

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

UÇUŞ KONTROL SİSTEMLERİNDE SENSÖR HATALARININ
TESPİTİ VE TEŞHİSİ

YÜKSEK LİSANS TEZİ

Behnam MAHMOUDI

OCAK 2016
TRABZON



KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

UÇUŞ KONTROL SİSTEMLERİNDE SENSÖR HATALARININ
TESPİTİ VE TEŞHİSİ

Behnam MAHMOUDI

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
“ELEKTRONİK YÜKSEK MÜHENDİSİ”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 23.12.2015
Tezin Savunma Tarihi : 13.01.2016

Tez Danışmanı : Yrd. Doç. Dr. Yusuf SEVİM

Trabzon 2016

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**Elektrik-Elektronik Mühendisliği Anabilim Dalında
Behnam MAHMOUDI Tarafından Hazırlanan**

**UÇUŞ KONTROL SİSTEMLERİNDE SENSÖR HATALARININ
TESPİTİ VE TEŞHİSİ**

**başlıklı bu çalışma, Enstitü Yönetim Kurulunun 29/12/2015 gün ve 1633 sayılı
kararıyla oluşturulan jüri tarafından yapılan sınavda
YÜKSEK LİSANS TEZİ
olarak kabul edilmiştir.**

Jüri Üyeleri

Başkan : Doç. Dr. Ayten ATASOY

Üye : Yrd. Doç. Dr. Yusuf SEVİM

Üye : Yrd. Doç. Dr. Yasin OĞUZ



Prof. Dr. Sadettin KORKMAZ

Enstitü Müdürü

ÖNSÖZ

Bu çalışma, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak hazırlanmıştır.

Öncelikle bu çalışmam esnasında ilgisini esirgemeyen, bana her konuda yardımcı olan, bilgi ve deneyimlerinden yararlandığım, değerli hocam Sayın Yrd. Doç. Dr. Yusuf SEVİM'e teşekkür eder, saygılarımı sunarım.

Eğitim hayatımın tüm aşamalarında bana emeği geçen tüm öğretmenlerime bu vesileyle teşekkür ederim.

Ayrıca vermiş oldukları manevi destekten dolayı arkadaşlarım İlker GÜNEŞ ve Ahmet KÜRKÇÜ'ye en içten duygularıyla teşekkür ederim.

Son olarak, hep yanımda olan, benden maddi-manevi desteğini esirgemeyen, güvenlerini her fırsatta gösteren aileme minnet ve şükranlarımı sunarım.

Behnam MAHMOUDI
Trabzon, 2016

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Uçuş Kontrol Sistemlerinde Sensör Hatalarının Tespiti ve Teşhisi” başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd. Doç. Dr. Yusuf SEVİM’in sorumluluğunda tamamladığımı, verileri kendim topladığımı, analizleri ilgili laboratuvarlarda yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 13/01/2016

Behnam MAHMOUDI

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VIII
SUMMARY	IX
ŞEKİLLER DİZİNİ	X
TABLolar DİZİNİ.....	XII
SEMBOLLER DİZİNİ	XIII
1. GENEL BİLGİLER.....	1
1.1. Giriş	1
1.2. Uçak Kazaları ve Hata Toleranslı Kontrol Sistemlerinin (HTKS) Önemi	3
1.3. HTKS	4
1.4. Hata Tespiti ve Teşhisi (HTT) Yöntemleri.....	5
1.4.1. Model Tabanlı Yöntemler	6
1.4.1.1. Nicel Model Tabanlı Yaklaşımlar	7
1.4.1.1.1. Gözleyici Tabanlı Yaklaşım	7
1.4.1.1.2. Kalman Filtreler Modeli	8
1.4.1.1.3. Eşli Uzay Yaklaşımı	8
1.4.1.1.4. Parametre Kestirimi Şeması	8
1.4.1.2. Nitel Model Tabanlı Yaklaşımlar	8
1.4.1.2.1. Yönlü Çizit Tabanlı Yaklaşım	9
1.4.1.2.2. Hata Ağacı Analizi	9
1.4.1.2.3. Fiziksel Modeller	9
1.4.2. Veri Tabanlı Yöntemler	9
1.4.2.1. Nitel Veri Tabanlı Yaklaşımlar	10
1.4.2.1.1. Uzman Sistemler.....	10
1.4.2.1.2. Trend Modellemesi.....	10
1.4.2.2. Nicel Veri Tabanlı Yaklaşımlar.....	11
1.4.2.2.1. Çok Değişkenli İstatistik Yaklaşımlar	11
1.4.2.2.2. İstatistiksel Sınıflandırma Yaklaşımları	11
1.4.2.2.3. Sinir Ağları Yaklaşımları.....	12

1.5.	Hata Türleri.....	12
1.6.	Hata Tespiti İçin Kullanılan Yöntemler	15
1.6.1.	Temel Bileşen Analizi	15
1.6.1.1.	Yüzde Varyans Testi	17
1.6.1.2.	Scree Testi	17
1.6.1.3.	Pareto Çizelgesi	18
1.6.1.4.	Tahmin Edilen Artık Hata Kareler Toplamı (PRESS)	19
1.6.2.	TBA ile Hata Tespiti.....	20
1.6.3.	Bağımsız Bileşenler Analizi (BBA)	22
1.6.3.1.	Kurtosis.....	25
1.6.3.2.	Negentropi	27
1.6.3.3.	Yaklaşık Negentropi	27
1.6.3.4.	Hızlı BBA Algoritması	28
1.6.4.	BBA ile Hata Tespiti	29
1.6.4.1.	Çekirdek Yoğunluk Tahmini	32
1.7.	Hata Teşhisi İçin Kullanılan Yöntemler	33
1.7.1.	Hata Sınıflama Yöntem Bilimi	33
1.7.2.	Makine Öğrenimi İçin Özellik Seçimi.....	34
1.7.2.1.	Sarıcı yöntemleri.....	35
1.7.3.	Destek Vektör Makineleri (DVM)	35
1.7.3.1.	Doğrusal Destek Vektör Makineleri.....	36
1.7.3.2.	Doğrusal Olmayan Destek Vektör Makineleri	38
1.7.3.3.	Çoklu Sınıflı Destek Vektör Makineleri.....	40
1.7.3.3.1.	Bire Karşı Kalanlar Yöntemi	40
1.7.3.3.2.	Bire Karşı Bir Yöntemi.....	41
1.8.	Uçak Sistemi ve Hareket Eksenleri	42
2.	YAPILAN ÇALIŞMALAR, BULGULAR VE İRDELEME	44
2.1.	Giriş	44
2.2.	Hata Tespiti İçin Elde Edilen Sonuçlar	46
2.2.1.	Ani Sensör Hata Tespiti İçin Elde Edilen Sonuçlar.....	46
2.2.2.	Artan Sensör Hata Tespiti İçin Elde Edilen Sonuçlar	51
2.2.3.	Geçici Sensör Hata Tespiti İçin Elde Edilen Sonuçlar	53
2.3.	Hata Teşhisi İçin Elde Edilen Sonuçlar	56
2.3.1.	Sınıflandırma İşlemi İçin Elde Edilen Hata Matrisleri	57
2.3.2.	Sınıflandırma İşlemi İçin Elde Edilen Sınıflandırma Başarım Oranı.....	68

3.	SONUÇLAR.....	72
4.	ÖNERİLER	74
5.	KAYNAKLAR.....	75
ÖZGEÇMİŞ		

Yüksek Lisans Tezi

ÖZET

UÇUŞ KONTROL SİSTEMLERİNDE SENSÖR HATALARININ TESPİTİ VE TEŞHİSİ

Behnam MAHMOUDI

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Ana Bilim Dalı
Danışman: Yrd. Doç. Dr. Yusuf SEVİM
2016, 80 Sayfa

Modern endüstride özellikle uçak endüstrisinde daha iyi bir üretim performansına duyulan ihtiyaç yüzünden, modern kontrol sistemleri gün geçtikçe karmaşık hale gelmektedirler. Geleneksel bir geri-beslemeli kontrol sistemi, aktüatör, sensör veya diğer sistem elemanlarındaki bazı arıza veya hatalardan dolayı hedeflenen başarıyı veya kararlılığı sağlamayabilir. Arızalar veya hatalar, sistemin dinamiğini değiştirip başarıyı azalmasına veya sistemin dengesiz çalışmasına yol açabilir. Dolayısıyla, kontrol sistemi tasarımında hata tespiti ve teşhisine ihtiyaç duyulmaktadır. Bir sürecin olağandışı durumlarının yönetimi için, ilk önce süreç durumunun olağandışı olup olmadığı belirlenmeli ve daha sonra olağandışı duruma neden olan aksaklıklar tespit edilmelidir. Günümüzde süreç üniteleri çok karmaşıktır ve süreç kontrol ve gözetimi için kullanılan süreç değişkenlerinin çeşitli ölçümlerini içermektedir. Böylece, süreç gözetimi için kafa karıştırıcı ve işlem hacmini artırıcı yüksek boyutlu veriler ile karşılaştırılır. Bu yüzden zamanında işlem yapabilen bağımsızlık ve ekonomi açısından da uygun olan akıllı bir hata tespit ve teşhis sisteminin tasarımı, çoğu araştırmacının dikkatini çekmiştir. Bu tezde, F4 Fantom savaş jetinin boylamasına modeli uçuş kontrol sistemlerinin bir örneği olarak seçilmiştir. Çok değişkenli istatistiksel teknikleri temel alan temel bileşenler analizi ve bağımsız bileşenler analizi metotları kullanarak modelin sensör hataları tespit edildi. Hatalar ise destek vektör makineleri yöntemi kullanılarak teşhis edildi.

Anahtar Kelimeler: Hata Tespiti ve Teşhisi, Uçuş Kontrol Sistemi, Temel Bileşen Analizi, Bağımsız Bileşenler Analizi, Destek Vektör Makineleri.

Master Thesis

SUMMARY

SENSOR FAULT DETECTION AND DIAGNOSIS IN FLIGHT CONTROL SYSTEMS

Behnam MAHMOUDI

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Electrical-Electronic Engineering Graduate Program
Supervisor: Assoc. Prof. Dr. Yusuf SEVİM
2016, 80 Pages

In the modern industry especially aircraft industry, due to the need for better performance, modern control systems are becoming increasingly complex. A conventional feedback control system due to error or fault of actuators, sensors or other components may not provide the targeted performance or stability. Faults or errors can change the dynamic of the system, reduce overall system performance or even lead to unstable work of the system. Therefore detection and diagnosis of faults in the design of a control system seems necessary. For managing unusual conditions of a process, it is necessary to determine whether the condition is unusual, and then identify the flaws causing the abnormal condition. Nowadays, process units are very complex and include numerous measurements of process variables that are used for controlling and monitoring the process. So, for monitoring a process, a user is often faced with high dimensional data that can cause confusion, and increase the volume of computations. Hence, designing an intelligent system for detection and diagnosis of faults that can act in a timely manner, and adequate immunity and economic, it has attracted the attention of many researchers. In this thesis, a longitudinal model of the F4 Phantom fighter jet was selected as an example of a flight control systems. In the model, the sensor faults were detected by using principal component analysis and independent component analysis methods based on multivariate statistical techniques. The faults were diagnosed by using support vector machines method.

Key Words: Fault Detection and Diagnosis, Flight Control System, Principle Component Analysis, Independent Component Analysis, Support Vector Machines.

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1.	Aktif hata toleranslı kontrol sistemlerinin genel yapısı 2
Şekil 2.	Hata tespiti ve teşhisi yöntemleri 6
Şekil 3.	Model tabanlı yöntem..... 7
Şekil 4.	Hataların modellenmesine göre a) toplamsal hata b) çarpımsal hata..... 12
Şekil 5.	Sistemde oluşum yerlerine göre hatalar 13
Şekil 6.	Zaman karakteristiklerine göre hatalar a) Ani hata b) Artan hata c) Geçici hata..... 14
Şekil 7.	Scree eğrisi 18
Şekil 8.	Pareto çizelgesi..... 19
Şekil 9.	F-dağılım 21
Şekil 10.	Kokteyl parti problemi 23
Şekil 11.	Gaussian, super Gaussian ve sub Gaussian için olasılık yoğunluk fonksiyonları..... 26
Şekil 12.	Yüzdelik L_2 norma karşılık bağımsız bileşen sayısı grafiği 30
Şekil 13.	Sarıcı yaklaşımının genel yapısı..... 35
Şekil 14.	DVM ve hiper-düzlem 37
Şekil 15.	Kernel fonksiyonu ile giriş uzayının doğrusal sınıflamanın çok boyutlu uzaya dönüşümü..... 39
Şekil 16.	Uçağın hareket eksenleri 42
Şekil 17.	Uçağın Simulink modeli..... 44
Şekil 18.	Ani sensör hatası 47
Şekil 19.	Ani sensör hatası için Pareto çizelgesi 48
Şekil 20.	Hatasız çalışan sistemden hesaplanan T^2 ve KTH istatistikleri..... 48
Şekil 21.	Ani sensör hatası için hesaplanan T^2 ve KTH istatistikler 49
Şekil 22.	Boyut azaltımı için kullanılacak olan bağımsız bileşenlerin sayısının grafiği 50
Şekil 23.	Hatasız çalışan sistemden hesaplanan I^2 , I_e^2 ve KTH değerleri ve ÇYT değerleri..... 50
Şekil 24.	Ani sensör hatası için hesaplanan I^2 , I_e^2 ve KTH istatistikleri 51
Şekil 25.	Artan sensör hatası 52
Şekil 26.	Artan sensör hatası için TBA algoritmasının T^2 ve KTH istatistikleri 53

Şekil 27.	Artan sensör hatası için BBA algoritmasının I^2 , I_e^2 ve KTH istatistikleri.....	53
Şekil 28.	Geçici sensör hatası.....	54
Şekil 29.	Geçici sensör hatası için TBA algoritmasının T^2 ve KTH istatistikleri.....	55
Şekil 30.	Geçici sensör hatası için BBA algoritmasının I^2 , I_e^2 ve KTH istatistikleri.....	55

TABLolar DİZİNİ

	<u>Sayfa No</u>
Tablo 1. Kernel fonksiyonları.....	40
Tablo 2. Uçağın durum değişkenleri ve kontrol girdileri	43
Tablo 3. Hata tespiti başarımı.....	56
Tablo 4. TBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi.....	58
Tablo 5. TBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi.....	59
Tablo 6. TBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi.....	60
Tablo 7. TBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi.....	61
Tablo 8. BBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi.....	63
Tablo 9. BBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi.....	64
Tablo 10. BBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi.....	65
Tablo 11. BBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi.....	66
Tablo 12. Skor değerleri kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan sınıflandırma başarıml oranı	68
Tablo 13. Skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan sınıflandırma başarıml oranı.....	69
Tablo 14. Skor değerleri kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan sınıflandırma başarıml oranı	69
Tablo 15. Skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan sınıflandırma başarıml oranı	70

SEMBOLLER DİZİNİ

HTT	: Hata Tespiti ve Teşhisi
HTKS	: Hata Toleranslı Kontrol Sistemler
TBA	: Temel Bileşen Analizi
BBA	: Bağımsız Bileşenler Analizi
KTH	: Kare Tahmin Hatası
T^2	: Hotelling'in T^2 İstatistiği
DVM	: Destek Vektör Makineleri
ÇYT	: Çekirdek Yoğunluk Tahmini
RTF	: Radyal Tabanlı Fonksiyon
V	: Öz Vektör Matrisi
P	: Yükleme Matrisi
T	: Skor Matrisi
k	: Öz Vektör Sayısı
A	: Karışım Matrisi
U	: Ortogonal Öz Vektör Matrisi
Λ	: Diyagonal Öz Değer Matrisi
α	: Hücüm Açısı
β	: Yana Kayma Açısı
V	: Hava Hızı
p	: Yatış Hızı
q	: Yunuslama Hızı
r	: Sapma Hızı
θ	: Yunuslama Açısı
φ	: Yatış Açısı
ψ	: Sapma Açısı
γ	: Uçuş Yolu Açısı
δ_e	: İrtifa Dümeni
δ_r	: Kuyruk Dümeni
δ_a	: Kanatçık

- a : Bileşen Sayısı
 C_a : Normal Dağılım Değeri
 d : Bağımsız Bileşenler Sayısı

1. GENEL BİLGİLER

1.1. Giriş

Geleneksel geri beslemeli kontrol sistemlerinde, hata oluşumunun ardından sistem istikrarı devam ettirilse bile, hata oluşumu esnasında istenilen başarıyı yakalamak oldukça zordur. Hata durumunda, kendiliğinden çalışmayı sürdürebilen dinamik sistemlere artan talep sebebiyle, endüstri (özellikle uçak endüstrisi) sistemlerdeki mevcut hataları saptayabilen ve hata bileşenlerini ayırabilen akıllı ve uyarlanabilir sistemler geliştirmek için zaman ve para harcamaktadırlar.

1970'den beri Hata Tespiti ve Teşhisi (HTT) teknikleri kapsamlı olarak çalışılmış ve hata oluşumu durumunda, hata ayırtmak ve hata azaltımının mümkün olduğu durumlarda kullanılmıştır. 1991 yılında, çalışmaların yaygınlaşmasıyla IFAC SAFEPROCESS komitesi kurulmuş ve 1993 yılında HTT hata ayırımı ve hataya dayanıklı denetim ile ilgili çeşitli tanımlamalar ve çalışmalar yapılmıştır [1,2].

Kabul edilebilir bir başarımlı seviyesi sağlayabildiği ve aktüatörlerde, sensörlerde ya da diğer sistem bileşenlerinde oluşabilecek hata durumlarında dahi sistem istikrarını sürdürebildiği için, geleneksel kontrol sisteminden farklı olan Aktif Hata Toleranslı Kontrol Sistemlerinin (AHTKS) ana kısmı olarak HTT'den faydalanılmıştır. Bundan dolayı, HTT'nin ana amacı, hata oluştuğunda hatalar hakkında bilgi (zamanı, çeşidi, büyüklüğü) sağlanmasıdır. HTT tarafından sağlanan bilgiye dayanarak, AHTKS ile başarılı bir sistem kontrolü yapılarak sistemin yeniden yapılandırılması sağlanabilir [3].

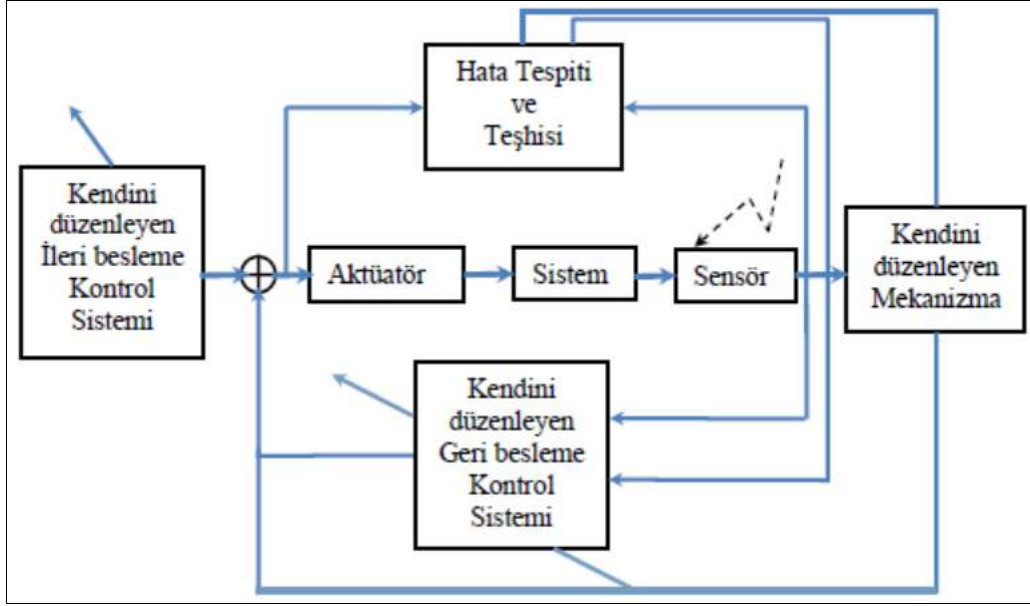
Hata Toleranslı Kontrol Sistemler (HTKS) aynı zamanda kendini onaran, yeniden yapılandırabilen ya da kendini düzenleyen kontrol sistemleri olarak da bilinirler ve genel olarak iki gruba ayrılabilirler:

1. Pasif HTKS (PHTKS); kontrolörün yeniden yapılandırılmasını içermez.
2. Aktif HTKS (AHTKS); bir HTT modülünden çıkan verilere dayanarak, kontrolörün yeniden yapılandırılması sağlanır.

PHTKS ve AHTKS arasındaki fark şudur; PHTKS'de bulunan kontrolörler, sistemin sabit ve meydana gelebilecek hataya karşı dirençli olabilmesi için tasarlanmışlardır. AHTKS kontrolörleri ise sistemde hata oluşması durumunda çevirim içi olarak sistemi

yeniden yapılandırılabilirler. AHTKS, sistem başarımını gözlemek ve sistemdeki hataları tespit etmek ve ayırmak için HTT işlemine bağımlıdır [4].

AHTKS'lerin genel yapısı Şekil 1'de görülmektedir. Bu tip sistemlerde, HTT yöntemleri vasıtasıyla hatalı parçaların algılanıp değiştirilmesiyle birlikte, güvenlik artırılmakta, onarım ve bakım masrafları da azaltılmaktadır.



Şekil 1. Aktif hata toleranslı kontrol sistemlerinin genel yapısı

HTKS'lerin kimyasal süreçlerde, uçak endüstrisinde, uzay araçları, nükleer güç sistemlerinde, türbinler, otomotiv endüstrisi vb. birçok mühendislik alanında çok sayıda uygulaması bulunmaktadır. HTKS ile ilgili uçak endüstrisinde özellikle uçaklardaki aktüatör ve sensör hatalarında kontrolün sağlanabilmesi amacıyla, farklı teorik çalışmalar yapılmıştır [5,6]. Askeri amaçlar için uçaklarda geliştirilecek olan HTT sistemi, uçuş sırasında oluşabilecek her türlü hatanın algılanmasını, belirlenmesini ve sonuç olarak savaş senaryosunda yaşama şansını artırılabilceğini göstermektedir. Vietnam savaşı ile ilgili ABD Hava Kuvvetleri tarafında meydana gelen uçak zararlarının %70'ine kadar önlenebileceği gerçeğini ortaya çıkarmıştır [7].

1.2. Uçak Kazaları ve Hata Toleranslı Kontrol Sistemlerinin (HTKS) Önemi

Uçak sistemlerindeki çok basit hataların motorların işleyişini durdurduğunu ve ya ekipmanların zarar görüp bozulmasına yol açtığını gösteren çok sayıda örnek var [8,9].

Örneğin:

- 1982 yılında, Aeroflot hava yollarına ait bir Ilyshine Il-62M uçağı, motor yangın uyarı sisteminin yanlış alarm vermesinden dolayı, motorları durmuş ve uçak yere düşmüştür. Bu uçak kazası 90 kişinin hayatına mal olmuştur.
- 1989 yılında, Amerika hava yollarına ait bir McDonnell Douglas DC-10 uçağı, merkez motordan kaynaklanan bir arıza yüzünden hidrolik sıvısını kaybetmiş ve yüksek bir hızla piste çarpmıştır. Bu uçak kazası 112 kişinin hayatına mal olmuştur.
- 1991 yılında, United Airlines'a ait bir Boeing B-737 uçağı, yatay sabitleyicisindeki deformasyonunun, hesaplananın üzerinde oluşması ile birlikte yatay sabitleyicinin kilitlenmesine neden olmuştur. Kilitlenen yatay sabitleyici, uçağın dönerek yere düşmesine neden olmuş ve bu kaza sonucunda 25 kişi hayatını kaybetmiştir.
- 1993 yılında, China Eastern hava yollarına ait bir McDonnell Douglas MD-11 uçağında, kontrol yüzey uyarıcıları istemsiz olarak çalışmış ve kontrol kanatlarının açılmasına neden olarak uçağın anlık irtifa kaybına sebep olmuştur. Bu kaza sonucunda 2 kişi hayatını kaybetmiştir.
- 1996 yılında, Birgenair hava yollarına ait bir Boeing 757'de hız sensöründeki pilot tüpü arızası, otomatik pilot ile pilotaj hatalarına neden oldu ve uçak Atlas Okyanusu'na düşmüştür. Bu kaza sonucunda 189 kişi hayatını kaybetmiştir.
- 1996 yılında, bir Arian-5 füzesi bir yazılımın kontrolöre yanlış veriler vermesi yüzünden patladı ve böylece 8,5 milyon dolarlık bir proje 37 saniye içerisinde boşa gitti.
- 2014 yılında, Sepahan Airlines'a ait bir HESA-Iran-140-100 uçağı, kalktıktan kısa bir süre sonra sağ tarafa doğru yatış yapmış ve irtifasını kaybederek bir yerleşim bölgesine çarpmıştır. Bu kaza sonucunda 39 kişi hayatını kaybetmiştir.

Bu tip kazaların meydana gelmesi HTKS ve HTT sistemlerinin öneminin artmasına yol açmaktadır.

Çoğu yöntemde, dinamik modellerin incelemesi ve ölçülebilir sinyallerin işlenmesi vasıtasıyla hatalar hakkında net bilgiler üretebilmektedir. Son yıllarda, hata sensörleri içeren ve hata anında yapısal değişiklik yapabilen sistemlerin tasarımı daha çok dikkat çekmektedir. Bu tür sistemlere hatayla uyum sağlayan kontrol sistemleri denir. Başarılı bir HTT sistemi ile birlikte, uçağın çalıştırılması aşamasında arıza veya hataların tespiti için ve bu hataların çözümleri için kabin görevlilerine yardımcı olunabileceği düşünülmektedir. Bir HTT sisteminin uçaklarda bulunması durumunda, uçakların arıza ya da tehlike derecesi azaltılmış olacak ve güvenli uçuş olasılığı artırılmış olacaktır.

1997 yılında NASA Havacılık Güvenlik Programı 500 milyon dolarlık sermaye ile birlikte, 10 yıl içinde %80 ve 25 yıl içinde %90 oranında ölümcül uçak kaza oranını azaltmak amacıyla uygun fiyatlı teknolojiler geliştirmek için başlatılmıştır [10]. Bundan dolayı, HTT sistemlerinin, hava araçlarının güvenli çalışması için havacılık sektöründe kritik ve gerekli olacağı açıkça görülmektedir.

Araştırmalara göre, ölümcül uçak kazalarının %53'ü uçuş ekibinin hatalarından, %20'si uçaktaki teknik hatalardan, %12'si hava şartlarından, %8'i sabotajdan, %6'sı insan faktörlerinden ve %1'i diğer hatalardan kaynaklanmaktadır [11]. Uçuş ekibinden kaynaklanan hataların eğitimle düzeltilebileceği düşünülürse, ikinci büyük hata sebebi olan teknik hataların HTT sistemi kullanılarak azaltılabileceği görülmektedir.

