

MAYOELEKTRİK PROTEZ ELİN YAPAY
ZEKA METOTLARI KULLANILARAK
GERÇEK ZAMANLI OLARAK
DENETLENMESİ

GÖKHAN KAYHAN

DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ ANABİLİM DALI

T.C

ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAYOELEKTRİK PROTEZ ELİN YAPAY ZEKA METOTLARI KULLANILARAK
GERÇEK ZAMANLI OLARAK DENETLENMESİ

GÖKHAN KAYHAN

DOKTORA TEZİ

ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

AKADEMİK DANIŞMAN
Yrd. Doç. Dr. İlyas EMİNOĞLU

SAMSUN - 2011

T.C
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bu çalışma jürimiz tarafından 06/10/2011 tarihinde yapılan sınav ile Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda DOKTORA tezi olarak kabul edilmiştir.

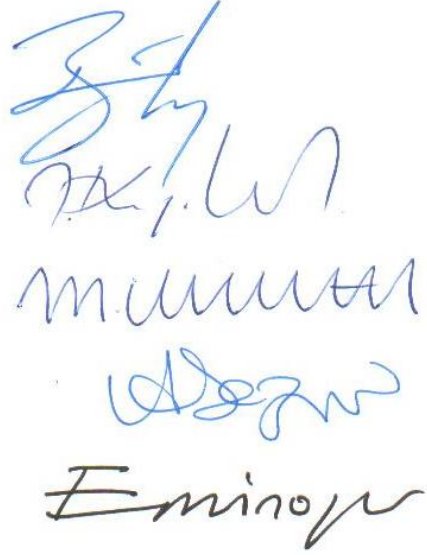
Başkan : Prof. Dr. Bekir KARLIK

Üye : Prof. Dr. Temel KAYIKÇIOĞLU

Üye : Yrd. Doç. Dr. Mustafa ULUTAŞ

Üye: Yrd. Doç. Dr. Abdullah SEZGİN

Üye: Yrd. Doç. Dr. İlyas EMİNOĞLU



ONAY :

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

....../....../20....

Prof. Dr. Hasan GÜMÜŞ
Fen Bilimleri Enstitüsü Müdürü

MAYOELEKTRİK PROTEZ ELİN YAPAY ZEKA METOTLARI KULLANILARAK GERÇEK ZAMANLI OLARAK DENETLENMESİ

ÖZ

Bir kasın kasılması sonucu oluşan mayoelektrik işaretler (EMG), kasılmanın nedeni olan hareketi niteleyen ayırt edici özellikler taşır. Sağlam kas gruplarından alınan EMG işaretleri uygun yöntemlerle işlenerek sınıflandırılabilir, bu kas gruplarına ait hareketler gerçek veya sanal bir yapay eli denetleyecek bir denetim işareti üretilebilir. Literatürdeki çalışmalarda bir elektromekanik eli denetleyecek beş aşamalı bir yapı kullanılmaktadır. Bu aşamalar sırasıyla; işaretin alınması , özellik vektörlerinin çıkarılması, boyut azaltılması, işaretin sınıflandırılması, denetim işaretinin üretilmesi ve elin sürülmesi şeklindedir.

Bu tezin amacı, bir elektromekanik elin gerçek zamanlı olarak yapay zeka metotları kullanılarak denetlenmesidir. Bunun için Doğrusal Ağırlıklandırılmış Normalize Radyal Taban İşlevi (DANRTİ) ağırları, Çok Katmanlı Yapay Sinir Ağları (ÇKYSA) ve Destek Vektör Makinaları (DVM), ile sınıflandırma işlemi yapılmış ve üretilen denetleme işaretleri ile elektromekanik el sürülmüştür. Çalışmanın ilk aşamasında, Karşı Yayılım Ağı (KYA) , Bulanık C Ortalamaları (BCO), Gustafson-Kessel (GK), kümeleme, Gradyan Azalması (GA), GA-Levenberg-Marquardt (LM) öğrenme algoritmalarından oluşan melez öğrenme yapıları tasarlanmıştır. İkinci aşamasında ise, iki ve dört kanallı EMG işaretleri ile deneysel bir elektromekanik elin gerçek zamanlı olarak denetimi gerçekleştirilmiştir.

Melez öğrenme yapılarından KYA, BCO ve GK kümeleme algoritmalarının başarımları GA algoritması ile sınanmıştır. GA algoritmasının başarımının düşük olması nedeniyle GA-LM ‘ ye geçilmiştir. Deneysel bir elektromekanik elin gerçek zamanlı olarak iki kanallı olarak denetimi için ilk olarak kişinin sağlam kaslarından alınan EMG işaretinin zarfı çıkarılmış ve eşik değerler ile karşılaştırılarak dört farklı el hareketi için PIC devresi ile üretilen Darbe Genişlik Bindirimi (DGB) işaretleri ile denetim gerçekleştirilmiştir. Daha sonra iki kanallı EMG verilerine , Dalgacık Dönüşümü uygulanarak özellik vektörleri çıkarılmış, Doğrusal Diskriminant Analiz (DDA) ile boyut azaltılarak tasarlanan melez ağ yapısı üzerindeki başarımları test

edilmiştir. İki kanallı EMG verileri elin dört farklı hareketini belirlemektedir. Yedi farklı hareketi yapabilmek için dört kanallı EMG işaretleri kullanılmıştır. Alınan EMG işaretlerinin dalgacık dönüşümü ile özellikleri vektörleri bulunmuş ve TBA ile boyut azaltılarak DVM , ÇKYSA ve DANRTİ sınıflandırıcı ile sınanmıştır.

Eğitim sonucunda üretilen denetleme işareti paralel port aracılığı ile bir PIC devresine verilmiştir. PIC devresi tarafından üretilen DGB işaretleri ile deneysel bir elektro-mekanik elin gerçek zamanlı olarak denetlenmesi sağlanmıştır. Sonuçlardan DVM ‘nin %94.30, ÇKYSA’ nın %92.90 ve DANRTİ ‘nin ise %92,68 başarımla sağladığı görülmüştür.

Anahtar Sözcükler: EMG, protez el, RTİ, kümeleme, gerçek zamanlı denetim, DVM, yapay sinir ağı.

REAL - TIME CONTROL OF MYOELECTRIC PROTHESIS HAND USING COMPUTATIONAL INTELLIGENCE METHODS

ABSTRACT

Myoelectric signals (EMG) which occur as a result of muscular contraction have differential features characterizing the movement that is the cause of the contraction. EMG signals acquired from the healthy muscle groups can be processed and classified through appropriate methods. A control signal monitoring the movements belonging to these muscle groups; or monitoring a real or virtual artificial hand can be produced. Studies in the literature use a five-stage structure to control an electromechanical hand. Respectively these stages are as follows (i) acquiring the signals, (ii) identifying feature vectors, (iii) reducing the dimension (iv) classification of the signal, (v) producing control signals and driving artificial hand.

This dissertation aims to control an electromechanical hand real-timely using artificial intelligence methods. For the purpose of the study, classification process has been completed using Support Vector Machines (SVM), Multilayer Perceptron (MLP) Networks and Linear Weighted Normalized Radial Basis Function (LWNRBF) Networks, and with the produced control signals, electromechanical hand has been controlled. In the first stage of the study, hybrid learning structures consisting of Counter Propagation Network (CPN), Fuzzy C Means (FCM) and Gustafson-Kessel (GK), grouping Gradient Descent (GD), GD-LM learning algorithms have been designed. In the second stage, using two or four channeled EMG signals, real-time control of an experimental electromechanical hand has been conducted.

The success of the hybrid learning structures, which are CPN, FCM and GK grouping algorithms, has been tested by GD algorithm. Due to the low success of GD algorithm, GD-LM has been used. For the real-time two-channeled control of an experimental electromechanical hand, initially EMG signals of a person's healthy muscles have been identified, threshold values have been compared, and for four different hand movement, PMW signals produced with PIC circuit have been used to perform the control process. Next, to the two-channeled EMG values, wavelet transform has been applied and feature vectors have been determined, size has been reduced via Linear Discriminant Analysis (LDA) and its success on hybrid network structure has

been tested. Two-channelled EMG values determine the four different movements of the hand. In order to be able to perform seven different movements four channelled EMG signals have been used. Wavelet transform of the acquired EMG signals and feature vectors have been identified, and size has been reduced with PCA; then tested with SVM, MLP and LWNRBF classifiers.

The control signal produced as the result of the education was transferred to a PIC circuit via a parallel port. An experimental electromechanical hand's real-time control has been enabled by PWM signals produced by PIC circuit. The results have shown that the success rate of SVM was 94.30 %, that of MLP was 92.90 % and LWNRBF's was 92.68 %.

Key Words: EMG, prosthesis hand, RBF, clustering, real-time control, SVM, neural network

TEŞEKKÜR

Bu tezin hazırlanmasındaki katkılarından dolayı, danışmanım Sayın Yrd. Doç. Dr. İlyas EMİNOĞLU başta olmak üzere, deneysel çalışmalarında işaretlerin alımı için vaktini ve emeğini koşulsuzca sunan Sercan KARACA'ya, olumlu manevi katkılarından dolayı arkadaşlarım Çetin KURNAZ'a, Seda ÜSTÜN ERCAN'a, Begüm KORUNUR ENGİZ'e , Canan ORAL'a , çalışmanın donanımsal altyapısının oluşmasında Bilimsel Araştırma Projeleri ile katkıda bulunan Proje Yönetim Ofisi'ne ve her türlü zorlukta desteklerini benden hiçbir zaman esirgemeyen aileme teşekkür ederim.

SİMGE VE KISALTMALAR LİSTESİ

SİMGELER

$SSD_x(a, b)$	Sürekli Dalgacık Dönüşümü
$\Psi_{ab}(t)$	Sürekli zaman Dalgacık Taban İşlevi
a	Dalgacık işlevi için ölçeklendirme
b	Dalgacık işlevi için zamanda öteleme
$\psi_{j,n}(t)$	Ayrık zaman Dalgacık Taban İşlevi
$g(n)$	Yüksek geçiren süzgeç
$h(n)$	Alçak geçiren süzgeç
$\wp$	TBA için veri seti
$\overline{\wp}$	TBA için ortalaması çıkarılmış veri seti
Γ	Özvektör Matrisi
λ	Özdeğer Matrisi
K	Kovaryans Matrisi
t_p	p temel bileşen için varyansın birikimli yüzdesi
x	Giriş verisi
y	Çıkış verisi
P	Giriş veri dizisi
$w(t)$	KYA için ağıın şimdiki ağırlık değeri
$w(t-1)$	KYA için ağıın bir önceki ağırlık değeri
D	Küme merkezlerine olana öklit uzaklığı
δ	KYA için yarıçap parametresi
N_s	Kazanan birim ait aktivasyon
α	KYA için öğrenme oranı
l	KYA için veri dizisinin indisi
j	KYA için oluşan birim sayısı
$\pi_j(t)$	Grossberg ağı için j. katın şimdiki ağırlığı
$\pi_j(t-1)$	Grossberg ağı için j. katın bir önceki ağırlığı
β	Grossberg ağı için öğrenme parametresi

Z_k	BCO için veri seti
V_i	BCO tarafından belirlenen küme merkezleri
c	Küme sayısı
μ	Üyelik derecesi
U	Üyelik matrisi
m	BCO için ağırlıklandırma üsteli
ε	BCO sonlandırma toleransı
J	BCO amaç işlevi
N	Veri sayısı
A	Endükleme matrisi
F	Küme kovaryans matrisi
H	BCO için bölüm doğrulama kriteri
V_{SC}	GK için bölüm doğrulama kriteri
$Sep(c)$	Bulanık kümelerin bulanık ayrımı
S_B	Küme içi bulanık dağılım matrisi
$Comp(c)$	Bulanık c bölümlerinin toplam yoğunluğu
Σi	Bulanık kovaryans matrisi
ϕ_j	RTİ çıkışı
σ	RTİ 'nin genişliği
$d(n)$	$x(n)$ girişine karşılık gelen gerçek çıkış değeri
$y(n)$	$x(n)$ girişine karşılık gelen ağın ürettiği çıkış değeri
E	Enerji işlevi
e	Giriş – çıkış hatası
μ_w	RTİ ağı için ağırlık öğrenme parametresi
μ_c	RTİ ağı için merkez öğrenme parametresi
μ_σ	RTİ ağı için genişlik öğrenme parametresi
ρ_i	Küme hacmi
ξ_{gi}	DVM de gevşek değişkenler
$K(x_i, x)$	Çekirdek işlevi
α_i	Lagrange çarpanları
w_{ji}^h	i.giriş biriminden h. gizli katmanın j. birimine gelen ağırlıklar
θ_j^h	h. katmanın j. birime ait eşik değeri

i_{pj}^h	gizli katman birim çıkışları
net_{pj}^h	h. Katmana ait j. yapay sinir hücresine ait çıkışı
o_{pk}	k.birim için çıkış
η	öğrenme parametresi
$\Delta_p w_{ji}^h$	Ağırlık değişimi miktarı
$H(n)$	Hessian matrisi
γ	Marquardt parametresi
JK	Jakobien matrisi
$\mathcal{G}$	Hataya bağlı ağırlık gradyanı

KISALTMALAR

EMG	Elektromyografik
MLP	Multi Layer Perceptron
PCA	Principial Component Analysis
ÇKA	Çok Katmanlı Algılayıcı
LDA	Linear Discriminant Analysis
TBA	Temel Bileşenler Analizi
AR	Özbağlısımlı
SOM	Self Organizing Map
FCM	Fuzzy C Means
OMD	Ortalama Mutlak Değer
RBF	Radial Basis Function
RTİ	Radyal Taban İşlevi
BCO	Bulanık C Ortalamaları
OLS	Orthogonal Least Square
RMS	Root Mean Square
YSA	Yapay Sinir Ağı
DDA	Doğrusal Diskriminant Analiz
MÜAP	Motor Ünite Aksiyon Potansiyel
CPN	Counter Propagation Network
KYA	Karşı Yayılım Ağları
HCM	Hard C Means
GK	Gustafson- Kessel
EDK	En Dik Kareler
GD	Gradient Descent
Gİ	Gradyan İnişi
LM	Levenberg- Marguardt
PWM	Pulse Width Modulation
DGB	Darbe Genişlik Bindirimi
DANRTİ	Doğrusal Ağırlıklandırılmış Normalize Radyal Taban İşlevi
ANFIS	Adaptive Neuro Fuzzy System

DVM	Destek Vektör Makinaları
SVM	Support Vector Machines
ÇKYSA	Çok Katmanlı Yapay Sinir Ağı
ÖDH	Öz Düzenleyeci Harita
TMR	Targetted Muscle Reinnervation
İEMG	İşlenmiş ElektroMayografik İşaret

İÇİNDEKİLER

1. GİRİŞ	1
2. GENEL BİLGİLER	14
2.1. Giriş	14
2.2. Mayoelektrik işaretler	14
2.3. EMG işareti kayıt yöntemleri.....	16
2.4. Denetim Stratejileri	22
2.4.1. Tek yerleşimli tek düzeyli istemli açma stratejisi	25
2.4.2. Tek yerleşimli iki düzeyli duyarlılık stratejisi.....	25
2.4.3. Tek yerleşimli oransal duyarlılık stratejisi	26
2.4.4. Tek Yerleşimli Sıralı Strateji	27
2.4.5. İki yerleşimli düzey duyarlı strateji	27
2.4.6. Mod anahtarlama stratejisi	29
2.4.7. İki yerleşimli çoklu hareket stratejisi.....	29
2.4.8. Hedeflenen kasın başka sinirlerle sinirlendirilmesi	30
3. MATERYAL VE YÖNTEMLER	33
3.1. DALGACIK DÖNÜŞÜMÜ.....	33
3.2. BOYUT AZALTMA ALGORİTMALARI.....	38
3.2.1. Temel Bileşenler Analizi	38
3.2.2. Doğrusal Diskriminant Analizi.....	40
3.3. KÜMELEME ALGORİTMALARI	44
3.3.1. Rekabetçi Öğrenme : Kohonen Öz-Düzenleyici Sistemler	44
3.3.1.1. Kohonen Öz-Düzenleme Algoritması:	45
3.3.1.2. Küme Kavramı:	47
3.3.1.3. Ağırlık Güncellenmesi Kavramı.....	50
3.3.2. Karşı Yayılım Ağları.....	54
3.3.3. Bulanık C-Ortalamları Algoritması	58
3.3.3.1. BCO Algoritması	59
3.3.3.2. BCO Yönteminin Eksik Yönleri:	60
3.3.4. Gustafson-Kessel Kümeleme Algoritması	64
3.3.4.1. GK Algoritması.....	67

3.3.5.	Bölüm Doğrulama	68
3.3.5.1.	BCO için bölüm doğrulama:	69
3.3.5.2.	GK için Bölüm Doğrulama:	70
3.4.	ÖĞRENME ALGORİTMALARI	71
3.4.1.	Radyal Tabanlı İşlevli Yapay Sinir Ağları	71
3.4.2.	Destek Vektör Makinaları	76
3.4.2.1.	Doğrusal Destek Vektör Makineleri	76
3.4.2.2.	Doğrusal Olmayan Destek Vektör Makinaları	82
3.4.3.	Çok Katmanlı Yapay Sinir Ağı	87
3.4.4.	Melez Yapı Önerileri	93
3.5.	GERÇEK ZAMANLI DENETİM UYGULAMALARI	97
3.5.1.	İki kanallı EMG ile 4 farklı hareketin gerçek zamanlı denetimi	97
3.5.2.	İki kanallı EMG verisi ile tasarlanan sınıflandırıcının sınanması	99
3.5.3.	Dört Kanallı EMG İşareti ile Elektromekanik Elin Gerçek Zamanlı Denetlenmesi	100
3.5.3.1.	Veri Alımı Aşaması	102
3.5.3.2.	Verinin Ön-işleme Aşaması	103
3.5.3.3.	Özelliklerin Sınıflandırılması	103
3.5.3.4.	Deneyisel (veya sanal) elektromekanik elin denetimi	104
4.	BULGULAR ve TARTIŞMA	106
4.1.	BCO ve GK algoritmalarının sınıflandırıcı başarımlarının sınanması:	106
4.2.	İki kanallı EMG işareti ile 4 farklı hareketin gerçek zamanlı denetimi	109
4.3.	İki kanallı EMG verisinden özellik çıkarma ve tasarlanan sınıflandırıcının sınanması	112
4.4.	Dört Kanallı EMG İşareti ile Elektromekanik Elin Gerçek Zamanlı Denetlenmesi	117
4.4.1.	Veri Alımı Aşaması	117
4.4.2.	Verinin Ön-işleme Aşaması	117
4.4.3.	Özelliklerin Sınıflandırılması	118
5.	SONUÇ VE ÖNERİLER	127
6.	KAYNAKLAR	130
7.	EKLER	138

EK-A : KYA Algoritması Matlab Programı	138
EK-B : BCO Algoritması Matlab Programı.....	140
EK-C: GK algoritması Matlab Programı	142
EK-D: RTİ Ağının GA Algoritması İle Eğitim için Matlab Programı	145
EK-E: DANRTİ Ağının LM- GA ile Eğitimi için Matlab Programı	148
EK-F: DDA Matlab Programı (Cai ve ark. , 2007)	153
EK-G : DDA Kullanarak Boyut Azaltma İşlemi için Matlab Programı	160
EK-H : TBA Kullanarak Boyut Azaltma İşlemi için Matlab Programı.....	161
EK-I : DVM Modelinden Elin Gerçek Zamanlı Denetimi için Matlab Programı	162
EK-J: Dört Hareket için PIC16F628 PICBASIC Programı	164
EK-K: Yedi Hareket için PIC16F628 PICBASIC Programı	167
EK-L : Bir Harekete ait EMG Kanal Kaydı için Matlab Programı.....	170
8. ÖZGEÇMİŞ	171

ŞEKİLLER LİSTESİ

Şekil 1.1 Pencereleme işlevinin incelenmesi	3
Şekil 1.2 Konik İşlevli Yapay Sinir Ağı	5
Şekil 1.3 Bulanık Kümelemeli Yapay Sinir Ağı	5
Şekil 1.4 Deneysel düzenek ve el hareketleri (Jun-Uk ve ark. , 2007)	7
Şekil 1.5 (a) TBA ile azaltılmış özelliklere ait iki temel bileşen (b) ÖDH tarafından elde edilen özellik düzlemi (Jun-Uk ve ark. , 2006).....	7
Şekil 1.6 GDA ve DVM den oluşan sınıflayıcı	8
Şekil 1.7 EMG tabanlı denetim için şematik gösterimi (Shenoy ve ark. , 2008)	8
Şekil 1.8 Protez kolun insan bedeni ile tümleştirilmiş gösterimi	10
Şekil 1.9 Ottobock firmasının mayoelektrik protez eli	10
Şekil 1.10 PHAPS protez düzeneği bileşenleri ve kullanımı (Farrell ve Weir, 2007)	11
Şekil 1.11 Kutu testi için kullanılan düzenek (Farrell ve Weir, 2007)	11
Şekil 1.12 Hızlı ve yavaş protezler için kutu testi sonuçları (Farrell ve Weir, 2007)	12
Şekil 1.13 Mayo-darbe denetimi grafiksel gösterimi (Farrell ve Weir, 2007).....	12
Şekil 1.14 Mevcut ticari protezler için doğrusal hızlar	13
Şekil 2.1 Motor Birim yapısı(Konrad, 2005).....	15
Şekil 2.2 Motor ünite aksiyon potansiyeli (MÜAP).....	16
Şekil 2.3. Kastan fark yükselteci ile işaret alınması (Luca, 2002)	17
Şekil 2.4 EMG işaretlerinin elde edilmesinde kullanılan sensör türleri (Konrad, 2005)	17
Şekil 2.5 Elektrotların sabitlenmesi ve engellilerde kullanım örnekleri (Ajiboye ve Weir, 2005).....	18
Şekil 2.6 EMG işareti güç sıklık tayfı (Luca, 2002).....	19
Şekil 2.7 Elektrotların kasta yerleşimi ve EMG genlik sinyalleri(Luca, 2003).....	20
Şekil 2.8. Yağ tabakası kalınlığının yüzey EMG kaydına etkisi (Konrad, 2005).....	21
Şekil 2.9 EMG işaretinin biyolojik kökenini, elektriksel benzetimini ve ölçülen EMG işaretinin gösterimi (Luca, 1979).....	22
Şekil 2.10 (a) EMG işaretinin zarfı çıkarılarak elde edilen denetim işareti (b) Oransal denetim işareti (c) Sayısal denetim işareti	23
Şekil 2.11 Kol için oransal denetim örneği(Muzumdar, 2004)	24
Şekil 2.12 Tek yerleşimli tek düzeyli istemli açılım stratejisi	25

Şekil 2.13 Tek yerleşimli iki düzeyli duyarlılık	26
Şekil 2.14 Tek yerleşimli oransal duyarlılık	26
Şekil 2.15 Tek yerleşimli sıralı strateji	27
Şekil 2.16 İki yerleşimli iki düzeyli ilk gelen hizmet eder denetim stratejisi	28
Şekil 2.17 İki yerleşimli iki düzeyli farksal denetim	28
Şekil 2.18 Kas aktivitelerine göre protez hareket şeması (Artigue ve Thomann, 2009) .	30
Şekil 2.19 Normal anatomi ve engelli anatomisi için TMR yöntemi (Kuiken ve ark. , 2009)	31
Şekil 2.20 Protez kolun TMR yöntemi ile kullanım örneği	31
Şekil 2.21 Ses denetimli protez el	32
Şekil 3.1 Kullanılan sistemin öbek çizgesi	33
Şekil 3.2 Dalgacık taban işlevi örnekleri	34
Şekil 3.3 Dalgacık işaretinin zamanda kaydırılması(Misiti ve ark. , 2002).....	35
Şekil 3.4 Dalgacık işaretinin ölçeklendirilmesi (Misiti ve ark. , 2002)	35
Şekil 3.5 Süzgeçleme ile ayırık dalgacık dönüşümü	36
Şekil 3.6 Süzgeçleme ve örnek azaltma	37
Şekil 3.7 Bir işarete ait 3. Dereceden Dalgacık Ayrışma Ağacı (Misiti ve ark. , 2002) .	37
Şekil 3.8 DDA kullanımında sınıf ayrılabilirliğinin gösterimi	41
Şekil 3.9 Verilerin ortalamasının sınıf ayrılabilirliğindeki başarımı	42
Şekil 3.10. Kohonen Öz - Düzenleyici Ağ Yapısı.....	45
Şekil 3.11 Yarıçapın büyütülmesi durumu	48
Şekil 3.12 Yarıçapın küçültülmesi durumu	48
Şekil 3.13 Kesişim alanında eleman bulundurmayan $r = 0.4$ yarıçaplı daire ile belirtilen iki küme	49
Şekil 3.14 Kesişim alanında eleman bulunduran $r = 0.4$ yarıçaplı daire ile belirtilen iki küme.....	50
Şekil 3.15 On elemanlı ve tüm elemanları aynı bir veri kümesi	51
Şekil 3.16 İki elemanlı küme.....	51
Şekil 3.17 Çoklu veri içeren tek kümeli veri	52
Şekil 3.18 Tek kümeli çoklu veri ağırlık güncellemeli olarak merkezinin bulunması	53
Şekil 3.19 Tek kümeli çoklu veri α öğrenme parametresinin değişimi	53
Şekil 3.20 Karşı Yayılım Ağ yapısı	54
Şekil 3.21 Kural çizelgesi benzetimi.....	55

Şekil 3.22 Örnek 1 için ağ çıkışları	56
Şekil 3.23 Örnek 2 için ağ çıkışları	57
Şekil 3.24 m parametresinin seçimi (Zhang, 2005)	61
Şekil 3.25 Küresel olmayan veri seti için merkez belirleme (Zhang, 2005)	61
Şekil 3.26 Farklı boyut ve yoğunluktaki veri seti için merkez belirleme (Zhang, 2005)	62
Şekil 3.27 Küme sayısına göre merkez belirleme (Zhang, 2005)	62
Şekil 3.28 Kelebek veri seti	63
Şekil 3.29 BCO üyelik işlevleri	63
Şekil 3.30 İç çarpım normu (Babuska, 1998)	65
Şekil 3.31 Farklı biçimlerdeki kümeler için BCO ve GK algoritmaları (Zhang, 2005) ..	66
Şekil 3.32 Farklı yönelimlerdeki veriler için BCO ve GK algoritmaları (Babuska, 1998)	66
Şekil 3.33 BCO ve GK algoritmaları için benzetim örneği	67
Şekil 3.34 Radyal Taban İşlevli Yapay Sinir Ağ Yapısı	72
Şekil 3.35 ÇKYSA ve RTİ karşılaştırılması	73
Şekil 3.36 İki sınıflı veri kümesini ayırabilen hiper-düzlemler	77
Şekil 3.37 İki boyutlu uzaydaki iki sınıflı veri kümesi için en uygun hiper-düzlem	77
Şekil 3.38 Eğitim verilerinin doğrusal olarak ayrılamadığı durum	80
Şekil 3.39 Giriş uzayından özellik uzayına verinin aktarılması	83
Şekil 3.40 DVM Ağ yapısı (Vapnik, 1998)	86
Şekil 3.41 Çok Katmanlı Yapay Sinir Ağı Örneği	88
Şekil 3.42 Aktivasyon işlevleri	88
Şekil 3.43 Çıkış katına ait bir birimin yapısı	90
Şekil 3.44 Melez yapı önerisi	94
Şekil 3.45 DANRTİ Sınıflandırıcının şematik gösterimi	95
Şekil 3.46 Sınıflandırılan dört el hareketi	97
Şekil 3.47 DGB işaretleri ve motor milinin konumu	98
Şekil 3.48 Elektro-mekanik protez el ve iki tane 16F84A mikro denetleyicili sürücü devre	98
Şekil 3.49 Tasarlanan sistemin şematik gösterimi	99
Şekil 3.50 EMG verisi alımında kullanılan deneysel düzenek	101
Şekil 3.51 EMG denetimli protez geliştirme düzeneğinin alt parçaları	101

Şekil 3.52 Önkol kaslarına yüzey elektrotlarının yerleşim şeması	102
Şekil 3.53 Belirlenen hareket seti.....	103
Şekil 3.54 Deneysel elektromekanik elin sürülmesi.....	104
Şekil 3.55 Gerçek zamanlı deneysel EMG denetimli sanal / gerçek protez el geliştirme düzeneği	105
Şekil 4.1 Doğrusal olmayan çok girişli tek çıkışlı sistem	106
Şekil 4.2. Sistem Modelleri	107
Şekil 4.3. Sistem Modelleri	109
Şekil 4.4 Gürültü ve DC bileşenden arındırılmış EMG işareti	110
Şekil 4.5 Doğrultulmuş 2 kanallı EMG işareti	110
Şekil 4.6 Denetim işaretinin üretimi: Her kanal için üretilen mantık seviyeleri ve Çizelge 4.3 kullanılarak denetim işareti (veya sınıf bilgisi) üretilmektedir.	111
Şekil 4.7. İki kanallı ham EMG verisi (1-3-5-7): Serbest durum, (2): Yukarı, (4): Aşağı, (6): Yumruk	113
Şekil 4.8 İki kanallı ham EMG verisinin ayrık dalgacık dönüşümü ile analizi	114
Şekil 4.9 (a) 1. Kanal ham EMG verisi (b) 2. Kanal ham EMG verisi	114
Şekil 4.10 (a) (b) 1. ve 2. Kanal ham EMG verisi (c) (d) 1. ve 2. özellik vektörü (e) Hareket çıkışı	115
Şekil 4.11 ANFIS ağ yapısı ile eğitim	116
Şekil 4.12 Melez ağ yapısı ile eğitim	116
Şekil 4.13 Kas hareketlerini temsil eden ham EMG verisi	117
Şekil 4.14 Dalgacık Dönüşümü ve TBA kullanılarak özellik çıkarımı	118
Şekil 4.15 Epoch=2 için merkez, genişlik ve ağırlık öğrenme parametrelerinin eşit seçilmesi durumunda eğitim ve test verilerinde yüzdelik ağ başarıımı	119
Şekil 4.16 Epoch=4 için merkez , genişlik ve ağırlık öğrenme parametrelerinin eşit seçilmesi durumunda eğitim ve test verilerinde yüzdelik ağ başarıımı	119
Şekil 4.17 Epoch=2 için merkez, genişlik ve ağırlık öğrenme parametrelerinin farklı seçilmesi durumunda eğitim ve test verilerinde yüzdelik ağ başarıımı	120
Şekil 4.18 DANRTİ ağı için hareket biçimine göre yüzdelik hata değişimi	120
Şekil 4.19 DANRTİ yapısı ile dört kanallı EMG sınıflandırması eğitim sonucu	121
Şekil 4.20 DANRTİ yapısı ile dört kanallı EMG sınıflandırması test sonucu	121
Şekil 4.21 ÇKYSA için nöron sayısına bağlı olarak yüzdelik ağ başarıımları	122
Şekil 4.22 Kullanılan ÇKYSA yapısı	122

Şekil 4.23 ÇKYSA için hareket biçimine göre yüzdelik hata değişimi	123
Şekil 4.24 $\gamma = 1$ için ceza faktörünün (C) değişimine bağlı olarak yüzdelik ağ başarımı	123
Şekil 4.25 $\gamma = 2$ için ceza faktörünün (C) değişimine bağlı olarak yüzdelik ağ başarımı	124
Şekil 4.26 $\gamma = 3$ için ceza faktörünün (C) değişimine bağlı olarak yüzdelik ağ başarımı	124
Şekil 4.27 DVM ağı için hareket biçimine göre yüzdelik hata değişimi	125
Şekil 4.28 El açık ve el aşağı hareketlerinin sanal el simülatörüne ve robot ele gerçek zamanlı olarak uygulanması.....	126

ÇİZELGELER LİSTESİ

Çizelge 2.1 İki yerleşimli Boston kolu için protez hız ve işlevi	24
Çizelge 4.1 Yöntemlere ait parametreler ve sonuçlar.....	107
Çizelge 4.2 Yöntemlere ait parametreler ve sonuçlar.....	109
Çizelge 4.3 Doğruluk Çizelgesi	112
Çizelge 4.4 Yöntemlere ait parametreler ve sonuçlar.....	125

1. GİRİŞ

Elektromyografik (EMG) işaret, kas hareketleri (kasılma/gevşeme) sonucu oluşan elektriksel bir büyüklüktür ve ilgili kas grubunun aktivitelerini kısmen tanımlayabilir. EMG işaretinin genliği tepeden tepeye 0-10 mV, işaretin enerjisi 0-500 Hz sıklık bandına yayılmış olup baskın enerjisi 20-150 Hz bandında bulunmaktadır (Delsys Inc., 2002; Jun-Uk ve ark. , 2006; Jun-Uk ve ark. , 2007). Kas aktivitelerini karakterize ettiği için EMG işareti, protez uzuvların denetimi, kas sistemine ait hastalıklara tanı koyma, spor hekimliği, jinekoloji ve ergonomik tasarım gibi pek çok alanda kullanılır.

1940'lı yılların ortalarında mekanik ellerin denetiminde, engelli insanların kesik uzuvlarından elde edilen mayoelektrik sinyallerin kullanılabileceği fikri öne sürüldü fakat 1950 li yıllara kadar herhangi bir klinik çalışma yapılmadı. İlk geçerli klinik protez el Ruslar tarafından üretilen elektrikli el oldu ve sınırlı bir miktarda İngiltere ve Kanada'ya ihraç edildi. Yapılan bu çalışmaların çoğu ticarileşememiş ve yayınlanmamış olmasına rağmen 1960 lı yıllarda bir çok güçlü cihaz (el ve kanca) tasarlandı. Sonraki 40 yıl içinde özellikle Kuzey-Batı Üniversitesi, Temple Üniversitesi Chalmers Üniversitesi, California Üniversitesi, New-Brunswick Üniversitesi ve Massachusetts Institute of Technology gibi Kuzey Amerika'daki üniversitelerin yanı sıra bazı Japon ve İtalyan üniversiteleri önemli başarılarla öncülük etti. (Englehart ve ark. , 2001)

Literatürde EMG sinyallerini anlamlandırılarak elde edilen işlevleri artırmak için istatistiksel ve matematiksel sinyal işleme tekniklerinin EMG işaretlerine uygulanmasına yönelik çalışmalar bulunmaktadır (Kwatny ve ark. , 1970; Hogan ve Mann, 1980; Lindstrom ve Magnusson, 1977). Bu konudaki ilk sayılabilecek çalışmalardan birinde, birkaç serbestlik dereceli kol yapay uzvunun denetiminde mikroişlemci donanımını kullanan bir protez denetim sistemi tanımlanmıştır (Graupe ve ark. , 1978). Bir başka çalışmada, felçli veya engelli bir kişinin biceps ve triceps kaslarından alınan EMG sinyallerinin örüntü tanıma ve istatistiksel çözümlemesi yapılmıştır (Saridis ve Gootee, 1982). Hareketlerin sınıf ayrılabilirliğinin başarılı olduğu gözlenmiş ve omuzdan içe-dışa dönme, kol açma-kapama ve bilek aşağı- yukarı bükme olmak üzere 6 basit hareket gerçekleştirilmiştir.

Bir veya iki kanallı EMG sinyallerinden elde edilen bilgi kapasitesini artırmak amacıyla zaman serisi modellerini kullanan çalışmalar yapılmıştır. Yöntemin sinyal genliğindeki değişimlere duyarlı olması yöntemin başarısını etkileyen faktörlerden olup umut verici sonuçlar elde edilmiştir (Doerschuk ve ark. , 1983).

EMG işaretlerinden denetim bilgisi çıkararak el, kol, bilek gibi protez uzuvları denetleyen EMG denetimli sistemlerle ilgili çeşitli çalışmalar mevcuttur (Saridis ve Gootee, 1982; Hudgins ve ark. , 1993; Kuruganti ve ark. , 1995). Bu çalışmalarda sistemler çok az sayıda denetim bilgisi gerektirdiği için başarılıdır ancak kanal sayısının artması dolayısıyla işlev sayılarının arttığı durumlar sistemlerin denetlenmesini zorlaştırmaktadır.

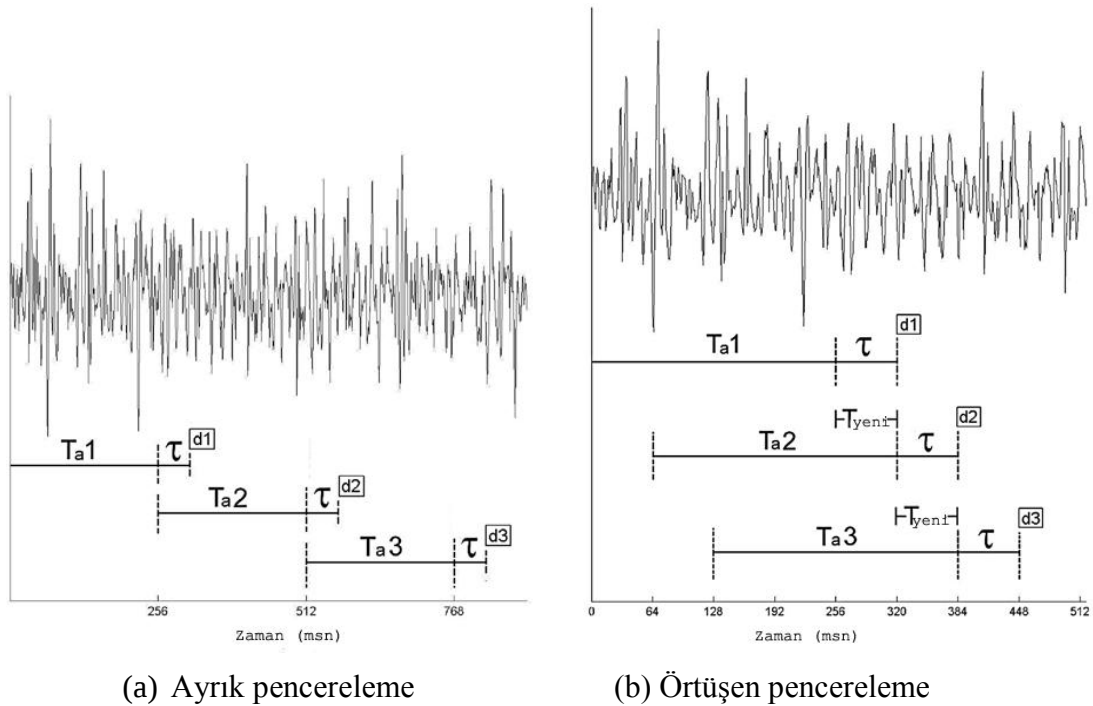
Hudgins mayoelektrik örüntülerin sınıflandırılması esasına dayanan çok işlevli protezlerin denetimi için yeni bir yaklaşım sunmuştur (Hudgins ve ark. , 1993). Mayoelektrik sinyalin, kas kasılmalarının başlangıç fazında deterministik bir yapı sergilediğini tespit etmiştir ve mayoelektrik sinyali çeşitli zaman dilimlerine ayırarak özellikler elde etmiş ve Çok Katmanlı Algılayıcı (Multi Layer Perceptron, MLP) yapay sinir ağ yapısını sınıflayıcı olarak kullanmıştır. Bu yeni yapıyla tek kanallı mayoelektrik işaret tarafından denetlenebilen işlevlerin sayısını artırmış ve %70 ile %98 arasında bir sınıflama doğruluğu elde etmiştir.

Englehart, kısa zamanlı Fourier dönüşümü, dalgacık dönüşümü, dalgacık paket dönüşümünü kapsayan zaman – sıklık yöntemlerini iki kanallı EMG işaretlerinden özellik çıkarmada kullanmıştır (Englehart ve ark. , 1999). Elde ettiği özelliklere Temel Bileşenler Analizi (Principal Component Analysis, PCA) yöntemiyle boyut azaltma işlemi uygulamış ve Çok Katmanlı Algılayıcı (ÇKA) ağ yapısı ve Doğrusal Diskriminat Analiz (Linear Discriminant Analysis, LDA) yapısı ile sınıflandırmıştır. Zaman-sıklık yöntemlerinin özellik çıkarma işleminde güçlü bir başarımlı sağladığını göstermiştir.

Temel olarak, işaret işleme altyapısının kullanıcı için en az zaman gecikmesine neden olacak biçimde tasarlanması gerekmektedir (Jun-Uk ve ark. , 2006; Farrell ve Weir, 2007). Bu nedenle kaynaklarda EMG işaretlerinin sınıflandırılması için yaygın olarak üç aşamalı bir yapı kullanılır. Bu yapı: i) *Öz-nitelik vektörü çıkarımı* ii) *Boyut azaltımı* ve iii) *Sınıflandırma* (Englehart ve Hudgins, 2003; Karlik ve ark. , 2003; Jun-

Uk ve ark. , 2006; Jun-Uk ve ark. , 2007; Oskoei ve Hu, 2007; Farrell ve Weir, 2007; Khushaba ve Al-Jumaily, 2007) aşamalarından oluşmaktadır.

Farrell (2011), çalışmasında, EMG işaretlerinin analizinde pencereleme işlevini incelemiştir ve örtüşen pencereleme kullanmayı önermiştir. Şekil 1.1 (a) da tek kanallı EMG verisinde ayırık pencerelemenin kullanımını gösteren bir örnek verilmiştir. ($T_a=256$ msn $\tau=64$ ms) Karar alma sürecinden ($d1$, $d2$ ve $d3$) önce her analiz penceresi (T_a) bir τ gecikme süresine sahip olmaktadır. Bu nedenle her pencereyi işlemek için gerekli zamanın (τ), analiz penceresi uzunluğundan çok daha az olmasına ve böylece işlemcinin gereksiz işlem zamanı harcamamasına dikkat edilmelidir. Şekil 1.1 (b)' de ise tek kanallı EMG verisinde örtüşen pencereleme kullanımı görülmektedir ($T_a=256$ msn, $\tau=T_{yeni}=64$ ms). Örtüşen pencereler kullanıldığı zaman analiz penceresi nispeten küçük artışlarla kaydırılır ve en eski verilerden atılarak yeni alınan veriler eklenir. Bu verilen örnekte her penceredeki yeni veri miktarı (T_{yeni}), önceki analiz penceresi verilerini işlemek için gerekli süreye (τ) tam olarak eşittir. Önceki penceredeki veriye ait karar tamamladığı zaman hemen bir sonraki sınıfın işlemini başlatmak için örtüşme miktarını işlem süresine eşit olarak ayarlanır. Bu işlem denetleyici tarafından sağlanır. Örtüşen pencereler, büyük gecikmeleri azaltan daha tutarlı denetleyiciler üretir.



Şekil 1.1 Pencereleme işlevinin incelenmesi

Englehart ve ark. (2001) çalışmasında, altı hareket için 4 kanallı yüzey EMG işaretlerinin dalgacık paket katsayılarını çıkarmıştır. Temel Bileşenler Analizi (TBA) yöntemi kullanılarak boyut azaltımı yapılmış ve Bayesian sınıflayıcı ile %2 sınıflama hatasıyla başarımlar sağlanmıştır.

Huang ve ark. (2003) çalışmasında, 3 kanallı yüzey EMG işaretlerinden , Özbağlaşım (Auto Regressive, AR) katsayıları ve histogram gibi zaman düzlemi özellik vektörlerini elde etmiş, Öz Düzenleyici Haritalama (Self Organizing Map, SOM) kullanarak özellikleri 2 boyutlu düzleme yansıtmış ve ÇKA sınıflayıcı kullanarak 7 el hareketinin ayırt edilmesini ortalama %96,25 - 98,75 doğrulukla başarmıştır.

Pan ve ark. (2004) 'nın çalışması ise doğrultulmuş EMG işaretinin genlik değerinin bir eşik değeri ile karşılaştırılması temeline dayanan, elektrik mühendisliği eğitiminde lisans öğrencilerinin eğitimi için tasarlanmış basit bir deneysel ele açma ve kapama hareketlerinin yapıldığı bir çalışmadır.

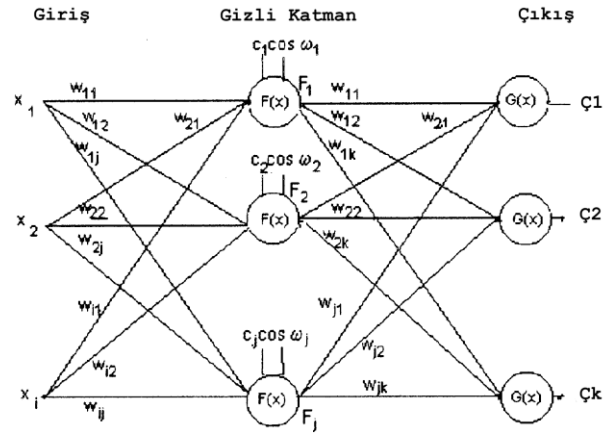
Ajiboye ve Weir (2005) çalışmasında çok işlevsel protezlerin denetimi için EMG örüntüleri tanıma işlemini bir bulanık mantık uygulaması olarak gerçekleştirilmiştir. Temel bulanık sistem, istatistiksel (Ortalama ve standart sapma) esaslara dayanan üyelik işlevleri kullanılarak tanımlanmış ve işaret genliklerine Bulanık C Ortalamaları (Fuzzy C Means, FCM) uygulanarak kural çıkarımı otomatikleştirilmiştir. Tüm sınıflandırmalarda %94 - %99 aralığında başarımlar elde edilmiştir.

Çoğu ticari mayoelektrik el için, EMG işaretinin Ortalama Mutlak Değerini (OMD) belirlenmiş bir eşik değeri ile karşılaştırılarak denetim işareti üretilir (Jun-Uk ve ark. , 2007; Oskoei ve Hu, 2007). Sade olmasına karşılık bu yöntem, elin açılması ve kapanması için gerekli olan denetim komutlarını üretebilir. OMD yöntemi, (Karlık ve ark. , 2003; Englehart ve Hudgins, 2003; Jun-Uk ve ark. , 2006; Jun-Uk ve ark. , 2007; Oskoei ve Hu, 2007; Khushaba ve Al-Jumaily, 2007) sunulan yöntemlerle karşılaştırıldığında, oldukça düşük hesapsal maliyete sahiptir.

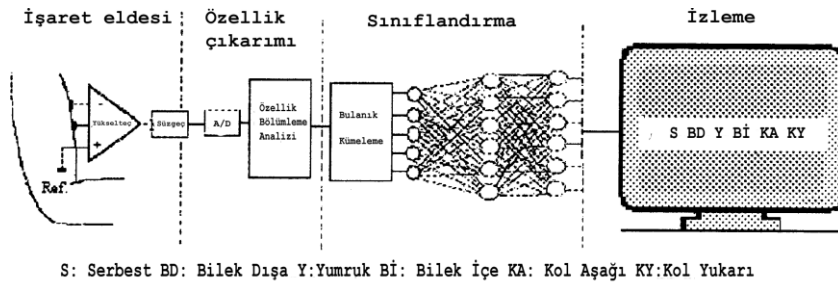
EMG işaretinin gerçek zamanlı sınıflandırılması ile ilgili yakın zamanda yapılan çalışmalarda oldukça iyi sınıflandırma doğruluğu oranları elde edilebilmiştir (Oskoei ve Hu, 2007) .

Karlık ve ark. (2003), yüzey EMG sinyalleri kullanarak çok fonksiyonlu bir protez elin denetiminde yeni bulanık kümelemeli yapay sinir ağ yapılarının sınıflama doğruluklarını karşılaştırmıştır. Radyal Taban İşlev (Radial Basis Function, RBF)

ağları ile ÇKA ağlarını birleştiren bir yapı olan (Şekil 1.2) Konik İşlevli Yapay Sinir Ağı (Conic Section Function Neural Network, CSFNN), Şekil 1.3 ile verilen Bulanık Kümelemeli Yapay Sinir Ağı (Fuzzy Clustering Neural Network, FCNN) ve Çok Katmanlı Algılayıcı Yapay Sinir Ağı (Multi Layer Perceptron Neural Network, MLPNN) olarak seçilen bu ağlar için 2000 iterasyonda sırasıyla %88, %98 ve %97 başarımlar sağlanmıştır. Özbağlaşım (AR) modeli kullanılarak EMG işaretinden elde edilen 4 AR katsayısına Bulanık Kümelemeli Yapay Sinir Ağı'nda önce Bulanık C Ortalamaları (BCO) algoritması uygulanmış Konik Fonksiyonlu Yapay Sinir Ağı 'nda ise En Dik Kareler (Orthogonal Least Square, OLS) yöntemi uygulanmıştır. Bu algoritmalarla elde edilen çıkışlar Yapay Sinir Ağı (YSA) için giriş olarak kullanılmıştır. YSA'nın serbest parametreleri ise geri yayılım ağı kullanılarak hesaplanmıştır.



Şekil 1.2 Konik İşlevli Yapay Sinir Ağı

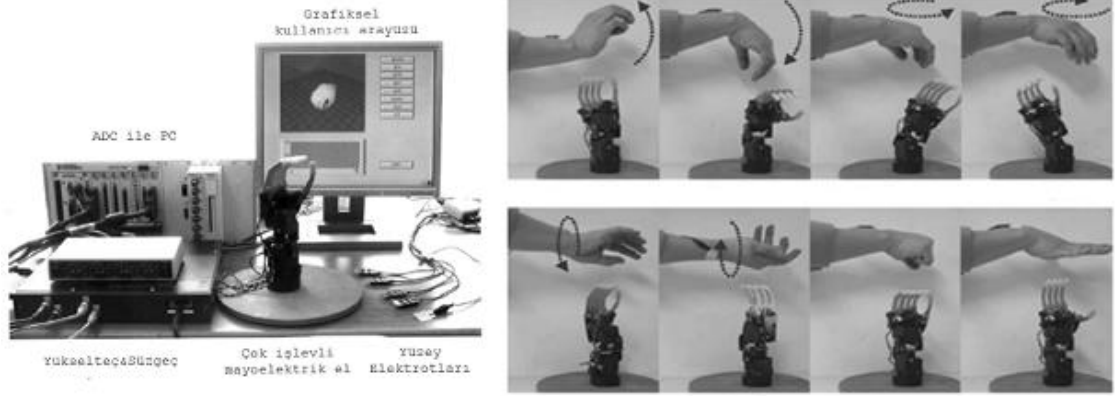


Şekil 1.3 Bulanık Kümelemeli Yapay Sinir Ağı

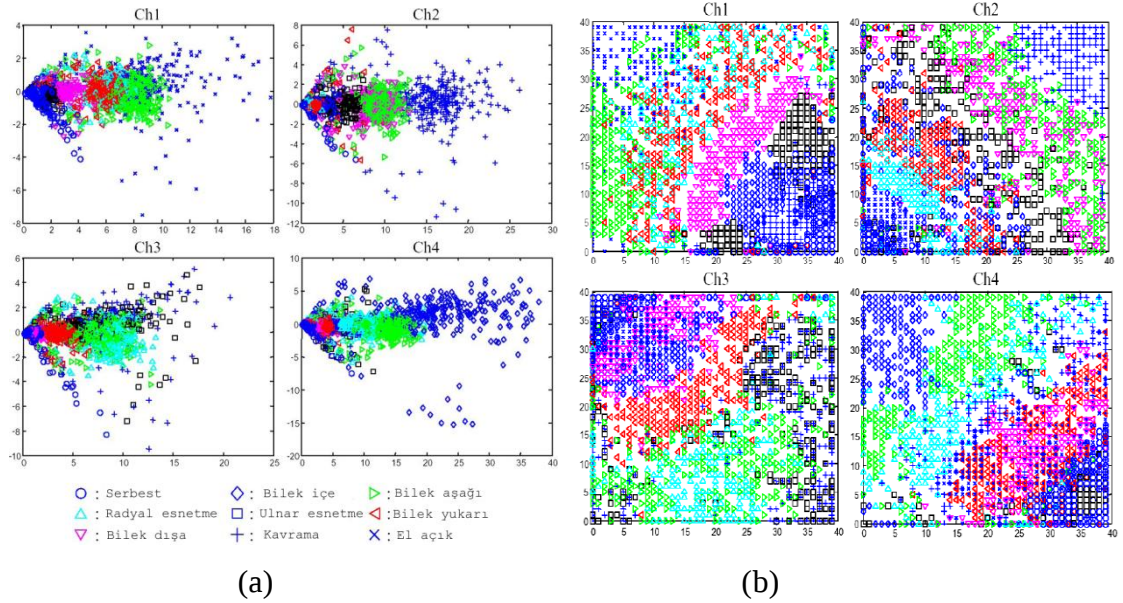
Momen ve ark.(2007)'da niyet edilen hareketle ilişkili olarak önkol kas işaretlerini gerçek zamanlı sınıflandırma yapmak için kullanılmaktadır. Önerilen bu yöntem kullanıcının tercih ettiği kas kasılmalarını kaydetmekte ve başarılı kasılmalar arasında rahatlatma periyodu olmaksızın bu kasılmalar sürekli olarak

sınıflanabilmektedir. Bu çalışmada, EMG sinyalleri bir doğuştan engelli ve 7 sağlam katılımcının ön kol (flexor ve extensor) kaslarından kaydedilmiştir. Katılımcıların serbest bir şekilde seçtiği kas kasılmaları bir eğitim protokolü ile etiketlenmiştir. EMG işareti 0.2 sn lik sabit pencere ile hesaplanmış ve karesel ortalamalarının karekökü (Root Mean Square, RMS) hata değerlerinin doğal logaritmasıyla parametreleştirilmiştir. Özellik vektörü BCO kümeleme yöntemi ile bölümlenerek hareketler sınıflandırılmıştır. Her kullanıcıdan alınan 2 dakika eğitim verisi ile 4 farklı hareketi ortalama % 92.7 + % 3.2 doğrulukla ayırt edilebilmektedir.

Jun-Uk ve ark. (2007) ise Şekil 1.4 de verilen düzenek ile belirlenen hareketler için kaydedilen 4 kanallı EMG işaretinden özellik vektörü çıkarma işlemi için dalgacık dönüşümünden yararlanılmıştır. Dalgacık dönüşümü her kanal için ayrı ayrı hesaplanmıştır ve dalgacık katsayılarının ortalama mutlak değerleri özellik vektörleri olarak kullanılmıştır. Sınıf ayrılabilirliğinin artırılması için dalgacık katsayılarının ortalama mutlak değerlerine Doğrusal Diskriminant Analiz (DDA) işlemi uygulanmış ve DDA çıkışları sınıflandırılması için bir YSA'ya verilmiştir. Veri alımında ise 250 ms' lik (256 örnek) kayan pencere kullanılmış pencere artımı olarak da 125 ms (128 örnek) seçilmiş, % 97.2 gibi bir doğruluk oranı ile sınıflandırma işlemi tamamlanmıştır. Jun-Uk ve ark. (2006) ise gerçek zamanlı 4 kanallı EMG işaretinin sınıflandırılması işleminde Jun-Uk ve ark.(2007)' dan farklı olarak dalgacık katsayılarının boyut azaltımı için Temel Bileşenler Analizi (TBA) işleminden yararlanmıştır. TBA işleminin sınıf ayrılabilirliğini arttırmak için ise Öz Düzenleyici Haritalar (ÖDH) işleminden yararlanılmıştır (Şekil 1.5). Bu çalışmalar, EMG örüntü tanıma için etkin bir özellik çıkarma ve yansıtma yöntemi geliştirmeyi amaçlamaktadır.

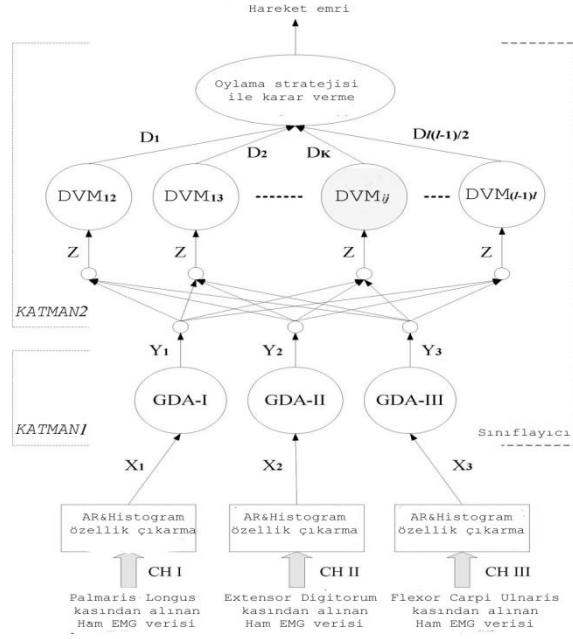


Şekil 1.4 Deneysel düzenek ve el hareketleri (Jun-Uk ve ark. , 2007)



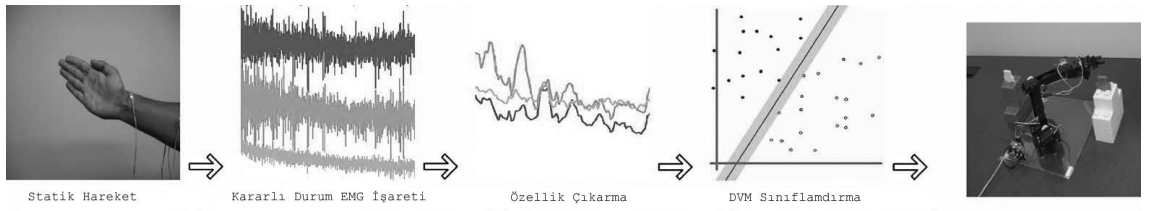
Şekil 1.5 (a) TBA ile azaltılmış özelliklere ait iki temel bileşen (b) ÖDH tarafından elde edilen özellik düzlemi (Jun-Uk ve ark. , 2006)

Liu ve ark. (2007), EMG tanımayı yüksek doğrulukla başarmak için geliştirilmiş diskriminant analiz (GDA) ve DVM'den oluşan bir sınıflayıcı (Şekil 1.6) önermiştir. Üç farklı kasa yerleştirilerek elde edilen EMG işaretlerinden histogram ve AR modeli çıkarılmış ve %93.53 sınıflandırma başarımı elde edilmiştir.



Şekil 1.6 GDA ve DVM den oluşan sınıflayıcı

Robot kolun gerçek zamanlı olarak yüzey EMG si ile denetlenmesi araştıran ve iki aşamadan oluşan başka bir çalışmanın birinci aşamasında önkol kaslarından elde edilen EMG işaretlerinin her bir kanal için 128 lik örnekleme pencerelerinin RMS genlikleri alınarak elde edilen yedi sınıflı bir sınıflama problemi %92-98 doğrulukla çevrimdışı olarak başarılmıştır. Sınıflayıcı olarak destek vektör makinaları kullanılmıştır (Şekil 1.7). İkinci aşamasında ise dört serbestlik dereceli bir robot kolun çevrimiçi denetimi tasarımı tanımlanmıştır (Shenoy ve ark. , 2008).



