

# Eş zamanlı Arıza Teşhisi için FPGA Tabanlı Akıllı Durum İzleme Yöntemlerinin Geliştirilmesi

## Development of FPGA Based Intelligent Condition Monitoring Methods for Synchronously Fault Diagnosis

İlhan Aydın, Mehmet Karaköse, Erhan Akin

Bilgisayar Mühendisliği Bölümü

Fırat Üniversitesi

iaaydin@firat.edu.tr, mkarakose@firat.edu.tr, eakin@firat.edu.tr

### Özet

Çağdaş endüstriyel sistemlerde bütün sistem başarımının iyi bir düzeyde tutulması gerekir. Bu sistemlerde asenkron motor önemli bileşenlerden biridir ve iş gücünün büyük bir kısmını karşılarlar. Bu motorlarda oluşan arızalar sistem çalışmasını önemli bir ölçüde etkiler. Bu motorlar genellikle çevrimdışı olarak belirli zamanlarda izlenir. Fakat bu yöntem hem maliyetli hem de fabrikada üretimin durmasına neden olur. Bu çalışmada asenkron motorlarda oluşan stator, rotor ve sonlandırıcı halka arızalarının gerçek zamanlı teşhisi için bir akıllı durum izleme yaklaşımı sunulmuştur. Stator arızalarının teşhisi için önerilen bulanık sistem üç faz akım sinyalinin büyüklüğünden faydalanmaktadır. Rotor ve sonlandırıcı halka arızaları ise negatif seçim tabanlı bağışık sistem algoritması ile teşhis edilmektedir. Donanımsal tasarım Altera Cyclone III FPGA (Sahada Programlanabilir Kapı Dizileri) kartı üzerinde gerçekleştirilmiştir.

**Anahtar kelimeler:** FPGA, bulanık mantık, negatif seçim algoritması, arıza teşhisi, asenkron motor.

### Abstract

In modern industrial systems, the overall system performance should be hold at a good level. In these systems, induction motor is one of major components and it constitutes a big part of work-power. The faults occurred in induction motors dramatically affect the system performance. These motors are generally monitored offline in a scheduled time. However, this method is both cost and it causes breakdown of the production in a factory. In this study, an intelligent condition monitoring approach is proposed to diagnose stator and rotor faults in real time. Fuzzy system which proposed to diagnose stator faults utilizes the magnitudes of three phase currents. Rotor and end-ring faults are diagnosed by negative selection based immune system algorithm. Hardware design was implemented on Altera Cyclone III FPGA.

**Keywords:** FPGA, Fuzzy logic, negative selection algorithm, fault diagnosis, induction motor.

### 1. Giriş

Endüstride kullanılan elektromekanik sistemler çalışma ortamlarındaki nem, toz ve aşınma gibi etkenlerden dolayı arızalara maruz kalabilirler [1]. Oluşan arızaların erken bir aşamada belirlenmesi, bileşenlerin çalışma durumlarının sürekli izlenmesine bağlıdır. Arıza ne kadar erken belirlenirse, bakım süreci o kadar erken başlatılabilir. Ayrıca maliyet ve

bakım zamanı da düşürülür. Asenkron motorlar endüstriyel uygulamaların %85'inde kullanılan enerji dönüşüm aygıtlarıdır [2]. Asenkron motorlar yalın yapıları ve sağlıklarından dolayı endüstride geniş yer bulmuşlardır. Fakat çalışma ortamlarından dolayı farklı arızalara maruz kalabilirler. Oluşan arızalar motorun stator, rotor ve mil yatağı bileşenleri ile ilgilidir. Stator ve rotor arızaları oluşan arızaların %28'ini oluşturur [3]. Bir asenkron motor arızası üretim yapan bir fabrikada üretimin aksamasına sebep olur. Bu durum maliyet, ürün kalitesi ve güvenlik üzerinde ciddi sonuçlar doğurur. Bu yüzden durum izleme ve elde edilen verim düşüklüklerine göre bakım işlemi bu motorlar için oldukça önemlidir.

Geçmişten günümüze kadar arıza teşhis ve durum izleme yöntemleri farklı şekillerde uygulanmıştır. Son yıllarda akım, titreşim, gerilim ve hız gibi ölçümlerin değerlendirilmesi yapılarak durum izleme algoritmaları geliştirilmiştir [3]. Özellikle kolay elde edilebilmesi ve bütün arıza türleri için kullanılabilmesinden dolayı akım sinyallerinin kullanımı yaygındır. Motor akım imza analizi arıza teşhisinde iyi sonuçlar vermektedir. Motor düşük yükte çalışırken arıza ile ilgili frekans bileşeni hat frekansına yakın çıktığından arızalı durumu ayırt etmek zorlaşır [4]. Arıza teşhisi için dalgacık yöntemi, park vektör dönüşümü, zaman serileri yöntemi gibi yöntemlerden faydalanılmıştır. Yüksüz durumlarda rotor çubuğu arızalarının teşhisi için ayrık dalgacık yöntemi kullanılmıştır [5, 6]. Kararlı durumdaki faz akım sinyali kullanılarak sürekli dalgacık değerlendirme yöntemi ile rotor arızaları belirlenmiştir [7]. İki boyutlu park vektör örüntüsündeki maksimum değişimlerin büyüklükleri temel bileşen çözümlemesi ile çıkarılarak rotor ve stator arızaları belirlenmiştir [8, 9]. Durum izleme ve arıza teşhis işlemlerinin otomatikleştirilmesi için işaret işleme sonucu elde edilen özellikler akıllı hesaplama teknikleri ile öğrenilerek farklı durumlar modellenilebilir. Sinyal işleme ile elde edilen özelliklere göre uygun bir akıllı hesaplama tekniği seçilmektedir. Arıza teşhisi ve durum izleme için destek vektör makinalar [5, 10], yapay sinir ağları [2, 11-12], bulanık mantık [13-14], yapay bağışık sistemler [9] ve Bayesian sınıflandırma [15] gibi akıllı teknikler kullanılmıştır. Arıza teşhisinde akıllı tekniklerin sinyal işleme yöntemleri ile birlikte kullanımı sonuçların yorumlanması için uzman bilgisi gerekliliğini ortadan kaldırmaktadır.

