67709

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

MİKRODENETLİYİCİ KULLANILARAK PWM AC KIYICI İLE ASENKRON MOTOR KONTROLUNUNUN İNCELENMESİ VE GERÇEKLEŞTİRİLMESİ

Elektronik ve Hab. Müh. A. Faruk BAKAN

F.B.E. Elektrik Mühendisliği Anabilim Dalı

YÜKSEK LİSANS TEZİ

Doç.Dr. Hacı BODUR
30.06.1997

Prof. Remzi GÜLGÜN

Jüri Üyesi
Prof. Metin YÜCEL
30/06/1997

Tez Danışmanı : Doç.Dr.Hacı BODUR

İSTANBUL, 1997

İÇİNDEKİLER

TEŞEKKÜR

Tez çalışmam boyunca bana yol gösteren değerli hocam Doç. Dr. Hacı BODUR'a teşekkür ederim.

A.Faruk BAKAN

ÖZET

Şebeke gerilimi ile beslenerek, yüke verilen ac gerilimin efektif değerini değiştiren ac kıyıcılar, ısıtma, aydınlatma, ac motor hız ayarı gibi endüstriyel uygulamalarda, güç kontrolu amacıyla yaygın olarak kullanılmaktadır. Faz kontrol tekniği kullanılarak yapılan güç kontrolunda, faz kontrol açısına bağlı olarak, yük gerilimi harmoniklerinin arttığı, yük akımında kesintilerin oluştuğu ve ac şebeke güç faktörünün düştüğü bilinmektedir. AC kıyıcılarda, faz kontrol tekniğinde meydana gelen mahzurların düzeltilmesi amacıyla PWM teknikleri kullanılmaktadır.

Bu tezde, akım kontrollu asimetrik PWM AC kıyıcı ile asenkron motor hız kontrolu sistemi incelenmiş, kıyıcının matematiksel modeli çıkarılmış ve devre simülasyonu yapılmıştır. PIC16C71 mikrodenetleyicisi ile PID algoritması kullanılarak ve hız geribeslemesi yapılarak, sistemin bir uygulaması gerçekleştirilmiştir. Kullanılan PIC mikrodenetleyicisi ve PID algoritmasının tanıtımına da geniş yer verilmiştir.

Sıhhatli bir şekilde çalışan sistem üzerinde, ölçü aletleri ve osiloskop ile muhtelif deneysel sonuçlar alınmıştır. Gerçekleştirilen sistemde, motorun geniş bir hız aralığında kararlı bir şekilde çalıştığı ve yüklenmelere karşı iyi cevap verdiği gözlenmiş, güç faktörü değerlerinin oldukça iyi olduğu ve çıkış gerilimi harmoniklerinin makul bir seviyede kaldığı tespit edilmiştir.

Ayrıca, kontrolun mikrodenetleyici ile yapılmasının, kontrol istekleri ile motor ve yük değişikliklerine program değişikliği ile cevap verme imkanı getirdiği, sistemin daha esnek ve kararlı, daha ekonomik ve endüstriyel olmasını sağladığı anlaşılmıştır.

ABSTRACT

Ac choppers that change the rms value of the ac voltage feeding a load from ac mains are in widespread use for purposes of power control in industrial applications such as heating, lighting and ac motor speed control. In this form of power control it is known that depending of phase control angle, the load voltage harmonics increase, discontinuities occur in load current and the ac mains power factor reduces. PWM techniques has been used to eliminate the disadvantages of phase control technique.

In this thesis, induction motor speed control with current controlled PWM AC chopper has been investigated, mathematical model of PWM AC chopper has been developed and the circuit has been simulated. The system has been implemented by using PIC microcontroller, speed feedback signal is taken and PID algorithm is used. Microcontroler and PID algorithm used in the system has been widely introduced.

Various experimental results has been obtained via oscilloscope from the system which works regularly. It has been observed that motor has worked stable in a wide speed range, given good response to sudden load changes in the system. The power factor values are quite well and voltage harmonics are at an acceptable level.

Realizing the current controlled PWM ac chopper with a microcontroller provided implementing desired control functions, adapting for different motors and loads by changing the program so the system became flexible, stable, economic and industrially applicable.

1. MİKRODENETLEYİCİLER

1.1. GİRİŞ

Mikroişlemci kullanılarak gerçekleştirilen kontrol sistemlerinde mikroişlemci ile birlikte RAM, ROM, A/D dönüştürücü, sayıcı ve I/O entegrelerinin de kullanılması gerekmektedir. Mikrodenetleyicilerde bu tür çevre elemanlarının kullanmasına gerek olmamaktadır. Mikroişlemci ile yapılan kontrol daha az elemanla mikrodenetleyici ile yapılabilmektedir. İstenilen kontrol amacına uygun olarak bütün çevre birimlerini içeren bir mikrodenetleyici seçmek mümkündür.

Güç elektroniği sistemlerinin mikrodenetleyici ile kontrolu aşağıda özetlenen birçok avantaj sağlamaktadır.

- Kontrol sisteminin mikrodenetleyici ile tasarlanması, sistemin donanım maliyetini düşürmektedir. Bu avantaj bilgisayar teknolojisinin gelişmesiyle daha da önemli hale gelmektedir. Daha hızlı ve daha fonksiyonel VLSI mikrodenetleyiciler üretilmektedir. Özel amaçlı ve genel amaçlı olarak tasarlanmış VLSI denetleyici yongaları, özel uygulamalarda yüksek sayıdaki üretimler için maliyet açısından etkin olmaktadır. Denetleyicinin hacim ve ağırlığının az oluşu ve düşük güç sarfiyatı ilave avantajlarıdır.

- Kontrol sisteminin gerçeklenebilirliği mikrodenetleyici ile önemli ölçüde artmaktadır. Bir LSI veya VLSI yongasının ortalama olarak iki arıza arasındaki zamanı, birbirine bağlı birçok elemandan oluşan elektronik devreden oldukça büyüktür. Mikrodenetleyicilerin gerçeklenebilirliği, güç elektroniği sisteminin diğer elemanlarına göre oldukça yüksektir. Bu durum, donanımın uygun bir şekilde tasarlanması, tanımlanan sınır değerlerinin mesela sıcaklık değerinin aşılmaması halinde doğrudur. Gerekirse askeri ve endüstri amaçlı daha yüksek sıcaklıklarda çalışabilen elemanlar kullanılabilir.

- Dijital devrelerde analog devrelerde olduğu gibi kayma, parametre değişimi gibi durumlar ortaya çıkmaz. İç yapıdaki elemanlar % 100 doğrudur. Uygun ölçeklendirme yapıldığı taktirde taşma ile ilgili problemler önlenebilir.

- Güç elektroniği devrelerinde yüksek gerilim ve akım süreksizliklerinden dolayı ortaya çıkan EMI problemi, yüksek entegrasyonun sağladığı toplam ekranlama ile çözülmektedir. Dışarıdan gelebilecek EMI, giriş ve çıkış işaretlerinin ve güç kaynağının uygun bir şekilde filtrelenmesi ile ortadan kaldırılabilir.

- Yerel bir güç elektroniği sisteminin mikrodenetleyici ile kontrolu, endüstriyel kontrol ortamlarında yüksek seviyeli bir merkezi bilgisayar ile uyumluluğa izin vermektedir. İleri ve geri bilgi akışının mümkün olması, fabrikalarda bilgisayar otomasyonuna doğru mevcut eğilim ile daha önemli hale gelmektedir.

- Güç elektroniği sistemlerinin birçoğu için genel bir donanım ile birlikte uygulamaya yönelik bir yazılım modülü kullanılabilir. Yazılım kontrolü çok esnektir. Fiziksel sistemin değişmesiyle yazılım modülleri kolaylıkla değiştirilebilir, güncelleştirilebilir, modülün bir kısmı silinebilir veya modüle ekleme yapılabilir.

- Kompleks hesaplamaların ve karar verme işlemleri içeren karmaşık kontrol fonksiyonlarının mikrodenetleyiciye yaptırılabilmesi, güç elektroniği sisteminin performansını, iyileştirir. Geçmişte bu tür fonksiyonların donanım kontrolu ile gerçekleştirilmesi çoğu zaman mümkün olamıyordu.

Mikrodenetleyici ile kontrolun bazı kısıtlamaları da aşağıda verilmiştir.

- Dijital hesaplamalar % 100 doğru olmakla beraber, analog dünya ile A/D ve D/A dönüştürücüleri vasıtasıyla bağlantı kurulurken, kuantalama ve örnekleme işlemlerinden dolayı belirli bir hata oluşmaktadır. Bu hata, analog bir sinüs işareti

A/D dönüştürücü ile örneklendiğinde ve daha sonra D/A ile yeniden elde edildiğinde açıkça görülebilmektedir. Hatalar analog veri toplama sisteminin bit sayısını ve örnekleme aralığını arttırarak azaltılabilir.

