KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

EN UZAĞI EN ÖNCE TARA ALGORİTMASI İLE ARIZA DAYANIMLI SİSTEM TASARIMI

DOKTORA TEZİ

Umut ALTINIŞIK

Anabilim Dalı: Elektronik ve Bilgisayar Eğitimi

Danışman: Doç. Dr. Mehmet YILDIRIM

KOCAELİ, 2012

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

EN UZAĞI EN ÖNCE TARA ALGORİTMASI İLE ARIZA DAYANIMLI SİSTEM TASARIMI

DOKTORA TEZİ

Umut ALTINIŞIK

Tezin Enstitüye Verildiği Tarih: 28 ARALIK 2011

Tezin Savunulduğu Tarih: 16 ŞUBAT 2012

Tez Danışmanı
Doç.Dr. Mehmet YILDIRIM
(.....)

Üye
Prof.Dr. Kadir ERKAN
(.....)

Üye
Doç.Dr. Nevcihan DURU
(.....)

Üye
Prof.Dr. Fevzi BABA
(.....)

Üye
Doç.Dr. Celal ÇEKEN
(.....)

KOCAELİ, 2012

ÖNSÖZ VE TEŞEKKÜRLER

Endüstriyel sistemlerde meydana gelen arızalar, sistemlerde performans kaybına, insan yaralanma ve ölümlerine, büyük miktarlarda maddi kayıplara neden olmaktadırlar. Endüstriyel sistemlerde meydana gelen arızaların azaltılması veya kullanılan sistemin arızaya rağmen çalışabilmesini sağlayabilmek için çeşitli arayışlar içine girilmiş ve arıza dayanımlı denetim alanındaki çalışmalar yoğunlaşmıştır. Bu çalışmada bilgi tabanlı arıza dayanımlı denetleyici tasarımı veri madenciliği kullanılarak gerçekleştirilmiştir.

Doktora tez çalışmam boyunca beni yönlendiren, her türlü desteğini esirgemeyen, danışmanım sayın Doç. Dr. Mehmet YILDIRIM'a, tez izleme komitesinde bulunan sayın hocalarım Prof. Dr.Kadir ERKAN ve Doç. Dr. Nevcihan DURU'ya teşekkürlerimi sunuyorum.

Tez çalışmam boyunca sabırla çalışmalarımı destekleyen canım eşim Nurcan'a, aileme ve yardımlarından dolayı değerli arkadaşım Dr. Arş. Gör. Selçuk ÖĞÜTCÜ'ye, canım yavrularım kızım Işıl ve oğlum Mehmet Efe'ye teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜRLER i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iv
TABLOLAR DİZİNİ vi
SEMBOLLER vii
ÖZET ix
İNGİLİZCE ÖZET x
1. GİRİŞ 1
2. ARIZA DAYANIMLI DENETİM SİSTEMLERİ 7
2.1. Arızanın Tanımı 7
2.2. Arızaların Sınıflandırılması 7
2.2.1. Algılayıcı arızası 8
2.2.2. Eyleyici arızası 9
2.2.3. Bileşen arızası 10
2.3. Arıza Dayanımlı Denetim Sistemleri 10
2.4. Arıza Dayanımlı Sistemlerin Kullanım Alanları 13
2.5. Arıza Dayanımlı Denetim Sistemlerin Sınıflandırılması 14
2.5.1. Pasif arıza dayanımlı denetim 15
2.5.2. Aktif arıza dayanımlı denetim 15
2.6. Arıza Tespit ve Tanılama Yöntemleri 16
2.6.1. Model tabanlı arıza tespit ve tanılama yöntemi 17
2.6.2. Bilgi tabanlı arıza tespit ve tanılama 19
2.6.3. Bilgi tabanlı yöntemlerin avantaj ve dezavantajları 22
2.7. Arıza Tespit ve Tanılama Sisteminden Beklenilen Öznitelikler 23
3. VERİ MADENCİLİĞİ VE KÜMELEME 26
3.1. Veri Madenciliği 26
3.2. Veri Madenciliği Teknikleri 27
3.2.1. Birliktelik kuralları 28
3.2.2. Sınıflandırma 30
3.2.2.1. Karar ağaçları 30
3.2.2.2. Bayes modeli 31
3.2.2.3. YSA 31
3.2.2.4. Genetik algoritmalar 32
3.3. Kümeleme 33
3.3.1. K-means kümeleme 37
3.3.2. En uzağı en önce tara algoritması (EUEÖTA) 39
4. ÜÇLÜ TANK İÇİN ARIZA DAYANIMLI DENETİM SİSTEMİ TASARIMI 42
4.1. Üçlü Tank Sistemi ve Matematiksel Modeli 42
4.2. Üçlü Tank Sisteminin PID ile Kapalı Çevrim Denetimi 44
4.2.1. PID denetleyici 44
4.2.2. Algılayıcı arızalarının üçlü tank sistemine etkisi 46
4.3. Algılayıcı Arızalarının İstatistiksel Analizi 54

4.3.1. Verilerin pencerelenmesi 54
4.3.2. Özellik çıkartımı..... 56
4.4. Arıza Dayanımlı Denetim Sistemi Tasarımı..... 58
4.4.1. Arızaların EUEÖTA ile çevrimdışı olarak kümelenmesi 61
4.4.2. Arızaların çevrimiçi tespit ve tanılanması..... 64
4.4.3. Önceden tanımlı arızaların gerçek zamanlı olarak dayanımı..... 65
4.4.4. Önceden tanımlı olmayan arızaların gerçek zamanlı olarak dayanımı 72
4.4.5. Önceden tanımlı olmayan arızaların iyileştirilmiş EUEÖTA yöntemiyle dayanımı..... 74
SONUÇLAR VE ÖNERİLER..... 80
KAYNAKLAR 84
EKLER..... 91
ÖZGEÇMİŞ 111

ŞEKİLLER DİZİNİ

Şekil 2.1: Arızaların zaman karakteristiğine göre sınıflandırılması 8
Şekil 2.2: Algılayıcı arızaları 9
Şekil 2.3: Eyleyici arıza tipleri 10
Şekil 2.4: Genel bir arıza dayanımlı denetim sistemi 12
Şekil 2.5. Arıza dayanımlı denetimin katmanlı gösterimi 13
Şekil 2.6: Arıza dayanımlı denetim sistemlerinin sınıflandırılması 14
Şekil 2.7: Arıza tespit ve tanılama yöntemleri 16
Şekil 2.8: Model tabanlı ATT genel yapısı 18
Şekil 2.9: Süreç model kullanılarak rezidü üretilmesi 18
Şekil 2.10: Bilgi tabanlı ATT genel yapısı 20
Şekil 2.11: Bilgi tabanlı süreç 21
Şekil 2.12: YSA tabanlı ATT yapısı 22
Şekil 3.1: Veri madenciliği süreci 27
Şekil 3.2: Karar ağacı örneği 30
Şekil 3.3: Genetik algoritmaların akış diyagramı 33
Şekil 3.4: Birleştirici algoritma genel akış diyagramı 35
Şekil 3.5: Kümeler arası mesafe hesaplama yöntemleri 36
Şekil 3.6: K-means algoritmasının akış diyagramı 37
Şekil 3.7: En uzağı en önce tara algoritması 39
Şekil 3.8: En uzağı önce tara algoritması ile on noktanın üç kümeye ayrılması 40
Şekil 3.9: EUEÖTA algoritmasına göre hiyerarşik kümeleme örneği 41
Şekil 4.1: Üçlü tank sistemi 42
Şekil 4.2: PID denetleyicinin genel yapısı 45
Şekil 4.3: Kapalı devre üçlü tank sistemi 45
Şekil 4.4: A1 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi 47
Şekil 4.5: A2 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi 48
Şekil 4.6: A3 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi 49
Şekil 4.7: A4 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi 50
Şekil 4.8: A5 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi 51
Şekil 4.9: A6 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi 52
Şekil 4.10: A7 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi 53
Şekil 4.11: Kayan pencere tekniği 54
Şekil 4.12: Tanklara ait seviye algılayıcılarından alınan işaretlerin pencerelenmesi 55
Şekil 4.13: Varyans değişim oranları 58
Şekil 4.14: Tasarlanan kapalı çevrim arıza dayanımlı denetim sistemi. 59

Şekil 4.15: Tasarlanan ATT sistemi genel yapısı 60
Şekil 4.16: WEKA yazılımının genel bileşenleri 62
Şekil 4.17: WEKA Explorer'da arıza küme merkezlerinin EUEÖTA ile bulunması 63
Şekil 4.18. WEKA'da arıza küme merkezlerinin grafik olarak gösterilmesi 64
Şekil 4.19 (a): A5 numaralı arıza için arıza dayanımsız denetim 66
(b): A5 numaralı arıza için arıza dayanımlı denetim 66
Şekil 4.20 (a): A1 numaralı arıza için arıza dayanımsız denetim 67
(b): A1 numaralı arıza için arıza dayanımlı denetim 67
Şekil 4.21 (a): A3 numaralı arıza için arıza dayanımsız denetim 68
(b): A3 numaralı arıza için arıza dayanımlı denetim 68
Şekil 4.22 (a): A7 numaralı arıza için arıza dayanımsız denetim 69
(b): A7 numaralı arıza için arıza dayanımlı denetim 69
Şekil 4.23: Önceden tanımlı tüm arızalar için arıza dayanımsız denetim sonuçları .. 70
Şekil 4.24: Önceden tanımlı tüm arızalar için arıza dayanımlı denetim sonuçları 71
Şekil 4.25: A20 arızasının yanlış tanılanması 73
Şekil 4.26: A20 arızasının doğru tanılanması 76
Şekil 4.27 (a): A10 arızasının ayrıştırma yapılmadan düzeltilmesi 77
(b): A10 arızasının ayrıştırma yapılarak düzeltilmesi 77
Şekil 4.28 (a): A19 arızasının ayrıştırma yapılmadan düzeltilmesi 78
(b): A19 arızasının ayrıştırma yapılarak düzeltilmesi 78
Şekil 4.29 (a): A20 arızasının ayrıştırma yapılmadan düzeltilmesi 79
(b): A20 arızasının ayrıştırma yapılarak düzeltilmesi 79

TABLOLAR DİZİNİ

Tablo 2.1: ATT'da bilgi tabanlı yöntemlerin avantajları ve dezavantajları 23
Tablo 4.1: Algılayıcı arızaları ve özellikleri 46
Tablo 4.2: Arıza küme merkezleri koordinatları 62
Tablo 4.3: Arızaların ADD sistemine uygulanma ve ATT süreleri 70
Tablo 4.4: Önceden tanımlı olmayan arızaların atandıkları arıza kümeleri 72
Tablo 4.5: Ayrıştırma sonrası arıza küme merkezleri koordinatları 74
Tablo 4.6: Ayrıştırma sonrasında önceden tanımlı olmayan arızaların atandıkları arıza kümeleri 75

SEMBOLLER

H_i	: Tank-i su seviyesi
K_p	: Orantısal kazancı
K_i	: İntegral kazancı
K_d	: Türev kazancı
N_s	: Örnekleme sayısı
$Q_{in,i}$	: Pompa-i'den sıvının akış oranı
$Q_{i,0}$	: Tank-i'den dışarı sıvının akış oranı
$Q_{i,j}$	: Tank-i ve j arasındaki sıvının akış oranı
r	: Varyans değişim oranı
r(t)	: Rezidü
R_i	: Valf-i deki drenaj direnci
$R_{i,j}$	: Tank-i ve tank-j arasındaki akış direnci
S_1	: Tank-1 alanı
S_2	: Tank-2 alanı
S_3	: Tank-3 alanı
u(t)	: Sistem girişi
y(t)	: Sistem çıkışı
$y_g(k)$	: Tank gerçek seviyesi
$y_r(k)$	: Tank referans seviyesi
z(t)	: Süreç model çıkışı
σ^2	: Varyans
ρ	: Sıvı yoğunluğu
μ	: Küme merkezi

Alt indisler

n	: Nesne sayısı
p	: Değişken sayısı

Kısaltmalar

ADD	: Arıza dayanımlı denetim
ATT	: Arıza tespit ve tanılama
BFS	: Breadth first search (Enlemesine arama)
DIC	: Dynamic itemset counting (Dinamik nesne kümesi sayma)
EUEÖTA	: En uzağı en önce tara algoritması
FFTA	: Farthest first traversal algorithm (En uzağı en önce tara algoritması)
ISE	: Integral squared error (Karesel hatalar toplamı)
NMPC	: Non-linear model predictive control (Doğrusal olmayan model öngörülü denetim)
PCA	: Principal component analysis (Temel bileşen analizi)

PID	: Proportional-integral-derivative (Orantı-integral-türev)
PLS	: Partial least squares (En küçük kareler)
SETM	: Set-oriented mining (Nesne kümelerine dayalı madencilik)
SOM	: Self organizing map (Öz düzenleyici haritalar)
VTBK	: Veri tabanlarında bilgi keşfi
YSA	: Yapay sinir ağları

EN UZAĞI EN ÖNCE TARA ALGORİTMASI İLE ARIZA DAYANIMLI SİSTEM TASARIMI

Umut ALTINIŞIK

Anahtar Kelimeler: Arıza Tespit ve Tanılama, Üçlü Tank, Veri Madenciliği, Arıza dayanımlı denetim, EUEÖTA

Özet: Endüstriyel sistemlerdeki arızalar denetlenen sürecin beklenmedik tepkisine veya tamamen durmasına neden olabilmektedir. Denetim sistemlerinin, arızanın meydana gelmesi halinde bile sistem başarımını sürdürmeleri hem ekonomik açıdan hem de insan hayatı ve sistem güvenliği bakımından çok önemlidir. Arıza dayanımlı denetimin (ADD) temel amacı, arızalara rağmen sistemin kararlı ve tatmin edici bir başarımla çalışmasını sürdürmesi için uygun bir denetleyicinin tasarlanmasıdır.

Bu tez çalışmasında, üçlü tank sıvı seviyesi denetim sistemindeki algılayıcı arızalarının tespiti ve tanılanması yapılarak, ADD gerçekleştirilmesi amaçlanmaktadır. Günümüzde kullanılmakta olan birçok uygulamada, sistemin matematiksel modelinin tam olarak oluşturulamadığı görülmektedir. Bu nedenle tez çalışmasında, herhangi bir matematiksel modele gereksinim duyulmadan, bilgi tabanlı yöntemler kullanarak arıza tespit ve tanılaması (ATT) yapılmaya çalışılmaktadır. Farklı özelliklere sahip önceden tanımlı olan ve olmayan arızalar üretilerek önerilen ADD sistemine uygulanmıştır. Önceden tanımlı arızaların tamamı ve önceden tanımlı olmayan arızaların da bir kısmı doğru olarak tespit edilmiş ve tanılanmıştır. Önceden tanımlı olmayan arızalar için, EUEÖTA algoritmasında iyileştirme sağlayarak bu arızaların da tamamının doğru olarak tanılanması sağlanmıştır. Sonuç olarak, çalışmada kullanılan ADD sisteminin arızalara rağmen süreç başarımını sürdürdüğü belirlenmiştir.

FAULT TOLERANT SYSTEM DESIGN WITH FFTA

Umut ALTINIŞIK

Keywords: Fault Detection and Isolation, Three Tank, Data Mining, Fault Tolerant Control, FFTA

Abstract: The failures on industrial systems can lead an unexpected reaction or completely stop of the process under the control. Even in the event of failure, it is very important to maintain the system performance with regard to economical, human life and system security issues. The main objective of fault tolerant control (FTC) is to design suitable controller for maintaining stable and satisfactory operation despite the failures of the system.

In this thesis, liquid level control of three tank systems is realized by detecting and isolating the sensor faults, therefore a FTC is achieved. It is observed in so many recent applications that the mathematical model of a system can not be completely determined. For this reason, a fault detection and isolation (FDI) scheme is utilized in this study by using knowledge-based methods without requiring any mathematical model. Predefined and non-predefined fault scenarios are applied to proposed FTC system. All of the predefined and some of the non-predefined faults are detected and isolated correctly. FFTA algorithm is modified in order to detect and isolate the non-predefined faults, correctly. Consequently, the FTC system proposed in this study succeeds on maintaining the performance of the process despite the sensor failures.

1. GİRİŞ

Modern teknolojinin hayatımıza girmesi sonucunda endüstriyel denetim sistemleri daha karmaşık bir yapıya büründüler. Endüstriyel sistemler karmaşıklaştıkça arızalar artmakta ve çeşitli güvenlik, insan ve para kaybı gibi sorunlar meydana gelmektedir. Endüstriyel sistemlerin güvenli, insan yaşamını tehlikeye atmayan, başarılı bir şekilde çalışması ve tüm bunların makul bir maliyetle gerçekleştirilmesi gerekmektedir.

Bir denetim sistemi içerisindeki arızalar; algılayıcı işaretlerinde meydana gelen bozulmalar, eyleyicilerin bozulması veya sistem bileşenlerinde ortaya çıkan bozulmalardan kaynaklanmaktadır. Literatürde, endüstriyel süreçlerde rastlanan bu tip arızalarla ilgili olarak çeşitli örnekler mevcuttur.

Örneğin, 1977 yılında San Dieogo–Los Angeles seferini gerçekleştirmekte olan bir uçakta, meydana gelen bir arızadan dolayı yükselmeyi gerçekleştiren sol taraf eyleyicisi açık pozisyonda sıkışıp kalmış, uçak, burun yüksekliği (pitch) ve yatay yönlendirici (roll) denetim yeteneğini kaybetmiştir. Pilot kabini içerisinde bu tür arızalar için herhangi bir gösterge bulunmamasına rağmen, pilot tecrübesini kullanarak herhangi bir enstrümantasyona dayanmadan, uçaktaki diğer eyleyiciler üzerindeki denetimi sonucunda, uçağı güvenli bir şekilde yere indirebilmiştir [1].

Diğer iki uçak kazası örneğinde ise, arızadan dolayı büyük bir insani ve ekonomik kayıp yaşanmıştır. Bunlardan birincisi, 1971 yılında Chicago-Los Angeles seferini gerçekleştirmekte olan Amerikan havayollarına ait bir uçak, havaalanından havalandıktan kısa bir süre sonra motor arızası sonucunda düşmüş 258 yolcu, 13 mürettebat ve yerde bulunmakta olan 2 kişi hayatını kaybetmiştir. Kaza ile ilgili raporda, uçak düşmeden önce pilotun 15 saniyelik bir reaksiyon süresine sahip olduğu ve bu süre içerisinde kazanın önlenebilir olduğu çeşitli benzetim programları ile gösterilmiştir [2].

İkinci uçak kazası örneğinde ise, 1992 yılında New York-Tel Aviv arasında uçuş gerçekleştirmekte olan bir uçağın arızalanması sonucunda 43 kişi hayatını kaybetmiştir. Uçağın motorlarının ikisi birden Hollanda'nın Amsterdam Schiphol havaalanı üzerinde bozulmuş ve uçak belirli bir süre sonunda çevredeki binalardan birine çarparak büyük bir felakete yol açmıştır. Arıza üzerinde yapılan incelemeler sonucunda, arıza dayanımlı denetiminin kullanılması halinde bu çarpışmanın önlenilebileceği ortaya konmuştur. İnceleme sonucunda, arızaya rağmen uçağın 15 dakika boyunca uçuşa devam ettiği, bu sürenin arızanın tespiti ve çevrimiçi denetleyici ile giderilmesi için yeterli bir süre olduğu kanaatine varılmıştır [3,4].

Uçak kazası örneklerinin yanı sıra, güvenliğin çok önemli olduğu nükleer enerji santrallerinde de tarih boyunca çeşitli kazalar meydana gelmiştir. Bunlardan en önemlileri 1979 yılında meydana gelen Three Miles Island ile 1986 yılında Rusya'da meydana gelen ve binlerce kişinin hayatını kaybetmesine neden olan Çernobil facialarıdır. Bu kazalar sonucunda, nükleer enerji endüstrisinde arızanın tespiti ve arıza dayanımlı denetimin kullanılması zorunluluğu ortaya çıkmıştır.

Bahsi geçen örneklerden de görülebileceği gibi, denetim sistemlerinin, arızanın meydana gelmesi halinde bile sistem başarımını sürdürmeleri hem ekonomik açıdan hem de insan hayatı ve sistem güvenliği bakımından çok önemlidir. Bu nedenle, sistemlerin sadece denetlenmesi yeterli olmayıp, meydana gelebilecek arızaların tespit ve tanılamalarının yapılması, sistemlerin çalışır, güvenli ve arıza dayanımlı hale getirilmeleri gerekmektedir.

Arıza dayanımlı denetim (ADD) sistemi, arıza tespit ve tanılama (ATT) işlemi ile sistemin yeniden yapılandırılmasını içermektedir. ATT ile sistemde bir arıza olup olmadığı, bir arıza söz konusu ise arızanın ne tür bir arıza olduğu, nerede ve ne zaman oluştuğu belirlenebilir. ATT sonucunda denetleyici yeniden yapılandırılır ve sistemin arızaya rağmen başarımı sağlanır. Arıza dayanımlı denetimin temel amacı, tüm algılayıcı, eyleyici ve bileşen arızalarına rağmen sistemin kararlı ve tatmin edici bir başarımla çalışmasını sürdürmesi için uygun bir denetleyicinin tasarlanmasıdır.

Arıza dayanımlı denetim sistemleri aktif ve pasif olmak üzere iki sınıfa ayrılmaktadır. Pasif arıza dayanımlı sistem gürbüz bir denetim yöntemidir. Bu yöntemde, sistem arızalardan duyarsız hale getirilir. Bu nedenle, sistemin yeniden yapılandırılması söz konusu değildir. Aktif arıza dayanımlı sistemde ise, ATT ile arızanın türü, yeri ve zamanı belirlendikten sonra, arızanın çevrimiçi düzeltilmesi için denetleyicinin yeniden yapılandırılması söz konusudur. Arıza dayanımlı sistemler ve ATT ile ilgili literatürde yer alan çalışmalardan bazıları aşağıda verilmektedir.

Saif ve Guan [5], doğrusal ve belirsiz dinamik sistemlerin denetiminde kullanmak üzere gürbüz bir arıza tespit ve tanılama sistemi önermişlerdir. Yöntemin özünü, ölçülemeyen durum değişkenlerinin eşzamanlı olarak tahmin edilmesi ve bu tahminlerin arıza dayanımlı denetimde kullanılması oluşturmaktadır. Yöntemin uygulanabilirliğini göstermek için, dördüncü dereceden dinamik bir sistem olan uçak kalkış ve iniş modeli benzetim çalışmalarında kullanılmıştır.

Dorr ve arkadaşları [6], algılayıcı arızalarının tespit ve tanılamasında kullanmak üzere fiziksel yedeklilik ve analitik yedeklilik olmak üzere iki yöntem önermişlerdir. Algılayıcı işareti yöntemlerin ilkinde yedek algılayıcı işareti ile ikincisinde ise analitik model işareti ile karşılaştırmak suretiyle rezidü üretilmektedir. Yöntemlerin başarısı öncelikle benzetim modeli üzerinde, sonrasında da gerçek nükleer güç santrali üzerinde denenmiştir. Analitik yedekliliğin fiziksel yedekliliğe ilave olarak kullanılması, arıza tespit başarısını artırmakta ve ayrıca algılayıcı arızasına ilave olarak sistem arızalarını da tespit edebilmektedir.

Sauter ve Hamelin [7], arıza etkilerinin sistem bozucularından ayrıştırılamadığı dinamik sistemlerde, arıza tespit ve tanılaması için en uygun filtre tasarımı yöntemi kullanmışlardır. Önerilen yöntem, frekans düzleminde bir performans ölçütünün optimize edilmesine dayanmaktadır. Tasarlanan arıza tespit filtresi, bozucu etkileri minimize ederken arızaların etkisini maksimize etmektedir.

Simani ve arkadaşları [8], dinamik bir sistemin giriş ve çıkış denetim algılayıcılarda meydana gelen arızaların tespit ve tanılaması için, analitik yedeklilik kullanan model

tabanlı bir yaklaşım önermişlerdir. Önerilen yöntem, Kalman filtresi kullanan bir durum tahmincisini ve rezidü analizini içermektedir. Yöntem, bir endüstriyel gaz tribünü modeli üzerinde denenmiş ve benzetim sonuçlarına göre çok küçük arızalar bile tespit edilebilmiştir.

Hou ve arkadaşları [9], binalarda kullanılan ısıtma, havalandırma ve klima sistemlerindeki algılayıcı arızalarının tespit ve tanılaması için, kaba küme teorisi ve yapay sinir ağlarını (YSA) birlikte kullanan ve geçmiş sistem verileri üzerinde veri madenciliğine dayanan bir strateji önermişlerdir. Önerilen yöntem gerçek bir sistem üzerinde denenmiş, veri madenciliğinin algılayıcı arızalarının tespiti ve tanılaması için kullanışlı rezidü üretmede faydalı bir araç olduğu sonucuna varılmıştır.

Huang [10], madeni yağ yalıtkanlı güç transformatörlerinde meydana gelen arızaları daha başlangıç aşamasında tespit edebilmek için, çözünmüş-gaz-analizi geçmiş verileri üzerinde veri madenciliği yapabilen, parametreleri genetik algoritma ile ayarlanan bir dalgacık ağları yöntemi önermiştir. Yöntem, transformatör yağında çözünen gaz içerikleri ile bunlara karşılık gelen arıza tipleri arsındaki kesin ilişkileri tanımlayabilmektedir. Önerilen yöntem yüksek doğruluk derecesine ve kısa eğitim süresine sahiptir.

Luo [11], yapmış olduğu tez çalışmasında, matematiksel modeller yerine bilgi tabanına dayanan arıza tespit yöntemleri geliştirmeye çalışmıştır. Yeni veri içerisindeki arızaları sınıflandırabilmek için çok geniş boyutta eski verileri kullanılmıştır. Geliştirdiği bir yaklaşımda normal veriler kullanarak tek-sınıf sınıflayıcı, diğer bir yaklaşımda ise veri madenciliği, örneğin destek vektör makinelerini (support vector machines - SVM) kullanarak çok-sınıf sınıflayıcı tanımlamıştır. B747 100/200 uçağına ait, altı serbestlik dereceli katı gövde modelinden alınan veriler kullanılarak eyleyici arızaları tespit edilmeye çalışılmıştır. Yaklaşımın etkinliği, üzerinde oynama yapılmayan algılayıcı verileri kullanılarak test edilmiş, altı tür arıza için %79,5 başarı sağlanmıştır.

Aydın [12], yapmış olduğu tez çalışmasında, asenkron motorların stator, rotor ve mil yatağı gibi bileşenlerinde oluşan arızaları yumuşak hesaplama ve veri madenciliği

teknikleri ile teşhis etmiştir. Arıza tespiti için motor akımı ve rotor hızı gibi kolayca elde edilebilir sinyalleri kullanmışlardır. Motorun içyapısını ve matematiksel dinamiklerini bilmeye gerek kalmadan; YSA' da, bulanık mantık ve yapay bağışık sistemler gibi yumuşak hesaplama tekniklerini kullanarak kırık rotor, sarım, mil yatağı sürtünmesi ve eksantriklik arızaları başarılı bir şekilde teşhis edilmiştir.

Detroja ve arkadaşları [13], arıza tespit ve tanılaması için olabilirlikli (possibilistic) kümeleme algoritması sunmuşlardır. Önerilen yöntem bulanık c-means kümeleme algoritmasının eksikliklerine çözüm getiren bilgi tabanlı bir yöntemdir. Öncelikle, dörtlü tank sisteminde yöntemin benzetimi yapılmış, daha sonra da gerçek deney düzeneğinde yöntem test edilmiş ve başarılı bulunmuştur.

Aydın ve arkadaşları [14], üç fazlı indüksiyon motorlarda oluşan kırık rotor arızalarının tespitinde, özellik çıkartımı için zaman serili veri madenciliğine dayanan bir yöntem önermişlerdir. Arızaları sınıflandırmak için bulanık c-means kümeleme yöntemini kullanmışlardır.

Aydın ve arkadaşları [15], üç fazlı indüksiyon motorlarda kırık rotor ve kırık bağlantı arızaların tespiti için yapay bağışıklık algoritmasını kullanmışlardır. Kümeleme yöntemi olarak bulanık c-means kümeleme yöntemini ve genetik algoritmaları kullanmışlardır. Önerilen yöntem ile üç fazlı indüksiyon motorlarındaki arızaları başarı ile tespit edilebilmişlerdir.

Postalcıoğlu ve Erkan [16], işaret işleme ve yumuşak hesaplama tabanlı arıza dayanımlı denetleyici tasarımı gerçekleştirmişlerdir. Benzetim çalışması olarak, üçlü tank sistemi üzerinde oluşabilecek algılayıcı arıza türleri incelenmiş, sistemin denetimi için bulanık mantık denetleyici kullanılmıştır. Algılayıcı arızalarının özellik çıkartımı için dalgacık analizi ve istatistiksel yöntemler, kümeleme için ise öz düzenleyici haritalar (self-organizing map-SOM) yöntemleri kullanılmıştır. Yapılan benzetim çalışmaları sonunda, arıza dayanımlı denetleyicinin sistemi kararlı durumda tutarak ve başarımını koruyarak çalışmasına devam ettiği görülmüştür.

Literatürde yer alan çalışmalardan da anlaşılacağı üzere, denetim sistemlerinde görülen farklı türde arızaların tespit, tanılama ve giderilmesinde çok geniş yelpazede yöntem ve teknikler kullanılmaktadır. Son yıllarda, model tabanlı yöntemlerle birlikte giderek artan sayıda bilgi tabanlı yöntemler de kullanılmaktadır. Bu tezin amacı, veri madenciliği kümeleme yöntemlerinden biri olan en uzağı en önce tara algoritması (EUEÖTA) yöntemini uyumlu bir şekilde kullanarak, önceden tanımlı olan ve olmayan algılayıcı arızalarının tespit ve tanılanması için arıza dayanımlı denetim sistemi tasarımı gerçekleştirmektir. Denetim sistemi olarak, üçlü tank denetim sisteminde benzetim çalışmaları yapılmıştır. Tezin diğer bölümlerinde ele alınan konular aşağıda belirtilmektedir.

Bölüm 2'de, arıza dayanımlı denetim sistemleri incelenmiştir. Arıza dayanımlı denetim sistemleri ile ilgili kavramlar açıklanmış, önemi vurgulanmış, kullanıldığı alanlara örnekler verilmiş, arıza çeşitleri anlatılmış, arıza tespit ve tanılama irdelenmiş, literatürde kullanılmış olan arıza tespit ve tanılama yöntemleri ve bunların kıyaslamalarına yer verilmiştir.

Bölüm 3'de, veri madenciliği ve kümeleme konusu incelenmiştir. Genel olarak veri madenciliği ve kapsamı, özelde ise kümeleme, kümeleme yöntemleri, kümeleme yöntemlerinden biri olan EUEÖTA ve bu yöntemden beklenilen durumlar anlatılmaktadır.

Bölüm 4'de, tez çalışmasında önerilen arıza dayanımlı denetim sistemi tasarımı yer almaktadır. Deney düzeneği olarak kullanılan üçlü tank sistemi ve matematiksel modeli verilmiş, arıza tespiti için kullanılan istatistiksel yöntemler açıklanmış, önceden tanımlı olan arızaların tanılanmasında EUEÖTA'nın başarısı incelenmiş, önceden tanımlı olmayan arızaların da tanılanabilmesi için bir uyarlama yöntemi önerilmiş ve bununla ilgili yapılan benzetim çalışmaları ve sonuçları incelenmiştir.

Bölüm 5'de, tez konusu ile ilgili sonuçlar ve öneriler yer almaktadır.

2. ARIZA DAYANIMLI DENETİM SİSTEMLERİ

2.1. Arızanın Tanımı

Ballé ve Isermann [17] arızayı; "fiziksel sistem içerisindeki bileşenler, algılayıcılar ve eyleyicilerden kaynaklanan ve olağan kabul edilen standart koşullardan beklenmedik bir şekilde ve izinsiz olarak sapılması" olarak tanımlamaktadır.

Otomasyon sistemlerindeki arızalar denetlenen sürecin beklenmedik tepkisine veya tamamen durmasına neden olabilmektedir. Arızalar sonucunda, sistemin performansı tamamen yok olabileceği gibi, kabullenilemeyecek derecede ekonomik kayıplar ve insanların yaralanma veya ölümlerine neden olabilecek tehlikeli kazalar meydana gelebilmektedir.

Sistemde meydana gelebilecek basit bir hata kontrol altına alınmaz ise ciddi başarısızlıklara veya performans kaybına neden olabilmektedir. Küçük bir hata kapalı çevrim denetim sayesinde büyüyerek gereksiz yere sistemi arızaya sokabilir veya büyük bir arıza kapalı çevrim sayesinde gizlenebilir ve fark edildiği ilk durumdan önlenemez bir hale gelebilir.

2.2. Arızaların Sınıflandırılması

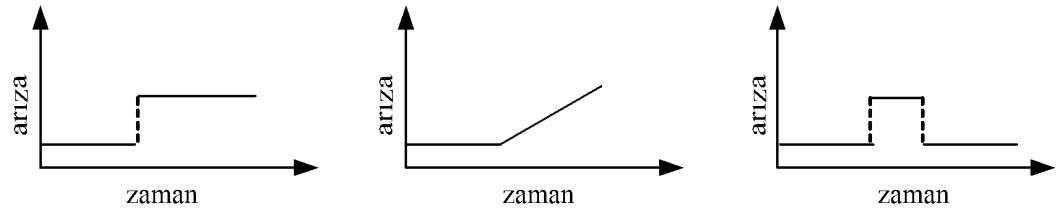
Sistemde oluşabilecek arızalar, arızanın oluşma yeri ve arızanın zaman karakteristiğine göre sınıflandırılabilirler. Arızanın oluşma yeri ile ilgili olarak; algılayıcı arızaları, eyleyici arızaları ve bileşen arızaları karşılaşabileceğimiz arıza çeşitleridir. Arızanın zaman karakteristiğine göre; ani, yeni başlayan ve aralıklarla oluşan arızalar olarak sınıflandırılabilirler.

- Ani arızalar; Donanımsal bozukluklardan dolayı ani olarak meydana gelen arızalardır. Denetim sisteminin performans ve kararlılığını etkiler ve arıza dayanımlı denetim sistemi tarafından çok çabuk sürede tepki verilmesi gerekir.

- Yeni başlayan arızalar; Genel olarak sistemin zamanla eskimesi sonucunda meydana gelmektedir. Zaman içinde parametreli değişiklikler yavaşça olur. Zaman karakteristiğinin yavaşlığından dolayı yeni başlayan arızaların tespiti çok zordur. Fakat yeni başlayan arızalar diğer arızalara göre daha az şiddetlidir.

- Aralıklarla oluşan arızalar; Sistem üzerinde, devamlı bir şekilde, arızalar ortaya çıkar ve kaybolur. Örneğin, elektrik şebekelerinde meydana gelen kısmi arızalardır.

Şekil 2.1'de arızaların zaman karakteristiğine göre sınıflandırılması gösterilmektedir [18].



ani arızalar yeni başlayan arızalar aralıklarla oluşan arızalar

Şekil 2.1: Arızaların zaman karakteristiğine göre sınıflandırılması

2.2.1. Algılayıcı arızası

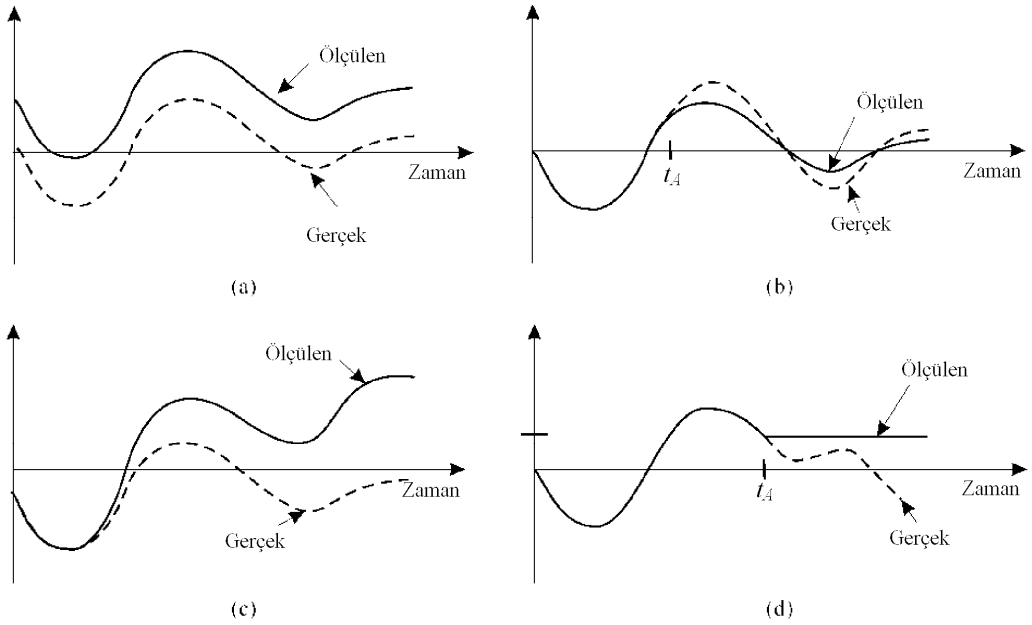
Algılayıcı arızası, algılayıcının işlevini tamamen kaybetmesi veya algılayıcıdan alınan işaretin yanlış ölçülmesinden kaynaklanmaktadır. Sistem üzerinde karşılaşılabilecek başlıca algılayıcı arızaları şunlardır:

- Algılayıcının sabit bir sapma değeri (bias) ile sapma arızası; Algılayıcı sabit bir sapma değeri ile tıkanır ve algılayıcı tarafından sürekli olarak sabit bir sapma farkı ile çıkış üretilir (Şekil 2.2.a).

- Ölçekleme (calibration) veya çarpımsal tipi algılayıcı arızası, Algılayıcı çıkışında meydana gelen ölçekleme hatasından kaynaklanır. Çarpımsal faktör algılayıcının anma değerine uygulanır (Şekil 2.2.b).

- Algılayıcının sürüklenmesi (drift) algılayıcı arızası; Analog algılayıcılarda sıklıkla rastlanan bir arıza türüdür. İç sıcaklık değişimi veya ölçme problemlerinde algılayıcı çıkışına sabit bir değer eklenir (Şekil 2.2.c).

- Tıkama (frozen) arızası; Algılayıcı arızanın meydana geldiği andan itibaren sabit bir tıkama değeri ile çıkış üretir (Şekil 2.2.d).

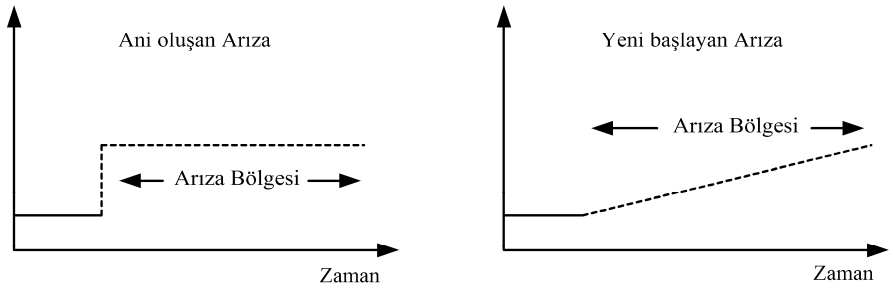


Şekil 2.2: Algılayıcı arızaları [19, 20]

2.2.2. Eyleyici arızası

Eyleyiciler elektriksel sinyali fiziksel harekete çeviren dönüştürücülerdir. Elektrik motorları, pnömatik valfler, hidrolik pistonlar eyleyicilere örnek olarak verilebilir. Eyleyici arızaları iki ana başlıkta toplanmıştır. Bunlar tam eyleyici ve kısmi eyleyici arızalarıdır [21]. Tam eyleyici arızası meydana geldiğinde, eyleyici tarafından denetim işaretlerine herhangi bir tepki verilmez. Kısmi eyleyici arızası meydana geldiğinde ise eyleyici tarafından denetim işaretlerine düşük bir verimle tepki verilir.

Eyleyici arızaları ani veya yeni başlamış arızalar olabilir. Ani arızalar sert arızalar olup denetim hareketini büyük ölçüde etkiler. Elektrik akımındaki kısa devreler veya çevresel faktörlerden dolayı denetlenen sistemde meydana gelen anlık değişimler ani arızalara örnek olarak verilebilir. Yeni başlamış arızalar ise yumuşak arızalardır. Hidrolik veya havalı sistemlerde sızıntıdan dolayı meydana gelen basınç azalması örnek olarak verilebilir. Eyleyici arıza tipleri Şekil 2.3'te gösterilmektedir [19].



Şekil 2.3: Eyleyici arıza tipleri

2.2.3. Bileşen arızası

Bileşen arızaları, genellikle sistemin fiziksel yapısında ve sistemin denklemlerinde değişikliklere neden olan arızalardır. Dağıtık yapısı nedeni ile uçaklar gibi büyük ölçekli sistemlerde, bileşen arızalarının ortaya çıkartılıp tespit edilmesi oldukça zordur. Savaş uçakları gibi sık-sık yapısal zararların görüldüğü yüksek performanslı sistemlerde bileşen arızaları oldukça önemli bir konudur. [19].

2.3. Arıza Dayanımlı Denetim Sistemleri

Kullanmakta olduğumuz sistemlerde meydana gelen arızalar, endüstriyel olarak performans düşüklüğüne, insan yaralanma ve ölümlerine, büyük miktarlarda maddi kayıplara neden olmaktadırlar. Son yıllarda, endüstriyel sistemlerde meydana gelen arızaların azaltılması veya kullanılan sistemin arızaya rağmen çalışabilmesini sağlayabilmek için çeşitli arayışlar içine girilmiş ve arıza dayanımlı denetim alanındaki çalışmalar yoğunlaşmıştır.

ATT yöntemlerinin kullanılması ile birlikte sistem güvenilirliği büyük ölçüde artmış ve işletim maliyetleri azalmıştır. Bu nedenlerden dolayı uçak tasarımı, süreç

denetimi, otomobil tasarımı, uzay ve imalat endüstrilerinde arıza tespit ve tanılama yaygın olarak kullanılmaktadır.

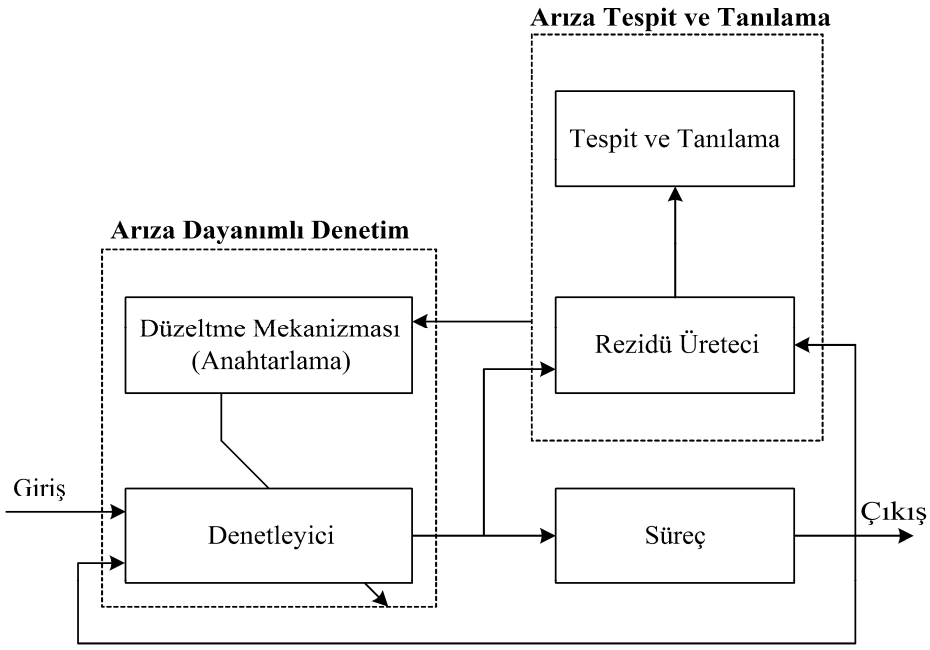
ATT'nın kullanıldığı klasik denetim sistemlerinde bulunan unsurlar iki ana grup altında toplanabilirler. Gruplardan birincisi eylem işlemlerini içerirken, diğeri ise bilginin elde edinilmesi ve işlenmesi görevlerini gerçekleştirir. Birinci grupta olan eyleyici ve sistem arızalarının giderilmesinde donanımsal yöntem kullanılmaktadır. İkinci grupta olan denetleyici ve algılayıcı arızalarında ise donanımsal yedeklik veya yazılımsal yedeklik kullanılmaktadır. Yazılımsal yedeklikte bilgi, farklı denetim amaçlarına göre farklı kararları gerçekleştirmek için çeşitli şekillerde elde edilebilir [22].

