NO Then DU=NO

If E=S and CE=NK Then DU=NK
If E=S and CE=S Then DU=S
If E=S and CE=PK Then DU=PK
If E=S and CE=PO Then DU=PO
If E=S and CE=PB Then DU=PB

If E=PK and CE=NB Then DU=NO
If E=PK and CE=NO Then DU=NK
If E=PK and CE=NK Then DU=S
If E=PK and CE=S Then DU=PK
If E=PK and CE=PK Then DU=PO
If E=PK and CE=PO Then DU=PB
If E=PK and CE=PB Then DU=PB

If E=PO and CE=NB Then DU=NK
If E=PO and CE=NO Then DU=S
If E=PO and CE=NK Then DU=PO
If E=PO and CE=S Then DU=PK
If E=PO and CE=PK Then DU=PB
If E=PO and CE=PO Then DU=PB
If E=PO and CE=PB Then DU=PB

If E=PB and CE=NB Then DU=S
If E=PB and CE=NO Then DU=PK
If E=PB and CE=NK Then DU=PO
If E=PB and CE=S Then DU=PB
If E=PB and CE=PK Then DU=PB
If E=PB and CE=PO Then DU=PB
If E=PB and CE=PB Then DU=PB

EK B

B.1 $f(x)-r=0$ denkleminin PI kontrolörle çözümüne ilişkin BASIC programı

KOK.BAS (Kaynak kodu ekteki diskette)

B.2 $f'(x_1, x_2)=0$ Denkleminin PI Kontrolörle Çözümüne İlişkin (Yerel Enbüyütme) BASIC programı

YENBUY.BAS (Kaynak kodu ekteki diskette)

B.3 Tek Değişkenli Bir Amaç Ölçüt Fonksiyonunun Bulanık Kontrolör ile Global Eniyilenmesine İlişkin BASIC programı

GEBNUY.BAS (Kaynak kodu ekteki diskette)

EK C

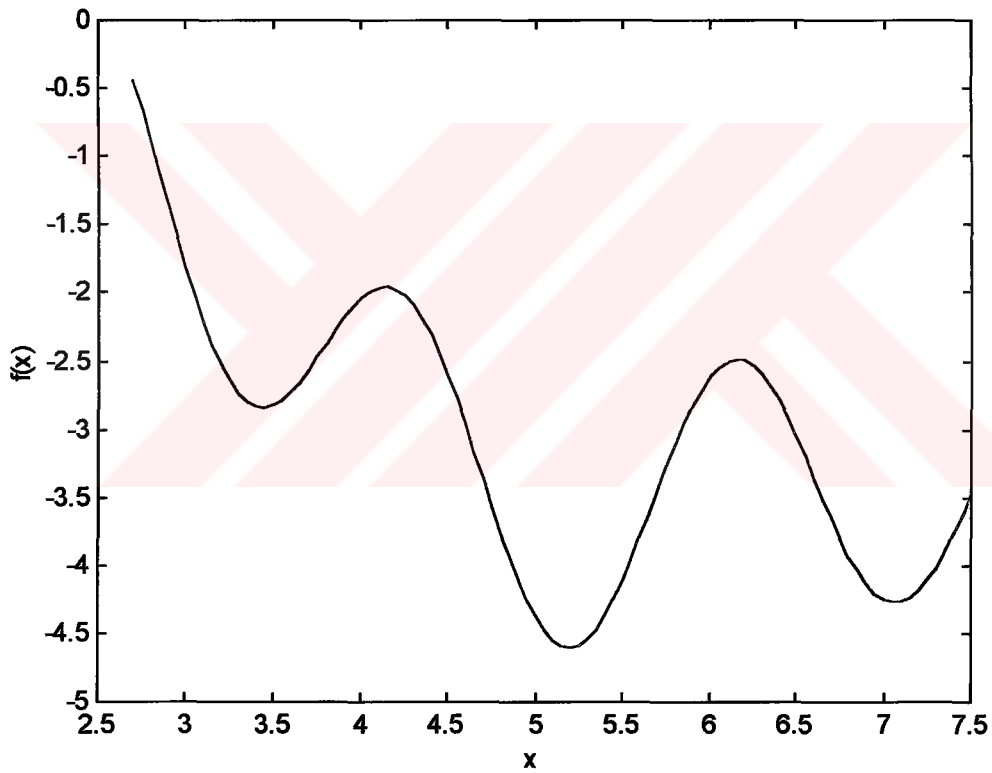
C.1 Tek Değişkenli Test Fonksiyonları :

