

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**DESTEK VEKTÖRÜ MAKİNELERİ TABANLI  
HATA BULMA, TANIMA VE HATA TOLERANSLI KONTROL  
YÖNTEMLERİ**

**DOKTORA TEZİ  
Rana ORTAÇ KABAOĞLU**

**Anabilim Dalı : Elektrik Mühendisliği**

**Programı : Kontrol ve Otomasyon Mühendisliği**

**MAYIS 2010**

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ**

**DESTEK VEKTÖRÜ MAKİNELERİ TABANLI  
HATA BULMA, TANIMA VE HATA TOLERANSLI KONTROL  
YÖNTEMLERİ**

**DOKTORA TEZİ  
Rana ORTAÇ KABAOĞLU  
(504022050)**

**Tezin Enstitüye Verildiği Tarih : 09 Eylül 2009**

**Tezin Savunulduğu Tarih : 25 Mayıs 2010**

**Tez Danışmanı : Prof. Dr. İbrahim EKSİN (İTÜ)  
Diğer Jüri Üyeleri : Prof. Dr. Müjde GÜZELKAYA (İTÜ)  
Prof. Dr. Hakan Ali ÇIRPAN (İÜ)  
Prof. Dr. İlhan KOCAARSLAN (İÜ)  
Yrd. Doç. Dr. Gülay ÖKE (İTÜ)**

**MAYIS 2010**

*İrem Naz'ima,*



## ÖNSÖZ

Doktora öğrenimim süresince bana her türlü desteği veren, yorumları ve yönlendirmeleriyle ufkumu açan, pozitif yapısıyla motivasyonumu üst düzeyde tutan, tez danışmanım, değerli hocam Prof. Dr. İbrahim EKSİN'e, farklı bakış açıları sunarak hep daha iyiye gitmem için beni teşvik eden, bilgi birikimi ve yardımlarını esirgemeyen değerli hocam Prof. Dr. Müjde GÜZELKAYA'ya en içten teşekkürlerimi sunarım.

Tez izleme komitemde yer alan, her zaman destekleyici olan değerli hocam Prof. Dr. Hakan Ali ÇIRPAN'a çok teşekkür ederim.

Harcadıkları zaman ve emek için, her türlü yardımları için sevgili arkadaşlarım, Dr. Engin YEŞİL'e ve Y. Müh. Tufan KUMBASAR'a çok teşekkür ederim.

Varlığı ile bana güç veren biricik kızıma, manevi desteklerini esirgemeyen ve her zaman sabır gösteren sevgili anneme, babama, ablama, tüm aileme ve dostlarıma çok müteşekkirim. Son olarak, her türlü destek ve yardımları için sevgili eşim Y. Doç. Dr. Nihat KABAOĞLU'na çok teşekkür ederim.

Bu tezin, hata bulma, tanıma ve hata toleranslı kontrol konularında çalışan, destek vektörü makinelerine ilgi duyan araştırmacılara faydalı bir Türkçe kaynak olmasını gönülden dilerim.

Eylül 2009

Rana ORTAÇ KABAOĞLU  
Elektrik-Elektronik Mühendisi



## İÇİNDEKİLER

### Sayfa

ÖNSÖZ .....	v
İÇİNDEKİLER .....	vii
KISALTMALAR .....	ix
ÇİZELGE LİSTESİ .....	xi
ŞEKİL LİSTESİ .....	xiii
ÖZET .....	xv
SUMMARY .....	xvii
<b>1. GİRİŞ .....</b>	<b>1</b>
1.1 Hata Toleranslı Kontrol.....	2
1.2 Hata Bulma ve Tanıma.....	3
1.3 Tezin Amacı ve Katkısı.....	5
<b>2. DESTEK VEKTÖRÜ MAKİNELERİ.....</b>	<b>9</b>
2.1 Giriş.....	9
2.2 Destek Vektörü Makineleriyle Sınıflandırma (DVM-S).....	9
2.2.1 Doğrusal ayrılabilir veriler.....	9
2.2.2 Doğrusal ayrılamayan veriler.....	13
2.2.3 Doğrusal olmayan veriler.....	15
2.2.3.1 Kernel fonksiyonu, kernel hilesi ve sınıflandırma mekanizması	15
2.2.4 Benzetim örneği.....	18
2.3 Destek Vektörü Makineleriyle Bağlanım (DVM-B).....	20
2.3.1 Doğrusal bağlanım .....	22
2.3.2 Doğrusal olmayan bağlanım .....	25
2.3.3 Benzetim örneği .....	27
<b>3. DESTEK VEKTÖRÜ MAKİNELERİ İLE HATA BULMA VE TANIMA..</b>	<b>29</b>
3.1 Giriş.....	29
3.2 Çalışma Bölgesi Kavramı ve Destek Vektörü Makineleri ile Hata Bulma.....	32
3.3 Destek Vektörü Makineleri ile Hata Tanıma .....	36
3.4 Benzetim Örneği .....	37
3.4.1 Yöntemin uygulanışı .....	40
3.5 Sonuçların Yorumlanması.....	45
<b>4. DESTEK VEKTÖRÜ MAKİNELERİ İLE HATA TOLERANSLI KONTROL .....</b>	<b>47</b>
4.1 Giriş.....	47
4.2 Destek Vektörü Makineleri ile Hata Bulma ve Tanıma.....	49
4.3 Yeniden Yapılandırılabilir Kontrolör Tasarımı.....	49
4.4 Benzetim Örneği .....	51
4.4.1 Hata bulma ve tanıma eğitimi .....	51
4.4.2 Karar verme, uygun kontrolörün seçimi .....	53
4.5 Sonuçların Yorumlanması.....	62
<b>5. DESTEK VEKTÖRÜ MAKİNELERİ İLE DOĞRUDAN HATA TOLERANSLI KONTROL .....</b>	<b>62</b>

5.1 Giriş .....	63
5.2 DVM-B ile Doğrudan Hata Toleranslı Kontrol Yöntemi.....	63
5.2.1 Kontrolör katsayılarının belirlenmesi .....	65
5.2.2 DVM-B ile hata bulma ve tanıma .....	66
5.2.2.1 Hatanın konumunun belirlenmesi .....	66
5.3 Benzetim Örneği.....	68
5.3.1 Örnek 1 .....	68
5.3.2 Örnek 2.....	75
5.3.3 Örnek 3.....	81
5.4 Sonuçların Yorumlanması .....	85
<b>6. SONUÇLAR .....</b>	<b>87</b>
<b>KAYNAKLAR.....</b>	<b>91</b>
<b>ÖZGEÇMİŞ.....</b>	<b>101</b>

## KISALTMALAR

<b>AHTK</b>	: Aktif Hata Toleranslı Kontrol
<b>DHTK</b>	: Doğrudan Hata Toleranslı Kontrol
<b>DV</b>	: Destek Vektörleri
<b>DVM</b>	: Destek Vektörü Makineleri
<b>DVM-B</b>	: Destek Vektörü Makineleri ile Bağlanım
<b>DVM-S</b>	: Destek Vektörü Makineleri ile Sınıflandırma
<b>GA</b>	: Genetik Algoritma
<b>GBT</b>	: Hata Bulma ve Tanıma
<b>HTK</b>	: Hata Toleranslı Kontrol
<b>ITSE</b>	: <u>I</u> ntegral of the <u>T</u> ime-weighted <u>S</u> quare <u>E</u> rror
<b>PHTK</b>	: Pasif Hata Toleranslı Kontrol
<b>PID</b>	: Proportional Integral Derivative
<b>RBF</b>	: Radial Basis Function
<b>SRS</b>	: Sözde Rasgele Seri



## ÇİZELGE LİSTESİ

	<u>Sayfa</u>
<b>Çizelge 2.1</b> : Örneğin giriş çıkış değerleri.....	27
<b>Çizelge 3.1</b> : Çift tanklı sistem parametreleri.....	38
<b>Çizelge 4.1</b> : Genetik algoritmayla bulunan PID katsayıları.....	52
<b>Çizelge 5.1</b> : GA ile bulunan ve eğitimde kullanılacak PID katsayıları.....	70
<b>Çizelge 5.2</b> : Eğitim dışı farklı yükseklikteki deliklere DHTK yönteminin atadığı PID katsayıları.....	71
<b>Çizelge 5.3</b> : Eğitim dışı farklı yükseklikteki deliklere ilişkin GA ile bulunan PID katsayıları.....	71
<b>Çizelge 5.4</b> : GA ile bulunan ve eğitimde kullanılacak PID katsayıları.....	76
<b>Çizelge 5.5</b> : Eğitim dışı farklı yarıçaplardaki deliklere DHTK yönteminin atadığı PID katsayıları.....	77
<b>Çizelge 5.6</b> : Eğitim dışı farklı yarıçaplardaki deliklere ilişkin GA ile bulunan PID katsayıları.....	77
<b>Çizelge 5.7</b> : Dört farklı hata durumu için DHTK yönteminin atadığı PID katsayıları. .....	82
<b>Çizelge 5.8</b> : Dört farklı hata durumu için genetik algoritmanın atadığı PID katsayıları.....	83



## ŞEKİL LİSTESİ

### Sayfa

Şekil 2.1 : İkili sınıflandırma .....	10
Şekil 2.2 : Kısıtlamaların aktif olduğu sınırlar üzerinde uzanan destek vektörleri. ..	13
Şekil 2.3 : Doğrusal olmayan eşleme mekanizması. ....	15
Şekil 2.4 : Benzetim örneğindeki verilerin doğrusal olmayan ayırma fonksiyonu ile giriş uzayında sınıflandırılması. ....	20
Şekil 2.5 : Kayıp fonksiyonları. ....	21
Şekil 2.6 : Destek vektörleri ile doğrusal bağlanım. ....	23
Şekil 2.7 : Destek vektörleri ile doğrusal olmayan bir boyutlu bağlanım ve parametreleri. ....	26
Şekil 2.8 : Doğrusal bağlanım ( $\epsilon = 0.5$ ). ....	27
Şekil 2.9 : $\epsilon = 0.5$ için 3. dereceden polinomal kernel ile doğrusal olmayan bağlanım. Çekirdeğin derecesi arttıkça kestirimin doğruluğu artar. ....	27
Şekil 2.10 : $\epsilon = 0.5$ ve $\sigma = 1$ için gauss RBF kernel ile doğrusal olmayan bağlanım. Sigma arttıkça kestirimin doğruluğu azalır. ....	28
Şekil 2.11 : $\epsilon = 0.5$ için lineer spline kernel ile doğrusal olmayan bağlanım. ....	28
Şekil 3.1 : DVM ile hata bulma ve tanıma düzeneği. ....	33
Şekil 3.2 : Örnek sözde rasgele seri (SRS) işareti. ....	35
Şekil 3.3 : Hata bulma - DVM ile sınırların kestirimi düzeneği. ....	35
Şekil 3.4 : Bire karşı diğerleri - çoklu DVM sınıflandırma yöntemi. ....	37
Şekil 3.5 : Çift tanklı sıvı seviye sistemi. ....	38
Şekil 3.6 : Kesim frekansının seçimi .....	40
Şekil 3.7 : SRS girişler ve bunlara ilişkin çıkışla ...	41
Şekil 3.8 : Hatasız çalışma bölgesi ve sınırları. ....	42
Şekil 3.9 : Hatasız sistemin basamak cevabı ve kestirilen sınırlar. ....	43
Şekil 3.10 : Dört durumun ayırımı için çoklu DVM-S algoritması ...	43
Şekil 3.11 : Uygulanan SRS işareti. ....	44
Şekil 3.12 : Filtrelenmiş test çıkışı, kestirilen sınırlar, hata alarmı. ....	45
Şekil 4.1 : Hata toleranslı kontrol sistemlerinin sınıflandırılması ...	48
Şekil 4.2 : Önerilen HTK yapısı. ....	49
Şekil 4.3 : Dört durum için olası yollar ( $n = 4$ ). ....	51
Şekil 4.4 : Çift tank sisteminin HTK yapısı ...	52
Şekil 4.5 : PID #i ile veri toplama düzeneği. ....	53
Şekil 4.6 : Yöntemin tank1'de delik – tank'de delik hataları için izleyeceği yol. ....	54
Şekil 4.7 : Tank1'nin delinmesi durumunda PID#1 devredeyken hatalı ve hatasız durum sistem yanıtları .....	55
Şekil 4.8 : Tank1'nin delinmesi durumunda PID#1 ve PID#2 devredeyken sistem yanıtları. ....	55
Şekil 4.9 : Tank2'nin delinmesi durumunda PID#2 devredeyken sistem yanıtları. ..	56
Şekil 4.10 : Tank2'nin delinmesi durumunda PID#2 ve PID#3 devredeyken sistem yanıtları ...	56
Şekil 4.11 : Yöntemin ard arda gelen tank1'de delik - tank2'de delik hatalarında işleyişi ve geleneksel kontrolörle sistem yanıtı. ....	57

<b>Şekil 4.12</b> : Yöntemin pompada tıkanma - tank1’de delik hataları için izleyeceği yol. .....	58
<b>Şekil 4.13</b> : Pompa tıkanıklığı oluştuğunda PID#1 devredeyken hatalı ve hatasız durum sistem yanıtları .....	59
<b>Şekil 4.14</b> : Pompa tıkanması durumunda PID#1 ve PID#4 devredeyken sistem yanıtları.....	59
<b>Şekil 4.15</b> : Tank1’nin delinmesi durumunda PID#4 devredeyken sistem yanıtları.	60
<b>Şekil 4.16</b> : Tank1’nin delinmesi durumunda PID#4 ve PID#2 devredeyken sistem yanıtları .....	60
<b>Şekil 4.17</b> : Yöntemin ard arda gelen pompada tıkanıklık - tank1’de delik hatalarında işleyişi ve geleneksel kontrolörle sistem yanıtı.....	61
<b>Şekil 5.1</b> : Destek vektörü makineleri ile DHTK yapısı.....	64
<b>Şekil 5.2</b> : Karar mekanizması .....	65
<b>Şekil 5.3</b> : Önerilen DVM-B ile HBT sisteminin çalışma şeması. ....	67
<b>Şekil 5.4</b> : Önerilen DHTK ve HBT sistemi.....	68
<b>Şekil 5.5</b> : Çift tanklı sıvı seviye kontrol sistemi .....	69
<b>Şekil 5.6</b> : Delik yüksekliği 10cm. için hatasız ve hatalı sistem çıkışları.....	72
<b>Şekil 5.7</b> : $h_x=10\text{cm}$ . için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları. ....	72
<b>Şekil 5.8</b> : Delik yüksekliği 30cm. için hatasız ve hatalı sistem çıkışları.....	73
<b>Şekil 5.9</b> : $h_x=30\text{cm}$ . için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları .....	74
<b>Şekil 5.10</b> : Delik yüksekliği 50cm. için hatasız ve hatalı sistem çıkışları.....	74
<b>Şekil 5.11</b> : $h_x=50\text{cm}$ . için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları. ....	75
<b>Şekil 5.12</b> : Birinci tankında $r_x$ yarıçaplı delik olan çift tanklı sistem .....	75
<b>Şekil 5.13</b> : Delik yarıçapı 8mm. için hatasız ve hatalı sistem çıkışları. ....	78
<b>Şekil 5.14</b> : $r_x=8\text{mm}$ . için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları. ....	78
<b>Şekil 5.15</b> : Delik yarıçapı 15mm. için hatasız ve hatalı sistem çıkışları .....	79
<b>Şekil 5.16</b> : $r_x=15\text{mm}$ . için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları. ....	79
<b>Şekil 5.17</b> : Delik yarıçapı 20mm. için hatasız ve hatalı sistem çıkışları. ....	80
<b>Şekil 5.18</b> : $r_x=20\text{mm}$ . için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları. ....	80
<b>Şekil 5.19</b> : Birinci tankında $h_x$ yüksekliğinde $r_x$ yarıçaplı bir delik olan çift tanklı sistem .....	81
<b>Şekil 5.20</b> : $r_x = 8\text{mm}$ ve $h_x = 20\text{cm}$ için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.....	83
<b>Şekil 5.21</b> : $r_x = 15\text{mm}$ ve $h_x = 5\text{cm}$ için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.....	84
<b>Şekil 5.22</b> : $r_x = 7\text{mm}$ ve $h_x = 30\text{cm}$ için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.....	84
<b>Şekil 5.23</b> : $r_x = 12\text{mm}$ ve $h_x = 10\text{cm}$ için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.....	85

## DESTEK VEKTÖRÜ MAKİNELERİ TABANLI HATA BULMA, TANIMA VE HATA TOLERANSLI KONTROL YÖNTEMLERİ

### ÖZET

Bu tezin amacı, çeşitli süreçler için destek vektörü makineleri tabanlı üç ayrı akıllı hata bulma, tanıma ve hata toleranslı kontrol yaklaşımı sunmaktır. Geçmişte pek çok farklı yöntem kullanan çeşitli çalışmalar yapılmıştır. Hata toleranslı kontrol sistemleri, geleneksel geri-beslemeli kontrol sistemlerinin arıza durumlarında gösterdiği kararsızlık ve başarımların düşüklüğü gibi zayıflıkların üstesinden gelmek üzere geliştirilmiştir. Hata bulma ve tanıma sistemi, hataların varlığını tespit eden ve türü, yeri boyutu ve/veya zamanı gibi özelliklerini belirleyen birimlerdir. Hata bulma ve tanıma problemi, hata toleranslı kontrolün bir alt konusu olmasına rağmen, hakkında literatürde pek çok bağımsız çalışma bulunmaktadır. Bu çalışmada sunulan yöntemlerin hepsi akıllı bir makine öğrenimi yöntemi olan destek vektörü makinelerine dayanmaktadır. Son yıllarda destek vektörü makineleri, üstün genelleştirme kapasiteleri, sınıflandırma, bağlanım ve modellemedeki başarımları ile sinir ağlarına çok iyi bir alternatif olmuşlardır.

Bu tezdeki ilk yöntem bir hata bulma ve tanıma yaklaşımıdır. Destek vektörü bağlanımı hata bulma işleminde, destek vektörü sınıflandırması ise hata tanıma işleminde kullanılmıştır. Hata bulma işleminde güvenli bölge fikri, sistemin normal çalışma koşullarını temsil eder. Güvenli bölgenin üst ve alt sınırları iki ayrı destek vektörü bağlanım makinesi ile modellenmiştir. Bir çıkış işareti, oluşturulan güvenli bölgenin alt veya üst sınırını aşarsa bir hata tespiti yapılır. Destek vektörü ile çoklu sınıflandırma yöntemlerinden biri olan bire-karşı-diğerleri, teknik sistemde tespit edilen hatayı beklenen ve önceden belirlenen hatalardan biri olarak sınıflar.

İkinci yöntem, yeniden yapılandırma mekanizması ‘çevrim-içi kontrolör seçen’ türde olan bir aktif hata toleranslı kontrol yöntemidir. Destek vektörü ile çoklu sınıflandırma yöntemlerinden biri olan bire-karşı-diğerleri, tespit edilen hatayı beklenen ve önceden belirlenen hatalardan biri olarak sınıflar. Bir hata tespit edildiğinde sistemin kapalı çevrim başarımının devamını sağlamak için uygun olan kontrolör çevrim-içi seçilir. Bu yöntemde PID kontrolörleri kullanılmıştır ve kontrolör parametreleri genetik algoritmayla çevrim-dışı belirlenmiştir.

Üçüncü yöntem, yeniden yapılandırma mekanizması ‘çevrim-içi kontrolör hesaplayan’ türde olan bir aktif hata toleranslı kontrol yöntemidir. Sunulan bu yöntemde yeniden yapılandırma ve hata tanıma birimleri birbirlerinden bağımsız çalışırlar. Her ikisi de gerçek zamanlı sistem çıkışlarını kullanırlar. Destek vektörü bağlanım makineleri hem hata bulma ve tanıma işleminde hem de yeniden yapılandırma biriminde kullanılmıştır. Yeniden yapılandırma alt-sisteminde PID kontrolörleri kullanılmıştır. Eğitimde kullanılacak hatalı ve hatasız durumlara ilişkin PID katsayıları, genetik algoritmayla çevrim-dışı belirlenir. Üç destek vektörü bağlanım makinesi sistemden gelen veriyi eşzamanlı değerlendirerek PID kontrolörün katsayılarını çevrim-içi üretirler. Katsayıları yeniden yapılandırılan

kontrolör sistem başarımını devam ettirmek için çalışmaya başlar. Hatanın türünün tespit edilmesi için destek vektörü bağlanım makinelerini kullanan benzer bir işlem gerçekleştirilir.

Yöntemlerin başarımı, çift tanklı sıvı seviye kontrol sisteminin hatalı durumlarını içeren benzetim örnekleri ile gösterilmiştir.

# **SUPPORT VECTOR MACHINES BASED FAULT DETECTION, DIAGNOSIS AND FAULT TOLERANT CONTROL METHODS**

## **SUMMARY**

The goal of this thesis is to introduce three independent, intelligent fault detection, diagnosis and fault tolerant control approaches for various processes based on support vector machines. Various studies utilizing many different methods were presented in the past. Fault tolerant control systems have been developed to overcome some weaknesses of the conventional feedback control design, such as instability and unsatisfactory performance in the faulty cases. A fault detection and diagnosis system is a unit that obtains the occurrence of faults and determines their features in terms of type, location, size and/or time. Although the fault detection and diagnosis problem is a sub-subject of the fault tolerant control, there are many independent studies about it in literature. The methods presented in this study are all based on support vector machine that is one of the intelligent machine learning approaches. In recent years, with their superior generalization capacity, classification, regression and modeling performance, support vector machines have become a very good alternative of neural networks.

The first method in this thesis is a fault detection and diagnosis approach. Support vector regression has been used in fault detection process and support vector classification has been used in diagnosis process. In fault detection process, the confidence band idea represents the normal operating conditions of the system. The upper and the lower boundaries of the confidence band are modelled by two different support vector regression machines. A fault is detected when an output signal exceeds the upper or lower bounds of the generated confidence band. A support vector multi-classification method, one-against-all, has been used to classify the occurring fault within the group of expected and predefined faults in technical system.

The second method is an active fault tolerant control method including on-line controller selection type reconfiguration mechanism. Support vector classification has been used in fault detection and diagnosis process. A support vector multi-classification method, one-against-all, has been used to classify the occurring fault within the group of expected and predefined faults in technical system. When a fault is detected a suitable controller has been selected in an on-line manner to maintain closed-loop performance of the system. In this method, PID controllers have been used and their parameters have been obtained in an off-line manner by genetic algorithms.

The third method is an active fault tolerant control method including on-line controller calculation type reconfiguration mechanism. In the presented method, reconfiguration mechanism and diagnosis unit work independently. Both of them use only real time system outputs. Support vector regression machines have been used in fault detection and diagnosis process and also in reconfigurable controller unit. PID type controllers have been used in reconfiguration sub-system. The PID coefficients

of faulty and un-faulty cases to be used in training stage are obtained by genetic algorithm approach in an off-line manner. Three of support vector regression machines are simultaneously evaluated the data sent by the system, and produce coefficients of the PID controller. The controller, its coefficients are reconfigured, starts to work to maintain system performance in an on-line manner. In order to determine the type of fault, a similar process is exploited using one support vector regression machine.

The performance of these three methods illustrated on simulation example involving a two-tank water level control system under faulty conditions.





## 1. GİRİŞ

Modern teknolojik sistemlerde performans artırımı ve güvenilirlik sağlanması amaçlarıyla karmaşık kontrol sistemlerinden faydalanılır. Bir sistemde, sistemin eskiliğinden ve kullanılma sıklığından zamanla hatalar oluşabileceği gibi kullanıcıların sebep olacağı kurulum ve bakımdan kaynaklanan hatalar da görülebilir. Ayrıca gürültü, bozucular ve kazalar gibi dış etmenler de hata sebebi olabilir. Geleneksel bir geri-beslemeli kontrol sistemi, eyleyici, algılayıcı veya diğer sistem elemanlarındaki bazı arıza durumlarında istenilen başarıyı veya kararlılığı sağlamayabilir. Bu tür zayıflıkların üstesinden gelebilmek için, bir yandan istenen kararlılık ve performans özelliklerini sağlayacak bir yandan da elemanların kayıplarını tolere edecek yeni kontrol sistem tasarımı yöntemleri geliştirilmiştir. Bu, özellikle uçaklar, uzay araçları, nükleer güç sistemleri ve tehlikeli maddeleri işleyen kimyasal sistemler gibi güvenliğin üst düzeyde olması gereken sistemlerde önemlidir. Bu tür sistemlerde çok küçük bir hatanın doğuracağı sonuçlar çok büyük ve yıkıcı olabilir. Bu sebepten, güvenilirliğe, emniyete ve hatanın toleransına yönelik talepler genellikle yüksektir. Sistemin güvenilirliğini sağlamak için sistemdeki potansiyel hatalara tolerans sağlayacak kontrol sistemini tasarlamak gerekmektedir. Bu tür kontrol sistemlerine Hata Toleranslı Kontrol (HTK) sistemleri denir (Blanke ve diğ., 2001). Geleneksel hata toleranslı kontrol sistemlerinde hatanın varlığı tespit edilir (*hata bulma*), yeri ve zamanı belirlenir (*hata tanıma*) ve en son olarak da bu hatanın yarattığı sapmaların etkisini yok ederek sistemi nominal davranışına geri getirecek düzenleme yapılır (*hata toleransı*) (Patton, 1997). Kapalı çevrim düzeneği, gözlenen bir hatayı belirli ölçüde giderme yetisine sahiptir. Ancak kapalı çevrimin hatayı tolere edemediği durumlarda kontrolörün yeniden düzenlenmesi kaçınılmaz olabilir (Yu ve diğ., 2005). Bu sorunların giderilmesi için çeşitli HTK teknikleri geliştirilmiştir.

## 1.1 Hata Toleranslı Kontrol

Hata toleranslı kontrol (HTK), çoğu endüstriyel sistemde donanım fazlalığı ile sağlanır. Ancak donanım fazlalığının iki önemli dezavantajı vardır. Bunlar, yüksek maliyet ve yer sıkıntısıdır. Son yirmi yılda donanım fazlalığı problemine çözüm olarak analitik fazlalık veya yazılım fazlalığı olarak bilinen yöntemler geliştirilmiştir. Analitik fazlalığı kullanan iki farklı yaklaşım vardır: Aktif Hata Toleranslı Kontrol (AHTK) ve Pasif Hata Toleranslı Kontrol (PHTK). Uygulanacak yaklaşım, sistemi etkileyen hatayı belirleme yeteneğine, hatanın sebep olduğu değişikliklerin etkilerine ve sistemde kullanılan fazlalık türüne bağlıdır (Patton, 1997, Blanke ve diğ., 2003).

Pasif HTK, gürbüz kontrol yöntemlerini kullanarak kapalı çevrimli sistemi tasarlar ve belli hatalara karşı sistemin duyarsız olmasını sağlar. Bu hatalar kontrolör tasarımından önce bilinmelidir. Kontrolör bir kere tasarlandıktan sonra herhangi bir çevrim-içi hata bilgisi girişi olmaksızın beklenen hataları telafi edebilir (Eterno ve diğ., 1985, Liang ve diğ., 2000). PHTK sistemi hatalara, sanki onlar modelleme belirsizliğinin kaynaklarıymış gibi davranır. Hata tolerans becerisi oldukça sınırlıdır. Çevrim-içi çalışırken pasif bir kontrolör, önceden düşünülen hatalara karşı gürbüzdür. Bu nedenle sadece PHTK'ye güvenmek oldukça riskli olacaktır (Blanke ve diğ., 2003, Veillette ve diğ., 1992).

Çoğu geleneksel kontrol sisteminde kontrolörler, hata oluşma olasılığı gözlemlenmeden hatasız sistemler için tasarlanır. Bazen de kontrol edilecek sistemin sınırlı fiziksel fazlalık kapasitesi nedeniyle donanım yapılandırmasını fiyat veya fiziksel kısıtlamalara bağlı olarak arttırmak veya değiştirmek mümkün olmaz. Bu gibi durumlarda beklenmeyen hataları düzenlemek için mevcut kaynakları, donanım ve analitik fazlalığı kullanarak bir AHTK sistemi tasarlanabilir (Blanke ve diğ., 2003). Bir AHTK sistemi, üç işlemi gerçekleştirir; anlamlı miktarda çevrim-içi hata bulma, gerçek zamanlı karar-alma ve kontrolör yeniden yapılandırması (Zhang ve Jiang, 2006). Genel olarak AHTK sistemleri aşağıdaki özelliklere sahiptir:

- Mevcut donanım fazlalığına ek olarak analitik fazlalığı da kullanır.
- Hata bulma ve tanıma algoritması ile yeniden yapılandırılabilir kontrolör kullanır.
- Hata ile bozulmuş başarıyı alır, kullanır.

- Çok sayıda hata ile başa çıkabilir.

AHTK sistemleri, genel olarak üç ana birimden oluşur: Bunlardan birincisi, klasik kontrol döngüsüdür. İkincisi, çevrim-içi hata bulma ve tanımayı içeren katmandır. Üçüncüsü ise karar veren denetçi bir mekanizmadır (Puig ve Quevedo, 2001). Birinci birimde algılayıcı, eyleyici ve kontrolör içeren geleneksel kontrol döngüsü bulunur. İkincide, hata bulma ve teşhisi yapılır. Üçüncüde, denetçi işlevi gerçekleştirilir. Denetçi, karar mekanizmasıdır ve yeniden yapılandırma (reconfiguration) hareketinde aktiftir. Denetçinin seçeceği kontrolörün parametreleri, her bir hata için önceden ayarlanmış olabileceği gibi gerçek zamanlı olarak da belirlenebilir (Puig ve Quevedo, 2001). Bu açıdan aktif hata toleranslı kontrol sistemlerinde yeniden yapılandırma mekanizmaları, (i) çevrim-içi kontrolör seçen ve (ii) çevrim-içi kontrolör parametresi hesaplayan teknikler olarak ikiye ayrılabilir (Mahmoud ve diğ., 2003a). Çevrim-içi kontrolör seçimi yaklaşımında, önceden belirlenmiş hatalara özel olan kontrolörler, çevrim-dışı tasarlanır ve hata bulma-tanıma algoritmasından gelen bilgilere göre çevrim-içi seçilir (Ducard ve Geering, 2006, Ortaç-Kabaoğlu ve diğ., 2009b, Zhang ve Jiang, 2001). Çevrim-içi kontrolör hesaplanması yaklaşımında ise kontrolör parametreleri hatanın tespiti ile birlikte gelen bilgilere göre hesaplanır (Kale ve Chipperfield, 2005, Kim ve diğ., 2003, Napolitano ve diğ., 1995, Ortaç-Kabaoğlu ve diğ., 2009c). Böylece sistemde bir hata oluştuğunda, kapalı çevrim başarımını devam ettirecek uygun bir kontrolör seçilebilir.

Bir AHTK sistemi, hataların etkisini önceden hesaplanan bir kontrol kuralını seçerek veya gerçek zamanda yeni bir kontrol kuralını çevrim-içi sentezleyerek telafi eder. Her iki yaklaşımda hata kaynaklı değişimleri tanımlamak ve kontrol kuralını yeniden yapılandırmak için bir “Hata Bulma ve Tanıma/Ayırma” algoritmasına ihtiyaç vardır.

## **1.2 Hata Bulma ve Tanıma**

HTK sistemleri, hata bulma, ayırma, sistemin kontrolü ve akıllı bir yolla hataların üstesinden gelme mekanizmalarının bir birleşimidir. Hatanın varlığının, yerinin, türünün zamanında ve doğru olarak belirlenmesi Hata Bulma ve Teşhisi (HBT, fault detection and diagnosis) olarak tanımlanır ve HTK sistemlerinin en önemli kısmı sayılabilir. Çünkü, varlığı, yeri, türü ve zamanı doğru tespit edilemeyen hatalar sistemde daha yıkıcı sonuçlar doğurabilir. Teknik sistemlerdeki emniyet, güvenilirlik, korunabilirlik ve kurtarılabilirlik isteklerinin son otuz yıldaki artışı HBT

konusundaki ilgiyi artırmıştır. Yapılan çalışmalar, pek çok yeni HBT yönteminin geliştirilmesini sağlamıştır (Isermann, 2005, Isermann, 2006). Literatürde Hata Bulma ve Ayırma (fault detection and isolation), Hata Bulma ve Teşhisi (fault detection and diagnosis) Hata Bulma ve Tanıma (fault detection and identification) deyimleri de sıkça kullanılmaktadır. Genellikle bu üç tanımlama aynı anlamda kullanılır. Bu çalışmada hatanın türünün, yerinin ve zamanın tespiti için en genel haliyle Hata Bulma ve Tanıma (HBT) tanımı kullanılacaktır.

Bir hata bulma sistemi, beklenen sistem davranışıyla gerçek davranışı karşılaştırır. Eğer beklenen değerden sapmalar varsa bir semptom (belirti) tespit edilir ve hata bulma sistemi alarm verir. Tanıma veya teşhis etme birimi ise gözlenen analitik belirtilere, buluşsal belirtilere hatalı durum davranışına bakarak hatanın türünü, boyutunu ve yerini belirler. Buna hata izolasyonu da denir. Hata tanıma yöntemleri, geniş ölçüde istatistiksel örüntü tanıma ve karar verme mekanizmalarından (sınıflandırma ve bulanık kural tabanlı teknikler) oluşur. Hata tanıma yöntemleri, genellikle 3 gruba ayrılır (Patton ve diğ., 2000); Model tabanlı, bilgi tabanlı ve işaret tabanlı yöntemler. Model tabanlı yöntemlerde, sistemin nitel veya nicel modeli elde edilir. Rezidüer (artıklar), sistem modeliyle belirlenen beklenen davranış ile ölçülen davranış karşılaştırılarak üretilir ve karar verme aşamasıyla da rezidü değerlendirmesi yapılır (Isermann, 2005, Venkatasubramanian ve diğ., 2003).

Bilgi tabanlı hata tanıma yöntemleri, yapay zeka, sinir ağları, bulanık mantık, ve öğrenme makineleri ve bunların birleşimlerini kullanırlar (Betta ve Pietrosanto, 1998, Kovoio, 1994, Maki ve Loparo, 1997, Simani, ve diğ., 2003, El-Shal ve Morris, 2000, Füssel ve diğ., 1997, Balle ve Isermann, 1998, Filippetti ve diğ., 2000, Frank ve Koppen-Seliger, 1997a, Frank ve Koppen-Seliger, 1997b). Sistemin modelinin belirlenmesinin zor olduğu doğrusal olmayan ve belirsiz sistem durumlarında kullanılırlar. Veri tabanlı model olarak da tanımlanabilirler. Bu yöntemlerde normal olan ve olmayan tüm farklı çalışma koşulları birer örüntü olarak alınır. Sonra örneğin sinir ağları, çevrim-içi ölçüm verilerini bilinen bir örüntüyle eşler. Böylece sistemin o anki durumu tanımlanmış olur. Diğer bir HBT sınıflandırmasına göre bilgi tabanlı yöntemler, nitel model tabanlı HBT yöntemlerindedir (Dash ve Venkatasubramanian, 2000). İşaret tabanlı hata tanıma yöntemleri, spektral analiz, dalgacık (wavelet) ayrıştırması, öz eleman analizi gibi yaklaşımlardan yararlanırlar,

bazen de bilgi tabanlı yöntemlerle birleştirilerek hata bulma ve tanımda kullanılırlar (Guo ve diğ., 2000, Yoon ve diğ., 2003).

### **1.3 Tezin Amacı ve Katkısı**

Bu çalışmada Destek Vektörü Makinelerinin (DVM) modelleme, sınıflandırma ve bağlanım yetenekleri ile doğrusal olmayan sistemlerdeki hataların bulunması, tanınması ve bu hataların etkilerinin giderilmesine ilişkin yöntemler geliştirilmiştir. Bu çalışmada kullanılan destek vektörü makineleri, istatistiksel öğrenme teorisini genelleştirme kontrolü ile birleştiren bir öğrenme algoritmasıdır (Cristianini, Shawe-Taylor, 2000, Vapnik, 1998). Formülasyonu, iç nokta (interior point) yöntemleri ile kolayca çözülen bir eniyileme problemi ile oluşturulmuştur. Eğitimleri nispeten kolay olan DVM'nde sinir ağlarından farklı olarak bölgesel bir en iyi yoktur. Dizi ve dallı yapı gibi geleneksel olmayan veriler de DVM'nin girişi olarak kullanılabilir. Yüksek boyutlu verileri iyi ölçeklerler. Sonuç modelleri minimal karışık modellerdir, gürültüye duyarsızdırlar ve daha karmaşık modellerin genelleştirme becerilerini korurlar. DVM bu güçlü yanları ile sinir ağlarına iyi bir alternatif oluştururlar. Karakter tanıma, sınıflandırma, bağlanım (regresyon), fonksiyon kestirimi, en iyi kontrol başlıca uygulama alanlarıdır. Çok sınıflı sınıflandırmada da kullanılabilirler. Bunun için, çok sayıda ikili sınıflandırıcı birleştirilir veya bire-karşı-diğerleri şeklinde olan sınıflandırıcılar eğitilir ve çok sayıda çiftli sınıflandırıcı düzgün bir şekilde bir araya getirilir.

İlk olarak, DVM ile doğrusal olmayan sistemlerde nominal çalışma bölgesi sınırları modellenmiş ve bu sınırları aşan sistem çıkışının hatalı olduğunun tespiti yapılmıştır. Bulunan hatanın tanınması için çoklu DVM sınıflandırıcısı oluşturulmuştur. Bu yapı bilgi tabanlı hata tanıma yöntemlerinden sayılabilir. Bilgi tabanlı HBT yöntemlerinin diğer yöntemlere göre özellikle doğrusal olmayan sistemlerde ve belirsizlik durumunda büyük avantajları vardır. Çünkü bu tür sistemlerin modellenmesi, formülasyonu, üzerinde çalışması oldukça zordur (Füssel ve diğ., 1997, Filippetti ve diğ., 2000). İşaret tabanlı HBT yöntemlerin uygulama alanları ise kısıtlıdır ve genellikle bilgi tabanlı yöntemlerle birleştirilerek kullanılırlar (Zhao and Xu, 2004). Sinir ağları, yapay zeka, bulanık mantık araçlarını kullanan bilgi tabanlı HBT yöntemlerindedir (Frank ve Koppen-Seliger, 1997). Bu tezde geliştirilen DVM tabanlı yöntem de bilgi tabanlı HBT yöntemlerindedir ve diğer bilgi tabanlı

yöntemler gibi doğrusal olmayan sistemlerde yüksek başarımla çalışmaktadır (Ortaç-Kabaoğlu ve diğ., 2009a). Yöntemde DVM'nin hem bağlanım hem de sınıflandırma özelliklerinden yararlanılmaktadır. Sistemin hatasız/normal durumundaki giriş-çıkış bilgileriyle oluşturulan güvenli çalışma bölgesinin alt ve üst sınırları, oluşturulan iki ayrı DVM bağlanım mekanizmasıyla çevrim-dışı modellenir. Bu modelleme aracıyla, beklenen sistem çıkışlarının sınırları sadece giriş işaretine dayanarak kestirilir. Eğer gözlenen çıkış işareti, sınırlardan birini aşarsa sistem hata sinyali verir. Bunun üzerine hatanın türünü belirlemeye ilişkin hata tanıma birimi devreye girer. Hatanın, önceden belirlenmiş hatalardan hangisi oluşunun tespiti bu birimle gerçekleştirilir. Buradaki çoklu hataların ayırımı için destek vektörü makineleri sınıflandırıcıları ile oluşturulmuş bir algoritma kullanılmıştır. Kestirimde ise DVM'nin hafızasız özel bir türü kullanılmıştır. Bu seçim, hafıza, yer ve zaman avantajı sağlamaktadır.