Şekil 1.7 EMG tabanlı denetim için şematik gösterimi (Shenoy ve ark. , 2008)

Son yıllarda çok işlevli mayoelektrik işaret denetimli protez uzuvlara hem akademik çevrelerden hem de ticari firmalardan yoğun bir ilgi oluşmuştur. Bu ilginin en temel gerekçesi, protez uzuvlara (sanayileşmiş ülkelerde iş ve trafik kazaları sonucu) gereksinimin artmasıdır. Protez uzuvların (el veya kol) serbestlik dereceleri (ve kanal

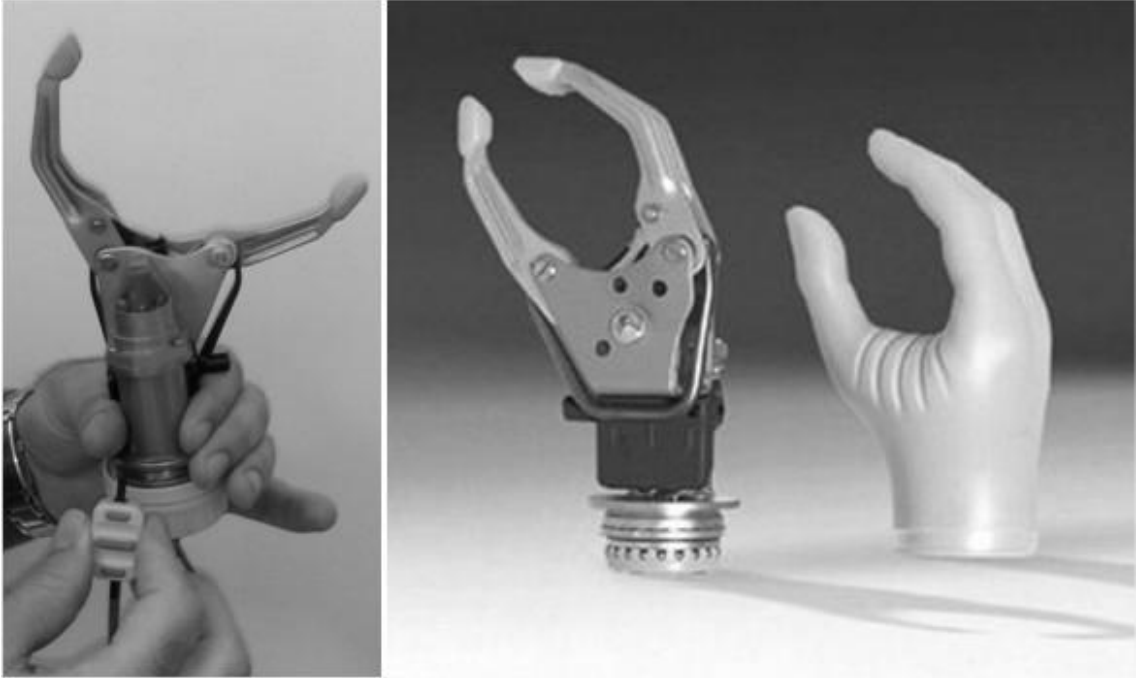
sayıları) arttıkça, doğal olarak daha güvenilir işaret işleme algoritmalarına, daha kullanışlı aktüatörlere ve sensörlere ihtiyaç duyulmaktadır.

Son yıllarda engelliler için aktif el protez cihazlarının geliştirilmesine teknolojinin ilerlemesiyle yeniden bir ilgi olmuştur. Pasif protezlerin aksine, aktif cihazlar, kullanıcının niyeti ile denetlenebilir. Aktif protez cihazlar gelişimi, daha işlevsel ve gerçekçi hareketler elde etmek için robot teknolojisine tümleştirilerek önemli ölçüde artmıştır. Protezlerin teknolojik gelişimindeki önemli ilerlemeye rağmen, klinik uygulamalar hala kullanıcının niyetini anlamak için kullanılabilecek biyolojik sinyallerin miktarı ve kalitesi ile istenen hareket için kullanıcının niyetini anlayan algoritmaların oldukça kötü performansı yüzünden sınırlı kalmaktadır. Mayoelektrik protezler, şu anda sadece aktif protezlerin klinik olarak uygulanmasından ibarettir. Bu sistemler engelli bir kişinin kalan kaslarını denetleme kaynağı olarak kullanır ve elde edilen sinyallerdeki kodu çözerler. Bu kas sinyalleri cilt yüzeyine yerleştirilen elektrotlar (yüzey EMG işaretleri) ile algılanır ve tipik olarak üst koldaki biceps ve triceps kaslarından elde edilirler. Ticari protezler el açma / kapama, bilek döndürme ve kol bükme işlevlerini sunarlar. Çoklu serbestlik derecesine sahip deneysel protez el/kol araştırma projelerinden biri avrupa birliği projesi kapsamında yürütülen “Cyberhand” ve ”Smarthand” projesidir. Bu projede 16 serbestlik dereceli protez elin teknolojik gelişimi amaçlanmıştır. Denetim için çevresel ve merkezi sinir sisteminden çıkarılan bilgi kullanılmakta olup protez aygıt geliştirilmiş durumdadır. Fakat sinirsel işaretleri kullanarak denetim ispatlanmamıştır. DARPA “Revolutionizing prothetics” projesinde ise el becerisi ve duyuşsal algılama açısından, sağlam bir insan kolu ile aynı işleve sahip bir protez kol geliştirmek amaçlanmıştır. Denetim için kaslar, çevresel sinirler ve beyinde çıkarılan bilgi kullanılmaktadır. Protez geliştirilmiş olup parmak, bilek ve kol hareketleri yapılabilmekte ve algısal geribesleme içermektedir. Engelli hasta için ameliyatla sinirsel bağlantıların yapılması gerekmektedir. Ottobock “Michelangelo” projesi kapsamında ise 5 serbestlik derecesine sahip yüzey EMG işaretleriyle denetlenebilen protez uzuv geliştirme çalışmaları yapılmakta olup 2-3 yıl içinde aygıtın piyasaya sürülmesi planlanmaktadır (Yoshida ve ark. , 2010).

Ticari olarak yapılan insan bedeniyle tümleşik çalışma örnekleri Şekil 1.8’de gösterilmiştir. Şekil 1.9’da ise Ottobock firmasının sadece mayoelektrik iki kanallı olarak çalışan ve sadece aç – kapa hareketi yapan protez eli görülmektedir.



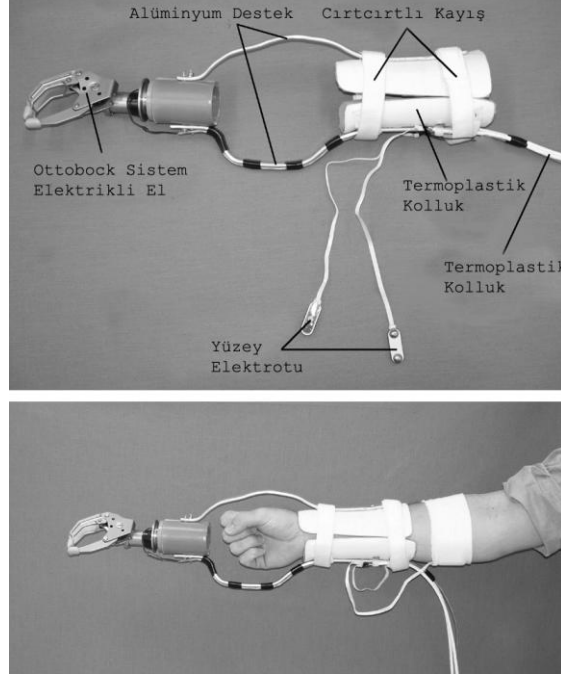
Şekil 1.8 Protez kolun insan bedeni ile tümleştirilmiş gösterimi



Şekil 1.9 Ottobock firmasının mayoelektrik protez eli

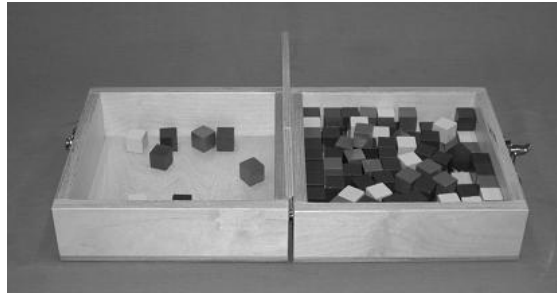
Ticari protezlerde büyük gecikmeler bir protezin tepkisinde azalmalar yarattığı için başarımların düşüklüğüne neden olmaktadır. Farrell ve Weir (2007) tarafından

Şekil 1.10'da verilen PHABS adı verilen bir düşük hızlı bir protez cihaz kullanılarak kutu testi yapılmaktadır. Testler iki farklı yapay el hızı ile ve yaklaşık 0-300 ms gecikme aralığındaki yedi farklı seviyede denetleyici ile gerçekleştirilmiştir.



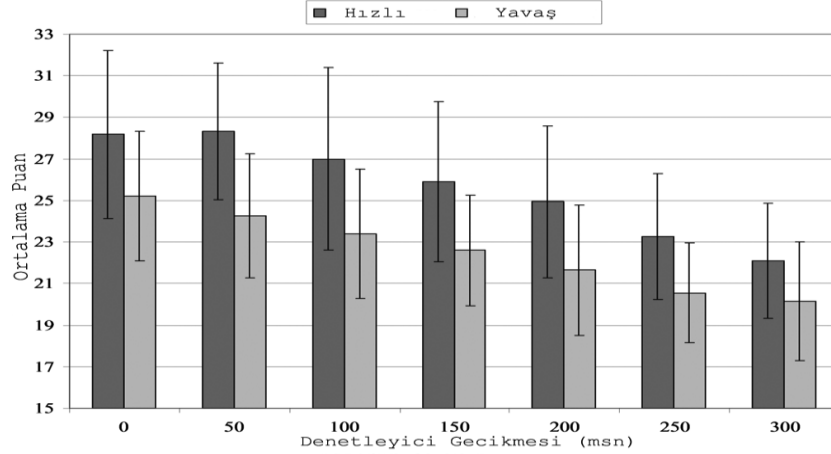
Şekil 1.10 PHABS protez düzeneği bileşenleri ve kullanımı (Farrell ve Weir, 2007)

Şekil 1.11'de gibi kutu testi ile protez el ile kutular bir taraftan alınarak bariyerin diğer tarafına aktarılmaktadır. 1 dakikada bariyerin diğer tarafına aktarılan kutuların sayısı puan olarak verilmektedir.



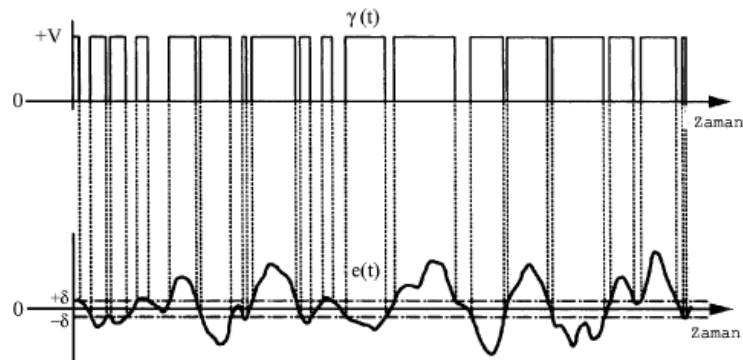
Şekil 1.11 Kutu testi için kullanılan düzenek (Farrell ve Weir, 2007)

Kutu testi ile hızlı ve yavaş protezler için ortalama puanların gecikme süresine göre grafiği Şekil 1.12'deki gibi elde edilmiştir. Denetleyici gecikmesi arttığında ortalama puanların azaldığı görülmektedir. Ayrıca hızlı denetleyicinin kullanıldığında başarımın daha iyi olduğu gözlenmiştir.

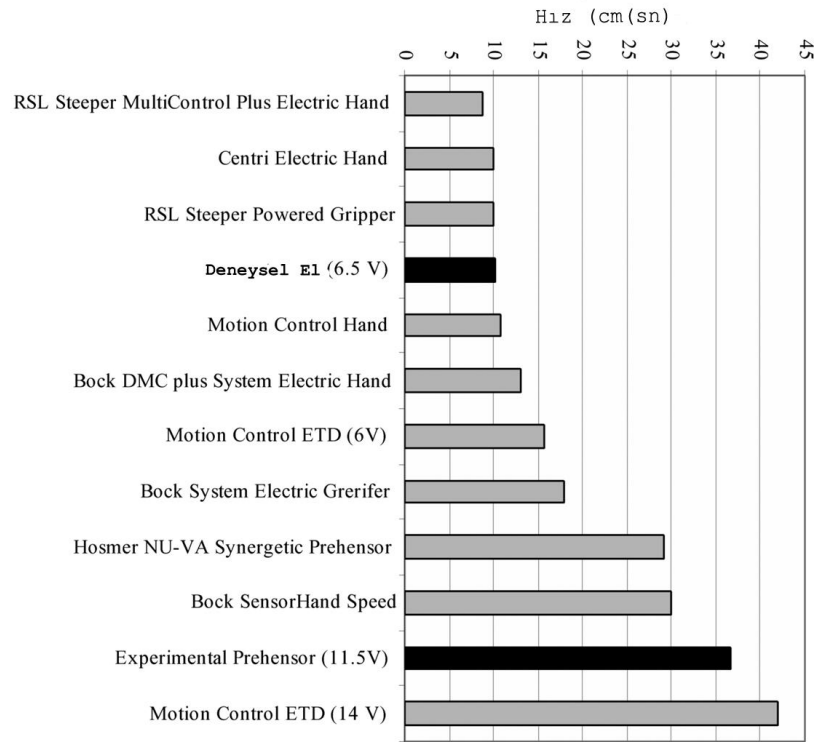


Şekil 1.12 Hızlı ve yavaş protezler için kutu testi sonuçları (Farrell ve Weir, 2007)

Denetleyici olarak mayo-darbe denetleyici DGB ve darbe sıklık bindirimi (DSB) in bir kombinasyonu gibi düşünülebilir. Mayo-darbe denetleyici EMG işaretinin mutlak değerini önceden belirli bir eşik değeri ile karşılaştırmaktadır. Bir kanal için mayo-darbe denetleyici grafiksel örneği Şekil 1.13'de görülmektedir. Mayo-darbe denetleyici Matlab (The Mathworks, Inc., Natick, MA) Simulink, Simulink Real Time ve XPC Target araçları kullanılarak oluşturulmuştur. Piyasada mevcut olan protez eller hızlarına göre Şekil 1.14'de verilmiştir. Deneyde kullanılan düşük hızlı protez elin altında hıza sahip 3 adet ticari protez olduğu görülmektedir .



Şekil 1.13 Mayo-darbe denetimi grafiksel gösterimi (Farrell ve Weir, 2007)



Şekil 1.14 Mevcut ticari protezler için doğrusal hızlar

Tekrarlanan ölçümler ve analizler sonucunda en iyi denetleyici gecikmesi ortalama bir kullanıcı için tahmini 100-175 ms arasında olmalıdır. 90. yüzdelerde bulunan bir kullanıcı için ise 100-125 ms arasında olması gerektiği önerilmektedir.

2. GENEL BİLGİLER

2.1. Giriş

Mayoelektrik işaretler vücudun ürettiği bir elektriksel büyüklük olmasının yanı sıra belli bir karakteristiği ve dolayısı ile kendini oluşturan kas hareketleri hakkında bilgi taşıması bu işaretlerin özellikle protez el kol tasarımı için bir ham madde durumunda olmasını sağlamıştır. Yapısı itibarı ile biyomedikal mühendisliğinin ilgi alanına giren EMG işaretleri pek çok üniversitenin biyomedikal mühendisliği bölümlerinde, daha işlevsel elektromekanik el-kol geliştirilmesi için incelenmektedir.

Ticari protezler, EMG işaretlerini zaman düzleminde inceleyerek denetleme işlemini gerçekleştirmektedir. Kullanılan elektrot sayısı veya işaret alınan kas grubu sayısına göre üretilen çeşitli denetim stratejileri ile protez uzuvlar genellikle AÇ-KAPA prensibiyle çalışmaktadır.

EMG işaretlerinden özellik çıkarma aşamasında bir başka yöntem ise işareti sıklık düzleminde incelemektir. Bu durumda elde edilen yüksek boyutlu veriler çeşitli boyut azaltma yöntemleri ile daha düşük boyutlu bir uzaya aktarılırlar. Elde edilen bu özellikler, anlamlandırılmak üzere bir sınıflandırıcıya verildiği zaman düşük boyutu sayesinde başarıyı yüksek olacaktır.

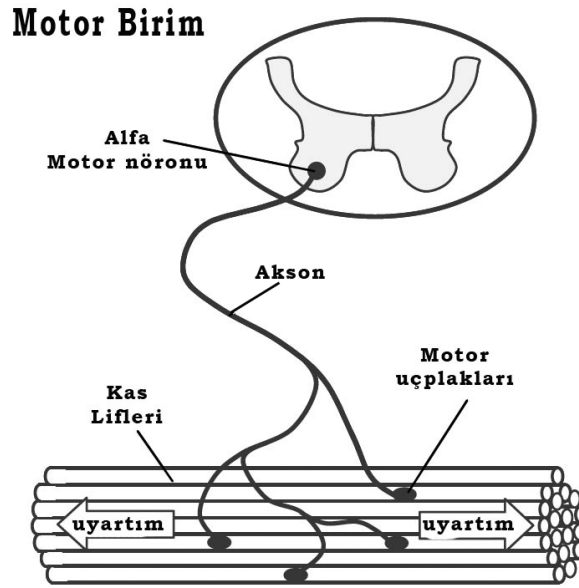
2.2. Mayoelektrik işaretler

Mayoelektrik işaretler ya da elektromyografik işaretler kas sistemi faaliyetlerini temsil eden kasılma sırasında kaslarda üretilen elektriksel akımını ölçen biyomedikal işaretlerdir. Kas aktiviteleri, sinir sistemi tarafından denetlenmekte olup ürettiği işaretlerin yapısı karmaşıktır (Raez ve ark. , 2006).

EMG işareti günümüzdeki çeşitli kullanım alanları sahiptir. Başlıcaları: Teşhis ve tanı koyma, kinesioloji, ergonomi, yapay uzuv denetimi (Sörnmo ve Laguna, 2005).

Kaslar, kasılabilen yapısal birimlerden oluşan ve birbirlerine hemen hemen paralel uzanan hücreler olan kas liflerinden oluşmuştur. İnsanlarda kas lifleri, birkaç milimetreden birkaç santimetreye kadar değişen uzunluklarda ve yaklaşık 10 ile 100 mikrometre arasında değişen çaplarda bulunmaktadır. Her bir lif, uyarıldığı zaman

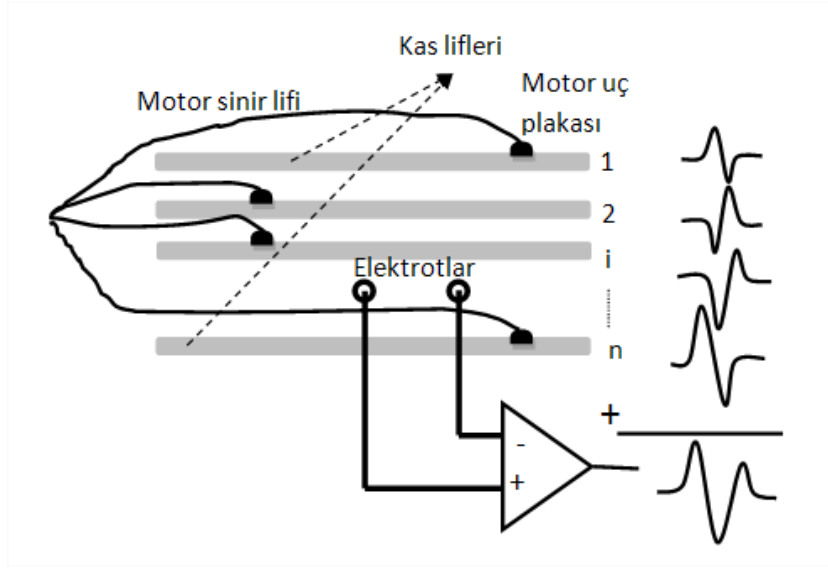
uzunluğunu kısaltabilir. Buna “kasın kasılması” adı verilir. Kas lifleri, motor nöronlara iletilen elektrik sinyalleri ile merkezi sinir sistemi tarafından aktive edilirler. Bir motor nöron, kasın en küçük işlev birimini oluşturan kas lifleri grubunu sinir sistemine bağlar. Motor nöronlar ve liflerin o sinir sistemine bağlantılarına motor birim denir (Zouridakis ve Moore, 2004).



Şekil 2.1 Motor Birim yapısı(Konrad, 2005)

İskelet kaslarının kasılmaları denetleyen elektriksel darbeler, merkezi sinir sisteminden çevresel sinir sistemine oradan da kaslara iletilir. Aksiyon potansiyelleri motor nöronların aksonları boyunca taşınır ve her bir motor nöronu (alfa motor nöron) kas fiberlerine özelleşmiş motor uç plakları ile bağlıdır. Nöronların taşıdığı bu elektriksel darbelere “nöral aksiyon potansiyelleri“ denir. Nöral aksiyon potansiyelleri nöromaskular (sinir ve kas sistemi yapısı) bağlantıya ulaştığında kas lifleri uyarılır. Motor birim aksiyon potansiyeli her bir darbeye kasın verdiği elektriksel tepkidir ve bu elektriksel etki kas lifleri boyunca yayılır. Şekil 2.2 ile gösterilen biçimde bir motor birimindeki motor plakalarında, uyarı geldiği zaman aynı anda aksiyon potansiyeli oluşur. Bu aksiyon potansiyelleri motor plakalarının elektrotlardan olan uzaklığına göre farklı genliklerde bifazik değişimler biçimindedir. Bu değişimlerin fazları ise motor uç plağının elektrotların sağ veya sol tarafında oluşuna göre değişir. Sonuç olarak elektrotlarda, motor uç plaklarının aksiyon potansiyel değişimlerinin etkileri, bu

potansiyellerin toplamı olarak gözlenir ve bu aksiyon potansiyellerin toplamına Motor Ünite Aksiyon Potansiyel (MÜAP) adı verilir.

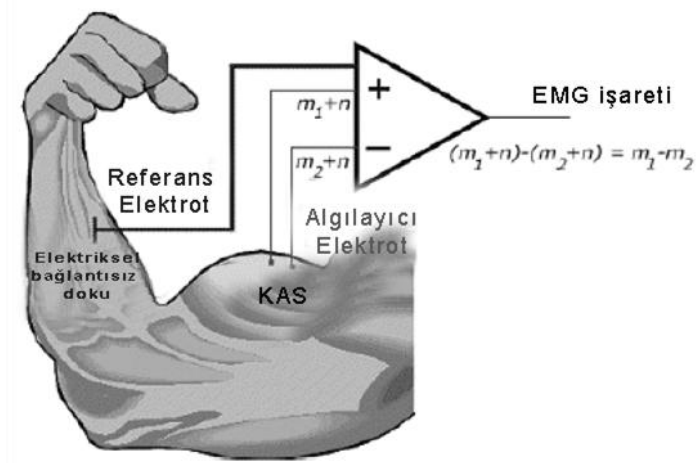


Şekil 2.2 Motor ünite aksiyon potansiyeli (MÜAP)

2.3. EMG işareti kayıt yöntemleri

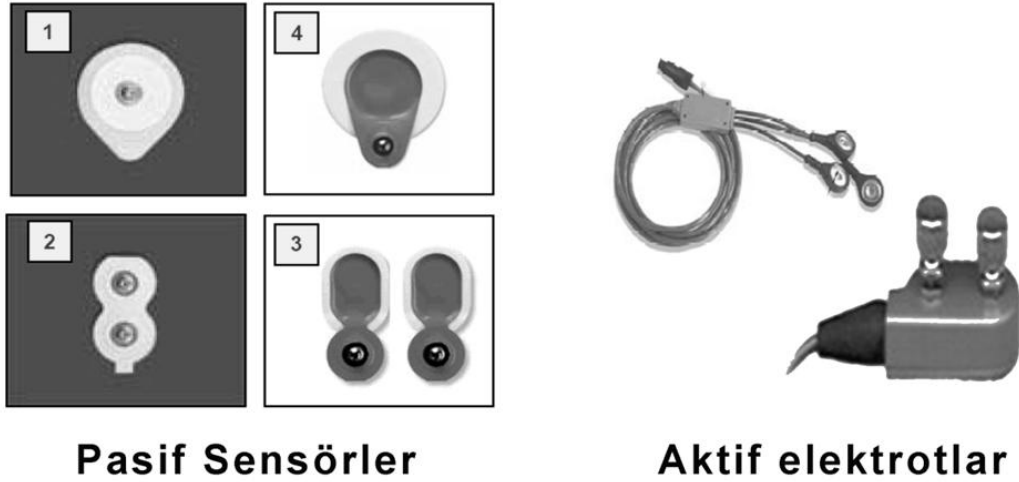
Kasların kasılması, sinirler aracılığıyla beyinden iletilmiş olan uyarıcı potansiyellerin kaslarda oluşturduğu Motor Ünite Aksiyon Potansiyelleri (MÜAP) olarak bilinen elektriksel potansiyeller sayesinde olur.

Bu aşamada ilk önce kaslardan elektrotlar (sensörler) aracılığıyla Şekil 2.3’de gösterildiği biçimde bir fark yükselteci ile biyomedikal kas işaretleri elde edilir. Görüleceği üzere algılama esnasında elektromagnetik gürültü sabittir fakat neticede sensörlerin yerleştirildiği iki noktada da aynı gürültü mevcut olduğundan fark, kaslarından aktivasyonunun yükseltilmesidir.



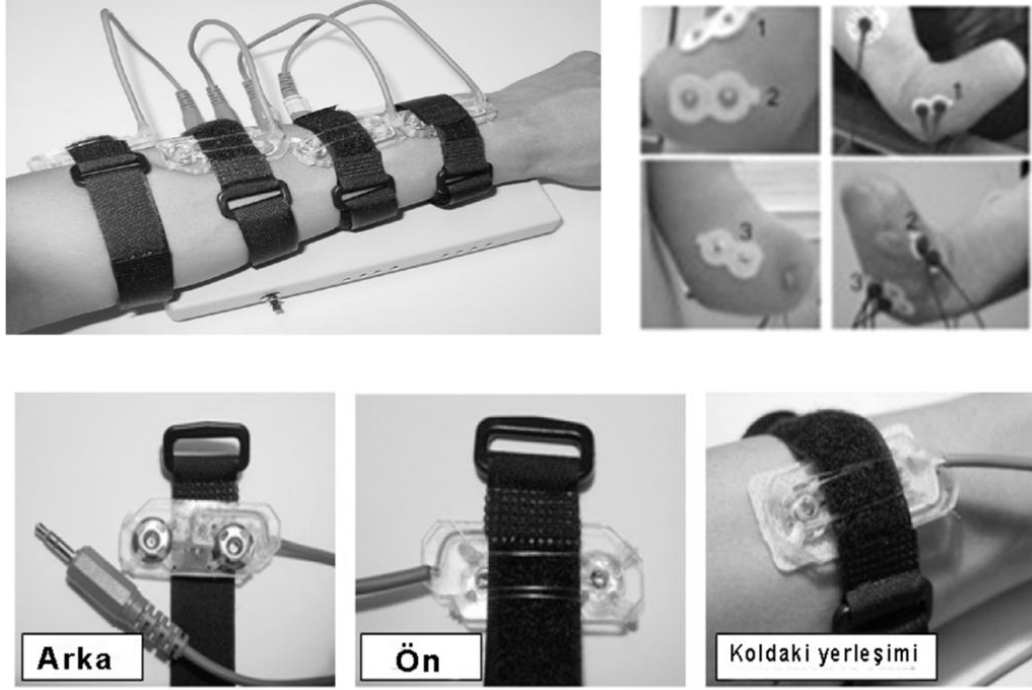
Şekil 2.3. Kasta fark yükseltici ile işaret alınması (Luca, 2002)

Bu işaretlerin elde edilmesinde kullanılan sensör türleri Şekil 2.4’de gösterilmiştir. Bunlar aktif ve pasif sensörler olup yüzeysel sensör gruplarıdır. Yani algılama işlevini kas yüzeyinden yaparlar. Ayrıca iğne elektrotları da mevcuttur ki bunlar ilgili kasa iğnenin yerleştirilmesi suretiyle kullanılırlar. Bu yönleriyle tıbbi personele ihtiyaç duyulması, yeniden konumlandırılabilmesinin zor olması gibi birçok dezavantaja karşılık yüksek duyarlılıkta ölçüm yapılabilmesi önemli bir avantajdır. Dezavantajları uygulanabilirliği olumsuz etkileyeceği için tez kapsamında yüzeysel sensörler kullanılacaktır.



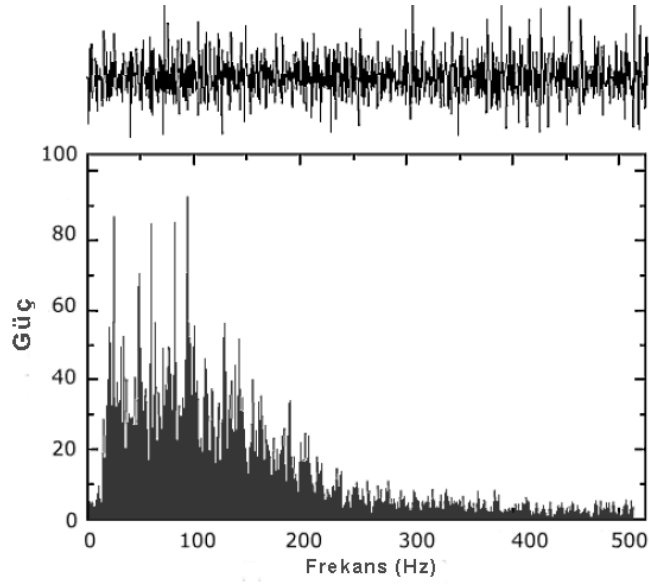
Şekil 2.4 EMG işaretlerinin elde edilmesinde kullanılan sensör türleri (Konrad, 2005)

Yüzeysel sensörler Şekil 2.5’de görüldüğü gibi hiçbir tıbbi uygulamaya ihtiyaç duyulmayan kullanımı kolay ve hızlı sensörlerdir ve ilgili kas yüzeyine yerleştirilmek suretiyle kullanılabilirler. Görüleceği gibi kişinin engelli veya sağlam olması sinyalin alınması durumunu etkilememektedir. Önemli olan kasların elektriksel etkinliğinin mevcut olmasıdır.



Şekil 2.5 Elektrotların sabitlenmesi ve engellilerde kullanım örnekleri (Ajiboye ve Weir, 2005)

Şekil 2.6’da ise bir EMG işaretinin sıklık güç spektrumu görülmektedir. Bir EMG işareti karakteristik olarak, tepeden tepeye 0-10mV genliğe sahiptir. Kullanılabilir enerji aralığı 0-500Hz , baskın enerji aralığı ise 50-150Hz ‘dir (Luca, 2002; Konrad, 2005).



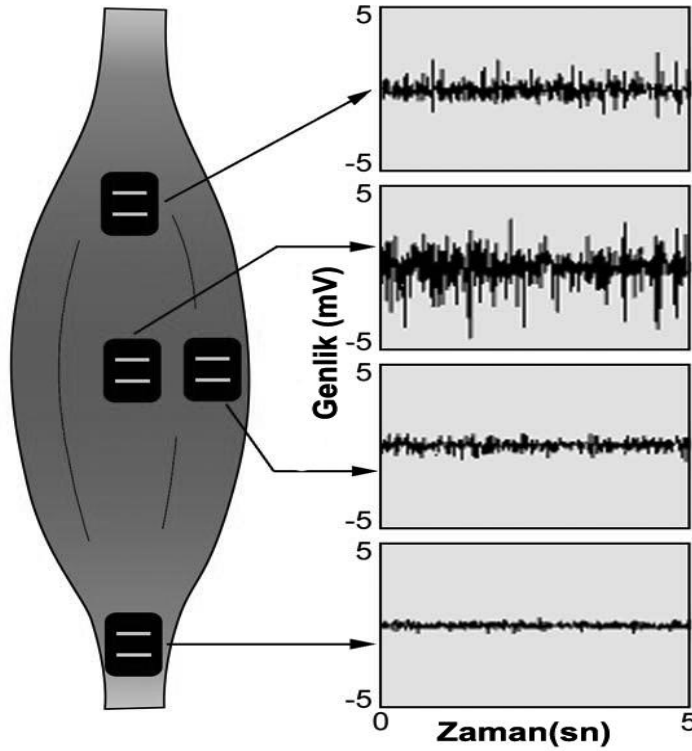
Şekil 2.6 EMG işareti güç sıklık tayfı (Luca, 2002)

Bu karakteristik özelliklere sahip olan EMG işareti birçok elektriksel gürültüden etkilenir. Bunlar 4 grupta toplanabilir (Luca, 2002; Day, ; Konrad, 2005) :

1. Elektriksel teçhizatın içsel gürültüsü: Tüm elektriksel teçhizat bu gürültüye sahiptir. Sıklık aralığı 0 ile binler düzeyinde Hz aralığında değişebilen bu gürültü giderilemez. Ancak yüksek kaliteli elemanlar kullanılarak azaltılabilir.
2. Çevre gürültüsü: Radyo iletimi, elektrik kabloları , flüorasan aydınlatmaları gibi elektromagnetik ışıyım kaynaklarının oluşturduğu gürültü biçimidir. Kaçınılmaz olan bu gürültü türünün genliği 1-3xEMG sinyali aralığında olup baskın sıklığı 50Hz dir.
3. Hareket gürültüsü: Elektrot /deri arayüzü ve elektrot kablosu kaynaklı bir gürültüdür. Uygun devre ve kurulum ile azaltılabilir. Sıklığı 0-20Hz aralığındadır.
4. Sinyalin içsel kararsızlığı: 0 - 20Hz sıklık aralığına sahip genliği doğasına göre rastgele olabilen gürültüdür.
5. Fizyolojik gürültü: Kayıt elektrotlarının yerinin uygun saptanamaması sonucu oluşan ve ilgilenilen kasa yakın diğer kasların ürettiği EMG işaretlerinin oluşturduğu gürültü bileşenidir. Bu istenmeyen etkinin %10-

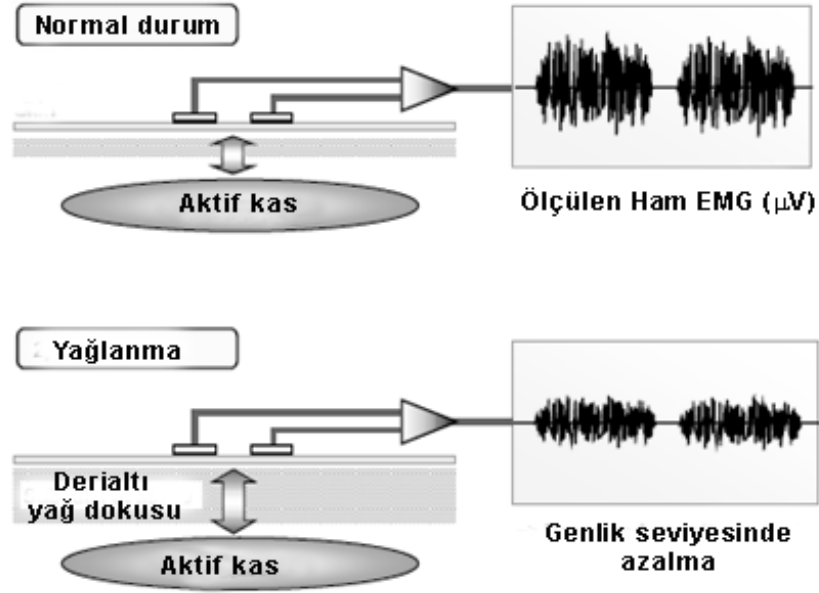
%15 sınırını aşması durumunda, EMG kaydı bundan büyük ölçüde olumsuz etkilenir.

Şekil 2.7, elektrotların kasta yerleşimini ve elde edilen işaretleri zaman ve sıklık düzlemine göre göstermektedir. Tendonlara yakın bölgelerde kas yoğunluğu az olmasından dolayı EMG işaretinin genliği oldukça düşük olacağı için elektrotlar tendonlardan uzak, diğer kasların merkezlerine daha yakın bölgelere yerleştirilmelidir. Ayrıca bu yerleşim kas liflerini kesecek biçimde paralel yönelimli olmalıdır. Ancak bu durum aynı kasa ait birden fazla kas fiberinin ürettiği EMG işaretinin kaydedilmesi mümkün kılmaktadır. Referans elektrotu ise bu elektrotlardan mümkün olan uzak bir noktaya, iyi bir elektriksel temas sağlayacak ve hareketten etkilenmeyecek biçimde yerleştirilmelidir.



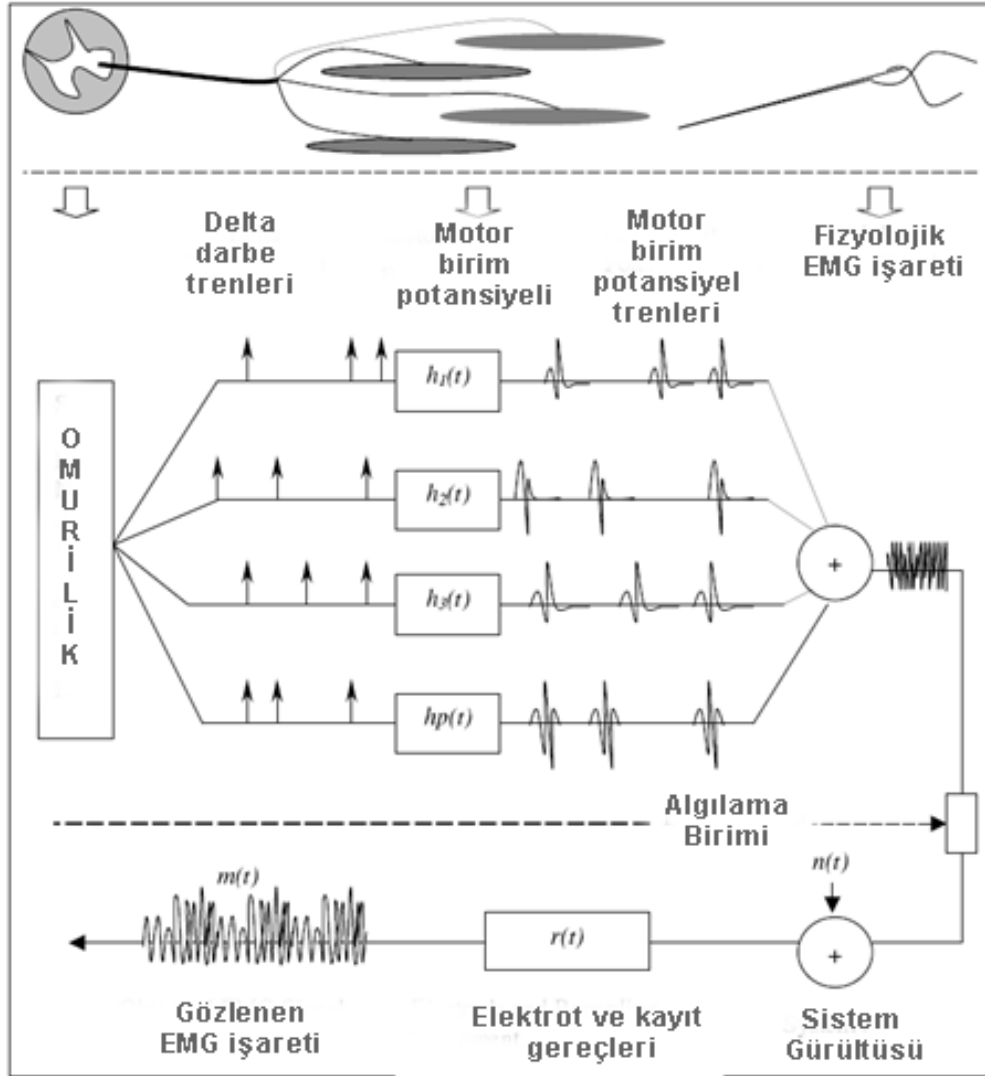
Şekil 2.7 Elektrotların kasta yerleşimi ve EMG genlik sinyalleri (Luca, 2003)

EMG kaydı, yaş, cinsiyet, deri altındaki yağ tabakası ve kayıt elektrotlarının vücutta yerleştirilme biçimleri gibi faktörlerden etkilenir. Şekil 2.8, yağ tabakasının EMG kaydına etkisini göstermektedir.



Şekil 2.8. Yağ tabakası kalınlığının yüzey EMG kaydına etkisi (Konrad, 2005).

Gürültü bileşenlerinin tümü düşünüldüğünde, gürültü genliği işaretin genliğinin 1-3 katı olabilmektedir. İşlenebilir işaret elde etmek için, ortamdaki gürültünün bastırılması gerekmektedir. Bunun için ortak-kip bastırma oranı (**Common Mode Rejection Ratio, CMRR**) çok yüksek (en az 85, ortalama 100 dB) ve kazancı düşük bir ön yükseltece gereksinim vardır. Bu şekilde gürültü etkisinden arıtılan işaret tekrar yükseltilerek uygun çalışma gerilimi aralığına taşınır (± 5 V, ± 15 V) ve sayısal bilgiye dönüştürülebilir hale gelir. Şekil 2.9’da EMG işaretinin elektriksel benzetimi ve modeli basit bir şema ile verilmiştir.

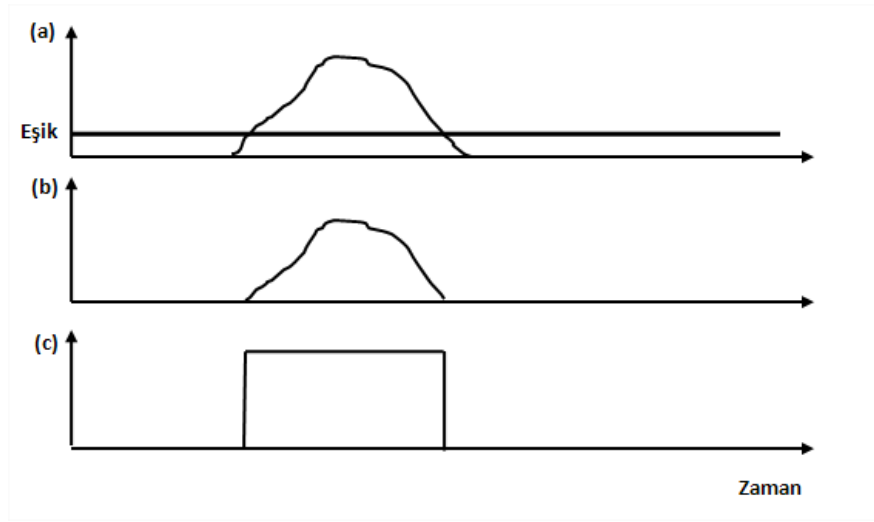


Şekil 2.9 EMG işaretinin biyolojik kökenini, elektriksel benzetimini ve ölçülen EMG işaretinin gösterimi (Luca, 1979).

2.4. Denetim Stratejileri

Denetim stratejisi, protez ile istenilen hareketi yaratmak için protezi kullanan kişiye ait girişlerin nasıl olacağını belirler. Kullanıcı için seçilen strateji türü bir çok faktöre bağlıdır. Bunlar arasında, kasların ürettiği sinyalin karakteristiği, kişinin isteklerini yerine getirecek bileşenlerin sayısı, kasların yerleşim sayısı, kullanılan giriş aygıtları ve kişinin bilişsel yeteneği sayılabilir.

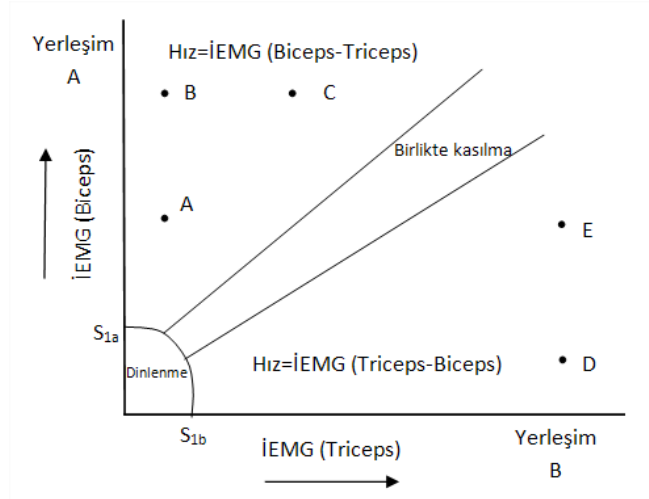
Denetim sistemi sayısal ve oransal olmak üzere protezi iki biçimde denetler. Sayısal denetimde giriş sinyali belirlenen eşiği aştığında motorun hızı en yüksek ve sabit olacaktır. Oransal sistemde ise motorun hızı giriş sinyalinin genliğindeki değişime bağlı olarak oransal biçimde değişecektir (Şekil 2.10). Engelli kişi, oransal sistemde protez bileşenlerinin hızını denetleyebildiğinden bu sistem parmak hareketleri gibi hassas hareket denetimlerinde kullanılır. Ayrıca bu sistem elektriksel elin yukarı ve aşağı hareket hızının denetiminde de oldukça kullanışlıdır (Muzumdar, 2004).



Şekil 2.10 (a) EMG işaretinin zarfı çıkarılarak elde edilen denetim işareti (b) Oransal denetim işareti (c) Sayısal denetim işareti

Örneğin; Liberating Technologies Inc.'nin Boston dirseği oransal denetimi desteklemektedir. Denetim iki yolla yapılmaktadır. Birincisinde tek yerleşimli iki düzeyli strateji ile belirlenen iki eşik kolun yukarı veya aşağı hareketini ayırt edilirken işlenmiş EMG işaretinin genliği ile dirseğin hızı denetlenmektedir. İkincisinde ise Biceps ve Triceps kaslarına yerleştirilen elektrotlar sırasıyla aşağı ve yukarı hareketi yaptırmaktadır. Bu fiziksel kolun denetiminde aynı kas gruplarını kullanan travmatik hastalar için kullanılan bir düzenlemedir. Her bir kastaki tek bir eşik düzeyi (S_{1a} ve S_{1b}) tıpkı iki yerleşimli AÇ-KAPA denetimdeki gibi kullanılmaktadır. Fakat burada her bir yerleşimdeki işlenmiş EMG işaretinin genlikleri arasındaki fark protezin hızını denetlemekte kullanılmaktadır. Eğer Biceps kasından alınan işaret Triceps kasından alınan işaretin genliğinden büyükse kol aşağı, tam tersi durumda ise kol yukarı hareket

eder. Bicep ve Triceps kaslarının belirlenmiş beş noktası A'dan E'ye etiketlenmiş olarak Şekil 2.11'de verilmektedir (Muzumdar, 2004).



Şekil 2.11 Kol için oransal denetim örneği(Muzumdar, 2004)

B ve C noktalarında kolun hızı düşünüldüğünde, Biceps işaretleri aynı olmasına rağmen C noktasından Triceps aktivasyonu daha yüksek olduğu için kolun hızı B noktasına göre daha azdır. Bu hız denetim yapısı iki kanal arasındaki aktivasyon farkları ile protez hareketini ilişkilendirmektedir (Çizelge 2.1).

Çizelge 2.1 İki yerleşimli Boston kolu için protez hız ve işlevi

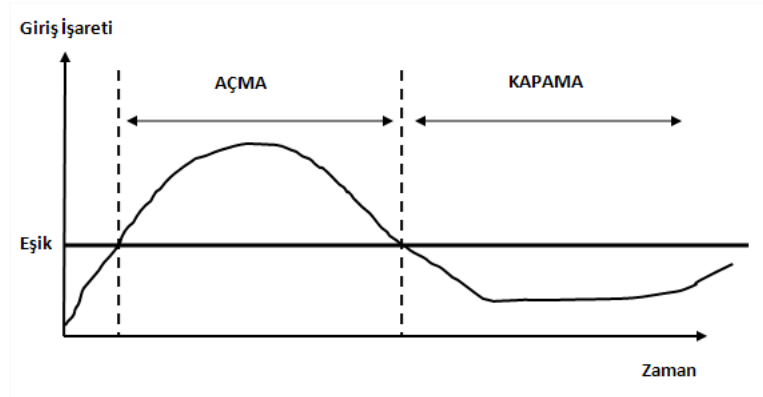
Nokta	Biceps	Triceps	İşlev
A	Orta	Düşük	Yavaş , aşağı hareket
B	Yüksek	Düşük	Hızlı, aşağı hareket
C	Yüksek	Orta	Yavaş, aşağı hareket
D	Düşük	Yüksek	Hızlı, yukarı hareket
E	Orta	Yüksek	Yavaş yukarı hareket

Oransal denetimin bir başka örneği Motion Control firmasının Utah Yapay Elinde kullanılmaktadır. Bu Ottobock elektrik elleri ile uyumlu iki yerleşimli bir sistemdir. Kol denetimi için kullanılan aynı iki kas yerleşimi elin denetimi için de kullanılmaktadır. Elin veya kolun denetim seçimi birlikte kasılma (co-contraction) işareti kullanılarak yapılır. Burada hangi cihazın (el/kol) aktif olduğu kullanıcının hatırlaması gereken bir noktadır (Muzumdar, 2004).

Ticari protez ellerde yaygın biçimde kullanılan altı farklı denetim stratejisi (Muzumdar, 2004) tarafından özetlenmiş olup aşağıda verildiği gibidir.

2.4.1. Tek yerleşimli tek düzeyli istemli açma stratejisi

Tek yerleşimli tek düzeyli istemli açma stratejisinde tek bir giriş (EMG elektrotu, güç duyarlı direnç vb.) vardır. Bu giriş işareti belirlenen eşik seviyesini aştığında el açılırken eşiğin altındaki değerlerde ise el kapanmaktadır (Şekil 2.12). Bu durumun tam tersi olan istemli açma hareketi yerine istemli hareketi de kişinin tercihinine bağlı olarak ayarlanabilir. Bu strateji 3 yaşından küçük çocuklar için uygundur. Çünkü bu yaştaki çocuklar istemli hareketleri yapmak için uygun yetenekte değildir ve kol kasları ile el açma kapama hareketini içgüdüsel olarak yaparlar.



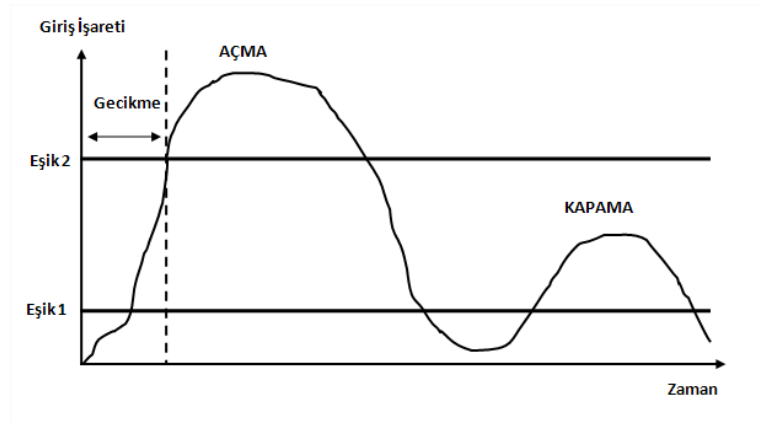
Şekil 2.12 Tek yerleşimli tek düzeyli istemli açma stratejisi

2.4.2. Tek yerleşimli iki düzeyli duyarlılık stratejisi

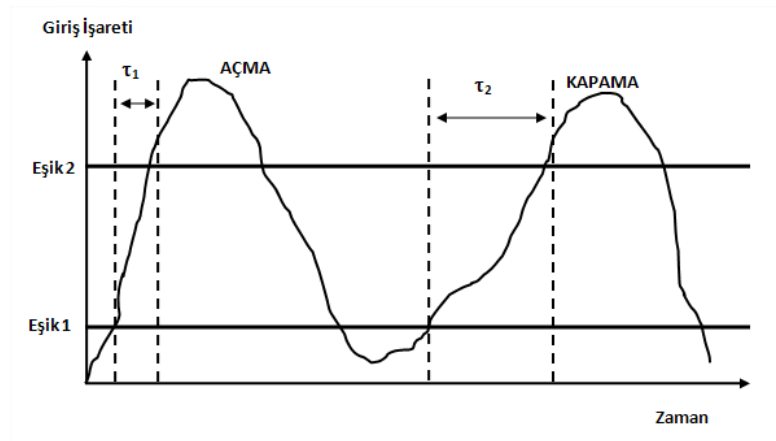
Tek yerleşimli istemli açma stratejisine benzer olarak bu stratejide de tek giriş vardır. Farklı olarak bu strateji eli açma/kapama veya eli iki yönde döndürme veya yukarı/aşağı hareketlerinin denetimine izin verir. İki düzeyli denetim stratejisinde Şekil 2.13’de görüldüğü gibi iki eşik değeri belirlenir. Eğer giriş işaretinin genliği düşük eşikten daha az ise protez el işlem yapmaz. Eğer giriş işaretinin genliği iki eşik değeri arasında ise el kapanır. Eğer giriş işareti yüksek eşiği aşarsa el açılır. Giriş - çıkış işareti arasındaki küçük gecikme düşük eşikten yüksek eşiğe geçişte yanlıyla elin kapanmasını önlemek için tanımlanmıştır.

2.4.3. Tek yerleşimli oransal duyarlılık stratejisi

Oran duyarlı denetim stratejisi düzey duyarlı denetim stratejisine bir seçenektir. İşaret genliğini kullanarak eli açma ve kapama işlevlerinin denetimini yapmak yerine kullanıcının kas kasılmalarındaki oran ile motorun yönünü belirler. Şekil 2.14’de görüldüğü gibi giriş işareti düşük eşiği geçtiğinde sistem zamanın periyodunu kurmak için msn’ler düzeyinde bekler ve sonrasında işareti üst eşik ile karşılaştırır. Kurma zamanı geçtikten sonra eğer işaret hala üst eşikten aşağıda ise bu yavaş yükselen bir işaretmiş gibi düşünülerek (τ_2 zamanı) el kapanır. Eğer sinyal kurma zamanından önce yüksek eşiği aşarsa bu hızlı yükselen bir işaret olarak düşünülür ve (τ_1 zamanı) sonucunda el açılır.



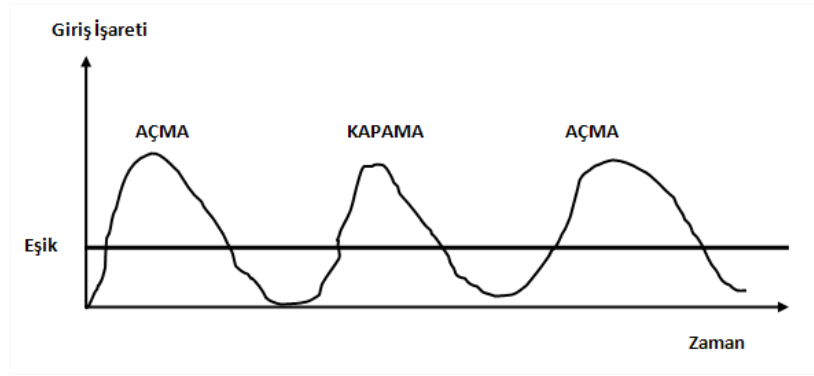
Şekil 2.13 Tek yerleşimli iki düzeyli duyarlılık



Şekil 2.14 Tek yerleşimli oransal duyarlılık

2.4.4. Tek Yerleşimli Sıralı Strateji

Sıralı strateji, tek yerleşimli ardışıl kas kasılmaları kullanılarak eli açma ve kapama arasındaki anahtarlamayı otomatik olarak yapmayı mümkün kılar. Şekil 2.15 ile verilen biçimde elin kullanım örneği şu biçimdedir: Giriş işareti eşiği ilk aştığında el açılır, ikinci eşik aşımında el kapanır, sonraki el aşımında el tekrar açılır ve bu sırada işlev devam eder. Kullanıcı son aktivasyondan itibaren önceden belirlenmiş bir zaman aralığından sonra benzer işlevlerden varsayılan denetim seçeneğini belirleyebilir. Bu kullanıcının daima bir sonraki giriş aktivasyonu olduğunda elin hangi hareketi yapacağını bilmesini sağlar.



Şekil 2.15 Tek yerleşimli sıralı strateji

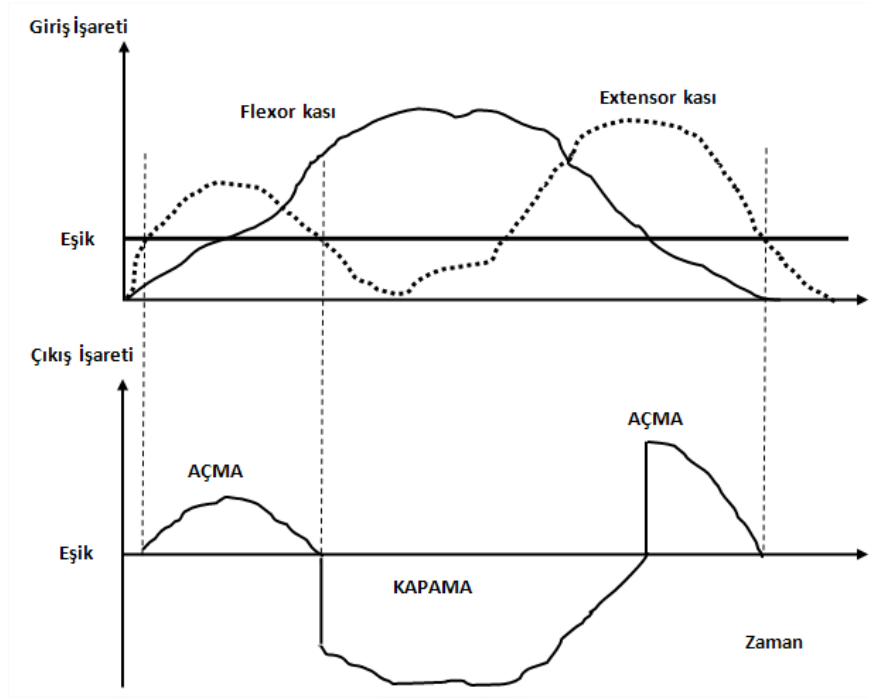
2.4.5. İki yerleşimli düzey duyarlı strateji

İki yerleşimli iki işlevli olarak da adlandırılan bu strateji en çok kullanılan denetim stratejisidir. El protezlerinde örneğin eli açmak için birinci yerleşim , eli kapamak için ikinci yerleşim kullanılır (Şekil 2.16). İki yerleşimli strateji kullanıcının yetenekleri ve tercihlerine bağlı olarak iki biçimde kullanılabilir.

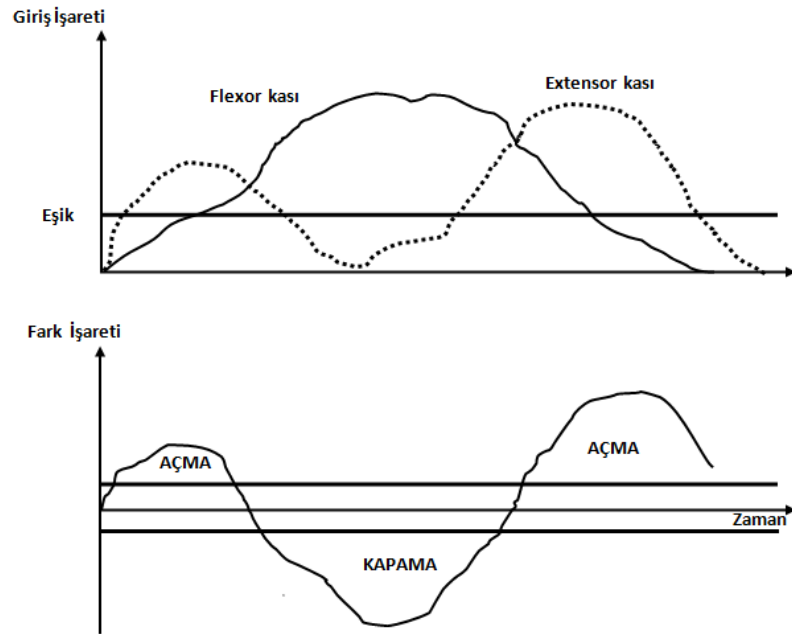
İlk gelen ilk hizmet eder: Bu stratejide eğer flexor kasları ilk olarak kasılırsa el kapanır, eğer extensor kasları ilk kasılırsa el açılır. Şekil 2.16 ile gösterilen bu durumda ikinci kas kasılmaları ihmal edilmiştir.

Farksal denetim: İlk gelen ilk hizmet eder stratejisine benzer olarak flexor kası kasılırsa el kapanır, extensor kası kasılırsa el açılır. Fakat her iki kas grubu birlikte kasılırsa (co-contraction) motorun yönü ve hızı flexor ve extensor

kaslarının sinyalleri arasındaki genlik farkıyla yönetilir (Şekil 2.17). İki yerleşimli düzey duyarlılık stratejisi sayısal ve oransal biçimde olabilir.



