

VEKTÖR KONTROLLÜ ASENKRON MOTORUN HIZININ

YAPAY SINIR AĞLARI İLE TAHMİNİ

YÜKSEK LİSANS TEZİ

Mehmet DEDE

DANIŞMAN

Yrd. Doç. Dr. Yüksel OĞUZ

ELEKTRİK EĞİTİMİ ANABİLİM DALI

Haziran 2008

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

VEKTÖR KONTROLLÜ ASENKRON MOTORUN HIZININ
YAPAY SİNİR AĞLARI İLE TAHMİNİ

Mehmet DEDE

DANIŞMAN
Yrd. Doç. Dr. Yüksel OĞUZ

ELEKTRİK EĞİTİMİ ANABİLİM DALI

Haziran 2008

ONAY SAYFASI

Yrd. Doç. Dr. Yüksel OĞUZ danışmanlığında,
Mehmet DEDE tarafından hazırlanan
VEKTÖR KONTROLLÜ ASENKRON MOTORUN HIZININ
YAPAY SİNİR AĞLARI İLE TAHMİNİ

başlıklı bu çalışma, lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri
uyarınca
02/06/2008
tarihinde aşağıdaki jüri tarafından
Elektrik Eğitimi Anabilim Dalında
Yüksek Lisans tezi olarak oybirliği/oy çokluğu ile kabul edilmiştir.

	Ünvanı, Adı, SOYADI	İmza
Başkan	Prof. Dr. Süleyman TAŞGETİREN	
Üye	Prof. Dr. Hasan ÇİMEN	
Üye	Yrd. Doç. Dr. Yüksel OĞUZ	

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetin Kurulu'nun
...../...../..... tarih ve
..... sayılı kararıyla onaylanmıştır.

Doç. Dr. Zehra BOZKURT
Enstitü Müdürü

ÖZET
YÜKSEK LİSANS TEZİ
VEKTÖR KONTROLLÜ ASENKRON MOTORUN HIZININ
YAPAY SİNİR AĞLARI İLE TAHMİNİ

Mehmet DEDE

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Eğitimi Anabilim Dalı

Danışman: Yrd. Doç. Dr. Yüksel OĞUZ

Doğru akım (DA) motorlarından elde edilebilen yüksek dinamik performans, güç yarı iletkenleri, sayısal sinyal işleyicileri (DSP) ve kontrol tekniklerindeki gelişmelerle birlikte, asenkron motorlarda da elde edilebilir olmuştur. Asenkron motorlarda alan yönlendirmeli kontrolün kullanılmasıyla, DA motorlarda olduğu gibi, akı ve moment bağımsız olarak kontrol edilebilir.

Asenkron motorlar gibi sistemlerin genellikle lineer olmaması, lineer kontrol teorisinin başarılı uygulamalarını yapmak için büyük engel teşkil etmektedir. Doğal olarak lineer olmayan dönüşüm kabiliyetlerine sahip olan yapay sinir ağları (YSA), lineer olmayan tanıma ve kontrol problemlerine güçlü ve alternatif bir çözüm getirmiştir.

Bu tez çalışmasında, dolaylı vektör kontrollü asenkron motorun kontrolünde sensörsüz hız tahminci olarak YSA kullanılmıştır. DSP'li uygulama devresi üzerinde yapılan deneylerden elde edilen veriler incelenerek YSA hız öngörücünün (tahmincinin) gerçek hızı tahminindeki başarısı ortaya konmuştur.

2008, 72 sayfa

Anahtar kelimeler: Asenkron Motorlar, Vektör Kontrolü, Yapay Sinir Ağları, Hız Tahmini, DSP

ABSTRACT

M. Sc. Thesis

NEURAL NETWORKS BASED SPEED ESTIMATION OF VECTOR CONTROLLED ASYNCHRONOUS MOTORS

Mehmet DEDE

Afyon Kocatepe University
Graduate School of Natural and Applied Sciences
Department of Electric Education

Supervisor: Asst. Prof. Yüksel OĞUZ

High dynamic performance that can be obtained from direct current (DC) motors has also proved obtainable from asynchronous motors together with the advances in power semi-conductors, digital signal processors (DSP) and control techniques. With the use of field-oriented control in asynchronous motors, flux and torque can be controlled separately like in DC motors.

Because systems like asynchronous motors are generally non-linear, such systems are obstacles to the implementation of successful applications of linear control theory. Naturally, Artificial Neural Networks (ANN) that have non-linear transformation capability have offered a more preferable and alternative solution to non-linear identification and control problems.

In this study, ANN was used as sensorless speed observer in controlling indirect vector-controlled asynchronous motors. Data obtained from experiments on DSP application board were analyzed and in the end the success of ANN speed observer was evident.

2008,72 pages

Keywords: Asynchronous Motors, Vector Control, Artificial Neural Networks, Speed Observer, DSP

TEŐEKKÜR

Bu alıőmanın hazırlanmasında yardım ve desteęini esirgemeyen danıőmanım Sayın Yrd. Do. Dr. Yüksel OĖUZ'a, deęerli hocam Prof. Dr. Hasan İMEN'e ve ayrıca, bu alıőmanın hazırlanmasında büyük payı olan ve desteęini esirgemeyen deęerli hocam Yrd. Do. Dr. Seydi Vakkas ÜSTÜN'e sonsuz saygı ve teőekkürlerimi sunarım.

Hayatım boyunca ilgilerini, desteklerini ve sevgilerini her zaman yanımda hissettięim aileme çok teőekkür ederim.

Mehmet DEDE

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
SİMGELER VE KISALTMALAR DİZİNİ	ix
ŞEKİLLER DİZİNİ	xi
RESİMLER DİZİNİ	xiii
ÇİZELGELER DİZİNİ	xiv
1. GİRİŞ	1
2. GENEL BİLGİLER	4
2.1 Asenkron Motorlar	5
2.1.1 Asenkron motorun 3-faz modeli	5
2.1.2 Moment denkleminin makina değişkenleri ile tanımlanması	11
2.2 Alan Yönlendirmeli Vektör Denetim	13
2.2.1 Clarke dönüşümü	16
2.2.2 Park dönüşümü	17
2.2.3 Ters Park dönüşümü	18
2.2.4 Ters Clarke dönüşümü	18
2.2.5 Vektör kontrol yöntemleri	19
2.2.5.1 Doğrudan vektör kontrolü yöntemi	20
2.2.5.2 Dolaylı vektör kontrolü yöntemi	20
2.2.6 Asenkron motorun senkron referans çatı (d-q) dinamik modeli	21
2.3 Yapay Sinir Ağları'nın Teori ve Uygulaması	25
2.3.1 YSA'nın Tanımı ve Modeli	26
2.3.1.1 Nöronun biyolojik yapısı ve nöron modeli	26
2.3.2 YSA'nın yapısı ve işlem elemanı	28
2.3.2.1 Giriş İşareti Sınıfları	30
2.3.2.2 Ağ Tipleri	31
2.3.2.3 Eşik Fonksiyonları	32

2.3.2.4 Ağırlık Uzayı	33
2.4 Sayısal İşaret İşlemci (DSP)	35
2.4.1 DSP uygulama alanları	39
3. MATERYAL VE METOT	42
3.1 Materyal	42
3.1.1 Asenkron motor	43
3.1.2 DSP programlama ve motor kontrol kartı	43
3.1.3 dsPIC30F6010 DSP mikroişlemcisi genel özellikleri	44
3.1.4 Sistem yazılımının tanıtılması	46
3.2 Metot	47
3.2.1 Kontrol sistemi yazılımının tanıtılması	47
3.2.2 Vektör kontrollü asenkron motor verilerinin elde edilmesi	48
4. BULGULAR	50
4.1 Deney Sonuçları	50
4.2 YSA ile Sistemin Modellenmesi	53
5. TARTIŞMA VE SONUÇ	57
6. KAYNAKLAR	58
6.1 İnternet Kaynakları	60
7. ÖZGEÇMİŞ	61
8. EKLER	62
EK-1 Uygulamada kullanılan dsPIC30F6010 pinlerinin fonksiyonları	62
EK-2 Hızın değişimi sırasında anlık Iq ve Vq değişimine ait örnek grafikler	65
EK-3 Hızın değişimi sırasında Iq ve Vq ortalama değerlerinin değişimine ait örnek grafikler	66
EK-4 YSA eğitime çalışmalarının yapıldığı program	67

SİMGELER VE KISALTMALAR DİZİNİ

1. Simgeler

L	Endüktans
v	Volt
i	Akım
r	Direnç
λ, Ψ	Akı
θ, φ	Açı
W	Enerji
Te	Elektromanyetik moment
TL	Yük moment
J	Kalkınma moment
μ	Manyetik geçirgenlikleri
l	Uzunluk
i_q, i_d	d-q düzleminde stator akım bileşenleri
v_q, v_d	d-q düzleminde stator gerilim bileşenleri
i_α, i_β	α - β düzleminde stator akım bileşenleri
v_α, v_β	α - β düzleminde stator gerilim bileşenleri
f	Frekans
ω_e	Senkron hızı
ω_r	Rotor hızı
τ_r	Rotor zaman sabiti
P	Kutup sayısı
N_s/N_r	ASM dönüştürme oranı
X_0, X_1	YSA giriş verileri
w	YSA ağırlık verileri

2. Kısaltmalar

ASM	Asenkron motor
AA	Alternatif akım
DA	Dođru akım
MMK	Manyetomotor kuvvet
YSA	Yapay sinir ađları
DSP	Sayısal iřaret iřleyici
ADC	Analog-dijital çevirici
PWM	Darbe genişlik modülasyonu
SVM	Uzay vektör modülasyonu
MLP	Çok katmanlı algı
MSE	Ortalama karesel hata
FFT	Hızlı fourier dönüşümü
ALU	Aritmetik Mantık Ünitesi

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 2.1	Yıldız bağlı, 2 kutuplu, 3 fazlı bir asenkron makine sargısı	6
Şekil 2.2	Vektör kontrol dönüşümleri	15
Şekil 2.3	(a) 3-faz referans düzlemi, (b) iki faz referans düzlemi, (c) Döner referans düzlemi	16
Şekil 2.4	Clarke dönüşümü	16
Şekil 2.5	Park dönüşümü	17
Şekil 2.6	Ters Park dönüşümü	18
Şekil 2.7	Ters Clarke Dönüşümü	18
Şekil 2.8	Vektör Kontrolü blok diyagramı	20
Şekil 2.9	Senkron düzlem (a) q^e - ekseni ve (b) d^e - ekseni dinamik model eşdeğer devresi	21
Şekil 2.10	$d^e - q^e$ ve $d^s - q^s$ düzlemler arası dönüşüm	24
Şekil 2.11	$as - bs - cs$ ve $d^s - q^s$ düzlemine dönüşümler	24
Şekil 2.12	Basit bir nöron yapısı	27
Şekil 2.13	Nöron modeli	28
Şekil 2.14	YSA'nın genel blok şeması	28
Şekil 2.15	Genel işlem elemanı yapısı	29
Şekil 2.16	Komşu hücrelerin merkez hücreye etkisi	30
Şekil 2.17	Tetikleyici ve yasaklanan girişlere sahip bir işlem elemanı	30
Şekil 2.18	YSA sınıflandırıcılarının grafiksel gösterimi	31
Şekil 2.19	Sıkça kullanılan (a) Linear, (b) Rampa, (c) Basamak ve (d) Sigmoid fonksiyonları.	32
Şekil 2.20	Motor kontrol sistem birimleri	40
Şekil 3.1	Uygulama devresinin blok Şeması	42
Şekil 3.2	Program akış şeması	48
Şekil 4.1	Hıza göre anlık I_q ve V_q değişimi	50
Şekil 4.2	Ortalaması alınan (a) V_q ve (b) I_q parametrelerinin hıza göre değişimleri	51
Şekil 4.3	C++' de yazılan YSA eğitime programının akış diyagramı	54

Şekil 4.4	Eđitme sırasında hatanın deęiřimi	55
Şekil 4.5	Hızın YSA ile tahmin edilen deęerleri ile gerek deęerleri	56

RESİMLER DİZİNİ

	Sayfa No
Resim 3.1 MPLAB PM3 programlama cihazı	43
Resim 3.2 Motor kontrol geliştirme kartı	44

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 2.1	Bilgisayar ile insan beyni arasındaki çalışma sistem yapısının karşılaştırılması	26
Çizelge 3.1	Asenkron motorun etiket bilgileri	43
Çizelge 4.1	YSA'nın eğitimi için kullanılan veriler	52
Çizelge 4.2	YSA testinde kullanılan veriler	53
Çizelge 4.3	Hızın YSA ile tahmin edilen değerleri ile gerçek değerleri	56

GİRİŞ

Alternatif akım (AA) motorları dayanıklı, güçlü ve düşük maliyetlidir. Güç elektroniğindeki gelişmelerle birlikte, yüksek güçlü anahtarlama cihazları AA motorlarda uygulanmaya başlanmıştır. Sürücüsüyle beraber AA motorlar paket olarak, fazla değişken olmayan hız kontrol uygulamalarında doğru akım (DA) motorların yerine tercih edilmektedir. AA motorlar, DA motorlara göre maliyet avantajının yanı sıra, daha az bakım gerektiren, daha küçük boyutlu ve güvenilirliği yüksektir. Düşük hassasiyetli hız kontrolü yapılan yerlerde genel amaçlı sürücüler kullanılabilir. Bu sürücüler, geçici tepkili ve düşük hız performansı kritik olmayan mekanizmalarda kullanılabilirler. Limitli kontrol esnekliğine ihtiyaç duyulan tipik uygulama alanları fan, pompa ve kompresör olarak sınırlandırılmıştır (Peterson 1996).

DA ve AA motorlarda hız kontrol yapısında benzerlikler vardır. Asenkron motorların açısal hızları, şebekeden değişken frekans uygulanarak kontrol edilebilmektedir. Bu çözümde çok pahalı elektronik sistemlere ihtiyaç duyulmaktadır. Genel olarak, çok fazla AA motor kontrol yapısı vardır. Bazıları DA motor karşılığı olanlar kadar hızlı tepki verememektedir. Bazı mekanizmalar, DA motorların baskın olduğu, bugüne kadar AA motorların pek başarılı olamadığı çok karışık gereksinimlere ve tasarıma sahiptir. Mikroelektronikte ve kontroldeki gelişmeler ile birlikte bu zorlukların üstesinden gelinebilmektedir. Hız, pozisyon, ivmelenme ve tork gibi duyarlı parametrelerin hassas kontrolü, AA motorların vektör kontrolü ile mümkündür. Vektör kontrollü AA motor sürücüleri yüksek performanslı uygulamalarda kademeli olarak DA sürücünün yerini almaya başlamıştır. Bu sürücülerin yapısı, tork kontrolü yapılan DA motor kavramına cevap verebilmektedir. Bu nedenle, uzay düzleminde alan akısı ve rotor manyetomotor kuvvet (MMK) arasındaki dikey açı olması için DA makinelerdeki sabit tutturulmuş komütatörlerin çalışma şekli, AA motorlarda akı ve torkun bağımsız kontrolü için, stator akımlarını rotor akısı referans alınarak ayarlanmasına benzetilmektedir.

Bu yenilik, Hasse (1969) ve Blaschke'nin Siemens Laboratuvarlarında (1972) çalışmasıyla başlamıştır. Vektör kontrolünde motor akımının akı ve tork bileşenleri

ayrılmıştır. Bu AA motorlarda hız - tork kontrolünü artırmıştır. Alan yönlendirmeli AA motorlar, DA motorlar ile eşleştirilebilmektedir (Marzi 2007).

Vektör kontrolünde, manyetik akı vektörüne paralel olan akım vektörü; manyetik akıyı oluşturan akımı, buna dik olan akım vektörü de momenti oluşturan akımı temsil etmektedir. Böylece, bu iki stator akım vektörü bileşeni ayrı kontrol edilerek asenkron makine de DA makinesi gibi kontrol imkanına kavuşmuştur.

Uygulamada farklı vektörel denetim yöntemleri kullanılmaktadır. Tüm yöntemlerdeki amaç, asenkron motor denklemlerinin bir doğru akım motor denklemlerine benzer hale getirilmesidir.

Son yıllarda AA motorların rotor hızlarının, akısının ve stator akımlarının ölçümünde önemli avantajlara sahip olunmuştur (Messaoudi 2006). Takogenerator veya enkoder kullanılarak motor hızı ölçülmektedir. Zorluklara rağmen, akı bobinleri ve akı sensörleri kullanılarak AA motorların akısı direkt ölçülebilmektedir. Bu hız ve akı sensörleri, mekaniksel sağlamlıkta azalmaya, maliyet artışına, gürültüye ve sistem güvenilirliğinde azalmaya neden olmaktadır. Ayrıca elektriksel gürültüye karşı özel önlemler gerektirmektedir (Jeon 2002). Ek olarak çok yüksek hızlı ve zorlu uygulamalarda bu sensörlerin yerleştirilmesi zor olmaktadır. Bundan dolayı stator akımları ve gerilimler ölçülerek, akı ve hızın tahmini yapılmaktadır (Varghese 1998).

Elektrik motorlarının kontrolünde en çok tahmini yapılan büyüklükler:

- Akı ve tork tahmini-vektör kontrolünde kullanılmaktadır.
- Pozisyon tabanlı hız ve ivmelenme tahmini-hız ve pozisyon kontrolünde kullanılmaktadır.
- Stator akımları ve gerilimleri ölçülerek yapılan hız ve pozisyon tahmini-sensörsüz hız kontrolünde kullanılmaktadır (Hovingh 2004).

Yapay Sinir Ağları (YSA), hız tahmininde olduğu gibi doğrusal olmayan sistemlerin denetiminde daha yoğun ilgi görmektedir. YSA ile ilgili çalışmalar, insan beyninin

işleviyle ilgili yeni kuramlar, modeller geliştirmek ve teorileri gerçek dünyadaki problemlere uygulamak gibi iki temel alana yönelmiştir. Bu iki araştırma alanı arasında karşılıklı bir ilişki olmakla beraber geleneksel yöntemlerle çözümü güç ve ekonomik olmayan problemlerin çözümünde YSA'nın uygulanması oldukça önem kazanmaktadır (Bingöl 2005).

Vektör kontrol tekniğinin gerektirdiği işlemleri çok hızlı bir şekilde yerine getirebilecek işlemcilerle yani sayısal işaret işleyicilere (DSP) ihtiyaç duyulmaktadır. DSP'lerin motor kontrol alanına göre geliştirilmesi ile kullanımı artmıştır. Özellikle motor kontrolü için üretilenlerde, analog-dijital çevirici (ADC), darbe genişlik modülasyonu (PWM) sinyallerini üretebilen birimler mevcuttur.

Bu tez çalışmasında, vektör kontrollü asenkron motorun kontrolünde ihtiyaç duyulan hız bilgisini algılayıcısız olarak tahmin edilmesi işlemi YSA ile sağlanmıştır. DSP uygulama devresi üzerinde yapılan çalışmadan elde edilen verilerle hız tahmincisinin performansı incelenmiştir.

Bu çalışmanın birinci bölümünde çalışmanın amacı anlatılmıştır. İkinci bölümde günümüze kadar yapılan çalışmalar tanıtılmış, asenkron motorlar (ASM), vektör kontrolü, YSA ve DSP hakkında genel bilgiler verilmiştir. Üçüncü bölümde kullanılan DSP programlama ve motor kontrol kartı tanıtılmış ve yapılan deneyler ile verilerin nasıl elde edildiği anlatılmıştır. Son bölümde ise YSA ile sistemin modellenmesi ve yapılan uygulama sonuçları açıklanmıştır.

2. GENEL BİLGİLER

Sincap kafesli asenkron motorun hız veya moment kontrolü doğru akım motoru ile kıyaslandığında asenkron motorun doğrusal olmayan yapısından dolayı kontrol ve dönüşüm algoritmaları oldukça karmaşık olmaktadır. Bu doğrusal olmayan yapıdan kurtulabilmek amacıyla farklı kontrol yöntemleri incelenmeye başlanmış ve ilk olarak Hasse (1969) ve Blaschke (1972) tarafından ortaya atılan vektörel kontrol ile bu dezavantaj da ortadan kalkmaya başlamıştır.

Temel olarak bütün alternatif akım motorlarına uygulanabilen bu yöntem doğrudan ve dolaylı alan yönlendirmeli kontrol olarak gerçekleştirilebilmektedir. Dolaylı alan yönlendirmeli kontrol (akı ileri besleme kontrolü), ilk olarak Hasse (1969) tarafından uygulanmıştır. Bu uygulama tarzında yönlendirme bilgisi, stator akımları ve hızını algılayarak yapılan işlemler sonucunda elde edilir. Dolaylı alan yönlendirme hesaplamalarında kayma bilgisi ise stator akımları üzerinden elde edilir. Doğrudan alan yönlendirmeli kontrol (akı geri besleme kontrolü), ilk olarak Blaschke (1972) tarafından uygulanmıştır. Bu yöntem uygulanırken, yönlendirme akısı hall etkili algılayıcılarının, sezici bobinlerin, stator geriliminin üçüncü harmoniğinin veya kademeli stator sargılarının kullanımı ile ya da stator akımları, gerilimi ve hız ölçümlerini kullanan gözlemleyiciler ile elde edilip kontrolde ve bu yöntemin temelini oluşturan ayrıştırma için gerekli olan dönüşümlerde kullanılmaktadır (Aydeniz 2005).

Joetten (1983) algılayıcısız vektör kontrolü konusunda çalışmalarında önemli katkıları olmuştur. Temel olarak zıt emk vektörü, rotor akı vektörünü 90° farkla izleyen bir vektördür. Joetten burada akı genliğinin diğer işaretlere göre çok daha yavaş değiştiği ön kabulünü yapmış ve bu yaklaşımla stator akımı, gerilimi ve kayma frekansı bilgilerini kullanarak rotor frekansı kestirimini yapmıştır.

Holtz (1993), asenkron motorların sensörsüz hız kontrolü üzerine çalışma yapmıştır. Terminal gerilim ve akım bilgilerini kullanarak düşük maliyetli ve orta performanslı gözlemci tasarlamıştır.

Vas (1998), pozisyon sensörsüz vektör kontrolü metotları üzerinde çalışma yapmıştır. Sensörsüz vektör ve direkt tork kontrolü üzerine de kitap yazmıştır.