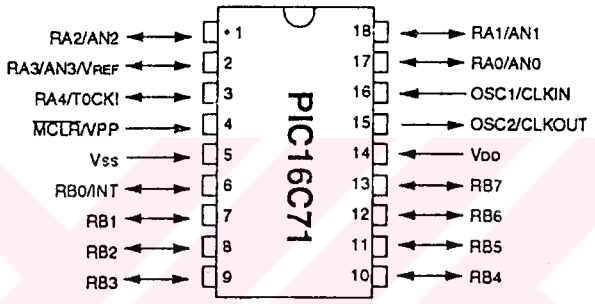
- Mikrodenetleyicinin cevap verme süresi analog bir sistemden daha yavaştır. Analog bir sistemde işaretler, paralel yollarda ihmal edilebilir bir gecikme ile eşzamanlı olarak işleme tabi tutulurlar. Diğer taraftan mikrodenetleyici işlemleri seri olarak yapar dolayısıyla analog sisteme göre daha fazla zaman harcar. Değişik fonksiyonların gerçekleştirilmesi için zaman paylaşımı söz konusu olduğunda işlem daha da yavaşlayacaktır. Hesaplamadaki yavaşlama yani hesaplama gecikmesi, geribesleme kontrol döngülerinde kararsızlık problemine ve hesaplanan işaretlerde bozulmaya yol açabilir. Bununla beraber hesaplama hızını arttırmak için hızlı işlemci kullanmak, programı assembly dilinde yazmak gibi değişik teknikler mevcuttur.

- Kontrolun mikrodenetleyici ile yapılması yazılım geliştirme çabasından dolayı pahalı, zaman harcayıcı olabilmektedir. Avantajlarından dolayı, özellikle yüksek sayıdaki üretimler için zaman ve maliyet bedeli önemsenmemektedir.

- Kontrolun yazılım ile gerçekleştirilmesi halinde, analog devrelerde osiloskop ve multimetre gibi cihazlarla ölçülebilen bazı değişkenlere ulaşmak daha zorlaşmakta, parametreler kolaylıkla izlenememektedir. Bunun için hata bulma cihazları ve ICE ortamı kullanmak gerekmektedir.

1.2. PIC16C71 MİKRODENETLEYİCİSİ

Bu kısımda uygulama devresinde kullanılan Microchip firmasının 16C71 mikrodenetleyicisi incelenmektedir. PIC16C71, PIC16C7X ailesinin en küçük üyesidir. PIC16C71 entegresinde 1024 byte ROM, 36 byte RAM, 13 adet I/O, 8-bit 4 kanallı A/D dönüştürücü ve bir adet zamanlayıcı / sayıcı bulunmaktadır. PIC16C71 entegresinin bacak bağlantıları Şekil 1.1'de verilmiştir.



Şekil 1.1. PIC16C71 mikrodenetleyicisi.

PIC16C67X serisinin program ve veri belleği kapasitesi Tablo 1.1'de, arabirim özellikleri Tablo 1.2'de, entegrenin bacak tanımlamaları Tablo 1.3'te verilmiştir.

Tablo 1.1. PIC16C7X'in program ve veri bellek kapasitesi.

Entegre	Program Belleği	Veri Belleği
PIC16C74/73	4K	192
PIC16C71	1K	36

Tablo 1.2. PIC16C7X'in arabirim özellikleri.

Entegre	I/O	A/D Kanal	TMR0	Akım (sink/source)
PIC16C74	33	8	Var	25mA / 25mA
PIC16C73	22	5	Var	25mA / 25mA
PIC16C71	13	4	Var	25mA / 25mA

Tablo 1.3. PIC16C71'in bacak tanımlamaları.

Bacak İsmi	DIP No	I/O/P Tipi	Buffer Tipi	Açıklama
OSC1 / CLKIN	16	I	ST/ MOS	Osilatör kristal girişi/ harici clock kaynağı girişi
OSC2 / CLKOUT	15	O	-	Osilatör kristal çıkışı
MCLR/Vpp	4	I/P	ST	Reset girişi / programlama gerilimi girişi
PORTA				
RA0/AN0	17	I/O	TTL	Analog giriş 0
RA1/AN1	18	I/O	TTL	Analog giriş 1
RA2/AN2	1	I/O	TTL	Analog giriş 2
RA3/AN3	2	I/O	TTL	Analog giriş 3/Vref
RA4/TOCKI	3	I/O	ST	TMR0'a saat girişi olarak tanımlanabilir. Çıkış olarak tanımlandığında open kollektör olmaktadır.
PORTB				
RB0/INT	6	I/O	TTL/ST	I/O veya INT. olarak kullanılabilir.
RB1	7	I/O	TTL	I/O olarak kullanılır.
RB2	8	I/O	TTL	I/O olarak kullanılır.
RB3	9	I/O	TTL	I/O olarak kullanılır.
RB4	10	I/O	TTL	Değişiklik olduğunda kesme oluşur.
RB5	11	I/O	TTL	Değişiklik olduğunda kesme oluşur.
RB6	12	I/O	TTL/ST	Değişiklik olduğunda kesme oluşur. Seri programlamada saat.
RB7	13	I/O	TTL/ST	Değişiklik olduğunda kesme oluşur. Seri programlamada veri.
VSS	5	P	-	Lojik ve I/O bacakları için toprak referansı
VDD	14	P	-	Lojik ve I/O bacakları için pozitif kaynak

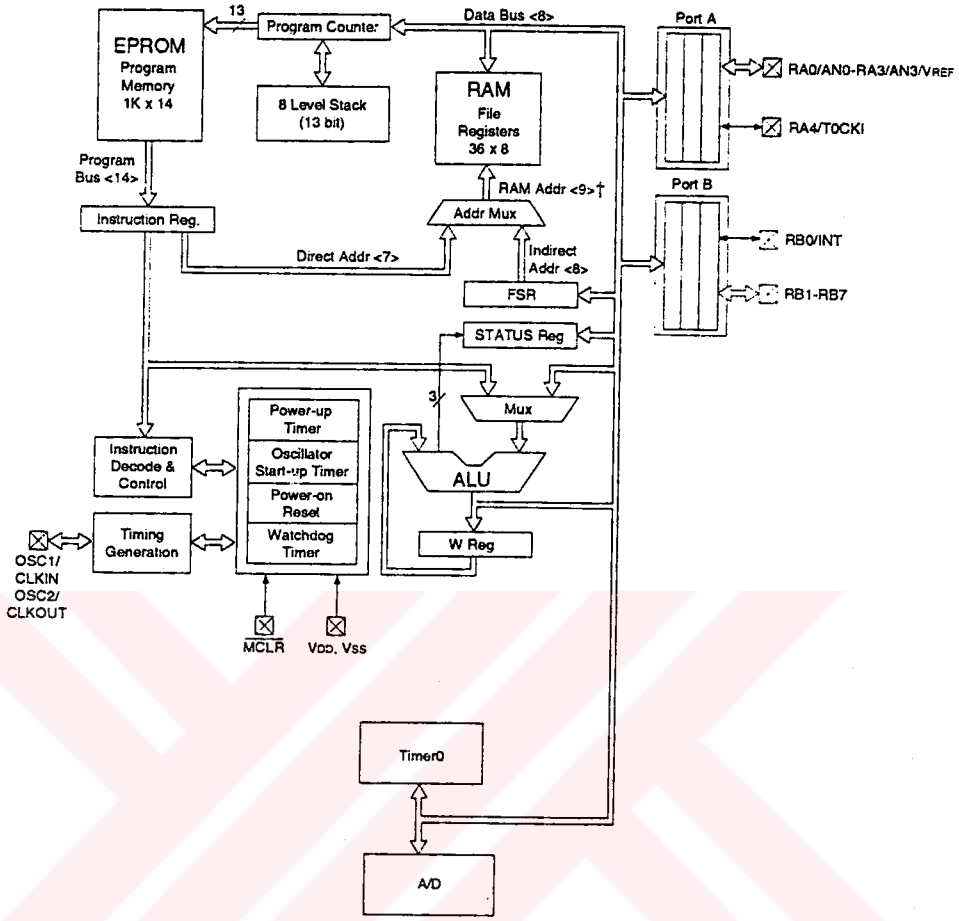
Kısaltmalar :
I = giriş
O = çıkış
P = kaynak
- = kullanılmıyor
TTL= TTL girişi
ST = Schmit tetikleme girişi

1.2.1. MİMARİ YAPISI

16CXX ailesi RISC işlemcileri grubuna girmekte ve Harvard mimarisini kullanmaktadır. Harvard mimarisinde program ve veri farklı belleklerde bulunmakta, program ve veriye farklı yollardan erişim sağlanmaktadır. Program ve veri belleklerinin ayrılması, komutların 8-bit genişliğindeki kelime uzunluğundan daha farklı büyüklükte tanımlanmasına imkan vermiştir. Komut işlem kodlarının 14 bit olarak tanımlanması bütün komutların 14-bitlik bir kelime olmasını sağlamıştır. 14-bitlik program belleğine ulaşan yoldan, 14-bitlik komut bir çevrimde getirilir. İki kademeli iş yapısında komut alma ve yürütme işlemleri aynı anda gerçekleştirilmektedir. Program dallanma komutları hariç bütün komutlar tek bir çevrimde gerçekleştirilmektedir. Bir komut çevrimi saat peryodunun dörtte biridir. 20 MHz'lik bir kristal bağlandığında komut çevrimi 200 ns olmaktadır.