1.3. HTKS

Bir HTT sisteminin tasarımının amacı, hata veya arızanın uygun zamanda doğru ve duyarlı bir şekilde algılanması, yayılması ve istenmeyen zararlara engel olması ve sistemin çökmesini önlemektir. HTT sistemlerinin önemi özellikle çok tehlikeli veya kritik öneme sahip sistemlerde öne çıkmaktadır. HTKS aşamaları genel olarak aşağıdaki dört aşamaya ayrılmaktadır [12,13].

- Hata Tespiti
- Hata Ayrımı
- Hata Tanımı
- Hata gidermesi (yeniden yapılandırma)

Hata tespiti aşamasında sistemin çalışmasının normal olup olmadığı araştırılır. Aslında, süreç izleme sadece sistemde bir hata olup olmadığını belirler. Hatanın tespiti

aşamasında sorunların uygun bir çözümünün bulunulabilmesi için yeterli kadar fırsat sağlanır ve bu yüzden bu aşama büyük bir öneme sahiptir.

Hata ayrımı, hata tespitinden sonra, hatanın hangi bileşenden kaynaklandığının belirlenmesi işlemidir. Bu aşamada, sistemde olağan dışı duruma neden olan altkümeler belirlenir.

Hata tanımı aşamasında, olağan dışı duruma yol açan hatalar ve nedenleri belirlenir. Bu aşamada, hata ayrımı aşamasında belirlenmiş değişkenlerin değerlerine göre, sistemin hatalarının türü, yeri ve bazen oluş zamanı belirlenir. Hata tanımı en önemli aşamadır, çünkü bu aşamada elde edilen bilgilere göre hataların tanımlanması ve süreç durumunu normalleştirmek için gereken faaliyetlere karar verilebilir.

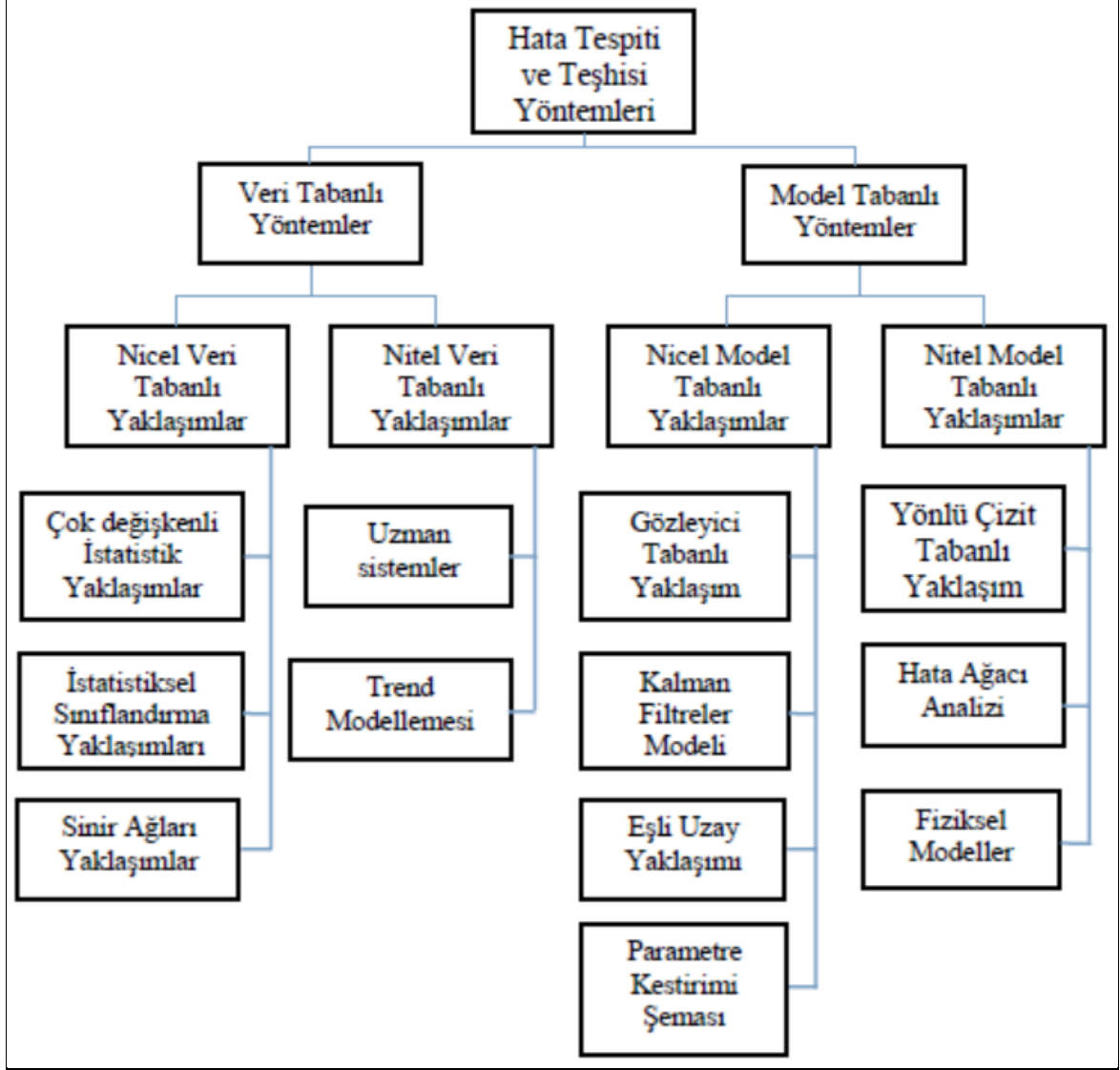
Hata tespiti ve tanımının yapılmasından sonra, sistemin yeniden düzenlenmesi ve normal çalışmasına geri dönmesi için yapılan faaliyetler yeniden yapılandırma olarak isimlendirilir.

Her hata durumunda HTKS'lerinin dört aşamasının uygulanmasına gerek yoktur ve bazı durumlarda sisteme göre bir veya birden fazla aşama atlanabilir. Örneğin hata ayrımı aşaması uygulanmadan, hata tanımlama aşaması kullanılarak sistemin hataları belirlenebilir.

Bu çalışmada, hata tespiti, hata ayrımı ve hata tanımlama aşamaları tek bir sistem olarak ele alınıp HTT sistemi olarak adlandırılmaktadır.

1.4. Hata Tespiti ve Teşhisi (HTT) Yöntemleri

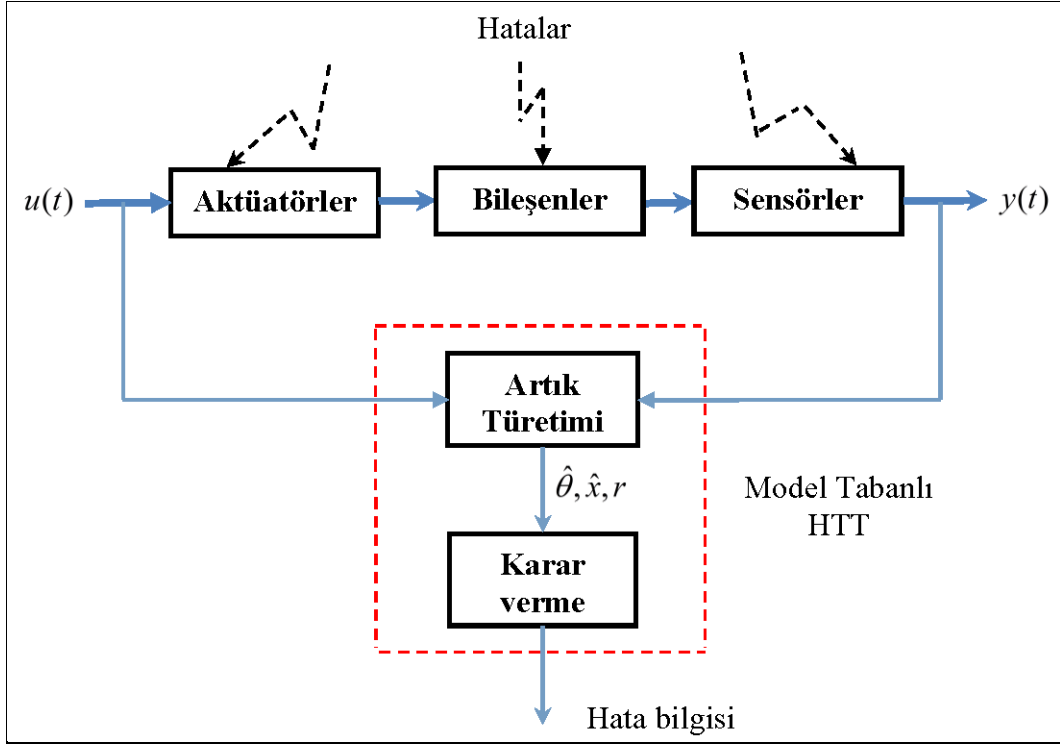
Farklı sistemler için oluşturulan HTT yöntemleri farklılıklar göstermektedir. Bu farklılıklar, sistemin özelliğine, hataların özelliğine, sistem başarımı ile davranışının bilinmesine, sistemden ölçülen sinyallerin türüne, sayısına ve hata içerip içermediğine göre belirlenmektedir. Genel olarak HTT yöntemleri, Şekil 2'de görülebileceği gibi, sinyal analiz yöntemine göre ve süreç modeline göre olmak üzere iki temel gruba ayrılır. Bu gruplar model tabanlı ve veri tabanlı HTT yöntemleri olarak adlandırılırlar [14-16]. Ayrıca bu iki yaklaşım Nicel ve Nitel yaklaşımlar olarak sınıflandırılabilirler.



Şekil 2. Hata tespiti ve teşhisi yöntemleri

1.4.1. Model Tabanlı Yöntemler

1970’li yıllardan günümüze kadar model tabanlı HTT yöntemleri üzerinde çalışma yapılmaktadır. Model tabanlı yöntemler iki adımdan oluşmaktadır. İlk adım arta kalan olarak adlandırılan dizgenin mevcut ve beklenen değerleri arasındaki fark işaretlerinin türetimidir. İkinci adım ise elde edilen bu işaretleri kullanarak bir karar kümesine bağlı olarak hata tespiti işleminin yapılmasıdır. Bu yöntemlerde, sistemin girişi $u(t)$, çıkışı $y(t)$, parametre tahminleri $\hat{\theta}$ ya da durum tahminleri $\hat{x}$, ve arta kalan sinyaller $r(t)$ gibi parametreler incelenmektedir [1,17,18].



Şekil 3. Model tabanlı yöntem

1.4.1.1. Nicel Model Tabanlı Yaklaşımlar

Nicel model tabanlı yaklaşımlardaki çalışmaların çoğu arta kalan sinyalleri üretmek için genel girdi-çıkı (diferansiyel denklemler) ve durum uzay modellerini temel almaktadır. Nicel modeller kendi aralarında gözleyici, Kalman filtresi, eşli uzay ve parametre kestirim yöntemleri olarak sınıflandırılabilir.

1.4.1.1.1. Gözleyici Tabanlı Yaklaşım

Bu yaklaşımda, gözlemcilerin yardımı ile ölçümlerden veya ölçümlerin bir alt kümesinden sistemin çıkışı yeniden yapılandırılır. Ölçülen çıkışlar ve tahmini çıkışlar arasındaki farklılık arta kalan sinyal vektörü olarak kullanılmaktadır [19].

1.4.1.1.2. Kalman Filtreler Modeli

Tekrarlamalı algoritmalar olan Kalman filtreler, sistem normal çalışma durumundayken sistem modelini oluşturmak için kullanılmaktadır. Bu yöntemde, filtre kazanç süre değişkenidir ve mevcut çalışma noktası etrafında doğrusallaştırılır. Teorik olarak, Kalman filtresi durumla doğrusal olarak ilgili olan ve beyaz gürültüyle bozulan ölçümler kullanılarak beyaz gürültünün karıştığı doğrusal dinamik bir sistemin anlık durumlarını kestirme problemi olan doğrusal kuadratik problemin bir tahmincisidir. Tipik etkin özyinelemeli bir filtre olan Kalman filtresi sıklıkla beyaz gürültü bozulmaları altındaki dinamik bir sistemin durumlarını kestirmek için kullanılır [17].

1.4.1.1.3. Eşli Uzay Yaklaşımı

Eşli uzay yaklaşımı, ölçümler ile sistemin matematik eşitliklerinin tutarlılığını kontrol eden bir yöntemdir. Bu yöntemde, önceden belirlenmiş hata bantları kullanılarak hata tespiti yapılmaktadır. Arta kalan sinyaller, doğrudan ölçümler ile hesaplandığı için, bu yöntem ölçüm gürültüsüne ve sistemde meydana gelen problemlere karşı çok hassastır [20].

1.4.1.1.4. Parametre Kestirimi Şeması

Bu yöntemde, sistemi temsil eden matematik modelinin parametreleri kestirilmekte ve bu kestirim parametreleri ile gerçek parametreler karşılaştırılarak hatalar tespit edilmektedir. Parametre kestirimi için kullanılan şemalar, gerçek zamanlı bulunan çevrimiçi sistem parametrelerini bir yol olarak sağlar ve bu parametrelerin yeniden yapılandırması için kullanılabilir [20,21].

1.4.1.2. Nitel Model Tabanlı Yaklaşımlar

Nitel model tabanlı yaklaşımlar, fonksiyonlara dayalı beklenen sistem davranışını belirleyecek şekilde değişkenler arasındaki bağlantıyı göstermektedir. Nitel modeller; diyagramlar, hata ağaçları ve nitel fiziksel yöntemleri olarak sınıflandırılabilir.

1.4.1.2.1. Yönlü Çizit Tabanlı Yaklaşım

Sebeup-sonuç ilişkisini ve modeli tanımlamak için imzalı yönlü çizitleri (İYÇ) kullanan yaklaşımlardır. İYÇ'lerin her düğümü bir etkinliği veya değişkeni temsil etmektedir ve kenar düğümler arasındaki ilişki anlamına gelmektedir. İYÇ'de düğümler arasındaki değişimi tanımlamak için yöneltilmiş kenarları kullanır. İYÇ'ler görsel ve doğrudan bir yolla modelin tanımlanmasını sağlar ve hata tespiti süreci sırasında geniş çapta kullanılmaktadırlar[15].

1.4.1.2.2. Hata Ağacı Analizi

Hata ağacı analizi (HAA) yöntemi 1961 yılında Bell Telefon Laboratuvarları'nda geliştirilmiştir. Bu yöntem AND, OR veya XOR gibi mantık operasyonları kullanan düğüm katmanlarından oluşur. Hata ağaçları genellikle sistem güvenilirlik analizinde kullanılır [15].

1.4.1.2.3. Fiziksel Modeller

Fiziksel modeller, bir sistemdeki bütün içeriklerin fiziksel ilişkileri ve karakteristikleri üzerindeki detaylı bilgileri içerir. Mekanik sistemlerde bu detaylı bilgilerin kullanımı ile kütle, moment, enerji dengeleri, ısı ve kütle transfer ilişkileri için detaylı bir matematik denklem takımı oluşturulur ve çözülür. Fiziksel modellere ilk yaklaşım, birleşik denklemler olarak bilinen eş-gidiş denklemlerden niteliksel denklemlerin türetilmesidir [22]. Sistemlerin niteliksel olarak modellenmesi ve etken bilgilerin temsili alanında hatırı sayılır çalışmalar yapılmıştır [23]. Nitel fiziksel modeller hususlarındaki diğer yaklaşım Adi Diferansiyel Denklemlerden niteliksel davranışın türetilmesidir.

1.4.2. Veri Tabanlı Yöntemler

Veri tabanlı HTT yöntemleri, birçok araştırmacı tarafından kullanılmaktadır [24,25]. Bazı araştırmacılar da veri tabanına dayalı kontrol yaklaşımlarını belirli amaçlara ulaşmak için kullanmışlardır [26]. Fakat birkaç çalışmada veriye dayalı HTKS için ele alınmıştır

[27-29]. Bu yöntemlerde, sistemden elde edilen bilgiler, sistemden örneklenen sayısal sinyaller ile sınırlıdır. Bu sinyaller tüm veya birkaç girişler, çıkışlar ile durum değişkenlerinden elde edilmektedir. Genel olarak, bu veriler test kümesi ve eğitim kümesi olarak kullanılmaktadır. Bu yöntemlerde, sistem bilgileri kullanılan algoritmaların özelliği, örnek sayısı, örnek kalitesi ve sistemin karmaşıklığı gibi faktörlere bağlı olmaktadır. Başka bir deyişle, bu yöntemlerin en büyük sorunu, verinin kalitesine yüksek bağımlılığının olmasıdır. Tüm bunlara rağmen, sadece sistem verilerine dayanan yöntemler, genellikle düşük maliyet ile uygulanabilir. Bu nedenle, bu yöntemler yaygın olarak, endüstri ve diğer sektörlerde, örüntü tanıma, sistem tanıma, biyomedikal sinyal işleme, jeodezi sinyal işleme, makine öğrenme gibi diğer uygulamalarda kullanılmaktadır. Veri tabanlı yöntemler nitel ve nicel olmak üzere iki alt kola ayrılmaktadır [16].

1.4.2.1. Nitel Veri Tabanlı Yaklaşımlar

1.4.2.1.1. Uzman Sistemler

Bu sistemler Feigenbaum tarafından “çözümleri için insan uzmanlığı gerektirecek kadar zor olan sorunları çözmek için bilgi ve müdahale prosedürleri kullanan akıllı bir bilgisayar programı” olarak tanımlanmıştır [30]. 1984’den beri uzman sistem hakkında çok sayıda yöntem geliştirilmiştir.

1.4.2.1.2. Trend Modellemesi

Trend modellemesi, gelecek durumların tahmin edilmesi ve hataların bulunması için bir işlemde gerçekleşen önemli etkinlikleri açıklamak için kullanılmaktadır. Cheung ve Stephanopoulos, işlem trendlerinin temsili için resmi bir çerçeve oluşturmuşlardır [31]. Yakın zamanlarda, Rengaswamy ve Ark., kontrol döngüsü performansının değerlendirilmesinde trend modelleme hizmetini kullanmışlardır [32].

1.4.2.2. Nicel Veri Tabanlı Yaklaşımlar

Nicel veri tabanlı yöntemler, istatistiksel olan ve istatistiksel olmayan yöntemler olarak tanımlanabilirler. Sinir ağları istatistiksel olmayan sınıflandırıcıların önemli bir sınıfıdır. Temel Bileşen Analizi (TBA), Bağımsız Bileşenler Analizi (BBA), Kısmi En Küçük Kareler (KEKK) ve istatistiksel örüntü sınıflandırıcıları istatistiksel özellik çıkartma yöntemlerinin önemli bir parçasını teşkil eder.

1.4.2.2.1. Çok Değişkenli İstatistik Yaklaşımlar

Birden fazla değişkene sahip durumlarda, bir işlemin tek değişkenli duruma karşın normal çalışma bölgesi altında olma olasılığı daha azdır. Benzer şekilde, tamamen kontrol dışında olan çok değişkenli bir işlem tek değişkenli durumunkinden daha azdır. Çok değişkenli kontrol diyagramlarının kullanımı ile istenilen kontrol seviyesi, değişkenler arasındaki çapraz korelasyon bilgilerinin kullanılması ile korunabilir. Böylece, süreç, kontrol diyagramında karmaşıklık olmaksızın sistemin kararlılığı için analiz edilebilir. Buna ek olarak, çok değişkenli istatistik süreç izleme (ÇDİSİ) yöntemleri, tahmini yapılan gizli değişkenler ile elde edilen Hotelling'in T^2 istatistiğini kullanmaktadır [33,34]. ÇDİSİ yöntemleri için yaygın bir şekilde kullanılan başka bir ölçüm yöntemi ise Kare Tahmin Hatası (KTH) olarak bilinen Q istatistiğidir. KTH sinyalden arta kalan değişkenin ölçümüdür ve aynı zamanda süreç izleme için kullanılmaktadır [35]. Klasik ÇDİSİ yöntemleri, örneğin TBA ve KEKK gibi metotlar süreç izleme problemlerinde kullanılmıştır. Bu yöntemler kullanılarak, yüksek ölçüde ilişkili değişken takımı ilişkisiz değişken takımına dönüştürülmektedir.

1.4.2.2.2. İstatistiksel Sınıflandırma Yaklaşımları

Hata teşhisi temel olarak bir sınıflandırma problemidir ve klasik istatistiksel örüntü tanıma çerçevesine dâhil edilebilir. İstatistiksel sınıflandırma metotlarında, sistemin hata durumlarının özellik bilgileri kullanılarak anlık sınıflandırma tahminleri ve zaman aşımı gibi sorunları değerlendirilmektedir [36].

1.4.2.2.3. Sinir Ağları Yaklaşımları

Hata teşhisi için kullanılan sinir ağları metotları iki gruba ayrılabilir: (i) kıvrık, radyal tabanlı ağ mimarisi ve (ii) denetlenen ve denetlenmeyen öğrenme stratejileri. Sinir ağlardaki en popüler denetlenen öğrenme stratejisi ters yayılım algoritmasıdır. Ters yayımlı sinir ağları yöntemini kullanarak hata tanılama yapmaya çalışan çeşitli çalışmalar vardır [37]. Watanabe ve Ungar hata teşhisi için sinir ağları metotlarını kullanılabileceğini gösteren ilk araştırmacılar arasındadır [38,39].

1.5. Hata Türleri

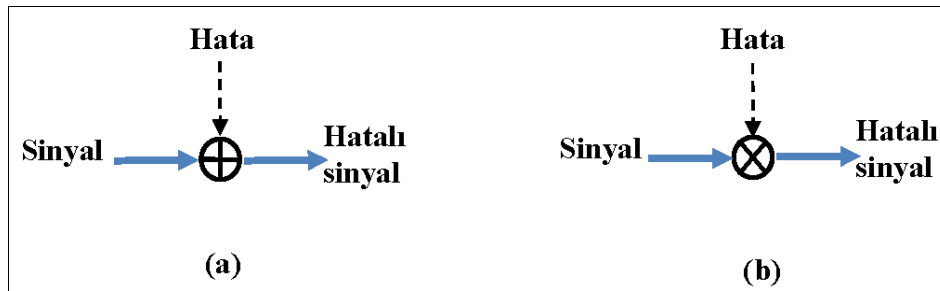
Sürecin normal çalışma durumundan sapmasına neden olan ve kontrol edilen sürecin farklı bölgelerinde yer alabilen istenmeyen bir değişikliğe hata denir. Bir hatanın zamanında algılanamaması ve sistemde oluşumunun önlenememesi sürecin bir kısmı veya tamamının çalışmamasına neden olabilir. Örneğin, bir motorun yakıt geçiş yolunda bir deliğin bulunması önce motor vasıtasıyla üretilen gücün azalmasına ve hata bulunamaz ise daha sonra motorun tamamen durmasına neden olur. Hatalar, sistemdeki oluşum yerlerine göre, modellenmesine göre ve zaman karakteristiklerine göre çeşitli kriterlere dayalı sınıflandırılabilir [1,40]. İlk sınıflandırma ise hataların modellenmesine göre yapılmaktadır:

a) Toplamsal Hatalar

Bu tip hatalar sistemin giriş ve çıkışında ek bir değer olarak gözlemlenir.

b) Çarpımsal Hatalar

Bu tip hatalar sistemin parametrelerin değişimine neden olurlar ve çarpan olarak sisteme eklenirler.



Şekil 4. Hataların modellenmesine göre a) toplamsal hata b) çarpımsal hata

İkinci sınıflandırma işlemi, hataların sistemdeki oluşum yerlerine göre yapılmaktadır:

i) Aktüatör Hatası

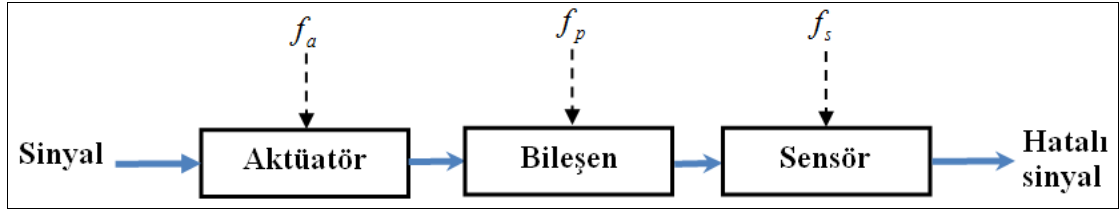
Aktüatör, dış enerjiyi gerekli harekete dönüştüren bir mekanik cihazdır ve herhangi bir kontrol sisteminde gerekli bir bileşendir. Aktüatörler yaygın olarak endüstriyel uygulamalarda kullanılmaktadır. Bu hata, kontrol etkisinin kısmen veya tamamen kaybını temsil etmektedir.

ii) Sensör Hatası

Sensör, sistemin iç durum sinyallerini ölçen ve daha sonra karşılık gelen sinyal çıktılarına dönüştüren bir cihazdır. Sensör hatası, sensörlerden gelen yanlış değerleri temsil etmektedir. Ayrıca sensör hataları bölümsel ve toplamı olarak alt bölümlere ayrılabilir.

iii) Bileşen Hatası

Bu hatalar, sistemin dinamiklerinde oluşur, sistemin giriş ve çıkış parametrelerinin değişimine neden olur.



Şekil 5. Sistemde oluşum yerlerine göre hatalar

Üçüncü sınıflandırma ise hataların zaman karakteristiklerine göre yapılmaktadır:

a) Ani Hatalar

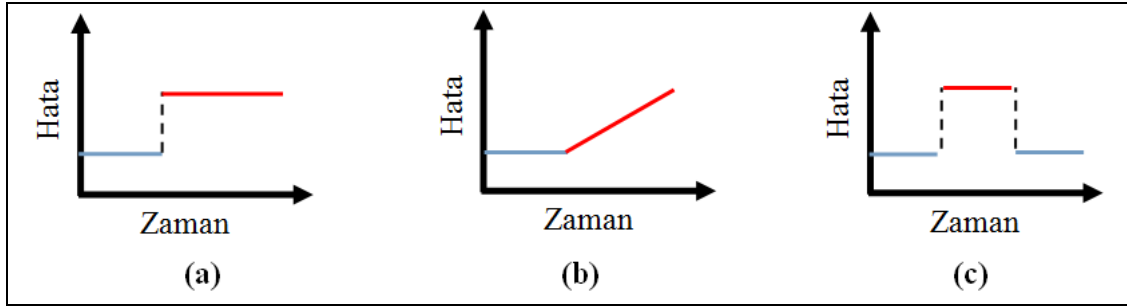
Bu tip hatalar, zaman düzleminde aniden ortaya çıkan hatalardır. Bu tür hatalarda, sistemin bir veya birden fazla değişkeni normal çalışma durumundan çıkıp bir daha eski durumuna dönmezler. Düzensizlik ve gürültü gibi dış etkenler sistemde atlamalara neden olabilir, fakat bu atlamalar genellikle bir anlık olup hızlı bir şekilde ortadan kaybolurlar. Buna ek olarak, genellikle düzensizlik ve gürültü gibi etkenlerin büyüklüğü, hataya göre çok az olur. Bu yüzden, bu tür hatalar diğer hatalara göre daha basit bir şekilde tespit edilebilir.

b) Artan Hatalar

Bu tip hatalar zamanla artarak değişen hatalardır. Artan hatalar, yavaş yavaş bir veya birden fazla değişkeni normal çalışma durumundan çıkarır. Bu tür hatalar, etkisini zamanla gösterdiği için, genellikle en zor tespit edilen hata türüdür.

c) Geçici Hatalar

Geçici hatalar aniden ortaya çıkar ve kaybolurlar. Bu tür hatalarda, bir anlık olarak sistemin bir veya birden fazla değişkeni normal çalışma durumundan çıkar ve bir süre sonra tekrar normal çalışma durumuna döner. Geçici hataların tespiti genellikle türüne ve etkisinin nasıl olduğuna bağlıdır. Eğer hatanın büyüklüğü ve süresi, muhtemel gürültü ve düzensizlik etkilerinden ayrılabilir şekilde olursa, bu tür hatalar kolayca tespit edilebilir.



