

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ANALOG MODÜLASYONLARDA DALGACIK
DÖNÜŞÜM TEMELLİ BİR AKILLI SINIFLANDIRMA
SİSTEMİ**

Sultan ERDEM YAKUT

Tez Yöneticisi

Yrd.Doç.Dr. Engin AVCI

YÜKSEK LİSANS TEZİ

ELEKTRONİK- BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

ELAZIĞ, 2007

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ANALOG MODÜLASYONLARDA DALGACIK
DÖNÜŞÜM TEMELLİ BİR AKILLI SINIFLANDIRMA
SİSTEMİ**

Sultan ERDEM YAKUT

Tez Yöneticisi

Yrd.Doç.Dr. Engin AVCI

YÜKSEK LİSANS TEZİ

ELEKTRONİK – BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

ELAZIĞ, 2007

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ANALOG MODÜLASYONLARDA DALGACIK
DÖNÜŞÜM TEMELLİ BİR AKILLI SINIFLANDIRMA
SİSTEMİ**

Sultan ERDEM YAKUT

Yüksek Lisans Tezi

Elektronik-Bilgisayar Eğitimi Anabilim Dalı

Bu tez, ~~25.05.2022~~ tarihinde aşağıda belirtilen jüri tarafından oybirliği /oyçokluğu ile başarılı / ~~başarısız~~ olarak değerlendirilmiştir.

Danışman: Yrd. Doç. Dr. Engin ~~AVCI~~ *E. Avcı*

Üye: Prof. Dr. Muammer GÖKBULUT *Muammer*

Üye: Yrd. Doç. Dr. Anıl GÜLTEN *Anıl*

Üye: *[Signature]*

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

İÇİNDEKİLER

İÇİNDEKİLER.....	I
ŞEKİLLER LİSTESİ.....	III
TABLolar LİSTESİ.....	VI
SİMGELER LİSTESİ.....	VII
KISALTMALAR LİSTESİ.....	IX
ÖZET.....	X
ABSTRACT.....	XI
BÖLÜM 1: GİRİŞ.....	1
1.1. Analog Modülasyon Tanıma Alanında Yapılan Çalışmalar	1
1.2. Tezin Amacı	3
1.3. Tezin Organizasyonu	4
BÖLÜM 2: ANALOG MODÜLASYON ÇEŞİTLERİ.....	5
2.1. Genlik Modülasyonu.....	5
2.2. Frekans Modülasyonu	7
2.3. Faz Modülasyonu	9
BÖLÜM 3: DALGACIK DÖNÜŞÜMÜ KULLANILARAK ÖZELLİK ÇIKARMA.....	10
3.1. Örüntü Tanıma	10
3.2. Örüntü Tanıma Sisteminin Bileşenleri.....	13
3.2.1.Ön işlem	13
3.2.2. Özellik Çıkarma	14
3.3. Dalgacık Dönüşümü.....	15
3.3.1. Dalgacık Aileleri	18
3.3.2. Sürekli Dalgacık Dönüşümü	20
3.3.3. Ayrık Dalgacık Dönüşümü	21
3.4. Dalgacık Dönüşümünden Entropi Hesaplanması.....	21
BÖLÜM 4: SINIFLANDIRMA.....	23
4.1. Yapay Sinir Ağı Sınıflandırıcısı.....	23
4.1.1. Yapay Sinir Ağlarının Özellikleri	24

4.1.2. Yapay Sinir Ağlarının Yapısı.....	25
4.1.3. Yapay Sinir Ağları Aktivasyon Fonksiyonları.....	27
4.1.4. Yapay Sinir Ağlarının Eğitimi	29
4.1.5. Yapay Sinir Ağlarında Öğrenme Algoritmaları	30
4.2. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi	30
BÖLÜM 5: MATLAB GUI İLE ARAYÜZ OLUŞTURMA.....	34
5.1. GUI Nesneleri	35
5.2. GUI Nesnelerinin Özellikleri	36
5.3. Arayüz Tasarım Aşamaları	39
5.3.1. ANA.m Dosyasının Oluşturulması	40
5.3.2. WAV.m Dosyasının Oluşturulması	47
5.3.3. SECIM.m Dosyasının Oluşturulması.....	50
5.3.4. YSA.m Dosyasının Oluşturulması	51
5.3.5. UATBCS.m Dosyasının Oluşturulması	56
BÖLÜM 6: ANALOG MODÜLASYON UYGULAMALARI.....	59
6.1. Analog Modüleli İşaretlerin Oluşturulması.....	60
6.2. Özellik Çıkarma	63
6.3. Sınıflandırma.....	68
6.3.1. Yapay Sinir Ağ Sınıflandırıcısı.....	68
6.3.2. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi Sınıflandırıcısı	71
BÖLÜM 7: SONUÇ.....	75
7.1. Sonuçlar ve Tartışma.....	75
7.2. Öneriler	75
7.3. Yayınlar.....	76
BÖLÜM 8: KAYNAKLAR.....	77
ÖZGEÇMİŞ	

ŞEKİLLER LİSTESİ

Şekil 2.1. Genlik modülasyonun blok diyagramı.....	5
Şekil 2.2. Genlik modülasyon ve demodülasyon dalga şekilleri.....	6
Şekil 2.3. Genlik modüleli dalganın frekans spektrumu.....	7
Şekil 2.4. Frekans modülasyonunun blok diyagramı.....	8
Şekil 2.5. Frekans modülasyon ve demodülasyonu dalga şekilleri.....	9
Şekil 2.6. Faz modülasyonunun blok diyagramı.....	9
Şekil 3.1. Örüntü tanıma sistemi blok diyagramı.....	10
Şekil 3.2. Yapısal örüntü tanıma sistemi blok diyagramı.....	11
Şekil 3.3. Akıllı örüntü tanıma sistemi blok diyagramı.....	12
Şekil 3.4. Örüntü tanıma sisteminin bileşenleri.....	13
Şekil 3.5. Dalgacık dönüşümünün blok diyagramı.....	17
Şekil 3.6. Üç seviyeli dalgacık dönüşümünün yapısı.....	18
Şekil 3.7. a) Meksika Sapkası b) Meyer c) Morlet dalga örnekleri.....	18
Şekil 3.8. Daubechies filtresinin dalga çeşitleri.....	19
Şekil 3.9. Biorthogonal filtresinin dalga çeşitleri.....	19
Şekil 3.10. Symlets filtresinin dalga çeşitleri.....	20
Şekil 3.11. Coiflets filtresinin dalga çeşitleri.....	20
Şekil 4.1. Enbasit YSA hücresi.....	24
Şekil 4.2. Yapay sinir ağlarının yapısı.....	26
Şekil 4.3. YSA sınıflandırıcılarının hiyerarşisi.....	27
Şekil 4.4. Eşik aktivasyon fonksiyonu.....	28
Şekil 4.5. Doğrusal aktivasyon fonksiyonu.....	28
Şekil 4.6. Tanjant hiperbolik aktivasyon fonksiyonu.....	28
Şekil 4.7. İki girişli bir çıkışlı UATBÇS sisteminin temel yapısı.....	31
Şekil 4.8. İki girişli dokuz kurallı bir UATBÇS yapısı.....	32
Şekil 4.9. Gauss üyelik fonksiyonu.....	32
Şekil 5.1. Yeni bir GUI dosyası oluşturma menüsü.....	34
Şekil 5.2. Yeni bir GUI dosyası oluşturma penceresi.....	34
Şekil 5.3. Yeni bir GUI penceresi.....	35
Şekil 5.4. Nesnelere ait özellik penceresi.....	37
Şekil 5.5. Tasarım alanının ve nesnelerin kullanımı.....	39
Şekil 5.6. Ana menü arayüz penceresi.....	40
Şekil 5.7. Axes nesnesine isim verilmesi.....	41
Şekil 5.8. Push buton nesnesine isim verilmesi.....	41

Şekil 5.9. String nesnesine ait özellik penceresi.	41
Şekil 5.10. Popup menüye değer girilmesi.....	42
Şekil 5.11. Frame nesnesinin rengini değiştirme.	42
Şekil 5.12. Ana menü m-file dosyası.	43
Şekil 5.13. Modülasyon tipini seçme menüsü.....	43
Şekil 5.14. F_c ve F_s metin kutusu için uyarı mesajı.....	45
Şekil 5.15. Gürültü metin kutusu uyarı mesajı.....	46
Şekil 5.16. Ana menü kullanıcı arayüz penceresi.	47
Şekil 5.17. Özellik çıkarma arayüz penceresi.	48
Şekil 5.18. Filtre seçim menüsü.	48
Şekil 5.19. P değeri hatalı girildiğinde ekrana gelen uyarı mesajı.....	49
Şekil 5.20. Özellik çıkarma kullanıcı arayüz penceresi.	49
Şekil 5.21. Seçim penceresi.	51
Şekil 5.22. YSA arayüz penceresi.....	52
Şekil 5.23. Eğitimin başladığını gösteren mesaj penceresi.	52
Şekil 5.24. Yapay sinir ağları ile eğitimin performans eğrisi.....	53
Şekil 5.25. Eğitim sonucunda YSA penceresinin görünümü.	53
Şekil 5.26. Test başladığını gösteren mesaj penceresi.	54
Şekil 5.27. Test İşlemi sonunda YSA penceresinin görünümü.....	54
Şekil 5.28. UATBCS arayüz penceresi.	57
Şekil 5.29. Eğitim sonundaki UATBCS arayüz penceresi.....	58
Şekil 5.30. Test sonundaki UATBCS arayüz penceresi.....	58
Şekil 6.1. Modülasyon tanıma için kullanılan ses işareti.....	60
Şekil 6.2. $m=1$ ve $\phi = \pi/4$ İçin bir GM modüleli işaret örneği.....	61
Şekil 6.3. $\phi = \pi/3$ için bir ÇYB modüleli işaret örneği.	61
Şekil 6.4. $\phi = \pi/10$ için bir ÜYB modüleli işaret örneği.	62
Şekil 6.5. $\phi = \pi/10$ için bir AYB modüleli işaret örneği.	62
Şekil 6.6. $\phi = \pi/4$ frekans sapması = 9 için bir FM modüleli işaret örneği.	63
Şekil 6.7. $\phi = \pi/4$ frekans sapması = 4 için bir PM modüleli işaret örneği.	63
Şekil 6.8. YSA ve Dalgacık Entropi ile Analog Modülasyon Tanımanın Yapısı.	64
Şekil 6.9. 7 seviyeli dalgacık dönüşümünün yapısı.	64
Şekil 6.10. GM modüleli işaretin dalgacık dönüşümü katsayı işaretleri.....	65
Şekil 6.11. ÇYB modüleli işaretin dalgacık dönüşümü katsayı işaretleri.....	65
Şekil 6.12. ÜYB modüleli işaretin dalgacık dönüşümü katsayı işaretleri.....	66
Şekil 6.13. AYB modüleli işaretin dalgacık dönüşümü katsayı işaretleri.....	66

Şekil 6.14. FM modüleli işaretin dalgacık dönüşümü katsayı işaretleri.	67
Şekil 6.15. PM modüleli işaretin dalgacık dönüşümü katsayı işaretleri.	67
Şekil 6.16. YSA sınıflandırıcı için arzu edilen çıkış matrisi.	69
Şekil 6.17. Öğrenme ve hata oranlarının değiştirildiği pencere.	69
Şekil 6.18. YSA eğitimi sonucunda elde edilen çıkış matrisi.	69
Şekil 6.19. YSA test işlemi sonucunda elde edilen çıkış matrisi.	69
Şekil 6.20. YSA sınıflandırıcının eğitim performansı.	70
Şekil 6.21. YSA eğitimi sonucunda modüleli işaretlerin başarı oranları.	70
Şekil 6.22. YSA test işlemi sonucunda modüleli işaretlerin başarı oranları.	71
Şekil 6.23. Hata oranının değiştirildiği pencere.	71
Şekil 6.24. UATBÇS eğitim aşamasında elde edilen çıkış matrisi.	72
Şekil 6.25. UATBÇS test aşamasında elde edilen çıkış matrisi.	72
Şekil 6.26. UATBÇS eğitimi sonucunda modüleli işaretlerin başarı oranları.	73
Şekil 6.27. UATBÇS test sonucunda modüleli işaretlerin başarı oranları.	73

TABLÖLAR LİSTESİ

Tablo 2.1. Analog Modülasyon Çeşitleri.....	5
Tablo 6.1. YSA sınıflandırıcı için başarı oranları.....	74
Tablo 6.2. UATBÇS sınıflandırıcı için başarı oranları.	74

SİMGELER LİSTESİ

V_m	:	Bilgi işaretinin genliğini temsil eder.
f_m	:	Bilgi işaretinin frekansını temsil eder.
V_c	:	Taşıyıcı işaretinin genliğini temsil eder.
f_c	:	Taşıyıcı işaretinin frekansını temsil eder.
$\gamma_{\max}$	:	Anlık genliğin normalize edilmiş spektrum güç yoğunluğunun maksimum değeri
σ_{ap}	:	Sinyalin zayıf olmayan aralığında, anlık fazın doğrusal olmayan parçasının mutlak değerinin standart sapması
σ_{dp}	:	Sinyal parçalarının kuvvetli aralıklarında anlık fazın doğrusal olmayan bileşeninin standart sapması
P	:	Alınan radyo sinyalinin taşıyıcı frekansının etrafındaki sinyal tayf simetri ölçüsü
$A_{cn}(i)$	:	Normalize edilmiş anlık genlik değeri
f_s	:	Örnekleme oranını temsil eder
N_s	:	Her işaret parçasındaki örnek sayısı
$\phi_{NL}(i)$	:	Lineer olmayan bileşenin anlık faz değeri
$X_c(i)$	:	Kesilen işaretin spektrum büyüklüğünün sırası
$f_{cn} + 1$	:	Taşıyıcı frekansa uygun örnek sayısıdır.
BG	:	Bant genişliğini temsil eder
Δf	:	Frekans sapması
$f(x)$	:	Fonksiyonu temsil eder.
$\psi(t)$	:	Dalgacık dönüşümünün fonksiyonu
$D_j(t)$	:	Detay katsayıları
A_j	:	Yaklaşık katsayıları
b	:	Dönüşümü etmeni
a	:	Ölçekleme etmeni
E	:	Entropi değerleri
(s_i)	:	İşareti oluşturan örnek sayısı
W	:	Ağın giriş ağırlığı
V	:	1. katman çıkışı

n	:	Giriş sayısı
x	:	Ağın girişi
e	:	Hata değeri
α	:	Öğrenme katsayısıdır
y	:	Çıkış
$C(t)$	:	Analog modüleli işaret
θ_0	:	Başlangıç açısı
K_f	:	Frekans sapma katsayısı
dB	:	Desibel

KISALTMALAR LİSTESİ

RF	:	Radyo Frekansı
SD	:	Sürekli Dalga
GM	:	Genlik Modülasyonu
ÇYB	:	Çift Yan Bant Modülasyonu
TYB	:	Tek Yan Bant Modülasyonu
ÜYB	:	Üst Yan Bant Modülasyonu
AYB	:	Alt Yan Bant Modülasyonu
ARYB	:	Artık Yan Bant Modülasyonu
FM	:	Frekans Modülasyonu
PM	:	Faz Modülasyonu
İ/G	:	İşaret / Gürültü Oranı
OKH	:	Ortalama Karesel Hata
BCO	:	Bulanık C Ortalamalar
ADD	:	Ayrık Dalgacık Dönüşümü
YSA	:	Yapay Sinir Ağları
UATBÇS	:	Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi
GKA	:	MATLAB Grafiksel Kullanıcı Arayüzü

ÖZET

Yüksek Lisans Tezi

ANALOG MODÜLASYONLARDA DALGACIK DÖNÜŞÜM TEMELLİ BİR AKILLI SINIFLANDIRMA SİSTEMİ

Sultan ERDEM YAKUT

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektronik Bilgisayar Eğitimi Anabilim Dalı

2007, Sayfa : 81

Bu çalışmada dalgacık dönüşümü, Yapay Sinir Ağları (YSA), Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi (UATBÇS) yöntemleri kullanılarak analog modülasyon tanıma için akıllı bir sistem geliştirilmiştir. Bunun için MATLAB Grafiksel Kullanıcı Arayüzü (GUI) değiştirilebilir parametrelerle en iyi analog modülasyon tanımayı gerçekleştirme imkanı sağlar. Dalgacık dönüşümü yöntemi işaretlerin özelliklerini çıkarmada, YSA ve UATBÇS yöntemleri ise sınıflandırmada kullanılmıştır. YSA sınıflandırıcı kullanılarak %98 lik ve UATBÇS sınıflandırıcı kullanılarak %94 lük ortalama tanıma başarı oranları elde edilmiştir. Bu oranlar her iki yöntemin de analog modülasyon türlerini doğru tanımadaki etkinliğini göstermiştir.

Anahatar Kelimeler : Otomatik Analog modülasyon tanıma, Örüntü Tanıma, Dalgacık dönüşümü, Entropi, YSA, UATBÇS, Özellik Çıkarma, Sınıflandırma, MATLAB GUI.

SUMMARY

MASTER THESIS

AN INTELLIGENT CLASSIFICATION SYSTEM BASED ON WAVELET TRANSFORM FOR ANALOG MODULATIONS

Sultan ERDEM YAKUT

Fırat University

Graduate School of Natural and Applied Sciences

Department of Electronic and Computer Education

2007, Page: 81

In this study, an intelligent system developed for analog modulation recognition by using wavelet transform, artificial neural network (ANN) and adaptive network based fuzzy inference system (ANFIS). The MATLAB GUI provides performing of optimum analog modulation recognition with variable parameters. The wavelet transform methods was used for extracting of signal features, the ANN and ANFIS methods were used for classification. The %98 and %94 correct recognition rates were obtained by using ANN and ANFIS classification respectively. These show that both of these methods are effective in analog modulation recognition.

Key words: Automatic Analog Modulation Recognition, Pattern Recognition, Wavelet Transform, Entropy, ANN, ANFIS, Feature Extraction , Classification, MATLAB GUI.

1.GİRİŞ

Haberleşme, bilginin elektronik devrelerle iletimi, alınması ve işlenmesidir. Bir haberleşme sistemi verici, alıcı ve iletim ortamı olmak üzere üç temel kısımdan oluşur. Alçak frekanslı bilgi işaretini iletmek için yeryüzü ortamı elverişli değildir [1]. Bundan dolayı, radyo iletişiminde alçak frekanslı işaretler yüksek frekanslı işaretlere bindirilerek iletilir. Bu şekilde bilginin (ses, resim, manzara,...) taşıyıcı kullanılarak iletilmesine *modülasyon* denir.

Bilgi işareti, iki veya daha fazla nokta arasında iletilmek istenen işarettir. Bilgi işareti ses, resim ve müzik işareti olabilir. Bilgi işaretinin genliği V_m , frekansı f_m ile gösterilir. Taşıyıcı işareti, modülasyonu gerçekleştirmek için kullanılan ve bazı özellikleri bilgi işaretine bağlı olarak değiştirilen yüksek frekanslı işarettir. Taşıyıcı işarete modüle edilen, radyo frekans (RF) işareti de denir. Taşıyıcı işaretin genliği V_c , frekansı ise f_c ile gösterilir.

1.1. Analog Modülasyon Tanıma Alanında Yapılan Çalışmalar

Kaynağı bilinmeyen işaretlerin modülasyon tipini belirleme modülasyon tanıma çok önemli yer teşkil etmektedir. İlk modülasyon tanıma çalışmalarında, operatörlerin radyo frekans spektrumunu taramasına ve göstergeleri kontrol etmelerine bağlı olarak sınıflandırma yapılmıştır [2-7]. Bu yöntem, operatörlerin yetenek ve deneyimleriyle sınırlı kalmıştır. Bu sınırlamalar otomatik modülasyon tanıma sistemlerinin gelişmesine ön ayak olmuştur [3,4]. Modülasyon tanıma çalışmalarında operatörlere bağlı kalmadan tanımlama yapabilmek için 1980 lerden sonra geliştirilen modülasyon sınıflandırma algoritmaları, istatistiksel örnek tanıma ve karar teori yaklaşımı olmak üzere temelde iki gruba ayrılmıştır [2-4],[8-10].

Karar teorisi, sınıflandırıcılar varsayımlar ile modülasyon tanıma problemlerini formülleştirilen olasılık yaklaşımıdır. 1990 lardan sonra otomatik modülasyon tanıma çalışmalarında yapay sinir ağları sınıflandırıcı olarak kullanılmıştır [11,12].

Haberleşme işaretleri, bir konumdan başka bir konuma farklı frekanslarda ve farklı modülasyon tiplerinde iletilirken kayıplara ve bozulmalara uğrayabilirler. Sivil ve savunma alanında kullanılan haberleşme işaretlerinin izlenmesi, tanımlanması gerekmektedir [2]. Modülasyon tanıma alanındaki çalışmalar aşağıda sıralanmıştır.

Kaynak 13’de modüleli işaretin anlık frekans ve anlık genliğindeki değişim kullanılarak analog modülasyon işaretleri için modülasyon tanıma yöntemi önerilmiştir. Bu çalışmada, kullanılan anahtar özellikler, tepe noktasının ortalaması ve anlık frekansın mutlak değerinin ortalamasının oranıdır. Bu yöntem, sürekli dalga (SD), frekans modülasyonu (FM) ve çift yan

bant modülasyonu (ÇYB) analog modüleli işaretleri arasında ayırım yapmak için kullanılmıştır. 35dB İşaret/Gürültü (İ/G) oranı ile gerçekleştirilen genlik modülasyonu (GM) ve FM işaretleri arasındaki ayırım iki anahtar özellik ile belirlenmiştir. Ayrıca tek yan bant modülasyonu (TYB) 5dB İ/G oranında GM ve FM işaretlerinden ayırt edilebilmektedir.

Kaynak 14’de radyo işaretlerinin zarf karakteristiklerine dayanan modülasyon tanıma önerilmiştir. Bu yöntemde, Hilbert dönüşümü kullanılmıştır. Bu sınıflandırıcı GM, FM, ÇYB, TYB analog modüleli işaretlerin tanımlanması için kullanılmıştır. Bu çalışmada FM işaret için %100, GM işaret için %90.5, ÇYB işaret için %94 ve TYB işaret için %80 lik bir başarı oranı bulunmuştur.

Kaynak 15’de analog radyo işaretleri için modülasyon tanıma önerilmiştir. Bu yöntemde anahtar özellik olarak kaynak 14’deki R parametresi ve anlık frekans değişikliği kullanılmıştır. Bu yöntem ile GM, ÇYB, TYB, FM ve SD modülasyon tipleri tanımlanmıştır. Bu modülasyon tiplerinden TYB hariç İ/G oranı 15 dB’de ortalama %90 lık başarı oranı ile sınıflandırma yapılmıştır. TYB işaretlerinin ortalama başarı oranı ise %66 olarak bulunmuştur. Bu yüzden Nagy anlık frekansın ortalama değeri ile farklı anahtar özellikleri önermiştir. Bu özellikler ile TYB işaretinde 15dB İ/G’de ortalama başarı oranı %94 ve diğer modülasyon tipleri için ortalama başarı oranı %100 olarak bulunmuştur.

Kaynak 16’da modülasyon tanımda düşük modülasyon işareti GM ve saf taşıyıcılı SD işaretlerinin gürültü ortamından ayrışımı önerilmiştir. Anahtar özellik olarak, işaret zarfının faz bileşenlerinin, dik bileşenlerine oranını kullanmıştır.

Kaynak 17’de üst yan bant modülasyonu (ÜYB) ve alt yan bant modülasyonu (AYB) işaretleri arasında aynı anda modülasyon tanıma önerilmiştir. Bu yöntem, ÜYB işaretinin anlık frekans değerini esas alır. ÜYB işaret AYB işaretin tersidir ve pozitiften daha fazla negatif frekans darbeye sahiptir. Tanımlanan anahtar özellik G, anlık frekansın pozitif ve negatif darbe sayısıdır. ÜYB için $G > 1$, AYB için $G < 1$ olmalıdır.

Kaynak 18’de analog modüleli işaretler için modülasyon tanıma önerilmiştir. Bu yöntemde, karar teori yaklaşımından yararlanılmıştır. GM, ÇYB, TYB, ÜYB, AYB, FM, ARYB (artık yan bant) ve birleştirilmiş modüleli işaretler arasında sınıflandırma yapmak için dört anahtar özellik kullanmıştır.

Bu yöntemde kullanılan anahtar özellikler sırasıyla açıklanmıştır.

1. γ_{max} = Anlık genliğin normalize edilmiş spektrum güç yoğunluğunun maximum değeri.
2. σ_{ap} = İşaretin zayıf olmayan aralığında, anlık fazın doğrusal olmayan parçasının mutlak değerinin standart sapması.
3. σ_{dp} = İşaret parçalarının kuvvetli aralıklarında anlık fazın doğrusal olmayan bileşeninin standart sapması.
4. P = Alınan radyo işaretinin taşıyıcı frekansının etrafındaki işaret tayf simetri ölçümü

Kaynak 19’da ise analog modüleli işaretler için bir modülasyon tanımlayıcı önerilmiştir. Bu modülasyon tanımlayıcı, geleneksel işaret işleme için dört anahtar özellik kullanmıştır. Bu yöntemde, önceki çalışmalardan farklı olarak, her bir özellik için uygun eşik değeri seçilmiş ve eşik değerinin her değişiminde, eşik değeri otomatik olarak uyarlanmıştır. Modülasyon tanımlayıcı olarak yapay sinir ağları (YSA) kullanılmıştır.

Yapılan diğer bir çalışmada ise GM, FM, SD analog modüleli haberleşme işaretlerinin otomatik sınıflandırılması Matlab Grafik Kullanıcı Arayüzü (GKA) kullanılarak gerçekleştirilmiştir [20]. Çalışmada sınıflandırıcı olarak Bulanık C Ortalamalar (Fuzzy C-Means- FCM (BCO)) kullanılmıştır.

1.2. Tezin Amacı

Sivil ve savunma amaçlı birçok uygulamada işaretlerin tanınması ve denetlenmesi büyük önem taşımaktadır. Tam ve doğru sınıflandırma yapabilmek için işaretlerin hangi modülasyon tipi ile modüle edildiğini bilmek gerekmektedir. İlk modülasyon tanıma çalışmalarında, işaretlerden ölçülen parametreler, operatörler tarafından yorumlanmıştır. Bir sonraki aşamada ise her modülasyon tipi için ayrı bir demodülatör tanımlanmıştır. Çalışmalarda operatörlerin kişisel bilgi ve becerisine olan bağımlılık, uygun demodülatör seçiminin zorluğu ve karmaşıklığı gibi dezavantajlar modülasyon tanıma alanında akıllı sistemlerin gelişmesini sağlamıştır. Analog modülasyon tanımada, ilk akıllı sistem çalışması olarak yapay sinir ağları kullanılmıştır. İlk modülasyon tanıma çalışmaları ile akıllı sistemler kıyaslandığında işaret tanıma başarı oranlarında artış gözlenmiştir. Akıllı sistemlerin gelişimiyle bilgisayarda program yazımına da ihtiyaç duyulmuştur.

Bu tez çalışmasında temel amaç, otomatik modülasyon tanımada dalgacık dönüşümü yöntemi kullanarak özellik çıkarmak, yapay sinir ağları ve uyarlamalı ağ tabanlı bulanık çıkarım

sistemi yöntemlerini kullanarak da sınıflandırma yapmaktır. Ayrıca Matlab GKA ile kullanıcı ara yüzü oluşturmaktır.

1.3. Tezin Organizasyonu

Bu çalışmada sırasıyla 2. Bölüm’de Analog Modülasyon Çeşitlerine, 3. Bölüm’de Dalgacık Dönüşümü Kullanarak Özellik Çıkarmaya, 4. Bölüm’de Sınıflandırma Çeşitlerine, 5. Bölüm’de Matlab GUI ile Arayüz Oluşturmaya, 6. Bölüm’de Analog Modülasyonlarda Dalgacık Dönüşümü ve Akıllı Sınıflandırıcı Kullanılarak Gerçekleştirilen Sınıflandırma Uygulamalarına, 7. Bölüm’de ise bu tez çalışmasından elde edilen genel sonuçlara ve önerilere değinilmektedir.

2. ANALOG MODÜLASYON ÇEŞİTLERİ

Taşıyıcı işaret sinüsoidal ya da darbe işaret olabilir. Taşıyıcı işaretin bu özelliğine göre modülasyonlar analog ve sayısal olmak üzere iki çeşide ayrılır [21].

- Analog modülasyonu, sürekli dalga (Continuous Wave - CW) modülasyonu olarak adlandırılır. Bu modülasyon türünde taşıyıcı işaret sinüzoidal dalgadır. Analog modülasyon dalga çeşitleri, taşıyıcı dalganın genlik, frekans ve faz parametreleri değiştirilerek elde edilir. Tablo 2.1’de analog modülasyon çeşitleri gösterilmiştir.