Modern teknoloji sistemlerinde başarım ve güvenlik ile ilgili yeterliliklerin sağlanması için karmaşık denetim sistemlerine ihtiyaç duyulmaktadır. Bu sistemler üzerinde başarımı engelleyecek ve kararlı yapıyı bozabilecek çeşitli bileşen arızaları meydana gelebilmektedir. Sistem güvenilirliğinin arttırılması ve istenilen başarım ölçütüne ulaşılması için arızalara karşı dayanıklı bir denetim sistemi tasarımı geliştirilmelidir. Bu sistemler, arıza dayanımlı denetim sistemleri (ADD) olarak adlandırılmaktadırlar. ADD sistemleri, bileşen arızalarını otomatik olarak düzeltme yeteneğine sahip denetim sistemleridir [23].

ADD sistemlerinde; sistemin güvenilirliği, sürdürülebilirliği ve güvenlik çok önemlidir. Güvenirlik, arıza oluşmasına rağmen sistemin başarılı bir şekilde çalışmasıdır. Sürdürülebilirlik, arızanın kolayca giderilebilmesi ve çalışmanın devam ettirilebilmesidir. Güvenlik, insan sağlığı ve denetim sistemlerinin tehlikeye atılmadan sistemin yürütülmesidir [24].

ADD sistemleri çok büyük arızaların meydana gelmesini önlemek için belirli bir seviyede gürbüz olmalıdırlar. Sistem üzerindeki teknik süreçlerin karmaşıklığının artması sonucunda, arızanın meydana gelme olasılığı da artmaktadır. Bu nedenle, güvenlik ile güvenilirlik önemli bir sistem gereksinimi olmaktadır [25]. ADD sistemleri zeki sistemlerdir.

ADD sistemlerinin temel amacı, sistem üzerinde bulunan tüm bileşenlerin normal olarak çalıştığı veya algılayıcı, eyleyici veya diğer sistem bileşenlerinden herhangi birinde bir arıza meydana geldiğinde, sistemin kararlı bir yapıda kalmasını ve tatmin edici bir performansla çalışabilmesini sağlayacak uygun bir denetleyicinin tasarlanmasıdır. Genel bir ADD sisteminin blok diyagramı Şekil 2.4'te gösterilmektedir [19].



Şekil 2.4: Genel bir arıza dayanımlı denetim sistemi

ADD sistemleri, denetim, arıza tanılama ve düzeltme, gözetim ve denetim katmanları olmak üzere üç katmanlı bir yapıda ifade edilebilmektedir [22, 26]. Bu katmanların işleyişi aşağıda verilmektedir.

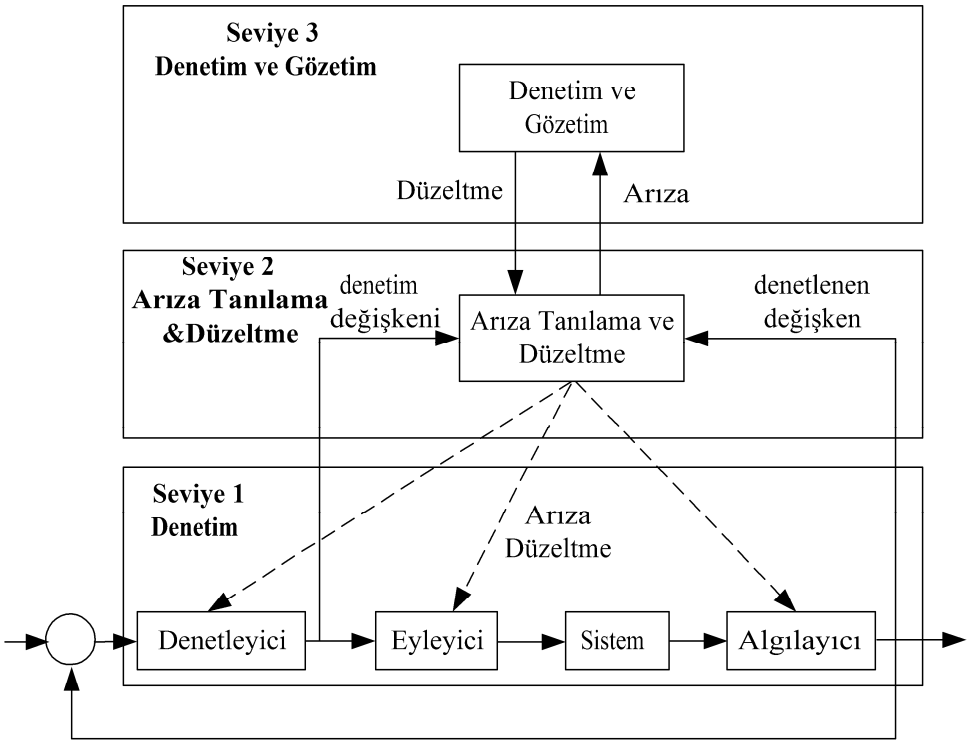
- Denetim katmanı; geleneksel denetim sistemlerinde bulunan algılayıcı, eyleyici, sistem ve denetleyici kısımlarından oluşmaktadır.

- Arıza tanılama ve düzeltme katmanında; tüm arıza tipleri için arıza detektörleri, gözetim ve denetim katmanında yapılmasına karar verilen düzeltme veya iyileştirme işlemlerini gerçekleştiren etkileyiciler yer almaktadır. Bu katmanın işlevleri arasında; donanımsal yedekliliğe dayalı arıza tespiti veya ATT yöntemlerine dayalı analitik yedeklilik yapmak, denetim algoritmalarında veya

uygulama yazılımlarında arıza tespiti yapmak, arızaların düzeltilebilmesi için etkileyici modülleri içermektir.

- Gözetim ve denetim katmanı; denetlenen sistemin mantıksal durumunu gösteren durum-olay mantığını içermektedir. Durumlar arası geçişler gerçekleşen olaylara bağlıdır. Bu katmanın işlevi, detektörlerin arıza tespiti için bir arayüz içermek ve arıza düzeltme aksiyonlarını talep etmektir.

Şekil 2.5'te ADD'in katmanlı yapısı gösterilmektedir [22, 26].



Şekil 2.5. Arıza dayanımlı denetimin katmanlı gösterimi

2.4. Arıza Dayanımlı Sistemlerin Kullanım Alanları

ADD sistemleri, dört ana kategoriye ayrılabilecek çok geniş ve çeşitli mühendislik uygulamalarını içermektedir [20, 23]. Bunlar;

- Güvenlik kritik sistemler; uçak, helikopter, uzay aracı, otomobil, nükleer güç ve tehlikeli kimyasal tesisler.

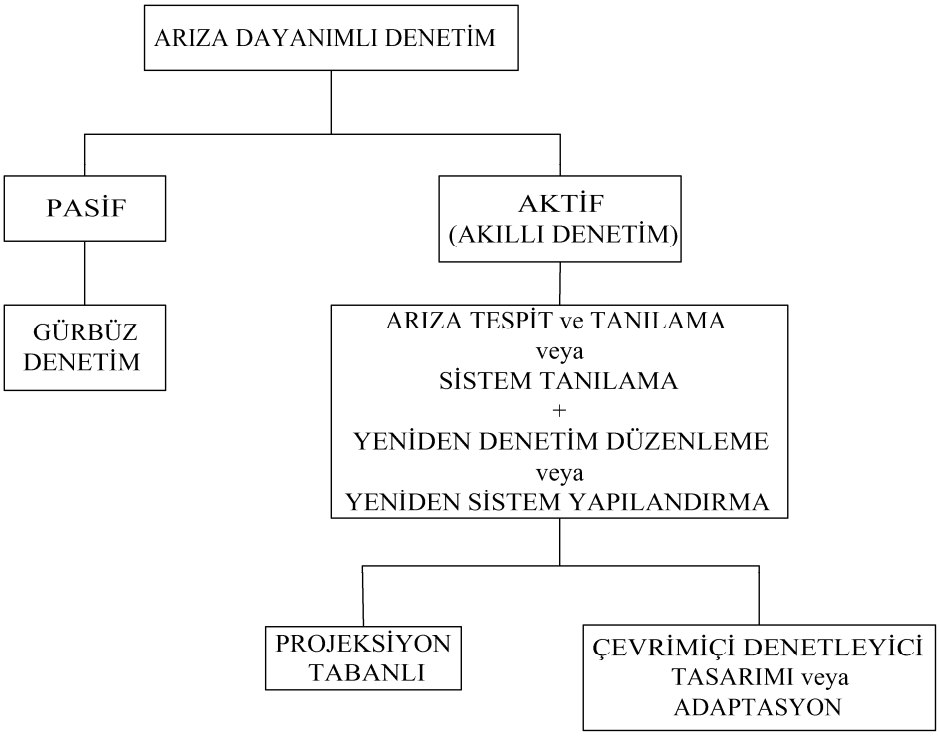
- Yaşam kritik sistemler; tele-robot cerrahi, implante kalp monitörü, nano teşhis aletleri, sayısal protezler, diğer tıbbi cihazlar, ayrıca yer trafik kontrolü ve otomatik otoyol sistemleri.

- Görev kritik sistemler; havacılık elektroniği ve hava kontrol sistemleri, savunma sistemleri, uzay araçları ve uzay istasyonları, otonom hava /uzay /sualtı araçları, sanayileşme sürecinde kullanılan robotlar ve iletişim ağları.

- Maliyet kritik sistemler; büyük ölçekli uzay yapıları, otomobillerin elektronik denetim sistemleri, dağıtık süreç denetimi, bilgisayar ve iletişim ağları.

2.5. Arıza Dayanımlı Denetim Sistemlerin Sınıflandırılması

ADD sistemleri Şekil 2.6'da gösterildiği gibi pasif ve aktif olmak üzere iki sınıfa ayrılmaktadır [27]. Arıza dayanımlı denetim arızaların tespiti, tanılanması ve izole edilmesi yeteneklerini içermekte, bu nedenle, ATT sistemi olarak da adlandırılmaktadır.



Şekil 2.6: Arıza dayanımlı denetim sistemlerinin sınıflandırılması

2.5.1. Pasif arıza dayanımlı denetim

Pasif ADD sistemlerinde, önceden belirlenmiş arıza türlerinin giderilmesi için, yine önceden ayarlanmış denetleyiciler kullanılmaktadır. Arızanın yeri ve mahiyetinin denetleyici devreye girmeden önce bilinmesi gerekmektedir. Pasif ADD sistemlerinde sabit denetleyici parametreleri kullanılır ve çevrim içi hata bilgilerine ihtiyaç duyulmaz. Arızalı sistem aynı denetleyici ve aynı yapı ile çalışmasını sürdürür. ADD sisteminin başarısı, nominal kapalı çevrim sisteminin gürbüzlüğüne bağlıdır. Bu nedenlerden dolayı, pasif türdeki ADD sistemleri sınırlı seviyede arıza dayanım yeteneğine sahiptirler.

2.5.2. Aktif arıza dayanımlı denetim

Aktif ADD sistemlerinde, orijinal seviyede olmasa da sistem başarımının istenilen bir ölçüde tutulması ve gürbüzlüğün sağlanması amacı ile sistem yeniden yapılandırılır. Geri beslemeli denetim sistemlerinin yeniden yapılandırılması veya yeniden yapılandırmasının gerçekleşmesi için, aktif ADD sistemi beklenen arıza türleri için önbilgiye, beklenmeyen arıza türleri için ise arıza tespit ve tanılama mekanizmasına gereksinim duyar.

Aktif yaklaşımlar kestirim tabanlı veya denetleyicinin çevrim içi yeniden tasarlanması yöntemlerinden oluşmaktadır. İlkinde, tanılanan arızanın tipine ve ihtiyaç duyulan denetleyici yapısına göre önceden hesapları yapılmış yeni denetim kuralı seçilir. İkincisi ise, herhangi bir denetim bozulmasına karşılık yeni denetleyici parametrelerinin hesaplanması ile ilgilenir.

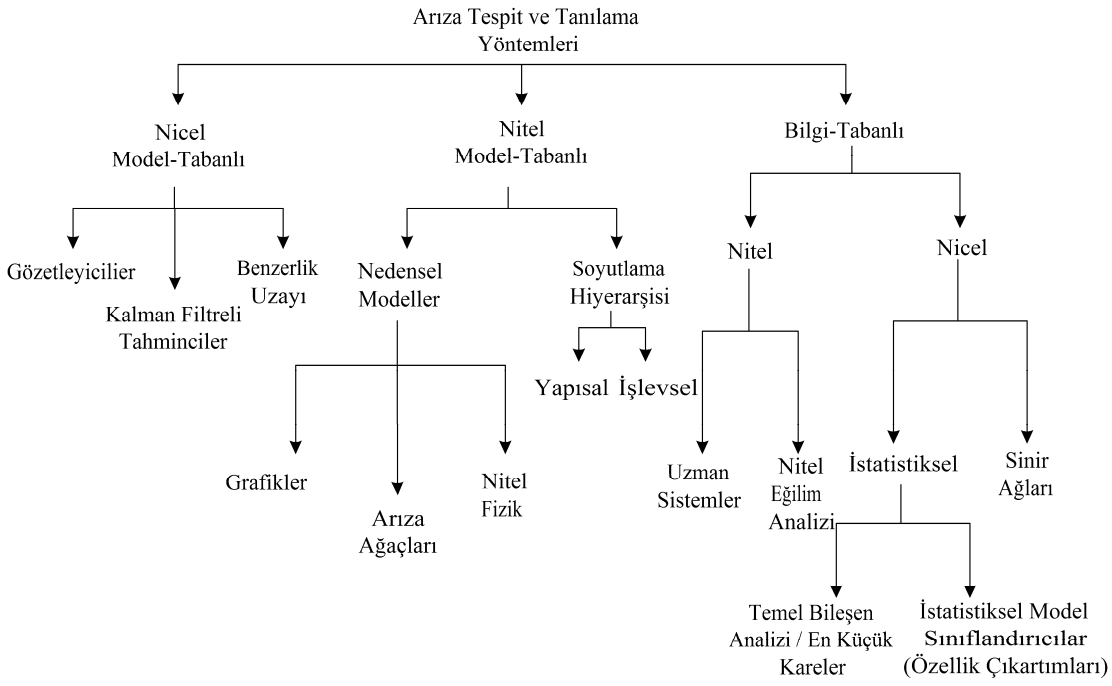
Beklenmedik arızalara dayalı aktif ADD sistemleri, normal olmayan sistem değişikliklerini tespit edebilecek mekanizmaya sahip olmalıdır. Temelde bu, ATT yöntemlerinin işlevidir. ATT yöntemleri kullanılması yeterli olmasına rağmen, bir arızanın türünün, şiddetinin ve arızayı geliştiren nedenlerin bilinmesi önemli olabilmektedir. Eğer ATT ile birlikte bu işlevler de içeriliyor ise, buna arıza teşhisi (fault diagnosis) alt sistemi adı verilmektedir.

Aktif ADD sistemi tasarımı, sadece normal çalışma değil arızalı çalışma durumlarında da geçici ve sürekli hal performanslarını dikkate almalıdır. Bu iki ayrı çalışma modunda sistem davranışları önemli ölçüde farklıdır. Normal çalışma modunda, sistem davranışlarının kalitesi üzerine vurgu yapılırken, arızalı çalışma modunda ise, sistemin kabul edilebilir bir başarımla hayatta kalması baskın konu haline gelir [23].

2.6. Arıza Tespit ve Tanılama Yöntemleri

Arıza tespit ve tanılamada kullanılan yöntemler başlıca iki sınıfa ayrılmaktadırlar. Bunlar, model tabanlı ve bilgi tabanlı arıza tespit ve tanılama yöntemleridir. Endüstriyel arıza tespit sistemlerinde genellikle model tabanlı yaklaşımlar kullanılmaktadır.

ATT yöntemleri, Venkat ve arkadaşları tarafından nicel model tabanlı, nitel model tabanlı ve tarihsel tabanlı yöntemler olmak üzere üçe ayrılmıştır. Arıza tespit ve tanılama yöntemleri Şekil 2.7'de gösterilmektedir [28].



Şekil 2.7: Arıza tespit ve tanılama yöntemleri

Doğrusal ve doğrusal olmayan süreç yaklaşımlarında, arıza tespit ve tanılamanın yapısı model tabanlı olarak oluşturulurken, genellikle model öngörülü denetim (model predictive control) ve genelleştirilmiş olabilirlik oranı (generalized likelihood ratio) yöntemleri birlikte kullanılmışlardır [29-32].

2.6.1. Model tabanlı arıza tespit ve tanılama yöntemi

Model tabanlı arıza tespit sistemi 1970'li yıllarda ortaya çıkmıştır. Model tabanlı ATT yöntemi, analitik bir yedeklilik yapısı içermektedir. Sistemde bulunan elemanların dinamik bağlantılarının analitik bir ilişki ile sağlanması fikrine dayanır [33].

Analitik yedeklilikte, hata işareti sistem ve sistem modeli arasındaki farkın ölçülmesi ile bulunur. Teorik olarak, ölçülen sistem değeri ile model sistem için tahmin edilen değer farklı ise hata oluşmuş demektir.

Model tabanlı ATT yönteminde önemli olan nokta, yedeklerin veya hataların üretilmesidir. Model Tabanlı ATT yöntemi iki temel unsurdan oluşur [34, 35]. Bunlar;

- Rezidü üretimi; Sistemin giriş ve çıkışlarının gözlemlenerek rezidü işaretlerin oluşturulması,

- Rezidü değerlendirmesi; Hata olabilirliği ve karar kuralları için rezidü işaretleri incelenir, hata varsa ortaya çıkarılır.

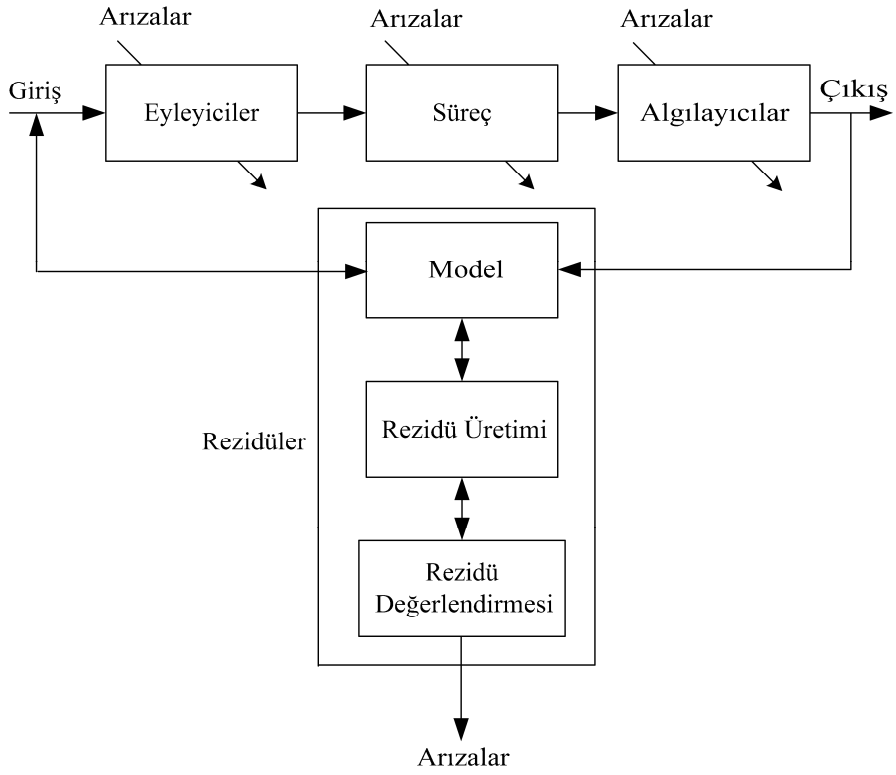
Model tabanlı ATT yöntemi, rezidü üreticini tasarlar ve rezidü işaretini oluşturur. Rezidü işareti kullanılarak ikili sayı sisteminde arıza durumunun olup olmadığına karar verilir. Rezidü işareti kullanılarak arızanın meydana gelip gelmediği tespit edilir ve arızanın neden kaynaklandığı tanımlanır [36, 37].

Model tabanlı ATT yönteminde giriş u(t) ve çıkış y(t) değişkenlerinin matematiksel olarak algılayıcı, eyleyici ve bileşenlerde ölçülmesi ile arıza tespiti gerçekleştirilir.

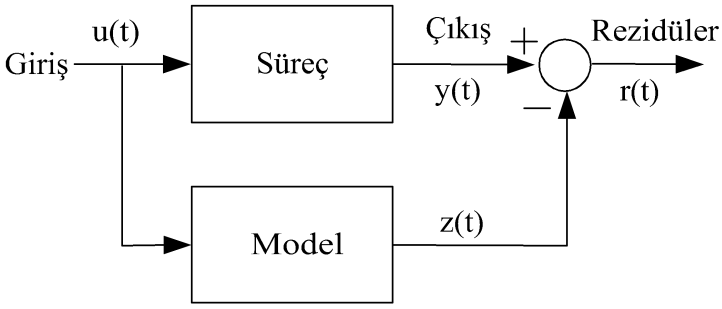
Model tabanlı ATT yönteminde oluşturulan süreç modelindeki z(t) işareti model çıkışını ifade etmektedir. Süreç modelinde oluşturulmuş olan rezidü işareti, Denklem (2.1) ile ifade edilmektedir.

$$r(t) = z(t) - y(t) \tag{2.1}$$

Şekil 2.8 ve 2.9'da Model tabanlı ATT yapısı ve süreç modelin kullanımı ile rezidü üretilmesi gösterilmektedir [34].



Şekil 2.8: Model tabanlı ATT genel yapısı



Şekil 2.9: Süreç model kullanılarak rezidü üretilmesi

Model tabanlı yöntemler kullanıldığında, karşılaşılabilecek temel problemler bellek kullanımı ve bakım maliyetleridir. Model tabanlı yöntemlerde karşılaşılabilecek diğer problemler ise Despande ve arkadaşları [32] tarafından aşağıdaki gibi ifade edilmektedir;

- Model tabanlı arıza tespit ve tanılamada en çok kullanılan Doğrusal Olmayan Model Öngörülü Denetim (Non-linear Model Predictive Control- NMPC) yönteminde, öngörü modeli başlangıç aşamasında ve bir kere gerçekleştirilmektedir. Zaman ilerledikçe, bozucu işaretlerdeki yavaş sapmalar ve süreç parametrelerindeki değişiklikler süreç ve model davranışı üzerinde önemli uyumsuzluklara sebep olabilir.

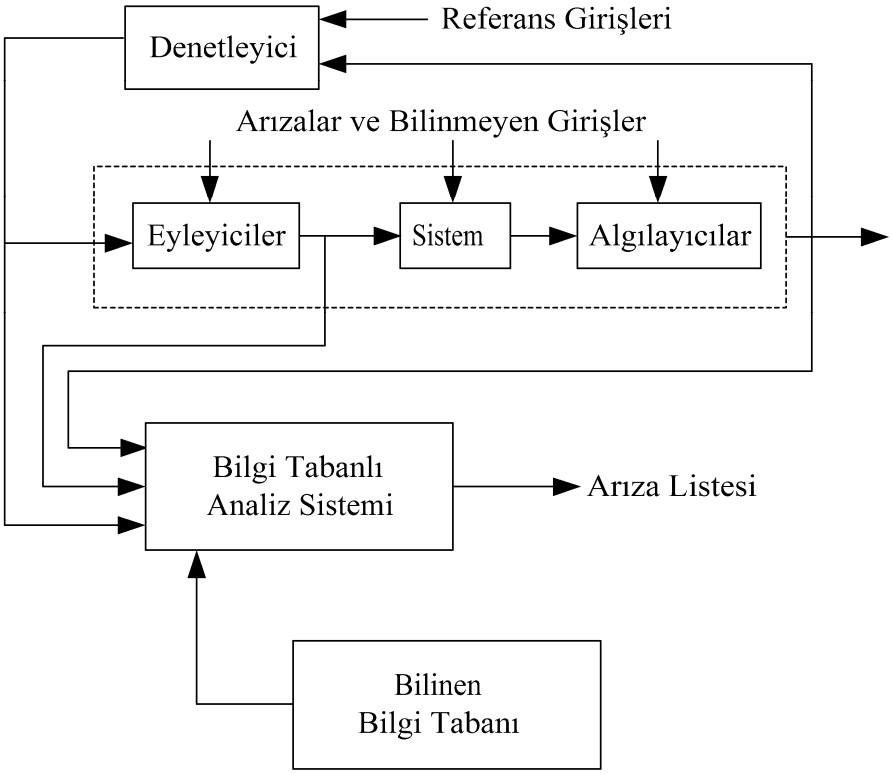
- NMPC yöntemleri genellikle, algılayıcı ve eyleyicilerin arızasız olduğu varsayımıyla geliştirilirler. Bazı algılayıcılar ve eyleyiciler süreç esnasında arıza durumuna düşebilirler. Bu da, model ve sistemin kararsız ve uyumsuz olmalarına neden olur.

Model tabanlı arıza tespitinde, modelin belirlenmesi için; parametre kestirimi, durum kestirimi, eşlik denklemleri yaklaşımları başlıca kullanılan yöntemlerdir [38].

2.6.2. Bilgi tabanlı arıza tespit ve tanılama

Günümüzde kullanılmakta olan birçok uygulamada, sistemin matematiksel modelinin tam olarak oluşturulamadığı görülmektedir. Bu gibi uygulamalarda, model tabanlı yöntemler sistem performansının yeterli düzeyde tutulacağını garanti etmezler. Bu nedenle, performans ölçütünün karşılanabilmesi için, zaman içerisinde bilgi tabanlı yöntemler geliştirilmiştir.

Bilgi tabanlı yöntemler model-bağımsız veya nitel yöntemler olarak isimlendirilmektedirler. Bilgi tabanlı yöntemlerin kullanılmasının sağlamış olduğu en büyük avantaj, herhangi bir matematiksel modele gereksinim duyulmadan, arıza tespit ve tanılamasının daha kolay yapılabilmesidir. Şekil 2.10’da, genel bir bilgi tabanlı ATT yapısı gösterilmektedir [39].

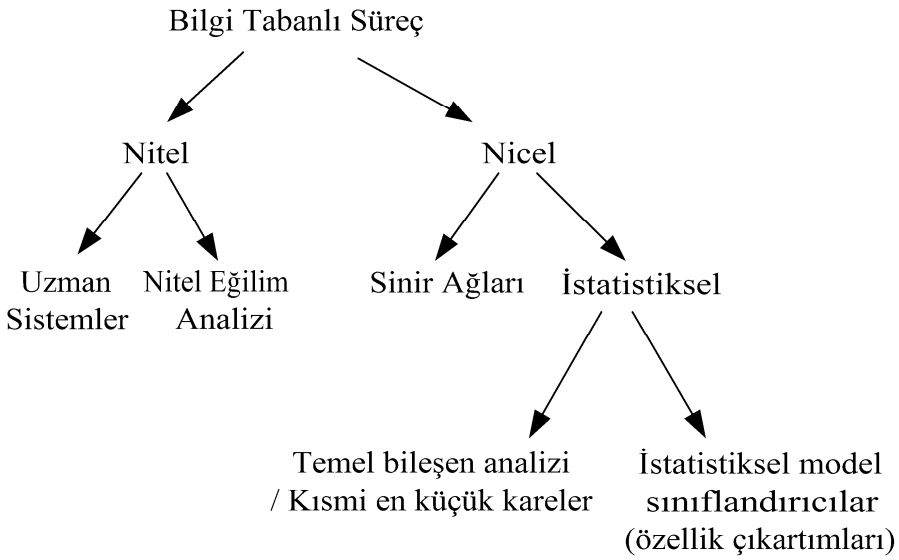


Şekil 2.10: Bilgi tabanlı ATT genel yapısı

Bilgi tabanlı arıza tespit ve tanılamada analitik ve sezgisel bilgi tabanı yaklaşımları söz konusudur. Analitik bilgi tabanı süreç içerisindeki nicel ve analitik bilgilerin kullanılması ile oluşturulur. Sezgisel bilgi tabanının oluşturulmasında ise nitel bilgiler kullanılmaktadır [40].

Bilgi tabanlı yöntemler aynı zamanda, tarihsel tabanlı süreçler (history based process) olarak da ifade edilmektedirler [41]. Tarihsel tabanlı süreçlerde uygulanan ATT sistemleri, kullandıkları yöntemler açısından, nitel ve nicel olarak iki temel sınıfa ayrılmaktadırlar.

Nitel sınıf kendi içerisinde uzman sistemler veya nitel eğilim analizi olarak iki temel sınıfa ayrılmaktadır. Nicel sınıf da yine kendi içinde sinir ağları veya istatistiksel gibi iki temel sınıfa ayrılmaktadır. Bilgi tabanlı yöntemlerin sınıflandırması Şekil 2.11'de [41] gösterilmektedir.



Şekil 2.11: Bilgi tabanlı süreç

Uzman sistemlerde çoğunlukla kural tabanlı özellik çıkartımı kullanılmaktadır. Uzman sistemler aktif ve pasif sistemlerin birleştirilmesi ve bilgi kurallarının IF-THEN mantığında işlenmesi ile oluşur [42].

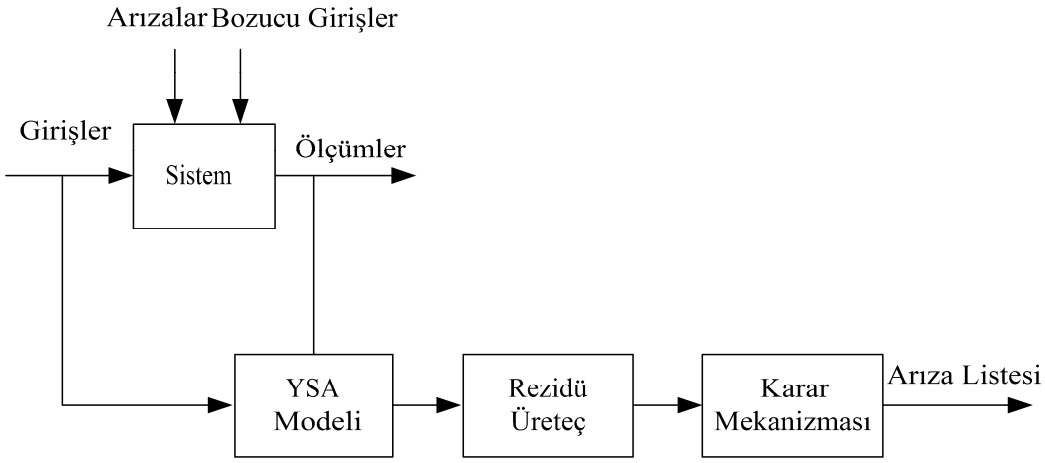
Uzman sistemler bilginin elde edilmesi, giriş ve çıkış ara yüzlerinin geliştirilmesi, bilgi tabanı içinde bilginin kodlanması, tespit süreçlerinin geliştirilmesi gibi bileşenlerden oluşmaktadır. Uzman sistemlerde bilgi tabanı analitik ve sezgisel bilgi olarak ikiye ayrılmaktadır. Tespit sürecinde problemin çözümü analitik ve sezgisel olarak gerçekleştirilmektedir.

Nitel eğilim analizinde sürecin izlenmesinde ve denetiminde eğilim analizi ve tahmin bileşenleri önemli bir rol oynamaktadır. Gürültüye bağlı olarak sinyal üzerindeki anlık değişimlerin saptanmasında, süreçsel bir bakışla sinyal eğilimini inceler. Örneğin, zaman serileri ve sinyaldeki gürültünün önlenmesinde filtre kullanılırken bilinen süreç davranışları kullanılmaktadır.

İstatistiksel yaklaşımda temel bileşen analizi (PCA- Principal Component Analysis), kısmi en küçük kareler yöntemleri (PLS- Partial Least Squares) ve/veya istatiksel model sınıflandırıcılar (özellik çıkartım) yöntemleri kullanılabilmektedir.

Temel bileşen analizi ve en küçük kareler yöntemlerinin amacı, orijinal veri setinin daha küçük parçacıklara ayrılması ile başlıca eğilimlerin bulunmasıdır. İstatistiksel sınıflandırma yönteminde ise Gaussian dağılımının kullanıldığı Bayes sınıflayıcılar kullanılmaktadır.

YSA, önceden sonuçları bilinen veri kümeleri eğitilerek, gerekli olan model ağırlıkları hesaplanmakta ve sisteme yeni gelecek veriler işlenmektedir. Veri kümelerinin eğitiminde kullanılan değerlerin sayısal değerler olması gerekmektedir. YSA ile oluşturulmuş bir ATT yapısı Şekil 2.12'de gösterilmektedir [39].



Şekil 2.12: YSA tabanlı ATT yapısı

2.6.3. Bilgi tabanlı yöntemlerin avantaj ve dezavantajları

Arıza tespit ve tanılamada kullanılan bilgi tabanlı ve model tabanlı yöntemlerin model tabanlı uzman sistemler ve kural tabanlı uzman sistemler ana başlıkları altında avantaj ve dezavantajları Tablo 2.1'de [43] gösterilmektedir.

Tablo 2.1'de gösterildiği gibi, bilgi tabanlı yöntemlerde herhangi bir süreç modeline ihtiyaç yoktur ve arıza tespit ve tanılamasında oldukça etkin ve verimlidir. Ayrıca bilgi tabanlı sistemlerde tümevarım ve tümden gelim işlemleri kolayca yapılabilmektedir.

Tablo 2.1: ATT'da bilgi tabanlı yöntemlerin avantajları ve dezavantajları

Avantajlar	Dezavantajlar
Kural tabanlı uzman sistemler	
• Kurallar kolayca eklenebilir veya kaldırılabilir. • Muhakeme süreci açıklanabilir • Tümevarım ve tümdengelim işlemleri kolaydır. • Bir süreç modeli gerekli değildir. • Arıza tespitinde etkin ve verimlidir	• Genelleme eksikliği vardır. • Yeni durumlara uyumsuz olabilir. • Zamanla değişen olayları temsil etmekte yetersiz kalabilir. • Kendi hatalarından öğrenme eksikliği vardır. • Uzmanlardan güvenilir bilgi elde edilemeyebilir. • Geliştirme ve bakım maliyetlidir.
Model tabanlı uzman sistemler	
• Cihazdan bağımsız tespit ve tanılama vardır. • Bilgi toplama gerekli değildir. • Yeni başlayan arızaları tespit edebilir. • Beklenmedik durumlarla başa çıkabilir. • Tasarım değişikliklerinde esneklik sağlar. • Dinamik arıza tespiti	• Uygulama alanı bağımlıdır. • Arızaların tanılanması zordur. • Bilgi toplama oldukça zahmetlidir.

2.7. Arıza Tespit ve Tanılama Sisteminden Beklenilen Öznitelikler

ATT sisteminden beklenen öznitelikler Venkat ve arkadaşları tarafından başlıca on grup altında toplanmaktadır [32]. Bunlar aşağıda sırası ile kısaca açıklanmaktadır.

- Erken Tespit ve Tanılama; arızanın erken tespiti ve tam tanılanması çok önemli bir özniteliktir.

- Yalıtılabilirlik; sistemde meydana gelen farklı arızaları birbirinden ayırt ederek tanılamasıdır.

- Gürbüzlük; çeşitli gürültü ve belirsizliklerin sistemi tamamen veya aniden başarısız kılması yerine, sistem başarımı ince bir şekilde azaltılarak gürültü etkisi azaltılabilir. Gürültü eşik değeri sıfıra yaklaştıkça, gürbüzlük arıza tanılama kararını zorlaştırmaktadır. Gürültünün mevcut olduğu durumlarda eşik değerleri seçiminde dikkatli olunmalıdır.

- Yeniliklerin çıkartılabilmesi; sistem üzerindeki mevcut sürecin normal veya arızalı bir durum mu olduğu en azından tespit edilebilmelidir. Eğer sistemde arıza tespit edilirse, bu arızanın bilinen veya yeni bir arıza mı olduğunun tanılanması gerekir. Bu kıstas, yeniliklerin tanınabilirliği olarak bilinir.

- Sınıflama hatası tahmini; kullanıcının güvenini sağlamak tanılama sistemlerinde önemli bir gereksinimdir. Bunun sağlanması için, tanılama sisteminin arıza sınıflandırmasında bir hata meydana gelmeden önce hata miktarı hakkında tahminde bulunulmalıdır. Hata ölçümleri sayesinde, tanılama kararlarının güvenilirliği sistem kullanıcısının kendini daha iyi hissetmesini sağlayacaktır.

- Uyumluluk; harici girişlerde veya donanım yapısındaki değişikliklerde süreç de genel olarak değişir veya evrim geçirir. Sistemin çalışma koşulları, sadece bozulmalardan değil aynı zamanda üretim miktarı, kalitesi gibi çevresel etkenlerden de değişir. Yeni durumlar ve problemlerle karşılaştıkça sistemin sahip olduğu bilgi miktarı artar ve arıza tanılama sistemi kademeli olarak yeni amaca adapte olabilmelidir.

- Açıklama imkanı; bir tanılama sistemi arıza kaynağının saptanmasına ek olarak, arızanın nasıl kaynaklandığına ve mevcut sisteme nasıl yayıldığına açıklama getirmesi gerekir. Bu özellik, süreç içerisindeki neden ve sonuç ilişkisinin anlaşılmasını sağlar.

- Modelleme gereksinimleri; arıza sınıflandırıcısının geliştirilmesinde modelleme gereksinimleri miktarı önemli bir konudur. Gerçek zamanlı sınıflandırıcılarının hızlı ve kolay kullanımı için, modellemeye harcanan çaba mümkün olduğunca az olmalıdır.

- Makul depolama ve hesaplama gereksinimi; hesaplama karmaşıklığı veya sistem başarımından birini tercih etmek gerekebilir. Hızlı çevrim içi kararlar, daha az karmaşık bir hesaplamaya ve yüksek kapasiteli bir depolamaya ihtiyaç duyarlar.

- Çoklu arızaların tanınabilirliği; sistemde meydana gelen arızaların birbirlerini etkileyen karakteristiklerinden dolayı, çoklu arızaları tespit etmek zor ve önemlidir.

3. VERİ MADENCİLİĞİ VE KÜMELEME

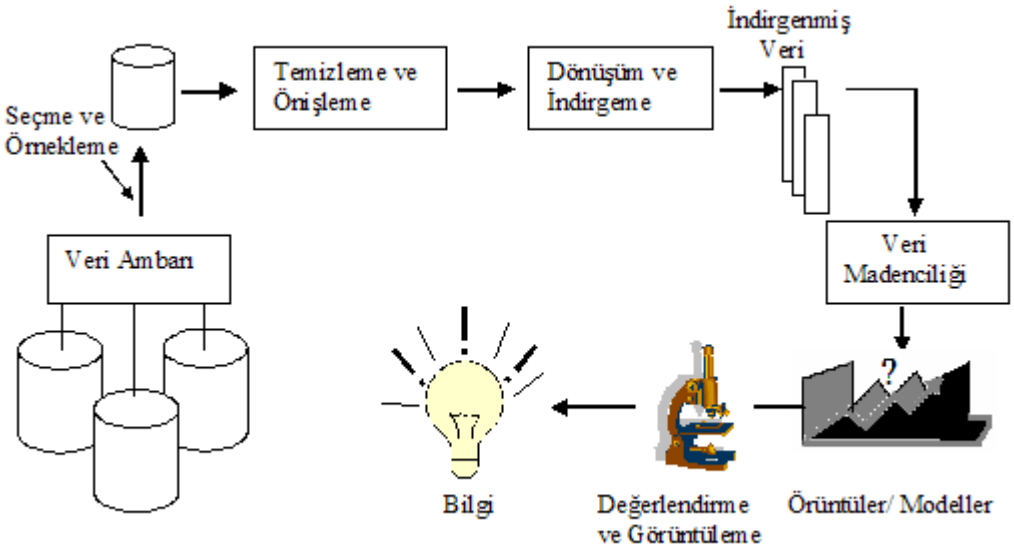
3.1. Veri Madenciliği

Bilim ve teknolojideki hızlı gelişim hayatın her alanına etki etmektedir. Bunun sonucunda, tüm dünya üzerinde toplanmakta olan veri miktarı büyük bir hızla her geçen gün artmaktadır. Hayatın her alanından elde edilen veriler veritabanlarında muhafaza edilmektedir. Veritabanlarında muhafaza edilmekte olan bu verilerden faydalanarak her hangi bir konuda anlamlı sonuçlara ulaşmak çok önemli bir süreçtir. Bu süreç içinde veri madenciliği en önemli safhalardan birisidir.

Veri madenciliği, büyük ölçekli veriler arasından anlamlı örüntülerin keşif ve analiz edilmesi süreci olarak tanımlanmaktadır [44]. Veri madenciliği, çeşitli veri analiz araçları kullanarak, veriler üzerinde gelecek ile ilgili geçerli tahminler yapmamız için kullanılacak örüntü ve ilişkilerin keşfedilmesi süreci olarak da tanımlanmıştır [45].

Veri madenciliği, yazılım mühendisliği kullanılarak elde edilen veri kümelerindeki önceden bilinmeyen kullanışlı verilerin çıkarılması sürecidir [46]. Veri Madenciliği, VTBK (veri tabanlarında bilgi keşfi) sürecinde yer alan bir adımdır [47]. VTBK, veriler içerisindeki potansiyel olarak kullanılabilir olan örüntüleri tespit etmeye yarayan önemli bir süreçtir.

VTBK sürecinde yer alan aşamalar, veri seçimi ve örnekleme, veri temizleme ve önişleme, veri dönüşümü ve indirgeme, veri madenciliği, değerlendirme ve görüntülemedir [48]. Şekil 3.1’de VTBK sürecinde yer alan aşamalar gösterilmektedir.



Şekil 3.1: Veri madenciliği süreci

Küçük veritabanlarında hızlı ve doğru olarak çalışan bir veri madenciliği uygulaması, çok büyük bir veritabanına uygulandığında farklı sonuçlar üretebilmektedir. Veri tabanı boyutu, gürültülü veri, boş (null) değerler, eksik veri, artık veri, dinamik veri ve farklı tipteki verileri ele alma gibi faktörler veri madenciliği sırasında sıkça karşılaşılabilen problemlerdir [49].

Günümüzde veri madenciliği kavramına sıkça rastlamaktayız. Pazarlama ve perakendecilik, bankacılık, sigortacılık, borsa, telekomünikasyon, sağlık ve ilaç, biyoloji ve genetik, endüstri ve mühendislik, sosyal bilimler, davranış bilimleri, metin madenciliği, meteoroloji, uzay bilimleri, görüntü tanıma ve robot görüş sistemleri, web madenciliği gibi birçok alanda veri madenciliği uygulamalarına rastlanmaktadır [50].