Şekil 6. Zaman karakteristiklerine göre hatalar a) Ani hata b) Artan hata c) Geçici hata

HTKS'ler ile ilgili verilen açıklamalara göre, HTT karmaşık sistemler için kullanımları kaçınılmazdır. Bu tezin amacı sensör hataları için TBA ve BBA yöntemleri vasıtasıyla hata tespiti işlemini ve DVM yöntemi vasıtasıyla hata teşhisi işlemini gerçekleştirmektir. Bunun için ilk olarak, TBA ve BBA yöntemler ile doğrusal olmayan sistemlerde normal çalışma bölgesinin sınırları belirlenmiş ve bu sınırları aşan sistem çıkışının hatalı olduğunun tespiti yapılmıştır. Bu işlem yapısı veri tabanlı HTT yöntemlerinden sayılmaktadır. Daha sonra bulunan hatanın teşhisi için çoklu DVM sınıflandırıcısı oluşturulmuştur. Çoklu DVM, sınıflandırma işlemi çok sayıda ikili sınıflandırıcı birleştirilmesiyle veya bire-karşı-kalanlar şeklinde gerçekleştirilir.

1.6. Hata Tespiti İçin Kullanılan Yöntemler

1.6.1. Temel Bileşen Analizi

Temel Bileşen Analizi (TBA), asal bileşen analizi olarak 1901 yılında Karl Pearson tarafından tanımlanmıştır ve daha sonra 1930 yılında Hotteling ve 1964 yılında Rao tarafından geliştirilmiştir [41].

TBA yüksek boyutlu veri kümelerini düşük boyutlara indirgeme ve değişkenler arasındaki bağımlılığı yok etme tekniği olarak kullanılmaktadır. TBA çok sayıdaki orijinal değişken uzayları arasındaki doğrusal ilişkiden (korelasyon) yararlanarak ve en az bilgi kaybıyla daha az sayıda değişken ile veriyi temsil eden bir çok değişkenli analiz yöntemidir. TBA daha önce ortaya çıkarılmamış ilişkileri ortaya çıkartmakta, sıradan sonuçlar diye nitelenmeyecek tahminler yapmaktadır.

Literatürde TBA yöntemi ile hata tespiti konusunda kayda değer miktarda araştırmalar mevcuttur [42-44]. TBA yöntemlerinde, eğitim için hata içeren ve hatasız verilere ihtiyaç vardır.

m adet gözlem ve n adet değişkenden oluşan X veri matrisi düşünülecek olursa ve veri matrisi $m \times n$ boyutunda olursa, bu vektörler aşağıdaki optimizasyon denkleminin çözülmesiyle hesaplanır:

$$\max_{v \neq 0} \frac{v^T X^T X v}{v^T v} \quad (1)$$

Çok değişkenli veri analizinde, çoğu zaman değişkenlerin ölçü birimlerinin birbirinden farklı olmaması için ön işlemler yapılır. Ön işlem olarak ilk önce, değişkenlerin ortalama değerleri değişkenlerden çıkarılarak değişkenler merkezileştirilir.

$$x = x - E\{x\} \quad (2)$$

İkinci ön işlem olarak her bir değişken kendi standart sapmasına bölünerek aynı birime dönüştürülür.

$$\hat{x} = \frac{x}{\sigma^2(x)} \quad (3)$$

burada σ^2 standart sapmadır.

Sonraki aşamada ortalaması sıfır olan gözlem matrisi transpozesi ile çarpılarak kovaryans matrisi elde edilir:

$$S = \frac{1}{m-1} X^T X \quad (4)$$

Hesaplanan kovaryans matrise ait öz değer ve öz vektörler Tekil Değer Ayırıştırma (TDA) yöntemi kullanarak hesaplanır [45]:

$$S = V \Lambda V^T \quad (5)$$

burada V öz vektör matrisi ve Λ kovaryans matrisin öz değerleridir. Λ öz değer matrisi λ_i öz değerlerini azalan bir sırayla ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$) içerir ve diyagonal matristir.

Bu hesaplanan öz değerlerin yüksek değere sahip olanları veriyi daha iyi temsil eden öz vektörlere karşılık düşmektedir.

Bütün vektörler arasında, veri standart sapmaları kabul edilebilir derecede büyük olan vektörler seçilir ve yükleme matrisi P oluşturulur. Yükleme matrisi kullanılarak, skor matrisi T aşağıdaki gibi hesaplanır:

$$T = XP \quad (6)$$

Yukarıdaki denklem kullanarak skor matrisi orijinal uzaya aşağıdaki gibi dönüştürülür:

$$\hat{X} = TP^T \quad (7)$$

Temel bileşenlerin sayısını belirlemek için yüzde varyans testi, scree testi, pareto çizelgesi ve tahmin edilen artık hata kareler toplamı (PRESS) istatistiği gibi çeşitli teknikler kullanılmaktadır.

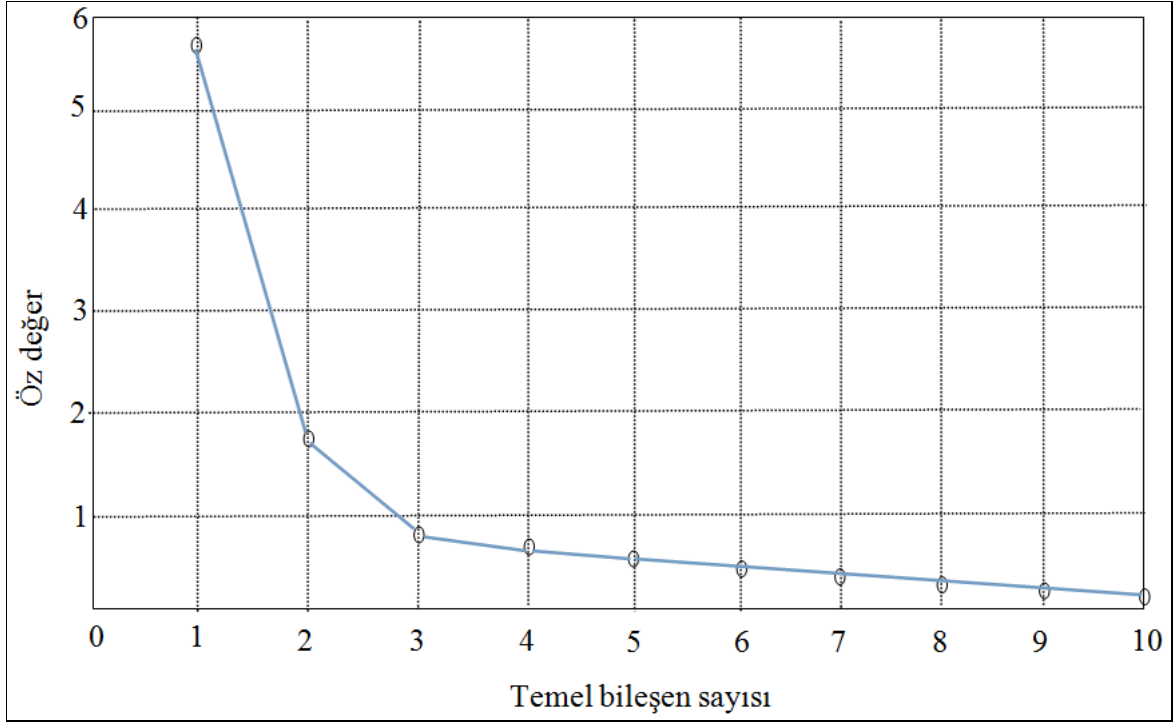
1.6.1.1. Yüzde Varyans Testi

Bu yöntem, toplam varyansının 2/3'ünü geçinceye kadar λ değerleri toplanarak bileşen sayısına karar verme yöntemidir. Eğer yeni bulunacak bileşen sayısına k denirse bunun sayısı ve toplam varyansdaki payı aşağıda verilen denklem yardımı ile hesaplanır [46]:

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i} > \frac{2}{3}, \quad k = 1, 2, \dots, p \quad (8)$$

1.6.1.2. Scree Testi

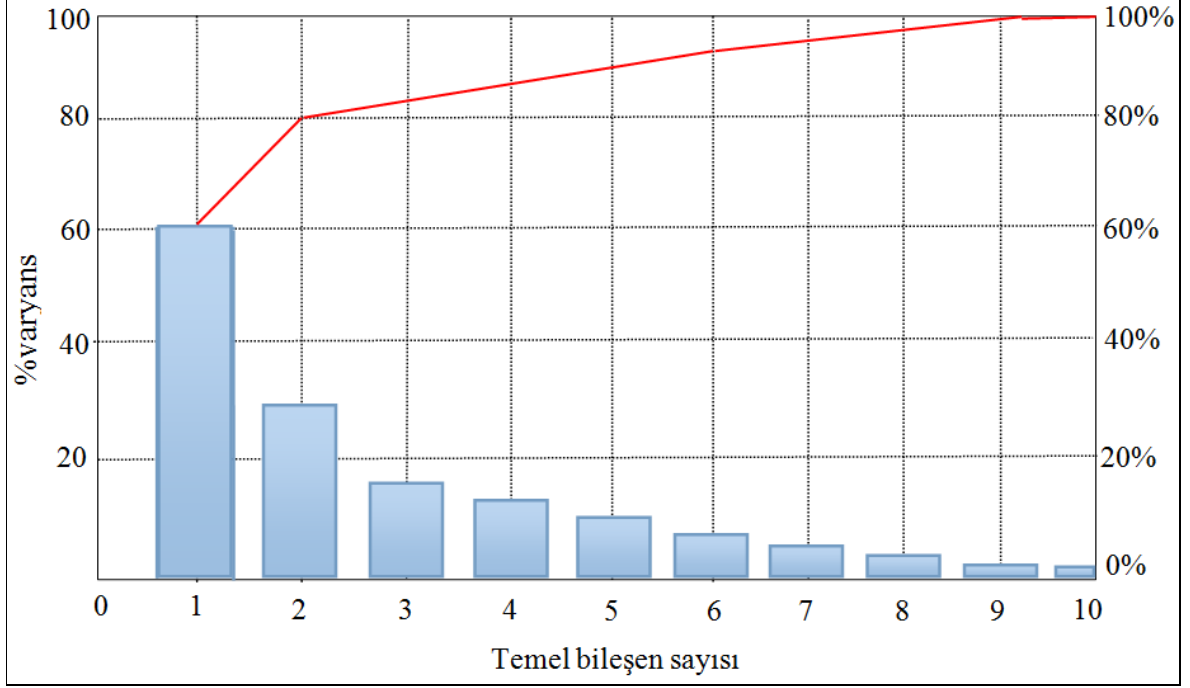
Bu yöntem, Şekil 7'de gösterilmiştir ve öz vektörler, öz değerlerinin genliklerine göre sıralanır. % 80-90 değişim düzeyine sahip öz vektörler temel bileşen olarak seçilir. Temel bileşenlerin sayısı arttıkça, öz değerlerin genlikleri azalır [47].



Şekil 7. Scree eğrisi

1.6.1.3. Pareto Çizelgesi

Bu yöntem, önceden bahsedilen iki metodun bir birleşimidir [47]. Şekil 8’de yatay eksen her öz değerin birikimli yüzdesini belirtir. Çubuk grafiklerde azalan sıralama her değişkenin yüzde değişken katkısını gösterir. Yatay eğrinin kırılmasını gözlemleyerek k azaltılmış veri sayısı seçilebilir. Eğer iki tane kırılma varsa, çubuk grafiğe bakılabilir ve seçilecek veri sayısının toplam değişimi üzerinden karar verilir. Şekil 8’deki Pareto çizelgesine bakarak $k = 2$ ’nin uygun seçim olabileceği gözlemlenebilir.



Şekil 8. Pareto çizelgesi

1.6.1.4. Tahmin Edilen Artık Hata Kareler Toplamı (PRESS)

Bu yöntemde, ana sinyalleri temsil etmesi gereken k azaltılmış veri sayısı aynı zamanda çapraz doğrulama ile artı kalan matrisinin kullanımıyla da belirlenebilir. Bu test **PRESS istatistiği** olarak bilinmektedir ve

$$PRESS(k) = \frac{1}{mp} \|X - \tilde{X}\|_F^2 \quad (9)$$

denklemleri ile verilir. Burada k 'nın seçilen veri sayısı olduğunu ve $\|\cdot\|_F$ 'nin Frobenius norm olduğunu göstermektedir. Bu tekniği uygulamak için X veri matrisi daha küçük alt gruplara bölünebilir ve $PRESS$ istatistiği k azaltılmış veri sayısı için hesaplanır. Bu adım her defasında farklı k değişkeninin kullanımı ile bütün alt gruplar için tekrar edilir. Bütün farklı alt gruplar için daha küçük $PRESS$ istatistiğinin ortalamasını alan k değeri seçilir [20].

1.6.2. TBA ile Hata Tespiti

Hata tespiti için, normal çalışma sürecinden elde edilen veriler kullanarak hata tespiti yapılmaktadır. Bunun için her gelen yeni veri vektörü otomatik olarak ölçeklendirilmektedir. Ölçeklendirme işlemi normal çalışma sürecinde elde edilen istatistiksel bilgiler ile yapılmaktadır.

Daha önceden bahsedildiği gibi, KTH ve Hotelling'in T^2 istatistiği yöntemleri sürecin kontrol dışına çıktığını saptayıp bunu bir sinyalle belirtir. Bu istatistik yöntemlerin kontrol sınırları vardır ve sınırı geçen noktalar bir kontrol-dışı sinyal olarak değerlendirilir. Bu istatistik yöntemler ile çok değişkenli gözlemleri tek bir değere indirgeyen bir fonksiyon ile elde edilir. Bu fonksiyon, değişkenler arasındaki kovaryans matrisini (etkileşimini) içerir [48].

T^2 istatistiği, çok değişkenli süreç izleme amacıyla kullanılan çok değişkenli süreç kontrol metodudur. n değişken ve m gözlemden oluşan eğitim kümesini düşünürsek, veri matrisi aşağıdaki gibi ifade edilir:

$$X = \begin{bmatrix} x_1(1) & \dots & x_n(1) \\ \vdots & \ddots & \vdots \\ x_1(m) & \dots & x_n(m) \end{bmatrix} \quad (10)$$

Hesaplanan T^2 istatistiği her yeni bir gözlem vektörü x_i için TBA tarafından aşağıdaki şekilde hesaplanır:

$$T^2 = x_i^T V A_a^{-1} V^T x_i \quad (11)$$

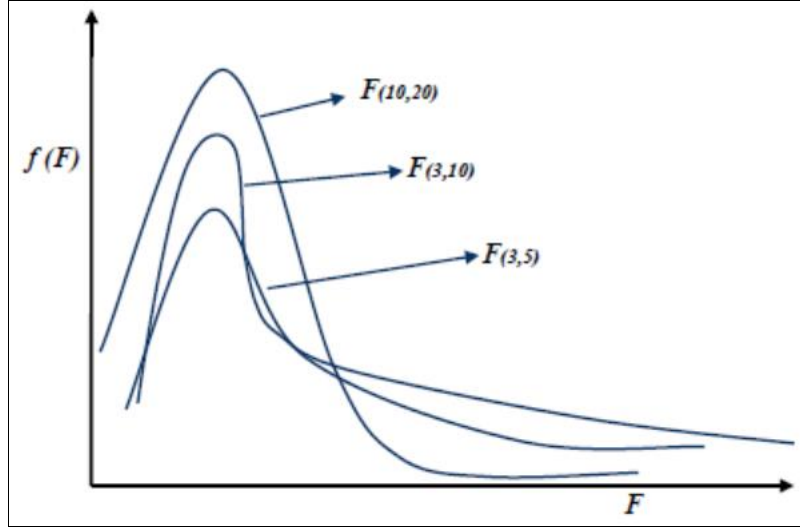
burada, x_i bir vektör ve A_a , diyagonal matrisin ilk a satır ve sütunlarından oluşturulmuş matristir.

Süreç ölçümlerinin çok değişkenli normal dağılıma sahip olduğu varsayılarak, kontrol sınırı hesaplaması F-dağılım ile ilişkilidir ve aşağıdaki gibi ifade edilir:

$$T_a^2 = \frac{a(m^2 - 1)}{m(m - a)} F_a(a, m - a) \quad (12)$$

burada $F_a(a, m-a)$, F-dağılımının m ve $m-a$ serbest dereceli kritik bir değeri, m örnek sayısı ve a değeri kontrol sınırını belirtir. a değerinin 95% ve 99% arasında yer alması tavsiye edilmektedir [49].

Çeşitli serbestlik derecelerine göre düzenlenmiş olan F-dağılımının grafiği Şekil 9'da görülmektedir. Şekil incelendiğinde dağılımın sağa çarpık olduğu görülmekte ve serbestlik derecesi büyüdükçe dağılımın dikliği artmaktadır [50].



Şekil 9. F-dağılım

Hesaplanan KTH istatistiği her yeni gözlem vektörü x_i için TBA yönteminde aşağıdaki gibi ifade edilir:

$$KTH = (x_i - \hat{x}_i)(x_i - \hat{x}_i)^T = x_i(I - PP^T)x_i^T \quad (13)$$

burada $\hat{x}_i$ çok değişkenli gözlemin tahminidir. KTH için kontrol sınırı aşağıdaki denklemle ifade edilir:

$$\begin{aligned}
KTE_{\alpha} &= \theta_0 \left[\frac{C_{\alpha} \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right] \frac{1}{h_0}, \\
\theta_i &= \sum_{j=a+1}^n \delta_j^{2i} \\
h_0 &= 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2}
\end{aligned} \tag{14}$$

burada C_{α} , normal dağılım değeri ve α anlamlılık düzeyidir [45].

TBA'nin bir diğer önemli faydası KTH ve T^2 süreç değişkenlerin katkılarını belirleme yeteneğidir. Bu katkılar arıza teşhisi için de kullanılabilir. Süreç izleme için, bu katkılar şu şekilde hesaplanabilir:

$$C(T^2, i) = \frac{t_i}{\sigma_i^2} P_{i,j} x_j \tag{15}$$

burada $P_{i,j}$, yükleme matrisinin (i, j) . elemanını temsil etmektedir.

$$C(KTH, i) = e_{yeni,i} (k)^2, \quad e_{yeni} = (x_i - \hat{x}_i) \tag{16}$$

1.6.3. Bağımsız Bileşenler Analizi (BBA)

BBA algoritmaları Kör Kaynak Ayırıştırma (KKA) yöntemlerinin en yaygın olanıdır. Sinyalleri bağımsız sinyallere ayırıştırma fikri 1991 yılında Hearault, Jutten ve Ans tarafından ortaya çıkmıştır [51,52]. Daha sonra Comon tarafından lineer BBA teorisi 1994 yılında yayınlanmıştır [53].

BBA, doğrusal kombinasyonlar olarak ölçülen değişkenlerden birçok bağımsız değişken bulmak için kullanılan ve ayrıca boyut indirgeme için de kullanılan yöntemdir. BBA giriş verisinin bulunduğu koordinat sistemi üzerindeki izdüşümlerini birbirinden bağımsız hale getirir ve aralarındaki ilişkiyi minimum seviyeye indirir. BBA'nin ana amacı, verilen bir veri dizisi içerisinde istatistiksel olarak bağımsız sinyalleri bulmaktır

[54]. Bu amaç için Hızlı BBA, Jade ve Infomax gibi kullanılan çeşitli BBA yöntemleri mevcuttur [17].

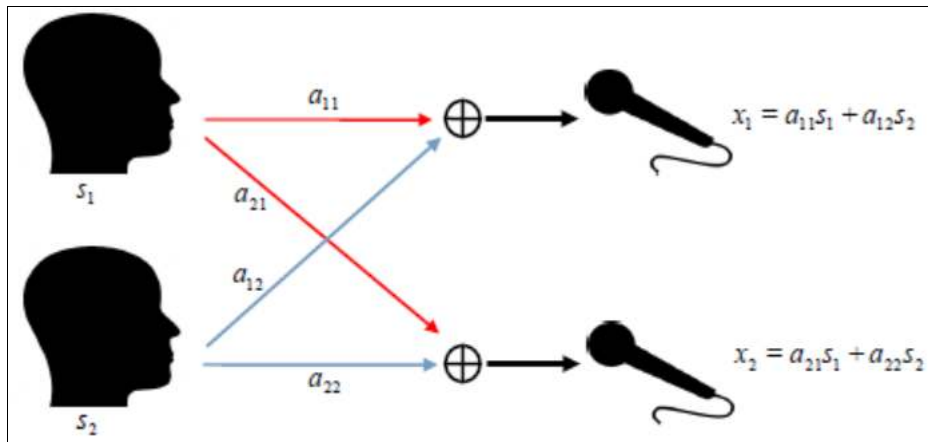
BBA algoritmaları için m tane gözlemin olduğu ve d tane kaynağın olduğu varsayılarak karışım modeli aşağıdaki gibi ifade edilir:

$$\begin{aligned} x_1 &= a_{11}s_1 + a_{12}s_2 \dots + a_{1d}s_d \\ x_2 &= a_{21}s_1 + a_{22}s_2 \dots + a_{2d}s_d \\ &\vdots \\ x_m &= a_{m1}s_1 + a_{m2}s_2 \dots + a_{md}s_d \end{aligned} \quad (17)$$

Üstteki denklem vektör matris halinde aşağıdaki gibi ifade edilir:

$$x = As \quad (18)$$

Burada A karışım matrisi katsayılarını temsil eder. BBA yöntemini en iyi açıklayan örnek kokteyl parti problemidir (KPP). KPP'de bir odanın içinde küçük gruplar halinde birden çok kişinin konuştuğu ve bu konuşmalar mikrofona ile kaydedildiği düşünülür. Buradaki amaç, kaynakları ve karışım matrisini bilmeden, kaydedilmiş konuşma sinyallerini kullanarak kaynak sinyallerini yeniden oluşturmaktır.



Şekil 10. Kokteyl parti problemi

Bu işlemin gerçekleştirilebilmesi için kaynakların istatistiksel olarak bağımsız oldukları varsayılır. Bağımsız bileşen vektörü ya da kaynak matrisinin elde edilebilmesi için aşağıdaki matematiksel ifade kullanılır.

$$A^{-1}x = A^{-1}As \rightarrow s = Wx \quad (19)$$

Üsteki ifadede görüleceği gibi önce A karışım matrisi kestirilir, daha sonra veri matrisi, A 'nın tersi ile çarpılarak, bağımsız bileşen matrisi bulunmaya çalışılır.

BBA algoritması uygulamadan önce veriler üzerinde bazı ön işlemler yapılması gerekir. Ön işlem olarak yapılan işlemler merkezileştirme ve beyazlaştırma işlemleridir.

BBA'da en temel ve gerekli ön işlem, ölçüm verisinin merkezileştirilmesidir. Merkezileştirme işlemi ile ölçüm verisinin ortalaması sıfır olacak şekilde dönüştürülür ve böylece bağımsız bileşenler sıfır ortamalı hale getirilir. Merkezileştirmenin matematiksel ifadesi,

$$x = x - E\{x\} \quad (20)$$

denklemleri ile tanımlanır.

BBA uygulamasında kullanılan diğer ön işlem beyazlaştırmadır. Beyazlaştırma işlemi, ölçüm vektörünün doğrusal bir dönüşüme tabi tutulması ve böylece yeni bir vektörün elde edilmesi işlemidir. Elde edilen bu yeni vektörün bileşenleri ilintisizdir ve standart sapmaları bire eşittir.

$$E\{xx^T\} = I \quad (21)$$

Beyazlaştırma işlemi için kullanılan en yaygın işlem kovaryans matrisinin öz değer ve öz vektörlerine ayrıştırma işlemidir.

$$C_x = E\{xx^T\} = U\Lambda U^T \quad (22)$$

burada U ortogonal öz vektör matrisi ve $A = \text{diag}(A_1, A_2, \dots, A_n)$ diyagonal öz değer matrisidir. Bu iki matris kullanılarak beyazlaştırma işlemi aşağıdaki denklem ile bulunabilir.

$$z = Qx = A^{-\frac{1}{2}} U^T x \quad (23)$$

Beyazlaştırma sonrası, karışım matrisi A daha iyi özelliklere sahip olan bir B matrisine dönüşür.

$$z = A^{-\frac{1}{2}} U^T x = A^{-\frac{1}{2}} U^T A s = B s \quad (24)$$

burada B ortogonal bir matristir.

$$E\{zz^T\} = BE\{ss^T\}B^T = BB^T = I \quad (25)$$

$$E\{ss^T\} = I \quad (26)$$

BBA'daki temel kısıtlama, birbirlerinden bağımsız olan kaynak işaretlerin Gauss dağılımına sahip olmamalarıdır. Kaynaklar bu şartı sağlamıyor ise BBA uygulanmaz.

BBA problemini çözmek için Merkezi limit teoremi kullanılır ve bu teoreme göre karışmış işaretlerin her biri kaynak işaretlerinden daha fazla Gauss dağılımına sahiptir. Rassal değişen bir değişkenin Gauss dağılımına sahip olup olmadığını anlamak için Kurtosis, Negentropi ve yaklaşık Negentropi gibi yöntemler kullanılabilir.

