

**GERÇEK ZAMANLI DURUM İZLEME VE ARIZA
TEŞHİSİ İÇİN BAĞIŞIK AKILLI HESAPLAMA
TEKNİKLERİNİN GELİŞTİRİLMESİ**

Yük. Müh. İlhan AYDIN

**Doktora Tezi
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Erhan AKIN**

AĞUSTOS-2011

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GERÇEK ZAMANLI DURUM İZLEME VE ARIZA TEŞHİSİ İÇİN
BAĞIŞIK AKILLI HESAPLAMA TEKNİKLERİNİN GELİŞTİRİLMESİ

DOKTORA TEZİ

Yük. Müh. İlhan AYDIN

(06113208)

Tezin Enstitüye Verildiği Tarih : 23 Ağustos 2011

Tezin Savunulduğu Tarih : 20 Eylül 2011

Tez Danışmanı : Prof. Dr. Erhan AKIN (F.Ü)

Diğer Jüri Üyeleri : Prof. Dr. Ahmet ARSLAN (S.Ü)

Prof. Dr. Hasan KÜRÜM (F.Ü)

Doç. Dr. Yetkin TATAR (F.Ü)

Yrd. Doç. Dr. Mehmet KARAKÖSE (F.Ü)

AĞUSTOS-2011

ÖNSÖZ

Bu tezde, arıza tespiti, teşhisi ve tahmini için akıllı hesaplama tekniklerini kullanan çevrimiçi ve gerçek zamanlı algoritmalar geliştirilmiştir. Yaptığım bu tez çalışmasının konuyla ilgilenen lisans, yüksek lisans ve doktora öğrencilerinin yanı sıra bu konuda çalışmak isteyen akademisyenlere bir rehber olacağını umuyorum. Tezdeki yazılımsal ve donanımsal uygulamaların temin edilmesinde maddi desteklerinden dolayı Türkiye Bilimsel ve Teknolojik Araştırma Kurumuna ve Fırat Üniversitesi Bilimsel Araştırma Projeleri Birimine teşekkür ederim.

Akademik hayatım boyunca bilgi ve önerileriyle beni yönlendiren, maddi ve manevi desteğini benden esirgemeyen danışman hocam, Prof. Dr. Erhan AKIN'a teşekkür ederim. Akademik çalışmayı bana öğreten, yaptığım bütün çalışmalarda yardımını esirgemeyen hocam, Yrd. Doç. Dr. Mehmet KARAKÖSE'ye teşekkür ederim.

İlhan AYDIN
ELAZIĞ-2011

İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	VI
SUMMARY	VII
ŞEKİLLER LİSTESİ.....	VIII
TABLolar LİSTESİ	XIII
SEMBOLLER LİSTESİ	XIV
KISALTMALAR.....	XVII
1. GİRİŞ.....	1
1.1 Durum İzleme ve Arıza Teşhisinin Genel Çatısı	3
1.1.1 Bazı Kavramlar	4
1.1.2 Arıza Teşhisinin Genel Çatısı.....	5
1.2 Bakım ve Durum İzlemenin Gerekli Olduğu Sistemler	10
1.3 Asenkron Motorlarda Oluşan Arızalar.....	12
1.3.1 Stator Arızaları.....	12
1.3.2 Rotor Arızaları	13
1.3.3 Mil Yatağı Arızaları	14
1.4 Literatür Özeti.....	15
1.5 Tezin Amacı ve Kapsamı	21
1.6 Tezin Yapısı.....	22
2. DURUM İZLEME VE ARIZA TEŞHİSİNİN TEMELLERİ.....	26
2.1 Duyargalar	27
2.2 Veri Toplama Kartı.....	30
2.3 Sinyal İşleme ve Özellik Çıkarımı.....	31
2.3.1 Sinyallerin Büyüklükleri.....	32
2.3.2 Park Vektör Dönüşümü	32
2.3.3 Temel Bileşen Analizi.....	34
2.3.4 Hilbert Dönüşümü	36
2.3.5 Motor Akım İmza Analizi.....	37

2.3.6	Zaman Serileri Analizi.....	42
2.4	Arıza Belirleme ve Teşhis	46
2.5	Çalıştırma Stratejisi	48
3.	YAPAY BAĞIŞIKLIĞA DAYALI ARIZA BELİRLEME YÖNTEMLERİ ..	49
3. 1	Yapay Bağışık Sistemler	49
3.1.1	Negatif Seçim Algoritması	50
3.1.2	Klonal Seçim Algoritması	53
3.2	Arıza Tespiti için Genetik Algoritma Tabanlı Negatif Seçim Algoritması ..	56
3.2.1	Genetik Algoritma	56
3.2.2	Yeni Özellik Sinyali ve Faz Uzayının Oluşturulması	57
3.2.3	Genetik Algoritma Tabanlı Detektör Üretimi	58
3.2.4	DeneySEL Sonuçlar.....	61
3.3	Arıza Tespiti için Kaotik Hibrit Negatif Seçim Algoritması	66
3.3.1	Detektör Üretim Aşaması	67
3.3.2	Tespit Aşaması.....	72
3.3.3	DeneySEL Sonuçlar.....	74
3.4	Bölüm Değerlendirmesi ve Sonuçlar	78
4	YAPAY BAĞIŞIKLIK TABANLI ARIZA TEŞHİS YÖNTEMLERİ	79
4.1	Arıza Teşhisi için Bağışık Sistem Tabanlı Destek Vektör Sınıflandırıcı.....	79
4.1.1	Destek Vektör Makinalar	80
4.1.2	Destek Vektör Makinaların Parametrelerinin Bağışıklık Tabanlı Seçimi...	83
4.1.3	DeneySEL Sonuçlar.....	86
4.2	Sürü Öğrenme Tabanlı Yapay Bağışık Arıza Teşhis Yöntemi	92
4.2.1	Parçacık Sürü Optimizasyonunun Prensipleri.....	92
4.2.2	Sürü Öğrenme Tabanlı Yapay Bağışık Sınıflandırıcı	94
4.2.3	DeneySEL Sonuçlar.....	99
4.3	Uyarlamalı Yapay Bağışık Sistem Tabanlı Arıza Teşhis Yöntemi	106
4.3.1	Özellik Çıkarımı	107
4.3.2	Uyarlamalı Bağışık Sistem Kullanan Arıza Teşhis Aşaması.....	110
4.3.3	DeneySEL Sonuçlar.....	114

	<u>Sayfa No</u>
4.4 Bölüm Değerlendirmesi ve Sonuçlar	119
5. ARIZA TAHMİN VE TEŞHİSİ İÇİN YENİ BİR METOT	121
5.1 Motor Akım İmza Analizinin Eksiklikleri	122
5.2 Arıza Tahmin ve Teşhisi için Kayan Pencere Tabanlı MAIA Yaklaşımı .	123
5.2.1 Aşama I: Arıza Tahmini için Kayan Pencere Tabanlı Sembol Serisi	124
5.2.2 Aşama II: MAIA Tabanlı Kayan Pencere ile Teşhis	127
5.2.3 Deneysel Sonuçlar.....	128
5.3 Bölüm Değerlendirmesi ve Sonuçlar	136
6. GERÇEK ZAMANLI ARIZA TESPİT, TEŞHİS VE TAHMİNİ.....	138
6.1 FPGA	138
6.1.1 FPGA Mimarisi	139
6.1.2 FPGA’da Uygulama Geliştirme Aşamaları	141
6.1.3 VHDL Donanım Tanımlama Dili ve FPGA’ın Programlanması.....	144
6.2 Bulanık Mantık Tabanlı Arıza Teşhis Algoritması.....	152
6.3 Yapay Sinir Ağı Tabanlı Arıza Tahmin Algoritması.....	161
6.3.1 Yapay Sinir Ağı için Özellik Çıkarımı	161
6.3.2 Yapay Sinir Ağı ile Arıza Tahmini	164
6.3.3 Deneysel Sonuçlar.....	165
6.4 Kaskad Yapay Sinir Ağı Tabanlı Arıza Teşhisi	168
6.4.1 Kaskad Yapay Sinir Ağının Yapısı ve Eğitimi	168
6.4.2 Kaskad Yapay Sinir Ağının FPGA’da Gerçekleştirilmesi	171
6.4.3 Deneysel Sonuçlar	173
6.5 Gerçek Zamanlı Arıza Tespiti için Negatif Seçim Algoritması.....	177
6.5.1 Özellik Çıkarımı ve Arıza Tespiti	178
6.5.2 Deneysel Sonuçlar.....	181
6.6 Bölüm Değerlendirmesi ve Sonuçlar	184
7. SONUÇLAR.....	186
7. 1 Öneriler ve Gelecek Çalışmalar	190
KAYNAKLAR.....	191
EKLER.....	206
EK-1.....	206
ÖZGEÇMİŞ.....	208

ÖZET

Bakım işlemi endüstriyel ürün süreci ve verimlilik için önemlidir. Kestirimci bakım reaktif bakıma göre beklenmeyen arıza durumunda maliyetlerin azaltılması için daha etkilidir. Bazı sistem parametrelerinin ölçülmesi ile veri toplamaya dayanan durum izleme tabanlı kestirimci bakım, alınan verilerden belirli özellikleri çıkarıp arıza tespiti, teşhisi veya tahmini için kullanılır.

Bu tezde, gerçek zamanlı ve çevrimiçi durum izleme ve arıza teşhisi için akıllı hesaplama tekniklerinin geliştirilmesi hedeflenmiştir. Bu amaç doğrultusunda teorik yöntemler geliştirilerek pratik uygulamaları gerçekleştirilmiştir. Endüstriyel sistemlerde çok yoğun kullanılan ve önemli bir bileşen olan elektrik makinaları tez kapsamında uygulama problemi olarak seçilmiş ve bu makinalarda arıza tespiti, teşhisi ve tahmini için akıllı hesaplama tekniklerine dayalı gerçek zamanlı yöntemler geliştirilmiştir.

Tezin teorik yönünü kapsayan yöntem geliştirme aşamasında arıza tespit, teşhis ve tahmini için yapay bağışık sistemler, destek vektör makinalar, yapay sinir ağları ve bulanık mantık tabanlı algoritmalar önerilmiştir. Önerilen algoritmaların yakınsama hızları ve teşhis performanslarını arttırmak için algoritmaların parametreleri parçacık sürü optimizasyonu, klonal seçim ve genetik algoritma ile ayarlanmıştır. Ayırt edici özelliklerin çıkarılması için sinyal işleme tabanlı yöntemler geliştirilmiştir. Geliştirilen yöntemler farklı çalışma hızlarında ve az veri gereksinimi ile arızaları belirlemektedir.

Tez çalışmasının uygulama aşamasında ise oluşturulan deney düzeneği ile sistemden sinyallerin alınması sağlanmıştır. Önerilen yöntemler çevrimiçi olarak gerçekleştirilmiş ve gerçek zamanlı olarak Altera Cyclone III FPGA kartı üzerinde uygulanmıştır. Geliştirilen yöntemler yapılan deneysel çalışmalar ile doğrulanmıştır.

Sonuç olarak, bu tezde geliştirilen teknikler tez sürecinde yürütülen FÜBAP ve TUBİTAK 1001 araştırma projeleri ile desteklenmiştir. Yapılan çalışmalar ulusal ve uluslararası düzeyde bilimsel yayınlar ile sonuçlandırılmıştır.

Anahtar Kelimeler: Arıza tespit, teşhis ve tahmini, akıllı hesaplama, sinyal işleme, FPGA (Sahada programlanabilir kapı dizileri), elektrik makinaları.

SUMMARY

DEVELOPMENT OF IMMUNE INTELLIGENT COMPUTING TECHNIQUES FOR REAL-TIME CONDITION MONITORING AND FAULT DIAGNOSIS

Maintenance operation is important for the process of industrial product and efficiency. Predictive maintenance is more effective than reactive one for reducing of costs in unexpected fault condition. Condition monitoring based predictive maintenance, which is based on measuring of some system parameters and data acquisition, is used to fault detection, diagnosis, and prognosis by extracting certain features from acquired data.

In this thesis, the aim is to develop intelligent computing techniques for real-time and online condition monitoring and fault diagnosis. In accordance with this aim, theoretical methods are developed and their practical applications are implemented. Electrical machines, which are rigorously used in industrial systems and important components, are selected as an application problem in thesis scope. Intelligent computing techniques based real time methods are developed for fault detection, diagnosis, and prognosis in these machines.

In method development stage covered the theoretical side of the thesis; artificial immune systems, support vector machines, artificial neural networks, and fuzzy logic are proposed for fault detection, diagnosis and prognosis. The parameters of the algorithms are adjusted by using particle swarm optimization, clonal selection, and genetic algorithm to improve the convergence speed and diagnosis performance of them. Signal processing based methods are developed for extracting distinctive features. Developed methods detect faults with a little data necessity and under different operation speeds.

In the application stage of the thesis, signals are acquired from system by founding experimental setup. The proposed methods are performed as online and they are implemented on Altera Cyclone III FPGA card as real time. The developed methods are verified by experimental studies.

As a result, developed techniques are supported by FUBAP and TUBITAK 1001 projects conducted in the thesis process. Studies are concluded by national and international level of academic publications.

Key words: Fault detection, diagnosis and prognosis, intelligent computing, signal processing, FPGA (Field Programmable Gate Arrays), electric machines.

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 1.1	Durum izleme bileşenleri	3
Şekil 1.2	Arıza teşhis sisteminin genel yapısı	5
Şekil 1.3	Arıza teşhis yöntemlerinin çeşitleri	7
Şekil 1.4	Veriye dayalı arıza teşhisinin eğitim ve test aşamaları	8
Şekil 1.5	Veriye dayalı arıza teşhisinin gerçek zamanlı gerçekleştirilmesi	9
Şekil 1.6	Stator kısa devre arızaları	13
Şekil 1.7	Üç adet kırık rotor çubuğuna sahip rotor	14
Şekil 2.1	Durum izleme bileşenleri	26
Şekil 2.2	Bir ölçüm cihazının şematik gösterimi	28
Şekil 2.3	Hall-etki duyarganın çalışma prensibi	29
Şekil 2.4	Hall-etki akım duyargası	30
Şekil 2.5	Veri toplamanın genel düzeni	30
Şekil 2.6	14-bit USB veri toplama kartı	31
Şekil 2.7	Park vektör dönüşümü	33
Şekil 2.8	Temel bileşen analizinin geometrik yorumu	35
Şekil 2.9	Periyodik bir sinyal ve Hilbert dönüşümü	37
Şekil 2.10	Fourier dönüşümünün uygulanması	38
Şekil 2.11	Ayrık Fourier dönüşümünde farklı pencereler	39
Şekil 2.12	5-noktalı çapraz doğrulama ile sınıflandırma başarımının hesabı	48
Şekil 3.1	Negatif seçim algoritması	52
Şekil 3.2	Biçim uzayı tabanlı antijen tanıma sistemi	52
Şekil 3.3	Klonal seçim prensibi.....	54
Şekil 3.4	Klonal seçim algoritması	55
Şekil 3.5	Genetik algoritmanın adımları	57
Şekil 3.6	Kullanılan kodlama	59
Şekil 3.7	Önerilen yöntemin eğitim aşaması.....	60
Şekil 3.8	Sağlam ve bir kırık rotor çubuğu motor akım sinyali	61
Şekil 3.9	Sağlam ve arızalı motor özellik sinyalleri.....	62
Şekil 3.10	İlk fark filtrelemesi ile oluşturulan özellik sinyallerinin faz uzayları	62

Şekil 3.11 Hilbert dönüşümü tabanlı çıkarılan özelliklerin faz uzayları	63
Şekil 3.12 Genetik algoritma ile üretilen detektörler ve algoritmanın performansı.....	63
Şekil 3.13 Sağlam motor için faz uzayı ve genetik algoritma ile üretilen detektörler	64
Şekil 3.14 Aktif detektör sayıları	65
Şekil 3.15 Her arıza türü için detektörlerin aktifleşme oranları	65
Şekil 3.16 Sinai kaotik haritası.....	68
Şekil 3.17 Kaotik mutasyon operatörü	70
Şekil 3.18 Detektör üretim aşaması.....	71
Şekil 3.19 Tespit aşaması	73
Şekil 3.20 Yeni test örneği için detektörlerin uyarlanması	73
Şekil 3.21 Deney düzeneği şeması	74
Şekil 3.22 Sağlam ve arızalı motor akım spektrumu	75
Şekil 3.23 Arızalı veriler için üretilen detektörler.....	76
Şekil 3.24 Maksimum detektör boyutuna göre tespit ve yanlış alarm oranı	77
Şekil 4.1 Ayrırma düzlemi ve destek vektörler	81
Şekil 4.2 Destek vektör makinaların eğitim aşaması	83
Şekil 4.3 Önerilen algoritmanın akış şeması	84
Şekil 4.4 Antikorların ikili kodlanması	84
Şekil 4.5 Mutasyon oranı ve ölçeklendirilmiş afinite arasındaki değişim	86
Şekil 4.6 Veri ön işleme adımı	87
Şekil 4.7 Özellik vektörü.....	87
Şekil 4.8 Deney düzeneği şeması	88
Şekil 4.9 Faz akımları, park vektör bileşenleri ve özellik vektörü.....	89
Şekil 4.10 Sağlam motor faz uzayı.....	89
Şekil 4.11 Birleştirilmiş faz uzayı	90
Şekil 4.12 Klonal seçim algoritmasının performansı	90
Şekil 4.13 Optimum çekirdek ve ceza parametresi ile DVM'nin sonucu	91
Şekil 4.14 İki parçacığın pbest ve gbest'e göre iki boyutlu gelişimi	93
Şekil 4.15 Orijinal PSO algoritmasının temel adımları.....	93
Şekil 4.16 Standart yapay bağışık sistem ve SÖTYBS'de antikorların hareketi	95
Şekil 4.17 SÖTYBS diyagramı	96

Şekil 4.18	Deney düzeneği.....	100
Şekil 4.19	380 V besleme geriliminde sağlam ve arızalı akım sinyalleri	101
Şekil 4.20	380 V besleme geriliminde faz akımının Hilbert dönüşümü	101
Şekil 4.21	Özellik sinyalinin spektrumu	103
Şekil 4.22	380 V için özellik veri kümesi	103
Şekil 4.23	SÖTYBS'nin yakınsama hızı.....	104
Şekil 4.24	Üretilen hafıza hücreleri ve her bir sınıfın faz uzayları	105
Şekil 4.25	Arıza teşhis ve tespiti için önerilen yöntemin blok şeması.....	107
Şekil 4.26	UYBS'nin akış şeması	111
Şekil 4.27	Antikor ve hafıza hücresinin gösterimi	111
Şekil 4.28	C _j için mutasyon işlemi	113
Şekil 4.29	Deney düzeneği şeması	115
Şekil 4.30	Kırık rotor çubuğu için kullanılan Hilbert dönüşümü ve spektrumu	116
Şekil 4.31	Sağlam ve arızalı motor için park vektör dönüşümleri ve temel bileşen analizi tabanlı yeni dönüşüm.....	117
Şekil 4.32	Üç sınıf verisinin üç boyutlu gösterimi.....	117
Şekil 4.33	Eğitim aşamasında UYBS'nin sınıflandırma başarımı	118
Şekil 4.34	Benzerlik ölçümüne göre hafıza hücrelerinin sayısı	119
Şekil 5.1	Yükün yan bant bileşenleri üzerindeki etkisi	122
Şekil 5.2	Arıza teşhis yönteminin şematik gösterimi	123
Şekil 5.3	Kayan pencere tabanlı online arıza tespit algoritması.....	124
Şekil 5.4	Reel verilerin sembolleştirilmesi ve sembol ağacının oluşturulması.....	125
Şekil 5.5	Kayan pencere gösterimi.....	126
Şekil 5.6	Deney düzeneği.....	128
Şekil 5.7	MATLAB/SIMULINK modeli	129
Şekil 5.8	Bir periyotluk faz akımı ve elde edilen özellik sinyali	130
Şekil 5.9	Farklı pencere boyutları için bilgi entropisi	131
Şekil 5.10	Sembol ağacının farklı seviyesine göre bilgi entropisi	132
Şekil 5.11	Faz akımı üzerinde uygulanan MAIA.....	133
Şekil 5.12	Hilbert dönüşümü tabanlı kayan pencerelerin spektrumu.....	134

Şekil 5.13 Dört besleme gerilimi için kayan pencere tabanlı Hilbert dönüşümünün spektrumu	135
Şekil 6.1 Bir FPGA'ın genel yapısı.....	140
Şekil 6.2 Lojik hücre yapısı.....	141
Şekil 6.3 FPGA kullanılarak yapılan tasarımın blok diyagramı	142
Şekil 6.4 A ³ yönteminde kullanılan mimari optimizasyon şekilleri.....	143
Şekil 6.5 FPGA tasarım örneği.....	143
Şekil 6.6 SOPC'nin yapısı.....	145
Şekil 6.7 Quartus II sayıcı blok diyagramı ve iç yapısı	148
Şekil 6.8 Quartus II'de derleme sonucu sayıcının kullandığı FPGA kaynakları	149
Şekil 6.9 Sayıcı için Altera Modelsim benzetim sonuçları	149
Şekil 6.10 Pin atamaları	150
Şekil 6.11 32-bit IEEE-754 standardı	151
Şekil 6.12 Bulanık arıza teşhis sisteminin blok diyagramı	152
Şekil 6.13 Kayan pencere modülünün VHDL kodu	153
Şekil 6.14 Bir faz akımının büyüklüğünün hesaplanması	154
Şekil 6.15 Bulanık sistemin blok diyagramı	155
Şekil 6.16 Üyelik fonksiyonları ve kural tabanı.....	156
Şekil 6.17 Durulandırma işlemi için PROCESS bloğu	157
Şekil 6.18 Sağlam ve arızalı durumlar için FPGA deneysel sonuçları	158
Şekil 6.19 Aynı giriş değerleri için MATLAB sonuçları	159
Şekil 6.20 FPGA sonuçları ile MATLAB bulanık mantık TOOLBOX'ın karşılaştırılması	159
Şekil 6.21 Bulanık sistemin FPGA uygulaması	160
Şekil 6.22 RAM'den okuma için kullanılan VHDL kodunun PROCESS bloğu	162
Şekil 6.23 Akım sinyalinden özellik sinyalinin elde edilmesi	162
Şekil 6.24 Özellik sinyalinin ortalama ve standart sapma değerinin hesaplanması.....	163
Şekil 6.25 Yapay sinir ağı yapısı.....	164
Şekil 6.26 Nöron modelinin blok şeması	164
Şekil 6.27 Aktivasyon fonksiyonunun Quartus II modeli.....	165
Şekil 6.28 Sağlam ve arızalı faz akımı ve özellik sinyalleri	166

Şekil 6.29 Yapay sinir ağının sağlam ve arızalı durumlar için FPGA üzerinde test edilmesi	167
Şekil 6.30 Arıza teşhisi için önerilen kaskad yapay sinir ağının yapısı	169
Şekil 6.31 Eğitim için önerilen PSO'nun akış şeması	169
Şekil 6.32 Kaskad yapay sinir ağının FPGA blok şeması	172
Şekil 6.33 Kaskad yapay sinir ağının VHDL ile yazılan kontrol birimi	173
Şekil 6.34 Gerçek çıkış ve yapay sinir ağı çıkışı	174
Şekil 6.35 Fonksiyon yaklaştırma için kaskad yapay sinir ağının benzetim sonuçları	174
Şekil 6.36. Kaskad yapay sinir ağı eğitim verileri ve eğitim hatasının değişimi	175
Şekil 6.37. Kontrol için kullanılan sonlu durum makinasının benzetim sonuçları	175
Şekil 6.38. Kaskad yapay sinir ağının arıza teşhisi için Modelsim benzetim sonuçları ...	176
Şekil 6.39 Önerilen negatif seçim tabanlı gerçek zamanlı arıza tespit yöntemi.....	178
Şekil 6.40 Pencere fonksiyonu ve faz kaydırma için VHDL kodu	179
Şekil 6.41 Özellik sinyalinin elde edilmesi	179
Şekil 6.42 Negatif seçim algoritmasının tespit aşaması	180
Şekil 6.43 Sağlam ve arızalı durum için akım ve özellik sinyalleri	182
Şekil 6.44 Sağlam ve arızalı faz uzayları ve üretilen detektörler	183
Şekil 6.45 Sağlam ve arızalı motor için Modelsim benzetim sonuçları	184

TABLÖLER LİSTESİ

Sayfa No

Tablo 1.1 Akıllı hesaplama ve sinyal işleme tabanlı arıza teşhis yöntemlerinin literatür sınıflandırılması	20
Tablo 2.1 2x2 olasılık tablosu.....	47
Tablo 3.1 Simülasyonda kullanılan asenkron motor karakteristikleri	61
Tablo 3.2 Yöntemin farklı veriler üzerinde test sonuçları	64
Tablo 3.3 İlk fark filtrelemesi genetik algoritma tabanlı arıza tespiti	65
Tablo 3.4 KHNSA algoritmasının eğitim parametreleri.....	72
Tablo 3.5 Deneyde kullanılan asenkron motorun özellikleri.....	74
Tablo 3.6 Farklı parametrelere göre KHNSA'nın performansı	76
Tablo 3.7 KHNSA'nın diğer akıllı teknikler ile karşılaştırılması.....	77
Tablo 4.1 Popüler çekirdek fonksiyonları.....	82
Tablo 4.2 Deneyde kullanılan asenkron motorun özellikleri.....	88
Tablo 4.3 Arıza teşhisi için yapay bağışık sistem tabanlı DVM algoritmasının sonuçları. 91	
Tablo 4.4 PSO'nun parametreleri	99
Tablo 4.5 SÖTYBS'nin parametreleri	104
Tablo 4.6 İlk özellik grubu için arıza teşhis doğruluğu	105
Tablo 4.7 İkinci özellik grubu üzerinde üç yöntemin performans karşılaştırması	106
Tablo 4.8 Eğitim ve test verilerinin özellikleri	115
Tablo 4.9 UYBS'nin parametreleri.....	118
Tablo 4.10 Diğer bağışık sınıflandırıcılar ile karşılaştırma	119
Tablo 5.1 Farklı besleme gerilimleri için bilgi entropilerinin meydana.....	132
Tablo 5.2 Yöntemin literatürdeki yeri	136
Tablo 6.1 FPGA tabanlı bulanık sistem tasarımı için gerekli kaynaklar	160
Tablo 6.2 YSA FPGA tasarımı için gerekli kaynaklar	166
Tablo 6.3 Kaskad yapay sinir ağı için kullanılan PSO'nun parametreleri	174
Tablo 6.4 Kaskad yapay sinir ağı için kullanılan FPGA kaynakları	177
Tablo 6.5 Negatif seçim algoritması için kullanılan FPGA kaynaklarının oranı	183

SEMBOLLER LİSTESİ

Acc^+, Acc^-	: Doğru pozitif ve negatif oranı
aff_{avg}, aff_{mci}	: Hafıza kümesinin ortalama afinitesi ve i. hafıza hücresinin afinitesi
AK, AN, ak, an	: Antikor ve antijen kümesi, bireysel antikor ve antijen
$ak_c \cdot f_{jd}$	: j. klonun d. Özelliği
$ak_{dc} \cdot f_j$	: Geçerli antikor ile farklı sınıftaki en yakın antikor
$ak_{eniyi} \cdot f_j$	: j. antikor tarafından şu ana kadar elde edilen en iyi pozisyon
$ak_{sc} \cdot f_j$	: Geçerli klon ile aynı sınıftaki en yakın antikor
AS	: Analitik sinyal
b	: Gömülme boyutu
$b\ddot{o}$	: Benzerlik ölçümlerinin değeri
b_s, b_y	: Mil yatağındaki bilyelerin sayısı ve yarıçapları
C ve σ	: Ceza parametresi ve çekirdek fonksiyonunun parametresi
c_1, c_2, c_3	: Sabit ağırlık faktörleri
C_i	: Klon kümesi
$D, d, d.af, d.r$	: Detektör kümesi, detektör, detektörün afinitesi ve yarıçapı
d_o	: İki nokta arasındaki değişim oranı
F	: Faz uzayı matrisi
f^*	: $[0, 1]$ aralığına çekilmiş afinite
Fa	: Negatif tespit oranı
$f_s, f_b, f_e, f_{ma}, f_{i,o}$	: Besleme, kırık rotor çubuğu, eksantriklik mil yatağı yan bant frekansları ve mil yatağı karakteristik titreşim frekansı
HD	: Hilbert dönüşümü
hh	: Hafıza hücresi
H_s	: Ağacın bilgi entropisi
I_A, I_B, I_C	: A, B ve C faz akım sinyalleri
I_d, I_q	: Park vektör bileşenleri
I_p	: Karmaşık pozitif sıra bileşenin modülünün ortalaması
i_M	: Faz besleme akımının maksimum değeri
K	: Detektör kümesinin kapsaması
k_{ij}	: x_i ve x_j terimleri arasındaki korelasyon değeri

k_{ii}^2	: x_i değişkenin varyansı
K_M	: Korelasyon matrisi
M_n	: İki nokta arasındaki mesafe
M_P	: Tek faz akımının maksimum gücü
m_y	: Mil yatağının yarıçapı
N_C	: Klon sayısı
n_{mc}	: Toplam hafıza hücrelerinin sayısı
NS	: Öz olmayan örneklerin sayısı
n_s, n_m, n_ω, n_d	: Motorun senkron ve mekanik hızı, stator eksantriklik düzeni, dönen eksantriklik frekansı
$\tilde{O}, \tilde{o}, \tilde{o}.r$	: Öz veri kümesi, Öz veri örneği ve öz veri örneğinin yarıçapı
P	: Çift kutup sayısı
p_g	: i. parçacığın komşuları tarafından bulunan en iyi pozisyon
P_{id}, P_{Iq}	: Id ve Iq bileşenlerinin tepeden tepeye değerleri arasındaki fark
p_i	: i. parçacığın şu ana kadarki en iyi pozisyonu
P_{xx}	: Güç spektrum yoğunluğu
rms_{IA}	: A faz akımının efektif değeri
s	: Motor kayması
S	: Öz örneklerin sayısı
Sdc	: Ağırlıklı oylama yöntemi
S_f	: Stator arızası için özellik değeri
S_H, S_A, S_U	: Hat, düşük ve yüksek yan ban frekanslarının genlikleri
SS	: Statik dönüşüm
T	: Periyot
U	: Dönüşüm matrisi
v	: Öz vektör
$v_{i,j}(t)$	: t adımımda i. parçacığın j. boyutunun hızı
v_j^{eski}	: j. antikoron eski hızı
w	: Pencere boyutu
$x_{i,j}(t)$	: t adımımda i. parçacığın j. boyutunun pozisyonu
x_k	: Kaotik bir haritanın genel formülü

x_n^{EYK}	: x_n 'e en yakın nokta
z	: Dönüşüm denklemi
Δf	: Fourier dönüşümünün çözünürlüğü
Δi	: İlk fark filtrelemesi
$\Delta \sigma$	: Standart sapmanın değişimi
α	: Mutasyon oranı
χ	: Kısıtlama faktörü
ε	: Eşik değeri
κ	: Parçacığın hızını kontrol eden moment parametresi
λ	: Öz değer
$\mu_{\omega i}$	: Bulanık üyelik derecesi
θ	: Halkadaki bilyelerin bağlantı açısı
ρ	: Kontrol parametresi
σ_H	: Standart sapma
τ	: Zaman gecikmesi
ω	: Besleme frekansının açısal hızı
ω_k	: Reel frekans değişkeni
ω_x	: Bir test örneğinin sınıfı
$\psi_1, \psi_2, \psi_3, \psi_4$	: $[0, 1]$ aralığında rastgele sayılar

KISALTMALAR

ADC	: Analog Dijital Dönüştürücü
ASICs	: Uygulamaya Özgü Tümleşik Devreler
CAD	: Bilgisayar Destekli Tasarım Araçları
CLBs	: Ayarlanabilen Lojik Bloklar
CMOS	: Tamamlamalı Metal-oksit Transistorlar
DSP	: Sayısal İşaret İşlemciler
DVM	: Destek Vektör Makinalar
FFT	: Hızlı Fourier Dönüşümü
FPGA	: Sahada Programlanabilir Kapı Dizileri
KHNSA	: Kaotik Hibrit Negatif Seçim Algoritması
LC	: Lojik Hücre
LE	: Logic Eleman
LUT	: Bakma Tabloları
MAIA	: Motor Akım İmza Analizi
PALs	: Programlanabilir Dizi Mantık Aygıtları
PLAs	: Programlanabilir Lojik Diziler
PSO	: Parçacık Sürü Optimizasyonu
RLAIC	: Kaynak Sınırlı Yapay Bağışık Sınıflandırıcı
SAIS	: Basit Yapay Bağışık Sistem
SOPC	: System on a Programmable Chip Builder
SÖTYBS	: Sürü Öğrenme Tabanlı Yapay Bağışık Sınıflandırıcı
SRAM	: Statik RAM
UYBS	: Uyarlamalı Yapay Bağışık Sistem
VHDL	: Very High Speed Integrated Circuits Description Language
VLSI	: Çok Yüksek Ölçekli Entegrasyon

1. GİRİŞ

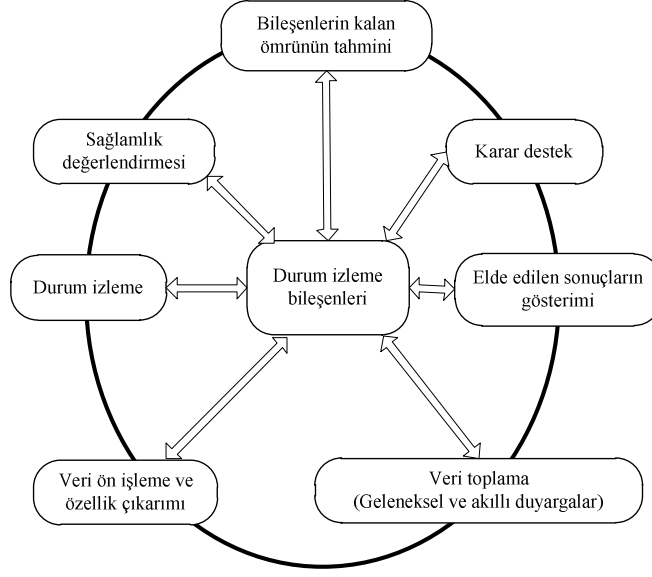
Endüstriyel sistemlerdeki gelişmeler ile birlikte üretilen yeni ürünler günlük yaşamda birçok işin kolaylıkla yapılabilmesini sağlamaktadır. Günlük hayatta kullanılan çamaşır makinesi, buzdolabı, bulaşık makinesi, mutfak robotları gibi ev aletleri, uzay araçları, hibrit otomobiller ve askeri araçlar gibi ürünler yoğun bir süreç sonrasında elde edilmektedir. Karmaşık olan bu süreçlerde devamlılığın sağlanması oldukça önemlidir [1]. Gittikçe daha da karmaşık bir yapıya sahip olan bu sistemler sık sık arızalara maruz kalabilirler. İstenmeyen sistem dinamikleri ile tanımlanan sistem arızaları üretim yapan bir fabrikada işlerin aksaması, büyük ekonomik kayıplar ve fabrikada çalışanların yaralanmaları gibi ciddi sonuçlara yol açabilir. Son yıllarda hızlı bilgi işleme ve kaydetme araçlarının gelişmesi ile birlikte, endüstride kullanılan sistemlerin izlenmesi ve oluşacak herhangi bir aksaklığın kısa bir sürede giderilmesi önemli bir gereksinim haline gelmiştir. Herhangi bir sistemden gerçek zamanlı ölçülen parametrelerdeki değişimler, sistemin çalışması hakkında genel bir fikir verir [2]. Bu değişimlerin algoritmik olarak incelenmesi ile durum izleme tabanlı koruma ve arıza teşhisi, hem askeri hem de ticari uygulamalarda kullanılan önemli bir teknoloji olarak ortaya çıkmıştır. Uygulama bileşenlerinin kalan ömürlerinin tahmini, durum izleme ve arıza teşhisi mühendislik bilimi, bilgisayar bilimi, yönetim ve iletişim gibi farklı disiplinler birleştirilerek sağlanır [3].

Mühendislik, tıp, bilim, iş ve finans gibi farklı disiplinlerdeki araştırmacılar, kullandıkları sistemlerde oluşan arızaların sebepleri, bunların sistemin sağlamlığı üzerindeki etkileri ve arızalara karşı sınır tolerans değerlerinin nasıl belirleneceği üzerine araştırmalar yapmışlardır [2]. Sistemlerin izlenmesi için ilk olarak genel amaçlı röleler kullanılmıştır [4]. Bu araçların temel çalışma prensibi belirli çalışma sınırlarının tanımlanmasına dayanır. Bu sınırlar ilgili bileşen için arıza toleransı olarak da bilinir. Eğer endüstriyel sistemdeki bir bileşen bu sınırların dışında belirli bir süre çalışırsa ilgili bileşen devre dışı bırakılarak sistem durdurulur. Fakat bu tür koruma araçları sadece arıza oluşuktan sonra devreye girerek sistemi durdururlar. Sistemin tamamen durması ile sonuçlanabilen bu arızalar üretim yapan bir fabrikada aksaklıklara sebep olur. Arızanın oluşmadan önce kestirilmesi oldukça zordur. Örneğin bir uçak veya uzay aracının önemli bir bileşeninde arızanın oluşumu erken bir aşamada belirlenemezse araçta bulunan kişiler ile ilgili ciddi yaralanmalar hatta ölümler olabilir. Dolayısıyla koruma kavramı, güvenlik

ve hassasiyetin üst seviyede olduğu sistemlerde tek başına yeterli olmamaktadır. Bunun yerine durum izleme tabanlı bakım kavramı ortaya çıkmıştır [4].

Durum izleme tabanlı bakım, bir endüstriyel sistemdeki herhangi bir bileşenin durumu hakkında bilgi edinmek için bileşenin çalışma zamanı verilerinin kullanılmasını gerektirir. Endüstriyel sistemlerin durumunun izlenmesindeki amaç, bu sistemlerin kullanıldığı süre boyunca performansının sürekli izlenmesidir. Durum izleme bu özelliği ile korumadan ayrılır. Korumada amaç, sistemde herhangi bir arıza oluşuktan sonra arızaya sebep olan bileşenin çıkarılmasıdır [1]. Durum izlemede ise sistemdeki arızanın önceden tahmin edilmesi amaçlanır. Sistemin sürekli izlenmesi ile herhangi bir arızanın gelişimi erken bir aşamada tespit edilebilir. Dolayısıyla durum izleme için oluşturulan sistem, koruma rölesi yerine de kullanılabilir. Sistemlerin durumları sürekli izlendiğinde erken bir aşamada arızanın belirlenmesi ve ilgili bileşenin tamiri veya değiştirilmesi mümkün olabilir. Durum izlenmesinde sistemlerin hangi aralıklarla izleneceği önemlidir. Sistemler belirli periyotlarda veya sürekli izlenebilir. Durumu izlenen sistemin davranışlarının incelenmesi, bileşenin kalan ömrünün tahmini ve sistemi korumak için gerekli planlamaların yapılması sistemin devamlılığı için oldukça önemlidir [5].

Tasarımcılar arızanın gelişimini tahmin etmek, arızayı belirlemek ve gerekirse izole etmek için gerekli olan donanımsal bileşenleri, algoritmayı ve uygun izleme stratejisini seçmelidir. Ayrıca geliştirilen donanımının doğruluğu ve performansının değerlendirilmesi için gerekli testler de uygulanmalıdır. Sistemden hangi parametrelerin ölçüleceği, ne tür verilerin sistemden alınacağı ve nasıl kullanılacağı da önemlidir. Durum izleme için genelde geleneksel veya akıllı duyargalar aracılığıyla veriler alınmaktadır. Farklı disiplinlerde durum izleme üzerine çalışan araştırmacılar bir sistem veya süreci sürekli izleyerek arızaları bulmak, onların yerini belirlemek veya izole etmek ve bu arızaların sistem üzerindeki etkilerinin belirlenmesi için yeni yöntemler üzerinde çalışmaktadırlar. Temel amaç, farklı duyargalardan alınan sinyal veya bilgilerin değerlendirilerek sistemin durumunu öğrenmek ve gerektiğinde sisteme müdahale edebilmeyi sağlamaktır. Genel yapı itibarı ile bir durum izleme ve arıza teşhis sisteminin bileşenleri Şekil 1.1 ile verilebilir.



Şekil 1.1. Durum izleme bileşenleri

Durum izleme tabanlı arıza teşhisi, tespiti ve tahmini aşağıda detayları verilen modüller ile bütünsel bir sistemden oluşur [2].

- Bir duyarga grubu ile uygun çözünürlükte verilerin alınması sağlanarak sistemin kritik değişkenlerinin ve parametrelerinin ölçülmesi,
- Sistemin geçerli çalışma durumu ile arızalı çalışma durumu arasındaki farklılıkların sınırlarının belirlenmesi,
- Duyargalardan elde edilen sinyallerden sistemin durumunu belirleyici özelliklerin çıkarılması,
- İzlenen sistemin kritik bileşenlerinin geçerli durumunun değerlendirilmesi ve arızaların teşhis edilmesi,
- Arızaya uğrayan sistemdeki bileşenlerin kalan ömürlerinin tahmini.

1.1. Durum İzleme ve Arıza Teşhisinin Genel Çatısı

Durum izleme ve arıza teşhisi birlikte düşünülmesi gereken iki kavramdır. Çünkü herhangi bir sistemde arızaların belirlenebilmesi için belirli bir periyot süresince sistemin izlenmesi gereklidir. Sistemlerin izlenmesi ve arıza teşhisi kavramı ortaya çıktığından beri farklı arıza teşhis modelleri kullanılmıştır. Bu bölümde arıza teşhisi ilgili bazı kavramlardan bahsedilerek arıza teşhisi ve durum izlemenin gelişimi özetlenecektir.

1.1.1. Bazı Kavramlar

Arıza teşhisi ile ilgili yöntemlerden bahsetmeden önce bu yöntemler ile ilgili bazı kavramların verilmesi gerekir. Bu kavramlar genellikle birbiri ile karıştırılmaktadır. Aşağıda arıza teşhisinde en çok karşılaşılan kavramlar ile ilgili kısa tanımlamalar verilmiştir [6].

Arıza, kabul edilebilir, genel ve standart bir durumdan bir sistemin en az bir karakteristiğinin izin verilmeyen şekilde davranmasıdır. Bir arıza, üretim kaybına veya azalmasına yol açan anormal bir durum olarak ifade edilir.

Durum izleme, yüksek dereceli otomasyona sahip modern endüstriyel sistemlerde önemli bir görevdir. Mevcut endüstriyel sistemler, izleme için kullanılabilen farklı duyarga ve kontrol sinyallerine sahiptir. Bu sinyaller ile arızalarının tanınması ve otomatik olarak belirlenmesi amaçlanır. İzleme, genellikle üzerinde çalışılan sistem hakkında özel bilgiler gerektirir. Durum izleme, bir sistemin çalışma karakteristiklerinin izlenmesi süreci veya tekniği olarak tanımlanabilir. İzlenen sinyallerin değişimi veya eğilimi, ciddi bir hasar veya bozulma oluşmadan önce sistemin durumunu tahmin etmek veya bakımının gerekli olup olmadığını belirlemek için kullanılır.

Otomatik koruma, tehlikeli çalışma durumlarında izleme fonksiyonunun otomatik olarak uygun bir karşı hareket başlatmasıdır. Genellikle izlenen bileşen bir arıza-güvenlik durumu ile karşılaştırılır ve acil olarak sistem durdurulur. Koruma röleleri buna örnek olarak verilebilir.

Güvenirlilik, verilen bir zaman periyodu boyunca belirlenmiş durumlar altında ve verilen bir aralıkta gereken bir fonksiyonu uygulamak için sistemin bir yeteneği olarak tanımlanır.

Arıza belirleme probleminde eğer sistemde arıza varsa, anlık ölçümler kullanılarak arıza belirlenir. Bu durumda kullanılan veri sayısı ne kadar az olursa sistem gerçek zamanlı bir arıza tahmin yöntemine de o kadar yaklaşır.

Arıza teşhisi ile sistemde arıza olup olmadığını belirlemekle birlikte arızanın sebepleri ve şiddeti de belirlenir. Böylece arızanın izole edilmesi sağlanabilir. Çünkü arızalarının sebebi anlaşılırsa bunların çoğundan kaçınmak mümkün olabilir. Bir arıza durumu oluştuğunda bu arızanın ve daha sonra arızanın sebebinin belirlenmesi ve eğer giderilecek bir durum ise giderilmesi veya onu gidermeye uygun ortamın hazırlanması gerekir.

Arıza tahmini ise arıza teşhisine ilave olarak donanım bileşenlerinin kalan ömrünün

tahmin edilmesi de sağlanır.

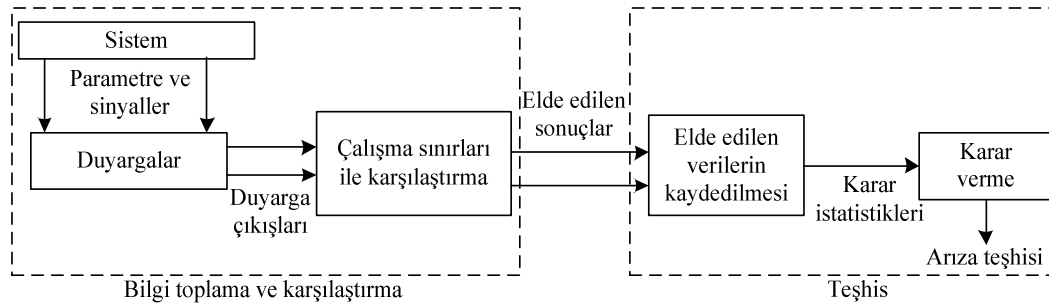
Bakım, bir sistemi içinde olduğu durumda tutmak için yapılan bir işlem veya sistemi tasarlanmış başlangıç çalışma durumuna döndürmektir.

Arıza toleransı, yapılan tasarımlarda güvenilirlik analizi uygulandıktan sonra, ürünün test edilmesi ve kalite kontrol yöntemlerinin uygulanması esnasında sistemlerin belirli bir arıza tolerans yeteneğine sahip olması gerekir.

Bilgi tabanlı arıza tespiti ve teşhisi, arıza teşhisi için ölçülen sinyallerin otomatik olarak işlenmesi analitik bir süreç bilgisi gerektirir ve gözlemlenen sinyallerin değerlendirilmesi sezgisel bilgi olarak isimlendirilen uzman bilgisi gerektirir. Bu işlem arıza teşhisi ve tespiti için bilgi tabanlı bir yaklaşım ile gerçekleştirilebilir.

1.1.2. Arıza Teşhisinin Genel Çatısı

Bir arıza teşhis sistemi genel olarak güvenilirlik, koruma ve maliyet gibi bazı gereksinimleri karşılamalıdır. Arıza teşhis sistemi minimum maliyette olmalı ve sisteme kolaylıkla monte edilebilmelidir [7]. Arıza teşhisi ve tanımlama sistemleri farklı uygulama alanlarında geniş bir şekilde yer bulmuştur. Arıza teşhisinden bahsederken özel bir bileşenin veya bir sistemin daha özel bir elemanında oluşan arızaların belirlenmesi göz önüne alınmalıdır. Çünkü arızalar daha çok temel bir bileşende oluşup sistemin diğer bileşenlerini de etkilemektedir. Bir arıza teşhis sisteminin genel yapısı Şekil 1.2’de verilmiştir.



Şekil 1.2. Arıza teşhis sisteminin genel yapısı

Şekil 1.2’de verilen arıza teşhis sisteminde ilk olarak izlenen sistemden belirli parametreler veya sinyaller duyargalar aracılığıyla ölçülür. Duyarga çıkışlarından elde edilen sinyaller sistemin normal çalışma sınırları ile karşılaştırılarak standart çalışma durumundan sapmalar bulunur. Elde edilen sonuçlar belirli bir süre boyunca kaydedilerek

karar verme aşamasında sistemin durumu hakkında bilgi verilir. Sistemin çalışma durumunun arızalı olup olmadığı, zaman veya frekans alanındaki sinyaller incelenerek veya sistemin modeli oluşturularak model çıkışı ile gerçek çıkış karşılaştırılarak değerlendirilir.

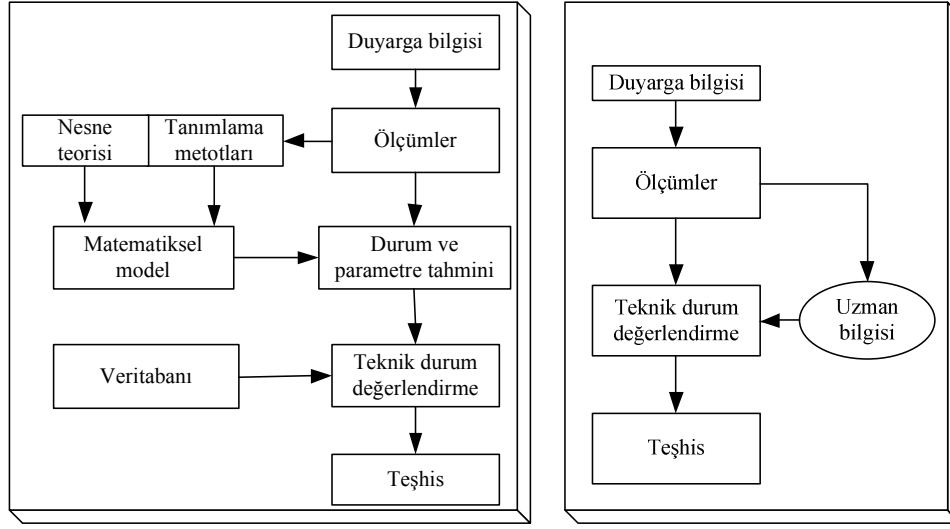
Arıza teşhis sisteminde aşağıdaki üç temel yaklaşım uygulanmaktadır [5]:

- Model tabanlı,
- Veriye dayalı,
- Uzman bilgisine dayalı.

Model tabanlı teknikler beklenmeyen bir arızanın anlık tespiti için sistemin dinamik modelinden faydalanır. Gerçek sistemden ölçülen ve benzetimi yapılan modelin özellikleri arasındaki fark, arıza tespit edici olarak kullanılır. Bu teknikler sistemin matematiksel modeline ihtiyaç duyarlar. Gerçek zamanlı fiziksel sistemlerde doğrusal olmayan davranışlardan dolayı sistemin matematiksel modeli elde edilmeden önce bazı varsayımlar yapılmalıdır. Bu tür bir arıza teşhis yönteminde sistemin modeli gürültü gibi nedenlerden dolayı yeterince sağlam değildir. Arıza teşhisinde model tabanlı yöntemlerin kullanılması, sistem modeli hakkında çok iyi bilgi gerektirmesinden dolayı her zaman mümkün olmayabilir [2]. Çünkü yeterli veya kesin matematiksel modellerin bilinmemesi durumunda sistem durum değişkenlerinin yanlış tahmininden dolayı yanıltıcı sonuçlar elde edilebilir. Model tabanlı bir arıza teşhis şemasının gerçekleştirilmesi için gerekli adımlar Şekil 1.3 (a)'da verilmiştir.

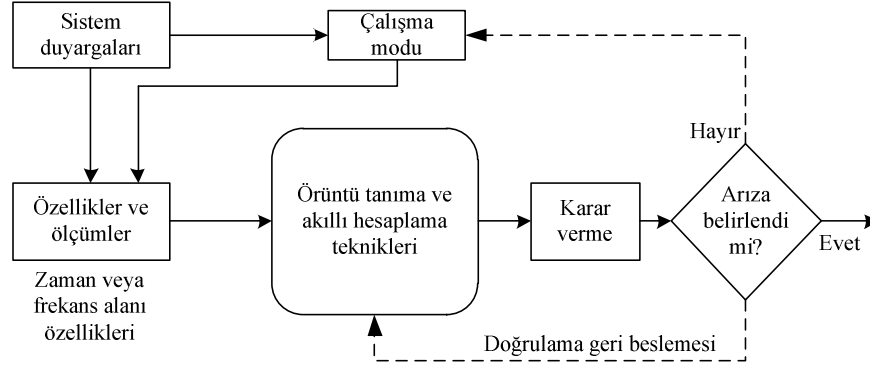
Deneyimli bir mühendis sistemin çalışma performansını sürekli gözlemleyerek arızaları kolayca tespit edebilir [3]. Bu çözüm analitik yöntemlerle karşılaştırıldığında hem daha kolay hem de daha kullanışlıdır, ancak gerçekleştirilmesi zordur. Uzman bilgisi ve deneyimine dayalı arıza teşhis şeması Şekil 1.3 (b)'de verilmiştir.

Veriye dayalı arıza teşhis yönteminde makine modeli ve parametreleri hakkında herhangi bir bilgiye gereksinim yoktur. Bu yöntem sadece sinyal işleme ve kümeleme tekniklerine ihtiyaç duyar [8]. Gerçek sistemden örneklenen sinyallerden özelliklerin çıkarımı için ön işleme uygulanır. Daha sonra bu özellikler kullanılarak arızaların sınıflandırılması sağlanır. Elde edilen özelliklerin sınıflandırılması ve arıza teşhisi için örüntü tanıma ve akıllı hesaplama teknikleri kullanılmaktadır. Veriye dayalı arıza teşhis yönteminin blok şeması Şekil 1.3 (c)'de verilmiştir.



(a) Model tabanlı

(b) Uzman bilgisi



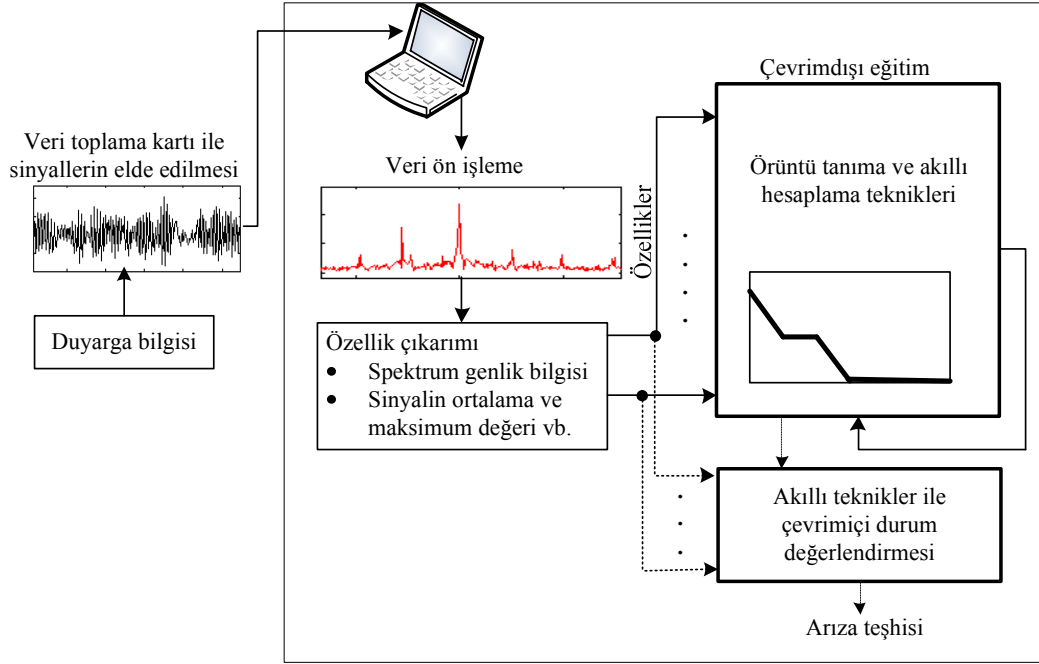
(c) Veriye dayalı

Şekil 1.3. Arıza teşhis yöntemlerinin çeşitleri

Her üç yöntemin ortak özelliği bir duyarga bilgisine ihtiyaç duymalarıdır. Ayrıca arıza ile ilgili özelliklerin veya çalışma sınırlarının iyi belirlenmesi de önemlidir. Veriye dayalı arıza teşhisinde en önemli bileşen özellik vektörünün elde edilmesi olup, bu vektör sistemin çalışması hakkında yeterli bilgiye sahip olmalı ve arızanın belirlenmesini sağlamalıdır.

Arıza teşhis yöntemleri uygulama biçimlerine göre çevrimiçi ve gerçek zamanlı olmak üzere ikiye ayrılmaktadır. Çevrimiçi yöntemde sistemden alınan ölçümler bir veri toplama kartı aracılığıyla bilgisayara aktarılır. Bilgisayar ortamında yazılan yazılımlar ile elde edilen veriler için eğitim veya model oluşturma aşaması gerçekleştirilir. Bu aşama genelde çevrimdışı olarak yapılmaktadır. Elde edilen model veya eğitim veri kümesi çevrimiçi

olarak sistemden alınan ölçümler ile değerlendirilerek arıza teşhisi yapılmaktadır. Veriye dayalı arıza teşhisi için bu yöntemin blok diyagramı Şekil 1.4'te verilmiştir.



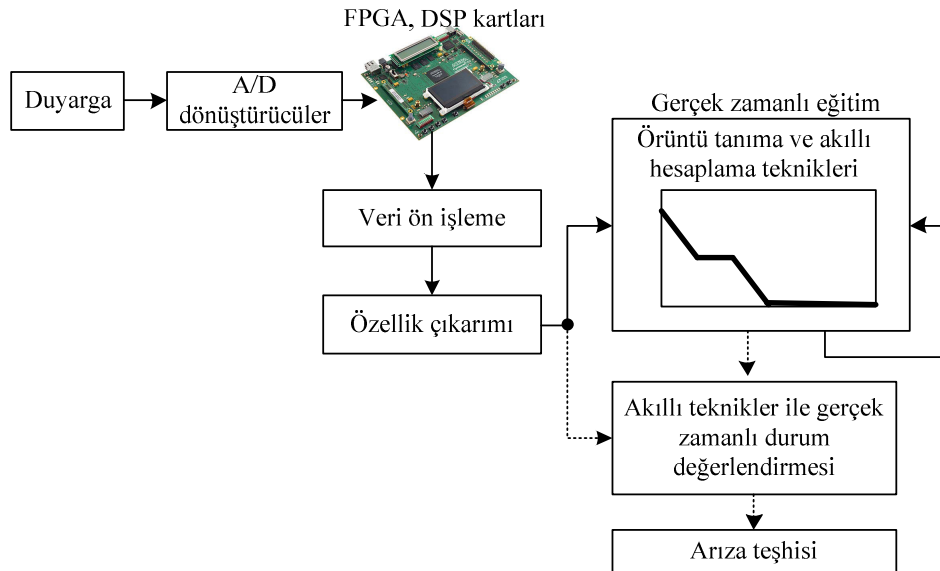
Şekil 1.4. Veriye dayalı arıza teşhis sisteminin eğitim ve test aşamaları

Şekil 1.4'teki çevrimdışı eğitim aşamasında ilk olarak duyargalar aracılığıyla alınan sinyaller veri toplama kartı ile bilgisayara aktarılır. Burada alınan sinyallerin belirli bir çözünürlükte alınması arıza teşhisi için önemlidir. Veri toplama kartı ile elde edilen sinyaller bilgisayara kaydedilir. Bu sinyaller üzerinde zaman ve frekans analizi ile bazı istatistiksel ve işaret işleme tabanlı özellikler elde edilir. Her bir durum için elde edilen özellikler gruplandırılarak örüntü tanıma ve akıllı tekniklerin eğitim verileri oluşturulur. Veri kümesi akıllı tekniklere verilerek eğitim hata oranı belli bir seviyenin altına düşünceye kadar eğitim işlemi gerçekleştirilir. Bu aşama çevrimdışı olarak yapılmaktadır. Test aşamasının özellik çıkarımına kadar olan kısımları, eğitim aşamasındakine benzerdir. Bu aşamada elde edilen özellikler ile eğitim aşamasında elde edilen akıllı tekniklerin parametreleri kullanılarak durum değerlendirme yapılır ve arızalar teşhis edilir.

Bir sistemin durumu model tabanlı veya veriye dayalı teknikler için bir yazılımın gerçekleştirilmesi ile çevrimiçi izlenebilir. Bu yöntem, herhangi bir şekilde sistem durumu hakkında uzman bilgisi ve yorumuna ihtiyaç kalmadan arıza teşhisinin yapılmasını sağlar.

Fakat gerçekte bu yöntemlerin bilgisayar yazılımına, veri toplama kartına ve bunlar arasındaki bağlantılara olan gereksinimlerinden dolayı endüstride pratik uygulanması zordur.

Uygulama şekline göre ikinci yöntem gerçek zamanlı arıza teşhisidir. Sinyallerin gerçek zamanlı olarak anlık ölçümü, sinyal işleme ve özellik çıkarımı erken bir arıza teşhis sistemi için oldukça önemlidir. İşaret işleme için yüksek performanslı donanımlar bilgisayar sistemlerinin gelişmesine paralel olarak artmaktadır [9]. Uygulama için özellikle sayısal işaret işlemciler (DSP) ve sahada programlanabilir kapı dizileri (FPGA) uygun seçeneklerdir. Bu teknolojiyi kullanan alanlar temel olarak sayısal işaret işleme, görüntü işleme, çoklu ortam uygulamaları, yönlendirici ve anahtarlama gibi yüksek hızlı iletişim ve ağ donanımlarından oluşmaktadır. Donanımsal sinyal işleme birimleri olarak da adlandırılan bu aygıtlar ile düşük maliyetli, çevrimiçi ve basit şekilde çalışan arıza teşhis yöntemleri geliştirilebilir. Bu tür sistemlerde, bilgisayar ortamında gerçekleştirilen arıza teşhis yazılımları donanıma yüklenerek istenilen sisteme monte edilebilir. Diğer bir avantaj ise bu donanımların yüksek çalışma frekansları ve paralel bilgi işleme yeteneklerinden dolayı birden fazla algoritmanın aynı anda çalıştırılabilmesidir. Dolayısıyla gerçekleştirilecek böyle bir sistemde veri toplama kartı ve bilgisayar gibi donanımlara ihtiyaç kalmaz. Şekil 1.5'te veriye dayalı arıza teşhisinin gerçek zamanlı gerçekleştiriminin blok diyagramı verilmiştir.



Şekil 1.5. Veriye dayalı arıza teşhisinin gerçek zamanlı gerçekleştirilmesi

Şekil 1.5 ile verilen gerçekleştirme model tabanlı arıza teşhisi için de kolaylıkla uyarlanabilir. Model tabanlı gerçekleştirmede yapılması gereken yöntemin donanımsal olarak gerçekleştirilmesi ve duyargalar aracılığıyla sistemden alınan sinyallerin çıkışları ile karşılaştırılmasıdır. Bu gerçekleştirmenin bir öncekinden farkı bütün sistemin tamamen donanımsal olarak gerçekleştirilebilmesidir. Veri toplama kartı ve bilgisayar gibi ekstra donanımlara ihtiyaç duyulmaz. Ayrıca bu tür donanımların yeniden programlanabilmesi özelliği göz önüne alındığında donanımı değiştirmeden farklı uygulamalar için yazılımlar geliştirilip tekrar sisteme yüklenerek kullanılabilirler.

1.2. Bakım ve Durum İzlemenin Gerekli Olduğu Sistemler

Sürekliliğin önemli olduğu sistemlerin anlık arızaya uğraması veya bakım amacıyla durdurulması, istenen bir durum değildir. Durum izleme ile endüstriyel sistemlerdeki iş gücünü oluşturan bileşenlerin sağlamlığının sürekli değerlendirilmesi ve bu bileşenlerin kalan ömürlerinin tahmini yapılabilir [10]. Koruma ve durum izleme bu bakımdan oldukça yakın kavramlar olmasına rağmen uygulama şekilleri farklıdır. Durum izleme, arıza oluşmadan önce tasarlanıp uygulanırken, koruma geriye dönük tasarlanır. Bazı durumlarda koruma amacıyla sistemlerin durumu izlenebilir. Fakat durum izlemenin asıl amacı, arıza oluşmadan önce arızanın gelişimini izlemek ve buna göre önlemlerin alınmasını sağlamaktır. Durumu izlenen sistemde arıza oluştuğundan sonra ilgili sistem bileşenleri için bir bakım süreci başlar. Bakım işlemi için genelde üç strateji izlenmektedir [11]:

- *Sistemin arıza oluşuncaya kadar çalıştırılması ve arıza oluştuğunda yenisi ile değiştirme:*

Bu yöntem bir endüstriyel sistemde iş yükünün büyük bir kısmını oluşturan ve oluşabilecek bir arıza durumunda sistemi aksatan bir bileşen için uygun değildir.

- *Belirli zamanlarda veya planlanmış bakım:*

Belirli zaman aralıklarında endüstriyel sistemlerdeki bileşenlerin incelenmesi ve aksaklıkların giderilmesi bir önceki yöntemle göre daha avantajlıdır. Fakat arızanın ne zaman oluşacağı ve ne kadar bir hızla gelişeceğini kestirmek zor olduğundan bu yöntem de pek uygun değildir.

- *Durum tabanlı bakım:*

Bu stratejide amaç sistem bileşenleri izleyerek erken bir aşamada arızaların tespiti ve tespit aşamasından sonra uygun bakımın yapılmasıdır. Kritik sistem bileşenlerinin izlenmesi için diğer iki yöntemden daha iyi sonuçlar verir. Bu yöntemde en önemli

kriterler sistemin izlenmesi için maliyetin minimum olması ve sistem durumu hakkında gerekli olan bilgilerin toplanmasıdır.

Bu aşamada bir sistemde izlenmesi gereken en önemli bileşenin ne olduğu ve arıza hakkında ne tür parametrelerin ölçülmesi gerektiği önemli iki problemdir. Endüstride iş gücünün %80 gibi büyük bir kısmını oluşturan elektrik makinaları [1, 5, 10], güç sistemlerinin güvenliğini sağlayan güç transformatörleri [12-13], rüzgar türbinlerindeki elektrik generatörleri [14-15], güneş enerjisinden elektrik üretimi sağlayan PV paneller [16] ve hava araçlarındaki motorlar [17] önemli sistem bileşenleri olup izlenmesi gerekmektedir.

Bu bileşenler arasında elektrik makinaları iş gücünün büyük bir kısmını karşılamaktadır. Elektrik makinaları modern toplumumuzun farklı ihtiyaçlarını karşılanmasında önemli rol oynamaktadır. Hem endüstriyel hem de ev aletleri ile ilgili modern yaşamımızın her alanına nüfuz etmiştir. Günlük faydalandığımız herhangi bir hizmette, kullandığımız veya tükettiğimiz birçok üründe üretime katkı sağlayan bir elektrik makinası vardır. Gelişen dünyada modern bir evde ısıtma sistemi, oyuncaklar, ev aletleri, saat ve hava izleme gibi sistemlerde 0-1 KW aralığında değişen elektrik makinalarını kullanmaktadırlar [1]. Modern araçlarda silecekler, pencereler ve hatta hibrit araçlarda arabanının hareketi için dahi elektrik makinalarından faydalanılmaktadır. Küçük uygulamalarda kullanılan elektrik makinelerinin izlenmesine ihtiyaç yoktur. Bu makinelerin bileşenleri elektrik makinesinin görevini yerine getirmesi için yeterince sağlam yapıdadırlar. Fakat artan yaşam standartlarını karşılamak için daha karmaşık ve büyük elektrik makinalarına ihtiyaç vardır. Her gün kullandığımız elektrik gücü 1000 MW'a kadar erişebilen elektrik santrallerinde üretilmektedir [5]. Bu santrallerde kullanılan pompalar, taşıyıcılar ve kompresörler elektrik makinaları ile bütünleşik çalışmaktadırlar. Kullanılan elektrik makinaları da büyük çapta olup güçleri 100 KW'ın üzerindedir. Ayrıca su dağıtım pompaları ve atık sistemlerde de elektrik makineleri kullanılır. Bütün bu özellikleri ile elektrik makinaları yüksek derecede karmaşık bir sistemin parçası olarak veya bağımsız bir şekilde kendi işlevlerini yerine getirirler. Bu tür büyük sistemlerde üretimin devamlılığı ve sistemin korunması için üretimde kullanılan her bir bileşen işlevini iyi bir şekilde yerine getirmelidir. Elektrik makinalarının güç-ağırlık oranı dizel, buhar ve gaz ile çalışan motorlara göre daha düşük olup daha güvenilirlerdir. Ayrıca elektrik makinaları için gerçekleştirilen arıza teşhis ve durum izleme sistemleri diğer sistemlerin izlenmesi için kolayca uyarlanabilmektedir.

İzlenecek olan parametreler, bakımın planlanması ve bileşenin çalıştığı fabrikanın güvenliğinin sağlanması açısından önemli detaylar sunmalıdır. Otomatik çevrimiçi izleme yöntemi son zamanlarda elektrik makinaları alanında bir etki oluşturmaya başlamıştır. Titreşim, akım, gerilim, sıcaklık gibi niceliklerin dışında dönme hızı ve akı durum izleme için en çok tercih edilen parametrelerdir.

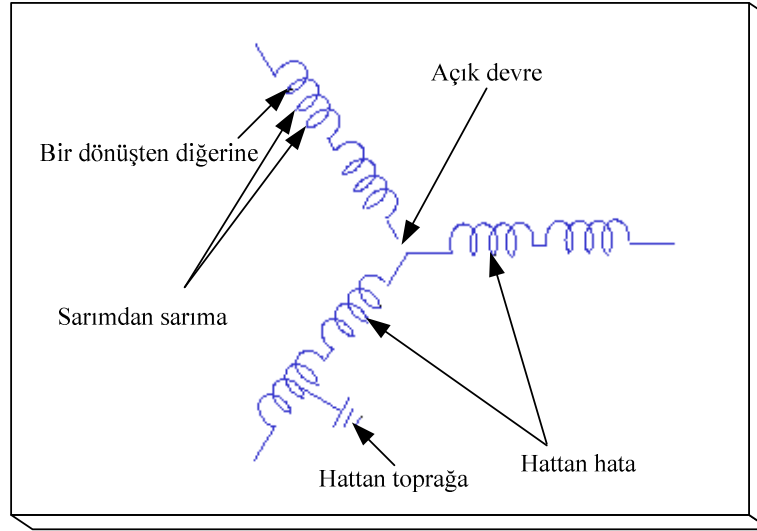
Endüstride iş gücünü sağlayan birçok kritik eleman olmakla birlikte, birçok fabrikanın iş gücünün büyük bir kısmını asenkron motorlar oluşturur. Dolayısıyla bir sonraki bölümde bu motor türleri için literatür özeti verilecek ve geliştirilen arıza teşhis ve durum izleme yöntemleri bu alana yönelik olacaktır. Bu tür bir makine için geliştirilen arıza teşhis ve durum izleme yöntemi, diğer elektrik makinaları ve transformatörler için kolaylıkla uyarlanabilmektedir.

1.3. Asenkron Motorlarda Oluşan Arızalar

Asenkron motorlarda arızalar, temel olarak motorun elektrik ve mekanik donanımlarında oluşur. Elektriksel arızalar motorun stator ve rotor bileşeni ile ilgilidir. Sargı arızaları ve dengesizlik gibi stator arızalarının yanı sıra, kırık rotor çubuğu ve kırık sonlandırıcı halka arızaları en çok rastlanan elektriksel arızalardır. Mekanik arızalar ise rotor ile stator arasındaki hava aralığının değişimi ile oluşan arızalar ve mil yatağı arızaları olmak üzere iki şekilde oluşabilir. Bu arızalar akım, gerilim ve diğer motor sinyalleri üzerinde çeşitli etkiler oluşturur. Motorda oluşan arızalar bu sinyallerden biri veya birkaçını etkiler. Herhangi bir arıza oluştuğunda hat akımları ve hava aralığı geriliminde dengesizlikler oluşur. Moment titreşimleri artarak, ortalama moment düşer. Verimlilik azalır ve motor aşırı ısınır.

1.3.1. Stator Arızaları

Stator arızaları; asenkron motorlarda en yaygın görülen arıza türlerinden biridir. Tipik olarak stator sargılarındaki kısa devreler, bir fazın, iki fazın veya bütün fazların dönüşleri arasında oluşabilir. Ayrıca stator çekirdeği ve sarım iletkeni arasında da kısa devreler oluşur. Stator arızalarının farklı tipleri Şekil 1.6'da gösterilmiştir. İlk aşama aynı sargıdaki dönüşler arasındaki kısa devre arızalarıdır. Eğer bu arızalar erken bir aşamada tespit edilebilirse daha ciddi arızaların oluşumu engellenebilir. Stator arızaları sonucunda statorun yeniden basit bir şekilde sarılması gerekebilir. Bu şekilde aksama süresi önemli bir oranda azaltılabilir.



Şekil 1.6. Stator kısa devre arızaları

Bir asenkron motorun stator sargısı sargı devirlerinin kısa devre olmasını önlemek amacıyla yalıtım yapılmıştır. Bu yalıtım maksimum çalışma sıcaklıkları ile orantılı olarak farklı sınıflara ayrılmıştır. Fakat bu maksimum çalışma sıcaklıklarına çalışma ortamları ve yük durumlarından dolayı uyulmayabilir. Bu ihmalden dolayı stator sargılarının belirli kısımlarında incelme, yalıtım kayıpları ve çatlaklar oluşur. Bu durum stator sargılarının kısa devre olmasına ve stator sarım sayılarının düşmesine yol açar. Sargı arızaları genellikle aşırı ısınma, gevşek çekirdek levhaları ve bağlantıları, yağ ve nemden dolayı sargıların kirlenmesi, motor yalıtımındaki kusurlardan dolayı oluşan elektriksel boşalmalar ve soğutma sistemlerindeki kaçaklardan dolayı oluşur [18].

1.3.2. Kırık Rotor Çubuğu Arızaları

Bir rotor çubuğunda oluşan arızalar iki şekilde olabilir. Bunlar rotor çubuğunun tamamen kırılması veya rotor çubuğunda oluşan çatlaklardır. Bu arızalar genellikle mil yatağı arızaları ile oluşan mekanik zorlanmalar, motorun sıklıkla durdurulması, aşırı sıcaklık ve üretim esnasında oluşan hasarlardan oluşmaktadır [10]. Şekil 1.7’de üç adet kırık rotor çubuğuna sahip bir motorun rotor kısmı görülmektedir.



Şekil 1.7. Üç adet kırık rotor çubuğuna sahip rotor

Bir rotor çubuğu tamamen kırılmadan önce çatlağın oluştuğu nokta aşırı ısınır. Kırığın oluştuğu noktada bir ark meydana gelir. Kırık rotor çubuğu üzerinden akım geçirmediği için sağında ve solunda bulunan rotor çubukları daha fazla akım çeker. Rotor arızalarından dolayı oluşan merkezkaç kuvvetinden dolayı stator sargı arızaları oluşur.

1.3.3. Mil Yatağı Arızaları

Mil yatağı arızaları yaklaşık olarak bütün makine arızalarının yarısını oluşturur. Mil yatağı hasarlarını azaltmak için elektrik makinelerinde mil yatağı durumlarının izlenmesi pratik öneme sahiptir. Mil yatağının eğilmesi, milin sürtünmesi, mil yatağındaki bilyelerde oluşan arızalar en çok karşılaşılan mil yatağı arıza türleridir. Mil yatağının eskimesi, yağ sızıntısı ve yağın eskimesi ile mil yatağında sürtünme arızaları oluşur [5]. Oluşan bu sürtünmeler erken bir aşamada giderilmezse mil yatağının kilitlenmesi ve motorun yanmasına sebep olur. Mil yatağında bilyeler iç ve dış halka olmak üzere iki halka içinde bulunmaktadır. Bu halkalar içinde yerleştirilmiş olan bilyeler motorun dönmesi ile dönerler. Mil yatağı bilyeleri rotora destek oluşturduğundan burada oluşacak herhangi bir arıza stator ve rotor arasında bir zorlanma oluşturur [8].