3.2. Veri Madenciliği Teknikleri

Günümüzde şirketlerin iş ve stratejik karar alma süreçlerinde en büyük gereksinimlerden biri de, sahip oldukları ham veriyi işleyerek yeni bilgi, fırsat ve aksiyonlar üretmek; ürettikleri bilgiyi piyasa ihtiyaçlarına yanıt verecek biçimde kullanmaktır. Veri madenciliği; finans, bankacılık, perakende, sigorta, telekomünikasyon başta olmak üzere pek çok sektörde, şirketlerin veritabanları veya

veri ambarlarında bulunan verilerden geleceğe yönelik tahminler, modeller elde etmeye olanak sağlamaktadır.

Veri Madenciliği yöntemleri ile şirketler, çok büyük veri yığınlarından önceden bilinmeyen, geçerli ve uygulanabilir değerli bilgiyi kısa sürede elde ederek önemli rekabet avantajları sağlayabilmektedir. Dünyanın önde gelen firmaları, veri madenciliği tekniklerini; müşteri ilişkileri yönetiminden, kredi derecelendirmeye; risk analizinden, satış tahminlerine kadar pek çok alanda başarıyla kullanmaktadırlar.

Veri madenciliğinin zaman içerisinde gelişmesi ile birlikte çeşitli veri madenciliği teknikleri ortaya çıkmıştır. Veri madenciliği algoritmaları Simoudis [51] tarafından iki sınıfa ayrılarak tanımlanmaktadır. Bunlar, doğrulamaya dayalı ve keşfe dayalı veri madenciliği teknikleridir.

Doğrulamaya dayalı veri madenciliğinde, kullanıcı tarafından öne sürülen hipotezden bilgi ortaya çıkarılır. Doğrulamaya dayalı veri madenciliği istatistiksel ve çok boyutlu analiz tekniklerini içermektedir.

Keşfe dayalı veri madenciliğinde, yeni bilgiler mevcut verilerin kullanılması sonucunda ortaya çıkarılır. Keşfe dayalı veri madenciliği birliktelik keşfi, sınıflama, kümeleme, YSA, genetik algoritma gibi yöntemleri içermektedir.

3.2.1. Birliktelik kuralları

Birliktelik kurallarının analiz süreci ilk olarak Agrawal ve arkadaşları [52] tarafından ortaya çıkarılmış ve “market sepeti analizi” olarak adlandırılmıştır. Birliktelik kuralları Han ve Kamber [53] tarafından, büyük miktarlardaki veri yığınlarının içerisinde bulunan ilginç birliktelikleri veya birliktelik ilişkilerini keşfeden veri madenciliği tekniği olarak tanımlanmıştır.

Birliktelik kuralları destek ve güven parametrelerine dayanır. Bu parametrelerin kolay anlaşılabilmesi için bir alış veriş örneğini ele alırsak; A ve B ürünleri arasında %30 destek ve %90 güven oranı değerleri belirlediğimizi düşünelim. Burada

belirtilen %30 destek oranı değeri, tüm alışverişlerde A ve B ürünlerinin %30 oranında birlikte satıldığını ifade etmektedir. %90 güven oranı değeri ise, A ürünü satın alanların aynı alışverişte %90 oranında B ürününü de satın aldığını ifade etmektedir. İki ürün kümesi arasındaki destek parametresi düşük olduğunda faydalı bir bilgi çıkarım sonucuna ulaşma olasılığı önemli ölçüde azalır.

Veri madenciliğinde, oluşturulan tüm birliktelik kurallarının veri tabanı içinde oluşturulmuş olan minimum destek (sık kullanılma kuralı) ve minimum güven eşik değerlerinden (güven kuralı) büyük olması gerekir. Bu işlem iki adımda gerçekleşir [54]. Bunlar;

1. Minimum desteğe sahip olan sık kullanılmış küme setleri bulunur. Sık tekrarlanan öğelerin arama uzayındaki sayısı 2^m'dir.

2. Sık tekrarlanan öğelerden minimum güven değerine sahip güçlü birliktelikler oluşturulur.

Birliktelik kurallarında kullanılan çeşitli algoritmalar vardır. Bu algoritmalar Veri tabanının enlemesine veya derinlemesine arama yöntemleri ile geliştirilmiştir. Enlemesine arama yöntemleri (Breadth First Search-BFS) ile geliştirilen algoritmalardan en önemli olanı Apriori algoritmasıdır. Bir veri madenciliği uygulamasında, Kocaeli Üniversitesinde okumakta olan 2003 yılı öğrencileri için Apriori algoritması kullanılarak derslerin birliktelikleri ortaya çıkarılmıştır [55]. Örneğin, çalışmada elde edilen birliktelik sonuçlarından bir tanesinde; I. dönem dersi olan Matematik-I ve Fizik-I derslerinin her ikisinden birden kalan öğrencilerin %100'ünün II. dönem dersi olan Matematik-II'den de kaldıkları görülmüştür.

Enlemesine arama yöntemleri ile geliştirilen diğer birliktelik algoritmalardan birkaçı; SETM (set-oriented mining) [56], Bölümleme [57] ve DIC (dynamic itemset counting) [58] algoritmalarıdır.

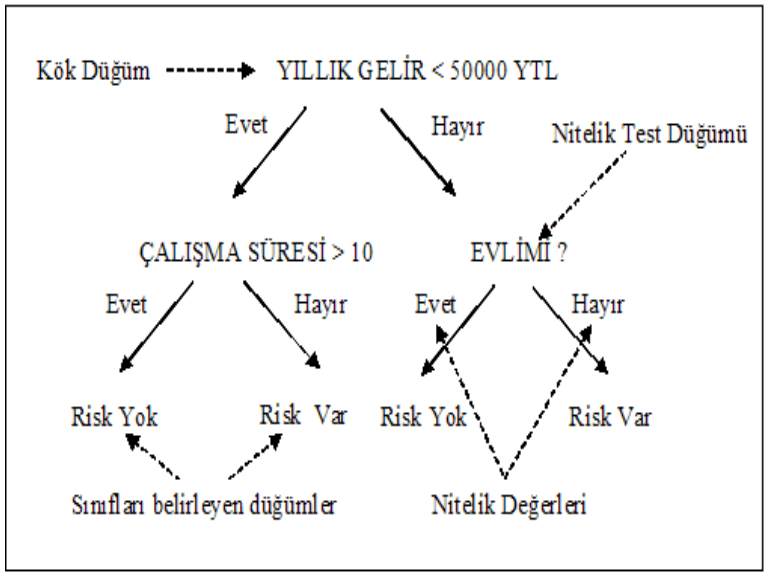
Derinlemesine arama yöntemleri ile geliştirilen algoritmalardan en yaygın olarak kullanılanı Eclat [59] yöntemidir.

3.2.2. Sınıflandırma

Sınıflandırma, önceden belirlenmiş olan sınıflara herhangi bir sınıfa ait olmayan yeni verilerin atanmasıdır [60]. Sınıflandırma modelinde farklı yöntemler kullanılabilir. (IF-THEN) kuralları, karar ağaçları, Bayes modeli [61] veya YSA sınıflandırmada kullanılan başlıca yöntemlerdir.

3.2.2.1. Karar ağaçları

Karar ağaçları, ağaç yapısında olan bir akış şeması içermektedir. Ağacın düğümleri üzerinde nitelik testi işlemi yapılmaktadır. Test işleminin sonucuna göre, ağaç üzerinde dallanmalar oluşturulmaktadır. Karar ağacı, sınıfların oluşturulması ile son bulur. Karar ağaçları; banka kredileri, kredi kartı ve sigorta işlemleri için risk analizi değerlendirmesinde kullanılır. Bir kişiye verilen kredinin risk analizinin, karar ağacı kullanılarak değerlendirildiği örnek bir karar ağacı yapısı Şekil 3.2'de verilmektedir.



Şekil 3.2: Karar ağacı örneği

Karar ağaçlarının kullanımı için çeşitli algoritmalar geliştirilmiştir. Bunlar; Briemen tarafından 1984 yılında geliştirilen CART algoritması, Quinlan tarafından geliştirilen C4.5 algoritması ve Hartigan tarafından 1975 yılında geliştirilen CHAID algoritmasıdır [44].

3.2.2.2. Bayes modeli

Bayes teoremi olasılıklar hesabında önemli bir yere sahiptir. Bayes teoremine dayanarak sınıflandırma yapmak mümkündür. Bayes sınıflandırıcıları istatistiksel sınıflandırma teknikleri arasında yer alır. Basit bir olasılıksal sınıflandırıcıdır. Bayesian kuramı, rastlantısal bir olayın değerlendirilmesinde tüm olasılıkların göz önüne alınarak sonuçların güncellenmesidir. Bayes sınıflandırıcılar olayların veya örneklerin hangi sınıfa hangi olasılıkla ait olduklarını bulur. Örneğin, paranın havaya atıldığında yazı veya tura gelmesi olasılığını %50 olarak değerlendirmek yerine, yazı veya tura gelme olasılığının farklı parametreleri de göz önüne alınarak bir sonucun çıkarılmasıdır. Havadaki o anki rüzgarın hızı, yağmurun şiddeti gibi durumları farklı parametreler olarak düşünebiliriz [61].

3.2.2.3. YSA

Son yıllarda, sınıflandırma problemlerinin çözümünde sınıflandırıcı olarak YSA modelleri oldukça fazla kullanılmaktadır. YSA ile öğrenme insanın deneyimleriyle öğrenmesine benzemektedir [62].

Sınıflandırma problemlerinde, sınıflara ait ayırt edici özelliklerden elde edilen veriler ağın girişini, sınıfları belirleyen ikili düzendeki sayılar da ağın çıkışlarını oluşturmaktadır. YSA'da, özetle, giriş ve çıkış değişkenlerinin ne olacağı, giriş değişkenlerinin 0-1 aralığına dönüştürülmesi, topolojinin belirlenmesi, ağın özel bir eğitim seti ile eğitilmesi ve ağ üzerindeki test setinin test edilmesi gibi işlemler gerçekleştirilir [44, 62].

YSA ile sınıflandırmada en sık uygulanan yöntemlerden birisi SOM (Self Organizing Map)'tir. SOM, Kohonen haritası olarak da adlandırılan, eğiticisiz olarak öğrenme özelliği bulunan YSA modelidir [63]. SOM, veri madenciliğinde kullanılan elverişli bir keşif aracıdır [64]. Kohonen haritalarını ilginç ve kullanılır kılan özellik ise çok boyutlu bir vektörün iki boyutlu izdüşümünün elde edilmesine olanak tanımasıdır. Böylece çok boyutlu bir uzaydaki benzeşimlerin görsel çıkarımı yapılabilmekte ve

iki boyutlu uzaydaki yakın oluşumlara göre girdi verilerinin sınıflandırılması sağlanabilmektedir.

3.2.2.4. Genetik algoritmalar

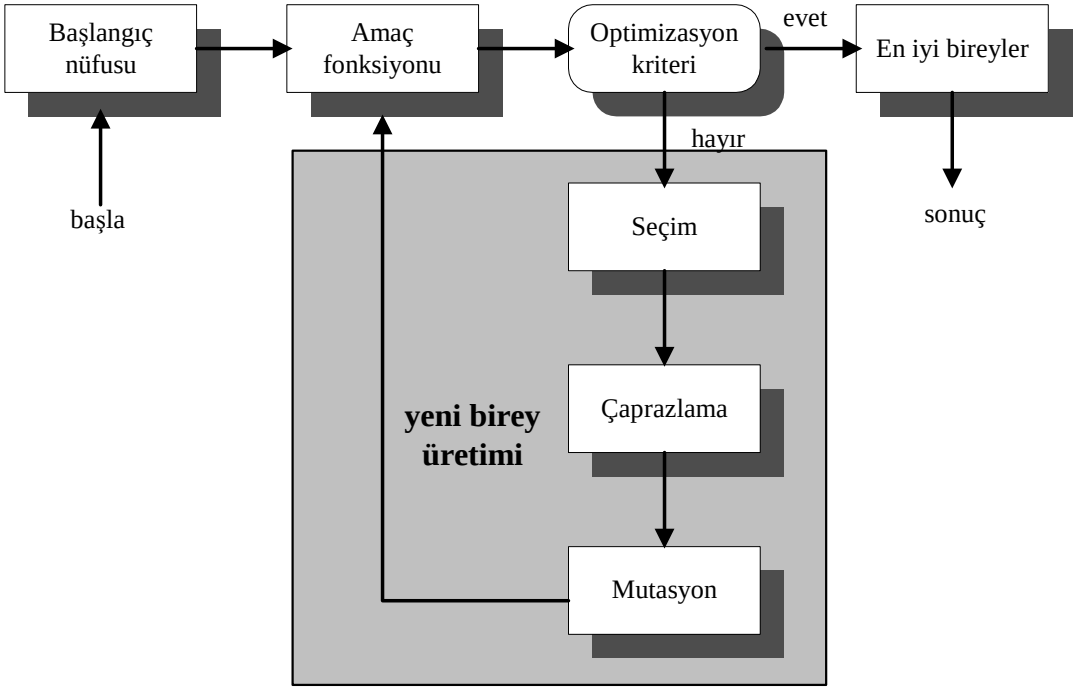
Genetik algoritmalar doğada var olan bir yarışma ortamında, ancak daha iyi ve daha kuvvetli olan bireylere kazanma şansının verildiği, biyolojik olaylara benzetilerek geliştirilmiş, paralel ve küresel bir arama tekniğidir. Arama uzayında aynı anda birçok nokta değerlendirildiği için, küresel çözüme ulaşma olasılığı fazladır. Genetik algoritmalar kod yapıları içinde, en iyi olanların yaşayabilme şansı ile yapılandırılmış olsa da, rasgele bilgi alışverişi olaylarını birleştirerek bir arama algoritması oluştururlar.

Genetik algoritmalar Darwin'in evrim teorisini kullanırlar. Darwin'e göre, mevcut organizmalar arasında meydana gelen üreme, çaprazlama ve mutasyon işlemlerinin sonucunda yeni tür canlılar oluşmaktadır [65].

Genetik algoritmalar yeterli ve etkili bir araştırma gerektiren problemlere uygun bir yaklaşım gerektirdiğinden işletme, fen ve mühendislik alanlarında gittikçe daha fazla uygulanan bir yöntemdir. Genetik algoritmalar; fonksiyon optimizasyonu, görüntü işleme, sistem tanılama, veri madenciliği, kontrol, robot sistemleri, mühendislik tasarımları gibi birçok farklı alanlarda kullanılmıştır [66].

Genetik algoritmalar ile problem çözümünde ilk adım, rasgele üretilen, büyüklüğü veya ne kadar birey içereceği probleme bağlı olan, başlangıç nüfusunun oluşturulmasıdır. Nüfus içerisindeki bireyler, parametre veya değişken bilgilerini temsil eden kodları içermektedir. İkinci adımda, başlangıç nüfusunun bireyleri için, amaç fonksiyon değerleri hesaplanır ve bu değerlere bağlı olarak uygunluk değerleri atanır. Uygunluk değerleri, nüfus içerisindeki bireylerin ne oranda çözüm sağladıklarını gösteren sayısal değerlerdir. Üçüncü adımda, nüfus içerisindeki bireylerin optimizasyon kriterini sağlayıp sağlamadıkları kontrol edilir. Optimizasyon kriteri, bireylerin amaç fonksiyon ve uygunluk değerleri için bir eşik değeri olabileceği gibi, önceden belirlenmiş bir üretim döngü sayısı da

olabilmektedir. Optimizasyon kriteri sağlanmıyorsa, yeni nüfus üretimine geçilir. Yeni (çocuk) bireylerin üretiminde kullanılacak olan mevcut (ebeveyn) bireylerin seçiminde, uygunluk değerleri esas alınır. Çocuk bireylerin üretimi için önce çaprazlama, sonra da mutasyon adımları izlenir. Çocuk bireylerin uygunluk değerleri hesaplanır ve ebeveyn bireylerle beraber yeni nüfus içerisine katılır. Yeni nüfus üretim döngüsü optimizasyon kriteri sağlanıncaya kadar sürdürülür. Optimizasyon kriteri sağlanıyor ise, nüfus içerisindeki uygunluk değeri en iyi olan birey problemin çözümünü oluşturmaktadır. Bu işlem problem çözümünün son adımıdır. Şekil 3.3'te genetik algoritmaların akış diyagramı gösterilmektedir [67].



Şekil 3.3: Genetik algoritmaların akış diyagramı

3.3. Kümeleme

Kümeleme, fiziksel veya soyut nesneleri benzer nesne sınıfları içerisinde gruplama sürecidir [68]. Kümeleme yöntemi örüntü tanıma, veri analizi, görüntü işleme, market araştırmaları, vb. gibi çeşitli alanlarda kullanılmaktadır. Gün geçtikçe veri tabanlarında muhafaza edilmekte olan veri miktarları büyük bir hızla artmaktadır.

Dolayısı ile analiz edilecek veri miktarları sürekli olarak arttığı için, kümeleme yöntemleri de veri madenciliği yöntemleri arasında daha çok kullanılan yöntemler haline gelmektedir.

Kümeleme yönteminde, sınıflandırma yöntemlerinde olduğu gibi kümelerin önceden belirlenmemiş olması en önemli farkıdır. Kümeleme yönteminde veriler iki çeşit yapıda saklanırlar. Bunlar;

- Veri Matrisi (Nesne – Değişken): n sayıda nesne ve p sayıda değişkenden oluşan bir yapıdır. Burada, nesneleri insanlar değişkenleri ise nitelikler (yaş, cins, boy, kilo,…) gibi örnekleyebiliriz. Bu ilişkisel yapı nxp matris (n nesne x p değişken) olarak Denklem (3.1)'de gösterilmektedir.

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{3.1}$$

- Benzemezlik Matrisi (Nesne – Nesne): Tüm nesne çiftlerinin yakınlıklarını nxn tablo içinde saklar. Denklem (3.2)'de bu tablo gösterilmektedir.

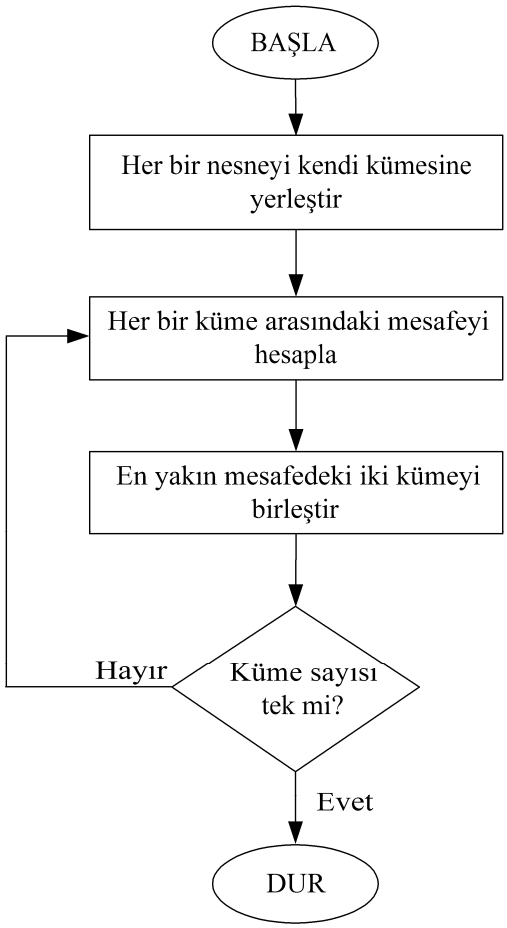
$$\begin{pmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \\ d(n,1) & d(n,2) & \cdots & 0 \end{pmatrix} \tag{3.2}$$

Kümeleme algoritmalarının çoğunda benzemezlik matrisi kullanılmaktadır. Eğer verinin yapısı veri matrisi şeklinde olursa, algoritma uygulanmadan önce yapının benzemezlik matrisine çevrilmesi gerekir.

Geleneksel olarak kümeleme algoritmaları iki ana kategoriye ayrılır. Bunlar, kısımlara ayırmalı ve hiyerarşik algoritmalardır [69]. Her bir yöntemin kendine özgü üstünlükleri, alt tipleri ve kümeleme için kullanılan farklı algoritmaları vardır. Hiyerarşik kümeleme bir sonraki kümeyi bulmak için bir önceki kümeleri kullanır.

Kısımlara ayırmalı kümeleme ise nesnelerin daha önceden belirlenmiş k sayıda kümeye ayrılmasıdır. Hiyerarşik kümeleme nesneler arasındaki uzaklıktan elde edilen bir benzerlik matrisi kullanırken kısımlara ayırmalı kümeleme özellik vektör matrisi kullanır [70].

Hiyerarşik kümeleme, küçük kümelerden daha büyük kümeler yaparak ilerler (birleştirici) veya büyük kümeleri bölerek (bölümleyici) ilerler [46]. Bu kümeleme yönteminde önemli olan hangi küçük kümelerin birleştirileceği veya hangi büyük kümenin bölüneceğidir. Birleştirici algoritmalar bölümleyici algoritmalardan daha hızlı oldukları için daha çok tercih edilmektedirler. Birleştirici algoritmalar aşağıdan-yukarıya (bottom-up), bölümleyici algoritmalar ise tam tersi yukarıdan-aşağıya (top-down) çalışırlar. Hiyerarşik kümeleme yöntemlerinden birleştirici algoritmaların genel akış diyagramı Şekil 3.4'te gösterilmektedir.

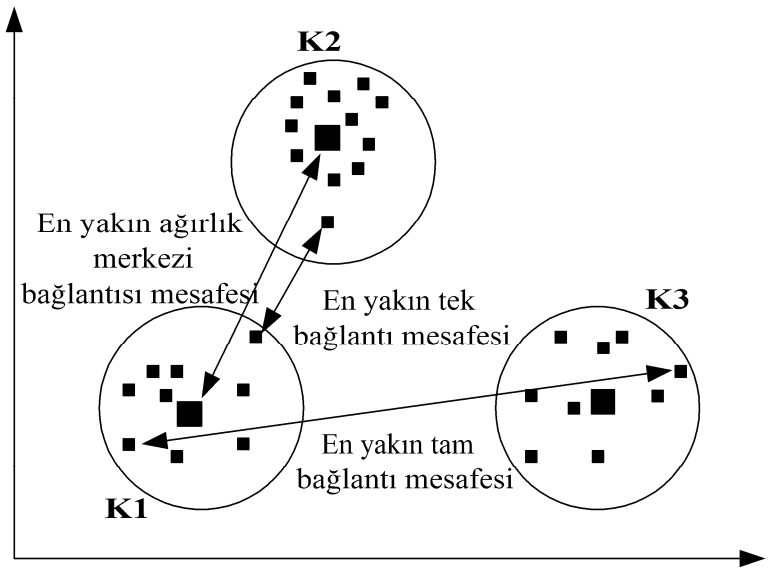


Şekil 3.4: Birleştirici algoritma genel akış diyagramı

Birleştirici kümeleme algoritmalarında kümeler arası uzaklık hesaplanmasında üç adet farklı yaklaşım mevcuttur. Bunlar;

1. Tek bağlantı (single linkage); iki küme arasındaki mesafe birbirine en yakın iki küme üyesi arasındaki mesafeye eşittir.
2. Tam bağlantı (complete linkage); iki küme arasındaki mesafe birbirine en uzak iki küme üyesi arasındaki mesafeye eşittir.
3. Ağırlık merkezi bağlantısı (centroid linkage); iki küme arasındaki mesafe kümeleri ağırlık merkezleri arasındaki mesafeye eşittir.

Kümeler arası mesafe hesaplamada kullanılan tek, tam ve ağırlık merkezi bağıntısı yöntemleri Şekil 3.5'te gösterilmektedir.



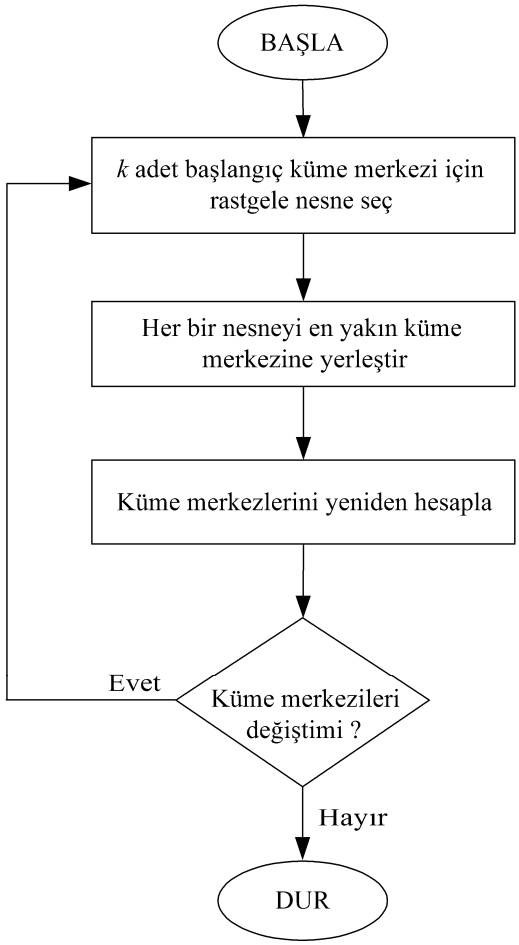
Şekil 3.5: Kümeler arası mesafe hesaplama yöntemleri

Kısımlara ayırmalı kümeleme, verileri doğrudan ayrık kümelere ayrıştırmaya çalışır. Bu yöntemde kümeleme algoritması verinin yerel yapısı üzerine odaklanır. Algoritmalar, tipik olarak her bir kümedeki veri örneklerinin arasındaki benzemezlikleri minimuma indirmeye çalışırken farklı veri kümeleri arasındaki benzemezlikleri maksimuma çıkarmaya çalışır. Yaygın olarak kullanılan bir kısımlara ayırmalı kümeleme yöntemi olan k-means kümeleme, SOM algoritmasıyla yakınlık gösterir [70].

3.3.1. K-means kümeleme

K-means algoritmasının akış diyagramı Şekil 3.6'da gösterilmektedir. K-means algoritmasının işlem basamakları şöyledir:

1. Başlangıç küme merkezleri belirlenir. Bunun için iki farklı yol vardır. Birinci yol nesneler arasından küme sayısı olan k adet rasgele nokta seçilmesidir. İkinci yol ise merkez noktaların tüm nesnelerin ortalaması alınarak belirlenmesidir,
2. Her nesnenin seçilen küme merkez noktalarına olan uzaklığı hesaplanır. Elde edilen sonuçlara göre tüm nesneler k adet kümeden kendilerine en yakın olan kümeye yerleştirilirler,
3. Oluşan kümelerin yeni merkez noktaları, o kümedeki tüm nesnelerin ortalama değeri ile değiştirilir,
4. Küme merkez noktaları sabitlenene kadar 2. ve 3. adımlar tekrarlanır.



Şekil 3.6: K-means algoritmasının akış diyagramı

Nesnelerin birbirleri ile olan benzerlik veya benzemezliklerine göre kümeleme işlemleri gerçekleştirilir. Benzemezlik ve benzerlik ölçümlerinde en yaygın olarak kullanılan mesafe ölçüm yöntemleri Euclidean, Manhattan ve Minkowski yöntemleridir. Euclidean, Manhattan ve Minkowski mesafelerinin hesaplanması Denklem (3.3)-(3.5)'te sırası ile gösterilmiştir.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{3.3}$$

$$d(p,q) = |p_1 - q_1| + |p_2 - q_2| + \ldots + |p_n - q_n| = \sum_{i=1}^{n} |p_i - q_i| \tag{3.4}$$

$$d(p,q) = \sqrt[j]{|p_1 - q_1|^j + |p_2 - q_2|^j + \ldots + |p_n - q_n|^j} = \sqrt[j]{\sum_{i=1}^{n} |p_i - q_i|^j} \tag{3.5}$$

K-means kümelemesinin dört belirgin özelliği vardır. Bunlar;

1. Her zaman k sayıda küme olması,
2. Her küme de en az bir nesne olması,
3. Kümeler hiyerarşik olmamalı, ayrıca her hangi bir örtüşme de olmamalıdır,
4. Kümelerin her elemanı, kendine diğer kümelerden daha yakın olmalıdır. Çünkü yakınlık her zaman kümelerin merkezlerini kapsamaz.

Veri sayısı çok fazla olan hesaplamalarda, k-means hesaplaması, eğer k küçük ise hesaplamaları hiyerarşik kümelemeden daha hızlı yapar. Yine k-means hesaplaması, eğer kümeler özellikle küresel ise, hiyerarşik kümelemeden daha sıkı bir kümeleme yapacaktır. Bunun yanında k-means algoritmasının en büyük eksikliği k değerini tespit edememesidir. Bu nedenle başarılı bir kümeleme elde etmek için farklı k değerleri için deneme-yanılma yönteminin uygulanması gerekmektedir [71].

K-means algoritmasının küresel kümelerde, her zaman doğru kümeleri bulamadığı ancak küme sayısı doğru seçildiğinde ayrık ve sıkışık bulutlar şeklindeki kümeleri etkili bir şekilde bulabildiği söylenebilir.

3.3.2. En uzağı en önce tara algoritması (EUEÖTA)

K-means ve türevleri yaygın olarak kullanılan kümeleme algoritmaları olmasına rağmen, birkaç çalışma haricinde, teorik bilgisayar bilimcileri tarafından pek rağbet görmemektedir. Diğer taraftan, kümeleme problemleri olan k-center ve k-median problemlerinin çözümü için pek çok teorik çalışma mevcuttur [72]. Her iki problemde de kümeleme girdileri, Euclide uzayında veya genellikle de metrik uzayda n adet nokta ve önceden belirlenmiş k adet kümedir. Burada amaç, küme yarıçapları veya çaplarına dayanan bir maliyet fonksiyonunu minimize ederken, noktaları C_1, C_2,.. C_k kümelerine ayırmak ve bu kümeler için yine metrik uzayda μ_1, μ_2, .. μ_k küme merkezlerini belirlemektir. Kümeleme problemlerinde en çok kullanılan maliyet fonksiyonları aşağıda Denklem (3.6) ve (3.7)'de verilmektedir:

k-center: maksimum küme yarıçapları

$$\max_{j} \max_{x \in C_j} d(\mu_j, x) \tag{3.6}$$

k-median: ortalama küme yarıçapları

$$\sum_{j} \sum_{x \in C_j} d(\mu_j, x) \tag{3.7}$$

Gonzáles [73], k-center probleminin çözümü için EUEÖTA adı verilen bir algoritma geliştirmiştir (Şekil 3.7). Bu algoritma maksimum küme yarıçapları maliyet fonksiyonunu kullanarak k adet optimum kümeyi bulabilmektedir.

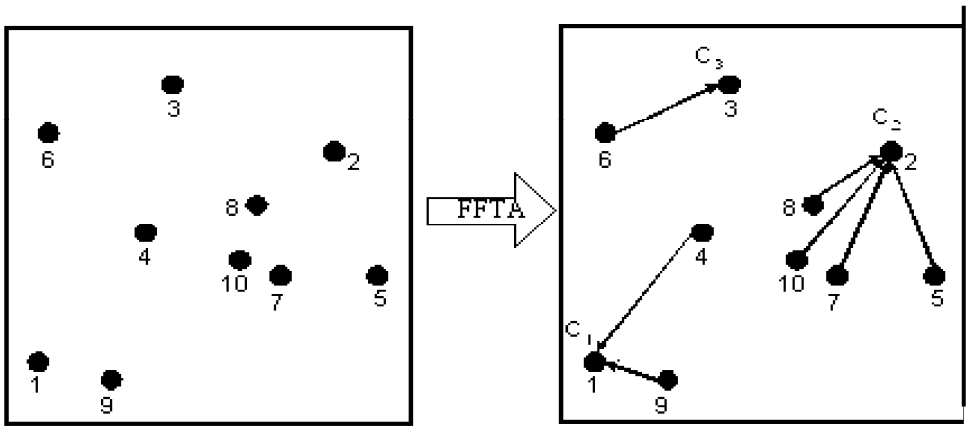
d(.,.) metrik uzaklıklı n adet data noktası gir.
Bir nokta seç ve 1 olarak etiketle.
For i=2,3, …,n
{1,2,…,i-1} noktalarından en uzak noktayı seç ve i olarak etiketle.
$\pi(i) = \arg \min_{j<i} d(i,j)$.
$R_i = d(i, \pi(i))$.

Şekil 3.7: En uzağı en önce tara algoritması

EUEÖTA ile problem çözümü iki aşamada gerçekleştirilir;

- Birinci aşamada; herhangi bir veri noktası başlangıç noktası olarak alınır ve bu noktaya en uzak nokta seçilir. Daha sonra, ilk iki noktaya en uzak olan nokta seçilir ve bu işlem k adet nokta seçilinceye kadar devam ettirilir. Böylece, EUEÖTA k adet veri noktasını sıralı küme merkezleri ($\mu_{1,\ldots,}\mu_n$) olarak yapılandırmış olur.

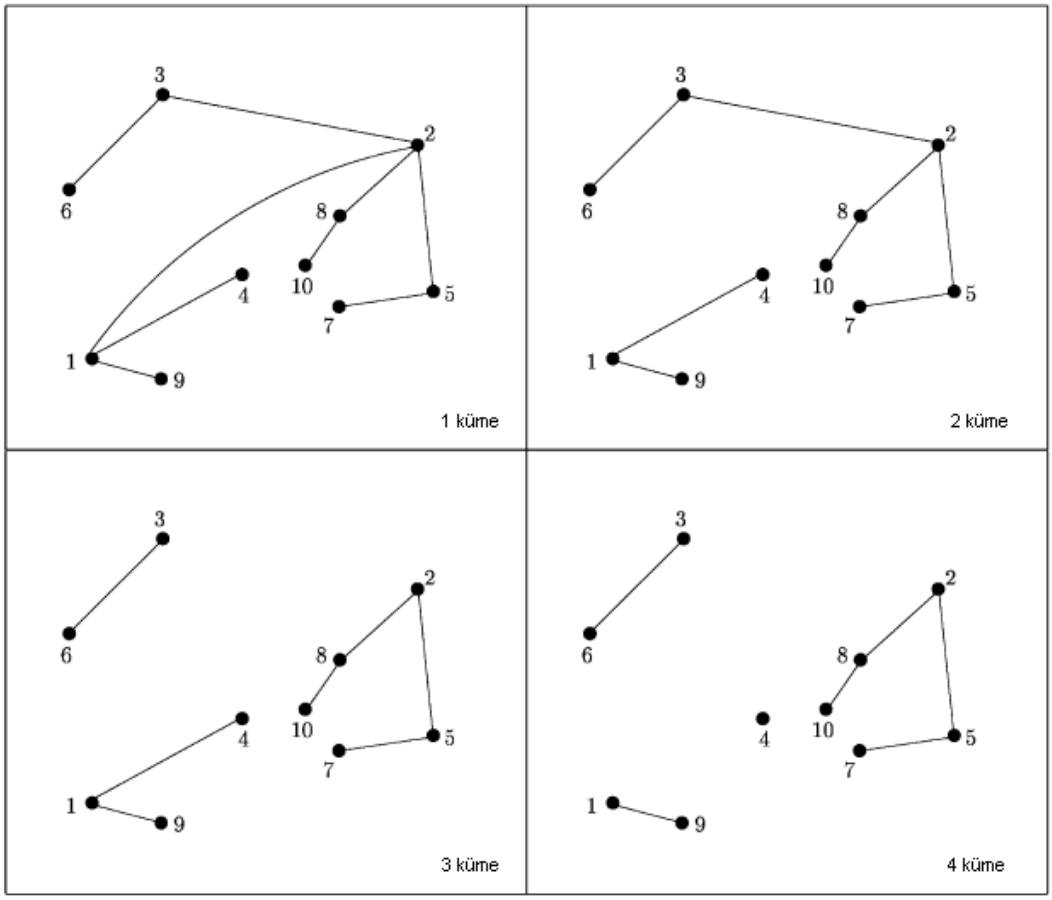
- İkinci aşamada ise; kalan diğer noktalar kendilerine en yakın k adet seçilmiş küme merkezlerinden birine atanır. Şekil 3.8'de rastgele olarak belirlenmiş olan on noktanın EUEÖTA kullanılarak üç kümeye ayrılmasının sonucu gösterilmektedir.



Şekil 3.8: En uzağı önce tara algoritması ile on noktanın üç kümeye ayrılması

EUEÖTA noktaları, herhangi k adet küme merkezine olan uzaklıklarına göre sıralayarak optimum kümeleme yapmaktadır. Burada, k yerine n adet küme merkezi olması durumunda yapılan kümeleme hiyerarşik olmayacaktır. Misal Şekil 3.8'de verilen on noktalı ve üç kümeli örnekte, noktalar üç yerine iki kümeye ayrılsa idi, 6 numaralı nokta C_1 kümesine, 3 numaralı nokta ise C_2 kümesine düşecekti. Oysaki hiyerarşik kümeleme, küçük kümeler tüm elemanlarıyla birlikte birleştirilerek daha büyük kümeler yaparak ilerler (birleştirici) veya büyük kümeler tüm elemanları kapsayan alt kümelere bölünerek (bölümleyici) ilerler. Gonzáles'in geliştirmiş olduğu ve hiyerarşik olmayan EUEÖTA, Dasgupta ve Long [72] tarafından ufak bir ekleme ile hiyerarşik bir kümeleme algoritmasına dönüştürülmüştür.

Kümeleme yapılacak olan bütün noktalar EUEÖTA'ya göre birbirlerine bağlanmaktadır (Şekil 3.9). Sonuçta, oluşan bağlantı şekli bir adet kümeyi göstermektedir. Bu küme, noktalar arasındaki en uzun kenarın silinmesi ile iki alt kümeye bölünmektedir. Oluşan her iki alt küme içerisindeki en uzun kenarın silinmesi ile de üç alt kümeye bölünmektedir. Bu şekilde, k adet küme oluşturabilmek için k-1 adet kenar silinerek istenilen sayıda hiyerarşik kümeleme yapılabilmektedir.



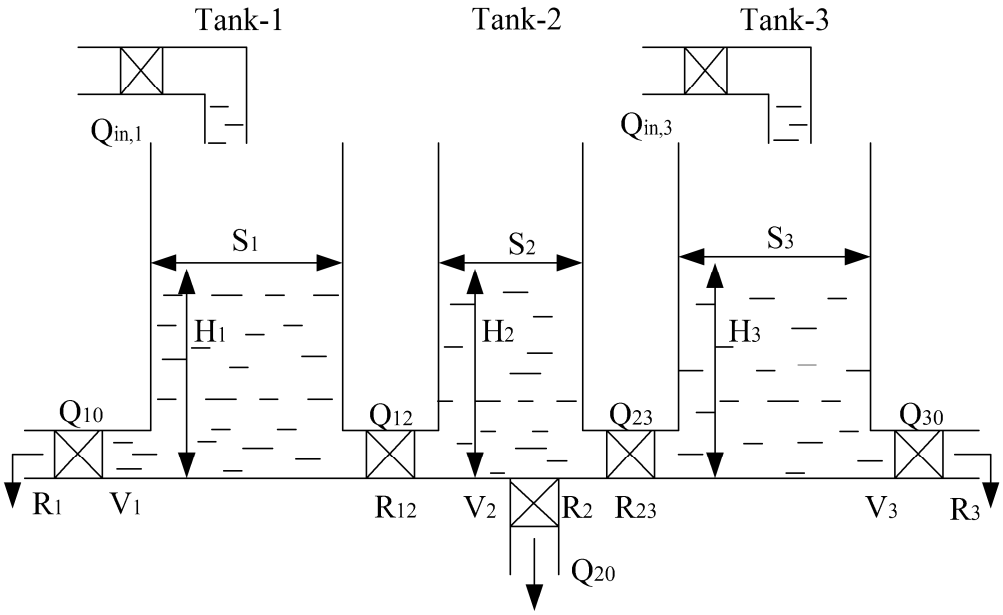
Şekil 3.9: EUEÖTA algoritmasına göre hiyerarşik kümeleme örneği

4. ÜÇLÜ TANK İÇİN ARIZA DAYANIMLI DENETİM SİSTEMİ TASARIMI

Tank sistemlerinde sıvı seviyesinin denetimi, endüstriyel süreçlerin temel problemlerinden birisidir. Tanklar arasındaki sıvı geçişlerinin düzenlenmesi ve sıvı seviyesinin muhafaza edilmesi önemli unsurlardır. Bu amaçla gerçekleştirilmiş olan ikili, üçlü ve dörtlü tank deney sistemleri mevcuttur.

4.1. Üçlü Tank Sistemi ve Matematiksel Modeli

Bu tez çalışmasında, kapalı devre bir denetim sistemindeki arızaların tespiti, tanılanması ve arıza dayanımlı bir denetimin gerçekleştirilmesi için üçlü tank sistemi kullanılmıştır. Üçlü tank sisteminde, birbirine bağlı üç adet su tankı, pompalar ve valflerden oluşmaktadır. Tank-1 ve tank-3'te sisteme sıvı girişini sağlayan iki adet pompa, tank-1 ile tank-2 ve tank-2 ile tank-3 arasında iki adet sıvı geçişini sağlayan valf mevcuttur. Ayrıca, her bir tank için sistemden sıvı çıkışını sağlayan birer adet valf mevcuttur. Üçlü tank sistemi Şekil 4.1'de gösterilmektedir [39].



Şekil 4.1: Üçlü tank sistemi

Üçlü tank sisteminde amaç, tank-2 ile etkileşim halinde olan tank-1 ve tank-3'e ait pompa ve valfleri denetleyerek tank-2'deki sıvı seviyesini istenilen düzeyde tutmaktır. Üçlü tank sisteminde kullanılan denge denklemlerinin matematiksel modeli Denklem (4.1)-(4.4) ile ifade edilmektedir [74].

$$\frac{dH_1}{dt} = -\frac{k_1}{S_1}\sqrt{gH_1(t)} - \frac{k_{12}}{S_1}\text{sgn}(H_1(t) - H_2(t))\sqrt{g\left|H_1(t) - H_2(t)\right|} + \frac{Q_{in,1}}{S_1} \tag{4.1}$$

$$\frac{dH_2}{dt} = -\frac{k_2}{S_2}\sqrt{gH_2(t)} + \frac{k_{12}}{S_2}\text{sgn}(H_1(t) - H_2(t))\sqrt{g\left|H_1(t) - H_2(t)\right|} - \frac{k_{23}}{S_2}\text{sgn}(H_2(t) - H_3(t))\sqrt{g\left|H_2(t) - H_3(t)\right|} \tag{4.2}$$

$$\frac{dH_3}{dt} = -\frac{k_3}{S_3}\sqrt{gH_3(t)} + \frac{k_{23}}{S_3}\text{sgn}(H_2(t) - H_3(t))\sqrt{g\left|H_2(t) - H_3(t)\right|} + \frac{Q_{in,3}}{S_3} \tag{4.3}$$

$$k_1 = \frac{\rho}{R_1}\ ,\ k_2 = \frac{\rho}{R_2}\ ,\ k_3 = \frac{\rho}{R_3}\ ,\ k_{12} = \frac{\rho}{R_{12}}\ ,\ k_{23} = \frac{\rho}{R_{23}} \tag{4.4}$$

Üçlü tank sisteminde kullanılmakta olan tüm değişkenler ve sabitler aşağıda verilmektedir [39]:

S_1 : tank-1 taban alanı (8.107×10^{-3} m^2)

S_2 : tank-2 taban alanı (4.560×10^{-3} m^2)

S_3 : tank-3 taban alanı (8.107×10^{-3} m^2)

ρ: Sıvı yoğunluğu (su için, 1×10^{3} kg/m^3)

H_i : tank-i su seviyesi (m)

$Q_{in,i}$: Pompa-i'den sıvı akış oranı (m^3/sn)

$Q_{i,0}$: tank-i'den dışarı sıvı akış oranı (m^3/sn)

$Q_{i,j}$: tank-i ve j arasındaki sıvı akış oranı (m^3/sn)

R_i : Valf-i deki drenaj direnci (kg/m^5)

$R_{i,j}$: tank-i ve tank-j arasındaki akış direnci (kg/m^5).