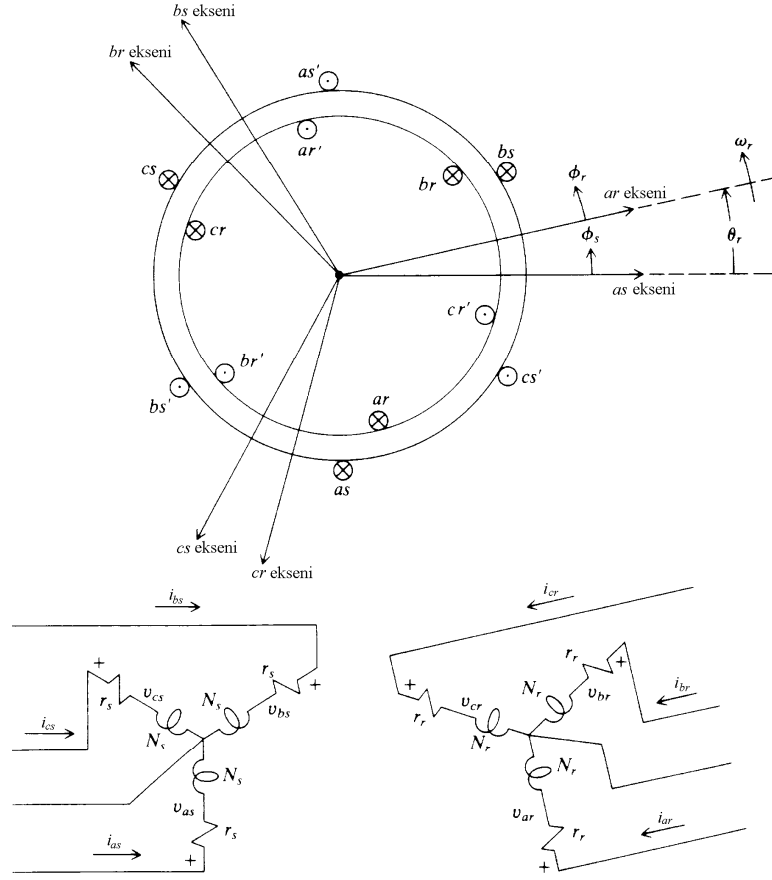
Lu, Hung ve Tsai (2000) asenkron motorların sensörsüz vektör kontrolü ile ilgili çalışma yapmışlardır. Çalışmalarında, rotor akısının tahmininde akı gözlemcisi olarak YSA kullanılmış ve hız tahminin performansında artış sağlamışlardır.

2.1 Asenkron Motorlar

Endüstriyel uygulamalarda asenkron motor (ASM) modellerinden özellikle üç fazlı ve rotoru sincap kafesli olan modelleri daha çok kullanılmaktadır. DA makinelerin dinamik davranış karakteristikleri ASM'lerden daha iyi olmasına rağmen ASM'ler DA makinelerine göre yapı itibariyle daha basittir. ASM'ler sağlam, güvenilir, maliyeti düşük, az bakım gerektiren, çevresel koşullardan etkilenmeme ve büyük güçte imal edilebilmelerinden dolayı daha çok tercih edilmektedir. Bunun yanında tehlikeli ve kirli ortamlarda da kullanılabilir (Bingöl 2005).

2.1.1 Asenkron motorun 3-faz modeli

Üç fazlı, yıldız bağlı, iki kutuplu, simetrik bir asenkron motorun sargı düzenlemesi Şekil 2.1'de gösterilmiştir. Stator sargılarının sarım sayıları ve dirençleri birbirinin aynıdır. Rotor faz sargıları da eşit sarım sayısı ve aynı direnç değerine sahiptir. Asenkron makinelerinin hava aralığı düzgün (her yerde aynı), stator ve rotor sargıları sinüzoidal olarak dağıtılmıştır.



Şekil 2.1 Yıldız bağlı, iki kutuplu, üç fazlı bir asenkron makine sargısı (Krause 1986).

Birçok uygulamada asenkron makineler 3 fazlı kaynaktan beslenen, rotoru kısa devre sargılı bir asenkron motor olarak kullanılmaktadır. Asenkron motorlarda stator sargılarından geçen dengeli üç fazlı akımlar hava aralığında dönen bir MMK oluşturmaktadır. Bu MMK'nın dönme hızı stator akımlarının frekansı ve stator kutup sayısı ile oranlıdır. Eğer rotor hızı, MMK'nın hızından farklı ise, dengeli üç fazlı rotor akımları kısa devre rotor sargılarında indükleneyecektir. Bundan dolayı bu makineler indüksiyon makine denmektedir. Rotor akımının frekansı, rotor hızı ile MMK hızı arasındaki farka bağlıdır. İndüklenen rotor akımları ve hava aralığı MMK'nın dönme hızı aynıdır. Bunlar, birbirine uyumlu bir şekilde ve aynı hızda dönen iki manyetik kutup gibi düşünülebilir. Bu iki kutbun karşılıklı etkisi ile moment (tork) üretilir. Eğer rotor ve hava aralığı MMK dönme hızı aynı ise (senkronizm anı) rotor akımı

indüklenmeyecek ve bundan dolayı moment üretilmeyecektir. Yani asenkron makinelerde senkron hızda moment üretilemez. Stator özendüktansları (2.1) de ifade edildiği gibi eşittir.

$$\begin{aligned} L_{asas} &= L_{bsbs} = L_{cs cs} \\ L_{asas} &= L_{ls} = L_{ms} \end{aligned} \quad (2.1)$$

(2.1) ifadesindeki L_{ls} kaçak endüktans, L_{ms} mıknatıslama endüktansıdır. Stator sargıları arasındaki ortak endüktanslar (2.2)'de ifade edildiği gibi eşittir.

$$L_{asas} = -\frac{1}{2}L_{ms} \quad (2.2)$$

Rotor özendüktansları da (2.3)'de gösterildiği gibi eşittir.

$$L_{arar} = L_{lr} + L_{mr} \quad (2.3)$$

Rotor sargıları arasındaki ortak endüktans ise (2.4)'de ifade edildiği gibidir.

$$L_{arbr} = -\frac{1}{2}L_{mr} \quad (2.4)$$

Stator ile rotor sargıları arasındaki ortak endüktans (2.5 - 2.7)'de ifade edilmiştir.

$$\begin{aligned} L_{asar} &= L_{bsbr} = L_{cs cr} \\ L_{asar} &= L_{sr} \cos\theta_r \end{aligned} \quad (2.5)$$

$$\begin{aligned} L_{asbr} &= L_{bscr} = L_{cs ar} \\ L_{asbr} &= L_{sr} \cos\left(\theta_r + \frac{2\pi}{3}\right) \end{aligned} \quad (2.6)$$

$$\begin{aligned} L_{as cr} &= L_{bs cr} = L_{cs br} \\ L_{as cr} &= L_{sr} \cos\left(\theta_r - \frac{2\pi}{3}\right) \end{aligned} \quad (2.7)$$

Yukarıdaki denklemlerde L_{sr} stator ve rotor sargıları arasındaki ortak endüktansın tepe değeridir. Fiziksel büyüklükler ile (2.8)'deki gibi tanımlanabilir.

$$L_{ascr} = \left(\frac{N_s}{2}\right) \left(\frac{N_r}{2}\right) \frac{\pi\mu_0 r l}{g} \quad (2.8)$$

Asenkron makinenin endüktansları tanımlandıktan sonra Şekil 2.1'deki makinenin gerilim denklemleri (2.9-2.14)'deki gibi ifade edilebilir.

$$\begin{aligned} v_{as} &= r_s i_{as} + \frac{d\lambda_{as}}{dt} \\ v_{bs} &= r_s i_{bs} + \frac{d\lambda_{bs}}{dt} \\ v_{cs} &= r_s i_{cs} + \frac{d\lambda_{cs}}{dt} \\ v_{ar} &= r_r i_{ar} + \frac{d\lambda_{ar}}{dt} \\ v_{br} &= r_r i_{br} + \frac{d\lambda_{br}}{dt} \\ v_{cr} &= r_r i_{cr} + \frac{d\lambda_{cr}}{dt} \end{aligned} \quad (2.9 - 2.14)$$

Burada r_s, r_r stator ve rotorun faz dirençleridir. Stator manyetik akısı bir faz için (2.15)'de ifade edilmiştir.

$$\lambda_{as} = L_{asas} i_{as} + L_{asbs} i_{bs} + L_{ascs} i_{cs} + L_{asar} i_{ar} + L_{asbr} i_{br} + L_{ascr} i_{cr} \quad (2.15)$$

Denklemlerden görüldüğü gibi, asenkron makinenin gerilim denklemleri stator ve rotor arasındaki ortak endüktansın zamanla değişmesi yüzünden çok karışıktır. Gerilim denklemleri (2.9 - 2.15) matris formunda, (2.16-2.17)'deki gibi yazılabilir.

$$v_{abcs} = r_s i_{abcs} + p\lambda_{abcs} \quad (2.16)$$

$$v_{abcr} = r_r i_{abcr} + p\lambda_{abcr} \quad (2.17)$$

Burada s ve r stator ve rotor değişken ve parametrelerini ifade etmektedir. r_s ve r_r diyagonal matrisleridir.

$$r_s = \begin{bmatrix} r_s & & \\ & r_s & \\ & & r_s \end{bmatrix} \quad r_r = \begin{bmatrix} r_r & & \\ & r_r & \\ & & r_r \end{bmatrix} \quad (2.18)$$

Manyetik akı denklemleri matris formundan (2.19) elde edilir.

$$\begin{bmatrix} \lambda_{abcs} \\ \lambda_{abcr} \end{bmatrix} = \begin{bmatrix} L_s & L_{sr} \\ (L_{sr})^T & L_r \end{bmatrix} \begin{bmatrix} i_{abcs} \\ i_{abcr} \end{bmatrix} \quad (2.19)$$

Sargı endüktansları (2.20-2.22)'de ifade edilmiştir.

$$L_s = \begin{bmatrix} L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L_{ls} + L_{ms} \end{bmatrix} \quad (2.20)$$

$$L_r = \begin{bmatrix} L_{lr} + L_{mr} & -\frac{1}{2}L_{mr} & -\frac{1}{2}L_{mr} \\ -\frac{1}{2}L_{mr} & L_{lr} + L_{mr} & -\frac{1}{2}L_{mr} \\ -\frac{1}{2}L_{mr} & -\frac{1}{2}L_{mr} & L_{lr} + L_{mr} \end{bmatrix} \quad (2.21)$$

$$L_{sr} = L_{sr} \begin{bmatrix} \cos\theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\theta_r \end{bmatrix} \quad (2.22)$$

Gerilim denklemlerinin stator referans alınarak (2.23-2.25)'deki gibi yazılması, analiz için daha uygun olmaktadır. Bunun için dönüştürme oranı (N_s/N_r) kullanılacaktır.

$$i'_{abcr} = \frac{N_r}{N_s} i_{abcr} \quad (2.23)$$

$$v'_{abcr} = \frac{N_s}{N_r} v_{abcr} \quad (2.24)$$

$$\lambda'_{abcr} = \frac{N_s}{N_r} \lambda_{abcr} \quad (2.25)$$

Mıknatıslama ve ortak endüktanslar aynı manyetik akı yolu ile ilgili olduklarından, (2.26-2.30)'daki gibi olur.

$$L_{ms} = \frac{N_s}{N_r} L_{sr} \quad (2.26)$$

$$L'_{sr} = \frac{N_s}{N_r} L_{sr} \quad (2.27)$$

$$L'_{sr} = L_{ms} \begin{bmatrix} \cos\theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) \\ \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\theta_r & \cos\left(\theta_r + \frac{2\pi}{3}\right) \\ \cos\left(\theta_r + \frac{2\pi}{3}\right) & \cos\left(\theta_r - \frac{2\pi}{3}\right) & \cos\theta_r \end{bmatrix} \quad (2.28)$$

$$L_{mr} = \left(\frac{N_r}{N_s}\right)^2 L_{ms} \quad (2.29)$$

$$L'_r = \left(\frac{N_r}{N_s}\right)^2 L_r \quad (2.30)$$

Denklem (2.19)'dan yararlanarak L'_r yeniden (2.31)'de ifade edildiği gibi yazılabilir.

$$L'_r = \begin{bmatrix} L'_{lr} + L_{ms} & -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & L'_{lr} + L_{ms} & -\frac{1}{2}L_{ms} \\ -\frac{1}{2}L_{ms} & -\frac{1}{2}L_{ms} & L'_{lr} + L_{ms} \end{bmatrix} \quad (2.31)$$

$$L'_{lr} = \left(\frac{N_s}{N_r}\right)^2 L_{lr} \quad (2.32)$$

Manyetik akılar, (2.33)'deki gibi açıklanabilir.

$$\begin{bmatrix} \lambda_{abcs} \\ \lambda'_{abcr} \end{bmatrix} = \begin{bmatrix} L_s & L'_{sr} \\ (L'_{sr})^T & L'_r \end{bmatrix} \begin{bmatrix} i_{abcs} \\ i_{abcr} \end{bmatrix} \quad (2.33)$$

Gerilim denklemleri statora göre, (2.34)'deki gibi yazılabilir;

$$\begin{bmatrix} v_{abcs} \\ v'_{abcr} \end{bmatrix} = \begin{bmatrix} r_s + \rho L_s & \rho(L'_{sr}) \\ \rho(L'_{sr})^T & r'_r + \rho L'_r \end{bmatrix} \begin{bmatrix} i_{abcs} \\ i'_{abcr} \end{bmatrix} \quad (2.34)$$

Burada

$$r'_r = \left(\frac{N_s}{N_r}\right)^2 r_r \quad (2.35)$$

$$\rho = \frac{d}{d_t} \quad (2.36)$$

2.1.2 Moment denkleminin makine değişkenleri ile tanımlanması

Ortak manyetik alanda depolanan enerji (2.37)'de ifade edilmiştir.

$$W_f = \frac{1}{2} (i_{abcs})^T (L_s - L_{ls} \mathbf{I}) i_{abcs} + (i_{abcs})^T L'_{sr} i'_{abcr} + \frac{1}{2} (i'_{abcr})^T (L'_r - L'_{lr} \mathbf{I}) i'_{abcr} \quad (2.37)$$

Burada \mathbf{I} birim matristir. Makinenin manyetik olarak doğrusal olduğu kabul edildiği için alan enerjisi ile koenerji eşit olur. P kutup sayılı bir makinede, bir mekanik girişe göre mekanik enerjideki değişim (2.38)'de verilmiştir.

$$dW_m = -T_e d\theta_{rm} \quad (2.38)$$

Burada T_e elektromanyetik moment olup motor olarak çalışma (moment çıkışı) için pozitifdir, (θ_{rm}) rotor pozisyonundaki gerçek yer değiştirmedir. Sargı akıları, akımları, alan ve koenerji ifadeleri elektriksel açısal yer değiştirmenin (θ_{rm}) fonksiyonu olarak tanımlanabilir.

θ_r ile θ_{rm} arasındaki bağıntı (2.39)'de ifade edilmiştir.

$$\theta_r = \left(\frac{P}{2}\right) \theta_{rm} \quad (2.39)$$

θ_r : Elektriksel açısal hız (rad / s)

P : Kutup sayısı

θ_{rm} : Mekanik açısal hız

$$dW_m = -T_e \left(\frac{2}{P}\right) d\theta_r \quad (2.40)$$

Elektromanyetik Moment (2.41-2.42)'de ifade edilmiştir.

$$T_e = \left(\frac{P}{2}\right) (i_{abcs})^T \frac{\partial}{\partial \theta_r} [L'_{sr}] i'_{abcr} \quad (2.41)$$

$$T_e = -\left(\frac{P}{2}\right) L_{ms} \left[\left[i_{as} \left(i'_{ar} - \frac{1}{2} i'_{br} - \frac{1}{2} i'_{cr} \right) + i_{bs} \left(i'_{br} - \frac{1}{2} i'_{ar} - \frac{1}{2} i'_{cr} \right) + i_{cs} \left(i'_{cr} - \frac{1}{2} i'_{br} - \frac{1}{2} i'_{ar} \right) \right] \sin \theta_r + \frac{\sqrt{3}}{2} \left[i_{as} (i'_{br} - i'_{cr}) + i_{bs} (i'_{cr} - i'_{ar}) + i_{cs} (i'_{ar} - i'_{br}) \right] \cos \theta_r \right] \quad (2.42)$$

Moment ile rotor hızı arasındaki ilişki (2.43)'de verilmiştir.

$$T_e = J \left(\frac{2}{P}\right) \rho \omega_r + T_L \quad (2.43)$$

Burada T_L yük momenti (Nm), J ise kalkınma (atalet) momenti (kgm^2).

(2.43) denkleminde rotor hızı (2.44)'de verilmiştir.

$$\rho \omega_r = \frac{d}{dt} \omega_r = \left(\frac{P}{2}\right) \left(\frac{T_e - T_L}{J}\right) \quad (2.44)$$

Rotor pozisyonu ise (2.45)'deki gibidir.

$$\theta_r = \int_0^t \omega_r dt + \theta_{r(\phi)} \quad (2.45)$$

Rotor hızı da (2.46)'daki gibi olur (Krause 1986).

$$\omega_r = \frac{d\theta_r}{dt} \quad (2.46)$$

2.2 Alan Yönlendirmeli Vektör Denetim

Vektör kontrolünde; eksen dönüşümleri ile stator akımının moment ve akıyı oluşturan bileşenleri birbirlerinden dekopple edilerek, geçiş tepki karakteristikleri hızlandırılmış ve bir doğru akım makinesinin karakteristiklerine benzetilmiştir (Bodson 2002).

Burada amaç makinenin kontrol büyüklükleri olan gerilimin genliği ve frekansı yanında fazını da göz önüne alarak bir kontrol yöntemi geliştirmektir. Burada d-q modelindeki doğru akım ve gerilimlerden oluşan denklemlerde rotor akısı sabit tutularak doğru akım makinesinin davranışına varmak mümkündür.

İlke olarak, alan akısı vektörünün stator akımının bir bileşeni doğrultusunda yönlendirilmesi nedeniyle bu yöntem "Alan Yönlendirilmeli Kontrol" olarak da bilinmektedir (Bose 2006).

Sincap kafesli asenkron motorlar basit, sağlam, bakım gerektirmeyen ve tüm güçlerde en ucuz motorlardır. Ayrıca, doğru akım motorlarında komütatör fırça düzeninin neden olduğu elektriksel ve mekanik gürültü, süreli bakım zorunluluğu ve patlayıcı ortamlarda çalışamama gibi sakıncalar, asenkron motorlarda yoktur. Bu nedenle, vektör kontrollü asenkron motorlar, mikroelektronik dalındaki hızlı gelişmelerin yardımı ile gelecekte duyarlı servo kontrol uygulamalarında doğru akım motorlarının kullanımını gereksiz kılacaktır. Yüksek dinamik performans gerektirmeyen uygulamalarda dahi, güvenilirlik ve enerji tasarrufu açısından, klasik yöntemlere tercih edilecektir (Vas 1990).

Vektör kontrol yöntemi ilk kez Blaschke tarafından uygulanmış ve Leonhard ve Bose başta olmak üzere çeşitli ülkelerden araştırmacılar bu yöntemin gelişmesine katkıda bulunmuşlardır.

Stator akımının moment ve akı bileşenlerine ayrıştırılması sırasında, referans eksenlerinin: mıknatıslama akısı, rotor akısı, stator akısı veya rotor üzerine çakıştırılması seçeneklerine göre, başlıca dört ayrı tip vektör kontrol yöntemi vardır (Vas 1990).

Ayrıca, dönüşümlerde kullanılan birim vektörlerini elde etme şekline göre, dolaylı ve dolaysız kontrol yöntemleri tanımlanmaktadır. Bu yöntemlerin uygulanmasında karşılaşılan başlıca sorunlar: Akı ölçümlerinin yeterince hassas olmaması, kontrol sisteminin motor parametrelerine bağlı olması ve sonuçta akım bileşenlerinin dekaple olma özelliğinin bozulmasıdır (Bose 2006).

Üç fazlı asenkron motorun stator ve rotor fazları arasındaki magnetik kuplaj nedeniyle, dinamik davranışının üç eksen sisteminde modellenmesi, zamanla değişen katsayılı diferansiyel denklemlerle mümkün olmakta ve çok karmaşık bir model yapısı ortaya çıkmaktadır. Bu nedenle, dengeli üç fazlı frekans dönüştürücüsünden beslenen bir asenkron motorun dinamik davranışı, d - q eksenlerinden oluşan iki eksenli sistemde modellenir (Hancock 1974).

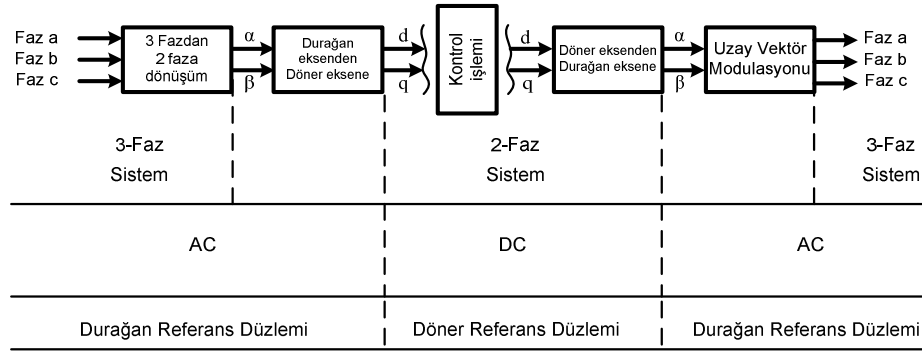
Bu sistemde, zamanla değişen parametreler yok edilmekte, tüm parametre ve değişkenler birbirinden dekaple olan ortogonal d ve q eksenleri üzerinde tanımlanmaktadır. Makinenin dinamik modeli, durağan veya döner eksen sistemlerinde tanımlanabilir. Durağan eksen sisteminde, d^s ve q^s referans eksenleri, statora göre sabit konumdadır. Döner eksen sisteminde ise, d^s ve q^s referans eksenleri, rotor hızında veya senkron hızda dönerler. Senkron hızda dönen eksen sistemi modelinin avantajı, sinüzoidal uyarılar altında, sürekli hal değişkenlerinin zamana göre sabit olmalarıdır (Şahin 2006).

Genel olarak vektör kontrolü aşağıdaki adımlarla gerçekleştirilmektedir.

1. Motor akım ve gerilim büyüklüklerinin ölçülmesi.
2. Bu büyüklüklerin Clarke Dönüşümü uygulanarak ($\alpha - \beta$) 2 faz sistemine aktarılması.
3. Rotor akı vektörü ve açısız pozisyonun hesaplanması.