Şekil 2.16 İki yerleşimli iki düzeyli ilk gelen hizmet eder denetim stratejisi



Şekil 2.17 İki yerleşimli iki düzeyli farksal denetim

2.4.6. Mod anahtarlama stratejisi

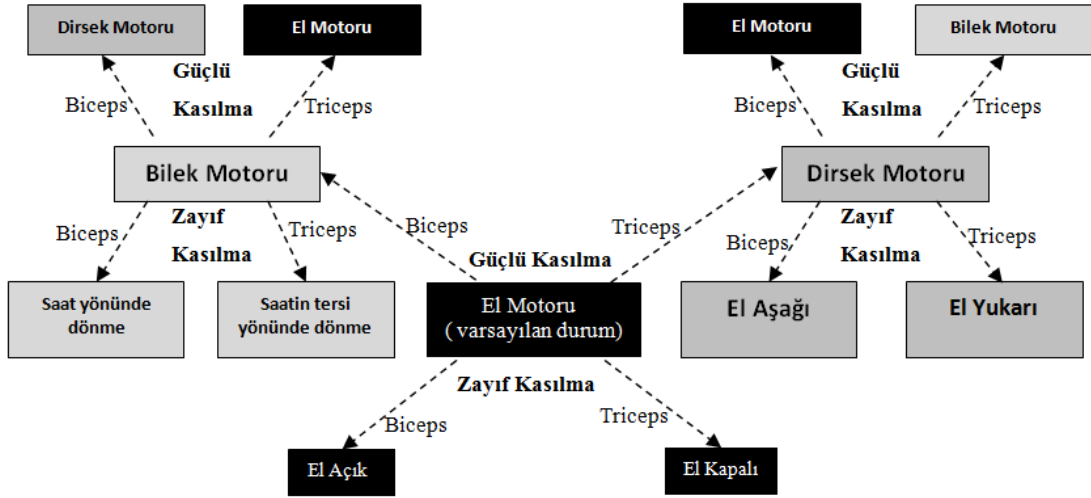
Mod anahtarlama, birlikte kasılma (co- contraction) durumunu kullanarak iki giriş sinyali ile bir veya birden çok aygıtı denetleme işlevini görebilen bir yöntemdir. Örneğin; flexor kaslarından elde edilen EMG işaretleri ile eli kapatırken, extensor kaslarından elde edilen EMG işareti ile eli açar. İki kas grubunun birlikte kasılması durumunda kullanıcı denetimi elden bileğe anahtarlayabilir. Bu durumda iken flexor kaslarından elde edilen EMG işaretleri bileği avuç içi yukarı bakacak biçimde dışa doğru döndürürken , extensor kaslarından elde edilen EMG işareti bileği ters yönde içe doğru döndürür. Tekrar elin denetlenmesi istenirse kullanıcı kas gruplarını tekrar birlikte kasmak durumundadır. Eğer varolunan durumda herhangi bir işaret üretilmezse de denetim otomatik olarak ele döner. Eğer kullanıcı yanlış bir kasılma gerçekleştirirse de bu durum hareket planının bozulması anlamına gelir ve hızlı bir biçimde başlangıç durumuna dönlür. Mod anahtarlama stratejisinin çok çeşitli biçimleri olabilir.

2.4.7. İki yerleşimli çoklu hareket stratejisi

Bu hareket stratejisi, birlikte kasılma (co-contraction) olmaksızın Biceps ve Triceps kaslarının güçlü veya zayıf kasılmalarına göre Şekil 2.18’de verilen şema kullanılarak protez hareketlerine karar verilmektedir. El motorunun seçimi varsayılan başlangıç durumu olarak belirlenir ve süreç üç ana başlıkla şu şekilde özetlenebilir:

1. Eğer Biceps kası zayıf kasılıyorsa el açılır, Triceps kası zayıf kasılıyorsa el kapanır.
2. Eğer Biceps kası güçlü kasılıyorsa bilek motoru aktif olur. Bu durumda Biceps kası zayıf kasılıyorsa el saat yönünde, Triceps kası zayıf kasılıyorsa saatin ters yönünde el döndürme hareketi yapılır. Eğer Biceps kası güçlü kasılıyorsa dirsek motoru, Triceps kası güçlü kasılıyorsa tekrar el motoru aktif edilir.
3. Eğer Triceps kası güçlü kasılıyorsa dirsek motoru aktif olur. Bu durumda Biceps kası zayıf kasılıyorsa el aşağı yönde, Triceps kası zayıf kasılıyorsa el yukarı yön hareket eder. Eğer Biceps kası güçlü kasılıyorsa el motoru, Triceps kası güçlü kasılıyorsa bilek motoru aktif edilir.

Kol hareketlerini de kapsayan bu strateji ile aynı anda sadece bir kası zayıf veya güçlü kasma işlemi ile toplam 6 hareketi denetleyebilmektedir.

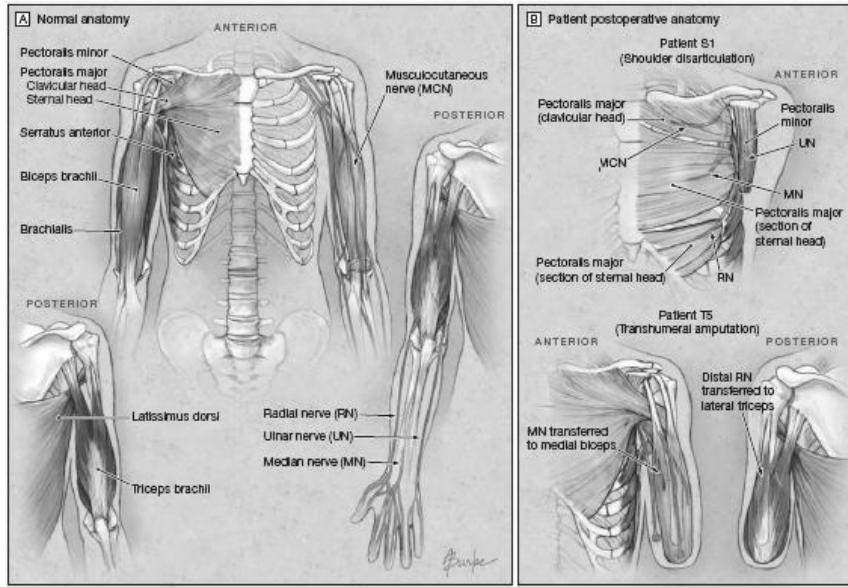


Şekil 2.18 Kas aktivitelerine göre protez hareket şeması (Artigue ve Thomann, 2009)

2.4.8. Hedeflenen kasın başka sinirlerle sinirlendirilmesi

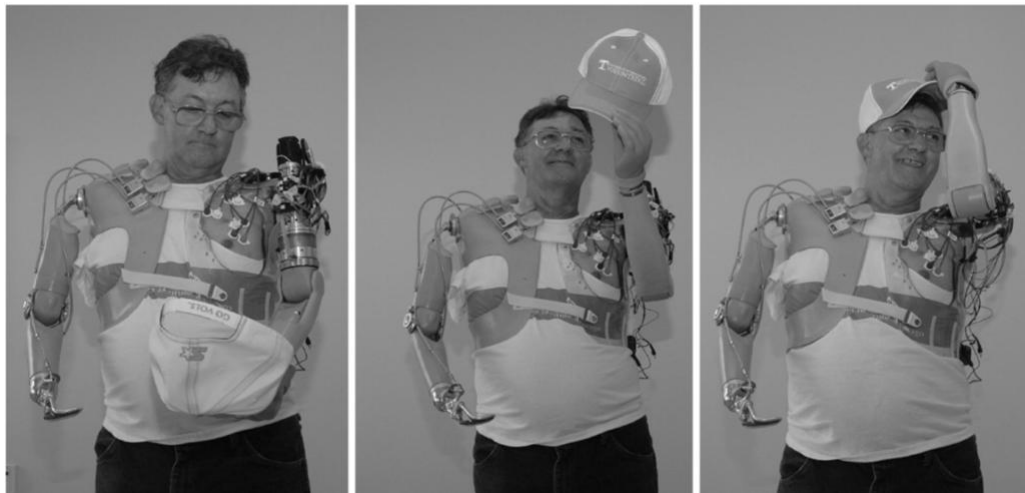
Mevcut denetim stratejileri protezin her bir işlevini çalıştırmak için engelli kişinin kalan bir veya iki kasına (Biceps veya Triceps) ait EMG işaretlerini sıralı biçimde kullanır. Kalan kasların fiziksel olarak ilişkisiz hareketleri denetlemesi nedeniyle bu çalışma şekli sezgisel değildir. Mayoelektrik protez denetimini geliştirmek için yeni bir yöntem olarak hedeflenen kasın başka sinirler ile sinirlendirilmesi (Targeted Muscle Reinnervation ,TMR) geliştirilmektedir (Kuiken ve ark. , 2009).

Otto Bock tarafından TMR yöntemiyle üretilen bir prototip kol protezi geliştirilmiştir. Dört ana kol sinirleri medyan sinir, radyal sinir, ulnar sinir ve muskulokütan sinir cerrahi işlemle göğüs kaslarına yönlendirilir (Şekil 2.19). Bu işlemten sonra cilt yüzeyinde EMG işaretleri üretebilen hedef kaslar ölçülerek protez kollar denetlenebilir.



Şekil 2.19 Normal anatomi ve engelli anatomisi için TMR yöntemi (Kuiken ve ark. , 2009)

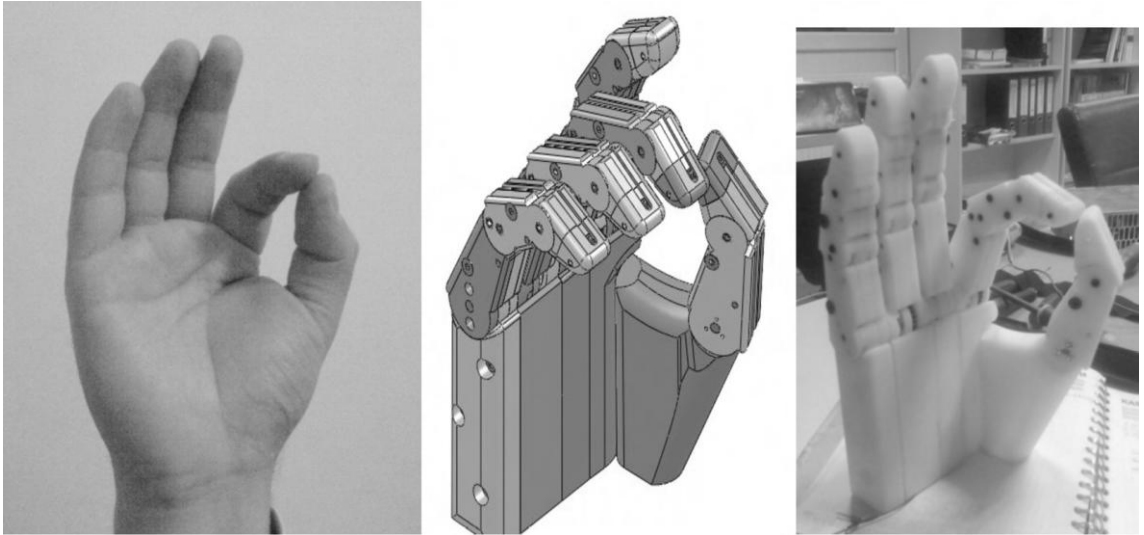
Bu yeni yöntem ile daha fazla serbestlik derecesine sahip protez kolların denetlenmesi mümkün olmaktadır. Yöntem, yaklaşık 8 saatlik bir cerrahi işlem, 10-14 saat hastanede kalma süreci, 3 aylık takip ve muayene süreci, özel montaj süreçleri kapsamakta olup yabancı hastalar için iletişimi kolaylaştıracak bir tercüman, hasta ile ilgilenecek bir fizyoterapistin yaklaşık 1 haftalık eğitimi gibi çeşitli giderlere sahiptir. Ayrıca sinirin yeni kas içinde büyümesi yaklaşık 1-3 yıl aldığı için ameliyat sonrasında zaman, sabır ve dayanıklılık gerektirmektedir (OttoBock, 2011). Şekil 2.20, deneysel ve kişiye özel tasarlanmış bir kolun TMR yöntemi ile kullanımını göstermektedir.



Şekil 2.20 Protez kolun TMR yöntemi ile kullanım örneği

TMR yöntemi, denetim için daha fazla kaynak sağlamaktadır. Bu cerrahi işlemi geçirmiş olan hastalarda kol, el bileği ve parmakları içeren 16 farklı harekete ait EMG işaretlerini kaydetmek mümkün olmaktadır. Farklı hareketlerin uygulanması sırasında elde edilen EMG işaretlerinden oluşturulan örüntüler bilgisayarda yaratılan bir sınıflayıcı tarafından kullanılır. Bu sınıflayıcı EMG işaretine karşılık gelen hareketi deşifre eder. Veri analizleri, istenen hareketleri ortalama %95 oranında sınıflandırma doğruluğu ile sınıflandırabildiğini ortaya koymaktadır (Kuiken ve ark. , 2009).

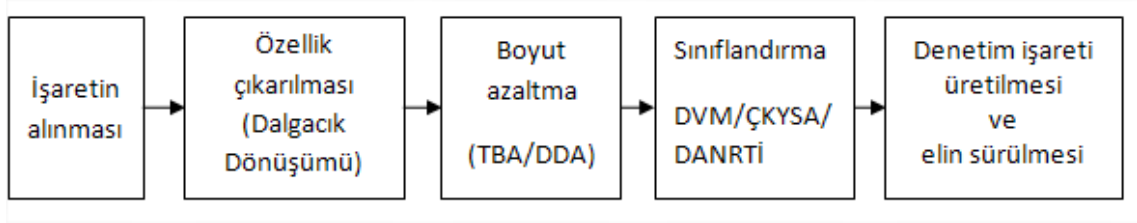
Protez elin denetlenmesinde kullanılan bir çalışma da elin denetlenmesinde ses komutlarını kullanmaktadır (Asyalı ve ark. , 2010). Tasarlanan özel bir ses tanıma devresi ile yine tasarlanan 3 eklemlili insansı parmaklara sahip bir protez el (Şekil 2.21) denetlenmektedir. “Tutmak”, “ bırakmak” gibi ses komutları özel ses tanıma devresi tarafından işlenir ve DC motorlara uygun sürücü işaretlerini gönderir. Elektronik denetleyici parmaklarda bulunan tutma güç algılayıcı işaretleri istenilen düzeye ulaştığı zaman sürücü işaretlerine ek olarak tutma güç algılayıcı işaretlerini ve ve motor sürücüsü kesme işaretlerini de göndermektedir.



Şekil 2.21 Ses denetimli protez el

3. MATERYAL VE YÖNTEMLER

Bu bölümde EMG verilerinden özellik çıkarılması, anlamlandırılması ve gerçek zamanlı olarak sınıflandırılması için geliştirilen algoritmalar açıklanacaktır. Kullanılan ana yapının öbek çizgesi Şekil 3.1’de verilmiştir.



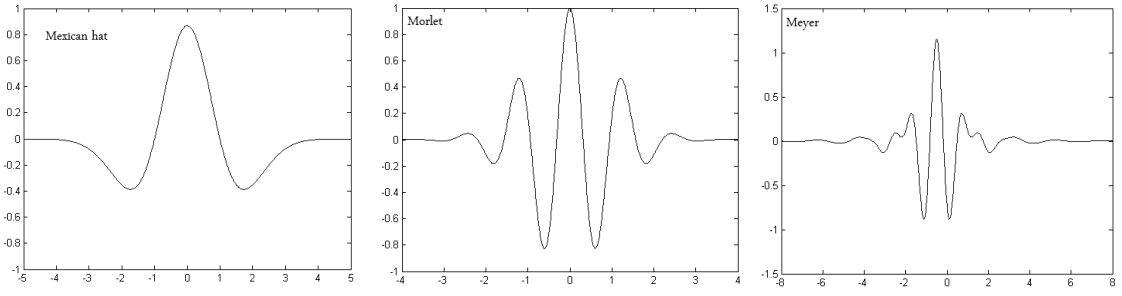
Şekil 3.1 Kullanılan sistemin öbek çizgesi

3.1. DALGACIK DÖNÜŞÜMÜ

EMG işaretinin anlamlandırılarak için gerekli verilerin elde edilmesinde kullanılan yöntemlerden biri olarak bu tez çalışmasında dalgacık dönüşüm kullanılmıştır.

Dalgacık dönüşümü değişken boyutlu pencereleme tekniği kullanan çok çözünürlüklü bir çözümleme yöntemidir. Böylelikle yöntem büyük işaretlerin yöresel analizine olanak sağlamaktadır. Yöntem Δt ve Δf çözünürlüğünün zaman – sıklık düzleminde değiştiğini kabul eder (Rioul ve Vetterli, 1991). Dalgacık dönüşümü sürekli ve ayırık dalgacık dönüşümü olmak üzere ikiye ayrılır.

Fourier analizi bir işareti, farklı sıklıklardaki sinüs dalgalarının birleşimi olarak çözümlerken, dalgacık dönüşümü kullanılan taban işlevinin ölçeklenmiş ve kaydırılmış birleşimleri olarak çözümler. Dalgacık dönüşümünde kullanılan taban işlevleri, zamanda yerel olarak tanımlı olup, sıklık düzleminde tek bir sıklık değeri ile değil bir sıklık bölgesi ile temsil edilen işlevlerdir. Taban işlevi Denklem (3.1) biçiminde olup ortalama değeri sıfır olan sınırlı süreli bir dalga biçimidir (Şekil 3.2) .



Şekil 3.2 Dalgacık taban işlevi örnekleri

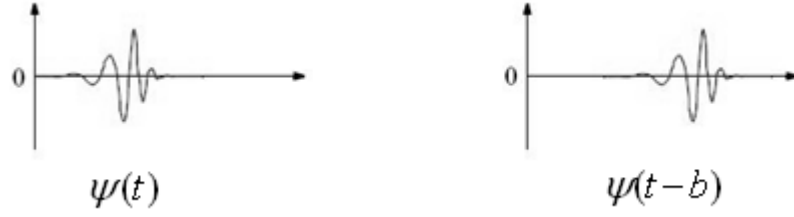
$$\Psi_{ab}(t) = \frac{1}{\sqrt{|a|}} \Psi\left(\frac{t-b}{a}\right) \quad (3.1)$$

Bu denklemde a ölçeklendirme, b kaydırma(zamanda öteleme) faktörleridir. $\frac{1}{\sqrt{|a|}}$ ise genlik normalizasyonu için kullanılır. Bu durumda $x(t)$ integrali alınabilir bir zaman işlevi olmak üzere Sürekli Dalgacık Dönüşümü (SDD), Denklem (3.2) de verilmiştir.

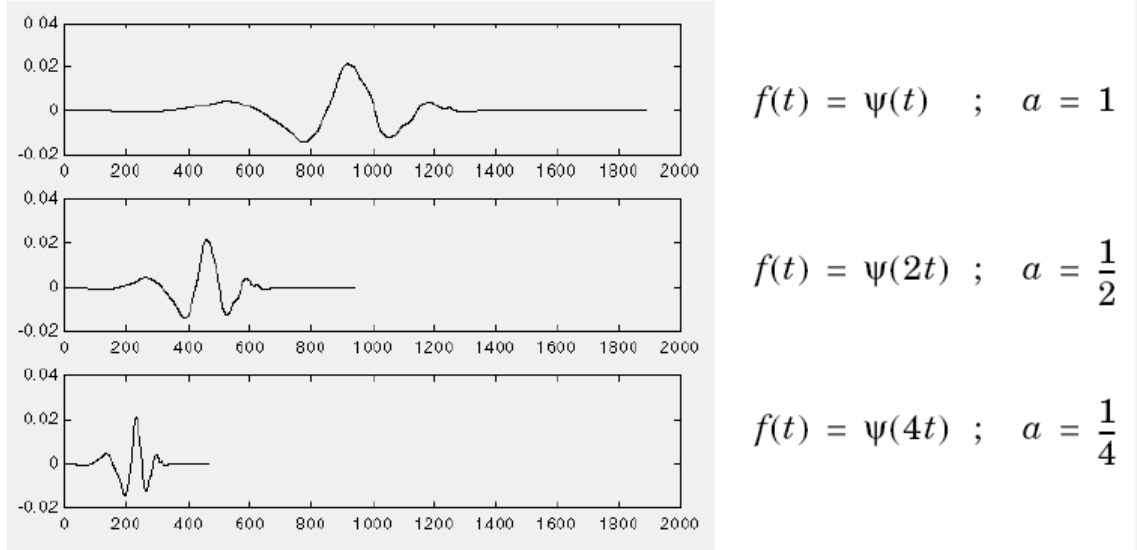
$$SSD_x(a, b) = \int_{-\infty}^{\infty} x(t) \cdot \Psi^*(\text{zaman, kayma, ölçek}) dt \quad (3.2)$$

$$SSD_x(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} x(t) \cdot \Psi^*\left(\frac{t-b}{a}\right) dt$$

Bu ifadede $a, b \in \mathbb{R}$, $a \neq 0$, ve $*$ karmaşık eşleniği göstermektedir. Eğer b değiştirilirse işaret zamanda hareket eder (Şekil 3.3), $a > 1$ için işlev kasılır veya daralır, $a < 1$ için ise işlev genişler veya açılır (Şekil 3.4). Çok büyük ölçekler global çok küçük ölçekler ise ayrıntılı görüntü anlamına gelmektedir. Bu temel dalgacık tüm süzgeç darbe tepkeleri için kullanılır. Yani ölçek artırıldıkça süzgeç darbe tepkisi zamandan yayılır.



Şekil 3.3 Dalgacık işaretinin zamanda kaydırılması(Misiti ve ark. , 2002)

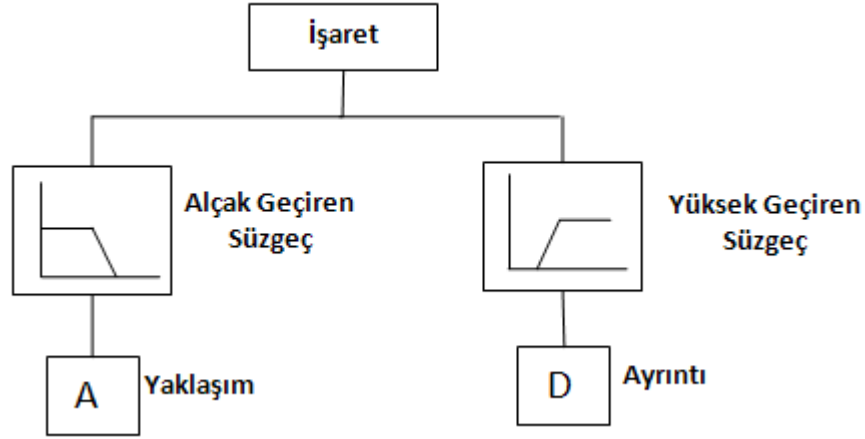


Şekil 3.4 Dalgacık işaretinin ölçeklendirilmesi (Misiti ve ark. , 2002)

Ayrık Dalgacık Dönüşümü (ADD), Dalgacık Dönüşümünün özel bir durumudur ve işaretin, zaman ve sıklık düzleminde verimli olarak işlenmesini sağlar. Ayrık zaman işaretlerini analiz etmek için a ve b 'nin özel değerleri alınır (Mallat, 1989). Eğer $a = 2^j$ ve $b = n.2^j$ (j ve n tamsayı) olarak alınırsa dalgacık işlevi Denklem(3.3) biçiminde olur:

$$\psi_{j,n}(t) = 2^{-j/2} \psi(2^{-j/2}t - n) \quad (3.3)$$

İşaret işleme bakış açısıyla dalgacık işlevleri sıklık düzleminde süzgeç karakteristiğine karşılık gelirler. Bu amaçla ADD, zaman düzleminde işaretleri alçak ve yüksek geçiren süzgeçlerden geçirerek sıklık bantlarına ayırıştırır. Böylece alçak sıklık işaretin önemli kısmını yüksek sıklık ise ayrıntı kısmını içerir (Şekil 3.5).



Şekil 3.5 Süzgeçleme ile ayrık dalgacık dönüşümü

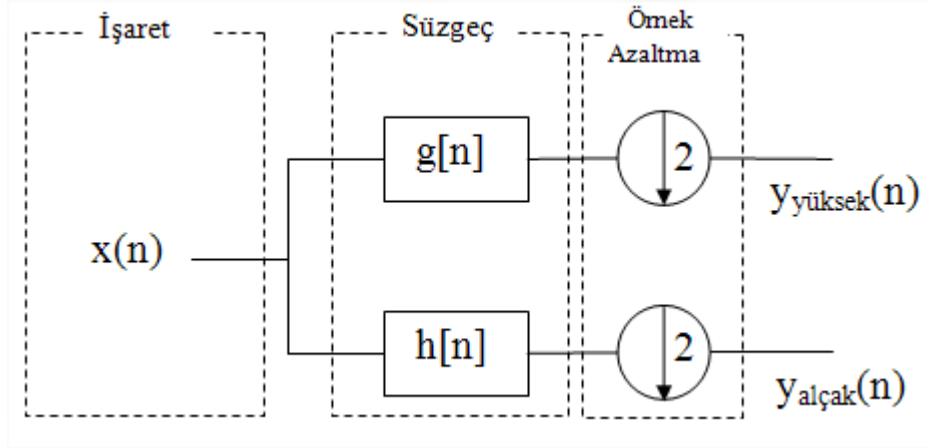
N örnekli bir S işaretine bu işlem uygulandığında N adet yaklaşım ve N adet detay olmak üzere toplam 2N örnek elde edilir. Bu sorunun üstesinden gelmek için örnek azaltımı (down sampling) yapılarak örnek sayısı her süzgeçleme işleminde yarıya indirilerek ayrık dalgacık dönüşümü işlemi gerçekleştirilir.

N örnekli $x(n)$ işareti için Şekil 3.6’ daki gibi $g(n)$ yüksek geçiren süzgeç ve $h(n)$ alçak geçiren süzgeç işlevlerinden geçirilirse katlama ve örnek azaltma işlemi matematiksel olarak Denklem (3.4) ‘de verildiği biçimde ifade edilebilir (Goswami ve Chan, 1999; Mallat, 1999).

$$\begin{aligned}
 y_{yüksek}(n) &= \sum_k x(k)g(2n - k) \\
 y_{alçak}(n) &= \sum_k x(k)h(2n - k) \quad n = 0,1,..(N - 1)
 \end{aligned}
 \tag{ 3.4 }$$

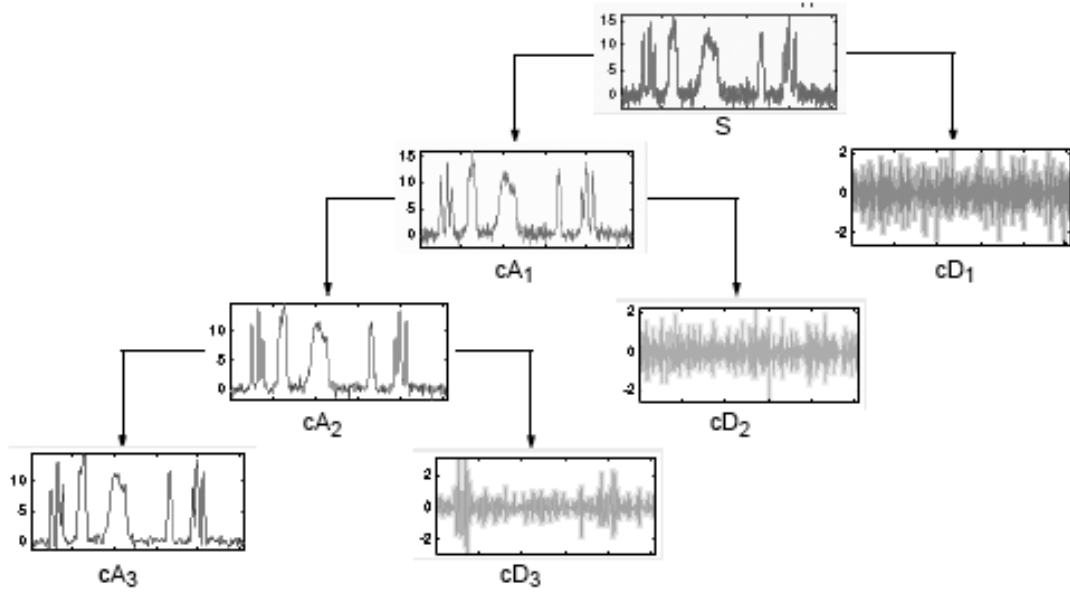
Orijinal işaret ise Denklem (3.5) deki biçimde yeniden oluşturulabilir.

$$x(k) = \sum_n \left(y_{yüksek}(n).g(2n - k) \right) + (y_{alçak}(n).h(2n - k))
 \tag{ 3.5 }$$



Şekil 3.6 Süzgeçleme ve örnek azaltma

Bu biçimde bir işaret çok sayıda düşük çözünürlüklü bileşene bölünerek ayrışma tekrarlanabilir. Buna Dalgacık Ayrışma Ağacı adı verilir .



Şekil 3.7 Bir işarete ait 3. Dereceden Dalgacık Ayrışma Ağacı (Misiti ve ark. , 2002)

3.2. BOYUT AZALTMA ALGORİTMALARI

3.2.1. Temel Bileşenler Analizi

Temel Bileşenler Analizi (TBA)'nin ana fikri, veri setindeki değişimi mümkün olduğunca koruyarak büyük sayıdaki veri boyutunu azaltmaktır. TBA ile yüksek varyanslı değerler korunarak düşük varyanslı değerler atılarak elde edilen sıralanmış temel bileşenler kullanır ve giriş vektörünün özelliklerinin sayısı (boyutu) azaltılmış yeni bir veri seti oluşturulur (Duda ve ark. , 2001; Jolliffe, 2002).

Sınıflandırma problemlerinde çok boyutlu uzaydaki bir giriş veri setinin daha düşük boyutlu bir uzaya haritalanması ve dolayısıyla boyutunun azaltılması başarımı etkileyen önemli bir adımdır. TBA yöntemi işlem adımları (Smith, 2002; Shlens, 2005):

1. Tüm özellikleri içeren $m \times n$ boyutlu bir matris ile ifade edilen yüksek boyutlu $\mathcal{P}$ veri kümesi Denklem(3.13) ile verilmiştir. Bu veri kümesi m adet örnek n adet özellik (boyut) içerir.

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} & \mathcal{P}_{13} & \dots & \mathcal{P}_{1n} \\ \mathcal{P}_{21} & \mathcal{P}_{22} & \mathcal{P}_{23} & \dots & \mathcal{P}_{2n} \\ \mathcal{P}_{31} & \mathcal{P}_{32} & \mathcal{P}_{33} & \dots & \mathcal{P}_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ \mathcal{P}_{m1} & \mathcal{P}_{m2} & \mathcal{P}_{m3} & \dots & \mathcal{P}_{mn} \end{bmatrix} \quad (3.6)$$

2. Verinin her bir boyutundan o boyutun ortalaması çıkarılır. Böylece ortalaması sıfır olan bir veri seti Denklem (3.14)'de verildiği biçimde üretilir.

$$\bar{\mathcal{P}} = \begin{bmatrix} \mathcal{P}_{11} - \overline{\mathcal{P}_1} & \mathcal{P}_{12} - \overline{\mathcal{P}_2} & \mathcal{P}_{13} - \overline{\mathcal{P}_3} & \dots & \mathcal{P}_{1n} - \overline{\mathcal{P}_n} \\ \mathcal{P}_{21} - \overline{\mathcal{P}_1} & \mathcal{P}_{22} - \overline{\mathcal{P}_2} & \mathcal{P}_{23} - \overline{\mathcal{P}_3} & \dots & \mathcal{P}_{2n} - \overline{\mathcal{P}_n} \\ \mathcal{P}_{31} - \overline{\mathcal{P}_1} & \mathcal{P}_{32} - \overline{\mathcal{P}_2} & \mathcal{P}_{33} - \overline{\mathcal{P}_3} & \dots & \mathcal{P}_{3n} - \overline{\mathcal{P}_n} \\ \dots & \dots & \dots & \dots & \dots \\ \mathcal{P}_{m1} - \overline{\mathcal{P}_1} & \mathcal{P}_{m2} - \overline{\mathcal{P}_2} & \mathcal{P}_{m3} - \overline{\mathcal{P}_3} & \dots & \mathcal{P}_{mn} - \overline{\mathcal{P}_n} \end{bmatrix} \quad (3.7)$$

3. Üretilen bu veri seti kullanılarak kovaryans matrisi hesaplanır. Kovaryans hesaplanarak, veri kümesinde boyutlar (özellikler) arasındaki ilişki

belirlenmektedir. Giriş veri kümesi n özellik içerdiği için kovaryans matris $n \times n$ boyutlu bir kare matristir ve sıfıra ortalananmış matris $\bar{\phi}$, kendi devriği $\bar{\phi}^T$ ile çarpılarak Denklem (3.15)'deki biçimde hesaplanmaktadır.

$$K = \frac{1}{m-1} (\bar{\phi}^T \cdot \bar{\phi}) \quad (3.8)$$

4. Kovaryans matrisinin Denklem(3.9)'u sağlayan özdeğerleri ve özvektörleri hesaplanır. λ özdeğer matrisi, K kovaryans matrisinin n tane özdeğerini içerirken, Γ özvektör matrisi de her sütunda bir öz vektör olmak üzere öz vektörleri içermektedir.

$$\Gamma \cdot K = \lambda \cdot K \quad (3.9)$$

5. Kovaryans matrisin özvektörleri birbirlerine dik olup Denklem(3.17) ile verilmektedir. Bu özvektörlerin her biri temel bileşen olarak adlandırılır.

$$\Gamma = \begin{bmatrix} \Gamma_{11} & \Gamma_{21} & \Gamma_{31} & \dots & \Gamma_{n1} \\ \Gamma_{12} & \Gamma_{22} & \Gamma_{32} & \dots & \Gamma_{n2} \\ \Gamma_{13} & \Gamma_{23} & \Gamma_{33} & \dots & \Gamma_{n3} \\ \dots & \dots & \dots & \dots & \dots \\ \Gamma_{1n} & \Gamma_{2n} & \Gamma_{3n} & \dots & \Gamma_{nn} \end{bmatrix} \quad (3.10)$$

6. Özdeğerler büyükten küçüğe sıralandığında en büyük özdeğere sahip özvektör veri setinin en temel bileşenidir. Bu sıra bileşenlerin önem sırasını verir. İsteğe bağlı olarak düşük öneme sahip bileşenler atılabilir. Bu veride bir takım kayıplara neden olacaktır. Fakat bileşenin değeri ne kadar küçükse bu kayıp o kadar küçük olacaktır. N boyutlu bir veri n adet özvektör ve özdeğere sahip olur. Eğer en önemli ilk p özvektör seçilirse yeni veri p boyutlu olacaktır.

7. Temel bileşen sayısı, seçilen p temel bileşen için Denklem(3.19) kullanılarak varyansın birikimli yüzdesi hesaplanarak belirlenebilir. Hesaplanan t_p değeri %70 ve %95 arasında ise bu bileşen sayısı seçilebilir (Martinez ve ark. , 2005). Seçilen temel bileşenleri kullanarak elde edilen özvektör matrisi Denklem(3.12) ile verilmiştir.

$$t_p = 100 \frac{\sum_{i=1}^z \lambda_i}{\sum_{j=1}^n \lambda_j} \quad (3.11)$$

$$\Gamma_p = \begin{bmatrix} \Gamma_{11} & \Gamma_{21} & \Gamma_{31} & \dots & \Gamma_{p1} \\ \Gamma_{12} & \Gamma_{22} & \Gamma_{32} & \dots & \Gamma_{p2} \\ \Gamma_{13} & \Gamma_{23} & \Gamma_{33} & \dots & \Gamma_{p3} \\ \dots & \dots & \dots & \dots & \dots \\ \Gamma_{1n} & \Gamma_{2n} & \Gamma_{3n} & \dots & \Gamma_{pn} \end{bmatrix} \quad (3.12)$$

8. *Öz_Vektör_Matrisi*, seçilmiş özvektör matrisinin (Γ_p) tümleyeni, *Ham_Veri_Seti*, ortalaması sıfır olan başlangıç veri seti ($\bar{\varphi}$) ve *Son_Veri_Seti*, boyutu azaltılmış veri seti olarak adlandırılırsa
 $\text{Son_Veri_Seti} = \text{Öz_Vektör_Matrisi} \times \text{Ham_Veri_Seti}$
 ile yeni veri seti elde edilebilir.

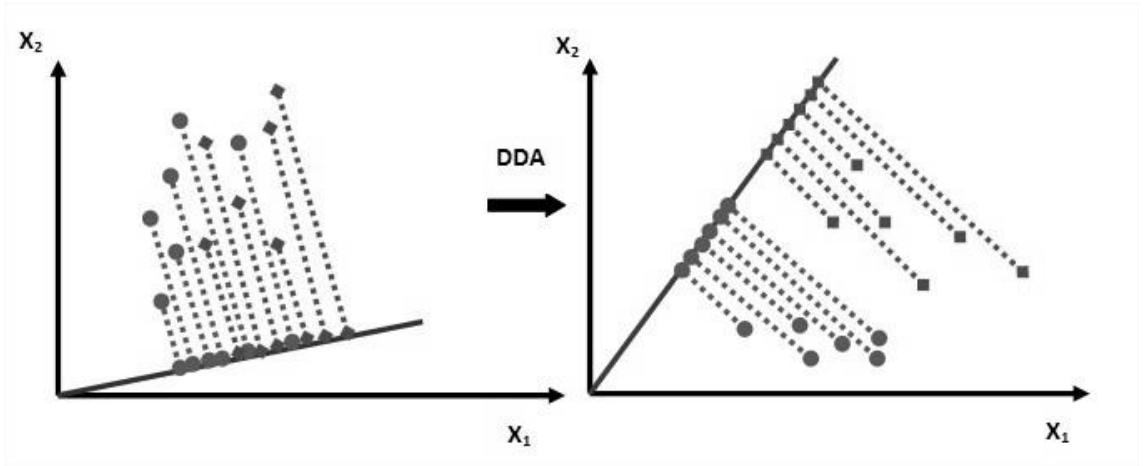
TBA işlemi sonucunda veri setinin tanımlı olduğu uzayda yayıldığı yön doğrultusunda taban vektörleri oluşturur.

3.2.2. Doğrusal Diskriminant Analizi

Fisher'ın doğrusal diskriminant analizi yöntemi sınıf bilgisini koruyarak veriyi daha düşük boyutlu bir uzaya haritalar. Sınıf ayrılabilirliğini büyütürken, sınıf içi

dağılımı küçülten bir doğrusal yapı sunan DDA bazen bir sınıflandırıcı olarak kullanıldığı gibi bu çalışmadaki gibi bir ön-işlem aracı olarak da kullanılabilir.

Şekil 3.24’ de bir boyutlu örneklerden oluşan w_1 sınıfına ait N_1 verileri $\{x^1, x^2, \dots, x^N\}$ ve w_2 sınıfına ait N_2 verilerinin farklı iki düzlem üzerine yansıtılmış biçimi görülmektedir. Olası bu düzlemlerin içinden sınıf ayrılabilirliğini en büyük yapan biri seçilebilir. Bu düzlemi bulmak için istatistiksel veri analiz tekniği olan Fisher’ın DDA’sı en uygun $y = w^T x$ ifadesini sağlayan bir w , yansıtma vektörü tanımlar (Belhumeur ve ark. , 1997; Cios ve ark. , 2007; Gutierrez-Osuna,).



Şekil 3.8 DDA kullanımında sınıf ayrılabilirliğinin gösterimi

İyi bir yansıtma vektörü bulmak için yansıtma ve ayrılabilirlik arasında bir ölçüm biçimi tanımlanır.

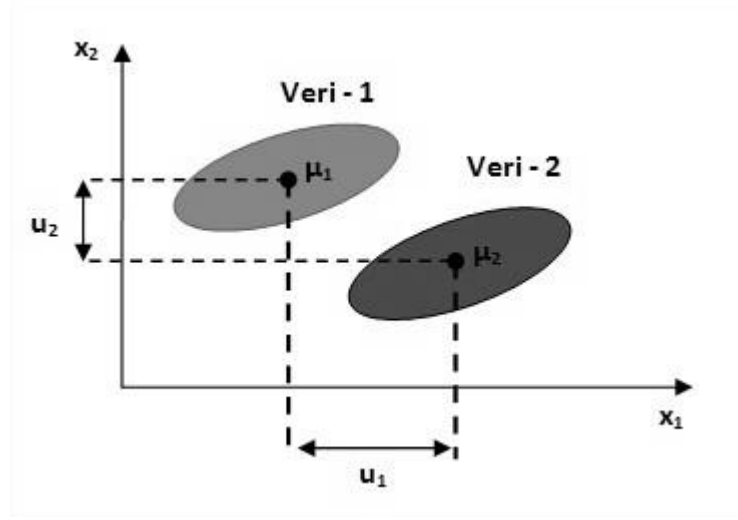
x ve özellik uzayındaki her bir sınıfın ortalama vektörü Denklem(3.13) ile tanımlandığı gibidir:

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x \quad \text{ve} \quad \tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in \omega_i} y = \frac{1}{N_i} \sum_{x \in \omega_i} W^T X = W^T \mu_i \quad (3.13)$$

Denklem (3.14)’de verildiği gibi yansıtılmış ortalamalar arasındaki uzaklık bir amaç fonksiyonu olarak seçilebilir.

$$J(w) = |\tilde{\mu}_1 - \tilde{\mu}_2| = |w^T (\mu_1 - \mu_2)| \quad (3.14)$$

Fakat sınıflar içindeki standart sapma hesaba katılmadığı için , yansıtılmış ortalamalar arasındaki mesafe iyi bir ölçüt değildir. Şekil 3.9'da görüldüğü gibi $u_1 > u_2$ olmasına rağmen küçük olan u_2 uzaklığı daha iyi bir sınıf ayrılabilirliği sağlar. Yani uzaklığın artması aynı zamanda sınıf ayrılabilirliğini artıran bir unsur değildir.



Şekil 3.9 Verilerin ortalamasının sınıf ayrılabilirliğindeki başarımı

Ortalamlar arasındaki fark işlevini en büyük yapmak için Fisher, Denklem (3.14) ile verilen amaç işlevinin sınıf içindeki dağılımın bir ölçütü tarafından normalize edilmesini önermiştir. Önerilen amaç işlevi Denklem(3.15) ile verilmiştir.

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{S}_1^2 + \tilde{S}_2^2} \quad (3.15)$$

Burada $\tilde{S}_1^2 + \tilde{S}_2^2$ miktarı , yansıtılmış örneklerin sınıf içindeki dağılımı olarak adlandırılır. Sınıf içi dağılım ise Denklem(3.16) ile tanımlanmıştır.

$$\tilde{S}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 \quad (3.16)$$

Fisher doğrusal ayırıcı, bu amaç fonksiyonunu maksimize eden, $w^T x$ biçimindeki bir doğrusal fonksiyon şeklinde tanımlanır. Bu sayede aynı sınıfa ait verilerin birbirine çok yakın olduğu, aynı zamanda farklı sınıflara ait verilerinde birbirlerinden mümkün olduğunca uzak olduğu bir yansıma uzayı aranabilir.

Denklem(3.17) yansıtılmamış uzayda her bir sınıfın standart sapmaları toplamını tanımlar ve bu standart sapmaların toplamı S_w , sınıf içi dağılım matrisi olarak adlandırılır (Martinez ve Kak, 2001; Cios ve ark. , 2007).

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T \quad S_1 + S_2 = S_w \quad (3.17)$$

Denklem(3.18) ve Denklem(3.19)'da verildiği gibi yansıtılmış y verisinin dağılımı da, x özellik uzayındaki dağılımın bir işlevi olarak ifade edilebilir.

$$\begin{aligned} \tilde{S}_i^2 &= \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{x \in \omega_i} (w^T x - w^T \mu_i)^2 \\ &= \sum_{x \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w = w^T S_i w \end{aligned} \quad (3.18)$$

$$\tilde{S}_1^2 + \tilde{S}_2^2 = w^T S_w w \quad (3.19)$$

Benzer biçimde Denklem(3.20)'de verildiği gibi, yansıtılmış ortalamalar arasındaki fark da orijinal özellik uzayındaki ortalamaların bir ifadesi biçiminde ifade edilebilir.

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (w^T \mu_1 - w^T \mu_2)^2 = w^T \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T}_{S_B} w = w^T S_B w \quad (3.20)$$

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \quad (3.21)$$

S_B Matrisi sınıflar arası dağılım matrisi olarak adlandırılır (Martinez ve Kak, 2001; Cios ve ark. , 2007). Sonuç olarak Denklem(3.15) ile verilen Fisher kriter S_w ve S_B cinsinden ifade edilebilir. Amaç bu kriteri en büyük yapmak olduğu için Denklem(3.23)'deki gibi türevi alınarak sifıra eşitlenir.

$$\begin{aligned} \frac{d}{dw} [J(w)] &= \frac{d}{dw} \left[\frac{w^T s_B w}{w^T s_w w} \right] = 0 \Rightarrow \\ \Rightarrow [w^T s_w w] \frac{d[w^T s_B w]}{dw} - [w^T s_B w] \frac{d[w^T s_w w]}{dw} &= 0 \Rightarrow \end{aligned} \quad (3.22)$$

$$\Rightarrow [w^T s_w w] 2s_B w - [w^T s_B w] 2s_w w = 0$$

Eşitliğin her iki yanını $w^T s_w w$ ya bölünürse Denklem(3.23) elde edilir;

$$\begin{aligned} \left[\frac{w^T s_w w}{w^T s_w w} \right] s_B w - \left[\frac{w^T s_B w}{w^T s_w w} \right] s_w w &= 0 \\ \Rightarrow s_B w - J s_w w = 0 \quad \Rightarrow s_w^{-1} s_B w - J w &= 0 \\ (s_w^{-1} s_B w - J w) & \end{aligned} \quad (3.23)$$

Sonuç olarak genelleştirilmiş özdeğer probleminin çözümü Denklem(3.24) ile verildiği biçimdedir (Cios ve ark. , 2007).

$$w^* = \arg \max_w \left\{ \frac{w^T s_B w}{w^T s_w w} \right\} = s_w^{-1} (\mu_1 - \mu_2) \quad (3.24)$$

3.3. KÜMELEME ALGORİTMALARI

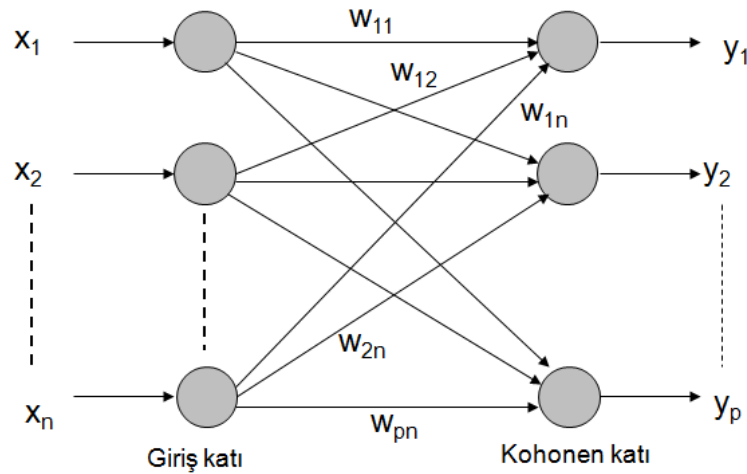
Bu bölümde tez de önerilen sınıflandırıcı yapı için ön-işlemci olarak kullanılan kümeleme algoritmaları tanıtılacaktır. Bu algoritmaların başarımları örneklerle açıklanacaktır.

3.3.1. Rekabetçi Öğrenme : Kohonen Öz-Düzenleyici Sistemler

Öz Düzenleme, giriş modeline göre çıkış verilmeksizin öğrenmenin gerçekleşmesi olarak tanımlanabilir. Kendiliğinden düzenlenen ağlar, giriş modelinin karakteristiğine dayalı kuvvet bağlantılarına göre veriyi yeniden düzenler. Bu işlemi sadece girdi değerleri arasındaki ilişkilere göre yaptığı için bu öğrenme tipinde öğretici yoktur. Bu ağ yapısı eğitim veri seti içindeki kümeleri bularak verileri kategoriler.

Rekabetçi öğrenme temeline dayanan bu öğrenme tipinde tüm ağıdaki nöronlar yarıştırılır ve onlardan sadece bir tanesi aktif olarak çıkışı belirler. Hiçbir zaman birden fazla nöron aktif olamaz. Yarışı kazanan çıkış nöronu “kazanan hepsini alır nöron” veya basitçe “kazanan nöron” olarak adlandırılır.

Şekil 3.10 “Kohonen Öz-Düzenleyici Ağ” yapısını göstermektedir. Bu ağ yapısında; Kohonen katındaki birimler birbirleriyle yarışır ve yarışmayı kazanan birim çıkışı “1” değerini alırken diğer çıkışlar “0” değerini alır (Nie ve Linkens, 1994).



Şekil 3.10. Kohonen Öz - Düzenleyici Ağ Yapısı

3.3.1.1. Kohonen Öz-Düzenleme Algoritması:

Verilmiş olan ağ yapısındaki n boyutlu giriş verisi (x), m adet olsun. Her bir giriş veri dizisi P olarak adlandırılırsa $P_1, P_2, P_3, \dots, P_m$ biçimli bir veri yapımız olmaktadır. Bu durumda algoritma işleyişi (Nie ve Linkens, 1994; Nie, 1995; Linkens ve Chen, 1999):

Adım 1: Başlangıç değerleri atanır.

Başlangıçta 1 adet birim olduğu ($j=1$) ve o birimin aktivasyonunun 1 olduğu ($Ns(j)=1$) kabul edilir. $l=1$ (veri indisi) ve yarıçap δ olarak belirlenir. Başlangıç ağırlıkları (w_l), veri dizisinin ilk elemanı (p_l) olarak seçilir.

$$w_1 = p_1 \quad (3.25)$$

Adım 2 : Bu durumda ilk olarak her bir p_1 örneğinin w_1 'ye olan uzaklıkları hesaplanır ve en minimum uzaklık seçilir.

$$D = \min \| w_1 - p_1 \| \quad (3.26)$$

Öklit uzaklığı olarak adlandırılan bu uzaklık $P=(p_1, p_2 \dots p_n)$ ve $Q=(q_1, q_2 \dots q_n)$ olduğu kabul edilirse (3.27) eşitliğindeki gibi tanımlanmaktadır.

$$\| p_n - q_n \| = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (3.27)$$

Başlangıç durumunda; Adım 1' de $l=1$ değerini aldığı için w 'nın p 'ye eşit, yani aradaki uzaklığın sıfır olduğu tek bir durum söz konusudur. Bu da Adım 3'teki Denklem(3.28) eşitliğinde görüleceği gibi o birimi kazanan birim yapacaktır.

Adım 3: Giriş verisinin hangi birime ait olduğu saptanır. Eğer;

$$D \leq \delta \quad \begin{array}{l} \text{j. birim kazanan birimdir. Yani} \\ \text{veri bu yarıçapın belirttiği küme içindedir.} \end{array} \quad (3.28)$$

$$D > \delta \quad \text{yeni bir birim yaratılır.}$$

j. birimin kazanan birim olması durumunda parametreler yeniden düzenlenir.

$$\phi(j)=1; \dots \text{Kazanan birim çıkışı 1, diğer çıkışlar 0 olur.} \quad (3.29)$$

$$w_j(t)=w_j(t-1)+ \alpha (p_1 -w_j(t-1)) \phi(j) : \text{Ağırlık güncellenir.} \quad (3.30)$$

$Ns(j)=Ns(j)+1$: Kazanan birimin aktivasyonu artırılır.

Öğrenme oranı azaltılmış olur.

(3.31)

$\alpha=\alpha_0/Ns(j)$: öğrenme oranıdır. $\alpha_0 \in [0,1]$

$l=l+1$ (veri dizisinin indisi)

(3.32)

Yeni birimin üretilmesi durumunda;

$j=j+1$ Yeni birim

(3.33)

$w(j)=p(l)$ Yeni birimin ağırlıkları o birimin

(3.34)

yaratılmasına neden olan veri olarak atanır.

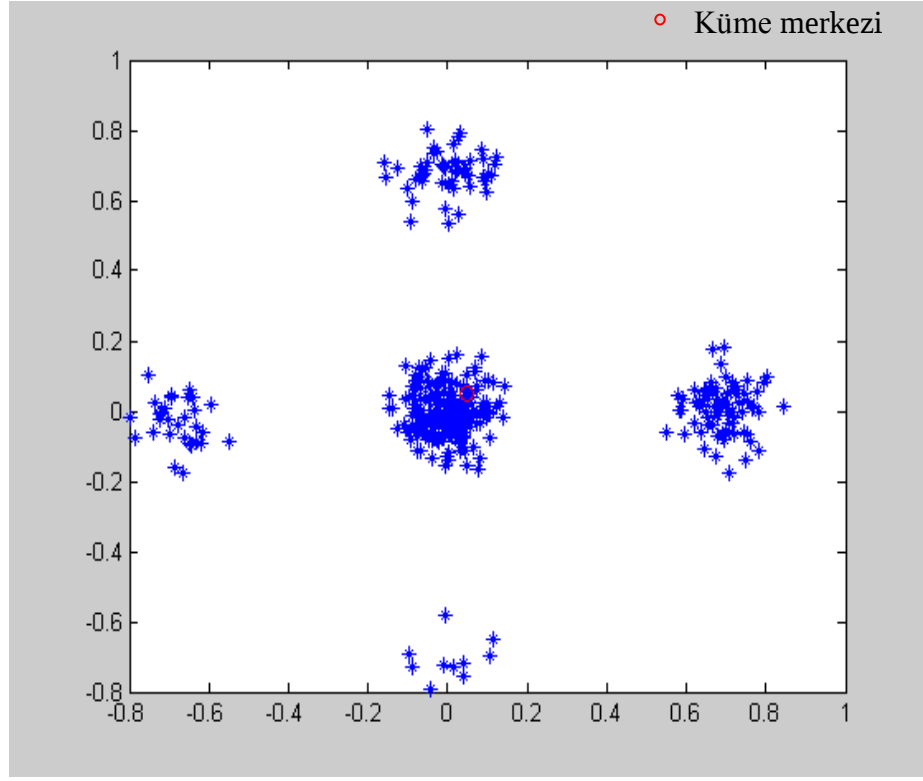
$Ns(j)=1$ Yeni birimin aktivasyonu

(3.35)

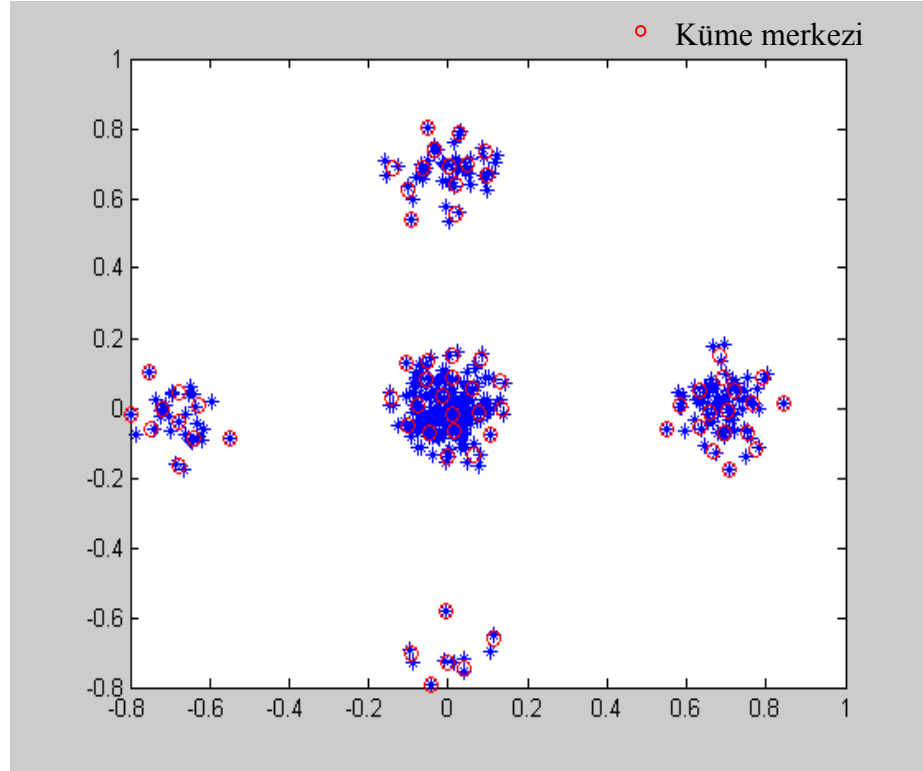
Adım 4: Eğer $l < m$ ise algoritma Adım 2'ye döner aksi durumda işlem sonlandırılır. Böylece başlangıçta belirlenen yarıçapa (δ) göre w ağırlıkları belirlenmiş olur. Bu ağırlık katsayıları aynı zamanda giriş verilerinin belirlenen yarıçapta içine alan kümenin merkez noktalarıdır.

3.3.1.2. Küme Kavramı:

Ağa girdi olarak verilen verilerin, başlangıçta belirlenen yarıçapın belirttiği bir kümeye ait olma ve olmama durumlarına göre sınıflandırıldığından ve bu sınıfları ise ağırlık çıkışındaki birimlerin temsil ettiğinden bahsedilmişti. Bu kümenin sayısını ise girilen yarıçap belirlemektedir. Yarıçap büyütüldükçe veriyi sınıflandıran küme sayısı en az bir olabilir. (Şekil 3.11) Yarıçapın küçültülmesi durumunda ise küme sayısı en fazla veri adeti kadar olur. Yani her bir veri bir kümeyi temsil eder (Şekil 3.12).

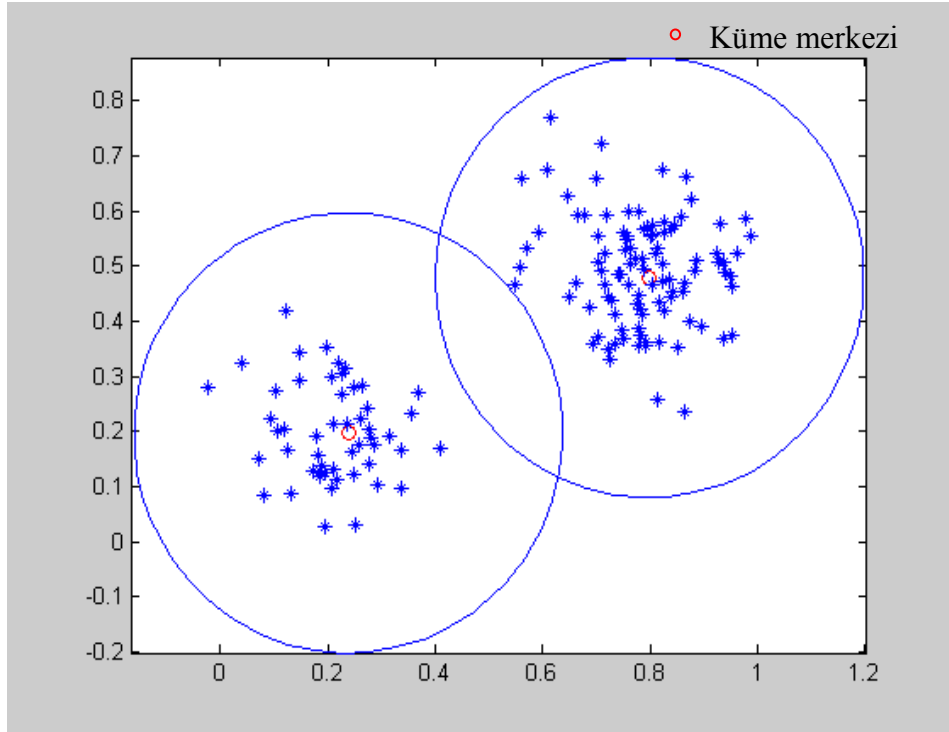


Şekil 3.11 Yarıçapın büyütülmesi durumu

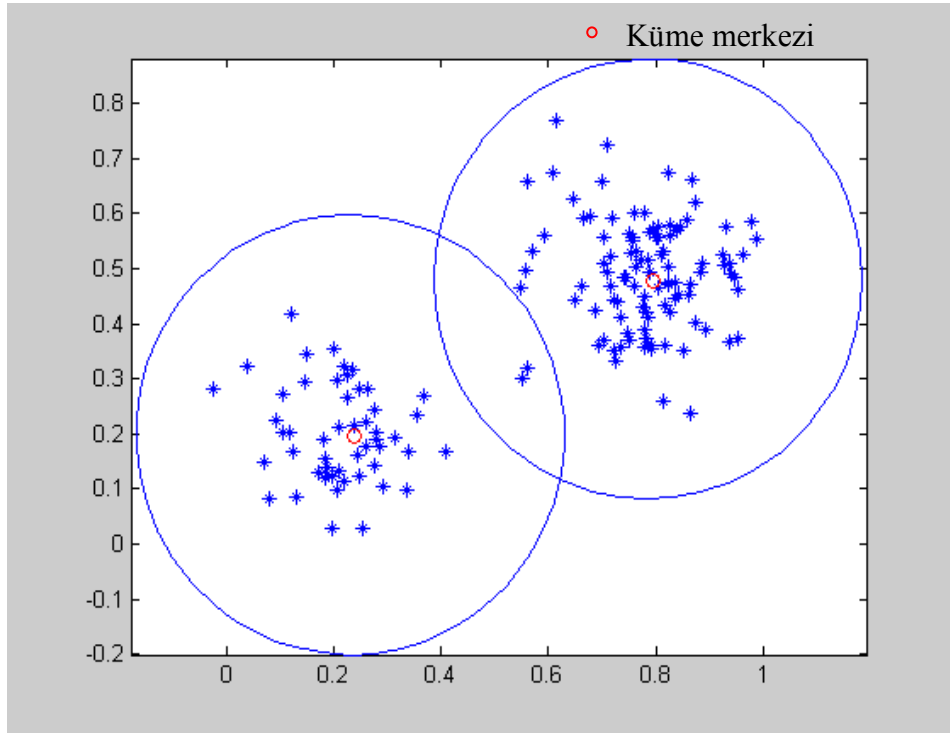


Şekil 3.12 Yarıçapın küçültülmesi durumu

Şekil 3.13 ve Şekil 3.14 yarıçapı $r=0.4$ olan iki farklı küme BCO algoritmasının başarımını göstermek üzere verilmiştir. Şekil 3.13’de gösterilen küme kesişim bölgesinde eleman bulundurmamaktadır ve algoritma küme merkezlerini saptayabilmektedir. Kesişim bölgesinde eleman bulunması durumunda ise Şekil 3.14’de görüldüğü gibi yine küme merkezleri bulunabilmektedir. Kesişim bölgesindeki elemanlar ise iki kümenin elemanı gibi görülse de aslında merkez olarak belirlenen noktaya en yakın mesafedeki kümeye aittir. Yani sadece bir kümenin elemanıdır.



