

YAPAY ZEKA YÖNTEMLERİYLE  
ROBOTLARDA MODEL TEMELLİ  
ARIZA YALITIMI VE ARIZAYA  
DAYANIKLI DENETİM

TOLGA YÜKSEL  
DOKTORA TEZİ  
ELEKTRİK-ELEKTRONİK  
MÜHENDİSLİĞİ ANABİLİM DALI

**T.C.  
ONDOKUZ MAYIS ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**YAPAY ZEKA YÖNTEMLERİYLE ROBOTLARDA MODEL  
TEMELLİ ARIZA YALITIMI VE ARIZAYA DAYANIKLI  
DENETİM**

**TOLGA YÜKSEL**

**DOKTORA TEZİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI**

**AKADEMİK DANIŞMAN  
Yrd. Doç. Dr. ABDULLAH SEZGİN**

**SAMSUN-2010**

**T.C.**  
**ONDOKUZ MAYIS ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**Bu çalışma jürimiz tarafından ...../...../2010 tarihinde yapılan sınav ile Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda DOKTORA tezi olarak kabul edilmiştir.**

**Başkan :**

**Üye :**

**Üye :**

**Üye :**

**Üye :**

**ONAY :**

**Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.**

**...../...../2010**

**Prof. Dr. Hasan GÜMÜŞ**  
**Fen Bilimleri Enstitüsü Müdürü**

## YAPAY ZEKA YÖNTEMLERİYLE ROBOTLARDA MODEL TEMELLİ ARIZA YALITIMI VE ARIZAYA DAYANIKLI DENETİM

### ÖZ

Gelişmiş kontrol yöntemleri yaygınlaşırken, kesinlik özelliklerinin yanında, güvenilirlik ve olası arızaların bulunması, yalıtılması ve mümkünse arızalara dayanıklı olması özellikle tehlikeli otomatik kontrol dizgeleri için bir zorunluluk olmuştur. Bu tez çalışmasında robot manipülatörler için yapay zeka yöntemlerini kullanan ve gerçek dizge ile dizge modeli arasındaki fark işaretlerinin türetilmesi ve bu işaretlerin değerlendirilmesinden oluşan model temelli üç farklı arıza bulma-yalıtım şeması ve kazanç çizgeleme yöntemini kullanan bir arızaya dayanıklı denetim şeması önerilmiştir.

Tez çalışmasında önerilen şemalar ANFIS ve sinir ağı gibi yapay zeka yöntemlerini modelleyici ve artık değerlendirici olarak kullanmaktadır.

Önerilen 1. şema, arıza bulma için artıkların ve analitik fazlalıkların kendileri için tanımlı eşiklerin aşımalarını kullanırken, arıza yalıtımı için bu işaretlerin giriş olarak verildiği bir sinir ağını kullanmaktadır. Önerilen 2. şema ise genelleştirilmiş gözlemci şeması yaklaşımından yararlanarak her biri bir arızayı tanımlayan arızalı manipülatör modellerini gerçek manipülatör dizgesiyle artık işaretleri elde edecek şekilde kullanmakta ve bu işaretler karar çizelgesiyle yorumlanarak aynı anda arıza bulma ve yalıtım işlemleri gerçekleştirilmektedir. Önerilen 3. şema ise yakınsayıcı M-ANFIS'ler ile arıza işlevine yakınsayarak diğer iki şemadan farklılaşmaktadır.

3. şemasının uzantısı olarak önerilen arızaya dayanıklı denetim şeması ise kazanç çizgeleme yöntemi ile robot manipülatöre ait denetliyicinin kazanç katsayılarını elde ettiği arıza işlevine göre değiştirerek arıza sonrasında geçici durum ve sürekli durum hatalarının arızalı durumdan daha az olmasını sağlamaktadır.

Önerilen şemaların benzetimleri MATLAB altında yapılmıştır. Ayrıca önerilen şemalar ileriki çalışmalarda gerçek bir manipülatör üzerinde kullanılmak üzere tümünden TMS320C6713 DSP üzerine ve ek olarak MATLAB-DSP ve Visual Basic-DSP arasında haberleşme ile C, MATLAB ve Visual Basic dilleri kullanılarak gömülmüştür.

**Anahtar Sözcükler** : Manipulator, arıza bulma-yalıtımı, arızaya dayanım, M-ANFIS, sinir ağı

# **MODEL BASED FAULT ISOLATION AND FAULT TOLERANT CONTROL OF ROBOT MANIPULATORS USING ARTIFICIAL INTELLIGENCE**

## **METHODS**

### **ABSTRACT**

While advanced control methods are becoming widespread, in addition to accuracy specifications, reliability and detection and isolation of possible faults and fault tolerance, if possible, have become an obligation for especially automatic control systems. In this thesis study, for robot manipulators, three different model based fault detection-isolation schemes which are using artificial intelligence methods and which are formed of generation of difference signals between the real system and the model of the system and evaluation of these signals and a fault tolerant control scheme which is using gain scheduling method are proposed.

Proposed schemes in the thesis study are using artificial intelligence methods like ANFIS and neural network as modeller and residual evaluator.

The 1. proposed scheme is using the excess of residuals and analytical redundancies defined for each for fault detection as using a neural network with these signals as inputs for fault isolation. The 2. proposed scheme is using faulty manipulator models defining each fault by utilizing the approach of generalized observer scheme to obtain residual signals with real manipulator and fault detection and isolation operations are implemented simultaneously by interpreting these signals with a decision table. The 3. proposed scheme differs from the other two schemes by approximating fault function with M-ANFIS approximators

As an extension of the 3. scheme, the proposed fault tolerant control scheme proves less transient and steady state errors than the faulty condition by scheduling gain coefficients of the controller of the robot manipulator with gain scheduling method.

Simulations of the proposed schemes are implemented under MATLAB. Furthermore, in order to use in the future studies on a real manipulator, the proposed schemes are embedded wholly on TMS320C6713 DSP and additionally, with communication between MATLAB-DSP and Visual Basic-DSP, they are embedded using C, MATLAB and Visual Basic languages.

**Keywords** : Manipulator, fault detection-isolation, fault tolerance, M-ANFIS, neural network

## TEŐEKKÖR

Bu alıőmamın tamamlanmasında benden desteęini esirgemeyen aileme, arkadaşlarıma ve danışmanım Yrd. Do. Dr. Abdullah SEZGİN'e teőekkÖr ederim.

Ayrıca alıőmamı Ondokuz Mayıs Üniversitesi Araőtırma Fonu, PYO.MUH.1904.09.001 ile destekleyen O.M.Ö. Proje Yönetim Ofisi'ne teőekkÖrü bir bor bilirim.

## İÇİNDEKİLER

<b>1. GİRİŞ.....</b>	<b>1</b>
<b>1.1. Arıza ve Arıza Bulma ile İlgili Tanımlar.....</b>	<b>1</b>
<b>1.2. Arıza Tipleri.....</b>	<b>4</b>
<b>1.2.1. Oluşma Yerlerine Göre Arıza Tipleri.....</b>	<b>4</b>
1.2.1.1. Hareketlendirici (Actuator) Arızaları.....	5
1.2.1.2. Bileşen (Component) Arızaları.....	5
1.2.1.3. Algılayıcı (Sensor) Arızaları.....	5
<b>1.2.2. Modellenmelerine Göre Arızalar.....</b>	<b>6</b>
1.2.2.1. Toplamlı (Additive) Arızalar.....	7
1.2.2.2. Çarpımlı (Multiplicative) Arızalar.....	7
<b>1.2.3. Zaman Karakteristiklerine Göre Arızalar.....</b>	<b>7</b>
1.2.3.1. Ani (Abrupt) Arızalar.....	7
1.2.3.2. Artan (Incipient) Arızalar.....	7
1.2.3.3. Geçici (Intermittent) Arızalar.....	8
<b>1.3. Model Temelli Arıza Bulma ve Yalıtım Yöntemleri.....</b>	<b>8</b>
1.3.1. Gözlemci (Observer) Temelli Yaklaşımlar.....	10
1.3.2. Eşlik Vektörü-İlişkisi (Parity Vector-Relation) Yaklaşımları.....	11
1.3.3. Parametre Kestirim (Parameter Estimation) Yaklaşımları.....	11
1.3.4. Gürbüz Artık Türetim Problemi.....	11
1.3.5. Doğrusal Olmayan Dizgeler İçin Arıza Bulma ve Yalıtımı.....	13
1.3.5.1. Doğrusal Olmayan Benzerlik Gözlemcisi (Nonlinear Identity Observer-NIO) Yaklaşımı.....	14
1.3.5.2. Doğrusal Olmayan Bilinmeyen Giriş Gözlemcisi (Nonlinear Unknown Input Observer-NUIO) Yaklaşımı.....	15
1.3.5.3. Bozucu Ayırıcı Doğrusal Olmayan Gözlemci (Disturbance Decoupled Nonlinear Observer-DDNO) Yaklaşımı.....	16
1.3.5.4. Sinir Ağı Gözlemcisi (Neural Network Observer) Yaklaşımı ve Sinir Ağı ile Artık Değerlendirme.....	17
1.3.5.5. Bulanık Gözlemciler (Fuzzy Observers) Yaklaşımı ve Bulanık Mantık ile Artık Değerlendirme.....	19

<b>1.4. Arızaya Dayanıklı Denetim.....</b>	<b>21</b>
<b>1.4.1. Pasif FTC.....</b>	<b>22</b>
<b>1.4.2. Aktif FTC.....</b>	<b>22</b>
<b>1.4.3. Doğrusal Olmayan Dizgeler İçin Arızaya Dayanıklı Denetim</b>	
<b>Tasarımı.....</b>	<b>24</b>
<b>1.4.3.1. Nazik Başarım Düşümü (Graceful Performance Degradation).....</b>	<b>26</b>
<b>1.4.3.2. Kararlılık ve Güzbüzlük.....</b>	<b>26</b>
<b>1.4.3.3. FDI Belirsizlikleri, Yapılandırma Gecikmesi ve Gerçek Zamanda</b>	
<b>Çalışma.....</b>	<b>26</b>
<b>2. ROBOT MANİPÜLATÖRLERDE ARIZA BULMA, YALITIMI VE</b>	
<b>ARIZAYA DAYANIKLI DENETİM.....</b>	<b>28</b>
<b>2.1. Robot Manipülatörlerde Görülen Arızalar.....</b>	<b>29</b>
<b>2.1.1. Robot Manipülatörlerde Hareketlendirici Arızaları.....</b>	<b>29</b>
<b>2.1.2. Robot Manipülatörlerde Bileşen Arızaları.....</b>	<b>30</b>
<b>2.1.3. Robot Manipülatörlerde Algılayıcı Arızaları.....</b>	<b>30</b>
<b>2.2. Robot Manipülatörlerde Arıza Bulma ve Yalıtımı.....</b>	<b>30</b>
<b>2.3. Robot Manipülatörlerde Arızaya Dayanıklı Denetim.....</b>	<b>32</b>
<b>2.3.1. Robot Manipülatörlerde Donanımsal Fazlalıklar İle Arızaya</b>	
<b>Dayanıklı Denetim.....</b>	<b>32</b>
<b>2.3.2. Robot Manipülatörlerde Denetim Yöntemleri ile Arızaya</b>	
<b>Dayanıklı Denetim.....</b>	<b>34</b>
<b>2.4. Tez Çalışmasının Özgün Değeri.....</b>	<b>35</b>
<b>3. TEZ ÇALIŞMASINDA KULLANILAN YAPAY ZEKA YÖNTEMLERİ...</b>	<b>39</b>
<b>3.1. SA ve Esnek Yayılım İle Öğrenme.....</b>	<b>39</b>
<b>3.2. ANFIS ve M-ANFIS.....</b>	<b>41</b>
<b>4. ÖNERİLEN FDI VE FTC ŞEMALARI.....</b>	<b>46</b>
<b>4.1. Önerilen 1. FDI Şeması - Robot Manipülatörlerde M-ANFIS ve SA</b>	
<b>Kullanarak Arıza Bulma ve Yalıtımı.....</b>	<b>46</b>
<b>4.1.1. M-ANFIS İle Artık Türetimi/Arıza Bulma.....</b>	<b>47</b>
<b>4.1.2. Sinir Ağı İle Artık Değerlendirme/Arıza Yalıtımı.....</b>	<b>49</b>
<b>4.2. Önerilen 2. FDI Şeması - Robot Manipülatörlerde M-ANFIS’li</b>	
<b>Genelleştirilmiş Gözlemcileri Kullanarak Arıza Bulma ve Yalıtımı.....</b>	<b>49</b>

4.3. Önerilen 3. FDI Şeması - Robot Manipulatörlerde M-ANFIS İle Arıza İşlevine Yakınsama.....	51
4.4. Önerilen FTC Şeması - Robot Manipulatörlerde Kazanç Çizgeleme İle Arızaya Dayanıklı Denetim.....	53
5. BENZETİM ÇALIŞMALARI ve BULGULAR.....	55
5.1. Önerilen 1. FDI Şemasına Ait Benzetim Sonuçları.....	57
5.2. Önerilen 2. FDI Şemasına Ait Benzetim Sonuçları.....	65
5.3. Önerilen 3. FDI Şemasına Ait Benzetim Sonuçları.....	76
5.4. Önerilen FTC Şemasına Ait Benzetim Sonuçları.....	80
6. ÖNERİLEN ŞEMALARIN TMS320C6713 DSP ÜZERİNE GÖMÜLMESİ VE BULGULAR.....	85
7. SONUÇLAR, TARTIŞMA ve ÖNERİLER.....	89
KAYNAKLAR.....	94
Ek A. Hesaplanmış Moment-PID (HM-PID) Denetimi.....	103
Ek B. Modelleyici M-ANFIS Eğitimi İçin Kullanılan Yörüngeler.....	104
Ek C. Önerilen 1. FDI Şemasına Ait MATLAB Programı ve Akış Diyagramı.	106
Ek D. Önerilen 2. FDI Şemasına Ait MATLAB Programı ve Akış Diyagramı.	112
Ek E. Önerilen 3. FDI Şemasına Ait MATLAB Programı ve Akış Diyagramı.	119
Ek F. Önerilen FTC Şemasına Ait MATLAB Programı ve Akış Diyagramı.....	126
Ek G. 1. Şema İçin DSP programı.....	135
Ek H. 2. Şema İçin MATLAB-DSP Programları.....	148
Ek I. 3. Şema İçin Visual Basic-DSP Programları.....	160
YAYINLAR LİSTESİ.....	171
ÖZGEÇMİŞ.....	173

## SİMGELER LİSTESİ

$x(t)$	: Durum vektörü
$u(t)$	: Denetim giriş vektörü
$u_R(t)$	: $u(t)$ denetim girişine karşılık hareketlendirici tepke vektörü
$y(t)$	: Dizge çıkış vektörü
$y_R(t)$	: Gözlenen dizge çıkış vektörü
$A, B, C, D$	: Durum matrisleri
$f(t)$	: Arıza vektörü
$f_a(t)$	: Hareketlendirici arıza işlev vektörü
$f_c(t)$	: Bileşen arıza işlev vektörü
$f_s(t)$	: Algılayıcı arıza işlev vektörü
$G_u(s)$	: Denetim girişi geçiş matrisi
$G_f(s)$	: Arıza işlevi geçiş matrisi
$t_f, T$	: Arıza anı
$r(t)$	: Artık işlev vektörü
$d(t)$	: Bilinmeyen giriş (bozucu) vektörü
$E_1, E_2$	: Bilinmeyen giriş dağılım matrisleri
$\Delta A, \Delta B, \Delta C, \Delta D$	: Modelleme hata matrisleri
$G_d(s)$	: Bilinmeyen giriş geçiş matrisi
$\xi(t), \hat{x}(t)$	: Durum kestirim vektörleri
$R_1, R_2$	: Arıza işlev matrisleri
$\hat{y}(t)$	: Çıkış kestirim vektörü
$e(t)$	: Hata vektörü
$w(t)$	: Çalışma noktası tanımlamada kullanılan vektör
$\mu_i(.)$	: Üyelik işlev değeri
$q(t)$	: Açısal konum vektörü
$\dot{q}(t)$	: Açısal hız vektörü
$\ddot{q}(t)$	: Açısal ivme vektörü
$M(.)$	: Eylemsizlik matrisi
$\eta_M$	: Eylemsizlik matrisi belirsizliği

$V(.)$	: Merkezci ve merkezkaç vektörü
$\eta_V$	: Merkezci ve merkezkaç vektörü belirsizliği
$G(.)$	: Yerçekimi vektörü
$\eta_G$	: Yerçekimi vektörü belirsizliği
$F(.)$	: Sürtünme vektörü
$\tau(.)$	: Eklemlere uygulanan moment vektörü
$\alpha$	: Kısmi eklem arıza matrisi
$w_{ij}$	: SA için $i$ . nöronun $j$ . nörona ağırlık
$E(n)$	: SA için $n$ . adımdaki hata işlevi
$\eta$	: SA için öğrenme oranı

**KISALTMALAR LİSTESİ**

FDI	: Fault Detection and Isolation - Arıza Bulma ve Yalıtımı
TBA	: Temel Bileşenler Analizi
ARMA	: Auto Regressive Moving Average
ARMAX	: Auto Regressive Moving Average with Exogenous Outputs
SA	: Sınır Ağları
BM	: Bulanık Mantık
FTC	: Fault Tolerant Control - Arızaya Dayanıklı Denetim
FTCS	: Fault Tolerant Control Systems - Arızaya Dayanıklı Denetim Dizgeleri
SVM	: Support Vector Machines - Destek Vektör Makinaları
RMMS	: Reconfigurable Modular Manipulator System
RLWR	: Reconfigurable Light Weight Robots
DSP	: Digital Signal Processor - Sayısal İşaret İşleyici
GA	: Genetik Algoritmalar
RP	: Resilient Propagation - Esnek Yayılım
ANFIS	: Adaptive Neuro-Fuzzy Inference System
M-ANFIS	: Multiple ANFIS - Çoklu ANFIS
C-ANFIS	: Coactive ANFIS
BP	: BackPropagation - Geriye Yayılım
HM-PID	: Hesaplanmış Moment-PID
DOS	: Dedicated Observer Scheme - Atanmış Gözlemci Şeması
GOS	: Generalized Observer Scheme - Genelleştirilmiş Gözlemci Şeması

## ŞEKİLLER LİSTESİ

Şekil 1.1. Oluşma yerlerine göre arızaların gösterilimi.....	4
Şekil 1.2. Toplamlı arıza gösterilimi.....	7
Şekil 1.3. Çarpımlı arıza gösterilimi.....	7
Şekil 1.4. Ani arızaların zaman içindeki davranışı.....	7
Şekil 1.5. Artan arızaların zaman içindeki davranışı.....	8
Şekil 1.6. Geçici arızaların zaman içindeki davranışı.....	8
Şekil 1.7. Model temelli FDI yapısı.....	9
Şekil 1.8. Yapısal artık kümeleri.....	10
Şekil 1.9. Yönlü artık vektörü gösterilimi.....	10
Şekil 1.10. Uyarlanabilir eşğin uygulanması.....	13
Şekil 1.11. Doğrusal olmayan bir dizgenin sinir ağı modeli.....	18
Şekil 1.12. İki sinir ağı kullanarak arıza bulma ve yalıtımı.....	19
Şekil 1.13. Üyelik işlevi gösterilimi.....	20
Şekil 1.14. Pasif FTC'in genel yapısı.....	22
Şekil 1.15. Aktif FTC'in genel yapısı.....	23
Şekil 2.1 Donanımsal fazlalıklı modüler robotlar.....	34
Şekil 3.1. ANFIS'in genel yapısı.....	42
Şekil 3.2. $(a_i, b_i, c_i)$ parametrelerinin genelleştirilmiş-çan üyelik işlevine etkileri....	42
Şekil 3.3. M-ANFIS'in genel yapısı.....	44
Şekil 4.1. Önerilen 1. FDI şeması.....	46
Şekil 4.2. M-ANFIS ile tek eklemlı robot manipülatör modelleme.....	47
Şekil 4.3. M-ANFIS ile artık türetimi.....	47
Şekil 4.4. Artıklar ve analitik fazlalıklar ile arıza bulma.....	48
Şekil 4.5. Sinir ağı ile arıza yalıtımı.....	49
Şekil 4.6. Önerilen 2. FDI şeması.....	50
Şekil 4.7. Önerilen 3. FDI şeması.....	52
Şekil 4.8. M-ANFIS ile arıza işlevine yakınsama.....	52
Şekil 4.9. Önerilen FTC Şeması.....	53
Şekil 4.10. KÇ yöntemi ile arızaya dayanıklı denetim.....	53
Şekil 5.1. İki eklemlı, düzlemsel, yerçekimi altındaki robot manipülatörün gösterilimi.....	55
Şekil 5.2. Robot manipülatörün x-y düzleminde izlediği yörünge.....	59

Şekil 5.3. Eklemlerin izlediği yörüngeler.....	60
Şekil 5.4. Eklemlerin yaptığı yörünge hataları.....	60
Şekil 5.5. Eklemlere uygulanan momentler.....	61
Şekil 5.6. 1. şema için artık $r_1$ .....	61
Şekil 5.7. 1. şema için artık $r_2$ .....	62
Şekil 5.8. 1. şema için artık $r_3$ .....	62
Şekil 5.9. 1. şema için artık $r_4$ .....	63
Şekil 5.10. 1. şema için artık $r_1'$ .....	63
Şekil 5.11. 1. şema için artık $r_3'$ .....	64
Şekil 5.12. 1. şema için sinir ağı çıkışları.....	64
Şekil 5.13. 1. şema için arıza yalıtım işaretleri.....	65
Şekil 5.14. 2. şema için artık $r_{1f_1}$ .....	67
Şekil 5.15. 2. şema için artık $r_{2f_1}$ .....	68
Şekil 5.16. 2. şema için artık $r_{3f_1}$ .....	68
Şekil 5.17. 2. şema için artık $r_{4f_1}$ .....	69
Şekil 5.18. 2. şema için artık $r_{1f_2}$ .....	69
Şekil 5.19. 2. şema için artık $r_{2f_2}$ .....	70
Şekil 5.20. 2. şema için artık $r_{3f_2}$ .....	70
Şekil 5.21. 2. şema için artık $r_{4f_2}$ .....	71
Şekil 5.22. 2. şema için artık $r_{1f_3}$ .....	71
Şekil 5.23. 2. şema için artık $r_{2f_3}$ .....	72
Şekil 5.24. 2. şema için artık $r_{3f_3}$ .....	72
Şekil 5.25. 2. şema için artık $r_{4f_3}$ .....	73
Şekil 5.26. 2. şema için artık $r_{1f_4}$ .....	73
Şekil 5.27. 2. şema için artık $r_{2f_4}$ .....	74
Şekil 5.28. 2. şema için artık $r_{3f_4}$ .....	74
Şekil 5.29. 2. şema için artık $r_{4f_4}$ .....	75
Şekil 5.30. 2. şema için arıza yalıtım işaretleri.....	75
Şekil 5.31. 3. FDI şeması için M-ANFIS ile arıza yakınsama ve arıza yalıtımı.....	77
Şekil 5.32. 3. FDI şeması için M-ANFIS ile arıza yakınsama işlevi.....	78
Şekil 5.33. 3. FDI şeması için Eklem 1 arıza işlevine yakınsama hatası.....	79
Şekil 5.34. 3. FDI şeması için arıza yalıtım işaretleri.....	79
Şekil 5.35. FTC şeması için eklemlerin yaptığı yörünge hataları.....	83

<b>Şekil 5.36.</b> FTC şeması için eklemelerin arıza işlevine yakınsama işlevleri.....	<b>84</b>
<b>Şekil 5.37.</b> FTC şeması için Eklem 1 arıza işlevine yakınsama hatası.....	<b>84</b>
<b>Şekil 6.1.</b> TMS320C6713 DSP'ye ait görünüm.....	<b>86</b>

## ÇİZELGELER LİSTESİ

<b>Çizelge 3.1.</b> Melez öğrenme için iki yönlü geçişte parametrelerin hesaplanması.....	<b>44</b>
<b>Çizelge 4.1.</b> Karar çizelgesi.....	<b>51</b>
<b>Çizelge 5.1.</b> Benzetim için tanımlanan arızalar ve isimlendirilmesi.....	<b>57</b>
<b>Çizelge 5.2</b> 2. FDI şeması için karar çizelgesi.....	<b>67</b>
<b>Çizelge 5.3.</b> 3. şema için tanımlanan arızalar ve isimlendirilmesi.....	<b>76</b>
<b>Çizelge 5.4.</b> $q_{d1}(t) = 0.8 * \cos(t/2) \text{ rad.}$ , $q_{d2}(t) = 0.8 * \sin(t/2) \text{ rad.}$ için Eklem 1'de %50 arıza ( $f_1$ ) ( <i>FTC oranı</i> = 0.50157) durumunda güncelleme karşılaştırması.....	<b>80</b>
<b>Çizelge 5.5.</b> $q_{d1}(t) = 0.8 * \cos(t/2) \text{ rad.}$ , $q_{d2}(t) = 0.8 * \sin(t/2) \text{ rad.}$ için Eklem 1'de %70 arıza ( $f_5$ ) ( <i>FTC oranı</i> = 0.69804) durumunda güncelleme karşılaştırması.....	<b>81</b>
<b>Çizelge 5.6.</b> $q_{d1}(t) = 0.8 * \cos(t/2) \text{ rad.}$ , $q_{d2}(t) = 0.8 * \sin(t/2) \text{ rad.}$ için Eklem 1'de %50 arıza ( $f_1$ ) ( <i>FTC oranı</i> = 0.50157) durumunda zamana bağlı artım karşılaştırılması.....	<b>82</b>
<b>Çizelge 5.7.</b> $q_{d1}(t) = 0.8 * \cos(t/2) \text{ rad.}$ , $q_{d2}(t) = 0.8 * \sin(t/2) \text{ rad.}$ için Eklem 1'de %70 arıza ( $f_2$ ) ( <i>FTC oranı</i> = 0.69804) durumunda zamana bağlı artım karşılaştırılması.....	<b>83</b>
<b>Çizelge 7.1</b> Öneriln FDI şemalarının karşılaştırılması.....	<b>89</b>
<b>Çizelge 7.2</b> Öneriln FTC şemasının özellikleri.....	<b>92</b>

## 1. GİRİŞ

Gelişen teknoloji ile birlikte karmaşık ve tümleşmiş dizgelerin daha güvenilir, daha dayanıklı ve daha az arızalanan olması isteği ortaya çıkmış, özellikle 1994 yılındaki USAir 427 uçuş numaralı Boeing 737-300 ve 1997 yılındaki UA 585 uçuş numaralı Boeing 737-200 uçaklarda denetim yüzeyinin kaybedilmesiyle oluşan ölümlü kazaların örnek gösterilebileceği olaylar bu isteği bir zorunluluk yapmıştır (National Transportation Safety Board, 1999; AirDisaster.Com, 2000). Bu zorunluluk dizgelere ait işaretlerin işlenerek anlamlandırılması, oluşan yada oluşabilecek olası arızaların bulunup birbirinden yalıtılması (fault detection and isolation-FDI) ve oluşan arızalara karşılık dizgelerin yapılandırılması (reconfiguration) gibi işlemleri işaret işleme ve otomatik kontrolün önemli araştırma alanlarından biri yapmıştır.

70'lerin başlarında bu konularla ilgili ilk çalışmalar ortaya çıkmıştır. Arıza bulmayla ilgili ilk çalışmalar kimyasal süreçlerin izlenmesi için ortaya çıkmışken daha sonra güvenilirliğin önemli olduğu hava ve uzay araçları, arabalar, nükleer reaktörler, türbinler, iklimlendirme sistemleri gibi birçok alana yayılmıştır. 1991 yılında çalışmaların yaygınlaşmasıyla IFAC SAFEPROCESS Komitesi kurulmuş, 1993 yılında arıza bulma ve yalıtımı, arıza teşhisi ve arızaya dayanıklı denetim ile ilgili durumlar ve işaretler, işlevler, modeller ve dizge özellikleri başlıkları altında çeşitli tanımlamalar yapılmıştır (Isermann ve Ballé, 1997). Bu tanımlamalar izleyen bölümde gerekli gruplandırmalarıyla birlikte verilmiştir.

### 1.1. Arıza ve Arıza Bulma ile İlgili Tanımlar

Arıza , dizge işlevinde fiziksel bozunum veya çökme olmasa da beklenmeyen değişim olarak tanımlanabilir (Chen ve Patton, 1999). Bu nedenle arıza otomatik bir dizgenin olağan çalışmasını rahatsız eder, böylece dizgenin başarımında kabul edilemez bir düşüşe veya tehlikeli durumlara yol açar. Arıza terimi bozunum (failure) terimi yerine kullanılmaktadır çünkü yıkımsal bir etki göstermemektedir. Bozunum terimi bir dizgenin bileşeninde veya işlevinde tamamen bir çöküşü tanımlarken, arıza oluştuğu anda dayanılabilir bir tutukluğu tanımlar. Bu tanımlamalara ek olarak aşağıdaki terimler şu şekilde tanımlanmaktadır (Chen ve Patton, 1999):

#### a) Durumlar ve İşaretler

##### Arıza (Fault)

Bir dizgenin en az bir karakteristik özelliğinin yada parametresinin kabul

edilebilir, olağan yada standart durumundan izinsiz olarak sapması

**Bozunum (Failure)**

Bir dizgenin belirlenen çalışma koşulları altında gerekli bir işlevi yerine getirme yeteneğindeki sürekli kesilme

**Tutukluk (Malfunction)**

Bir dizgenin istenen işlevi yerine getirme yeteneğindeki kesikli düzensizlik

**Hata (Error)**

Bir çıkış değişkeninin ölçülmüş yada hesaplanmış değeri ile doğru yada teorik olarak doğru değeri arasındaki sapma

**Bozucu (Disturbance)**

Bir dizge üzerinde etkiyen, bilinmeyen yada denetlenemeyen giriş

**Karıştırma (Perturbation)**

Bir dizge üzerinde varolan durumdan geçici olarak ayrılmayla sonuçlanan bir etki yaratan giriş

**Artık (Residual)**

Ölçmeler ile model-eşitlik tabanlı hesaplamalar arasındaki sapma temelli arıza belirteci

**Belirti (Symptom)**

Gözlenebilir bir niceliğin olağan davranışındaki değişiklik

**b) İşlevler**

**Arıza Bulma (Fault Detection)**

Bir dizgede bulunan arızaları bulma ve bulma zamanını belirleme.

**Arıza Yalıtma (Fault Isolation)**

Bir arızanın tipini, konumunu ve bulma zamanını belirleme. Arıza bulmayı takip eder.

**Arıza Tanıma (Fault Identification)**

Bir arızanın büyüklüğünü ve zamana bağlı davranışını belirleme. Arıza yalıtmayı takip eder.

**Arıza Teşhisi (Fault Diagnosis)**

Bir arızanın tipini, büyüklüğünü, konumunu ve bulunma zamanı belirleme. Arıza bulmayı takip eder. Arıza yalıtmayı ve tanımayı içerir.

**İzleme (Monitoring)**

Bilgi kaydederek, davranıştaki olağandışılıkları tanıyarak ve belirterek fiziksel

bir dizgenin koşullarını sürekli olarak gerçek zamanlı belirleme görevi

### **Gözetim (Supervision)**

Fiziksel bir dizgeyi izleme ve arıza durumunda çalışmaya devam etmek için uygun hareketleri yapma.

### **Koruma (Protection)**

Eğer mümkünse dizgenin henüz gerçekleşmemiş tehlikeli davranışının bastırılması veya tehlikeli davranışın sonuçlarından kaçınma

### **c) Modeller**

#### **Nicel Model (Quantitative Model)**

Bir dizgenin davranışlarını nicel matematiksel terimler cinsinden tanımlamak amacıyla dizge değişkenleri ve parametreleri arasında statik yada dinamik ilişkilerin kullanılması

#### **Nitel Model (Qualitative Model)**

Bir dizgenin davranışlarını nedensellikler yada eğer-ise kuralları gibi nitel terimler cinsinden tanımlamak amacıyla dizge değişkenleri ve parametreleri arasında statik yada dinamik ilişkilerin kullanılması.

#### **Teşhis Modeli (Diagnostic Model)**

Özel giriş değişkenlerini-belirtileri özel çıkış değişkenlerine-arızalar bağlayan statik yada dinamik ilişkiler kümesi.

#### **Analitik Fazlalık (Analytical Redundancy)**

Bir değişkeni tanımlamak için, biri analitik şekilde bir matematiksel süreç modelini kullanırken, iki veya daha fazla yolu (benzer olmayacak şekilde) kullanma.

### **d) Dizge Özellikleri**

#### **Güvenilirlik (Reliability)**

Verilen dönem zamanı boyunca verilen alan içinde tanımlanan koşullar altında bir dizgenin istenen işlevi yerine getirebilme yeteneği.

#### **Güvenlik (Safety)**

Bir dizgenin insanlara veya araçlara veya çevresine zarar vermeme yeteneği

#### **Bulunabilirlik (Availability)**

Herhangi bir zaman anında bir dizgenin yada aracın doyum veren yada etkin olarak çalışma olasılığı.

#### **Bağımlılık (Dependability)**

Gerek duyulduğunda varolma özelliğine olan varolma şekli. Dizgenin herhangi

belirlenmiş çalışma zamanında herhangi rasgele seçilmiş zamanda gerek duyduğu işlevi yerine getirebilir ve çalışır olma derecesi. Bu özellik tanımlı dönemin başında dizgenin varolduğunu kanıtlar.

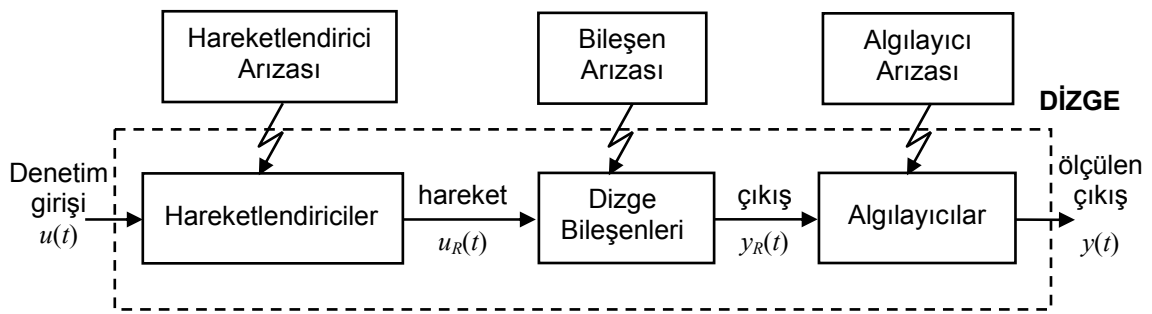
## 1.2. Arıza Tipleri

Bir arızanın tipinin tanımlanmasından önce varolan dizgenin yapısına bakılmalıdır. Bir dizgenin yapısı ise dizgede hareketi sağlayan hareketlendirici (actuator), dizgenin dinamiklerini ve iç işleyişini oluşturan bileşenler (components) ve durum değişkenlerini veya dizge çıktıları algılayıp kapalı çevrimde geribeslemeyi sağlayan algılayıcı (sensor) olmak üzere üç ana kısımdan oluşur ve her bir kısımda arıza tanımı yapılabilir (Chen ve Patton, 1999). Bunun yanında arızanın dizgeye etkisine ve zamanda gösterdiği değişime göre de tip tanımlamaları yapılabilir. Bu yaklaşımdan yola çıkarak arıza tiplerini sınıflandırmak için üç yöntem kullanılmaktadır (Chen ve Patton, 1999; Gertler, 1998):

- Oluşma yerlerine göre
- Modellenmelerine göre
- Zaman karakteristiklerine göre

### 1.2.1. Oluşma Yerlerine Göre Arıza Tipleri

Bir dizgenin yapısı ve bu yapıda oluşma yerlerine göre arızaların gösterilimi Şekil 1.1.'de gösterilmiştir.



Şekil 1.1. Oluşma yerlerine göre arızaların gösterilimi

Dizgenin doğrusal olduğu kabul ederek durum uzayı tanımlaması Eş. (1.1)'de verilmiştir :

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu_R(t) \\ y_R(t) &= Cx(t) + Du_R(t) \end{aligned} \quad (1.1)$$

Burada  $x(t) \in \mathfrak{R}^n$  durum vektörü,  $u(t) \in \mathfrak{R}^r$  denetim giriş vektörü,  $u_R(t) \in \mathfrak{R}^r$   $u(t)$  denetim girişine karşılık hareketlendirici tepke vektörü,  $y_R(t) \in \mathfrak{R}^m$  gerçek dizge çıkış vektörü,  $A, B, C$  ve  $D$  dizgenin uygun boyutlarda tanımlanmış durum matrisleridir.

### 1.2.1.1. Hareketlendirici (Actuator) Arızaları

Bu tip arızalar tümünden veya kısmi denetim hareketinin kaybı şeklinde düşünülebilir. Örneğin tamamen kaybedilmiş-takılmış (stuck) bir hareketlendirici girişine ne uygulanırsa uygulansın hiçbir hareket sağlamaz. Tümünden hareketlendirici arızası kısa devre veya yanmış kablo gibi sebeplerden oluşabilir. Kısmi hareketlendirici arızası ise istenen denetim işaretinin sadece bir kısmını (örneğin %50) verebilen arıza olarak düşünülebilir. Bu tip arıza ise hidrolik veya pnömatik sızıntı veya güç kaynağındaki gerilim düşmelerinden oluşabilir. Bu tip arıza durumlarında hareketlendirici tepkesi şu şekilde hesaplanır:

$$u_R(t) = u(t) + f_a(t) \quad (1.2)$$

Eş. (1.2)'de tanımlanan  $f_a(t)$  hareketlendirici arıza işlevi olarak tanımlanır. Her ne kadar Eş.(1.2)'de artı bir değermiş gibi görünse de etkisi eksi bir değerdir.

### 1.2.1.2. Bileşen (Component) Arızaları

Bu tip arızalar dizgenin kendi bileşenlerinde oluşan ve dizgede çeşitli koşulların değişmesiyle sonuçlanan arızalar olarak düşünülebilir. Bu tip arızalar çoğunlukla yapısal hasar sonucu dizgenin fiziksel parametrelerindeki (örn. sönüm sabiti) değişimler veya dizge dinamiklerindeki değişiklikler (örn. üç tanklı dolum dizgesindeki sızıntı) olarak ortaya çıkar. Bu tip arızalar dizgenin dinamik davranışını etkiler. Bu tip arıza durumlarında dizge modeli şu duruma gelir:

$$\dot{x}(t) = Ax(t) + Bu_R(t) + f_c(t) \quad (1.3)$$

Burada tanımlanan  $f_c(t)$  bileşen arıza işlevi olarak tanımlanır.

### 1.2.1.3. Algılayıcı (Sensor) Arızaları

Bu tip arızalar algılayıcılardan yanlış okuma şeklinde düşünülebilir. Bu tip arızalar da tümünden veya kısmi olarak düşünülebilir. Tümünden algılayıcı arızasında algılayıcı ölçülen fiziksel parametre değerinden alakasız bir değer üretir. Tümünden

algılayıcı arızası kopan teller veya yüzeyle temasın kesilmesiyle oluşabilir. Kısmi algılayıcı arızası ise ölçülen işarete bağlı olarak gerçek işaretin elde edilebileceği bir işaret üretir (*örn.* öndeğerli (biased) bir algılayıcı). Bu tip arıza durumunda model şu duruma gelir:

$$y(t) = y_R(t) + f_s(t) \quad (1.4)$$

Burada tanımlanan  $f_s(t)$  algılayıcı arıza işlevi olarak tanımlanır.

Bu üç tip arızanın oluşması olasılığı durumunda dizge durum uzayı modeli şu duruma gelir:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Bf_a(t) + f_c(t) \\ y(t) &= Cx(t) + Du(t) + Df_a(t) + f_s(t) \end{aligned} \quad (1.5)$$

Eş. (1.5)'teki tanımda arıza işlevleri vektör haline getirilip tekrar yazılırsa

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + R_1 f(t) \\ y(t) &= Cx(t) + Du(t) + R_2 f(t) \end{aligned} \quad (1.6)$$

şeklinde yazılabilir. Burada  $f(t) \in \mathfrak{R}^g$  arıza vektörüdür ve  $f_i(t)$  ( $i = 1, 2, \dots, g$ ) ile tanımlı her bir ögesi bir arıza işlevine karşılık gelmektedir.

Eş. (1.6)'daki tanım zaman düzleminde ve çoğu FDI yöntemi tarafından kullanılmaktadır (Chen ve Patton, 1999). Bunun yanında bazı FDI yöntemleri s-düzleminde işlemler yapmaktadır (Frank,1990; Frank ve Ding, 1997) ve burada bu geçişin nasıl yapıldığını vermek yararlı olacaktır. Eş. (1.6) s-düzlemine geçilerek tekrar yazılır ve giriş-çıkış geçiş matris gösterilimine dönüştürülürse

$$y(s) = G_u(s)u(s) + G_f(s)f(s) \quad (1.7)$$

olur ve  $G_u(s)$  ve  $G_f(s)$  şu şekilde tanımlanır:

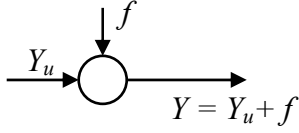
$$\begin{aligned} G_u(s) &= C(sI - A)^{-1} B + D \\ G_f(s) &= C(sI - A)^{-1} R_1 + R_2 \end{aligned} \quad (1.8)$$

### 1.2.2. Modellenmelerine Göre Arızalar

Arızaların yaptıkları etkilere göre modellenmesi ayırım olarak toplamlı ve çarpımlı olmak üzere iki tip arızayı ortaya çıkarmıştır (Isermann, 2005).

### 1.2.2.1. Toplamlı (Additive) Arızalar

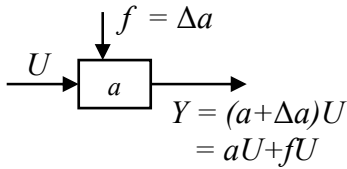
Bu tip arızalar bir  $Y_u$  değerine bir ek  $f$  değeri olarak gözlemlenir (*örn.* öndeğerli bir algılayıcı). Bu durum Şekil 1.2’de gösterilmiştir.



Şekil 1.2. Toplamlı arıza gösterilimi

### 1.2.2.2. Çarpımlı (Multiplicative) Arızalar

Bu tip arızalar bir  $U$  değerinin bir  $f$  değeriyle çarpımı (*örn.* dizge parametresindeki değişimler) olarak gözlemlenir. Bu durum Şekil 1.3’de gösterilmiştir.



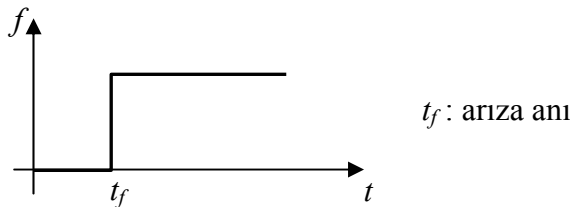
Şekil 1.3. Çarpımlı arıza gösterilimi

### 1.2.3. Zaman Karakteristiklerine Göre Arızalar

Bir dizgede oluşabilecek arızalar zaman içinde gösterdikleri davranışa göre ani, artan ve geçici olmak üzere üç tipe ayrılırlar (Isermann, 2005).

#### 1.2.3.1. Ani (Abrupt) Arızalar

Bu tip arızalar zaman düzleminde aniden ortaya çıkan ve bir kez oluştuğundan sonra sürekli olan arızalardır. Çoğunlukla donanımsal bir arızanın sonucu olarak ortaya çıkarlar. Bu tip arızaların zaman içindeki davranışı Şekil 1.4’te gösterilmiştir.

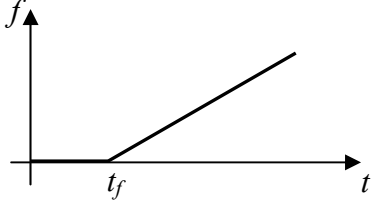


Şekil 1.4. Ani arızaların zaman içindeki davranışı

#### 1.2.3.2. Artan (Incipient) Arızalar

Bu tip arızalar belirli bir zaman anında başlayıp zaman içinde yavaş yavaş

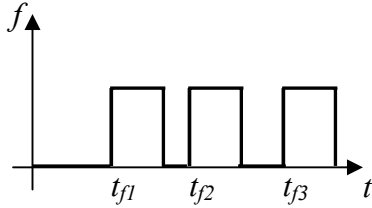
büyüyen veya artan arızalardır ve özellikle kimyasal süreçlerde ortaya çıkarlar. Zaman içinde süreklilik gösteren bu tip arızaların zaman içindeki davranışı Şekil 1.5'te gösterilmiştir.



Şekil 1.5. Artan arızaların zaman içindeki davranışı

### 1.2.3.3. Geçici (Intermittent) Arızalar

Bu tip arızalar zaman içinde aniden ortaya çıkıp kaybolan yani zaman içinde kesikli olarak oluşan ve periyodik olmayan ani tipte arızalar olarak kabul edilir. Bu tip arızaların zaman içindeki davranışı Şekil 1.6'da gösterilmiştir.



Şekil 1.6. Geçici arızaların zaman içindeki davranışı

## 1.3. Model Temelli Arıza Bulma ve Yalıtım Yöntemleri

Arıza bulma ve yalıtımı ile ilgili çalışmalar ve yöntemler model temelli ve veri temelli olmak üzere ikiye ayrılmaktadır (Chen ve Patton, 1999). Veri temelli yöntemler varolan dizgenin giriş-çıkış işaretlerini kaydederek bu verilerin işlenmesini temel alır ve Temel Bileşen Analizi-TBA (Principal Component Analysis/PCA), *CUSUM* testi gibi çokdeğişkenli istatistiksel yöntemler üzerine yoğunlaşmıştır. Veri temelli yöntemler bu tez çalışmasının konusu olmadığından ayrıntılı bilgiye yer verilmemiştir ve konu ile ilgili ayrıntılı tarama ve örnekler Venkatasubramanian ve arkadaşlarının gözden geçirme makalesinde ve Patton ve arkadaşlarının kitabında bulunabilir (Venkatasubramanian ve ark., 2003; Patton ve ark., 2000).

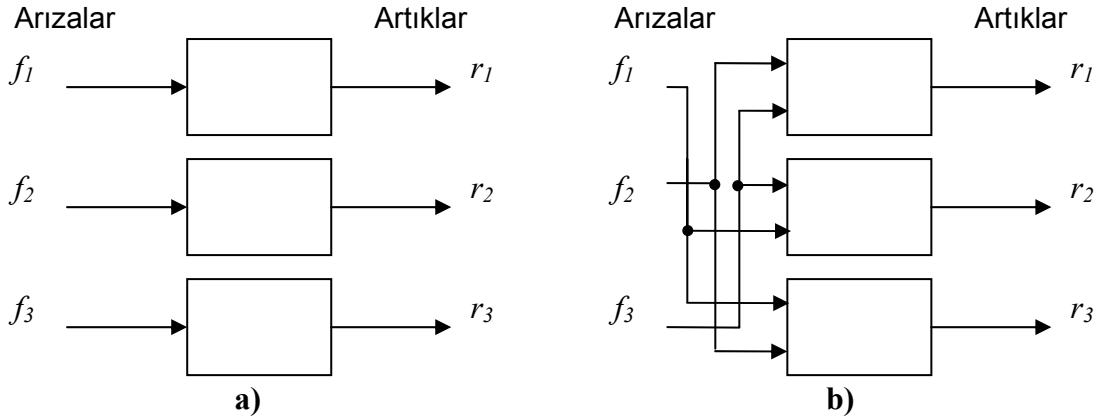
Model temelli yöntemler ise varolan dizgenin modellenmesi ve model ile gerçek dizge arasındaki fark işaretlerini işlemeyi esas alır. Bu anlamda Chen ve Patton model temelli arıza teşhisini (bulma ve yalıtım) dizgenin matematiksel modeliyle tanımlanan



### 1.3.1. Gözlemci (Observer) Temelli Yaklaşımlar

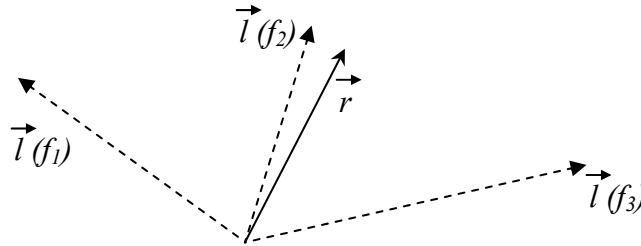
Gözlemci veya süzgeç temelli yaklaşımların altında yatan düşünce, ölçümlerden veya ölçümlerin bir altkümesinden ya deterministik ayarlamalarla Luenberger gözlemcilerini veya rasgelelik (stokastik) ayarlamalarla Kalman süzgeçlerini kullanarak dizgenin çıkışlarını kestirmektir (Chen ve Patton, 1999; Frank ve Ding, 1997; Garcia ve Frank, 1996; Magni ve Mouyon, 1994; Patton ve Chen, 1997). Bu kestirimden yararlanarak oluşturulan ağırlıklandırılmış kestirim hatası artık türetimi için kullanılır.

Gözlemci temelli yöntemler arıza yalıtımı için yapısal artık kümelerini veya yönlü artık vektör tasarımlarını kullanabilir. Yapısal artık kümelerinde artıkların oluşturduğu kümedeki her bir artık belirli bir arızaya duyarlı, diğerlerine duyarsızdır. Her bir artığın bir arızaya duyarlı olduğu artık kümelerini kullanan gözlemciler atanmış (dedicated) gözlemciler, her bir artığın biri hariç diğer tüm durumlara duyarlı olduğu artık kümelerini kullanan gözlemciler de genelleştirilmiş (generalized) gözlemciler olarak adlandırılır. Bu tip artık kümeleri Şekil 1.8’de gösterilmiştir.



Şekil 1.8. Yapısal artık kümeleri a) Atanmış b) Genelleştirilmiş

Yönlü artık vektörlerinde ise artıkların oluşturduğu  $n$  boyutlu ( $n$ : artık sayısı) vektörün arızalı durumlarda oluşan artık vektörlerine izdüşümlerinin büyüklüğüne bakılarak arıza yalıtımı yapılır. Bu tip vektörlerin gösterilimi Şekil 1.9’da verilmiştir.



Şekil 1.9. Yönlü artık vektörü gösterilimi

### 1.3.2. Eşlik Vektörü-İlişkisi (Parity Vector-Relation) Yaklaşımları

Eşlik vektörü yaklaşımları aynı değişkeni birden fazla yolla tanımlamak olarak da nitelendirilen fazlalık tanımıyla eşleştirilebilir. Bu tanıma bağlı olarak aynı değişkeni ölçmek için kullanılan birden fazla algılayıcı kullanan donanımsal fazlalıklardan (hardware redundancy) ve bir değişkeni matematiksel olarak tanımlamada kullanılan yazılımsal fazlalıklardan (software redundancy) veya geçici fazlalıklardan (temporal redundancy) söz edilebilir. Eşlik vektörleri bu denk ölçümlerden elde edilen sonuçların birbiri ile eşliliğini (tutarlılığını) denetler (Chow ve Willsky, 1984; Chen ve Patton, 1999; Patton ve ark., 2000).

### 1.3.3. Parametre Kestirim (Parameter Estimation) Yaklaşımları

Dizgedeki parametrelerin değerlerini bulmada veya bulmadan dizgeyi modellemede kullanılan dizge tanısı (system identification) yöntemleri (Auto Regressive Moving Average-ARMA, Auto Regressive Moving Average with Exogenous Outputs-ARMAX vb.) model temelli FDI için de kullanılabilir (Simani ve ark., 2002; Liu ve ark., 2000; Moseler ve Isermann, 2000). Bu yaklaşımlar arızaların sürtünme, ağırlık, akışkanlık, direnç, endüktans, sığa vb. fiziksel dizge parametrelerine yansıdığı kabulüne dayanmaktadır. Bu yaklaşımların temel ilkesi varolan sürecin parametrelerinin süreç devam ederken iyi bilinen parametre kestirim yöntemleri ile tekrar edecek şekilde kestirilmesi ve sonuçların arıza olmayan durumda daha önceden elde edilmiş sağlıklı model parametreleri ile karşılaştırılmasıdır. Herhangi bir andaki farklılık arıza olarak algılanır.

### 1.3.4. Gürbüz Artık Türetim Sorunu

Arıza teşhis yöntemlerinin güvenilirliğinin izlenen dizgeden daha yüksek olma zorunluluğu gürbüzlük sorununu ortaya çıkarmıştır. Model temelli FDI, gözlenen dizgenin matematiksel modelinin kullanımına dayanır. Dizgenin dinamik davranışını tanımlamak için daha iyi bir model kullanıldığında arızaların teşhisindeki güvenilirliğin ve başarımın geliştirilmesi için daha çok şans ortaya çıkacaktır. Yine de modelleme hataları-belirsizlikler (modelling errors-uncertainties) ve bozucular (disturbances) karmaşık mühendislik dizgelerinde kaçınılmazdır ve bu yüzden gürbüz FDI algoritmaları geliştirmek gereklidir. Bir arıza teşhis dizgesinin gürbüzlüğü, model-gerçeklik farklarının (örn. parametre değişimleri, turbülans vs.) varlığında bile sadece

arızalara duyarlı olmasıdır. Çoğunlukla parametre değişimleri ve bozucular gerçek bir süreç üzerinde kesin olmayan bir yoldan etki eder, bu yüzden arızalara çok duyarlı iken belirsizliklere ve modellenmemiş bozuculara duyarsız bir arıza yalıtım şeması tasarlamak zor olabilir.

Model temelli arıza teşhisinin temeli artıkların türetilmesidir. Arızalar ve belirsizlikler artığı etkilediğinden bunların etkilerini ayırmak zordur. Bu yüzden gürbüz bir FDI tasarımında görev, belirsizliklere duyarsız, aynı zamanda arızalara duyarlı yani gürbüz artıklar türetmektir.

Çözüme pratikte oluşabilecek tüm tipte modelleme belirsizliklerini ve etkilerini içeren izlenen dizgenin matematiksel bir tanımıyla başlanması gerekmektedir. Bu yüzden dizgenin Eş. (1.1) ile tanımlanmış durum uzay modeli şu şekilde verilir :

$$\begin{aligned} \dot{x}(t) &= (A + \Delta A)x(t) + (B + \Delta B)u(t) + E_1 d(t) + R_1 f(t) \\ y(t) &= (C + \Delta C)x(t) + (D + \Delta D)u(t) + E_2 d(t) + R_2 f(t) \end{aligned} \quad (1.9)$$

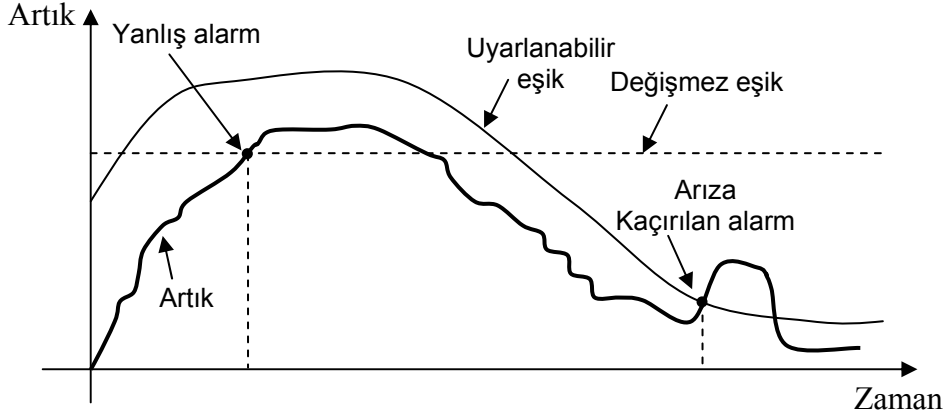
Burada  $d(t) \in R^q$  bilinmeyen giriş (bozucu) vektörü iken  $E_1$  ve  $E_2$  bilinmeyen giriş dağılım matrisleri bilinmek zorundadır.  $\Delta A$ ,  $\Delta B$ ,  $\Delta C$  ve  $\Delta D$  modelleme hatalarını temsil eden parametre hataları veya değişimleridir. Dizgenin Eş. (1.7) ile tanımlanan geçiş işlevi tanımı ise şu hale gelir :

$$y(s) = (G_u(s) + \Delta G_u(s))u(s) + G_d(s)d(s) + G_f(s)f(s) \quad (1.10)$$

Burada  $G_d(s)d(s)$  bozucu etkisidir ve  $G_d(s) = E_2 + C(sI - A)^{-1}E_1$  ile tanımlanır.  $\Delta G_u(s)$  modelleme hatalarını tanımlamak için kullanılmıştır.  $G_d(s)d(s)$  ve  $\Delta G_u(s)u(s)$  birlikte modelleme belirsizliklerini temsil etmektedir.

Artık üretiminde gürbüzlük için bozucuları ve modelleme hatalarını bilinmeyen girişler olarak kabul eden Bilinmeyen Giriş Gözlemcileri (Unknown Input Observer-UIO) ve dizgeye ait özdeğer ve özvektörlerin belirli bir yapıya getirilmesiyle bozuculara ve modelleme hatalarına duyarsız hale getiren Özyapı Ataması (Eigenstructure Assignment-EA) gibi yaklaşımlar bulunmaktadır (Chen ve Patton, 1999). Ayrıca pratik durumda türetilen artığın hiçbir zaman sıfır olmayacağı gözönünde bulundurulmalıdır. Bu durumda artık değerlendirme aşamasında bir eşik kullanılır. Doğal olarak eşik, arıza olmayan durumdaki en geniş artık genliğinden biraz fazlasına ayarlanır. Eşiğin değişmez olarak ayarlanması gürbüzlüğe olumsuz şekilde etki eden önemli bir etmendir ve özellikle çok geniş çalışma aralığı olan dizgelerde yanlış ve kaçırılan alarm

oranlarının yükselmesine yol açar. Bu sebeple artıklarda gürbüzlüğü sağlamanın diğer bir yolu uyarlanabilir eşik kullanmaktır. Uyarlanabilir eşik nasıl uygulandığı ve getirdiği yarar Şekil 1.10'da verilmiştir. Uyarlanabilir eşik kullanımındaki sorun eşik kuralının işlevsel durumunun nasıl belirleneceğidir (Chen ve Patton, 1999).



**Şekil 1.10.** Uyarlanabilir eşik uygulanması

### 1.3.5. Doğrusal Olmayan Dizgeler İçin Arıza Bulma ve Yalıtımı

Tez çalışması robot manipülatörlerdeki arızaları konu olarak ele aldığından ve robot dizgelerinin doğrusal olmayan yapıda olmalarından dolayı doğrusal olmayan dizgeler için arıza bulma-yalıtımı konusu ayrıntılı biçimde bu altbölümde incelenmiştir.

Model temelli arıza teşhis yöntemlerinin çoğu doğrusal dizge modelleri üzerine kurulmuştur. Doğrusal olmayan dizgeler için arıza teşhisi sorununa genel olarak iki şekilde yaklaşılır (Chen ve Patton, 1999). Birincisinde model, bir çalışma noktasında doğrusallaştırılır ve çalışma noktasının çok küçük komşuluğunda parametre değişmelerine duyarsız olacak şekilde artık işaretleri türetmek için gürbüz teknikler uygulanır. Bu yaklaşım sadece doğrusallaştırma sonucunda doğrusal ve doğrusal olmayan modeller arasında çok büyük farkların olmadığı durumlarda iyi sonuçlar vermektedir. Fakat çok yüksek doğrusal olmamaların ve çok geniş çalışma aralığının bulunduğu dizgeler için doğrusallaştırma yaklaşımı doyurucu sonuçlar vermekten uzaktır. Buna bir çözüm, çalışma nokta aralığına karşılık gelecek şekilde çok sayıda doğrusallaştırılmış model kullanmaktır. Bu ise tüm çalışma noktalarına karşılık bir FDI şeması getirecektir. Bu da gerçek zamanlı uygulamalarda uygulanabilir olmayacaktır.

Yukarıda anlatılan sebeplerden dolayı doğrusal olmayan dizge modelleriyle doğrudan uğraşabilecek FDI yöntemlerinin geliştirilmesi gerekmektedir. Bu amaçla

daha önce doğrusal dizgeler için kullanılmış çoğu yöntem doğrusal olmayan dizgelere uyarlanmaya çalışılmıştır. Doğrusal olmayan dizgelerde FDI için doğrusal olmayan gözlemcileri kullanmak en genel yaklaşımdır (Chen ve Patton, 1999; Frank ve Ding, 1997; Patton ve ark., 2000; Adjallah ve ark., 1994; Garcia ve Frank, 1997). Doğrusal olmayan bir dizgenin modeli aşağıdaki gibi verilir :

$$\begin{aligned}\dot{x}(t) &= g(x(t), u(t), f(t), d(t)) \\ y(t) &= h(x(t), u(t), f(t), d(t))\end{aligned}\quad (1.11)$$

Burada  $x(t)$  durum vektörü,  $y(t)$  çıkış vektörü,  $u(t)$  giriş vektörü,  $f(t)$  arıza vektörü,  $d(t)$  bozucu vektörü ve  $g(\cdot, \cdot, \cdot, \cdot)$  ve  $h(\cdot, \cdot, \cdot, \cdot)$  doğrusal olmayan işlevlerdir. FDI sorunu aşağıdaki gözlemci yapısını kullanarak  $r(t)$  artık vektörünü türetmektir :

$$\begin{aligned}\dot{\xi}(t) &= g_r(\xi(t), u(t), y(t)) \\ r(t) &= h_r(\xi(t), u(t), y(t))\end{aligned}\quad (1.12)$$

Burada da  $\xi(t)$  durum kestirimcisini temsil etmektedir ve artık şu koşulu sağlamalıdır :

$$\|r(t)\| \begin{cases} \approx 0 & f(t) = 0 \\ \gg 0 & f(t) \neq 0 \end{cases}\quad (1.13)$$

Burada amaçlanan  $g_r(\cdot, \cdot, \cdot)$  ve  $h_r(\cdot, \cdot, \cdot)$  işlevlerini tasarlamaktır. Literatürde özellikle belirli tipte dizgeler için tanımlanmış doğrusal olmayan gözlemci temelli çeşitli tasarım yaklaşımları bulunmaktadır. Bu yaklaşımlar izleyen altbölümlerde verilmiştir.

### 1.3.5.1. Doğrusal Olmayan Benzerlik Gözlemcisi (Nonlinear Identity Observer-NIO) Yaklaşımı

Bu yaklaşımda doğrusal olmayan dizgeye ait model aşağıdaki gibi tanımlanmış ve kabul edilmiştir (Adjallah ve ark., 1994; Garcia ve Frank, 1997):

$$\begin{aligned}\dot{x}(t) &= g(x(t), u(t)) + R_1 f(t) \\ y(t) &= h(x(t), u(t)) + R_2 f(t)\end{aligned}\quad (1.14)$$

Bu model için Eş. (1.14)'teki durum değişkenlerini, çıkışın kestirimlerini ve artık işlevini veren doğrusal olmayan benzerlik gözlemcisi şu şekilde tasarlanabilir :

$$\begin{aligned}
\dot{\hat{x}}(t) &= g(\hat{x}(t), u(t)) + K(\hat{x}(t), u(t))[y(t) - \hat{y}(t)] \\
\hat{y}(t) &= h(\hat{x}(t), u(t)) \\
r(t) &= y(t) - \hat{y}(t)
\end{aligned} \tag{1.15}$$

Eş. (1.14) ve (1.15)'ten yararlanarak  $r(t)$  ve durum kestirim hatası  $e(t) = x(t) - \hat{x}(t)$  tanımlanırsa :

$$\begin{aligned}
\dot{e}(t) &= F(t)e(t) + O_1(e^2(t), t) + R_1 f(t) - K(\hat{x}(t), u(t))R_2 f(t) \\
r(t) &= H(t)e(t) + O_2(e^2(t), t) + R_2 f(t)
\end{aligned} \tag{1.16}$$

olur. Burada  $O_1(e^2(t), t)$  ve  $O_2(e^2(t), t)$   $e(t)$ 'nin ikinci ve daha yüksek dereceden terimlerini ifade etmektedir ve

$$\begin{aligned}
F(t) &= \frac{\partial g(\hat{x}(t), u(t))}{\partial \hat{x}(t)} - K(\hat{x}(t), u(t))H(t) \\
H(t) &= \frac{\partial h(\hat{x}(t), u(t))}{\partial \hat{x}(t)}
\end{aligned} \tag{1.17}$$

şeklinde tanımlanır. Eş. (1.17) incelendiğinde eğer  $e(t)$  durum kestirim hatası asimptotik olarak sifıra yakınsarsa, sadece  $f(t)$  arıza vektöründen etkilendiği görülmektedir. Geriye kalan problem  $e(t) = 0$  bir asimptotik kararlı denge noktası olacak şekilde bir  $K(\hat{x}(t), u(t))$  matrisi tasarlamaktır.

### 1.3.5.2. Doğrusal Olmayan Bilinmeyen Giriş Gözlemcisi (Nonlinear Unknown Input Observer-NUIO) Yaklaşımı

Doğrusal dizgeler için oluşturulan bilinmeyen giriş gözlemcisi yaklaşımı doğrusal olmayan dizgelere de uyarlanmıştır. Bu yaklaşımın uygulanabildiği dizgeler aşağıdaki gibi tanımlanmıştır (Adjallah ve ark., 1994; Garcia ve Frank, 1997):

$$\begin{aligned}
\dot{x}(t) &= Ax(t) + B(y(t), u(t)) + E_1 d(t) + R_1 f(t) \\
y(t) &= Cx(t) + E_2 d(t) + R_2 f(t)
\end{aligned} \tag{1.18}$$

Burada  $f(t)$  arıza vektörü ve  $d(t)$  bilinmeyen giriş vektörüdür. Burada  $B(y(t), u(t))$  doğrusal olmayan teriminin sadece  $y$  ve  $u$ 'ya bağımlı olduğuna dikkat edilmelidir. Belirtilen dizge modeli için aşağıdaki gibi bir gözlemci ile doğrusal olmamalarla başa çıkılabilir :

$$\begin{aligned}
\dot{\xi}(t) &= F\xi(t) + J(y(t), u(t)) + Gy(t) \\
r(t) &= L_1 \xi(t) + L_2 y(t)
\end{aligned} \tag{1.19}$$

Eş. (1.19) ile verilen gözlemcinin bilinmeyen girişlere gürbüz ve arızalara duyarlı olması için gerekli koşullar Adjallah ve arkadaşlarının ve Garcia ve Frank'ın gözden geçirme makalelerinde verilmiştir (Adjallah ve ark., 1994; Garcia ve Frank, 1997). Bu koşullar sağlanırsa  $r(t)$  artığı ve  $e(t) = \xi(t) - Tx(t)$  kestirim hatası şu şekilde tanımlanır :

$$\begin{aligned} \dot{e}(t) &= Fe(t) + GR_2 f(t) - TR_1 f(t) \\ r(t) &= L_1 e(t) + L_2 R_2 f(t) \end{aligned} \quad (1.20)$$

Bu yaklaşımın en büyük eksikliği sadece Eş. (1.18) tipinde tanımlanabilen dizgeler için geçerli olması ve bu tipte tanımlanabilecek dizgelerin sınırlı olmasıdır.