PIC16C71 1K x 14 bit program belleğini adresleyebilmektedir. Bütün program belleği entegrenin içindedir. PIC16C71 doğrudan veya dolaylı olarak saklayıcı dosyasına veya veri belleğine erişebilmektedir. Program sayıcı da dahil olmak üzere bütün özel amaçlı saklayıcılar veri belleğinde bulunmaktadır.

PIC16C71, 8-bit ALU ve w saklayıcısı içermektedir. W işlemlerin yapıldığı saklayıcıdır. ALU genel amaçlı bir aritmetik birimdir. ALU herhangi bir saklayıcı ile w saklayıcısı arasındaki veriler üzerinde aritmetik ve mantıksal fonksiyonlar gerçekleştirir. ALU toplama, çıkarma, öteleme ve mantıksal işlemler yapabilmektedir. Aksi belirtilmediği taktirde aritmetik işlemler ikinin bütünleyeni olarak yapılmaktadır. İki operand'lı komutlarda bir operand w saklayıcısıdır. Diğer operand bir saklayıcı veya sabittir. W saklayıcısı ALU işlemlerinde kullanılan 8 bitlik adreslenemeyen bir saklayıcıdır. Yürütülen komuta bağlı olarak ALU, STATUS saklayıcısında bulunan Carry, Digit Carry ve Zero bitlerini değiştirebilir. PIC16C71'in sadeleştirilmiş blok diyagramı Şek.1.2'de görülmektedir.

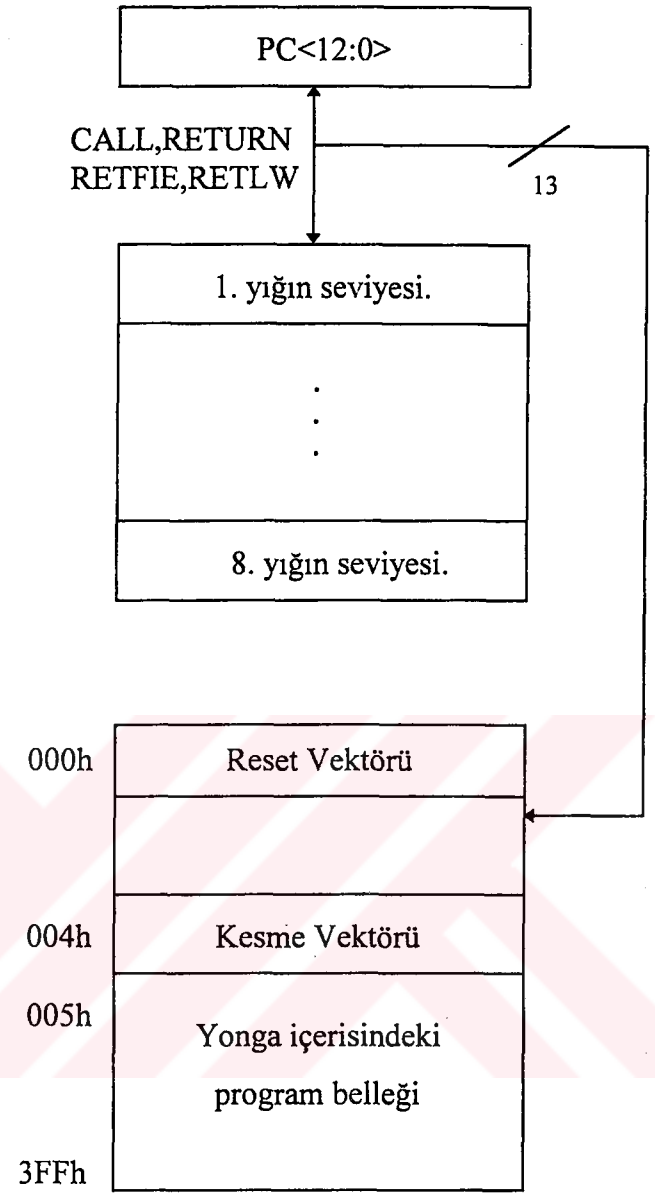


Şekil 1.2. PIC16C71'in blok diyagramı.

1.2.2. BELLEK ORGANİZASYONU

1.2.2.1. PROGRAM BELLEĞİ

PIC16C7X ailesinin 13-bitlik bir program sayıcısı vardır. Bu sayıcı PIC16C71'de 0000h-03FFh adresleri arasındaki 1K x 14 bitlik alanı adresleyebilmektedir. Reset vektörü 0000h, kesme vektörü 0004h adresindedir. Şek.1.3'te Program Belleği ve Yığın gösterilmektedir.



Şekil 1.3. Program belleği ve yığın yapısı.

1.2.2.2. VERİ BELLEĞİ

Veri belleği genel amaçlı saklayıcıları ve özel fonksiyon saklayıcılarını içeren iki bölüme ayrılmıştır. Bölüm 0 STATUS saklayıcındaki RP0 biti sıfır yapıldığında , bölüm 1 STATUS saklayıcındaki RP0 biti bir yapıldığında seçilmektedir.

1.2.2.2.1. GENEL AMAÇLI SAKLAYICI DOSYASI

Saklayıcı dosyasına doğrudan veya FSR saklayıcısı kullanılarak dolaylı olarak erişilebilir. PIC16C71'de 8Ch-AFh bölgesi fiziksel olarak mevcut değildir. Bu bölge bölüm 0'daki 0Ch-2Fh bölgesine yönlendirilmiştir. Tablo 1.4'te saklayıcı dosyası haritası verilmektedir.

Tablo 1.4. Saklayıcı dosyası haritası.

Adres	Bölüm 0	Bölüm 1	Adres
00	INDF	INDF	80
01	TMR0	OPTION	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07			87
08	ADCON0	ADCON1	88
09	ADRES	ADRES	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C	Genel Amaçlı Saklayıcılar	1. bölüme yönlendirilir.	8C
2F			AF

1.2.2.2.2. ÖZEL FONKSİYON SAKLAYICILARI

Özel fonksiyon saklayıcıları CPU ve arabirim fonksiyonları tarafından entegreden istenilen çalışmanın sağlanması için kullanılırlar. Bu saklayıcılar static RAM'dır. Özel saklayıcılar iki gruba ayrılabilirler. CPU ile bağlantılı saklayıcılar bu bölümde tanıtılmaktadır. Arabirim saklayıcıları ileride anlatılacaktır. Kısaltmalar w = yazılabilir, u = gerçekleştirilmemiş, r = okunabilir olarak gösterilmiştir.

STATUS SAKLAYICISI

03h veya 83h adresindedir. STATUS saklayıcısı ALU'nun aritmetik durumunu, işlemcinin RESET durumunu gösteren ve veri belleğinde bölüm seçimini sağlayan bitleri içermektedir.

R/W	R/W	R/W	R	R	R/W	R/W	R/W
IRP	RP1	RP0	NOT_TO	NOT_PD	Z	DC	C
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

C : Elde/ödünç bitidir. ADDWF ve ADDLW komutları için kullanılır. Sonucun en anlamlı bitinden elde oluştuğunda 1, oluşmadığında 0 olur.

DC : Digit elde/ödünç bitidir. ADDWF ve ADDLW komutları için kullanılır. Sonucun 4. bitinden bir carry oluştuğunda 1, oluşmadığında 0 olur.

Z : Sıfır bitidir. Aritmetik veya mantıksal işlemlerde sonucun sıfır olmasına göre değişir. İşlem sonucu sıfır ise 1, değilse 0 olur.

NOT_PD: MCLR girişine uygulanan gerilimin belli bir değerin altına düşmesi ile etkilenir. Gerilim düştükten sonra veya CLRWDT komutu çalıştırılırsa 1 olur. SLEEP komutunun çalıştırılması ile 0 olur.

NOT_TO: Zamanlama hatası bitidir. Besleme gerilimi gerekli seviyeye yükseldiğinde ve CLRWDT ile SLEEP komutlarının çalıştırılmasıyla 1 olur. WDT zamanlayıcısında taşma meydana geldiğinde 0 olur.