İkinci olarak geliştirilen yöntem bir AHTK yöntemidir (Ortaç-Kabaoğlu ve diğ., 2009b). DVM ile HTK yönteminde çevrim-içi kontrolör seçen bir yeniden yapılandırma düzeni mevcuttur. Yöntemde, hata bulma ve tanıma mekanizmasının temeli DVM'dir. DVM'nin çoklu sınıflandırma becerisi, hatayı ve türünü tespit etmede kullanılmıştır. DVM hatalı ve hatasız durumların çıkış verileri ile eğitilir ve çıkışlar belli aralıklarla gözlenerek hata aranır. HBT biriminin eğitimleri farklı hataların birbirini izlemesi ihtimalleri göz önüne alınıp yapılarak yöntemin güvenilirliği artırılmıştır. HBT biriminde tespit edilen ve tanınan hata verisini karar birimine gönderir. Karar birimi, önceden tasarlanmış kontrolörlerden en uygun olanını çevrim-içi seçer. Burada kullanılan PID tipli kontrolörlerin katsayı hesapları, genetik arama algoritmasıyla çevrim-dışı olarak gerçekleştirilir.

Üçüncü olarak da yine bir AHTK sistemi geliştirilmiştir (Ortaç-Kabaoğlu ve diğ., 2009c). DVM-B ile doğrudan hata toleranslı kontrol yöntemi (DHTK), sistem çıkışına göre kontrolör katsayılarını çevrim-içi ayarlayan bir yeniden yapılandırma birimi içermektedir. Yöntem, hatanın türünün belirlenmesini öncelikle gerektirmez. Hatanın türü de istenirse kontrolör parametrelerinin kestirilmesine benzer biçimde bağımsız bir DVM mekanizmasıyla tespit edilebilir. DVM eğitimlerinde kullanılacak PID tipli kontrolör parametreleri genetik algoritmayla çevrim-dışı aranır ve DHTK yapısındaki akıllı kontrol düzeneğinde yerleştirilir. Sistem çalışırken bir hata oluştuğunda akıllı kontrol düzeneği, içerdiği üç DVM-B ile uygun PID kontrolör

katsayılarını gelen bilgiye göre kestirir. Yöntemin önceden belirlenmiş hatalarla sınırlı olmaması ve yüksek başarımı en büyük avantajıdır.

Yöntemlerin çift tanklı sıvı seviye sisteminde uygulamaları yapılmıştır. Sonuçlar oldukça tatmin edicidir. Ayrıca, DVM'nin yapısının getirdiği üstünlükleri de mevcuttur. Bir matematiksel model çıkarmaya, sinir ağlarındaki aktivasyon fonksiyonu ve katman sayısı gibi sonucu etkileyecek seçimlere, bulanık mantıktaki kural tabanı düzenlemesine ihtiyaç duymaması, sadece sistemin giriş çıkış verileriyle çalışabilmesi gibi üstünlükleri de mevcuttur. Hızlı olması, bölgesel enküçüklere düşmemesi, az bir veri kümesiyle geniş bir bölgeyi taraması ve çoklu sınıflandırmaya uygun olan yapısı da önemli özelliklerindedir.



## **2. DESTEK VEKTÖRÜ MAKİNELERİ**

### **2.1 Giriş**

Sonlu sayıda eğitim verisine sahip bir öğrenme görevi için en iyi genelleştirme başarımı, özel eğitim kümesi üzerinde varılan kesinlik ile makinenin kapasitesi denen herhangi bir eğitim kümesini hatasız öğrenme yeteneği arasında doğru denge bulunursa gerçekleşecektir. Pratik uygulamalarda geleneksel sinir ağları yaklaşımları, genelleştirmede ve verilere uyan modelleri üretmede (özellikle küçük veriler için) bazı zorluklarla karşılaşmıştır. Vapnik, 70'lerin sonlarında “Destek Vektörü Makineleri” (DVM) algoritmalarını sunmuştur. Bu algoritma, istatistik, makine öğrenimi ve sinir ağlarından pek çok yöntemi birleştirmiştir. Formülasyonu, yapısal riski en-küçükleme (Structural Risk Minimization) prensibini dahil etmektedir. Kernel fonksiyonu eşleme (mapping) yönteminin kullanımı ile DVM, küçük veriler üzerinde de iyi bir sınıflandırma genelleştirmesi sağlayabilir. Ayrıca DVM pek çok çekici özelliği ile teorik ve mühendislik uygulamalarında büyüyen bir ilgi görmektedir. Model tanıma, izole edilmiş el yazısı tanıma, nesne tanıma, ses tanıma, görüntüde yüz arama ve yazı karakterize etme gibi konularda ve ayrıca bağlantım (regresyon), fonksiyon kestirimi ve en iyi kontrol alanlarında kullanılmaktadır.

### **2.2 Destek Vektörü Makineleriyle Sınıflandırma (DVM-S)**

DVM'lerin sınıflandırma mekanizması, üç ayrı veri durumu için detaylandırılabilir;

- Doğrusal olarak ayrılabilir veriler
- Doğrusal olarak ayrılamaz veriler
- Ayırımı doğrusal olmayan veriler

#### **2.2.1 Doğrusal ayrılabilir veriler**

Eğitim verileri kümesinin iki ayrı sınıftan verildiğini düşünelim.

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), \quad x \in R^N, \quad y \in \{-1, +1\} \quad (2.1a)$$

Burada  $\{x_1, \dots, x_n\}$  kümesi bizim veri kümemiz ve  $y_i \in \{-1, +1\}$ ,  $x_i$  vektörlerinin ait oldukları sınıfların belirteçleridir. Amaç,  $(x, y)$  görünmeyen verilerin belirteçlerini kesin olarak bulacak ve sınıflandırma yanlılığını da en küçük kılacak karar fonksiyonlarını  $g(x) = \text{sign}(f(x))$  olarak bulmaktır. Eğer  $f(x)$  doğrusal bir fonksiyon ise aşağıdaki gibi yazılabilir.

$$f(x) = (w \cdot x) + b, \quad w \in R^N \text{ ve } b \in R \quad (2.1b)$$

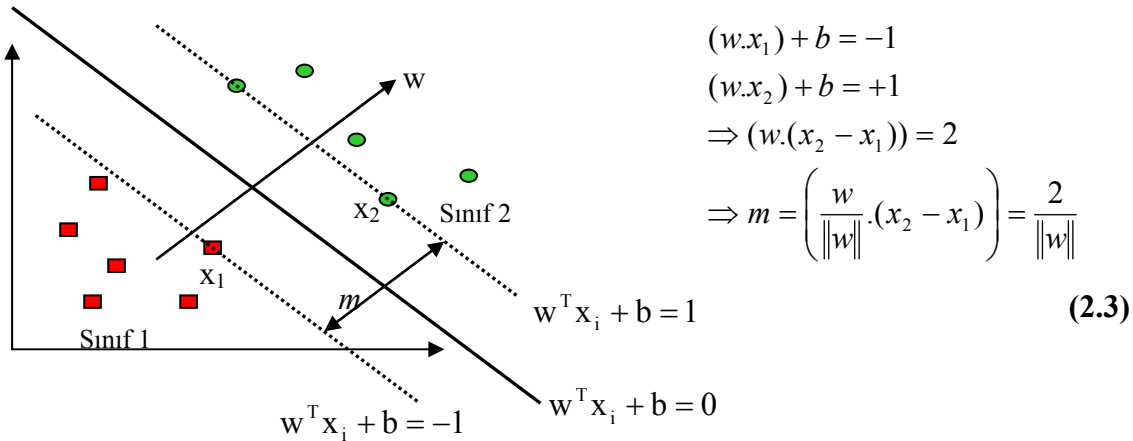
Bu,  $\{x \mid f(x) = 0\}$  karar sınırını, “+1” ve “-1” sınıflarını birbirinden ayıran N-1 boyutlu bir hiperdüzlem olan bir sınıflandırma kuralı biçiminde verir. Şekil 1 bunu göstermektedir. Veriden öğrenme problemi,  $(w, b)$  parametreleri ile tanımlanan bir ayırıcıyı (karar sınırını) bularak aşağıdaki gibi formüle edilir.

$$\begin{aligned} w^T x_i + b &\geq 1 \text{ eğer } y_i = 1 \\ w^T x_i + b &\leq -1 \text{ eğer } y_i = -1 \end{aligned} \quad (2.2a)$$

veya

$$\text{sign}((w \cdot x_i) + b) = y \quad i = 1 \dots n \quad (2.2b)$$

Şekil 1’den optimal ayırıcı hiperdüzlem payının  $m = \frac{2}{\|w\|}$  olduğu görülmektedir.



Şekil 2.1 : İkili sınıflandırma.

En iyi ayırıcı, hiperdüzlem payı enbüyüklenerek verilir. Çünkü her iki sınıf verilerinden de mümkün olduğunca uzakta olan karar sınırı, en iyi ayırıcıdır. Büyük pay, kestirimin eğitim setinde güveninilir olmasını ve görünmeyen örnekler üzerinde kestirimin başarımının iyi olmasını sağlar. Karar sınırı bulma problemi, aşağıdaki kısıtlamalı eniyileme problemi ile ifade edilir.

$$\text{Minimize } \frac{1}{2} \|w\|^2 \quad (2.4)$$

$$\text{subject to } y_i (w^T x_i + b) \geq 1 \quad \forall i \quad (2.5)$$

Bu kısıtlamalı eniyileme problemi (2.4), (2.5), Lagrange çarpanları ( $\alpha_i > 0$ ) ve bir Lagrangian ile ele alınır;

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i (y_i (w^T x_i + b) - 1) \quad \alpha_i \geq 0 \quad \forall i \quad (2.6)$$

Bu Lagrangian, w ve b değişkenlerine göre enküçüklenir,  $\alpha$  çarpanlarına göre enbüyüklenir. Problem;

$$\max_{\alpha} \min_{w, b} L(w, b, \alpha) \quad (2.7a)$$

olarak yazılır. Eniyileme problemleri ikincil biçimlerine dönüştürülebilirler. Bu, Lagrangian'ın özgün değişkenlerine göre kısmi türevleri alınıp çözülmesi ve sonuçların Lagrangian'da yerlerine konarak elenmesi şeklinde yapılır. Sonuç, sadece Lagrange çarpanlarında enbüyüklenecek bir bağıntıdır. Özgün değişkenlerdeki eşitsizlik kısıtlamaları da artık çarpanlardaki eşitlik kısıtlamalarına dönüşmüştür. İkincil eniyileme problemi,

$$\max_{\alpha} W(\alpha) , \quad \alpha \geq 0 \quad (2.7b)$$

biçiminde olacaktır. Birincil Lagrangian'ın w ve b'ye göre kısmi türevlerinden

$$\begin{aligned} \frac{\partial L}{\partial w} = 0 &\Rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i \\ \frac{\partial L}{\partial b} = 0 &\Rightarrow 0 = \sum_{i=1}^n \alpha_i y_i \end{aligned} \quad (2.8)$$

(2.6), (2.7), (2.8) den yararlanarak ikincil problem;

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.9)$$

ve kısıtlamaları

$$\begin{cases} \alpha_i \geq 0, & i = 1 \dots n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (2.10)$$

olarak yazılır. Bu bir karesel programlama (QP) problemidir ve  $\alpha_i$ 'nin bir evrensel enbüyüğü her zaman bulunabilir. Problem, gradient ve Newton gibi yöntemlerle çözülebilir. İkincil Lagrangian ile; daha basit kısıtlamalar gelir ve problem, semer noktasından basit bir enbüyüklemeye çevrilir. Fakat ikincil Lagrangian kullanmanın esas sebebi, problemi “Kernel Hilesi”nin kullanılmasına izin veren bir biçime çevirmesidir. Kernel hilesi, aşağıdaki bölümlerde açıklanmıştır.

(2.8) bağıntısının ilk terimi, çözüm vektörünün, eğitim örüntülerinin (pattern) bir alt kümesinin terimlerinde açılıma sahip olduğunu göstermektedir. Başka bir deyişle bu örüntüler, Lagrange çarpanları sıfır olmayan aktif kümelerle bulunurlar. Bu, dikkat edilmesi gereken bir noktadır.  $\alpha_i$  lerden pekçoğu sıfırdır. Optimal ayırıcı, aktif kısıtlamalar yani sıfırdan farklı  $\alpha_i$  lerle bulunur.  $w$ , az sayıdaki veri noktalarının doğrusal bir birleşimidir.  $\alpha_i$ 'nin sıfır olmadığı  $x_i$  lere “*destek vektörleri*” (DV) denir ve karar sınırı sadece destek vektörleri ile belirlenir (Şekil 2.2). Eğer veri, doğrusal olarak ayrılabilir ise tüm destek vektörleri, Karush-Kuhn-Tacker (KKT) tamamlayıcı koşulundan,

$$\alpha_i (y_i (w^T x_i + b) - 1) = 0, \quad i = 1, \dots, n \quad (2.11)$$

bağıntısını sağlayan yardımcı hiperdüzlemler üzerinde bulunurlar. Bu nedenle DV sayısı çok küçük olabilir. Sonuçta, en iyi hiperdüzlem eğitim kümesinin bir alt kümesi tarafından belirlenir, diğer noktalar eğitim kümesinden atılabilir. Bunu matematiksel olarak ifade edelim.  $t_j$  ( $j=1, \dots, s$ ) ler  $s$  tane destek vektörünün indisleri olsun. Şöyle yazabiliriz;

$$w = \sum_{j=1}^s \alpha_{t_j} y_{t_j} x_{t_j} \quad (2.12)$$

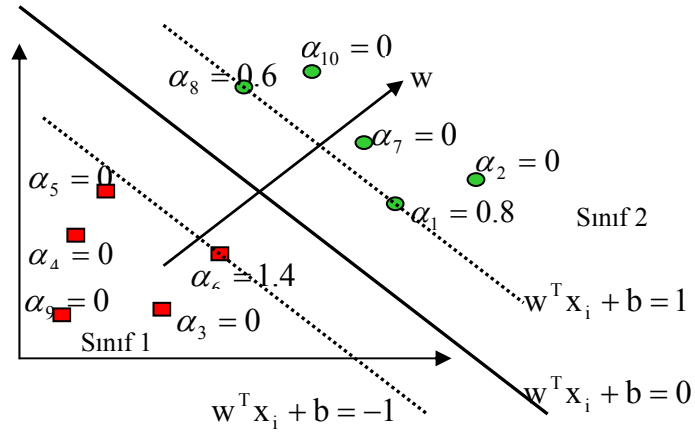
Yeni bir z datasını test etmek için

$$w^T z + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j} (x_{t_j}^T z) + b \quad (2.12a)$$

hesaplanır ve z, bu toplam pozitif ise sınıf 1'e, diğer hallerde sınıf 2'ye aittir denir, yani karar fonksiyonu aşağıdaki gibi yazılabilir;

$$f(z) = \text{sign}(w^T z + b) = \text{sign}\left(\sum_{j=1}^s \alpha_{t_j} y_{t_j} (x_{t_j}^T z) + b\right) \quad (2.12b)$$

b parametresi de, (2.11) bağıntısından çekilerek hesaplanabilir.



**Şekil 2.2 :** Kısıtlamaların aktif olduğu sınırlar üzerinde uzanan destek vektörleri.

### 2.2.2 Doğrusal ayrılmayan veriler

Doğrusal olarak ayrılmayan veriler için “yapay” değişkenler ( $\xi_i > 0$ ) tanımlanır. Böylelikle bir  $\xi$  yanılıgısına izin verilir. Bu durumda (2.5) kısıtlaması aşağıdaki gibi değiştirilir;

$$\begin{cases} w^T x_i + b \geq 1 - \xi_i & \text{eğer } y_i = 1 \\ w^T x_i + b \leq -1 + \xi_i & \text{eğer } y_i = -1 \\ \xi_i \geq 0 & \forall i \end{cases} \quad (2.13)$$

Eniyilemede  $\xi_i$  lere yapay veya gevşek değişkenler denir.  $x_i$  için bir yanılıgı yok ise  $\xi_i = 0$  olacaktır. Genelleştirilmiş en iyi ayırma hiperdüzlemi, aşağıdaki fonksiyoneli enküçükleyen  $w$  vektörü ile belirlenir;

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad (2.14)$$

Burada C, yanılı ile sınır arasındaki ödünleşim (tradeoff) parametresidir. Eniyileme problemi aşağıdaki gibidir.

$$\begin{aligned} \text{Minimize } & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{subject to } & y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned} \quad (2.15)$$

(2.15) kısıtlamalı eniyileme probleminin çözümü, Lagrangian'ın eğer noktası ile verilir;

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} \|w\|^2 + C \sum_i \xi_i - \sum_i \alpha_i (y_i (w^T x_i + b) - 1 + \xi_i) - \sum_i \xi_i \beta_i \quad (2.16)$$

Burada  $\alpha_i$  ve  $\beta_i$  pozitif değerli Lagrange çarpanlarıdır. Önceki gibi Lagrange çiftlemesi (2.16) birincil problemini ikincil probleme çevirir:

$$\max_{\alpha, \beta} W(\alpha, \beta) = \max_{\alpha, \beta} \min_{w, b, \xi} L(w, b, \xi, \alpha, \beta) \quad (2.17)$$

L Lagrangian'ının w,b,ξ'ye göre kısmi türevleri alınıp yerlerine konarak ikincil problem şöyle verilir;

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \quad (2.18)$$

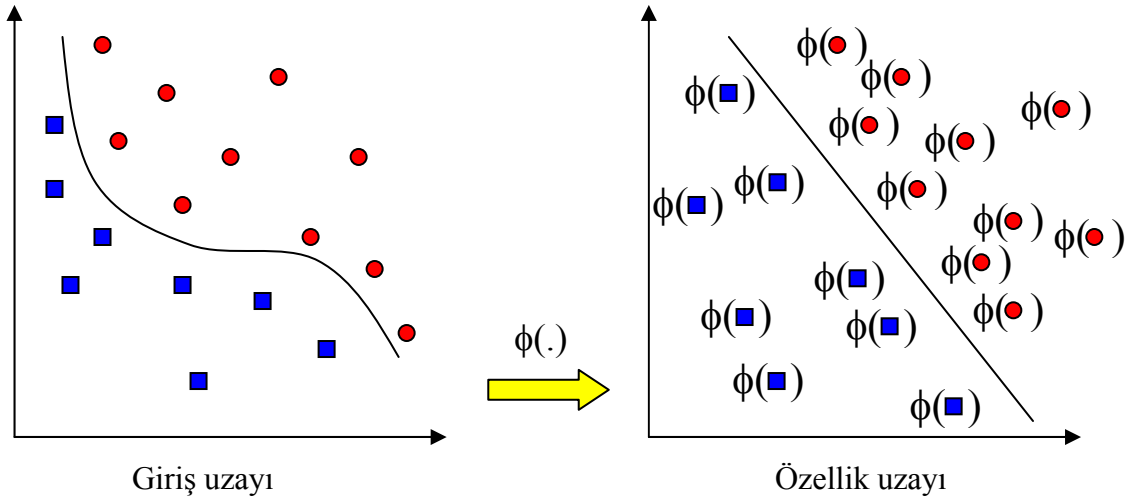
ve kısıtlamalar,

$$\begin{cases} C \geq \alpha_i \geq 0, & i = 1 \dots n \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases} \quad (2.19)$$

olur. Bu, doğrusal olarak ayrılabilen durumdaki eniyileme problemi ile çok benzerdir, problemin çözümü, ayrılabilir durumla aynıdır. Tek fark,  $\alpha_i$  üzerindeki C üst sınırıdır. C, parametresi sınıflandırıcı içinde ek bir kapasite denetimi getirir. Bir kere daha QP çözücüsü,  $\alpha_i$  leri bulmak üzere kullanılabilir.

### 2.2.3 Doğrusal olmayan veriler

Verileri ayırmak için doğrusal bir sınırın mümkün olmadığı durumlarda DVM ler giriş uzayını yüksek boyutlu bir özellik uzayına eşlemleyebilirler (mapping). Bu sayede, doğrusal olmayan bir eşleme seçerek daha yüksek boyutlu uzayda bir doğrusal en iyi hiperdüzlem inşa edilebilir.  $K(x,y)$ , özellik uzayına geçiren doğrusal olmayan eşlemeyi sağlayan *kernel* fonksiyonudur. Uygun kernel fonksiyonunun seçimi ile giriş uzayındaki doğrusal olmayan işlem, özellik uzayındaki doğrusal bir işlemle gerçekleştirilirken sınıflandırma daha basit bir hale getirilir.



Şekil 2.3 : Doğrusal olmayan eşleme mekanizması.

Özellik uzayında hesap yapma masraflı olabilir, çünkü bu uzay yüksek boyutludur, sonsuz boyutlu da olabilir. Kernel hilesi kurtarıcıdır.

#### 2.2.3.1 Kernel fonksiyonu, kernel hilesi ve sınıflandırma mekanizması

Önceki bölümde anlatılan eniyileme probleminin çözümü  $w$  ve  $b$  ile belirlenen hiperdüzlem bulunarak sonlanır. Verilerin ayrılması için doğrusal sınırların bulunamadığı durumlarda ise kernel hilesi, DVM'in doğrusal olmayan karar sınırları oluşturmasına olanak verir. (2.18) ile verilen eniyileme problemine yeniden bakalım. Dikkat edilecek olunursa, sınıflandırılacak verilerin sadece birbirleri ile iç çarpımları formülasyonda yer almaktadır. Bu noktada, belirlenecek doğru kernel fonksiyonu ile yüksek boyutlu özellik uzayına geçilerek giriş uzayında yapılacak işlemlere bu yeni uzayda olanak verilir ve iç çarpımların özellik uzayında hesaplanması gerekmez. Bu teori, "Reproducing Kernel Hilbert Spaces" (RKHS)'e dayanır. Kernel fonksiyonuna

dayanan eşlemlenmeyi daha iyi açıklamak için iki boyutlu verileri üç boyutlu uzaya taşıyan  $\Phi$  dönüşümünü aşağıdaki gibi tanımlayalım;

$$\Phi : R^2 \rightarrow R^3; \quad z = \Phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = [z_1 \quad z_2 \quad z_3]^T = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T \quad (2.19)$$

Eğer eşlemlenmiş veriyi doğrusal olarak ayırmaya çalışırsak karar sınırlarımız  $R^3$  de  $w^T z + b = 0$  biçiminde olan hiperdüzlemler olacaktır.  $x$ 'in bir fonksiyonu olarak da hiperdüzlemlerin biçimi;

$$w^T z = w_1 z_1 + w_2 z_2 + w_3 z_3 = w_1 x_1^2 + w_2 \sqrt{2} x_1 x_2 + w_3 x_2^2 = 0 \quad (2.20)$$

olur. Bu oldukça ilginçtir, çünkü bu sayede doğrusal olmayan bir algoritmayı efor harcamadan elde etmek için verilerin dönüştürülmüş hali üzerinde doğrusal algoritmamızı kullanabiliriz. Aslında algoritmada kullanılan tüm verilerin “K Gram Matrisi”dir.

$$K = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 & \cdots & x_1^T x_n \\ x_2^T x_1 & x_2^T x_2 & \cdots & x_2^T x_n \\ \vdots & \vdots & \cdots & \vdots \\ x_n^T x_1 & x_n^T x_2 & \cdots & x_n^T x_n \end{bmatrix}_{n \times n} = X X^T, \quad X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix}_{n \times d} \quad (2.21)$$

Burada  $x_i$  ler n tane dx1 boyutlu verileri temsil etmektedir. K gram matrisinin elemanları, bu vektörlerin birbirleriyle olan iç çarpımlarıdır. X matrisine “Tasarım Matrisi” denir ve tüm verileri içermektedir. Bir  $\Phi$  fonksiyonuna göre veriler eşlemlendiğinde gram matrisi;

$$K = \begin{bmatrix} \Phi(x_1)^T \Phi(x_1) & \Phi(x_1)^T \Phi(x_2) & \cdots \\ \Phi(x_2)^T \Phi(x_1) & \ddots & \\ \vdots & & \end{bmatrix} \quad (2.22)$$

olacaktır. İşleme daha yakından bakmak için  $R^2$  deki a ve b vektörleri ile bunların  $R^3$  deki dönüştürülmüş r ve s vektörlerinin iç çarpımları arasındaki ilişkiyi bulalım.

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad \xrightarrow{\Phi} \quad \mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad \mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad (2.23a)$$

$$\mathbf{a} \in \mathbb{R}^2 \xrightarrow{\Phi} \mathbf{r} \in \mathbb{R}^3 \Rightarrow (a_1, a_2) \xrightarrow{\Phi} (r_1, r_2, r_3) = (a_1^2, \sqrt{2}a_1a_2, a_2^2)$$

$$\mathbf{b} \in \mathbb{R}^2 \xrightarrow{\Phi} \mathbf{s} \in \mathbb{R}^3 \Rightarrow (b_1, b_2) \xrightarrow{\Phi} (s_1, s_2, s_3) = (b_1^2, \sqrt{2}b_1b_2, b_2^2)$$

$$\langle \mathbf{r}, \mathbf{s} \rangle = \mathbf{r}^T \cdot \mathbf{s} = r_1s_1 + r_2s_2 + r_3s_3 = a_1^2b_1^2 + 2a_1a_2b_1b_2 + a_2^2b_2^2 \quad (2.23b)$$

$$= (a_1b_1 + a_2b_2)^2 = \langle \mathbf{a}, \mathbf{b} \rangle^2$$

$$\Rightarrow K(\mathbf{a}, \mathbf{b}) = \langle \mathbf{r}, \mathbf{s} \rangle = \langle \Phi(\mathbf{a}), \Phi(\mathbf{b}) \rangle = \langle \mathbf{a}, \mathbf{b} \rangle^2$$

$\langle \mathbf{a}, \mathbf{b} \rangle$  iç çarpım ifadesi yerine onun başka boyuta taşınarak değiştirilmiş iç çarpım ifadesini koyabiliriz. Uygun bir kernel fonksiyonu ile bunu yaparak doğrusal ayrılmayan verileri ayıracak bir hiperdüzlem bulabiliriz. Böylece  $\Phi$ 'ye göre verilerimizi eşlemlenmek ve iç çarpımlarını hesaplamak yerine bunları tek bir işlemle yapabiliriz. Sonuçta  $\Phi$ 'nin de bilinmesinin gerekmediği görülür. Tüm bilmemiz gereken değiştirilmiş iç çarpımı nasıl hesaplayacağımızdır. “değiştirilmiş iç çarpım” uzun bir isim olduğu için buna kernel,  $K(x,y)$ , denir. Her fonksiyon kernel olarak kullanılamaz. Kerneli karakterize etmek için Mercer Teoreminin açıklanması gerekir.

**Teorem** (Mercer Teoremi):  $K(x,y)$  simetrik fonksiyonunun aşağıdaki gibi bir iç çarpımla ifade edilebilmesinin gerek ve yeter koşulu  $K(x,y)$  nin pozitif-yarı tanımlı, yani,

$$\int K(x,y)g(x)g(y)dxdy \geq 0, \quad \forall g \quad (2.24)$$

olması veya eşdeğer olarak;

$$\begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots \\ K(x_2, x_1) & \ddots & \\ \vdots & & \end{bmatrix} \quad (2.25)$$

matrisinin her  $\{x_1, \dots, x_n\} \subset X$  için pozitif-yarı tanımlı olmasıdır. Sonuç olarak ya verileri bir  $\Phi$  ile eşlemlenip nokta çarpımlarını alırız ya da  $\Phi$  nin neye benzediğini

dikkate almadan herhangi bir kerneli alıp doğru şekilde kullanırız. Doğru kerneli seçmek DVM problemlerinin en ustalık isteyen kısmıdır. Pratikte en çok kullanılan kernel fonksiyonlarından bazıları şunlardır:

p dereceli kernel :  $K(u, v) = (\langle u, v \rangle)^p$

p dereceli polinomal kernel :  $K(u, v) = (\langle u, v \rangle + 1)^p$

r genişlikli radyal tabanlı fonksiyon kernel :  $K(u, v) = e^{-\frac{\|u-v\|^2}{2r^2}}$

Sigmoid kernel :  $K(u, v) = \tanh(a \langle u, v \rangle - \gamma)$

Kernel fonksiyonuna göre ikincil Lagrangian (2.18) yeniden yazılırsa,

$$\begin{aligned} \max_{\alpha} W(\alpha) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to } C &\geq \alpha_i \geq 0 \quad , \quad \sum_{i=1}^n \alpha_i y_i = 0 \end{aligned} \tag{2.26}$$

ve yeni z test verisinin sınıfını belirleyecek karar fonksiyonu,

$$\begin{aligned} w &= \sum_{j=1}^s \alpha_{t_j} y_{t_j} \Phi(x_{t_j}) \\ f &= \langle w, \Phi(z) \rangle + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j} K(x_{t_j}, z) + b \end{aligned} \tag{2.27}$$

şeklinde olur. DVM'in sınıflandırma mekanizmasını daha iyi açıklayabilmek için doğrusal olmayan karar sınırlı bir benzetim örneğine bakalım.

### 2.2.4 Benzetim örneği

Beş adet 1 boyutlu veri,  $x_1 = 1, x_2 = 2, x_3 = 4, x_4 = 5, x_5 = 6$  verilmiştir. Bunlardan 1, 2 ve 6 değerli olanlar sınıf 1'e 4 ve 5 değerli olanlar sınıf 2'ye aittir yani  $y_1 = 1, y_2 = 1, y_3 = -1, y_4 = -1, y_5 = 1$  dir. Bu verilerden yararlanarak ayırıcı karar fonksiyonunu belirleyelim. Kullanacağımız kerneli, ikinci dereceden polinomal kernel  $K(u, v) = (\langle u, v \rangle + 1)^2$  ve  $C = 100$  olarak seçelim. Öncelikle  $\alpha_i (i = 1, \dots, 5)$  katsayılarını belirlemek için aşağıdaki problemi bir karesel programlama (QP) çözücüsü ile çözelim.

$$\max_{\alpha} \sum_{i=1}^5 \alpha_i - \frac{1}{2} \sum_{i=1}^5 \sum_{j=1}^5 \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2$$

$$\text{subject to } C \geq \alpha_i \geq 0, \quad \sum_{i=1}^5 \alpha_i y_i = 0$$

Problemin MATLAB yardımı ile çözülmesi ile aşağıdaki sonuçlara ulaşılır.

$$\alpha_1 = 0, \alpha_2 = 2.5, \alpha_3 = 0, \alpha_4 = 7.333, \alpha_5 = 4.833$$

Destek vektörlerinin  $\alpha$  ların sıfır olmadığı  $x$ 'ler olduğunu biliyoruz, öyleyse;

DV  $\Rightarrow \{x_2 = 2, x_4 = 5, x_5 = 6\}$  olur. Ayırıcı fonksiyon bu destek vektörleri ile aynı indisli olan  $\alpha$  lar ve  $y$  lerle elde edilir,

$$f(z) = 2.5(1)(2z+1)^2 + 7.333(-1)(5z+1)^2 + 4.833(1)(6z+1)^2 + b$$

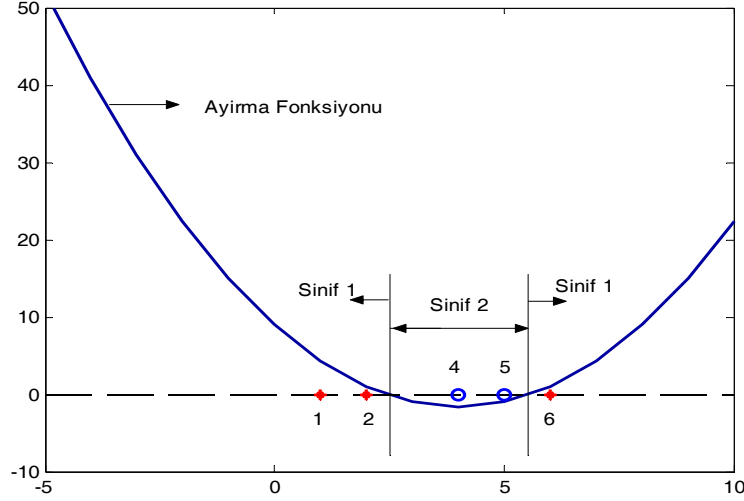
olur.  $b$  değeri  $f(2)=1$  veya  $f(5)=-1$  veya  $f(6)=1$  den çekilerek bulunur, hepsi aynı sonucu verecektir;  $b=9$ . Bu durumda ayırıcı karar fonksiyonu,

$$f(z) = 0.6667z^2 - 5.333z + 9$$

olur. Grafiksel gösterim Şekil 2.4'deki gibidir.

Destek vektörü makinelerinin sınıflandırma algoritması aşağıdaki gibi özetlenebilir,

- Tasarım (pattern) matrisini hazırla
- Kullanılacak kernel fonksiyonunu seç
- Kernel fonksiyonunun parametrelerini ve  $C$  değerini seç
- Eğitim algoritmasını uygula ve  $\alpha_i$  leri belirle
- Bilinmeyen verileri, bu  $\alpha_i$  ler ve destek vektörleri ile sınıflandır



**Şekil 2.4 :** Benzetim örneğindeki verilerin doğrusal olmayan ayırma fonksiyonu ile giriş uzayında sınıflandırılması.

### 2.3 Destek Vektörü Makineleriyle Bağlanım (DVM-B)

Destek vektörü tekniklerinin bağlanım (fonksiyon kestirimi) problemlerinin çözümünde de başarı ile uygulanabileceği yine Vapnik tarafından 1997’de keşfedilmiştir. Örüntü tanıma problemlerinin tersine burada *gerçek-değerli* fonksiyonlar söz konusudur. Genel olarak, bağlanım öğrenme problemleri aşağıdaki gibi açıklanabilir.

Giriş-çıkış ilişkisi olan  $f(x)$  fonksiyonunun öğrenilmesi için kullanılacak eğitim verileri kümesi aşağıdaki gibi verilmiş olsun.

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n), \quad x \in R^N, \quad y \in R \quad (2.28)$$

Burada, giriş verileri  $x$ ’ler,  $N$  boyutlu vektörler ve sistem cevabı olan  $y$ ’ler de skaler değerlerdir. Destek vektörü makineleri kestirilecek fonksiyonu aşağıdaki biçimde ele alır;

$$f(x, w) = \sum_i w_i \phi_i(x) \quad (2.29)$$

Burada  $\phi(x)$ , doğrusal olmayan sınıflandırmada olduğu gibi nitelikler uzayındaki değişkenlerdir. Burada yanlılık (bias) terimi olan  $b$  gösterilmemiş,  $w$  ağırlık vektörlerine dahil edilmiştir.  $f(x, w)$  fonksiyonu, öğrenmenin esas konusu olan  $w$

ağırlıklarının bir fonksiyonu olarak yazılır. Bu denklem doğrusal olmayan bir modeli ifade eder. Çünkü sonuç hiper-düzlemi, N boyutlu  $x$ -uzayında uzanan ve doğrusal olmayan bir düzlemdir. Destek Vektörü ile Bağlanım'ın tüm içeriğini ve gerekli ilişkileri incelemek için öncelikle doğrusal bağlanımı ele almakta fayda vardır. Giriş-çıkış ilişkisi nitelik uzayında,

$$f(x, w) = w^T x + b \quad (2.30)$$

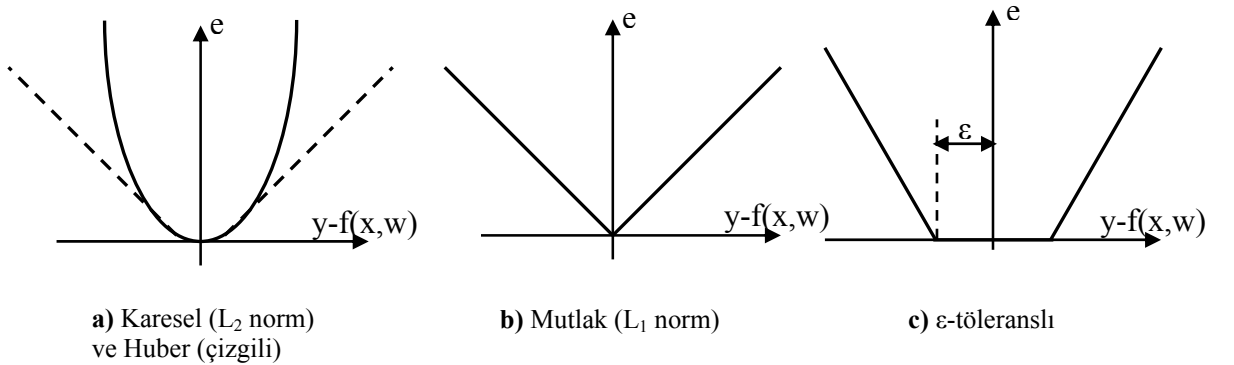
biçiminde doğrusal bir bağlanım modeliyle modellenebilir. Bağlanımda, sınıflandırmadan farklı olarak, 'optimal ayırıcı hiper-düzlem' ile destek vektörleri arasındaki pay (margin) yerine *yaklaşım yanılığı* kullanılır. Uygulamada çeşitli yanılığı (kayıp) fonksiyonu bulunmaktadır. İki klasik yanılığı fonksiyonu, karesel yanılığı ve mutlak yanılığı fonksiyonları, daha çok modeli hakkında bir şey bilinmeyen gürültü söz konusu olduğunda güvenilir olan ve dayanıklı bağlanım sağlayan Huber yanılığı fonksiyonu ve Vapnik tarafından geliştirilen  $\epsilon$ -töleranslı *kayıp fonksiyonu* bunlardandır (Şekil 2.5). Bu kayıp fonksiyonları aşağıdaki gibi verilir.

**Karesel kayıp fonksiyonu ( $L_2$  norm) :**  $Y_{karesel} = (y - f(x, w))^2$

**Mutlak kayıp fonksiyonu ( $L_1$  norm) :**  $Y_{mutlak} = |y - f(x, w)|$

**Huber kayıp fonksiyonu :**  $Y_{huber} = \begin{cases} 1/2(y - f(x, w))^2 & \text{eger } |y - f(x, w)| < \mu \\ \mu|y - f(x, w)| - \mu^2/2 & \text{diger hallerde} \end{cases}$

**$\epsilon$ -töleranslı kayıp fonksiyonu :**  $Y_{\epsilon} = \begin{cases} 0 & \text{eger } |y - f(x, w)| < \epsilon \\ |y - f(x, w)| - \epsilon & \text{diger hallerde} \end{cases}$



Şekil 2.5 : Kayıp fonksiyonları.

Fonksiyonların herbiri için farklı kısıtlamalar olduğundan farklı Lagrangian'lar oluşur. Burada bağıntılar,  $\varepsilon$ -töleranslı kayıp fonksiyonu için verilecektir.

### 2.3.1 Doğrusal bağlanım

Problemi tekrar hatırlayalım; (2.28) ile verilen veri kümesi eğitim kümesidir. Giriş-çıkış ilişkisini veren ve kestirilecek doğrusal fonksiyon (2.30) ve kullanılacak kayıp (yanılgı) fonksiyonu  $\varepsilon$ -töleranslı kayıp fonksiyonu;

$$Y_\varepsilon = \begin{cases} 0 & \text{eger } |y - f(x, w)| < \varepsilon \\ |y - f(x, w)| - \varepsilon & \text{diger hallerde} \end{cases} \quad (2.31)$$

DVM ile bağlanımın ana fikri şudur: Kestirim fonksiyonu  $f$ 'in etrafında  $\varepsilon$  yarıçaplı bir tüp veya band tanımlanır (Şekil 2.6). Eğer  $f$  değeri  $\varepsilon$  tüpünün içinde yer alırsa kayıp (yanılgı) yok demektir. Bir diğer deyişle tahmin edilen  $f$  ile ölçülen değer  $y$  arasındaki fark  $\varepsilon$ 'dan az ise kayıp sıfırdır. Tüpün dışında yer alan diğer tüm tahmin noktaları için kayıp, tahmin noktası ile  $\varepsilon$  yarıçapının farkının mutlak değerine eşittir.  $\varepsilon=0$  için Vapnik'in kayıp fonksiyonu, mutlak kayıp fonksiyonuna eşittir. Bu, Şekil 2.6'dan da gözlenebilir.