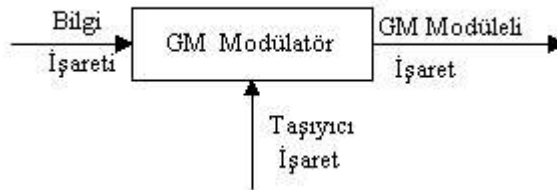
Tablo 2.1 Analog modülasyon çeşitleri.

Genlik Modülasyonu	(GM)
Frekans Modülasyonu	(FM)
Faz Modülasyonu	(PM)

2.1. Genlik Modülasyonu

Genlik modülasyonu (GM), bilgi işaretine bağlı olarak taşıyıcı işaretin genliğinin değiştirilmesidir. Şekil 2.1’de genlik modülasyonun en basit blok diyagramı gösterilmiştir. Genlik modülasyon tür olarak doğrusal (lineer) bir modülasyondur. Üç farklı çeşidi vardır. Bunlar;

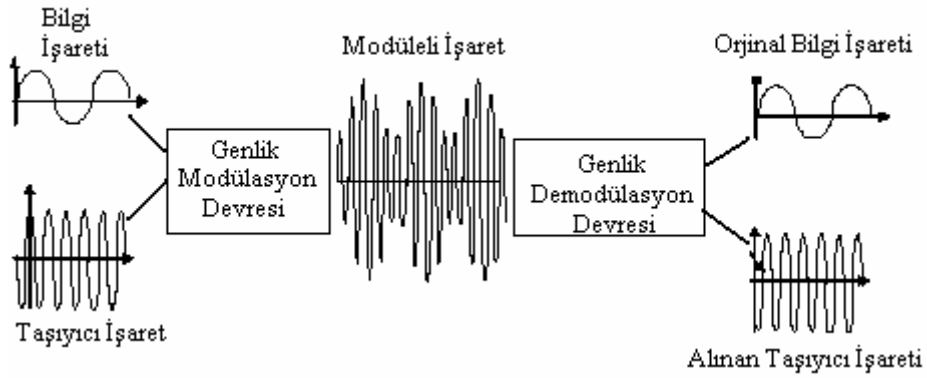
1. Taşıyıcılı Genlik Modülasyonu (TGM)
2. Çift Yan Bantlı Modülasyon (ÇYB)
3. Taşıyıcısı Bastırılmış Tek Yan Bantlı Modülasyon (TYB)
 - Üst Yan Bant Modülasyonu (ÜYB)
 - Alt Yan Bant Modülasyonu (AYB)



Şekil 2.1. Genlik modülasyonun blok diyagramı.

Genlik modülasyonu ses ve görüntü iletiminde diğer modülasyon türlerine göre daha ucuz ve basittir. Genlik modülasyonu dünyada uzun dalga, orta dalga vericilerinde kullanılmaktadır. Ayrıca televizyon yayıncılığında video işareti negatif genlik modülasyonuna tabi tutulmaktadır.

Bilgi işaretinin frekansı, taşıyıcı işaretin frekansından daha düşüktür. Genlik Modülasyonunun uygulama alanlarından biri de yüksek güçlü orta dalga ve uzun dalga radyo vericileridir. Son yıllarda üretilen orta ve uzun dalga vericilerinde uygulanan genlik modülasyonu ile vericinin verimi %85' in üzerine çıkarılabilmektedir. Genlik modülasyon ve demodülasyon işaret şekilleri Şekil 2.2'de gösterilmiştir.



Şekil 2.2. Genlik modülasyon ve demodülasyon dalga şekilleri.

$$C(t) = A(t) \cdot \cos(\omega_0 t + \phi) \quad (2.1)$$

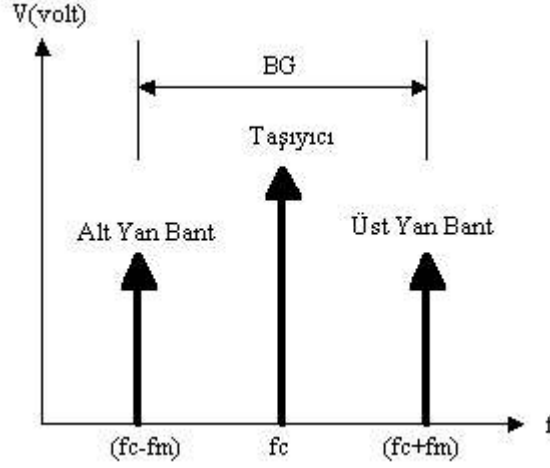
$$A(t) = [1 + m \cdot f(t)] \quad (2.2)$$

$$C(t) = A \cdot [1 + m \cdot f(t)] \cos(\omega_0 t) \quad (2.3)$$

Genliği modüle edilmiş dalga şeklinin genel ifadesi Denklem (2.1) ve Denklem (2.2) de, taşıyıcılı işaretin dalga biçimi de Denklem (2.3) de gösterilmiştir. $A(t)$ genliği ile modüle edilen $f(t)$ bilgi işareti arasındaki ilişkiye bağlı olarak genlik modülasyonunun değişik biçimleri elde edilir. Burada, $f(t)$ modüle eden işaret (bilgi işareti), ω_0 ise taşıyıcı frekansını belirtir. m ve A değişmez sabitler olup, m modülasyon derecesidir [22].

Genlik modülasyonu sonucunda iki yeni frekans oluşur. Taşıyıcı frekansının altında ve üstünde meydana gelen bu iki yeni frekansa yan bant (kenar bant) ismi verilir. Bu yan bantlardan $f_c + f_m$ frekansına *üst yan bant* denir ve ÜYB ile gösterilir. $f_c - f_m$ frekansına *alt yan bant* denir ve AYB ile gösterilir. Denklem (2.4) de AYB ve ÜYB işaretlerinin matematiksel

ifadesi gösterilmiştir. AYB, ÜYB ve taşıyıcı işaretlerinin frekans spektrumu Şekil 2.3’de gösterilmiştir.



Şekil 2.3. Genlik modüleli dalganın frekans spektrumu.

Modüleli işaretin frekans spektrumunda kapladığı alana *bant genişliği* (BG) denir. Bant genişliği Denklem (2.5) de gösterilmiştir [23].

$$C(t) = A \cos w_0 t + \frac{mA}{2} \cos(w_0 - w_m)t + \frac{mA}{2} \cos(w_0 + w_m)t \quad (2.4)$$

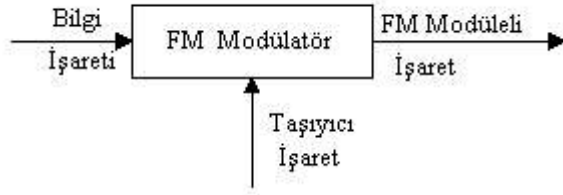
$$BG = f_{ÜYB} - f_{AYB} = 2.f_m \quad (2.5)$$

Modülasyon işlemi sonucunda, taşıyıcı işaretin genliğinde meydana gelen değişim miktarına *modülasyon indeksi* denir [23]. Modülasyon indeksi m ile gösterilir. Modülasyon indeksi, bilgi işaretinin genliğinin (E_m), taşıyıcı işaretin genliğine (E_c) oranıdır (Denklem 2.6).

$$m = \frac{E_m}{E_c} \quad (2.6)$$

2.2. Frekans Modülasyonu

Frekans modülasyonu, bilgi işaretinin genlik değişimlerine bağlı olarak taşıyıcı işaret frekansının değiştirilmesiyle elde edilir. Atmosferden ve çevredeki elektrik tesis ve sistemlerinden gelen gürültüler genlik üzerine bindiği için frekans modülasyonu ile kablolu taşıma ve radyo ile iletimde gürültüden etkilenme az olmaktadır [21]. Şekil 2.4’de frekans modülasyonunun blok diyagramı gösterilmektedir.



Şekil 2.4. Frekans modülasyonunun blok diyagramı.

Frekans modülasyonunda, bilgi işaretinin genliğine bağlı olarak taşıyıcı merkez frekansının altında ve üstünde frekans değişimleri oluşmaktadır. Taşıyıcı frekansının, modüle eden bilgi işaretinin pozitif ve negatif maksimum tepe değerlerine bağlı olarak frekans değişim miktarına frekans *sapması* denir. Frekans sapması Δf ile gösterilir. Frekans sapmasının formülü Denklem (2.7) de gösterilmiştir.

$$\Delta f = \frac{f_{\max} - f_{\min}}{2} \quad (2.7)$$

Frekans modülasyonunun sivil amaçlı vericilerin frekans sapması $\pm 75\text{kHz}$, askeri amaçlı vericilerin frekans sapması $\pm 40\text{kHz}$ dir. Bilgi işareti $f(t)$, taşıyıcının frekansı w_0 , modüle edilmiş FM işaretin dalga biçimi $C(t)$ ile gösterilmektedir.

$$w_0 = k.f(t) \quad (2.8)$$

$$C(t) = \cos\theta(t) \quad (2.9)$$

$$\theta(t) = w_0 t + \theta \quad (2.10)$$

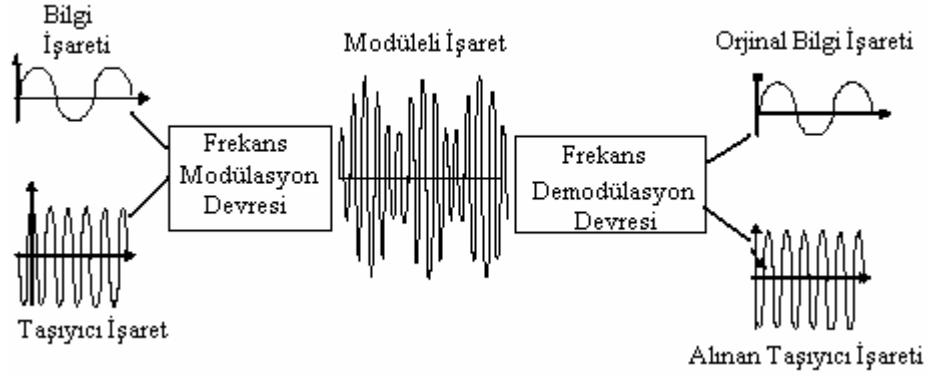
Frekans modülasyonunun dalga biçiminin matematiksel ifadesi aşağıdaki gibidir.

$$C(t) = A \cos[w_0 t + k_f \int_{t_0}^t f(\tau) d\tau] \quad (2.11)$$

Frekans modülasyonunun modülasyon indeksi, taşıyıcı dalganın frekans sapmasının, bilgi işaretinin frekansına oranıdır. Bu formül aşağıdaki gibidir.

$$m = \frac{\Delta f}{f_m} \quad (2.12)$$

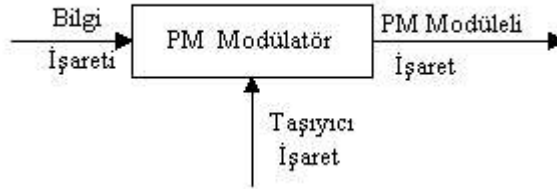
Şekil 2.5’de frekans modülasyon ve demodülasyon dalga şekilleri görülmektedir.



Şekil 2.5. Frekans modülasyon ve demodülasyonu dalga şekilleri.

2.3. Faz Modülasyonu

Bilgi işaretine bağlı olarak taşıyıcı işaretinin fazının değiştirilmesidir. Faz modülasyonlu işaret elde edilirken bilgi işaretinin türevi alınır. Şekil 2.6'da faz modülasyonunun blok diyagramı verilmiştir. Frekans modülasyonunda frekans sapması meydana gelirken, faz modülasyonunda faz sapması meydana gelmektedir [23].



Şekil 2.6. Faz modülasyonunun blok diyagramı.

3. DALGACIK DÖNÜŞÜMÜ KULLANILARAK ÖZELLİK ÇIKARMA

Otomatik modülasyon tanımda sınıflandırma yapabilmek için önce işarete ait özelliklerin çıkarılması gerekmektedir. Özellik çıkarabilmek için 1. bölümde bahsi geçen yöntemlerle parametre hesabı yapılır. Bu yöntemlerin dışında dalgacık dönüşümü kullanılarak da işarete ait özellikler çıkarılmaktadır.

3.1. Örüntü Tanıma

Örüntü, ilgilenilen varlıklar ile ilgili gözlenebilir veya ölçülebilir bilgilere verilen isimdir [24]. Örüntü tanıma, insanların çeşitli ses, görüntü ve benzeri tüm örüntülerin biçimsel şekillerinden çıkardıkları dilsel şekillendirmedir. Aslında, örüntü tanıma bilimin, mühendisliğin ve günlük hayatın geniş bir alanındaki etkinlikleri kapsamaktadır. Örneğin, hava değişiminin algılanması, bitki ve hayvan tanımlama, kitap okuma, yüz ve ses tanıma gibi birçok etkinlikte örüntü tanıma kullanılmaktadır. İnsan örüntü tanınması, geçmiş tecrübelerle dayalı öğrenmeden oluşur. Böylece, insanlar karşılaştıkları örüntü tanıma olaylarını tecrübeleri ile değerlendirme yeteneğine sahip olmaktadır. Belirli bir sesi tanımak için kullanılan kuralları tanımlamak mümkün değildir. İnsanlar bu işlemlerin birçoğunu oldukça iyi yapmalarına rağmen, bu işlemlerin hızlı, ekonomik, kaliteli ve otomatik olarak yapılabilmesi için bilgisayarları kullanırlar.

Örüntü tanıma, aralarında ortak özellik bulunan ve aralarında bir ilişki kurulabilen karmaşık işaret örneklerini veya nesneleri bazı tespit edilmiş özellikler veya karakterler vasıtası ile tanımlama veya sınıflandırmadır [25]. Örüntü tanıma bir çok mühendislik, tıp, askeri ve bilim alanlarında ses tanıma, radar hedef sınıflama, biyomedikal kontrol, veri madenciliği gibi uygulamalarda kullanılmaktadır [26]. Örüntü tanıma uygulamalarına, makine öğrenmesi, örüntü sınıflandırma, ayırım analizi ve nitelik tahmini gibi isimler de verilmektedir. Örüntü tanıma, Şekil 3.1’de gösterildiği gibi üç bölümden oluşmaktadır.



Şekil 3.1. Örüntü tanıma sistemi blok diyagramı.

İşaret / Görüntü İşleme, işaret veya görüntünün filtre edildiği, çeşitli dönüşüm ve gösterim teknikleri ile işlendiği, bileşenlerine ayrıldığı veya modellendiği bölümdür.

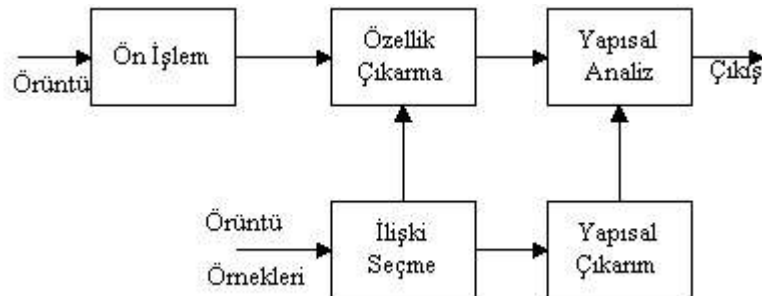
Özellik Çıkarma, işaret ve görüntünün tanımlayıcı anahtar özelliklerinin tespit edildiği ve normalizasyonun yapıldığı bölümdür. Sistemin başarımında en etkili rolü oynayan en önemli aşamadır [27].

Sınıflandırma, çıkarılan özellik örüntü özelliklerine bağlı olarak gruplandırmanın yapıldığı bölümdür.

Örüntü tanıma sistemleri, istatistiksel örüntü tanıma, yapısal örüntü tanıma ve akıllı örüntü tanıma olmak üzere üç gruba ayrılmaktadır [28].

1. İstatistiksel örüntü tanıma : İstatistiksel örüntü tanıma yönteminde, sınıflama algoritmaları istatistiksel analiz temeli üzerine kurulmuştur. Aynı sınıfa ait örüntüler, istatistiksel olarak tanımlanan benzer karakteristiklere sahiptirler. Bu yöntemde, özellik olarak nitelendirilen karakteristik ölçümler giriş örüntü örneklerinden çıkarılır. Her örüntü bir özellik vektörü ile tanımlanır. Sınıflandırıcı tasarımı, ölçümler ve olasılıklar gibi işlenebilir örüntü bilgilerini birleştirmeyi esas alır. Böylece sınıflama, giriş veri uzayının olasılık yoğunluk fonksiyonlarının tahmini üzerine kurulu bir istatistiksel yapıdır. İstatistiksel örüntü tanıma Bayes Karar Teorisi üzerine kurulmuştur [29].

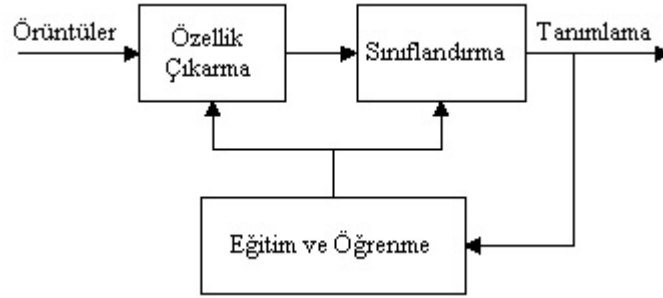
2. Yapısal örüntü tanıma : Yapısal örüntü tanıma yaklaşımında, verilen bir örüntü, şekilsel yapıdan temel karakteristik tanımlanmaya indirgenir. Çoğunlukla, örüntülerden çıkarılan bilgi, yalnızca özellikler kümesinin verilerinden oluşmaz. Özelliklerin birbirine bağlanması veya aralarındaki karşılıklı ilişki, tanımlamayı ve sınıflandırmayı kolaylaştıran örüntünün köşe sayısı, kenar açıları gibi önemli yapısal bilgilere sahiptir [30]. Yapısal yöntemde her örüntü, bileşenlerinin bir kompozisyonu olarak ele alınır. Şekil 3.2’de bir yapısal örüntü tanıma sistemi görülmektedir.



Şekil 3.2. Yapısal örüntü tanıma sistemi blok diyagramı.

Yapısal örüntü tanıma yönteminde çeşitli birimler arasındaki ilişki çok büyük önem taşır. Örneğin, ekrandaki bir masayı tanıma, “köşelerinden eşit uzunlukta bacaklar tarafından desteklenen yatay bir dikdörtgen yüzey” gibi yapısal tanımlamayı temel alarak gerçekleştirilebilir [24]. Bu yöntemde, çevre uzunluğu, alan, ağırlık merkezi, eylemsizlik momenti ve Fourier tanımlayıcıları gibi genel özellikleri kullanır. Otoresif model, poligonlar yaklaşımı ve zincir kodları yapısal örüntü tanıma yöntemine örnek olarak verilebilir [31].

3. Akıllı örüntü tanıma : Örüntü tanıma sistemi, daha önceden öğrendiklerini tutabilecek bir hafızaya sahip, çıkarım, genelleme ve belirli bir hata toleransı ile karar verebilme yeteneklerini içermekte ise bu sistem akıllı örüntü tanıma sistemi olarak değerlendirilir [31]. Şekil 3.3’de akıllı ve öğrenebilen örüntü tanıma yaklaşımının blok diyagramı verilmiştir.



Şekil 3.3. Akıllı örüntü tanıma sistemi blok diyagramı.

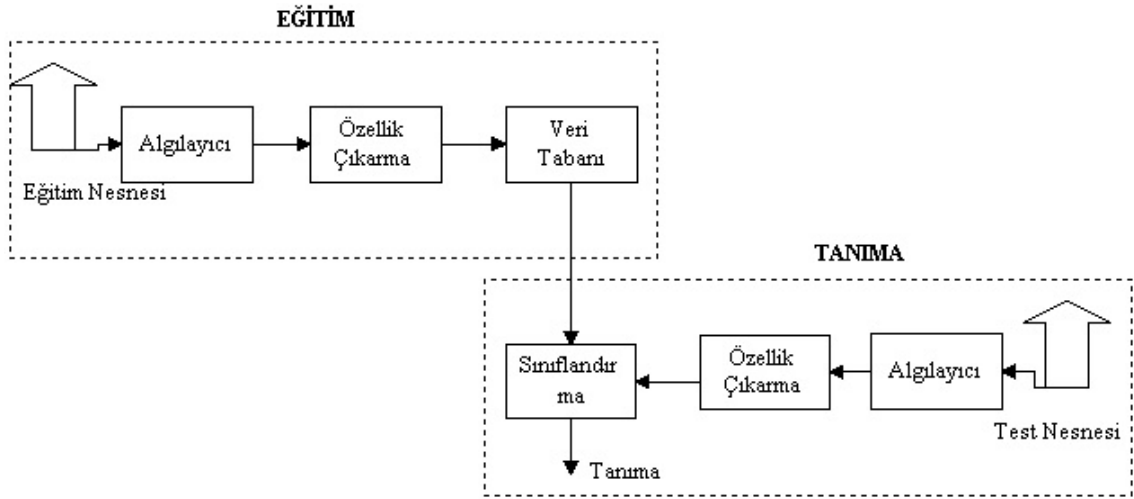
Akıllı örüntü tanıma yaklaşımları, öğrenme tabanlı olup, karar aşamasında geçmiş tecrübelerinden sonuç üretmektedirler. Günümüzde, öğrenmeli örüntü tanıma algoritmaları yapay sinir ağı merkezli olarak gelişmektedir. YSA yaklaşımları istatistik yaklaşıma karşı belirleyici olarak ifade edilebilir. Çünkü öğrenme algoritmaları örüntü sınıflarının istatistiksel özellikleri hakkında hiçbir şey kullanmamaktadır. Bununla birlikte, istatistiksel ve YSA örüntü tanıma yaklaşımları şekil ve amaç olarak çok benzer olup, hatta YSA 'nın geleneksel istatistiksel örüntü tanımanın bir uzantısı olarak ifade edilen görüşlerde bulunmaktadır [32].

İstatistiksel, yapısal ve yapay sinir ağları ile örüntü tanıma yaklaşımları arasında kesin bir ayrım yoktur. Verilen belirli bir örüntü tanıma probleminin çözümünde istatistiksel yaklaşıma göre örüntünün yapısı anlamsız olabilir. Yapı ancak uygun özellik seçimiyle yansıtılabilir. İstatistiksel örüntü tanıma; yapısal bilginin ifade edilmesinde görülen zorluk, yapısal örüntü tanıma kendini yapısal kuralların öğrenilmesinde gösterir. Buna karşın yapay sinir ağı yaklaşımı, istatistiksel ve yapısal yaklaşımlardan türetilmiştir. Açık bir şekilde örüntü

hakkındaki yapısal bilgi değerli olduğunda yapısal örüntü tanıma yaklaşımını seçmek daha doğrudur. Yapısal bilgi değersiz ve maksada uygun değilse istatistiksel yöntemi seçmek daha doğrudur. Yapay sinir ağları, istatistiksel ve yapısal yaklaşıma alternatif teknikler sağlayan ve örüntü tanımaya öğrenme boyutu katarak akıllı tanıma niteliği kazandıran bir teknik olarak düşünülebilir [28].

3.2. Örüntü Tanıma Sisteminin Bileşenleri

Örüntü tanıma sistemi, eğitim ve tanıma olmak üzere iki evreden oluşur. Sistemin en önemli elemanları; özellik çıkarma, veri tabanı oluşturma ve sınıflandırmadır. Eğitim ve tanıma evrelerindeki algılayıcılar ve özellik çıkarma elemanları değişebilir [24].



Şekil 3.4. Örüntü tanıma sisteminin bileşenleri.

3.2.1.Ön işlem

Algılayıcılar ile, bilgisayara sayısal olarak alınmış olan örüntü basitleştirilmek için bir dizi ön işlem sürecinden geçirilir. Örneğin alınan örüntü bir nesne görüntüsü ise; sırasıyla, görüntü eşikleme, kenar çıkarma gibi aşamalardan geçirilir.

Eşikleme; Sayısal bir görüntünün eşikleme işlemine tutulmasındaki amaç, nesne görüntüsünün özelliklerini belirlemede kolaylık sağlamaktır [29]. Eşikleme işlemiyle, görüntü iki renkle ifade edilebilir biçime getirilir. Görüntüyü eşikleme işlemine tabi tutmadan önce bir eşik değeri saptanır. Eşik değerinden daha yüksek gri seviye değerine sahip olan piksellere “1” değeri, daha küçük değerlere sahip olan piksellere ise “0” değeri ataması yapılarak görüntü daha basit bir biçim olan siyah-beyaz hale getirilir.

Kenar çıkarma; Kenar çıkarmadaki amaç, görüntünün içerdiği bilgiyi değerlendirip, gereksiz ve tanıma işlemlerinde zaman kaybettiren bilgiyi eleyerek yeterli düzeye indirgemektir. Kenar çıkarma, görüntü işleminin en temel konularından biridir [31]. Görüntü analizinde ve sınıflandırılmasında çok değerli bilgiler taşıdığından, ilgi çeken bir araştırma alanı olmuştur. Kenar çıkarma, robotik nesne örüntüsü tanımda çok önemli rol oynar. İnsan görme sistemi, nesneleri tanıma süreci içinde nesnenin dış çizgilerini izleyerek bir göz gezdirilir. Çoğu görüntüler somut nesneleri içermez ve bu görüntüleri anlamak onların yapısal özelliklerine bağlıdır. Yapısal özelliklerin çıkarımı ise kenar çıkarma ile ilgilidir. Bir kenar çıkarma yönteminin başarısını değerlendirmede kenar noktasının tanımı çok önemli yer tutar. Görüntüdeki bir nesnenin kenarı, nesnenin yüzey yoğunluğundaki değişimi ile ilgilidir [33].

3.2.2. Özellik Çıkarma

Özellik çıkarma örüntü tanımanın en önemli kısmını oluşturur. Özellik çıkarımının yapılmasındaki en önemli sebepler aşağıda sıralanmıştır [24].

1. Sınıflandırıcının küçük hatalar ile eğitimi ve karar aşamasının kısa sürede gerçekleşmesi için gereklidir.
2. Sabit olmayan zaman serilerinde karakteristikleri bulabilmek için özellik çıkarımı şarttır. Böylece karar aşamasının güvenilirliği artacaktır.
3. Sistem içi veya dışındaki kontrolsüz girişimlerin zarar vermemesi için bir özellik çıkarımı kararlı bir yapının oluşmasında etken olacaktır. Bu tür kararlı özellikler, sınıflandırıcının genelleme ve ayrışım yeteneğinin yüksek olmasında önemlidir.

Örüntü özelliklerini belirlemede ana problem verilen esas örüntüden en iyi özellikleri seçmektir [25]. İşaretler örüntülerinin özellikleri belirlendiğinde, özellik çıkarımı için genelde zaman ve frekans bölgesi gösterimi ön plandadır. Böylece karmaşık örüntü yapısının hem geçici ve hem de zamana bağlı olarak frekans değişimlerini içeren tanımlayıcı özellik bilgileri çıkarılabilir. Bu özellikler, işaretin zaman ve frekans bölgelerindeki yerel bilgilerini karakterize eder.

Nesne örüntülerinin özellik çıkarımı yapıldığında, özellikler kullanılan veritabanı ve uygulama alanına göre farklılık gösterir. Temel özellikler kenar, köşe, doğru ve eğri çizgiler, delik ve sınır eğriselliğidir. Nesne örüntüsü tanımlamaları bu özelliklerin birinden veya birkaçının birleştirilmesinden elde edilir.

Endüstriyel uygulamalarda çoğu zaman nesne örüntüsü sınırları ve bu sınırlardan türetilmiş ölçümler, özellik olarak kullanılır. Bu özellikler genel, yerel ve ilişkisel olmak üzere üç gruba ayrılabilir. Genel özelliklere örnek olarak çevre, ağırlık merkezi, sınır noktalarının ağırlık merkezine uzaklığı, eğrisellik, alan, eylemsizlik momenti gibi özellikler verilebilir. Doğru parçaları, sabit eğriselliği olan çember parçaları yerel özellikler için örnek verilebilir. İlişkisel özelliklere örnek olarak, nesne örüntüsünün alt parçalarının birbirlerine göre uzaklıkları, açıları gibi parametreler verilebilir. Özellikler çıkarılırken aşağıdaki yöntemler kullanılır.

Otoregresif model; Bu modelin esası, nesnenin sınırının bir boyutlu simgelenmesinde merkezsiz uzaklığın kullanımına dayanır. Algoritmanın ilk aşamasında merkez noktası hesaplanır. Daha sonraki aşamada nesnenin kenarını oluşturan her bir noktanın merkeze göre öklit uzaklığı bulunur [24].