RP<0> : Doğrudan adreslemede saklayıcı bölümü seçmek için kullanılır. Bölüm 1'i seçmek için 1, bölüm 0'ı seçmek için 0 yapılır.

RP<1> : 0 olarak programlanmalıdır. PIC16C7X'te sadece RP<0> kullanılmaktadır.

IRP : Dolaylı adreslemede saklayıcı bölümü seçme biti. IRP biti PIC16C7X'te kullanılmamaktadır. Gelecekteki ürünlerle uyum açısından 0 olarak programlanmalıdır.

OPTION SAKLAYICISI

81h adresindedir. OPTION saklayıcısı TMR0/WDT katsayısının değerini, dış INT kesmesini, TMR0'ı ve PORTB'nin yüksek seviyeye çekilmesini düzenleyen kontrol bitlerini içermektedir. Okunabilen ve yazılabilen bir saklayıcıdır.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NOT_RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

PS2,PS1,PS0 : TMR0 veya WDT'nin artış hızını belirleyen katsayıyı tanımlar.

PS2	PS1	PS0	TMR0 HIZI	WDT HIZI
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1;256	1:128

PSA : Katsayı atamak için kullanılır. <PS2:PS0> ile seçilen katsayı, PSA 1 yapılırsa WDT ' ye, 0 yapılırsa TMR0'a atanır.

TOSE : TMR0'ın RA4/TOCKI bacağından uygulanan saat kaynağının hangi kenarında tetikleneceğini belirler. 1 yapılırsa yüksek seviyeden düşük seviyeye geçişinde, 0 yapılırsa düşük seviyeden yüksek seviyeye geçişinde tetiklenir.

TOCS : TMR0 'ın saat kaynağını belirler. 1 yapılırsa TOCKI bacağına uygulanan kare dalga ile artar. 0 yapılırsa saat frekansının dörtte bir hızı ile orantılı olarak artar.

INTEDG: Kesme kenarının seçimini belirler. 1 yapılırsa kesme RB0/INT bacağının yükselen kenarında, 0 yapılırsa RB0/INT bacağının düşen kenarında oluşur.

NOT_RBPU: PortB'nin yüksek seviyeye seçilmesini (pull-up) sağlar. 1 yapılırsa PORTB'den pull-up kaldırılır. 0 yapılırsa PORTB bir direnç ile +Vcc'ye bağlanmış gibi davranır.

INTCON SAKLAYICISI

0Bh veya 8Bh adresindedir. Kesme denetimi ile ilgili önemli bir saklayıcıdır. PortB değişme kesmesi, INT kesmesi, TMR0 kesmesi ve ADC kesmesine ait düzenlemeler bu saklayıcı kullanılarak gerçekleştirilir.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
GIE	ADIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

RBIF : PORTB'de değişiklik olup olmadığını gösteren bir bayraktır. RB<7:4>'den herhangi biri değiştiğinde 1 olur. Yazılım ile sıfırlanmalıdır. RB<7:4>'den hiçbiri değişmediyse 0'dır.

INTF : Dış INT kesmesi ile ilgili bayraktır. Dış INT kesmesi oluştuğunda 1 olur. Dış INT kesmesi oluşmadığında 0'dır.

T0IF : TMR0 taşma kesmesi oluştuğunu gösterir. TMR0 taştığında 1 olur, yazılım ile sıfırlanması gerekir. TMR0'da taşma oluşmadıysa 0'dır.

RBIE : RB<7:4> bitlerinin değişmesiyle kesme olup olmayacağına karar verir. 1 yapılırsa RBIF'nin kesme olduğunda değişmesine imkan verir. 0 yapılırsa kesme olduğunda RBIF değişmez.

INTE : INT kesmesini mümkün kılma bitidir. 1 yapılırsa INT kesmesini mümkün kılar. 0 yapılırsa INT kesmesini iptal eder.

T0IE : TMR0'da taşma oluşması halinde kesme olup olmayacağına karar verir. 1 yapılırsa TMR0'da taşma olduğunda kesmeyi mümkün kılar. 0 yapılırsa TMR0'da taşma olduğunda kesme oluşmaz. T0IF bayrağı 0 olur.

ADIE : A/D dönüştürmesinin tamamlandığını gösteren kesme bitidir. 1 yapılması A/D kesmesini mümkün kılar. 0 yapılması A/D kesmesini iptal eder.

GIE : Genel kesme mümkün kılma bitidir. 1 yapılırsa iptal edilmemiş bütün kesmeleri mümkün kılar. 0 yapılırsa bütün kesmeleri iptal eder.

PC VE PCLATH

PC 13 bit genişliğindedir. Düşük anlamlı kısmı PCL, yüksek anlamlı kısmı PCH'tır. PCL okunabilen ve yazılabilen bir saklayıcıdır. PCH okunabilir ve yazılabilir bir saklayıcı değildir. PCLATH, PC<13:8> değerini tutan bir saklayıcıdır. PC'ye bir CALL, GOTO komutu ile veya PCL'ye yazma işlemi ile yeni bir değer atandığında PC'nin yüksek kısmı PCLATH'dan elde edilmektedir.

1.2.3. YIĞIN

PIC16CXX 8 adet 13 bit genişliğinde yığın seviyesine sahiptir. Yığın alanı program belleğinin veya veri belleğinin bir bölgesi değildir. Yığın işaretleyicisi okunabilir veya yazılabilir değildir. CALL komutu çalıştırıldığında veya bir kesme ortaya çıktığında PC yığına atılır. RETURN, RETLW veya RETFIE gibi bir komut çalıştırıldığında yığın'a en son atılan bilgi geri çekilir. Yığın dairesel bir buffer gibi çalışır. Yığın'a 8 defa ardarda veri atıldığında, 9. verinin atılması yazma işleminin birinci verinin üzerine yapılmasına yol açacaktır.

1.2.4. DOLAYLI ADRESLEME

INDF saklayıcısını kullanan komutlar FSR tarafından işaret edilen veriye ulaşılar. INDF'in kendisini okumak dolaylı olarak 00h üretir. INDF üzerine yazmak dolaylı olarak nop komutuna karşılık düşer. INDF'in kendisi fiziksel bir saklayıcı değildir.

Dolaylı adresleme metodu ile 20h-2Fh adreslerini temizlemek için FSR ve INDF'in nasıl kullanılacağı aşağıdaki örnek programda gösterilmiştir.

```
        MOVLW   0x20
        MOVF    FSR

NEXT    CLRF    INDF
        INC     FSR
        BTFSS   FSR, 4
        GOTO    NEXT

CONTINUE:
```

1.2.5. I/O PORTLARI

PIC16C71 PORTA ve PORTB olmak üzere iki adet porta sahiptir. Portların özellikleri aşağıda verilmektedir.

1.2.5.1. A PORTU

PORTA 5-bitlik bir tutucudur. RA4 Schmitt tetikleyici girişine ve açık kollektör çıkışına sahiptir. Diğer bütün RA bacakları TTL giriş seviyelerine ve CMOS çıkış seviyelerine sahiptir. Bütün bacaklar veri yönlendirme bitlerine sahiptir. Veri yönlendirme işlemi TRIS saklayıcıları ile yapılmaktadır. Veri yönlendirme bitleri ile bu bacaklar giriş veya çıkış olarak düzenlenebilmektedir.

TRISA saklayıcısındaki "1" buna karşı düşen çıkış sürücüsünü yüksek empedans moduna çeker. TRISA saklayıcısındaki "0" çıkıştaki tutucunun içeriğini seçilen bacaklara aktarır.

RA4 bacağı TMR0 saat girişi ile çoğullanmıştır. Diğer PortA bacakları analog girişler ve Vref girişi ile çoğullanmıştır. Bu bacakların nasıl çalışacakları A/D kontrol saklayıcısı ADCON1 kullanılarak seçilmektedir.

Analog giriş olarak kullanılan bacaklar giriş olarak koşullandırılmalıdırlar.