Bağlanım için DV algoritmasının formülasyonunda yanılgı toplamını ifade eden görgül (deneysel ve gözlemsel) riskin ve  $\|w\|^2$  nin eşzamanlı enküçüklenmesi amaçtır.

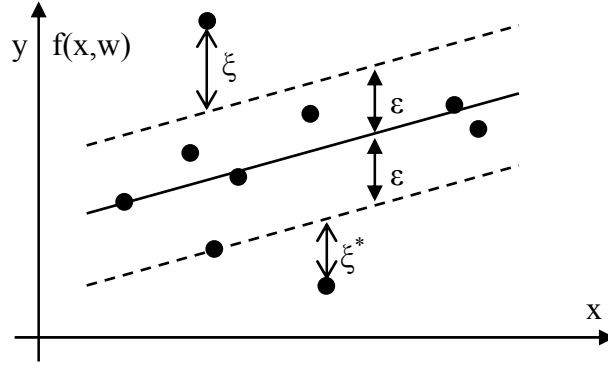
Dolayısıyla doğrusal bir  $f(x, w) = w^T x + b$  bağlanım hiperdüzlemi,

$$R = \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^n |y_i - f(x_i, w)|_\varepsilon \right) \quad (2.32)$$

ifadesinin enküçüklenmesi ile bulunur. Kayıp fonksiyonunun ifadesinden ve Şekil 2.6'dan  $\varepsilon$  tüpünün dışında yer alan tüm eğitim verileri için aşağıdaki ifade yazılabilir;

$$\varepsilon \text{ tüpünü üstünde olan veriler için } |y - f(x, w)| - \varepsilon = \xi$$

$$\varepsilon \text{ tüpünü altında olan veriler için } |y - f(x, w)| - \varepsilon = \xi^*$$



**Şekil 2.6 :** Destek vektörleri ile doğrusal bağlanım.

Buna göre  $R$ 'nin enküçüklenmesi  $\xi$  ve  $\xi^*$  gevşek (esnek) değişkenleri cinsinden yazılabilir. Dolayısıyla eniyi bağlanım fonksiyonu aşağıdaki fonksiyonelin enküçüklenmesi ile verilir.

$$R(w, \xi, \xi^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi + \xi^*) \quad (2.33)$$

ve kısıtlamalar;

$$\begin{aligned} y_i - w^T x_i - b &\leq \varepsilon + \xi \quad i = 1, \dots, n \\ w^T x_i + b - y_i &\leq \varepsilon + \xi^* \quad i = 1, \dots, n \\ \xi &\geq 0 \quad i = 1, \dots, n \\ \xi^* &\geq 0 \quad i = 1, \dots, n \end{aligned} \quad (2.34)$$

Burada  $\xi$  ve  $\xi^*$  ise çıkışın alt ve üst sınırlarını belirleyen esnek değişkenlerdir.  $\xi$  ve  $\xi^*$  ile ilişkili olan  $\alpha_i$  ve  $\alpha_i^*$  Lagrange katsayıları  $\varepsilon$  tüpünün üstünde ve altında olan eğitim noktaları için sıfırdan farklı olacaktır. Hiçbir eğitim verisi tüpün her iki tarafında da olamayacağı için  $\alpha_i$  veya  $\alpha_i^*$  sıfır olacaktır. Tüpün içindeki veriler için her iki katsayı da sıfır olacaktır.  $C$ , yanılğı ile  $\|w\|$  ağırlık vektörü normu arasındaki ödünleşim (tradeoff) parametresidir ve kullanıcı tarafından seçilir.  $C$ 'nin büyük olması yanılğının daha büyük olmasına (büyük  $\xi$  ve  $\xi^*$ ) ve bu da yaklaşım yanılğısının daha küçük olmasına neden olur. Bu,  $\|w\|$  ağırlık vektörü normundaki artışla yapılabilir. Ancak  $\|w\|$  deki artış aynı zamanda modelin iyi bir genelleştirme

başarımı göstermesini olumsuz olarak etkiler. C gibi kullanıcının seçeceği diğer bir parametre olan  $\varepsilon$  değeri de  $\varepsilon$  tüpünün boyutunu belirler.

Destek vektörü ile sınıflandırma probleminde uygulanan işlemler bağlanım için de uygulanır. Yukarıdaki kısıtlamalı eniyileme problemi birincil Lagrangian oluşturularak çözülür,

$$\begin{aligned}
L(\mathbf{w}, \mathbf{b}, \xi, \xi^*, \alpha_i, \alpha_i^*, \beta_i, \beta_i^*) = & \frac{1}{2} \|\mathbf{w}\|^2 + C \left( \sum_{i=1}^n \xi_i + \sum_{i=1}^n \xi_i^* \right) \\
& - \sum_{i=1}^n \alpha_i^* (y_i - \mathbf{w}^T \mathbf{x}_i - \mathbf{b} + \varepsilon + \xi_i^*) \\
& - \sum_{i=1}^n \alpha_i (y_i - \mathbf{w}^T \mathbf{x}_i - \mathbf{b} + \varepsilon + \xi_i) \\
& - \sum_{i=1}^n (\beta_i^* \xi_i^* + \beta_i \xi_i)
\end{aligned} \tag{2.35}$$

Birincil terimli bir Lagrangian  $L$ , birincil  $\mathbf{w}$ ,  $\mathbf{b}$ ,  $\xi$  ve  $\xi^*$  değişkenlerine göre enküçüklenmeli ve negatif olmayan Lagrange çarpanlarına  $\alpha$ ,  $\alpha^*$ ,  $\beta$  ve  $\beta^*$  göre de enbüyüklenmelidir. Problem, ikincil (dual) uzayda da çözülebilir. Burada, semer noktası problemini ortadan kaldırmak, kısıtlamaları basitleştirmek ve doğrusal olmayan bağlanımda kernel hilesini kullanabilmek için ikincil Lagrangian seçilmiştir. Karush-Kuhn-Tucker (KKT) koşulları uygulanarak ikincil Lagrangian enbüyüklenir. Bu, Lagrangian'ın özgün değişkenlerine göre kısmi türevleri alınıp çözülmesi ve sonuçların Lagrangian'da yerlerine konarak elenmesi şeklinde yapılır. Sonuç, sadece Lagrange çarpanlarında enbüyüklenecek bir bağıntıdır. İkincil eniyileme problemi,

$$\begin{aligned}
\max_{\alpha, \alpha^*} W(\alpha, \alpha^*) = & -\varepsilon \sum_{i=1}^n (\alpha_i^* + \alpha_i) + \sum_{i=1}^n (\alpha_i^* - \alpha_i) y_i \\
& - \frac{1}{2} \sum_{i=1, j=1}^n (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) x_i^T x_j
\end{aligned} \tag{2.36}$$

olur. Kısıtlamalar;

$$\begin{aligned}\sum_{i=1}^n \alpha_i^* &= \sum_{i=1}^n \alpha_i \\ 0 \leq \alpha_i^* &\leq C \quad i = 1, \dots, n \\ 0 \leq \alpha_i &\leq C \quad i = 1, \dots, n\end{aligned}\tag{2.37}$$

İkincil Lagrangian sadece  $\alpha$  ve  $\alpha^*$  lagrange çarpanları ile ifade edilmektedir. Öğrenme n adet Lagrange çarpan çiftinin  $(\alpha, \alpha^*)$  bir karesel programlama (QP) problemi çözücüsü ile bulunmasıyla sonuçlanır. Sıfırdan farklı olan (serbest)  $\alpha_i$  veya  $\alpha_i^*$  parametrelerinin sayısı destek vektörü (DV) sayısına eşittir. Bu sayı giriş uzayının boyutsallığına bağlı değildir ve yüksek boyutlu uzaylarda çalışırken özellikle önemlidir.  $(\alpha_i, \alpha_i^*)$  çiftlerinin herbirinde elemanlardan biri sıfır olduğu için çarpımları da hep sıfırdır.

$\alpha_i$  ve  $\alpha_i^*$  lagrange çarpanlarının hesaplanmasından sonra bağlanım hiper-düzleminin istenen eniyi ağırlık vektörü aşağıdaki gibi hesaplanır.

$$w_0 = \sum_{i=1}^n (\alpha_i^* - \alpha_i) x_i\tag{2.38}$$

w vektörü, sadece girilen eğitim verilerinin doğrusal bir kombinasyonu olarak ifade edilir. Eğitim verileri arasında  $(\alpha_i - \alpha_i^*)$  katsayısı sıfırdan farklı olan x vektörlerine *destek vektörü* denir. Bağlanım hiper-düzleminin eniyi yanlılık (bias) terimi,

$$b_0 = \frac{1}{n} \sum_{i=1}^n (y_i - x_i^T w_0)\tag{2.39}$$

ile hesaplanır. Bulunanlar birleştirilerek eniyi bağlanım hiper-düzlemi aşağıdaki gibi yazılır.

$$z = f(x, w) = w^T x + b = \sum_{i=1}^n (\alpha_i^* - \alpha_i) \langle x_i, x \rangle + b\tag{2.40}$$

### 2.3.2 Doğrusal olmayan bağlanım

Doğrusal olmayan bağlanım fonksiyonun DVM ile bulunmasında ise Mercer koşullarını sağlayan (simetrik, kesin pozitif) kernel fonksiyonları kullanılır. Bir  $z = \phi(x)$  dönüşümü ile (burada  $\phi, \mathbb{R}^N \Rightarrow \mathbb{R}^F$  sağlar) doğrusal olmayan uzaydan

doğrusal fakat daha yüksek boyutlu nitelik uzayına geçilir. Bu uzayda yapılacak her iç çarpıma denk gelen bir kernel fonksiyonu vardır.

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \quad (2.41)$$

Dolayısıyla kernel fonksiyonunun kullanımı ile bu çarpımların yapılmasına gerek kalmaz. Doğrusal olan F nitelik uzayında DV ile doğrusal bağlanım algoritması uygulanır. Böylece nitelik uzayında doğrusal olan  $f(x, w) = w^T x + b$  bağlanım hiper-düzlemi, özgün giriş uzayında doğrusal olmayan bağlanım hiper-düzlemi yaratacaktır. Pratikte en popüler olan kerneller, polinom kernel fonksiyonu, Gauss radyal tabanlı kernel fonksiyonu ve spline kernel fonksiyonudur.

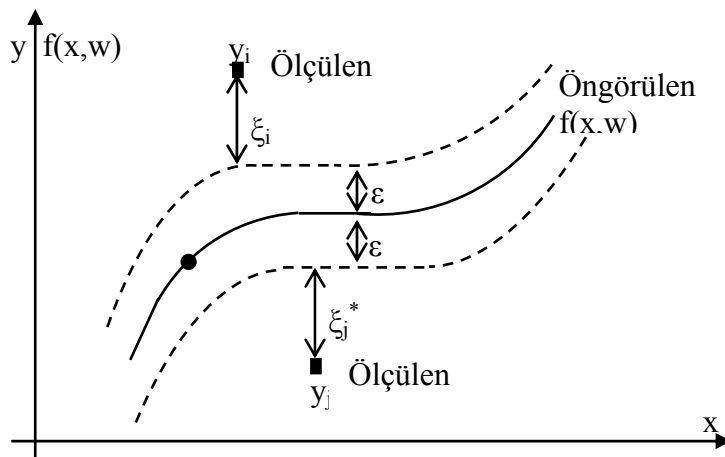
DV ile doğrusal olmayan bağlanım formülasyonu aşağıdaki şekilde yazılır.

$$f(x, w) = \sum_{i=1}^n (\alpha_i^* - \alpha_i) K(x_i, x) + b \quad (2.42)$$

$(\alpha_i - \alpha_i^*)$  katsayısı sıfırdan farklı olan x destek vektörleri ile formül tekrar yazılırsa,

$$f(x, w) = \sum_{i=1}^{S1/2} (\alpha_i^* - \alpha_i) K(x_i, x) + b \quad (2.43)$$

elde edilir.



**Şekil 2.7 :** Destek vektörleri ile doğrusal olmayan bir boyutlu bağlanım ve parametreleri.

### 2.3.3 Benzetim örneği

Çizelge 2.1’de verilen örnek bir boyutlu veri kümesini ele alalım.

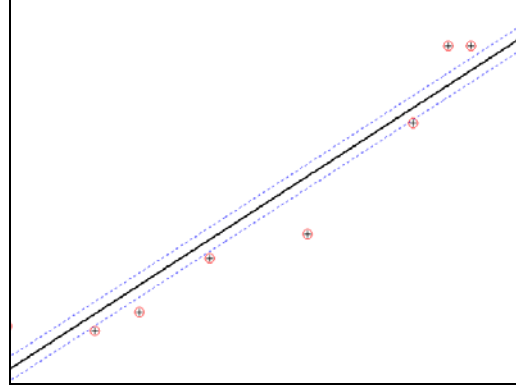
**Çizelge 2.1** : Örneğin giriş çıkış değerleri.

x	1.0	3.0	4.0	5.6	7.8	10.2	11.0	11.5	12.7
y	-1.6	-1.8	-1.0	1.2	2.2	6.8	10.0	10.0	10.0

Verileri doğrusal olan ve olmayan bağlanım fonksiyonlarını bulmada kullanacağız.

*DVM ile doğrusal bağlanım:*

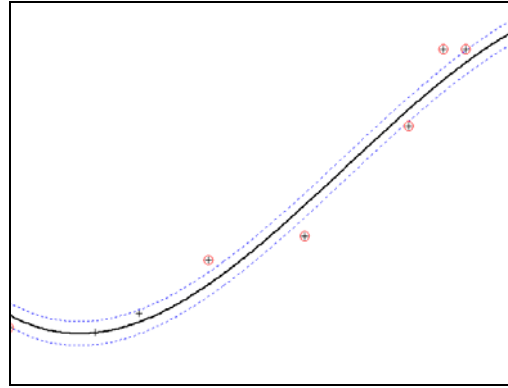
DVM ile doğrusal bağlanım fonksiyonu,  $L_1$  kayıp fonksiyonu kullanarak Şekil 2.8.deki gibi elde edilir. Burada  $\varepsilon = 0.5$  alınmıştır.



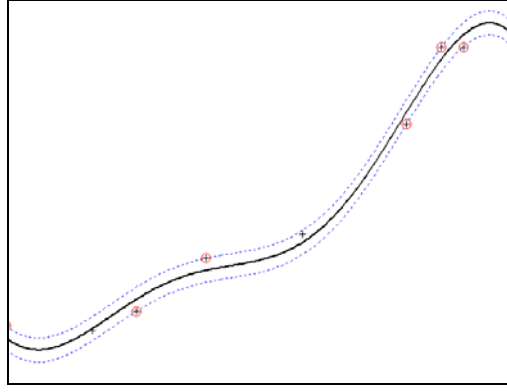
**Şekil 2.8** : Doğrusal bağlanım ( $\varepsilon = 0.5$ ).

*DVM ile doğrusal olmayan bağlanım:*

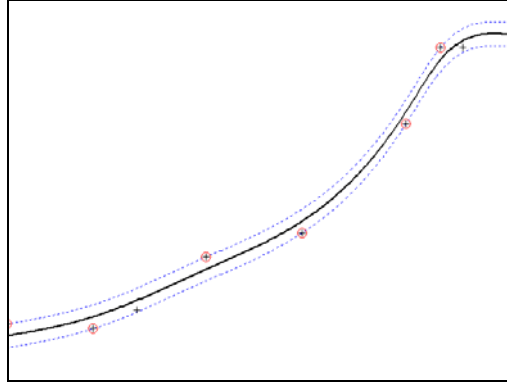
Aynı veriler ile  $\varepsilon$ -töleranslı kayıp fonksiyonu ve üç ayrı kernel fonksiyonu için bağlanım fonksiyonları Şekil 2.9, 2.10, 2.11’deki gibidir.



**Şekil 2.9** :  $\varepsilon = 0.5$  için 3. dereceden polinomal kernel ile doğrusal olmayan bağlanım. Kernelin derecesi arttıkça kestirimin doğruluğu artar.



**Şekil 2.10** :  $\epsilon = 0.5$  ve  $\sigma = 1$  için gauss RBF kernel ile doğrusal olmayan bağlanım. Sigma arttıkça kestirimin doğruluğu azalır.



**Şekil 2.11** :  $\epsilon = 0.5$  için lineer spline kernel ile doğrusal olmayan bağlanım.

### 3. DESTEK VEKTÖRÜ MAKİNELERİ İLE HATA BULMA VE TANIMA

#### 3.1 Giriş

Hata bulma, arama ve ayırma, güvenilir sistemlerin tasarımında göz önüne alınan önemli özelliklerdendir. Doğrusal sistemler için hata bulma ve tanıma problemleri, özellikle son otuz yıldır üzerinde durulan ve çözmek için pek çok önemli yöntem geliştirilen problemlerdir. Bu yöntemlerden bazıları sadece hata bulmaya odaklanırken bazıları, bulunan hatanın ayırdı üzerinde de durur (Frank, 1990, Isermann 1997, Patton ve Chen, 1994, Maybeck, 1999). Ancak çalışılan sistem doğrusal değilse bu doğrusal yöntemler yetersiz kalmaktadır. Endüstriyel sistemlerinin çoğunun doğrusal olmayan yapısı, araştırmacıları yeni hata bulma ve tanıma yöntemleri aramaya yönlendirmiştir. Geliştirilen yöntemleri farklı şekillerde sınıflandırmak mümkündür. Genellikle hata bulma ve tanıma yöntemleri, model tabanlı, bilgi tabanlı ve işaret tabanlı olarak üç ana sınıfa ayrılır.

Model tabanlı yöntemler de kendi aralarında nicel ve nitel yöntemler olarak ayrılabilir. Nicel modeller, diferansiyel denklemler, durum uzayı yöntemleri, transfer fonksiyonu gibi kontrol teorisinin konularını kullanır. Nitel modellerde, beklenen sistem davranışını belirleyecek değişkenler arasındaki ilişkiler nitel fonksiyonlarla ifade edilir. Model tabanlı hata bulma ve tanıma yöntemleri iki aşamada çalışır; rezidü üretimi ve rezidü değerlendirmesi (karar-verme). Rezidüer, sistem modeliyle belirlenen beklenen davranış ile ölçülen davranış karşılaştırılarak üretilir. Model tabanlı HBT yöntemlerinin rezidü üretimi üç ana kategoriye ayrılabilir;

- Gözleyici tabanlı yaklaşımlar: Sistem parametrelerini (durumları ve çıkışları) kestirme ilkesine göre çalışırlar. Deterministik durum için Luenberger gözleyicisi, stokastik durum için Kalman filtresi kullanılabilir. Bu yaklaşımda rezidü olarak kestirme hatası alınır. Bu yöntemler, eğer proses parametreleri biliniyorsa uygulanabilir. Hata modelleme, girişe eklenen (eyleyici ve proses) ve çıkış (algılayıcı ofset) hataları ile çalışır. Uygun gözleyicinin tasarlanması için de çeşitli yöntemler mevcuttur. Öz-yapı atama (Patton ve Chen, 1991,

Patton ve Kangethe, 1989, Shen ve diğ., 1998), bilinmeyen giriş gözleyicisi (Hou ve Muller, 1994, Chen ve diğ., 1996), hata duyarlı filtre (Douglas ve Speyer, 1996), frekans bölgesi eniyileme yaklaşımı (Frank ve Ding, 1994), bunlardan bazılarıdır. HBT yöntemlerinde geliştirilen Kalman filtresi uygulamaları (Guasch ve diğ., 2000, Keller ve Darouach, 1997, Wu ve diğ., 2000, Zhang ve Li, 1998) ve genişletilmiş Kalman filtresi çalışmaları da mevcuttur (Rupp ve diğ., 2005, Ducard ve Geering, 2006). HBT konusunda doğrusal olmayan gözleyici tasarımının kullanıldığı daha bir çok yöntem geliştirilmiştir (Garcia ve Frank, 1997, Hermans ve diğ., 1999). Bu yöntemlerde gözleyicinin sisteme uygun tasarlanması yöntemin güvenilirliği için çok önemlidir. Fakat bu tasarım, özellikle doğrusal olmayan sistemlerde oldukça karmaşıktır.

- Eşli uzay yaklaşımları: Bu yaklaşımlar sistemin giriş- çıkış modelini kullanır. Rezidüler, ölçülen ve kestirilen çıkışların ve onların türevlerinin farkları alınarak hesaplanır. Zaman ve frekans bölgesinde eşli uzay yaklaşımları geliştirilmiştir (Gertler, 1997, Chow ve Willsky, 1984). Rezidüler sadece ek giriş ve çıkış hatalarına bağlıdır. Bu yöntemlerin tasarımı ve gerçekleştirilmesi gözleyici tabanlı rezidü üretim yöntemlerine göre nispeten daha kolaydır (Isermann, 2004). Bu yöntemlerde birincil rezidü işaretleri, bir dönüşüm matrisiyle yeniden şekillendirilir. Burada amaç, bilinmeyen bozuculara karşı rezidüleri duyarsız yapmak ve hata ayırımındaki doğruluk seviyesini yükseltmektir. Bu işleme yapısal rezidü üretimi denir (Cocquempot ve diğ., 2003, Cocquempot ve diğ., 2004, Ye ve diğ., 2004, Hagenblad ve diğ., 2003, Asokan ve Sivakumar, 2007). Bu yöntemlerde de dönüşüm matrisini doğru bulmak ana unsurdur.
- Parametre kestirimi yaklaşımları: Hataların prosesin fiziksel katsayılarına olan etkisi bu yaklaşımın temelidir. Proses modelin parametrelerinin sürekli bir biçimde kestirilmesi ile rezidüler kestirim hatası olarak hesaplanır. Hataları başarıyla ayırmak için model katsayılarından proses parametrelerine geçişin/eşlemenin var olması ve bilinmesi gerekmektedir. Parametre kestirimi için, en küçük kareler kestirimi, çıkış yanılıgısı yöntemleri, kayan mod kestirimi, sinir ağları kestirimi, genişletilmiş Kalman filtresi ve bulanık kümeleme gibi yöntemler kullanılmaktadır (Isermann, 1984, Yogg, 2001,

Isermann, 1993, Isermann, 1997, Hermans ve Zarrop, 1997, Han ve Frank, 1997, Walker ve Huang, 1995).

Model tabanlı hata bulmada analitik belirtiler, rezidülerdir. Eğer rezidü yönteminin tasarımı ile rezidülerle hatalar arasındaki ilişki tamamıyla bilinebiliyorsa hata bilgisi, rezidülerden doğrudan çıkartılabilir. Belirtiler ve hatalar arasındaki ilişkiler bilinmeyebilir veya kısmen bilinebilir. Bu nedenle hata tanıma için sınıflandırma ve çıkarım yöntemleri kullanılır. Eğer rezidüler (özellikler) ve hatalar arasında hiçbir gelecek bilgisine erişilemiyorsa sınıflandırma veya örüntü tanıma yöntemleri kullanılabilir. Rezidüler, belli hatalar için deneysel olarak belirlenir. Rezidüler ve hatalar arasındaki ilişki deneysel olarak öğrenilir (veya eğitilir) ve depolanarak kesin bir bilgi tabanı oluşturulur. Hatalar da, gözlenen rezidüler ve nominal özellikler karşılaştırılarak tespit edilir. Sinir ağları ve bulanık ayırım mekanizmaları bu yöntem için elverişli yapılara sahiptir. Sinir ağları, doğrusal olmayan durumlara yatkınlığı ve esnek karar alanları ile sürekli ve ayrık yapılarda oldukça yaygın kullanılmaktadır (Betta ve Pietrosanto, 1998, De la Fuente ve Vega, 1998). Bulanık kümeleme de benzer bir uyuma sahiptir (Yogg ve diğ., 2001). Çıkarım yöntemleri, hatalar ve belirtiler arasındaki temel ilişkiler en azından kısmen bilinirse kullanılabilir. Bu ön bilgi nedensel ilişkilerle temsil edilebilir: Hata-olaylar-belirtiler. Tanımayı yapabilmek için bu nitel bilgiler, kurallar biçiminde ifade edilebilir. (Eğer-öyleyse) Koşul kısmı, belirtileri giriş olarak kullanırken sonuç kısmı da olayları ve hataları girişlerin mantıksal bir nedeni olarak kullanır. Çeşitli belirtiler bir olayı veya hatayı işaret ediyorsa girişler VE veya VEYA bağlantıları ile ilişkilendirilir. Bu durumda belirtiler ve olaylar ikili değişkenler olarak kabul edilir. Kuralların koşul kısmı, paralel seri bağlantılar için Boolean eşitlikleri ile hesaplanabilir. Hataların ve belirtilerin sürekli bir yapıya sahip olmaları nedeniyle bu işlemin başarılı olacağı kanıtlanmamıştır. Bu nedenle, yaklaşık muhakeme ve bulanık mantık, teknik süreçlerin tanınması için daha uygundur (Isermann, 2004, Rakar ve diğ., 1999, Rakar ve Juricic, 2002).

Bilgi tabanlı yöntemler, diğer bir HBT sınıflandırmasına göre nitel model tabanlı HBT yöntemlerindedir. Yapay zeka yöntemleri, sinir ağları, bulanık mantık, makine öğrenimi ve bunların birleşimi olan yöntemlerdir. Sistemin modelinin bilinmediği, belirlenmesinin zor olduğu, doğrusal olmayan ve belirsiz sistem durumlarında kullanılır. Veri tabanlı model olarak da adlandırılabilirler. Bu yöntemlerde normal

olan ve olmayan tüm farklı çalışma koşulları birer örüntü olarak alınır ve çevrim-içi ölçüm verileri bilinen bir örüntüyle eşlenir. Böylece sistemin o anki durumu tanımlanmış olur. (Simani, ve diğ., 2003, El-Shal ve Morris, 2000, Füssel ve diğ., 1997, Balle ve Isermann, 1998, Filippetti ve diğ., 2000, Chen ve diğ., 1999, Lee ve Vagners, 1997, Theilliol ve diğ., 2002, Mendonça ve diğ., 2008, Zio ve Gola, 2006, Mok ve Chan, 2008, Fagarasan ve diğ., 2004). Bigi tabanlı yöntemlerin, diğer yöntemlerden en önemli üstünlüğü, doğrusal olmayan sistemlere uygun olmalarıdır. Çoğu bilgi tabanlı yöntem, sistemin modeline ihtiyaç duymaz ve nicel model tabanlı HBT yöntemlerine göre nispeten daha kolay uygulanırlar.

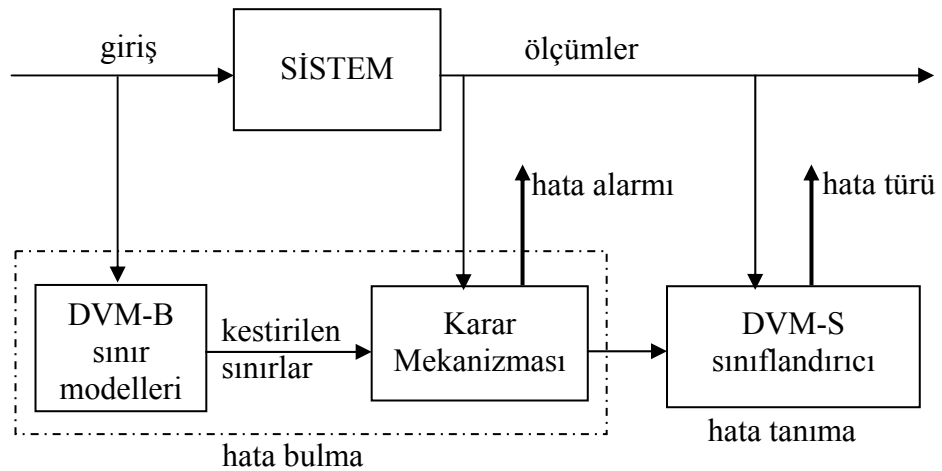
İşaret tabanlı HBT yöntemleri, işaret işleme yöntemleri olan spektral analiz, dalgacık ayrıştırması, öz eleman analizi gibi yaklaşımlar kullanırlar. Bazen HBT yöntemi olarak tek başına değil bilgi tabanlı yöntemlerle birleştirilerek kullanılırlar (Guo ve diğ., 2000, Yoon ve diğ., 2003, Postalcioglu ve diğ., 2006, Kourti, 2002, Zhao ve Xu, 2004, Oblak ve diğ., 2005)

Bu tezde geliştirilen destek vektörü makineleri tabanlı yöntem, bilgi tabanlı hata bulma ve tanıma yöntemlerinden biridir (Ortaç-Kabaoğlu ve diğ., 2009a). Doğrusal olmayan sistemlerde de yüksek bir başarıma sahip olan yöntemde DVM'nin hem bağlanım hem de sınıflandırma özelliklerinden yararlanılmaktadır. Sistemin normal çalışma durumunu temsil eden *güvenli çalışma bölgesi* fikri hata bulma aşamasının temelindedir (Oblak, ve diğ., 2007). Sistemin hatasız/normal durumundaki giriş-çıkış bilgileriyle oluşturulan güvenli çalışma bölgesinin alt ve üst sınırları, oluşturulan iki ayrı DVM-B ile çevrim-dışı modellenir. Oluşturulan güvenli bölgenin sınırlarından birini aşan bir çıkış işareti geldiğinde sistem hata sinyali verir. Bunun üzerine hatanın türünü belirlemeye ilişkin hata tanıma birimi devreye girer. Oluşan hatanın, beklenen ve önceden belirlenmiş hatalardan hangisi oluşunun tespiti bu birimle gerçekleştirilir. Buradaki çoklu hataların ayırımı için destek vektörü makineleri sınıflandırıcıları ile oluşturulmuş bir algoritma kullanılmıştır. Yöntemin uygulaması çift tanklı sıvı seviye sisteminde yapılmıştır.

### **3.2 Çalışma Bölgesi Kavramı ve Destek Vektörü Makineleri ile Hata Bulma**

Bu tez çalışmasında geliştirilen HBT yönteminde, hatanın bulunması ve tanınması işlemleri birbirini izleyen iki algoritmayla gerçekleştirilir (Şekil 3.1). Hata bulma, hatanın varlığının tespit edilmesi anlamına gelmektedir. Hatanın varlığı, sistemin

beklenen/normal çalışma koşullarından sapması ile belirlenir. Burada bu amaçla sistemin hatasız işleyişini temsil eden *güvenli çalışma bölgesi* fikri kullanılmıştır (Oblak ve diğ., 2007). Bu bölge, sistemin hatasız çalışma koşullarında toplanan çok sayıda sistem giriş-çıkış veri çiftleri ile çevrim-dışı oluşturulur. Bölgenin içi normal ve hatasız sistem koşullarını temsil eder. Bölgenin alt ve üst sınırları hatasız durumun bittiği yerlerdir. Sınırlardan en az birinin aşılması hata oluşumuna işaret eder. Yöntemde, genellikle sağlanması amacıyla değişik sıklıkta ve genlikteki çok sayıda basamak işaretinin ard arda gelmesi ile oluşan *sözde rasgele seri* (SRS) işaretleri giriş olarak kullanılır (Şekil 3.2). İlgili çıkışlar toplanır. Giriş olarak uygulanan SRS işaretlerinin sayısı, sıklık ve genlik çeşitliliği genelleştirmeyi etkileyeceği için yeterince yüksek olmalıdırlar. Toplanan hatasız durum giriş-çıkış verileri ile güvenli bölge oluşturulur. Güvenli çalışma bölgesinin alt ve üst sınırları belirlenir. Her iki sınır, birer DVM-B ile modellenir. Buraya kadar olan işlemler çevrim-dışı gerçekleştirilen eğitim aşamasını oluşturur. Eğitim aşamasında DVM-B araçlarının girişi, hatasız sistem girişleri ve çıkışı da sınırlar üzerindeki sistem çıkış verileridir. Test aşamasında (sistem çalışırken), DVM-B alt ve üst sınır kestiricileri, sistem girişlerini girişleri olarak alır. Hatasız durumdaki sistemin bu girişlere beklenen cevabının alt ve üst sınırları kestiricilerin çıkışlarıdır. DVM-B alt-üst sınır kestiricileri çevrim-içi çalışır. Gözlenen sistem çıkışının, kestiricilerin ürettiği alt ve üst sınırlar arasında kalması beklenir. Eğer gözlenen gerçek sistem çıkışı, kestirilen hatasız durum sınırlarından birini aşıyorsa hata tespit edilmiş olur ve hata alarmı verilir.



Şekil 3.1 : DVM ile hata bulma ve tanıma düzeneği.

Güvenli çalışma bölgesinin iki DVM-B ile kestirilen alt ve üst sınırları  $\underline{y}_f(u_f, t)$  ve  $\overline{y}_f(u_f, t)$  ile gösterilmiştir (Şekil 3.3). Sınırların modellenmesi için NARMAX modelleme kullanılarak DVM-B'lar tasarlanmıştır. Matematiksel modelinin bilinmediği durumlarda sistemin  $k+1$  anındaki çıkışı,  $n$  tane önceki çıkış örneğinin ve  $m$  tane önceki giriş örneğinin bir işlevi olarak ifade edilebilir (NARMAX model).

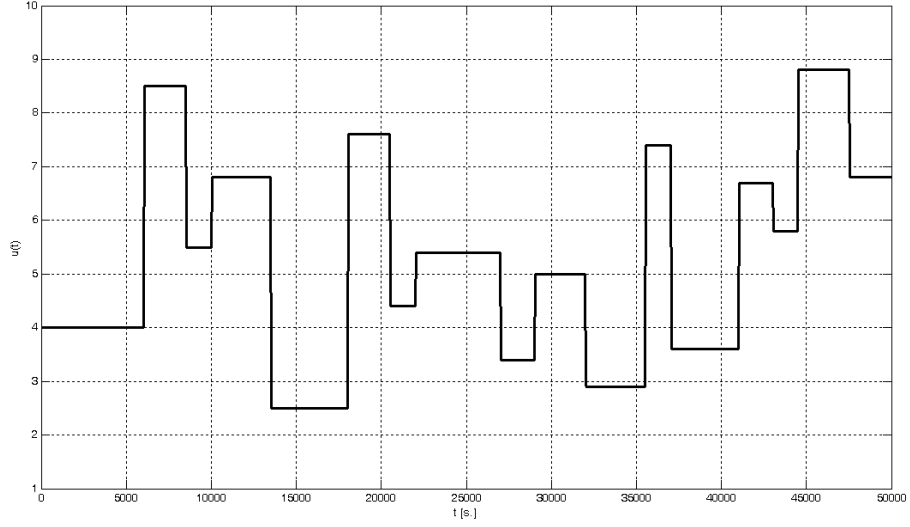
$$\begin{aligned} y(k+1) &= f(y(k), \dots, y(k-n+1), u(k-d), \dots, \\ &u(k-d-m+1), \eta(k), \dots, \eta(k-l)) = f(x(k)) \end{aligned} \quad (3.1)$$

Burada  $y$ ; sistem çıkışını,  $u$ ; kontrol girişini,  $\eta(k)$ ; belirsizlikleri (gürültü, modellenemeyen dinamikler, v.b.),  $n$ ,  $m$ ,  $l$ ; çıkış, giriş ve gürültü için en büyük gecikmeleri,  $d$ ; örnek aralarında ölçülen gecikme adımı (eğer örnekleme periyodu  $T_s$  ise sistemin ölü zamanı  $d.T_s$  olur) gösterir.  $f(\cdot)$ , doğrusal olmayan bir işlevdir. DVM ye giriş vektörü;

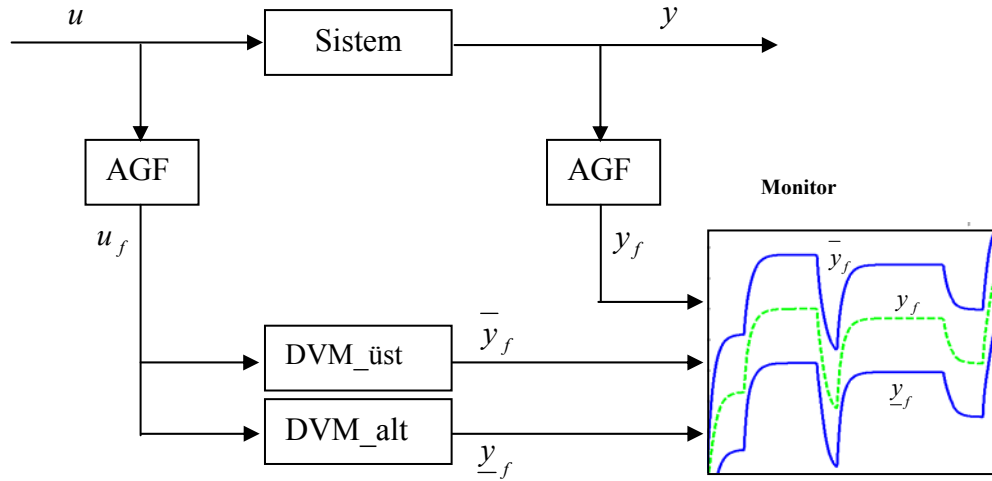
$$\begin{aligned} x(k) &= [y(k), \dots, y(k-n+1), u(k-d), \dots, u(k-d-m+1), \\ &\eta(k), \dots, \eta(k-l)]^T \end{aligned} \quad (3.2)$$

$x \in R^{n+m+l}$ ,  $k=d+1, \dots, d+N$ . Çıkışı ise  $k+1$  anındaki çıkış  $y(k+1)$ 'in kestirilmiş değeri olan  $\hat{y}(k+1)$  dir. Dolayısıyla doğrusal olmayan sistemin modeli (2.36) dan aşağıdaki gibi yazılır;

$$\hat{y}(k+1) = \sum_{i=1}^{SVs} (\alpha_i^* - \alpha_i) K(x_i, x(k)) + b \quad (3.3)$$



Şekil 3.2 : Örnek sözde rasgele seri (SRS) işareti.



Şekil 3.3 : Hata bulma - DVM ile sınırların kestirimi düzeneği.