1.6.3.1. Kurtosis

Rassal değişen değişkenlerin aralarındaki Gauss dağılım yapısının incelenmesinde kullanılan klasik yöntemlerden olan kurtosis aynı zamanda dördüncü dereceden istatistik olarak da adlandırılır. Kurtosis, birinci, ikinci, üçüncü ve dördüncü momentlere bağlıdır ve sıfır ortalamaya sahip bir y değişkeni için aşağıdaki şekilde tanımlanmıştır [55];

$$\text{kurt}(y) = E\{y^4\} - 3\{E\{y^2\}\}^2 \quad (27)$$

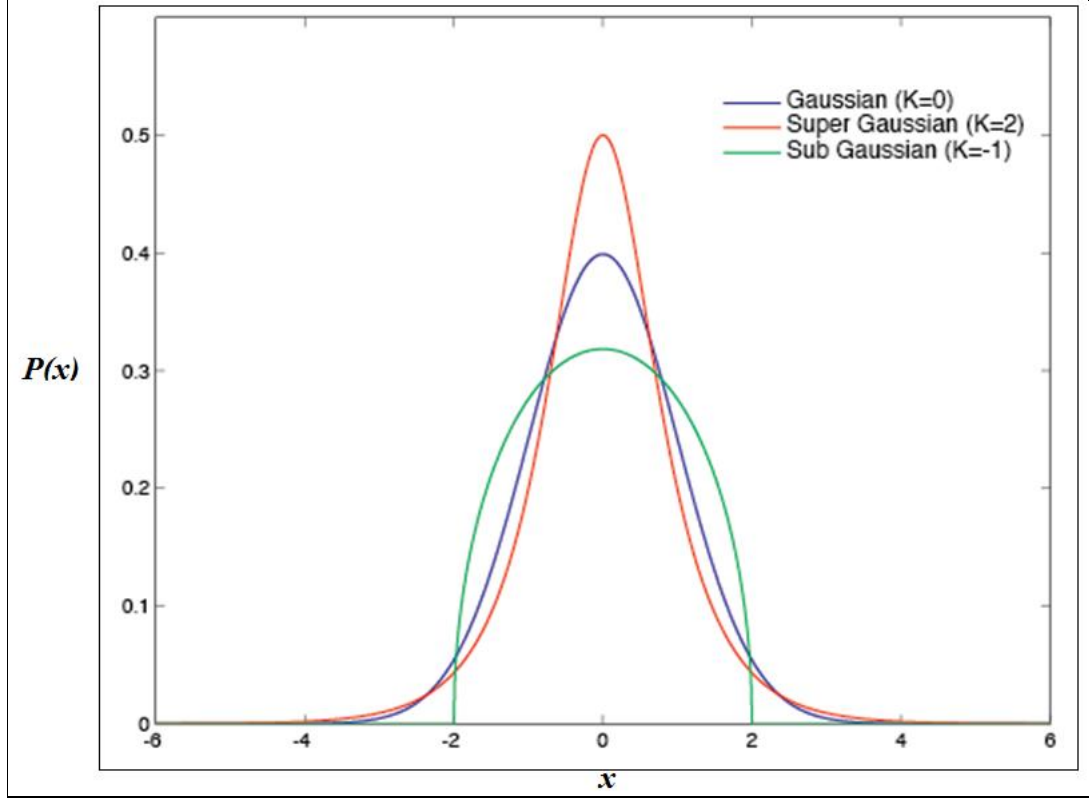
Normal dağılıma sahip bir değişken için Kurtosis sifıra eşittir. Pozitif kurtosis'e sahip olan bir değişken super Gaussian olasılık dağılımına sahipken, negatif kurtosis'e sahip olan değişkenler sub Gaussian olasılık dağılımına sahiptirler.

Eğer;

Kurtosis = 0 ise Gaussian

Kurtosis > 0 ise super Gaussian

Kurtosis < 0 ise sub Gaussian



Şekil 11. Gaussian, super Gaussian ve sub Gaussian için olasılık yoğunluk fonksiyonları

Kurtosis'in mutlak değeri ne kadar sifıra yakınsa, o değişkenin olasılık dağılımı o kadar Gauss dağılımına yakındır. Kurtosis'in en önemli problemi aşırı değerlere karşı olan hassasiyetidir. Az sayıda olan aşırı değerler Kurtosis'in büyümesine neden olabilir ve değişkenin olasılık dağılımı hakkında yanlış yorum yapmaya neden olabilir.

1.6.3.2. Negentropi

Gauss dağılıma sahip verinin önemli bir özelliği, aynı varyansa sahip veriler arasında entropisi en büyük olan veriler olmalarıdır. Diğer her bir olasılık dağılımı Gauss dağılımından düşük entropiye sahiptir. Bu özellik, entropinin Gauss olmayan dağılım ölçümü için negentropi ifadesi olarak ölçümde kullanabileceğini göstermektedir [54]. Ayrık rassal değişkenler için entropi ifadesi aşağıdaki şekildedir.

$$H(y) = -\sum_i P(y = a_i) \log P(y = a_i) \quad (28)$$

Negentropi, sürekli rassal değişkenler için diferansiyel entropi olarak adlandırılır ve aşağıdaki denklem ile ifade edilebilir.

$$H(y) = -\int f(y) \log f(y) dy \quad (29)$$

Negentropi, diferansiyel entropinin Gauss dağılıma sahip değişkenin Entropi'sinden çıkarılmasından elde edilir ve aşağıdaki denklem ile ifade edilebilir.

$$J(y) = H(y_{gauss}) - H(y) \quad (30)$$

Bu ifadeye göre, negentropi negatif değer alamıyor [55]. Negentropi Gauss olmayan ölçümler için kullanılabilmesine rağmen hesaplaması oldukça zordur. Bu hesaplama probleminin üstesinden gelmek için yaklaşık negentropi yöntemi kullanılır.

1.6.3.3. Yaklaşık Negentropi

Yüksek dereceli momentlere dayanan yaklaşık negentropy yöntemlerinden bir tanesi aşağıdaki gibi tanımlanmaktadır [55]:

$$J(y) \approx \frac{1}{2} E\{y^3\}^2 + \frac{1}{48} \text{kurt}(y)^2 \quad (31)$$

Üsteki denklemde Kurtosisin hesaplanmasında ortaya çıkan aşırı değer probleminden dolayı yeni bir yaklaşım geliştirilmiştir [55]:

$$J(y) \approx \sum_{i=1}^p K_i [E\{G_i(y)\} - E\{G_i(v)\}]^2 \quad (32)$$

burada K_i pozitif sabit bir sayı, v sıfır ortalama ve birim varyanslı standardize edilmiş değişken (Gauss), y sıfır ortalamalı ve birim varyanslı değişken, G_i ikinci dereceden olmayan bir fonksiyonu temsil etmektedir. İkinci dereceden olmayan bir G fonksiyonu kullanıldığında yaklaşım aşağıdaki şekilde yazılabilir:

$$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2 \quad (33)$$

burada önemli olan uygulama sırasında G fonksiyonunu mantıklı bir şekilde seçerek çok daha iyi sonuç veren bir negentropi yaklaşımının elde edilmesidir. Farklı dağılımlar için G 'nin aşağıdaki ifadeleri kullanılabilir.

$$\begin{aligned} G_1(y) &= \frac{1}{a_1} \log \cosh(a_1 y) \\ G_2(y) &= -\exp\left(-\frac{y^2}{2}\right) \end{aligned} \quad (34)$$

Sonraki aşama ise, W karışım matrisinin tahmin edilmesi aşamasıdır. Kullanılan her BBA algoritmasında karışım matrisinin tahmini için farklı yöntemler kullanılmaktadır. Hızlı BBA algoritması bu yöntemlerden biridir.

1.6.3.4. Hızlı BBA Algoritması

Hızlı BBA algoritmasında bağımsız kaynakları elde etmek için uygulanan aşamaları aşağıda verilmektedir [55]:

ADIM 1: Veri sıfır ortalamalı olacak şekilde merkezileştirilir.

ADIM 2: Veriyi ilintisiz yapmak için beyazlaştırılır.

ADIM 3: Bağımsız bileşen sayısını m belirlenir, $p \leftarrow 1$

ADIM 4: W_p vektörü için rastlantısal sayılar kullanarak başlangıç değeri seçilir.

ADIM5: W vektörü aşağıdaki formül ile güncellenir.

$$W_p \leftarrow E\{zg(W_p^T z)\} - E\{g'(W_p^T Z)\}W_p$$

Bu formüldeki g fonksiyonu aşağıdaki fonksiyonlardan biri olarak seçilir;

$$g(y) = \tanh(a_1 y), \quad g(y) = y \cdot \exp\left(\frac{-y^2}{2}\right)$$

ADIM 6: Ortogonalizasyon işlemi gerçekleştirilir.

$$W_{p+1} = W_{p+1} - \sum_{j=1}^p W_{p+1}^T W_j W_j$$

ADIM 7: W_p vektörünü normalleştirilir.

$$W_{p+1} = \frac{W_{p+1}}{\sqrt{W_{p+1}^T W_{p+1}}}$$

ADIM 8: Eğer istenen yakınsama derecesine ulaşılmadıysa ADIM 4'e geri dönülür.

ADIM 9: $p \leftarrow p + 1$, ve $p \leq m$ ise, ADIM 2'ye geri dönülür.

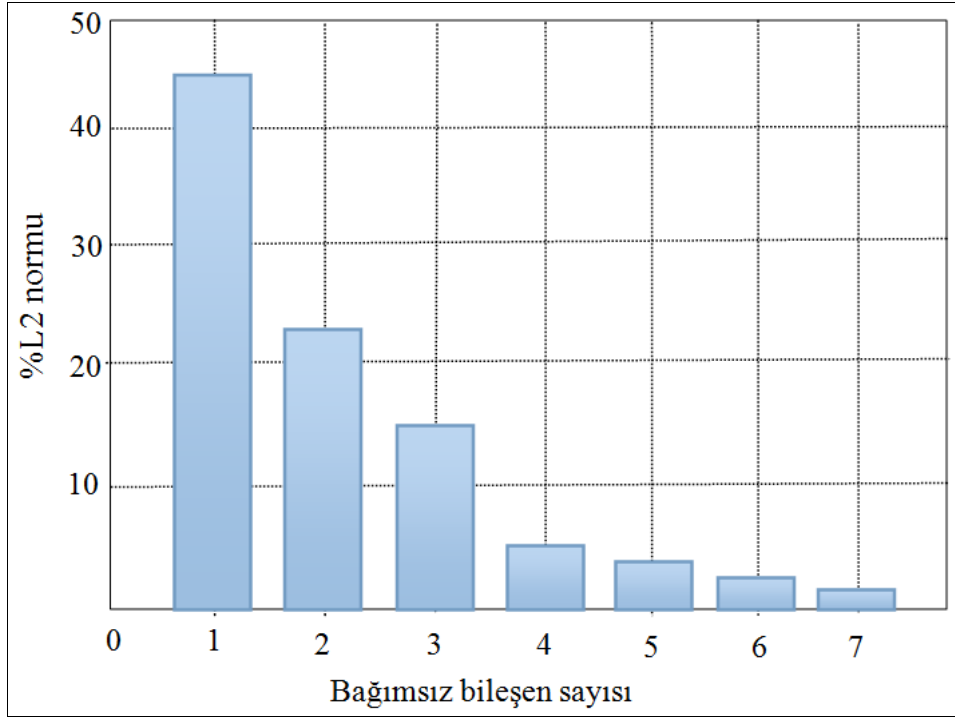
ADIM 10: $|W_{old} - W| < 10^{-5}$ ise işlem sonlandırılır.

1.6.4. BBA ile Hata Tespiti

BBA'da bağımsız bileşenlerin sıralanmasına TBA'daki gibi değişkenlerin standart sapma değerlerine bakılarak karar verilmez. Bağımsız bileşenlerin sayısına karar vermek için standart bir ölçüt yoktur, bunun için birkaç farklı yöntem önerilmiştir. Örneğin, Hyvarinen çalışmasına göre bileşenler Gauss dağılıma yakınlıklarına göre sıralanır [56]. Back ve Weigend'in çalışmasına göre ise bileşenler tek tek normlarına göre L_∞ sıralanır [57]. Cardoso ve Souloumici ise karışım matrisinin satırlarının sıralanması için bir öklid normu kullanmışlardır [58].

Basit bir yöntem olduğundan ve iyi sonuçlar verdiği için bu çalışmada karışım matrisinin satırlarının sıralaması için L_2 öklid normu kullanılmıştır. Böylece, bağımsız bileşenlerin sırasına her bir karışım matrisinin normu ile karar verilir ve karışım matrisinin satır sayısı: $\arg_i \max \|W_i\|_2$ şeklinde yazılır.

Sonraki aşamada ise en uygun bağımsız bileşenleri seçmek gerekmektedir. Çok sayıda bileşenin seçilmesi, işin içine gürültünün girmesine ve performans kaybına neden olabilir. Bundan dolayı bağımsız bileşen sayısını belirlemek için TBA yönteminde kullanılan scree grafiğine benzer bir grafik tekniği kullanılmaktadır. Şekil 12’de gösterildiği gibi karışım matrisinin her satırının normunun yüzdesi bağımsız bileşenlerin sayısına karşılık gelmektedir. Grafiğe bakıldığında, son dört bileşenin normlarının değerleri daha küçüktür ve ilk üç bileşen ile geri kalan dört bileşen arasında bir kırılma görülmektedir. Bu nedenle ilk üç bağımsız bileşen seçilebilir.



Şekil 12. Yüzdelik L_2 norma karşılık bağımsız bileşen sayısı grafiği

Hata tespiti için daha önce de bahsedildiği gibi, değişken sayısı m ile bağımsız bileşenler sayısı d 'nin eşit olduğu varsayılarak karışım matrisi W ve bileşenlerin tahmini hızlı BBA algoritması ile bulunur. Ayrıca B, Q ve A matrisleri bir beyazlaştırma aşamasından elde edilir. Önceden de belirtildiği gibi karışım matrisinin birkaç satırı seçilerek veri boyutu azaltılabilir. Böylece, karışım matrisi iki parçaya; seçilen satırlar W_d (karışım matrisinin baskın parçası) ve diğer kalan satırlara W_e 'ye bölünebilir [59]. Seçilen indeksleri kullanarak, B matrisi aynı şekilde iki parçaya B_d ve B_e 'ye ayrıştırılır.

B_d aşağıdaki denklemi kullanarak direk hesaplanabilir:

$$B_d = (W_d Q^{-1})^T \quad (35)$$

B matrisinin geriye kalan sütunları B_e matrisini oluşturur. Her yeni gelen çok değişkenli gözlem x_{yeni} için, bağımsız veri vektörleri beyazlatma aşamasından sonra aşağıdaki şekilde yazılabilir:

$$\begin{aligned} \hat{S}_{yeni,d} &= W_d x_{yeni} \\ \hat{S}_{yeni,e} &= W_e x_{yeni} \end{aligned} \quad (36)$$

TBA'da kullanılan T^2 ve KTH istatistiği benzer şekilde BBA için de uygulanabilir [59]. BBA'da her bir örnek için, T^2 istatistiği I^2 istatistiği olarak bilinmektedir ve aşağıdaki gibi tanımlanmaktadır:

$$I^2 = \hat{S}_{yeni,d}^T \hat{S}_{yeni,d} \quad (37)$$

Ayrıca her bir örnek için KTH istatistiği, aşağıdaki gibi tanımlanmaktadır:

$$KTH = (x_{yeni} - \hat{x}_{yeni})^T (x_{yeni} - \hat{x}_{yeni}) \quad (38)$$

burada $\hat{x}_{yeni}$ aşağıdaki şekilde hesaplanır:

$$\hat{x}_{yeni} = Q^{-1} B_d W_d x_{yeni} \quad (39)$$

Simoglu ve ark. kanonik korelasyon analizi yöntemine dayanarak bir ikinci T^2 istatistiği önermiştir [60]. Bu istatistik I_e^2 olarak tanımlanır ve bu istatistiğin avantajı, bileşenler baskın kısımları yanlış sayıda seçilseler bile sonuçların hatalarının telafi edilebiliyor olabilmesidir. I_e^2 istatistiği aşağı şekilde hesaplanabilir:

$$I_e^2 = \hat{S}_{yeni,e}^T \hat{S}_{yeni,e} \quad (40)$$

BBA ile hata izlemede, bağımsız bileşenler bir süre boyunca çok değişkenli Gauss dağılıma uymayabilir. Dolayısıyla I^2 , I_e^2 ve KTH istatistiklerinin kontrol sınırları belli bir yaklaşık dağılımla belirlenmez. Bunun için Çekirdek Yoğunluk Tahmini (ÇYT) gibi alternatif bir yöntem kullanılması gerekmektedir [61].

1.6.4.1. Çekirdek Yoğunluk Tahmini

ÇYT, rastgele değişen süreçlerin olasılık yoğunluk fonksiyonunu (OYF) tahmin etmede kullanılan bir yaklaşımdır [62]. x noktasındaki $\hat{p}(x)$ olasılık yoğunluk fonksiyonu tahmini, çekirdek fonksiyonu $K(\cdot)$ üzerinden aşağıdaki gibi tanımlanmıştır:

$$\hat{p}(x) = \frac{1}{Mh} \sum_{k=1}^M K\left(\frac{x - x_k}{h}\right) \quad (41)$$

burada h bant genişliğidir ve ÇYT'de bant genişliği seçimi önemli bir konudur. Çünkü çok küçük bant genişliğinin seçimi yoğunluk tahmin edicisinin çok aşırı işlem yapmasına ve bant genişliğinin çok büyük olması da yoğunluk tahmin edicisinin çok az işlem yapmasına sebep olur. Bant genişliğini belirlemek için kullanılabilecek yöntem yoktur. Fakat, tümleştirilmiş hataların karelerinin ortalamasının yaklaşık değerinin minimum olabilmesi için, optimal bant genişliğinin kaba tahmini h_{opt} aşağıdaki denklemden türetilebilir.

$$h_{opt} = 1.06\sigma N^{\frac{-1}{5}} \quad (42)$$

burada σ standart sapmadır. I^2 , I_e^2 ve KTH istatistiklerinin kontrol sınırları ÇYT yöntemi kullanılarak elde edilir. Yoğunluk fonksiyonlarının %99 alanını kapsayan değerler bulunarak kontrol sınırları belirlenir.

BBA'da, $x(k)$ 'nin I^2 , I_e^2 ve KTH'ye yaptığı değişken katkıları aşağıdaki denklemler sırayla kullanılarak elde edilebilir.

$$C(I^2) = \frac{Q^{-1} B_d \hat{S}_{yeni,d}}{\|Q^{-1} B_d \hat{S}_{yeni,d}\|} \hat{S}_{yeni,d} \quad (43)$$

$$C(I_e^2) = \frac{Q^{-1} B_e \hat{S}_{yeni,e}}{\|Q^{-1} B_e \hat{S}_{yeni,e}\|} \hat{S}_{yeni,e} \quad (44)$$

$$C(KTH) = x_{yeni} - \hat{x}_{yeni} \quad (45)$$

1.7. Hata Teşhisi İçin Kullanılan Yöntemler

1.7.1. Hata Sınıflama Yöntem Bilimi

Hatanın varlığı tespit edildiğinde, operatörlere hatanın temel nedenini belirlemede ve daha fazla düzenleme yapmada yardımcı olması için hata tanılama aşaması devreye sokulur. Hata teşhisine yönelik teşhis yapacak farklı yöntemler bulunmaktadır. Bu tezde hata teşhisi yapan, deney verilerini sınıflandıran ve verilerin ait olduğu hata sınıfını belirleyen sınıflama çalışmaları yapılmıştır. Hata sınıfı belirlenirken, her bir hata verisi hatayı temsil eden bir sınıf olarak tanımlanır. Sınıflama yöntemine bağlı olarak bir veriye birden fazla sınıf atanabilir.

Hata teşhisinin adımları ve adımlarda kullanılan yöntemler kısaca tanımlanacak olursa;

1. Veri toplama işlemi gerçekleştirilir (her hata için 6 simülasyon) ve daha sonra istatistiksel özellikler kullanılarak özellikler çıkarılır.
2. TBA ve BBA'den elde edilen t_i skor değerleri ve değişkenlerin katkıları sınıflama algoritmaları için özellik olarak kullanılmaktadır.
3. Özellik çıkarma kısımda veri boyutu azaltılır ve veri daha kolay sınıflandırılacak hale getirilir.
4. Özellik seçimi Sarıcı yaklaşımı kullanılarak yapılır.
5. Sınıflandırma Destek Vektör Makineleri (DVM) kullanılarak yapılır. Burada ilk olarak eğitim seti ile eğitim yapılır ve test kısmında bulunan hatanın hangi sınıfa ait olduğuna bakılarak yöntemin başarısı hesaplanır.

1.7.2. Makine Öğrenimi İçin Özellik Seçimi

Makine öğreniminin başarısını etkileyen en önemli faktörlerden biri özellik verisidir. Teorik olarak, daha fazla özellik verisinin olması sınıflandırma algoritmalarına daha fazla ayırt etme gücü sağlamalıdır. Fakat makine öğrenim algoritmalarıyla ilgili pratik tecrübeler bu durumun her zaman iyi sonuç vermediğini göstermiştir. Birçok öğrenme algoritması özellik kümesindeki sınıf etiketinin olasılığında kestirim yapabildiği görülmektedir. Fakat veriler makine öğrenimi için uygunsa, öğrenilecek göreve ilişkin alakasız veya gereksiz olan veri özellikleri ortadan kaldırılarak öğrenme işlemi daha kolay ve kısa zamanda yapılabilir. Bu süreç özellik seçimi olarak adlandırılır ve bu sürecin özellik seçimi iyi tanımlanmış olup tamamen otomatik, sayısal olarak izlenebilir bir süreç olma potansiyeli vardır.

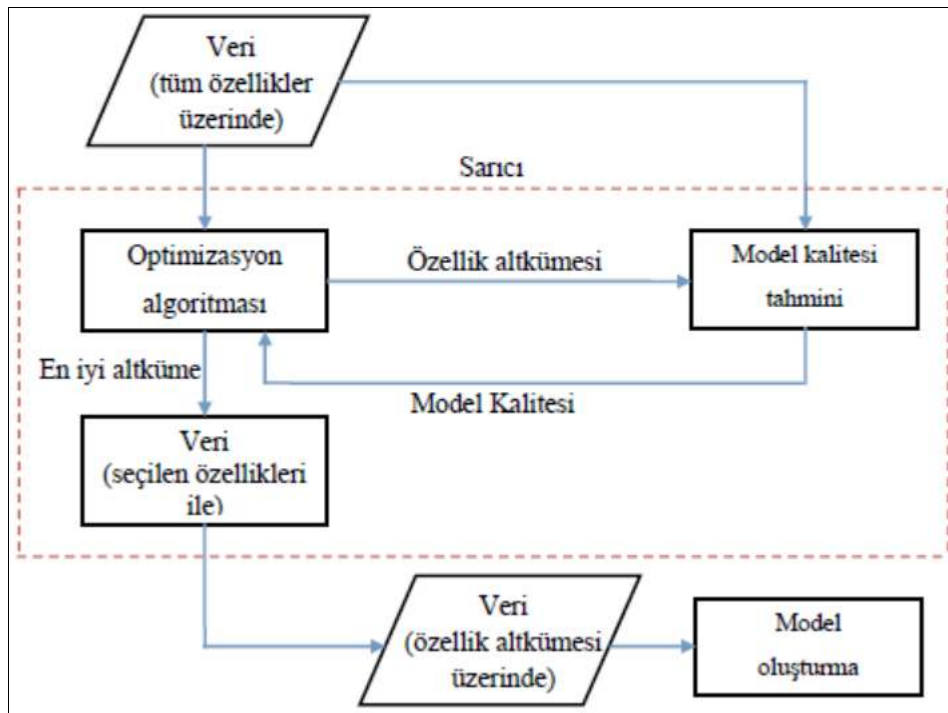
Öğrenme için özellik seçiminin faydaları olarak, öğrenmeyi gerçekleştirmek için gereken veri sayısında azalmaya neden olduğu, kestirim doğruluğunun artmasına neden olduğu, daha yoğun ve daha anlaşılabilir bilgi öğrenimi ve azalan uygulama süresine neden olduğu söz edilebilir.

Özellik seçimi mümkün olduğu kadar alakasız ve gereksiz bilgileri tespit etme ve ortadan kaldırma sürecidir. Bunu yapmak, verilerin boyutunu azaltır ve daha hızlı ve daha etkili işlem yapmak için algoritmaların öğrenmesine olanak sağlayabilir. Bazı durumlarda, gelecekteki sınıflamanın doğruluğu artırılabilir, bazı durumlarda ise sonuç hedef kavramının daha yoğun ve daha kolay yorumlanmış gösterimini sağlayabilir.

Makine öğrenimine yönelik mevcut özellik seçim yöntemleri iki kategoriye ayrılır. Bu kategoriler, en sonunda verilere uygulanacak olan öğrenme algoritmasını kullanarak özelliklerin sayısını belirleyenler ve verilerin genel özelliklerine dayalı olarak buluşsal yöntemler kullanarak özelliklerin sayısını belirleyen yöntemlerdir. İlk yöntem grubu sarıcılar, ikinci yöntem grubu ise filtreler olarak ifade edilir [63,64]. Her iki kategori içindeki yöntemler, özellik sayısı belirleme fonksiyonlarının yapısına ve özelliklerdeki alt kümelerin nasıl incelendiğine göre daha alt kategorilere ayrıştırılırlar. Sarıcılar, filtrelerle kıyasla (kestirim doğruluğu açısından) sıklıkla daha iyi sonuçlar vermektedir. Çünkü bu yöntemlerde kullanılan belli bir öğrenme algoritması için özellik seçimi en uygun hale getirilebilmektedir.

1.7.2.1. Sarıcı Yöntemleri

Sarıcı yöntem fikri, öğrenme için kullanılacak öğrenme algoritmasının aynı olan algoritmaya dayalı olarak özellik sayısı belirleme veya bir özellik alt kümesi seçme işlemidir [65] (Şekil 13). Bu yöntemlerde, entropi veya bir çeşit kara kutu fonksiyonu olan özellik sayısı belirleme yöntemleri kullanılmaktadır. Yani, her bir aday özellik alt kümesine yönelik özellik sayısı belirleme fonksiyonu, özellik alt kümesini kullanan öğrenme algoritmasının sevk ettiği modelin tahmin ettiği sayısını vermektedir [66].



Şekil 13. Sarıcı yaklaşımının genel yapısı

1.7.3. Destek Vektör Makineleri (DVM)

DVM'leri Vapnik'in ortaya koyduğu istatistiksel öğrenme teorisine dayalı yeni bir öğrenim yöntemidir [67]. DVM, ilk giriş alanı bir özellik alanı olarak adlandırılan yüksek boyutlu bir iç çarpım alanına eşlenir ve sınıflandırıcının genelleme yeterliğini en üst düzeye çıkarmak için özellik alanında ideal maksimal bir düzlem belirlenir. Maksimal düzlem, optimizasyon teorisi kullanılarak ve istatistiksel öğrenme teorisinin sunduğu kavramlara riayet edilerek bulunur. DVM, veri setinin doğrusal olarak ayrılıp ayrılamama

durumuna göre temel olarak ikiye ayrılmaktadır. Doğrusal ve doğrusal olmayan DVM olmak üzere iki iki gruba ayrılmaktadır.