PortA'nın nasıl koşullandırılacağı ile ilgili bir program örneği aşağıda verilmiştir.

```
CLRF    PORTA          ; PortA'yı çıkış tutucusunu kurarak koşullandır.
BSF     STATUS,RP0     ; Bölüm 1'i seç
MOVLW   0xCF           ; Veri yönünü belirlemek için kullanılan değer.
MOVWF   TRISA          ; RA<3:0> giriş olarak tanımlanıyor.
                       ; RA<4> çıkış olarak tanımlanıyor.
```

1.2.5.2. B PORTU

PORTB 8-bit genişlikte iki yönlü porttur. 06h adresindedir. Buna karşılık gelen veri yönlendirme saklayıcısı TRISB 86h adresindedir. Aşağıda PORTB'nin giriş ve çıkış olarak tanımlanmasına ilişkin bir örnek program verilmiştir.

```
CLRF    PORTB       ; PortB'yı çıkış tutucusunu kurarak koşullandır.
BSF     STATUS,RP0  ; Bölüm 1'i seç
MOVLW   0xCF        ; Veri yönünü belirlemek için kullanılan değer
MOVWF   TRISB       ; RB<3:0> giriş olarak tanımlanıyor.
                    ; RB<5:4> çıkış olarak tanımlanıyor.
                    ; RB<7:6> giriş olarak tanımlanıyor.
```

PORTB bacaklarından herbiri yüksek seviyeye çekilebilir. NOT_RBPU kontrol biti sıfır yapıldığında bütün bacaklar yüksek seviyeye çekilir(pull-up işlemi). Port bacağı çıkış olarak tanımlandığında otomatik olarak pull-up kaldırılır. POR (Power On Reset) durumunda pull-up iptal edilir.

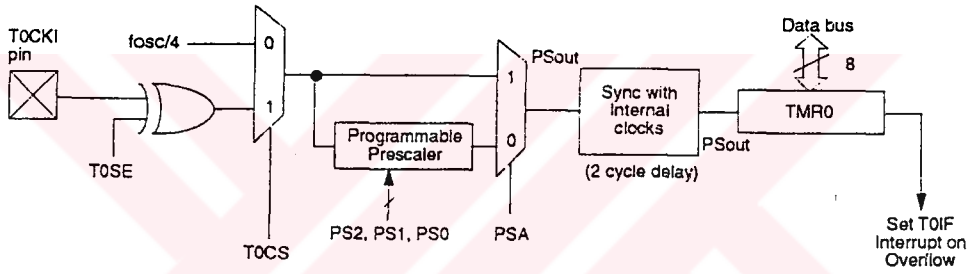
PORTB bacaklarından 4'ü RB<7:4> giriş olarak tanımlanırlarsa, değişiklik olduğunda kesme oluşturma özelliğine sahiptirler. RB7-RB4 bacaklarından alınan girişler, PORTB'nin en son okunmasında tutucuda kalan değerler ile karşılaştırılır. Bu çıkışlar OR'lanarak RBIF kesmesi üretilir (INTCON<0>). Bu kesme entegreyi SLEEP modundan uyandırabilmektedir. Kesme hizmet altprogramında kullanıcı kesmeyi iki yoldan biri ile temizleyebilir.

- RBIE bitini sıfır yaparak kesmeyi iptal ederek.
- PORTB'yi tekrar okuyarak. Bu durumda tutucu ve RB girişleri aynı olur ve uyumsuzluk durumu ortadan kalkar. Daha sonra RBIF bayrağı temizlenir.

1.2.6. ZAMANLAYICI MODÜLÜ

PIC16C71 bir adet zamanlayıcı / sayıcı' ya sahiptir. TMR0 modülü aşağıdaki özelliklere sahiptir. TMR0 modülünün blok diyagramı Şekil 1.4 ' te gösterilmiştir.

- 8 bit zamanlayıcı/sayıcı.
- Okunabilir ve yazılabilir.
- Yazılımla programlanabilen katsayı.
- Dahili veya harici saat seçimi.
- Harici saatte kenar seçimi.



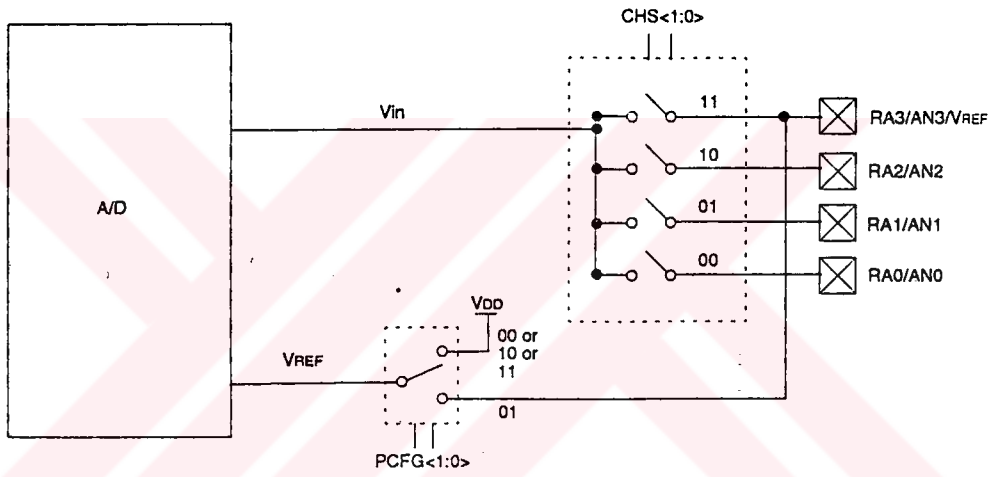
Şekil 1.4. TMR0 modülünün blok diyagramı.

TMR0 ve Watchdog zamanlayıcısı tarafından ortak kullanılan bir adet katsayı mevcuttur. Dolayısıyla TMR0'a bir katsayı atanmışsa, Watchdog Timer'a katsayı atanmamış demektir. Aşağıdaki programda katsayı atamasının nasıl yapıldığı gösterilmektedir.

```
BCF     STATUS,RP0      ; Bölüm 0.
CLRF    TMR0            ; TMR0 ve katsayısını sıfırla.
BSF     STATUS,RP0      ; Bölüm 1.
CLRWDT                  ; WDT'yi sıfırla.
MOVLW   B'xxxx1xxx'     ; Yeni katsayı değerini seç.
MOVWF   OPTION          ; Değeri güncelle.
```

1.2.7. ANALOG DİJİTAL DÖNÜŞTÜRÜCÜ MODÜLÜ

A/D dönüştürücü analog bir giriş işaretini 8 bitlik dijital bir sayıya dönüştürmektedir. PIC16C71'de 4 adet analog giriş mevcuttur. Bu 4 giriş, tek bir örnekle ve tut devresi üzerine çoğullanmıştır. Örnekle ve tut devresinin çıkışı A/D dönüştürücünün girişi olup, ardışıl yaklaşıklık metodu ile sonucu üretmektedir. Analog referans gerilimi, yazılım ile entegrenin pozitif besleme gerilimi Vdd veya RA3 / AN3 / Vref bacağındaki gerilim olarak seçilebilir. A/D dönüştürücü entegre SLEEP modunda iken çalışabilme özelliğine sahiptir. Şekil 1.5'te A/D dönüştürücü'nün blok diyagramı verilmektedir.



Şekil 1.5. PIC16C71'de A/D dönüştürücü blok diyagramı.

A/D modülünün, ADRES, ADCON0 ve ADCON1 olmak üzere üç adet saklayıcısı bulunmaktadır. ADRES A/D sonuç saklayıcısıdır. ADCON0 ve ADCON1 A/D ile ilgili kontrol saklayıcılarıdır.

ADCON0 SAKLAYICISI

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADCS1	ADCS0	Reserved	CHS1	CHS0	GO/NOT_DONE	ADIF	ADON
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

ADON : A/D modülü açma kapama biti. 1 yapılırsa A/D dönüştürücü modülü çalışır. 0 ise A/D dönüştürücü modülü kapalıdır ve güç harcamaz.

ADIF : A/D dönüşümü tamamlandı kesme bayrağı biti. Dönüşüm tamamlandığında donanım tarafından 1 yapılır. Yazılım ile sıfırlanmalıdır.

GO/NOT_DONE : A/D dönüşümü durum bitidir. Eğer ADON=1 ise bu bitin 1 olması A/D dönüşümü devam ediyor demektir. A/D dönüşümünü başlatmak için de bu biti 1 yapmak gerekir. 0 olduğunda A/D devam etmiyor/tamamlandı demektir. Bu bit A/D dönüşümü tamamlandığında otomatik olarak sıfırlanır.

CHS<1:0> Analog kanal seçme bitleridir.

00 : Kanal 0 (RA0 / AN0).

01 : Kanal 1 (RA1 / AN1).

10 : Kanal 2 (RA2 / AN2).

11 : Kanal 3 (RA3 / AN3 / Vref).

ADCS<1:0> A/D dönüşümü için kullanılan saatin seçim bitleridir.

00 : Fosc/2.

01 : Fosc/8.

10 : Fosc/32.

11 : F_{RC}. Saat RC osilatörden elde edilir.

ADCON1 SAKLAYICISI

u	u	u	u	u	u	R/W	R/W
-	-	-	-	-	-	PCFG1	PCFG2
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

u = kullanılmayan bitler.

PCFG<1:0> : A/D port konfigürasyon bitleri. Bu bitler port bacaklarının aşağıdaki şekillerde düzenlemesini sağlar.