Poligonal yaklaşım; Bu modelde, nesneye ait kritik noktaların hesabı yapılmaktadır. Nesnenin en sol üst sınır pikseli ile en sağ alt sınır pikseli başlangıç kritik noktaları olarak alınır. Bu iki nokta arasında düz bir çizgi çizilir. Daha sonra bu düz çizgiye dik doğrular saptanır. Bu doğruların kenar piksellerini kestiği noktalar belirlenir. Düz çizgi ile bu noktalar arasındaki dik doğru parçalarının uzunluğu hesaplanır. Belirli bir eşik değerinin üstündeki noktalar kritik noktalar olarak tespit edilir. Bu noktaların tek boyutlu simgelenmesi ile elde edilen özelliklerden öznitelik vektörü oluşturulur [24].

Zincir kodları; Sayısal görüntü, kenar çıkarma işleminden geçirildikten sonra, nesnenin sadece dış çizgileri kalır. Bundan sonraki adımda nesnenin kenar bilgilerinin kodlanması gerekir. Kodlama işlemi için zincir kodları kullanılabilir dördü ve sekizli olmak üzere İki tip zincir kodlama türü bulunmaktadır. Sekizli zincir kodunun dördü zincir koduna göre en önemli üstünlüğü yatay ve dikey kenar çizgilerinin yanı sıra çapraz kenar çizgilerini de kodlama imkanı tanımasıdır. Zincir kodları ile nesnenin kenarlarının hangi yönde ne kadar devam ettiği tespit edildikten sonra elde edilen bu öznitelik vektörüne, çeşitli işlemler uygulanarak daha küçük boyutta ve tüm nesneler için daha genel bir vektör oluşturulabilir [33].

3.3. Dalgacık Dönüşümü

Birçok işarete düşük frekans bileşenleri işaretin en önemli kısmını oluşturur. Yüksek frekans bileşenleri ise küçük ayrıntıları içerir. Örneğin bir ses işaretindeki yüksek frekans bileşenlerini kaldırırsak vurgu yok olur ve yalın bir ses kalır. Bununla birlikte düşük ses bileşenlerini kaldırırsak anlamsız sesler duyulur [34].

Dalgacık dönüşümü yeni bir yöntem olmasına rağmen 19. yüzyılda Joseph Fourier' in çalışmaları bu dönüşümün temelini oluşturmaktadır. Fourier, periyodu 2π olan herhangi bir $f(x)$ fonksiyonunun, fourier serilerinin toplamı ile temsil edilebileceğini iddia etmiştir [34]. Fourier serilerinin matematiksel ifadesi aşağıda verilmiştir.

$$a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx) \quad (3.1)$$

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x) dx \quad (3.2)$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(kx) dx \quad (3.3)$$

$$b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin(kx) dx \quad (3.4)$$

Fourier'in buluşundan sonra araştırmacıların dikkati frekans tabanlı analizden skala tabanlı analize doğru kaymıştır. Dalgacık adından ilk bahseden 1909 yılında Alfred Haar'dır. Dalgacığın teorik olarak yapısını ilk olarak Jean Morlet ile Fransa'daki Marseille Teorik Fizik merkezinde çalışan Alex Grossmann sunmuştur [34].

Dalgacık dönüşümünün yapısı çoğunlukla Y. Meyer tarafından geliştirilmiştir. 1988 de Stephane Mallat'ın çalışmalarıyla esas yapısına kavuşmuştur. Bu tarihten sonra araştırma uluslar arası bir yapıya kavuşmuş ve araştırmalar özellikle Amerika'da Daubechies, Ronald Coifman, ve Victor Wickerhauser bilim adamlarının öncülüğünde devam etmiştir [34].

Dalgacık dönüşümü, işaret işleme, özellik çıkarma, ses tanıma, resim sıkıştırma , bilgisayar görme gibi bir çok alanda kullanılır [35]. Dalgacık dönüşümü arızalı noktalar, daha yüksek süreksizlik türevlerinde ve öz-benlik gibi öteki işaret tekniklerinden verinin görünümünü ortaya çıkartmada daha yeteneklidir. Dalgacık dönüşümü sık sık fark edilebilir ayrışım olmadan işarete gürültüyü bastırabilir. Dalgacık dönüşümünün fonksiyonu Ψ ile gösterilir ve matematiksel ifadesi aşağıda gösterilmiştir.

$$C(a,b) = \int_R s(t) \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{b}\right) dt \quad (3.5)$$

$a = 2^j, b = 2^j, (j, k) \in Z^2$ dir. Burada s ayrık dalgacık dönüşüm işareti, $C(a,b)$ a ve b iki değişkenine bağlı katsayılarıdır. Dalgacık dönüşümü s işareti ile başlar, katsayılar ile sonuçlanır.

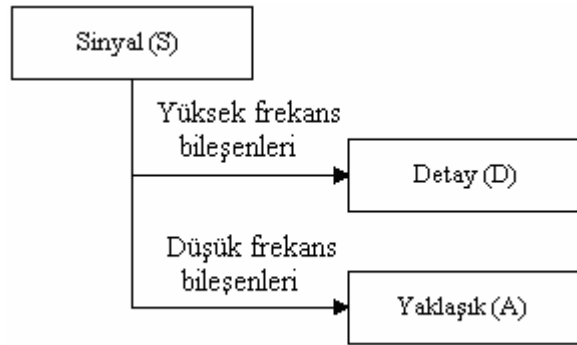
$$s(t) = \sum_{j \in Z} \sum_{k \in Z} C(j, k) \Psi_{j,k}(t) \quad (3.6)$$

$$D_j(t) = \sum_{k \in Z} C(j, k) \Psi_{j,k}(t) \quad (3.7)$$

$$A_j = \sum_{j > J} D_j \quad A_{j-1} = A_j + D_j \quad (3.8)$$

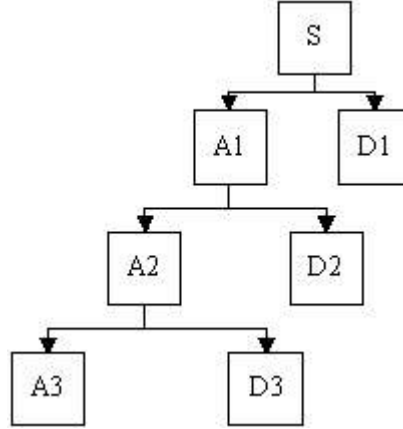
Burada $s(t)$ dalgacık dönüşümünün analizi, $D_j(t)$ detay(D) katsayıları, A_j yaklaşık (A) katsayılarıdır.

Dalgacık dönüşümüyle yaklaşık ve detay olmak üzere iki işaret elde edilir. Yaklaşık A ile detay D ile gösterilir. Yaklaşık (A) , işaretin yüksek skala – düşük frekans bileşenlerinden oluşur. Detaylar (D) ise işaretin düşük skala – yüksek frekans bileşenleridir. Dalgacık dönüşümünün en basit blok diyagramı Şekil 3.5’de gösterilmiştir.



Şekil 3.5. Dalgacık dönüşümünün blok diyagramı.

Dalgacık dönüşümüyle istenilen seviyede ayrışım yapılır [36]. Çok seviyeli ayrışımında elde edilen detay (D) ve yaklaşık (A) katsayılarından yaklaşık katsayısı ile ayrışımaya devam edilmekte ve bunun sonucunda yeniden A , D işaretleri elde edilmektedir. Ayrışım n seviyesine kadar yapılabilmekte ve buna “Dalgacık ayrıştırma ağacı” adı verilir. 3 seviyeli dalgacık dönüşümü Şekil 3.6’da gösterilmiştir. Örneğin $n,3$ yani üç aşamalı bir dalgacık dönüşümü sonucunda 3 adet A ve 3 adet D işareti olmak üzere toplam 6 adet işaret elde edilir. Dalgacık dönüşümü kullanılarak yapılan ayrıştırmada en küçük detaylar ortaya çıkarılır [37].

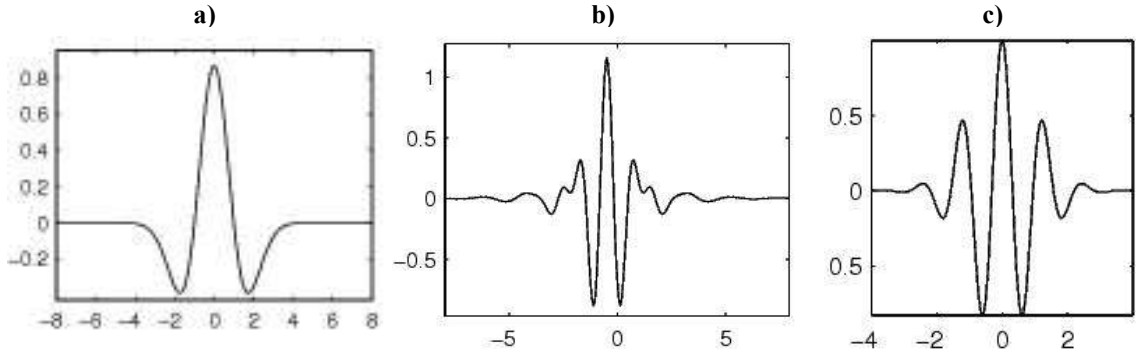


Şekil 3.6. Üç seviyeli dalgacık dönüşümünün yapısı.

Dalgacık dönüşümü yapılmadan önce filtreye karar verilmelidir. Filtre seçimi çok önemlidir. Filtreler, dalgacık aileleri olarak isimlendirilir.

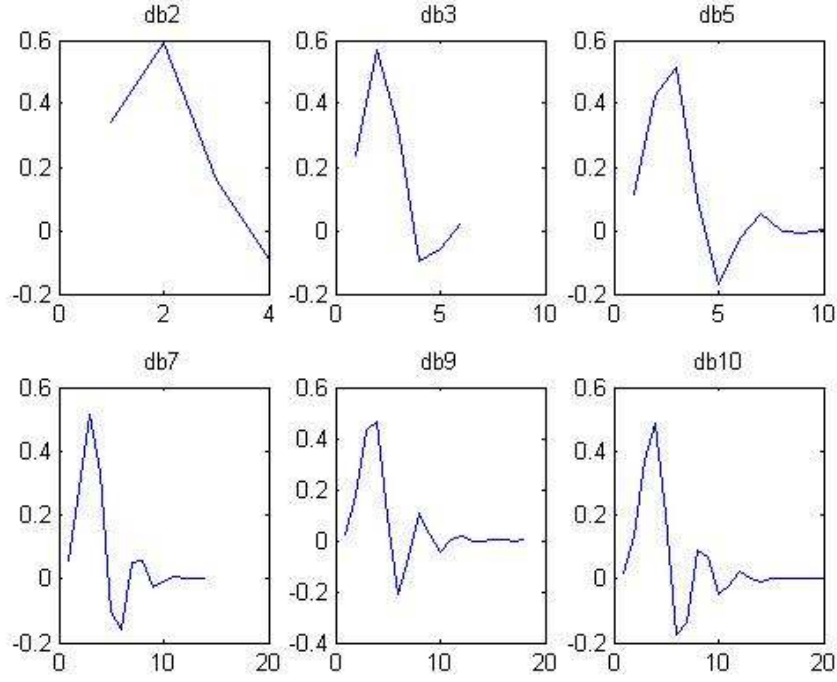
3.3.1. Dalgacık Aileleri

Dalgacık aileleri olarak Haar, Daubechies, Biorthogonal, Coiflets, Symlets dalgacık fonksiyonları ayrık dalgacık dönüşümünde kullanılır. Morlet, Meksika şapkası ve Meyer dalgacıkları ise skala fonksiyonunda kullanılır. Şekil 3.7’de bu dalgacık örnekleri verilmiştir.



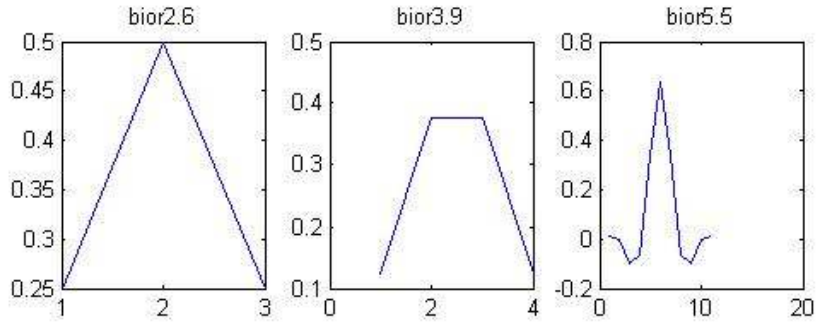
Şekil 3.7. a) Meksika Sapkası b) Meyer c) Morlet örnek dalgacık aileleri.

Daubechies dbN sembolü ile gösterilir. Burada N, 1 ile 10 arasında değişebilen sabit bir sayıdır. db1 dalgası Haar olarak isimlendirilir. Şekil 3.8’de Daubechies dalgacık örnekleri verilmiştir.



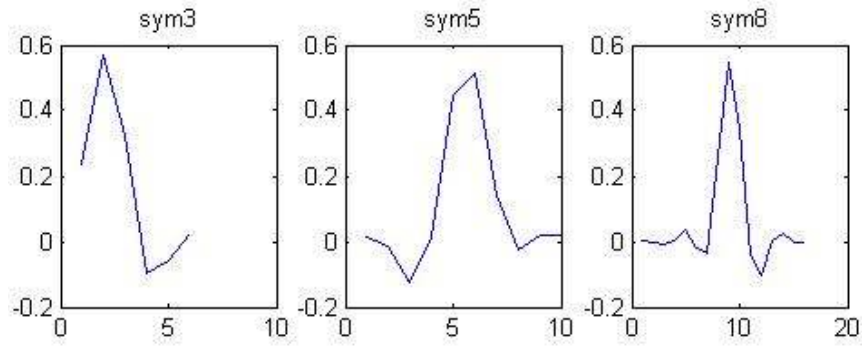
Şekil 3.8. Daubechies filtresinin dalga çeşitleri.

Biorthogonal ailesi işaret ve yeniden resim oluşturmaya gereksinim olan doğrusal fazın özelliklerini gösterir. Biorthogonal fonksiyonu biorNr.Nd olarak gösterilir. Bu fonksiyonun bior 1.3 , bior2.2 , bior 2.6 , bior3.1 , bior3.5 , bior3.9 ve bior 5.5 dalgacıkları ayırık dalgacık dönüşümünde (ADD) kullanılır. Şekil 3.9’da daubechies dalgacık örnekleri verilmiştir.



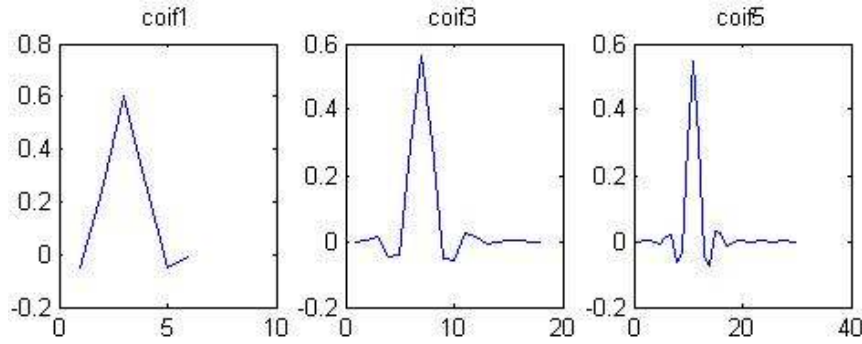
Şekil 3.9. Biorthogonal filtresinin dalga çeşitleri.

Symlets db ailesinin değişikliği gibi Daubechies tarafından önerilen simetrik dalgacıktır. İki dalgacık ailesinin özellikleri benzerdir. Daubechies ailesinin simetriği şeklindedir. SymN şeklinde gösterilir. N 1 ile 8 arasında değişir. Şekil 3.10’da fonksiyon örnekleri gösterilmiştir.



Şekil 3.10. Symlets filtresinin dalga çeşitleri.

Coiflets dalgacık fonksiyonu 0'a eşit $2N$ momentumuna sahip ve ölçekleme fonksiyonu 0'a eşit $2N-1$ momentumuna sahiptir. İki fonksiyon $6N-1$ uzunluğuyla desteklenir. Coiflet fonksiyonunu coifN olarak gösterilir ve N 1 ile 5 arasında değişebilir. Şekil 3.11'de fonksiyon örnekleri gösterilmiştir.



Şekil 3.11. Coiflets filtresinin dalga çeşitleri.

Dalgacık dönüşümünün en büyük avantajı local analiz performansında ortaya çıkar [38]. Analiz, büyük işaretleri küçük bir alana sınırlamaktadır. Sinüsoydal işaretlerde ise küçük ve durgun bir parçasında hesaplama yapılır. Dalgacık dönüşümü, kendi benzerliği, duran yüksek türev, bozuk noktalar, görünüş benzeri eğilimler, işaret analiz tekniklerinin kaçırdığı detayları ortaya çıkarmada yeteneklidir. Dalgacık dönüşümü, sürekli ve ayrık dalgacık olmak üzere ikiye ayrılır.

3.3.2. Sürekli Dalgacık Dönüşümü

Sürekli Dalgacık Dönüşümü, sıkıştırımlar, genişlemeler ve dönüşümler ile tüm zaman ve ölçeklerdeki işaretler ile ana dalgacık arasında uygunluk sağlar [39].

$$C(a,b) = \int_R s(t) \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{b}\right) dt \quad (3.9)$$

Burada $s(t)$ işaret, $\psi(t)$ dalgacıktır, b dönüşüm etmeni olup, farklı frekans seviyelerinde ayrışım filtreleri tanımlar ve a ise ölçekleme etmeni olup her seviye için ayrışım filtrelerini ölçeklendirir.

3.3.3. Ayrık Dalgacık Dönüşümü

Sürekli dalga dönüşümünün hesaplama yükü fazla ve dönüşümü sağlamak için kullandığı bilgi miktarı oldukça büyük olduğundan ayrık dalgacık dönüşümü kullanılır [24]. Ayrık dalgacık dönüşümü ana dalgacığın sadece belirli genişlemelerinde işarete bakar. Tek bir filtre belirli bir frekans cevabına sahiptir ve işaret filtreden geçtikten sonra işaretin içindeki bilginin bir kısmı kaybolur. Bu yüzden tek bir filtre, filtrelenmiş işaretten tekrar asıl işareti yeniden elde etmek için kullanılamaz. Çünkü kaybolmuş bilgi tekrar kazanılamaz. İki filtre kullanılırsa, biri düşük frekans bilgisini, diğeri yüksek frekans bilgisini tutar. Bir başka deyişle işaret içindeki bilginin tümüne ulaşılır. Böylece bu iki filtrenin çıkışı asıl işareti yeniden elde etmek için birleştirilebilir. Filtrelerin çıkışından elde edilen veriyi örnekleme gerekir. Çünkü işaret, filtre bankasının her bir seviyesinden geçirildiğinde verinin miktarı iki katına çıkar [28].

3.4. Dalgacık Dönüşümünden Entropi Hesaplanması

Entropi, bir sistemin düzensizliğinin ölçüsüdür. Sistemin düzensizliği arttıkça artan herhangi bir fonksiyon rahatça entropi fonksiyonu olabilir. Entropi fonksiyonların doğruluğunu kanıtlar. Entropi, birçok alanda ve özellikle işaret işlemede kullanılır. Entropi işlemi için Shannon entropisi, Log energy entropisi, Norm entropi ve Thershold entropisi gibi kriterler kullanılmaktadır. Bu entropilerin matematiksel gösterimi aşağıda sırasıyla verilmiştir [28].

$$E(s) = -\sum_i s_i^2 \log(s_i^2) \quad (3.10)$$

$$E(s) = \sum_i \log(s_i^2) \quad (3.11)$$

$$E(s) = \sum_i |s_i|^p = \|s\|_p^p \quad (3.12)$$

$$E(s_i) = 1 \quad |s_i| > p \quad (3.13)$$

Burada s işaret, (s_i) işareti oluşturan örnek sayısı, i indis, E entropi değerleri (özellik vektörü), p isteğe bağlı bir parametredir. Denklem (3.10) daki Shannon entropide ve Denklem (3.11) daki Log energy entropisinde p parametresi kullanılmaz. Denklem (3.12) daki norm entropide p parametresi $1 \leq p$ ve Denklem (3.13) daki threshold entropide p parametresi $0 \leq p$ olmalıdır.

4. SINIFLANDIRMA

Sınıflandırmanın amacı, işaretleri özellik uzaylarına göre kendilerine en yakın sınıflara minimum hata ile eşleştirmektir. Sınıflandırmanın başarısı iyi belirlenmiş özellikler ile orantılıdır. Sınıflandırıcıları geleneksel ve akıllı olarak gruplara ayırmak mümkündür [24].

Geleneksel sınıflandırma algoritmaları istatistiksel bir yapı olan Bayes karar teorisi üzerine kuruludur. Bunların dezavantajları, özellik uzayını sınıflandırma uzayını dönüştürürken bir gürültünün çıkması ve her bir sınıf için hata ölçütlerinin belli olmamasıdır. Geleneksel sınıflandırıcılara çok değişkenli Gauss modelleri, en yakın komşu, maksimum olabilirlik, ikili karşın akıllı sınıflandırma yapıları genellikle YSA tabanlı olup, günümüzde en yaygın kullanılan ve başarısını ispatlanmış çok güçlü sınıflandırıcı türleridir. Aşağıda başlıca parametrik ve akıllı sınıflandırıcılar verilmiştir [24].

Parametrik sınıflandırıcılar

1. Bayes
2. Maksimum olabilirlik
3. En yakın komşu (k- NN)
4. Parzen pencereleri

Akıllı Sınıflandırıcılar

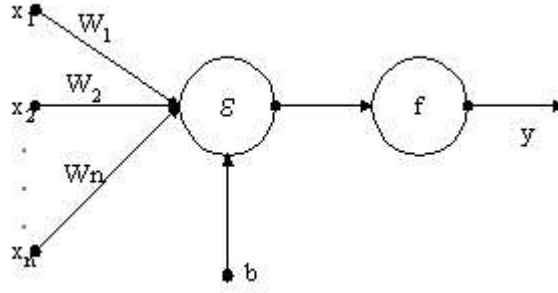
1. Yapay sinir ağları
2. Bulanık mantık
3. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi sınıflandırıcılardır.

4.1 Yapay Sinir Ağı Sınıflandırıcısı

Beynin bütün davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Yapay sinir ağları biyolojik nöron hücresinin yapısı ve öğrenme karakteristiklerinden esinlenerek geliştirilmiş bir hesaplama sistemi olduğundan örüntü tanımada iyi bir sınıflandırıcıdır. Yapay sinir ağları, özellikle bilgisayar teknolojisinin gelişmesi ile mühendislik alanında çok geniş bir yapıya sahiptir [28].

YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. En basit YSA yapısı Şekil 4.1’de gösterilmektedir. Donanım olarak elektronik devrelerle ya da bilgisayarlarda yazılım olarak

gerçekleştirebilmektedir. Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir. Öğrenme süreci, arzu edilen amaca ulaşmak için YSA ağırlıklarının yenilenmesini sağlayan öğrenme algoritmalarını içerir [40].



Şekil 4.1. En basit YSA hücresi.

Yapay sinir ağları aşağıdaki karakteristiklere sahip paralel bilgi işleme yapılarıdır [41].

- Biyolojik bir nörondan esinlenerek matematiksel modeli ortaya konmuştur.
- Birbirine bağlanan çok fazla sayıdaki işlem elemanlarından oluşur.
- Bağlantı ağırlıkları ile bilgiyi tutar.
- Bir işlem elemanı giriş uyarılarına dinamik olarak tepki verebilir ve tepki tamamen yerel bilgilere bağlıdır.
- Eğitim verisi ile ayarlanan bağlantı ağırlıkları sayesinde öğrenme, hatırlama ve genelleme yeteneklerine sahiptir.

4.1.1. Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağ geleneksel sınıflandırma metotlarından ayıran temel özellikler aşağıda sıralanmıştır.

- **Öğrenme:** Yapay sinir ağlarının istenilen davranışı gösterebilmesi için hücreler arasındaki bağlantıların doğru yapılması gerekir. YSA probleminden aldığı örnekleri inceleyerek aradaki ilişkiyi kavramaya çalışır yani problemi öğrenir.
- **Genelleme:** Klasik sınıflandırma yöntemlerinde, istenilen çıkışı üretmek için girişlerin tamamına ihtiyaç duyulmaktadır. Fakat yapay sinir ağları, girişinde değişimler olsa bile

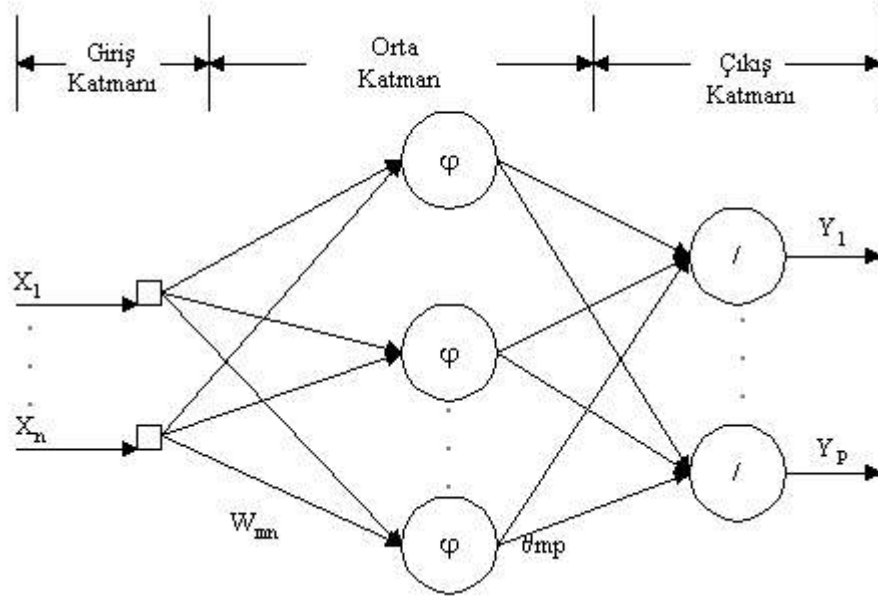
doğru çıkışı üretebilirler [42]. YSA, eğitim aşamasında verilmeyen giriş işaretleri için de sistemle aynı davranışı gösterebilir.

- **Çıkarım Yapma:** Yapay sinir ağları tam doğru olmayan bir eğitime kümesinden, tam doğruyu çıkarabilirler. Ses tanımak için eğitilmiş bir yapay sinir ağına, gürültülü bir ses verildiğinde ağ çıkışta doğru sesi üretebilir. Yani, eksik veya bozuk girişlere uygun sonucu verebilmektedir.
- **Hata toleransı:** İyi eğitilmiş ve genelleme kapasitesi yüksek bir sinir ağı, veriler eksik olsa da karar verme işlemine devam eder. Bunun dışında yapay sinir ağı üzerinde bir takım problemler ve bozukluklar olabilir. Geleneksel sistemlerin tersine yapay sinir ağları, bu durumda da çalışmasına devam eder. Verilerdeki eksiklik veya yapay sinir ağındaki yapısal bozukluk arttıkça yapay sinir ağının performansı yavaş yavaş azalmaya başlar. Fakat sistem fonksiyonunu tamamen durdurmaz ve mutlaka bir sonuç üretir. Bu özellikler yapay sinir ağının yapısından kaynaklanmaktadır. Çünkü ağ paralel bir yapıya sahip olduğundan , bilgi tüm hücrelere dağılmış durumdadır. Bundan dolayı birkaç bağlantının etkisiz hale gelmesi, sonucu etkilemez [43].
- **Hız:** Gerçek zaman uygulamalarında bilgi işleme hızı önemli bir yer teşkil eder. Sistemlerin her geçen gün biraz daha karmaşık olduğu, dolayısı ile daha fazla hacimde veriyi daha verimli bir şekilde işleme gerekliliği, yeni yazılım/donanım sistemlerinin zorunluluğunu ortaya çıkarmıştır. Yapay sinir ağlarının da, yine birbirlerine bağlı ve paralel işlem elemanlarından oluştuğundan böyle hızlı işleyebilmeleri, bu ağlara özellikle endüstriyel hayatta çok önemli olan gerçek zamanlı çalışma kabiliyeti kazandırır [28].

4.1.2. Yapay Sinir Ağlarının Yapısı

Yapay sinir ağları giriş, orta ve çıkış olmak üzere 3 katmandan oluşur. Şekil 4.2’de yapay sinir ağlarının yapısı gösterilmiştir.