4.2. Üçlü Tank Sisteminin PID ile Kapalı Çevrim Denetimi

4.2.1. PID denetleyici

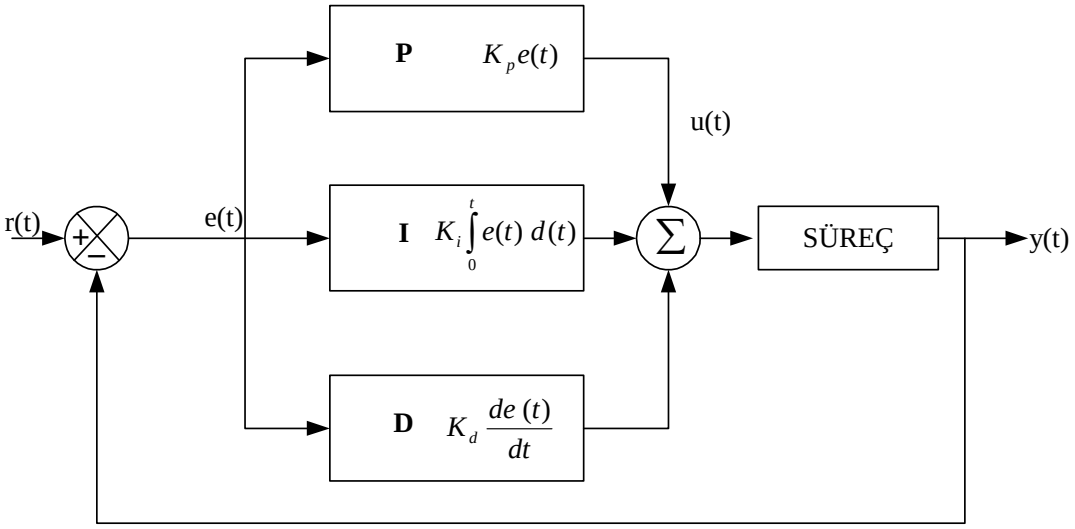
PID (proportional-integral-derivative) denetim, üç temel denetim etkisinin; orantı, integral ve türev birleşiminden meydana gelmekte ve endüstriyel uygulamalarda yaygın olarak kullanılmaktadır. PID denetleyiciler, esneklik ve güvenirlik nedenlerinden dolayı kontrol mühendislerinin geleneksel tercihi olmuşlardır. PID denetimin transfer fonksiyonu;

$$K(s) = K_p + \frac{K_i}{s} + K_d s \tag{4.5}$$

ile ifade edilmektedir. Denklemde, K_p orantısal kazancı, K_i integral kazancını ve K_d türev kazancını temsil etmektedir.

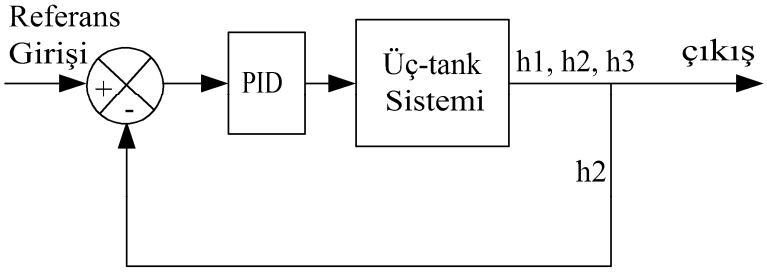
PID denetleyici üç temel denetimin üstünlüklerini tek bir birim içinde birleştiren bir denetleyicidir. P denetim, yalnızca orantı etkisi ile çalışır ve orantı kazancı Kp'nin ayarı ile denetleyicinin denetim duyarlılığı artırılabilir. PI denetim, orantı etkiye integral etki ilavesi ile elde edilmektedir. En yaygın kullanım alanları; basınç, seviye ve akış denetim sistemleridir. İntegral etki denetlenen çıkış büyüklüğünde meydana gelebilecek kalıcı-durum hatalarını ortadan kaldırır. İntegral etkinin kullanım amacı sistemin değişen talepleri üzerinde yeterli bir denetim etkisi sağlamaktır. PD denetim, orantı etkiye türev etki ilavesi ile elde edilmektedir. PD denetim kalıcı-durum hatasını sıfırlayamamakla beraber, bozucu girişten doğan kalıcı-durum hatasının fazla önemsenmediği, fakat buna karşılık orantı etkiye göre geçici-durum davranışının iyileştirilmesi istenen konum-servo mekanizmalarında tercih edilir. İntegral denetim, sistemde ortaya çıkabilecek kalıcı-durum hatasını sıfırlarken türev denetim, yalnızca PI denetim etkisi kullanılması haline göre sistemin aynı bağıl kararlılığı için cevap hızını artırır. Buna göre PID denetim organı sistemde sıfır kalıcı-durum hatası olan hızlı bir cevap sağlar. PID denetleyici diğerlerine göre daha karmaşık yapıda olup o oranda pahalıdır. Burada K_p, K_i ve K_d parametrelerinin uygun seçimi ile uygun bir denetim sağlanabilir [75].

Denetlenecek sistemin dinamik yapısına bağlı olarak üç temel denetim etkisinin mümkün olan en basit bileşimleri kullanılır. PID denetleyicili bir sistemin blok diyagramı Şekil 4.2'de [76] verilmektedir.



Şekil 4.2: PID denetleyicinin genel yapısı

Üçlü tank sisteminde, tank-2 sıvı seviyesi bir denetleyici ile istenilen seviyede tutulurken, tank-1 ve tank-3'e pompalanan veya valflerle bu tanklardan dışarı atılan sıvı akış oranları aynı kabul edilmektedir. Sıvı seviyesi, kapalı çevrim bir denetim sisteminde, tank valflerinin PID denetleyici ile maniple edilmesiyle gerçekleştirilmektedir. Benzetim çalışmalarının gerçekleştirildiği üçlü tank kapalı çevrim denetim sistemi Şekil 4.3'de gösterilmektedir. Sistem üzerinde kullanılan PID denetleyicinin transfer fonksiyonu Denklem (4.6)'da gösterilmektedir.



Şekil 4.3: Kapalı devre üçlü tank sistemi

$$K(s) = 4.8 + \frac{0.02}{s} + 0.5s \tag{4.6}$$

4.2.2. Algılayıcı arızalarının üçlü tank sistemine etkisi

İleriki bölümlerde yapısı ve uygulamaları sunulacak olan arıza dayanımlı denetim sisteminin benzetimini yapabilmek ve ihtiyaç duyulan arıza verilerini elde edebilmek için, PID denetleyicili üçlü tank seviye denetim sisteminin MatLab-Simulink modeli yapılmıştır. Bu model Ek'lerde Şekil A.1-A.4'te verilmektedir.

PID denetleyicinin sistemde meydana gelebilecek arızalara karşı tepkisini inceleyebilmek için benzetim çalışması üzerinde; toplamsal, çarpımsal ve tıkama tipinde algılayıcı arızaları oluşturulmuştur. Sistem modeline, Tablo 4.1'de verilmiş olan yapay algılayıcı arızaları [74] uygulanarak, arızaların sistem üzerindeki etkileri gözlemlenmiş ve bilgi tabanı oluşturmak üzere, her bir arıza durumuna ait sistemin verileri elde edilmiştir.

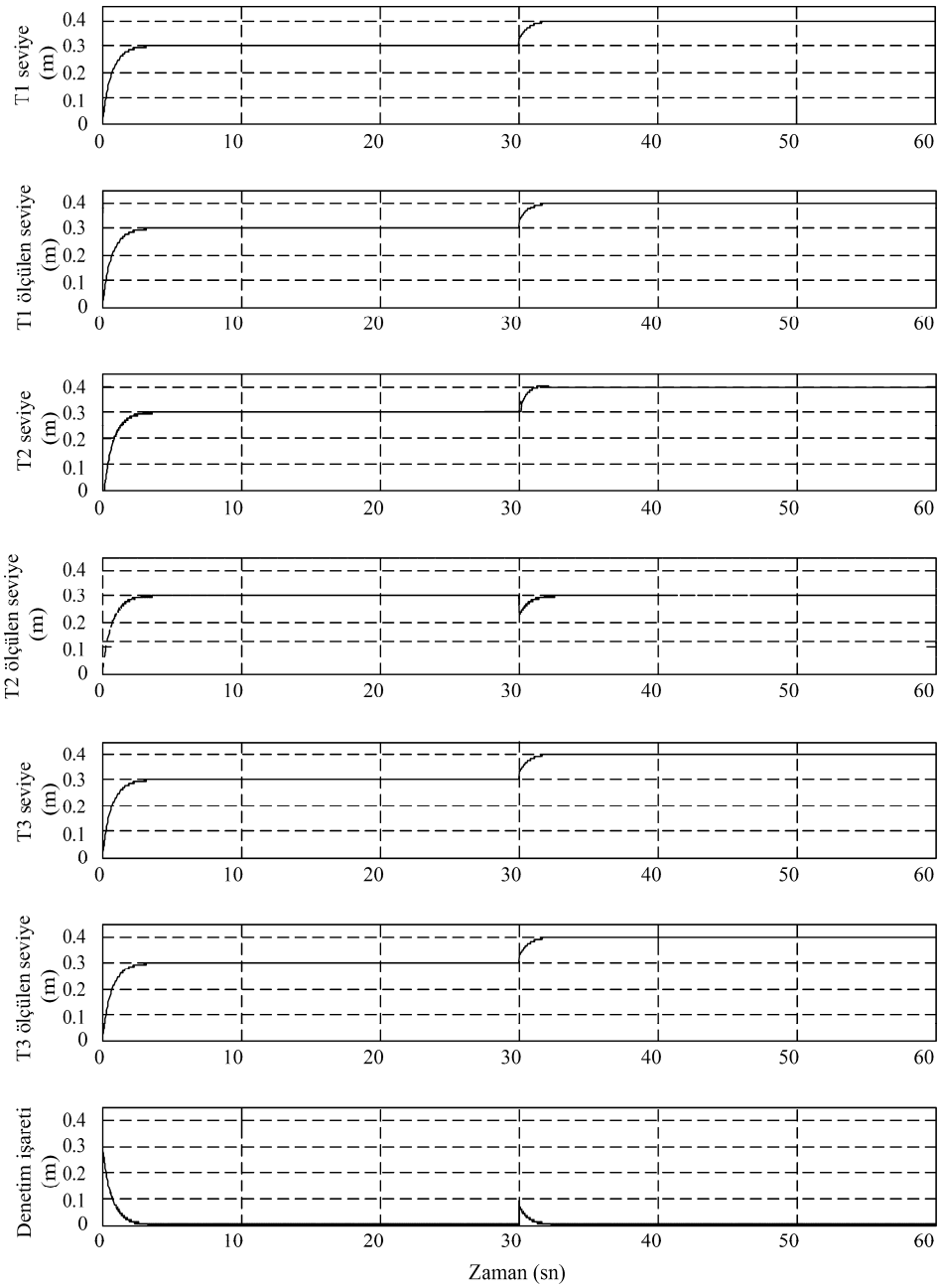
Tablo 4.1: Algılayıcı arızaları ve özellikleri

Durum No	**Arıza Tipi**
N	Arızasız, normal çalışma koşulu
A1	tank-2 seviye algılayıcısı için toplamsal arıza (-0.1)
A2	tank-3 seviye algılayıcısı için çarpımsal arıza (x1.1)
A3	tank-2 seviye algılayıcısı için çarpımsal arıza (x0.8)
A4	tank-1 seviye algılayıcısı için çarpımsal arıza (x0.7)
A5	tank-3 seviye algılayıcısı için toplamsal arıza (-0.1)
A6	tank-1 seviye algılayıcısı için tıkama tipinde arıza (0.4)
A7	tank-2 seviye algılayıcısı için toplamsal arıza (+0.05)

Tablo 4.1'deki yapay algılayıcı arızaları birer birer üçlü tank sistemine uygulanmış; her bir arıza durumunda tank-1, tank-2 ve tank-3'ün gerçek seviyeleri, seviye algılayıcısı vasıtasıyla ölçülen seviyeleri ve PID denetleyicisinin ürettiği denetim işareti sistemin geçmiş verileri (historical-base) olarak sistemden alınmış ve analizleri yapılmıştır.

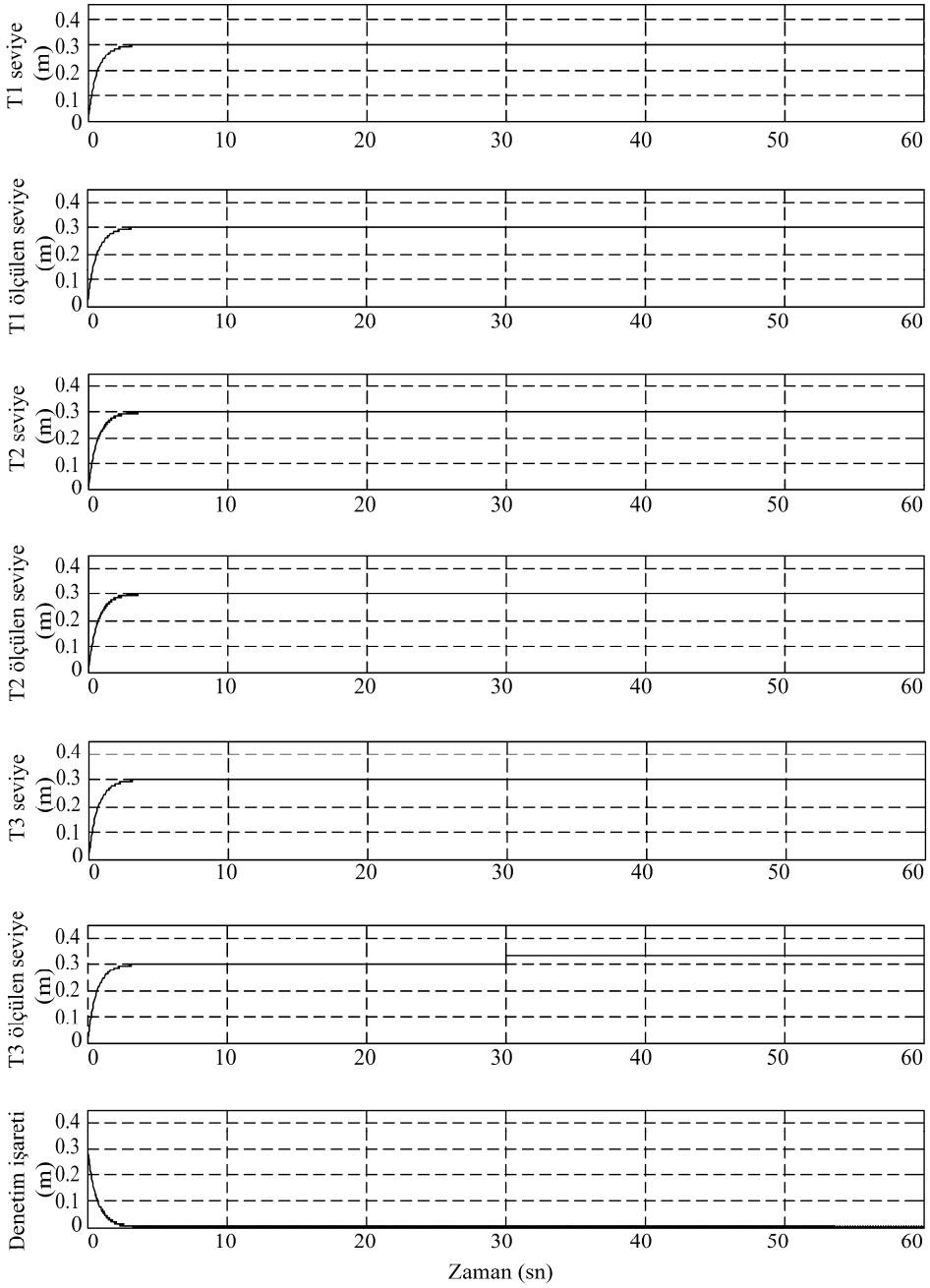
Tank-2 seviye algılayıcısına 30. saniyede, A1 numaralı toplamsal -0.1'lik bir arıza uygulandığında, tank-2 gerçek seviyesinde herhangi bir değişme olmamasına rağmen, PID denetleyicisi algılayıcı arızasını seviye azalması olarak algılayıp hemen

devreye girerek, tank-2 seviyesini yeniden referans seviyesi olan 0.3'e getirmek ister. PID denetleyicisinin tüm sistemi etkilemesi sonucunda tank-1 ve tank-3 gerçek ve ölçülen seviyeleri, tank-2 gerçek seviyesi 0.1 artar. Arızanın uygulanması sonucunda elde edilen tank-1, tank-2, tank-3 gerçek ve ölçülen seviyeleri ile arızanın denetim işaretine olan etkisi Şekil 4.4'de gösterilmektedir. A1 numaralı arızanın Matlab-Simulink uygulama modeli Şekil A.5'de gösterilmektedir.



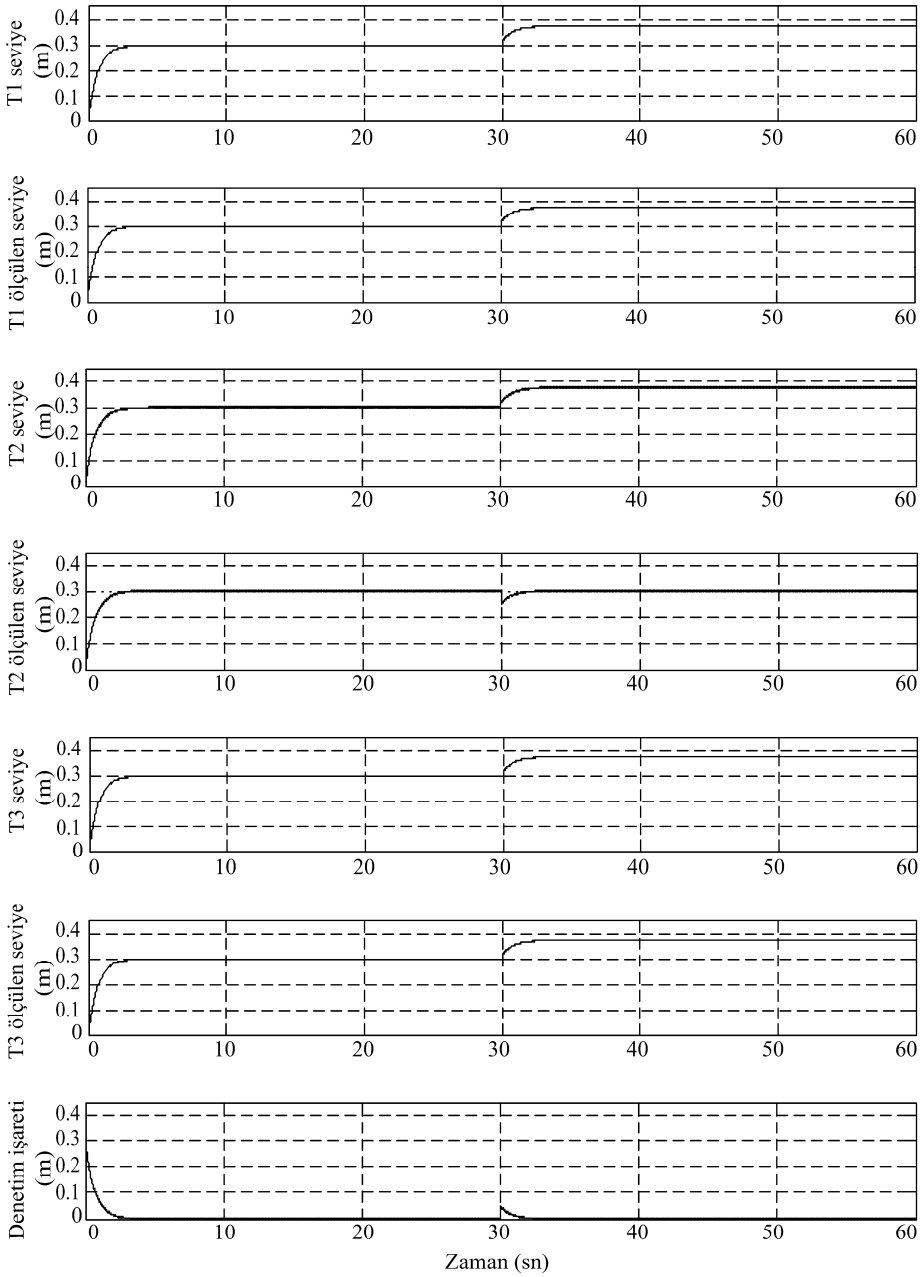
Şekil 4.4: A1 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi

Tank-3 seviye algılayıcısına 30. saniyede, A2 numaralı çarpımsal 1.1'lik bir arıza uygulandığında, tank-2 ölçülen seviyesinde herhangi bir değişme olmadığı için, PID denetleyicisi devreye girmez. Bunun sunucunda sadece tank-3 seviyesi yanlış gösterilmiş olur, tankların seviyelerinde herhangi bir değişiklik meydana gelmez. Arızanın uygulanması sonucunda elde edilen tank-1, tank-2, tank-3 gerçek ve ölçülen seviyeleri ile arızanın denetim işaretine olan etkisi Şekil 4.5'te gösterilmektedir. A2 numaralı arızanın Matlab-Simulink uygulama modeli Şekil A.6'da gösterilmektedir.



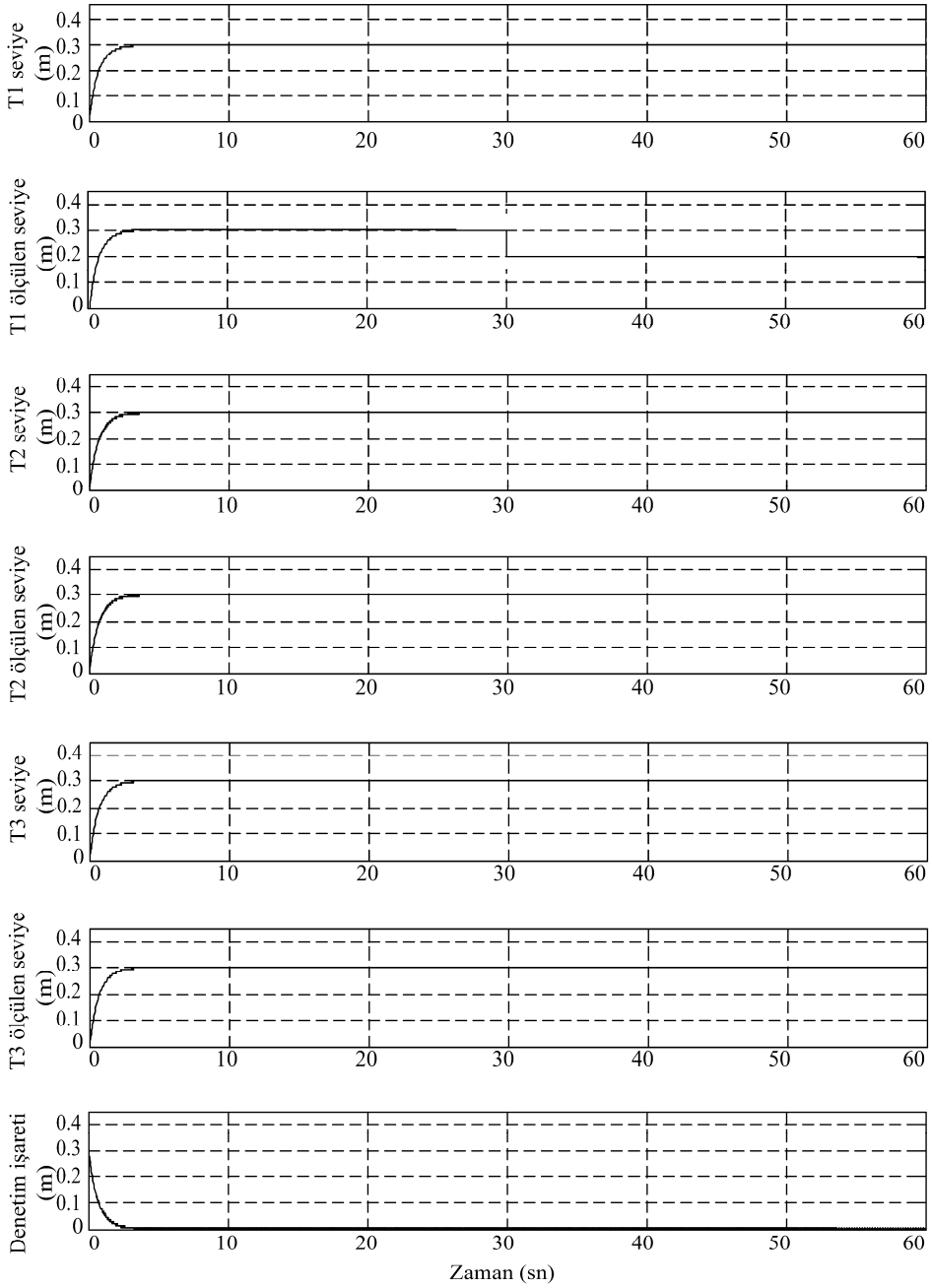
Şekil 4.5: A2 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi

Tank-2 seviye algılayıcısına 30. saniyede, A3 numaralı çarpımsal 0.8'lik bir arıza uygulandığında, tank-2 gerçek seviyesinde herhangi bir değişme olmamasına rağmen, PID denetleyicisi tank-2 seviyesini 0.24'e düşmüş olarak algılar ve yeniden referans seviyesi olan 0.3'e getirmek ister. PID denetleyicisinin tüm sistemi etkilemesi sonucunda tank-1 ve tank-3 gerçek ve ölçülen seviyeleri, tank-2 gerçek seviyesi 0.06 artar. Arızanın uygulanması sonucunda elde edilen tank-1, tank-2, tank-3 gerçek ve ölçülen seviyeleri ile arızanın denetim işaretine olan etkisi Şekil 4.6'da, arızanın Matlab-Simulink uygulama modeli Şekil A.7'de gösterilmektedir.



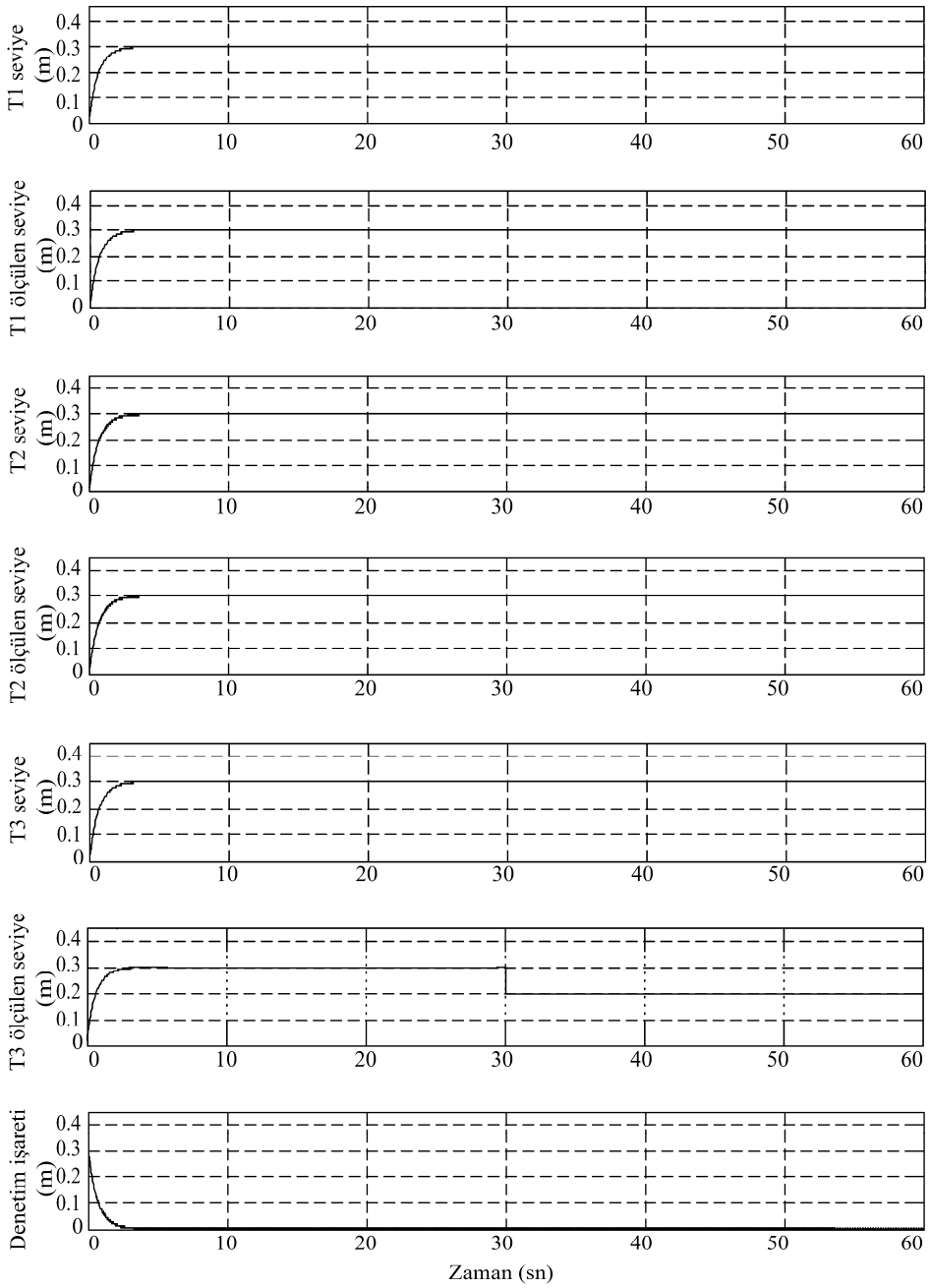
Şekil 4.6: A3 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi

Tank-1 seviye algılayıcısına 30. saniyede, A4 numaralı çarpımsal 0.7'lik bir arıza uygulandığında, tank-2 ölçülen seviyesinde herhangi bir değişme olmadığı için, PID denetleyicisi devreye girmez. Bunun sunucunda sadece tank-1 seviyesi yanlış gösterilmiş olur, tankların seviyelerinde herhangi bir değişiklik meydana gelmez. Arızanın uygulanması sonucunda elde edilen tank-1, tank-2, tank-3 gerçek ve ölçülen seviyeleri ile arızanın denetim işaretine olan etkisi Şekil 4.7'de gösterilmektedir. A4 numaralı arızanın Matlab-Simulink uygulama modeli Şekil A.8'de gösterilmektedir.



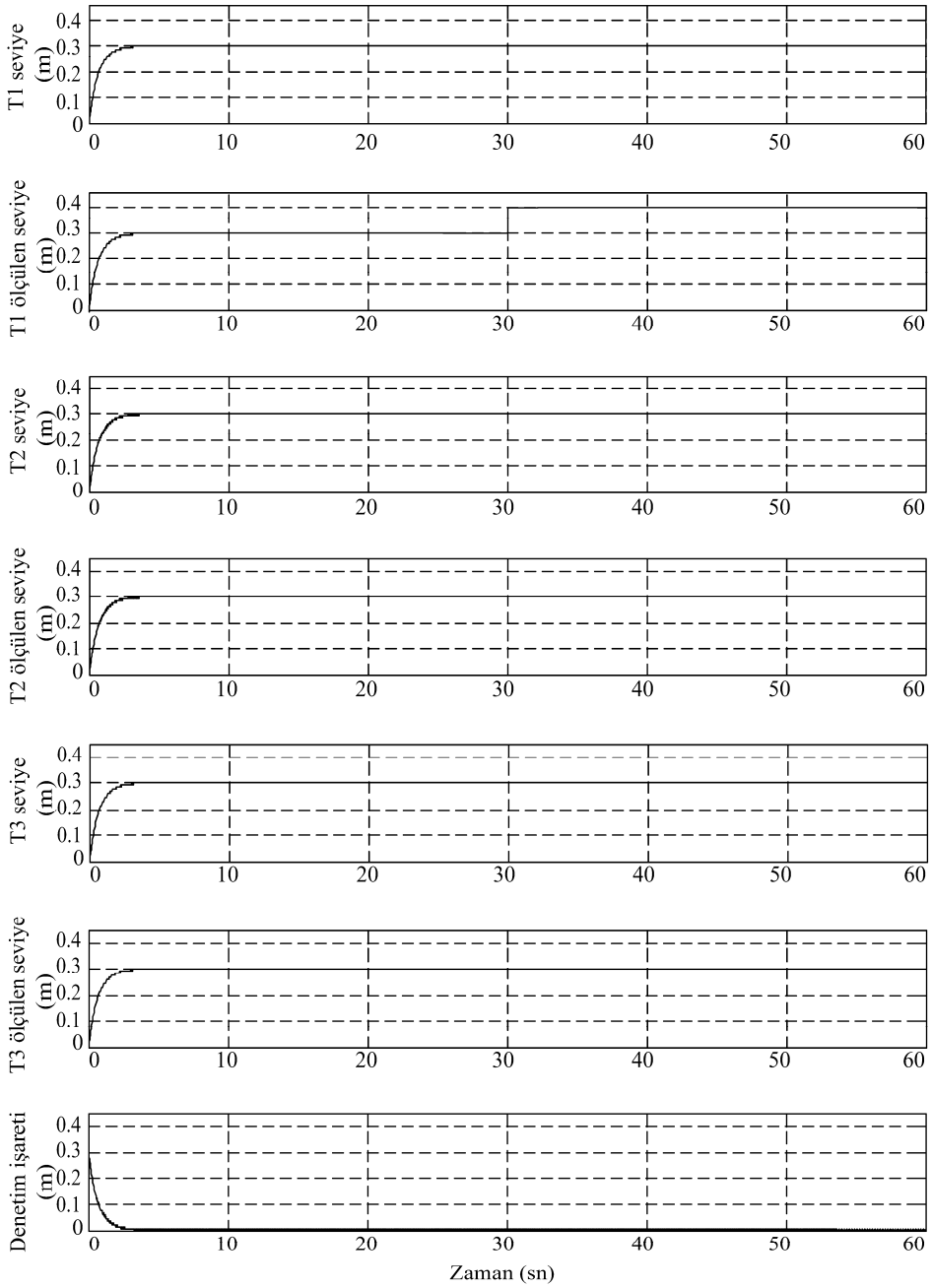
Şekil 4.7: A4 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi

Tank-3 seviye algılayıcısına 30. saniyede, A5 numaralı toplamsal -0.1'lik bir arıza uygulandığında, tank-2 ölçülen seviyesinde herhangi bir değişme olmadığı için, PID denetleyicisi devreye girmez. Bunun sunucunda sadece tank-3 seviyesi yanlış gösterilmiş olur, tankların seviyelerinde herhangi bir değişiklik meydana gelmez. Arızanın uygulanması sonucunda elde edilen tank-1, tank-2, tank-3 gerçek ve ölçülen seviyeleri ile arızanın denetim işaretine olan etkisi Şekil 4.8'de gösterilmektedir. A5 numaralı arızanın Matlab-Simulink uygulama modeli Şekil A.9'da gösterilmektedir.



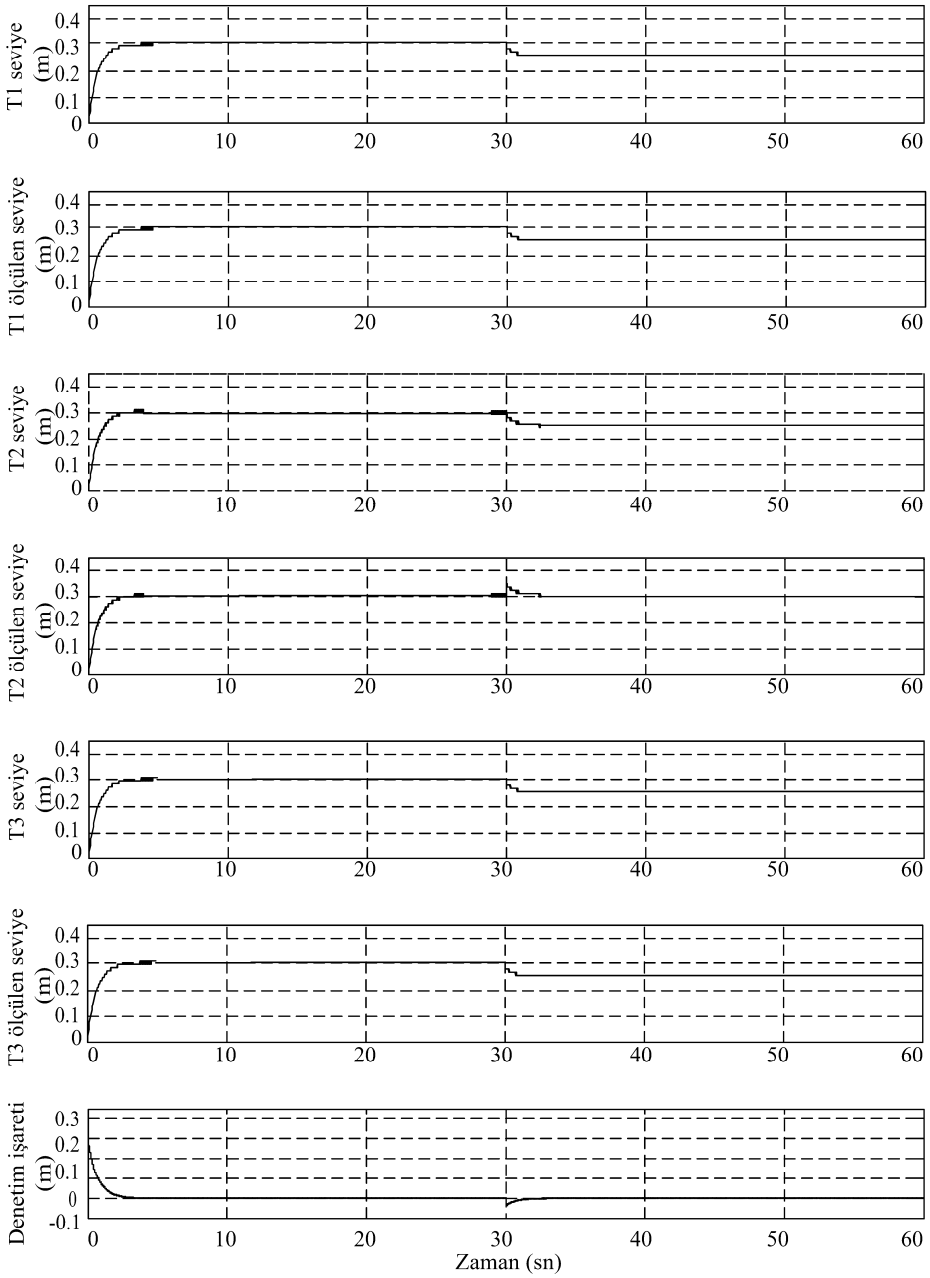
Şekil 4.8: A5 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi

Tank-1 seviye algılayıcısına 30. saniyede, A6 numaralı tıkama tipi 0.4'lük bir arıza uygulandığında, tank-2 ölçülen seviyesinde herhangi bir değişme olmadığı için, PID denetleyicisi devreye girmez. Bunun sunucunda sadece tank-1 seviyesi yanlış gösterilmiş olur, tankların seviyelerinde herhangi bir değişiklik meydana gelmez. Arızanın uygulanması sonucunda elde edilen tank-1, tank-2, tank-3 gerçek ve ölçülen seviyeleri ile arızanın denetim işaretine olan etkisi Şekil 4.9'da gösterilmektedir. A6 numaralı arızanın Matlab-Simulink uygulama modeli Şekil A.10'da gösterilmektedir.



Şekil 4.9: A6 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi

Tank-2 seviye algılayıcısına 30. saniyede, A7 numaralı toplamsal +0.05'lik bir arıza uygulandığında, tank-2 gerçek seviyesinde herhangi bir değişme olmamasına rağmen, PID denetleyicisi tank-2 seviyesini 0.35'e yükselmiş olarak algılar ve yeniden referans seviyesi olan 0.3'e getirmek ister. PID denetleyicisinin tüm sistemi etkilemesi sonucunda tank-1 ve tank-3 gerçek ve ölçülen seviyeleri, tank-2 gerçek seviyesi 0.05 azalır. Arızanın uygulanması sonucunda elde edilen tank-1, tank-2, tank-3 gerçek ve ölçülen seviyeleri ile arızanın denetim işaretine olan etkisi Şekil 4.10'da, arızanın Matlab-Simulink uygulama modeli Şekil A.11'de gösterilmektedir.



Şekil 4.10: A7 arızasının gerçek seviyelere, ölçülen seviyelere ve denetim işaretine etkisi

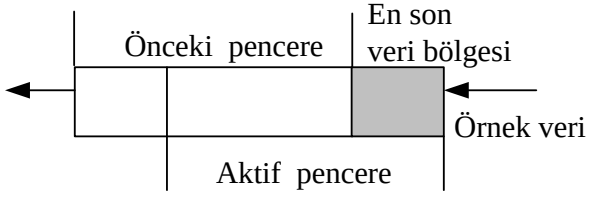
4.3. Algılayıcı Arızalarının İstatistiksel Analizi

Arızalar, endüstriyel sistemler ve süreçler üzerinde sistemin normal olmayan cevaplar vermesine neden olurlar. Öte yandan, normal sistem uygulamalarında referans girişin değiştirilmesi de aynı değişikliklere neden olur. Normal ve arızalı durumlarda, sistemin ürettiği verilerin çevrim dışı olarak analizi, arıza durumlarını ayırt etmemizde kullanabileceğimiz performans ölçütlerinden biridir.

Sistemden elde edilmiş verilerin işlenmesi ile sinyal üzerinde meydana gelen değişiklikler gözlenebilmekte, bu amaçla sistemin geçmişine yönelik tarihsel veriler kullanılarak özellik çıkartımı yapılmaktadır.

4.3.1. Verilerin pencerelenmesi

Özellik çıkartımının gerçekleştirilmesinde ilk adım olarak kayan pencere yöntemi kullanılmaktadır. Kayan pencere yöntemi ile olası hataya bağlı geçici sinyal değişimleri tespit edilmektedir. Bu yöntemde, sinyal örnekleri önceden belirlenmiş boyutta pencerelere ayrılır ve pencere içerisindeki sinyalde meydana gelen değişiklikler gözlemlenir. Bu işlem bir süreç içerisinde gerçekleştirilir iken, zaman ilerledikçe her bir pencere önceden belirlenmiş boyutta kaydırılır ve yeni pencere ile eski pencere içerisindeki özellikler karşılaştırılarak arızalı durumlar tespit edilir. Kayan pencere yönteminin çevrimiçi ve gerçek zamanlı uygulamalardaki kullanımı Şekil 4.11'de gösterilmiştir.