PCFG<1:0>	RA0	RA1	RA2	RA3	Ref
00	A	A	A	A	Vdd
01	A	A	A	Vref	RA3
10	A	A	D	D	Vdd
11	D	D	D	D	Vdd

A : Analog giriş.

D : TRIS bitine bağlı olarak dijital giriş veya çıkış olabilir.

ADRES SAKLAYICISI

ADRES saklayıcısı A/D dönüşümünün sonucunu saklar. A/D dönüşümü tamamlandığında sonuç ADRES saklayıcısına yazılır, ADCON<0>=G0/NOT_DONE biti sıfırlanır ve A/D kesme bayrağı ADIF 1 yapılır. A/D modülü düzenlendikten sonra, dönüşümü başlatmadan istenilen kanal seçilmelidir. Analog giriş kanallarına karşılık düşen TRIS bitleri giriş olarak tanımlanmalıdır. A/D örnekleme zamanı T_{AD} olarak tanımlanır. A/D dönüşümü yapılırken aşağıdaki basamaklar izlenmelidir.

1. A/D modülünü düzenle.
   - Analog bacakları / gerilim referansını / dijital I/O'yu düzenle (ADCON1).
   - A/D giriş kanalını seç (ADCON0).
   - A/D dönüştürme saatini seç (ADCON0).
   - A/D modülünü çalıştır (ADCON0).

2. A/D kesmesini düzenle.
   - ADIF bitini sıfırla.
   - ADIE bitini "1" yap.
   - GIE bitini "1" yap.

3. Gerekli örnekleme zamanı kadar bekle.

4. Dönüşümü başlat.
   - GO/NOT_DONE bitini "1" yap.

5. A/D dönüşümünün tamamlanması için bekle. Bekleme iki şekilde yapılabilir.
   - GO/NOT_DONE bitinin "0" olmasını bekleyerek.
   - A/D kesmesinin gelmesini bekleyerek.

6. A/D sonuç saklayıcısı ADRES'i oku. Gerekli ise ADIF'yi sıfırla.

7. Bir sonraki dönüşüm için duruma göre basamak 1 veya 2'ye git. Bir sonraki örneklemenin başlayabilmesi için en az 2 T_{AD} kadar beklemek gereklidir.

TAD süresi için dört farklı seçenek mevcuttur. 2 Tosc, 8 Tosc, 32 Tosc veya dahili RC osilatör. $(T_{AD})_{min}$ PIC16C71 için 2 µs'dir.

Aşağıda A/D dönüştürme program örneği verilmiştir. Örnekte RA bacakları analog giriş olarak, Vref entegrenin Vdd değeri olarak tanımlanmıştır.

```
BSF     STATUS,RP0      ; Sayfa 1'i seç.
CLRF    ADCON1          ; A/D girişlerini düzenle.
BCF     STATUS,RP0      ; Sayfa 0'ı seç.
MOVLW   0xC1            ; RC saati, A/D açık, Kanal 0 seçildi.
MOVWF   ADCON0          ; ADCON'a yükle.
BSF     INTCON,ADIE     ; A/D kesmesini mümkün kıl.
BSF     INTCON,GIE      ; Bütün kesmeleri mümkün kıl.
        .
        .
BSF     ADCON0,GO       ; A/D dönüşümünü başlat.
```

```
                        ; Dönüşüm tamamlandığında ADIF biti "1" yapılacak ve
                        ; GO/NOT_DONE biti sıfırlanacaktır.
MOVF    ADRES,W         ; ADRES saklayıcısındaki sonuç w'ye alınır.
```

A/D kullanılırken Vss < Vin < Vdd olmasına dikkat edilmelidir. Ayrıca analog kaynaklar için tavsiye edilen en büyük empedans 10 kohm'dur.

1.2.8. KONFİGÜRASYON KELİMESİ

CONFIG saklayıcısı 2007h adresinde bulunan bir saklayıcıdır. Osilatör, WDT, Power up timer ve program koruma ile ilgili düzenlemeler yapılmasına imkan tanımaktadır.

-	-	-	-	-	-	-	-	CP0	PWRTE	WDTE	FOSC1	FOSCO
bit13												bit0

FOSC<1:0> osilatör seçme bitleri.

11 : RC osilatör.

10 : HS osilatör.

01 : XT osilatör.

00 : LP osilatör.

WDTE : WDT'ı mümkün kılma biti. 1 yapılırsa WDT çalışır. 0 yapılırsa WDT çalıştırılmaz.

PWRTE: Power up zamanlayıcısını mümkün kılma biti. 1 yapılırsa power up zamanlayıcısı çalışır. 0 yapılırsa zamanlayıcı çalıştırılmaz.

CP0 : Kod koruma biti. Bu bit 1 yapılırsa program koruması yapılmaz. 0 yapılırsa tüm bellek kod koruma moduna alınır. Sadece 0-3F bölgesine yazılabilir.

1.2.9. KOMUT KÜMESİ

16C71 mikrodenetleyicisinin komut kümesi Tablo 1.5'te verilmektedir.

Tablo 1.5. 16C71 mikrodenetleyicisinin komut kümesi.

1	addlw	k	(w) + k→w
2	andlw	k	(w) AND (k)→w
3	addwf	f,d	(w) + (f)→d
4	andwf	f,d	(w)AND(f)→d
5	bcf		f,b 0→f(b)
6	bsf	f,b	1→f(b)
7	btfsc	f,b	f(b)=0 ise bir satır atla
8	btfss	f,b	f(b)=1 ise bir satır atla
9	call	k	k adresine dallan
10	clrf	f	00h→f
11	clrw		00h→w
12	clrwdt		00h→WDT
13	comf	f,d	(f')→d
14	decf	f,d	(f)-1→d
15	decfsz	f,d	(f)-1→d sonuç sıfır ise bir satır atla
16	goto	k	k adresine git
17	incf	f,d	(f) + 1→d
18	incfsz	f,d	(f) + 1→d sonuç sıfır ise bir satır atla
19	iorlw	k	(w) OR (k) →w
20	iorwf	f,d	(w) OR (f) →w
21	movlw	f,d	k→w
22	movf	f,d	(f)→d
23	movwf	f	(w)→f
24	nop		işlem yok
25	option		w→option
26	retfie		TOS→PC, GIE→1
27	retlw		k→w, TOS→PC
28	return		TOS→PC
29	rlf	f,d	f, C üzerinden sola ötelenir
30	rrf	f,d	f, C üzerinden sağa ötelenir
31	sleep		Düşük güç sarfiyatı moduna geçer
32	sublw	k	k-(w)→w
33	subwf	f,d	f-(w)→dest
34	swapf	f,d	f<0:3>→d<4:7>, f<4:7>→d<0:3>
35	xorlw	k	w (XOR) k → w
36	xorwf	f,d	(w) XOR f→dest

Kısaltmalar :
f : saklayıcı dosyası adresi(0x00-0x2F)
w: üzerinde işlem yapılan saklayıcı
b : saklayıcının b. biti
k : sabit veri ya da etiket
d : Sonucun yazılacağı bölge. d=0 ise w'ye yazılır. d=1 ise f'e yazılır
TOS : yığının en üst byte'ı

2. AKIM KONTROLLU PWM AC KIYICININ İNCELENMESİ

2.1. GİRİŞ

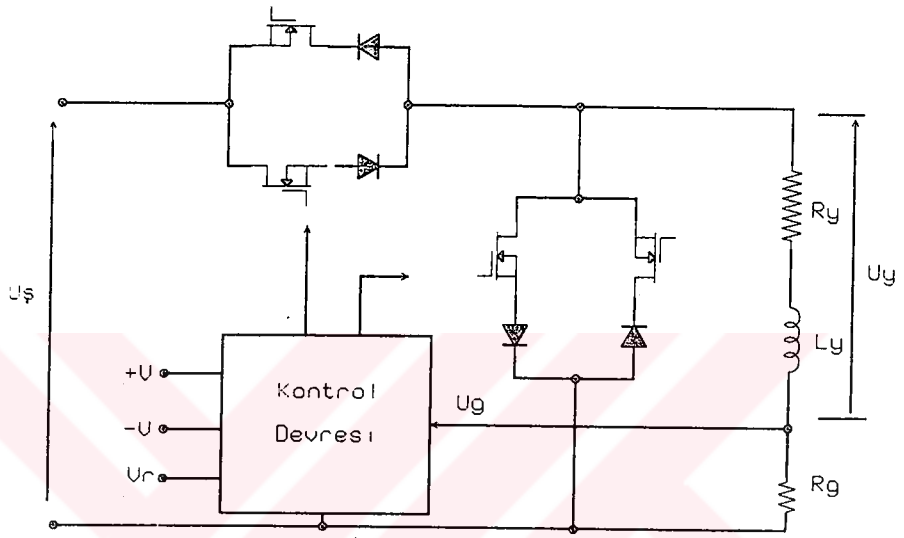
Sabit gerilimli bir ac kaynaktan yüke verilen ac gerilimin efektif değerini değiştiren ac kıyıcılar, ısıtma, aydınlatma, ac motor hız ayarı gibi endüstriyel uygulamalarda, güç kontrolu amacıyla yaygın olarak kullanılmaktadır. Uzun yıllardan beri, tristör ve triyaklarla gerçekleştirilen doğal komütasyonlu ac kıyıcı devreleri ile faz kontrol tekniği kullanılarak, ekonomik ve basit bir şekilde, çok büyük güçlere kadar ac güç kontrolu yapılmaktadır. Bu şekilde yapılan güç kontrolunda, faz kontrol açısına bağlı olarak, yük gerilimi harmoniklerinin arttığı, yük akımında kesintilerin oluştuğu ve ac şebeke güç faktörünün düştüğü bilinmektedir.