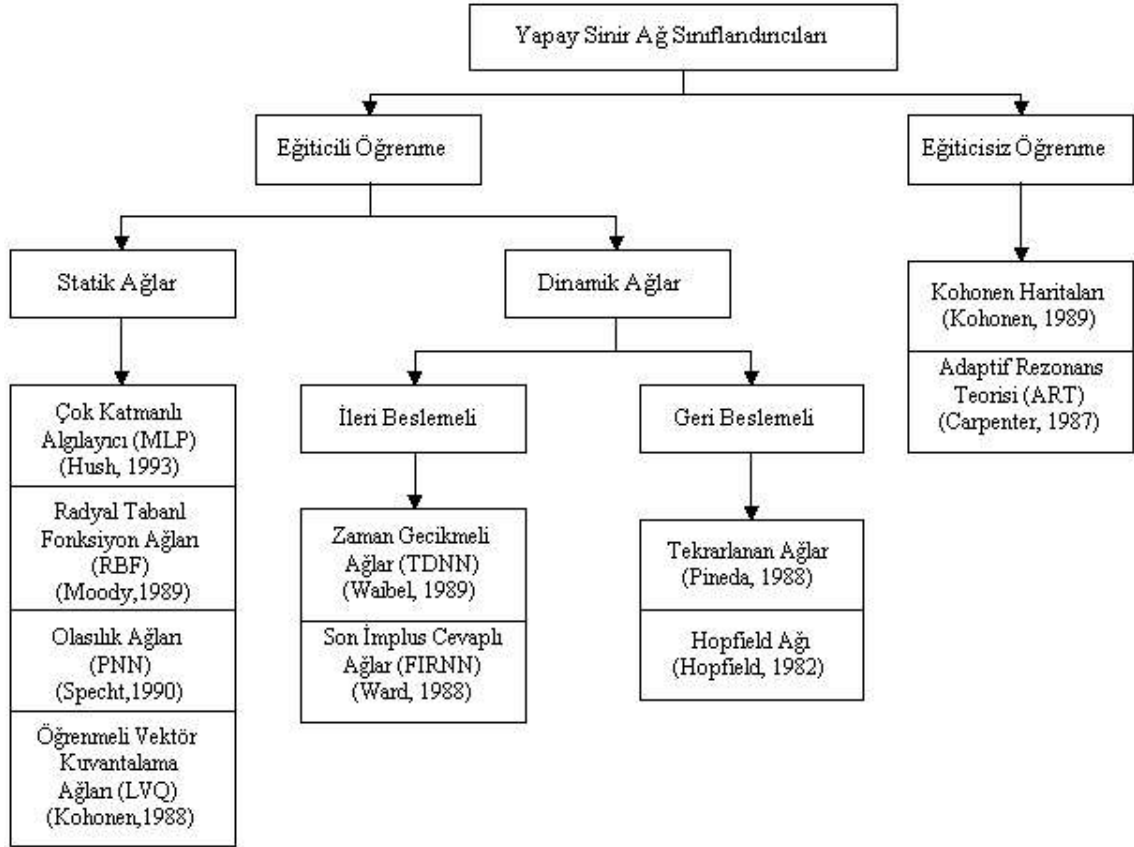
- Sinir ağının dış dünya ile bağlantısını kuran , *giriş katmanı*
- Gelen bilgileri işleyen, orta (*gizli*) *katman*
- Sinir ağının kararlarını dışarıya aktaran, *çıkış katmanı* bulunmaktadır.



Şekil 4.2. Yapay sinir ağlarının yapısı.

Giriş katmanında çoğu zaman bilgi işleme yapılmaz [40]. Bu katmandaki işlem elemanları aldıkları bilgiyi herhangi bir değişikliğe uğratmadan orta katmandaki işlem elemanlarına aktarırlar. Burada sözü edilen bilgi, sinir ağının işlem elemanları arasındaki bağlantı hatları üzerinde bulunan ağırlıklarla ifade edilir. Dolayısı ile bilgi, bütün ağa dağıtılmıştır. Bu bilgiler, yapay sinir ağları eğitildikten sonra kullanılacağı sistem için bir veri tabanıdır.

Tüm YSA modelleri sınıflandırıcı olarak kullanılabilir, fakat en yaygın kullanılan ve en güçlü sınıflandırıcı çok katmanlı ileri beslemeli ağıdır ve yapay sinir ağı uygulamalarının %90 nını kaplamaktadır. Sınıflandırıcı olarak kullanılan YSA türleri hiyerarşik bir biçimde Şekil 4.3’de gösterilmiştir [24].



Şekil 4.3. YSA sınıflandırıcılarının hiyerarşisi [28].

4.1.3. Yapay Sinir Ağları Aktivasyon Fonksiyonları

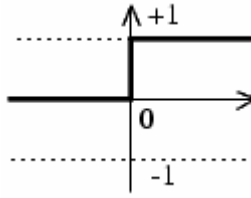
Yapay sinir ağı hücresinde temel olarak aşağıdaki kısımlar bulunmaktadır.

- girişler
- ağırlıklar
- toplama fonksiyonu
- aktivasyon fonksiyonu
- çıkışlar

Dış ortamdan alınan veri ağırlıklar aracılığıyla hücelere bağlanır ve bu ağırlıklar ilgili girişin etkisini belirler. Toplam fonksiyonu net girişi hesaplar, net giriş, girişlerle bu girişlerle ilgili ağırlıkların çarpımının bir sonucudur. Aktivasyon fonksiyonu işlem süresince net çıkışını hesaplar Genelde aktivasyon fonksiyonu doğrusal olmayan (nonlinear) bir fonksiyondur. Aşağıda aktivasyon fonksiyonlarının grafikleri ve matematiksel ifadeleri verilmiştir.

- **Eşik Fonksiyonu**

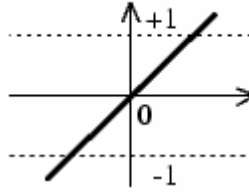
$$y = \begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \quad (4.1)$$



Şekil 4.4. Eşik aktivasyon fonksiyonu.

- **Doğrusal Fonksiyonu**

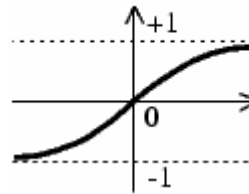
$$y = v \quad (4.2)$$



Şekil 4.5. Doğrusal aktivasyon fonksiyonu.

- **Tanjant Hiperbolik Fonksiyonu**

$$y = \frac{1 - e^{-v}}{1 + e^{-v}} \quad (4.3)$$



Şekil 4.6. Tanjant hiperbolik aktivasyon fonksiyonu.

4.1.4. Yapay Sinir Ağlarının Eğitimi

Yapay sinir ağlarının eğitimi, orta ve çıkış katman ağırlıklarının, katman çıkışlarının, hatanın hesaplanması, ağırlıkların yenilenmesi gibi aşamalardan oluşur. Eğitim aşaması aşağıda sırasıyla verilmiştir [40].

- Ağın giriş ağırlıklarını ve polarmaları belirlenir. Başlangıç ağırlıkları rasgele seçilir.
- Giriş katmanın ağırlıkları hesaplanır. Burada x giriş, b polarma ağırlığı, W giriş ağırlığı, V 1. katman çıkışı, n giriş sayısı ve m orta katman sayısıdır. ($i = 1,2,\dots,n$, $j = 1,2,\dots,m$)

- $$V_j = \sum_1^n W_{ji}x_i + b_j \quad (4.4)$$

- Giriş katmanının çıkışı aktivasyon fonksiyonundan geçirilir.

- $$O_j = v(V_j) \quad (4.5)$$

- Çıkış katmanı hesaplanır. Çıkış lineer olduğundan fonksiyondan geçirilmez. Burada $\square$ orta katman ağırlıklarıdır.

- $$y = \theta_j O_j + B \quad (4.6)$$

- Hata hesaplanır ve hatanın karelerinin toplamı bulunur. (SSE Sum Squared Error). Burada y , elde edilen çıkış, d , istenilen çıkıştır.

- $$e = d - y \quad (4.7)$$

- $$E = \frac{1}{2} e^2 \quad (4.8)$$

- Hataya bağlı olarak ağırlıklar yenilenir. Giriş katmanın ağırlıklarının yenilenmesi aşağıdaki gibi yapılır. Burada α , öğrenme katsayısıdır.

- $$W_{ji(k+1)} = W_{ji(k)} - \alpha \Delta W_{ji(k)} \quad (4.9)$$

- $$\Delta W_{ji} = \nabla E W_{ji} \quad (4.10)$$

- $$\nabla E W_{ji} = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial O_j} \frac{\partial O_j}{\partial v_j} \frac{\partial v_j}{\partial W_{ji}} \quad (4.11)$$

- Çıkış katmanın ağırlıkları yenilenir.

- $$\theta_j = \theta_j - \alpha \nabla E \theta_j \quad (4.12)$$

- $\nabla E \theta_j = \frac{\delta E}{\delta e} \frac{\delta e}{\delta y} \frac{\delta y}{\delta \theta}$ (4.13)

- Polarmaların ağırlıkları yenilenir.

- $\nabla EB = \frac{\delta E}{\delta e} \frac{\delta e}{\delta y} \frac{\delta y}{\delta B}$ (4.14)

- $\nabla Eb_j = \frac{\delta E}{\delta e} \frac{\delta e}{\delta y} \frac{\delta y}{\delta O_j} \frac{\delta O_j}{\delta v_j} \frac{\delta v_j}{\delta b_j}$ (4.15)

4.1.5. Yapay Sinir Ağlarında Öğrenme Algoritmaları

Öğrenme algoritmaları temelde üç grupta toplanmaktadır. Eğitici öğrenme, eğitici öğrenme ve takviyeli öğrenme algoritmalarıdır.

Eğitici öğrenmede, her bir örnekleme zamanında giriş uygulandığında sistemin arzu edilen çıkışı d eğitici tarafından sağlanır. Arzu edilen çıkış d ile yapay sinir ağı çıkışı y arasındaki fark hata ölçüsüdür ve ağ parametrelerini yenilemede kullanılır.

Eğitici öğrenmede, eğitici öğrenmedeki gibi arzu edilen d çıkışları bilinmemektedir. Bu yüzden kesin bir hata bilgisini ağı davranışını değiştirmekte kullanmak mümkün değildir. Cevabın doğruluğu veya yanlışlığı hakkında bilgi sahibi olunmadığı için öğrenme, girişlerin verdiği cevaplar gözlenerek başarıya ulaşılır.

Takviyeli öğrenmede, ağı davranışının uygun olup olmadığını belirten bir öz yetenek bilgisine ihtiyaç duyulur. Bu bilgiye göre ağırlıklar ayarlanır. Gerçek zamanda öğrenme yöntemi olup deneme-yanılma esasına göre sinir ağı eğitilmektedir.

Geri yayılım algoritması eğitici öğrenmede kullanılan en genel algoritmadır. Basit olması ve iyi bir öğrenme kapasitesine sahip olması birçok alana uygulanmasını sağlamıştır.

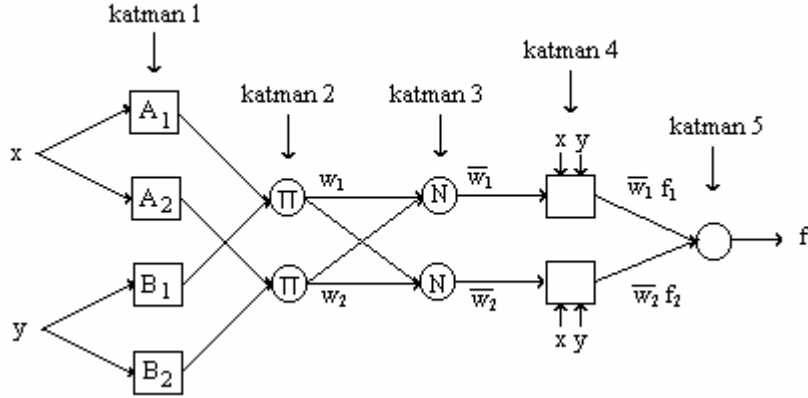
4.2. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi

Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sisteminin (UATBÇS) yapısında hem yapay sinir ağları hem de bulanık mantık kullanılmaktadır [44,45]. Yapı bakımından UATBÇS, bulanık çıkarım sistemindeki eğer-ise kuralları ve giriş çıkış bilgi çiftlerinden oluşur. Ancak sistem eğitiminde ve denetiminde yapay sinir ağı öğrenme algoritmaları kullanılır [40,46]. x ve y giriş, z ise çıkış olarak alındığında temel kural yapısı aşağıdaki gibi yazılmaktadır:

$$\text{Eğer } x \text{ A}_1 \text{ ve } y \text{ B}_1 \text{ ise } f_1 = p_1 x + q_1 y + r_1 \quad (4.16)$$

$$\text{Eğer } x \text{ A}_2 \text{ ve } y \text{ B}_2 \text{ ise } f_2 = p_2 x + q_2 y + r_2 \quad (4.17)$$

Burada p ve q lineer çıkış parametreleridir. İki girişli ve bir çıkışlı bir UATBÇS'nin temel yapısı Şekil 4.7'de verilmiştir. Bu yapı, 5 katman ve 4 adet eğer-ise kuralı kullanılarak oluşturulmuştur.



Şekil 4.7. İki girişli bir çıkışlı UATBÇS sisteminin temel yapısı.

UATBÇS sisteminde sugeno tipi bulanık denetleyici kullanılmıştır. İlk katmanda üyelik dereceleri, ikinci katmanda kuralların kesinlik dereceleri hesaplanmaktadır. Üçüncü katmanda normalizasyon yapılmakta ve bu katmanda aktivasyon fonksiyonları kullanılmaktadır. Dördüncü katmanda sugeno fonksiyonundan (lineer fonk.) geçirilmektedir. Beşinci katmanda ise toplam çıkış hesaplanmaktadır [47,48]. Şekil 4.8'de 2 girişli, 9 kurallı UATBÇS'nin şekli gösterilmiştir.

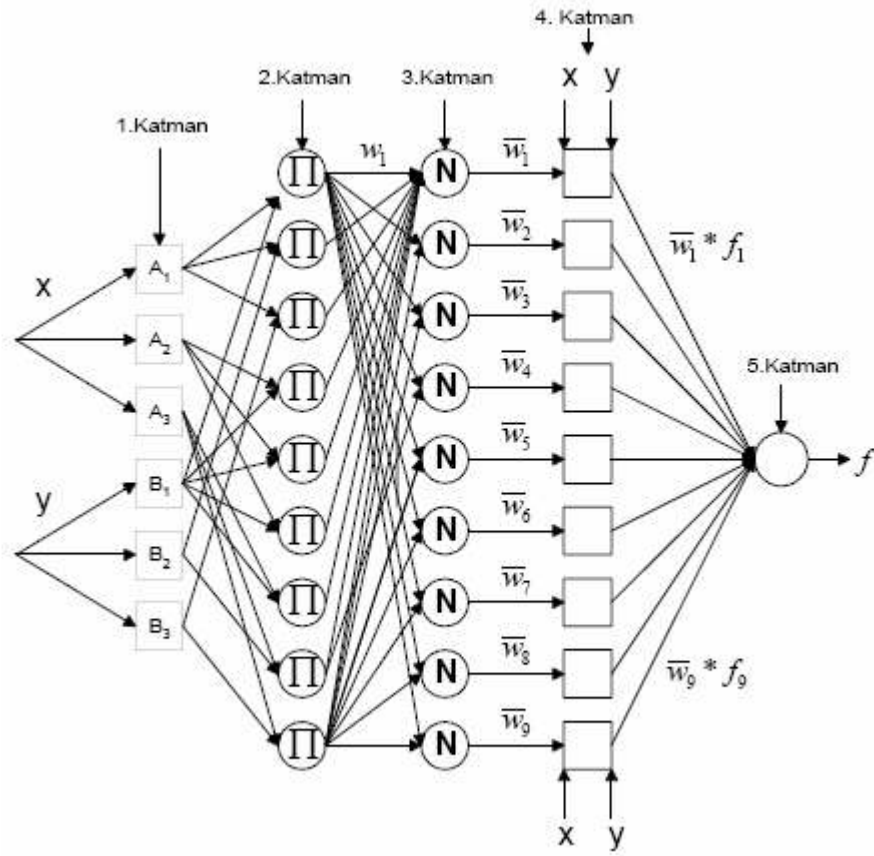
1.Katman: İki giriş ve bu iki girişin her birine tanımlanan üyelik fonksiyonları ile bu katmanda, A₁, A₂, A₃, B₁, B₂, B₃ olmak üzere toplam 6 hücre bulunmaktadır.

$$O_{1,i} = \mu_{A_i}(x), \quad i=1,2,3 \text{ için} \quad (4.18)$$

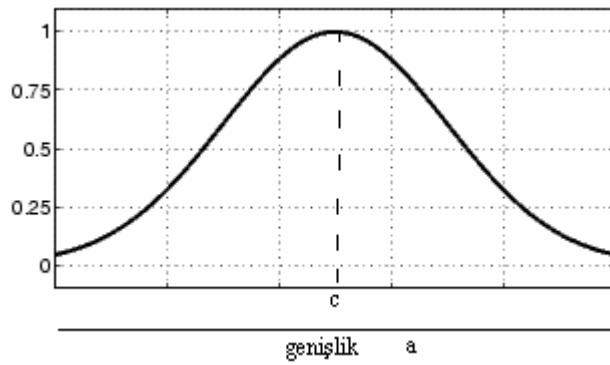
$$O_{1,i} = \mu_{B_{i-3}}(y), \quad i=4,5,6 \text{ için} \quad (4.19)$$

Burada x ve y girişlerdir. Bu katmanın çıkışı kuralların varsayım (eğer) kısımlarının üyelik fonksiyonlarına olan üyelik dereceleridir. Bu katmanda Şekil 4.9'da gösterildiği gibi gauss üyelik fonksiyonu kullanılmıştır. UATBÇS'nin temelinde gauss fonksiyonundaki c ve a değerlerinin ayarlanması vardır. c ve a giriş üyelik fonksiyon parametreleridir.

$$\mu_{Ai}(x), \mu_{Bi-3}(y) = e^{\left(\frac{(xi-ci)^2}{ai}\right)} \quad (4.20)$$



Şekil 4.8. İki girişli dokuz kurallı bir UATBÇS yapısı.



Şekil 4.9. Gauss üyelik fonksiyonu.

2.Katman: Bu katmanda kuralların kesinlik dereceleri, girişlerin cebirsel çarpımı ile bulunmaktadır.

$$O_{2,i} = w_i = \mu_{Ai}(x) * \mu_{Bi}(y) \quad i=1,2,3,\dots,9 \quad (4.21)$$

3.*Katman*: Bu katmanda kuralların normalizasyon işlemi yapılır.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{(w_1 + w_2)} \quad i=1,2,3,\dots,9 \quad (4.22)$$

4.*Katman*: Bu katmanda normalize edilmiş her bir kural kendine ait çıkış fonksiyonu ile çarpılmaktadır. Burada p,q ve r lineer parametreleri çıkış parametreleridir.

$$O_{4,i} = \bar{w}_i * f_i = \bar{w}_i * (p_i x + q_i y + r_i) \quad (4.23)$$

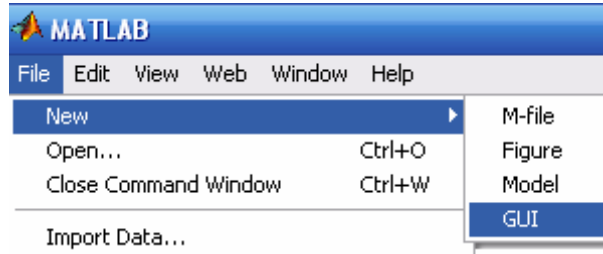
5.*Katman*: 4. katman çıkışlarının toplanarak toplam çıkışın hesaplandığı katmandır.

$$O_{5,i} = \text{toplam çıkış} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (4.24)$$

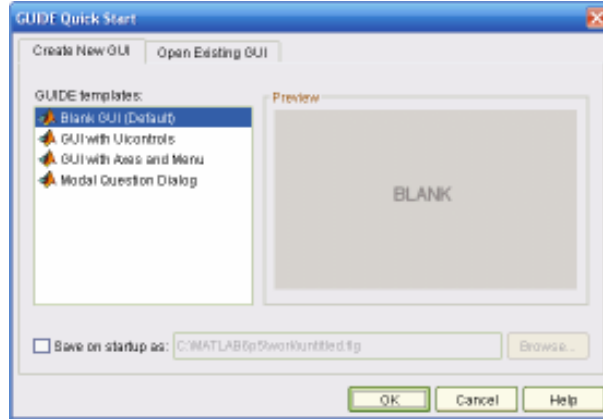
5. MATLAB GUI İLE ARAYÜZ OLUŞTURMA

Uygulama programlarında, kullanıcının program ile etkileşim içinde olabilmesi için kolayca veri değerlerini girip çıkışları anlamlı bir biçimde değerlendirebilmesi için, çeşitli grafiksel nesnelerden oluşan kullanıcı ara yüzleri oluşturulmaktadır [49]. Bu kullanım yapısına Grafiksel Kullanıcı Arayüzü (Graphical User Interface / GUI) denir. Birçok programlama dilinde olduğu gibi Matlab ile de GUI lar oluşturmak mümkündür.

GUI oluşturmak için, Şekil 5.1’de gösterildiği gibi File menüsünden New / GUI seçildiğinde Şekil 5.2’deki GUI penceresi ekrana gelir.

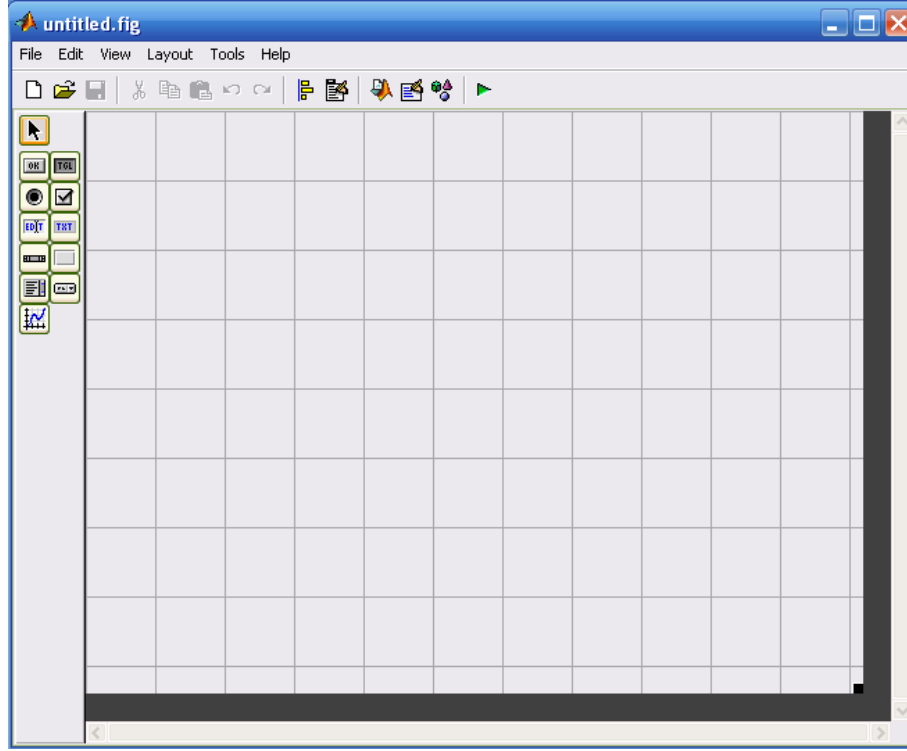


Şekil 5.1. Yeni bir GUI dosyası oluşturma menüsü.



Şekil 5.2. Yeni bir GUI dosyası oluşturma penceresi.

Açılan pencerede Create New GUI menüsünden yeni bir GUI dosyası, Open Existing GUI menüsünden ise daha önceden yüklü olan bir GUI dosyası açılır. Yeni bir GUI dosyası açıldığında Şekil 5.3’deki boş GUI penceresi ekrana gelir.

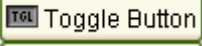


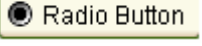
Şekil 5.3. Yeni bir GUI penceresi.

GUI dosyalarını kaydetmek ve çalıştırmak için grafik ve m dosyası olmak üzere iki dosya kullanılır. Grafik dosyası, “*fig*” uzantılı nesnelerin bulunduğu arayüz dosyasıdır. M dosyası “*m*” uzantılı her nesneye ait kodların yazıldığı dosyadır [50].

5.1. GUI Nesneleri

 Fare ile üzerine tıklandığında basılmış görünümü alan, fare bırakıldığında önen, yaylı buton özelliğindeki bileşendir. Basılı konumdan normal konuma dönerken “callback” fonksiyonu işleme konulmaktadır.

 Fare ile üzerine tıklandığında basılı konuma geçen, tekrar tıklandığında eski buton özelliğindeki bileşendir. Üzerine tıklandığında yani her konum değişiminde “callback” fonksiyonu işleme koyulmaktadır.

 Tıklandığında yuvarlak kutucuğa işaret koyan, tekrar tıklandığında işareti kaldıran bileşendir. GUI çıktısını, işaretlenen duruma göre sonuçlandırmak için kullanılmaktadır.

 **Checkbox** Tıklandığında işaret koyan, tekrar tıklandığında işareti kaldıran işaretleme butonudur. İşaretleme butonlarına atanan özellikler, işaretli olduğunda işleme konulmaktadır, aksi durumda işleme konulmaz.

 **Edit Text** Kullanıcının karakter dizisi girmesini sağlamaktadır. Kullanıcının metin girmesi gereken yerlerde kullanılmaktadır.

 **Static Text** Başlık atamak, isim vermek veya açıklama yapmak için kullanılan ve kullanıcı tarafından değiştirilemeyen kutucuklardır.

 **Slider** Fare ile oklara veya çubuğa tıklanarak yukarı-aşağı veya sağa-sola hareket edilmesini sağlayan, görüntüyü nümerik değerlere karşılık gelecek şekilde kaydıran bileşendir.

 **Frame** Axes haricindeki bileşenleri çerçeve içine alarak gruptandırmaya yarayan bileşendir. İşlem yaptırılan “callback” fonksiyonları bulunmamaktadır.

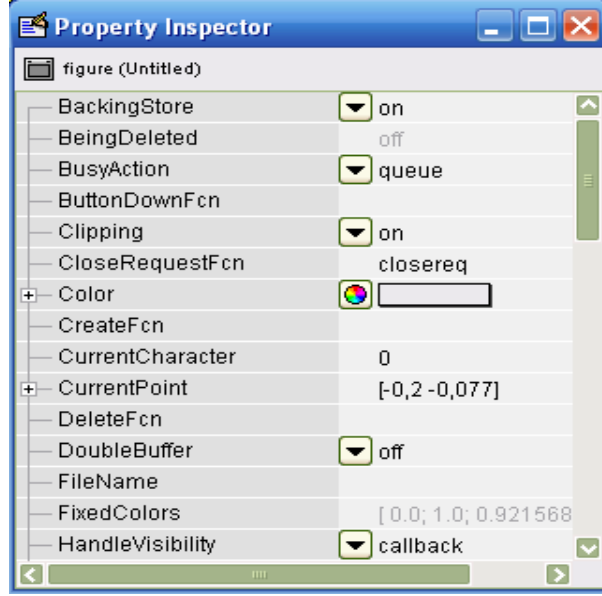
 **Listbox** Kullanıcı için birçok seçeneği kutucuk içinde görüntülemeyi sağlayan bileşendir. Kullanıcının bir seçeneği veya birkaç seçeneği aynı anda seçmesi sağlanmaktadır.

 **Popup Menu** Ok bölümüne tıklandığında birçok seçeneğin görüntülendiği, menü düzenindeki bileşendir.

 **Axes** Grafiklerin görüntülenmesi için kullanılan, eksenleri ve grafik alanını içeren bileşendir.

5.2. GUI Nesnelerinin Özellikleri

Her bir nesne seçilip çift tıklandığında o nesneye ait özellik penceresi (Property Inspector) açılmaktadır. Nesnelerle ilgili tüm özellik değişimleri (rengi, yazı türü, puntosu..) bu pencere yardımıyla yapılmaktadır. Şekil 5.4’de özellik penceresi görülmektedir.



Şekil 5.4. Nesnelere ait özellik penceresi.

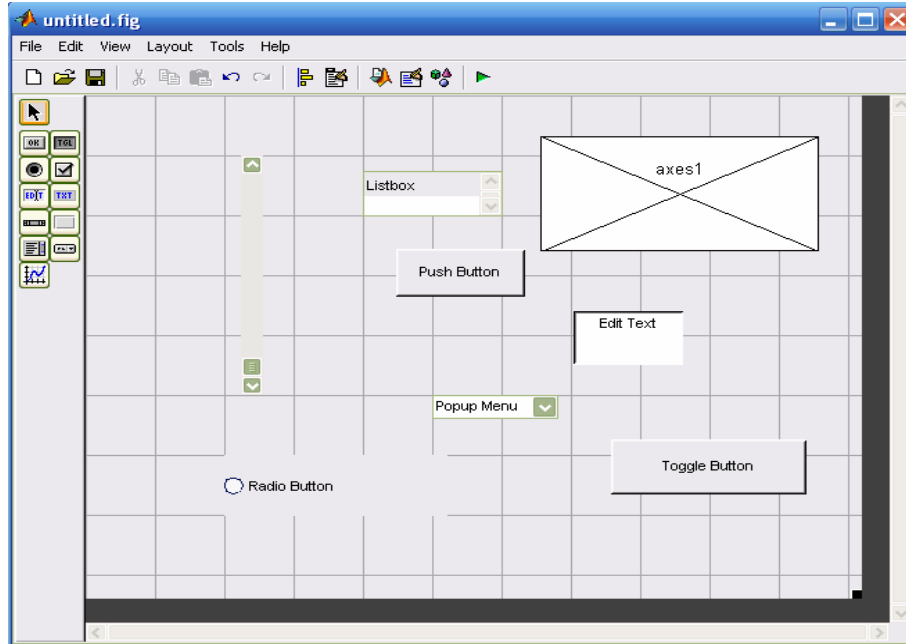
Nesnelere ait özellikler aşağıda açıklanmıştır [50].