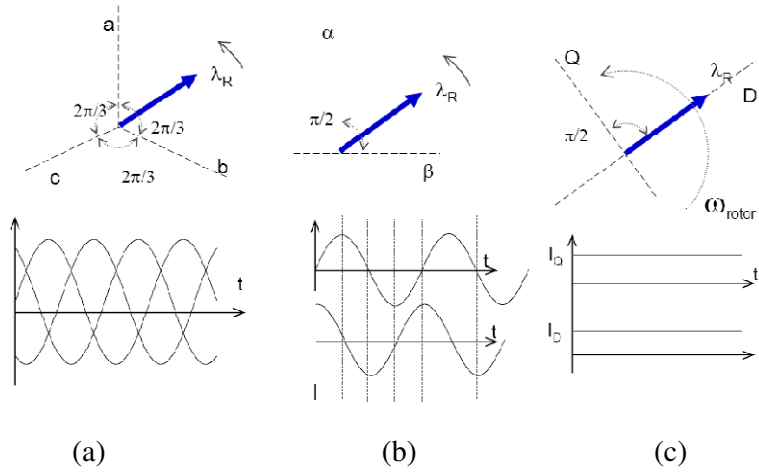
4. Park dönüşümünü kullanarak Stator akımları d, q referans düzlemine aktarılması.
 5. Stator akımlarından moment bileşeni (i_{sq}) ve akı bileşeni (i_{sd}) ayrı ayrı kontrol edilmesi.
 6. Referans gerilim değerleri v_q ve v_d hesaplanması.
 7. Ters Clarke dönüşümü kullanılarak v_a, v_b, v_c elde edilmesi.
 8. Uzak vektör modülasyonu (SVM) kullanılarak üç faz çıkış gerilimi üretilmesi.
- (Sustek 2007)

Vektör kontrolündeki işlem sırasında kullanılan dönüşümler, görsel olarak Şekil 2.2’de verilmiştir.



Şekil 2.2 Vektör kontrol dönüşümleri (Sustek 2007).

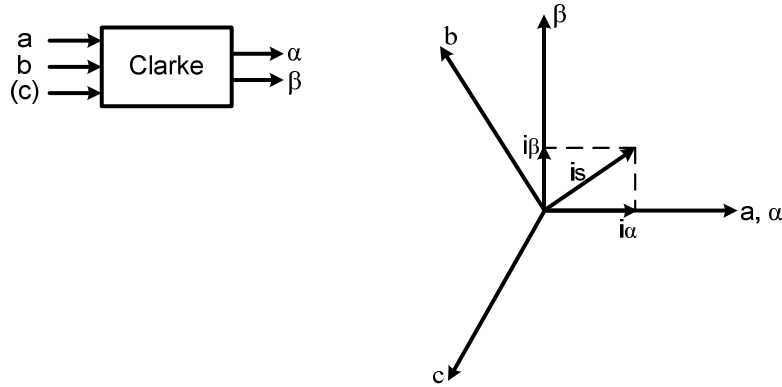
Daha önce oluşturulan ve asenkron motorun dinamik performansını analiz etmek için kullanılan gerilim denklemlerinde, asenkron motorun bazı endüktanslarının zamanla değiştiği görülmüştür. Yani bu endüktanslar rotor devrinin bir fonksiyonuydu. Diferansiyel denklemlerdeki katsayılar rotorun durması haricinde zamanla değişmektedir. Değişkenlerin başka değişkenlerle değiştirilmesi yani yeniden tanımlanması ile diferansiyel denklemlerin karmaşıklığı azaltılabilir. Bu işlem transformasyon (dönüşüm) olarak tanımlanmaktadır. Genel dönüşümlerde makinenin gerçek değişkenleri başka bir referans düzleminde tanımlanabilir. Bu düzlem genellikle keyfi referans düzlemdir. Bilinen bütün dönüşümler, bu genel dönüşümlerden elde edilebilir. Bunun için sadece kullanılacak referans düzlemin döner hızının bilinmesi yeterli olmaktadır. Şekil 2.3’de referans düzlemlerine ait vektörel gösterimler verilmiştir.



Şekil 2.3 (a) 3-faz referans düzlemi, (b) iki faz referans düzlemi, (c) döner referans düzlemi.

2.2.1 Clarke dönüşümü

Uzay vektörü (α - β) denilen iki eksenle başka bir referans düzleminde yazılabilir. a eksenini ve α eksenini Şekil 2.4'te verilen vektör diyagramında aynı yönde varsayarak gösterebiliriz.



Şekil 2.4 Clarke dönüşümü.

Üç fazlı sistemi iki boyutlu (α - β) sisteme çeviren izdüşümü (2.47-2.48)'de verilmiştir.

$$i_a = i_a \quad (2.47)$$

$$i_{\beta} = \frac{1}{\sqrt{3}}i_a + \frac{2}{\sqrt{3}}i_b \quad (2.48)$$

2.2.2 Park dönüşümü

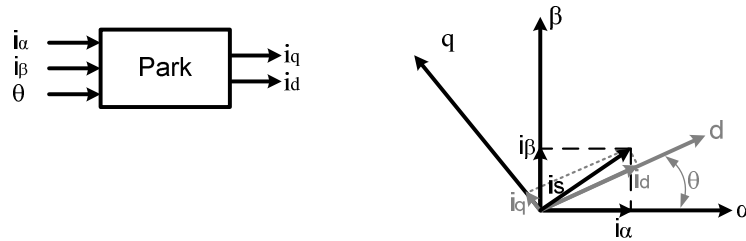
Bu dönüşüm vektör denetiminin en önemli kısmıdır. Gerçekte, bu izdüşümü d-q dönen referans düzleminde iki fazlı bir sisteme dönüştürür. d ekseninin rotor akısıyla uyarlandığını düşünürsek, diyagram akım vektörü için iki referans düzlemi arasındaki ilişkiyi göstermektedir (Şekil 2.5).

Burada θ rotor akı pozisyonudur. Akım vektörünün akı ve moment bileşenleri (2.49-2.50) denklemleri ile ifade edilmiştir.

$$i_d = i_{\alpha}\cos\theta + i_{\beta}\sin\theta \quad (2.49)$$

$$i_q = -i_{\alpha}\sin\theta + i_{\beta}\cos\theta \quad (2.50)$$

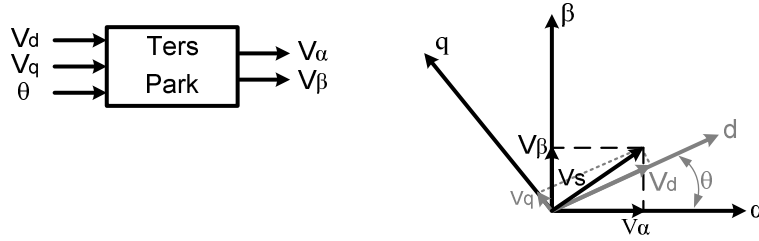
Bu bileşenler akım vektörlerindeki (α - β) bileşenlerine ve rotor akı pozisyonuna bağlıdır. Doğru rotor akı pozisyonu bilinirse, bu izdüşümde d-q elemanları sabit olur. Buradan, i_d (akı bileşeni) ve i_q (moment bileşeni) ile direkt moment denetimi mümkün ve kolay olan iki koordinatlı zamandan bağımsız bir sistem elde edilmiştir.



Şekil 2.5 Park dönüşümü.

2.2.3 Ters Park dönüşümü

Kontrol işleminden sonra d, q düzleminde elde edilen iki gerilim vektörlerini (v_d, v_q) üç faz motor gerilimine dönüştürmek için önce iki eksenli keyfi referans düzleminde iki eksenli durağan referans düzlemine (v_α, v_β) geçirmeliyiz. Bunun için önce Ters Park dönüşümü (2.51-2.52) kullanılır (Şekil 2.6).



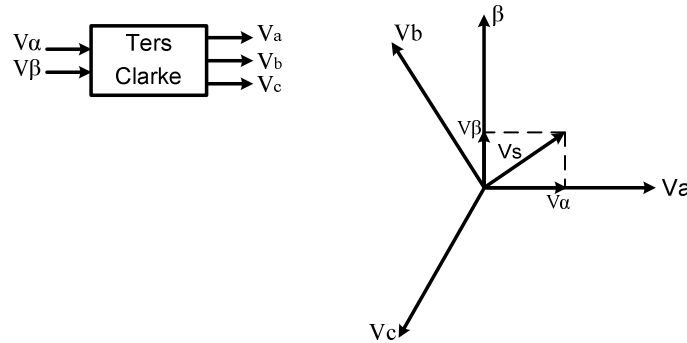
Şekil 2.6 Ters Park dönüşümü.

$$V_\alpha = V_d \cos\theta - V_q \sin\theta \quad (2.51)$$

$$V_\beta = V_d \sin\theta + V_q \cos\theta \quad (2.52)$$

2.2.4 Ters Clarke dönüşümü

Bir sonra ki adım iki eksenli durağan referans düzleminde (v_α, v_β) statorun üç faz referans düzlemine (v_a, v_b, v_c) geçilmesidir. Bunun için de Ters Clarke dönüşümü (2.53-2.55)'deki ifadeler kullanılmaktadır (Şekil 2.7).



Şekil 2.7 Ters Clarke Dönüşümü.

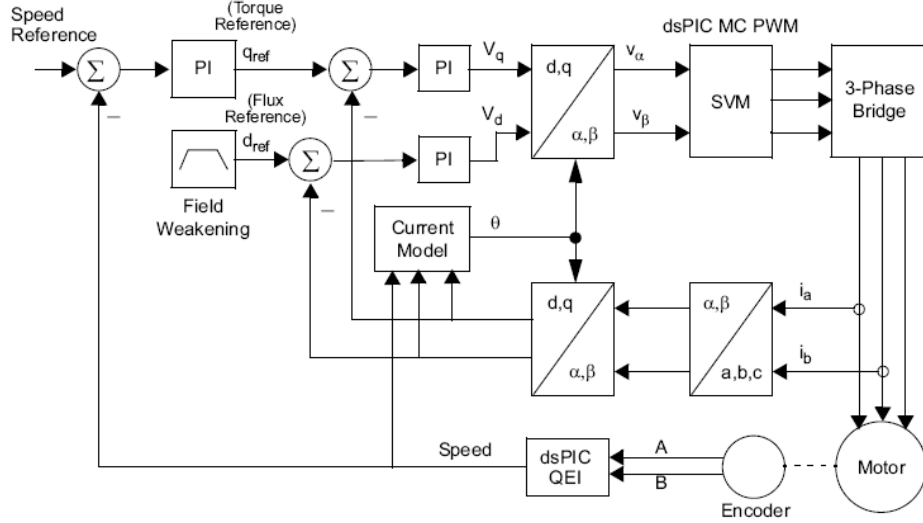
$$v_a = v_\beta \quad (2.53)$$

$$v_b = \frac{-v_\beta + \sqrt{3}v_\alpha}{2} \quad (2.54)$$

$$v_c = \frac{-v_\beta - \sqrt{3}v_\alpha}{2} \quad (2.55)$$

2.2.5 Vektör kontrol yöntemleri

Vektör kontrolü doğrudan vektör kontrolü ve dolaylı vektör kontrolü olmak üzere iki şekilde uygulanabilir. Doğrudan vektör kontrolünde rotor akısının pozisyonu doğrudan sensörlerle ölçülmektedir. Dolaylı vektör kontrolünde ise rotor hızı ve kayma hızından faydalanılır ve özel bir yapıya ihtiyaç yoktur. Sadece, hız geri beslemesi kullanılarak hesaplanabilirler. Vektörel kontrolde makinenin moment ifadesinin seçimi çok önemlidir. Moment ifadesi olarak stator akımı ile stator akısı veya rotor akısının oluşturduğu moment ifadeleri seçilecektir. Bu yöntemde stator akımı kolayca elde edilebilir ve sadece akı oluşturma problemi vardır. Moment ifadesinde yer alan akı (stator veya rotor akısı) sabit tutulabilirse, moment sadece stator akımının q eksenindeki bileşeni ile lineer olarak kontrol edilebilmektedir. Bu kontrol sırasında, tıpkı serbest uyarımalı doğru akım makinesinde olduğu gibi, akı değişmemelidir. Bu da ancak rotor akısı üzerinde yapılacak bir kontrol ile gerçekleştirilebilir. Stator akısı sabit tutularak moment kontrolü, stator üzerinden yapıldığında, akı bu kontrolden etkileneyeceği için lineer bir kontrol oluşamaz. Rotor akısı sabit tutulduğunda ise q eksenindeki stator akımı ile moment arasındaki ilişki lineerdir. Böylece akıyı ve momenti kontrol eden stator akım bileşenleri birbirlerine diktir ve birbirlerinden bağımsız olarak kontrol edilebilirler. Aralarında manyetik bir etkileşim yoktur (Öztürk 2000). Şekil 2.8'de vektör kontrolü blok diyagramı gösterilmiştir.



Şekil 2.8 Vektör Kontrolü blok diyagramı (Ross 2004).

2.2.5.1 Doğrudan vektör kontrolü yöntemi

Doğrudan vektör kontrolünde akı doğrudan ölçülerek bulunmaktadır. Ölçme yönteminde hava aralığına yerleştirilen akı algılayıcıları, statora özel olarak yerleştirilen akı bobinleri tarafından veya gözlemleyici model kullanarak bulunmaktadır. Akı algılayıcılarının sıcaklıktan etkilenmesi ve kırılğan olması dezavantajdır.

2.2.5.2 Dolaylı vektör kontrolü yöntemi

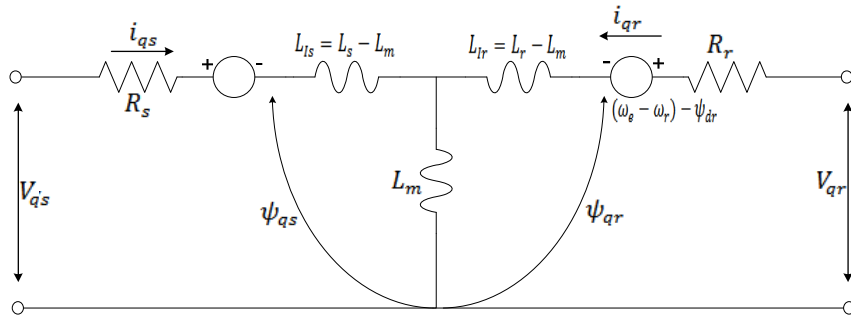
Doğrudan vektör kontrolünde, birim vektörlerin oluşturulmasına zorluklar ortaya çıkmaktadır. Dolaylı vektör kontrol yönteminde ise i_{ds} , i_{qs} çiftinin sadece tek bir kayma açısıl frekansı tanımlaması ilkesinden yararlanarak rotor akı vektörünün açısıl konumunun hesap yolu ile belirlenmesidir (Vas 1990).

i_{ds} , i_{qs} kontrol büyüklükleri olarak alındığında, rotor frekansı i_{qs} 'nün bir fonksiyonu olur. i_{qs} ile f_r , i_{ds} ile ψ_{dr} orantılıdır. Motor denklemleri ve hız geri beslemesi

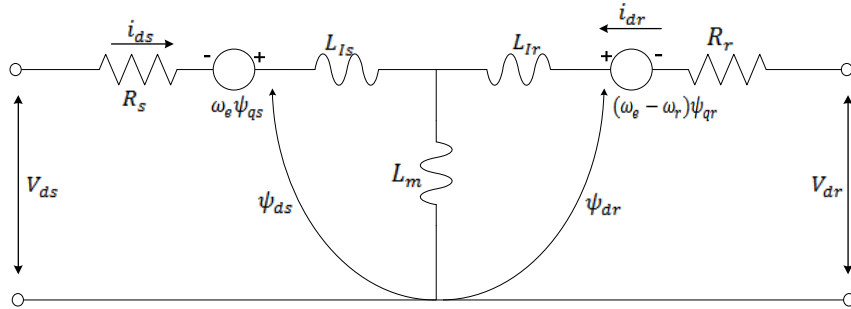
kullanılarak gerilimin etkin değeri, frekansı ve fazına ilişkin kontrol büyüklükleri oluşturulmuştur.

2.2.6 Asenkron motorun senkron referans çatı (d-q) dinamik modeli

Elektrik makinelerinin herhangi bir referans çatıda analizini sağlayan ve R.H. Park tarafından önerilen d-q dönüşümü (2.56-2.58)'deki gibi tanımlanır.



(a)



(b)

Şekil 2.9 Senkron düzlem (a) q^e- eksen ve (b) d^e- eksen dinamik model eşdeğer devresi (Bose 2006).

$$I_{qd0s} = S(\varphi)I_{abcs} = \frac{2}{3} \begin{bmatrix} \cos(\varphi) & \cos(\varphi - \frac{2\pi}{3}) & \cos(\varphi + \frac{2\pi}{3}) \\ \sin(\varphi) & \sin(\varphi - \frac{2\pi}{3}) & \sin(\varphi + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} + I_{abcs} \quad (2.56)$$

Burada

$$I_{qd0s} = [i_{qs} \quad i_{ds} \quad i_{0s}]^T, \quad (2.57)$$

$$I_{abcs} = [i_{as} \quad i_{bs} \quad i_{cs}]^T, \quad (2.58)$$

φ ise i_{qs} ile i_{as} arasındaki açıdır.

Denklem (2.56) kullanılarak senkron referans çatıda asenkron motorun matematiksel modeli (2.59-2.62)'deki gibi elde edilir (Bose 2006).

$$v_{qs}^e = R_s i_{qs}^e + \frac{d}{dt} \psi_{qs}^e + \omega_e \psi_{ds}^e \quad (2.59)$$

$$v_{ds}^e = R_s i_{ds}^e + \frac{d}{dt} \psi_{ds}^e + \omega_e \psi_{qs}^e \quad (2.60)$$

$$v_{qr}^e = 0 = R_r i_{qr}^e + \frac{d}{dt} \psi_{qr}^e + (\omega_e - \omega_r) \psi_{dr}^e \quad (2.61)$$

$$v_{dr}^e = 0 = R_r i_{dr}^e + \frac{d}{dt} \psi_{dr}^e - (\omega_e - \omega_r) \psi_{qr}^e \quad (2.62)$$

Burada,

v_{qs} , v_{ds} - stator gerilimleri,

v_{qr} , v_{dr} - rotor gerilimleri,

ψ_{qr} , ψ_{dr} - rotor akı halkalamaları,

ω_e - senkron hız,

ω_r - elektriksel rotor hız,

L_s - stator indüktansı,

R_s - stator direnci,

L_r - statora indirgenmiş rotor indüktansı,

R_r - statora indirgenmiş rotor direncidir.

Bu denklemlerdeki stator, rotor hava aralığı akı halkalanmaları akımlar cinsinden (2.63-2.65)'deki gibi yazılabilir.

$$\begin{bmatrix} \Psi_{qs}^e \\ \Psi_{ds}^e \\ \Psi_{qr}^e \\ \Psi_{dr}^e \end{bmatrix} = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \begin{bmatrix} i_{qs}^e \\ i_{ds}^e \\ i_{qr}^e \\ i_{dr}^e \end{bmatrix} \quad (2.63)$$

$$\Psi_{qm}^e = L_m (i_{qs}^e - i_{qr}^e) \quad (2.64)$$

$$\Psi_{dm}^e = L_m (i_{ds}^e - i_{dr}^e) \quad (2.65)$$

Motorun ürettiği elektriksel moment ise senkron referans çatıda (2.66)'teki gibi elde edilir.

$$T_e = \frac{3PL_m}{4L_r} (\Psi_{dr}^e i_{qs}^e - \Psi_{qr}^e i_{ds}^e) \quad (2.66)$$

Senkron referans çatıda rotor alan yönlendirmesinin bir sonucu olarak q^e - eksenini rotor akı halkalanması sıfır olduğundan ($\Psi_{qr}^e = 0$) yukarıdaki denklemlerin sadeleşmesiyle elektriksel moment ve senkron hız ise denklem (2.67-2.68)'deki gibi bulunur.

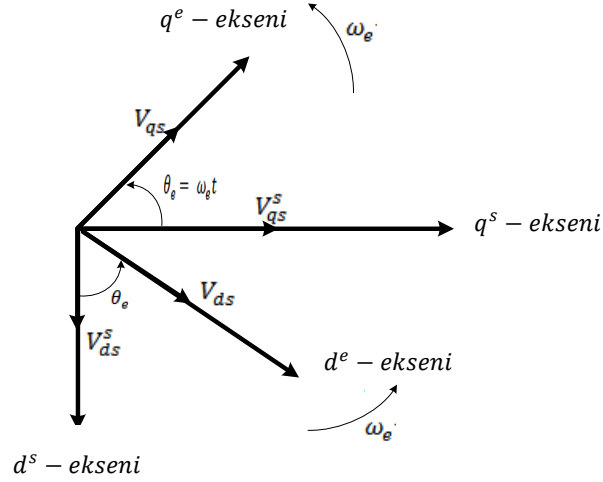
$$T_e = \frac{3PL_m}{4L_r} \Psi_{dr}^e i_{qs}^e \quad (2.67)$$

$$\omega_e = \omega_r + \frac{L_m}{\tau_r \Psi_{dr}^e} i_{qs}^e \quad (2.68)$$

Motorun mekanik devresinden yararlanarak motor hızı ve konumu denklem (2.69)'daki gibi belirlenebilir.

$$\frac{d\omega_r}{dt} = \frac{P}{2J} (T_e - T_L) , \quad \frac{d\theta_r}{dt} = \omega_r \quad (2.69)$$

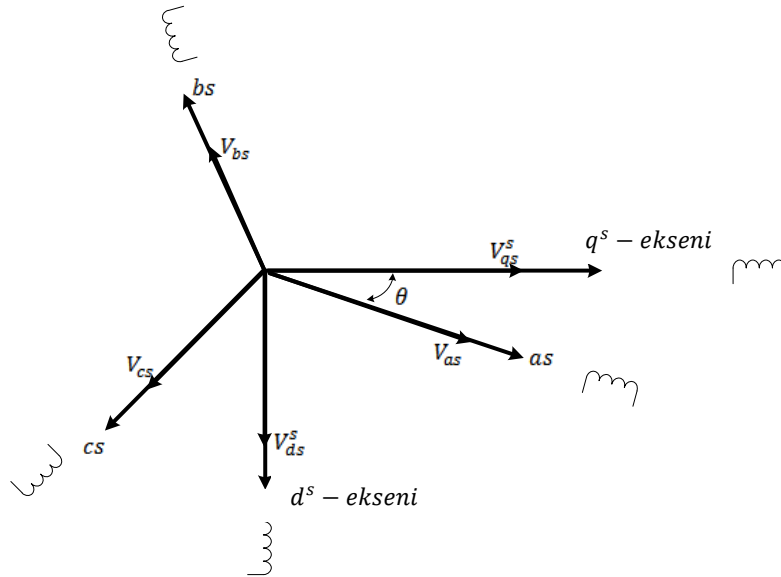
Senkron çatı eksenlerinden duran çatı eksenlerine ve buradan da üç faz gerilimlerine dönüşüm (2.70-2.74)'deki gibi elde edilir. Düzlemler arası dönüşümler Şekil 2.10 ve Şekil 2.11'de vektörel olarak verilmiştir.