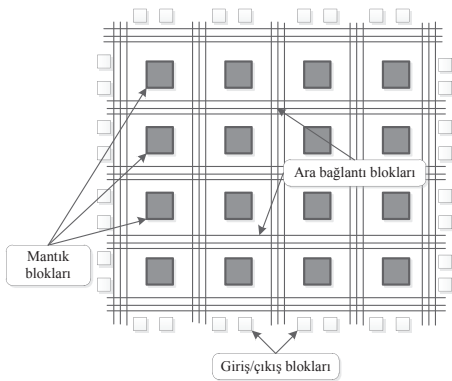
Fakat şimdiye kadar önerilen durum izleme ve arıza teşhis algoritmalarının büyük bir kısmı bilgisayar ortamında

gerçekleştirilmiştir. Bu tür bir durum izleme sistemi; bilgisayar, veri toplama kartı ve diğer çevre birimlerine gereksinim duyar. Ayrıca bu tür sistemler ile dönemsel aralıklar ile izleme işlemi yapılabilir. Fakat arızanın ne zaman oluşacağını kestirmek mümkün olmadığından dönemsel izleme kullanışlı değildir. FPGA ve DSP gibi sayısal işaret işlemcilerin gelişimi ile birlikte yerinde ve sürekli izleme ile arızalar daha hızlı teşhis edilebilir. Fakat bu işlemciler ile yapılan gerçek zamanlı durum izleme algoritmaları sınırlıdır. Geçiş akım sinyalleri ve dalgacık çözümü kullanılarak rotor arızalarının teşhisi için FPGA tabanlı bir yöntem sunulmuştur [16]. Fakat önerilen yöntem geçiş akım sinyallerini kullandığından arıza teşhisi için motorun her defasında yeniden başlatılması gerekmektedir. Faz akım sinyalinden hesaplanan entropi değeri, bulanık mantık ile değerlendirilerek FPGA tabanlı gerçek zamanlı bir arıza teşhis yöntemi sunulmuştur [2]. Önerilen yöntemin özellik çıkarım aşaması basit olmasına rağmen, farklı arıza durumları için elde edilen entropi değerleri birbirine yakındır. Ayrıca farklı yük durumları altında sağlam ve arızalı sinyallerin entropisinin değişimi belirtilmemiştir. Arıza teşhis, tespit ve tahmini için akıllı hesaplama tekniklerine dayalı yöntemler FPGA üzerinde gerçekleştirilmiştir [17, 18].

Bu çalışmada stator, rotor ve sonlandırıcı halka arızalarının eş zamanlı teşhisi için iki akıllı hesaplama tekniğinden faydalanılmaktadır. Stator arızalarının teşhisi için üç faz akım sinyalinin büyüklüğü bulanık sistem ile değerlendirilip arızalar belirlenmektedir. Negatif seçim algoritmasının eğitim aşamasında elde edilen detektörleri ile sına örnekleri karşılaştırılarak etkinleşen detektörlere göre kırık rotor çubuğu ve sonlandırıcı halka arızaları teşhis edilmektedir. Her iki yöntemin bütün aşamaları FPGA üzerinde gerçekleştirilmektedir.

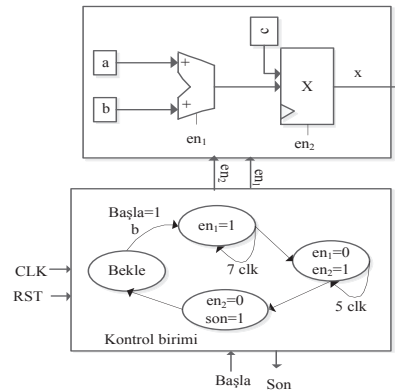
## 2. FPGA'da Algoritma Geliştirme

Programlanabilir devrelerin geçmişi, 1970'li yıllarda programlanabilir mantık dizilerin üretilmesine dayanır [19]. Özel mantık öbekler dizisine dayalı kapı dizileri tekniğinin gelişimi ile birlikte 1984'te Xilinx firması tarafından ilk FPGA geliştirildi [20]. Bir FPGA donanımı temel olarak yeniden ayarlanabilen mantık öbekler (CLBs) matrisi olarak tanımlanır. Bir FPGA G/Ç öbekleri, ara bağlantı öbekleri ve mantık öbekleri olmak üzere üç kısımdan oluşur. Şekil 1'de bir FPGA'nın genel yapısı gösterilmiştir.



Şekil 1: Bir FPGA'nın genel yapısı

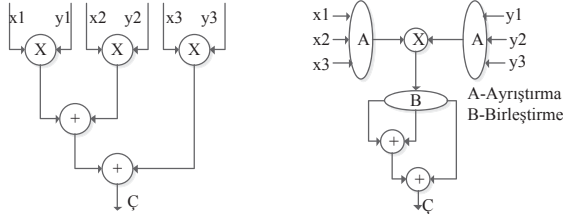
FPGA ortamında algoritma geliştirmek için VHDL ve Verilog gibi donanım tanımlama dilleri kullanılır. İki önemli FPGA üreticisi olan Altera ve Xilinx bu iki dili destekler. SOPC Builder tasarım aracı ile 32-bit Nios II işlemcisi ve çevre birimleri arasında bağlantıların gerçekleştirilmesi ve oluşturulan donanımın C programlama dili kullanılarak programlanması ile FPGA kartına yüklenebilir hale getirilir. MATLAB/SIMULINK tarafından desteklenen DSP Builder gibi araçlar kullanılarak SIMULINK ortamında tasarımlar geliştirilebilmektedir. Fakat bu şekilde yapılan tasarımlar FPGA kaynaklarını optimum şekilde kullanmazlar. Bu yüzden tasarımların VHDL veya Verilog gibi diller ile gerçekleştirilmesi daha hızlı çalışan tasarımların yapılmasını sağlar. Bu çalışmada Altera Cyclone III FPGA geliştirme kiti kullanılmıştır. Altera tasarımların gerçekleştirilmesi için Quartus II ortamını sağlamaktadır. Quartus II ortamında VHDL kodları ile tasarım yapılabildiği gibi öbek çizimler ile işlem yapılabilmektedir. VHDL tamamen sayısal çalışmaktadır. Bu yüzden reel sayılar üzerinde işlem yapmak için sayıların sabit noktalı veya IEEE-754 kayan noktalı biçimde ifade edilmesi gerekir. Sabit noktalı sayı biçimi hızlı ve kolay uygulanabilmesine rağmen, kayan noktalı sayı biçimi daha duyarlı işlemler için kullanılır. Quartus II ortamında 32 veya 64-bit kayan noktalı sayılar tanımlanıp kullanılabilir. Quartus II ortamı aynı zamanda kayan noktalı sayılar üzerinde birçok aritmetik işlemi de desteklemektedir. Her bir aritmetik işlem belirli saat çevrimi kadar sürede yapılmaktadır. Örneğin toplama işlemi 7 saat çevrimi süren çarpma işlemi 5 saat çevriminde hesaplanır. FPGA paralel çalışmayı desteklerken yapacağımız birçok uygulamada ardışık çalışan işlemlere de gereksinim duyarız. Örneğin  $x=a*b+c$  işlemi FPGA'da gerçekleştirilirken önce a ile b çarpılmalı daha sonra elde edilen sonuç c ile toplanmalıdır. Quartus II ortamında her bir işlemin çalışma sırası VHDL'de yazılan sonlu durum makinası ile denetlenebilir. Şekil 2'de bu örnek için öbek diyagramı ve sonlu durum makinası verilmiştir.



Şekil 2: FPGA'da ardışık işlemler

Tasarımlar karmaşıktıkça FPGA kaynakları yetersiz kalabilir. Bu yüzden tasarımcılar FPGA kaynaklarını en uygun şekilde kullanabilmek için uygulamasını kendisi kodlayarak geliştirmelidir. FPGA tabanlı mimarinin en iyileştirilmesi A<sup>3</sup> yöntemine dayalıdır [19]. Bu yöntemin temeli minimum sayıda operatör kullanarak paralel şekilde maksimum işlemin gerçekleştirilmesine dayanır. Bu yöntemde iki grafiksel gösterim kullanılır: veri akış çizgesi ve

ayrıştırılmış veri akış çizgesi. Şekil 3'te bu grafiksel gösterimler verilmiştir.