1.7.3.1. Doğrusal Destek Vektör Makineleri

$x_i (i = 1, 2, \dots, M)$ verisinde M verinin örnek sayısını temsil etmektedir. Bu verideki örnekler, pozitif ve negatif sınıf olmak üzere iki sınıfa ait olduğu varsayılır ve pozitif sınıfa ait veri $y_i = 1$ etiketiyle, negatif sınıf ait veri $y_i = -1$ etiketiyle ilişkilendirilir. Verilerin doğrusal olduğu varsayılır ise verileri ayıran $f(x) = 0$ hiper-düzlemi aşağıdaki denklemdeki gibi tanımlanmaktadır.

$$f(x) = w^T x + b = \sum_{j=1}^M w_j x_j + b = 0 \quad (46)$$

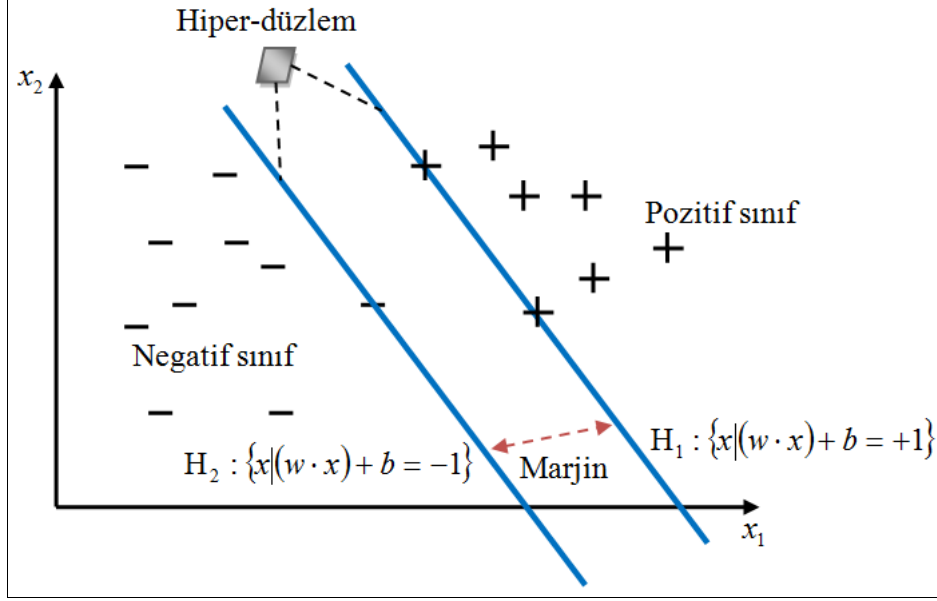
Pozitif veya negatif sınıftaki verileri sınıflandıran ayırıcı bir hiper-düzlem oluşturmak için $f(x)$ fonksiyonu kullanılır ve karar fonksiyonu oluşturulur. Ayrıca örnekler $y_i = 1$ ve $y_i = -1$ etiketli örneklere eşit uzaklıkta olan en uygun ayırıcı hiper-düzlemin bulunması için eğitim veri kümesinin aşağıdaki kısıtlamaları sağladığı varsayılırsa:

$$\begin{cases} f(x_i) = 1 & \text{if } y_i = 1 \\ f(x_i) = -1 & \text{if } y_i = -1 \end{cases} \quad (47)$$

veya tam denklemde aşağıdaki gibi gösterilebilir:

$$y_i f(x_i) = y_i (w^T x_i + b) \geq 1 \quad (48)$$

Denklem (48)'i sağlayan hiper-düzlemin iki tarafındaki en yakın örneklere olan dik uzaklıkların toplamı Marjin olarak adlandırılır. En uygun ayırıcı hiper-düzlem ya da ideal hiper-düzlem, marjin'i maksimum yapan hiper-düzlemdir ve Şekil 14'de gösterilmektedir.



Şekil 14. DVM ve hiper-düzlem

Verideki gürültü ξ_i ve sınıflandırmada yapılabilecek hata C göz önüne alınacak olur ise, hiper-düzlem aşağıdaki optimizasyon problemlerine bir çözüm olarak elde edilebilir:

$$\text{minimize } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^M \xi_i \quad (49)$$

$$\text{kısıtlama: } \begin{cases} y_i (w^T x_i + b) \geq 1 - \xi_i, \\ \xi_i \geq 0 \end{cases} \quad (50)$$

burada sınıflara ait destek vektörlerin arasındaki uzaklığı maksimize edecek optimal ayırma hiper-düzlemi optimizasyon probleminin çözülmesi sonucunda elde edilir. Hesaplamayı gerçekleştirmek için, Karush-KuhnTucker (KKT) koşulları kullanılarak temel problem, ikili problem haline dönüştürülerek basitleştirilebilir ve aşağıdaki denklem takımı oluşturulabilir.

$$\text{minimize } L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^M \alpha_i y_i (w \cdot x_i + b) + \sum_{i=1}^M \alpha_i \quad (51)$$

Minimizasyon işlemi w ve b 'ye göre denklem (51)'i minimum yapmaktır. Bu problem için KKT koşulları şunlardır;

$$\frac{\partial L}{\partial w} = 0, \quad \frac{\partial L}{\partial b} = 0, \quad (52)$$

$$w = \sum_{i=1}^M \alpha_i y_i x_i, \quad \sum_{i=1}^M \alpha_i y_i = 0 \quad (53)$$

Denklem (52)'den, x_i 'nin kapsadığı alt alanda w 'nın bulunduğu tespit edilebilir. Denklem (53)'un denklem (51)'de yerleştirilmesi ile optimizasyon problemi ikili biçime dönüşür ve aşağıdaki gibi elde edilir.

$$\text{maximize } L(\alpha) = \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=0}^M \alpha_i \alpha_j y_i y_j x_i x_j \quad (54)$$

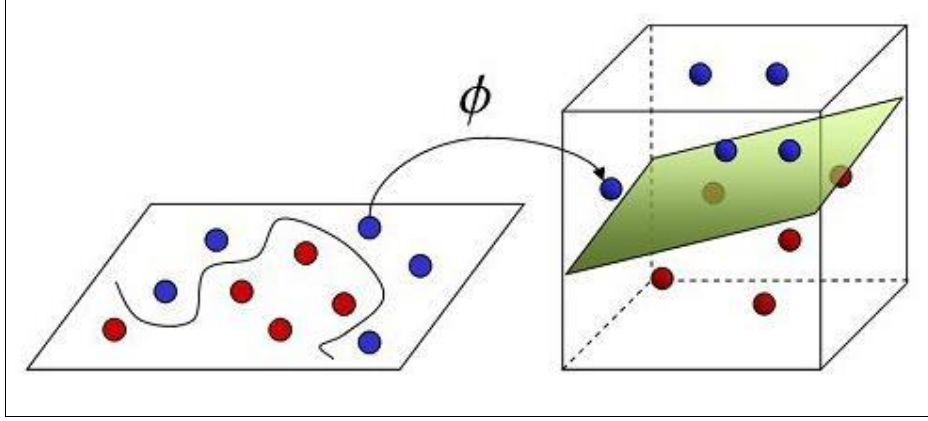
$$\begin{aligned} & \alpha_i \geq 0, \\ \text{kısıtlama: } & \sum_{i=1}^M \alpha_i y_i = 0 \end{aligned} \quad (55)$$

Bu ikili problemi çözmek ikinci dereceden optimizasyon problemi ile gerçekleştirilebilir. Denklem (54), aşağıdaki verilen karesel programlama ile çözülerek α_i değerleri bulunur, w ise denklem (49)'dan elde edilir.

$$f(x) = \text{sign} \left(\sum_{i,j=1}^M \alpha_i y_i (x_i x_j) + b \right) \quad (56)$$

1.7.3.2. Doğrusal Olmayan Destek Vektör Makineleri

DVM'lerin de kernel fonksiyonlarından yararlanılarak doğrusal olmayan sınıflamalar için de DVM'leri kullanılabilir. Bu yöntemde sınıflanacak veriler doğrusal sınıflamanın mümkün olduğu yüksek boyutlu bir özellik uzayına haritalandırılır (Şekil 15).



Şekil 15. Kernel fonksiyonu ile giriş uzayının doğrusal sınıflamanın çok boyutlu uzaya dönüşümü

n -boyutlu girdi vektörünü l -boyutlu özellik uzayına eşlemek için $\Phi(x) = (\phi_1(x), \dots, \phi_l(x))$ doğrusal olmayan fonksiyonları kullanılarak, ikili biçimde doğrusal karar fonksiyonu (57)'deki gibi elde edilir.

$$f(x) = \text{sign} \left(\sum_{i,j=1}^M \alpha_i y_j (\phi^T(x_i) \cdot \phi(x_j)) + b \right) \quad (57)$$

Yüksek boyutlu özellik alanında çalışma ile karmaşık fonksiyonların kolay şekilde ifade edilmesi sağlanır, fakat bu yöntem yüksek boyutlu vektörlerde başka bir problem de ortaya çıkarmaktadır. Kernel, veri noktalarının özellik alanı eşlemelerinin iç çarpımını veren ve $K(x_i, x_j) = (\phi^T(x_i) \cdot \phi(x_j))$ olarak belirtilen bir fonksiyondur. Kernel fonksiyonuna dayalı karar fonksiyonu aşağıdaki denklemde verilmektedir.

$$f(x) = \text{sign} \left(\sum_{i,j=1}^M \alpha_i y_j K(x_i, x_j) + b \right) \quad (58)$$

Uygun kernel fonksiyonunun seçilmesi çok önemlidir, çünkü kernel öğrenme kümesi örneklerin sınıflanacağı özellik alanını tanımlar. Uygun kernel fonksiyonu tanımı Mercer'in teoreminde sürekli ve pozitif tanımlı olarak yapılmıştır. Mercer'in teoremini yerine getiren herhangi bir fonksiyon özellik alanındaki bir iç çarpımı hesaplamak için kernel fonksiyonu olarak kullanılabilir [67]. DVM'lerde kullanılan polinom, Radyal

Tabanlı Fonksiyon (RTF), Pearson VII (PUK) fonksiyonu ve normalleştirilmiş polinom gibi farklı kernel fonksiyonları mevcuttur ve Tablo 1’de formülleri ve parametreleriyle birlikte verilmektedir.

Tablo 1. Kernel fonksiyonları

Kernel Fonksiyonu	Matematiksel İfadesi	Parametre
Polinom	$K(x, y) = ((x \cdot y) + 1)^d$	Polinom derecesi (d)
Normalleştirilmiş Polinom	$K(x, y) = \frac{((x \cdot y) + 1)^d}{\sqrt{((x \cdot x) + 1)^d ((y \cdot y) + 1)^d}}$	Polinom derecesi (d)
Radyal Tabanlı Fonksiyon	$K(x, y) = e^{-\gamma \ x - x_i\ ^2}$	Kernel boyutu (γ)
Pearson VII (PUK)	$\frac{1}{\left[1 + \left(\frac{2 \cdot \sqrt{\ x - y\ ^2} \sqrt{2 \left(\frac{1}{\omega} \right)} - 1}{\sigma} \right)^2 \right]^\omega}$	Pearson genişliği parametreleri (σ, ω)

1.7.3.3. Çoklu Sınıflı Destek Vektör Makineleri

DVM aslında ikili sınıflama için tasarlanmıştır. DVM’lerinin çoklu sınıflı sistemlerde nasıl etkili bir şekilde uygulanabileceği halen devam eden bir araştırma konusudur. Şu anda çoklu sınıflı sistemler için DVM’nin iki tür yaklaşımı mevcuttur. İlkinde birden fazla ikili DVM oluşturup birleştirirken diğerinde ise doğrudan bir optimizasyon formülündeki bütün verileri hesaba katarak sınıflandırma yapılmaktadır. Birinci türün iki alt yöntemi vardır: bire karşı kalanlar ve bire karşı bir.

1.7.3.3.1. Bire Karşı Kalanlar Yöntemi

DVM çoklu sınıflı sınıflandırma için kullanılan en eski uygulama bire karşı kalanlar yöntemidir [68]. k sayıda sınıf için k sayıda DVM oluşturulur. i . DVM modeli pozitif etiketlerin olduğu i . sınıf ve negatif etiketlerin olduğu diğer tüm sınıflardaki örneklerin hepsiyle denir. Dolayısıyla l test verisi $(x_1, y_1), \dots, (x_l, y_l)$, göz önüne alınarak

$x_i \in R^n, i = 1, \dots, l$ ve $y \in \{1, \dots, k\}$, x_i 'nin sınıfı olduğunda, i . DVM aşağıdaki problemi çözer:

$$\text{minimize } \frac{1}{2} (w^i)^T w^i + C \sum_{j=1}^l \xi_j^i \quad (59)$$

$$\begin{aligned} \text{kısıtlama: } & (w^i)^T \varphi(x_j) + b^i \geq 1 - \xi_j^i \quad \text{if } y_j = i \\ & (w^i)^T \varphi(x_j) + b^i \leq 1 - \xi_j^i \quad \text{if } y_j \neq i \end{aligned} \quad (60)$$

Denklem (59) çözüldükten sonra, k karar fonksiyonu meydana getirilir:

$$\begin{aligned} & (w^1)^T \varphi(x) + b^1 \\ & \dots \\ & (w^k)^T \varphi(x) + b^k \end{aligned} \quad (61)$$

x 'in karar fonksiyonu en büyük değerli hata sınıfında olduğu söylenebilir.

$$x \text{ 'in hata sınıfı} = \operatorname{argmax} \left((w^i)^T \varphi(x) + b^i \right) \quad (62)$$

1.7.3.3.2. Bire Karşı Bir Yöntemi

Bu yöntemde $\frac{k(k-1)}{2}$ sayıda ikili DVM oluşturulur ve bunların her biri iki hata sınıfına ait veriler ile denenir. i . ve j . hata sınıflarındaki veriler için aşağıdaki sınıflama problemi çözülerek sınıflandırma yapılır [69,70]:

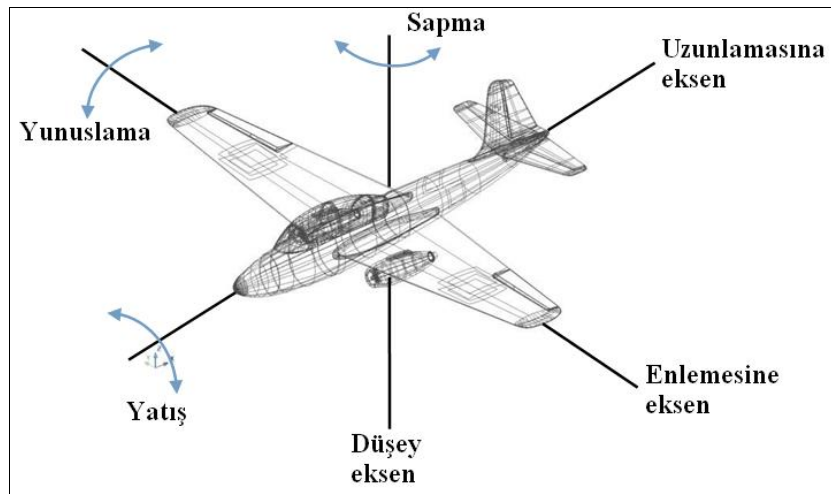
$$\text{minimize } \frac{1}{2} (w^{ij})^T w^{ij} + C \sum_{t=1}^l \xi_t^{ij} \quad (63)$$

$$\begin{aligned} \text{kısıtlama: } & (w^{ij})^T \varphi(x_t) + b^{ij} \geq 1 - \xi_t^{ij} \quad \text{if } y_t = i \\ & (w^{ij})^T \varphi(x_t) + b^{ij} \leq 1 - \xi_t^{ij} \quad \text{if } y_t \neq i \end{aligned} \quad (64)$$

$\frac{k(k-1)}{2}$ sayıda DVM oluşturulduktan sonra test yapmak için farklı yöntemler kullanılabilir. Friedman'ın ortaya koyulan oylama stratejisi yöntemi karar fonksiyonu için kullanılır: eğer $\text{sign}\left(\left(w^{ij}\right)^T \varphi(x)+b^{ij}\right) x$ 'in i . sınıfında olduğunu gösteriyorsa, bu durumda i . sınıf için verilen oya bir eklenir. Aksi takdirde, i . bir artırılır ve ardından x 'in en çok oyu aldığı sınıfta olduğu söylenir [71].

1.8. Uçak Sistemi ve Hareket Eksenleri

Bir uçağın etrafında hareket ettiği üç eksen vardır. Bunlardan; uzunlamasına eksen, uçağın ağırlık merkezinden geçen burnundan kuyruğuna uzanan eksendir. Uçağın boyuna eksen etrafında yaptığı harekete yatış hareketi denir. Uçağın bu eksen etrafında hareketleri uçak tipine bağlı olarak kanatçık, elevon veya spoiler ile kontrol edilir. Enlemesine eksen, uçağın ağırlık merkezinden geçerek bir kanat ucundan diğer kanat ucuna doğru uzanan eksendir. Uçağın bu eksen etrafında yaptığı harekete yunuslama hareketi denir. Uçağın bu eksen etrafında yaptığı hareket, irtifa dümeni (elevatör), hareketli yatay stabilize (stabilizör) ve elevonlar tarafından kontrol edilir. Düşey eksen, uçağın ağırlık merkezinden geçerek, gövde üst kısmından gövde alt kısmına uzanan eksendir. Uçağın düşey eksen etrafında yaptığı harekete sapma hareketi denir. Uçağın düşey eksen etrafındaki hareketi, istikamet dümeni tarafından sağlanır [5]. Şekil 16'da uçağın üç eksen üzerinde yaptığı hareketler gösterilmiştir.



Şekil 16. Uçağın hareket eksenleri

Bir uçağın durum değişkenleri ve kontrol girdileri denklem (65) ve (66)'da gösterilmektedir [9]:

$$x = [\alpha \quad \beta \quad V \quad p \quad q \quad r \quad \theta \quad \varphi \quad \psi \quad \gamma] \quad (65)$$

$$u = [\delta_e \quad \delta_r \quad \delta_a] \quad (66)$$

burada α hücum açısı, β yana kayma açısı, V hava hızı, p yatış hızı, q yunuslama hızı, r sapma hızı, θ yunuslama açısı, φ yatış açısı, ψ sapma açısı, γ uçuş yolu açısı, δ_e irtifa dümeni, δ_r kuyruk dümeni ve δ_a kanatçık olarak tanımlanmaktadır.

Ayrıca denklem (65) ve (66) yanlamasına hareket ve boylamasına hareket iki ayrılmış kümeler şeklinde Tablo 2'de gösterilmektedir.

Tablo 2. Uçağın durum değişkenleri ve kontrol girdileri

	Boylamasına model	Yanlamasına model
Durum değişkenleri	$x_{long} = [\alpha \quad V \quad q \quad \theta \quad \gamma]$	$x_{lat} = [\beta \quad p \quad r \quad \psi \quad \varphi]$
Kontrol girdileri	$u_{long} = [\delta_e]$	$u_{lat} = [\delta_r \quad \delta_a]$

- 1) Sistemde hata yok iken elde edilen hatasız veriler merkezleştirilir.
- 2) Hatasız veri için TBA modeli oluşturulur:
 - Kovaryans matrisi hesaplanır.

- Özdeğer ayrıştırma yöntemi ile kovaryans matrisin özdeğer ve özvektörleri bulunur.
 - Optimal Temel Bileşen sayısı (TBs) Pareto çizelgesi ile “ a ” hesaplanır.
 - Yükleme matrisi P ’nin temel bileşen sayısı “ a ” ya göre seçilir.
 - Skor matrisi T hesaplanır ve orijinal uzaya dönüştürülür.
 - Hatasız sistem için T^2 ve KTH istatistikleri için kontrol sınırı hesaplanır.
- 3) Hatalı veya hatası olarak gelen yeni veri testi:
- Yeni gelen hatalı veya hatasız veriler merkezleştirilir.
 - Yeni veriler oluşturulan TBA modeline yansıtılır.
 - Her bir yeni gözlem için T^2 ve KTH istatistikleri hesaplanır.
 - Sınırı geçen noktalar bir kontrol-dışı sinyal (hatalı) olarak değerlendirilmektedir.

Ayrıca hata tespiti için önerilen BBA algoritması aşağıdaki adımlarda ifade edildiği gibi uygulanmıştır.

- 1) Sistemde hata yok iken elde edilen hatasız veriler merkezleştirilir.
- 2) Merkezileştirilen veriler beyazlaştırılır:
 - B, Q ve A matrisleri bir TBA algoritması ile elde edilir.
 - Karışım matrisi W hızlı BBA algoritması ile elde edilir.
 - Optimal Bağımsız Bileşen sayısı (BBs) karışım matrisinin her satırının normu ile karar verilir.
 - Çekirdek yoğunluk tahmini yöntemi kullanılarak hatasız sistem için I^2 , I_e^2 ve KTH istatistiklerinin kontrol sınırları belirlenir.
- 3) Hatalı veya hatasız olarak gelen yeni veri testi:
 - Yeni gelen hatalı veya hatasız veriler merkezleştirilir.
 - Yeni veriler oluşturulan BBA modeline yansıtılır.
 - Her bir yeni gözlem için I^2 , I_e^2 ve KTH istatistikleri hesaplanır.
 - Sınırı geçen noktalar bir kontrol-dışı sinyal (hatalı) olarak değerlendirilmektedir.

2.2. Hata Tespiti İçin Elde Edilen Sonuçlar

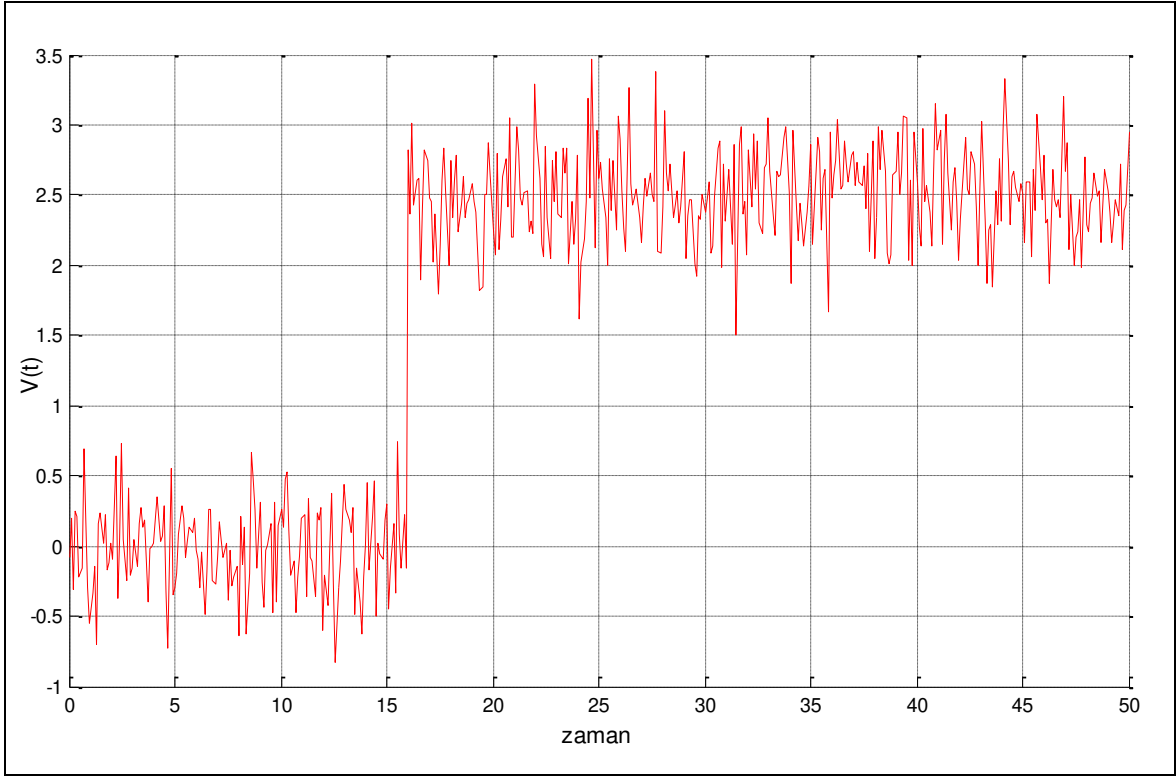
Bu tezde hata tespiti için üç hata türü sırasıyla yunuslama hız, yunuslama açısı ve hücum açısı jiroskoplarına uygulanmaktadır. Bu sensörlerde bir hata gerçekleştiğinde, geri bildirim döngüsünde mevcut olan hatalı durum diğer sensörleri de etkilemektedir. Ayrıca hatalara sıfır ortalama ve 0.1 standart sapmaya sahip olan bir Gauss gürültüsü eklenmiştir.

Bu sensörlerde farklı nedenlerden dolayı hata oluşabilir. Genel olarak bazı hata tipleri farklı sensörler içindir ve diğer hatalar yalnızca tek bir sensöre aittir. Bu tezde çok yaygın olan ve genellikle ısı aralıklarının ve sensör kalibrasyon sorunlarının sebebiyet verdiği toplamsal ve zaman karakteristiklerine göre meydana gelen hataların üç türü (ani, artan ve geçici hatalar) çalışılmıştır.

Analiz için veriler F_4 uçağının doğrusal olmayan bağımsız Simulink modeli kullanılarak elde edilmiştir. Simülasyon süresi 50 saniyedir ve 0.1 saniye örnekleme zamanına sahiptir.

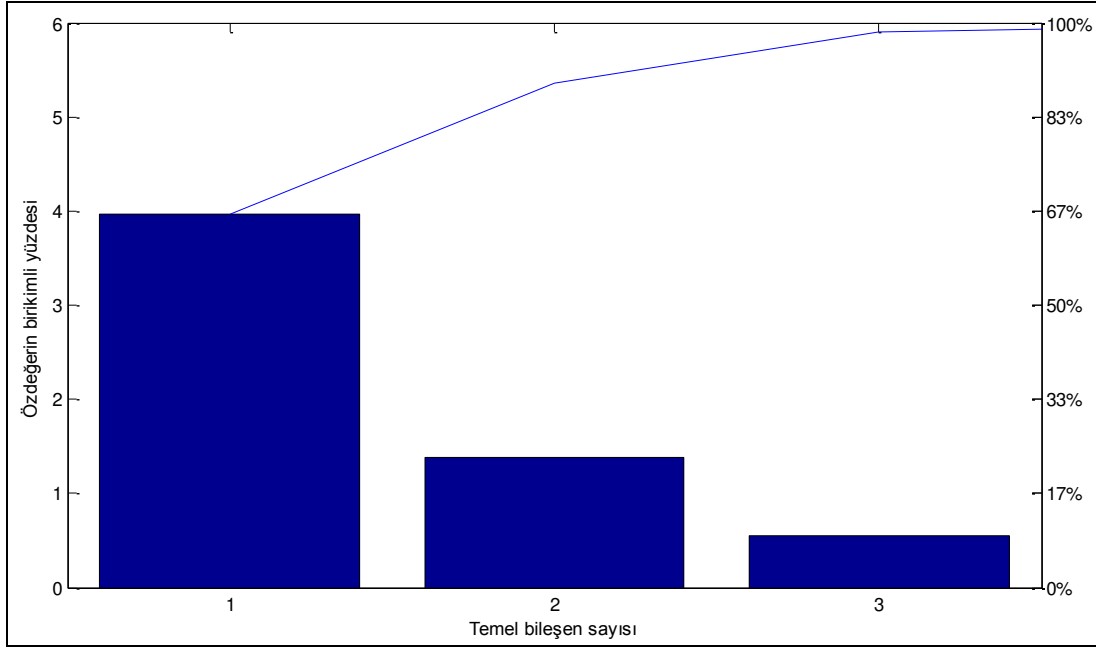
2.2.1. Ani Sensör Hata Tespiti İçin Elde Edilen Sonuçlar

Ani sensör hatası, 16. saniyeden uygulanmaya başlanmış ve basamak fonksiyonu ile modellenmiştir ve yunuslama hız sensörüne uygulanmıştır. Şekil 18'de görüleceği gibi, basamak fonksiyona ayrıca ortalaması sıfır, standart sapması 0.1 olan gauss gürültüsü de eklenmiştir.