Motorun yataklama sistemini oluşturan rulmanlardaki herhangi bir bozukluk, hava boşluğu akısında anormalliklere sebep olur ve makinanın statoru ile rotoru arasında radyal bir hareket üretir. Bu da mil yatağı aşınması ve arızalarını meydana getirir.

1.4. Literatür Özeti

Elektrik makinalarında oluşan arızalar, motor yapısını çok fazla bozmadan tespit edilebildiğinde uygun bakımın yapılması mümkün olur. Son zamanlarda ekonomik olması ve verimliliklerinden dolayı akıllı arıza teşhis yöntemlerine ilgi artmıştır. Daha güvenilir, etkili ve maliyetli koruyucu bakım ve arıza teşhis sistemlerine sahip olmak bugünkü küresel ekonomik ortam için rekabet sağlar.

Asenkron motorlarda da arıza teşhisi için model tabanlı ve veri işlemeye dayalı yöntemler kullanılmıştır. Model tabanlı yöntemlerde motorun modeli oluşturularak çalışan motordan alınan akım, gerilim ve hız gibi sinyaller ölçülür. Modelin çıkışı ile ölçülen sinyaller arasındaki farklar arıza tespit indeksi olarak kullanılır [18]. Sağlam ve arızalı motor modelinin davranışlarının modellenmesi için yapay sinir ağları ve Bayesian modellemelerden faydalanılmıştır [19]. Stator arızalarının tespiti için oluşturulan yapay sinir ağı modeli gerçek zamanlı olarak uygulanmıştır [20]. Veri işlemeye dayalı arıza teşhisinde motordan sinyaller alınarak sinyal işleme yöntemleri ile belirli özellikler elde edilir. Bu özelliklerin sağlam ve arızalı durumlarda davranışları farklılık gösterir. Arıza teşhisi için literatürde en çok tercih edilen sinyal işleme yöntemleri üç temel sınıfa ayrılabilir: Bunlar, spektrum tahmin teknikleri, zaman-alan analizi ve zaman-frekans analizidir [21].

Spektrum tahmin teknikleri Hızlı Fourier Dönüşümü (FFT), anlık güç frekans dönüşümü, ikili spektrum analizi, yüksek çözünürlüklü spektrum analizi ve park vektör dönüşümü tabanlı spektrum analizi yöntemlerinden oluşmaktadır.

- *Hızlı Fourier dönüşümüne* dayalı motor akım imza analizi (MAIA), kırık rotor çubuğu arızalarını teşhis etmek için en çok tercih edilen yöntemdir. Bu yöntem motor akımı ve hızı giriş olarak alır [22]. Alınan stator akımı, frekans spektrumunda incelenerek belirli frekanslardaki genliklerin büyüklüklerine göre arıza hakkında bilgiler elde edilir. Bu frekanslar hat frekansından büyük ve küçük ilk yan bantlar olarak adlandırılır. İlk olarak stator akımının Hızlı Fourier Dönüşümü (HFD) alınır. Frekans ve motor hızından elde edilen kayma değeri ile sağ ve sol yan bantlar bulunur. Sol yan bant bir eşik değeri ile karşılaştırılarak arıza olup olmadığı belirlenir. Motor akım imza analizi hava aralığı eksantrikliği, stator arızaları ve mil yatağı arızaların tespiti için de uyarlanmıştır [23]. Fourier yaklaşımı sinyallerin durgun olduğu birçok uygulama için çok kullanışlıdır. Fakat frekans eğilimleri, ani değişimler ve sapmalar gibi geçici bir karakteristiğe sahip bir sinyali analiz etmek için uygun değildir.

- *Anlık güç frekans dönüşümü* MAIA için bir ortam olarak görülür. Anlık güç dönüşümü ile elde edilen spektrumda hat frekansı ve iki yan bant bileşeni dışında modülasyon frekansında ek bir bileşen içerir [24]. Fourier dönüşümüne göre avantajı arıza ile ilgili harmoniklerin daha belirgin olmasıdır. Fakat güç spektrumunda gürültünün etkisi tam giderilemez. Gerilim dengesizliği ve açık faz arızalarının tespiti için kullanılmıştır [25].
- *İkili spektrum analizi* bir motorun moment sinyalinin iki boyutlu Fourier dönüşümü olarak ifade edilir. Periyodik bir dönüşüm olup hem büyüklük hem de faz bilgisinin korunmasını sağlar. Bu iki bilginin korunması sayesinde arıza tespiti ve teşhisinin performansı artırılmıştır. Bu yöntem stator gerilim dengesizliği gibi elektriksel arızaların teşhisi için daha uygun olmasına rağmen mekanik arızaların teşhisi için de kullanılmıştır [26-27].
- *Yüksek çözünürlüklü spektrum analizi* çoklu sinyal sınıflandırması ve öz vektör gibi teknikleri içerir. Durgun olmayan sinyaller üzerinde bu iki teknik uygulanmıştır [28]. Parametrik olmayan bu yöntemler geleneksel Fourier dönüşümündeki frekans çözünürlüğü ile ilgili problemleri çözemez. Frekans kestirimi sayesinde hızlı Fourier dönüşümünün performansı artırıldığından arızaların teşhis doğruluğu da artırılmıştır. Fakat arıza teşhisinde etkili yeni frekanslar bu yöntem ile bulunamamaktadır.
- *Park vektör dönüşümü tabanlı spektrum analizinde* öncelikle üç faz bilgisinden iki boyutlu bir vektör uzayı elde edilir. Daha sonra bu elde edilen iki boyutlu vektörün mutlak değerinin Fourier dönüşümü alınarak frekans bileşenleri arıza teşhisi için incelenir. Bu yöntem kırık rotor çubuğu arızalarının teşhisi için kullanılmıştır [29]. Haji ve Toliyat [30] kırık rotor çubuğu ve eksantriklik arızalarını tespit etmek için park vektör dönüşümü ve motor akım imza analizini birleştiren bir yöntem önermiştir.

Zaman Alan Analizi geçiş sinyallerinden arıza teşhisi yapmak için oldukça kullanışlıdır. Bu teknik hem düşük hesaplama maliyeti hem de daha az veri ile arıza teşhis işlemini gerçekleştirir. Geçiş anında arızaların teşhisi elektrik makinası yüksüz çalıştığında bile arıza ile ilgili frekans bileşenlerini elde etmek mümkün olduğundan oldukça kullanışlıdır. Bu yöntem mekanik arızaların teşhisi için zaman domaininde elektrik güç salınımını kullanmıştır [31]. Maksimum kovaryans metodu zaman domaininde ölçülen sinyal ile referans sinyal arasındaki kovaryansı inceleyerek arıza teşhisinde kullanılmıştır

[32]. Bu yöntem arıza teşhisi için iyi bir frekans çözünürlüğü ve geniş frekans aralığı istendiğinde uzun zaman alır.

Zaman Frekans Analizi bir sinyali zaman, genlik ve frekans boyutlarında incelemek için kullanılmaktadır. Bu yöntemin bileşenleri Wigner dönüşümü, Hilbert dönüşümü, kısa süreli Fourier dönüşümü ve dalgacık dönüşümüdür.

- *Wigner dönüşümü* geçiş durumlarında mekanik yük arızalarının tespiti için kullanılmıştır [33]. Mekanik yük arızaları mekanik rotor hızıyla ilgili frekanslarda yük moment salınımlarıyla ilgilidir. Mekanik arızaların teşhisi ve bir elektrik makinasının kalan ömrünün tahmini için wigner dönüşümü diğer yöntemler ile karşılaştırılmıştır [34].
- *Dalgacık dönüşümü* yüksek frekans bilgisini istediğimiz daha kısa alanlar ve daha keskin alçak frekans bilgisi istediğimiz uzun zaman aralıklarının kullanımına izin verir. Ayrık ve sürekli dalgacık dönüşümü motor akımı, titreşim bilgisi, motor gerilimi gibi sinyaller üzerinde uygulanmıştır. Bu dönüşüm sonucunda elde edilen özellikler ile mil yatağı, kırık rotor çubuğu ve stator arızaları gibi arızalar teşhis edilmiştir [35-39].
- *Hilbert dönüşümü* reel değerlere sahip bir zaman serisinden karmaşık değerlikli bir seri üretir. Bu dönüşümün avantajı negatif bileşenlerin filtrelenmesini sağlamasıdır. Akım ve titreşim sinyallerini üzerinde Hilbert dönüşümünün uygulanması ile kırık rotor çubuğu ve mil yatağı arızalarının teşhisi sağlanmıştır [40-41].

İşaret işleme tabanlı arıza teşhis yöntemlerinin ortak amacı her bir arıza türü ile ilgili bir tanımlayıcı indeks bulmaktır. Fakat elde edilen bu sinyallerin değerlendirilmesi için de ayrıca uzmana ihtiyaç vardır. Arıza teşhis yöntemlerinin amaçlarından biri de karar verme aşamasında uzman gerekliliğini minimuma indirmektedir. Bu yüzden son zamanlarda geliştirilen arıza teşhis yöntemlerinin çoğu son aşamada karar vermek için akıllı hesaplama tekniklerini kullanmaktadır. Bu tekniklerin kullanımı arıza teşhis yönteminin etkinliğini artırmaktadır. Literatürde elektrik motorlarında arıza teşhisi için en yaygın kullanılan akıllı hesaplama teknikleri yapay sinir ağları, bulanık mantık, yapay bağışık sistemler, destek vektör makinalar, Bayesian sınıflandırma ve iki veya daha fazla akıllı hesaplama tekniğinin birleşimi ile oluşturulan hibrit yöntemlerdir.

Yapay sinir ağları çoklu giriş-çıkış haritalama ve doğrusal olmayan fonksiyonları gösterebilme gibi özelliklerinden dolayı arıza teşhisinde geniş bir şekilde kullanılmıştır [5, 35, 42-48]. Arıza teşhisi için en çok tercih edilen yapay sinir ağı modeli ileri beslemeli

yapay sinir ağı olmasına rağmen Kohonen ağları ve geri beslemeli yapay sinir ağları da arıza teşhisine uygulanmıştır. Yapay sinir ağlarının arıza teşhisinde kullanım yerleri aşağıda verilmiştir:

- Arıza teşhisi için doğrusal olmayan haritalama, parametre tahmini ve örüntü tanıma,
- Simülasyon veya deneysel olarak elde edilen sinyaller üzerinde eğitim,
- Çevrimiçi eğiticişiz teşhis,
- Gürültü ve çıkıştaki dengesizliklerin filtrelenmesi,
- Anomali çalışmalara sahip başlama aşamalarında arıza kestirimi,
- Arıza türlerine göre çalışma durumlarının kümelenmesi.

Bulanık sistem kullandığı bulanık kurallar sayesinde dilsel değişkenleri işleyebilir. Bu özelliğinden dolayı arıza teşhisinde en çok tercih edilen akıllı hesaplama tekniklerinden biridir. Adaptif bulanık sistemler önceki bilgiye göre akıllı sistem performansını geliştirmek için parametreleri ayarlayarak genetik algoritmanın optimizasyon gücünü ve yapay sinir ağlarının öğrenme yeteneklerini kullanır. Bulanık ve adaptif bulanık sistemler arıza teşhisinde aşağıdaki görevler için kullanılmıştır [5, 49-55].

- Dilsel değişkenleri kullanarak performans indislerinin değerlendirilmesi,
- Arızalı bileşenin yerinin tespiti,
- Anomali çalışmaların kestirimi,
- Bulanık kurallar ile uzman bilgisinden faydalanılması,
- Adaptif bulanık sistemler aracılığıyla arıza teşhisinin optimizasyonu, sistem modelleme, doğrusal olmayan haritalama,
- Arıza sınıflandırma ve tahmin.

Destek vektör makinalar kullandıkları çekirdek fonksiyonları ile farklı boyutlardaki veriler arasında doğrusal olmayan ilişkileri kolaylıkla öğrenebilirler. İki sınıflı sınıflandırma özelliği olan bu yöntemin parametreleri bağışık sistemler ve parçacık sürü optimizasyonu ile ayarlanarak arıza teşhisinde kullanılmıştır [38, 56-58]. İşaret işleme ile elde edilen özelliklerin çok sınıflı sınıflandırılması için destek vektör makinalardan faydalanılmıştır [59]. Arıza teşhis alanında çok boyutlu özellik uzayında sınıflandırma işlemlerini kolaylaştırdığından birçok araştırmacı tarafından arıza hakkında karar verme aşaması için destek vektör makinalar tercih edilmiştir.

Yapay bağışık sistem bileşenlerinden olan negatif seçim ve klonal seçim algoritmaları arıza teşhisi için kullanılmıştır. Negatif seçim algoritmasında arızalı durumlar anomali olarak alınıp sağlam ve arızalı durumlar kolayca ayırt edilebilir. Yapılan çalışmaların büyük bir kısmında sadece arıza belirleme işlemi yapılmıştır [60-62].

Bayesian sınıflandırma arıza teşhisi ile ilgili motor sinyallerinin sinyal işlemesi ile elde edilen özelliklerinin dağılımına dayalı bir sınıflandırma yöntemidir. Bayesian sınıflandırma motor akım imza analizi ve park vektör dönüşümü ile elde edilen özellikler ile kırık rotor ve eksantriklik arızalarının tespitinde kullanılmıştır [30]. Ayrıca motor sinyallerinin doğrusal olmayan zaman serileri analizi ile oluşturulan faz uzayında Gaussian karışım modeli ile elde edilen motor durumlarının sınıflandırılmasında faydalanılmıştır [63-64]. Kullanılan bu yöntemler aynı zamanda zaman serisi veri madenciliği olarak ta adlandırılmıştır.

Literatürde önerilen akıllı hesaplama tekniklerine dayalı arıza teşhis yöntemlerinin ortak özelliği hepsinin bir sinyal ön işlemeye ihtiyaç duymasındır. Ön işleme adımı ile elde edilen özellikler akıllı hesaplama teknikleri tarafından alınarak motorun farklı durumları öğrenilmektedir. Elde edilen özelliklere göre farklı akıllı hesaplama teknikleri kullanılabilir. Eğer çok boyutlu özellik sinyalleri üzerinde işlem yapılıyor ve özellikler arasında doğrusal olmayan bir bağıntı var ise yapay sinir ağları ve destek vektör makinalar tercih edilmektedir. Yapay bağışık sistemler sınırlı sayıda özelliklere göre sağlam motor durumu dışında bir durumun tespitinde kullanılmıştır. Sinyal büyüklükleri ve Fourier dönüşümü ile elde edilen spektrumda belirli yan bant bileşenlerinin büyüklüklerine göre arızaların belirlenmesi için bulanık mantık oldukça kullanışlıdır. Ayrıca yapı olarak bulanık mantığa benzemesine rağmen sadece belirli özelliklerin her bir sınıf için değer aralıklarını beyazlaştırma fonksiyonlarına göre inceleyerek sınıflandırma işlemi yapan gri kümeleme sistemleri de arıza teşhisi için yeni bir yaklaşım olabilir.

Elektrik makinalarında arıza teşhisi için literatürde kullanılan sinyal işleme ve hesaplama tekniklerinin arıza türlerine göre sınıflandırılması Tablo 1.1’de gösterilmiştir.

Tablo 1.1. Akıllı hesaplama ve sinyal işleme tabanlı arıza teşhis yöntemlerinin literatür sınıflandırılması

Arıza Türü	Giriş Sinyali	Sinyal işleme yöntemi	Kullanılan hesaplama teknikleri	Kaynaklar
Rotor	Stator akımı Rotor hızı	Frekans analizi	- Yapay sinir ağları ve bulanık mantık [22] - Bayesian sınıflandırma [30] - Yapay sinir ağları [44, 46, 49] - Genetik algoritma [46]	[22-23, 30, 40, 43-46]
Rotor	Stator akımı	Yüksek çözünürlüklü yöntemler		[28]
Rotor	Stator akımı Gerilim Rotor hızı	Zaman frekans analizi	- Yapay sinir ağları [35] - Destek vektör makinalar [39] - Dalgacık tabanlı destek vektör makinalar [56]	[33, 35-36, 39-41, 56]
Rotor	Moment	Frekans analizi		[65-66]
Rotor	Stator akımı	Sinyal enjeksiyonu		[67-69]
Stator	Akı	Frekans analizi		[70]
Stator	Stator akımı	Frekans analizi	- K-ortalama metodu [34] - Yapay sinir ağları [35, 46] - Sinirsel Bulanık sistem [51] - Bulanık mantık [55] - Yapay bağışık sistemler [61]	[4-5, 7, 11, 18-19, 29, 34, 38, 44]
Mil yatağı	Stator akımı	Frekans analizi	- Yapay sinir ağları [35] - Sinirsel Bulanık sistem [51]	[35, 37, 51]
Mil yatağı	Titreşim	Zaman-Frekans analizi	- Yapay sinir ağı [25, 47] - Yapay sinir ağları ve destek vektör makinalarının performans karşılaştırması [54] - Bağışık sistem tabanlı destek vektör makinalar - Çok sınıflı destek vektör makinalar - Klonal seçim tabanlı arıza sınıflandırıcı	[25-26, 41, 47, 54, 57, 59, 62]
Hibrit arızalar	Stator akımı	Park vektör dönüşümü	- Bayesian sınıflandırma - Eğitici yapay sinir ağı - Bulanık çıkarım sistemi - Bağışık sistem tabanlı destek vektör makinalar	[30, 48-49, 58]
Hibrit arızalar	Stator akımı	Zaman serileri analizi	- Genetik algoritma tabanlı bağışık sistem [60] - Bayesian sınıflandırma [63-64]	[60, 63-64]
Hibrit arızalar	Stator akımı Rotor hızı Moment	Sinyal büyüklükleri	- Yapay sinir ağı [5] - Bulanık mantık [5, 46, 52-53] - Bulanık sinirsel model [50]	[2, 46, 50, 52-53]

Tablo 1.1'den görülebileceği gibi literatürde geliştirilen arıza teşhis yöntemleri genelde iki yönde ilerlemiştir. Birincisi arızaları belirlemek için yeni arıza indekslerinin bulunması ile ilgilidir. İkinci yön ise arızalar ile ilgili indekslerin değerlendirilmesi ve arıza teşhisinin akıllı hesaplama teknikleri ile gerçekleştirilmesidir. Buna karşın elektrik makinalarının arıza teşhisi ile ilgili eksiklikler şu şekilde sıralanabilir:

- Geliştirilen arıza teşhis yöntemleri motor türüne bağımlıdır.
- Arıza teşhis yöntemleri gürültü, yük ve motor çalışma durumlarından kolayca etkilenmektedir.
- Geçiş durumlarında dalgacık dönüşümü gibi yöntemler iyi sonuç vermesine rağmen, arıza teşhisi için her defasında motorun yeniden başlatılması gerekir.
- Literatürdeki yöntemlerin çoğu arıza teşhisi için makinadan alınan sinyallerin belirli periyotlarda incelenmesine dayalıdır. Bazen arıza izleme aşaması saatlerce hatta günlerce devam edebilmektedir. Fakat arızaların ne zaman oluşacağını kestirmek zor olduğundan bu yöntemler pratik uygulamalar için uygun değildir.
- Akıllı hesaplama tekniklerini kullanan arıza teşhis yöntemleri genellikle eğitim ve test olmak üzere iki aşamadan oluşmaktadır. Eğitim aşaması çevrimdışı olarak yapıp motor durumları öğrenildikten sonra test aşaması çevrimiçi olarak yapılmaktadır. Fakat test aşamasında da her defasında motor sinyalleri bilgisayar ortamına aktarılıp motor durumu izlenmektedir.

1.5. Tezin Amacı ve Kapsamı

Bu tezde arıza belirleme, teşhis ve tahmini için akıllı hesaplama teknikleri kullanan algoritmalar geliştirilerek, bunların gerçek zamanlı uygulamaları yapılmıştır. Bu amaçla temel olarak insanın doğal bağışık sisteminden esinlenilerek geliştirilen bir yumuşak hesaplama tekniği olan yapay bağışık sistem algoritmalarından faydalanılmıştır. Arıza teşhis ve tespitinin performansının artırılması için yapay bağışık algoritmalarına yeni eklentiler yapılmıştır. Ayrıca diğer yumuşak hesaplama tekniklerinden olan destek vektör makinalar, parçacık sürü optimizasyonu, genetik algoritmalar, yapay sinir ağları ve bulanık mantık arıza teşhisi için kullanılmıştır. Önerilen algoritmaların performansları literatürdeki diğer yöntemler ile karşılaştırılmış ve bu yöntemlerle yakın veya daha iyi sonuçlar alındığı gösterilmiştir.

Önerilen arıza belirleme, teşhis ve tahmin yöntemlerinin gerçek zamanlı olarak uygulanması için FPGA kartı kullanılmıştır. Aynı anda birden fazla algoritmanın gerçek zamanlı olarak çalıştırılabilmesi ve farklı türden arızaların teşhisi de FPGA üzerinde gerçekleştirilmiştir. Geliştirilen algoritmalar ile genel arıza teşhis yöntemlerine aşağıdaki katkılar getirilmiştir.

- Arızadan ve motor tipinden bağımsız olmaları,
- Gerçek zamanlı olarak gerçekleştirilebilmeleri,
- Geleneksel tekniklerle birlikte kullanılabilir olmaları,
- Düşük maliyetli ve gürültü bağıışıklığına sahip olmaları,
- Az bilgi ile hızlı cevap verebilmeleri.

Bu tezin kapsamı;

- Öncelikle arıza belirleme için sadece makinada arıza olup olmadığına karar veren yapay bağıışık sistem bileşenlerinden olan negatif seçim algoritması sunmak,
- Negatif seçim algoritmasının performansının geliştirilmesi için klonal seçim, genetik algoritma ve K-en yakın komşu algoritmasını kullanmak,
- Arıza belirleme için yeni bir özellik uzayı sunmak,
- Arıza teşhisi için destek vektör makinaları ve yapay bağıışık sistem tabanlı örüntü tanıma yöntemlerinin geliştirilmesi ve bu amaçla klonal seçim ve parçacık sürü optimizasyonundan faydalanmak,
- Arıza tahmini için entropi tabanlı yeni bir özelliği elde etmek ve önerilen yöntemi motor akım imza analizi ile doğrulamak,
- Geliştirilecek arıza teşhis ve tahmin yöntemlerini donanımsal olarak birlikte çalıştırmak ve bu amaçla FPGA tabanlı paralel çalışan algoritmalar geliştirmektir.

1.6. Tezin Yapısı

Tez *giriş bölümü* ile birlikte yedi bölümden oluşmaktadır.

İkinci bölümde, veriye dayalı durum izleme ve arıza teşhisinin temel bileşenleri açıklanmış ve özellik çıkarımı için kullanılan sinyal işleme yöntemleri özetlenmiştir.

Üçüncü bölümde, arıza belirleme amacıyla yapay bağıışık sistem bileşenlerinden iki farklı negatif seçim algoritması önerilmiştir. Negatif seçim algoritmasının detektörlerinin üretimi ve optimizasyonu için genetik algoritma, klonal seçim, K-en yakın komşu

algoritması ve kaotik haritalardan faydalanılmıştır. Bu şekilde negatif seçim algoritmasının detektörlerinin öz olmayan uzayı maksimum kapsamaları sağlanmıştır. Aynı zamanda arıza teşhisi için yeni bir özellik çıkarımı yapılarak elde edilen sonuçlar literatürde var olan yöntemler ile karşılaştırılmıştır. Bu bölüm ile ilgili yapılan akademik yayın çalışmaları aşağıda verilmiştir [71-75]:

- “Artificial Immune Inspired Fault Detection Algorithm Based on Fuzzy Clustering and Genetic Algorithm Methods”, IEEE Computational Intelligence for Measurement System and Applications (CIMS2008), pp. 93-98, Istanbul, Turkey, 2008.
- “K-NN Based Hybrid Negative Selection Algorithm for Anomaly Detection”, Innovations on Intelligent Systems and Technology Applications (INISTA), 28-30 June, Trabzon, Turkey, 2009.
- “Genetik Algoritma Kullanan Yapay Bağışık Sistem Tabanlı Arıza Teşhis Modeli”, Dokuz Eylül Üniversitesi Fen ve Mühendislik Dergisi, Cilt 11, Sayı 31, Sayfa: 57-72, 2009.
- “Kırık Rotor Çubuğu ve Stator Arızalarının Teşhisinde Yapay Sinir Ağı Yaklaşımı”, Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi, Cilt 25, No.1-2, S. 134 – 149, 2009.
- “Chaotic-based Hybrid Negative Selection Algorithm and Its Applications in Fault and Anomaly Detection”, Expert System with Applications, Vol. 37, No. 7, pp. 5285-5294, 2010.

Dördüncü bölümde, arıza teşhisi için destek vektör makinalar ve yapay bağışık sınıflandırıcı kullanan üç yöntem sunulmuştur. Destek vektör makinaların çekirdek ve ceza parametreleri çok amaçlı klonal seçim yöntemi ile elde edilmiştir. Arıza teşhisi için park vektör dönüşümü üzerinden yeni bir özellik sinyali oluşturularak faz uzayları sınıflandırılmıştır. Rotor ve stator arızaları için uyarlamalı bir bağışık sistem algoritması geliştirilerek yapay bağışık sistemlerin yakınsama hızı arttırılmıştır. Ayrıca düşük besleme gerilimleri ve yüksüz durumda stator ve rotor arızaları için yeni bir arıza indeksi önerilmiştir. Parçacık sürü optimizasyonu bağışık sistemin mutasyon aşamasında kullanılarak sistem performansı arttırılmıştır. Bu bölüm ile ilgili kabul edilen ve yayımlanan çalışmalar aşağıda verilmiştir [76-81].

- “Zaman Serisi Veri Madenciliği ve Destek Vektör Makinalar Kullanan Yeni bir Arıza Sınıflandırma Yöntemi”, Gazi Üniv. Müh. Mim. Fak. Der., Vol. 23, No 2, pp. 431-440, 2008.
- “Fuzzy Supervised Artificial Immune System for Fault Diagnosis”, First International Fuzzy Conference (FUZYSS), 28-30 Eylül, Ankara, Turkey, 2009.
- "Asenkron Motorlarda Arıza Teşhisi için Kaotik Sürü Optimizasyonu Tabanlı Yapay Bağışık Algoritması", Otomatik Kontrol Ulusal Toplantısı (TOK'10), pp. 574-579, 21-23 Eylül 2010, Gebze, Kocaeli.
- "An Adaptive Artificial Immune System for Fault Classification", Journal of Intelligent Manufacturing, 2010 (Baskı aşamasında), DOI: 10.1007/s10845-010-0449-5.
- "Artificial immune classifier with swarm learning", Engineering Applications of Artificial Intelligence, Vol. 23, Issue 8, pp. 1291-1302, December 2010.
- “A Multi-objective Artificial Immune Algorithm for Parameter Optimization in Support Vector Machine”, Applied Soft Computing, Vol. 11, No:1, pp. 120-129, 2011.

Beşinci bölümde, arıza tahmin ve teşhisi için yeni bir özellik çıkarım yöntemi sunulmuştur. Önerilen özellik sinyali akım sinyalinin belirli sayıdaki periyodunu alarak Hilbert dönüşümü uygulamakta ve elde edilen sinyali sembol serisine dönüştürmektedir. Sembol serisi üzerinden entropi hesabı yapılarak rotor çubuklarının durumu hakkında tahmin yapılmaktadır. Entropi normalde düzensizliklerin ölçümü için kullanılmaktadır. Entropinin bu özelliği arıza teşhisi için kullanılmıştır. Önerilen özellik sinyalinin sonuçlarının doğruluğunu ispatlamak amacıyla motor akım imza analizi ile her bir sağlam ve arızalı durum için rotor frekansının genliği de incelenmiştir. Bu bölüm ile ilgili yapılan çalışmalar aşağıdaki dergi ve konferanslara sunulmuştur [82-84].

- “Early Fault Detection of Broken Rotor Bars Using Sliding Window Symbol Sequence Statistics”, Australasian Universities Power Engineering Conference (AUPEC 09), 17-19 September, Adelaide, Australia, 2009.
- “Bulanık Otomata Yaklaşımı Kullanan Arıza Tespit Algoritması”, 13. Ulusal Elektrik, Elektronik, Bilgisayar, Biyomedikal Kongresi, 23-26 Aralık 2009, Ankara.

- “A new method for early fault detection and diagnosis of broken rotor bars”, Energy Conversion and Management, Vol. 52, No. 4, pp. 1790-1799, 2011.

Altıncı bölümde, gerçek zamanlı olarak durum izlemesi yapabilmek ve herhangi bir arıza oluştuğunda oluşan arızaların tahmini, tespiti ve teşhisi için FPGA tabanlı yöntemler geliştirilmiştir. Bu amaçla negatif seçim tabanlı arıza belirleme algoritması, bulanık mantık tabanlı arıza teşhis algoritması, kaskad yapay sinir ağı tabanlı arıza teşhis algoritması ve yapay sinir ağı tabanlı tahmin algoritması önerilmiştir. Bu bölüm ile ilgili aşağıdaki çalışmalar yapılmıştır [85-87].

- “FPGA Based Real Time Fuzzy Fault Detection Algorithm”, IEEE International Soft Computing and Pattern Recognition Conference (SocPar), pp. 389-394, 7-10 December 2010, Paris, France.
- “FPGA Based Intelligent Condition Monitoring of Induction Motors: Detection, Diagnosis, and Prognosis”, IEEE International Conference on Industrial Electronics (ICIT) & Southeastern Symposium on System Theory (SSST), 14-17 March 2011, Auburn, Alabama, USA.
- “Arıza Teşhisi için Gerçek Zamanlı Bağışık Sistem Uygulaması”, Otomatik Kontrol Türk Milli Komitesi 2011 Ulusal Toplantısı, 13-14 Eylül 2011, İzmir, Türkiye.

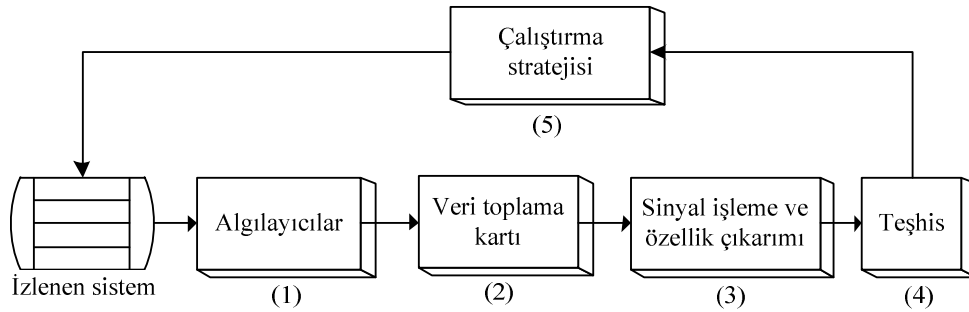
Yedinci bölümde ise tezde yapılan çalışmalar değerlendirilmiş ve gelecek çalışmalar ile ilgili öneriler sunulmuştur.

2. DURUM İZLEME VE ARIZA TEŞHİSİNİN TEMELLERİ

Durum izleme sistemlerinin geliştirilmesi; izlenen sistemin çalışma esnasındaki parametrelerinin ölçülmesi, elde edilen verilerin yönetimi ve yorumlanmasını kapsar. Birçok endüstriyel sistemde elde edilen ölçümler aynı zamanda kontrol ve koruma amaçlı da kullanılmaktadır. Koruma ve kontrol amaçlı kullanılan ölçümlerin durum izleme için kullanılması maliyet açısından oldukça önemlidir.

Erken aşamalarda arızaların gelişimini tespit etmek durum izlemenin önemli özelliklerinden biridir. Fakat bu amaçla kullanılan ölçümler genellikle gürültü içerdiğinden arıza ile gürültünün birbirinden ayırılması gerekir. Elde edilen sinyaller kullanılarak doğru teşhis yapılması, izlenen sistem için minimum aksama süresinde teşhis yapılması ve planlanmamış hızlı bakımı sağlar [21]. Arıza teşhisi ve durum izleme sistemleri, izlenen sisteme zarar vermeden teşhisi sağlamalıdır. İdeal bir arıza teşhis ve durum izleme sistemi bir sistemden minimum ölçümleri almalı ve arıza ile ilgili özellikleri çıkarmalıdır.

Yukarıda bahsedilen durum izleme sisteminin bu görevleri yerine getirebilmesi için sahip olması gereken bazı temel bileşenler vardır. Bir durum izleme sisteminde olması gereken temel bileşenler Şekil 2.1 ile gösterilmektedir.



Şekil 2.1. Durum izlemenin bileşenleri

Şekil 2.1’de verilen bileşenlerden (1) ve (2) ile ilgili görevler veriye dayalı arıza teşhis sistemleri için genellikle izlenmekte olan sistem çalışırken uygulanır. Bu görevleri yerine getirmek için ilk olarak çalışan sistemden ne tür bir verinin alınacağı belirlenir. Daha sonra ilgili duyarga ile sistemden ham verilerin alınması için gerekli bağlantılar yapılır. Sistemden analog sinyallerin okunarak sayısal bir biçimde bilgisayara kaydedilmesi için veri toplama kartları kullanılır. Bileşenlerden (3) ve (4) ise bilgisayar

ortamında uygulandıđından çevrimdışı olarak gerçekleştirilir. Motor durumu teşhis edildikten sonra (5) bileşeni ile motorun çalıştırılması ile ilgili karar verilir.

Duyargalar fiziksel nicelikleri elektriksel sinyallere dönüştürürler. Bu sinyallerin kendisinin veya onlardan elde edilecek özelliklerin arıza teşhisinde kullanılabilmesi için, ciddi arızalar oluşmadan önce sağlam duruma göre değişimlerin izlenebilmesi gerekir [88]. Yani arızanın gelişimini yansıtacak sinyallerin algılanması gerekir. Duyarga seçimi genellikle izleme metodu ve izlenen makinenin arıza mekanizmasındaki bilgiye bağlıdır. Duyarga seçiminde göz önüne alınması gereken ortak kriter, çevrimiçi ölçümün yapılabilmesidir. Ayrıca duyarlılık, maliyet ve izlenen sistemin yapısına zarar vermeme temel gereksinimlerdir.

Bir veri toplama birimi, duyargalardan alınan çıkış sinyallerinin örneklenmesi ve bilgisayara aktarılmasını sağlar. Duyargalardan alınan analog sinyallerin sayısala dönüştürülmesi ve duyarga hatalarının düzeltilmesi işlemi veri toplama birimi sayesinde yapılır.

Duyargalardan alınan sinyaller üzerinden arıza ile ilgili özelliklerin çıkarımı, genelde frekans ve zaman alanında yapılan analizler ile gerçekleştirilmektedir. Arıza ile ilgili özelliklerin sağlam ve arızalı durum için farklı davranışı elde edilen sinyallerin örnekleme oranı ve sinyallerin ne kadar süre kaydedildiđi ile ilgilidir.

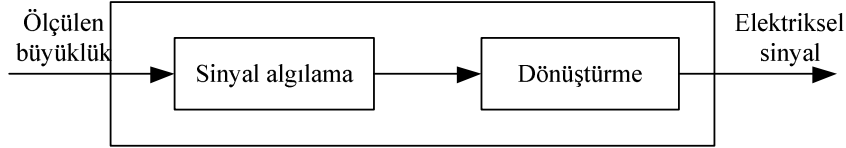
Sinyal işleme ve özellik çıkarımı ile tespit edilen özelliklerden izlenen sistemin durumunun değerlendirilmesi, arızanın şiddeti ve tipinin belirlenmesi bu aşamada gerçekleşir. Geçmişte bu işlem uzmanlar tarafından veya çevrimdışı olarak gerçekleştirilirken yeni eğilimler bu aşamanın gelişmiş teknolojiler ile bütünleşik olarak otomatik bir şekilde çevrimiçi yapılmasına dayalıdır.

Yarı iletkenler, duyargalar ve sinyal işleme alanındaki gelişmeler otomatik durum izlemenin daha da önem kazanmasını sağlamıştır [89]. Teknolojik gelişmeler ile birlikte büyük sistemlerin izlenmesi, durumlarının çevrimiçi değerlendirilmesi ve arıza teşhisi otomatik bir şekilde yapılabilir. Modern durum izleme sistemleri hızlı hesaplama, akıllı analiz ve düşük maliyet gereksinimlerini karşılamalıdır.

2.1. Duyargalar

Duyarga sıcaklık, ışık, elektriksel empedans, radyasyon seviyesi, nem ve benzeri sinyalleri bir kontrol veya ölçüm cihazına ileten aygıttır. Bir dönüştürücü ise herhangi bir sistemden enerjiyi alan ve farklı bir biçimde başka bir sisteme ileten aygıttır. Ölçüm cihazı,

bir sinyalin ölçümü esnasında iki aşamadan geçer. İlk aşama sıcaklık, ivme, akım, gerilim ve benzeri büyüklüklerin algılanmasıdır. İkinci aşama ise ölçülen sinyalin uygun bir biçimde iletimi için dönüştürülmesidir [2]. Dönüştürme aşamasının çıkışı genellikle sayısallaştırılan elektriksel bir sinyaldir. Algılama ve dönüştürme aşamalarının gösterimi Şekil 2.2’de verilmiştir.



Şekil 2.2. Bir ölçüm cihazının şematik gösterimi

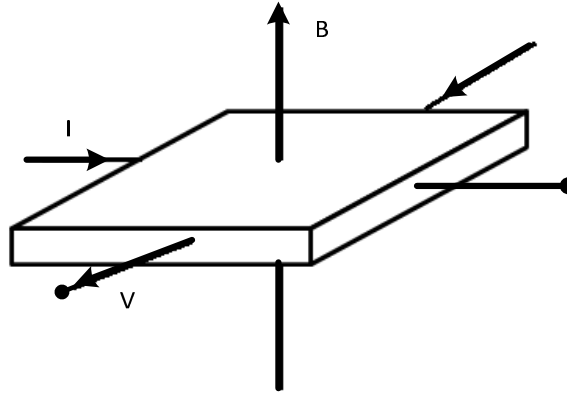
Durum izleme ve arıza teşhisi için farklı duyargalar kullanılmaktadır. Duyargaların çeşitliliğinin temel sebebi, daha iyi arıza teşhisinin yapılabilmesinin sağlanmasıdır. Elektriksel sürücülerin izlenmesinde sıcaklık, titreşim, akım, gerilim, akı ve hız gibi duyargalardan faydalanılmıştır.

Sıcaklık duyargaları ile elektriksel sürücüler ve generatörler geniş bir şekilde izlenmektedir. Türbin generatörleri gibi büyük elektrik makinalarının soğutma sistemi ve stator çekirdeğinin izlenmesi için sıcaklık duyargaları kullanılmaktadır. Sıcaklık ölçümü ile elde edilen bilgi, izlenen makinenin çevre şartları ve yük durumları bilgisi ile birleştirildiğinde arıza durumu hakkında önemli bir parametre elde edilir [5]. Bir asenkron motorun sargıları, sarımların kısa devre olmasını engellemek için izole edilmiştir. Bu izolasyon farklı maksimum çalışma sıcaklıklarına göre kategorilere ayrılmıştır. Fakat çalışma şartları ve yük durumlarından dolayı maksimum sıcaklıklar ihmal edilebilir. Bu ihmaller stator sargılarının belirli noktalarında izolasyonun kaybolması, sargıların incilmesi veya çatlaması ile sonuçlanır [90]. Sıcaklık ölçümü için ısı çiftler, rezistans ölçüm detektörleri ve gömülü sıcaklık detektörleri tercih edilmektedir. Bu tür duyargalar aktif bölgenin sıcaklığını ölçmek için genelde stator sargısı veya çekirdeğine ve mil yatağına monte edilirler.

Titreşim duyargaları, genellikle elektrik makinalarının şaft bağlantıları, rotor arızaları, hava aralığı dengesizliği ve asimetrik güç beslemesi gibi mekanik arızaların izlenmesi ve tespiti için kullanılmaktadır. Son zamanlardaki araştırmalar, akım ve titreşim gibi elektriksel algılayıcılardan alınan durum bilgisinin arızayı belirlemede önemli rol

oynadığını ortaya çıkarmıştır [88]. Titreşim ölçümünde yer değiştirme, hız ve ivmenin ölçümü kullanılmaktadır. Hangi ölçümün kullanılacağı genelde izlenen makinenin büyüklüğüne ve frekans aralığına bağlıdır. Benzer tip ve boyuttaki makinalar genelde az çok sabit bir titreşim hızına sahiptir. Titreşim frekansı arttırıldığında yer değişim seviyeleri düşer ve ivme seviyeleri yükselir [91]. Dolayısıyla alçak frekanslarda yer değiştirme ve hız ölçümleri kullanılırken daha yüksek frekanslarda ivme ölçümü daha kullanışlıdır.

Herhangi bir elektromekanik bir devreden akım ve gerilim sinyallerinin okunması için Hall-etki akım ve gerilim duyargaları kullanılmaktadır. Hall-etki aygıtı küçük bir alandaki akı yoğunluğunu hesaplayabilme kabiliyetine sahiptir. Şekil 2.3'te bir Hall-etki duyarganın çalışma prensibi verilmiştir.



Şekil 2.3. Hall-etki duyarganın çalışma prensibi

Şekil 2.3'te dikey olarak uygulanan B manyetik alan şiddetine göre Hall aygıtından I akımı geçtiğinde aygıtta enine bir gerilim oluşturan Lorentz kuvvetinden dolayı ön yüzeyde elektronlar oluşur. Çıkış gerilimi, uygulanan akım ve manyetik alan şiddeti ile ilişkilidir. Hall-etki nano saniyeden daha az bir sürede oluşturulabildiği için bu türden duyargaların cevap bant genişliği kolayca 100 MHz'in üzerine çıkarılabilmektedir. Küçük ve orta ölçekli elektrik makinaları için Hall-etki akım ve gerilim duyargaları kullanılarak akım ve gerilim sinyalleri ölçülebilmektedir [92]. Şekil 2.4'te tipik bir akım duyargası verilmiştir.

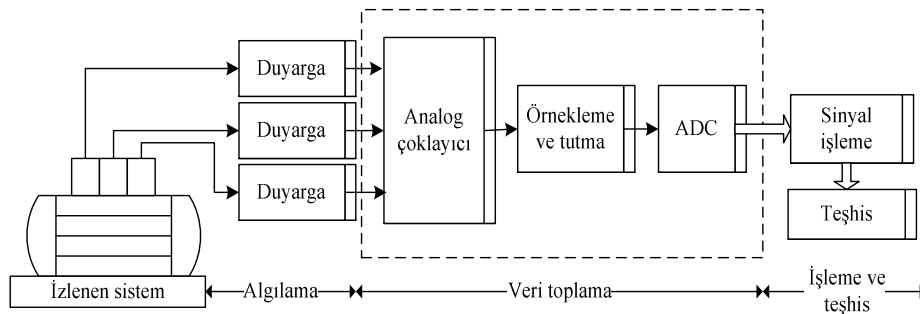


Şekil 2.4. Hall-etki akım duyargası [1].

Akım ve gerilim duyargalarından alınan sinyallerdeki değişimler sağlam ve arızalı durumları belirlemede tek başına yeterli değildir. Elektrik makinasında oluşan arızaların mil yatağı ve rotorda oluşturduğu asimetrliler belirli frekans bileşenlerinde ortaya çıkar. Fakat dikkat edilmesi gereken nokta; spektrumun besleme, statik ve dinamik yük şartları ve motor geometrisi gibi durumlardan da etkilendiğidir. Bu etkiler çoğu zaman yanlış arıza teşhisine de yol açar [21]. Spektrumda ilgili frekansların belirlenmesi için hız bilgisinin de ölçülmesi gerekebilir. Bu durumda hız kodlayıcılarının da kullanılması gerekir.

2.2. Veri Toplama Kartı

Algılayıcılardan okunan sinyallerin uygun bir ortamda kaydedilmesi için veri toplama kartı kullanılmaktadır. Okunan veya kaydedilen sinyallerin izleme sisteminin gereksinimlerini karşılayacak şekilde gürültüsüz ve tutarlı olması istenir. Çoklu girişe sahip karmaşık sistemlerde girişler sürekli veya örnekleme tabanlıdır. Girişlerden okunan analog sinyallerin sayısallaştırılması da gerekmektedir. Basit bir veri toplama kartı sırasıyla çoğullama, örnekleme ile tutma ve analog-sayısal dönüştürme fonksiyonlarını yerine getirir. Şekil 2.5'te veri toplamanın genel düzeni verilmiştir.



Şekil 2.5. Veri toplamanın genel düzeni

Birden fazla kanaldan sinyallerin okunarak bilgisayara aktarılması gerekiyorsa çoğullayıcı kullanılmaktadır. Veri toplama kartı elde edilen analog sinyali sayısalaya dönüştürmek için yüksek çözünürlüklü analog dijital dönüştürücü (ADC)'ye sahiptir. ADC, analog bir gerilim sinyalini belirlenen ölçekleme oranı veya karşılaştırma şemasına göre bilgisayarın anlayabileceği sayısal bir değere dönüştürür. Dönüştürme esnasında giriş analog sinyali örneklenir ve sabit tutulur [5]. ADC'nin çözünürlüğü bit sayısı ile ifade edilir ve durum izleme sisteminin duyarlılığını etkiler. Durum izlemede amaç, zayıf sinyalleri yakalayarak arızanın oluşumunu erken bir aşamada tespit etmektir. Bu yüzden yüksek çözünürlük, veri toplamada önemlidir. 12- bitlik bir analog dijital dönüştürücü 1/4096 çözünürlüğe sahiptir. Dolayısıyla dönüştürme işlemi 0.01 μ s'den az sürer. Çoğullayıcının anahtarlama oranı, çoklu kanallar arasındaki önemli zaman gecikmelerinden kaçınmak için en az 10 MHz olmalıdır.

Farklı ADC çözünürlüğüne sahip genel amaçlı veri toplama kartları durum izleme ve arıza teşhisi için kullanılmaktadır. Şekil 2.6'da bu tezde de kullanılan 8 analog kanala sahip 14-bit bir USB veri toplama kartı gösterilmiştir.



Şekil 2.6. 14-bit USB veri toplama kartı

National Instruments tarafından geliştirilen veri toplama kartı, sinyallerin okunarak bilgisayara aktarılabilmesi için basit bir ara yüz programına sahiptir [93]. Analog sinyallerin okunması için .NET, MATLAB ve LABVIEW ortamlarını desteklenmektedir.

2.3. Sinyal İşleme ve Özellik Çıkarımı

Arıza teşhisinde en önemli aşama, elde edilen sinyallerden arıza ile ilgili özelliklerin çıkarılmasıdır. İşaret işleme tabanlı tekniklerdeki gelişmeler ile birlikte, normal motor koruma işlevlerine ek olarak arızaların gerçek zamanlı olarak tespiti için sinyal işleme

tabanlı yöntemler geliştirilmiştir. Bu çok fonksiyonlu izleme teknikleri yıllardır kullanılan elektromekanik durum izleme tekniklerinin yerini almıştır. Elektriksel duyargalardan alınan sinyalleri direkt kullanmak yerine bir ön işleme yöntemi ile gerekli özellikler çıkarılmalıdır. Çünkü bir sinyaldeki küçük bir değişimi direkt olarak tespit etmek oldukça zordur. Fakat bazı durumlarda direkt olarak sinyal büyüklüklerinin incelenmesi ile de arıza tespiti mümkündür. Tezin bu bölümünde özellikle akım duyargalarından ölçülen faz akım sinyalleri üzerinde uygulanan özellik çıkarım yöntemlerinden bahsedilecektir. Ayrıca elektrik makinalarından asenkron motorlarda arıza teşhisi için özellik çıkarım yöntemleri anlatılacaktır. Bu tür makinalar için geliştirilen arıza teşhis yöntemleri diğer elektrik motorlarına da kolaylıkla uyarlanabilmektedir.

2.3.1. Sinyallerin Büyüklükleri

Üç fazlı bir asenkron motordaki faz akım sinyallerinin büyüklükleri özellikle elektriksel arızaların tespiti için oldukça kullanışlıdır. Gerilim dengesizliği, fazlardan birinin devre dışı kaldığı açık faz arızası ve sargıların kısa devre olması gibi arızalar üç faz akımının büyüklükleri (efektif değeri) incelenerek tespit edilebilmektedir [49, 53, 94]. Rotor hızı ve akım sinyalinin büyüklüğü mil yatağı sürtünme ve stator sargı arızalarını belirlemek için kullanılmıştır [95]. Örneğin fazlardan herhangi bir devre dışı kaldığında diğer iki faz normal değerden bir buçuk kat fazla akım çeker. Motorun A faz akımının efektif değeri aşağıdaki gibi hesaplanmaktadır.

$$rms_{I_A} = \sqrt{\frac{1}{T} \int_t^{t+T} I_A(t)^2 dt} \quad (2.1)$$

Denklemden $I_A(t)$ A faz akım sinyalini, T ise efektif büyüklüğün hesaplandığı periyodu göstermektedir.

2.3.2. Park Vektör Dönüşümü

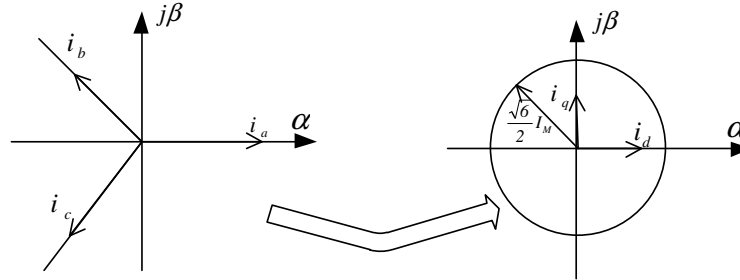
Park vektör dönüşümü, üç fazlı asenkron motorlarda faz akımlarının özelliklerinin iki bileşen ile ifade edilmesini sağlar [29]. Üç fazlı asenkron motorda, üç faz akımı kullanılarak elde edilen park vektör bileşenleri denklem 2.2’de gösterilmiştir.

$$\begin{aligned}
I_d &= \sqrt{\frac{2}{3}}i_A - \frac{1}{\sqrt{6}}i_B - \frac{1}{\sqrt{6}}i_C \\
I_q &= \frac{1}{\sqrt{2}}i_B - \frac{1}{\sqrt{2}}i_C
\end{aligned}
\tag{2.2}$$

Bu denklemde i_A , i_B , ve i_C faz akımlarını gösterir. İdeal şartlar altında, üç faz akımı aşağıdaki bileşenler ile park vektörünü oluşturur.

$$\begin{aligned}
I_d &= \frac{\sqrt{6}}{2}i_M \sin \alpha \\
I_q &= \frac{\sqrt{6}}{2}i_M \sin(\alpha - \frac{\pi}{2})
\end{aligned}
\tag{2.3}$$

Bu denklemde i_M faz besleme akımının maksimum değerini, ω besleme frekansını ve t ise zaman değişkenini gösterir. Üç faz akım sinyali üzerinden yapılan park dönüşümü Şekil 2.7’de gösterilmiştir.



Şekil 2.7. Park vektör dönüşümü

Park vektör dönüşümü, iki bileşenin merkezinde dairesel bir örüntü oluşturur. Bu dönüşüm ile elde edilen örüntülerdeki sapmalar izlenerek arızalı durumlar tespiti edilir. Gerilim dengesizliği, kırık rotor çubuğu arızaları [29, 96] sargı arızaları [48] ve stator açık faz arızalarının [49] belirlenmesinde Park vektör dönüşümünden faydalanılmıştır. Sağlam bir motorda Park vektör bileşenlerinin oluşturduğu iki boyutlu örüntü çember şeklindeyken, arızalı durumda elips biçimine dönüşmektedir.

2.3.3. Temel Bileşen Analizi

Temel bileşen analizi, istatistiksel bir veri analiz yöntemidir [97]. Bu yöntem orijinal bir veri kümesinden temel bileşen olarak adlandırılan yeni bileşenler elde etmek için kullanılır. Temel bileşenler, artık verinin olmadığı dikey bileşenlerden oluşur. Bu bileşenler orijinal değişkenlerin lineer birleşimi ile gösterilirler. Elimizde p adet değişken olduğu varsayılırsa temel bileşen analizi, sistem değişkenlerinin kümesini değişkenleri ilişkisiz olan yeni bir p boyutlu uzaya dönüştürür. Eldeki veri kümesi kullanılarak p boyutlu problem için korelasyon matrisi aşağıdaki gibi elde edilir.

$$K_M = \begin{bmatrix} k_{11}^2 & k_{12} & \dots & k_{1p} \\ k_{12} & k_{22}^2 & \dots & k_{2p} \\ \dots & \dots & \dots & \dots \\ k_{1p} & k_{2p} & \dots & k_{pp}^2 \end{bmatrix} \quad (2.4)$$

Denklemden gösterilen k_{ii}^2 ifadesi x_i değişkeninin varyansını gösterir ve aşağıdaki gibi hesaplanır. k_{ij} ise x_i ve x_j terimleri arasındaki korelasyon değerini ifade edip denklem 2.6'ya göre hesaplanır.

$$k_{ii}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x}_i)^2 \quad (2.5)$$

$$k_{ij} = \frac{1}{n-1} \sum_{j=1}^n (x_i - \bar{x}_i)(x_j - \bar{x}_j) \quad (2.6)$$

Denklem 2.5'teki $\bar{x}_i$ ve $\bar{x}_j$ her iki serinin ortalama değerini ve n ise her serideki örnek sayısını gösterir. Temel bileşen analizi yöntemi $x=(x_1, x_2, x_3, \dots, x_p)$ vektörü ile gösterilen p adet orijinal değişkeni yine p adet ilişkisiz değişken $z=(z_1, z_2, z_3, \dots, z_p)$ ile yeni bir kümeye dönüştürür. Yeni koordinatlar kümesi, orijinal lineer kombinasyonu ile oluşturulan temel bileşenler olarak adlandırılır. Örneğin z_1 ilk temel bileşeni en büyük veri dağılımının olduğu yönü gösteren ilk temel bileşendir. Yeni koordinat sistemi, u_i ile ifade edilen ve korelasyon matrisinden hesaplanan i adet öz vektör ile ifade edilir. Her öz vektör veri kümesinin dağılımının yönünü gösterir. Öz vektörlerin oluşturduğu küme, x 'in koordinatlarını temel bileşenlere dönüştürmek için kullanılan U dönüşüm matrisini sağlar. Bu dönüşüm denklem 2.7'de gösterilmiştir.

$$z = U^T x \quad (2.7)$$

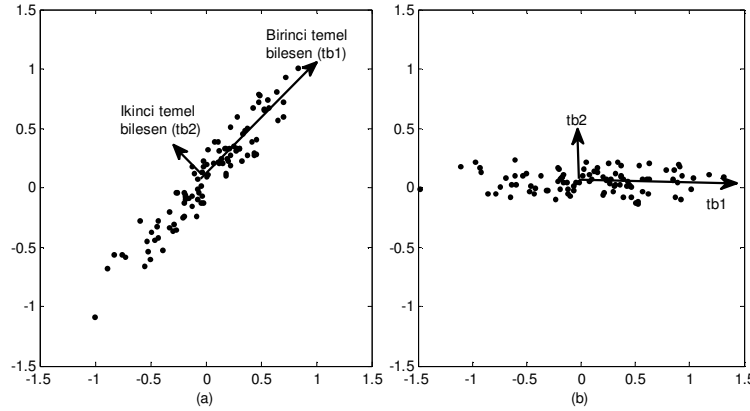
Her bir temel bileşen z_i sıfır ortalama değeri ve u_i vektörünün öz değerine karşılık gelen bir λ_i varyans değeri gösterir. Öz değer λ_i toplanan veri kümesinin öz vektör yönünde nasıl dağılım gösterdiğini belirleyen belirli bir ağırlık verecektir. Temel bileşen analizinin bu özellikleri 2 boyutlu ve her bir boyutu A ve B olan bir veri kümesine uygulanarak her bir boyuttaki maksimum değişimler elde edilebilir. İlk olarak elde edilen A ve B boyutları [A B] şeklinde bir X matrisi formunda yazılır. Daha sonra korelasyon matrisi bulunur. Korelasyon matrisinin öz değerleri ile öz vektörler arasında aşağıdaki ilişki vardır.

$$K_M * v = v * \lambda \quad (2.8)$$

Yukarıdaki denklemde elde edilen öz vektörler orijinal özellik kümesi ile çarpılarak yeni sinyal aşağıdaki gibi elde edilmektedir.

$$X_0 = X * v \quad (2.9)$$

Şekil 2.8’de temel bileşen analizinin geometrik yorumlanması gösterilmiştir.



Şekil 2.8. Temel bileşen analizinin geometrik yorumu (a) Orijinal veri (b) Öz vektör üzerinde yansıtılan veri

Şekil 2.8 (a)’da orijinal veri kümesi gösterilirken, Şekil 2.8 (b)’de ise aynı veri kümesi üzerinde öz vektör uzayı gösterilmiştir. Şekil 2.8 (b)’de gösterildiği gibi birinci temel bileşen daha çok enerji ile “1” değerini alırken ikinci temel bileşen ise daha az enerji ile “1” değerini almaktadır.

Temel bileşen analizi arıza teşhisinde veri azaltma ve farklı boyutlardaki verilerden en fazla enerjiye sahip özelliklerin çıkarılması ile arızaların belirlenmesinde kullanılır. Kırık rotor çubuğu ve stator arızalarının teşhisinde diğer özellik çıkarım yöntemleri ile birlikte kullanılmıştır [8, 96, 98].

2.3.4. Hilbert Dönüşümü

Matematik ve işaret işleme alanında Hilbert dönüşümü herhangi bir $I(t)$ fonksiyonunu alarak bir $H(I(t))$ fonksiyonu üretir. Hilbert dönüşümü periyodik sinyaller için tanımlanmış olup orijinal sinyal ile Hilbert çekirdeği arasındaki konvolüsyon ile verilir. Bu konvolüsyon genelde Cauchy konvolüsyonu olarak bilinir. Hilbert dönüşümü, orijinal seriyi aralarında 90 derecelik faz kayması olan iki seriye dönüştürür. Bu dönüşüm genelde analitik dönüşüm olarak da bilinir. Matematiksel olarak bir $I(t)$ serisinin Hilbert dönüşümü $I(t)$ ile $1/\pi t$ arasındaki konvolüsyon ile hesaplanır [22]. Bu dönüşüm denklem 2.10'da verilmiştir.

$$HD(I(t)) = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{I(\tau)}{t - \tau} d\tau \quad (2.10)$$

Denklem 2.10 ortalama değer teoremi kullanılarak aşağıdaki gibi yazılabilir.

$$HD(I(t)) = \frac{1}{\pi} * I(t) \quad (2.11)$$

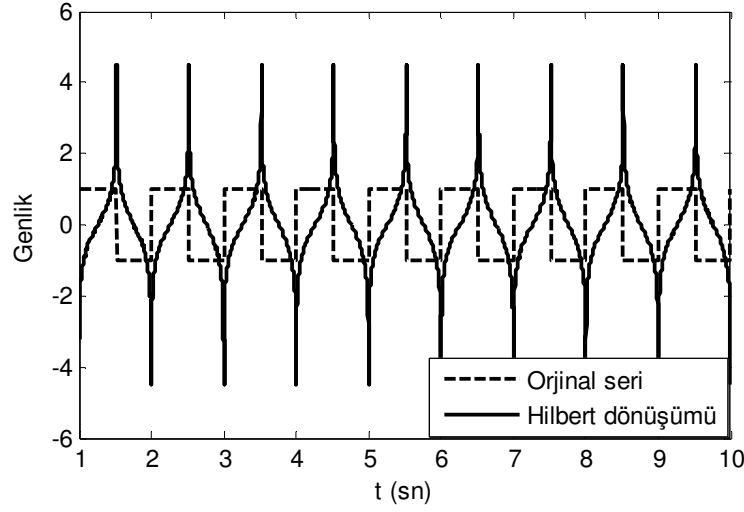
Analitik sinyal bir $I(t)$ sinyalinin Hilbert dönüşümü ve sinyalin kendisi kullanılarak aşağıdaki gibi oluşturulur.

$$AS(t) = I(t) + jH(I(t)) = a(t)e^{j\theta(t)} \quad (2.12)$$

Bu denklemde $a(t)$ ve $\theta(t)$ aşağıdaki gibi hesaplanabilir.

$$a(t) = \sqrt{[I^2(t) + HD^2(t)]} \quad \theta(t) = \arctan(I(t)/H(t)) \quad (2.13)$$

Analitik sinyalin avantajı orijinal sinyalden bütün negatif bileşenlerin filtrelenmesidir. Şekil 2.9'da periyodik bir kare dalga işareti ve bu işaretin Hilbert dönüşümü verilmiştir.



Şekil 2.9. Periyodik bir sinyal ve Hilbert dönüşümü

Hilbert dönüşümünün bileşenleri üzerinden elde edilen analitik sinyal orijinal sinyaldeki değişimleri yansıtmaktadır. Motor akım sinyalleri sinüzoidal olduğundan, Hilbert dönüşümü ile elde edilen bileşenlerden biri sinüs diğeri ise kosinüs olacaktır. Bu iki sinyalin modülü orijinal sinyalde belirlenemeyen değişimlerin ortaya çıkmasını sağlar. Durum izleme ve arıza teşhisinde Hilbert dönüşümünden faydalanılmaktadır. Kırık rotor çubuğu, stator ve mil yatağı arızalarının belirlenmesinde bir önışleme olarak Hilbert dönüşümü kullanılmıştır [40-41, 99-100]. Hilbert dönüşümü ile elde edilen özellik sinyali direkt arıza teşhisi için kullanılabileceği gibi dalgacık analizi ve Fourier dönüşümü uygulanarak da arızalar belirlenebilir.

2.3.5. Motor Akım İmza Analizi (MAIA)

Motor akım imza analizi, bir asenkron motorun üç faz akım sinyallerinden biri kullanılarak stator, rotor ve mil yatağı arızalarının belirlenmesinde faydalanılan bir yöntemdir. Bu yöntemin temeli Fourier dönüşümüne dayanır. Besleme frekansı asenkron motorlarda 60 Hz veya 50 Hz şeklindedir. Dolayısıyla üç fazlı bir asenkron motorun herhangi bir fazından alınan akım bilgisi üzerinden Fourier dönüşümü alındıktan sonra elde edilen spektrumda çalışma frekansının sağ ve solundaki ilk harmonikler yan bant olarak isimlendirilir.

İlgili harmonik bileşenleri hesaplamak için Fourier dönüşümü kullanılır. Hızlı Fourier dönüşümü minimum hesaplama gücü ile ayrık bir zaman serisinin ayrık Fourier

dönüşümünü hesaplamak için kullanılan bir işaret işleme metodudur. Yaygın olarak kullanılan bu spektrum analiz metodu Coleey ve Tukey [101] tarafından yapılan çalışma ile önemli bir hale geldi. Hızlı Fourier dönüşümü N noktalı ayrık Fourier dönüşümü hesaplamalarını daha küçük boyutlu hesaplamalara ardışık olarak ayrıştırarak verilen bir zaman serisinin ayrık Fourier dönüşümünü hesaplar. Eğer bu ayrıştırmalar uygun bir şekilde yapılırsa, hesapsal karmaşıklıkta önemli bir azalma elde edilir. Ayrık Fourier dönüşümünün temellerini incelemek için, N adet eşit parçaya bölünmüş ayrık zamanlı bir $x[n]$ ($0 \leq n \leq N-1$) dizisini düşünelim. Bu ayrık zamanlı $x[n]$ dizisinin ayrık Fourier dönüşümü aşağıdaki şekilde gösterilebilir.

$$X(e^{-j\omega n}) = \sum_{n=1}^{N-1} x[n]w(n)e^{-j\omega n} \quad (2.14)$$

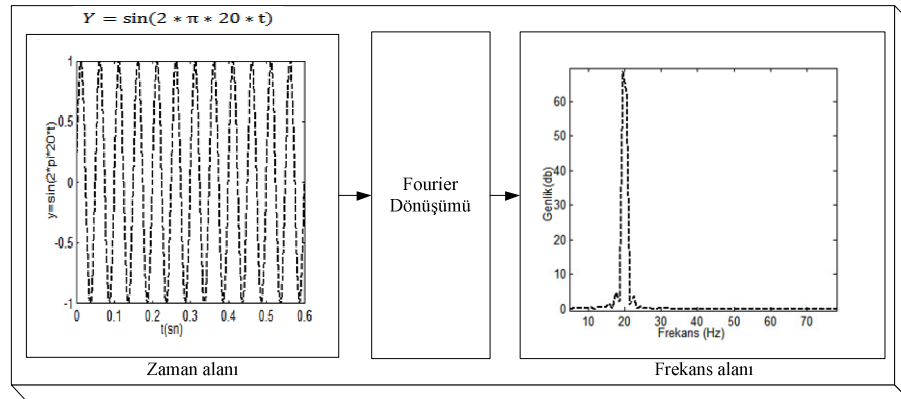
Bu denklemden ω reel frekans değişkeni olup $0 \leq \omega \leq 2\pi$ arasındadır ve değeri denklem 2.15'te verilmiştir.

$$\omega_k = \frac{2\pi k}{N}, \quad 0 \leq k \leq N-1 \quad (2.15)$$

Yukarıdaki değerler denklem 2.14'te yerine konulursa, denklem aşağıdaki gibi olur.

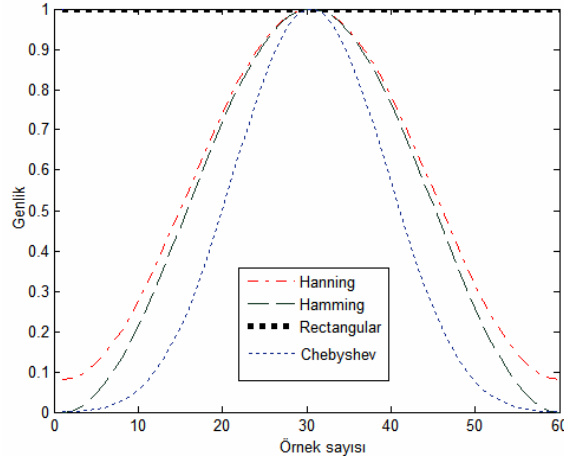
$$X[k] = \sum_{n=1}^{N-1} x[n]e^{-j(2\pi/N)kn}, \quad 0 \leq k \leq N-1 \quad (2.16)$$

Şekil 2.10'da zaman alanında bir sinyal ve bu sinyalin Fourier dönüşümü verilmiştir.



Şekil 2.10. Fourier dönüşümünün uygulanması

Şekil 2.10'daki dönüşümde frekansı 20 Hz olan trigonometrik sinüs fonksiyonunun 1024 noktalı Fourier dönüşümü alınmıştır. Bu şekilde frekans alanında 20 Hz'de genliği 60 db'in üstünde olan bir tepe noktası elde edilmiştir. Ayırık zamanlı bir $x[n]$ serisinin ayırık Fourier dönüşümünün hesapsal karmaşıklığı $O(N^2)$ ile ifade edilir. En popüler pencere fonksiyonları Hamming, Hanning, dikdörtgen ve Chebyshev'dir. Bu fonksiyonların genel biçimi Şekil 2.11'de verilmiştir.



Şekil 2.11. Ayırık Fourier dönüşümünde farklı pencereler

Herhangi bir sinyalin spektrumunu hesaplamada parametrik olmayan periyodogram ve Welch şeklinde iki yöntem bulunmaktadır. Bu yöntemler Ayırık Fourier dönüşümünün performansını arttırmak için geliştirilmiştir. Sürekli zamanlı bir $x(t)$ sinyalinin örneklenmesi ile elde edilen ayırık zamanlı bir $x[n]$ sinyalinin periyodogram spektrumu otokorelasyon fonksiyonunun güç spektrumunun hesaplanması ile elde edilir. Periyodogram metodu sadece ayırık zamanlı ve sonlu elemanlı sinyaller üzerinde N noktalı hızlı Fourier dönüşümünü uygular. Uzunluğu L olan bir $X_L[n]$ sinyalinin periyodogramı, N noktalı güç spektrum yoğunluğu ile aşağıdaki şekilde hesaplanır.

$$P_{xx}(f) = \frac{|X_L[f]|^2}{f_s L}, \quad f_k = \frac{k f_s}{N}, \quad k = 0, 1, \dots, N-1 \quad (2.17)$$

Burada $X_L[f]$ aşağıdaki gibi hesaplanır.

$$X_L = \sum_{i=0}^{N-1} X_L[n] e^{\frac{-2\pi j k n}{N}} \quad (2.18)$$

Welch periyodogram metodunda ilk olarak ayrık zamanlı sinyal K tane parçaya ayrılır. Her bir parça L örnekten oluşur. Bu parçalar $(L-S)$ örtüşme örneği ile bir diğeri ile örtüşebilir. Burada S parçalar arasında kaydırılacak noktaların sayısıdır [22].

Parça 1: $x[0], x[1], \dots, x[L-1]$

Parça 2: $x[S], x[S+1], \dots, x[L+S-1]$

.

.

.

Parça K : $x[N-L], x[N-L+1], \dots, x[N-1]$

Burada ağırlıklı K . parça aşağıdaki örneklerden oluşur.

$$x^z[n] = w[n]x[n + kS], \quad 0 \leq n \leq L-1, \quad 0 \leq k \leq K \quad (2.19)$$

Pencere fonksiyonu $w[n]$ parçanın periodogramı hesaplanmadan önce her bir parçaya uygulanır. Ağırlıklı k . parçanın örnek spektrumu ω reel frekansı için denklem 2.20'deki gibi hesaplanır.

$$P_{xx}^k(e^{j\omega}) = \frac{1}{UL} |X^k(e^{j\omega w})|^2 \quad (2.20)$$

Burada U ayrık zaman pencere enerjisi olup denklem 2.21'deki gibi hesaplanır.

$$U = \sum_{n=0}^{L-1} w^2[n] \quad (2.21)$$

Burada $X^k(e^{j\omega})$ k . parçanın ayrık Fourier dönüşümü olup aşağıdaki gibi hesaplanır.

$$X^k(e^{j\omega}) = \sum_{n=0}^{L-1} x^z[n]e^{-j\omega n} \quad (2.22)$$

Son olarak, Welch periodogramın güç spektrum yoğunluğu bütün parçaların periodogram değerlerinin ortalaması hesaplanarak elde edilir. Bu denklem aşağıda verilmiştir.

$$\hat{P}_w(e^{j\omega}) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{UL} \left| \sum_{n=0}^{L-1} x^{(k)}[n]e^{-j\omega n} \right|^2 \quad (2.23)$$

U faktörü Welch tahmininde pencere enerji eşiğinin etkisini silmek için kullanılır. Fourier dönüşümünün performansının geliştirilmesi için kullanılan Welch ve periyodogram yöntemleri arıza teşhisinde de kullanılmıştır [22, 40]. Fourier dönüşümüne dayalı MAIA kırık rotor çubuğu, stator ve mil yatağı arızalarının tespiti için oldukça uygundur. Temel prensip, Fourier dönüşümü ile oluşturulan frekans spektrumunda her bir arıza türü ile ilgili frekansların genliklerinin incelenmesidir. Bir rotor çubuğu kırıldığı zaman ilgili çubukta akım geçişi olmaz. Bu sebeple rotorun manyetik alanında bir asimetri oluşur. Sincap kafes asenkron motorlarda bu asimetri stator akımının Fourier dönüşümü ile elde edilen spektrumda yan bant frekans bileşenleri oluşturur [102]. MAIA, hat frekansının sağında ve solunda oluşan frekans bileşenlerinin genliklerine göre kırık rotor çubuğu arızalarının belirlenmesi için geniş bir şekilde kullanılmıştır [64]. Akım spektrumunun yan bant bileşenlerinin frekansları aşağıdaki gibi hesaplanır.

$$f_b = (1 \pm 2s)f_s \text{ Hz} \quad (2.24)$$

Denklemden f_s besleme frekansı olup 50 veya 60 Hz'tir. Rotor kayması ise s ile ifade edilmiştir. Yan bant bileşenlerinden $(1-2s)f_s$ sol yan bant, $(1+2s)f_s$ ise sağ yan bant bileşenidir. Sol yan bant bir asenkron motorun rotor kafesinde oluşan elektrik ve manyetik tabanlı asimetriden dolayı oluşurken sağ yan bant ise dönme momentinin titreşimleri ile oluşan hız salınımlarından dolayıdır [103]. Motor kayması motorun mekanik ve senkron hızı kullanılarak denklem 2.25'e göre hesaplanır.

$$s = \frac{n_s - n_m}{n_s} \quad (2.25)$$

Denklem 2.25'te n_s senkron hızı ve n_m ise mekanik hızı gösterir. Motor senkron hızı besleme frekansı ile ilgili olup aşağıdaki gibi hesaplanır.

$$n_s = \frac{120f_s}{P} \quad (2.26)$$

Denklem 2.26'da P motor kutuplarının sayısıdır ve 120 değeri motorun senkron hızını göstermek için kullanılır. Motor akım imza analizi aynı zamanda hava boşluğu eksantrikliği ve mil yatağı arızalarının belirlenmesi için de kullanılır. Motor akım imza analizi, slot frekanslarının yan bantlarındaki harmonikleri inceleyerek hava boşluğu

eksantrikliği arızalarını belirler [33]. Eksantriklik ile ilgili yan bantlar aşağıdaki gibi hesaplanır.

$$f_e = f_s ((kR + n_d) \left(\frac{1-s}{P} \right) \pm n_\omega) \quad (2.27)$$

Burada $k=1, 2, 3, \dots$ şeklinde değer alır. R rotor slot sayısını, n_d dönen eksantriklik sırasını ve n_ω ise stator eksantriklik düzenini gösterir ve $\pm 1, \pm 3, \pm 5$ şeklinde değer alır. Denklem 2.27’de n_d değeri statik eksantriklik için 0, dinamik eksantriklik için ise 1, 2, 3, ... gibi değerler alır [23]. Mil yatağı arızalarını belirlemek için motor akım imza analizi kullanılırken belirli frekans bileşenlerinin belirlenmesi ve arızaların bu frekansların genliklerine göre değerlendirilmesini gerektirir. Mil yatağı rotoru desteklediğinden mil yatağındaki herhangi bir kötüleşme makinenin stator ve rotoru arasında radyal bir hareket üretecektir [90]. Mil yatağı ile ilgili yan bant frekansı hesaplama denklemi aşağıda verilmiştir.

$$f_{ma} = f_s \pm m f_{i,o} \mid m=1, 2, 3, \dots \quad (2.28)$$

Burada $f_{i,o}$ mil yatağının boyutuna dayalı olan karakteristik titreşim frekanslarından biridir. Bu frekans bileşeni aşağıdaki denklem ile hesaplanır.

$$f_{i,o} = \frac{b_s}{2} f_r \left[1 \pm \frac{b_y}{m_y} \cos \theta \right] \quad (2.29)$$

Denklem 2.29’da b_s değeri mil yatağındaki bilyelerin sayısını, f_r Hz cinsinden mekanik rotor hızını, b_y bilyenin yarıçapını, m_y mil yatağının yarıçapını ve θ halkadaki bilyelerin bağlantı açısını gösterir.

2.4.6. Zaman Serileri Analizi

Bir zaman serisi, eşit örnekleme zamanı ile kaydedilmiş olan bilgiler topluluğudur. Herhangi bir zaman serisinden özellik çıkarımı amacıyla kullanılan yöntemler zaman serileri analizi olarak isimlendirilir. Zaman serileri yöntemleri lineer ve lineer olmayan şekilde iki kısma ayrılır. Lineer zaman serileri analizinde Box-Jenkins ve otoregresif birleştirilmiş hareket ortalaması gibi yöntemler kullanılır [104]. Birleştirilmiş hareket ortalaması zaman serisinin durgun olmasını gerektirir. Durgun bir zaman serisinin

istatistiksel karakteristikleri zaman boyunca sabit kalır. Hareket ortalaması metodunda zaman serisinin modeli ile ölçülen zaman serisi arasında farklar oluşabilir. Lineer yöntemler baskın frekanslar, lineer ilişkiler gibi bir veri kümesindeki bütün düzenli yapıları yorumlarlar [105]. Bu yöntemlerde sistem dinamikleri küçük değişimlerin küçük etkilere yol açtığı lineer modeller ile yönetilir. Kaos teorisine göre rastgele bir giriş tek başına bir sistemin çıkışındaki düzensizliğin kaynağı değildir. Lineer olmayan kaotik sistemler hareketin sadece deterministik denklemleri ile çok düzensiz veri üretebilir. Bir sistem hem düzensizlik hem de rastgele girişe sahip olabilir. Bu ikisinin de düzensiz veri üretmesi olasıdır.