Şekil 2.10 $d^s - q^s$ ve $d^e - q^e$ düzlemler arası dönüşüm.

$$v_{qs}^s = v_{qs}^e \cos\theta_e + v_{ds}^e \sin\theta_e \quad (2.70)$$

$$v_{ds}^s = -v_{qs}^e \sin\theta_e + v_{ds}^e \cos\theta_e \quad (2.71)$$



Şekil 2.11 $as - bs - cs$ ve $d^s - q^s$ düzlemlerine dönüşümler.

$$V_{as} = v_{qs}^s \quad (2.72)$$

$$V_{bs} = -\frac{1}{2}v_{qs}^s - \frac{\sqrt{3}}{2}v_{ds}^s \quad (2.73)$$

$$V_{cs} = -\frac{1}{2}v_{qs}^s + \frac{\sqrt{3}}{2}v_{ds}^s \quad (2.74)$$

Denklem (2.71-2.74) deki deęişkenlerde e-üstü senkron referans çatıyı, s-üstü ise duraęan referans çatıyı göstermektedir (Gökbulut 2005).

2.3 Yapay Sinir Ağları'nın Teori ve Uygulaması

Yapay sinir ağları ya da kısaca YSA; insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır. En genel anlamda bir YSA insan beynindeki birçok nöronun (sinir hücresinin), ya da yapay olarak basit işlemcilerin birbirlerine deęişik etki seviyeleri ile bağlanması sonucu oluşan karmaşık bir sistem olarak düşünülebilir. Önceleri temel tıp birimlerinde insan beynindeki nöronların matematiksel modelleme çabaları ile başlayan çalışmalar, geçtiğimiz on sene içerisinde, disipline bir şekil almıştır. YSA bugün fizik, matematik, elektrik ve bilgisayar mühendisliği gibi çok farklı bilim dallarında araştırma konusu haline gelmiştir. YSA'nın pratik kullanımı genelde, çok farklı yapıda ve formlarda bulunabilen enformasyon verilerini hızlı bir şekilde tanımlama ve algılama üzerinedir. Aslında mühendislik uygulamalarında YSA'nın geniş çaplı kullanımının en önemli nedeni, klasik tekniklerle çözümü zor problemler için etkin bir alternatif oluşturmasıdır. Çünkü bilgisayarlar insanın beyinsel yeteneğinin en zayıf olduğu çarpma, bölme gibi matematiksel ve algoritmik hesaplama işlemlerinde hız ve doğruluk açısından yüzlerce kat başarılı olmalarına rağmen insan beyninin öğrenme ve tanıma gibi işlevlerini hala yeteri kadar gerçekleştirememektedir. Çizelge 2.1'de bilgisayar ile insan beyni arasındaki çalışma sistem yapısı karşılaştırmalı olarak verilmiştir (Güleç, 1999).

Çizelge 2.1 Bilgisayar ile insan beyni arasındaki çalışma sistem yapısının karşılaştırılması.

BİLGİSAYAR	İNSAN BEYNİ
Sayısal	Analog
Seri	Paralel
Komut Kümeli	Bilgiye Adapte Olma
Yanlış Hesaplamalar Sonucu Etkiler	Birimlerin Ana İşlemlere Etkisi Azdır
Giriş Verilerindeki Hatalar Sonucu Etkiler	Giriş Verilerindeki Hatalara Her Zaman Duyarlı Değil

2.3.1 YSA'nın tanımı ve modeli

YSA paralel dağılmış bir bilgi işleme sistemidir. Yani, YSA'nın temelinde, zeka gerektiren işlemlerden oluşan bilgi işleme işlevi vardır. Bu sistem tek yönlü işaret kanalları (bağlantılar) ile birbirine bağlanan işlem elemanlarından oluşmuştur. Çıkış işareti bir tane olup isteğe göre çoğaltılabilir. YSA yaklaşımının temel düşüncesiyle, insan beyninin fonksiyonları arasında benzerlik vardır. Bu yüzden YSA sistemine insan beyninin modeli denilebilir. YSA çevre şartlarına göre davranışlarını şekilleyebilir. Girişler ve istenen çıkışların sisteme verilmesi ile kendisini farklı cevaplar verebilecek şekilde ayarlayabilir. Ancak son derece karmaşık bir içyapısı vardır. Onun için bugüne kadar gerçekleştirilen YSA; biyolojik fonksiyonların temel nöronlarını örnek alarak yerine getiren kompoze elemanlar olmuştur.

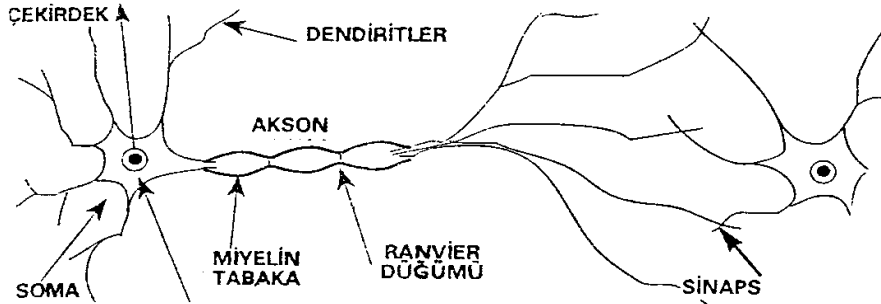
2.3.1.1 Nöronun biyolojik yapısı ve nöron modeli

İnsanın bilgi işleme olayı beyinde gerçekleşir. Gerçekten de en karmaşık sinir ağı Cerebral Cortex denilen "beyin"dir. Sinir sisteminin en basit yapısı nöronlardır. Beyinde yaklaşık olarak 10^{10} sinir hücresi vardır. Yine hücre başına bağlantı sayısı ise 10^4 mertebesinde. Beyin için çalışma frekansı 100 Hz'dir. Fiziksel boyutları ise 1.3

kg ve 0.15 m² kesitlidir. Vücudun değişik yerleri ile bilgi alışverişi yapan nöron hücresidir. Şekil 2.12’de basit bir nöron hücresi görülmektedir.

Nöron, soma adı verilen hücre gövdesi dentrit denilen kıvrımlı uzantılar ve somanın dalları sayesinde nöronu dallarına bağlayan tek sinir fiberli aksondan oluşmaktadır. Dendrit’ler hücreye gelen girişleri toplamaktadırlar. Dendrit tarafından alınan işaretler hücrede birleştirilerek bir çıkış darbesi üretilip üretilmeyeceğine karar verirler. Eğer bir iş yapılacaksa üretilen çıkış darbesi aksonlar tarafından taşınarak diğer nöronlarla olan bağlantılara veya terminal organlara iletilmektedir. Beyindeki korteksde her nöronun bir karşılığı vardır. Bir nöronun çıkışı ona bağlı olan bütün nöronlara iletilmektedir. Fakat korteks, işin yapılabilmesi için hangi nöron harekete geçirilecekse, sadece ona komut göndermektedir.

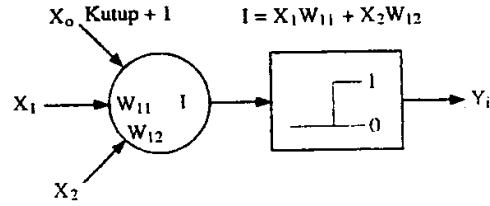
Somanın içinde ve çevresinde sodyum, kalsiyum, potasyum ve klor iyonları vardır. Potasyum yoğunluğu nöronun içinde, sodyum yoğunluğu dışındadır. Somanın zarı elektriksel olarak uyarılınca (söz konusu uyarı genellikle bir gerilim düşmesidir) zar, Na ve Ca gibi diğer iyonların içeri geçmesine izin verir ve somanın iç durumunu değiştirmektedir. Nöronlar arasındaki bağlantılar hücre gövdesinde veya "sinaps" adı verilen dendritlerdeki geçişlerde olmaktadır.



Şekil 2.12 Basit bir nöron yapısı (Güleç 1999).

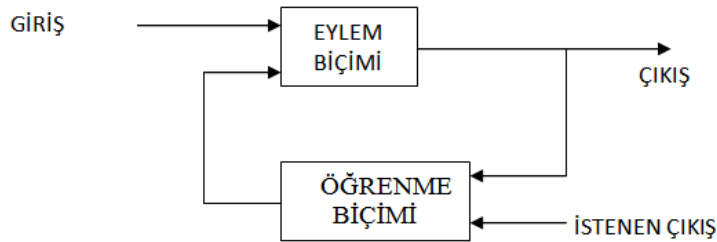
Yardımcı bir benzetme aksonlarla, dendritleri elektrik sinyallerini nörona ileten değişik empedanstaki yalıtılmış iletken olmasıdır. Sinir sistemi milyarlarca nöron ile tek bir nörondan çıkan aksonun 10000 kadar diğer nöronu bağlayan bir ağdır. Sinapslarla

düzeltilen işaretleri taşıyan aksonlar ve dendritlerle iç içe geçmiş nöronlar bir sinir ağı oluşturur. Şekil 2.13’de en basit formda gösterilen nöron modeli, bir eşik birimi olarak algılanabilmektedir. Şekil 2.14’de ise YSA’nın genel blok şeması gösterilmektedir.



Şekil 2.13 Nöron modeli (Gülez 1999).

Eşik birimi, çıkışları toplayan ve sadece girişin toplamı iç eşik değerini aştığında bir çıkış üreten işlem elemanıdır. Bir eşik birimi olarak nöron sinapslarındaki işaretleri almakta ve hepsini toplamaktadır. Eğer toplanan işaret gücü eşiği geçecek kadar güçlü ise diğer nöronları ve dendritleri uyaran akson boyunca bir işaret gönderilmektedir. Kesişen dendritlerden gelen sinapslarla kapılanan bütün işaretleri soma toplamaktadır. Toplam işaret daha sonra nöronun iç eşik değeri ile karşılaştırılmakta ve eşik değerini aşmışsa aksona bir işaret yaymaktadır. YSA, bu basit nöronların (düğümünün ya da ünitelerin) bağlanarak bir ağ’a dönüştürülmesiyle meydana getirilmiştir.

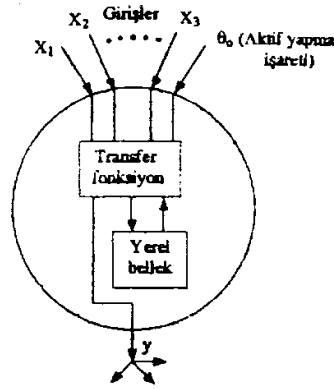


Şekil 2.14 YSA’nın genel blok şeması.

2.3.2 YSA’nın yapısı ve işlem elemanı

YSA temel olarak, basit yapıda ve yönlü bir graf biçimindedir. Her bir düğüm hücre denilen n. dereceden lineer olmayan bir devredir. Düğümler işlem elemanı olarak

tanımlanmaktadır. D ğ mler arasında baėlantılar vardır. Her baėlantı tek y nl  iŐaret iletim yolu (gecikmesiz) olarak g rev yapmaktadır. Her iŐlem elemanı istenildiėi sayıda giriş baėlantısı ve tek bir ıkıŐ baėlantısı alabilmektedir. Fakat bu baėlantı kopya edilebilmektedir. Yani bu tek ıkıŐ birok h creyi besleyebilmektedir. Aė'daki tek gecikme, ıkıŐları ileten baėlantı yollarındaki iletim gecikmeleridir. İŐlem elemanının ıkıŐı istenilen matematiksel tipte olabilmektedir. Kısmen s rekli alıŐma konumunda "aktif" halde eleman bir ıkıŐ iŐareti  retmektedir. GiriŐ iŐaretleri YSA'ya bilgi taŐımaktadır. Sonu ise ıkıŐ iŐaretlerinden alınabilmektedir. Őekil 2.15'de genel bir iŐlem elemanı (n ron, d ė m) g sterilmiŐtir.

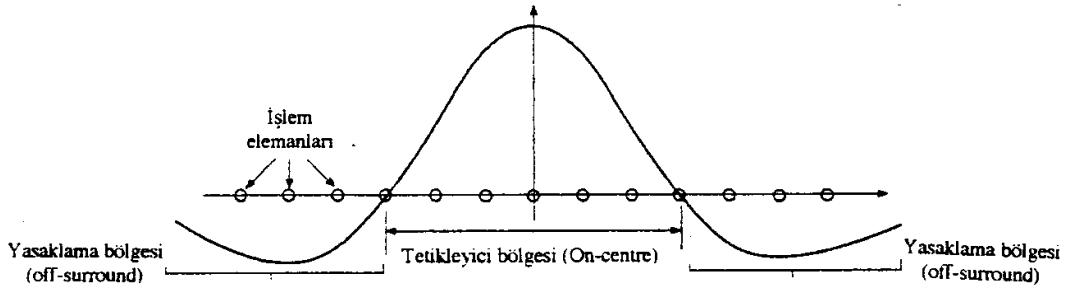


Őekil 2.15 Genel iŐlem elemanı yapısı (G lez 1999).

YSA birtakım alt k melere ayrılabilir. Bu alt k melerdeki elemanların transfer fonksiyonları aynıdır. Bu k çük gruplara "katman" layer adı verilmektedir. ( rn: ok katmanlı perceptron MLP) aė katmanların birbirlerine hiyerarŐik bir Őekilde baėlanmasından oluŐmuŐtur. DıŐ d nyadan alınan bilgi giriş katmanı ile taŐınmaktadır. Bir transfer fonksiyonları yoktur. YSA transfer fonksiyonu ve yerel bellek elemanı bir  ėrenme kuralı ile giriş ıkıŐ iŐareti arasındaki baėlantıya g re ayarlanmaktadır. Aktif yapma giriŐi iin bir zamanlama fonksiyonu tanımlaması gerekebilir. Kısaca bir YSA'dan beklenen g rev, gerek d nyadaki nesnelere biyolojik sinir aėının yaptıėı iŐlevi, benzer bir yolla yerine getirmesidir. YSA'nın giriş veri tipleri ikili (binary) 0-1 veya s rekli deėerlerdir. Bu giriş durumlarından baŐka, iŐlem elemanlarına ait girişleri matematiksel olarak da sınıflamak gerekmektedir.  nk  bir iŐlem elemanına gelen girişlerin bir kısmı azaltıcı uyarma girişleri olmaktadır. Bu arttırıcı veya azaltıcı girişler "giriŐ sınıflarını" oluŐturur.

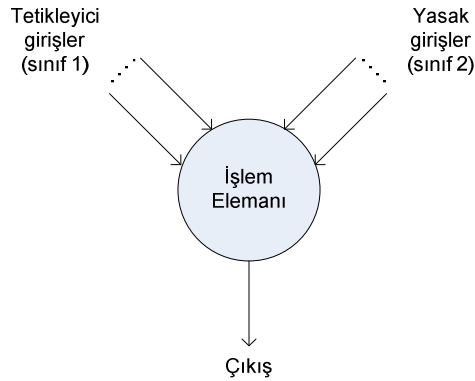
2.3.2.1 Giriş işareti sınıfları

İşlem elemanının transfer fonksiyonu gelen bütün giriş işaretleri için tanımlanmıştır. Bazen değişik katman davranışlarının farklı olması tabiidir. İşaretlerin hangi bölgelerden geldiğinin bilinmesi gerekmektedir. Değişik bölgelere göre işaretlerin sınıfları tamamlanabilir. Sıkça izlenen bir yapı ise merkezde evet/çevrede hayır (on centre/off surround) yapısındadır. Şekil 2.16'da bu yapı gösterilmektedir. Meksika şapkasına benzer bağlantı tipindedir.



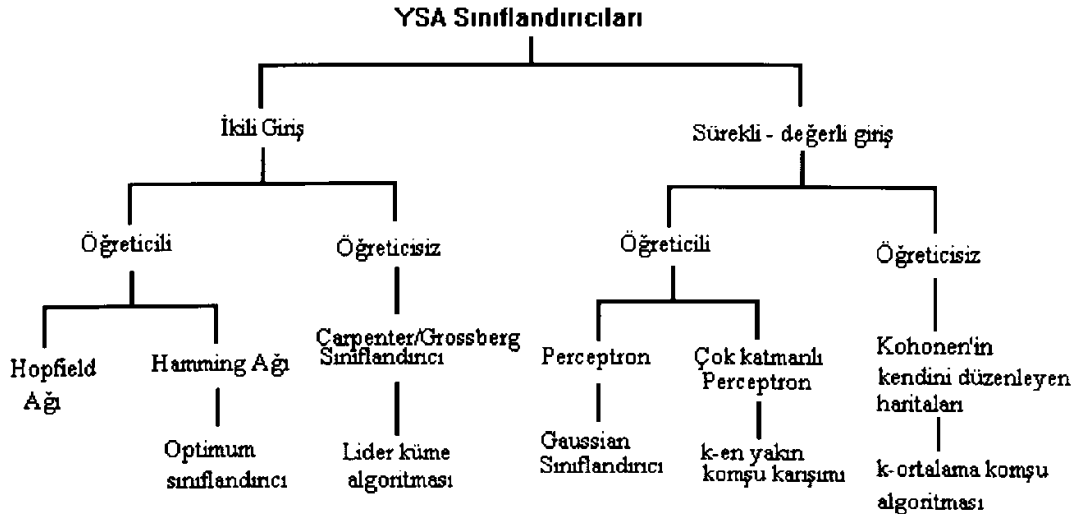
Şekil 2.16 Komşu hücrelerin merkez hücreye etkisi (Güleç 1999).

İşlem elemanı tetikleyici girişlerin kendine yakın komşu girişlerden yasaklanan girişlerini daha uzaktan almaktadır. Böylece işlem elemanına gelen girişler sınıflarına göre değerlendirilmiş olmaktadır. Tetikleyici bölgeden gelen girişler yasaklanan sınıfı oluşturur. Şekil 2.17 böyle bir işlem elemanını göstermektedir.



Şekil 2.17 Tetikleyici ve yasaklanan girişlere sahip bir işlem elemanı.

Bir işlem elemanına gelen girişler matematiksel tiplerine göre etiketlenilerek sınıflandırılmıştır. YSA, giriş veri tiplerine göre ikili giriş (0,1) ve sürekli değerli giriş olmak üzere aşağıdaki gibi sınıflandırılmıştır (Şekil 2.18).



Şekil 2.18 YSA sınıflandırıcılarının grafiksel gösterimi (Güleç 1999).

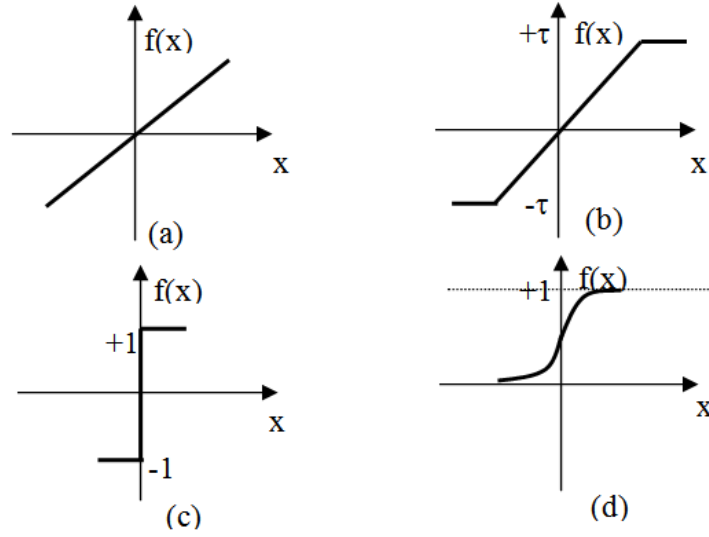
2.3.2.2 Ağ tipleri

Üç çeşit ağ tipi vardır:

- 1- İleri beslemeli ağ: Her bir katmandaki hücreler sadece bir önceki katmanın hücrelerince beslenir.
- 2- Kaskat bağlantılı ağ: Hücreler sadece önceki katmanlardaki hücrelerce beslenir.
- 3- Geri beslemeli ağ: En az bir hücre sonraki katmanlardaki hücrelerce de beslenir.

2.3.2.3 Eşik fonksiyonları

Transfer veya işaret fonksiyonları olarak da adlandırılan eşik fonksiyonları, muhtemel sonsuz domen girişli işlem elemanlarını önceden belirlenmiş sınırdaki çıkış olarak düzenlemiştir. Dört tane yaygın eşik fonksiyonu vardır. Bunlar lineer, rampa, basamak ve sigmoid fonksiyonudur. Şekil 2.19'da bu fonksiyonlar gösterilmiştir.



Şekil 2.19 Sıkça kullanılan (a) Lineer, (b) Rampa, (c) Basamak ve (d) Sigmoid fonksiyonları.

Şekil 2.19 (a)'da gösterilen lineer fonksiyonun denklemi (2.75)'deki gibidir.

$$f(x) = \alpha \cdot x \quad (2.75)$$

α : işlem elemanının x aktivitesini ayarlayan reel değerli bir sabittir. Lineer fonksiyon $[-\tau, +\tau]$ sınırları arasında kısıtlandığında Şekil 1.19 (b)'deki rampa eşik fonksiyonu olur ve denklemi (2.76) şeklini alır.

$$f(x) = \begin{cases} +\tau: \text{eğer } x \geq \tau \text{ ise} \\ x: \text{eğer } |x| < \tau \\ -\tau: \text{eğer } x \leq -\tau \text{ ise} \end{cases} \quad (2.76)$$

$+\tau$ ($-\tau$) işlem elemanının maksimumu (minimumu) çoğu zaman doyma seviyesi olarak adlandırılan çıkış değeridir. Şekil 2.19 (c), basamak eşik fonksiyonunu gösterir ve denklemi (2.77) şeklindedir.

$$f(x) = \begin{cases} +1: \text{eğer } x > 0 \\ -1: \text{diğer durumlar} \end{cases} \quad (2.77)$$

Son ve en önemli eşik fonksiyonu (bu çalışmada kullanılan) sigmoid fonksiyonudur. Şekil 2.19 (d)'de gösterilen S biçimindeki sigmoid fonksiyonu; seviyeli, lineer olmayan çıkış veren, sınırlı, monoton artan fonksiyondur. Denklemi (2.78) biçimindedir.

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.78)$$

Her işlem elemanı kendisine verilen yerel veriye göre, kendisini ayarlayacak bütün YSA'nın enformasyon bölgesinin öğrenmesini sağlamaktadır. (Enformasyon bölgesi olasılık-yoğunluk fonksiyonu ile de tanımlanabilir). Enformasyon bölgesi birçok uygulamada, gerçek değerler "0" ile "1" arasında normalize edilmesi gerekmektedir. (Normalize etmek: gerçek değeri 85 olan bir girişi 0.85 şeklinde ağa uygulamaktır.) Normalizasyon aynı anda bütün girişlere uygulanabilir.