### 1.3.5.3. Bozucu Ayırıcı Doğrusal Olmayan Gözlemci (Disturbance Decoupled Nonlinear Observer-DDNO) Yaklaşımı

Doğrusal olmayan dizgelerde FDI için başka bir yaklaşım daha geniş bir dizge sınıfını kapsayacak şekilde oluşturulmuştur (Seliger ve Frank, 1991). Bu yaklaşım Eş. (1.21) ile tanımlanabilen dizgelere uygulanabilir:

$$\begin{aligned} \dot{x}(t) &= A(x) + B(x)u + E(x)d + R(x)f \\ y(t) &= Cx(t) \end{aligned} \quad (1.21)$$

Burada amaçlanan ve tasarlanması gereken  $\xi = T(x)$  şeklinde modelin bozucusuz kısmından bozuculu kısmını ayıracak bir doğrusal olmayan dönüşüm bulmaktır. Bu ayırım sadece ve sadece Eş. (1.22)'de verilen koşul gerçekleşiyorsa başarılabilir:

$$\frac{\partial T(x)}{\partial x} E(x) = 0 \quad (1.22)$$

Bu eşitlik  $\xi = T(x)$  ile aynı anda çözülmesi gereken 1. derece doğrusal diferansiyel eşitlikler takımı oluşturmaktadır. Eğer Eş. (1.22)'nin  $\xi = T(x)$  şeklinde çözümleri ve  $x = \Psi(\xi, y^*)$  şeklinde bir ilişkinin var olduğunu kabul edilirse model şu şekilde tekrar yazılabilir :

$$\dot{\xi} = \frac{\partial T(x)}{\partial x} (A(x) + B(x)u + R(x)f) \Big|_{x=\Psi(\xi, y^*)} \quad (1.23)$$

Burada  $y^* = C^*(y)$ ,  $y = C(x)$  ile tanımlı varolan ölçümlerin Eş. (1.24) koşulunu sağlayan bir alt kümesidir:

$$\text{boyut}(y^*) < \text{boyut}(y) \quad (1.24)$$

Ayrıca  $Q(T(x), C(x))=0$  şeklinde bir durumun varolduğu kabul edilirse  $x$  durumunun  $\xi$  bozucusuz kısmını kestirmek için doğrusal olmayan bir gözlemci şu şekilde düzenlenebilir :

$$\dot{\hat{\xi}} = \frac{\partial T(\hat{x})}{\partial \hat{x}} (A(\hat{x}) + B(\hat{x}) + K(\hat{\xi}, y, u)Q(\hat{\xi}, y)) \Big|_{x=\Psi(\hat{\xi}, y^*)} \quad (1.25)$$

Burada  $K(\hat{\xi}, y, u)$  geribesleme matrisi ile sağlanan tasarım serbestliği  $e = \hat{\xi} - \xi$  ile tanımlı kestirim hata dinamiğini oluşturan diferansiyel eşitliği kararlı duruma getirmek için kullanılabilir. Bu durumda artık ise şu şekilde tanımlanır:

$$r = Q(\hat{\xi}, y) = Q(\xi + e, y) \quad (1.26)$$

$e$  kestirim hatası ise Eş. (1.26) ile tanımlı diferansiyel eşitliklerle ilişkilidir :

$$\dot{e} = \rho(e, t) - \frac{\partial T(x)}{\partial x} R(x) f \quad (1.27)$$

Burada  $\dot{e} = \rho(e, t)$  doğrusal olmayan diferansiyel eşitliklerii  $e = 0$ 'da yerel asimptotik kararlı olacak şekilde tasarlanmalıdır. Böylece arızasız durumda artık sifıra yakınsayacaktır. Bozucu ayırıcı doğrusal olmayan gözlemcilerle ilgili daha ayrıntılı bir çözümlene ve tank dizgesi içeren bir örnek Seliger ve Frank'ın çalışmasında bulunabilir (Seliger ve Frank, 1991).

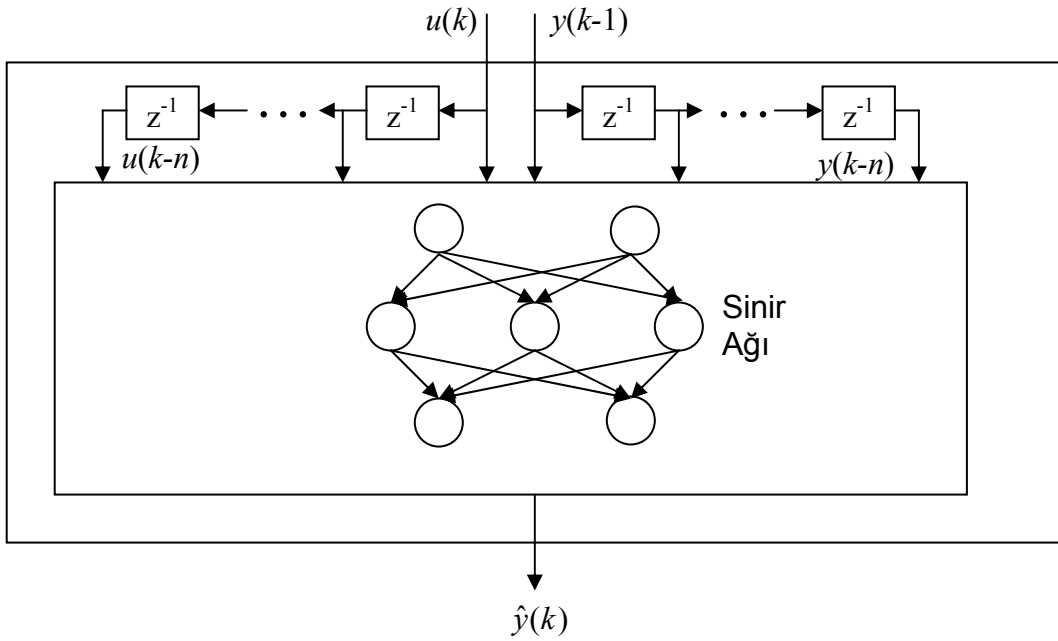
Doğrusal olmayan dizgelerin bir sınıfı olan ve özellikle kimyasal süreçler, hidrolik sürüş ve ısı aktarım dizgeleri olarak karşılaşılan bilineer dizgeler için de literatürde çalışmalar bulunmaktadır (Chen ve Patton, 1999). Bu tip dizgeler tez konusu kapsamı dışında olduğundan burada yer verilmemiştir.

#### 1.3.5.4. Sinir Ağı Gözlemcisi (Neural Network Observer) Yaklaşımı ve Sinir Ağı ile Artık Değerlendirme

Bir önceki altbölümlerde anlatılan doğrusal olmayan gözlemcilerin üzerine kurulduğu analitik modelleri pratikte elde etmek çok kolay değildir. Model olmadan da gözlemci temelli FDI imkansızdır. Bu sorunu aşmak için doğrusal olmayan dizgeleri tanımlayabilecek “evrensel” yaklaşım modelleri bulmak tercih edilir. Bu evrensel modeli tanımlamak için de otomatik bir düzeneğin olması şarttır. Sinir Ağları (SA)

doğrusal olmayan haritalama yapabilme özellikleri sayesinde çok giriş-çok çıkışlı doğrusal olmayan bir dizgeyi modellemek için kullanılabilir fakat ileri beslemeli SA durağan doğrusal olmayan haritalama yaptıklarından doğrusal olmayan bir dizgenin sadece giriş değerlerini aldıklarında çıkış değerleri tanımlamada kullanılamazlar (Haykin, 1999). Bu sebeple sinir ağı giriş olarak dizgenin o anki girişlerini ve daha önceki anlara ait çıkışlarını almalıdır. Ayrıca sinir ağlarının özellikleri incelendiğinde model kurmada gerekli olan gürültüye ve belirsizliklere gürbüzlüğün kendi doğasında var olduğu evrensel yaklaşımcı kanıtlarında açıkça görülmektedir (Haykin, 1999).  $W$  ağırlık matrisli bir ileri beslemeli ağı yapısı Eş. (1.28)'de ve doğrusal olmayan bir dizgenin sinir ağı modeli Şekil 1.11'de verilmiştir.

$$\hat{y}(k) = SA(W, y(k-1), \dots, y(k-n), u(k), \dots, u(k-n)) \quad (1.28)$$

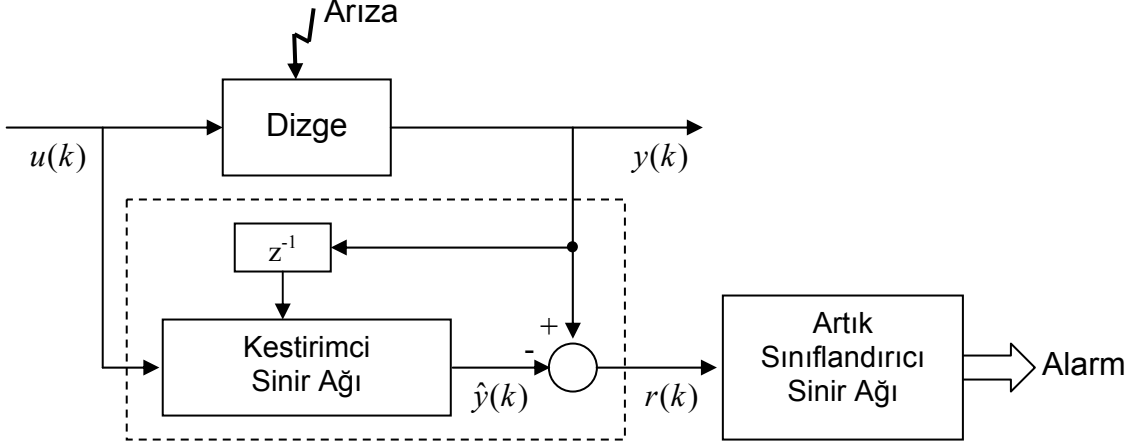


**Şekil 1.11.** Doğrusal olmayan bir dizgenin sinir ağı modeli

Eğitimden sonra bir sinir ağı verilen girişe karşılık çıkışın çok kesin bir kestirim değerini verebilir. Ayrıca SA kendilerinden gelen doğrusal olmayan haritalama özellikleri ve giriş verilerinin doğrusal olmayan yüzeylerle birbirlerinden ayrılması ile verileri sınıflandırma için de kullanılmaktadır (Haykin, 1999).

Bu özellikleri ile SA bir FDI şemasında gerçek dizge çıkışları ve sinir ağı tarafından kestirilmiş çıkışların karşılaştırılması ile artık türetimi işlevini yerine getirilebilir. İkinci bir sinir ağı ile artıkların özellikleri değerlendirilerek (sınıflandırma)

herhangi analitik bir yöntem kullanılmadan arıza yalıtım işlemi gerçekleştirilebilir. Sinir ağlarının kestirim ve sınıflandırma için kullanıldığı model temelli FDI şeması Şekil 1.12’de verilmiştir.



**Şekil 1.12.** İki sinir ağı kullanarak arıza bulma ve yalıtımı (Chen ve Patton, 1999)

Verilen özellikleriyle çeşitli tiplerde SA, FDI ile ilgili çalışmalarda çeşitli uygulamalar için hem doğrusal, hem de doğrusal olmayan dizgelerde artık türetme/arıza bulma, artık değerlendirme/arıza yalıtma amacıyla kullanılmıştır (Patton ve ark., 2000; Lehtoranta ve Koivo, 2005; Marcu ve ark, 1998; Pei ve Chowdhury, 1999; Karpenko ve ark., 2001; Ayoubi, 1994; Altuğ ve ark, 1999; Liu ve ark, 2000).

Yukarıda anlatılan yaklaşımların haricinde online yaklaşım olarak adlandırılan ve SA ve uyarlanabilir denetim için tanımlı uyarlanabilir öğrenmeyi kullanan, sinir ağına ait parametrelerin giriş işaretleri ve arıza işlevleri cinsinden tanımlandığı çalışmalar da bulunmaktadır (Zhang ve ark, 2002; Polycarpou ve Helmicki, 1995; Polycarpou ve Trunov, 2002).

### 1.3.5.5. Bulanık Gözlemciler (Fuzzy Observers) Yaklaşımı ve Bulanık Mantık ile Artık Değerlendirme

Sinir ağlarının kendi kendine öğrenen ve içi bilinmez yapısı (siyah kutu) her ne kadar FDI için uygunmuş gibi görünse de kullanıcının tecrübesini, çıkarımlarını ve gerektiğinde karışmasını uzak tutması istenen bir özellik değildir. Kullanıcıyı da model tasarımına katmak ve bu zayıflıkların üstesinden gelmek amacıyla Bulanık Mantık (BM) doğrusal olmayan dizgelerde FDI için modelleyici, gözlemci ve uyarlanabilir artık seçici olarak kullanılmıştır (Dexter, 1995; Patton ve ark., 1998; Isermann, 1998; Ballé,

1998). Bulanık gözlemci yaklaşımının temelinde yatan düşünce Takagi-Sugeno bulanık modelini kullanarak doğrusal olmayan bir dizgeyi bir dizi EĞER-İSE kuralları ile yerel olarak doğrusallaştırılmış gözlemcilerden yararlanarak tanımlamaktır. Yerel bölge tanımları ise çalışma noktasından yararlanılarak çıkarılır.

Belirli bir çalışma noktasında dizgeyi tanımlayacak kural şu şekildedir:

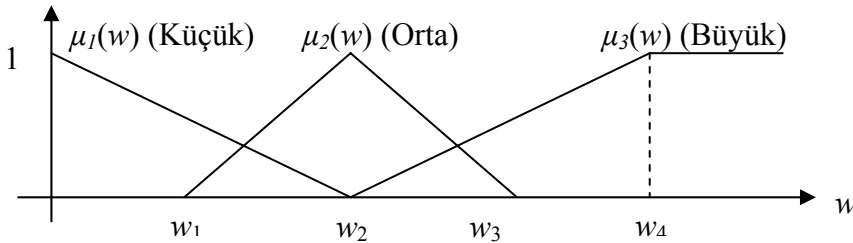
Kural  $i$  : ( $i = 1, 2, \dots, N$ )

$$\text{Eğer } w(t) = M_i \text{ İse } \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) + D_i u(t) \end{cases} \quad (1.29)$$

Burada  $x(t) \in \mathfrak{R}^n, u(t) \in \mathfrak{R}^r, y(t) \in \mathfrak{R}^m$  ve  $A_i, B_i, C_i, D_i$  uygun boyuttaki durum matrisleridir.  $w(t)$  vektörü ise çalışma noktasını tanımlamada kullanılan (örn. bir DA motoru için açılmal hız) öncül (premise) değişkendir.  $M_i$  bulanık kümeyi tanımlarken  $N$  EĞER-İSE kurallarının sayısını belirtmektedir. Bulanık mantıkta her bir modelin belirli bir üyelik işlevi vardır ve bu üyelik işlevlerine bağlı olarak çıkış hesaplanır (Jang ve ark, 1998). Sonuçta kurallara ve o andaki  $u(t)$  girişine bağlı olarak çıkış, şu şekilde yorumlanır :

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^N \mu_i(w(t)) [A_i x(t) + B_i u(t)] \\ y(t) &= \sum_{i=1}^N \mu_i(w(t)) [C_i x(t) + D_i u(t)] \end{aligned} \quad (1.30)$$

Burada  $\mu_i(w(t))$  ise üyelik değeri  $0 \leq \mu_i \leq 1$  olarak tanımlanır ve Şekil 1.13'te üyelik işlevlerine ait bir örnek verilmiştir.



**Şekil 1.13.** Üyelik işlevi gösterilimi

Eş. (1.30)'daki model tanımına benzer şekilde yerel bölgeler için tanımlanan gözlemcilerden yararlanılarak bulanık gözlemciler oluşturulur :

Kural  $i$  : ( $i = 1, 2, \dots, N$ )

$$\text{Eğer } w(t) = M_i \text{ İse } \begin{cases} \dot{\hat{x}}(t) = A_i \hat{x}(t) + B_i u(t) + K_i [y(t) - \hat{y}(t)] \\ y(t) = C_i \hat{x}(t) + D_i u(t) \end{cases} \quad (1.31)$$

ve bunların ağırlıklı ortalamasından dizgeye ait durum denklemleri

$$\begin{aligned} \dot{\hat{x}}(t) &= \sum_{i=1}^N \mu_i(w(t)) [A_i \hat{x}(t) + B_i u(t) + K_i (y(t) - \hat{y}(t))] \\ y(t) &= \sum_{i=1}^N \mu_i(w(t)) [C_i \hat{x}(t) + D_i u(t)] \end{aligned} \quad (1.32)$$

olarak elde edilir. Elde edilen gözlemci ve gerçek dizge arasındaki farktan ( $r(t) = y(t) - \hat{y}(t)$ ) yararlanılarak artık türetimi gerçekleştirilir.

Bulanık mantık ayrıca artık değerlendirme için de kullanılmıştır (Schneider ve Frank, 1996). Artık değerlendirmedeki en büyük sorun eşik değerini ayarlamaktır. Uyarlanabilir bir eşik, değişmez eşiğin oluşturabileceği yanlış alarmları yok edebileceği Şekil 1.10 ve Bölüm 1.3.4'te açıklanmıştır. Bulanık mantık da bu uyarlanabilir eşiğin seçimi için kullanılabilir. Bu kullanımın genel gösterilimi Eş. (1.33)'te verilmiştir :

$$J(u, y) = J_0 + \Delta J(u, y) \quad (1.33)$$

Bu kullanımda değişmez bir eşiğe ( $J_0$ ) eklenecek olan değer ( $\Delta J(u, y)$ ), dizge giriş ve çıkışlarının bir bulanık mantık eşik seçiciye verilmesiyle elde edilir.

#### 1.4. Arızaya Dayanıklı Denetim

Arızaya dayanıklı denetim dizgeleri (Fault Tolerant Control Systems-FTCS) bir dizgede oluşan arızayı/arızaları otomatik olarak çözme yeteneğine ve arıza durumlarında tüm dizgenin kararlılık ve kabul edilebilir başarımlı özelliklerini sürdürebilme yeteneği olan dizgeler olarak adlandırılır (Kanev, 2004; Zhang ve Jiang, 2008).

Yukarıda verilen tanıma bağlı olarak dizgelerde arızalarla başa çıkmak için iki yaklaşım kullanılır. Birinci olarak dizgeye ait arıza gösterebilecek tüm bileşenlerin fazlaları (redundant) kullanılır (*örn.* hareketlendirici arızası anında devreye giren ve varolan dişliye bağlanabilen ikinci bir hareketlendirici kullanma veya bir ölçümü eş iki tane algılayıcı kullanarak yapma). İkinci olarak arızalara karşı kendi kendini yenileyebilecek denetleyiciler kullanılır. Arızaya dayanıklı denetim (FTC) başlığı asıl

olarak bu ikinci yaklaşımı ele almakta ve işlemektedir. Bu sebeple bundan sonra anlatılacak konular ikinci yaklaşım göz önüne alınarak hazırlanmıştır.

Arızaya dayanıklı denetimin kullanım alanları tamamen arıza bulma ve yalıtım alanlarıyla ortaktır, zaten arıza bulma ve yalıtımının uzantısı olarak da düşünülebilir. Kimyasal süreçler, hava ve uzay araçları, arabalar, nükleer reaktörler, robotlar ve bina havalandırma sistemleri gibi alanlar FTC'in uygulama alanlarını oluşturmaktadır.

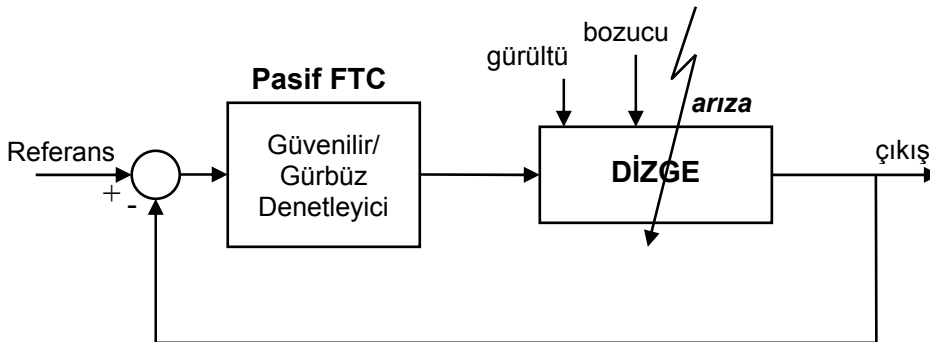
FTC, denetleyicilerinin tiplerine bağlı olarak iki ana başlık altında toplanır:

- Pasif FTC
- Aktif FTC

#### 1.4.1. Pasif FTC

Pasif FTC, çalışması sırasında herhangi bir arıza bilgisine ihtiyaç olmayan ve dizgeyi belirli tipte arızalara duyarsız hale getirmek için tasarlanmış denetim yöntemleridir (Kanev, 2004; Zhang ve Jiang, 2008). Arıza bilgisine ihtiyaçları olmaması bu dizgeleri çekici hale getirir fakat uygulanabilirlik alanları çok sınırlıdır.

Pasif FTC kullanan şemalarda arızaya dayanıklılık çoğunlukla belirli tipteki arızaları dizgedeki belirsizlik olarak tanımlayarak başarılır. Pasif FTC kapalı çevrim dizgeyi güvenilir yapmayı, bu sayede bazı artımlı arızalarda kararlılık ve başarıyı değişmez tutmayı amaçlayan güvenilir denetleyiciler (reliable controllers) ve belirli tipteki arızalar için dizgelerdeki belirsizlikler ve bozucularla başa çıkmak için tasarlanan gürbüz denetleyiciler (robust controller) olmak üzere iki başlıkta toplanır. Pasif FTC'in genel yapısı Şekil 1.14'te verilmiştir.

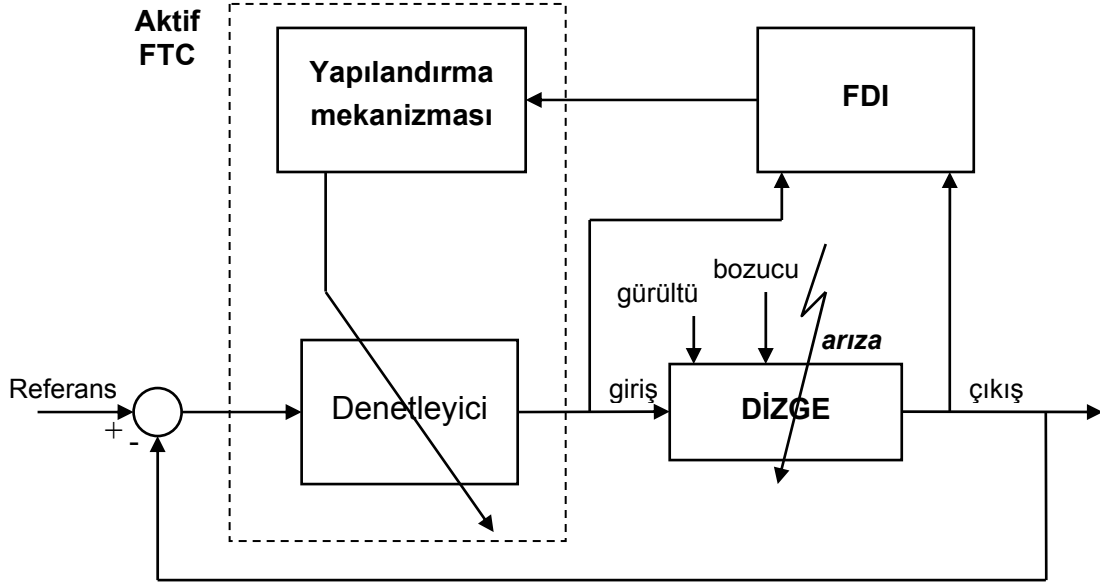


Şekil 1.14. Pasif FTC'in genel yapısı

#### 1.4.2. Aktif FTC

Aktif FTC, pasif FTC'den farklı olarak arıza bulma ve yalıtımına bağlı olarak

denetleyicilerin yapılandırılması (reconfiguration) yada belirli denetleyicilerin seçimi veya karıştırılması ilkesine dayanır (Kanev, 2004; Zhang ve Jiang, 2008). Aktif FTC'in genel yapısı Şekil 1.15'te verilmiştir :



Şekil 1.15. Aktif FTC'in genel yapısı

Yapı, kısaca açıklanacak olursa, FDI kısmı dizgeye ait giriş ve çıkış ölçümlerinden arızaları bulur ve teşhis eder. Kestirilen arızalar, arıza sonrası (post-fault) dizge başarımını sağlamak için denetleyicinin parametrelerini ve/veya yapısını değiştiren yapılandırma mekanizmasına geçer. Sonuç olarak denetleyici yeniden yapılandırılarak arızaya karşı dayanıklı bir yapı elde edilmiş olur.

Doğrusal dizgeler için birçok FTC yöntemi önerilmiştir. Tez çalışmada bu yöntemlerle ilgili ayrıntılı yer verilmeden sadece genel başlıklara ve yapılan çalışmalarla ilgili kaynaklara yer verilmiştir.

- Söзде tersi (Pseudo-inverse), (Kanev, 2004; Patton, 1993)
- Özyapı ataması (Eigenstructure Assignment), (Konstantopoulos ve Antsaklis, 1996)
- Çoklu model (Multiple Model), (Maybeck ve Stevens, 1990)
- Model takibi (Model Following), (Zhang ve Jiang, 2003)
- Uyarlanabilir denetim (Adaptive Control), (Diao ve Passino, 2001)
- Model öngörülü denetim (Model Predictive Control-MPC), (Bošković ve Mehra, 2002; Maciejowski ve Jones, 2002)
- Nicel Geribesleme Kuramı Denetimi (Quantitative Feedback Theory-QFT Control),

(Wu ve ark, 1999)

- Sinir Ağları ve Bulanık Mantık (Polycarpou ve Helmicki, 1995)

### 1.4.3. Doğrusal Olmayan Dizgeler İçin Arızaya Dayanıklı Denetim Tasarımı

Tez çalışmasında robot manipülatörler ele alındığından ve robot dizgelerinin doğrusal olmayan yapıda olmalarından dolayı doğrusal olmayan dizgeler için FTC tasarımı ile ilgili çalışmalar ayrıntılı biçimde incelenmiştir. Bölüm 1.3.5'te verilen yaklaşımın aynısı FTC tasarımı için de geçerli olmuş, doğrusal dizgeler için Bölüm 1.4.1 ve 1.4.2'de verilen yöntemler doğrusal olmayan dizgelere uyarlanmaya çalışılmıştır. Bu uyarlama sırasında yine benzer şekilde dizgeler çeşitli çalışma noktalarında doğrusallaştırılmıştır. Her bir denge noktasında bir model oluşturulmuş ve bu modeller üzerinde çalışmalar yapılmıştır. Bu yaklaşımdaki en büyük sorun çalışma noktaları fazlalaştıkça ve her bir çalışma noktasında ayrı ayrı arızalar için de ayrıca modeller oluşturulması gerekliliği ve buna bağlı ortaya çıkan hesapsal alanda artan ihtiyaçtır.

Bu altbölümün izleyen kısmında yapılan çalışmalara dair bir literatür özeti verilmiştir. FTC tasarımına dair yapılan çalışmaların her biri sadece FTC tasarımını değil, ayrıca FDI tasarımını da içinde barındırdığından ve uzun kararlılık çözümlenmeleri sonucunda elde edildiğinden çalışmalara dair teorik bilgilere burada yer verilmemiştir.

Diao ve Passino, doğrusal olmayan dizgelerin denetimi için tasarlanan uyarlanabilir denetleyicileri FTC tasarımı için kullanmışlardır (Diao ve Passino, 2002).

Zhang, Zhang ve arkadaşları, daha önce arıza teşhisi için tasarladıkları uyarlanabilir öğrenen dizgelerine yapılandırılabilir bir denetleyici eklemiştir (Zhang, 2002; Zhang ve ark., 2001, 2004).

Yen ve Lima, optimal kontrolde tanımlanan Hamilton-Jacobi-Bellman işlevini çözümede kullanılan dual sezgisel programlama yöntemini doğrusal olmayan dizgeler için FTC tasarımında kullanmışlardır (Yen ve Lima, 2002, 2005).

Qu ve arkadaşları, normal ve arızalı durum için iki farklı gürbüz denetleyici tasarlamışlardır (Qu ve ark., 2001, 2003).

Yu ve arkadaşları, Zhou ve Frank, FTC olarak dizge durumuna göre katsayılarını değiştirebilecek PID denetleyiciler tasarlamışlardır (Yu ve ark, 2005; Zhou ve Frank, 1998).

Lim ve Shin, Ganguli ve arkadaşları, Stoustrup ve Niemann, özel tip bir doğrusal

olmayan dizge sınıfı olan doğrusal parametre değişimli (Linear Parameter Varying-LPV) dizgeler için doğrusal matris eşitsizliklerine (Linear Matrix Inequalities-LMA) dayanan çeşitli FTC'ler, Sauter ve arkadaşları aynı tipte dizge için doğrusal karesel (Linear Quadratic-LQ) denetleyici tasarımına dayanan bir FTC tasarlamışlardır (Lim ve Shin, 2004; Ganguli ve ark.,2002; Stoustrup ve Niemann, 2004; Sauter ve ark., 1998).

Kabore ve Wang, Bajpai, Bajpai ve arkadaşları, Thomas ve arkadaşları, doğrusal olmayan geribesleme tasarımına dayanan FTC'ler tasarlamışlardır (Kabore ve Wang, 2001; Bajpai, 2001; Bajpai ve ark, 2002; Thomas ve ark., 2003).

Diallo ve arkadaşları, FTC'ye farklı bir yaklaşım getirerek ayrı durumlarda devreye girecek farklı tipte denetleyiciler tasarlamışlardır (Daillo ve ark., 2004).

Yen ve Ho, doğrusal olmayan dizgelerin denetimi için kullanılan kayan kipli (Sliding Mode-SM) denetimi FTC tasarımı için kullanmışlardır (Yen ve Ho, 2000).

Wang ve Wang, iki sinir ağı denetleyicisini, birincisi sağlıklı durumda denetimi yapacak, ikincisi arıza durumunda arızayı dengeleyecek şekilde kullanmıştır (Wang ve Wang, 1999). Polycarpou ve arkadaşları, Pashilkar ve arkadaşları, sinir ağını arızayı dengeleyecek şekilde kullanmıştır (Polycarpou ve ark., 2004; Pashilkar ve ark., 2006). Tang ve Yao, bulanık-sinirsel bir denetleyiciyi tek bir denetleyici olarak sağlıklı ve arızalı durum için kullanmışlardır (Tang ve Yao, 2005). Ho ve Yen ,sinir ağını tasarlanan denetleyiciye arıza durumunda dizge dinamikleri hakkında bilgi vermek için kullanmıştır (Ho ve Yen, 2001).

Toribio ve arkadaşları, bulanık gözlemci yaklaşımını bulanık durum geribesleme denetleyicileri ile birleştirerek bir FTC tasarlamışlardır (Toribio ve ark., 2000) . Zidani ve arkadaşları, iki denetleyiciyi, birincisi sağlıklı durumda çalışacak, ikincisi arıza durumunda çalışacak şekilde tasarlamışlar, ikinci denetleyici olarak bir bulanık denetleyiciyi kullanmışlardır (Zidani ve ark., 2003). Kwong ve arkadaşları, bulanık mantığı hem modellemede hem de FTC olarak denetleyici tasarımında kullanmışlardır (Kwong ve ark., 1995).

FTC tasarımı için yöntemler ve çalışmalar yeterli görünse de belirli sorunlar üzerindeki çelişkiler hala devam etmektedir (Zhang ve Jiang, 2008). Bu sorunlar üç temel başlık altında toplanabilir:

- Nazik başarımlar düşümü
- Karalılık ve gürbüzlük
- FDI belirsizlikleri, yapılandırma gecikmesi ve gerçek zamanda çalışma

#### **1.4.3.1. Nazik Başarım Düşümü (Graceful Performance Degradation)**

Herhangi bir FTC tasarımında en önemli konulardan biri özgün dizge başarımının tamamen korunup korunmayacağına veya arıza sonrasında belirli bir başarım düşümünün kabul edilip edilmeyeceğine karar vermektir. “Eğer başarım düşümü gözardı edilirse bunun dizgeye sonuçları ne olur? Başarım düşümü kabul edilirse bu tasarım sürecinde hesaba nasıl katılmalıdır?” gibi sorular FTCS tasarımında önem kazanmaktadır. Pratikte algılayıcı arızaları durumunda özgün dizge başarımı fazladan algılayıcılar veya gözlemci veya Kalman kestirimcileri yardımıyla elde edilebilir. Fakat bir hareketlendirici arızası olduğunda dizge üzerindeki denetim yetenekleri belirgin bir şekilde düşer. Eğer tasarımın amacı arıza durumunda bile özgün dizge başarımını sağlamaksa arızalarla başa çıkmak için kalan diğer hareketlendiriciler olağan çalışmalarının ötesinde çalışmaya zorlanabilir. Bu yaklaşım hareketlendiricilerin fiziksel sınırlarından dolayı fazla tercih edilmez. Ayrıca bu yaklaşım hareketlendiricilerde hareketlendirici doyumlarına (saturation) sebep olabilir ve dizgede daha büyük arızalara veya bozunumlara sebep olabilir. Bu yüzden FTC tasarımlarında başarılılabilecek dizge başarımlarıyla varolan hareketlendirici yetenekleri arasındaki ödünleşmeye dikkat edilmelidir. Bu duruma nazik başarım düşümü adı verilir.

#### **1.4.3.2. Kararlılık ve Güzbüzlük**

Kararlılık herhangi bir denetleyicinin ilk gereksinimidir. FTC içeren dizgelerin kararlılığı üç dizge çalışma dönemini içine alır: 1) Arıza olmayan dönem 2) yapılandırma sırasında geçici dönem ve 3) yapılandırma sonrasındaki sürekli durum. Ayrıca herhangi pratik denetleyici için kararlılıkta ve başarımında güzbüzlük gerçekte kullanılabilir olması için çok önemlidir.

#### **1.4.3.3. FDI Belirsizlikleri, Yapılandırma Gecikmesi ve Gerçek Zamanda Çalışma**

Kesin bir arıza teşhisi denetim yapılandırmasında öncü olan önemli bir konudur. Yine de pratikte FDI belirsizlikleri adı verilen kestirim veya teşhis hataları çoğunlukla bulunmaktadır. Ayrıca FDI ve denetim yapılandırılması ile ilgili gecikmeler de bulunmaktadır. Aktif FTC içeren dizgeler gerçek zamanda çalışan dizgeler olarak tasarlandığından arızaların bulunması, teşhisi ve arızaya karşı yapılandırma hesaplarının ve işleminin hızlı bir şekilde yapılması gerekmektedir. FTC tasarım amaçlarından biri

de yapılandırılabilir denetleyici tasarımında bu FDI belirsizliklerini gözönüne almak ve olabildiğince gecikmelerin etkilerini azaltmaktır.

## 2. ROBOT MANİPÜLATÖRLERDE ARIZA BULMA, YALITIMI VE ARIZAYA DAYANIKLI DENETİM

Robotlar çoğu endüstriyel veya özel amaçlı uygulamada destekleyici bir alt dizge veya karmaşık bir dizgenin özel bir kısmı olarak görev yaparlar. Seri üretim hatları gibi insanlardan daha hızlı, daha zorlu ve daha fazla doğrulukta çalışabilecekleri uygulamaların yanında nükleer santrallerde temizlik, denizaltı ve uzay yolculuklarında veri ve örnek toplama, bakım gibi insanlar için tehlikeli sayılabilecek birçok görevi yerine getirmektedir. Sonuç olarak bir seri üretim hattındaki robotta oluşan arıza esnek otomasyon (flexible automation) dizgelerinde bile tüm bağlantılı hatlarda bir duraklamaya veya özellikle tehlikeli görevlerin başarımı sırasında oluşabilecek fark edilmeyen en küçük arıza veya hata büyük parasal maliyetlerle tüm görevin iptaline sebep olabilir, hatta insanlara zarar verecek sonuçlar doğurabilir (Visinsky, 1991). Bu sebeple robotlar üzerinde yapılan çoğu çalışma robot denetimi üzerineyken son yıllarda robot güvenilirliği ve robotlarda arıza bulma ve teşhisi üzerine çalışmalar yavaş yavaş artmaktadır. Yapılan çalışmalara ek olarak, yukarıda açıklanan durumlarla gerçek hayattaki örneklerinin artması sonucunda NASA ve Amerikan ordusu robotlar ve robotlarda kullanılan parçaların güvenilirliği ve arıza olasılıkları üzerine belirli standartlar yayınlamışlardır (Cavallaro ve Walker, 1994).

Robotlar üzerindeki FDI ve FTC çalışmaları birçok tipte robot (gezgin, esnek, paralel, iki ayaklı veya çok ayaklı, kinematik fazlalıklı vs.) üzerinde devam etmektedir. Tez çalışması seri ve sert robot manipülatörleri konu olarak aldığından sadece bu başlıkla ilgili ayrıntılar verilecektir. Diğer başlıklarla ilgili kaynaklar araştırılabilir (Goel ve ark., 2000; Alkoç, 2008; Notash ve Moore, 2002).

Robot manipülatörlere ilişkin genelleştirilmiş dinamikler Eş. (2.1) ile tanımlanmaktadır :

$$M(q(t))\ddot{q}(t) + V(q(t), \dot{q}(t)) + G(q(t)) + F(\dot{q}(t)) = \tau \quad (2.1)$$

Burada  $n$  eklem sayısı olmak üzere  $q(t), \dot{q}(t), \ddot{q}(t) \in \mathcal{R}^n$  sırasıyla zamana bağlı eklem açısız konum, hız ve ivme vektörlerini,  $M(q(t)) \in \mathcal{R}^{n \times n}$  pozitif tanımlı eylemsizlik matrisini,  $V(q(t), \dot{q}(t)) \in \mathcal{R}^n$  merkeçil ve merkezkaç vektörünü,  $G(q(t)) \in \mathcal{R}^n$  yerçekimi vektörünü,  $F(\dot{q}(t)) \in \mathcal{R}^n$  sürtünme vektörünü ve  $\tau \in \mathcal{R}^n$  eklemlere uygulanan moment vektörünü temsil etmektedir. Eş. (2.1)'de doğrusal

olmayan terimlerin daha kolay gösterilebilmesi amacıyla açısız ivme içeren terimler dışındaki terimler  $N(q(t), \dot{q}(t)) = V(q(t), \dot{q}(t)) + G(q(t)) + F(\dot{q}(t))$  şeklinde tanımlanırsa

$$M(q(t))\ddot{q}(t) + N(q(t), \dot{q}(t)) = \tau \quad (2.2)$$

olur. İzleyen altbölümde robot manipülatörlerde görülen arızaların Eş. (2.1) ve (2.2) ile tanımlı robot dinamiklerine olan etkileri anlatılacaktır.

## 2.1. Robot Manipülatörlerde Görülen Arızalar

Bölüm 1.2’de anlatıldığı üzere arızalar oluşma yerlerine, modellenmelerine ve zaman karakteristiklerine göre ayrılmaktadır. Bu altbölümde robot manipülatörlerde görülebilecek arızaların oluşma yerlerine göre nasıl sınıflandırıldığı anlatılacaktır (Fantuzzi ve ark., 2003). Robot manipülatörlerdeki arızaların modellenmeleri ve zaman karakteristikleri ise Bölüm 1.2 ve izleyen altbölümlerinde verilen özellikler ile aynı özellikleri taşımaktadır.

### 2.1.1. Robot Manipülatörlerde Hareketlendirici Arızaları

Robot manipülatörlerde hareketlendirici olarak motorlar görev yapmaktadır ve bunlarda görülen tipik arızalar robotun hareket yeteneklerini etkilemektedir. Çoğunlukla güç katındaki arızalardan, eklemdaki magnetik frenlerden veya güç aktarım bileşenlerindeki arızalardan kaynaklanan bu tip arızalar şu şekilde adlandırılmaktadır:

**a) Kilitli eklem arızası (Locked joint failure):** Bu arızada eklemin bağlı olduğu motor, eklem frenindeki bir arızadan dolayı değişmez konumda kalır ve harekete izin vermez. Bu tip arızalarda arızanın tanımlandığı  $i$ . ekleme ait  $\tau_i$  serbest halde davranırken  $q_i(t)$  değişmez olarak davranır.

**b) Serbest sallanan eklem arızası (Free-swinging joint failure):** Bu arızada olası bir güç kablosundaki kopmadan dolayı ekleme etki eden motorun uyguladığı moment sıfır olur. Bu tip arızalarda arızanın tanımlandığı  $i$ . ekleme ait  $\tau_i$  sıfır,  $q_i(t)$  ise serbest ve diğer eklemlerin veya yerçekiminin etkisi altındadır.

**c) Kısmi (Partial) eklem arızası:** Bu arızada ekleme etki eden motorun uygulaması gereken moment motordaki arızadan dolayı belirli bir oranda (örn. %20) azalır. Bu tip arızalarda Eş. (2.2) ile tanımlı dinamikler aşağıdaki hale gelir:

$$M(q(t))\ddot{q}(t) + N(q(t), \dot{q}(t)) = \tau(t) - \alpha u(t - T) \cdot \tau(t) \quad (2.3)$$

Burada  $\alpha \in \mathfrak{R}^{n \times n}$  kısmi eklem arıza matrisi,  $u(t) \in \mathfrak{R}$  birim adım ve  $T$  arıza oluşum anıdır.

### 2.1.2. Robot Manipülatorlerde Bileşen Arızaları

Robot manipülatorlerde bileşen arızası olarak eklem kırılmaları, dişli aşınması ve zincir sıkışma yada kurtulması gibi arızalar görülmektedir. Bu tip arızaların oluşması durumunda Eş. (2.2) ile tanımlı dinamikler aşağıdaki gibi değişir:

$$M(q(t))\ddot{q}(t) + (N(q(t), \dot{q}(t)) + f_N) = \tau \quad (2.4)$$

Burada  $f_N \in \mathfrak{R}^n$  bileşen arıza vektörünü temsil etmektedir.

### 2.1.3. Robot Manipülatorlerde Algılayıcı Arızaları

Robot manipülatorlerde algılayıcı olarak eklem konumlarını algılayan optik kodlayıcılar, hız algılama için takogeneratörler bulunmaktadır. Bunların haricinde özel olarak motor akımını algılayan Hall etkili algılayıcılar, kuvveti algılayan dokunsal (tactile) algılayıcılar bulunmaktadır. Bu algılayıcılarda da diğer algılayıcılar için belirtilen arızalar (öngerilim vs.) görülmektedir. Bu tip arızaların oluşması durumunda Eş. (2.2) ile tanımlı dinamikler aşağıdaki gibi değişir:

$$y(t) = [q_1(t) \cdots q_n(t) \dot{q}_1(t) \cdots \dot{q}_n(t)]^T + \Delta f_{q,\dot{q}} \quad (2.5)$$

Burada  $\Delta f_{q,\dot{q}} \in \mathfrak{R}^{2n}$  vektörü algılayıcı arıza vektörünü temsil etmektedir.

## 2.2. Robot Manipülatorlerde Arıza Bulma ve Yalıtımı

Robot manipülatorlerde arıza bulma ve yalıtım çalışmaları son 20 yıl içinde ortaya çıkmış ve hızlanmıştır. Bu çalışmalarla birlikte ortaya çıkan en büyük sorun Eş. (2.1-2.5) arasında verilen robot dinamiklerinin doğrusal olmayan eşitliklerle tanımlanmasıdır. Yapılan çoğu çalışmada doğrusal olmayan dizgeler için uygulanan FDI yöntemleri robot manipülatorlere uygulanmaya çalışılmıştır. Bu altbölümün izleyen kısmında şu ana kadar yapılan çalışmalar ile ilgili bilgiler verilmiştir.

Robot manipülatorlerde FDI üzerine çalışmaların çoğu gözlemci temelli yaklaşımlar üzerine kurulmuştur. Caccavale ve Walker, robot dinamiklerini belirli tipte doğrusal olmayan dizgelere benzetmeye çalışmış ve bir gözlemci tasarlamışlardır (Caccavale ve Walker, 1997). Daha sonra Caccavale ve arkadaşları, bu çalışmanın

devamı şeklinde belirsizlikleri kestirecek destek vektör makinalarını (Support Vector Machines-SVM) kullanan bir çalışma yapmışlardır (Caccavale ve ark., 2009). Schneider ve Frank, benzer şekilde doğrusal olmayan dizgeler için tasarlanmış gürbüz bir gözlemciyi robot manipülatör için kullanmış ve artık değerlendirme/arıza yalıtımı için bulanık mantıktan yararlanmışlardır (Schneider ve Frank, 1996).

Leuschen, Leuschen ve arkadaşları, analitik fazlalık tanımını doğrusal dizgelerden doğrusal olmayan dizgelere taşımış, doğrusal olmayan analitik fazlalık tanımını yaparak hidrolik bir robot manipülatör ile iki eklemlili IMI robot üzerinde çeşitli FDI çalışmaları yapmışlardır (Leuschen, 2001; Leuschen ve ark., 2005). Halder ve Sarkar, bu çalışmanın devamı ve uzantısı niteliğinde gürbüz doğrusal olmayan analitik fazlalık tanımını yapmış ve PUMA560 robot manipülatörü üzerinde FDI çalışmaları yapmışlardır (Halder ve Sarkar, 2007).

De Luca ve Mattone, robot dinamiklerini uyarlanabilir denetleyiciler için tanımlanmış genelleştirilmiş gösterilime benzeterek uyarlanabilir yapıda bir FDI şeması oluşturmuşlardır (De Luca ve Mattone, 2004).

Dixon ve arkadaşları, robot manipülatöre uygulanan moment işaretini kendi önerdikleri bir süzgeçten geçirip elde ettikleri işaret ve bu işaretin kestirimi arasındaki farkı kullanarak özellikle parametrik belirsizliklere gürbüz bir FDI yöntemi önermişlerdir (Dixon ve ark., 2000).

Abdul ve Liu, modüler yapıdaki robot manipülatörler için konum ve hız kestirimi için analitik bir yöntem önermiş ve bu değere bağlı olarak bir arızaya dayanıklı denetleyici tasarlamışlardır (Abdul ve Liu, 2008).

Brambilla ve arkadaşları, ters robot modeli yardımıyla hareketlendirici arızalarını ve algılayıcı arızalarını kendi önerdikleri kayan kipli gözlemcileri kullanan genelleştirilmiş gözlemci şemaları yardımıyla yalıtmışlardır (Brambilla ve ark., 2008).

Chen ve Saif, robot modelini durum uzayında bilinmeyen girişli dizgelere benzetmiş ve çıkış kestirimcileri yardımıyla arıza bulma işlemini gerçekleştirmişlerdir (Chen ve Saif, 2008).

Doğrusal olmayan dizgelerde FDI için kabul gören SA robot manipülatörler için de kullanılmıştır. Naughton ve arkadaşları artık türetimi için Adjallah ve arkadaşlarının önerdiği doğrusal olmayan gözlemciyi kullanmış, artık değerlendirme için SA'ndan yararlanmışlardır (Naughton ve ark., 1996). Vemuri ve Polycarpou, arızayı robot model işlevinin bir bileşeni olarak kabul ederek SA için tanımlanan uyarlanabilir öğrenme

yaklaşımını arıza işlevine yakınsamak için kullanmışlardır (Vemuri ve Polycarpou, 1997). Terra ve Tinos, çeşitli tipte SA yapılarını hem artık türetimi, hem de artık değerlendirme için kullanmışlardır (Terra ve Tinos, 2001). Lee ve arkadaşları, parametre kestirimi yöntemleriyle arıza bulma ve ART tipindeki SA'nı kullanarak arıza yalıtma işlemini bileşen ve algılayıcı tipinde arızalar için denemişlerdir (Lee ve ark., 2003). Datta ve arkadaşları, ayırık dalgacık dönüşümü yardımıyla elde ettikleri katsayıları bir sinir ağı yardımıyla sınıflandırmaya çalışmışlardır (Datta ve ark., 2007).

### **2.3. Robot Manipülatörlerde Arızaya Dayanıklı Denetim**

FTC tanımına bağlı olarak robot manipülatörlerde FTC, varolan robotun verilen bir görevi (örn. belirli bir yörüngeyi takip etme) yerine getirirken oluşabilecek bir arıza karşısında arızaları bulma, yalıtım ve başa çıkabilme yeteneği olarak tanımlanır. Verilen bu tanıma bağlı olarak robot manipülatörlerde iki türlü FTC yaklaşımından söz edilebilir :

- Robot manipülatörlerde donanımsal fazlalıklar ile arızaya dayanıklı denetim
- Robot manipülatörlerde denetim yöntemleri ile arızaya dayanıklı denetim

#### **2.3.1. Robot Manipülatörlerde Donanımsal Fazlalıklar İle Arızaya Dayanıklı Denetim**

Donanımsal fazlalıklar ile robot manipülatörlerde FTC temel olarak bir dizgede tüm olası arızaları dengeleyecek (veya karşılayacak) şekilde fazlalıklar ekleme tanımına dayanmaktadır (Paredis, 1996). Fakat bu tanıma bağlı olarak eklenen tüm fazlalıkların arızaya dayanımla sonuçlanmayacağına dikkat edilmelidir. Bu durumda robot manipülatörlerde donanımsal fazlalıklar ile arızaya dayanıklı denetim konusuna açıklık getirmek gerekmektedir.

Robot manipülatörlerde arızaya dayanıklı denetimde donanımsal fazlalık kullanım yaklaşımında ilk temel adım modül yapısında olan, kullanıcıya kendi robot manipülatör tasarım özgürlüğünü veren, tüm eklemleri ve uzuvları söküp takılabilen, aktif hareketlendiricilerin yanında pasif hareketlendiricilerin de kullanılabilmesi, arıza durumlarında eklemlerin ve uzuvların birbirlerini etkilemedikleri Reconfigurable Modular Manipulator System-RMMS dizgesi ile olmuştur (Paredis, 1996; Paredis ve Khosla, 1994). Bu ve daha sonra ortaya çıkan benzer tasarımlar hareketlendirici fazlalıkları üzerinde durmuş ve sonuçta hareketlendirici fazlalıkları için 4 adımlı bir

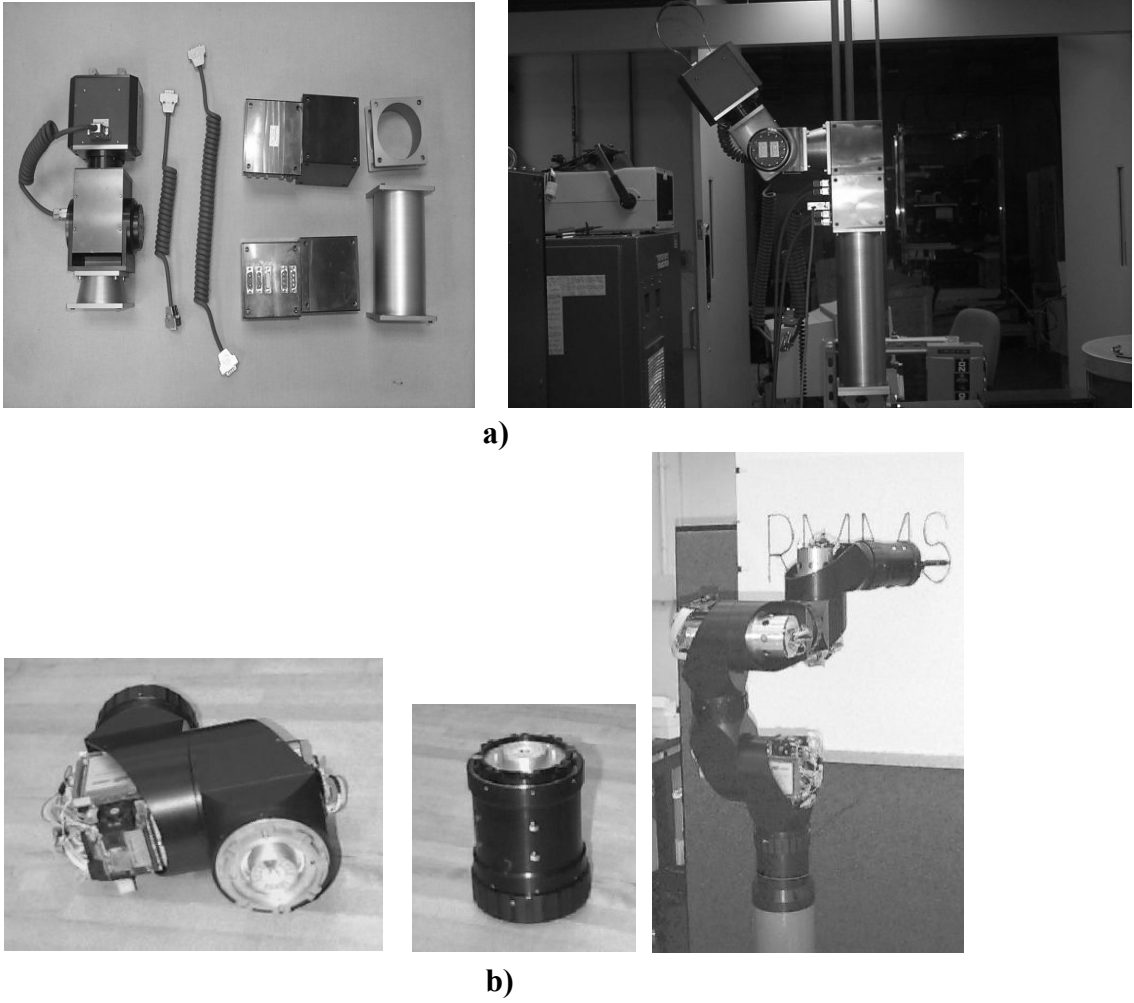
tasarım yordamı oluşmuştur :

- Adım 1 : Çifte hareketlendiriciler-her eklem için fazladan bir hareketlendirici
- Adım 2 : Paralel yapılar- her serbestlik derecesi (Degree of freedom-DOF) için fazladan bir eklem
- Adım 3 : Fazlalıklı manipülatörler- her manipülatör kol için fazladan bir DOF
- Adım 4 : Her manipülatör dizge için fazladan bir kol

Bu 4 adımı izleyen çalışmaların sonucunda arızaya dayanıklı robot dizge tasarımları temel olarak ikiye ayrılmıştır. İlk yaklaşımda amaç genel amaçlı arızaya dayanıklı bir manipülatör tasarlamaktır. Burada sadece arıza sonrasında oluşan ve kartezyen uzayın bir alt uzayı olan görev uzayı belirlenir. Robot manipülatörün arıza durumunda görev uzayında izleyeceği yol hakkında herhangi bir kabul yapılmaz. İkinci yaklaşımda ise amaç göreve özel arızaya dayanıklı bir manipülatör tasarlamaktır. Burada robot manipülatörün kartezyen uzayda izlemesi gereken yol biliniyor kabul edilir ve fazlalıklar ile olası eklem yörünge uzayları buna bağlı olarak belirlenir. Belirtilen bu tanımlara bağlı robot tasarımlarıyla ilgili geniş ayrıntı kaynaklarda bulunabilir (Paredis, 1996; Visinsky, 1991). Şekil 2.1.a'da piyasada ticari olarak bulunan Amtec Robotics Reconfigurable Light Weight Robots-RLWR'a ait eklem ve uzuv modülleri ile 3 eklemlili RLWR robotu, Şekil 2.1.b'de ise Carnegie-Mellon Üniversitesi tarafından tasarlanmış olan Reconfigurable Modular Manipulator Systems-RMMS'e ait eklem ve uzuv modülleri ile 7 eklemlili RMMS robotu görülmektedir.

Robot manipülatörlerde donanımsal fazlalıklar ile arızaya dayanıklı denetim yaklaşımında unutulmaması gereken arıza bulma ve yalıtım işlemlerinin hala gerekli olduğudur. Arızanın teşhisinden sonra eğer gerekiyorsa bozulan hareketlendirici frenler yardımıyla pasif hale getirilebilir ve arıza sonrası durum için yörünge tayini yapılır.

Robot manipülatörlerde donanımsal fazlalıklar ile arızaya dayanıklı denetimle ilgili yapılan çoğu çalışma eklem kilitlenmesi ve serbest sallanan eklem arızaları konu aldığından kinematik olarak fazlalıklı robotların arızalanması sonucu kinematik konfigürasyonun ve dinamik denetimin yeniden nasıl yapılandırılacağı üzerinde durmaktadır. Özellikle Paredis ve Khosla'nın hem manipülatör tasarımının nasıl yapılması gerektiği, hem de tasarımdan sonra yörünge planlaması ile ilgili çalışmalarını incelenebilir (Paredis ve Khosla, 1995, 1996a, 1996b). Ayrıca kinematik fazlalıklı manipülatörlerde serbest sallanan eklem arızası ile ilgili English ve Maciejewski'nin de çalışmaları incelenebilir (English ve Maciejewski, 1998).



**Şekil 2.1** Donanımsal fazlalıklı modüler robotlar

**a)** RLWR eklem ve uzuv modülleri ve RLWR 3 eklemlili robot

**b)** RMMS eklem ve uzuv modülleri ve RMMS 7 eklemlili robot (Paredis, 1996)

### 2.3.2. Robot Manipülörlerde Denetim Yöntemleri ile Arızaya Dayanıklı Denetim

Bölüm 2.3.1.'de anlatıldığı üzere robot manipülörlerde FTC ile ilgili yapılan çoğu çalışma eklem kilitlemesi ve serbest salınan eklem arızaları üzerinde durmaktadır. Bu tip bir arızadan sonra manipülatöre ait bir eklem tamamen devre dışı kalır veya bırakılır ve manipülatör düşük hareketli veya özürlü manipülatör (underactuated manipulator) durumuna gelir. Bu altbölümde tez çalışmasında hem konu alınan kısmi eklem arızaları ve FTC için denetleyici tasarımı üzerine yapılmış çalışmalara hem de özürlü manipülatörde yapılan FTC çalışmalarına yer verilmiştir.

Izumikawa ve ark. esnek bir manipülatör için sağlıklı ve arızalı durumda çalışacak iki denetleyicili bir FTC şeması tasarlamışlardır (Izumikawa ve ark., 2002).

Ayrıca başka bir çalışmada sağlıklı ve arızalı durumlarda katsayılarını değiştirecek bir PD denetleyicili FTC şeması tasarlamışlardır (Izumikawa ve ark., 2004).

Sreevijayan ve ark., Tosunoğlu, robot manipülatörlerde oluşabilecek arızalar için hem fazlalık hem de kayan kipli ve uyarlanabilir denetleyiciler ile FTC tasarımları yapmışlardır (Sreevijayan ve ark.,1994; Tosunoğlu, 1995).

Bonivento ve ark., robot manipülatörler için sağlıklı durumda çalışacak bir denetleyici ve kayan kipli denetleyicilere benzer bir denetleyici ile iki denetleyicili bir FTC şeması tasarlamışlardır (Bonivento ve ark.,2004).

Shin ve ark., özürlü bir manipülatör için sağlıklı ve arızalı durumda çalışacak gürbüz uyarlanabilir bir denetleyici tasarlamışlardır (Shin ve ark.,1999).

Siqueira ve Terra, özürlü manipülatör için doğrusal olmayan dizgelerde denetim için kullanılan  $H_2$  ve  $H_\infty$  denetleyicilerini kullanmışlardır (Siqueira ve Terra, 2004).

Zhang ve ark., robot manipülatörler arıza öncesinde kararlılık güvencesi sağlayan, arıza sonrasında ise elde ettiği arıza bilgisini değerlendirerek arızayı dengelemeye (compensate) çalışan bir FTC tasarlamışlardır (Zhang ve ark., 2009).

Kuntze ve ark., insan etkileşimli bir gezgin manipülatör için birçok karışık (PID, BM ve Model Temelli Denetleyici) alt-denetleyiciyi kullanan ve bunlara ait arızalı durumdaki seçimi bir bulanık durum seçici tarafından yapan bir FTC tasarlamışlardır (Kuntze ve ark., 2003).

#### **2.4. Tez Çalışmasının Özgün Değeri**

Tez çalışmasında robot manipülatörler için, konum ve hız işaretlerini kullanan, belirsizliklere gürbüz yapıda 3 farklı FDI şeması ve 3. şemanın uzantısı şeklinde 1 FTC şeması önerilmiştir. Önerilen bu şemalar iki eklemlili düzlemsel bir manipülatör için MATLAB altında benzetim olarak gerçekleştirilmiştir. Ayrıca önerilen şemalar ileride gerçek bir manipülatör üzerinde yapılabilecek çalışmalara uygun bir şekilde bir sayısal işaret işleyiciye (Digital Signal Processor-DSP) gömülmüştür. Şemaların ayrıntı olarak açıklaması izleyen bölümlerde verilmiştir.

Literatürde yapılan çalışmaların çoğu doğrusal olmayan dizgeler için uygulanmış analitik yöntemlerin robot manipülatörlere uygulanmış halleridir. SA ve BM gibi yöntemleri kullanan çalışmaların sayısı çok azdır. Bu yöntemleri kullanan çalışmalarda yine analitik yöntemlerdeki çözümleri gerektiren parametre güncelleme-uyarlanabilir öğrenme yaklaşımı tercih edilmektedir. Varolan çalışmalar ise

kısmi eklem arızaları yerine tespiti daha kolay olan kilitli ve serbest sallanan eklem arızaları üzerine odaklanmıştır. Bu tez çalışması ise yapay zeka tekniklerini doğrudan veri öğrenen yöntemler olarak kullanarak elde olan verilerin öğrenilmesini temel almaktadır. Tez çalışması robot manipülatörlerde üzerinde daha az çalışma yapılmış ve belirlenmesi serbest sallanan eklem ve kilitli eklem arızalarına göre daha zor olan kısmi eklem arızalarına odaklanmıştır ve literatürdeki bu eksikliği gidermeyi hedeflemektedir.

Literatürde yapılan çalışmaların çoğu arıza bir kez belirlendikten sonra artıkların durumlarıyla ilgilenmemektedir. Bu durum ise arızaya dayanıklı denetimi imkansız hale getirmektedir. Ayrıca yapılan çoğu çalışma FTC tasarım amacıyla olmadığından arıza işlevine yakınsama önemsenmemiştir. Tez çalışmasında bu durum gözönüne alınarak sürekli olarak artık türetimi gözetlenmiş, önerilen 3. şemada arıza işlevine yakınsanarak tam bir FTC tasarımına ulaşılmıştır.

Bununla birlikte edinilen arıza bilgisi hem hangi eklem arızalandığını söyleyebilmekte, hem de arızanın boyutları hakkında bilgi verebilmektedir. Bu özelliği ile tam bir arıza teşhisi yapan tez çalışması diğer çalışmalardan ayrılmaktadır.

Her ne kadar tez çalışması benzetim seviyesinde kalmış görünse de benzer şekilde literatürde yapılan çoğu çalışmanın benzetim seviyesinde kaldığı görülmektedir. Gerçek bir robotun arızalandırılması durumunda geri dönüşü olmayan yıkımsal sonuçların doğabileceği düşünüldüğünde bu yaklaşım doğaldır. Çalışmada benzetimlerin yanında, robot manipülatörlere ait denetleyicilerin doğrudan bir bilgisayar yerine özellikle denetim kartlarından oluştuğu düşünülerek bir manipülatör dizgesine doğrudan eklenebilecek şekilde, önerilen şemaların ve denetleyicinin bir sayısal işaret işleyiciye gömülmesi işlemi gerçekleştirilmiştir. Buna ek olarak gerçekliğe daha yakın olması açısından tüm şemanın gömülmesinin yanısıra robot manipülatör dizgesi bir bilgisayara, FDI ve FTC şemaları DSP üzerine gömülme işlemi gerçekleştirilmiştir.

Çizelge 2.1'de robot manipülatörlerde FDI üzerine yapılan çalışmalarda ve tez çalışmasında önerilen şemalarda hangi tip arızalar üzerinde durulduğu, hangi yöntemlerin kullanıldığı, belirsizlik ve gürültünün gözönünde bulundurulup bulundurulmadığı ve benzetim olarak ve gerçek manipülatör üzerinde çalışma yapıp yapılmadığı hakkında karşılaştırmalı bir gösterilim verilmiştir. Çizelge 2.2'de ise robot manipülatörlerde FTC üzerine yapılan çalışmalar ve tez çalışmasının bir karşılaştırılması eklenmiş, Çizelge 2.1'deki durumlara ek olarak aktif veya pasif FTC seçimi, uyarlanabilir denetleyici veya denetleyici değiştirme başlıkları incelenmiştir.

Çizelge 2.1. Tez çalışmasının robot manipülatörlerde FDI çalışmalarındaki yeri

Robot Manipülatörlerde FDI üzerine Yapılan Çalışmalar	Bileşen ve/veya algılayıcı arızaları	Kilitli ve/veya serbest sallanan eklemler arızaları	Kısmi eklemler arızaları (Yüzde ve sabit kavıp)	Analistik yöntemler	SA'da parametre güncelleme-uyarılabilir öğrenme	SA'da veri kümesi	Bulanık Mantık	Belirsizlik	Gürültü	Benzetim	Gercekleme
Caccavale ve Walker, 1997	✓		✓	✓				✓		✓	
Schneider ve Frank, 1996	✓			✓			✓	✓			✓
Leuschen, 2001; Leuschen ve ark., 2005	✓	✓		✓					✓	✓	✓
De Luca ve Mattone, 2004			✓	✓				✓			✓
Dixon ve ark., 2000		✓	✓	✓				✓	✓		✓
Abdul ve Liu, 2008			✓	✓				✓		✓	✓
Brambilla ve ark., 2008	✓		✓	✓				✓	✓	✓	✓
Chen ve Saif, 2008	✓	✓	✓	✓				✓		✓	
Naughton ve ark., 1996		✓				✓			✓	✓	
Vemuri ve Polycarpau, 1997	✓				✓			✓	✓	✓	
Terra ve Tinos, 2001	✓	✓				✓			✓	✓	✓
Tez Çalışması			✓			✓	✓	✓		✓	

Çizelge 2.2. Tez çalışmasının robot manipulatörlerde FTC çalışmalarındaki yeri

Robot Manipulatörlerde FTC üzerine Yapılan Çalışmalar	Bileşen ve/veya algılayıcı arızaları	Kilitli ve/veya serbest sallanan eklem arızaları	Kısmi eklem arızaları	Pasif FTC	Aktif FTC	Uyarlanabilir Denetleyici-Güncelleme	Denetleyici Değiştirme	Belirsizlik	Gürültü	Benzetim	Gerçekleme
Izumikawa ve ark., 2002, 2004	✓				✓		✓	✓		✓	
Tosunoğlu, 1995		✓			✓		✓	✓		✓	
Bonivento ve ark., 2004			✓	✓			✓	✓		✓	
Shin ve ark., 1999		✓		✓				?	?	✓	
Siqueira ve Terra, 2004		✓		✓					✓		✓
Zhang ve ark., 2009	✓				✓			✓		✓	
Kuntze ve ark., 2003		✓	✓		✓		✓	?	?	✓	
Tez Çalışması			✓		✓			✓		✓	

? işaretli konular hakkında ilgili çalışmalarda bilgi verilmemiştir.

### 3. TEZ ÇALIŞMASINDA KULLANILAN YAPAY ZEKA YÖNTEMLERİ

FDI ve FTC ile ilgili yapılan çalışmalarda doğrusal olmayan dizgelerin analitik olarak modellenmesindeki ve arıza yalıtım işleminde sınıflandırmadaki zorluklar, belirsizliklere ve gürültüye karşı gürbüzlük isteği gibi koşullar araştırmacıları kendi kendine otomatik öğrenen, doğrusal olmayan haritalama özelliklerine ve uyarlanma yeteneğine sahip yapay zeka yöntemlerine ve yapılarına itmiştir. Bu yöntemler sırasıyla

- Sinir Ağları-SA
- Bulanık Mantık-BM
- Genetik Algoritmalar-GA

Tez çalışmasının izleyen bölümünde önerilen şemalar tanıtılacaktır. Bu bölümde ise şemalar tanıtılmadan önce şemalarda kullanılan yapay zeka yöntemleri tanıtılacaktır. Bu yöntemler ise

- SA ve Esnek Yayılım Öğrenme (Resilient Propagation-RP Learning)
- Uyarlanabilir Sinirsel Bulanık Çıkarım Dizgesi (Adaptive Neuro Fuzzy Inference System-ANFIS) ve Çoklu-ANFIS (Multiple-ANFIS, M-ANFIS)

#### 3.1. SA ve Esnek Yayılım İle Öğrenme

SA gerçek yaşamda kullanım alanı olarak iki amaç için kullanılabilir :

- İşlev Yakınsama
- Örüntü Tanıma

İşlev yakınsama incelendiğinde SA'nın bir işlevin giriş değerlerine karşılık doğrusal olmayan bir haritalama yaparak uygun çıkış değerlerini üretmeye çalıştığı, örüntü tanıma incelendiğinde ise SA'nın verilen giriş değerlerini doğrusal olmayan yüzeylerle bir sınıflandırma işlemine soktuğu görülmektedir. SA'nın çeşitli yapılarına (İleri beslemeli SA, çapsal tabanlı işlev ağları, destek vektör makinaları, yinelemeli SA vb.) dair ayrıntılı bilgi Haykin'in kitabında bulunabilir (Haykin, 1999). Burada SA'nın yapısına ayrıntılı bir şekilde inilmeden ileri beslemeli SA'nda öğrenme için kullanılan ve en küçükleştirme yöntemlerinden Eğim Azalan (gradient descent) yöntemini kullanan Geriye Yayılım (Backpropagation-BP) algoritmasına seçenek olarak kullanılan Esnek Yayılım algoritması tanıtılacaktır.