Kaos, düzenli sinüzoidal veya yarı periyodik hareketler ile belirsiz ve stokastik davranışlar arasındaki sinyalleri oluşturur. Kaotik bir zaman serisi geniş bant frekans spektrumuna sahip, başlangıç şartlarına yüksek derecede duyarlı olan lineer olmayan bir deterministik süreç ile üretilir [106]. Kaotik bir zaman serisi, lineer olmayan dinamikler, fark ve diferansiyel denklemler ile tanımlanan yörüngeler ile ifade edilir. Bu yüzden kaotik bir zaman serisi, bir durum değişkeni gözlemlenerek üretilmiş bir seri olarak ifade edilir [107].

Kaotik ve lineer olmayan zaman serilerinin incelenmesi için Takens teoremi, zaman gecikmeli gömülmeyi kullanan durum uzaylarının oluşturulması için teorik bir ayarlama sunar [108-109]. Bu alan faz uzayı olarak isimlendirilir. Eğer bir zaman serisi için faz uzayı uygun şekilde oluşturulursa, zaman serisini kendisine benzer fakat herhangi bir gürültü veya değişime sahip seriden ayırt etmek kolay olur. Faz uzayı, her biri orijinal zaman serisinin geciktirilmiş versiyonları olan çok boyutlu bir alanda serinin incelenmesini sağlar. Zaman üzerinde örnekleri eşit aralıklar ile alınmış bir zaman serisi düşünelim. Bu zaman serisi aşağıdaki gibi verilmiş olsun.

$$X = \{x_t, t = 1, \dots, N\} \quad (2.30)$$

Burada t zaman indeksini ve N ise alınan örnek sayısını gösterir. Faz uzayı en az iki boyutlu bir uzayda bir x_t sinyalinin dinamiklerini incelemek için kullanılan bir tekniktir. Herhangi bir sistemden bir veri toplama kartı ile alınan veriler genellikle bir boyutlu olarak kaydedilir. Faz uzayı tekniği farklı gecikme ve gömülme boyutlarına göre elde edilen sinyalin farklı koordinatlarda incelenmesini sağlar. Bir sinyalin faz alanını oluşturmak için kullanılan koordinatların sayısı boyut (b) olarak ifade edilir. Faz alanındaki her bir

koordinat, orijinal zaman serisinin geciktirilmiş halidir. Denklem 2.31'deki gibi bir zaman serisi için faz uzayında bir nokta aşağıdaki gibi verilebilir.

$$X_t = (x_{t+b(\tau-1)}, x_{t+b(\tau-2)}, \dots, x_t) \quad (2.31)$$

Bu denklemde b gömülme boyutunu ve τ ise zaman gecikmesini gösterir. Zaman gecikmesi τ ardışık olmayan zaman gecikmeli örnekler üzerinde faz uzayının oluşturulmasını sağlar. Örneğin $\tau = 3$ ve $b=4$ olduğunda $X_t = (x_{t+8}, x_{t+4}, x_t)$ faz uzayında bir noktaya karşılık gelir. Faz uzayı ise üç boyutlu bir yapı olur. Bu dönüşüm orijinal zaman serisinin lineer olmayan dinamiklerini korur. Bir zaman serisi b gömülme boyutu ve τ zaman gecikmesine göre bir faz uzayına dönüştürülecekse faz uzayı matrisi aşağıdaki gibi gösterilebilir [104].

$$F = \begin{bmatrix} x_{1+(b-1)\tau} & x_{1+(b-2)\tau} & \dots & x_{1+\tau} & x_1 \\ x_{2+(b-1)\tau} & x_{2+(b-2)\tau} & \dots & x_{2+\tau} & x_2 \\ \dots & \dots & \dots & \dots & \dots \\ x_N & x_{N-\tau} & \dots & x_{N-(b-2)\tau} & x_{N-(b-1)\tau} \end{bmatrix}_{(N-(b-1)\tau) \times b} \quad (2.32)$$

Yukarıdaki faz uzayı matrisi F 'ye göre her bir satır faz uzayında bir noktaya karşılık gelir. Matrisin her bir sütunu belirli bir zaman gecikmesi ve gömülme boyutuna göre oluşturulur. Povinelli ve Feng [108]'e göre zaman serisi için gömülme boyutu yeterince büyük seçildiğinde, faz uzayı benzer biçimli olur. Faz uzayı için zaman gecikmesi ve gömülme boyutunun uygun bir şekilde seçilmesi gerekir. Gömülme boyutu orijinal sistemin boyutunun en az iki katından büyük olmalıdır. Orijinal zaman serisi hem stokastik hem de deterministik özelliklerde olduğunda gömülme boyutu sonsuz olabileceğinden değerini tahmin etmek zordur [64]. Faz uzayı için zaman gecikmesinin tahmini karşılıklı bilgi yöntemi ve gömülme boyutunun seçimi yanlış en yakın komşu algoritması ile belirlenir.

Bir zaman serisinin faz uzayı oluşturulurken uygun bir zaman gecikmesi seçilmelidir. Zaman gecikmesi çok küçük seçildiğinde faz uzayındaki noktalar birbirine çok yakın olur. Dolayısıyla bütün noktalar bir köşede toplanır [64]. Eğer zaman gecikmesi çok büyük seçilirse, noktalar faz uzayının tamamını kapsar ve zaman serisindeki değişimleri izlemek zor olur. Zaman gecikmesini tahmin etmek için karşılıklı bilgi yöntemi kullanılır. Bu yöntem aşağıda verilmiştir.

$$M(X_t, X_{t+\tau}) = \sum_{x_n, x_{n+\tau}} P(x_n, x_{n+\tau}) \ln\left(\frac{P(x_n, x_{n+\tau})}{P(x_n)P(x_{n+\tau})}\right) \quad (2.33)$$

Denklem 2.33'te $P(x_n, x_{n+\tau})$ değeri x_n ve $x_{n+\tau}$ sinyali için ortak olasılık yoğunluğudur. $P(x_n)$ x_n için bireysel olasılık yoğunluğunu ve $P(x_{n+\tau})$ ise $x_{n+\tau}$ için bireysel olasılık yoğunluğunu gösterir. Burada x_n ve $x_{n+\tau}$ birbirinden τ zaman gecikmesi ile geciktirilmiş aynı serinin sinyal örnekleridir. Zaman gecikmesi karşılıklı bilgi yönteminin ilk sifıra düştüğü nokta olarak seçilir. Karşılıklı bilgi fonksiyonu iki boyutta optimum bir değere sahiptir. Daha yüksek boyutlar için τ 'nin seçimi henüz hesaplanmadığından boyut hesaplanmasından önce bu değer hesaplanır.

Faz uzayı için minimum gömülme boyutunun hesaplanmasında yanlış en yakın komşu algoritması kullanılmaktadır. Yanlış en yakın komşu algoritması sinyalin bütün noktaları görününceye kadar faz uzayı boyutunu yavaş yavaş artırır. Böylece faz uzayı yanlış en yakın komşu noktalardan bağımsız olur. Yanlış en yakın komşu algoritması aşağıdaki şekilde tanımlanır. İlk olarak iki nokta arasındaki mesafe aşağıdaki gibi hesaplanır.

$$M_n(b)^2 = \|x_n(b) - x_n^{EYK}(b)\|^2$$

$$M_n(b)^2 = \sum_{i=0}^{b-1} [x_{n-i\tau}(b) - x_{n-i\tau}^{EYK}(b)]^2 \quad (2.34)$$

Burada $x_n(b)$ b gömülme boyutu ve τ zaman gecikmesi ile faz alanında bir noktadır. $x_n^{EYK}(b)$ ise $x_n(b)$ 'ye en yakın noktadır. Eğer gömülme boyutu b 'den $b+1$ 'e arttırılırsa iki nokta arasındaki değişim oranı aşağıdaki denklem ile verilir.

$$d_o = \sqrt{\frac{M_n(b+1) - M_n(b)}{M_n(b)}} \quad (2.35)$$

Bu denklemde değişim oranı d_o 0.001 ile 0.01 aralığında bir eşik değeri ile karşılaştırılır ve eğer d_o verilen bu eşik değerini aşarsa $x_n(b)$ yanlış bir komşuya sahiptir. Gömülme boyutu, R^b 'de $d_o > 0.001$ şartını sağlayan noktaların sayısı ilk sifıra yaklaştığı boyut ile ifade edilir. Denklem 2.36'da 1000 örnek üzerinde zaman gecikmesi 4 ve gömülme boyutu 3 seçilerek elde edilen faz uzayı matrisi gösterilmiştir.

$$F = \begin{bmatrix} x_9 & x_5 & x_1 \\ x_{10} & x_6 & x_6 \\ \dots & \dots & \dots \\ x_{1000} & x_{996} & x_{992} \end{bmatrix} \quad (2.36)$$

Oluşturulan bu matriste her bir satır bir noktaya karşılık gelirken, her bir sütun ise faz uzayının bir boyutunu oluşturur. Farklı sinyaller için farklı faz uzayları oluşturulabilir. Her sinyal için faz uzayı hesaplanmadan önce o faz uzayını oluşturacak uygun zaman gecikmesi ve gömülme boyutu belirlenmelidir. Eğer faz uzayı uygun bir şekilde oluşturulursa, zaman serilerini birbirinden ayırt etmek oldukça kolay bir biçime dönüşebilir.

Lineer olmayan zaman serileri analizi, arıza belirleme için önemli avantajlar sunar. Yöntem eğitim aşamasında faz uzaylarının oluşturulması için sadece arızalı ve sağlam veri kümelerini kullandığından geniş veri kümelerine ihtiyaç duymaz. Sağlam ve arızalı faz uzaylarındaki farklılıklar arızaların belirlenmesi için yeterlidir. Kırık rotor çubuğu, kırık konektör arızaları ve stator arızalarının teşhisinde zaman serileri analizinden faydalanılmıştır [63-64, 110-111].

2.4. Arıza Belirleme ve Teşhis

Arıza belirleme ve teşhis aşaması, elde edilen özelliklerden oluşan eğitim veri kümesinin akıllı hesaplama tekniklerine verilerek arıza teşhis sistemin parametrelerinin ayarlanması ve test aşamasında gerçekleştirilen teşhis adımlarından oluşur. Arıza belirleme aşamasında sistemde arıza olup olmadığı belirlenirken, teşhis aşamasında arızanın türü de belirlenmektedir. Arıza tespiti, iki sınıflı sınıflandırıcı olarak düşünülebilir. Arıza teşhisi ise çok sınıflı sınıflandırma işlemine karşılık gelir.

Arıza belirlemede, sağlam ve arızalı durum örneklerine göre işlem yapılır. Arıza belirleyici sistemin performansı olasılık tablosuna göre belirlenir. Tablo 2.1’de 2x2’lik gerçek sınıf ile tespit edilen sınıfın olasılık tablosu verilmiştir.

Tablo 2.1. 2x2 olasılık tablosu

		Eğitim	
		Negatif	Pozitif
Test	Negatif	Doğru pozitif (DP)	Yanlış pozitif (YP)
	Pozitif	Yanlış negatif (YN)	Doğru negatif (DN)

Doğru pozitif oranı (Acc^+) pozitif örneklerin tespit oranını gösterir ve doğru negatif oranı (Acc^-) negatif örneklerin tespit oranını gösterir. G-ortalama ise algoritmanın performansını gösterir. Bu hesaplamalar aşağıda verilmiştir.

$$Acc^+ = \frac{DP}{DP + YN} \quad (2.37)$$

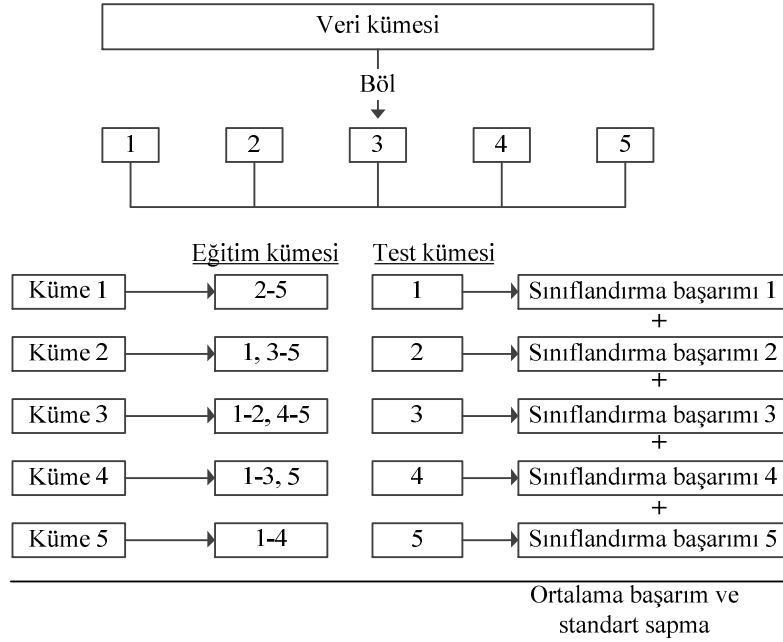
$$Acc^- = \frac{DN}{DN + YP} \quad (2.38)$$

$$G - ortalama = \sqrt{Acc^+ \times Acc^-} \quad (2.39)$$

Performans ölçümü için diğer bir faktör ise yanlış alarm oranını gösteren negatif tespit oranıdır. Bu oran aşağıdaki gibi hesaplanır.

$$Fa = \frac{YP}{YP + DN} \quad (2.40)$$

Sınıflandırma performansının test edilmesi için en çok başvurulan yöntem k-noktalı çapraz doğrulamadır. Bu yöntem veri kümesini k parçaya bölerek, k-1 parçayı eğitim için kullanırken geriye kalan parçayı sınıflandırıcının performansını test etmek için kullanır. Test işlemi bütün parçalar için gerçekleştirilerek ortalama bir başarı oranı ve standart sapma elde edilir. Şekil 2.12’de 5-noktalı çapraz doğrulama işlemi gösterilmiştir.



Şekil 2.12. 5-noktalı çapraz doğrulama ile sınıflandırma başarımının hesabı

Şekil 2.12’de görüldüğü gibi eğitim için her defasında bölünmüş veri kümelerinden dördü eğitim için kullanılırken bir tanesi elde edilen sınıflandırıcı parametrelerine göre test işleminde kullanılır. Her bir test işlemi sonucunda elde edilen sınıflandırma başarımlarının ortalaması alınarak teşhis edicinin başarımlar oranı ve standart sapması elde edilir.

2.5. Çalıştırma Stratejisi

Veriye dayalı durum izlemede en son aşama izlenen sistemin çalışma durumu ile ilgili karar verilmesidir. Eğer sistemde herhangi bir arıza yok ise sistem geçerli durumda çalıştırılmaya devam ettirilir. Arıza tespit edildikten sonra ilgili arıza için sisteme hemen müdahale edilmelidir. Çünkü izlenen sistemin bir bileşeninde oluşan arızaya müdahale edilmezse diğer bileşenlerin de arızalanması kaçınılmaz olur. Örneğin bir asenkron motorda mil yatağında arıza oluştuğunda, ilgili bakım yapılmazsa hava aralığı eksantrikliği ve rotor arızalarının oluşumu hızlanır.

3. YAPAY BAĞIŞIKLIĞA DAYALI ARIZA BELİRLEME YÖNTEMLERİ

Tezin bu bölümünde negatif seçim algoritmasındaki detektörlerin optimizasyonu için iki farklı yöntem sunulmaktadır. Böylece negatif seçim algoritmasının tespit yeteneğinin geliştirilmesi ve detektör üretim aşamasının daha hızlı yapılması amaçlanmıştır. Birinci yöntem öz olmayan uzayın maksimum kapsanması için genetik algoritma tabanlı bir optimizasyon algoritması sunmaktadır. Önerilen algoritmada detektörlerin öz olmayan uzayı maksimum bir şekilde kapsaması için bir amaç fonksiyonu belirlenmiştir. Detektör üretimi için önerilen yöntem daha önce üretilen detektörlerin yeniden üretimini engellemektedir. Böylece sağlam durum dışında arızaya sebep olan küçük değişimler detektörler tarafından kolayca tespit edilebilecektir. Yöntemin performansını değerlendirmek için kullanılan veri kümesi üzerinde yeni bir özellik çıkarım yöntemi de sunulmuştur. Asenkron motorlarda oluşan kırık rotor çubuğu arızalarının tespiti için akım sinyalleri üzerinde önerilen özellik çıkarımı, sağlam ve arızalı durumların ayırt edilmesinde oldukça kullanışlıdır. İkinci yöntem detektör üretimi için klonal seçim ve K-en yakın komşu algoritmasından faydalanmaktadır. Detektör üretimi aşamasında hem sağlam hem de arızalı sinyaller kullanılmakta ve öz olmayan uzay yerine öz olmayan veri kümesinin tam kapsanması için detektörler üretilmektedir. Ayrıca bir test verisi herhangi bir detektör tarafından tespit edilemediği durumda ilgili veriye en yakın detektör K-en yakın komşu algoritması ile bulunarak bu veri üzerinde detektör oluşturulmaktadır. Önerilen yöntem kırık rotor çubuğu arızalarının tespiti için kullanılmıştır.

Aşağıda alt bölümlerde sırasıyla yapay bağışık algoritmasının prensipleri verilerek, bu algoritmanın iki temel bileşeni olan negatif seçim ve klonal seçim algoritmaları anlatılmıştır. Arıza belirlemek için önerilen iki yöntem sunularak deneysel sonuçları verilmiştir. Her bir yöntemin tespit performansları diğer akıllı teknikler ile karşılaştırılmıştır. Bölüm sonuçlar ve analiz ile sonuçlandırılmıştır.

3.1. Yapay Bağışık Sistemler

Doğal bağışıklık sisteminin amacı, vücuttaki bütün hücrelerin tanınması ve bunların öz veya öz olmayan olarak ayırt edilmesidir. Öz olmayan hücreler, daha sonradan uygun bir koruma mekanizmasının oluşturulması için kullanılır. Bağışıklık sistemi, bakteri ve virüs olarak adlandırılan yabancı hücreler ile öz hücreleri birbirinden ayırt eder [112].

Doğal bağışıklık sistemi bileşeni olan negatif seçim öz hücreler için bir tolerans belirler. Bağışık sistemin öğrenme yeteneği kullanılarak öz olmayan hücreler ile reaksiyona girmeyen antijenler tanınır. Negatif seçim algoritması, yabancı antijenlerin tanımlanması için doğal bağışıklık sisteminden esinlenilerek geliştirilmiş bir algoritmadır. Doğal bağışıklık sistemindeki negatif seçim, vücudun antikorlarının diğer antikorları tanıyarak herhangi bir yabancı hücre veya antijen vücuda girdiğinde bu hücrelere karşı bir cevap üretmesidir. Yapay bağışık sistemdeki negatif seçim ise öz olarak kabul edilen verileri eğitim aşamasında kullanarak, bu veriler dışında gelebilecek verileri tanıyacak detektör veya tespit edicilerin üretilmesidir.

Negatif seçim algoritmasının çalışması prensibi oldukça basit olup anomali tespiti ve arıza belirleme için uygundur. Arıza tespitinde sağlam veriler öz örnekler olarak alınıp negatif seçim algoritması eğitilerek detektörler üretilir ve test aşamasında gelen öz olmayan veriler detektörler tarafından tespit edilir. Fakat orijinal negatif seçim algoritması birçok dezavantaja sahiptir [113]. Bu algoritmada eğitim aşamasında detektörler genellikle rastgele üretilip herhangi bir kontrol mekanizması yoktur. Dolayısıyla aynı detektörün bir sonraki aşamada üretilme ihtimali vardır. Üretilen detektörler öz olmayan uzayı kapsamayabilir [114]. Bu durumda test aşamasında gelen öz olmayan bir veri detektörler tarafından tespit edilemez.

Negatif seçim algoritmasının performansını geliştirmek için, detektörlerin optimize edilmesi gerekir. Eğitim aşamasında üretilen detektörler için bir kontrol mekanizması geliştirilerek minimum sayıda detektör ile öz olmayan uzayın maksimum kapsanması sağlanabilir. Detektör üretiminde bir diğer amaç ise aynı tür detektörün üretiminin engellenmesidir. Ayrıca negatif seçim algoritması için uygun bir ön işlemenin yapılması da algoritma için önemlidir.

3.1.1. Negatif Seçim Algoritması

Doğal bağışık sistemi, insan vücudunu yabancı hücreler olarak bilinen antijen ve patojenlerden koruyan etkili bir mekanizmadır [115]. Virüs, bakteri, mantar ve parazitler gibi mikroorganizmalar patojen olarak adlandırılır ve bunlar vücuda girdikten sonra hastalığa sebep olurlar. Bağışık sistemdeki temel amaç, bu patojenlerin tanınması ile ilgilidir. Patojenler bağışık sistem bileşenleri tarafından direkt olarak tanınmazlar. Patojenlerin antijen gibi bazı küçük molekülleri, bağışık sistem tarafından tanınabilir. Hastalığa sebep olan bir antijen belirlendikten veya tanındıktan sonra bağışık sistemi onu

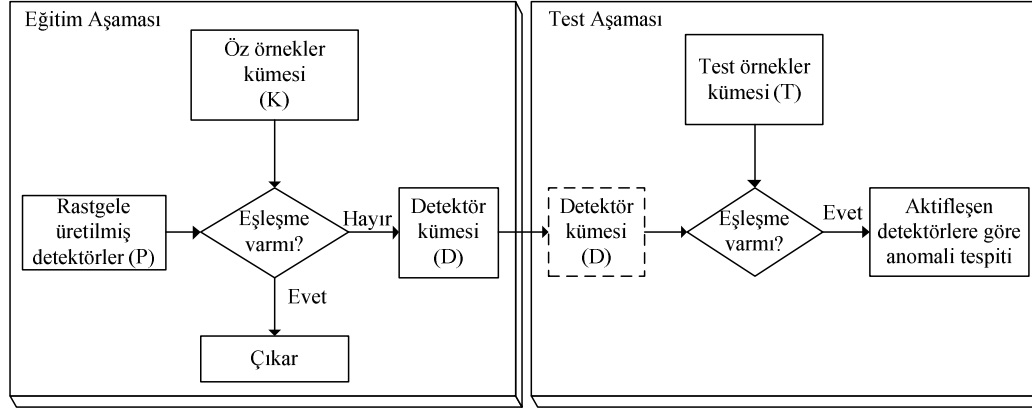
yok etmekten sorumludur. Doğal bağışıklık sisteminde vücuda giren bu patojen veya antijenleri tanıyan sistem negatif seçim olarak adlandırılır.

Doğal bağışıklık sistemlerdeki yabancı hücrelerin tanınması kemik iliğinde üretilen B ve T gibi iki lenfosit hücre ile yapılmaktadır [112]. T hücreleri timüste negatif seçim olarak adlandırılan bir süreçten geçer ve sadece hücrenin kendi proteinleri ile eşleşmeyen hücreler kalır. Diğer T hücrelerinin negatif seçimi, kendi antijenlerini tanıyan hücre reseptörlerini yok eder. Bu yolla hayatta kalan bütün T hücrelerinin sadece öz olmayan hücreleri tanıdığı varsayılır. T hücrelerinin bu şekilde elenmesi vücudun kendi proteinlerinin saldırılarından bağışıklık sistemimizi korur.

Daha önce görülmeyen kural dışı davranışlara karşılık doğal bağışıklığın cevabı bilgisayar algoritmaları için ilham kaynağı olmuştur. Bu fikirden esinlenilerek, Forrest ve diğ. [116] timüs 'teki T-hücrelerinin negatif seçimine dayalı bir kural dışı tespit algoritması geliştirdi. Negatif seçim olarak isimlendirilen bu algoritma bilgisayar güvenliği için kullanılmıştır. Algoritmanın önemli bir avantajı, sistem için bilinmeyen örüntü kümesi hakkında bilgi depolayarak örüntü tanıma gibi görevleri uygulamak için kullanılabilmesidir.

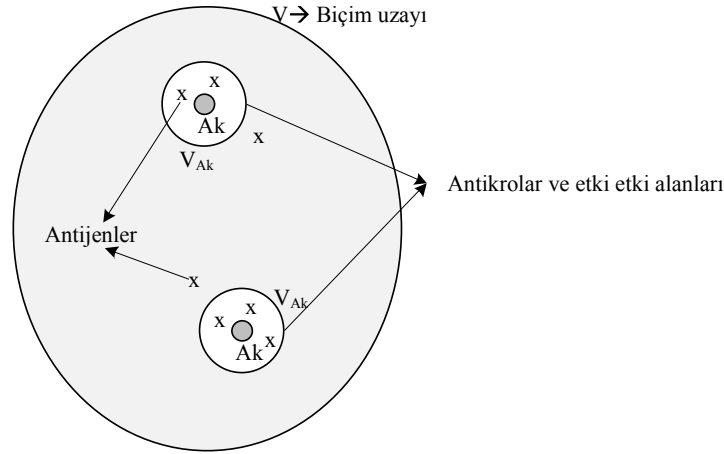
Negatif seçim algoritması basit olup temel adımları aşağıda verilmiştir. Korunmuş bir K öz örüntü kümesi alınarak detektör olarak isimlendirilen örüntü tanımlayıcı D kümesi üretilir. Bu detektör kümesi K'ya ait olan herhangi bir örnek ile eşleşmez. D kümesinin üretilmesi süreci aşağıdaki gibi tanımlanabilir ve Şekil 3.1'de özetlenmiştir.

1. Rastgele örnekler üret ve onları P kümesine yerleştir.
2. Öz küme K'nın bütün örnekleri ile P'deki bütün örneklerin benzerliğini belirle.
3. K'nın en az bir örneği ile P'nin bir örneğinin benzerliği verilen bir benzerlik eşik değerine eşit veya büyük ise, P'deki örnek öz örneği tanır ve elenir, aksi takdirde P'deki örnek öz olmayan bir kümeye ait olur ve D kümesine taşınır.



Şekil 3. 1. Negatif seçim algoritması

Negatif seçim algoritmasında ilk aşama öz hücrelerin tanınmasıdır. Öz hücreler belirlendikten sonra, rastgele bir şekilde aday detektörler üretilir. Üretilen detektörlerden öz hücreler ile eşleşenler çıkarılır ve eşleşmeyenler ise detektör kümesine atılır. Bir aday detektör ile öz hücre arasındaki benzerliği ölçerek eşleşmenin hesaplanması için Öklid, Hamilton ve Manhattan mesafe ölçümleri kullanılır. Bir antijen ve antikorı modellemek için kullanılan biçim uzayı, kendi benzerliklerini hesaplamak için kısmen bir ölçüm belirler. Biçim uzayı tabanlı detektör tanıma sistemi Şekil 3.2’de gösterilmiştir.



Şekil 3.2. Biçim uzayı tabanlı antijen tanıma sistemi

Şekil 3.2’de herhangi bir antikor, V_{Ak} gibi bir daire içerisinde kalan x ile ifade edilen bir antijeni tanıyabilir. Bunun dışında kalan antijenleri tanıyamaz. Örneğin bir antikor örneği $A_k = [ak_1, ak_2, \dots, ak_m]$ ve bir antijen örneği ise $An = [an_1, an_2, \dots, an_m]$ ile ifade

edilir. Burada m antikor ve antijenin özellik sayısıdır. Bu iki örnek arasındaki öklid benzerlik ölçümü denklem 3.1, Hamilton benzerlik ölçümü denklem 3.2 ve Manhattan benzerlik ölçümü ise denklem 3.3'te verilmiştir.

$$b\ddot{o} = \sqrt{\sum_{i=1}^m (Ak_i - An_i)^2} \quad (3.1)$$

$$b\ddot{o} = \sum_{i=1}^m |Ak_i - An_i| \quad (3.2)$$

$$b\ddot{o} = \sum_{i=1}^m \delta_i \text{ burada } \delta_i = \begin{cases} 1 & Ak_i \neq An_i \\ 0 & \text{diğer} \end{cases} \quad (3.3)$$

Burada elde edilen benzerlik ölçümlerinin değeri ($b\ddot{o}$) ε gibi bir eşik ile denklem 3.4'teki gibi karşılaştırılır.

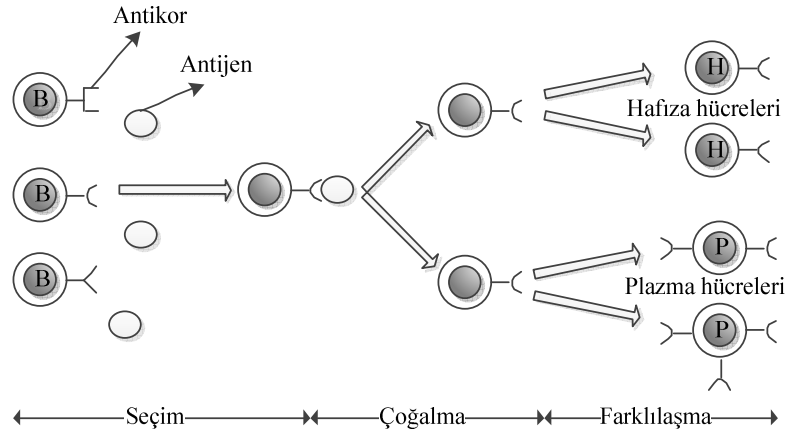
$$E = b\ddot{o} - \varepsilon \quad (3.4)$$

Eğer $E > 0$ ise $An = [an_1, an_2, \dots, an_m]$ antijeni öz örnekler ile eşleşmez ve eğer $An = [an_1, an_2, \dots, an_m]$ bütün öz örnekler ile eşleşmez ise detektör kümesine eklenir. Belirli sayıda detektör üretildikten sonra, test aşamasında bu detektörler kullanılarak gelen örneklerden öz kümelerin dışındakiler belirlenir. Dolayısıyla $Y = [y_1', y_2', \dots, y_m']$ gibi bir yeni örnek veri geldiğinde $An = [an_1, an_2, \dots, an_m]$ ile eşleşiyorsa bir anormallik tespit edilir.

3.1.2. Klonal Seçim Algoritması

Klonal seçim uyarlamalı bağışık sistemin yabancı hücrelere karşı davranışından türetilmiştir. Uyarlamalı bağışık sistem tarafından tespit edilebilen yabancı hücreler antijen olarak isimlendirilir. Vücut bir antijene maruz kaldığında B lenfositleri tarafından üretilen antikorlar ile bir tepki oluşturulur. Antikorlar B hücrelerinin yüzeylerindeki moleküllerdir. Bu moleküllerin amacı antijenlere bağlanarak onları tanımadır. Her B hücresinde bir antijeni tanıyacak şekilde tek tip bir antikor vardır. Herhangi bir antikor, antijeni tanıdığı anda B hücreleri T yardımcı hücreler sayesinde bölünerek çoğalma ve olgunlaşma sürecine girer [117]. Hücre bölünmesi ile tek bir hücre veya bir hücre kümesinin kopyaları (klon) oluşur.

Plazma hücresinde çoğalma ve farklılaşma sürecine giren B hücreleri daha uzun süreli hayatta kalabilen hafıza hücrelerine dönüşürler. Hafıza hücreleri ikinci defa bir antijenik uyarıya maruz kaldığında antikorların antijenlere olan afinitelerinin yüksek olmasını sağlamak amacıyla doku, kan ve lenfositlerden dolaşırlar [112]. Şekil 3.3'te doğal bağışıklık sistemindeki klonal süreç verilmiştir.



Şekil 3.3. Klonal seçim prensibi

Şekil 3.3'te verilen klonal seçimin temel özellikleri aşağıdaki gibi verilebilir.

- Antijenler tarafından uyarılan hücrelerin çoğalması ve farklılaşması,
- Hafıza kümesinin korunması,
- Antijenler tarafından uyarılmayan antikorların elenmesi,
- Afiniteleri ile orantılı olarak klonların yeniden seçimi, üretimi ve çeşitliliğinin korunması,
- Afinite olgunlaşması olarak adlandırılan süreç ile antikorlar üzerinde rastgele değişimin sağlanması,
- Antijenlere karşı düşük afinite taşıyan farklılaşmış lenfositlerin tahmini.

Doğal bağışıklık sistemindeki klonal seçim teorisi optimizasyon ve örüntü tanıma gibi problemlerin çözümü için uyarlanmıştır. Klonal seçim algoritmasında temel olarak iki popülasyon vardır. Bunlar antijen ve antikor kümesidir. Antikor kümesi problemin optimizasyon veya örüntü tanıma olmasına göre alt kümeler ayrılabilir. Şekil 3.4'te klonal seçim algoritması verilmiştir [112].

S→Tanınacak antijen kümesi g→ Adım sayısı N→Popülasyon boyutu n_1→Klonlama için seçilecek yüksek afiniteli antikor sayısı n_2→İterasyon sonunda yer değiştirilecek düşük afiniteli eleman sayısı
J←0 P←Rastgele olarak N boyutlu popülasyonu üret While j<g do Afinite(P, S)←Antikor ile antijenler arasında afinite hesabı yap P_1←Seçim(P, n_1) //Afinitesi iyi olan antikorlardan klonlama için n_1 adet seç C←Klonlama(P_1, afinite(P_1)) // Afinitelerine göre P_1'den klonlar oluştur C_1←Mutasyon(C, afinite(C)) // Afinitelerine göre C'ye mutasyon uygula Afinite(C_1, S)←Klon kümesi(C_1) ile antijenler arasındaki afiniteyi hesapla M←Seçim(C_1) P←Yerdegistir(P, n_2) j←j+1 End while

Şekil 3.4. Klonal seçim algoritması

Şekil 3.4'te verilen klonal seçim algoritması örüntü tanıma problemleri için kullanılmaktadır. Afinite ölçümü, uygulanan probleme göre antikor ile antijen arasındaki Hamming, Öklid veya Manhattan mesafesi ile hesaplanır. Afinite ile orantılı olarak klon oluşturma işlemi yapılır. Seçim işlemi ile elde edilen yüksek afiniteye sahip antikorların her biri için eşit sayıda klon üretmek yerine afinitelerine göre klon üretmek daha uygundur [118]. Klon için seçilen n_1 adet antikor afinitelerine göre sıralanarak her bir antikor için üretilecek klon sayısı denklem 3.5'e göre hesaplanır.

$$N_C = \sum_{i=1}^{n_1} \text{round}\left(\frac{\beta \cdot N}{i}\right) \quad (3.5)$$

Denklemde N_C antijenlerinin her biri için üretilen toplam klon sayısını, β çarpan faktörü ve N ise toplam antikor sayısını gösterir. Yuvarlama fonksiyonu round(.) kullanılarak tam sayı formatında klon üretimi sağlanır. Örneğin $N=10$, $\beta=1$ ve en yüksek afiniteli antikor için ($i=1$) 10 adet klon, ikinci en yüksek afiniteli antikor için 5 adet klon üretilir. Mutasyon işlemi de afinite ile ters orantılı olarak uygulanmaktadır. Afinite oranı denklem 3.6'ya göre uygulanır [112].

$$\alpha = e^{-(\rho \cdot f^*)} \quad (3.6)$$

Denklem 3.6'da α mutasyon oranını, ρ kontrol parametresini, f^* ise $[0, 1]$ aralığına çekilmiş afiniteyi gösterir. Denklemden kullanılan f^* , f / f_{maks} ile bulunur. Örüntü tanıma için kullanılan klonal seçim algoritması üzerinde birkaç değişiklik yapılarak optimizasyon problemlerini çözmek için de uyarlanabilir. Algoritmada tanınacak antijen kümesi çıkarılarak maksimum veya minimum yapılacak bir amaç fonksiyonu belirlenir. Dolayısıyla bir antikoru afinitesi amaç fonksiyonu olarak ifade edilir [115].

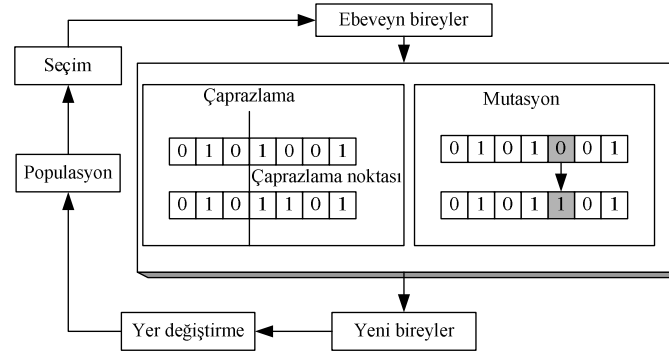
3.2. Arıza Tespiti için Genetik Algoritma Tabanlı Negatif Seçim Algoritması

Tez çalışmasının bu bölümünde kırık rotor çubuğu arızalarının tespiti için tek faz akım sinyali üzerinden Hilbert dönüşümü uygulanarak yeni bir özellik sinyali elde edilmiştir. Ayrıca aynı akım sinyali üzerinden ilk fark filtrelemesi ile özellik sinyali elde edilerek her iki sinyalin faz uzayları oluşturulmuştur. Bu özellik sinyallerinin faz uzayı doğrusal olmayan zaman serileri analizi yöntemi ile elde edilerek negatif seçimin giriş verisi oluşturulur. Hilbert tabanlı dönüşüm olarak adlandırılan yeni özellik sinyali faz uzayında kırık rotor çubuğu arızalarını ayırt etmek için kullanışlıdır. Orijinal negatif seçim algoritmasında detektörler rastgele üretilir. Fakat rasgele üretilen detektörler iki probleme sahiptir. Birincisi öz olmayan uzay kapsamayabilir. İkincisi benzer detektörlerin üretimini engellemek için herhangi bir sınırlama yoktur. Detektörlerin optimize edilmesi ve öz olmayan uzayın maksimum kapsamı için genetik algoritma tabanlı yeni bir yöntem sunulmuştur. Önerilen yöntemde minimum detektör sayısı ile öz olmayan uzayın maksimum kapsamı sağlanmıştır.

3.2.1. Genetik Algoritma

Genetik algoritma evrimsel programlamanın bir parçası olarak ilk defa Michigan üniversitesinde psikolog ve bilgisayar uzmanı olan J. H. Holland tarafından önerilmiştir [119]. Genetik algoritmalar doğal seçim ve en iyi bireyin korunmasına dayalı bir arama yöntemidir. Genetik algoritmanın en önemli özelliği kör arama metodu olmasıdır. Çünkü genetik algoritma en iyi çözümün ne olacağını başlangıçta bilmez. Geliştirilecek algoritma ile amaç fonksiyonuna göre en iyiye yakınsar. Standart bir genetik algoritmada, problemin çözümünü oluşturacak aday bireyler eşit boyutlu vektörler olarak ifade edilir. Başlangıçta bu vektörler rastgele olarak üretilir. Buradaki her bir birey kromozom olarak adlandırılır.

Her birey problemin çözümü olabilir. Bu kromozomlar üzerinde çaprazlama, mutasyon ve seçim işlemleri sonrasında değişiklikler olur ve problemin çözümünü sağlayacak en iyi birey elde edilir. Şekil 3.5'te genetik algoritmanın temel adımları gösterilmiştir.



Şekil 3.5. Genetik algoritmanın adımları

Genetik algoritmada popülasyon rastgele üretilir. Popülasyondaki her bir bireyin uygunluğu çaprazlama ve mutasyon ile değiştirilir. Seçim işlemi için rulet tekerleği ve turnuva seçimi en çok tercih edilen yöntemlerdir. Seçim operatörü bir bireyin bir sonraki adımda hayatta kalıp kalmayacağını belirler. Daha iyi kromozomların bir sonraki adımda hayatta kalma şansları daha yüksektir.

3.2.2. Yeni Özellik Sinyali ve Faz Uzunluğunun Oluşturulması

Motor akım sinyallerinden özellik çıkarımı için burada önerilen iki yöntem lineer olmayan zaman serileri analizine dayalıdır. İlk yöntemde zaman serisinden özellik çıkarımı için ilk fark filtrelemesi kullanılmıştır. Bu filtrelemede her bir zaman serisi elemanı bir önceki örnekten çıkarılarak yeni bir zaman serisi elde edilmektedir. Bu özellik denklem 3.7'de verilmiştir.

$$\Delta i(k) = i(k+1) - i(k) \quad (3.7)$$

İkinci özellik çıkarım yönteminde ise Hilbert dönüşümünden faydalanılmıştır. Akım sinyalinin Hilbert dönüşümü alındıktan sonra yeni özellik sinyali aşağıdaki eşitlik ile elde edilmiştir.

$$E(t) = |A(t)| = |I(t) + jH(t)| \quad (3.8)$$

Özellik vektörü elde edildikten sonra, bu vektörün faz uzayı oluşturulur. Faz uzayı lineer olmayan zaman serileri analizine dayalıdır. Faz uzayı bir zaman serisini belirlenen zaman gecikmesi ve gömülme boyutuna göre farklı boyutlarda haritalar. $E(t)$ sinyalinin faz uzayı denklem 3.9'a göre hesaplanır.

$$F = \begin{bmatrix} E_{1+(b-1)\tau} & \dots & E_{1+\tau} & E_1 \\ E_{2+(b-1)\tau} & \dots & E_{2+\tau} & E_2 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ E_N & \dots & E_{N-(b-2)\tau} & E_{N-(b-1)\tau} \end{bmatrix}_{(N-(b-1)\tau) \times b} \quad (3.9)$$

Burada F faz uzayı matrisidir ve bu matrisin her bir satırı faz uzayında bir noktayı gösterir. Her sütun belli bir zaman gecikmesi τ ve gömülme boyutu b 'ye göre orijinal zaman serisinin geciktirilmiş versiyonudur. Zaman gecikmesi karşılıklı bilgi yöntemine göre, gömülme boyutu ise yanlış en yakın komşu algoritması ile belirlenmektedir.

3.2.3. Genetik Algoritma Tabanlı Detektör Üretimi

Akım sinyalinden gerekli özellikler çıkarılıp faz uzayı oluşturulduktan sonra sağlam motor verilerine göre detektörlerin üretimi yapılmalıdır. Negatif seçim algoritmasındaki en önemli problem detektörlerin üretilmesidir. Bu çalışmada üretilen detektörlerin optimize edilmesi sağlanarak öz olmayan uzayı maksimum bir şekilde kapsayacak detektörlerin üretilmesi için bir yöntem sunulmaktadır. Yöntem detektör üretimi için genetik algoritmayı kullanmaktadır. Bu çalışmada reel-değerlikli negatif seçim göz önüne alınmıştır. Bu amaçla afinite ölçümü için Öklid mesafesi kullanılmaktadır. Her bir detektör, özellik vektörü ve detektör yarıçapı olmak üzere iki alandan oluşmaktadır. Aşağıda detektör ve veriler ile ilgili bazı tanımlamalar yapılmıştır:

- $\bar{O}$ öz veri kümesini ve $\bar{o}$ ise bu küme içerisindeki herhangi bir elamanı gösterebilir.
- Herhangi bir öz veri kümesi elamanı için $\bar{o}.r$ yarıçapı ve $\bar{o}.f$ ise küme merkezi özelliğini gösterir.
- D detektör kümesi ve d ise bu detektör kümesinde bir eleman olmak üzere her detektörün de bir yarıçapı ($d.r$) ve küme merkezi ($d.f$) vardır.
- Her bir özellik $[0,1]$ aralığında normalleştirilmiştir.

- Bir d detektörü ile bu detektöre en yakın olan s verisi arasındaki mesafe aşağıdaki eşitliğe göre hesaplanmaktadır.

$$mesafe = \left[\min_{\delta \in \tilde{O}} \text{Öklid}(\delta, d) \right] \quad (3.10)$$

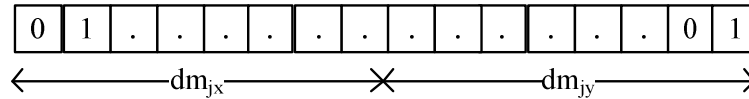
Bir detektörün yarıçapı o detektöre en yakın olan öz veri kümesi elemanına göre belirlenmektedir. Her bir detektör için bu mesafe minimum olmalıdır. Çünkü üretilecek bir detektör öz bir antikora en yakın olacak şekilde seçilmelidir. Detektör yarıçapı denklem 3.11'e göre hesaplanır.

$$d.r = mesafe - \delta.r \quad (3.11)$$

Bu eşitlikte d bir detektör ve $d.r$ bu detektörün yarıçapı olmak üzere, s bu detektöre en yakın öz veri kümesi elemanı ve $\delta.r$ ise bu elemanın yarıçapıdır. Amaç fonksiyonu öz olmayan uzayın maksimum kapsanmasını sağlamalıdır. Bu amaçla denklem 3.12'deki gibi bir amaç fonksiyonu seçilmiştir.

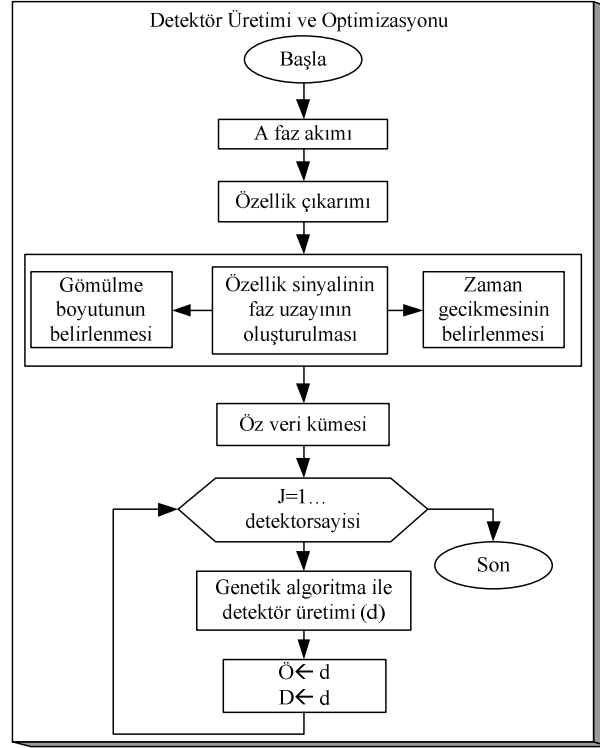
$$\underbrace{F(\delta.r, mesafe)}_{\max} = \frac{mesafe - \delta.r}{\delta.r} \quad (3.12)$$

Genetik algoritma için her bir birey ikili kodlama ile kodlanmıştır. Kodlama detektörlerin küme merkezlerinden oluşmaktadır. Şekil 3.6'da kullanılan kodlama gösterilmiştir. Her bir küme merkezi için 8 bitlik kodlama kullanılmıştır. Dolayısıyla her bir kromozom 16 bit uzunluğundadır.



Şekil 3.6. Kullanılan kodlama

Genetik algoritmanın popülasyon boyutu 50, mutasyon olasılığı %5 ve çaprazlama olasılığı %40 olarak seçilmiştir. Ayrıca çaprazlama ve mutasyon sonucu en iyi birey popülasyondan kaybedilebileceği için, en iyi bireyin sonraki popülasyonlarda da olmasını sağlamak için elitizm (seçicilik) uygulanmıştır. Önerilen algoritmanın akış şeması Şekil 3.7'de verilmiştir.



Şekil 3.7. Önerilen yöntemin eğitim aşaması

Şekil 3.7’de gösterilen negatif seçim algoritmasının eğitim aşamasında özellik çıkarımı daha önce anlatılan ilk fark filtrelemesi veya Hilbert dönüşümüne dayalı yapılmaktadır. Daha sonra sağlam motora ait bu özellik sinyalinin faz uzayını oluşturmak için gerekli olan zaman gecikmesi ve gömülme boyutu belirlenir. Genetik algoritma ile her defasında bir detektör üretilmektedir. Her üretilen detektör hem detektör kümesine hem de öz veri kümesine eklenmektedir. Böylece daha önce üretilen bir detektörün bir sonraki aşamada üretilmesi engellenmiştir.

Negatif seçimin test aşamasında ise eğitim aşamasında genetik algoritma ile üretilen detektörler arıza teşhisi için kullanılacaktır. Bu aşamada iki önemli özellik kullanılmaktadır. Bunlardan birincisi her bir arızalı durum için bütün detektörler içerisinde aktifleşenlerin sayısı, diğer özellik ise her bir detektörün aktifleşme oranıdır. Ayrıca izleme aşamasında detektörlerin sağlam ve arızalı verileri tespit oranları hesaplanarak metodun performansı ölçülmüştür. Bu performans ölçümü Acc^+ ile elde edilmektedir.

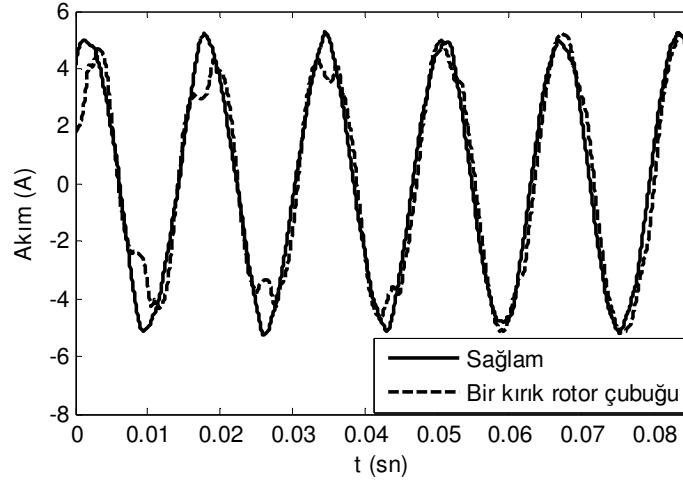
3.2.4. Deneysel Sonuçlar

Önerilen yöntem üç fazlı asenkron motorda bir, iki ve üç kırık rotor çubuğu arızalarını teşhis etmek amacıyla kullanılmıştır. Yöntem, arızaları belirlemek için üç fazlı bir asenkron motorun A fazından alınan akım sinyallerine gereksinim duyar. Akım sinyalleri zaman adımlı sonlu eleman durum uzayı metodu ile elde edilmiştir [110]. Yöntemin performansını değerlendirmek amacıyla her bir motor durumu için 20 örnek bulunmaktadır. Kullanılan asenkron motorun özellikleri Tablo 3.1’de detaylı olarak verilmiştir.

Tablo 3.1. Simülasyonda kullanılan asenkron motor karakteristikleri

Özellik	Değer
Besleme gerilimi	208 V
Frekans	60 Hz
Rotor çubuklarının sayısı	34
Kutup sayısı	2
Güç	1.2 hp

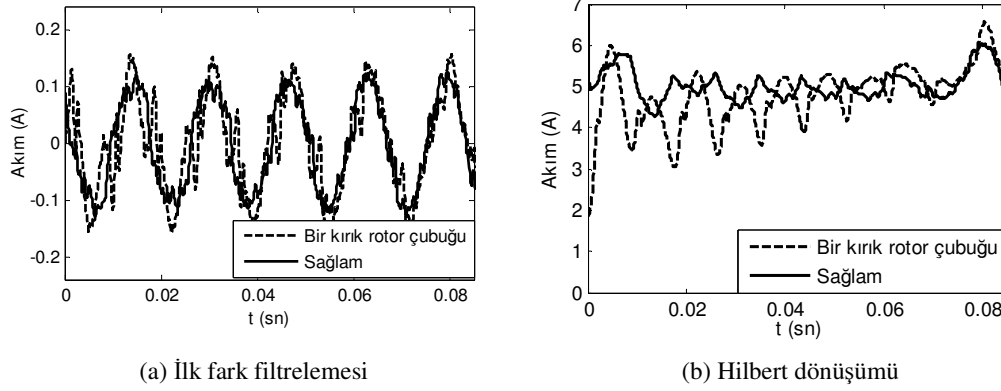
Şekil 3.8’de sağlam motor ve bir kırık rotor çubuğuna sahip motor durumu için akım sinyali verilmiştir.



Şekil 3.8. Sağlam ve bir kırık rotor çubuğu motor akım sinyali

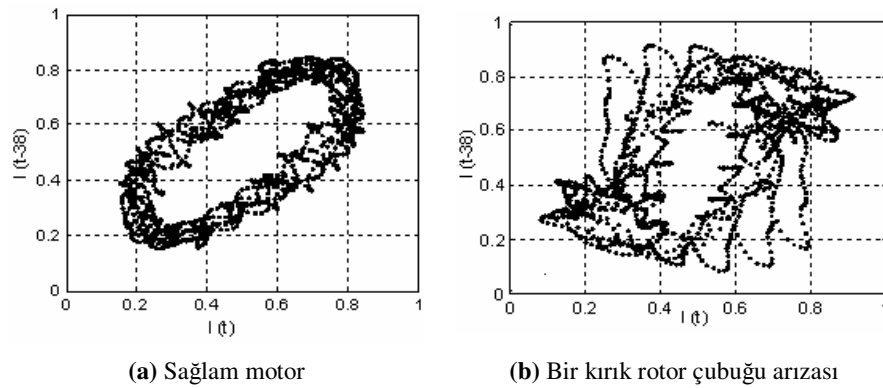
Özellik çıkarımı için ilk fark filtrelemesi ve Hilbert dönüşümü tabanlı yöntemler sunulmuştur. Her iki yöntem ile elde edilen yeni özellik sinyalleri ayrı ayrı

değerlendirilerek yöntemin performansı ölçülmektedir. Şekil 3.9’da her iki yöntem ile elde edilen sağlam motor ve kırık rotor çubuğu arızası ile ilgili özellik vektörleri gösterilmiştir.



Şekil 3.9. Sağlam ve arızalı motor özellik sinyalleri

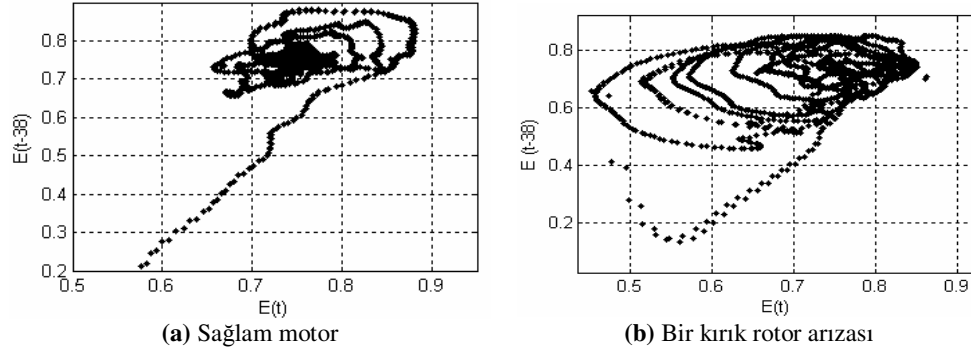
Faz uzayı için gömülme boyutu ve zaman gecikmesi iki önemli parametredir. Eğer bunlar uygun seçilmezse zaman serilerinin farklılığı faz alanında ortaya çıkmaz. Faz uzayı için gömülme boyutu yanlış en yakın komşu algoritmasına göre ve zaman gecikmesi ise karşılıklı bilgi yöntemine göre bulunmuştur. Gömülme boyutu her iki özelliğin faz uzayı için de 2 olarak alınmıştır. Zaman gecikmesi ise 38 olarak karşılıklı bilgi yöntemine göre elde edilmiştir. Şekil 3.10’da ilk fark filtrelemesi ile elde edilen özellik sinyali için sağlam ve bir kırık rotor çubuğuna sahip motorun faz uzayları gösterilmiştir.



Şekil 3.10. İlk fark filtrelemesi ile oluşturulan özellik sinyallerinin faz uzayları

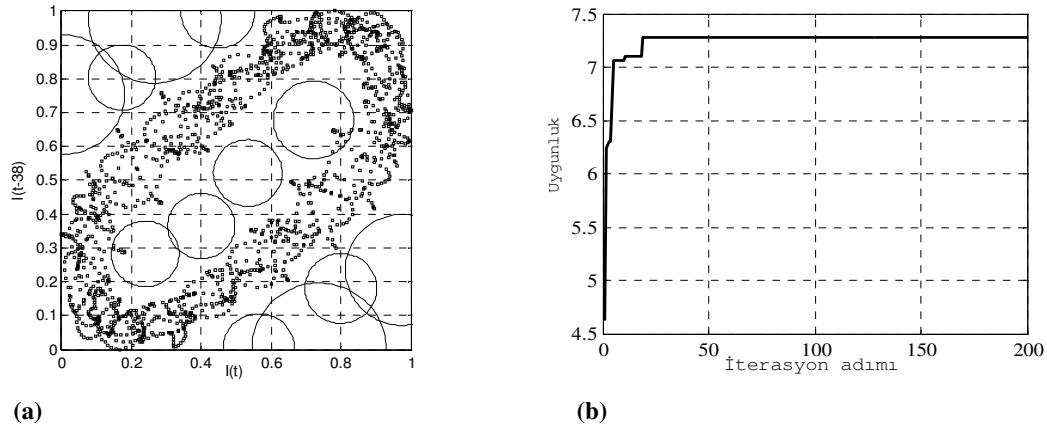
Faz uzayındaki bütün verileri kullanmak negatif seçim algoritmasının eğitim aşamasını yavaşlatır. Faz uzayındaki veriler birbirine yakından olduğundan ve sağlam faz

uzayı verileri için bir yarıçap değeri belirlendiği için k-ortalamlar kümeleme metodu ile veri azaltma işlemi yapılmıştır. Şekil 3.11’de Hilbert dönüşümü ile elde edilen özellik sinyali için faz uzayı gösterilmiştir.



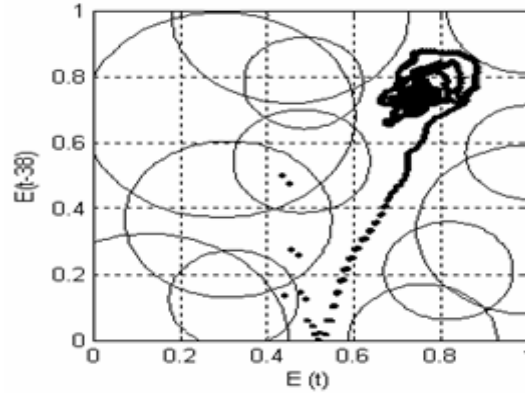
Şekil 3.11. Hilbert dönüşümü tabanlı çıkarılan özelliklerin faz uzayları

Sağlam motor için elde edilen küme merkezlerinden her birinin yarıçapı 0.07 olarak alınmıştır. Üretilcek detektör sayısı 12 olarak belirlenmiştir. Şekil 3.12’de ilk fark filtrelemesi için sağlam motor verilerine göre üretilen detektörler ve ilk detektör üretiminde genetik algoritmanın performans değişimi verilmiştir.



Şekil 3.12. Genetik algoritma ile üretilen detektörler ve algoritmanın performansı

Hilbert dönüşümü tabanlı elde edilen özellik sinyali için sağlam durum faz uzayı ve genetik algoritma ile üretilen detektörler Şekil 3.13’te gösterilmiştir.



Şekil 3.13. Sağlam motor için faz uzayı ve genetik algoritma ile üretilen detektörler

Test aşamasında önerilen yöntemin performansı denklem 3.13'e göre belirlenmiştir. Başlangıçta eğitim aşamasındaki sağlam motor verisi ve üretilen detektörler ile gelen test verisine göre doğruluk oranları belirlenmiştir. Daha sonra farklı sağlam motor verileri, üretilen detektör kümesi ve farklı test örnekleri ile algoritmanın performansı ölçülmüştür. Tablo 3.2'de Hilbert dönüşümü tabanlı özellik çıkarımına göre bazı test sonuçları verilmiştir.

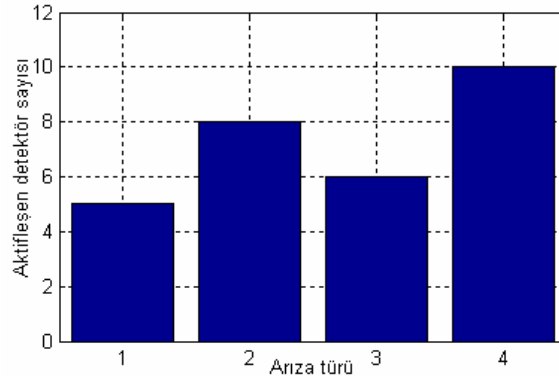
Tablo 3.2. Yöntemin farklı veriler üzerinde test sonuçları

Arıza türü	Test 1			Test 2		
	DP	YN	Acc ⁺ (%)	DP	YN	Acc ⁺ (%)
Bir kırık rotor çubuğu	231	5	97.88	578	22	96.33
İki kırık rotor çubuğu	1456	5	99.65	1461	33	97.79
Üç kırık rotor çubuğu	571	5	99.13	657	10	98.50

Tablo 3.2'de Test1 sütununda sağlam motor verileri olarak eğitim aşamasında detektör üretimi için kullanılan veriler kullanılmıştır. Şekil 3.12'den de görüldüğü gibi bu verilerde 5 adet sağlam veri detektörler tarafından tanınmaktadır. Bu YN olarak ifade edilmiştir. Detektörler tarafından tespit edilen arızalı durum verileri Test1'in DP sütununda gösterilmiştir. Test 2'de ise elimizde olan farklı sağlam ve arızalı motor verilerine göre yöntemin performansı ölçülmüştür. Dolayısıyla iki veri kümesi birbirinden farklıdır. Aynı yöntem ilk fark filtrelemesi ile elde edilen özellik sinyalinin faz uzayında da uygulanmıştır. Bir, iki ve üç kırık rotor çubuğu verisi üzerinde uygulanan yöntemin sonuçları Tablo 3.3'te verilmiştir. Hilbert dönüşümü tabanlı özellik çıkarımı ile her bir arıza durumu için aktifleşen detektör sayısı Şekil 3.14'te verilmiştir.

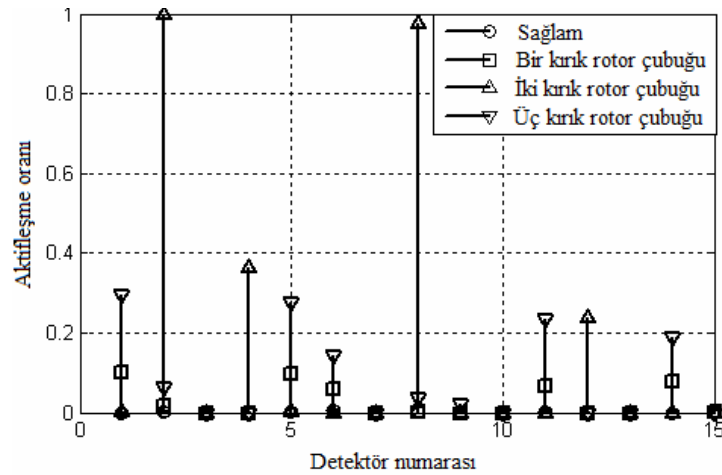
Tablo 3.3. İlk fark filtrelemesi genetik algoritma tabanlı arıza tespiti

Arıza türü	DP	YN	Acc ⁺ (%)
Bir kırık rotor çubuğu	249	88	73.88
İki kırık rotor çubuğu	523	88	85.59
Üç kırık rotor çubuğu	230	88	72.32



Şekil 3.14. Aktif detektör sayıları

Şekil 3.14'te sağlam motor durumundan üç kırık rotora kadar her bir durumda aktifleşen toplam detektör sayıları görülmektedir. Bu şekilde aktifleşen detektörlere göre arızanın türü de belirlenebilir. Ayrıca hangi tür arızanın oluştuğunu belirlemek için aktifleşme oranı kavramı da kullanılmıştır. Aktifleşme oranı detektörler tarafından tespit edilen veri sayısının toplam veri sayısına oranı alınarak hesaplanmaktadır. Bu hesaplama göre izleme esnasında her bir arıza durumu için aktifleşme oranları Şekil 3.15'te verilmiştir.



Şekil 3.15. Her bir arıza türü için detektörlerin aktifleşme oranı

Şekil 3.15'e göre arıza türlerini aktifleşme oranlarına göre belirlemek oldukça kolaydır. Sağlam motor durumunda bütün detektörler 0.1'den çok düşük bir aktifleşme oranına sahiptir. Bir kırık rotor çubuğu arızasında aktifleşme oranı maksimum 0.1 ile 0.2 arasındadır. İki kırık rotor çubuğu arızasında ise 0.9'dan büyük aktifleşme oranları vardır. Aktifleşme oranı üç kırık rotor çubuğu arızasında ise 0.2 ile 0.4 arasında değişmektedir. İlk fark filtrelemesine göre toplam 13 detektör bulunmaktadır. Bir kırık rotor çubuğu arızasında detektörlerin %76.92'si aktifleşirken, iki kırık rotor çubuğu arızasında ise detektörlerin %23.08'i aktifleşmektedir.

Sonuç olarak, önerilen yöntem ile asenkron motorlarda oluşan bir, iki ve üç kırık rotor çubuğu arızaları tespit edilmiştir. Ön işleme için daha önce literatürde de kullanılan ilk fark filtrelemesi ve yeni bir dönüşüm olarak Hilbert tabanlı bir özellik çıkarımı kullanılmıştır. İlk fark filtrelemesi ile elde edilen sinyalin faz uzayı için zaman gecikmesi büyük seçildiğinde iyi sonuçlar almak mümkün olmaktadır. Hilbert dönüşümü ile elde edilen faz uzayları daha iyi sonuçlar vermektedir. Çünkü elde edilen özellik sinyali sağlam ve arızalı durumları birbirinden başarılı bir şekilde ayırt etmektedir. Yapay bağışık sistem bileşeni olan negatif seçim algoritmasında üretilen detektörlerin genetik algoritma ile optimize edilmesi sağlanarak öz olmayan uzayın maksimum kapsanması sağlanmıştır. Ayrıca test aşamasında arıza türünü belirlemek amacıyla detektörlerin aktifleşme oranları kullanılmıştır. Yöntem temel olarak iki avantaj sunmaktadır. Birincisi negatif seçim algoritmasının detektörleri optimize edilerek, hem öz olmayan uzayın maksimum kapsanması hem de üretilen detektörlerin birbirlerinden farklı olması sağlanmıştır. İkinci avantaj ise arıza teşhisi için önerilen Hilbert dönüşümü tabanlı özellik çıkarımıdır. İlk fark filtrelemesinden farklı olarak arızalı ve sağlam durumların daha iyi ayırt edilmesi sağlanmıştır. Özellik sinyali, sinyaldeki bütün farklılıkları pozitif bileşenlerde ve ayırt edilebilir bir şekilde gösterir.

3.3. Arıza Tespiti İçin Kaotik Hibrit Negatif Seçim Algoritması (KHNSA)

Bu kısımda arıza tespiti için hibrit bir negatif seçim algoritması sunulmaktadır. Önerilen yöntem parametre seçimi için kaotik haritaları kullanmaktadır. Orijinal negatif seçim algoritmasında başlangıç popülasyonu ve mutasyon işlemi rastgele uygulanmaktadır. Negatif seçim algoritmasının öz olmayan uzayı maksimum şekilde kapsamı kaotik haritalar ile gerçekleştirilmiştir. Birbirini kapsayamayacak şekilde optimum detektörlerin üretilmesi için klonal seçim optimizasyon yöntemi kullanılmıştır. Arıza teşhis

sistemlerinde eğitim verisi her zaman normal veriyi tamamen kapsamayabilir ve öz ve öz olmayan uzay zamanla değişebilir. Test aşamasında eğitim aşamasında üretilen öz veya öz olmayan herhangi bir detektör bir test verisini tespit edemiyorsa bu veriye en yakın detektör K-en yakın komşu algoritmasına göre bulunur. Bu test verisi üzerinde yeni bir detektör üretilir. Üretilen bu detektörün yarıçapı kendi sınıfı dışında en az veri kapsayana kadar küçültülür. Önerilen Kaotik Hibrit Negatif Seçim Algoritması (KHNSA) kırık rotor çubuğu arızalarının tespiti için kullanılmıştır.

Önerilen hibrit algoritma detektör üretimi ve tespit aşamasından oluşur. Detektör üretimi her iki aşamada da yapılmaktadır. Çünkü test aşamasında tespit edilemeyen veriler için yeni detektörler üretilmektedir. Yöntemin performansını değerlendirmek için eğitim aşamasında üretilen detektörlerden faydalanılmaktadır. Lineer olmayan ve karmaşık sınıflandırma problemlerinde öz ve öz olmayan uzay sürekli değişir. Yeni bir test örneği sisteme verildiğinde ve bu örnek herhangi bir detektör tarafından tespit edilemezse KHNSA bu örneği tespit etmek için algoritmayı uyarlar. Tespit edilemeyen test örnekleri için negatif seçim algoritmasının uyarlanması K-en yakın komşu algoritması ile gerçekleştirilmektedir. Önerilen KHNSA algoritmasının test ve eğitim aşamalarının detayları alt bölümlerde detaylı olarak verilmiştir.

3.3.1. Detektör Üretim Aşaması

Orijinal negatif seçim algoritmasında sadece öz olmayan (anomali veya arızalı durumlar) veri kümeleri için detektörler üretilmektedir. Önerilen KHNSA'da ise hem öz hem de öz olmayan veri kümeleri için detektörler üretilmektedir. Dolayısıyla algoritma eğitim aşamasında hem öz hem de öz olmayan veri kümesine ihtiyaç duyar. Bu aşama orijinal negatif seçim algoritmasına benzemesine rağmen, detektör üretimi iki kez yapılmaktadır. Orijinal negatif seçim algoritmasında başlangıç popülasyonu için detektörler rastgele üretilmektedir. Rastgele üretilen detektörlerin dezavantajı aynı detektörlerin tekrar üretilme olasılığıdır. Önerilen yöntemde detektörlerin başlangıç popülasyonunun üretilmesi kaotik haritalar ile gerçekleştirilmektedir.

Kaos tabanlı algoritmalar olasılık tabanlı aramaya dayalı optimizasyon için kullanılabilir. Kaosun tekrar etmeyen ve rastgele özelliği ile geniş bir arama uzayında hızlı ve etkili bir arama yapılabilir. Bu özelliği sayesinde kaos kuramı karınca koloni [120] ve parçacık sürü optimizasyonu [121] gibi farklı optimizasyon yöntemlerinin

performansının artırılması için kullanılmıştır. Bir kaotik harita kaotik bir durumda çalışan ayrık zamanlı dinamik bir sistemdir. Kaotik bir haritanın genel formülü aşağıda verilmiştir.

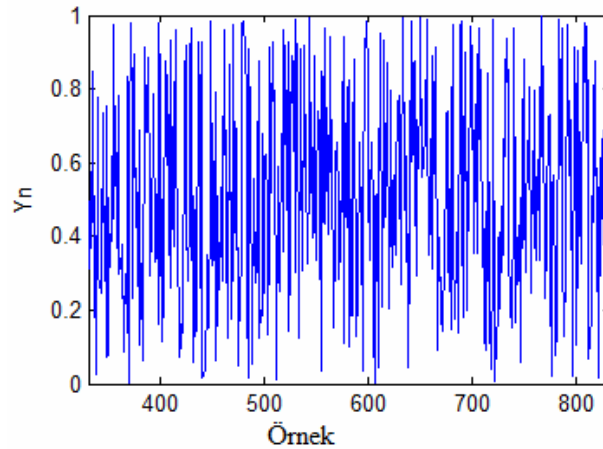
$$x_k = f(x_{k+1}), 0 < x_k < 1, k = 0, 1, 2, \dots \quad (3.13)$$

Kaotik haritalar farklı boyutlarda rastgele değerler üretmek için oldukça kullanışlıdır. Literatürde farklı kaotik haritalar mevcuttur. Bu haritalardan en çok kullanılanları lojistik harita, çadır haritası, çember haritası, Arnold'un kedi haritası [122] ve Sinai haritası [123]'dür. Bu iki haritanın denklemleri aşağıda verilmiştir.

$$\begin{aligned} X_{n+1} &= X_n + Y_n \mod(1) \\ Y_{n+1} &= X_n + 2Y_n \mod(1) \end{aligned} \quad (3.14)$$

$$\begin{aligned} X_{n+1} &= X_n + Y_n + a \cos(2\pi Y_n) \mod(1) \\ Y_{n+1} &= X_n + 2Y_n \mod(1) \end{aligned} \quad (3.15)$$

Arnold kedi haritasında $X_n \in (0, 1)$ ve $Y_n \in (0, 1)$ aralığında değer almaktadır. Sinai kaotik haritasında ise $a=1$ seçildiğinde $[0,1]$ aralığında değerler üretilmektedir. Bu iki harita $[0,1]$ aralığında ilginç rastgele değerler üretir. Belirlenen X_n ve Y_n başlangıç şartları için Sinai kaotik haritası için üretilen değerler Şekil 3.16'da verilmiştir.