2.3.2.4 Ağırlık uzayı

Birçok YSA öğrenme işlemi, işlem elemanlarının ağırlığı değiştirilerek sağlanmaktadır. Böylece tanımlanan ağırlık değiştirilerek öğrenmede iyi bir model kullanıp, ağırlıkların bu modele göre değiştirilmesi esastır. Basit bir matematiksel model olarak her bir işlem elemanının "n" adet gerçek ağırlığı olduğu düşünülerek ve N adet işlem elemanı göz önüne alınırsa (2.79-2.80) elde edilir.

$$w = (w_{11}, w_{12}, \dots, w_{1n}, w_{21}, w_{22}, \dots, w_{2n}, \dots, w_{N1}, w_{N2}, \dots, w_{Nn})^T \quad (2.79)$$

$$w = (w_1^T, w_2^T, w_3^T, \dots, w_N^T) \quad (2.80)$$

w_1, w_2, \dots, w_N : işlem elemanlarının ağırlık vektörleridir.

$$\begin{array}{r} w_{11} \\ w_{12} \\ w_1 = \cdot \quad \dots \quad w_N = \cdot \\ \cdot \\ w_{1n} \end{array} \quad \begin{array}{r} w_{N1} \\ w_{N2} \\ \cdot \\ w_{Nn} \end{array} \quad (2.81)$$

YSA ağırlık vektörü N, n boyutlu öklid uzayında yayılır. YSA'nın enformasyon işleme performansı, ağırlık vektörünün belirli bir değeri ile bulunacaktır.

Hata değişimini inceleyen iki çeşit kural vardır.

- 1- Hata düzeltme kuralları,
- 2- Gradyen kuralları.

Hata düzeltme kuralları; Her bir giriş bağıntısında ağırlıkları yeniden ağırlayarak çıktı hatasını en aza indirmeye çalışmaktadır. Gradyen kurallarında ise, ağırlıklar yeniden ayarlanarak ortalama karesel hata (MSE) en aza indirilmeye çalışılmaktadır.

Bu noktada Gradyen kuralından kısaca bahsedecek olursak, hatayı düzeltmenin (yani minimize etmenin) geometrik bir yorumunu yapmak mümkündür. Bunu yapabilmek için ağırlıkların mümkün olan tüm değerleri, hataların kareleri toplamına karşı gelecek şekilde üç boyutlu koordinat sisteminde çizilir. Bu çizim sonunda hata yüzeyi küresel bir top şeklindedir. Bu yüzeyi bir tase da benzetmek mümkündür. Tasın en alt kısmı hataların kareleri toplamının en küçük değerlerine karşı gelmektedir. Eğitim sırasında amaç, ağırlıklar kümesinin en iyisini bulmak olan, en alt kısmına ulaşmaktır. Geriye-yayılım algoritması o andaki ağırlıklar yerine, yüzey hatasının eğimini hesaplayarak amacına ulaşmaktadır. Daha sonra da bu ağırlıkları tasın alt kısmına doğru artımsal olarak değiştirmektedir. İşte bu artımsal olarak tasın üst kısmından alt kısmına doğru ilerleme işlemine "Gradyen iniş" denir.

Ağırlık vektörü ile çalışan YSA'da önemli noktalardan birisi, bir öğrenme kuralı geliştirip, enformasyon bölgesi kullanarak (eşik fonksiyonu ile) ağırlık vektörü "w" iyi istenilen YSA performansı verecek noktaya yöneltmektir. Genellikle öğrenme kuralı için bir performans ya da maliyet fonksiyonu tanımlanmıştır. Minimizasyon veya maksimizasyon ile "w" vektörü bulunur. Bir performasyon çeşidi olarak bilinen, MSE (karesel ortalama hata) (2.82)'deki şekilde tanımlanmıştır. Amaç F'yi küçültmeye çalışmaktır.

$$F(w) = \int_A |f(x) - G(x, w)|^2 \rho(x) dv(x) \quad (2.82)$$

$y=G(w,x)$ - Sistemin giriş-çıkış fonksiyonu.

y - Çıkış işareti vektörü

x - Giriş işareti vektörü

w - Ağırlık vektörü

$\rho(x)$ - Olasılık yoğunluk fonksiyonudur.

2.4 Sayısal İşaret İşlemci (DSP)

Kontrol algoritmalarının gerçek zamanda işlem yapabilmesi, dış dünyadan; yani üzerinde çalışılan sistemden elde edilen fiziksel büyüklüklerin değerlendirilip aynı anda gerekli olan kontrol parametrelerinin hesaplanması anlamına gelmektedir. Sistemler karmaşık bir yapıya sahip olup, algoritmalar daha kapsamlı hale geldikçe mikroişlemciler yapıları itibariyle çözüm için yeterli olmamaya başlamışlardır. Tüketiciden gelen talepler doğrultusunda üretici firmalar uygulamalara özel bazı işlemciler üretmişlerdir. Bu işlemciler genel amaçlı işlemcilerden daha hızlı, fakat uygulamalarda özel olduklarından kullanım alanı açısından genel anlamda sınırlı işlem kapasitesinde kalmışlardır. Bu tür teknolojiye yönelik uygulamalarda ulaşılmak istenen hedef, mikroişlemcilerin yararlarını yapısında bulunduran ve yetersiz kaldığı durumlarda yeni olanaklar sunan bir işlemci geliştirmek olmuştur (Üstün 2001).

Mikroişlemcilerin üretilmesinde kullanılan yarı iletken teknolojisi, gün geçtikçe yeni gelişmelere tanık olmaktadır. Very Large Scale Integration (VLSI) denilen üretim tekniği yardımıyla, çok daha hızlı ve güçlü yapılar, çok daha küçük alanlara sıkıştırılarak, üretilmeye başlanmıştır. İşlemcilerin güç kaybının düşük değerde olmasını sağlayan bu sistem, tasarımcıya karmaşık problemlerin kolayca çözümlenebilmesi kolaylığını getirmiştir. Mikroişlemcilerin mimari yapıları iki çeşittir. Bu mimariler kendilerini geliştiren kişilerin adları ile anılmaktadır. Bunlardan bir tanesi Von Nuemann olarak adlandırılmıştır. Veri ve komutların saklanması için aynı bellek biriminin kullanılması, aynı bilgi transfer yolu üzerinde veri daralması olarak da tanımlayabileceğimiz veri sıkışması problemlerine neden olmaktadır. Komut ile veri aynı yol üzerinden taşınmaktadır. Bu da işlemlerin çok daha uzun sürede tamamlanmasına ve saat çevrim süresinin uzamasına sebep olmaktadır. Örneğin 16 bitlik sayılarla işlem yaparken, 8 bitlik bir işlemci, sayılan iki parça olarak alıp, işlemektedir. Bu da 16 bitlik bir işlemcinin yan hızında çalışması anlamına gelmektedir. Yeni nesil işlemcilerde, üniteler bağımsız olarak çalıştırılarak, hızı artırma yoluna gidilmiş ve kısıtlı bir paralel çalışma sağlanmıştır. Veri taşınması, eklenen yeni komut ve yapı değişikliği ile tek saat çevriminde gerçekleştirilebilmektedir. Harvard mimarisi olarak adlandırılan ikinci tip mimari DSP'lerde ve yeni nesil işlemcilerde kullanılmaya başlanmıştır. Bu mimari tipinde veri ve adres yollarının birbirinden ayrı olması nedeniyle, işlemci çok daha hızlı işlem yapabilmektedir. Bunun yanı sıra DSP'lerde bulunan paralel işlem yapabilme özelliği, aynı anda birkaç komut işleyerek hızının katlanmasını sağlayan bir diğer etken olmaktadır. Bir mikroişlemcinin işlem yapabilme kapasitesini gösteren komut sayısı ve adresleme modları, DSP sistemlerinde diğer işlemcilere göre daha kısıtlı gibi görünse de, sistemin özelliklerini tam olarak kullanmayı sağlayan özel komutlar sayesinde, birçok işlem tek bir saat çevrimi süresinde yapılabilmektedir. Özellikle kontrol sistemlerinde çok kullanılan çarpma ve toplama işlemlerinin birkaç makine çevriminde gerçekleştirilebilmesi, hız yönünden büyük bir avantaj getirmektedir. Bir algoritma içinde tekrarlanan döngü işlemleri; DSP'lerin özel komutları sayesinde çok daha hızlı ve verimli olarak gerçekleştirilebilmektedir. Donanım olarak yapıların birbirinden bağımsız olması, aynı yollar üzerinden veri ve adres iletebilmesi; merkezi işlem birimi, merkezi mantık birimi ve yardımcı

kaydedici aritmetik ünitelerinin yoğun paralellik içersinde çalışması sistemin üstün özellikleri olarak tanımlanabilir.

Uygulamaya yönelik olarak, işlemlerde kullanılan sayıların yapısal özellikleri doğrudan doğruya yolların genişliğine bağlıdır. Genel olarak sabit nokta aritmetiği ile çalışan sistemlerde 16 bit, daha büyük sayılar kullanan kayan nokta aritmetiği ile çalışan sistemlerde 32 bit yol genişliği bulunmaktadır. Çarpma ve toplama işlemleri sonrasında oluşan yuvarlama ve kesme hataları, kontrol edilen sistem üzerinde büyük hatalara neden olabilir. Bunu önleyebilmek için, işlemcinin işlem yapabilme kapasitesinin yeterince büyük olması gerekmektedir. DSP'lerde donanım olarak bulunan çarpma ünitesinin çıkışı, çarpma işleminden sonra bit sayısının iki katı bir sonuç oluşturmaktadır. Veri yolunun iki katı büyüklüğünde olan bu sayı ile belirli bir yuvarlatma hatası sonucu tek bir kelime boyutunda veya doğrudan çift kelime boyutunda işlem yapılabilir.

DSP'nin program çevrimleri sırasında hangi adresteki işlemi gerçekleştireceğini otomatik olarak belirlemesi, istenen bir özelliktir. Bu işlemin Aritmetik Lojik Ünite (ALU) tarafından gerçekleştirilmesi mümkündür. Fakat belirli bir gecikmeye neden olacağından, DSP içinde ayrı bir veri adres jeneratörü bulunmaktadır. Adresleme işleminin önem kazandığı, sıralı adreslemenin yapılmadığı uygulamalarda bu ünite, bağımsız çalışan bir işlemci gibi adresleri üretmektedir.

Algoritmanın işlenmesi sırasında komutların sıralı ve doğru olarak işlenmesi, komut sıralama ünitesi sayesinde olmaktadır. Bu ünite sayesinde, komutların işlenmesi sırasında, işlemcinin mikro kod olarak çevrim ve döngüleri kullanmasına gerek kalmamaktadır. Bu da komutların daha az çevrim ile daha kısa sürede tamamlanmasını sağlamaktadır.

Ünitelerde yapılan işlemler sonunda elde edilen verilerin genişliğinin, veri yolunun genişliğinden fazla olmasından dolayı oluşan hatalar, kaydırma ünitesi yardımı ile ana işlemciye gerek kalmadan, tek bir çevrim süresinde çözümlenmektedir. Kaydırma

ünitesi aynı zamanda, çok bitli tek çevrim kaydırmalar, normalize/denormalize işlemleri, kayan nokta aritmetiği gibi işlemlerde büyük kolaylık sağlamaktadır.

Yukarıdaki kısımlarda sözü edilen, sayısal işaret işleme sırasında karşılaşılan sorunları çözmek için, DSP'ler çeşitli yöntemlerle (bu bir yazılım olabilir) en az hata ile işlem yapabilmelidir. Komut işleme sırasında komutların aynı anda aynı veri yollarını kullanma isteğinden ortaya çıkan durum veri sıkışması olarak adlandırılmaktadır. Aynı durumla işlemcilerin veri işleminde de karşılaşılmaktadır. Bu sorunu ortadan kaldırmak için işlemcinin paralel çalışma denilen parçalara ayırarak çalışma yapısının kurulması gerekmektedir. Bu sayede komut çevrim süresi en az düzeyde olacak ve aynı anda birden fazla komut işlenebilecektir. İşlemi paralel yollara ayırarak yapılan çalışma, her bir parçanın bağımsız olarak işlenmesi prensibine dayanmaktadır. Bu çalışma şekli, parça sayısı ile ters orantılı olacak oranda, işlemin kısa sürede tamamlanmasını sağlamaktadır. Bunun yanında, işlemlerin en uygun olarak dağıtılması her zaman söz konusu olmayabilir. Fazladan eklenen kütükler işlemin hızını arttırmaya yarayacaktır. Örneğin çarpma işleminde, çarpanların ayrı yollar üzerinden taşınması, çarpma işleminin süresini iki katı oranda azaltmaktadır (Üstün 2001).

Yukarıdaki kısımda bahsettiğimiz paralel çalışma özetle, aynı anda birden fazla komutun işlenmesine izin veren yapıya sahip olmasının yanı sıra, komut saat çevrimini en düşük seviyesine indirirken, işlemcinin çıkışından en yüksek performansın alınmasını sağlamaktadır. Bu da performansı artırıcı bir diğer etkidir.

Paralel çalışma işleminin tam olarak yeterli olmadığı durumlarda paralel çalışma alternatif bir çözüm olmaktadır. Bir işlemin alt parçalara ayrılıp işlenmesi yerine, paralel işlemciler ile aynı işlemin yapılması çok daha iyi sonuç vermektedir. Bu tür çalışma ise, donanımın karmaşıklığına ve fiyatının biraz artmasına sebep olsa da yapılacak işlemlerin birbirinin yerine geçebilir yapıda ayarlanması çalışmadaki verimi arttıracaktır. Bütün bu anlatılanlar dikkate alındığında bu ünitelerin tümünün birbiri ile uyumlu çalışabilme özelliklerinden dolayı, karmaşık algoritmalar hızlı bir şekilde işlenebilmektedir. Mimari yapıyı oluşturan ünitelerin birbirinden bağımsız olarak

çalışabilmesi, sistemin hızını katlamaktadır. Komut çevrim süresinin kısa olması, belirtilen bu özelliklerden çok daha verimli olarak yararlanabilmeyi sağlamaktadır. DSP'ler de arzu edilen özellik, tüm komutların tek bir saat çevriminde işlenmesidir. Bunun mümkün olmaması durumunda, gelişmiş mikro işlemcilerdeki gibi, komutların mikro kod olarak çözümlenmesi yerine, mimari yapı, komutların gerektirdiği işlemleri doğrudan yapabilen lojik bir yapı olarak kurulmuştur. Sistem, mümkün olduğu kadar çok işlemi mümkün olan en az saat çevriminde gerçekleştirdiğinde, en verimli mimari yapıya sahip demektir.

Sistem tasarımı sırasında, DSP sisteminin seçiminde belirli faktörlerin önemi bulunmaktadır. Bunlar; mimarisini oluşturan ünitelerin içyapısı, işlem yapabilme kapasiteleri, komut giriş grupları ve geliştirme üniteleridir. Genel amaçlı işlemcilerde benzer yapılar bulunmasına rağmen, DSP'lerin kendilerine özgü bir sınıflandırması bulunmaktadır. Sistem mimarisi, kontrol edilecek sisteme çok bağımlıdır. Gerekli olan yapı ancak kontrol edilecek sistemin çok iyi tanınması ile mümkündür.

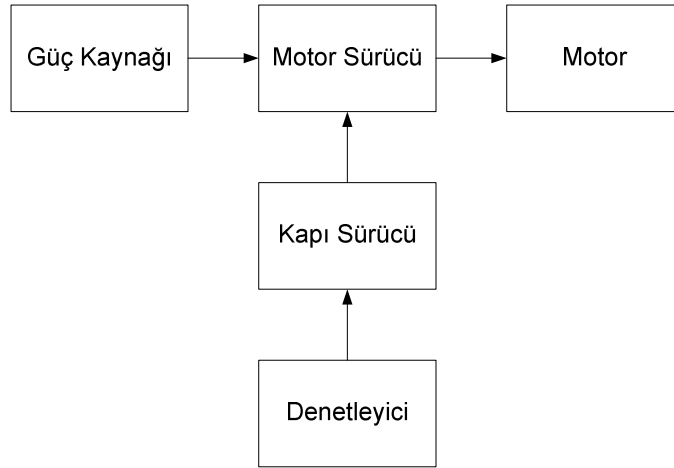
Seçilecek mimari yapı, kontrol algoritmasını en az hata ile en hızlı biçimde en yüksek performansı sağlayacak şekilde gerçekleştirmelidir. Bunun yanında, başka bir sisteme ne kadar kolay uyum sağladığı etkili bir faktördür. Kontrol sisteminin isteğine göre, yardımcı birimler, bellek birimleri ve giriş-çıkış birimlerini bünyesinde bulundurmalı veya bu birimler ile kolaylıkla iletişim kuracak yapıda olmalıdır. DSP sistemlerinde kolay programlanabilme de önemli bir özelliktir. Bunların yanında performans-fiyat ve performans-verim arasındaki ilişkilerinde göz önüne alınması gereklidir.

2.4.1 DSP uygulama alanları

VLSI teknolojisindeki gelişmeler, DSP sistemlerinin çok kısa bir süre içinde kontrol uygulamalarının bütün alanlarına girmesine olanak sağlamıştır. DSP'nin genel işaret işleme alanında, dijital filtreleme, konvolusyon, korelasyon, hilbert dönüşümleri, Hızlı Fourier Dönüşümü (FFT), adaptif filtreleme ve sinyal üretimi gibi geniş uygulama alanları bulunmaktadır. Başlıcaları; görme sistemleri, spektrum analizi, ses tanıma,

robot ve motor kontrolü, radar işleme, modemler ve cep telefon sistemleri, adaptif yol kontrolü, işitme cihazlarıdır. Bütün bu örneklerden de anlaşıldığı gibi, dijital sistemlerin olduğu her alanda DSP kullanılmaya başlanmıştır.

Motor kontrol sistemlerinde DSP'nin kullanılması diğer uygulama alanları ile karşılaştırıldığında oldukça geç olmuştur. Bunun nedeni, mikroşlemcilerin yeterli düzeyde bir performans göstermesi ve maliyeti düşürerek, her türlü uygulamaya uyumlu olabilmesidir. Ancak motorların klasik kontrol algoritmaları yerine karmaşık kontrol algoritmaları ile kontrolü söz konusu olduğunda, mikroşlemciler zayıf kalmaktadır. İşlem sayısı arttıkça mikroşlemci yavaşlamakta, bu da kontrol edebileceği sistemlerin performansını etkilemektedir. Bu gibi sorunlara çözüm olarak, mikroşlemcilerin paralel çalışması, kontrol edilen sistemin belirli aşamalarında ayrı işlemcilerin kullanılması yoluna gidilmişse de, hem böyle bir sistemin kontrolünün ve uygulanabilirliğinin zor olması, hem de maliyet artışı DSP'leri üstün hale getirmiştir. Şekil 2.20'de motor kontrol sisteminin genel birimleri gösterilmiştir.



Şekil 2.20 Motor kontrol sistem birimleri.

Kontrol edilecek sistemin özelliklerine uygun bir güç elektroniği devresi kullanılmalıdır. Alternatif akım şebekesinden beslenme durumunda güç akışını kontrol etmek amacıyla güç kaynağı katı kullanılmaktadır. Alternatif gerilim, doğrultucu ile doğrultulduktan sonra eğer gerekiyorsa gerilimin genliğini ayarlamak için doğru akım kıyıcısı kullanılabilir. Daha sonra kullanılan motora göre güç elektroniği güç katı

devresi gelmektedir. Asenkron ve fırçasız doğru akım motor kontrolünde inverter kullanılır. Güç katının yapısı nasıl olursa olsun devrede kullanılan güç elektroniği anahtarlarını sürmek için sürücü devreye gerek vardır. Burada anahtarlama elemanı olarak Insulated Gate Bipolar Transistor (IGBT) kullanılmıştır. Gerilim kontrollü olan bu ve bunun gibi kontrol elemanların yüksek anahtarlama frekanslarına ulaşması, devrenin PWM işaretinin daha yüksek frekanslarda elde edilmesini sağlamaktadır (Üstün 2001).

Denetleyici yapısı ise, sistemden bağımsızdır, sisteme göre değişen sadece işlemcinin programıdır. Bu esnek yapı sayesinde, değişik uygulamalar da donanım yapısında bir değişiklik yapmadan sadece yazılımı değiştirerek kullanılabilir. Tasarlanan kontrol sisteminin yapısından bağımsız olarak, gerçekleştirilen yazılımın belirli bölümleri aynı kalmaktadır. Kurulan algoritmanın denetleyici sistemine hangi dil kullanılarak programlanacağı bir diğer tasarım aşamasıdır. DSP yazılım sistemleri, makine kodu yanında C dilini desteklemekte, böylece programın bu dilde yazılıp, işletilmesine olanak sağlamaktadır. Yalnız sistemin hızından tam olarak yararlanma durumu için, gerçek zamanda kontrol uygulamalarında işlemcinin makine dilinde programlanması tercih edilmelidir. Düşük seviyeli diller yardımıyla, işlemcinin bütün mimari üstünlükleri göz önüne alınıp, program gerçekleştirilmelidir.

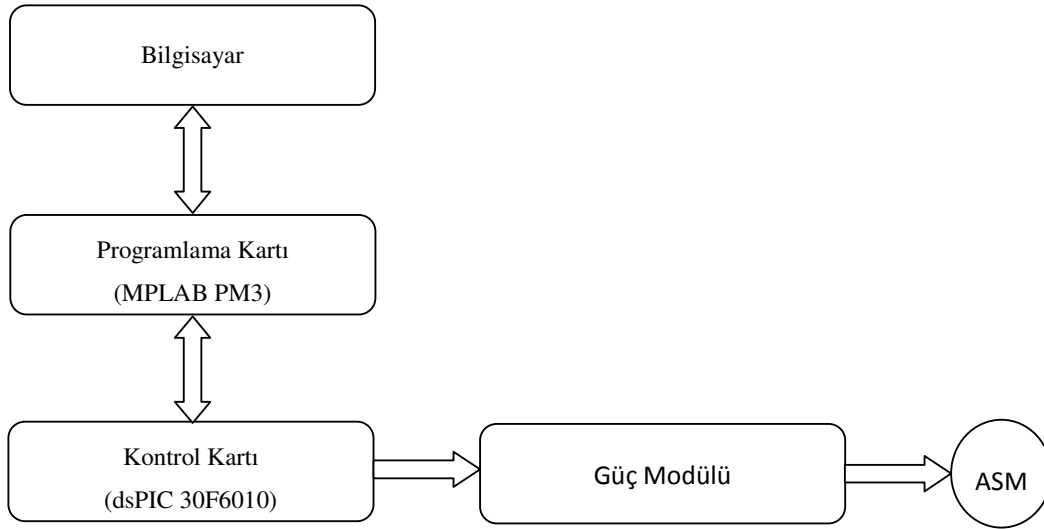
Kontrol sistemlerinde, kontrol edilen sistemin davranışını düzenlemek amacıyla denetleyiciler kullanılmaktadır. Klasik kontrol yapısı olarak sistemlerde en çok kullanılan yapılardan biri oranlama ve bütünleşmiş etme (PI) yapısıdır. Sistemin tam ve doğru olarak kontrolünü gerçekleştiren bu yapı, istenen kontrol cevabını parametrelerini değiştirerek sağlamaktadır. Yapının bu kadar geniş uygulama alanı bulmasının sebebi, basit bir yapıda olması ve kolaylıkla sisteme uyumlu hale getirilebilmesidir.