- **BackgroundColor** : Seçilen nesnenin yüzey rengini belirlemek için kullanılmaktadır.
- **Busy Action** : GUI işlemlerinin üst üste gelmesi durumunda, bileşenin davranışını belirlemektedir.
- **ButtonDownFcn** : Bileşen üzerinde veya bileşene beş piksel yakınlıkta, fare ile tıklandığında çalıştırılacak olan fonksiyonu m-dosyasından çağıran komutun girildiği bölümdür.
- **CData** : Bileşenin yüzeyinin alacalı renkler almasını sağlayan özelliktir. Bu özellik kullanıldığında bileşen üzerindeki yazı görüntülenmez.
- **Callback** : GUIde editöründe hazırlanan GUI kaydedildikten sonra, GUI'nin tüm kodlarını içeren m-dosyasının içine, bileşenin gerçekleştireceği işlemleri içeren “callback” fonksiyonun temeli oluşturulmaktadır.
- **CreateFcn** : Bileşen oluşturulduğunda MATLAB programının çalıştırdığı “callback” fonksiyonudur. Yeni oluşturulacak bileşen için bu fonksiyon çalıştırılmaktadır.
- **DeleteFcn** : Bileşen silinmeden önce MATLAB programı tarafından çalıştırılan “callback” fonksiyonudur.
- **Enable** : Bileşenin pasif duruma gelmesini sağlamaktadır.

- **FontAngle** : Bileşen üstündeki metnin hafif yatık veya çok yatık görüntülenmesini sağlamaktadır.
- **FontName** : Bileşen üstündeki metnin yazı tipini belirlemek için kullanılmaktadır.
- **Font Size** : Bileşenin üstündeki metnin yazı boyutunu belirlemek için kullanılmaktadır.
- **FontWeight** : Bileşen üstündeki metnin kalın, koyu veya ince görüntülenmesini sağlamaktadır.
- **FontUnits** : Bileşenin üstündeki yazının büyüklüğünün santimetre, inch, pixel, point, gibi birimlerle ifade edilmesini sağlar.
- **ForegroundColor** : Bileşenin üstündeki yazı rengini belirlemek için kullanılmaktadır.
- **HandleVisibility** : GUI ile ilgili “handle” fonksiyonlarının komut penceresinden görüntülenip görüntülenmemesini ayarlamak için kullanılmaktadır.
- **HorizontalAligment** : Bileşenin üzerindeki yazının, “Position” özelliğinde belirtilen alan içinde, yatay olarak nasıl görüntülenmesi gerektiğini belirlemek için kullanılmaktadır.
- **Interruptible** : “No” seçeneği seçilirse, bu bileşene tıklandıktan sonra işlem sürmekte iken başka bir bileşene işlem yaptırılmak istenildiğinde, ilk işlem sonlandırılmakta ve sonra ikinci işleme geçilmektedir. ”Yes” seçeneği seçilirse, önce ikinci işlemi sonlandırılır daha sonra birinci işleme döner ve birinci işlemi sonlandırmaktadır.
- **Max** : “Value” bölümüne girilmesine izin verilen maksimum değeri belirlemektedir.
- **Min** : “Value” bölümüne girilmesine izin verilen minimum değeri belirlemektedir.
- **Position** : Bileşenin GUI üzerindeki yerinin, koordinatlar cinsinden ve boyutlarının da sayısal olarak belirlenebilmesini sağlamaktadır.
- **SelectionHighlight** : “On” seçeneği seçildiğinde, bileşen ile işlem yaptırıldıktan sonra bileşenin etrafı belirginleştirici siyah bir kutucuğa dönüşmektedir. ”Off” seçeneği seçildiğinde ise işlem yaptırıldıktan sonra bileşen siyah çerçeve içine alınmamaktadır.
- **SliderStep** : GUI çalıştığında bileşen üzerinde, “Slider” çubuğunun, oka her tıklandığında yatay veya düşey eksenlerde girilen değerler kadar hareket etmesini sağlamaktadır.

- **String** : Bileşenin üzerine yazı yazmak için kullanılmaktadır.
- **Style** : Bileşenin cinsini değiştirmek için kullanılmaktadır.
- **Tag** : Bu bölüme girilen metin, bileşenin m-dosyasındaki “callback” alt fonksiyonunun ismi olarak atanmaktadır.
- **TooltipString** : GUI penceresinde fare, bileşen üzerinde bekletildiğinde beliren ve bileşenle ilgili açıklamayı içeren metnin görüntülenmesini sağlamaktadır.
- **UIContextMenu** : Bileşenin üzerinde farenin sağ tuşuna tıklandığında belirmesi istenen menünün, “Menu Editör” ile oluşturulan “Context” menüleri arasından seçilmesini sağlamaktadır.
- **Units** : “Position” bölümüne girilen değerlerin birimini belirlemeye yaramaktadır.

Kullanıcı sol taraftaki araç kutusundan istediği kontrol nesnesini seçerek, fare yardımı ve sürükle – bırak yöntemi ile tasarım alanına yerleştirebilmektedir [49]. Örneğin metin kutusu nesnesi seçili iken bunu tasarım alanına taşıdıktan sonra farenin sol tuşu basılı tutarak fare hareket ettirilerek kutu istenilen boyuta ayarlanabilmektedir. Şekil 5.5’de gösterildiği gibi istenilen nesneler tasarım alanına taşınabilmektedir.



Şekil 5.5. Tasarım alanının ve nesnelerin kullanımı.

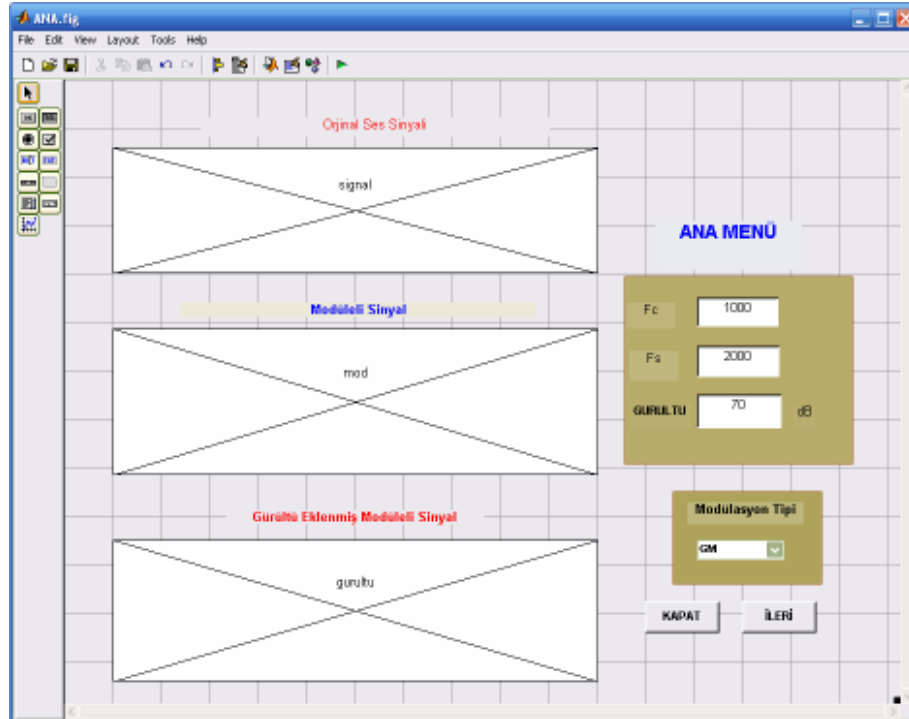
5.3. Arayüz Tasarım Aşamaları

Analog modülasyon tanıma çalışmalarında kolaylık sağlaması ve etkileşimi arttırması amacıyla MATLAB 6.5 ile arayüz tasarlanmıştır. Arayüz tasarımında aşağıdaki dosyalar kullanılmıştır.

- ANA.m: Projenin çalıştığı ana dosyadır. Arayüze bu dosya üzerinde ulaşılmaktadır.
- WAV.m: Özellik çıkarmak için dalgacık dönüşümünün yapıldığı dosyadır.
- SECIM.m: Sınıflandırma yapmak için YSA veya UATBÇS 'nin seçildiği dosyadır.
- YSA.m: Yapay sinir ağları kullanılarak modüleli işaretler için eğitim ve testin yapıldığı dosyadır.
- UATBÇS.m: Uyarlamalı ağ tabanlı bulanık çıkarım sistemi kullanılarak modüleli işaretler için eğitim ve testin yapıldığı dosyadır.

5.3.1. ANA.m Dosyasının Oluşturulması

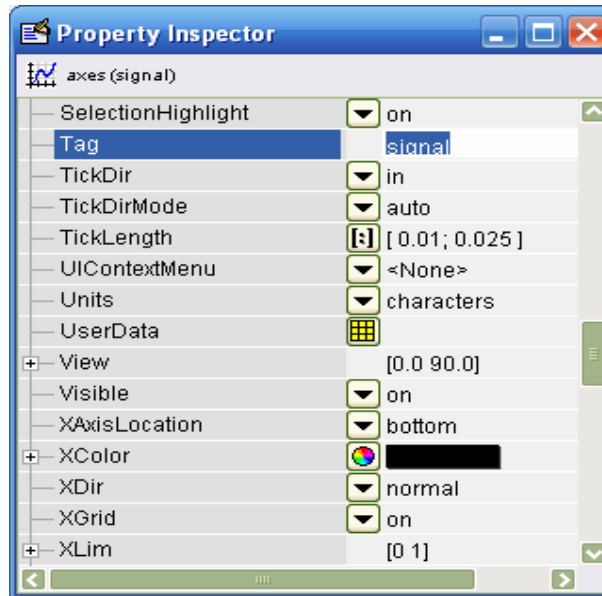
MATLAB GUI ile arayüz tasarlanırken iki ayrı dosya ile çalışılır. Birinci dosya arayüz tasarımının yapıldığı şekil dosyasıdır ve “fig” uzantılı olarak kaydedilir. İkinci dosya şekil dosyasına ait kodların yazıldığı m-file dosyasıdır ve “m” uzantılı olarak kaydedilir. M-file dosyasına ulaşmak için şekil dosyası kaydedilir ve ardından View/m-file editör menüsüne girilir veya  simgesine tıklanır. Ana menü arayüzü Şekil 5.6’da görülmektedir.



Şekil 5.6. Ana menü arayüz penceresi.

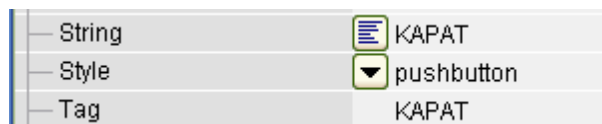
Ana menü penceresinde ses işareti, modüleli işaret ve gürültü ilave edilmiş modüleli işareti gösteren üç adet axes nesnesi, kapat ve ileri push butonları, modülasyon tipini belirleyen popup menü kutusu, Fc, Fs ve GÜRÜLTÜ değerlerinin girildiği edit text, tüm yazılar için static text nesneleri ve frame kullanılmıştır.

Axes nesnesi plot veya herhangi bir grafik komutuyla çizilen grafiklerin koordinat ekseninde gösterilmesini sağlayan nesnedir. Axes nesnesinin seçilip tasarım ekranına bırakılması yeterlidir. Axes nesnelerine signal, mod ver gurultu isimleri verilmiştir. Axes nesnesine isim vermek için Şekil 5.7'deki özellik penceresinden Tag özelliği seçilir. Özellik penceresini açmak için seçilen nesneye çift tıklanır veya  simgesine tıklanır.



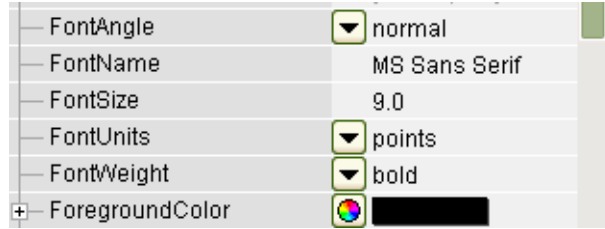
Şekil 5.7. Axes nesnesine isim verilmesi

Push butonlara isim vermek için özellik penceresinden String özelliği kullanılır. Tag özelliğine verilen isim m-file dosyasında kullanılır. Nesnelerin String ve Tag isimleri karışıklığı önlemek amacıyla aynı olmalıdır. Şekil 5.8'de KAPAT push butona isim verilmesi görülmektedir.



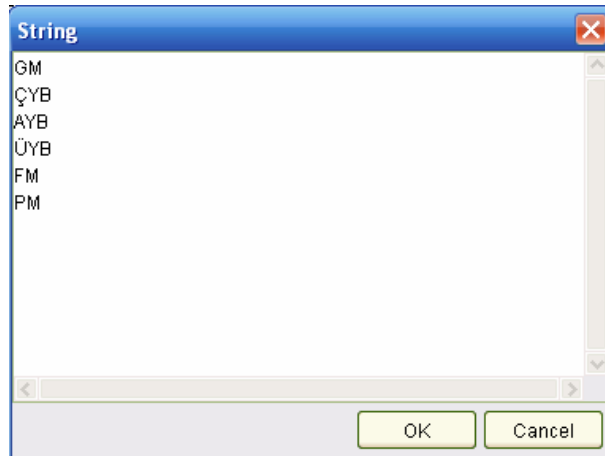
Şekil 5.8. Push buton nesnesine isim verilmesi.

Static text nesnelere isim vermek için String özelliği kullanılır. Bu nesnelerin yazı tipini, boyutunu, kalınlığını ve rengini değiştirmek için özellik penceresi kullanılır. Şekil 5.9’da gösterildiği gibi FontName yazı tipini, FontSize boyutunu, FontWeight kalınlığını ve ForegroundColor rengini değiştirmektedir.



Şekil 5.9. String nesnesine ait özellik penceresi.

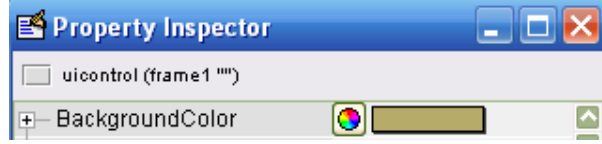
Popup menü ile kullanıcının modülasyon tipini seçmesi sağlanır. Popup menüye değer girmek için Şekil 5.10’daki gibi String özelliği kullanılır.



Şekil 5.10. Popup menüye değer girilmesi.

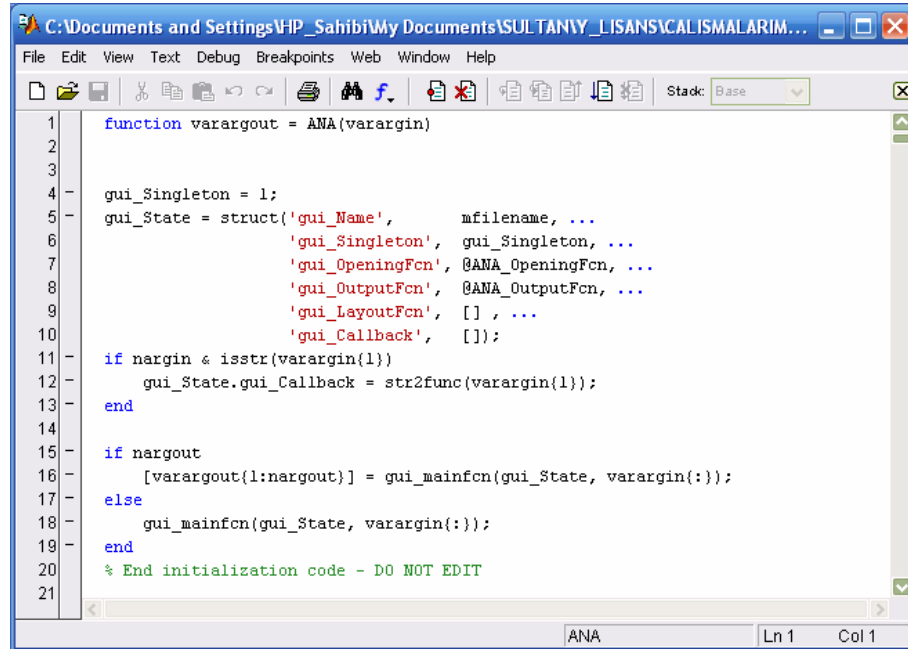
F_c , F_s ve gürültü oranının girildiği üç adet edit text kutusu kullanılmaktadır. Bu değişkenlerin varsayılanı yani program ilk çalıştırıldığında ekranda görülmesi istenen değeri string özelliğiyle belirlenmektedir. F_c 1000 Hz, F_s 2000 Hz ve gürültü oranı da 70dB olarak belirlenmiştir. Bu değerler kullanıcı tarafından değiştirilebilmektedir.

Frame nesnesi gruplandırma yapmak için kullanılmaktadır. Frame nesnesinin rengini değiştirmek için Şekil 5.11’de gösterildiği gibi BackgroundColor özelliği kullanılmaktadır.



Şekil 5.11. Frame nesnesinin rengini değiştirme.

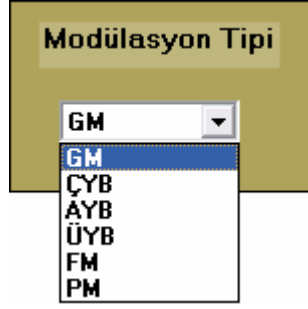
Tasarım ekranındaki düzenlemeler yapıldıktan sonra File menüsünden Save komutu seçilir ve dosya “ANA.fig” olarak kaydedilir. Kayıt işleminin ardından Şekil 5.12’de gösterildiği gibi ANA.m dosyası açılır.



Şekil 5.12. Ana menü m-file dosyası.

M-file dosyasına nesnelerin görevini yerine getirebilmesi için komutlar yazılır. KAPAT butonu, aktif olan ANA penceresini kapatır. İLERİ butonu, ANA penceresini kapatıp WAV penceresini açar. ANA dosyasından WAV dosyasına geçerken modülasyon tipi, gürültü oranı, F_c ve F_s değerleri de beraber gönderilir.

Ses işareti Şekil 5.13’de gösterildiği gibi kullanıcı tarafından belirlenen GM, ÇYB, AYB, ÜYB, FM, PM modülasyon tiplerinden birine tabi tutulur. Modüleli işaretler oluşturulduktan sonra işaretlere 0 – 100 dB arasında istenilen oranda gürültü ilavesi yapılmaktadır.



Şekil 5.13. Modülasyon tipini seçme menüsü.

Axes nesnelerinden birincisi bilgi işareti olan gerçek ses işaretini, ikincisi kullanıcının seçtiği modülasyon tiplerinden biriyle modüle edilmiş işareti, üçüncüsü ise modüleli işarete gürültü ilave edilmiş işareti gösterir. Kullanıcının yaptığı değişiklikler axes nesnesinde net olarak gözlenebilmektedir. M-file dosyasına yazılan nesnelerin çalışmasına ait tüm komutlar callback ifadesinden sonra yazılır. Programda açıklama yapmak için “%” işareti kullanılır. M-file dosyasına yazılan komut parçaları aşağıda sıralanmıştır.

- *Kapat butonunu oluşturmak*

```
function varargout = KAPAT_Callback(hObject, eventdata, handles, varargin)
close(ANA) ; ANA.m dosyasını kapat
```

- *İleri butonunu oluşturmak*

```
function ILERI_Callback(hObject, eventdata, handles)
WAV(z,a,fcc,fss,N) ; WAV dosyasını aç
close(ANA) ; ANA dosyasını kapat
```

- *Modüleli işaret çizmek*

Ana menü penceresinde kullanıcının girişi için Fc, Fs ve gürültü değerlerinin programda kullanılması “get” komutu ile gerçekleştirilir. Alınan değerler string olduğundan bunları sayısal ifadelere dönüştürmek için str2double komutu kullanılır.

```
function varargout = ANA(varargin)
fcc = str2double(get(handles.fc,'String')); ; fc değerini oku
fss = str2double(get(handles.fs,'String')); ; fs değerini oku
N= str2double(get(handles.nois,'String')); ; Gürültü değerini oku
```

am1_1= amod(handles.x1,fcc,[fss 0],'amdsb-tc');	; GM işaretini oluştur
axes(handles.mod)	; Mod axes nesnesini seç
plot(am1_1)	;GM işaretini axese çiz
set(handles.mod,'XMinorTick','on')	;Axes çizgilerini aktif yap
grid on	;Axese çizgi çiz

- F_c ve F_s metin kutularının güncellenmesi

Metin kutusuna sayısal bir ifade girilmesi gerekir. Kullanıcı yanlışlıkla string bir ifade girerse kullanıcı Şekil 5.14'deki gibi uyarı mesajı ile karşılaşır ve ardından metin kutularının içeriği varsayılan olan değerler ile değiştirilir. Mesajı yazmak için fc ve fs metin kutularının callback kısmına aşağıdaki program parçaları yazılır.

```
function fc_Callback(hObject, eventdata, handles)
fcc = str2double(get(handles.fc,'String'));           ;fc değerini oku
if isnan(fcc)                                         ;fc sayısal olmayan bir değerse
msgbox('Lütfen sayısal bir değer giriniz','HATALI GİRİŞ','warn') ;Mesajını göster
set(handles.fc,'string','1000');                     ;fc değerini 1000 olarak değiştir
end
```



Şekil 5.14. F_c ve F_s metin kutusu için uyarı mesajı.

- $G_{\text{ürültü}}(N)$ metin kutusunun güncellenmesi

Gürültü metin kutusuna 0 ile 100 dB arasında bir sayı girilmesi gerekmektedir. Girilen sayı bu sınırlar arasında veya sayısal bir ifade değilse Şekil 5.15'de gösterildiği gibi bir uyarı mesajı verilmektedir. Gürültü metin kutusu için yazılan program parçası aşağıda verilmiştir.

```
function nois_Callback(hObject, eventdata, handles)
N= str2double(get(handles.nois,'String'));           ; N değerini oku
```

```

if isnan(N)                                ; N sayısal olmayan bir değerse
msgbox('Lütfen sayısal bir değer giriniz','HATALI GİRİŞ','warn') ; Mesajı göster
set(handles.nois,'string','70');           ; N değerini 70 yap
elseif N<0                                ; N 0 dan küçükse
msgbox('Lütfen 0 dan büyük bir değer giriniz.','HATALI GİRİŞ','warn') ; Mesajı göster
set(handles.nois,'string','70');           ; N değerini 70 yap
elseif N>100                               ; N 100 den büyükse
msgbox('Lütfen 0 ile 100 arasında bir değer giriniz','HATALI GİRİŞ','warn') ; Mesajı
göster
set(handles.nois,'string','70');           ; N değerini 70 yap
end

```



Şekil 5.15. Gürültü metin kutusu uyarı mesajı.

- *Modülasyon tipinin seçilmesi ve yazılması*

Şekil 5.13’de gösterildiği gibi GM, ÇYB, AYB, ÜYB, FM ve PM olmak üzere 6 adet modülasyon tipi kullanılmıştır. Modülasyon tipini belirlemek için önce sıra bulunur ardından popup menüdeki tüm string değerleri alınır. Bunun için aşağıdaki program parçası yazılır.

```

val = get(handles.module,'Value');         ; Seçilen modülasyon tipinin sırasını al
str = get(handles.module, 'String');       ; Popup menüdeki tüm stringleri al

```

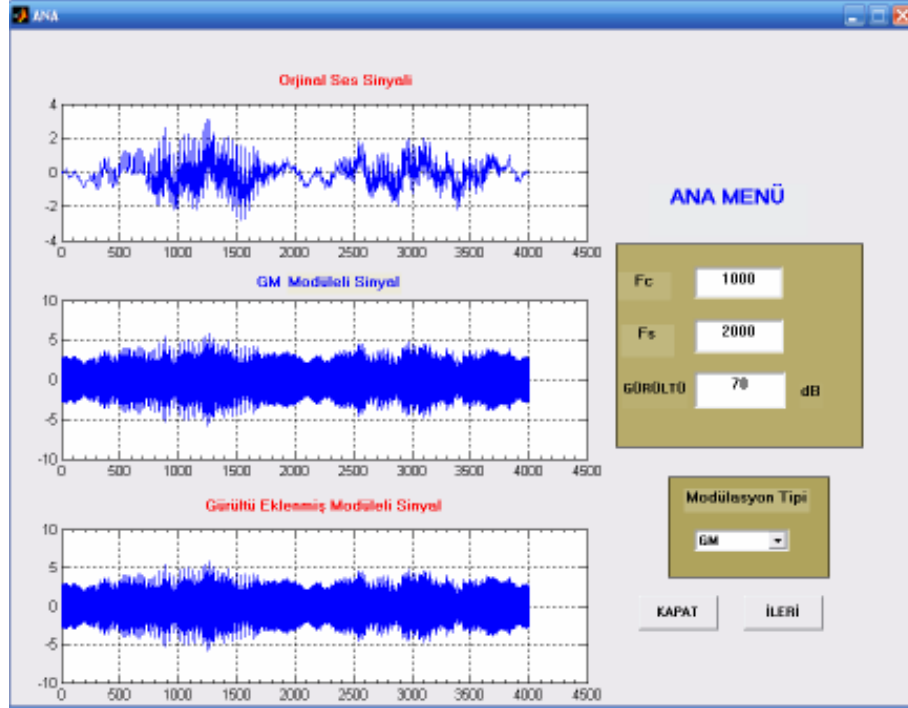
Modülasyon tiplerinden ÇYB seçildiğinde MATLAB ekranında val=2, str= ‘GM’, ‘CYB’, ‘AYB’, ‘UYB’, ‘FM’, ‘PM’ elde edilir. Ekranda CYB yazdırmak için aşağıdaki komutlar yazılır.

```

handles.m=str{val}                         ; handles.m değişkenini CYB olarak seç
set(handles.moode,'string',handles.m);    ; Değişkeni moode text kutusuna yaz

```

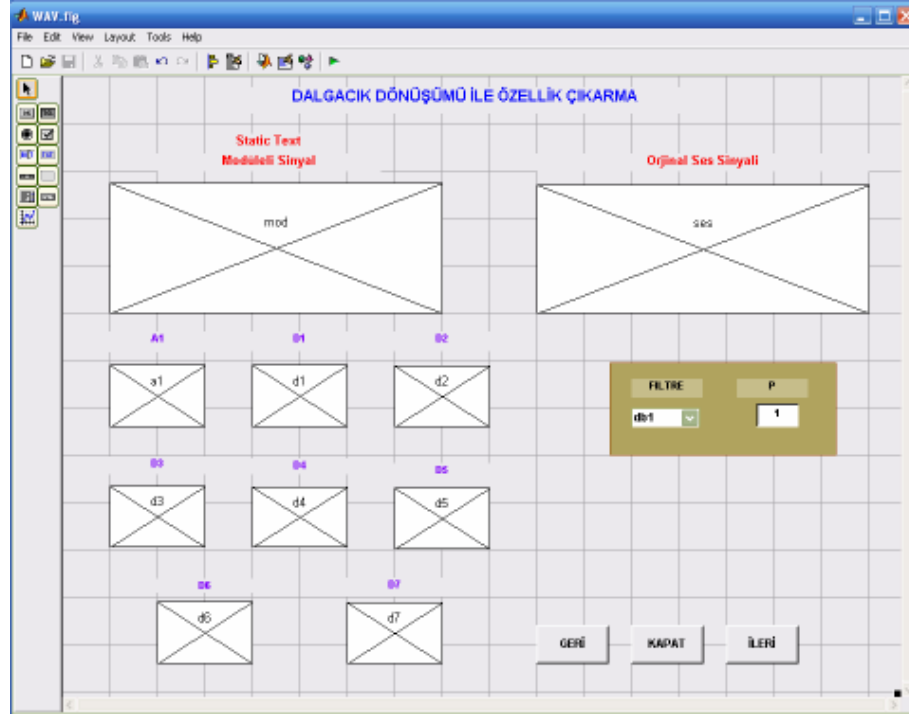
M-file dosyası oluşturulduktan sonra kaydedilir. Daha sonra Debug menüsünden Run komutuna veya  simgesine tıklanarak çalıştırıldığında Şekil 5.16'daki kullanıcı arayüz ekranı karşımıza gelir.



Şekil 5.16. Ana menü kullanıcı arayüz penceresi.