Şekil 3.16. Sinai kaotik haritası

Önerilen algoritma ile negatif seçim algoritmasındaki detektörlerin öz/öz olmayan uzayı maksimum kapsamaları için kaotik haritalamadaki düzensizlik ve olasılık tabanlı

özellik kullanılmaktadır. Negatif seçim algoritmasının giriş veri kümesi $[0, 1]$ aralığında normalize edilmiştir. Detektör kümesi üretilirken Sinai ve Arnold kedi haritası kullanılmıştır. İki boyutlu reel özellik uzayında, özellik vektörü Arnold kedi haritası veya Sinai haritası ile üretilmiştir. Detektör üretimi için bir örnek gösterim aşağıda verilmiştir.

$$d.f = \text{Sinai}(X_n, Y_n), \text{ for } n > 0 \quad (3.16)$$

Üç ve daha büyük boyutlu uzayda, bu kaotik haritaların kombinasyonu bir aday detektör üretmek için kullanılmıştır. Detektörün yarıçapı Öklid mesafesine göre hesaplanmaktadır. Bu hesaplama aşağıda verilmiştir.

$$d.r = \left[\min_{\tilde{o} \in \tilde{O}} \text{Öklid}(\tilde{o}, d) \right] \times a \quad (3.17)$$

Detektör üretimi yapılırken önce öz olmayan veri kümesi için detektörler üretilmektedir. Daha sonra öz veri kümesi için detektörler üretilmektedir. Öz olmayan veri kümesi için detektör üretilirken öz veri kümesi için yarıçap kullanılmamıştır. Yarıçap hesaplanırken öz olmayan bir veri kümesi ile en yakın öz örnek arasındaki mesafe a kontrol parametresi ile azaltılmıştır. Bu kontrol parametresi detektörlerin maksimum yarıçapını azaltarak öz veri kümelerinin minimum sayıda kapsanması sağlanmıştır. Böylece detektör ile en yakın öz örnek arasındaki örtüşme önlenmiştir. Önerilen yöntem reel değerler ile çalışan arıza tespit problemine uygulandığından uzaydaki bir noktanın yeri mutasyon operatörü kullanılarak taşınmıştır. Bir detektörün her bir özelliği mutasyon olasılığına göre değişmektedir. Kaotik haritanın bir boyutu mutasyon olasılığını hesaplamak için kullanılır. Her bir özellikteki değişim kaotik harita kullanılarak sağlanır. Mutasyon operatörünün adımları Şekil 3.17’de verilmiştir.

```

b ← 0
While b=0 do
    For each d.f(i) in d.f
        rn ← Xn
        If (rn < Pmut)
            mr ← random(kmap(-Xn, Yn))
            d.f(i) ← d.f(i) + mr
            b ← 1
        end if;
        If ((d.f(i) < 0) || (d.f(i) > 1))
            d.f(i) ← 0
        end if;
    End For
End while

```

Şekil 3.17. Kaotik mutasyon operatörü

Şekil 3.17’de r_n değeri kaotik harita kullanılarak $[0, 1]$ aralığında rastgele üretilmiştir. $kmap(-X_n, Y_n)$ fonksiyonu kaotik harita kullanarak $-X_n$ ile Y_n arasında rastgele bir değer üretir. Mutasyon süreci detektörün en az bir özelliği değiştirilene kadar devam eder. Her detektörün afinitesi iki kritere göre hesaplanmaktadır: tespit sayısı (ts) ve alt tespit sayısı (ats). Birinci kriter (ts) başka bir detektör tarafından tespit edilmeyen ve afinitesi hesaplanan detektör tarafından tanınan öz olmayan veri örneği sayısıdır. İkinci kriter (ats) ise hem geçerli detektör hem de en az başka bir detektör tarafından tanınan örnek sayısıdır. İkinci kriter (ats) üretilen detektörler arasındaki örtüşmeyi önlemek için kullanılmıştır. Bu kriterlere göre detektörlerin afinitesi aşağıdaki gibi hesaplanır.

$$d.af = d.ts + d.ats \quad (3.18)$$

Eğitim aşamasında üretilen detektör kümesinin kapsamı (K) öz ve öz olmayan özellik sinyali kullanılarak hesaplanmaktadır. Eğitim aşamasında üretilen detektörler tarafından tanınan öz ve öz olmayan örneklerin sayısı S ve NS ile ifade edelim. Buna göre detektör kümesinin kapsamı aşağıdaki gibi hesaplanır.

$$K = \frac{NS}{S + NS} \times 100 \times \left(\frac{NS}{m}\right) \quad (3.19)$$

Yukarıdaki denklemde m öz olmayan veri kümesindeki örnek sayısıdır. Kapsamanın %100 olması idealdir. Fakat gerçekte böyle bir kapsama mümkün değildir. İdeal şartlarda kapsama performansı %100'den daha düşük olup ve sabit bir kapsamaya erişilinceye kadar algoritmanın çalıştırılması gerekir. Bu aynı zamanda algoritmanın durdurma kriteridir. Önerilen KHNSA algoritmasının kapsama performansı ortalama 200 adımdan sonra sabit bir değere ulaşmaktadır. Üretilen detektörler mutasyondan sonra çoğalma işlemine tabii tutulur. Bir detektörün klonu seçim işlemi ile geçerli detektörün yerine geçebilir. Eğer klonu oluşturulan detektörün afinitesi geçerli detektörün afinitesinden daha büyük ise geçerli detektör yenisi ile değiştirilir. Bu süreç klon sürecinin verilen adım sayısı kadar devam ettirilir ve her adımda en iyi detektör korunur. Şekil 3.18'de detektör kümesini elde etmek için detektör üretim algoritması gösterilmektedir. Algoritmanın eğitim parametreleri Tablo 3.4'de verilmiştir.

```

For i=1 to adım_sayısı
    d ← Kaotik harita tabanlı aday detektör üret
    d.r ← Aday detektörün yarıçapını hesapla
    d.afinite ← detektörün afinitesini hesapla
    For j=1 to klon_sayısı
        C ← Mutasayon(d)
        C.r ← Klonun yarıçapını hesapla
        C.afinite ← Klonun afinitesini hesapla
        If (C.afinite > d.afinite)
            d ← C
        End If
    End For
    D ← DUd
End For

```

Şekil. 3.18. Detektör üretim aşaması

Tablo 3.4. KHNSA algoritmasının eğitim parametreleri

Özellik	Tanımlama	Değer
Pmut	Mutasyon olasılığı	0.2
Numclones	Klon sayısı	100
Maxiteration	Maksimum iterasyon sayısı	200

Orijinal negatif seçim algoritmasında üretilen detektörler sadece herhangi bir öz olmayan veri örneği geldiğinde aktifleşir. Dolayısıyla öz veri kümesini tanıyan detektörler üretilmemektedir. Bu çalışmada hem öz hem de öz olmayan veri örnekleri için detektörler üretilmektedir. Test aşamasında bu detektörler kullanılarak hangi sınıftan ne kadar veri tanındığı belirlenmektedir.

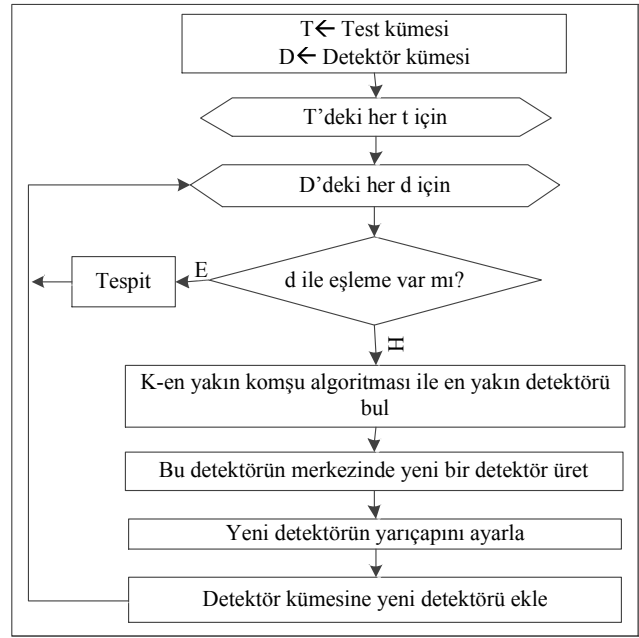
3.3.2. Tespit Aşaması

Tespit aşamasında öz ve öz olmayan detektörler sağlam veya arızalı bir durumu belirlemek için kullanılır. Fakat bazı durumlarda bir test örneği hiçbir detektör tarafından tanınmayabilir. Bu şekildeki veri örneklerinin sınıfını belirlemek için K-en yakın komşu algoritması kullanılmaktadır. Test aşamasında gelen herhangi bir veri örneği eğitim aşamasında üretilen hiçbir detektör tarafından tanınmıyorsa K-en yakın komşu algoritmasına göre en yakın detektör bulunur ve bu veri örneğinin üzerinde detektörün bir kopyası oluşturulur. Ağırlıklı oylama yöntemi, bir test örneğine en yakın komşuyu bulmak için kullanılmıştır. Bu yöntemaşağıda verilmiştir.

$$sdc_1 = \sum_{i=1}^k \frac{1}{Öklid(d(i), test(j))^2}, \text{ for } S$$

$$sdc_2 = \sum_{i=1}^k \frac{1}{Öklid(d(i), test(j))^2}, \text{ for } NS$$
(3.20)

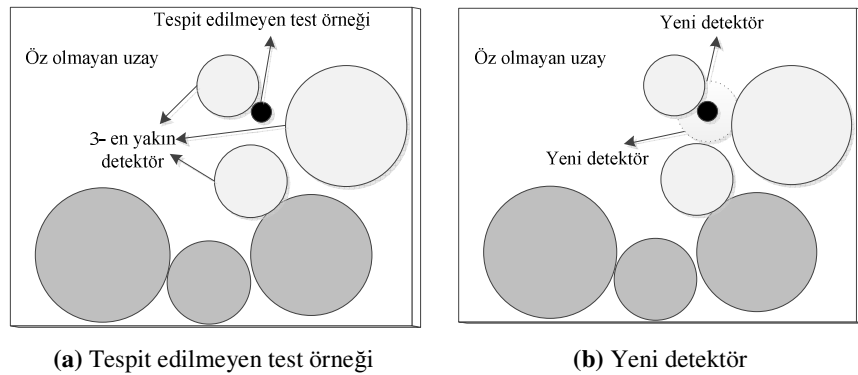
Toplama sembolünde kullanılan k değeri K-en yakın komşu algoritmasının parametresidir. İki denklemden elde edilen sdc1 ve sdc2 değerleri karşılaştırılarak büyük olanı bulunur. Elde edilen bu değer test örneğinin ait olduğu sınıfı belirler. Test aşaması Şekil 3.19'da verilmiştir.



Şekil 3.19. Tespit aşaması

Şekil 3.19’da herhangi bir detektör tarafından tespit edilemeyen bir test örneği için en yakın detektör ve detektörün sınıfı K-en yakın komşu algoritması ile elde edilmektedir. Yeni detektör bu veri örneği üzerinde oluşturulmaktadır. Fakat yeni detektör farklı sınıftan verileri de kapsayabilir. Bu yüzden bu detektörün yarıçapı ayarlanmalıdır. Bu durum ile ilgili bir örnek Şekil 3.20’de verilmiştir. K-en yakın komşu algoritmasındaki k parametresi 3 olarak seçilmiştir. Detektörün yarıçapı aşağıdaki gibi ayarlanmaktadır.

$$d.r = d.r - d.r * \frac{1}{w} \quad (3.21)$$



Şekil 3.20. Yeni test örneği için detektörlerin uyarlanması

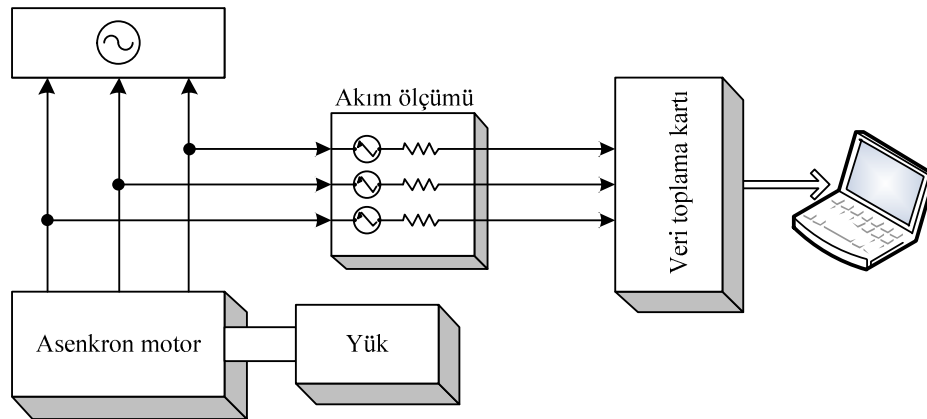
Yukarıdaki denklemde w yarıçapın azaltma parametresidir ve 5 olarak alınmıştır. Metodun performansı karmaşıklık matrisi ile doğrulanmıştır. Karmaşıklık matrisinden doğru pozitif oranı (Acc^+), doğru negatif oranı (Acc^-), yanlış alarm oranı (Fa) ve G-ortalama değerleri hesaplanarak performans karşılaştırması yapılmıştır.

3.3.3. Deneysel Sonuçlar

Önerilen KHNSA asenkron motorlarda oluşan kırık rotor çubuğu arızalarının belirlenmesi için kullanılmıştır. Yöntem üç fazlı sincap kafes bir motordan deneysel olarak alınan sağlam ve kırık rotor çubuğu arızası verileri üzerinde uygulanmıştır. Önerilen yöntem, aynı şartlar altında bulanık mantık ve yapay sinir ağları ile karşılaştırılmıştır. Deneyde kullanılan asenkron motorun özellikleri Tablo 3.5'te verilmiştir. Motor akım sinyalleri sağlam ve bir kırık rotor çubuğu arızası için tam yükte çalışma altında alınmıştır. Veri toplama için kullanılan deney düzeneğinin şeması Şekil 3.21'de verilmiştir.

Tablo 3.5. Deneyde kullanılan asenkron motorun özellikleri

Özellik	Değer
Güç	0.37 KW
Tam yük akımı	1.2 A
Besleme frekansı	50 Hz
Kutup sayısı	4
Rotor slot sayısı	22
Tam yük hızı	1390 rpm
Besleme gerilimi	380 V



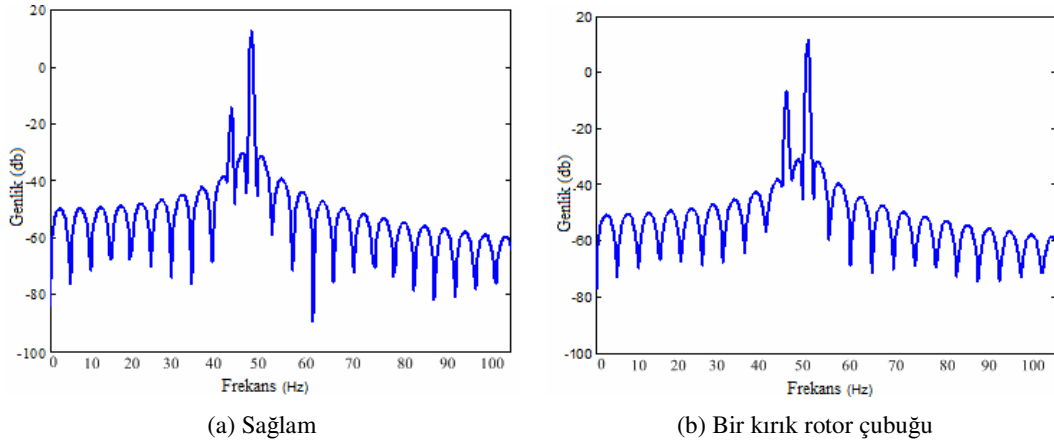
Şekil 3.21. Deney düzeneği şeması

Kırık rotor çubuğu arızalarını belirlemek için motor akım imza analizinden faydalanılmıştır. Motor akım imza analizi motor akımının bir fazından elde edilen veriler üzerinde Fourier dönüşümünü uygulamaktadır. Faz akımının spektrumu ayırık zamanlı Fourier dönüşümü ile hesaplanmıştır. Pencereleme fonksiyonu olarak Hanning penceresi kullanılmıştır. Akım sinyalinin örnekleme oranı 8 000 örnek/saniye olarak seçilmiştir. Elde edilen spektrumda kırık rotor çubuğu ile ilgili iki frekans bileşeni incelenmiştir. Bu frekanslar 50 Hz hat frekansından düşük ve yüksek olan yan bantlar olup, $f_{bl} = (1 - 2s)f_h$ ve $f_{bu} = (1 + 2s)f_h$ ile ifade edilirler. Bu frekans bileşenlerinin genel gösterimi aşağıda verilmiştir.

$$f_b = (1 \pm 2ks)f_h, k=1,2 \quad (3.22)$$

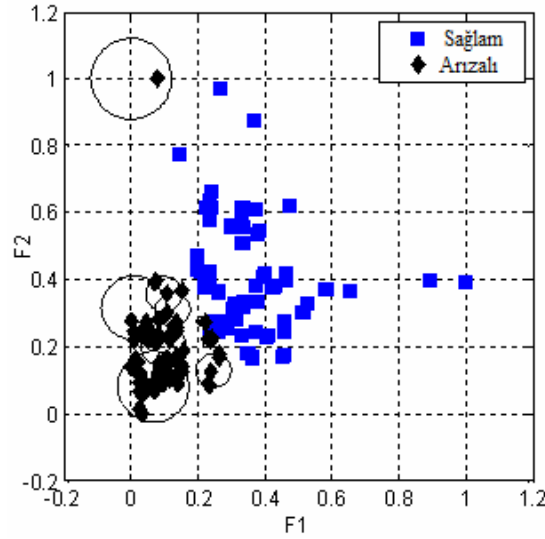
Denklemden f_b kırık rotor çubuğu ile ilgili yan bant frekansını gösterirken f_h ise hat frekansını gösterir. Hat frekansının genliği S_H , düşük ve yüksek yan bant bileşenlerinin genliği S_A ve S_U ile gösterilmek üzere kırık rotor çubuğunun tespiti ile ilgili iki özellik aşağıdaki gibi hesaplanır. Sağlam ve arızalı spektrumlar Şekil 3.22’de verilmiştir.

$$\begin{aligned} F_1 &= S_H - S_A \\ F_2 &= S_H - S_U \end{aligned} \quad (3.23)$$



Şekil 3.22. Sağlam ve arızalı motor akım spektrumu

Eğitim ve test aşamasında kullanılan sağlam ve arızalı veri kümelerinin boyutu 70 adet örnekten oluşmaktadır. Sağlam ver arızalı verilerin görünümü ve üretilen detektörler Şekil 3.23’te verilmiştir.



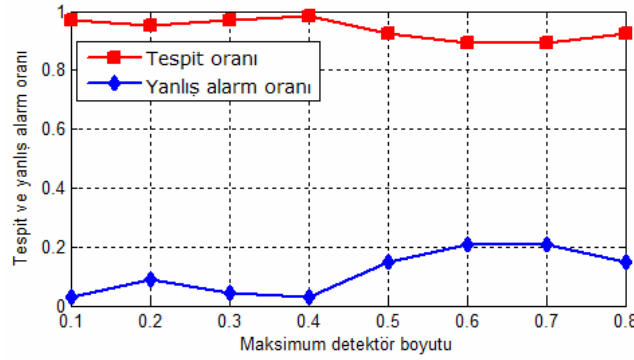
Şekil 3.23. Arızalı veriler için üretilen detektörler

Başlangıç detektörlerini üretmek için Sinai kaotik haritası kullanılmıştır. Algoritmada kullanılan r_n değeri Sinai kaotik haritasının X_n bileşeni kullanılarak üretilmiştir. Taşıma veya mutasyon oranı Arnold kedi haritası ile gerçekleştirilmektedir. Tablo 3.6’da farklı parametrelere göre önerilen yöntemin performansı değerlendirilmiştir. Her bir durum için uygulanan farklı deneylerde sınıflardan biri sağlam diğeri arızalı alınmıştır. Deneylerde eğitim veri kümesi sağlam ve arızalı veri kümelerinin belirli sayıdaki örneklerinden oluşmaktadır. İlk deneyde sağlam veri kümesinin tamamı ve arızalı veri kümesinin %70’i alınmıştır. İkinci deneyde ise bu durumun tersi söz konusudur. Detektörler sağlam ve arızalı veri kümeleri için üretilmiştir.

Tablo 3.6. Farklı parametrelere göre KHNSA’nın performansı

Eğitim kümesi	$r_{\max}$	Acc^+	Acc^-	G-ortalama	Fa	Eğitim detektör sayısı	Test detektör sayısı
Kırık rotor çubuğu 70%	0.15	96.88	100	98.42	0	33	64
	0.25	93.94	100	96.92	0	20	34
	0.35	91.17	100	95.48	0	15	27
Sağlam 70%	0.15	98.53	97.06	97.79	2.94	35	71
	0.25	98.53	97.06	97.79	2.94	22	42
	0.35	97.06	95.60	96.32	4.41	16	25

Tablo 3.6'ya göre eğer detektörlerin maksimum boyutu ($r_{\max}$) küçük seçilirse eğitim aşamasında üretilen detektör sayısı artar. Test aşamasında gelen test verileri eldeki detektörler ile tespit edilemiyorsa yeni detektörler K- en yakın komşu algoritmasına göre üretilmektedir. Tespit algoritmasının maliyet fonksiyonu iki kritere göre belirlenmektedir. Bu kriterler arızalı durumu tespit eden tespit oranı ve sağlam durumda bir alarm üreten yanlış alarm oranıdır. Tespit oranı ve yanlış alarm oranı farklı maksimum detektör yarıçaplarına göre Şekil 3.24'te verilmiştir.



Şekil 3.24. Maksimum detektör boyutuna göre tespit ve yanlış alarm oranı

Önerilen KHNSA yönteminin sağlam ve arızalı durumları tespit başarımı aynı şartlar altında bulanık mantık ve yapay sinir ağları ile karşılaştırılmıştır. Yapay sinir ağı iki giriş ve iki çıkışa sahip olup orta katmanda 5 nöron bulunmaktadır. Tablo 3.7'de eğitim ve test aşamasında üç yöntemin tespit oranı verilmiştir.

Tablo 3.7. KHNSA'nın diğer akıllı teknikler ile karşılaştırılması

Yöntem	Eğitim (%)	Test (%)
Yapay sinir ağı [124]	98.89	98.00
KHNSA	98.53	97.79
Bulanık mantık [125]	98.11	97.67

Tablo 3.7'de görüldüğü gibi önerilen yöntem bulanık mantık ve yapay sinir ağı ile yakın performans göstermektedir. Test aşamasında yeni bir örnek herhangi bir öz veya öz olmayan detektör tarafından kapsamıyorsa bu örneği tespit etmek için yeni detektör üretilmektedir. Bu özellik negatif seçim algoritmasının performansını arttırarak yeni örnekler için kendini uyarlayabilmesini sağlamıştır. Uygulama sonuçları KHNSA yönteminin değişen öz ve öz olmayan uzayda iyi sonuçlar verdiğini göstermektedir.

3.4. Bölüm Değerlendirmesi ve Sonuçlar

Bu bölümde arıza tespiti için negatif seçim algoritmasını kullanan iki yöntem sunulmuştur. İlk yöntem arıza belirleme için motor akım sinyalinin elde edilen özellik sinyalinin faz uzayını kullanmaktadır. Detektörlerin faz uzayında öz olmayan alanı maksimum bir şekilde kapsamaları için genetik algoritmadan faydalanılmıştır. Önerilen yöntemin performansı faz akımına uygulanan ilk fark filtrelemesi ve Hilbert dönüşümü ile elde edilen iki farklı özellik sinyali için değerlendirilmiştir. Deneysel sonuçlar Hilbert dönüşüm tabanlı özellik sinyali kullanıldığında arızalı faz uzaylarının daha iyi ayırt edilebildiğini göstermektedir.

Önerilen ikinci yöntem olan KHNSA arıza teşhisinde yeni bir yaklaşım sunmaktadır. Yapay bağışık sistemlerin anomali tespitindeki yetenekleri arıza teşhisine uyarlanmıştır. Kırık rotor çubuğu arızalarının tespiti için motor akım imza analizi ile elde edilen iki özellik negatif seçim algoritması ile değerlendirilerek arızaların tespiti sağlanmıştır. Detektör üretimi hem eğitim hem de test aşamasında yapılmaktadır. Eğitim aşamasındaki detektörlerin üretimi kaotik haritalar ve klonal seçim algoritması ile yapılmıştır. Seçilen amaç fonksiyonu ile birbiri ile örtüşmeyen ve öz olmayan verileri maksimum kapsayan detektörlerin üretimi sağlanmıştır. Yöntem anomali tespitine uygulanabilmekte ve sınıflandırma problemleri için kolaylıkla değiştirilebilmektedir. Yöntemin avantajları aşağıdaki gibi sıralanabilir.

- Minimum sayıda detektör üretilerek öz ve öz olmayan uzayın maksimum kapsanması sağlanmıştır.
- Detektörlerin birbirini kapsamaları minimuma indirgenmiştir.
- Eğitim aşamasında üretilen detektörler ile tespit edilemeyen örnekler için yeni detektörler test aşamasında K-en yakın komşu algoritması ile üretilmiştir.
- Kaotik haritalar yardımıyla detektörlerin başlangıç popülasyonunun çeşitliliği artırılarak mutasyon işleminde rastgelelik sağlanmıştır.

Sonuç olarak, birinci yöntemde optimize edilen detektörler ile öz olmayan uzayın maksimum kapsanması sağlanarak arızalı faz uzaylarının belirlenmesi sağlanmıştır. İkinci yöntemde ise detektörlerin öz olmayan verileri maksimum şekilde kapsamaları sağlanmıştır. Test aşamasında detektörler tarafından tespit edilemeyen bir veri geldiğinde bu veriye en yakın detektör bulunarak verinin geldiği noktada yeni detektör oluşturulur. Böylece zamanla değişen durumlarda da arıza tespiti yapılabilir.

4. YAPAY BAĞIŞIKLIK TABANLI ARIZA TEŞHİS YÖNTEMLERİ

Farklı arıza türlerinin veya aynı arıza türünün farklı seviyelerinin tespiti ve birbirinden ayırt edilerek sınıflandırılması problemi arıza teşhisi ve durum izleme alanlarının süregelen bir konusudur. Farklı arıza durumlarının sınıflandırılması, her bir durumun sınırlarının iyi belirlenmesine bağlıdır. Akıllı hesaplama teknikleri eğitim aşamasında kullandıkları veriler ile farklı durumları modelleyebilir. Eğitim aşamasında çevrimdışı olarak öğrenilen parametreler test aşamasında gelen verilerin hangi durumlara ait olduğunu belirlemek için kullanılır. Akıllı hesaplama dayalı yöntemler kullanılarak sadece arızaların sınıflandırılması yapılmayıp aynı zamanda arızanın şiddeti de belirlenebilmektedir. Bu yöntemlerin bütün arıza durumlarını kapsaması ve iyi bir arıza teşhis doğruluğu sağlaması için belirli parametrelerinin iyi seçilmesi gerekmektedir.

Tezin bu bölümünde arıza teşhisi için üç farklı yöntem sunulmaktadır. Birinci yöntem arıza teşhisi için destek vektör makinalarını kullanmaktadır. Üç faz akım sinyali üzerinden çıkarılan yeni özellik sinyali ile farklı çalışma durumları için kırık rotor çubuğu arızaları belirlenmektedir. Destek vektör makinalarındaki çekirdek ve ceza parametreleri çok amaçlı klonal seçim optimizasyonu ile ayarlanmıştır. İkinci yöntem sürü öğrenme mekanizmasını bağışık örüntü tanımada kullanmaktadır. Yapay bağışık sistemlerde rastgele uygulanan mutasyon işlemi sürü öğrenmesine dayalı olarak gerçekleştirilmiştir. Her bir klon gideceği hedefi bildiğinden algoritmanın yakınsama performansı artırılmıştır. Önerilen yöntem akım sinyali üzerinden elde edilen istatistiksel özellikler üzerinde ve destek vektör makinalar için elde edilen özellik sinyallerine uygulanmıştır. Üçüncü yöntem ise stator ve rotor arızalarının birlikte tespiti için uyarlamalı bir bağışık sistem önermektedir. Üç faz akım sinyali üzerinden stator ve rotor arızası için yeni özellik sinyalleri elde edilmektedir.

4.1. Arıza Teşhisi için Bağışık Sistem Tabanlı Destek Vektör Sınıflandırıcı

Kırık rotor çubuğu arızalarının teşhisi için önerilen yöntem motorun üç faz akımını kullanır. Arıza teşhisi için park vektör bileşenleri üzerinden yeni bir özellik sinyali elde edilmiştir. Alınan üç faz akımından gerekli ön işleme adımları ile özellik vektörü elde edildikten sonra, lineer olmayan zaman serisi analizi ile zaman serisinin faz uzayı oluşturulur. Destek vektör makinaların çekirdek ve ceza parametreleri çok amaçlı bağışık

sistem ile ayarlanmaktadır. Önerilen bağışık sistem aynı zamanda destek vektör sayısını da minimum yapmaktadır. Sağlam ve arızalı özellik sinyallerinin faz uzayı, destek vektör makinalara verilerek destek vektör makinaların eğitimi gerçekleştirilmektedir. Sağlam ve arızalı motor verilerinin faz uzayı dağılımları destek vektör makinalar ile sınıflandırılarak arızalar teşhis edilmektedir.

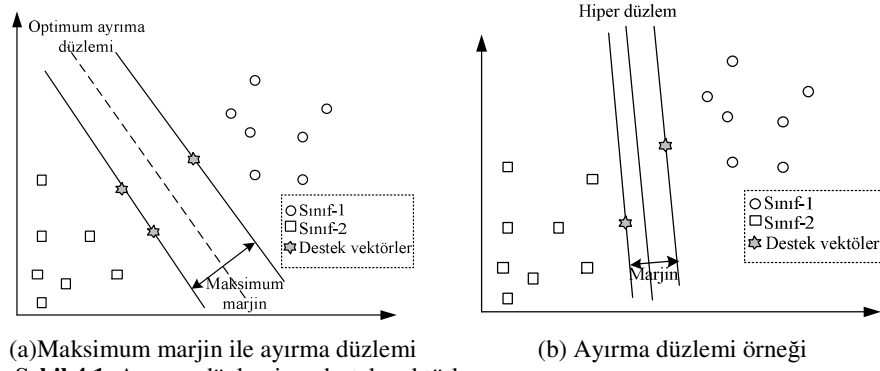
4.1.1. Destek Vektör Makinalar (DVM)

Destek vektör makinalar, doğrusal ve doğrusal olmayan sınıflandırma problemlerinin çözümünde kullanılan eğitici bir öğrenme algoritması olarak tanımlanır [126]. DVM iki sınıflı bir sınıflandırıcıdır. Yüksek boyutlu uzayda x_i ($i=1\dots k$) giriş verisi -1 sınıf etiketi ile birinci sınıfa veya +1 sınıf etiketi ile ikinci sınıfa ait olabilir. DVM'nin amacı maksimum bir aşırı düzlem ile iki sınıfın bir birinden ayırt edilmesini sağlamaktır. Bu düzlem sınıflandırma problemine göre doğrusal veya doğrusal olmayan bir yapıda olabilir. Bulunan aşırı düzlem ayırma düzlemine en yakın veriler arasındaki mesafenin maksimum olmasını garanti eder [127]. Eğer giriş verisi lineer bir şekilde ayırt edilebiliyorsa aşağıdaki aşırı düzlem denklemi kullanılır.

$$f(X) = w^T x + b \quad (4.1)$$

Yukarıdaki denklemde w parametresi n boyutlu vektör ve b ise skaler çarpan veya sapma değeridir. Bu denklem pozitif ve negatif sınıfı ayırmak için maksimum ayırma düzlemini bulur. İki sınıfı ayırt etmek için denklem 4.2 kullanılır. Şekil 4.1'de ise doğrusal bir şekilde ayırt edilebilen veriler için ayırma düzlemi örnekleri verilmiştir.

$$y_i (w \cdot x_i + b) \geq 1 \quad i = 1 \dots k \quad (4.2)$$



(a) Maksimum marjin ile ayırma düzlemi
Şekil 4.1. Ayırma düzlemi ve destek vektörler

(b) Ayırma düzlemi örneği

Test verileri denklem 4.2'ye göre değerlendirilir. Denklem 4.2'deki şartı sağlayan bir test örneğinin doğru bir şekilde sınıflandırıldığı söylenebilir. Şekil 4.1'de kare ve halkalar iki sınıf verisini gösterir. Destek vektör makinasının test aşmasında gelen verilerin sınıfını belirlemek için yıldız ile gösterilen destek vektörler kullanılmaktadır. Şekil 4.1 (b)'de gösterilen düzlem göz önüne alındığında destek vektörler düzleme daha yakındır. Şekil 4.1 (a)'da verilen aşırı düzlem eğitim aşamasında kullanılmayan bir veri geldiğinde doğru bir şekilde sınıflandırma yapabilir. İki sınıfın destek vektörleri arasındaki marjin kısa olursa yanlış sınıflandırma ihtimali daha yüksek olur. Eğer veriler lineer bir şekilde ayırt edilemiyorsa, bazı serbestlik değişkenleri tanımlanır. Dolayısıyla maksimum marjine sahip bir sınıflandırıcı oluşturmak için denklem 4.3'teki şartları sağlayacak ikinci dereceden bir konveks programlama probleminin çözülmesi gerekir.

$$\left\{ \begin{array}{ll} \text{Minimum} & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^k \xi_i \\ \text{Amaç} & y_i (w_i x_i + b) \geq 1 - \xi_i \end{array} \right\} \quad (4.3)$$

Serbestlik değişkenleri ξ_i pozitif tanımlanmış değişkenlerdir. Bu değişkenler karar sınırının yanlış tarafında olan veya sınıflandırılmayan verilere izin verir. Marjinin genişliği ceza parametresi ile kontrol edilebilir. Bu parametre kullanıcı tarafından seçilen ağırlık parametresidir. Sınıflandırmanın performansını doğrudan etkileyen bu parametre, Lagrange çarpanlarının sınırını belirler. Optimizasyon problemi olarak ele alınan Lagrange çarpanları denklem 4.4'te verilmiştir.

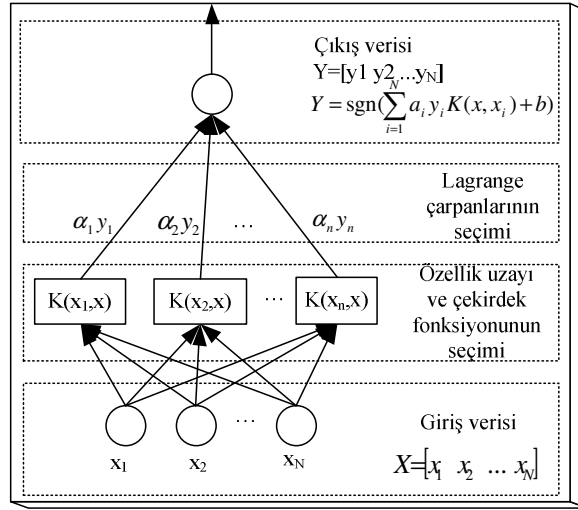
$$\left\{ \begin{array}{l} \text{Maksimum } L(\alpha) = \sum_{i=1}^k \alpha_i - \frac{1}{2} \sum_{i,j=0}^k \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{Amaç} \quad \sum_{i=1}^k y_i \alpha_i = 0 \\ \alpha_i \geq 0 \text{ for } i = 1 \dots k \end{array} \right\} \quad (4.4)$$

Denklemden α_i Lagrange çarpanlarıdır. Optimum aşırı düzlem genelleştirme yeteneğini maksimum yapmak için belirlenir. Bu optimizasyon problemi, ardışık minimum yapma algoritması ile çözülür [128]. Bu algoritma ikinci dereceden bir programlama problemini sadece iki Lagrange çarpanı içeren alt problemlere böler. Eğer problem lineer uzayda çözülemiyorsa, orijinal giriş uzayı özellik uzayı olarak isimlendirilen daha yüksek boyutlu bir uzaya dönüştürülür. Bu uzay karmaşık nokta çarpımları içerir. Çekirdek fonksiyonu yüksek boyutlu bu uzayda optimum aşırı düzlemi bulmak için kullanılır. Birçok çekirdek fonksiyonu bulunmakta olup, en çok kullanılanları Tablo 4.1’de verilmiştir.

Tablo 4.1 Popüler çekirdek fonksiyonları

Çekirdek fonksiyonu türü	Matematiksel gösterimi
Lineer	$K(x, x_i) = (x^T x_i)$
Polinom	$K(x, x_i) = [(x^T x_i) + 1]^d$
Radyal tabanlı	$K(x, x_i) = e^{-\ x - x_i\ ^2 / (2\sigma^2)}$
Sigmoid	$K(x, x_i) = \tanh((x^T x_i) + b)$

Tablo 4.1’de σ ve b her bir çekirdek fonksiyonunun çekirdek parametreleridir ve bu parametreler destek vektör makinasının performansını etkilerler. Destek vektör makinasının eğitim aşaması Şekil 4.2’de verilmiştir.

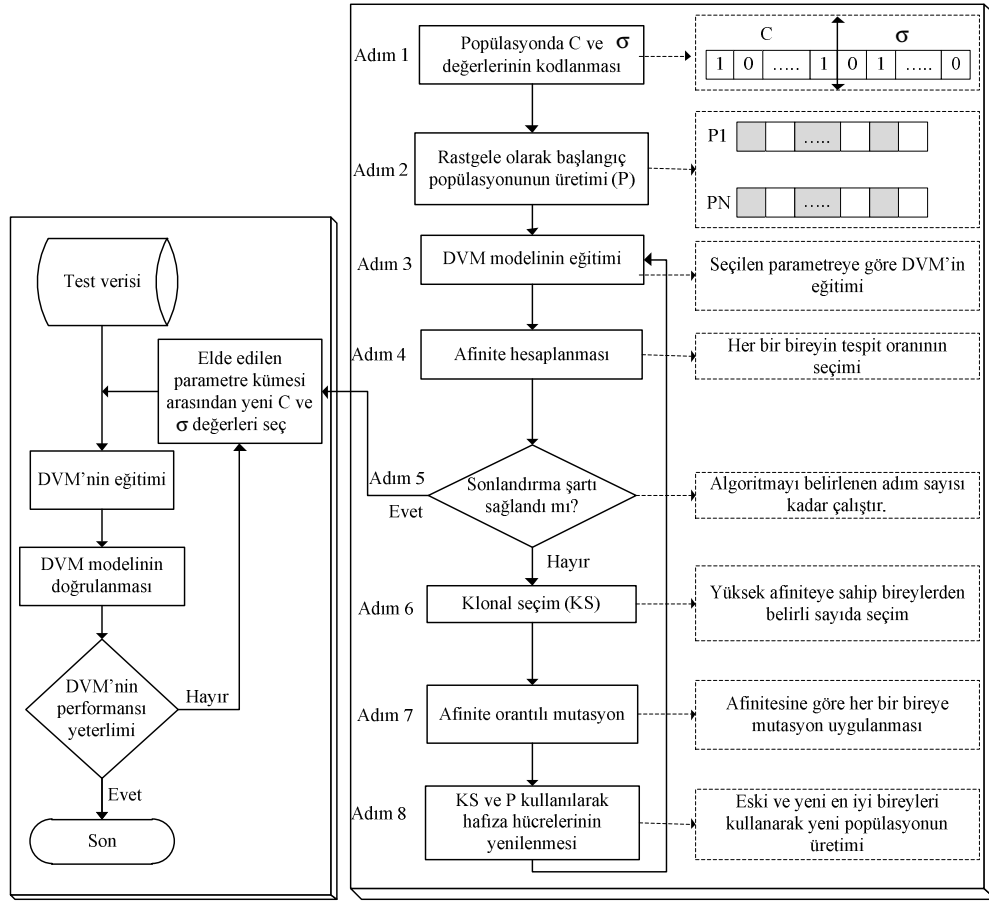


Şekil 4.2. Destek vektör makinaların eğitim aşaması

Destek vektör makinaların eğitiminde, ilk olarak çekirdek matrisi giriş verisi kullanılarak oluşturulur ve Lagrange çarpanları ardışık minimum optimizasyon algoritması ile bulunur. Son olarak elde edilen bu sıfırdan farklı Lagrange çarpanları algoritmayı test etmek için kullanılır.

4.1.2. Destek Vektör Makinaların Parametrelerinin Bağımsızlık Tabanlı Seçimi

Destek vektör makinaların performansı çekirdek parametrelerinin seçimi, büyüklüğü ve ceza terimine bağlıdır. Farklı çekirdek fonksiyonları farklı tipteki uygulama verileri için farklı başarımlar elde eder. Ceza parametresi C'nin değeri çok büyük veya çok küçük seçildiğinde destek vektör makinaların genelleştirme yeteneği azaltılabilir. Eğer çekirdek parametreleri ve ceza parametresi uygun seçilirse, destek vektör makinaların performansı maksimum yapılmış olur. Bu amaçla destek vektör makinalarının en uygun parametrelerinin seçimi için klonal seçim tabanlı bir algoritma önerilmiştir. Burada yapılan optimizasyonun amacı destek vektör makinaların performansını arttırmaktır. Önerilen yöntem optimizasyon için iki uygunluğu göz önüne alır. Biri destek vektör makinaların doğru sınıflandırma başarımları ve diğeri ise destek vektörlerin sayısıdır. Destek vektör makinaların parametreleri klonal seçim algoritması uygulanarak dinamik bir şekilde optimize edilmekte ve bu parametreler test aşamasında kullanılarak sınıflandırma başarımları maksimum yapılmaktadır. Önerilen algoritma Şekil 4.3'te verilmiştir.

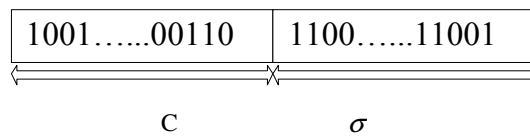


Şekil 4.3. Önerilen algoritmanın akış şeması

Çekirdek fonksiyonunun seçimi destek vektör makinasının eğitiminde önemli bir adımdır. Bu fonksiyon DVM'nin performansını etkiler. Bu çalışmada, uygulamalarımız için radyal tabanlı çekirdek fonksiyonu kullanılmıştır. Şekil 4.3'teki algoritmanın adımları detaylı olarak aşağıda verilmiştir.

Adım 1

Her antikor ikili bir string ile kodlanır. Antikorların belirli sayısı C için $[1, 255]$ arasında ve σ için $[0.1, 2.55]$ arasında seçilir. Her antikor C ve σ 'den oluşur. Antikoru kodlanması Şekil 4.4'te verilmiştir.



Şekil 4.4. Antikorların ikili kodlanması

Adım 2

Her bir antikoru başlangıç popülasyonu rastgele üretilir. Popülasyon boyutu 20 olarak seçilmiştir. Çekirdek parametresi ve ceza terimi her bir antikorda bulunmalıdır.

Adım 3

Antikorların başlangıç popülasyonu üretildikten sonra her bir antikor için destek vektör makina ardışık minimum optimizasyon algoritması ile eğitilir. Bu adımda destek vektör makinanın performansı her bir antikor için radyal tabanlı çekirdek fonksiyonu ile test edilir.

Adım 4

Bir ak antikoru amaç fonksiyonun değeri $f(ak)$ ile ifade elde edilir. Bu ağırlıklı amaç fonksiyonu denklem 4.5 'te verilmiştir.

$$\left\{ \begin{array}{l} f_1(ak) = (100 - err) \\ f_2(ak) = 100 * (\frac{dvs}{N}) \\ f(ak) = \alpha.f_1(ak) - \beta.f_2(ak) \quad \alpha > \beta \quad \text{için} \end{array} \right\} \quad (4.5)$$

Bu denklemde err değişkeni, yanlış sınıflandırılmış verilerin sayısını, dvs destek vektörlerinin sayısını ve N ise eğitim verisi sayısını gösterir. Eğer bir antikor ile antijen arasındaki afinite yüksek ise, bu antikor sonraki adımda seçilebilir ve bu antikor optimum çözüm olabilir.

Adım 5

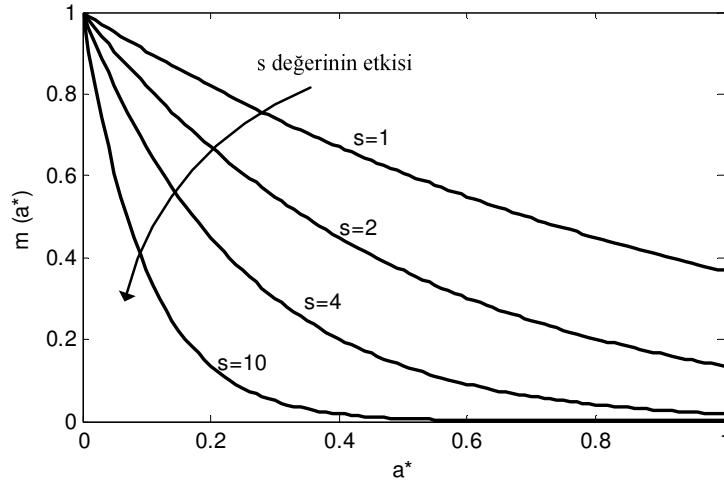
Algoritma belirli adım sayısı kadar çalıştırılır. Eğer sonlandırma şartı sağlanırsa, algoritma durdurulur ve optimum çözüm bulunmuştur.

Adım 6

Bu adımda en yüksek afiniteye sahip n_1 eleman seçilir ve antijen ile afinite oranlarına göre bu bireylerin kopyaları üretilir. Daha yüksek afinite daha yüksek kopyalar ile sonuçlanır.

Adım 7

Bir bireyin mutasyon oranı kendi afinitesi ile ters orantılıdır. Bu çalışmada mutasyon oranı ve normalleştirilmiş afinite arasındaki ilişkiyi kurmak için üstel fonksiyon kullanılmıştır. Bu yaklaşım Şekil 4.5 ve denklem 4.6 'da verilmiştir.



Şekil 4.5. Mutasyon oranı ve ölçeklendirilmiş afinite arasındaki değişim

Bu şekilde s değerinin etkisi afinite orantılı mutasyon için önemli bir parametredir ve klonal seçimin performansını etkiler. Burada s çok küçük olduğunda, mutasyon hemen hemen her bireye uygulanır. Ters durumda çok az birey mutasyona uğrar ve popülasyonun çeşitliliği sağlanamaz.

$$m(a^*) = \exp(-sf^*) \quad (4.6)$$

Burada f^* normalleştirilmiş afinitedir ve $f^* = f / f_{maks}$ ile belirlenir. Bu çalışmada s parametresi 4 olarak seçilmiştir.

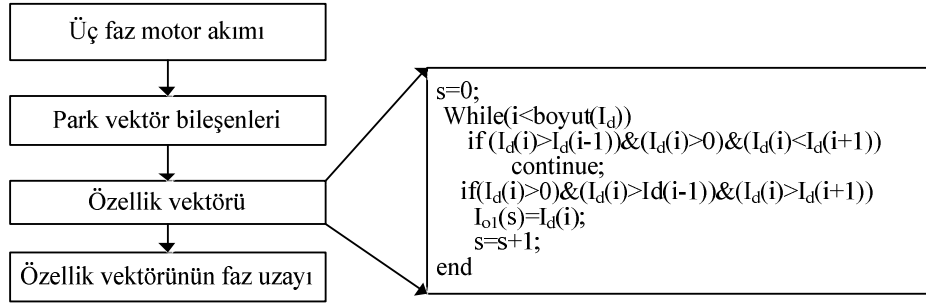
Adım 8

Popülasyonun çeşitliliğini sağlamak için yeni tespit edilen en iyi antikorlar eski en kötüler ile değiştirilir. Eğer en iyi birey üç adım boyunca değişmiyorsa, afinite orantılı mutasyon uygulanır. Yerel minimuma yakalanmamak için, her on adımda bir antikorların yenilenir. Eğer sınıflandırmanın hata oranı istenilen seviyenin altına düşmemişse 3 nolu adıma gönderilir.

4.1.3. Deneysel Sonuçlar

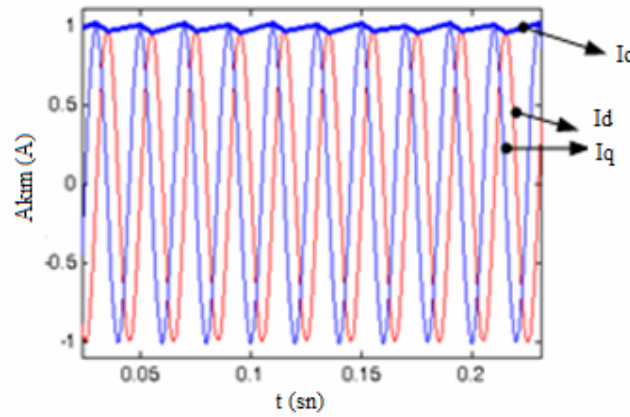
Önerilen yöntem üç fazlı bir asenkron motorda oluşan kırık rotor çubuğu arızalarının tespiti için kullanılmıştır. Bu amaçla bir, iki ve üç kırık rotor çubuğu arızası dört farklı çalışma hızı altında tespit edilmektedir. Önerilen yöntem arıza tespiti için sadece üç faz akımına ihtiyaç duyar. Kırık rotor çubuğu arızaları ile ilgili özellikler üç faz akımından

elde edilmiştir. Üç faz motor akımından park vektör bileşenleri hesaplanır. Yeni özellik vektörü bu park vektör bileşenleri kullanılarak hesaplanmaktadır. Bu iki park vektör bileşeni kullanılarak yeni bir özellik vektörü elde edilmiştir. Önerilen ön işleme adımı Şekil 4.6’da gösterilmektedir.



Şekil 4.6. Veri ön işleme adımı

Şekil 4.6’daki algoritma ile sadece I_d park vektör bileşeninin tepe noktasına yakın bölgelerdeki değişimlerden oluşan I_{o1} vektörü elde edilmiştir. Bu işlem, I_q vektörü için de tekrarlanarak I_{o2} elde edilir ve iki vektör birleştirilerek I_o vektörü elde edilir. Park vektör bileşenleri üç faz akımından elde edilir. Burada bu bileşenler I_d ve I_q olarak isimlendirilmiştir. İki park vektör bileşeni kullanılarak elde edilen özellik vektörü Şekil 4.7’de verilmiştir.



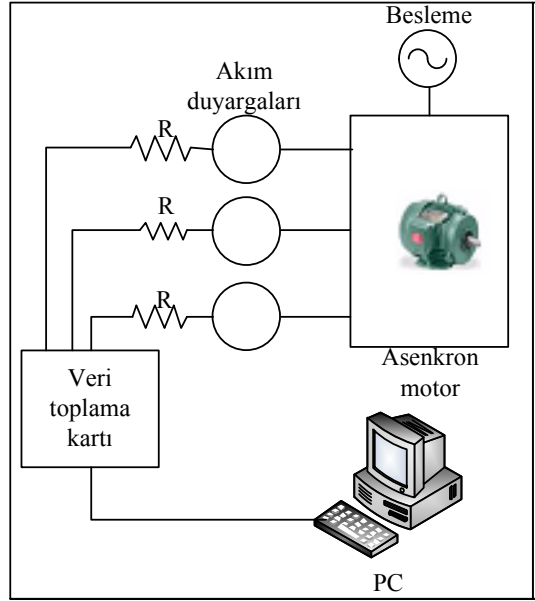
Şekil 4.7. Özellik vektörü

Özellik vektörü elde edildikten sonra, bu vektörün faz uzayı oluşturulur. Önerilen algoritmayı test etmek için üç fazlı gerçek bir asenkron motordan alınan motor akımları

kullanılmıştır. Akım sinyallerinin alınması için oluşturulan deney düzeneğinin şeması Şekil 4.8’de verilmiştir. Tablo 4.2’de deneyde kullanılan asenkron motorun özellikleri verilmektedir.

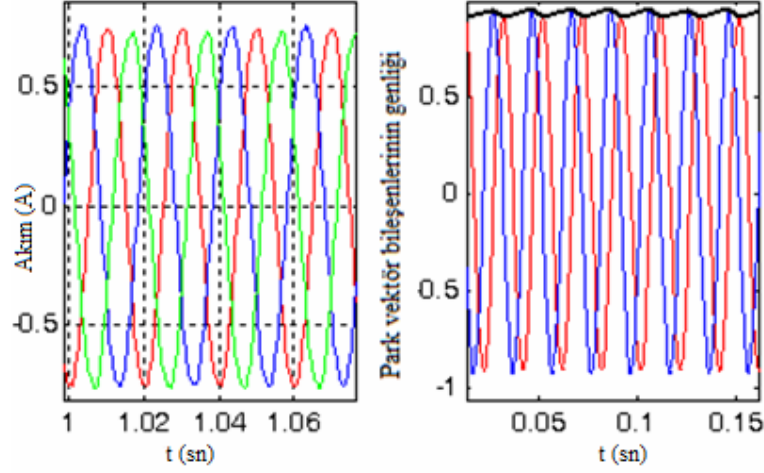
Tablo 4.2. Deneyde kullanılan asenkron motorun özellikleri

Özellik	Değer
Güç	0.37 kW
Giriş gerilimi	380 V
Tam yük akımı	1.2 A
Besleme frekansı	50 Hz
Kutup sayısı	4
Rotor çubuklarının sayısı	22
Tam yük hızı	1390 rpm



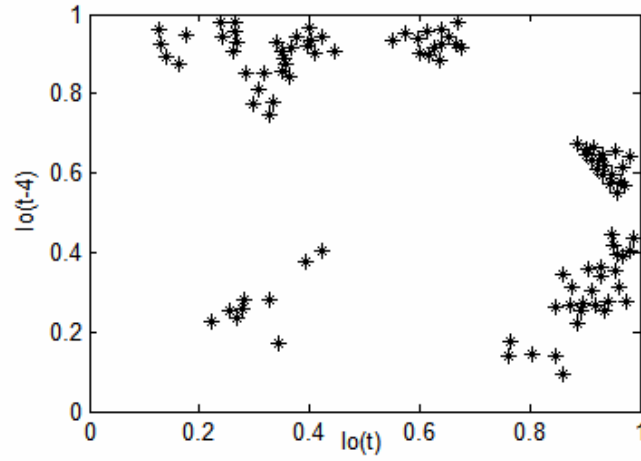
Şekil 4.8. Deney düzeneği şeması

Motor sağlam, bir, iki ve üç kırık rotor çubuğu arızası verileri ile test edilmiştir. Deneysel motor faz akımları 3 kHz ile örneklenip 3 sn boyunca kaydedilmektedir. Her bir durum için besleme gerilimleri 380, 340, 300 ve 260 V ‘tur. 300 V ile beslenen bir kırık rotor çubuğu arızası için üç faz akımı, park vektör bileşenleri ve özellik vektörleri Şekil 4.9’da verilmektedir.



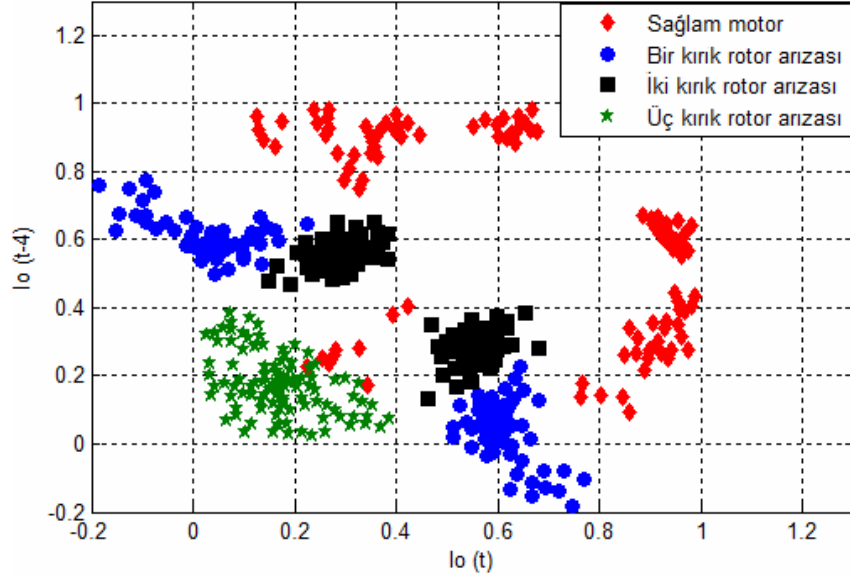
Şekil 4.9. Faz akımları, park vektör bileşenleri ve özellik vektörü

Özellik vektörü elde edildikten sonra her bir durum için faz uzayı oluşturulur. Faz uzayı için gömülme boyutu b yanlış en yakın komşu algoritmasına göre iki, zaman gecikmesi τ ise karşılıklı bilgi yöntemine göre 4 olarak elde edilmiştir. Sağlam motor için faz uzayı Şekil 4.10'da verilmiştir.



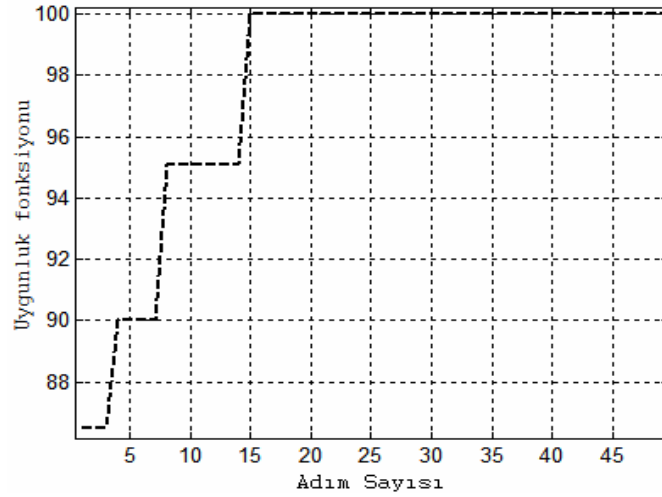
Şekil 4.10. Sağlam motor faz uzayı

Her bir motor durumu için faz uzayları elde edildikten sonra, bu faz uzayları Şekil 4.11'deki gibi birleştirilmektedir.



Şekil 4.11. Birleştirilmiş faz uzayı

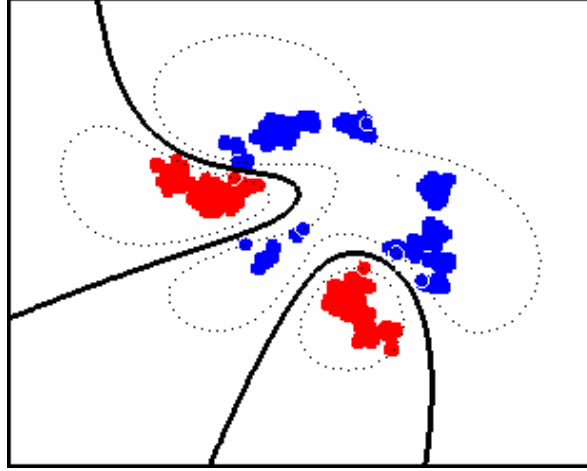
Yapay bağışık tabanlı destek vektör makine sınıflandırma algoritması bu faz uzayında uygulanır. Sağlam ve bir kırık rotor çubuğu arızalarını ayırt eden klonal seçim algoritmasının performansı Şekil 4.12’de verilmiştir.



Şekil 4.12. Klonal seçim algoritmasının performansı

Şekil 4.12’den de görüldüğü gibi klonal seçim algoritması birkaç adımda optimum çözümü bulmaktadır. Destek vektör makinaları normalde iki sınıflı sınıflandırma yöntemidir. Bu çalışmada birine karşı diğeri yöntemi kullanılarak çok sınıflı sınıflandırma

işlemi yapılmıştır. Bir kırık rotor çubuğu ve sağlam motorun faz alanlarını ayırt eden destek vektör makinasının sonucu Şekil 4.13'te grafiksel olarak verilmiştir.



Şekil 4.13. Optimum çekirdek ve ceza parametresi ile DVM'nin sonucu

Destek vektör makinasının eğitimi için yapay bağışık sistem ile elde edilen optimum çekirdek ve ceza parametreleri Tablo 4.3'te verilmiştir.

Tablo 4.3. Arıza tespiti için yapay bağışık sistem tabanlı DVM algoritmasının sonuçları

Sınıf	DVS	σ	C	Doğruluk oranı (%)
S1KR	11	0.27	213	99.00
S2KR	12	0.17	159	99.00
S3KR	15	1.38	140	97.50
1KR2KR	9	0.38	242	99.00
1KR3KR	4	0.63	88	99.00
2KR3KR	5	0.45	244	99.00
Ortalama doğruluk oranı				98.85

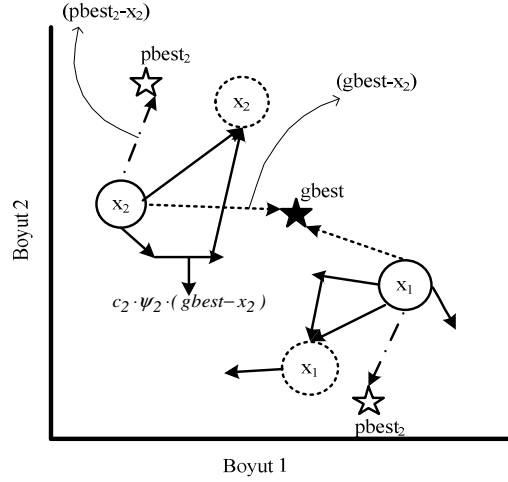
Bu tabloda S1KR sağlam motor faz alanına göre bir kırık rotor çubuğu arızasının sınıflandırıldığını ve DVS değeri ise elde edilen destek vektör sayısını gösterir. Rotor arızalarının tespiti için en çok kullanılan yöntem motor akım imza analizidir. Bu yöntem kırık rotor çubuğu arızalarını tespit etmek için motor kaymasına ihtiyaç duyar. Kayma değeri de motor hızının bilinmesini gerektirir. Bu da ikinci bir duyarga ile hız ölçümü anlamına gelir. Burada önerilen yöntem sadece üç faz motor akımına ihtiyaç duyar.

4.2. Sürü Öğrenme Tabanlı Yapay Bağışık Arıza Teşhis Yöntemi

İnsanın doğal bağışıklık sisteminden esinlenilerek geliştirilen yapay bağışık sistemler hesapsal özelliklere sahip olup sınıflandırma, örüntü tanıma ve anomali tespiti gibi problemlere kolaylıkla uygulanabilmektedir. Bununla birlikte yapay bağışık sistemler temel olarak üç probleme sahiptir. Birincisi popülasyon boyutunu kontrol eden herhangi bir mekanizma olmadığından eğitim aşamasında popülasyon aşırı derecede büyüyerek algoritmanın yavaşlamasına neden olur. İkinci problem ise eğitim aşamasında her defasında bir eğitim örneği alınarak afinite hesaplandığından daha sonraki örnekler için iyi afiniteye sahip hafıza hücreleri elenmektedir. Son problem ise yapay bağışık sistemlerde klonlama ve mutasyon işlemleri rastgele bir şekilde uygulanmakta ve bu süreç eğitim aşamasını yavaşlatmaktadır. Bu problemleri çözmek amacıyla sürü öğrenme tabanlı yeni bir yapay bağışık sınıflandırıcı önerilmektedir. Önerilen yöntem bağışık sistemdeki hafıza hücrelerinin gelişimi için sürü öğrenmesini kullanır. Hafıza hücreleri popülasyonunun boyutu algoritmanın eğitim aşamasında değişmeyip sabit kalmaktadır. Böylece önerilen yöntem diğer bağışık sınıflandırıcılardan daha hızlı çalışmaktadır. Ayrıca mutasyon işleminde rastgelelik yerine belli bir hedefe yönelme sürü optimizasyonu ile gerçekleştirildiğinden algoritmanın performansı arttırılmıştır. Önerilen sınıflandırıcının performansı arıza teşhisi için iki ayrı veri kümesi üzerinde uygulanarak sonuçlar yapay sinir ağları ve DVM gibi diğer doğrusal olmayan sınıflandırıcılar ile karşılaştırılmıştır.

4.3.1. Parçacık Sürü Optimizasyonunun Prensipleri

Parçacık sürü optimizasyonu (PSO) ilk olarak 1995 yılında Kennedy ve Eberhart tarafından önerilen popülasyon tabanlı bir optimizasyon yöntemidir [129]. Yöntem adını kuş ve balık gibi sürülerin hareketinden almaktadır. PSO’da popülasyon sürü olarak, sürüdeki her bir birey ise parçacık olarak isimlendirilir. Her parçacık, iki parametreye sahiptir: hız ve pozisyon. Optimizasyon probleminin çözümü parçacık ile ifade edilir. Parçacıkların yeni pozisyonu, parçacığın kendisinin şu ana kadarki en iyi pozisyonu (pbest) ve popülasyonun en iyi pozisyonu (gbest) ile belirlenmektedir. Eğer bir parçacık daha iyi bir pozisyon keşfederse, diğer bütün parçacıklar ona yaklaşacaktır. PSO algoritmasında iki parçacığın yerel en iyi ve global en iyi pozisyonlara göre gelişimi Şekil 4.14’te iki boyutlu olarak gösterilmiştir.



Şekil 4.14. İki parçacığın pbest ve gbest'e göre iki boyutlu gelişimi

Şekil 4.15'te PSO algoritmasının temel adımları görülmektedir. Algoritmanın ilk adımında her bir parçacığın pozisyonu ve hızı aşağıdaki adımlara göre başlatılmalıdır:

- Her bir $x_{i,j}$ parçacığının pozisyonunu $i \in 1, \dots, n$ ve $j \in 1, \dots, m$ için $[-x_{maks}, x_{maks}]$ aralığında rastgele bir şekilde başlat.
- Her bir $v_{i,j}$ parçacığının hızını $i \in 1, \dots, n$ ve $j \in 1, \dots, m$ için $[-v_{maks}, v_{maks}]$ aralığında rastgele bir şekilde başlat.
- Her bir parçacığın yerel en iyi hızını belirle.

$x_{i,j}$ ve $v_{i,j}$ 'yi rastgele olarak üret.

Do

For i=1:parçacık sayısı

Parçacıkların uygunluklarını hesapla (p_i)

If($p_{i+1} > p_i$) $p_i = p_{i+1}$

$p_g = \max(x_{pbest})$ // Global en iyiyi bul

For j=1:boyut sayısı

$$v_{i,j}(t+1) = \chi(v_{i,j}(t) + c_1 \cdot \psi_1 \cdot (p_i(t) - x_{i,j}(t)) + c_2 \cdot \psi_2 \cdot (p_g(t) - x_{i,j}(t)))$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1)$$

End j

End i

Until Sonlandırma Kriteri

Şekil 4.15. Orijinal PSO algoritmasının temel adımları

Şekil 4.15'deki sonlandırma kriteri problemden probleme değişir. Bazı problemlerde sabit adımlı bir sonlandırma seçilirken bazılarında ise en iyi çözüme ne kadar yaklaşıldığı bir önceki çözüm ile aradaki farka bakılarak belirlenir. PSO'nun iki denklemindeki değişkenlerin anlamları aşağıda verilmiştir.

$v_{i,j}(t)$	t adımında i. parçacığın j. boyutunun hızı
$x_{i,j}(t)$	t adımında i. parçacığın j. boyutunun pozisyonu
c_1, c_2	sabit ağırlık faktörleri
p_i	i. parçacığın şu ana kadarki en iyi pozisyonu
p_g	i. parçacığın komşuları tarafından bulunan en iyi pozisyon
ψ_1, ψ_2	[0, 1] aralığında rastgele değerler
χ	kısıtlama faktörü

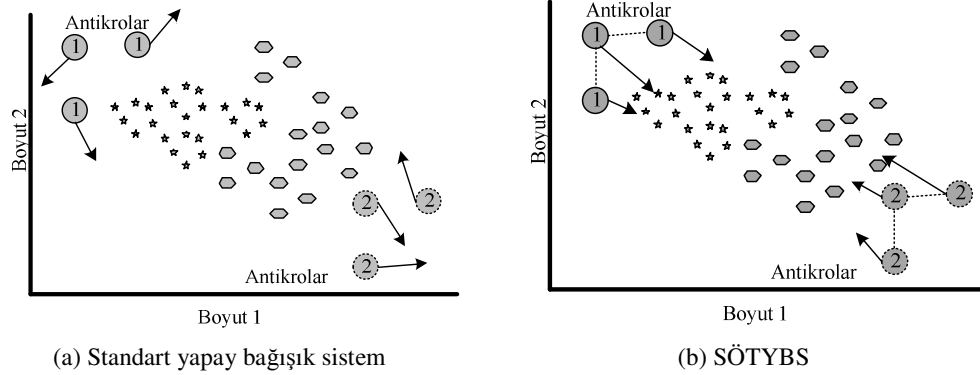
Her bir parçacığın komşuluğu bütün sürü veya sürünün bir alt kümesi olarak alınabilir. Hızın her bir boyutu $[-v_{maks}, v_{maks}]$ arasında sınırlandırılarak parçacıkların problem uzayından ayrılmaları engellenir.

PSO'da bütün parçacıkların hız ve pozisyonları anlık olarak güncellenir. Üstelik diğer optimizasyon yöntemlerinden daha az sistem bilgisi gerektirir. Standart PSO'da bütün parçacıklar optimizasyon süreci boyunca diğerleri ile çarpışabilir. Dolayısıyla standart PSO dinamik olarak değişen optimizasyon problemlerine uygulanamaz. Bu problemin üstesinden gelmek için değiştirilmiş bir PSO olarak anti-çarpışma yöntemi geliştirilmiştir [130]. Bu algoritmanın avantajı; parçacıkların birbirlerini itmesi, çarpışmanın engellenmesi ve en iyi çözüme yaklaşma arasında ağırlık merkezine göre bir denge olmasıdır. PSO tabanlı öğrenme yöntemimizin temel prensibi anti-çarpışma yöntemine dayanmaktadır.

4.2.2. Sürü Öğrenme Tabanlı Yapay Bağışık Sınıflandırıcı (SÖTYBS)

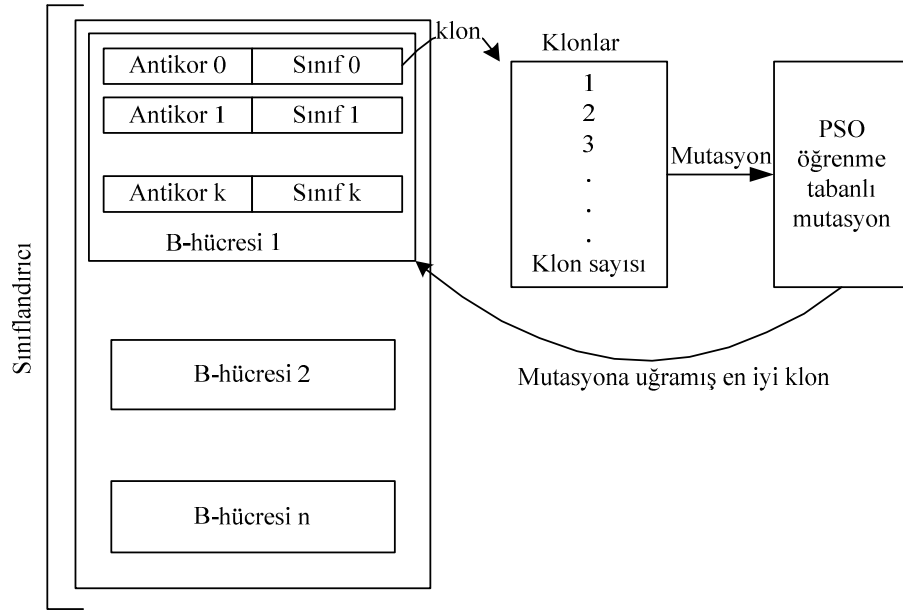
Önerilen SÖTYBS yönteminin temel prensibi, biyolojik bağışıklık sistemimizdeki B hücrelerinin klonal seçimi ve parçacık sürü öğrenmesine dayalıdır. Eğitim aşamasında B-hücreleri klonal seçim, afinite gelişimi ve mutasyon işlemlerine uğramaktadır. Eğitim ve test aşamasında verilen özellik sinyalleri antijen olarak alınmaktadır. Her antikor B hücrelerinin bir parçasıdır. Bir antijen herhangi bir antikora sunulduğunda antikorlar klonal genişlemeye maruz kalır. Antikorlar klonlandıktan sonra antijenler ile afinitelerine göre

ters orantılı olarak mutasyona uğrarlar. Önerilen SÖTYBS yönteminde standart bağışık sistem tabanlı sınıflandırıcılara temel olarak iki geliştirme yapılmıştır. Standart bağışık sistemde her adımda bir antijen antikorlara verilerek antikorların gelişimi sağlanmaktadır. SÖTYBS’de ise bütün antijenler antikorlara aynı anda verilmektedir. Dolayısıyla afinitesi iyi olan antikorların bir sonraki aşamada elenmesi engellenmektedir. Ayrıca standart yapay bağışık sistemlerde mutasyon rastgele uygulanmaktadır. Bu çalışmada ise mutasyon işlemi PSO tabanlı öğrenmeye dayalıdır. Bir klonun mutasyona uğramış bireyi dört kritere göre üretilmektedir: klonun afinitesi, klonun ebeveyninin şu ana kadarki en iyi afinitesi, aynı sınıftaki en yakın antikor ve farklı sınıftaki en yakın antikor. Bu özellikler ile her klon gideceği hedefi bilir ve belirlenen yörüngeyi izler. Bir antikorun en iyi klonları B-hücresi olarak seçilir. Şekil 4.16’da standart yapay bağışık sistemler ile önerilen SÖTYBS algoritmasında antikorların hareketi gösterilmiştir.



Şekil 4.16. Standart yapay bağışık sistem ve SÖTYBS’de antikorların hareketi

Şekil 4.16 (a)’da gösterildiği gibi standart yapay bağışık sistemde antikorların hareketini sağlayan mutasyon işlemi rastgele bir şekilde yapılmaktadır. Dolayısıyla her bir eğitim adımında antikorların ait oldukları sınıfa ilerlemeleri garanti edilemez. Şekil 4.16 (b)’de antikorların hareketi parçacık sürü optimizasyonu ile gerçekleştirilmektedir. Her antikorun sınıf etiketi antikorun üzerine yazılmıştır. Birbirine komşu olan antikorların çarpışmalarını engellemek için bir itme kuvveti vardır. Antikorlar klonlandıktan sonra antijenik afiniteleri ile ters orantılı olarak mutasyona uğramaktadırlar. Fakat mutasyon işlemi rastgele bir tarzda yapılmamaktadır. Bu süreç parçacık sürü optimizasyonuna dayalıdır. SÖTYBS metodunun diyagramı Şekil 4.17’de verilmiştir.



Şekil 4.17. SÖTYBS diyagramı

Şekil 4.17’de her antikor bir sınıf etiketine sahiptir. Bu sınıf etiketi algoritmanın gelişimi boyunca değişmez. K-sınıflı bir problem için her sınıf n örnek ile gösterilir. En iyi afiniteye sahip mutasyona uğramış klon kendi ebeveyn antikoru ile karşılaştırılır. Eğer mutasyona uğramış antikoru afinitesi ebeveyninkinden daha iyi ise, onun yerini alır. Her B-hücresi heterojen olup farklı sınıfların antikorlarını tutar. Aşağıda sınıflandırma probleminde antikor ve antijenlerin gösterimi verilmektedir.

- Her bir antikor ak ile gösterilmektedir. Antikorların sınıfı $ak.c$ ve özellik vektörü ise $ak.f$ ile ifade edilir. Özellik vektörünün i . bileşeni $ak.f_i$ ile gösterilir. Antikorlar için kullanılan $ak.b$ iki değer alır. Eğer ak gerçek sınıfına ait ise “1” değerini aksi halde “0” değerini alır. Ayrıca bu parametre birbirleri ile çarpışmayan antikorların üretilmesini de sağlar. Klonlama ve mutasyon işleminden geçen daha iyi antikorlar yeni sınıflandırıcılar olur. Bu yüzden benzer antikorların elenmesi için herhangi bir mekanizmaya ihtiyaç yoktur.
- Her bir antijen an ile gösterilmektedir. Eğitim ve test veri kümeleri antijen olarak alınır. Antijenlerin sınıf etiketi $an.c$ ve özellik vektörü ise $an.f$ ile gösterilmektedir.

Eğitim ve test aşamasında kullanılan veri kümesinin her bir özelliği $[0, 1]$ aralığına çekilmiştir. Örneğin iki sınıflı ve üç özelliğe sahip bir örnek için sınıf $an.c=2$ ve özellik $an.f=(0.1, 0.5, 0.6)$ ile ifade edilir. SÖTYBS yönteminin adımları aşağıda verilmiştir:

Adım 1: Eğitim veri kümesinin yüklenmesi

Bu aşamada biçim uzayındaki antijenlerin kümesi eğitim veri kümesi olarak belirlenir. Her antijenin bir sınıf etiketi ve özellik vektörü bulunur.

Adım 2: Başlangıç popülasyonunu oluşturan antikörlerin üretilmesi.

Boyutları $n \times k$ olacak şekilde antikörler üretilir ve üretilen antikörler B-hücreleri veya sınıflandırıcı olarak alınır.

Adım 3: Maksimum adım sayına kadar

Adım 3.1: Her bir sınıf için hangi antikörün kendi sınıfında hangisinin yanlış sınıfta olduğunu bul. Dolayısıyla her bir antikör için $ak.b$ değeri bulunur.

Sınıf etiketi i olan bir ak antikörü için ($ak.c=i$) eğer ak i sınıfından en az bir antijen sınıflandırıyorsa $ak.b$ “1” değerini alır. Bu parametre eğer ak j sınıfından ($i \neq j$) en az bir antijen sınıflandırıyorsa sıfır değerini alır.

Adım 3.1.1: Her bir antikör için

Adım 3.1.1.1: Antikörlerin afinitesini hesapla

Afinite özellik uzayında Öklid mesafesi ile ters orantılıdır. Bir antikörün afinitesi iki faktör göz önüne alınarak hesaplanır. Birincisi geçerli antikörün kendi sınıfındaki antijenleri sınıflandırma oranı, ikinci faktör ise geçerli antikör ile farklı sınıftaki antijenlerin sınıflandırma oranıdır. Eğer bir antijen geçerli antikör ile aynı sınıfta ve geçerli antikörün tanıma alanında ise antijen doğru sınıflandırılmıştır. Antijenin yanlış sınıflandırılması ise kapsama alanında bulunduğu antikörün farklı sınıfta olmasıdır. Her antikör için bir yarıçap (σ) tanımlanmıştır. Antijen ile antikör arasındaki afinite aşağıdaki şekilde hesaplanmaktadır.

$$ak.af = \begin{cases} \frac{F_1 - F_2}{F_1 + F_2} + 1.0, & \text{If } F_1 + F_2 \neq 0 \\ -2.0, & \text{If } F_1 + F_2 = 0 \end{cases} \quad (4.7)$$

Denklem 4.7’de F_1 ve F_2 değerleri ak antikörü tarafından sırasıyla doğru ve yanlış sınıflandırılan antijen sayısını gösterir. Her bir antikörün afinite değeri ($ak.af$) $[-2, 2]$ aralığında değer alır. Bir antikörün afinitesinin yüksek olması ancak aynı sınıftaki antijenleri tanıması ile gerçekleşir.