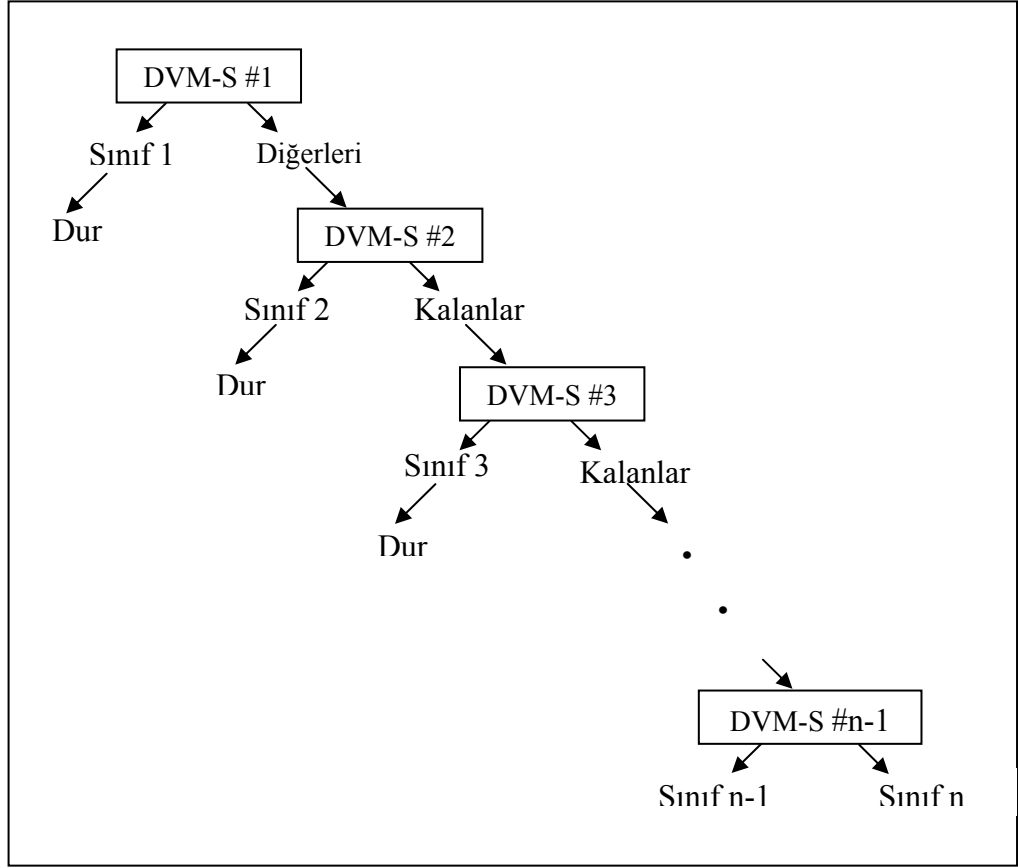
Burada (3.1) ve (3.2) ile verilen DVM-B'nin hafızasız özel bir türü kullanılmıştır. Bu seçim, hafıza, yer ve zaman avantajı sağlamaktadır. Az sayıda veri ile geniş bölgeleri tarayan DVM'nin yapısı buna uygundur. DVM'ne giriş ve çıkış vektörleri sırasıyla,  $x(k) = [u_f(k)]$  ve  $y(k) = [y_f(k+1)]$  dir. Hata bulmada ana fikir şudur; eğer filtrelenmiş bir çıkış işareti, sınırları kestirilen  $[y_f(u_f, t), \bar{y}_f(u_f, t)]$  aralığında kalıyorsa hata yoktur, sınırlardan birini geçiyorsa hata vardır ve hata işareti verilir. Giriş ve çıkış işaretlerinin modellemeye önce filtrelenmesi, hata alarmına olan duyarlılığı azaltır. Burada  $Q(s) = 1/(T_f s + 1)$  yapısındaki alçak geçiren filtre (AGF)

kullanılmıştır.  $T_f$  filtrenin zaman sabitidir. Kesim frekansı  $w_f$ , çıkış işaretinin fourier dönüşümünün mutlak değerine göre seçilir ve filtrenin zaman sabiti  $T_f = 1/w_f$  ile belirlenir.

### 3.3 Destek Vektörü Makineleri ile Hata Tanıma

Hatanın varlığının tespiti, HBT yöntemlerinin ilk aşamasını oluşturur. Hata tanıma veya ayırma, varlığı anlaşılan hatanın türünü, yerini, zamanını belirleme anlamına gelmektedir. Özetle hatanın özelliklerini belirlemeye ilişkin bir aşama olarak tanımlanabilecek olan *hata tanıma*, karmaşık sistemlerde hataların giderilmesi veya telafisi için oldukça önemlidir. Çünkü yanlış teşhis ve dolayısıyla gelen yanlış tedavi, durumu daha kötü koşullara sürükleyebilir. Destek vektörü makineleri, sınıflandırmadaki yüksek başarımları ve genelleştirme kapasiteleri ile hata tanıma yöntemleri arasında önemli bir yere sahiptir. DVM ikili sınıflandırma yapısındadır. Sistemde meydana gelebilecek olası hataların sayısı birden fazla olduğunda, durumu hatalı veya hatasız olarak ikiye ayırmak yeterli olmayacaktır. Tek bir ikili DVM sınıflandırıcısı (DVM-S) tüm hataların ayırımını yapamaz. Bu tür çoklu sınıflandırmalar için bazı DVM-S algoritmaları geliştirilmiştir. Çoklu sınıflandırma için kullanılan DVM tabanlı yöntemlerin en yaygınlarından biri *bire karşı diğerleri* yöntemidir (Hsu ve Lin, 2002).

Bu algoritma, ayrılacak sınıf sayısından bir eksik sayıda DVM sınıflandırıcısının bir araya getirilmesi ile oluşturulur. Ayrılacak sınıf sayısı  $n$  ise  $n-1$  tane DVM-S kurulur (Şekil 3.4). Öncelikle bu  $n$  tane sınıfın her birine ait veriler toplanır. İlk DVM-S ile bu  $n$  tane sınıftan biri diğer  $n-1$  adet sınıfa karşı eğitilir; bir sınıf (+1) ile işaretlenip ilk DVM-S'nin birinci sınıfına, diğer tüm sınıflar (-1) ile işaretlenip DVM-S'nin ikinci sınıfına atanır. İkinci DVM-S ile ilk DVM-S'nde ikinci sınıfı oluşturan gruptan biri alınır ve kalanlara karşı eğitilir. Algoritma bu şekilde çalışan " $n-1$ " tane DVM-S'nden oluşur. Eğitim tamamlandıktan sonra test verisi uygulanabilir. Test verisinin sınıfı tespit edildiğinde tüm DVM sınıflandırıcıları çalışmamış olsa bile algoritma durur. Örneğin henüz 2. DVM-S ile test verisinin sınıfı belirlenmişse 3. DVM-S çalışmaz. Sınıflandırma birimi yeni bir hata alarmına kadar çalışmaz.



Şekil 3.4 : Bire karşı diğerleri - çoklu DVM sınıflandırma yöntemi.

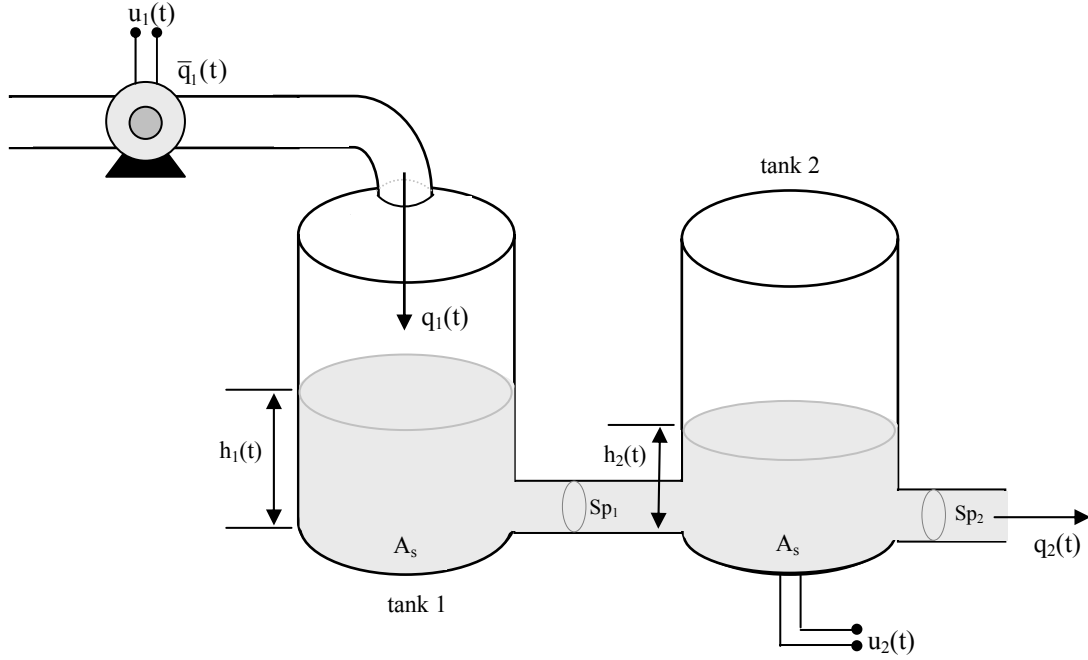
### 3.4 Benzetim Örneği

Sunulan hata bulma ve tanıma yönteminin benzetimi, bu tür problemlerde iyi bilinen ve sıkça kullanılan *çift tanklı sıvı seviye sisteminde* yapılmıştır (Zhang ve diğ., 2004). Sistem, aynı boyutlu iki silindirik sıvı tankının bir bağlantı burusuyla birleştirilmesiyle oluşturulur (Şekil 3.5).  $h_1(t)$  ve  $h_2(t)$ , tanklardaki sıvı seviyelerini,  $q_1(t)$ , birinci tanka elektrikli bir pompadan gelen kaynak akışını,  $q_2(t)$  de ikinci tanktan çıkış akışını göstermektedir.

Kütle-denge ve enerji denklemlerine göre aşağıdaki sistem denklemleri elde edilir. Sistemin parametreleri Çizelge 3.1’de verildiği gibidir.

$$\begin{aligned} \dot{h}_1(t) &= \frac{1}{A_s} \left( -K_{p1} \text{sign}(h_1(t)-h_2(t)) \sqrt{2g|h_1(t)-h_2(t)|} + q_1(t) \right) \\ \dot{h}_2(t) &= \frac{1}{A_s} \left( K_{p1} \text{sign}(h_1(t)-h_2(t)) \sqrt{2g|h_1(t)-h_2(t)|} - K_{p2} \sqrt{2gh_2(t)} \right) \end{aligned} \quad (3.4)$$

Burada,  $K_{p1}=a_1S_{p1}$  ve  $K_{p2}=a_2S_{p2}$  çıkış sabitleri,  $g$  yerçekimi kuvvetidir. Basitlik için  $a_1= a_2=1$  alınmıştır.



Şekil 3.5 : Çift tanklı sıvı seviye sistemi

Çizelge 3.1 : Çift tanklı sistem parametreleri.

Tankların taban alanları [m <sup>2</sup> ]	$A_s = 0.0154$
Bağlantı borularının alanları [m <sup>2</sup> ]	$S_{p1}, S_{p2} = 3.6 \times 10^{-5}$
Yerçekimi ivmesi [m/s <sup>2</sup> ]	$g = 9.81$
Su seviyeleri [m]	$h_1, h_2$
Pompadan tank1'e akış oranı [m <sup>3</sup> /s]	$q_1(t)$
İkinci tanktan sıvı çıkış oranı [m <sup>3</sup> /s]	$q_2(t)$
Çıkış katsayıları	$K_{p1}, K_{p2}$

Sistem girişi, elektrikli pompa gerilimi olan  $u_1(t)$  dir.  $u_1(t)$ , aşağıdaki giriş akışını üretir.

$$\bar{q}_1(t) = K_u (1 + v_1(t)) u_1(t) \quad (3.4)$$

$K_u=8.8 \times 10^{-6} \text{ m}^3/\text{Vs}$  , gerilimden akışa çevirme katsayısıdır, burada  $v_1(t)$  çevirme hatasını göstermektedir. Gerçek giriş akışı oranı ise aşağıdaki gibi tanımlanır:

$$q_1(t) = \bar{q}_1(t) + (1 - K_f)\bar{q}_1(t) \quad (3.5)$$

$\bar{q}_1(t)$ , hata olmadığı durumdaki akıştır.  $K_f \in [0,1]$  hata katsayısıdır. Modelleme aşamasında bu katsayının sıfır olduğu hatasız durum kullanılacaktır. Ölçülebilen tek çıkış işareti, ikinci tanktaki sıvı seviyesi olan  $h_2(t)$  yi çıkış gerilimi  $u_2(t)$  ye aşağıdaki denkleme göre çeviren basınç sensörünün gerilimidir.

$$u_2(t) = K_h(1 + v_2(t))h_2(t) \quad (3.6)$$

Burada  $K_h=16.667 \text{ V/m.}$ , yükseklikten gerilime çevirme katsayısıdır ve  $v_2(t)$  çevirme hatasını ifade etmektedir. Hataların üst sınırları,  $\bar{v}_1 = \bar{v}_2 = 0.03$  dür. Yukarıdaki tüm denklemler ve yapılan tüm tanımlamalar kullanıldığında aşağıdaki matematiksel sistem modelleri elde edilir.

$$\begin{aligned} \dot{h}_1(t) &= -1.035454 \times 10^{-2} \text{sign}(h_1(t) - h_2(t))\sqrt{|h_1(t) - h_2(t)|} + 1.143 \times 10^{-3} u_1(t) \\ \dot{h}_2(t) &= 1.035454 \times 10^{-2} \text{sign}(h_1(t) - h_2(t))\sqrt{|h_1(t) - h_2(t)|} - 1.035454 \times 10^{-2} \sqrt{h_2(t)} \end{aligned} \quad (3.7)$$

(3.7) ile verilen çift tank sistem modeli, sistemde bir arıza olmadığı durumu ifade etmektedir.

Aşağıda olası sistem arıza veya hatalarından bazıları verilmiştir:

- *Pompada eyleyici hatası* : (3.5) ile verilen gerçek giriş akışı formülünde  $K_f \in [0,1]$  nin sıfırdan farklı olma durumunda oluşur. Örneğin  $K_f = 0.2$  ise %20 eyleyici hatası var demektir:

$$q_1(t) = \bar{q}_1(t) + (1 - K_f)\bar{q}_1(t) \quad (3.8)$$

- *Tank 1'de delik* : Tank 1'in tabanında  $r_1$  yarıçaplı daire biçiminde bir delik oluşması halinde sistem denkleminde bir sıvı çıkışı ifadesi daha yer alacaktır. Bu sızıntının akışı aşağıdaki gibi ifade edilir:

$$q_{f1}(t) = a_1 \pi(r_1)^2 \sqrt{2gh_1(t)} \quad (3.9)$$

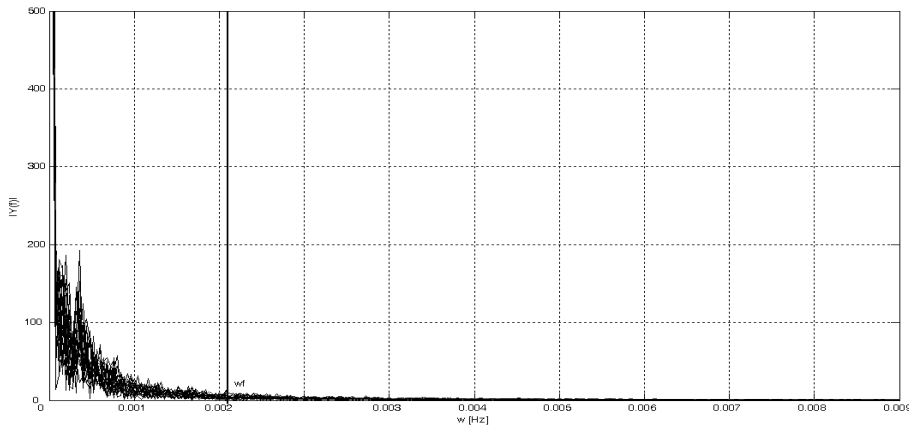
- *Tank 2'de delik* : Tank2'nin tabanında  $r_2$  yarıçaplı daire biçiminde bir delik oluşması halinde sızıntının akışı:

$$q_{f2}(t) = a_2 \pi(r_2)^2 \sqrt{2gh_2(t)} \quad (3.10)$$

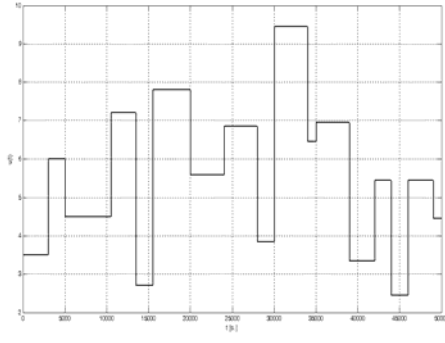
### 3.4.1 Yöntemin uygulanışı

Herşeyden önce sistemin güvenli çalışma bölgesinin belirlenmesi gerekmektedir. Bunun için Matlab-Simulinkte oluşturulan 20 adet SRS giriş işareti hatasız sisteme uygulanır ve normal koşullardaki filtreli sistem giriş-çıkış verileri toplanır, filtre parametresi belirlenir (Şekil 3.6). Şekil 3.7'da bu verilerden birkaç tanesinin zamanla değişimi görülmektedir. Toplanan (giriş,çıkış) verileriyle güvenli çalışma bölgesi oluşturulur (Şekil 3.8). Çalışma bölgesinin üst sınırının modellenmesi için ilk DVM-B olan DVM\_üst, üst sınırdaki  $(u_f, y_f)$  veri çiftleri ile eğitilir. Benzer şekilde DVM\_alt da eğitilir. Burada DVM parametreleri,  $C=200$ ,  $\varepsilon=0.0001$  ve kernel fonksiyonu olarak polinom kernel ( $p=1$ ) seçilmiştir.

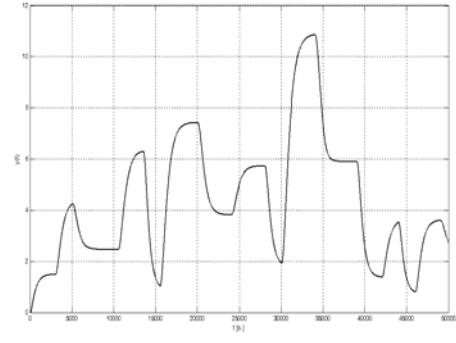
Burada, alçak geçiren filtre  $Q(s)=1/(T_f s+1)$  yapısındadır.  $T_f$  filtrenin zaman sabitidir. Kesim frekansı  $w_f$ , çıkış işaretinin fourier dönüşümünün mutlak değerine göre seçilir. Filtrenin zaman sabiti  $T_f = 1/w_f$  ile belirlenir (Oblak ve diğ., 2007). Bu örnekte  $T_f = 476s$ . seçilmiştir (Şekil 3.6).



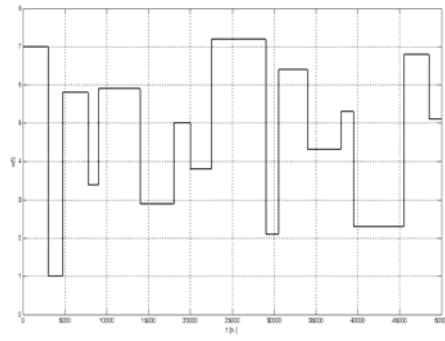
Şekil 3.6 : Kesim frekansının seçimi.



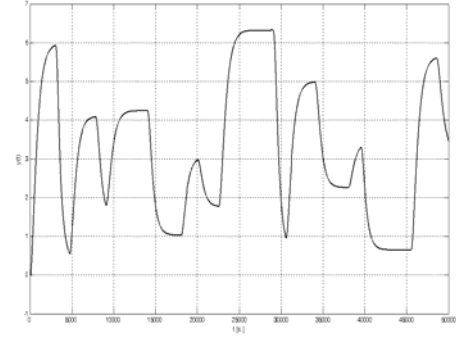
**a- u1 girişi**



**b- y1 çıkışı**



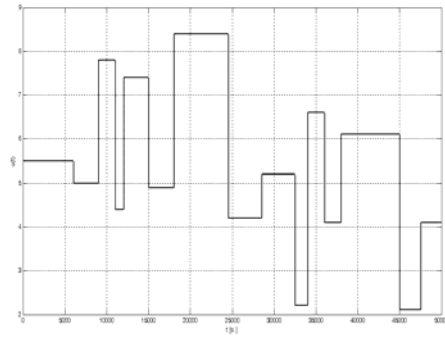
**c- u2 girişi**



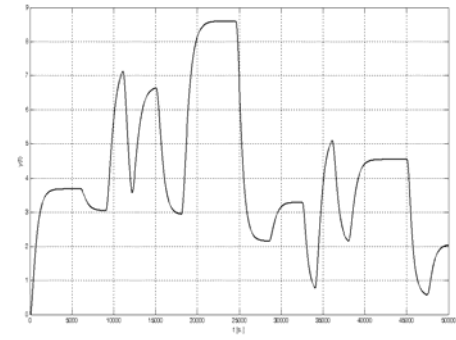
**d- y2 çıkışı**

•  
•  
•

•  
•  
•

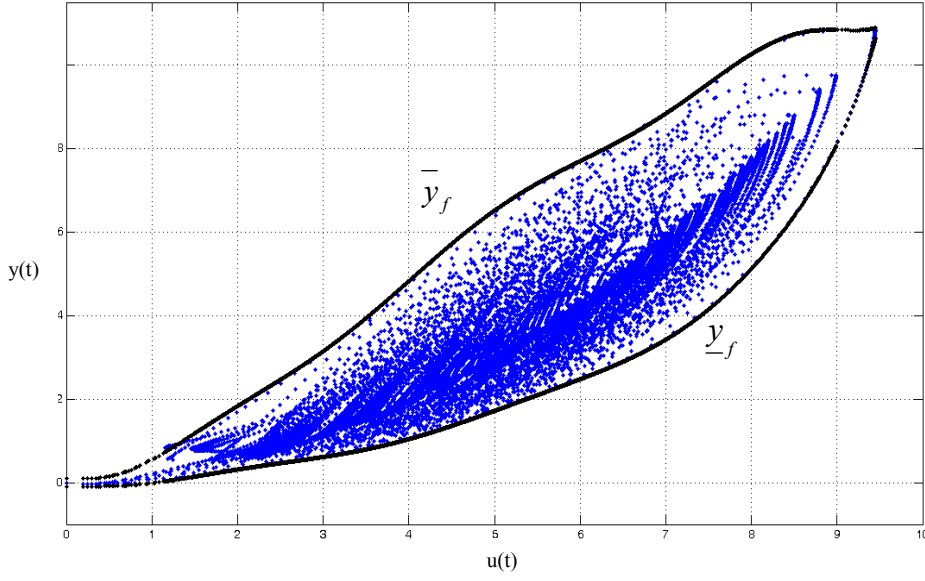


**e- u20 girişi**



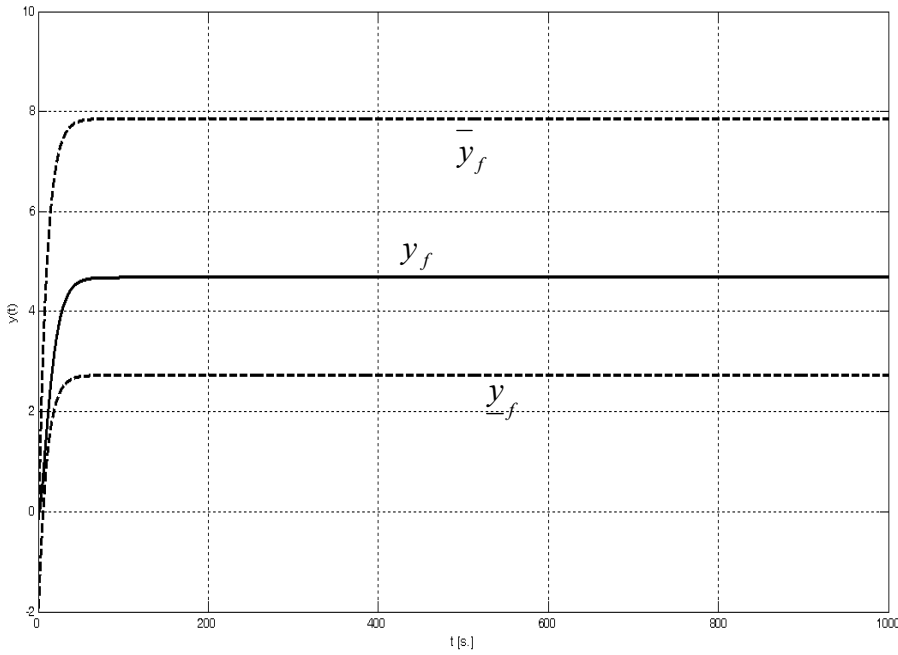
**f- y20 çıkışı**

**Şekil 3.7 : SRS girişler ve bunlara ilişkin çıkışlar.**



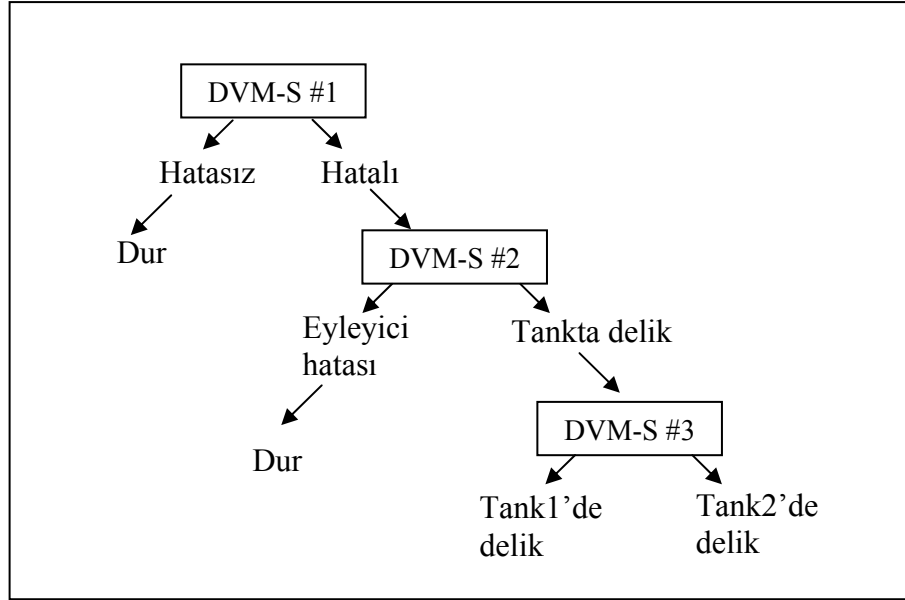
**Şekil 3.8 :** Hatasız çalışma bölgesi ve sınırları.

Herhangi bir hatasız çıkış belirlenen sınırlar arasında kalacaktır. Sınırların, normal çalışma koşullarındaki olası en kötü durumları ifade ettiği söylenebilir. Bunu genelleştirmek için hatasız çalışma bölgesinin sınırları, bu noktalara karşılık gelen girişlerle beraber eğitim verisi olarak alınır ve DVM-B'ler eğitilir. Hatasız bir sistemin basamak cevabı, kestirilen alt ve üst sınırlar arasında kalacaktır (Şekil 3.9).



**Şekil 3.9 :** Hatasız sistemin basamak cevabı ve kestirilen sınırlar.

Hata bulma eğitiminden sonra hata tanıma eğitimi de çevrim-dışı yapılıır. Buradaki dört durum (bir hatasız ve üç hatalı,  $n = 4$ ) için üç adet DVM-S kurulur (Şekil 3.10). Bir hatanın türünün ayırtını yapmak için DVM'nin “bire karşı diğerleri” çoklu sınıflandırma algoritması aşağıdaki gibi kurulur ve çalışır;



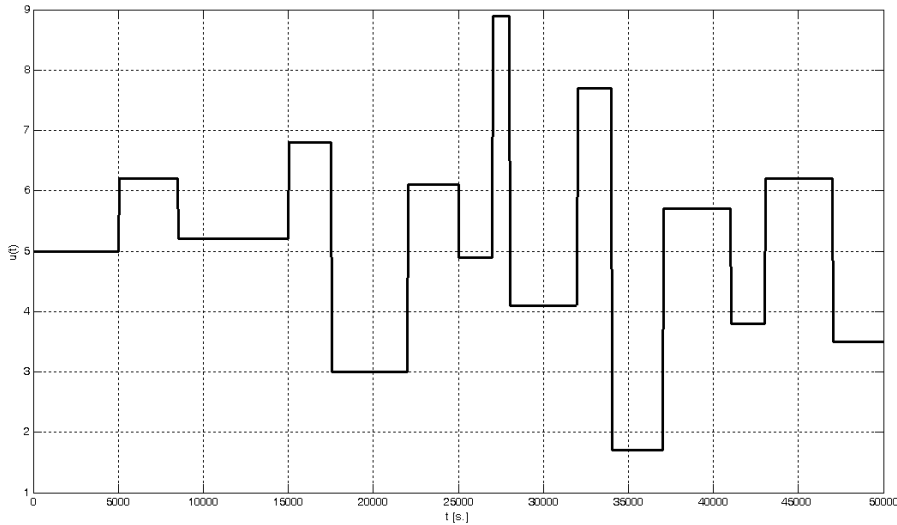
Şekil 3.10 : Dört durumun ayırımı için çoklu DVM-S algoritması.

1. Eğitim kümeleri oluşturulur; 20 SRS girişe karşı gelen hatasız çıkışlar, eyleyici hatalı sistem çıkışları, ve her bir tanktaki delik olma durumlarına ilişkin sistem çıkışları toplanır.
2. 3 adet DVM-S eğitimi aşağıdaki şekilde yapılıır:
  - İlk DVM-S ile hatasızlar +1 ile hatalıların tümü -1 ile işaretlenir.
  - İkinci DVM-S, eğer birinci DVM-S'nin sonucu -1 çıkarsa yani sistemde hata olduğu tespit edilirse çalışır ve hataların türünü ayırmak içindir. Sınıflandırma, eğer hata eyleyicideyse sonuç +1, hata tanklardan birinde delik varsa -1 ile ayrılarak yapılıır.
  - Üçüncü DVM-S, ikinci DVM-S'nin sonucu -1 olursa çalışacak şekilde düzenlenmiştir ve deliğin hangi tankta olduğunu tespit eder. Bu DVM-S, eğer delik birinci tankta ise +1, ikinci tankta ise -1 verir.
3. Yeni bir verinin testi şöyle yapılıır: Test işareti girilir. Veri hatalı değilse ekrana “Hata yok!” yazılıır ve program durur. Herhangi bir hata varsa “Hata var!” yazısı ekrana çıkar ve ikinci DVM-S çalışmaya başlar.

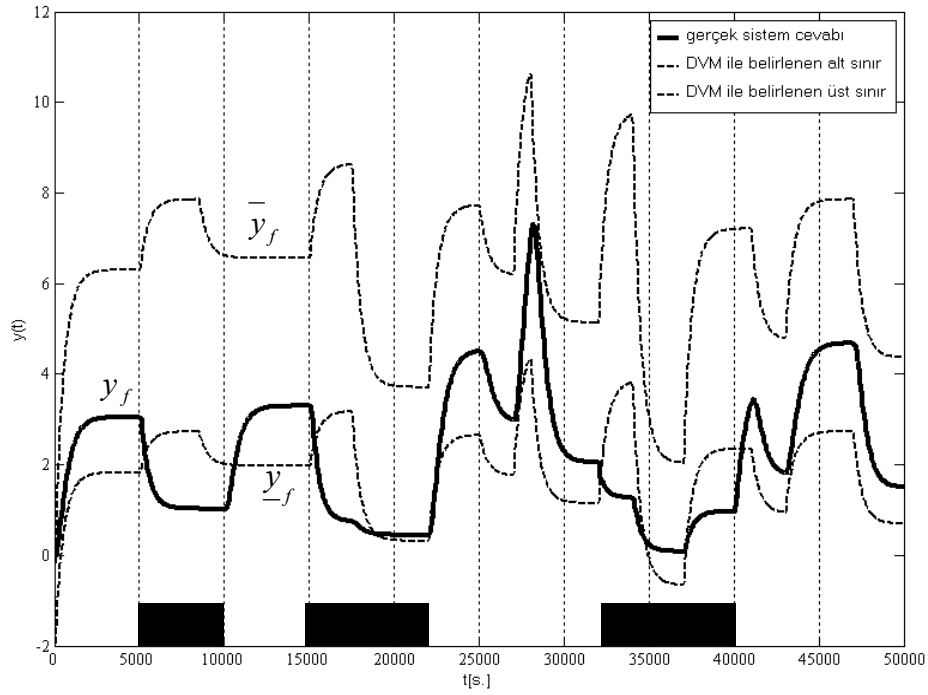
4. Eğer hata eyleyicide ise “Pompa hatası!” yazısı ekrana çıkar ve program durur. Eğer tankta delik varsa “Tankta delik var!” yazısı ekranda görülür ve üçüncü DVM-S çalışır.
5. Birinci tankta delik olması durumunda ekrana “Delik 1. tankta!” yazılır. İkinci tankta delik varsa ekrana “Delik 2. tankta!” yazılır.

DVM-S eğitimi için  $C=200$ ,  $\varepsilon = 0.0001$  ve kernel fonksiyonu olarak radyal tabanlı kernel ( $\sigma = 1$ ) seçilmiştir.

Test aşamasında,  $t = [5000-10000]$  s. aralığında birinci tankta  $r_1 = 4$ mm. yarıçaplı bir delik olduğu,  $t = [15000-22000]$  s. aralığında %30 luk bir eyleyici hatasının olduğu ( $K_f = 0.3$ ),  $t = [32000-40000]$  s. aralığında ikinci tankta  $r_2 = 5$ mm.yarıçaplı bir delik olduğu varsayılmıştır. Test girişi olarak eğitimde kullanılmayan bir sözde rasgele seri kullanılmıştır (Şekil 3.11). Belirtilen zaman aralıklarında gerçek sistem çıkışı, kestirilen sınırların dışına çıkar ve hata alarmı verilir (Şekil 3.12). Hata alarmı ile birlikte DVM-S mekanizması çalışmaya başlar ve hatanın türünü tespit eder.



**Şekil 3.11** : Uygulanan SRS işareti.



**Şekil 3.12** : Filtrelenmiş test çıkışı, kestirilen sınırlar, hata alarmı.

Hata alarmının verildiği bölgeler siyah bantla işaretlenmiştir. Hata alarmının verilmesiyle hata tanıma birimi devreye girer. Şekil 3.10'daki DVM-S tabanlı hata tanıma birimi, hata alarmından sonraki ilk örnek alma noktasında hatanın yerini tespit eder. Yöntemin gerçek zamanlı sınıflandırma yapabilmesi için örnekleme zamanının seçilen sınıflandırma bölgesinin sınıflandırılması için geçen süreden büyük seçilmesi gerekmektedir. DVM-S algoritmasının en büyük avantajlarından biri, hızlı tarama yapabilmesidir. Örneğin, 1000 noktalı verideki hata türü 0.1 s. gibi kısa bir zamanda belirlenir. Bu, aynı zamanda çevrim-içi çalışabilme imkanını sağlayan özelliktir.

### 3.5 Sonuçların Yorumlanması

Bir hata bulma ve tanıma yöntemi, sistemde oluşan hatayı doğru olarak tespit etmelidir. Çünkü hataların doğru olarak tespit edilmesi bazı durumlarda hayati önem taşıyabilir. Bu nedenle hata bulma ve tanıma sistemlerinde, hataların tanınması için güçlü bir izolasyon algoritması kullanılmalıdır. Burada önerilen HBT yönteminde hata bulma ve tanıma aşamaları ayrı ayrı gerçekleştirilir. Yöntemde, destek vektörü makineleri ile bağlanım mekanizması (DVM-B) hata bulma aracı ve destek vektörü

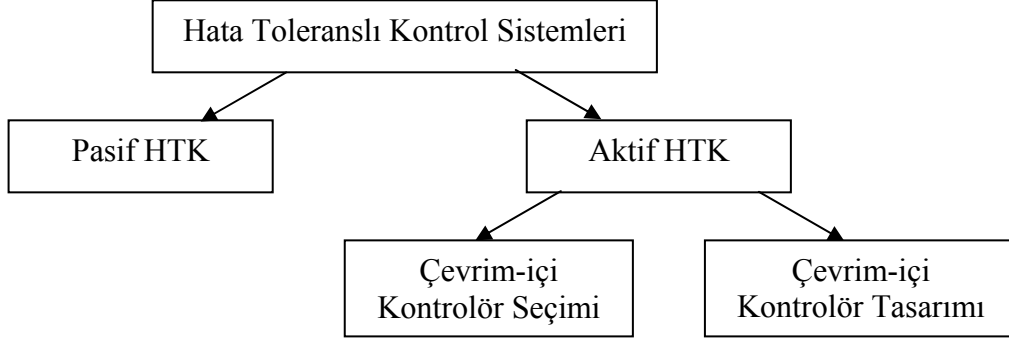
makinelere ile sınıflandırma mekanizması (DVM-S) hata tanıma aracı olarak kullanılmıştır. Hatasız sistem giriş-çıkış verileriyle oluşturulan güvenli çalışma bölgesinin alt ve üst sınırları iki farklı DVM-B ile modellenmiştir. DVM-B eğitimlerinde sınırları oluşturan veri çiftleri kullanılmıştır. Bu sayede hatasız sistemin herhangi bir giriş işaretine sistemin vereceği cevabın alt ve üst sınırlarının kestirilmesi sağlanmıştır. Eğer gözlenen çıkış işareti, eş zamanlı kestirilen bu sınırlardan en azından birini aşarsa sistemde bir hatanın varlığı tespit edilmiş olur. Verilen hata işareti ile birlikte DVM tabanlı çoklu sınıflandırıcı devreye girerek hatanın türünü tespit eder.

Yöntemin hızlı çalışması, az veriyle yüksek başarılı sınıflandırma ve kestirimlerin yapılabilmesi, hata tespiti için geçen sürenin çok kısa olması ve karmaşık algoritmalara, formüllere ve sistem modeline ihtiyaç duymaması en önemli avantajlarıdır. Kestirimde DVM'nin hafızasız özel bir türünün kullanılması da, hafıza, yer ve zaman tasarrufu açısından önemlidir.

## 4. DESTEK VEKTÖRÜ MAKİNELERİ İLE HATA TOLERANSLI KONTROL

### 4.1 Giriş

Günümüz modern teknoloji sistemlerinde artan güvenilirlik, emniyetlilik ve kurtarılabirlik gibi istekler, hata toleranslı kontrol (HTK) sistemlerine olan ilginin de artmasına neden olmuştur. HTK sistemleri, geleneksel geri beslemeli kontrol sistemlerinin bazı hatalı durumlarda gösterdiği kararsızlık ve başarımlı düşüklüğü gibi zayıflıkların üstesinden gelmek üzere geliştirilmiştir. Uçaklar, nükleer güç istasyonları, büyük kimyasal düzenekler gibi karmaşık sistemlerde küçük bir hata bile bazen yıkıcı sonuçlar doğurabilir. Bu sebepten, hataların etkisini telafı edecek kontrol sistemlerinin tasarlanması gerekmiştir. Hata telafisi kapasitesi olan bu tür kontrol sistemleri, genel olarak hata toleranslı olarak tanımlanır (Patton, 1997). Bir HTK sisteminden beklenen, hataları çok kısa bir sürede tespit edip uygun kontrolör düzenlemesiyle hatanın etkisini gidermesidir. Bu, sistemin normal çalışmasının devamı demektir ve oldukça önemlidir. Son yıllarda geliştirilen pek çok hata toleranslı kontrol yöntemi vardır. HTK sistemleri, pek çok parametreye göre sınıflandırılabilirler. Kullanılan matematiksel tasarım aracına (doğrusal karesel, sözde ters, akıllı kontrol, doğrusal parametre değişimi, model izleme, uyarlamalı kontrol, çoklu model, özyapı atama, geribesleme doğrusallaştırması,  $H_{\infty}$  ve diğer dayanıklı kontrol teknikleri, model öngörülü kontrol, nicel geribesleme teorisi, doğrusal matris eşitsizlikleri, değişken yapı kontrolü, kayan mod kontrolü, genelleştirilmiş içsel model kontrolü), tasarım yaklaşımına (önceden hesaplanan kontrol kuralı, çevrim-içi otomatik tasarım), yeniden yapılandırma mekanizmasına (eniyeleme, anahtarlama, eşleme, izleme, karşılama), ilgilendiği sistem türüne göre (doğrusal olan ve olmayan) sınıflandırma yapmak mümkündür. Genel kabul gören en temel sınıflandırma, HTK sistemlerinin aktif ve pasif olarak ikiye ayrılmasıdır (Şekil 4.1).