AC kıyıcılarda, faz kontrol tekniğinde meydana gelen mahzurların düzeltilmesi amacıyla, yarı iletken güç elemanlarıyla paralel olarak hızla gelişen PWM tekniklerinin kullanılması çalışmaları yoğunluk kazanmıştır. PWM metodları genel olarak simetrik ve asimetrik olarak iki gruba ayrılır. Simetrik PWM metodlarında, yük gerilim ve akımının sinüsoidala yaklaştırılabilmesine karşılık, ac şebeke güç faktörü düzeltilememektedir. Asimetrik PWM metodlarında ise, yük gerilimi şebeke gerilimine göre ileri alınmak suretiyle, yük geriliminde bir miktar harmoniğe müsaade edilerek, güç faktörünün yükseltilmesi imkanları sağlanmaktadır.

2.2. OMİK ENDÜKTİF YÜK İÇİN AKIM KONTROLLU TEK FAZLI PWM AC KIYICININ İNCELENMESİ

PWM kontrollu bütün ac kıyıcılarda, birisi yüke seri ve diğeri yüke paralel olmak üzere, ters paralel bağlı kontrollu iki elemandan oluşan iki grup bağlanır. Seri grup yüke verilen gücü kontrol eder, paralel grup ise yük akımının sürekliliğini sağlar veya yükte biriken enerjiyi söndürür. PWM kontrollu ac kıyıcının prensip şeması Şekil 2.1'de görülmektedir.

Omik-endüktif bir yükü besleyen devrede, akım örneği yüke seri bağlı küçük değerli bir direnç üzerinden alınmaktadır. Bu dirençten alınan akım örneği ile şebeke geriliminden sağlanan referans akımın karşılaştırılması suretiyle, mosfetler için gerekli olan sürme sinyalleri elde edilmektedir. Gerçek yük akımının, seçilen bir histerizis bantı içerisinde referans akımı takip etmesi sağlanmaktadır. Referans akımının değiştirilmesi ile yük akımı kontrol edilebilmektedir.



Şekil 2.1. PWM kontrollu ac kıyıcının prensip şeması.

2.3. PWM KONTROLLU AC KIYICININ MATEMATİKSEL MODELİ VE SİMÜLASYONU

Şekil 2.1'de prensip şeması verilen PWM ac kıyıcı devresinde anahtarlama yapılmadığı durumda omik-endüktif bir yük için çıkış gerilimi,

$$v_Y(\theta) = L_Y \frac{di_y(\theta)}{d(\theta)} + R_Y . i_Y(\theta) \qquad (2.1)$$

olarak elde edilmektedir. Bu diferansiyel denklemin nümerik olarak çözülebilmesi için,

$$i_Y(k+1) = \frac{D}{L} v_Y(k) + i_Y(k)\left[1 - \frac{D.R}{L}\right] \qquad (2.2)$$

ifadesi elde edilmiştir. (2.2) nolu eşitlikte D örneklenmiş işaretin iki örneği arasındaki uzaklık olarak tanımlanmıştır. i_R referans akımı, şebekeden alınan akım örneği olup sıfır ile yük akımının maksimum değeri arasında değiştirilmektedir.

$$i_R=I_R.\sin(\omega t), \qquad 0<I_R<I_{YM} \tag{2.3}$$

Burada, I_R referans akımının genliği, I_{YM} ise yük akımının maksimum değeridir.Akım kontrollu PWM devresinde histerezis bandı IH olarak tanımlanmıştır. Yük akımı pozitif alternansta referans akımın IH/2 kadar altına düşerse veya negatif alternansta referans akımın IH/2 kadar üstüne çıkarsa yüke şebeke gerilimi verilmektedir. Bu durumda yük akımı (2.2) nolu formülde verildiği gibidir. Eğer yük akımı pozitif alternansta referans akımın IH/2 kadar üstüne çıkarsa veya negatif alternansta referans akımın IH/2 kadar altına düşerse gerilim kesilmektedir. Bu durum için ise yük akımı ,

$$i_Y(k+1)=i_Y(k)\left[1-\frac{D.R}{L}\right] \tag{2.4}$$

şeklinde elde edilmiştir.

TGK şebeke tarafındaki toplam güç katsayısı, BK bozulma kaysayısı ve TBGK temel bileşen güç katsayısı olarak tanımlanmaktadır. Buna göre temel bileşen güç katsayısı,

$$TGK=P/S=(I_1 / I).\cos(\varphi_1) \tag{2.5}$$

bozulma katsayısı,

$$BK=\frac{1}{I}\left[\sum_{n=2,3,..}^{\infty} I_n^{\,2}\right]^{\frac{1}{2}}=\frac{I_h}{I}=\sqrt{1-(\frac{I_1}{I})^2} \tag{2.6}$$

temel bileşen güç katsayısı,

$$TBGK=\cos(\varphi_1) \qquad (2.7)$$

olarak elde edilir. (2.5), (2.6) ve (2.7) ifadelerinde I şebeke akımının efektif değeri, I_1 temel bileşen ve φ_1 temel bileşenin faz farkıdır.

Yukarıdaki matematiksel modele göre çalışan ve fourier analizi de yapan bir simülasyon programı geliştirilmiştir Bu program ile yük akımının, şebeke akımının ve yük geriliminin dalga şekilleri elde edilmekte, girişteki toplam güç katsayısı, temel bileşen güç katsayısı ve bozulma katsayısının ve giriş akımının harmonik içeriğinin grafikleri çizilmektedir. Simülasyon programı Tablo 2.1'de verilmiştir.

Tablo 2.1. Omik-endüktif yükte PWM ac kıyıcının incelenmesi için yazılan simülasyon programı.