(a) Veri akış çizgesi

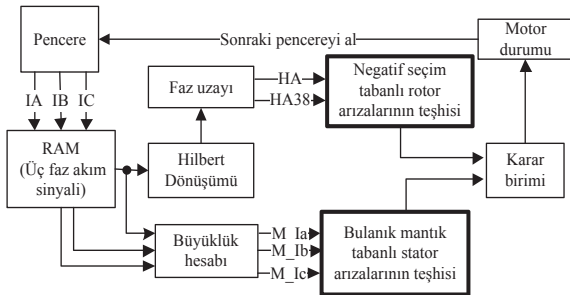
(b) Ayrıştırılmış veri akış çizgesi

Şekil 3: A³ yöntemindeki mimari en iyileştirme şekilleri [19]

Şekil 3'te verilen mimariler iki aritmetik operatör içeren  $C = x_1.y_1 + x_2.y_2 + x_3.y_3$  denkleminin gerçekleştirilmesidir. Şekil 3 (a) paralel çalışan üç adet çarpma işlemi ve ardışık çalışan iki adet toplayıcıdan ibarettir. Bu mimari paralel çalışan kısımlardan dolayı hızlı çalışmasına rağmen her bir işlem için ayrı bir FPGA kaynağı kullanmaktadır. Yapılan tasarım karmaşıklıkla birlikte kullanılan FPGA kaynakları da daha fazla olmaktadır. Şekil 3 (b)'de verilen mimaride ise bir adet çarpıcı ve iki adet toplayıcı kullanılmaktadır.

### 3. Eş zamanlı Arıza Teşhisi için FPGA Tabanlı Akıllı Durum İzleme Yöntemleri

Erken bir aşamada arıza teşhisi için önerilen akıllı teknikler FPGA üzerinde gerçekleştirilmektedir. Birinci yöntem RAM'den okuduğu tek faz akım sinyali üzerinden elde ettiği özellik sinyalinin faz uzayını negatif seçim algoritması ile değerlendirerek rotor ve sonlandırıcı halka arızalarını belirlemektedir. İkinci yöntem ise üç faz akım sinyalinin büyüklüklerini bulanık sistem ile değerlendirerek stator arızalarını belirlemektedir. İki algoritmadan gelen sonuçlara göre motor durumu karar biriminde değerlendirilerek karakter LCD'de gösterilir. Önerilen eş zamanlı arıza teşhisi yönteminin öbek çizimi Şekil 4'te verilmiştir.

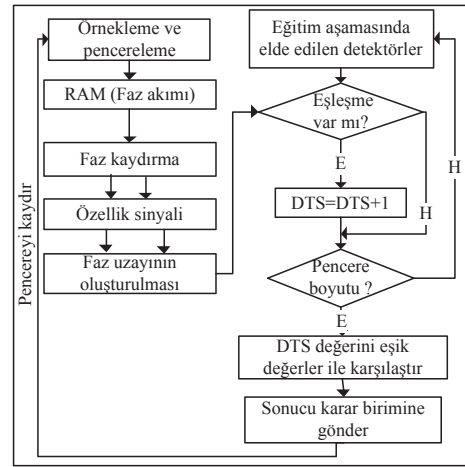


Şekil 4: FPGA tabanlı akıllı durum izleme yönteminin öbek çizimi

Şekil 4'te VHDL dilinde yazılan pencere fonksiyonu ile üç faz akım sinyali RAM'den okunmaktadır. Okunan üç fazın büyüklüğü hesaplanarak bulanık sisteme verilmekte ve stator arızaları teşhis edilmektedir. Tek faz akım sinyali üzerinde uygulanan Hilbert dönüşümünün faz uzayı oluşturularak rotor ve sonlandırıcı halka arızaları negatif seçim algoritması ile belirlenmektedir. Karar birimi algoritmaların çıktılarına göre motor durumunu LCD göstergede göstermektedir.

### 3.1. Rotor ve Sonlandırıcı Halka Arızalarının Teşhisi için Bağışık Sistem Algoritması

Rotor ve sonlandırıcı halka arızaların otomatik olarak belirlenmesi için negatif seçim algoritmasından faydalanılmıştır. Negatif seçim algoritması sağlam veri örneklerini kullanarak problemi öğrenir ve arızalı durumları teşhis edebilen detektörler üretir. Üretilen detektörler sınamaya aşamasında kullanılarak arızalar belirlenebilir. Önerilen negatif seçim algoritması; özellik çıkarımı ve teşhis aşamalarından oluşur. Önerilen yöntemin öbek diyagramı Şekil 5'te verilmiştir.



Şekil 5: Gerçek zamanlı bağışık arıza teşhisi sistemi

Şekil 5'te RAM 'den okunan akım sinyali örnekleri için özellik sinyali oluşturulmakta ve oluşturulan sinyalin faz uzayı lineer olmayan zaman serileri yöntemi ile elde edilmektedir. Faz uzayında alınan bir örnek ile herhangi bir detektör arasında eşleşme var ise detektör teşhis sayısı (DTS) arttırılmaktadır. Pencere boyutuna ulaşıldığında ise teşhis edilen örnek sayısına göre motor durumu gösterilmektedir. İşlem bir sonraki pencere alınarak gerçek zamanlı bir şekilde devam ettirilmektedir.

#### 3.1.1. Özellik Çıkarımı

Faz akım sinyali RAM'den okunmaktadır. RAM'den akım sinyallerini okumak ve faz kaydırmasını gerçekleştirmek için VHDL'de yazılan pencere fonksiyonu kullanılmaktadır. Yazılan VHDL kodunun bir kısmı Şekil 6'da verilmiştir.

```

If (clk'event and clk='1') then
  If(enb='1') then
    adr1 := adr1 + 1;
    adr2:=adr2+1;
    If(adr1=pen+1410) then
      pen:=pen+100;
      adr1:=pen;
      adr2:=pen+89;
    End if;
  End if;
End if;

```

Şekil 6: Pencere ve faz kaydırma için VHDL kodu

Şekil 6'da adr1 değişkeni RAM'den okunacak birinci adresi adr2 ise faz kaydırma için kullanılacak ikinci adresi gösterir.