Şekil 3.13 Kesişim alanında eleman bulundurmayan $r=0.4$ yarıçaplı daire ile belirtilen iki küme



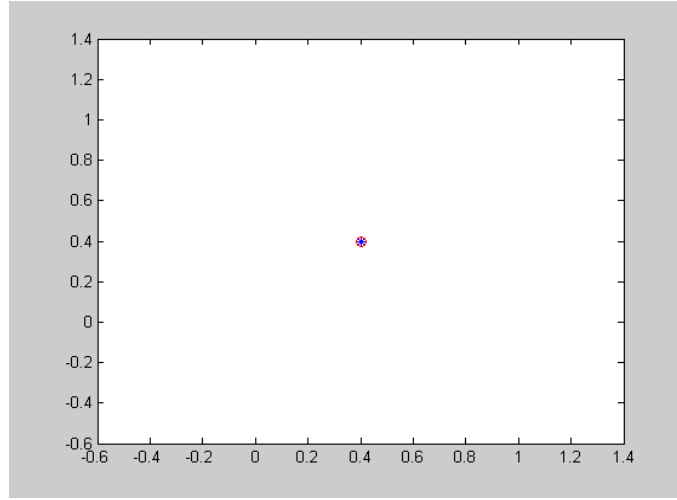
Şekil 3.14 Kesişim alanında eleman bulunduran $r = 0.4$ yarıçaplı daire ile belirtilen iki küme

3.3.1.3. Ağırlık Güncellenmesi Kavramı

Denklem(3.30) ile verilen ağırlık güncellenmesi kavramı; yeni verilen giriş verisine (p_j), bir önceki hesaplanan merkezden ($w_j(t-1)$) uzaklığı kadar belli bir öğrenme oranıyla yaklaşılmaması anlamını taşır .

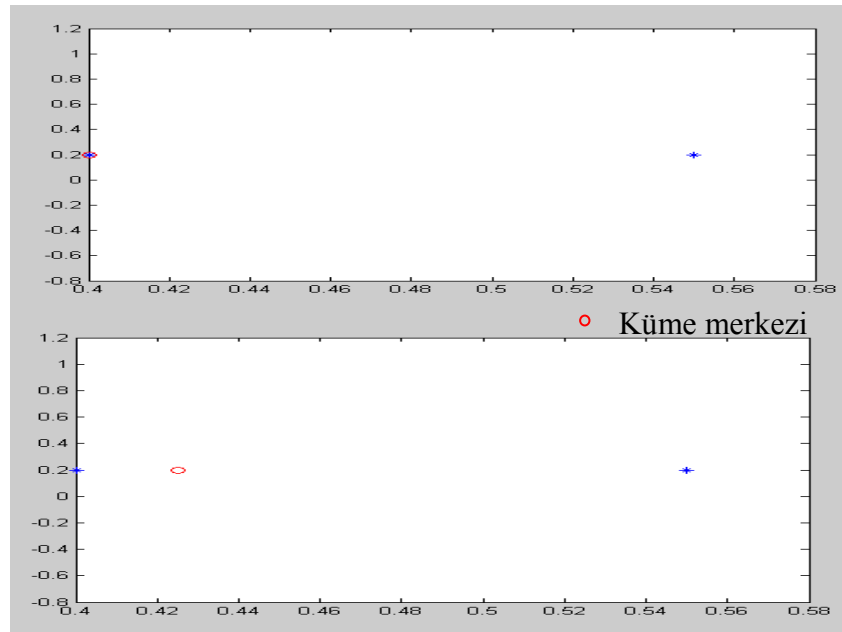
Durum 1: Şekil 3.15, tüm elemanları aynı olan 10 elemanlı bir veri kümesini göstermektedir. Bu veri örneği için $w_j(t-1)=p_1$ olduğu için algoritmadaki Denklem(3.30), Denklem (3.36) ile verilen biçime dönüşecektir. Bu da aynı veriye sıfır mesafe ile yaklaşmayı yani verinin kendisini merkez seçmeyi sağlayacaktır.

$$w_j(t)=w_j(t-1)+\underbrace{\alpha (p_1 - w_j(t-1))}_{0} \quad \Rightarrow \quad w_j(t)=w_j(t-1) \quad (3.36)$$



Şekil 3.15 On elemanlı ve tüm elemanları aynı bir veri kümesi

Durum 2: Şekil 3.16, $(0.4, 0.2)$ ve $(0.55, 0.2)$ verilerini içeren iki elemanlı kümenin, $\alpha=0.25$ öğrenme parametresi seçildiği zaman algoritmanın sonuçlarını göstermektedir. Şekil 3.16 görüldüğü gibi; $(0.4, 0.2)$ noktasındaki veri başlangıçta merkez seçilir. Bunun nedeni bu verinin algoritmaya verilen ilk girdi verisi olmasıdır. Ağ ikinci veriyi girdi aldığı anda fark (uzaklık) $(0.15, 0)$ olmakta ve Denklem(3.30) 'a göre yeni merkez $0.25 \cdot 0.15 = 0.035$ kadar yeni veriye yaklaşmaktadır. Bu durum Şekil 3.16'da $(0.435, 0.2)$ noktasında merkezlenmiş ağ çıktısında gözlenmektedir.



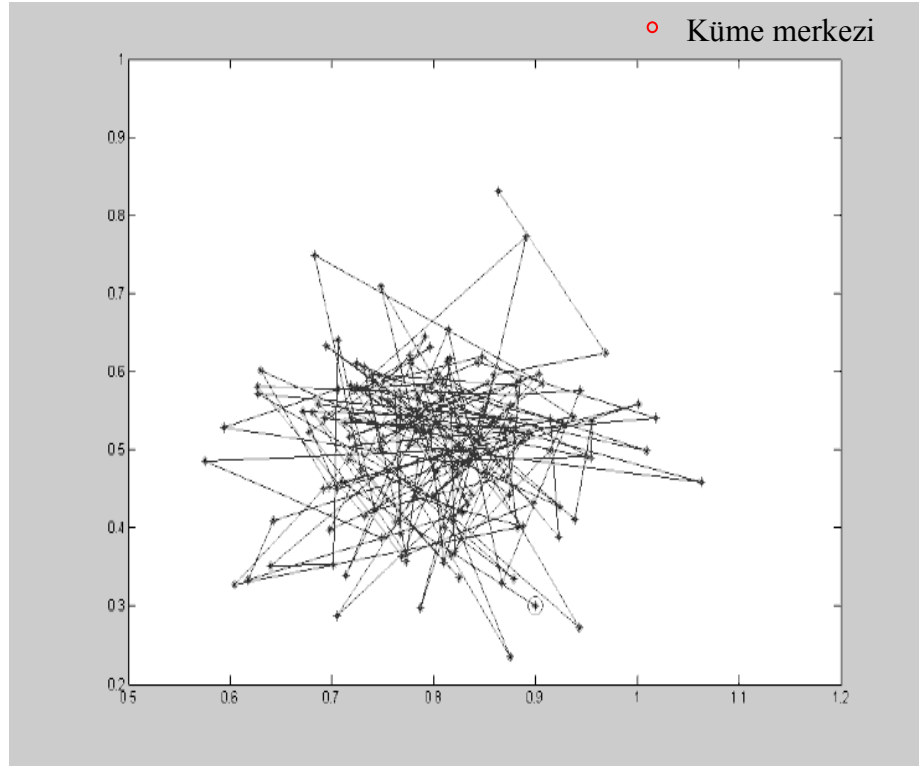
Şekil 3.16 İki elemanlı küme

Durum 3: Eğer ağırlık güncellenmesi için öğrenme parametresi $\alpha=1$ seçilirse küme merkezi en son girilen veri olarak belirlenir. Kendiliğinden öğrenme algoritmasının her yinelenen adımında yeni girilen veri ait olduğu kümenin merkezi olarak seçilir. Bu (3.37) ile de görülmektedir.

$$w_j(t) = w_j(t-1) + \alpha (p_1 - w_j(t-1)) \quad \Rightarrow \quad w_j(t) = p_1 \quad (3.37)$$

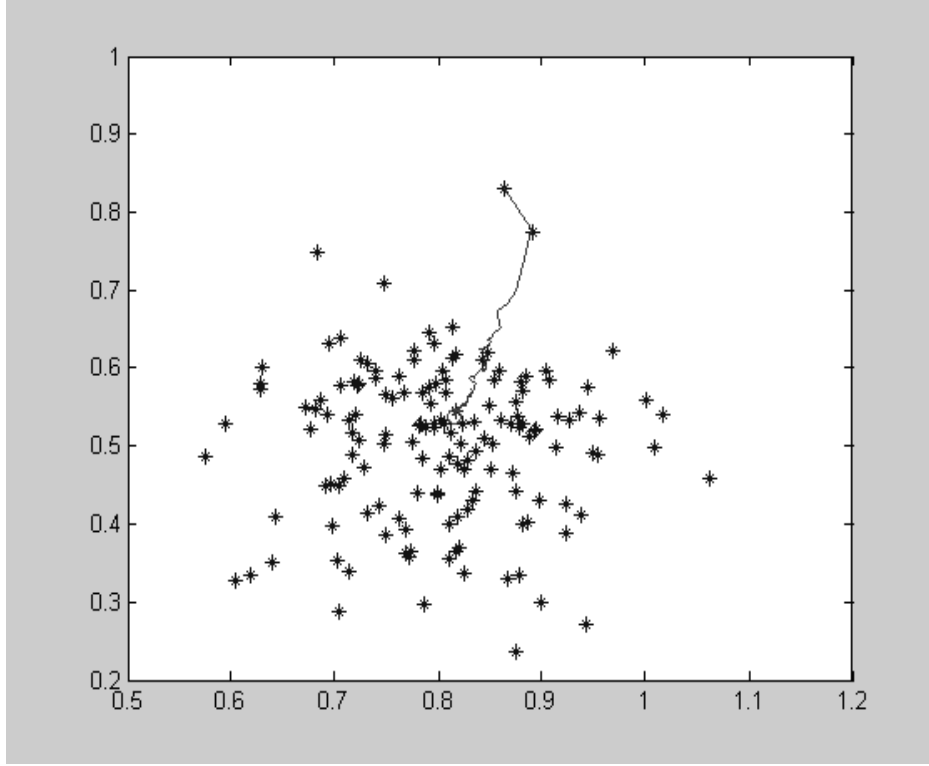
$\downarrow$ $\downarrow$
 1 1

Durum 4: Şekil 3.17 tek kümeli çoklu veri içeren bir veri örneği için her adımda merkezin değişimini göstermektedir. Görüleceği üzere, her adımda girdi olan veri merkez olarak seçildiği için merkez en son girilen veri seçilmiş olacaktır ve merkez doğru belirlenemeyecektir.

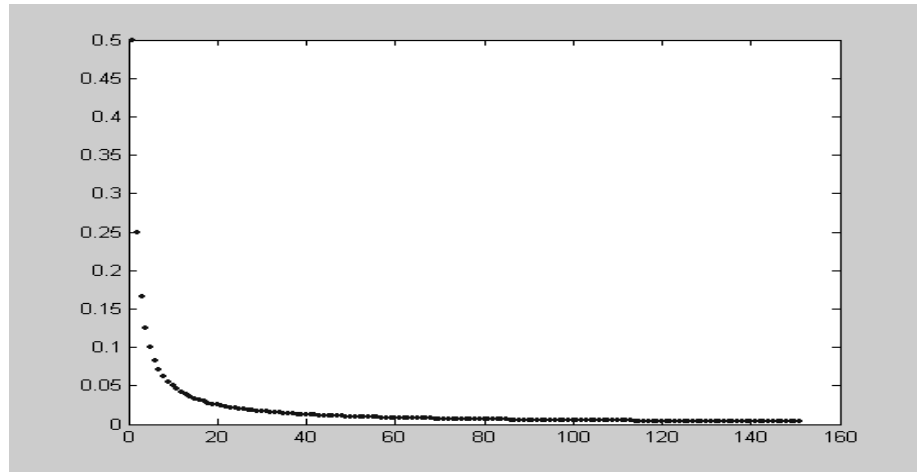


Şekil 3.17 Çoklu veri içeren tek kümeli veri

Durum 5: Tek kümeli çoklu veri için eğer ağırlık güncellenmesi yapılırsa Şekil 3.18’de verildiği gibi veri kümenin merkezini yakalayabilecektir. Bu durumda α öğrenme parametresinin değişimi Şekil 3.19 ‘da verilmiştir.



Şekil 3.18 Tek kümeli çoklu veri ağırlık güncellemeli olarak merkezinin bulunması

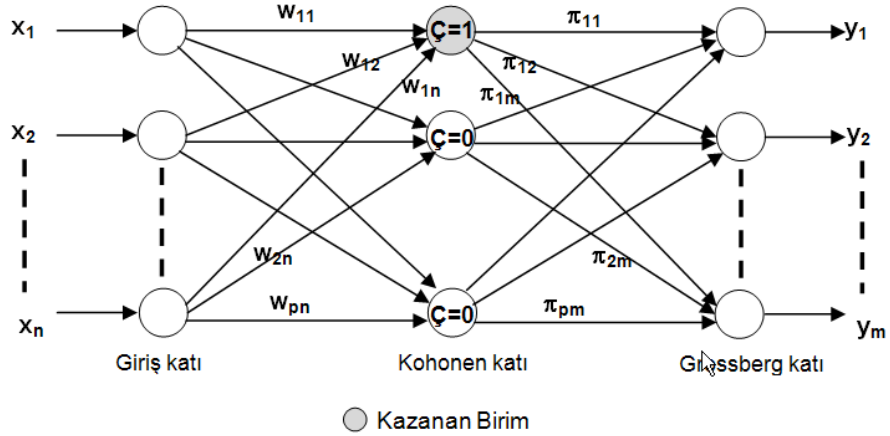


Şekil 3.19 Tek kümeli çoklu veri α öğrenme parametresinin değişimi

3.3.2. Karşı Yayılım Ağları

Karşı Yayılım Ağları (Counter Propagation Network, CPN) 1987 yılında Robert Hecht–Nielsen tarafından geriye yayılım ağlarına alternatif olarak geliştirilmiştir. Bu ağ, Şekil 3.10’da gösterilmiş olan Kohonen ağ yapısına Grossberg katı adı verilen katın eklenmesi ile oluşturulmuştur. Yapısı Şekil 3.20 ile verilmiş olan bu ağlar, uzun eğitim zamanlarına olanak tanınmayan durumlarda kullanılmaktadır. Bu ağ yapısındaki Grossberg katı, yapı olarak Kohonen katından çok farklı değildir. En belirgin farkı, Kohonen katının öğretmensiz, Grossberg katı ise öğretmenli eğitim yapmasıdır. Yani Grossberg katında ağa girdi verisinin yanı sıra istenilen çıktı verisi de verilir ve ağın bu yapıyı öğrenmesi beklenir.

Karşı Yayılım Ağları (KYA), Bulanık Mantık kuralları gibi sisteme ait kurallar üretmektedir. Bulanık Mantıktan farklı olarak KYA ‘da sadece bir kural aktif olabilmektedir. Bilindiği gibi Bulanık Mantık’ta birden çok kural farklı üyelik dereceleriyle aktif olabilmektedir.



Şekil 3.20 Karşı Yayılım Ağ yapısı

Grossberg katının öğrenme algoritması Kohonen katının öğrenme algoritması ile aynıdır. Yalnızca bu katın ağırlıkları güncellenirken Denklem(3.38) kullanılır. Bu eşitliğin ise yapı olarak Denklem(3.30) ‘dan çok farklı olmadığı görülür.

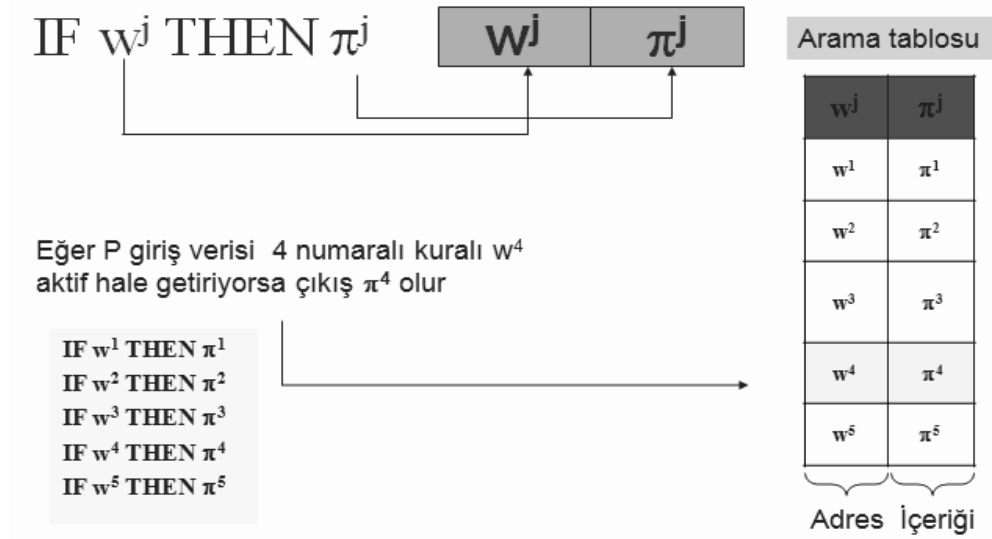
$$\pi_j(t) = \pi_j(t-1) + \beta (y_1 - \pi_j(t-1)) \phi(j) \quad (3.38)$$

$\pi_j(t)$: j. Katın şimdiki ağırlığı $\pi_j(t-1)$: j. Katın bir önceki ağırlığı
 β : Öğrenme parametresi $\phi(j)$: j. Katın çıkışı(1 veya 0 değerini alır.)
 $y(l)$: İstenilen çıkış

Karşı yayılım ağları, Grossberg ağının iyileştirme yeteneğiyle Kohonen ağının gruplaştırma yeteneğini bir araya getirir. Bu özelliği ile bu ağ yapısı Denklem(3.39)'daki gibi IF – THEN kuralı olarak ifade edilebilir (Chen ve Linkens, 2001; Chen, 1998)

$$\text{IF } w_j \text{ THEN } \pi_j \quad (3.39)$$

Sonuç olarak bu kural yapısı da bir arama çizelgesi (look-up table) biçimini sunmaktadır. Yani w 'lar adresleri π 'ler ise o adresin içeriğini gösterir. Bu yapı Şekil 3.21 ile gösterilmiştir.

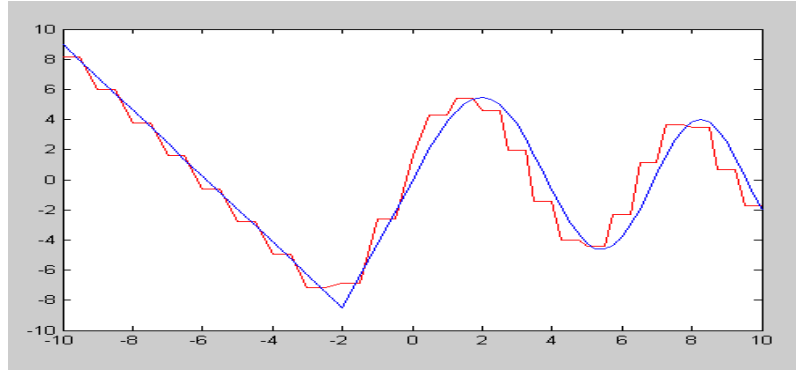


Şekil 3.21 Kural çizelgesi benzetimi

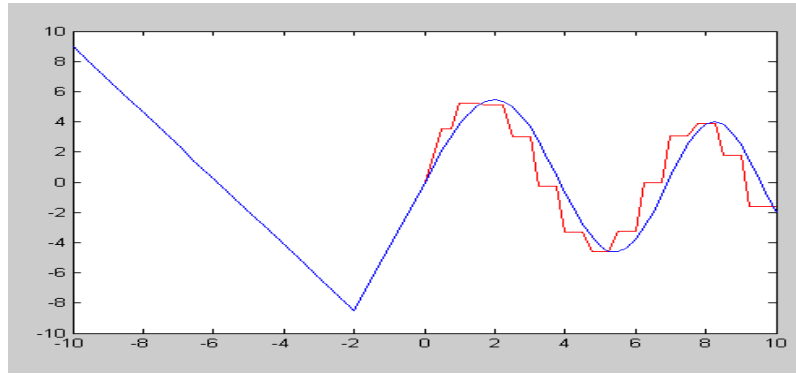
Örnek 3.1:

$$f(x) = \begin{cases} -2.186x - 12.864 & -10 \leq x < -2 \\ 4.246x & -2 \leq x < 0 \\ 10e^{-0.05x-0.5} \sin[(0.03x + 0.7)x] & 0 \leq x \leq 10 \end{cases}$$

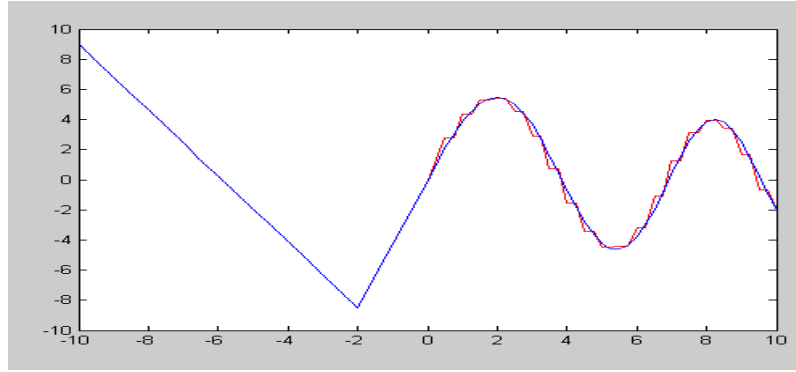
biçimli bir fonksiyona sahip tek giriş tek çıkışlı bir sistem ait 121 adet veri oluşturularak çeşitli yarıçaplarda ağ çıkış grafikleri ve kural sayıları verilmiştir.



Yarıçapın 0.7 seçilmesi durumu



Yarıçapın 0.5 seçilmesi durumu



Yarıçapın 0.3 seçilmesi durumu

- : Olması istenen çıkış - : Ağın ürettiği çıkış

Şekil 3.22 Örnek 1 için ağ çıkışları

Şekil 3.22 ile verilen ağ çıkışları incelendiğinde en iyi çıkışın (c)'de olduğu görülmektedir. Çünkü daha önce de bahsedildiği gibi yarıçap küçüldükçe küme sayısı dolayısıyla kural sayısı artmaktadır. Yarıçapın 0.7 olduğu (a)'da 24 kural, 0.5 olduğu (b)'de 34, 0.3 olduğu (c)'de ise 41 kural oluşmuştur.

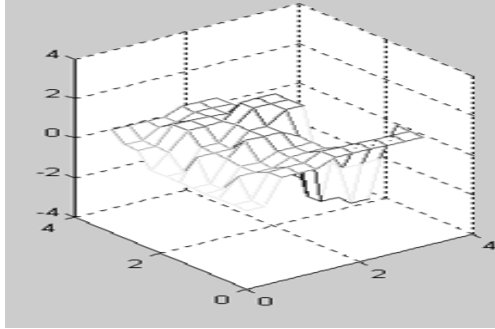
Örnek 3.2:

$$y = x_2 \sin(x_1) + x_1 \cos(x_2),$$

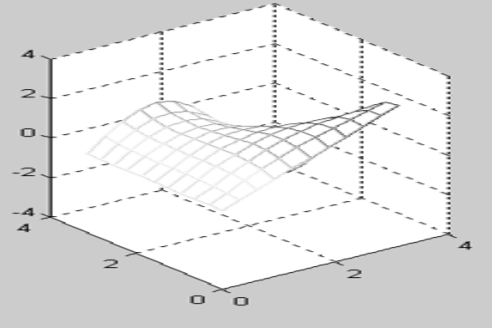
$$0 \leq x_1, x_2 \leq \pi$$

biçiminde bir fonksiyona sahip iki giriş bir çıkışlı bir sisteme ait 121 adet veri oluşturularak çeşitli yarıçaplarda ağ çıkış grafikleri ve kural sayıları verilmiştir.

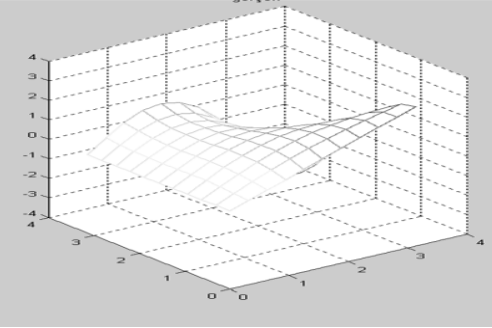
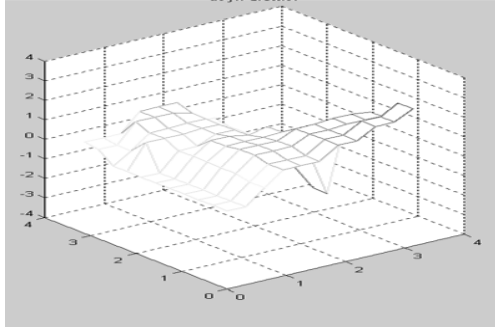
Ağ çıkışı



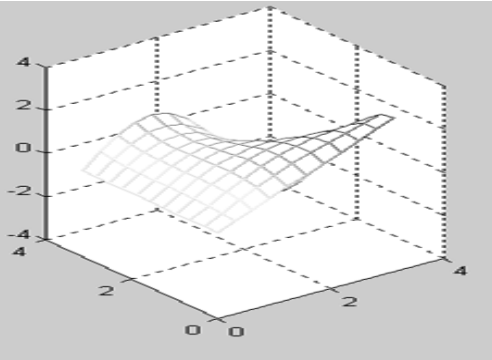
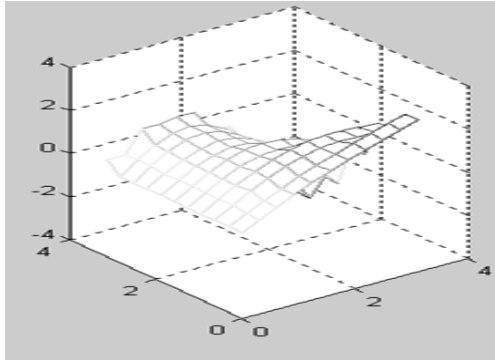
Gerçek çıkış



Yarıçapın 0.7 seçilmesi durumu



Yarıçapın 0.5 seçilmesi durumu



Yarıçapın 0.3 seçilmesi durumu

Şekil 3.23 Örnek 2 için ağ çıkışları

Şekil 3.23 ile verilen ağ çıkışları incelendiğinde en iyi çıkışın (c)'de olduğu görülmektedir. Örnek 3.1 ile verilen tek giriş tek çıkışlı sistemde olduğu gibi yarıçap küçüldükçe küme sayısının dolayısıyla kural sayısı artması ağın daha iyi öğrenmesi sağlamaktadır. Yarıçapın 0.7 olduğu (a) 'da 19 kural , 0.5 olduğu (b)'de 35, 0.32 olduğu (c)'de ise 61 kural oluşmuştur

3.3.3. Bulanık C-Ortalamları Algoritması

Bulanık C Ortalamaları algoritması, bulanık bölünmeli kümeleme tekniklerinden en iyi bilinen ve yaygın kullanılan yöntemdir. BCO algoritması 1973 yılında Dunn tarafından ortaya atılmış ve 1981'de Bezdek tarafından geliştirilmiştir (Frank ve ark. , 2000). Bulanık C-Ortalamları algoritması da amaç fonksiyonu temelli bir yöntemdir. BCO yöntemi, nesnelerin iki veya daha fazla kümeye ait olabilmesine izin verir. Bulanık mantık prensibi gereği her veri, kümelerin her birine $[0,1]$ arasında değişen birer üyelik değeri ile aittir. Bir verinin tüm sınıflara olan üyelik değerleri toplamı "1" olmalıdır. Nesne hangi küme merkezine yakın ise o kümeye ait olma üyeliği diğer kümelere ait olma üyeliğinden daha büyük olacaktır. Amaç fonksiyonun belirlenen minimum ilerleme değerine yakınsaklaşmasıyla kümeleme işlemi tamamlanır. Algoritma, en küçük kareler yönteminin genellemesi olan Denklem(3.40) ile verilen amaç fonksiyonunu öteleyerek minimize etmek için çalışır (Babuska, 1998; Frank ve ark. , 2000; Wolkenhaue,).

$$J(Z;U,V) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m \|z_k - v_i\|^2 \quad (3.40)$$

$$U = [\mu_{ik}] \quad (3.41)$$

Küme merkez vektörü V

$$V = [V_1 \ V_2 \dots V_c] \quad (3.42)$$

Z bulanık bölümlleme matrisi olup küme merkezlerine olan mesafesi, karesel iç çarpım uzaklık normu Denklem(3.43) ile hesaplanabilir. Kümenin biçimi A matrisinin seçimiyle karar verilen bir durumdur. Genel seçim, A'nın birim matris ($A=I$) olmasıdır ki bu standart Öklit normunu oluşturur.

$$D_{ik}^2 = \|Z_k - V_i\|^2 = (Z_k - V_i)^T A (Z_k - V_i) \quad (3.43)$$

$$m \frac{1}{2} \in [1, \infty) \quad (3.44)$$

m, sonuçta oluşan kümenin bulanıklığını hesaplayan ağırlıklandırma üstelidir. Denklem(3.40) daki benzersizlik ölçümü, küme merkezi (V_i) ile her bir veri noktası (Z_k) arasındaki karesel mesafedir. Bu mesafe, noktanın üyelik derecesini gücü (μ_{ik})^m ile ağırlıklandırılır. Bu durumda Denklem(3.40) işlevi, Z_k 'nin V_i 'den toplam değişimisinin (varyansının) ölçümü olarak tanımlanmaktadır.

3.3.3.1. BCO Algoritması

Denklem(3.40) verilen C ortlamaları işlevinin minumumlaştırılması, çeşitli mevcut yöntemler kullanılarak çözülebilen doğrusal olmayan bir optimizasyon problemini temsil eder. Bu yöntemlerden en popüler ise BCO Algoritması olarak bilinen yöntemdir.

Bu algoritma aşağıdaki biçimdedir (Babuska, 1998; Sun ve ark. , 2004).

Adım 1: N adet verinin oluşturduğu set Z olmak üzere, küme sayısı ($1 < c < N$), ağırlıklandırma üsteli ($m > 1$), sonlandırma toleransı ($\varepsilon > 0$), norm oluşturma matrisi ($A=I$) seçilir.

Başlangıç üyelik matrisi U, rastgele olarak 0-1 arasında değerlerle sınırlı olarak oluşturulur.

$$\sum_{i=1}^c u_{ij} = 1 \quad (c: \text{küme sayısı}, j=1 \dots N) \quad (3.45)$$

Adım 2: Bulanık kümeleme merkezleri V_i hesaplanır.

$$V_i = \frac{\sum_{j=1}^n U_{ij}^m Z_j}{\sum_{j=1}^n U_{ij}^m}, \quad i = 1 \dots c \quad (3.46)$$

Adım 3: Veri setinin küme merkez merkezlerine olan mesafesi hesaplanır.

$$D_{ij}^2 = \|Z_j - V_i\|^2 = (Z_j - V_i)^T (Z_j - V_i) \quad (3.47)$$

$$1 \leq i \leq c \quad 1 \leq j \leq N$$

Adım 4: BCO amaç işlevi $J < \varepsilon$ olduğu sürece hesaplanır.

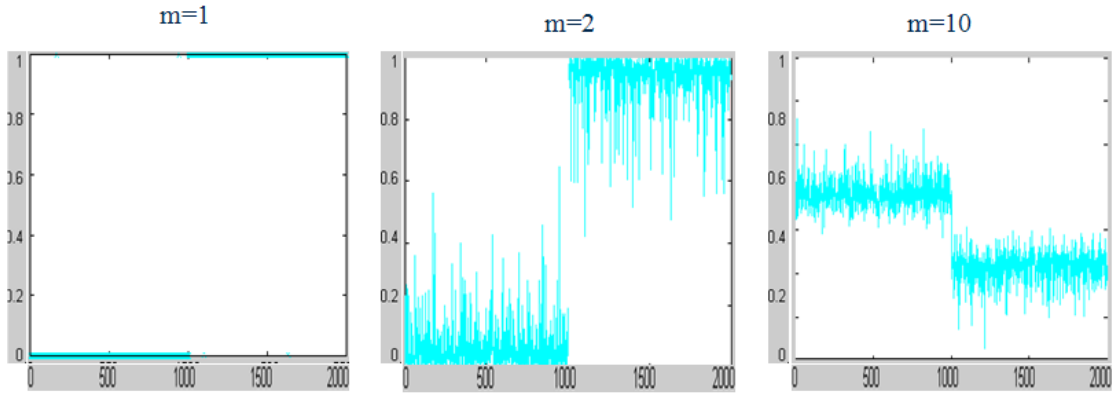
$$J = \sum_{i=1}^c \sum_{j=1}^N U_{ij}^m D_{ij}^2 \quad (3.48)$$

Adım 5: Yeni üyelik matrisi hesaplanır ve Adım 2 ‘ den itibaren işlem yinelenir.

$$U_{ij} = \frac{1}{\sum_{k=1}^n \left(\frac{D_{ij}}{D_{kj}} \right)^{2/(m-1)}} \quad (3.49)$$

3.3.3.2. BCO Yönteminin Eksik Yönleri:

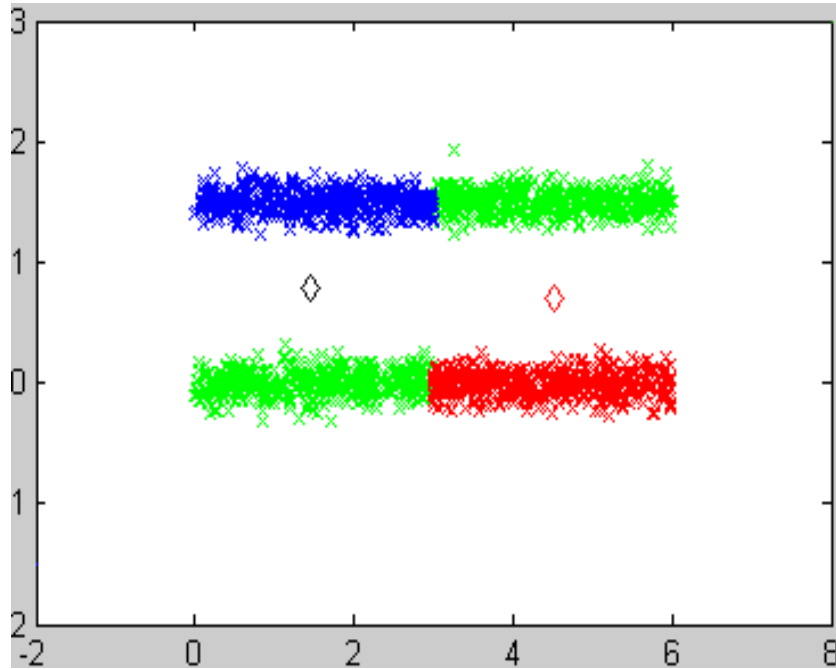
1. Bulanıklaştırma Üstelinin (m) optimum seçimi için teorik bir esas yoktur. m büyüdükçe çözüm bulanıklaşır. m=1 de BCO, Sabit C Ortalamaları (Hard C Means, HCM) gibi daralır ve katı sonuçlar verir. Bulanıklaştırma üsteli (m)’ nin çok büyük değerlerinde tüm noktalar tüm kümelerde eşit üyelik derecelerine sahip olacaktır (Klawonn ve Höppner, 2003).



Simülasyonu yapılan veri seti için: $N=2000$, $C=2$

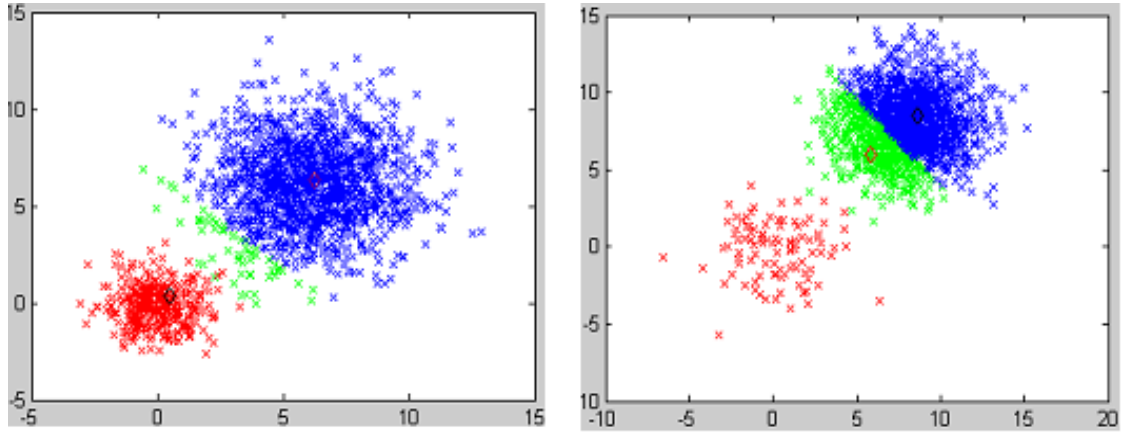
Şekil 3.24 m parametresinin seçimi (Zhang, 2005)

2. Kümeler küresel olmayan biçimlerde olabilir. Bu durumda Şekil 3.25’de görüldüğü gibi dört farklı küme iki farklı küme varmış gibi saptanır.



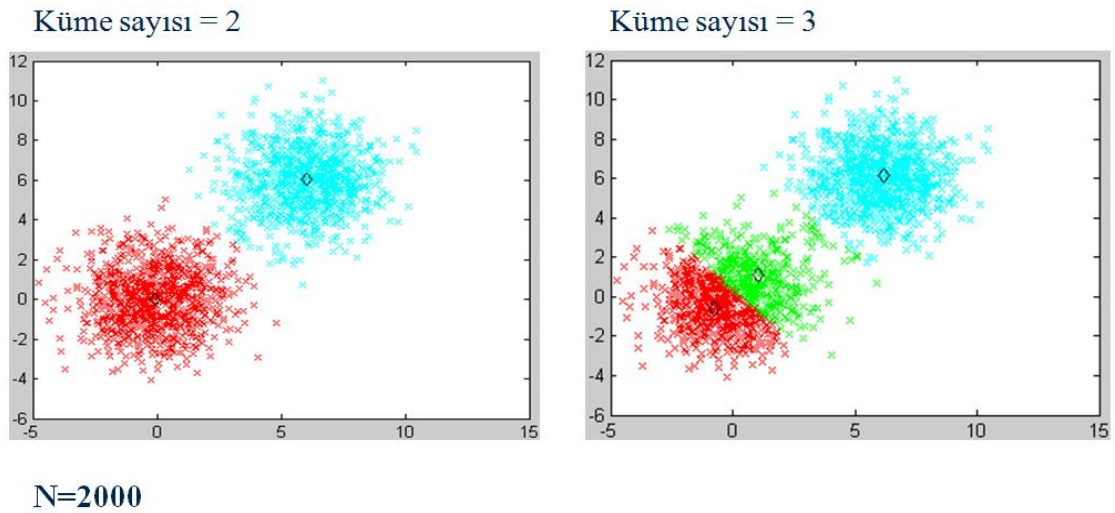
Şekil 3.25 Küresel olmayan veri seti için merkez belirleme (Zhang, 2005)

3. Kümeler farklı boyutlarda ve yoğunluklarda olabilirler. BCO algoritması bu durumda düşük yoğunluklu veya küçük boyutlu kümeleri göz ardı edebilmektedir (Şekil 3.26).



Şekil 3.26 Farklı boyut ve yoğunluktaki veri seti için merkez belirleme (Zhang, 2005)

4. Küme sayısının bilinmesi gereklidir. Algoritma veri içerisinde belirlenen küme sayısı kadar küme merkezi saptamaktadır (Şekil 3.27).

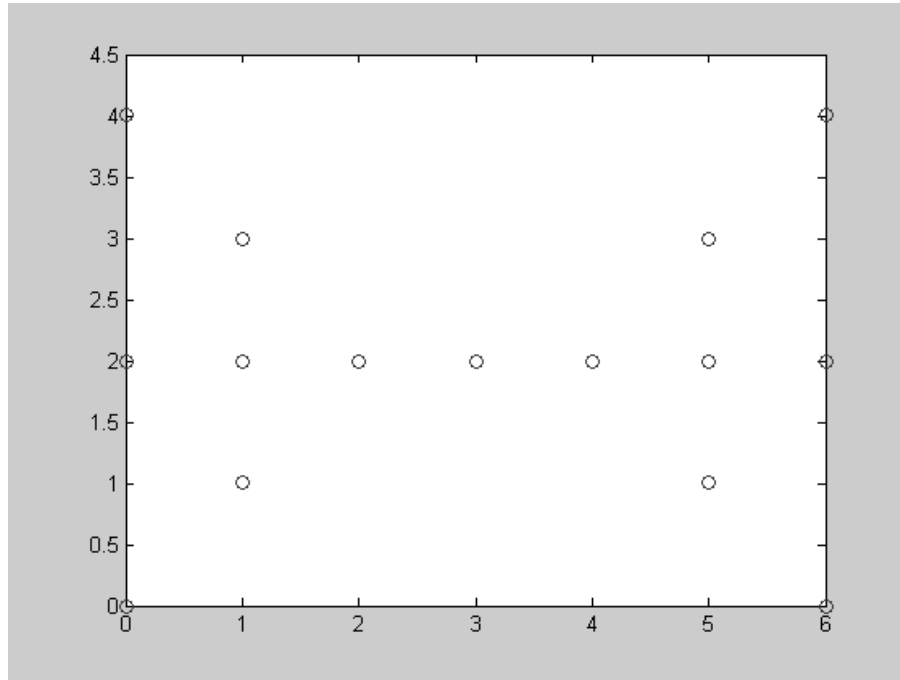


Şekil 3.27 Küme sayısına göre merkez belirleme (Zhang, 2005)

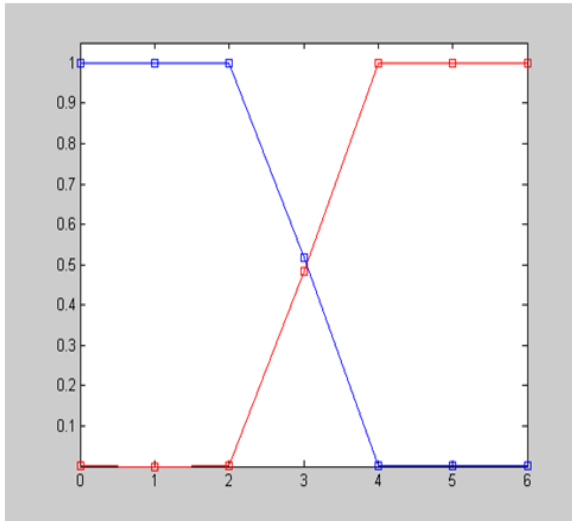
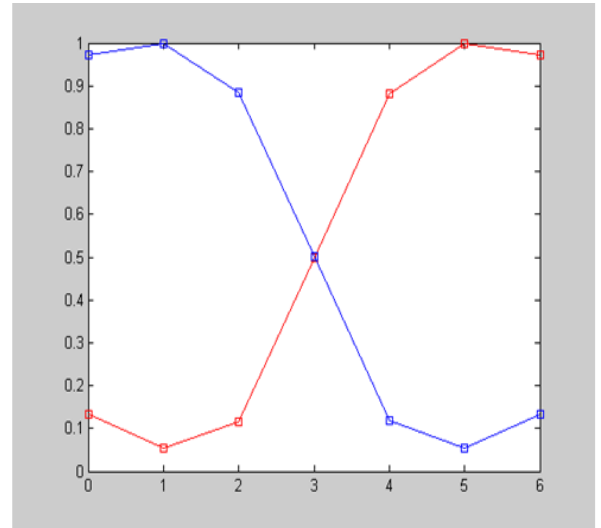
Örnek 3.3 : Kelebek veri seti (Wolkenhaue,) ile m bulanıklaştırma üstelinin etkisi

Düzlemdeki 15 noktanın oluşturduğu M veri seti :

j	1	2	3	4	5	6	7	8
m_j	(0,0)	(0,2)	(0,4)	(1,1)	(1,2)	(1,3)	(2,2)	(3,2)
j	9	10	11	12	13	14	15	
m_j	(4,2)	(5,1)	(5,2)	(5,3)	(6,0)	(6,2)	(6,4)	



Şekil 3.28 Kelebek veri seti

(a) $m=1.25$ için(b) $m=2$ için

Şekil 3.29 BCO üyelik işlevleri

3.3.4. Gustafson-Kessel Kümeleme Algoritması

Gustafson- Kessel (GK), bir veri setindeki farklı geometrik biçimlerdeki kümeleri algılamak için standart BCO algoritmasında geliştirilerek elde edilmiş bir algoritmadır (Wolkenhaue,). Her küme iç çarpım normunun içerdiği bir A_i norm endükleme matrisine sahiptir.

Denklem(3.43) ile verilen A_i matrisi, ortalama C fonksiyonunda optimizasyon değişkenleri olarak kullanılır. Böylece her bir küme, verinin topolojik yapısına uygun biçimde uzaklık normu uyarlanarak belirlenir. A endükleme matrisi $A=(A_1, A_2, \dots, A_c)$ ile gösterilsin. GK algoritması için amaç fonksiyonu Denklem(3.50) ile tanımlanır.

$$J(Z;U,V,A) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ikA}^2 \quad (3.50)$$

A_i sabit bir değer olması durumunda Denklem(3.43) doğrudan uygulanabilir. Fakat A_i doğrusal olduğunda Denklem(3.50) ile verilen amaç fonksiyonu doğrudan minimize edilemez. A_i daha pozitif tanımlı yapılarak istenen küçüklükte J elde edilebilir. Uygun bir çözüm elde etmek için A_i bazı yollarla zorlanmalıdır. Bunu başarmanın genel yolu A_i 'nin determinantını sınırlamaktır (Babuska, 1998).

$$|A_i| = \rho_i, \quad \rho_i > 0, \quad \forall i \quad (3.51)$$

Langrange çarpım yöntemi kullanılarak Denklem(3.52) de verilen A_i elde edilir.

$$A_i = [\rho_i \det(F_i)]^{1/n} F_i^{-1} \quad (3.52)$$

F_i i. kümenin bulanık kovaryans matrisi olarak tanımlanır ve Denklem(3.53) ile verilir.

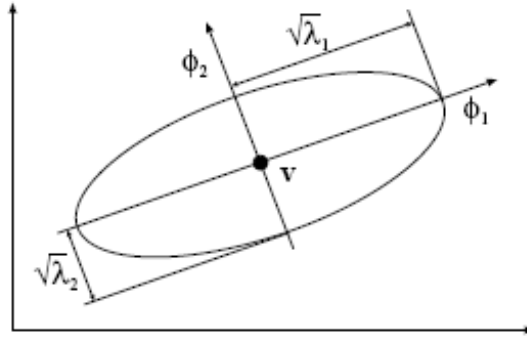
$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (Z_k - V_i)(Z_k - V_i)^T}{\sum_{k=1}^N (\mu_{ik})^m} \quad (3.53)$$

Denklem (3.52) ve (3.53) ‘ü içeren Denklem (3.43), Z_k verileri ve V_i küme merkezleri arasındaki genelleştirilmiş karesel Mahalonobis uzaklık normunu verir.

Açıklama 1: Bulanıklaştırma üstelinin (m) optimum seçimi için teorik bir esas yoktur. m büyüdükçe tıpkı BCO algoritmasındaki gibi çözüm bulanıklaşır (Babuska, 1998).

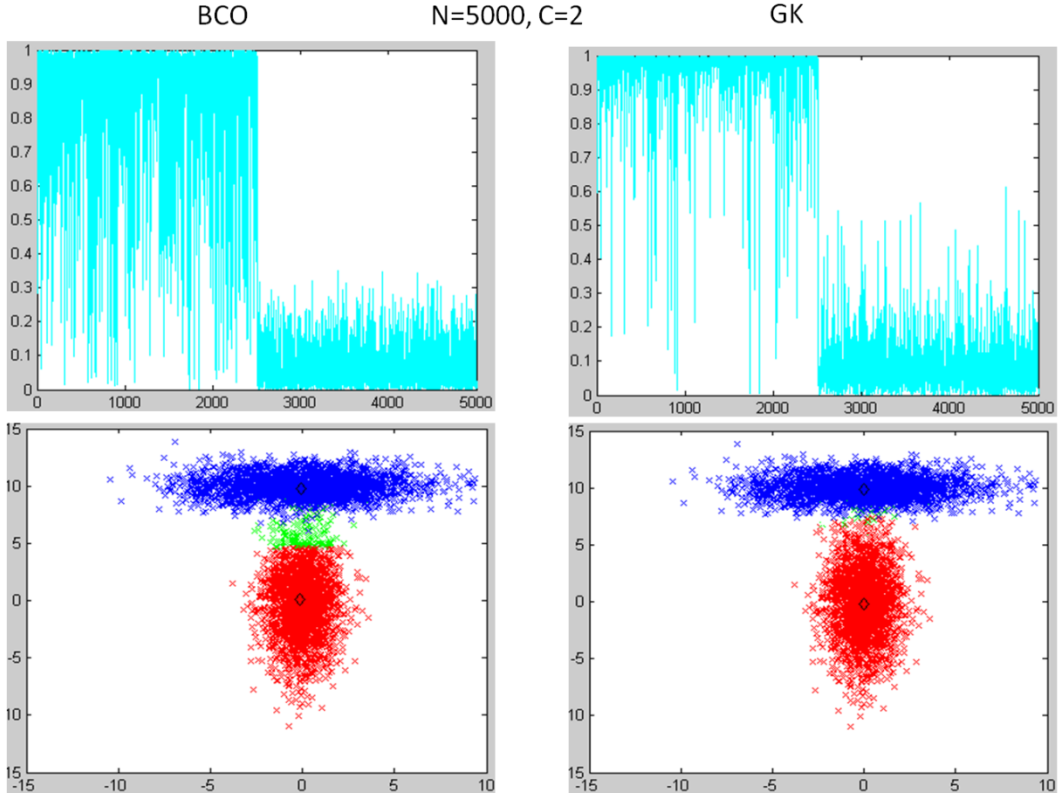
Açıklama 2: Hiçbir ön bilgi olmaksızın küme hacmi ρ_i , basit bir biçimde her küme için 1 seçilir. Denklem(3.51)’de belirtilen kısıt yüzünden GK algoritmasının dezavantajı eşit hacimdeki kümeleri bulabiliyor olmasıdır. Ayarlanan hacimli GK uygulaması için uygun küme birleştirme yordamıyla (Compatible Cluster Merging–CCM) kombinasyonu kullanılmaktadır (Babuska, 1998).

Açıklama 3: Küme kovaryans matrisinin öz yapısı kümenin yönelimi ve biçimi hakkında bilgi verir. Şekil 3.30’da görüldüğü gibi kümenin hiper elipsoid eksenlerinin uzunluğunu F_i ’nin özdeğerlerinin (λ) karekökü, eksenlerin doğrultusunu F_i ’nin özvektörleri verir (Babuska, 1998).

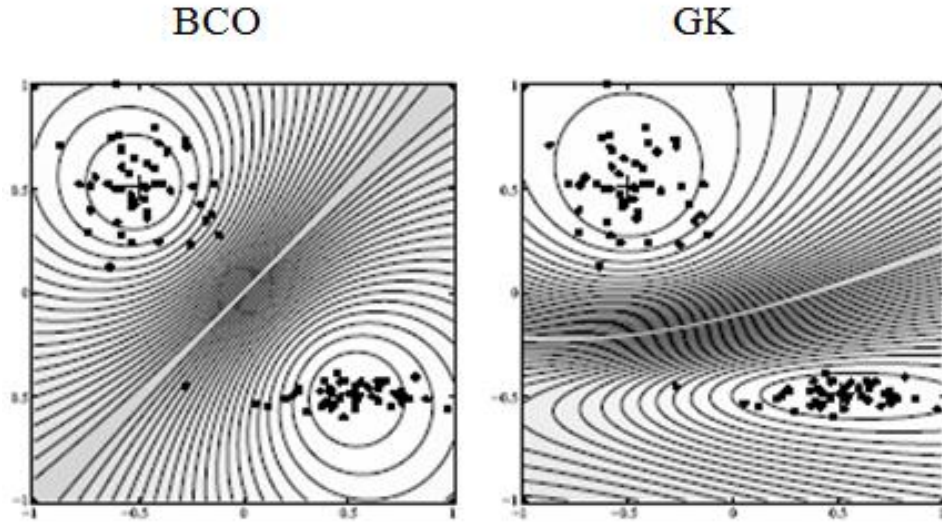


Şekil 3.30 İç çarpım normu (Babuska, 1998)

Açıklama 4: GK algoritmasının BCO algoritmasından üstün yönü, bir veri seti içindeki farklı şekil ve yönelimdeki kümeleri algılayabilmesidir. Fakat GK, BCO’dan daha fazla hesap gücü içermektedir. Çünkü küme kovaryans matrisinin tersi ve determinantı her bir iterasyonda hesaplanmak zorundadır (Babuska, 1998).

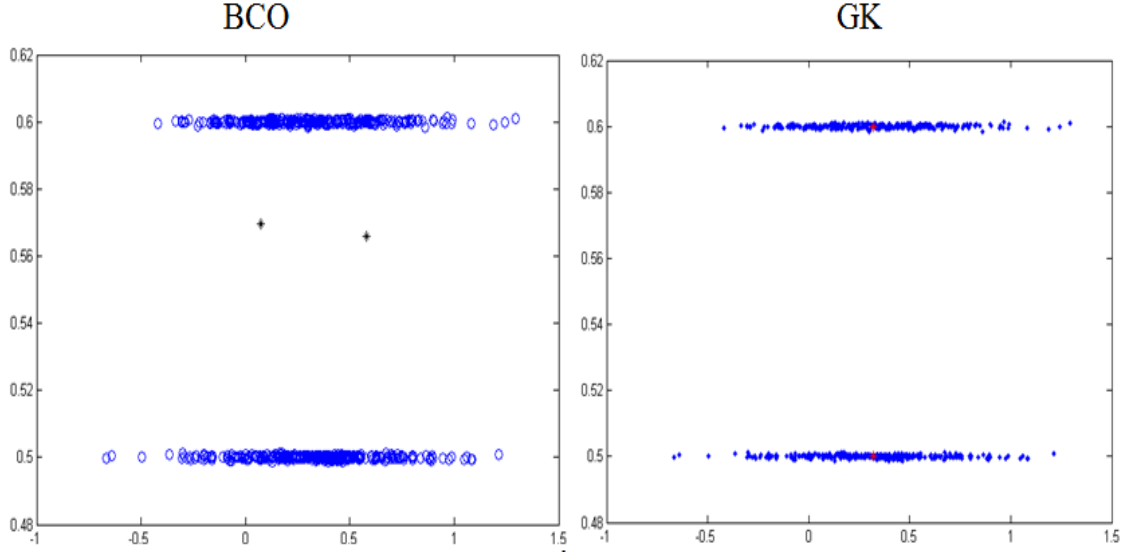


Şekil 3.31 Farklı biçimlerdeki kümeler için BCO ve GK algoritmaları (Zhang, 2005)



Şekil 3.32 Farklı yönelimlerdeki veriler için BCO ve GK algoritmaları (Babuska, 1998)

Örnek 3.4: Rastgele oluşturulmuş 300 elemanlı küresel olmayan bir veri seti için BCO ve GK algoritmaları çalıştırılarak benzetim sonucu verilmiştir.



Şekil 3.33 BCO ve GK algoritmaları için benzetim örneği

Şekil 3.33 ile görüleceği gibi bazı veri tiplerinde GK algoritmasının başarımı BCO algoritmasının başarımından daha iyidir.

3.3.4.1. GK Algoritması

Veri seti Z , küme sayısı $1 < c < N$, ağırlıklandırma üsteli $m > 1$ ve sonlandırma kriteri $\varepsilon > 0$ ve Başlangıç bölümlene matrisi rastgele seçimle $U^{(0)}$ seçilirse;

Tekrarla $l=1, 2, \dots$

Adım 1: Küme merkezlerini hesapla:

$$V_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m Z_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c \quad (3.54)$$

Adım 2: Küme kovaryans matrisini hesapla:

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m (Z_k - V_i^{(l)})(Z_k - V_i^{(l)})^T}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c \quad (3.55)$$

Adım 3: Uzaklıkları hesapla:

$$D_{ikA_i}^2 = (Z_k - V_i^{(l)})^T [\rho_i \det(F_i)^{1/n} F_i^{-1}] (Z_k - V_i^{(l)}), \quad 1 \leq i \leq c, \quad 1 \leq k \leq N \quad (3.56)$$

Adım 4: Bölümleme matrisini güncelle:

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikA_i} / D_{jkA_i})^{2/(m-1)}} \quad (3.57)$$

$\|U^{(l)} - U^{(l-1)}\| < \varepsilon$ oluncaya kadar...

3.3.5. Bölüm Doğrulama

Bölüm doğrulama J amaç işlevini minimize edecek en uygun c küme sayısının bulunması işlemidir. Çeşitli bölüm doğrulama kriterleri, bölüm katsayıları, bölüm entropisi, bölüm yoğunluğu vb doğrulama ölçütleri literatürde mevcuttur (Bezdek, 1981; Gath ve Geva, 1989; Fan ve ark. , 2000; Chen ve Linkens, 2001; Chen ve Linkens, 2004).

Doğrulama indeksleri 3 ana grupta toplanabilir (Bouguessa ve ark. , 2006).

1. Oluşan bölümün bulanık üyelik derecesinin özelliklerini kullanan grup
2. Veri setinin geometrik yapısı ve üyelik derecesinin özelliğinin birleşiminden oluşan grup
3. Yüksek hacim ve yoğunluk konseptini kullanan grup

Birinci grup indeksler sadece üyelik matrislerinin elemanları kullanılarak hesaplandığı için verinin geometrik özellikleriyle doğrudan bağlantıdan yoksundur.