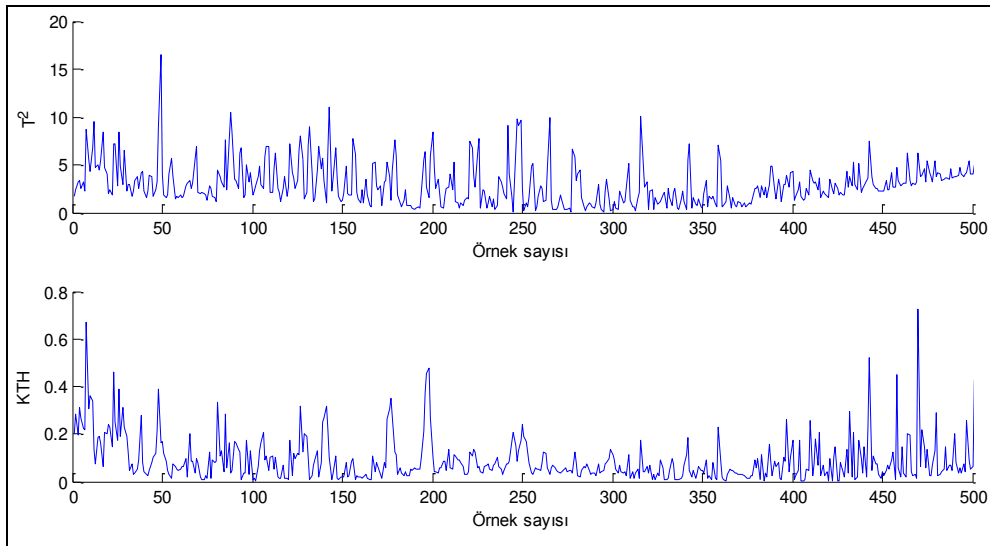


Şekil 18. Ani sensör hatası

İlk olarak, her biri 500 örnekten oluşan 6 değişkenli veri kümesi, hata sınırlarını belirlemek için elde edilir. Bu veriye karışmış veriyi ayrıştırmak için TBA algoritması uygulanır ve Pareto yöntemi uygulanarak temel bileşenlerin sayısı bulunur. TBA'nin Pareto çizelgesi Şekil 19'da verilmektedir. Bu grafikte %98 değişinti düzeyine sahip olan öz değerler temel bileşen olarak seçilir. Bu veri için temel bileşen sayısı üç olarak alınması uygun görülmektedir. Sistem davranışlarını incelemek için denklem (11) ve (13)'ü kullanılarak T^2 ve KTH hesaplanmaktadır. Hatasız çalışan sistemden hesaplanan T^2 ve KTH değerleri Şekil 20'de verilmiştir. Daha sonra denklem (12) ve (14)'ü kullanılarak %98'lik kontrol sınırı için $T_{\alpha} = 9.965$ ve $KTH_{\alpha} = 0.395$ değerleri hesaplanmıştır.



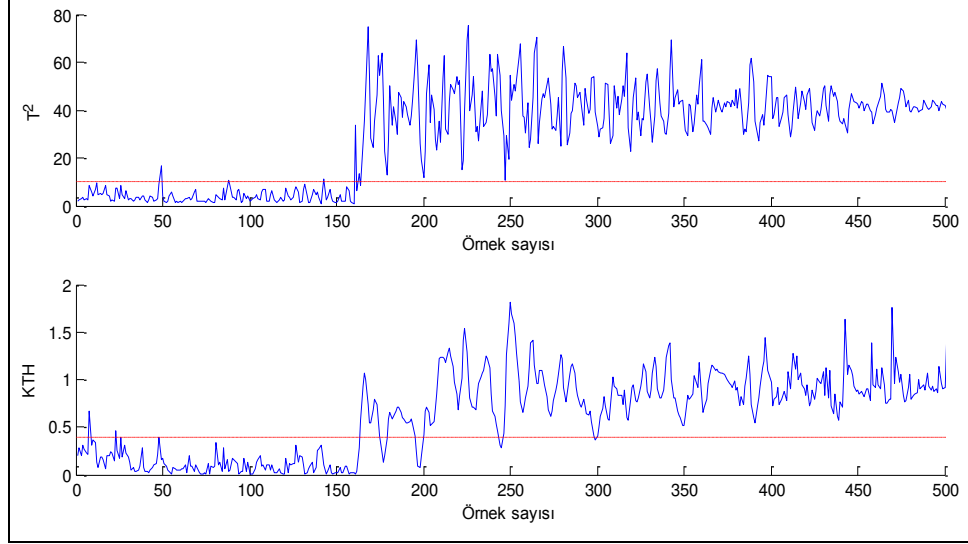
Şekil 19. Ani sensör hatası için Pareto çizelgesi



Şekil 20. Hatasız çalışan sistemden hesaplanan T^2 ve KTH istatistikleri

Hatasız çalışan sistemden hata sınırlarını hesapladıktan sonra, TBA algoritması hatalı veriyi ayırtmada ve ayırtılmış hatalı veri için T^2 ve KTH istatistik değerlerini hesaplamak için kullanılır. Hatalı çalışan sistemden elde edilen T^2 ve KTH istatistik değerleri Şekil 21'de verilmekte ve hata sınırları kırmızı noktalı ve tireli çizgi ile çizilmiştir. Bu hata sınırı üzerine çıkan değerler sistemde hata meydana geldiğini

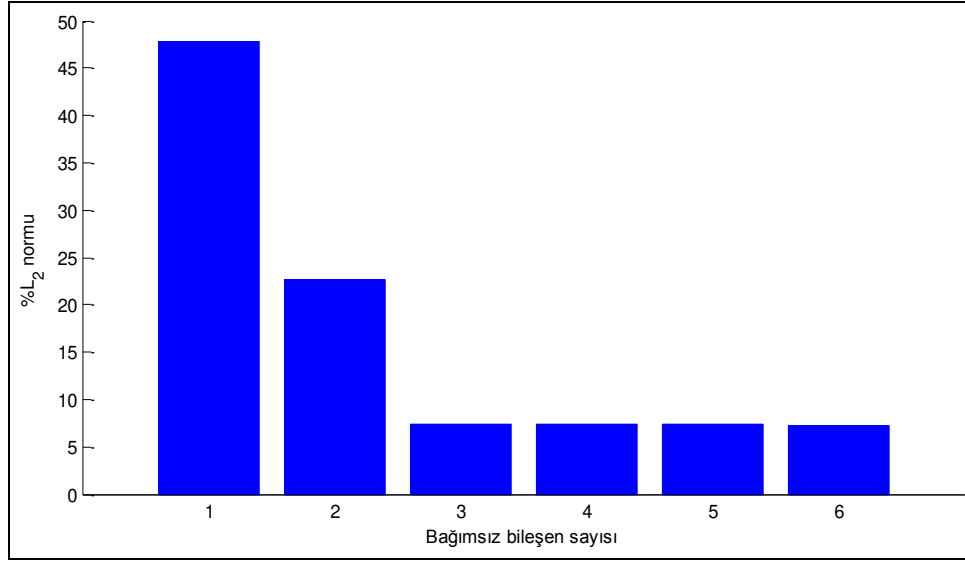
belirtmektedir. Bu grafik incelendiğinde, T^2 istatistiği yunuslama hız sensöründe meydana gelen hata sinyalini doğru bir şekilde tespit edebildiği görülmektedir.



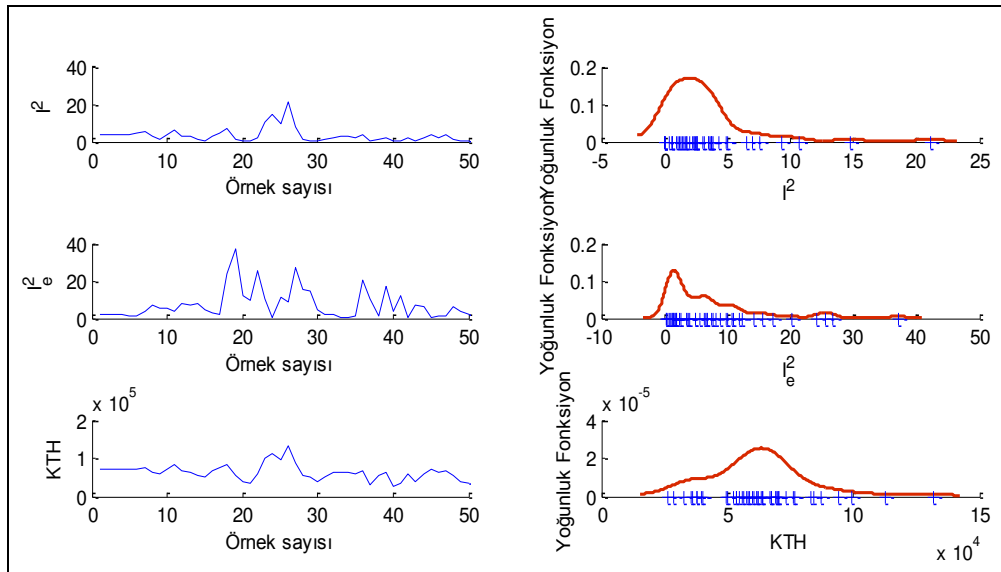
Şekil 21. Ani sensör hatası için hesaplanan T^2 ve KTH istatistikler

BBA algoritmasını uygulamak ve hata sınırlarını belirlemek için TBA algoritmasında olduğu gibi 500 örnekli 6 değişkenden oluşan veri kümesi kullanılmıştır. BBA algoritması ile ayrıştırılan veride boyut azaltımı için kullanılacak olan bağımsız bileşenlerin sayısı Şekil 22'de verilen grafiğe göre belirlenmiş olup bu veri kümesi için iki olarak seçilmiştir. Bu grafiğe bakıldığında, ayrıştırma matrisinin her satırının normunun yüzdesi bağımsız bileşen sayısına karşılık gelmektedir. Sistemde meydana gelen hataların tespiti için denklem (37), (38) ve (40)'i sırasıyla kullanarak I^2 , I_e^2 ve KTH değerleri hesaplanmıştır. Bunlara ek olarak ÇYT yöntemini kullanılarak %98'lik hata kontrol sınırı için $I_\alpha = 17.415$, $I_{e_\alpha} = 29.79$ ve $KTH_\alpha = 1.34 \times 10^5$ değerleri hesaplanmıştır.

Hatasız çalışan sistemden hesaplanan I^2 , I_e^2 ve KTH değerleri ve ÇYT değerleri Şekil 23'te verilmiştir.



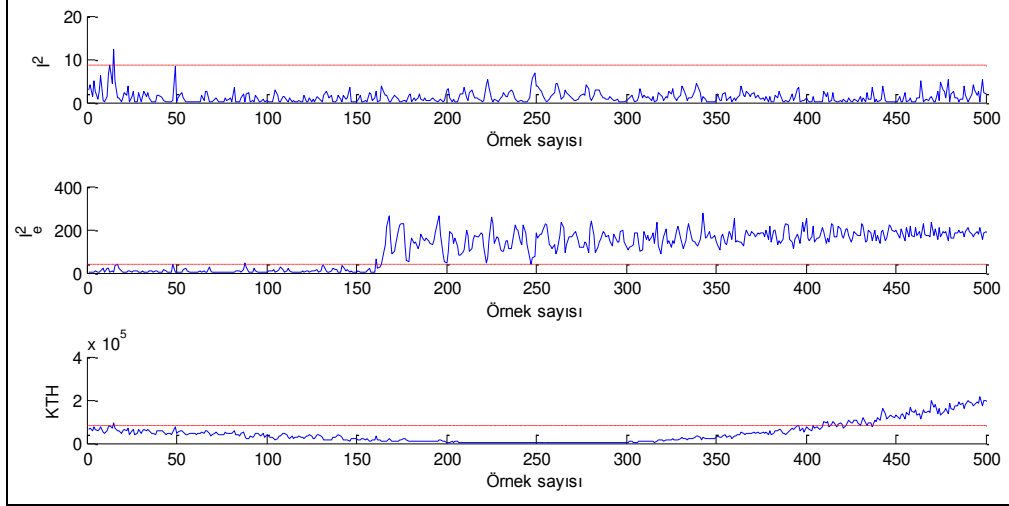
Şekil 22. Boyut azaltımı için kullanılacak olan bağımsız bileşenlerin sayısının grafiği



Şekil 23. Hatasız çalışan sistemden hesaplanan I^2 , I_e^2 ve KTH değerleri ve ÇYT değerleri

Hatasız çalışan sistemden hata sınırlarını hesapladıktan sonra, BBA algoritması kullanılarak işaretlerdeki karışımlar ayrıştırılır ve ayrıştırılan işaretlerde hatalı örneklerin olup olmadığı kontrol edilir. Bunun için yeni gelen ölçüm verilerine de BBA algoritması uygulanır ve ayrıştırılan verilerin I^2 , I_e^2 ve KTH değerleri hesaplanır. Hatalı çalışan sistemden elde edilen I^2 , I_e^2 ve KTH istatistikleri Şekil 24'te görülmektedir. Şekillerde

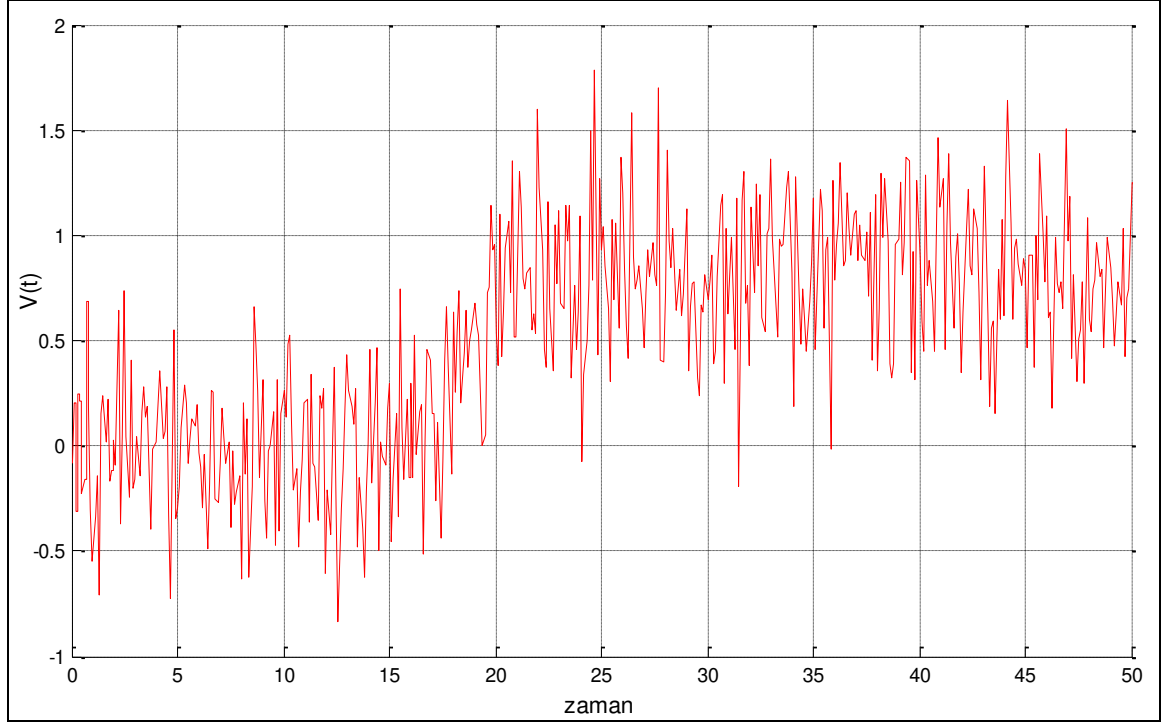
görülen kırmızı tire ve noktalı eşik değerleri her bir istatistik için ayrı ayrı hesaplanan hata sınırlarını göstermektedir. Bu hata sınırı üzerindeki değerler sistemde hata meydana geldiğini göstermektedir. I_e^2 istatistiğinin grafiği incelendiğinde, yunuslama hız sensöründe meydana gelen hata sinyalinin doğru bir şekilde tespit edilebildiği görülmektedir.



Şekil 24. Ani sensör hatası için hesaplanan I^2 , I_e^2 ve KTH istatistikleri

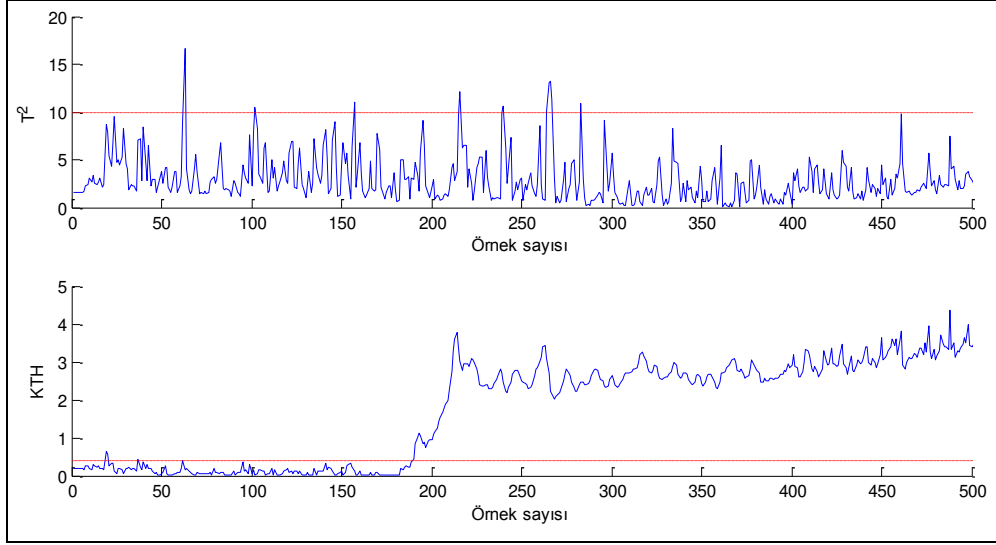
2.2.2. Artan Sensör Hata Tespiti İçin Elde Edilen Sonuçlar

Artan sensör hatası 16. saniyeden sonra rampa fonksiyonu ile modellenmiştir ve bu hata yunuslama açısı sensörüne uygulanmıştır. Şekil 25’de görüleceği gibi, rampa fonksiyona ayrıca ortalaması sıfır, standart sapması 0.1 olan gauss gürültüsü de eklenmiştir.

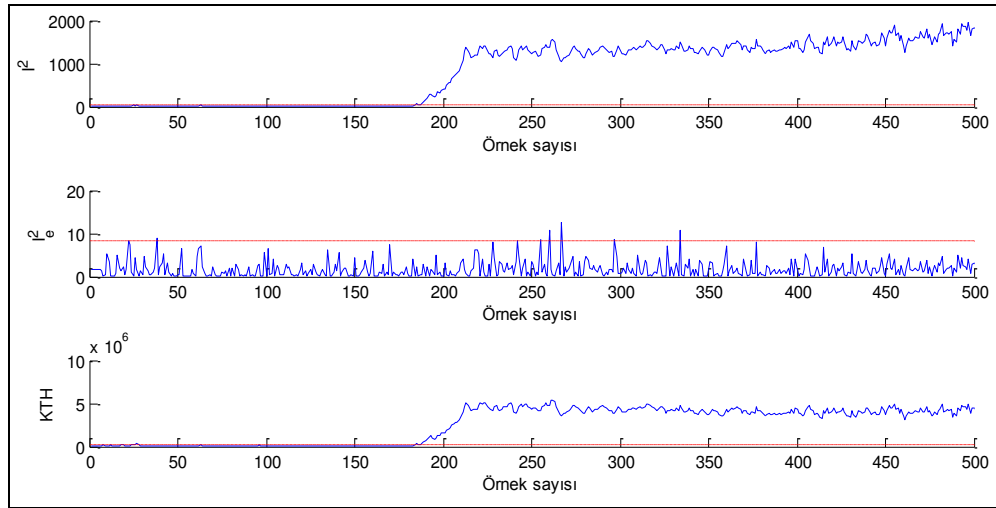


Şekil 25. Artan sensör hatası

Yunuslama hız sensöründe TBA ve BBA yöntemleri ile hata tespiti için yapılan işleminin aynısı yunuslama açısı sensörüne uygulanır. Hatasız çalışan sistem verisi ve TBA veya BBA algoritmaları kullanılarak işaretlerdeki karışımlar ayrıştırılır. Ayrıştırılan işaretler kullanarak hata sınırları hesapladıktan sonra hatalı örneklerin olup olmadığı kontrol edilir. Bunun için yeni gelen ölçüm verilerine de TBA ve BBA algoritmaları uygulanır ve daha sonra TBA'nın T^2 ve KTH değerleri ve BBA'nın I^2 , I_e^2 ve KTH değerleri hesaplanır. Hatalı çalışan sistemden elde edilen sonuçlar Şekil 26 ve 27'de görülmektedir. Şekillerde görülen kırmızı tire ve noktalı eşik değerleri her bir istatistik için ayrı ayrı hesaplanan hata sınırlarını göstermektedir. Bu hata sınırı üzerindeki değerler sistemde hata meydana geldiğini göstermektedir. Bu grafikler incelendiğinde, TBA'nın KTH istatistiği ve BBA'nın I^2 ve KTH istatistikleri yunuslama açısı sensöründe meydana gelen hata sinyalini doğru bir şekilde tespit edebildiği görülmektedir.



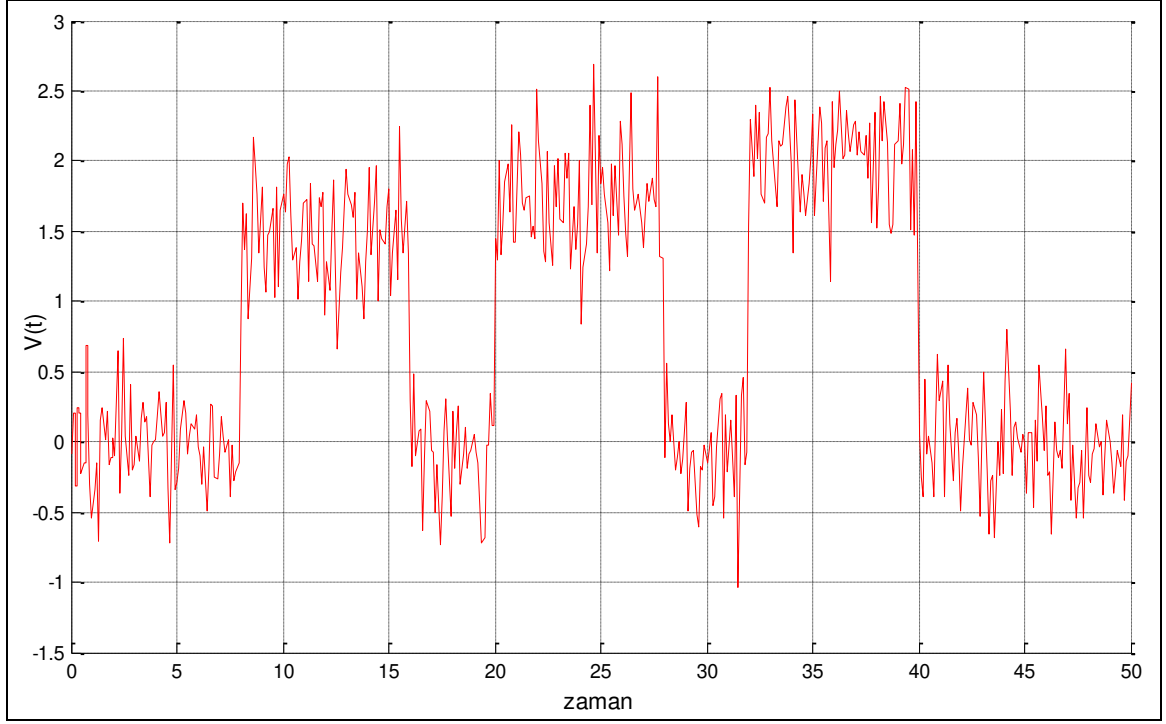
Şekil 26. Artan sensör hatası için TBA algoritmasının T^2 ve KTH istatistikleri



Şekil 27. Artan sensör hatası için BBA algoritmasının I^2 , I_e^2 ve KTH istatistikleri

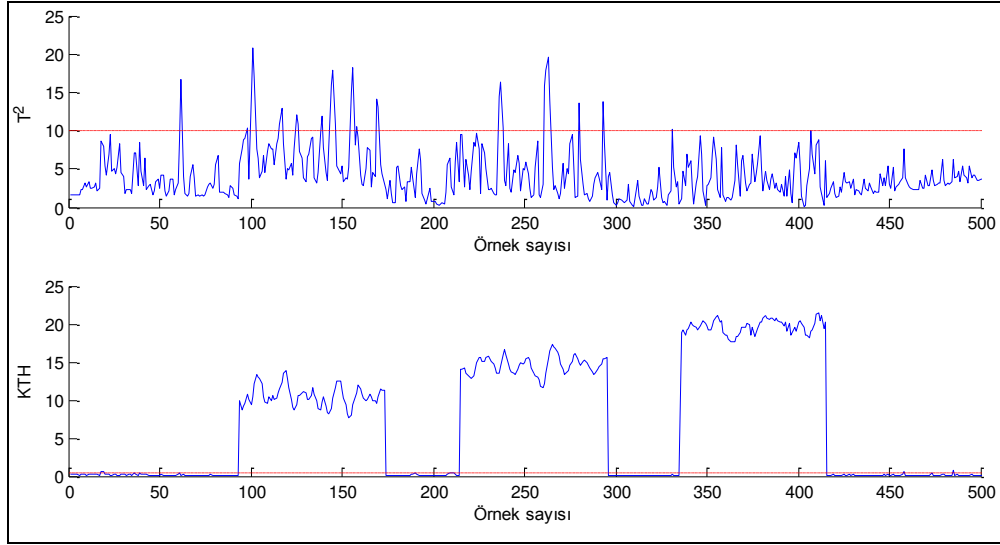
2.2.3. Geçici Sensör Hata Tespiti İçin Elde Edilen Sonuçlar

Geçici sensör hatası, farklı genliklerde impuls fonksiyonlarının birleşimi olarak modellenmiştir ve hücum açısı sensörüne uygulanmıştır. Geçici hata 8 saniye ile 16 saniyeler arasında genliği 1.5 olacak şekilde, 20 ile 28 saniyeler arasında genliği 1.75 olacak şekilde ve 32 ile 40 saniyeler arasında genliği 2 olacak şekilde uygulanmıştır. Şekil 28’de görüleceği gibi, impuls fonksiyonlarına ayrıca ortalaması sıfır, standart sapması 0.1 olan gauss gürültüsü de eklenmiştir.