SA'nın eğitimi ile amaçlanan kendilerine ait parametrelerin tanımlanan öğrenme algoritmalarıyla uygun girişlere karşılık uygun çıkışları vermesini sağlamaktır. Bu amaçla kullanılan en yaygın öğrenme algoritması BP algoritmasıdır ve  $E(n)$   $n$ . adımdaki

hata işlevi,  $w_{ij}$   $i$ . nöronun  $j$ . nörona ağırlık ve  $\eta$  öğrenme oranı parametresi olmak üzere eğitim azalan yöntemiyle Eş. (3.1) ile tanımlanmıştır (Haykin, 1999):

$$w_{ij(n+1)} = w_{ij}(n) + \Delta w_{ij}(n)$$

$$\Delta w_{ij}(n) = -\eta \cdot \frac{\partial E(n)}{\partial w_{ij}} \quad (3.1)$$

Eş. (3.1)'deki  $\partial E(n) / \partial w_{ij}$  terimi ilgili nöronun etkinleştirme (aktivasyon) veya geçiş işlevi olarak adlandırılan işlevine göre türevini içermektedir. Bu işlevler çoğunlukla sigmoid biçimli (tanjant sigmoid ve logsig) hiperbolik işlevlerdir.  $\eta$  öğrenme oranı ise hatanın türevine göre değişim adımının büyüklüğünü ayarlar. Bu değer ağırlık istenen en küçük hata değerine yakınsamasında küçük seçildiğinde çok fazla sayıda işlem adımı gerektirdiği, büyük seçildiğinde istenen en küçük değere erişemeyip etrafında salınım yapacağı bilinmektedir (Haykin, 1999). Bu zayıflıktan kurtulmak ve yakınsamayı hızlandırmak amacıyla Eş. (3.1)'e  $\mu$  momentum parametresi olmak üzere momentum terimi eklenmiştir :

$$\Delta w_{ij}(n) = -\eta \cdot \frac{\partial E(n)}{\partial w_{ij}} + \mu \cdot \Delta w_{ij}(n-1) \quad (3.2)$$

Eklenen momentum terimine karşın yapılan düzeltmenin yine seçilen  $\mu$  momentum parametresine bağlı olduğu görülmüştür. Bu gözlemler sonucunda öğrenme sırasında uyarlanabilir öğrenme oranı yapılması düşünülmüştür.

Bu tip düzeltmeler yapılırken gözardı edilen ağırlık güncellemesinin sadece öğrenme oranına değil, ayrıca  $E(n)$ 'nin  $w_{ij}$ 'ye göre kısmi türevine de bağlı olduğudur. Esnek yayılım, ağırlık güncellemesini bu bulanık uyarlamadan kurtararak kısmi türevin genliğine bakmadan doğrudan yapmaktadır (Riedmiller ve Braun, 1993). Esnek yayılım algoritmasında öncelikle tüm ağırlıklara  $\Delta_{ij}$  şeklinde rasgele güncelleme değeri atanır. Bu değer ise aşağıdaki gibi güncellenir:

$$\Delta_{ij}(n) = \begin{cases} \eta^+ \cdot \Delta_{ij}^{(n-1)} & \text{eger } \frac{\partial E(n-1)}{\partial w_{ij}} \cdot \frac{\partial E(n)}{\partial w_{ij}} > 0 \\ \eta^- \cdot \Delta_{ij}^{(n-1)} & \text{eger } \frac{\partial E(n-1)}{\partial w_{ij}} \cdot \frac{\partial E(n)}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(n-1)} & \text{diger} \\ 0 < \eta^- < 1 < \eta^+ \end{cases} \quad (3.3)$$

Bu tanımlama her bir ağırlığa karşılık gelen kısmi türevin işareti değiştiğinde güncellenmenin çok büyük olduğunu, yerel bir en küçüğü atladığını, bu yüzden güncelleme değerinin  $\eta^-$  çarpanıyla azaltıldığını, işaret aynı kalıyorsa yakınsamayı hızlandırması için güncelleme katsayısının  $\eta^+$  çarpanıyla arttırıldığını söylemektedir. Her bir ağırlık için güncelleme değeri hesaplandıktan sonra ağırlık güncellemesi Eş. (3.4) gibi hesaplanır :

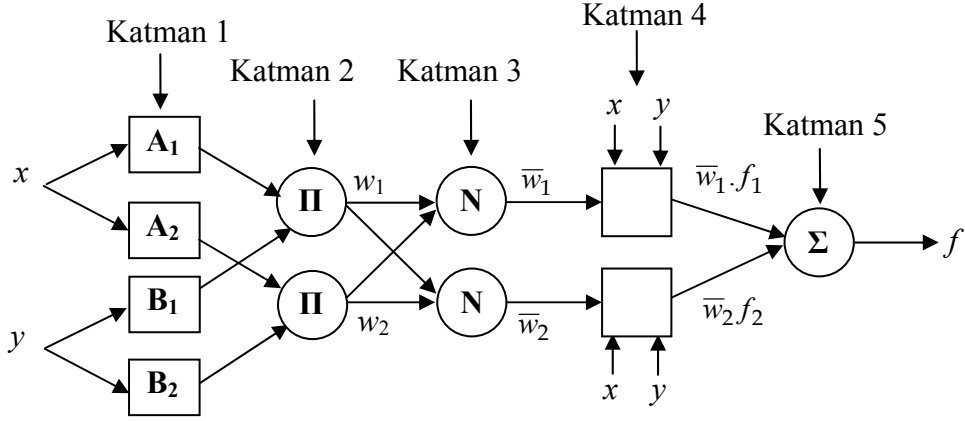
$$\Delta w_{ij}^{(n)} = \begin{cases} -\Delta_{ij}^{(n)} & \text{eger } \frac{\partial E(n)}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(n)} & \text{eger } \frac{\partial E(n)}{\partial w_{ij}} < 0 \\ 0 & \text{diger} \end{cases} \quad (3.4)$$

Anlatılan bu süreçle tanımlanan Esnek Yayılım Öğrenme algoritması diğer öğrenme algoritmalarının aksine kısmi türevlerin sadece işaret değerlerini kullanmasıyla daha şeffaf ve güçlü bir uyarılma sürecine sahiptir ve zaman ve bellek tüketimi açısından etkin bir algoritmadır (Riedmiller ve Braun, 1993).

### 3.2. ANFIS ve M-ANFIS

Doğrusal olmayan dizgelerin analitik olarak modellenmesindeki zorluklar, belirsizliklere ve gürültüye gürbüzlük araştırmacıları analitik yöntemlerden kendi kendine öğrenen ve doğrusal olmayan haritalama özelliğine sahip yapılara doğru itmiştir. SA bu özellikleriyle birçok farklı çeşite ve öğrenme algoritmasına sahip olsa da doğasından gelen kendi kendine öğrenmenin temeliyle siyah kutu şeklinde davranış sergilemektedir. Yine bu yapılardan BM ise modelleme sırasındaki tüm parametrik ayarlamaları kullanıcıya bırakmakta ve kullanıcı tecrübesini bir parametre haline getirmektedir. Her iki yapının olumlu yanları Jang ve arkadaşları tarafından görülmüş ve ANFIS denen ve işlevsel olarak bulanık çıkarım dizgesine (fuzzy inference system) denk olan uyarlanabilir bir ağda birleştirilmiştir (Jang ve ark., 1997).

ANFIS, 5 katmanlı bir yapıya sahiptir ve Şekil 3.1'de iki girişli ve tek çıkışlı bir ANFIS örnek olarak verilmiştir. Verilen örnekteki giriş sayısı tamamen rasgele seçilmiştir. Şekil 3.1'in devamında katmanların işlevleri ve matematiksel tanımları verilmiştir.



Şekil 3.1. ANFIS'in genel yapısı

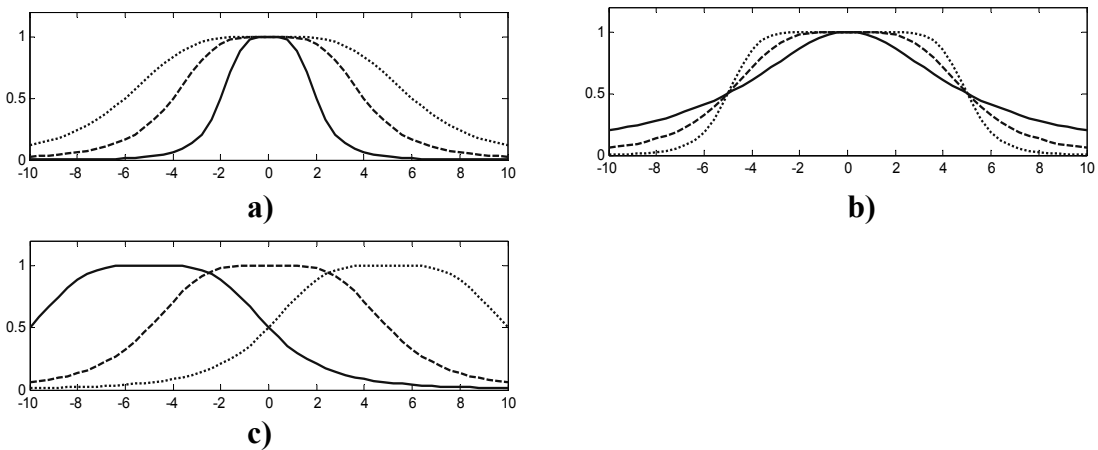
**Katman 1:** Bu katman girişlere ait üyelik işlevlerini barındırır ve tüm girişler bu işlevlere uygulanır.

$$\begin{aligned} K1x_i &= \mu_{A_i}(x) \quad i = 1,2 \\ K1y_i &= \mu_{B_i}(y) \quad i = 1,2 \end{aligned} \quad (3.5)$$

$\mu(\cdot)$  ile gösterilen üyelik işlevlerinin tipi ve şekli kullanıcı tarafından belirlenir. Çoğunlukla bu işlev için genelleştirilmiş-çan şekli tercih edilir, diğer üyelik işlevleri ile ilgili ayrıntılı bilgi Jang ve arkadaşlarının çalışmasında bulunabilir (Jang ve ark., 1997):

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_{A_i}}{a_{A_i}} \right|^{2b_{A_i}}}, \quad \mu_{B_i}(y) = \frac{1}{1 + \left| \frac{y - c_{B_i}}{a_{B_i}} \right|^{2b_{B_i}}} \quad i = 1,2 \quad (3.6)$$

Burada  $a_i$  tanımlanan üyelik işlevinin genişliği,  $c_i$  merkezi ve  $b_i$  iniş eğimidir ve bu parametreler öncül (premise) parametreler olarak adlandırılır. Bu işleve  $-10 \leq x \leq 10$  için sırasıyla  $(a_i, b_i, c_i)$  değişimlerinin etkileri Şekil 3.2'de gösterilmiştir.



Şekil 3.2.  $(a_i, b_i, c_i)$  parametrelerinin genelleştirilmiş-çan üyelik işlevine etkileri  
**a)**  $a_i$  değişiminin etkisi **b)**  $b_i$  değişiminin etkisi **c)**  $c_i$  değişiminin etkisi

**Katman 2:** Katman 1'den gelen her bir işlev değeri tanımlanmış kurallar ve kural tabanı doğrultusunda diğer işlev değerleriyle çarpılır ve ortaya çıkan değerler her bir kuralın ateşleme gücü (firing strength) olarak adlandırılır.

$$K2_i = w_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \quad i = 1,2 \quad (3.7)$$

Kural tabanına bağlı Eş. (3.7) ile gösterilen çarpım tamamen kullanıcı tarafından belirlenir. Şekil 3.1'deki gösterilimde Eş. (3.7) ile verilen çarpımların sadece aynı indisli çarpımlar olduğuna dikkat edilmelidir ve bu seçim kullanıcı tarafından yapılmıştır.

**Katman 3:** Bu katmanda ateşleme açıları normalize edilir.

$$K3_i = \bar{w}_i = \frac{w_i}{\sum_{j=1}^2 w_j} \quad i = 1,2 \quad (3.8)$$

**Katman 4:** Normalize edilmiş ateşleme güçleri birinci dereceden bir işlev ile çarpılır.

$$K4_i = \bar{w}_i \cdot f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad i = 1,2 \quad (3.9)$$

Burada  $(p_i, q_i, r_i)$  birinci derece işleve ait katsayılarıdır ve bu parametreler sonuç (consequent) parametreleri olarak adlandırılır.

**Katman 5:** Katman 4'ten gelen tüm değerler toplanır ve çıkış değeri elde edilir.

$$f = K5 = \sum_{i=1}^2 \bar{w}_i f_i \quad (3.10)$$

ANFIS'te bulanık mantıktaki gibi kullanıcı tecrübesinin kullanıldığı yerler Katman 1 ve Katman 2'deki seçimlerdir. Katman 1'de tanımlanan üyelik işlevlerinin tipi ve Katman 2'de tanımlanan kural tabanına bağlı çarpma işlemi tamamen kullanıcıya bağlıdır. Jang ve arkadaşlarının çalışmasında bu bağımlılık ANFIS'in çeşitli koşullar altında bulanık mantık çıkarımına denklik gösteriliminde net olarak görülmektedir (Jang ve ark., 1997).

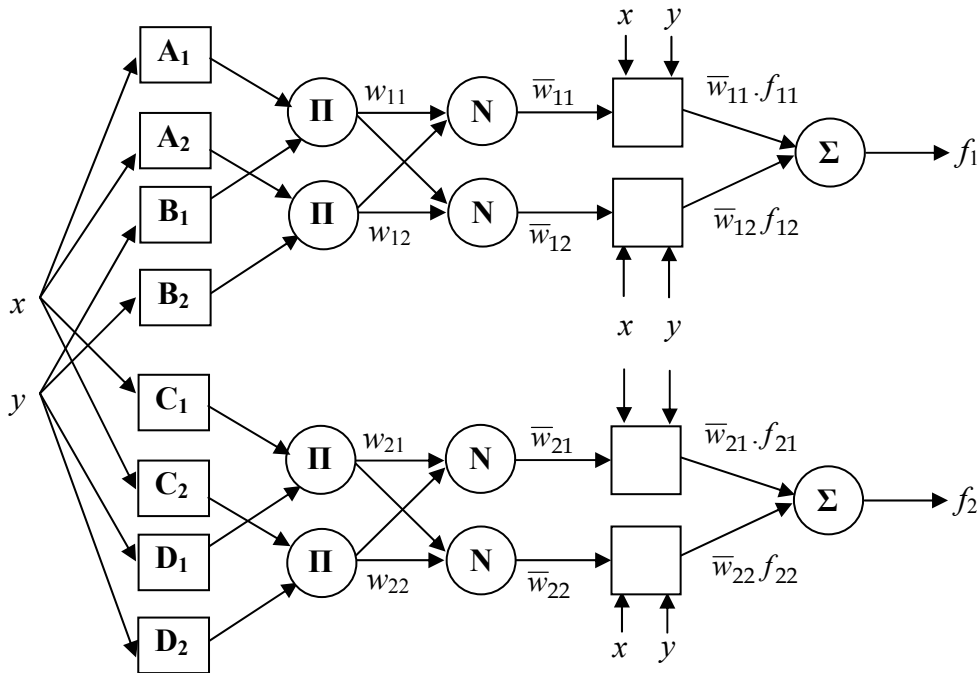
Tüm ağ yapılarında amaçlanan uygun girişlere karşılıklı uygun çıkışları sağlayacak şekilde ağ parametrelerinin güncellenmesi veya uyarlanmasıdır. Bu yönüyle ANFIS de SA gibi öğrenme algoritmaları kullanarak kendine ait parametrelerini uygun şekilde güncellemektedir. Öğrenme algoritması olarak istenirse geriye yayılım istenirse ileri ve

geri yönlü geçişlerle ifade edilen melez öğrenme (hybrid learning) tercih edilebilir. Çizelge 3.1 melez öğrenmeyi açıklamaktadır. İleri yönlü geçişte girişler öncül parametreler sabit tutularak Katman 4'e kadar ilerler ve en küçük kareler (least square) yöntemi ile çıkış parametreleri belirlenir. Geri yönlü geçişte ise sonuç parametreleri sabitlenir. Hata Katman 1'e kadar geri yayılır ve öncül parametreler Eğitim Azalan yöntemi ile hesaplanır. Melez öğrenmeye ait ayrıntılı bilgi Jang ve arkadaşlarının çalışmasında bulunabilir (Jang ve ark., 1997).

**Çizelge 3.1.** Melez öğrenme için iki yönlü geçişte parametrelerin hesaplanması

Parametre	İleri yönlü geçiş	Geri yönlü geçiş
Öncül	Sabit	Eğim azalan
Sonuç	En küçük kareler	Sabit

Şekil 3.1 ANFIS'in yapısından gelen ve temel zayıflığı olan çok-giriş-tek-çıkışlı (MISO) olma özelliğini açıkça göstermektedir. Bu sebeple çok-giriş-çok-çıkışlı (MIMO) dizgelerin modellenmesinde birbirinden bağımsız parametrelere ve çıkışlara sahip Çoklu-ANFIS (M-ANFIS) yapısı kullanılır. M-ANFIS yapısına ait 2 giriş - 2 çıkışlı bir örnek Şekil 3.3'de verilmiştir.



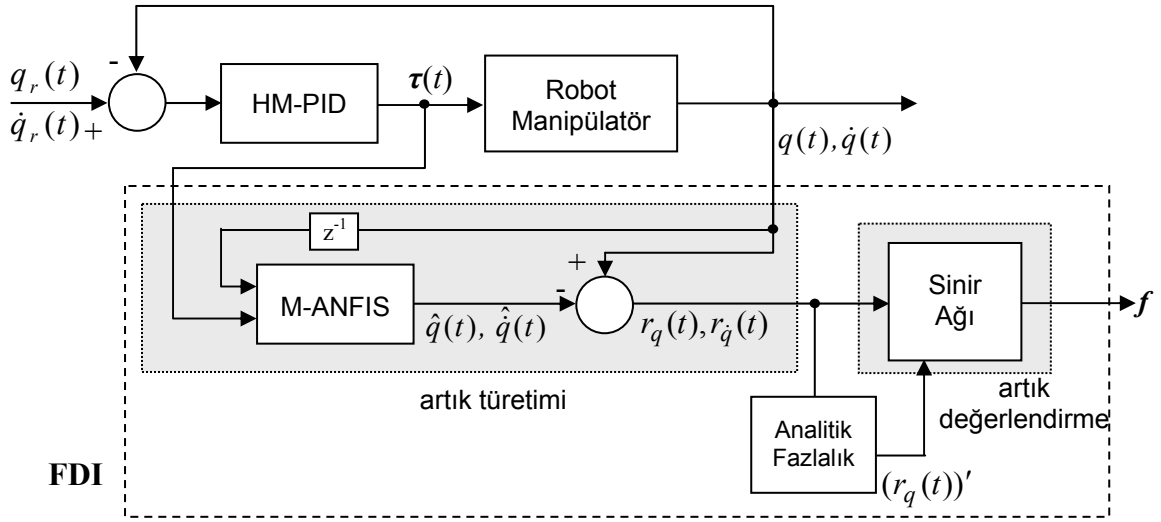
**Şekil 3.3.** M-ANFIS'in genel yapısı

MIMO dizgelerin modellenmesinde M-ANFIS'den başka çıkışlardaki ilintileri de göz önüne alan ve artan parametre sayısının getirdiği yükü azaltmak amacıyla bir takım ortak parametreler kullanan Coactive-ANFIS (C-ANFIS) yapısı da bulunmaktadır (Jang ve ark., 1997). Bu çalışmada ise benzetim için kullanılan MATLAB, M-ANFIS yapısına izin verdiğinden ve Jang ve arkadaşlarının kitabında C-ANFIS ile ilgili yeterince bilgi bulunmadığından M-ANFIS tercih edilmiştir.

#### 4. ÖNERİLEN FDI VE FTC ŞEMALARI

Bu bölümde önerilen FDI ve FTC şemaları tanıtılacaktır. Şemaların herbirinin kendine özgü özellikleriyle ortak olan özellikleri şemaya ait altbölümde verilecektir. İzleyen bölümde ise şemaların benzetimle nasıl gerçekleştirildiği ve elde edilen bulgular verilecektir. Şemaların tanıtımında genelleştirilmiş manipülötör tanımı kullandığından şemalarda varolan işaretlere ait vektörel gösterimler gözardı edilmiştir.

##### 4.1. Önerilen 1. FDI Şeması - Robot Manipülötörlerde M-ANFIS ve SA Kullanarak Arıza Bulma ve Yalıtımı

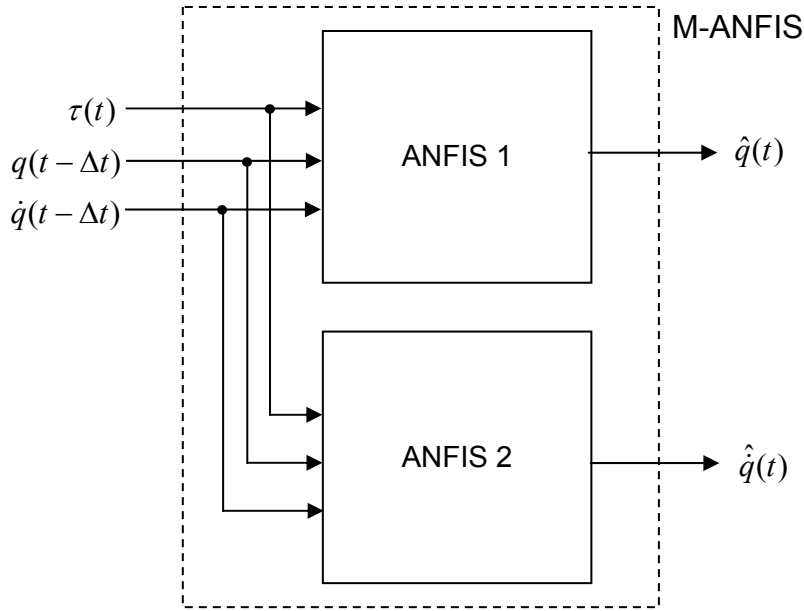


Şekil 4.1. Önerilen 1. FDI şeması

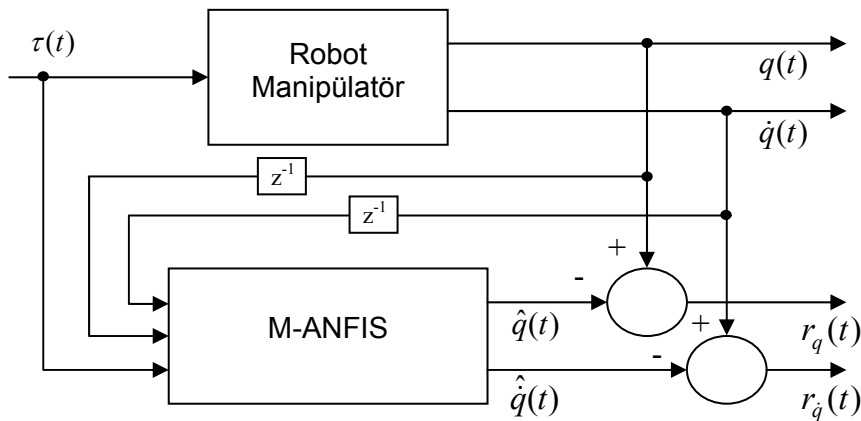
Önerilen ilk şemaya ait blok diyagram Şekil 4.1'de verilmiştir. Blok diyagram tüm dizgeyi göstermek amacıyla robot manipülötör denetim bloğunu da içermektedir. Robot manipülötör denetimi için Hesaplanmış Moment-PID (HM-PID) yöntemi kullanılmış ve bu yöntem hakkında ayrıntılı bilgi Ek A'da verilmiştir (Lewis ve ark., 1993). FDI şemasının ilk bloğunu oluşturan M-ANFIS bloğunda çok sayıda bağımsız ANFIS yapısı bir araya getirilerek robot manipülötör modelleme gerçekleştirilmiştir. Gerçek robot manipülötör ile model arasındaki farktan yararlanılarak artık türetimi gerçekleştirilmiştir. Artık türetiminden sonra analitik fazlalık tanımından yararlanılarak fazlalık bir artık işareti türetilmiştir. Arıza bulma işlemi ise türetilen fazlalık işareti dahil türetilen artıklardan herhangi birinin daha önceki benzetimlerden elde edilmiş sağlıklı durumdaki artık eşik değerlerinin aşımı olarak kabul edilmiştir. Arıza yalıtım işlemi ise artıkların çok katlı bir sinir ağına verilerek sınıflandırılması ile gerçekleştirilmiştir.

#### 4.1.1. M-ANFIS İle Artık Türetimi/Arıza Bulma

Robot modelleme için kullanılan M-ANFIS bloğu giriş olarak anlık moment değerlerini ( $\tau(t)$ ) ve bir önceki örnekleme anında ( $t - \Delta t$ ) her bir ekleme ait açısıl konum ve hız değerlerini ( $q(t - \Delta t), \dot{q}(t - \Delta t)$ ) giriş olarak almakta, her bir ANFIS ile her bir eklem için anlık açısıl konum ve hız değerlerinin kestirim değerlerini ( $\hat{q}(t), \hat{\dot{q}}(t)$ ) çıkış olarak vermektedir. Örnek olarak tek eklemlerli bir manipülatör için bu durum Şekil 4.2’de açık bir şekilde gösterilmiştir. Robot manipülatörün modeli oluşturulduktan sonra modele ait açısıl konum ve hız kestirim işaretleri ile gerçek robot manipülatör açısıl konum ve hız işaretleri arasındaki fark alınarak artık türetim işlemi gerçekleştirilmiştir. Artık türetim işlemi ise Şekil 4.3’de verilmiştir. İzleyen bölümde benzetimler sırasında oluşturulan M-ANFIS bloğuyla ilgili ayrıntılı bilgiler verilecektir.



Şekil 4.2. M-ANFIS ile tek eklemlerli robot manipülatör modelleme



Şekil 4.3. M-ANFIS ile artık türetimi

Tez çalışmasındaki benzetimler ve diğer bazı çalışmalar sadece elde varolan artık işaretleriyle arızaların hem bulunmasının hem de yalıtımının zor olduğunu göstermiştir (Chen ve Patton, 1999; Leuschen, 2001; Leuschen ve ark., 2005). Bu durum benzetim çalışmalarında sinir ağı ile arıza yalıtımında elde edilen düşük başarımlar yüzdeleri olarak ortaya çıkmıştır. Bu zorluktan kurtulmak amacıyla diğer çalışmalarda analitik fazlalık tanımından yararlanılarak arızalara duyarlılığı analitik olarak kanıtlanmış fazladan işaretler türetilmiştir (Leuschen, 2001; Leuschen ve ark., 2005). Bu çalışmada herhangi bir analitik yöntem kullanılmadığından doğrudan fazladan bir işaret türetilmiş ve bu işaret arıza bulma ve arıza yalıtımı için doğrudan kullanılmıştır.

Bu amaçla matematiksel bir değişkeni iki veya daha fazla yoldan belirleme olarak tanımlanan analitik fazlalık tanımından yola çıkarak konum artıklarının türev değerlerinin matematiksel olarak hız artıklarına denk olacağı düşüncesiyle bu artıkların türev değerleri elde edilmiş ve kullanılmıştır. Artıkların ve analitik fazlalıkların elde edilmesiyle ilgili eşitlikler sırasıyla Eş. (4.1) ve (4.2)'de verilmiştir.

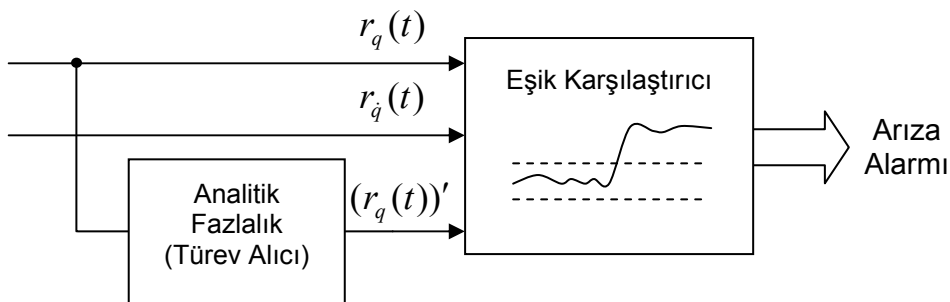
$$r_q(t) = q(t) - \hat{q}(t) \quad (4.1)$$

$$r_{\dot{q}}(t) = \dot{q}(t) - \hat{\dot{q}}(t)$$

$$(r_q(t))' = (q(t) - \hat{q}(t))' = \dot{q}(t) - \hat{\dot{q}}(t) \quad (4.2)$$

Elde edilen artıklar ve analitik fazlalığın herhangi bir anda sağlıklı durum için belirlenmiş alt ve üst eşikleri aşması arıza belirteci olarak kabul edilir ve şemada arıza bulma - arıza alarmı olarak görev yapmaktadır. Bu duruma ait koşul Eş. (4.3)'te, artık ve analitik fazlalıklar ile arıza bulmaya ait gösterilim Şekil 4.4'te verilmiştir.

$$\left. \begin{aligned} (r_q(t) > \text{üst\_e}_{r_q}) \vee (r_q(t) < \text{alt\_e}_{r_q}) \vee \\ (r_{\dot{q}}(t) > \text{üst\_e}_{r_{\dot{q}}}) \vee (r_{\dot{q}}(t) < \text{alt\_e}_{r_{\dot{q}}}) \vee \\ (r_q(t))' > \text{üst\_e}_{(r_q)'} \vee (r_q(t))' < \text{alt\_e}_{(r_q)'} \end{aligned} \right\} \Rightarrow \text{Arıza Alarmı} \quad (4.3)$$



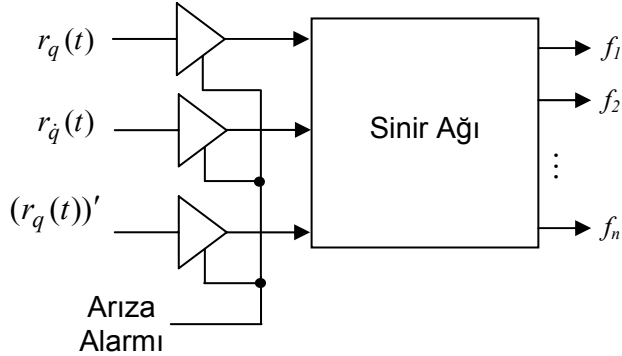
Şekil 4.4. Artıklar ve analitik fazlalıklar ile arıza bulma

#### 4.1.2. Sinir Ağı İle Artık Değerlendirme/Arıza Yalıtımı

Artık üretim işleminden sonra arızaları birbirinden ayıran arıza yalıtım işlemi gelir. Bu işlem farklı tipte arızaların farklı tipte artık karakteristiği göstermesi ilkesine dayanır ve bir sınıflandırma işlemi olarak kabul edilir. Sınıflandırma ve örüntü tanıma için kullanılabilen tüm teknikler bu işlem için de kullanılabilir. Tez çalışmasında önerilen bu 1. şemada bu işlem için RP öğrenme algoritmasını kullanan bir sinir ağı tercih edilmiştir.

Arıza alarm işaretleriyle devreye giren sinir ağı, artık ve analitik fazlalık işaretlerini giriş olarak almaktadır. Sinir ağı, bu işaretlerin arıza durumlarındaki benzetimlerle elde edilmiş ve gerekli sinir ağı çıkış tanımlamaları (her bir arıza için tanımlı çıkış 1, diğer çıkışlar 0) yapılmış giriş-çıkış işaret kümesi kullanılarak ve hata hedefi belirlenerek uygun şekilde eğitilir.

Eğitiminden sonra sinir ağı kullanıma hazır duruma gelir. Arıza bulma alarmı ile etkin hale gelen sinir ağı ile arızaların yalıtımına ait gösterilim Şekil 4.5'te verilmiştir.



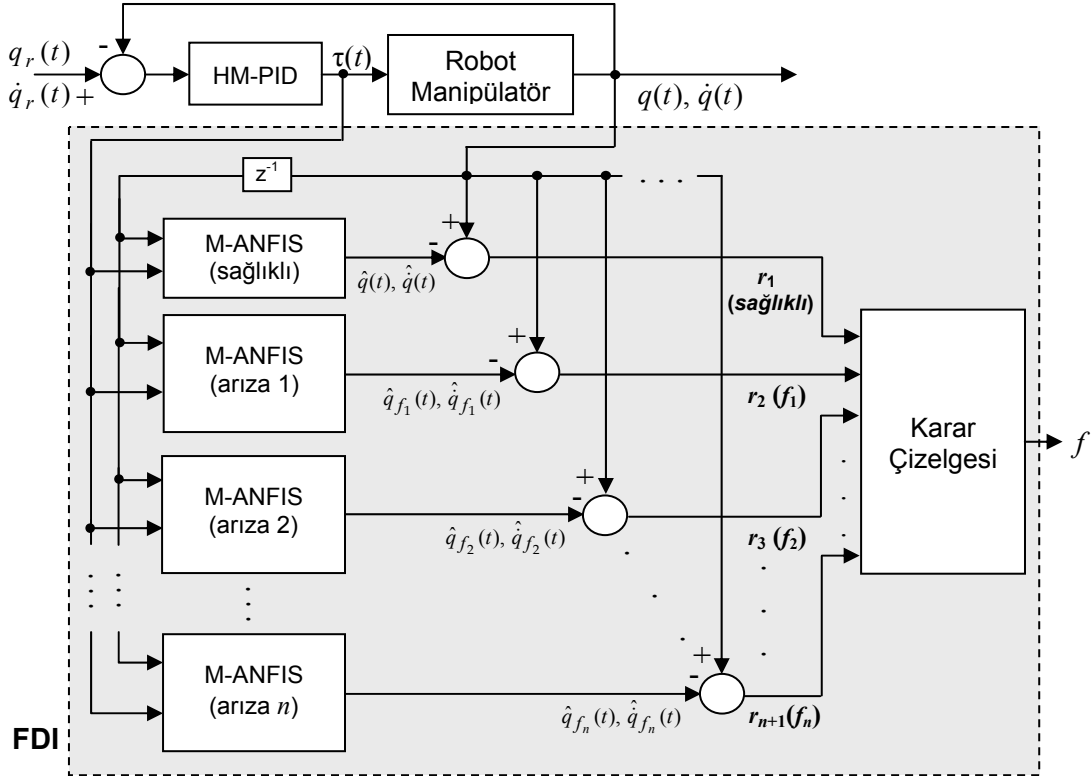
Şekil 4.5. Sinir ağı ile arıza yalıtımı

#### 4.2. Önerilen 2. FDI Şeması - Robot Manipülatörlerde M-ANFIS'li Genelleştirilmiş Gözlemcileri Kullanarak Arıza Bulma ve Yalıtımı

Gözlemcileri kullanarak arıza bulma ve yalıtımında kullanılan tasarım yordamı artıklarla arızalar arasında tanımlanan ilişkiler üzerine kuruludur. Eğer artıkların tümünü tanımlayan artık kümesi tüm arızaları yalıtabiliyorsa gerekli arıza yalıtım özelliği vardır denir ve arıza yalıtım özelliğini sağlamak amacıyla tasarımında iki yol izlenir. Atanmış gözlemci şemaları (Dedicated Observer Scheme-DOS) olarak adlandırılan birinci yolda türetilen her bir artık bir arızaya duyarlı, diğerlerine karşı duyarsızdır. Bu tür artıklar kulağa hoş gelse de tasarımı ve modelleme hatalarına gürbüzlük gibi özellikleri analitik olarak sağlamanın zor olduğu literatürde belirtilmiştir (Chen ve Patton, 1999). Bu yüzden tasarımda daha fazla kullanılan ve her bir artığın bir

arıza hariç geri kalan tüm arızalara duyarlı olduğu genelleştirilmiş gözlemci şemaları (Generalized Observer Scheme-GOS) tercih edilir. GOS yaklaşımında her bir arızalı dizge durumu tek tek modellenir ve bunlardan yararlanılarak artıklar ve artık kümesi oluşturulur. Artık kümesinin yorumlanmasıyla arıza bulma ve yalıtım işlemleri aynı anda gerçekleştirilmiş olur. Bu tip şemaların oluşturduğu artıklara ait gösterim Şekil 1.8’de verilmiştir.

Bu tip şemalarda her bir dizge durumu tek tek modellendiği için belirsizliklerle baş etmek daha kolaydır fakat her bir durumun tek tek modellenmesi ve bunun getirdiği işlemsel yük bu şemaların gerçek zamanlı olarak çalışmasını zorlaştırmaktadır. Bunun yanında arıza bulma ve yalıtım işlemlerinin aynı anda yapılması ve yalıtım işlemi için ayrıca herhangi bir işleme gerek duyulmaması bu şemaları çekici duruma getirmektedir.



Şekil 4.6. Önerilen 2. FDI şeması

Şekil 4.6’da GOS yaklaşımı ile önerilen 2. FDI şemasının blok diyagramı verilmiştir. Dizgenin denetim kısmı yine HM-PID tarafından gerçekleştirilmektedir. Robot manipülatör için bir sağlıklı ve  $n$  arıza durumu olmak üzere toplam  $n + 1$  durum tanımlanmıştır. Tanımlanan bu durumların her biri birer M-ANFIS tarafından modellenmiştir. Bu modelleme işleme 1. şemada anlatılan modelleme işlemiyle aynı

şekilde yapılmaktadır.

Modelleme işlemini artık türetim işlemi izlemektedir. Bu işlem de 1. şema ile aynı şekildedir ve her bir duruma ait açısasal konum ve hız artıkları türetilmektedir.

Herhangi bir arızalı durumda sağlıklı duruma ait artıklar 1. şemayla benzer şekilde kendileri için tanımlanan eşik seviyelerini aşmakta ve böylece arıza bulma işlemi gerçekleştirilmektedir.

Arızalı robot manipülatörün ürettiği açısasal konum ve hız işaretleri, eşleşen duruma ait kestirim işaretleri ile uyuşmakta, böylece varolan duruma ait artıklar sıfır değerini vermektedir. Bu şekilde GOS yaklaşımıyla tasarlanan şemalarda arıza bulma ve yalıtım işlemleri aynı anda Çizelge 4.1'e bağlı olarak yapılmaktadır. Bu çizelge ise Şekil 4.6'da karar çizelgesi bloğu olarak adlandırılan blok ile temsil edilmektedir.

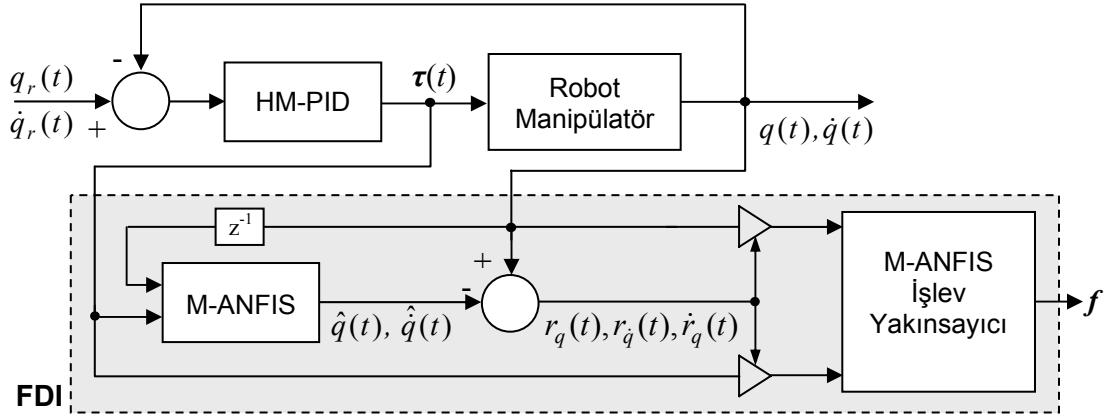
**Çizelge 4.1.** Karar çizelgesi

Arızalar	$r_1$ (sağlıklı)	$r_2 (f_1)$	$r_3 (f_2)$	...	$r_{n+1} (f_n)$
Sağlıklı	0	1	1	1	1
$f_1$	1	0	1	1	1
$f_2$	1	1	0	1	1
...	1	1	1	0	1
$f_n$	1	1	1	1	0

### 4.3. Önerilen 3. FDI Şeması - Robot Manipülatörlerde M-ANFIS İle Arıza İşlevine Yakınsama

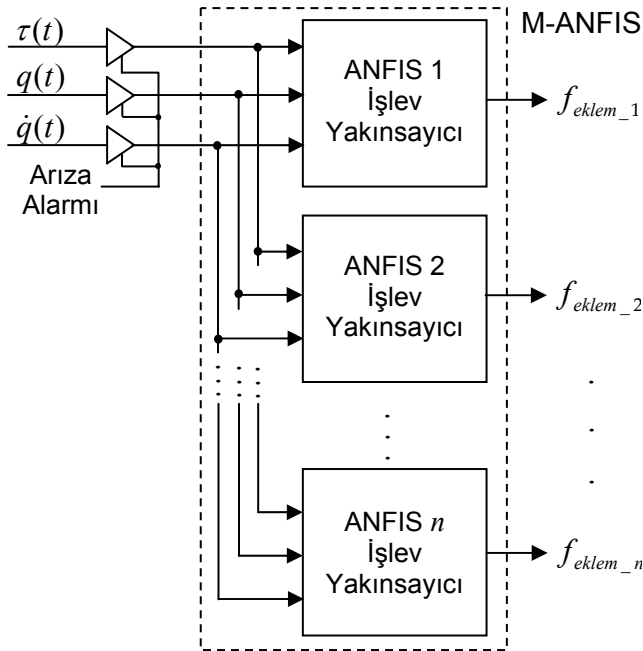
Önerilen 1. ve 2. şema, arıza bulma ve yalıtımı için geçerli ve başarılı şemalar olsa da tanımları gereği sadece tanımlanmış arızaların yalıtımı için kullanılabilirler ve bu da onları daha önce tanımlanmamış herhangi bir yüzdeyle oluşabilecek kısmi hareketlendirici arızalarına karşı zayıf duruma düşürmektedir. Ayrıca arıza bulma ve yalıtımından gelen bilgiyi kullanan FTC tasarımı, kesin bir arıza işlev bilgisine gerek duymasından dolayı 1. ve 2. şema tam başarımla elde edemez. Bu iki sebepten dolayı önerilen 3. şemada arıza işlevine yakınsama amaçlanmıştır.

Şekil 4.7'de önerilen ve arıza işlevine yakınsama için tasarlanmış 3. şema görülmektedir. Şemada modelleme, artık türetimi ve arıza bulma işlemleri 1. şema ile aynı şekilde yapılmaktadır.



Şekil 4.7. Önerilen 3. FDI şeması

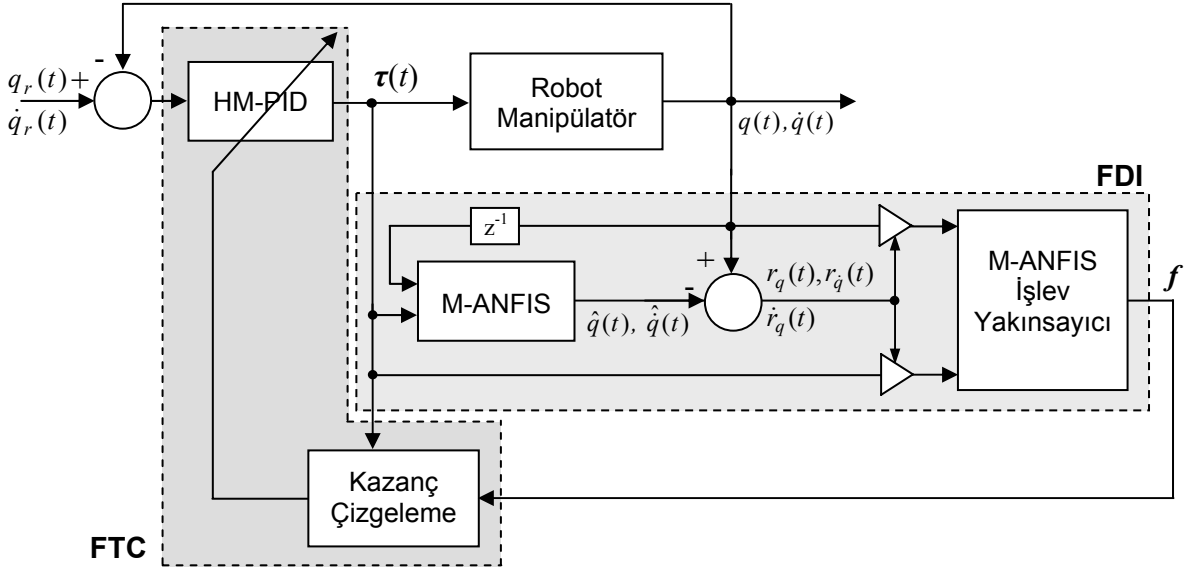
Arıza bulmayı izleyen arıza yalıtım işlemi ise farklı bir şekilde yapılmaktadır. Şemada ilk olarak arıza işlevine yakınsamak için yakınsayıcı yapıya giriş olarak artıklar ve analitik fazlalıklar verilmiş fakat istenen yakınsama başarımı elde edilememiştir. Daha sonra bunun yerine varolan anlık moment, açısal konum ve hız işaretleri giriş olarak uygulanmış ve arıza işlevine yakınsanmaya çalışılmıştır. Arıza işlev yakınsayıcı olarak önce bir sinir ağı denenmiş, yeterli arıza yalıtım başarımı elde edilemediğinden daha sonra bir M-ANFIS yapısı tercih edilmiştir. Girişler, arıza bulma işlemiyle birlikte etkin duruma geçmekte ve işlev yakınsayıcı çıkış olarak her bir ekleme ait arıza işlevini moment değeri olarak vermektedir. Bu durum Şekil 4.8’de açık bir şekilde gösterilmiştir.



Şekil 4.8. M-ANFIS ile arıza işlevine yakınsama

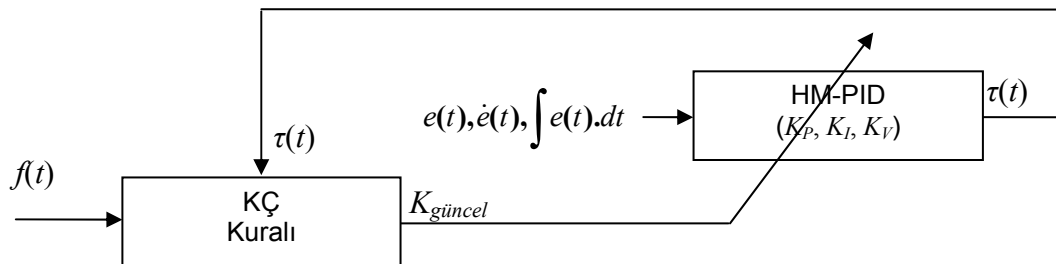
#### 4.4. Önerilen FTC Şeması - Robot Manipülörlerde Kazanç Çizgeleme İle Arızaya Dayanlı Denetim

Tez çalışmasında önerilen FDI şemalarına ek olarak aktif bir FTC şeması önerilmiş, bu şema Şekil 4.9’da verilmiştir.



Şekil 4.9. Önerilen FTC Şeması

Önerilen bu şema, 3. FDI şemasının bir uzantısı olarak tasarlanmıştır. Arızaya dayanlı denetleyici tasarımı için Kazanç Çizgeleme (KÇ) yöntemi temel alınmıştır. Bu yöntemde göre dizgeyi denetleyen denetleyici parametreleri dizge durumuna göre önceden tanımlanmış herhangi bir kural veya kurallar kümesiyle veya bir çizelgeyle veya tanımlanan BM yapısı ile değiştirilebilir (Zhao ve ark., 1993). 3. şemadaki arızalı ekleme ait işlev çıkışı ve anlık uygulanan moment değerleri bir kazanç çizgeleme bloğuna uygulanarak HM-PID denetim yöntemindeki  $K_p$ ,  $K_i$ ,  $K_v$  katsayıları tanımlanan belirli bir kurala göre güncellenmektedir. Bu kuralın tez çalışmasındaki tanımı Bölüm 5'teki benzetim çalışmalarında verilecektir. KÇ yöntemi ile katsayıların nasıl güncellendiği Şekil 4.10'da gösterilmiştir.



Şekil 4.10. KÇ yöntemi ile arızaya dayanlı denetim

Tanımlanan güncelleme kuralı ise üç sorunu birlikte getirmektedir. Birinci sorun güncellemenin yani FTC'nin etkileridir. Katsayı güncellemesinin getirdiği geçici durumdaki aşımın yanında sürekli durum hatalarındaki düşüm arasında kullanıcı bir seçim yapmak zorundadır.

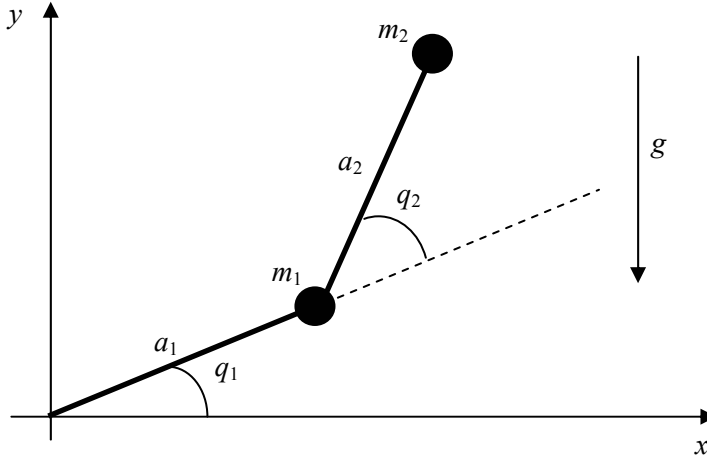
İkinci sorun katsayı güncellemesinin hangi sıklıkta yapılacağıdır. Güncellemeler istenirse arıza yalıtımından sonra 1 kez, istenirse belirenen saniye aralıklarında yapılabilir. Bu noktadaki seçimi her katsayı değişimindeki yerleşmeye bağlı hata miktarındaki değişim (aşım) ve yerleşme zamanı belirlemektedir.

Ortaya çıkan üçüncü sorun ise katsayı değişiminin ani olarak gerçekleştirilmesine bağlı hata miktarındaki artıştır. Bu durumu değiştirmek amacıyla katsayıları zamana bağlı olarak istenen değere götüren bir PID değişim kuralı tanımlanması düşünülmüş ve gerçekleştirilmiştir.

Ortaya çıkan bu sorunlar ve etkileri izleyen bölümde ayrıntılı olarak incelenmiştir.

## 5. BENZETİM ÇALIŞMALARI ve BULGULAR

Bu bölümde Bölüm 4'te önerilen şemalara ait yapılmış benzetim çalışmaları ve elde edilen bulgular ayrıntılı biçimde verilecektir. Önerilen şemaların benzetimi MATLAB 6.5 altında *Fuzzy Logic Toolbox* ve *Neural Network Toolbox* kullanılarak gerçekleştirilmiştir. Önerilen şemalarda kullanılan robot manipülatör, Lewis ve ark. çalışmalarında tanımladıkları iki eklemlilik, düzlemsel, yer çekimi altında ve ağırlıkları her uzvun sonunda tanımlanmış bir robot manipülatördür (Lewis ve ark., 1993). Bu manipülatörün gösterilimi Şekil 5.1'de verilmiştir.



Şekil 5.1. İki eklemlilik, düzlemsel, yerçekimi altındaki robot manipülatörün gösterilimi (Lewis ve ark, 1993)

Bu manipülatöre ait dinamik eşitlikleri ve dinamiklerinin genelleştirilmiş gösterilimi sırasıyla Eş.(5.1) ve (5.2)'de verilmiş, açısal konum ve hız işaretlerine ait vektörel gösterimler gözardı edilmiştir:

$$\begin{aligned}
 M_{11} &= (m_1 + m_2)a_1^2 + m_2a_2^2 + 2m_2a_1a_2 \cos(q_2(t)) \\
 M_{12} &= M_{21} = m_2a_2^2 + m_2a_1a_2 \cos(q_2(t)), \quad M_{22} = m_2a_2^2 \\
 V_1 &= -m_2a_1a_2(2\dot{q}_1(t)\dot{q}_2(t) + \dot{q}_2(t)^2) \sin(q_2(t)) \\
 V_2 &= m_2a_1a_2(\dot{q}_1(t))^2 \sin(q_2(t)) \\
 G_1 &= (m_1 + m_2)ga_1 \cos(q_1(t)) + m_2ga_2 \cos(q_1(t) + q_2(t)) \\
 G_2 &= m_2ga_2 \cos(q_1(t) + q_2(t))
 \end{aligned} \tag{5.1}$$

$$M(q(t)) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \quad V(q(t), \dot{q}(t)) = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \quad G(q(t)) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}$$

$$M(q(t))\ddot{q}(t) + V(q(t), \dot{q}(t)) + G(q(t)) = \tau \tag{5.2}$$

Burada  $t$  zamanı göstermek üzere  $q(t), \dot{q}(t), \ddot{q}(t) \in \mathbb{R}^2$  eklemlerin açısal konum,

hız ve ivme vektörlerini,  $\tau \in \mathfrak{R}^2$  eklemlere uygulanan moment vektörünü,  $M(q(t)) \in \mathfrak{R}^{2 \times 2}$  eylemsizlik matrisini,  $V(q(t), \dot{q}(t)) \in \mathfrak{R}^2$  merkezci ve merkezkaç vektörünü,  $G(q(t)) \in \mathfrak{R}^2$  yerçekimi vektörünü temsil etmektedir. Uzuvcu ağırlıkları  $m_1 = m_2 = 1 \text{ kg.}$ , uzuv uzunlukları  $a_1 = a_2 = 1 \text{ m.}$ , örnekleme frekansı 100 Hz. ve manipülatör uzuvlarının başlangıç açısal konumları  $q_1(0) = 0.1 \text{ rad}, q_2(0) = 0 \text{ rad}$  olarak kabul edilmiştir\*. Dinamiklerde Eş.(2.1)'de tanımlanan sürtünme etkileri ve gürültü bileşenleri gözardı edilmiştir.

Benzetimi yapılan tüm şemalarda manipülatöre ait alınan işaretlerde  $(q(t), \dot{q}(t))$  gürültü bileşeni ihmal edilmiştir. Bunun yanında tanımlanan şemaların gürbüzlüğüne kanıtlamak amacıyla dinamik bileşenlerine  $\eta_M$  eylemsizlik matrisindeki,  $\eta_V$  merkezci ve merkezkaç vektöründeki,  $\eta_G$  yerçekimi vektöründeki belirsizlikler olmak üzere her birine %5 değerinde zamanla değişen belirsizlik eklenmiş ve bu durumla ilgili tanım Eş (5.3) ile verilmiştir:

$$\begin{aligned} \|M(q(t)) + \eta_M\| &\leq 1.05M(q(t)) \\ \|V(q(t), \dot{q}(t)) + \eta_V\| &\leq 1.05V(q(t), \dot{q}(t)) \\ \|G(q(t)) + \eta_G\| &\leq 1.05G(q(t)) \end{aligned} \quad (5.3)$$

Önerilen tüm şemalarda denetleyici olarak HM-PID yöntemi kullanılmıştır. Hesaplanmış Moment yöntemi doğrusal olmayan dizgeler için tanımlanmış geribesleme doğrusallaştırma (feedback linearization) yönteminin robot manipülatörlere uygulanmış özel bir halidir. Genellikle PD ve PID tipinde olan bu denetim yönteminin HM-PID tipi bu çalışma için seçilmiştir ve elde edilmesine dair bilgi Ek A'da verilmiştir (Lewis ve ark., 1993). HM-PID yöntemi ile denetim işareti olan moment işaretinin hesabı Eş. (5.4)'te verilmiştir:

$$\begin{aligned} e(t) &= q_d(t) - q(t) \\ \dot{e}(t) &= \dot{q}_d(t) - \dot{q}(t) \\ \ddot{e}(t) &= \ddot{q}_d(t) - \ddot{q}(t) \\ \tau &= M(q(t))(\ddot{q}_d(t) + K_v \dot{e}(t) + K_p e(t) + K_i \varepsilon(t)) + V(q(t), \dot{q}(t)) + G(q(t)) \end{aligned} \quad (5.4)$$

Burada  $e(t)$  hata işareti,  $\varepsilon(t)$  işaretinin türevidir. Yapılan benzetimler için kazanç matrisleri  $K_v = 100 I_{2 \times 2}$ ,  $K_p = 20 I_{2 \times 2}$ ,  $K_i = 500 I_{2 \times 2}$  şeklinde tanımlanmıştır.

\* Robot dinamiklerinin ve şemaya ait diğer programların yazımı sırasında sayısal integral ve türev hesabı için Euler yöntemi kullanılmıştır.

Tez çalışmasında daha önce söz edildiği üzere eklem (aktüatör) arızaları gözönünde bulundurulmuş, literatürde üzerinde daha az çalışma yapılmış olan kısmi eklem arızaları üzerinde durulmuş ve Eş. (2.3)'teki kısmi eklem arızalı manipülatör tanım eşitliği kullanılmıştır. Tanımlanan iki eklemli manipülatör için her iki ekleme arızalar tanımlanmış ve bu arızalar Çizelge 5.1'de verilmiştir.

**Çizelge 5.1.** Benzetim için tanımlanan arızalar ve isimlendirilmesi

Arıza	İsmi
Eklem 1 %50 kayıp	$f_1$
Eklem 1 %30 kayıp	$f_2$
Eklem 2 %50 kayıp	$f_3$
Eklem 2 %30 kayıp	$f_4$

### 5.1. Önerilen 1. FDI Şemasına Ait Benzetim Sonuçları

Bu altbölümde Şekil 4.1 ile Bölüm 4.1'de önerilen 1. FDI şemasına ait benzetim sonuçları verilecektir. Önceki bölümde anlatıldığı üzere robot manipülatör ve denetim blokları oluşturulduktan sonra modelleyici M-ANFIS ve arıza yalıtımı için kullanılan SA blokları gerçekleştirilmiştir.

Tanımlanan robot manipülatörü modellemek için M-ANFIS, her iki ekleme ait anlık moment ve birim adım önceki açısız konum ve hız bilgilerini alan ortak 6 girişli  $(\tau_1(t), \tau_2(t), q_1(t - \Delta t), q_2(t - \Delta t), \dot{q}_1(t - \Delta t), \dot{q}_2(t - \Delta t))$  ve ilgili ekleme ait anlık açısız konum veya hız kestirim işaretini  $(\hat{q}_1(t), \hat{q}_2(t), \hat{\dot{q}}_1(t), \hat{\dot{q}}_2(t))$  veren 4 bağımsız ANFIS'den oluşturulmuştur.

Eğitim için genlikleri  $\pm 1$  arasında ve sin- ve cos- şeklinde Ek B'de verilen 146 farklı yörüngenin benzetimi yapılmış, bu benzetimlere ait 76 örnekle veriler her bir ANFIS'in eğitiminde kullanılmıştır. Her bir ANFIS'te her bir giriş için genelleştirilmiş-çan şeklinde iki üyelik işlevi ile tanımlanmış ve öğrenme algoritması olarak melez öğrenme algoritması kullanılmıştır.

M-ANFIS ile oluşturulan robot manipülatör modeli ile gerçek manipülatör dizgesi arasındaki açısız konum ve hız fark işaretleri elde edilmiş, böylece artık işaretlerinin türetim işlemi gerçekleştirilmiştir. Türetilen artıklara ait eşitlikler Eş. (5.5) ile verilmiştir.

$$\begin{aligned}
r_1(t) &= q_1(t) - \hat{q}_1(t) \\
r_2(t) &= \dot{q}_1(t) - \dot{\hat{q}}_1(t) \\
r_3(t) &= q_2(t) - \hat{q}_2(t) \\
r_4(t) &= \dot{q}_2(t) - \dot{\hat{q}}_2(t)
\end{aligned} \tag{5.5}$$

Türetilen artık işaretlerine ek olarak arıza bulma ve yalıtım yüzdelerini arttırmak amacıyla analitik fazlalık işaretler türetilmiştir. Bu işaretlere ait eşitlikler de Eş. (5.6) ile verilmiştir.

$$\begin{aligned}
(r_1(t))' &= (q_1(t) - \hat{q}_1(t))' = \dot{q}_1(t) - \dot{\hat{q}}_1(t) \\
(r_3(t))' &= (q_2(t) - \hat{q}_2(t))' = \dot{q}_2(t) - \dot{\hat{q}}_2(t)
\end{aligned} \tag{5.6}$$

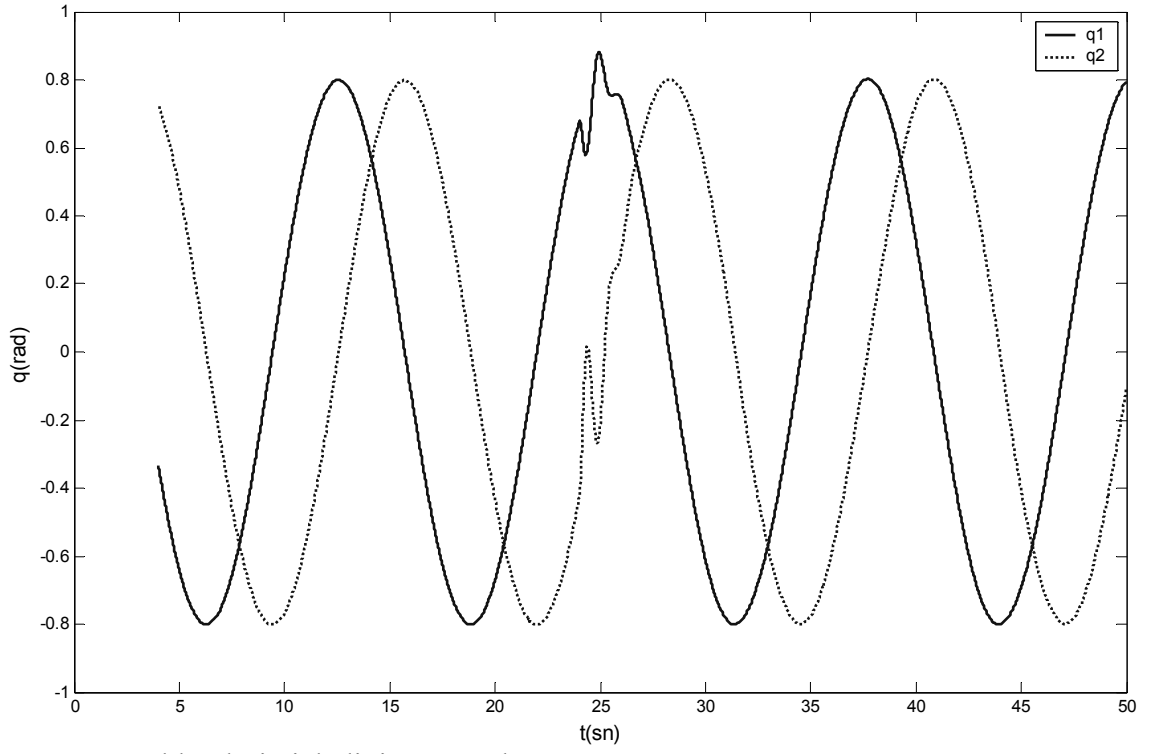
Arıza bulma ve onu takip eden arıza yalıtımında kullanılacak sinir ağının eğitimi için 146 yörüngeden 12 yörünge seçilmiştir. Artık türetimi için bu yörüngelere ait sağlıklı durumda benzetimler yapılmıştır. Robot manipülatörün arızasız durumda artık değerlerinin dinamiklerin içerdiği belirsizlik sebebiyle belirli bir aralıkta değer almasından dolayı bu artıklara ait alt ve üst eşik değerleri benzetimler yardımıyla elde edilmiştir. Artıkların herhangi birinin herhangi bir anda bu eşik değerlerini aşması arıza bulma işareti olarak algılanmaktadır.

Arıza yalıtımı için 6 girişli ve her biri Çizelge 5.1'deki bir arızaya karşılık gelen 4 çıkışlı, 10-20-20-4 şeklinde nöronlu, RP algoritmasını kullanan bir sinir ağı kullanılmıştır. Ağı eğitimi için Çizelge 5.1'deki arızaların 12 yörünge için benzetimi yapılmış, 101 örnekli 4 artık ve 2 analitik fazlalık giriş olarak (6 giriş\*101 örnek sayısı\*4 işaret\*12 yörünge-6\*4848 veri noktası) ve her biri bir arızayı temsil eden 4 çıkış 0.9 değeri verecek şekilde kullanılmıştır. Eğitim parametresi olarak 2000 dönem ve 0 hata hedefi seçilmiştir. Eğitim süreci 5 dakikadan daha kısa bir sürede tamamlanmış ve hata değeri  $10^{-3}$ 'den aşağı elde edilmiştir.

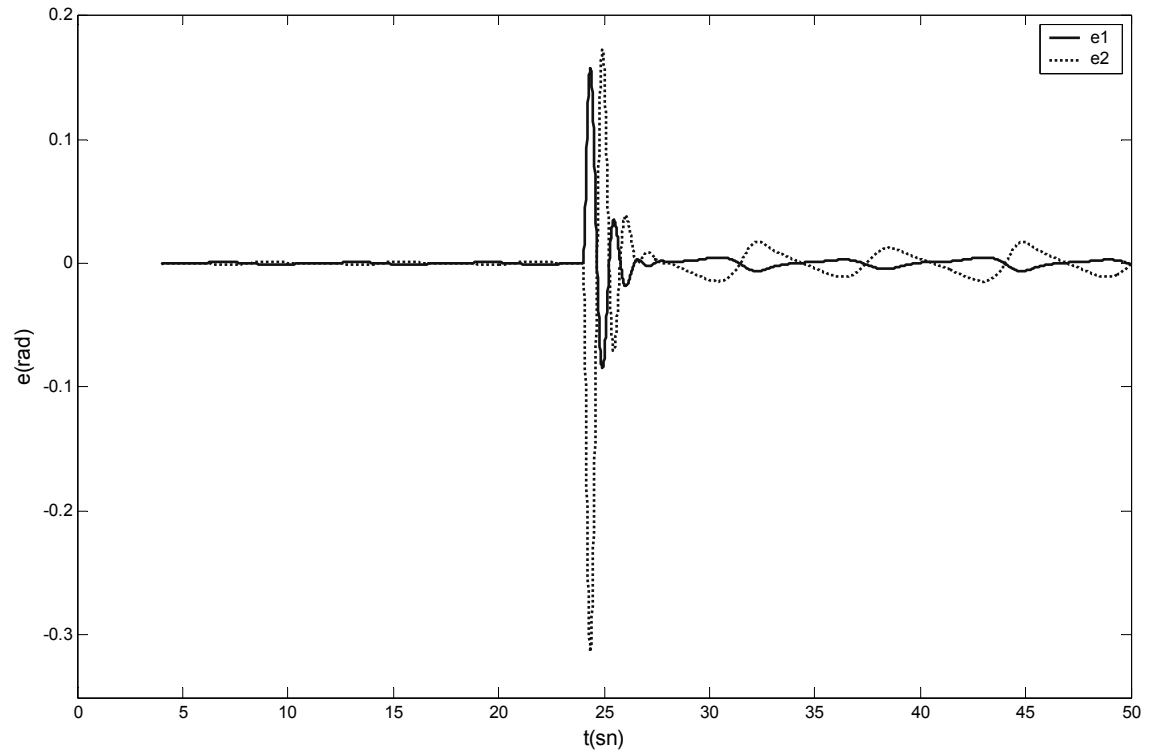
Her ne kadar üzerinde çalışılan kısmi arızalar sürekli tipte arızalar olarak kabul edilse de tasarlanan şemalar arıza bulma anından sonra da sürekli olarak artıkları gözlemlemeye devam etmektedir. Bu sebeple arıza bulmayı takip eden arıza yalıtımında sinir ağı, geçici durumdan sürekli duruma geçişte yanlış arıza vermesini önlemek amacıyla arıza bulma işaretinden 3.85 sn. sonra devreye girecek şekilde ayarlanmıştır.