Kümeler geometrik yapısı ve üyelik derecesinin özelliğinden oluşan grup olan 2. grup indekslerden bazıları;

Fukuyama ve Sugeno Doğrulama indeksi: Kümelerin yoğunluğu ve ayrımı arasındaki uyumsuzluğu ölçer. Örtüşen veri setlerinde iyi sonuç veren bir indekstir.

Zahid Doğrulama indeksi: Bulanık ayırma ve yoğunluk oranıyla verilen iki fonksiyonun birleşiminden oluşur. Küresel biçimli örtüşen kümeler içeren durumlarda iyi bir performans gösterir.

Geva Doğrulama indeksi: Orta derecede örtüşen kümelerde iyi sonuç veren bir indekstir.

Xie Doğrulama indeksi: Örtüşmenin düşük olduğu durumlarda iyi sonuçlar üretir.

Sun Doğrulama indeksi: Düşük ve orta dereceli örtüşmelerde iyi sonuçlar verir.

Bouguessa indeksi: Kümelerin örtüşmesinden etkilenme kararlı sonuçlar üreten bir indekstir.

3.3.5.1. BCO için bölüm doğrulama:

Karşı yayılım ağları, ham veriden başlangıç kümelerinin üretilmesinde doğal bir seçimdir. KYA ile bulunan kümeler optimal değildir. Çünkü, kullanıcının girdiği yarıçap parametresine bağlı sayıda değişen küme bulan bir algoritmadır. Bu yönüyle oluşturduğu model de optimal bir model değildir.

Bezdek tarafından önerilmiş olan Bölüm katsayıları ve Bölüm dağınımı indeksleri 1. grup doğrulama indeksleridir. Bunlardan Denklem(3.58) ile verilen Bölüm dağınımı indeksi, BCO algoritmalarıyla birlikte kullanılan basit ve yaygın bir küme doğrulama kriteridir.

$$H(U,c) = -\frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c U_{ik} \log(U_{ik}) \quad (3.58)$$

Optimal kümeleme için $H(U,c)$ minimize edilir. Fakat doğrulamanın güvenilirliği etkileyen U_{ik} nın değeri çok küçük olduğu zaman logaritmik fonksiyonların hesaplama hatası çok büyük olacaktır. Bu eksikliğin üstesinden gelmek için Denklem(3.59) ile verilen yeniden düzenlenmiş bölüm dağınım indeksi kullanılmaktadır (Chen, 1998) .

$$H(u,c) = -\frac{1}{n} \sum_{k=1}^n \max_i(U_{ik}) \log(\max_i(U_{ik})) \quad (3.59)$$

Denklem(3.59) ile verilen indeksi kullanan Bölüm doğrulama algoritması şu biçimdedir:

Adım 1: Maksimum küme sayısı $c_{\max}$, yineleme sınırı T , ağırlıklandırma üsteli m ve sonlandırma kriteri $\varepsilon_{CLU} > 0$ seçilir.

Adım 2: $c = 2, 3, \dots, c_{\max}$ ve küme merkezlerinin başlangıç konumları $V_0 = (v_{10}, v_{20}, \dots, v_{c0})$ alınır.

Adım 3: $t=1, 2, \dots, T$ alınır. U üyelik matrisi ve V küme merkezleri hesaplanır.

$$U_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{D_{ij}}{D_{kj}} \right)^{2/(m-1)}} \quad V_i = \frac{\sum_{j=1}^c U_{ij}^m x^j}{\sum_{j=1}^c U_{ij}^m}, \quad i = 1 \dots c \quad (3.60)$$

Eğer $\|V_t - V_{t-1}\| < \varepsilon$ ise bir sonraki adıma geçilir aksi halde adım 3 yinelenir.

Adım 4: Eşitlik (8) den $H(u, c)$ hesaplanır.

Eğer $H(c-1) < H(c)$ ve $H(c-1) < H(c-2)$ ise işlem sonlandırılır. Optimal küme sayısı $c=c-1$ olur. Aksi takdirde Adım 2 ye dönülür.

3.3.5.2. GK için Bölüm Doğrulama:

2006 yılında M. Bouguessa tarafından önerilen bölüm doğrulama indeksi (V_{SC}), üyelik derecelerini (ayırışma) ve verinin geometrik yapısını dikkate alma özelliğiyle GK algoritmasıyla kullanılması uygundur. V_{SC} indeksi Denklem(3.61) eşitliğiyle verilmektedir.

$$V_{SC}(U, V, X) = \frac{Sep(c)}{Comp(c)} \quad (3.61)$$

$Sep(c)$, bulanık kümelerin bulanık ayrımı olarak adlandırılır ve $Sep(c) = \text{trace}(S_B)$ ile hesaplanır. S_B , küme içi bulanık dağılım matrisi olarak adlandırılır ve Denklem(3.62) ile tanımlanır.

$$S_B = \sum_{i=1}^c \sum_{k=1}^L (u_{ik})^m (V_i - \bar{V})(V_i - \bar{V})^T \quad (3.62)$$

Trace, köşegen elemanlarının toplamı, $\bar{V}$ ise merkezlerin ortalama değerleridir. Sep(c)' nin büyük değerleri iyi ayrılmış bulanık kümeler ile karakterize edilen bulanık c bölümlerini gösterir.

Comp(c) , bulanık c bölümlerinin toplam yoğunluğudur ve Denklem(3.63) ile tanımlanır. Comp(c) 'nin küçük olması yoğunluğun fazla olmasıyla ilişkilidir.

$$Comp(c) = \sum_{i=1}^c trace(\sum i) \quad (3.63)$$

Σi , bulanık kovaryans matrisi olup Denklem(3.64) ile tanımlıdır.

$$\Sigma i = \frac{\sum_{k=1}^L (u_{ik})^m (x_k - V_i)(x_k - V_i)^T}{\sum_{k=1}^L (u_{ik})^m} \quad (3.64)$$

Buradan hareketle yoğun ve iyi ayrılmış bir bölümlere için V_{SC} nin maksimum yapılması gerektiği sonucuna ulaşılır.

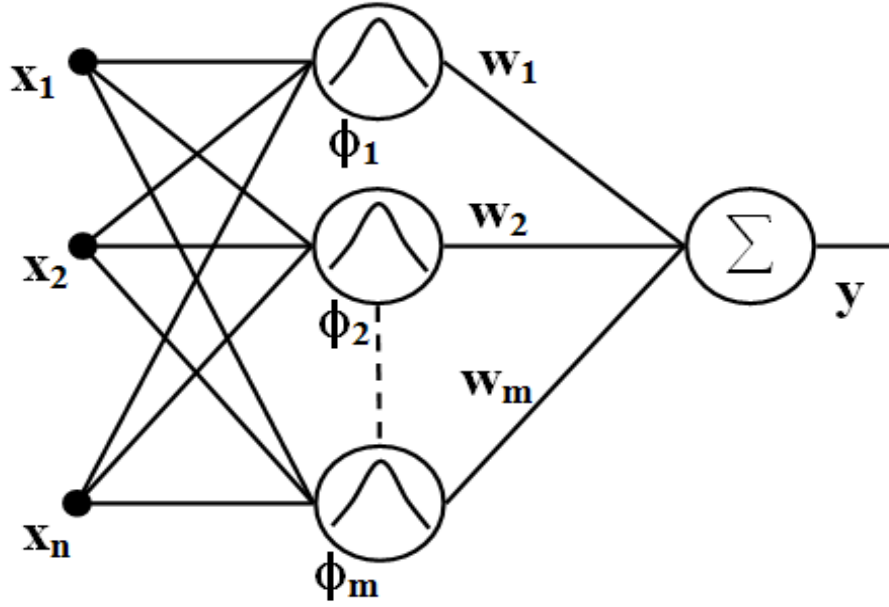
3.4. ÖĞRENME ALGORİTMALARI

3.4.1. Radyal Tabanlı İşlevli Yapay Sinir Ağları

Radyal Tabanlı İşlevli (RTİ) yapay sinir ağları Broomhead ve Lowe (1988) tarafından sinir ağı literatürüne girmiştir (Schwenker ve ark. , 2001). Radyal Tabanlı İşlevli (RTİ) Yapay Sinir Ağları basit biçimde yerel özelliklerinin ayarlanabilirliği ile sigmoid işlevli çok katmanlı yapay sinir ağlarına alternatiftir (Neruda ve Kudov, 2005). Doğrusal bir çıkış katı ve Radyal Tabanlı İşlevli' nden oluşan bir gizli katmandan oluşan ağ yapısı Şekil 3.34' de verilmiştir.

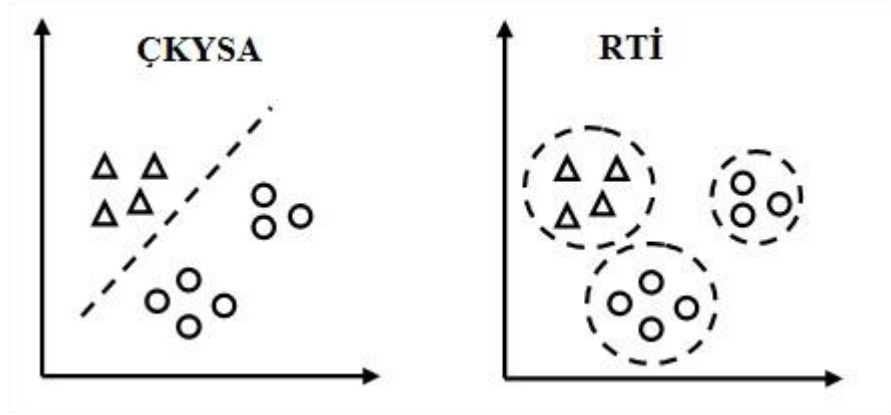
Üç katmandan oluşan bu yapıda birinci katman n boyutlu giriş verisini içermektedir. Gizli katman olarak da adlandırılan ikinci katman m adet radyal tabanlı işlevlerinden oluşmaktadır. Giriş verisi bu katmanda Radyal Tabanlı İşlevlerinden geçirilerek m adet çıkış(ϕ) hesaplanır. Son katmanda ise her işlev çıkışı w ağırlıklarıyla

çarpılarak toplanır. Böylece RTİ ağıının üretecek olduğu çıkış, gizli katmanda bulunan her bir Radyal Taban İşlevinin ürettiği çıkışların doğrusal bileşkesidir.



Şekil 3.34 Radyal Taban İşlevli Yapay Sinir Ağ Yapısı

Yapıları çok katmanlı yapay sinir ağlarına göre oldukça sade olması nedeniyle eğitilme ve test edilme aşamalarındaki hesaplama süreleri daha azdır. RTİ ağlarında aktivasyon fonksiyonu olarak, yaygın bir biçimde gauss işlevi kullanılır Çok katmanlı yapay sinir ağlarında eğitim aşamasındaki tüm parametreler rastsal biçimde seçilir. Fakat RTİ ağlarında kullanılan gaussların sayısı ve merkez eğitimlerinde kümeleme algoritmalarından yararlanılabilir (Gomm ve Yu, 2000; Schwenker ve ark. , 2001). Çok Katmanlı Yapay Sinir Ağları doğrusal karar yüzeyleri sınıflandırma yaparken RTİ ağları doğrusal olmayan bir sınıflandırma yapmaktadır. Bu durum Şekil 3.35 ile gösterilmiştir.



Şekil 3.35 ÇKYSA ve RTİ karşılaştırılması

RTİ ağları aktivasyon işlevi olarak Gauss işlevinin kullanıldığı durumda, giriş setindeki her bir veri için j.gizli katman nöronunun üreteceği çıkış değeri Denklem(3.65) ile hesaplanır. Burada “ v_j ”, j.RTİ ‘nin merkezini, “ σ_j ” ise j.RTİ ‘nin genişliğini temsil eden parametrelerdir. Bu denklemde; $\|x-v_j\|$ ifadesi x giriş veri setindeki verilerin, j. gizli katman nöronunun içinde yer alan RTİ’nin merkezine olan öklid uzaklığını göstermektedir. Yani veri setinde bulunan her bir giriş verisi için, ağdaki her bir RTİ bu verinin kendi merkezine olan uzaklığı ile doğru orantılı bir değer üretir.

$$\phi_j = e^{\left[-\frac{\|x-v_j\|^2}{\sigma_j^2} \right]} \quad (3.65)$$

Her bir RTİ, doğrusal ağırlıklandırılarak ağ çıkışı Denklem(3.66) ile verildiği gibi hesaplanır.

$$y = \sum_{j=1}^m w_j \phi_j \quad (3.66)$$

Ağ çıkışının istenen çıkışı ne kadar hatayla yaklaştığı merkez, genişlik ve ağırlık parametrelerinin doğru saptanmasına bağlıdır. Bu nedenle bu ağların eğitiminden bu parametrelerin ayarlanması sağlanır. Bu parametrelerin ayarlanmasında kullanılan en

temel yöntem Gradyan Azalması (Gradient Descent, GD) yöntemidir (Neruda ve Kudov, 2005).

Gradyan Azalması (GA) algoritması, birinci dereceden bir optimizasyon algoritmasıdır. GA kullanarak bir işlevin yerel minumunu bulmak için işlevin mevcut noktasının negatif gradyanının bir adım oranıyla çarpılarak eklenmesi esasına dayanır. Bu yöntem kısaca hata karelerin toplamını minimum etmeye çalışarak minimum hatayı sağlayacak parametreleri bulmaya çalışır. Bu nedenle yinelemeli bir yöntemdir

Herhangi bir $x(n)$ giriş veri setine karşılık ağız ürettiği çıkış $y(n)$, bu giriş veri seti için gerçekte olması gereken çıkış değeri $d(n)$ olmak üzere, yöntem Denklem(3.67) ile verilen hatanın enerjisini minimum yapmayı amaçlar.

$$E = \frac{1}{2} \sum_{n=1}^m (d(n) - y(n))^2 \quad (3.67)$$

E hata yüzeyinde azalma yönünde hareket edebilmek, bu hareketi de ağıdaki merkez genişlik ve ağırlık parametrelerini değiştirerek yapabilmek için hata yüzeyinin bu parametrelere göre türevine bakılmak zorundadır. Böylece parametrelerdeki uygun değişimler ile her bir adımda eğimin negatif olduğu tarafa doğru ilerleyerek ve minimum noktasına ulaşmak amaçlanır. Çıkış işlevi Denklem(3.67)'de yerleştirilirse ve vektörel olarak ifade edilirse:

$$E = \frac{1}{2} (\overline{d(n)} - \overline{w\phi(n)})^2 \quad (3.68)$$

Denklem(3.68)'in w ağırlıklarına göre negatif yönde türevi alınırsa ağırlıkların değişim miktarı Denklem(3.69)'da verildiği gibi hesaplanır.

$$-\frac{dE}{dw} = -(\overline{d(n)} - \overline{w\phi(n)}) \cdot \overline{\phi(n)} \quad (3.69)$$

Bu deęişim miktarı kullanılarak güncellenmiř aęırlık parametresi , $(\overline{d(n)} - \overline{w}\overline{\phi(n)})$, hata ($e = d - y$) vektörü cinsinden Denklem (3.71)'de verildięi gibi hesaplanır. Hatadaki azalma yönündeki her bir adım μ_w öğrenme parametresi ile belirlenir.

$$w_{i+1} = w_i + (\overline{d(n)} - \overline{w}\overline{\phi(n)}) \cdot \overline{\phi(n)} \quad (3.70)$$

$$w_{i+1} = w_i + \mu_w \cdot \overline{e} \cdot \overline{\phi(n)} \quad (3.71)$$

Denklem(3.67)'deki enerji işlevine gauss işlevi yerleřtirildięinde Denklem(3.72) elde edilir.

$$E = \frac{1}{2} \left(\overline{d(n)} - \overline{w} e^{-\frac{(\overline{x(n)} - \overline{v(n)})^2}{\sigma^2}} \right)^2 \quad (3.72)$$

Denklem (3.72)'nin benzer biçimde merkez (v) ve genişliklere (σ) göre negatif türevleri alınarak gizli katmandaki j . RTİ biriminin güncellenmiř parametreleri Denklem(3.73) ve Denklem(3.74) de verilmiřtir. Burada μ_c ve μ_σ öğrenme parametreleridir.

$$v(n+1)_j = v(n)_j + \mu_v \cdot e(n) \cdot w(n)_j \cdot e^{-\frac{(x(n)-v(n)_j)^2}{\sigma(n)_j^2}} \cdot \frac{(x(n) - v(n)_j)}{\sigma(n)_j^2} \quad (3.73)$$

$$\sigma(n+1)_j = \sigma(n)_j + \mu_\sigma \cdot e(n) \cdot w(n)_j \cdot e^{-\frac{(x(n)-v(n)_j)^2}{\sigma(n)_j^2}} \cdot \frac{(x(n) - v(n)_j)^2}{\sigma(n)_j^3} \quad (3.74)$$

3.4.2. Destek Vektör Makinaları

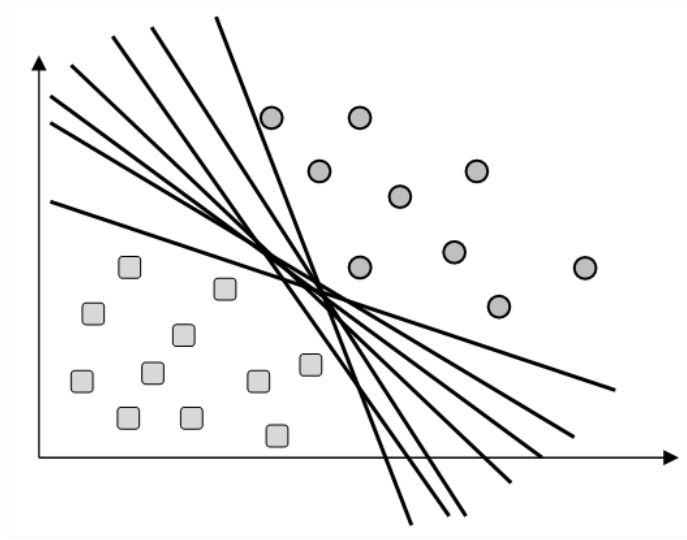
Teorik temelleri 1960'lı yılların sonlarında Vapnik tarafından atılan Destek Vektör Makinaları (DVM) ikili sınıflandırma problemlerinin çözümünde kullanılan istatistiksel öğrenme teoremine dayanan örüntü sınıflandırma yöntemidir. DVM iki sınıf arasındaki uzaklığın en büyük olduğu durumu saptayarak veri kümesini mümkün olduğu kadar iyi sınıflandıran en uygun ayırıcı düzlemi bulunmayı amaçlar. Bu amaç, doğrusal olmayan örnek uzayının doğrusal olarak ayrılabilceği yüksek boyuta aktarıldıktan sonra, farklı örnekler arasındaki en büyük sınırın bulunmasıyla gerçekleştirilir (Vapnik, 1998; Zheng ve Xue, 2009).

Algoritmik açıdan DVM veri bölümlene tabanlı bir öğrenme yöntemidir. Böyle yöntemler sınıflandırma işlevi (hiper-düzlem) yardımıyla veri uzayını bölerek bilinmeyen verilerin sınıf etiketlerini tanımlamaktadır. Bu amaçla veriyi bölmek için en uygun ayırıcı hiper- düzlemi kullanılmaktadır. DVM, biyolojik bilgi işleme, görüntü sınıflandırma, el yazısını ve ses tanıma gibi uygulama alanlarına sahiptir (Abe, 2005; Zheng ve Xue, 2009).

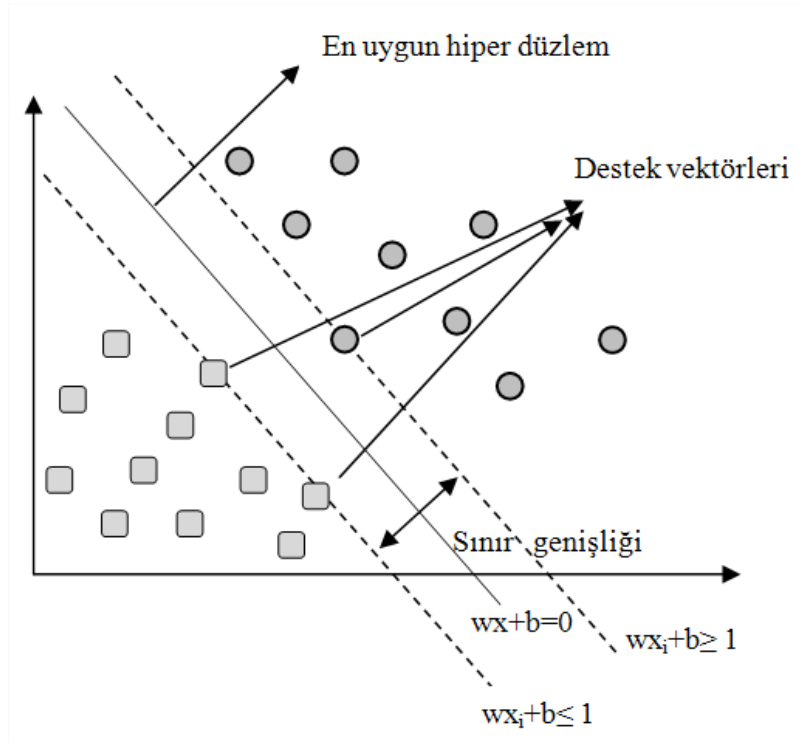
DVM' ler doğrusal ve doğrusal olmayan DVM olmak üzere ikiye ayrılır. Doğrusal DVM' nin yapısındaki çekirdek işlevi sadece giriş uzayının bir ürünüdür. Doğrusal olmayan DVM' de ise, probleme uygun doğrusal olmayan bir çekirdek fonksiyonunun seçilmesine ihtiyaç vardır.

3.4.2.1. Doğrusal Destek Vektör Makineleri

İkili sınıflandırma probleminde, Şekil 3.36'da görüldüğü gibi bir veri kümesini doğrusal olarak ayırabilen sonsuz sayıda hiper düzlem vardır. Bu düzlemler arasında en geniş sınıra sahip sadece bir hiper düzlem bulunmaktadır. Veri kümesini hatasız bir şekilde ayırabilen bu hiper-düzlemler arasından en geniş sınırlı hiper düzlemin bulunması amaçlanmaktadır. DVM karar yüzeyini oluştururken iki sınıfa olan uzaklığı maksimum yapmaya çalışmaktadır. En uygun ve en büyük sınır genişliğine sahip hiper düzlemi bulmak için Şekil 3.37'de görülen ve destek vektörleri olarak adlandırılan noktaları kullanılmaktadır. Bu noktalar en büyük geometrik mesafeye sahiptirler (Burges, 1998; Zheng ve Xue, 2009).



Şekil 3.36 İki sınıflı veri kümesini ayırabilen hiper-düzlemler



Şekil 3.37 İki boyutlu uzaydaki iki sınıflı veri kümesi için en uygun hiper-düzlem

Doğrusal ayrılabilir bir veri kümesi, $(x_1, y_1), \dots, (x_l, y_l)$, $x \in R^n$, vektörü eğitim örneklerini, $y \in \{+1, -1\}$ vektörü ise bu örneklerle karşılık gelen sınıfları temsil edecek biçimde verilsin. Buna göre DVM, Denklem(3.75) ile verilen doğrusal karar

işlevi ile tanımlanan en uygun ayırıcı düzlemi bulunmayı amaçlar (Burges, 1998; Abe, 2005; Zheng ve Xue, 2009).

$$f(x) = w^T x + b \quad (3.75)$$

Denklem(3.75)'de w , l boyutlu ağırlık vektörü, b ise eşik değerini göstermektedir. Her bir eğitim örneği x_i için karar işlevi Denklem(3.76) ile verilmektedir.

$$\begin{aligned} f(x_i) = w^T x_i + b &\geq 0 \quad y_i = 1 \quad \text{için} \\ f(x_i) = w^T x_i + b &< 0 \quad y_i = -1 \quad \text{için} \quad i = 1..l \end{aligned} \quad (3.76)$$

Bu kısıtlamalar Denklem(3.77)'de verildiği gibi bir arada biçimde de ifade edilebilir.

$$y_i[(w^T x_i) + b] \geq 1, \quad i = 1..l \quad (3.77)$$

Bu durumda en geniş sınıra sahip en uygun hiper-düzlem Denklem(3.76) ile verilen kısıtlamalar altında $J(w)$ fonksiyonu en küçük yapılarak bulunabilmektedir.

$$J(w) = \frac{1}{2} w^T w = \frac{1}{2} \|w\|^2 \quad (3.78)$$

Denklem(3.78) ile verilen ifade ikinci dereceden bir optimizasyon problemidir. $J(w)$ fonksiyonunu en küçük yapmak için Lagrange çarpanları yöntemi kullanılmaktadır (Burges, 1998; Vapnik, 1998; Abe, 2005; Zheng ve Xue, 2009; Campbell ve Ying, 2011). Lagrange fonksiyonu Denklem(3.79) ile verilmiş olup α_i pozitif Lagrange çarpanlarıdır.

$$L(w, b, \alpha) = \frac{1}{2}(w^T w) - \sum_{i=1}^l \alpha_i \{[(w^T x_i) + b]y_i - 1\} \quad (3.79)$$

Bu denklemin çözülmesi oldukça karmaşıktır. Uygun çözümün bulunması için Karush-Kuhn-Tucker (KKT) şartları kullanılarak ikili probleme dönüştürülmelidir. Böylece Denklem(3.79), $L(w, b, \alpha)$ fonksiyonu, $\alpha_i \geq 0$ değerlerine göre en büyük yapılarak, w ve b değerlerine göre ise en az yapılarak çözülebilmektedir. $L(w, b, \alpha)$ işlevinin w ve b değerlerine göre en küçük noktalarını bulabilmek için $L(w, b, \alpha)$ işlevinin, w ve b değerlerine göre türevleri sıfıra eşitlenir. Türev eşitlikleri Denklem(3.80) ve Denklem(3.81) ile verilmektedir.

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i \quad (3.80)$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^l \alpha_i y_i = 0 \quad (3.81)$$

Karush-Kuhn-Tucker (KKT) koşulları Denklem(3.82) - Denklem(3.85) ile verilmektedir.

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \quad (3.82)$$

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \quad (3.83)$$

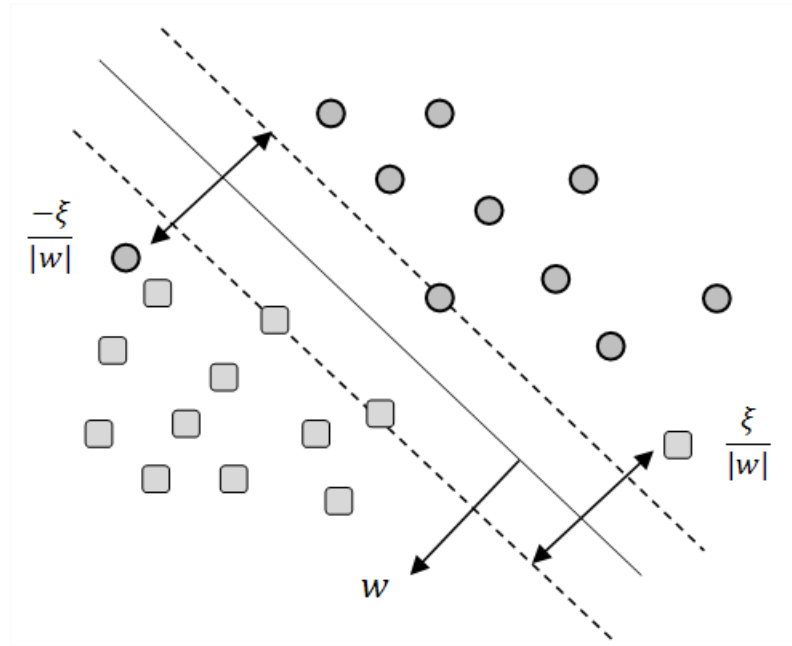
$$\alpha_i \{[(w^T x_i) + b]y_i - 1\} = 0, \quad i = 1, \dots, l \quad (3.84)$$

$$\alpha_i \geq 0, \quad i = 1, \dots, l \quad (3.85)$$

Veri kümesindeki her bir örnek Denklem (3.84)'i sağlamalıdır. $\alpha_i \neq 0$ olan her x_i giriş verisi destek vektörü olarak adlandırılmaktadır. Lagrange işlevinde KKT koşulları yerleştirilerek Denklem(3.86)'da verildiği biçimde düzenlenir.

$$L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (3.86)$$

Eğer, örnekler doğrusal olarak tamamen ayrılabilir durumda değilse, problemin çözümü için pozitif zayıflık değişkenleri ($\xi_i \geq 0$) kullanılır. Bu duruma ait en uygun ayırıcı düzlemin temsili gösterimi Şekil 3.38’ de verilmiştir.



Şekil 3.38 Eğitim verilerinin doğrusal olarak ayrılmadığı durum

Buna göre Denklem(3.88) verilen kısıtlamalar altında $J(w)$ işlevi tekrar Denklem(3.87) verildiği gibi yazılabilmektedir.

$$J(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (3.87)$$

$$y_i (w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0, \quad i = 1, \dots, l \quad \text{için} \quad (3.88)$$

Denklem(3.87)’de ikinci terim yanlış sınıflandırma noktasını kontrol etmektedir. $\xi_i = 0$ olması durumunda x_i örneği doğru sınıflandırılmış, $\xi_i \geq 0$ ise yanlış sınıflandırılmış demektir. Doğrusal olarak ayırlamama durumunda, eğitim verisi içindeki olası her duruma karşı bir çözüm üretilmesini engellemek için sisteme bir C parametresi eklenir. Bu parametre, Lagrange çarpanlarının alabilecekleri en büyük değeri de göstermektedir ve DVM’nin eğitimi aşamasında belirlmesi gereken parametrelerden biridir.

Doğrusal olarak ayrılabilir durumda olduğu gibi Denklem(3.87)’i en küçük yapmak için Lagrange çarpanları yöntemleri kullanılmaktadır. Lagrange işlevi α_i ve β_i Lagrange çarpanlarını temsil edecek biçimde Denklem(3.89)’daki gibi yazılmaktadır.

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i (y_i (w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^l \beta_i \xi_i \quad (3.89)$$

β_i parametresi ξ_i zayıflık değerlerinin pozitif olarak kalmasını sağlamak için eklenmiştir. $L(w, b, \xi, \alpha, \beta)$ fonksiyonunun, w, b ve ξ değerlerine göre en küçük noktalarını bulabilmek için $L(w, b, \xi, \alpha, \beta)$ fonksiyonunun, w, b ve ξ değerlerine göre türevleri Denklem(3.90) - Denklem(3.92) ile verildiği biçimde sıfıra eşitlenir.

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^l \alpha_i y_i x_i \quad (3.90)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial b} = 0 \Rightarrow \sum_{i=1}^l \alpha_i y_i = 0 \quad (3.91)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial \xi} = 0 \Rightarrow C = \alpha_i + \beta_i \quad (3.92)$$

Buna göre Karush-Kuhn-Tucker (KKT) koşulları Denklem(3.93) - Denklem(3.98) ile verilmektedir.

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial w} = 0 \quad (3.93)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial b} = 0 \quad (3.94)$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial \xi} = 0 \quad (3.95)$$

$$\alpha_i (y_i (w^T x_i + b) - 1 + \xi_i) = 0 \quad i = 1, \dots, l \quad \text{için} \quad (3.96)$$

$$\beta_i \xi_i = 0 \quad i = 1, \dots, l \quad \text{için} \quad (3.97)$$

$$\alpha_i \geq 0, \quad \beta_i \geq 0, \quad \xi_i \geq 0 \quad i = 1, \dots, l \quad \text{için} \quad (3.98)$$

Lagrange işlevinde KKT koşulları yerleştirildiğinde Denklem(3.99) elde edilir ve Denklem(3.100) ile verilen kısıtlamaları sağlamalıdır .

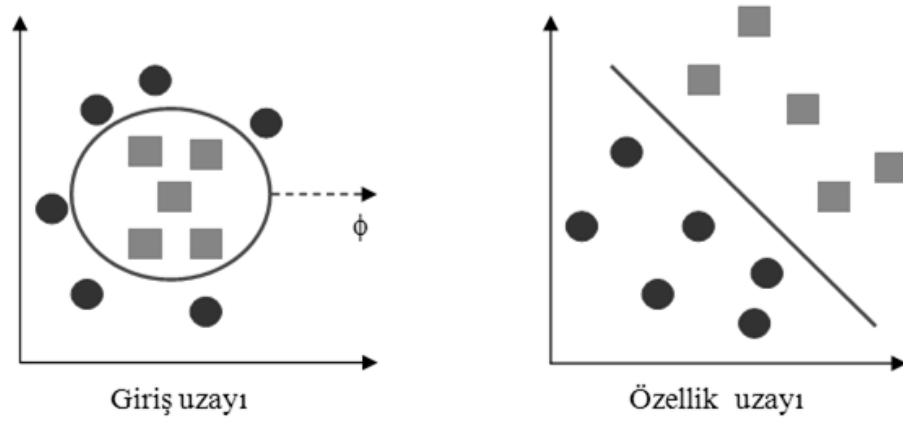
$$L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (3.99)$$

$$\sum_{i=1}^l y_i \alpha_i = 0, \quad C \geq \alpha_i \geq 0 \quad i = 1, \dots, l \quad \text{için} \quad (3.100)$$

Denklem(3.99)'a göre $\alpha_i \neq 0$ olan her x_i giriş verisi destek vektörü olarak adlandırılmaktadır.

3.4.2.2. Doğrusal Olmayan Destek Vektör Makinaları

Gerçek problemlerin birçoğu doğrusal bir karar yüzeyi ile hatasız olarak ayrılamaz. Doğrusal olmayan bu problemlerde çözümü bulmanın yolu, çekirdek işlevleri ile örneklerin öncelikle daha yüksek boyutlu ve doğrusal olarak ayrılabilirleri bir uzaya taşınıp çözümün bu yeni uzayda aranmasıdır. Şekil 3.39'da görüldüğü gibi doğrusal olarak ayrılamayan giriş uzayı ayrılabilir duruma gelmektedir (Vapnik, 1998; Burges, 1998; Abe, 2005; Zheng ve Xue, 2009).



Şekil 3.39 Giriş uzayından özellik uzayına verinin aktarılması

$\phi(x_i) = \phi_1(x_i), \phi_2(x_i), \dots, \phi_l(x_i)$, $i = 1, \dots, l$, doğrusal olmayan haritalama işlevi kullanılarak l boyutlu x giriş vektörü, m boyutlu özellik uzayına haritalanmaktadır. Dönüşüm sonucunda, özellik uzayında doğrusal karar fonksiyonu Denklem (3.101) daki biçimde yazılmaktadır. (Vapnik, 1998; Burges, 1998; El-Naqa, 2002; Abe, 2005; Zheng ve Xue, 2009).

$$f(x) = w^T \phi(x) + b \quad (3.101)$$

Denklem(3.101)'de w , l boyutlu ağırlık vektörünü, b eşik değerini göstermektedir. Buna göre en büyük sınıra sahip en uygun hiper-düzlem Denklem(3.103)'de verilen kısıtlamalar altında $J(w, \xi)$ fonksiyonu en küçük yapılarak bulunabilmektedir.

$$J(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \quad (3.102)$$

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \quad \xi_i \geq 0, \quad i = 1, \dots, l \quad \text{için} \quad (3.103)$$

Lagrange çarpımları yöntemi kullanılarak KKT koşullarına göre ağırlık vektörü, $\phi(x_i)$ dönüşüm işlevinin doğrusal birleşimi olarak Denklem(3.104)'deki gibi yazılmaktadır.

$$w = \sum_{i=1}^l \alpha_i y_i \phi(x_i) \quad (3.104)$$

En uygun hiper-düzlemi bulmak için yüksek boyutlu özellik uzayında iç çarpımın tanımlanması gerekmektedir. w , Denklem(3.101)'de verilen karar fonksiyonunda yerleştirildiğinde Denklem(3.105)' deki biçim elde edilir.

$$f(x) = \sum_{i=1}^l \alpha_i y_i \phi^T(x_i) \phi(x) + b = \sum_{i=1}^l \alpha_i y_i K(x_i, x) + b \quad (3.105)$$

Yüksek boyutlu öznelik uzayındaki $K(x_i, x) = (\phi(x_i)^T \phi(x))$ iç çarpımının hesaplanması çok güç olacağı için bunun yerine düşük boyutlu giriş uzayında, çekirdek işlevinin hesaplanması yeterli olacaktır. Çekirdek işlevi kullanılarak Denklem(3.107) eşitliğinde verilen kısıtlamalar altında Lagrange işlevi çözülerek Lagrange çarpımları hesaplanabilmektedir.

$$L(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (3.106)$$

$$\sum_{i=1}^l y_i \alpha_i = 0 \quad \text{ve} \quad 0 \leq \alpha_i \leq C \quad i = 1, \dots, l \quad (3.107)$$

Denklem(3.106)'ya göre 1 destek vektör sayısını, $\alpha_i \neq 0$ olan her x_i giriş verisi ise destek vektörünü belirtir.

Literatürde farklı alanlarda çekirdek fonksiyonlarının sınıflandırma başarımını değerlendirmek üzerine bazı çalışmalar yapılmıştır. Bu çalışmalarda yüksek sınıflandırma başarımının genellikle radyal tabanlı çekirdek işlevleri kullanılarak elde

edildiği belirtilmiştir (Schölkopf ve ark. , 1997). Çünkü , radyal tabanlı çekirdek işlevi, seçilen parametre aralıklarına bağlı olarak, hem sigmoid çekirdek işlevi hem de doğrusal çekirdek işlevinin özelliklerini gösterebilmektedir (Keerthi ve Lin, 2003).

Destek vektör makinelerinde yaygın olarak kullanılan çekirdek işlevleri şunlardır (Abe, 2005; Zheng ve Xue, 2009):

Doğrusal Çekirdek İşlevi: Bu çekirdek işlevi giriş verisinin doğrusal olarak ayrılabilirdiği durumlarda kullanılmaktadır.

$$K(x_i, x) = x_i^T x \quad (3.108)$$

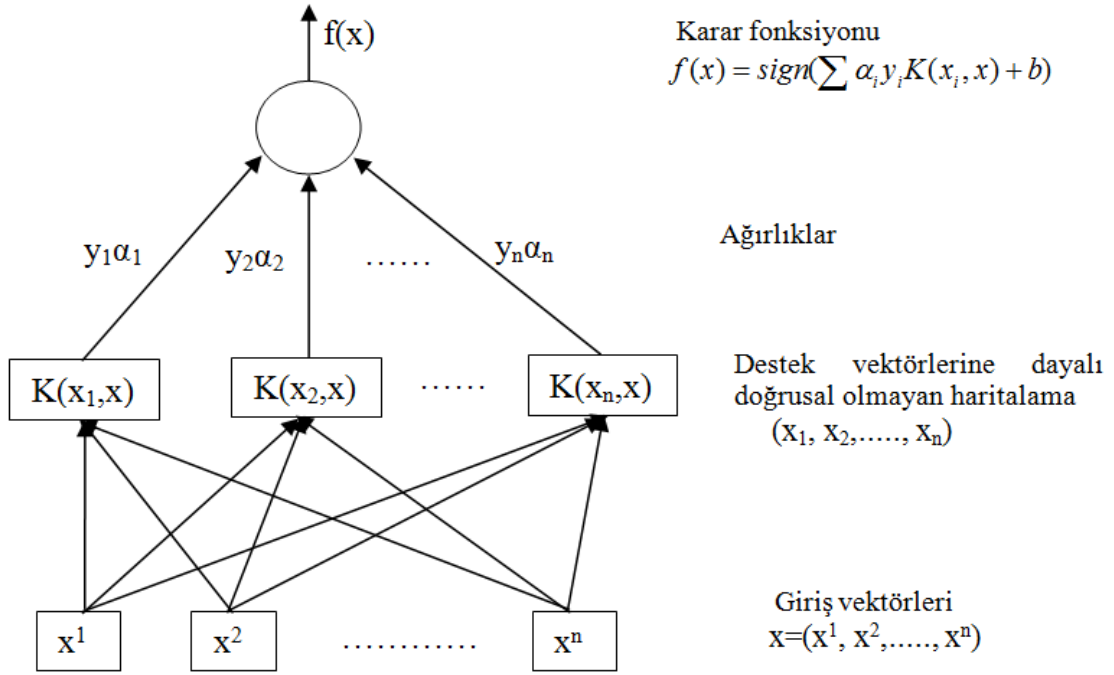
Çok terimli çekirdek fonksiyonu: p çekirdek derecesini göstermekte olup işlev Denklem(3.109) ile verilmiştir.

$$K(x_i, x) = (x_i^T x + 1)^p \quad (3.109)$$

Radyal Tabanlı (RT) çekirdek işlevi: γ yarıçapı kontrol eden pozitif değerli bir parametre olup işlev Denklem(3.110) ile verilmiştir.

$$K(x_i, x) = \exp(-\gamma \|x_i - x\|^2) \quad (3.110)$$

Doğrusal olmayan bir DVM sınıflandırıcının ağ yapısı Şekil 3.40'da verilmektedir.



Şekil 3.40 DVM Ağ yapısı (Vapnik, 1998)

DVM temelde ikili sınıflandırma için tasarlanmış bir yöntemdir. Çoklu DVM sınıflandırmanın yapılması için ikili sınıflandırma temeline dayanan birçok yöntem vardır. En yaygın kullanılan çoklu sınıflandırma yöntemleri (Hsu ve Lin, 2002):

Bire-Karşı-Bir (BKB) DVM

Bire-Karşı-Diğerleri (BKD) DVM

BKB yönteminde, k adet sınıf varsa $k(k-1)/2$ adet sınıflandırıcı yapısı oluşturulur ve oluşan ikili sınıflandırma probleminin çözümü aranır.

$$\min_{w^{ij}, b^{ij}, \xi^{ij}} \frac{1}{2} (w^{ij})^T (w^{ij}) + C \sum_t \xi_t^{ij} (w^{ij})^T \quad (3.111)$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij} \quad \Leftarrow \quad y_t = i \text{ ise} \quad (3.112)$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij} \quad \Leftarrow \quad y_t \neq i \text{ ise}$$

$$\xi_t^{ij} \geq 0$$

Çoklu sınıflandırma yöntemlerinden en eski kullanılan yöntem BKD yöntemidir. Sınıf sayısı k olmak üzere k adet DVM yapısı oluşturulur. i . sınıfa ait bütün örnekler pozitif diğer bütün örnekler negatif etiketlenerek i . DVM eğitimi Denklem(3.113), Denklem(3.114)’deki gibi yapılır.

$$\min_{w^i, b^i, \xi^i} \frac{1}{2} (w^i)^T (w^i) + C \sum_{j=1}^l \xi_j^i (w^i)^T \quad (3.113)$$

$$(w^i)^T \phi(x_j) + b \geq 1 - \xi_j^i \quad \Leftarrow \quad y_j = i \text{ ise} \quad (3.114)$$

$$(w^i)^T \phi(x_j) + b \leq -1 + \xi_j^i \quad \Leftarrow \quad y_j \neq i \text{ ise}$$

$$\xi_j^i \geq 0 \quad j = 1, \dots, l$$

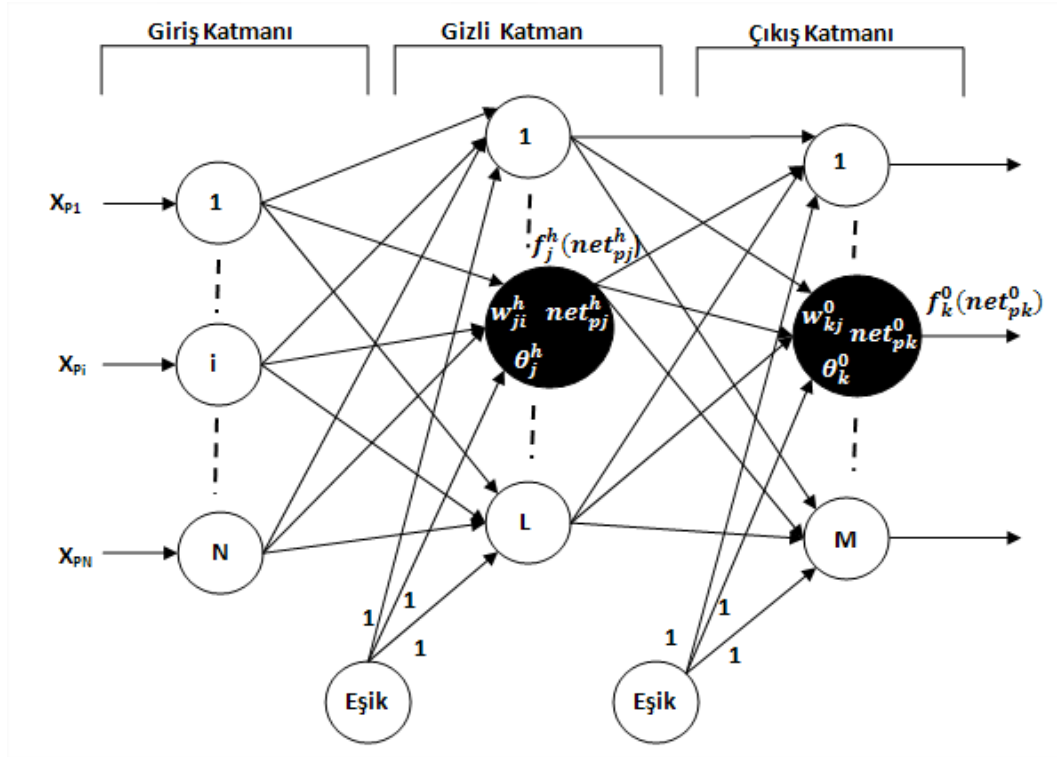
3.4.3. Çok Katmanlı Yapay Sinir Ağı

Bu ağlar giriş katmanı, çıkış katmanı ve en az bir gizli katman olmak üzere üç katmandan oluşur. Gizli katman ve gizli katmandaki düğüm sayısı değiştirilebilir. Düğüm sayısının artması ağın hatırlama yeteneğini artırmakla birlikte öğrenme süresini uzatmaktadır. Bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilmektedir. Giriş katmanı, dış ortamlardan aldığı bilgileri değişikliğe uğratmadan gizli katmandaki birimlere ileterek, bilgi gizli katmanda ve çıkış katmanında işlenerek ağın çıkışı belirlenmektedir (Haykin, 1999). Şekil 3.41’de N girişli M çıkışlı ve L sinir hücreli h adet gizli katmandan oluşan bir ÇKYSA yapısı verilmiştir. Bu ağlarda kullanılabilen aktivasyon işlevleri ise Şekil 3.42 ile verilmiştir.

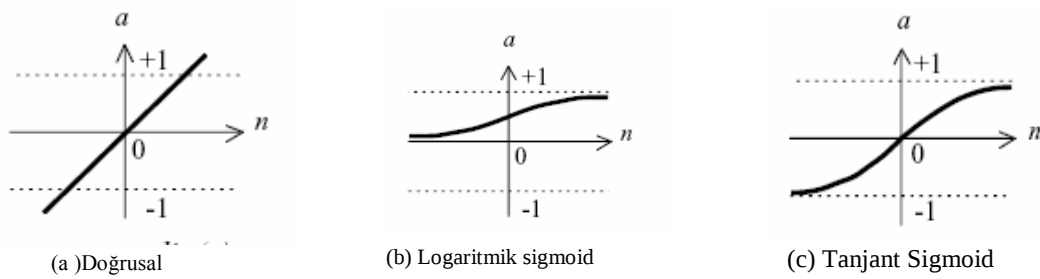
Yapay sinir ağlarının eğitilmesinde en yaygın kullanılan algoritma Geriye Yayılım Algoritması’dır. Geriye yayılım öğrenme yöntemi, delta öğrenme kuralı veya en küçük kareler yöntemi olarak bilinen kuralda olduğu gibi sistem hatasını azaltma esasına dayanan bir eniyileme işlemidir. Bu hatayı azaltma işlemini de ağırlıkları değiştirmek suretiyle yaptığı için bir başka deyişle çıkış hatasını geriye yaydığı için “Geriye Yayılım” olarak adlandırılmaktadır. Geriye yayılım iki aşamadan oluşmaktadır.

İleri doğru hesaplama : Ağın çıktısının hesaplandığı aşamadır.

Geriye doğru hesaplama: Ağırlıkların değiştirildiği aşamadır.



Şekil 3.41 Çok Katmanlı Yapay Sinir Ağı Örneği



Şekil 3.42 Aktivasyon işlevleri

Geriye yayılım öğrenme kullanıldığında çıkış katmanındaki hataları kullanılarak gizli katmanın ağırlıkları ayarlanır. Bu işlem ilk gizli katmana kadar aynı biçimde yinelenir. Bu yolla hatalar katman katman ilgili katmanın ağırlık düzenlemeleri

yapılarak geriye doğru yayılır ve toplam hata en aza indirilinceye kadar bu işlemler tekrarlanır.

Şekil 3.41 ile verilen ağ yapısına göre h. Katmana ait j. yapay sinir hücresine ait çıkışı Denklem(3.115) ile verildiği biçimde hesaplanır (Freeman ve Skapura, 1991).

$$net_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h \quad (3.115)$$

Bu denklemde w_{ji}^h i. giriş biriminden h. gizli katmanın j. birimine gelen ağırlıklar, θ_j^h ise h. katmanın j. birime ait eşik değeri olarak tanımlanır. i_{pj}^h gizli katman birim çıkışları olmak üzere net girişinin aktivasyon işlevinden geçirilmesi ile elde edilir (Denklem(3.116)).

$$i_{pj}^h = f_j^h(net_{pj}^h) \quad (3.116)$$

k. çıkış birimi için Denklem (3.118) elde edilir :

$$net_{pk}^0 = \sum_{j=1}^N w_{kj}^0 x_{pj} + \theta_k^0 \quad (3.117)$$

$$o_{pk} = f_k^0(net_{pk}^0) \quad (3.118)$$

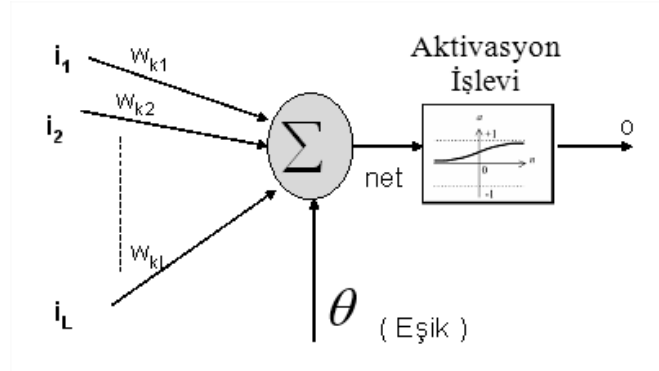
İleri beslemeli bir ağın p. Eğitim vektörü için n. yinelemeye ait başarımlar işlevi Denklem(3.119) ile verilmiştir. Burada e_{pk} hata, M ağın çıkış katmanındaki bütün nöronların sayısıdır.

$$E_p(n) = \frac{1}{2} \sum_{j=1}^M e_k^2(n) \quad (3.119)$$

y_{pk} istenen çıkış, o_{pk} ağ çıkışı olmak üzere e_{pk} hatası Denklem(3.120) ile verilir.

$$e_{pk}(n) = y_{pk}(n) - o_{pk}(n) \quad (3.120)$$

Bazı hata değerlerinin negatif olma ihtimaline karşı ağırlık toplam hatasını bulmak için toplam karesel hata hesaplanmaktadır. Eğitimin amacı, ağ parametrelerini ayarlayarak, hatanın enerjisi olarak ta adlandırılan toplam karesel hata değerini (E) en aza indirmektir. Bu amaçla kullanılan eğitim algoritmalarından biri olan geriye yayılım algoritması hatayı geriye doğru hesaplama aşamasında ağırlıkları değiştirerek azaltma esasına dayanır. Bu amaçla hatanın enerjisinin ağırlıklara göre negatif türevini alır ve öğrenme parametresi ile çarparak bir önceki ağırlıklara ekler. Bu eşitlik, hatanın ağırlıklara göre türevi söz konusu olduğu için parçalı türev alınarak elde edilir. Şekil 3.43 ile verilen bir çıkış birim için bu türev Denklem(3.121)'deki biçimdedir.



Şekil 3.43 Çıkış katına ait bir birimin yapısı

$$-\frac{\partial E_p}{\partial w_{kj}^o} = (y_{pk} - o_{pk}) \underbrace{\frac{\partial f_k^o}{\partial (net_{pk}^o)}}_{f_k^o'(net_{pk}^o)} \frac{\partial (net_{pk}^o)}{\partial w_{kj}^o} \quad (3.121)$$

Ağın ağırlıklarını değiştirmek için iki durum söz konusudur.

- Gizli katman ile çıktı katmanı arasındaki ağırlıkların değiştirilmesi
- Gizli katmanlar arası veya gizli katman ile girdi katmanı arasındaki ağırlıkların değiştirilmesi

Çıktı katmanı ile gizli katman arasında elde edilen Denklem(3.121)'den:

$$\underbrace{\frac{\partial(net_{pk}^o)}{\partial w_{kj}^o}} = \left(\frac{\partial}{\partial w_{kj}^o} \sum_{j=1}^L w_{kj}^o i_{pj} + \theta_k^o \right) = i_{pj} \quad (3.122)$$

elde edilir ve yeniden düzenlenerek η öğrenme parametresi adı verilen bir sabitle çarpılırsa çıkış ağırlıklarının değişimi Denklem (3.123)'deki biçimdedir.

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta(y_{pk} - o_{pk}) f_k^{o'}(net_{pk}^o) i_{pj} \quad (3.123)$$

Benzer biçimde parçalı türev gizli katmanlar arası ve gizli katman ile giriş katmanı arasında Denklem(3.124)'de verildiği gibi alınır.

$$\frac{\partial E_p}{\partial w_{ji}^h} = - \sum (y_{pk} - o_{pk}) \underbrace{\frac{\partial o_{pk}^o}{\partial(net_{pk}^o)}}_{f_k^{o'}(net_{pk}^o)} \underbrace{\frac{\partial(net_{pk}^o)}{\partial i_{pj}^o}}_{w_{kj}^o} \underbrace{\frac{\partial i_{pj}^o}{\partial(net_{pj}^h)}}_{f_j^{h'}(net_{pj}^h)} \underbrace{\frac{\partial(net_{pj}^h)}{\partial w_{ji}^h}}_{x_{pi}} \quad (3.124)$$

Ağırlıkların değişimi Denklem(3.125) ile verildiği biçimde düzenlenebilir.

$$\Delta_p w_{ji}^h = \eta \cdot f_j^{h'}(net_{pj}^h) x_{pi} \sum_k \underbrace{(y_{pk} - o_{pk}) f_k^{o'}(net_{pk}^o)}_{\delta_{pk}^o} w_{kj}^o \quad (3.125)$$

Ara katmanın ağırlık güncellemesi yine eski ağırlıklara bu değişimin eklenmesiyle Denklem(3.126) kullanılarak hesaplanır.

$$\delta_{pj}^h = f_j^{h'}(net_{pj}^h) \sum_k \delta_{pk}^o w_{kj}^o \quad (3.126)$$

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_{pj}^h x_i$$

Geriye yayılım yöntemi, öğrenme oranını değiştirme, momentumu kullanma ve değişkenleri ölçekleme ile iyileştirmelere rağmen deneme-yanılma içermesi ile pek çok problemin çözümünde yavaş kalmaktadır. Bir yapay sinir ağına ait ağırlıkların eğitilmesinde en etkin algoritmalarından biri standart optimizasyon teknikleri içeren Levenberg - Marquardt (LM) algoritmasıdır (Hagan ve Menhaj, 1994; Wilamowski ve ark. , 1999; Kermani ve ark. , 2005).

Başarım işlevinin w ağırlıklarına göre ikinci dereceden türevlerinin oluşturduğu Hessian matrisi ağırlık uzayının farklı doğrultularındaki gradyan değişimini gösterir ve Denklem(3.127) ile verildiği biçimde hesaplanır.

$$H(n) = \frac{\partial^2 E(n)}{\partial w^2(n-1)} \quad (3.127)$$

Hessian matrisi çok karmaşık ve ileri beslemeli bir yapay sinir ağı için hesaplanması zor bir matristir. Jakobien matrisi ise ağ ağırlıklarının birinci dereceden türevlerini içeren bir matris olduğu için hesaplanması daha kolaydır. Bu nedenle Jakobien matrisi ile elde edilen Hessian matrisi kabul edilebilir bir yaklaşıklıktadır ve Denklem(3.128) ile verilmiştir.

$$H(n) = JK^T(n)JK^T(n) + \gamma.I \quad (3.128)$$

Burada γ Marquardt parametresi, I, birim matris, J matrisi, Jakobien matrisi olarak adlandırılır .

$$JK(n) = \frac{\partial e(n)}{\partial w(n-1)} \quad (3.129)$$

Hataya bağılı olarak ağırlık gradyanı Denklem(3.130) ve hatayı en aza indirecek ağırlık parametreleri Denklem(3.131) ile hesaplanır.

$$\mathcal{J}(n) = JK^T(n)e(n) \quad (3.130)$$

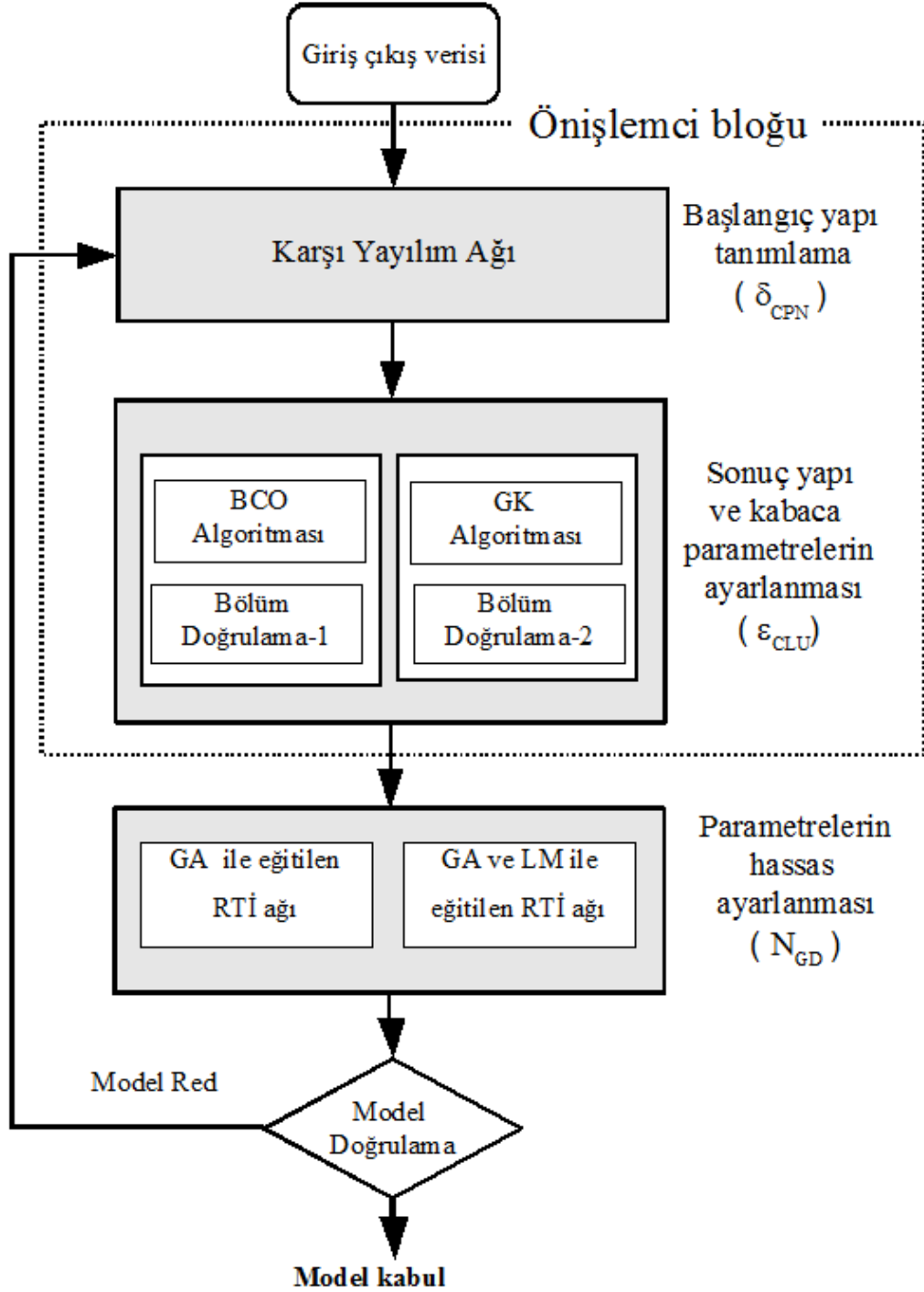
$$w(n+1) = w(n) - [H(n)]^{-1} \mathcal{J}(n) \quad (3.131)$$

3.4.4. Melez Yapı Önerileri

RTİ ağlarının merkez, genişlik ve ağırlık parametrelerinin saptanmasında ve eğitiminde farklı yapılar literatürde kullanılmaktadır. Bu çalışmalarda RTİ ağındaki radyal işlev parametrelerini saptamada genellikle kümeleme algoritması kullanılmaktadır(Montazer ve ark. , 2009; Staiano ve ark. , 2006).