a) Zilinskas1 [4]

$$f(x) = \sin x + \sin(10x/3) + \ln x - 0.84x, \quad 2.7 \leq x \leq 7.5$$

Global minimum noktası $x^* = 5.1997784$

$$f(x^*) = -4.6013075$$



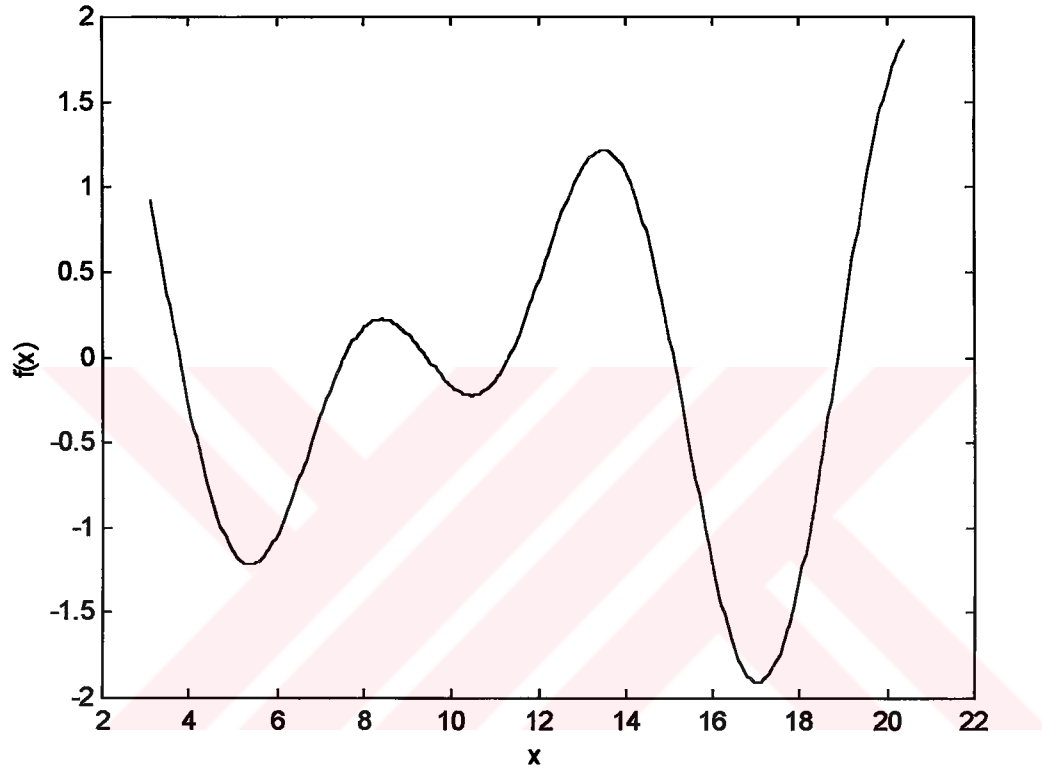
Şekil C.1 $2.7 \leq x \leq 7.5$ aralığında $f(x) = \sin x + \sin(10x/3) + \ln x - 0.84x$ fonksiyonu.

b) Zilinskas2 [4]

$$f(x) = \sin x + \sin(2x/3), \quad 3.1 \leq x \leq 20.4$$

Global minimum noktası $x^* = 17.0391986$

$$f(x^*) = -1.9059611$$



Şekil C.2 $3.1 \leq x \leq 20.4$ aralığında $f(x) = \sin x + \sin(2x/3)$ fonksiyonu.