Sinir ağı çıkışları değerlendirilirken çıkışlarından herhangi birinin 0.5 değerini aşması o arızaya ait yalıtım işareti olarak tanımlıdır. Bazı yörüngelere ait bazı arızalarda

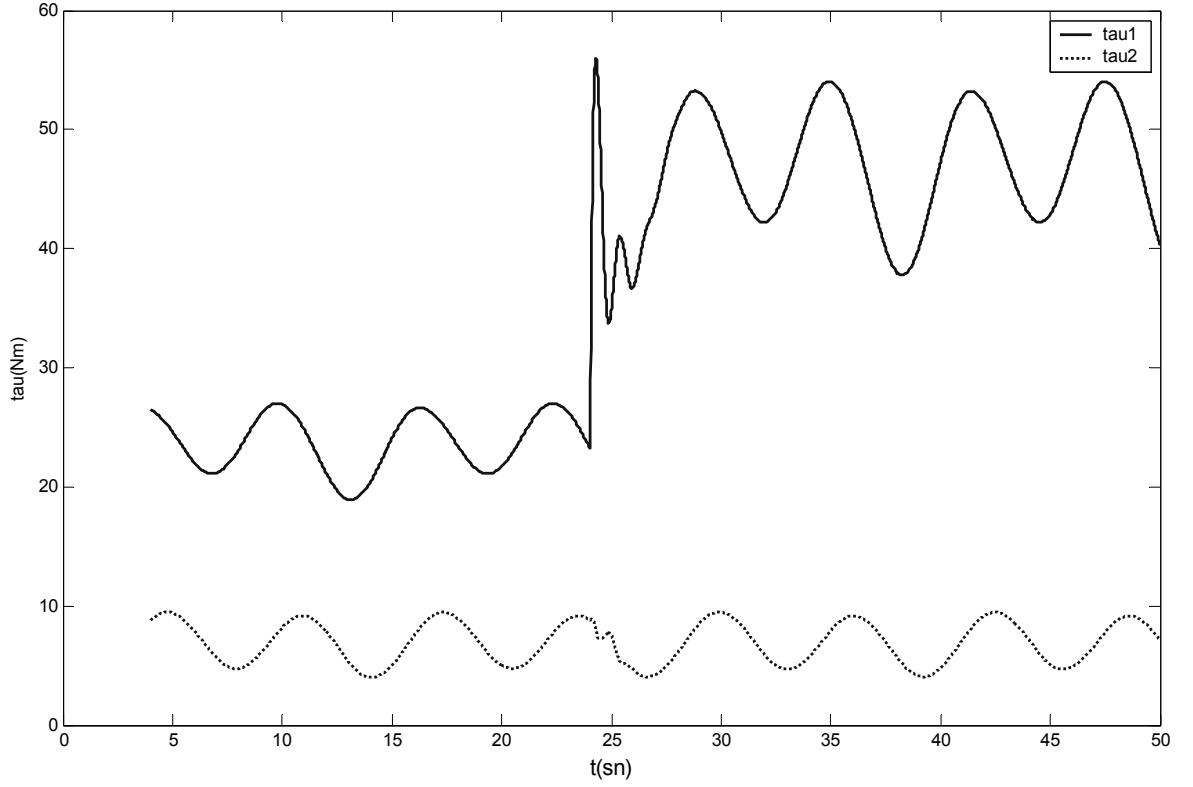




Şekil 5.3. Eklemlerin izlediği yörüngeler

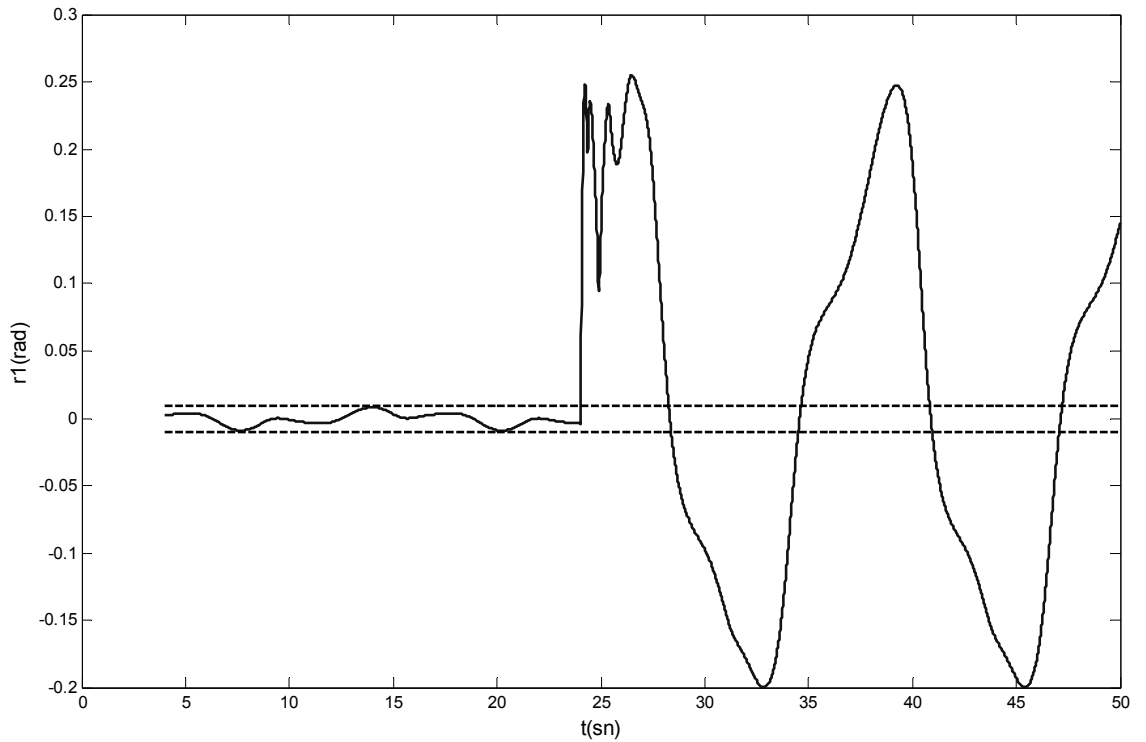


Şekil 5.4. Eklemlerin yaptığı yörünge hataları

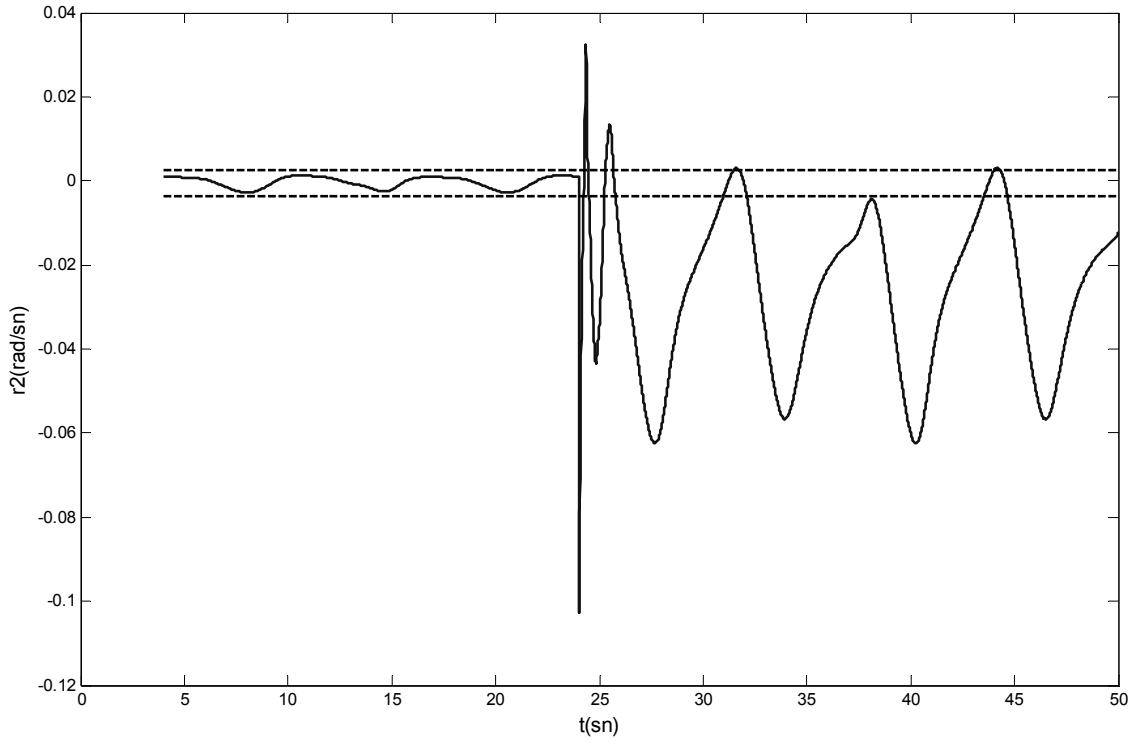


**Şekil 5.5.** Eklemlere uygulanan momentler

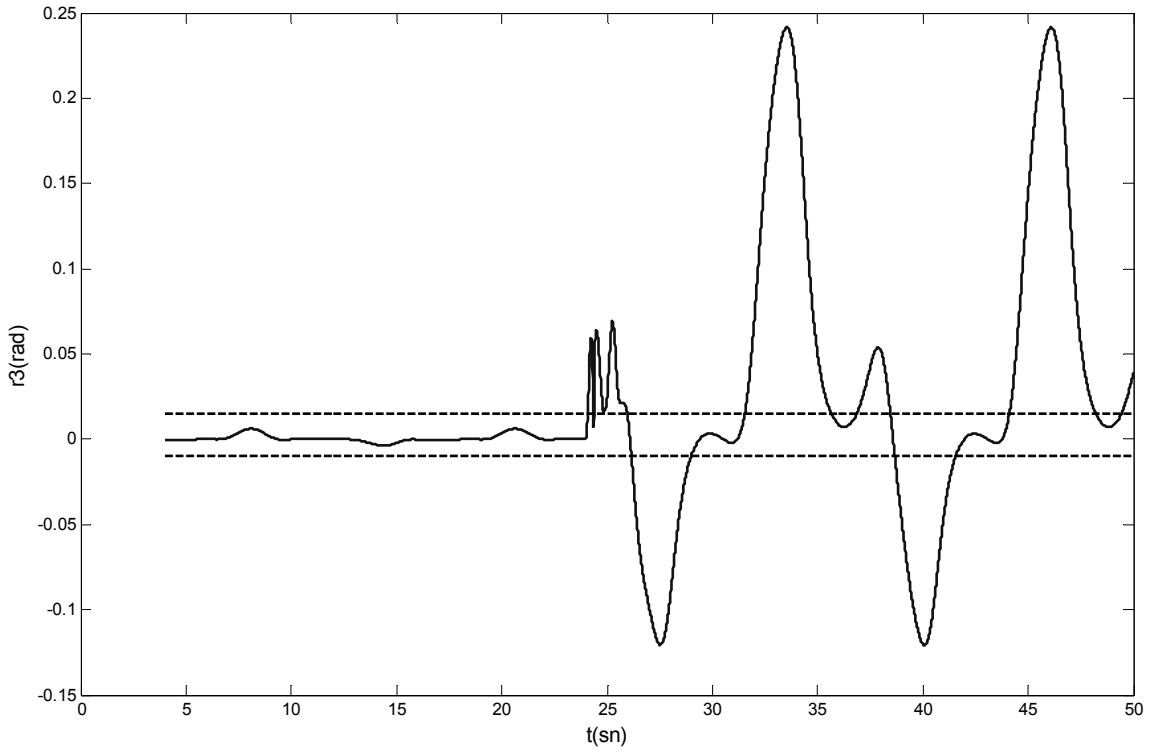
Tanımlanan yörünge ve arıza için M-ANFIS tarafından türetilen artıklar ve bunlara bağlı türetilen analitik fazlalıklar ise izleyen şekillerde alt ve üst eşik sınırlarıyla birlikte verilmiştir.



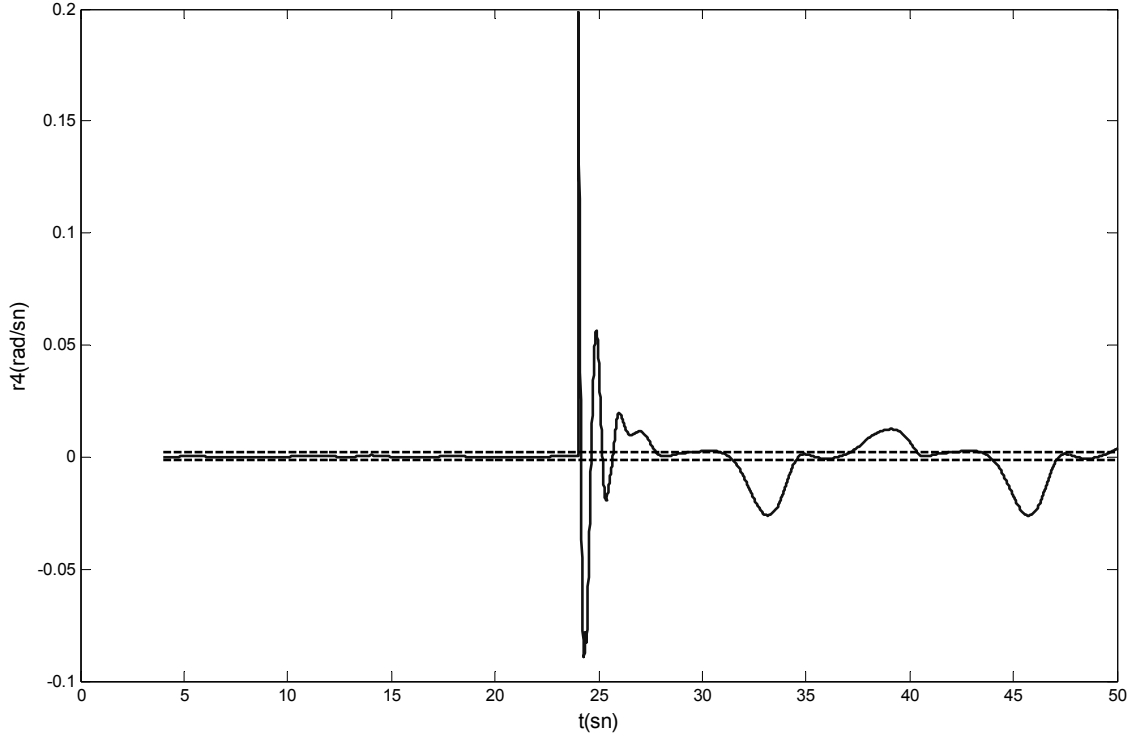
**Şekil 5.6.** 1. şema için artık  $r_1$  ( $r_1(t) = q_1(t) - \hat{q}_1(t)$ )



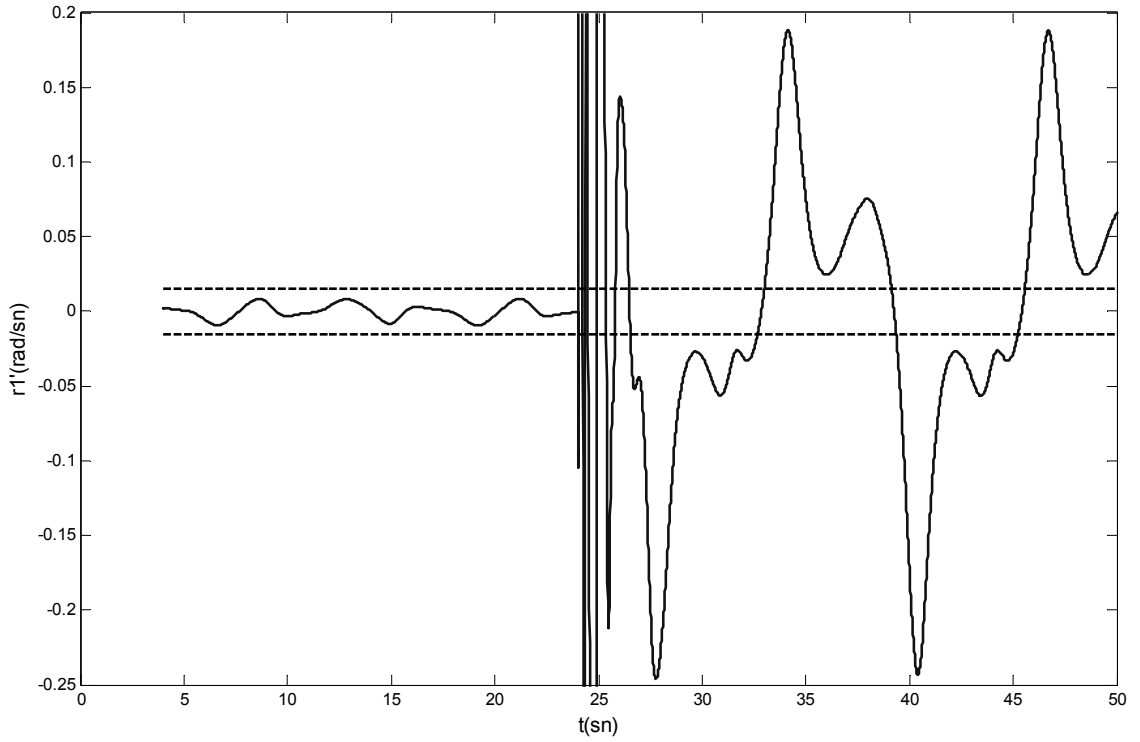
**Şekil 5.7.** 1. şema için artık  $r_2$  ( $r_2(t) = \dot{q}_1(t) - \hat{\dot{q}}_1(t)$ )



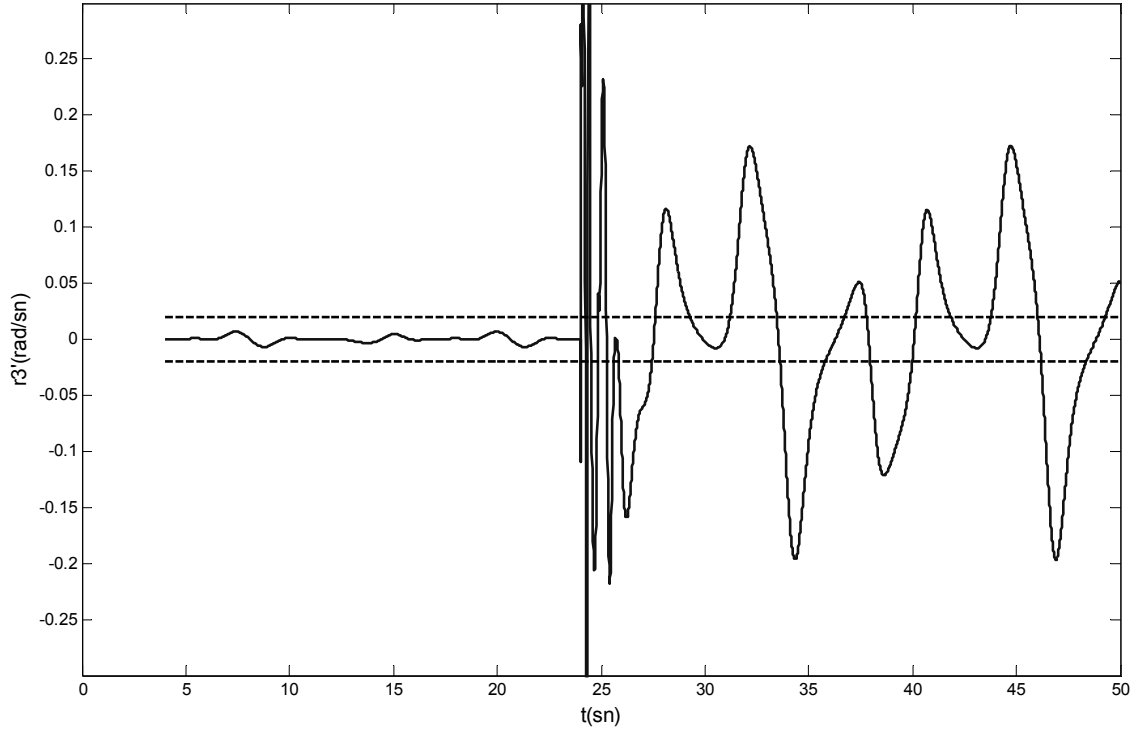
**Şekil 5.8.** 1. şema için artık  $r_3$  ( $r_3(t) = q_2(t) - \hat{q}_2(t)$ )



**Şekil 5.9.** 1. şema için artık  $r_4$  ( $r_4(t) = \dot{q}_2(t) - \hat{\dot{q}}_2(t)$ )

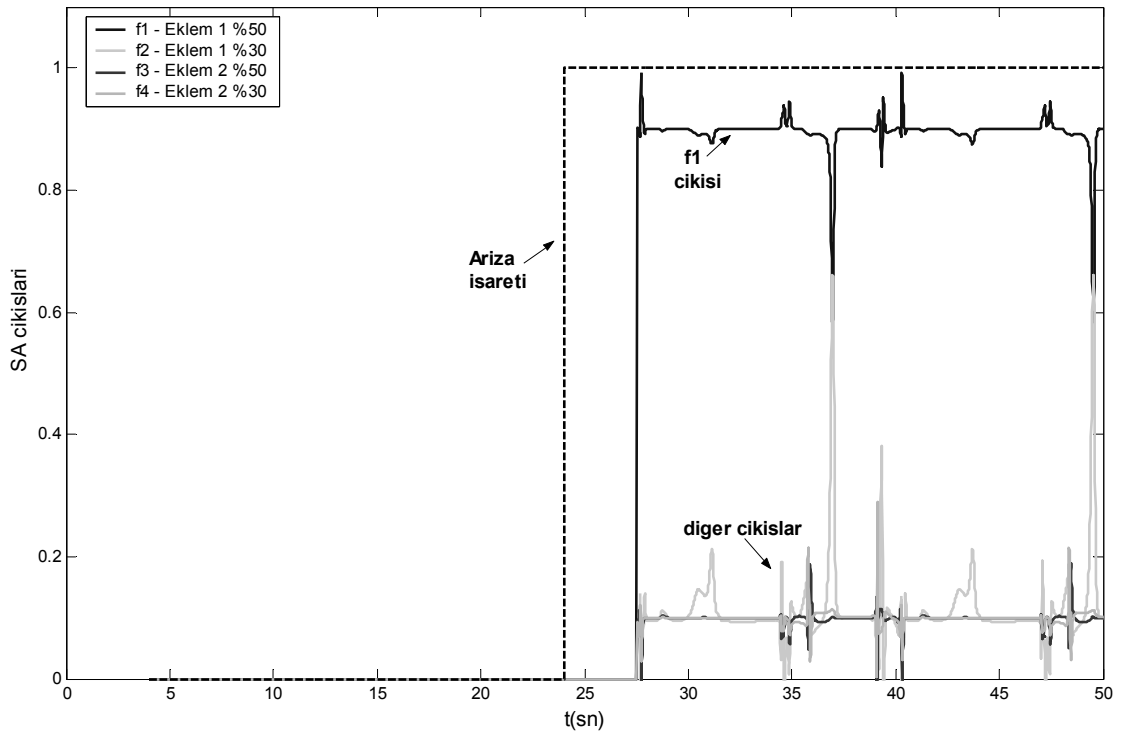


**Şekil 5.10.** 1. şema için artık  $r_1'$  ( $(r_1(t))' = (q_1(t) - \hat{q}_1(t))' = \dot{q}_1(t) - \hat{\dot{q}}_1(t)$ )

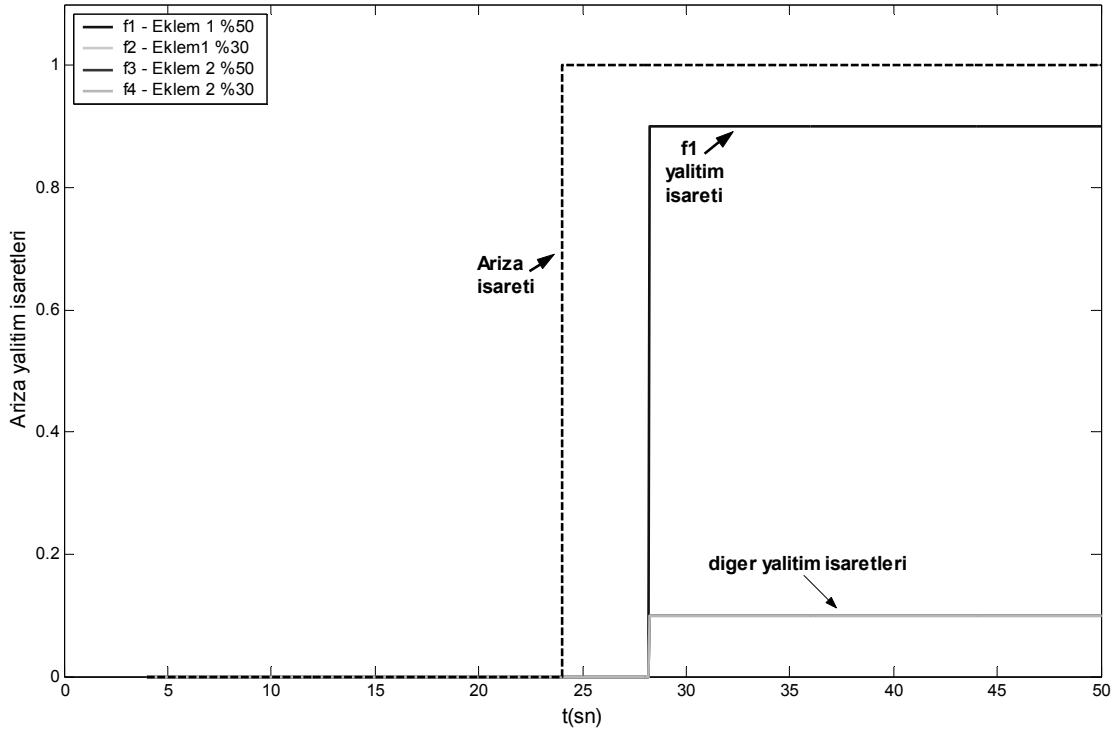


**Şekil 5.11.** 1. şema için artık  $r_3'$   $((r_3(t))' = (q_2(t) - \hat{q}_2(t))' = \dot{q}_2(t) - \hat{\dot{q}}_2(t))$

Türetilen artıkları ve analitik fazlalıkları giriş olarak alan sinir ağına ait çıkış işaretleri Şekil 5.12'de, süreklilik taramasından sonra elde edilen arıza yalıtım işaretleri ise Şekil 5.13'de verilmiştir.



**Şekil 5.12.** 1. şema için sinir ağı çıkışları



**Şekil 5.13.** 1. şema için arıza yalıtım işaretleri

Şekil 5.12'deki sinir ağı çıkışları incelendiğinde  $f_1$  ve  $f_2$ 'ye ait çıkışların 36.9-37.1 sn. arasında 0.5 değerlerini aştığı görülmektedir. Süreklilik ölçütü bu durumun gözardı edilmesini sağlamıştır. Şekil 5.13 incelendiğinde  $t = 28.21$  sn. anı arıza yalıtım anı olarak görülmektedir.

Önerilen 1. FDI şemasına ait MATLAB programı ve program akış diyagramı Ek C'de verilmiştir.

## 5.2. Önerilen 2. FDI Şemasına Ait Benzetim Sonuçları

Bu altbölümde Şekil 4.6 ile Bölüm 4.2'de önerilen 2. FDI şemasına ait benzetim sonuçları verilecektir. Bu şemada M-ANFIS blokları ile 1 sağlıklı manipülatör ve Çizelge 5.1 ile tanımlı 4 arıza için 4 arızalı manipülatör modeli olmak üzere 5 ayrı manipülatör modeli oluşturulmuştur.

Her bir M-ANFIS 1. şemadakine benzer şekilde her iki ekleme ait anlık moment ve birim adım önceki açısal konum ve hız bilgilerini alan ortak 6 girişli  $(\tau_1(t), \tau_2(t), q_1(t - \Delta t), q_2(t - \Delta t), \dot{q}_1(t - \Delta t), \dot{q}_2(t - \Delta t))$  ve ilgili ekleme ait anlık açısal konum veya hız kestirim işaretini  $(\hat{q}_1(t), \hat{q}_2(t), \hat{q}_1(t), \hat{q}_2(t), \hat{q}_{1f_1}(t), \hat{q}_{2f_1}(t), \hat{q}_{1f_1}(t), \hat{q}_{2f_1}(t), \hat{q}_{1f_2}(t), \hat{q}_{2f_2}(t), \hat{q}_{1f_2}(t), \hat{q}_{2f_2}(t), \hat{q}_{1f_3}(t), \hat{q}_{2f_3}(t), \hat{q}_{1f_3}(t), \hat{q}_{2f_3}(t), \hat{q}_{1f_4}(t), \hat{q}_{2f_4}(t), \hat{q}_{1f_4}(t), \hat{q}_{2f_4}(t))$  veren 4 bağımsız ANFIS'den (toplamda 20 ayrı ANFIS) oluşturulmuştur.

Eğitim için 1. şemadakinine benzer şekilde genlikleri  $\pm 1$  arasında ve sin ve cos olarak tanımlı 146 ayrı yörünge için benzetimi sağlıklı ve arızalı durumlar için yapılmış ve bu benzetimlere ait 76 örneklili veriler uygun M-ANFIS eğitimi için kullanılmıştır. Her bir M-ANFIS'deki ANFIS'ler yine benzer şekilde her bir giriş için genelleştirilmiş-çan şeklinde iki üyelik işlevi ile tanımlanmış ve öğrenme algoritması olarak melez öğrenme algoritması kullanılmıştır. Eğitim sonrasında elde edilen 20 farklı artığa ait tanımlar Eş. (5.8)'de verilmiştir:

$$\begin{aligned}
 & \left. \begin{aligned} r_1(t) &= q_1(t) - \hat{q}_1(t) \\ r_2(t) &= \dot{q}_1(t) - \hat{\dot{q}}_1(t) \\ r_3(t) &= q_2(t) - \hat{q}_2(t) \\ r_4(t) &= \dot{q}_2(t) - \hat{\dot{q}}_2(t) \end{aligned} \right\} \text{Sağlıklı,} \\
 & \left. \begin{aligned} r_{1f_1}(t) &= q_1(t) - \hat{q}_{1f_1}(t) \\ r_{2f_1}(t) &= \dot{q}_1(t) - \hat{\dot{q}}_{1f_1}(t) \\ r_{3f_1}(t) &= q_2(t) - \hat{q}_{2f_1}(t) \\ r_{4f_1}(t) &= \dot{q}_2(t) - \hat{\dot{q}}_{2f_1}(t) \end{aligned} \right\} f_1, \quad \left. \begin{aligned} r_{1f_2}(t) &= q_1(t) - \hat{q}_{1f_2}(t) \\ r_{2f_2}(t) &= \dot{q}_1(t) - \hat{\dot{q}}_{1f_2}(t) \\ r_{3f_2}(t) &= q_2(t) - \hat{q}_{2f_2}(t) \\ r_{4f_2}(t) &= \dot{q}_2(t) - \hat{\dot{q}}_{2f_2}(t) \end{aligned} \right\} f_2, \quad (5.8) \\
 & \left. \begin{aligned} r_{1f_3}(t) &= q_1(t) - \hat{q}_{1f_3}(t) \\ r_{2f_3}(t) &= \dot{q}_1(t) - \hat{\dot{q}}_{1f_3}(t) \\ r_{3f_3}(t) &= q_2(t) - \hat{q}_{2f_3}(t) \\ r_{4f_3}(t) &= \dot{q}_2(t) - \hat{\dot{q}}_{2f_3}(t) \end{aligned} \right\} f_3, \quad \left. \begin{aligned} r_{1f_4}(t) &= q_1(t) - \hat{q}_{1f_4}(t) \\ r_{2f_4}(t) &= \dot{q}_1(t) - \hat{\dot{q}}_{1f_4}(t) \\ r_{3f_4}(t) &= q_2(t) - \hat{q}_{2f_4}(t) \\ r_{4f_4}(t) &= \dot{q}_2(t) - \hat{\dot{q}}_{2f_4}(t) \end{aligned} \right\} f_4
 \end{aligned}$$

Önerilen 2. şemada belirtildiği üzere GOS yaklaşımı kullanan şemalarda arıza bulma ve yalıtım işleminin aynı anda yapıldığı açıkça görülmektedir. Her ne kadar Çizelge 4.1'de varolan duruma ait artıkların sıfır olması gerektiği belirtilse de yine dinamiklerdeki belirsizliklere bağlı olarak belirli değerler alacaklardır. Bunun için her biri duruma ait artıkların (1 sağlıklı+4 arıza) seçilen 80 yörünge için benzetimleri yapılmış, alt ve üst eşik değerleri belirlenmiş, Çizelge 4.1'e bağlı olarak artıkların bu değerleri aşımı durumunun kendi haricinde bir duruma denk geldiği kabul edilmiştir.

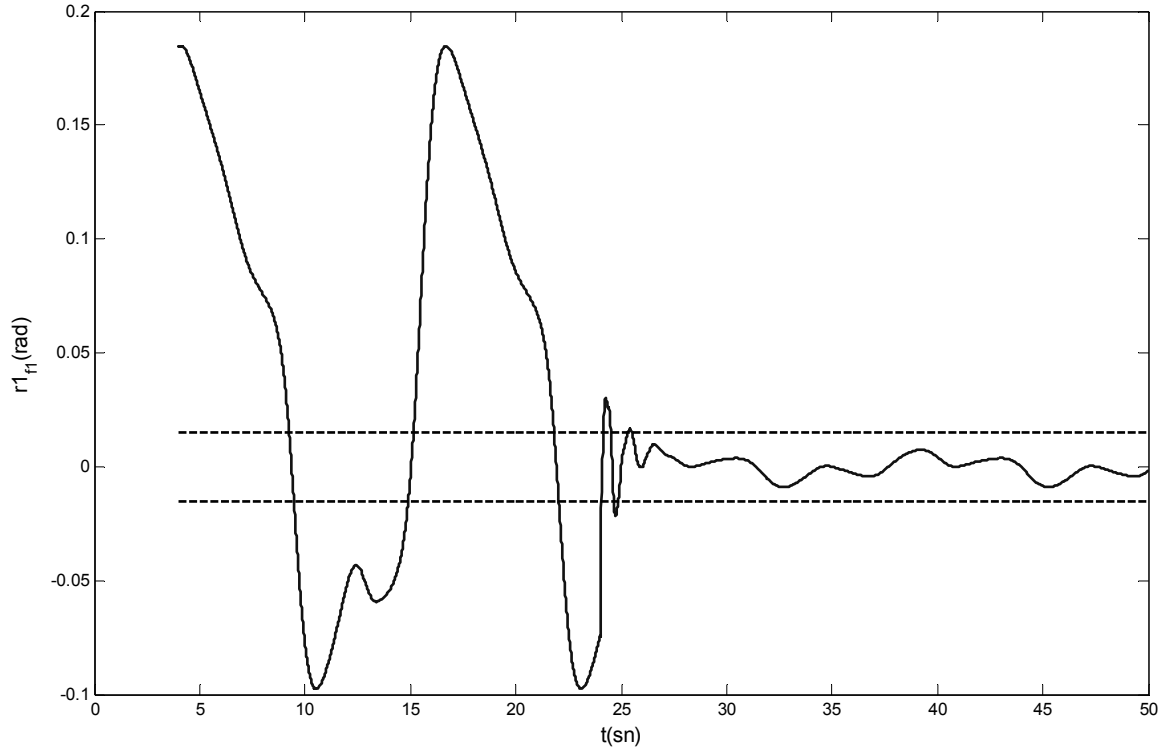
Arıza yalıtım yüzdesini elde etmek için yapılan denemelerde çeşitli yörüngeler için çeşitli arızaların aynı anda iki arıza artığını birden eşikleri arasına düşürdüğü görülmüştür. Bu sebeple yüzdeyi artırmak amacıyla artıklar *sağlıklı-f<sub>2</sub>-f<sub>1</sub>-f<sub>4</sub>-f<sub>3</sub>* sırasını izleyecek şekilde değerlendirmeye alınmıştır. Ayrıca aynı durumu engellemek için süreklilik ölçütü kabul edilmiş ve artıkların 0.8 sn.'den daha kısa süreli eşik aşımı ve eşik arasına düşümler gözardı edilmiştir.

Tanımlanan bu özelliklerle, tanımlanan arızalar ve verilen yörüngeler için %87.81 arıza yalıtım yüzdesi elde edilmiştir. Örnek olarak 1. şema için verilen Eş. (5.7)'deki yörünge için aynı arıza ( $f_1$ ,  $t = 24$  sn. anında) tanımlanmıştır. Eklemlerin izlediği yörüngeler, yaptığı hatalar ve uygulanan momentler aynı olacağından bu işaretlere ait şekiller tekrar verilmemiştir. İzleyen şekillerde 4-50 sn. arasında sırasıyla manipülâtör modellerinin türettikleri artıklar eşikleriyle birlikte verilmiştir. Sağlıklı duruma ait artıklar aynı olduğundan (Şekil 5.6-9) tekrar yer verilmemiştir.

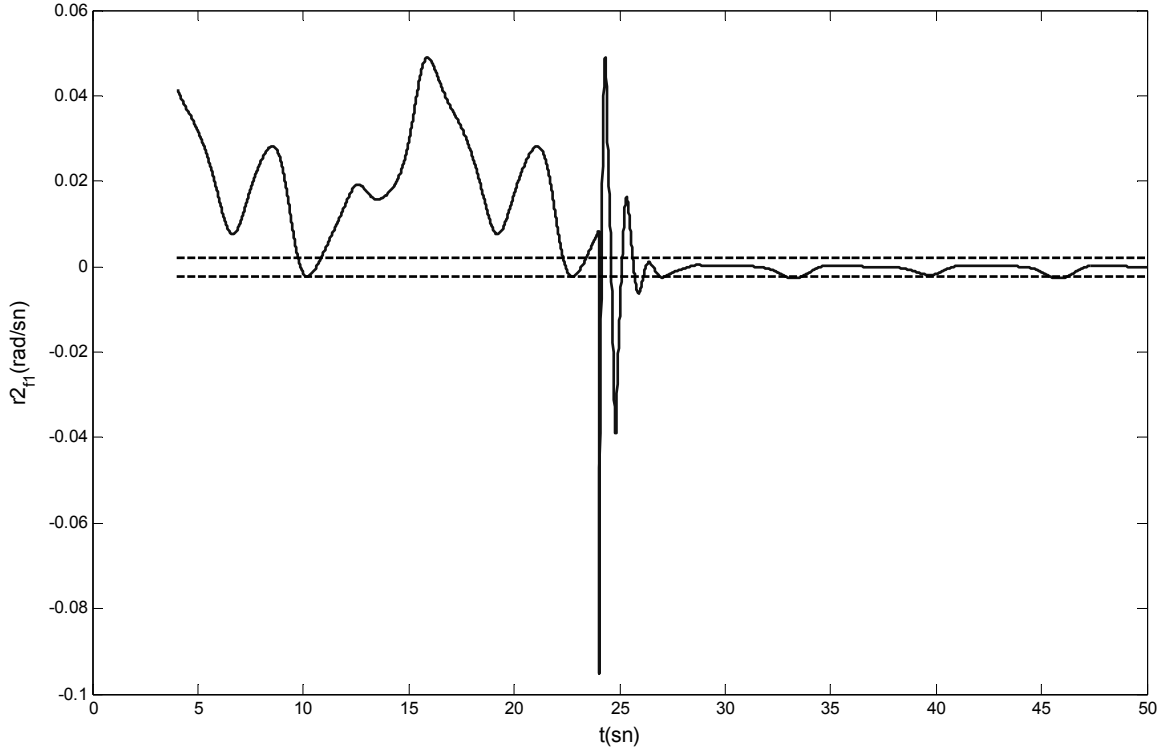
Şemaya ait artıkların arıza yalıtımında kullanımı için Çizelge 4.1'deki karar çizelgesi Çizelge 5.2 şeklinde uyarlanmıştır.

**Çizelge 5.2** 2. FDI şeması için karar çizelgesi

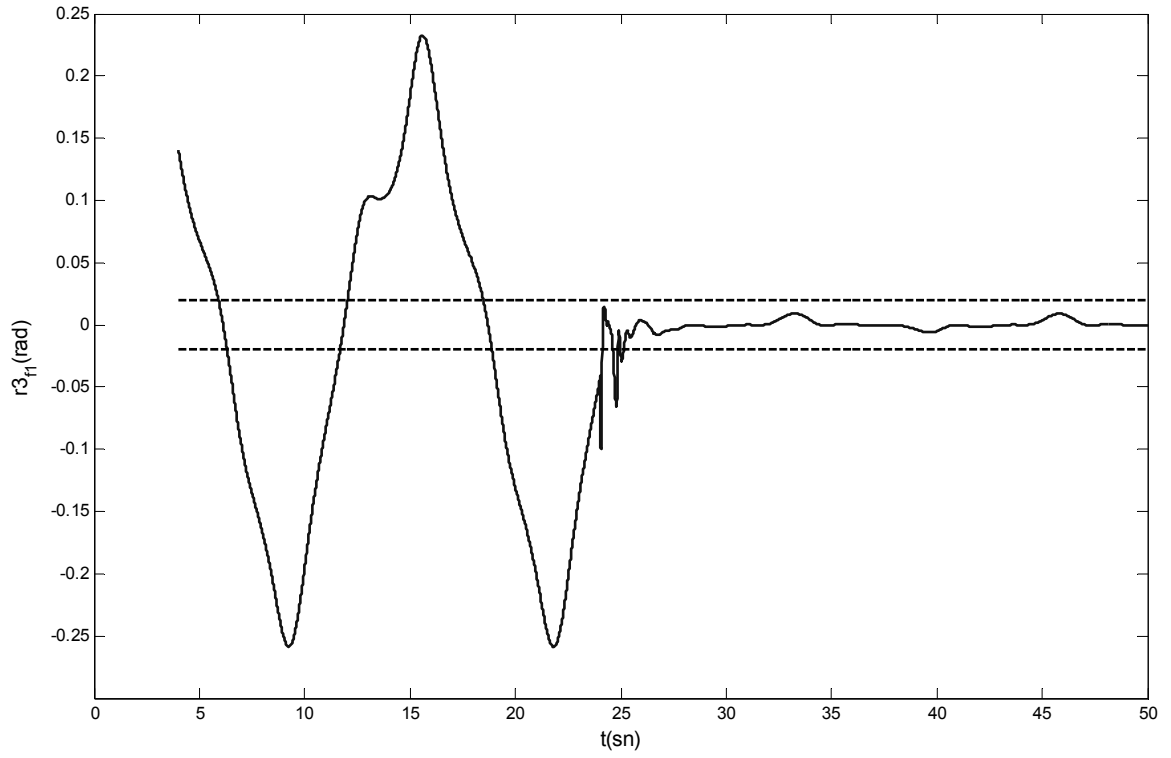
Arızalar	$r_1, r_2, r_3, r_4$	$r_{1f_1}, r_{2f_1}, r_{3f_1}, r_{4f_1}$	$r_{1f_2}, r_{2f_2}, r_{3f_2}, r_{4f_2}$	$r_{1f_3}, r_{2f_3}, r_{3f_3}, r_{4f_3}$	$r_{1f_4}, r_{2f_4}, r_{3f_4}, r_{4f_4}$
Sağlıklı	0	1	1	1	1
$f_1$	1	0	1	1	1
$f_2$	1	1	0	1	1
$f_3$	1	1	1	0	1
$f_4$	1	1	1	1	0



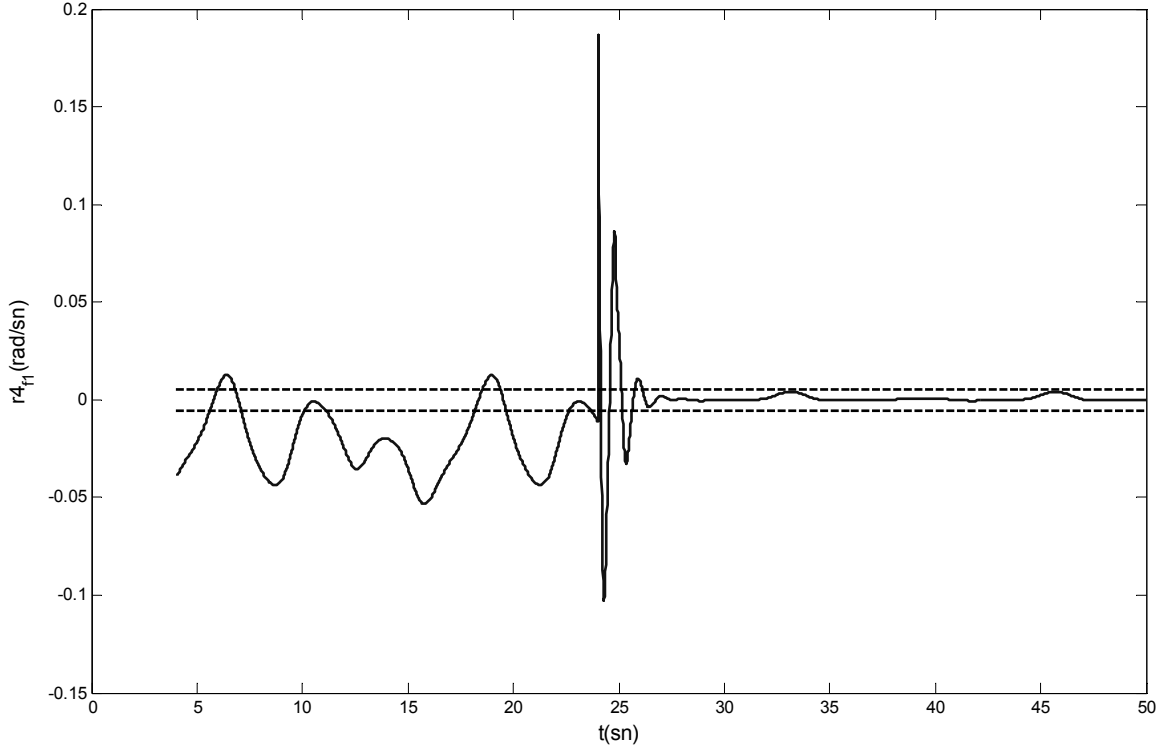
**Şekil 5.14.** 2. şema için artık  $r_{1f_1}$



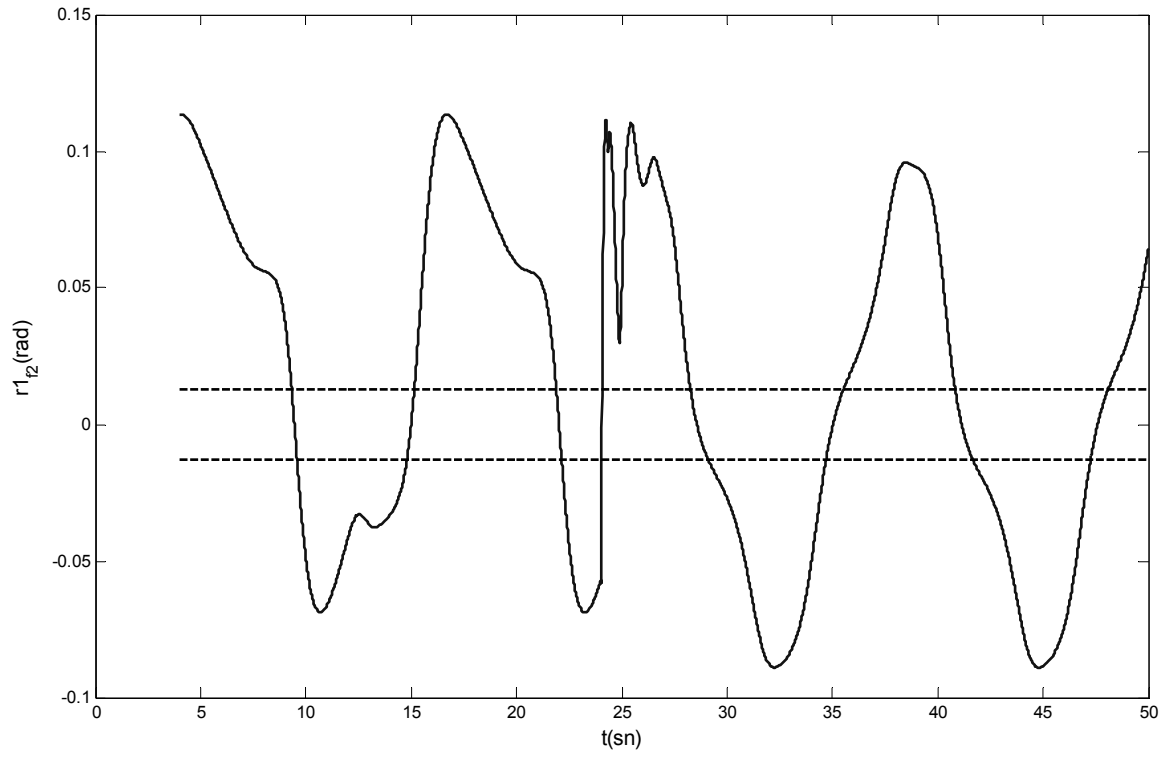
Şekil 5.15. 2. şema için artık  $r_{2f_1}$



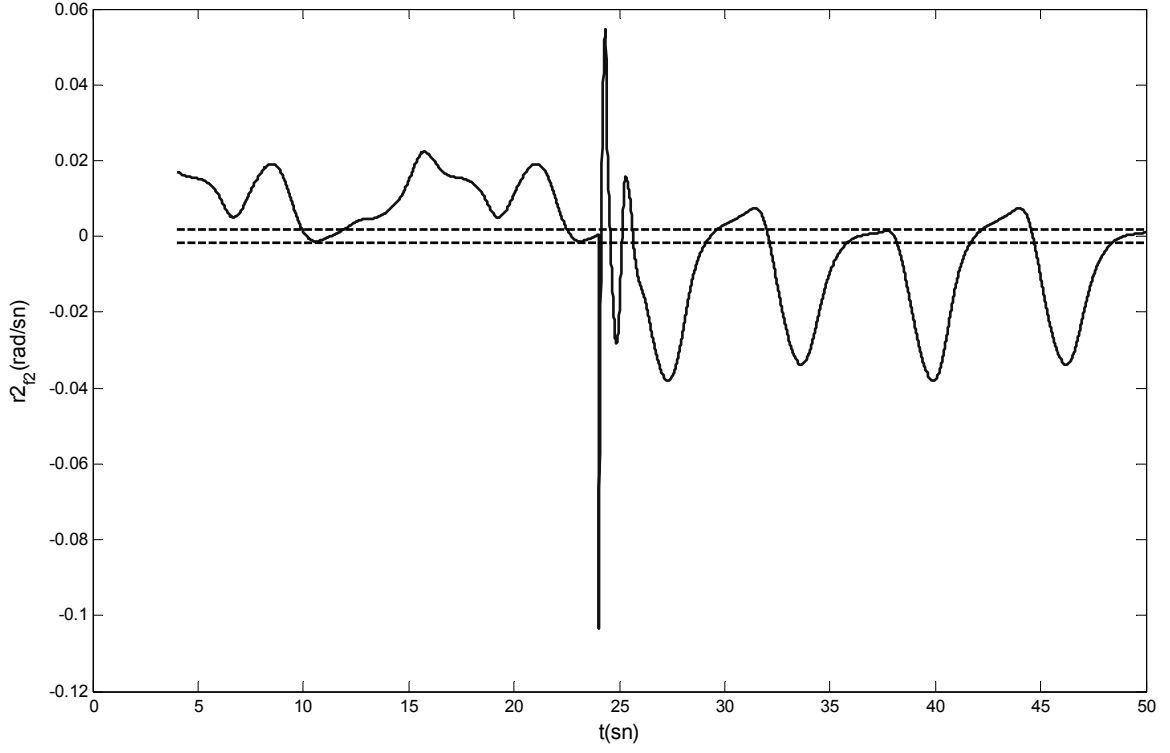
Şekil 5.16. 2. şema için artık  $r_{3f_1}$



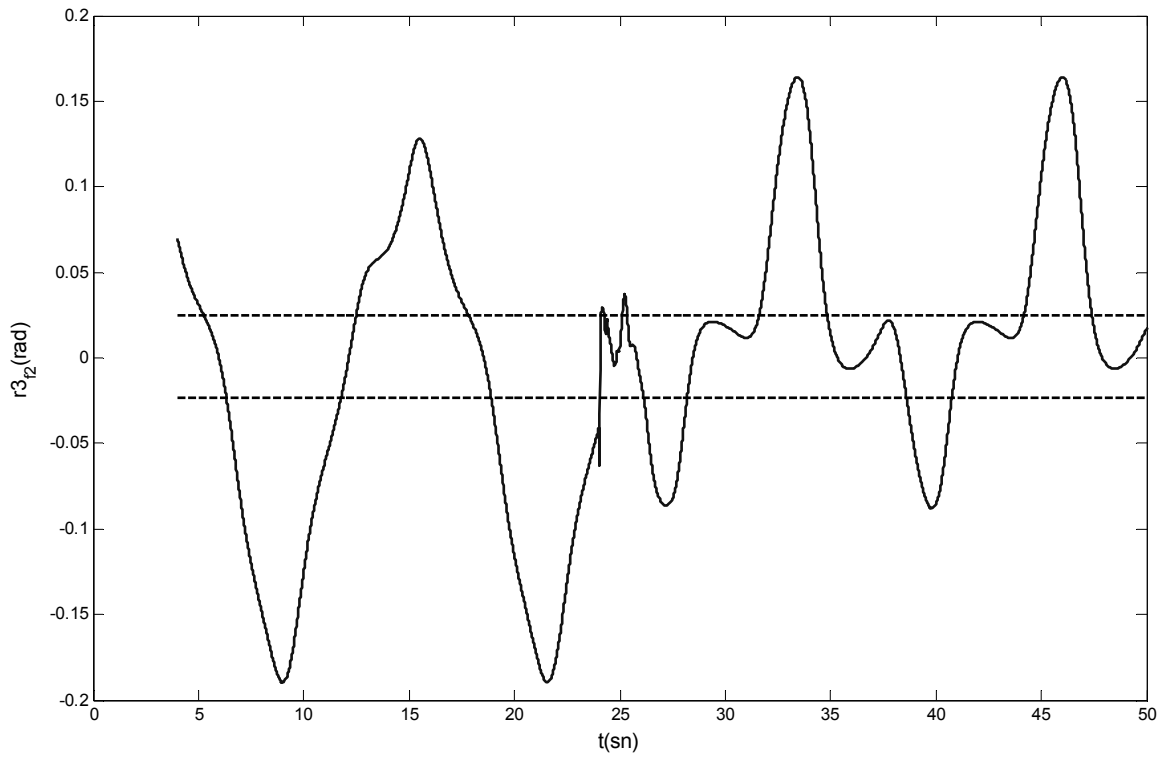
Şekil 5.17. 2. şema için artık  $r_{4,f_1}$



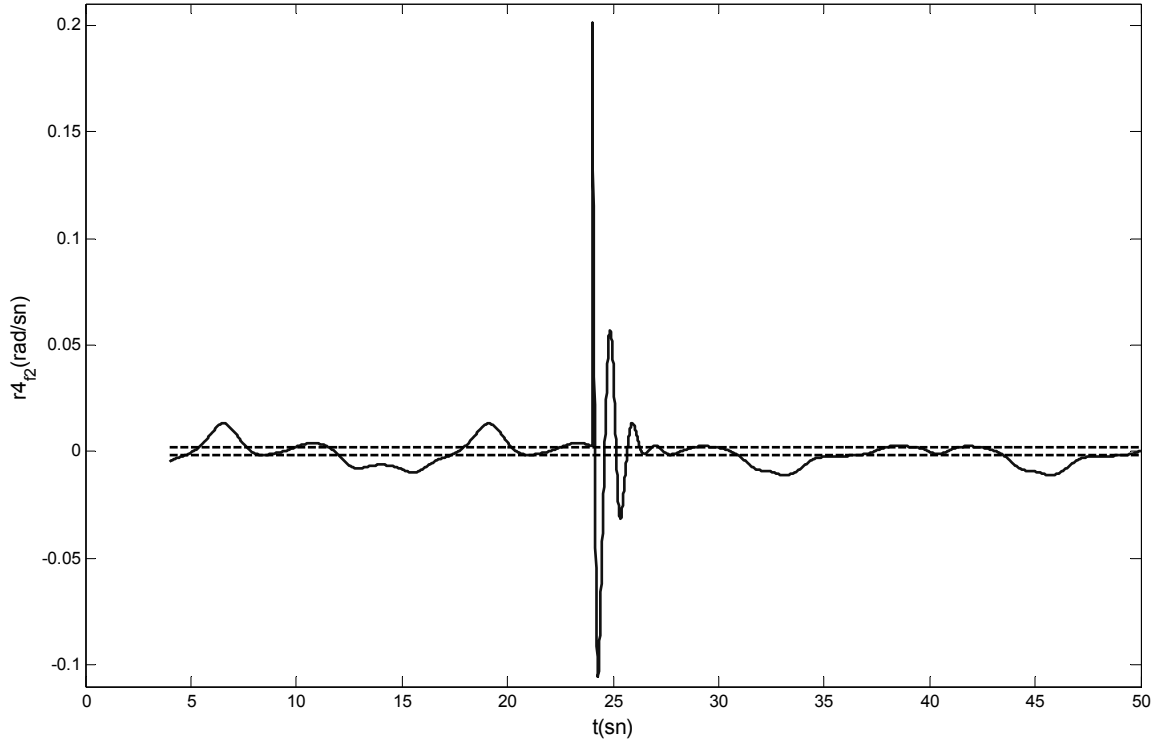
Şekil 5.18. 2. şema için artık  $r_{1,f_2}$



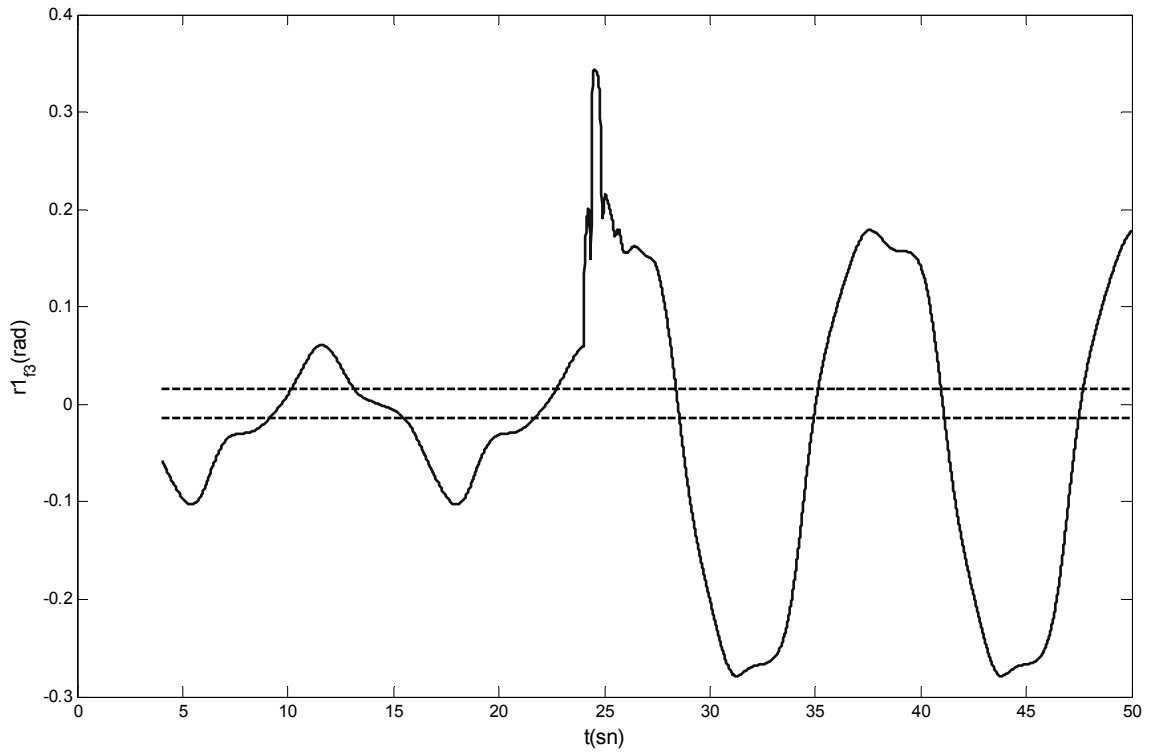
Şekil 5.19. 2. şema için artık  $r_{2f_2}$



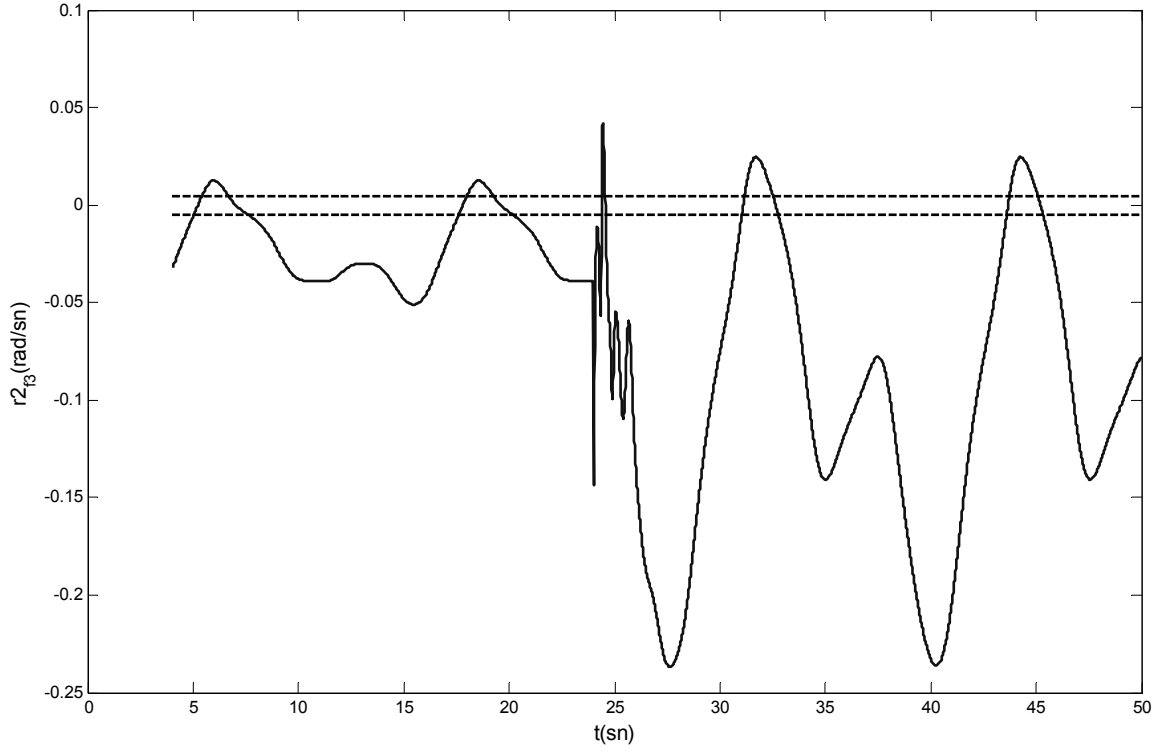
Şekil 5.20. 2. şema için artık  $r_{3f_2}$



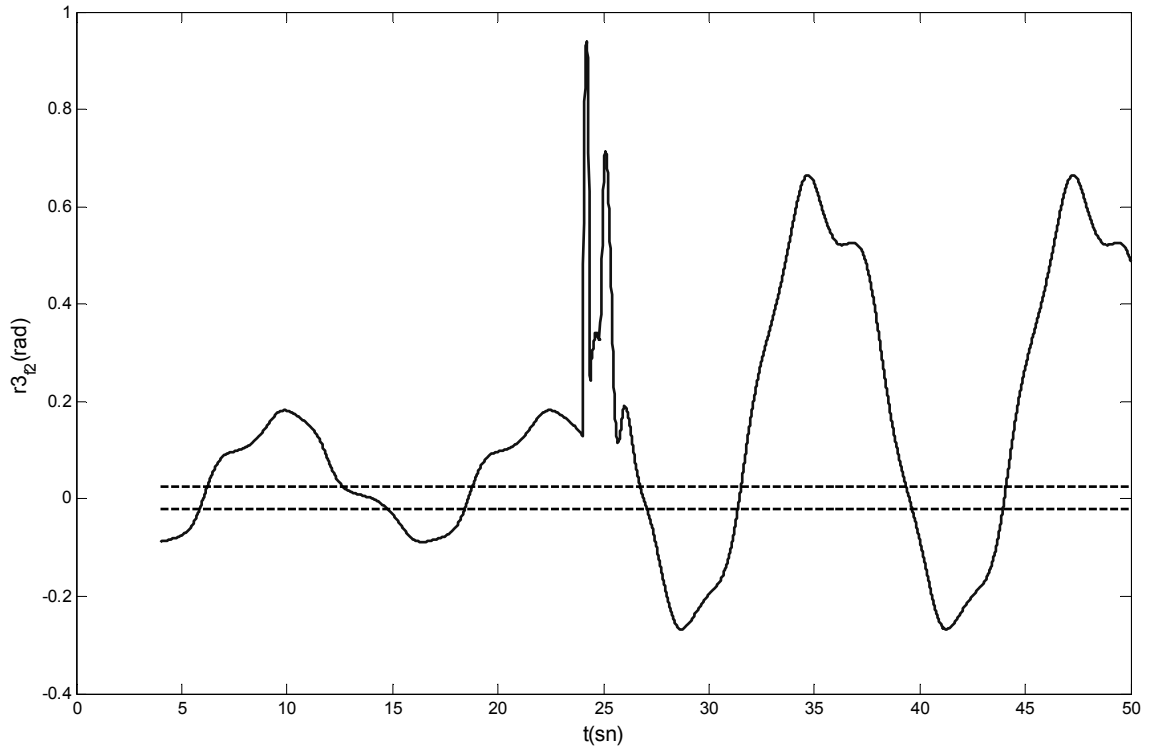
Şekil 5.21. 2. şema için artık  $r_{4f_2}$



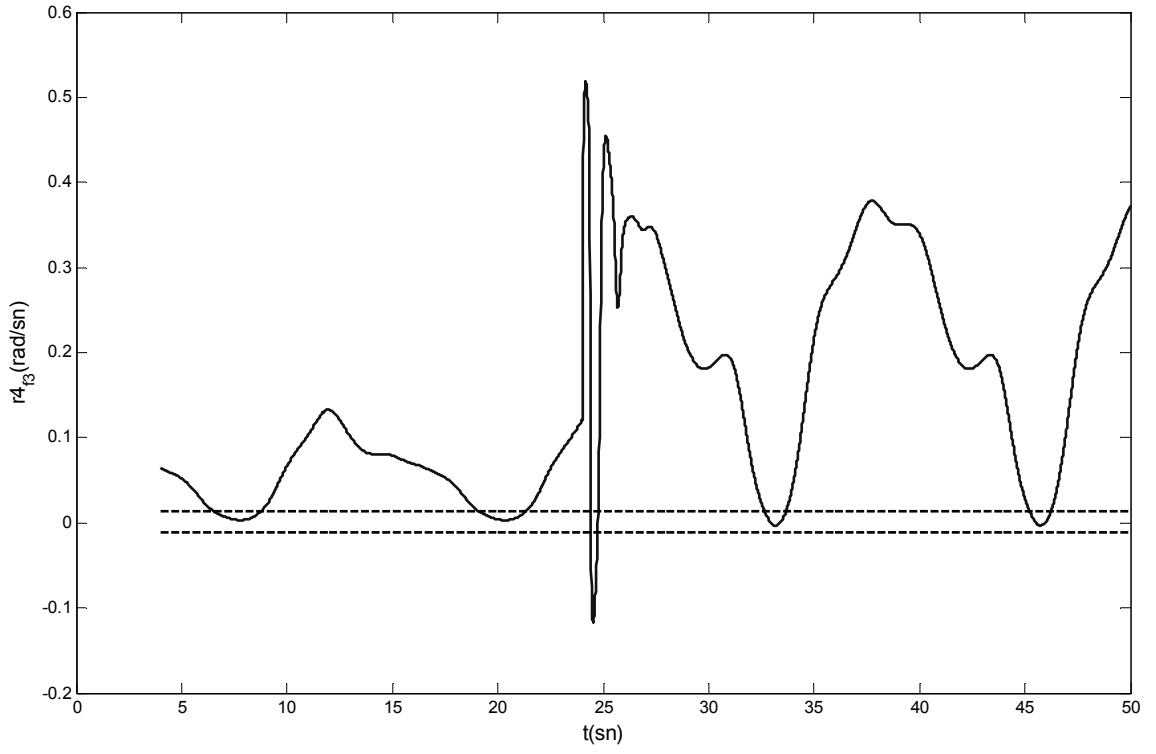
Şekil 5.22. 2. şema için artık  $r_{1f_3}$