Adım 3.1.1.2: Geçerli antikör ile aynı sınıftaki en yakın antikörü bul.

Adım 3.1.1.3: Geçerli antikör ile farklı sınıfta olan en yakın antikörü bul.

Adım 3.1.1.4: Klon sayısına kadar (klonsayısı)

Adım 3.1.1.4.1: Geçerli antikör için klonlar üret

Adım 3.1.1.4.2: Klonları mutasyona uğrat

Orijinal yapay bağışık sistemde bir antikörün afinitesi ile mutasyon arasında ters bir bağıntı olup afinite ne kadar yüksek ise mutasyon oranı o kadar az olur. Fakat mutasyon her antikora rastgele uygulanmaktadır. PSO tabanlı mutasyon denklem 4.8, 4.9 ve 4.10'a göre uygulanmaktadır.

$$\begin{aligned} ak_c \cdot v_j^{yeni} = & ak_c \cdot f_j + \chi(w \cdot v_j^{eski} + c_1 \cdot \psi_1 \cdot (ak_{eniye} \cdot f_j - ak_c \cdot f_j) + \\ & c_2 \cdot \psi_2 \cdot \text{sign}(ak_{sc} \cdot f_j - ak_c \cdot f_j) \cdot ak_{eniye} \cdot m_oran + \\ & c_3 \cdot \psi_3 \cdot \text{sign}(ak_c \cdot f_j - ak_{dc} \cdot f_j) \cdot ak_{eniye} \cdot m_oran + c_4 \cdot \psi_4 \cdot m_oran) \end{aligned} \quad (4.8)$$

$$ak_c \cdot f_j^{yeni} = ak_j^{eski} \cdot f_j + ak_c \cdot v_j^{yeni} \quad (4.9)$$

$$m_oran = e^{-3f^*} \quad (4.10)$$

Denklem 4.10'da f^* normalleştirme faktörü olup $f^* = ak_c \cdot af / ak_{eniye} \cdot af$ arasında değer alır. Denklemlerde kullanılan diğer semboller aşağıda tanımlanmıştır.

v_j^{eski}	j. antikörün eski hızı
$ak_{eniye} \cdot f_j$	j. antikör tarafından şu ana kadar elde edilen en iyi pozisyon
$ak_c \cdot f_{jd}$	j. klonun d. özelliği
$ak_{sc} \cdot f_j$	Geçerli klon ile aynı sınıftaki en yakın antikör
$ak_{dc} \cdot f_j$	Geçerli antikör ile farklı sınıftaki en yakın antikör;
c_1, c_2, c_3	sabit ağırlık faktörleri
$\psi_1, \psi_2, \psi_3, \psi_4$	[0, 1] aralığında rastgele sayılar
sign ()	Etkinin yönünü belirten fonksiyon

Adım 3.1.1.5: Geçerli antikör ile mutasyona uğramış en iyi antikörün afinitesini karşılaştır ve eğer mutasyona uğramı birey daha iyi afiniteye sahip ise geçerli antikör ile değiştir.

Adım 3.2: B-hücrelerinin sınıflandırma performansını değerlendir.

Antijenleri sınıflandırmak için minimum mesafe sınıflandırıcı kullanılmıştır. Bu yöntem örnek tabanlı öğrenmeden uyarlanmaktadır. Eğitimde kullanılan antijen örnekleri en yakın antikörlere göre sınıflandırılır. Sınıflandırma performansı denklem 4.11'e göre hesaplanmaktadır.

$$Sınıflandırma_basarisi = \frac{dogru_s_say}{antijen_boyutu} \times 100 \quad (4.11)$$

Denklem 4.11’de herhangi bir antikor tarafından bir antijen doğru sınıflandırıldığında *dogru_s_say* değeri bir arttırılır. B-hücreleri havuzu sınıflandırıcıyı gösterip, bu hücrelerin antikorları klonlama ve sürü tabanlı öğrenmeye göre gelişir. Sınıflandırma performansı maksimum adım sayısına ulaşıncaya kadar değerlendirilir. Algoritmanın adım sayısı eldeki verilere göre belirli deneme yanımlar ile belirlenmiştir.

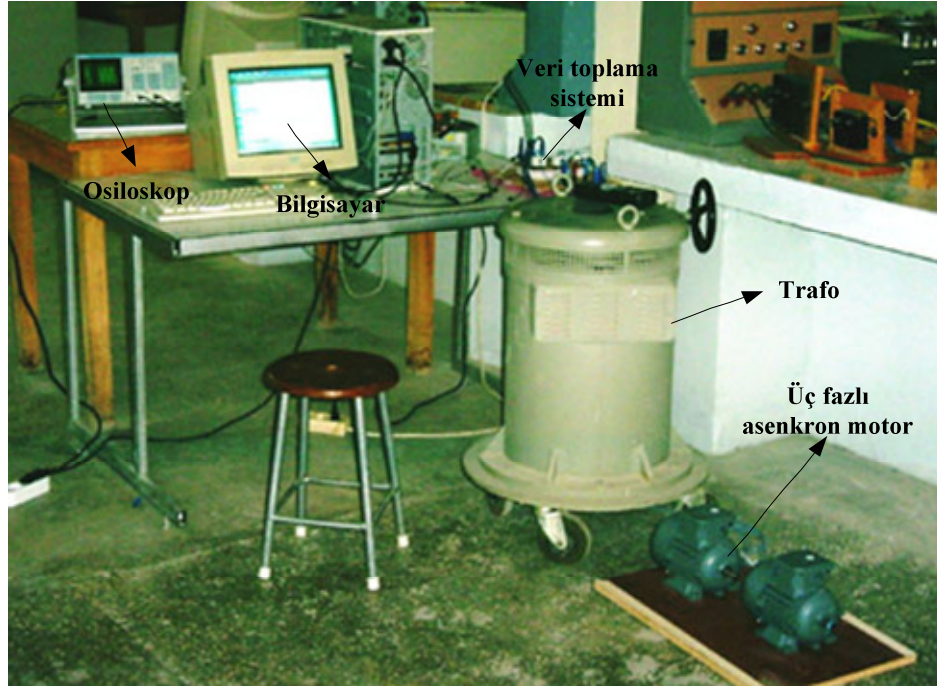
4.2.3. Deneysel Sonuçlar

Önerilen SÖTYBS’nin performansı arıza teşhisi ile ilgili iki veri kümesi üzerinde test edilmiştir. Sınıflandırma performansının testi için 10 noktalı çapraz doğrulama yönteminden faydalanılmıştır. PSO’nun parametreleri Tablo 4.4’te verilmiştir.

Tablo 4.4. PSO’nun parametreleri

Parametre	Değer
χ	0.5
w	0.1
c_1	1.0
c_2	1.0
c_3	0.25
c_4	0.25
σ	0.05

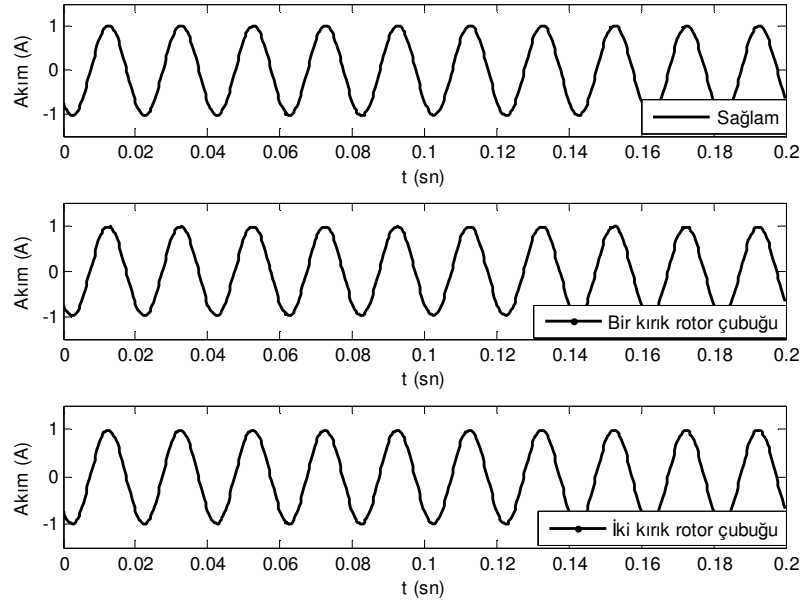
SÖTYBS yönteminin performansının test edilmesi için gerçek bir asenkron motordan deneysel olarak alınan sinyaller kullanılmıştır. Motor akım sinyalleri 3 saniye boyunca 3 kHz’lik bir örnekleme frekansı ile kaydedilmiştir. Testler sağlam bir motor ve bir ile iki kırık rotor çubuğu arızasına sahip motorlar üzerinde gerçekleştirilmiştir. Her iki durumda da motor akım sinyalleri yüksüz durumda alınmıştır. Düşük hızlarda önerilen metodun performansını değerlendirmek için beş farklı besleme gerilimi altında sağlam ve arızalı durumlar değerlendirilmiştir. Her bir durum için besleme gerilimleri 380, 340, 300, 260 ve 220 V olarak seçilmiştir. Ölçülen bu akım değerleri 16 bitlik Advantech PCL-816 model bir veri toplama kartı aracılığıyla bilgisayara aktarılmıştır. Kurulan deney düzeneği Şekil 4.18’de verilmiştir.



Şekil 4.18. Deney düzeneği

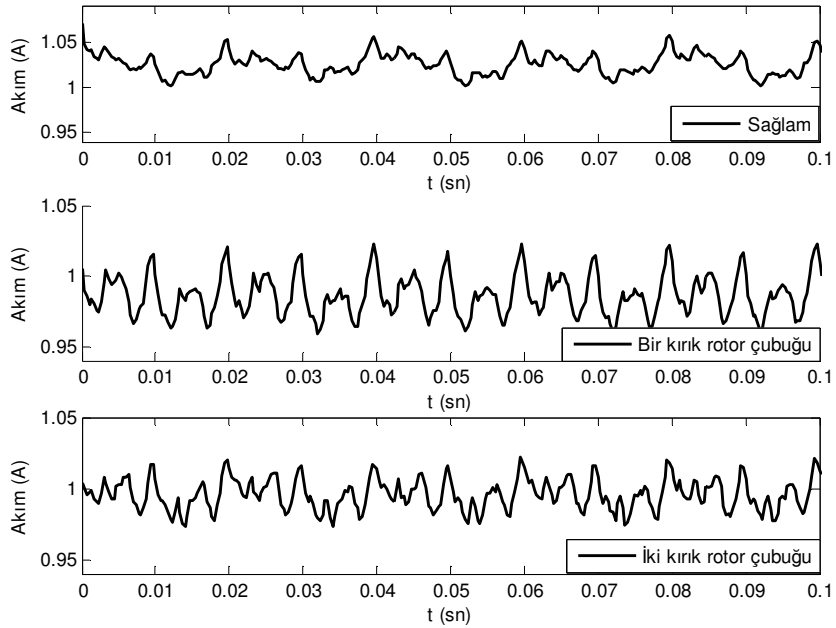
SÖTYBS yönteminin eğitim ve test verilerini oluşturacak özelliklerin çıkarımı için iki farklı yöntem kullanılmıştır. Her iki yöntem de akım sinyallerinden özellikler çıkarılmıştır. İlk yöntem akım sinyallerinin belirli periyotlarını alarak istatistiksel bazı özellikler ve MAIA tabanlı bir yan bant bileşeni elde edilir. İkinci yöntem ise üç faz akım sinyalini alarak park vektör dönüşümü ile iki bileşen oluşturmakta ve bu bileşenlerden bir özellik çıkarılmaktadır.

İlk yöntemde arıza ile ilgili özellikler tek faz akımından alınmıştır. Elde edilen özellikler tam yükte çalışan bir motorda kırık rotor çubuğu arızalarını belirlemek için de kullanılabilir. Hilbert dönüşümü tabanlı özellik çıkarım yöntemi ile yeni özellik sinyalini oluşturulmuştur. Hilbert dönüşümü ile elde edilen iki karmaşık bileşenin modülü alınarak özellik sinyali oluşturulmuştur. Şekil 4.19’da 3 kHz ile örneklenen sağlam ve arızalı akım sinyalleri gösterilmiştir.



Şekil 4.19. 380 V besleme geriliminde sağlam ve arızalı akım sinyalleri

Şekil 4.19’da verilen akım sinyalleri sağlam ve arızalı durumları ayırt etmek için uygun değildir. Bu yüzden arıza teşhisi için direkt olarak kullanımı uygun değildir. Şekil 4.20’de her bir sağlam ve arızalı motor akım sinyallerinin 10 periyoduna uygulanan Hilbert dönüşümü sonuçları verilmiştir.



Şekil 4.20. 380 V besleme geriliminde faz akımının Hilbert dönüşümü

Şekil 4.20’de elde edilen sinyal üzerinden her bir durum için bazı istatistiksel özellikler çıkarılmıştır. Bu özellikler enerji, maksimum, çarpıklık, basıklık ve özellik sinyalinin MAIA’dır. Özellik sinyalinin enerjisi aşağıdaki gibi elde edilir.

$$F_1 = \sum_{k=1}^w HD_k^2 \quad (4.12)$$

Bu denklemde HD_k sinyalinin k. örneğini gösterir. Denklemdeki w değeri ise pencere boyutunu gösterir. Mutlak değer maksimumu aşağıdaki gibi elde edilir.

$$F_2 = \max\{|HD_k|, k = 1 \dots w\} \quad (4.13)$$

Diğer iki özellik basıklık ve çarpıklık olup aşağıdaki gibi hesaplanırlar.

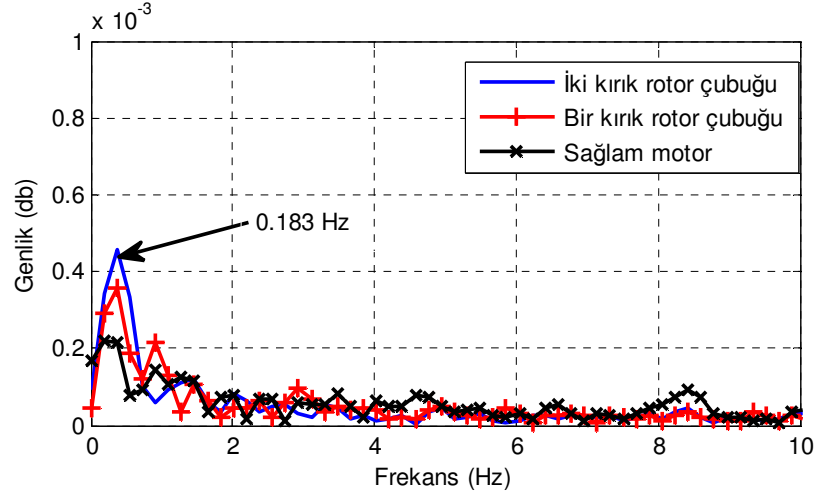
$$F_3 = \frac{\sum_{k=1}^w (HD_k - m_H)^4}{(w-1)\sigma_H^4} - 3 \quad (4.14)$$

$$F_4 = \frac{\sum_{k=1}^w (HD_k - m_H)^3}{(w-1)\sigma_H^3} \quad (4.15)$$

Denklemlerde kullanılan σ_H ve m_H sırasıyla standart sapma ve ortalamayı gösterir. Son özellik ise Hilbert dönüşümünün spektrumu için uygulanan MAIA’dır. Hilbert dönüşümünün bütün periyotları üst üste toplanır ve Hilbert dönüşümü hesaplanır. Özellik sinyali aşağıda verilmiştir.

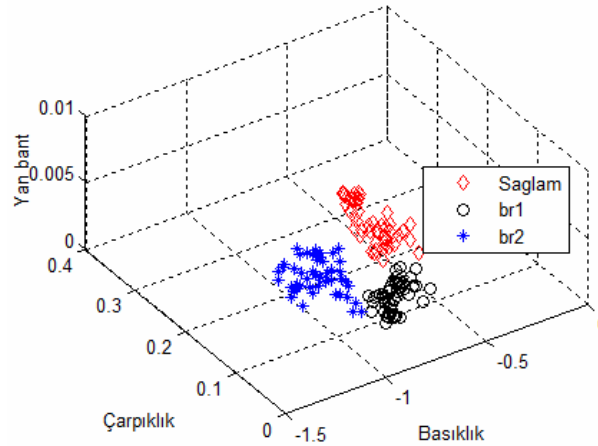
$$I_h = |HD(I(t))| - \text{mean}(|HD(I(t))|) \quad (4.16)$$

Denklemlerde $| |$ modülü hesaplamak için kullanılmıştır. Bu sinyal analitik sinyal ve onun ortalaması arasındaki farkı gösterir. Analitik sinyalin modülü faz akımının enerji değişimini yansıtır. Faz akımının enerjisi sağlam bir motorda zamanla değişmezken rotor arızasına sahip bir motorda salınım yapar. Özellik sinyalinin spektrumunda $2sf_h$ frekansında bir tepe noktası oluşur. Bu frekans kırık rotor çubuğu arızası ile ilgilidir. Yüksüz bir motorda kayma değeri %0,2 olarak alınır. Bu yüzden ilgili frekans 0.18 Hz olarak hesaplanır ($2sf_h = 2 * 50 * 0,002 = 0,2$ Hz). Sağlam ve arızalı durumlar için spektrum Şekil 4.21’de verilmiştir.



Şekil 4.21. Özellik sinyalinin spektrumu

Şekil 4.21’de görüldüğü gibi motorun sağlam ve arızalı durumları kolayca tanımlanabilmektedir. Arıza sınıflandırma süreci iki özellik grubu için yapılmaktadır. Her iki durumda da motor akım sinyalleri 3 kHz örnekleme frekansı ile 3 saniye boyunca elde edilmiştir. İlk özellik grubunda akım sinyalleri eşit ve sabit boyutlu pencerelere bölünmüştür. Periyot sayısı 3 alınarak her bir penceredeki örnek sayısı 180 olarak hesaplanmıştır. Her bir sağlam ve arızalı durum için 48 örnek elde edilmiştir. Şekil 4.22’de 380 V besleme gerilimi ile beslenen bir asenkron motor için veri kümesi verilmiştir.



Şekil 4.22. 380 V için özellik veri kümesi

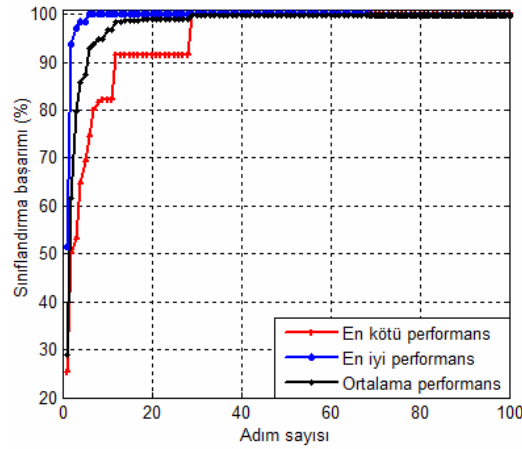
Şekil 4.22’de eğitim veri kümesi antijen olarak alınmaktadır. Bu antijenler için bir antikör popülasyonu üretilmekte ve popülasyon klonlama ile mutasyon işlemlerine maruz

kalarak optimum çözüm elde edilene kadar devam ettirilmektedir. Bütün eğitim veri kümesi eğitim başlamadan önce [0, 1] aralığına çekilmiştir. Bunun amacı çok geniş bir aralığa sahip bir özelliğin diğer özellikleri bastırmasını engellemektir. SÖTYBS'nin eğitiminde kullanılan parametreler Tablo 4.5'te verilmiştir.

Tablo 4.5. SÖTYBS'nin parametreleri

Parametre	Değeri
Klon_oran	5
Sınıf başına antikor sayısı	7
Adım sayısı	100
Popülasyon boyutu	21

SÖTYBS'nin performansı 40 adımdan sonra sabit olur. Şekil 4.23'te en iyi, en kötü ve ortalama eğitim performansı gösterilmiştir.



Şekil 4.23. SÖTYBS'nin yakınsama hızı

Şekil 4.23'te elde edilen sonuçlar eğitim veri kümesi üzerinde 10 katlı çapraz doğrulama ile elde edilmiştir. Önerilen yöntemin performansını antikor sayısı, sınıf sayısı ve özellik sayısı etkilemektedir. SÖTYBS aynı veri kümesi üzerinde yapay sinir ağları ile karşılaştırılmıştır. Beş giriş (F_1 , F_2 , F_3 , F_4 ve F_5) ve üç çıkışa (Sağlam, br_1 ve br_2) sahip ileri beslemeli üç katmanlı bir yapay sinir ağı geriye yayılım eğitim algoritması ile eğitilmiştir. Orta katmanda beş nöron kullanılmıştır. Çıkış veri kümesi her bir motor durumu için aşağıdaki şekilde seçilmiştir.

$[1\ 0\ 0]^T$ - Sağlam durum

$[0\ 1\ 0]^T$ - Bir kırık rotor çubuğu

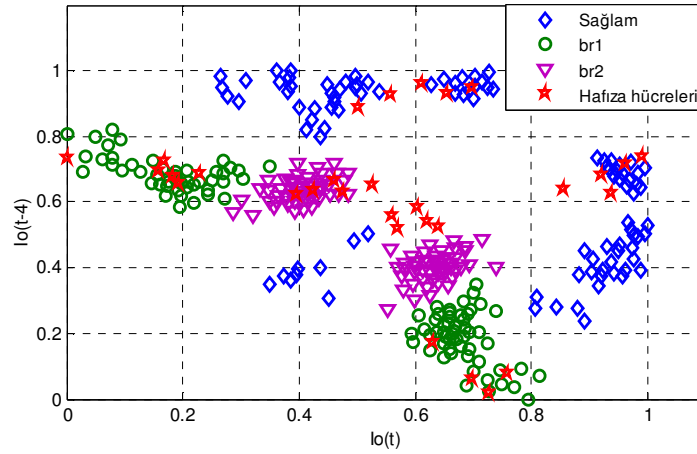
$[0 \ 0 \ 1]^T$ - İki kırık rotor çubuğu

Tablo 4.6’da veri kümesinin ilk grubu üzerinde yapay sinir ağı ve SÖTYBS yönteminin performansları verilmiştir. Doğruluk oranı dışında standart sapma da gösterilmiştir.

Tablo 4.6. İlk özellik grubu için arıza teşhis doğruluğu

Besleme gerilimi (V)	Doğruluk oranı (%)	
	Yapay sinir ağları	SÖTYBS
380	98.42 ± 4.99	99.28 ± 2.26
340	96.25 ± 3.22	98.78 ± 2.28
300	100 ± 0.00	100 ± 0.00
260	100 ± 0.00	100 ± 0.00

Tablo 4.6’da görüldüğü üzere SÖTYBS yöntemi yapay sinir ağından daha iyi sonuçlar vermektedir. İkinci özellik grubu her bir motor durumu için oluşturulan faz uzayları üzerinden elde edilmektedir. Destek vektör makinaları tabanlı sınıflandırıcıda elde edilen özellik sinyalleri SÖTYBS’nin performansının değerlendirilmesi için kullanılmıştır. Sağlam ve arızalı durumların faz uzayları lineer olmayan bir davranış sergiler. Her bir sınıf için 10 antikör seçilmiştir. Adım sayısı 100 olarak seçilmiştir. Her bir durum için faz uzayları ve eğitim sonucu elde edilen hafıza hücreleri Şekil 4.24’te verilmiştir.



Şekil 4.24. Üretilen hafıza hücreleri ve her bir sınıfın faz uzayları

İkinci veri kümesi üzerinde SÖTYBS’nin başarımı aynı zamanda yapay sinir ağı ve DVM ile karşılaştırılmıştır. Destek vektör makinalarının çekirdek ve penaltı parametresi çok amaçlı klonal seçim ile ayarlanmıştır. Bu veri kümesi karmaşık ve lineer olmayan

özellikler gösterir. Veri kümesi üzerinde SÖTYBS, yapay sinir ağı ve DVM yöntemlerinin performansı Tablo 4.7’de verilmiştir.

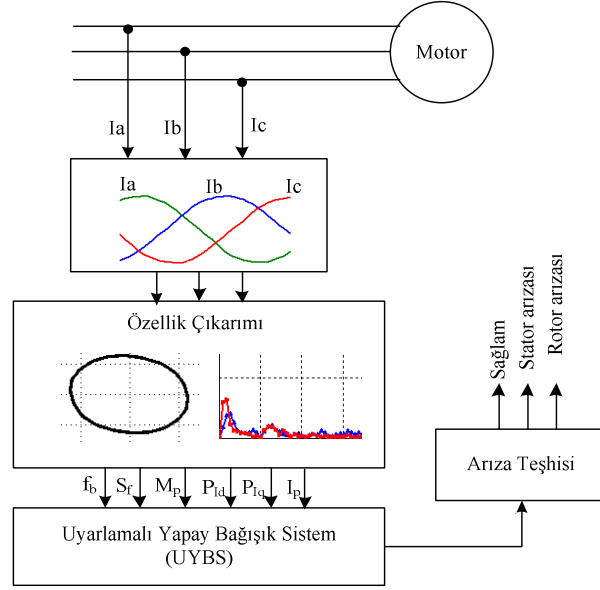
Tablo 4.7. İkinci özellik grubu üzerinde üç yöntemin performans karşılaştırması

Yöntem	Eğitim veri kümesi boyutu	Doğruluk oranı (%)
SÖTYBS	300	97.91 ± 1.93
DVM	300	98.95 ± 1.20
Yapay sinir ağı	300	96.40 ± 1.21

Seçilen ileri beslemeli yapay sinir ağı gizli katmanda 40 adet nörona sahiptir. Destek vektör makinaları normalde iki sınıflı bir sınıflandırıcı olmasına rağmen çoklu sınıflandırma için burada birine karşı diğerleri yöntemi kullanılmıştır. SÖTYBS yöntemi yapay sinir ağı ve DVM ile karşılaştırıldığında oldukça iyi sonuçlar verdiği sadece DVM’nin performansının biraz daha iyi olduğu görülür. İki özellik grubunun sonuçları karşılaştırıldığında birinci özellik grubunun daha kolay ayırt edilebilir olduğu ve çevrimiçi izleme için daha az veri depoladığı görülmektedir.

4.3. Uyarlamalı Yapay Bağışık Sistem (UYBS) Tabanlı Arıza Teşhis Yöntemi

Asenkron motorlarda oluşan arızaların erken bir aşamada teşhisi için uyarlamalı bir yapay bağışık algoritması önerilmiştir. Önerilen algoritma, her adımda ortalama afinite değişimi ile elde edilen standart sapmanın büyüklüğünü kullanarak hafıza hücrelerini ayarlar. Temel olarak algoritma üç aşamadan oluşmaktadır. Birinci aşamada üç faz motor akım sinyali akım duyargaları ile ölçülerek veri toplama kartı aracılığıyla bilgisayara aktarılmaktadır. Elde edilen akım sinyallerinden arıza ile ilgili özellikler ikinci aşamada elde edilmektedir. Sağlam ve arızalı durumlar için elde edilen özellik sinyalleri uyarlamalı bağışık sisteme verilerek kırık rotor çubuğu ve stator arızaları belirlenmiştir. Şekil 4.25’te arıza teşhis ve tespiti için önerilen yöntemin blok şeması verilmiştir.



Şekil 4.25. Arıza teşhis ve tespiti için önerilen yöntemin blok şeması

Şekil 4.25'te asenkron motorun üç fazından elde edilen akım sinyallerinden altı özellik elde edilmekte ve bu özellikler uyarlamalı bağışık sisteme verilmektedir. Uyarlamalı bağışık sistemi gelen eğitim verilerine göre motor durumunu öğrendikten sonra eğitim aşamasında elde edilen hafıza hücrelerini kullanarak arızaları sınıflandırmaktadır.

4.3.1. Özellik Çıkarımı

Özellik çıkarım aşamasında akım sinyalleri kullanılarak arıza ile ilgili özellikler elde edilir. Motor durumu her bir arıza durumu için belirlenen sınıflardaki referans değerler ile karşılaştırılarak belirlenir. Motor akım sinyalleri stator ve kırık rotor çubuğu arızalarının belirlenmesi için oldukça kullanışlıdır [131]. Akım sinyalinden rotor ve stator durumu için birer adet ve teşhis algoritmasının doğruluğunun artırılması için dört adet olmak üzere altı özellik elde edilmektedir.

Orijinal motor akım imza analizi tek faz akım sinyali üzerinden Fourier dönüşümü uygulayarak kırık rotor çubuğu ile ilgili frekans bileşenlerinin büyüklüğünü inceler. Bu yöntem motor yüklü durumda ise iyi sonuç vermektedir. Fakat yöntemin endüstriyel ortamlarda uygulanması zordur. Ayrıca düşük yükte çalışma durumlarında yöntem ile kırık rotor çubuklarının tespiti oldukça güç olmaktadır. Bu yüzden orijinal motor akım imza analizi yöntemi akım sinyaline direkt olarak uygulanmamaktadır. Tek faz akım sinyaline Hilbert dönüşümü uygulanarak yeni özellik sinyali elde edilmekte ve bu özellik sinyaline

motor akım imza analizi uygulanmaktadır. Arıza teşhisinde kullanılacak özellik sinyali denklem 4.17'ye göre elde edilmektedir.

$$I_h = |HD(I(t))| - \text{mean}(|HD(I(t))|) \quad (4.17)$$

Bu denklemde $| |$ karmaşık Hilbert bileşenlerinin büyüklüğünü gösterir. $H(I(t))$ ifadesi akım sinyalinin Hilbert dönüşümünü gösterir. Özellik sinyali, analitik sinyal ve bu sinyalin ortalama değerinin farkı alınarak oluşturulur. Faz akımının enerjisi sağlam bir asenkron motorda zamanla değişmezken kırık rotor çubuğu arızası durumunda bir salınım yapar [132]. Oluşturulan özellik sinyalinin frekans spektrumu ayrık Fourier dönüşümü uygulanarak elde edilir. Hilbert dönüşümü ile oluşturulan I_h sinyalinin spektrumu $2sf_h$ frekansında bir tepe noktası oluşturur. Kırık rotor çubuğu ile ilgili özellik denklem 4.18'de verilmiştir.

$$f_b = 2sf_h \quad (4.18)$$

Burada f_h hat frekansını ve f_b ise kırık rotor çubuğu ile ilgili yan bant frekansını göstermektedir. Denklemde kullanılan s değeri ise rotor kaymasını ifade eder.

Stator arızası ile ilgili özellikler üç faz akım sinyali kullanılarak elde edilmektedir. İlk olarak üç faz motor akım sinyalinden iki boyutlu Park vektör bileşenleri elde edilmektedir. Yeni özellik sinyali bu bileşenler kullanılarak elde edilecektir. Temel bileşen analizi yöntemi $x=(x_1, x_2, x_3, \dots, x_p)$ vektörü ile gösterilen p adet orijinal değişkeni yine p adet ilişkisiz değişken $z=(z_1, z_2, z_3, \dots, z_p)$ ile yeni bir kümeye dönüştürür. Yeni koordinatlar kümesi orijinal lineer kombinasyonu ile oluşturulan temel bileşenler olarak adlandırılır. Örneğin z_1 ilk temel bileşeni en büyük veri dağılımının olduğu yönü gösteren ilk temel bileşendir. Yeni koordinat sistemi u_i ile ifade edilen ve korelasyon matrisinden hesaplanan i adet öz vektör ile ifade edilir. Her öz vektör veri kümesinin dağılımının yönünü gösterir. Öz vektörlerin oluşturduğu küme x 'in koordinatlarını temel bileşenlere dönüştürmek için kullanılan U dönüşüm matrisini sağlar. Bu dönüşüm denklem 4.19'da gösterilmiştir.

$$z = U^T x \quad (4.19)$$

Her bir temel bileşen z_i sıfır ortalama değeri ve u_i vektörünün öz değerine karşılık gelen bir λ_i varyans değerini gösterir. Öz değer λ_i toplanan veri kümesinin öz vektör yönünde nasıl dağılım gösterdiğini belirleyen belirli bir ağırlık verecektir. Temel bileşen

analizinin bu özellikleri 2 boyutlu park vektör bileşenlerinin her iki boyuttaki maksimum dağılımlarını elde etmek için kullanılmıştır. İlk olarak elde edilen I_d ve I_q park vektör bileşenleri $[I_d \ I_q]$ şeklinde bir X matrisi formunda yazılır. Daha sonra korelasyon matrisi bulunur. Korelasyon matrisinin öz değerleri ile öz vektörler arasında aşağıdaki ilişki vardır.

$$K_M * v = v * \lambda \quad (4.20)$$

Yukarıdaki denklemde elde edilen öz vektörler orijinal özellik kümesi (X) ile çarpılarak yeni sinyal aşağıdaki gibi elde edilmektedir.

$$X_o = X * v \quad (4.21)$$

Yeni özellik matrisi $X_o [I_{d0} \ I_{q0}]$ ile gösterilir. Özellik sinyali bu iki boyutun maksimum değerlerini kullanarak aşağıdaki şekilde elde edilir.

$$S_f = 1 - \frac{\max(I_{d0})}{\max(I_{q0})} \quad (4.22)$$

Arıza teşhis sisteminin daha detaylı incelenmesi ve sağlamlığının artırılması için bazı ekstra özellikler de elde edilmiştir. Bu özellikler aşağıda verilmiştir:

- Tek faz akımının maksimum gücü (M_p):

$$M_p = \frac{1}{N} \sum_{k=1}^N \|I_A(k)\|^2 \quad (4.23)$$

- I_d ve I_q bileşenlerinin tepeden tepeye değerleri arasındaki fark:

$$P_{Id} = |\max(I_d) - \min(I_d)| \quad (4.24)$$

$$P_{Iq} = |\max(I_q) - \min(I_q)| \quad (4.25)$$

- Son özellik ise karmaşık pozitif sıra bileşenin modülünün ortalaması olarak alınmıştır.

$$I_p = \text{mean} \left| \frac{1}{3} (I_a + \alpha I_b + \alpha^2 I_c) \right|, \quad \alpha = e^{j(2\pi/3)} \quad (4.26)$$

4.3.2. Uyarlamalı Bağışık Sistem Kullanan Arıza Teşhis Aşaması

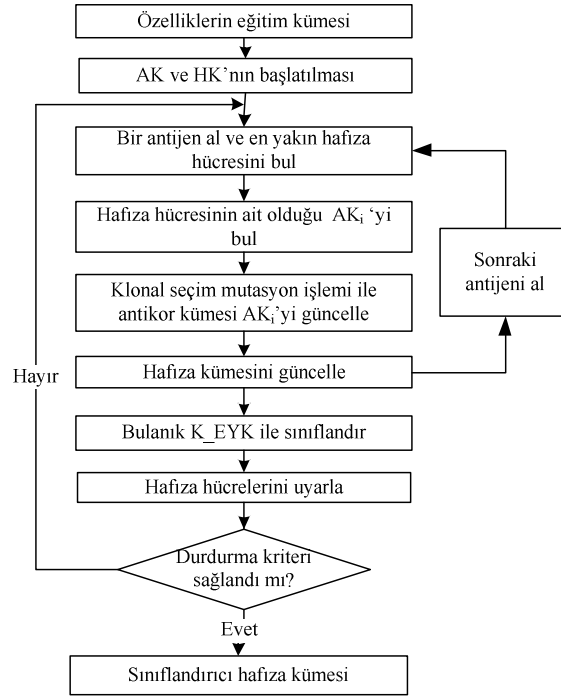
Önerilen uyarlamalı bağışıklık sistemi iki aşamadan oluşmaktadır. İlk aşamada asenkron motordan elde edilen akım sinyalleri ile oluşturulan özellikler eğitim ve test verisi olarak ayrılmakta ve antikörlerin başlangıç popülasyonu rastgele başlatılmaktadır. Daha sonra bağışık sistem hafıza hücrelerini elde etmek için eğitilmektedir. İkinci aşamada test antijenleri ve hafıza hücreleri kullanılarak arızaları sınıflandırmak için bulanık K-en yakın komşu algoritması kullanılmaktadır. K-en yakın komşu sınıflandırıcı eğitici bir sınıflandırma algoritması olup bir test örneğinin sınıfı bu örneğe en yakın K adet komşu ile belirlenmektedir [133]. Sınıfın belirlenmesinde en çok tercih edilen yöntem ise ağırlıklı oylamadır. Fakat yöntem bazı dezavantajlara sahiptir. Birincisi her komşu giriş verisinin sınıflandırılmasında eşit ağırlığa sahiptir. Bu durum veri örnekleri örtüştüğünde yanlış sınıflandırmaya neden olur. İkincisi ise K-en yakın komşu algoritması test verisine sadece bir sınıf etiketi atayıp sınıf içerisindeki üyelik derecesi belirlenmez. Bulanık K-en yakın komşu algoritması klasik K-en yakın komşu algoritmasındaki bu sınırlamaları gidermek için kullanılan bir sınıflandırma yöntemidir [134]. Bu yöntem her bir örneğe bu örneğin en yakın K komşusuna olan mesafesine göre bir bulanık üyelik derecesi atar. Üyelik derecesi aşağıdaki gibi hesaplanmaktadır.

$$\mu_{\omega_i}(x) = \frac{\sum_{j=1}^k \mu_{\omega_i}(x_j) d_j^{-2/(M-1)}}{\sum_{j=1}^k d_j^{-2/(M-1)}} \quad (4.27)$$

Denklemdaki $x_1, x_2, \dots, x_k$ test örneğinin k- en yakın komşusu olup $d_j = \|x - x_j\|$ ifadesi ise x örneği ve ona en yakın j. örnek arasındaki mesafedir. Bir test örneğinin sınıfı denklem 4.28'e göre hesaplanmaktadır.

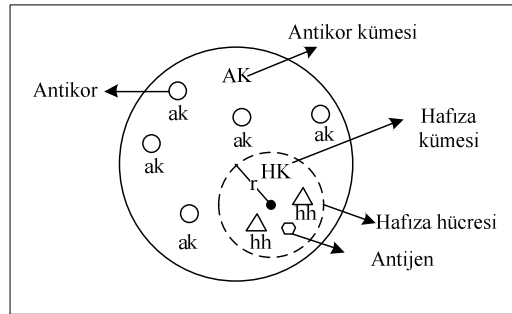
$$\omega(x) = \arg \max_{i=1..M} (\mu_{\omega_i}(x)) \quad (4.28)$$

Bu denklemde M sınıf sayısını gösterir. Herhangi bir örneğin sınıfı, örneğin bütün sınıflara olan üyelik dereceleri arasından maksimum olana göre belirlenir. Önerilen uyarlamalı yapay bağışık sistemin akış şeması Şekil 4.26'da verilmiştir.



Şekil 4.26. UYBS'nin akış şeması

Öğrenme algoritmasının temel prensibi biyolojik bağışık sistemindeki klonal seçime dayalıdır. Eğitim aşamasının amacı, hafıza hücrelerinin oluşturulması, motorun her bir durumunun gösterilmesi ve arıza teşhisidir. Yeni bir antijen antikorlara verildiği zaman, bu antijene yüksek afinite ile bağlanan antikorların genişlemesi sağlanır [135]. Daha sonra klon antikorlar mutasyon işlemine tabii tutulurlar. Bu süreç, antikorların çoğalma ve yok olma prensibine göre sürekli değişimini sağlar. Eğitim aşamasında birçok antikor olmasına karşın, arızalı örnekleri sınıflandırmak için daha az hafıza hücresi kullanılacaktır [119]. Antikor ve hafıza hücresinin gösterimi Şekil 4.27'de verilmiştir.



Şekil 4.27. Antikor ve hafıza hücresinin gösterimi

Şekil 4.27’de antikor kümesi (AK) bir sınıf için veriyi gösterir. Antijenleri tanıyan antikorlar hafıza hücresi (hh) olarak alınır. Hafıza kümesi (HK) hafıza hücrelerinden oluşur. Antikor kümesi r yarıçap aralığındaki antijenleri tanır. Bir antikorun afinitesi antikor ile antijen arasındaki mesafe ile ters orantılıdır. Antijen ile bir hafıza hücresi arasındaki mesafe aşağıdaki gibi hesaplanmaktadır.

$$Afinite(an, hh) = e^{\left(\frac{\|hh-ag\|}{2r^2}\right)} \quad (4.29)$$

Her bir sınıf için bir antikor kümesi tanımlanır. Denklemdeki $\| \cdot \|$ ifadesi hafıza hücresi ve antijen arasındaki Öklid mesafesini tanımlar. Öğrenme algoritmasının adımları aşağıda verilmiştir.

Adım 1: Popülasyonun başlatılması Başlama aşamasında asenkron motordan alınan sağlam ve arızalı veri örnekleri antijen kümesi (AN) olarak alınır. Antikor kümesi ise AN’den rastgele üretilir. Daha sonra bir hafıza kümesi AK’den elde edilir.

Adım 2: Bir eğitim örneği al (an_j): Algoritmada eğitim örneği antijen olarak tanımlanmıştır ve burada özellik vektörü eğitim örneğini oluşturur.

Adım 3: Eğitim örneğine en yakın hafıza hücresini bul: Bir antijen için, en yakın hafıza hücresini (mc_i) aşağıdaki gibi bul.

$$mc_i = \arg \max_{mc \in M} afinite(an_j, hh) \quad (4.30)$$

Adım 4: Antikor kümesinin gelişimi: Antijene en yakın hafıza hücresi bulunduğundan sonra hafıza hücresinin ait olduğu antikor kümesi (AK_i) çoğalma ve mutasyon işlemine tabii olur. AK_i popülasyonundaki her bir ak ’nın an_j ’ye göre afinitesi belirlenir ve n adet yüksek afiniteli antikor klon kümesini (C_i) oluşturmak için seçilir. Klon sayısı (N_C) aşağıdaki gibi hesaplanır.

$$N_c = \sum_{k=1}^n round(k_oran * afinite(an, ak_k)) \quad (4.31)$$

Denklemden k_oran an_j için üretilen klon sayısını gösterir. Round() fonksiyonu ise değeri en yakın tamsayıya yuvarlar.

Adım 5: Mutasyon işlemi: C_i klon kümesindeki her antikor mutasyon işlemine tabii tutulur. Mutasyon bir antikorun özellikleri üzerinde oluşur. Mutasyon işlemi afinite ile ters

orantılı olup afinite ne kadar yüksek ise mutasyona uğrama olasılığı o kadar düşük olur. Bu işlem Şekil 4.28’de verilmiştir.

```

for i=1:Nc
    M_oran=1-afinite(ag, abi)
    for k=1:ozellik_sayısı
        d=random(-M_oran, M_oran);
        Cjm(i, k)=Cj(i, k)+d*M_oran;
        if(Cjm(i, k)<0)
            Cjm(i, k)=0;
        end
        if(Cjm(i, k)>1)
            Cjm(i,k)=1;
        end
    end
end
end

```

Şekil 4.28. C_j için mutasyon işlemi

Şekil 4.28’de $random(-M_{oran}, M_{oran})$ fonksiyonu $-M_{oran}$ ile M_{oran} arasında rastgele bir reel sayı üretir. Özellik vektörünün herhangi bir elemanı problem uzayını aştığında değeri sıfır veya bire getirilir.

Adım 6: Afinite hesaplama: Mutasyona uğrayan klonlar (C_{jm}) ile antijen (an_j) arasındaki afiniteyi hesapla.

Adım 7: Antikor ve hafıza kümesini güncelle: Antijene (an_j) en yakın antikor kümesi mutasyona uğrayan klon kümesi kullanılarak güncellenir. AK_j ’deki en düşük ε afiniteli antikorun C_{jm} kümesindeki d en yüksek afiniteli antikor seçilir. Buradaki ε parametresi değişim parametresidir. Eğer C_{jm} ‘de bulunan aday hafıza hücresi hafıza kümesindeki en iyi çözüme benzer ise, hafıza hücresi yeni aday ile değiştirilir. Bu amaçla bir benzerlik ölçüsü ($b\ddot{o}$) belirlenmiştir. Bu ölçüm benzer hafıza hücrelerinin üretimini engellemektedir.

Adım 8: Sonraki antijeni al ve 2. Adıma git.

Adım 9: Bulanık K-en yakın komşu ile sınıflandırma: Eğitim aşaması ile hafıza hücrelerinin her güncellenmesi sonucunda bulanık en yakın komşu sınıflandırma algoritması ile elde edilen hücrelerin sınıflandırma başarımı test edilir.

Adım 10: Hafıza Hücrelerinin uyarlanması: Hafıza hücrelerinin özellik uzayını maksimum kapsamasını sağlamak için, bir uyarlama süreci uygulanmıştır. Bu süreç her hafıza hücresinin sınıflandırma performansını artırır. Her hafıza hücresinin pozisyonu dinamik olarak bütün hafıza kümesinin ortalama afinitesine göre değişir. Her hafıza

hücresi, hafıza kümesinin varyansı kullanılarak güncellenir.

$$\sigma^2 = \sum_{i=1}^{n_{mc}} \left(\frac{aff_{mci} - aff_{avg}}{aff} \right) \quad (4.32)$$

Denklem 4.32’de aff_{avg} geçerli adımda hafıza kümesinin ortalama afinitesini, aff_{mci} ise popülasyondaki i . hafıza hücresinin afinitesini ve n_{mc} ise toplam hafıza hücrelerinin sayısını gösterir. Denklemdeki aff değeri ise aşağıdaki gibi hesaplanır.

$$aff = \max(|aff_{mci} - aff_{avg}|), i = 1, 2, \dots, M \quad (4.33)$$

Denklem 4.33’te aff standart sapmayı (σ) sınırlamak için kullanılan normalizasyon faktörüdür. Eğer σ büyük ise hafıza hücresi optimum çözümden uzaklaşır. Buna karşın küçük σ değerleri için hafıza hücrelerinin sınıflandırma performansı erken yakınsamaya giderek en iyi hafıza hücreleri elde edilir. En iyi performansı elde etmek için hafıza hücreleri standart sapmanın değişimine göre güncellenir. Standart sapmanın değişimi aşağıdaki gibi hesaplanır.

$$\Delta\sigma = \sigma(i+1) - \sigma(i) \quad (4.34)$$

Standart sapmadaki değişim elde edildikten sonra aşağıdaki denkleme göre hafıza hücresi güncellenir.

$$mc(i, j) = mc(i, j) + rand() * |\Delta\sigma|, \text{ for } i = 1, \dots, m \text{ and } j = 1, \dots, n \quad (4.35)$$

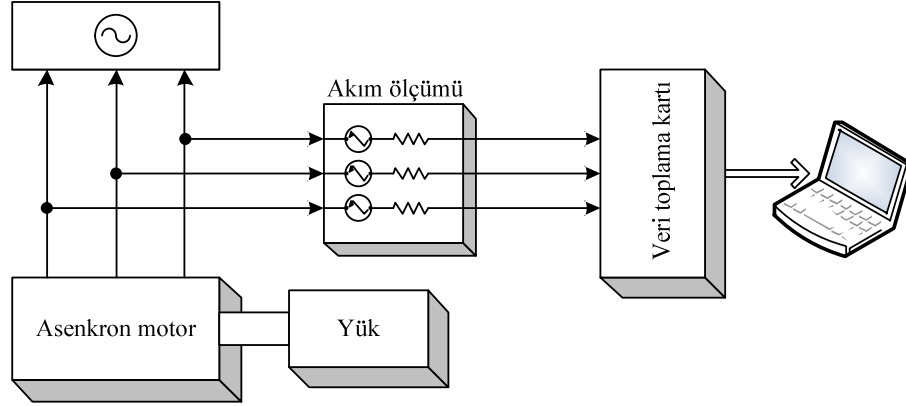
Denklem 4.35’te m hafıza hücrelerinin sayısını ve n ise her bir hafıza hücresindeki özellik sayısını gösterir. Algoritmanın yakınsama hızı bu adımda arttırılmaktadır. Eğer algoritma maksimum performansına ulaşırsa iki ardışık adım arasındaki standart sapma çok küçük olacaktır. Böylece hafıza hücreleri çok az değişecektir.

Adım 11: Durdurma kriterinin değerlendirilmesi: Belirlenen maksimum adım sayısı boyunca önceki adımlar devam ettirilir.

4.3.3. Deneysel Sonuçlar

Önerilen UYBS sistemin performansı gerçek bir asenkron motordan alınan veriler ile değerlendirilmiştir. Motor sağlam durum, stator açık faz arızası ve bir kırık rotor çubuğu

arızası ile test edilmiştir. Her bir durum için besleme gerilimleri 300, 340 ve 380 V olarak seçilmiştir. Her bir test durumu için 105 adet motor akım sinyali 3 kHz örnekleme frekansı ile 3 saniye boyunca kaydedilmiştir. Ölçülen bu akım değerleri 16 bitlik Advantech PCL-816 model bir veri toplama kartı aracılığıyla bilgisayara aktarılmıştır. Kurulan deney düzeneğinin blok şeması Şekil 4.29’da verilmiştir.



Şekil 4.29. Deney düzeneği şeması

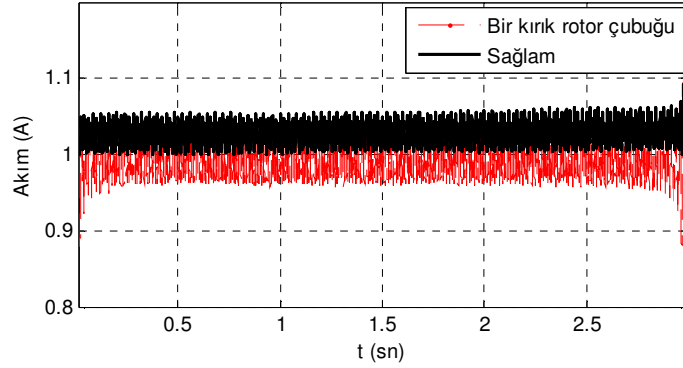
Örnek sayısının büyük seçilmesinin amacı akım özellik sinyallerinin sağlamlığını değerlendirmektir. Asenkron motor sağlam, bir kırık rotor çubuğu arızası ve stator arızası durumları altında ayrı ayrı çalıştırılmıştır. Her bir durum için Tablo 4.8’de sınıflandırmada kullanılan örneklerin sayısı verilmiştir.

Tablo 4.8. Eğitim ve test verilerinin özellikleri

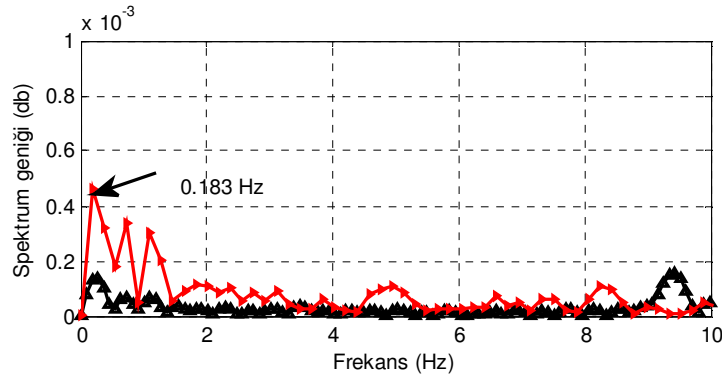
Sınıf	Çalışma modu	Besleme gerilimi	Örnek sayısı
H	Sağlam durum	380 V	15
		340 V	15
		300 V	15
B1	Bir kırık rotor çubuğu	380 V	15
		340 V	15
		300 V	15
S	Stator arızası	380 V	15

Şekil 4.30 (a)’da 380 V besleme gerilimi altında çalışan sağlam ve bir kırık rotor çubuğu arızasına sahip motor için tek faz akımına uygulanan analitik sinyalin modülü gösterilmektedir. Aynı besleme gerilimi altında sağlam durum ve bir kırık rotor çubuğu

arızası için analitik sinyalin MAIA Şekil 4.30 (b)'de gösterilmiştir. Şekil 4.26 (b)'de arızalı ve sağlam durumlar kırık rotor çubuğu ile ilgili yan bant bileşenleri incelenerek açık bir biçimde ayırt edilebilmektedir.



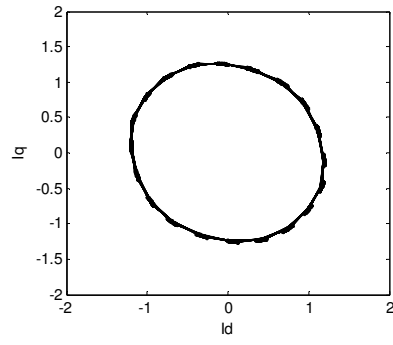
(a) Sağlam ve bir kırık rotor çubuğu arızası için tek faz akımının modülü



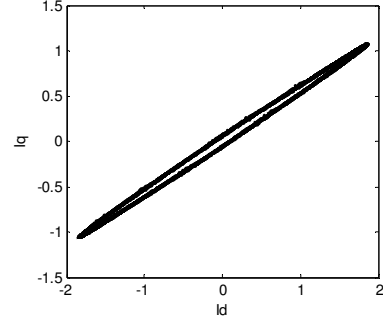
(b) Motor akım imza analizi ile elde edilen f_b frekansı

Şekil 4.30. Kırık rotor çubuğu için kullanılan hilbert dönüşümü ve spektrumu

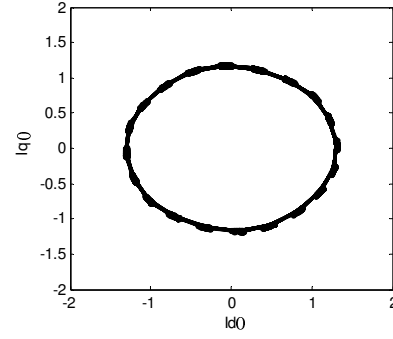
Bu frekansın yeri $2sf_s$ denkleminde 0.183 Hz olarak hesaplanmaktadır. Orijinal motor akım imza analizinde en az iki yan bant bileşeni arızaları tespit etmek için değerlendirilmektedir [90]. Önerilen tek faz akım sinyali üzerinden sadece bir frekans bileşenine ihtiyaç duyar. Kayma değeri mutlak yüksüz çalışma durumunda % 0.2 olarak hesaplanır. Stator arızaları için iki boyutlu park vektör bileşenleri ve onların döndürülmüş versiyonları Şekil 4.31'de verilmiştir.



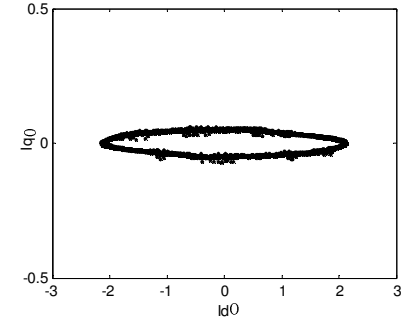
(a) Sağlam motor park vektör dönüşümü



(b) Stator arızası park vektör dönüşümü



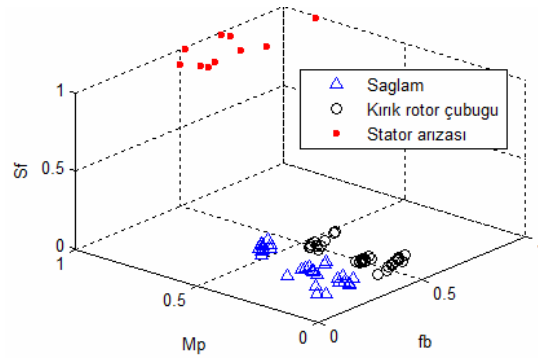
(c) Sağlam motor yeni özellik



(d) Arızalı motor yeni özellik

Şekil 4.31. Sağlam ve arızalı motor için park vektör dönüşümleri ve temel bileşen analizi tabanlı yeni dönüşüm

Diğer özellikler farklı besleme gerilimleri altında metodun doğruluğunu arttırmak için kullanılmıştır. Şekil 4.32’de üç sınıf için eğitim veri kümesinin 3 boyutlu gösterimi verilmiştir.



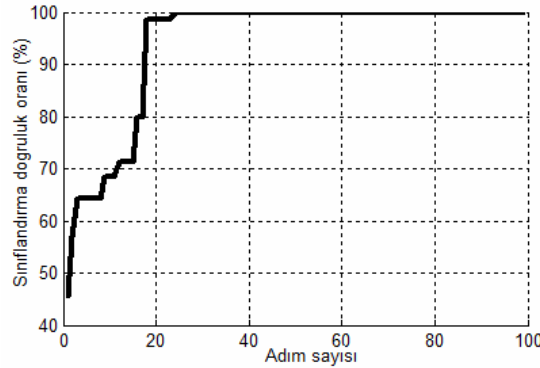
Şekil 4.32. Üç sınıf verisinin üç boyutlu gösterimi

Arıza ile ilgili özellikler elde edilip eğitim veri kümesi oluşturulduktan sonra öğrenme algoritması bağışık sistemin sınıflandırıcı hafıza kümesini elde etmek için kullanılır. Bağışık sistemin bazı parametreleri motor durumunu en iyi sınıflandırılması için ayarlanmalıdır. Kullanılan parametreler ve değerleri Tablo 4.9’da verilmiştir.

Tablo 4.9. UYBS’nin parametreleri

Parametre	Değeri
k_oran	10
r	2.65
Adım sayısı	100
bö	0.35
ε	0.1

Uyarlamalı yapay bağışık sistemin sınıflandırma başarımının eğitim aşamasında gelişimi Şekil 4.33’te verilmiştir. Burada sınıflandırma başarımı algoritmanın adım sayısının bir fonksiyonu olarak eğitim veri kümesi üzerinde gerçekleştirilmiştir.



Şekil 4.33. Eğitim aşamasında UYBS’nin sınıflandırma başarımı

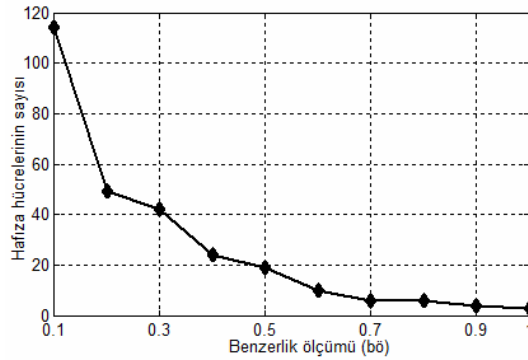
Önerilen algoritmanın sonuçları literatürdeki iki bağışık sınıflandırıcı ile karşılaştırılmıştır. Diğer bağışık sınıflandırıcılar ile karşılaştırma yapabilmek için 10 kat çapraz doğrulama yönteminden faydalanılmıştır. Her bir veri kümesi için ortalama adım sayısı, sabit adım sayısını belirlemek için gereklidir. Tablo 4.10 önerilen yöntemin kaynak sınırlı yapay bağışık sınıflandırıcı (RLAIC) [114] ve basit yapay bağışık sistem (SAIS) [136] ile karşılaştırma sonuçlarını göstermektedir. Bu sonuçlar her bir yöntemin on defa çalıştırılması ile elde edilmiştir. Performans karşılaştırması hafıza hücrelerinin sayısı, doğruluk oranı ve adım sayısına göre yapılmıştır.

Tablo 4.10. Diğer bağışık sınıflandırıcılar ile karşılaştırma

Yöntem	Doğruluk oranı	HH sayısı	Adım sayısı
RLAIC [111]	94.75 ± 4.5	30	20
SAIS [132]	96.34 ± 3.75	60	150
UYBS	98.14 ± 5.87	24	100

Adım sayısı ve hafıza hücrelerinin sayısı sınıflandırıcıların performansını değerlendirmek için iki önemli parametredir. Tablo 5’te görüldüğü gibi önerilen uyarlamalı bağışık sistem algoritması SAIS ve RLAIC’ten daha iyi sonuç vermektedir. Önerilen yöntem hafıza hücreleri için kullandığı uyarlama fonksiyonu ile sınıflandırma için gerekli olan hafıza hücrelerinin sayısını azaltır. Yöntem eğitim aşamasında 100 adım çalıştırılmaktadır. Bununla birlikte algoritmanın performansı daha düşük adımda maksimum performansına ulaşmaktadır.

Uyarlamalı yapay bağışık sistemde benzerlik ölçümü (bö) hafıza hücrelerinin güncellenmesi ve çeşitliliğinin korunması için oldukça önemlidir. Şekil 4.34’te seçilen bö değerine göre üretilen hafıza hücrelerinin sayısı verilmiştir.



Şekil 4.34. Benzerlik ölçümüne göre hafıza hücrelerinin sayısı

Şekil 4.31’de bö=1 değeri için hafıza kümesinin boyutu sınıf sayısına eşit olur. Bu değer ne kadar küçük seçilirse hafızanın boyutu o kadar artar.

4.4. Bölüm Değerlendirmesi ve Sonuçlar

Tezin bu bölümünde arıza teşhisi için üç yöntem sunulmuştur. Birinci yöntem farklı besleme gerilimleri altında kırık rotor çubuğu arızalarının tespiti için destek vektör makinalarını kullanmaktadır. Arıza teşhisi için iki park vektör bileşeninden yeni bir özellik sinyali elde edilmiştir. Bu özellik sinyali faz akım sinyalinin pozitif bileşenlerindeki

değişimleri almaktadır. Sağlam ve her bir arızalı durum için elde edilen özellik sinyallerinin faz uzayları destek vektör makinalar ile sınıflandırılmıştır. Destek vektör makinaların çekirdek ve ceza parametreleri klonal seçim algoritması ile ayarlanmıştır. Önerilen yöntem minimum sayıda destek vektör kullanarak sınıflandırma başarımını maksimum yapmaktadır. İkinci yöntemde arıza teşhisi için sürü öğrenme tabanlı bağışık bir sınıflandırıcı önerilmektedir. Bağışık sistemlerdeki mutasyon işlemi sürü öğrenmesi ile gerçekleştirilerek bağışık sistemin performansı arttırılmıştır. Bağışık sistemde oluşturulan her bir klon gideceği yönü bilmektedir. Ayrıca sürü öğrenmedeki anti çarpışma özelliği kullanıldığından aynı antikorların üretimi engellenmiştir. Yapay bağışık sistemlerde popülasyon boyutu sürekli artmasına rağmen, önerilen yöntemde sabit olduğundan eğitim daha hızlı yapılmaktadır. Akım sinyallerinden çıkarılan çok boyutlu özellik sinyalleri üzerinde yöntemin performansı değerlendirilmiş ve iyi sonuçlar alınmıştır. Arıza teşhisi için faz akım sinyalinin minimum sayıda periyodu kullanılmıştır. Dolayısıyla önerilen özellik sinyali az veri depolayarak arızaları belirleyebilmektedir. Yöntemin performansı aynı zamanda destek vektör makinalar aşamasında sunulan özellik sinyali üzerinde test edilmiş ve iyi sonuçlar elde edilmiştir. Arıza teşhisi için önerilen üçüncü yöntem stator ve rotor arızalarının tespiti için üç faz akımını kullanmaktadır. Stator arızaları için park vektör dönüşümü ve temel bileşen analizi kullanılarak elde edilen ayırt edici özellik sinyali faydalanılmıştır. Rotor arızaları için MAIA tabanlı bir yöntem sunulmuştur. Ayrıca arıza teşhis edicinin daha iyi karar verebilmesi için ek özellikler elde edilmiştir. Arıza teşhisi için önerilen yöntem temel olarak klonal seçim ve hafıza hücrelerinin gelişimine dayalıdır. Üretilen hafıza hücreleri bulanık K- en yakın komşu algoritmasına verilerek sınıflandırma işlemi yapılmaktadır. Önerilen yöntemin yakınsama hızı hafıza hücreleri üzerinde uyarılama işlemi yapılarak arttırılmaktadır. Gerçek bir asenkron motordan elde edilen üç faz akım sinyalleri kullanılarak kırık rotor çubuğu ve stator arızaları belirlenmiştir. Önerilen yöntem yüksüz durumda ve farklı besleme gerilimleri altında kararlı durum akımı ile kırık rotor çubuklarını tespit edebilmektedir.

5. ARIZA TAHMİN VE TEŞHİSİ İÇİN YENİ BİR YÖNTEM

Herhangi bir sistemde oluşan arızanın erken bir aşamada tahmini ve daha sonra elde edilen sonuçlara göre arızanın sebebinin belirlenmesi oldukça önemlidir. Literatürde var olan arıza teşhis ve tahmin yöntemleri iyi bir çözünürlükte alınmış büyük veri kümelerine ihtiyaç duyarlar. Kararlı durum akımını kullanarak kırık rotor çubuğu arızalarının teşhisi için önerilen yöntemler, belirli bir yük sınırında çalışma durumu için iyi sonuç vermesine rağmen düşük veya yüksüz durumda yanlış sonuç verirler. Yüksüz durumda rotor arızalarının teşhisi için geçiş akımının analizi önerilmiştir. Bu sinyali elde etmek için motorun her defasında yeniden başlatılması gerekir. Ayrıca önerilen yöntemler, yüksek çözünürlüklü akım sinyalleri dışında rotor kaymasının hesaplanmasına ihtiyaç duyarlar.

Tezin bu bölümünde önerilen yöntem arıza tahmini ve teşhisi olmak üzere iki aşamadan oluşmaktadır. Önerilen yöntem kırık rotor çubuğu arızalarını yüksüz durumda belirleyebilmektedir. Arıza tahmin aşaması, faz akım sinyalinin belirli sayıdaki periyoduna uygulanan Hilbert dönüşümü ile düzensizliklerin belirlenmesidir. Arızalar iki ardışık pencere arasındaki entropinin hesaplanması ile belirlenir. Bu aşama arıza tespit aşaması olarak ifade edilir. Önerilen yöntem daha öncekilerden farklı olarak kaymanın hesaplanmasına ihtiyaç duymaz. Ayrıca arıza tahmini için oldukça az veriye gereksinim duyulur. Akım sinyalinin on periyotluk kısmının Hilbert dönüşümü, arıza durumunu belirlemek için yeterlidir. İlk adımda hesaplanan iki ardışık pencere arasındaki entropi değeri belirli bir eşik değerini aştığında ikinci aşama aktifleştirilir. İkinci aşamada ise arızanın büyüklüğü, elde edilen özellik sinyalinin spektrumu MAIA ile elde edilerek belirlenir. Önerilen yöntemin diğer bir avantajı oldukça düşük besleme gerilimlerinde iyi bir hassasiyet ile tespit işleminin gerçekleştirilmesidir.

Kararlı durumda kırık rotor çubuğu arızalarının tespiti motor düşük yükte veya yüksüz durumda çalışırken klasik işaret işleme teknikleri ile oldukça zordur. Ana hat frekansı arıza ile ilgili yan bant frekanslarını bastırarak belirsizleştirir. Düşük yük durumunda arıza analizi aşağıdaki faydaları sağlar:

- Yeni motorların kalite kontrolünü geliştirir.
- Bütün motor türlerinin yüksüz analizini sağlar.
- Yük ile indüklenen akım salınımı ile arızanın ayırt edilmesini sağlar.
- Arıza analizinin maliyetini düşürür.

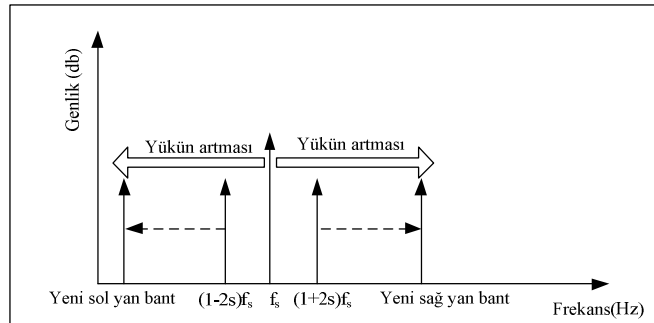
- Üretim esnasında motorun durdurulması ve yeniden başlatılması gerekliliğini ortadan kaldırır.

5.1. Motor Akım İmza Analizinin Eksiklikleri

Bir rotor çubuğu kırıldığı zaman akım geçişi olmaz. Bu sebeple rotorun manyetik alanında bir asimetri oluşur. Sincap kafes asenkron motorlarda bu asimetri stator akımının Fourier dönüşümü ile elde edilen spektrumda bir yan bant frekans bileşeni oluşturur [87]. Hat frekansının sağında ve solunda oluşan bu frekans bileşenlerinin genlikleri incelenerek kırık rotor çubuğu arızalarının belirlenmesi için MAIA geniş bir şekilde kullanılmıştır [21]. Yan bant bileşenlerinden $(1-2s)f_s$ sol yan bant $(1+2s)f_s$ ise sağ yan bant bileşenidir. Sol yan bant bir asenkron motorun rotor kafesinde oluşan elektrik ve manyetik tabanlı asimetriden dolayı oluşurken sağ yan bant ise dönme momentinin titreşimleri ile oluşan hız salınımlarından dolayıdır [64]. İlgili harmonik bileşenleri hesaplamak için kullanılan Fourier dönüşümünün çözünürlüğü denklem 5.1'e göre hesaplanır.

$$\Delta f = \frac{f_{\text{örnekleme}}}{N} = \frac{1}{N T_{\text{örnekleme}}} = \frac{1}{t_{\text{gözlem}}} \quad (5.1)$$

Denklem 5.1'de $f_{\text{örnekleme}}$ örnekleme oranını, $T_{\text{örnekleme}}$ örnekleme periyodunu ve $t_{\text{gözlem}}$ ise izleme aralığını belirtir. Motor sağlam yükte çalışırken MAIA iyi sonuçlar verir. Eğer motorun yükü örnekleme zamanı esnasında düşürülürse yan bant frekanslarının yeri Şekil 5.1'de gösterildiği gibi rotor kaymasına bağlı olarak değişir.



Şekil 5.1. Yükün yan bant bileşenleri üzerindeki etkisi

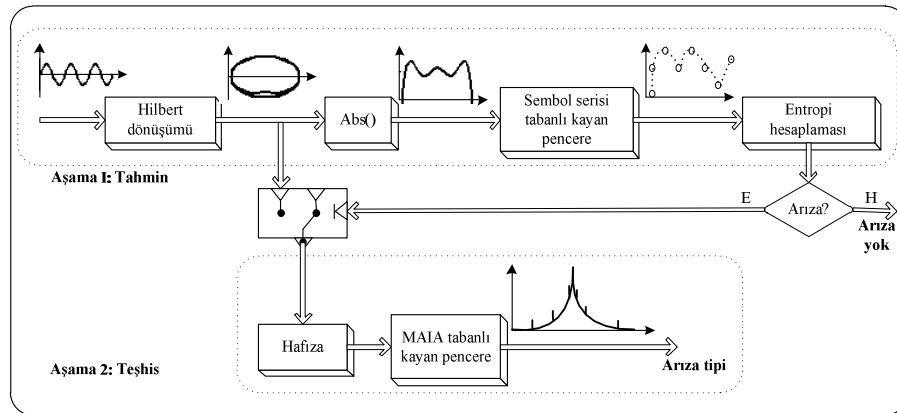
Yan bant frekanslarının değişimi MAIA tabanlı teşhisin yanlış sonuçlar vermesine neden olur. Çünkü hızdaki değişimler ile oluşan yan bant bileşenleri kırık rotor çubuğu ile

ilgili frekansları gizler. Bu problemin üstesinden gelebilmek için Hamming, Hanning ve Tukey gibi pencere fonksiyonları MAIA'nın çözünürlüğünü arttırmak için önerilmiştir [22].

Önerilen MAIA yöntemi yüksüz durumda kırık rotor çubuğu arızalarının büyüklüğünü belirlemek için faz akımının kayan pencere tabanlı Hilbert dönüşümünü kullanır. Motor yüksüz durumda çalıştığında yöntem iyi sonuç verir. Eğer kayan pencere tabanlı entropi hesabı ile bir arıza belirlenmiş ise arızanın detaylı analizi MAIA ile yapılır. Önerilen yöntem birçok avantaja sahiptir: tek bir duyargaya ihtiyaç duyma, uygulanabilme kolaylığı ve geçiş durumları altında analize ihtiyaç duymama.

5.2. Arıza Tahmin ve Teşhisi için Kayan Pencere Tabanlı MAIA Yaklaşımı

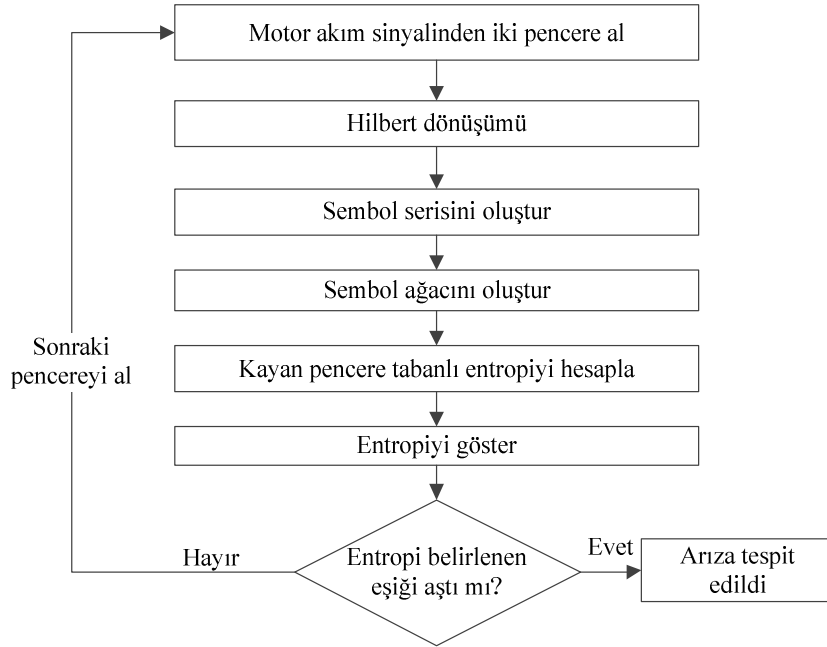
Arıza tahmin ve teşhisi için önerilen yöntem iki aşamadan oluşur. Birinci aşamada belirlenen periyot sayısı için faz akımının Hilbert dönüşümü hesaplanır. Bu aşamada zaman serisi sembol serisine dönüştürülür ve kayan pencere tabanlı bilgi entropisi hesaplanır [137]. Bu entropi zaman serisindeki herhangi bir arızayı tespit etmek için kullanılır. On periyotluk izleme ile kırık rotor çubuğu arızalarını otomatik olarak tespit edebilmek için bilgi entropisi bir eşik değeri ile karşılaştırılmaktadır. Bu değer eşiği aşarsa algoritma bir arıza rapor eder. Aksi durumda motorun sağlam olduğu kanaatine varılır ve izleme devam eder. Arıza belirlendikten sonra teşhis aşaması aktifleşir. İkinci aşamada oluşan arızanın boyutu belirlenecektir. Arızanın boyutu faz akımının Hilbert dönüşümü tabanlı MAIA'i kullanılarak belirlenir. Önerilen iki aşamalı yöntemin şematik gösterimi Şekil 5.2'de verilmiştir.



Şekil 5.2. Arıza teşhis yönteminin şematik gösterimi

5.2.1. Aşama I: Arıza Tahmini için Kayan Pencere Tabanlı Sembol Serisi

Arıza tahmin aşamasında tek faz akımının on periyotluk kısmı alınır ve bu sinyalin Hilbert dönüşümü hesaplanır. Daha sonra elde edilen özellik sinyalinin sembol serisi üretilir. Sembol serisinden sembol ağacı oluşturularak entropi hesaplanır. Kayan pencere tabanlı entropi motor durumunu izlemek için bir arıza indeksi olarak kullanılır. Sonraki aşamada geçerli pencere bir örnek kaydırılır ve yeni pencere elde edilerek aynı adımlar bu pencere için de tekrarlanır. Bu algoritma Şekil 5.3'te verilmiştir.