**Şekil 4.1** : Hata toleranslı kontrol sistemlerinin sınıflandırılması.

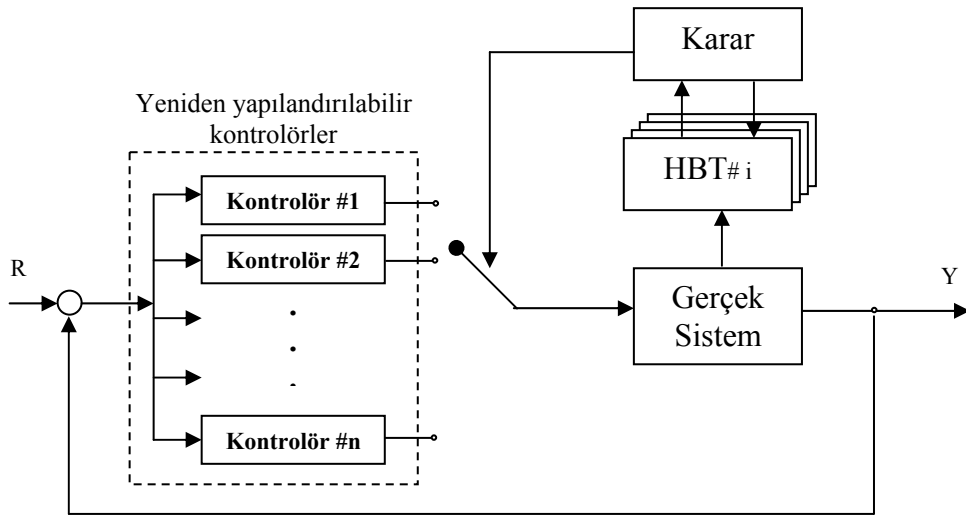
Belli hatalara karşı gürbüz olan pasif hata toleranslı kontrol sistemleri, ne HBT düzeneğine ne de kontrolör yeniden yapılandırmasına ihtiyaç duyarlar. Ancak hata telafisi kapasiteleri sınırlıdır (Hsieh, 2002, Jiang ve Zhao, 2000, Liang ve diğ., 2000, Liao ve diğ., 2002, Yang ve diğ., 2000, Yang ve diğ., 2001). Aktif hata toleranslı kontrol sistemleri ise yeniden yapılandırma mekanizmaları ile kontrol hareketini yeniden şekillendirerek hataların etkisiyle bozulan sistem cevabını tekrar istenen seviyeye getirme ilkesiyle çalışır (Chandler, 1984, Stengel, 1991, Monaco ve diğ., 1997, Belcastro ve Belcastro, 2001, Napolitano ve diğ., 1995, Polycarpou ve Vemuri, 1995, Theilliol ve diğ., 2002, Yen ve Ho, 2003, Maybeck ve Stevens, 1991, Rauch, 1995, Zhang ve Jiang 2001a, Zhang ve Jiang 2001b, Zhang ve Jiang 2002). AHTK sistemlerinin tasarımlarının en önemli konusu, sistemin hem geçici hem de sürekli durum başarımını, sadece normal çalışma koşullarında değil, hatalı koşullarda da koruyabilmektir. Değişik uygulamalar ve farklı amaçlar çeşitli AHTK geliştirilmiştir (Zhang ve Jiang, 2008). Gelişen teknoloji ve yazılımlar sayesinde, donanım yükünün azaltılmasında önemli ilerlemeler sağlanmıştır (Tortora, 2002, Jiang ve Chowdhury, 2005, Jiang ve Zhang, 2006, Huo ve Fang, 2007). Özellikle sinir ağları, bulanık mantık ve makine öğrenimi gibi akıllı algoritmalar, bu ilerlemenin en çok kullanılan araçları olmuşlardır. Bu algoritmaların en büyük avantajı doğrusal olmayan sistemlerle çalışmak için elverişli olmalarıdır. Makine öğrenimlerinden olan destek vektörü makineleri, bu avantajlara ek olarak problemlere uyarılma kolaylığına sahiptir. Ayrıca, örneğin sinir ağlarındaki katman ve yapı düzenlemesinin olmaması ve içerdiği eniyileme mekanizmasının bölgesel eniyilere takılmaması gibi üstünlükleri de mevcuttur.

Bu tezde geliştirilen DVM ile HTK yöntemi bir aktif hata toleranslı kontrol (AHTK) yöntemidir. Yöntem, kontrolörün yeniden yapılandırma aşamasında çevrim-içi kontrolör seçimi mekanizmasını kullanır. Hata bulma ve tanıma aşamasında DVM

tabanlı çoklu sınıflandırma algoritması ve yeniden yapılandırma alt sisteminde PID tipli kontrolörler kullanılmıştır. Kontrolör parametreleri genetik arama algoritmasıyla ITSE başarıml ölçütüne göre aranmıştır (Ortaç-Kabaoğlu ve diğ., 2009b).

#### 4.2 Destek Vektörü Makineleri ile Hata Bulma ve Tanıma

Yöntem, esas olarak üç alt birimin bir arada çalışmasıyla işler. Bunlar; hata bulma ve tanıma birimi, karar birimi ve yeniden yapılandırılabilir kontrolör birimidir (Şekil 4.2). Hata bulma ve tanıma, hata toleranslı kontrolün çok önemli olan ilk adımdır. Hassas bir sınıflandırma algoritması olan DVM'nin hataları tanıma becerisi, 3.2 bölümündekine benzer şekilde burada da kullanılmıştır.



Şekil 4.2 : Önerilen HTK yapısı.

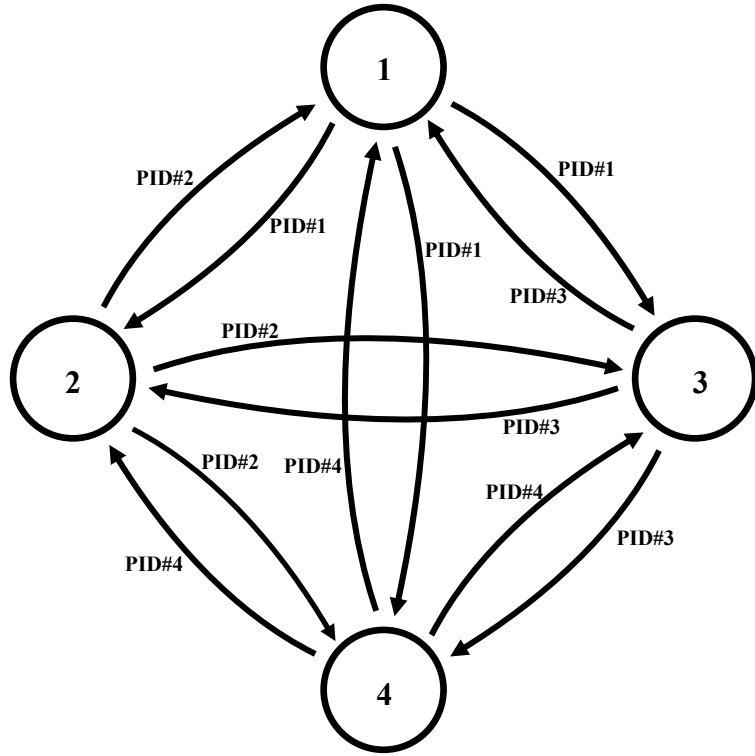
#### 4.3 Yeniden Yapılandırılabilir Kontrolör Tasarımı

Hata toleranslı kontrol mekanizmasının kontrolör yeniden yapılandırma birimi, hatanın ve yerini tespitiyle birlikte çalışmaya başlar. Yeniden yapılandırma, gözlemlenen sistemde bir hata tespit edildiğinde hatanın neden olduğu sapmayı giderip sistemi normal çalışma davranışına geri getirecek kontrolörün devreye sokulması olarak özetlenebilir. Bu çalışmada, hatanın türüne göre önceden katsayıları tasarlanmış PID kontrolörlerden uygun olanını devreye sokacak akıllı bir yapı tasarlanmıştır. Bu yapıda öncelikle, belirli her bir hatalı durumun etkisini gidererek sistemi normal çalışma durumunda tutacak PID kontrolörlerin katsayıları çevrim-dışı

bulunur. PID kontrolör tasarımları için bu çalışmada *genetik algoritma* ve ITSE başarıml ölçütü kullanılmıştır (4.1). Genetik algoritmalar ile katsayıları belirlenen kontrolörler, Şekil 4.2 de verilen hata toleranslı kontrol yapısına yerleştirilir.

$$J = \int te^2(t)dt \quad (4.1)$$

Sistem, belirlenen hatanın türüne göre uygun kontrolörü seçer ve onu devreye sokar. Burada dikkat edilecek hususlardan biri de şudur; bir hatalı durumun arkasından bir diğerinin gelmesi olasılığı her zaman vardır. Dolayısıyla, hatalı ve hatasız durumların toplam sayısı  $n$  ise  $n*(n-1)$  tane olasılık var demektir (Şekil 4.3). Bu yüzden sınıflandırma eğitimleri, hatalı, hatasız her durum için ayrı yapılır. Örneğin 1 numara hatasız durumu ve 2, 3, 4 numaralar da hatalı durumları temsil etsin. Öncelikle hatasız durum için belirlenen kontrolör devredeyken 4 tane durumu ayıracak sınıflandırma eğitimi yapılır. Bu eğitim ile sistem 1 durumunda çalışmasına devam ederken ve dolayısıyla PID#1 devredeyken 2, 3, 4 durumlarından biri gerçekleştiğinde ayırımın yapılması sağlanır. Benzer eğitimler tüm durumlar için çevrim-dışı gerçekleştirilir. Böylece iki farklı hatanın birbirini izleme ihtimaline karşı önlem alınmış olur. Bu önlem, çevrim-içi çalışan hata bulma ve tanıma biriminin güvenilirliği için gereklidir. Sistem herhangi bir  $i$  durumunda çalışırken o duruma ait PID# $i$  kontrolörü ve HBT# $i$  birimi devrededir. HBT# $i$  birimi, belirli aralıklarla sistem çıkışını kontrol ederek sapma olup olmadığını kontrol eder. Bir sapma tespit edilmişse hatanın türünü belirleyerek karar birimine iletir. Karar birimi, hatanın türüne ait olan ve sistemi istenen çalışma koşulunda tutacak kontrolörü seçerek devreye sokar. Aynı zamanda yeni duruma ait HBT birimi de çalışmaya başlayarak gelen çıkış verilerini gözler.



Şekil 4.3 : Dört durum için olası yollar ( $n = 4$ ).

Sunulan HTK sisteminin benzetimi yine çift tanklı sıvı seviye sisteminde gerçekleştirilmiştir.

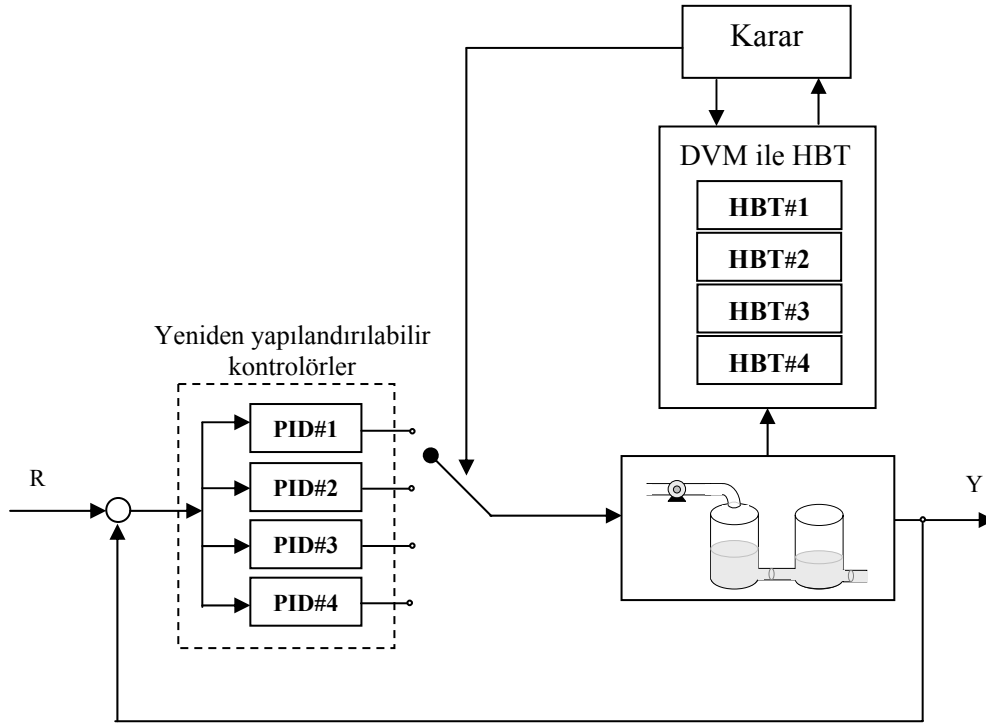
#### 4.4 Benzetim Örneği

Yöntemin, 3.4 bölümünde verilen çift tanklı sıvı seviye sisteminde uygulaması yapılmıştır. Sırasıyla (3.8), (3.9), (3.10) bağıntıları ile verilen pompada eyleyici hatası, birinci tankta delik ve ikinci tankta delik hataları göz önüne alınmıştır. (3.4) bağıntısıyla verilen hatasız sistem durumu ile birlikte toplam dört farklı durum için önerilen HTK yöntemi uygulanmıştır.

##### 4.4.1 Hata bulma ve tanıma eğitimi

Yöntemde, hata bulma ve tanıma birimi destek vektörü makineleri tabanlıdır. Destek vektörü makinelerinin eğitimi yapılmadan önce (3.4), (3.8), (3.9), (3.10) bağıntıları ile ifade edilen hatasız ve üç farklı hatalı sistemin çıkışlarını referans girişe en az yanılğı ile getirecek PID kontrolörlerin katsayılarının ayarlanması gerekir. Bunu sağlamak için ITSE amaç fonksiyonunu minimize edecek  $K_p, K_i, K_d$  kontrolör

katsayıları genetik algoritma yöntemiyle çevrim-dışı aranmıştır. Bulunan kontrolör katsayıları Şekil 4.4'deki HBT yapısına yerleştirilir.



Şekil 4.4 : Çift tank sisteminin HTK yapısı.

Dört durum için genetik arama algoritmasıyla bulunan PID katsayıları aşağıdaki gibidir.

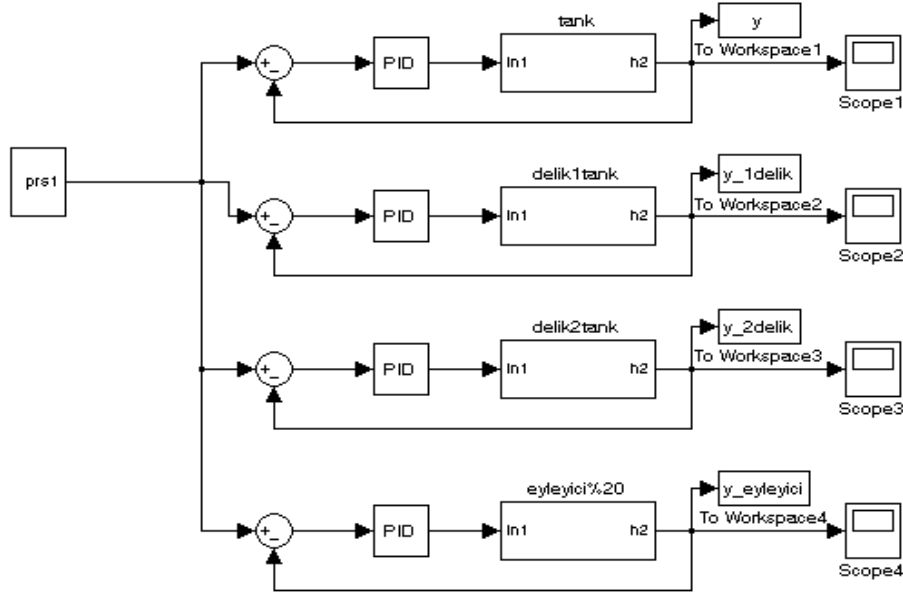
Çizelge 4.1 : Genetik algoritmayla bulunan PID katsayıları.

	PID #1	PID #2	PID #3	PID #4
$K_p$	13.01	17.6342	620.58	18.3674
$K_i$	0.0441	1.357	1.285	0.0651
$K_d$	0.0145	0.4037	0.82017	1.9879

Burada PID#1, PID#2, PID#3, PID#4 kontrolörleri sırasıyla, hatasız, birinci tankta delik, ikinci tankta delik ve eyleyici hatası durumlarına aittir. Tanklardaki deliklerin yarıçaplarının 7mm. ve eyleyici hatasının %30 olduğu varsayılmıştır.

Eğitim verilerinin toplanması için kontrolörler tüm durumlar için çalıştırılır. Böylece arda arda farklı hata türleri oluşmasına hazırlıklı olunur. Burada, genelleştirme sağlanması için 20 adet sözde rasgele seri (SRS) işareti kullanılmıştır. Şekil 4.5 bir *i*

kontrolörü devredeyken bir SRS işaretiyle veri toplama düzeneğini göstermektedir. Örnekleme zamanı 50s.dir. Tüm eğitim verileri toplanıp herbir HBT alt biriminin oluşturan çoklu DVM-S yapısı çevrim-dışı eğitilir. DVM-S eğitimi için  $C=200$ ,  $\varepsilon = 0.0001$  ve kernel fonksiyonu olarak radyal tabanlı kernel ( $\sigma = 1$ ) seçilmiştir. Tüm benzetimler Matlab-Simulink programında gerçekleştirilmiştir.

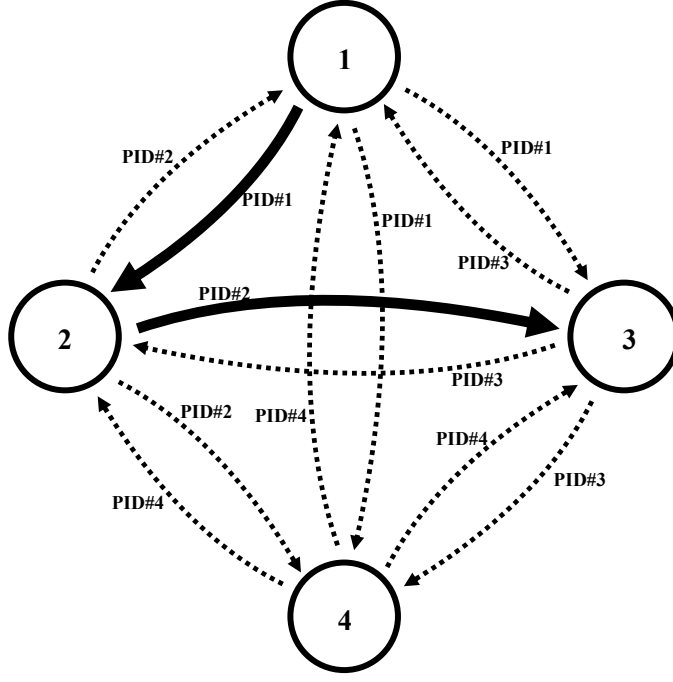


Şekil 4.5 : PID #i ile veri toplama düzeneği.

#### 4.4.2 Karar verme, uygun kontrolörün seçimi

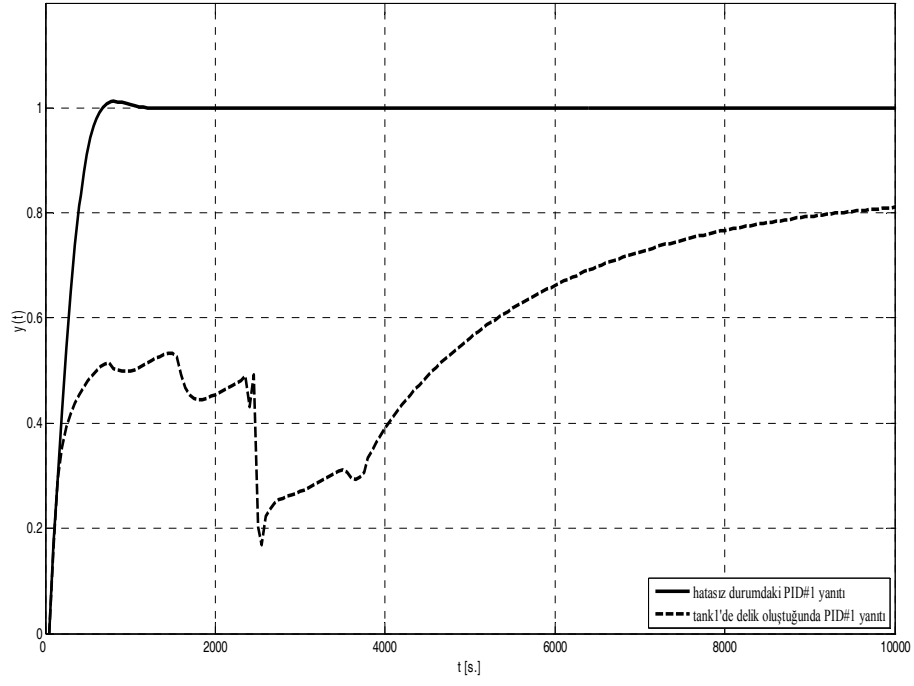
Buraya kadar yapılan tüm katsayı arama, veri toplama ve eğitimler çevrim-dışı gerçekleştirilir ve HTK yapısında yerleştirilir. Burada, sistemin hatasız konumda çalışmaya başladığı varsayımı yapılmıştır. Kapalı çevrimli sistem PID#1 den gelen işaretle çalışmasına devam eder. HBT birimi, kapalı çevrim cevabını uygun sıklıkla gözlemleyerek hata arar. HBT birimi tarafından bir hata tespit edildiğinde karar birimi uygun kontrolörü seçerek devreye sokar. Seçilen hata senaryosu ve yöntemin işleyişi aşağıdaki gibidir.

*Birinci hata senaryosu:* Sistem hatasız çalışmaya başlar. Önce tank1'de, daha sonra tank2'de delik oluşur. Yöntem, Şekil 4.6'deki yolu izleyerek hataları telafi eder.

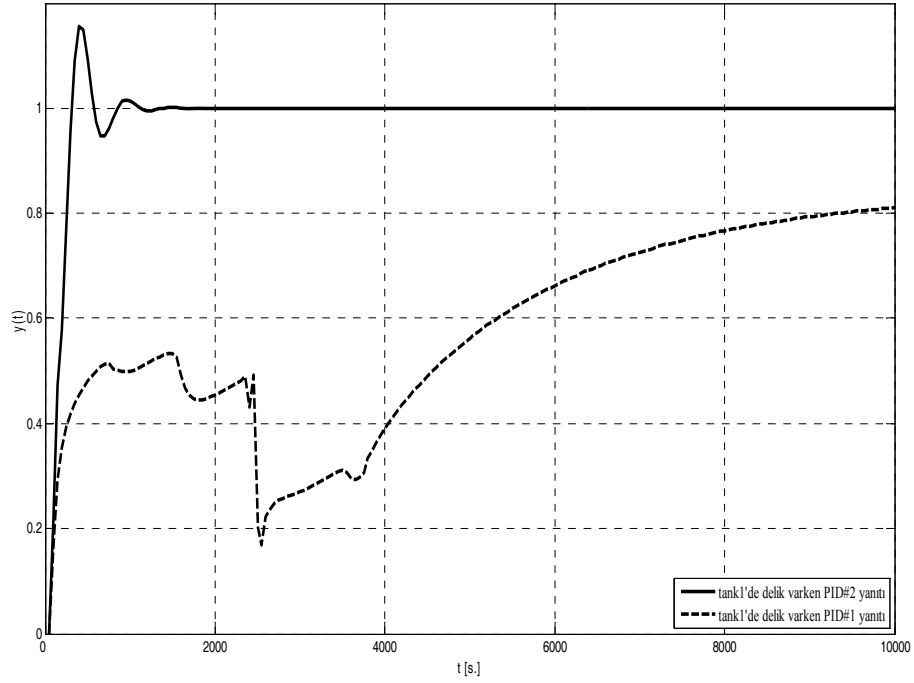


**Şekil 4.6 :** Yöntemin tank1’de delik – tank2’de delik hataları için izleyeceği yol.

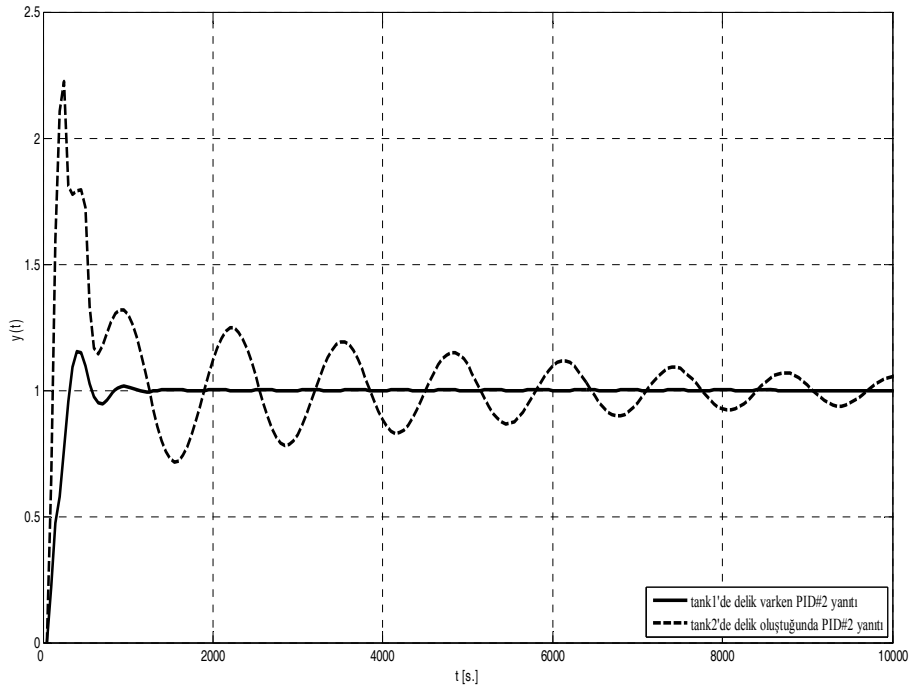
*Önerilen DVM ile HTK sisteminin işleyişi :* Tank1’de delik oluştuğunda sistem cevabı istenen değerlerden sapar (Şekil 4.7). HBT#1 birimi hatanın türünü belirler ve bilgiyi karar birimine gönderir. Hata tank1’de olduğu için uygun olan PID#2 kontrolörü seçilir. Böylece hatanın etkisi giderilmiş olur (Şekil 4.8). Aynı zamanda artık HBT#2 alt birimi çalışmaya başlar. Tank2’de delik oluştuğunda (Şekil 4.9) HBT#2, hatanın türünü belirler ve karar mekanizmasına bilgiyi yollar. Benzer şekilde un uygun kontrolör PID#3 seçilir ve sistem cevabı istenilen yere geri çekilir (Şekil 4.10). Beraberinde HBT#3 çalışmaya başlar.



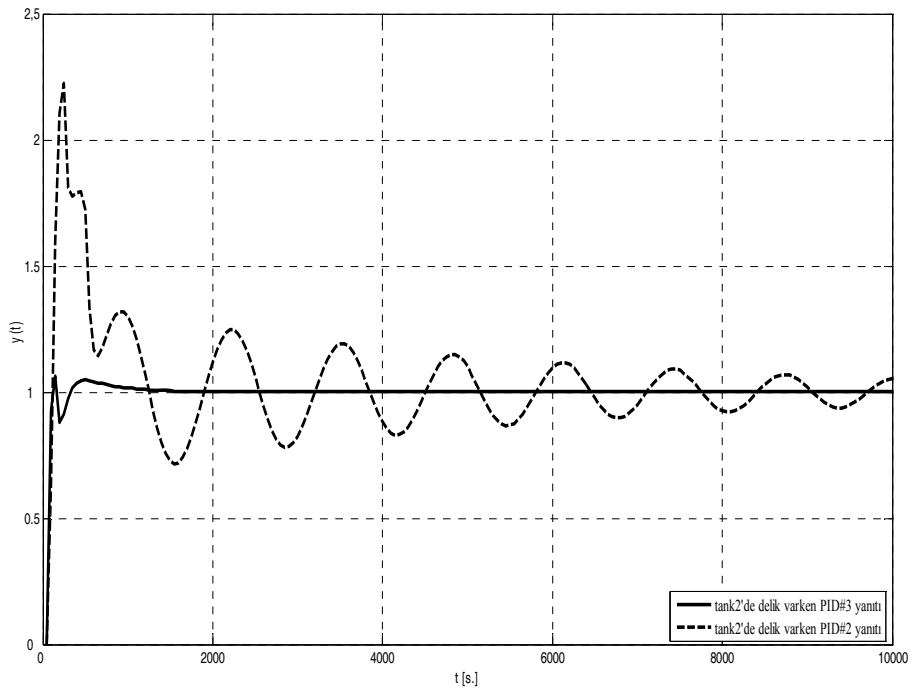
**Şekil 4.7 :** Tank1'nin delinmesi durumunda PID#1 devredeyken hatalı ve hatasız durum sistem yanıtları.



**Şekil 4.8 :** Tank1'nin delinmesi durumunda PID#1 ve PID#2 devredeyken sistem yanıtları.

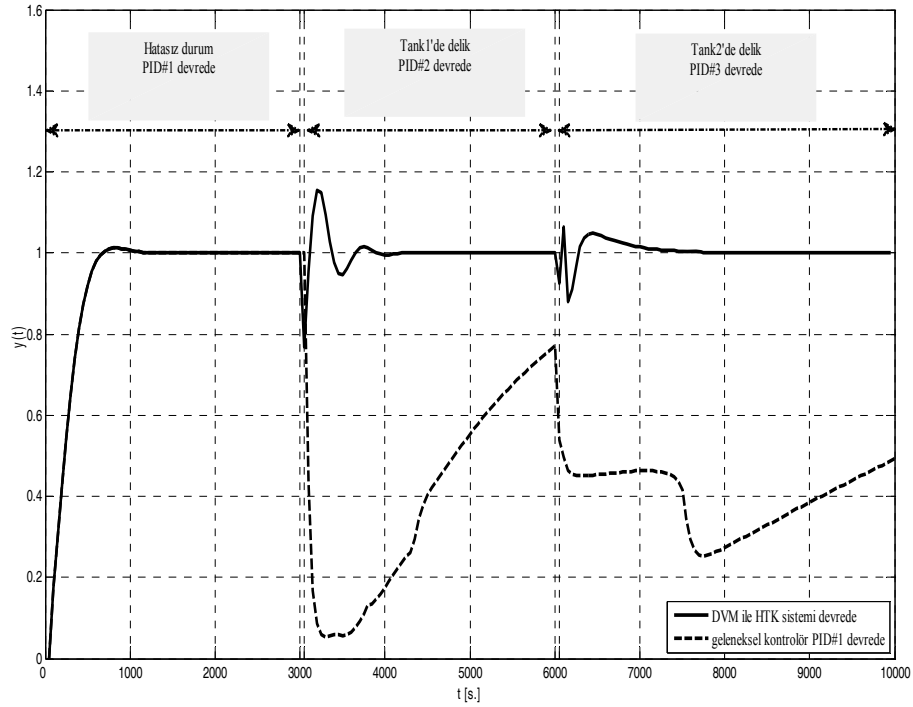


Şekil 4.9 : Tank2'nin delinmesi durumunda PID#2 devredeyken sistem yanıtları.



Şekil 4.10 : Tank2'nin delinmesi durumunda PID#2 ve PID#3 devredeyken sistem yanıtları.

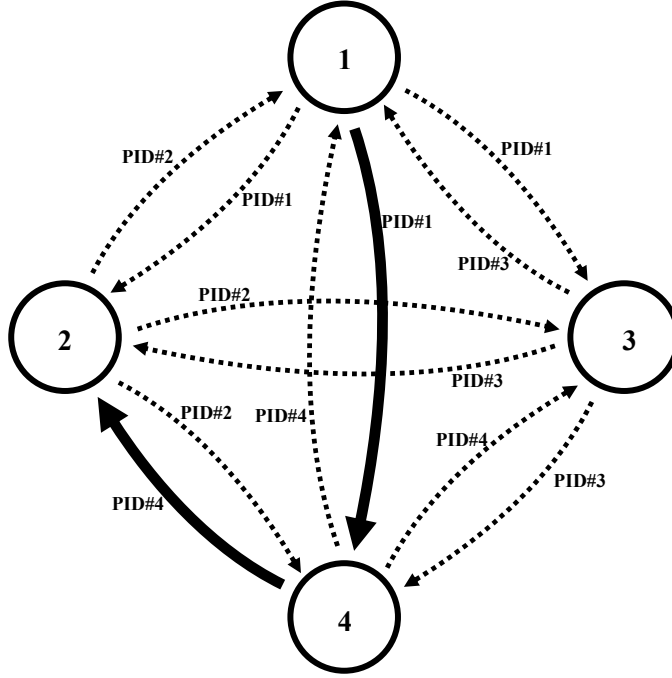
İlk hata seneryosu şu şekilde detaylandırılmıştır: Sistem hatasız çalışmaya başladıktan sonra,  $t=3000s.$  de birinci tankta delik hatası oluşur. Hata oluşuktan sonraki ilk örnek almada, hata bulma ve tanıma birimi hatayı ve türünü tespit eder ve ilgili kontrolör olan PID#2'yi devreye sokar. Sistem toparlanır ve istenilen davranışına geri döner.  $t=6000s.$  de ikinci tankta delik hatası oluşur. Benzer şekilde hata belirlenir ve PID#3 devreye girer ve sistem beklenen davranışına döner (Şekil 4.11).



**Şekil 4.11 :** Yöntemin ard arda gelen tank1’de delik - tank2’de delik hatalarında işleyişi ve geleneksel kontrolörle sistem yanıtı.

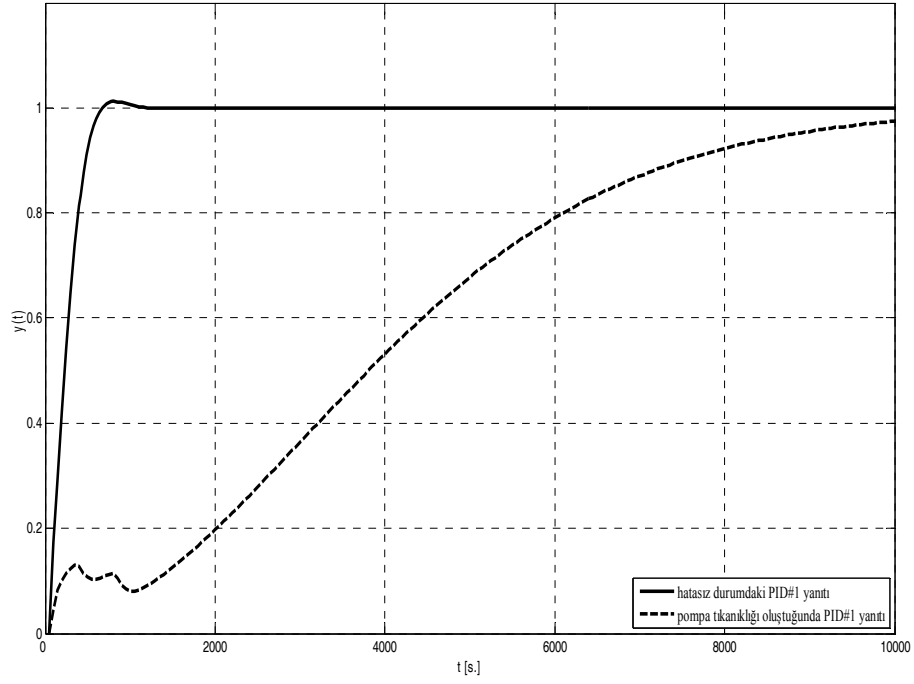
Kapalı çevrimli sistem, hatasız durum için tasarlanan kontrolörü ile bu hataları toparlayamaz. Hata toleranslı kontrol sistemlerinin geliştirilmesindeki amaç, bu gibi durumlara başa çıkmaktır. Belli sıklıklarla gözlenen sistem çıkışından gelen bilgilerle kısa sürede hata tespiti yapılmış ve hatanın etkisi doğru kontrolör seçilerek giderilmiştir. Yeni bir hatanın gelişiminden sonra aynı işlem yeni hataya göre tekrarlanmıştır. Şekil 4.11, önerilen HTK sistemi devredeyken ve sadece hatasız durum kontrolörü devredeyken sistem cevaplarının nasıl olduğunu göstermektedir.

*İkinci hata senaryosu:* Sistem hatasız çalışmaya başlar. Önce pompada tıkanıklık, daha sonra tank1’de delik oluşur. Yöntem, Şekil 4.12’deki yolu izleyerek hataları telafi eder.