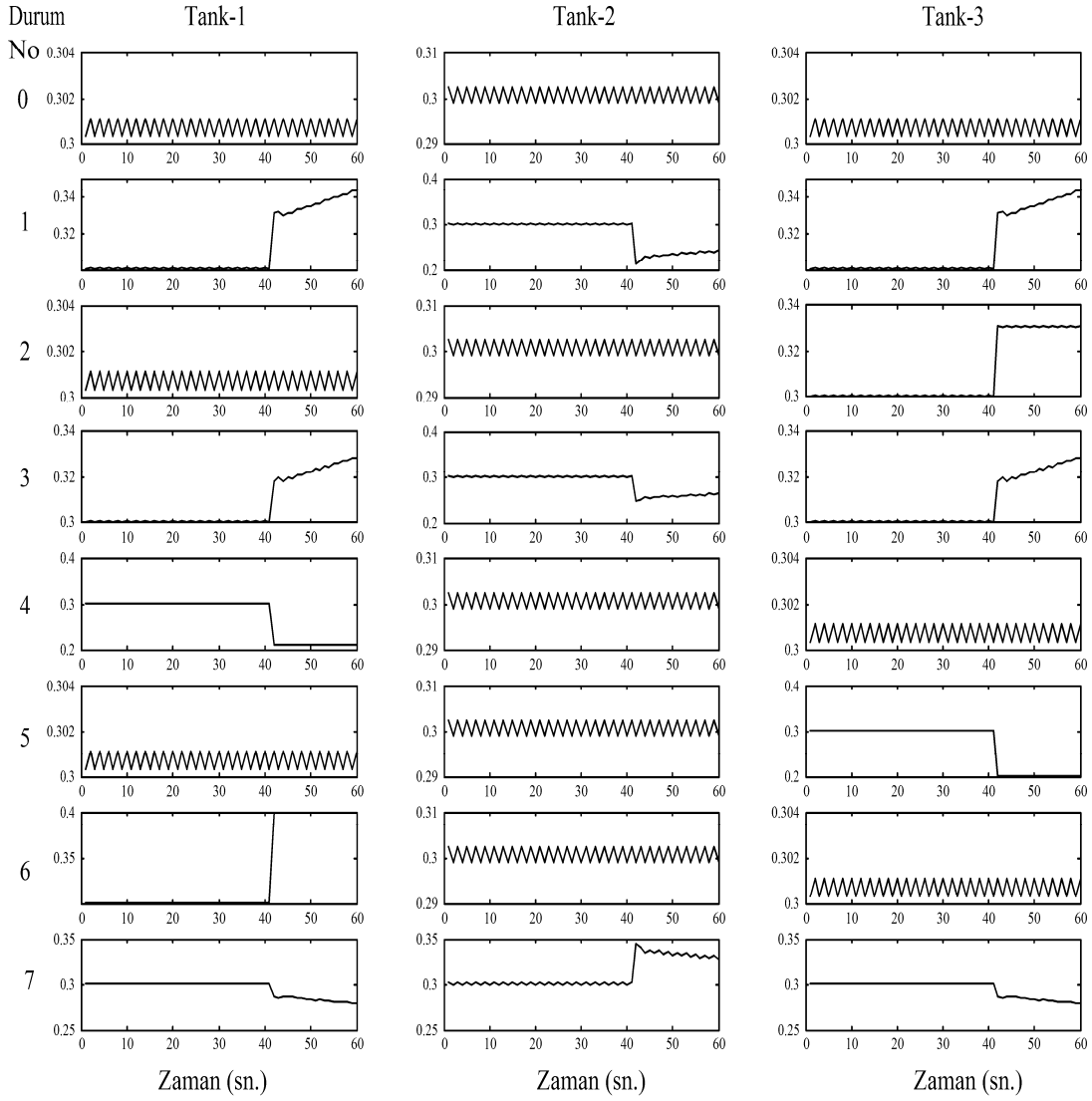


Şekil 4.11: Kayan pencere tekniği

Şekilde görüldüğü gibi, aynı uzunlukta ve birbiri ile örtüşen iki adet veri penceresi bulunmaktadır. Bunlar, önceki pencere ve aktif pencere olarak isimlendirilmişlerdir. Önceki pencerede geçmiş zamanda elde edilmiş olan veriler bulunurken, aktif pencerede son zamanda elde edilen veriler bulunmaktadır. Kayan pencere

yönteminde, sinyalin en son örneklerinde meydana gelen değişiklikleri içeren en güncel bilgiler, aktif penceredeki en son veri bölgesinde barınmaktadır [39].

Tablo 4.1'teki tüm arıza durumları için benzetim çalışması sonuçlarında elde edilen veriler pencerelenerek gözlemleme işlemi gerçekleştirilmiştir. Pencereleme safhasında, pencerelenen işaretin genişliği 60, son veri bölgesi genişliği ise 20 örnek olarak alınmıştır. Pencereleme işlemi gerçekleştirilirken, her bir pencere son veri bölgesi genişliği kadar kaydırılmaktadır. Şekil 4.12'de normal ve tüm arıza durumları için arızanın görüldüğü pencereler gösterilmektedir.



Şekil 4.12: Tanklara ait seviye algılayıcılarından alınan işaretlerin pencerelenmesi

4.3.2. Özellik çıkartımı

Özellik çıkartımında, kayan pencere yöntemi ile sinyal pencerelere ayrıldıktan sonra, pencere içerisindeki verilerin ortalamadan ne kadar uzaklaştığı varyans ile hesaplanır. Varyans kavramı dağılıma ait her bir değerin dağılımın ortalamasından ne kadar uzak olduğu ile ilgilidir. Her iki pencere için de varyansların belirlenmesinin ardından, varyansların değişim oranı (*r*) Denklem (4.7)'de gösterildiği gibi hesaplanarak özellik çıkartımı gerçekleştirilmiş olunur. Denklemde, bölme işleminin sonsuza gitmemesi için ε ile ifade edilen küçük bir pozitif sayı denkleme eklenmektedir.

$$r = \frac{\text{var}(\textbf{\textit{ Aktif pencere }}) \; - \text{var}(\textbf{\textit{ Önceki pencere }})}{\text{var}(\textbf{\textit{ Önceki pencere }}) + \varepsilon} \tag{4.7}$$

Olasılık kuramı ve istatistik bilim dallarında varyans bir rastlantısal değişken, bir olasılık dağılımı veya örneklem için istatistiksel yayılımın mümkün bütün değerlerin beklenen değer veya ortalamadan uzaklıklarının karelerinin ortalaması şeklinde bulunan bir ölçüdür. Varyans için ölçülme birimi orijinal değişkenin biriminin karesidir. Varyansın karekökü standart sapma olarak adlandırılır; bunun ölçme birimi orijinal değişkenle aynı birimde olur ve bu nedenle daha kolayca yorumlanabilir.

Eğer beklenen değer var ise, bir olasılık dağılımı için varyans, dağılımın kendi ortalamasından sapmasının karesinin beklenen değeridir. Varyans kavramı dağılıma ait her bir değerin dağılımın ortalamasından ne kadar uzak olduğuyla ilgilidir. Varyans söz konusu sapmaların ortalama değerini ölçmektedir.

X değişkeninin beklenen değeri μ = E(X) olmak üzere, varyans Denklem (4.8)'de gösterildiği gibi tanımlanır:

$$\text{var}(X) = E((X - \mu)^2) \tag{4.8}$$

Matematik notasyon kullanılarak bir rastlantısal değişken *X* için varyans ya var(*X*) ya da σ_X^2 veya daha basitçe σ^2 olarak gösterilir. Bu tanımlama, eğer beklenen değer varsa, hem ayrık rastlantısal değişkenler hem sürekli rastlantısal değişkenler hem de

karmaşık değişkenler için genel olarak doğrudur. Bu tanımdan ve beklenen değerlerin doğrusal olma niteliğinden varyans için Denklem (4.9)'da gösterildiği gibi formül çıkartılabilir:

$$var(X) = E(X^2 - 2X.E(X) + (E(X))^2)$$

$$= E(X^2) - 2(E(X))^2 + (E(X))^2,$$

$$= E(X^2) - (E(X))^2 \qquad (4.9)$$

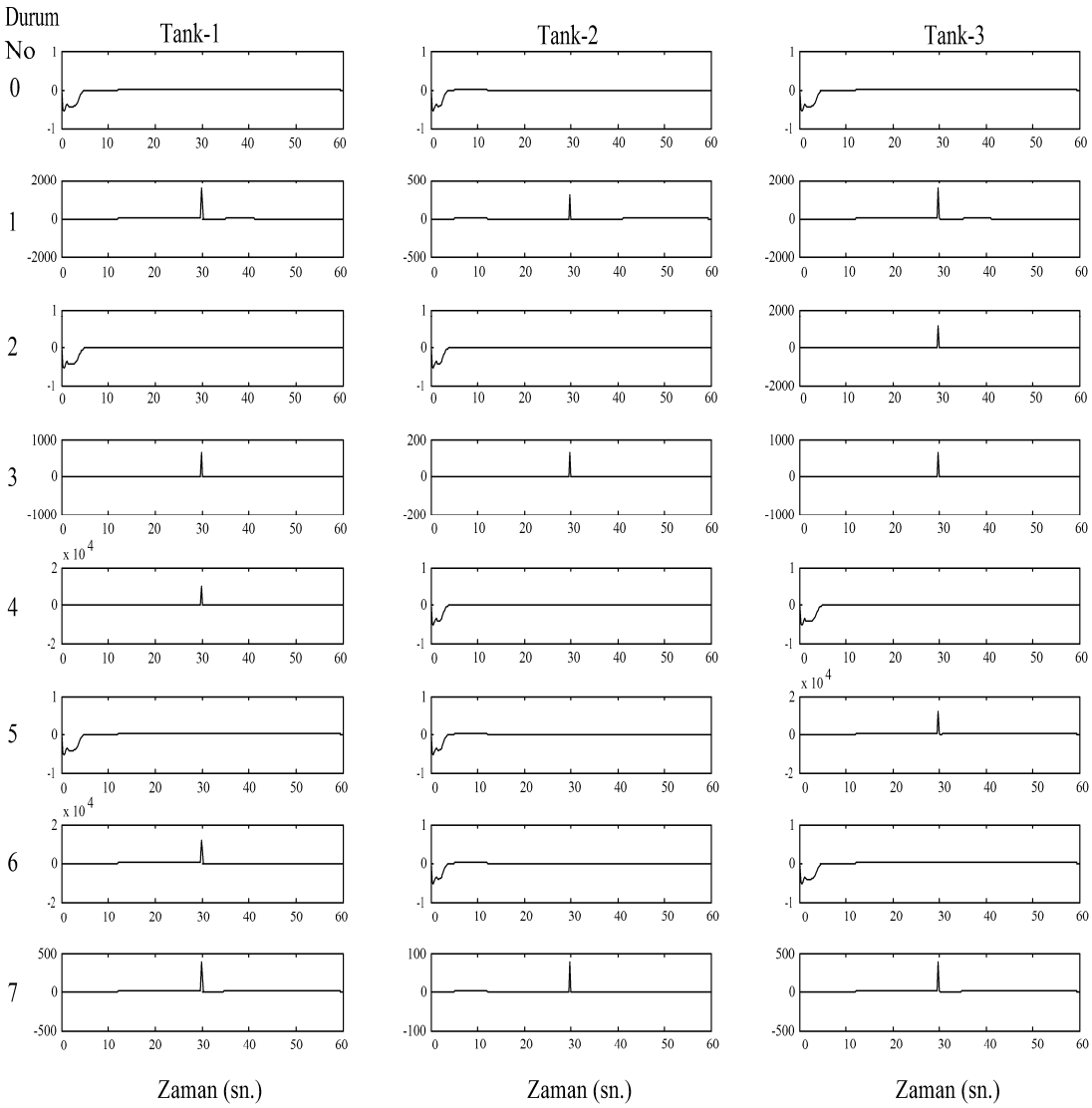
Buna hesaplama formülü adı da verilir. Bu formüle göre varyans, karelerin ortalaması eksi ortalamanın karesine eşittir. Varyans verilerin aritmetik ortalamadan sapmalarının karelerinin aritmetik ortalaması olduğuna göre, {2, 2, 3, 5, 3} serisinin varyansı şu şekilde bulunur;

Verilerin aritmetik ortalaması $A.O. = \frac{2+2+3+5+3}{5} = 3$ olarak hesaplanır. Aritmetik ortalamadan, verilerin sapmalarının karelerinin toplamının, toplam veri sayısının bir eksiğine bölünmesi ile serinin varyansı bulunur;

$$\frac{[(2-3)^2 + (2-3)^2 + (3-3)^2 + (5-3)^2 + (3-3)^2]}{4} = \frac{6}{4}$$

Tez çalışmasında, Tablo 4.1'teki tüm arıza durumları için pencerelemeler yapıldıktan sonra, her bir pencere için varyansların değişim oranı (*r*) hesaplanarak, özellik çıkartımı gerçekleştirilmiştir. Bu aşamada hesaplanan varyans değişim oranları Şekil 4.13'te gösterilmektedir.

Şekil 4.13'te gösterildiği gibi, tüm pencerelerin varyans değişim oranlarına bakarak hangi pencerede arızanın meydana geldiği tespit edilebilmektedir. Tank-2 arızalarında tüm tankların pencere varyanslarında değişiklik meydana gelirken, tank-1 ve tank-3 arızalarında ise sadece ilgili tankların pencere varyanslarında değişiklik meydana gelmektedir. Varyans değişimi kullanılarak sistem üzerinde oluşabilecek gürültülerde ortadan kaldırılmaktadır [77].



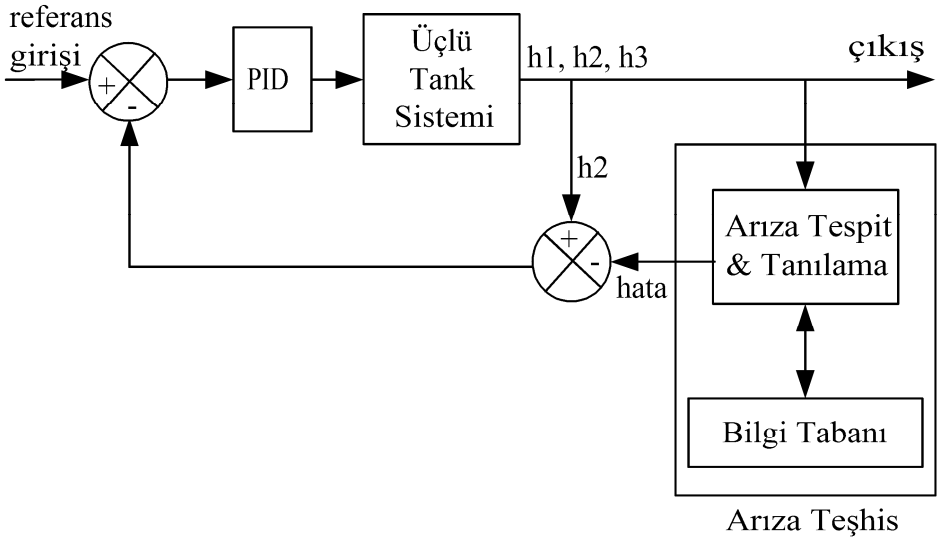
Şekil 4.13: Varyans değişim oranları

4.4. Arıza Dayanımlı Denetim Sistemi Tasarımı

ADD sistemi, ATT işlemi ile sistemin yeniden yapılandırılmasını içermektedir. ATT ile sistemde bir arıza olup olmadığı, bir arıza söz konusu ise arızanın ne tür bir arıza olduğu, nerede ve ne zaman oluştuğu belirlenebilir.

ADD'nin temel amacı, tüm algılayıcı, eyleyici ve bileşen arızalarına rağmen sistemin kararlı ve tatmin edici bir başarımla çalışmasını sürdürmesi için uygun bir denetleyicinin tasarlanmasıdır. Bu tez çalışmasında kullanılan ADD sisteminin genel yapısı Şekil 4.14'te gösterilmektedir.

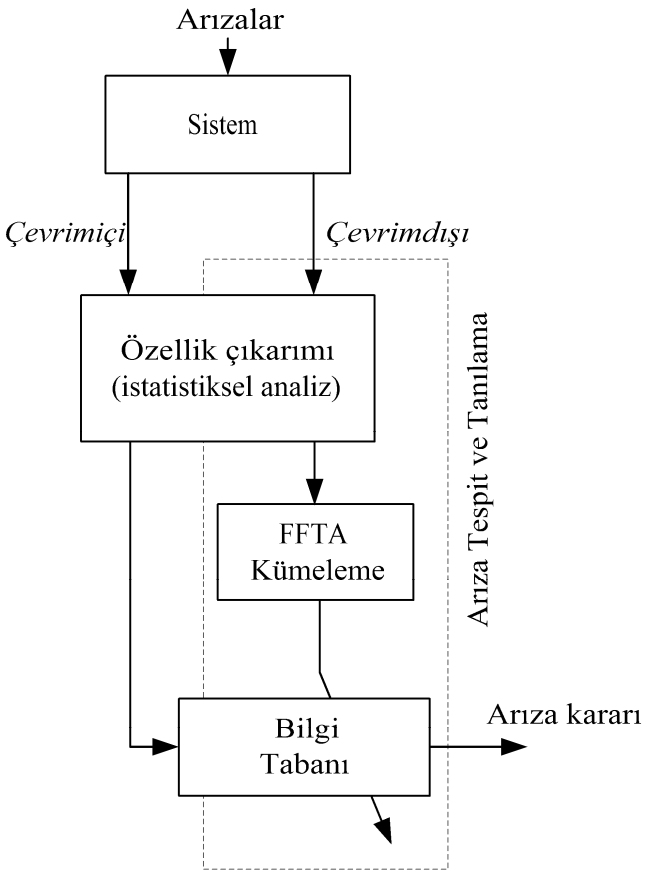
Şekil 4.14'te gösterilmekte olan kapalı çevrim ADD sistemi PID denetleyici, üçlü tank sistemi ve arıza teşhis bölümlerinden oluşmaktadır. Üçlü tank sistemindeki tank seviye algılayıcılarından elde edilen h_1, h_2 ve h_3 işaretleri ATT bölümünde işlenmektedir.



Şekil 4.14: Tasarlanan kapalı çevrim arıza dayanımlı denetim sistemi.

Tez çalışmasında kullanılan arıza teşhis sisteminin genel yapısı Şekil 4.15'te gösterilmektedir. Arıza teşhis bölümü, ATT ve bilgi tabanı kısımlarından oluşmaktadır. ATT da kendi içerisinde, istatistiksel analiz ve kümeleme olmak üzere iki aşamaya ayrılmaktadır. Geçmiş sistem verileri üzerinde istatistiksel teknikler kullanılarak sistemin normal durumlardan sapması tespit edilmektedir. Bu bölümde; kayan pencere yöntemi ile sinyal pencerelere ayrıldıktan sonra, pencere içerisindeki verilerin ortalamadan ne kadar uzaklaştığı varyans değişim oranı ile hesaplanarak özellik çıkartımı işlemi gerçekleştirilmektedir.

Sistem üzerinde arıza tespit edildikten sonra, arızanın tanılanma ve büyüklüğünü belirleme işlevi gerçekleştirilmektedir. ATT tarafından çevrimdışı olarak elde edilmiş tank verileri EUEÖTA tarafından işlenerek bilgi tabanı oluşturulmaktadır. Burada, EUEÖTA'nın kullanım amacı, normal ve arızalı geçmiş tank verilerinin kümelendirilmesidir. Sistemin çalışmasının normal veya arızalı olduğu verilerin farklı kümelere ayrılması ile ortaya çıkarılmaktadır.



Şekil 4.15: Tasarlanan ATT sistemi genel yapısı.

EUEÖTA ile kümeleme aşamasında, her bir tank seviyesi için varyans değişim oranları göz önünde tutularak kümeleme işlemi gerçekleştirilir. Önceden tanımlanmış tüm arızalar için, tank seviyelerine göre belirlenen özellik vektörleri bilgi tabanına küme merkezleri olarak kaydedilmektedir.

ATT sistemi çevrimiçi çalışırken tank seviyelerinden gelen verilerin özellik çıkarımı ile elde edilen varyans değerleri, bilgi tabanındaki veriler ile Euclidean uzaklığı hesaplanarak karşılaştırır. Böylece sistemin çevrimiçi çalışırken normal veya arızalı olduğu kararı verilir. Eğer arıza tespit edilirse, arızanın hangi kümeye ait olduğu, bilgi tabanında yer alan arıza küme merkezlerine olan uzaklıklara göre karar verilir.

Arızanın tipinin belirlenmesi işine arızanın tanılanması veya izolasyonu adı verilmektedir. Arıza bir kez tanılandıktan sonra, o kümedeki arızaları gidermek için önceden belirlenmiş denetim değişikliği veya düzenlemesi işleme sokulur. Bu aşamada genellikle anahtarlamalı denetim kullanılmaktadır.

ATT sisteminde hem çevrimiçi hem de çevrimdışı olarak gerçekleştirilen özellik çıkartımı, sisteminin genel performansı üzerinde çok önemli bir role ve etkiye sahiptir. Özellik çıkartımı kullanılarak normal ve arızalı sistem durumlarının özellik vektörleri oluşturulur. Özellik vektörleri çevrimdışı süreçte geçmiş sistem verilerinin kullanılması ile elde edilirken, çevrimiçinde ise gerçek zamanlı ölçümlerden elde edildiği dikkat edilmesi gereken önemli bir husustur.

4.4.1. Arızaların EUEÖTA ile çevrimdışı olarak kümelenmesi

Tez çalışmasında, üçlü tank sistemi algılayıcı arızaları hiyerarşik bir kümeleme yöntemi olan EUEÖTA algoritması ile kümelenmiştir. Kümeleme işlemi, hem çevrimdışı yapıldığı ve hız gerektirmediği için hem de görsel özelliklerinden de yararlanabilmek için, bir veri madenciliği yazılımı olan WEKA programı ile yapılmıştır.

WEKA, 1993 yılından beri Yeni Zelanda hükümeti tarafından finanse edilmektedir ve bilgi tabanlı bir makine öğrenmesi benzetim projesidir. WEKA yazılımı ile veri madenciliğinde yer alan yöntemlerin birçoğu kullanılarak veri analizleri gerçekleştirmek mümkündür. WEKA'da verilerin ön işlenmesi aşamasında, yerel bir dosya biçimi olan ARFF kullanılmakla birlikte; CSV, Matlab, ASCII biçimleri de desteklenmektedir. Veri filtrelemesi için yetmiş beşin üzerinde desteklenen yöntem bulunmaktadır.

Verilerin analizi aşamasında; sınıflandırma, regresyon, kümeleme ve birliktelik kuralları madenciliği yapmak amacıyla birçok algoritma kullanılabilmektedir. Veriler, yazılımın veri görselleştirme bölümünde, dahil oldukları sınıf, küme veya diğer özniteliklerine göre görsel olarak incelenebilmektedir.

WEKA programı; Explorer (araştırma modülü), Experimenter (deney modülü), KnowledgeFlow (bilgi akış modülü) ve Simple CLI (basit komut satırı arayüzü) alt bölümlerinden oluşmaktadır. WEKA programının genel bileşenlerinin gösterildiği arayüz yazılımı Şekil 4.16'da gösterilmektedir.



Şekil 4.16: WEKA yazılımının genel bileşenleri

Tez çalışmasında, WEKA Explorer arayüzünde EUEÖTA kümeleme yöntemi seçilerek, sistemden alınan geçmiş veriler kümelere ayrıştırılmış ve küme merkezleri bulunmuştur.

Tablo 4.1 ile özellikleri ve durum numaraları verilen algılayıcı arızalarının küme merkezleri de Tablo 4.2'de verilmektedir.

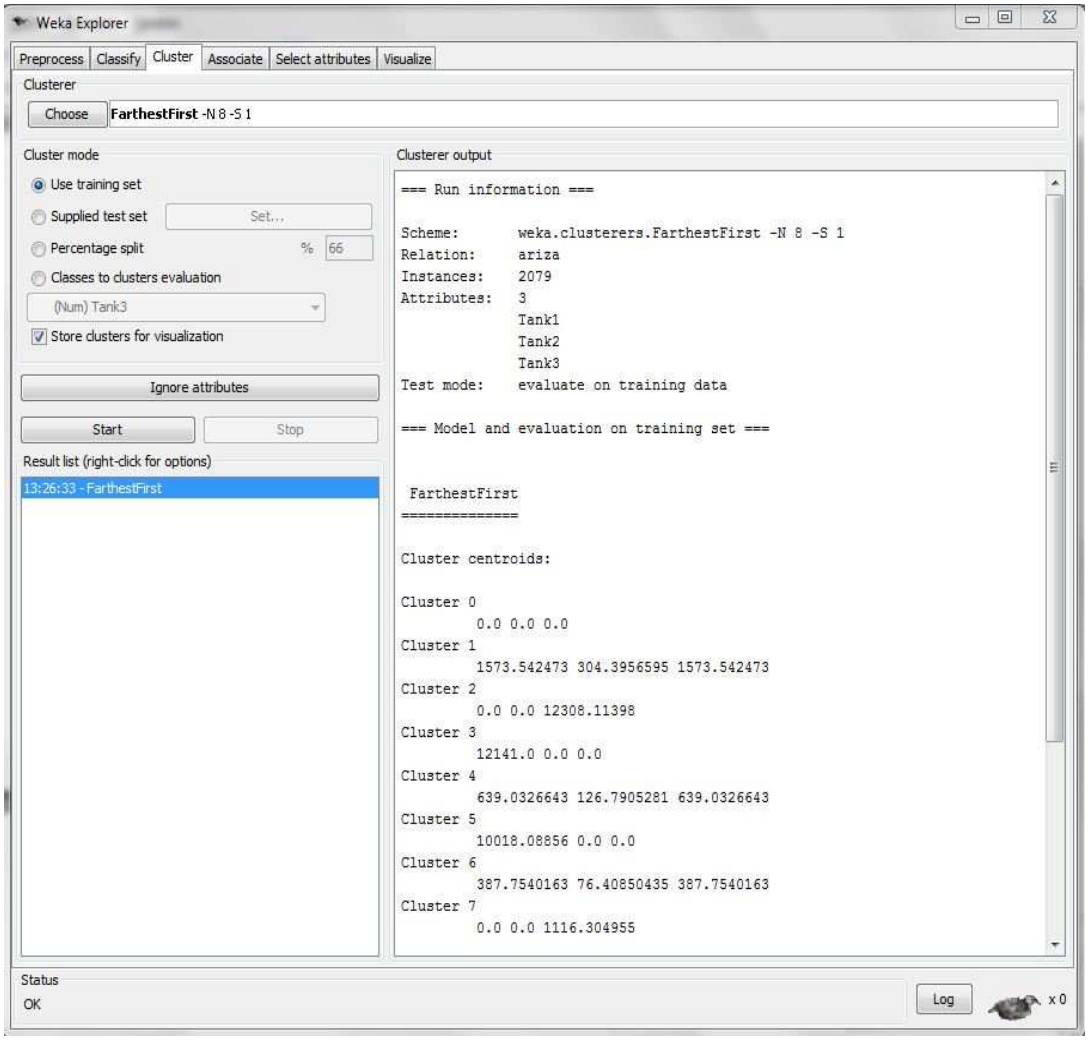
Tablo 4.2: Arıza küme merkezleri koordinatları

Durum	**Küme merkezleri koordinatları**		
	tank-1	**tank-2**	**tank-3**
N	0	0	0
A1	1573.54	304.39	1573.54
A2	0	0	1116.30
A3	639.03	126.79	639.03
A4	10018.08	0	0
A5	0	0	12308.11
A6	12141	0	0
A7	387.75	76.40	387.75

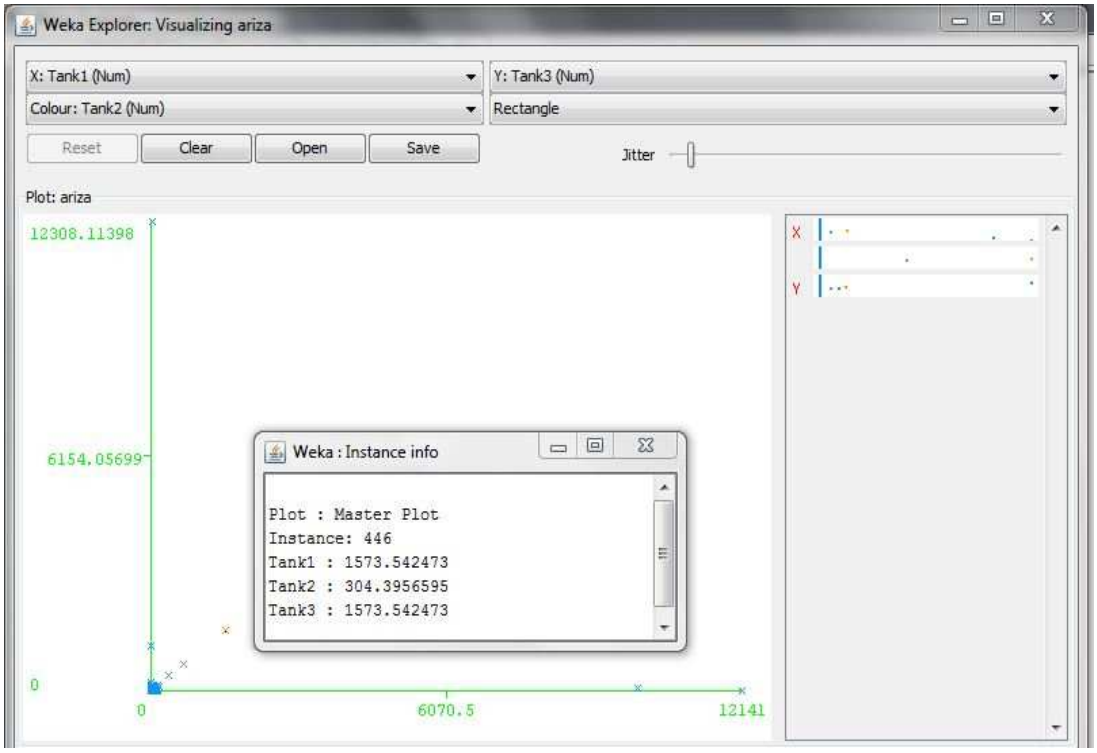
Tablo 4.2'de tasarlanan ADD sisteminin bilgi tabanını oluşturmaktadır ve her bir satırında bir arıza özellik vektörü içermektedir. "N" etiketli durum, arızanın olmadığı Normal durumu işaret etmekte, algılayıcılardan alınan işaretlerde herhangi bir sapma olmadığı için de "0" değerlerini almaktadır.

Şekil 4.17'de, Tablo 4.2'de gösterilmekte olan arıza küme merkezlerinin koordinatlarının elde edildiği WEKA Explorer arayüzü görüntüsü ve EUEÖTA ile elde edilen sekiz adet küme merkezi bilgileri verilmektedir.

WEKA Explorer arayüzünde, bilgi tabanından gelen tüm verilerin işlenmesi sonucunda elde edilmiş olan küme merkezleri, Şekil 4.18'deki gibi görselleştirilebilmektedir. Sistemden alınan ölçüm işaretleri sayısı üç olduğu için, şekildeki grafik, küme merkezlerini üç boyutlu uzayda göstermektedir.



Şekil 4.17: WEKA Explorer'da arıza küme merkezlerinin EUEÖTA ile bulunması



Şekil 4.18. WEKA'da arıza küme merkezlerinin grafik olarak gösterilmesi

4.4.2. Arızaların çevrimiçi tespit ve tanılanması

PID kullanılarak gerçekleştirilmiş olan üçlü tank sisteminin denetiminde, tüm arızaların sisteme uygulanabilmesi ve arızaların tespit edilmesi için oluşturulan Matlab-Simulink modeli genel olarak Şekil A.12'de, tanklardaki seviye algılayıcılarına uygulanacak arızanın seçimi için oluşturulan Matlab-Simulink modeli ise Şekil A.13'de gösterilmektedir.

Sisteme yeni bir arıza uygulanmak istendiğinde, tankın algılayıcılarından arızalı ölçümlerin alınabilmesi amacıyla, arıza seçim modelinden tank modellerine arıza verileri gönderilmektedir. Bu amaçla oluşturulan tank-1, tank-2 ve tank-3 arıza modelleri Şekil A.14, A.15 ve A.16'da sırası ile gösterilmektedir.

Sistem üzerinde istenilen arızanın referans değerinin, tank bilgisinin, zaman ve büyüklüğünün girilerek verilerin tankların arıza modellerine gönderildiği ve arıza sonuçlarının gözlemlendiği arıza seçim Matlab-Gui programı Şekil A.17'de gösterilmektedir.

Tasarlanan ADD ile üçlü tank sistemi benzetim modeline bir arıza uygulandığında, sistemin çalışması anında uygulanan arızanın tespit ve tanılanması yapılabilmektedir.

Sistemin çalışması süresince, tankların seviye algılayıcılarından elde edilen veriler gerçek zamanlı olarak pencerelenmektedir. Burada, her bir pencere için varyans değişim oranları hesaplanmaktadır. Sistemin çalışması anında hesaplanan varyans değişim oranları ile Tablo 4.2'de verilmiş olan sekiz adet küme merkezi arasındaki Euclidean uzaklıkları karşılaştırılarak, arıza tespit edilmekte ve tanılanmaktadır.

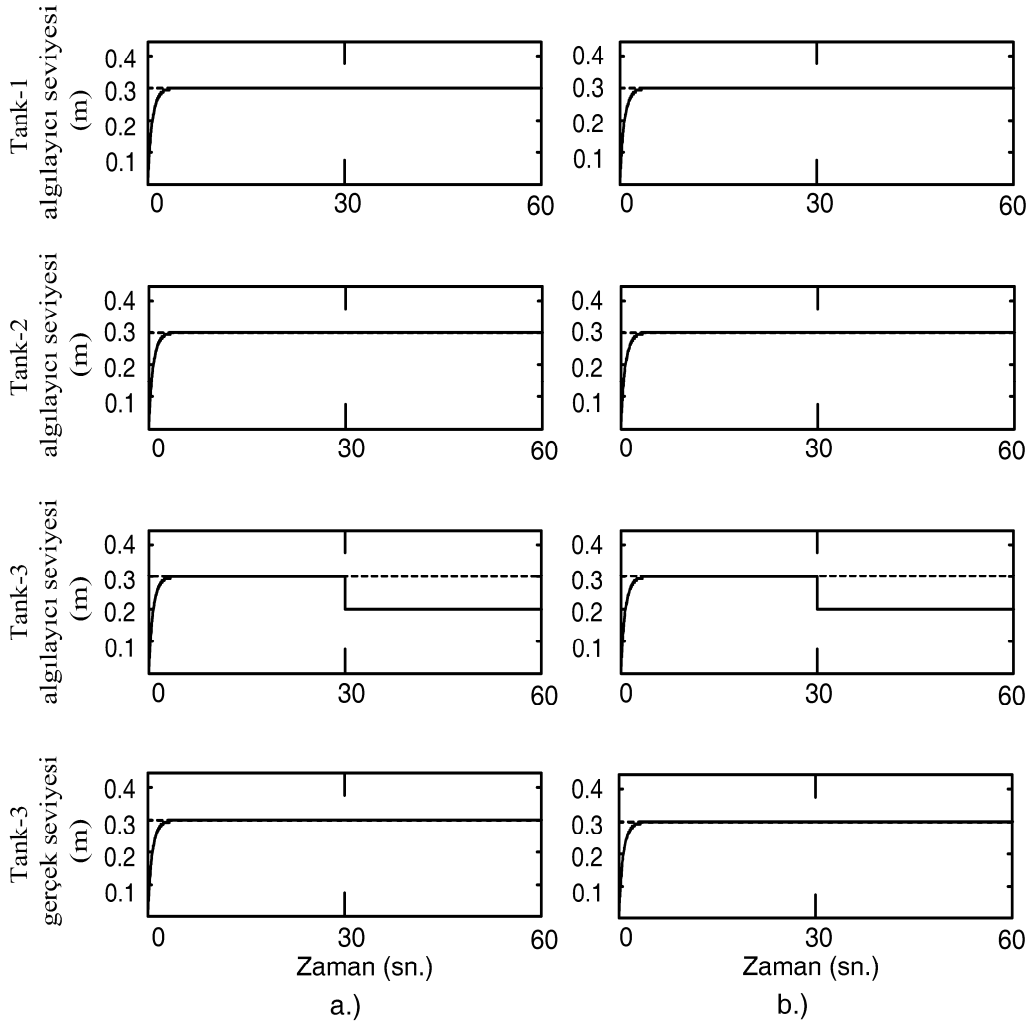
4.4.3. Önceden tanımlı arızaların gerçek zamanlı olarak dayanımı

Sistem üzerinde arızanın hangi kümeye ait olduğu, en kısa Euclidean uzaklığına göre tespit edildikten sonra, ilgili kümeye göre arızanın dayanım işlemi gerçekleştirilmektedir.

ATT işleminden sonra düzeltmenin yapılması için kullanılan Matlab-Simulink modeli Şekil A.18'de verilmektedir. Tasarlanan ADD sistemi ile tank-1, tank-2 ve tank-3 algılayıcı arızaları ile ilgili olarak; arızanın tipi, büyüklüğü ve hangi tankta meydana geldiği tam olarak tespit edilebilmektedir.

Tablo 4.1'de verilmiş olan A2, A4, A5 ve A6 numaralı arızalar tank-1 ve tank-3 algılayıcılarında meydana geldiği için, arıza öncesi ve sonrası tankların gerçek seviyeleri değişmemektedir. Bunun sebebi, üçlü tank sisteminde amaç tank-2'nin seviyesini sabit tutmak olduğundan, tank-2 ölçülen seviyesinde herhangi bir değişme olmadığı sürece denetleyici tank-2 seviyesine müdahale etmek maksadıyla tank-1 ve tank-3 seviyelerini değiştirmemektedir. Bu nedenle, tank-1 ve tank-3 algılayıcı arızaları tespit edildiğinde, herhangi bir arıza düzeltme işlemi yapılmamaktadır.

Tank-1 ve tank-3 arızalarının tümünde, sistemin arıza dayanımlı ve dayanımsız olarak vermiş olduğu cevaplar aynı olduğundan, örnek olarak sadece A5 numaralı tank-3 arızası Şekil 4.19'de gösterilmektedir.



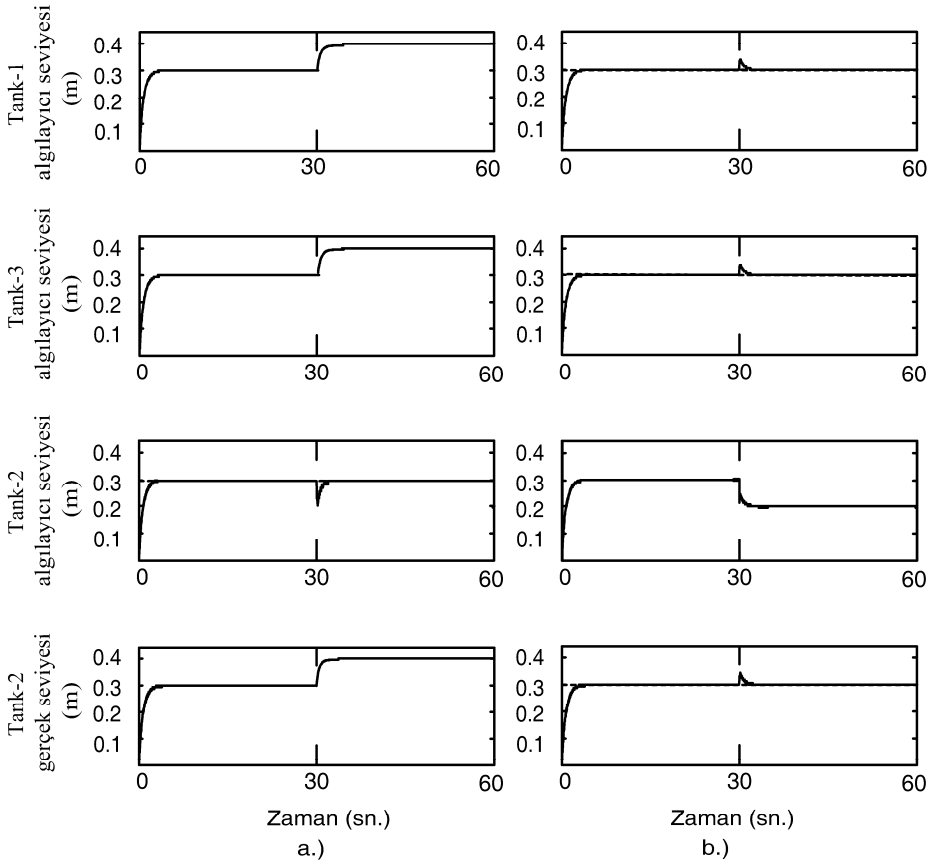
Şekil 4.19 (a): A5 numaralı arıza için arıza dayanımsız denetim
(b): A5 numaralı arıza için arıza dayanımlı denetim

Tablo 4.1'de verilmiş olan A1, A3 ve A7 numaralı tank-2 arızaları uygulandığında ise, tank-2 seviyesini korumak amacıyla PID denetleyicisinin devreye girmesi sonucunda, arıza dayanımsız denetimde tankların tümünün gerçek seviyelerinin değiştiği gözlemlenmektedir.

Arıza dayanımlı denetimde ise, bu arızalar EUEÖTA ile tespit edilen arıza kümelerine göre düzeltilmekte ve tankların seviyeleri referans değeri ile aynı seviyede, doğru olarak ölçülmektedir. A1, A3 ve A7 numaralı tank-2 arızalarında, sistemin arıza dayanımsız denetim ve arıza dayanımlı denetim koşullarında vermiş olduğu cevaplar sırası ile Şekil 4.20, 4.21 ve 4.22'de gösterilmektedir.

Tank-2 üzerinde, Tablo 4.1'deki A1 numaralı toplamsal -0.1 arızası 30. saniyede uygulandığında, arıza dayanımsız sistemde, Şekil 4.20-(a)'da görüldüğü gibi, gerçek seviyede herhangi bir değişim olmamasına rağmen, tank-2 algılayıcısı arıza sonrasında 0.2m ölçüm göstermektedir. PID denetleyicisi arızalı 0.2m'lik ölçüm değerini yeniden referans girişi ile aynı seviyeye getirebilmek için tank-1 ve tank-3 seviyelerini dolayısı ile de tank-2 seviyesini 0.1m artırmaktadır. Hatalı düzeltme işleminden sonra, her üç tankın da gerçek seviyeleri 0.4m olmasına rağmen, tank-1 ve tank-3 algılayıcıları 0.4m ölçüm gösterirken tank-2 algılayıcısı 0.3m ölçüm göstermektedir.