3. MATERYAL VE METOT

Bu çalışmada, vektör kontrollü asenkron motorun hız bilgisinin sensör kullanmadan YSA ile tahmin edilmesi amaçlanmıştır. Bunun için Microchip firmasına ait olan motor kontrol geliştirme kartı ile çalışılmıştır. Aşağıda bu amaçla kullanılan dsPIC mikroişlemcisi, kontrol kartı ve asenkron motor tanıtılacaktır.

3.1 Materyal

Uygulaması yapılan sistemin blok diyagramı Şekil 3.1’de verilmiştir.



Şekil 3.1 Uygulama devresinin blok Şeması.

3.1.1 Asenkron motor

Uygulamada kullanılan asenkron motorun etiket bilgileri Çizelge 3.1’de verilmiştir.

Çizelge 3.1 Asenkron motorun etiket bilgileri.

GAMAK		4404354284			
3 ~ MOT		TYPE AGM71 2b		EFF	
I.C.L.F		IP 55	B3	S1	CE
V	Hz	A	kW	Cosφ	1/min
Δ 220		2,3			
Y 380	50	1,34	0,55	0,84	2780

3.1.2 DSP programlama ve motor kontrol kartı

Uygulamada kullanılan programlama kartı MPLAB PM3 Microchip firmasının dsPIC uygulamaları için geliştirdiği bir settir. Bu cihaz üzerindeki USB arabirimi aracılığıyla bilgisayardan dsPIC programlanabilir ve hafızasındaki veriler bilgisayara kolayca alınabilmektedir. Resim 3.1’de cihaz üzerinde ki durumunu gösteren ledler, butonlar, LCD ekran ve harici programlama çıkışları görülmektedir.



Resim 3.1 MPLAB PM3 programlama cihazı (İnt. Kyn. 1).

Kontrol kartı üzerinde 2 adet trimpot, LCD ekran, ledler ve butonlar mevcuttur. Bunların dışında harici güç kaynağı ve haberleşme port çıkışları vardır. Uygulamada kullanılan motor kontrol kartı ve güç modülü Resim 3.2’de gösterilmiştir.



Resim 3.2 Motor kontrol geliştirme kartı (İnt. Kyn. 1).

3.1.3 dsPIC30F6010 DSP mikroişlemcisi genel özellikleri

Yüksek performanslı, geliştirilmiş merkezi işlem birimi:

- Modifiye edilmiş Harvard mimarisi,
 - C derleyicisi en iyi şekilde kullanan komut seti mimarisi,
 - Esnek adresleme modlu 84 temel komut,
 - 24 bit genişliğinde komut, 16 bit genişliğinde veri yolu,
 - 144 kbyte dahili flash program alanı,
 - 8 kbyte dahili RAM,
 - 4 kbyte kalıcı veri EEPROM,
 - 30 MIP'e kadar çalışma alanı,
- DC de 40 MHz harici saat girişi,
PLL etkinliğiyle (4x,8x,16x) 4 MHz – 10 MHz osilatör girişi,

- 44 kesme kaynađı
- 5 dıř kesme kaynađı
- Her bir kesme iin 8 kullanıcı seilebilir ncelik dzeyleri,
- 16x16 bit alıřma register dizisi.

DSP cihaz zellikleri

- Modlo ve bit tersleme modları,
- İki adet 40 bit geniřliđinde toplayıcı,
- 17-bit x 17-bit tek evrimde kesirli veya tam arpma donanımı,
- ift veri alıř veriři,
- ± 16 bit tek evrimde kaydırma.

evre birimi zellikleri

- Yksek akım alıcı/kaynak giriř ıkıř pinleri: 25 mA / 25 mA,
- 5 adet 16 bitlik zamanlayıcı/sayıcı; isteđe gre 32 bitlik zamanlayıcı modl iinde 16 bitlik zamanlayıcı ifti,
- 16 bitlik yakalayıcı (capture) giriř fonksiyonu,
- 16 bitlik karřılařtırıcı/PWM ıkıř fonksiyonu,
- 3 telli SPI modl (4 ereve modu ile desteklenen),
- oklu master/slave modu ve 7 bit/ 10 bitlik adreslemeyle desteklenen I²C modl,
- FIFO arabellekli 2 UART modl,
- 2 CAN modl.

Motor Kontrol PWM modl zellikleri

- 8 PWM ıkıř kanalı,
Tmleyici veya bađımsız ıkıř modları,
- Kenar veya merkez hizalı modlar,
- Atanmıř zaman tabanı,
- Programlanabilir ıkıř polaritesi,
- Tmleyici mod iin l zaman kontrol,
- Manuel ıkıř kontrol,
- A/D dnřtrmeler iin tetikleme devresi.

Quadratik enkoder arayz modl zellikleri

- A,B fazı ve indeks pals girişi,
- 16 bitlik yukarı / aşağı konum sayacı,
- Sayma yön durumları,
- Pozisyon ölçüm durumu(x2 ve x4),
- Programlanabilir dijital giriş gürültü filtreleri
- Alternatif 16-bit zamanlayıcı / sayıcı modu,
- Sayıcı ters dönme / taşma durumunda kesme.

Analog özellikler

- 10 bit analog – dijital çevirici,
500 Ksps dönüştürme oranı
16 bit giriş kanalı
Uyku ve boş durum süresince dönüştürme kullanırlığı,
- Programlanabilir düşük voltaj yakalama,
- Programlanabilir Brown – out reset.

Özel mikro kontrolör özellikleri:

- Arttırılabilir flash memory,
100K endüstriyel sıcaklıkta 1000 silme / yazma çevrimi (min.)
- Veri EEPROM hafızası,
1M endüstriyel sıcaklıkta 100000 silme / yazma çevrimi (min.)
- Yazılım kontrolü altında kendi kendini programlama,
- Power-on reset (POR), Power-up timer (PWRT) ve Osilatör Start-up Timer (OST),
- Güvenli çalışma için dahili küçük güçlü RC osilatörlü esnek güvenlik zamanlayıcısı,
- Bozulmaya dayanıklı saat izleme işletimi,
- Programlanabilir kod koruma,
- In-Circuit Serial Programming™ (ICSP™) programlama kabiliyeti,
- Seçimli güç yönetim modu,
- Uyku, boşta ve alternatif saat modları.

CMOS teknolojisi

- Düşük güçlü, yüksek hızlı flash teknolojisi,

- Geniş çalışma gerilim aralığı (2,5V - 5,5V),
- Endüstriyel ve geniş sıcaklık aralıkları,
- Düşük güç tüketimi.

dsPIC'in pinleri ile ilgili resim ve bilgi EK-1'de verilmiştir.

3.1.4 Sistem yazılımının tanıtılması

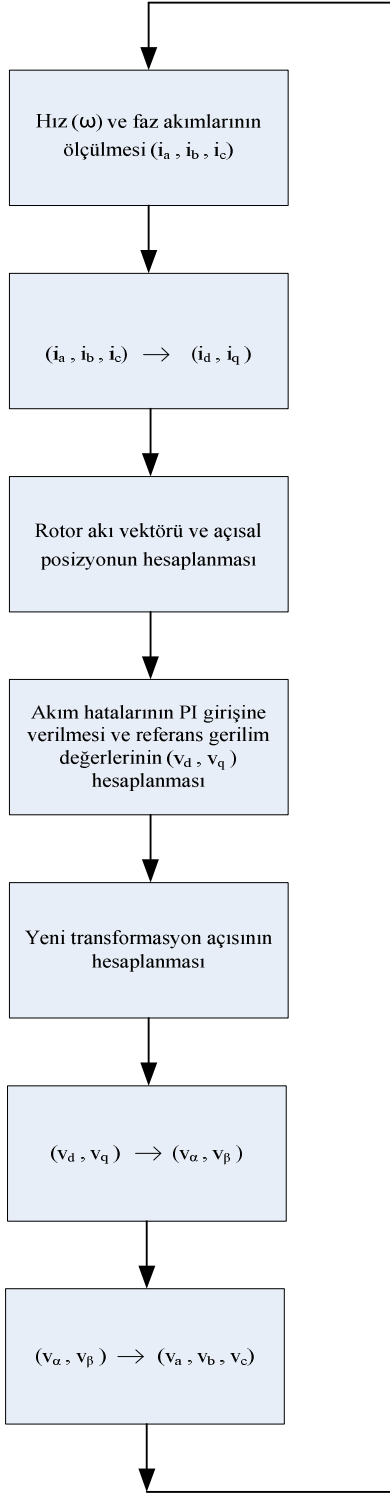
Kontrol işlemini gerçekleştiren program, MPLAB IDE programı aracılığı ile C dilinde yazıldıktan sonra denetimi ve derlemesi yapıp USB hattından MPLAB PM3 programlama cihazı üzerinden mikroişlemciye yüklenmesiyle çalıştırılmaktadır. Yine MPLAB IDE programı ile işlemciye kaydedilmiş veriler bilgisayar ortamına alınmaktadır.

Uygulamada kullanılan yazılım, MICROCHIP firmasına ait örnek yazılımdır.

3.2 Metot

3.2.1 Kontrol sistemi yazılımının tanıtılması

Program genel olarak, analog girişlerinden aldığı akım ve gerilim verileri ile enkoderden aldığı hız bilgisini giriş verileri olarak kullanmaktadır. Vektör kontrolü algoritmasını uygulayarak yeni PWM oluşturup güç modülüne göndermektedir. Programın akış diyagramı Şekil 3.2'de gösterilmiştir.



Şekil 3.2 Program akış şeması.

3.2.2 Vektör kontrollü asenkron motor verilerinin elde edilmesi

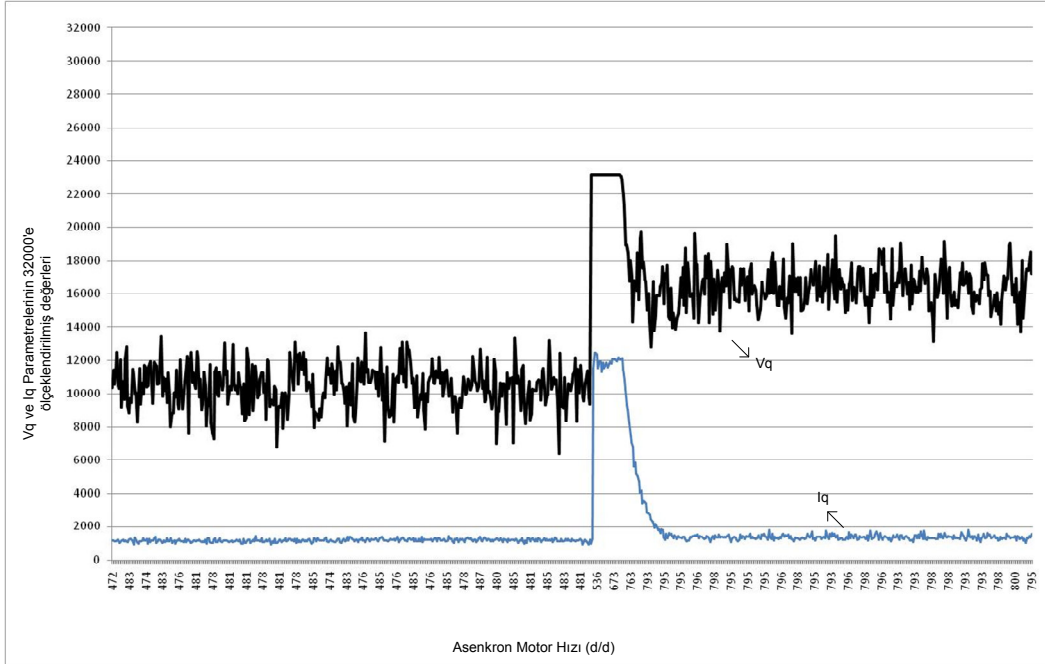
Vektör kontrollü bořta alıřan asenkron motor 500 d/d hızdan bařlayarak 1000 d/d hıza kadar 10'ar devirlik artıřlarla alıřtırılmıřtır. Her alıřmada Hız - I_q ve Hız - V_q veri iftleri alınmıřtır. Bu verilerin iřlemciye kaydedilmesi iin yazılımın ierisine komut satırları ilave edilmiřtir. Kaydetme iřleminin bařlaması iin istenilen bir butona basılması řartı konulmuřtur. Sistem alıřırken vektör kontrol modunda 480 d/d hızla dđnmektedir. İstenilen bir anda butona basılarak ilk 500 veri kaydedildikten sonra hız otomatik olarak istenilen hıza artmakta ve diđer 500 veri de kaydedilmektedir. Bu verilerin rnekleme zamanı 4 ms dir.

4. BULGULAR

4.1 Deney Sonuçları

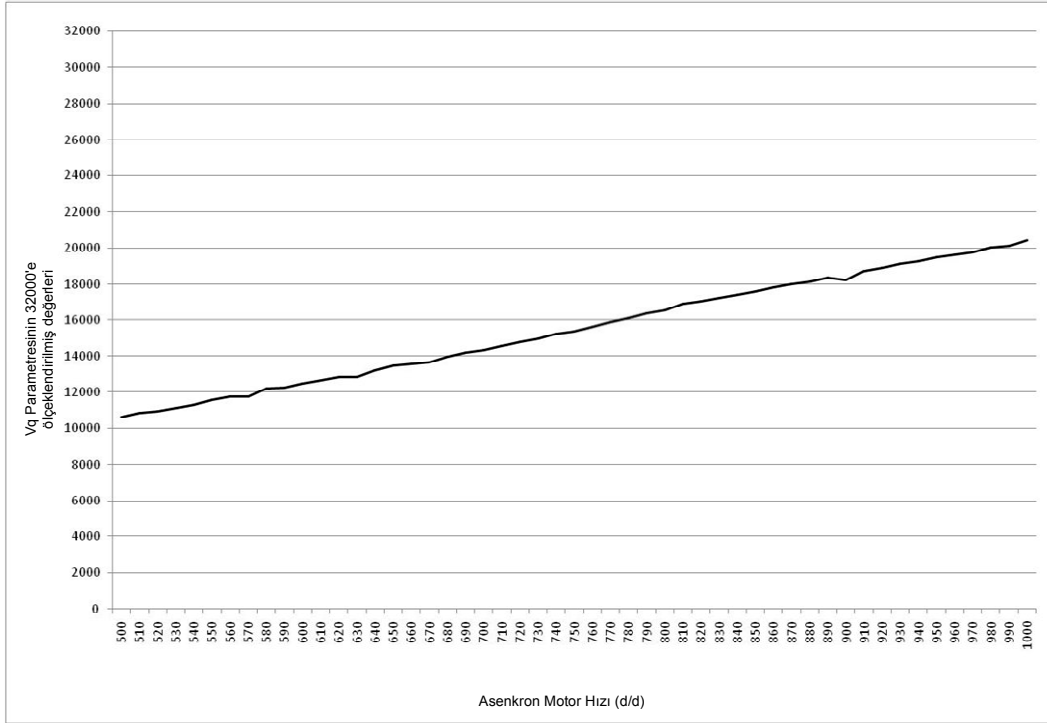
Farklı hız kademelerinde çalıştırılan motordan elde edilen Hız – I_q ve Hız – V_q verilerinden toplam 50 veri grubu alınmıştır. Bu elde edilen V_q ve I_q verilerinin ortalamaları alınarak hıza göre değişimi ile ilgili 4 veri grubu oluşturulmuştur.

Bu verilerin arasından motor hızının 800 d/d artışı sırasındaki I_q ve V_q değişimi ve bu değişimin arasındaki bağıntı Şekil 4.1’de gösterilmiştir. Diğer devir verilerine ait örnek şekiller EK-2’de verilmiştir.



Şekil 4.1 Hıza göre anlık I_q ve V_q değişimi.

Devir hızı farklı olduğunda, bu verilerler arasında sayısal olarak seviye farkı vardır. Bunun için EK-2’deki veriler karşılaştırıldığında aradaki fark görülmektedir. Elde edilen ve tamamı Şekil 4.1’deki ilişkiye bağlı olarak çıkan verilerin ortalamaları alındığında daha lineer grafikler elde edilmiştir. Bu elde edilen verilere ait değişim Şekil 4.2’de gösterilmiştir. Diğer verilere ait grafikler EK-3’te verilmiştir.



(a)



(b)

Şekil 4.2 Ortalaması alınan (a) Vq ve (b) Iq parametrelerinin hıza göre değişimleri .

Şekil 4.2'deki grafiklerden anlaşılacağı gibi hız ile Vq ve Iq parametreleri ile lineer bir bağ vardır. Bu şekilde 4 veri grubu elde edilmiştir. Bu verilerin bir grubu ile YSA eğitilmiş diğer grup ile testler yapılmıştır. Eğitimde kullanılan veriler Çizelge 4.1'de verilmiştir.

Çizelge 4.1 YSA'nın eğitimi için kullanılan veriler.

Hız (d/dk)	Vq	Iq	Hız (d/dk)	Vq	Iq
500	10696,1	1163,13	750	15442,36	1269,61
510	10913,26	1167,74	760	15368,38	1299,24
520	10978,29	1168,72	780	15935,34	1288,33
530	11102,31	1172,81	790	16342,53	1321,67
550	11615,22	1209,52	810	16714,49	1363,19
560	11829,56	1195,33	820	16889,13	1319,54
570	11779,87	1173,65	830	17029,32	1334,25
590	12237,28	1202,06	840	17059,48	1357,85
610	12707,98	1207,65	850	17292,82	1359,06
620	13156,5	1218,07	860	17538,94	1306,12
630	12988,33	1270,22	880	17758,64	1364,38
640	13398,96	1267,22	890	18074,85	1318,5
660	13635,37	1247,71	900	17844,09	1355,26
670	13623,12	1257,44	910	18472,37	1351,47
680	13930,32	1240,82	920	18572,41	1310,43
690	14368,58	1271,96	930	19029,92	1343,77
700	14547,13	1266,25	940	18919,34	1364,04
720	14751,32	1263,9	960	19215,21	1371,74
730	14880,76	1310,61	970	19394,23	1350,02
740	15232,35	1303,01	1000	20283,67	1432,19

YSA'da test için kullanılacak veriler ise 2. veri grubundan seçilmiştir. Test için 10 veri seçilmiştir. Bu veriler Çizelge 4.2'de verilmiştir.

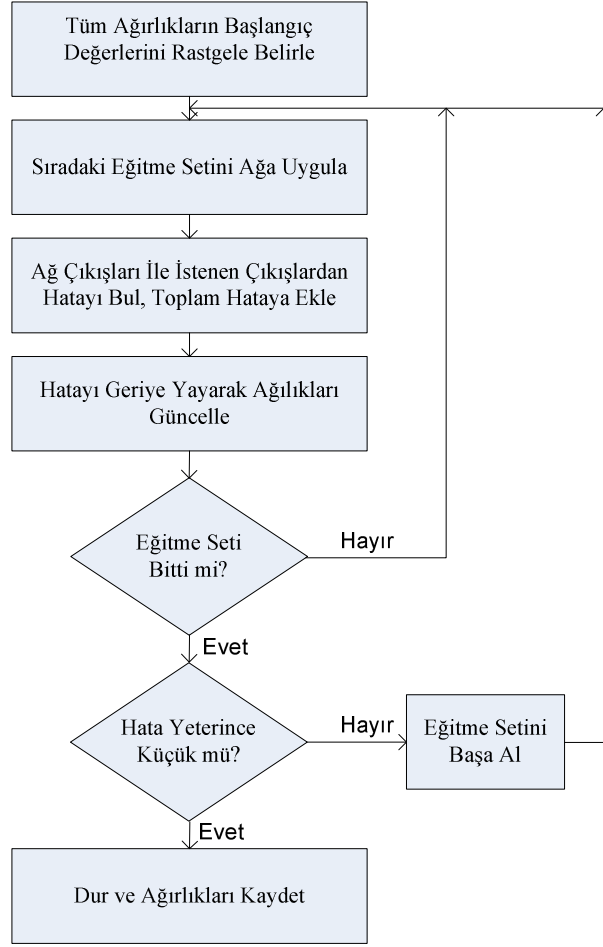
Çizelge 4.2 YSA testinde kullanılan veriler.

Hız (d/dk)	Vq	Iq	Hız (d/dk)	Vq	Iq
510	10925,76	1132,77	760	15366,56	1320,98
540	11381,67	1180,65	870	17606,56	1363,98
580	12104,86	1187,31	910	18273,08	1344,23
610	12804,96	1261,86	970	19343,09	1355,52
670	13793,27	1263,88	1000	20203,5	1412,27

4.2 YSA ile Sistemin Modellenmesi

Kontrol sistemini modellemek için kullanılan YSA Bölüm 2’de anlatılmıştır. Kullanılan YSA modeli giriş ve çıkış katmanları arasında birden fazla katmanın kullanıldığı çok katmanlı perceptron modelidir. Eğitim algoritması olarak öğreticili eğitimden hatanın geriye yayılması algoritması kullanılmıştır. Hatanın geriye yayılması algoritması, karesi alınmış hata fonksiyonunu minimize eden kodlu bir algoritma olup ve geliştirilmiş delta kuralını eğitim için kullanmaktadır.

Bu YSA modelinin eğitim işlemi Şekil 4.3’te görülen akış diyagramında görülmektedir. Şekil 4.3’teki akış diyagramına göre YSA’nın eğitim programı C++ programlama dilinde yazılmış ve EK-4’de verilmiştir.



Şekil 4.3 C++’ de yazılan YSA eğitme programının akış diyagramı.

YSA ile sistemin modelinin çıkartılması aşağıdaki 4 aşamadan oluşmaktadır.

1. Sistemin giriş-çıkış bilgilerinin elde edilmesi,
2. YSA yapısının seçimi,
3. Eğitim işleminin gerçekleştirilmesi,
4. Sistemin YSA modelinin uygunluğunun testi.