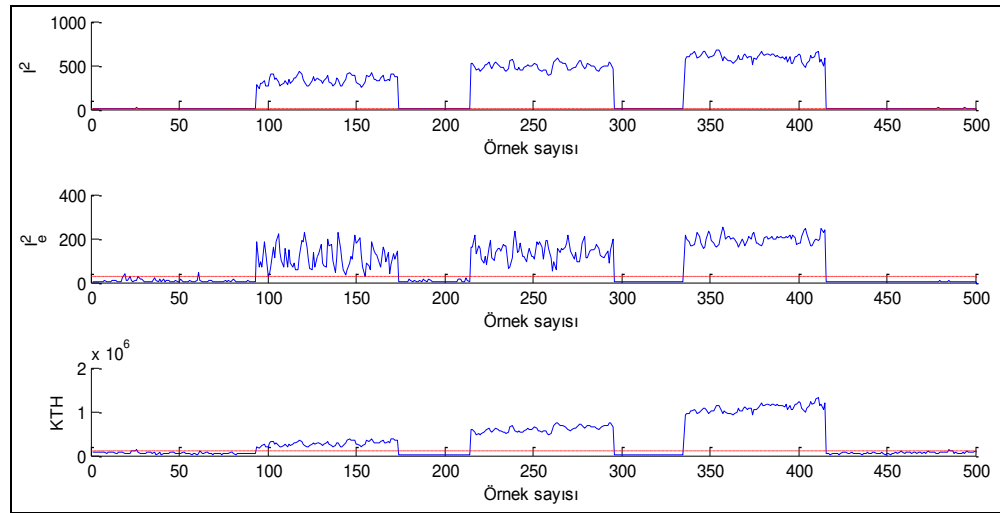


Şekil 28. Geçici sensör hatası

Hücum açısında hata tespiti için, yunuslama hız ve yunuslama açısı sensörlerinde hata tespitinde yapılan işlemlerin aynısı uygulanır. Hatasız çalışan sistem verisi ve TBA veya BBA algoritmaları kullanılarak işaretlerdeki karışımlar ayrıştırılır. Ayrıştırılan işaretler kullanarak hata sınırları hesapladıktan sonra hatalı örneklerin olup olmadığı kontrol edilir. Bunun için yeni gelen ölçüm verilerine de TBA ve BBA algoritmaları uygulanır ve daha sonra TBA'nın T^2 ve KTH değerleri ve BBA'nın I^2 , I_e^2 ve KTH değerleri hesaplanır. Hatalı çalışan sistemden elde edilen sonuçlar Şekil 29 ve 30'da görülmektedir. Şekillerde görülen kırmızı tire ve noktalı eşik değerleri her bir istatistik için ayrı ayrı hesaplanan hata sınırlarını göstermektedir. Bu hata sınırı üzerindeki değerler sistemde hata meydana geldiğini göstermektedir. Bu grafikler incelendiğinde, TBA'nın KTH istatistiği ve BBA'nın I^2 , I_e^2 ve KTH istatistikleri hücum açısı sensöründe meydana gelen hata sinyalini doğru bir şekilde tespit edebildiği görülmektedir.



Şekil 29. Geçici sensör hatası için TBA algoritmasının T^2 ve KTH istatistikleri



Şekil 30. Geçici sensör hatası için BBA algoritmasının I^2 , I_e^2 ve KTH istatistikleri

Hata tespiti işlemi daha önce belirtilen hata tipleri üzerinde TBA ve BBA algoritmaları ile gerçekleştirilmiştir. Yöntemlerin farklı hatalara karşı hata bulma başarımlarını karşılaştırmak için algoritmaların hata tespit oranları ve yanlış hata alarm oranları başarımların ölçütü olarak kullanılmıştır. Her yöntemden elde edilen hata tespiti başarımları; tespit oranları ve yanlış alarm oranları Tablo 3'te görülmektedir.

Tablo 3. Hata tespiti başarımı

	Tespit oranı					Yanlış alarm oranı				
	TBA		BBA			TBA		BBA		
Hata türü	T^2	KTH	I^2	I_e^2	KTH	T^2	KTH	I^2	I_e^2	KTH
Ani hata	99.41	95.88	0	99.41	24.70	1.87	1.25	0.62	1.25	0.62
Artan hata	2.05	91.47	93.23	17.64	92.64	1.87	1.87	0.62	1.25	0.62
Geçici hata	10	83.33	82.50	80.41	82.50	1.25	7.25	6.91	6.66	6.25

2.3. Hata Teşhisi İçin Elde Edilen Sonuçlar

Hatanın varlığı tespit edildiğinde, sistemin yöneticisine hatanın temel nedenini belirlemede ve daha fazla düzenleme yapmada yardımcı olması için hata tanıma aşaması devreye sokulur. Bu tezde hatanın sınıflandırma işlemini gerçekleştirmek için çoklu sınıflı DVM'nin iki tür yaklaşımı olan bire karşı kalanlar ve bire karşı bir uygulanmaktadır.

Bu işlem için toplamsal ve zaman karakteristiklerine göre meydana gelen hatalar; ani, artan (rampa genişliği = 12sn), artan (rampa genişliği = 20sn) ve geçici hatalar sırasıyla yunuslama hız, yunuslama açısı, hücum açısı ve uçuş yolu açısı jiroskoplarına uygulanmaktadır. Bu dört hata veri örtüşmesinin iyi bir temsilidir ve dolayısıyla sınıflandırması kolay değildir.

Genel bir sınıflandırma modeli oluşturmak için, bu hatalarla ilgili eğitim ve test verileri birbirlerinden bağımsız olarak F₄ uçağının doğrusal olmayan simulink modeli kullanılarak elde edilmiştir. Toplam simülasyon süresi 50sn'dir ve simülasyonun örnekleme zamanı 0.1sn olarak alınmıştır. Eğitim seti hazırlanırken her hata için 3 simülasyon gerçekleştirilmiştir ve toplamda 1200 örnek alınmıştır. Test verisi hazırlanırken her hata için tekrar 3 simülasyon gerçekleştirilmiştir ve toplam 600 örnek alınmıştır.

Sınıflama algoritmaları uygulamadan önce TBA ve BBA gibi sinyal ayrıştırma algoritmaları uygulanarak sinyaller ayrıştırılmış ve veri boyutu azaltılması için kullanılmıştır. Temel bileşenlerin sayısını belirlemek için pareto çizelgesi ve bağımsız bileşen sayısını belirlemek için TBA yönteminde kullanılan scree grafiğine benzer bir grafik tekniği kullanılmaktadır.

TBA ve BBA'dan elde edilen t_i skor değerleri ve değişkenlerin katkıları sınıflama algoritmaları için özellik olarak kullanılmaktadır. Sarıcı tekniğine dayalı özellik seçim yöntemi en uygun özellikleri seçmek için kullanılmıştır. En iyi özellikler, tüm özelliklerin (skor değerleri ve değişkenlerin katkıları) varlığında hesaplanmıştır.

Daha önce bahis edildiği gibi DVM algoritmaları için uygun kernel fonksiyonunun seçilmesi çok önemlidir. DVM algoritmalarının başarımını değerlendirmek için polinom ve RTF kernelleri daha sade ve anlaşılabilir oldukları için kernel fonksiyonları olarak kullanılmıştır. Bu kernel fonksiyonların parametreleri sonuçlar üzerinde önemli bir etkiye sahiptir. Bu tezde test kümesi için düzenleme parametresi olan C , polinom ve RTF kernelleri için 1 olarak belirlenmiştir. Ayrıca polinom kerneli için 1. dereceden bir polinom fonksiyonu kullanılmıştır. Diğer taraftan, RTF kerneli için, kernel boyutu γ değeri 0.01 olarak belirlenmiştir.

2.3.1 Sınıflandırma İşlemi İçin Elde Edilen Hata Matrisleri

Bire karşı kalanlar ve bire karşı bir yöntemleri kullanılarak gerçekleştirilen sınıflandırma işlemi için hesaplanan hata matrisleri Tablo 4'ten Tablo 11'e kadar verilmiştir. Bu yöntemlerden elde edilen sonuçlar dört kısımdan oluşmaktadır. Bu kısımlar sırasıyla sadece skor değerleri, tüm özellikler (skor değerleri ve değişkenlerin katkıları), her biri için özellik seçme yöntemi uygulanmadan ve özellik seçme yöntemi uygulandıktan sonraki sonuçları temsil etmektedir. Bu tablolara bakıldığında, hesaplanan hata matrislerin i satırının i . elemanı (i . köşegen eleman) sınıflandırıcı tarafından doğru etiketlenmiş olan ve operatör tarafından i sınıfına dâhil edilen hataların sayısını kapsamaktadır. Aynı satırın diğer sütunlarındaki elemanlar yanlış olarak sınıflandırılmış hataların sayısını göstermektedir.

Tablo 4. TBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani Hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	0	121	29	0	Artan hata 1	0	122	28	0
Artan hata 2	0	0	150	0	Artan hata 2	0	0	150	0
Geçici Hata	0	0	73	77	Geçici hata	0	0	73	77
Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani Hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	17	121	12	0	Artan hata 1	25	121	4	0
Artan hata 2	0	0	150	0	Artan hata 2	0	0	150	0
Geçici Hata	22	0	56	72	Geçici hata	30	0	50	70

Sınıflandırma işlemi için hesaplanan Tablo 4'deki hata matrislerine bakıldığında, sadece skor değerleri kullanıldığında, özellik seçme yöntemi uygulanmadan TBA'lı DVM'nin bire karşı bir yönteminde polinom kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 29 tanesi hatalı olarak sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten 77 tanesi doğru sınıflandırılırken 73 tanesi hatalı olarak sınıflandırılmıştır. Diğer yandan ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. TBA'lı DVM'nin bire karşı bir yönteminde RTF kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 28 tanesi yanlış sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten 77 tanesi doğru sınıflandırılırken 73 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Ayrıca TBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 29 tanesi yanlış sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten toplam 78 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata ve artan hata 2 sınıfları için belirlenen

hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. TBA'lı DVM'nın bire karşı kalanlar yönteminde RTF kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 29 tanesi, geçici hata sınıfı için belirlenen 150 hatalı örnekten 80 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir.

Tablo 5. TBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	1	20	8	121	Ani hata	145	0	1	4
Artan hata 1	60	29	44	17	Artan hata 1	133	2	8	7
Artan hata 2	9	0	141	0	Artan hata 2	12	0	138	0
Geçici hata	1	0	0	149	Geçici hata	35	0	0	115
Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	1	8	0	141	Ani hata	66	0	0	84
Artan hata 1	0	17	0	133	Artan hata 1	26	5	0	119
Artan hata 2	12	0	1	137	Artan hata 2	16	0	1	133
Geçici hata	0	3	0	147	Geçici hata	33	0	0	117

Sınıflandırma işlemi için hesaplanan Tablo 5'deki hata matrislerine bakıldığında, skor değerleri ve değişkenlerin katkıları kullanıldığında, özellik seçme yöntemi uygulanmadan TBA'lı DVM'nın bire karşı bir yönteminde polinom kernel kullanıldığında ani hata sınıfı için belirlenen 150 hatalı örnekten toplam 149 tanesi yanlış sınıflandırılmıştır. Artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 121 tanesi, artan hata 2 sınıfı için belirlenen 150 hatalı örnekten 9 tanesi ve geçici hata sınıfı için belirlenen 150 hatalı örnekten sadece 1 tanesi yanlış sınıflandırıldığı görülmektedir. TBA'lı DVM'nın bire karşı bir yönteminde RTF kernel kullanıldığında ani hata sınıfı için belirlenen 150 hatalı örnekten toplam 5 tanesi, artan hata 1 sınıfı için belirlenen 150 hatalı örnekten

toplam 148 tanesi, artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 12 tanesi ve geçici hata sınıfı için belirlenen 150 hatalı örnekten 35 tanesi yanlış olarak sınıflandırılmıştır. Ayrıca TBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında ani hata sınıfı için belirlenen 150 hatalı örnekten toplam 149 tanesi, artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 133 tanesi, artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 149 tanesi ve geçici hata sınıfı için belirlenen 150 hatalı örnekten sadece 3 tanesi yanlış sınıflandırıldığı görülmektedir. TBA'lı DVM'nin bire karşı kalanlar yönteminde RTF kernel kullanıldığında ani hata sınıfı için belirlenen 150 hatalı örnekten 84 tanesi, artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 145 tanesi, artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 149 tanesi ve geçici hata sınıfı için belirlenen 150 hatalı örnekten sadece 33 tanesi yanlış olarak sınıflandırılmıştır.

Tablo 4'deki sonuçlar ile Tablo 5'deki sonuçlar karşılaştırıldığında, tüm özellikler kullanıldığında sınıflandırıcının performansının azaldığı görülmektedir. Bunun için özellik seçimi yapılması gerektiği görülmektedir.

Tablo 6. TBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata1	0	121	29	0	Artan hata1	0	128	22	0
Artan hata2	0	0	150	0	Artan hata2	0	0	150	0
Geçici hata	0	0	67	83	Geçici hata	0	0	78	72
Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	17	121	12	0	Artan hata 1	18	128	4	0
Artan hata 2	0	0	150	0	Artan hata 2	0	0	150	0
Geçici hata	14	0	67	69	Geçici hata	31	0	52	67

Sınıflandırma işlemi için hesaplanan Tablo 6'daki hata matrislerine bakıldığında, sadece skor değerlerine özellik seçme yöntemi uygulandığında ise TBA'lı DVM'nin bire karşı bir yönteminde polinom kernel kullanıldığında ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 29 tanesi hatalı olarak sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten 67 tanesi hatalı olarak sınıflandırılmıştır. TBA'lı DVM'nin bire karşı bir yönteminde RTF kernel kullanıldığında ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 22 tanesi yanlış sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten 72 tanesi doğru sınıflandırılırken 78 tanesi yanlış sınıflandırılmıştır. Ayrıca TBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 29 tanesi yanlış sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten toplam 81 tanesi yanlış sınıflandırılmıştır. TBA'lı DVM'nin bire karşı kalanlar yönteminde RTF kernel kullanıldığında ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 22 tanesi, geçici hata sınıfı için belirlenen 150 hatalı örnekten 83 tanesi yanlış sınıflandırılmıştır.

Tablo 7. TBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	148	0	2	0	Ani hata	149	0	1	0
Artan hata 1	0	150	0	0	Artan hata 1	0	150	0	0
Artan hata 2	0	0	150	0	Artan hata 2	0	0	150	0
Geçici hata	0	9	12	129	Geçici hata	0	14	9	127

Tablo 7'nin devamı

Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	0	150	0	0	Artan hata 1	0	150	0	0
Artan hata 2	0	0	150	0	Artan hata 2	0	0	150	0
Geçici hata	56	0	1	93	Geçici hata	0	0	57	93

Sınıflandırma işlemi için hesaplanan Tablo 7'deki hata matrislerine bakıldığında, skor değerlerine ve değişkenlerin katkılarına özellik seçme yöntemi uygulandığında ise TBA'lı DVM'nin bire karşı bir yönteminde polinom kernel kullanıldığında artan hata 1 ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan ani hata sınıfı için belirlenen 150 hatalı örnekten toplam 148 tanesi, geçici hata sınıfı için belirlenen 150 hatalı örnekten 129 tanesi doğru sınıflara atandığı görülmektedir. TBA'lı DVM'nin bire karşı bir yönteminde RTF kernel kullanıldığında artan hata 1 ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan ani hata sınıfı için belirlenen 150 hatalı örnekten 149 tanesi doğru sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten 127 tanesi doğru sınıflandırılırken toplam 23 tanesi yanlış sınıflandırılmıştır. Ayrıca TBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında ani hata, artan hata 1 ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan geçici hata sınıfı için belirlenen 150 hatalı örnekten 93 tanesi doğru sınıflandırılırken 57 tanesi yanlış sınıflandırılmıştır. TBA'lı DVM'nin bire karşı kalanlar yönteminde RTF kernel kullanıldığında ani hata, artan hata 1 ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan geçici hata sınıfı için belirlenen 150 hatalı örnekten 83 tanesi yanlış sınıflandırılmıştır.

Üstteki sonuçlar ile karşılaştırıldığında, tüm özellikler kullanıldığında ve özellik seçimi yapılması sınıflandırıcının performansının arttığı görülmektedir.

Tablo 8. BBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	0	150	0	0	Artan hata 1	0	150	0	0
Artan hata 2	0	0	150	0	Artan hata 2	0	0	150	0
Geçici hata	0	0	1	149	Geçici hata	0	0	1	149
Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	20	122	8	0	Artan hata 1	65	85	0	0
Artan hata 2	0	0	150	0	Artan hata 2	149	0	1	0
Geçici hata	34	0	40	76	Geçici hata	93	0	0	57

Sınıflandırma işlemi için sadece skor değerleri kullanılarak hesaplanan Tablo 8'deki hata matrislerine bakıldığında, özellik seçme yöntemi uygulanmadan BBA'lı DVM'nin bire karşı bir yönteminde polinom kernel kullanıldığında ani hata, artan hata 1 ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan geçici hata sınıfı için belirlenen 150 hatalı örnekten sadece 1 tanesi hatalı olarak sınıflandırılmıştır. BBA'lı DVM'nin bire karşı bir yönteminde RTF kernel kullanıldığında polinom kernel ile aynı sonuçları vermektedir. Ayrıca BBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 28 tanesi yanlış sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten 76 tanesi doğru sınıflandırılırken 74 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. BBA'lı DVM'nin bire karşı kalanlar yönteminde RTF kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 65 tanesi, artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 149 tanesi ve geçici hata sınıfı için belirlenen 150 hatalı örnekten 93 tanesi yanlış

sınıflandırılmıştır. Diğer yandan ani hata sınıfı için belirlenen 150 hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Bu tablodan görüleceği gibi bire karşı bir yöntemi bire karşı kalanlar yöntemine göre oldukça iyi sonuçlar vermektedir.

Tablo 9. BBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	5	132	13	0	Artan hata 1	0	132	18	0
Artan hata 2	150	0	0	0	Artan hata 2	149	0	1	0
Geçici hata	143	0	7	0	Geçici hata	121	0	29	0
Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	0	0	150	0	Ani hata	27	0	123	0
Artan hata 1	0	127	23	0	Artan hata 1	0	126	24	0
Artan hata 2	0	0	150	0	Artan hata 2	146	0	4	0
Geçici hata	20	0	150	0	Geçici hata	108	0	42	0

Sınıflandırma işlemi için skor değerleri ve değişkenlerin katkıları kullanılarak hesaplanan Tablo 9'daki hata matrislerine bakıldığında, özellik seçme yöntemi uygulanmadan BBA'lı DVM'nin bire karşı bir yönteminde polinom kernel kullanıldığında ani hata sınıfı için belirlenen 150 hatalı örneklerinin tamamı doğru sınıfa atandığı görülmektedir. Diğer yandan artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 132 tanesi doğru sınıflandırılmıştır. Artan hata 2 ve geçici hata sınıfları için belirlenen 150 hatalı örnekten tamamı yanlış sınıflara atandığı görülmektedir. BBA'lı DVM'nin bire karşı bir yönteminde RTF kernel kullanıldığında ani hata sınıfı için belirlenen hatalı örneklerinin tamamı doğru sınıfa atandığı görülmektedir. Artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 18 tanesi ve artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 149 tanesi yanlış sınıflandırılmıştır. Diğer yandan geçici hata sınıfı için belirlenen

150 hatalı örnekten tamamı yanlış sınıfa atandığı görülmektedir. Ayrıca BBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında artan hata 2 sınıfı için belirlenen hatalı örneklerinin tamamı doğru sınıfa atandığı görülmektedir. Artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 127 tanesi doğru sınıflandırılırken 23 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata ve geçici hata sınıfları için belirlenen hatalı örneklerinin tamamı yanlış sınıflara atandığı görülmektedir. BBA'lı DVM'nin bire karşı kalanlar yönteminde RTF kernel kullanıldığında ani hata sınıfı için belirlenen 150 hatalı örnekten toplam 123 tanesi, artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 24 tanesi ve artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 146 tanesi yanlış sınıflandırılmıştır. Diğer yandan geçici hata sınıfı için belirlenen 150 hatalı örnekten tamamı yanlış sınıfa atandığı görülmektedir.

Tablo 8'deki sonuçlar ile Tablo 9'daki sonuçlar karşılaştırıldığında, tüm özellikler kullanıldığında sınıflandırıcının performansının azaldığı görülmektedir. Bunun için özellik seçimi yapılması gerektiği görünmektedir. Aşağıdaki Tablo 10 ve Tablo 11'de özellik seçim sonuçları görünmektedir.

Tablo 10. BBA için skor değerleri kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	0	143	7	0	Artan hata 1	0	150	0	0
Artan hata 2	0	0	150	0	Artan hata 2	0	0	150	0
Geçici hata	0	0	71	79	Geçici hata	0	7	15	128
Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	3	147	0	0	Artan hata 1	88	62	0	0
Artan hata 2	0	0	150	0	Artan hata 2	150	0	0	0
Geçici hata	72	0	0	78	Geçici hata	98	0	0	52

Sınıflandırma işlemi için skor değerlerine özellik seçme yöntemi uygulanarak hesaplanan Tablo 10'daki hata matrislerine bakıldığında, BBA'lı DVM'nin bire karşı bir yönteminde polinom kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 7 tanesi, geçici hata sınıfı için belirlenen 150 hatalı örnekten toplam 71 tanesi yanlış sınıflandırıldığı görülmektedir. Diğer yandan ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. BBA'lı DVM'nin bire karşı bir yönteminde RTF kernel kullanıldığında ani hata, artan hata 1 ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan geçici hata sınıfı için belirlenen 150 hatalı örnekten 128 tanesi doğru sınıflandırılırken 22 tanesi yanlış sınıflandırılmıştır. Ayrıca BBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında geçici hata sınıfı için belirlenen 150 hatalı örnekten toplam 72 tanesi ve artan hata 1 sınıfı için belirlenen 150 hatalı örnekten sadece 3 tanesi yanlış sınıflandırıldığı görülmektedir. Diğer yandan ani hata ve artan hata 2 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. BBA'lı DVM'nin bire karşı kalanlar yönteminde RTF kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 88 tanesi ve geçici hata sınıfı için belirlenen 150 hatalı örnekten toplam 98 tanesi yanlış olarak sınıflandırılmıştır. Diğer yandan ani hata sınıfı için belirlenen hatalı örneklerinin tamamı doğru sınıflara ve artan hata 2 sınıfı için belirlenen hatalı örneklerinin tamamı yanlış sınıflara atandığı görülmektedir. Tablo 8 ile Tablo 10 sonuçları karşılaştırıldığında, skor değerlerine özellik seçimi uygulanması sınıflandırma performansının düşmesine neden olmaktadır.

Tablo 11. BBA için skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan hata matrisi

Bire Karşı Bir Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	0	150	0	0	Artan hata 1	2	100	6	42
Artan hata 2	1	11	138	0	Artan hata 2	150	0	0	0
Geçici hata	91	4	2	53	Geçici hata	119	0	0	31

Tablo 11'in devamı

Bire Karşı Kalanlar Yöntemi									
Polinom Kernel					RTF Kernel				
Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata	Sınıf	Ani hata	Artan hata 1	Artan hata 2	Geçici hata
Ani hata	150	0	0	0	Ani hata	150	0	0	0
Artan hata 1	6	138	0	6	Artan hata 1	118	28	0	4
Artan hata 2	3	0	147	0	Artan hata 2	150	0	0	0
Geçici hata	65	0	73	12	Geçici hata	150	0	0	0

Sınıflandırma işlemi için skor değerlerine ve değişkenlerin katkılarına özellik seçme yöntemi uygulanarak hesaplanan Tablo 11'deki hata matrislerine bakıldığında, BBA'lı DVM'nin bire karşı bir yönteminde polinom kernel kullanıldığında ani hata ve artan hata 1 sınıfları için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. Diğer yandan artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 12 tanesi, geçici hata sınıfı için belirlenen 150 hatalı örnekten toplam 97 tanesi doğru sınıflara atandığı görülmektedir. BBA'lı DVM'nin bire karşı bir yönteminde RTF kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 50 tanesi yanlış sınıflandırılmıştır. Geçici hata sınıfı için belirlenen 150 hatalı örnekten 31 tanesi doğru sınıflandırılırken toplam 119 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata sınıfı için belirlenen hatalı örneklerinin tamamı doğru sınıflara ve artan hata 2 sınıfı için belirlenen hatalı örneklerinin tamamı yanlış sınıflara atandığı görülmektedir. Ayrıca BBA'lı DVM'nin bire karşı kalanlar yönteminde polinom kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten toplam 12 tanesi, artan hata 2 sınıfı için belirlenen 150 hatalı örnekten toplam 3 tanesi ve geçici hata sınıfı için belirlenen 150 hatalı örnekten toplam 138 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata sınıfı için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı görülmektedir. BBA'lı DVM'nin bire karşı kalanlar yönteminde RTF kernel kullanıldığında artan hata 1 sınıfı için belirlenen 150 hatalı örnekten 28 tanesi doğru sınıflandırılırken 122 tanesi yanlış sınıflandırılmıştır. Diğer yandan ani hata sınıfı için belirlenen hatalı örneklerinin tamamı doğru sınıflara atandığı ve artan hata 2 ve geçici hata sınıfları için belirlenen hatalı örneklerinin tamamı yanlış sınıflara atandığı görülmektedir.

Üstteki sonuçlar ile karşılaştırıldığında, tüm özellikler kullanıldığında ve özellik seçimi yapılması sınıflandırıcının performansının azaldığı görülmektedir.

2.3.2 Sınıflandırma İşlemi İçin Elde Edilen Sınıflandırma Başarım Oranı

Sınıflandırma algoritmalarının başarımlarının karşılaştırılması için genel sınıflandırma başarım oranları Tablo 12'den Tablo 15'e kadar görülebilmektedir. Genel sınıflandırma başarım oranları (yüzde olarak) her bir sınıf başarımların ortalamasıdır.