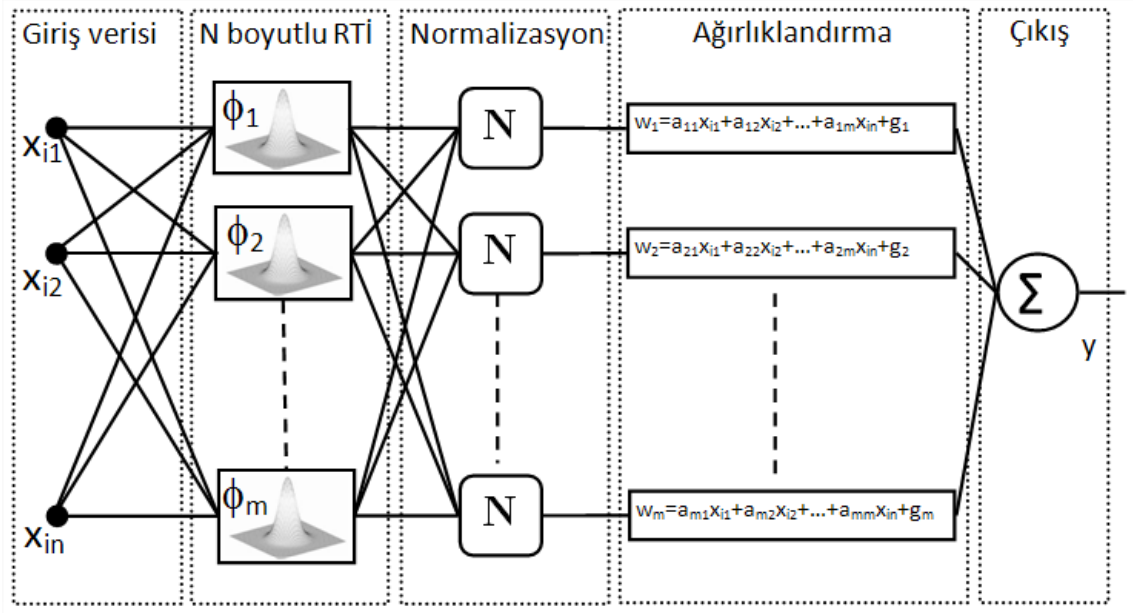
BCO ve GK algoritmaları veri setleri içindeki kümeleri saptamada oldukça başarılıdır. Her iki algoritmanın da eksik olan en önemli yönü başlangıçta küme sayısına ihtiyaç duymasıdır. KYA algoritması kullanıcının belirlediği yarıçapa bağlı olarak ham veriden optimal olmayan sayıda küme bulan bir algoritmadır. Çok kaba bir kümeleme yapan bu algoritma optimal olmasa bile veri içindeki kümeleri saptayabilme özelliğiyle BCO ve GK algoritmalarında küme sayısının bilinmesi sorununu ortadan kaldırmaktadır. Bu durumdan hareketle KYA algoritmasıyla BCO veya GK algoritmaları tümleştirildiği bir yapı tanımlanmıştır. BCO algoritması sadece dairesel dağılımlı bir kümeleme yapmaktadır. Veriler ise her zaman dairesel bir dağılım sergilemek durumunda değildir. GK kümeleme algoritması eliptik kümelerde de başarımlı sağlama nedeniyle iyi bir seçenek olarak görülmüştür. Küme sayılarını optimize etmek için algoritmalar bölüm doğrulama algoritmaları ile tümleşik bir yapıya dönüştürülmüştür. Bu yapı, veri seti için yine de kaba bir modeldir. Ancak merkez, genişlik ve ağırlık parametreleri rastgele girilen bir RTİ ağı için çok iyi bir önışlemci bloğudur. Çünkü BCO ve GK algoritmaları bu parametreleri veri setinden elde edebilen algoritmalar. Önerilen bu yapıya ait blok şema Şekil 3.44'de verilmiştir. BCO ve

GK yapılarının başarımları, GA algoritmasıyla eğitilen sabit ağırlıklı bir RTİ ağı kullanılarak literatürde kullanılan iki örnek üzerinde sınanmıştır. Bu yapı sadece BCO ve GK algoritmalarının başarımlarını karşılaştırmak amacıyla kullanılmıştır.



Şekil 3.44 Melez yapı önerisi

Bu blok şema ile verilen ve ağıın başarımını artırmak için eğitim aşamasında GD algoritmasının yerine GD ve LM algoritmaları kullanan yapı DANRTİ olarak adlandırılmıştır. Şekil 3.45 ile şematik olarak verilen bu yapı bir MATLAB GUI olarak tasarlanmıştır.



Şekil 3.45 DANRTİ Sınıflandırıcısının şematik gösterimi

Şekil 3.45 ile gösterilmiş olan sınıflandırıcı yapısı, ANFIS (Adaptive Neuro Fuzzy System) (Jang ve ark. , 1997), ağ yapısına benzemekle birlikte önemli bir avantajlara sahiptir. ANFIS ağ yapısı giriş verisinin her bir boyutu için bir üyelik işlevi bulunmaktadır. Bu nedenden dolayı yüksek boyutlu giriş verileri kural sayısında üstel bir artışa neden olmaktadır. DANRTİ yapısında ise kural sayısını belirleyen önışlemci bloğudur. DANRTİ yapısında öğrenme aşamasında ağ iki aşamalı çalışmaktadır.

- Merkez ve genişlikler GA algoritması ile eğitilmektedir.
- Ağırlıklar ise LM algoritması ile eğitilmektedir.

Şekil 3.45 ile gösterilmiş olan sınıflandırıcı yapısında RTİ birim çıkışı Denklem(3.132) ile verildiği gibi tanımlanırsa,

$$\phi_j = e^{\frac{(v_{jn}-x_{iz})^2}{\sigma_{jn}^2}} \quad j = 1, \dots, m \quad \text{ve} \quad z = 1, \dots, n \quad (3.132)$$

Normalize ağ çıkışı Denklem(3.134) ile verildiği biçimde tanımlanır.

$$y = \frac{w_1\phi_1 + w_2\phi_2 + \dots + w_m\phi_m}{\phi_1 + \phi_2 + \dots + \phi_m} \quad (3.133)$$

Merkez ve genişliklerin GA ile eğitilmesi için hatanın enerjisinin merkezlere ve genişlikleri göre negatif türevleri alınarak sırasıyla Denklem(3.134) ve Denklem(3.135)'de verildiği biçimde elde edilir.

$$\begin{aligned} \Delta v &= -\frac{\partial E}{\partial v_{jz}} \\ &= \frac{(d - y) \phi_j (v_{jz} - x_{jz}) [(w_1\phi_1 + \dots + w_j\phi_j) - w_j(\phi_1 + \dots + \phi_j)]}{\sigma_{jz}^2 (\phi_1 + \dots + \phi_j)^2} \end{aligned} \quad (3.134)$$

$$\begin{aligned} \Delta \sigma &= -\frac{\partial E}{\partial \sigma_{jz}} \\ &= \frac{(d - y) \phi_j (v_{jz} - x_{jz})^2 [(w_1\phi_1 + \dots + w_j\phi_j) - w_j(\phi_1 + \dots + \phi_j)]}{\sigma_{jz}^3 (\phi_1 + \dots + \phi_j)^2} \end{aligned} \quad (3.135)$$

Ağırlıkların güncellenmesinde kullanılacak olan LM algoritması için Jakobien matrisi Denklem(3.136) ile verildiği biçimde hesaplanır.

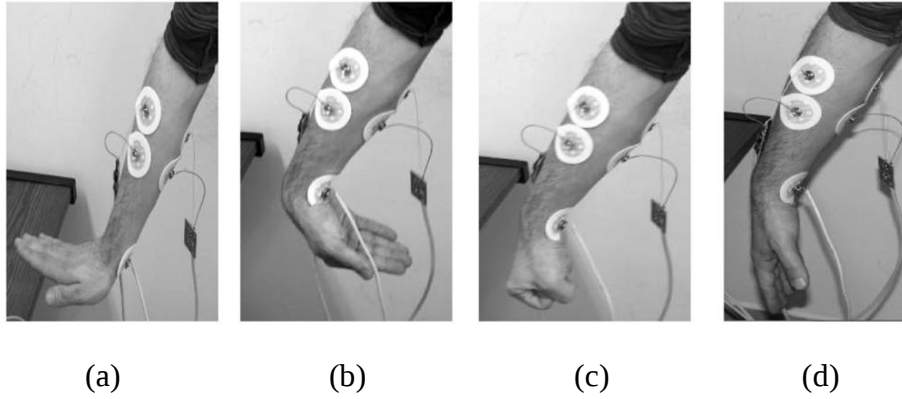
$$JK = \begin{bmatrix} -\frac{y \cdot x_{11}}{\phi_1 + \dots + \phi_m} & \dots & -\frac{y \cdot x_{1n}}{\phi_1 + \dots + \phi_m} & -\frac{y}{\phi_1 + \dots + \phi_m} \\ \vdots & \dots & \vdots & \vdots \\ -\frac{y \cdot x_{i1}}{\phi_1 + \dots + \phi_m} & \dots & -\frac{y \cdot x_{in}}{\phi_1 + \dots + \phi_m} & -\frac{y}{\phi_1 + \dots + \phi_m} \end{bmatrix} \quad (3.136)$$

3.5. GERÇEK ZAMANLI DENETİM UYGULAMALARI

Bu bölümde iki kanallı ve dört kanallı EMG işaretleri kullanılarak elektromekanik elin denetlenmesi için gerekli algoritmik aşamaların gerçekleştirilmesi yapılmıştır.

3.5.1. İki kanallı EMG ile 4 farklı hareketin gerçek zamanlı denetimi

Dört farklı el hareketi için EMG işaretlerinin iki kanallı olarak kaydedilmesi yeterlidir. Seçilen bu dört farklı el hareketi için EMG işareti Şekil 3.46'da görüldüğü gibi deri yüzeyinden aktif elektrotlar kullanılarak alınmıştır ve deneysel bir elektromekanik elin gerçek zamanlı olarak denetimi gerçekleştirilmiştir (Özdemir ve ark. , 2009) .



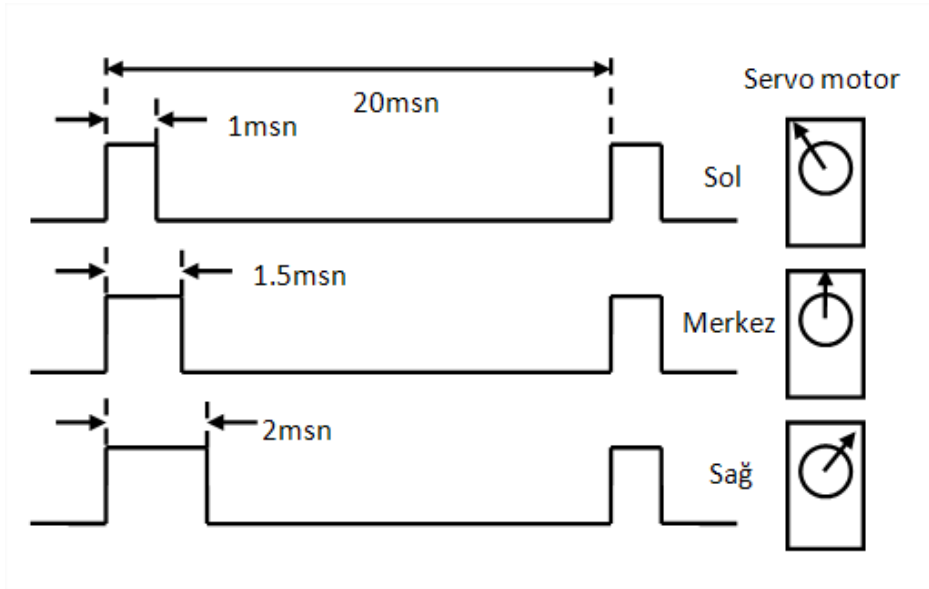
Şekil 3.46 Sınıflandırılan dört el hareketi

(a) Yukarı (b) Aşağı (c) Yumruk (d) Serbest

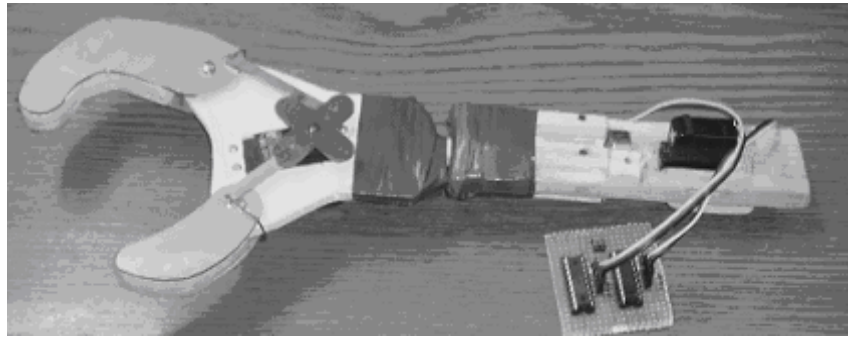
EMG işaretinden denetim işaretinin üretilmesi ise aşağıda belirtilen adımlardan gerçekleştirilmiştir.

1. İki kanallı EMG işaretinin kayıt edilmesi ve gürültüden arındırılması,
2. EMG işaretinin doğrultulması,
3. Alçak geçiren filtre ile EMG işaretinin zarfının çıkarılması,
4. EMG işaretinin enerji seviyesinin hesaplanması,
5. Her bir kanaldaki enerji seviyesinin önceden belirlenmiş eşik değerleri ile karşılaştırılması,
6. Karşılaştırma sonuçlarından, kasların aktivitesinin mantık (1 ve 0) değerleri ile etiketlenmesi.

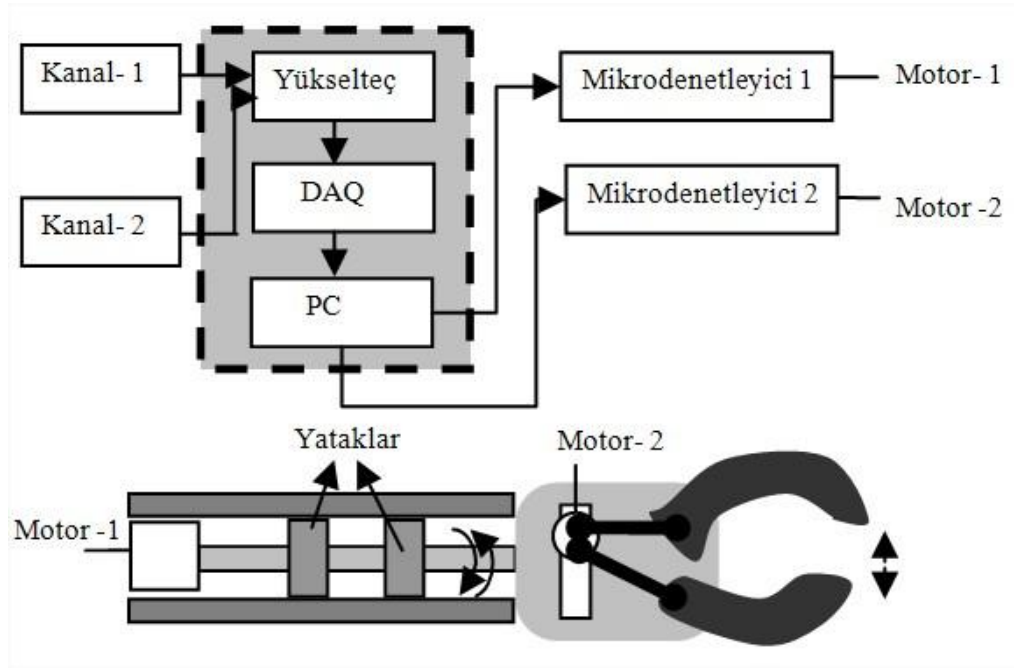
Mikrodenetleyici devresi tarafından elektromekanik eldeki iki servo motoru süren Darbe Genişlik Bindirimi (Pulse Width Modulation, PWM) işaretleri üretilmiştir. Mikrodenetleyici olarak kullanılan PIC16F84A tümdevresi Şekil 3.47 ile verildiği biçimde Darbe Genişlik Bindirimi (DGB) denetleme işaretlerini üretmektedir. Şekil 3.48’de mikrodenetleyici ile denetlenen yapay el ve Şekil 3.49’da ise tasarlanan sistemin şematik gösterimi verilmiştir.



Şekil 3.47 DGB işaretleri ve motor milinin konumu



Şekil 3.48 Elektro-mekanik protez el ve iki tane 16F84A mikro denetleyicili sürücü devre



Şekil 3.49 Tasarlanan sistemin şematik gösterimi

3.5.2. İki kanallı EMG verisi ile tasarlanan sınıflandırıcının sınanması

EMG işaretlerinin mümkün olduğunca gürültüsüz bir biçimde alınmasından sonra yapılacak işlem EMG verilerinden özellik çıkarılması ve çıkarılan düşük boyutlu ve yüksek sınıf ayrılabilirliğine sahip EMG özelliklerinin sınıflandırılmasıdır. EMG verisinden özellik vektörü çıkarım aşamaları aşağıdaki biçimde özetlenebilir.

1- Kaydedilmiş ham EMG verisi DC ofsetten arındırılması.

2- DC ofsetten arındırılmış ham EMG verisi kayan pencere yöntemi ile analiz edilmesi. Bu aşamada pencere genişliği ve kayma miktarı 128 olarak belirlenmiştir. Bunun amacı yöntemin gerçek zamanlı olarak da çalıştırılabilir olmasıdır.

3- İşarete ADD uygulanması. Ayrık dalgacık dönüşümünün derinliği dört olarak seçilmiş ve analiz için “db4” dalgacığı kullanılmıştır. “db4” dalgacığının seçimi deneysel olarak yapılmıştır.

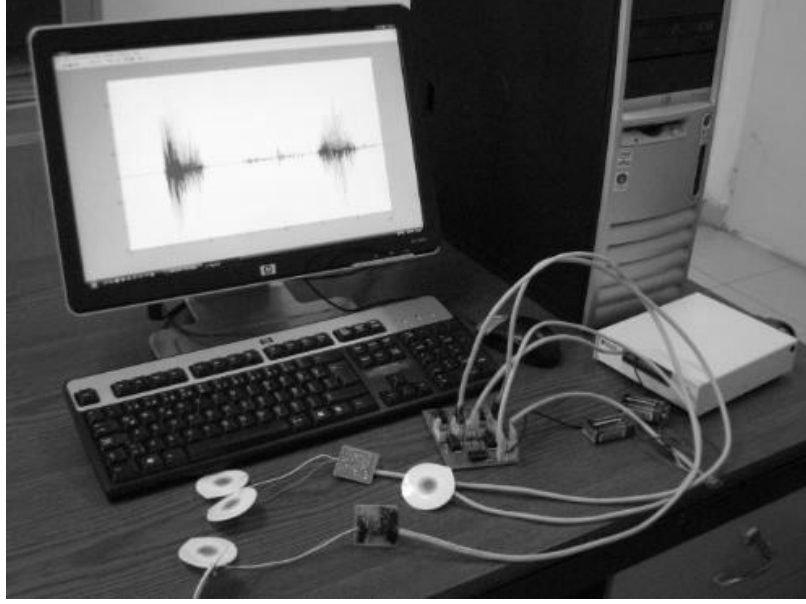
4- Boyut azaltma işleminin yapılması. Ayırık dalgacık dönüşümü ve boyut azaltma yöntemlerinden seçilen DDA her bir kanala ayrı ayrı uygulanmıştır.

İki kanallı olarak kaydedilen EMG işaretleri dört farklı hareket yapılarak kaydedilmiştir. Bu hareketler, serbest durum , elin yukarı bükülmesi (flexion), elin aşağı bükülmesi (extension) ve yumruk olup her biri iki saniye boyunca yapılmıştır. Örnekleme sıklığı 1KHz olup, her saniyede $2^{10} = 1024$ örnek alınmış ve toplamda kaydedilen EMG verisi $7 \times 2 \times 1024 = 14336$ adet örnekten oluşmuştur. Bu verinin doğrudan sınıflandırma işlemi için kullanılamayacağı açıktır. Bunun başlıca sebebi verinin oldukça fazla sayıda örnekten oluşmasıdır. Bunun yanı sıra verideki ani çöküşlerde sınıflandırma işlemini olumsuz yönde etkileyecektir.

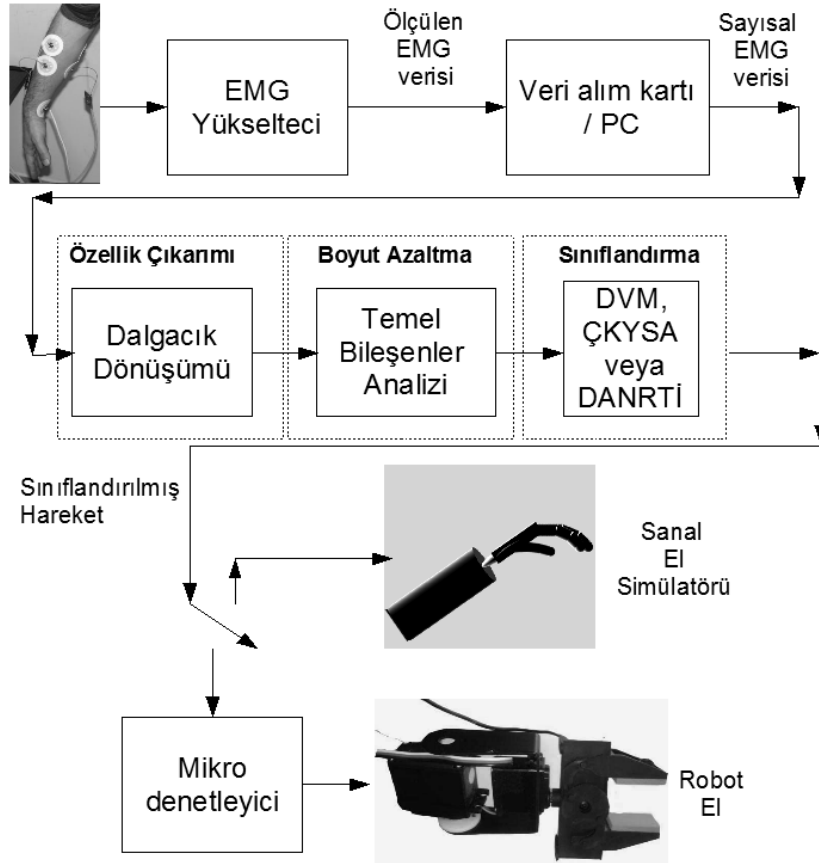
3.5.3. Dört Kanallı EMG İşareti ile Elektromekanik Elin Gerçek Zamanlı Denetlenmesi

EMG verisi dört kanallı olarak gürültüsüz biçimde kaydedilmiştir. Ham EMG verisinden özellik çıkarımı için Dalgacık Dönüşümü ve Temel Bileşenler Analizi (TBA) seçilmiştir. EMG örüntüleri doğrusal olmayan özellikler içerdiği için doğrusal doğası gereği DDA giriş uzayında her zaman iyi bir boyut azaltma yöntemi değildir. Bu nedenle EMG örüntüleri DDA uygulandıktan sonra hala iyi bir sınıf ayrılabilirliğine sahip olmayabilir(Liu ve ark. , 2007). Elde edilen özellikler DANRTİ ağ yapısı, DVM ve ÇKYSA yapıları tarafından sınıflandırılmıştır. Sınıflandırma işlemi sonucu üretilen denetim işareti, gerçek zamanlı elektromekanik (veya sanal) elin denetiminde kullanılmıştır. Ham EMG işaretlerinin sınıflandırılması ve protez elin sürülmesi için kullanılan genel yapı Şekil 3.51 ile gösterilmiştir.

Bu çalışmada, EMG yükselteci ile elde edilen bilgiler NI-SCB68 veri alım kartı ile sayısallaştırılarak bilgisayara aktarılmıştır. İşaret işleme algoritmaları için P4 2.4 GHz 2GB RAM özellikli bir bilgisayar kullanılmıştır. EMG verisinin elde edilmesi, özellik çıkarımı ve denetim işaretinin üretilmesi için kullanılan deneysel düzenek ise Şekil 3.50’de verilmiştir. Denetlemek için HS-85BB mikro servolarını kullanan Lynxmotion firmasına ait yedi hareketi yapabilme yeteneğine sahip elektromekanik el kullanılmıştır.



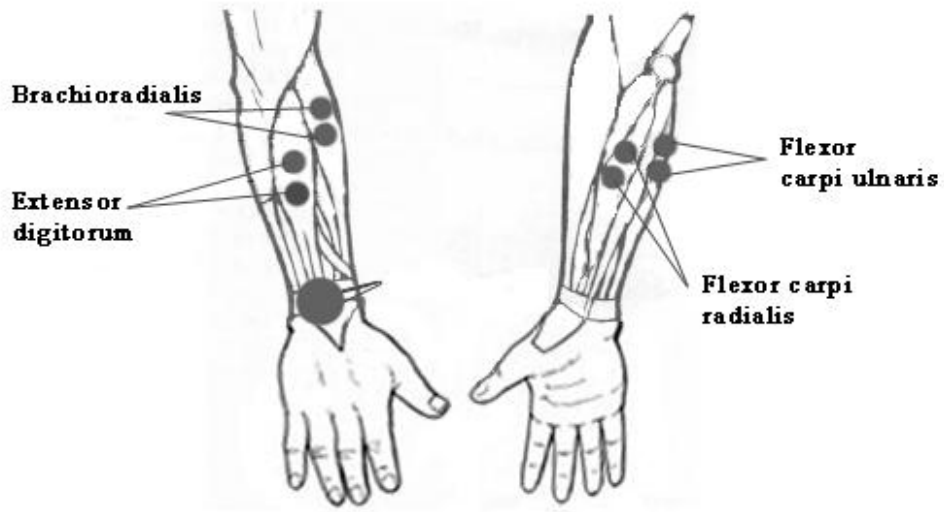
Şekil 3.50 EMG verisi alımında kullanılan deneysel düzenek



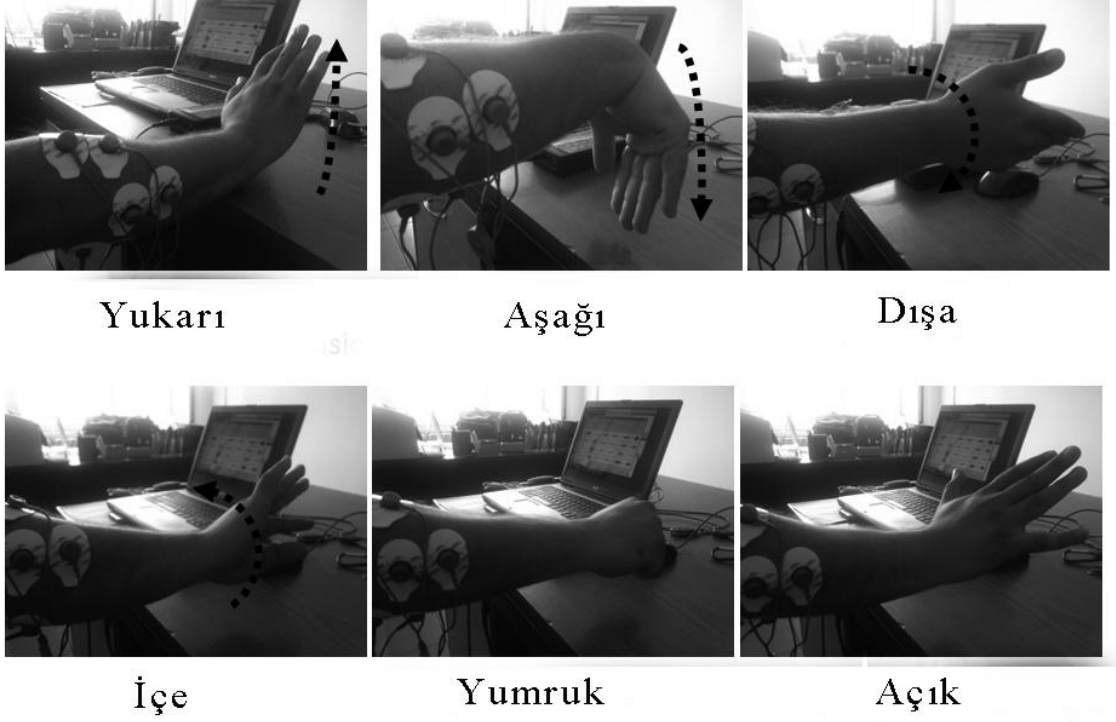
Şekil 3.51 EMG denetimli protez geliştirme düzeneğinin alt parçaları

3.5.3.1. Veri Alımı Aşaması

Şekil 3.53’de verildiği biçimde belirlenen 7 el hareketi için (yukarı, aşağı, içe döndürme, dışa döndürme, yumruk, açma ve serbest) 4 kanallı EMG ölçümleri 30 yaşında sağlam bir bireyden alınmıştır. Şekil 3.52’de verilen biçimde sağ önkolun “Brachioradialis”, “Flexor carpi ulnaris”, “Flexor carpi radialis”, “Extensor Digitorum” kaslarına bağlanan standart Ag/Ag-Cl yüzey elektrotları ile belirlenen 7 hareket yapılmıştır. Kurulan düzenek 2000Hz örnekleme sıklığı ile çalıştırılmış olup her bir hareket 2 sn süresince yapılmıştır.



Şekil 3.52 Önkol kaslarına yüzey elektrotlarının yerleşim şeması



Şekil 3.53 Belirlenen hareket seti

3.5.3.2. Verinin Ön-işleme Aşaması

Literatürde EMG işaretinden özellik çıkartımı için kullanılan pek çok yöntem (Mutlak değerlerin ortalaması, varyans, dalga biçimi uzunluğu vb.) mevcuttur (Tenore ve ark. , 2009). Bu çalışmada ise özellik çıkarım yöntemi olarak ADD kullanılmıştır.

100 ms uzunluğunda pencereler seçilerek ham EMG işaretleri işlenmiştir. Her bir kanala 4. dereceden dalgacık dönüşümü uygulanmıştır. Dalgacık dönüşümü sonrası her bir kanal için özellik vektörü olarak 4 adet detay ve 1 adet yaklaşım katsayıları vektörü seçilmiştir. Sınıflandırma işleminden önce, elde edilen bu çok boyutlu özelliklerin boyutlarının azaltılması gerekmektedir. Boyut azaltma işlemi için ise TBA yöntemi kullanılmıştır. TBA yöntemi her kanala bağımsız biçimde uygulanmıştır.

3.5.3.3. Özelliklerin Sınıflandırılması

Sınıflandırma algoritmalarının eğitilmesi için kullanılan özelliklerin elde edilmesi için aşağıda verilen işlem adımları uygulanmıştır:

1. Her bir hareket, her oturumda bir kez ve 2 saniye boyunca gerçekleştirilmiştir.

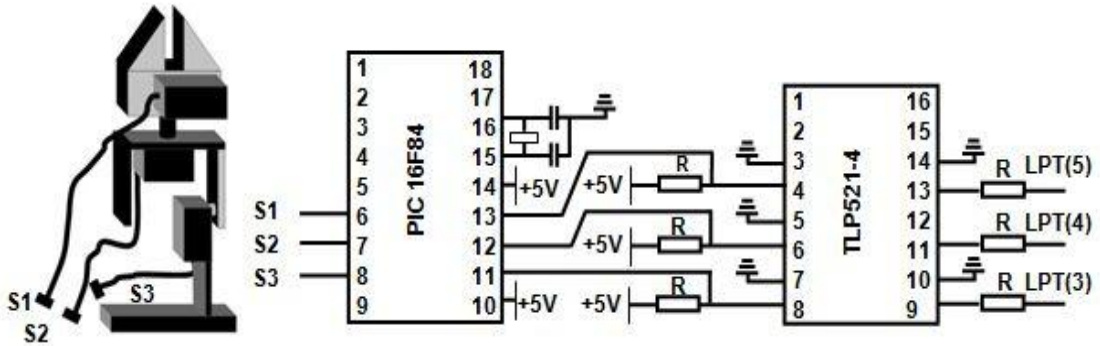
2. Örnekleme sıklığı 2 KHz seçilmiştir.
3. 100ms lik pencereler kullanıldığı için 2 saniye boyunca gerçekleştirilen her bir hareketten, bir oturumda 20 adet özellik elde edilmiştir.
4. Her hareket bir defa tekrarlandığı için, bir oturumdan toplam $7 \times 20 = 140$ adet özellik elde edilmiştir.
5. Veri seti 5 oturumdan elde edildiği için toplam $5 \times 140 = 700$ adet veriden oluşmaktadır.
6. Elde edilen 700 adet veriden 560 adet veri yöntemlerin eğitimi, tüm hareketleri içeren 70 adet veri ise yöntemlerin test edilmesi için kullanılmıştır.

Sınıflandırma aşamasında sırasıyla DANRTİ, ÇKYSA ve DVM yapıları kullanılmıştır.

3.5.3.4. Deneyisel (veya sanal) elektromekanik elin denetimi

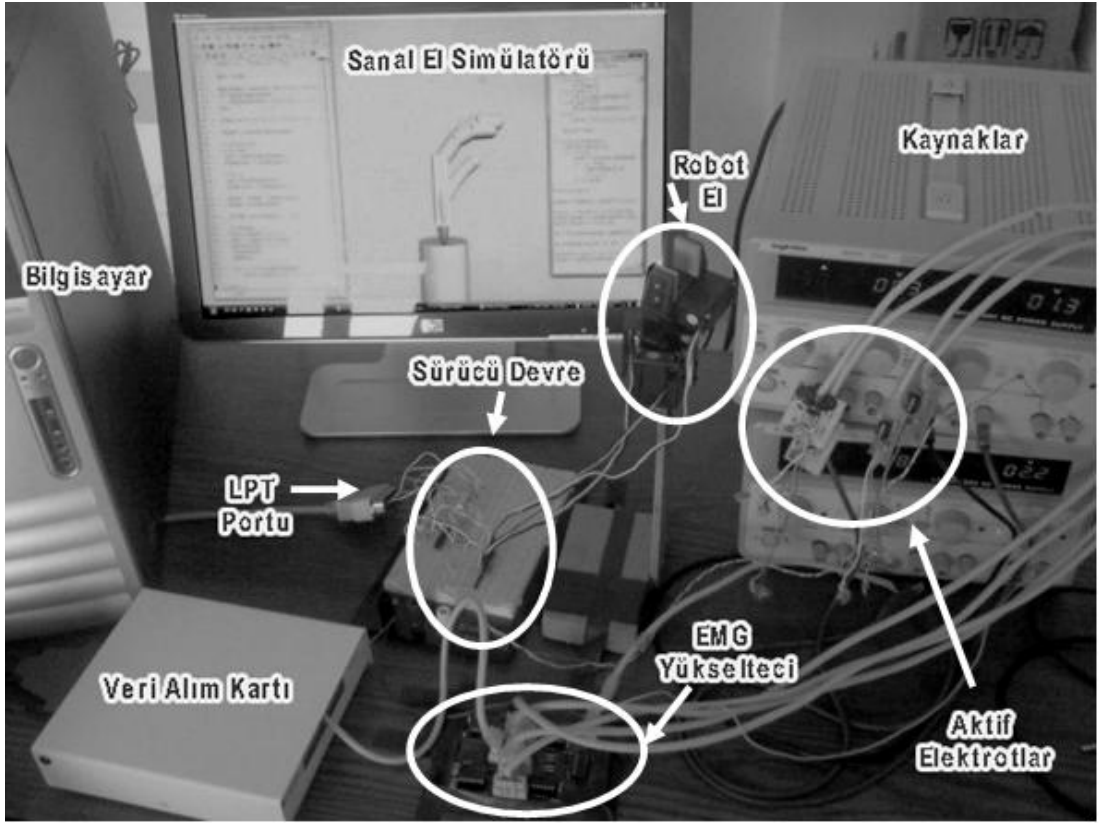
PIC16F84 devresi, paralel port aracılığı ile girişine yollanan 3 bitlik hareket bilgisine göre üretilen DGB işaretleri ile deneyisel elektromekanik eli sürmektedir.

Şekil 3.54 ile deneyisel elektromekanik eli DGB ile süren, sürücü devresi verilmiştir. PWM işaretleri PIC16F84 kullanılarak oluşturulmuştur. TLP521-4 optokuplör entegresi kullanılarak paralel port (LPT) ve sürücü devre elektriksel olarak birbirlerinden izole edilmiştir. Sürücü devresi, PIC yazılımında yapılacak küçük değişikliklerle birden fazla servo motoru aynı anda sürececek biçime dönüştürülebilir.



Şekil 3.54 Deneyisel elektromekanik elin sürülmesi

Tasarlanan gerçek zamanlı deneyisel EMG denetimli sanal / gerçek protez el geliştirme düzeneği Şekil 3.55'de verilmiştir.



Şekil 3.55 Gerçek zamanlı deneysel EMG denetimli sanal / gerçek protez el geliştirme düzeneği

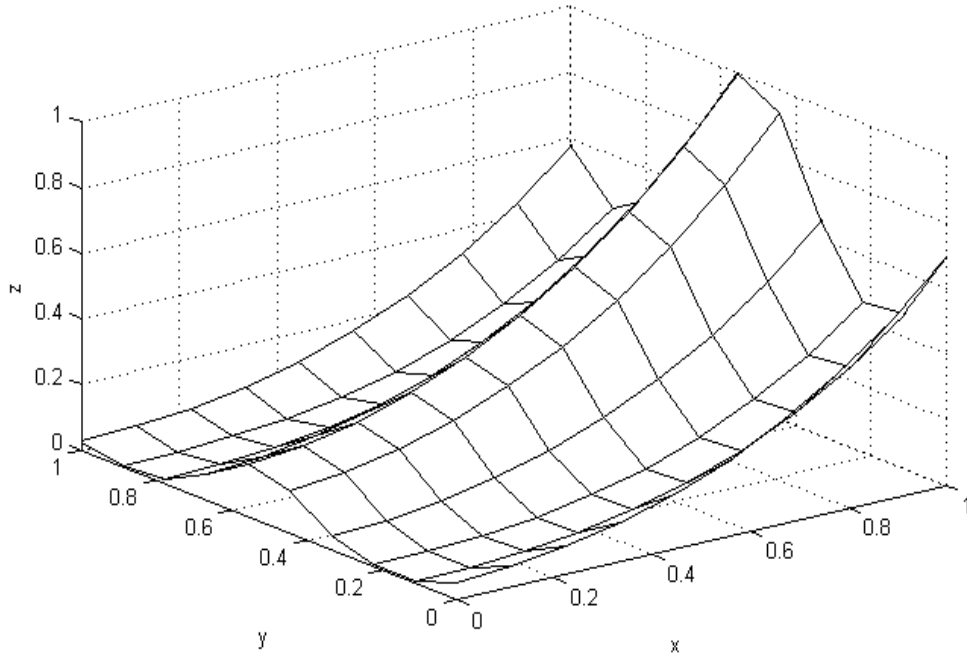
4. BULGULAR VE TARTIŞMA

4.1. BCO ve GK algoritmalarının sınıflandırıcı başarımlarının sınanması:

Literatürde kullanılan iki örnek veri seti için KYA ile bulunan küme sayısı BCO ve GK algoritmaları ile optimize edilerek elde edilen küme parametreleri bir RTİ nin parametreleri olarak saptanmıştır. GA algoritması ile eğitim sonucunda elde edilen bulgular verilmiştir.

Örnek 4.1 (Özdemir ve ark. , 2009; Kayhan ve ark. , 2009): Doğrusal olmayan çok girişli tek çıkışlı bir sistem örneği olarak 2 giriş ve tek çıkıştan oluşan sistem Şekil 4.1’de verilmiştir. 100 adet benzetim verisi [0 3] aralığında rastgele seçilen (x,y) giriş verisine karşılık Denklem (4.1) ile oluşturulmuştur (Chen, 1998).

$$z=(2+x^{1.5}-1.5\sin(3y))^2 \quad (4.1)$$

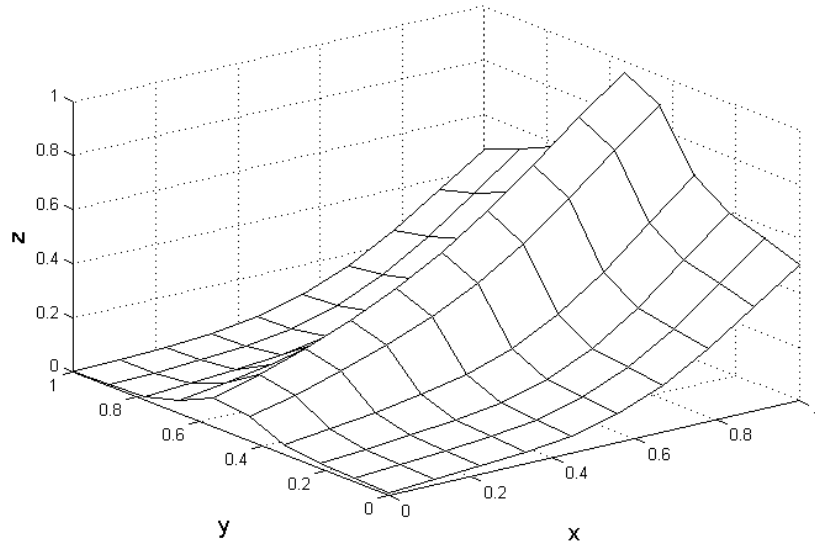


Şekil 4.1 Doğrusal olmayan çok girişli tek çıkışlı sistem

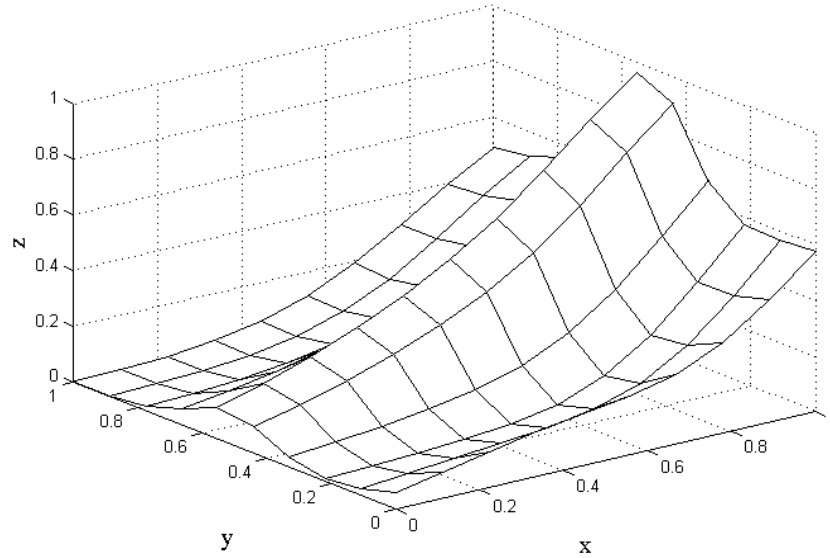
Modelleme işleminin sonucu Şekil 4.2’de, sonuçlar ise Çizelge 4.1’de verilmiştir.

Çizelge 4.1 Yöntemlere ait parametreler ve sonuçlar

Yöntem	KYA+BCO+GA	KYA+GK+GA
Başlangıç küme sayısı	4	4
Sonuç küme sayısı	3	4
Hata (RMS)	0.057257	0.04060
Parametreler	$\delta_{CPN}=0.4$, $N_{GD}=100$, $\mu_C=\mu_\sigma=\mu_w=0.1$	



(a) KYA+BCO+GA



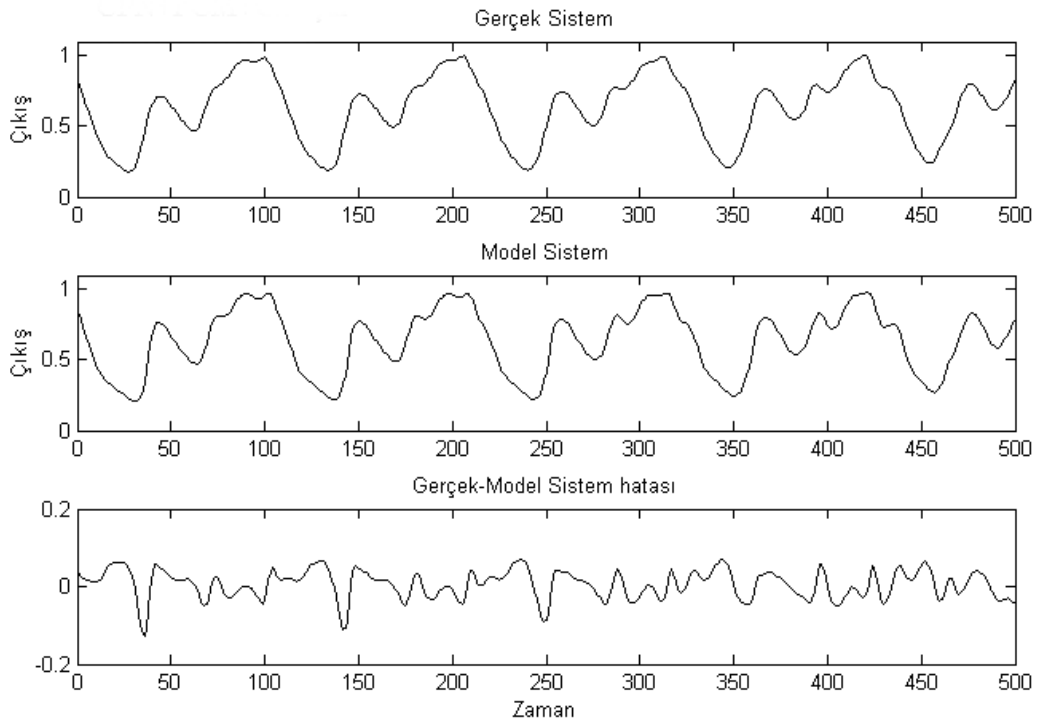
(b) KYA+GK+GA

Şekil 4.2. Sistem Modelleri

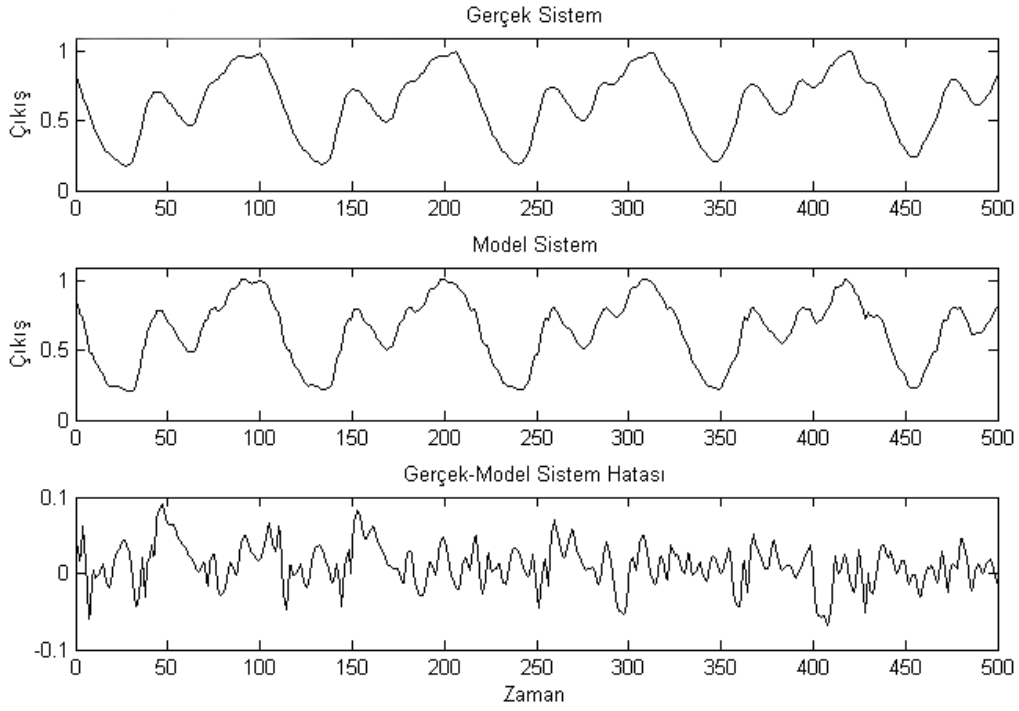
Örnek 4.2 (Özdemir ve ark. , 2009) : Kaotik Mackey-Glass (MG) zaman serisi modelleme. Son örnek olarak verilen MG serisi Denklem(4.2)’den üretilen ve literatürde oldukça fazla kullanılan bir örnektir (Shing ve Jang, 1993; Chng ve ark. , 1996; Duro ve Reyes, 1999).

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \quad (4.2)$$

MG serisi modelleme için veri seti üretiminde kullanılan Denklem(4.2)’de $\tau=17$ olarak seçilmiş eğitim için üretilen 500 adet veri $X=[x(t-3),x(t)]$ ve $Y=[x(t+3)]$ şeklinde üç boyutlu olarak oluşturulmuştur. Modelleme işleminin sonucu Şekil 4.3’de , sonuçlar ise Çizelge 4.2’de verilmiştir.



(a) KYA+BCO+GA



(b) KYA+GK+GA

Şekil 4.3. Sistem Modelleri

Çizelge 4.2 Yöntemlere ait parametreler ve sonuçlar

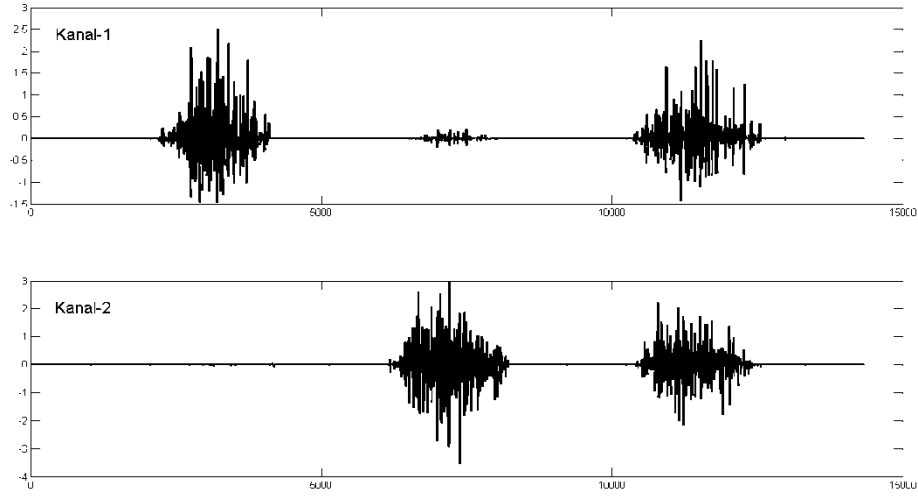
Yöntem	KYA+BCO+GA	KYA+GK+GA
Başlangıç küme sayısı	10	10
Sonuç küme sayısı	5	6
Hata(RMS)	0.035964	0.021712
Parametreler	$\delta_{CPN}=0.13$, $N_{GD}=50$, $\mu_C=\mu_\sigma=\mu_w=0.007$	

4.2. İki kanallı EMG işareti ile 4 farklı hareketin gerçek zamanlı denetimi

İki kanallı EMG işareti ile 4 farklı hareketin gerçek zamanlı hareketin yapılması için gerekli işaret işleme aşamalar ve elde edilen bulgular sırasıyla aşağıda verilmiştir.

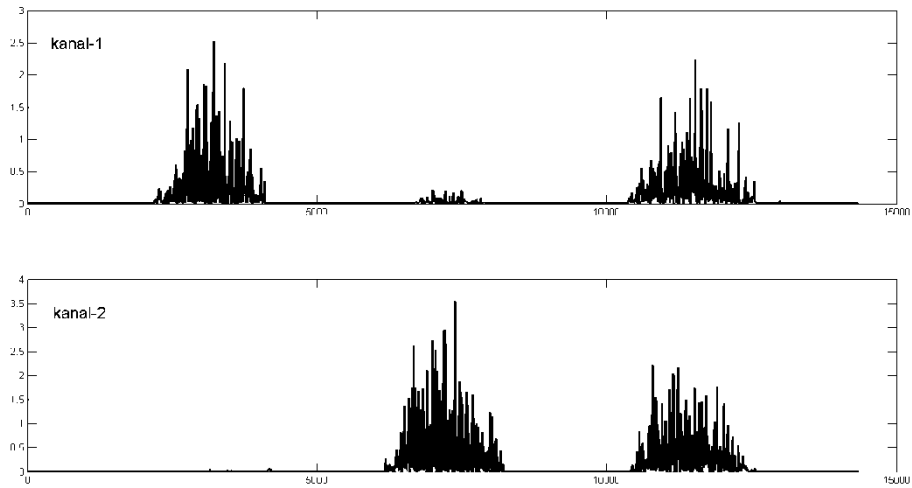
a) Gürültüden Arındırma: Kayıtlı EMG verilerindeki istenmeyen yüksek sıklık bileşenleri dalgacık dönüşümü kullanılarak giderilmiştir. Gürültüden arındırılan işaret

daha sonra Fourier Dönüşümü kullanılarak DC bileşenden arındırılmıştır. Şekil 4.4 gürültüden ve DC bileşenden arındırılmış iki kanallı EMG işaretini göstermektedir.



Şekil 4.4 Gürültü ve DC bileşenden arındırılmış EMG işareti

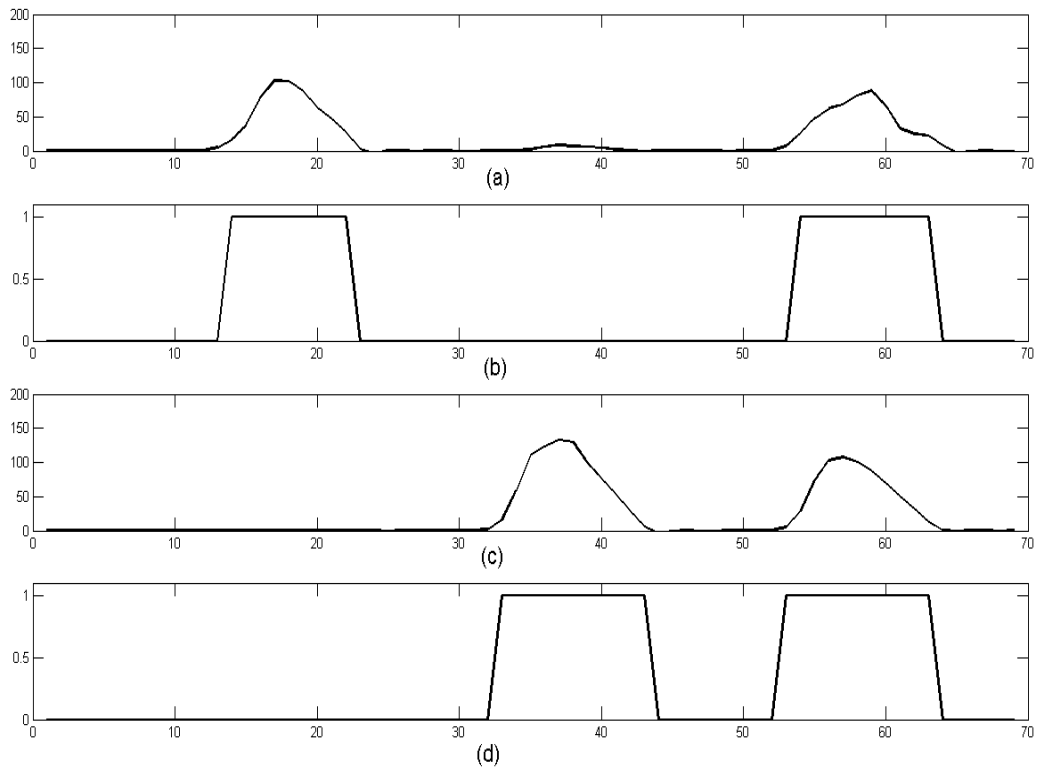
b) EMG İşaretinin Doğrultulması ve Zarfının Çıkarılması: İşaret mutlak değeri alınarak doğrultulmuş ve bir alçak geçiren filtre kullanılarak zarfı çıkarılmıştır. Şekil 4.5 ise doğrultulmuş EMG işaretini göstermektedir.



Şekil 4.5 Doğrultulmuş 2 kanallı EMG işareti

c) Eşikleme ve Denetim İşaretinin Üretilmesi: Gerçek zamanlı olarak 100 ms süreli sabit pencere aralığında alınan doğrultulmuş ve zarfı çıkarılmış EMG verisinin (x_i)

genlik karelerinin toplamı $(\bar{x} = \sum_{i=1}^N |X_i|^2)$ hesaplanmıştır. Her iki kanal içinde deneysel olarak farklı seçilmiş olan eşik değerleri ile karşılaştırılır. Bu eşik değerlerinin işareti alınan kişiye göre değişiklik göstermektedir. Bu karşılaştırma sonunda Şekil 4.6’ da görüldüğü gibi kas aktivitelerini karakterize eden mantık (1 ve 0) değerleri üretilir. Hareketler ve denetim işaretleri Çizelge 4.3’ de verilmiştir.



Şekil 4.6 Denetim işaretinin üretimi: Her kanal için üretilen mantık seviyeleri ve Çizelge 4.3 kullanılarak denetim işareti (veya sınıf bilgisi) üretilmektedir.