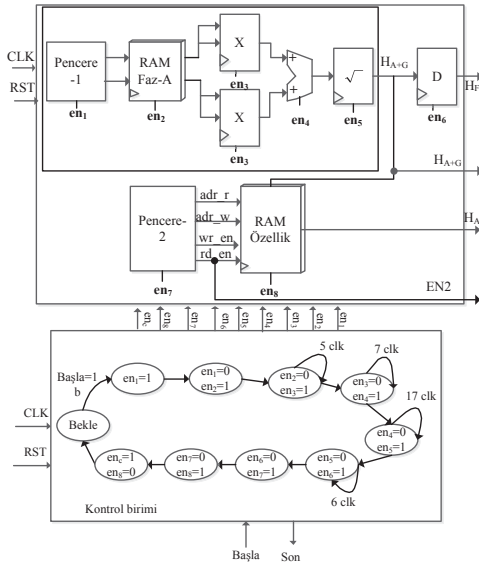
Her bir pencere boyutu 1500 adet örnekten oluşmaktadır. Pencere boyutuna ulaşıldığında pen değişkeni 100 artırılarak bir sonraki pencere için veriler okunmaktadır. Özellik sinyali, tek faz akım sinyali üzerinden elde edilmektedir. Bu sinyal tek faz akım sinyaline Hilbert dönüşümü uygulanması ile oluşturulur. Sinüs bir sinyalin Hilbert dönüşümünün iki karmaşık bileşeni olup, biri sinyalin kendisi diğeri ise orijinal sinyalin  $90^\circ$  faz kaydırılmış halidir. Dolayısıyla ROM'dan veri okunurken bu faz kaydırması göz önüne alınmıştır. Faz kaydırması ile oluşturulan iki bileşene Hilbert dönüşümünün uygulanması aşağıda verilmiştir.

$$H(I) = I[k] + jI_{S\_90}[k] \quad (1)$$

Denklem (1)'den de görüldüğü gibi Hilbert dönüşümü iki karmaşık bileşenden oluşur. Bu bileşenlerin büyüklüğü hesaplanarak özellik sinyali denklem (2)'ye göre elde edilmektedir.

$$|H_A[k]| = \sqrt{I[k]^2 + I_{S\_90}[k]^2} \quad (2)$$

RAM'den okunan ve aralarında faz farkı olan iki örnek alındıktan sonra özellik sinyali oluşturulur. Elde edilen özellik sinyalinin her bir örneği zaman gecikmeli faz uzayını oluşturmak için ikinci bir pencere bloğu ile başka bir RAM'e yazılmaktadır. Önerilen özellik çıkarma yönteminin öbek çizimi Şekil 7'de verilmiştir.



Şekil 7: Özellik sinyalinin elde edilmesi

Şekil 7'de öncelikle pencere-1 fonksiyonu ile RAM'den faz akım örneği ve faz kaydırması oluşturulmuş örnek okunmaktadır. Okunan iki örnek için özellik sinyali denklem (2)'ye göre hesaplanmaktadır. Hesaplanan özellik sinyali ikinci pencere fonksiyonu yardımıyla ikinci RAM'e yazılırken aynı zamanda sabit noktalı sayıya dönüştürülerek (D bloğu) benzetim esnasında grafiksel gösterim sağlanmaktadır. Yazılan örnek sayısı faz uzayı için belirlenen zaman gecikmesinde ulaşıldığında ikinci pencere bloğu ikinci RAM'den okumayı (rd\_en) etkinleştirir. Örneğin zaman gecikmesi 10 seçilmiş ise hesaplanan özellik sinyalinin örnekleri ikinci RAM'e yazılmakta ve ikinci pencere fonksiyonu ile yazılan örnek sayısı denetlenmektedir. RAM'e yazılan örnek sayısı 10 olduğunda rd\_en etkinleştirilerek yazılan örneklerin ilkinden itibaren okuma işlemi

gerçekleştirilir. Dolayısıyla iki boyutlu faz uzayı için elimizde ilk örnek ile onuncu örnek olmaktadır. İkinci RAM'den okuma işlemi etkinleştirilen sinyal, aynı zamanda okunan faz uzayı örneklerini detektörler ile karşılaştırmak için negatif seçim bloğunu (EN2) etkinleştirir. Önerilen yöntemde pencere fonksiyonu dışında bütün işlemler 32-bit kayan noktalı sayı biçimine göre yapılmaktadır. Seri çalışan öbeklerin çalışma sıralarının belirlenmesi için VHDL dilinde yazılan sonlu durum makinası tabanlı bir denetleme bloğu kullanılmaktadır. Özellik vektörü elde edildikten sonra, bu vektörün faz uzayı oluşturulur. Faz uzayı lineer olmayan zaman serileri yöntemine dayalıdır. Faz uzayı bir zaman serisini farklı boyutlarda haritalar. Bir  $X_t$  zaman serisi için faz uzayında bir nokta aşağıdaki gibi verilebilir.

$$X_t = (x_{t+b(\tau-1)}, x_{t+b(\tau-2)}, \dots, x_t) \quad (4)$$

Bu denklemde  $b$  gömülme boyutunu ve  $\tau$  ise zaman gecikmesini gösterir. Zaman gecikmesi  $\tau$  ardışık olmayan zaman gecikmeli örnekler üzerinde faz uzayının oluşturulmasını sağlar. Örneğin  $\tau=3$  ve  $b=4$  olduğunda  $X_t = (x_{t+8}, x_{t+4}, x_t)$  faz uzayında bir noktaya karşılık gelir.

### 3.1.2. Arıza Teşhisi: Negatif Seçim Algoritması

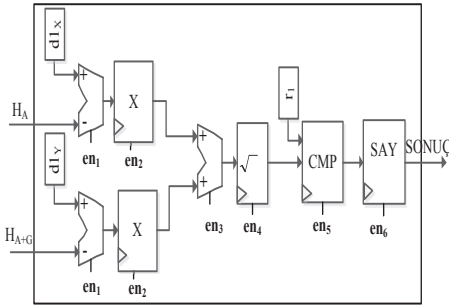
Doğal bağışık sistemi, insan vücudunu yabancı hücreler olarak bilinen antijen ve patojenlerden koruyan etkili bir mekanizmadır [21]. Virüs, bakteri, mantar ve parazitler gibi mikroorganizmalar patojen olarak adlandırılır ve bunlar vücuda girdikten sonra hastalığa sebep olurlar. Bağışık sistemlerindeki temel problem, bu patojenlerin tanınması ile ilgilidir. Patojenlerin antijen gibi bazı küçük molekülleri, bağışık sistem tarafından tanınabilir. Doğal bağışık sistemde vücuda giren bu patojen veya antijenleri tanıyan sistem negatif seçim olarak adlandırılır. Bu kural dışı davranışlara karşılık doğal bağışıklık davranışını model alan negatif seçim algoritması ilk kez Forrest ve diğ. [22] tarafından bilgisayar güvenliğine uyarlandı. Negatif seçim algoritmasının eğitim aşaması temel olarak üç adımdan oluşup aşağıdaki gibi tanımlanabilir.

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

Orijinal negatif seçim algoritmasının eğitimindeki temel problem aynı detektörlerin bir sonraki aşamada üretilmesi olasılığıdır. Negatif seçim algoritmasının eğitimi genetik algoritma ile gerçekleştirilerek en uygun detektörlerin elde edilmesi sağlanmıştır [23]. Detektör ile sına veri örneği arasındaki eşleşme için kullanılan Öklid uzaklığı aşağıda verilmiştir.

$$B = \sqrt{\sum_{i=1}^L (K_i - D_i)^2} \quad (5)$$

Denklem (5)'te L veri örneğinin boyutunu gösterir. Elde edilen B mesafesi bütün detektörlerin yarıçapı ile karşılaştırılarak teşhis işlemi gerçekleştirilir. Şekil 8'de faz uzayında bir örnek ile bir detektör arasındaki eşleşme hesabı için oluşturulan öbek diyagramı verilmiştir.



Şekil 8: Negatif seçim algoritması teşhis aşaması.