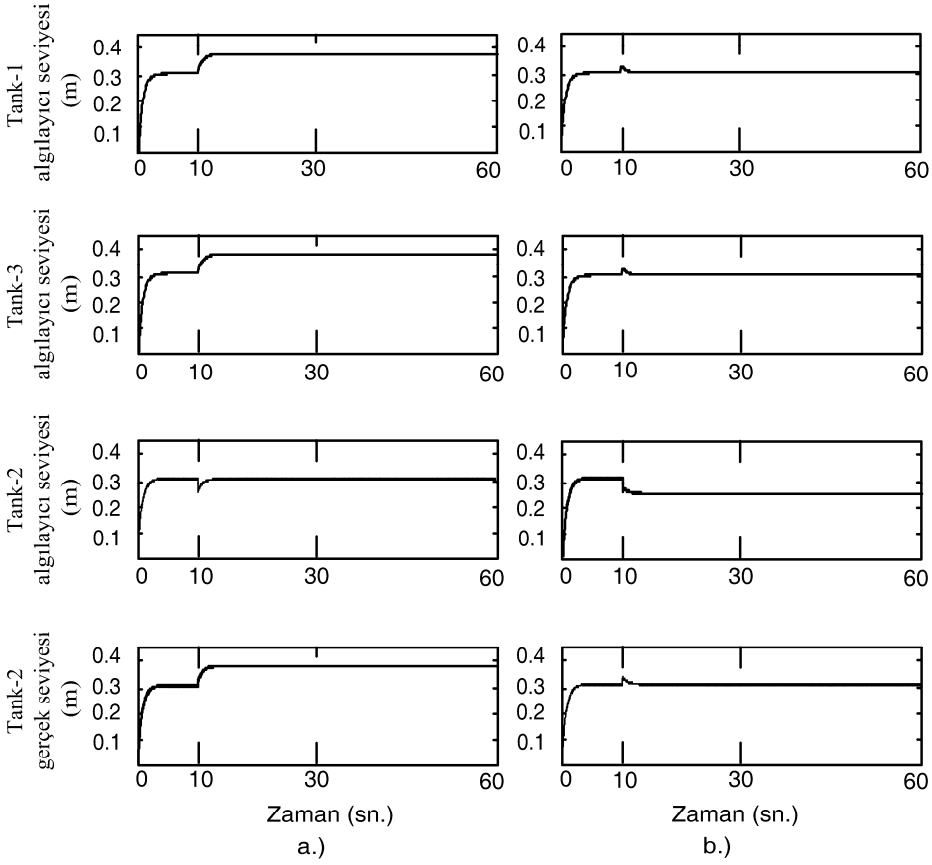
Şekil 4.20-(b)'de arıza dayanımlı denetim ile arızanın tank-2 toplamsal -0.1 arıza kümesine ait olduğu yaklaşık 0.2 saniye içerisinde tespit edilebilmektedir. Arıza sonrasında, tank-2 seviye algılayıcısı 0.2m'lik hatalı ölçümü sürdürür iken, tank-1, tank-2 ve tank-3 gerçek seviyeleri arıza düzeltme işlemi sonucunda 0.3m'lik referans girişi ile aynı değerde ve doğru olarak gözlenmektedir.



Şekil 4.20 (a): A1 numaralı arıza için arıza dayanımsız denetim
(b): A1 numaralı arıza için arıza dayanımlı denetim

Tank-2 üzerinde, Tablo 4.1'deki A3 numaralı çarpımsal 0.8 arızası 10. saniyede uygulandığında, arıza dayanımsız sistemde, Şekil 4.21-(a)'da görüldüğü gibi, gerçek seviyede herhangi bir değişim olmamasına rağmen, tank-2 algılayıcısı arıza sonrasında 0.24m ölçüm göstermektedir. PID denetleyicisi arızalı 0.24m'lik ölçüm değerini yeniden referans girişi ile aynı seviyeye getirebilmek için tank-1 ve tank-3 seviyelerini dolayısı ile de tank-2 seviyesini 0.06m artırmaktadır. Hatalı düzeltme işleminden sonra, her üç tankın da gerçek seviyeleri 0.36m olmasına rağmen, tank-1 ve tank-3 algılayıcıları 0.36m ölçüm gösterirken tank-2 algılayıcısı 0.3m ölçüm göstermektedir.

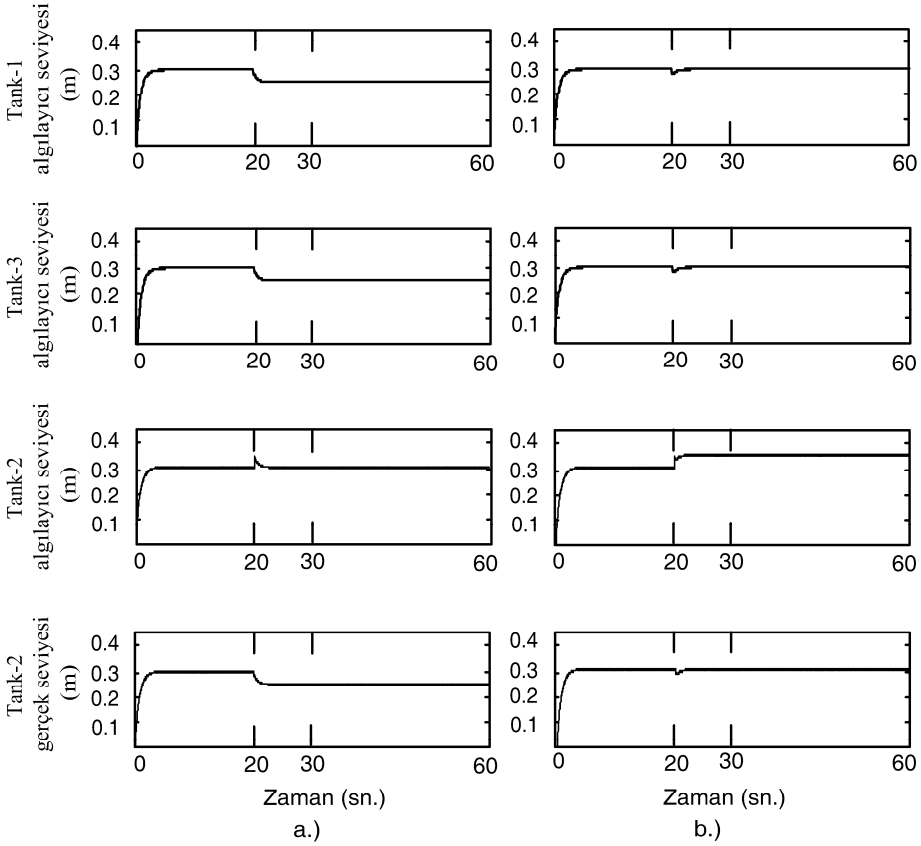
Şekil 4.21-(b)'de arıza dayanımlı denetim ile, arızanın tank-2 çarpımsal 0.8 arıza kümesine ait olduğu yaklaşık 0.2 saniye içerisinde tespit edilebilmektedir. Arıza sonrasında, tank-2 seviye algılayıcısı 0.24m'lik hatalı ölçümü sürdürür iken, tank-1, tank-2 ve tank-3 gerçek seviyeleri arıza düzeltme işlemi sonucunda 0.3m'lik referans girişi ile aynı değerde ve doğru olarak gözlenmektedir.



Şekil 4.21 (a): A3 numaralı arıza için arıza dayanımsız denetim
(b): A3 numaralı arıza için arıza dayanımlı denetim

Tank-2 üzerinde, Tablo 4.1'deki A7 numaralı toplamsal +0.05 arızası 20. saniyede uygulandığında, arıza dayanımsız sistemde, Şekil 4.22-(a)'da görüldüğü gibi, gerçek seviyede herhangi bir değişim olmamasına rağmen, tank-2 algılayıcısı arıza sonrasında bir anlık 0.35m'lik ölçüm göstermektedir. PID denetleyicisi arızalı 0.35m'lik ölçüm değerini yeniden referans girişi ile aynı seviyeye getirebilmek için tank-1 ve tank-3 seviyelerini dolayısı ile de tank-2 seviyesini 0.05m azaltmaktadır. Hatalı düzeltme işleminden sonra, her üç tankın da gerçek seviyeleri 0.25m olmasına rağmen, tank-1 ve tank-3 algılayıcıları 0.25m ölçüm gösterirken tank-2 algılayıcısı 0.3m ölçüm göstermektedir.

Şekil 4.22-(b)'de arıza dayanımlı denetim ile arızanın tank-2 toplamsal +0.05 arıza kümesine ait olduğu yaklaşık 0.2 saniye içerisinde tespit edilebilmektedir. Arıza sonrasında, tank-2 seviye algılayıcısı 0.35m'lik hatalı ölçümü sürdürür iken, tank-1, tank-2 ve tank-3 gerçek seviyeleri arıza düzeltme işlemi sonucunda 0.3m'lik referans girişi ile aynı değerde ve doğru olarak gözlenmektedir.



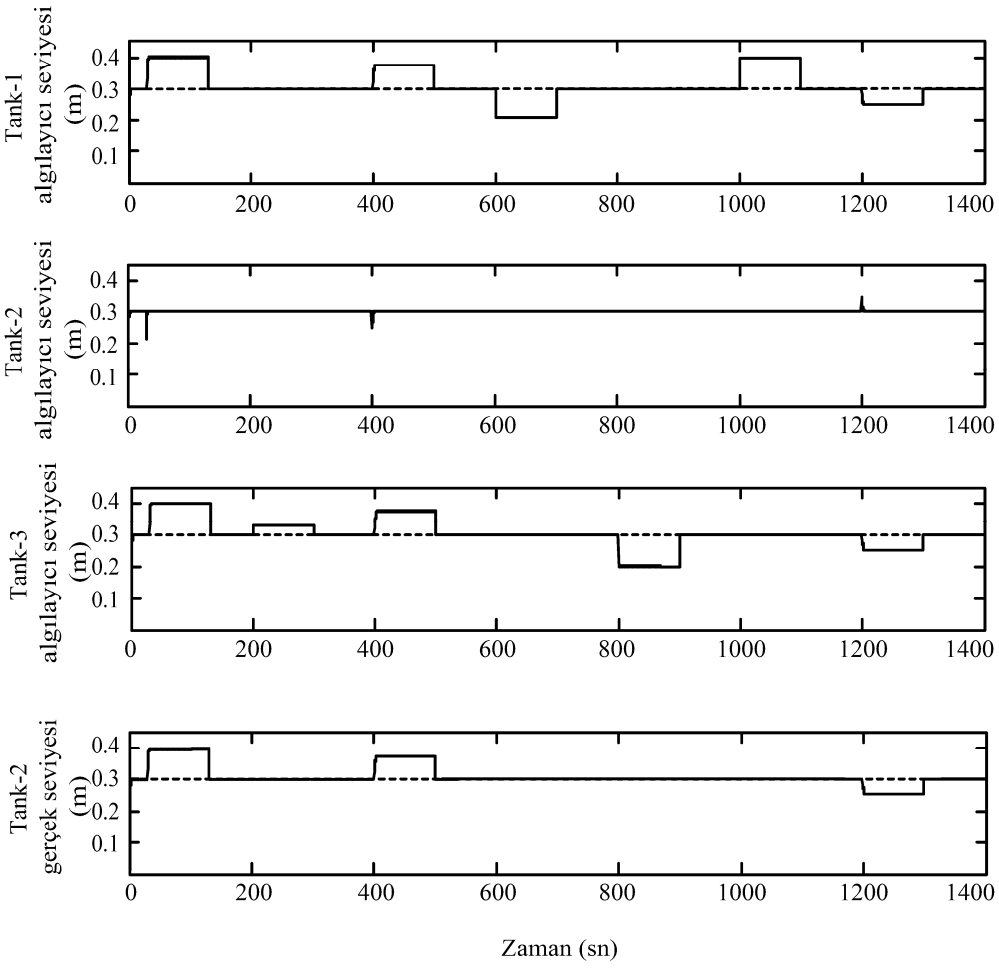
Şekil 4.22 (a): A7 numaralı arıza için arıza dayanımsız denetim
(b): A7 numaralı arıza için arıza dayanımlı denetim

Tablo 4.1’deki A1-A7 arızalarının tümü Tablo 4.3’te verilen süreler boyunca ADD sistemine uygulanmış, kullanmış olduğumuz arıza tespit ve tanılama yönteminin etkisi belirlenmeye çalışılmıştır. Bunun sonucunda, her bir arızanın yaklaşık olarak 0.2 saniye içerisinde tespit edildiği gözlenmiştir. Örneğin, A1 arızası 30 ve 130’ncu saniyeler arasında uygulanmış ve 30.2’nci saniyede tespit edilmiştir. Her bir arızanın uygulanma süreleri Tablo 4.3’te verilmektedir.

Tablo 4.3: Arızaların ADD sistemine uygulanma ve ATT süreleri

	Durum						
	A1	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**
Arıza Süresi (sn.)	30-130	200-300	400-500	600-700	800-900	1000-1100	1200-1300
ATT (sn.)	30.2	200.2	400.2	600.2	800.2	1000.2	1200.2

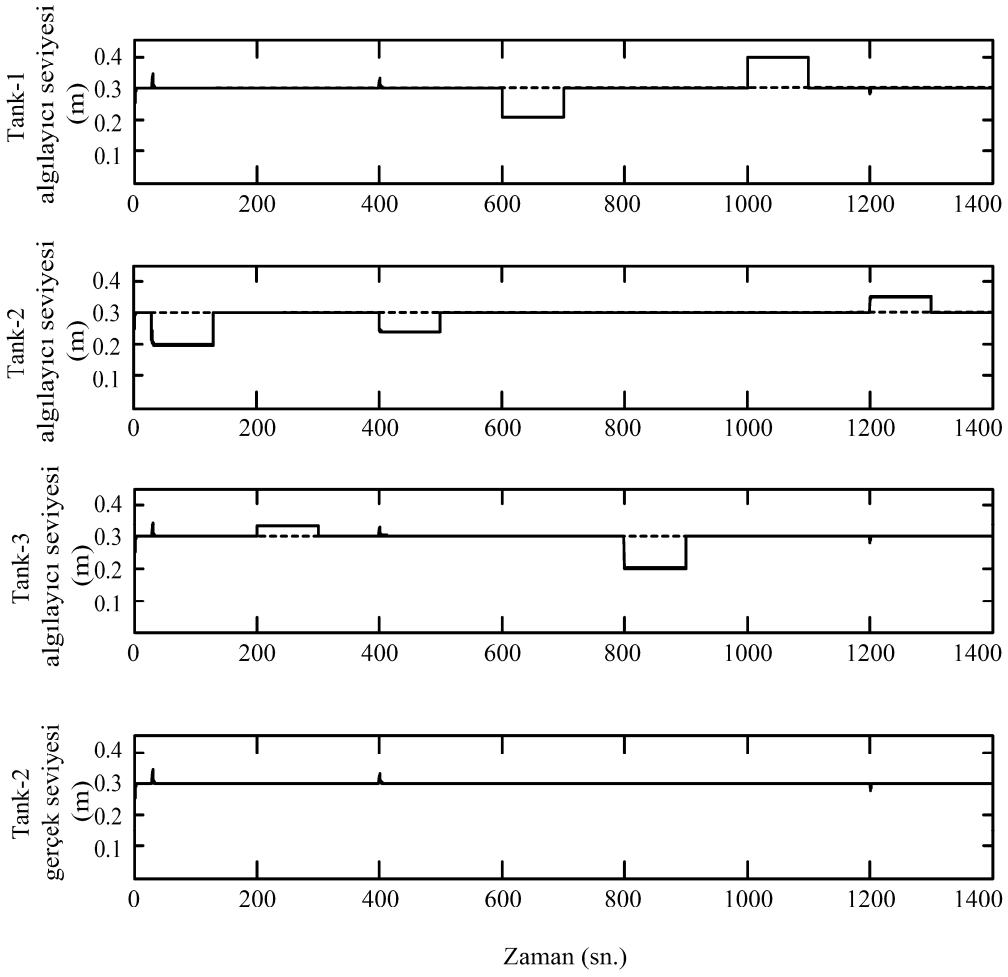
Tablo 4.3’te gösterilmekte olan arızaların, arıza dayanımsız denetim sistemlerinde gözlenen sonuçları Şekil 4.23 gösterilmektedir.



Şekil 4.23: Önceden tanımlı tüm arızalar için arıza dayanımsız denetim sonuçları

Şekil 4.23'te görüldüğü gibi, tank-2 gerçek seviyesi arıza dayanımsız bir denetim sisteminde, A1 arızası için 30–130. saniyeler boyunca 0.4m'ye yükselmekte, A3 arızası için 400–500. saniyeler arasında 0.36m'ye yükselmekte ve A7 arızası için 0.25m'ye düşmektedir. Tank-2 algılayıcısından ölçülen değerin sırası ile 0.2, 0.24 ve 0.35 olması gerekirken arızalı algılayıcı 0.3m olarak ölçülmektedir. Arıza dayanımlı denetim sisteminde ise, tank-2 algılayıcısı 30–130 saniyeler arasında 0.2m'lik ölçüm yaparken, tank-2 gerçek seviyesi 0.3m olarak düzeltilmektedir.

Tablo 4.3'te gösterilmekte olan arızaların, arıza dayanımlı denetim sistemlerinde gözlenen sonuçları ise Şekil 4.24 gösterilmektedir. Şekil 4.24'te görüldüğü gibi, tank-2 gerçek seviyesi, algılayıcılar üzerinde arızalar uygulanmasına rağmen 0.3m olarak tutulabilmektedir.



Şekil 4.24: Önceden tanımlı tüm arızalar için arıza dayanımlı denetim sonuçları

4.4.4. Önceden tanımlı olmayan arızaların gerçek zamanlı olarak dayanımı

Üçlü tank sistemi üzerinde, önceden tanımlı olan 7 arıza durumuna göre sistemin çalışması önceki bölümlerde irdelenmişti. Ancak, gerçek zamanlı uygulamalarda sistem üzerinde önceden tanımlı olmayan arızalar da meydana gelebilmektedir. Bu nedenle, bilinen 7 arızadan farklı olarak, toplamsal ve çarpımsal olmak üzere 13 adet yeni arıza üretilmiştir. Önceden tanımlı olmayan bu algılayıcı arızaları, özellikleri ve durum numaraları ile birlikte Tablo 4.4'de verilmektedir. Önceden tanımlı olmayan bu 13 arızanın, önceden bilinen 7 adet arıza için belirlenmiş küme merkezlerine göre arıza tespit ve tanılamaları yapıldığında, 10 tanesinin arıza tipinin ve arızanın meydana geldiği tank numarasının doğru tanılandığı, 3 tanesinde ise yanlış tanılandığı görülmektedir. Yeni arızaların atanmış oldukları arıza kümeleri, sistemin vermiş olduğu arıza alarmı ve karesel hatalar toplamı dönütleri (ISE-integral squared error) Tablo 4.4'te gösterilmektedir.

Tablo 4.4: Önceden tanımlı olmayan arızaların atandıkları arıza kümeleri

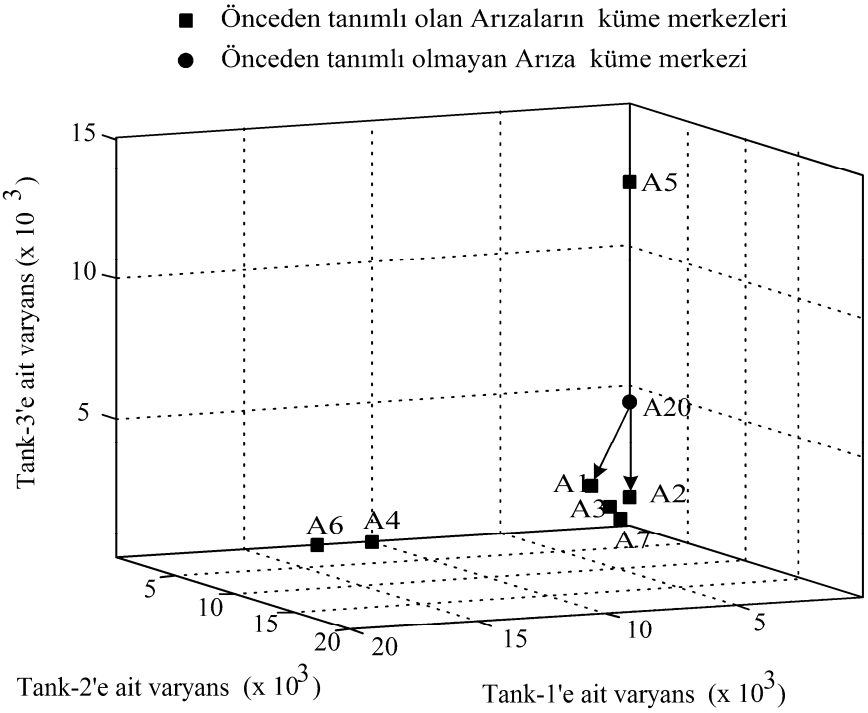
Durum	**Arıza Tipi**	**Arıza Kümesi**	**Doğru Alarm**	**ISE**
A8	tank-1 seviye algılayıcısı çarpımsal arıza (0.6)	A6	✓	2.97
A9	tank-1 seviye algılayıcısı çarpımsal arıza (0.75)	A4	✓	2.97
A10	tank-1 seviye algılayıcısı çarpımsal arıza (0.8)	A1	X	31.51
A11	tank-2 seviye algılayıcısı toplamsal arıza (-0.075)	A3	✓	3.82
A12	tank-2 seviye algılayıcısı toplamsal arıza (-0.09)	A1	✓	3.82
A13	tank-2 seviye algılayıcısı toplamsal arıza (-0.125)	A1	✓	5.14
A14	tank-2 seviye algılayıcısı çarpımsal arıza (0.6)	A1	✓	6.80
A15	tank-2 seviye algılayıcısı çarpımsal arıza (0.7)	A1	✓	3.59
A16	tank-2 seviye algılayıcısı çarpımsal arıza (0.9)	A7	✓	4.35
A17	tank-3 seviye algılayıcısı toplamsal arıza (-0.075)	A5	✓	2.97
A18	tank-3 seviye algılayıcısı toplamsal arıza (-0.125)	A5	✓	2.97
A19	tank-3 seviye algılayıcısı çarpımsal arıza (0.8)	A1	X	31.51
A20	tank-3 seviye algılayıcısı çarpımsal arıza (1.2)	A1	X	31.51

Doğru olarak tanılanan 10 yeni arızanın arıza tipi ve tank numarası her ne kadar doğru olarak belirlenmiş olsa da, düzeltmenin boyutu bilinen küme merkezlerine göre önceden belirlendiği için, %100'lük tam bir düzeltme gerçekleşmemekte ve

sistem bir miktar hata ile çalışmasını sürdürmektedir. Sistem üzerindeki tüm arıza durumları için karesel hataların integrali hesaplanmıştır. Tablo 4.4'te görüldüğü gibi arızanın doğru tespit edildiği durumlarda ISE küçük çıkmaktadır. Ancak, yanlış tanılanan 3 yeni arıza A10, A19 ve A20 durumlarında ISE değeri istenilmeyen ölçüde yüksektir. ISE'nin hesaplanmasında kullanılan Denklem (4.10)'da verilmektedir. Denklemde, $y_g(k)$ tank gerçek seviyesini, $y_r(k)$ tank referans seviyesini, N_s örnekleme sayısını göstermektedir.

$$ISE = \sum_{k=1}^{N_s} (y_g(k) - y_r(k))^2 \tag{4.10}$$

Tablo 4.4'teki önceden bilinmeyen arızalar sisteme uygulandığında; A10, A19 ve A20 arızaları hatalı tanılanmış ve yanlış arıza kümelerine atanmışlardır. Bu nedenle, yapılan düzeltmeler ters etki yapmış ve bu arızaların ISE'lerinin diğer arızalar ile karşılaştırıldığında nispeten yüksek çıkmasına sebep olmuştur. Şekil 4.25 incelendiğinde; A20 arızası ile önceden bilinen A1 arızası arasındaki Euclidean uzaklığı, A20 arızası ile önceden bilinen A2 arızası arasındaki Euclidean uzaklığına göre daha azdır. Bu sebeple, A20 algılayıcı arızası A1 arıza kümesine daha yakın olduğu için, hatalı bir şekilde A2 arızası yerine A1 arızası olarak tanılanmıştır.



Şekil 4.25: A20 arızasının yanlış tanılanması

4.4.5. Önceden tanımlı olmayan arızaların iyileştirilmiş EUEÖTA yöntemiyle dayanımı

Üçlü tank sistemi üzerinde önceden tanımlı olmayan 13 yeni arızanın uygulanması sonucunda, A10 numaralı tank-1, A19 ve A20 numaralı tank-3 arızaları tank-2 arızası olarak yanlış tanılanmaktadır. Arızanın hatalı olarak tespit edilmesi ve yanlış arıza kümesine dahil edilmesi nedeniyle, sistemdeki tankların sıvı seviyeleri yanlış olarak düzeltilmektedir. Arızaların yanlış olarak tespit edilmesini önlemek için, Tablo 4.2'de verilmiş olan mevcut arıza küme merkezleri birbirlerinden ayrıştırılarak EUEÖTA ile kümelemede bir iyileştirme yapılmıştır. Bu yöntemde, çevrimdışı olarak algılayıcı işaretlerinden alınan değerlerin varyans değişim oranı hesaplanırken, arızanın meydana geldiği tanka ait değere, Tablo 4.2'de yer alan varyans değerlerinin en büyüğüne eşit veya daha büyük bir değer eklenir. Bu tez çalışmasında, tablodaki en büyük değer olan 12308 değeri yerine yaklaşık olarak $1x10^4$ değeri ilgili tankın varyansına eklenmektedir. Böylece, arızalı tank ve diğer tanklar arasındaki küme merkezi mesafesi artırılmakta ve arıza kümeleri birbirlerinden ayrıştırılmaktadır. Ayrıştırma yönteminin kullanılması sonucunda, her bir arıza sadece ilgili olduğu tanka ait arıza kümesine girmekte ve arızaların birbirinden izole edilmesi sağlanmaktadır. Tablo 4.2'deki mevcut 7 arıza ve normal durum için ayrıştırma katsayılarının kullanımı ile oluşturulan tank-1, tank-2 ve tank-3'ün yeni küme merkezlerinin koordinatları Tablo 4.5'te verilmektedir.

Tablo 4.5: Ayrıştırma sonrası arıza küme merkezleri koordinatları

Durum	**Küme merkezleri koordinatları**		
	tank-1	**tank-2**	**tank-3**
N	0	0	0
A1	1573.54	10304.39	1573.54
A2	0	0	11116.30
A3	639.03	10126.79	639.03
A4	20018.08	0	0
A5	0	0	22308.11
A6	22141	0	0
A7	387.75	10076.40	387.75

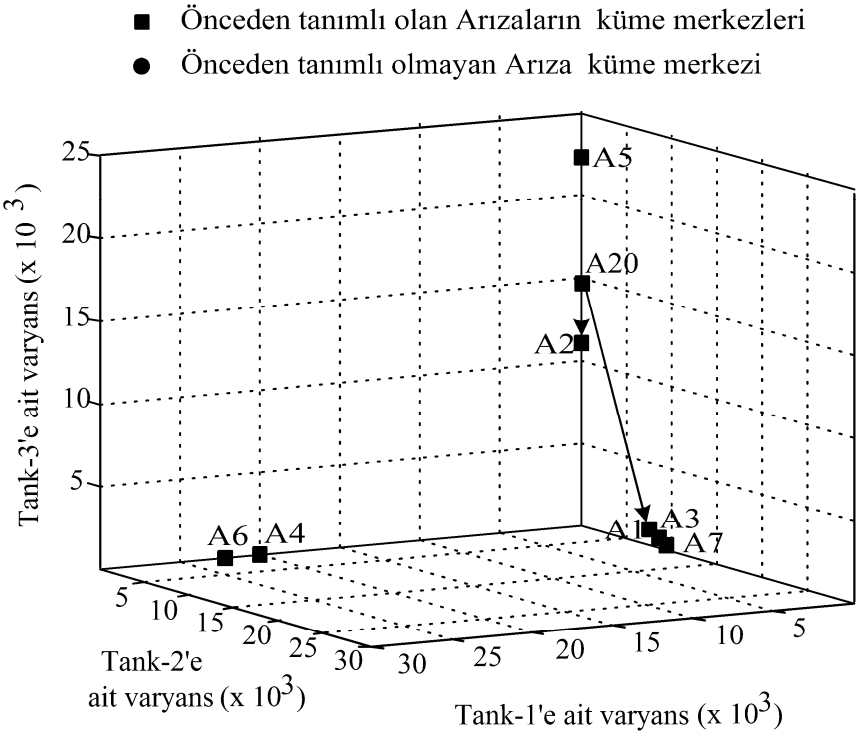
Tablo 4.4'te verilmiş olan önceden tanımlı olmayan yeni 13 arıza, üçlü tank sistemi üzerinde, Tablo 4.5'te verilen ayrıştırılmış yeni küme merkezlerine göre tekrar uygulanmıştır. Arızaların atanmış oldukları arıza kümeleri, sistemin vermiş olduğu arıza alarmı ve karesel hatalar toplamı dönütleri (ISE) Tablo 4.6'da gösterilmektedir. Tabloda da görüldüğü gibi, tüm arızalar doğru olarak tespit edilmekte, sistem üzerinde yanlış alarm meydana gelmemekte ve arızaların ait oldukları tanklar doğru tespit edildiği için karesel hataların integrali (ISE) istenilen ölçüde küçük çıkmaktadır.

Tablo 4.6: Ayrıştırma sonrasında önceden tanımlı olmayan arızaların atandıkları arıza kümeleri

Durum	**Arıza Tipi**	**Arıza Kümesi**	**Doğru Alarm**	**ISE**
A8	tank-1 seviye algılayıcısı çarpımsal arıza (0.6)	A6	✓	2.97
A9	tank-1 seviye algılayıcısı çarpımsal arıza (0.75)	A4	✓	2.97
A10	tank-1 seviye algılayıcısı çarpımsal arıza (0.8)	A4	✓	2.97
A11	tank-2 seviye algılayıcısı toplamsal arıza (-0.075)	A3	✓	3.82
A12	tank-2 seviye algılayıcısı toplamsal arıza (-0.09)	A1	✓	3.82
A13	tank-2 seviye algılayıcısı toplamsal arıza (-0.125)	A1	✓	5.14
A14	tank-2 seviye algılayıcısı çarpımsal arıza (0.6)	A1	✓	6.80
A15	tank-2 seviye algılayıcısı çarpımsal arıza (0.7)	A1	✓	3.59
A16	tank-2 seviye algılayıcısı çarpımsal arıza (0.9)	A7	✓	4.35
A17	tank-3 seviye algılayıcısı toplamsal arıza (-0.075)	A5	✓	2.97
A18	tank-3 seviye algılayıcısı toplamsal arıza (-0.125)	A5	✓	2.97
A19	tank-3 seviye algılayıcısı çarpımsal arıza (0.8)	A2	✓	2.97
A20	tank-3 seviye algılayıcısı çarpımsal arıza (1.2)	A2	✓	2.97

Ayrıştırma yönteminin kullanılması sonucunda, Tablo 4.6'da görüldüğü gibi, A20 arıza durumu A2 tank-3 arıza kümesine ait olmakta ve A2 arızası düzeltme işlemine tabi tutulmaktadır. A20 durumlu arızanın doğru olarak tanılanması Şekil 4.26'da gösterilmektedir. Şekildeki A1-A7 önceden tanımlı olan arıza küme merkezlerini, A20 ise önceden tanımlı olmayan bir arızayı göstermektedir. Şekil 4.26, Şekil 4.25 ile kıyaslandığında, A1-A7 önceden tanımlı arıza küme merkezleri arasındaki mesafelerin, tank numaraları dikkate alınarak arttırılmış olduğu görülmektedir. Bunun sonucunda, A20 arızası ile A1 küme merkezi arasındaki Euclidean mesafesi

artmakta iken A2 küme merkezi ile arasındaki mesafe azalmaktadır. Sonuç olarak, A20 arızası A2 arızası kümesine girerek tank-3 arızası olarak tanılanmaktadır. Ayrıca, tank-1, tank-2 ve tank-3 arıza küme merkezlerinin birbirinden tamamen ayrıştırıldığı şekil üzerinde görülmektedir. Böylece, tüm arızalar meydana geldikleri tank ile ilgili olan arıza kümelerine göre tespit edilmekte ve düzeltilmektedirler.

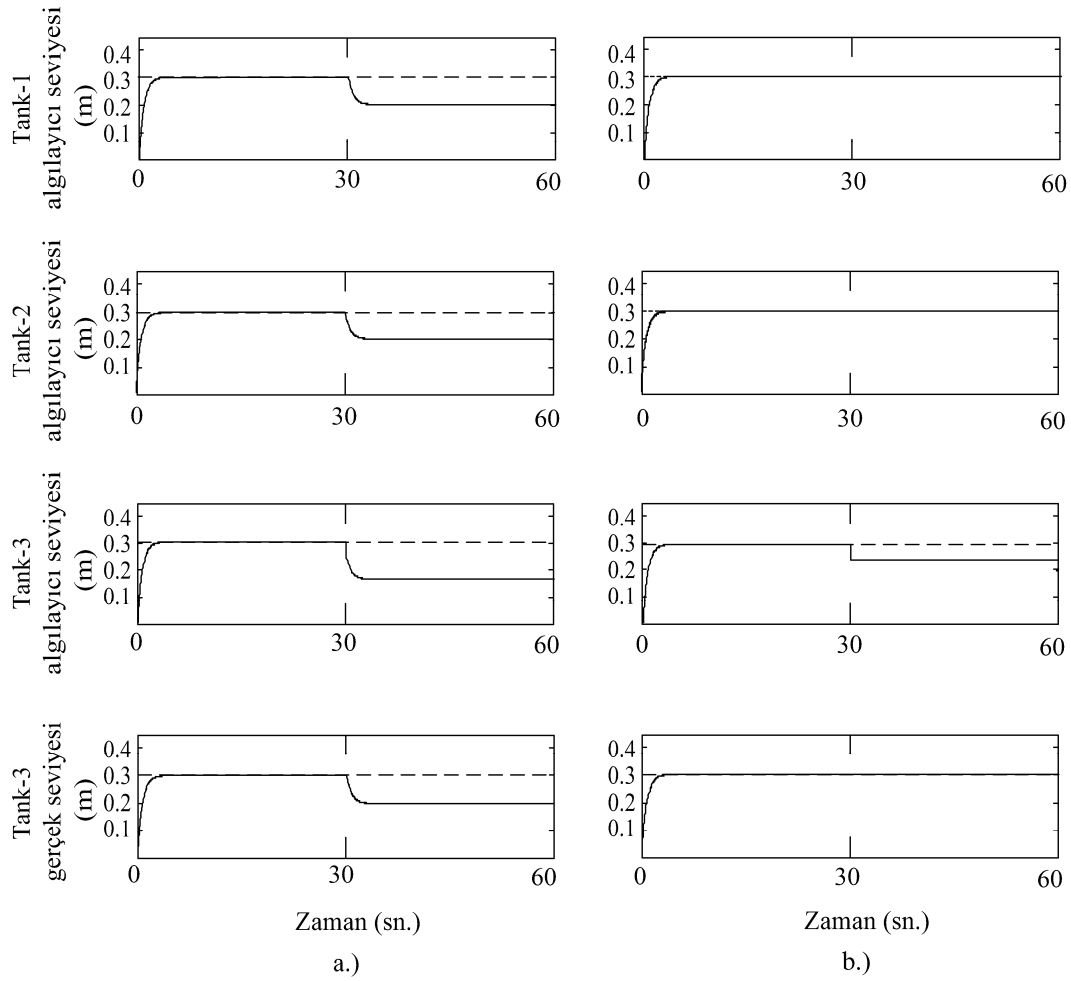


Şekil 4.26: A20 arızasının doğru tanılanması

Şekil 4.27'de, önceden bilinmeyen A10 arızasının düzeltilmesi gösterilmektedir. Ayrıştırma yapılmadığında; tank-1 çarpımsal 0.8 arızası A10, tank-2 toplamsal -0.1 arızası A1 olarak yanlış tanılanmakta ve A1 arızası ile karşılaşıldığında 0.3m olan tank-2 seviyesini 0.2m'ye düşürmek için 0.1m'lik azaltma yapan bir düzeltme işlemi kullanılmaktadır. Arıza düzeltilmesi yapıldıktan sonra, tank-2 gerçek seviyesi 0.2m'ye düşecek, tank-2 algılayıcısı sağlam olduğu için 0.2m'yi ölçecek, tank-1 gerçek seviyesi tank-2'ye eşit olacağı için 0.2m olacak, tank-1 algılayıcısı çarpımsal 0.8 arızalı olduğu için 0.2mx0.8=0.16m ölçecektir.

Ayrıştırma yapıldığında ise; tank-1 çarpımsal 0.8 arızası A10, tank-1 çarpımsal 0.7 arızası A4 olarak tanılanmakta ve tank-1 arızası ile karşılaşıldığı için hiçbir düzeltme yapılmamaktadır. Bu nedenle, tank-2 gerçek seviyesi 0.3m'de sabit kalacak, tank-2

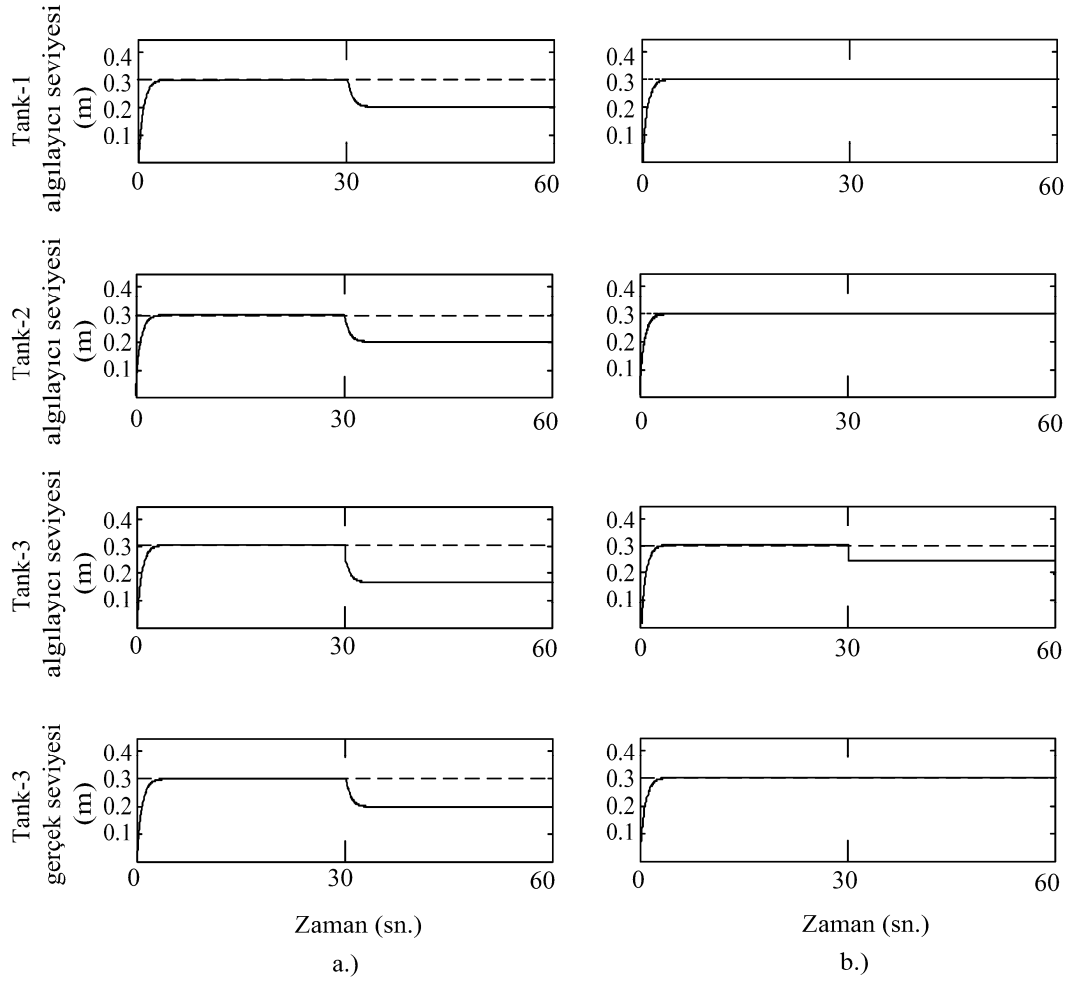
algılayıcısı sağlam olduğu için 0.3m'yi ölçecek, tank-1 gerçek seviyesi tank-2'ye eşit olacağı için 0.3m olacak, tank-1 algılayıcısı çarpımsal 0.8 arızalı olduğu için 0.3mx0.8=0.24m ölçecektir. Ayrıştırma yapılarak ADD'in sistem başarımını sürdürdüğü açık bir şekilde görülmektedir.



Şekil 4.27 (a): A10 arızasının ayrıştırma yapılmadan düzeltilmesi
(b): A10 arızasının ayrıştırma yapılarak düzeltilmesi

Şekil 4.28'de, önceden bilinmeyen A19 arızasının düzeltilmesi gösterilmektedir. Ayrıştırma yapılmadığında; tank-3 çarpımsal 0.8 arızası A19, tank-2 toplamsal -0.1 arızası A1 olarak yanlış tanılanmakta ve A1 arızası ile karşılaşıldığında 0.1m'lik azaltma yapan bir düzeltme işlemi kullanılmaktadır. Arıza düzeltilmesi yapıldıktan sonra, tank-2 gerçek seviyesi 0.2m'ye düşecek, tank-3 gerçek seviyesi tank-2'ye eşit olacağı için 0.2m olacak, tank-3 algılayıcısı çarpımsal 0.8 arızalı olduğu için 0.2mx0.8=0.16m ölçecektir.

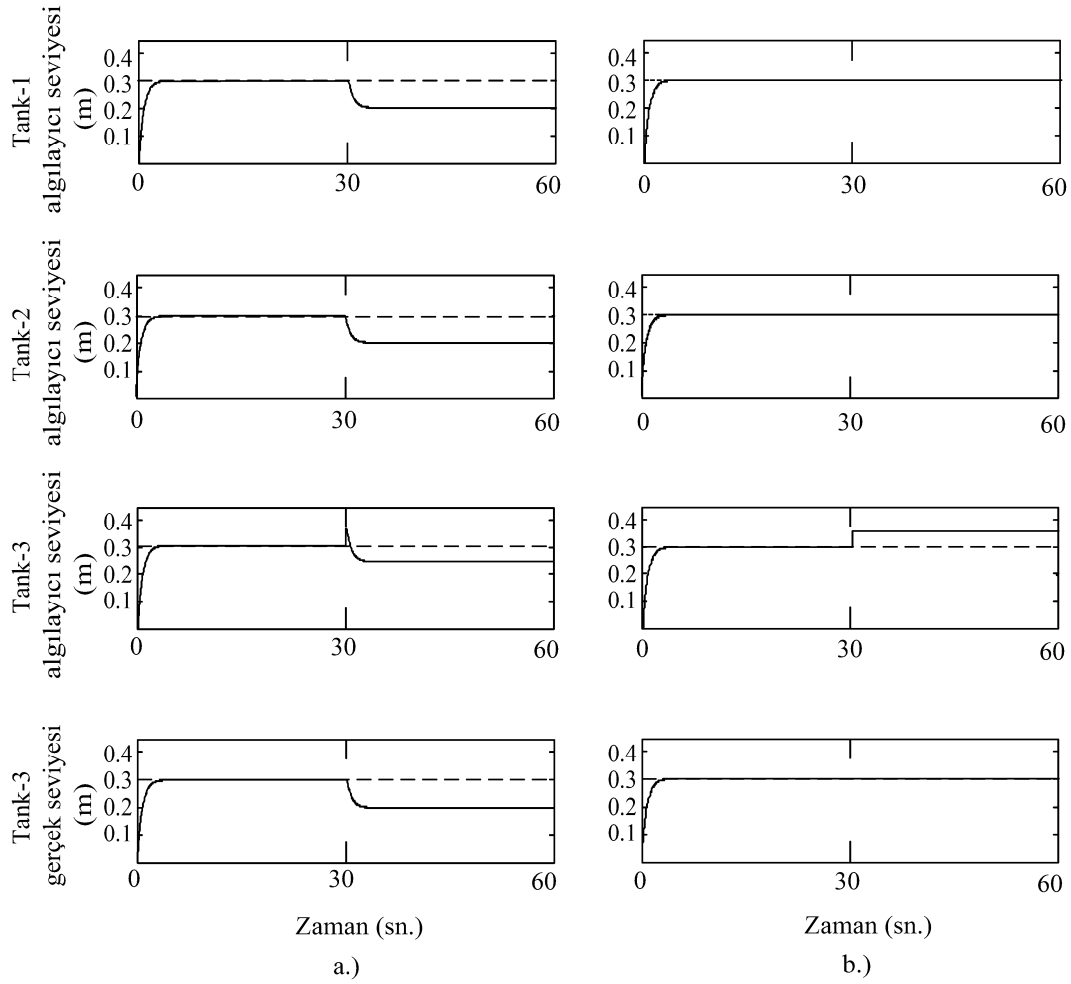
Ayrıştırma yapıldığında ise; tank-3 çarpımsal 0.8 arızası A19, tank-3 çarpımsal 1.1 arızası A2 olarak tanılanmakta ve tank-3 arızası ile karşılaşıldığı için hiçbir düzeltme yapılmamaktadır. Bu nedenle, tank-2 gerçek seviyesi 0.3m'de sabit kalacak, tank-3 gerçek seviyesi tank-2'ye eşit olacağı için 0.3m olacak, tank-3 algılayıcısı çarpımsal 0.8 arızalı olduğu için 0.3mx0.8=0.24m ölçecektir.