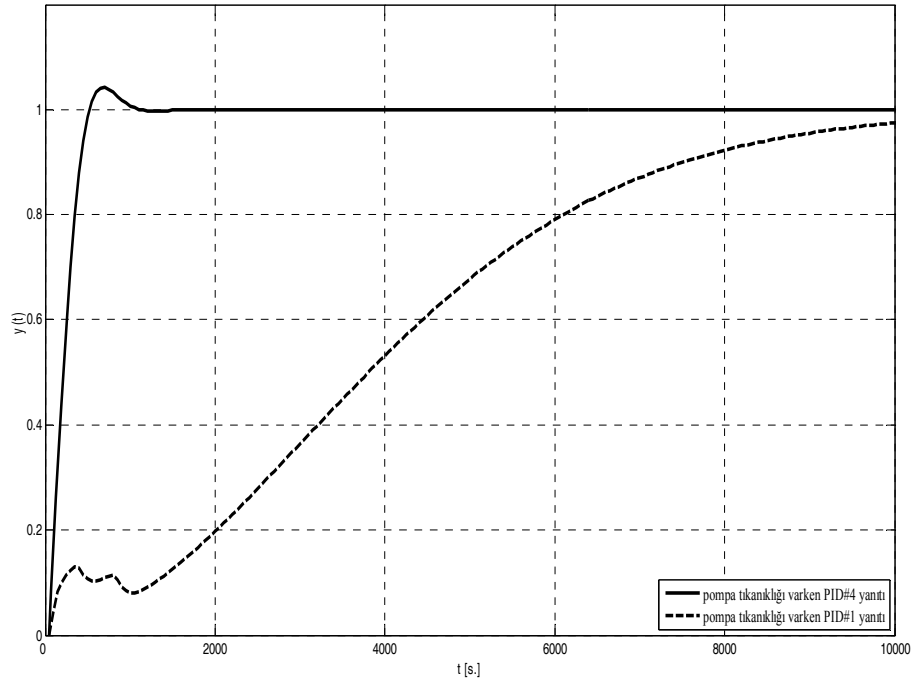


**Şekil 4.12 :** Yöntemin pompada tıkanma - tank1’de delik hataları için izleyeceği yol.

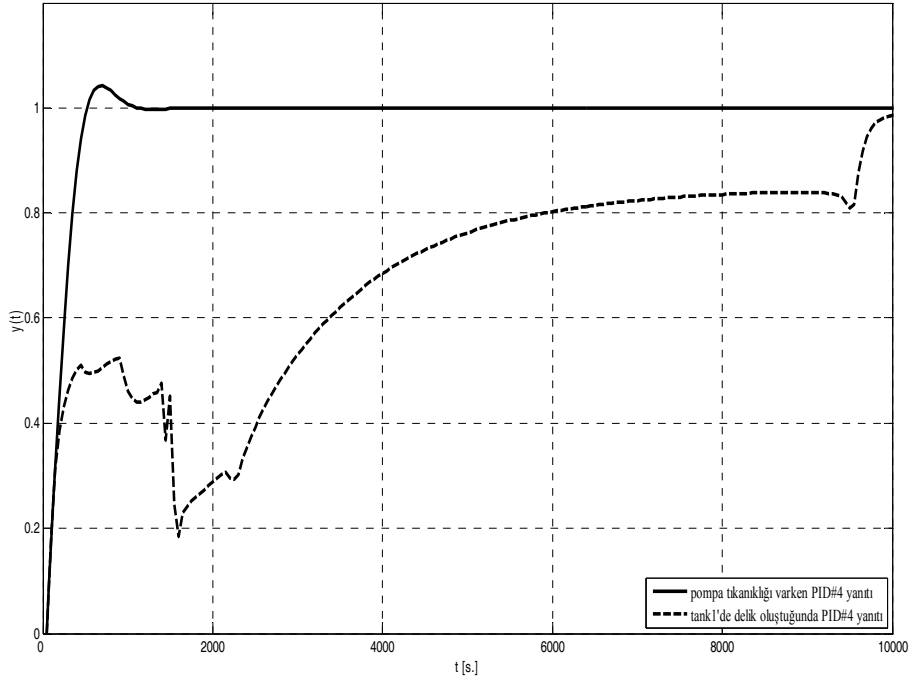
*Önerilen DVM ile HTK sisteminin işleyişi :* Pompada tıkanıklık oluştuğunda sistem cevabı istenen değerlerden sapar (Şekil 4.13). HBT#1 birimi hatanın türünü belirler ve bilgiyi karar birimine gönderir. Hata pompada olduğu için uygun olan PID#4 kontrolörü seçilir. Böylece hatanın etkisi giderilmiş olur (Şekil 4.14). Aynı zamanda artık HBT#4 alt birimi çalışmaya başlar. Tank1’de delik oluştuğunda (Şekil 4.15) HBT#4, hatanın türünü belirler ve karar mekanizmasına bilgiyi yollar. Benzer şekilde un uygun kontrolör PID#2 seçilir ve sistem cevabı istenilen yere geri çekilir (Şekil 4.16). Beraberinde HBT#2 çalışmaya başlar.



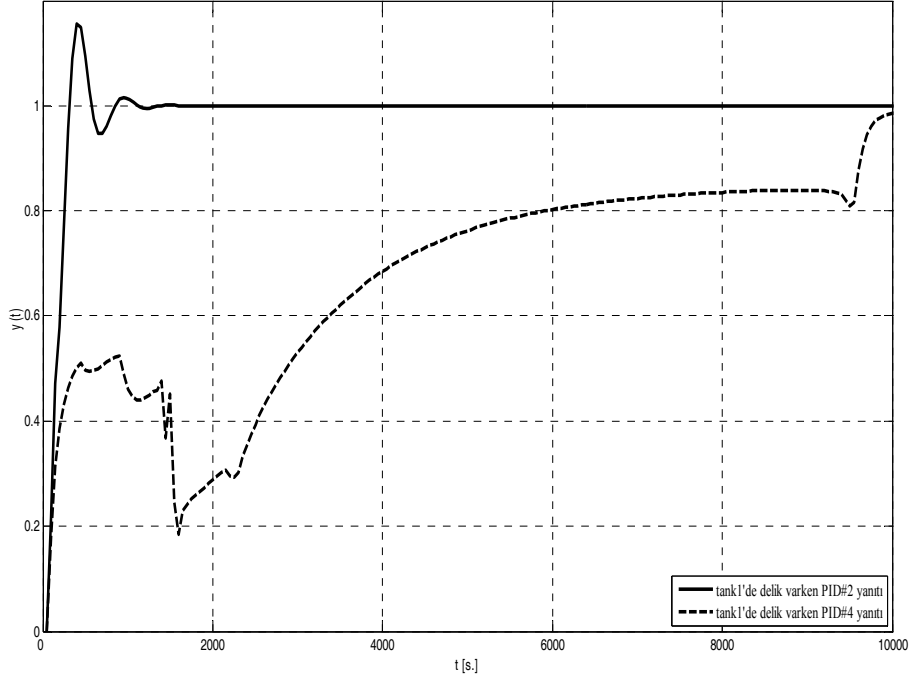
Şekil 4.13 : Pompa tıkanıklığı oluştuğunda PID#1 devredeyken hatalı ve hatasız durum sistem yanıtları.



Şekil 4.14 : Pompa tıkanması durumunda PID#1 ve PID#4 devredeyken sistem yanıtları.



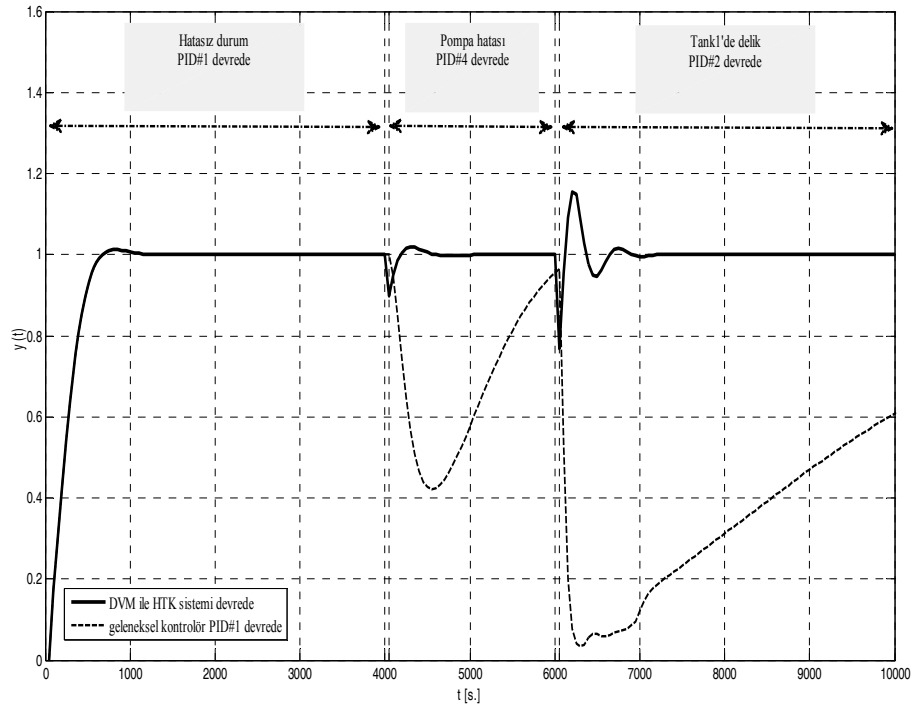
Şekil 4.15 : Tank1'nin delinmesi durumunda PID#4 devredeyken sistem yanıtları.



Şekil 4.16 : Tank1'nin delinmesi durumunda PID#4 ve PID#2 devredeyken sistem yanıtları.

İkinci hata seneryosu şu şekilde detaylandırılmıştır: Sistem hatasız çalışmaya başladıktan sonra,  $t=4000s.$  de pompada tıkanma hatası oluşur. Hata oluşuktan sonraki ilk örnek almada hata bulma ve tanıma birimi hatayı ve türünü tespit eder ve ilgili kontrolör olan PID#4'ü devreye sokar. Sistem toparlanır ve istenilen davranışına geri döner.  $t=6000s.$  de birinci tankta delik hatası oluşur. Benzer şekilde hata belirlenir ve PID#3 devreye girer ve sistem beklenen davranışına döner (Şekil 4.17).

Şekil 4.17, pompada hata – tank1'de delik hataları için, önerilen HTK sistemi devredeyken ve sadece hatasız durum kontrolörü devredeyken sistem cevaplarının nasıl olduğunu göstermektedir.



**Şekil 4.17 :** Yöntemin ard arda gelen pompada tıkanıklık - tank1'de delik hatalarında işleyişi ve geleneksel kontrolörle sistem yanıtı.

#### 4.5 Sonuçların Yorumlanması

Hata toleranslı kontrol algoritmaları, sistemde kapalı çevrimin toparlayamadığı hatalı durumların telafisi için geliştirilen yöntemlerdir. Geleneksel kapalı çevrimli kontrol sistemleri, özellikle doğrusal olmayan dinamik sistemlerin kontrolünde bazen yetersiz kalmaktadır. Bu yetersizliğin telafisi için geliştirilen pek çok HTK yöntemi vardır. Farklı uygulamalar veya amaçlar için geliştirilen çok sayıda HTK yönteminin ortak amacı, sistemi istenen başarımlı seviyesinde tutmaktır. HTK yöntemleri, hatayı bulma, tanıma, karar verme ve hatanın etkisini telafi etme aşamalarında izlenen yollar, kullanılan yöntemler ve araçları ile birbirlerinden ayrılırlar. Burada geliştirilen DVM ile HTK yöntemi, çevrim-içi çalışan bir yeniden yapılandırma birimi içeren aktif hata toleranslı kontrol yöntemlerindedir. DVM'nin çoklu sınıflandırmadaki ve genetik algoritmaların aramadaki yüksek başarımlı, DVM ile HTK sisteminin temelini oluşturmaktadır. Destek vektörü makinelerinin sınıflandırma algoritması, sistemdeki çok sayıda hatanın bulunup ayrılabilmesi için oldukça uygun bir yöntemdir. Önerilen HTK yöntemi, kontrolörün yeniden yapılandırma aşamasında çevrim-içi kontrolör seçimi mekanizmasını kullanır. Hata bulma ve tanıma biriminden gelen bilgiye göre karar birimi, önceden tasarlanmış kontrolörlerden en uygun olanını seçer. Hata bulma ve tanıma biriminde DVM tabanlı çoklu sınıflandırma algoritması ve yeniden yapılandırma alt sisteminde PID tipli kontrolörler kullanılmıştır. Kontrolör parametreleri genetik arama algoritmasıyla ITSE başarımlı ölçütüne göre aranmıştır. Yöntemde DVM sınıflandırıcıları (DVM-S), hatasız ve önceden belirlenmiş hatalı sistem çıkışları ile çevrim-dışı olarak eğitilmiştir.

Yöntem, özellikle olası hataların bilindiği HTK problemlerinin çözümü için güçlü bir araç sunmaktadır. Destek vektörü makineleri tabanlı HBT birimi sadece sistem çıkışını gözlemleyerek hataları tespit eder ve bilgiyi karar birimine gönderir. Karar birimi, türü tespit edilen hatanın etkisini giderecek kontrolörü seçerek devreye sokar. Burada HBT biriminin eğitimleri, farklı hataların birbirini izlemesi ihtimalleri göz önüne alınıp yapılarak yöntemin güvenilirliği artırılmıştır. Yöntemin başarımlı, çift tanklı sıvı seviye kontrol sisteminde olası hatalı durumlar için incelenmiş ve büyük hatalar için bile başarımlı yüksek olduğu belirlenmiştir.

## **5. DESTEK VEKTÖRÜ MAKİNELERİ İLE DOĞRUDAN HATA TOLERANSLI KONTROL**

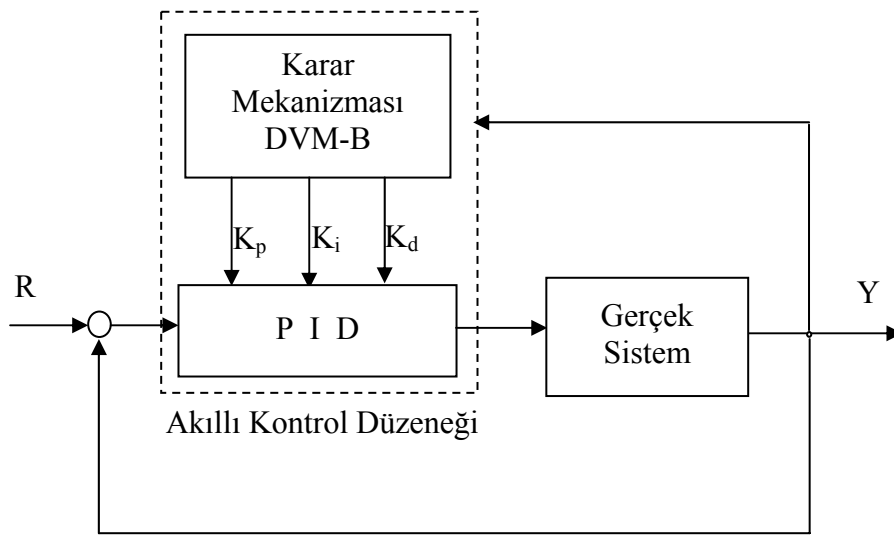
### **5.1 Giriş**

Geleneksel hata toleranslı kontrol sistemlerinde hatanın varlığı tespit edilir (*hata bulma*), yeri ve zamanı belirlenir (*hata tanıma*) ve en son olarak da bu hatanın yarattığı sapmaların etkisini yok ederek sistemi nominal davranışına geri getirecek düzenleme yapılır (*hata toleransı*). Sistemin güvenilirliğini sağlamak için sistemde oluşabilecek olası hatalara tolerans sağlayacak kontrol sistemini tasarlamak gerekmektedir. Kapalı çevrim düzeneği, gözlenen bir hatayı belirli ölçüde giderme yetisine sahiptir. Ancak kapalı çevrimin hatayı telafi edemediği durumlarda kontrolörün yeniden düzenlenmesi kaçınılmaz olabilir. Bu amaçla çeşitli aktif hata toleranslı kontrol teknikleri geliştirilmiştir. Daha önce de belirtildiği gibi bir AHTK sistemi, hata bulma ve tanıma, yeniden yapılandırılabilir kontrolör ve karar birimi olmak üzere üç alt sistemden oluşur. AHTK sistemlerinde yeniden yapılandırma mekanizmaları, çevrim-içi kontrolör seçen ve çevrim-içi kontrol parametresi hesaplayan olarak ikiye ayrılır. Bu bölümde önerilen destek vektörü makineleri ile doğrudan hata toleranslı kontrol yöntemi, akıllı bir yeniden yapılandırılabilir kontrolör birimi içerir. Amaç, sistemde oluşabilecek hatanın etkisini çevrim-içi çalışan bir düzenleme ile ortadan kaldırmaktır. Önerilen yöntemde, sistemin çıkışları belli aralıklarla gözlenir ve bu çıkışları istenen sistem davranışına getirecek PID tipi kontrolörün katsayıları çevrim-içi belirlenir. Katsayıların belirlenmesi için kullanılacak mekanizmanın hızlı ve genelleştirme becerisi yüksek bir yöntemle çalışması gerekmektedir. Bu amaçla bu özellikleri taşıyan destek vektörü makineleri ile bağlanım (DVM-B) kullanılmıştır. Yöntemin, çift tanklı sıvı seviye sisteminde uygulaması yapılmıştır. Sonuçlar oldukça iyidir.

### **5.2 DVM-B ile Doğrudan Hata Toleranslı Kontrol Yöntemi**

Hata Toleranslı Kontrol Sistemlerinde (HTKS) ilk aşamada olası hatalar tespit edilir, sonra kararlılığı ve kabul edilebilen bir başarıyı sağlayacak yeniden düzenlemeyi

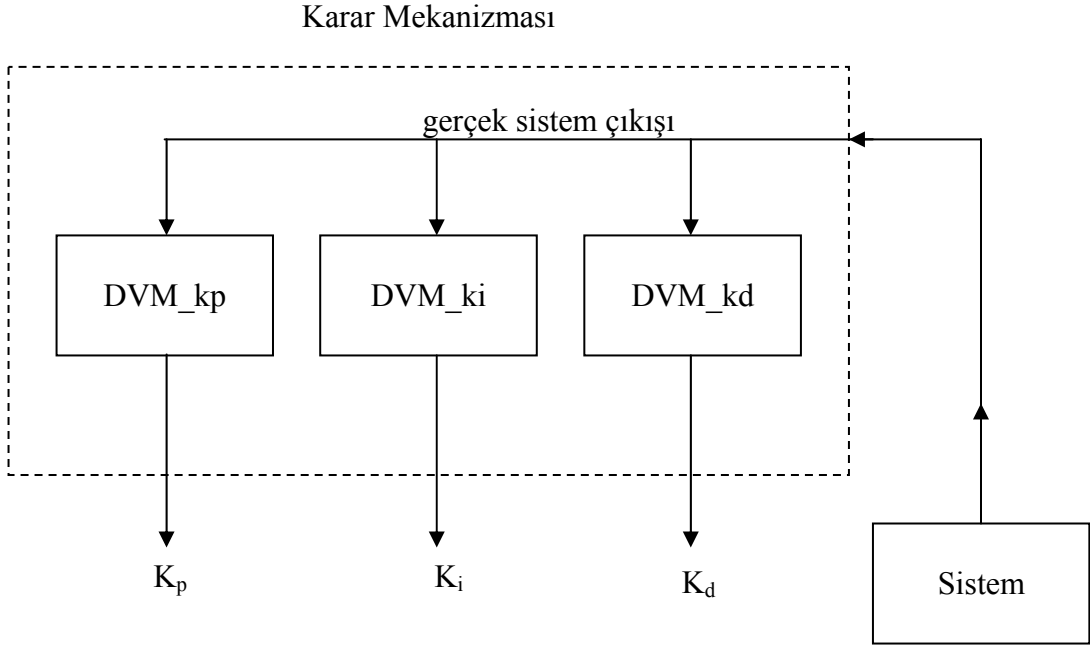
gerçekleştirmesi için akıllı bir karar mekanizması devreye girer. Bununla, tespit edilmiş duruma en uygun olan önceden tasarlanmış kontrolör devreye sokulur. Bu yöntemle hatanın varlığı, yeri ve zamanı hakkında bilgi edinilir, uygulanacak kontrolör önceden düşünülen olası hataların her biri için tasarlanarlardan uygun olanı olur. Bu çalışmada önerilen destek vektörü makineleri ile doğrudan hata toleranslı kontrol (DHTK) yöntemi, hatanın etkisini giderecek düzenleme yapması için, hata bulma, hata tanıma aşamalarına gereksinim duymaz. DHTK yönteminde bir akıllı kontrolör sistemi kurulur. Bu yöntemde kontrolör katsayıları, sistem cevabına göre çevrim-içi ayarlanır ve sistemin istenen davranışa götürülmesi sağlanır. PID tipi kontrolör katsayıları, destek vektörleri makineleri ile bağlanım (DVM-B) kullanılarak oluşturulan akıllı bir algoritma ile ayarlanır. Belli aralıklarla alınan sistem cevabı, bu akıllı algoritmanın girişi, uygun kontrolör katsayıları da çıkışıdır (Ortaç-Kabaoğlu ve diğ., 2009c). HTK sistemlerinin uygulamaları farklı amaçlar için farklı olabilir. Bu yöntemde, hatanın varlığının veya türünün belirlenmesinin bir işlevi bulunmamaktadır. Sistem cevabının istenildiği gibi olması için sadece gerçek sistem çıkışı ile çalışan akıllı bir kontrolör sistemi kullanılır (Şekil 5.1). İstenirse hatanın yerinin belirlenmesi de sağlanabilir. Bu çalışmada, hatanın yerine bakmadan düzenlemeyi sadece sistem cevabındaki sapmaya göre çevrim-içi gerçekleştiren akıllı bir sistemin yanı sıra hatanın yerinin belirlenmesi için de bir algoritma geliştirilmiştir. Bu sayede sistem çalışırken hem hatanın etkisi düzeltilir hem de hatanın yeri kullanıcının bilgisine sunulur.



**Şekil 5.1** : Destek vektörü makineleri ile DHTK yapısı.

### 5.2.1 Kontrolör katsayılarının belirlenmesi

Bu yöntemde, sistem çıkışına göre kontrolör katsayılarını belirleyecek olan *karar mekanizması* destek vektörleri makineleri tabanlıdır. Mekanizmanın eğitimi çevrim-dışı gerçekleştirilirken, sistem çıkışına uygun kontrolör parametresi ayarlama aşaması çevrim-içi çalışır. Şekil 5.1'deki karar mekanizmasına yakından bakalım. Karar mekanizması, her bir PID parametresi için bir tane olmak üzere toplam üç tane DVM-B içerir (Şekil 5.2). Üç DVM-B, sistem çıkışından gelen verileri giriş olarak alır ve uygun katsayıları çıkış olarak verir. Aşağıda DVM tabanlı mekanizmanın eğitim ve çalışma aşamaları verilmiştir;



Şekil 5.2 : Karar mekanizması.

*Eğitim (çevrim-dışı) :*

- Sistemin hatasız ve önceden belirlenmiş hatalı durumları için *en uygun* PID kontrolör katsayıları belirlenir. Bu amaçla *genetik algoritma* arama yöntemi ve başarımlı ölçütü olarak *ITSE* kullanılmıştır.
- PID kontrolör katsayılarından her birinin tayini için bir DVM kurulur. Kurulan her DVM, PID katsayıları ile ilgili olarak sırasıyla, DVM\_kp, DVM\_ki, DVM\_kd şeklinde adlandırılmıştır.
- Sistemin hatasız durumu için belirlenen PID katsayıları ile sistem kapalı çevrim çalışmasını sürdürürken, belirlenen hatalı durumların oluşması

durumunda sistem çıkışları çeşitli referans girişler için toplanır. Referans işaretlerinin çeşitliliği, yöntemin genelleştirme kapasitesini yükseltecektir.

- Kontrolör katsayılarını bulacak DVM lerin her biri, toplanan hatalı durum sistem cevapları, giriş ve ilgili kontrolör katsayısı, çıkış olacak şekilde farklı giriş işaretleri için çevrim-dışı eğitilir.

*İşleyiş (çevrim-İçi) :*

- Sistem istenen referans giriş için çalıştırılır. Belli aralıklarla sistem cevapları karar mekanizmasına gönderilir.
- DVM lerin hepsi gelen verileri aynı anda değerlendirir ve ürettikleri kontrolör katsayılarını çıkış olarak verir. Üç DVM den gelen üç katsayı ile oluşturulan kontrolör sisteme uygulanır.

### **5.2.2 DVM-B ile hata bulma ve tanıma**

Hatanın yerinin belirlenmesi için destek vektörü makinelerinin yüksek genelleştirme kapasitesine sahip bağlanım mekanizması benzer şekilde kullanılabilir. Hatanın yerinin belirlenmesi, bir hata bulma ve tanıma işlemidir. Destek vektörü makinelerinin bağlanım mekanizmasıyla farklı türdeki belli sayıda hataların ayırımını yapmak mümkündür. Bunun için her bir hata farklı bir değerle işaretlenir ve DVM-B'nin çıkışı olarak eğitime katılır. Hatalı durumlara ait sistem cevapları, DVM-B'nin girişi olarak alınır. Bir hatanın yeri, bir yerden uzaklığı, boyutu gibi ölçülebilen değerleri varsa bunları da DVM-B'nin kestirim yeteneklerini kullanarak bulmak mümkündür. Bu durumda, hatalı sistem çıkışları DVM-B girişi ve hatanın ölçülen değeri çıkışı olarak alınır ve eğitim gerçekleştirilir (Şekil 5.3). Burada sadece hatanın belli değerleri için kestirim yapılmaz, ara değerlerdeki hata parametresi de kestirilebilir. Hatanın konumunun belirlenmesi için düzenlenen algoritmanın detayları aşağıda verilmiştir.

#### **5.2.2.1 Hatanın konumunun belirlenmesi**

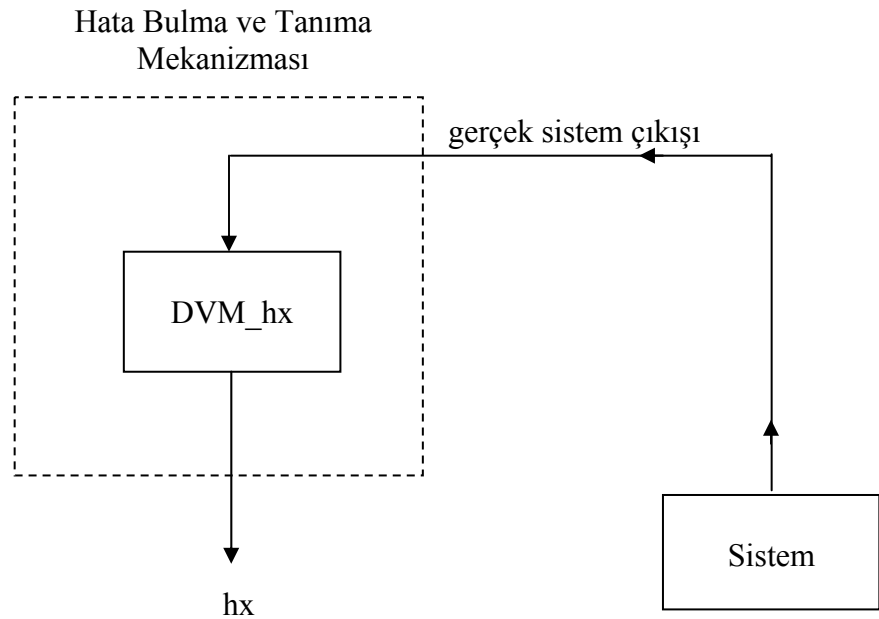
Algoritmanın eğitim aşaması çevrim-dışı iken işleyişi çevrim-İçi olacaktır. Eğitim ve işleyiş mekanizması aşağıda özetlenmiştir;

*Eđitim (evrim-dıřı) :*

- Hatanın yerinin tespiti iin bir DVM-B kurulur; DVM\_hx.
- Sistemin hatasız durumu iin belirlenen kontrolr aktif olduėu durumda iken, eđitim amacıyla seilmiř olan hatalı konumlara (seviyelere) iliřkin sistem ıkıřları eřitli referans giriřler iin toplanır.
- Hatanın yerini kestirecek DVM\_hx, toplanan hatalı durum sistem ıkıřları, giriř ve ilgili hata yeri ise, ıkıř olacak řekilde farklı referans giriř iřaretleri iin evrim-dıřı eđitilir.

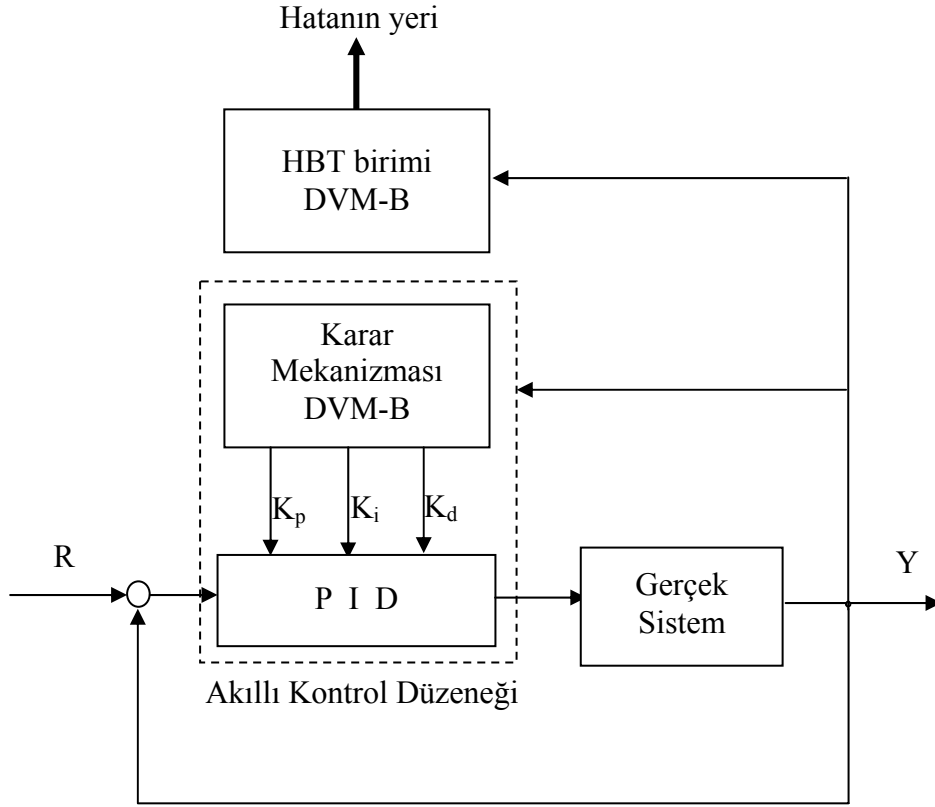
*İřleyiř (evrim-ii) :*

- Sistem istenen referans giriř iin alıřtırılır. Belli aralıklarla sistem cevapları karar mekanizmasına gnderilir.
- DVM\_hx gelen verileri deđerlendirir ve hatanın yerini kestirerek ıkıř olarak verir.



**řekil 5.3 :** nerilen DVM-B ile HBT sisteminin alıřma řeması.

Hatanın konumunun belirlenmesi iin geliřtirilen DVM-B ile hata bulma ve tanıma birimi PID katsayılarının kestirildiėi akıllı kontrol biriminden bađımsız alıřır. İstenirse geliřtirilen HTK sistemine řekil 5.4'deki gibi eklenebilir.



**Şekil 5.4 :** Önerilen DHTK ve HBT sistemi

Yöntemin çift tanklı sıvı seviye kontrol sisteminde farklı yükseklikteki ve farklı büyüklükteki delik hataları için uygulamaları yapılmıştır.

### 5.3 Benzetim Örneği

#### 5.3.1 Örnek 1

Önerilen DHTK yönteminin uygulaması, önceki bölümlerde de kullanılan çift tanklı sıvı seviye kontrol sisteminde yapılmıştır. Sistemde ikinci tanktaki sıvı seviyesi giriş akışı ile kontrol edilmektedir. Bu örnekte, birinci tankta bilinmeyen bir  $h_x$  yüksekliğinde  $r_x=7\text{mm}$ . yarıçaplı bir deliğin olduğu varsayılmaktadır (Şekil 5.5). Deliğin sisteme etkisi,

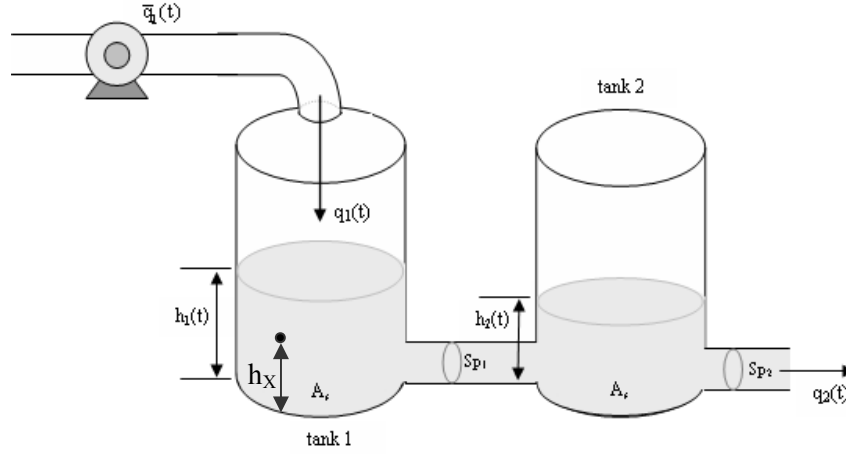
$$q_x(t) = a_1 \pi (r_x)^2 \sqrt{2gh_x(t)} \quad (5.1)$$

ile ifade edilir. Parametreleri Çizelge 3.1’de verilen sistemin denge denklemlerinde birinci tankta açılan deliğin etkisi -  $q_x(t)$  olacaktır.

$$\dot{h}_1(t) = \frac{1}{A_s} \left( -K_{p1} \text{sign}(h_1(t) - h_2(t)) \sqrt{2g|h_1(t) - h_2(t)|} + q_1(t) - q_x(t) \right)$$

$$\dot{h}_2(t) = \frac{1}{A_s} \left( K_{p1} \text{sign}(h_1(t) - h_2(t)) \sqrt{2g|h_1(t) - h_2(t)|} - K_{p2} \sqrt{2gh_2(t)} \right)$$
(5.2)

Amaç, deliğin yüksekliği ne olursa olsun, neden olduğu olumsuz etkiyi çevrim-içi olarak telafi etmektir.



Şekil 5.5 : Çift tanklı sıvı seviye kontrol sistemi.

Yüksekliği bilinmeyen deliğin sistem cevabındaki etkisini toparlayacak kontrolörün tasarımı için 5.2.1. bölümündeki aşamalar aşağıdaki gibi işletilir;

- PID kontrolör katsayılarından her birinin tayini için bir DVM kurulur; DVM\_kp, DVM\_ki, DVM\_kd şeklinde adlandırılır.
- $h_x = 0\text{cm}, 40\text{cm}, 80\text{cm}$ . için uygun kontrolörler genetik algoritma (GA) ile belirlenir (Çizelge 5.1). Burada aramalar, ITSE başarımlı ölçütüne göre yapılmıştır.
- Hatasız sistem için belirlenen PID katsayıları döngüye konarak söz konusu hatalı durumların çıkışları basamak, sinüs ve sözde rasgele seri giriş işaretleri için toplanır.
- Her üç DVM, toplanan hatalı durum sistem cevapları giriş, ilgili kontrolör katsayısı çıkış olacak şekilde eğitilir. Eğitimde DVM parametreleri,  $C=200$ ,  $\varepsilon=0.0001$  ve kernel fonksiyonu olarak polinom kernel ( $p = 1$ ) olarak seçilmiştir. DVM-B'ler eğitim tamamlandıktan sonra DHTK sisteminde yerleştirilir.

- Sistem istenen referans giriş için çalıştırılır. Belli aralıklarla sistem cevapları karar mekanizmasına gönderilir.
- DVM'lerin üçü de aynı anda gelen verileri değerlendirir ve ürettiği kontrolör katsayısını çıkış olarak verir. Üç DVM den gelen üç katsayı ile oluşturulan kontrolör sisteme uygulanır.

**Çizelge 5.1** : GA ile bulunan ve eğitimde kullanılacak PID katsayıları.

	PID(hatasız)	PID(0cm)	PID(40cm)	PID(80cm)
$K_p$	26.58776	24.8138	26.83225	28.92151
$K_i$	0.0805	0.90161	1.49756	1.8599
$K_d$	6.15532	8.14017	0.25	2.16411

Bağımsız bir DVM-B ile deliğin yüksekliği de kestirilebilir. Bunun için 5.2.2.1 bölümündeki aşamalar gerçekleştirilir. Toplanan hatalı durum çıkışları DVM nin girişi ve deliklerin yükseklikleri de çıkışı olarak eğitim yapılır. Gerçek sistem çıkışı ile test yapılır. DVM-B'nin eğitimi ve işleyişinin ayrıntıları aşağıdaki gibidir:

- Bilinmeyen yüksekliğin tayini için bir DVM kurulur; DVM\_hx.
- $h_x = 0\text{cm}, 40\text{cm}, 80\text{cm}$ . de delik olan hatalı sistemler eğitimde kullanılmak üzere hazırlanır.
- Hatasız sistem için belirlenen PID katsayıları döngüye konarak söz konusu hatalı durumların çıkışları basamak, sinüs ve sözde rasgele seri giriş işaretleri için toplanır.
- DVM\_hx, toplanan hatalı durum sistem cevapları giriş, ilgili  $r_x$  değeri çıkış olacak şekilde eğitilir. Eğitimde DVM parametreleri,  $C=200$ ,  $\varepsilon = 0.0001$  ve polinom kernel ( $p= 1$ ) olarak seçilmiştir. DVM\_hx eğitim tamamlandıktan sonra HBT sisteminde yerleştirilir.
- Sistem istenen referans giriş için çalıştırılır. Belli aralıklarla sistem cevapları sınıanır.
- DVM\_hx, aynı anda gelen verileri değerlendirir ve hatanın yerini kestirerek çıkış olarak verir.

**Çizelge 5.2 :** Eğitim dışı farklı yükseklikteki deliklere DHTK yönteminin atadığı PID katsayıları.

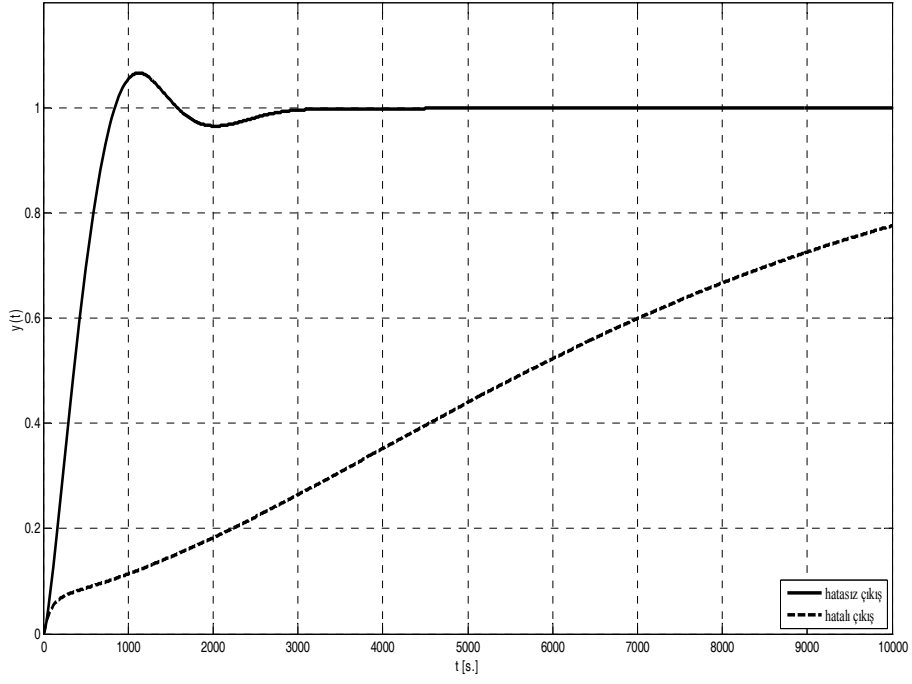
Gerçek yükseklik(cm)	Kestirilen yükseklik(cm)	$K_p$	$K_i$	$K_d$
10	9.9972	25.3375	1.0732	5.2897
30	29.9924	26.3402	1.3711	1.3311
50	50.0141	27.3271	1.6092	0.2213

Çizelge 5.2’de birinci tankta sırasıyla 10cm., 30cm., 50cm., yüksekliğinde açılan deliklere sistemin çevrim-içi atadığı kontrolör katsayıları verilmiştir. Sistem, açılan deliğin yüksekliğini yani hatanın yerini de kestirecek şekilde tasarlanmıştır. Çizelge 5.3, aynı yükseklikteki delikler için genetik algoritma ile karşılaştırma amacıyla bulunan katsayıları göstermektedir.