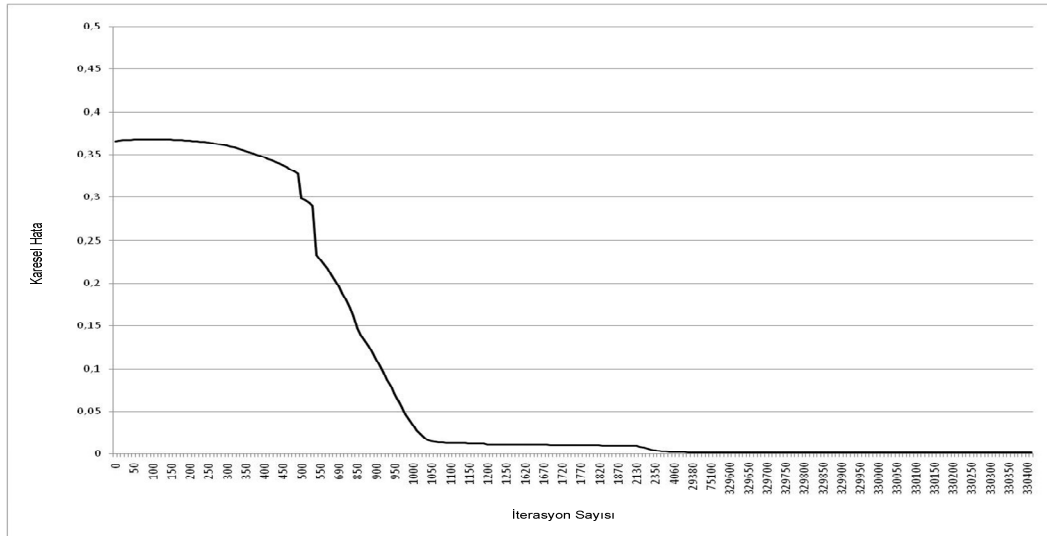
Sistemi modelleyen YSA parametreleri aşağıda verilmiştir.

1. Giriş sayısı 2 dir.
2. Çıkış sayısı 1 dir.

3. Katman sayısı 1 dir.
4. Katmandaki hücre sayısı 4 tür.
5. Katman aktivasyon fonksiyonu Sigmoid dir.
6. Maksimum iterasyon sayısı 500000 dir.
7. Öğrenme katsayısı 0,7 dir.
8. Momentum katsayısı 0,9 dur.

YSA yapısının katman sayısı ve her katmandaki hücre sayısını seçmede her hangi kesin bir kıstas bulunmadığından deneme yolu izlenerek katman sayısı ve hücre sayılarına karar verilmiştir. Benzer şekilde öğrenme ve momentum katsayıları da daha önce yapılan çalışmalarda tecrübelerle dayanılarak yukarıda ki şekilde alınmıştır. Giriş verileri 5'e normalize edilmiştir. Çıkış ise 0,1 ila 0,99 arasındadır.

YSA yapısı ile eğitim işlemindeki hatanın değişimi Şekil 4.4'te verilmiştir. Karese hatanın 500000. iterasyonda 0,0009' a indiği görülerek eğitime son verilmesine karar verilmiştir.

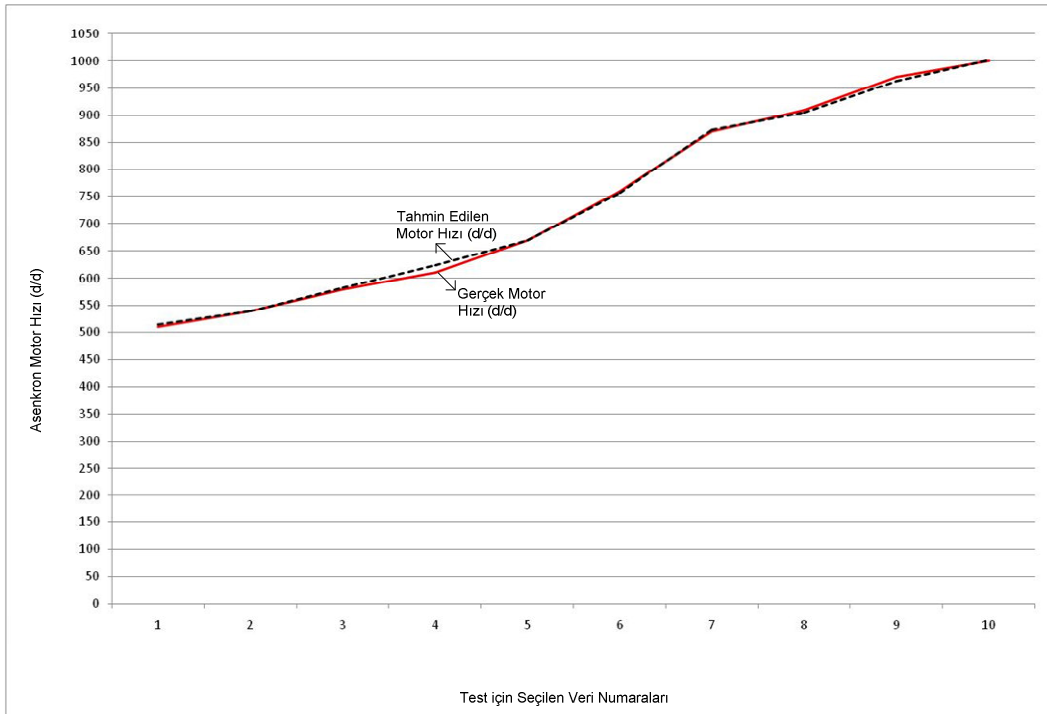


Şekil 4.4 Eğitim sırasında hatanın değişimi.

YSA ile eğitim işlemi tamamlandıktan sonra 2. gruptan alınan veriler test edilmiştir. Elde edilen sonuçlar ile gerçek değerler Çizelge 4.3'te ve Şekil 4.5'te verilmiştir.

Çizelge 4.3 Hızın YSA ile tahmin edilen değerleri ile gerçek değerleri.

YSA ile elde edilen Sonuçlar	YSA ile elde edilen verilerin Hız (d/dk) olarak karşılığı	Gerçek veriler	Gerçek verilerin Hız (d/dk) olarak karşılığı
0,126076	514,64	0,1178	510
0,169747	539,18	0,1712	540
0,246175	582,12	0,2424	580
0,321902	624,66	0,2958	610
0,403430	670,46	0,4026	670
0,557564	757,05	0,5628	760
0,766499	874,43	0,7586	870
0,822947	906,15	0,8298	910
0,924769	963,35	0,9366	970
0,994801	1002,69	0,99	1000



Şekil 4.5 Hızın YSA ile tahmin edilen değerleri ile gerçek değerleri.

5. TARTIŞMA VE SONUÇ

Bu çalışmada, vektör kontrolü yapılan asenkron motor farklı devirlerde çalıştırılmıştır. Asenkron motorun vektör kontrolünde moment bileşenleri olan V_q ve I_q parametrelerinin hıza bağlı olarak durumları izlenmiş ve ilgili veriler alınmıştır. Hızın artışına bağlı olarak izlenen bu iki parametre değerlerinin de orantılı arttığı gözlemlenmiştir.

Vektör kontrolünde geri dönüt için gerekli olan hız bilgisinin bu iki parametreden yararlanarak elde edilebileceği yapılan çalışmalar sonrasında görülmektedir. Nitekim Bölüm 4'te verilen Çizelge 4.3'te görüldüğü gibi, YSA uygulaması ile tahmin edilen hız bilgileri gerçek değerlerine yakın ve doğru çıkmıştır. Bu parametreler vektör kontrolünde kullanıldığı için ek eleman kullanımına da ihtiyaç yoktur. Yazılımın içerisinde bu iki verinin değişimi izlenerek hızın doğru bir şekilde tahmini gerçekleştirilmiştir.

Uygulamalarda hız tahmini, farklı algoritmalar kullanılarak yapılabilmektedir. Bu tür uygulamalarda matris işlemleri yoğun olarak kullanıldığı için tüm vektör kontrolü algoritmasının çevrim süresinin 15 katına kadar zamana ihtiyaç duyulmaktadır. Bu istenmeyen bir durumdur; örnekleme zamanını uzatabilmektedir. YSA algoritması bu zamanı da kısaltmaktadır.

YSA'nın eğitimi ve testi, sistem yüksüz ve hız sabitlendikten sonra ki alınan veriler ile gerçekleştirilmiştir. İleride yapılacak olan çalışmalarda farklı yük durumlarda çalışan asenkron motorun hız tahminin yapılabilmesi düşünülmektedir.

6. KAYNAKLAR

Aydeniz, M. G., 2005, “Asenkron Motorların Hız Algılayıcısız Kontrolünde Yeni Bir Algoritmanın Geliştirilmesi ve Uygulaması”, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Bayındır, N. S., Narlı, E., 1991, “Dolaylı Alan Yönlendirmeli Bir Asenkron Motor Hız Kontrol Sistemi Simülasyonu”, Elektrik Mühendisliği IV.Ulusal Kongresi.

Bingöl, O., 2005, “Yapay Sinir Ağı ile Modellenen Alan Yönlendirmeli Bir Asenkron Motorun Üç Seviyeli Evirici ile Hız Denetimi”, Doktora Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Bodson, M., Chiasson, J., 2002, “A Comparison of Sensorless Speed Estimation Methods for Induction Motor Control”, Proceedings of the American Control Conference Anchorage, AK.

Bose, B. K., 2006, “Power Electronics and Motor Drives Advances and Trends”, Condra Chair of Excellence in Power Electronics/Emeritus, The University of Tennessee, Ademic Press is an imprint of Elsevier.

Gökbulut, M., Dandıl, B., 2005, “Asenkron Motorların Sinirsel-Bulanık Denetleyici ile Uyarlamalı Denetimi”, Gazi Üniv. Müh. Mim. Fak. Der. Cilt 20, No 2, 145-153.

Güleç, K., 1999 , “Asenkron Motorun DSP (Sayısal İşaret İşleyici) Tabanlı Bir Kontrol Sistemi Kullanarak YSA (Yapay Sinir Ağları) ile Performansının Arttırılması”, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Hancock, N.N., 1974, “Matrix Analysis of Electrical Machinery”, Pergamon Press.

- Holtz, J., 1993, "Speed Estimation and Sensorless Control of AC Drives", Industrial Electronics, Control, and Instrumentation, Proceedings of the IECON, International Conference, vol. 2 Pages 649-654.
- Hovingh B.W., Keerthipala, W.L., Yan, W. Y., 2004, "Sensorless Speed Estimation of an Induction Motor in a Field Orientated Control System", School of Electrical and Computer Engineering Curtin University of Technology, Australia.
- Jeon, S.H., Oh, K.K., Choi, J.Y., 2002, Flux Observer with Online Tuning of Stator and Rotor Resistances for Induction Motors, IEEE Trans. Ind. Electron. 49(3): 653-664.
- Joetten, R., Maeder, G., 1983, "Control Methods for Good Dynamic Performance Induction Motor Drives Based on Current and Voltage as Measured Quantities", IEEE Transactions on Industry Applic., vol. IA-19, no. 3, pp. 356-363.
- Krause, P.C., 1986, "Analysis of Electric Machinery", Mc Graw-Hill Series in Electrical Engineering.
- Lu, H., Hung, T., Tsai, C., 2000, "Sensorless Vector Control of Induction Motor Using Artificial Neural Network", IEEE International Symposium on Volume 2, Page(s):489 - 492 vol. 2.
- Marzi, H., 2007, "Using AC Motors in Robotics, International Journal of Advanced Robotic Systems", Vol. 4, No.3, ISSN 1729-8806, pp. 365-370.
- Messaoudi, M., Sbita, L. and Abdelkrim, M. N., 2007, "Robust Nonlinear Observer For States and Parameters Estimation and On-Line Adaptation of Rotor Time Constant in Sensorless Induction Motor Drives", International Journal of Physical Sciences Vol. 2 , pp. 217-225.
- Nasar, S.A., Boldea, I., 1992, "Vector Control of AC Drives", CRC Pres, New York.

Öztürk, N., 2000, “Alan Yönlendirmeli Asenkron Motora Bulanık Kontrolün Uygulanması”, Y. Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

Peterson, B., 1996 , “Induction Machine Speed Estimation Observations on Observers”, Lund Institute of Technology (LTH), Sweden.

Ross, D., Theys, J., 2004, “Using the dsPIC30F for Vector Control of an ACIM”, Microchip Technology Inc., DS00908A.

Sustek, P., Stekl, P., 2007, “Washing Machine Three-Phase AC-Induction Direct Vector Control Based on MC56F8013”, Freescale Semiconductor, Inc., AN3476, Rev.1.

Şahin, K., 2006, “Asenkron Motorların Dinamik Yapay Sinir Ağları ile Hız Kontrolü”, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.

Üstün, S. V., 2001, “Asenkron Motorun DSP (Sayısal İşaret İşlemci) Temelli Kontrolünde Yapay Sinir Ağları ve Genetik Algoritmalar Kullanılarak PI Katsayılarının Optimizasyonu”, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Varghese, G.C., Sanders S.R., 1998, “Observers for Flux Estimation in Induction Machines”, IEEE Trans. Ind. Electron., 35(1): 85-94.

Vas, P., 1990, “Vector Control of AC Machines”, Oxford: Clarendon Press, New York.

Vas, P., 1998, “Sensorless Vector and Direct Torque Control” , Oxford University Press, New York.

6.1 İnternet Kaynakları

1- <http://www.microchip.com>

Erişim Tarihi

20.01.2008

7. ÖZGEÇMİŞ

Adı Soyadı : Mehmet DEDE

Doğum Yeri : Sandıklı/AFYON

Doğum Tarihi : 02.01.1982

Medeni Hali : Bekar

Yabancı Dili : İngilizce

Eğitim Durumu (Kurum ve Yıl)

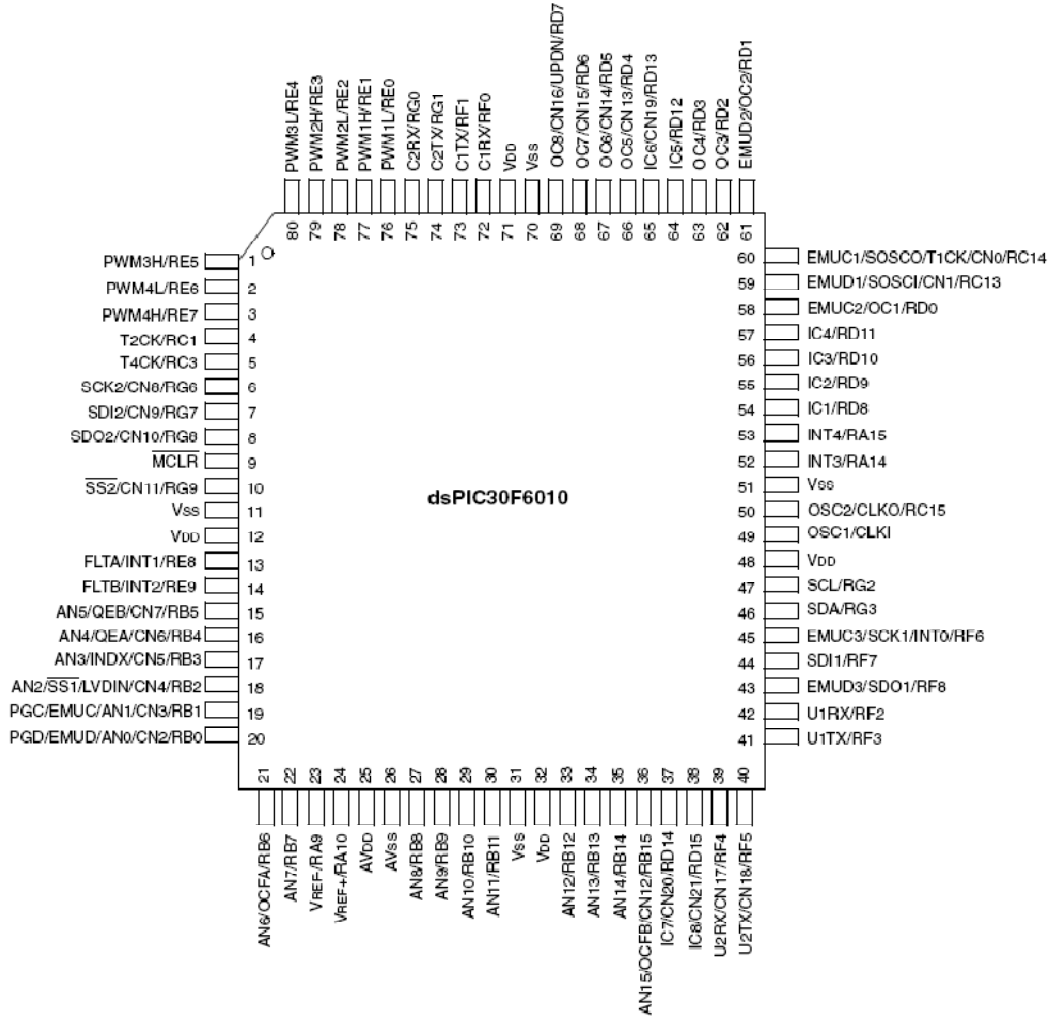
Lise : Sandıklı Endüstri Meslek Lisesi, 1999

Lisans : Gazi Üniversitesi, Teknik Eğitim Fakültesi, 2005

Yüksek Lisans

8. EKLER

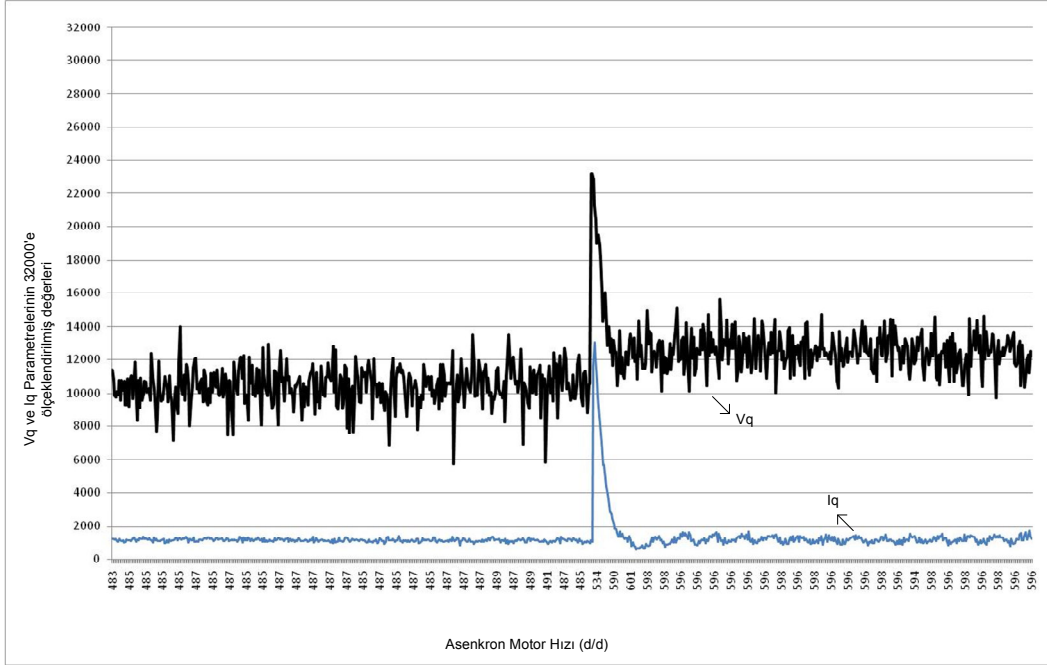
EK-1 Uygulamada kullanılan dsPIC30F6010 pinlerinin fonksiyonları.