Şekil 4.28 (a): A19 arızasının ayrıştırma yapılmadan düzeltilmesi
(b): A19 arızasının ayrıştırma yapılarak düzeltilmesi

Şekil 4.29'da, önceden bilinmeyen A20 arızasının düzeltilmesi gösterilmektedir. Ayrıştırma yapılmadığında; tank-3 çarpımsal 1.2 arızası A20, tank-2 toplamsal -0.1 arızası A1 olarak yanlış tanılanmakta ve A1 arızası ile karşılaşıldığında 0.1m'lik azaltma yapan bir düzeltme işlemi kullanılmaktadır. Arıza düzeltilmesi yapıldıktan sonra, tank-2 gerçek seviyesi 0.2m'ye düşecek, tank-3 gerçek seviyesi tank-2'ye eşit olacağı için 0.2m olacak, tank-3 algılayıcısı çarpımsal 1.2 arızalı olduğu için 0.2mx1.2=0.24m ölçecektir.

Ayrıştırma yapıldığında ise; tank-3 çarpımsal 1.2 arızası A20, tank-3 çarpımsal 1.1 arızası A2 olarak tanılanmakta ve tank-3 arızası ile karşılaşıldığı için hiçbir düzeltme yapılmamaktadır. Bu nedenle, tank-2 gerçek seviyesi 0.3m'de sabit kalacak, tank-3 gerçek seviyesi tank-2'ye eşit olacağı için 0.3m olacak, tank-3 algılayıcısı çarpımsal 1.2 arızalı olduğu için 0.3mx1.2=0.36m ölçecektir.



Şekil 4.29 (a): A20 arızasının ayrıştırma yapılmadan düzeltilmesi
(b): A20 arızasının ayrıştırma yapılarak düzeltilmesi

SONUÇLAR VE ÖNERİLER

Arızalar, endüstriyel sistemler ve süreçler üzerinde sistemin normal olmayan cevaplar vermesine neden olurlar. Öte yandan, normal sistem uygulamalarında referans girişin değiştirilmesi de aynı değişikliklere neden olur. Normal ve arızalı durumlarda, sistemin ürettiği verilerin çevrim dışı olarak analizi, arıza durumlarını ayırt etmemizde kullanabileceğimiz performans ölçütlerinden biridir. Denetim sistemlerinin, arızanın meydana gelmesi halinde bile sistem başarımını sürdürmeleri hem ekonomik açıdan hem de insan hayatı ve sistem güvenliği bakımından çok önemlidir. Bu nedenle, sistemlerin sadece denetlenmesi yeterli olmayıp, meydana gelebilecek arızaların tespit ve tanılamalarının yapılması, sistemlerin çalışır, güvenli ve arıza dayanımlı hale getirilmeleri gerekmektedir.

ADD sistemi, ATT işlemi ile sistemin yeniden yapılandırılmasını içermektedir. ATT ile sistemde bir arıza olup olmadığı, bir arıza söz konusu ise arızanın ne tür bir arıza olduğu belirlenebilir. ATT sonucunda denetleyici yeniden yapılandırılır ve sistemin arızaya rağmen başarımı sağlanır. Arıza dayanımlı denetimin temel amacı, tüm algılayıcı, eyleyici ve bileşen arızalarına rağmen sistemin kararlı ve tatmin edici bir başarımla çalışmasını sürdürmesi için uygun bir denetleyicinin tasarlanmasıdır.

Tank sistemlerinde sıvı seviyesinin denetimi, endüstriyel süreçlerin temel problemlerinden birisidir. Bu tez çalışmasında, kapalı devre bir denetim sistemindeki arızaların tespiti, tanılanması ve arıza dayanımlı bir denetimin gerçekleştirilmesi için üçlü tank sistemi kullanılmıştır. Üçlü tank sistemi, birbirine bağlı üç adet su tankı, pompalar ve valflerden oluşmaktadır. Üçlü tank sisteminde amaç, tank-2 ile etkileşim halinde olan tank-1 ve tank-3'e ait pompa ve valfleri denetleyerek tank-2'deki sıvı seviyesini istenilen düzeyde tutmaktır.

Tez çalışmasında, üçlü tank sisteminin benzetimini yapabilmek ve ihtiyaç duyulan arıza verilerini elde edebilmek için, PID denetleyicili üçlü tank seviye denetim

sisteminin MatLab-Simulink modeli yapılmıştır. PID denetleyicinin sistemde meydana gelebilecek arızalara karşı tepkisini inceleyebilmek için benzetim çalışması üzerinde; 7 adet toplamsal, çarpımsal ve tıkama tipinde algılayıcı arızaları oluşturulmuştur. Sistem modeline, yapay algılayıcı arızaları uygulanarak, arızaların sistem üzerindeki etkileri gözlemlenmiş ve bilgi tabanı oluşturmak üzere, her bir arıza durumuna ait sistemin verileri elde edilmiştir. Yapay algılayıcı arızaları birer birer üçlü tank sistemine uygulanmış; her bir arıza durumunda tank-1, tank-2 ve tank-3'ün gerçek seviyeleri, seviye algılayıcısı vasıtasıyla ölçülen seviyeleri ve PID denetleyicisinin ürettiği denetim işareti sistemin geçmiş verileri olarak sistemden alınmış ve analizleri yapılmıştır.

Çalışmada kullanılmış olan ADD sisteminin genel yapısı PID denetleyici, üçlü tank sistemi ve arıza teşhis bölümlerinden oluşmaktadır. Arıza teşhis sistemi ATT ve bilgi tabanı kısımlarını içermektedir. ATT da kendi içerisinde, istatistiksel analiz ve kümeleme olmak üzere iki aşamaya ayrılmaktadır. Geçmiş sistem verileri üzerinde istatistiksel teknikler kullanılarak özellik çıkartımı yapılmaktadır. Özellik çıkartımının gerçekleştirilmesinde ilk adım olarak kayan pencere yöntemi kullanılmıştır. Önceden tanımlanan tüm arıza durumları için benzetim çalışması sonuçlarında elde edilen veriler pencerelenerek gözlemleme işlemi gerçekleştirilmiştir. Pencereleme safhasında, pencerelenen işaretin genişliği 60, son veri bölgesi genişliği ise 20 örnek olarak alınmıştır. Bu aşamada, tüm pencerelerin varyans değişim oranlarına bakarak hangi pencerede arızanın meydana geldiği tespit edilebilmektedir.

ATT tarafından çevrimdışı olarak elde edilmiş tank verileri, hiyerarşik bir kümeleme yöntemi olan EUEÖTA ile kümelenerek bilgi tabanı oluşturulmuştur. Kümeleme işlemi, hem çevrimdışı yapıldığı ve hız gerektirmediği için hem de görsel özelliklerinden de yararlanabilmek için, bir veri madenciliği yazılımı olan WEKA programı ile yapılmıştır. Önceden tanımlanmış tüm arızalar için, tank seviyelerine göre belirlenen özellik vektörleri bilgi tabanına küme merkezleri olarak kaydedilmiştir.

ATT sistemi çevrimiçi çalışırken, tank algılayıcılarından gelen verilerin varyans değişim oranları ile bilgi tabanında yer alan küme merkezleri arasındaki Euclidean uzaklıkları karşılaştırılarak, arıza tespit edilmekte ve tanılanmaktadır. Sistem üzerinde arızanın hangi kümeye ait olduğu tespit edildikten sonra, ilgili kümeye göre arızanın düzeltilmesi işlemi gerçekleştirilmektedir.

Tez çalışmasında, farklı özelliklerde 7 adet arıza üretilerek, önerilen ADD sistemine uygulanmış ve önerilen yönteminin etkisi belirlenmeye çalışılmıştır. Uygulamalar sonucunda, her bir arızanın yaklaşık olarak 0.2 saniye içerisinde tespit edildiği gözlenmiştir. Tüm arızalar doğru olarak tespit edilmiş, sistem üzerinde yanlış alarm meydana gelmemiş ve arızaların ait oldukları tanklar doğru tespit edildiği için ISE değerleri de istenilen ölçüde küçük çıkmıştır.

Gerçek zamanlı uygulamalarda sistem üzerinde önceden tanımlı olmayan arızalar da meydana gelebilmektedir. Bu nedenle, bilinen 7 arızadan farklı olarak, toplamsal ve çarpımsal olmak üzere 13 adet yeni arıza üretilmiştir. Önceden tanımlı olmayan bu 13 arızanın, önceden bilinen 7 adet arıza için belirlenmiş küme merkezlerine göre arıza tespit ve tanılamaları yapıldığında, 10 tanesinin arıza tipinin ve arızanın meydana geldiği tank numarasının doğru tanılandığı, 3 tanesinde ise yanlış tanılandığı görülmüştür. Doğru olarak tanılanan 10 yeni arızanın arıza tipi ve tank numarası her ne belirlenmiş olsa kadar doğru olarak da, düzeltmenin boyutu bilinen küme merkezlerine göre önceden belirlendiği için, %100'lük tam bir düzeltme gerçekleşememekte, sistem bir miktar hata ile çalışmasını sürdürmektedir. Yanlış tanılanan 3 yeni arıza içinse, ISE değeri çok yüksek çıkmıştır.

Arızaların yanlış olarak tespit edilmesini önlemek için, mevcut arıza küme merkezleri birbirlerinden ayrıştırılarak EUEÖTA ile kümelemede bir iyileştirme yapılmıştır. Bu yöntemde, arızalı tank ve diğer tanklar arasındaki küme merkezi mesafesi artırılmakta ve arıza kümeleri birbirlerinden ayrıştırılmaktadır. Önceden tanımlı olmayan yeni 13 arıza, üçlü tank sistemi üzerinde, ayrıştırılmış yeni küme merkezlerine göre tekrar uygulanmıştır. Tüm arızalar doğru olarak tespit edilmiş, sistem üzerinde yanlış alarm meydana gelmemişe ve arızaların ait oldukları tanklar doğru tespit edildiği için ISE değerleri de istenilen ölçüde küçük çıkmıştır.

Ayrıştırma yapılarak ADD'nin arızaya rağmen sistem başarımını sürdürdüğü açık bir şekilde görülmüştür.

Gelecek çalışmalarda; bu tez çalışmasında kullanılan yöntem ile, çok daha geniş bir bilinmeyen arıza kümesinin tanılanması ve arıza dayanımı gerçekleştirilebilir, sistem içerisindeki diğer bileşen ve eyleyici arızalarının tespiti, tanılanması ve arıza dayanımının başarısı araştırılabilir.

KAYNAKLAR

[1] McMahan, J., "Flight 1080", ***Airline Pilot***, Vol 47-7, 6-11, (1978).

[2] Montoya, R., J., Howell, W., E., Bundick, W., T., Ostroff, A., J., Hueschen, R.,M., Belcastro, C., M., "Restructurable controls"***Tech. Rep. NASA CP-2277 Proceedings of a workshop held at NASA Langley Research Center***, Virginia, (1983).

[3] Wikipedia, "El Al Flight 1862", http://en.wikipedia.org/wiki/El_Al_Flight_1862, ***(Erişim Tarihi 5 Temmuz 2011)***.

[4] Maciejowski, J., M., Jones, C., N., "MPC fault-tolerant flight control case study: flight 1862" ***International Federation of Automatic Control on Safeprocess Sympoisum***., Washington, 119-124, (2003).

[5] Saif, M., Guan, Y.,. "A new aproach to robust fault-detection and identification", ***IEEE Transactions on Aerospace and Electronic Systems*** , Vol. 29-3, 685-695, (1993).

[6] Dorr, R., Kratz, F., Ragot, J. ,Loisy, F., Germain, J., L., "Detection, isolation, and identification of sensor faults in nuclear power plants", ***IEEE Transactions on Control Systems Technology***, Vol. 5-1, 42-60, (1997).

[7] Sauter, D., Hamelin, F., "Frequency –domain optimization for robust fault detection and isolation in dynamic systems", ***IEEE Transactions on Automat Control***, Vol. 44-4, 878-882, (1999).

[8] Simani, S., Fantuzzi, C., Beghelli, S., "Diagnosis techniques for sensor faults of industrial processes", ***IEEE Transactions on Control Systems Technology***, Vol. 8-5, 848-855, (2000).

[9] Hou, Z., Lian, Z., Yao, Y., Yuan, X., "Data mining based sensor fault diagnosis and validation for building air conditioning system", ***Energy Conversion and Management***, Vol 47-15, 2479-2490, (2006).

[10] Huang, Y., C., "A New Data Mining Approach to Dissolved Gas Analysis of Oil-Insulated Power Apparatus", ***IEEE Transactions on Power Delivery***, Vol 18-4, 1257-1261, (2003).

[11] Luo, M., "Data-Driven Fault Detection Using Trending Analysis", Doktora Tezi, ***Louisiana State University***, (2006).

[12] Aydın, İ., "Arıza Teşhisinde Veri Madenciliği ve Yumuşak Hesaplama Tekniklerinin Kullanımı", Yüksek Lisans Tezi, ***Fırat Üniversitesi***, (2006).

[13] Detroja, K., P., Gudi, R., D., Patwardhan, S., C., "A possibilistic clustering approach to novel fault detection and isolation", ***Journal of Process Control***, Vol. 16, 1055-73, (2006).

[14] Aydın, İ., Karaköse, M., Akın, E., "A simple and efficient method for fault diagnosis using time series data mining", ***IEEE Electric Machines & Drives Conference***, Antalya, 596-600, (2007).

[15] Aydın, İ., Karaköse, M., Akın, E., "Artificial Immune Inspired Fault Detection Algorithm Based on Fuzzy Clustering and Genetic Algorithm Methods", ***IEEE International Conference on Computational Intelligence for Measurement Systems And Applications***, İstanbul, (2008).

[16] Postalcıoğlu, S., Erkan, K., "Soft computing and signal processing based active fault tolerant control for benchmark process," ***Journal of Neural Computing & Applications***, Vol. 18, 77-85, (2009).

[17] Balle, P., Isermann, R., "Fault detection and isolation for nonlinear processes based on local linear fuzzy models and Parameter Estimation, ***Proceedings of the American Control Conference***, Philadelphia, 1605-1609, (1998).

[18] Kanev, S., "Robust Fault Tolerant Control", Doktora Tezi, ***University of Twente***, The Netherlands, 3-5, (2004).

[19] Rachiniyani, P., K., "Robust Fault-Tolerant Control for Aircraft Systems", Yüksek lisans", ***University of Louisiana State***, 13-15, (2006).

[20] Chihaia, C., I., "Active fault-tolerance in wireless wetworked control Systems", Doktora Tezi", ***University ofDuisburg-Essen***, 9-10, (2010).

[21] Hallouzi, R., Verdult, V., Babuska, R., Verhaegen, M., "Fault Detection and Identification of Actuator Faults Using Linear Parameter Varying Models", ***IFAC World Congress***, Prague, (2005).

[22] Puig, V., Quevedo, J., "Fault-tolerant controllers using a passive robust fault diagnosis approach", ***Control Engineering Practice***, Vol 9., 1221-1234, (2001).

[23] Zhang, Y., Jiang, J., "Bibliographical review on reconfigurable fault-tolerant control systems", ***Annual Reviews in Control***, Vol. 32-2, 229-252, (2008).

[24] Stengel, R., F., "Intelligent failure-tolerant control", ***IEEE Control Systems Magazine*** , Vol. 11-4, 14-23, (1991).

[25] Mendonça, L., F., Sousa, J., M., C., Costa, J., M., G., "Fault accommodation of an experimental three tank system using fuzzy predictive control", ***IEEE International Conference On Fuzzy Systems*** , Hong Kong, 1619-1625, (2008).

[26] Blanke, M., "Fault-tolerant control systems. In New trends in advanced control", Berlin: Springer, 171–196, (1999).

[27] Patton, R., J., "Fault-tolerant control systems: The 1997 situation", ***IFAC Ifac Safeprocess***, Hull, UK., Vol 3, 1033-1054, (1997).

[28] Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S., N., "A Review of process fault detection and diagnosis Part I: Quantitative model-based methods", ***Computers and Chemical Engineering***, Vol.27, 293-311, (2003).

[29] Patwardhan, S., C., Manuja, S., Narasimhan, S., Shah, S., L., "From data to diagnosis and control using generalized orthonormal basis filters, Part II: model predictive and fault tolerant control", ***Journal of Process Control***, Vol 16, 157-175, (2006).

[30] Prakash, J., Patwardhan, S., C., Narasimhan, S., "A supervisory approach to fault tolerant control of linear multivariable systems", ***Industrial Engineering. Chemistry Research***, Vol 41, 2270–2281, (2002).

[31] Zhang, X., Parisini, T., Polycarpou, M., M., "Adaptive fault-tolerant control of nonlinear uncertain systems: An information-based diagnostic approach", ***IEEE Transactions on automatic Control***, Vol 49, 1259–1274, (2004).

[32] Deshpande, A., P., Patwardhan, S., C., Narasimhan, S., "Intelligent state estimation for fault tolerant nonlinear predictive control", ***Journal of Process Control***, Vol 19, 187-204, (2009).

[33] Patton, R., "Fault detection and diagnosis in aerospace systems using analytical redundancy", ***IEEE Computing and Control Journal***, 127-136, (1991).

[34] Mendonça, L., F., Sousa, J., M., Costa, J., M., G., "Fuzzy Model-Based Fault Detection and Isolation", ***IEEE Emerging Technologies and Factory Automation Proceedings***, Vol.2***,*** 768-774, (2003).

[35] Chow, E., Y., Willsky, A., S., "Analytical Redundancy and the Design of Robust Failure Detection Systems", ***IEEE Transaction On Automatic Control***, Volume: 29-7, 603-614, (1984).

[36] Saberi, A., Stoorvogel, A., A., Sannuti, P., Nieman, H., "Fundamental problems in fault detection and identification", ***International Journal of Robust and Nonlinear Control***, Vol.10-14, 1209-1236, (2000).

[37] Nakkabi, Y., Kabbaj, N., Dahhou, B., Roux, G., Aguilar-Martin, J., "A combined analytical and knowledge based method for fault detection and isolation", ***IEEE Conference on Emergin Technologies and Factory Automation***, Lisbonne , Vol 2, 161-164, (2003).

[38] Isermann, R., Model Based Fault Detection and Diagnosis Methods, ***Proceedings of the American Control Conference***, Seatle, 1605-1609, (1995).

[39] Xu, Z., "Design of Knowledge-based Fault Detection and Identification for Dynamical Systems" Yüksek Lisans Tezi, ***University of Alberta***, Canada, (2002).

[40] Isermann, R., "On fuzzy logic applications for automatic control, supervision, and fault diagnosis", ***IEEE Transactions on Systems, Man, and Cybernetics- Part A: Systems and Humans***, Vol. 28-2, 221-235, (1998).

[41] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., N., Yin, K., "A review of process fault detection and diagnosis part III: Process history-based methods", ***Computers and Chemical Engineering***, Vol.27, 327-346, (2003).

[42] Betta, G., ,Liguori, C., Pietrosanto, A., "Comparison between IFDI schemes based on expert systems and neural networks", ***IEEE Transactions on Instrumentation and Measurement***,Vol. 47-5, 1106 -1111, (1998).

[43] Angeli, C., "Online Expert Systems for Fault Diagnosis in Technical Processes", ***Expert Systems***, Vol.25-2, 115-132, (2008).

[44] Berry, M., J., A., Linoff, G.,. Data Mining Techniques: For Marketing, Sales and Customer Support, Second Edition, ***Wiley Publishing***, ,165-209, (1997).

[45] Introduction to Data Mining and Knowledge Discovery [online], Third Edition, Two Crows Corporation, USA, http://www.twocrows.com/intro-dm.pdf, (**Ziyaret Tarihi: 30 Mayıs 2011**).

[46] Mendonca, M., Sunderhaft, N., A State-of-the-Art-Report Mining Software Engineering Data: A Survey, ***DACS-SOAR-99-3***, ***DoD Data & Analysis Center for Software (DACS)***, (1999).

[47] Fayyad, U., M., "Data mining and knowledge discovery: making sense out of data" **IEEE Expert**, 20-25, (1996).

[48] Fayyad, U., M., "Data Mining and Knowledge Discovery in Databases: Implications for Scientific Databases, ***IEEE Ninth International Conference on Scientific and Statistical Database Management***, Washington, 2-11(1997).

[49] Sever, H., Oğuz, B., "Veritabanlarında Bilgi Keşfine Formal Bir Yaklaşım. Kısım I: Eşleştirme Sorguları ve Algoritmalar", ***Bilgi Dünyası***, 173-204, (1999).

[50] Akpınar, H., "Veri Tabanlarında Bilgi Keşfi ve Veri Madenciliği", ***İ.Ü. İşletme Fakültesi Dergisi***, Cilt. 29-1, 1-22, (2000).

[51] Simoudis, E., Reality Check for Data Mining. ***IEEE Expert: Intelligent Systems and Their Applications***, Vol. 11-5, 26-33, (1996).

[52] Agrawal, R., Srikant, R., "Fast algorithms for mining association rules", ***Proceedings of the 20th VLDB conference***, Santiago, 487-499, (1994).

[53] Han,J., Kamber,M., "Data Mining: Concepts and Techniques", ***Morgan Kauffmann Publishers***, San Francisco, (2000).

[54] Zaki, M., J., "Parallel and Distributed Association Mining: A Survey", ***IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining***, Vol.7-5, 14-25, (1999).

[55] Altınışık, U., "Öğrenci Bilgi Sisteminde Veri Madenciliğinin Uygulanması", Yüksek Lisans Tezi, ***Kocaeli Üniversitesi Fen Bilimleri Enstitüsü***, İzmit, (2006).

[56] Houtsma, M., Swami, A., "Set oriented mining for association rules in relational databases", ***IEEE Proceedings of the Eleventh International Conference on Data Engineering***, Taipei, 25-34, (1995).

[57] Savasere, A., Omiecinski, E., Navathe, S., "An Efficient Algorithm for Mining Association Rules in Large Databases", ***Proceedings 21st International Conference Very Large Databases***, Zurich, 432-444, (1995).

[58] Brin, S., Motwani, R., Ullman, J., D., Tsur, S., "Dynamic Itemset Counting and Implication Rules for Market Basket Data", ***Proceedings ACM SIGMOD International Conference on Management of Data***, Tucson, Arizona, 255-264, (1997).

[59] Zaki, M., J., Ogihara, M., Parthasarathy, S., Li, W., "Parallel Algorithms for Fast Discovery of Association Rules", ***Data Mining and Knowledge Discovery: An International Journal***, Vol. 1-4, 343-373, (1997).

[60] Weiss, S., M., Kulikowski, C., A., "Computer Systems that learn: Classification and Prediction Methods from Statistics, Neural Net, Machine Learning, and Expert Systems, ***Morgan Kauffman***, San Mateo, (1991).

[61] Bayes, T., An Essay Toward Solving a Problem in the Doctrine of Chances. ***Philosophical Transactions of the Royal Society***, London, Vol. 53, 370-418, (1764).

[62] Bigus, J., P., "Data Mining with Neural Networks: Solving Business Problems – From Application Development to Decision Support", ***McGraw Hill***, (1996).

[63] Kohonen, T., "The Self Organizing Map", ***Proceedings of IEEE***, Vol 78-9, (1990).

[64] Vesanto, J., Alhoniemi, E., "Clustering of the Self-Organizing Map", ***IEEE Transaction on Neural Networks***, Vol. 11-3, 586-600, (2000).

[65] Tan, K., C., Li, Y., Murray-Smith, D., J., Sharman, K., C., "System Identification and Linearisation Using Genetic Algorithms with Simulated annealing", ***Proceedings of first IEE/IEEE International Conference on Genetic Algorithms in Engineering System: Innovations and Applications***, Sheffield, UK, 164-169, (2000).

[66] Kristinsson, K., Dumont, G., A., "System Identification and Control Using Genetic Algorithms", ***IEEE Transaction on Systems, Man and Cybernetics***, Vol. 22-5, 1033-1046, (1992).

[67] Yıldırım, M., "Genetik Algoritmalar ve Benzetilmiş Tavlama ile Uzun Dönem Üretim Genişletme Planlaması", Doktora Tezi, ***Kocaeli Üniversitesi Fen Bilimleri Enstitüsü***, İzmit, (2003).

[68] Kaufman, L., Rousseeuw, P., J., "Clustering by means of medoids", ***Statistical Data Analysis Based on The L1–Norm and Related Methods***, 405–416, (1987).

[69] Jain, A., K., Dubes, R.,C., "Algorithms for Clustering Data". ***Prentice Hall***, (1988).

[70] Alpdoğan, Y., "Kendinden Düzenlenen Haritalar ile Doküman Sınıflandırma", Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, (2007).

[71] Alsabti, K., Ranka, S., Singh, V, "An Efficient K-means Clustering Algorithm", ***IPPS/SPDP Workshop on High Performance Data Mining***, **IEEE Computer Society Press**, Orlando, Florida, 125-130 (1998).

[72] Dasgupta, S., Long, P., M., "Performance guarantees for hierarchical clustering", ***Journal of Computer System Sciences*** , Vol. 70, 555-569, (2005).

[73] Gonzalez, T., F., "Clustering to minimize the maximum intercluster distance", ***Theoretical Computer Sciences*** , Vol. 38, 293-306, (1985).

[74] Postalcıoğlu, S., "İşaret İşleme ve Yapay Zeka Tabanlı Arıza Dayanımlı Denetleyici Tasarımı", Doktora Tezi, ***Kocaeli Üniversitesi Fen Bilimleri Enstitüsü***, İzmit, (2007).

[75] Dumanay, A., V., "PID, Bulanık Mantık ve Kayan Kip Kontrol Yöntemleri ile İnternet Üzerinden DC Motor Hız Kontrolü", Yüksek Lisans Tezi, ***Balıkesir Üniversitesi Fen Bilimleri Enstitüsü***, Balıkesir, (2009).

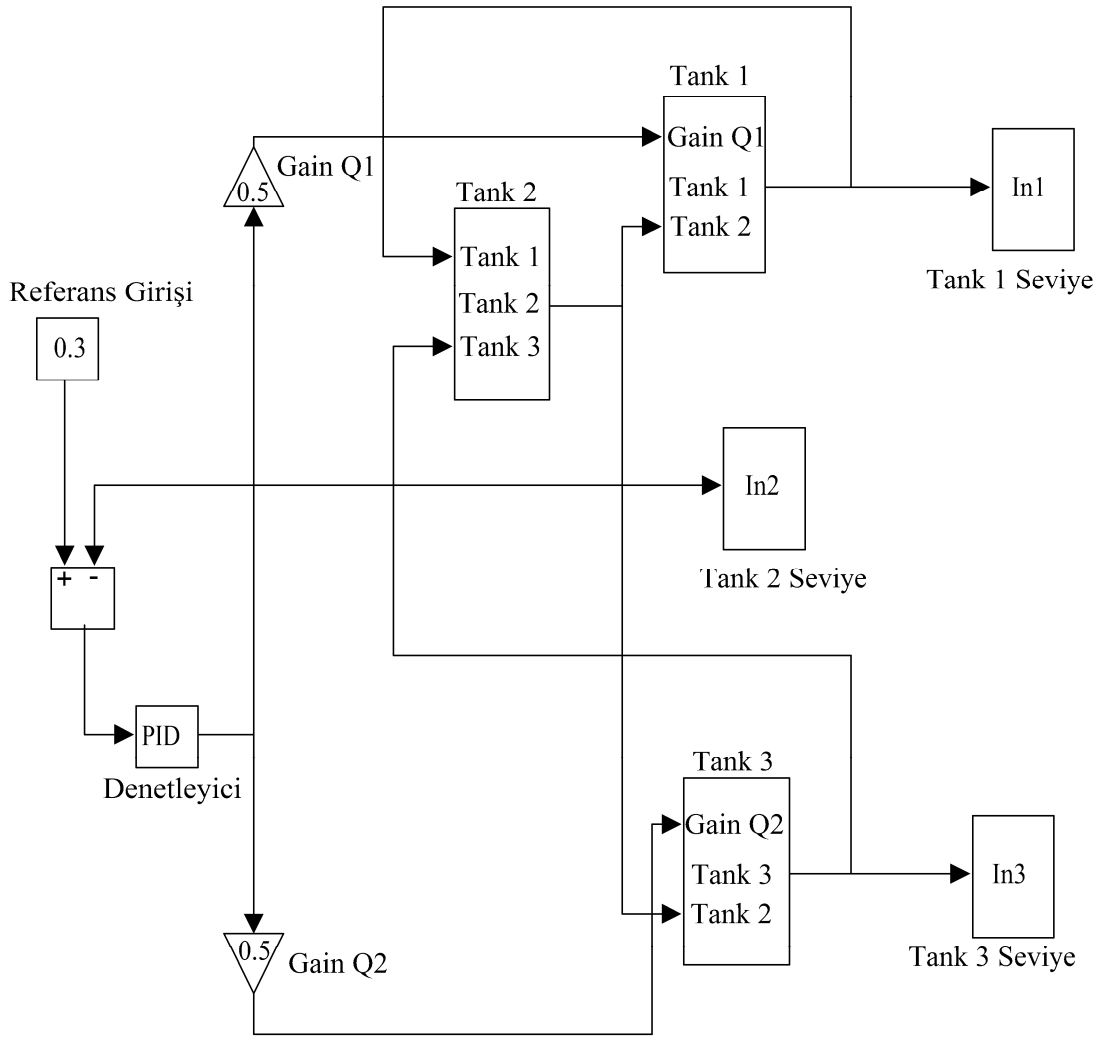
[76] Manoj, G., "Proportional-Integral-Derivative (PID) Controller Design for Robust Stability of Arbitrary Order Plants with Time-Delay and Additive Uncertainty", Yüksek Lisans Tezi, ***Wichita State University***, Kansas, (2010).

[77] Zhao, Q., Xu, Z.,“Design of a novel knowledge-based fault detection and isolation scheme”, ***IEEE Trans on Systems Man and Cybernetics—Part B: Cybernetics***, Vol. 34-2, 1089-1095, (2004).

EKLER

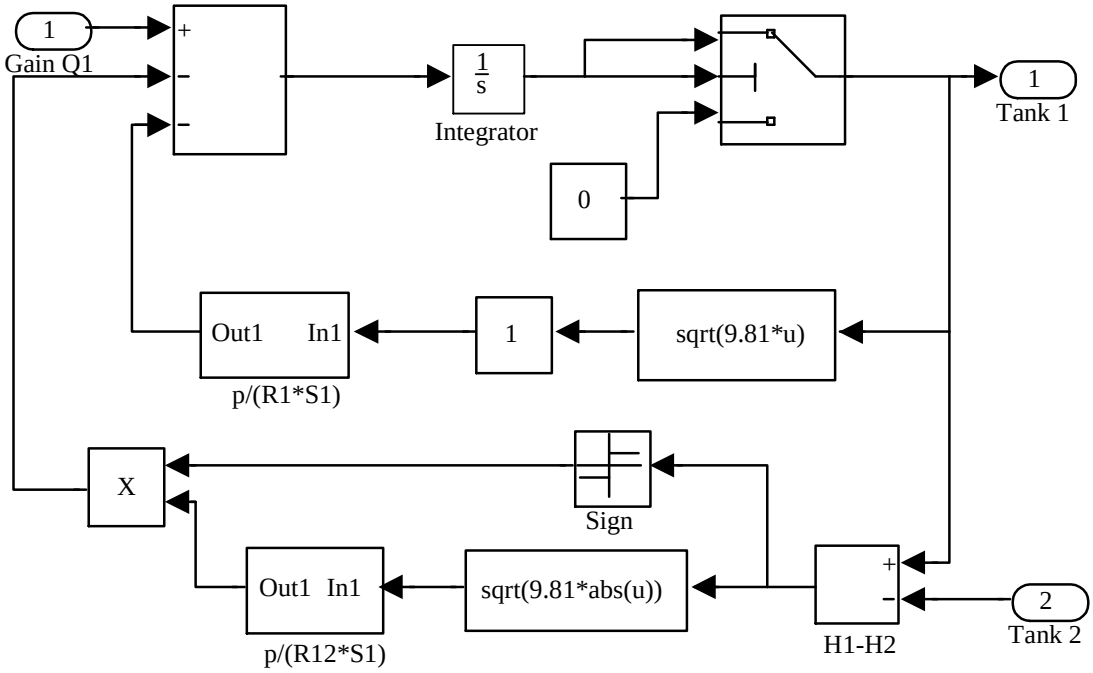
Ek-A. MATLAB - SIMULINK ŞEMALARI

Üçlü Tank Sisteminin Matlab Simulink modeli genel olarak Şekil A.1 verilmiştir.



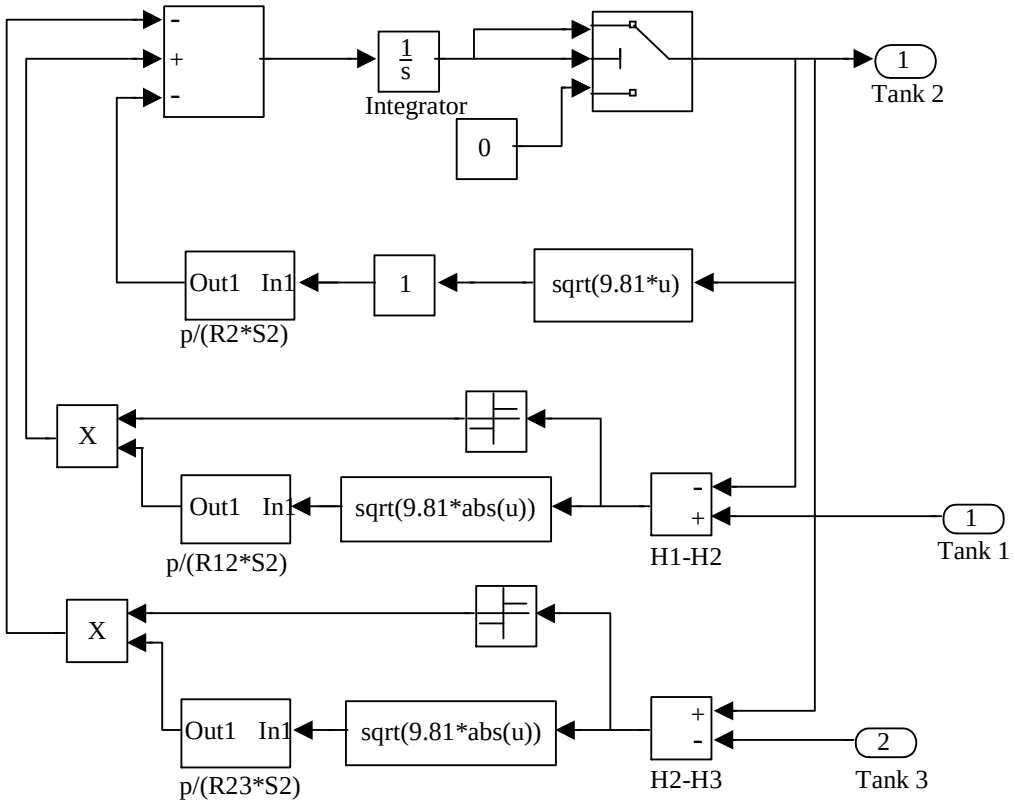
Şekil A.1: Üçlü tank sistemi matlab-simulink modeli

Tank-1’ in Matlab-Simulink modeli Şekil A.2’de verilmiştir.



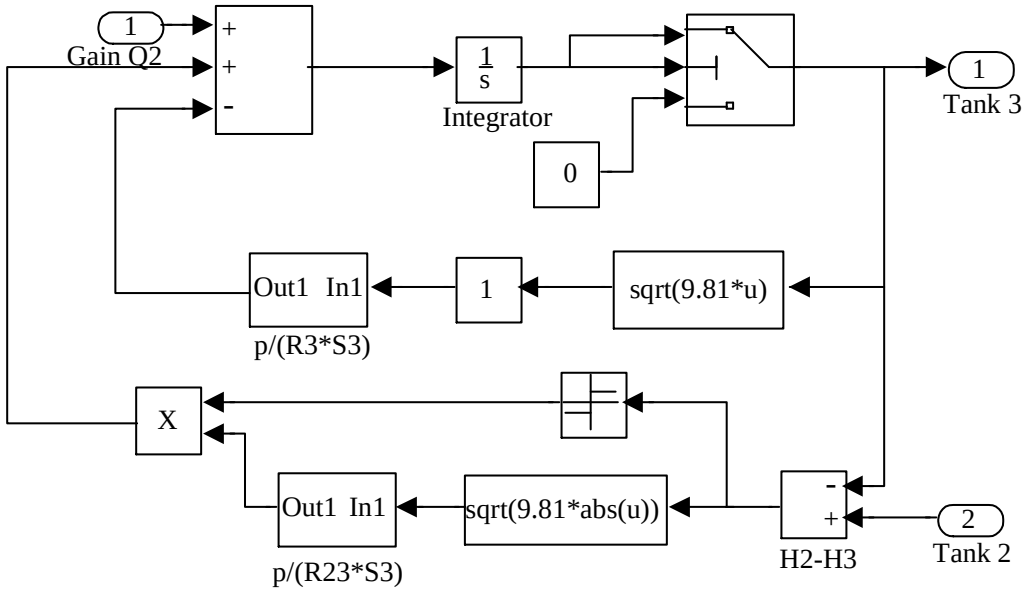
Şekil A.2: Tank-1 matlab-simulink modeli

Tank-2’ in Matlab-Simulink modeli Şekil A.3’te verilmiştir.



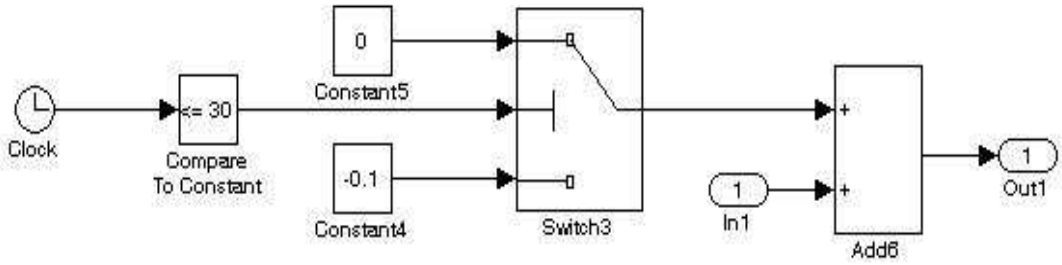
Şekil A.3: Tank-2 matlab-simulink modeli

Tank-3' ün Matlab-Simulink modeli Şekil A.4'te verilmiştir.



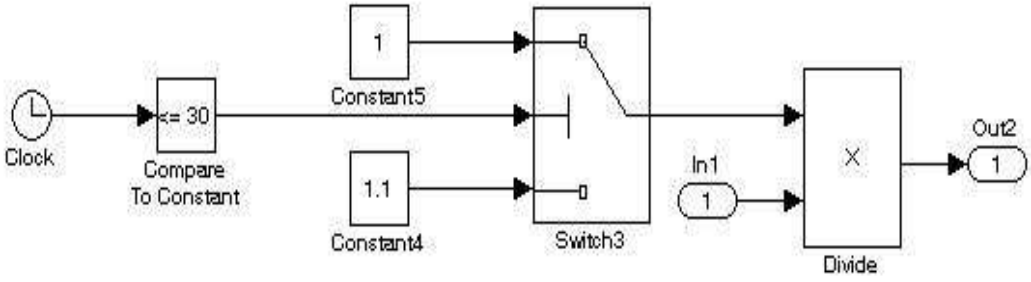
Şekil A.4: Tank-3 matlab-simulink modeli

1 nolu arıza Matlab-Simulink modeli Şekil A.5'te verilmiştir.



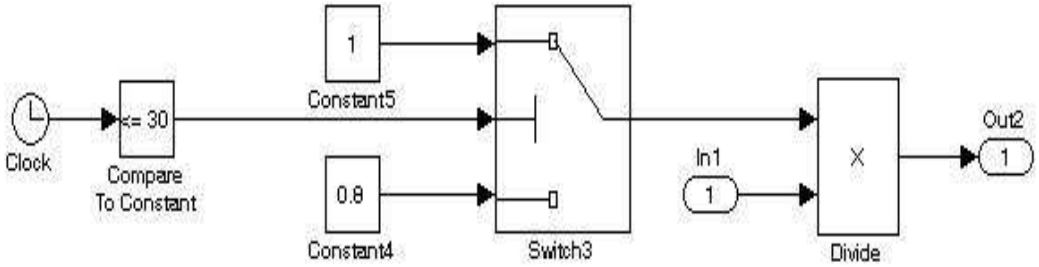
Şekil A.5: Arıza-1 matlab-simulink modeli

2 nolu arıza Matlab-Simulink modeli Şekil A.6'da verilmiştir.



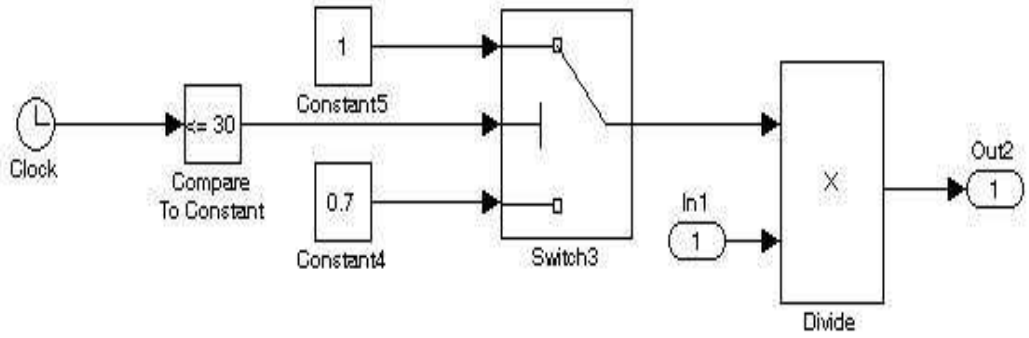
Şekil A.6: Arıza-2 matlab-simulink modeli

3 nolu arıza Matlab-Simulink modeli Şekil A.7’de verilmiştir.



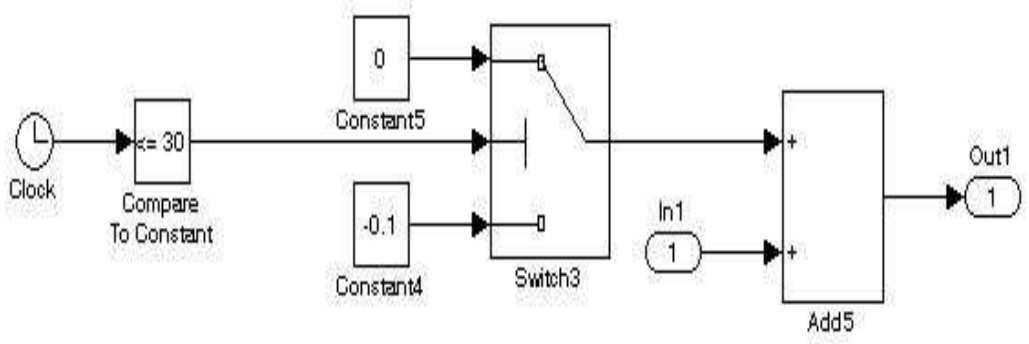
Şekil A.7: Arıza-3 matlab-simulink modeli

4 nolu arıza Matlab-Simulink modeli Şekil A.8’de verilmiştir.