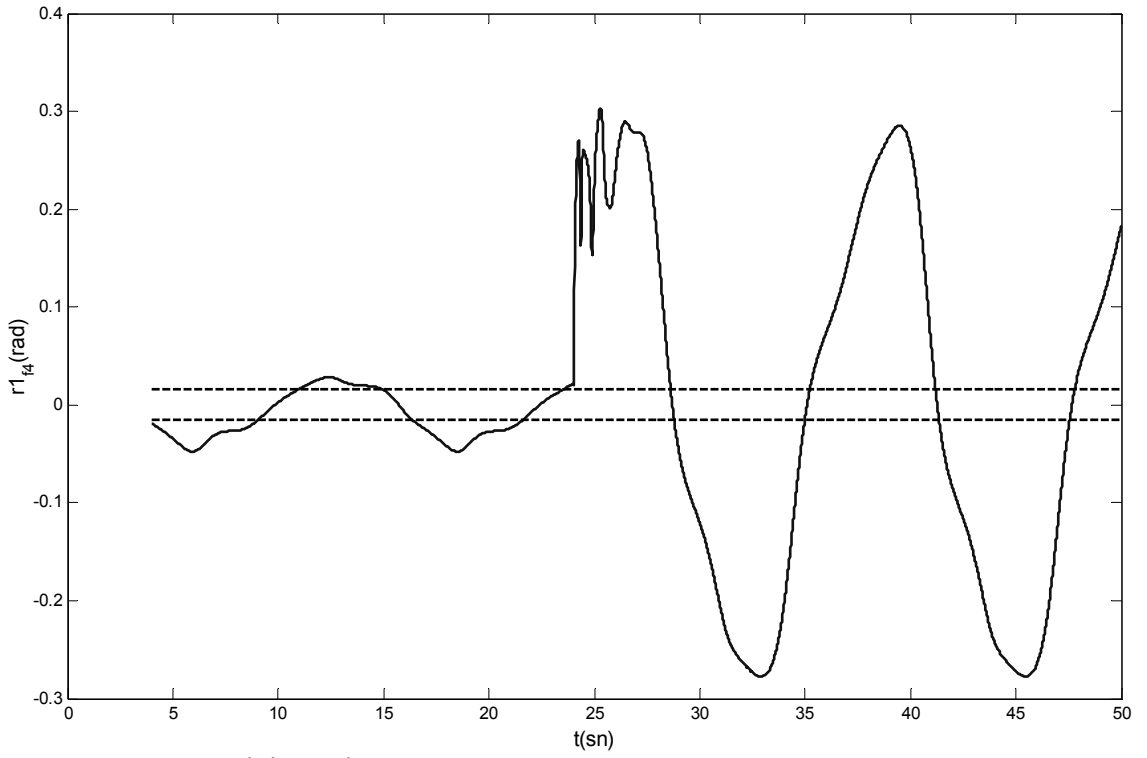
Şekil 5.23. 2. şema için artık  $r_{2f_3}$



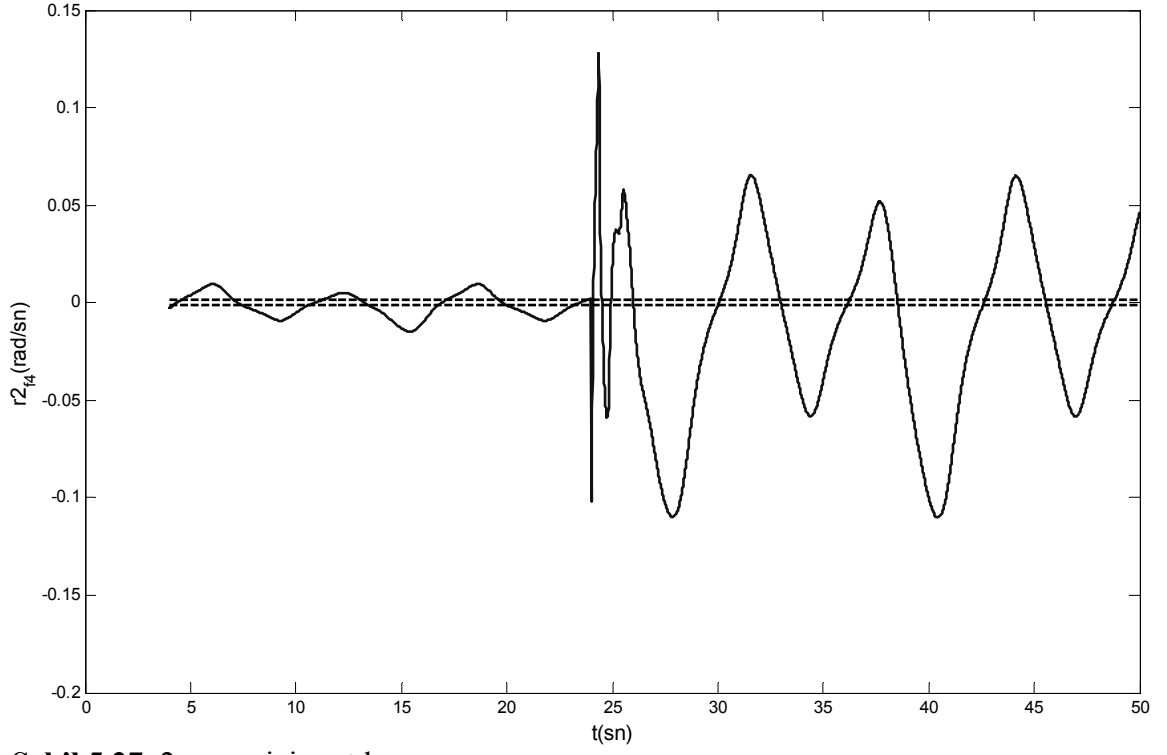
Şekil 5.24. 2. şema için artık  $r_{3f_3}$



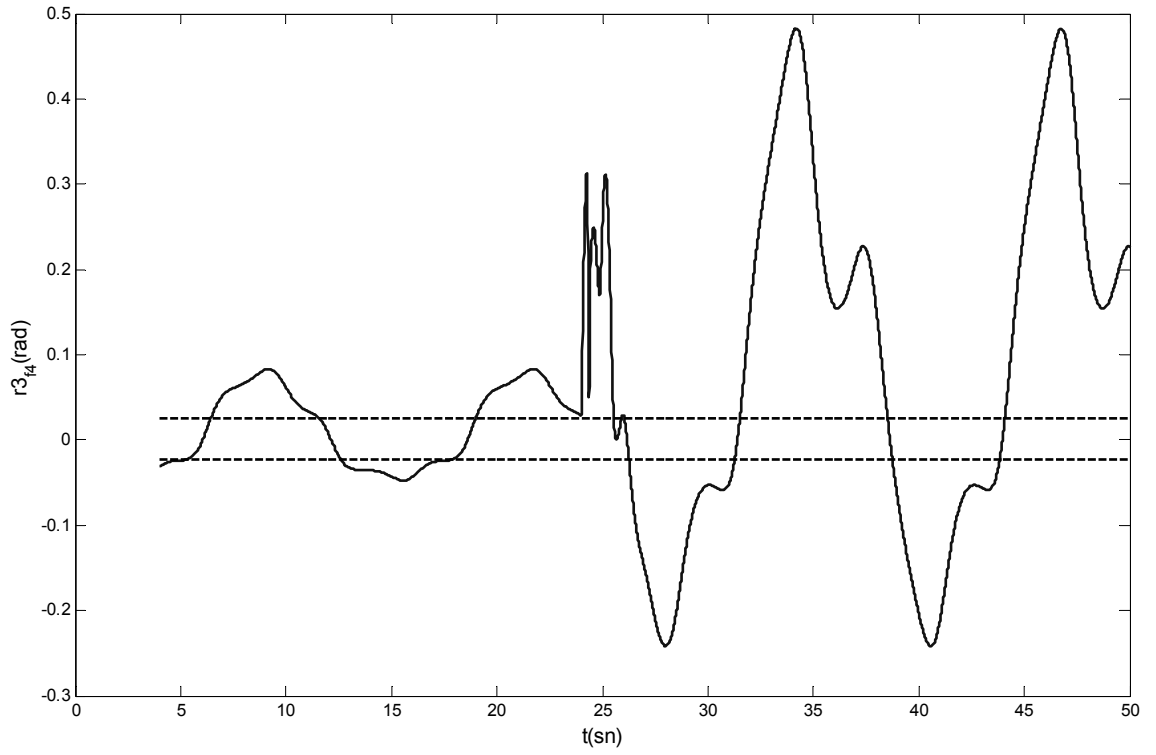
Şekil 5.25. 2. şema için artık  $r_{4f_3}$



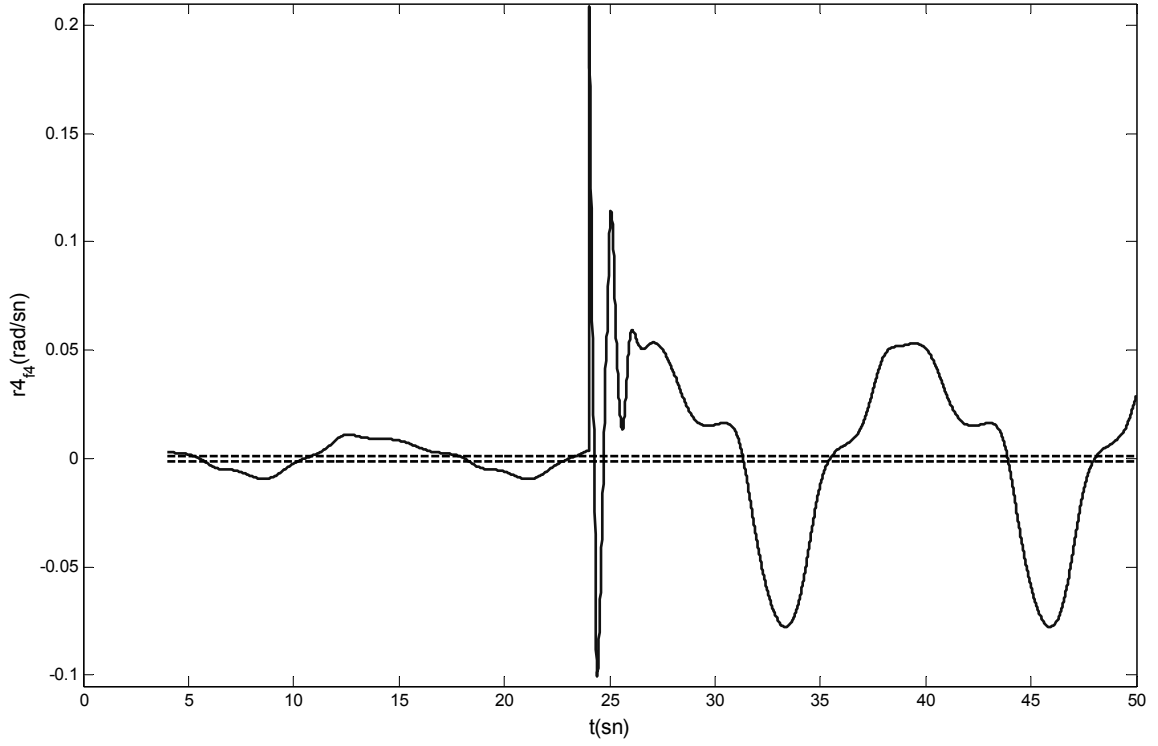
Şekil 5.26. 2. şema için artık  $r_{1f_4}$



Şekil 5.27. 2. şema için artık  $r_{2f_4}$

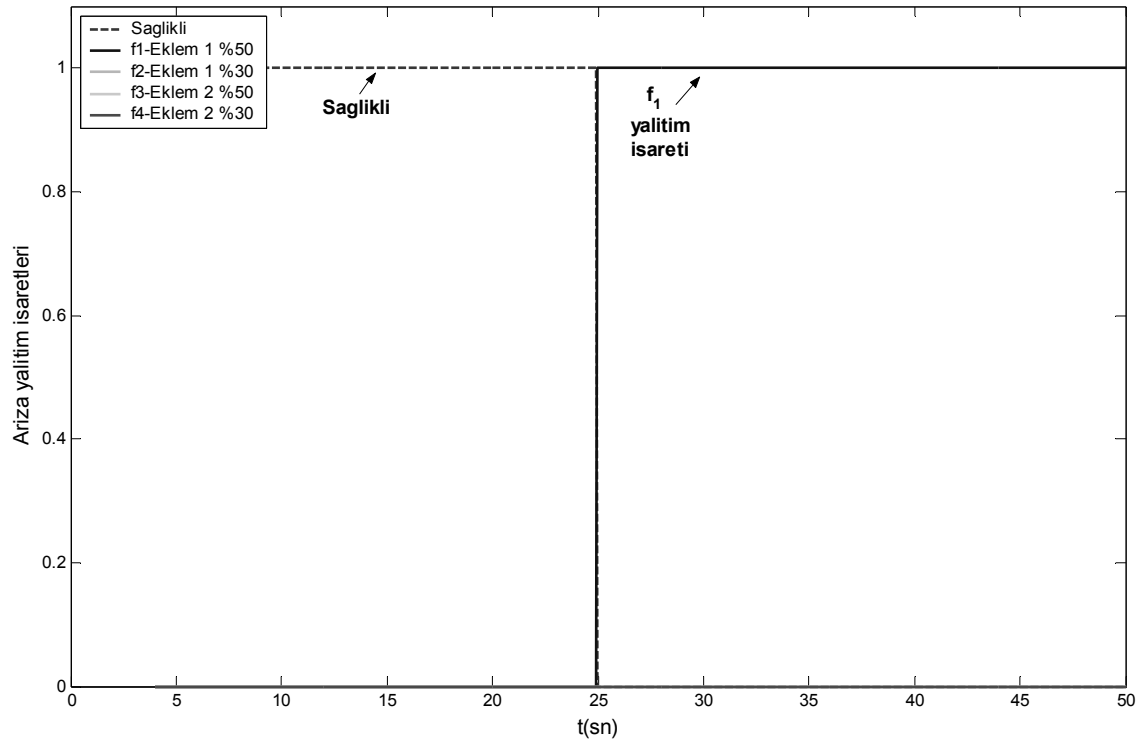


Şekil 5.28. 2. şema için artık  $r_{3f_4}$



Şekil 5.29. 2. şema için artık  $r_{4f_4}$

Şemaya ait arıza yalıtım işareti Şekil 5.30'da verilmiştir. Şekil 5.30 incelendiğinde  $t = 24.93$  sn. anının arıza yalıtım anı olduğu görülmektedir. Önerilen 2. FDI şemasına ait MATLAB programı ve akış diyagramı Ek D'de verilmiştir.



Şekil 5.30. 2. şema için arıza yalıtım işaretleri

### 5.3. Önerilen 3. FDI Şemasına Ait Benzetim Sonuçları

Bu altbölümde Şekil 4.7 ile Bölüm 4.3'te önerilen 3. FDI şemasına ait benzetim sonuçları verilecektir.

Robot modelleme, artık ve analitik fazlalık türetimi, arıza bulma işlemleri 1. şema ile aynı şekilde yapılmaktadır. Önerilen şema, 2. şema için seçilen 80 yörünge için denenmiş ve artık ve analitik fazlalıklar için arıza bulmada kullanılan artık eşikleri elde edilmiştir.

Arıza işlevine yakınsamak için her biri bir ekleme ait arıza işlevini veren 2 bağımsız ANFIS'den oluşan bir M-ANFIS oluşturulmuştur. Bu M-ANFIS Bölüm 4.3'te anlatıldığı üzere giriş olara anlık moment, açısız konum ve hız işaretlerini  $(\tau_1(t), \tau_2(t), q_1(t), q_2(t), \dot{q}_1(t), \dot{q}_2(t))$  giriş olarak almaktadır. M-ANFIS'e verilecek olan bu işaretler arıza bulma işareti ile etkin duruma geçmektedir. M-ANFIS'in eğitiminde robot manipülatöre ait tanımlanacak arızalara daha iyi bir yakınsamak ve sadece tanımlanan arızalara değil tüm arıza aralığına yakınsayabilmek için ek olarak 2 arıza daha tanımlanmıştır. Böylece M-ANFIS'in eğitiminde kullanılan tanımlı arıza sayısı 6'ya çıkarılmıştır. Tanımlanan bu arızalarla birlikte oluşan arıza çizelgesi Çizelge 5.3'te verilmiştir.

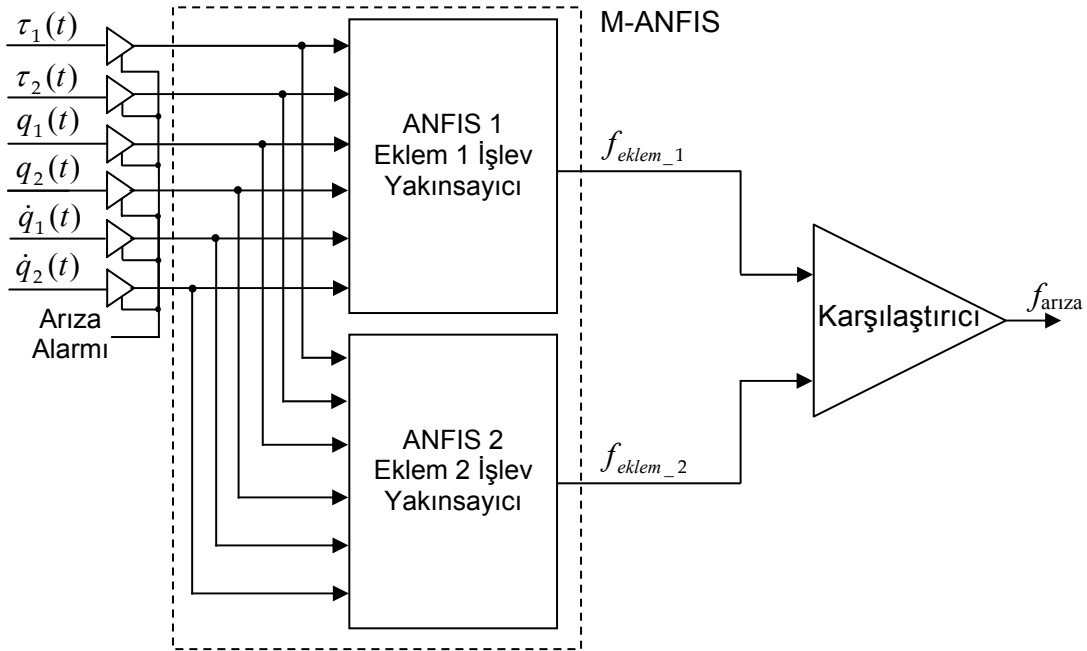
**Çizelge 5.3.** 3. şema için tanımlanan arızalar ve isimlendirilmesi

Arıza	İsmi
Eklem 1 %50 kayıp	$f_1$
Eklem 1 %30 kayıp	$f_2$
Eklem 2 %50 kayıp	$f_3$
Eklem 2 %30 kayıp	$f_4$
Eklem 1 %70 kayıp	$f_5$
Eklem 2 %70 kayıp	$f_6$

M-ANFIS eğitimi için bu 6 arıza için benzetimler yapılmış ve 101 örnekle veriler eğitim için kullanılmıştır.

İşlev yakınsayıcı M-ANFIS içindeki ANFIS'ler her bir ekleme ait arıza çıkışını vermesi bir sorunu birlikte getirmektedir. Arızasız ekleme ait ANFIS'in belirsizliklerden dolayı doğrudan sıfır çıkışı veremeyeceğinden arıza işlevine yakınsama

yapıldığı anda her iki çıkış da çeşitli işlev değerleri üretebilmektedir. Bu durumda işlev doğru olsa da hangi işlev çıkışının kullanılacağı sorunu ortaya çıkmaktadır. Bu sorun yapılan benzetimlerdeki gözlemler sonucu görülen ve beklenen “Daha büyük genlikli işaret doğru işarettir” kuralı ile çözülmüştür. Bu kural tanımı ise arızalı ekleme ait ANFIS çıkışının arıza işlevine yakınsayarak için büyük değerli olacağı ve diğer ANFIS çıkışının düşük değerli bir işlev vermesi gerektiği yaklaşımıyla yapılmıştır. Bu kural sadece her iki çıkışın birbirine çok yakın olduğu düşük yüzdeli ve buna bağlı düşük genlikli arızalarda yanlış işlev çıkışının seçilmesine sebep olmaktadır. Şekil 5.31’de 3. şemada M-ANFIS ile arıza işlevine yakınsama ve arıza yalıtımı gösterilmiştir.



Şekil 5.31. 3. FDI şeması için M-ANFIS ile arıza yakınsama ve arıza yalıtımı

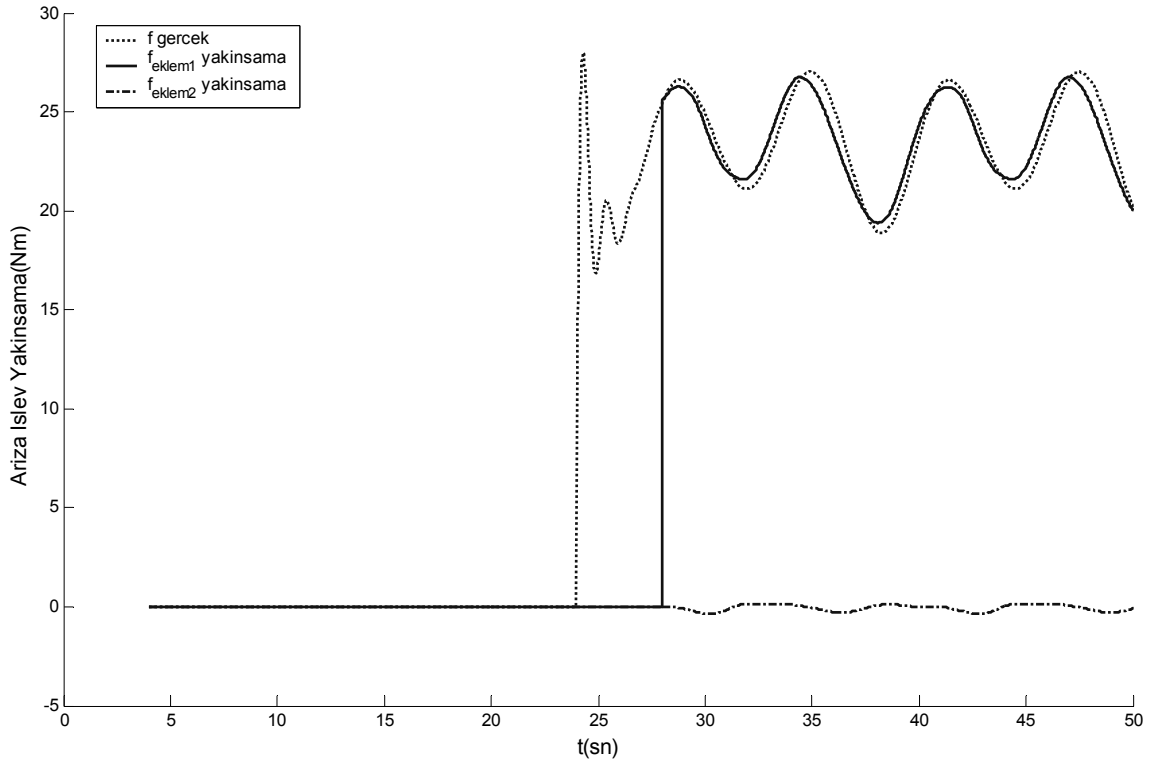
Tanımlanan arızaların genişletilmesi ise başka bir sorunu ortaya çıkarmıştır. Yüksek yüzdeli arıza durumlarında manipülâtör dizgesi kararlı bölgeden kararsız bölgeye geçerek yüksek frekans bileşenleri içeren moment işaretlerine sebep olabilmektedir. Yüksek frekans bileşenlerinin getirdiği hızlı değişimler sonuçta bir işlev yakınsayıcı olan M-ANFIS yapısını yanıltabilmekte ve yanlış ekleme dair arıza işlevi seçilebilmektedir. Momentteki bu yüksek frekans bileşenleri HM-PID denetime ek olarak kayan kipli denetime benzer şekilde bir süzgeç ve buna bağlı doyum ögesi ile giderilse de arıza yalıtımında benzer sorun devam etmiştir. Bu sebeple bu sorun dizgenin bir parçası olarak kabul edilmiştir.

1. şemaya benzer şekilde M-ANFIS’in arıza oluştuğunda geçici durumdan sürekli

duruma geçerken bu geçici durumdan etkilenmemesi için arıza bulma anından 3.2 sn. sonra devreye girmesi sağlanmıştır.

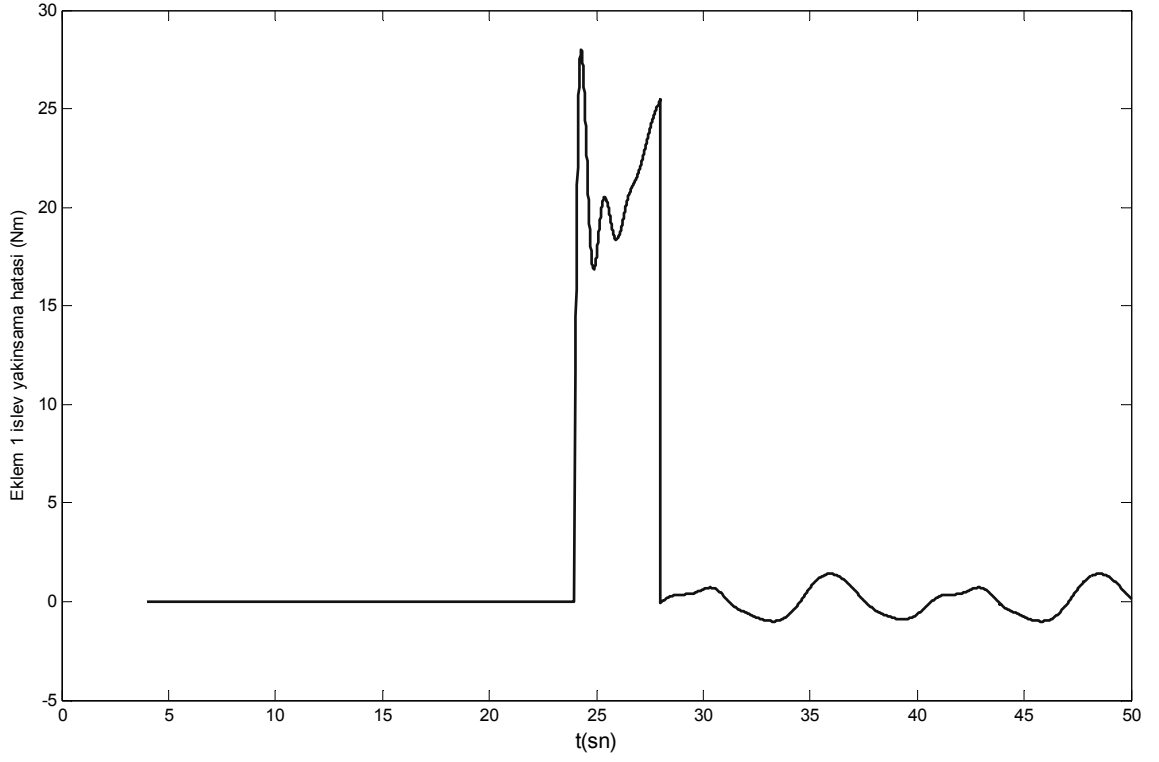
Bu şema için de arıza yalıtım yüzdelerini yükseltebilmek için 0.8 saniyelik süreklilik ölçütü benzetime eklenmiştir.

Tanımlanan bu özelliklerle, tanımlanan arızalar ve verilen yörüngeler için %87.083 arıza yalıtım yüzdesi elde edilmiştir. Örnek olarak 1. şema için verilen Eş. (5.7)'deki yörünge için aynı arıza ( $f_1$ ,  $t = 24$  sn. anında) tanımlanmıştır. Eklemlerin izleyeceği yörüngeler, yapacakları hatalar, uygulanan momentler, türetilen artıklar ve analitik fazlalıklar aynı olacağından (eğitimde kullanılan yörünge sayısı değişimine bağlı artık eşikleri değişse de) bu işaretlere ait şekiller tekrar verilmemiştir. 4-50 sn. arasında sırasıyla arıza işlevi arıza yakınsama işlevleri Şekil 5.32'de, Eklem 1'e ait yakınsama işlev hatası Şekil 5.33'te verilmiştir.

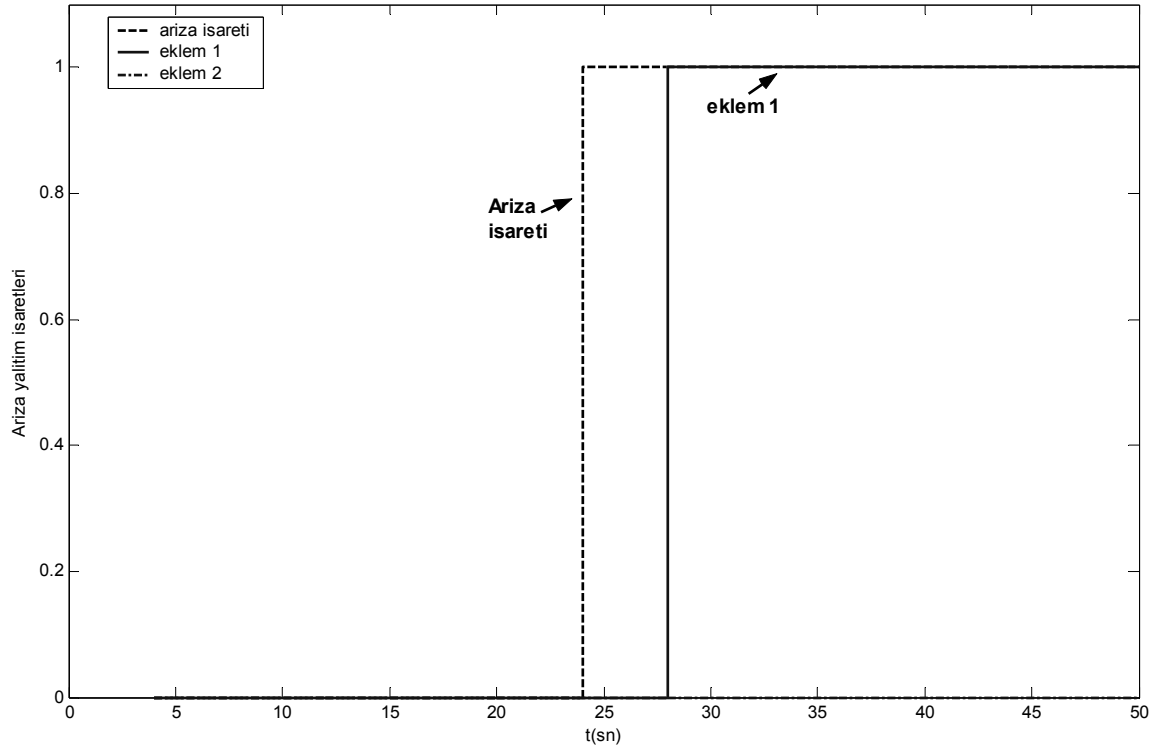


**Şekil 5.32.** 3. FDI şeması için M-ANFIS ile arıza yakınsama işlevi

Şemaya ait arıza yalıtım işareti Şekil 5.34'te verilmiştir. Şekil 5.34 incelendiğinde  $t = 28$  sn. anının arıza yalıtım anı olduğu görülmektedir. Önerilen 3. FDI şemasına ait MATLAB programı ve akış diyagramı Ek E'de verilmiştir. Bu örnek için arıza yalıtımından sonra en büyük hata miktarları  $\max(|e_1(t)|) = 0.006459$  rad.,  $\max(|e_2(t)|) = 0.017502$  rad. olarak elde edilmiştir.



**Şekil 5.33.** 3. FDI şeması için Eklem 1 arıza işlevine yakınsama hatası



**Şekil 5.34.** 3. FDI şeması için arıza yalıtım işaretleri

#### 5.4. Önerilen FTC Şemasına Ait Benzetim Sonuçları

Bu altbölümde Şekil 4.9 ile Bölüm 4.4'te önerilen FTC şemasına ait benzetim sonuçları verilecektir. Önerilen şema 3. FDI şemasının bir uzantısı olarak çalıştığından FDI kısmı ile ilgili bilgiler tekrar verilmemiştir.

Önerilen FTC şeması Bölüm 4.4'te belirtildiği üzere manipülâtör denetimi için HM-PID yöntemini kullanmakta ve kazanç çizgeleme yöntemiyle  $K_p$ ,  $K_i$ ,  $K_v$  katsayılarının güncellenmesiyle manipülâtörün arızalı durumda daha düşük hata değerleri ile çalışmasını sağlamaktadır. FTC elemanı olan kazanç çizgeleme bloğu arıza işlevine yakınsama işlemi yapılırken arıza yalıtım anında devreye girmektedir. Burada katsayıların hangi kurala göre güncelleneceği kullanıcı tarafından belirlenir. Önerilen FTC şeması için bu kural Eş. (5.9) ile tanımlanmıştır:

$$K_{güncel} = K + K \cdot \frac{f_{arıza}}{\tau_{arızalı ekleme uygulanan moment}} = K + K \cdot FTC \text{ oranı} \quad (5.9)$$

Bu kural, Eş. (2.3)'te tanımlı arızalı durumdaki dinamikler ve HM-PID yönteminden arızalı ekleme ait moment değerindeki arızaya bağlı artışın PID katsayılarına da yansıtılması gerektiği yaklaşımından yararlanılarak oluşturulmuştur. Tanımlanan kural yerine Zhao ve arkadaşlarının kullandığına benzer bulanık mantık tabanlı ve hata ve hatanın türevini giriş olarak alan bir kazanç çizgeleme bloğu oluşturulmuş fakat istenen başarımlar elde edilememiştir (Zhao ve ark., 1993).

Tanımlanan güncelleme kuralı Bölüm 4.4'te bahsedilen üç sorunu taşımaktadır. Bu sorunların etkilerini göstermek amacıyla Eş. (5.7)'deki yörüngede  $f_1$  ve  $f_5$  arızaları durumunda benzetimler yapılmış ve güncelleme yapılmadan, 1 kez güncelleme

**Çizelge 5.4.**  $q_{d1}(t) = 0.8 * \cos(t/2) \text{ rad.}$ ,  $q_{d2}(t) = 0.8 * \sin(t/2) \text{ rad.}$  için Eklem 1'de %50 arıza ( $f_1$ ) ( $FTC \text{ oranı} = 0.50157$ ) durumunda güncelleme karşılaştırması

Güncelleme miktarı	0	1 kez güncelleme	Sürekli güncelleme
Aşım (geçici durum) $\max( e_1(t) )$ (28.00-37.00 sn)	0.006458	0.034347	0.034347
Aşım (geçici durum) $\max( e_2(t) )$ (28.00-37.00 sn)	0.017500	0.056559	0.056559
Sürekli durum hatası $\max( e_1(t) )$ (37.00-50.00 sn)	0.006459	0.004083	0.004164
Sürekli durum hatası $\max( e_2(t) )$ (37.00-50.00 sn)	0.017502	0.011166	0.011361

**Çizelge 5.5.**  $q_{d1}(t) = 0.8 * \cos(t/2)$  rad.,  $q_{d2}(t) = 0.8 * \sin(t/2)$  rad. için Eklem 1’de %70 arıza ( $f_5$ ) (FTC oranı = 0.69804) durumunda güncelleme karşılaştırması

Güncelleme miktarı	0	1 kez güncelleme	Sürekli güncelleme
Aşım (geçici durum) (28.00-37.00 sn) $\max( e_1(t) )$	0.137060	0.124871	0.124871
Aşım (geçici durum) (28.00-37.00 sn) $\max( e_2(t) )$	0.312311	0.189688	0.189688
Sürekli durum hatası (37.00-50.00 sn) $\max( e_1(t) )$	0.117481	0.010151	0.010182
Sürekli durum hatası (37.00-50.00 sn) $\max( e_2(t) )$	0.269350	0.026525	0.026615

ve sürekli güncelleme durumlarındaki arıza anından 4 sn. sonraki geçici durum ve sürekli durum hatalarını içeren karşılaştırma Çizelge 5.4 ve Çizelge 5.5’te verilmiştir. Bu örnek için KÇ bloğu arıza yalıtım anı olan  $t = 28$  sn. anında devreye girmektedir.

Birinci sorun katsayı değişimi yani FTC’nin etkileridir. Çizelge 5.4 ve 5.5 gözönüne alındığında ve güncelleme yapılıp yapılmaması durumları (0-1 kez güncelleme sütunları) incelendiğinde göze çarpan sürekli durum hatalarının her arızada belirgin şekilde azaldığı, geçici durum hatalarının ise PID katsayılarının değişim anında bir aşım yaparak yerleştiğidir. Özellikle manipülâtörün çalışma bölgesini değiştirmesinden kaynaklı salınımlar bu aşımaların daha üst seviyelere çıkmasına sebep olmaktadır. Bu durumda kullanıcı geçici durumda ortaya çıkan büyük hatalara karşılık küçülen sürekli durum hatalarına göre bir seçim yapmalıdır.

İkinci sorun ise güncel katsayıların bulunma işleminin belirli bir sıklıkta mı yoksa bir kez mi yapılacağı sorunudur. Güncelleme istenirse 1 kez, istenirse belirli saniye aralıklarla yapılabilir. Katsayıların belirli bir sıklıkta güncellenmesini uygulamak amacıyla arıza bulma ve yalıtımındaki zaman gecikmesi gözönünde bulundurularak bulunan en düşük süre olan 4 sn. aralıklarla güncelleme yapılması denenmiştir. Bu durumda sürekli durum hata miktarlarının 1 kez güncellemeye göre az da olsa büyüdüğü görülmektedir. Bunun sebebi ise sürekli olarak katsayıların değişmesinin her bir güncellemeyle ortaya çıkan geçici duruma bağlı aşımardan dolayı hata miktarlarının artmasıdır. Bu sebeple katsayı güncelleme değerinin 1 kez bulunarak devam edilmesi tercih edilmiştir.

Üçüncü sorun ise katsayı değişiminin ani olarak gerçekleştirilmesine bağlı hata miktarındaki artıştır. Bu durumu değiştirmek amacıyla zamana bağlı olarak denemeler sonucu uygun olarak bulunan 0.4 sn.de 1 kez artarak ilerleyen bir PID değişim kuralı tanımlanmış ve bu kural Eş. (5.10) ile verilmiştir:

$$t = t_{yalitim} : 0.4 : t_{son} \text{ için } (artim = 0.05, 0.1, 0.15, 0.2, 0.25)$$

$$K(t) = \begin{cases} K + K * artim * ((t - t_{yalitim} + 0.4) / 0.4) & K(t) < K_{güncel} \\ K_{güncel} & K(t) \geq K_{güncel} \end{cases} \quad (5.10)$$

Bu kural tanımına bağlı olarak elde edilen hata sonuçları Çizelge 5.6 ve 5.7'da verilmiştir.

**Çizelge 5.6.**  $q_{d1}(t) = 0.8 * \cos(t/2) \text{ rad.}$ ,  $q_{d2}(t) = 0.8 * \sin(t/2) \text{ rad.}$  için Eklem 1'de %50 arıza ( $f_1$ ) (FTC oranı = 0.50157) durumunda zamana bağlı artım karşılaştırılması

Artım miktarı	0	0.05	0.1	0.15	0.2	0.25	1 kez güncelleme
Aşım (geçici durum) (28.00-37.00 sn) $\max( e_1(t) )$	0.006458	0.006009	0.009021	0.013249	0.017098	0.020601	0.034347
Aşım (geçici durum) (28.00-37.00 sn) $\max( e_2(t) )$	0.017500	0.013880	0.014751	0.021628	0.027922	0.033688	0.056559
Yerleşme zamanı(sn)	-	4.0	2.0	1.2	0.8	0.8	-
Sürekli durum hatası (37.00-50.00 sn) $\max( e_1(t) )$	0.006459	0.004083	0.004083	0.004083	0.004083	0.004083	0.004083
Sürekli durum hatası (37.00-50.00 sn) $\max( e_2(t) )$	0.017502	0.011166	0.011166	0.011166	0.011166	0.011166	0.011166

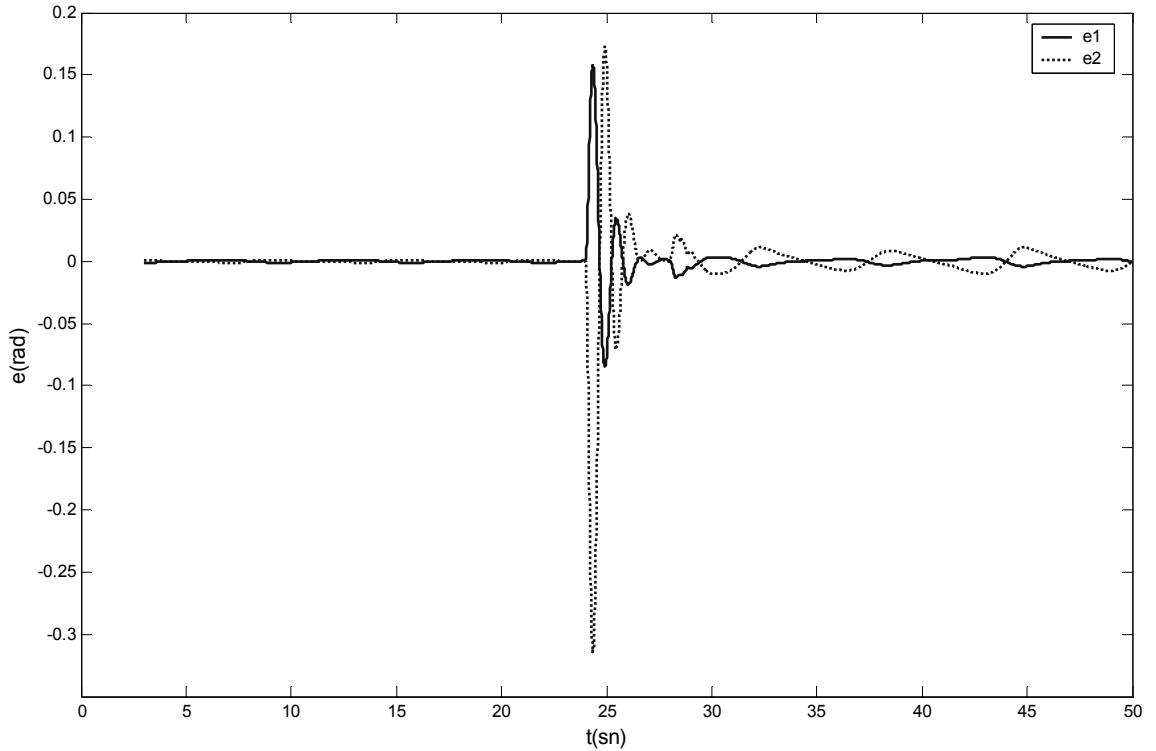
Üçüncü sorunun yanısıra bazı yörüngelerde FTC'nin dizgeyi kararsız duruma götürdüğü, yine bazı durumlarda FTC'nin yanlış arıza işaretinin verildiği durumu ortadan kaldırarak doğru bir şekilde arıza yalıtımını sağladığı görülmüştür. Bunun sebebi ise PID katsayılarının değişim anındaki yakınsamaya bağlı olarak FTC oranı değerinin doğru bulunup bulunamamasıdır. Eş. (5.9) ile tanımlı oranın doğru yakınsayamamadan kaynaklı PID katsayılarının değişimine iyi veya kötü şekilde yansması bu şekilde sonuçlar ortaya çıkarmaktadır.

Daha önce tanımlanan aynı örnek yörünge ve arızada  $artim = 0.15$  için 4-50 sn.

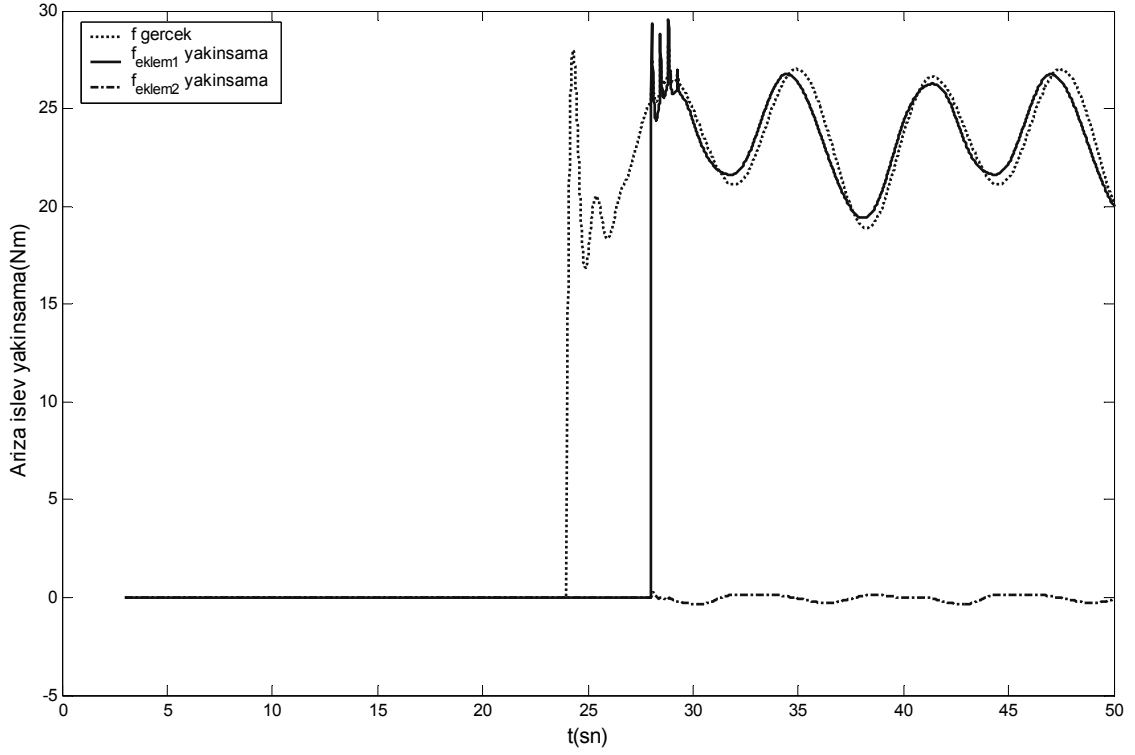
arasında sırasıyla eklemelerin yaptığı yörünge hataları Şekil 5.35'te, eklemelerin arıza işlevine yakınsama işlevleri 5.36'da ve Eklem 1 için arıza işlevine yakınsama hatası Şekil 5.37'de verilmiştir.

**Çizelge 5.7.**  $q_{d1}(t) = 0.8 * \cos(t/2) \text{ rad.}$ ,  $q_{d2}(t) = 0.8 * \sin(t/2) \text{ rad.}$  için Eklem 1'de %70 arıza ( $f_2$ ) (FTC oranı = 0.69804) durumunda zamana bağlı artım karşılaştırılması

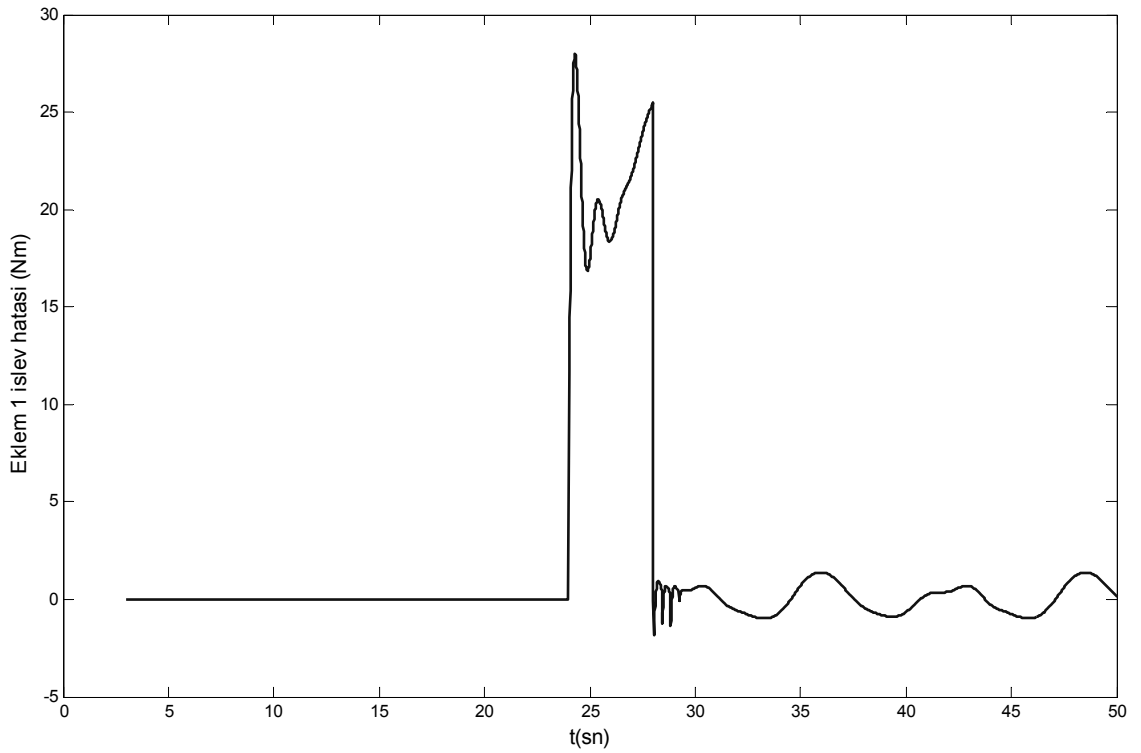
Artım miktarı	0	0.05	0.1	0.15	0.2	0.25	1 kez güncelleme
Aşım (geçici durum) (28.00-37.00 sn) $\max( e_1(t) )$	0.137060	0.094003	0.093887	0.093820	0.098254	0.102767	0.124871
Aşım (geçici durum) (28.00-37.00 sn) $\max( e_2(t) )$	0.312311	0.115446	0.115446	0.124254	0.134398	0.143119	0.189688
Yerleşme zamanı(sn)	-	5.2	2.4	1.6	1.2	0.8	-
Sürekli durum hatası (37.00-50.00 sn) $\max( e_1(t) )$	0.117481	0.010151	0.010151	0.010151	0.010151	0.010151	0.010151
Sürekli durum hatası (37.00-50.00 sn) $\max( e_2(t) )$	0.269350	0.026525	0.026525	0.026525	0.026525	0.026525	0.026525



**Şekil 5.35.** FTC şeması için eklemelerin yaptığı yörünge hataları



**Şekil 5.36.** FTC şeması için eklemlerin arıza işlevine yakınsama işlevleri



**Şekil 5.37.** FTC şeması için Eklem 1 arıza işlevine yakınsama hatası

Önerilen FTC şemasının zamanla değişen yaklaşıma ait MATLAB programı ve akış diyagramı Ek F’de verilmiştir.

## 6. ÖNERİLEN ŞEMALARIN TMS320C6713 DSP ÜZERİNE GÖMÜLMESİ VE BULGULAR

Tez çalışması kapsamında önerilen şemalar gerçek bir robot manipülatör için kullanılmak istendiğinde mutlaka bir DSP'ye gömülmesi gerekmektedir. Gömülü DSP, bir robot manipülatör denetleyici panosuna doğrudan monte edilebilir, ayrıca bilgisayardan farklı olarak işlemci hızı ve arayüz destekleri endüstriyel bir otomasyon dizgesine kolaylıkla uyarlanabilmesini sağlar.

Gömme işlemi için iki farklı yaklaşım benimsenmiştir:

- Manipülatör, manipülatör denetleyici ve FDI-FTC şemalarının tümünden DSP üzerine gömülmesi (Tümleşik robot manipülatör denetim ve gözetleme dizgesi)
- Manipülatör ve manipülatör denetleyicinin bilgisayar, FDI-FTC şemalarının DSP üzerine gömülmesi (Robot manipülatör gözetleme dizgesi)

Gömme işlemi için bu iki farklı yaklaşımın benimsenmesinin sebebi işlem yükü açısından tüm denetim ve FDI-FTC şemalarını kendi üzerinde toplayabilecek bir DSP programı oluşturarak işlem yükünü karşılayıp karşılayamayacağını denemek ve gerçek bir robot manipülatör dizgesinde robotu denetleyen denetleyici kart veya bilgisayar ile FDI-FTC şemalarının farklı bir kartta olmasını birebir benzetmektir.

Tez çalışmasında önerilen FDI-FTC şemaları yüksek işlem gücüne ihtiyaç duymakta ve robot manipülatöre ait açısız konum ve hız işaretlerini başka bir kart üzerinden almak için sayısal giriş/çıkışların yanında analog giriş/çıkışlara sahip bir DSP'ye ihtiyaç duyulmaktadır. Ayrıca kullanılacak DSP yüksek hassasiyette hesaplamalar yapması açısından kayan noktalı bir DSP olmalıdır. Bu istekler doğrultusunda kullanılmak üzere Texas Instruments TMS320C6713 DSP kartı seçilmiştir. Özellikle ses işlemede kullanılan bu DSP, 2 stereo analog girişe(4 analog giriş) ve 2 stereo analog çıkışa(4 analog çıkış) sahiptir.

TMS320C6713 DSP'ye ait diğer özellikler aşağıdaki gibidir ve bu DSP'ye ait görünüm Şekil 6.1'de verilmiştir (Chassaing, 2005):

- 225 Mhz kayan nokta işlemci
- 32 bit AIC23 stereo codec (8-96 KHz arası örnekleme frekansı)
- 256 KB Flash Bellek
- 16 MB SDRAM
- Denetlenebilir 4 LED
- USB arayüz

- CCStudio v3.1 ile C dilinde programlanabilme (Windows XP)
- MATLAB (v3.1 için sadece MATLAB R2007a ile), Visual C++ ve Visual Basic ile gerçek zamanlı çift yönlü haberleşme (RTDX)



**Şekil 6.1.** TMS320C6713 DSP'ye ait görünüm

DSP programlama için kullanılan CCStudio v3.1 programı C dilinde programlama amaçlı oluşturulmuş bir kod geliştirme yazılımıdır. Kendi içinde C derleyici, makine kod çevirici ve DSP'ye yükleyici içeren bu ortam içindeki gerçek zamanlı veri izleme ve veri çizim özellikleriyle kolay bir kod yazım ortamı sağlamaktadır. Tez çalışmasında kullanılan ve DSP geliştirme seti ile birlikte gelen v3.1 haricinde ayrıca satın alınabilecek, özellikle Windows Vista uyumlu v3.3 ve v4.0 sürümleri de bulunmaktadır.

Gömme işlemleri ile ilgili ayrıntılara geçmeden ortak özellikler hakkında bilgi vermek gerekirse öncelikle DSP'de kullanılan kayar nokta sayı düzeninden bahsetmek gerekir. TMS320C6713 DSP sayı düzeni olarak kayar noktalı sayılar için float (32 bit) ve double (64 bit) tanımlamalar yapılabilmektedir. Sayı düzeni ve buna dayalı tüm programın yazımında float kullanılması durumunda MATLAB altındaki benzetimlerle noktadan sonraki basamaklarda uyumsuzluklar görülmekte (örn.  $\sin(x)$  hesabında noktadan sonraki 9. basamaktan itibaren), bu uyumsuzluklara bağlı olarak MATLAB benzetimleri ile aynı sonuçlar elde edilememektedir. Bu sorun ise float yerine MATLAB'ta da kayan noktalı sayı tanımı için kullanılan double kullanılarak

çözümüştür. Bu kullanım ise daha fazla bellek alanı kullanımı ve daha uzun program yürütme zamanı olarak geri dönmüştür.

Gömülü DSP için öncelikle ANFIS ve sinir ağı yapılarını oluşturmak amacıyla bu yapıların MATLAB altında elde edilen katsayıları, C başlık (header) kütüphanelerine kaydedilmiştir. Bu yapılara ait katsayıları tez çalışmasında yer verilmemiştir. Programlarda ANFIS ve sinir ağı gerçekleştirme için alt programlar yazılmıştır. Bu altprogramlar şemalardakiyle aynı şekilde moment, açısal konum, hız veya artıkları ve yapıya ait katsayıları giriş olarak almakta ve şemalardaki uygun çıkışları vermektedir.

DSP üzerine program yazımında donanım biçimlendirmesi (konfigürasyon) özellikle gözetleme işlemi gibi büyük vektörlere veya matrislere ihtiyaç duyulan uygulamalarda uyumsuzluk sorunu yaratmaktadır. Bu sorunun aşımı bellek bölgelerinin tanımlandığı “C6713DSK.cmd” adlı dosyada gerekli değişiklikler yapılarak flash bellek haricindeki SDRAM de kullanıma alınmıştır.

CCStudio programı double sayı düzeni kullanıldığında elde edilen verilerin sadece izlenmesine izin vermekte, çizime sadece diğer sayı düzenlerine (short, long, float vb.) izin vermektedir. Bu sebeple elde edilen onaltı sayı tabanındaki (hex) veriler MATLAB altında onlu sayı tabanına dönüştürülmüştür ([http://en.wikipedia.org/wiki/Double\\_precision\\_floating-point\\_format](http://en.wikipedia.org/wiki/Double_precision_floating-point_format), 2009). Bu dönüşüme eski sürümler izin vermediğinden ilgili dönüşüm programı MATLAB R2007a altında yazılmıştır.

Önerilen ikinci yaklaşım için bilgisayar programı olarak öne MATLAB denenmiş, MATLAB’ın kendi getirdiği haberleşme yükünden dolayı gerçek zamanlı çalışmaya daha uygun olması açısından daha sonra Visual Basic 6.0 ile gömme çalışmaları yapılmıştır.

MATLAB ve Visual Basic ile gerçek zamanlı haberleşmeyi sağlayan programlarda aşağıdaki ayarlamalar yapılmalıdır:

- DSP ile gerçek zamanlı haberleşme için CCStudio içinde Tools→RTDX→Enable RTDX seçeneği işaretlenmelidir.
- Visual Basic-DSP haberleşmesi sırasında gerekli “C6713DSK.cmd” dosyası çıkarılıp onun yerine konfigürasyon ayarlarını tutan “Configuration1.cdb” dosyası eklenmelidir. Ayrıca csl6713.lib, dsk6713bsl.lib ve rts6700.lib kütüphaneleri programa eklenmelidir.
- Haberleşme için Tools→RTDX içinde size of buffer = 640000’e ayarlanarak haberleşmenin sürekli olması sağlanmalıdır. Visual Basic’te yazılan program

göndereceđi verileri hızlı bir şekilde bilgisayarın belleđine yüklemekte ve yürütülmesini tamamlamaktadır. DSP ise bu verileri daha yavaş bir şekilde almakta ve uygun olduđu anlarda bilgisayardan kendi belleđine almaktadır.

Gömme işlemindeki yaklaşımlara program örneđi olarak manipölatör, manipölatör denetleyici ve FDI-FTC şemalarının tümünden DSP üzerine gömülmesi yaklaşımı için 1. şemaya ait C programı Ek G'de, manipölatör ve manipölatör denetleyicinin bilgisayar, FDI-FTC şemalarının DSP üzerine gömülmesi yaklaşımı için MATLAB-DSP arasında haberleşmeyi kullanan 2. şemaya ait MATLAB ve C programları Ek H'de, yine aynı yaklaşım için Visual Basic-DSP arasında haberleşmeyi kullanan 3. şemaya ait Visual Basic ve C programları Ek I'da verilmiştir.

## 7. SONUÇLAR, TARTIŞMA ve ÖNERİLER

Tez çalışmasının bu bölümünde benzetim çalışmaları ile elde edilen sonuçlara göre önerilen şemalar birbirleriyle karşılaştırılması ve gelecekte yapılabilecek devam çalışmaları hakkında bilgiler verilecektir.

Bu tez çalışmasında robot manipülatörler için yapay zeka yöntemlerini kullanan model temelli üç farklı FDI şeması ve bir FTC şeması önerilmiştir. Önerilen tüm şemalar manipülatör modelleme için M-ANFIS yapısını kullanmaktadır. Gerçek manipülatör dizgesi ile manipülatör modeli arasında elde edilen artık işaretleri ve bunlardan elde edilen analitik fazlalık işaretleri şemalarda arıza bulma için kullanılmaktadır.

Önerilen 1. şema, arıza bulma için artıkların ve analitik fazlalıkların kendileri için tanımlı eşiklerin aşımalarını kullanırken, arıza yalıtımı için bu işaretlerin giriş olarak verildiği bir sinir ağını kullanmaktadır. Önerilen 2. şema ise GOS yaklaşımından yararlanarak her biri bir arızayı tanımlayan arızalı manipülatör modellerini gerçek manipülatör dizgesiyle artık işaretleri elde edecek şekilde kullanmakta ve bu işaretler karar çizelgesiyle yorumlanarak aynı anda arıza bulma ve yalıtım işlemleri gerçekleştirilmektedir. Önerilen 3. şema ise yakınsayıcı M-ANFIS'ler ile arıza işlevine yakınsayarak diğer iki şemadan farklılaşmaktadır. Önerilen FDI şemalarının çeşitli FDI özelliklerine göre karşılaştırılması Çizelge 7.1'de verilmiştir.

**Çizelge 7.1** Önerilen FDI şemalarının karşılaştırılması

	<b>Yanlış Alarm Oranı</b>	<b>Güvenilirlik</b>	<b>Arıza Bulma Gecikmesi</b>	<b>Arıza Yalıtım Gecikmesi</b>	<b>Süreklilik Denemesi</b>
<b>1. Şema</b>	düşük	güvenilir	yok	var	var
<b>2. Şema</b>	düşük	güvenilir	yok	yok	var
<b>3. Şema</b>	düşük	güvenilir	yok	var	var
	<b>Yeni Yörüngeler Ekleme&amp;Eğitim</b>	<b>Hesapsal Yük</b>	<b>Arıza İşlevine Yakınsama</b>	<b>FTC için Kullanılabilirlik</b>	
<b>1. Şema</b>	kolay	düşük	sadece tanımlı arızalar	sadece tanımlı arızalar	
<b>2. Şema</b>	kolay	çok yüksek	sadece tanımlı arızalar	sadece tanımlı arızalar	
<b>3. Şema</b>	kolay	düşük	var	var	

Çizelge 7.1 incelendiğinde önerilen tüm şemaların yanlış alarm oranı açısından

düşük seviyelerde olduğu görülmektedir. Robot manipülatörler için yapılan çoğu FDI çalışmasında bu değerler verilmemekle birlikte özellikle veriye dayalı olarak öğrenen yapıların kullanıldığı bir FDI şemasında bu değer öne çıkmaktadır. Ayrıca bu değer önerilen bir şemanın başarımı için bir göstergedir.

Önerilen şemalar gürbüzlük açısından karşılaştırıldığında yapay zeka tekniklerine dayanmaları sayesinde güvenilir oldukları görülmektedir. Bu güvenilirlik teorik olarak çözümlenmese bile benzetimlerde yapılan denemelerde açıkça görülmektedir.

FDI şemalarında en fazla aranan özelliklerden biri arıza bulma ve yalıtım gecikmelerinin olmaması veya olabildiğince kısa olmasıdır. Önerilen şemalarda arıza bulma işlemleri artıkların ve analitik fazlalıkların eşik aşımalarının gözlemlenmesiyle yapılmaktadır. Arıza oluştuğunda varolan geçici durumdan etkilenmeyen bu değerlendirme işlemi sayesinde şemalarda herhangi bir arıza bulma gecikmesi yaşanmamaktadır. Bu sayede sadece arıza bulmanın önemli olabileceği bir manipülatör uygulamasında şemalar doğrudan kullanılabilir.

Arıza yalıtım aşamasına bakıldığında ise 1. ve 3. şemalarda artıkların ve analitik fazlalıkların değerlendirilmesi işleminin geçici durumdan etkilendiği görülmüş, bu sebeple bulunan uygun gecikme sürelerinden sonra arıza yalıtım işleminin başlaması sağlanmıştır. Özellikle gerçek zamanlı ve gerçek bir manipülatör üzerinde yapılacak bir çalışmada, bu gecikme süresine, vereceği gecikmiş tepkiden dolayı dikkat edilmelidir. 2. şema ise arıza bulma ve yalıtım işlemini aynı anda yapabilmesi sayesinde bu gibi bir gecikmeye gerek duymamaktadır.

Arızalı durumdaki artıkların ve analitik fazlalıkların sürekli olarak gözlemlendiği ve analitik yöntemlerin kullanılmadığı şemalarda bu işaretlerin sürekli olarak tanımlanan eşiklerin üstünde kalması, gerçekleşmesi zor bir yaklaşımdır ve düşük arıza yalıtım yüzdelerine sebep olmaktadır. Bundan dolayı tüm şemalarda artıkların değerlendirilmesinde süreklilik arama-deneme yaklaşımı kabul edilmiştir.

Önerilen şemalara ait veri kümeleri tüm yörünge kümesini kapsayacak şekilde tasarlanmıştır. Tanımlanan tipte arızalar gözönüne alınarak hazırlanan şemalarda veri kümesi haricinde yörüngeler manipülatöre uygulandığında ise açık olarak bir yaklaşım belirlenemese de,  $\pm 1$  genliği aşmayan yörüngelerde arıza yalıtımının sağlanabildiği fakat bu genliği aşan yörüngelerde, yörünge frekansının daha yüksek olduğu yörüngelerde ve farklı matematiksel tipte (örn.  $q_{d1}(t) = 0.8\cos(t/2)$ ,  $q_{d2}(t) = 0.8\cos(t/2)$ )

yörüngelerde arıza yalıtımında başarımlar sağlanamadığı görülmektedir. Bu sebeple bu şemaların kullanımı için öncelikle kullanılacak manipülâtör için tanımlı çalışma yörüngeleri ve çalışma uzayları hakkında ayrıntılı bilgi edinilmeli, bu bilgiye bağlı olarak kapsamlı bir veri kümesi hazırlanmalı ve uygulanmalıdır.

ANFIS ve M-ANFIS yapılarında kullanıcı tecrübesi Şekil 3.1’de tanımlı Katman 2 ile verilen kural kümesinde kural sayısının azaltılması için kullanılabilir (Jang ve ark., 1997). Bu çalışmada ise bu azaltma işlemi gözardı edilmiştir.

Önerilen bu şemalara herhangi bir anda yeni bir yörünge veya arıza eklemek ise oldukça kolaydır. Çalışma anı haricinde (offline) eğitilen yapay zeka yapıları eldeki yeni veri kümesine göre tekrar eğitilmeli, artıklara ve varsa analitik fazlalıklara ait eşikler, arıza yalıtım gecikme ve süreklilik deneme süreleri istenen arıza yalıtım yüzdesine erişecek şekilde tekrar gözden geçirilmelidir.

Önerilen bir FDI şemasının gerçekleştirilebilir olması için gerçek zamanlı olarak çalışabilir olması, bunun için de hesapsal yükünün az olması gerekmektedir. Hesapsal yük açısından bakıldığında 1. ve 3. şemaların çok ağır hesapsal yükler getirmediği, benzetim sürelerinden gerçek zamanlı çalışmaya yatkın oldukları ve kod iyileştirmeleriyle gerçek bir manipülâtör dizgesiyle tümleştirilebilecekleri görülmektedir. Bunlarla birlikte 2. şema her bir zaman adımında yürütmek ve çıkışını hesaplamak zorunda olduğu 20 farklı ANFIS sebebiyle çok büyük bir hesapsal yüke sahiptir. Bu özelliği ile 2. şemanın Chen ve Patton’un çalışmalarını destekler şekilde gerçek zamanlı çalışmaya uygun olmadığı görülmektedir (Chen ve Patton, 1999).

1. ve 2. şemanın en zayıf ve en yoruma açık taraflarından biri ise sadece kendileri için tanımlı arızalara tepki vermeleridir. Her ne kadar kendileri haricindeki arızalarda da dikkate değer yalıtım başarımları sağlasalar da bu durum özellikle arıza işlev yakınsamada ve onu takip eden arızaya dayanıklı denetimde sınırlayıcı bir özellik olarak ortaya çıkmaktadır. Gerçek zamanlı bir uygulamaya geçildiğinde çoğunlukla hareketlendirici arıza tipleri bilinmektedir ve her üç şema böyle bir çalışmada kullanılabilir. Ancak arıza işlevine yakınsama, arıza işlevinin nasıl değiştiğini görmek, denetim işareti olan moment işaretinin doymaya gidip gitmediğini gözlemlemek, nazik başarımlar düşümünü gözönüne alacak tasarımlar yapmak ve arızaya dayanıklı denetim şemasını tasarlamak açısından önemlidir. İşleve yakınsayan yapıyla 3. şema bu istekleri tamamiyle karşılayabilmektedir.