Şekil 5.3. Kayan pencere tabanlı online arıza tespit algoritması

Hilbert dönüşümü akım sinyalinin belirli periyotlarından oluşan bir seriye uygulanmaktadır. Bu seri pencere olarak alınır. Her pencere farklı sayıda periyottan oluşabilir. Pencere boyutu aşağıdaki gibi hesaplanmaktadır.

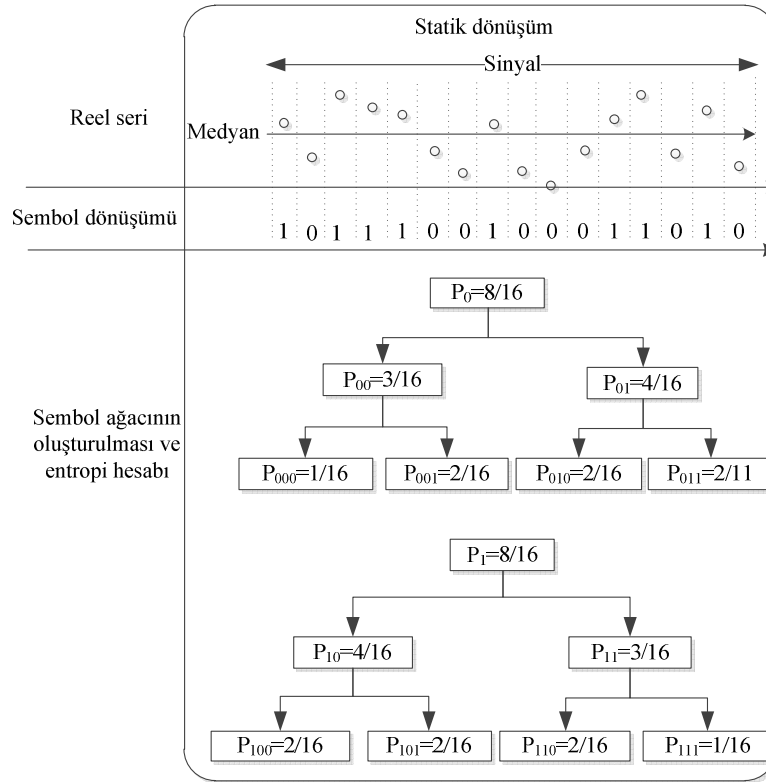
$$w = f_{ornekleme} * \frac{1}{f_s} * n \quad (5.2)$$

Denklem 5.2'de $f_{ornekleme}$ akım sinyalinin örnekleme oranını, f_s hat frekansını (50 Hz) ve n ise periyot sayısını gösterir. Özellik sinyali elde edildikten sonra sembol serisi oluşturulur. Özellik sinyalinin sembol serisi statik dönüşüme dayalı olup bu dönüşümde

verilen bir örneğe alınan pencerenin medyanına göre sıfır veya bir atanır [138]. Bir $H(t)$ özellik sinyali için statik dönüşüm aşağıdaki gibi hesaplanır.

$$SS(t) = \begin{cases} 1, & H(t) > \text{median}(H(t)) \\ 0, & H(t) \leq \text{median}(H(t)) \end{cases}, t = 1, \dots, n \quad (5.3)$$

Her bir pencere sembol serisi aracılığıyla ayrıştırıldıktan sonra ilgili sembol ağacı oluşturulur. Oluşturulan sembol ağacı sembollerin bütün olanaklı durumlarını gösteren paralel dallara sahiptir [139]. Sembol serisi ve ağacı oluşturma süreci Şekil 5.4'te verilmiştir.



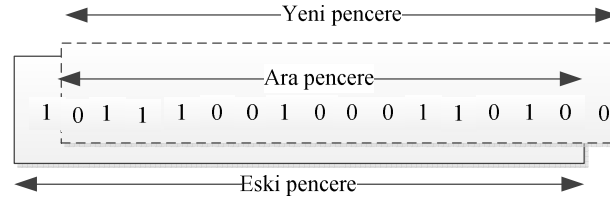
Şekil 5.4. Reel verinin sembolleştirilmesi ve sembol ağacının oluşturulması

Şekil 5.4'te P bir sembol serisinin olasılığını gösterir. P_{00} "00" sembolünün olasılığını ifade eder. Sembol serisindeki "00" sembolünün sayısı üçtür. Sembol serisinin uzunluğu 16 olduğu için P_{00} 3/16 olarak hesaplanır. Ağacın derinliği sembol serisinin uzunluğu ile belirlenir. Ağacın bilgi entropisi denklem 5.4'e göre hesaplanır [140].

$$H_s = -\frac{1}{\log n} \sum_{m=1}^n P_m \log P_m \quad (5.4)$$

Denklem 5.4'te n sembol serisinin uzunluğunu P_m ise olasılığı sıfırdan farklı olan sembollerin sayısını gösterir. H_s ağacın bilgi entropisini gösterir ve değeri sıfır ile bir aralığındadır.

Arızalar çıkarılan özelliğe sembol serisi tabanlı kayan pencere yöntemi uygulanarak belirlenir. Kayan pencerenin karmaşıklığı pencere boyutu ve sembol ağacının seviyesine bağlıdır [141]. Bir kayan pencere örneği Şekil 5.5'te verilmiştir.



Şekil 5.5. Kayan pencerenin gösterimi

Yeni pencerenin bilgi entropisi denklem 5.5'te verilmiştir.

$$H_s = -\frac{1}{\log n} \left(\sum_{i=1}^L \left(\sum_{meski} \frac{P_{meski} - 1}{w} \log \frac{P_{meski} - 1}{w} + \sum_{myeni} \frac{P_{myeni} + 1}{w} \log \frac{P_{myeni} + 1}{w} \right) \right) + \sum_{mara} P_{mara} \log P_{mara} \quad (5.5)$$

Yukarıdaki denklemde n sembol serisinin uzunluğunu, L ağacın seviyesini ve w ise pencere boyutunu gösterir. Yeni, ara ve eski pencerelerin olasılığı ise sırasıyla P_{myeni} , P_{mara} ve P_{meski} ile ifade edilir. Kayan pencereler arasındaki bilgi entropisi belirlenen bir eşik değeri aştığında bir arıza rapor edilir. Eğer sistem bir arızaya sahip ise incelenen pencere arızanın boyutunu belirlemek için hafızaya kaydedilir. Bu durum aynı zamanda teşhis bölümünü aktifleştirir. Herhangi bir arıza durumunda geniş periyodik hareketler olduğundan bilgi entropisi de değişir. Geniş periyodik hareketler sembol serisini dolayısıyla entropiyi değiştirir. Bu özellik kırık rotor çubuklarının tespitinde çok etkilidir.

5.2.2. Aşama II: MAIA Tabanlı Kayan Pencere ile Teşhis

Arıza teşhis bölümü Hilbert dönüşümü kullanılarak analiz edilecek analitik sinyalin çıkarılmasına dayalıdır. Bu bölüm tespit aşamasında bir alarm üretildiğinde aktifleşir. Arızanın boyutu ve tipi bu aşamada belirlenir. Arıza teşhisinde kullanılan özellik sinyali aşağıdaki gibi tanımlanır.

$$I_h = |H(I(t)) - \text{mean}(|H(I(t))|)| \quad (5.6)$$

Bu denklemde $| |$ modülü hesaplamak için kullanılır. $H(I(t))$ faz akımının Hilbert dönüşümünü tanımlar. Bu sinyal analitik sinyal ve onun ortalamasının farkıdır. Hilbert dönüşümünün spektrumu $2sf_h$ frekansında bir tepe değer oluşturur [131]. Bu dönüşüm aşağıda verilmiştir.

$$I(t) = A \sin(\omega t) + B \sin((\omega + 2s\omega)t) \quad (5.7)$$

Denklem 5.7’de ω makinenin dönme frekansını gösterir ve değeri $2\pi f$ ’tir. Denklemdeki s değeri kaymayı ve f_h değeri ise Hertz olarak frekansı belirtir. Denklem 5.7’de verilen analitik sinyal aşağıdaki gibi tekrar yazılabilir [141].

$$AS(t) = I(t) + jH(I(t)) = -A \cos(\omega t) - B \cos((\omega + 2s\omega)t) \quad (5.7)$$

Denklemde AS(t) aşağıdaki gibi yazılır.

$$AS(t) = -jAe^{j\omega t} - jBe^{j2s\omega t}e^{j\omega t} = -je^{j\omega t}(A + Be^{j2s\omega t}) \quad (5.8)$$

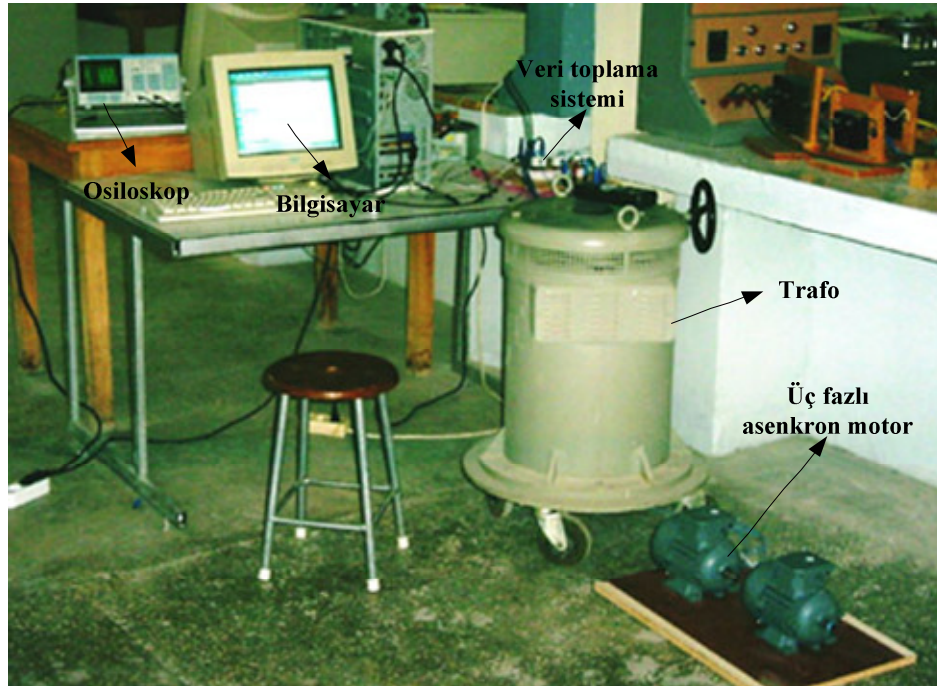
Orijinal MAIA’deki iki yan bant yerine sadece arıza ile ilgili tek yan bant bileşeni $2sf_h$ elde edilmektedir. Böylece aşağıdaki kazanımlar sağlanır:

- MAIA’ni tek faz akımı üzerinden uygulamak ve tek bir yan bant ile teşhis önemlidir.
- Hilbert dönüşümü sayesinde standart MAIA’de oluşan frekans kayıpları oluşmaz.
- Arıza ile ilgili frekans düşük olduğundan özellik sinyalinin örnekleme oranı spektrumu tamamını incelemeyi gerektirmez. Bu özellik kullanılacak veri sayısını azaltır.

Önerilen yöntem önemli yenilikler sunar. İlk adımda kayan pencere tabanlı arıza tespiti arıza olup olmadığını belirler. Eğer incelenen pencerede arıza tespit edilirse hafızaya kaydedilerek MAIA yöntemi ile daha detaylı incelenerek arızanın şiddeti tahmin edilir.

5.2.3. Deneysel Sonuçlar

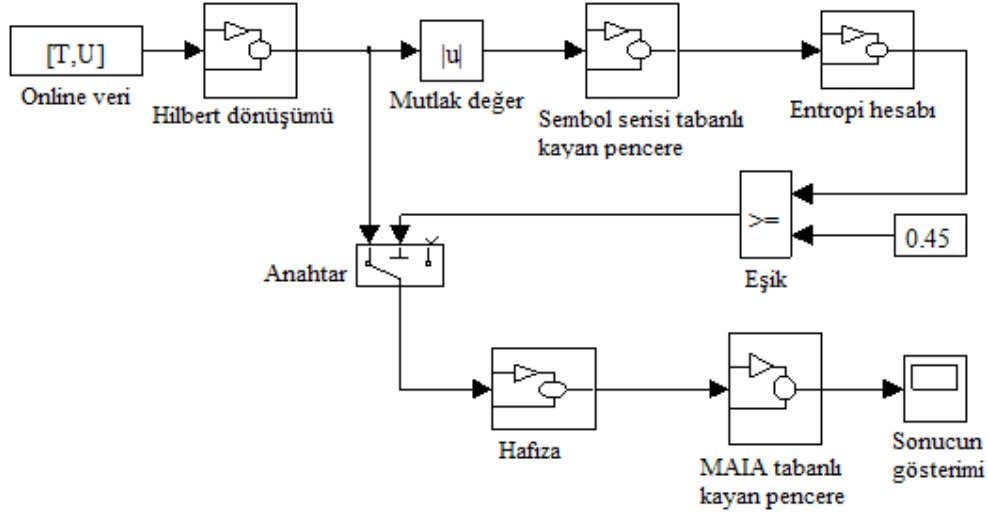
Önerilen yöntemin performansının değerlendirilmesi ve doğruluğunun ispatı için 0.37 kW'lık asenkron motordan elde edilen akım sinyalleri kullanılmıştır. Testler bir ve iki kırk rotor çubuğu arızası ile sağlam motor durumu için gerçekleştirilmiştir. Rotor çubukları matkap ile delinerek akım geçirmeleri engellenmiştir. Her iki durumda da motor shaftı boş bırakılarak yüksüz durumda akım sinyalleri elde edilmiştir. Ayrıca motorun besleme gerilimleri bir trafo aracılığıyla değiştirilerek beş farklı durum için ek testler uygulanmıştır. Besleme gerilimleri her bir durum için 380, 340, 300, 260 ve 220 V olarak değiştirilmiştir. Yedi ardışık ölçümün ortalaması alınarak arıza tespit sisteminin gürültü seviyesi azaltılmıştır. Akım sinyalleri veri toplama kartı üzerinden bilgisayara 3000 örnek/saniye örnekleme oranı ile 3 saniye boyunca kaydedilmektedir. Ölçülen bu akım değerleri 16 bitlik Advantech PCL-816 model bir veri toplama kartı aracılığıyla bilgisayara aktarılmıştır. Kurulan deney düzeneği Şekil 5.6'da verilmiştir.



Şekil 5.6. Deney düzeneği

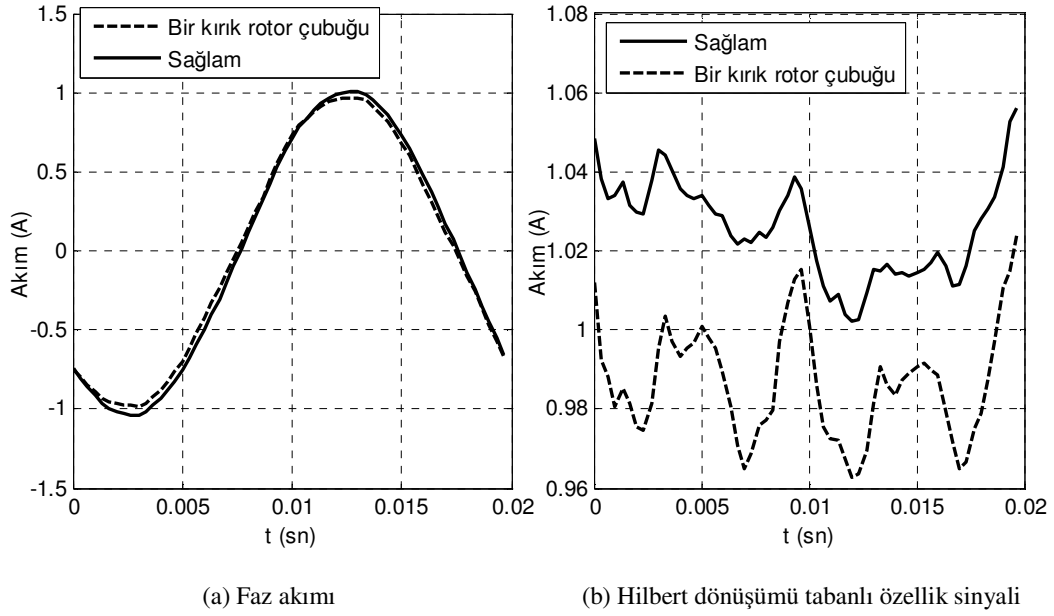
Yöntemin uygulaması MATLAB ortamında yapılmıştır [142]. Simülasyon her besleme gerilimi için bir ve iki kırk rotor çubuğu göz önüne alınarak MATLAB-

SIMULINK ortamında gerçekleştirilmiştir. Arıza teşhis ve tespit sisteminin SIMULINK modeli Şekil 5.7’de verilmiştir.



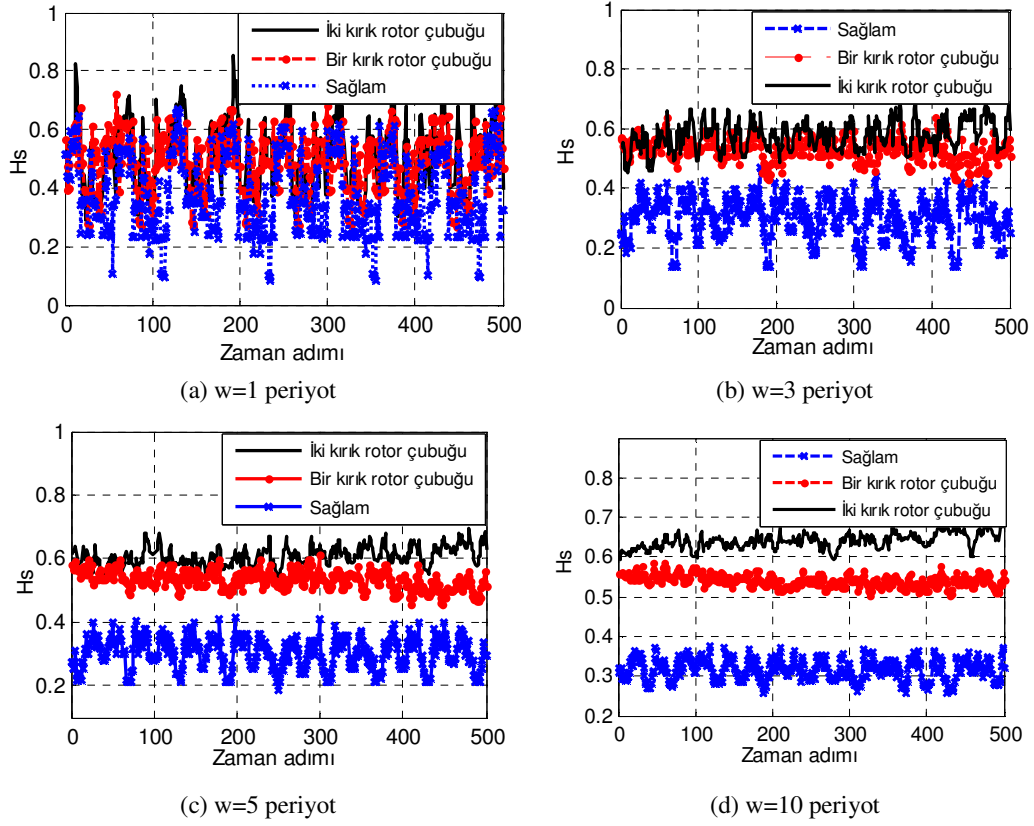
Şekil 5.7. MATLAB/SIMULINK modeli

Şekil 5.7’deki kayan pencere denklem 5.2’deki faz akımının periyotları kullanılarak oluşturulmaktadır. Örnekleme oranı ($f_{\text{örnekleme}}$) 3000 örnek/saniye ve besleme frekansı (f_h) 50 Hz’dir. Böylece denklemden pencere boyutu bir periyot için 60 örnek olarak bulunur. Motor durumunun gerçek zamanlı izlenmesinin gerçekleştirilmesi için tespit aşaması kayan pencere tabanlı entropi hesaplamasından faydalanır. İki pencere arasında hesaplanan entropi değeri bir eşik değeri ile karşılaştırılır ve eğer bu değeri belirlenen eşik değeri aşarsa ilgili pencere hafızaya kaydedilerek MAIA tabanlı teşhis bölümü aktifleştirilir. Bilgi entropisi sağlam motor için 0.45 değerinden düşük çıktığından dolayı eşik değeri 0.45 seçilmiştir. Sağlam ve bir kırık rotor çubuğu arızası için tek faz akım sinyali ve bu sinyallerin Hilbert dönüşümleri ile elde edilen özellik sinyali Şekil 5.8’de verilmiştir.



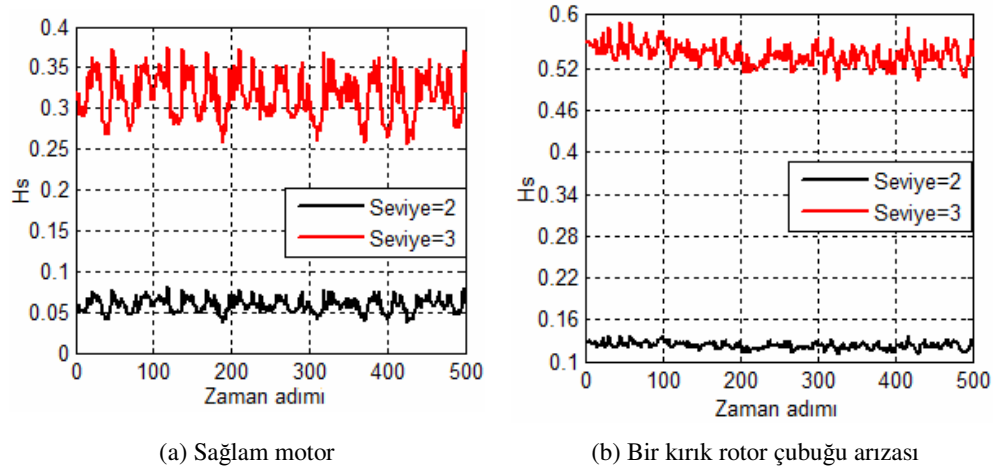
Şekil 5.8. Bir periyotluk faz akımı ve elde edilen özellik sinyali

Şekil 5.8 (a)'da görüldüğü gibi sağlam motor ve arızalı motor faz akım sinyalleri arasındaki fark çok küçüktür. Bu yüzden akım sinyallerinin zaman domeninde incelenmesi ile rotor arızalarının belirlenmesi zordur. Şekil 5.8 (b)'de Hilbert dönüşümüne dayalı çıkarılan özellik sinyalinin arızalı ve sağlam durumu ayırt etmede oldukça uygun olduğu görülmektedir. Bilgi entropisi ile elde edilen özellik sinyalindeki periyodik değişimler incelenebilir. Kayan pencerenin performansı pencere boyutu ve sembol ağacının seviyesi ile direkt ilişkilidir. Bilgi entropisi farklı pencere boyutları ve sembol ağacı seviyesine göre değerlendirilmiştir. Şekil 5.9'da farklı pencere boyutları için her bir durumun entropi değeri gösterilmiştir.



Şekil 5.9. Farklı pencere boyutları için bilgi entropisi

Şekil 5.9’da pencere boyutunun küçük seçilmesi yöntemin performansını etkilemektedir. Pencere boyutu beşten küçük seçildiğinde arızalı ve sağlam durumların ayırt edilmesi zorlaşır. Sembol ağacının seviyesi ise üç seçilmiştir. Periyot sayısı 10 alındığında sağlam ve arızalı durumlar arasındaki farkın da arttığı görülmektedir. Bilgi entropisi belirlenen eşik değeri aştığında tespit aşaması bir alarm üretir ve teşhis aşaması aktifleşir. Aksi durumda motor durumunun normal olduğu bildirilir. Ayrıca sembol ağacının seviyesi de arıza teşhis ve tespit sisteminin hem doğruluğunu hem de karmaşıklığını etkiler. Şekil 5.10’da sağlam ve arızalı durumlar için farklı ağaç seviyelerine göre bilgi entropisi sonuçları gösterilmiştir.



Şekil 5.10. Sembol ağacının farklı seviyesine göre bilgi entropisi

Sağlam ve arızalı durumların ayırt edilmesi sembol ağacının seviyesine duyarlıdır. Sembol ağacının seviyesi iki seçilirse sağlam ve bir kırık rotor çubuğu arızası için bilgi entropilerinin farkı çok küçük olur. Bu değer üç seçilirse fark iki durum ayırt edebilecek kadar büyük olacaktır. Önerilen yöntemin performansı aynı zamanda 220, 260, 300 ve 340 V gibi düşük besleme gerilimleri için de test edilmiştir. Her bir motor durumu için bilgi entropisinin medyanı ayırt edici özellik olarak alınmıştır. Tablo 5.1’de bu durumlar için bilgi entropileri verilmektedir.

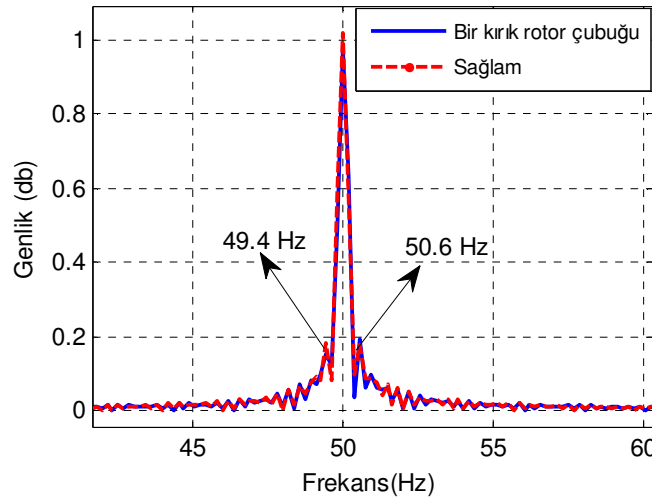
Tablo 5.1. Farklı besleme gerilimleri için bilgi entropilerinin medyanı

Besleme gerilimi (V)	Bilgi Entropilerinin Medyanı (Hs)		
	Sağlam	Bir kırık rotor çubuğu	İki kırık rotor çubuğu
380	0.35	0.54	0.64
340	0.33	0.56	0.66
300	0.37	0.50	0.66
260	0.51	0.56	0.67
220	0.49	0.51	0.38

Tablo 5.1’de görüldüğü gibi besleme gerilimi metodun performansını etkiler. Besleme gerilimleri 380, 340 ve 300 V seçildiğinde her bir sağlam ve arızalı durumun besleme entropileri yaklaşık olarak aynı seviyededir. Fakat besleme gerilimlerinin 300 V’tan düşük olduğu durumlarda sağlam ve arızalı medyan değerleri birbirine çok yakın olur.

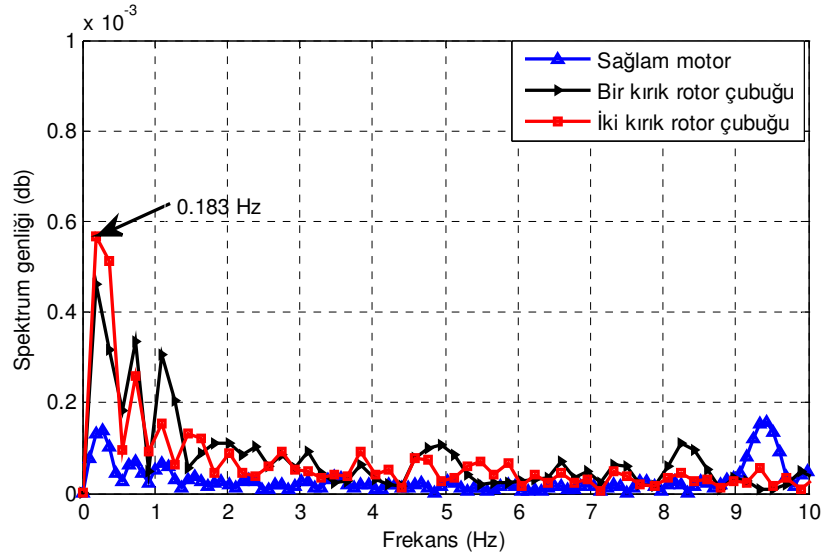
Arıza tespit aşamasında bir arıza tespit edilirse özellik sinyali arızanın tipini ve boyutunu belirlemek için kaydedilir ve teşhis aşaması aktifleşir. Bu adım kayan

pencerelere MAIA'nin uygulanmasını gerektirir. Standart MAIA direkt olarak faz akımlarına uygulanmaktadır. Motor düşük yük veya yüksüz durumda çalışırken yan bant frekansları hat frekansına yakın olur. Önerilen yöntemin arıza teşhis aşamasında ise Hilbert dönüşümü ile elde edilen özellik sinyaline MAIA uygulanmaktadır. Sağlam ve bir kırık rotor çubuğu arızası için akım sinyali üzerinde gerçekleştirilen MAIA sonuçları 380 V besleme gerilimi için Şekil 5.11'de verilmiştir.



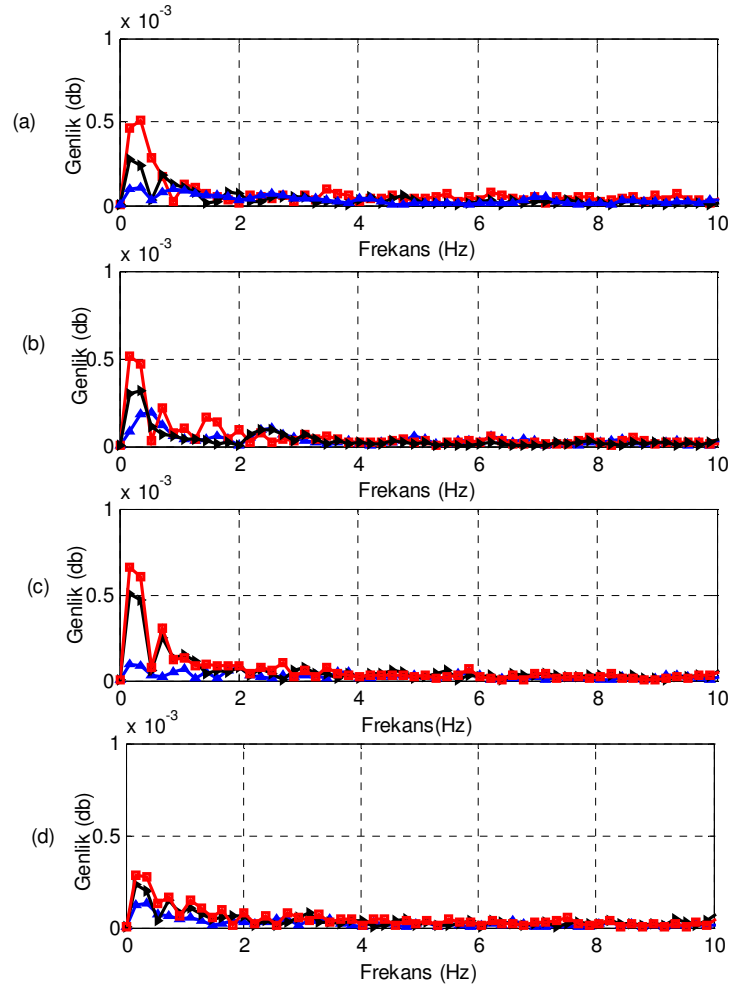
Şekil 5.11. Faz akımı üzerinde uygulanan MAIA

Şekil 5.11'de gösterildiği gibi rotor çubuğu arızaları ile ilgili yan bant frekansları hat frekansına çok yakın olduğundan teşhis zorlaşır. Şekil 5.12'de ise Hilbert dönüşümü ile kayan pencerelerden elde edilen özellik sinyali üzerinde MAIA'nin üç durum için uygulaması verilmiştir.



Şekil 5.12. Hilbert dönüşümü tabanlı kayan pencerelerin spektrumu

Şekil 5.12’de sağlam ve arızalı motorlar için kayma değeri 0.002 ve 0.0019 olarak alınmıştır. Motor akım sinyallerinin örnekleme oranı 3 kHz’dir. Akım sinyalleri 3 saniye kaydedildiğinden her bir durum için 9 000 örnek alınmaktadır. Spektrumu elde etmek için 16384 noktalı FFT uygulanmıştır. Standart MAIA kullanıldığında yüksüz durumda yan bant frekansları ile arızalı durumlar ve sağlam durum ayırt edilememektedir. Önerilen yöntemde arıza ile ilgili frekans $2sf_h$ ile ifade edilip bu frekansın yeri 0.183 Hz olarak bulunmuştur. Orijinal MAIA’de iki yan bant bileşeni incelenirken önerilen yöntem bir adet yan bant bileşeni ile arıza analizi yapar. Önerilen yöntemin dört farklı besleme gerilimi altında sağlam durum ile bir ve iki kırık rotor çubuğu arızası için uygulama sonuçları Şekil 5.13’te verilmiştir.



Şekil 5.13. Dört besleme gerilimi için kayan pencere tabanlı Hilbert dönüşümünün spektrumu (a) 340 V (b) 300 V (c) 260 V (d) 220 V

Önerilen yöntem besleme gerilimi 260 V ve üzeri olduğunda iyi sonuçlar vermekte 220 V besleme geriliminde ise sağlam ve arızalı durumların ayırt edilmesi zorlaşmaktadır. Belirli bir yük durumu altında kırık rotor çubuğu arızalarının tespiti için literatürde birçok yöntem önerilmiştir. Bu yöntemlerin ortak dezavantajı yük düşürüldüğünde yöntemlerin performansı da düşmektedir. Tablo 5.2’te önerilen yöntemin literatürdeki çalışmalar açısından yeri verilmiştir.

Tablo 5.2. Yöntemin literatürdeki yeri

Çalışma	Sinyal	Yöntem	Çözünürlük	Yük seviyesi (%)	
				Min	Maks
[22]	Akım	MAIA	10 kHz-1 s	25	100
[38]	Akım	Analitik sinyalin karmaşık dalgacık analizi	1 kHz-10 s	10	100
[39]	Geçiş akımı	Faz akımının dalgacık analizi	8kHz-1 s	0	100
[40]	Akım	Hilbert dönüşümü tabanlı dalgacık analizi	10 kHz- 1 s	-	100
[131]	Akım	Hilbert dönüşümünün MAIA	10kHz-100 s	0	100
Önerilen yöntem	Akım	Kayan pencere tabanlı MAIA	Arıza tespiti- 600 örnek Teşhis- 3 kHz -3 s	0- Düşük besleme gerilimleri	-

Tablo 5.2 kırık rotor çubuğu arızalarının tespitinde yükün ne kadar önemli parametre olduğunu göstermektedir. Bazı araştırmacılar kırık rotor çubuğu arızaların tespitinde yükün etkisini ortadan kaldırmak için geçici durum akımını kullanmışlardır. Bununla birlikte bu yöntemlerin pratik uygulaması oldukça zordur. Çünkü arıza teşhisi için motorun her defasında yeniden başlatılması gerekir. Büyük endüstriyel sistemlerde bu istenen bir durum değildir. Kararlı durumda kırık rotor çubuğu arızalarını analiz eden yöntemler belirli yüklerde iyi sonuç verirler. Yük ne kadar azaltılırsa doğruluk oranı da azalır. Diğer bir dezavantaj ise bu yöntemlerin çok büyük verilerin depolanmasına ihtiyaç duymasıdır.

5.3. Bölüm Değerlendirmesi ve Sonuçlar

Tezin bu bölümünde kırık rotor çubuğu arızalarının tahmini ve teşhisi için yeni bir yöntem sunulmuştur. Tek faz akım sinyalinin belirlenen sayıda periyoduna uygulanan Hilbert dönüşümünün bilgi entropisi arıza tahmini için kullanılmıştır. Entropi değeri ardışık iki kayan pencere arasında hesaplanmaktadır. Bilgi entropisi belirlenen bir eşik değeri aştığında geniş ölçekli periyodik bir hareket oluşumu tespit edilir ve tahmin algoritması arızayı belirler. Arıza belirlendikten sonra teşhis aşaması aktifleştirilerek arızanın teşhisi sağlanmaktadır. Yöntem yüksüz ve düşük besleme gerilimleri altında kırık rotor çubuğu arızalarını tespit edebilmektedir. Önerilen yöntem 260 V besleme geriliminin üstünde doğru sonuçlar vermektedir. Yöntemin en önemli avantajı diğer var olan arıza tespit ve teşhis yöntemlerine göre daha az hesaplama maliyetinin olmasıdır. Ayrıca rotor

ubuęu kırıldıktan sonra on periyot akım sinyali kullanılarak arıza belirlenebilmektedir. Önerilen yöntem arıza tespiti ve teşhisi için yalnızca tek faz akım sinyaline ihtiyaç duyar. Önerilen yöntemin avantajları aşağıdaki gibi sıralanabilir:

- Arıza tahmini için oldukça az sinyal bilgisi gerektirmesi,
- Düşük besleme gerilimleri altında kırık rotor ubuęu arızalarının teşhisi,
- Yöntemin çevrimiçi durum deęerlendirmesi için oldukça uygun olması.

6. GERÇEK ZAMANLI ARIZA TESPİT, TEŞHİS VE TAHMİNİ

FPGA ve DSP gibi donanımsal işaret işleme birimlerinin gelişimi ile durum izleme ve arıza teşhisi için düşük maliyetli, taşınabilir ve gömülü durum izleme donanımları geliştirilebilir [143]. Bu tür donanımların sağladığı en önemli avantaj yerinde izlemenin gerçekleştirilebilmesidir. Dolayısıyla veri toplama kartı, bilgisayar ve diğer bağlantı elemanlarına ihtiyaç duyulmadan güvenilir ve taşınabilir basit durum izleme ekipmanları geliştirilebilmektedir. Ayrıca FPGA kartlarının yeniden programlanabilme özellikleri sayesinde, farklı arıza durumları veya izlenen sistemin parametrelerine göre tasarım yeniden uyarlanabilmektedir.

Tezin bu bölümünde ilk olarak FPGA mimarisi ve programlanması ile ilgili temel bilgiler verilecektir. FPGA'nın programlanması için kullanılan VHDL donanım tanımlama dilinden bahsedilerek FPGA'da basit bir uygulamanın nasıl gerçekleştirileceği adım adım anlatılacaktır. FPGA üzerinde gerçekleştirilen arıza teşhis, tespit ve tahmin algoritmalarının prensipleri anlatılarak önerilen yöntemler deneysel sonuçlar ile doğrulanacaktır. İlk yöntem stator arızalarının teşhisi için üç faz motor akım sinyalinin kullanılmaktadır. Faz akım sinyallerinin büyüklükleri bulanık mantık ile değerlendirilerek motor durumu belirlenecektir. İkinci yöntem herhangi bir arızanın anlık tespiti için yapay sinir ağından faydalanmaktadır. Tek faz akım üzerinde uygulanan Hilbert dönüşüm sinyalinin ortalaması ve standart sapması yapay sinir ağı tabanlı tahmin algoritması ile değerlendirilmekte ve arızalar tahmin edilmektedir. Üçüncü yöntem rotor çubuğu arızalarının teşhisi için kaskad yapay sinir ağından faydalanmaktadır. Yapay sinir ağının eğitimi için PSO algoritması önerilmiştir. Önerilen yöntem minimum sayıda nöron ile doğrusal olmayan verileri doğru şekilde sınıflandırmaktadır. Dördüncü yöntem ise rotor çubuğu arızalarının tespiti için negatif seçim algoritmasından faydalanır. Tek faz akım sinyalinin kaydırılması ile oluşturulan özellik sinyalinin faz uzayı sağlam ve arızalı durumları ayırt edebilmektedir. Negatif seçim algoritmasının detektörleri üçüncü bölümde anlatılan genetik algoritma ile optimize edilmiştir.

6.1. FPGA

Programlanabilir devrelerin geçmişi, 1970'lerde programlanabilir lojik dizilerin (PLAs) üretilmesine dayanır [144]. Bu devrelerin kullanımı teknolojik sebeplerden dolayı

oldukça sınırlıydı. 1980’lerde ise hızlı tasarım, yüksek yapılandırılabilirlik ve değişiklik özelliklerine sahip programlanabilir dizi mantık aygıtları (PALs) geliştirildi. Fakat bu aygıtlar ile karmaşık tasarımlar gerçekleştirilemeyip güç tüketimi ile ilgili problemler mevcuttu. Özel lojik bloklar dizisine dayalı kapı dizileri tekniğinin gelişimi ile birlikte 1984’te Xilinx firması tarafından ilk sahada programlanabilir kapı dizileri (FPGA) geliştirildi. İlk geliştirilen FPGA tasarımı 85 000 transistörden oluşmaktaydı [145]. Aynı dönemde uygulamaya özgü tümleşik devreler (ASICs) özel uygulamalar için tasarlandı. Karmaşık yapıları ve geniş kullanım olanaklarına rağmen, yeniden programlanamama ve pahalı bir teknoloji olmalarından dolayı ASICs devreleri kullanışlı değildi.

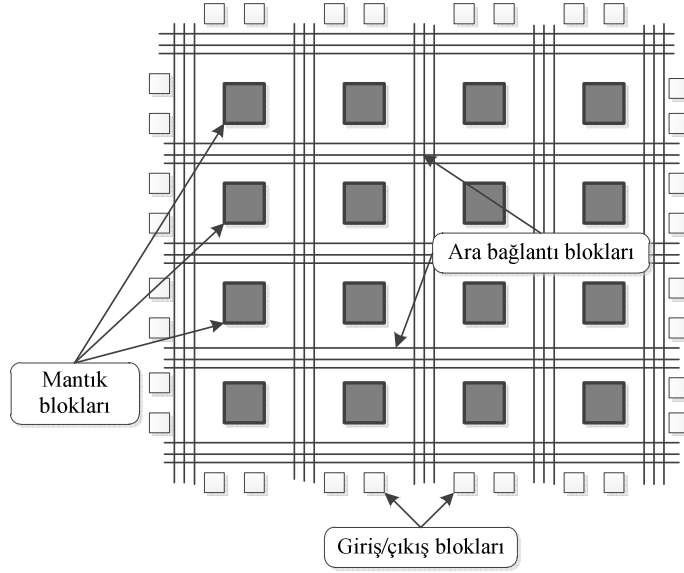
İlk FPGA’lar CMOS tabanlı olmasına rağmen daha sonra geliştirilenler SRAM’e dayalıydı. FPGA teknolojisinin popüler olmasının temelinde maliyet açısından yapılan tasarımlar etkili olmuştur. 2000’li yıllarda FPGA’nın temel lojik blokları olan bakma tabloları (LUT) ve flip-flop’lar oldukça düşük bir maliyet ile gerçekleştirilebildi. Dolayısıyla milyonlarca kapı elemanı içeren yüksek hızlı FPGA donanımlarının geliştirilmesine başlandı. Günümüzde geliştirilen FPGA kartları gömülü mikroişlemci çekirdekleri, yüksek hızlı giriş/çıkış arabirimleri ve DSP’ye sahiptir. Bu özellikleri sayesinde FPGA kartları robotik, otomotiv endüstrisi, sinyal ve görüntü işleme, uzay araçları ve uçakların gömülü kontrolü için oldukça geniş bir şekilde kullanılmaktadır [9].

6.1.1. FPGA Mimarisi

FPGA geniş bir programlanabilir lojik bileşenler familyasına sahiptir. Bir FPGA donanımı temel olarak yeniden ayarlanabilen lojik bloklar (CLBs) matrisi olarak tanımlanır [146]. Lojik blokların her biri diğerine yeniden programlanabilir bir ara bağlantı ağı ile bağlanmıştır. Genel olarak bir FPGA üç temel birimden oluşmaktadır:

- G/Ç blokları,
- Ara bağlantı blokları,
- Mantık blokları.

Bir FPGA farklı tasarımları oluşturabilecek şekilde bağlanabilen mantık bloklar dizisinden oluşur. Mevcut FPGA kartları transistör çiftleri, temel basit kapılar, çoklayıcılar ve LUT gibi mantık bloklarından biri kullanılarak oluşturulur [147]. FPGA kartının yeniden programlanması statik RAM (SRAM), antifuse veya Flash temelli yapılmaktadır. Şekil 6.1’de SRAM tabanlı FPGA’nın genel yapısı gösterilmiştir.

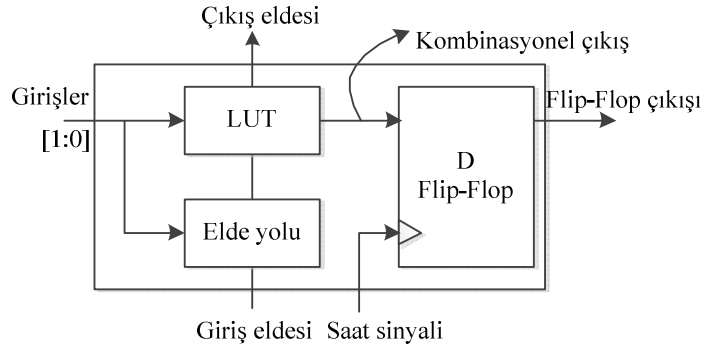


Şekil 6.1. Bir FPGA'nın genel yapısı

Şekil 6.1'de gösterilen giriş/çıkış blokları, mantık blokları ve ara bağlantı blokları bir FPGA kartında bulunan en temel bileşenlerdir. Günümüzde kullanılan FPGA kartları bu bileşenlerin dışında hafıza blokları, fiziksel bağlantılı DSP'ler, saat yönetim blokları ve iletişim bloklarını içermektedir [146]. Uygulanacak fonksiyona bağlı olarak her lojik blok kombinasyonel veya ardışık işlemleri uygulamak için uyarlanabilir. Kombinasyonel işlemler için LUT'lar kullanılır. Bu işlem, ardışık işlemler için flip-flop kullanımına benzer. Bir mantık bloğu hafıza, toplama/çıkarma, çoklayıcı ve kayan yazmaç işlemlerini gerçekleştirmek için de kullanılır. Ara bağlantı ağı gerekli mantık bloklarını birbirine bağlamak için kullanıcı tarafından programlanır. FPGA kaynaklarını optimum şekilde kullanabilmek için toplayıcı, çarpıcı ve diğer aritmetik işlemleri içeren fiziksel bağlantılı DSP blokları mevcuttur. Aynı amaç için bütünleşik RAM, ROM ve Flash ROM gibi hafıza blokları da bütünleşik olarak bulunmaktadır [148]. Bütünleşik saat yönetim bloğu faz kilitli döngülere (PLLs) dayalı olup saat kaynaklarının yönetimini sağlar. Saat yönetim bloğu, saat frekansını bölme, çarpma, faz kaydırma ve yayılma gecikmesi gibi işlemleri yönetir. FPGA donanımları aynı zamanda USB, PCI, Ethernet gibi protokolleri de desteklemektedir.

FPGA donanımlarını diğerlerinden ayıran en önemli özellik programlanabilir ara bağlantılar içerisinde gömülü olarak bulunan küçük programlanabilir lojik bloklardır [149]. Lojik blokları oluşturan hücreler lojik hücre (LC) veya lojik eleman (LE) olarak

isimlendirilir. Birim hücre mimarisine göre MUX veya LUT tabanlı FPGA'lar üretilmiştir. Karmaşık donanımları tasarlamak amacıyla LUT tabanlı mimari daha uygundur. LUT tabanlı mimaride LUT oluşturmak için esnek hafıza kullanım imkanı veren SRAM kullanılır. İki büyük FPGA üreticisi Xilinx ve Altera LUT tabanlı hücre mimarisini kullanmaktadır. Yapılandırılabilir lojik bloklar (CLBs), lojik elemanlar olarak isimlendirilen iki, dört veya daha fazla lojik hücreden oluşur. Bir lojik hücrenin temel yapısı Şekil 6.2'de verilmiştir.

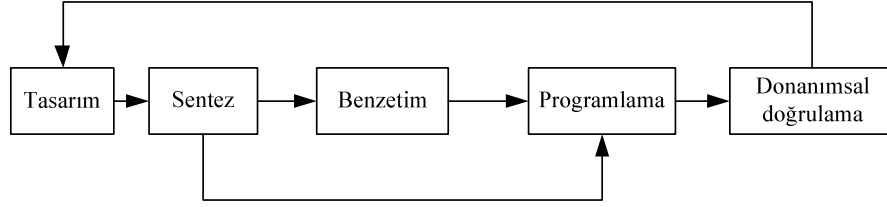


Şekil 6.2. Lojik hücre yapısı

Şekil 6.2'de verilen mantık hücresi iki bitlik bir LUT'a sahiptir. Bu LUT 4X1'lik ROM, RAM veya kombinyasyonel bir fonksiyon olarak ayarlanabilir. Elde veri yolu ise aritmetik işlemleri gerçekleştirmek için eklenmiştir.

6.1.2. FPGA'da Uygulama Geliştirme Aşamaları

FPGA uygulamalarının geliştirilmesi bilgisayar destekli tasarım araçlarının (CAD) gelişmesi ile daha kolay hale gelmiştir. Günümüzde kullanılan FPGA kartları oldukça karmaşık fonksiyonları gerçekleştirebilmektedir. Tek bir FPGA kartı üzerinde aritmetik lojik ünite, hafızalar ve iletişim birimleri gibi tam bir sayısal sistemin geliştirilmesi oldukça kolaydır. FPGA kartlarındaki bu gelişim VLSI (Very Large Scale Integration) devrelerin tasarımındaki gelişmelere paralel olarak uygun tasarım araçları ve yöntemlerin gelişimi de etkili olmuştur. Bu araçlar genellikle VHDL (Very High Speed Integrated Circuits Description Language) ve Verilog gibi donanım tanımlama dillerine dayalıdır [149]. Bu dillerin IEEE (Institute of Electrical and Electronical Engineers) standartlarına göre geliştirilmesi mikro elektronik alanında yeni CAD araçlarının gelişimini sağlamıştır. Şekil 6.3'te FPGA tasarım sürecinin blok diyagramı gösterilmiştir.



Şekil 6.3. FPGA kullanılarak yapılan tasarımın blok diyagramı

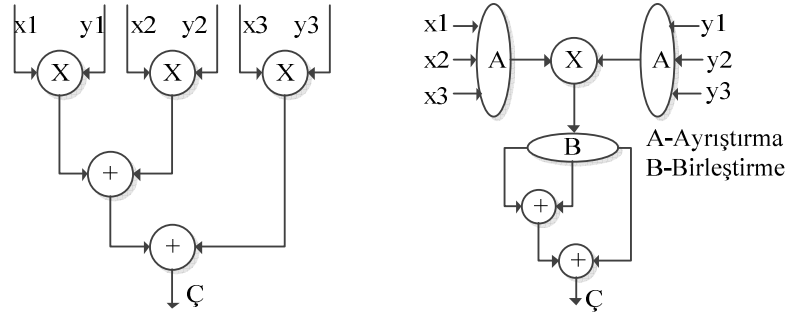
Tasarım aşamasında; istenen devre şematik veya blok çizimler ile gerçekleştirilir. Bu aşamada VHDL veya Verilog gibi donanım tanımlama dillerini kullanmak esnek ve hızlı tasarımı sağlar.

Sentez aşamasında; devrenin FPGA üzerinde gerçekleştirilmesi için gerekli olan mantıksal öğeler ve bu öğeler arasındaki bağlantılar oluşturulur.

Benzetim aşamasında; sentezlenen devre işlevsel doğruluğu açısından test edilir. Herhangi bir hata oluşması durumunda tasarımda düzenlemeler yapılarak sentez işlemi yeniden yapılır.

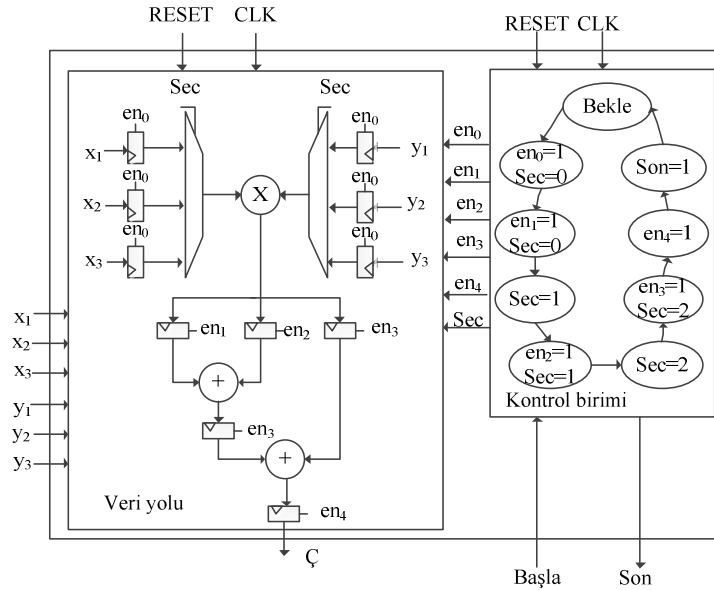
Programlama aşamasında; FPGA kartı USB kablosu üzerinden bilgisayara bağlanarak kart çalıştırılır. Sentez aşamasında elde edilen dosyalar karta yüklenerek tasarımın doğruluğu donanımsal olarak test edilir.

Geliştirilen algoritmanın FPGA üzerinde gerçekleştirilmesi için tasarımcı ilgili FPGA tabanlı mimarinin üzerinde uygulamayı tasarlar. FPGA üreticileri, tasarımcılara uygulamalarını geliştirebilmeleri için birçok ortam sunmaktadır. MATLAB/SIMULINK tarafından desteklenen DSP Builder gibi araçlar kullanılarak SIMULINK ortamında tasarımlar geliştirilebilmektedir. Fakat bu şekilde yapılan tasarımlar FPGA kaynaklarını optimum şekilde kullanmazlar. Tasarımlar karmaşıktıkça FPGA kaynakları yetersiz kalabilir. Bu yüzden tasarımcılar FPGA kaynaklarını optimum şekilde kullanabilmek için uygulamasını kendisi kodlayarak geliştirmelidir. FPGA tabanlı mimarinin optimizasyonu A^3 yöntemine dayalıdır [149]. Bu yöntemin temeli minimum sayıda operatör kullanarak paralel şekilde maksimum işlemin gerçekleştirilmesine dayanır. Bu yöntemde iki grafiksel gösterim kullanılır: veri akış çizgesi ve ayrıştırılmış veri akış çizgesi. Şekil 6.4'te bu grafiksel gösterimler verilmiştir.



Şekil 6.4. A³ yönteminde kullanılan mimari optimizasyon şekilleri

Şekil 6.4'te verilen mimariler iki aritmetik operatör içeren $C = x_1 \cdot y_1 + x_2 \cdot y_2 + x_3 \cdot y_3$ denkleminin gerçekleştirilmesidir. Şekil 6.4 (a) paralel çalışan üç adet çarpma işlemi ve ardışık çalışan iki adet toplayıcıdan ibarettir. Bu mimari paralel çalışan kısımlardan dolayı hızlı çalışmasına rağmen her bir işlem için ayrı bir FPGA kaynağı kullanmaktadır. Yapılan tasarım karmaşıktıkça kullanılan FPGA kaynakları da daha fazla olmaktadır. Şekil 6.4 (b)'de verilen mimaride ise bir adet çarpıcı ve iki adet toplayıcı kullanılmaktadır. Her defasında çarpma işlemi için iki parametre alınmakta ve tek bir çarpıcı ile çarpımı bulup toplama eklemektedir. Şekil 6.5'te bu hesaplamaların yapıldığı bir FPGA tasarım örneği verilmiştir.



Şekil 6.5. FPGA tasarım örneği

Şekil 6.5'te verilen FPGA tasarım örneği; veri yolu ve kontrol birimi, saat ve reset sinyallerine sahiptir. Her bir işlemin çalışma sırası bir sonlu durum makinası ile kontrol edilmiştir. Sonlu durum makinasına başla komutu geldikten sonra son duruma ulaşınca kadar denklemin çıkışını elde etmektedir. Her bir işlemin çalışma sırası seçme uçları (en_x) ile ifade edilmiştir. Denklemin x ve y parametrelerinin alındığı veri çoğullayıcıların seçme uçları alınacak x ve y parametresini belirler. FPGA kartlarının en önemli özelliği eş zamanlı çalışabilmeleridir. Yani eş zamanlı olarak iki sayı hem toplanabilir hem de çarpılabilir. Ayrıca her bir komutun bir saat gecikmesi vardır. Örneğin kayan noktalı sayıları çarparken minimum 5 saat sinyali sonrasında doğru sonuç verilir. Elde edilen çarpma sonucu başka bir denklemde kullanılacaksa bu denklem 5 saat sinyali sonrasında aktifleştirilmelidir. Bu şekilde tasarımda ardışık çalışan blokların çalışma sıraları VHDL veya Verilog donanım tanımlama dillerinde yazılan sonlu durum makineleri ile kontrol edilebilir.

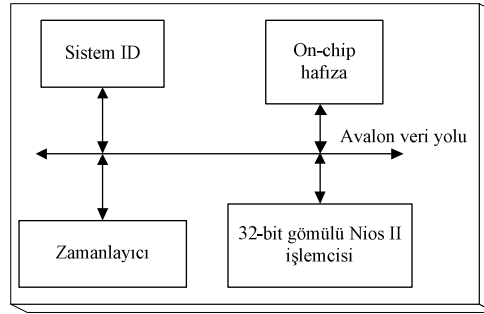
6.1.3. VHDL Donanım Tanımlama Dili ve FPGA'ın Programlanması

FPGA'da istenilen tasarımın gerçekleştirilmesi için tasarıma uygun olarak FPGA içindeki lojik bloklar, bu bloklar arasındaki bağlantılar ve çevresel aygıtlar ile olan bağlantıların oluşturulması gerekir. FPGA kartının nasıl davranacağını belirlemek için tasarımcı donanım tanımlama dillerini veya şematik tasarım araçlarını kullanır. Donanım tanımlama dillerinden en çok bilineni VHDL ve Verilog dilleridir [150]. Tasarımcı ilk olarak gerçekleştirmek istediği fonksiyona ait şematik veya donanım tanımlama dillerinden birinde tasarımını gerçekleştirir ve gerekli benzetimleri yapar. Bu benzetimler sonucunda tasarlanan devre istenildiği gibi sonuç veriyorsa devrenin son hali USB bağlantısı aracılığıyla FPGA kartına yüklenir ve FPGA programlanabilir birimi istenilen işi yapmak üzere programlanmış olur.

FPGA kartlarının iki önemli üreticisi olan Altera ve Xilinx kullanıcılara farklı tasarım araçlarını sunmaktadırlar. Xilinx firması kullanıcılarına tasarımlarını gerçekleştirmek için ISIE Design Suite geliştirme ortamını sunmaktadır [144]. Sunulan ortamda VHDL, Verilog veya şematik tasarım araçları ile projeler geliştirilebilmektedir. Altera firması ise Quartus II geliştirme ortamında tasarımların oluşturulmasına izin vermektedir [151]. Quartus II ortamı da VHDL, Verilog ve şematik tasarımları destekler. Ayrıca her iki FPGA üreticisinin de sunduğu DSP Builder programı Matlab/Simulink ortamında tasarımların gerçekleştirilmesine olanak vermektedir. Kullanıcılar tasarımları

için gerekli olan kodlamaları kendileri yapabileceği gibi hazır olarak gelen fonksiyonları da kullanabilmektedir. Bu fonksiyonlar karmaşık işlemlerin FPGA’da tasarlanması için önceden hazırlanmış ve optimize edilmiş şekilde sunulmaktadır.

Tezde kullanılan Altera Cyclone III geliştirme kitinin programlanması ve gerekli tasarımların oluşturulması için Quartus II 10.1 ve Nios II IDE 10.1 yazılımları mevcuttur. Quartus II içerisinde şematik tasarım blokları oluşturarak istenen tasarımlar yapılabileceği gibi VHDL, Verilog veya SOPC (System on a Programmable Chip Builder) kullanılarak tasarımlar gerçekleştirilebilir. Bir SOPC Builder tasarımı 32-bit Nios II işlemcisi ve çevre birimleri arasında bağlantıların gerçekleştirilmesi ve oluşturulan donanımın C programlama dili kullanılarak programlanması ile FPGA kartına yüklenebilir hale getirilir. SOPC Builder donanım yapısı için gerekli tasarım Quartus II yazılımı ile gerçekleştirilir. Programlama kısmında ise Nios II IDE programı ile gerekli C kodları yazılarak tasarımın istenen işlevi görmesi sağlanır. SOPC Builder’ın donanım yapısı Şekil 6.6’da gösterilmiştir.



Şekil 6.6. SOPC'nin yapısı

Şekil 6.6'daki SOPC sistemi bir adet Nios II işlemcisi, bir adet hafıza birimi, bir adet zamanlayıcı, bir adet JTAG ve bir adet Sistem ID ile donatılmıştır. Sistem donanımları arasındaki iletişim, Altera firması tarafından üretilen Avalon veri yolu ile gerçekleştirilmektedir. Gerekli donanım yapısının bilgisayar ortamında tasarlanması için Quartus II yazılımı içinde bulunan SOPC Builder yazılım ortamı kullanılmaktadır.

Quartus II içinde yazılım aracılığıyla devre tasarlamak için kullanılan bir diğer araç VHDL dilini kullanmaktır. VHDL bir donanım tanımlama dilidir. VHDL VHSIC donanım tanımlama dilinin kısaltmasıdır. VHSIC ise çok yüksek hızlı tümlşik devreleri ifade eder. VHDL ilk olarak 1980'li yıllarda VHDL 87 versiyonu ile geliştirildi. Daha sonra VHDL

93 versiyonuna yükseltildi. VHDL IEEE tarafından standartlaştırılan ilk donanım tanımlama dilidir. VHDL'in iki temel uygulaması FPGA ve ASICs alanlarıdır. VHDL kodu yazıldıktan sonra herhangi bir programlanabilir devreye yüklenebilir veya bir çip üretimi için kullanılabilir. VHDL'in diğer programlama dillerinden farkı eş zamanlı çalışabilmesidir. VHDL'de bu eş zamanlılık dışında ardışık çalışan kod blokları da bulunmaktadır. Bir VHDL dilinde kullanılan ifadeler aşağıdaki gibi verilebilir:

- **Kütüphane (Library):** Tasarımda kullanılan kütüphanelerin listesini kapsar. Bir LIBRARY bildirimi için en basit hali ile iki satırlık kod gerekmektedir. Birinci satır kullanılan kütüphaneyi, diğeri ise bu kütüphanede bulunan paketlerin gösterir. Bu yapı aşağıda gösterilmiştir.

LIBRARY kütüphane_adı;

USE kütüphane_adı.paket_adı.paket_adı.paket_bolumleri;

Yukarıda LIBRARY kısmında belirtilen kütüphane_adı parametresi genelde "IEEE" olarak yazılır. En çok kullanılan paketler aşağıdaki gibidir:

- USE ieee.std_logic_1164.all: STD_LOGIC ve STD_ULOGIC veri tiplerini tanımlar.
- USE std.standard.all: BIT, BOOLEAN, INTEGER ve REAL veri tiplerini tanımlar.
- USE std_logic_arith.all: İşaretili (SIGNED) ve işaretsiz (UNSIGNED) veri tiplerinin tanımlanması ve veri tipleri arasındaki dönüşümlerin gerçekleştiren fonksiyonların kullanılması için gerekli olan bir pakettir.

- **Varlık (Entity):** Oluşturulacak olan donanım devresinin giriş/çıkış pinlerinin belirlenmesini sağlar.

ENTITY tanımlamasında PORT kısmı devrenin giriş/çıkış pinlerinin tanımlandığı kısımdır. PORT kısmındaki sinyal modu IN, OUT, BUFFER veya INOUT olabilir. Bu modlardan IN ve OUT tek yönlü olup sırasıyla giriş ve çıkış değişkenleri tanımlamayı sağlar. İki yönlü olan INOUT pini hem çıkış hem de giriş olarak kullanılabilir. BUFFER modundan ise çıkış sinyali içsel olarak kullanılması gerektiğinde faydalanılabilir. Basit bir sayıcı tasarımı için VHDL'de yazılan entity bloğu aşağıda verilmiştir.

```

Entity sayici is
Port(
    clk, rst: in std_logic;
    Sayac: out std_logic_vector(2 downto 0));
End sayici;

```

- **Mimari (Architecture):** Mimari bloğu oluşturulan devre veya tasarımın nasıl davranacağını yani işlevinin tanımlandığı yerdir. Bir mimari bloğu mutlaka bir entity'e bağlanmalıdır.
- **İşlem (Process):** Mimari bloğunun içinde ardışık olarak çalışan kısımlardan biri Process bloğudur. Process bloğu kendi duyarlılık listesindeki sinyallerde değişimler olduğunda çalışır. Process bloğu içinde if, case, loop gibi karar yapıları ve döngüler kullanılmaktadır. Bir mimari bloğunda içinde birden fazla Process kullanılabilir. Sayıcı tasarımı için Architecture ve Process bloğu aşağıda verilmiştir.

```

Architecture davranis of sayici is
    signal c:std_logic_vector(2 downto 0);
    begin
        process (clk ,rst)
            variable s: integer range 0 to 7;
            begin
                if(rst='1') then
                    s:=0;
                elsif(clk'event and clk='1') then
                    s:=s+1;
                    c<=conv_std_logic_vector(s,3);
                end process;
                sayac<=c;
            end davranis;
    end davranis;

```

Process bloğunda değişken tanımlanırken Variable ifadesi kullanılır. Burada tanımlanan değişkenlere atama işlemi ':= ' ifadesi ile yapılır. Architecture ve entity kısmında tanımlanan değişkenler ise signal olarak ifade edilir. Bu değişkenlere atama işlemi '<=' ile yapılır. VHDL içerisinde kullanılan değişken tiplerinin bazıları aşağıda verilmiştir.

Bit: 0 ve 1 değerini alabilir.

Bit-vector: Aynı isim altında bir dizi ile 0 ve 1'i göstermek için kullanılır.

Integer: Pozitif ya da negatif tamsayıları tanımlar.

Natural: 0'dan başlayıp istenen bir limit değere kadar tamsayılar tanımlamak için kullanılır.

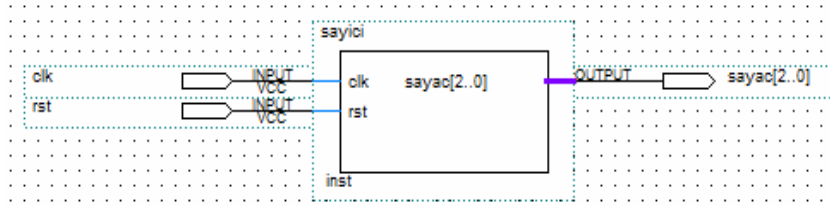
Positive: 1'den başlayıp istenen bir limit değere kadar tamsayılar tanımlamak için kullanılır.

Boolean: Doğru ve yanlış olmak üzere 2 değerli bir veri tipidir.

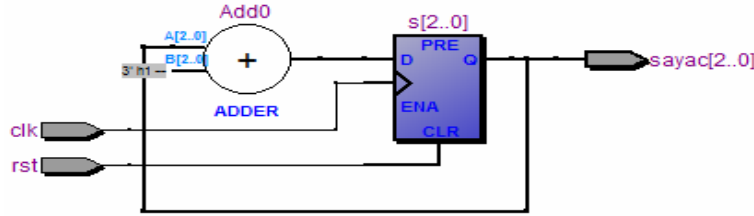
Std_logic: '1' veya '0' değerini alır ve IEEE 1164 standardında tanımlıdır.

Std_logic_vector: '0' ve '1'lerden oluşan dizi tanımlamak için kullanılır.

FPGA'da gerçekleştirilecek tasarımın VHDL kodu yazıldıktan sonra blok diyagramının oluşturulması gerekir. Şekil 6.7'de VHDL kodundan oluşturulan blok diyagramı ve bu diyagramın lojik iç yapısı verilmiştir.



(a) Sayıcı blok diyagramı



(b) Sayıcının iç yapısı

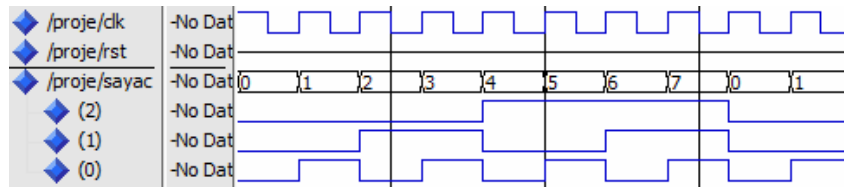
Şekil 6.7. Quartus II sayıcı blok diyagramı ve iç yapısı

Sayıcı devresinin clk (clock) ve rst (Reset) girişleri vardır. Çıkış ise sayaç olarak gösterilmiştir. Sayıcının iç yapısında sayacın başlangıç değeri bir verilerek bir önceki değer ile sürekli toplandığı bir toplama devresi ve istenilen anda sayacı sıfırlamak için D türü flip-flopların kullanıldığı görülmektedir. Quartus II ortamında devre oluşturulduktan sonra tasarım derlenir. Derleme aşamasından sonra tasarımın kullandığı FPGA kaynakları listelenir. Sayıcı devrenin kullandığı FPGA kaynakları Şekil 6.8'de verilmiştir.

Flow Status:	Successful - Wed Apr 20 13:10:11 2011
Quartus II Version	10.1 Build 197 01/19/2011 SP 1 SJ Full Version
Revision Name	proje
Top-level Entity Name	proje
Family	Cyclone III
Device	EP3C120F780C7
Timing Models	Final
Total logic elements	3 / 119,088 (< 1 %)
Total combinational functions	3 / 119,088 (< 1 %)
Dedicated logic registers	3 / 119,088 (< 1 %)
Total registers	3
Total pins	5 / 532 (< 1 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 576 (0 %)
Total PLLs	0 / 4 (0 %)

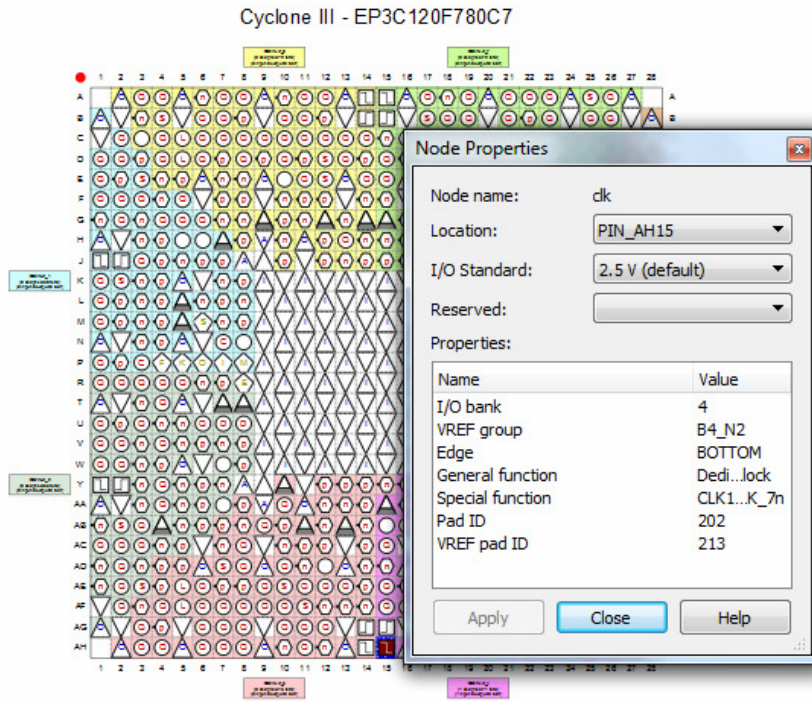
Şekil 6.8. Quartus II’de derleme sonucu sayıcının kullandığı FPGA kaynakları

Şekil 6.8’de görüldüğü gibi sayıcı devresi 3 adet lojik eleman kullanmaktadır. Ayrıca FPGA’nın 532 bacak bağlantısından 5 adet (clk için 1, rst için 1 ve sayaç için 3 olmak üzere) kullandığı görülmektedir. Derleme işleminden sonra tasarımın doğruluğunu onaylamak için benzetim gerçekleştirilir. Benzetim için Quartus II’nin 9.1 ve öncesi sürümlerinde Vector Waveform aracı hazır olarak gelmektedir. Fakat bu araç grafiksel gösterim için uygun değildir. Quartus II 10.0 sürümü ile birlikte Vector Waveform aracı kaldırılmıştır. Bunun yerine benzetim amacıyla Altera Modelsim Starter Edition programı kullanılmaktadır. Ayrıca Signaltap Logic Analyzer programı, tasarım FPGA kartına yüklendikten sonra çalışma esnasında gerçek zamanlı olarak sonuçları gözlemlene imkanı sunmaktadır. Şekil 6.9’da Altera Modelsim benzetim aracıyla sayıcı için yapılan benzetim sonuçları gösterilmiştir.



Şekil 6.9. Sayıcı için Altera Modelsim benzetim sonuçları

Tasarımın doğruluğu benzetim sonuçları ile doğrulandıktan sonra tasarım ile ilgili FPGA pin atamaları yapılarak gerçekleştirilen tasarım karta yüklenir. Şekil 6.10’da Quartus II’de Cyclone III pin atama aracı görülmektedir.



Şekil 6.10. Pin atamaları

VHDL’de tam sayı ve lojik tabanlı işlemler yoğunluktadır. Noktalı sayılar üzerinde işlem yapmak için VHDL’de belirli sınırlamalar vardır. Arıza teşhisinde elde edilen akım, gerilim ve hız gibi sinyaller genelde noktalı sayı olup bu uygulamalarda hassasiyet önemli bir parametredir. Noktalı sayı işlemleri için farklı sayı formatları kullanılmaktadır. En yaygın olarak kullanılan format IEEE-754 kayan noktalı (floating point) sayı formatıdır. Tezde yapılan algoritmaların büyük bir kısmında kayan noktalı sayılara ihtiyaç duyulmuştur. Bu amaçla FPGA’da gerçekleştirilen işlemlerde 32 bitlik IEEE-754 kayan noktalı sayı formatı kullanılmıştır. Dolayısıyla FPGA’da kayan noktalı sayı işlemleri için kullanılan fonksiyonlara bu formata dönüştürülmüş sayıların verilmesi ve çıkış olarak sayıların bu formatta alınması gerekmektedir. Noktalı sayıları sayısal ifade etmenin diğer bir yolu ise sabit noktalı (fixed point) sayı formatını kullanmaktır. Sabit noktalı gösterimde sayının virgülden önceki ve sonraki kısımları için belirli uzunlukta yer ayrılır. Bu gösterim bellekte fazla yer kapladığından kullanımı uygun değildir [152]. Ayrıca sabit noktalı gösterimde gösterilemeyen sabit ve belirsiz sayılar kayan noktalı gösterimde kolaylıkla gösterilebilir. IEEE-754 kayan noktalı gösterimde sayılar 3 kısımdan oluşur:

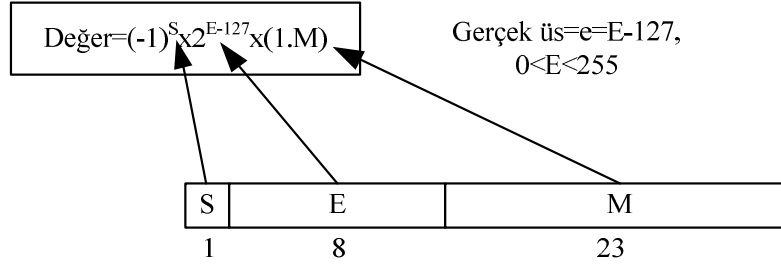
- İşaret
- Üs

- Mantis

IEEE-754 'de 32 bit tek duyarlı ve 64 bit çift duyarlı gösterim mevcuttur. Her iki gösterim biçimi aşağıda verilmiştir.

Sayı formatı	İşaret biti	Üs	Mantis	Bias
32 bit	1 bit [0]	8 bit [1-8]	23 bit [9-31]	127
64 bit	1 bit [0]	11 bit [1-11]	52 bit [12-63]	1023

Şekil 6.11'de 32 bitlik IEEE-754 standardı tanımlanmıştır.