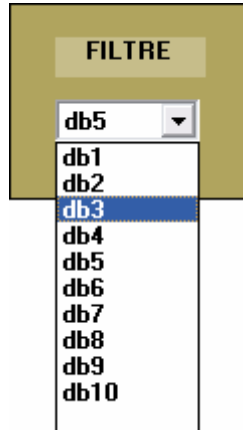
5.3.2. WAV.m Dosyasının Oluşturulması

WAV yani özellik çıkarma penceresinde ses işareti, modüleli işaret dalgacık dönüşümünden geçirilen işaretin detay ve yaklaşıklarını gösteren on adet axes nesnesi, kapat, ileri ve geri push butonları, dalgacık dönüşümünde kullanılacak olan filtre tipini belirleyen popupmenü kutusu, P değerinin girildiği edit text, tüm yazılar için static text nesneleri kullanılmıştır. Özellik çıkarma arayüzü Şekil 5.17'de görülmektedir.



Şekil 5.17. Özellik çıkarma arayüz penceresi.

Özellik çıkarma penceresinde ana menüden seçilen modüleli işaret 7 seviyeli dalgacık dönüşümünden geçirilir. 7 seviyeli dalgacık dönüşümünde bir tane yaklaşık, 7 tane de detay işaretleri elde edilir. Dalgacık dönüşümünde Haar, Daubechies, Biorthogonal, Coiflets, Symlets filtreleri kullanılmaktadır. Bu simülasyonda Şekil 5.18’de gösterildiği gibi daubechies ailesi filtre olarak (db1 – db10) kullanılmıştır.



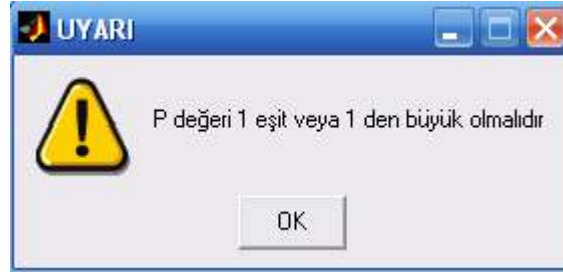
Şekil 5.18. Filtre seçim menüsü.

Dalgacık dönüşümünün sonucunda elde edilen özellikler entropi işleminden geçirilir. Entropi olarak norm entropi kullanılmıştır. Norm entropi işleminde eşik değeri $P=1$ veya $P>1$ olmak zorundadır. Simülasyonda P değeri 1’den küçük girildiğinde Şekil 5.19’daki gibi hata mesajı ile kullanıcı uyarılmaktadır. Hata mesajını elde etmek için aşağıdaki komut satırı yazılır.

```

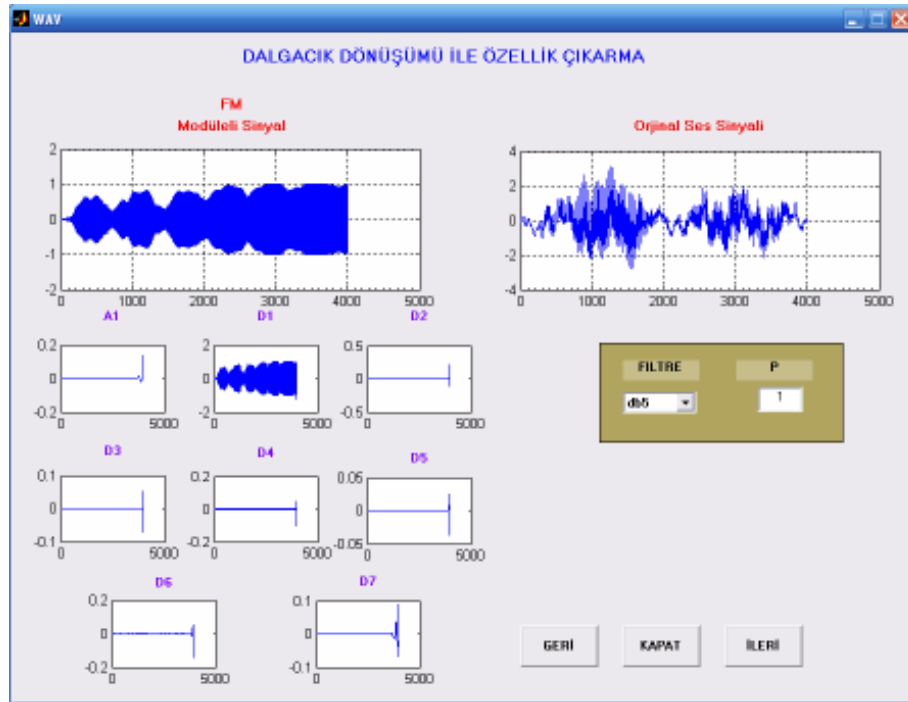
function P_Callback(hObject, eventdata, handles)
P = str2double(get(handles.P,'String'));           ; P değerini sayısal olarak al
if P<1                                             ; P birden küçükse
msgbox('P değeri 1 eşit veya 1 den büyük olmalıdır','UYARI','warn'); ; mesajı göster
set(handles.P,'string','1');                     ; P kutusuna tekrar 1 yaz
else                                              ; P birden küçük değilse
end                                              ; Programdan çık

```



Şekil 5.19. P değeri hatalı girildiğinde ekrana gelen uyarı mesajı.

Özellik çıkarma penceresinde gerekli düzenlemeler yapıldıktan sonra simülasyon çalıştırıldığında Şekil 5.20'deki arayüz penceresi açılmaktadır.



Şekil 5.20. Özellik çıkarma kullanıcı arayüz penceresi.

Özellik penceresi m-file dosyasına yazılan komut parçaları aşağıda sıralanmıştır:

- *Geri butonunu oluşturmak*

```
function GERI_Callback(hObject, eventdata, handles)
close('WAV') ; Özellik penceresini kapat
open (ANA) ; Ana menü penceresini aç
```

- *İleri butonunu oluşturmak*

İleri butonuna tıklandığında ana menü dosyasından alınan Fc, Fs, N değerleri ve özellik penceresi dosyasında bulunan P, filtre değerleri değişkenlere kaydedilir. Ardından “mode.m” dosyası açılarak buraya yazdırılmaktadır. Bu dosya YSA ve UATBCS dosyalarında açılarak kullanılmaktadır.

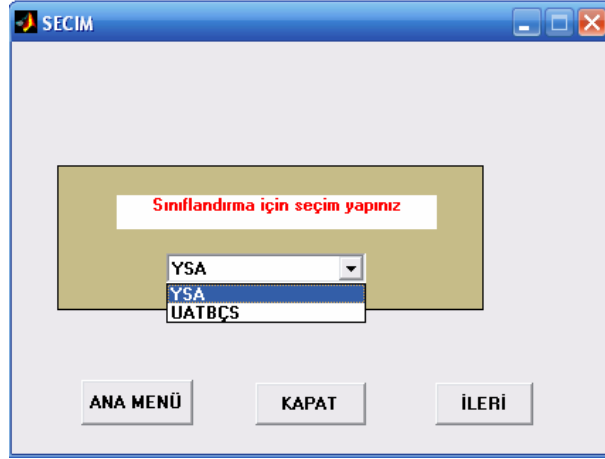
```
function ILERI_Callback(hObject, eventdata, handles)
P = str2double(get(handles.P,'String')); ; P değerini al
FC= handles.fcc ; fcc değerini al
FS= handles.fss ; fss değerini al
G= handles.N ; Gürültü değerini al
handles.val = get(handles.db,'Value') ; Popup menüden seçilen
; filtrenin sırasını al
fltr=handles.val ; Bu değeri fltr değişkenine ata
ffm = fopen('mode.m','w'); ; mode.m dosyasını aç
fprintf(ffm,'%d %d %d %f %d\n',FC,FS,G,P,fltr); ; FC, FS, G, P, fltr değerlerini
; mode.m dosyasına yaz
fclose(ffm); ; mode.m dosyasını kapat
close('WAV') ; Özellik penceresini kapat
open(SECIM) ; Seçim penceresini aç
```

5.3.3. SECIM.m Dosyasının Oluşturulması

Bu dosyada sınıflandırma yapmak için Şekil 5.21’de gösterildiği gibi yapay sinir ağları ve uyarlamalı ağ tabanlı bulanık çıkarım sistemi yöntemlerinden biri seçilir. Seçim için aşağıdaki komutlar yazılır.

```
val = get(handles.SINIF,'Value'); ; Popup menüden seçilen değeri al
if val==1 ; Değer 1 ise
```

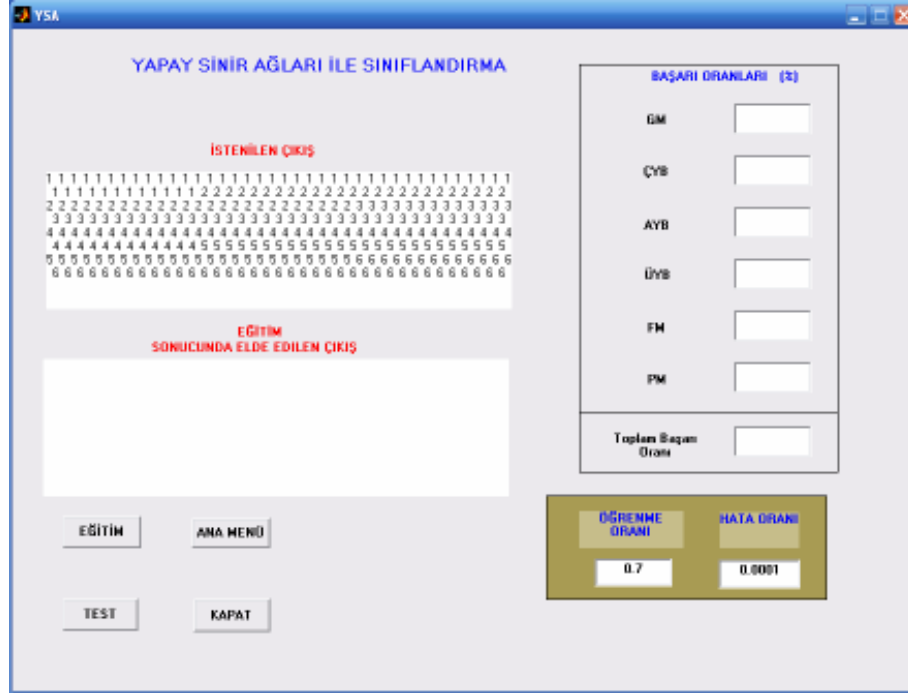
close('SECIM')	; Seçim penceresini kapat
open(YSA)	; YSA penceresini aç
else	; Değer 1 değilse
close('SECIM')	; Seçim penceresini kapat
open(UATBCS)	; UATBCS penceresini aç
end	



Şekil 5.21. Seçim penceresi.

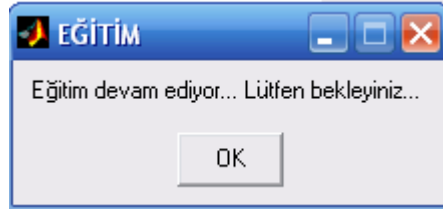
5.3.4. YSA.m Dosyasının Oluşturulması

YSA penceresinde istenilen çıkışı, eğitim ve test sonucunu, başarı oranlarını gösteren 9 adet static text kutusu, öğrenme oranı ve hata oranını gösteren 2 adet edit text kutusu, eğitim, test, kapat ve ana menüye dönüş için 4 adet push buton kullanılmıştır. Şekil 5.22’de YSA arayüz penceresi görülmektedir.



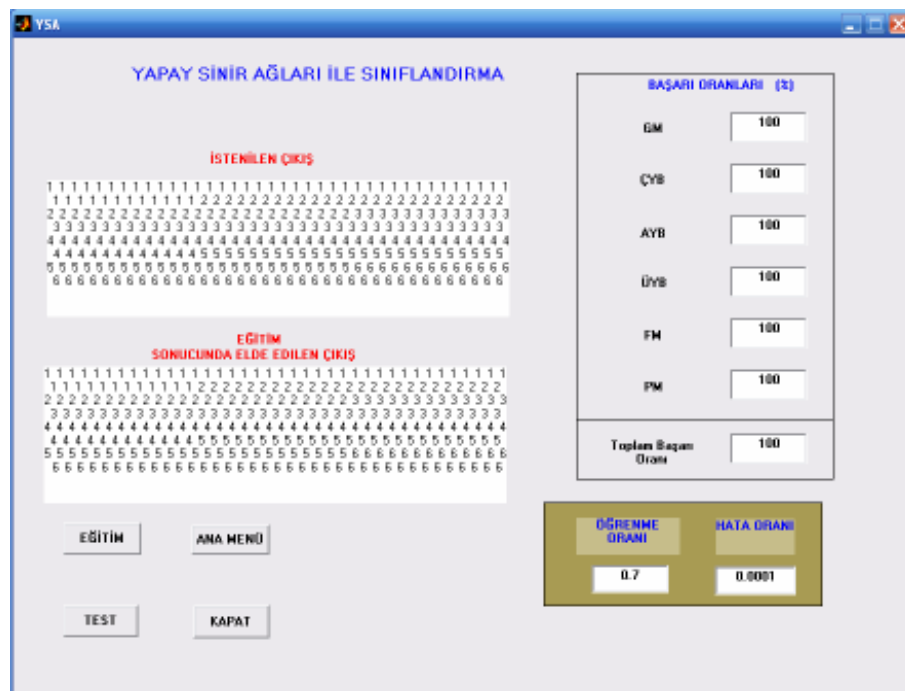
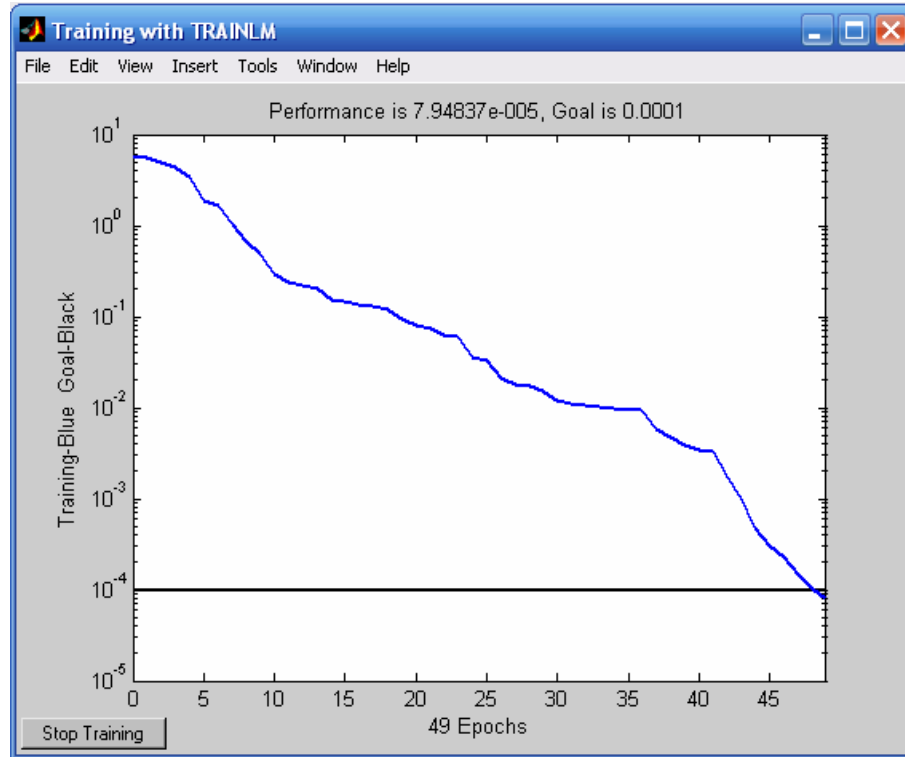
Şekil 5.22. YSA arayüz penceresi.

Yapay sinir ağları ile sınıflandırma yapmak için önce ağ eğitilir ardından test işlemine tabi tutulmaktadır. Eğitim butonuna basıldığında Şekil 5.23'deki gibi eğitim başladığını gösteren mesaj penceresi açılmaktadır.



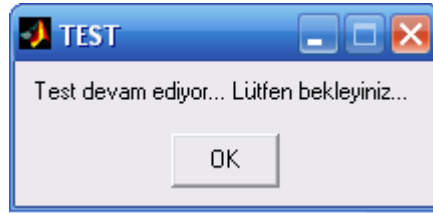
Şekil 5.23. Eğitimin başladığını gösteren mesaj penceresi.

Eğitim işlemi bittiğinde Şekil 5.24'de gösterildiği gibi eğitimin performans eğrisi ekrana gelir. Eğitim sonucunda başarı oranları ve çıkış Şekil 5.25'de gösterilmiştir. Eğitim sonucunda istenilen çıkış ile elde edilen çıkışlar aynı olduğundan tüm başarı oranları %100 olarak bulunmuştur.

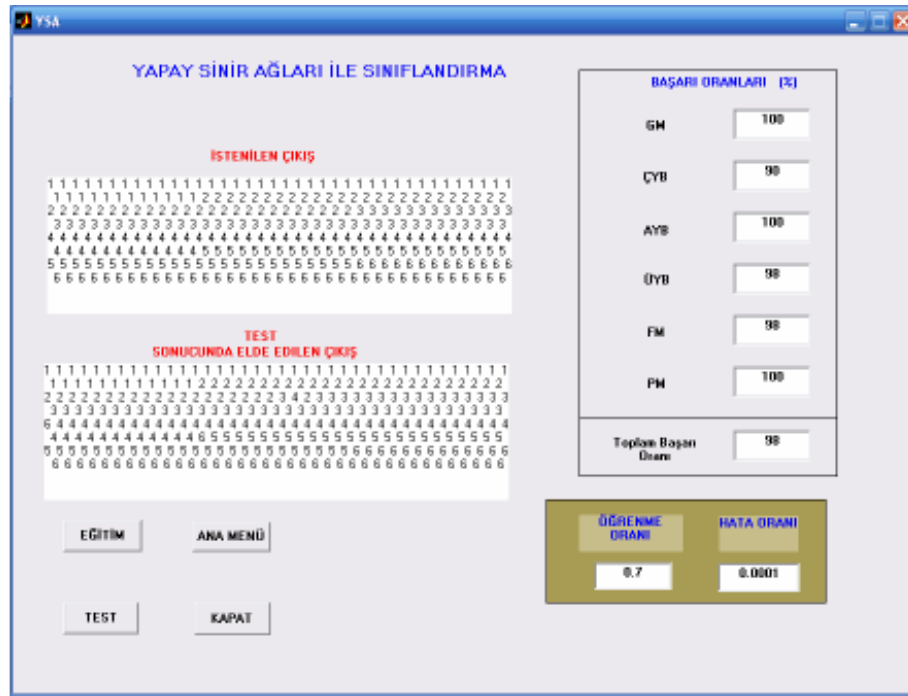


Test butonuna basıldığında Şekil 5.26'daki gibi test işleminin başladığını gösteren mesaj penceresi ekrana gelmektedir. Test işlemi sonunda çıkış ve başarı oranları Şekil 5.27'de gösterilmiştir. Yapay sinir ağlarının eğitiminde öğrenme oranı ($\alpha=0.7$), hata oranı ise 0.0001

olarak seçilmiştir. Bu değerler değiştirildiğinde başarı oranlarında da değişiklik gözlenmektedir. Kullanıcı en uygun değerleri deneyerek bulma şansına sahip olmaktadır.



Şekil 5.26. Test başladığını gösteren mesaj penceresi.



Şekil 5.27. Test İşlemi sonunda YSA penceresinin görünümü.

YSA penceresi m-file dosyasına yazılan komut parçaları aşağıda sıralanmıştır.,

- *Hata ve öğrenme oranı değerlerinin okunması*

Metin kutusuna yazılan ifadeler string özelliğindedir. Sayısal bir değeri metin kutusundan almak için “str2double” ya da “str2num” komutlarından biri seçilir. Alınan değer ondalıklı bir sayı ise str2double, tamsayı ise str2num ifadesi kullanılmaktadır.

ALFA = str2double(get(handles.ALFA,'String')); ; Öğrenme oranını al
HATA = str2double(get(handles.HATA,'String')); ; Hata değerini al

Eğitim ve test işleminde “y” hesaplanan çıktıdır. Bu değeri metin kutusuna yazdırmak için sayısal değeri string değerine dönüştürmek gerekir. Bunun için num2str komutu kullanılır. Eğitim işleminde static text kutusunda “EĞİTİM”, test işleminde ise “TEST” ifadesi yazmaktadır.

Y= num2str(y);	; Çıkışı string ifadesine dönüştür
set(handles.CIKIS,'string',Y);	; Çıkışı metin kutusuna yaz
set(handles.SONUC,'string','EĞİTİM');	; Text kutusuna EĞİTİM yaz
set(handles.SONUC,'string','TEST');	; Text kutusuna TEST yaz

- *Başarı oranlarının hesaplanması*

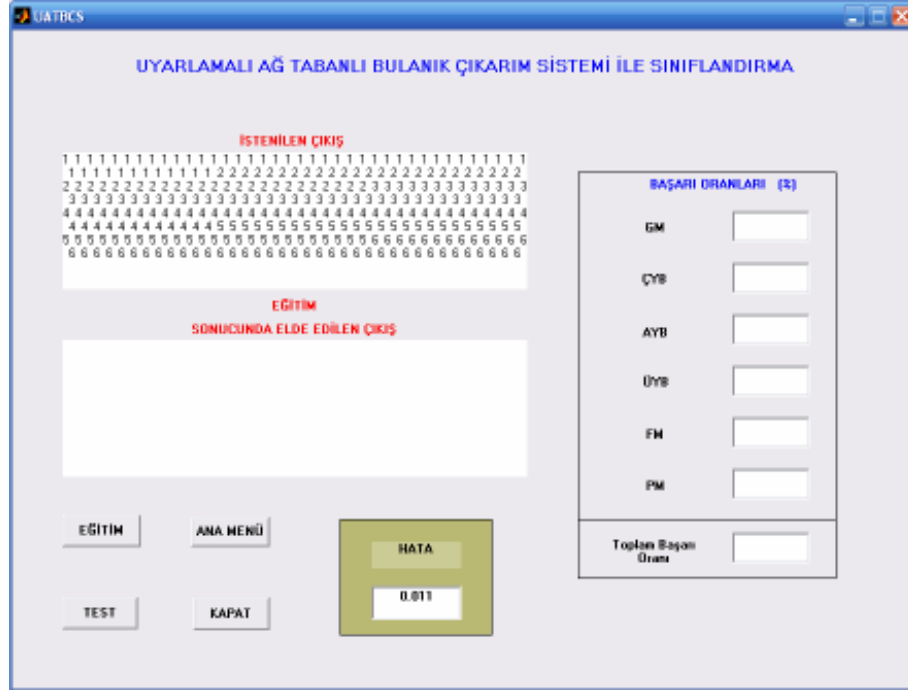
Başarı oranlarını hesaplamak için test işlemi sonunda bulunan çıkış test.m dosyasına yazdırılmaktadır. Modülasyon tanıma işleminde GM, ÇYB, AYB, ÜYB, FM, PM modülleri işaretlerden 50 adet kullanıldığından dosyaya yazdırılan matrisin boyutu 6X50 dir. Çıkış, test.m dosyasına desimal sayı tipi olarak yazdırılmaktadır. Test.m dosyası T değişkenine kaydedilip ardından her satır ilgili sayı ile kontrol edilmektedir. Her satır için bulunan toplam değer yüzdesi alınmaktadır. Birinci satır GM, ikinci satır ÇYB, üçüncü satır AYB, dördüncü satır ÜYB, beşinci satır FM ve altıncı satır PM başarı oranını vermektedir. Her başarı oranı ilgili static text kutusuna yazdırılmaktadır. Toplam başarı oranı için tüm başarı oranları toplanıp ve altıya bölünmektedir.

[illegible]

yanlis1(i)=0;	; yanlis1=0
end	; Döngüden çık
for i=1:6	; 6 satırlık
for j=1:50	; 50 sütunluk döngü aç
if T(j,i)==i	; Çıkış i ye eşitse
toplaml1(i)=toplaml1(i)+1;	; toplam1 değişkenini bir artır
else	; Eşit değilse
yanlis1(i)=yanlis1(i)+1;	; yanlış1 değişkenini bir artır
end	
end	
end	
basari=0;	; basari=0
for i=1:6	
test(i)=toplaml1(i)*100/50;	; Testin başarı oranını hesapla
basari= basari+test(i);	; Toplam başarı oranını hesapla
end	
set(handles.AM,'string',test(1));	; GM başarı oranını text kutusuna yaz
set(handles.DSB,'string',test(2));	; ÇYB başarı oranını text kutusuna yaz
set(handles.LSB,'string',test(3));	; AYB başarı oranını text kutusuna yaz
set(handles.USB,'string',test(4));	; ÜYB başarı oranını text kutusuna yaz
set(handles.FM,'string',test(5));	; FM başarı oranını text kutusuna yaz
set(handles.PM,'string',test(6));	; PM başarı oranını text kutusuna yaz
set(handles.BASARI,'string',round(basari/6));	; Toplam başarı oranını yaz

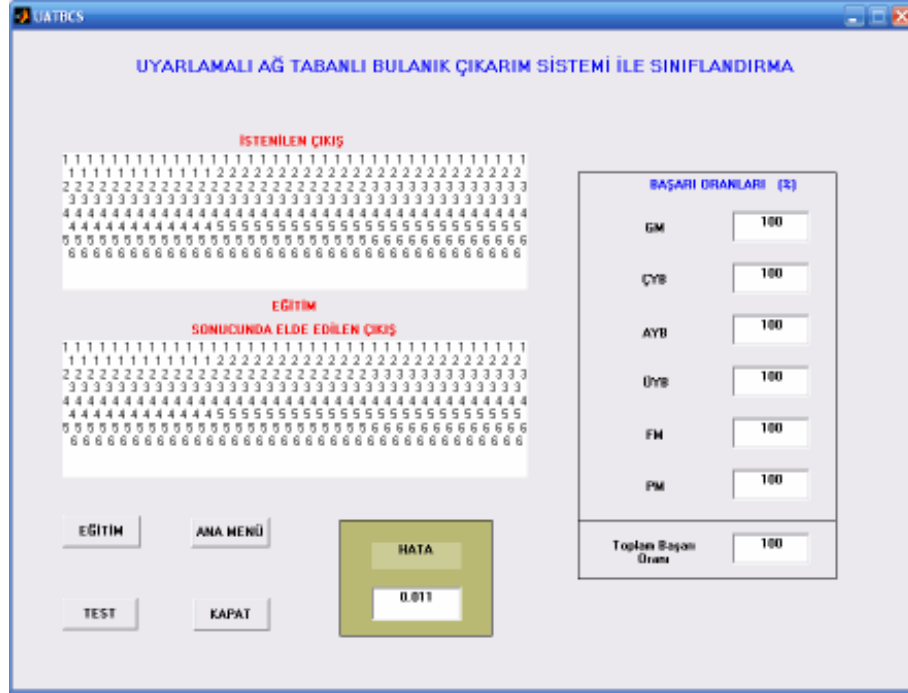
5.3.5. UATBCS.m Dosyasının Oluşturulması

UATBCS penceresinde istenilen çıkışı, eğitim ve test sonucunu, başarı oranlarını gösteren 9 adet static text, hata oranını gösteren 1 adet edit text kutusu, eğitim, test, kapat ve ana menü için 4 adet push buton kullanılmıştır. Şekil 5.28’de UATBCS arayüz penceresi görülmektedir.

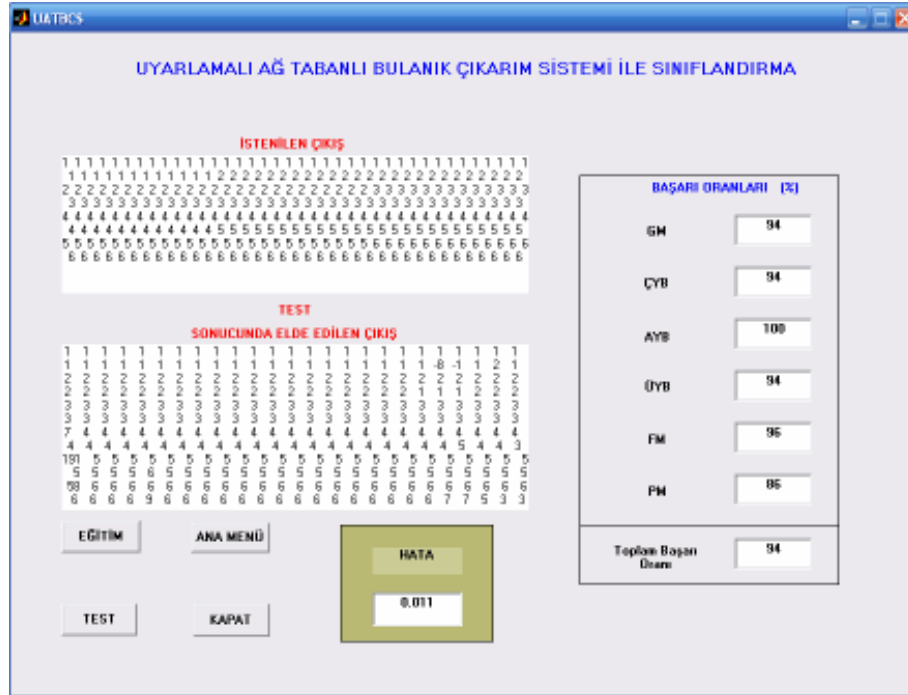


Şekil 5.28. UATBCS arayüz penceresi.