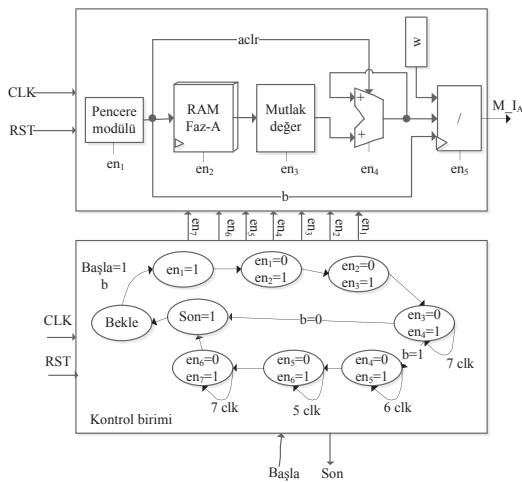
Eğer hesaplanan Öklid uzaklığı detektörün yarıçapından küçük ise VHDL’de yazılan SAY bloğu teşhis edilen detektör sayısını bir arttırmaktadır. Eğer herhangi bir detektör eşleşme teşhis ederse SAY bloğu DTS’yi bir arttırır.

### 3.2. Bulanık Mantık Tabanlı Stator Arızalarının Teşhisi

Stator ile ilgili arızaların gerçek zamanlı olarak teşhis edilmesi için bulanık mantık tabanlı bir yöntem önerilmiştir. Önerilen yöntem arıza teşhisi için üç faz akım sinyalinin büyüklüğünü bulanık sistemin girişleri olarak alır. Bulanık sistemin çıkışı motor durumu hakkında bilgi vermektedir. Bulanık sistem, özellik çıkarımı ve bulanık uygulama aşamalarından oluşur. Her iki aşama da FPGA üzerinde gerçekleştirilmektedir.

#### 3.2.1. Özellik Çıkarımı

Okunan ardışık sinyallerin mutlak değeri alınarak elde edilen pozitif örnekler pencere boyutuna ulaşıncaya kadar üst üste toplanmaktadır. Elde edilen toplam değeri pencere boyutuna bölünerek her bir fazın ayrı ayrı büyüklüğü hesaplanmaktadır. Bulanık sistem üç faz büyüklüğünü alarak motor durumu hakkında bir sonuç elde eder. Tek faz akım sinyalinin büyüklüğünün hesaplanması için oluşturulan tasarımın öbek çizimi Şekil 9’da verilmiştir.



Şekil 9: Bir faz akımının büyüklüğünün hesaplanması

Şekil 9’da pencere bloğu ile okunan her bir örneğin mutlak değeri hesaplanmakta ve örnekler üst üste toplanarak okunan penceredeki örnek sayısına ulaşıldığında toplam pencere boyutuna bölünmektedir. Böylece ilgili penceredeki akım sinyallerinin büyüklüğü hesaplanmaktadır. Bulanık sistem

kısmı VHDL ortamında yazılmış olup tam sayı biçiminde çalışmaktadır. Bu yüzden elde edilen akım büyüklükleri 100 ile çarpılıp tam sayıya dönüştürülerek bulanık sisteme verilmektedir.

#### 3.2.2. Bulanık Sistem Tabanlı Stator Arızalarının Teşhisi

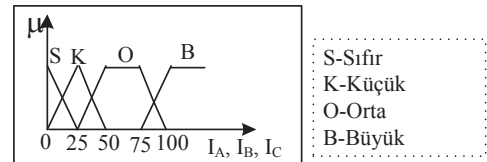
Bulanık sistemin her bir bloğu saat sinyali dışında bir etkinleştirme girişine sahiptir. Bulanık sistemin giriş ve çıkışlarını belirleyen VHDL kodu aşağıdaki gibidir.

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

Bulanıklaştırma bloğu üyelik fonksiyonlarını kullanarak giriş değerlerini bulanık değerlere dönüştürür. Önerilen bulanık sistem tam sayı biçiminde çalıştığı için bulanıklaştırma işlemi eğime göre hesaplanmaktadır [24]. Örneğin S üyelik fonksiyonunun eğimi  $1/25=0.004$ 'tür. Bu değer 1000 ile çarpılarak bir tamsayı değere dönüştürülür. Bulanıklaştırma için yazılan VHDL kodunun bir kısmı aşağıdaki gibidir.

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

Önerilen bulanık sistemin üyelik fonksiyonları ve kural tabanı Şekil 10’da verilmiştir. Bulanık sistemin kural tabanı Tablo 1’de verilmiştir.



Şekil 10: Üyelik fonksiyonları

Tablo 1: Kural tabanı

Kural No	Girişler			Çıkış
	IA	IB	IC	
1	S	-	-	BA
2	-	S	-	BA
3	-	-	S	BA
4	B	-	-	BA
5	-	B	-	BA
6	-	-	B	BA
7	K	K	O	KA
8	K	O	O	KA
9	O	K	O	KA
10	K	O	K	KA
11	O	K	K	KA
12	O	O	K	KA
13	K	K	K	S
14	O	O	O	S

Çıkarım mekanizması olarak Mamdani Min-Max yöntemi kullanılmıştır. Örneğin Kural-7 "IF IA is K and IB is K and IC is O then BC is KA" kuralı için  $C^7 = \min[K_{IA}, K_{IB}, O_{IC}]$  ile verilir. Tasarlanan bulanık sistem 3 sınıfa sahip olduğundan kuralları birleştirmek için aşağıdaki işlem uygulanmaktadır [24].

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

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

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

Durulama aşaması ağırlıklı ortalama yöntemine göre yapılmıştır. Bu yöntem denklem 6'da verilmiştir.

$$y = \frac{\sum \mu(z_i) \times z_i}{\sum \mu(z_i)} \quad (6)$$

Durulandırma işlemi için yazılan VHDL kodunun PROCESS bloğu Şekil 11'de verilmiştir.

```

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

```

Şekil 11: Durulandırma işlemi için PROCESS bloğu

Bütün uygulamalar Quartus II 10.1 SP1 ortamında VHDL donanım tanımlama dili ile gerçekleştirilmiştir [25]. Quartus II yazılımı aynı zamanda FPGA üzerinde tasarımın uygulanması ve çözümlenmesi için kullanılmıştır.

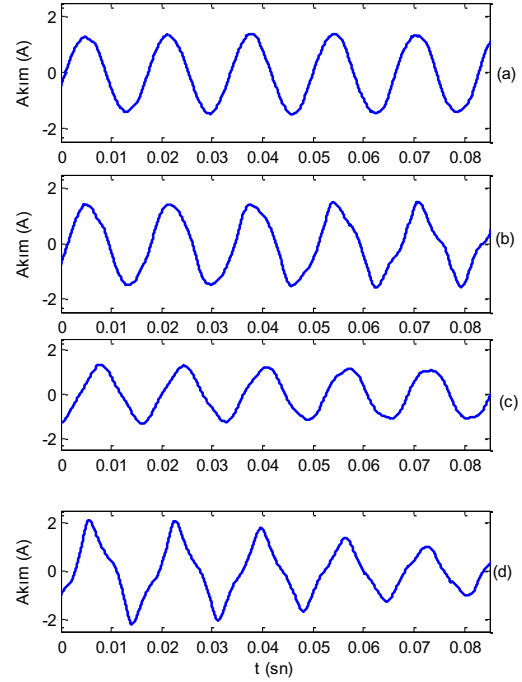
#### 4. Deneysel Sonuçlar

Önerilen eş zamanlı arıza teşhis yönteminin doğruluğu üç fazlı bir asenkron motordan alınan akım sinyalleri ile doğrulanmıştır. Sağlam, stator arızası, kırık rotor çubuğu ve sonlandırıcı halka arızasına sahip bir asenkron motordan alınan sinyaller ile özellik sinyalleri elde edilmektedir. FPGA uygulaması Quartus II 10.1 ortamında gerçekleştirilmiş olup benzetim sonuçları Altera Modelsim 6.5 ile elde edilmiştir [25]. Deneyde kullanılan motorun özellikleri Tablo 2'de verilmiştir.