Her üç şema karşılaştırıldığında, arıza yalıtım gecikmesi ve süreklilik

denemesinin getirdiği eksiler yanında düşük hesapsal yükün ve işleve yakınsamanın getirdiği büyük artılarla 3. şemanın tasarlanan FDI şemaları içinde bir FDI şeması olarak gerçek zamanlı kullanılmaya en uygun olduğu görülmektedir.

Önerilen FTC şemasının FTC özellikleri açısından değerlendirilmesi ise Çizelge 7.2’de verilmiştir.

**Çizelge 7.2** Önerilen FTC şemasının özellikleri

<b>FTC öncesi-sonrası hata</b>	<b>Arıza işlevine yakınsamaya etki-Gürbüzlük</b>	<b>Yapılandırma gecikmesi</b>	<b>Doyma-Nazik Başarım Düşümü</b>	<b>Hesapsal Yük</b>
azalmakta	yok	yok	denetlenmiyor	düşük

Önerilen FTC şeması amaçlandığı gibi devreye girdikten sonra manipülatörlerde özellikle önemli olan hata tepe değerini neredeyse yarı yarıya azaltmaktadır. Bu başarımla ise kullanılan kazanç çizgeleme yönteminin başarımına büyük oranda bağlıdır.

Önerilen FTC şeması arıza işlevine yakınsamada bozucu bir etki göstermemekle beraber çok düşük oranda arızanın yalıtılmadığı durumlarda yalıtımı sağlamakta veya yalıtımın olduğu yerlerde yalıtımı bozmaktadır. Yalıtımın bozulduğu istenmeyen duruma özellikle arıza yüzdelerinin büyük olduğu durumlarda FTC oranının düzgün elde edilememesiyle dizgenin kararsız duruma geçişlerinde görülmektedir. Şemanın bu özelliğine dikkat edilerek gerçek zamanlı bir manipatöre uygulanmalıdır.

FTC şemalarının kullanıldığı yerlerde arıza yalıtımı ile yapılandırma hesabı ve uygulanması arasında bir gecikme bulunabilir. Önerilen şemada kazanç çizgeleme yönteminin tek bir kurala bağlı olması ise hesapsal yükün düşük olmasını sağlamakta, yapılandırma hesabı ve uygulaması arıza yalıtım anında tek bir kurala bağlı olarak kazanç çizgeleme ile hızlı bir şekilde yapılmakta, bu sayede şemada yapılandırma gecikmesi görülmemektedir.

Önerilen FTC şemasında güncelleme katsayısı Eş. (5.9) ile tanımlı bir kural tarafından bulunmaktadır. Bu tanımlı kural tamamen kullanıcı tarafından tecrübeyle tanımlanmış bir kuraldır. Bu kuralın iyileştirilmesiyle veya kullanıcı tecrübesini doğrudan kullanan bulanık kazanç çizgeleme yöntemiyle daha düşük hata değerleri elde edilebilir.

Ayrıca bulunan güncel katsayıya ulaşmak için zamana bağlı tanımlı bir değişim

farklı yörüngeler ve arızalar için farklı sonuçlar vermektedir. Bu durumda yapılacak seçim tamamen kullanıcı tarafından aşım - yerleşme zamanı - sürekli durum hatası arasındaki ödünleşmeye göre yapılmalıdır.

Gerçek zamanlı çalışmada oluşan bir hareketlendirici arızası hareketlendiriciyi doymaya götürebilir. FTC şemalarının bu gibi durumlarda nasıl bir yaklaşım içine girecekleri tamamen kullanıcı tarafından belirlenir. Gerçek manipülatörlerde kullanılan hareketlendiriciler arızasız durumda ürettikleri momentlerden çok daha fazlasını taşıyacak şekilde tasarlandıklarından ve bu durum tamamen gerçek bir manipülatörde kullanılan hareketlendirici tipine bağlı olduğundan benzetim çalışmalarında bu doyma-nazik başarımların düşümü özelliği gözardı edilmiştir ve denetlenmemektedir.

Tasarlanan tüm şemalar bir DSP üzerine gömülerek gerçekleştirme işlemine bir adım yaklaşmış, DSP ile yapılan gömme çalışmaları hem şemaların gerçek zamanlı çalışma başarımlarını denemek, hem de gömme işleminde oluşabilecek sorunları görmek açısından iyi bir deneyim sağlamıştır.

Tez çalışmasını izleyen çalışmalarda önerilen FDI ve FTC şemalarını öncelikle gerçek bir manipülatör üzerinde denenmesi hedeflenmektedir. Şu ana kadar her ne kadar tez çalışması bilimsel yayınlarla ileri bir derecede desteklenmiş olsa da, gerçekleştirme, bilimsel açıdan çalışmayı bir adım öteye taşıyacak ve literatürde kabul görmesi açısından kolaylık sağlayacaktır. Ayrıca gerçek dünyada oluşan diğer sorunlarla başetmede diğer çalışmalara örnek olması çalışmanın izleyen hedeflerinden birisidir.

Önerilen FDI ve FTC şemalarının üzerinde şu ana kadar fazla çalışma yapılmamış diğer özel doğrusal olmayan dizgelere uygulanması ilerideki hedeflerden birisidir. Bu sayede çalışmanın özgünlüğü bir kez daha ortaya konacaktır.

İzleyen çalışmalarda robot manipülatörleri için literatürde yeterince çalışmanın olduğu FDI şema tasarımı yerine hem robot manipatörler hem de doğrusal olmayan dizgeler için daha az çalışmanın olduğu FTC şema tasarımları üzerinde durulacaktır.

**KAYNAKLAR**

- Abdul, S., Liu, G., 2008. Decentralised fault tolerance and fault detection of modular and reconfigurable robots with joint torque sensing. IEEE International Conference on Robotics and Automation, s: 3520-3526
- Adjallah, K., Maquin, D., Ragot, J., 1994. Non-linear observer-based fault detection. Third IEEE Conf. on Control Applications, s: 1115-1120
- AirDisaster.Com, 2000. *United Airlines Flight 585, Special Report*
- Altuğ, S., Chow, M., Trussell, H. J., 1999. Fuzzy inference systems implemented on neural architectures for motor fault detection and diagnosis. IEEE Transactions On Industrial Electronics, Cilt: 46, No: 6, s: 1069-1079
- Ayoubi, M., 1994. Nonlinear dynamic systems identification with dynamic neural networks for fault diagnosis in technical processes. IEEE International Conference On Systems, Man and Cybernetics, Cilt: 3, s: 2120-2125
- Bajpai, G., 2001. Reconfigurable Control Of Aircraft Undergoing Sensor And Actuator Failures, Doktora Tezi, Drexel Universitesi, 112 s.
- Bajpai, G., Chang, B.C., Kwatny, H. G., 2002. Design of fault-tolerant systems for actuator failures in nonlinear systems. Proceedings of the American Control Conference, s: 3618-3623
- Ballé, P., 1998. Fuzzy model-based symptom generation and fault diagnosis for nonlinear processes. IEEE International Conference On Fuzzy Systems, Cilt: 2, s: 945-950
- Bonivento, C., Gentili, L., Paoli, A., 2004. Internal model based fault tolerant control of a robot manipulator. IEEE International Conference on Decision and Control, Cilt: 5, s: 5260 - 5265
- Boskovic, J. D., Mehra, R. K., 2002. Fault accommodation using model predictive methods. Proceedings of the American Control Conference, s: 5104-5109
- Brambilla, D., Capisani, L. M., Ferrara, A., Pisu, P., 2008. Fault detection for robot manipulators via second-order sliding modes. IEEE Transactions On Industrial Electronics, Cilt. 55, No. 11, s: 3954-3963
- Caccavale, F., Cilibrizzi, P., Pierri, F., Villani, L., 2009. Actuators fault diagnosis for robot manipulators with uncertain model. Control Engineering Practice, Cilt: 17, s: 146-157
- Caccavale, F., Walker, I. D., 1997. Observer-based fault detection for robot manipulators. IEEE International Conference on Robotics and Automation, s:2881-2887

- Cavallaro, J.R., Walker, I.D., 1994. A survey of NASA and military standards on fault tolerance and reliability applied to robotics, AIAA/NASA Conf. on Intelligent Robots in Field, Factory, Service, Space, s: 282-286
- Chassaing, R., Reay, D., 2008. *Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK-Second Edition*. Wiley&Sons Inc. Publishing, 518 s, ABD
- Chen, J., Patton, R.J., 1999. *Robust Model-Based Fault Diagnosis For Dynamic Systems*, Kluwer Academic Publishers, 356 s, Londra-İngiltere
- Chow, E.Y., Willsky, A.S., 1984. Analytical redundancy and the design of robust failure detection systems. IEEE Transactions On Automatic Control. Cilt: 29, No: 7, s: 603-614
- Datta, A., Mavroidis, C., Krishnasamy J., Hosek, M., 2007. Neural network based fault diagnostics of industrial robots using wavelet multi-resolution analysis. American Control Conference, s:1858-1863
- De Luca, A. Mattone, R., 2004. An adapt-and-detect actuator FDI for robot manipulators. IEEE International Conference on Robotics and Automation, s: 879-884
- Dexter, A.L., 1995. Fuzzy model-based fault diagnosis. IEE Proceedings On Cont. Theory and Applications. Cilt: 142, No: 6, s: 545-550
- Diallo, D., Benbouzid, M. H., Makouf, A. , 2004. A fault-tolerant control architecture for induction motor drives in automotive applications. IEEE Transactions On Vehicular Technology, Cilt: 53, No: 6, s: 1847-1855
- Diao, Y. , Passino, K. M., 2002. Intelligent fault-tolerant control using adaptive and learning methods. Control Engineering Practice, Cilt: 10, s: 801–817
- Diao, Y., Passino, K. M. , 2001. Intelligent fault tolerant control using adaptive schemes and multiple models. Proceedings of the American Control Conference, s: 2854-2859
- Dixon, W. E., Walker, I. D., Dawson, D. M., Hartranft, J. P., 2000. Fault detection for robot manipulators with parametric uncertainty: a prediction-error-based approach. IEEE Transactions On Robotics And Automation, Cilt:16, No: 6, s: 689-699,
- Fantuzzi C., Secchi C., Visioli A., 2003. On the fault detection and isolation of industrial robot manipulators. Proc. of 7th International IFAC Symposium on Robot Control
- Frank, P.M., 1990. Fault diagnosis in dynamic systems using analytical and knowledge based redundancy- a survey and some new results. Automatica, Cilt: 26, No: 3, s: 459-474

- Frank, P.M., Ding, X, 1997. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, Cilt: 7, No: 6, s: 403-424
- Ganguli, S., Marcos, A., Balas, G., 2002. Reconfigurable LPV control design for Boeing 747-100/200 longitudinal axis. *Proceedings of the American Control Conference*, s: 3612-3617
- Garcia, E.A., Frank, P. M., 1996. Analysis of a class of dedicated observer schemes to sensor fault isolation. *UKACC International Conference on CONTROL '96*, s: 60-65
- Garcia, E.A., Frank, P.M., 1997. Deterministic nonlinear observer-based approaches to fault diagnosis: a survey. *Control Eng. Practice*, Cilt: 5, No: 5, s: 663-670
- Gertler, J.J., 1998. *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker Publishing, 478 s, Virginia-ABD
- Goel, P., Dedeoglu, G., Roumeliotis, S. I., Sukhatme, G. S., 2000. Fault detection and identification in a mobile robot using multiple model estimation and neural network. *IEEE International Conference on Robotics and Automation*, s: 2302-2309
- Halder, B., Sarkar, N., 2007. Robust fault detection of a robotic manipulator. *The International Journal of Robotics Research*, Cilt: 26, No: 3, s: 273-285
- Haykin, S., 1999. *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, 2. baskı 823 s, Kanada
- Ho, L. W., Yen, G. G., 2001. Reconfigurable control system design for fault diagnosis and accomodation. *Proceedings of the 40th IEEE Conference On Decision And Control*, s: 1873-1878
- Wikipedia, Double precision floating-point format [http://en.wikipedia.org/wiki/Double\\_precision\\_floating-point\\_format](http://en.wikipedia.org/wiki/Double_precision_floating-point_format), 2009
- Isermann, R., 1998. On fuzzy logic applications for automatic control, supervision, and fault diagnosis. *IEEE Transactions On Systems, Man, and Cybernetics-Part A:Systems and Humans*, Cilt: 28, No: 2, s: 221-235
- Isermann, R., Ballé, P., 1997. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Eng. Practice* Cilt: 5, No: 5, s: 709-719
- Izumikawa, Y., Yubai, K., Hori, T., 2002. Vibration suppression control of flexible arm robot by pd gain switching considering sensor failure. *IEEE International Conference on Industrial Technology*, Cilt: 2, s: 684-689

- Izumikawa, Y., Yubai, K., Hirai, J., 2004. Fault-tolerant control system of flexible arm for sensor fault by using reaction force observer. IEEE International Conference On Advanced Motion Control, s: 583-588
- Jang, J.R., Sun, C.T., Mizutani E., 1997. *Neuro-Fuzzy and Soft Computing*, Prentice-Hall Inc., 614 s, ABD
- Kabore, P., Wang, H., 2001. Design of fault diagnosis filters and fault-tolerant control for a class of nonlinear systems. IEEE Transactions On Automatic Control, Cilt: 46, No:11, s: 1805-1810
- Kanev, S.K., 2004. *Robust Fault-Tolerant Control*, FEBO-DRUK Publisher, 283 s, Hollanda
- Karpenko, M., Sepheri, M., Scuse, D., 2001. Neural network detection and identification of actuator faults in a pneumatic process control valve. IEEE International Symposium on Computational Intelligence In Robotics and Automation, s: 166-171
- Konstantopoulos, I. K., Antsaklis, P. J., 1996. Eigenstructure Assignment In Reconfigurable Control Systems. Technical Report of the ISIS Group at the University of Notre Dame
- Kuntze, H.-B., Frey, Ch., Giesen, K., Milighetti, G., 2003. Fault tolerant supervisory control of human interactive robots. Proceedings IAR Annual Meeting, ACD Workshop, s. 55-60
- Kwong, W. A., Passino, K. M., Laukonen, E. G., Yurbovich, S., 1995. Expert supervision of fuzzy learning systems for fault tolerant aircraft control. Proceedings of IEEE, Cilt: 83, No: 3, s: 466-483
- Lee, I. S., Kim, J. T., Lee, J. W., Lee, D. Y., Kim, K. Y., 2003. Model-based fault detection and isolation method using ART2 neural network. International Journal Of Intelligent Systems, Cilt:18, s:1087-1100
- Lehtoranta, J., Koivo, H. N., 2005. Fault diagnosis of induction motors with dynamical neural Networks. IEEE International Conference On Systems, Man and Cybernetics, Cilt: 3, s: 2979-2984
- Leuschen, M. L., Walker, I. D., Cavallaro, J. R., 2005. Fault residual generation via nonlinear analytical redundancy. IEEE Transactions On Control Systems Technology, Cilt: 13, No: 3, s: 452-458
- Leuschen, M.L., 2001. Derivation and Application of Nonlinear Analytical Redundancy Techniques with Applications to Robotics. Doktora tezi, Rice Üniversitesi, Texas, 194 s.
- Lewis, F. L., Abdallah, C. T., Dawson, D. M., 1993. *Control of Robot Manipulators*, MacMillan Publishing, 424 s, ABD
- Lim, S., Shin, J. Y., 2004. Fault Tolerant Controller Design Using Hybrid Linear Parameter-Varying Control. NASA, NIA Report No:03

- Liu, X.Q, Zhang, H.Y., Liu, J., Yang, J., 2000. Fault detection and diagnosis of permanent-magnet dc motor based on parameter estimation and neural network. *IEEE Transactions On Industrial Electronics*, Cilt: 47, No: 5, s: 1021-1030
- Lopez-Toribio, C.J., Patton, R.J., Daley, S., 2000. Takagi–Sugeno fuzzy fault-tolerant control of an induction motor. *Neural Comput. and Applic.*, Cilt: 9, s:19–28
- Maciejowski, J. M., Jones, C. N., 2003. MPC fault--tolerant flight control case study: flight 1862. *IFAC Safeprocess Conference*, s: 265–276
- Magni, J.F, Mouyon, P., 1994. On residual generation by observer and parity space approaches. *IEEE Transactions On Automatic Control*. Cilt: 39, No: 2, s: 441-447
- Marcu, T., Mirea, L., Frank, P. M, 1998. Neural observer schemes for robust detection and isolation of process faults. *UKACC International Conference on CONTROL '98*, s: 958-963
- Maybeck, P. S., Stevens, R. D., 1990. Reconfigurable flight control via multiple model adaptive control methods. *Proceedings of the 29th Conference on Decision and Control*, s: 3351- 3356
- Moseler, O. , Isermann, R., 2000. Application of model based fault detection to a brushless dc motor. *IEEE Transactions on Industrial Electronics*, Cilt: 47, No: 5, s: 1015-1020
- National Transportation Safety Board, 1999. *Abstract of Final Report, USAir flight 427*
- Naughton, J. M., Chen, Y. C., Jiang, J., 1996. A neural network application to fault diagnosis for robotic manipulator. *IEEE International Conference on Control Applications*, s: 988-993
- Notash, L., Moore, T. N. 2002, Fault Analysis in Mechatronic Systems. *The Mechatronics Handbook*, (Editör: R. H. Bishop), Bölüm 39, CRC Press, ABD
- Paredis, C. J. J., 1996. An Agent-Based Approach to the Design of Rapidly Deployable Fault Tolerant Manipulators, *Doktora Tezi*, Carnegie-Mellon University, Pennsylvania, 215 s.
- Paredis, C. J. J., Khosla, P. K., 1995. Design of modular fault tolerant manipulators. *The Algorithmic Foundations of Robotics*, s: 373-383
- Paredis, C. J. J., Khosla, P. K., 1996a. Designing fault tolerant manipulators: how many degrees-of-freedom. *International Journal of Robotics Research*, Cilt: 15, No: 6, s: 611-628
- Paredis, C. J. J., Khosla, P. K., 1996b. Fault tolerant task execution through global trajectory planning. *Reliability Engineering and System Safety (special issue on Safety of Robotic Systems)*, Cilt: 53, No: 3, s: 225-235

- Pashilkar, A.A., Sundararajan, N., Saratchandran, P., 2006. A fault-tolerant neural aided controller for aircraft auto-landing. *Aerospace Science and Technology*, Cilt:10, s: 49-61
- Patton, R. J., 1993. Robustness issues in fault-tolerant control. *IEE Colloquium on Fault Diagnosis and Control System Reconfiguration*, Cilt: 1, s: 1-25
- Patton, R. J., Uppal, F. J., Lopez-Toribio, C. J., 2000. Soft computing approaches to fault diagnosis for dynamic systems: a survey. *IFAC Symposium SAFEPROCESS*, s: 298-311
- Patton, R.J., Chen, J., 1997. Observer-based fault detection and isolation: robustness and applications. *Control Eng. Practice*, Cilt: 5, No: 5, s: 671-682
- Patton, R.J., Chen, J., Lopez-Toribio, C.J., 1998. Fuzzy observers for nonlinear dynamic systems fault diagnosis. *37th IEEE Conference on Decision & Control*, s: 84-89
- Patton, R.J., Frank, P.M., Clark R.N., 2000. *Issues Of Fault Diagnosis For Dynamic Systems*. Springer-Verlag, 598 s, Londra-İngiltere
- Pei, X., Chowdhury, F. N., 1999. Unsupervised neural network for fault detection and classification in dynamic systems. *IEEE International Conference on Control Applications*, s: 640-645
- Polycarpou, M. M., Helmicki A. J., 1995. Automated fault detection and accommodation: a learning systems approach. *IEEE Transactions On Systems, Man and Cybernetics*, Cilt: 25, No: 11, s: 1447-1458
- Polycarpou, M. M., Trunov, A.B., 2002. Learning approach to nonlinear fault diagnosis: detectability analysis. *IEEE Transactions On Automatic Control*, Cilt: 45, No: 4, s: 806-812
- Polycarpou, M., Zhang, X., Xu, R., Yang, Y., Kwan, C., 2004. A neural network based approach to adaptive fault tolerant flight control. *IEEE International Symposium on Intelligent Control*, s: 61-66
- Qu, Z., Ihlefeld, C. M., Yuafang, J., 2001. Robust control of a class of nonlinear uncertain systems. *Proceedings of the 40th IEEE Conference On Decision and Control*, s: 1472-1478
- Qu, Z., Ihlefeld, C.M., Jin, Y., Saengdeejing, A., 2003. Robust fault-tolerant self-recovering control of nonlinear uncertain systems. *Automatica*, Cilt: 39, s: 1763-1771
- Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *IEEE International Conference on Neural Networks*, Cilt: 1, s: 586-591

- Sauter, D., Hamelin, F., Noura, H., 1998. Fault diagnosis and accommodation in dynamic systems ; application to a dc motor. Proceedings of the American Control Conference, s: 2872-2873
- Schneider, H., Frank, P. M., 1996. Observed based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation. IEEE Transactions On System Technology, Cilt: 4, No: 3, s: 274-282
- Seliger, R., Frank, P.M., 1991. Fault diagnosis by disturbance decoupled nonlinear observers. 30th Conf. on Decision and Control, s: 2248-2253
- Shin, J. H., Lee, C.Y., Lee, J.J., 1999. Robust fault-tolerant control framework for robot manipulators with free-swinging joint failures: fault detection, identification and accommodation. IEEE International Conference on Intelligent Robots and Systems, s: 185-190
- Simani, S., Fantuzzi, C., Patton, R.J., 2002. *Model-Based Fault Diagnosis In Dynamic Systems Using Identification Techniques*. Springer-Verlag, 291 s, Londra-İngiltere
- Siqueira, A. A. G., Terra, M. H., 2004. A fault tolerant manipulator robot based on  $H_2$ ,  $H_\infty$  and mixed  $H_2/H_\infty$  markovian controls. International Conference on Control Applications, s: 309-314
- Sreevijayan, D., Tosunoglu, S., Tesar, D., 1994. Architectures for fault-tolerant mechanical systems. Proceedings of 7th Mediterrian Electrotechnical Conference, Cilt: 3, s: 1029-1033
- Stoustrup, J., Niemann, H., 2004. Fault tolerant control for unstable systems : a linear time varying approach. Proceeding of the 2004 American Control Conference, s: 1794-1798
- Tang, T., Yao, G., 2005. A fault-tolerant control method based on adaptive fnn for ship control system. IEEE International Conference on Control and Automation, s: 930-935
- Terra, M. H., Tinos, R., 2001. Fault detection and isolation in robotic manipulators via neural networks: a comparison among three architectures for residual analysis. Journal of Robotic Systems, Cilt: 18, No: 7, s: 357-374
- Thomas, S., Chang, B. C., Kwatny, H.G. , 2003. Controller reconfiguration for nonlinear systems using composite observer. Proceedings of the American Control Conference, s: 4779-4784
- Tosunoglu, S., 1995. Fault-tolerant control of mechanical systems. IEEE International Conference On Industrial Electronics, Control and Instrumentation. Cilt: 1 s: 127-132
- Vemuri, A. T., Polycarpou, M. M., 1997. Neural-network-based robust fault diagnosis in robotic systems. IEEE Transactions on Neural Networks, Cilt: 8, No: 6, s: 1410-1420

- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin., K., 2003. A review of process fault detection and diagnosis part III: process history based methods. *Computers and Chemical Engineering*. Cilt: 27, s: 327-346
- Visinsky, M. L., 1991. *Fault Detection And Fault Tolerance Methods For Robotics*. Yüksek Lisans Tezi, Rice University, Texas, 103 s.
- Wang, H., Wang, Y., 1999. Neural-network-based fault-tolerant control of unknown nonlinear systems. *IEE Proceedings-Control Theory Appl.*, Cilt: 146, No: 5, s: 389-398
- Wu, S.F., Grimbale, M. J., Wei, W., 1999. QFT based robust fault tolerant flight control design for a remote pilotless vehicle. *IEEE International Conference on Control Applications*, s: 57-62
- Yen, G. G., Ho, L.W., 2000. Fault tolerant control: an intelligent sliding mode control strategy. *Proceedings of the American Control Conference*, s: 4204-4208
- Yen, G. G., Lima, P. G. D., 2005. Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor. *IEEE Transactions On Automation Science and Engineering*, Cilt: 2, No: 2, s: 121-131
- Yen, G. G., Lima, P. G. D., 2002. Dynamic database approach for fault tolerant control using dual heuristic programming. *Proceedings of the American Control Conference*, s: 5080-5085
- Yu, D.L., Chang, T. K., 2005. Fault tolerant control of multivariable processes using auto-tuning PID controller. *IEEE Transactions On Systems, Man, and Cybernetics-Part B : Cybernetics*, Cilt: 35, No: 1, s: 32-43
- Zhang, X., 2002. *Fault Diagnosis and Fault Tolerant Control In Nonlinear Systems*, Doktora Tezi, University of Cincinnati, Cincinnati, 166 s.
- Zhang, X., Polycarpou, M.M., Parisini, T., 2001. Integrated design of fault diagnosis and accomodation schemes for a class of nonlinear systems. *IEEE Conference On Decision And Control*, s: 1448-1453
- Zhang, X., Farfan-Ramos, L., Rattan, K., 2009. Adaptive fault-tolerant fuzzy control of robotic systems. *The 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009)*, s: 1-9
- Zhang, X., Parisini, T., Polycarpou, M.M., 2004. Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach. *IEEE Transactions On Automatic Control*, Cilt: 49, No: 8, s: 1259-1274

- Zhang, X., Polycarpou, M. M., Parisini, T. , 2002. A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. IEEE Transactions On Automatic Control, Cilt: 47, No: 4, s: 576-592
- Zhang, Y., Jiang , J., 2003. Fault tolerant control system design with explicit consideration of performance degradation. IEEE Transactions on Aerospace and Electronic Systems, Cilt: 39 , No: 3, s: 838-848
- Zhang, Y., Jiang, J., 2008. Bibliographical review on reconfigurable fault-tolerant control systems. Annual Reviews In Control, Cilt: 32, s: 229-252
- Zhao, Z.-Y., Tomizuka, M., Isaka, S., 1993. Fuzzy gain scheduling of PID controllers. IEEE Transactions On Systems, Man, And Cybernetics. Cilt: 23, No: 5, s: 1392-1398
- Zhou, D. H., Frank, P. M., 1998. Fault diagnostics and fault toierant control. IEEE Transactions On Aerospace and Electronic Systems, Cilt: 34, No: 2 , s: 420-427
- Zidani, F., Benbouzid, M.E.H., Diallo, D., Benchat, A., 2003. Active fault-tolerant control of induction motor drives in ev and hev against sensor failures using a fuzzy decision system. IEEE International Conference on Electrical Machines and Drives, Cilt: 2, s: 677-683

### Ek A. Hesaplanmış Moment-PID (HM-PID) Denetimi

HM-PID denetimi modern denetim dizgelerinde ve robot manipülatör denetiminde yaygın şekilde kullanılan bir yöntemdir (Lewis ve ark., 1993). Robot dinamiklerinin Eş. (2.2) ile verilen genelleştirilmiş hali yazılırsa

$$\tau = M(q(t))\ddot{q}(t) + N(q(t), \dot{q}(t))$$

şeklinde tanımlanır. Hata işaretleri ise şu şekilde tanımlanır.

$$e(t) = q_d(t) - q(t)$$

$$\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t)$$

$$\ddot{e}(t) = \ddot{q}_d(t) - \ddot{q}(t)$$

$$\dot{\varepsilon}(t) = e(t)$$

Birinci eşitlikten  $\ddot{q}(t)$  terimi çekilir ve  $\ddot{e}(t)$  için yerine koyulursa

$$\ddot{e}(t) = \ddot{q}_d(t) + M^{-1}(q(t))(N(q(t), \dot{q}(t)) - \tau)$$

haline gelir. Denetim işareti için ise hatayı sifıra götürecektir şekilde bir denetleyici tasarlanır:

$$\tau = M(q(t))(\ddot{q}_d(t) - u) + N(q(t), \dot{q}(t))$$

Denetim işareti tasarımı için çeşitli yöntemler mevcuttur. Hata işaretini kullanan PID seçilirse denetim işareti şu hale gelir

$$u = -K_v \dot{e}(t) - K_p e(t) - K_i \varepsilon(t)$$

Sonuç olarak tasarlanan moment işareti ise şu hale gelir :

$$\tau = M(q(t))(\ddot{q}_d(t) + K_v \dot{e}(t) + K_p e(t) + K_i \varepsilon(t)) + N(q(t), \dot{q}(t))$$

### Ek B. Modelleyici M-ANFIS Eğitimi İçin Kullanılan Yörüngeler

Sırasıyla  $q_{d1}(t)(rad.)$

$q_{d2}(t)(rad.)$

$1*\sin(t/2)$ $1*\cos(t/2)$	$-1*\sin(t/2)$ $1*\cos(t/2)$	$1*\sin(t/2)$ $-1*\cos(t/2)$	$-1*\sin(t/2)$ $-1*\cos(t/2)$
$0.8*\sin(t/2)$ $0.8*\cos(t/2)$	$-0.8*\sin(t/2)$ $0.8*\cos(t/2)$	$0.8*\sin(t/2)$ $-0.8*\cos(t/2)$	$-0.8*\sin(t/2)$ $-0.8*\cos(t/2)$
$0.6*\sin(t/2)$ $0.6*\cos(t/2)$	$-0.6*\sin(t/2)$ $0.6*\cos(t/2)$	$0.6*\sin(t/2)$ $-0.6*\cos(t/2)$	$-0.6*\sin(t/2)$ $-0.6*\cos(t/2)$
$0.4*\sin(t/2)$ $0.4*\cos(t/2)$	$-0.4*\sin(t/2)$ $0.4*\cos(t/2)$	$0.4*\sin(t/2)$ $-0.4*\cos(t/2)$	$-0.4*\sin(t/2)$ $-0.4*\cos(t/2)$
$0.2*\sin(t/2)$ $0.2*\cos(t/2)$	$-0.2*\sin(t/2)$ $0.2*\cos(t/2)$	$0.2*\sin(t/2)$ $-0.2*\cos(t/2)$	$-0.2*\sin(t/2)$ $-0.2*\cos(t/2)$

$1*\sin(t/2)$ $0.4*\cos(t/2)$	$-1*\sin(t/2)$ $0.4*\cos(t/2)$	$1*\sin(t/2)$ $-0.4*\cos(t/2)$	$-1*\sin(t/2)$ $-0.4*\cos(t/2)$
$0.8*\sin(t/2)$ $0.4*\cos(t/2)$	$-0.8*\sin(t/2)$ $0.4*\cos(t/2)$	$0.8*\sin(t/2)$ $-0.4*\cos(t/2)$	$-0.8*\sin(t/2)$ $-0.4*\cos(t/2)$
$0.6*\sin(t/2)$ $0.4*\cos(t/2)$	$-0.6*\sin(t/2)$ $0.4*\cos(t/2)$	$0.6*\sin(t/2)$ $-0.4*\cos(t/2)$	$-0.6*\sin(t/2)$ $-0.4*\cos(t/2)$
$0.2*\sin(t/2)$ $0.4*\cos(t/2)$	$-0.2*\sin(t/2)$ $0.4*\cos(t/2)$	$0.2*\sin(t/2)$ $-0.4*\cos(t/2)$	$-0.2*\sin(t/2)$ $-0.4*\cos(t/2)$
$0*\sin(t/2)$ $0.4*\cos(t/2)$	$-0.1*\sin(t/2)$ $0.4*\cos(t/2)$	$0.0*\sin(t/2)$ $-0.4*\cos(t/2)$	$-0.1*\sin(t/2)$ $-0.4*\cos(t/2)$

$1*\sin(t/2)$ $0.6*\cos(t/2)$	$-1*\sin(t/2)$ $0.6*\cos(t/2)$	$1*\sin(t/2)$ $-0.6*\cos(t/2)$	$-1*\sin(t/2)$ $-0.6*\cos(t/2)$
$0.8*\sin(t/2)$ $0.6*\cos(t/2)$	$-0.8*\sin(t/2)$ $0.6*\cos(t/2)$	$0.8*\sin(t/2)$ $-0.6*\cos(t/2)$	$-0.8*\sin(t/2)$ $-0.6*\cos(t/2)$
$0.2*\sin(t/2)$ $0.6*\cos(t/2)$	$-0.2*\sin(t/2)$ $0.6*\cos(t/2)$	$0.2*\sin(t/2)$ $-0.6*\cos(t/2)$	$-0.2*\sin(t/2)$ $-0.6*\cos(t/2)$
$-0.1*\sin(t/2)$ $0.6*\cos(t/2)$	$0*\sin(t/2)$ $0.6*\cos(t/2)$	$0.1*\sin(t/2)$ $-0.6*\cos(t/2)$	$0*\sin(t/2)$ $-0.6*\cos(t/2)$

$1*\cos(t/2)$ $1*\sin(t/2)$	$-1*\cos(t/2)$ $1*\sin(t/2)$	$1*\cos(t/2)$ $-1*\sin(t/2)$	$-1*\cos(t/2)$ $-1*\sin(t/2)$
$0.8*\cos(t/2)$ $0.8*\sin(t/2)$	$-0.8*\cos(t/2)$ $0.8*\sin(t/2)$	$0.8*\cos(t/2)$ $-0.8*\sin(t/2)$	$-0.8*\cos(t/2)$ $-0.8*\sin(t/2)$
$0.6*\cos(t/2)$ $0.6*\sin(t/2)$	$-0.6*\cos(t/2)$ $0.6*\sin(t/2)$	$0.6*\cos(t/2)$ $-0.6*\sin(t/2)$	$-0.6*\cos(t/2)$ $-0.6*\sin(t/2)$
$0.4*\cos(t/2)$ $0.4*\sin(t/2)$	$-0.4*\cos(t/2)$ $0.4*\sin(t/2)$	$0.4*\cos(t/2)$ $-0.4*\sin(t/2)$	$-0.4*\cos(t/2)$ $-0.4*\sin(t/2)$
$0.2*\cos(t/2)$ $0.2*\sin(t/2)$	$-0.2*\cos(t/2)$ $0.2*\sin(t/2)$	$0.2*\cos(t/2)$ $-0.2*\sin(t/2)$	$-0.2*\cos(t/2)$ $-0.2*\sin(t/2)$

$1*\cos(t/2)$ $0.4*\sin(t/2)$	$-1*\cos(t/2)$ $0.4*\sin(t/2)$	$1*\cos(t/2)$ $-0.4*\sin(t/2)$	$-1*\cos(t/2)$ $-0.4*\sin(t/2)$
$0.8*\cos(t/2)$ $0.4*\sin(t/2)$	$-0.8*\cos(t/2)$ $0.4*\sin(t/2)$	$0.8*\cos(t/2)$ $-0.4*\sin(t/2)$	$-0.8*\cos(t/2)$ $-0.4*\sin(t/2)$
$0.6*\cos(t/2)$ $0.4*\sin(t/2)$	$-0.6*\cos(t/2)$ $0.4*\sin(t/2)$	$0.6*\cos(t/2)$ $-0.4*\sin(t/2)$	$-0.6*\cos(t/2)$ $-0.4*\sin(t/2)$
$0.2*\cos(t/2)$ $0.4*\sin(t/2)$	$-0.2*\cos(t/2)$ $0.4*\sin(t/2)$	$0.2*\cos(t/2)$ $-0.4*\sin(t/2)$	$-0.2*\cos(t/2)$ $-0.4*\sin(t/2)$
$0*\cos(t/2)$ $0.4*\sin(t/2)$	$-0.1*\cos(t/2)$ $0.4*\sin(t/2)$	$0.0*\cos(t/2)$ $-0.4*\sin(t/2)$	$-0.1*\cos(t/2)$ $-0.4*\sin(t/2)$

$1*\cos(t/2)$ $0.6*\sin(t/2)$	$-1*\cos(t/2)$ $0.6*\sin(t/2)$	$1*\cos(t/2)$ $-0.6*\sin(t/2)$	$-1*\cos(t/2)$ $-0.6*\sin(t/2)$
$0.8*\cos(t/2)$ $0.6*\sin(t/2)$	$-0.8*\cos(t/2)$ $0.6*\sin(t/2)$	$0.8*\cos(t/2)$ $-0.6*\sin(t/2)$	$-0.8*\cos(t/2)$ $-0.6*\sin(t/2)$
$0.2*\cos(t/2)$ $0.6*\sin(t/2)$	$-0.2*\cos(t/2)$ $0.6*\sin(t/2)$	$0.2*\cos(t/2)$ $-0.6*\sin(t/2)$	$-0.2*\cos(t/2)$ $-0.6*\sin(t/2)$
$-0.1*\cos(t/2)$ $0.6*\sin(t/2)$	$0*\cos(t/2)$ $0.6*\sin(t/2)$	$0.1*\cos(t/2)$ $-0.6*\sin(t/2)$	$0*\cos(t/2)$ $-0.6*\sin(t/2)$

$1*\cos(t/2)$ $0.2*\sin(t/2)$	$-1*\cos(t/2)$ $0.2*\sin(t/2)$	$1*\cos(t/2)$ $-0.2*\sin(t/2)$	$-1*\cos(t/2)$ $-0.2*\sin(t/2)$
$0.8*\cos(t/2)$ $0.2*\sin(t/2)$	$-0.8*\cos(t/2)$ $0.2*\sin(t/2)$	$0.8*\cos(t/2)$ $-0.2*\sin(t/2)$	$-0.8*\cos(t/2)$ $-0.2*\sin(t/2)$
$-0.1*\cos(t/2)$ $0.2*\sin(t/2)$	$0*\cos(t/2)$ $0.2*\sin(t/2)$	$0.1*\cos(t/2)$ $-0.2*\sin(t/2)$	$0*\cos(t/2)$ $-0.2*\sin(t/2)$

$1*\sin(t/2)$ $0.2*\cos(t/2)$	$-1*\sin(t/2)$ $0.2*\cos(t/2)$	$1*\sin(t/2)$ $-0.2*\cos(t/2)$	$-1*\sin(t/2)$ $-0.2*\cos(t/2)$
$0.8*\sin(t/2)$ $0.2*\cos(t/2)$	$-0.8*\sin(t/2)$ $0.2*\cos(t/2)$	$0.8*\sin(t/2)$ $-0.2*\cos(t/2)$	$-0.8*\sin(t/2)$ $-0.2*\cos(t/2)$
$-0.1*\sin(t/2)$ $0.2*\cos(t/2)$	$0*\sin(t/2)$ $0.2*\cos(t/2)$	$0.1*\sin(t/2)$ $-0.2*\cos(t/2)$	$0*\sin(t/2)$ $-0.2*\cos(t/2)$

$0.1*\cos(t/2)$ $-0.4*\sin(t/2)$	$-0.1*\cos(t/2)$ $-0.6*\sin(t/2)$		
$1*\sin(t/2)$ $0.8*\cos(t/2)$	$-1*\sin(t/2)$ $0.8*\cos(t/2)$	$1*\sin(t/2)$ $-0.8*\cos(t/2)$	$-1*\sin(t/2)$ $-0.8*\cos(t/2)$
$1*\cos(t/2)$ $0.8*\sin(t/2)$	$-1*\cos(t/2)$ $0.8*\sin(t/2)$	$1*\cos(t/2)$ $-0.8*\sin(t/2)$	$-1*\cos(t/2)$ $-0.8*\sin(t/2)$

## Ek C. Önerilen 1. FDI Şemasına Ait MATLAB Programı ve Akış Diyagramı

C:\MATLAB6p5\computed\_torque\residual\_generation\CT\_actuator\_fault\_NN\_SIU\_tez.m

//

```

clc
clear all
close all
tic
a1=1;
a2=1;
m1=1;
m2=1;
kp=100;
kv=20;
ki=500;
deltat=0.01;
g=9.81;
Kp=[100 0;0 100];
Kv=[20 0;0 20];
Ki=[500 0;0 500];
t=0.01:0.01:60.01;
aa=400;
tt=2400;

qd1=0.8*cos(t/2);
qd2=0.8*sin(t/2);
load network4.mat %MATLAB6p5\work

for i=2:6000
    qd1us(i)=(qd1(i)-qd1(i-1))/deltat;
    qd2us(i)=(qd2(i)-qd2(i-1))/deltat;

    qd1uss(i)=(qd1us(i)-qd1us(i-1))/deltat;
    qd2uss(i)=(qd2us(i)-qd2us(i-1))/deltat;
end

% Comp_torque_PID_ANFIS

q1(1)=0.1;
q2(1)=0;
q1us(1)=0;
q2us(1)=0;
q1(2)=0.1;
q2(2)=0;
q1us(2)=0;
q2us(2)=0;

e1(1)=qd1(1)-q1(1);
e2(1)=qd2(1)-q2(1);
e1(2)=e1(1);
e2(2)=e2(1);

e1us(2)=(e1(2)-e1(1))/deltat;
e2us(2)=(e2(2)-e2(1))/deltat;

```

```

eps1(1)=0;
eps2(1)=0;
eps1(2)=eps1(1)+deltat*e1(1);
eps2(2)=eps2(1)+deltat*e2(1);

q1us(1)=(q1(2)-q1(1))/deltat;
q2us(1)=(q2(2)-q2(1))/deltat;

robot1=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q1\robot_q1_deneme.fis');
robot2=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data\robot_q1us.fis');
robot3=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2\robot_q2.fis');
robot4=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2us\robot_q2us.fis');
for i=2:aa
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M22(i)=m2*a2^2;
    M=[M11(i) M12(i);M21(i) M22(i)];

    V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i))+q2us(i)^2)*sin(q2(i));
    V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
    V=[V1(i);V2(i)];

    G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
    G2(i)=m2*g*a2*cos(q1(i)+q2(i));
    G=[G1(i);G2(i)];

    tau1(i+1)=[M11(i)
M12(i)]*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)]+V1(i)+G1(i)
);
    tau2(i+1)=[M21(i)
M22(i)]*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)]+V2(i)+G2(i)
);
    tau=[tau1(i+1);tau2(i+1)];

    % belirsizlik ekleme
    uncertainty_function

    quss=inv(Ma)*(tau-Va-Ga);

    q1uss(i+1)=quss(1);
    q2uss(i+1)=quss(2);
    q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
    q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
    q1(i+1)=q1(i)+deltat*q1us(i+1);
    q2(i+1)=q2(i)+deltat*q2us(i+1);

    e1(i+1)=qd1(i+1)-q1(i+1);
    e2(i+1)=qd2(i+1)-q2(i+1);
    e1us(i+1)=qd1us(i)-q1us(i);
    e2us(i+1)=qd2us(i)-q2us(i);
    eps1(i+1)=eps1(i)+deltat*e1(i+1);
    eps2(i+1)=eps2(i)+deltat*e2(i+1);

    q1cap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot1);
    q1uscap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot2);
    q2cap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot3);
    q2uscap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot4);
    r1(i)=q1(i+1)-q1cap(i);

```

```

r2(i)=q1us(i+1)-q1uscap(i);
r3(i)=q2(i+1)-q2cap(i);
r4(i)=q2us(i+1)-q2uscap(i);
r2r2(i)=(r1(i)-r1(i-1))/deltat;
r4r4(i)=(r3(i)-r3(i-1))/deltat;

end
faulty=0;
faults=[];
thr=0;

% load network31.mat
for i=aa+1:5000
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M22(i)=m2*a2^2;
    M=[M11(i) M12(i);M21(i) M22(i)];

    V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i))+q2us(i)^2)*sin(q2(i));
    V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
    V=[V1(i);V2(i)];

    G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
    G2(i)=m2*g*a2*cos(q1(i)+q2(i));
    G=[G1(i);G2(i)];

    tau1(i+1)=[M11(i)
M12(i)]*( [qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)]+V1(i)+G1(i)
);
    tau2(i+1)=[M21(i)
M22(i)]*( [qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)]+V2(i)+G2(i)
);
    tau=[tau1(i+1);tau2(i+1)];

    % belirsizlik ekleme
    uncertainty_function
    if (i>tt) %ariza anı
    % actuator fault
        tau=[tau(1)*0.5;tau(2)*1];
    end
    quss=inv(Ma)*(tau-Va-Ga);

    q1uss(i+1)=quss(1);
    q2uss(i+1)=quss(2);
    q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
    q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
    q1(i+1)=q1(i)+deltat*q1us(i+1);
    q2(i+1)=q2(i)+deltat*q2us(i+1);

    e1(i+1)=qd1(i+1)-q1(i+1);
    e2(i+1)=qd2(i+1)-q2(i+1);
    e1us(i+1)=qd1us(i)-q1us(i);
    e2us(i+1)=qd2us(i)-q2us(i);
    eps1(i+1)=eps1(i)+deltat*e1(i+1);
    eps2(i+1)=eps2(i)+deltat*e2(i+1);

    q1cap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot1);
    q1uscap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot2);

```

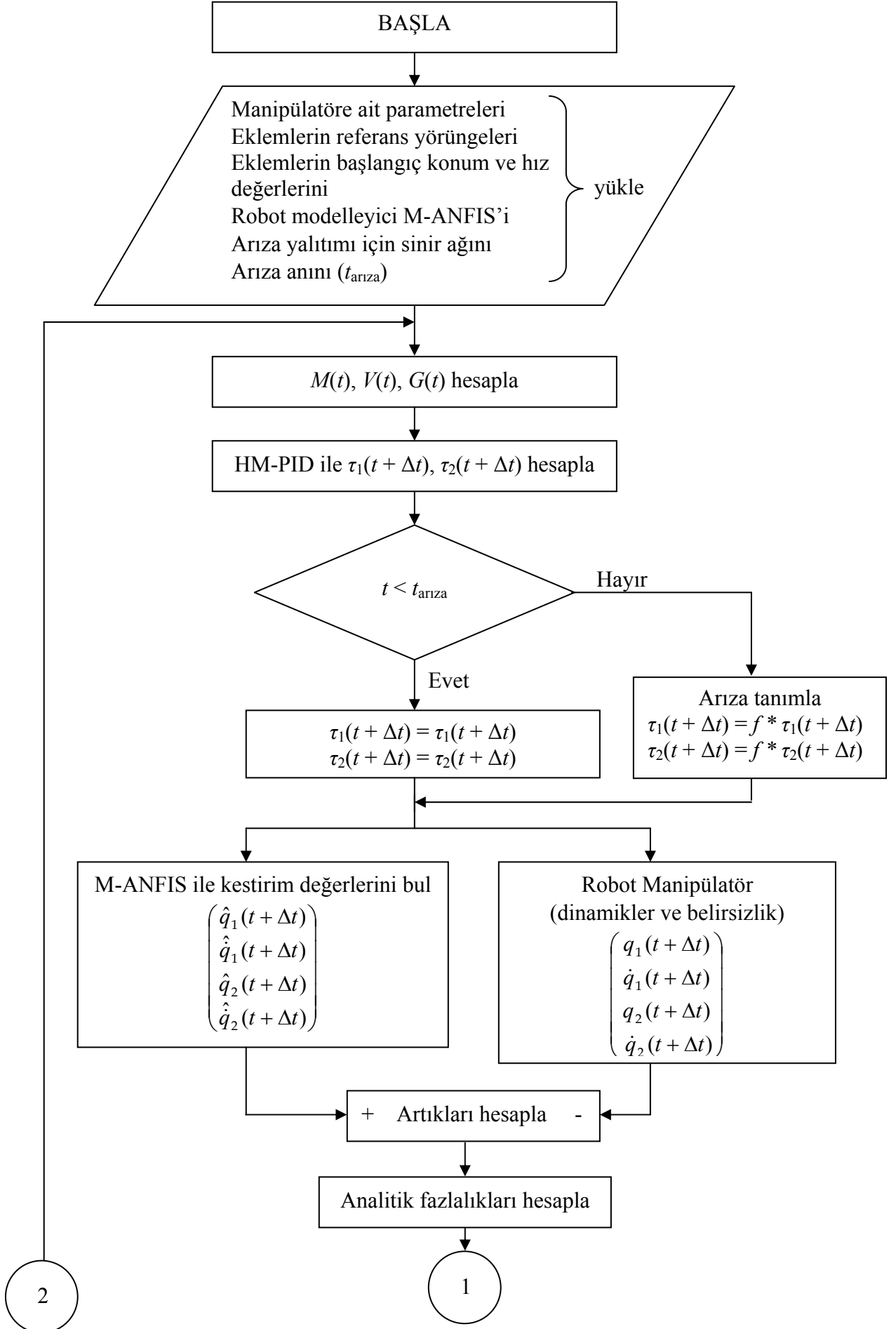
```

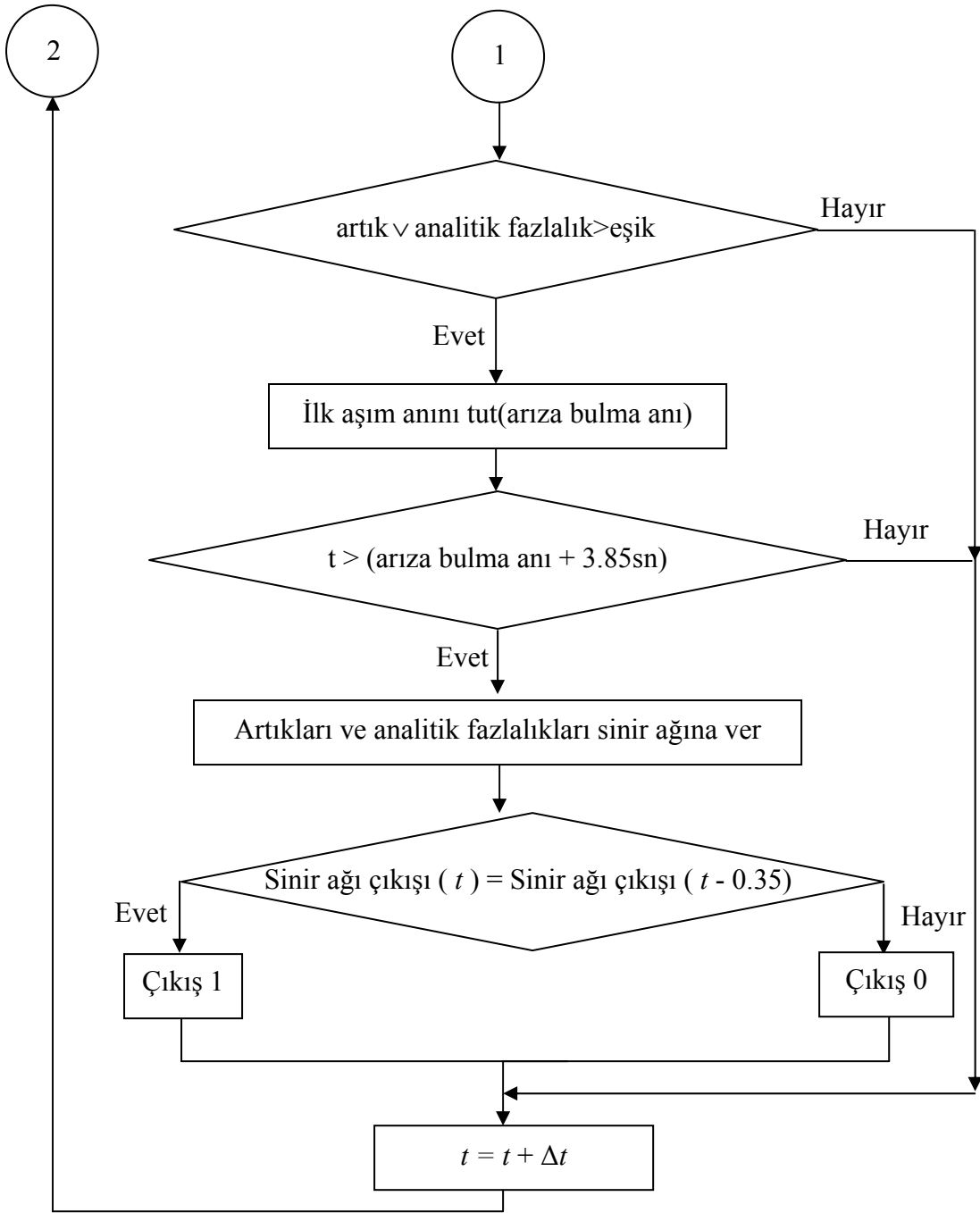
q2cap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot3);
q2uscap(i)=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot4);
r1(i)=q1(i+1)-q1cap(i);
r2(i)=q1us(i+1)-q1uscap(i);
r3(i)=q2(i+1)-q2cap(i);
r4(i)=q2us(i+1)-q2uscap(i);
r2r2(i)=(r1(i)-r1(i-1))/deltat;
r4r4(i)=(r3(i)-r3(i-1))/deltat;
p=[r1(i);r2(i);r3(i);r4(i);r2r2(i);r4r4(i)];
if (r1(i)>0.009|r1(i)<-0.0105)|(r2(i)>0.0025|r2(i)<-0.0035)|(r3(i)>0.015|r3(i)<-
0.01)|(r4(i)>0.0025|r4(i)<-0.0015)|...
(r2r2(i)>0.015|r2r2(i)<-0.015)|(r4r4(i)>0.02|r4r4(i)<-0.02)

% i
% disp('fault')
faulty(i)=1;% Fault detection time
I=find(faulty);
if(i>I(1)+350)%fault detection sonrası geçici zaman
faults1=sim(network1,p);
% pause;

faults=[faults faults1];
thr=thr+1;
if(i>I(1)+420) % Sinir ağı çıkışlarında 34 adım boyunca süreklilik sağlanıyor.
if(faults(1,thr-69)>0.5&faults(1,thr-34)>0.5&faults(1,thr)>=0)||...
(faults(1,thr-69)>0.5&faults(1,thr-34)>=0&faults(1,thr)>0.5)||
(faults(1,thr-69)>=0&faults(1,thr-34)>0.5&faults(1,thr)>0.5)
faults2(1,i)=0.9;
else
faults2(1,i)=0.1;
end
if(faults(2,thr-69)>0.5&faults(2,thr-34)>0.5&faults(2,thr)>=0)||...
(faults(2,thr-69)>0.5&faults(2,thr-34)>=0&faults(2,thr)>0.5)||
(faults(2,thr-69)>=0&faults(2,thr-34)>0.5&faults(2,thr)>0.5)
faults2(2,i)=0.9;
else
faults2(2,i)=0.1;
end
if(faults(3,thr-69)>0.5&faults(3,thr-34)>0.5&faults(3,thr)>=0)||...
(faults(3,thr-69)>0.5&faults(3,thr-34)>=0&faults(3,thr)>0.5)||
(faults(3,thr-69)>=0&faults(3,thr-34)>0.5&faults(3,thr)>0.5)
faults2(3,i)=0.9;
else
faults2(3,i)=0.1;
end
if(faults(4,thr-69)>0.5&faults(4,thr-34)>0.5&faults(4,thr)>0)||...
(faults(4,thr-69)>0.5&faults(4,thr-34)>=0&faults(4,thr)>0.5)||
(faults(4,thr-69)>=0&faults(4,thr-34)>0.5&faults(4,thr)>0.5)
faults2(4,i)=0.9;
else
faults2(4,i)=0.1;
end
end
end
else
i
% disp('no fault')
end
end

```





## Ek D. Önerilen 2. FDI Şemasına Ait MATLAB Programı ve Akış Diyagramı

C:\MATLAB6p5\Computed\_torque\CT\_GOS\CT\_act\_fault\_GOS\_SIU\_tez.m

//

```

clear all
close all
tic
a1=1;
a2=1;
m1=1;
m2=1;
g1=pi*6/180;
g2=pi*6/180;
wn=10;
kp=100;
kv=20;
ki=500;
deltat=0.01;
g=9.81;
Kp=[100 0;0 100];
Kv=[20 0;0 20];
Ki=[500 0;0 500];
t=0.01:0.01:70.01;
tt=2400
qd1=0.8*cos(t/2);
qd2=0.8*sin(t/2);
for i=2:7000
    qd1us(i)=(qd1(i)-qd1(i-1))/deltat;
    qd2us(i)=(qd2(i)-qd2(i-1))/deltat;

    qd1uss(i)=(qd1us(i)-qd1us(i-1))/deltat;
    qd2uss(i)=(qd2us(i)-qd2us(i-1))/deltat;
end

% Comp_torque_PID_ANFIS

q1(1)=0.1;
q2(1)=0;
q1us(1)=0;
q2us(1)=0;
q1(2)=0.1;
q2(2)=0;
q1us(2)=0;
q2us(2)=0;

e1(1)=qd1(1)-q1(1);
e2(1)=qd2(1)-q2(1);
e1(2)=e1(1);
e2(2)=e2(1);

e1us(2)=(e1(2)-e1(1))/deltat;
e2us(2)=(e2(2)-e2(1))/deltat;

eps1(1)=0;
eps2(1)=0;

```

```

eps1(2)=eps1(1)+deltat*e1(1);
eps2(2)=eps2(1)+deltat*e2(1);

q1us(1)=(q1(2)-q1(1))/deltat;
q2us(1)=(q2(2)-q2(1))/deltat;

robot1=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q1\robot_q1_deneme.fis');
robot2=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data\robot_q1us.fis');
robot3=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2\robot_q2.fis');
robot4=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2us\robot_q2us.fis');

robot5=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau105_q1\robot_tau105_q1.fis');
robot6=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau105_q1\robot_tau105_q1us.fis');
);
robot7=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau105_q1\robot_tau105_q2.fis');
robot8=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau105_q1\robot_tau105_q2us.fis');
);

robot9=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau107\robot_tau107_q1.fis');
robot10=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau107\robot_tau107_q1us.fis');
robot11=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau107\robot_tau107_q2.fis');
robot12=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau107\robot_tau107_q2us.fis');

robot13=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau205\robot_tau205_q1.fis');
robot14=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau205\robot_tau205_q1us.fis');
robot15=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau205\robot_tau205_q2.fis');
robot16=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau205\robot_tau205_q2us.fis');
%
robot17=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau207\robot_tau207_q1.fis');
robot18=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau207\robot_tau207_q1us.fis');
robot19=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau207\robot_tau207_q2.fis');
robot20=readfis('C:\MATLAB6p5\Computed_torque\CT_GOS\ANFIS_tau207\robot_tau207_q2us.fis');
faults=[];
faults2=[];
f2=0;%Diğer koşulların oluşup oluşmadığını kontrol için
f3=0;
f4=0;
f5=0;
f6=0;
f1_i=ones(1,400);
f2_i=ones(1,400);
for i=2:5000
    m2=1;
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M22(i)=m2*a2^2;
    M=[M11(i) M12(i);M21(i) M22(i)];

    V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i))+q2us(i)^2)*sin(q2(i));
    V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
    V=[V1(i);V2(i)];

    G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
    G2(i)=m2*g*a2*cos(q1(i)+q2(i));
    G=[G1(i);G2(i)];

```

```

tau1(i+1)=[M11(i)
M12(i)]*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V1(i)+G1(i)
);
tau2(i+1)=[M21(i)
M22(i)]*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V2(i)+G2(i)
);
tau=[tau1(i+1);tau2(i+1)];

% belirsizlik ekleme
uncertainty_function
if (i>tt) %ariza anı
% actuator fault
tau=[tau(1)*0.5;tau(2)*1];
end
quss=inv(Ma)*(tau-Va-Ga);

q1uss(i+1)=quss(1);
q2uss(i+1)=quss(2);
q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
q1(i+1)=q1(i)+deltat*q1us(i+1);
q2(i+1)=q2(i)+deltat*q2us(i+1);

e1(i+1)=qd1(i+1)-q1(i+1);
e2(i+1)=qd2(i+1)-q2(i+1);
e1us(i+1)=qd1us(i)-q1us(i);
e2us(i+1)=qd2us(i)-q2us(i);
eps1(i+1)=eps1(i)+deltat*e1(i+1);
eps2(i+1)=eps2(i)+deltat*e2(i+1);

q1cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot1);
q1uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot2);
q2cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot3);
q2uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot4);

q1cap_tau105=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot5);
q1uscap_tau105=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot6);
q2cap_tau105=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot7);
q2uscap_tau105=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot8);

q1cap_tau107=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot9);
q1uscap_tau107=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot10);
q2cap_tau107=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot11);
q2uscap_tau107=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot12);

q1cap_tau205=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot13);
q1uscap_tau205=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot14);
q2cap_tau205=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot15);
q2uscap_tau205=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot16);
%
q1cap_tau207=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot17);
q1uscap_tau207=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot18);
q2cap_tau207=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot19);
q2uscap_tau207=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot20);

r1(i)=q1(i+1)-q1cap;
r2(i)=q1us(i+1)-q1uscap;
r3(i)=q2(i+1)-q2cap;

```

```

r4(i)=q2us(i+1)-q2uscap;

r1_tau105(i)=q1(i+1)-q1cap_tau105;
r2_tau105(i)=q1us(i+1)-q1uscap_tau105;
r3_tau105(i)=q2(i+1)-q2cap_tau105;
r4_tau105(i)=q2us(i+1)-q2uscap_tau105;

r1_tau107(i)=q1(i+1)-q1cap_tau107;
r2_tau107(i)=q1us(i+1)-q1uscap_tau107;
r3_tau107(i)=q2(i+1)-q2cap_tau107;
r4_tau107(i)=q2us(i+1)-q2uscap_tau107;

r1_tau205(i)=q1(i+1)-q1cap_tau205;
r2_tau205(i)=q1us(i+1)-q1uscap_tau205;
r3_tau205(i)=q2(i+1)-q2cap_tau205;
r4_tau205(i)=q2us(i+1)-q2uscap_tau205;
%
r1_tau207(i)=q1(i+1)-q1cap_tau207;
r2_tau207(i)=q1us(i+1)-q1uscap_tau207;
r3_tau207(i)=q2(i+1)-q2cap_tau207;
r4_tau207(i)=q2us(i+1)-q2uscap_tau207;
if i>400
% faults = [healty tau105 tau107 tau205 tau207]
% if
% (r1(i)<0.008&r1(i)>-0.008)&(r2(i)<0.003&r2(i)>-0.003)&(r3(i)<0.022&r
% 3(i)>-0.022)&(r4(i)<0.0045&r4(i)>-0.0045) eski dosya
% if (r1(i)<0.008&r1(i)>-0.008)&(r2(i)<0.004&r2(i)>-0.004)&(r3(i)<0.03&r3(i)>-
0.03)&(r4(i)<0.005&r4(i)>-0.005)
% NN_func_app'den alınmiş eşikler
if (r1(i)<0.011&r1(i)>-0.01)&(r2(i)<0.0035&r2(i)>-0.003)&(r3(i)<0.015&r3(i)>-
0.013)&(r4(i)<0.0025&r4(i)>-0.0025)
f2_i(i)=1;
if f2_i(i-80)==1
f1_i(i)=1;
f1=[1 0 0 0];
faults=[faults;f1];
faults2=[faults2;f1];
f6=f6+1;
continue
else f1_i(i)=0;

end
end
if (r1_tau107(i)<0.013&r1_tau107(i)>-0.013)&(r2_tau107(i)<0.0018&r2_tau107(i)>-
0.0018)&(r3_tau107(i)<0.025&r3_tau107(i)>-0.023)&...
(r4_tau107(i)<0.0022&r4_tau107(i)>-0.0018)
f1=[0 0 1 0 0];
f2=f2+1;
end
if ((r1_tau105(i)<0.015&r1_tau105(i)>-0.015)&(r2_tau105(i)<0.0022&r2_tau105(i)>-
0.0024)&(r3_tau105(i)<0.02&r3_tau105(i)>-0.02)&...
(r4_tau105(i)<0.0055&r4_tau105(i)>-0.0055))&((r1_tau105(i)<0.015&r1_tau105(i)>-
0.015)&(r2_tau105(i)<0.0022&r2_tau105(i)>-0.0024)&(r3_tau105(i)<0.02&r3_tau105(i)>-0.02)&...
(r4_tau105(i)<0.0055&r4_tau105(i)>-0.0055)&(r1_tau107(i)<0.013&r1_tau107(i)>-
0.013)&(r2_tau107(i)<0.0018&r2_tau107(i)>-0.0018)&(r3_tau107(i)<0.025&r3_tau107(i)>-0.023)&...
(r4_tau107(i)<0.0022&r4_tau107(i)>-0.0018))
f1=[0 1 0 0 0];
f3=f3+1;
end