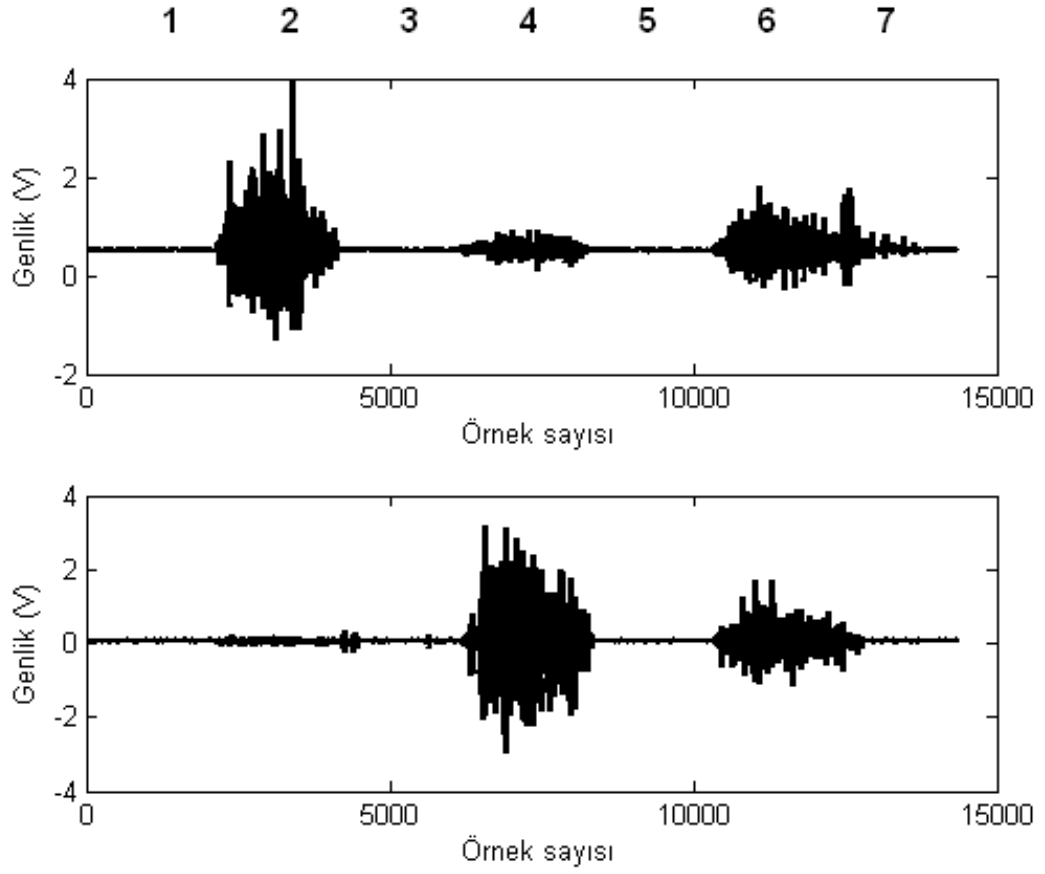
- (a) Kanal-1 için 100 ms’lik işaretin toplamı (b) Kanal-1 için üretilen mantık seviyeleri
(c) Kanal-2 için 100 ms’lik işaretin toplamı (d) Kanal-2 için üretilen mantık seviyeleri

Çizelge 4.3 Doğruluk Çizelgesi

Kanal-1	Kanal-2	El Hareket tipi
1	0	Dışa bükme
0	1	İçe bükme
1	1	Kavrama
0	0	Serbest

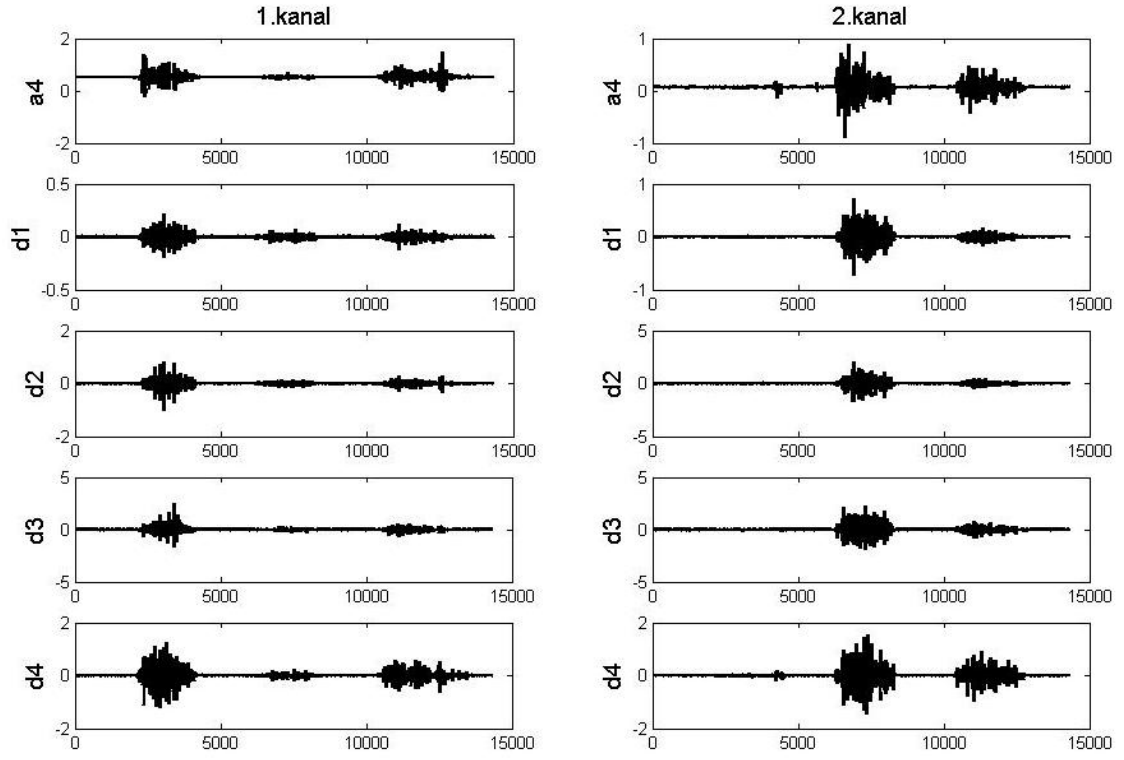
4.3. İki kanallı EMG verisinden özellik çıkarma ve tasarlanan sınıflandırıcının sınanması

Şekil 4.7, bu özellik çıkarım aşamalarının uygulamasına örnek olarak kullanılacak olan, 28 yaşında erkek bir denekten iki kanallı olarak kaydedilmiş ham EMG verisini göstermektedir. Her bir kayan pencereye uygulanan dalgacık dönüşümü sonucunda 5 adet yaklaşım ve detay katsayılar matrisi elde edilmiştir. Her bir kanal için elde edilen 5x128 boyutlu veriye DDA uygulanmış ve her pencere için 5x128 boyutlu veri 1x4 boyuta indirgenmiştir. Elde edilen bu 4 adet verinin mutlak değerlerinin toplamından her 128 msn lik pencere artımında her bir kanal için 1 adet veri elde edilmiştir. Elde edilen bu özellikler, ham EMG verisini oldukça iyi temsil ettiklerinden sınıflandırıcıya giriş olarak verilmişlerdir.

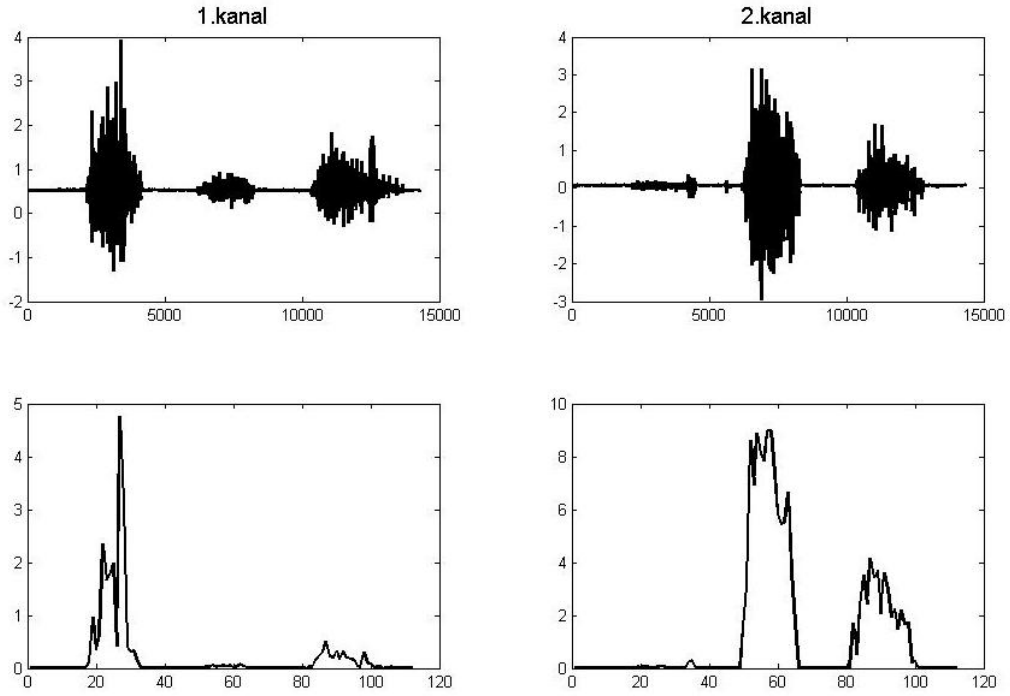


Şekil 4.7. İki kanallı ham EMG verisi (1-3-5-7): Serbest durum, (2): Yukarı, (4): Aşağı, (6): Yumruk

Ham EMG verilerine ADD uygulandığında elde edilen dalgacık katsayıları Şekil 4.8 ile gösterildiği gibidir. Fakat elde edilen bu yüksek boyutlu özellikler hala sınıflandırıcıya verilemeyecek kadar büyük bir boyuta sahiptir.



Şekil 4.8 İki kanallı ham EMG verisinin ayrık dalgacık dönüşümü ile analizi

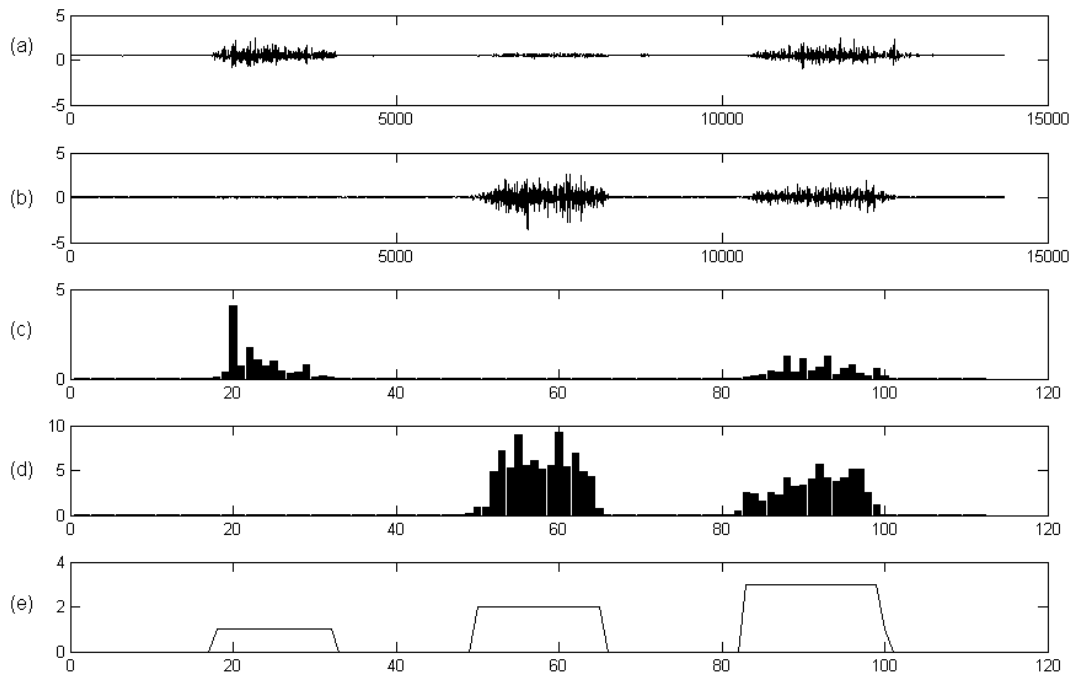


Şekil 4.9 (a) 1. Kanal ham EMG verisi (b) 2. Kanal ham EMG verisi
(c) 1.Kanal için özellik çıkarılmış veri (d) 2.Kanal için özellik çıkarılmış veri

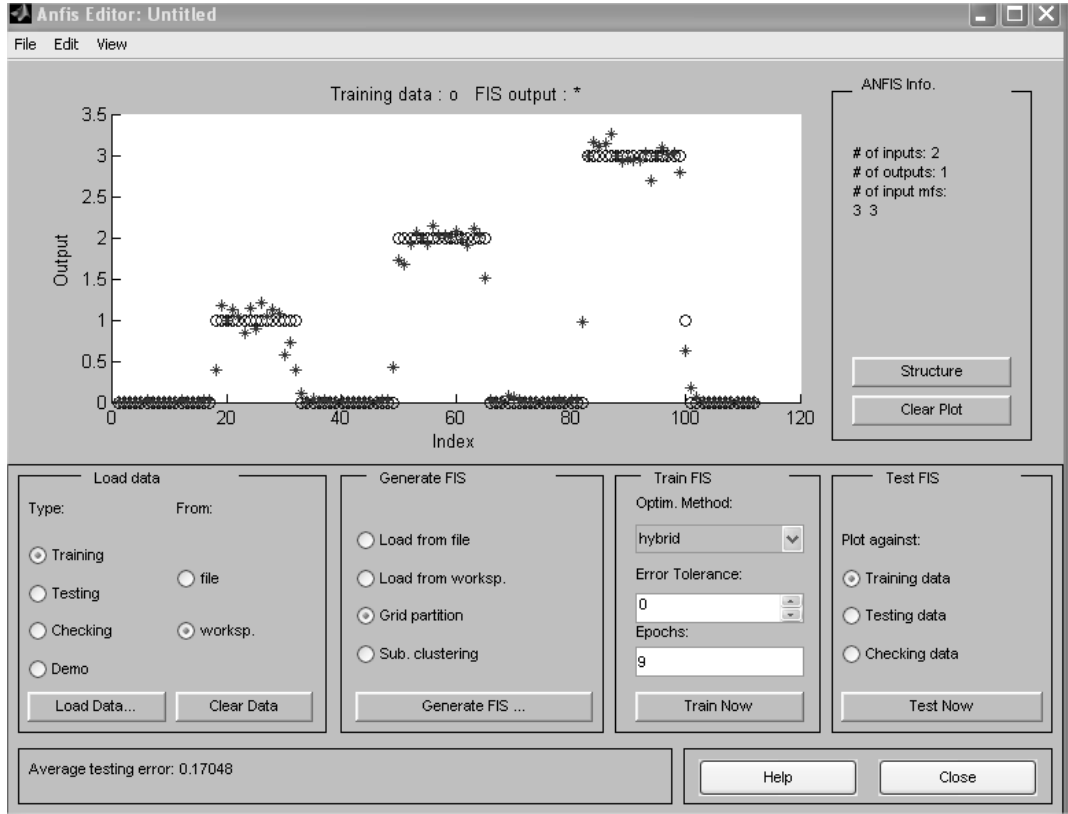
Şekil 4.9, ham EMG verisi ile önerilen özellik çıkarım yöntemi olan Dalgacık+DDA çıkışlarını göstermektedir. Görüleceği üzere 14336 boyutlu ham EMG verisinin, DDA'nın sınıf bilgisini dikkate alması özelliğiyle daha iyi bir sınıf ayrılabilirliğine sahip düşük boyutlu (112) verilerle çok iyi bir biçimde temsil edilebildiği görülmektedir.

Şekil 4.10 DDA ile özellik vektörü çıkarılmış ham EMG işaretlerini ve hareketlere karşılık gelen çıkış (sınıf) bilgisini göstermektedir.

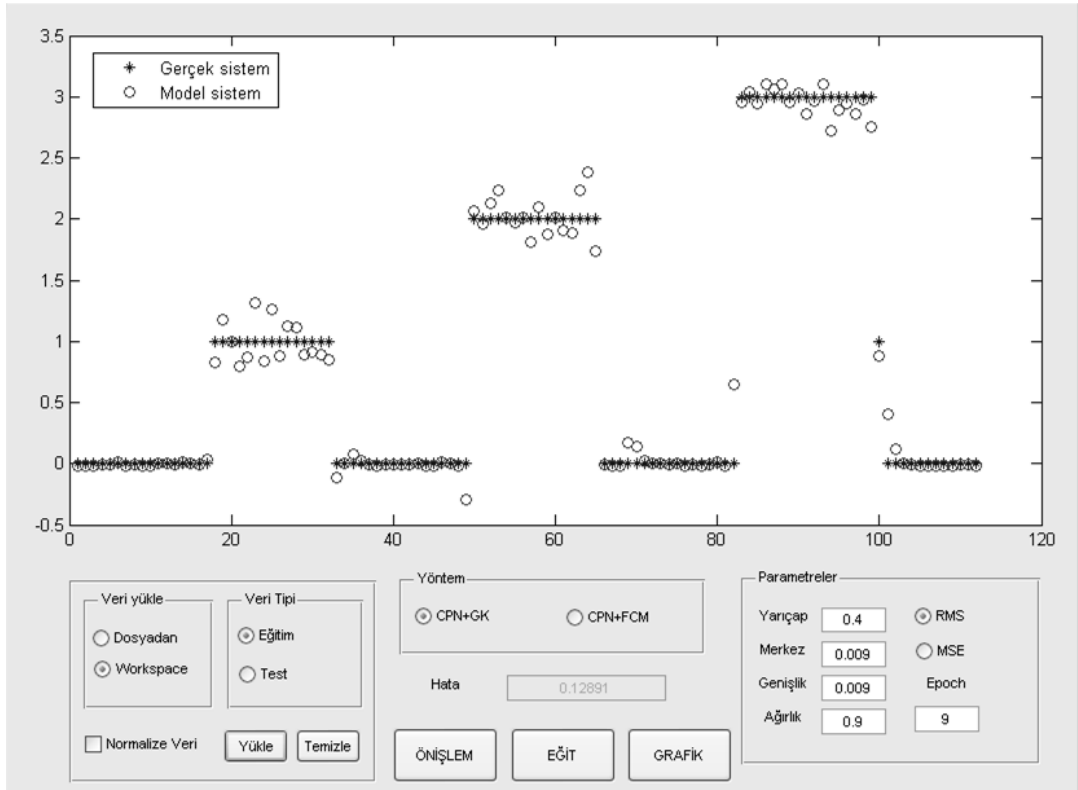
Şekil 4.11 DDA ile özellik çıkarılmış EMG verisinin ANFIS yapısında eğitilmesini , Şekil 4.12 ise melez ağ yapısındaki eğitim sonuçlarını göstermektedir. DDA ile oluşturulan özellik vektör verisi melez ağ yapısına ve ANFIS ağ yapısına giriş olarak verilmiştir. Yarıçap parametresi 0.4 seçilerek KYA 'nın bulduğu 21 adet başlangıç merkezi , GK algoritması ile 8 merkeze indirilmektedir. 8 kuralla eğitilen melez ağ için RMS hatası 0.12891 iken ANFIS ağında RMS hatası 0.17048 bulunmuştur. Bu iki yapı sınıflandırma hatası bakımından karşılaştırıldığında melez yapının sınıflandırma doğruluğu %99.1071 , ANFIS yapısının sınıflandırma doğruluğu %97.3214 olarak hesaplanmıştır.



Şekil 4.10 (a) (b) 1. ve 2. Kanal ham EMG verisi (c) (d) 1. ve 2. özellik vektörü
(e) Hareket çıkışı



Şekil 4.11 ANFIS ağ yapısı ile eğitim



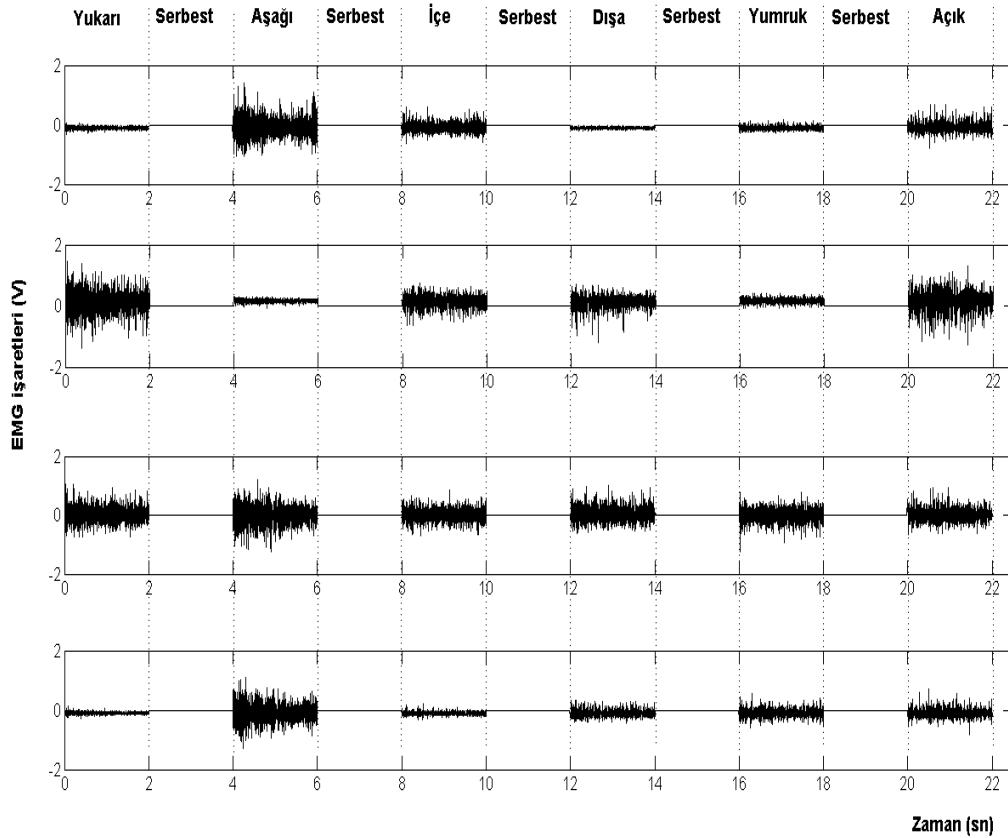
Şekil 4.12 Melez ağ yapısı ile eğitim

4.4. Dört Kanallı EMG İşareti ile Elektromekanik Elin Gerçek Zamanlı Denetlenmesi

Dört kanallı olarak kaydedilen EMG verilerinden elektromekanik elin denetlenmesi aşamasında elde edilen bulgular aşağıda verilmiştir.

4.4.1. Veri Alımı Aşaması

Belirlenen 7 hareket için dört kanallı EMG verisi kaydedildiği zaman elde edilen ham EMG verisi ve karşılık gelen hareketler Şekil 4.13’de verilmiştir.

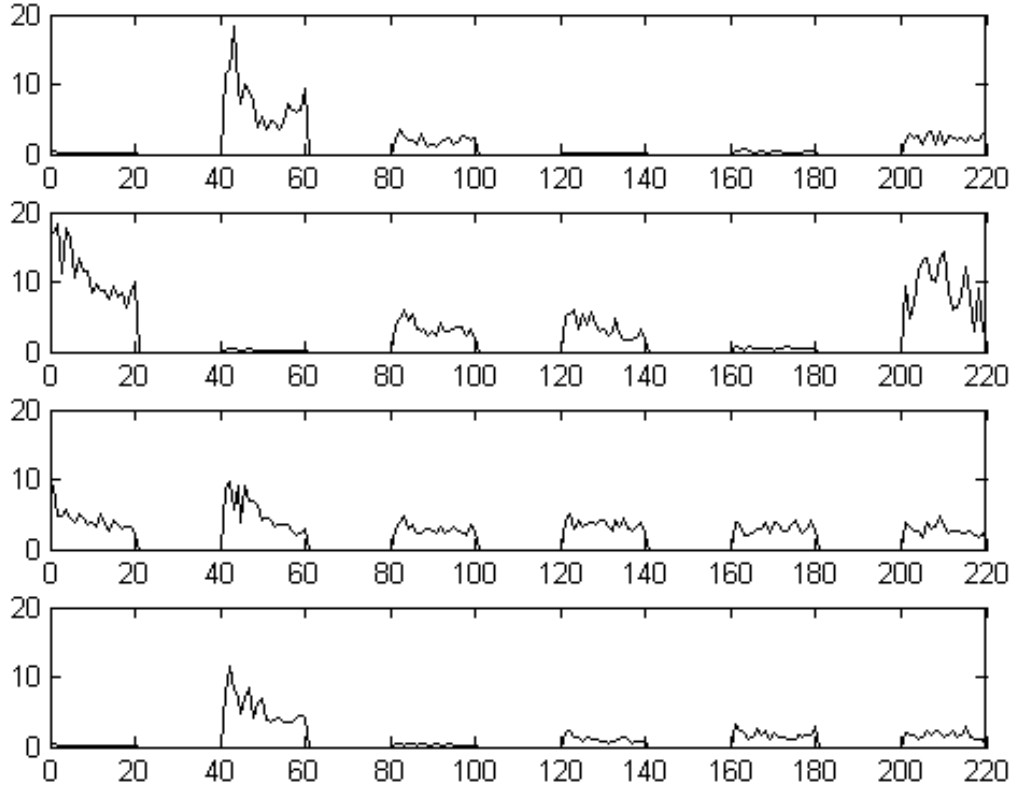


Şekil 4.13 Kas hareketlerini temsil eden ham EMG verisi

4.4.2. Verinin Ön-işleme Aşaması

Her bir pencere artımı içinde TBA ile boyut azaltma işlemi sonucunda, her bir kanal için 4 adet katsayı elde edilmiştir. Elde edilen bu katsayıların mutlak değerlerinin

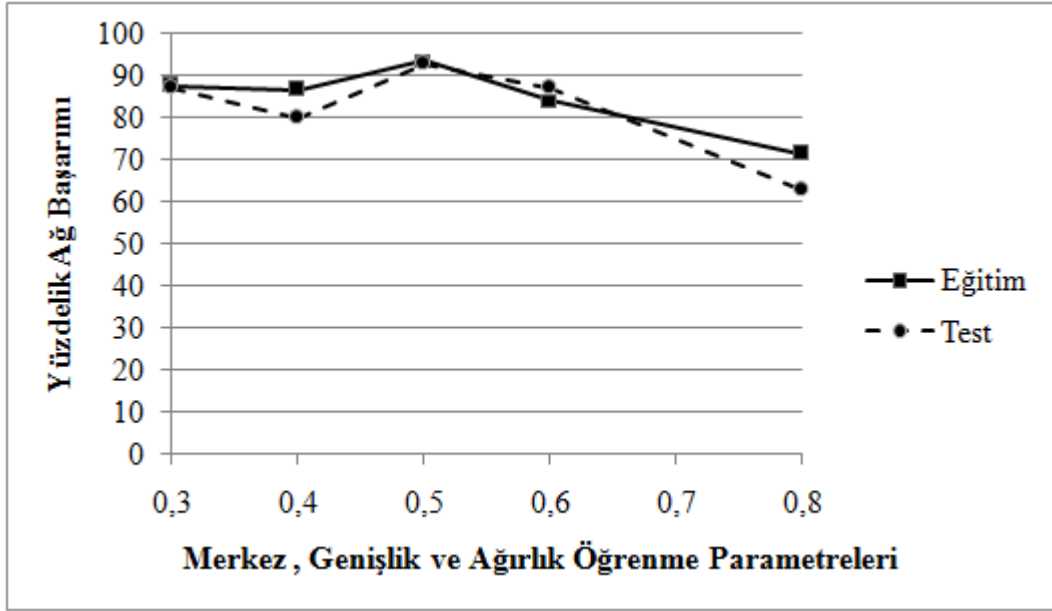
toplamı ise özellik vektörü olarak kullanılmıştır. Bu özellik çıkarımı stratejisiyle, pencere artımında her bir kanaldan 1, toplamda ise 4 adet özellik elde edilmiştir. Elde edilen özellik verisi Şekil 4.14 ile verilmektedir.



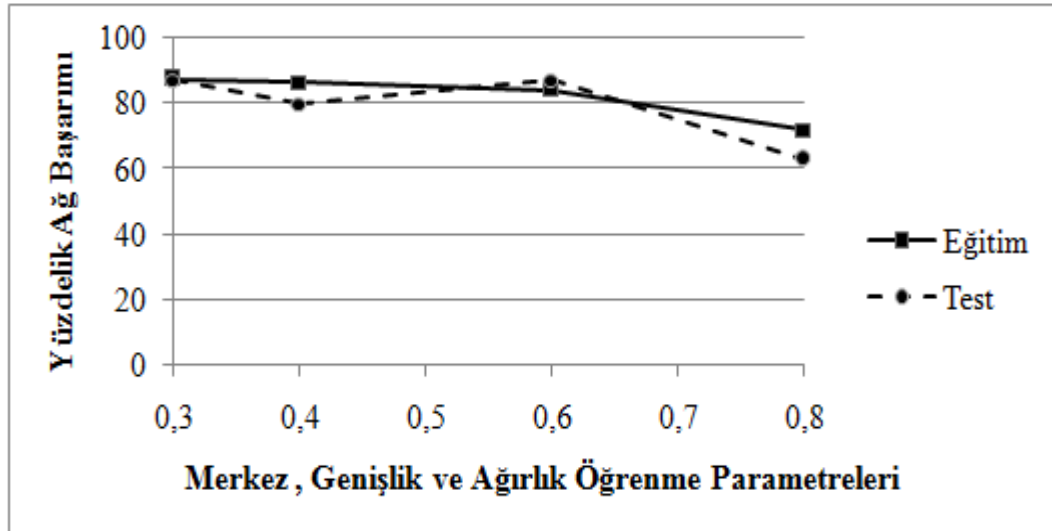
Şekil 4.14 Dalgacık Dönüşümü ve TBA kullanılarak özellik çıkarımı

4.4.3. Özelliklerin Sınıflandırılması

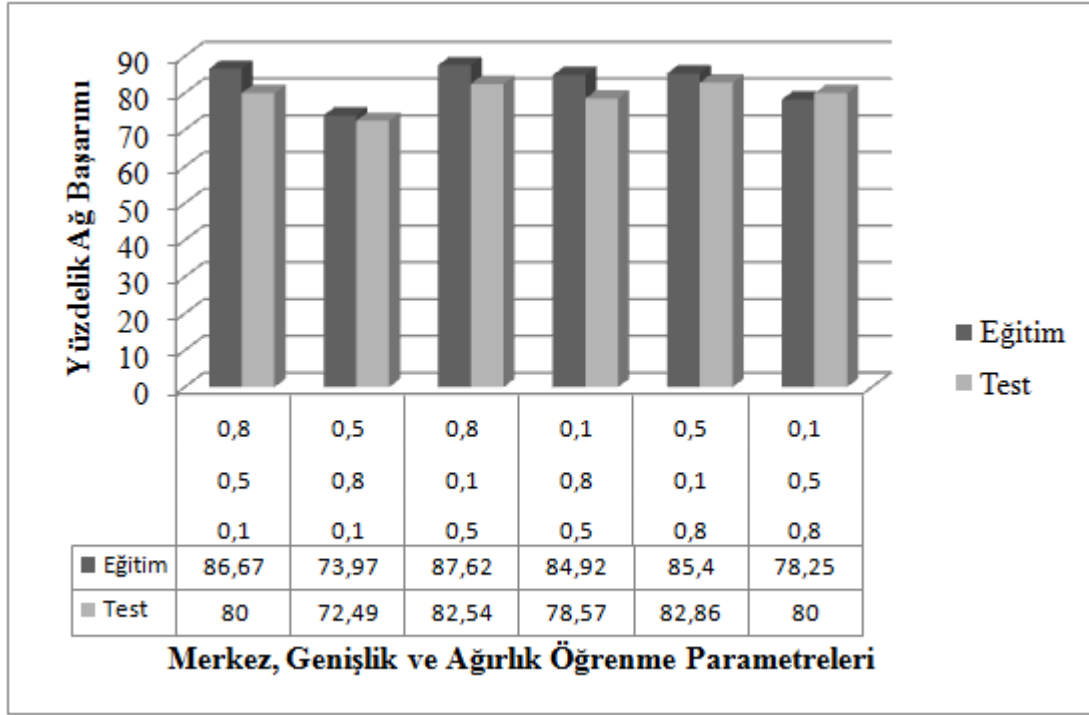
Bu çalışmada üç sınıflandırıcı yapısı ile başarımların karşılaştırılması yapılmıştır. İlk olarak DANRTİ yapısı kullanılmıştır. Bu yapıda KYA ve GK önışlemci olarak belirlenmiştir. Yarıçap parametresi 4 seçildiğinde KYA tarafından bulunan 39 adet küme GK algoritması tarafından 34 kümeye optimize edilmiştir. Farklı merkez, genişlik, öğrenme parametreleri ve epoch sayısına bağlı yüzdelik ağ başarımları Şekil 4.15, Şekil 4.16 ve Şekil 4.17 ile verilmiştir.



Şekil 4.15 Epoch=2 için merkez, genişlik ve ağırlık öğrenme parametrelerinin eşit seçilmesi durumunda eğitim ve test verilerinde yüzdelerik ağ başarıımı

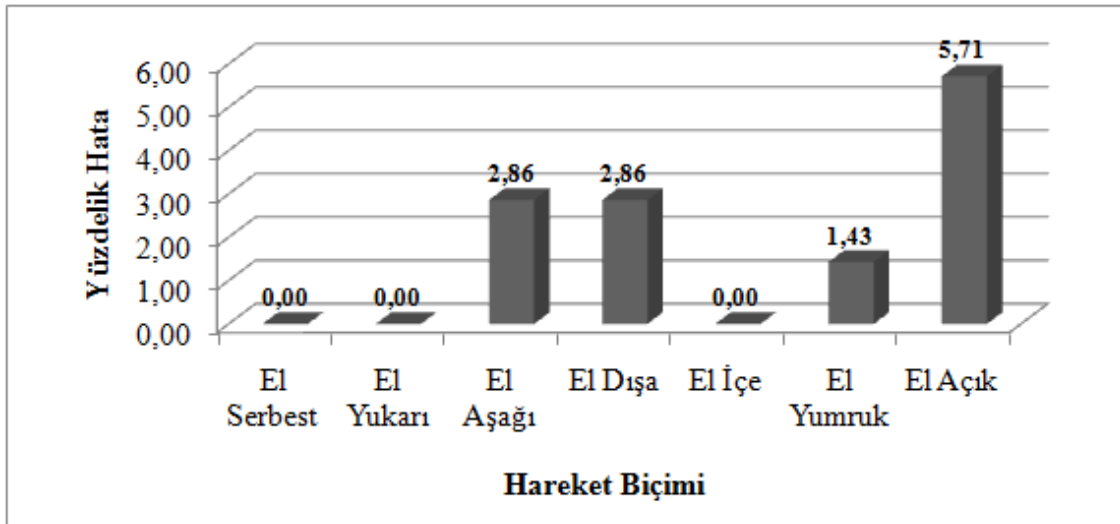


Şekil 4.16 Epoch=4 için merkez , genişlik ve ağırlık öğrenme parametrelerinin eşit seçilmesi durumunda eğitim ve test verilerinde yüzdelerik ağ başarıımı

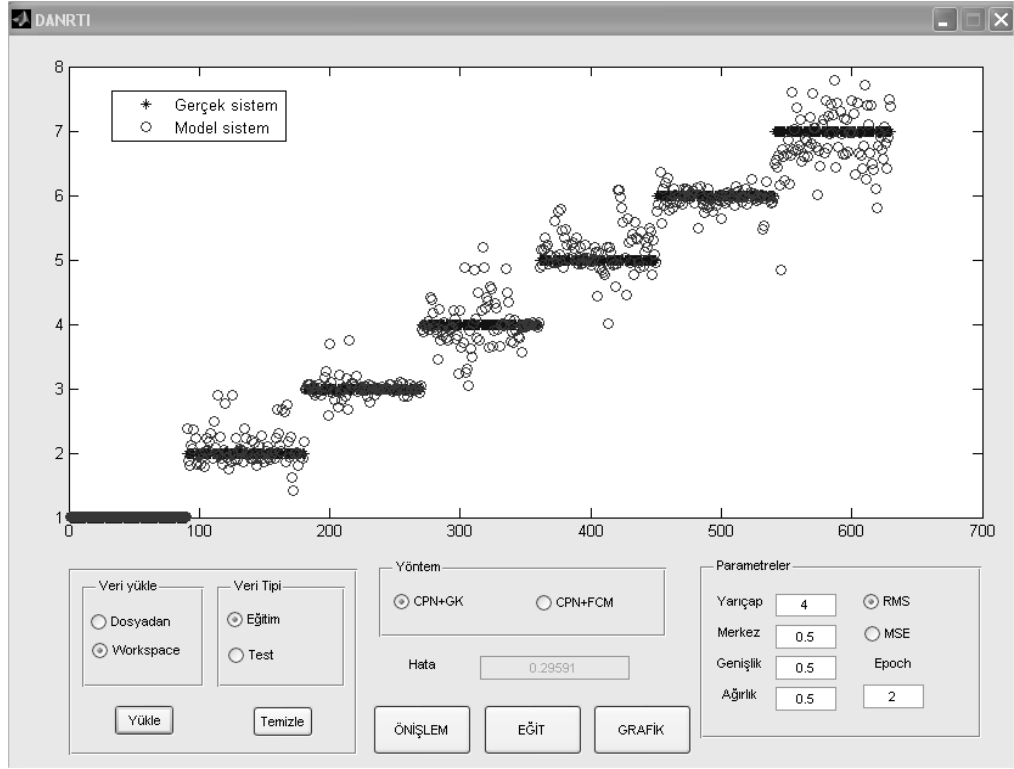


Şekil 4.17 Epoch=2 için merkez, genişlik ve ağırlık öğrenme parametrelerinin farklı seçilmesi durumunda eğitim ve test verilerinde yüzdelik ağ başarımları

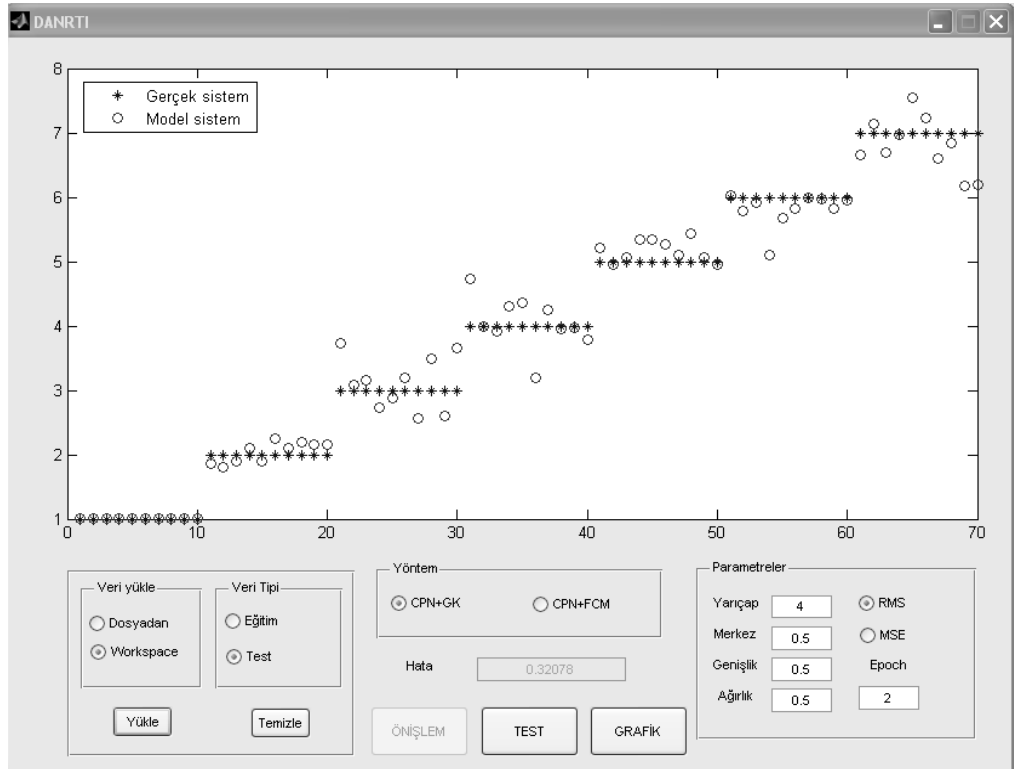
Öğrenme parametrelerinin bu deneme / yanılma yöntemi sonuçları neticesinde Şekil 4.15 ile verildiği biçimde Merkez , Genişlik ve Ağırlık öğrenme parametreleri 0.5 belirlenmiş 2 epoch için DANRTİ ağı eğitilmiş (Şekil 4.19) ve test edilmiştir (Şekil 4.20). Test edilen hareketlere ait yüzdelik hata Şekil 4.18 'de verilmiştir.



Şekil 4.18 DANRTİ ağı için hareket biçimine göre yüzdelik hata değişimi

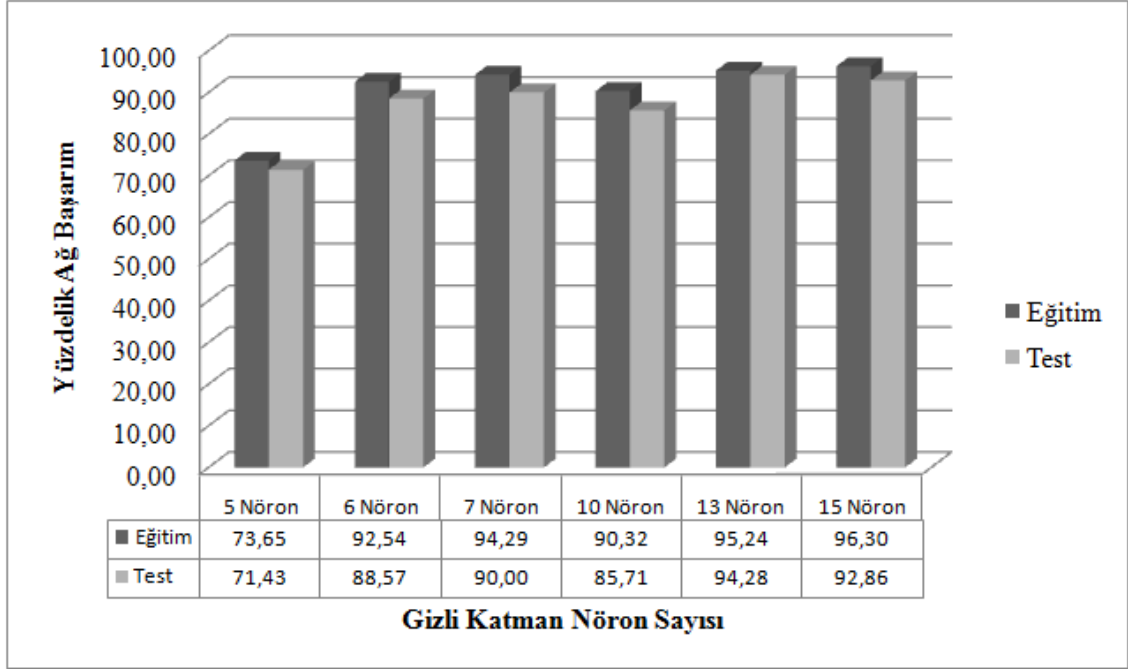


Şekil 4.19 DANRTİ yapısı ile dört kanallı EMG sınıflandırması eğitim sonucu



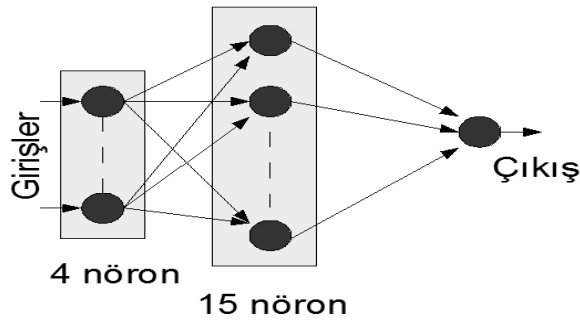
Şekil 4.20 DANRTİ yapısı ile dört kanallı EMG sınıflandırması test sonucu

ÇKYSA ağının oluşturulması aşamasında ara katmanda kullanılan nöron sayısına göre aktivasyon işlevi logaritmik sigmoid ve epoch =50 seçildiği durumdaki eğitim ve test başarımları Şekil 4.21 ile verilmiştir.

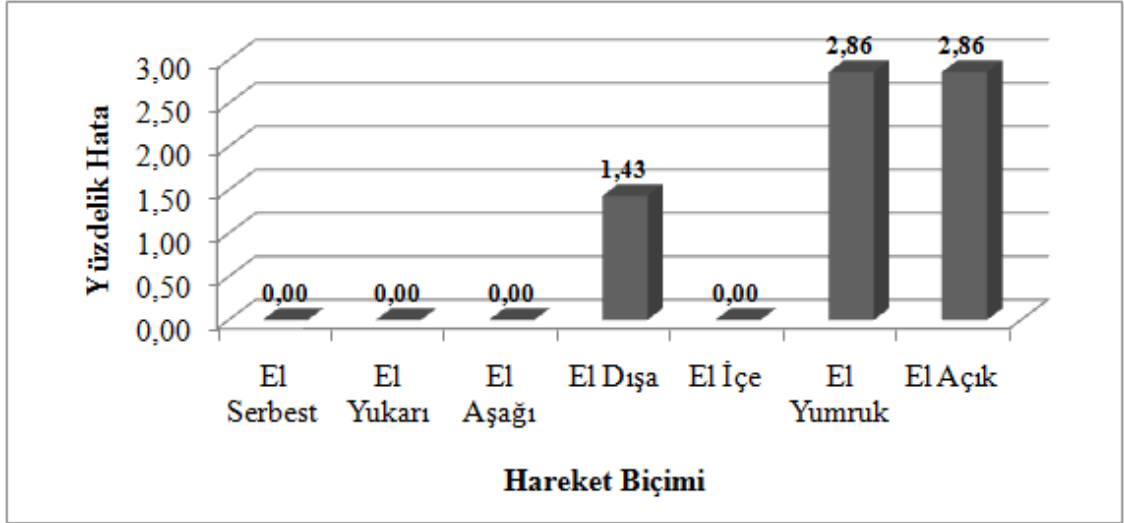


Şekil 4.21 ÇKYSA için nöron sayısına bağlı olarak yüzdelik ağ başarımları

ÇKYSA ağının eğitilmesi sürecinde ara katmanında 15 adet nörondan oluşan Şekil 4.22’de şematik gösterimi verilen tek katmanlı ağ yapısı seçilmiştir. Bu durumda test edilen hareketlere ait yüzdelik hata Şekil 4.23’de verilmiştir.

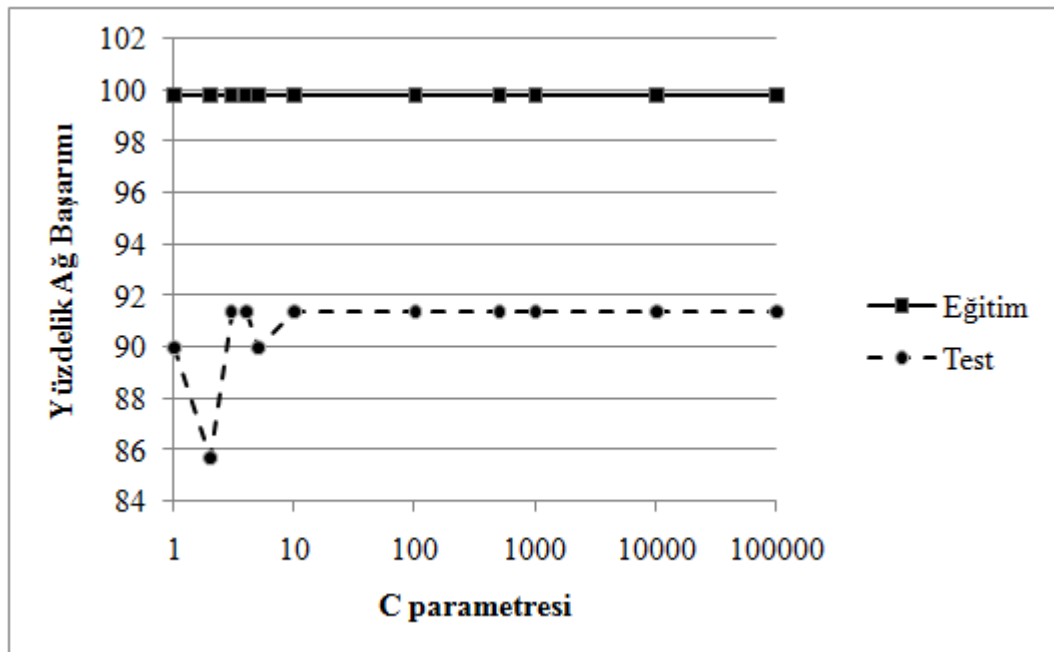


Şekil 4.22 Kullanılan ÇKYSA yapısı

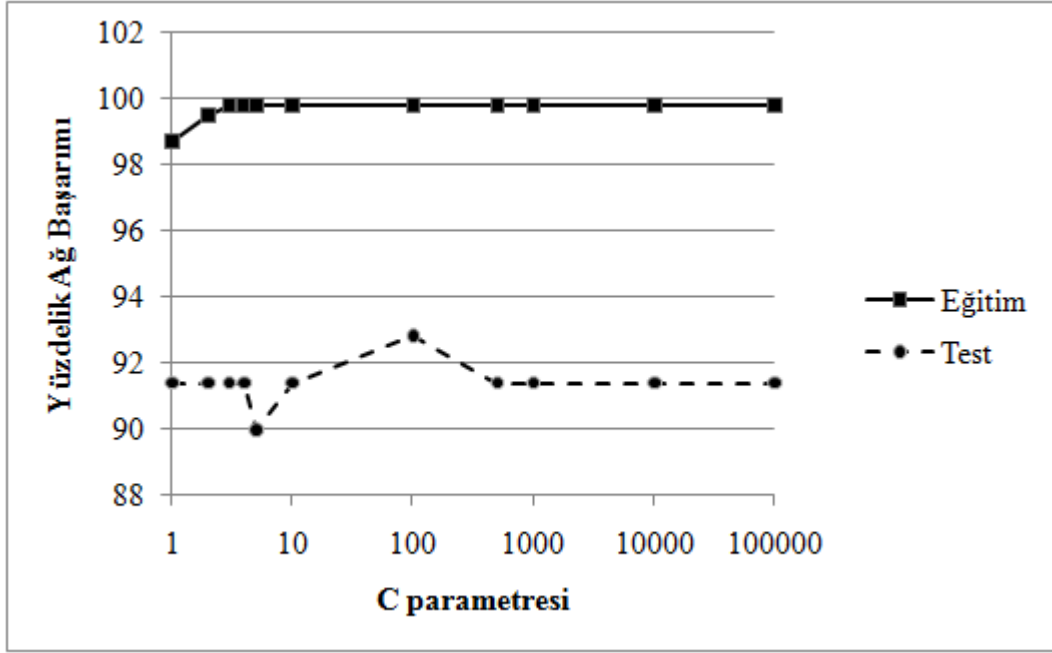


Şekil 4.23 ÇKYSA için hareket biçimine göre yüzdelik hata değişimi

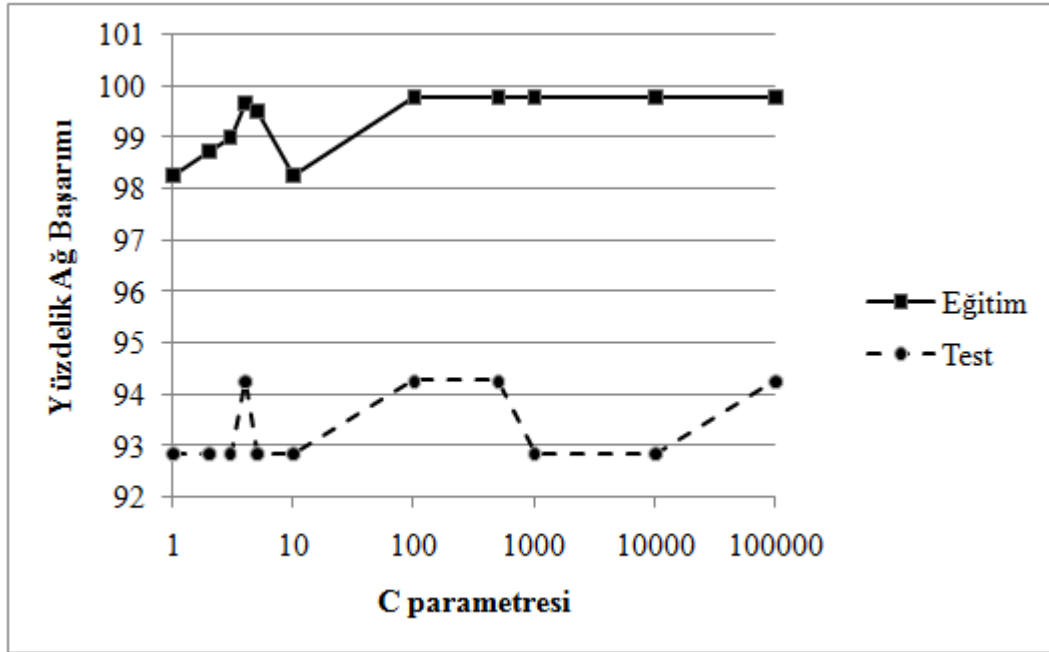
Son olarak, DVM ile sınıflandırma yapmak için STPRTOOL (Vojtech ve Hlavac, 2004) adlı yazılım kullanılmıştır. Çekirdek işlevi olarak RTİ, çoklu sınıflandırma yöntemi olarak BKB seçilmiştir. Farklı γ ve C parametrelerine (logaritmik eksen) göre DVM ağının yüzdelik ağ başarımı Şekil 4.24, Şekil 4.25 ve Şekil 4.26 ile verilmiştir.



Şekil 4.24 $\gamma = 1$ için ceza faktörünün (C) değişimine bağlı olarak yüzdelik ağ başarımı

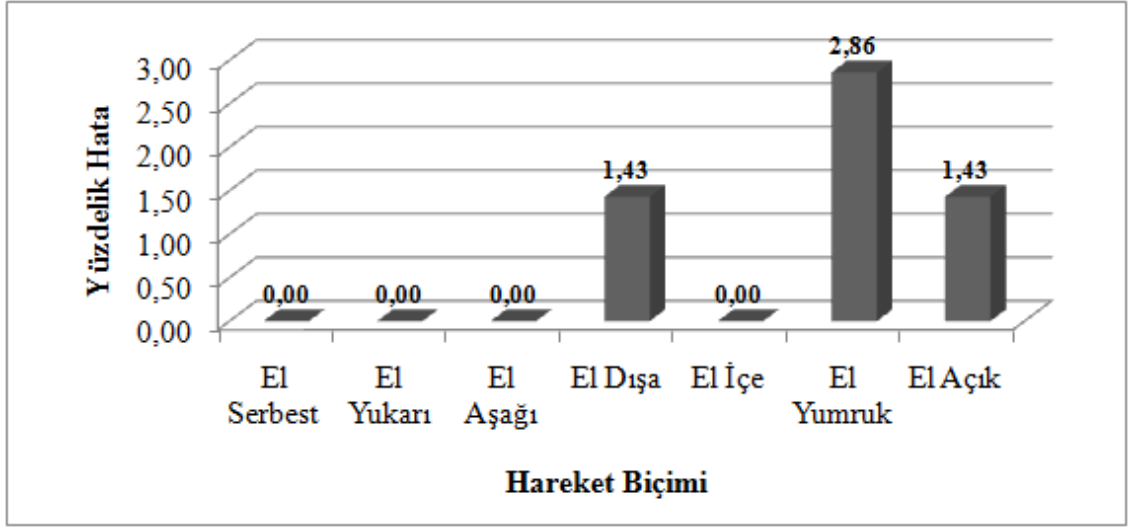


Şekil 4.25 $\gamma = 2$ için ceza faktörünün (C) değişimine bağlı olarak yüzdelik ağ başarımı



Şekil 4.26 $\gamma = 3$ için ceza faktörünün (C) değişimine bağlı olarak yüzdelik ağ başarımı

Belirlenen $\gamma = 3$ ve $C=100000$ için DVM ağı oluşturulduğunda hareket biçimlerine göre yüzdelik hata değişimi Şekil 4.27' de verilmiştir.



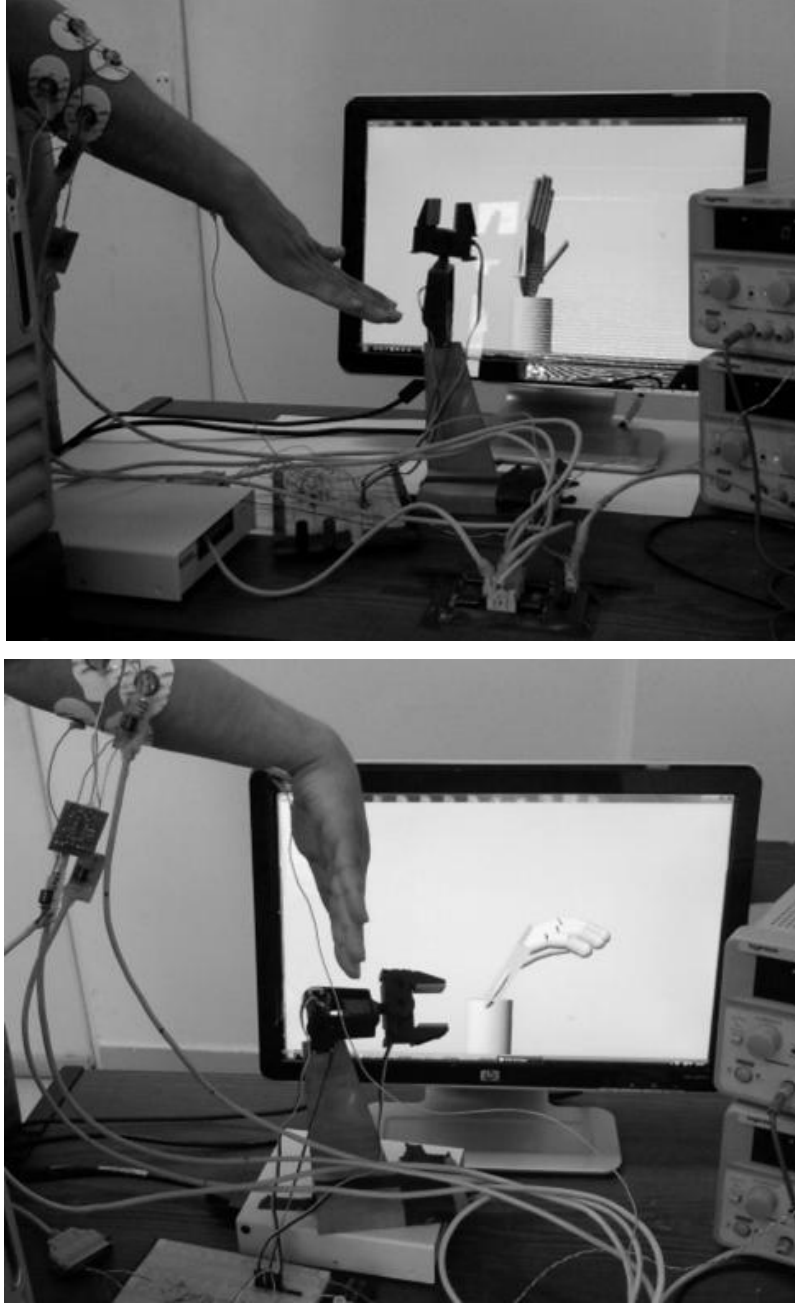
Şekil 4.27 DVM ağı için hareket biçimine göre yüzdelik hata değişimi

Her üç yöntemden de elde edilen sonuçlar, 7 hareketi temsil edecek biçimde 3 bitlik denetim işaretleriyle Çizelge 4.4 deki gibi belirlenmiştir.

Çizelge 4.4 Yöntemlere ait parametreler ve sonuçlar

Denetim Bitleri	Servo/Hareket Denetimi	El Hareket tipi
0 0 1	1. Servo Sağa	Yumruk
0 1 0	2. Servo Sağa	Dışa
0 1 1	3. Servo Sağa	Yukarı
1 0 0	Servolar merkeze	Serbest
1 0 1	1.Servo Sola	Açık
1 1 0	2.Servo Sola	İçe
1 1 1	3. Servo Sola	Aşağı

Robot elin ve bir sanal elin gerçek zamanlı olarak denetlenmesi Şekil 4.28 ‘de seçilen iki hareket için gösterilmiştir.



Şekil 4.28 El açık ve el aşağı hareketlerinin sanal el simülatörüne ve robot ele gerçek zamanlı olarak uygulanması

5. SONUÇ VE ÖNERİLER

Bu tezin algoritmik geliştirme aşaması için KYA ve BCO algoritmaları incelenmiştir. KYA algoritması kaba bir sınıflama yapmaktadır. BCO algoritması ise küme sayısına bağlı bir algoritmadır. Ayrıca dairesel bir küme yapısına göre kümeleme yapmaktadır. Veriler her zaman dairesel bir dağılım sergilemek durumunda değildir. Bunun eksiklikleri gidermek için alternatif kümeleme algoritması olarak GK kümeleme algoritmasına geçiş yapılmış ve küme sayılarını optimize etmek için algoritmalar bölüm doğrulama algoritmaları ile tümleşik bir yapıya dönüştürülmüştür. Oluşturulan bu yapı ile, tezin algoritmik aşaması için optimum kural sayısını elde edebilen bir ön işlem bloğu elde edilmiştir.