Tablo 2: Deneyde kullanılan asenkron motorun özellikleri

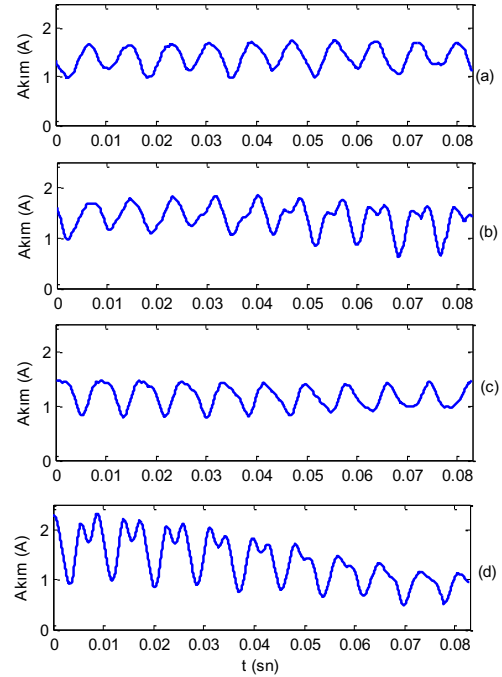
Özellik	Değer
Güç	0.37 kW
Giriş gerilimi	380 V
Tam yük akımı	1.2 A
Besleme frekansı	50 Hz
Kutup sayısı	4
Rotor çubuklarının sayısı	22
Tam yük hızı	1390

Şekil 12'de sağlam ve kırık rotor çubuğu arızası için akım sinyalleri ve özellik sinyalleri verilmiştir.



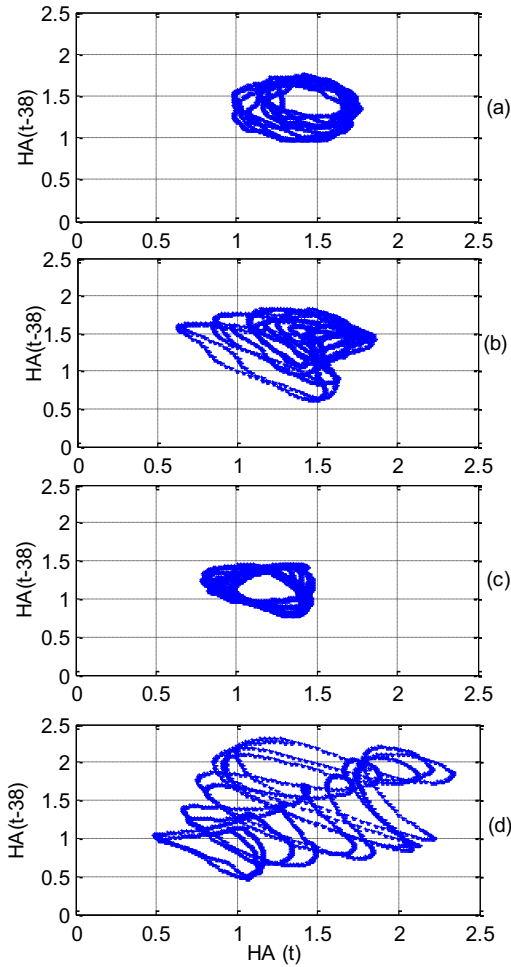
Şekil 12: Sağlam ve arızalı motor akım sinyalleri (a) Sağlam motor (b) Bir kırık rotor çubuğu arızası (c) İki kırık rotor çubuğu arızası (d) Sonlandırıcı halka arızası

Şekil 13'te sağlam, bir kırık rotor çubuğu ve bir kırık sonlandırıcı arızası için elde edilen özellik sinyalleri verilmiştir.



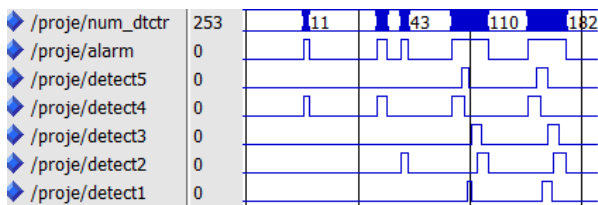
Şekil 13: Sağlam ve arızalı motor özellik sinyalleri (a) Sağlam motor (b) Bir kırık rotor çubuğu arızası (c) İki kırık rotor çubuğu arızası (d) Sonlandırıcı halka arızası

Orijinal sinyal ve bu sinyalin 90 derecelik bir faz kaydırması ile oluşturulan ikinci sinyalin kareleri toplanmakta ve toplamın karekökü alınarak negatif seçim için özellik sinyali oluşturulmaktadır. Özellik sinyali elde edildikten sonra faz uzayı oluşturularak detektörler üretilir. Faz uzayı için zaman gecikmesi 38 ve gömülme boyutu iki alınmıştır. Sağlam, bir kırık rotor çubuğu ve bir kırık sonlandırıcı arızası için faz uzayları Şekil 14’te verilmiştir.



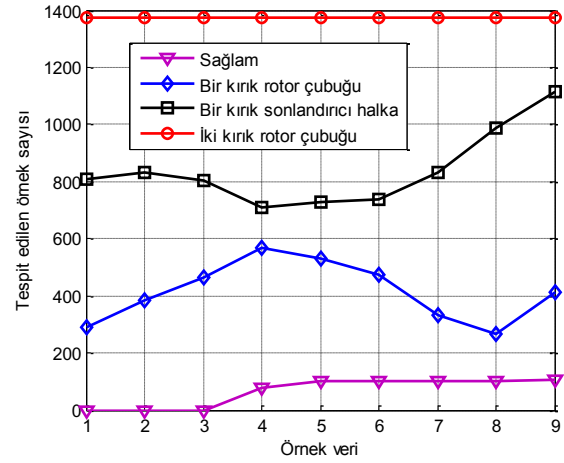
Şekil 14: Sağlam ve arızalı faz uzayları (a) Sağlam motor (b) Bir kırık rotor çubuğu arızası (c) İki kırık rotor çubuğu arızası (d) Sonlandırıcı halka arızası

MATLAB ortamında elde edilen detektörlerden öz olmayan uzayı en iyi kapsayanlar sına için kullanılacaktır. Sına veri kümesi detektörlerden herhangi birini etkinleştirirse DTS bir arttırılır. Bu durumu gösteren Modelsim benzetimi Şekil 15’te verilmiştir.