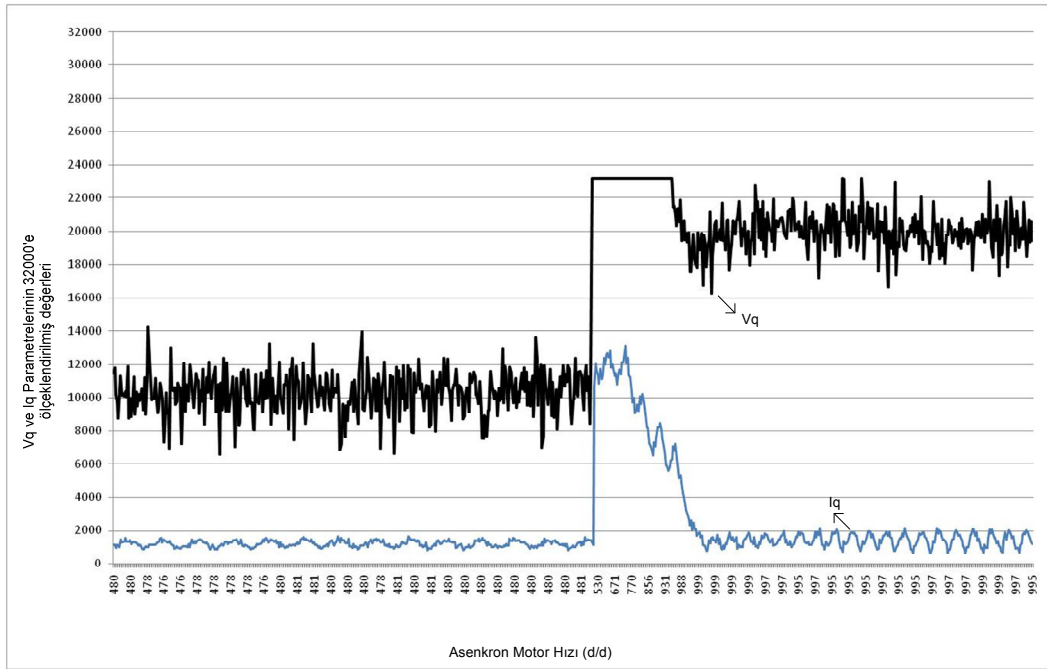
Pin	Fonksiyon	Tipi	Uygulamadaki Kullanım Alanı
PORTA			
RA9	VREF-	O	LED1,D6
RA10	VREF+	O	LED2,D7
RA14	INT3	O	LED3,D8
RA15	INT4	O	LED4,D9
PORTB			
RB0	PGD/EMUD/AN0/CN2	AI	Faz 1 akımı/Cihaz Programlama
RB1	PGC/EMUC/AN1/CN3	AI	Faz 2 akımı/Cihaz Programlama
RB2	AN2/SS1/LVDIN/CN4	AI	Uygulamada kullanılmıyor
RB3	AN3/INDX/CN5	I	QEI Index
RB4	AN4/QEA/CN6	I	QEI A
RB5	AN5/QEB/CN7	I	QEI B
RB6	AN6/OCFA	AI	Uygulamada kullanılmıyor
RB7	AN7	AI	Pot (VR1)
RB8	AN8	AI	Uygulamada kullanılmıyor
RB9	AN9	AI	Uygulamada kullanılmıyor
RB10	AN10	AI	Uygulamada kullanılmıyor
RB11	AN11	AI	Uygulamada kullanılmıyor
RB12	AN12	AI	Uygulamada kullanılmıyor
RB13	AN13	AI	Uygulamada kullanılmıyor
RB14	AN14	AI	Uygulamada kullanılmıyor
RB15	AN15/OCFB/CN12	O	Uygulamada kullanılmıyor
PORTC			
RC1	T2CK	O	LCD R/W
RC3	T4CK	O	LCD RS
RC13	EMUD1/SOSC2/CN1		ICD2 Haberleşme Ucu
RC14	EMUC1/SOSC1/T1CK/CN0		ICD2 Haberleşme Ucu
RC15	OSC2/CLKO		
PORTD			
RD0	EMUC2/OC1	I/O	LCD D0
RD1	EMUD2/OC2	I/O	LCD D1
RD2	OC3	I/O	LCD D2
RD3	OC4	I/O	LCD D3
RD4	OC5/CN13	O	Uygulamada kullanılmıyor
RD5	OC6/CN14	O	Uygulamada kullanılmıyor
RD6	OC7/CN15	O	İzleme için PWM çıkışı
RD7	OC8/CN16/UPDN	O	İzleme için PWM çıkışı
RD8	IC1	I	Uygulamada kullanılmıyor
RD9	IC2	I	Uygulamada kullanılmıyor
RD10	IC3	I	Uygulamada kullanılmıyor
RD11	IC4	O	Demo kartında PWM çıkışı aktif
RD12	IC5		Uygulamada kullanılmıyor
RD13	IC6/CN19	O	LCD E
RD14	IC7/CN20		Uygulamada kullanılmıyor
RD15	IC8/CN21		Uygulamada kullanılmıyor

Pin	Fonksiyon	Tipi	Uygulamadaki Kullanım Alanı
PORTE			
RE0	PWM1L	O	FAZ 1 L
RE1	PWM1H	O	FAZ 1 H
RE2	PWM2L	O	FAZ 2 L
RE3	PWM2H	O	FAZ 2 H
RE4	PWM3L	O	FAZ 3 L
RE5	PWM3H	O	FAZ 3 H
RE6	PWM4L	O	Uygulamada kullanılmıyor
RE7	PWM4H	O	Uygulamada kullanılmıyor
RE8	FLTA/INT1	I	Güç modulu hata çıkışı (0 aktif)
RE9	FLTB/INT2	O	Güç modulu hata reset (1 aktif)
PORTF			
RF0	C1RX	I	Uygulamada kullanılmıyor
RF1	C1TX	O	Uygulamada kullanılmıyor
RF2	U1RX	I	Uygulamada kullanılmıyor
RF3	U1TX	O	Uygulamada kullanılmıyor
RF4	U2RX/CN17	I	Uygulamada kullanılmıyor
RF5	U2TX/CN18	O	Uygulamada kullanılmıyor
RF6	EMUC3/SCK1/INT0	I	Uygulamada kullanılmıyor
RF7	SDI1	I	Uygulamada kullanılmıyor
RF8	EMUD3/SDO1	O	Uygulamada kullanılmıyor
PORTG			
RG0	C2RX	O	Uygulamada kullanılmıyor
RG1	C2TX	O	Uygulamada kullanılmıyor
RG2	SCL	I/O	Uygulamada kullanılmıyor
RG3	SDA	I/O	Uygulamada kullanılmıyor
RG6	SCK2/CN8	I	BUTON 1 (S4) (0 Aktif)
RG7	SDI2/CN9	I	BUTON 2 (S5) (0 Aktif)
RG8	SDO2/CN10	I	BUTON 3 (S6) (0 Aktif)
RG9	SS2/CN11	I	BUTON 4 (S7) (0 Aktif)

EK-2 Hızın değişimi sırasında anlık I_q ve V_q değişimine ait örnek grafikler.

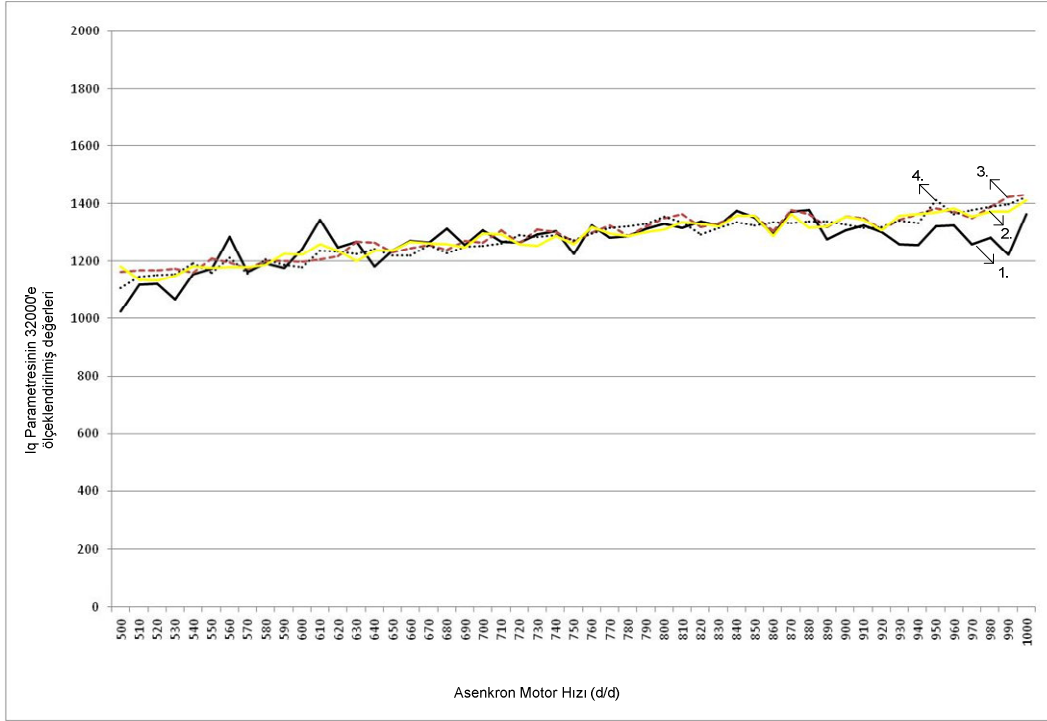


(a) Hızın 600 d/d ya artışı sırasındaki V_q - I_q değişimi

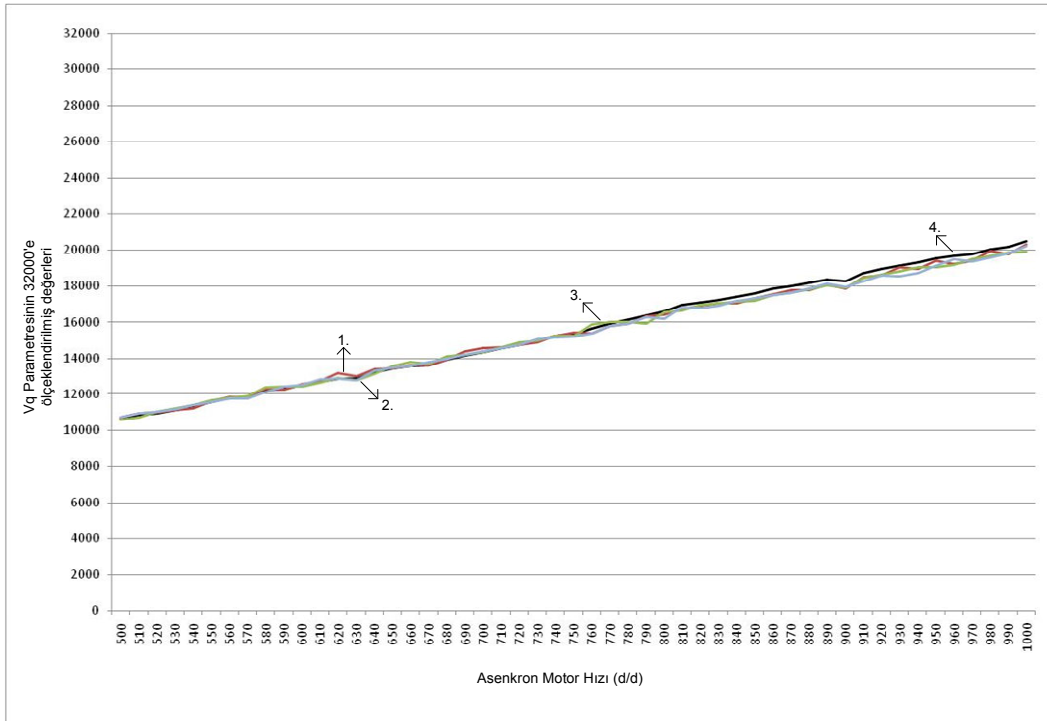


(b) Hızın 1000 d/d ya artışı sırasındaki V_q - I_q değişimi

EK-3 Hızın değişimi sırasında I_q ve V_q ortalama değerlerinin değişimine ait örnek grafikler.



(a) Hızın 1000 d/d ya artışı sırasındaki I_q ortalama değerlerinin değişimi



(b) Hızın 1000 d/d ya artışı sırasındaki V_q ortalama değerlerinin değişimi

EK-4 YSA eğitime çalışmalarının yapıldığı program.

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <process.h>
#include <alloc.h>
#include <ctype.h>
#ifdef VAX
#include <alloc.h>
#endif
#define NMXUNIT 10
#define NMXHLLR 10
#define NMXOATTR 8
#define NMXINP 440
#define NMXIATTR 120
#define SEXIT 3
#define RESTRT 2
#define FEXIT 1
#define CONTNE 0
float eta, alpha, err_curr, maxe, maxep;
float huge *wtptr[NMXHLLR+1];
float huge *outptr[NMXHLLR+2];
float huge *errptr[NMXHLLR+2];
float huge *delw[NMXHLLR+1];
float huge target[NMXINP][NMXOATTR];
float huge input[NMXINP][NMXIATTR],se5,se6;
float huge ep[NMXINP];
float huge outpt[NMXINP][NMXOATTR];
int nunit[NMXHLLR+2],nhlayer,ninput,ninattr,noutattr,sd;
int result,i,m,errorcode;
long int cnt_num,cnt;
int nsnew,nsold;
char task_name[20],dr;
FILE *fp1,*fp2,*fp3,*fopen();
int fplot10;
long randseed=568731L;
char def[30],def1[20];
int mx,my;
void *p;
void git()
{ clrscr();printf("YAPILAN İTERASYON=%ld",cnt);
printf("\n ŞU ANKI HATA   =%f\n",err_curr);
printf("ÇIKMAK İSTİYORMUSUNUZ ?d(dur)-HERHANGİ BİR TUŞA BASINIZ");
if(getch()=='d')cnt=cnt_num-1;
}
/*****Rastgele sayı üretilmesi*****/
int random()
{
randseed=15625L*randseed+22221L;
return((randseed>>16)&0x7FFF);
}
/*****Gizli katmanlar için dinamik hafızada yer ayrılması*****/
init()
{
int len1,len2,i,k;
float *p1,*p2,*p3,*p4;
len1=len2=0;
nunit[nhlayer+2]=0;
for(i=0;i<(nhlayer+2);i++){
len1+=(nunit[i+1])*nunit[i+1];
len2+=nunit[i]+1;
}
p1=(float *) calloc(len1+1,sizeof(float));
p2=(float *) calloc(len2+1,sizeof(float));
p3=(float *) calloc(len2+1,sizeof(float));
```

```

p4=(float *) calloc(len1+1,sizeof(float));
wtptr[0]=p1;
outptr[0]=p2;
errptr[0]=p3;
delw[0]=p4;
for (i=1;i<(nhlayer+1);i++){
wtptr[i]=wtptr[i-1]+nunit[i]*(nunit[i-1]+1);
delw[i]=delw[i-1]+nunit[i]*(nunit[i-1]+1);
}
for (i=1;i<(nhlayer+2);i++){
outptr[i]=outptr[i-1]+nunit[i-1]+1;
errptr[i]=errptr[i-1]+nunit[i-1]+1;
}
for (i=0;i<nhlayer+1;i++){
*(outptr[i]+nunit[i])=1.0;
}
return(0);
}
//***** Ağırlık değerleri başlangıçta rasgele atanır.*****
initwt()
{
int i,j;
for (j=0;j<nhlayer+1;j++){
for (i=0;i<(nunit[j]+1)*nunit[j+1];i++){
*(wtptr[j]+i)=random()/pow(2.0,15.0)-0.5;
*(delw[j]+i)=0.0;
}
}
return(0);
}
//*****YSA mimarisinin belirlenmesi*****
set_up()
{
int i;
eta=0.9;
// printf("\n Alfa momentum katsayısı (default=0.9)? : ");
// scanf("%f",&eta);
eta=0.9;
alpha=0.7;
// printf("\n Epsilon öğrenme oranı (default=0.7)? : ");
//scanf("%f",&alpha);
alpha=.7;
maxe=0.00001;maxep=0.00001;
printf("\nMaksimum toplam hata (default=0.00001)? : ");
printf("\nMaksimum başlangıç hatası(default=0.00001)? : ");
cnt_num=1000;
printf("\nMaksimum iterasyon sayısı (default=1000)? : ");
//scanf("%ld",&cnt_num);
cnt_num=100000;
printf("\nGizli katman sayısı? : ");
scanf("%d",&nhlayer);
for(i=0;i<nhlayer;i++){
printf("\nt %d. gizli katmandaki düğüm sayısı?:",i+1);
scanf("%d",&nunit[i+1]);
}
fplot10=1;
clrscr();
printf("\nBilgisayar hesap yapıyor.");
nunit[nhlayer+1]=noutattr;
nunit[0]=ninattr;
return(0);
}

//*****Eğitim sonunda çıkışların dosyaya atanması*****
dwrite(char *taskname)
{
int i,j,c;
char var_file_name[20];
strcpy(var_file_name,taskname);
strcat(var_file_name,"_v.dat");
if ((fp1=fopen(var_file_name,"w+"))==NULL)
{
perror("\n Veri dosyası açılmıyor");
exit(0);
}
}

```

```

}
fprintf(fp1,"%u %u %u %f %f %u %lu\n",
ninput,noutattr,ninattr,eta,alpha,nhlayer,cnt_num);
for (i=0; i<nhlayer+2;i++){
    fprintf(fp1,"%d ",nunit[i]);
}
fprintf(fp1,"\n");
for (i=0; i<ninput;i++){
    for (j=0; j<noutattr;j++){
        fprintf (fp1,"%f ",outpt[i][j]);
        fprintf (fp1,"\n");
    }
}
if ((c=fclose(fp2))!=0)
    printf("\nFile cannot be closed %d",c);
return(0);
}
//*****Eğitilmiş ağırlıkların dosyaya yüklenmesi*****
wtwrite( char *taskname)
{
    int i,j,c,k;
    char wt_file_name[20];
    strcpy(wt_file_name,taskname);
    strcat(wt_file_name,"_w.dat");
    if ((fp2=fopen(wt_file_name,"w+"))==NULL)
    {
        perror("\n Veri dosyası açılmıyor");
        exit(0);
    }
    k=0;
    for (i=0; i<nhlayer+1;i++){
        for (j=0; j<(nunit[i]+1)*nunit[i+1];j++){
            if(k==8){
                k=0;
                fprintf (fp2,"\n");
            }
            fprintf (fp2,"%f ",*(wtptr[i]+j));
            k++;
        }
    }
    if ((c=fclose(fp2))!=0)
        printf("\n%d.Dosya kapatılmıyor ",c);
    return(0);
}
//*****Çıktıların Üretilmesi *****
void forward(i)
{
    int m,n,p,offset;
    float net;
    for (m=0; m<ninattr;m++){
        *(outptr[0]+m)=input[i][m];
        for (m=1; m<nhlayer+2;m++){
            for (n=0; n<nunit[m];n++){
                net=0.0;
                for (p=0; p<nunit[m-1]+1;p++){
                    offset=(nunit[m-1]+1)*n+p;
                    net += *(wtptr[m-1]+offset)
                        *(*(outptr[m-1]+p));
                }
                *(outptr[m]+n)= 1/(1+exp(-net));
            }
        }
        for(n=0; n<nunit[nhlayer+1];n++)
            outpt[i][n]=*(outptr[nhlayer+1]+n);
    }
}
//*****HATANIN İŞLEMİN SONA ERDİRİLMESİ İÇİN KONTROLÜ*****
int introspective (int nfrom,int nto)
{
    int i,flag;
    if (cnt>=cnt_num) return(FEXIT);
    nsnew=0;
    flag=1;
    for (i=nfrom;(i<nto)&&(flag==1);i++){
        if (ep[i]<=maxep)nsnew++;
        else flag=0;
    }
}

```

```

}
if (flag==1) return(SEXIT);
if (err_curr<=maxe) return(SEXIT);
return(CONTNE);
}
//*****AĞIRLIKLARIN AYARLANMASI*****
int rumelhart(int from_snum,int to_snum)
{
int i,j,k,m,n,p,offset,index;
float out;
char *err_file="hata.dat";
nsold=0;
cnt=0;
result=CONTNE;
if ((fp3=fopen(err_file,"w"))==NULL)
{
perror("Veri dosyası açılmıyor.");
exit(0);
}
do {
err_curr=0.0;
for (i=from_snum;i<to_snum;i++){
forward(i);
for(m=0;m<nunit[nhlayer+1];m++){
out=*(outptr[nhlayer+1]+m);
*(errptr[nhlayer+1]+m)=(target[i][m]-out)
*(1-out)*out;
}
for (m=nhlayer+1;m>=1;m--){
for (n=0; n<nunit[m-1]+1;n++){
*(errptr[m-1]+n)=0.0;
for (p=0; p<nunit[m];p++){
offset=(nunit[m-1]+1)*p+n;
*(delw[m-1]+offset)=eta*(*(errptr[m]+p))
*(*(outptr[m-1]+n))
+alpha*(*(delw[m-1]+offset));
*(errptr[m-1]+n)+=*(errptr[m]+p)
*(*(wtptr[m-1]+offset));
}
*(errptr[m-1]+n)=*(errptr[m-1]+n)*
(1-*(outptr[m-1]+n))*(*(outptr[m-1]+n));
}
}
for (m=1;m<nhlayer+2;m++){
for(n=0;n<nunit[m];n++){
for(p=0;p<nunit[m-1]+1;p++){
offset=(nunit[m-1]+1)*n+p;
*(wtptr[m-1]+offset)+=(delw[m-1]+offset);
}
}
}
}
ep[i]=0.0;
for(m=0;m<nunit[nhlayer+1];m++){
ep[i]+=fabs((target[i][m]-
*(outptr[nhlayer+1]+m)));
}
err_curr+=ep[i]*ep[i];
}
err_curr=0.5*err_curr/ninput;
if(kbhit())git();
fprintf(fp3,"%ld%f\n",cnt,err_curr);
cnt++;
result=introspective(from_snum,to_snum);
}
while (result==CONTNE);
for (i=from_snum; i<to_snum;i++) forward(i);
for(i=0; i<nhlayer+1;i++){
index=0;
for (j=0;j<nunit[i+1];j++){
printf("\n\n Weights between unit %d of layer %d" ,j,i+1);
printf("and units of layer %d\n",i);
for (k=0; k<nunit[i];k++)
printf ("%f",*(wtptr[i]+index++));
}
}

```

```

        printf("\n Threshold of unit %d of layer%dis%f",
            j,i+1,*(wtptr[i]+index++));
    }
}
for (i=0; i<ninput; i++)
    for (j=0;j<noutatr;j++)
        printf("\n\n sample %d output %d=%f target %d=%f",
            i,j,output[i][j],j,target[i][j]);getch();
        printf("\n\n Toplam iterasyon sayısı %ld",cnt);
        printf("\n Normalize edilmiş sistem hatası %f\n\n",err_curr);
        return(result);
    }
}
//*****Eğitim verilerinin YSA'ya tanıtılması*****
user_session()
{
    int i,j,showdata;
    char fnam[20],task_name[20],dtype[20];
    FILE *fp;
    printf("\n Eğitime safhasının başı ");
    printf("\n\t Veri dosyasının adını giriniz (taskname) : ");
    //scanf("%s",task_name);
    strcpy(task_name,"sin");
    printf("\n Kaç tane giriş düğümü (veri sayısı):");
    //scanf("%d",&ninatr);
    ninatr=2;
    printf("\nKaç tane çıkış düğümü?: ");
    //scanf("%d",&noutatr);
    noutatr=2;
    printf("\n Toplam kaç tane grup?: ");
    scanf("%d",&ninput);
    strcpy(fnam,task_name);
    strcat(fnam,".dat");
    printf("\n Giriş dosyasının adı %s",fnam);
    if ((fp=fopen(fnam,"r"))==NULL)
    {
        printf("\n %s dosyası yok", fnam);
        exit(0);
    }
    //for (m=0;m<12;m++) fscanf(fp, "%f",&se5);
    printf("\n Verilere bakmak istermisiniz.?"");
    printf("\n Answer yes or no :");
    scanf("%s",dtype);
    showdata=((dtype[0]=='y')||(dtype[0]=='Y'));
    for (i=0;i<ninput;i++){
        for (j=0;j<ninatr;j++){
            fscanf(fp, "%f",&se5);
            input[i][j]=(se5)/1.1;
            if(showdata)printf("%f ",input[i][j]);
        }
    }
    fclose(fp);
    printf("\n\t Çıkış dosyası adı giriniz : ");
    //scanf("%s",task_name);
    strcpy(task_name,"sin1");;
    strcpy(fnam,task_name);
    strcat(fnam,".dat");
    if ((fp=fopen(fnam,"r"))==NULL)
    {
        printf("\n %s dosyası yok", fnam);
        exit(0);
    }
    for (i=0;i<ninput;i++){
        for (j=0;j<noutatr;j++){
            fscanf(fp, "%f",&se6);
            target[i][j]=se6;
            if (showdata)printf(" %f",target[i][j]);
        }
    }
    if ((i=fclose(fp))!=0)
    {
        printf("\n %d dosya kapatılamıyor.",i);
        exit(0);
    }
    return(0);
}

```

```

    }
    /*******
learning()
    {
        int result;
        user_session();
        set_up();
        init();
        do {
            initwt();
            result=rumelhart(0,ninput);
        } while (result==RESTRT);
        dwrite(task_name);
        wtwrite(task_name);
        return(0);
    }
    /*******
main()
{ learning();
  printf("\nEğitim işlemi bitti. ");
  return(0);
}

```