Uyarlamalı ağ tabanlı bulanık çıkarım sistemi ile sınıflandırma yapmak için önce ağ eğitilir ardından test işlemine tabi tutulur. Eğitim sonunda istenilen çıkış ile elde edilen çıkış aynı olduğundan tüm başarı oranları %100 olarak bulunmuştur. Hata oranı varsayılan olarak 0.011 olarak seçilmiştir. Hata oranı azaltıldığında UATBCS sisteminin eğitim süresi de uzamaktadır. Şekil 5.29’da eğitim sonunda, Şekil 5.30’da ise test sonundaki UATBCS arayüz penceresi görülmektedir.



Şekil 5.29. Eğitim sonundaki UATBCS arayüz penceresi.



Şekil 5.30. Test sonundaki UATBCS arayüz penceresi.

6. ANALOG MODÜLASYON UYGULAMALARI

Bu tez çalışmasında, analog modülasyon tanıma için taşıyıcılı akıllı sınıflandırıcı geliştirilmiştir. GM, ÇYB, ÜYB, AYB, FM, PM analog modüleli işaretlerin dalgacık dönüşümü ile özellikleri çıkarılmıştır. Bu özellikler kullanılarak yapay sinir ağları ve uyarlamalı ağ tabanlı bulanık çıkarım sistemi yöntemleri ile de sınıflandırma yapılmıştır. Kullanıcının etkileşimini artırmak amacıyla MATLAB GUI ile arayüz tasarlanmıştır. Kullanılan analog modüleli işaret $c(t)$ ile gösterilir ve matematiksel ifadesi Denklem (6.1) deki gibidir [22].

$$c(t) = b_c r(t) \cos(2\pi f_c t + \psi(t) + \theta_0) \quad (6.1)$$

Burada, $r(t)$ işaretin zarfı, $f_c(t)$ taşıyıcı işaret, $\psi(t)$ açı, θ_0 başlangıç açısı, b_c taşıyıcının gücüdür. Analog modüleli işaretlerin matematiksel gösterimi aşağıda sırayla verilmiştir [22].

- **GM Modülasyonu:**

$$c(t) = [1 + m.f(t). \cos(2\pi f_c t)] \quad (6.2)$$

- **ÇYB Modülasyonu:**

$$c(t) = f(t). \cos(2\pi f_c t) \quad (6.3)$$

- **ÜYB Modülasyonu:**

$$c(t) = f(t). \cos(2\pi f_c t) - h(t) \sin(2\pi f_c t) \quad (6.4)$$

- **AYB Modülasyonu:**

$$c(t) = f(t). \cos(2\pi f_c t) + h(t) \sin(2\pi f_c t) \quad (6.5)$$

- **FM Modülasyonu:**

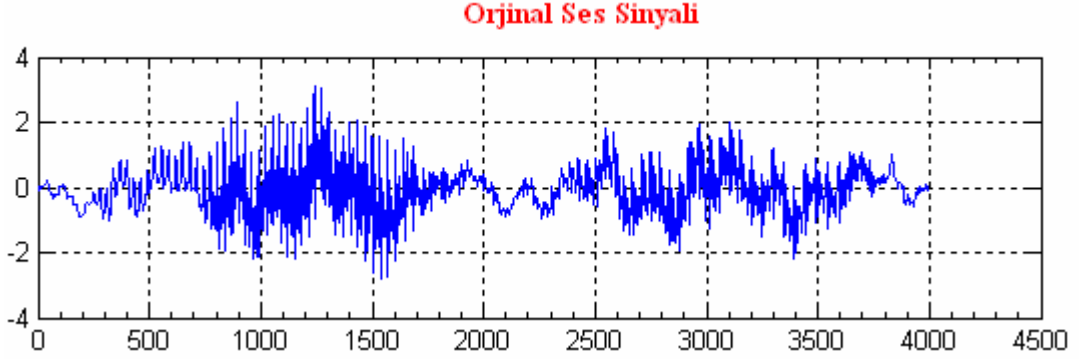
$$c(t) = \cos \left[2\pi f_c t + K_f \int_{-\infty}^t x(\tau) d\tau \right] \quad (6.6)$$

- **PM Modülasyonu:**

$$c(t) = \cos(2\pi f_c t + 2\pi \psi(t) + \theta_0) \quad (6.7)$$

Burada m modülasyon indeksi, $f(t)$ bilgi işareti, $f_c(t)$ taşıyıcı işaret, $h(t)$ Hilbert dönüşümü, K_f FM işaret için frekans sapma katsayısıdır.

Bu uygulama çalışmasında bilgi işareti olarak gerçek ses işareti kullanılmıştır. Kullanılan ses işareti Şekil 6.1’de gösterilmiştir.



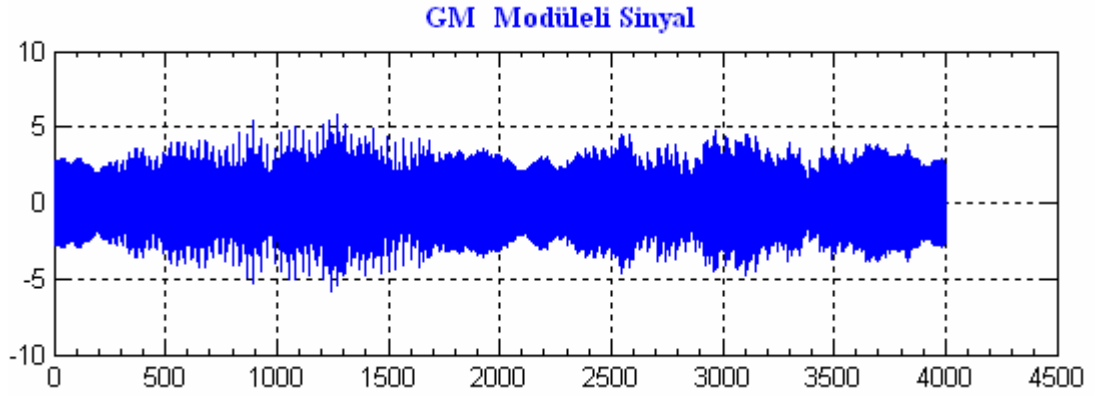
Şekil 6.1. Modülasyon tanıma için kullanılan ses işareti.

6.1. Analog Modüleli İşaretlerin Oluşturulması

Analog modüleli işaretler elde etmek için Şekil 6.1’de gösterilen ses işaretine 0 – 70 dB İ/G aralığında Gaussian beyaz gürültü (Additive White Gaussian Noise – AWGN) ilave edilerek her bir modülasyon çeşidinden 10 adet işaret oluşturulmuştur. İşaretlerin başlangıç açısı, modülasyon indeksi, faz sapması ve frekans sapması değiştirilerek analog modüleli işaretler oluşturulmuştur. Her modülasyon tipinde 100 adet, toplam 600 adet işaret elde edilmiştir. Bu işaretlerin yarısı sınıflandırma eğitimi için diğer yarısı da sınıflandırma testi için kullanılmıştır.

- **GM İşaretlerinin Oluşturulması:**

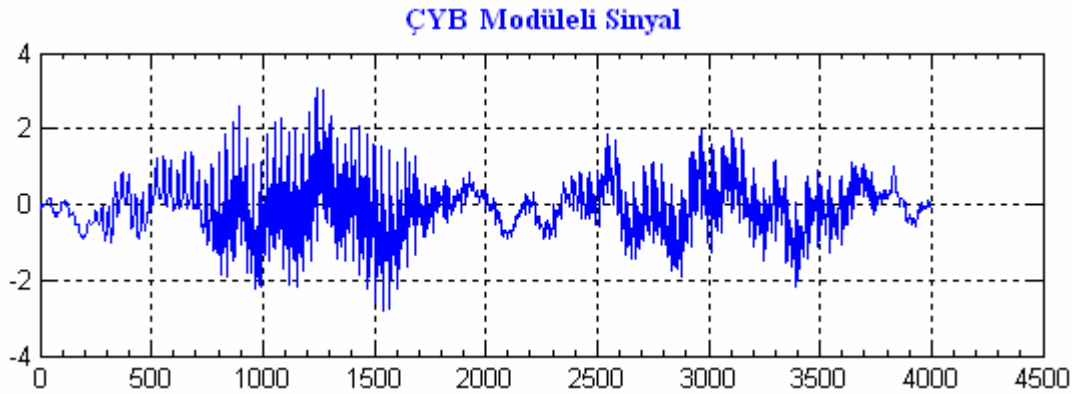
Gürültü ilave edilerek oluşturulan 10 adet işaretin başlangıç açısı (θ_0), 0, $\pi/2$, $\pi/4$ ve modülasyon indeksi (m), 0.3, 0.5 ve 1 olarak değiştirilerek toplam 100 adet GM modüleli işaret elde edilmiştir. Bir GM modüleli işaret örneği Şekil 6.2’de gösterilmektedir.



Şekil 6.2. $m=1$ ve $\phi = \pi/4$ için bir GM modüleli işaret örneği.

- **ÇYB İşaretlerinin Oluşturulması:**

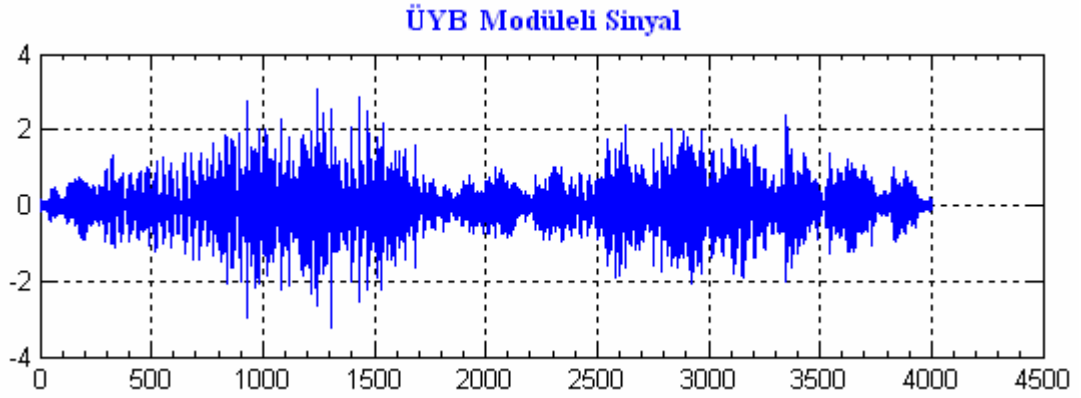
Gürültü eklenerek oluşturulan 10 adet ÇYB işaretinin her birinin başlangıç açısı (θ_0), 0, $\pi/2$, $\pi/3$, $\pi/4$, $\pi/5$, $\pi/6$, $\pi/7$, $\pi/8$, $\pi/9$, $\pi/10$ şeklinde değiştirilerek toplam 100 adet ÇYB işareti elde edilmiştir. Bir ÇYB işaret örneği Şekil 6.3’de gösterilmektedir.



Şekil 6.3. $\phi = \pi/3$ için bir ÇYB modüleli işaret örneği.

- **ÜYB İşaretlerinin Oluşturulması:**

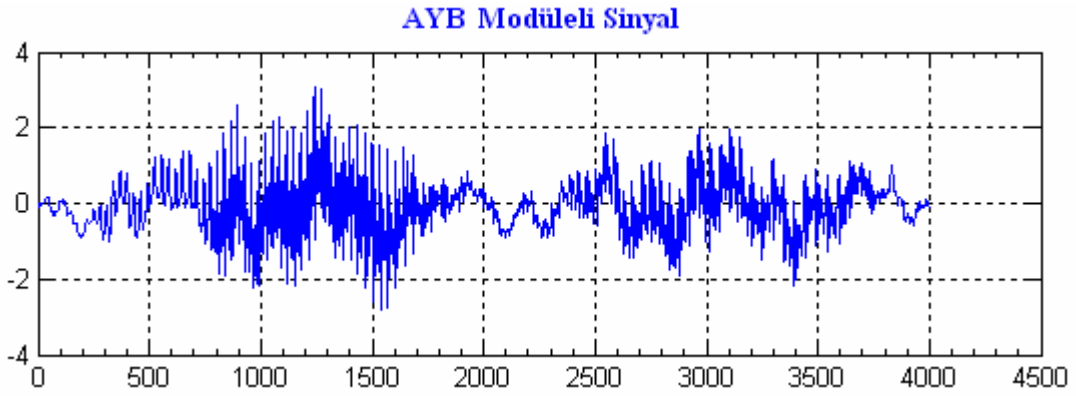
Gürültü eklenerek oluşturulan 10 adet ÜYB işaretinin her birinin başlangıç açısı (θ_0), 0, $\pi/2$, $\pi/3$, $\pi/4$, $\pi/5$, $\pi/6$, $\pi/7$, $\pi/8$, $\pi/9$, $\pi/10$ şeklinde değiştirilerek toplam 100 adet ÜYB işareti elde edilmiştir. Bir ÜYB işaret örneği Şekil 6.4’de gösterilmektedir.



Şekil 6.4. $\square = \pi/10$ için bir ÜYB modüleli işaret örneği.

- **AYB İşaretlerinin Oluşturulması:**

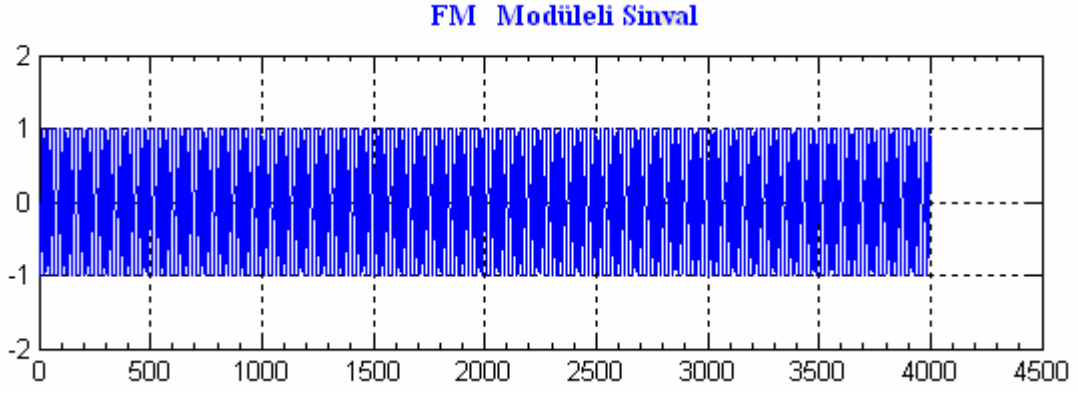
Gürültü eklenerek oluşturulan 10 adet AYB işaretinin her birinin başlangıç açısı (θ_0), $0, \pi/2, \pi/3, \pi/4, \pi/5, \pi/6, \pi/7, \pi/8, \pi/9, \pi/10$ şeklinde değiştirilerek toplam 100 adet AYB işareti elde edilmiştir. Bir AYB işaret örneği Şekil 6.5’de gösterilmektedir.



Şekil 6.5. $\square = \pi/10$ için bir AYB modüleli işaret örneği.

- **FM İşaretlerinin Oluşturulması:**

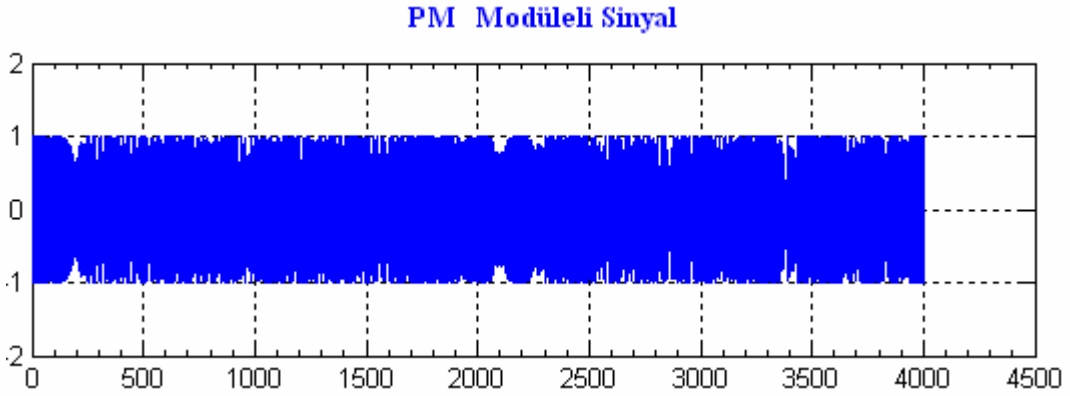
Gürültü eklenerek oluşturulan 10 adet FM işaretinin her birinin başlangıç açısı (θ_0), $0, \pi/2, \pi/4$ ve frekans sapma katsayısı (K_f) 2, 7 ve 9 şeklinde değiştirilerek toplam 100 adet FM işareti elde edilmiştir. Bir FM işaret örneği Şekil 6.6’da gösterilmektedir.



Şekil 6.6. $\Delta f = \pi/4$ frekans sapması = 9 için bir FM modüledi işaret örneği.

- **PM İşaretlerinin Oluşturulması:**

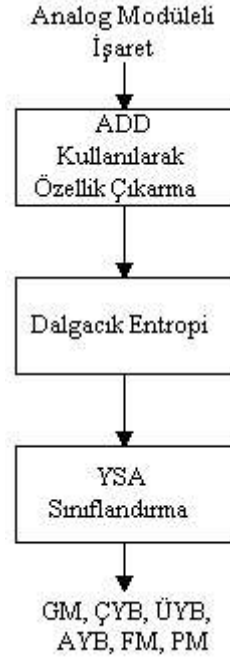
Gürültü eklenerek oluşturulan 10 adet PM işaretinin her birinin başlangıç açısı (θ_0), 0, $\pi/2$, $\pi/4$ ve frekans sapma katsayısı 2, 3, 4 şeklinde değiştirilerek toplam 100 adet PM işareti elde edilmiştir. Bir PM işaret örneği Şekil 6.7’de gösterilmektedir.



Şekil 6.7. $\Delta f = \pi/4$ frekans sapması = 4 için bir PM modüledi işaret örneği.

6.2. Özellik Çıkarma

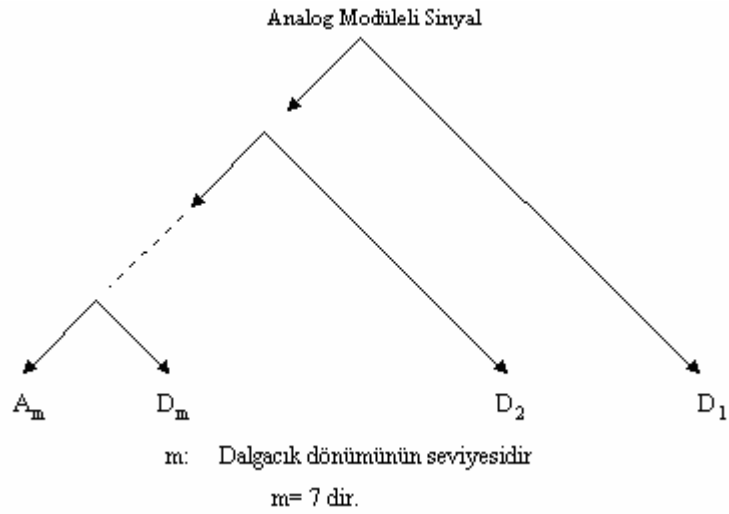
YSA ve Dalgacık entropi kullanılarak analog modülasyon tanımının genel yapısı Şekil 6.8’de gösterilmektedir. Bu çalışmada, analog modülasyonda özellik çıkarma için ayrık dalgacık dönüşümü kullanılmıştır. Elde edilen her bir modüledi işaret dalgacık dönüşümünden geçirilerek özellikleri çıkarılmıştır.



Şekil 6.8. YSA ve Dalgacık Entropi ile Analog Modülasyon Tanımının Yapısı.

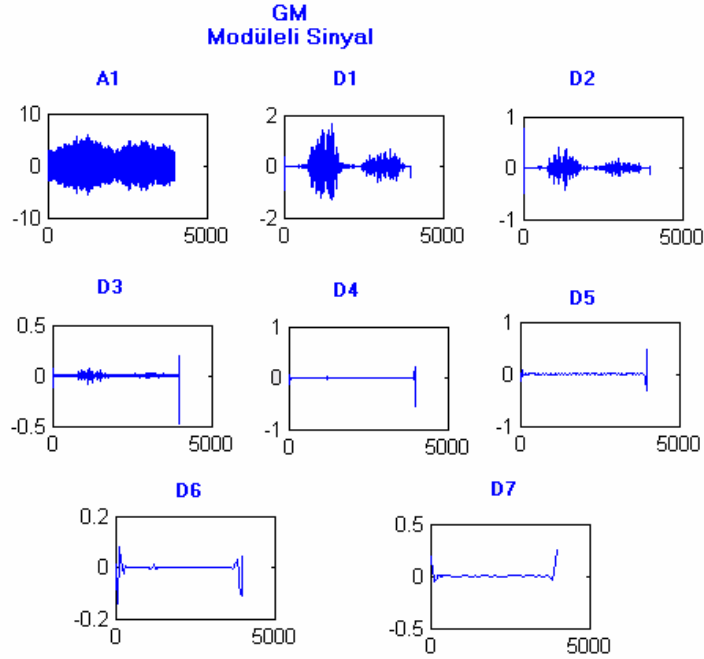
Özellik çıkarmada dalgacık dönüşümü yöntemi kullanılarak, diğer klasik modülasyon tanıma yöntemlerinden farklı olarak bu çalışmada işaretleri karakterize eden en küçük ayrıntılar bile elde edilmektedir. Şekil 6.9’da gösterildiği gibi dalgacık dönüşümü 7 seviyede yapılmaktadır.

Dalgacık dönüşümünde detay (D) ve yaklaşık (A) olmak üzere iki çeşit işaret elde edilmektedir. GM, ÇYB, ÜYB, AYB, FM ve PM işaretlerinin her birinden 7 detay (D) , bir tane de yaklaşık (A) işareti elde edilmiştir.

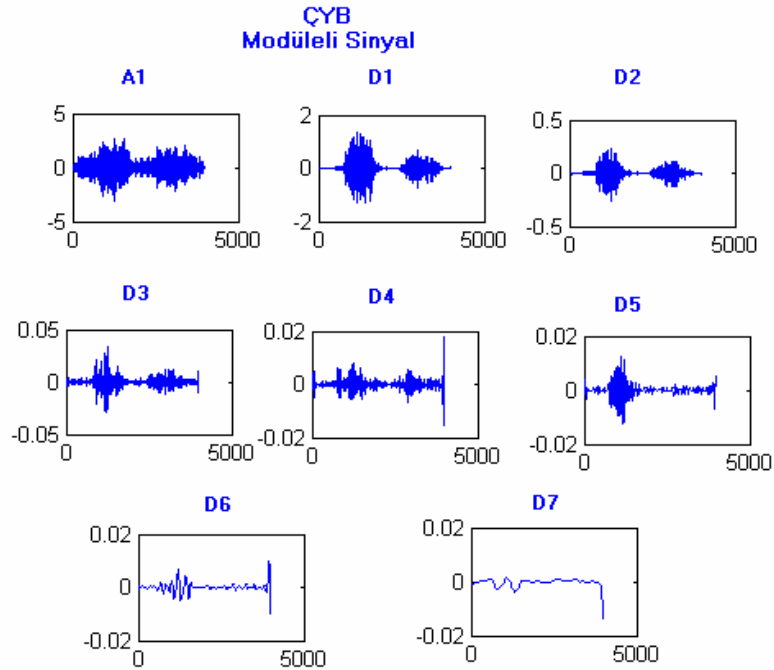


Şekil 6.9. 7 seviyeli dalgacık dönüşümünün yapısı.

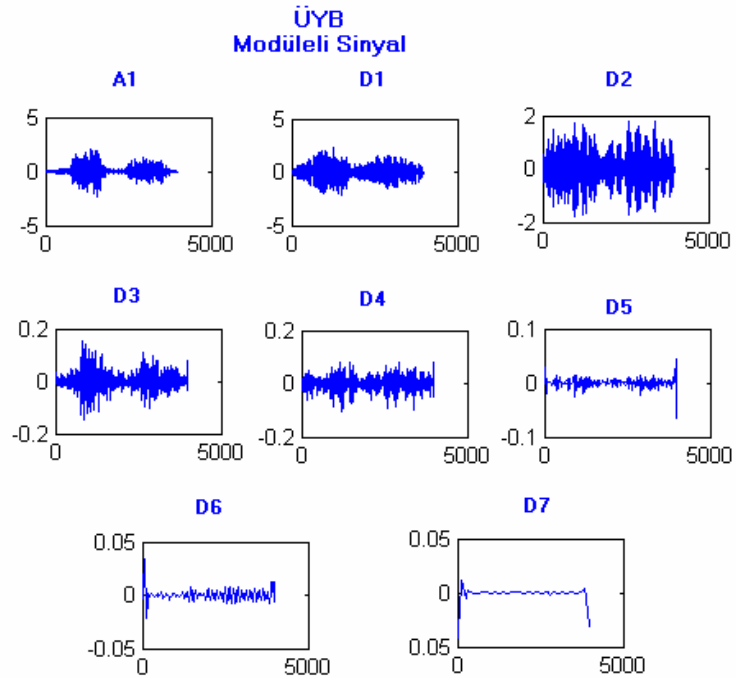
Dalgacık dönüşümünde Haar, Daubechies, Biorthogonal, Coiflets, Symlets filtreleri kullanılmıştır. Bu çalışmada Daubechies ailesinden db7 filtresi kullanılmıştır. Şekil 6.10 – Şekil 6.15’de sırasıyla GM, ÇYB, ÜYB, AYB, FM, PM modülasyonları dalgacık dönüşümü uygulanarak elde edilen dalgacık dönüşümü katsayı örnek işaretleri görülmektedir.



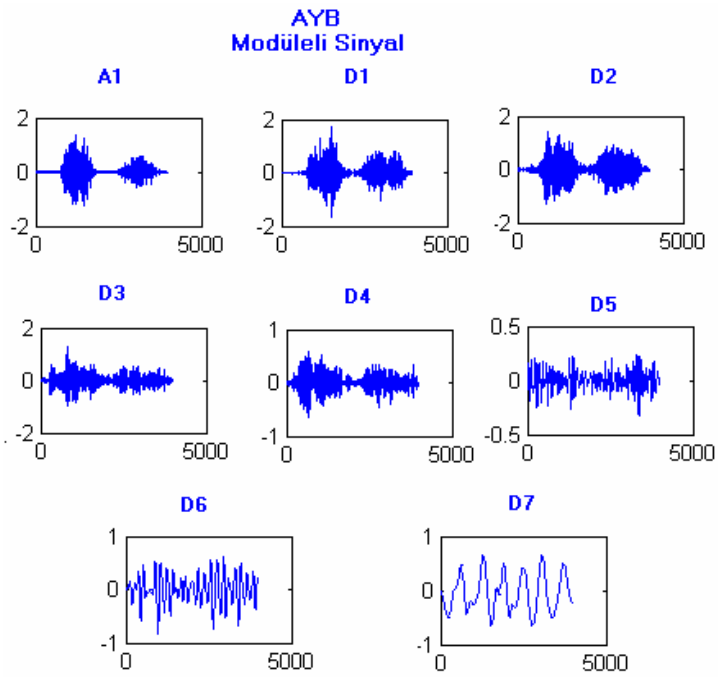
Şekil 6.10. GM modüledi işaretin dalgacık dönüşümü katsayı işaretleri.



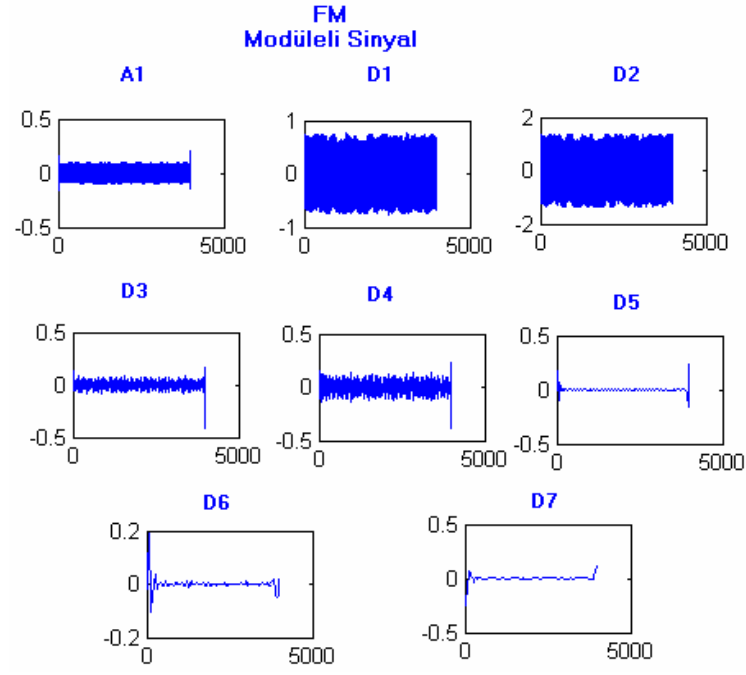
Şekil 6.11. ÇYB modüledi işaretin dalgacık dönüşümü katsayı işaretleri.