Tablo 12. Skor değerleri kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan sınıflandırma başarım oranı

Kernel	Çoklu sınıflı stratejileri	Sınıflama başarım oranı (%)	
		TBA	BBA
Polinom kerneli	Bire karşı bir	83	98.3
	Bire karşı kalanlar	82.2	83
RTF kerneli	Bire karşı bir	83.2	98.3
	Bire karşı kalanlar	81.8	48.8

Sınıflandırma işlemi için hesaplanan Tablo 12'deki genel sınıflandırma başarım oranlarına bakıldığında, sadece skor değerleri kullanıldığında, özellik seçme yöntemi uygulanmadan TBA'lı DVM'nin bire karşı bir yöntemi (polinom kerneli) için %83 doğruluğa, bire karşı bir yöntemi (RTF kerneli) için %83.2 doğruluğa, bire karşı kalanlar yöntemi (polinom kerneli) için %82.2 doğruluğa, bire karşı kalanlar yöntemi (RTF kerneli) için %81.8 doğruluğa sahip olduğu bulunmuştur. Diğer yandan BBA'lı DVM'nin bire karşı bir yöntemi (polinom kerneli) için %98.3 doğruluğa, bire karşı bir yöntemi (RTF kerneli) için %98.3 doğruluğa, bire karşı kalanlar yöntemi (polinom kerneli) için %83 doğruluğa, bire karşı kalanlar yöntemi (RTF kerneli) için %48.8 doğruluğa sahip olduğu bulunmuştur. Burada sınıflama yöntemine göre başarım analizi yapıldığında bire karşı bir yöntemi bire karşı kalanlar yöntemine göre daha iyi sınıflandırma başarımına sahip olduğu görülmektedir. Ayrıca BBA algoritması sonuçları, TBA algoritması sonuçlarına göre DVM kullanılarak başarım analizi yapıldığında, en yüksek sınıflandırma doğruluğu BBA algoritması sonuçlarından elde edildiği görülmektedir.

Tablo 13. Skor değerleri ve değişkenlerin katkılarını kullanılarak ve özellik seçme yöntemi uygulanmadan hesaplanan sınıflandırma başarımları

Kernel	Çoklu sınıflı stratejileri	Sınıflama başarımları (%)	
		TBA	BBA
Polinom kerneli	Bire karşı bir	53.3	47
	Bire karşı kalanlar	27.7	46.2
RTF kerneli	Bire karşı bir	66.7	47.2
	Bire karşı kalanlar	31.5	26.2

Sınıflandırma işlemi için hesaplanan Tablo 13'deki genel sınıflandırma başarımları oranlarına bakıldığında, skor değerleri ve değişkenlerin katkılarını kullanıldığında, özellik seçme yöntemi uygulanmadan TBA'lı DVM'nin bire karşı bir yöntemi (polinom kerneli) için %53.3 doğruluğa, bire karşı bir yöntemi (RTF kerneli) için %66.7 doğruluğa, bire karşı kalanlar yöntemi (polinom kerneli) için %27.7 doğruluğa, bire karşı kalanlar yöntemi (RTF kerneli) için %31.5 doğruluğa sahip olduğu bulunmuştur. Diğer yandan BBA'lı DVM'nin bire karşı bir yöntemi (polinom kerneli) için %47 doğruluğa, bire karşı bir yöntemi (RTF kerneli) için %47.2 doğruluğa, bire karşı kalanlar yöntemi (polinom kerneli) için %46.2 doğruluğa, bire karşı kalanlar yöntemi (RTF kerneli) için %26.2 doğruluğa sahip olduğu bulunmuştur. Burada sınıflama yöntemine göre başarımları analizi yapıldığında bire karşı bir yöntemi bire karşı kalanlar yöntemine göre daha iyi sınıflandırma başarımlarına sahip olduğu görülmektedir. Ayrıca BBA algoritması sonuçları, TBA algoritması sonuçlarına göre DVM kullanılarak başarımları analizi yapıldığında, en yüksek sınıflandırma doğruluğu TBA algoritması sonuçlarından elde edildiği görülmektedir.

Tablo 12'deki sonuçlar ile Tablo 13'deki sonuçlar karşılaştırıldığında değişkenlerin katkılarının özellik olarak seçilmesinin sınıflandırma başarımlarını düşürdüğü görülmektedir.

Tablo 14. Skor değerleri kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan sınıflandırma başarımları

Kernel	Çoklu sınıflı stratejileri	Sınıflama başarımları (%)	
		TBA	BBA
Polinom kerneli	Bire karşı bir	84	87
	Bire karşı kalanlar	81.7	87.5
RTF kerneli	Bire karşı bir	83.3	96.3
	Bire karşı kalanlar	82.5	44

Sınıflandırma işlemi için hesaplanan Tablo 14'deki genel sınıflandırma başarımlarına bakıldığında, sadece skor değerlerine özellik seçme yöntemi uygulandığında, TBA'lı DVM'nin bire karşı bir yöntemi (polinom kernelli) için %84 doğruluğa, bire karşı bir yöntemi (RTF kernelli) için %83.3 doğruluğa, bire karşı kalanlar yöntemi (polinom kernelli) için %81.7 doğruluğa, bire karşı kalanlar yöntemi (RTF kernelli) için %82.5 doğruluğa sahip olduğu bulunmuştur. Diğer yandan BBA'lı DVM'nin bire karşı bir yöntemi (polinom kernelli) için %87 doğruluğa, bire karşı bir yöntemi (RTF kernelli) için %96.3 doğruluğa, bire karşı kalanlar yöntemi (polinom kernelli) için %87.5 doğruluğa, bire karşı kalanlar yöntemi (RTF kernelli) için %44 doğruluğa sahip olduğu bulunmuştur. Burada sınıflama yöntemine göre başarımların analizi yapıldığında bire karşı bir yöntemi bire karşı kalanlar yöntemine göre daha iyi sınıflandırma başarımına sahip olduğu görülmektedir. Ayrıca BBA algoritması sonuçları, TBA algoritması sonuçlarına göre DVM kullanılarak başarımların analizi yapıldığında, en yüksek sınıflandırma doğruluğu BBA algoritması sonuçlarından elde edildiği görülmektedir.

Tablo 15. Skor değerleri ve değişkenlerin katkılarına kullanılarak ve özellik seçme yöntemi uygulandıktan sonraki hesaplanan sınıflandırma başarımları

Kernel	Çoklu sınıflı stratejileri	Sınıflama başarımları (%)	
		TBA	BBA
Polinom kerneli	Bire karşı bir	96.2	81.8
	Bire karşı kalanlar	90.5	74.5
RTF kerneli	Bire karşı bir	96	46.8
	Bire karşı kalanlar	90.5	29.7

Sınıflandırma işlemi için skor değerlerine ve değişkenlerin katkılarına özellik seçme yöntemi uygulanarak hesaplanan Tablo 15'deki genel sınıflandırma başarımlarına bakıldığında, TBA'lı DVM'nin bire karşı bir yöntemi (polinom kernelli) için %96.2 doğruluğa, bire karşı bir yöntemi (RTF kernelli) için %96 doğruluğa, bire karşı kalanlar yöntemi (polinom kernelli) için %90.5 doğruluğa, bire karşı kalanlar yöntemi (RTF kernelli) için %90.5 doğruluğa sahip olduğu bulunmuştur. Diğer yandan BBA'lı DVM'nin bire karşı bir yöntemi (polinom kernelli) için %81.8 doğruluğa, bire karşı bir yöntemi (RTF kernelli) için %46.8 doğruluğa, bire karşı kalanlar yöntemi (polinom kernelli) için %74.5 doğruluğa, bire karşı kalanlar yöntemi (RTF kernelli) için %29.7 doğruluğa sahip olduğu bulunmuştur. Burada sınıflama yöntemine göre başarımların analizi yapıldığında bire

karşı bir yöntemi bire karşı kalanlar yöntemine göre daha iyi sınıflandırma başarımına sahip olduğu görülmektedir. Ayrıca BBA algoritması sonuçları, TBA algoritması sonuçlarına göre DVM kullanılarak başarımların analizi yapıldığında, en yüksek sınıflandırma doğruluğu TBA algoritması sonuçlarından elde edildiği görülmektedir.

Üstteki sonuçlar ile karşılaştırıldığında, tüm özellikler kullanıldığında ve özellik seçimi yapılması TBA'lı DVM'nin sınıflandırma başarımını arttırdığı ve BBA'lı DVM'nin sınıflandırma başarımını düşürdüğü görülmektedir.

3. SONUÇLAR

Bu çalışmada, uçuş kontrol sistemlerinde toplamsal ve zaman karakteristiklerine göre meydana gelen sensör hatalarının tespiti ve teşhisi gerçekleştirilmiştir. Hatanın erken teşhisinin amacı, hata veya arızanın uygun zamanda, doğru ve duyarlı bir şekilde algılanması, yayılmasına ve istenmeyen zararlar oluşturmaya engel olmak ve sistemin çökmesini önlemektir. Bu amaçla bu çalışmada, sensörlerdeki hata tespiti uçağın yunuslama hız sensörü, yunuslama açısı sensörü ve hücum açısı sensörlerine hata sinyalleri eklenmiş ve daha sonra TBA ve BBA algoritmaları uygulanarak işaretler ayrıştırılmıştır. Ayrıştırılan işaretlerin farklı istatistiksel özellikleri kullanılarak özellik çıkarma işlemi yapılmıştır. Çıkarılan özellikler çoklu sınıflı DVM'nin iki tür yaklaşımı olan bire karşı kalanlar ve bire karşı bir algoritmaları vasıtasıyla sınıflandırma işlemi gerçekleştirilmiştir.

Hata tespiti sonuçları kullanılarak, yöntemleri ve yöntemlerin farklı hatalara karşı hata tespiti başarımları karşılaştırılmıştır.

Algoritmaların hata tespit oranlarına bakıldığında ani hata için TBA'nın T^2 istatistiği ve BBA'nın I_c^2 istatistiği yunuslama hız sensöründe meydana gelen hata sinyalini doğru bir şekilde tespit edebildiği ve hata tespit oranlarının yüksek olduğu görülmektedir (Tablo 3). Ayrıca yanlış alarm oranları karşılaştırıldığında I_c^2 istatistiği daha düşük yanlış alarm oranına sahip olduğu görülmektedir (Tablo 3). Zamanla artan hata değerleri için BBA'nın I^2 istatistiği yunuslama açısı sensöründe meydana gelen hata sinyalini doğru bir şekilde tespit edebildiği ve en yüksek hata tespit oranına sahip olduğu görülmektedir (Tablo 3). Yanlış alarm oranlarına baktığında ise I^2 istatistiğinin en düşük yanlış alarm verme oranına sahip olduğu görülmektedir (Tablo 3). Geçici hata için TBA'nın KTH istatistiği hücum açısı sensöründe meydana gelen hata sinyalini doğru bir şekilde tespit edebildiği ve en yüksek hata tespit oranına sahip olduğu görülmektedir. Ayrıca yanlış alarm oranları açısından baktığımızda bu KTH istatistiği en yüksek başarımla orana sahip olduğunu ve T^2 istatistiği en düşük başarımla orana sahip olduğunu görülmektedir (Tablo 3).

Sınıflandırma işlemi için hesaplanan hata matrisleri ve genel sınıflandırma başarımlarına bakıldığında sadece skor değerleri kullanıldığında, özellik seçme yöntemi uygulanmadan BBA'lı DVM'nin bire karşı bir yöntemi (polinom kernelli) en yüksek sınıflandırma doğruluğa sahip olduğu görülmektedir (Tablo 12). Özellik seçme yöntemi

uygulanmadan TBA yönteminin tüm özellikleri DVM'nin bire karşı bir yöntemi (RTF kernelli) en yüksek sınıflandırma doğruluğuna sahip olduğu görülmektedir (Tablo 13). Sadece skor değerlerine özellik seçme yöntemi uygulandığında ise BBA'lı DVM'nin bire karşı bir yöntemi (RTF kernelli) en yüksek sınıflandırma doğruluğuna sahip olduğu görülmektedir (Tablo 14). Özellik seçme yöntemi tüm özelliklere uygulandığında TBA'lı DVM'nin bire karşı bir yöntemi (polinom kernelli) en yüksek sınıflandırma doğruluğuna sahip olduğu görülmektedir (Tablo 15).

Genel olarak sınıflama yöntemine göre başarımların analizi yapıldığında bire karşı bir yöntemi bire karşı kalanlar yöntemine göre daha iyi sınıflandırma başarımına sahip olduğu görülmektedir. Ayrıca BBA algoritması sonuçları, TBA algoritması sonuçlarına göre DVM kullanılarak başarımların analizi yapıldığında, en yüksek sınıflandırma doğruluğu BBA algoritması sonuçlarından elde edildiği görülmektedir. Özellik seçimi olarak sadece skor değerlerinin kullanılması sınıflandırma için en yüksek başarımlar oranını verdiği görülmektedir.

4. ÖNERİLER

Tüm gürültülerin gauss dağılıma sahip olduğu durumda TBA ve BBA yöntemleri bu gürültü işaretlerini birbirinden ayıştıramaz. Bunun için ikinci dereceden istatistiğe dayalı yöntemler hata tespiti için kullanılmalı ve elde edilen sonuçlar TBA ve BBA algoritmalarıyla karşılaştırılmalıdır.

5. KAYNAKLAR

1. Yüksel, T. ve Sezgin, A., Doğrusal Olmayan Dizgeler İçin Model Temelli Arıza Bulma Yalıtımı ve Robot Manipülatörlere Uygulanması, Gazi Üniversitesi Müh. Mim. Fak. Der., 24, 1 (2009) 79-94.
2. Isermann, R. ve Ballé, P., Trends In The Application of Model-based Fault Detection and Diagnosis of Technical Processes, Control Engineering Practice, 5 (1997) 709-719.
3. Zhang, M. Ve Jiang, J., Design of Integrated Fault Detection, Diagnosis and Reconfigurable Control Systems, Proceedings of the 38th IEEE Conference on Decision and Control, 1999, Arizona, 3587-3592.
4. Mahmoud, M., Jiang J. ve Zhang, Y., Active Fault Tolerant Control Systems, Springer-Verlag, Berlin, 2003.
5. Kıyak, E., Kahvecioğlu, A. ve İyibakanlar, G., Uçak Eyleyici Arızasının Tespiti, Yalıtımı ve Sistemin Yeniden Yapılandırılması, Uludağ Üniversitesi mühendislik mimarlık Fakültesi Dergisi, 15, 1 (2010) 13-25.
6. Napolitano, M., An, Y. ve Seanor, B., A Fault Tolerant Flight Control System for Sensor and Actuator Failures Using Neural Networks, Aircraft Design, 3 (2000) 103-128.
7. Mascarell, J. C., Design and Comparison of Neural Network and Fuzzy Logic Actuator Failure Schemes for Flight Control System, Master Thesis, West Virginia University, Department of Mechanical and Aerospace Engineering, West Virginia, 1996.
8. www.aviation-safety.net/index.php, Aviation Safety Network. 14 Kasım 2015.
9. www.planecrashinfo.com/recent.htm, Aviation Accident History. 10 Kasım 2015.
10. www.nasa.gov/centers/langley/news/factsheets/AvSP-factsheet.html, NASA Aviation Safety Program. 14 Kasım 2015.
11. www.planecrashinfo.com/cause.htm, Aviation Accident History, 10 Kasım 2015.
12. Verhaegen, M., Kanev, S., Hallouzi, R., Jones, C., Maciejowski, J. ve Smail, H., Fault Tolerant Flight Control- A Survey, Springer-Verlag, Berlin, 2010.
13. Isermann, R., Fault-Diagnosis Systems, Springer-Verlag, Berlin, 2006.

14. Venkatasubramanian, V., Rengaswamy, R., Yin, K. ve Kavuri, S. N., Review of Process Fault Diagnosis-Part I: Quantitative Model-based Methods, Computers and Chemical Engineering, 27, (2003) 293-311.
15. Venkatasubramanian, V., Rengaswamy, R., Yin, K. ve Kavuri, S. N., Review of Process Fault Diagnosis-part II: Qualitative Models and Search Strategies, Computers and Chemical Engineering, 27, (2003) 313-326.
16. Venkatasubramanian, V., Rengaswamy, R., Yin, K. ve Kavuri, S. N., Review of Process Fault diagnosis-part III: Process History Based Methods, Computers and Chemical Engineering, 27, (2003) 327-346.
17. Kinnaert, M., Fault Diagnosis Based on Analytical Models for Linear and Nonlinear Systems- a tutorial, In IFAC Proceedings of SAFEPROCESS, 2003, Washington, DC, USA.
18. Isermann, R., Schwarte, A. ve Kimmich, F., Model-based fault detection of a Diesel engine with turbo charger –a case study–, IFAC Symposium Automotive Systems, 2004, Salerno, Italy.
19. Frank, P. M. ve Ding, X., Survey of Robust Residual Generation and Evaluation Methods in Observer-based Fault Detection Systems, Journal of Process Control, 7, 6 (1997) 403-424.
20. Chiang, L. H., Russell, E. L. ve Braatz, R. D., Fault Detection and Diagnosis in Industrial Systems, Springer-Verlag, London, 2001.
21. Zhang, Y. M. ve Wu, N. E., Fault Diagnosis for a Ship Propulsion Benchmark: Part I, Proceedings of the 14th Triennial World Congress, 1999, China, 569-574.
22. Yang, Q., Model-based and Data Driven Fault Diagnosis Methods with Applications to Process Monitoring, Ph.D. Thesis, Case Western Reserve University, Electrical Engineering and Computer Sciences, Ohio, 2004.
23. Kleer, J. ve Brown, S., A Qualitative Physics Based on Confluences, Artificial Intelligence, 24, (1984) 7-83.
24. Wang, H., Chai, T. Y., Ding, J. ve Brown, M., Data Driven Fault Diagnosis and Fault Tolerant Control: Some Advances and Possible New Directions, Acta Automatica Sinica, 35, 6 (2009) 739-747.
25. Russell, E. L., Chiang, L. H. ve Braatz, R. D., Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes, Springer-Verlag, London, 2000.
26. Aangenent, W., Kostic, D., Jager, B., Molengraft, R. ve Steinbuch, M., Data-based Optimal Control, American Control Conference, 2005, Portland, USA, 1460-1465.

27. Dong, J., Data Driven Fault Tolerant Control: A Subspace Approach, PhD Thesis, Delft University of Technology, Department of Delft Center for Systems and Control, Delft, 2009.
28. Dong, J., Verhaegen, M. ve Holweg, E., Closed-loop Subspace Predictive Control for Fault Tolerant MPC Design, Proc. of 17th World Congress IFAC Conference, 2008, Korea, 3216-3221.
29. Hallouzi, R. ve Verhaegen, M., Reconfigurable Fault Tolerant Control of a Boeing 747 Using Subspace Predictive Control, AIAA Guidance, Navigation and Control Conference and Exhibit, 2007, Hilton Head, South Carolina.
30. Angeli, C., On-line Fault Detection Techniques for Technical Systems: a Survey, International Journal of Computer Science & Applications, 1, 1 (2004) 12-30.
31. Cheung, J. T. ve Stephanopoulos, G., Representation of Process Trends Part i. a Formal Representation Framework, Computers and Chem. Eng., 14, (1990) 495-510.
32. Rengaswamy, R., Hagglund, T. ve Venkatasubramanian, V., A Qualitative Shape Analysis Formalism for Monitoring Control Loop Performance, Eng. Applications of Artificial Intelligence, 14, 1 (2001) 23-33.
33. Lennox, B., Montague, G. A., Hiden, H. G., Kornfeld, G. ve Goulding, P. R., Process Monitoring of An Industrial Fed-batch Fermentation, Biotechnology and Bioengineering, 74, 2 (2001) 125-135.
34. Crosier, R. B., Multivariate Generalizations of Cumulative Sum Quality-control Schemes, Technometrics, 30 (1988) 291-303.
35. MacGregor, J. F., Yu, H., Munoz, S. G. ve Flores-Cerrillo, J., Data-based Latent Variable Methods for Process Analysis Monitoring and Control, Computers and Chemical Engineering, 29 (2005) 1217-1223.
36. Ocak, H., Fault Detection, Diagnosis and Prognosis of Rolling Element Bearings: Frequency Domain Methods and Hidden Markov Modeling, PhD thesis, Case Western Reserve University, Electrical Engineering and Computer Sciences, Ohio, 2003.
37. Hoskins, J. C., Kaliyur, K. M. ve Himmelblau, D. M., Fault Diagnosis In Complex Chemical Plants Using Artificial Neural Networks, AIChE J., 37, 1 (1991) 137-141.
38. Watanabe, K., Matura, I., Abe, M., Kubota, M. ve Himmelblau, D. M., Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks, AIChE J., 35, 11 (1989) 1803-1812.
39. Ungar, L. H., Powell, B. A. ve Kamens, S. N., Adaptive Networks for Fault Diagnosis and Process Control, Computers and Chem. Eng., 14, 4-5, (1990) 561-572.

40. Chen, J. ve Patton, R. J., Robust Model-Based Fault Diagnosis For Dynamic Systems, Kluwer Academic Publishers, USA, 1999.
41. Timm, N. H., Applied Multivariate Analysis, Springer-Verlag, USA, 2002.
42. Choia, S. W., Leeb, C., Leeb, J. M., Parkc, J. H. ve Leeb, I. B., Fault Detection and Identification of Nonlinear Processes Based on Kernel PCA, Chemometrics and Intelligent Laboratory Systems, 75 (2005) 55-67.
43. Wang, Y., Zhang, M. ve Guo, Y., Fault Detection and Data Restoration Based on PCA for Sensors of Autonomous Underwater Vehicle, International Conference on Mechatronics and Automation (ICMA), 2009, Changchun, 4801-4805.
44. Lee, C., Choi, S. W. ve Leem, I. B., Sensor Fault Identification Based on Time-lagged PCA In Dynamic Processes, Chemometrics and Intelligent Laboratory Systems, 70 (2004) 165-178.
45. Jackson, J. E. ve Mudholkar, G. S., Control Procedures For Residuals Associated with Principal Component Analysis, Technometrics, 21, (1979) 341-349.
46. Sangün, L., Temel Bileşenler Analizi, Ayırma Analizi, Kümeleme Analizleri ve Ekolojik Verilere Uygulanması Üzerine Bir Araştırma, Doktora Tezi, Çukurova Üniversitesi, Fen Bilimleri Enstitüsü, Adana, 2007.
47. Yalçın, T., Temel Bileşenler Yardımıyla Gerçek zamanlı Motor Koruma Algoritması, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Samsun, 2010.
48. Koçer, B. ve Birgören, B., Approaches for Problem Diagnosis via Statistical Process Control Charts, G.U. Journal of Science, 17, 4 (2004) 59-69.
49. Wise, B. ve Gallagher, N., The Process Chemometrics Approach to Process Monitoring and Fault Diagnosis, Journal of Process Control, 6 (6), 329-348.
50. Özçomak, M. S., Çok Değişkenli İstatistiksel Proses Kontrol Metotları, Doktora Tezi, Erzurum Üniversitesi, Sosyal Bilimleri Enstitüsü, Erzurum, 2004.
51. Herault, J. ve Ans, B., Circuits Neuronaux a Synapses Modifiables: decodage de messages composites par apprentissage non supervise, Comptes-Rendus de l'Academie des Sciences, 299, 13 (1984) 525-528.
52. Jutten, C. ve Herault, J., Blind Separation of Sources, part I: An Adaptive Algorithm Based on Neuromimetic Architecture, Signal Processing, 24, (1991) 1-10.
53. Comon, P., Independent Component Analysis: A New Concept, Signal Processing, 36, (1994) 287-314.

54. Sevim, Y., Yapay Toplamsallık Yaklaşımı ile Anne ve Karnındaki Çocuğa Ait EKG İşaretlerinin Ayırıştırılması, Doktora Tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Trabzon, 2009.
55. Hyvarinen, A. ve Erkki, O., Independent Component Analysis: Algorithms and Applications, Neural Networks, 13, 4-5, (2000) 411-430.
56. Hyvarinen, A., Survey on Independent Component Analysis, Neural Computing Surveys 2, 1999, Finland, 94-128.
57. Back, A. D. ve Weigend, A. S., A First Application of Independent Component Analysis to Extracting Structure from Stock Returns, International Journal of Neural Systems, 8, 4 (1997) 473-484.
58. Cardoso, J. F. ve Souloumiac, A., Blind Beamforming for Non-gaussian Signals, IEE Proceedings, 140, 6 (1993) 362-370.
59. Lee, J. M., Yoo, C. K. ve Lee, I. B., Statistical Process Monitoring with Independent Component Analysis, Journal of Process Control, 14, 5 (2004) 467-485.
60. Simoglou, A., Martin, E. B., ve Morris, A. J., Statistical Performance Monitoring of Dynamic Multivariate Process Using State Space Modeling, Comp. Chem. Eng., 26, 6, (2002) 909-920.
61. Lee, J. M., Yoo, C. K. ve Lee, I. B., Statistical Monitoring of Dynamic Processes Based on Dynamic Independent Component Analysis, Chemical Engineering Science, 59, 14 (2004) 2995-3006.
62. Bowman, A. W. ve Azzalini, A., Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plu Illustrations, Clarendon Press, Oxford, 1997.
63. Kohavi, R. ve John, G., Wrappers for Feature Subset Selection, Artificial Intelligence, 97, 1-2 (1996) 273-324.
64. Kohavi, R., Wrappers for Performance Enhancement and Oblivious Decision Graphs, PhD thesis, Stanford University, Department of Computer Science, California, 1995.
65. John, G. H., Kohavi, R. ve Pfleger, K., Irrelevant Features and The Subset Selection Problem, Proceedings of the 1th International Conference on Machine Learning, 1994, California, 121-129.
66. Saunders, C., Grobelnik, M., Gunn, S. ve Shawe-Taylor, J., Subspace, Latent Structure and Feature Selection, Statistical and Optimization Perspectives Workshop, Bohinj, Slovenia, 2005.
67. Vapnik, V. N., The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1999.

68. Botton, L., Cortes, C., Denker, J., Drucker, H. ve Vapnik, V., Comparison of Classifier Methods: A Case Study in Handwriting Digit Recognition, International Conference on Pattern Recognition, 1994, Jerusalem, 77-87.
69. Knerr, S., Personnaz, L. ve Dreyfus, G., Single-layer Learning Revisited: A Stepwise Procedure for Building and Training a Neural Network, Springer-Verlag, Berlin, 1990.
70. Hsu, C. W. ve Lin, C. J., A Comparison of Methods for Multiclass Support Vector Machines, IEEE Transactions on Neural Networks, 13, 2, (2002) 415-425.
71. Friedman, J., Another Approach to Polychotomous Classification, Technical Report, Department of Statistics, Stanford University, 1996.
72. Abaspour, A., Sadeghi, M. ve Sadati, S. H., Using Fuzzy Logic in Dynamic Inversion Flight Controller with Considering Uncertainties, 13th Iranian Conference on Fuzzy Systems (IFSC), 2013, Iran, 1-6.

ÖZGEÇMİŞ

Behnam MAHMOUDI, 22.01.1987 tarihinde İran'ın Tahran Şehrinde doğdu. İlkokulu, ortaokulu ve liseyi Tahran'da okudu. 2005 yılında Farzanegan Lisesi'nden mezun oldu. 2006 yılında Azad Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümü'nde lisans öğrenimine başladı. Öğrenimi sırasında onur alma başarısını da göstererek 4 yılda mezun oldu. 2012 yılında Karadeniz Teknik Üniversitesi, Yabancı Diller Bölümü, İngilizce Hazırlık Sınıfını başarıyla bitirdi. 2012 yılında Karadeniz Teknik Üniversitesi Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başladı. 2013-2014 akademik yılında, Erasmus değişim programı kapsamında Almanya'da Kassel Üniversitesi'nde eğitim almaya hak kazandı ve İngilizce bilmektedir.