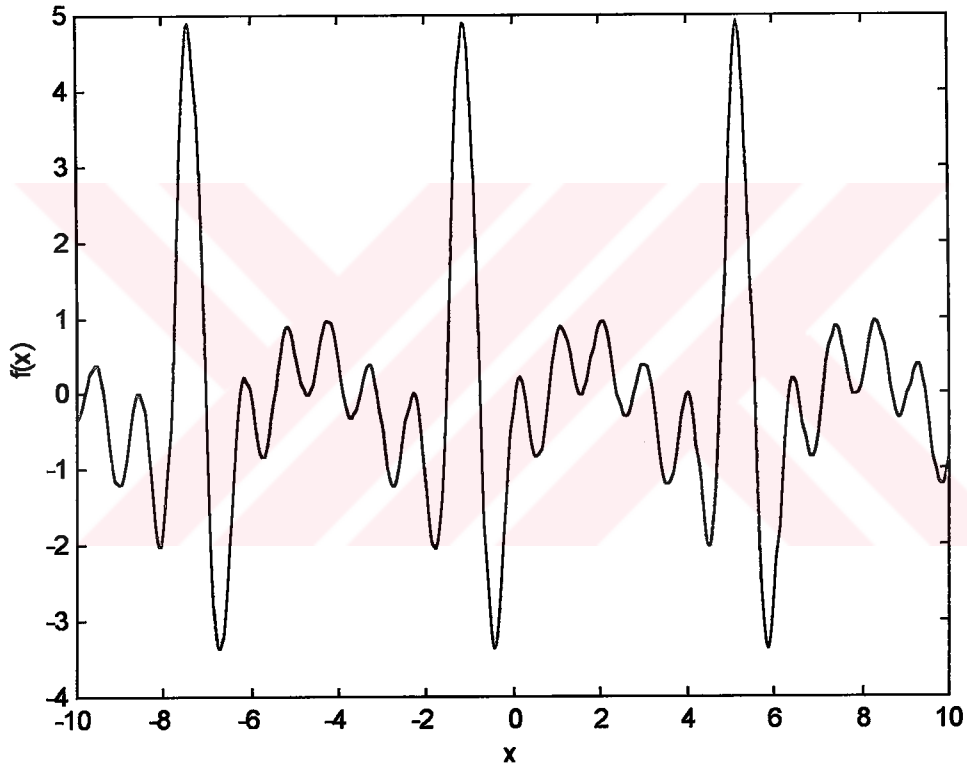
c) Zilinskas 3 [4]

$$f(x) = -\sum_{i=1}^5 \sin((i+1)x + i) \quad , -10 \leq x \leq 10,$$

Global minimum noktaları $x_1^* = -6.7745760$ $f(x^*) = -3.26469$

$x_2^* = -0.4913908$ $f(x^*) = -3.26469$

$x_3^* = 5.7917947$ $f(x^*) = -3.26469$



Şekil C.3 $-10 \leq x \leq 10$ aralığında $f(x) = -\sum_{i=1}^5 \sin((i+1)x + i)$ fonksiyonu

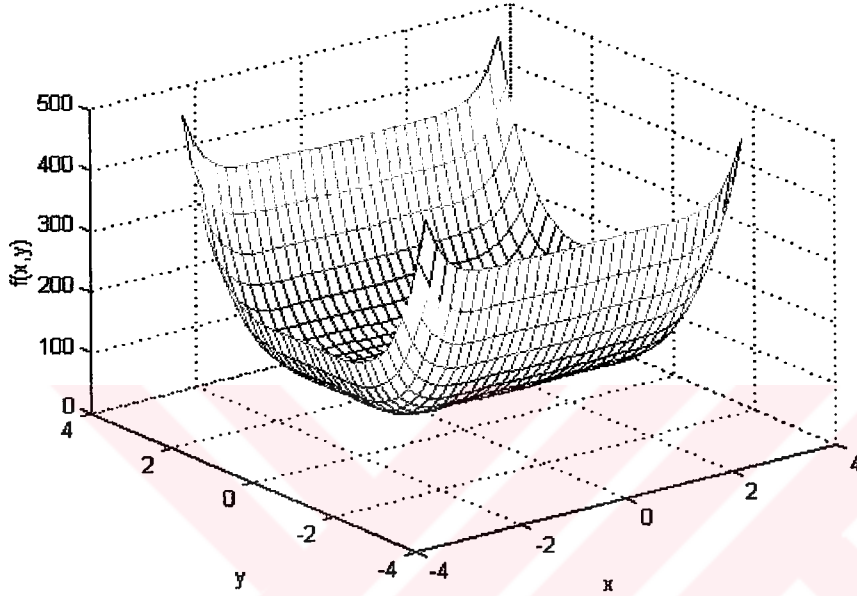
A.2 İki Değişkenli Test Fonksiyonları :