```

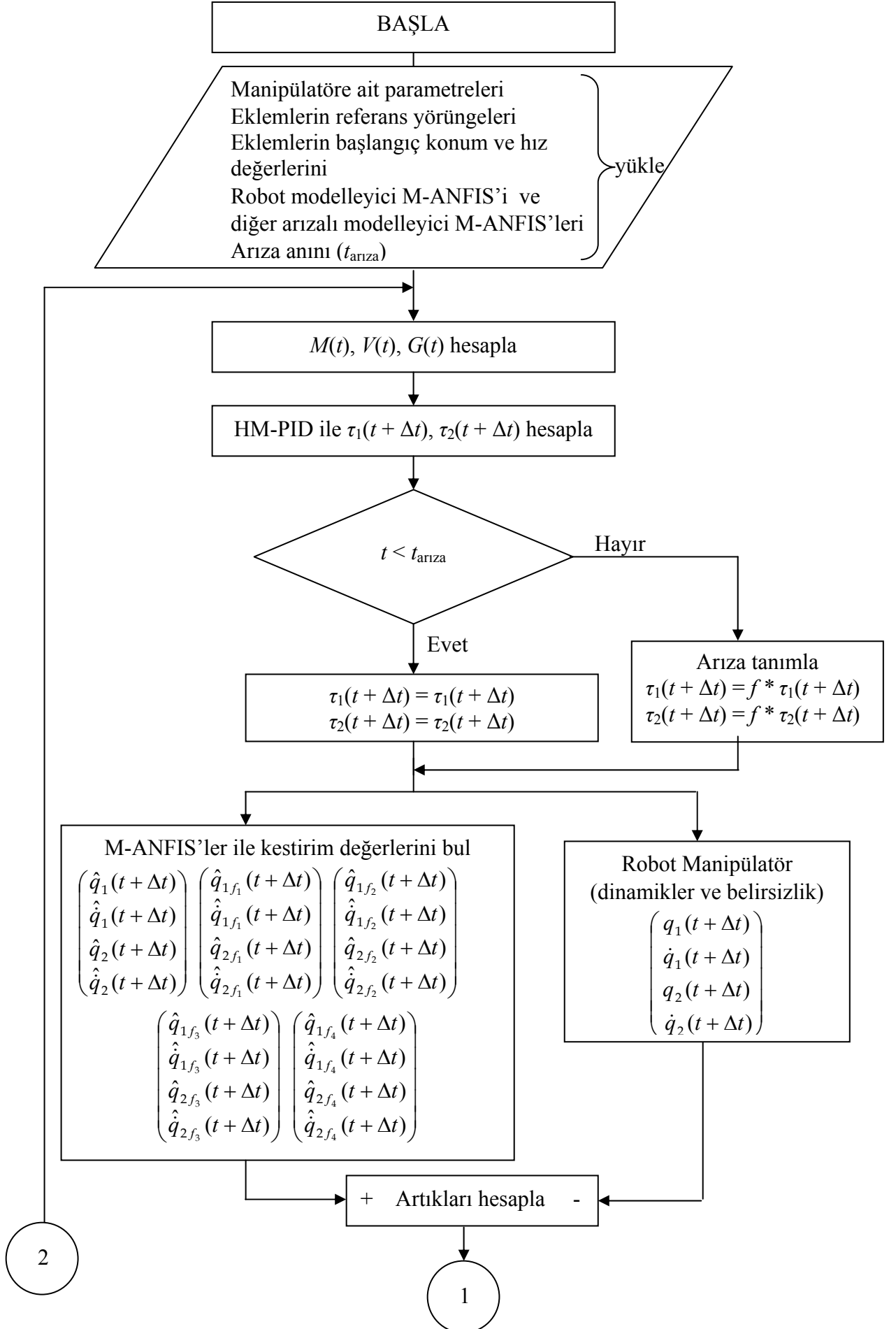
```

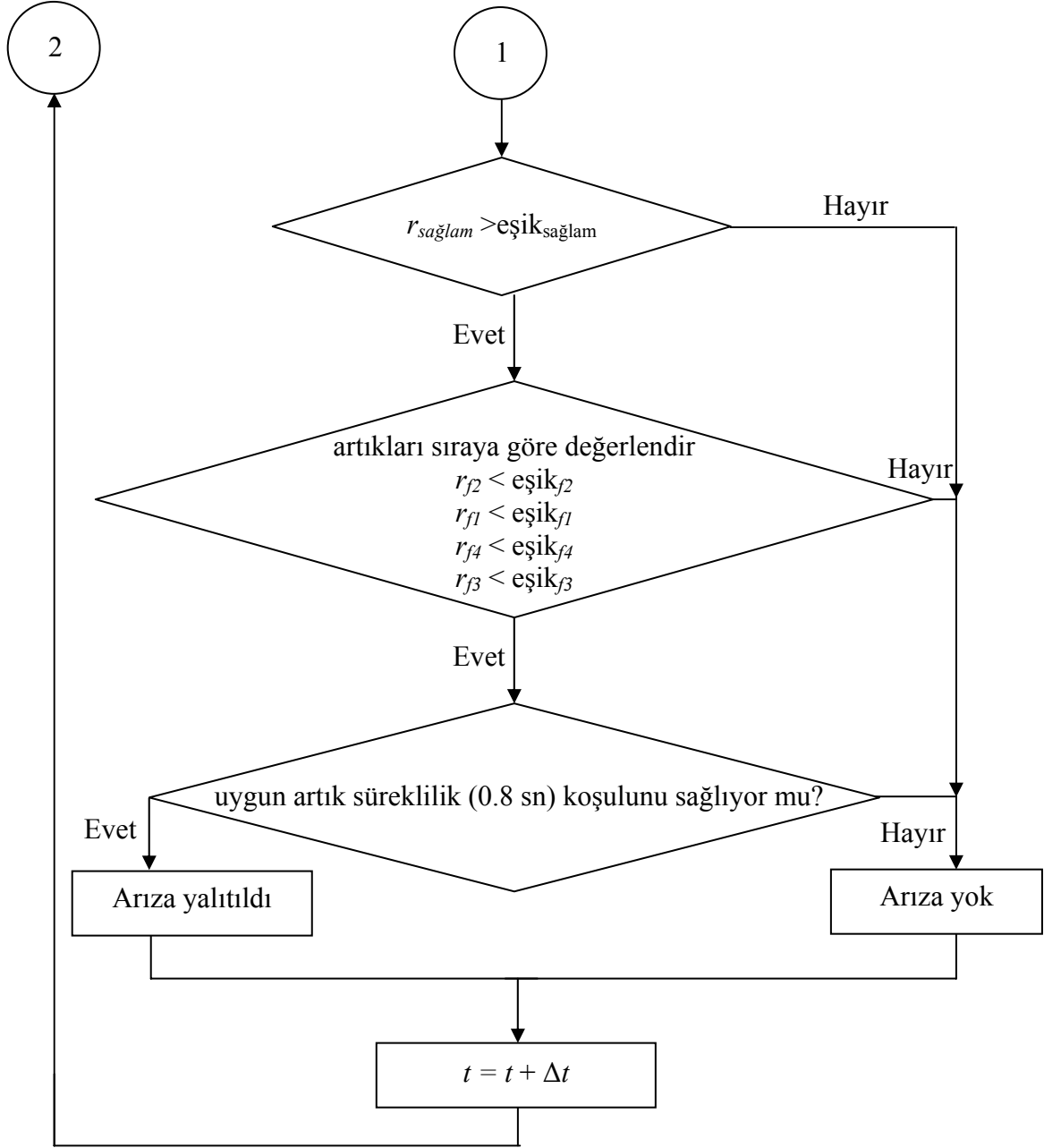
    if (r1_tau207(i)<0.016&r1_tau207(i)>-0.015)&(r2_tau207(i)<0.0014&r2_tau207(i)>-
0.0015)&(r3_tau207(i)<0.025&r3_tau207(i)>-0.022)&...
        (r4_tau207(i)<0.0014&r4_tau207(i)>-0.0013)
        f1=[0 0 0 0 1];
        f4=f4+1;
    end
    if ((r1_tau205(i)<0.016&r1_tau205(i)>-0.014)&(r2_tau205(i)<0.0042&r2_tau205(i)>-
0.005)&(r3_tau205(i)<0.025&r3_tau205(i)>-0.022)&...
        (r4_tau205(i)<0.013&r4_tau205(i)>-0.012))|((r1_tau205(i)<0.02&r1_tau205(i)>-
0.02)&(r2_tau205(i)<0.005&r2_tau205(i)>-0.005)&(r3_tau205(i)<0.025&r3_tau205(i)>-0.025)&...
        (r4_tau205(i)<0.015&r4_tau205(i)>-0.015)&(r1_tau207(i)<0.016&r1_tau207(i)>-
0.015)&(r2_tau207(i)<0.0014&r2_tau207(i)>-0.0015)&(r3_tau207(i)<0.025&r3_tau207(i)>-0.022)&...
        (r4_tau207(i)<0.0014&r4_tau207(i)>-0.0013))
        f1=[0 0 0 1 0];
        f5=f5+1;
    end

%   faults=[faults;f1];
    faults2=[faults2;f1];
%   f1=[0 0 0 0 0];% Hiçbirkoşul tutmuyorsa
%   faults2=[faults2;f1];
    if faults2(i-480,1)==faults2(i-400,1)&faults2(i-480,2)==faults2(i-400,2)&faults2(i-480,3)==faults2(i-
400,3)&...
        faults2(i-480,4)==faults2(i-400,4)&faults2(i-480,5)==faults2(i-400,5)
        faults=[faults;f1];
    else

        faults=[faults;faults(i-401,:)];
    end
end
% f1=[0 0 0 0 0];%Hiçbir koşul tutmuyorsa
end

```





### Ek E. Önerilen 3. FDI Şemasına Ait MATLAB Programı ve Akış Diyagramı

C:\MATLAB6p5\Computed\_torque\residual generation\CT\_actuator\_fault\_func\_app\_SIU\_tez.m

//

```

clear all
close all
tic
a1=1;
a2=1;
m1=1;
m2=1;
g1=pi*6/180;
g2=pi*6/180;
wn=10;
kp=100;
kv=20;
ki=500;
deltat=0.01;
g=9.81;
Kp=[100 0;0 100];
Kv=[20 0;0 20];
Ki=[500 0;0 500];
t=0.01:0.01:60.01;

tt=2400;
qd1=0.8*cos(t/2);
qd2=0.8*sin(t/2);

for i=2:6000
    qd1us(i)=(qd1(i)-qd1(i-1))/deltat;
    qd2us(i)=(qd2(i)-qd2(i-1))/deltat;

    qd1uss(i)=(qd1us(i)-qd1us(i-1))/deltat;
    qd2uss(i)=(qd2us(i)-qd2us(i-1))/deltat;
end

% Comp_torque_PID_ANFIS

q1(1)=0.1;
q2(1)=0;
q1us(1)=0;
q2us(1)=0;
q1(2)=0.1;
q2(2)=0;
q1us(2)=0;
q2us(2)=0;

e1(1)=qd1(1)-q1(1);
e2(1)=qd2(1)-q2(1);
e1(2)=e1(1);
e2(2)=e2(1);

e1us(2)=(e1(2)-e1(1))/deltat;
e2us(2)=(e2(2)-e2(1))/deltat;

```

```

eps1(1)=0;
eps2(1)=0;
eps1(2)=eps1(1)+deltat*e1(1);
eps2(2)=eps2(1)+deltat*e2(1);

q1us(1)=(q1(2)-q1(1))/deltat;
q2us(1)=(q2(2)-q2(1))/deltat;

robot1=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q1\robot_q1_deneme.fis');
robot2=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data\robot_q1us.fis');
robot3=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2\robot_q2.fis');
robot4=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2us\robot_q2us.fis');

faults_1=[];
faults_2=[];

func_app1=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS_NN_func_app\func_app_f1_genisletil
mis_15_epoch.fis');
func_app2=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS_NN_func_app\func_app_f2_genisletil
mis_40_epoch.fis');

a=0;
for i=2:400
    m2=1;
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M22(i)=m2*a2^2;
    M=[M11(i) M12(i);M21(i) M22(i)];

    V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i))+q2us(i)^2)*sin(q2(i));
    V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
    V=[V1(i);V2(i)];

    G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
    G2(i)=m2*g*a2*cos(q1(i)+q2(i));
    G=[G1(i);G2(i)];

    tau1(i+1)=[M11(i)
M12(i)*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V1(i)+G1(i)
);
    tau2(i+1)=[M21(i)
M22(i)*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V2(i)+G2(i)
);
    tau=[tau1(i+1);tau2(i+1)];

    % belirsizlik ekleme
    uncertainty_function

    quss=inv(Ma)*(tau-Va-Ga);

    q1uss(i+1)=quss(1);
    q2uss(i+1)=quss(2);
    q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
    q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
    q1(i+1)=q1(i)+deltat*q1us(i+1);
    q2(i+1)=q2(i)+deltat*q2us(i+1);

```

```

e1(i+1)=qd1(i+1)-q1(i+1);
e2(i+1)=qd2(i+1)-q2(i+1);
e1us(i+1)=qd1us(i)-q1us(i);
e2us(i+1)=qd2us(i)-q2us(i);
eps1(i+1)=eps1(i)+deltat*e1(i+1);
eps2(i+1)=eps2(i)+deltat*e2(i+1);

q1cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot1);
q1uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot2);
q2cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot3);
q2uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot4);
r1(i)=q1(i+1)-q1cap;
r2(i)=q1us(i+1)-q1uscap;
r3(i)=q2(i+1)-q2cap;
r4(i)=q2us(i+1)-q2uscap;
r2r2(i)=(r1(i)-r1(i-1))/deltat;
r4r4(i)=(r3(i)-r3(i-1))/deltat;
p=[r1(i);r2(i);r3(i);r4(i);r2r2(i);r4r4(i)];

end

faults_1(400)=0;
faults_2(400)=0;
f_out1(401)=0;
f_out2(401)=0;
faulty=0;

faults1_i=zeros(1,tt+1);
thr=0;

for i=401:5000
    m2=1;
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M22(i)=m2*a2^2;
    M=[M11(i) M12(i);M21(i) M22(i)];

    V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i)+q2us(i)^2)*sin(q2(i));
    V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
    V=[V1(i);V2(i)];

    G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
    G2(i)=m2*g*a2*cos(q1(i)+q2(i));
    G=[G1(i);G2(i)];

    tau1(i+1)=[M11(i)
M12(i)*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V1(i)+G1(i)
);
    tau2(i+1)=[M21(i)
M22(i)*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V2(i)+G2(i)
);
    tau=[tau1(i+1);tau2(i+1)];

    % belirsizlik ekleme
    uncertainty_function
    if (i>tt)
    % actuator fault

```

```

    f_tau1(i+1)=0.5*tau(1);
    f_tau2(i+1)=0;

    tau=[tau(1)*0.5;tau(2)*1];

end
quss=inv(Ma)*(tau-Va-Ga);

q1uss(i+1)=quss(1);
q2uss(i+1)=quss(2);
q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
q1(i+1)=q1(i)+deltat*q1us(i+1);
q2(i+1)=q2(i)+deltat*q2us(i+1);

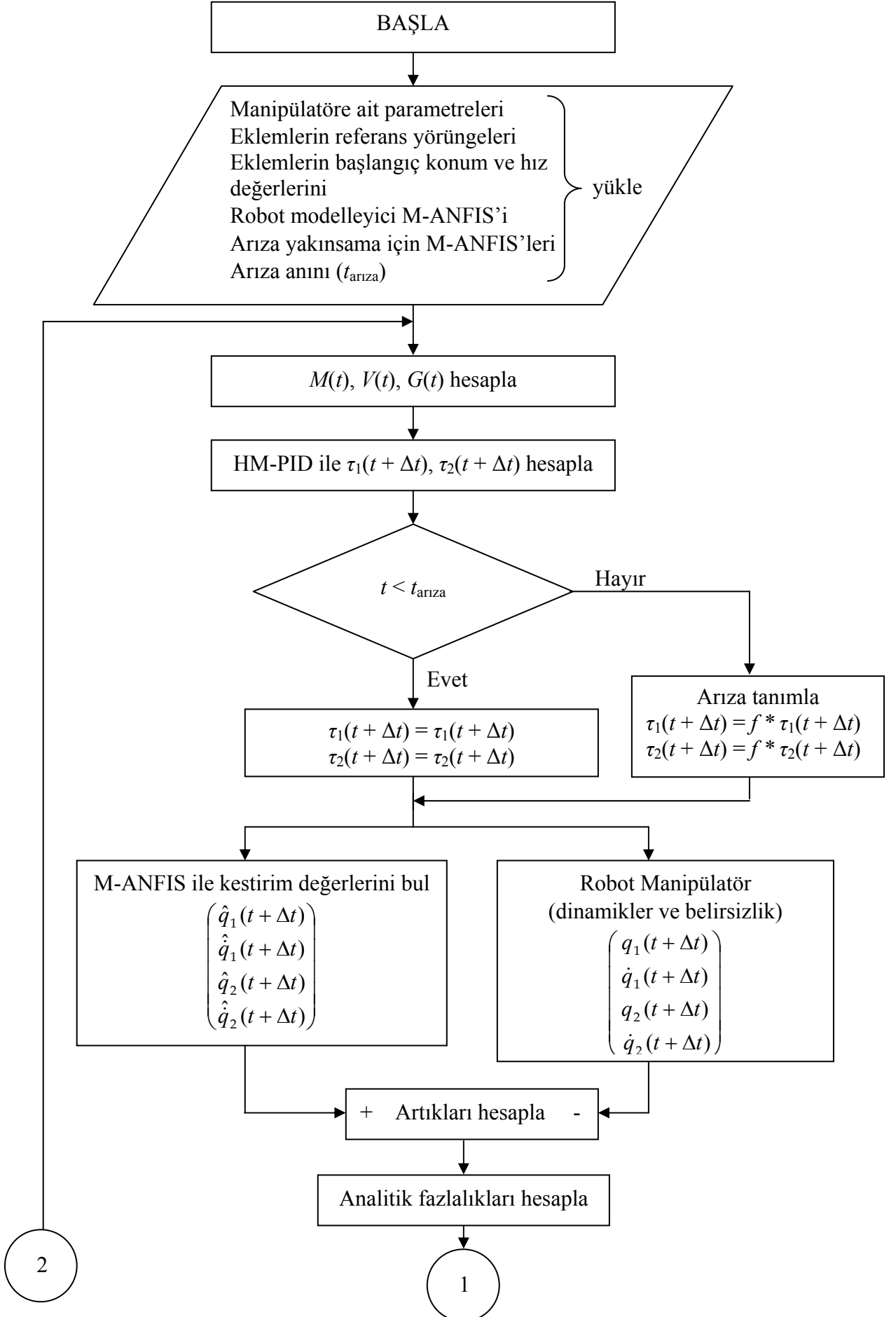
e1(i+1)=qd1(i+1)-q1(i+1);
e2(i+1)=qd2(i+1)-q2(i+1);
e1us(i+1)=qd1us(i)-q1us(i);
e2us(i+1)=qd2us(i)-q2us(i);
eps1(i+1)=eps1(i)+deltat*e1(i+1);
eps2(i+1)=eps2(i)+deltat*e2(i+1);

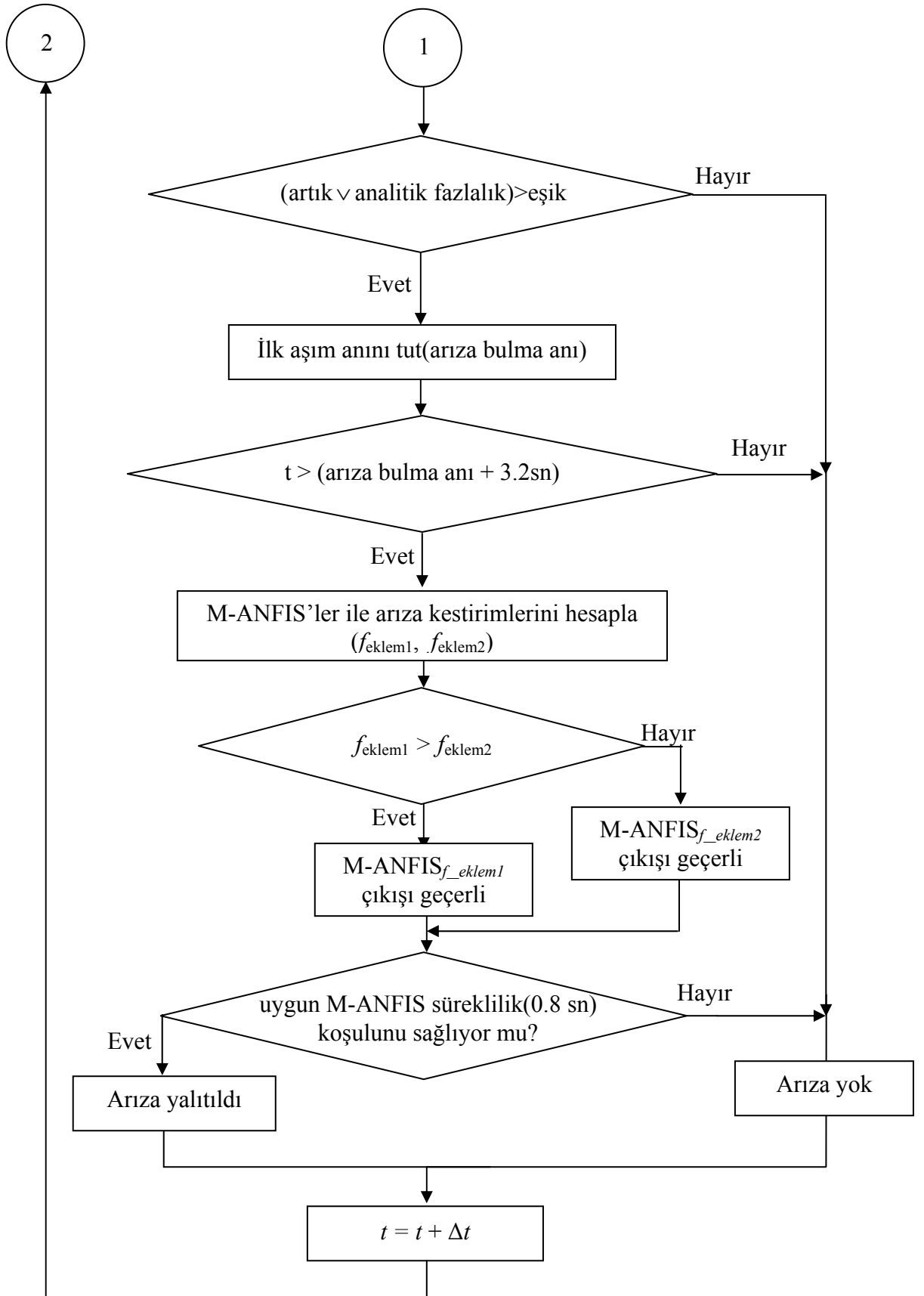
q1cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot1);
q1uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot2);
q2cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot3);
q2uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot4);
r1(i)=q1(i+1)-q1cap;
r2(i)=q1us(i+1)-q1uscap;
r3(i)=q2(i+1)-q2cap;
r4(i)=q2us(i+1)-q2uscap;
r2r2(i)=(r1(i)-r1(i-1))/deltat;
r4r4(i)=(r3(i)-r3(i-1))/deltat;
p=[r1(i);r2(i);r3(i);r4(i);r1(i-1);r2(i-1);r3(i-1);r4(i-1);...
;r1(i-2);r2(i-2);r3(i-2);r4(i-2)];
if (r1(i)>0.011|r1(i)<-0.01)|(r2(i)>0.0035|r2(i)<-0.003)|(r3(i)>0.015|r3(i)<-0.013)|(r4(i)>0.0025|r4(i)<-
0.0025)|...
(r2r2(i)>0.015|r2r2(i)<-0.015)|(r4r4(i)>0.016|r4r4(i)<-0.017)
    if a==0
        a=i;
    end
    if a~=0&i>=a+400

        faults1=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app1);
        faults2=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app2);
        faults_1(i+1)=faults1;
        faults_2(i+1)=faults2;
        faults_11(i+1)=faults1;
        faults_22(i+1)=faults2;
        if faults_1(i+1)>faults_2(i+1)
            f_out1(i)=1;
            f_out2(i)=0;
        else
            f_out2(i)=1;
            f_out1(i)=0;
        end
    else
        faults_1(i+1)=0;
        faults_2(i+1)=0;
        faults_11(i+1)=0;

```

```
        faults_22(i+1)=0;
    end
else
    faults_1(i+1)=0;
    faults_2(i+1)=0;
    faults_11(i+1)=0;
    faults_22(i+1)=0;
end
    if faults_1(i+1)==0&faults_2(i+1)==0&faults_1(i-80)==0&faults_2(i-80)==0
        faults_11(i+1)=0;
        faults_22(i+1)=0;
    else
        faults_11(i+1)=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app1);
        faults_22(i+1)=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app2);
        if faults_11(i+1)>faults_22(i+1)
            f_out1(i)=1;
            f_out2(i)=0;
        else
            f_out2(i)=1;
            f_out1(i)=0;
        end
    end
end
end
```







```

e1us(2)=(e1(2)-e1(1))/deltat;
e2us(2)=(e2(2)-e2(1))/deltat;

eps1(1)=0;
eps2(1)=0;
eps1(2)=eps1(1)+deltat*e1(1);
eps2(2)=eps2(1)+deltat*e2(1);

q1us(1)=(q1(2)-q1(1))/deltat;
q2us(1)=(q2(2)-q2(1))/deltat;

robot1=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q1\robot_q1_deneme.fis');
robot2=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data\robot_q1us.fis');
robot3=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2\robot_q2.fis');
robot4=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS\ANFIS_data_q2us\robot_q2us.fis');

faults_1=[];
faults_2=[];

func_app1=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS_NN_func_app\func_app_f1_genisletil
mis_15_epoch.fis');
func_app2=readfis('C:\MATLAB7\Computed_torque\CT_ANFIS_NN_func_app\func_app_f2_genisletil
mis_40_epoch.fis');

a=0;
for i=2:300
    m2=1;
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M22(i)=m2*a2^2;
    M=[M11(i) M12(i);M21(i) M22(i)];

    V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i))+q2us(i)^2)*sin(q2(i));
    V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
    V=[V1(i);V2(i)];

    G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
    G2(i)=m2*g*a2*cos(q1(i)+q2(i));
    G=[G1(i);G2(i)];

    tau1(i+1)=[M11(i)
M12(i)*([qd1uss(i);qd2uss(i))+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V1(i)+G1(i)
);
    tau2(i+1)=[M21(i)
M22(i)*([qd1uss(i);qd2uss(i))+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V2(i)+G2(i)
);
    tau=[tau1(i+1);tau2(i+1)];

    % belirsizlik ekleme
    uncertainty_function

    quss=inv(Ma)*(tau-Va-Ga);

    q1uss(i+1)=quss(1);
    q2uss(i+1)=quss(2);
    q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
    q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
    q1(i+1)=q1(i)+deltat*q1us(i+1);

```

```

q2(i+1)=q2(i)+deltat*q2us(i+1);

e1(i+1)=qd1(i+1)-q1(i+1);
e2(i+1)=qd2(i+1)-q2(i+1);
e1us(i+1)=qd1us(i)-q1us(i);
e2us(i+1)=qd2us(i)-q2us(i);
eps1(i+1)=eps1(i)+deltat*e1(i+1);
eps2(i+1)=eps2(i)+deltat*e2(i+1);

q1cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot1);
q1uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot2);
q2cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot3);
q2uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot4);
r1(i)=q1(i+1)-q1cap;
r2(i)=q1us(i+1)-q1uscap;
r3(i)=q2(i+1)-q2cap;
r4(i)=q2us(i+1)-q2uscap;
r2r2(i)=(r1(i)-r1(i-1))/deltat;
r4r4(i)=(r3(i)-r3(i-1))/deltat;
p=[r1(i);r2(i);r3(i);r4(i);r2r2(i);r4r4(i)];

end

faults_1(300)=0;
faults_2(300)=0;
faults_11(6000)=0;

f_out1(301)=0;
f_out2(301)=0;
ftc=0;
b_ilk=0;
FTC_ratio=0;
artim=0.15;
faulty=0;
faults1_i=zeros(1,tt+1);
thr=0;
Kvii=Kvi;
Kpii=Kpi;
Kiii=Kii;
for i=301:son
    m2=1;
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
    M22(i)=m2*a2^2;
    M=[M11(i) M12(i);M21(i) M22(i)];

    V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i)+q2us(i)^2)*sin(q2(i));
    V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
    V=[V1(i);V2(i)];

    G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
    G2(i)=m2*g*a2*cos(q1(i)+q2(i));
    G=[G1(i);G2(i)];

    tau1(i+1)=[M11(i)
M12(i)]*( [qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)]+V1(i)+G1(i)
);

```

```

tau2(i+1)=[M21(i)
M22(i)]*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V2(i)+G2(i)
);

tau=[tau1(i+1);tau2(i+1)];

% belirsizlik ekleme
uncertainty_function

% actuator fault
if(i>2400) %ariza anı
    f_tau1(i+1)=0.5*tau1(i+1);
    f_tau2(i+1)=0;

    tau=[tau(1)*0.5;tau(2)*1];
end
quss=inv(Ma)*(tau-Va-Ga);

q1uss(i+1)=quss(1);
q2uss(i+1)=quss(2);
q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
q1(i+1)=q1(i)+deltat*q1us(i+1);
q2(i+1)=q2(i)+deltat*q2us(i+1);

e1(i+1)=qd1(i+1)-q1(i+1);
e2(i+1)=qd2(i+1)-q2(i+1);
e1us(i+1)=qd1us(i)-q1us(i);
e2us(i+1)=qd2us(i)-q2us(i);
eps1(i+1)=eps1(i)+deltat*e1(i+1);
eps2(i+1)=eps2(i)+deltat*e2(i+1);

q1cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot1);
q1uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot2);
q2cap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot3);
q2uscap=evalfis([tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i)],robot4);
r1(i)=q1(i+1)-q1cap;
r2(i)=q1us(i+1)-q1uscap;
r3(i)=q2(i+1)-q2cap;
r4(i)=q2us(i+1)-q2uscap;
r2r2(i)=(r1(i)-r1(i-1))/deltat;
r4r4(i)=(r3(i)-r3(i-1))/deltat;
p=[r1(i);r2(i);r3(i);r4(i);r1(i-1);r2(i-1);r3(i-1);r4(i-1);...
;r1(i-2);r2(i-2);r3(i-2);r4(i-2)];
if (r1(i)>0.011|r1(i)<-0.01)|(r2(i)>0.0035|r2(i)<-0.003)|(r3(i)>0.015|r3(i)<-0.013)|(r4(i)>0.0025|r4(i)<-
0.0025)|...
(r2r2(i)>0.015|r2r2(i)<-0.015)|(r4r4(i)>0.016|r4r4(i)<-0.017)
if a==0
    a=i;
end
if a~=0&i>=a+400

faults1=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app1);
faults2=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app2);
faults_1(i+1)=faults1;
faults_2(i+1)=faults2;
faults_11(i+1)=faults1;
faults_22(i+1)=faults2;
if faults_1(i+1)>faults_2(i+1)

```

```

        f_out1(i)=1;
        f_out2(i)=0;
    else
        f_out2(i)=1;
        f_out1(i)=0;
    end
else
    faults_1(i+1)=0;
    faults_2(i+1)=0;
    faults_11(i+1)=0;
    faults_22(i+1)=0;
end

else
    faults_1(i+1)=0;
    faults_2(i+1)=0;
    faults_11(i+1)=0;
    faults_22(i+1)=0;
end

%Sürekli test
if faults_1(i+1)==0&faults_2(i+1)==0&faults_1(i-80)==0&faults_2(i-80)==0
    faults_11(i+1)=0;
    faults_22(i+1)=0;
else
    faults_11(i+1)=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app1);
    faults_22(i+1)=evalfis([tau1(i+1) tau2(i+1) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1)],func_app2);
    if faults_11(i+1)>faults_22(i+1)
        f_out1(i)=1;
        f_out2(i)=0;
    else
        f_out2(i)=1;
        f_out1(i)=0;
    end
    bb=i;

    ftc=ftc+1;
    if ftc==1||bb>=b+40
        if b_ilk==0
            b_ilk=i;
        end
        if f_out1(i)==1
            if FTC_ratio==0
                FTC_ratio=(faults_11(i+1)/(tau1(i+1)));
            end

            %Katsayıların zamana bağlı artış miktarı
            if (i-b_ilk)<=floor(FTC_ratio*100)*4/(artim*10)
                if toplam_artim>=FTC_ratio
                    toplam_artim=FTC_ratio
                    yerlesme=bb-b_ilk
                    end%40 80 yerine
                    Kvii=Kvi+Kvi*toplam_artim
                    Kpii=Kpi+Kpi*toplam_artim
                    Kiii=Kii+Kii*toplam_artim
                end
            elseif f_out2(i)==1

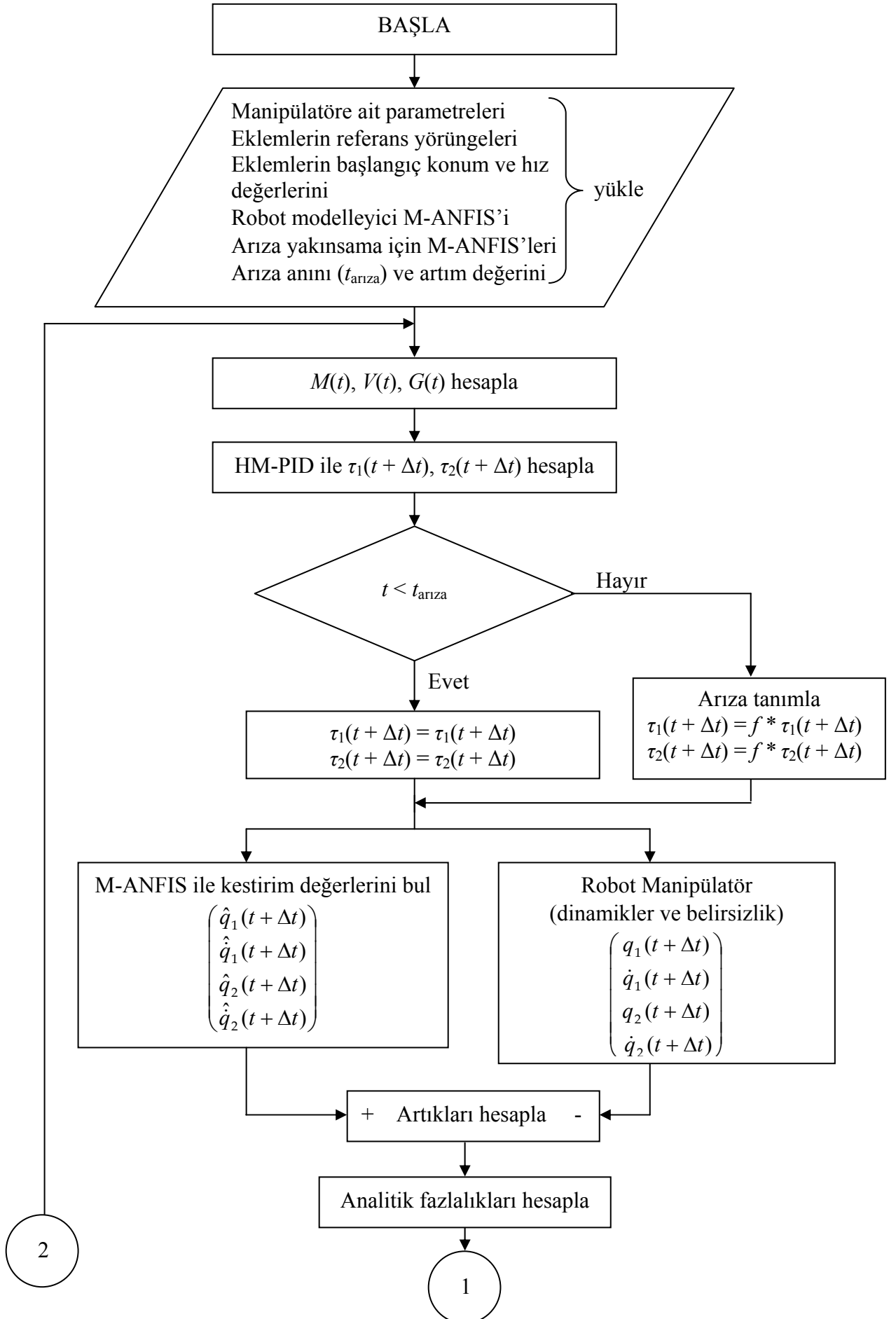
                if FTC_ratio==0

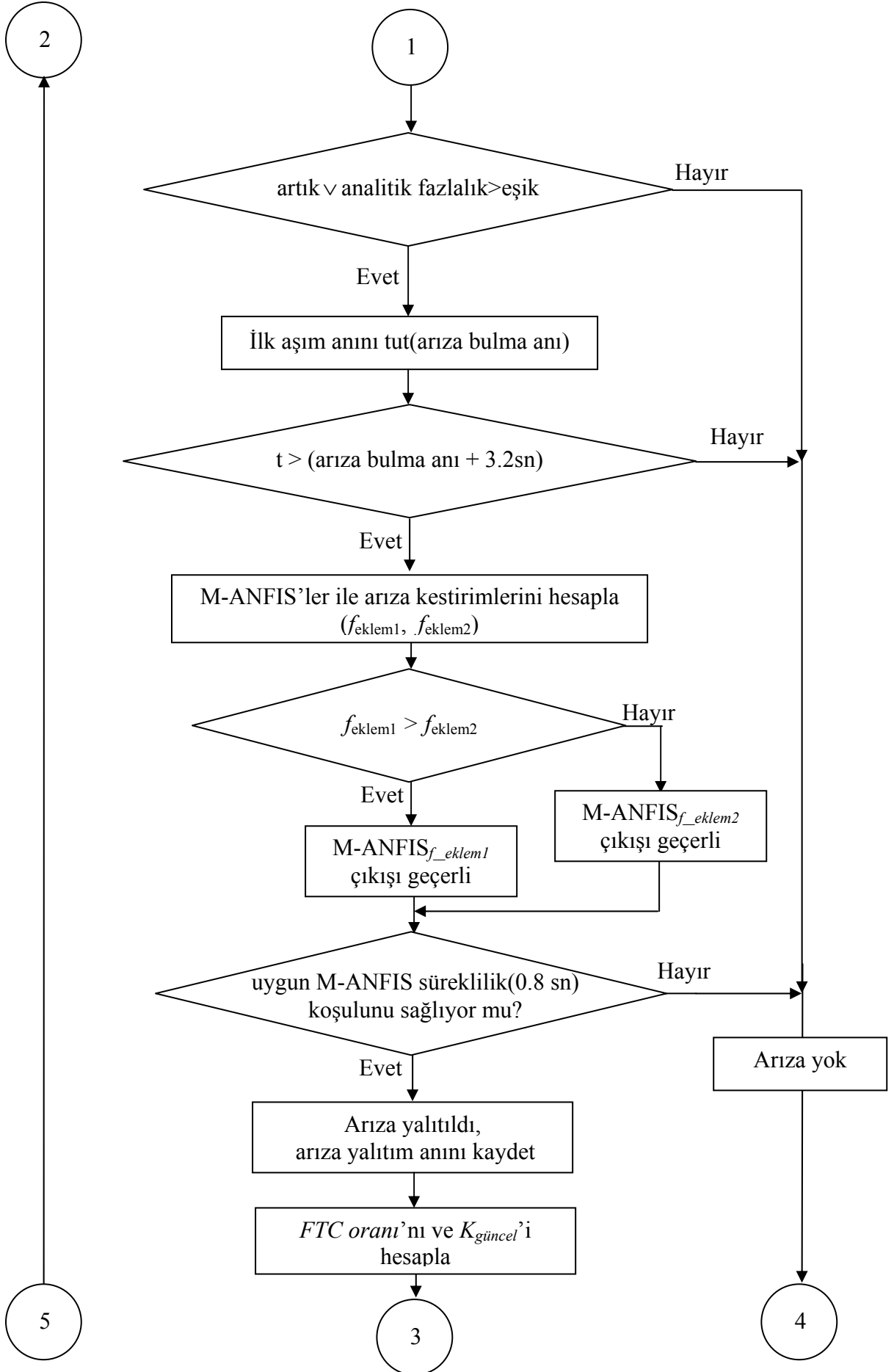
```

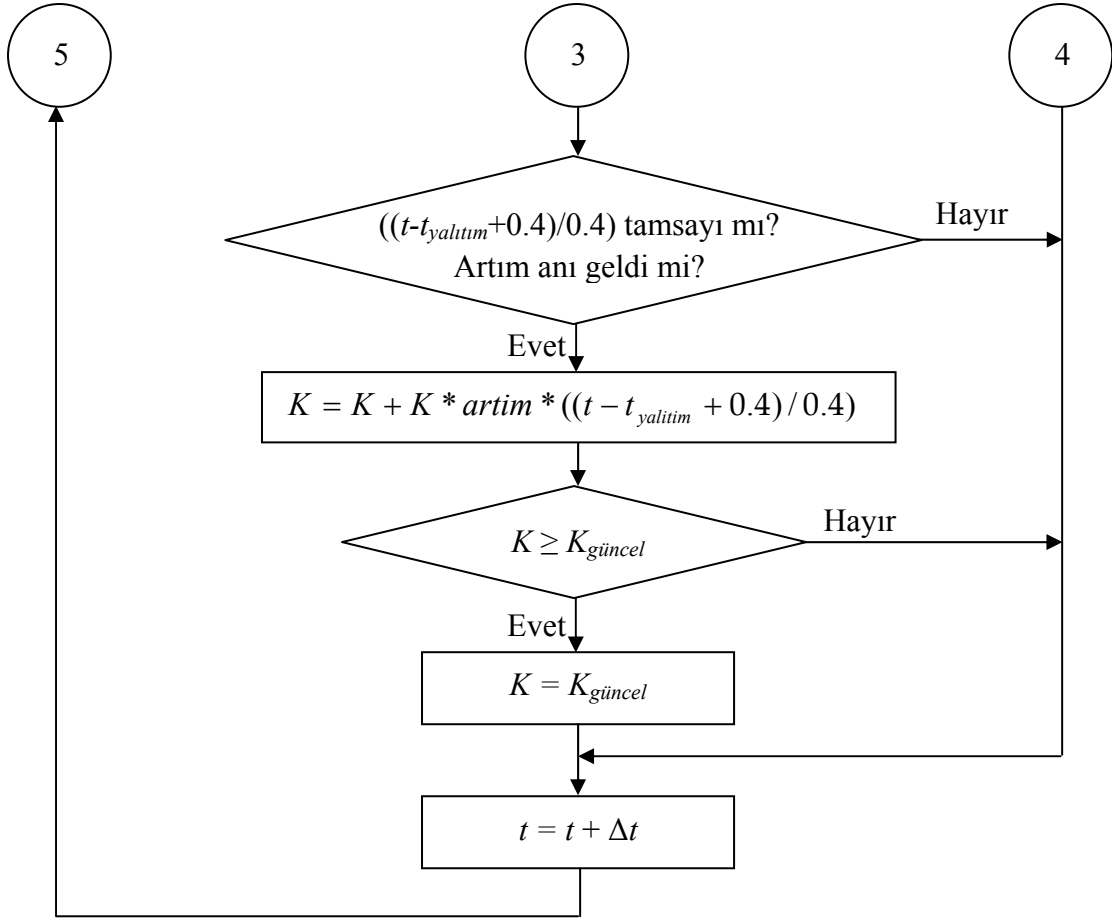
```

    FTC_ratio=(faults_22(i+1)/(tau2(i+1)));
end
if (i-b_ilk)<=floor(FTC_ratio*100)*4/(artim*10)
    i-b_ilk
    i
    FTC_ratio
    bb-b_ilk
    toplam_artim=artim*((bb-b_ilk+40)/40)
    if toplam_artim>=FTC_ratio
        toplam_artim=FTC_ratio
        yerlesme=bb-b_ilk
        pause
        end%40 80 yerine
    Kvii=Kvi+Kvi*toplam_artim
    Kpii=Kpi+Kpi*toplam_artim
    Kiii=Kii+Kii*toplam_artim
end
else
end
Kv=[Kvii 0;0 Kvii]
Kp=[Kpii 0;0 Kpii]
Ki=[Kiii 0;0 Kiii]
faults_22(i+1);
b=i;
end
end
end
end

```







## Ek G. 1. Şema İçin DSP Programı

D:\c6713\projects\sine\_floating\ Sine\_floating\_ariza\_ANFIS\_NN\_double.c

//

```
#include <math.h>
#include <stdio.h>
#define PI 3.14159265358
#pragma DATA_SECTION(tau1, ".EXT_RAM")
#pragma DATA_SECTION(tau2, ".EXT_RAM")
#pragma DATA_SECTION(qd1, ".EXT_RAM")
#pragma DATA_SECTION(qd2, ".EXT_RAM")
#pragma DATA_SECTION(r1, ".EXT_RAM")
#pragma DATA_SECTION(r2, ".EXT_RAM")
#pragma DATA_SECTION(r3, ".EXT_RAM")
#pragma DATA_SECTION(r4, ".EXT_RAM")
#pragma DATA_SECTION(r2r2, ".EXT_RAM")
#pragma DATA_SECTION(r4r4, ".EXT_RAM")
#pragma DATA_SECTION(faults1, ".EXT_RAM")
#pragma DATA_SECTION(faults2, ".EXT_RAM")
#pragma DATA_SECTION(faults3, ".EXT_RAM")
#pragma DATA_SECTION(faults4, ".EXT_RAM")
#pragma DATA_SECTION(faults, ".EXT_RAM")
#pragma DATA_SECTION(IW1, ".EXT_RAM")
#pragma DATA_SECTION(LW1, ".EXT_RAM")
#pragma DATA_SECTION(LW2, ".EXT_RAM")
#pragma DATA_SECTION(LW3, ".EXT_RAM")
#pragma DATA_SECTION(v_gecici1, ".EXT_RAM")
#pragma DATA_SECTION(v_gecici2, ".EXT_RAM")
#pragma DATA_SECTION(g_gecici1, ".EXT_RAM")
#pragma DATA_SECTION(g_gecici2, ".EXT_RAM")
#pragma DATA_SECTION(qd1us, ".EXT_RAM")
#pragma DATA_SECTION(qd2us, ".EXT_RAM")
#pragma DATA_SECTION(qd1uss, ".EXT_RAM")
#pragma DATA_SECTION(qd1uss, ".EXT_RAM")
#pragma DATA_SECTION(q1us, ".EXT_RAM")
#pragma DATA_SECTION(q2us, ".EXT_RAM")
#pragma DATA_SECTION(q1cap, ".EXT_RAM")
#pragma DATA_SECTION(q1uscap, ".EXT_RAM")
#pragma DATA_SECTION(q2cap, ".EXT_RAM")
#pragma DATA_SECTION(q2uscap, ".EXT_RAM")

/* #pragma DATA_SECTION(q1cap, "SDRAM") */
#include "q1cap_coef_double.h" //q1cap ANFIS kaysayıları
#include "q1uscap_coef_double.h" //q1uscap ANFIS kaysayıları
#include "q2cap_coef_double.h" //q2cap ANFIS kaysayıları
#include "q2uscap_coef_double.h" //q2uscap ANFIS kaysayıları
#include "Neural_network_coef_double.h" //q2uscap ANFIS kaysayıları

double ANFIS1(double ANFIS_giris[6],double a_ai22[6][2],double b_bi22[6][2],double
c_ci22[6][2],double pqr22[64][7]);
void NNet1(double NNet_giris[6],double IW[10][6],double LW11[20][10],double LW22[20][20],double
LW33[4][20],double bias11[10],double bias22[20],double bias33[20],double bias44[4]);

double deltat=0.01,a1=1,a2=1,m1=1,m2=1,g=9.81;
double qd1[4001],qd2[4001],t[4001];
double q1us_eski,q2us_eski,q1uss_eski,q2uss_eski;
double q1us[4001],q2us[4001];
double e1_eski,e2_eski,e1us_eski,e2us_eski,eps1_eski,eps2_eski;
```

```

double m_gecici,v_gecici1[4000],v_gecici2[4000],g_gecici1[4000],g_gecici2[4000];
double tau11[4000],tau22[4000],q1[4000],q2[4000];
double tau1[4000],tau2[4000],r1[4000],r2[4000],r3[4000],r4[4000],r2r2[4000],r4r4[4000];
double giris[6];//,q1cap,q1uscap,q2cap,q2uscap;
double M[2][2],inv_M[2][2],qd1us_yeni,qd2us_yeni;
double qd1us[4000],qd2us[4000],qd1us_eski2,qd2us_eski2,qd1uss[4000],qd2uss[4000];
double q1_yeni,q2_yeni,q1us_yeni,q2us_yeni,q1uss_yeni,q2uss_yeni;
double e1_yeni,e2_yeni,e1us_yeni,e2us_yeni,eps1_yeni,eps2_yeni;
double faults1[4000],faults2[4000],faults3[4000],faults4[4000],total[4];
double faults[4][4000],M00,M01,M10,M11;
int thr,faulty;
double q1cap[4001],q1uscap[4001],q2cap[4001],q2uscap[4001];

```

```
//ANFIS altprogramı
```

```
double ANFIS1(double ANFIS_giris[6],double a_ai22[6][2],double b_ai22[6][2],double
c_ai22[6][2],double pqr22[64][7])
```

```
{
int ii,jj;
/*double a_ai[6,2],b_ai[6,2],c_ai[6,2];*/
double mu_AB[6][2];
double gecici,gecici1,w[64],norm_w[64],L4[64];
double sum,total1;
//Girişler
//ANFIS_giris[0]://tau1(i+1)
//ANFIS_giris[1]://tau2(i+1)
//ANFIS_giris[2]://q1(i)
//ANFIS_giris[3]://q1us(i)
//ANFIS_giris[4]://q2(i)
//ANFIS_giris[5]://q2us(i)
sum=0;
total1=0;
// KAT 1
// mu kattaki üyelik işlevi
// tanımlamalara göre her katta 2 üyelik işlevi var.
// her işlev için
// ai:merkez
// bi:eğim
// ci:genişliği temsil etmektedir.
//6 giriş için tanımlama
for (ii=0;ii<6;ii++){
for (jj=0;jj<2;jj++){
gecici=(ANFIS_giris[ii]-c_ai22[ii][jj])/a_ai22[ii][jj];
if (gecici<0) gecici=-gecici;
mu_AB[ii][jj]=1/(1+pow(gecici,2*b_ai22[ii][jj]));
}
}
}

```

```
//KAT 2
```

```
//tanımlanan üyelik işlevleri birbiri ile çarpılır.
```

```
//Bu çarpımlara ateşleme gücü adı verilir ve kurallara göre tanımlanır.
```

```

w[0]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[1]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[2]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[3]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[4]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[5]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[6]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];

```



```

//Tanımlanan ateşleme güçleri normalize edilir.
    for (ii=0;ii<64;ii++)
        sum=sum+w[ii];
    for (ii=0;ii<64;ii++)
        norm_w[ii]=w[ii]/sum;

//KAT 4
//Normalize edilen ateşleme açıları birinci derece bir fonksiyon ile çarpılır.
    for (ii=0;ii<64;ii++){
        gecici1=0;
        for (jj=0;jj<6;jj++){
            gecici1=gecici1+pqr22[ii][jj]*ANFIS_giris[jj];
        }
        gecici1=gecici1+pqr22[ii][6];
        L4[ii]=norm_w[ii]*gecici1;
    }
//KAT 5
//Kat 4'ten gelen tüm değerler toplanır
    for (ii=0;ii<64;ii++)
        total1=total1+L4[ii];
    return total1;
}

void NNet1(double NNet_giris[6],double IW[10][6],double LW11[20][10],double LW22[20][20],double
LW33[4][20],double bias11[10],double bias22[20],double bias33[20],double bias44[4])
{
    int ii,jj;
    double n1[10],a1[10],n2[20],a2[20],n3[20],a3[20],n4[4],a4[4];
    for (ii=0;ii<10;ii++){n1[ii]=0;a1[ii]=0;}
    for (ii=0;ii<20;ii++){n2[ii]=0;a2[ii]=0;n3[ii]=0;a3[ii]=0;}
    for (ii=0;ii<4;ii++){n4[ii]=0;a4[ii]=0;}
    //GİRİŞ KATI
    for (ii=0;ii<10;ii++){
        for (jj=0;jj<6;jj++){
            n1[ii]=n1[ii]+NNet_giris[jj]*IW[ii][jj];
        }
        n1[ii]=n1[ii]+bias11[ii];
        a1[ii]=tanhf(n1[ii]);
    }

//1. KAT
    for (ii=0;ii<20;ii++){
        for (jj=0;jj<10;jj++){
            //Giriş katmanı için
            n2[ii]=n2[ii]+a1[jj]*LW11[ii][jj];
        }
        n2[ii]=n2[ii]+bias22[ii];
        a2[ii]=tanhf(n2[ii]);
    }

//2. KAT
    for (ii=0;ii<20;ii++){
        for (jj=0;jj<20;jj++){
            //Giriş katmanı için
            n3[ii]=n3[ii]+a2[jj]*LW22[ii][jj];
        }
        n3[ii]=n3[ii]+bias33[ii];
        a3[ii]=tanhf(n3[ii]);
    }
}

```

```

//3. KAT
for (ii=0;ii<4;ii++){
    for (jj=0;jj<20;jj++){
        //Giriş katmanı için
        n4[ii]=n4[ii]+a3[jj]*LW33[ii][jj];
    }
    n4[ii]=n4[ii]+bias44[ii];
    a4[ii]=tanhf(n4[ii]);
}

for(ii=0;ii<4;ii++){total[ii]=a4[ii];}
}

//ANA PROGRAM
void main()
{
int i;

for(i=0;i<4001;i++){
t[i]=0.01*(i+1);
qd1[i]=0.8*cos(t[i]/2);
qd2[i]=0.8*sin(t[i]/2);
}

q1[0]=0.1;
q2[0]=0;
q1[1]=0.1;
q2[1]=0;
q1[2]=0.1;
q2[2]=0;
q1us[0]=0;
q2us[0]=0;
q1us[1]=0;
q2us[1]=0;
q1us_eski=0;
q2us_eski=0;
e1_eski=qd1[0]-q1[0];
e2_eski=qd2[0]-q2[0];
e1us_eski=0;
e2us_eski=0;
eps1_eski=deltat*e1_eski;
eps2_eski=deltat*e2_eski;
for (i=0;i<=4000;i++){ r1[i]=0;r2[i]=0;r3[i]=0;r4[i]=0;r2r2[i]=0;r4r4[i]=0;}
for (i=0;i<=4000;i++){ faults[0][i]=0;faults[1][i]=0;faults[2][i]=0;faults[3][i]=0;
                        faults1[i]=0;faults2[i]=0;faults3[i]=0;faults4[i]=0;

v_gecici1[i]=0;v_gecici2[i]=0;g_gecici1[i]=0;g_gecici2[i]=0;

tau1[i]=0;tau2[i]=0;qd1us[i]=0;qd2us[i]=0;qd1uss[i]=0;qd2uss[i]=0;
                        q1cap[i]=0;q1uscap[i]=0;q2cap[i]=0;q2uscap[i]=0;}

faulty=0;thr=0;
for(i=1;i<=4000;i++){

    qd1us[i]=(qd1[i]-qd1[i-1])/deltat;
    qd2us[i]=(qd2[i]-qd2[i-1])/deltat;
    qd1uss[i]=(qd1us[i]-qd1us[i-1])/deltat;
    qd2uss[i]=(qd2us[i]-qd2us[i-1])/deltat;

```

```

    qd1us_yeni=(qd1[i+1]-qd1[i])/deltat;
    qd2us_yeni=(qd2[i+1]-qd2[i])/deltat;
}
for(i=1;i<=299;i++){

    M00=(m1+m2)*a1*a1+m2*a2*a2+2*m2*a1*a2*cos(q2[i]);
    M01=m2*a2*a2+m2*a1*a2*cos(q2[i]);
    M10=m2*a2*a2+m2*a1*a2*cos(q2[i]);
    M11=m2*a2*a2;
/*  M[0][0]=(m1+m2)*a1*a1+m2*a2*a2+2*m2*a1*a2*cos(q2[i]); */
/*  M[0][1]=m2*a2*a2+m2*a1*a2*cos(q2[i]); */
/*  M[1][0]=m2*a2*a2+m2*a1*a2*cos(q2[i]); */
/*  M[1][1]=m2*a2*a2; */

    v_gecici1[i]=-m2*a1*a2*(2*(q1us[i]*q2us[i]+q2us[i]*q2us[i])*sin(q2[i]);
    v_gecici2[i]=m2*a1*a2*q1us[i]*q1us[i]*sin(q2[i]);

    g_gecici1[i]=(m1+m2)*g*a1*cos(q1[i])+m2*g*a2*cos(q1[i]+q2[i]);
    g_gecici2[i]=m2*g*a2*cos(q1[i]+q2[i]);

tau1[i+1]=(M00*(qd1uss[i]+20*e1us_eski+100*e1_eski+500*eps1_eski)+M01*(qd2uss[i]+20*e2us_esk
i+100*e2_eski+500*eps2_eski))+v_gecici1[i]+g_gecici1[i];

tau2[i+1]=(M10*(qd1uss[i]+20*e1us_eski+100*e1_eski+500*eps1_eski)+M11*(qd2uss[i]+20*e2us_esk
i+100*e2_eski+500*eps2_eski))+v_gecici2[i]+g_gecici2[i];

    tau11[i+1]=tau1[i+1];
    tau22[i+1]=tau2[i+1];
    M[0][0]=M00*(1+0.05*sin(t[i]));
    M[0][1]=M01*(1+0.05*sin(t[i]));
    M[1][0]=M10*(1+0.05*sin(t[i]));
    M[1][1]=M11*(1+0.05*sin(t[i]));
    v_gecici1[i]=v_gecici1[i]*(1+0.05*sin(t[i]));
    v_gecici2[i]=v_gecici2[i]*(1+0.05*sin(t[i]));
    /*V[1]=V[1]*(1+0.05*sin(t[i]));*/
    g_gecici1[i]=g_gecici1[i]*(1+0.05*sin(t[i]));
    g_gecici2[i]=g_gecici2[i]*(1+0.05*sin(t[i]));

    inv_M[0][0]=M[1][1]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
    inv_M[0][1]=-M[0][1]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
    inv_M[1][0]=-M[1][0]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
    inv_M[1][1]=M[0][0]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);

    q1uss_yeni=inv_M[0][0]*(tau11[i+1]-v_gecici1[i]-g_gecici1[i])+inv_M[0][1]*(tau22[i+1]-
v_gecici2[i]-g_gecici2[i]);
    q2uss_yeni=inv_M[1][0]*(tau11[i+1]-v_gecici1[i]-g_gecici1[i])+inv_M[1][1]*(tau22[i+1]-
v_gecici2[i]-g_gecici2[i]);
    q1us[i+1]=q1us[i]+deltat*q1uss_yeni;
    q2us[i+1]=q2us[i]+deltat*q2uss_yeni;
    q1[i+1]=q1[i]+deltat*q1us[i+1];
    q2[i+1]=q2[i]+deltat*q2us[i+1];

    e1_eski=qd1[i+1]-q1[i+1];
    e2_eski=qd2[i+1]-q2[i+1];
    e1us_eski=qd1us[i]-q1us[i];

```

```

e2us_eski=qd2us[i]-q2us[i];
eps1_eski=eps1_eski+deltat*e1_eski;
eps2_eski=eps2_eski+deltat*e2_eski;
/* Comp_torque_PID_ANFIS*/
/*q1_eski=q1_yeni;
q2_eski=q2_yeni;*/
giris[0]=tau1[i+1];giris[1]=tau2[i+1];
giris[2]=q1[i];giris[3]=q1us[i];
giris[4]=q2[i];giris[5]=q2us[i];
q1cap[i]=0;
r1[i]=0;

q1uscap[i]=0;
r2[i]=0;

q2cap[i]=0;
r3[i]=0;

q2uscap[i]=0;
r4[i]=0;

r2r2[i]=0;
r4r4[i]=0;

q1uss_eski=q1uss_yeni;
q2uss_eski=q2uss_yeni;

q1cap[i]=ANFIS1(giris,a_ai1,b_ai1,c_ai1,pqr1);
r1[i]=q1[i+1]-q1cap[i];

q1uscap[i]=ANFIS1(giris,a_ai2,b_ai2,c_ai2,pqr2);
r2[i]=q1us[i+1]-q1uscap[i];

q2cap[i]=ANFIS1(giris,a_ai3,b_ai3,c_ai3,pqr3);
r3[i]=q2[i+1]-q2cap[i];

q2uscap[i]=ANFIS1(giris,a_ai4,b_ai4,c_ai4,pqr4);
r4[i]=q2us[i+1]-q2uscap[i];

r2r2[i]=(r1[i]-r1[i-1])/deltat;
r4r4[i]=(r3[i]-r3[i-1])/deltat;
}
for(i=300;i<=2399;i++){

M00=(m1+m2)*a1*a1+m2*a2*a2+2*m2*a1*a2*cos(q2[i]);
M01=m2*a2*a2+m2*a1*a2*cos(q2[i]);
M10=m2*a2*a2+m2*a1*a2*cos(q2[i]);
M11=m2*a2*a2;
/* M[0][0]=(m1+m2)*a1*a1+m2*a2*a2+2*m2*a1*a2*cos(q2[i]); */
/* M[0][1]=m2*a2*a2+m2*a1*a2*cos(q2[i]); */
/* M[1][0]=m2*a2*a2+m2*a1*a2*cos(q2[i]); */
/* M[1][1]=m2*a2*a2; */

v_gecici1[i]=-m2*a1*a2*(2*(q1us[i]*q2us[i]+q2us[i]*q2us[i])*sin(q2[i]));
v_gecici2[i]=m2*a1*a2*q1us[i]*q1us[i]*sin(q2[i]);

g_gecici1[i]=(m1+m2)*g*a1*cos(q1[i])+m2*g*a2*cos(q1[i]+q2[i]);
g_gecici2[i]=m2*g*a2*cos(q1[i]+q2[i]);

```

```
tau1[i+1]=(M00*(qd1uss[i]+20*e1us_eski+100*e1_eski+500*eps1_eski)+M01*(qd2uss[i]+20*e2us_esk
i+100*e2_eski+500*eps2_eski))+v_gecici1[i]+g_gecici1[i];
```

```
tau2[i+1]=(M10*(qd1uss[i]+20*e1us_eski+100*e1_eski+500*eps1_eski)+M11*(qd2uss[i]+20*e2us_esk
i+100*e2_eski+500*eps2_eski))+v_gecici2[i]+g_gecici2[i];
```

```
tau11[i+1]=tau1[i+1];
tau22[i+1]=tau2[i+1];
M[0][0]=M00*(1+0.05*sin(t[i]));
M[0][1]=M01*(1+0.05*sin(t[i]));
M[1][0]=M10*(1+0.05*sin(t[i]));
M[1][1]=M11*(1+0.05*sin(t[i]));
v_gecici1[i]=v_gecici1[i]*(1+0.05*sin(t[i]));
v_gecici2[i]=v_gecici2[i]*(1+0.05*sin(t[i]));
/*V[1]=V[1]*(1+0.05*sin(t[i]));*/
g_gecici1[i]=g_gecici1[i]*(1+0.05*sin(t[i]));
g_gecici2[i]=g_gecici2[i]*(1+0.05*sin(t[i]));
```

```
inv_M[0][0]=M[1][1]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
inv_M[0][1]=-M[0][1]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
inv_M[1][0]=-M[1][0]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
inv_M[1][1]=M[0][0]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
```

```
q1uss_yeni=inv_M[0][0]*(tau11[i+1]-v_gecici1[i]-g_gecici1[i])+inv_M[0][1]*(tau22[i+1]-
v_gecici2[i]-g_gecici2[i]);
```

```
q2uss_yeni=inv_M[1][0]*(tau11[i+1]-v_gecici1[i]-g_gecici1[i])+inv_M[1][1]*(tau22[i+1]-
v_gecici2[i]-g_gecici2[i]);
```

```
q1us[i+1]=q1us[i]+deltat*q1uss_yeni;
q2us[i+1]=q2us[i]+deltat*q2uss_yeni;
q1[i+1]=q1[i]+deltat*q1us[i+1];
q2[i+1]=q2[i]+deltat*q2us[i+1];
```

```
e1_eski=qd1[i+1]-q1[i+1];
e2_eski=qd2[i+1]-q2[i+1];
e1us_eski=qd1us[i]-q1us[i];
e2us_eski=qd2us[i]-q2us[i];
eps1_eski=eps1_eski+deltat*e1_eski;
eps2_eski=eps2_eski+deltat*e2_eski;
```

```
/* Comp_torque_PID_ANFIS*/
/*q1_eski=q1_yeni;
q2_eski=q2_yeni;*/
```

```
giris[0]=tau1[i+1];giris[1]=tau2[i+1];
giris[2]=q1[i];giris[3]=q1us[i];
giris[4]=q2[i];giris[5]=q2us[i];
q1cap[i]=0;
r1[i]=0;
```

```
q1uscap[i]=0;
r2[i]=0;
```

```
q2cap[i]=0;
r3[i]=0;
```

```
q2uscap[i]=0;
r4[i]=0;
```

```
r2r2[i]=0;
r4r4[i]=0;
```

```

q1uss_eski=q1uss_yeni;
q2uss_eski=q2uss_yeni;

q1cap[i]=ANFIS1(giris,a_ai1,b_ai1,c_ai1,pqr1);
r1[i]=q1[i+1]-q1cap[i];

q1uscap[i]=ANFIS1(giris,a_ai2,b_ai2,c_ai2,pqr2);
r2[i]=q1us[i+1]-q1uscap[i];

q2cap[i]=ANFIS1(giris,a_ai3,b_ai3,c_ai3,pqr3);
r3[i]=q2[i+1]-q2cap[i];

q2uscap[i]=ANFIS1(giris,a_ai4,b_ai4,c_ai4,pqr4);
r4[i]=q2us[i+1]-q2uscap[i];

r2r2[i]=(r1[i]-r1[i-1])/deltat;
r4r4[i]=(r3[i]-r3[i-1])/deltat;
if ((r1[i]>0.009|r1[i]<-0.0105)|(r2[i]>0.0025|r2[i]<-0.0035)|(r3[i]>0.015|r3[i]<-
0.01)|(r4[i]>0.0025|r4[i]<-0.0011)|(r2r2[i]>0.015|r2r2[i]<-0.015)|(r4r4[i]>0.02|r4r4[i]<-0.02))
{
    if (faulty==0)    faulty=i;
                    if (i>faulty+350){

                        giris[0]=r1[i];
                        giris[1]=r2[i];
                        giris[2]=r3[i];
                        giris[3]=r4[i];
                        giris[4]=r2r2[i];
                        giris[5]=r4r4[i];
                        NNet1(giris,IW1,LW1,LW2,LW3,bias1,bias2,bias3,bias4);
                        faults[0][i]=total[0];
                        faults[1][i]=total[1];
                        faults[2][i]=total[2];
                        faults[3][i]=total[3];
                        thr=thr+1;
                        if (i>faulty+420){
                            if (((faults[0][i-69]>0.5)&&(faults[0][i-
34]>0.5)&&(faults[0][i]>=0))||((faults[0][i-69]>0.5)&&(faults[0][i-
34]>=0)&&(faults[0][i]>0.5))||((faults[0][i-69]>=0)&&(faults[0][i-34]>0.5)&&(faults[0][i]>0.5)))){
                                faults1[i]=0.9;
                            }
                        }
                        else {
                            faults1[i]=0.1;
                        }
                        if (((faults[1][i-69]>0.5)&&(faults[1][i-
34]>0.5)&&(faults[1][i]>=0))||((faults[1][i-69]>0.5)&&(faults[1][i-
34]>=0)&&(faults[1][i]>0.5))||((faults[1][i-69]>=0)&&(faults[1][i-34]>0.5)&&(faults[1][i]>0.5)))){
                                faults2[i]=0.9;
                            }
                        }
                        else {
                            faults2[i]=0.1;
                        }
                        if (((faults[2][i-69]>0.5)&&(faults[2][i-
34]>0.5)&&(faults[2][i]>=0))||((faults[2][i-69]>0.5)&&(faults[2][i-
34]>=0)&&(faults[2][i]>0.5))||((faults[2][i-69]>=0)&&(faults[2][i-34]>0.5)&&(faults[2][i]>0.5)))){
                                faults3[i]=0.9;
                            }
                        }
                    }
}

```

```

        else {
            faults3[i]=0.1;
        }
        if (((faults3[i-69]>0.5)&&(faults3[i-34]>0.5)&&(faults3[i]>0))||((faults3[i-69]>0.5)&&(faults3[i-34]>=0)&&(faults3[i]>0.5))||((faults3[i-69]>=0)&&(faults3[i-34]>0.5)&&(faults3[i]>0.5))) {
            faults4[i]=0.9;
        }
        else {
            faults4[i]=0.1;
        }
        //faults1[i]=total[0];
        //faults2[i]=total[1];
        //faults3[i]=total[2];
        //faults4[i]=total[3];
    }
}
else {
    faults1[i]=0;
    faults2[i]=0;
    faults3[i]=0;
    faults4[i]=0;
}
}

for(i=2400;i<4000;i++){
/*  qd1us_eski=(qd1[i]-qd1[i-1])/deltat;    */
/*  qd2us_eski=(qd2[i]-qd2[i-1])/deltat;    */
/*  qd1us_eski2=(qd1[i-1]-qd1[i-2])/deltat;    */
/*  qd2us_eski2=(qd2[i-1]-qd2[i-2])/deltat;    */
/*  qd1uss_eski=(qd1us_eski-qd1us_eski2)/deltat; */
/*  qd2uss_eski=(qd2us_eski-qd2us_eski2)/deltat; */
/*  qd1us_yeni=(qd1[i+1]-qd1[i])/deltat;    */
/*  qd2us_yeni=(qd2[i+1]-qd2[i])/deltat;    */

    M00=(m1+m2)*a1*a1+m2*a2*a2+2*m2*a1*a2*cos(q2[i]);
    M01=m2*a2*a2+m2*a1*a2*cos(q2[i]);
    M10=m2*a2*a2+m2*a1*a2*cos(q2[i]);
    M11=m2*a2*a2;
/*  M[0][0]=(m1+m2)*a1*a1+m2*a2*a2+2*m2*a1*a2*cos(q2[i]); */
/*  M[0][1]=m2*a2*a2+m2*a1*a2*cos(q2[i]);    */
/*  M[1][0]=m2*a2*a2+m2*a1*a2*cos(q2[i]);    */
/*  M[1][1]=m2*a2*a2;                        */

    v_gecici1[i]=-m2*a1*a2*(2*(q1us[i]*q2us[i]+q2us[i]*q2us[i])*sin(q2[i]);
    v_gecici2[i]=m2*a1*a2*q1us[i]*q1us[i]*sin(q2[i]);

    g_gecici1[i]=(m1+m2)*g*a1*cos(q1[i])+m2*g*a2*cos(q1[i]+q2[i]);
    g_gecici2[i]=m2*g*a2*cos(q1[i]+q2[i]);

    tau1[i+1]=(M00*(qd1uss[i]+20*e1us_eski+100*e1_eski+500*eps1_eski)+M01*(qd2uss[i]+20*e2us_eski+100*e2_eski+500*eps2_eski))+v_gecici1[i]+g_gecici1[i];

    tau2[i+1]=(M10*(qd1uss[i]+20*e1us_eski+100*e1_eski+500*eps1_eski)+M11*(qd2uss[i]+20*e2us_eski+100*e2_eski+500*eps2_eski))+v_gecici2[i]+g_gecici2[i];
}