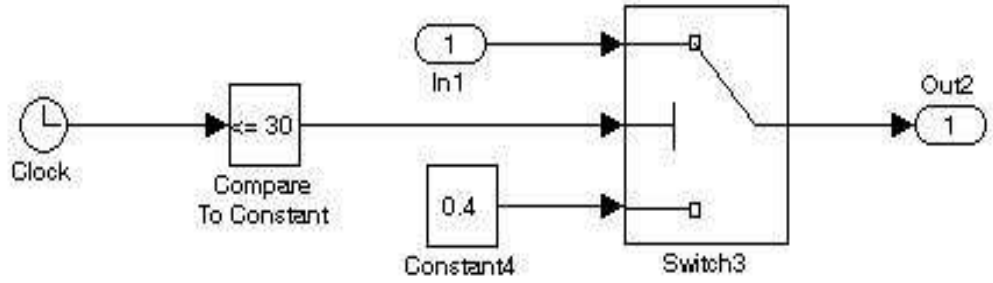
Şekil A.8: Arıza-4 matlab-simulink modeli

5 nolu arıza Matlab-Simulink modeli Şekil A.9’da verilmiştir.



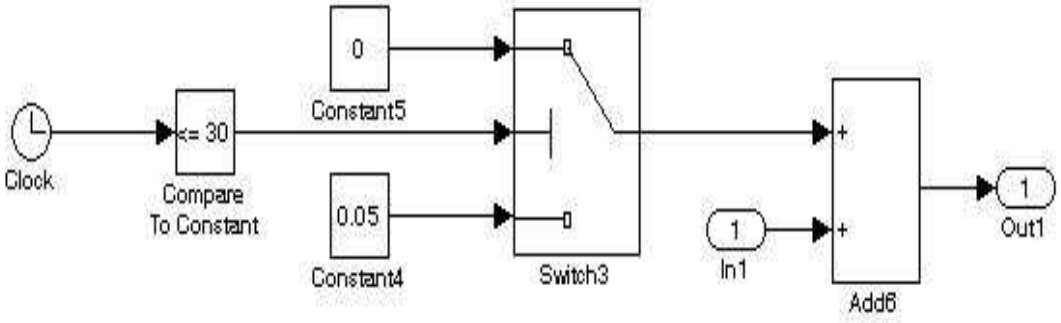
Şekil A.9: Arıza-5 matlab-simulink modeli

6 nolu arıza Matlab-Simulink modeli Şekil A.10'da verilmiştir.



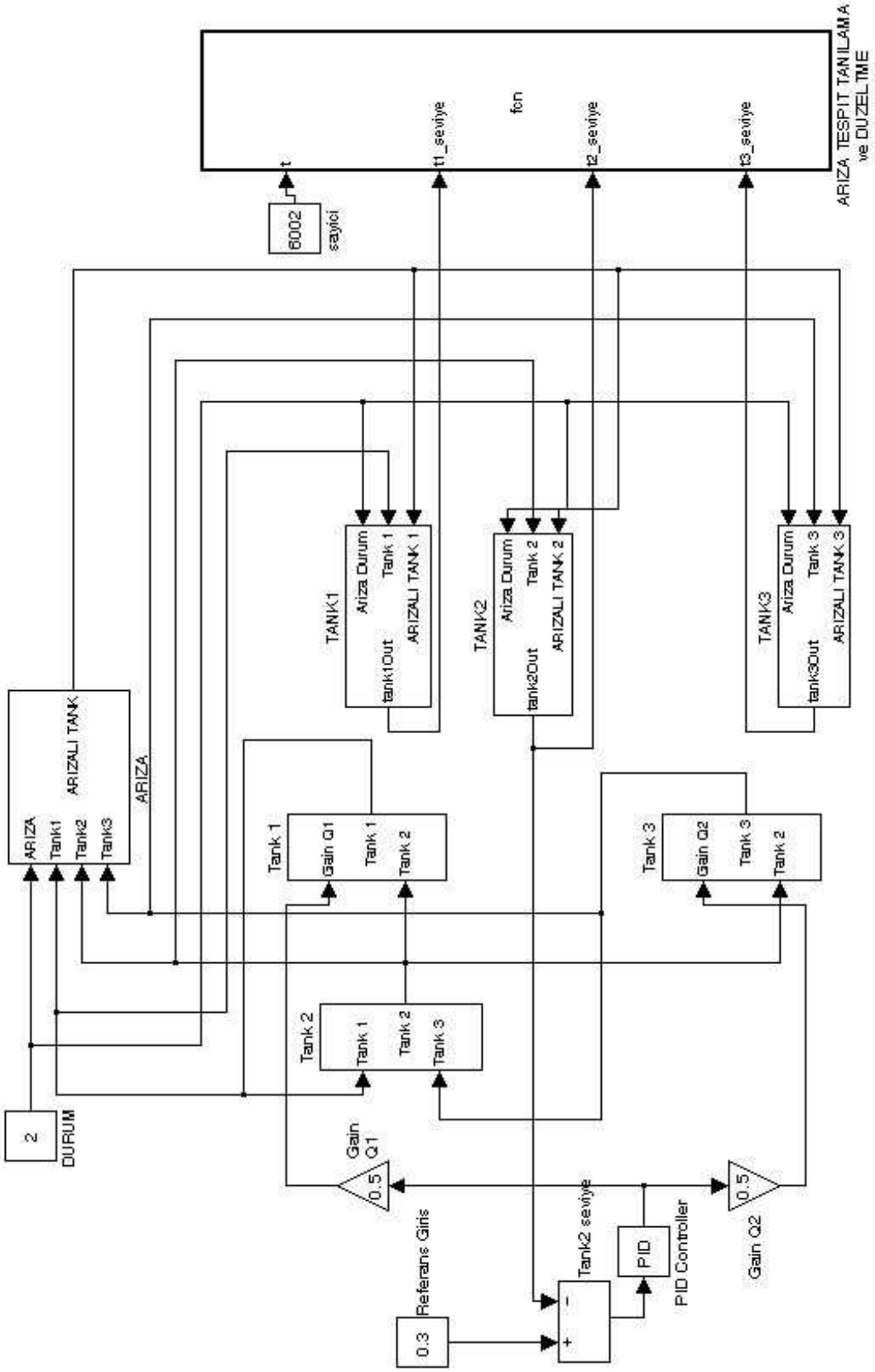
Şekil A.10: Arıza-6 matlab-simulink modeli

7 nolu arıza Matlab-Simulink modeli Şekil A.11'de verilmiştir.



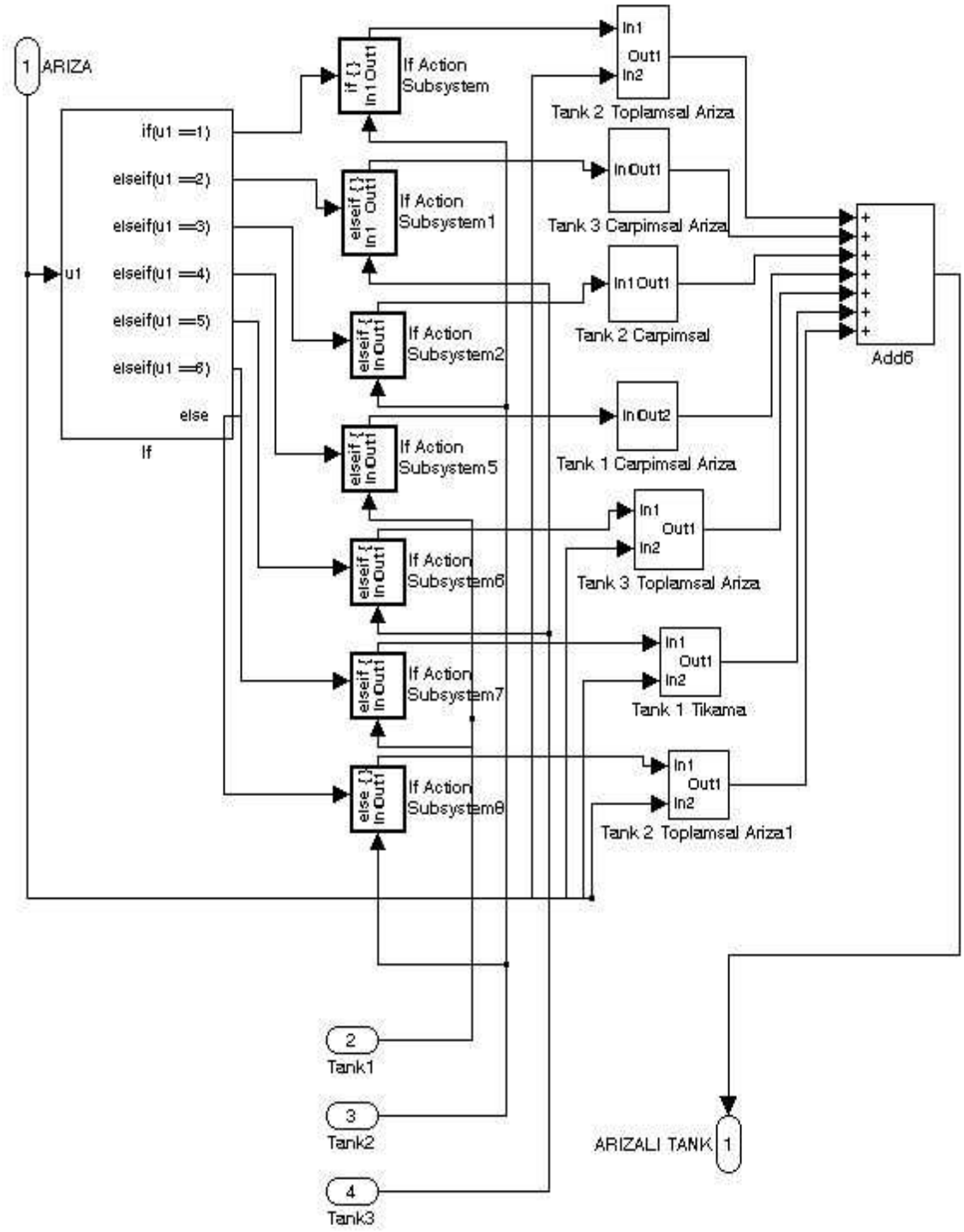
Şekil A.11: Arıza-7 matlab-simulink modeli

Arıza tespit ve tanılama Matlab-Simulink Modeli Şekil A.12’de verilmiştir.



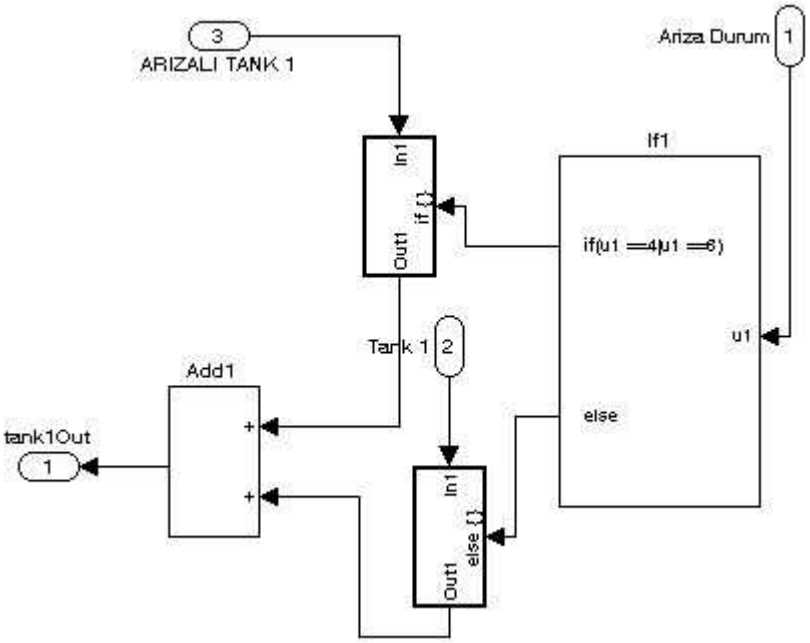
Şekil A.12: Arıza tespit ve tanılama matlab-simulink modeli

Arıza Seçim Matlab-Simulink Modeli Şekil A.13'te verilmiştir.



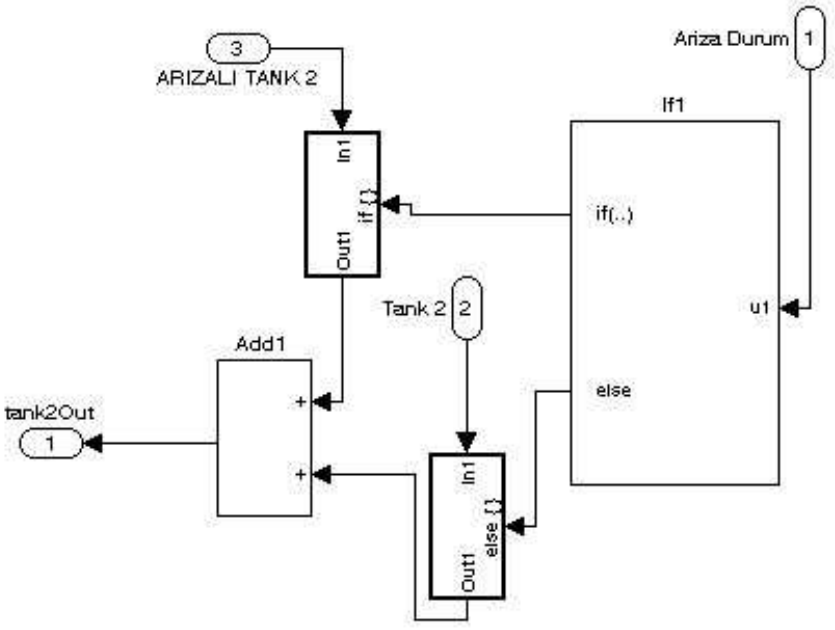
Şekil A.13: Arıza seçim matlab-simulink modeli

Arızalı Tank-1 Matlab-Simulink Modeli Şekil A.14'de verilmiştir.



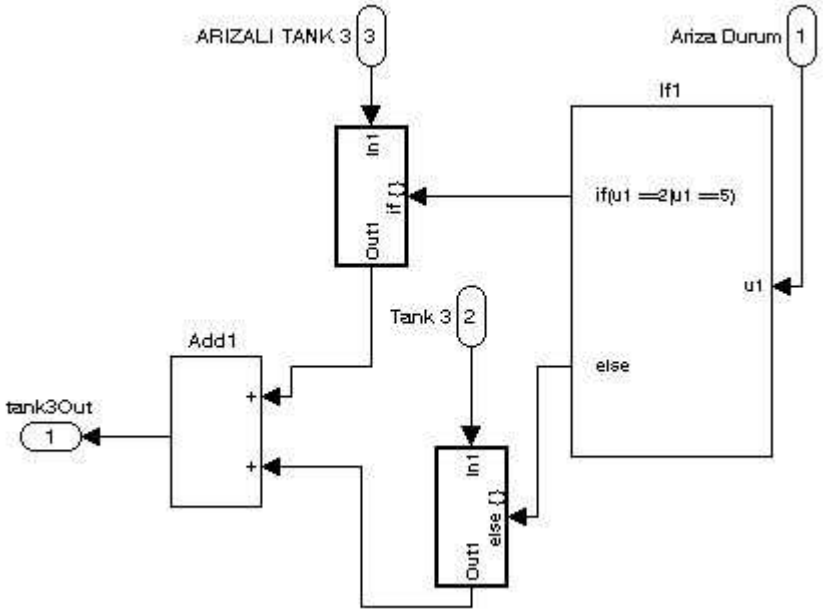
Şekil A.14: Arızalı tank-1 matlab-simulink modeli

Arızalı Tank-2 Matlab-Simulink Modeli Şekil A.15'te verilmiştir.



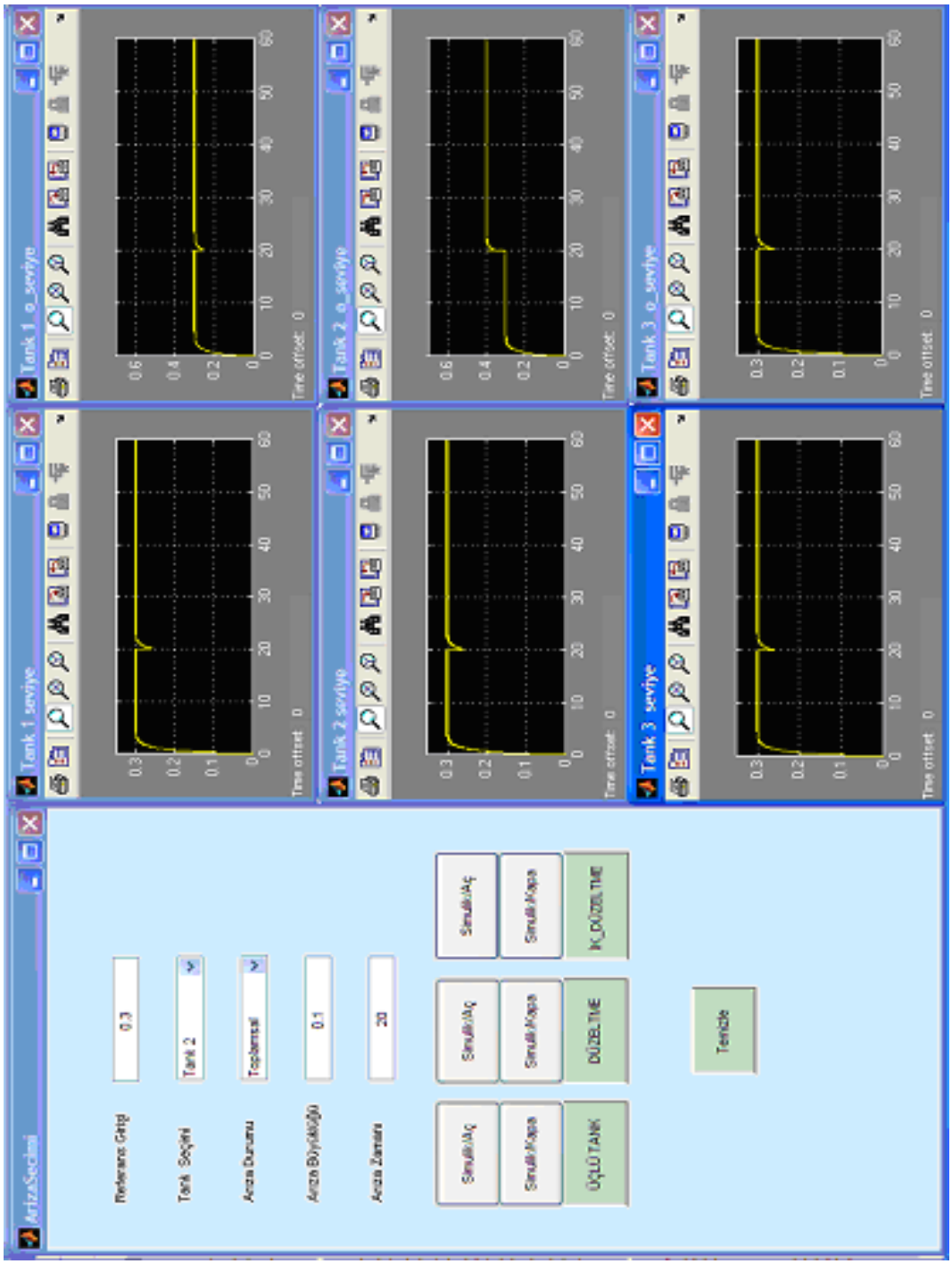
Şekil A.15: Arızalı tank-2 matlab-simulink modeli

Arızalı Tank-3 Matlab-Simulink Modeli Şekil A.16’da verilmiştir.



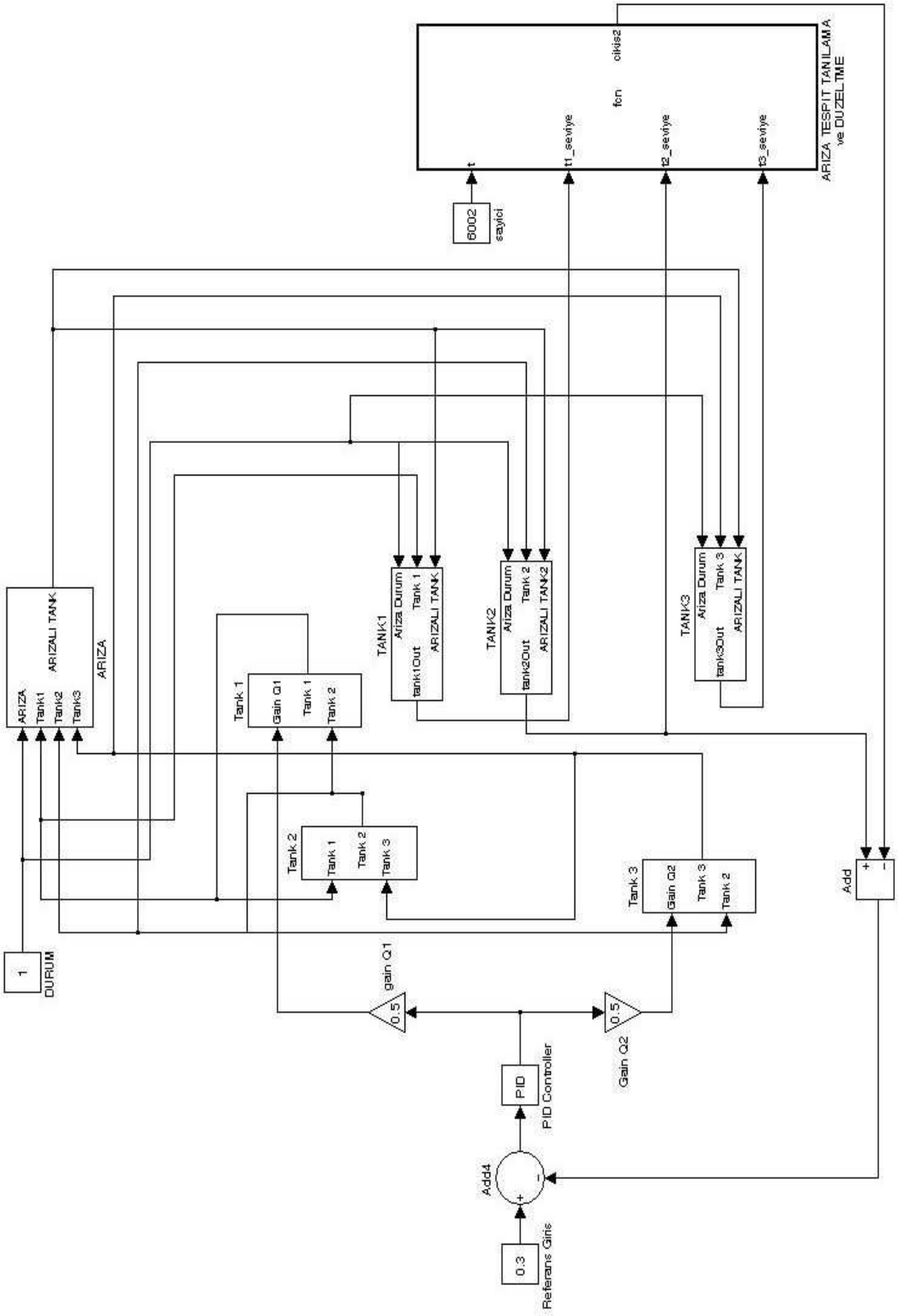
Şekil A.16: Arızalı tank-3 matlab-simulink modeli

Arıza seçimi Matlab Gui Programı Şekil A.17’da verilmiştir.



Şekil A.17: Arıza seçimi matlab-gui programı

Arıza tespit tanılama ve düzeltme Matlab-Simulink Modeli Şekil A.18'de verilmiştir.



Şekil A.18: Arıza tespit tanılama ve düzeltme matlab-simulink modeli

Ek-B. PROGRAM LİSTESİ

```
%BİLGİ TABANIN OLUŞTURULMASI ve PENCERELENMESİ %
pengen=60;ozgen=20;

%Normal Durum%

load ../DATALAR/normal/t1os;
[t1pen]=pencere(t1os,pengen,ozgen);
[t1penvar]=pen_var(t1pen);
t1r1=detay(t1penvar);
t1r2=yaklasim(t1pen,pengen,ozgen);
load ../DATALAR/normal/t2os;
[t2pen]=pencere(t2os,pengen,ozgen);
[t2penvar]=pen_var(t2pen);
t2r1=detay(t2penvar);
t2r2=yaklasim(t2pen,pengen,ozgen);
load ../DATALAR/normal/t3os;
[t3pen]=pencere(t3os,pengen,ozgen);
[t3penvar]=pen_var(t3pen);
t3r1=detay(t3penvar);
t3r2=yaklasim(t3pen,pengen,ozgen);
load ../DATALAR/normal/k;
[kpen]=pencere(k,pengen,ozgen);
[kpenvar]=pen_var(kpen);
kr1=detay(kpenvar);
kr2=yaklasim(kpen,pengen,ozgen);

%Arıza-1 Durumu%

load ../DATALAR/ariza1/a1t1os;
[a1t1pen]=pencere(a1t1os,pengen,ozgen);
[a1t1penvar]=pen_var(a1t1pen);
a1t1r1=detay(a1t1penvar);
a1t1r2=yaklasim(a1t1pen,pengen,ozgen);
load ../DATALAR/ariza1/a1t2os;
[a1t2pen]=pencere(a1t2os,pengen,ozgen);
[a1t2penvar]=pen_var(a1t2pen);
a1t2r1=detay(a1t2penvar);
a1t2r2=yaklasim(a1t2pen,pengen,ozgen);
load ../DATALAR/ariza1/a1t3os;
[a1t3pen]=pencere(a1t3os,pengen,ozgen);
[a1t3penvar]=pen_var(a1t3pen);
a1t3r1=detay(a1t3penvar);
a1t3r2=yaklasim(a1t3pen,pengen,ozgen);
load ../DATALAR/ariza1/a1k;
[a1kpen]=pencere(a1k,pengen,ozgen);
[a1kpenvar]=pen_var(a1kpen);
```

```
a1kr1=detay(a1kpenvar);
a1kr2=yaklasim(a1kpen,pengen,ozgen);

% Arıza-2 Durumu %

load ../DATALAR/ariza2/a2t1os;
[a2t1pen]=pencere(a2t1os,pengen,ozgen);
[a2t1penvar]=pen_var(a2t1pen);
a2t1r1=detay(a2t1penvar);
a2t1r2=yaklasim(a2t1pen,pengen,ozgen);
load ../DATALAR/ariza2/a2t2os;
[a2t2pen]=pencere(a2t2os,pengen,ozgen);
[a2t2penvar]=pen_var(a2t2pen);
a2t2r1=detay(a2t2penvar);
a2t2r2=yaklasim(a2t2pen,pengen,ozgen);
load ../DATALAR/ariza2/a2t3os;
[a2t3pen]=pencere(a2t3os,pengen,ozgen);
[a2t3penvar]=pen_var(a2t3pen);
a2t3r1=detay(a2t3penvar);
a2t3r2=yaklasim(a2t3pen,pengen,ozgen);
load ../DATALAR/ariza2/a2k;
[a2kpen]=pencere(a2k,pengen,ozgen);
[a2kpenvar]=pen_var(a2kpen);
a2kr1=detay(a2kpenvar);
a2kr2=yaklasim(a2kpen,pengen,ozgen);

% Arıza-3 Durumu %

load ../DATALAR/ariza3/a3t1os;
[a3t1pen]=pencere(a3t1os,pengen,ozgen);
[a3t1penvar]=pen_var(a3t1pen);
a3t1r1=detay(a3t1penvar);
a3t1r2=yaklasim(a3t1pen,pengen,ozgen);
load ../DATALAR/ariza3/a3t2os;
[a3t2pen]=pencere(a3t2os,pengen,ozgen);
[a3t2penvar]=pen_var(a3t2pen);
a3t2r1=detay(a3t2penvar);
a3t2r2=yaklasim(a3t2pen,pengen,ozgen);
load ../DATALAR/ariza3/a3t3os;
[a3t3pen]=pencere(a3t3os,pengen,ozgen);
[a3t3penvar]=pen_var(a3t3pen);
a3t3r1=detay(a3t3penvar);
a3t3r2=yaklasim(a3t3pen,pengen,ozgen);
load ../DATALAR/ariza3/a3k;
[a3kpen]=pencere(a3k,pengen,ozgen);
[a3kpenvar]=pen_var(a3kpen);
a3kr1=detay(a3kpenvar);
a3kr2=yaklasim(a3kpen,pengen,ozgen);
```

```
% Arıza-4 Durumu %

load ../DATALAR/ariza4/a4t1os;
[a4t1pen]=pencere(a4t1os,pengen,ozgen);
[a4t1penvar]=pen_var(a4t1pen);
a4t1r1=detay(a4t1penvar);
a4t1r2=yaklasim(a4t1pen,pengen,ozgen);
load ../DATALAR/ariza4/a4t2os;
[a4t2pen]=pencere(a4t2os,pengen,ozgen);
[a4t2penvar]=pen_var(a4t2pen);
a4t2r1=detay(a4t2penvar);
a4t2r2=yaklasim(a4t2pen,pengen,ozgen);
load ../DATALAR/ariza4/a4t3os;
[a4t3pen]=pencere(a4t3os,pengen,ozgen);
[a4t3penvar]=pen_var(a4t3pen);
a4t3r1=detay(a4t3penvar);
a4t3r2=yaklasim(a4t3pen,pengen,ozgen);
load ../DATALAR/ariza4/a4k;
[a4kpen]=pencere(a4k,pengen,ozgen);
[a4kpenvar]=pen_var(a4kpen);
a4kr1=detay(a4kpenvar);
a4kr2=yaklasim(a4kpen,pengen,ozgen);

% Arıza-5 Durumu %

load ../DATALAR/ariza5/a5t1os;
[a5t1pen]=pencere(a5t1os,pengen,ozgen);
[a5t1penvar]=pen_var(a5t1pen);
a5t1r1=detay(a5t1penvar);
a5t1r2=yaklasim(a5t1pen,pengen,ozgen);
load ../DATALAR/ariza5/a5t2os;
[a5t2pen]=pencere(a5t2os,pengen,ozgen);
[a5t2penvar]=pen_var(a5t2pen);
a5t2r1=detay(a5t2penvar);
a5t2r2=yaklasim(a5t2pen,pengen,ozgen);
load ../DATALAR/ariza5/a5t3os;
[a5t3pen]=pencere(a5t3os,pengen,ozgen);
[a5t3penvar]=pen_var(a5t3pen);
a5t3r1=detay(a5t3penvar);
a5t3r2=yaklasim(a5t3pen,pengen,ozgen);
load ../DATALAR/ariza5/a5k;
[a5kpen]=pencere(a5k,pengen,ozgen);
[a5kpenvar]=pen_var(a5kpen);
a5kr1=detay(a5kpenvar);
a5kr2=yaklasim(a5kpen,pengen,ozgen);

% Arıza-6 Durumu %

load ../DATALAR/ariza6/a6t1os;
```

```
[a6t1pen]=pencere(a6t1os,pengen,ozgen);
[a6t1penvar]=pen_var(a6t1pen);
a6t1r1=detay(a6t1penvar);
a6t1r2=yaklasim(a6t1pen,pengen,ozgen);
load ../DATALAR/ariza6/a6t2os;
[a6t2pen]=pencere(a6t2os,pengen,ozgen);
[a6t2penvar]=pen_var(a6t2pen);
a6t2r1=detay(a6t2penvar);
a6t2r2=yaklasim(a6t2pen,pengen,ozgen);
plot(a6t2pen(149,1:60))
load ../DATALAR/ariza6/a6t3os;
[a6t3pen]=pencere(a6t3os,pengen,ozgen);
[a6t3penvar]=pen_var(a6t3pen);
a6t3r1=detay(a6t3penvar);
a6t3r2=yaklasim(a6t3pen,pengen,ozgen);
load ../DATALAR/ariza6/a6k;
[a6kpen]=pencere(a6k,pengen,ozgen);
[a6kpenvar]=pen_var(a6kpen);
a6kr1=detay(a6kpenvar);
a6kr2=yaklasim(a6kpen,pengen,ozgen);

% Arıza-7 Durumu %

load ../DATALAR/ariza7/a7t1os;
[a7t1pen]=pencere(a7t1os,pengen,ozgen);
[a7t1penvar]=pen_var(a7t1pen);
a7t1r1=detay(a7t1penvar);
a7t1r2=yaklasim(a7t1pen,pengen,ozgen);
load ../DATALAR/ariza7/a7t2os;
[a7t2pen]=pencere(a7t2os,pengen,ozgen);
[a7t2penvar]=pen_var(a7t2pen);
a7t2r1=detay(a7t2penvar);
a7t2r2=yaklasim(a7t2pen,pengen,ozgen);
load ../DATALAR/ariza7/a7t3os;
[a7t3pen]=pencere(a7t3os,pengen,ozgen);
[a7t3penvar]=pen_var(a7t3pen);
a7t3r1=detay(a7t3penvar);
a7t3r2=yaklasim(a7t3pen,pengen,ozgen);
load ../DATALAR/ariza7/a7k;
[a7kpen]=pencere(a7k,pengen,ozgen);
[a7kpenvar]=pen_var(a7kpen);
a7kr1=detay(a7kpenvar);
a7kr2=yaklasim(a7kpen,pengen,ozgen);

%Pencere varyasyonlarının hesaplanması fonksiyonu%

function [penvar]=pen_var(pen)
[n m]=size(pen);
```

```
for i=1:n,
penvar(i,1)=var(pen(i,:));
end;
```

%Pencere fonksiyonu%

```
function [pen]=pencere(data,pengen,ozgen)
[n m]=size(data);
maxpen=floor((m-pengen)/ozgen);
for i=1:maxpen,
for j=1:pengen,
  pen(i,j)=data(2,((i-1)*ozgen)+j);
end;
end;
```

%Detay fonksiyonu%

```
function [kdetay]=detay(penvar)
[n m]=size(penvar);
for i=2:n,
  kdetay(i,1)=(penvar(i,1)-penvar(i-1,1))./penvar(i-1,1);
end;
```

%Weka tarafından tank değerlerinin alınması%

```
@relation
@attribute Tank1 numeric
@attribute Tank2 numeric
@attribute Tank3 numeric
@attribute Tank1G numeric
@attribute Tank2G numeric
@attribute Tank3G numeric
@data
```

%Arıza Tespit ve Tanılama modülü%

```
function cikis1  = fcn(t,t1_seviye,t2_seviye,t3_seviye)
persistent cikis_var;
persistent veriler1;
persistent veriler2;
persistent veriler3;
persistent baslangic;
persistent bitis;
persistent hesap1;
persistent hesap2;
persistent hesap3;
persistent k;
```

```
persistent ariza_kume;
persistent kume_mesafe;
persistent duzeltme;
persistent ck_1;
persistent ck_2;
persistent ck_3;
ariza_kume=[1573.542473 304.395695 1573.542473 0 10000 0;2514.058942
472.9565604 2514.058942 0 10000 0;639.0326643 126.7905281 639.0326643 0
10000 0;0 0 1116.304955 0 0 10000;4450.8918 0 0 10000 0 0;0 0 19233.84383 0 0
10000;17813.11921 0 0 10000 0 0];
kume_mesafe=[0;0;0;0;0;0;0];
varyans1=0;varyans1_2=0;
varyans2=0;varyans2_2=0;
varyans3=0;varyans3_2=0;
if isempty(ck_1)
   ck_1=0;
end
if isempty(ck_2)
   ck_2=0;
end
if isempty(ck_3)
   ck_3=0;
end
if isempty(hesap1)
   hesap1=0;
end
if isempty(duzeltme)
   duzeltme=0;
end
if isempty(k)
   k=0;
end
if isempty(hesap2)
   hesap2=0;
end
if isempty(hesap3)
   hesap3=0;
end
if isempty(baslangic)
   baslangic=21;
end
if isempty(bitis)
   bitis=80;
end
if isempty(cikis_var)
   cikis_var=0;
end
if isempty(veriler1)
   veriler1=zeros(1,9002);
```

```
end
if isempty(veriler2)
   veriler2=zeros(1,9002);
end
if isempty(veriler3)
   veriler3=zeros(1,9002);
end
veriler1(t)=t1_seviye;
veriler2(t)=t2_seviye;
veriler3(t)=t3_seviye;
if ((t>=80) && (mod(t,20)==0))

% çevrim içi varyansın hesabi%

   n = 0;
   toplam1_1 = 0;
   ortalama1=0;
   toplam2_1 = 0;
   ortalama2=0;
   toplam3_1 = 0;
   ortalama3=0;
   for i=(baslangic:bitis)
      n = n + 1;
      toplam1_1 = toplam1_1 + veriler1(i);
      toplam2_1 = toplam2_1 + veriler2(i);
      toplam3_1 = toplam3_1 + veriler3(i);
   end
   ortalama1 = toplam1_1/n;
   toplam1_2 = 0;
   toplamc1 = 0;
   ortalama2 = toplam2_1/n;
   toplam2_2 = 0;
   toplamc2 = 0;
   ortalama3 = toplam3_1/n;
   toplam3_2 = 0;
   toplamc3 = 0;
   for i=(baslangic:bitis)
      toplam1_2 = toplam1_2 + (veriler1(i) - ortalama1)^2;
      toplamc1 = toplamc1 + (veriler1(i) - ortalama1);
      toplam2_2 = toplam2_2 + (veriler2(i) - ortalama2)^2;
      toplamc2 = toplamc2 + (veriler2(i) - ortalama2);
      toplam3_2 = toplam3_2 + (veriler3(i) - ortalama3)^2;
      toplamc3 = toplamc3 + (veriler3(i) - ortalama3);
   end
   varyans1 = (toplam1_2 - toplamc1^2/n)/(n - 1);
   varyans2 = (toplam2_2 - toplamc2^2/n)/(n - 1);
   varyans3 = (toplam3_2 - toplamc3^2/n)/(n - 1);
   n = 0;
   toplam1_1 = 0;
```

```
   ortalama1=0;
   toplam2_1 = 0;
   ortalama2=0;
   toplam3_1 = 0;
   ortalama3=0;
   for i=((baslangic-20):(bitis-20))
      n = n + 1;
      toplam1_1 = toplam1_1 + veriler1(i);
      toplam2_1 = toplam2_1 + veriler2(i);
      toplam3_1 = toplam3_1 + veriler3(i);
   end
   ortalama1 = toplam1_1/n;
   toplam1_2 = 0;
   toplamc1 = 0;
   ortalama2 = toplam2_1/n;
   toplam2_2 = 0;
   toplamc2 = 0;
   ortalama3 = toplam3_1/n;
   toplam3_2 = 0;
   toplamc3 = 0;
   for i=((baslangic-20):(bitis-20))
      toplam1_2 = toplam1_2 + (veriler1(i) - ortalama1)^2;
      toplamc1 = toplamc1 + (veriler1(i) - ortalama1);
      toplam2_2 = toplam2_2 + (veriler2(i) - ortalama2)^2;
      toplamc2 = toplamc2 + (veriler2(i) - ortalama2);
      toplam3_2 = toplam3_2 + (veriler3(i) - ortalama3)^2;
      toplamc3 = toplamc3 + (veriler3(i) - ortalama3);
   end
   varyans1_2 = (toplam1_2 - toplamc1^2/n)/(n - 1);
   varyans2_2 = (toplam2_2 - toplamc2^2/n)/(n - 1);
   varyans3_2 = (toplam3_2 - toplamc3^2/n)/(n - 1);
   hesap1=(varyans1-varyans1_2)/varyans1_2;
   hesap2=(varyans2-varyans2_2)/varyans2_2;
   hesap3=(varyans3-varyans3_2)/varyans3_2;
%varyans hesap sonu %
baslangic=baslangic+20;
end
set_param('uclutanksistemi_duzelt_ik/sayici', 'Value',int2str(t+1));

%Arızanın ait olduğu kümenin bulunması %

      if ((hesap1+hesap2+hesap3>10)& k==0)
        if (hesap1>10 & hesap2<10) ck_1=10000; end
        if (hesap3>10 & hesap2<10) ck_3=10000; end
        if (hesap2>10) ck_2=10000; end
        for j=(1:7)
        kume_mesafe(j)=((hesap1-ariza_kume(j,1))^2+(hesap2-
ariza_kume(j,2))^2+(hesap3-
```

```
        ariza_kume(j,3))^2+((ck_1)-ariza_kume(j,4))^2+((ck_2)-
ariza_kume(j,5))^2+((ck_3)-
        ariza_kume(j,6))^2)^(1/2);
        end
        [x y]=min(kume_mesafe);
        k=1;
        if (y==1)
         set_param('uclutanksistemi_duzelt_ik/ariza_cikis1', 'Value','1');
         duzeltme=-0.1;
        end
      if (y==2)
      set_param('uclutanksistemi_duzelt_ik/ariza_cikis2', 'Value','1');
      duzeltme=-0.125;
      end
      if (y==3)
      set_param('uclutanksistemi_duzelt_ik/ariza_cikis3', 'Value','1');
      duzeltme=0.06;
      end
      if (y==4)
      set_param('uclutanksistemi_duzelt_ik/ariza_cikis4', 'Value','1');
      end
      if (y==5)
      set_param('uclutanksistemi_duzelt_ik/ariza_cikis5', 'Value','1');
      end
      if (y==6)
      set_param('uclutanksistemi_duzelt_ik/ariza_cikis6', 'Value','1');
      end
      if (y==7)
      set_param('uclutanksistemi_duzelt_ik/ariza_cikis7', 'Value','1');
      duzeltme=+0.05
      end
      cikis1=duzeltme;
end

%Temizleme fonksiyonu

function temizle;
set_param('uclutanksistemi_duzelt_ik/sayici', 'Value','1');
set_param('uclutanksistemi_duzelt_ik/ariza_cikis1', 'Value','0');
set_param('uclutanksistemi_duzelt_ik/ariza_cikis2', 'Value','0');
set_param('uclutanksistemi_duzelt_ik/ariza_cikis3', 'Value','0');
set_param('uclutanksistemi_duzelt_ik/ariza_cikis4', 'Value','0');
set_param('uclutanksistemi_duzelt_ik/ariza_cikis5', 'Value','0');
set_param('uclutanksistemi_duzelt_ik/ariza_cikis6', 'Value','0');
set_param('uclutanksistemi_duzelt_ik/ariza_cikis7', 'Value','0');
```

ÖZGEÇMİŞ

1972 yılında Trabzon, Of'ta doğdu. İlk ve orta öğrenimini Sakarya'da tamamladı. 1990 yılında Adapazarı Ali Dilmen Lisesi'nden mezun olarak lise öğrenimini tamamladı. 1998 yılında Marmara Üniversitesi Bilgisayar Mühendisliği Bölümü'nden mezun oldu. 2001 yılında, Kocaeli Üniversitesi Enformatik Bölümü'nde, öğretim görevlisi olarak çalışmaya başladı. 2003 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü'nde başlamış olduğu yüksek lisans eğitimini 2006'da tamamladı. Aynı yıl yine Kocaeli Üniversitesi Fen Bilimleri Enstitüsü'nde doktora öğrenimine başladı. Halen Kocaeli Üniversitesi Enformatik Bölümü'nde öğretim görevlisi olarak çalışmaktadır.