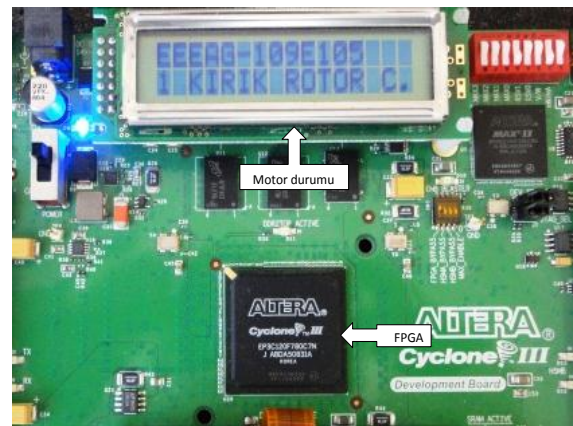
Şekil 15: Etkinleşen detektörlere göre DTS'nin değışimi

Şekil 15’te herhangi bir detektör bir sına örneğini tespit ederse num\_dtctr değışkeninin değeri bir arttırılmaktadır. Önerilen yöntemin başarımı farklı örnek sinyaller için değerlendirilmiştir. Toplam 30 adet veri örneği için sistemin başarımı değerlendirilmiştir. Şekil 16’da her bir durum için DTS değeri gösterilmiştir.



Şekil 16: Farklı durumlar için teşhis edilen örnek sayısı

Şekil 16’da ilk on veri örneği farklı sağlam durumları ifade etmektedir. Yönteme sağlam durum örnekleri verildiğinde maksimum 80 örnek detektörler tarafından teşhis edilir. Bir kırık rotor çubuğu arızasında ise minimum 250 örnek teşhis edilmektedir. Teşhis edilen örnek sayısı iki kırık rotor çubuğu arızasında okunan penceredeki örneklerin tamamına yakın olmaktadır. Her bir pencerenin sonunda detektörler tarafından teşhis edilen örnek sayısına göre motor durumu FPGA kartı üzerindeki LCD gösterge üzerinde gösterilmektedir. Şekil 17’de bir kırık rotor çubuğu arızası için yöntemin çalışması gösterilmiştir.

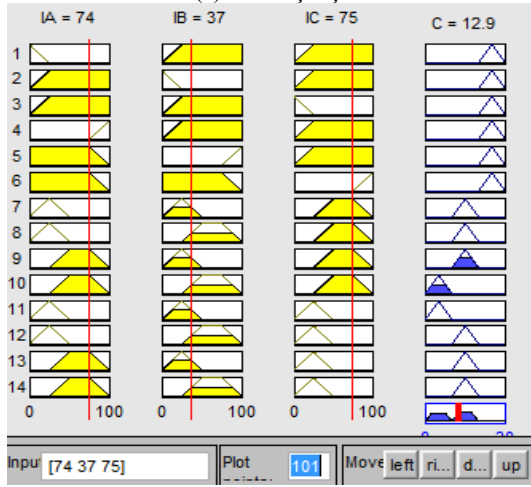


Şekil 17: Önerilen yöntemin FPGA uygulaması

Stator arızasına sahip motor için Quartus II ortamında ve MATLAB’da gerçekleştirilen deneysel sonuçlar Şekil 18’de verilmiştir.

+	/proje/acc_a	413C6646	42...42...42...	42AFBA32	00.
+	/proje/acc_b	3F6119DF	42...42...42...	42434286	00.
+	/proje/acc_c	4141120E	42...42...42...	42A9FD4	00.
+	/proje/imag...	3F3F74AB	3F3E582D	Intege(0.7436*100) → 74	3F.
+	/proje/imag...	3ED8949D	3EBEEA9F	Intege(0.3729*100) → 37	3E.
+	/proje/imag_c	3F369674	3F409528	Intege(0.7523*100) → 75	3F.
+	/proje/bc	15	15		
+	/proje/ba	0			
+	/proje/ka	1			
+	/proje/s	0			
+	/proje/incl0	1			

(a) FPGA çıkışı



(b) Matlab çıkışı

Şekil 18: Bulanık sistemin FPGA ve Matlab sonuçları

Şekil 18'da okunan akım sinyallerinden her bir örneğin mutlak değeri alınarak üst üste toplanmaktadır. Her bir fazın toplam ifadesi acc\_a, acc\_b ve acc\_c ile gösterilmiştir. Bölme sonucu elde edilen faz akım sinyallerinin büyüklükleri imag\_c, imag\_b ve imag\_a ile gösterilmiştir. Elde edilen bu büyüklükler 100 sabit değeri ile çarpılarak altf\_convert bileşeni ile tam sayıya dönüştürülmektedir. Dönüşüm sonucu elde edilen üç faz akım sinyalinin büyüklüğü bulanık sisteme verilerek arıza durumu hakkında bulanık sistem bir çıkış (bc) üretir. FPGA tabanlı bulanık sistem VHDL'de yazılıp tamsayı tabanlı çalışmaktadır. Bu yüzden Matlab Fuzzy Logic toolbox ile FPGA sonucu arasında bir hata değeri oluşmaktadır. Önerilen yöntem Cyclone III FPGA geliştirme kitinin çok az miktarda kaynağını kullanır. Gerçekleştirilen tasarımı kullandığı FPGA kaynaklarının oranı Tablo 3'te verilmiştir.

Tablo 3: FPGA kaynaklarının kullanım oranı

Hafıza birimi	NSA	BS	Toplam
Mantıksal elemanlar	% 15	% 10	% 25
Mantıksal kayıtlar	% 8	% 3	% 11
Hafıza bitleri	% 3	% 2	% 5
Gömülü çarpıcılar	% 15	% 13	% 28

Tablo 3'te bulanık sistem (BS) ve negatif seçim algoritmasının (NSA) kullandığı FPGA kaynaklarından en önemli kısmı gömülü çarpıcılardır. Çünkü IEEE-754 kayan noktalı sayılar üzerinde yapılan aritmetik işlemler bu kaynakları yoğun bir şekilde kullanmaktadır. Çalışmada kullanılan Cyclone III FPGA kartı 576 adet gömülü çarpıcıya

sahtir. Her iki algoritma toplamda 157 adet (%28) gömülü çarpıcıyı kullanmaktadır.

## 5. Sonuçlar

Bu çalışmada eş zamanlı arıza teşhisi için negatif seçim ve bulanık mantık algoritmalarını kullanan bir yöntem sunulmuştur. Önerilen yöntem arıza teşhisi için üç faz akım sinyalinin kullanır. Okunan akım sinyallerinden özellik çıkarımı ve iki akıllı hesaplama algoritmasının bütün aşamaları FPGA üzerinde gerçekleştirilmiştir. İki algoritmanın eş zamanlı çalışması sağlanarak birden çok arızanın gerçek zamanlı teşhisi sağlanmıştır. FPGA'nın paralel çalışabilme yeteneği sayesinde önerilen yöntem arıza teşhisi için düşük maliyetli çözümler sunar. Önerilen yöntemin başarımı Modelsim benzetim aracıyla doğrulanmıştır. Aynı zamanda tasarım Altera Cyclone III FPGA kartına yüklenerek çalıştırılmıştır.