a) Altı kamburlu deve sırtı fonksiyonu

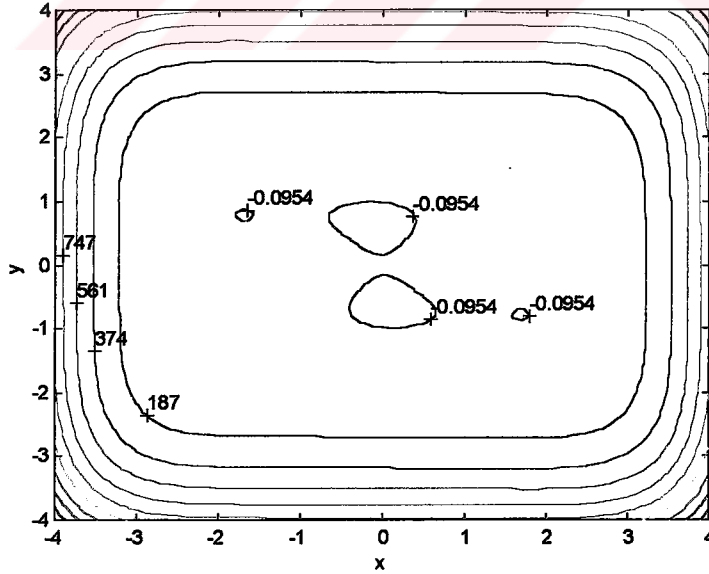
$$f(x,y)=4x^2-2.1x^4+x^6/3+xy-4y^2+4y^4 \quad -5 \leq x \leq 5, -5 \leq y \leq 5$$

Global minimum noktaları $(x^*,y^*)=(0.0898,-0.7126), (-0.0898,0.7126)$

$$f(x^*,y^*)=-1.0316$$



(a)