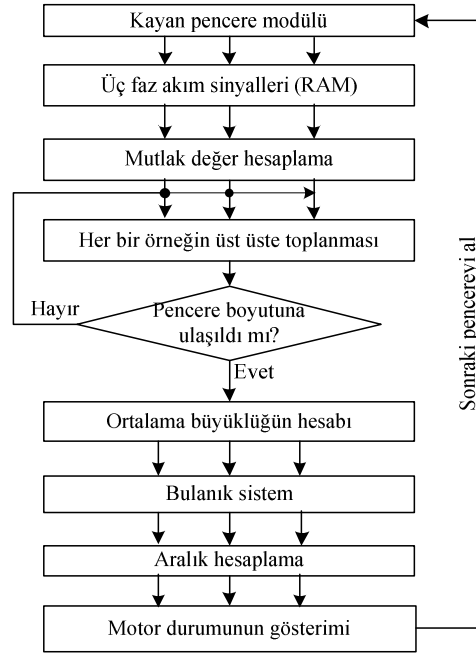


Şekil 6.11. 32 bit IEEE-754 standardı

İşaret biti pozitif sayılar için 0 ve negatif sayılar için ise 1 değerini alır. Üs alanı hem pozitif hem de negatif sayılarda üsleri ifade eder. Bu amaçla bias değeri gerçek üs değeri ile toplanıp üs kısmı oluşturulur. IEEE-754 formatında bias değeri 127 olarak alınır. Mantis ise sayıyı ifade eden bitleri gösterir. Bu, sayının tam ve kesir kısımlarından oluşur. Kayan noktalı sayının 32- bitlik IEEE-754 standardına dönüştürülmesi için MATLAB'ta `num2hex(single(sayı))` fonksiyonu kullanılır. Ters dönüşüm için ise `q=quantizer('single','nearest','saturate')` ve `hex2num(q,"sayı")` fonksiyonları kullanılır. FPGA tasarımlarının gerçekleştirildiği Quartus II programında kayan noktalı sayılar üzerinde temel aritmetik işlemleri gerçekleştirmek için şematik bloklar bulunmaktadır. Bu bloklar kullanılarak kayan noktalı sayılar üzerinde toplama, çıkarma, bölme, çarpma, karekök alma, üs alma ve karşılaştırma gibi işlemler yapılabilmektedir.

6.2. Bulanık Mantık Tabanlı Arıza Teşhis Algoritması

Stator ile ilgili arızaların gerçek zamanlı olarak teşhis edilmesi için bulanık mantık tabanlı bir yöntem önerilmiştir. Önerilen yöntem arıza teşhisi için üç faz akım sinyalinin büyüklüğünü bulanık sistemin girişleri olarak kullanmaktadır. Bulanık sistemin çıkışı motor durumu hakkında bilgi vermektedir. Önerilen yöntemin blok diyagramı Şekil 6.12’de verilmiştir.



Şekil 6.12. Bulanık arıza teşhis sisteminin blok diyagramı

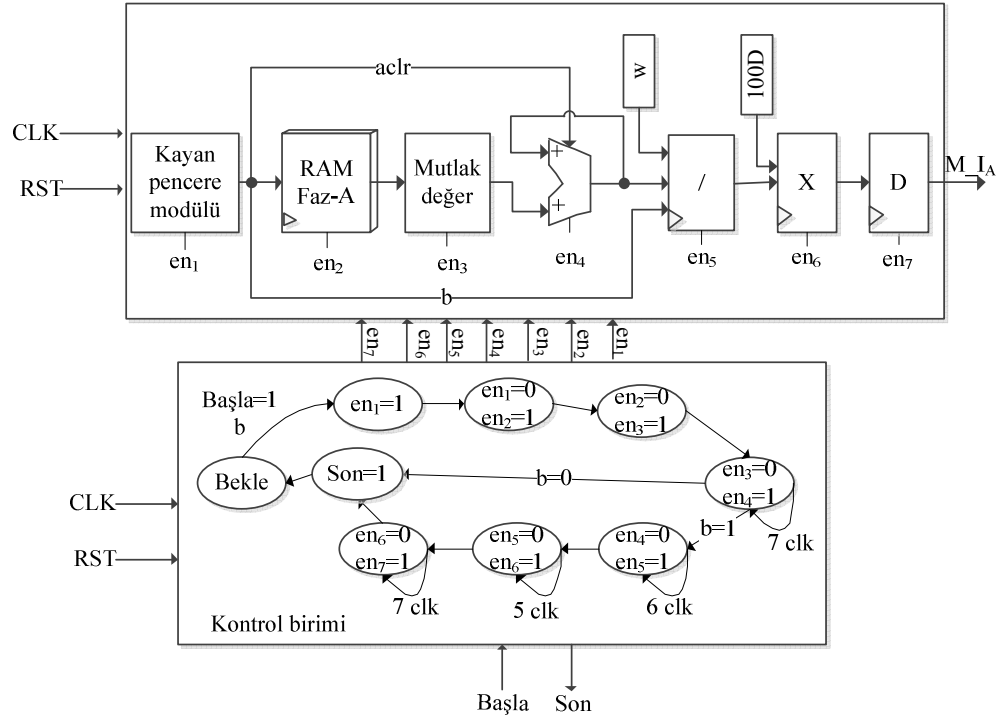
Şekil 6.12’de verilen bulanık sistem, özellik çıkarımı ve bulanık uygulama aşamalarından oluşur. Her iki aşama da FPGA üzerinde gerçekleştirilmektedir. Kayan pencere modülü, belirlenen pencereden ardışık örnekleri okumayı ve sonraki pencereye geçişi sağlar. Bu modül VHDL’de yazılmıştır. Okunan ardışık sinyallerin mutlak değeri alınarak elde edilen pozitif örnekler pencere boyutuna ulaşınca kadar üst üste toplanmaktadır. Elde edilen toplam değeri pencere boyutuna bölünerek her bir fazın ayrı ayrı büyüklüğü hesaplanmaktadır. Bulanık sistem üç faz büyüklüğünü alarak motor durumu hakkında bir sonuç elde eder. Elde edilen sonuç aralık hesaplama modülü ile istenilen dilsel ifadeye dönüştürülerek 2 satır 16 karakterli LCD display’de gösterilmektedir. Pencere fonksiyonu, ardışık olarak kaç adet örneğin alınacağını

dolayısıyla kaç periyot akım sinyalinin okunacağını belirler. VHDL’de yazılan kayan pencere modülü RAM’den okunan ardışık sinyallerin adresini arttırmak ve akım büyüklüklerinin hesaplanmasını sağlayan ortalama bloğunun aktifleştirmek için kullanılır. Kayan pencere için yazılan VHDL kodunun bir kısmı Şekil 6.13’te verilmiştir.

```
ARCHITECTURE behav OF slidingwindow IS
BEGIN
PROCESS(clk)
VARIABLE count,plus,ax : INTEGER range 0 to 1000 ;
BEGIN
IF (clk'EVENT AND clk='1') THEN
    if(enb='1') then
        count := count + 1;
        if(count=ax+60) then
            plus:=plus+1;
            ax:=ax+1;
            count:= plus;
        end if;
    end if;
end if;
RAMADR<=count;
END PROCESS;
END behav;
```

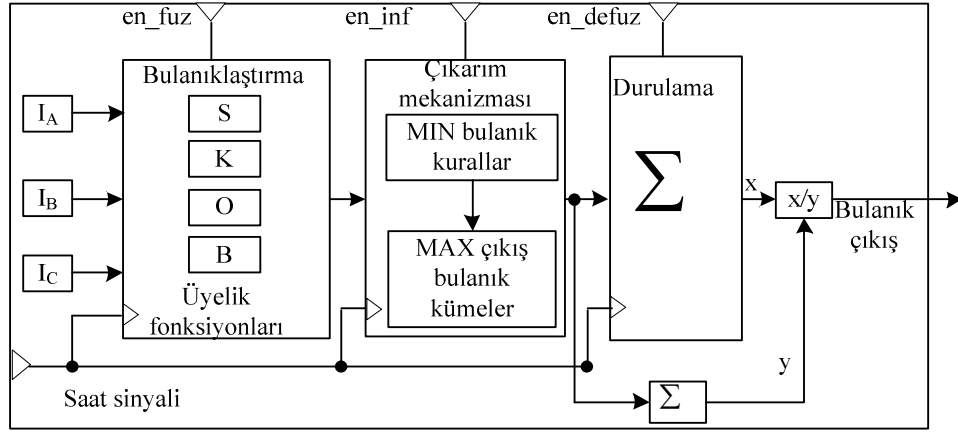
Şekil 6.13. Kayan pencere modülünün VHDL kodu

Pencere boyutu $w=135$ örnek olarak alınmıştır. Tek faz akım sinyalinin büyüklüğünün hesaplanması için oluşturulan tasarımın blok şeması Şekil 6.14’te verilmiştir.



Şekil 6.14. Bir faz akımının büyüklüğünün hesaplanması

Şekil 6.14'te kayan pencere bloğu ile okunan her bir örneğin mutlak değeri hesaplanmakta ve örnekler üst üste toplanarak okunan penceredeki örnek sayısına ulaşıldığında toplam pencere boyutuna bölünmektedir. Böylece ilgili penceredeki akım sinyallerinin büyüklüğü hesaplanmaktadır. Bulanık sistem kısmı VHDL ortamında yazılmış olup tam sayı formatında çalışmaktadır. Bu yüzden elde edilen akım büyüklükleri 100 ile çarpılarak bulanık sisteme verilmektedir. Ardışık olarak çalışan blokların çalışma sıraları sonlu durum makinası ile kontrol edilmektedir. Her bir bloğun aktifleştirme (en) girişi belirli saat sinyali boyunca aktifleştirilerek istenen çıkış elde edilmekte ve bir sonraki bloğa verilmektedir. Kayan pencere modülündeki sayaç değeri pencere boyutuna ulaşıldığında bölme işlemi yapılması sonucunda, akım büyüklüğü elde edilmekte ve toplam ac_{lr} girişi ile sıfırlanmaktadır. Elde edilen akım değeri 100 desimal değeri ile çarpılarak tam sayıya dönüştürülmektedir. Bulanık sistem üç giriş ve tek çıkışa sahiptir. Bulanık sistemin blok şeması Şekil 6.15'te verilmiştir.



Şekil 6.15. Bulanık sistemin blok diyagramı

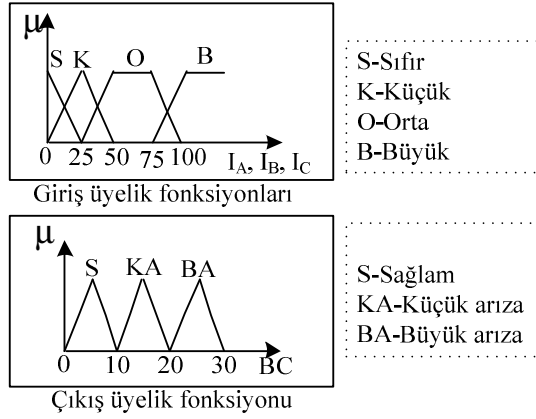
Bulanık sistemin her bir bloğu saat sinyali dışında bir aktifleştirme girişine sahiptir. Bulanık sistemin giriş ve çıkışlarını belirleyen VHDL kodu aşağıdaki gibidir.

```
ENTITY bulanik IS
PORT( IA, IB, IC: IN integer;
      Clock: IN std_logic;
      en_fuz, en_inf, en_defuz: IN std_logic;
      FO: OUT integer
);
```

Bulanıklaştırma bloğu üyelik fonksiyonlarını kullanarak giriş değerlerini bulanık değerlere dönüştürür. Önerilen bulanık sistem tam sayı formatında çalıştığı için bulanıklaştırma işlemi eğime göre hesaplanmaktadır [153]. Örneğin S üyelik fonksiyonunun eğimi $1/25=0.004$ 'tür. Bu değer 1000 ile çarpılarak bir tamsayı değere dönüştürülür. Bulanıklaştırma için yazılan VHDL kodunun bir kısmı aşağıdaki gibidir.

```
MIA_S<=0 WHEN (IA<=0)
ELSE
  (4*(25-IA)) WHEN (IA<=25)
ELSE 0;
```

Önerilen bulanık sistemin üyelik fonksiyonları ve kural tabanı Şekil 6.16'da verilmiştir.



(a) Üyelik fonksiyonları

Kural-1: IF I_A is S THEN BC is BA
 Kural-2: IF I_B is S THEN BC is BA
 Kural-3: IF I_C is S THEN BC is BA
 Kural-4: IF I_A is B THEN BC is BA
 Kural-5: IF I_B is B THEN BC is BA
 Kural-6: IF I_C is B THEN BC is BA
 Kural-7: IF I_A is K and I_B is K and I_C is O THEN BC is KA
 Kural-8: IF I_A is K and I_B is O and I_C is O THEN BC is KA
 Kural-9: IF I_A is O and I_B is K and I_C is O THEN BC is KA
 Kural-10: IF I_A is K and I_B is O and I_C is K THEN BC is KA
 Kural-11: IF I_A is O and I_B is K and I_C is K THEN BC is KA
 Kural-12: IF I_A is O and I_B is O and I_C is K THEN BC is KA
 Kural-13: IF I_A is K and I_B is K and I_C is K THEN BC is S
 Kural-14: IF I_A is O and I_B is O and I_C is O THEN BC is S

(b) Kural tabanı

Şekil 6.16. Üyelik fonksiyonları ve kural tabanı

Çıkarım mekanizması olarak Mamdani Min-Max yöntemi kullanılmıştır. Örneğin Kural-7 “IF I_A is K and I_B is K and I_C is O then BC is KA” kuralı için $C^7 = \min[K_{IA}, K_{IB}, O_{IC}]$ ile verilir. Tasarlanan bulanık sistem 3 sınıfa sahip olduğundan kuralları birleştirmek için aşağıdaki işlem uygulanmaktadır [154].

$$D1 = \max[C^1, C^2, C^3, C^4, C^5, C^6]$$

$$D2 = \max[C^7, C^8, C^9, C^{10}, C^{11}, C^{12}]$$

$$D3 = \max[C^{13}, C^{14}]$$

Durulama aşaması ağırlıklı ortalama yöntemine göre yapılmıştır. Bu yöntem denklem 6.1’de verilmiştir.

$$y = \frac{\sum \mu(z_t) \times z_t}{\sum \mu(z_t)} \quad (6.1)$$

Durulandırma işlemi için yazılan VHDL kodunun PROCESS bloğu Şekil 6.17’de verilmiştir.

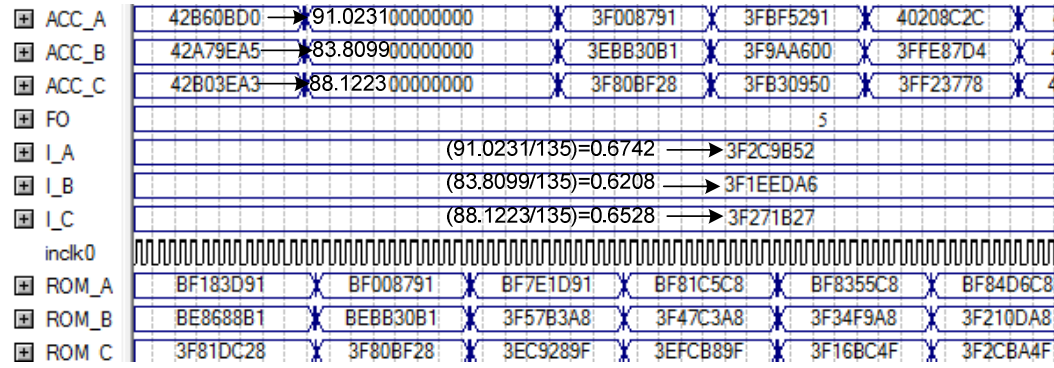
```

PROCESS(clk)
VARIABLE Dv, Ds:integer;
BEGIN
  IF(defuz_enb='1') then
    IF(CLK'EVENT AND CLK='1') THEN
      DV:=(25*D1)+(15*D2)+(5*D3);
      DS:=(D1+D2+D3);
      IF(Ds=0) THEN
        Y:=0;
      ELSE
        Y:=(DV/DS);
      END IF;
    END IF;
  END IF;
END PROCESS;

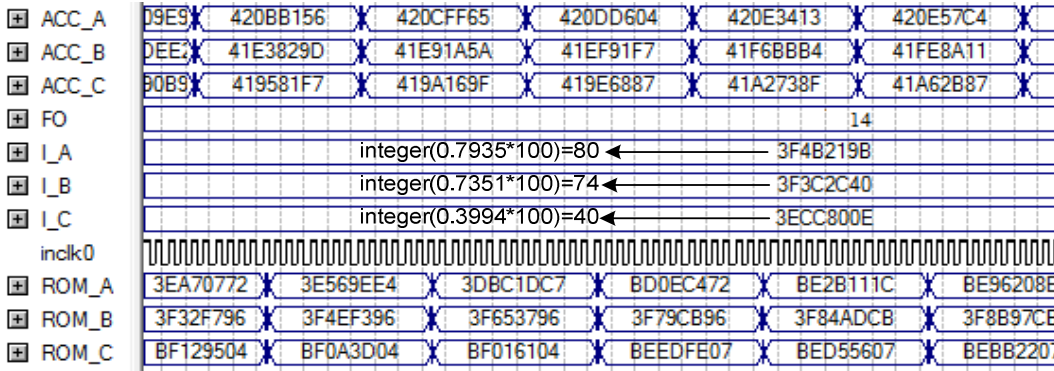
```

Şekil 6.17. Durulandırma işlemi için PROCESS bloğu

Bütün uygulamalar Quartus II 10.1 SP1 ortamında VHDL donanım tanımlama dili ile gerçekleştirilmiştir. Quartus II yazılımı aynı zamanda FPGA üzerinde tasarımın uygulanması ve analizi için kullanılmıştır. Sağlam ve arızalı motor durumu için Quartus II ortamında gerçekleştirilen deneysel sonuçlar Şekil 6.18’de verilmiştir.



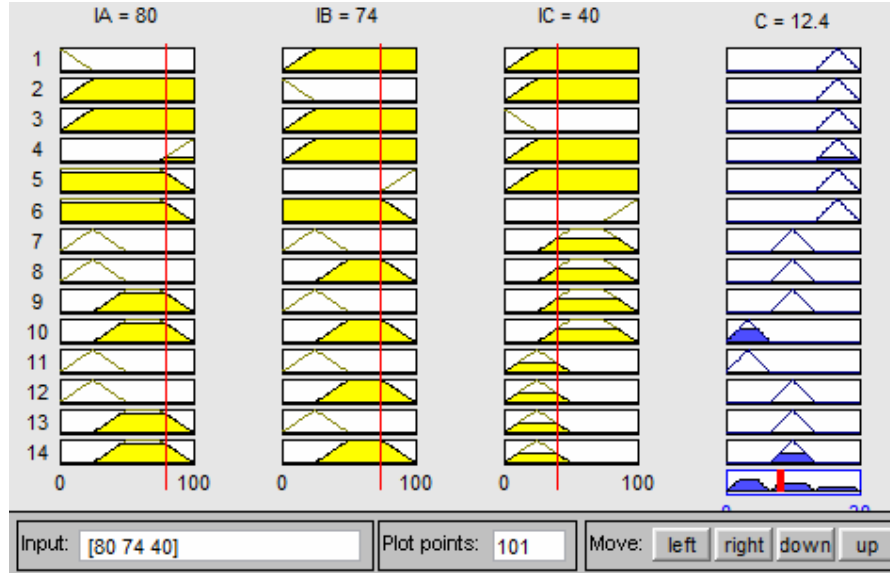
(a) Sağlam durum



(b) Stator arızası

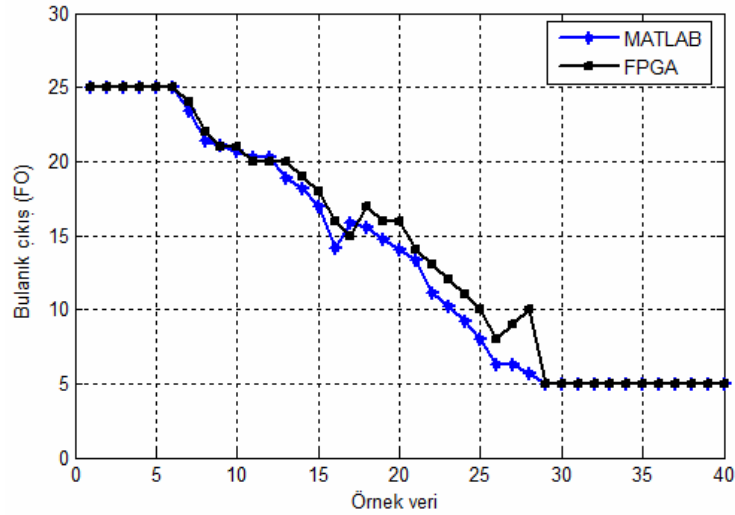
Şekil 6.18. Sağlam ve arızalı durumlar için FPGA deneysel sonuçları

Şekil 6.18’de bir pencereden okunan faz akımı sinyali örnekleri ROM_A, ROM_B ve ROM_C ile ifade edilmiştir. Okunan akım sinyallerinden her bir örneğin mutlak değeri alınarak üst üste toplanmaktadır. Her bir fazın toplam ifadesi ACC_A, ACC_B ve ACC_C ile gösterilmiştir. Pencere boyutu 135 örnek alınarak 135. örnek okunduğunda toplam ifadesi pencere boyutuna bölünmektedir. Dolayısıyla toplam ifadeleri şekilden de görüldüğü gibi sıfırlanmaktadır. Bölme sonucu elde edilen faz akım sinyallerinin büyüklükleri I_A, I_B ve I_C ile gösterilmiştir. Elde edilen bu büyüklükler 100 sabit değeri ile çarpılarak altfp_convert bileşeni ile tam sayıya dönüştürülür. Dönüşüm sonucu elde edilen üç faz akım sinyalinin büyüklüğü bulanık sisteme verilerek arıza durumu hakkında bulanık sistem bir çıkış (FO) üretir. Aynı giriş değerleri için MATLAB bulanık mantık TOOLBOX sonuçları Şekil 6.19’da verilmiştir.



Şekil 6.19. Aynı giriş değerleri için MATLAB sonuçları

FPGA üzerinde gerçekleştirilen yöntemin sonuçları MATLAB Bulanık mantık TOOLBOX ile karşılaştırılmıştır. Karşılaştırma sonuçları Şekil 6.20’de verilmiştir.



Şekil 6.20. FPGA sonuçları ile MATLAB bulanık mantık TOOLBOX’ın karşılaştırılması

Şekil 6.20’ye göre FPGA ve MATLAB bulanık mantık TOOLBOX sonuçlarının oldukça yakın olduğu görülmektedir. Aradaki hatanın sebebi, FPGA ortamında gerçekleştirilen bulanık sistemin tamsayı tabanlı çalışmasıdır. Tasarlanan bulanık sistem FPGA kartına yüklenerek sonuçları gözlemlenmiştir. Şekil 6.21’de bulanık sistemin çıkışı

20 ile 30 arasında bir deęer çıktığında, dolayısıyla arıza büyük olduęunda FPGA kartındaki LCD display’de ilgili mesaj gösterilmektedir.



Şekil 6.21. Bulanık sistemin FPGA uygulaması

Oluşturulan tasarımın FPGA üzerinde gerçekleştirilmesi sonucu harcadığı donanım kaynakları Tablo 6.1’de verilmiştir.

Tablo 6.1. FPGA tabanlı bulanık sistem tasarımı için gerekli kaynaklar

Hafıza birimi	Kullanım oranı
Lojik elemanlar	11.008/119.088 (%9)
Bileşimsel lojik fonksiyonlar	9.825/119.088 (%8)
Kullanılan lojik kayıtcılar	3.882/119.088 (%3)
Toplam hafıza bitleri	96.336/3.981.312 (%2)
Gömülü çarpıcılar	73/576 (%13)

Tablo 6.1’de görüldüğü gibi tasarlanan bulanık arıza tespit sistemi FPGA’ın sınırlı kaynaklarını kullanır. Önerilen bulanık sistem tabanlı arıza teşhis yöntemi faz akım sinyallerinin büyüklüklerine göre arıza durumu hakkında bilgi vermektedir. Yöntemin doğruluğu hem benzetim hem de donanımsal gerçekleştirme ile ispatlanmıştır. Bulanık çıkarım sistemi giriş verilerini değerlendirerek motor durumu hakkında gerekli bilgiyi LCD display’de kullanıcıya göstermektedir.

6.3. Yapay Sinir Ağı Tabanlı Arıza Tahmin Algoritması

Arıza tahmininde amaç anlık değişimlerin belirlenmesini sağlamaktır. Motordan alınan üç faz akım sinyallerinden herhangi birinde bir değişim olduğunda bu değişimin tespit edilmesi gerekir. Önerilen arıza tahmin algoritması özellik çıkarımı ve durum tahmini olmak üzere iki aşamadan oluşmaktadır. Özellik çıkarım aşamasında yapay sinir ağının girişleri için tek faz akım sinyalinden özellikler çıkarılmaktadır. Durum tahmini aşamasında ise girişler gerçek zamanlı olarak yapay sinir ağına verilerek motor durumu belirlenmektedir.

6.3.1. Yapay Sinir Ağı için Özellik Çıkarımı

Özellik sinyali, tek faz akım sinyali üzerinden elde edilmektedir. Bu sinyal tek faz akım sinyali Hilbert dönüşümü uygulanması ile elde edilmektedir. Sinüs bir sinyalin Hilbert dönüşümünün iki karmaşık bileşeni olup, biri sinyalin kendisi diğeri ise orijinal sinyalin 90^0 faz kaydırılmış halidir. Dolayısıyla ROM'dan veri okunurken bu faz kaydırması göz önüne alınmıştır. Faz kaydırması ile oluşturulan iki bileşene sahip Hilbert dönüşümünün yazımı aşağıda verilmiştir.

$$H(I) = I[k] + jI_{s_{90}}[k] \quad (6.2)$$

Denklem 6.2'den de görüldüğü gibi Hilbert dönüşümü iki karmaşık bileşenden oluşur. Bu bileşenlerin büyüklüğü hesaplanarak özellik sinyali denklem 6.3'e göre elde edilmektedir.

$$|H_A[k]| = \sqrt{I[k]^2 + I_{s_{90}}[k]^2} \quad (6.3)$$

Özellik sinyalinin elde edilmesi için RAM'dan aralarında 90 derecelik faz farkı olan iki adresin okunması gerekmektedir. Okunan akım sinyallerinin örnekleme oranı bilindiğinden herhangi bir penceredeki örnek sayısı kolayca hesaplanır. Dolayısıyla 90 derecelik faz kaydırması da bulunabilir. RAM'den adreslerin okunması için kullanılan pencere fonksiyonunun VHDL kodu Şekil 6.22'de verilmiştir.

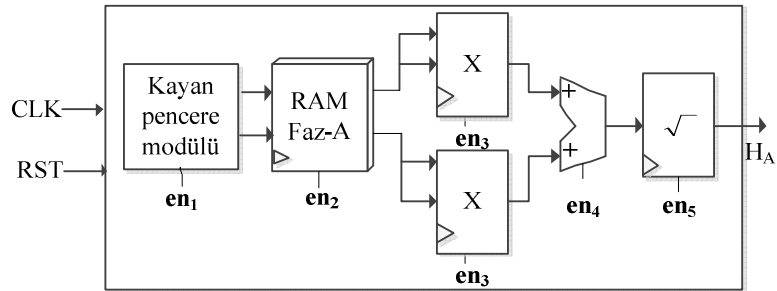
```

PROCESS(clk)
Variable s1,plus : Integer range 0 to 30000 ;
Variable s2: Integer range 88 to 30000;
Begin
If (clk'event and clk='1') then
    If(enb='1') then
        s1 := s1+ 1;
        s2 := s2+ 1;
        If(s1=plus+135) then
            plus:=plus+1;
            s1:= plus;
            s2:=plus+88;
        end if;
    end if;
end if;
ROMADR1<=s1; ROMADR2<=s2;
End PROCESS;

```

Şekil 6.22. RAM'den okuma için kullanılan VHDL kodunun PROCESS bloğu

Şekil 6.22'de s1 ve s2 değişkenleri okunacak RAM adresini göstermektedir. Okunan iki ardışık pencere arasındaki kaydırma miktarı ise plus değişkeni ile belirtilmiştir. Şekil 6.23'te RAM'den okuma ve özellik sinyalinin hesaplanması için oluşturulan tasarımın blok şeması verilmiştir.



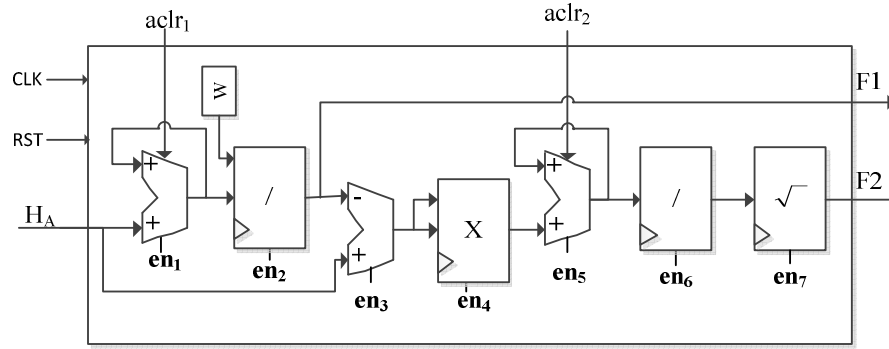
Şekil 6.23. Akım sinyalinden özellik sinyalinin elde edilmesi

Şekil 6.23'te özellik sinyali elde edilirken bileşenin çalışma sıraları sonlu durum makinası tabanlı kontrol bloğu ile gerçekleştirilmektedir. Özellik sinyalinin ortalama değeri ve standart sapması yapay sinir ağının iki girişini oluşturmaktadır. Girişlerin standart sapma ve ortalama alınmasının sebebi, anlık arızaların tahmin edilebilmesinin sağlanmasıdır. Arıza ilk oluştuğunda ortalama değer ve standart sapmada bir değişim gözlemlenir. Motor arızalı durumda sürekli çalışırken standart sapma çok küçük olmasına

rağmen, ortalama değişmektedir. Elde edilen özellik sinyalinin ortalama değeri ve standart sapması denklem 6.4'e göre hesaplanmaktadır.

$$F_1 = \frac{1}{N} \sum_{n=1..N} H_A[n], \quad F_2 = \sqrt{\frac{1}{N} \sum_{n=1..N} (H_A[n] - F_1)^2} \quad (6.4)$$

Denklem 6.4'te elde edilen iki özellik sinyali birbirine bağımlıdır. Standart sapma hesaplanmadan önce ortalamanın bulunması gerekmektedir. Bu işlem aynı pencerenin RAM'den iki kez okunmasını gerektirir. Kayan pencere modülünde RAM'den aynı verilerin iki kez okunması sağlanmıştır. İlk okumada ortalama hesaplanmakta ve ikinci okumada her bir değer ortalamaдан çıkarılıp karesi alınarak üst üste toplanmakta ve standart sapma belirlenmektedir. Şekil 6.24'te standart sapma ve ortalamanın hesaplanması için oluşturulan tasarım verilmiştir.



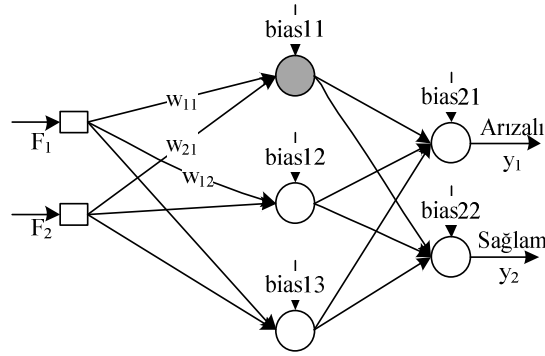
Şekil 6.24. Özellik sinyalinin ortalama ve standart sapma değerinin hesaplanması

Şekil 6.24'te yapay sinir ağının girişlerini hesaplamak için elde edilen özellik sinyali kullanılmaktadır. Özellik sinyalinin bir pencere boyutu kadar alınan örnekler üst üste toplanmaktadır. Bu bloğun aclr₁ girişi bir sonraki pencere verileri geldiğinde toplamı sıfırlamak için kullanılmıştır. Ayrıca her bloğun en_x girişleri blokların çalışma sırasını düzenlemektedir. Bir pencere boyutu kadar veri okunduktan sonra üst üste toplanan verilerin ortalaması, toplam ifadesi pencere boyutuna (w) bölünerek hesaplanmaktadır. Ortalama elde edildikten sonra aynı penceredeki veriler okunarak, her bir veri ortalama değerden çıkarılmaktadır. Elde edilen her bir fark değerinin karesi alınıp üst üste toplanmaktadır. Pencere boyutuna ulaşıncı toplam değer pencere boyutuna bölünerek elde edilen sonucun karekökü alınır. Böylece standart sapma da hesaplanmış olunur.

Yapay sinir ağıının girişlerini oluşturan standart sapma ve ortalama özellikleri tamamen IEEE-754 kayan noktalı sayı formatındadır.

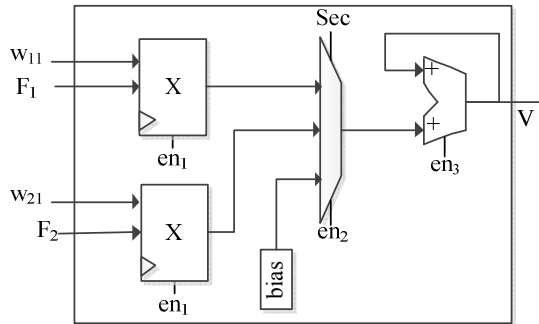
6.3.2. Yapay Sinir Ağı ile Arıza Tahmini

Arıza tahmini için üç katmanlı ileri beslemeli bir yapay sinir ağı önerilmiştir. Yapay sinir ağı iki giriş ve iki çıkışa sahiptir. Orta katmanda ise üç adet nöron kullanılmıştır. Oluşturulan yapay sinir ağıının yapısı Şekil 6.25'te verilmiştir.



Şekil 6.25. Yapay sinir ağı yapısı

Yapay sinir ağıının eğitimi MATLAB'ta gerçekleştirilerek ağıın ağırlık değerleri elde edilmiştir. Elde edilen ağırlıklar FPGA ortamında oluşturulan test yapay sinir ağıının ağırlıkları olup, gelen test verilerine göre yapay sinir ağı arıza durumu hakkında bilgi vermektedir. Her bir nöronun aynı zamanda bir bias girişi bulunmaktadır. Quartus II'de tasarlanan bir nöronun blok şeması Şekil 6.26'daki gibi verilebilir.

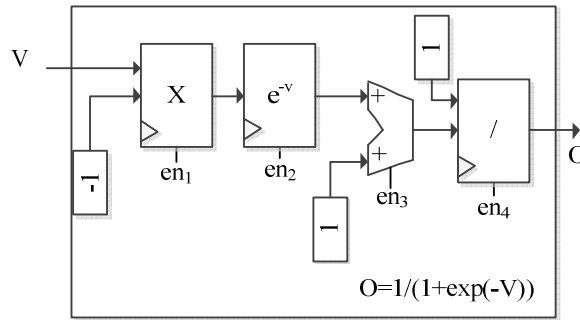


Şekil 6.26. Nöron modelinin blok şeması

Şekil 6.26’da verilen nöron modeli birinci katmandaki iki özellik ve bias değerini almaktadır. Her bir özellik eğitim aşamasında elde edilen ağırlık değerleri ile çarpılıp bias değeri ile toplanmakta ve elde edilen toplam aktivasyon fonksiyonunun giriş değerini oluşturmaktadır. Orta katmanda tek yönlü sigmoid aktivasyon fonksiyonu kullanılmıştır. Bu aktivasyon fonksiyonu denklem 6.6’da verilmiştir.

$$F = \frac{1}{1 + e^{-\nu}} \quad (6.6)$$

Orta katmandaki aktivasyon fonksiyonunun Quartus II tasarımının blok şeması Şekil 6.27’de verilmiştir.



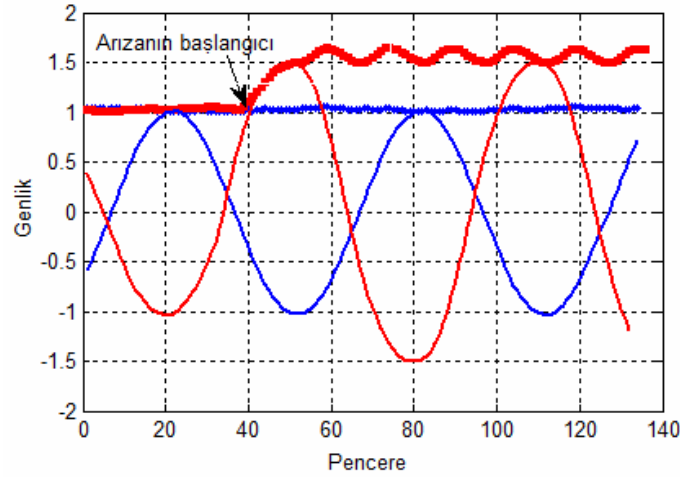
Şekil 6.27. Aktivasyon fonksiyonunun Quartus II modeli

Şekil 6.27’de bir nörona gelen her bir girişin ilgili ağırlık ile çarpılarak üst üste toplanması ile aktivasyon fonksiyonunun girişi (V) elde edilir. Elde edilen giriş ilk olarak (-1) ile çarpılmakta ve üst alma işlemi (e^{-V}) yapılmaktadır. Üst alma sonucu elde edilen değer +1 ile toplanmaktadır. Paydaki 1 değeri sonuca $(1+e^{-V})$ bölünerek O çıkışı elde edilmektedir. Orta katmandaki her bir nöron için bu işlem gerçekleştirilmektedir.

6.3.3. Deneysel Sonuçlar

Önerilen tahmin algoritmasının performansının değerlendirilmesi için 0.37 kW'lık asenkron motordan elde edilen akım sinyalleri kullanılmıştır. Testler motorun sağlam ve faz arızası durumu için gerçekleştirilmiştir. Önerilen yöntem üç fazlı bir asenkron motorun herhangi bir fazında oluşan anlık değişimleri tespit etmek için kullanılmaktadır. Akım sinyalleri veri toplama kartı üzerinden bilgisayara 3000 örnek/saniye örnekleme oranı ile 3 saniye boyunca kaydedilmektedir. RAM'den okunan akım sinyalleri sağlam bir motora ait

ise özellik sinyalinin ortalaması sabit olup standart sapma sıfıra çok yakın çıkar. Fakat okunan sinyallerde arıza oluşmaya başlamış ise ortalama değerde bir artış gözlenirken standart sapma büyür. Şekil 6.28’de sağlam ve arızalı durumda faz akımları ve arıza oluşma durumu modellenmiştir.



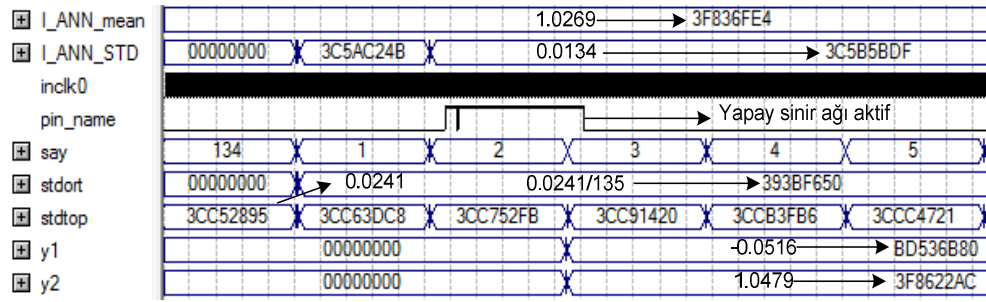
Şekil 6.28. Sağlam ve arızalı faz akımı ve özellik sinyalleri

Yapay sinir ağı bloğunun FPGA kartına yüklenmesi sonucu kullanılan hafıza birimlerinin oranı Tablo 6.2’de verilmiştir.

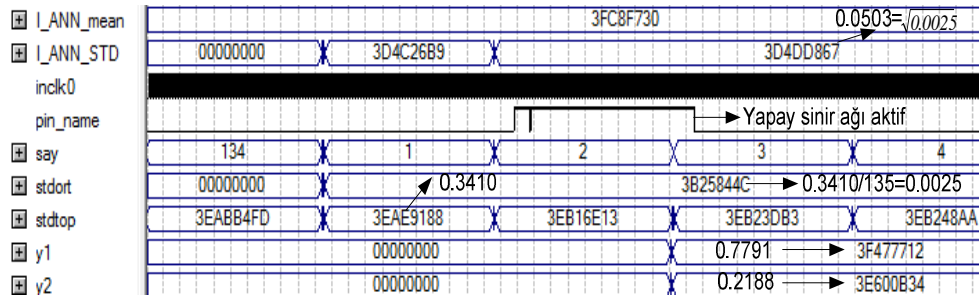
Tablo 6.2. Önerilen yapay sinir ağı için kullanılan FPGA kaynakları

Hafıza birimi	Kullanım oranı
Lojik elemanlar	24.516/119.088 (%21)
Bileşimsel lojik fonksiyonlar	23.071/119.088 (%19)
Kullanılan lojik kayıtlar	12.071/119.088 (%11)
Toplam hafıza bitleri	46.936/3.981.312 (%1)
Gömülü çarpıcılar	299/576 (%52)

Tablo 6.2’de görüldüğü gibi FPGA kartı üzerinde yapay sinir ağı bloğunun en çok harcadığı birimler 9 bitlik gömülü çarpıcılardır. Bunun sebebi yapay sinir ağının temelini oluşturan çarpma, bölme ve aktivasyon fonksiyonlarında kullanılan üs alma işlemlerinin gömülü çarpıcıları yoğun şekilde kullanmasıdır. Şekil 6.29’da sağlam ve arızalı durumda FPGA üzerinde oluşturulan yapay sinir ağının giriş ve çıkış değerleri verilmiştir.



(a) Sağlam durum



(b) Arıza oluşması

Şekil 6.29. Yapay sinir ağının sağlam ve arızalı durumlar için FPGA üzerinde test edilmesi

Şekil 6.29 (a)'da sağlam durum için Quartus II ortamında gerçekleştirilen uygulama sonuçları görülmektedir. Sağlam durumda özellik sinyalinin ortalaması 1.029 ve standart sapması 0.013 çıkmaktadır. Şekil 6.29'da gösterilen stdtop standart sapmadaki toplam ifadesini, stdort standart sapma hesaplanırken stdtop değerinin pencere boyutuna (135) bölünmesi ile elde edilen ortalama değeri ifade eder. Elde edilen ortalamanın karekökü alındığında standart sapma hesaplanmış olur. Özellik sinyalinin elde edilmesi ve bu sinyalin ortalaması ile standart sapmasının hesaplanması özellik çıkarım aşaması olup uzun hesaplamalar gerektirmektedir. Örneğin standart sapmanın elde edilmesi için ilk önce ortalamanın bulunması ve bu ortalamadan her bir değer çıkarılması gerekmektedir. Yapay sinir ağı kısmı 86 saat sinyalinde çıkış vermektedir. Dolayısıyla yapay sinir ağının girişleri elde edildikten sonra 86 clock boyunca aktif (pin_name) edilerek çıkış hesaplanmaktadır. Bu arada diğer pencerenin akım sinyalleri de okunmaktadır. Motor durumu ile ilgili bilgiler FPGA kartı üzerinde bulunan LCD display'de gösterilmektedir.

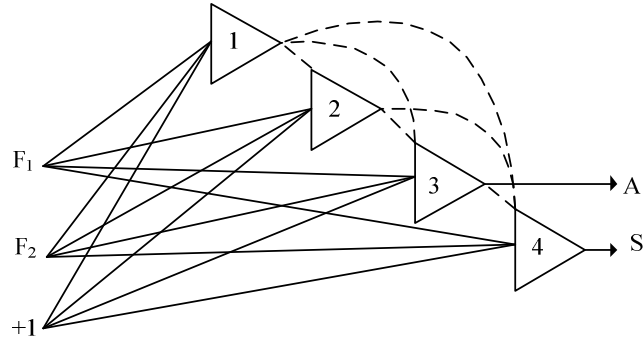
6.4. Kaskad Yapay Sinir Ağı Tabanlı Arıza Teşhisi

İleri beslemeli yapay sinir ağları fonksiyon yaklaştırma, örüntü tanıma, sınıflandırma ve kontrol problemleri için uygundur. Fakat ileri beslemeli yapay sinir ağları problemi öğrenmek için orta katmanda çok fazla nöron kullanmaktadır. Orta katmanda fazla nöron kullanılması yapay sinir ağının genelleme yeteneğini düşürerek ağın problemi ezberlemesine neden olur. Ayrıca ileri beslemeli yapay sinir ağlarının eğitiminde kullanılan geriye yayılım öğrenme algoritması diğer öğrenme yöntemlerine göre oldukça yavaş çalışmaktadır [155]. Yapay sinir yapısı dışında ağ boyutunun seçimi önemli bir parametredir. Çok sayıda nöron içeren yapay sinir ağlarının eğitiminde optimum yakınsamayı sağlamak oldukça kolaydır. Fakat ağın eğitimindeki bu başarımla çoğunlukla aldatıcıdır. Çünkü çok sayıda nöron içeren bir yapay sinir ağı iyi bir sonuç çıkarma yeteneğine sahip olmadığı gibi eğitim sürecinde kullanılmayan verilere karşı yanlış sonuçlar verebilir. Yapay sinir ağlarında istenen az sayıda ve basit bir yapıda bir sinir ağı oluşturularak karmaşık problemlerin çözümünün sağlanmasıdır. Özellikle gömülü sistemlerde donanımsal olarak gerçekleştirim için ağ boyutu oldukça önemli bir parametredir. Bilgisayardan farklı olarak gömülü sistemler sınırlı hafıza boyutu ve hesaplama gücüne sahiptir [156]. Dolayısıyla gerçekleştirilecek tasarımlar hem donanım hem de yazılım açısından basit olmalıdır.

Tezin bu bölümünde arıza teşhisi için kaskad bir yapay sinir ağı yapısı kullanılmıştır. Kaskad yapay sinir ağı minimum sayıda nöron kullanmaktadır. Dolayısıyla FPGA üzerinde gerçekleştirilecek yapay sinir ağı minimum donanım kaynaklarına ihtiyaç duymaktadır. Yapay sinir ağının eğitimi PSO ile gerçekleştirilerek optimum ağırlıklar elde edilmiştir. Yapay sinir ağı, dördüncü bölümde DVM'nin eğitimi için kullanılan veriler üzerinde test edilerek sağlam ve bir kırık rotor çubuğu arızası sınıflandırılmıştır.

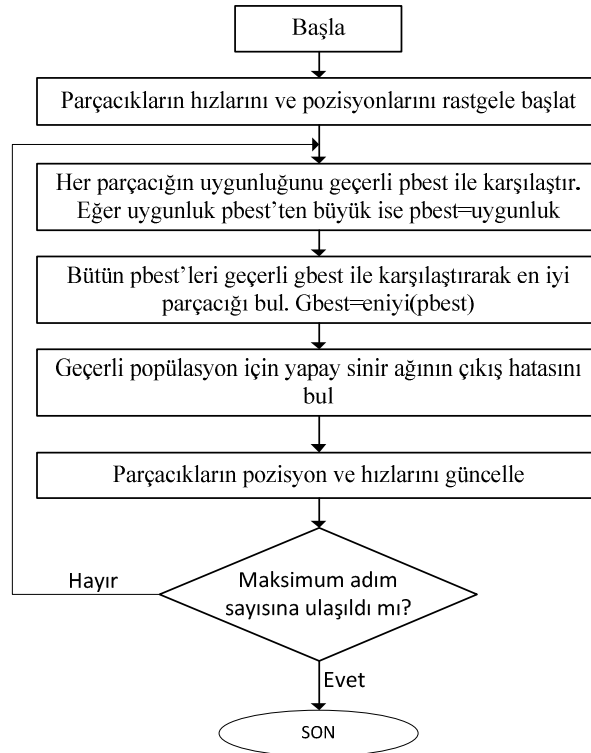
6.4.1. Kaskad Yapay Sinir Ağının Yapısı ve Eğitimi

Arıza teşhisi için kullanılan kaskad yapay sinir iki giriş ve iki çıkışa sahiptir. Yapay sinir ağının girişleri bütün katmanların girişini oluşturmaktadır. Giriş katmanı dışındaki katmanlar bir önceki katmanın çıkışını giriş olarak alırlar. Kaskad yapay sinir ağı toplamda dört nörona sahiptir. Farklı problemler için yapay sinir ağının yapısı kolayca değiştirilebilmektedir. Arıza teşhisi için kullanılan kaskad yapay sinir ağı Şekil 6.30'da verilmiştir.



Şekil 6.30. Arıza teşhisi için önerilen kaskad yapay sinir ağının yapısı

Önerilen kaskad yapay sinir ağı iki giriş ve iki çıkışa sahiptir. Girişler DVM algoritmasında elde edilen iki boyutlu faz uzayındaki her bir boyuttur. Çıkış ise sağlam ve arızalı durumlar olarak ifade edilmiştir. Her bir nöronun çıkışı sonraki nöronların girişi olarak kullanılmıştır. Yapay sinir ağının eğitimi PSO algoritması ile gerçekleştirilmiştir. Eğitim için önerilen PSO algoritmasının akış şeması Şekil 6.31’de verilmiştir.



Şekil 6.31. Eğitim için önerilen PSO'nun akış şeması

PSO ile ağ ağırlıklarını değerlendirmek için her parçacık ağın ağırlıkları olarak kodlanmıştır. Yapay sinir ağının ağırlıkları $3+4+5+6=18$ olduğundan her bir parçacığın boyutu 18 olarak belirlenmiştir. Şekil 6.31’de akış şeması verilen eğitim algoritmasının adımları aşağıda verilmiştir.

Adım 1: Her parçacığın pozisyonunu $[-1, 1]$ ve hızını $[0, 1]$ aralığında rastgele üret.

Kaskad bağlantılı yapay sinir ağının eğitiminde her bir katmandaki ağırlıklar parçacığın pozisyonu olarak kodlanmıştır. Bir parçacık denklem 6.7 ile gösterilebilir.

$$W = \{W_i^{[1]}, W_i^{[2]}, W_i^{[3]}, W_i^{[4]}\} \quad (6.7)$$

Denklem 6.7’de $W_i^{[1]}$ i. parçacık için ağın birinci katmanındaki ağırlıklar vektörünü göstermektedir. Bu adımda her bir parçacığın o ana kadarki en iyi pozisyonu ve popülasyonun en iyi pozisyonu geçerli W vektörü olarak alınmaktadır. Herhangi bir i parçacığının o ana kadarki en iyi pozisyonu denklem 6.8 ile verilmiştir.

$$P_{best} = \{P_i^{[1]}, P_i^{[2]}, P_i^{[3]}, P_i^{[4]}\} \quad (6.8)$$

Denklem 6.8’de $P_i^{[1]}$ birinci katman için o ana kadar bulunan en iyi ağırlıkları göstermektedir. Popülasyondaki parçacıklar arasından en iyi parçacığın pozisyonu denklem 6.9 ile verilmiştir.

$$P_{gbest} = \{P_{gbest}^{[1]}, P_{gbest}^{[2]}, P_{gbest}^{[3]}, P_{gbest}^{[4]}\} \quad (6.9)$$

Adım 2: Her parçacığın o ana kadarki en iyi pozisyonunu ($P_i^{[j]}$) belirle.

Her bir parçacığın pozisyonu ile ilgili parçacığın o ana kadarki en iyi pozisyonunu karşılaştır. Eğer daha iyi bir pozisyon bulunmuş ise en iyi pozisyonu yenisi ile değiştir.

Adım 3: Bütün parçacıkların o ana kadarki en iyi pozisyonları içerisinde en iyi parçacığı belirle.

Adım 4: Popülasyonun en iyi parçacığını kullanarak yapay sinir ağının çıkış hatasını bul.

Bu adımda yapay sinir ağının ağırlıkları en iyi parçacığın pozisyonundan seçilerek bütün eğitim veri kümesi için ağ çıkış hatası bulunur. Bu hata değeri aynı zamanda her bir parçacığın uygunluk fonksiyonunu gösterir. Yapay sinir ağının eğitim hatası ortalama kareler hatası olarak seçilmiştir. Dolayısıyla herhangi bir i parçacığının uygunluğu denklem 6.10’a göre belirlenmektedir.

$$f(W_i) = \frac{1}{S} \sum_{k=1}^S \left[\sum_{i=1}^N (t_{ki} - o_{ki}(W_i))^2 \right] \quad (6.10)$$

Denklem 6.10’da f uygunluk fonksiyonunun değeri, t_{ki} hedef çıkışı, o_{ki} W_i kullanılarak elde edilen çıkışı gösterir. S değeri eğitim kümesindeki örnek sayısını gösterirken N değeri ise ağı çıkış sayısını gösterir.

Adım 5: Parçacıkların hız ve pozisyonunu güncelle.

Her bir parçacığın hızı denklem 6.11 pozisyonu ise denklem 6.12 kullanılarak güncellenmektedir.

$$V_i^{[j]} = \kappa V_i^{[j]} + c_1 w_1 (P_i^{[j]} - W_i^{[j]}) + c_2 w_2 (P_{gbest}^{[j]} - W_i^{[j]}) \quad (6.12)$$

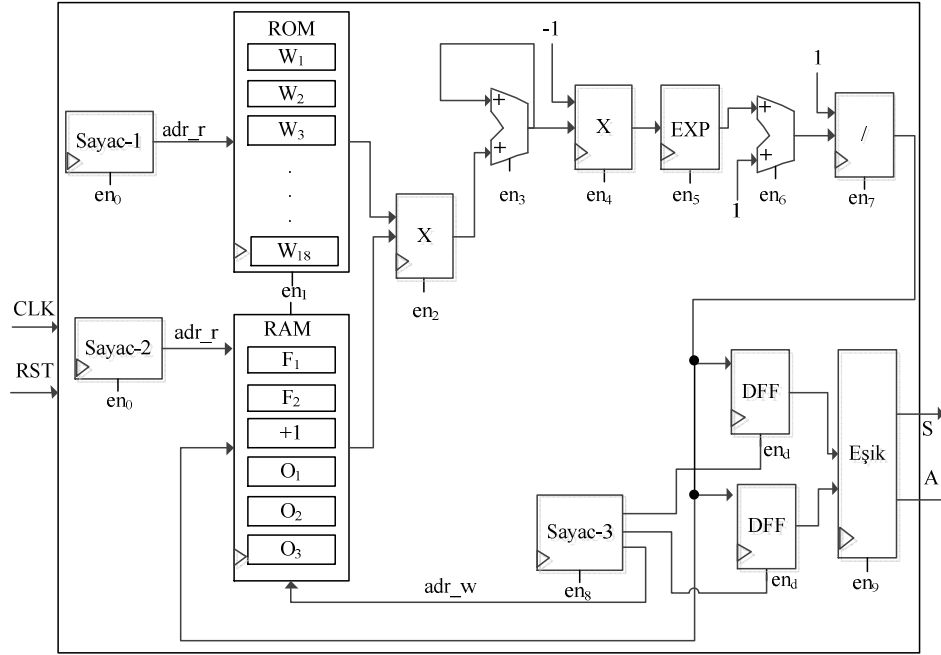
$$W_i^{[j]} = W_i^{[j]} + V_i^{[j]} \quad (6.13)$$

Denklem 6.12’de kullanılan $V_i^{[j]}$ i. parçacığın j. Katmandaki hız vektörünü gösterir. Denklemde kullanılan c_1 ve c_2 parametreleri sosyal hızlandırma parametreleri, w_1 ve w_2 ise $[0, 1]$ aralığında rastgele sayılardır. Denklemde kullanılan κ parametresi parçacığın geçerli hızının bir sonraki hız üzerindeki etkisini kontrol eden moment parametresidir.

Adım 6: Eğer maksimum adım sayısına ulaşılmamış ise Adım 2’ye git. Aksi takdirde programı sonlandır.

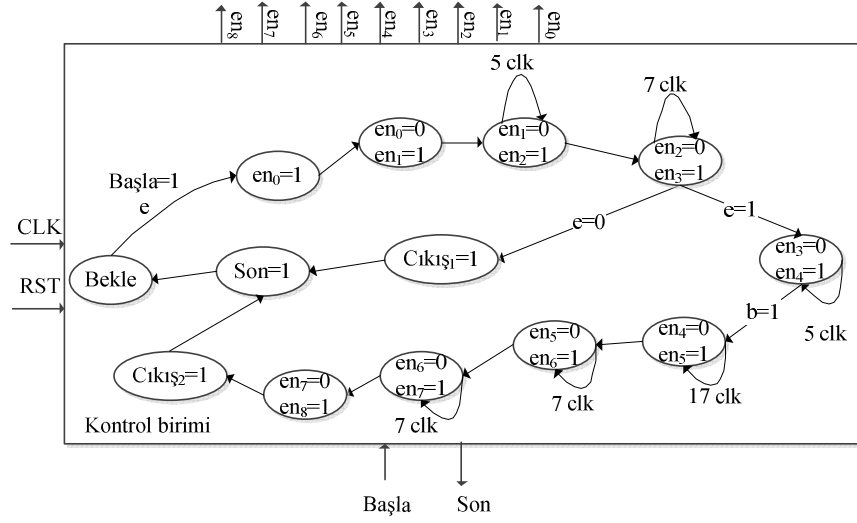
6.4.2. Kaskad Yapay Sinir Ağı’nın FPGA’da Gerçekleştirilmesi

Yapay sinir ağı PSO ile eğitilip ağırlıklar elde edildikten sonra test işlemi FPGA üzerinde gerçekleştirilebilir. Arıza teşhisi için önerilen kaskad yapay sinir ağı iki giriş ve iki çıkışa sahiptir. İki boyutlu faz uzayındaki her bir boyut yapay sinir ağı’nın bir girişini oluşturur. 4 katmanlı kaskad yapay sinir ağı 18 ağırlığa sahiptir. Bütün katmanlarda tek yönlü sigmoid aktivasyon fonksiyonu kullanılmıştır. Her katmanın çıkışı hesaplanırken o katman ile ilgili bir giriş değeri ile ilgili girişe ait ağırlık çarpılır. Nörona gelen giriş sayısı kadar çarpılan değerler üst üste toplanır ve hesaplanan çıkış sigmoid aktivasyon fonksiyonundan geçirilir. Elde edilen çıkış aynı zamanda bir sonraki nöronun girişi olduğundan RAM’e yazılır. FPGA’daki blokların çalışma sıraları sonlu durum makinesi ile kontrol edilmektedir. Şekil 6.32’de önerilen yöntemin FPGA blok şeması verilmiştir.



Şekil 6.32. Kaskad yapay sinir ağının FPGA blok şeması

Şekil 6.32’de verilen FPGA blok şeması tek bir nöronun çıkışı gibi görünmektedir. Yapay sinir ağı ağırlıklarının tutulduğu ROM’un adresi sayac-1 ile kontrol edilmektedir. ROM’dan okunan ağırlık ile RAM’den okunan giriş değeri çarpılmaktadır. Elde edilen çarpımların sonucu üst üste toplanmaktadır. İlgili katmanda üç giriş var ise toplam bloğuna kadar olan kısım üç kez çalıştırılmaktadır. Daha sonra elde edilen toplam tek yönlü sigmoid aktivasyon fonksiyonundan geçirilmektedir. Çıkış nöronu dışında elde edilen sonuçlar bir sonraki nöronun girişini oluşturmak için RAM’e yazılmaktadır. RAM’in yazma adresi Sayac-3 ile kontrol edilmektedir. Sayac-3 aynı zamanda yapay sinir ağının iki çıkışını da kontrol etmek için kullanılmaktadır. Eğer çıkış üçüncü nöron için yapılıyorsa hem RAM’e hem de flip-flop’a yazılmaktadır. Dördüncü nörona ise sadece flip-flop’a çıkış yazılmaktadır. Sayac-1 ilk nöronun çıkışını hesaplarken ROM’dan ilk üç ağırlığı okumakta ikinci nöronun çıkışında ise dördüncü nörona itibaren dört ağırlık okunmaktadır. Gerçekleştirilen tasarım tamamen 32-bit IEEE-754 kayan noktalı sayı formatına göre çalışır. Dolayısıyla her bir blok belirli sayıda clock işareti sonrasında çıkış üretir. Blokların çalışma süreleri VHDL’de yazılan sonlu durum makinası ile kontrol edilmektedir. Kontrol bloğunun şeması Şekil 6.33’te verilmiştir.



Şekil 6.33. Kaskad yapay sinir ağının VHDL ile yazılan kontrol birimi

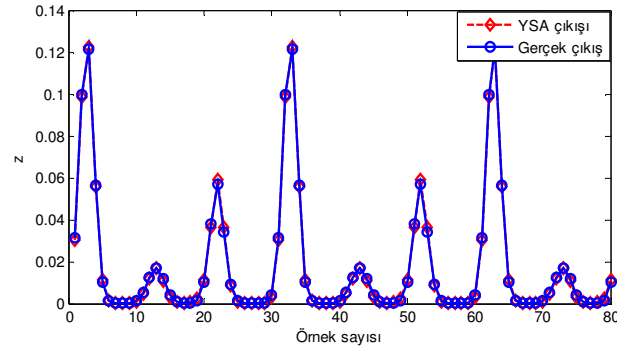
Şekil 6.33'te verilen VHDL'de yazılmış sonlu durum makinası ile yapay sinir ağındaki blokların çalışma sırası belirlenmektedir. Durum makinasındaki e değeri çıkışı hesaplanan nörona aktivasyon fonksiyonunun uygulanıp uygulanmayacağını belirler. Örneğin birinci nöronun çıkışı hesaplanırken eğer toplama işlemi üç kez yapılmamış ise $e=0$ olup başa dönülmekte ve sonraki ağırlık ile giriş okunup aynı süreç devam ettirilmektedir. Eğer $e=1$ ise bloğun geri kalan kısmı aktif olmakta ve nöronun çıkışı hesaplanmaktadır.

6.4.3. Deneysel Sonuçlar

Önerilen kaskad yapay sinir ağı doğrusal olmayan sınıflandırma ve fonksiyon yaklaştırma problemlerini çözebilmektedir. Önerilen yapay sinir ağı yapısı doğrusal olmayan bir fonksiyonu öğrenerek onu tahmin edebilmektedir. Bu amaçla denklem 6.14'teki problem göz önüne alınmıştır.

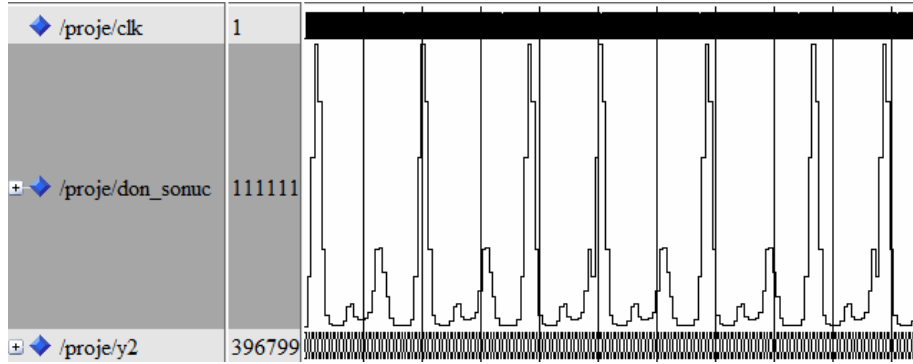
$$z = 4 \exp(-0.15(x-4)^2 - 0.5(y-3)^2) + 10^{-9} \quad (6.14)$$

Önerilen yapay sinir ağı x ve y girişlerini alarak z çıkışını hesaplamaktadır. Yapay sinir ağı bu fonksiyonu minimum hata oranı ile öğrenebilmektedir. Şekil 6.34'te ilgili fonksiyonun gerçek çıkışı ve yapay sinir ağı çıkışı verilmiştir.



Şekil 6.34. Gerçek çıkış ve yapay sinir ağı çıkışı

Şekil 6.34'te görüldüğü gibi önerilen yapay sinir ağı doğrusal olmayan fonksiyonu doğru bir şekilde öğrenebilmektedir. Önerilen kaskad yapay sinir ağı FPGA üzerinde test edilmiştir. Fonksiyon yaklaştırma için kullanılan yapay sinir ağının modelsim benzetim sonuçları Şekil 6.35'te verilmiştir.



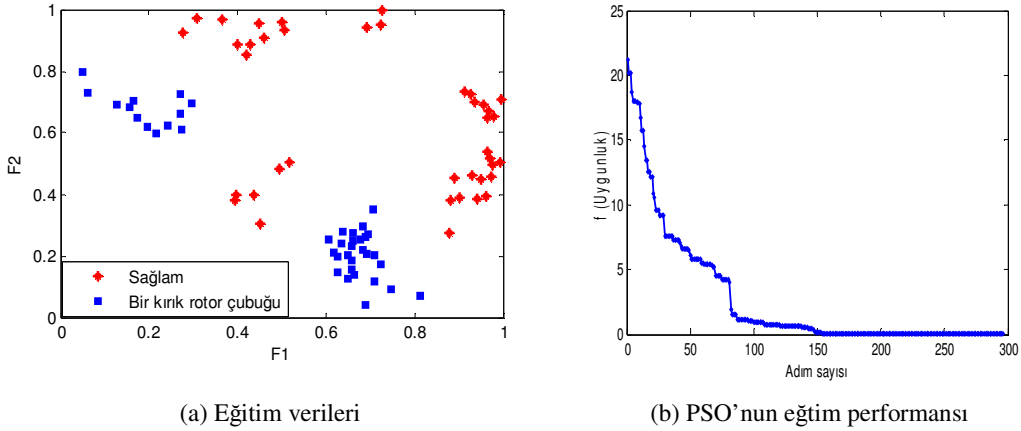
Şekil 6.35. Fonksiyon yaklaştırma için kaskad yapay sinir ağının benzetim sonuçları

Kaskad yapay sinir ağı dördüncü bölümde DVM'nin eğitimi için kullanılan veriler üzerinde test edilerek sağlam ve bir kırık rotor çubuğu arızası sınıflandırılmıştır. Eğitim için kullanılan PSO'nun parametreleri Tablo 6.3'te verilmiştir.

Tablo 6.3. Kaskad yapay sinir ağı için kullanılan PSO'nun parametreleri

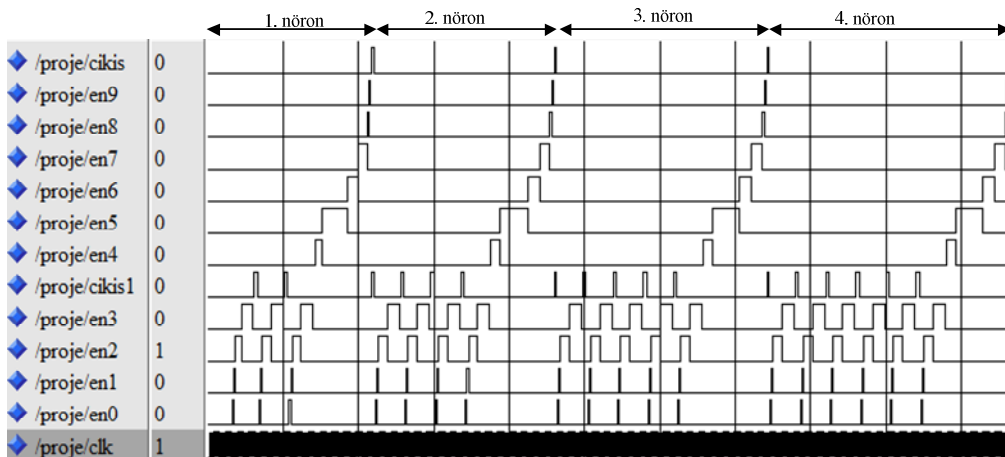
Özellik	Değer
c_1	2.05
c_2	2.05
κ	0.9-0.4
Adım sayısı	400
Popülasyon boyutu	30

Tablo 6.3'te verilen PSO parametrelerinden κ değeri 0.9'dan 0.4'e kadar adım sayısı ile orantılı olarak düşürülmektedir. Şekil 6.36'da sağlam ve arızalı sınıf verileri ile PSO'nun bu veriler üzerinde performansı verilmiştir.



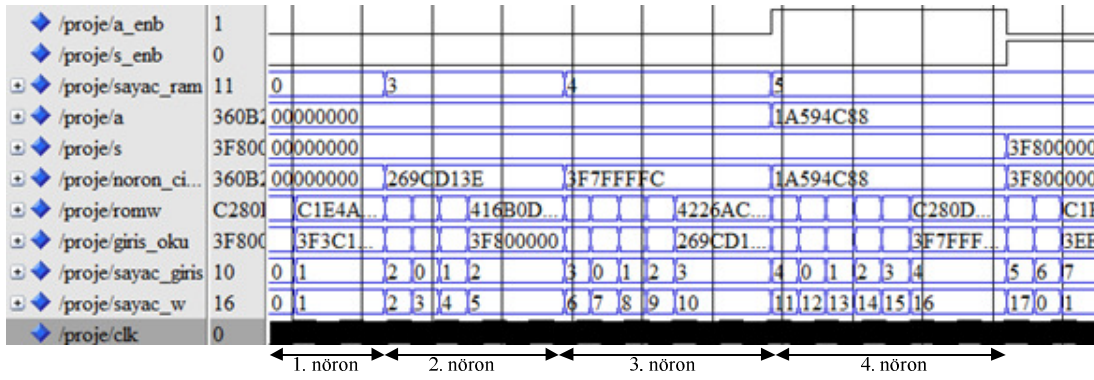
Şekil 6.36. Kaskad yapay sinir ağı eğitim verileri ve eğitim hatasının değişimi

Şekil 6.36 (a)'da görüldüğü gibi veriler doğrusal olmayan bir şekilde dağılım göstermektedir. Verilerin sınıflandırılması için PSO'nun eğitilmesi sonucu Şekil 6.36 (b)'deki hata performansı elde edilmiştir. Sınıflandırma hatası 150 adım sonunda sıfıra düşmüştür. FPGA'da gerçekleştirilen yapay sinir ağının kontrolü için kullanılan sonlu durum makinasının Modelsim benzetim sonucu Şekil 6.37'de verilmiştir.

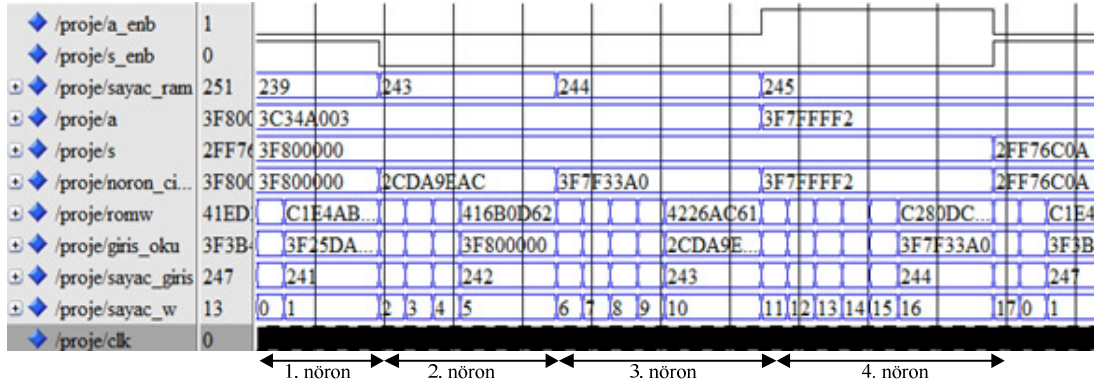


Şekil 6.37. Kontrol için kullanılan sonlu durum makinasının benzetim sonuçları

Şekil 6.37’de çalışması gösterilen sonlu durum makinasında cikis1’e kadar olan kısımlar her bir nörondaki net giriş hesaplamaktadır. İlk nöronda girişler için üç toplam ve üç çarpma işlemi yapıldıktan sonra elde edilen sonuç aktivasyon fonksiyonuna verilerek nöron çıkışı bulunmaktadır. Bu işlem ikinci nöronda dört kez, üçüncüsünde beş ve dördüncüsünde ise altı kez yapılmaktadır. Her bir nöron için sabit FPGA kaynakları kullanıldığından kaynak kullanımı minimuma indirgenmiştir. Şekil 6.38’de kaskad yapay sinir ağının sağlam ve kırık rotor çubuğu arızası için Modelsim benzetim sonuçları verilmiştir.



(a) Sağlam



(b) Bir kırık rotor çubuğu

Şekil 6.38. Kaskad yapay sinir ağının arıza teşhisi için Modelsim benzetim sonuçları

Şekil 6.38’de verilen benzetim sonuçlarında her adım için ilgili nöronun çıkışı, ROM’dan okunan değer ve adresi, okunan ağ girişleri ve adresleri, bir sonraki nöronun giriş olarak yazılacağı RAM adresi ve yapay sinir ağının çıkışları verilmiştir. Yapay sinir ağı ilk nöron çıkışını hesaplarken üç ağırlık ve giriş değeri okumaktadır. İkinci nöronda

dört, üçüncüsünde beş ve dördüncüsünde altı giriş kullanılarak nöron çıkışı hesaplanmaktadır. Her bir nöron için hesaplanan çıkış bir sonraki nöronun girişini oluşturmak için RAM'e kaydedilmektedir. Örneğin Şekil 6.38 (a)'da birinci nöron için hesaplanan "269CD13E" çıkış değeri bir sonraki nöronun girişi olarak alınmıştır. İlk üç nöronun çıkışı kendisinden sonraki nöron girişini oluşturmak için RAM'e yazılırken son nöron çıkışı RAM'e yazılmamaktadır. Ayrıca yapay sinir ağında üçüncü ve dördüncü nöronların çıkışı hesaplandığında ilgili yapay sinir ağı çıkışları aktifleştirilerek çıkış verilmektedir.

Önerilen kaskad yapay sinir ağı minimum sayıda FPGA kaynaklarını kullanır. Çünkü önerilen model tek bir nöron çıkışı için gerekli olan hesaplamaları yapmaktadır. Her nöron için sadece giriş sayısı ve bu girişler ile ilgili ağırlıklar değişir. Tablo 6.4'ten kaskad yapay sinir ağı için kullanılan FPGA kaynakları verilmiştir.

Tablo 6.4. Kaskad yapay sinir ağı için kullanılan FPGA kaynakları

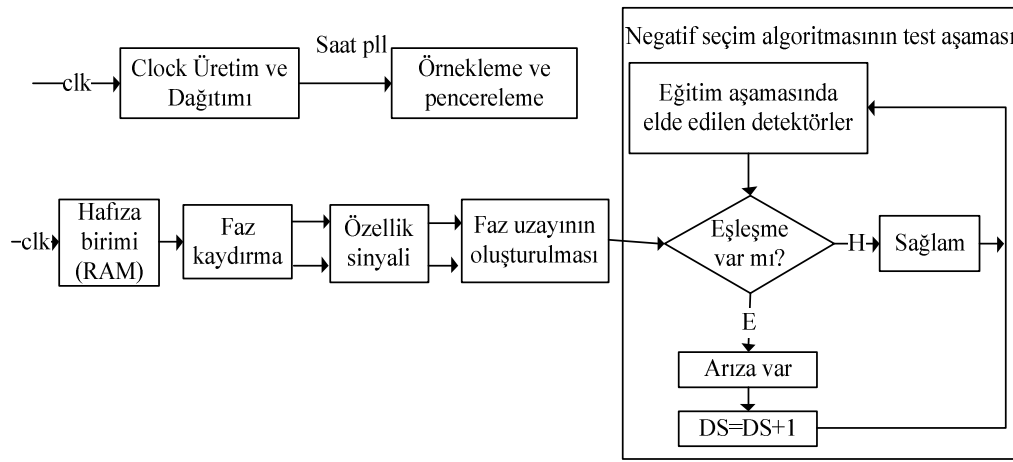
Hafıza birimi	Kullanım oranı
Lojik elemanlar	3,840 / 119,088 (% 3)
Bileşimsel lojik fonksiyonlar	3,365 / 119,088 (% 3)
Kullanılan lojik kayıtlar	2,047 / 119,088 (%2)
Toplam hafıza bitleri	22,361 / 3,981,312 (< %1)
Gömülü çarpıcılar	61 / 576 (%11)

Yapay sinir ağlarında bir nöronun çıkışı hesaplanırken FPGA üzerinde oldukça fazla kaynak tüketimine neden olan çarpma, toplama, üs alma ve bölme gibi aritmetik işlemler yoğun bir şekilde kullanılmaktadır. FPGA üzerinde kayan noktalı sayı tabanlı aritmetik işlemler sınırlı sayıda gömülü çarpıcıları tüketirler. Burada önerilen kaskad yapay sinir ağı modeli Tablo 6.4'ten de görüldüğü gibi gömülü çarpıcıların %11'ini harcamaktadır. Bütün işlemler seri bir şekilde yapılmasına rağmen yapay sinir ağı çok az nöron kullandığından hızlı sonuç vermektedir.