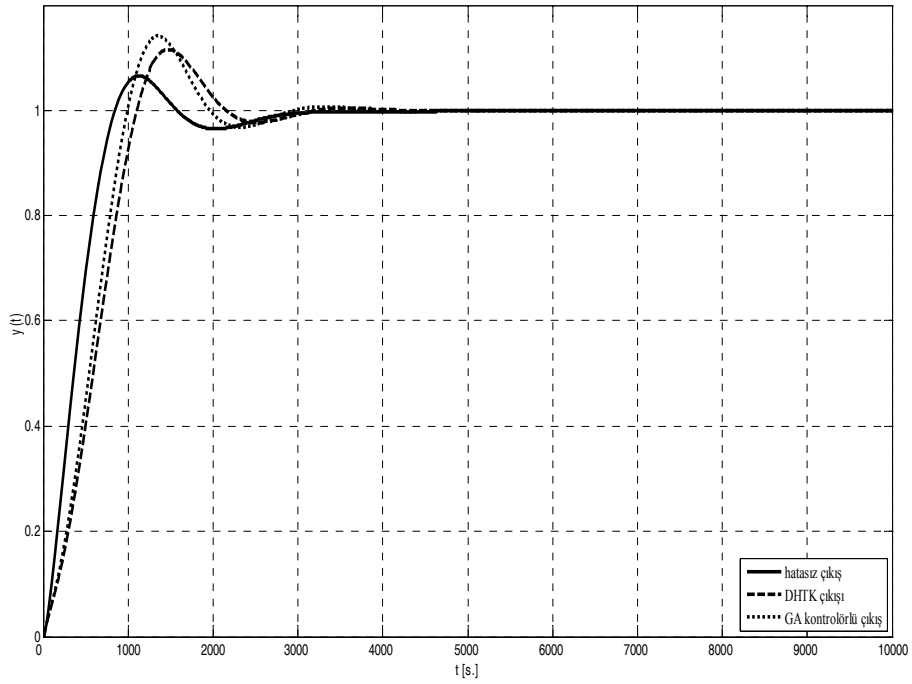
**Çizelge 5.3 :** Eğitim dışı farklı yükseklikteki deliklere ilişkin GA ile bulunan PID katsayıları.

Gerçek yükseklik(cm)	$K_p$	$K_i$	$K_d$
10	26.3375	1.2044	2.2327
30	28.9587	1.2810	0.4694
50	29.7307	1.5142	0.4103

Hatalı ve düzeltilmiş durumlara ait benzetim sonuçları ve genetik algoritma sonuçlarıyla karşılaştırmalar aşağıda verildiği gibidir.

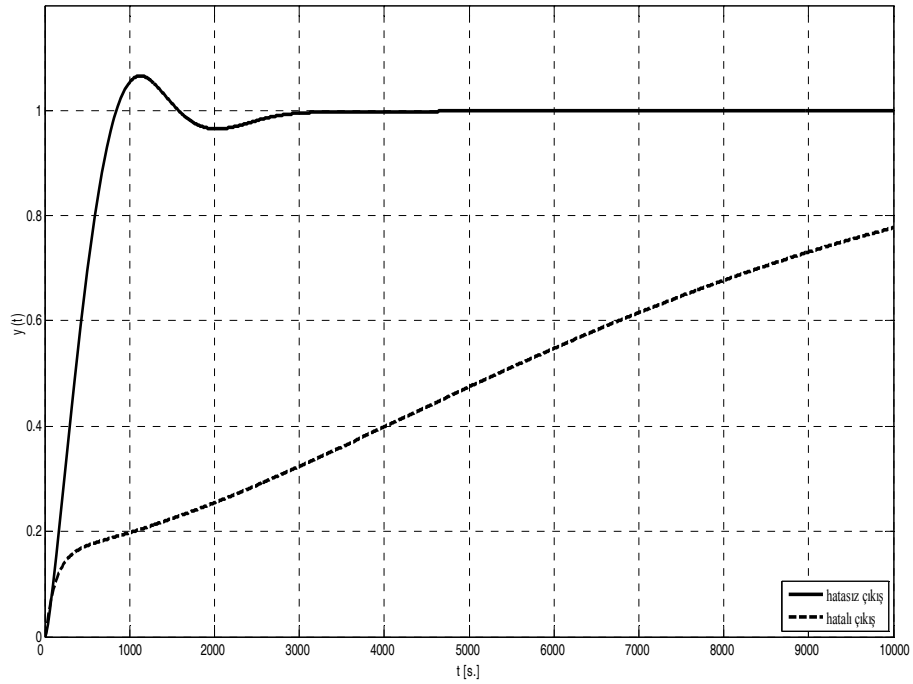


Şekil 5.6 : Delik yüksekliği 10cm. için hatasız ve hatalı sistem çıktıları.

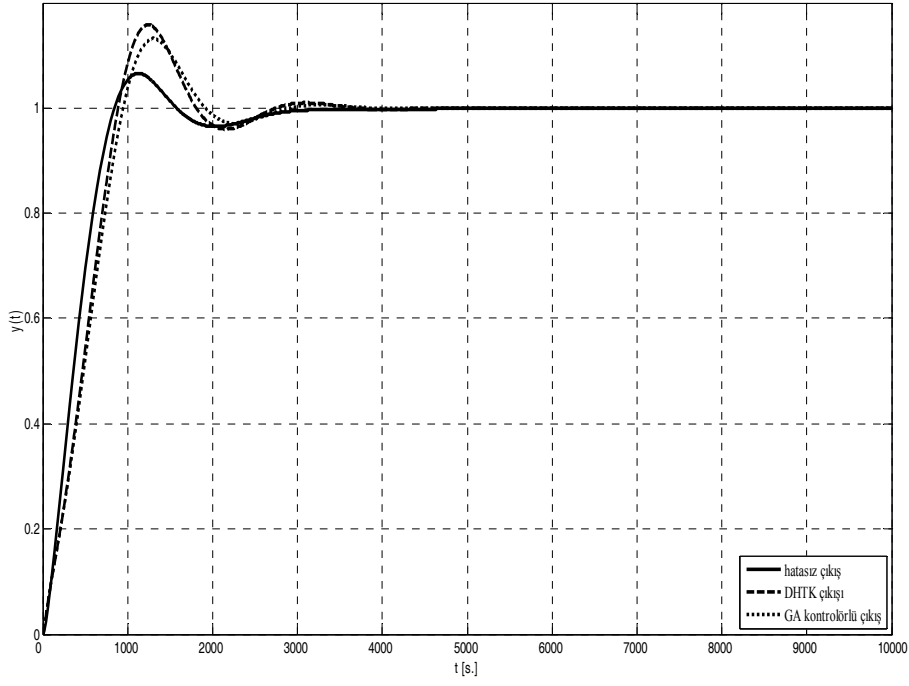


Şekil 5.7 :  $h_x=10$ cm. için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıktıları.

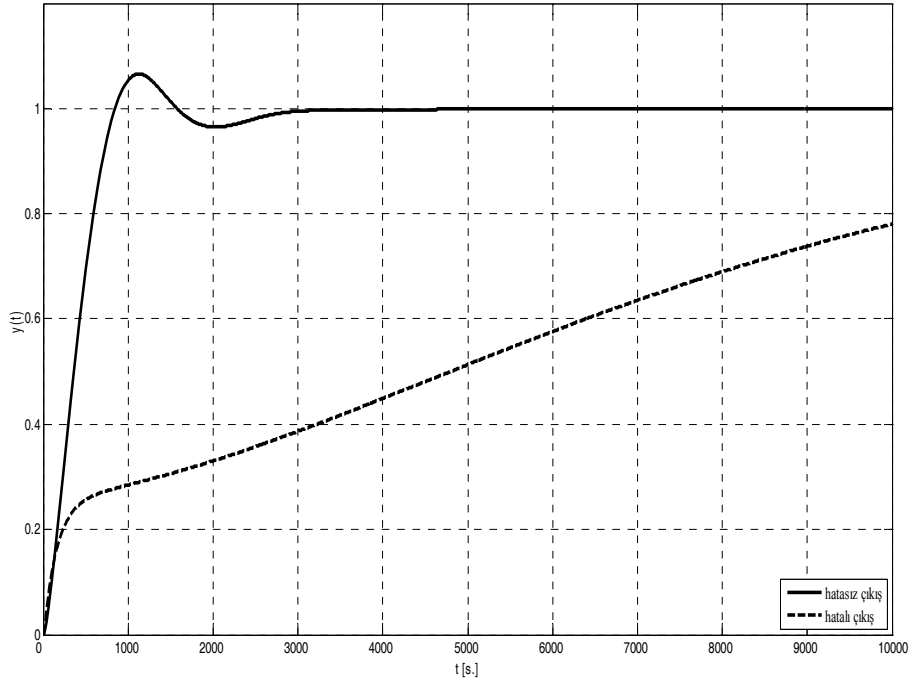
Şekil 5.6'de 10 cm. yüksekliğinde oluşan deliğin sistemde yarattığı sapma görülmektedir. Kapalı çevrim, sistem cevabını toparlayamamaktadır. DHTK sisteminde ise sistem cevabı kontrolör parametrelerinin yeni duruma uygun şekilde yenilenmesiyle toparlanır ve sistemi istenen davranışta tutar. Şekil 5.7, delik yüksekliğinin 10 cm. olduğu duruma ait DHTK ile düzeltilmiş sistem çıkışı göstermektedir. Karşılaştırma amacıyla aynı delik için genetik algoritma ile kontrolör katsayıları çevrim-dışı olarak aranmıştır. Görüldüğü gibi yöntemin çevrim-içi uyguladığı kontrolör ile GA aracıyla çevrim-dışı bulunan kontrolörün başarımı neredeyse aynıdır. Aynı inceleme, delik yüksekliğinin 30 cm. ve 50 cm. olduğu durumlar için de yapılmıştır. Sonuçlar, Şekil 5.8, Şekil 5.9, Şekil 5.10 ve Şekil 5.11'de verildiği gibidir.



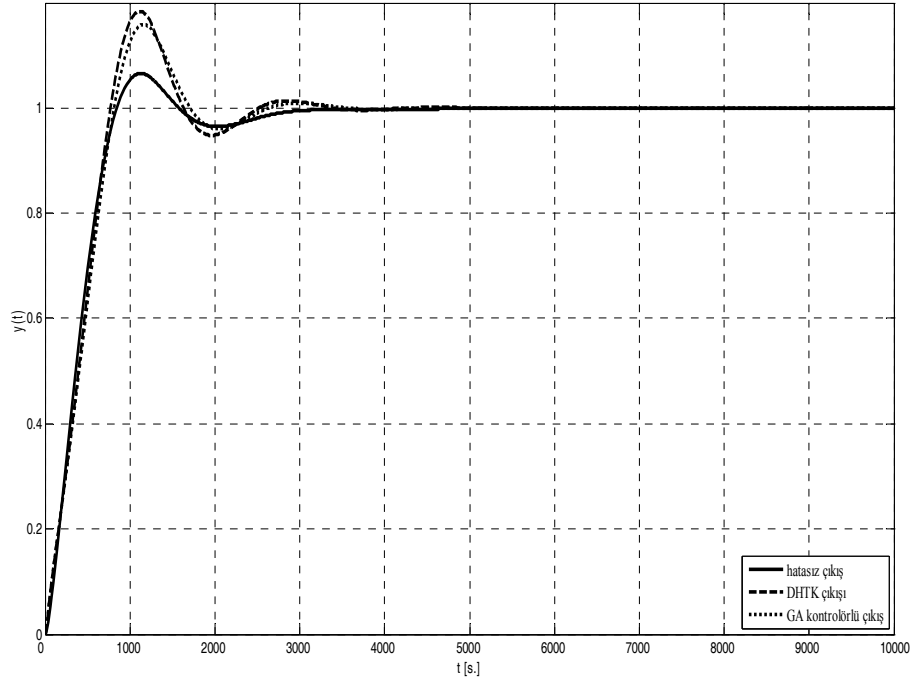
Şekil 5.8 : Delik yüksekliği 30cm. için hatasız ve hatalı sistem çıkışları.



Şekil 5.9 :  $h_x=30$ cm. için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.



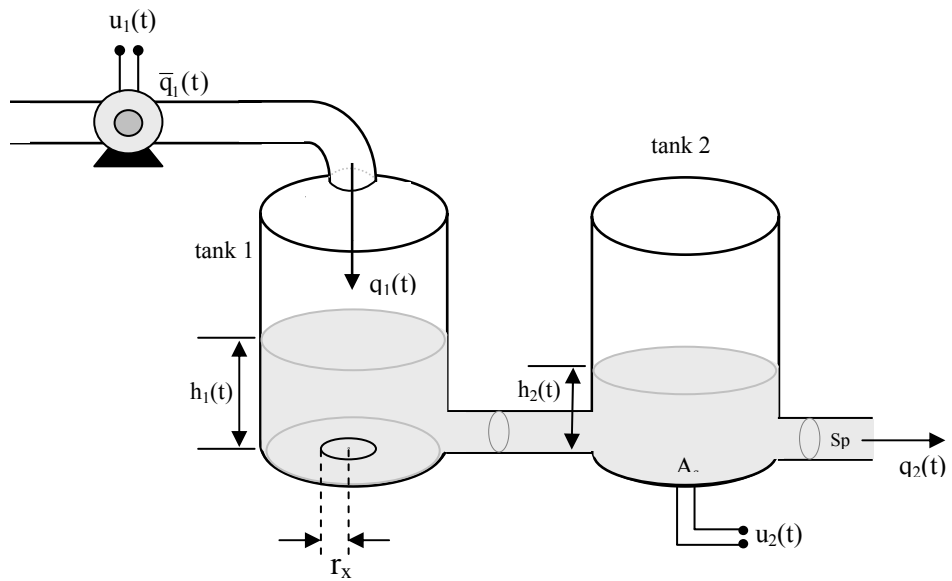
Şekil 5.10 : Delik yüksekliği 50cm. için hatasız ve hatalı sistem çıkışları.



Şekil 5.11 :  $h_x=50\text{cm}$ . için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.

### 5.3.2 Örnek 2

Önerilen DHTK yönteminin ikinci uygulaması yine çift tanklı sıvı seviye kontrol sisteminde başka bir hata türü için yapılmıştır. Örnek için, birinci tankın tabanında bilinmeyen bir  $r_x$  yarıçaplı bir deliğin olduğu varsayılmaktadır (Şekil 5.12).



Şekil 5.12 : Birinci tankında  $r_x$  yarıçaplı delik olan çift tanklı sistem.

Yarıçapı bilinmeyen delik hataları için DHTK yöntemi aşağıdaki gibi işletilir:

- PID kontrolör katsayılarından her birinin tayini için bir DVM kurulur; Kurulan her DVM, PID katsayıları ile ilgili olarak sırasıyla, DVM<sub>kp</sub>, DVM<sub>ki</sub>, DVM<sub>kd</sub> şeklinde adlandırılmıştır.
- $r_x = 5\text{mm}, 10\text{mm}, 18\text{mm}$ . için uygun kontrolörler genetik algoritma (GA) ile belirlenir (Çizelge 5.4). Burada aramalar, ITSE başarımlı ölçütüne göre yapılmıştır.
- Hatasız sistem için belirlenen PID katsayıları döngüye konarak söz konusu hatalı durumların çıkışları basamak, sinüs ve sözde rasgele seri giriş işaretleri için toplanır.
- Her üç DVM, toplanan hatalı durum sistem cevapları giriş, ilgili kontrolör katsayısı çıkış olacak şekilde eğitilir. Eğitimde DVM parametreleri,  $C=200$ ,  $\varepsilon = 0.0001$  ve radyal tabanlı kernel ( $\sigma = 1$ ) olarak seçilmiştir. DVM-B'ler eğitim tamamlandıktan sonra DHTK sisteminde yerleştirilir.
- Sistem istenen referans giriş için çalıştırılır. Belli aralıklarla sistem cevapları karar mekanizmasına gönderilir.
- DVM'lerin üçü de aynı anda gelen verileri değerlendirir ve ürettiği kontrolör katsayısını çıkış olarak verir. Üç DVM den gelen üç katsayı ile oluşturulan kontrolör sisteme uygulanır.

**Çizelge 5.4 :** GA ile bulunan ve eğitimde kullanılacak PID katsayıları.

	PID(hatasız)	PID(5mm)	PID(10mm)	PID(18mm)
$K_p$	29.67529	29.9194	29.9006	27.6250
$K_i$	0.09439	0.2319	0.8589	3.1832
$K_d$	9.06868	0.0789	0.0344	0.0002

Bağımsız bir DVM-B ile deliğin yarıçapı da kestirilebilir. Bunun için, toplanan hatalı durum çıkışları, DVM nin girişi ve deliklerin yarıçapları da, çıkışı olarak eğitim yapılır. Gerçek sistem çıkışı ile test yapılır. DVM-B'nin eğitimi ve işleyişinin ayrıntıları aşağıdaki gibidir:

- Bilinmeyen yüksekliğin tayini için bir DVM kurulur; DVM\_hx.
- $r_x = 5\text{mm}, 10\text{mm}, 18\text{mm}.$  de delik olan hatalı sistemler çıkışları hatasız sistem için belirlenen PID katsayıları döngüye konarak çeşitli giriş işaretleri için toplanır.
- DVM\_rx, toplanan hatalı durum sistem cevapları giriş, ilgili  $r_x$  değeri çıkış olacak şekilde eğitilir. Eğitim tamamlandıktan sonra HBT sisteminde yerleştirilir.
- Sistem istenen referans giriş için çalıştırılır. Belli aralıklarla sistem cevapları sınıanır.
- DVM\_rx, aynı anda gelen verileri değerlendirir ve hatanın boyutunu kestirerek çıkış olarak verir.

**Çizelge 5.5 :** Eğitim dışı farklı yarıçaplardaki deliklere yöntemin atadığı PID katsayıları.

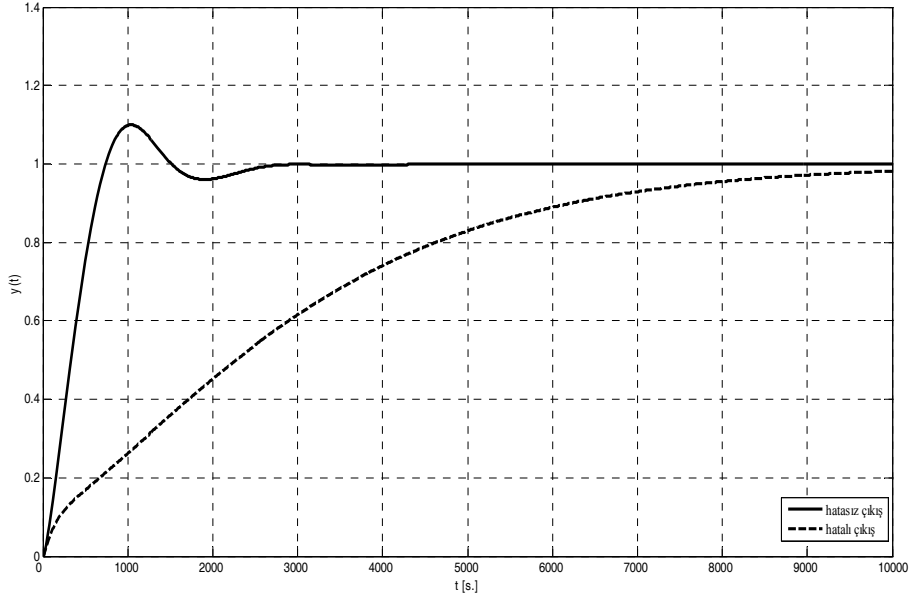
Gerçek yarıçap(mm)	Kestirilen yarıçap(mm)	$K_p$	$K_i$	$K_d$
8	8.2361	29.3652	0.5564	0.0537
15	15.5917	28.8118	2.3510	0.0054
20	19.1924	26.8720	3.4141	0.0013

Çizelge 5.5’de birinci tankta sırasıyla 8mm., 15mm., 20mm., yarıçapında açılan deliklere sistemin çevrim-içi atadığı kontrolör katsayıları verilmiştir. Sistem, açılan deliğin yarıçapını yani hatanın büyüklüğünü de kestirecek şekilde tasarlanmıştır. Çizelge 5.6, aynı yarıçaptaki delikler için genetik algoritma ile karşılaştırma amacıyla bulunan katsayıları göstermektedir.

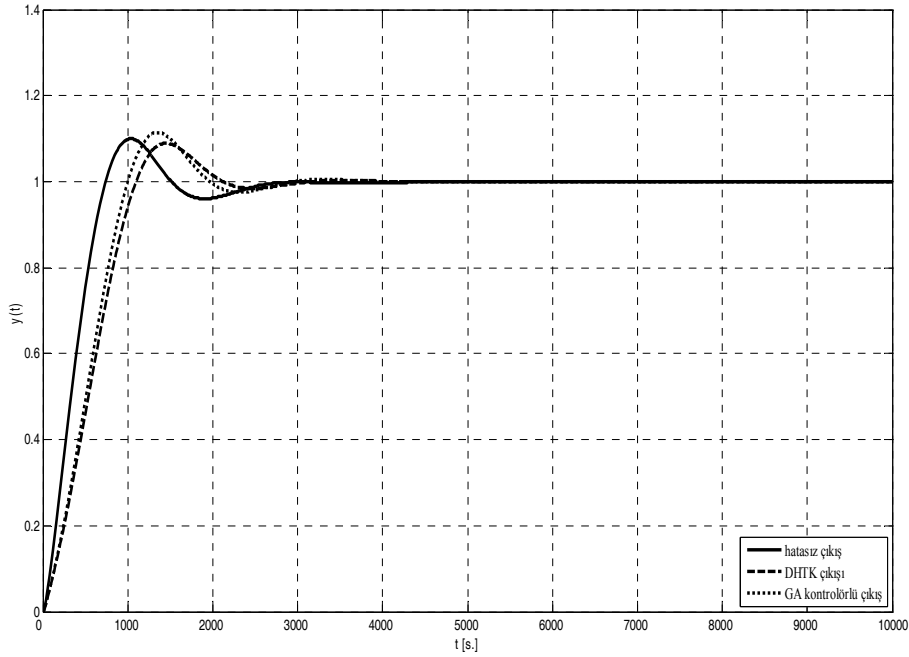
**Çizelge 5.6 :** Eğitim dışı farklı yarıçaplardaki deliklere ilişkin GA ile bulunan PID katsayıları.

Delik yarıçapı(mm)	$K_p$	$K_i$	$K_d$
8	29.50	0.61	0.0611
15	29.8638	2.0767	0.3816
20	29.9136	3.7599	0.9538

Hatalı ve düzeltilmiş durumlara ait benzetim sonuçları ve genetik algoritma sonuçlarıyla karşılaştırmalar aşağıda verildiği gibidir. Şekil 5.13, test için kullanılan 8mm. yarıçaplı delik için hatalı ve hatsız çıkışları göstermektedir. Şekil 5.14’de, DHTK yöntemi ve genetik algoritmayla bulunan sonuçlar karşılaştırılmıştır.

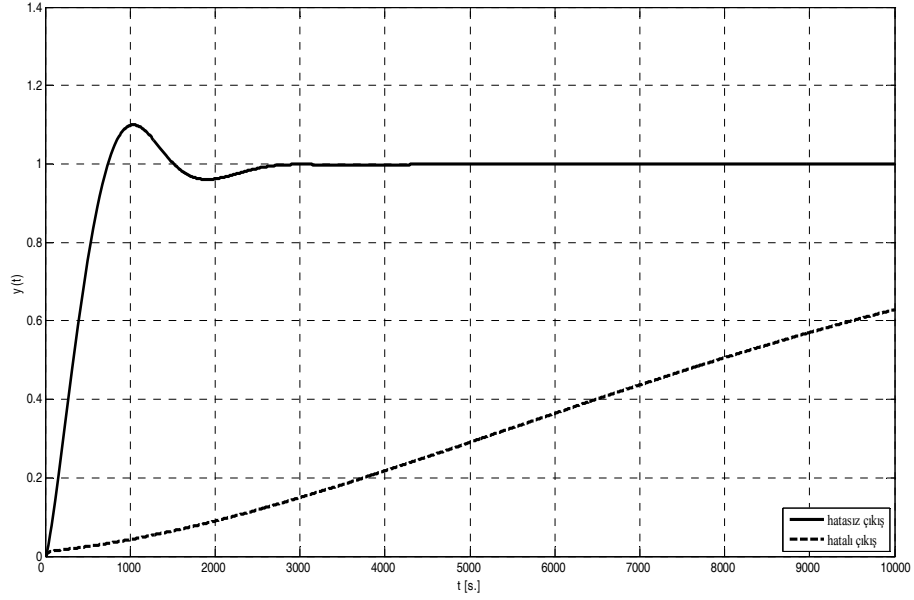


Şekil 5.13 : Delik yarıçapı 8mm. için hatalı ve hatsız sistem çıkışları.

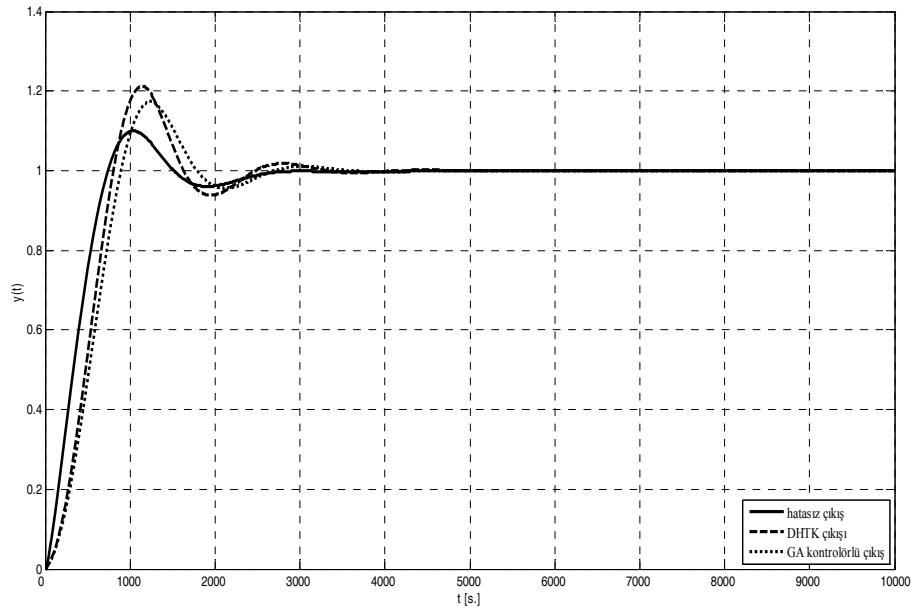


Şekil 5.14 :  $r_x=8\text{mm.}$  için hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.

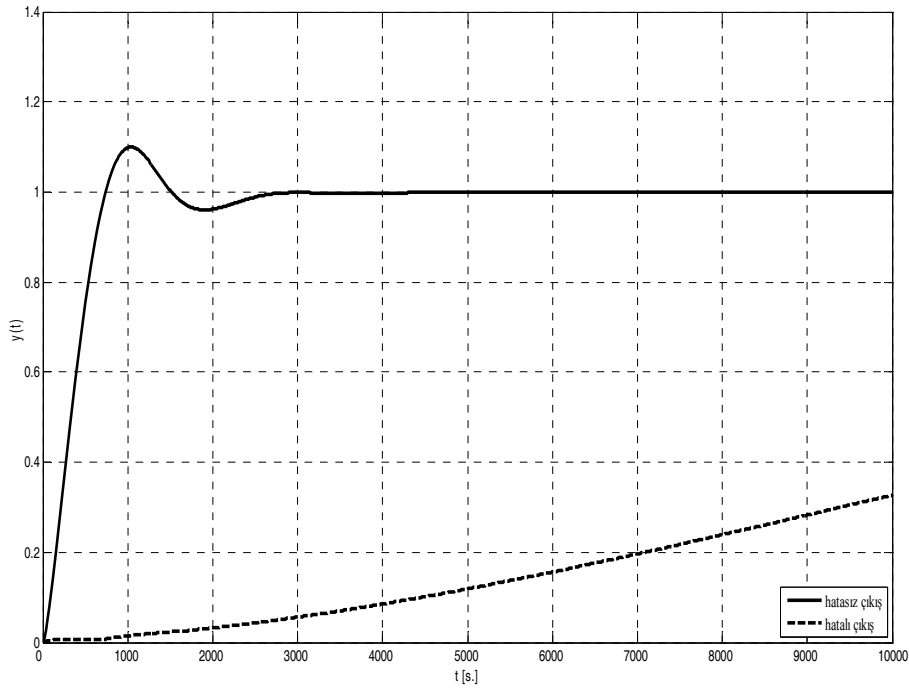
Şekil 5.15, test amaçlı kullanılan 15mm. yarıçaplı delik için hatalı ve hatasız çıkışları göstermektedir. Şekil 5.16’de, DHTK yöntemi ve genetik algoritmayla bulunan sonuçlar karşılaştırılmıştır. Benzer inceleme 20mm. yarıçaplı delik için de yapılmıştır. Şekil 5.17 ve Şekil 5.18’de verilmiştir.



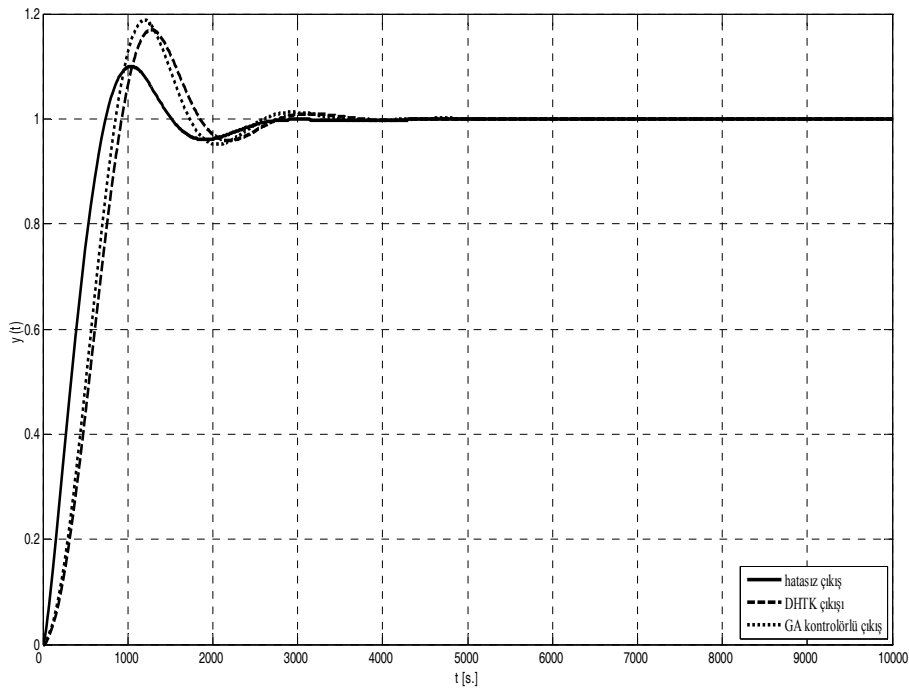
Şekil 5.15 : Delik yarıçapı 15mm. için hatasız ve hatalı sistem çıkışları.



Şekil 5.16 :  $r_x=15\text{mm.}$  için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.



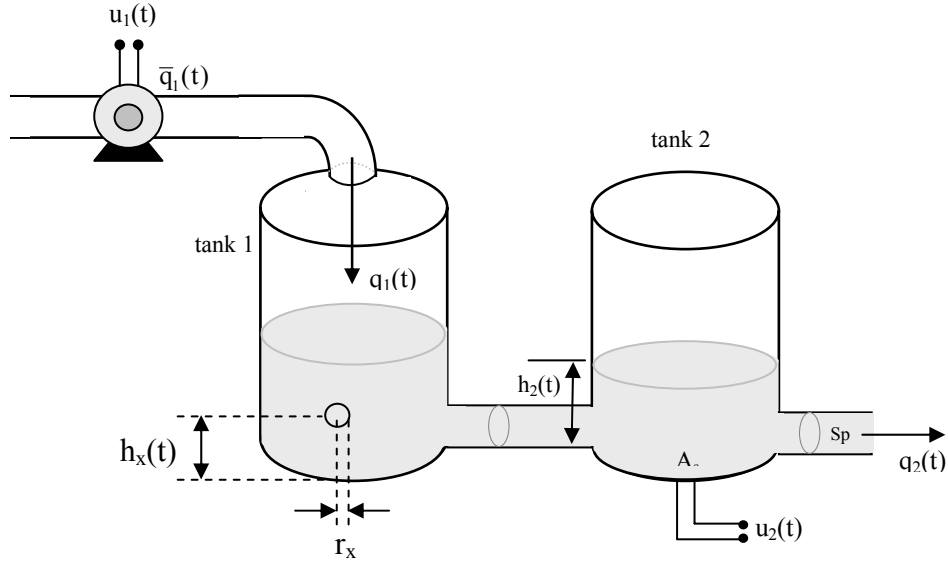
Şekil 5.17 : Delik yarıçapı 20mm. için hatasız ve hatalı sistem çıkışları.



Şekil 5.18 :  $r_x=20$ mm. için hatasız, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.

### 5.3.3 Örnek 3

Önerilen DHTK yönteminin üçüncü uygulaması yine çift tanklı sıvı seviye kontrol sisteminde başka bir hata türü için yapılmıştır. Örnek için, birinci tankta  $h_x$  yüksekliğinde  $r_x$  yarıçaplı bir deliğin olduğu varsayılmaktadır (Şekil 5.19).



Şekil 5.19 : Birinci tankında  $h_x$  yüksekliğinde  $r_x$  yarıçaplı bir delik olan çift tanklı sistem.

Burada, deliğin iki değişken parametresi vardır. Amaç, oluşan bu deliğin etkisini telafi edecek kontrolör katsayılarının çevrim-içi hesaplanmasıdır. Bu amaç doğrultusunda önceki iki örnek için genetik algoritma ile çevrim-dışı elde edilen hatalı ve hatasız durum kontrolör parametreleri ve toplanan sistem çıkışları kullanılmıştır. İki farklı özelliği değişken olan hatanın telafisi için aşağıdaki yol izlenmiştir.

- PID kontrolör katsayılarından her birinin tayini için bir DVM kurulur; DVM\_kp, DVM\_ki, DVM\_kd.
- $h_x = 0\text{cm}, 40\text{cm}, 80\text{cm}$ . ve  $r_x = 5\text{mm}, 10\text{mm}, 18\text{mm}$  için genetik algoritma ile hesaplanan kontrolör katsayıları DVM'in çıkışı olarak atanır. Söz konusu durumlara ilişkin hatalı sistem çıkışları DVM'in girişi olarak atanır. Her üç DVM de altı girişli ve bir çıkışlı olarak eğitilmiş olur.
- Eğitimde DVM parametreleri,  $C=200$ ,  $\varepsilon=0.0001$  ve kernel fonksiyonu olarak polinom kernel ( $p = 1$ ) olarak seçilmiştir. DVM-B'ler eğitim tamamlandıktan sonra DHTK sisteminde yerleştirilir.

- Sistem istenen referans giriş için çalıştırılır. Belli aralıklarla sistem cevapları karar mekanizmasına gönderilir.
- DVM'in üçü de aynı anda gelen verileri değerlendirir ve ürettiği kontrolör katsayısını çıkış olarak verir. Üç DVM den gelen üç katsayı ile oluşturulan kontrolör sistemde oluşan sapmanın giderilmesi için devreye girer.

Yöntemin hata telafisindeki başarımını görmek için aşağıdaki özellikteki hataların oluştuğu varsayılmıştır:

- Yarıçapı  $r_x = 8\text{mm}$  olan  $h_x = 20\text{cm}$  yüksekliğinde bir delik.
- Yarıçapı  $r_x = 15\text{mm}$  olan  $h_x = 5\text{cm}$  yüksekliğinde bir delik.
- Yarıçapı  $r_x = 7\text{mm}$  olan  $h_x = 30\text{cm}$  yüksekliğinde bir delik.
- Yarıçapı  $r_x = 12\text{mm}$  olan  $h_x = 10\text{cm}$  yüksekliğinde bir delik.

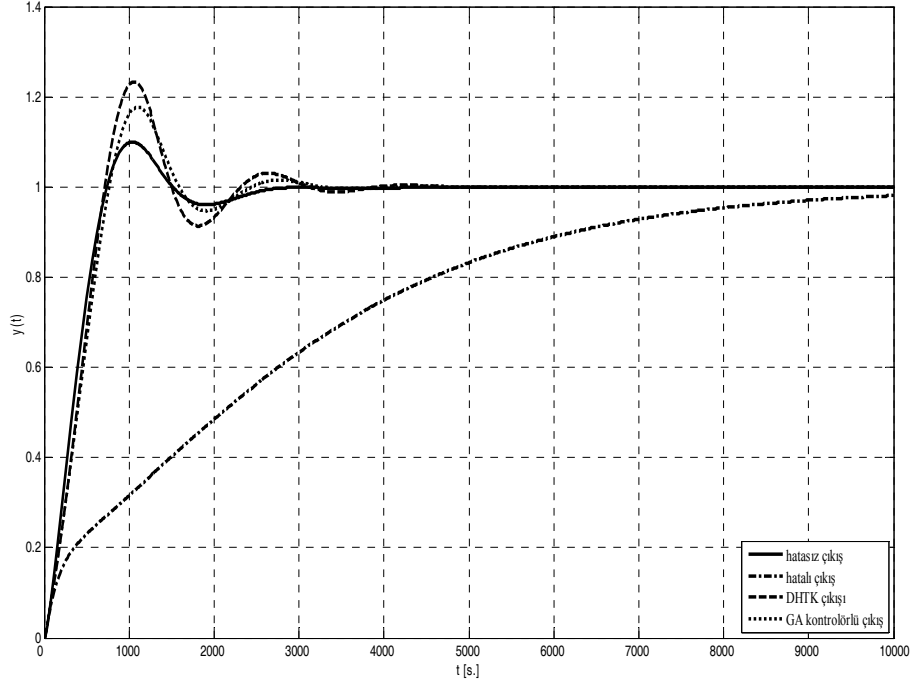
Bu hatalara doğrudan hata toleranslı kontrol yönteminin çevrim-içi ürettiği PID katsayıları ve genetik arama algoritmasının çevrim-dışı ürettiği PID katsayıları Çizelge 5.7'de ve Çizelge 5.8'de verilmiştir. Görsel karşılaştırma için sistem çıkışları, Şekil 5.20, 21, 22, 23'de verilmiştir.