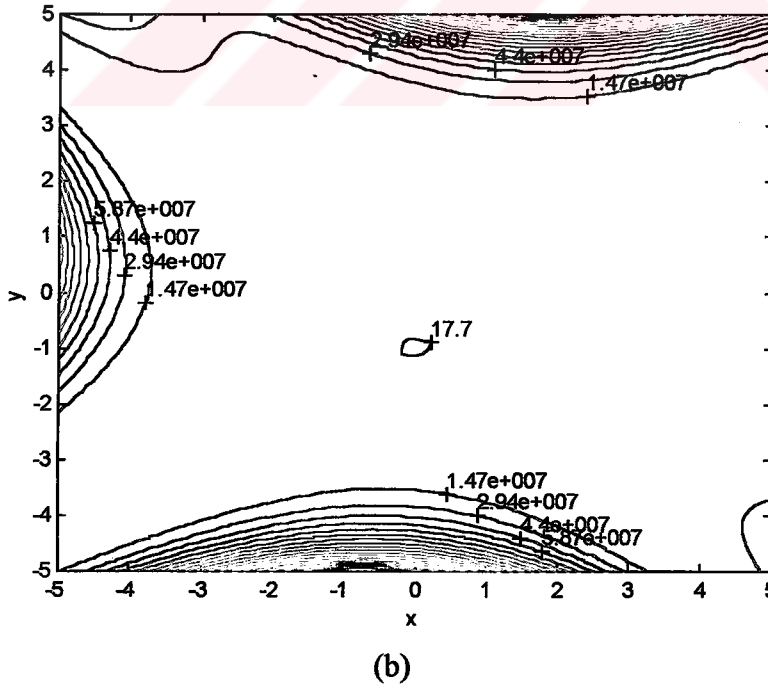
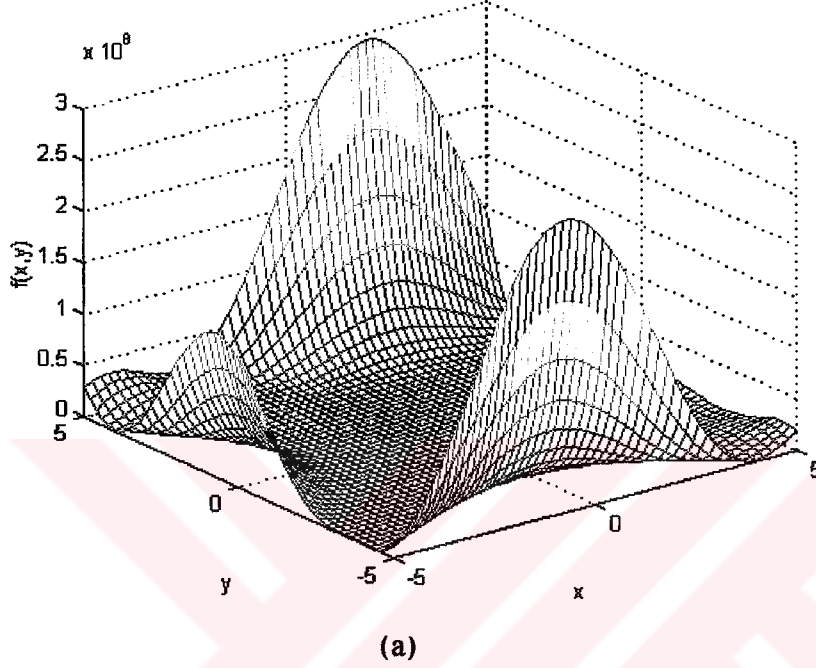
(b)

Şekil C.4 a) $-5 \leq x \leq 5, -5 \leq y \leq 5$ aralığında $f(x,y)=4x^2-2.1x^4+x^6/3+xy-4y^2+4y^4$ fonksiyonun değişimi b) Eşyüksekti eğrisi.

b) Goldstein-Price test fonksiyonu

$$f(x,y)=[1+(x+y+1)^2(19-14x+3x^2+14y+6xy+3y^2)] \times [30+(2x-3y)^2(18-32x+12x^2+48y-36xy+27y^2)] \quad -2 \leq x \leq 2, -2 \leq y \leq 2$$

Global minimum noktası $(x^*, y^*) = (0, -1)$ $f(x^*, y^*) = 3$



Şekil C.5 a) $-5 \leq x \leq 5$, $-5 \leq y \leq 5$ aralığında Goldstein-Price test fonksiyonu

b) Eşyüksekti eğrisi.