Tezin algoritmik geliştirme aşamasının ikinci basamağında, ön-işlem bloğunda oluşturulan kümeleri başlangıç kabul eden melez ağ yapıları oluşturulmuştur. Başlangıç olarak önerilen melez yapı, kullanılan GA algoritmasının doğası en iyi modeli oluşturmamakta ve sınıflandırma çalışmalarında istenen başarıyı sağlamamaktadır. Bu model literatürde kullanılan örnekler üzerinde BCO ve GK yapılarından oluşan ön-işlem bloğunun başarımının sınanması amacıyla kullanılmıştır. Verilen örneklerde GK ile oluşturulan yapının başarımının daha iyi olduğu gözlenmiştir. Bu melez yapının başarımını artırmak amacıyla ağırlık katmanı yeniden düzenlenerek ve ağ normalize RTİ lerden oluşturulmuştur. GA algoritmasının yerine GA ve LM algoritmaları kullanılmıştır.

Tezin ana amacı olan elektromekanik elin gerçek zamanlı denetimi için başlangıç olarak iki kanallı EMG verileri üzerinde çalışmalar yapılmıştır. Seçilen 4 tip el hareketine ait EMG işaretleri kaydedilmiş ve 6 aşamalı ve hesapsal yükü düşük olan bir yapı kullanılmıştır. 2 kanallı EMG işaretleri ile 4 tip el hareketi sınıflandırılmış ve paralel port aracılığıyla sürülen mikro denetleyicilerle elektro-mekanik protez el denetlenmiştir. Denemeler sonucunda elektro-mekanik deneysel elin denetlenmesi sürecinde yaklaşık 40 ms'lik hesapsal gecikme gözlenmiştir. Tasarlanan yapı hiçbir ileri seviye örüntü tanıma tekniklerini kullanmadan, deneysel bir protez el gibi elektro-mekanik işçilik, işaret işleme, denetleme ve gömülü işlemci pratikleri gerektiren bir yapıyı deneme ve geliştirme fırsatı vermektedir.

EMG işaretlerini tanıma, işaret ve özelliklerindeki büyük değişimlerden dolayı karmaşık örüntü tanıma problemlerindendir. Ham EMG verisinden özellik çıkarımı sınıflandırma işleminin en önemli aşamalarından biridir. Elde edilen özelliklerin yüksek sınıf ayrılabilirliğine sahip olması sınıflandırıcı başarımını doğrudan etkileyecektir. Tasarlanan hesapsal yükü düşük yapının hareket sayısı dolayısıyla kanal sayısı arttığında yetersiz kaldığı gözlenmiştir. Çıkarılan özellikler hem önerilen melez ağ yapısında hem de literatürdeki geçerli yapılardan biri olan ANFIS ağ yapısında eğitilerek başarımının tercih edilebilir olduğu sonucuna varılmıştır.

Tezin son adımında ise gerçek zamanlı olarak çalışan EMG denetimli protez el için tasarlanan düzenek, EMG işaretini insan önkol kaslarından EMG yükselteci ve veri alım kartıyla 4 kanallı olarak bilgisayara aktarmaktadır. İşaret işleme algoritması ile işlenen ham EMG verileri, yedi el hareketinden hangisinin yapılacağına karar vermektedir. Her 100 ms 'de bir üretilen denetim işareti ile sanal el simülatörü ve elektromekanik (robot) el ilgili hareketi gerçek zamanlı olarak yapmaktadır. Bu düzende kullanılan sınıflandırıcılar farklı parametre değerleri ile sınanmış olup belirlenen uygun parametreler ile ağ yapısı oluşturulmuştur. DANRTİ eğitim aşamasını %93.65, test aşamasını %92.68 doğrulukla, DVM eğitim aşamasını %99.80, test aşamasını %94.30 doğrulukla, ÇKYSA ise eğitim aşamasını %95.0 , test aşamasını % 92.90 doğrulukla gerçekleştirmiştir. Bu başarımlar bir elektromekanik elin gerçek zamanlı olarak denetlenmesinde yeterli sınırlarda kabul edilebilir.

Bu düzeneğin her bir alt parçası geliştirmeye açık bir yapıdadır. Geliştirilen deneysel düzenek parçalı bir yapıya sahip olduğu için, her bir alt parça diğerlerinden bağımsız olarak geliştirmeye açıktır. Kullanılan yazılımlarda yapılacak değişiklikler sayesinde farklı özellik çıkarımı, boyut azaltımı, sınıflandırma ve denetim algoritmaları denenebilir. Geliştirilen deneysel düzenek esnek ve parçalı olmasının yanı sıra lisans ve lisansüstü eğitim gören öğrencilerin kullanabileceği etkileşimli bir deney ve çalışma ortamı da sunmaktadır.

Gerçek zamanlı çalışan bir protez el veya kol için kullanılan mikrodenetleyici ve yazılımlarda yapılacak güncellemeler ile daha etkin ve hızlı bir yapı elde edilebilir. Ayrıca EMG işaretinden elin hangi hızla açılıp kapandığı bilgisinin çıkarılması veya elin düşmeyecek biçimde aşırı sıkmadan kavradığı cisimlere uyguladığı kuvvetin tahmin edilmesi bu çalışmanın devamı niteliğinde olacaktır.

Bu çalışmalar ticari bir protez elin geliştirilmesine kaynak oluşturacak kapsamdadır. Protezlerin gerçek bir elin/kolun işlevselliğine yaklaşması için ve mevcut mayoelektrik protezlerin başarımını artırmak için de çalışmalar sürdürülmektedir. Ticari bir firmanın (Ottobock) deneysel kolunda kullandığı TMR yöntemi daha fazla kaynak kullanarak daha çok hareketi yapabilme başarısını göstermektedir. Ancak cerrahi işlemler, sürecin uzun ve maliyetin fazla oluşu yöntemin kullanımını kısıtlamaktadır. Bu kapsamda yeni melez yaklaşımlar uygulanması mümkün olabilir. Mevcut ticari protezlerde kullanılan bir denetim stratejisi olan iki kastan alınan EMG işaretleri ile el, bilek ve dirsek eklemlerini ardışıl olarak hareket ettirme (kasların eş kasılması veya co-contraction) işlevi yerine el motorunun, bilek motorunun ve dirsek motorunun (veya ekleminin) seçiminde dört farklı yol izlenebilir. Bunlar; kişinin ses komutu, kasların kasılırken ürettiği sesin işlenmesi, kolun kinematik (ivme ve dönü ölçer) yapısının kullanılması ve bunların bazılarını içeren melez çözüm yöntemleri olup üzerinde çalışma gerektiren bir alan olarak beklemektedir.

6. KAYNAKLAR

- Abe, S., 2005. *Support Vector Machines for Pattern Classification*. New York: Springer.
- Ajiboye, A.B. ve ark., 2005. A Heuristic Fuzzy Logic Approach To EMG Pattern Recognition For Multifunctional Prosthesis Control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3), 280-291s.
- Artigue, V. ve ark., 2009. Development of a prosthetic arm: Experimental validation with the user and an adapted software. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*., 2009.
- Asyalı, M.H. ve ark., 2010. Design and implementation of a voice-controlled prosthetic hand. *Tubitak, Turkish Journal of Electrical Engineering& Computer Sciences*, 19(1), 1-14s.
- Babuska, R., 1998. *Fuzzy Modelling for Control*. Kluwer Academic Publishers.
- Belhumeur, P.N. ve ark., 1997. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7), 711 -720s.
- Bezdek, J.C., 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*.. MA, USA,: Kluwer Academic Publishers.
- Bouguessa, M. ve ark., 2006. An objective approach to cluster validation. *Pattern Recognition Letters*, 27(12), 1419-1430s.
- Burges, C.J.C., 1998. *A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery*.
- Cai, D. ve ark., 2007. Learning a Spatially Smooth Subspace for Face Recognition. In *CVPR'2007*., 2007.
- Campbell, C. ve ark., 2011. *Learning with Support Vector Machines Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan&Claypool.
- Chen, M.Y., 1998. *Integrated Knowledge-based Intelligent Control Systems: Principles, Methodologies and Practice*. PhD thesis: The University of Sheffield.
- Chen, M.-Y. ve ark., 2001. A Systematic Neuro-Fuzzy Modeling Framework with Application to Material Property Prediction. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 31(5), 781-790s.

- Chen, M.-Y. ve ark., 2004. Rule-base self- generation and simplification for data-driven fuzzy models. *Fuzzy Sets and Systems*, 142, 243-265s.
- Chng, E.S. ve ark., 1996. Gradient radial basis function networks for nonlinear and nonstationary time series prediction. *IEEE Transactions On Neural Networks*, 7(1), 190-194s.
- Cios, K.J. ve ark., 2007. *Data Mining A Knowledge Discovery Approach*. Springer.
- Day, D.S., . *Important Factors in Surface EMG Measurement*. Bortec Biomedical Ltd.
- Delsys Inc., 2002. *Surface Electromyography: Detection and Recording*. Boston, MA.
- Doerschuk, P.C. ve ark., 1983. Upper Extremity Limb Function Discrimination Using EMG Signal Analysis. *Biomedical Engineering, IEEE Transactions on*, BME-30(1), 18 -29s.
- Duda, R.O. ve ark., 2001. *Pattern Classification*. New York: Wiley-Interscience.
- Duro, R.J. ve ark., 1999. Discrete-time backpropagation for training synaptic delay-based artificial neural networks. *IEEE Transactions on Neural Networks*, 10(4), 779 - 789s.
- Englehart, K. ve ark., 2001. Multifunction control of prostheses using the myoelectric signal. In Teodorescu, H.-N.L. ve ark. *Intelligent Systems and Technologies in Rehabilitation Engineering*. CRC Press LLC. Ch. 5.
- Englehart, K. ve ark., 2003. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Transactions Biomedical Engineering*, 50 (7), 848-854s.
- Englehart, K. ve ark., 2001. A wavelet-based continuousclassification scheme for multifunctionmyoelectric control. *IEEE Trans. Biomed. Eng.*, 48(3), 302-311s.
- Englehart, K. ve ark., 1999. Classification of the myoelectric signal using time-frequency based representations. *Medical Engineering & Physics*, 21(6-7), 431-438s.
- Fan, J.-L. ve ark., 2000. A modified partition coefficient. In *WCCC-ICSP 2000. 5th International Conference on.*, 2000.
- Farrell, T.R., 2011. Determining delay created by multifunctional prosthesis controllers. *Journal of Rehabilitation Research and Development* , 48(6), xxi-xxxviii s.

- Farrell, T.R. ve ark., 2007. The Optimal Controller Delay for Myoelectric Prostheses. *Transactions On Neural Systems And Rehabilitation Engineering*, 15(1), 111-118s.
- Frank, H. ve ark., 2000. *Fuzzy Cluster Analysis Methods for Classification, Data Analysis and image recognition*. Wiley.
- Freeman, J.A. ve ark., 1991. *Neural Networks Algorithms, Applications and Programming Techniques*. Addison-Wesley.
- Gath, I. ve ark., 1989. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), 773-80s.
- Gomm, J.B. ve ark., 2000. Selecting radial basis function network centers with recursive orthogonal least squares training. *IEEE Transactions on Neural Networks*, 11(2), 306-314s.
- Goswami, J.C. ve ark., 1999. *Fundamentals of Wavelets: Theory, Algorithms, and Applications*. A Wiley - Interscience Publication.
- Graupe, D. ve ark., 1978. A microprocessor system for multifunctional control of upper-limb prostheses via myoelectric signal identification. *Automatic Control, IEEE Transactions on*, 23(4), 538 - 544s.
- Gutierrez-Osuna, R., . *PRISM Lectures*. [Online] Texas A&M University Available at: http://research.cs.tamu.edu/prism/lectures/pr/pr_110.pdf [Accessed Mayıs 2011].
- Hagan, M.T. ve ark., 1994. Training feedforward networks with the Marquardt algorithm. *Neural Networks, IEEE Transactions on*, 5(6), 989-993s.
- Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*. New Jersey: Prentice Hall.
- Hogan, N. ve ark., 1980. Myoelectric Signal Processing: Optimal Estimation Applied To Electromyography-Part I: Derivation Of The Optimal Myoprocessor. *BME-27*(7), 382-395s.
- Hsu, C.W. ve ark., 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2), 415-425s.
- Huang, H.-P. ve ark., 2003. EMG classification for prehensile postures using cascaded architecture of neural networks with self-organizing maps. In *ICRA '03. IEEE International Conference on*, 2003.

- Hudgins, B. ve ark., 1993. A new strategy for multifunction myoelectric control. *Biomedical Engineering, IEEE Transactions on*, 40(1), 82 -94s.
- Jang, J.S.R. ve ark., 1997. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine intelligence*. Prentice-Hall, Inc.
- Jolliffe, I.T., 2002. *Principal Component Analysis*. 2nd ed. Springer - Verlag New York, Inc.
- Jun-Uk, C. ve ark., 2007. A Supervised Feature-Projection-Based Real-Time EMG Pattern Recognition for Multifunction Myoelectric Hand Control. 12(3).
- Jun-Uk, C. ve ark., 2006. A Real-Time EMG Pattern Recognition System Based on Linear-Nonlinear Feature Projection for a Multifunction Myoelectric Hand. 53(11).
- Karlik, B. ve ark., 2003. A fuzzy clustering neural network architecture for multifunction upper-limb prosthesis. *IEEE Transactions on Biomedical Engineering*, 50(11), 1255–1261s.
- Kayhan, G. ve ark., 2009. Designing pre-processing units for RBF networks part-1: Initial structure identification. In *INISTA 2009 International Symposium on Innovations in Intelligent Systems and Applications*. Trabzon,Türkiye, 2009.
- Keerthi, S.S. ve ark., 2003. Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel. *Neural Computation*, 15(7), 1667-1689s.
- Kermani, B.G. ve ark., 2005. Performance of the Levenberg-Marquardt neural network training method in electronic nose applications. *Sensors and Actuators B: Chemical*, 1(13 - 22), 110s.
- Khushaba, R.N. ve ark., 2007. Fuzzy wavelet packet based feature extraction method for multifunction myoelectric control. *International Journal Of Biomedical Sciences*, 2, 186-194s.
- Klawonn, F. ve ark., 2003. What is Fuzzy About Fuzzy Clustering? Understanding and Improving the Concept of the Fuzzifier. In M.R. Berthold et al., eds. *Advances in Intelligent Data Analysis V*. Berlin. 254-264s.
- Konrad, P., 2005. *The ABC of EMG A Practical Introduction to Kinesiological Electromyography*. Noraxon INC.
- Kuiken, T.A. ve ark., 2009. Targeted Muscle Reinnervation for Real-time Myoelectric Control of Multifunction Artificial Arms. *JAMA*, 301(6), 619-628s.

- Kuruganti, U. ve ark., 1995. Two-channel enhancement of a multifunction control system. *Biomedical Engineering, IEEE Transactions on*, 42(1), 109 -111s.
- Kwatny, E. ve ark., 1970. An Application of Signal Processing Techniques to the Study of Myoelectric Signals. *BME-17*(4), 303-313s.
- Lindstrom, L.H. ve ark., 1977. Interpretation of Myoelectric Power Spectra:A Model and Its Applications. *Proceedings of the IEEE*, 65(5), 653-662s.
- Linkens, D.A. ve ark., 1999. Input selection and partition validation for fuzzy modelling using neural network. *Fuzzy Sets and Systems*, 107, 299-308s.
- Liu, Y.-H. ve ark., 2007. Recognition of Electromyographic Signals Using Cascaded Kernel Learning Machine. *Mechatronics, IEEE/ASME Transactions on*, 12(3), 253 -264s.
- Luca, C.J.D., 1979. Physiology and Mathematics of Myoelectric Signals. *IEEE Transaction on Biomedical Engineering*, BME-26(6), pp.313-324.
- Luca, C.J.D., 2002. *Surface Electromyography:Detection And Recording*. DelSys Incorporated.
- Luca, C.J.D., 2003. *The Use of Surface Electromyography In Biomechanics*. Delsys.
- Mallat, S.G., 1989. A Theory for multiresolution signal decomposition:the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2).
- Mallat, S.G., 1999. *A Wavelet Tour of Signal Processing*. Academic Press.
- Martinez, A.M. ve ark., 2001. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), 228 -233s.
- Martinez, W.L. ve ark., 2005. *Exploratory Data Analysis with Matlab*. USA: Chapman & Hall.
- Misiti, M. ve ark., 2002. *Wavelet Toolbox For Use With Matlab*. The MathWorks.
- Momen, K. ve ark., 2007. Real-Time Classification of Forearm Electromyographic Signals Corresponding to User-Selected Intentional Movements for Multifunction Prosthesis Control. *IEEE Transactions On Neural Systems And Rehabilitation*, 15(4).
- Montazer, G.A. ve ark., 2009. Three-phase strategy for the OSD learning method in RBF neural networks. *Neurocomputing*, 72(7-9), 1797-1802s.

- Muzumdar, A., 2004. *Powered Upper Limb Prostheses: Control, Implementation and Clinical Application*. 1st ed. Springer.
- Neruda, R. ve ark., 2005. Learning methods for radial basis function networks. *Future Generation Computer Systems*, 21(7), 1131-1142s.
- Nie, J., 1995. Construction Fuzzy Model by Self Organizing Counter Propagation Network. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6), 963-970s.
- Nie, J. ve ark., 1994. Fast self-learning multivariable fuzzy controllers constructed from a modified CPN network. *International Journal of Control*, 60(3), 369-393s.
- Oskoei, M.A. ve ark., 2007. Myoelectric control systems-A survey. *Biomedical Signal Processing and Control*, 2(4), 275-294s.
- Ottobock, 2011. *Targeted Muscle Reinnervation*. [Online] Available at: http://www.ottobock.com.tr/cps/rde/xbcr/ob_tr_tr/646D385-GB-03-1006w.pdf [Accessed Ağustos 2011].
- Özdemir, A.E.Ö. ve ark., 2009. Designing pre-processing units for RBF networks part-2: Final identification and course tuning of parameters. In *INISTA 2009 International Symposium on INnovations in Intelligent SysTems and Applications, 190s*. Trabzon, Türkiye, 2009.
- Özdemir, A.E. ve ark., 2009. A Real-Time EMG Driven Virtual Prosthesis Hand. In *CLAWAR 2009 12. International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. İstanbul, 2009.
- Pan, T.-T. ve ark., 2004. Mechatronic Experiments Course Design : A Myoelectric Controlled Partial-Hand Prosthesis Project. *IEEE Transactions On Education*, 47(3), 348-355s.
- Raez, M.B. ve ark., 2006. Techniques of EMG signal analysis: detection, processing, classification and applications. *Biological procedures online*, 8, 11-35s.
- Rioul, O. ve ark., 1991. Wavelets and Signal Processing. *IEEE Signal Processing Magazine*, 8(4), 14-38s.
- Saridis, G.N. ve ark., 1982. EMG Pattern Analysis and Classification for a Prosthetic Arm. *BME-29*(6), 403 -412s.

- Schölkopf, B. ve ark., 1997. Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. *IEEE Transaction on Signal Processing*, 45(11), 2758-2765s.
- Schwenker, F. ve ark., 2001. Three learning phases for radial-basis-function networks. *Neural Networks*, 14(4-5), 439-458s.
- Shenoy, P. ve ark., 2008. Online Electromyographic Control of a Robotic Prosthesis. *IEEE Transactions on Biomedical Engineering*, 55(3), 1128--1135s.
- Shing, J. ve ark., 1993. ANFIS : Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on systems, man, and cybernetics*, 23(3), 665-685s.
- Shlens, J., 2005. *A Tutorial on Principal Component Analysis*. San Diego: Institute for Nonlinear Science, University of California.
- Smith, L., 2002. *A tutorial on Principal Components Analysis*. [Online] Available at: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.
- Sörnmo, L. ve ark., 2005. *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. New York: Academic.
- Staiano, A. ve ark., 2006. Improving RBF networks performance in regression tasks by means of a supervised fuzzy clustering. *Neurocomputing*, 69(13-15), 1570-1581s.
- Sun, H. ve ark., 2004. FCM-Based Model Selection Algorithms for Determining the Number of Clusters. *Pattern Recognition*, 37(10), 2027 - 2037s.
- Tenore, F. ve ark., 2009. Decoding of individuated finger movements using surface electromyography. *Biomedical Engineering, IEEE Transactions on*, 56(5), 1427 –1434s.
- Vapnik, V.N., 1998. *Statistical Learning Theory*. New York: Wiley-Interscience.
- Vojtech, F. ve ark., 2004. *Statistical Pattern Recognition Toolbox for MATLAB(SPRTOOL), User's Guide*.
- Wilamowski, B.M. ve ark., 1999. Efficient Algorithm for Training Neural Networks with one Hidden Layer. In *IJCNN '99. International Joint Conference on..*, 1999.
- Wolkenhaue, O., . *Fuzzy Clustering*. [Online] UMIST [Accessed 16 Haz. 2011].
- Yoshida, K. ve ark., 2010. Multichannel Intraneural and Intramuscular Techniques for Multiunit Recording and Use in Active Prostheses. *Proceedings of the IEEE*, 98(3), 432 - 449s.

- Zhang, H., 2005. *Objective Function Based Fuzzy Clustering*. [Online]
Available at: http://www.cse.uconn.edu/cms/samples/Zhang_presentation.ppt
[Accessed 2006].
- Zheng, N. ve ark., 2009. *Statistical Learning and Pattern Analysis for Image and Video Processing*. London: Springer-Verlag London Limited.
- Zouridakis, G. ve ark., 2004. *Biomedical Technology and Devices Handbook*. CRC Press.

7. EKLER

EK-A : KYA Algoritması Matlab Programı

```
function cpmnerkez=cpnga(x,z,delta)
%-----
% Giriş: x ... normalize sistem giriş verisi
%      z ... normalize sistem çıkış verisi
%      delta ..yarıçap
%-----
% Çıkış : cpmnerkez ... yarıçapa göre bulunan merkezler
%----- Başlangıç Değerleri-----
[n1 n2]=size(z);
teta(1)=zeros(1,n2);          %Başlangıç Grossberg ağırlığı
nt=1; l=1; j=1; ns(1)=1;
w(1,:)=x(1,:);               %Başlangıç Kohonen ağırlığı
yinele=500;
%-----Yineleme/Güncelleme-----
for a=1:yinele
    while l~=(n1+1)
        for s=1:j
            d1(s)=norm(x(l,:)-w(s,:)); % Uzaklık Hesabı
        end
        d=min(d1);                % min. uzaklık
        [j1,j2]=find(d1==min(d1)); % min uzaklığa bağlı katın konumu

        if d<delta
            teta(j2)=teta(j2)+0.7*(z(l)-teta(j2)); % Grossberg ağırlıkları
            y(l)=teta(j2);
            ksi(l,j2)=1;
            alfa=0.7/(ns(j2));          % Öğrenme parametresi
            w(j2,:)=w(j2,:)+alfa*(x(l,:)-w(j2,:)); % Kohonen ağırlıkları
            ns(j2)=ns(j2)+1;           % Aktivasyon
```

```

        l=l+1;                % Veri indis artışı
    else
        j=j+1;                % Yeni birim indisi
        w(j,:)=x(1,:);        % Yeni birim kohonen ağırlıkları
        ns(j)=1;              % Yeni birim aktivasyonu
        if a==1
            teta(j)=zeros(1,n2); % Yeni birim grossberg ağırlıkları
        end
    end
end
end
end

%----- Sonuç-----
cm= length(teta);
for s=1:cm
    cpmmerkez(s,:)= [w(s,:) teta(s)]; % Giriş çıkış verisi aynı matris [giriş çıkış]
    biçimli birleştiriliyor
end

```

EK-B : BCO Algoritması Matlab Programı

```

function mrkz=fcmga(x,cm,merkez,tol)
%-----
% Giriş: x ... n1xn2 boyutlu veri
%      cm ... küme sayısı
%      tol ... sonlandırma kriteri (tol > 0)
%      merkez..merkezlerin başlangıç değerleri
%-----
% Çıkış : mrkz ... güncellenmiş merkez
%----- Başlangıç Değerleri -----
[n1 n2]=size(x);          % veri boyutu
m=2;                      % ağırlıklandırma üsteli
opt_kume=cm;               % maks. küme sayısı
yinele=200;                % maks. yineleme
%-----Yineleme/Güncelleme-----
for c=1:cm
    for dn=1:yinele        % Döngü (1)
        for i=1:c
            for j=1:n1
                d(i,j)=norm(merkez(i,:)-x(j,:));
                if d(i,j)==0
                    d(i,j)=0.000001;
                end
            end
        end
        u=(d.^(-2/(m-1)))./(ones(c,1)*sum(d.^(-2/(m-1))));
        merkez_yeni = (u.^m)*x./((ones(n2, 1)*sum((u.^m)'))'); % yeni merkez
        % ones(n2, 1) toplamı verinin sütun sayısı kadar çoğaltır
        fark=norm(merkez(1:c,:)-merkez_yeni);
        if fark<tol
            dn=yinele;      % Döngü(1) Kesme
        else

```

```

        merkez(1:c,:)=merkez_yeni;
    end
end
merkezdepo{c}=merkez_yeni;
uy{c}=u;
hece(c)=-(max(sum(u).*log(max(u))))/n1;    % doğrulama kriteri
end
%----- Optimum Küme Karar Aşaması-----
if cm>3
    for i=2:cm-1
        if ((hece(i)<hece(i+1))&&(hece(i)<hece(i-1)))
            opt_kume=i;
            break;
        end
    end
end
mrkz=merkezdepo{opt_kume};    % optimum küme merkezleri

```

EK-C: GK algoritması Matlab Programı

```

function prmtr = gkga(Z,cm,merkez,tol)
%-----
% Giriş: Z ... Nxn boyutlu veri
%      cm ... küme sayısı
%      tol ... sonlandırma kriteri (tol > 0)
%      merkez..merkezlerin başlangıç değerleri
%-----
% Çıkış : prmtr ... küme merkez ve genişlikleri
%----- Başlangıç Değerleri -----
if (nargin < 5), tol = 1e-4;
elseif isempty(tol), tol = 1e-4; end;          % sonlandırma
opt_kume=cm;                                %Maks. küme sayısı
yinele=200;                                %Maks. yineleme
for c=1:cm
    clear sd cf;
    %----- Matrislerin oluşturulması -----
    [N,n] = size(Z);                % veri boyutu
    N1 = ones(N,1); n1 = ones(1,n); c1 = ones(1,c); % başlangıç matrisleri
    U = zeros(N,c);                % üyelik matrisi
    d = U;                        % uzaklık matrisi
    F = zeros(n,n,c);            % kovaryans matrisi
    m=2;                        % Ağırlıklandırma üsteli
    gamma=0;
    beta = 1e15;
    %----- Başlangıç U hesabı -----
    V=merkez(1:c,:);              %ham merkezler
    f0= eye(n)*det(cov(Z)).^(1/n);
    for j = 1 : c,
        ZV = Z - N1*V(j,:);
        d(:,j) = sum((ZV.^2)')';    % uzaklıklar
    end;

```

```

d = (d+1e-10).^(-1/(m-1));
U0 = (d ./ (sum(d,2)*c1));          % bölümlene matrisi
%----- yineleme/güncelleme -----
for dn=1:yinele                      % Döngü(1)
    U = U0; Um = U.^m; sumU = sum(Um);          % başlangıç değerleri
    V1 = (Um'*Z)./(sumU'*n1);                  % küme merkezleri
    for j = 1 : c,                            % tüm kümeler için parametre güncelleme
        ZV = Z - N1*V1(j,:);
        f = n1'*Um(:,j)'.*ZV'*ZV/sumU(j);
        f = (1-gamma)*f+gamma*(f0.^(1/n));
        if cond(f)> beta;
            [ev,ed]=eig(f); ed_max = max(diag(ed));
            ed(beta*ed < ed_max) = ed_max/beta;
            f = ev*diag(diag(ed))*inv(ev);
        end;
        d(:,j)=sum(ZV*(det(f)^(1/n)*pinv(f)).*ZV,2);
    end;
    d = (d+1e-10).^(-1/(m-1));
    U1 = (d ./ (sum(d,2)*c1));
    if max(max(U1-U))<tol
        dn=yinele;                          %Döngü(1) kesme
    else
        V(1:c,:)=V1;
    end
end
%----- Sonuç F ve U -----
U = U0; Um = U.^m; sumU = n1'*sum(Um);
MV = V-c1'*mean(V);  %(Vi-Vort) Merkezlerden ortalamalar çıkarıldı
MK=MV*MV';  % (Vi-Vort)*(Vi-Vort)'
for j = 1 : c,
    ZV = Z - N1*V(j,:);
    sd(j,:)=sum(N1*MK(j,:).*Um);  % Kümeler arası dağılım matrisi

```

```

cf{j} = n1'*Um(:,j)'.*ZV'*ZV/sumU(1,j); % fuzzy kovaryans matrisi
cd(j)=trace(cf{j}); % Fuzzy kovaryans özdeğer toplamı
R3(j,:)=2.*sqrt(eig(cf{j})(1:n-1,1:n-1))); % genişlik hesabı
end;
sm(c)=trace(sd); % Kümeler arası dağılım matrisi özdeğer toplamı
merkezdepo{c}=V; uy{c}=U; gen{c}=R3;
end
%----- Optimum Küme Karar Aşaması-----
vsc=sm./cd; % Bölüm doğrulama kriteri
[idx idy]=find(vsc==max(vsc)); % Bölüm doğrulama kriteri maks olanı
gen1=gen{idy}; %genişlik
mrkz1=merkezdepo{idy}; %merkez
prmtr={mrkz1 gen1 }; % döndürülen parametreler

```

EK-D: RTİ Ağının GA Algoritması İle Eğitim için Matlab Programı

```

function rbfprmtr=rbfga(x,y,merkez,genislik,agirlik,iter,mkat,gkat,akat)
%-----
% Giriş: x ... normalize sistem giriş verisi
%      y ... normalize sistem çıkış verisi
%      merkez... merkezlerin başlangıç değerleri
%      genislik..genişliklerin başlangıç değerleri
%      agirlik ..ağırlıkların başlangıç değerleri
%      iter... yineleme sayısı
%      mkat... merkezler için öğrenme parametresi
%      gkat... genişlikler için öğrenme parametresi
%      akat... ağırlıklar için öğrenme parametresi
%-----
% Çıkış : rbfprmtr ... rbfprmtr {merkez genislik ağırlık hata} bilgilerini kapsar.
Örneğin;
% rbfprmtr{1}=merkez rbfprmtr{2}=genişlik rbfprmtr{3}=ağırlık
% rbfprmtr{4}=son yinelemedeki hata kareler toplamı

%-----Başlangıç parametrelerinin atama denetimi -----
if (nargin < 6), iter = 500;
elseif isempty(iter), iter=500; end;
if (nargin < 7), mkat = 1e-2;
elseif isempty(mkat), mkat = 1e-2; end;
if (nargin < 8), gkat = 1e-2;
elseif isempty(gkat), gkat = 1e-2; end;
if (nargin < 9), akat = 1e-2;
elseif isempty(mkat), akat = 1e-2; end;
%.....
[m,n]=size(x); [k,p]=size(y);
[o,z]=size(merkez); rbfsayisi=o; yonelim=z; hatadizi=[];
%-----RBF AĞININ EĞİTİMİ-----
for i=1:iter % her bir örnek ağa "iter" değeri kadar gösterilecek

```

```

p=0;
for j=1:m % giriş setindeki tüm örnekler sıra ile ağı gösterilecekler
    v=x(j,:);
    % a) Gizli katmandaki RBF lerin ürettiği çıkışlar kendilerine
    % ait ağırlıklarla çarpılarak ağı üreteceği çıkış hesaplanır
    toplam=0;
    for bb=1:rbfSayisi
        cikislar=rbfcga(merkez(bb,:),genislik(bb,:),v)*agirlik(bb);
        toplam=toplam+cikislar;
    end
    % b) olması gereken çıkış ve ağı ürettiği çıkış arasındaki fark=hata hesaplanır.
    hata=y(j)-toplam;
    % c) hata ,öğrenme katsayısı ve eski değerleri kullanılarak ağırlıklar ,merkezler ve
    genişlikler eğitilir
    for bb=1:rbfSayisi
        for mem=1:yonelim
            merkez1(bb,mem)=merkez(bb,mem)+mkat*hata...
            * agirlik(bb)* rbfcga(merkez(bb,:),genislik(bb,:),v)...
            *(v(:,mem)-merkez(bb,mem))/((genislik(bb,mem))^2);
            genislik1(bb,mem)=genislik(bb,mem)+gkat*hata...
            *agirlik(bb)* rbfcga(merkez(bb,:),genislik(bb,:),v)...
            *((v(:,mem)-merkez(bb,mem))^2)/((genislik(bb,mem))^3);
        end
        agirlik1(bb)=agirlik(bb)+hata*akat*rbfcga(merkez(bb,:),genislik(bb,:),v);
    end
    merkez=merkez1; agirlik=agirlik1; genislik=genislik1;    p=p+(hata)^2;
end
    hatadizi=[hatadizi p];
end
    hatadizi=sqrt(hatadizi./m); % RMS cinsinden hata hesabı
    rbfprmtr={merkez genislik agirlik hatadizi}; % Çıkış parametreleri

```

```

..... ALT İŞLEV .....
function [c]=rbfcga(m,g,v)
% bir RBF'in belli bir merkez ve genişlik parametresi için
% anlık bir veri için üreteceği çıkışı hesaplayan fonksiyondur.
% m: merkez   [m1 m2 ...mn] (1xn boyutlu)
% g: genişlik  [g1 g2 ...gn] (1xn boyutlu)
% v: anlık veri [v1 v2 ...vn] (1xn boyutlu giriş verisi)
% c: RBF çıkışı
% cikis=exp(- ( ((v1-m1)/g1)^2 + ((v2-m2)/g3)^2 + ...+((vn-mn)/gn)^2   ) )
s=( (v-m) ./g ).^2;
k=sum(s);
c=exp(-k);

```

EK-E: DANRTİ Ağının LM- GA ile Eğitimi için Matlab Programı

```

function rbfprmtr=Nrbf(x,y,merkez,genislik,agirlik,iter,mkat,gkat,lamda)
%-----
% Giriş: x ... normalize sistem giriş verisi
%      y ... normalize sistem çıkış verisi
%      merkez... merkezlerin başlangıç değerleri
%      genislik..genişliklerin başlangıç değerleri
%      agirlik ..ağırlıkların başlangıç değerleri ( bir boyutlu )
%      iter... yineleme sayısı
%      mkat... merkezler için öğrenme parametresi
%      gkat... genişlikler için öğrenme parametresi
%      lamda.. Marquardt parametresi
%-----
% Çıkış : rbfprmtr ... rbfprmtr {merkez genislik ağırlık hata} bilgilerini kapsar.
Örneğin;
% rbfprmtr{1}=merkez rbfprmtr{2}=genislik rbfprmtr{3}=ağırlık
% rbfprmtr{4}=son yinelemedeki hata kareler toplamı
%-----Başlangıç parametrelerinin atama denetimi -----
if (nargin < 6), iter = 500;
elseif isempty(iter), iter=500; end;
if (nargin < 7), mkat = 1e-2;
elseif isempty(mkat), mkat = 1e-2; end;
if (nargin < 8), gkat = 1e-2;
elseif isempty(gkat), gkat = 1e-2; end;
if (nargin < 9), lamda = 1e-2;
elseif isempty(lamda), lamda = 1e-2; end;
%-----
[m,n]=size(x);
hatadizi=[];
[rbfsayisi,yonelim]=size(merkez);
agirlik=zeros(rbfsayisi,n) agirlik];
for i1=1:iter

```

```

for j=1:m
    v=x(j,:);
    toplam=0;    toplam2=0;
    for bb=1:rbfsayisi %k.RBF için çıkış değerini hesapla
        tw=agirlik(bb,1:size(agirlik,2)-1);
        vw=tw.*v;
        sw(bb)=sum(vw)+agirlik(bb,size(agirlik,2));
        cikislar(bb)=rbfcga(merkez(bb,:),genislik(bb,:),v);
        toplam=toplam+cikislar(bb).*sw(bb);
        toplam2=toplam2+cikislar(bb);
    end
    yrbf=toplam/toplam2;
    hata=y(j)-yrbf;
    merkez1=merkez+mkat.*hata.*(ones(rbfsayisi,1)*v...
        -merkez).*(cikislar'*ones(1,n)).*((sw.*toplam2-toplam)*ones(1,n))...
        ./((toplam2^2).*(genislik.^2));
    genislik1=genislik+gkat.*hata.*((ones(rbfsayisi,1)*v...
        -merkez).^2).*(cikislar'*ones(1,n)).*((sw.*toplam2-toplam)*ones(1,n))...
        ./((toplam2^2).*(genislik.^3));
    p(j)=(hata)^2;
    merkez=merkez1;    genislik=genislik1;
end
hatadizi=[hatadizi sum(p)];
if i>=3
    if hatadizi(i)<hatadizi(i-1)
        gkat=gkat/10;
        mkat=mkat/10;
    else
        gkat=gkat*2;
        mkat=mkat*2;
    end
end
end

```

```

lm_prmtr =Nrbf_lm(x,y,merkez,genislik,agirlik,lamda);
agirlik=lm_prmtr{1};
lamda=lm_prmtr{4};
end
rbfprmtr={merkez genislik agirlik lm_prmtr{2} lm_prmtr{3}}; % Çıkış parametreleri

```

```

..... ALT İŞLEV .....
function lm_prmtr =Nrbf_lm(x,d,c,g,w,lamda)
% (NRTİ ağırlıklarının LM algoritması ile eğitimi)
%-----
% Giriş: x ... normalize sistem giriş verisi
%      d ... normalize sistem çıkış verisi
%      c ... merkezlerin başlangıç değerleri
%      g ... genişliklerin başlangıç değerleri
%      w ... ağırlıkların başlangıç değerleri
%      lamda... Marquardt parametresi
%-----
% Çıkış : lm_prmtr ... çıkış ağırlıkları ( $w_{lm} = ax_1 + \dots + bx_n + d$ )
L=size(x,1); % L->veri sayısı
M=size(c,1); % M->RBF sayısı
psay=M*size(w,2); % (a,b,...) parametre sayısı
updatej=1;
%%----- LM ile ağırlık eğitimi -----
for i=1:100
    if updatej==1
        % 1) Tüm veriler için hata ve y->çıkışlar matrisini oluştur
        y1=[];
        for j=1:L
            v0=x(j,:); toplam0=0; toplam20=0;
            for bb=1:M %k.RBF için çıkış değerini hesapla
                tww=w(bb,1:size(w,2)-1);

```

```

    vw=tww.*v0;
    sww=sum(vw)+w(bb,end);
    cikislar=rbfcga(c(bb,:),g(bb,:),v0);
    toplam0=toplam0+cikislar.*sww;
    toplam20=toplam20+cikislar;
end
yrbf=toplam0/toplam20;
y1=[y1;yrbf];
end
e=d-y1;
if i==1
hata1=dot(e,e);
end
pause(0.001); plot(d,'b*');hold on;plot(y1,'ro');hold off;
%% 1) -----
% 2) jakobien matrisini hesapla
for j=1:L
    v0=x(j,:);    is=0;    fitop=0;
    for zz=1:M
        fitop=fitop+rbfcga(c(zz,:),g(zz,:),v0);
    end
    for k=1:M %k.RBF için çıkış değerini hesapla
        is=is+1;
        cik=rbfcga(c(k,:),g(k,:),v0);
        for n=1:size(v0,2)
            jak(j,is)=-cik*v0(:,n)/fitop;    is=is+1;
        end
        jak(j,is)=-cik/fitop;
    end
end
end
end
% 2) -----

```

```

% 3) Gradyan hesapla
gr=inv(jak'*jak+lamda*eye(psay,psay))*jak'*e;
% 3) -----
% 4) RBF lerin ağırlık parametrelerini hesapla
gr=reshape(gr,size(w,2),M)';
w1=w-gr;
% 5) Hata vektörünün elemanları sabitlenmişse döngüden çık
y2=[];
for j=1:L
    v0=x(j,:);    toplam0=0;    toplam20=0;
    for bb=1:M %k.RBF için çıkış değerini hesapla
        tww=w(bb,1:size(w,2)-1);
        vw=tww.*v0;    sww=sum(vw)+w1(bb,end);
        cikislar=rbfcga(c(bb,:),g(bb,:),v0);
        toplam0=toplam0+cikislar.*sww;
        toplam20=toplam20+cikislar;
    end
    yrbf=toplam0/toplam20;    y2=[y2;yrbf];
end
e_lm=d-y2;    hata2=dot(e_lm,e_lm);
if hata2<hata1
    lamda=lamda/10;    w=w1;    hata1=hata2;    updatej=1;
else
    updatej=0;    lamda=lamda*10;
end
end
w_lm=w;    lm_prmtr={w_lm d y1 lamda};

```

EK-F: DDA Matlab Programı (Cai ve ark. , 2007)

```
function [eigvector, eigvalue, elapse] = LDA(gnd,options,data)
% Deng Cai tarafından yazılmıştır
if ~exist('data','var')
    global data;
end
if (~exist('options','var'))
    options = [];
end
if ~isfield(options,'Regu') | ~options.Regu
    bPCA = 1;
    if ~isfield(options,'PCARatio')
        options.PCARatio = 1;
    end
else
    bPCA = 0;
    if ~isfield(options,'ReguType')
        options.ReguType = 'Ridge';
    end
    if ~isfield(options,'ReguAlpha')
        options.ReguAlpha = 0.1;
    end
end
end
tmp_T = cputime;
[nSmp,nFea] = size(data);
if length(gnd) ~= nSmp
    error('gnd and data mismatch!');
end
classLabel = unique(gnd);
nClass = length(classLabel);
Dim = nClass - 1;
if bPCA & isfield(options,'Fisherface') & options.Fisherface
```

```

    options.PCARatio = nSmp - nClass;
end
if issparse(data)
    data = full(data);
end
sampleMean = mean(data,1);
data = (data - repmat(sampleMean,nSmp,1));
bChol = 0;
if bPCA & (nSmp > nFea+1) & (options.PCARatio >= 1)
    DPrime = data'*data;
    DPrime = max(DPrime,DPrime');
    [R,p] = chol(DPrime);
    if p == 0
        bPCA = 0;
        bChol = 1;
    end
end
if bPCA
    if nSmp > nFea
        ddata = data'*data;
        ddata = max(ddata,ddata');
        [eigvector_PCA, eigvalue_PCA] = eig(ddata);
        eigvalue_PCA = diag(eigvalue_PCA);
        clear ddata;
        maxEigValue = max(abs(eigvalue_PCA));
        eigIdx = find(eigvalue_PCA/maxEigValue < 1e-12);
        eigvalue_PCA(eigIdx) = [];
        eigvector_PCA(:,eigIdx) = [];
        [junk, index] = sort(-eigvalue_PCA);
        eigvalue_PCA = eigvalue_PCA(index);
        eigvector_PCA = eigvector_PCA(:, index);
        if options.PCARatio > 1

```

```

    idx = options.PCARatio;
    if idx < length(eigvalue_PCA)
        eigvalue_PCA = eigvalue_PCA(1:idx);
        eigvector_PCA = eigvector_PCA(:,1:idx);
    end
elseif options.PCARatio < 1
    sumEig = sum(eigvalue_PCA);
    sumEig = sumEig*options.PCARatio;
    sumNow = 0;
    for idx = 1:length(eigvalue_PCA)
        sumNow = sumNow + eigvalue_PCA(idx);
        if sumNow >= sumEig
            break;
        end
    end
    eigvalue_PCA = eigvalue_PCA(1:idx);
    eigvector_PCA = eigvector_PCA(:,1:idx);
end
eigvalue_PCA = eigvalue_PCA.^-.5;
data = (data*eigvector_PCA).*repmat(eigvalue_PCA',nSmp,1);
else
    ddata = data*data';
    ddata = max(ddata,ddata');
    [eigvector, eigvalue_PCA] = eig(ddata);
    eigvalue_PCA = diag(eigvalue_PCA);
    clear ddata;
    maxEigValue = max(eigvalue_PCA);
    eigIdx = find(eigvalue_PCA/maxEigValue < 1e-12);
    eigvalue_PCA(eigIdx) = [];
    eigvector(:,eigIdx) = [];
    [junk, index] = sort(-eigvalue_PCA);
    eigvalue_PCA = eigvalue_PCA(index);

```

```

eigvector = eigvector(:, index);
if options.PCARatio > 1
    idx = options.PCARatio;
    if idx < length(eigvalue_PCA)
        eigvalue_PCA = eigvalue_PCA(1:idx);
        eigvector = eigvector(:,1:idx);
    end
elseif options.PCARatio < 1
    sumEig = sum(eigvalue_PCA);
    sumEig = sumEig*options.PCARatio;
    sumNow = 0;
    for idx = 1:length(eigvalue_PCA)
        sumNow = sumNow + eigvalue_PCA(idx);
        if sumNow >= sumEig
            break;
        end
    end
    eigvalue_PCA = eigvalue_PCA(1:idx);
    eigvector = eigvector(:,1:idx);
end
eigvalue_PCA = eigvalue_PCA.^-.5;
eigvector_PCA = (data'*eigvector).* repmat(eigvalue_PCA',nFea,1);
data = eigvector;
clear eigvector;
end
else
    if ~bChol
        DPrime = data'*data;
        switch lower(options.ReguParam)
            case {lower('Ridge')}
                for i=1:size(DPrime,1)
                    DPrime(i,i) = DPrime(i,i) + options.ReguParam;

```

```

        end
        case {lower('Tensor')}
            DPrime = DPrime + options.RegAlpha*options.regularizerR;
        case {lower('Custom')}
            DPrime = DPrime + options.RegAlpha*options.regularizerR;
        otherwise
            error('ReguType does not exist!');
        end
        DPrime = max(DPrime,DPrime');
    end
end
[nSmp,nFea] = size(data);
Hb = zeros(nClass,nFea);
for i = 1:nClass,
    index = find(gnd==classLabel(i));
    classMean = mean(data(index,:),1);
    Hb (i,:) = sqrt(length(index))*classMean;
end
elapsed.timeW = 0;
elapsed.timePCA = cputime - tmp_T;
tmp_T = cputime;
if bPCA
    [dumpVec,eigvalue,eigvector] = svd(Hb,'econ');
    eigvalue = diag(eigvalue);  eigIdx = find(eigvalue < 1e-3);
    eigvalue(eigIdx) = [];  eigvector(:,eigIdx) = [];  eigvalue = eigvalue.^2;
    eigvector = eigvector_PCA*(repmat(eigvalue_PCA,1,length(eigvalue)).*eigvector);
else
    WPrime = Hb'*Hb;      WPrime = max(WPrime,WPrime');      dimMatrix =
size(WPrime,2);
    if Dim > dimMatrix
        Dim = dimMatrix;
    end
end

```

```

if isfield(options,'bEigs')
    if options.bEigs
        bEigs = 1;
    else
        bEigs = 0;
    end
else
    if (dimMatrix > 1000 & Dim < dimMatrix/10) | (dimMatrix > 500 & Dim <
dimMatrix/20) | (dimMatrix > 250 & Dim < dimMatrix/30)
        bEigs = 1;
    else
        bEigs = 0;
    end
end
if bEigs
    option = struct('disp',0);
    if bChol
        option.cholB = 1;
        [eigvector, eigvalue] = eigs(WPrime,R,Dim,'la',option);
    else
        [eigvector, eigvalue] = eigs(WPrime,DPrime,Dim,'la',option);
    end
    eigvalue = diag(eigvalue);
else
    [eigvector, eigvalue] = eig(WPrime,DPrime);
    eigvalue = diag(eigvalue);    [junk, index] = sort(-eigvalue);
    eigvalue = eigvalue(index);    eigvector = eigvector(:,index);
    if Dim < size(eigvector,2)
        eigvector = eigvector(:, 1:Dim);    eigvalue = eigvalue(1:Dim);
    end
end
end

```

```
for i = 1:size(eigvector,2)
    eigvector(:,i) = eigvector(:,i)./norm(eigvector(:,i));
end
elapsed.timeMethod = cputime - tmp_T;
elapsed.timeAll = elapsed.timePCA + elapsed.timeMethod;
```

EK-G : DDA Kullanarak Boyut Azaltma İşlemi için Matlab Programı

```

data=data';
[sat,sut]=size(data);
tt1=[]; tt2=[]; options = []; options.Fisherface = 1;
for i=1:128:sut
    is=data(1,i:i+127);
    [c,l] = wavedec(is,4,'db4');
    a4 = wrcoef('a',c,l,'db4',4);
    d1 = wrcoef('d',c,l,'db4',1);
    d2 = wrcoef('d',c,l,'db4',2);
    d3 = wrcoef('d',c,l,'db4',3);
    d4 = wrcoef('d',c,l,'db4',4);
    v=[a4;d4;d3;d2;d1];
    [eigvector, eigvalue] = LDA(1:5, options, v);
    y = v*eigvector;
    tt1=[tt1 sum(abs(y(2:end))))];
    is=data(2,i:i+127);
    [c,l] = wavedec(is,4,'db4');    a4 = wrcoef('a',c,l,'db4',4);
    d1 = wrcoef('d',c,l,'db4',1);    d2 = wrcoef('d',c,l,'db4',2);
    d3 = wrcoef('d',c,l,'db4',3);    d4 = wrcoef('d',c,l,'db4',4);
    v=[a4;d4;d3;d2;d1];
    [eigvector, eigvalue] = LDA(1:5, options, v);
    y = v*eigvector;    tt2=[tt2 sum(abs(y(2:end))))];
end
O=[tt1; tt2];    c=zeros(112,1);
c(find(and(O(1,:)>0.07,O(2,:)<0.55))))=1;
c(find(and(O(1,:)<0.07,O(2,:)>0.55))))=2;
c(find(and(O(1,:)>0.07,O(2,:)>0.55))))=3;
subplot(211);bar(tt1);hold on;plot(c,'r');
subplot(212);bar(tt2);hold on;plot(c,'r');

```

EK-H : TBA Kullanarak Boyut Azaltma İşlemi için Matlab Programı

```

data=data';
[sat,sut]=size(data);
tt1=[]; tt2=[];
for i=1:128:sut % Süre 128 msn seçilmiştir.
    is=data(1,i:i+127);
    [c,l] = wavedec(is,4,'db4');
    a4 = wrcoef('a',c,l,'db4',4);
    d1 = wrcoef('d',c,l,'db4',1);
    d2 = wrcoef('d',c,l,'db4',2);
    d3 = wrcoef('d',c,l,'db4',3);
    d4 = wrcoef('d',c,l,'db4',4);
    v=[a4;d4;d3;d2;d1];
    [k, y] = PRINCOMP(v);
    tt1=[tt1 sum(sum(abs(y(:,2:4))))];
    is=data(2,i:i+127);
    [c,l] = wavedec(is,4,'db4');
    a4 = wrcoef('a',c,l,'db4',4);
    d1 = wrcoef('d',c,l,'db4',1);
    d2 = wrcoef('d',c,l,'db4',2);
    d3 = wrcoef('d',c,l,'db4',3);
    d4 = wrcoef('d',c,l,'db4',4);
    v=[a4;d4;d3;d2;d1];
    [k, y] = PRINCOMP(v);
    tt2=[tt2 sum(sum(abs(y(:,2:4))))];
end
O=[tt1; tt2]; c=zeros(112,1);
c(find(and(O(1,:)>3,O(2,:)<3)))=1;
c(find(and(O(1,:)<3,O(2,:)>3)))=2;
c(find(and(O(1,:)>3,O(2,:)>3)))=3;
d=[O ;c]';

```

EK-I : DVM Modelinden Elin Gerçek Zamanlı Denetimi için Matlab Programı

```

for dd=1:1000
ai2 = analoginput('nidaq','Dev1');
addchannel(ai2,[7 1 6 5]);
hs=1; dur = 0.1; %bir hareket süresi (sn)
for i=1:hs %hareket sayısı
    set(ai2,'InputType','SingleEnded');
    ai2.Channel.InputRange=[-10 10];
    ai2.SampleRate = 2000;
    set(ai2,'SampleRate', 2000);
    ActualRate=get(ai2,'SampleRate');
    set(ai2,'SamplesPerTrigger', ActualRate*dur);
    set(ai2, 'TriggerType', 'Manual');
    start(ai2); trigger(ai2); dat{i} = getdata(ai2); stop(ai2);
end
data=[];
for i=1:hs
    data=[data;dat{i}];
end;
d=data'; [sat,sut]=size(d); o=[];
% ----- Özellik çıkarma(WAVELET + PCA) -----
for j=1:sat
    is=d(j,i:i+199);
    [c,l] = wavedec(is,4,'db4');
    a4 = wrcoef('a',c,l,'db4',4);
    d1 = wrcoef('d',c,l,'db4',1);
    d2 = wrcoef('d',c,l,'db4',2);
    d3 = wrcoef('d',c,l,'db4',3);
    d4 = wrcoef('d',c,l,'db4',4);
    v=[a4;d4;d3;d2;d1];
    [k, y] = PRINCOMP(v);
    o=[o (sum(sum((y(:,2:4)).^2)))]

```

```

end
% -----
y = mvsvmclass(o', model); % model çıkışları
if y==2
    kontr=[1 0 0]
elseif y==3
    kontr=[0 0 0]
elseif y==4
    kontr=[0 0 1]
elseif y==5
    kontr=[1 0 1]
elseif y==6
    kontr=[1 1 0]
elseif y==7
    kontr=[0 1 0]
else
    kontr=[0 1 1]
end
diego=digitalio('parallel','LPT1');
dato=addline(diego,1:3,'out');
putvalue(dato,contr);
pause(1);
d=d+1;
end

```

EK-J: Dört Hareket için PIC16F628 PICBASIC Programı

TRISA=0

TRISB=%00001100 'portb.0 el / portb.1 bilek PWM çıkışı
 'portb.2 + portb.3 bilgi girişi

CMCON=7

Q1 VAR PORTB.0 'çıkış Q1 degiskene atandı EL

Q2 VAR PORTB.1 'çıkış Q2 degiskene atandı BİLEK

D1 VAR PORTB.2 'veri D1 degiskene atandı

D2 VAR PORTB.3 'veri D2 degiskene atandı

AA VAR BYTE

BB VAR BYTE

ACIK CON 1400

KAPALI CON 2000

SAG CON 2240

SOL CON 700

ORTA CON 1470

'BASLANGIC DEGERLERI-----

PORTA=0

PORTB=0

Q1=0

Q2=0

D1=0

D2=0

AA=10

BB=10

'ANA MENU-----

ANA_MENU:

 IF D1=0 & D2=1 THEN

 GOTO SAGA_DON

 ENDIF

 IF D1=1 & D2=0 THEN

 GOTO SOLA_DON

```

ENDIF
IF D1=1 & D2=1 THEN
    GOTO EL_AC
ENDIF
AA=10
BB=10
A:
AA=AA-1
    Q1=1
    PAUSEUS KAPALI
    Q1=0
    PAUSE 18
IF AA!=0 THEN A
B:
BB=BB-1
    Q2=1
    PAUSEUS ORTA
    Q2=0
    PAUSE 18
IF BB!=0 THEN B
    GOTO ANA_MENU
'SAGA DONUS-----
SAGA_DON:
BB=10
A2:
BB=BB-1
    Q2=1
    PAUSEUS SAG
    Q2=0
    PAUSE 18
IF BB!=0 THEN A2
GOTO ANA_MENU

```

'SOLA DONUS-----

SOLA_DON:

AA=10

BB=10

A3:

BB=BB-1

Q2=1

PAUSEUS SOL

Q2=0

PAUSE 18

IF BB!=0 THEN A3

GOTO ANA_MENU

'ELI KAPAT-----

EL_AC:

AA=10

A4:

AA=AA-1

Q1=1

PAUSEUS ACIK

Q1=0

PAUSE 18

IF AA!=0 THEN A4

GOTO ANA_MENU

EK-K: Yedi Hareket için PIC16F628 PICBASIC Programı

```

@ DEVICE pic16F628
@ DEVICE pic16F628, WDT_OFF
@ DEVICE pic16F628, PWRT_ON
@ DEVICE pic16F628, PROTECT_OFF
@ DEVICE pic16F628, MCLR_ON
@ DEVICE pic16F628, HS_OSC
TRISA=255
TRISB=%11100000
CMCON=7
M1 VAR BYTE
Y1 VAR BIT
Y2 VAR BIT
SAG CON 2000
SOL CON 1000
ORTA CON 1500
'BASLANGIC DEGERLERI-----
HIGH Y1
LOW Y2
ANA:
  IF PORTB.5=Y2 AND PORTB.6=Y2 AND PORTB.7=Y1 THEN
    FOR M1=1 TO 10
      HIGH PORTB.0
      PAUSEUS SAG
      LOW PORTB.0
      PAUSE 18
    NEXT M1
    GOTO ANA
  ENDIF
  IF PORTB.5=Y2 AND PORTB.6=Y1 AND PORTB.7=Y2 THEN
    FOR M1=1 TO 10
      HIGH PORTB.1

```

```

    PAUSEUS SAG
    LOW PORTB.1
    PAUSE 18
    NEXT M1
    GOTO ANA
ENDIF
IF PORTB.5=Y2 AND PORTB.6=Y1 AND PORTB.7=Y1 THEN
    FOR M1=1 TO 10
        HIGH PORTB.2
        PAUSEUS SAG
        LOW PORTB.2
        PAUSE 18
    NEXT M1
    GOTO ANA
ENDIF
IF PORTB.5=Y1 AND PORTB.6=Y2 AND PORTB.7=Y2 THEN
    FOR M1=1 TO 10
        HIGH PORTB.0
        HIGH PORTB.1
        HIGH PORTB.2
        PAUSEUS ORTA
        LOW PORTB.0
        LOW PORTB.1
        LOW PORTB.2
        PAUSE 18
    NEXT M1
    GOTO ANA
ENDIF
IF PORTB.5=Y1 AND PORTB.6=Y2 AND PORTB.7=Y1 THEN
    FOR M1=1 TO 10
        HIGH PORTB.0
        PAUSEUS SOL

```

```
        LOW PORTB.0
        PAUSE 18
    NEXT M1
    GOTO ANA
ENDIF
IF PORTB.5=Y1 AND PORTB.6=Y1 AND PORTB.7=Y2 THEN
    FOR M1=1 TO 10
        HIGH PORTB.1
        PAUSEUS SOL
        LOW PORTB.1
        PAUSE 18
    NEXT M1
    GOTO ANA
ENDIF
IF PORTB.5=Y1 AND PORTB.6=Y1 AND PORTB.7=Y1 THEN
    FOR M1=1 TO 10
        HIGH PORTB.2
        PAUSEUS SOL
        LOW PORTB.2
        PAUSE 18
    NEXT M1
    GOTO ANA
ENDIF
```

EK-L : Bir Harekete ait EMG Kanal Kaydı için Matlab Programı

```

ai2 = analoginput('nidaq','Dev1');
addchannel(ai2,[7 1 6 5]);
hs=1    %Hareket sayısı
dur = 2; %bir hareket süresi (sn)
for i=1:hs %hareket sayısı
    set(ai2,'InputType','SingleEnded');
    ai2.Channel.InputRange=[-10 10];
    ai2.SampleRate = 2000;
    set(ai2,'SampleRate', 2000);
    ActualRate=get(ai2,'SampleRate');
    set(ai2,'SamplesPerTrigger', ActualRate*dur);
    set(ai2, 'TriggerType', 'Manual');
    start(ai2);
    trigger(ai2);
    dat{i} = getdata(ai2);
end
data=[];
for i=1:hs
    data=[data;dat{i}];
end;

```

8. ÖZGEÇMİŞ

Adı Soyadı : Gökhan KAYHAN

Doğum Yeri : Vakfıkebir

Doğum Tarihi : 30.01.1978

Medeni Hali : Bekar

Bildiği Yabancı Diller: İngilizce

Eğitim Durumu

Lise : Samsun Namık Kemal Lisesi (1994)

Lisans : Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü (1999)

Yüksek Lisans : Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı (2004)

Çalıştığı Kurumlar ve Yıl : Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü (2001 -)

İletişim Bilgileri : Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü, Kurupelit, SAMSUN

Telefon : 0533 339 15 73

e-posta adresi : gkayhan@omu.edu.tr