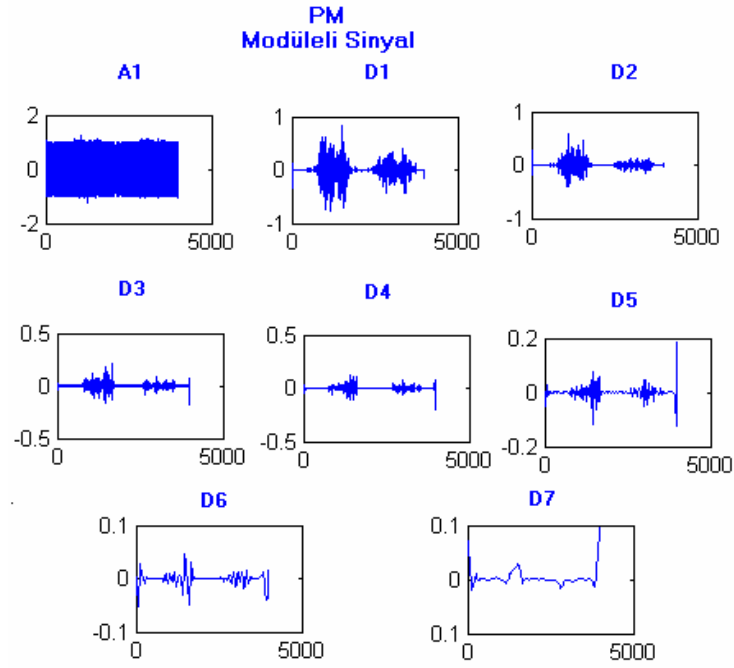
Şekil 6.12. ÜYB modüleli işaretin dalgacık dönüşümü katsayı işaretleri.



Şekil 6.13. AYB modüleli işaretin dalgacık dönüşümü katsayı işaretleri.



Şekil 6.14. FM modüleli işaretin dalgacık dönüşümü katsayı işaretleri.



Şekil 6.15. PM modüleli işaretin dalgacık dönüşümü katsayı işaretleri.

Dalgacık dönüşümü sonucunda elde edilen işaretler entropi işleminden geçirilmiştir. Entropi, işaretteki düzensizliğin ölçütüdür. Kaynağı bilinmeyen modüleli işaretlerin özelliklerinin çıkarılması için bu yöntem kullanılmıştır. Entropi birçok alanda uygulanmaktadır fakat en fazla işaret işlemede kullanılmaktadır [36-37].

YSA, UATBÇS ve dalgacık dönüşümü kullanılarak analog modülasyon tanıma çalışmasında p parametresi 1 olarak alınmıştır. Entropi olarak norm entropi kullanılmış ve entropi işlemi sonunda işaretlerin özellik vektörü çıkarılmıştır. Bu vektör her bir modülasyon tipi için 50x6x8 boyutundadır. Dalgacık dönüşümü 7 seviyede alındığından 7 adet D, 1 adet A işareti olmak üzere 8 adet işaret oluşturulmuş ve bu işaretlerin her birinin Norm entropi değerleri hesaplanarak, her bir işaret için özellik çıkarılmıştır. 6 (modülasyon çeşidi) x 50 (her bir modülasyon çeşidi için kullanılan örnek sayısı) x 8 (her bir modüleli işaretin özellik sayısı) matrisi eğitim aşamasında ve diğer 300 x 8 lik matris de test aşamasında kullanılmıştır.

6.3. Sınıflandırma

Bu çalışmada, sınıflandırma için yapay sinir ağları ve uyarlamalı ağ tabanlı bulanık çıkarım sistemi yöntemleri kullanılmıştır.

6.3.1. Yapay Sinir Ağ Sınıflandırıcısı

Yapay sinir ağı olarak geriye yayılım algoritması kullanılmıştır. Bu ağ yapısı 3 katmandan oluşmaktadır. Giriş katmanı, orta katman (gizli katman) ve çıkış katmanıdır.

YSA ağ yapısında;

- Giriş katmanı, 8 giriş hücresinden,
- Orta katman, 15 orta katman hücresinden,
- Çıkış katmanı, 1 çıkış hücresinden oluşmaktadır.

YSA yapısında orta katman sayısı, giriş ve çıkış katman sayısının toplamından fazla olmalıdır [38]. Giriş ve orta katmanda hiperbolik tanjant sigmoid aktivasyon fonksiyonu, çıkış katmanında ise doğrusal aktivasyon fonksiyonu kullanılmıştır.

YSA eğitiminde analog modüleli işaretlerden 300 tanesi kullanılmıştır. Modülasyon tiplerinden GM 1, ÇYB 2, AYB 3, ÜYB 4, FM 5, ve PM 6 olarak numaralandırılmıştır. Yapay sinir ağları arayüz penceresinde istenilen çıkışlar Şekil 6.16'da gösterilmiştir.

İSTENİLEN ÇIKIŞ																													
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Şekil 6.16. YSA sınıflandırıcı için arzu edilen çıkış matrisi.

YSA’da eğitim tur sayısı 1500 olarak alınmıştır. Eğitimde hata oranı ve öğrenme oranı değerleri Şekil 6.17’de gösterildiği gibi kullanıcı tarafından arayüz penceresinde değiştirilmektedir. YSA eğitiminde elde edilen çıkış Şekil 6.18’de görülmektedir.

ÖĞRENME ORANI	HATA ORANI
0.7	0.0001

Şekil 6.17. Öğrenme ve hata oranlarının değiştirildiği pencere.

EĞİTİM SONUCUNDA ELDE EDİLEN ÇIKIŞ																													
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

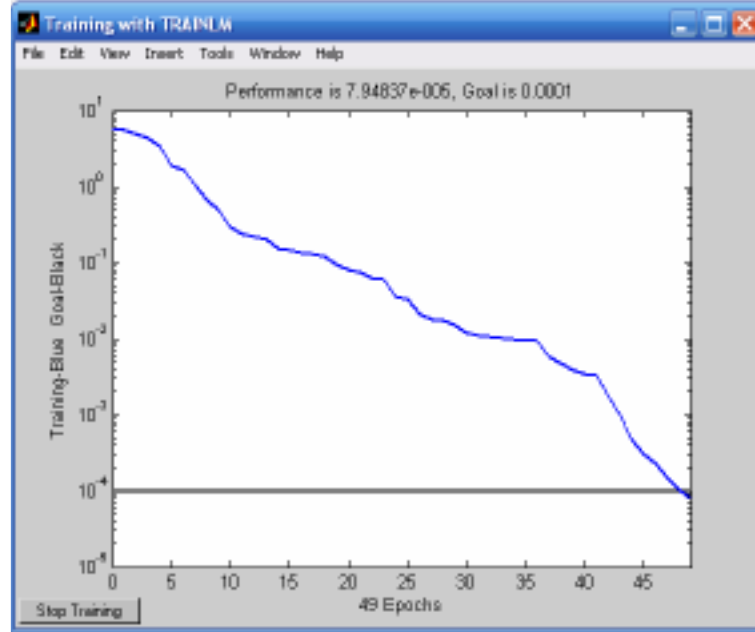
Şekil 6.18. YSA eğitimi sonucunda elde edilen çıkış matrisi.

YSA test işleminde analog modüleli işaretlerden kalan 300 tanesi kullanılmıştır. YSA test işleminde elde edilen çıkış Şekil 6.19’da gösterilmiştir.

TEST SONUCUNDA ELDE EDİLEN ÇIKIŞ																													
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
6	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6

Şekil 6.19. YSA test işlemi sonucunda elde edilen çıkış matrisi.

Bu çalışmada 49 tur sonunda (ortalama karesel hata) $OKH \leq 10^{-4}$ olduğu için eğitim sona ermiştir. Eğitim performansı Şekil 6.20’de görülmektedir.

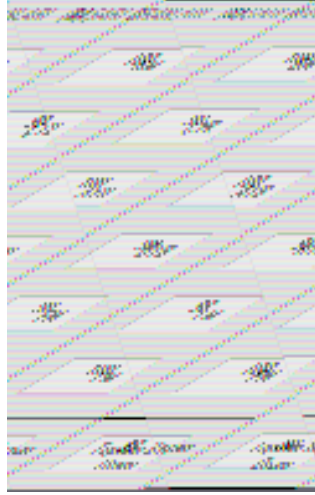


Şekil 6.20. YSA sınıflandırıcının eğitim performansı.

Analog modüleli işaretler GM, ÇYB, ÜYB, AYB, FM ve PM olmak üzere 6 farklı işaret deneysel olarak incelenmiştir. Her bir işaretten 100 adet örnek alınmış ve toplam 600 adet işaret deneyde kullanılmıştır. Bu işaretlerden 300 adet eğitimde, 300 adet ise test aşamasında kullanılmıştır. Eğitim için kullanılan 300 adet işaret test için kullanıldığında başarı oranı %100 olarak bulunmuştur ve Şekil 6.21’de gösterilmiştir. Test için ayrılan işaretler kullanıldığında Şekil 6.22. de gösterildiği gibi ortalama başarı oranı %98 bulunmuştur.



Şekil 6.21. YSA eğitimi sonucunda modüleli işaretlerin başarı oranları.



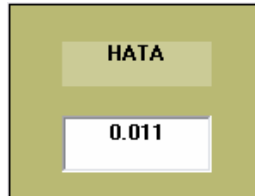
Şekil 6.22. YSA test işlemi sonucunda modüleli işaretlerin başarı oranları.

6.3.2. Uyarlamalı Ağ Tabanlı Bulanık Çıkarım Sistemi Sınıflandırıcısı

Uyarlamalı ağ tabanlı bulanık çıkarım sisteminde hem yapay sinir ağları hem de bulanık mantık kullanılır [48]. Bulanık mantıkta kullanılan eğer – ise kuralları ağın giriş ve çıkış bilgisini oluşturur. Ağın eğitimi ve testi ise yapay sinir ağları ile yapılmaktadır.

UATBÇS sınıflandırıcıda giriş üyelik dereceleri hesaplanırken Şekil 4.9’da gösterilen gauss üyelik fonksiyonu kullanılmıştır. Bulanık mantık yönteminde Sugeno ve Mamdani olmak üzere temelde iki çeşit bulanık denetleyici kullanılmaktadır [46]. Mamdani bulanık denetleyicide çıkışlar kümelerle, Sugeno bulanık denetleyicide ise çıkışlar fonksiyonlarla ifade edilmektedir. Bu çalışmada Sugeno tipi bulanık denetleyici kullanılmıştır.

UATBÇS’de eğitim tur sayısı 300 olarak alınmıştır. Eğitimde hata oranı Şekil 6.23’de gösterildiği gibi kullanıcı tarafından arayüz penceresinde değiştirilebilmektedir.



Şekil 6.23. Hata oranının değiştirildiği pencere.

Yapay sinir ağı olarak en küçük kareler yöntemi ve geriye yayılım algoritmasının birleşiminden oluşan hibrid öğrenme algoritması kullanılmaktadır. UATBÇS yapısında her bir giriş için 2 adet üyelik fonksiyonu tanımlanmıştır.

UATBÇS eğitiminde analog modüleli işaretlerden 300 tanesi kullanılmıştır. Modülasyon tiplerinden GM 1, ÇYB 2, AYB 3, ÜYB 4, FM 5, ve PM 6 olarak numaralandırılmıştır. UATBÇS arayüz penceresinde eğitim sonucunda elde edilen çıkışlar Şekil 6.24’de gösterilmiştir.

EĞİTİM																													
SONUCUNDA ELDE EDİLEN ÇIKIŞ																													
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
4	4	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	

Şekil 6.24. UATBÇS eğitim aşamasında elde edilen çıkış matrisi.

UATBÇS test işleminde analog modüleli işaretlerden 300 tanesi kullanılmıştır. UATBÇS test işleminde elde edilen çıkış Şekil 6.25’de gösterilmiştir.

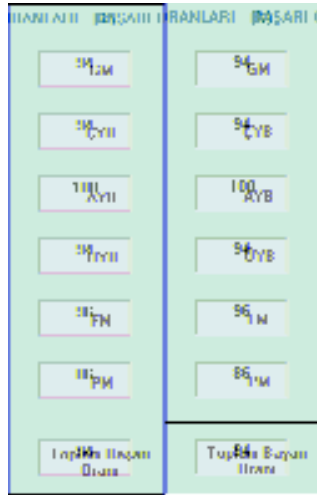
TEST																													
SONUCUNDA ELDE EDİLEN ÇIKIŞ																													
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-8	-1	1	1	2	1			
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	2	2	2	2	
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
7	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
191	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
	5	5	5	5	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
58	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
	6	6	6	6	9	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	

Şekil 6.25. UATBÇS test aşamasında elde edilen çıkış matrisi.

UATBÇS ile sınıflandırmada GM, ÇYB, ÜYB, AYB, FM ve PM olmak üzere 6 farklı işaret deneysel olarak incelenmiştir. Her bir işaretten 100 adet örnek alınmış ve toplam 600 adet işaret deneyde kullanılmıştır. Bu işaretlerden 300 adet eğitimde, 300 adet ise test aşamasında kullanılmıştır. Eğitim için kullanılan 300 adet işaret test için kullanıldığında başarı oranı %100 olarak bulunmuştur ve Şekil 6.26’da gösterilmiştir. Test için ayrılan işaretler kullanıldığında Şekil 6.27’de gösterildiği gibi ortalama başarı oranı %94 olarak bulunmuştur.



Şekil 6.26. UATBÇS eğitimi sonucunda modüleli işaretlerin başarı oranları.



Şekil 6.27. UATBÇS test sonucunda modüleli işaretlerin başarı oranları.

YSA test aşamasında 50 adet GM işaretinden 50 si, 50 adet ÇYB işaretinden 45 i, 50 adet ÜYB işaretinden 49 u, 50 adet AYB işaretinden 50 si, 50 adet FM işaretinden 49 u ve 50 adet PM işaretinden 50 si tanınmıştır.

UATBÇS test aşamasında 50 adet GM işaretinin 47 si, 50 adet ÇYB işaretinin 47 si, 50 adet AYB işaretinin 50 si, 50 adet ÜYB işaretinin 47 si, 50 adet FM işaretinin 48 si, 50 adet PM işaretinin 43 ü tanınmıştır. YSA sınıflandırıcı başarı oranları Tablo 6.1’de, UATBÇS sınıflandırıcı başarı oranları ise Tablo 6.2’de gösterilmiştir.

Tablo 6.1. YSA sınıflandırıcı için başarı oranları.

Modülasyon Tipi	GM	ÇYB	ÜYB	AYB	FM	PM	Toplam Başarı
Başarı Oranları	%100	%90	%98	%100	%98	%100	%98

Tablo 6.2. UATBÇS sınıflandırıcı için başarı oranları.

Modülasyon Tipi	GM	ÇYB	ÜYB	AYB	FM	PM	Toplam Başarı
Başarı Oranları	%94	%94	%94	%100	%96	%86	%94

7. SONUÇ

7.1. Sonuçlar ve Tartışma

Teknolojinin gün geçtikçe gelişmesi ve haberleşme sistemlerinin kullanım alanlarının artmasından dolayı sistemlerinin güvenilir, hızlı ve doğru bir şekilde iletilmesi büyük önem kazanmıştır. Bu amaçla bu tez çalışmasında, haberleşmede en fazla kullanılan analog modülasyon işaretlerinin tanınmasını kolaylaştıracak, tanımanın otomatik olarak gerçekleştirilmesini sağlayacak akıllı bir tanıma sistemi geliştirilmiştir.

Bu tezin temel amacı, analog modülasyon tanıma alanında yapılabilecek gerçek uygulamalara zemin oluşturmaktır. Bu amaçla tez çalışmasında, alınan analog modüleli işaretlerin en iyi özelliklerinin çıkarılması ve bu özelliklerin YSA, UATBÇS sınıflandırma yöntemleri ile sınıflandırılmasını yapan bir arayüz sistemi geliştirilmiştir.

Bu sistem, özellik çıkarma, eğitim ve test olmak üzere 3 aşamadan oluşmaktadır. İlk aşama, birbirinden farklı analog modüleli işaretlerin dalgacık dönüşümü ve dalgacık entropi yöntemi ile en iyi özelliklerinin çıkarılmasına dayanmaktadır. İkinci aşama, elde edilen özelliklerin eğitime ve üçüncü aşama ise özelliklerin test edilmesine bağlıdır. Eğitim ve test aşamalarında kullanıcı arayüz ile sınıflandırıcı çeşidini seçebilmektedir.

Bu tez çalışmasında, kullanılan analog modülasyon tanıma uygulamalarındaki parametrelerin tamamı tasarlanan arayüz ile değiştirilme özelliğine sahiptir. Bu sebeple en iyi sonuç alınıncaya kadar sistemin özellik çıkarma, eğitim ve test işlemleri isteğe bağlı olarak devam edebilmektedir. Analog modülasyon tanıma uygulamalarında elde edilen performans sonuçları Tablo 6.1. ve Tablo 6.2. de gösterilmiştir. Elde edilen bu sonuçlar, bu sistemin otomatik analog modülasyon tanıma alanında kullanılabilecek etkili bir yöntem olduğunu göstermektedir.

7.2. Öneriler

- Bu çalışmada sınırlı sayıda analog modüleli işaret kullanılmıştır. İleride yapılacak çalışmalarda daha geniş bir veri tabanı kullanılarak bu sistemin başarısı denenebilir.
- Günümüz teknolojisinde analog modülasyonların otomatik olarak tanınmasının gerektiği her yerde bu sistem kullanılabilir.
- Bu çalışmada analog modüleli işaretlerin yorumlanması amaçlanmıştır. Aynı yöntemler sayısal modüleli işaretlerin yorumlanması içinde uygulanabilir.

- Bu sistem gerek haberleřme sistemlerinde de sayısal veya analog modlasyon tanıma iin kullanılabilir.
- Bu alıřmada MATLAB GUI ile oluřturulan arayz, analog ve sayısal modlasyon tanımada kullanılan tm yntemler iin geliřtirilebilir.

7.3. Yayınlar

Bu tez alıřmasında gerekleřtirilen uygulamalar iki adet dergi ve bir adet uluslararası konferans yayını olarak sonulandırılmıřtır [43].

8. KAYNAKLAR

- [1] Tomasi,W., 1994, Elektronik İletişim Teknikleri. MEB, Ankara,.
- [2] Guldemir, H, Sengür, A.,2006, Comparison of Clustering Algorithms for Analog Modulation Classification, Expert Systems with Applications, 30, No.4, 642-649.
- [3] Azzouz, E. E., Nandi, A. K., 1997, Automatic Modulation Recognition-I, Journal of Franklin Inst., 334B, No.2, 241-273.
- [4] Avcı, E., 2007, An Expert System Based on Wavelet Neural Network-Adaptive Norm Entropy For Scale Invariant Texture Classification, Elsevier Expert Systems with Applications, 32, No.3, (in Press).
- [5] Guldemir, H, Şengür, A., 2007, Online Modulation Recognition of Analog Communication Signals using Neural Network, Expert Systems with Applications, 33, No.1.
- [6] Wong, M. L. D., Nandi, A. K., 2004, Automatic Dijital Modulation Recognition Using Artificial Neural Network and Genetic Algorithm, Signal Processing, 84, 351-365.
- [7] Hsue, S. Z., Soliman, S. S., 1990, Automatic modulation classification using zero crossing, IEE Proceedings, 137, 459-464.
- [8] Aisbett, J., 1997, Automatic modulation recognition using the time domain parameters, Signal Processing, 13, 323-328.
- [9] Dominguez, L. V., Borello, J. M. P., Garcia, J. P., &Mezaa, B. P., 1991, A general approach to the automatic classification of radio communication signals, Signal Processing, 22, 239-250.
- [10] Jondral, F., 1985, Automatic classification of high frequency signals,Signal Processing, 9, 177-190.
- [11] Azzouz, E.E., Nandi, A.K., 1996, Automatic modulation recognition of communication signals, Kluwer Academic Publishers.
- [12] Ghani, N., Lamontagne, R., 1993, Neural networks applied to the classification of spectral features for automatic modulation recognition, Military Communication Conference, 93, 111-115.

- [13] Fabrizio, P. M., Lopes L. B., Lockhart, G. B., 1986, Receiver recognition of analogue modulation types, in IERE Conference on Radio Receiver and Associated Systems, Bangor Wales, 135-140.
- [14] Chan, Y. T., Gadbois, L. G., 1989, Identification of the modulation type of a signal, Signal Processing, 16, No. 2, 149-154.
- [15] Nagy, P. A. J., 1994, Analysis of a method for classification of analogue modulated radio signals, in European Association for Signal Processing VII Conference 94, Edinburgh, Scotland, 1015-1018.
- [16] Jovanovic, S. D., Doroslovacki, M. I., Dragosevic, M. V., 1994, Recognition of low modulation index AM signals in additive Gaussian noise, in European Association for Signal Processing V Conference, Edinburgh, Scotland, 1923-1926.
- [17] Al-jalili, Y. O., 1995, Identification algorithm for upper sideband and lower sideband SSB signals, Signal Processing, 42, No. 2, 207-213.
- [18] Nandi, A.K., Azzouz, E.E., 1995, Automatic analogue modulation recognition, Signal Processing, 46, No. 2, 211-222.
- [19] Nandi, A. K., Azzouz, E. E., 1997, Modulation recognition using artificial neural networks, Signal Processing, 56, 165-175.
- [20] Güldemir, H., Şengür, A., 2005, An Educational Interface for Automatic Recognition of Analog Modulated Signals, Journal of Applied Sciences, 5, No.3, 513-516.
- [21] Türkoğlu, İ., 2002, Haberleşme Tekniği-I Ders Notları. Fırat Üniversitesi, Elazığ.
- [22] Derin, H., Aşkar, M., 1987, İletişim Kuramı Modülasyon Yöntemleri. ODTÜ Mühendislik Fakültesi, Ankara.
- [23] Yarcı, K., Öztürk, O., 1999, Elektronik Ders Kitabı. Beta Yayınevi, İstanbul.
- [24] Türkoğlu, İ., 2003, Örüntü Tanıma Ders Notları. Fırat Üniversitesi, Elazığ.
- [25] Avcı, E., 2006, İleri Yapay Zeka Ders Notları. Fırat Üniversitesi, Elazığ.

- [26] Türkoğlu, İ., Şengür, A., ve Toraman, S., 2003, Tıbbi Görüntülerden İstenen Bir Örüntünün Ayırıştırılması, DAUM, 3, 97-101.
- [27] Türkoğlu, İ., ve Arslan, A., 1996, Yapay sinir ağları ile bozuk örüntü tanıma, Fırat Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, 8, 147-158.
- [28] Türkoğlu, İ., 2002, Durağan Olmayan İşaretler İçin Zaman-Frekans Entropilerine Dayalı Akıllı Örüntü Tanıma, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.
- [29] Avcı, E., Türkoğlu, İ., ve Poyraz, M., 2004, Darbeli Radar Sistemlerinde Bayes Karar Kuralı İle Hedef Sınıflama Uygulaması, DAUM, 2, No.2, 25-29.
- [30] Avcı, E., Türkoğlu, İ., ve Poyraz, M., 2005, Darbeli Radar Sistemlerinde Geliştirilen Periodogram Çıkarımlı Akıllı Hedef Tanıma, Gazi Üniversitesi Fen Bilimleri Dergisi, 18, No.2, 259-272.
- [31] Hanbay, D., Türkoğlu, İ., ve Demir, Y., 2007, Predicting Chemical Oxygen Demand (COD) Based on Wavelet Decomposition and Neural Networks, Wiley-VCH: CLEAN - Soil Air Water, 35, No.3, 250-254.
- [32] Şengür, A., ve Türkoğlu, İ., 2004, Değişmez Momentlerle Türkçe Karakter Tanıma, DAUM, 2, No.2, 115-118.
- [33] Türkoğlu, İ., 1996, Yapay Sinir Ağları ile Nesne Tanıma, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü.
- [34] Türkoğlu, İ., Arslan, A. ve İlkay, E., 2003, An Intelligent System for Diagnosis of Heart Valve Diseases with Wavelet Packet Neural Networks, Computer in Biology and Medicine, 33, No.4, 319-331.
- [35] Avcı, E., Türkoğlu, İ., ve Poyraz, M., 2005, Intelligent Target Recognition Based on Wavelet Packet Neural Network, Expert Systems with Applications, 29, No.1, 175-182.
- [36] Türkoğlu, İ., 2006, An Intelligent Pattern Recognition System Based on Neural Networks and Wavelet Decomposition For Interpretation of Lung Sounds, e-journal of New World Sciences Academy, 1, No.1, 23-31.

- [37] Avcı, E., Türkoğlu, İ., Poyraz, M., 2005, Intelligent Target Recognition Based on Wavelet Adaptive Network Based Fuzzy Inference System, Lecture Notes in Computer Science, Springer-Verlag , 3522, 594-601.
- [38] Şengür, A., Türkoğlu, İ., ve İnce, M.C., 2007, Wavelet Packet Neural Networks For Texture Classification, Expert systems with applications, 32, No.2, 527-533.
- [39] Hanbay, D., Türkoğlu, İ., ve Demir, Y., 2008, Prediction of Wastewater Treatment Plant Performance Based on Wavelet Packet Decomposition and Neural Networks, Expert systems with applications, 34, No.4.
- [40] Gökbulut, M., 2000, Yapay Sinir Ağları Ders Notları, Fırat Üniversitesi, Elazığ.
- [41] Gökbulut, M., Akpolat, Z.H., Güldemir, H., 2000, Emulation of nonlinear mechanical loads using multi-layer neural networks”, International Journal for Engineering Modelling, 13, No.3-4, 69-75.
- [42] Avcı, E., 2007, Performance comparison of wavelet families for analog modulation classification using expert discrete wavelet neural network system, Elsevier Expert Systems with Applications, 33, No.1, (in Press).
- [43] Avcı, E., Bicer, Z. Erdem Yakut, S., 2006, An Adaptive Wavelet Entropy Method for Intelligent Digital Modulation Classification, Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks (TAINN 2006), MUĞLA, 117-124, No.2.
- [44] Kosko, B., 1991, Neural Networks and Fuzzy Systems, A Dynamical Systems Approach.
- [45] Jang, J. S. R., Sun, C. T., 1995, Neuro-Fuzzy Modeling and Control, proceedings of the IEEE, 3, No. 3.
- [46] Aydın, H., 2000, Bulanık Mantık Uygulamaları Ders Notları, Fırat Üniversitesi, Elazığ.
- [47] Özçalık, H., Uygur, A., 2003, Dinamik Sistemlerin Uyumlu Sinirsel-Bulanık Ağ Yapısına Dayalı Etkin Modellenmesi, KSU J. Science and Engineering, 6, No.1.
- [48] Avcı, E., Türkoğlu, İ., 2003, Modelling of Tunnel Diode by Adaptive-Network-Based Fuzzy Inference System, International Journal of Computational Intelligence, 1, No.1, 231-233.

- [49] Uysal, M., 2004, Matlab ile matematiksel uygulamalar ve mühendislik uygulamaları, İstanbul.
- [50] Özsarı, F., 2007, Matlab Gui İle Kullanıcı Arayüzü Tasarlama, www.teknikogretmenler.net.

ÖZGEÇMİŞ

Sultan ERDEM YAKUT

s_erdem23@yahoo.com

Fırat Üniversitesi Teknik Eğitim Fakültesi

Elektronik Bilgisayar Eğitimi Bölümü

ELAZIĞ

1979 yılında Elazığ'da doğdu. İlk, orta ve lise öğrenimini Elazığ'da tamamladıktan sonra 1996 yılında Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik Bilgisayar Eğitimi Elektronik Öğretmenliği Bölümü'nü kazandı. 2000 yılında bu bölümden mezun oldu. Aynı yıl İstanbul Bahçelievler Erkan Avcı Endüstri Meslek Lisesi'nde Elektronik Öğretmeni olarak göreve başladı. 2005 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar Eğitimi Ana Bilim Dalı Telekomünikasyon Bilim Dalı'nda yüksek lisans eğitimine başladı. Halen Elektronik Öğretmeni olarak görevini sürdürmektedir. Yabancı dili İngilizce'dir.