6.5. Gerçek Zamanlı Arıza Tespiti için Negatif Seçim Algoritması

Arızaların otomatik olarak belirlenmesi için negatif seçim algoritmasından faydalanılmıştır. Herhangi bir sistemin negatif seçim algoritması ile gerçek zamanlı olarak izlenebilmesi ve arıza tespiti için problemin öğrenilmesi gerekir. Negatif seçim algoritması sağlam veri örneklerini kullanarak problemi öğrenir ve arızalı durumları tespit edebilen detektörler üretir. Üretilen detektörler test aşamasında kullanılarak arızalar belirlenebilir.

Negatif seçim algoritmasının detektörleri genetik algoritma ile MATLAB ortamında optimize edilmektedir. Detektör üretimi için kullanılan eğitim algoritması üçüncü bölümde verilmiştir. Önerilen negatif seçim algoritması özellik çıkarımı ve tespit aşamalarından oluşur. Arıza ile ilgili özellikler tek faz akım sinyalinin elde edilmektedir. Faz akım sinyalinin orijinal veri örnekleri ile bu örneklerin 90 derecelik faz kaydırması ile elde edilen bileşenler özellik sinyalini oluşturmak için kullanılmaktadır. Elde edilen özellik sinyalinin faz uzayı oluşturularak eğitim aşamasında elde edilen detektörlere göre arızalar belirlenmektedir. Önerilen yöntemin blok diyagramı Şekil 6.39’da verilmiştir.



Şekil 6.39. Önerilen negatif seçim tabanlı gerçek zamanlı arıza tespit yöntemi

Şekil 6.39’da verilen gerçek zamanlı arıza tespit algoritmasında saat sinyalinin üretimi için 50 MHZ ‘lık bir PLL kullanılmaktadır. RAM’den akım sinyallerini okumak amacıyla VHDL’de yazılan örnekleme ve pencere fonksiyonlarından faydalanılır. RAM’den faz kaydırması için iki adres okunarak özellik sinyali elde edilir. Elde edilen özellik sinyalinin faz uzayı daha önce elde edilen detektörler ile karşılaştırılarak en az bir detektör ile eşleşme var ise detektör sayısı bir arttırılır.

6.5.1. Özellik Çıkarımı ve Arıza tespiti

Önerilen arıza tespit sisteminin ilk aşaması özellik çıkarımıdır. Akım sinyalleri bir RAM’den okunmaktadır. RAM’den akım sinyallerini okumak ve faz kaydırmasını gerçekleştirmek için VHDL’de yazılan pencere fonksiyonu kullanılmaktadır. Yazılan VHDL kodunun bir kısmı Şekil 6.40’de verilmiştir.

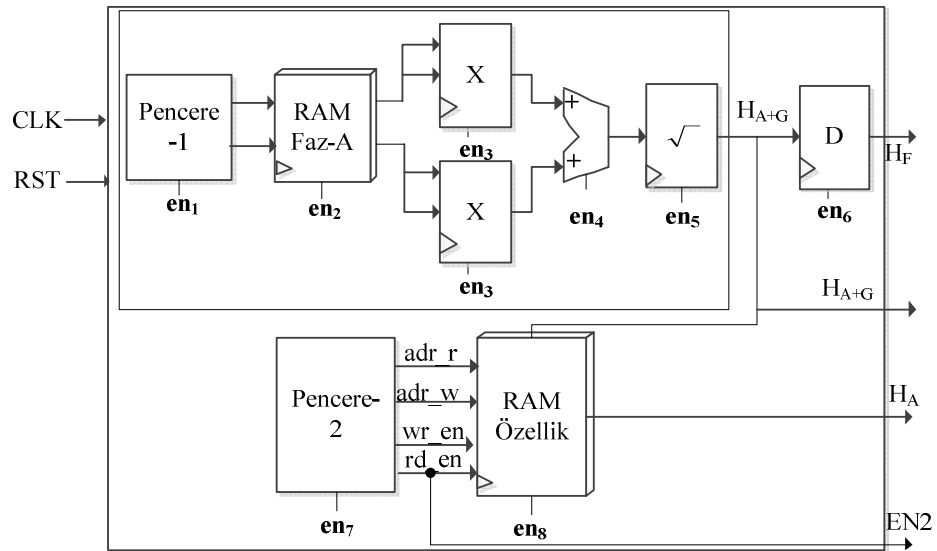
```

If (clk'event and clk='1') then
  If(enb='1') then
    adr1 := adr1 + 1;
    adr2:=adr2+1;
    If(adr1=pen+1410) then
      pen:=pen+100;
      adr1:=pen;
      adr2:=pen+89;
    End if;
  End if;
End if;

```

Şekil 6.40. Pencere fonksiyonu ve faz kaydırma için VHDL kodu

Şekil 6.40'ta adr1 değişkeni RAM'den okunacak birinci adresi adr2 ise faz kaydırma için kullanılacak ikinci adresi gösterir. Her bir pencere boyutu 1410 adet örnekten oluşmaktadır. Pencere boyutuna ulaşıldığında pen değişkeni 100 arttırılarak bir sonraki pencere için veriler okunmaktadır. RAM'den okunan ve aralarında faz farkı olan iki örnek alındıktan sonra özellik sinyali oluşturulur. Elde edilen özellik sinyalinin her bir örneği zaman gecikmeli faz uzayını oluşturmak için ikinci bir pencere örneği ile başka bir RAM'e yazılmaktadır. Önerilen özellik çıkarım yönteminin blok şeması Şekil 6.41'de verilmiştir.

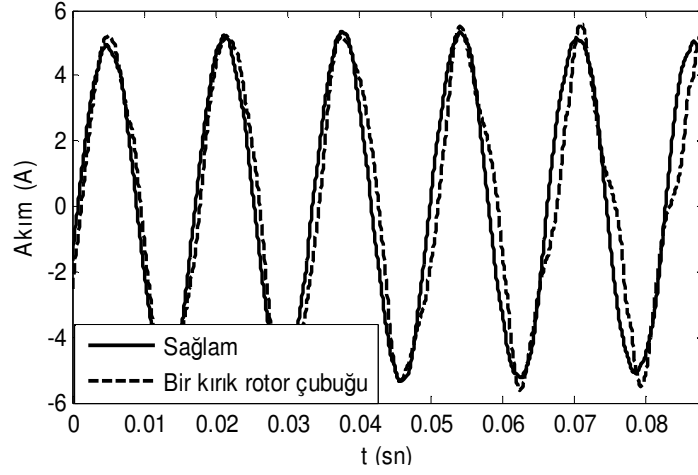


Şekil 6.41. Özellik sinyalinin elde edilmesi

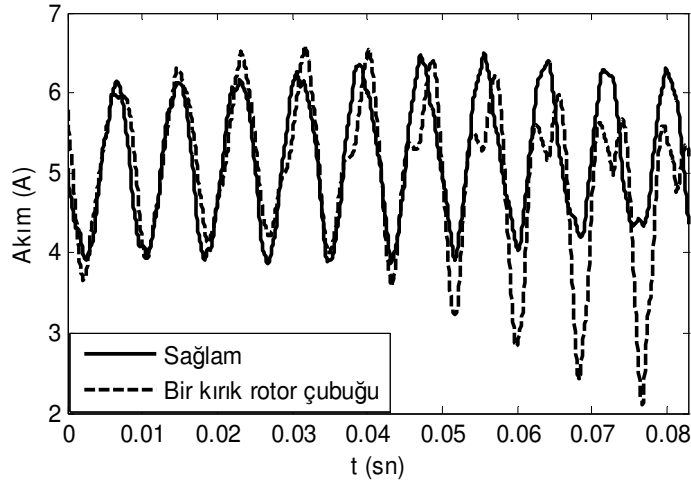
her bir detektörün koordinatlarını ve yarıçapını RAM'den okuyarak gelen test veri örneğini geçerli detektör ile karşılaştırmak ve aktifleşen detektörlere göre arızayı tespit etmektir. Bu işlem yukarıdaki bloğun seri bir şekilde çalıştırılması dolayısıyla on detektör varsa bu işlemin seri bir şekilde on kez yapılmasını gerektirir. Böyle bir tasarımın avantajı daha az kaynak kullanımı sağlamasıdır. Fakat çıkış daha uzun süreli saat gecikmesi gerektirir. İkinci yöntem ise aynı anda çalışacak paralel blokların kullanımıdır. Dolayısıyla gelen test verisi aynı anda bütün detektörler ile karşılaştırılır. Eğer herhangi bir detektör eşleşme tespit ederse SAY bloğu tespit edilen örnek sayısını bir artırır. Bu yöntem daha fazla FPGA kaynağı kullanmasına rağmen daha hızlı çalışır. Ayrıca faz uzayı için bir örnek veri negatif seçim algoritmasının tespit aşamasına sunulduğunda sonuç hesaplanana kadar faz uzayının diğer verisi elde edilmektedir.

6.5.2. Deneysel Sonuçlar

Önerilen yöntemin doğruluğu üç fazlı bir asenkron motorun tek fazından alınan akım sinyalleri kullanılarak doğrulanmıştır. Sağlam ve kırık rotor çubuğu arızasına sahip bir asenkron motorun tek fazından alınan akım sinyalleri ile özellik sinyali elde edilmektedir. FPGA uygulaması Quartus II 10.1 ortamında gerçekleştirilmiş olup benzetim sonuçları Modelsim ile elde edilmiştir. Şekil 6.43'te sağlam ve kırık rotor çubuğu arızası için akım sinyalleri ve özellik sinyalleri verilmiştir.



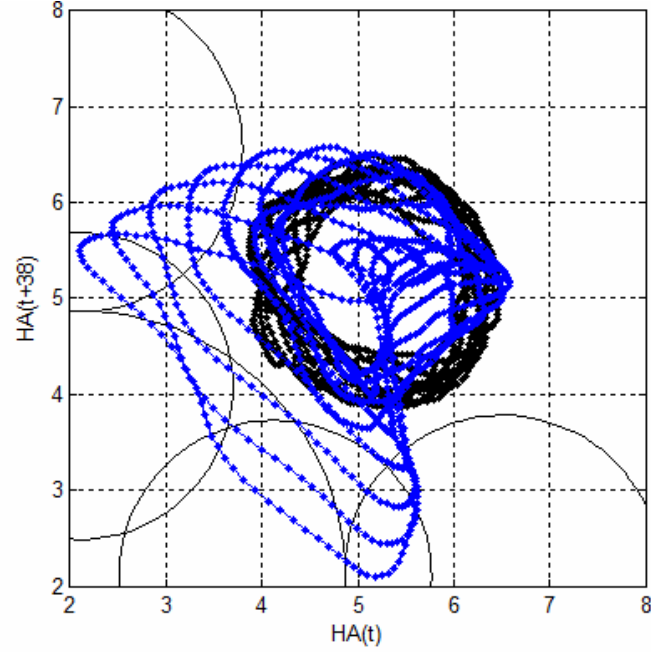
(a) Akım sinyalleri



(b) Özellik sinyalleri

Şekil 6.43. Sağlam ve arızalı durum için akım ve özellik sinyalleri

Orijinal sinyal ve bu sinyalin 90 derecelik bir faz kaydırması ile oluşturulan ikinci sinyalin kareleri toplanmakta ve toplamın karekökü alınarak özellik sinyali oluşturulmaktadır. Faz kaydırması oluşturulurken her bir periyottaki örnek sayısı bilinmelidir. Bu özellik denklem 5.2 kullanılarak hesaplanmaktadır. Özellik sinyali elde edildikten sonra faz uzayı oluşturularak detektörler üretilir. Faz uzayı için zaman gecikmesi 38 ve gömülme boyutu iki alınmıştır. Detektör üretim aşaması üçüncü bölümde önerilen genetik algoritma ile yapılmaktadır. Sağlam veri örnekleri alınarak üretilen detektörler Şekil 6.44'te verilmiştir.



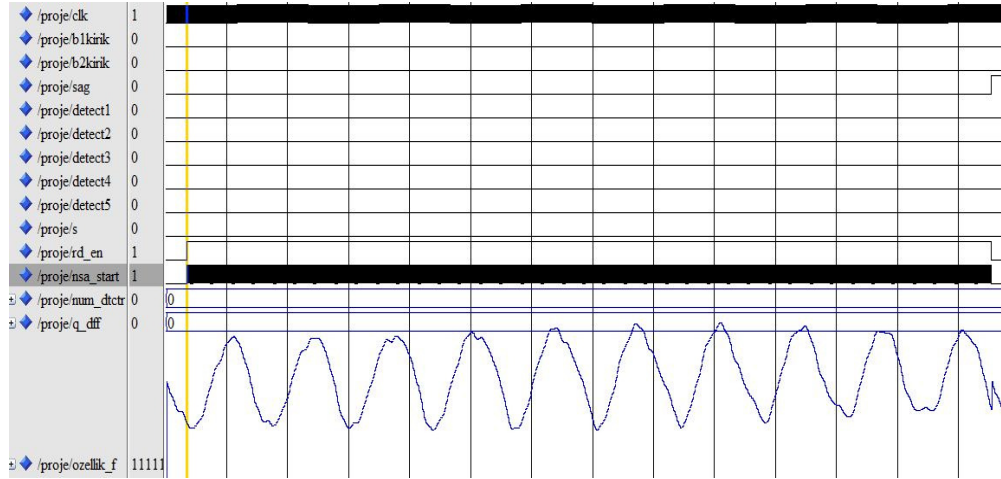
Şekil 6.44. Sağlam ve arızalı faz uzayları ve üretilen detektörler

MATLAB ortamında genetik algoritma ile optimize edilmiş detektörlerden öz olmayan uzayı en iyi kapsayanlar test için kullanılacaktır. Önerilen yöntem Cyclone III FPGA geliştirme kitinin çok az kaynağını kullanır. Gerçekleştirilen tasarımın kullandığı FPGA kaynaklarının oranı Tablo 6.5'te verilmiştir.

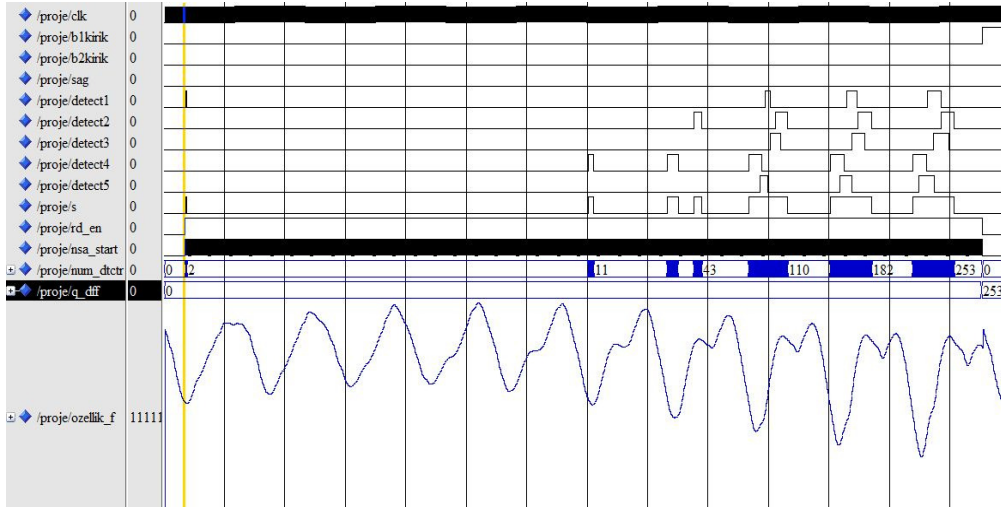
Tablo 6.5. Negatif seçim algoritması için kullanılan FPGA kaynaklarının oranı

Hafıza birimi	Kullanım oranı
Lojik elemanlar	18.269 / 119.088 (15 %)
Bileşimsel lojik fonksiyonlar	17.425 / 119.088 (15 %)
Kullanılan lojik kayıtlar	9.933 / 119.088 (8 %)
Toplam hafıza bitleri	109.814 / 3.981,312 (3 %)
Gömülü çarpıcılar	84 / 576 (15 %)

Arıza tespiti için gerçekleştirilen tasarımın benzetimi Altera Modelsim 6.5'te gerçekleştirilmiştir. Şekil 6.45'de sağlam ve arızalı motor için Modelsim benzetim sonuçları verilmiştir.



(a) Sağlam



(b) Kırık rotor çubuğu arızası

Şekil 6.45. Sağlam ve arızalı motor için Modelsim benzetim sonuçları

Şekil 6.45'te negatif seçim algoritmasına sunulan faz uzayının bir örneği için detektörlerden en az biri aktifleştirse tespit edilen örnek sayısı (num_dtctr) arttırılmaktadır. Sağlam test örnekleri için Şekil 6.36 (a)'dan da görüldüğü üzere herhangi bir detektör aktifleşmemiştir. Arızalı test örnekleri için ise başlangıçta özellik sinyalinde çok bozulma olmadığı için daha az örnek tespit edilmiştir. Fakat bozulma olduğu andan itibaren detektörler tarafından tespit edilen örnek sayısı artmıştır.

6.6. Bölüm Değerlendirmesi ve Sonuçlar

Tezin bu bölümünde arıza tahmin, tespit ve teşhisi için gerçek zamanlı yöntemler sunulmuştur. Önerilen yöntemler FPGA kartına yüklenerek çalıştırılabilmektedir. Stator

arızalarının teşhisi için bulanık mantık tabanlı bir yöntem sunulmuştur. Üç faz akım sinyalinin büyüklükleri stator durumunu belirlemek için bulanık mantık modülüne verilmekte ve motor durumu belirlenmektedir. Bulanık mantık modülü VHDL dilinde yazılmıştır. Üç faz akım sinyallerinden bulanık girişlerin elde edilmesi IEEE-754 kayan noktalı sayı formatına göre yapılmıştır. Bulanık modül ise VHDL’de yazıldığından tamsayı tabanlı çalışmaktadır. Dolayısıyla giriş verileri normalize edilerek tam sayıya dönüştürülmüştür. Bulanık mantığın aktifleşen çıkış üyelik fonksiyonuna göre motor durumu ile ilgili dilsel ifade LCD displayde gösterilmektedir. İkinci yöntem arıza tahmini için ileri beslemeli yapay sinir ağını kullanmaktadır. Yapay sinir ağının eğitimi Matlab ortamında gerçekleştirilerek ağırlıklar elde edilmektedir. Yapay sinir ağının girişleri faz akım sinyalinin Hilbert dönüşümü ile elde edilen özellik sinyalinin ortalaması ve standart sapmasıdır. Önerilen yöntemde amaç hem arızalı durumları tespit etmek hem de geçiş anında arızayı belirlemektir. Arızalı durumda sürekli çalışırken ortalama büyük çıkarken standart sapma düşmektedir. Geçiş anında ise standart sapma değişirken ortalama daha düşük çıkmaktadır. Yapay sinir ağı sinyalin bu özelliğini modelleyerek arızaları tahmin eder. Üçüncü yöntem arıza teşhisi için kaskad yapay sinir ağından faydalanmaktadır. Kaskad yapay sinir ağı giriş ile çıkışlar arasındaki doğrusal olmayan ilişkiyi az sayıda nöron ile öğrenebilmektedir. Kaskad yapay sinir ağı FPGA üzerinde seri bir şekilde gerçekleştirildiğinden dolayı her bir nöronun çıkışı hesaplanırken bir nöron için gerekli olan işlemler gerçekleştirilmektedir. Dolayısıyla FPGA kaynaklarının minimum kullanımı sağlanmıştır. Ayrıca kırk rotor çubuğu arızalarının teşhisi için önerilen özellik çıkarımı farklı arızalı durumları ayırt etmek için kullanışlıdır. Dördüncü yöntem ise tek faz akım sinyalinin belirlenen faz kadar kaydırılması ile oluşturulan özellik sinyalinin faz uzayını oluşturmakta ve negatif seçim algoritması ile arızaları belirlemektedir. Okunan akım sinyallerinden özellik çıkarımı ve negatif seçim algoritması ile arıza teşhisinin bütün aşamaları FPGA üzerinde 32-bit kayan noktalı sayı formatı ile gerçekleştirilmiştir. Önerilen yöntemin performansı hem benzetim hem de deneysel olarak doğrulanmıştır.

7. SONUÇLAR

Bu tezde durum izleme, arıza tespiti ve teşhisi için akıllı hesaplama tekniklerine dayalı yöntemler sunularak bu yöntemlerin performansının geliştirilmesi için algoritmalar önerilmiştir. Arıza durumlarının daha az veri kullanan ve tespiti kolaylaştırıcı özellikler ile saptanması için yeni özellik çıkarım yöntemleri sunulmuştur. Ayrıca durum izleme için FPGA tabanlı paralel çalışabilen gerçek zamanlı algoritmalar geliştirilmiştir. Arıza tespiti için Hilbert dönüşümü tabanlı yeni bir özellik sinyali elde edilmiştir. Bu özellik sinyalinin faz uzayı sağlam ve arızalı durumları ayırt etmek için oldukça kullanışlıdır. Aynı veriler üzerinde daha önce kullanılan bir zaman serisi yöntemi olan ilk fark filtrelemesi ile karşılaştırıldığında önerilen özellik çıkarımının daha iyi olduğu gerçekleştirilen deneysel sonuçlar ile kanıtlanmıştır. Arıza tespiti için yapay bağışık sistem bileşenlerinden negatif seçim algoritması kullanılmıştır. Orijinal negatif seçim algoritması iki temel probleme sahiptir: öz olmayan uzayın tam kapsanamaması ve detektörlerin örtüşmesi. Arıza tespiti için önerilen negatif seçim algoritmasının detektörleri genetik algoritma ile optimize edilerek öz olmayan uzayın maksimum kapsanması sağlanmıştır. Genetik algoritma sağlam durum faz uzayını kullanarak genetik algoritma ile her adımda bir detektör üretmektedir. Üretilen detektör aynı zamanda sağlam verilere eklenerek bir sonraki adımda aynı detektörün üretimi engellenmiştir. Önerilen uygunluk fonksiyonu, minimum sayıda detektör ile öz olmayan uzayı maksimum bir şekilde kapsamaktadır. Farklı arıza durumları için yöntemin performansı değerlendirilerek iyi sonuçlar verdiği gözlemlenmiştir.

Motor akım imza analizi ile tek faz akım sinyali üzerinden elde edilen yan bant bileşenleri kullanılarak rotor arızaları tespit edilmiştir. İki sınıflı arıza tespit edici olarak negatif seçim algoritmasından faydalanılmıştır. Negatif seçim algoritmasının eğitim aşamasında kaotik haritalar ve klonal seçim algoritmasından faydalanılmıştır. Orijinal negatif seçim algoritmasında sadece sağlam veriler alınarak detektörler üretilmektedir. Önerilen yöntemde hem sağlam hem de arızalı veriler için detektörler üretilmiştir. Üretilen detektörler öz ve öz olmayan verileri maksimum şekilde kapsamaktadır. Dolayısıyla her detektörün aynı zamanda bir sınıfı vardır. Klonal seçim algoritmasında başlangıç popülasyonu ve mutasyon işlemi için kaotik haritalardan faydalanılmıştır. Negatif seçim algoritmasının test aşamasında da yeni bir yaklaşım sunulmuştur. Önerilen detektörler öz ve öz olmayan verileri kapsadığından bu veriler dışında gelen bir test örneği mevcut

detektörler ile tespit edilemeyebilir. Yeni örnekleri tespit etmek amacıyla K- en yakın komşu algoritmasından faydalanılmıştır. Test aşamasında gelen bir örneğe en yakın detektörün sınıfı bulunarak bu veri üzerinde bir detektör oluşturulmaktadır. Bu detektörün yarıçapı diğer ait olduğu sınıf dışından veri kapsamayana kadar küçültülmektedir. Aynı veriler üzerinde yapay sinir ağları ve bulanık mantık ile karşılaştırılması sonucu önerilen yöntemin bu yöntemlerle eşdeğer düzeyde bir performansa sahip olduğu görülmüştür.

Üç faz akım sinyali kullanılarak yeni bir özellik sinyali elde edilmiştir. Bu özellik sinyali üç faz akım sinyalinin davranışını temsil etmektedir. Farklı durumlar için elde edilen özellik sinyallerinin faz uzayları arızalı ve sağlam durumları kolayca ayırt edebilecek şekilde oluşmaktadır. İki sınıflı bir sınıflandırıcı olan destek vektör makinalar tabanlı bir arıza teşhis edici önerilmiştir. Destek vektör makinalardaki en önemli problem çekirdek ve ceza parametrelerinin seçilmesidir. Bu iki parametrenin seçimi çok amaçlı klonal seçim optimizasyon yöntemi ile belirlenmiştir. Önerilen amaç fonksiyonu hem destek vektör sayısını minimum yapmakta hem de sınıflandırma başarımını maksimum yapmaktadır. Destek vektörlerinin minimum yapılması test aşamasını hızlandırmaktadır. Destek vektör makinaları ile çok sınıflı sınıflandırma yapabilmek için birine karşı diğeri yönteminden faydalanılmıştır.

Kırık rotor çubuğu arızalarının teşhisi ve sınıflandırılması için PSO tabanlı bir bağışık sınıflandırıcı önerilmiştir. Önerilen yöntem arıza teşhisi için elde edilen özellik sinyalinden çarpıklık, basıklık, enerji ve maksimum değer gibi istatistiksel özelliklerin yanında yan bant frekansının genliğini de elde etmektedir. Bu sinyaller farklı arızalı ve sağlam durumları ayırt etmek için oldukça kullanışlıdır. Önerilen SÖTYBS sınıflandırıcı aynı zamanda yapay bağışık sistemlerdeki iki önemli probleme çözüm getirmektedir. Yapay bağışık sistemlerde uygulanan mutasyon işlemi rastgele bir şekilde yapılmaktadır. Önerilen yöntemde PSO ile mutasyon işlemi gerçekleştirilmektedir. Önerilen yöntem sayesinde antikorlar gideceği yönü bildiğinden algoritmanın yakınsama hızı artırılmıştır. Bir diğer avantaj ise popülasyon boyutunun sabit olmasıdır. Klasik yapay bağışık sistemde klonlama işleminden sonra popülasyon boyutu sürekli artmaktadır. Önerilen yöntemde ise popülasyon boyutu sürekli sabit kalmaktadır. Önerilen algoritma ayrıca daha önceki özellik çıkarım yöntemleri ile elde edilen faz uzayına da uygulanarak performansı yapay sinir ağı ve DVM ile karşılaştırılması sonucunda yapay sinir ağından daha iyi sonuç verdiği ve DVM'ye ise eşdeğer bir performans sergilediği kanıtlanmıştır.

Farklı arıza türlerinin teşhisi edilmesi için uyarlamalı bir bağışık sistem sunulmuştur. Stator arızalarının tespiti için üç faz akım sinyalinin elde edilen iki park vektör bileşeni kullanılmıştır. Park vektör bileşenleri ile oluşturulan iki boyutlu matrisin örüntüsü stator arızalarının tespiti için oldukça uygundur. Arıza durumuna göre bu örüntünün biçimi de değişir. Fakat değişim sürekli aynı yönde olmadığından bu değişimi x ve y koordinatlarına sabitlemek için temel bileşen analizinden faydalanılmıştır. İki boyutlu örüntü matrisinden temel bileşenler aracılığıyla öz değerler ve öz vektörler elde edilerek örüntünün maksimum enerjiye sahip öz vektörleri x ve y koordinatında gösterilmiştir. Elde edilen yeni özellik stator arızalarının büyüklüğünün tespiti için önemli bir arıza indeksidir. Rotor arızalarının tespiti için ise Hilbert dönüşümü üzerinden MAIA uygulanarak $2s_f$ frekansında bir yan bant bileşeni elde edilmiştir. Ayrıca arıza teşhis sisteminin güvenilirliğini sağlamak için ek özellikler de çıkarılmıştır. Yapay bağışık sistemin performansı, her bir adımda hafıza hücrelerinin ortalama afiniteye göre değişimi göz önüne alınarak uyarlanmakta ve yakınsama hızı artırılmaktadır.

Literatürde var olan arıza tahmin ve teşhis sistemleri yüksek çözünürlüklü ve oldukça fazla veriye ihtiyaç duymaktadır. Arıza tahmini ve teşhisinde amaç minimum maliyette ve olabildiğince erken bir aşamada arızayı tespit etmektir. Bu amaçla arıza tahmini için yeni bir özellik çıkarımı sunulmaktadır. Önerilen yöntem farklı çalışma hızları için yüksüz durumda kırık rotor çubuğu arızalarını belirleyebilmektedir. Akım sinyalinin sadece on periyotluk bir kısmı alınarak sembol serisi oluşturulmakta ve serinin entropi değeri hesaplanarak arızalar belirlenmektedir. Entropi normalde verilerdeki düzensizlikleri belirlemek için oldukça uygundur. Entropinin bu özelliği arıza tahmininde kullanılarak sembol serisindeki periyodik olmayan değişimlerden arızalar belirlenmiştir. Yöntemin doğruluğunun belirlenmesi için elde edilen özellik sinyalinin frekans spektrumu MAIA ile elde edilmiş ve rotor arızası ile ilgili yan bant frekansı incelenerek yöntemin performansı doğrulanmıştır. Yöntemin en önemli avantajları rotor kaymasına ihtiyaç duymaması, yüksüz ve düşük çalışma hızlarında arızaları tespit edebilmesi ve veri gereksiniminin az olmasıdır.

Bilgisayar ortamında gerçekleştirilen çevrimiçi durum izleme ve arıza teşhis yöntemlerine ek olarak donanımsal çalışan arıza teşhis yöntemleri geliştirilmiştir. Yeniden programlanabilen ve oldukça hızlı çalışan FPGA kartı üzerinde arıza teşhis, tespit ve tahmin yöntemleri geliştirilmiştir. Stator arızalarının gerçek zamanlı tespiti için önerilen bulanık sistem üç faz akım sinyalinin büyüklüğüne göre arızanın türü hakkında bilgi

vermektedir. RAM'den faz akım sinyallerinin okunarak büyüklüklerinin hesaplanması IEEE-754 kayan noktalı sayı formatında yapılmaktadır. Arıza teşhis edici olarak kullanılan bulanık sistem ise VHDL dilinde yazılmış olup tam sayı tabanlı çalışmaktadır. Önerilen yöntemin sonuçları MATLAB Bulanık mantık toolbox sonuçları ile karşılaştırılmış ve başarılı sonuçlar elde edilmiştir. Motorun herhangi bir fazında oluşan değişimlerin anlık tespiti için yapay sinir ağı tabanlı bir sistem önerilmiştir. Önerilen yöntem tek faz akım sinyali üzerinden elde ettiği özellik sinyalinin ortalamasını ve standart sapmayı giriş olarak alır. Bu iki değer kullanılması sebebi arızalı duruma geçiş erken bir aşamada tespit etmek ve bu durumda sürekli kaldığında da arızayı belirlemektir. Arıza durumuna geçiş anında ortalama ve standart sapma değişirken, arızalı durum esnasında standart sapma sabit kalırken ortalama değişir. Gerçekleştirilen sistem tamamen IEEE-754 kayan noktalı sayı formatına göre çalışmaktadır. Arıza teşhisi için kullanılan kaskad yapay sinir ağı minimum sayıda nöron ile doğrusal olmayan verileri sınıflandırabilmektedir. Kaskad yapay sinir ağının FPGA gerçekleştirimi tek bir nöron üzerinden yapıldığından bir nöron çıkışının hesaplanması için gerekli olan hesaplamaları gerektirir. Dolayısıyla minimum FPGA kaynaklarını kullanmaktadır. Negatif seçim algoritması ile gerçek zamanlı olarak arıza tespiti gerçekleştirilmiştir. Negatif seçim algoritmasının detektörleri üçüncü bölümde önerilen genetik algoritma ile üretilerek FPGA'da RAM'e kaydedilmiştir. Test aşaması bu detektörler kullanılarak gerçekleştirilmiştir. Tek faz akım sinyalinden elde edilen özellik sinyalinin faz uzayı oluşturularak detektörler tarafından tespit edilen veri sayısına göre arıza tespiti yapılmaktadır.

Önerilen yöntemlerin sonuçları değerlendirildiğinde durum izleme, arıza tespiti ve teşhisi için aşağıdaki yenilikler getirilmiştir:

- Arıza tespiti için yeni özellik sinyali elde edilmesi,
- Akıllı hesaplama tekniklerinin arıza tespit ve teşhisinde etkin kullanımı için eklentilerin yapılması,
- Arıza tahmininde sunulan yeni özellik çıkarım yöntemi sayesinde minimum veri ile arıza tahmini,
- Arıza tahmin, tespit ve teşhisinin donanımsal olarak FPGA kartı ile gerçekleştirilmesi.

Önerilen yöntemler ile hem arıza teşhisinde yeni özellikler çıkarılmış hem de akıllı tekniklerin performansının artırılması sağlanmıştır. Bağışık sistem algoritmalarının performansı diğer optimizasyon yöntemleri ile iyileştirilmiş ve yakınsama hızları

arttırılmıştır. Eğitimde kullanılan optimizasyon yöntemlerinin lokal minimuma yakalanmasını engellemek için yöntemler sunulmuştur. Ayrıca arıza teşhisi için donanımsal çalışan FPGA tabanlı akıllı durum izleme teknikleri geliştirilerek algoritmaların bilgisayardan bağımsız çalışabilmeleri sağlanmıştır.

7.1. Öneriler ve Gelecek Çalışmalar

Bu tezde önerilen yöntemler genel amaçlı olup, farklı tür arıza teşhis, tespit problemlerinin yanı sıra sınıflandırma ve anomali tespiti gibi problemlere kolaylıkla uyarlanabilir. Durum izleme için önerilen özellik çıkarım yöntemleri farklı tip sinyallere uygulanabilir. Yapay bağışık sistemlerin performansının geliştirilmesi için önerilen optimizasyon teknikleri diğer akıllı teknikler için de kullanılabilir. FPGA için geliştirilen akıllı tekniklere dayalı modeller genel amaçlı olup farklı problemlere uyarlanabilmektedir.

Arıza teşhisi için yöntemler motorun farklı çalışma durumları göz önüne alınarak performansları değerlendirilebilir. Ayrıca sürücü sistemler ile beslenen motorlardan alınan sinyaller önerilen özellik çıkarım ve akıllı teşhis yöntemleri kullanılabilir. Endüstride iş gücünün büyük kısmını asenkron motorlar karşılamasına rağmen, sürekli mıknatıslı senkron motorlar da boyut ve güç oranlarına göre oldukça kullanışlıdır. Geliştirilen yöntemlerin bu motor türleri için uyarlanması sağlanabilir.

KAYNAKLAR

- [1] **Tanver, P., Li, R., Penman, J., Sedding, H.**, 2008. Condition monitoring of rotating electrical machines. *IET Power and Energy Series 56*, London, United Kingdom.
- [2] **Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A. and Wu B.**, 2006. Intelligent fault diagnosis and prognosis for engineering systems. *Wiley & Sons*, Canada.
- [3] **Barlas, I., Zhang, G., Propes, N., Vachtsevanos, G.**, 2003. Confidence metrics and uncertainty management in prognosis. *MARCON*, Knoxville.
- [4] **Isserman, R.**, 2006. Fault-Diagnosis Systems: An introduction from fault detection to fault Tolerance. *Springer-Verlag*, Berlin, Germany.
- [5] **Chow, M.Y.**, 1998. Methodologies of using artificial neural network and fuzzy logic technologies for motor incipient fault detection. *World Scientific Publishing*, Farrer Road, Singapore.
- [6] **Byington, C., Safa-Bakhsh, R., Watson, M., Kalgren, P.** 2003. Metrics evaluation and tool development for health and usage monitoring system technology, *American Helicopter Society (AHS) International Forum*, Phoenix, Arizona, May 6-8.
- [7] **Propes, N., Lee, S., Zhang, G., Barlas, I., Zhao, Y., Vachtsevanos, G., Thakker, A., Galie, T.**, 2002. A real-time architecture for prognostic enhancement to diagnostic systems, *MARCON*, Knoxville, TN, May 1-3.
- [8] **Xu, L., Ayhan B., Chow, M. Y.**, 2005. Health management systems for power engineering applications: power system distribution systems and rotating machinery systems, *International Conference on Electrical Machines and Systems (ICEMS2004)*, Jeju Island, Korea, Nov. 1-3, s. 1-4.
- [9] **Monmasson E. and Cirstea, M. N.**, 2007. FPGA design methodology for industrial control systems—a review, *IEEE Trans. on Industrial Electronics*, **54 (4)**, 1824–1842.
- [10] **Singh G.K. and Al Kazzaz S.A.S**, 2003. Induction machine drive condition monitoring and diagnostic research – a survey, *Electric Power Systems Research*, **64 (2)**, 145–158.

- [11] **Han Y. and Song Y. H.**, 2003. Condition monitoring techniques for electrical equipment – a literature survey, *IEEE Transactions on Power Delivery* **18** (1), 4–13.
- [12] **Duraisamy, V., Devarajan, N., Somasundareswari, D., Vasanth, A. A. M., Sivanandam, S.N.**, 2007. Neuro fuzzy schemes for fault detection in power transformers, *Applied Soft Computing*, **7** (2), 534-539.
- [13] **Hossam Eldin, A. A., Refaey, M. A.**, 2011. A novel algorithm for discrimination between inrush current and internal faults in power transformer differential protection based on discrete wavelet transform, *Electric Power Systems Research*, **81** (1), 19-24.
- [14] **Amirat, Y., Benbouzid, M.E.H., Al-Ahmar, E., Bensaker, B., Turri, S.**, 2009. A brief status on condition monitoring and fault diagnosis in wind energy conversion systems, *Renewable and Sustainable Energy Reviews*, **13** (9), 2629-2636.
- [15] **Kusiak, A., Li, W.**, 2011. The prediction and diagnosis of wind turbine faults, *Renewable Energy*, **36** (1), 16-23.
- [16] **Chao, K.-H., Ho, S.-H., Wang, M.-H.**, 2008. Modeling and fault diagnosis of a photovoltaic system, *Electric Power Systems Research*, **78** (1), 97-105.
- [17] **Yoon, S., Kim, S., Bae, J., Kim, Y., Kim, E.**, 2011. Experimental evaluation of fault diagnosis in a skew-configured UAV sensor system, *Control Engineering Practice*, **19** (2), 158-173.
- [18] **De Angelo, C. H., Bossio, G.R., Giaccone, S. J., Valla, M. I., Solsona, J. A., Garcia, G. O.**, 2011. Online model-based stator fault detection and identification in induction motors”, *IEEE Trans. On Industrial Electronics*, **56** (11), 4671-4680.
- [19] **Masrur, A. M., Chen, Z., Murphey, Y.**, 2010. Intelligent diagnosis of open and short circuit faults in electric drive inverters for real-time applications, *IET Power Electronics*, **3** (2), 279-291.
- [20] **Cho, H.C., Knowles, J., Fadali, M. S., Kwon, S. L.**, 2010. Fault detection and isolation of induction motors using recurrent neural networks and dynamic Bayesian modeling, *IEEE Trans. On Control Systems Technology*, **18** (2), 430-437.

- [21] Bellini, A., Filipetti, F., Tassoni, C., Capolino, G. A., 2008. Advances in diagnostics techniques for induction machines, *IEEE Trans. On Industrial Electronics*, **55** (12), 4109-4126.
- [22] Ayhan, B., 2005. Linguistic rule generation for broken rotor bar detection in squirrel-cage induction motors, *PhD Thesis, North Carolina State University*, Raleigh, NC.
- [23] Nandi, S., Toliyat, H. A., Xiaodong, L., 2005. Condition monitoring and fault diagnosis of electrical motors-a review, *IEEE Trans. On Energy Conversion*, **20** (4), 719-729.
- [24] Trzynadlowski, A. M., Ghassemzadeh, M., Legowski, S. F., 1999. Diagnostics of mechanical abnormalities in induction motors using instantaneous electric power, *IEEE Trans. on Energy Conversion*, **14**, 1417–1423.
- [25] Wu, S., Chow, T. W. S., 2004. Induction machine fault detection using som-based rbf neural networks, *IEEE Trans. On Industrial Electronics*, **51** (1), 183-194.
- [26] Jang, B., Shin, C., Powers, E. J., Grady, W. M., 2004. Machine fault detection using bicoherence spectra, *21th IEEE Instrumentation and Measurement Technology Conference*, Como, Italy, 18-20 May, s. 1661-1666.
- [27] Benbouzid, M. E. H., Vieira, M., Theys, C., 1999. Induction motors faults detection and localization using stator current advanced signal processing techniques, *IEEE Trans. on Power Electronics*, **14**, 14–22.
- [28] Curpetino, F., de Vanna, E., Salvatore, L., Stasi, S., 2004. Analysis techniques for detection of im broken rotor bars after supply disconnection, *IEEE Trans. On Industrial Applications*, **40** (2), 526-533.
- [29] Cruz, S. M. A., Cardoso, A. J. M., 2001. Rotor cage fault diagnosis in three-phase induction motors by extended park's vector approach, *IEEE Trans. on Industrial Applications*, **37** (5), 1227 - 1233.
- [30] Haji, M., Toliyat, H. A., 2001. Pattern recognition-a technique for induction machines rotor faults detection 'eccentricity and broken bar fault, *IEEE International 36th Industry Applications Conference*, 30 Sept.- 4 Oct., Chicago, Illusions, USA, 1572 – 1578.
- [31] Kral, C., Habetler, T. G., Harley, R. G., 2004. Detection of mechanical imbalances of induction machines without spectral analysis of time domain signals” *IEEE Trans. on Industry Applications*, **40** (4), 1101–1106.

- [32] Bellini, A., Franceschini, G., Tassoni, C., 2006. Monitoring of induction machines by maximum covariance method for frequency tracking, *IEEE Trans. Industry Applications*, **42** (1), 69–78.
- [33] Blodt, M., Regnier, J., Faucher, J., 2006. Distinguishing load torque oscillations and eccentricity faults in induction motors using stator current wigner distributions, *IEEE Trans. On Industry Applications*, **3**, 1549-1556.
- [34] Elias, G. S., , Aviyente, S., Zaidi, S. S. H., 2008. Time–frequency analysis for efficient fault diagnosis and failure prognosis for interior permanent-magnet ac motors, *IEEE Trans. on Industrial Electronics*, **55** (12), 4191-4199.
- [35] Kim, K., Parlos, A. G., 2002. Induction motor fault diagnosis based on neuropsychopredictors and wavelet signal processing, *IEEE Trans. on Mechatronics*, **7** (2), 201-219.
- [36] Abbaszadeh, K., Milimonfared, J., Haji, M., Toliyat, H. A., 2001. Broken bar detection in induction motor via wavelet transformation, *The 27th Annual Conference of IEEE Industrial Electronics Society*, Nov. 29, Denver Colorado, USA, s. 95-100.
- [37] Eren, L., Devaney, M. J., 2004. Bearing damage detection via wavelet packet decomposition of the stator current, *IEEE Trans. on Instrumentation and Measurement*, **53** (2), 431–436.
- [38] Tsoumas, I. P., Georgoulas, G., Mitronikas, E. D., Safacas, A. N., 2008. Asynchronous machine rotor fault diagnosis technique using complex wavelets, *IEEE Trans. on Energy Conversion*, **23** (2), 444-459.
- [39] Briz, F., Degner, M. W., Garcia, P., Bragado, D., 2008. Broken rotor bar detection in line-fed induction machines using complex wavelet analysis of startup transients, *IEEE Trans. on Industry Applications*, **44** (3), 760-768.
- [40] Guillermo A. J., Alfredo, O. M., Manuel A. D. M., 2007. Fault detection in induction motors using hilbert and wavelet transforms, *Journal of Electrical Engineering*, **89** (3), 205-220.
- [41] Zhang, Y., 2006. Hilbert-huang transform and marginal spectrum for detection of bearing localized defects, *IEEE 6th Intelligent Control and Automation Conference*, June 21-23, Dalian, China, s. 5457-5461.

- [42] Kowalski, C.T. and Kowalska T.O., 2003. Neural networks application for induction motor faults diagnosis, *Mathematics and Computers in Simulation*, **63**, 435-448.
- [43] Filippetti, F., Franceschini, G., Tassoni, C., Vas, P., 1998. AI Techniques in Induction Machines Diagnosis including the Speed Ripple Effect, *IEEE Trans. on Industry Applications.*, **34** (1), 98–108.
- [44] Salles, G., Filippetti, F., Tassoni, C., Crellet, G., Franceschini, G., 2000. Monitoring of induction motor load by neural network techniques, *IEEE Trans. Power Electronics*, **15** (4), 762–768.
- [45] Filippetti, F., Franceschini, G., Tassoni, C., 1995. Neural networks aided online diagnostics of induction motor rotor faults, *IEEE Trans. Industry Applications*, **31** (4), 892–899.
- [46] Filippetti, F., Franceschini, G., Tassoni, C., Vas, P. 2000. Recent developments of induction motor drives fault diagnosis using AI techniques, *IEEE Trans. on Industrial Electronics*, **47** (5), 994–1004.
- [47] Su, H., Chong, K. T., 2007. Induction machine condition monitoring using neural network modeling, *IEEE Trans. on Industrial Electronics*, **54** (1), 241–249.
- [48] Martins, J. F., Pires, V. F., Pires, A. J., 2007. Unsupervised neural-network based algorithm for an on-line diagnosis of three-phase induction motor stator fault, *IEEE Trans. Industrial Electronics*, **54** (1), 259–264.
- [49] Zidani, F., Benbouzid, M. E. H., Diallo, D. Nait-Said, M. S., 2003. Induction motor stator faults diagnosis by a current concordia pattern-based fuzzy decision system, *IEEE Trans. Energy Conversion*, **18** (4), 469–475.
- [50] Altug, S. Chow, M. Y., Trussell, H. J., 1999. fuzzy inference systems implemented on neural architectures for motor fault detection and diagnosis, *IEEE Trans. Industrial Electronics*, **46** (6), 1069–1079.
- [51] Ballal, M. S., Khan, Z. J., Suryawanshi, H. M., Sonolikar, R. L., 2007. Adaptive neural fuzzy inference system for the detection of inter-turn insulation and bearing wear faults in induction motor, *IEEE Trans. Industrial Electronics*, **54** (1), 250–258.
- [52] Zidani, F., Diallo, D., Benbouzid, M. E. H., Nait-Said, R., 2008. A fuzzy based approach for the diagnosis of fault modes in a voltage-fed PWM inverter induction motor drive, *IEEE Trans. Industrial Electronics*, **55** (2), 586–593.

- [53] **Rodriguez, P. V. J., Arkkio, A.,** 2008. Detection of stator winding fault in induction motor using fuzzy logic, *Applied Soft Computing*, **8**, 1112–1120.
- [54] **Samanta, B., Nataraj, C.,** 2008. Prognostics of machine condition using soft computing, *Robotics and Computer-Integrated Manufacturing*, **24**, 816–823.
- [55] **Rodriguez, P. V. J., Negrea, M., Arkkio, A.,** 2008. A simplified scheme for induction motor condition monitoring, *Mechanical Systems and Signal Processing*, **22**, 1216–1236.
- [56] **Widodo, A., Yang, B.,** 2008. Wavelet support vector machine for induction machine fault diagnosis based on transient current signal, *Expert Systems with Applications*, **35**, 307–316.
- [57] **Yuan, S., Chu, F.,** 2007. Fault diagnostics based on particle swarm optimisation and support vector machines, *Mechanical Systems and Signal Processing*, **21** (4), 1787-1798.
- [58] **Hu, Q., He, Z., Zhang, Z., Zi, Y.** 2007. Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble, *Mechanical Systems and Signal Processing*, **21** (2), 688-705.
- [59] **Sugumaran, V., Sabareesh, G. R., Ramachandran, K.I.,** 2008. Fault diagnostics of roller bearing using kernel based neighborhood score multi-class support vector machine, *Expert Systems with Applications*, **34** (4), 3090-3098.
- [60] **Gao, X. Z., Ovaska, S. J., Wang, X., Chow, M. Y.,** 2006. Clonal optimization of negative selection algorithm with applications in motor fault detection, *IEEE International Conf. on Systems Man Cybernetics*, Oct. 8-11, Taipei, Taiwan, s. 5118-5123.
- [61] **Branco, P. J. C., Dente, J. A., Mendes, R. V.,** 2003. Using immunology principles for fault detection, *IEEE Trans. On Industrial Electronics*, **50** (2), 362-373.
- [62] **Gan, Z., Zhao, M. B., Chow, T. W. S.,** 2008. Induction machine fault detection using clone selection programming, *Expert Systems with Applications*, **36** (4), 8000-8012.
- [63] **Yeh, C., Povinelli, R. J., Mirafzal, B., Demerdash, N. A. O.,** 2004. Diagnosing of stator winding inter-turn shorts in induction motors fed by PWM-inverter

- drive systems using a time series data mining, *IEEE International Conference on Power Systems*, Nov. 21-24, Singapore, s. 891- 896.
- [64] da Silva, A. M., Povinelli, R. J., Demerdash, N. A. O., 2008. Induction machine broken bar and stator short-circuit fault diagnostics based on three phase stator current envelopes, *IEEE Trans. on Industrial Electronics*, **55** (3), 1310-1318.
- [65] Kral, C., Wieser, R. S., Pirker, F., Schagginger, M., 2000. Sequences of field-oriented control for the detection of faulty rotor bars in induction machines—the Vienna monitoring method, *IEEE Trans. on Industrial Electronics*, **47** (5), 1042–1050.
- [66] Hsu, J. S., 1995. Monitoring of defects in induction motors through airgap torque observation, *IEEE Trans. on Industrial Applications*, **31** (5), 1016–1021.
- [67] Demian, C., Mpanda-Mabwe, A., Henao, H., Capolino, G. A., 2004. Detection of induction machines rotor faults at standstill using signals injection, *IEEE Trans. Industrial Applications*, **40** (6), 1550–1559.
- [68] Briz, F., Degner, M. W., Diez, A. B., Guerrero, J. M., online diagnostics in inverter-fed induction machines using high-frequency signal injection, *IEEE Trans. on Industrial Applications*, **40** (4), 1153–1161.
- [69] Bellini, A., Filippetti, F., Franceschini, G., Petrolini, N., Tassoni, C., 2001. Online diagnosis in induction drives rotor by signal injection technique: fault localization and severity classification, *IEEE International Conference on Diagnostics for Electrical Machines, Power Electronics and Drives*, Sept. 1-3, Grado, Italy, s. 531–536.
- [70] Henao, H., Demian, C., Capolino, G. A., 2003. A frequency-domain detection of stator winding faults in induction machines using an external flux sensor, *IEEE Trans. on Industrial Applications*, **39** (5), 1272–1279.
- [71] Aydin, I., Karaköse, M., Akin, E., 2008. Artificial immune inspired fault detection algorithm based on fuzzy clustering and genetic algorithm methods, *IEEE Conf. on Computational Intelligence for Measurement Systems and Applications*, July 14-16, Istanbul, Turkey, s. 93 – 98.
- [72] Aydin, I., Karaköse, M., Akin, E., 2009. K-NN based hybrid negative selection algorithm for anomaly detection, *International Conference of Innovations*

on *Intelligent Systems and Technology Applications (INISTA)*, June 28-30, Trabzon, Turkey.

- [73] **Aydin, I., Karaköse, M., Akin, E.**, 2009. Genetik algoritma kullanan yapay bağışık sistem tabanlı arıza teşhis modeli, *Dokuz Eylül Üniversitesi Fen ve Mühendislik Dergisi*, **11 (31)**, 57-72.
- [74] **Aydin, I., Karaköse, M., Akin, E.**, 2009. Kırık rotor çubuğu ve stator arızalarının teşhisinde yapay sinir ağı yaklaşımı, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, **25 (1-2)**, 134 – 149.
- [75] **Aydin, I., Karaköse, M., Akin, E.**, 2010. Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection, *Expert System with Applications*, **37 (7)**, 5285-5294.
- [76] **Aydin, I., Karaköse, M., Akin, E.**, 2008. Zaman serisi veri madenciliği ve destek vektör makinalar kullanan yeni bir arıza sınıflandırma yöntemi, *Gazi Üniv. Müh. Mim. Fak. Der.*, **23 (2)**, 431-440.
- [77] **Aydin, I., Karakose, M., Akin, E.**, 2009. Fuzzy supervised artificial immune system for fault diagnosis, *First International Fuzzy Conference (FUZYSS)*, Sept. 28-30, Ankara, Turkey, s. 59-65.
- [78] **Aydin, I., Karaköse, M., Akin, E.**, 2010. Asenkron motorlarda arıza teşhisi için kaotik sürü optimizasyonu tabanlı yapay bağışık algoritması, *Otomatik Kontrol Ulusal Toplantısı (TOK'10)*, 21-23 Eylül 2010, Gebze, Kocaeli, s. 574-579.
- [79] **Aydin, I., Karakose, M., Akin, E.**, 2010. An adaptive artificial immune system for fault classification, *Journal of Intelligent Manufacturing*, DOI: 10.1007/s10845-010-0449-5.
- [80] **Aydin, I., Karakose, M., Akin, E.**, 2010. Artificial immune classifier with swarm learning, *Engineering Applications of Artificial Intelligence*, **23 (8)**, 1291-1302.
- [81] **Aydin, I., Karakose, M., Akin, E.**, 2011. A multi-objective artificial immune algorithm for parameter optimization in support vector machine, *Applied Soft Computing*, **11 (1)**, 120-129.
- [82] **Aydin, I., Karakose, M., Akin, E.**, 2009. Early fault detection of broken rotor bars using sliding window symbol sequence statistics, *Australasian Universities*

- Power Engineering Conference (AUPEC 09)*, Sept. 17-19, Adelaide, Australia, s. 1-6.
- [83] **Aydin, I., Karakose, M., Akin, E.**, 2009. Bulanık otomata yaklaşımı kullanan arıza tespit algoritması, *13. Ulusal Elektrik, Elektronik, Bilgisayar, Biyomedikal Kongresi*, 23-26 Aralık, Ankara, Türkiye, s. 121-126.
- [84] **Aydin, I., Karakose, M., Akin, E.**, 2011. A new method for early fault detection and diagnosis of broken rotor bars, *Energy Conversion and Management*, **52 (4)**, 1790-1799.
- [85] **Aydin, I., Karakose, M., Akin, E.**, 2010. FPGA based real time fuzzy fault detection algorithm, *IEEE International Soft Computing and Pattern Recognition Conference (SocPar)*, Dec. 7-10 2010, Paris, France, s. 389-394.
- [86] **Akin, E., Aydin, I., Karakose, M.**, 2011. FPGA based intelligent condition monitoring of induction motors: detection, diagnosis, and prognosis, *IEEE International Conference on Industrial Electronics (ICIT) & Southeastern Symposium on System Theory (SSST)*, March 14-17, Auburn, Alabama, USA, s. 373-378.
- [87] **Akin, E., Aydin, I., Karakose, M.**, 2011. Arıza Teşhisi için Gerçek Zamanlı Bağışık Sistem Uygulaması, *Otomatik Kontrol Türk Milli Komitesi 2011 Ulusal Toplantısı*, 13-14 Eylül 2011, İzmir, Türkiye, s. 671-676.
- [88] **Tanver, P.J.**, 2008. Review of condition monitoring of rotating electrical machines, *IET Electr. Power Appl.*, **2 (4)**, 215–247.
- [89] **Han, Y., Song, Y.H.**, 2003. Condition monitoring techniques for electrical equipment- a literature survey, *IEEE Trans. on Power Delivery*, **18 (1)**, 4-13.
- [90] **Benbouzid, M. E. H., Kliman, G. B.**, 2003. What stator current processing-based technique to use for induction motor rotor fault diagnosis?, *IEEE Trans. on Energy Conversion*, **18 (2)**, 238-244.
- [91] **Wang, W.**, 2008. An intelligent system for machinery condition monitoring, *IEEE Trans. on Fuzzy Systems*, **16 (1)**, 110-122.
- [92] **Riley, C. M., Lin, B. K., and Habetler, T. G.**, 1999. Stator current harmonics and their causal vibrations: A preliminary investigation of sensorless vibration monitoring applications, *IEEE Trans. Ind. Applications*, **35**, 94–99.

- [93] **National Instruments.** 2011. Data Acquisition Board Reference Manual, Access Date:2011.
- [94] **Benbouzid, M.E.H., Nejjari, H.** 2001. A simple fuzzy logic approach for induction motors stator condition monitoring, *IEEE International Electric Machines and Drives Conference (IEMDC)*, 17-20 June, Cambridge, MA, USA, s. 634-639.
- [95] **Ayhan, B.,** 2000. Fault detection of induction motors using neural networks, *Ms. Dissertation*, Boğaziçi Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.
- [96] **Chilengue, Z., Dente, J. A., Branco, P. J. C.,** 2011. An artificial immune system approach for fault detection in the stator and rotor circuits of induction machines, *Electric Power Systems Research*, **81 (1)**, 158-169.
- [97] **Jolliffe, I. T.,** 1986. Principal component analysis, *New York:664 Springer*.
- [98] **Pires, V. F., Martins, J. F., Pires, A. J.,** 2010. Eigenvector/eigenvalue analysis of a 3d current referential fault detection and diagnosis of an induction motor, *Energy Conversion and Management*, **51 (5)**, 901-907.
- [99] **Trajin, B., Chabert, M., Regnier, J., Faucher, J.,** 2009. Hilbert versus concordia transform for three-phase machine stator current time-frequency monitoring, *Mechanical Systems and Signal Processing*, **23 (8)**, 2648-2657.
- [100] **Daviu, J. A., Rodriguez, P. J., Guasp, M. R., Arkiko, A., Folch, J. R. , Perez, R. B.,** 2009. Transient detection of eccentricity-related components in induction motors through the hilbert-huang transform, *Energy Conversion and Management*, **50 (7)**, 1810-1820.
- [101] **Cooley, J.W., Tukey, J.W.,** 1965. An algorithm for machine calculation of complex series, *Mathematics of Computation Journal*, **19 (90)**, 297–301.
- [102] **Kohler, J. L., Sottile, J., Trutt, F. C.,** 1992. Alternatives for assessing the electrical integrity of induction motors, *IEEE Transactions on Industry Applications*, **28**, 1109–1117.
- [103] **Siddique, A., Yadava, G. S., Singh, B.,** 2005. A review of stator fault monitoring techniques of induction motors, *IEEE Transactions on Energy Conversion*, **20**, 106–114.
- [104] **Povinelli, R. J.,** 1999. Time series data mining: identifying temporal patterns for characterization and prediction of time series events, *PhD Dissertation*, Marquette University, Milwaukee, WI.

- [105] **Kantz, H., Schreiber, T.**, 2004. Nonlinear time series analysis, *Cambridge university press*, Cambridge, United Kingdom.
- [106] **Abarbanel, H. D. I.**, 1996. Analysis of observed chaotic data, *Springer*, New York, USA.
- [107] **Tufillaro, N.B., Abbott, T., Reilly, J.**, 1992. An experimental approach to nonlinear Dynamics and chaos, *Addison-Wesley*, California, USA.
- [108] **Povinelli, R.J., Feng, X.**, 2003. A new temporal pattern identification method for characterization and prediction of complex time series events, *IEEE Transactions on Knowledge and Data Engineering*, **15 (2)**, 339 – 352.
- [109] **Takens, F. C.**, 1981. Detecting strange attractors in turbulence, *Lecture Notes on Mathematics*, **898 (1981)**, 366–381.
- [110] **Povinelli, R.J., Bangura, J. F., Demerdash, N. A. O., Brown, R. H.**, 2002. Diagnostics of bar and end-ring connector breakage faults in polyphase induction motors through a novel dual track of time-series data mining and time-stepping coupled FE-state space modeling, *IEEE Trans. on Energy Conversion*, **17 (1)**, 39-46.
- [111] **Bangura, J. F., Povinelli, R. J. Demerdash, N. A. O., Brown, R. H.**, 2003. Diagnostics of eccentricities and bar/end-ring connector breakages in polyphase induction motors through a combination of time-series data mining and time-stepping coupled FE-state space techniques, *IEEE Trans. on Industry Applications*, **39 (4)**, 1005-1013.
- [112] **De Castro, L. N., Timmis, J. I.**, 2002. Artificial immune systems: a new computational intelligence approach, *Springer-Verlag*, London, England.
- [113] **Igawa, K., Ohashi, H.**, 2009. A negative selection algorithm for classification and reduction of noise effect, *Applied Soft Computing*, **9 (1)**, 431–438.
- [114] **Watkins, A. B.**, 2001. AAIRS: A Resource Limited Artificial Immune Classifier, *M.S. Thesis*. Mississippi State University, MS., USA.
- [115] **De Castro, L.N., Zuben, F. J. V.**, 2002. Learning and optimization using the clonal selection principle, *IEEE Trans. on Evolutionary Computation*, **6 (3)**, 239-251.
- [116] **Forrest, S., Perelson, A. S., Allen, L., Cherukuri, R.**, 1994. Self-non-self-discrimination in a computer, *Proceedings of IEEE symposium on research in security and privacy*, May 16-18, Oakland, CA , USA, s. 202–212.

- [117] Dasgupta, D., 2006. Advances in artificial immune systems, *IEEE Computational Intelligence Magazine*, 1 (4), 40–49.
- [118] Zhong, Y., Zhang, L., Huang, B., Li, P., 2006. An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery, *IEEE Trans. Geoscience Remote Sensing*, **44** (2), 420–431.
- [119] Nابی‌ی‌ف، V. V. 2003. Yapay zeka-problemler, yöntemler, algoritmalar, *Seçkin Yayıncılık*, Ankara, Türkiye.
- [120] Coelho, L. D. S., Mariani, V. C., 2008. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization, *Expert System with Applications*, **34**, 1905-1913.
- [121] Alatas, B., Akin, E., 2009. Chaotically Encoded Particle Swarm Optimization Algorithm, *Chaos Solutions & Fractals*, **41** (2), 939-950.
- [122] Heidari, B. G., McGillem, C. D. A., 1994. Chaotic direct-sequence spread spectrum communication system, *IEEE Transactions on Communications*, **42** (2), 1524–1527.
- [123] Sinai, J.G., 1991. Finite-dimensional randomness, *Russ. Math. Surv.*, **46**, 177-191.
- [124] Ayhan, B., Chow, M.-Y., Song, M.-H., 2005. Multiple discriminant analysis and neural-network-based monolith and partition fault-detection schemes for broken rotor bar in induction motors. *IEEE Trans. on Industrial Electronics*, **53** (4), 1298-1308.
- [125] Ayhan, B., Chow, M. Y., Trussell H. J., 2007. A fuzzy membership function design methodology based on histogram and ROC curve analyses for broken rotor bar detection, *Proceedings of IEEE International Symposium on Industrial Electronics*, June 4-7, Vigo, Spain, s. 1080-1085.
- [126] Cristianini, N. and Shawe, T. J., 2000. An introduction to support vector machine and other kernel-based learning methods, *Cambridge University Press*, Cambridge, United Kingdom.
- [127] Wang, L., 2005. Support vector machine theory and applications, *Springer*, Berlin Germany.
- [128] Platt, C., 1999. Fast training of support vector machine using sequential minimal optimization, in *Advances in Kernel Methods—Support Vector Learning*, pp. 185-208, Eds. Schölkopf, B., Burges, C., and Smola, A., Editors, MIT Press, Cambridge.

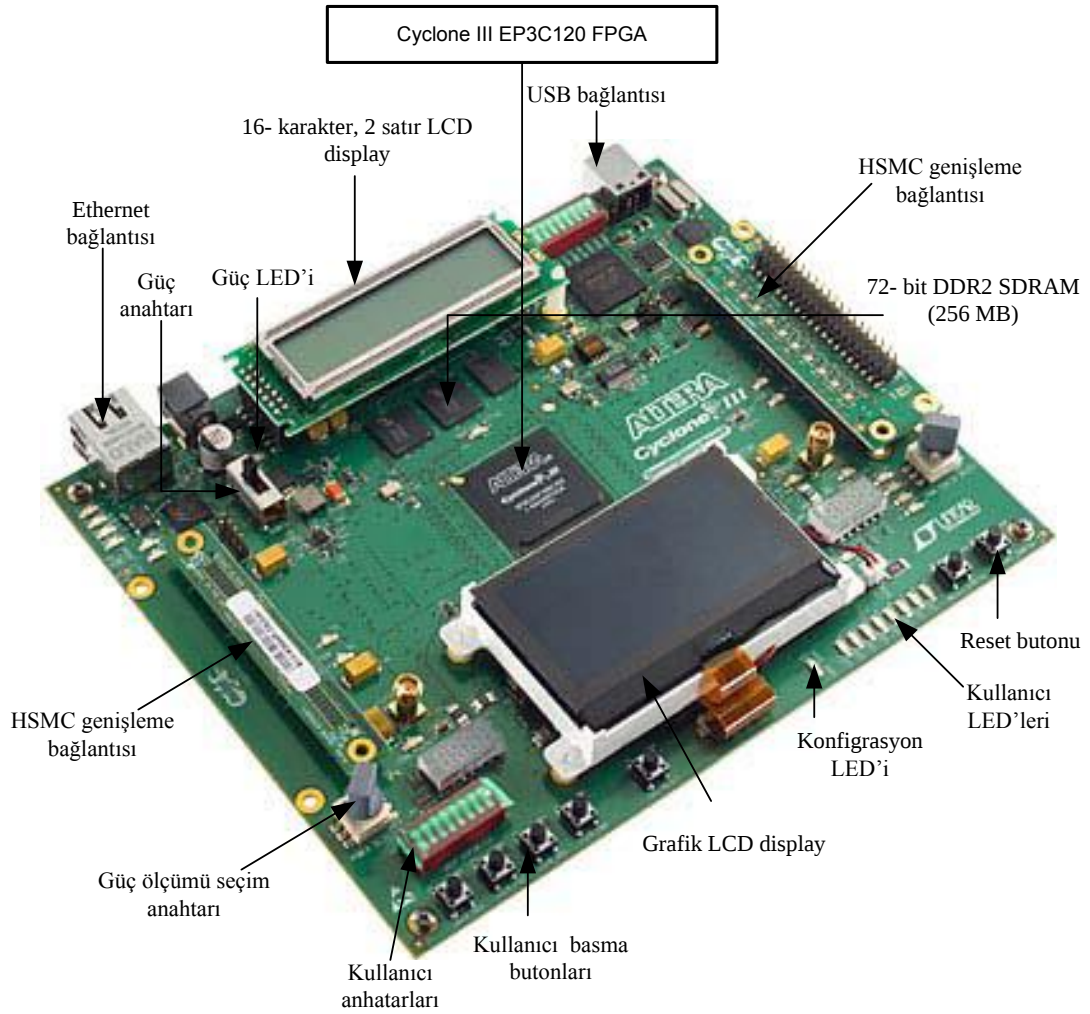
- [129] **Kennedy, J., Eberhart, R.C.**, 1995. Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, Nov. 27- Dec.1, Perth, WA , Australia, s. 1942–1948.
- [130] **Blackwell, T.M., Bentley, P.J.**, 2002. Don't push me! collision-avoiding swarms, *Proceedings of the IEEE Computational Intelligence*, May 12-17, Honolulu, HI, USA, s. 1691–1696.
- [131] **Panadero, R. P., Sanchez, M. P., Guasp, M. R. , Folch, J. R., Perez, E. H., Cruz, J. P.**, 2009. Improved resolution of the MCSA method via Hilbert transform, enabling the diagnosis of rotor asymmetries at very low slip, *IEEE Transactions on Energy Conversion*, **24**, 52-59.
- [132] **Thomson, W. T., Fenger, M.**, 2001. Current signature analysis to detect induction motor faults, *IEEE Industrial Applications Magazine*, **7**, 26–34.
- [133] **Duda, R.O., Hart, P. E.**, 1973. Pattern classification and scene analysis, New York: Wiley.
- [134] **Keller, J. M., Gray, M. R., Givens, J. A.**, 1985. A fuzzy k-nearest neighbor algorithm, *IEEE Transactions on Systems Man and Cybernetics*, **15**, 580-585.
- [135] **Hao, X., Cai-xin, S.** 2007. Artificial immune network classification algorithm for fault diagnosis of power transformer. *IEEE Trans. on Power Delivery*, **22** (2), 930-935.
- [136] **Leung, K., Cheong, F., Cheong, C.**, 2007. Generating compact classifier systems using a simple artificial immune system, *IEEE Trans. on Systems, Man and Cybernetics-Part B*, **37** (5), 1344-1356.
- [137] **Kurths, J., Schwarz, U., Witt, A., Krampe, R.Th., Abel, M.**, 1996. Measures of complexity in signal analysis, in *Chaotic, fractal, and nonlinear signal processing*, pp. 128-134, Ed. Katz, R.A., Woodbury, NY: AIP Press.
- [138] **Zhao, Z., Jia, M., Wang, F., Wang, S.**, 2009. Intermittent chaos and sliding window symbol sequence statistics-based early fault diagnosis for hydraulic pump on hydraulic tube tester, *Mechanical Systems and Signal Processing*, **23**, 1573–1585.
- [139] **Chin, S.C., Ray, A., Rajagopalan, V.**, 2005. Symbolic time series analysis for anomaly detection: a comparative evaluation, *Journal of Signal Processing*, **85**, 1573–1585.

- [140] Daw, C. S., Finney, C. E. A., Tracy, E. R., 2002. A Review of symbolic analysis of experimental data, *Review of Scientific Instruments*, **74**, 915-930.
- [141] Trzynadlowski, A. M. and Ritchie, E., 2000. Comparative investigation of diagnostic media for induction motors: A case of rotor cage faults, *IEEE Trans. on Industrial Electronics*, **47**, 1092–1099.
- [142] MATLAB, The Mathworks Inc., *Natwick, MA (USA)*, License Reference Code: 585775.
- [143] Moreno, A. O., Troncoso, R. J. R., Frias, J. A. V., Gillen, J. R. R. G., Perez, A.G., 2008. Automatic online diagnosis algorithm for broken-bar detection on induction motors based on discrete wavelet transform for FPGA implementation, *IEEE Trans. on Industrial Electronics*, **55** (5), 2193- 2202.
- [144] Xilinx Staff, 2004. Celebrating 20 years of innovation, *Xcell J.*, **48**.
- [145] Andina, R. J. J., Moure, M. J., Valdes, M. D., 2007. Features, design tools, and application domains of FPGAs, *IEEE Trans. On Industrial Electronics*, **54** (4), 1810-1823.
- [146] Valsan, P. S. and Swarup, K. S., 2009. High-speed fault classification in power lines: Theory and FPGA-based implementation, *IEEE Trans. on Industrial Electronics*, **56** (5), 1793–1800.
- [147] Wolf, W. H., 2004. FPGA-based system design. *Englewood Cliffs, NJ: Prentice-Hall*.
- [148] Chu, P. P., 2008. FPGA prototyping by VHDL examples- xilinx spartan-3 version, *John Wiley & Sons Press*, New Jersey.
- [149] Monmasson, E., Idkhajine, L., Cirstea, M. N., Bahri, I., Tisan, A., Naouar, M. W., 2011. FPGAs in industrial control applications, *IEEE Trans. on Industrial Informatics*, **7** (2), 224-243.
- [150] Pedroni, V. A., 2004. Circuit design with VHDL, *MIT Press*, Cambridge, London, England.
- [151] Altera Data Book, 2011. [Online]. Available: www.altera.com.
- [152] Chen, Y. Dinavahi, V., 2011. An iterative real-time nonlinear electromagnetic transient solver on FPGA, *IEEE Trans. on Industrial Electronics*, **58** (6), 2547-2555.

- [153] **Barriga, A., Sanchez-Solano, S., Brox, P., Cabrera, A., Baturone, I.,** 2006. Modeling and implementation of fuzzy systems based on VHDL, *International J. of Approximate Reasoning*, **41**, 164-178.
- [154] **Cirstea, M. N., Dinu, A., Khor, J. G., McCormick, M.,** 2002. Neural and fuzzy logic control of drives and power systems, *Newness Publisher*, Wildwood, Avenue, Woburn, MA.
- [155] **Wilamowski B.M.,** 2009. Neural network architectures and learning algorithms, *IEEE Industrial Electronics Magazine*, **3 (4)**, 56-63.
- [156] **Malinowski, A. and Yu, H.,** 2011. Comparison of embedded system for industrial applications, *IEEE Trans. on Industrial Informatics*, **7 (2)**, 244-254.

EK-1. FPGA GELİŞTİRME KARTI VE ÖZELLİKLERİ

Bu tez çalışmasında, Altera firmasının üretmiş olduğu Cyclone ailesinden Cyclone III model FPGA geliştirme kiti kullanılmıştır. Bu bölümde kullandığımız FPGA geliştirme kitinin özellikleri verilecektir. Şekil EK-1.1’de kullanılan geliştirme kartının görünüşü verilmiştir.



Ek Şekil 1.1. FPGA geliştirme kartı

Geliştirme kartının teknik özellikleri şu şekilde sıralanabilir:

- 780-pin Cyclone III EP3C120 FPGA
 - 119 K lojik elemanlar

- 3.888 Kbit hafıza
- 288 adet 18X18 çarpma blokları
- Dört adet PLL
- 20 genel saat ağı
- 531 adet kullanıcı I/O pini
- 1.2 V güç
- Hafıza birimleri
 - 256-MB çift kanallı DDR2 SDRAM
 - 8 MB SRAM
 - 64-MB Flash RAM
- Clock devresi
 - İki yüksek hızlı clock devresi
 - 50 MHZ
 - 125 MHZ
- Genel amaçlı kullanıcı ve karta ait ara yüzler
 - Push-button anahtarları
 - Kullanıcı DIP switch
 - Kullanıcı LEDleri
 - Borda özgü DIP switch ve Push-butonlar
 - 128x64 grafik LCD
 - 16x2 satır karakter LCD
 - Kullanıcı 7- parçalı display
 - Karta özgü güç tüketim display'i

ÖZGEÇMİŞ

1981- Elazığ doğumlu olan İlhan AYDIN ilk, orta ve lise öğrenimini Elazığ'da tamamladı. 1997'yılında Atatürk Lisesini birincilikle bitirdikten sonra girdiği Fırat Üniversitesi Bilgisayar Mühendisliği Bölümünden 2001 yılında mezun oldu. 2001 yılında Fırat Üniversitesi Kemaliye Hacı Ali Akın Meslek Yüksekokuluna Öğretim Görevlisi olarak atandı. 2003 yılında Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim dalında yüksek lisans yapma hakkı kazandı. 2006 yılında yüksek lisansını tamamlayarak Bilgisayar Yüksek Mühendisi unvanını aldı ve aynı yıl Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Ana Bilim dalında doktora eğitimine başladı. 2009 yılında Fırat Üniversitesi Bilgisayar Mühendisliği Bölümüne Araştırma Görevlisi olarak atandı. Halen Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak görevine devam etmektedir.