**Çizelge 5.7** : Dört farklı hata durumu için DHTK yönteminin atadığı PID katsayıları.

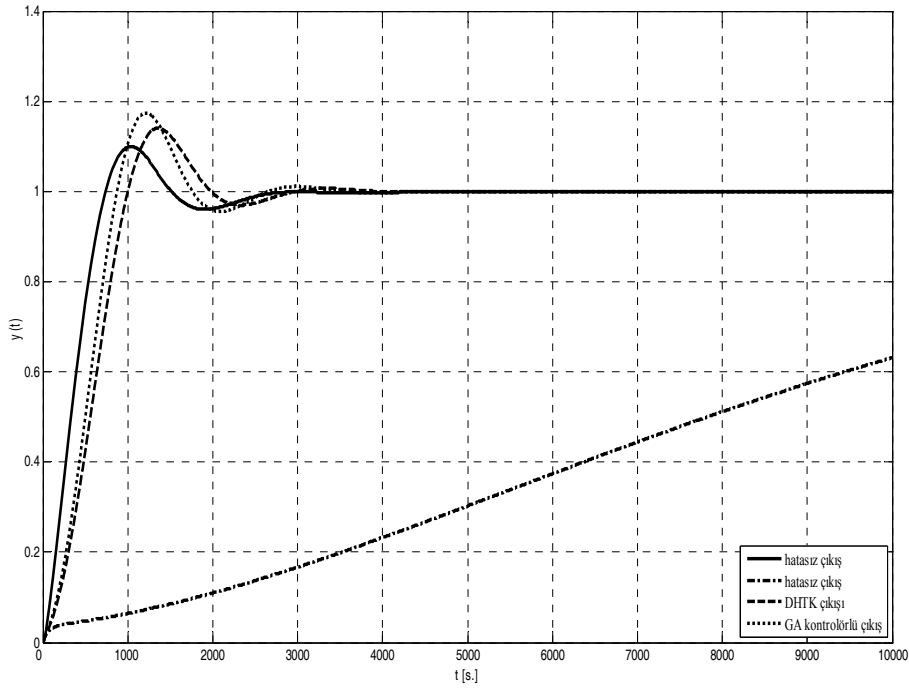
Delik yarıçapı-yüksekliği	$K_p$	$K_i$	$K_d$
8mm-20cm	30.3495	0.9040	2.3814
15mm-5cm	28.985	1.831	3.501
7mm-30cm	29.6539	0.6111	5.8891
12mm-10cm	25.3705	1.2767	4.4959

**Çizelge 5.8 :** Dört farklı hata durumu için genetik algoritmanın atadığı PID Katsayıları.

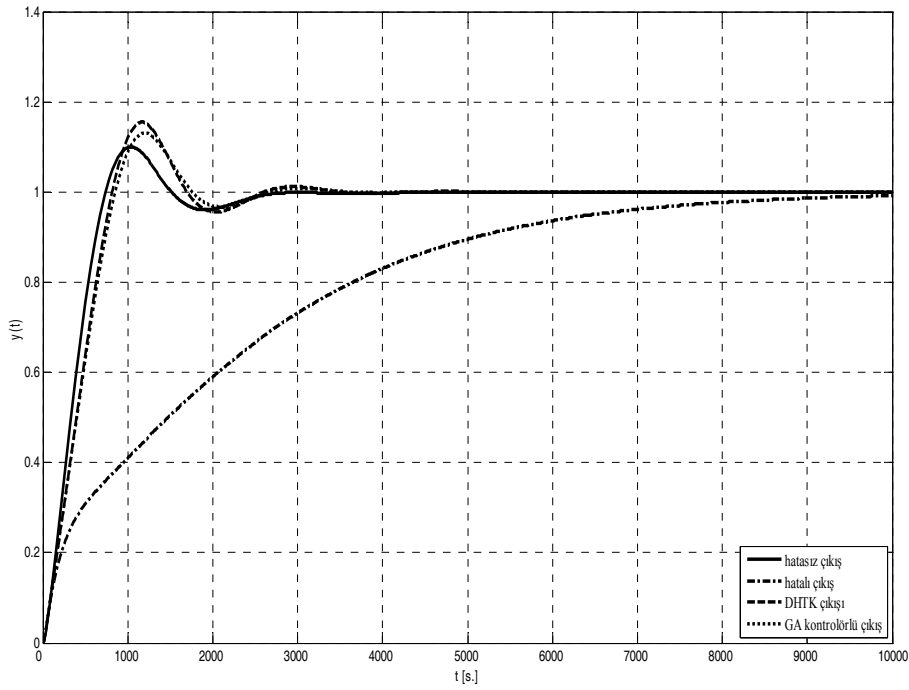
Delik yarıçapı-yüksekliği	$K_p$	$K_i$	$K_d$
8mm-20cm	33.3495	0.8240	6.8186
15mm-5cm	30.9258	2.137	4.7178
7mm-30cm	30.5421	0.5816	7.4272
12mm-10cm	27.4901	1.4012	3.5236



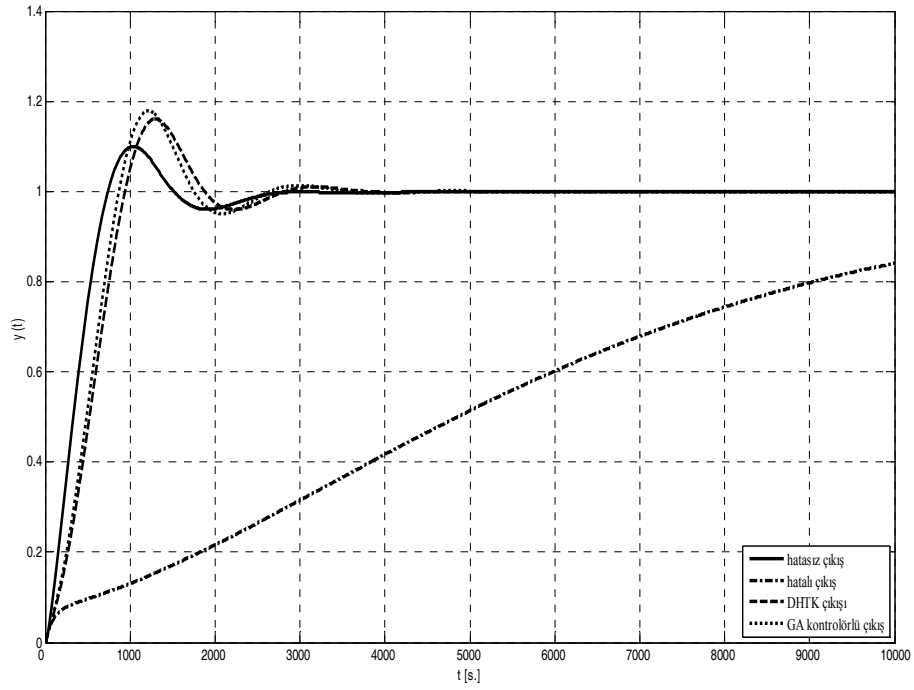
**Şekil 5.20 :**  $r_x = 8\text{mm}$  ve  $h_x = 20\text{cm}$  için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörle düzeltilmiş sistem çıkışları.



Şekil 5.21 :  $r_x = 15\text{mm}$  ve  $h_x = 5\text{cm}$  için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörlü düzeltilmiş sistem çıkışları.



Şekil 5.22 :  $r_x = 7\text{mm}$  ve  $h_x = 30\text{cm}$  için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörlü düzeltilmiş sistem çıkışları.



**Şekil 5.23 :**  $r_x = 12\text{mm}$  ve  $h_x = 10\text{cm}$  için hatasız, hatalı, DHTK ile düzeltilmiş, GA ile elde edilen kontrolörlü düzeltilmiş sistem çıkışları.

#### 5.4 Sonuçların Yorumlanması

Bu çalışmada önerilen DVM-B ile Doğrudan Hata Toleranslı Kontrol (DHTK) yönteminde kontrolör katsayılarının çevrim-içi ayarlanması destek vektörü makineleri tabanlı akıllı bir sistemle yapılmaktadır. Kapalı çevrimli sistemin baş edemediği hatalarda da yöntemin başarıyla uygulandığı görülmüştür. Bu yöntem ile hatanın etkilerini en aza indirme işlemi, hata bulma-tanıma işlemlerinden bağımsız olarak gerçekleştirir. Yöntem, sistemi nominal davranışına geri getirecek düzenlemeyi yapmak için öncelikle hatanın yerinin veya türünün belirlenmesini gerektirmez. Öte yandan, bu yöntemin yan ürün olarak, hatanın yerinin veya türünün de çok büyük bir yakınlıkla tespit edildiği görülmüştür. Hatanın ve yerinin tespiti, sistemlerdeki arızanın giderilmesinde çok önemli bir unsurdur.

Hata tespiti ve tanınmasından bağımsız olarak çalışan DVM-B ile DHTK sistemi, sadece sistem çıkışını gözleyerek uygun PID kontrolör katsayılarını çevrim-içi olarak üretir. Yöntemin bulduğu çevrim-içi kontrolör katsayılarının, genetik arama algoritmasıyla çevrim-dışı elde edilen katsayılarla, neredeyse çakışması, optimal sonuca çok yaklaşıldığını göstermektedir. İki farklı hata türü için toplanan verilerle

eđitilen yeniden yapılandırma mekanizmasının bu hata özelliklerinin aynı anda oluşması durumunda da son derece iyi sonuçlar verdiği görülmüştür. Tüm bu algoritmanın temeli olan destek vektörü makinelerinin çok küçük bir veri kümesi ile eğitildiđi halde oldukça geniş bir bölgeyi doğru taraması, karmaşık işlemlere gerek duymaması ve hızlı olması yöntemin diđer olumlu tarafları olarak söylenebilir. Ayrıca, çevrim-içi kontrolör seçen yöntemlerde karşılaşılabilecek olan sıçramalı geçiş sorununa önlem alınmasının gerekmemesi ve önceden belirlenmiş hatalarla sınırlı olmaması da yöntemin diđer olumlu yönleri olarak belirtilebilir.

## 6. SONUÇLAR

Hata toleranslı kontrol sistemleri, geleneksel geri-beslemeli kontrol sistemlerinin yetersiz kaldığı bazı arıza durumlarındaki kararsızlık ve başarımların düşüklüğü gibi zayıflıkların üstesinden gelmek üzere geliştirilmiştir. Geçmişte bu konular üzerinde yapılmış pek çok çalışma bulunmaktadır. Bu konudaki çalışmalar, son yıllarda giderek yaygınlaşan yapay sinir ağları, bulanık mantık ve genetik arama gibi pek çok akıllı algoritmanın kullanımlarıyla daha da hız kazanmıştır. Hata bulma ve tanıma sistemleri, hataların varlığını tespit eden ve türü, yeri, boyutu ve/veya zamanı gibi özelliklerini belirleyen birimlerdir. Hata bulma ve tanıma problemi, hata toleranslı kontrolün bir konusu olmasına rağmen, bu konuda literatürde değişik başlıklar altında da çalışmalar bulunmaktadır. Bu tezde geliştirilen yöntemlerde, bir makine öğrenimi yöntemi olan destek vektörü makinelerinin sınıflandırma ve bağlantım özellikleri kullanılmıştır. Son yıllarda destek vektörü makineleri, üstün genelleştirme kapasiteleri, sınıflandırma, bağlantım ve modellemedeki başarımları ile yapay sinir ağlarına çok iyi bir alternatif olmuşlardır.

Tezdeki ilk yöntem bir hata bulma ve tanıma yöntemidir. Hata bulma işleminde, destek vektörü bağlantımı, hata tanıma işleminde destek vektörü sınıflandırması kullanılmıştır. Hata bulma işleminde, hatasız durum giriş-çıkış verileriyle oluşturulan normal (hatasız) bölge, sistemin normal çalışma koşullarının gösterildiği bölgedir. Güvenli bölgenin üst ve alt sınırları iki ayrı destek vektörü bağlantım makinesi ile modellenmiştir. Bir çıkış işareti, oluşturulan güvenli bölgenin alt veya üst sınırını aşarsa bir hata tespiti yapılır. Destek vektörü ile çoklu sınıflandırma yöntemlerinden biri olan bire-karşı-diğerleri, teknik sistemde tespit edilen hatayı beklenen ve önceden belirlenen hatalardan biri olarak sınıflandırır. Yöntemin hızlı çalışması, az veriyle yüksek başarımlı sınıflandırma ve kestirimlerin yapılabilmesi, hata tespiti için geçen sürenin çok kısa olması ve karmaşık algoritmalara, formüllere ve sistem modeline ihtiyaç duymaması en önemli avantajlarıdır. Kestirimde DVM'nin hafızasız özel bir türünün kullanılması da, hafıza, yer ve zaman tasarrufu açısından önemlidir.

İkinci yöntem, yeniden yapılandırma mekanizması ‘çevrim-içi kontrolör seçen’ türde olan bir aktif hata toleranslı kontrol yöntemidir. Yöntem, hatayı destek vektörü makinelerinin bir çoklu sınıflandırma yöntemi ile, beklenen ve önceden belirlenen hatalardan biri olarak sınıflar. Bir hata tespit edildiğinde sistemin kapalı çevrim başarımının devamını sağlamak için uygun olan kontrolör çevrim-içi seçilir. Bu yöntemde PID kontrolörleri kullanılmıştır ve kontrolör parametreleri genetik algoritmayla çevrim-dışı belirlenmiştir. Yöntem, özellikle olası hataların bilindiği HTK problemlerinin çözümü için güçlü bir araç sunmaktadır. HBT biriminin eğitimleri, farklı hataların birbirini izlemesi ihtimalleri göz önüne alınıp yapılarak yöntemin güvenilirliği artırılmıştır. Yöntemin başarımı, çift tanklı sıvı seviye kontrol sisteminde olası hatalı durumlar için incelenmiş ve büyük hatalar için bile başarımın yüksek olduğu belirlenmiştir. Bu yöntem önceden belirlenmiş hatalar için uygundur. Ayrıca hata toleranslı kontrol yönteminde kontrolörler arasında geçiş yapıldığında oluşabilecek dalgalanmalar da göz önüne alınmalıdır.

Üçüncü yöntem, yeniden yapılandırma mekanizması ‘çevrim-içi kontrolör hesaplayan’ türde olan bir aktif hata toleranslı kontrol yöntemidir. Geliştirilen yöntemde yeniden yapılandırma ve hata tanıma birimleri birbirlerinden bağımsız çalışırlar. Her ikisi de gerçek zamanlı sistem çıkışlarını kullanırlar. Destek vektörü bağlanım makineleri hem hata bulma ve tanıma işleminde hem de yeniden yapılandırma biriminde kullanılmıştır. Eğitimde kullanılacak hatalı ve hatasız durumlara ilişkin PID katsayıları, genetik algoritmayla çevrim-dışı belirlenir. Üç destek vektörü bağlanım makinesi, sistemden gelen veriyi eşzamanlı değerlendirerek PID kontrolörün katsayılarını çevrim-içi üretirler. Hatanın türünün tespit edilmesi için destek vektörü bağlanım makinelerini kullanan benzer bir işlem gerçekleştirilir. Yöntemin bulduğu çevrim-içi kontrolör katsayılarının, genetik arama algoritmasıyla çevrim-dışı elde edilen katsayılarla, neredeyse çakışan sonuçlar vermesi, optimal sonuca çok yaklaştığını göstermektedir. Ayrıca, çevrim-içi kontrolör seçen yöntemlerde karşılaşılabilecek olan sıçramalı geçiş sorununa önlem alınmasının gerekmemesi ve önceden belirlenmiş hatalarla sınırlı olmaması da yöntemin diğer olumlu yönleri olarak belirtilebilir.

Yöntemlerin başarımı, çift tanklı sıvı seviye kontrol sisteminin çeşitli hatalı durumlarını içeren benzetim örnekleri ile gösterilmiştir. Destek vektörü makinelerinin yüksek genelleştirme yeteneği ve hızlı yapısı, yöntemlerin başarımlarının da yüksek olmasını sağlamıştır. Buradan yola çıkarak daha sonra yapılabilecek olası araştırma konuları şöyle sıralanabilir:

- Uygun düzenlemelerle çok girişli-çok çıkışlı sistemlerde hata bulma, tanıma ve telafisi problemlerine uyarlanabilir.
- Yapay sinir ağları ve bulanık mantık gibi diğer akıllı algoritmalarla birlikte kullanılarak daha alternatifli çözümler geliştirilebilir.
- Gerçek sistemler üzerinde deneysel çalışmalar gerçekleştirilerek benzetim sonuçları ile karşılaştırılabilir.



## KAYNAKLAR

- Asokan, A., and Sivakumar, D.,** 2007: Model based fault detection and diagnosis using structured residual approach in a multi-input multi-output system. *Serbian Journal of Electrical Engineering*, Vol. 4, no. 2, pp. 133-145.
- Balle, P., and Isermann, R.** 1998: Fault detection and isolation for nonlinear process based on local linear fuzzy models and parameter estimation. *In Proc. American Control Conference, Philadelphia, PA.*, pp. 1605-1609.
- Belcastro, C. M., Belcastro, C. M.,** 2001: Application of fault detection, identification, and accommodation methods for improved aircraft safety. *In proceeding of the 2001 American control conference*, pp. 2623-2624.
- Blanke, M., Marcel, S., and Wu, E. N.,** 2001: Concept and methods in fault tolerant control. *Proceeding of the American Control Conference*, pp. 2606-2620.
- Blanke, M., Kinnaert, M., Lunze, I., and Staroswecki, M.,** 2003: Diagnosis and Fault Tolerant Control, Springer.
- Betta, G. C. L. G., and Pietrosanto, A.,** 1998: An advanced neural-network based instrument fault detection scheme. *IEEE Trans. Instrumentation and Measurement*, Vol. 47, no. 2, pp. 507-513.
- Campbell, C.,** 2002: Kernel Methods: A Survey of Current Techniques, *Neurocomputing* Vol. 48, pp. 63-84.
- Chandler, P. R.,** 1984: Self-repairing flight control system reliability and maintainability program – Executive overview. *In proceedings of the IEEE national aerospace and electronics conference*, pp. 586-590.
- Chen, J., Lopez-Toribio, C. J., and Patton, R. J.,** 1999: Nonlinear dynamic systems fault detection and isolation using fuzzy observers. *Proceeding of Institution of Mechanical Eng. Part 1: Journal of Systems and Control Eng.*, Vol. 213, no. 6, pp. 467-476.
- Chen, J., Patton, R. J., and Zhang, Y. H.,** 1996: Design of unknown input observer and robust fault detection filters. *International Journal of Control*. Vol. 63, no.1, pp. 85-105.
- Chow, E. Y., and Willsky, A. S.,** 1984: Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. on Automatic Cont.* Vol. 29, no. 7, pp. 603-614.

- Cocquempot, V., Staroswiecki, M., and Elmezyani, T., 2003:** Switching time estimation and fault detection for hybrid system using structure parity residuals. *In Proceeding IFAC Symposium SAFEPROCESS'03*, Washington, USA, pp. 2045-2055.
- Cocquempot, V., Staroswiecki, M., and Elmezyani, T., 2004:** Fault detection and isolation for hybrid system using structure parity residuals. *In Proceeding Asia Control Conference (ASCC)*, Australia, pp. 1203-1211.
- Cristianini, N., and Shawe-Taylor, J., 2000:** An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge.
- Dash, S., and Venkatasubramanian, V., 2000:** Challenges in the industrial application of fault diagnostic system. *Computer and Chemical Engineering*, Vol. **24**, pp.785-791.
- Douglas, R. K., and Speyer, J. L., 1996:** Robust fault detection filters design. *AIAA Journal of Guidance, Control, and Dynamics*. Vol. **19**, no. 1, pp. 214-218.
- Ducard, G., and Geering, H. P., 2006:** A reconfigurable flight control system based on the EMMAE method. *American Control Conference*, USA, pp. 5499-5504.
- El-Shal, S. M., and Morris, A. S., 2000:** A fuzzy expert system for fault detection in statistical process control of industrial processes. *IEEE Trans. System Man. Cybern.*, Vol. **30**, pp. 281-289.
- Eterno, J. S., Weiss, J. L., Looze, D. P., and Willsky, A. S., 1985:** Design issues for fault tolerant-restructurable aircraft control. *In proceedings of the 24<sup>th</sup> IEEE conference on decision and control*, pp. 900-905.
- Fagarasan, I., Ploix, S., and Gentil, S., 2004:** Causal fault detection and isolation based on a set-membership approach. *Automatica*, Vol. **40**, pp. 2099-2110.
- Filippetti, F., Franceschini, C., Tassoni, C., and Vas, P., 2000:** Recent developments of induction motor drives fault diagnosis using AI techniques. *IEEE Trans. Ind. Electron.* Vol. **47**, no. 5, pp. 994-1004.
- Frank, P. M., 1990:** Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results. *Automatica*, **26**, no. 3, pp. 459-474.
- Frank, P. M., and Ding X., 1994:** Frequency domain approach to optimality robust residual generation. *Automatica*, Vol. **30**, no. 5, pp. 789-804.
- Frank, P. M., and Koppen-Seliger B., 1997a:** Fuzzy logic and neural network applications to fault diagnosis. *International Journal of Approximate Reasoning*, Vol. **16**, no. 1, pp. 67-88.
- Frank, P. M., and Koppen-Seliger B., 1997b:** New developments using AI in fault diagnosis. *Engineering Applications of Artificial Intelligence*, Vol. **10**, no. 1, pp. 3-14.

- Füssel, D., Balle, P., and Isermann, R.,** 1997: Closed loop fault diagnosis based on a nonlinear process model and automatic fuzzy rule generation. *In Proceeding of 3rd IFAC Symposium SAFEPROCESS'97*, Hull,UK, pp. 359-364.
- Guasch, A., Quevedo, J., and Milne, R.,** 2000: Fault diagnosis for gas turbines based on the control systems. *Engineering Application of Artificial Intelligence*, Vol. **13**, no. 4, pp.477-483.
- Guo, H., Crossman J. A., Murphey, Y. L., and Coleman, M.,** 2000: Automotive signal diagnosis using Wavelets and machine learning. *IEEE Trans. Vehicular Tech.* Vol. **49**, no. 5, pp. 1650-1662.
- Hagenblad, A., Gustafsson, F., and Klein, I.,** 2003: A comparison of two methods for stochastic fault detection: The parity space approach and principle components analysis. *In Proceeding 13th IFAC Symposium on System Identification (SYSID)*, Rotterdam, Netherland. pp. 27-29.
- Han, Z., and Frank, P.,** 1997: Physical parameter estimation based FDI with neural networks. *In Proceeding IFAC Symposium SAFEPROCESS'97*, Hull, UK., pp. 294-299.
- Hermans, F. J., and Zarrop, M. B.,** 1997: Parameter estimation using sliding mode principles. *In Proceeding IFAC Symposium SAFEPROCESS'97*, Hull, UK., pp. 282-287.
- Hou, M., and Muller, P. C.,** 1994: Fault detection and isolation observers. *International Journal of Control.* Vol. **60**, no.5, pp. 827-846.
- Hsieh, C. S.,** 2002: Performance gain margins of two-stage LQ reliable control. *Automatica*, Vol. **38**, no. 11, pp. 1985-1990.
- Hsu, C. W., and Lin, C. J.,** 2002: A comparison of methods for multiclass support vector machines. *IEEE Trans. on Neural Networks*, Vol. **13**, no. 2, pp. 415-425.
- Huo, Z., and Fang, H.,** 2007: Research on robust fault tolerant control for networked control system with packet dropout. *Journal of Systems Engineering and Electronics*, Vol. **18**, no. 1, pp. 76-82.
- Isermann, R.,** 1984: Process fault detection based on modeling and estimation methods – A survey. *Automatica*, Vol. **20**, no. 4, pp. 387-404.
- Isermann, R.,** 1993: Fault diagnosis of machines via parameter estimation and knowledge processing. *Automatica*. Vol. **29**, pp. 815-836.
- Isermann, R.,** 1997: Supervision, fault detection and fault diagnosis methods. *Control Engineering Practice*, Vol. **5**, no. 5, pp. 639-652.
- Isermann, R.,** 2004: Model-based fault-detection and diagnosis - status and applications. *16th symposium on Automatic Control*, Petersburg, Russia.
- Isermann, R.,** 2005: Model-based fault-detection and diagnosis—Status and applications. *Annual Reviews in Control*, Vol. **29**, no. 1, pp. 71–85.
- Isermann, R.,** 2006: Fault-diagnosis systems: An introduction from fault detection to fault tolerance. *Berlin, Germany: Springer*.

- Jiang, B., and Chowdhury, F. N.,** 2005: Fault estimation and accommodation for linear MIMO discrete-time systems. *IEEE Trans. on Control Systems Technology*, Vol. **13**, no. 3, pp. 493-499.
- Jiang, J., Zhang, Y. M.,** 2006: Accepting performance degradation in fault tolerant control system design. *IEEE Trans. on Control Systems Technology*, Vol. **14**, no. 2, pp. 284-292.
- Jiang, J., and Zhao, Q.,** 2000: Design of reliable control systems possessing actuator redundancies. *Journal of Guidance, Control and Dynamics*, Vol. **23**, no. 4, pp. 709-718.
- Junli, C., and Licheng, J.,** 2000: Classification Mechanism of Support Vector Machines. *IEEE Proceedings of ICSP*, pp. 1556-1559.
- Kale, M. M., and Chipperfield, A. J.,** 2005: Stabilized mpc formulations for robust reconfigurable flight control. *Control Engineering Practise*, Vol. **13**, no. 6, pp. 771-778.
- Kecman, V.,** 2001: Learning and Soft Computing : Support Vector Machines, Neural Networks, and Fuzzy Logic Models, Cambridge, Mass. : MIT Press.
- Keller, J. Y., and Darouach, M.,** 1997: A new estimation for dynamic stochastic systems with unknown inputs: application to robust fault diagnosis. In *Proceeding IFAC SAFEPROCESS'97 Symposium*, Hull, U.K., pp. 177-180.
- Kim, K. S., Lee, K. J., and Kim, Y.,** 2003: Reconfigurable flight control system using direct adaptive method. *Journal of Guidance, Control, and Dynamics*, Vol. **26**, no. 4, pp. 543-550.
- Kourti, T.,** 2002: Process analysis and abnormal situation detection: From theory to practice. *IEEE Control System Magazine*, Vol. **22**, no. 5, pp. 10-25.
- Kovio, H. N.,** 1994: Artificial neural networks in fault diagnosis and control. *Control Eng. Practice*, Vol. **2**, no.7, pp. 89-101.
- Lee, K. S., and Vagners, J.,** 1997: Reliable decision unit utilizing fuzzy logic for observer based fault detection systems. In *Proceeding IFAC SAFEPROCESS'97 Symposium*, Hull, U.K., pp. 693-698.
- Liang, Y. W., Liaw, D.C., and Lee, T. C.,** 2000: Reliable control of nonlinear systems. *IEEE Trans. on Automatic Control*, Vol. **45**, no.4, pp. 706-710.
- Liao, F., Wang, J. L., and Yang, G. H.,** 2002: Reliable robust flight tracking control: An LMI approach. *IEEE Trans. on Control Systems Technology*, Vol. **10**, no. 1, pp. 76-89.
- Mahmoud, M., Jiang, J., and Zhang, Y.,** 2003a: Active fault tolerant control systems: Stochastic analysis and synthesis. *Lecture Notes in Control and Information Science*, Berlin, Springer.
- Mahmoud, M., Jiang, J., and Zhang, Y.,** 2003b: Stabilization of active fault tolerant control systems with imperfect fault detection and diagnosis. *Stochastic Analysis and Application*, Vol. **21**, no.3, pp. 673-701.

- Maki, Y., and Loparo, K. A.,** 1997: A neural network approach to fault detection and diagnosis in industrial processes. *IEEE Trans. Control System Tech.*, Vol. **5**, no. 6, pp. 529-541.
- Mangasarian, O. L., and Musicant, D. R.,** 2000: Robust Linear and Support Vector Regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. **22**, no.9.
- Maybeck, P. S.,** 1999: Multiple model adaptive algorithms for detecting and compensating sensor and actuator/surface failures in aircraft flight control systems. *International Journal of Robust and Nonlinear Control*, Vol. **9**, no. 4, pp. 1051-1070.
- Maybeck, P. S., Stevens, R. D.,** 1991: Reconfigurable flight control via multiple model adaptive control methods. *IEEE Trans. on Aerospace and Electronic Systems*, Vol. **27**, no. 3, pp. 470-479.
- Mendonça, L. F., Sousa, J. M. C., and Costa, J. M. G.,** 2009: An architecture for fault detection and isolation based on fuzzy methods. *Expert Systems with Applications*, Vol. **36**, no. 2, pp. 1092-1104.
- Miao, Q., and Wang, S.,** 2002: Nonlinear Model Predictive Control Based on Support Vector Regression. *IEEE Proceedings of the First International Conference on Machine Learning and Cybernetics*, Beijing, 4-5 November 2002.
- Mok, H. T., and Chan, C. W.,** 2008: Online fault detection and isolation of nonlinear systems based on neurofuzzy networks. *Engineering Applications of Artificial Intelligence*, Vol. **21**, no. 2, pp. 171-181.
- Monaco, J., Ward, D., Barron, R., and Bird, R.,** 1997: Implementation and flight test assessment of an adaptive, reconfigurable flight control system. *In Proceedings of 1997 AIAA Journal of Guidance, Control, and Dynamics*, pp. 1443-1454.
- Napolitano, M. R., Neppah, C., Casdorff, V., and Naylor, S.,** 1995: On-line learning nonlinear direct neurocontrollers for restructurable control systems, *Journal of Guadiance, Control, and Dynamics*, Vol. **18**, no. 1, pp. 170-176.
- Oblak, S., Skrjanc, I., and Blazic, S.,** 2005. On applying interval fuzzy model to fault detection and isolation for nonlinear input-output systems with uncertain parameters. *Proceedings of IEEE Conference on Control Applications*, Toronto, Canada, pp. 465-470.
- Oblak, S., Skrjanc, I., and Blazic, S.,** 2007. Fault detection for nonlinear systems with uncertain parameters based on the interval fuzzy model. *Engineering App. of Artificial Intelligent*, Vol. **20**, pp. 503-510.
- Ortaç-Kabaoğlu, R., Eksin, İ., Yeşil, E., ve Güzelkaya, M.,** 2009a: Fault Detection and Diagnosis for Nonlinear Systems: A Support Vector Machine Approach. *IFAC International Conference on Intelligent Control Systems and Signal Processing ICONS 09*.
- Ortaç-Kabaoğlu, R., Eksin, İ., Yeşil, E., ve Güzelkaya, M.,** 2009b: Fault Tolerant Control Systems: A Support Vector Machine Approach. *IFAC International Conference on DECOM' 09*.

- Ortaç-Kabaoğlu, R., Eksin, İ., Güzelkaya, M., ve Yeşil, E.,** 2009c: Destek Vektörü Makineleri ile Doğrudan Hata Toleranslı Kontrol. *TOK'09*.
- Patton, R. J.,** 1997: Fault-tolerant control: the 1997 situation. *In Proceeding of 3rd IFAC Symposium SAFEPROCESS'97*, Hull,UK, pp. 1033-1055.
- Patton, R. J., and Chen, J.,** 1991: Robust fault detection using eigenstructure assignment: tutorial consideration and some new results. *In Proceeding 30th.IEEE Conf. Decision and Control*, Brighton, England, pp. 2242-2247.
- Patton, R. J., and Chen, J.,** 1994: A review of parity space approaches to fault diagnosis for aerospace systems. *Journal of Guidance, Control and Dynamics*, Vol. 17, no. 2, pp. 278-285.
- Patton, R. J., Frank, P. M. and Clark, R. N.,** 2000: Issues of fault diagnosis for dynamic systems, *Springer*.
- Patton, R. J., and Kangethe, S. M.,** 1989: Robust fault detection using eigenstructure assignment of observers in fault diagnosis in dynamic systems – Theory and application. New York, Prentice Hall International.
- Patton, R. J., Uppal, F. J., and Lopez-Toribio, C. J.,** 2001: Soft Computing Approaches to Fault Diagnosis for Dynamic Systems : A Survey, Notes for Control and Intelligent System Engineering in the University of Hull, U.K.
- Polycarpou, M. M., and Vemuri, A. T.,** 1995: Learning methodology for failure detection and accommodation. *IEEE Control Systems Magazine*, Vol. 15, no. 3, pp. 16-24.
- Postalcioglu, S., Erkan, K., and Bolat, E.,** 2006: Discrete wavelet analysis based fault detection. *Wseas Trans Syst.*, Vol. 5, no. 10, pp. 2391-2398.
- Puig, V., and Quevedo, J.,** 2001: Fault-tolerant PID controllers using a passive robust fault diagnosis approach. *Control Engineering Practice*. Vol. 9, pp. 1221-1234.
- Rakar, A., and Juricic, D.,** 2002: Diagnostic reasoning under conflicting data: the application of the transferable belief model. *Journal of Process Control*, Vol. 12, no. 1, pp. 55-67.
- Rakar, A., Juricic, D., and Balle, P.,** 1999: Transferable belief model in fault diagnosis. *Engineering Application of Artificial Intelligence*, Vol. 12, no. 5, pp. 555-567.
- Shen, L. C., Chang, S. K., and Hsu, P. L.,** 1998: Robust fault detection and isolation with unstructured uncertainty using eigenstructure assignment. *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 21, no. 1, pp. 50-55.
- Simani, S., Fantuzzi, C., and Patton, R. J.,** 2003: Model-based Fault Diagnosis in Dynamic Systems using Identification Techniques. New York, NY: Springer.

- Smola, A., and Scholkopf, B.,** 1998: A Tutorial on Support Vector Regression. *NeuroCOLT2 Technical Report NC-TR-98-30.*
- Stengel, R. F.,** 1991: Intelligent failure-tolerant control. *IEEE Control Systems Magazine*, Vol. **11**, no. 4, pp. 14-23.
- Sun, Z., and Sun S.,** 2003: Optimal Control by Weighted Least Squares Generalized Support Vector Machines. *IEEE Proceedings of the American Control Conference Denver, Colorado June 4-6.*
- Suykens, J. A. K.,** 2001: Nonlinear Modelling and Support Vector Machines. *IEEE Instrumentation and Measurement Technology Conference*, Budapest, Hungary, May 21-23.
- Suykens, J. A. K., Vandewalle, J., De Moor, B.,** 2001: Optimal Control by Least Squares Support Vector Machines. *Neural Network*, Vol. **14**, p:23-35.
- Theilliol, D., Noura, H., and Ponsart J. C.,** 2002: Fault diagnosis and accomodation of a three tank system based on analytical redundancy. *ISA Transactions*, Vol. **41**, no. 3, pp. 365-382.
- Tortora, G.,** 2002: Fault-tolerant control and intelligent instrumentation. *Computer and Control Engineering Journal*, Vol. **13**, no. 5, pp. 259-262.
- Tsang, E. C. C., Yeung, D. S., and Chan, P. K.,** 2003: Fuzzy Support Vector Machines for Solving Two-Class Problems. *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, Wan, 2-5 November.
- Vapnik, V. N.,** 1995: The Nature of Statistical Learning Theory, *Springer-Verlag*. New York.
- Vapnik, V. N.,** 1998: Statistical Learning Theory, *John Wiley and Sons*, New York.
- Veillette, R. J., Medanic, L. V., and Porking, W. R.,** 1992: Design of reliable control systems. *IEEE Trans. Automatic Control*. Vol. **32**, no. 3, pp. 290-304.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. N.,** 2003: A review of process fault detection and diagnosis – part 1: Quantitative model based methods. *Computer and Chemical Engineering*. Vol. **27**, pp. 293-311.
- Walker, B. K., and Huang, K. Y.,** 1995: FDI by exented Kalman filter parameter estimation for industrial actuator benchmark. *Control Eng. Practice*, Vol. **3**, no. 12, pp. 1769-1774.
- Wolfe, P.,** 1961: A Duality Theory for Nonlinear Programming. *Quarterly of Applied Mathematics*, Vol. **19**, pp. 239-244.
- Wu, N. E., Zhang, Y. M., and Zhou, K.,** 2000: Detection, estimation, and accomodation of loss of control effectiveness. *International Journal of Adaptive Control and Signal Processing*, Vol. **14**, no. 7, pp. 775-795.
- Yang, G. H., Wang, J.L., and Soh, Y. C.,** 2000: Reliable LQG control with sensor failures. *IEE Proceedings – Control Theory and Applications*, Vol. **147**, no. 4, pp. 433-439.

- Yang, Y., Yang, G. H., and Soh, Y. C.,** 2001: Reliable control of discrete-time systems with actuator failure. *IEE Proceedings – Control Theory and Applications*, Vol. **147**, no. 4, pp. 428-432.
- Ye, H., Wang, G., and Ding, S. X.,** 2004: A new parity space approach for fault detection based on stationary wavelet transform. *IEEE Trans. Automatic Control*, Vol. **49**, no. 2, pp. 281-287.
- Yen, G. G., and Ho, L. W.,** 2003: Online multiple-model-based fault diagnosis and accomodation. *IEEE Trans. on Industrial Electronics*, Vol. **50**, no. 2, pp. 296-312.
- Yogg, D., Shafai, E., and Geering, H. P.,** 2001: A fault diagnosis for heat pumps. In *Proceeding IEEE International Conf. Control Applications*, Mexico City, pp. 70-76.
- Yoon, S., Landry, J. N., Kettaneh, N., Pepe, W., and Wold, S.,** 2003: Multivariate process monitoring and early fault detection (MSPC) using PCA and PLS. *Plant Automation and Decision Support Conf., Hyatt Regency, San Antonio, Texas*.
- Yu, D.L., Chang, T., and Yu, D. W.,** 2005: Fault tolerant control of multivariable process using auto-tuning PID controller. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. **35**, no. 1, pp. 32-43.
- Zhang, H., Wang, X., Zhang, C., and Xu, X.,** 2005: Modeling Nonlinear Dynamical Systems Using Support Vector Machine, Proceeding of 4. International Conf. On Machine Learning and Cybernetics, pp. 3204-3209.
- Zhang, Y. M., and Jiang, J.,** 2001a: Integrated active fault tolerant control using IMM approach. *IEEE Trans. Aerospace and Electronic Systems*, Vol. **37**, no. 4, pp. 1221-1235.
- Zhang, Y. M., and Jiang, J.,** 2001b: Integrated design of reconfigurable fault tolerant control systems. *Journal of Guidance, Control, and Dynamics*, Vol. **24**, no. 1, pp. 133-136.
- Zhang, Y. M., and Jiang, J.,** 2002: An active fault tolerant control system against partial actuator failures. *IEE Proceedings-Control Theory and Applications*, Vol. **149**, no. 1, pp. 95-104.
- Zhang, Y. M., and Jiang, J.,** 2006: Issues on integration of fault diagnosis and reconfigurable control in active fault tolerant control systems. *IFAC Fault detection, supervision and safety of technical processes*, pp. 1437-1448.
- Zhang, Y. M., and Jiang, J.,** 2008: Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, Vol. **32**, pp. 229-252.
- Zhang, X., Parisini, T., and Polycarpou, M. M.,** 2004: Adaptive fault-tolerant control of nonlinear uncertain systems: An information-based diagnostic approach. *IEEE Trans. on Automatic Cont.* Vol. **49**, no. 8, pp. 1259-1274.

- Zhao, S., and Xu, Z.,** 2004: Design of a novel knowledge-based fault detection and isolation scheme, *IEEE Tran. System man and cybernetics*, Vol. **34** no.2 pp.1089-1095.
- Zhong, M., Fang, H., and Ye, H.,** 2007: Fault diagnosis of networked control systems. *Annual Reviews in Control*, Vol. **31**, no. 1, pp.55–68.
- Zio, E., and Gola, G.,** 2006: Neuro-fuzzy pattern classification for fault diagnosis in nuclear components. *Annals of Nuclear Energy*, Vol. **33**, no. 5, pp. 415-426.



## ÖZGEÇMİŞ

**Ad Soyad:** Rana ORTAÇ KABAOĞLU

**Doğum Yeri ve Tarihi:** İstanbul, 15 Temmuz 1976

**Adres:** Şehit Fethi S. Çim Ap. No:14/15 Acıbadem Üsküdar İstanbul

**Lisans Üniversitesi:** İstanbul Üniversitesi – Elektrik-Elektronik Mühendisliği

### Yayın Listesi:

- Ortaç-Kabaoğlu, R., Eksin, İ., Yeşil, E., ve Güzelkaya, M., 2009a: Fault Detection and Diagnosis for Nonlinear Systems: A Support Vector Machine Approach. *IFAC International Conference on Intelligent Control Systems and Signal Processing ICONS 09*.
- Ortaç-Kabaoğlu, R., Eksin, İ., Yeşil, E., ve Güzelkaya, M., 2009b: Fault Tolerant Control Systems: A Support Vector Machine Approach. *IFAC International Conference on DECOM' 09*.
- Ortaç-Kabaoğlu, R., Eksin, İ., Güzelkaya, M., ve Yeşil, E., 2009c: Destek Vektörü Makineleri ile Doğrudan Hata Toleranslı Kontrol. *TOK' 09*.
- Ortaç-Kabaoğlu, R., Eksin, İ., Güzelkaya, M., ve Yeşil, E., 2010: Active Fault Tolerant Control Methods based on Support Vector Machine. *Mathematics and Computers in Simulation, gönderildi.*