## 6. Teşekkür

Bu çalışma Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TUBITAK) tarafından 109E105 numaralı proje ile desteklenmiştir.

## 7. Kaynaklar

- [1] Chow, M.Y., "Methodologies of Using Artificial Neural Network and Fuzzy Logic Technologies for Motor Incipient Fault Detection", World Scientific Publishing, Singapore, 1998.
- [2] Troncoso, R.J., Gallaga, R. S., Yopez, E.C., Perez, A.G., Rios, R. A. O., Salas, R.A., Vidales, H. M., Huber, N., "FPGA-Based Online Detection of Multiple Combined Faults in Induction Motors through Information Entropy and Fuzzy Inference", *IEEE Trans. On Industrial Electronics*, Vol. 58, No. 11, pp. 5263 – 5270, 2011.
- [3] Bellini, A., Filippetti, F., Tassoni, C., Capolino, G., "Advances in Diagnostic Techniques for Induction Machines", *IEEE Trans. Indus. Electr.*, Vol. 55, No. 12, pp. 4109-4126, 2008.
- [4] Panadero, R. P., Sanchez, M. P., Guasp, M. R., Folch, J. R., Perez, E. H., Cruz, J. P., "Improved Resolution of the MCSA Method via Hilbert Transform, Enabling the Diagnosis of Rotor Asymmetries at Very Low Slip", *IEEE Trans. on Energy Conv.*, Vol. 24, No. 1, pp. 52-59, 2009.
- [5] Briz, F., Degner, M. W., Garcia, P., Bragado, D., "Broken Rotor Bar Detection in Line-Fed Induction Machines using Complex Wavelet Analysis of Startup Transients", *IEEE Trans. on Industry Applications*, Vol. 44, No. 3, pp. 760-768, 2008.
- [6] Bouzida, A., Touhami, O., Ibtouen, R., Belouchrani, A., Fadel, M., Rezzoug, A., "Fault Diagnosis in Industrial Induction Machines through Discrete Wavelet Transform", *IEEE Trans. On Industrial Electronics*, Vol. 58, No. 9, pp. 4385 – 4395, 2011.
- [7] Tsoumas, I. P., Georgoulas, G., Mitronikas, E. D., Safacas, A. N., "Asynchronous Machine Rotor Fault Diagnosis Technique Using Complex Wavelets", *IEEE Trans. on Energy Conversion*, Vol. 23, No. 2, pp. 444-459, 2008.
- [8] Pires, V. F., Martins, J. F., Pires, A. J., "Eigenvector/eigenvalue Analysis of a 3D Current



- Referential Fault Detection and Diagnosis of an Induction Motor”, *Energy Conversion and Management*, Vol. 51, No. 5, pp. 901-907, 2010.
- [9] Chilengue, Z., Dente, J. A., Branco, P. J. C., “An Artificial Immune System Approach for Fault Detection in the Stator and Rotor Circuits of Induction Machines”, *Electric Power Systems Research*, Vol. 81, No. 1, pp. 158-169, 2011.
- [10] Aydin, I., Karakose, M., Akin, E., “A Multi-objective Artificial Immune Algorithm for Parameter Optimization in Support Vector Machine”, *Applied Soft Computing*, Vol. 11, No.1, pp. 120-129, 2011.
- [11] Ayhan, B., “Linguistic Rule Generation for Broken Rotor Bar Detection in Squirrel-Cage Induction Motors”, *PhD Thesis, North Carolina State University, Raleigh, NC, 2005.*
- [12] Su, H., Chong, K. T., “Induction Machine Condition Monitoring using Neural Network Modeling”, *IEEE Trans. on Industrial Electronics*, Vol. 54, No. 1, pp. 241–249, 2007.
- [13] Zidani, F., Benbouzid, M. E. H., Diallo, D. Nait-Said, M. S., “Induction Motor Stator Faults Diagnosis by a Current Concordia Pattern-Based Fuzzy Decision System”, *IEEE Trans. Energy Conversion*, Vol. 18, No. 4, pp. 469–475, 2003.
- [14] Rodriguez, P. V. J., Arkkio, A., “Detection of stator winding fault in induction motor using fuzzy logic”, *Applied Soft Computing*, Vol. 8, pp. 1112–1120, 2008.
- [15] da Silva, A. M., Povinelli, R. J., Demerdash, N.A.O., “Induction machine broken bar and stator short-circuit fault diagnostics based on three phase stator current envelopes”, *IEEE Trans. on Industrial Electronics*, Vol. 55, No. 3, pp. 1310-1318, 2008.
- [16] Moreno, A. O., Troncoso, R. J. R., Frias, J. A. V., Gillen, J. R., Perez, A. G., “Automatic Online Diagnosis Algorithm for Broken-Bar Detection on Induction Motors Based on Discrete Wavelet Transform for FPGA Implementation”, *IEEE Trans. on Industrial Electronics*, Vol. 55, No. 5, pp. 2193- 2202, 2008.
- [17] Akin, E., Aydin, I., Karakose, M., “FPGA Based Intelligent Condition Monitoring of Induction Motors: Detection, Diagnosis, and Prognosis”, *IEEE International Conference on Industrial Electronics (ICIT) & Southeastern Symposium on System Theory (SSST)*, March 14-17, Auburn, Alabama, USA, pp. 373-378, 2011.
- [18] Aydin, I., Karakose, M., Akin, E., “Arıza Teşhisi için Gerçek Zamanlı Bağışık Sistem Uygulaması”, *Otomatik Kontrol Türk Milli Komitesi 2011 Ulusal Toplantısı*, 13-14 Eylül, İzmir, Türkiye, 2011.
- [19] Monmasson, E., Idkhajine, L., Cirstea, M. N., Bahri, I., Tisan, A., Naouar, M. W., “FPGAs in Industrial Control Applications”, *IEEE Trans. on Industrial Informatics*, Vol. 7, No. 2, pp. 224-243, 2011.
- [20] Xilinx Staff, “Celebrating 20 Years of Innovation”, *Xcell Journal*, 48, 2004.
- [21] de Castro, L.N., Zuben, F. J. V., “Learning and Optimization using the Clonal Selection Principle”, *IEEE Trans. on Evolutionary Computation*, Vol. 6, No. 3, pp. 239-251, 2002.
- [22] Forrest, S., Perelson, A. S., Allen, L., Cherkuri, R., “Self-non-self-discrimination in a Computer”, *Proceedings of IEEE symposium on research in security and privacy*, May 16-18, Oakland, CA , USA, pp. 202–212, 1994.
- [23] Aydin, I., Karaköse, M., Akin, E., “Genetik Algoritma Kullanan Yapay Bağışık Sistem Tabanlı Arıza Teşhis Modeli”, *Dokuz Eylül Üniversitesi Fen ve Mühendislik Dergisi*, Cilt: 11, No: 31, s:57-72, 2009.
- [24] Barriga, A., Sanchez-Solano, S., Brox, P., Cabrera, A., Baturone, I., “Modeling and Implementation of Fuzzy Systems based on VHDL”, *International J. of Approximate Reasoning*, Vol. 41, pp. 164-178, 2006.
- [25] Altera Data Book, 2010. [Online]. Available: [www.altera.com](http://www.altera.com).