```

```

M[0][0]=M00*(1+0.05*sin(t[i]));
M[0][1]=M01*(1+0.05*sin(t[i]));
M[1][0]=M10*(1+0.05*sin(t[i]));
M[1][1]=M11*(1+0.05*sin(t[i]));
v_gecici1[i]=v_gecici1[i]*(1+0.05*sin(t[i]));
v_gecici2[i]=v_gecici2[i]*(1+0.05*sin(t[i]));
/*V[1]=V[1]*(1+0.05*sin(t[i]));*/
g_gecici1[i]=g_gecici1[i]*(1+0.05*sin(t[i]));
g_gecici2[i]=g_gecici2[i]*(1+0.05*sin(t[i]));

inv_M[0][0]=M[1][1]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
inv_M[0][1]=-M[0][1]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
inv_M[1][0]=-M[1][0]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);
inv_M[1][1]=M[0][0]/(M[0][0]*M[1][1]-M[0][1]*M[1][0]);

/*ARIZA İŞLEVI*/ /*tau11 arızalı tork değeri tau1 uygulanmaya çalışılan tork değeri*/
tau11[i+1]=tau1[i+1]*0.5;
tau22[i+1]=tau2[i+1]*1;

q1uss_yeni=inv_M[0][0]*(tau11[i+1]-v_gecici1[i]-g_gecici1[i])+inv_M[0][1]*(tau22[i+1]-
v_gecici2[i]-g_gecici2[i]);
q2uss_yeni=inv_M[1][0]*(tau11[i+1]-v_gecici1[i]-g_gecici1[i])+inv_M[1][1]*(tau22[i+1]-
v_gecici2[i]-g_gecici2[i]);
q1us[i+1]=q1us[i]+deltat*q1uss_yeni;
q2us[i+1]=q2us[i]+deltat*q2uss_yeni;
q1[i+1]=q1[i]+deltat*q1us[i+1];
q2[i+1]=q2[i]+deltat*q2us[i+1];

e1_eski=qd1[i+1]-q1[i+1];
e2_eski=qd2[i+1]-q2[i+1];
e1us_eski=qd1us[i]-q1us[i];
e2us_eski=qd2us[i]-q2us[i];
eps1_eski=eps1_eski+deltat*e1_eski;
eps2_eski=eps2_eski+deltat*e2_eski;
/* Comp_torque_PID_ANFIS*/
/*q1_eski=q1_yeni;
q2_eski=q2_yeni;*/
giris[0]=tau1[i+1];giris[1]=tau2[i+1];
giris[2]=q1[i];giris[3]=q1us[i];
giris[4]=q2[i];giris[5]=q2us[i];
q1cap[i]=0;
r1[i]=0;

q1uscap[i]=0;
r2[i]=0;

q2cap[i]=0;
r3[i]=0;

q2uscap[i]=0;
r4[i]=0;

r2r2[i]=0;
r4r4[i]=0;
q1cap[i]=ANFIS1(giris,a_ai1,b_ai1,c_ai1,pqr1);
r1[i]=q1[i+1]-q1cap[i];

q1uscap[i]=ANFIS1(giris,a_ai2,b_ai2,c_ai2,pqr2);

```

```

r2[i]=q1us[i+1]-q1uscap[i];

q2cap[i]=ANFIS1(giris,a_ai3,b_ai3,c_ai3,pqr3);
r3[i]=q2[i+1]-q2cap[i];

q2uscap[i]=ANFIS1(giris,a_ai4,b_ai4,c_ai4,pqr4);
r4[i]=q2us[i+1]-q2uscap[i];

r2r2[i]=(r1[i]-r1[i-1])/deltat;
r4r4[i]=(r3[i]-r3[i-1])/deltat;

// NN çalıştırma koşulları
if ((r1[i]>0.009|r1[i]<-0.0105)|(r2[i]>0.0025|r2[i]<-0.0035)|(r3[i]>0.015|r3[i]<-
0.01)|(r4[i]>0.0025|r4[i]<-0.0011)|(r2r2[i]>0.015|r2r2[i]<-0.015)|(r4r4[i]>0.02|r4r4[i]<-0.02))

{
    if (faulty==0)    faulty=i;
                    if (i>faulty+350){

                        giris[0]=r1[i];
                        giris[1]=r2[i];
                        giris[2]=r3[i];
                        giris[3]=r4[i];
                        giris[4]=r2r2[i];
                        giris[5]=r4r4[i];
                        NNet1(giris,IW1,LW1,LW2,LW3,bias1,bias2,bias3,bias4);
                        faults[0][i]=total[0];
                        faults[1][i]=total[1];
                        faults[2][i]=total[2];
                        faults[3][i]=total[3];
                        thr=thr+1;
                        if (i>faulty+420){
                            if (((faults[0][i-69]>0.5)&&(faults[0][i-
34]>0.5)&&(faults[0][i]>=0))||((faults[0][i-69]>0.5)&&(faults[0][i-
34]>=0)&&(faults[0][i]>0.5))||((faults[0][i-69]>=0)&&(faults[0][i-34]>0.5)&&(faults[0][i]>0.5)))){
                                faults1[i]=0.9;
                            }
                            else {
                                faults1[i]=0.1;
                            }
                            if (((faults[1][i-69]>0.5)&&(faults[1][i-
34]>0.5)&&(faults[1][i]>=0))||((faults[1][i-69]>0.5)&&(faults[1][i-
34]>=0)&&(faults[1][i]>0.5))||((faults[1][i-69]>=0)&&(faults[1][i-34]>0.5)&&(faults[1][i]>0.5)))){
                                faults2[i]=0.9;
                            }
                            else {
                                faults2[i]=0.1;
                            }
                            if (((faults[2][i-69]>0.5)&&(faults[2][i-
34]>0.5)&&(faults[2][i]>=0))||((faults[2][i-69]>0.5)&&(faults[2][i-
34]>=0)&&(faults[2][i]>0.5))||((faults[2][i-69]>=0)&&(faults[2][i-34]>0.5)&&(faults[2][i]>0.5)))){
                                faults3[i]=0.9;
                            }
                            else {
                                faults3[i]=0.1;
                            }
                            if (((faults[3][i-69]>0.5)&&(faults[3][i-
34]>0.5)&&(faults[3][i]>0))||((faults[3][i-69]>0.5)&&(faults[3][i-
34]>=0)&&(faults[3][i]>0.5))||((faults[3][i-69]>=0)&&(faults[3][i-34]>0.5)&&(faults[3][i]>0.5)))){

```

```

                                                                    faults4[i]=0.9;
    }
    else {
    faults4[i]=0.1;
    }
                                                                    //faults1[i]=total[0];
                                                                    //faults2[i]=total[1];
                                                                    //faults3[i]=total[2];
                                                                    //faults4[i]=total[3];
                                                                    }
    }
else {
                                                                    faults1[i]=0;
                                                                    faults2[i]=0;
                                                                    faults3[i]=0;
                                                                    faults4[i]=0;
    }
    q1uss_eski=q1uss_yeni;
    q2uss_eski=q2uss_yeni;
}
                                                                    //end of while (1) infinite loop
} //end of main
```

## Ek H. 2. Şema İçin MATLAB-DSP Programları

### DSP Programı

D:\c6713\_book\_cd\digital\_signal\rtdx\_matlab\_sim\RTDX\_MATLAB\_sim.pjt

D:\c6713\projects\sine\_floating\Sine\_floating\_ariza\_GOS\_RTDX\_double.c

//

```
#include <math.h>
#include <stdio.h>
#include "C:\CCStudio_v3.1\C6000\rtdx\include\rtdx.h"
#include "C:\CCStudio_v3.1\C6000\cgtools\include\c6x.h" //RTDX support file
#include "C:\CCStudio_v3.1\examples\disk6713\rtdx\shared\target.h"
#define PI 3.14159265358
#pragma DATA_SECTION(tau1, ".EXT_RAM")
#pragma DATA_SECTION(tau2, ".EXT_RAM")
#pragma DATA_SECTION(qd1, ".EXT_RAM")
#pragma DATA_SECTION(qd2, ".EXT_RAM")
#pragma DATA_SECTION(r1, ".EXT_RAM")
#pragma DATA_SECTION(r2, ".EXT_RAM")
#pragma DATA_SECTION(r3, ".EXT_RAM")
#pragma DATA_SECTION(r4, ".EXT_RAM")
#pragma DATA_SECTION(v_gecici1, ".EXT_RAM")
#pragma DATA_SECTION(v_gecici2, ".EXT_RAM")
#pragma DATA_SECTION(g_gecici1, ".EXT_RAM")
#pragma DATA_SECTION(g_gecici2, ".EXT_RAM")
#pragma DATA_SECTION(qd1us, ".EXT_RAM")
#pragma DATA_SECTION(qd2us, ".EXT_RAM")
#pragma DATA_SECTION(qd1uss, ".EXT_RAM")
#pragma DATA_SECTION(qd1uss, ".EXT_RAM")
#pragma DATA_SECTION(q1us, ".EXT_RAM")
#pragma DATA_SECTION(q2us, ".EXT_RAM")
#pragma DATA_SECTION(q1cap, ".EXT_RAM")
#pragma DATA_SECTION(q1uscap, ".EXT_RAM")
#pragma DATA_SECTION(q2cap, ".EXT_RAM")
#pragma DATA_SECTION(q2uscap, ".EXT_RAM")
/*tau105 arızalı sistem*/
#pragma DATA_SECTION(q1cap_tau105, ".EXT_RAM")
#pragma DATA_SECTION(q1uscap_tau105, ".EXT_RAM")
#pragma DATA_SECTION(q2cap_tau105, ".EXT_RAM")
#pragma DATA_SECTION(q2uscap_tau105, ".EXT_RAM")
#pragma DATA_SECTION(r1_tau105, ".EXT_RAM")
#pragma DATA_SECTION(r2_tau105, ".EXT_RAM")
#pragma DATA_SECTION(r3_tau105, ".EXT_RAM")
#pragma DATA_SECTION(r4_tau105, ".EXT_RAM")
/*tau107 arızalı sistem*/
#pragma DATA_SECTION(q1cap_tau107, ".EXT_RAM")
#pragma DATA_SECTION(q1uscap_tau107, ".EXT_RAM")
#pragma DATA_SECTION(q2cap_tau107, ".EXT_RAM")
#pragma DATA_SECTION(q2uscap_tau107, ".EXT_RAM")
#pragma DATA_SECTION(r1_tau107, ".EXT_RAM")
#pragma DATA_SECTION(r2_tau107, ".EXT_RAM")
#pragma DATA_SECTION(r3_tau107, ".EXT_RAM")
#pragma DATA_SECTION(r4_tau107, ".EXT_RAM")
/*tau205 arızalı sistem*/
#pragma DATA_SECTION(q1cap_tau205, ".EXT_RAM")
#pragma DATA_SECTION(q1uscap_tau205, ".EXT_RAM")
#pragma DATA_SECTION(q2cap_tau205, ".EXT_RAM")
#pragma DATA_SECTION(q2uscap_tau205, ".EXT_RAM")
```

```

#pragma DATA_SECTION(r1_tau205,".EXT_RAM")
#pragma DATA_SECTION(r2_tau205,".EXT_RAM")
#pragma DATA_SECTION(r3_tau205,".EXT_RAM")
#pragma DATA_SECTION(r4_tau205,".EXT_RAM")
/*tau207 arızalı sistem*/
#pragma DATA_SECTION(q1cap_tau207,".EXT_RAM")
#pragma DATA_SECTION(q1uscap_tau207,".EXT_RAM")
#pragma DATA_SECTION(q2cap_tau207,".EXT_RAM")
#pragma DATA_SECTION(q2uscap_tau207,".EXT_RAM")
#pragma DATA_SECTION(r1_tau207,".EXT_RAM")
#pragma DATA_SECTION(r2_tau207,".EXT_RAM")
#pragma DATA_SECTION(r3_tau207,".EXT_RAM")
#pragma DATA_SECTION(r4_tau207,".EXT_RAM")

#pragma DATA_SECTION(f1_i,".EXT_RAM")
#pragma DATA_SECTION(f2_i,".EXT_RAM")
#pragma DATA_SECTION(faults_0,".EXT_RAM")
#pragma DATA_SECTION(faults_1,".EXT_RAM")
#pragma DATA_SECTION(faults_2,".EXT_RAM")
#pragma DATA_SECTION(faults_3,".EXT_RAM")
#pragma DATA_SECTION(faults_4,".EXT_RAM")
#pragma DATA_SECTION(faults2_0,".EXT_RAM")
#pragma DATA_SECTION(faults2_1,".EXT_RAM")
#pragma DATA_SECTION(faults2_2,".EXT_RAM")
#pragma DATA_SECTION(faults2_3,".EXT_RAM")
#pragma DATA_SECTION(faults2_4,".EXT_RAM")

/* #pragma DATA_SECTION(q1cap,"SDRAM") */
#include "q1cap_coef_double.h" //q1cap ANFIS kaysayıları
#include "q1uscap_coef_double.h" //q1uscap ANFIS kaysayıları
#include "q2cap_coef_double.h" //q2cap ANFIS kaysayıları
#include "q2uscap_coef_double.h" //q2uscap ANFIS kaysayıları

#include "q1cap_tau105_coef_double.h" //q1cap ANFIS kaysayıları
#include "q1uscap_tau105_coef_double.h" //q1uscap ANFIS kaysayıları
#include "q2cap_tau105_coef_double.h" //q2cap ANFIS kaysayıları
#include "q2uscap_tau105_coef_double.h" //q2uscap ANFIS kaysayıları

#include "q1cap_tau107_coef_double.h" //q1cap ANFIS kaysayıları
#include "q1uscap_tau107_coef_double.h" //q1uscap ANFIS kaysayıları
#include "q2cap_tau107_coef_double.h" //q2cap ANFIS kaysayıları
#include "q2uscap_tau107_coef_double.h" //q2uscap ANFIS kaysayıları

#include "q1cap_tau205_coef_double.h" //q1cap ANFIS kaysayıları
#include "q1uscap_tau205_coef_double.h" //q1uscap ANFIS kaysayıları
#include "q2cap_tau205_coef_double.h" //q2cap ANFIS kaysayıları
#include "q2uscap_tau205_coef_double.h" //q2uscap ANFIS kaysayıları

#include "q1cap_tau207_coef_double.h" //q1cap ANFIS kaysayıları
#include "q1uscap_tau207_coef_double.h" //q1uscap ANFIS kaysayıları
#include "q2cap_tau207_coef_double.h" //q2cap ANFIS kaysayıları
#include "q2uscap_tau207_coef_double.h" //q2uscap ANFIS kaysayıları

double ANFIS1(double ANFIS_giris[6],double a_ai22[6][2],double b_bi22[6][2],double
c_ci22[6][2],double pqr22[64][7]);
RTDX_CreateInputChannel(ichan); //create input channel-data transfer PC-->DSK
RTDX_CreateOutputChannel(ochan);

```

```

double deltat=0.01,a1=1,a2=1,m1=1,m2=1,g=9.81;
double qd1[4001],qd2[4001],t[4001];
double q1us_eksi,q2us_eksi,q1uss_eksi,q2uss_eksi;
double q1us[4001],q2us[4001];
double e1_eksi,e2_eksi,e1us_eksi,e2us_eksi,eps1_eksi,eps2_eksi;
double m_gecici,v_gecici1[4000],v_gecici2[4000],g_gecici1[4000],g_gecici2[4000];
double tau11[4000],tau22[4000],q1[4000],q2[4000];
double tau1[4000],tau2[4000],r1[4000],r2[4000],r3[4000],r4[4000];
double M[2][2],inv_M[2][2],qd1us_yeni,qd2us_yeni;
double qd1us[4000],qd2us[4000],qd1us_eksi2,qd2us_eksi2,qd1uss[4000],qd2uss[4000];
double q1_yeni,q2_yeni,q1us_yeni,q2us_yeni,q1uss_yeni,q2uss_yeni;
double e1_yeni,e2_yeni,e1us_yeni,e2us_yeni,eps1_yeni,eps2_yeni;
double total[4],M00,M01,M10,M11;
int thr,faulty;
double q1cap[4001],q1uscap[4001],q2cap[4001],q2uscap[4001];
double q1cap_tau105[1],q1uscap_tau105[1],q2cap_tau105[1],q2uscap_tau105[1];
double q1cap_tau107[1],q1uscap_tau107[1],q2cap_tau107[1],q2uscap_tau107[1];
double q1cap_tau205[1],q1uscap_tau205[1],q2cap_tau205[1],q2uscap_tau205[1];
double q1cap_tau207[1],q1uscap_tau207[1],q2cap_tau207[1],q2uscap_tau207[1];
double r1_tau105[4000],r2_tau105[4000],r3_tau105[4000],r4_tau105[4000];
double r1_tau107[4000],r2_tau107[4000],r3_tau107[4000],r4_tau107[4000];
double r1_tau205[4000],r2_tau205[4000],r3_tau205[4000],r4_tau205[4000];
double r1_tau207[4000],r2_tau207[4000],r3_tau207[4000],r4_tau207[4000];
double f1_i[4000],f2_i[4000],f1[5];
double faults_0[4000],faults_1[4000],faults_2[4000],faults_3[4000],faults_4[4000];
double faults2_0[4000],faults2_1[4000],faults2_2[4000],faults2_3[4000],faults2_4[4000];
double f2,f3,f4,f5,f6;
double buffer[11],giris[6];//,q1cap,q1uscap,q2cap,q2uscap;

```

```

//ANFIS altprogramı
double ANFIS1(double ANFIS_giris[6],double a_ai22[6][2],double b_ai22[6][2],double
c_ai22[6][2],double pqr22[64][7])
{
int ii,jj;
/*double a_ai[6,2],b_ai[6,2],c_ai[6,2];*/
double mu_AB[6][2];
double gecici,gecici1,w[64],norm_w[64],L4[64];
double sum,total1;
//Girişler
//ANFIS_giris[0]://tau1(i+1)
//ANFIS_giris[1]://tau2(i+1)
//ANFIS_giris[2]://q1(i)
//ANFIS_giris[3]://q1us(i)
//ANFIS_giris[4]://q2(i)
//ANFIS_giris[5]://q2us(i)
sum=0;
total1=0;
// KAT 1
// mu kattaki üyelik işlevi
// tanımlamalara göre her katta 2 üyelik işlevi var.
// her işlev için
// ai:merkez
// bi:eğim
// ci:genişliği temsil etmektedir.
//6 giriş için tanımlama
for (ii=0;ii<6;ii++){

```

```

for (jj=0;jj<2;jj++){
gecici=(ANFIS_giris[ii]-c_ai22[ii][jj])/a_ai22[ii][jj];
if (gecici<0) gecici=-gecici;
mu_AB[ii][jj]=1/(1+pow(gecici,2*b_ai22[ii][jj]));
}
}

```

//KAT 2

//tanımlanan üyelik işlevleri birbiri ile çarpılır.

//Bu çarpımlara ateşleme gücü adı verilir ve kurallara göre tanımlanır.

```

w[0]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[1]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[2]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[3]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[4]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[5]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[6]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[7]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];
w[8]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[9]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[10]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[11]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[12]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[13]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[14]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[15]=mu_AB[0][0]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];
w[16]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[17]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[18]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[19]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[20]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[21]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[22]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[23]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];
w[24]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[25]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[26]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[27]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[28]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[29]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[30]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[31]=mu_AB[0][0]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];
w[32]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[33]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[34]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[35]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[36]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[37]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[38]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[39]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];
w[40]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[41]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[42]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[43]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[44]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[45]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[46]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[47]=mu_AB[0][1]*mu_AB[1][0]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];

```

```

w[48]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[49]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[50]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[51]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[52]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[53]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[54]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[55]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];
w[56]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[57]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[58]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[59]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[60]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[61]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[62]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[63]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];

```

```
//KAT 3
```

```
//Tanımlanan ateşleme güçleri normalize edilir.
```

```

for (ii=0;ii<64;ii++)
sum=sum+w[ii];
for (ii=0;ii<64;ii++)
norm_w[ii]=w[ii]/sum;

```

```
//KAT 4
```

```
//Normalize edilen ateşleme açıları birinci derece bir fonksiyon ile çarpılır.
```

```

for (ii=0;ii<64;ii++){
gecici1=0;
for (jj=0;jj<6;jj++){
gecici1=gecici1+pqr22[ii][jj]*ANFIS_giris[jj];
}
gecici1=gecici1+pqr22[ii][6];
L4[ii]=norm_w[ii]*gecici1;
}

```

```
//KAT 5
```

```
//Kat 4'ten gelen tüm değerler toplanır
```

```

for (ii=0;ii<64;ii++)
total1=total1+L4[ii];
return total1;
}

```

```
//ANA PROGRAM
```

```
void main()
```

```
{
```

```
int i=0,j;
```

```
double q1_i_1,q1us_i_1,q2_i_1,q2us_i_1;
```

```
TARGET_INITIALIZE();
```

```
for (j=0;j<=4000;j++){ r1[j]=0;r2[j]=0;r3[j]=0;r4[j]=0;
```

```

}
for (j=0;j<=4000;j++){ faults_0[j]=0;faults_1[j]=0;faults_2[j]=0;faults_3[j]=0;faults_4[j]=0;
q1cap[j]=0;q1uscap[j]=0;q2cap[j]=0;q2uscap[j]=0;
}

```

```
for (j=1;j<=11;j++){
```

```
giris[j]=0;
```

```
}
```

```
q1cap_tau105[0]=0;q1uscap_tau105[0]=0;q2cap_tau105[0]=0;q2uscap_tau105[0]=0;
```

```

q1cap_tau107[0]=0;q1uscap_tau107[0]=0;q2cap_tau107[0]=0;q2uscap_tau107[0]=0;
q1cap_tau205[0]=0;q1uscap_tau205[0]=0;q2cap_tau205[0]=0;q2uscap_tau205[0]=0;
q1cap_tau207[0]=0;q1uscap_tau207[0]=0;q2cap_tau207[0]=0;q2uscap_tau207[0]=0;

```

```
f2=0;f3=0;f4=0;f5=0;f6=0;
```

```
faulty=0;thr=0;
```

```

while(!RTDX_isInputEnabled(&ichan))
    puts("\n\n Waiting to read ");
    RTDX_read(&ichan,buffer,sizeof(buffer));
    puts("\n\n Read Completed");
    printf("i=%d",i);

```

```
while(i<=2000){
```

```

while(!RTDX_isInputEnabled(&ichan))
    puts("\n\n Waiting to read ");
    RTDX_read(&ichan,buffer,sizeof(buffer));
    puts("\n\n Read Completed");
    printf("i=%d",i);

```

```

/* tau1[i+1]=buffer[0]; */
/* tau2[i+1]=buffer[1]; */
/* q1[i]=buffer[2]; */
/* q1us[i]=buffer[3]; */
/* q2[i]=buffer[4]; */
/* q2us[i]=buffer[5]; */
/* q1[i+1]=buffer[6]; */
/* q1us[i+1]=buffer[7]; */
/* q2[i+1]=buffer[8]; */
/* q2us[i+1]=buffer[9]; */
giris[0]=buffer[0];giris[1]=buffer[1];//tau1(i+1)==tau1_i_1
giris[2]=buffer[2];giris[3]=buffer[3];//q1(i)==q1_i
giris[4]=buffer[4];giris[5]=buffer[5];
q1_i_1=buffer[6];q1us_i_1=buffer[7];
q2_i_1=buffer[8];q2us_i_1=buffer[9];
i=buffer[10];

```

```

r1_tau105[i]=0;r2_tau105[i]=0;r3_tau105[i]=0;r4_tau105[i]=0;
r1_tau107[i]=0;r2_tau107[i]=0;r3_tau107[i]=0;r4_tau107[i]=0;
r1_tau205[i]=0;r2_tau205[i]=0;r3_tau205[i]=0;r4_tau205[i]=0;
r1_tau207[i]=0;r2_tau207[i]=0;r3_tau207[i]=0;r4_tau207[i]=0;

```

```

q1cap[i]=0;
r1[i]=0;

```

```

q1uscap[i]=0;
r2[i]=0;

```

```

q2cap[i]=0;
r3[i]=0;

```

```

q2uscap[i]=0;
r4[i]=0;

```

```

q1cap[i]=ANFIS1(giris,a_ai1,b_ai1,c_ai1,pqr1);
r1[i]=q1_i_1-q1cap[i];

```

```

q1uscap[i]=ANFIS1(giris,a_ai2,b_ai2,c_ai2,pqr2);
r2[i]=q1us_i_1-q1uscap[i];

q2cap[i]=ANFIS1(giris,a_ai3,b_ai3,c_ai3,pqr3);
r3[i]=q2_i_1-q2cap[i];

q2uscap[i]=ANFIS1(giris,a_ai4,b_ai4,c_ai4,pqr4);
r4[i]=q2us_i_1-q2uscap[i];
/*tau105 arızalı sistem*/
q1cap_tau105[0]=ANFIS1(giris,a_ai5,b_ai5,c_ai5,pqr5);
r1_tau105[i]=q1_i_1-q1cap_tau105[0];

q1uscap_tau105[0]=ANFIS1(giris,a_ai6,b_ai6,c_ai6,pqr6);
r2_tau105[i]=q1us_i_1-q1uscap_tau105[0];

q2cap_tau105[0]=ANFIS1(giris,a_ai7,b_ai7,c_ai7,pqr7);
r3_tau105[i]=q2_i_1-q2cap_tau105[0];

q2uscap_tau105[0]=ANFIS1(giris,a_ai8,b_ai8,c_ai8,pqr8);
r4_tau105[i]=q2us_i_1-q2uscap_tau105[0];

/*tau107 arızalı sistem*/
q1cap_tau107[0]=ANFIS1(giris,a_ai9,b_ai9,c_ai9,pqr9);
r1_tau107[i]=q1_i_1-q1cap_tau107[0];

q1uscap_tau107[0]=ANFIS1(giris,a_ai10,b_ai10,c_ai10,pqr10);
r2_tau107[i]=q1us_i_1-q1uscap_tau107[0];

q2cap_tau107[0]=ANFIS1(giris,a_ai11,b_ai11,c_ai11,pqr11);
r3_tau107[i]=q2_i_1-q2cap_tau107[0];

q2uscap_tau107[0]=ANFIS1(giris,a_ai12,b_ai12,c_ai12,pqr12);
r4_tau107[i]=q2us_i_1-q2uscap_tau107[0];
/*tau205 arızalı sistem*/
q1cap_tau205[0]=ANFIS1(giris,a_ai13,b_ai13,c_ai13,pqr13);
r1_tau205[i]=q1_i_1-q1cap_tau205[0];

q1uscap_tau205[0]=ANFIS1(giris,a_ai14,b_ai14,c_ai14,pqr14);
r2_tau205[i]=q1us_i_1-q1uscap_tau205[0];

q2cap_tau205[0]=ANFIS1(giris,a_ai15,b_ai15,c_ai15,pqr15);
r3_tau205[i]=q2_i_1-q2cap_tau205[0];

q2uscap_tau205[0]=ANFIS1(giris,a_ai16,b_ai16,c_ai16,pqr16);
r4_tau205[i]=q2us_i_1-q2uscap_tau205[0];

/*tau207 arızalı sistem*/
q1cap_tau207[0]=ANFIS1(giris,a_ai17,b_ai17,c_ai17,pqr17);
r1_tau207[i]=q1_i_1-q1cap_tau207[0];

q1uscap_tau207[0]=ANFIS1(giris,a_ai18,b_ai18,c_ai18,pqr18);
r2_tau207[i]=q1us_i_1-q1uscap_tau207[0];

q2cap_tau207[0]=ANFIS1(giris,a_ai19,b_ai19,c_ai19,pqr19);
r3_tau207[i]=q2_i_1-q2cap_tau207[0];

q2uscap_tau207[0]=ANFIS1(giris,a_ai20,b_ai20,c_ai20,pqr20);
r4_tau207[i]=q2us_i_1-q2uscap_tau207[0];

```

```

        if ((r1[i]<0.011&&r1[i]>-0.01)&&(r2[i]<0.0035&&r2[i]>-
0.003)&&(r3[i]<0.015&&r3[i]>-0.013)&&(r4[i]<0.0025&&r4[i]>-0.0025)){
            f2_i[i]=1;
            if(f2_i[i-80]==1){
                f1_i[i]=1;
                f1[0]=1;f1[1]=0;f1[2]=0;f1[3]=0;f1[4]=0;

                faults_0[i]=f1[0];faults_1[i]=f1[1];faults_2[i]=f1[2];faults_3[i]=f1[3];faults_4[i]=f1[4];

                faults2_0[i]=f1[0];faults2_1[i]=f1[1];faults2_2[i]=f1[2];faults2_3[i]=f1[3];faults2_4[i]=f1[4];

                /*for parantezi*/
                f6=f6+1;
                continue;}}/*ikinci if parantezi*/
        else {
            f1_i[i]=0;
        }/*else parantezi*/

        if ((r1_tau107[i]<0.013&&r1_tau107[i]>-0.013)&&(r2_tau107[i]<0.0018&&r2_tau107[i]>-
0.0018)&&(r3_tau107[i]<0.025&&r3_tau107[i]>-0.023)&&(r4_tau107[i]<0.0022&&r4_tau107[i]>-
0.0018)){
            f1[0]=0;f1[1]=0;f1[2]=1;f1[3]=0;f1[4]=0;
            f2=f2+1;
        }

        if (((r1_tau105[i]<0.015&&r1_tau105[i]>-0.015)&&(r2_tau105[i]<0.0022&&r2_tau105[i]>-
0.0024)&&(r3_tau105[i]<0.02&&r3_tau105[i]>-0.02)&&
            (r4_tau105[i]<0.0055&&r4_tau105[i]>-0.0055))||((r1_tau105[i]<0.015&&r1_tau105[i]>-
0.015)&&(r2_tau105[i]<0.0022&&r2_tau105[i]>-0.0024)&&(r3_tau105[i]<0.02&&r3_tau105[i]>-
0.02)&&
            (r4_tau105[i]<0.0055&&r4_tau105[i]>-0.0055)&&(r1_tau107[i]<0.013&&r1_tau107[i]>-
0.013)&&(r2_tau107[i]<0.0018&&r2_tau107[i]>-0.0018)&&(r3_tau107[i]<0.025&&r3_tau107[i]>-
0.023)&&
            (r4_tau107[i]<0.0022&&r4_tau107[i]>-0.0018))){
            f1[0]=0;f1[1]=1;f1[2]=0;f1[3]=0;f1[4]=0;
            f3=f3+1;
        }

        if ((r1_tau207[i]<0.016&&r1_tau207[i]>-0.015)&&(r2_tau207[i]<0.0014&&r2_tau207[i]>-
0.0015)&&(r3_tau207[i]<0.025&&r3_tau207[i]>-0.022)&&(r4_tau207[i]<0.0014&&r4_tau207[i]>-
0.0013)){
            f1[0]=0;f1[1]=0;f1[2]=0;f1[3]=0;f1[4]=1;
            f4=f4+1;
        }

        if (((r1_tau205[i]<0.016&&r1_tau205[i]>-0.014)&&(r2_tau205[i]<0.0042&&r2_tau205[i]>-
0.005)&&(r3_tau205[i]<0.025&&r3_tau205[i]>-0.022)&&
            (r4_tau205[i]<0.013&&r4_tau205[i]>-0.012))||((r1_tau205[i]<0.02&&r1_tau205[i]>-
0.02)&&(r2_tau205[i]<0.005&&r2_tau205[i]>-0.005)&&(r3_tau205[i]<0.025&&r3_tau205[i]>-
0.025)&&
            (r4_tau205[i]<0.015&&r4_tau205[i]>-0.015)&&(r1_tau207[i]<0.016&&r1_tau207[i]>-
0.015)&&(r2_tau207[i]<0.0014&&r2_tau207[i]>-0.0015)&&(r3_tau207[i]<0.025&&r3_tau207[i]>-
0.022)&&
            (r4_tau207[i]<0.0014&&r4_tau207[i]>-0.0013))){
            f1[0]=0;f1[1]=0;f1[2]=0;f1[3]=1;f1[4]=0;
            f5=f5+1;
        }
    }

```

```

    faults2_0[i]=f1[0];faults2_1[i]=f1[1];faults2_2[i]=f1[2];faults2_3[i]=f1[3];faults2_4[i]=f1[4];

    if ((faults2_0[i-80]==faults2_0[i])&&(faults2_1[i-80]==faults2_1[i])&&(faults2_2[i-
80]==faults2_2[i])&&
        (faults2_3[i-80]==faults2_3[i])&&(faults2_4[i-80]==faults2_4[i])){
        faults_0[i]=f1[0];faults_1[i]=f1[1];faults_2[i]=f1[2];faults_3[i]=f1[3];faults_4[i]=f1[4];
    }
    else {
        faults_0[i]=faults_0[i-401];faults_1[i]=faults_1[i-401];faults_2[i]=faults_2[i-
401];faults_3[i]=faults_3[i-401];faults_4[i]=faults_4[i-401];
    }
}
}
//end of while (1) infinite loop
}/*main() parantezi*/
//end of main

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### **MATLAB Programı**

```

C:\Documents and Settings\Tolga\Belgelerim\MATLAB\CT_actuator_fault_GOS_RTDX.m

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Computed torque 2-link robot model constants

```

```

% Ex.3.4

```

```

% s.131

```

```

clear all

```

```

close all

```

```

tic

```

```

a1=1;

```

```

a2=1;

```

```

m1=1;

```

```

m2=1;

```

```

g1=pi*6/180;

```

```

g2=pi*6/180;

```

```

wn=10;

```

```

kp=100;

```

```

kv=20;

```

```

ki=500;

```

```

deltat=0.01;

```

```

g=9.81;

```

```

Kp=[100 0;0 100];

```

```

Kv=[20 0;0 20];

```

```

Ki=[500 0;0 500];

```

```

t=0.01:0.01:70.01;

```

```

ccsboardinfo %board info

```

```

cc = ccsdsp('boardnum',0); %set up CCS object

```

```

reset(cc) %reset board

```

```

visible(cc,1); %for CCS window

```

```

enable(cc.rtdx); %enable RTDX

```

```

if ~isEnabled(cc.rtdx)

```

```

    error('RTDX is not enabled')

```

```

end

```

```

cc.rtdx.set('timeout', 20); %set 20sec time out for RTDX

```

```

open(cc,'D:\c6713_book_cd\digital_signal\rtdx_matlab_sim\RTDX_MATLAB_sim.pjt'); %open

```

```

project

```

```

load(cc,'D:\c6713_book_cd\digital_signal\rtdx_matlab_sim\debug\RTDX_MATLAB_sim.out'); %load
executable file
run(cc); %run
configure(cc.rtdx,64000,1000); %configure two RTDX channels
open(cc.rtdx,'ichan','w'); %open input channel
open(cc.rtdx,'ochan','r'); %open output channel
pause(3) %wait for RTDX channel to open

qd1=0.8*cos(t/2);
qd2=0.8*sin(t/2);
for i=2:7000
    qd1us(i)=(qd1(i)-qd1(i-1))/deltat;
    qd2us(i)=(qd2(i)-qd2(i-1))/deltat;

    qd1uss(i)=(qd1us(i)-qd1us(i-1))/deltat;
    qd2uss(i)=(qd2us(i)-qd2us(i-1))/deltat;
end

% Comp_torque_PID_ANFIS

q1(1)=0.1;
q2(1)=0;
q1us(1)=0;
q2us(1)=0;
q1(2)=0.1;
q2(2)=0;
q1us(2)=0;
q2us(2)=0;

e1(1)=qd1(1)-q1(1);
e2(1)=qd2(1)-q2(1);
e1(2)=e1(1);
e2(2)=e2(1);

e1us(2)=(e1(2)-e1(1))/deltat;
e2us(2)=(e2(2)-e2(1))/deltat;

eps1(1)=0;
eps2(1)=0;
eps1(2)=eps1(1)+deltat*e1(1);
eps2(2)=eps2(1)+deltat*e2(1);

q1us(1)=(q1(2)-q1(1))/deltat;
q2us(1)=(q2(2)-q2(1))/deltat;

faults=[];
faults2=[];
f2=0;%Diğer koşulların oluşup oluşmadığını kontrol için
f3=0;
f4=0;
f5=0;
f6=0;
f1_i=ones(1,400);
f2_i=ones(1,400);
for i=2:500
    m2=1;
    M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
    M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));

```

```

M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
M22(i)=m2*a2^2;
M=[M11(i) M12(i);M21(i) M22(i)];

V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i)+q2us(i)^2)*sin(q2(i)));
V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
V=[V1(i);V2(i)];

G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
G2(i)=m2*g*a2*cos(q1(i)+q2(i));
G=[G1(i);G2(i)];

tau1(i+1)=[M11(i)
M12(i)*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V1(i)+G1(i)
)];
tau2(i+1)=[M21(i)
M22(i)*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V2(i)+G2(i)
)];
tau=[tau1(i+1);tau2(i+1)];

% belirsizlik ekleme
uncertainty_function

quss=inv(Ma)*(tau-Va-Ga);

q1uss(i+1)=quss(1);
q2uss(i+1)=quss(2);
q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
q1(i+1)=q1(i)+deltat*q1us(i+1);
q2(i+1)=q2(i)+deltat*q2us(i+1);

e1(i+1)=qd1(i+1)-q1(i+1);
e2(i+1)=qd2(i+1)-q2(i+1);
e1us(i+1)=qd1us(i)-q1us(i);
e2us(i+1)=qd2us(i)-q2us(i);
eps1(i+1)=eps1(i)+deltat*e1(i+1);
eps2(i+1)=eps2(i)+deltat*e2(i+1);
end

for i=501:2400
m2=1;
M11(i)=(m1+m2)*a1^2+m2*a2^2+2*m2*a1*a2*cos(q2(i));
M12(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
M21(i)=m2*a2^2+m2*a1*a2*cos(q2(i));
M22(i)=m2*a2^2;
M=[M11(i) M12(i);M21(i) M22(i)];

V1(i)=-m2*a1*a2*(2*(q1us(i)*q2us(i)+q2us(i)^2)*sin(q2(i)));
V2(i)=m2*a1*a2*q1us(i)^2*sin(q2(i));
V=[V1(i);V2(i)];

G1(i)=(m1+m2)*g*a1*cos(q1(i))+m2*g*a2*cos(q1(i)+q2(i));
G2(i)=m2*g*a2*cos(q1(i)+q2(i));
G=[G1(i);G2(i)];

tau1(i+1)=[M11(i)
M12(i)*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V1(i)+G1(i)
)];

```

```

    tau2(i+1)=[M21(i)
M22(i)]*([qd1uss(i);qd2uss(i)]+Kv*[e1us(i);e2us(i)]+Kp*[e1(i);e2(i)]+Ki*[eps1(i);eps2(i)])+V2(i)+G2(i)
);
    tau=[tau1(i+1);tau2(i+1)];

    % belirsizlik ekleme
    uncertainty_function

    quss=inv(Ma)*(tau-Va-Ga);

    q1uss(i+1)=quss(1);
    q2uss(i+1)=quss(2);
    q1us(i+1)=q1us(i)+deltat*q1uss(i+1);
    q2us(i+1)=q2us(i)+deltat*q2uss(i+1);
    q1(i+1)=q1(i)+deltat*q1us(i+1);
    q2(i+1)=q2(i)+deltat*q2us(i+1);

    e1(i+1)=qd1(i+1)-q1(i+1);
    e2(i+1)=qd2(i+1)-q2(i+1);
    e1us(i+1)=qd1us(i)-q1us(i);
    e2us(i+1)=qd2us(i)-q2us(i);
    eps1(i+1)=eps1(i)+deltat*e1(i+1);
    eps2(i+1)=eps2(i)+deltat*e2(i+1);

    /* tau1[i+1]=buffer[0]; */
    /* tau2[i+1]=buffer[1]; */
    /* q1[i]=buffer[2]; */
    /* q1us[i]=buffer[3]; */
    /* q2[i]=buffer[4]; */
    /* q2us[i]=buffer[5]; */
    /* q1[i+1]=buffer[6]; */
    /* q1us[i+1]=buffer[7]; */
    /* q2[i+1]=buffer[8]; */
    /* q2us[i+1]=buffer[9]; */
    enable(cc.rtdx,'ichan'); %enable channel TO DSK
    indata=[tau1(i+1) tau2(i+1) q1(i) q1us(i) q2(i) q2us(i) q1(i+1) q1us(i+1) q2(i+1) q2us(i+1) i];
    if isenabled(cc.rtdx,'ichan')
        writemsg(cc.rtdx,'ichan', double(indata)) %send 16-bit data to DSK
        pause(3)
        i
    else
        error('Channel "ichan" is not enabled')
    end

end

```

### Ek I. 3. Şema İçin Visual Basic-DSP Programları

#### DSP Programı

D:\c6713\_book\_cd\digital\_signal\rtdx\_vbsine\rtdx\_vbsine.pjt

D:\c6713\_book\_cd\digital\_signal\rtdx\_vbsine\rtdx\_vbsine.c

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
//#include "Configuration1cfg.h" //generated by .cdb file
#include <C:\CCStudio_v3.1\C6000\rtdx\include\rtdx.h> // for rtdx support
#include <math.h>
#include <stdio.h>
#include <c6x.h> //RTDX support
file
#include "D:\c6713_book_cd\digital_signal\rtdx_vbsine\target.h"
```

```
#define PI 3.14159265358
#pragma DATA_SECTION(tau1, ".EXT_RAM")
#pragma DATA_SECTION(tau2, ".EXT_RAM")
#pragma DATA_SECTION(qd1, ".EXT_RAM")
#pragma DATA_SECTION(qd2, ".EXT_RAM")
#pragma DATA_SECTION(r1, ".EXT_RAM")
#pragma DATA_SECTION(r2, ".EXT_RAM")
#pragma DATA_SECTION(r3, ".EXT_RAM")
#pragma DATA_SECTION(r4, ".EXT_RAM")
#pragma DATA_SECTION(r2r2, ".EXT_RAM")
#pragma DATA_SECTION(r4r4, ".EXT_RAM")
#pragma DATA_SECTION(faults1, ".EXT_RAM")
#pragma DATA_SECTION(faults2, ".EXT_RAM")
#pragma DATA_SECTION(v_gecici1, ".EXT_RAM")
#pragma DATA_SECTION(v_gecici2, ".EXT_RAM")
#pragma DATA_SECTION(g_gecici1, ".EXT_RAM")
#pragma DATA_SECTION(g_gecici2, ".EXT_RAM")
#pragma DATA_SECTION(qd1us, ".EXT_RAM")
#pragma DATA_SECTION(qd2us, ".EXT_RAM")
#pragma DATA_SECTION(qd1uss, ".EXT_RAM")
#pragma DATA_SECTION(qd1uss, ".EXT_RAM")
#pragma DATA_SECTION(q1us, ".EXT_RAM")
#pragma DATA_SECTION(q2us, ".EXT_RAM")
#pragma DATA_SECTION(q1cap, ".EXT_RAM")
#pragma DATA_SECTION(q1uscap, ".EXT_RAM")
#pragma DATA_SECTION(q2cap, ".EXT_RAM")
#pragma DATA_SECTION(q2uscap, ".EXT_RAM")
```

```
/* #pragma DATA_SECTION(q1cap, "SDRAM") */
#include "D:\c6713\projects\sine_floating\q1cap_coef_double.h" //q1cap ANFIS katsayıları
#include "D:\c6713\projects\sine_floating\q1uscap_coef_double.h" //q1uscap ANFIS katsayıları
#include "D:\c6713\projects\sine_floating\q2cap_coef_double.h" //q2cap ANFIS katsayıları
#include "D:\c6713\projects\sine_floating\q2uscap_coef_double.h" //q2uscap ANFIS katsayıları
```

```
#include "D:\c6713\projects\sine_floating\func_app1.h" //func_app1 katsayıları
#include "D:\c6713\projects\sine_floating\func_app2.h" //func_app2 katsayıları
```

```
double ANFIS1(double ANFIS_giris[6], double a_ai22[6][2], double b_bi22[6][2], double
c_ci22[6][2], double pqr22[64][7]);
```

```
void NNet1(double NNet_giris[6],double IW[10][6],double LW11[20][10],double LW22[20][20],double
LW33[4][20],double bias11[10],double bias22[20],double bias33[20],double bias44[4]);
RTDX_CreateInputChannel(control_channel); //create input channel-data transfer PC-->DSK
RTDX_CreateOutputChannel(ochan);
```

```
double deltat=0.01,a1=1,a2=1,m1=1,m2=1,g=9.81;
double qd1[4001],qd2[4001],t[4001];
double q1us_eski,q2us_eski,q1uss_eski,q2uss_eski;
double q1us[4001],q2us[4001];
double e1_eski,e2_eski,e1us_eski,e2us_eski,eps1_eski,eps2_eski;
double m_gecici,v_gecici1[4000],v_gecici2[4000],g_gecici1[4000],g_gecici2[4000];
double tau11[4000],tau22[4000],q1[4000],q2[4000];
double tau1[4000],tau2[4000],r1[4000],r2[4000],r3[4000],r4[4000],r2r2[4000],r4r4[4000];
double M[2][2],inv_M[2][2],qd1us_yeni,qd2us_yeni;
double qd1us[4000],qd2us[4000],qd1us_eski2,qd2us_eski2,qd1uss[4000],qd2uss[4000];
double q1_yeni,q2_yeni,q1us_yeni,q2us_yeni,q1uss_yeni,q2uss_yeni;
double e1_yeni,e2_yeni,e1us_yeni,e2us_yeni,eps1_yeni,eps2_yeni;
double faults1[4000],faults2[4000],faults3[4000],faults4[4000],total[4];
double faults[4][4000],M00,M01,M10,M11;
int thr,faulty,a;
double q1cap[4001],q1uscap[4001],q2cap[4001],q2uscap[4001];
double buffer0[1],buffer1[1],buffer2[1],buffer3[1],buffer4[1],buffer5[1];
double buffer6[1],buffer7[1],buffer8[1],buffer9[1],buffer10[1],giris[6];
double faults11[4001],faults22[4001];
int f_out1[4000],f_out2[4000];
```

```
//ANFIS altprogramı
double ANFIS1(double ANFIS_giris[6],double a_ai22[6][2],double b_ai22[6][2],double
c_ai22[6][2],double pqr22[64][7])
{
int ii,jj;
/*double a_ai[6,2],b_ai[6,2],c_ai[6,2];*/
double mu_AB[6][2];
double gecici,gecici1,w[64],norm_w[64],L4[64];
double sum,total1;
//Girişler
//ANFIS_giris[0];//tau1(i+1)
//ANFIS_giris[1];//tau2(i+1)
//ANFIS_giris[2];//q1(i)
//ANFIS_giris[3];//q1us(i)
//ANFIS_giris[4];//q2(i)
//ANFIS_giris[5];//q2us(i)
sum=0;
total1=0;
// KAT 1
// mu kattaki üyelik işlevi
// tanımlamalara göre her katta 2 üyelik işlevi var.
// her işlev için
// ai:merkez
// bi:eğim
// ci:genişliği temsil etmektedir.
//6 giriş için tanımlama
for (ii=0;ii<6;ii++){
for (jj=0;jj<2;jj++){
gecici=(ANFIS_giris[ii]-c_ai22[ii][jj])/a_ai22[ii][jj];
if (gecici<0) gecici=-gecici;
```

mu\_AB[ii][jj]=1/(1+pow(gecici,2\*b\_ai22[ii][jj]));

}  
}

//KAT 2

//tanımlanan üyelik işlevleri birbiri ile çarpılır.

//Bu çarpımlara ateşleme gücü adı verilir ve kurallara göre tanımlanır.

w[0]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[1]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[2]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[3]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[4]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[5]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[6]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[7]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[8]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[9]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[10]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[11]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[12]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[13]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[14]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[15]=mu\_AB[0][0]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[16]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[17]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[18]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[19]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[20]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[21]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[22]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[23]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[24]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[25]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[26]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[27]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[28]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[29]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[30]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[31]=mu\_AB[0][0]\*mu\_AB[1][1]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[32]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[33]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[34]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[35]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[36]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[37]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[38]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[39]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][0]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[40]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[41]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[42]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[43]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[44]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[45]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[46]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][0];  
w[47]=mu\_AB[0][1]\*mu\_AB[1][0]\*mu\_AB[2][1]\*mu\_AB[3][1]\*mu\_AB[4][1]\*mu\_AB[5][1];  
w[48]=mu\_AB[0][1]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][0];  
w[49]=mu\_AB[0][1]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][0]\*mu\_AB[5][1];  
w[50]=mu\_AB[0][1]\*mu\_AB[1][1]\*mu\_AB[2][0]\*mu\_AB[3][0]\*mu\_AB[4][1]\*mu\_AB[5][0];

```

w[51]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[52]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[53]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[54]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[55]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][0]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];
w[56]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][0];
w[57]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][0]*mu_AB[5][1];
w[58]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][0];
w[59]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][0]*mu_AB[4][1]*mu_AB[5][1];
w[60]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][0];
w[61]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][0]*mu_AB[5][1];
w[62]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][0];
w[63]=mu_AB[0][1]*mu_AB[1][1]*mu_AB[2][1]*mu_AB[3][1]*mu_AB[4][1]*mu_AB[5][1];

```

```
//KAT 3
```

```
//Tanımlanan ateşleme güçleri normalize edilir.
```

```

for (ii=0;ii<64;ii++)
sum=sum+w[ii];
for (ii=0;ii<64;ii++)
norm_w[ii]=w[ii]/sum;

```

```
//KAT 4
```

```
//Normalize edilen ateşleme açıları birinci derece bir fonksiyon ile çarpılır.
```

```

for (ii=0;ii<64;ii++){
gecici1=0;
for (jj=0;jj<6;jj++){
gecici1=gecici1+pqr22[ii][jj]*ANFIS_giris[jj];
}
gecici1=gecici1+pqr22[ii][6];
L4[ii]=norm_w[ii]*gecici1;
}

```

```
//KAT 5
```

```
//Kat 4'ten gelen tüm değerler toplanır
```

```

for (ii=0;ii<64;ii++)
total1=total1+L4[ii];
return total1;

```

```
}
```

```
//ANA PROGRAM
```

```
void main(void)
```

```
{
```

```
int i=0,j;
```

```
double q1_i_1,q1us_i_1,q2_i_1,q2us_i_1;
```

```
faulty=0;
```

```
TARGET_INITIALIZE();
```

```

for (j=0;j<=500;j++){
r1[j]=0;r2[j]=0;r3[j]=0;r4[j]=0;r2r2[j]=0;r4r4[j]=0;
}

```

```

for (j=0;j<=11;j++){
giris[j]=0;
}

```

```

buffer0[0]=0;buffer1[0]=0;buffer2[0]=0;buffer3[0]=0;buffer4[0]=0;buffer5[0]=0;
buffer6[0]=0;buffer7[0]=0;buffer8[0]=0;buffer9[0]=0;buffer10[0]=0;

```

```
while(i<=1000){
```

```
RTDX_enableInput(&control_channel); //enable input channel */
```

```
if(!RTDX_channelBusy(&control_channel)){
```

```
RTDX_read(&control_channel,buffer0,sizeof(buffer0));
```

```

RTDX_read(&control_channel,buffer1,sizeof(buffer1));
RTDX_read(&control_channel,buffer2,sizeof(buffer2));
RTDX_read(&control_channel,buffer3,sizeof(buffer3));
RTDX_read(&control_channel,buffer4,sizeof(buffer4));
RTDX_read(&control_channel,buffer5,sizeof(buffer5));
RTDX_read(&control_channel,buffer6,sizeof(buffer6));
RTDX_read(&control_channel,buffer7,sizeof(buffer7));
RTDX_read(&control_channel,buffer8,sizeof(buffer8));
RTDX_read(&control_channel,buffer9,sizeof(buffer9));
RTDX_read(&control_channel,buffer10,sizeof(buffer10));
}

/* tau1[i+1]=buffer[0]; */ /* tau2[i+1]=buffer[1]; */
/* q1[i]=buffer[2]; */ /* q1us[i]=buffer[3]; */
/* q2[i]=buffer[4]; */ /* q2us[i]=buffer[5]; */
/* q1[i+1]=buffer[6]; */ /* q1us[i+1]=buffer[7]; */
/* q2[i+1]=buffer[8]; */ /* q2us[i+1]=buffer[9]; */
giris[0]=buffer0[0];giris[1]=buffer1[0];//tau1(i+1)==tau1_i_1
giris[2]=buffer2[0];giris[3]=buffer3[0];//q1(i)==q1_i
giris[4]=buffer4[0];giris[5]=buffer5[0];
q1_i_1=buffer6[0];q1us_i_1=buffer7[0];
q2_i_1=buffer8[0];q2us_i_1=buffer9[0];
i=buffer10[0];

q1cap[i]=0;
r1[i]=0;

q1uscap[i]=0;
r2[i]=0;

q2cap[i]=0;
r3[i]=0;

q2uscap[i]=0;
r4[i]=0;

r2r2[i]=0;
r4r4[i]=0;
q1cap[i]=ANFIS1(giris,a_ai1,b_ai1,c_ai1,pqr1);
r1[i]=q1_i_1-q1cap[i];

q1uscap[i]=ANFIS1(giris,a_ai2,b_ai2,c_ai2,pqr2);
r2[i]=q1us_i_1-q1uscap[i];

q2cap[i]=ANFIS1(giris,a_ai3,b_ai3,c_ai3,pqr3);
r3[i]=q2_i_1-q2cap[i];

q2uscap[i]=ANFIS1(giris,a_ai4,b_ai4,c_ai4,pqr4);
r4[i]=q2us_i_1-q2uscap[i];

r2r2[i]=(r1[i]-r1[i-1])/deltat;
r4r4[i]=(r3[i]-r3[i-1])/deltat;
if(i>502){

if ((r1[i]>0.011|r1[i]<-0.01)|(r2[i]>0.0035|r2[i]<-0.003)|(r3[i]>0.015|r3[i]<-0.013)|(r4[i]>0.0025|r4[i]<-
0.0025)|(r2r2[i]>0.015|r2r2[i]<-0.015)|(r4r4[i]>0.016|r4r4[i]<-0.017)){
if (a==0){
a=i;
}
}
}

```



**Visual Basic Programı**

D:\c6713\_book\_cd\digital\_signal\rtdx\_vbsine\PC\rtdx\_vbsine.vbp

D:\c6713\_book\_cd\digital\_signal\rtdx\_vbsine\PC\slider.frm

//

```

                                'RTDX OLE API Status Return codes
Const SUCCESS = &H0                'Method call successful
Const FAIL = &H80004005            'Method call failure
Const ENoDataAvailable = &H8003001E 'No data is currently available
Const EEndOfLogFile = &H80030002   'End of log file

Dim rtdx As Object

Private Sub Command1_Click()
'Burada RTDX ile DSP haberleşmesi başlatılır ve sağlanır
'ROBOT DİZGESİNİN GERÇEKLENMESİ'
  Dim a1, a2, m1, m2, wn, deltata As Double
  Dim Kv(1 To 2, 1 To 2), Kp(1 To 2, 1 To 2), Ki(1 To 2, 1 To 2) As Double
  Dim t(1 To 6001), qd1(1 To 6001), qd2(1 To 6001) As Double
  Dim qd1us(1 To 6001), qd2us(1 To 6001) As Double
  Dim qd1uss(1 To 6001), qd2uss(1 To 6001) As Double
  Dim q1(1 To 6001), q2(1 To 6001) As Double
  Dim q1us(1 To 6001), q2us(1 To 6001) As Double
  Dim q1uss(1 To 6001), q2uss(1 To 6001) As Double
  Dim e1(1 To 6001), e2(1 To 6001) As Double
  Dim e1us(1 To 6001), e2us(1 To 6001) As Double
  Dim eps1(1 To 6001), eps2(1 To 6001) As Double
  Dim M00, M01, M10, M11, V1, V2, G1, G2 As Double
  Dim M00_b, M01_b, M10_b, M11_b, V1_b, V2_b, G1_b, G2_b As Double
  Dim inv_M00, inv_M01, inv_M10, inv_M11 As Double
  Dim tau1(1 To 6001), tau2(1 To 6001), tau11(1 To 6001), tau22(1 To 6001) As Double
  Dim i As Integer

  Dim status, status2 As Long
  Dim data11, data12 As Double
  Dim bufstate As Long

  m1 = 1
  m2 = 1
  a1 = 1
  a2 = 1
  wn = 10
  g = 9.81
  deltata = 0.01

  Kv(1, 1) = 20
  Kv(1, 2) = 0
  Kv(2, 1) = 0
  Kv(2, 2) = 20

  Kp(1, 1) = 100
  Kp(1, 2) = 0
  Kp(2, 1) = 0
  Kp(2, 2) = 100

  Ki(1, 1) = 500
  Ki(1, 2) = 0

```

Ki(2, 1) = 0  
Ki(2, 2) = 500

qd1us(1) = 0  
qd2us(1) = 0  
q1(1) = 0.1  
q2(1) = 0  
q1us(1) = 0  
q2us(1) = 0  
q1(2) = 0.1  
q2(2) = 0  
q1us(2) = 0  
q2us(2) = 0

For i = 1 To 6001  
t(i) = 0.01 \* i  
qd1(i) = 0.8 \* Cos(t(i) / 2)  
qd2(i) = 0.8 \* Sin(t(i) / 2)  
Next i

For i = 2 To 6000  
qd1us(i) = (qd1(i) - qd1(i - 1)) / deltat  
qd2us(i) = (qd2(i) - qd2(i - 1)) / deltat  
  
qd1uss(i) = (qd1us(i) - qd1us(i - 1)) / deltat  
qd2uss(i) = (qd2us(i) - qd2us(i - 1)) / deltat  
Next i  
e1(1) = qd1(1) - q1(1)  
e2(1) = qd2(1) - q2(1)  
e1(2) = e1(1)  
e2(2) = e2(1)

e1us(2) = (e1(2) - e1(1)) / deltat  
e2us(2) = (e2(2) - e2(1)) / deltat

eps1(1) = 0  
eps2(1) = 0  
eps1(2) = eps1(1) + deltat \* e1(1)  
eps2(2) = eps2(1) + deltat \* e2(1)

q1us(1) = (q1(2) - q1(1)) / deltat  
q2us(1) = (q2(2) - q2(1)) / deltat

For i = 2 To 1000  
M00 = (m1 + m2) \* a1 ^ 2 + m2 \* a2 ^ 2 + 2 \* m2 \* a1 \* a2 \* Cos(q2(i))  
M01 = m2 \* a2 ^ 2 + m2 \* a1 \* a2 \* Cos(q2(i))  
M10 = m2 \* a2 ^ 2 + m2 \* a1 \* a2 \* Cos(q2(i))  
M11 = m2 \* a2 ^ 2

V1 = -m2 \* a1 \* a2 \* (2 \* (q1us(i) \* q2us(i)) + q2us(i) ^ 2) \* Sin(q2(i))  
V2 = m2 \* a1 \* a2 \* q1us(i) ^ 2 \* Sin(q2(i))

G1 = (m1 + m2) \* g \* a1 \* Cos(q1(i)) + m2 \* g \* a2 \* Cos(q1(i) + q2(i))  
G2 = m2 \* g \* a2 \* Cos(q1(i) + q2(i))

tau1(i + 1) = (M00 \* (qd1uss(i) + 20 \* e1us(i) + 100 \* e1(i) + 500 \* eps1(i)) + M01 \* (qd2uss(i) + 20 \* e2us(i) + 100 \* e2(i) + 500 \* eps2(i))) + V1 + G1  
tau2(i + 1) = (M10 \* (qd1uss(i) + 20 \* e1us(i) + 100 \* e1(i) + 500 \* eps1(i)) + M11 \* (qd2uss(i) + 20 \* e2us(i) + 100 \* e2(i) + 500 \* eps2(i))) + V2 + G2

```

tau11(i + 1) = tau1(i + 1)
tau22(i + 1) = tau2(i + 1)
'BELİRSİZLİK EKLEME _b
M00_b = M00 * (1 + 0.05 * Sin(t(i)))
M01_b = M01 * (1 + 0.05 * Sin(t(i)))
M10_b = M10 * (1 + 0.05 * Sin(t(i)))
M11_b = M11 * (1 + 0.05 * Sin(t(i)))

V1_b = V1 * (1 + 0.05 * Sin(t(i)))
V2_b = V2 * (1 + 0.05 * Sin(t(i)))
G1_b = G1 * (1 + 0.05 * Sin(t(i)))
G2_b = G2 * (1 + 0.05 * Sin(t(i)))

inv_M00 = M11_b / (M00_b * M11_b - M01_b * M10_b)
inv_M01 = -M01_b / (M00_b * M11_b - M01_b * M10_b)
inv_M10 = -M10_b / (M00_b * M11_b - M01_b * M10_b)
inv_M11 = M00_b / (M00_b * M11_b - M01_b * M10_b)

q1uss(i + 1) = inv_M00 * (tau11(i + 1) - V1_b - G1_b) + inv_M01 * (tau22(i + 1) - V2_b - G2_b)
q2uss(i + 1) = inv_M10 * (tau11(i + 1) - V1_b - G1_b) + inv_M11 * (tau22(i + 1) - V2_b - G2_b)

q1us(i + 1) = q1us(i) + deltat * q1uss(i + 1)
q2us(i + 1) = q2us(i) + deltat * q2uss(i + 1)
q1(i + 1) = q1(i) + deltat * q1us(i + 1)
q2(i + 1) = q2(i) + deltat * q2us(i + 1)

e1(i + 1) = qd1(i + 1) - q1(i + 1)
e2(i + 1) = qd2(i + 1) - q2(i + 1)
e1us(i + 1) = qd1us(i) - q1us(i)
e2us(i + 1) = qd2us(i) - q2us(i)
eps1(i + 1) = eps1(i) + deltat * e1(i + 1)
eps2(i + 1) = eps2(i) + deltat * e2(i + 1)

status = rtdx.WriteF8(tau11(i + 1), bufstate)
status = rtdx.WriteF8(tau22(i + 1), bufstate)
status = rtdx.WriteF8(q1(i), bufstate)
status = rtdx.WriteF8(q1us(i), bufstate)
status = rtdx.WriteF8(q2(i), bufstate)
status = rtdx.WriteF8(q2us(i), bufstate)
status = rtdx.WriteF8(q1(i + 1), bufstate)
status = rtdx.WriteF8(q1us(i + 1), bufstate)
status = rtdx.WriteF8(q2(i + 1), bufstate)
status = rtdx.WriteF8(q2us(i + 1), bufstate)
status = rtdx.WriteF8(i, bufstate)

Select Case status
  Case Is = SUCCESS
  Case Is = FAIL
    ' MsgBox "Error: Writing data failed" & dataI4
  Case Else
    MsgBox "Volume data backed up waiting for DSP to read it"
End Select
Next i
MsgBox "RTDX bitti"
Text1.Text = Format(i, "Fixed")

```

End Sub

Private Sub Form\_Load()

Dim status As Long

BoardProc\_frm.Show 1

' Get application objects

Set rtdx = CreateObject("RTDX")

status = rtdx.SetProcessor(BoardProc\_frm.CurrentSelectedBoard, \_  
BoardProc\_frm.CurrentSelectedProcessor)

If (status <> SUCCESS) Then

MsgBox "Set Processor failed", vbCritical

Exit Sub

End If

' open target's input channel

status = rtdx.Open("control\_channel", "W")

Select Case status

Case Is = SUCCESS

Case Is = FAIL

MsgBox "Unable to open control\_channel", vbCritical, "Error"

Exit Sub

Case Else

MsgBox "Unknown return value from control\_channel open", vbInformation

Exit Sub

End Select

End Sub

Private Sub Form\_Unload(Cancel As Integer)

' close target's input channel

status = rtdx.Close()

Select Case status

Case Is = SUCCESS

Case Is = FAIL

MsgBox "Unable to close input channel", vbCritical, "Error"

Case Else

MsgBox "Unknown return value from input close", vbInformation

End Select

Set rtdx = Nothing

' kill RTDX OLE object

Unload BoardProc\_frm

End Sub

Private Sub Slider1\_Change()

Dim status, status2 As Long

Dim data1, data2 As Double

Dim bufstate As Long

data1 = 0.1

data2 = 0.2

' write volume to target input channel

' data1 = Slider1.Value

status = rtdx.WriteF8(data1, bufstate)

status = rtdx.WriteF8(data2, bufstate)

Select Case status

Case Is = SUCCESS

Case Is = FAIL

```
' MsgBox "Error: Writing data failed" & dataI4  
Case Else  
  MsgBox "Volume data backed up waiting for DSP to read it"  
End Select
```

```
'Dim status, status2 As Long  
'Dim dataI1, dataI2 As Double  
'Dim bufstate As Long  
' Dim i As Integer  
'For i = 1 To 10  
  ' dataI1 = 0.1 * i  
  ' dataI2 = 0.2 * i  
  ' write volume to target input channel  
  ' dataI4 = Slider1.Value  
  ' status = rtdx.WriteF8(dataI1, bufstate)  
  ' status = rtdx.WriteF8(dataI2, bufstate)  
End Sub
```

## YAYINLAR LİSTESİ

### A. Uluslararası Hakemli Dergilerde Yayımlanan Makaleler (SCI-Expanded):

1. T. Yüksel, A. Sezgin, "Two Fault Detection and Isolation Schemes for Robot Manipulators Using Soft Computing Techniques", *Applied Soft Computing*, Cilt: 10, s: 125-134, Ocak 2010, doi:10.1016/j.asoc.2009.06.011
2. T. Yüksel, A. Sezgin, "Doğrusal Olmayan Dizgeler İçin Model Temelli Arıza Bulma-Yalıtımı ve Robot Manipülörlere Uygulanması", *Gazi Üniv. Müh. Mim. Fak. Der.*, Cilt: 24, No: 1, s: 79 - 94, Mart 2009

### B. Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler:

1. T. Yüksel, A. Sezgin, "A Model-Based Fault Function Approximation Scheme for Robot Manipulators Using M-ANFIS", *The 15. IEEE Mediterranean Electrotechnical Conference - MELECON' 2010*, Valetta - MALTA, Nisan 2010
2. T. Yüksel, A. Sezgin, "Fault Detection and Insulation for Robot Manipulators Using Anfis and Wavelet", *5. Int. Conference on Electrical and Electronics Engineering - ELECO'2007*, s: 372 - 376, Bursa - TÜRKİYE, 2007

### C. Ulusal Bilimsel Toplantılarda Sunulan ve Bildiri Kitabında Basılan Bildiriler:

1. T. Yüksel, A. Sezgin, "Robot Manipülörlere İçin Kazanç Çizgeleme Yöntemiyle Arızaya Dayanımlı Denetim", *Otomatik Kontrol Ulusal Toplantısı - TOK'10*, GYTE - Kocaeli, 2010 (Kabul edildi)
2. T. Yüksel, A. Sezgin, "Robot Manipülörlere İçin Hesapsal Zeka Teknikleri Kullanan Model Temelli İki FDI Şeması", *Otomatik Kontrol Ulusal Toplantısı - TOK'08*, s: 154 - 159, İTÜ - İSTANBUL, 2008
3. T. Yüksel, A. Sezgin, "Robot Manipülörlere İçin M-ANFIS ve SA ile Model Temelli Arıza Bulma ve Yalıtımı", *Sinyal İşleme, İletişim ve Uygulamaları Kurultayı - SIU 2008*, AYDIN, 2008
4. T. Yüksel, A. Sezgin, "Robot Manipülörlere İçin Anfis ve Dalgacık Kullanarak Arıza Bulma ve İzolasyonu", *Otomatik Kontrol Ulusal Toplantısı - TOK'07*, s: 140 - 143, Sabancı Üniversitesi - İSTANBUL, 2007

**D. Kitap ve Kitap Bölümleri:**

1. T. Yüksel, A. Sezgin, “Model Based FDI Schemes For Robot Manipulators Using Soft Computing Techniques”, *Fault Detection*, IN-TECH Publishing, Mayıs-2010, ISBN 978-953-307-037-7

**ÖZGEÇMİŞ**

Adı Soyadı : TOLGA YÜKSEL  
Doğum Yeri : KARABÜK  
Doğum Tarihi : 23.10.1980  
Medeni Hali : BEKAR  
Bildiği Yabancı Diller : İNGİLİZCE

**Eğitim Durumu (Kurum ve Yıl)**

Lise : KARABÜK ANADOLU LİSESİ 1998  
Lisans : O.M.Ü. ELEKTRİK-ELEKTRONİK MÜH. 2002  
Yüksek Lisans: O.M.Ü. ELEKTRİK-ELEKTRONİK MÜH. 2004

**Çalıştığı Kurum/Kurumlar ve Yıl :**

O.M.Ü. ELEKTRİK-ELEKTRONİK MÜH. 2002 - 2004  
O.M.Ü. ELEKTRİK-ELEKTRONİK MÜH. 2007 -

**İletişim Bilgileri :**

Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi  
Elektrik-Elektronik Mühendisliği Bölümü  
Kurupelit Kampüsü - Kurupelit  
55139 Samsun  
0 362 312 19 19 / 1118  
tyuksel@omu.edu.tr