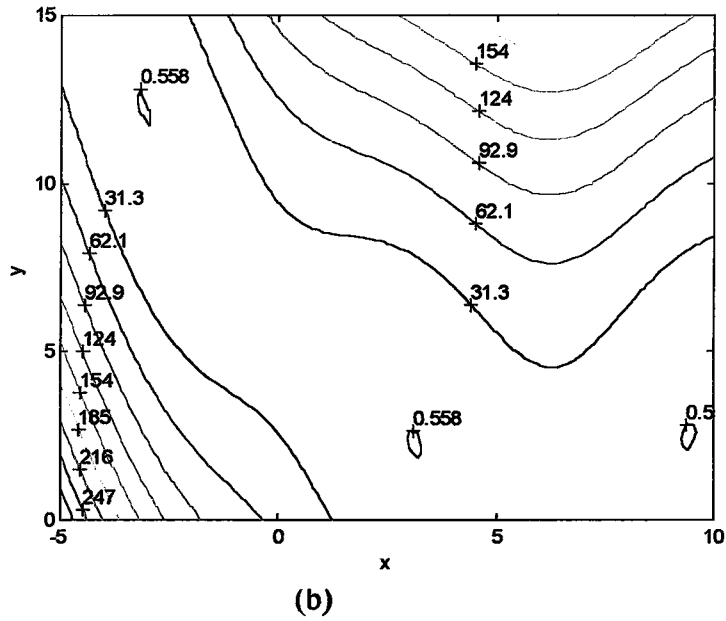
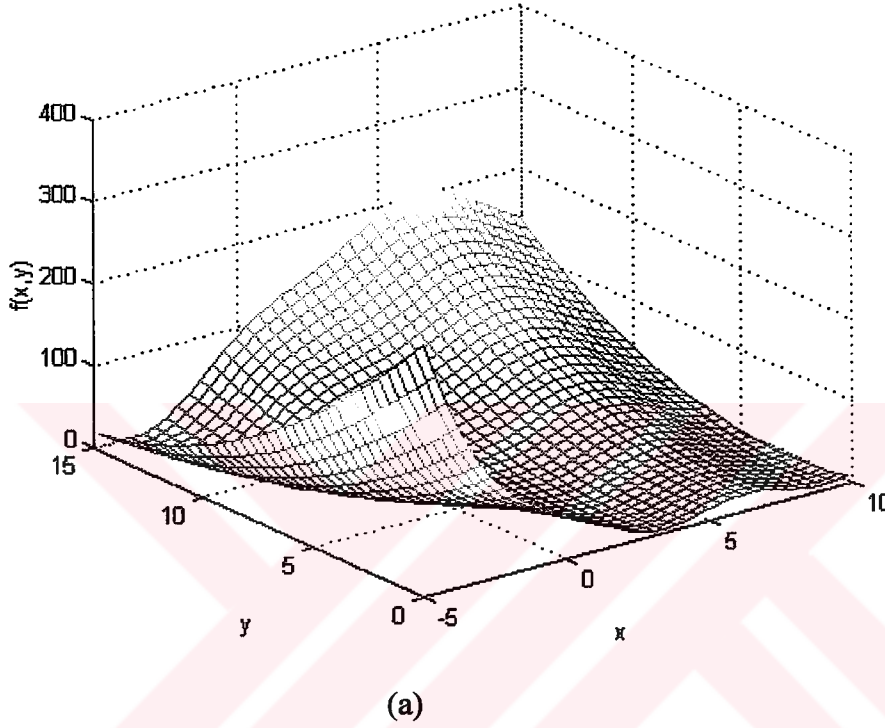
c) Branin test fonksiyonu [4]

$$f(x, y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{5}{\pi}x - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x + 10$$

$$-5 \leq x \leq 10, \quad 0 \leq y \leq 15$$

global minimum noktası $(x^*, y^*) = (-3.142, 12.275), (3.142, 2.275), (9.425, 2.425)$

$$f(x^*, y^*) = 0.398$$



Şekil C.6 a) $-5 \leq x \leq 10, 0 \leq y \leq 15$ aralığında iki değişkenli Branin test fonksiyonu

b) Eşyüksekti eğrisi.

ÖZGEÇMİŞ

Berk ÜSTÜNDAĞ 1970 yılında İstanbul'da doğdu. 1987 yılında Haydarpaşa Teknik Lisesi Elektronik Bölümü'nü bitirdi. 1991 yılında Elektrik-Elektronik Fakültesi Elektrik Mühendisliği Bölümü'nden mezun oldu. Lisans eğitimi bitirme projesi, sayısal kontrol düzeni ile asenkron motorlarda yumuşak yolvermedir. 1991-1994 yılları arasında İ.T.Ü. Fen Bilimleri Enstitüsüne bağlı Kontrol ve Bilgisayar Mühendisliği Bölümünde yüksek lisans eğitimini tamamladı. Yüksek lisans tezi yük hücresi tasarımı ve sayısal ağırlık ölçümü üzerinedir. Bu dönemde bir KOSGEB projesi dahilinde şu an ticari olarak üretimi yapılan 16 değişik model sayısal ölçüm cihazı geliştirmiştir. 1994 yılında İ.T.Ü. Fen Bilimleri Enstitüsü Kontrol ve Bilgisayar Mühendisliği Bölümünde doktora eğitimine başlamıştır. İki adet patent, bir adet faydalı model başvurusu bulunmaktadır.