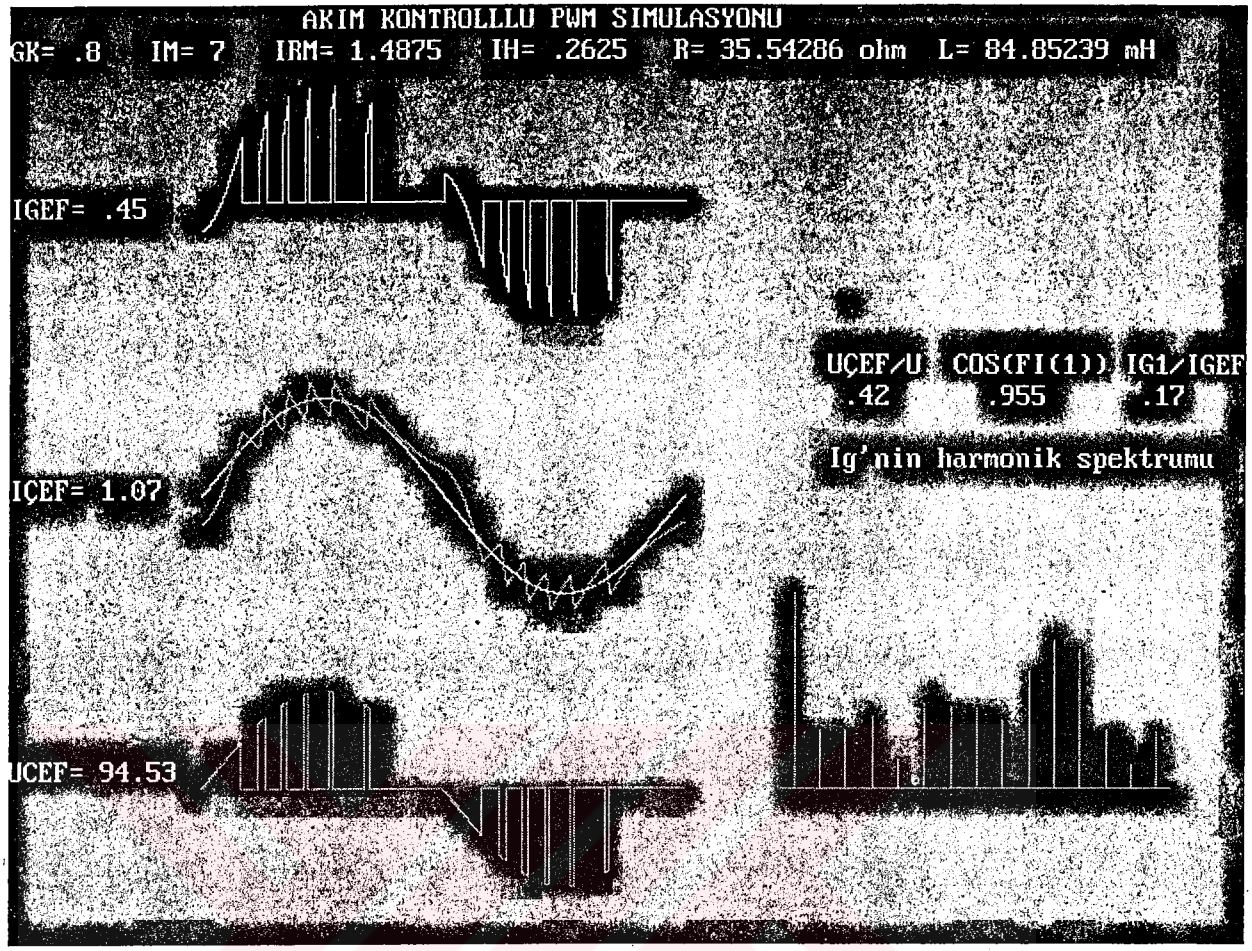
```
COPYRIGHT AHMET FARUK BAKAN 1997
SCREEN 12
CONST PI = 3.14159
CONST W = 2 * PI * 50
CONST PERYOD = 2 * PI / W
CONST GK = .8
REM DIFERANSIYEL UZUNLUK
CONST D = .000003
CONST IM = 7                    'MAKSIMUM AKIM 7A
CONST IH = IM * 7.5 / 200       'HISTERIZIS BANDI
DIM A(1 + PERYOD / D), B(1 + PERYOD / D), U(1 + PERYOD / D)

R = GK * 311 / IM
L = SQR(1 / GK ^ 2 - 1) * R / W
M = D * R / L
OPEN "DATA.WRI" FOR OUTPUT AS #1
WRITE #1, "TGK", "TBGK", "BK", "UN"
BASLA:
FOR IRM = IM / 80 TO IM STEP IM / 80
FOR J = 1 TO 2
FOR T = 0 TO PERYOD STEP D
  U = 311 * SIN(W * T)
  IY = D * U / L + IY * (1 - M)
  A(T / D) = IY
  B(T / D) = IY
  U(T / D) = U
```

```
  IR = IRM * SIN(W * T)
  IF U > 0 THEN IF IY - IR > IH THEN GOSUB SPHP
  IF U < 0 THEN IF IR - IY > IH THEN GOSUB SNHN
 NEXT T
NEXT J
GOTO HESAP:

SPHP:
  DO WHILE IY > IR - IH
  U = 0
  T = T + D
  IY = D * U / L + IY * (1 - M)
  B(T / D) = IY
  A(T / D) = 0
  U(T / D) = U
  IR = IRM * SIN(W * T)
  IF IR < 0 THEN RETURN
  LOOP
  RETURN

SNHN:
  DO WHILE IY < IR + IH
  U = 0
  T = T + D
  IY = D * U / L + IY * (1 - M)
  B(T / D) = IY
  A(T / D) = 0
  U(T / D) = U
  IR = IRM * SIN(W * T)
  IF IR > 0 THEN RETURN
  LOOP
  RETURN

HESAP:
CLS
LOCATE 1, 20: PRINT "AKIM KONTROLLLU PWM SIMULASYONU"
LOCATE 2, 1: PRINT "GK="; GK; "  IM="; IM; "  IRM="; IRM; "  IH="; IH; "  R="; R;
"ohm"; "  L="; L * 1000; "mH"
K = PERYOD / D

AEF = 0: BEF = 0: UEF = 0

FOR I = 0 TO K
LINE (100 + I * 250 / K, 100 - 50 * A(I) / IRM)-(100 + I * 250 / K, 100 - 50 * A(I + 1) / IRM),
1i
PSET (100 + I * 250 / K, 250 - 50 * SIN(W * I * D)), 13
LINE (100 + I * 250 / K, 250 - 50 * B(I) / IRM)-(100 + I * 250 / K, 250 - 50 * B(I + 1) / IRM),
2
LINE (100 + I * 250 / K, 400 - 50 * U(I) / 311)-(100 + I * 250 / K, 400 - 50 * U(I + 1) / 311), 2
```

```
AEF = AEF + A(I) ^ 2
BEF = BEF + B(I) ^ 2
UEF = UEF + U(I) ^ 2
NEXT I
AEF = AEF / K: AEF = SQR(AEF)
BEF = BEF / K: BEF = SQR(BEF)
UEF = UEF / K: UEF = SQR(UEF)
LOCATE 7, 1: PRINT "IGEF="; INT(100 * AEF) / 100: LOCATE 16, 1: PRINT "IÇEF=";
INT(100 * BEF) / 100: LOCATE 25, 1: PRINT "UCEF="; INT(100 * UEF) / 100


REM ************FOURIER ANALIZI**************

CONST NE = 30
DIM AN(NE), BN(NE), FI(NE), CN(NE)
FOR N = 1 TO NE STEP 2
AN(N) = 0: BN(N) = 0
FOR T = 0 TO PERYOD STEP D
AN(N) = AN(N) + (D * A(T / D) * COS(N * W * T))
BN(N) = BN(N) + (D * A(T / D) * SIN(N * W * T))
NEXT T
AN(N) = INT(100 * AN(N) * 2 / PERYOD) / 100
BN(N) = INT(100 * BN(N) * 2 / PERYOD) / 100
IF BN(N) <> 0 THEN FI(N) = ATN(AN(N) / BN(N))
CN(N) = SQR(AN(N) ^ 2 + BN(N) ^ 2) / SQR(2)
NEXT N

 LOCATE 3, 50: PRINT " N   AN    BN    CN    Fi"
 FOR I = 1 TO NE STEP 2
 LOCATE (I + 1) / 2 + 3, 50: PRINT I; " "; INT(1000 * AN(I)) / 1000; TAB(61); INT(1000 *
BN(I)) / 1000; TAB(69); INT(100 * CN(I)) / 100; TAB(75); INT(180 * FI(I) / PI)
 NEXT I

 LOCATE 19, 54: PRINT "UÇEF/U  COS(FI(1)) IÇ1/IÇEF"
 LOCATE 20, 54: PRINT INT(100 * UEF / 220) / 100; "    "; INT(1000 * COS(FI(1))) / 1000; "
"; INT(100 * CN(1) / BEF) / 100

LINE (400, 400)-(600, 400), 3
FOR I = 1 TO NE STEP 2
LINE (400 + I * 200 / NE, 400)-(400 + I * 200 / NE, 400 - 100 * CN(I) / CN(1)), 2
NEXT I


 TBGK = COS(FI(1))
 TBK = CN(1) / BEF
 TGK = TBK * TBGK
 BK = SQR(1 - (TBK) ^ 2)
```

```
REM *****ÇIKIŞ GERILIMININ 1. HARMONIGININ EFEKTIF DEĞERİNİN
BULUNUŞU*****
N = 1
FOR T = 0 TO PERYOD STEP D
AN(N) = AN(N) + (D * U(T / D) * COS(N * W * T))
BN(N) = BN(N) + (D * U(T / D) * SIN(N * W * T))
NEXT T
AN(N) = INT(100 * AN(N) * 2 / PERYOD) / 100
BN(N) = INT(100 * BN(N) * 2 / PERYOD) / 100
CN(N) = SQR(AN(N) ^ 2 + BN(N) ^ 2) / SQR(2)
CH$ = INKEY$
IF CH$ = "F" OR CH$ = "f" THEN GOTO KAYDET
IF CH$ = CHR$(27) THEN GOTO SON
WRITE #1, 0
WRITE #1, INT(TGK * 1000)
WRITE #1, INT(TBGK * 1000)
WRITE #1, INT(BK * 1000)
WRITE #1, INT(1000 * CN(1) / 220)

NEXT IRM
PRINT "TBGK,TBK,UÇ1/UG değerlerinin 1000 katı DATA.WRI dosyasına kaydedildi."
DO WHILE INKEY$ = ""
LOOP

SON:
SCREEN 0
CLOSE #1

KAYDET:
OPEN "DATA1.WRI" FOR OUTPUT AS #2
WRITE #2, K
FOR I = 1 TO K
WRITE #2, A(I)
WRITE #2, B(I)
WRITE #2, U(I)
NEXT I
CLOSE #2

END
```

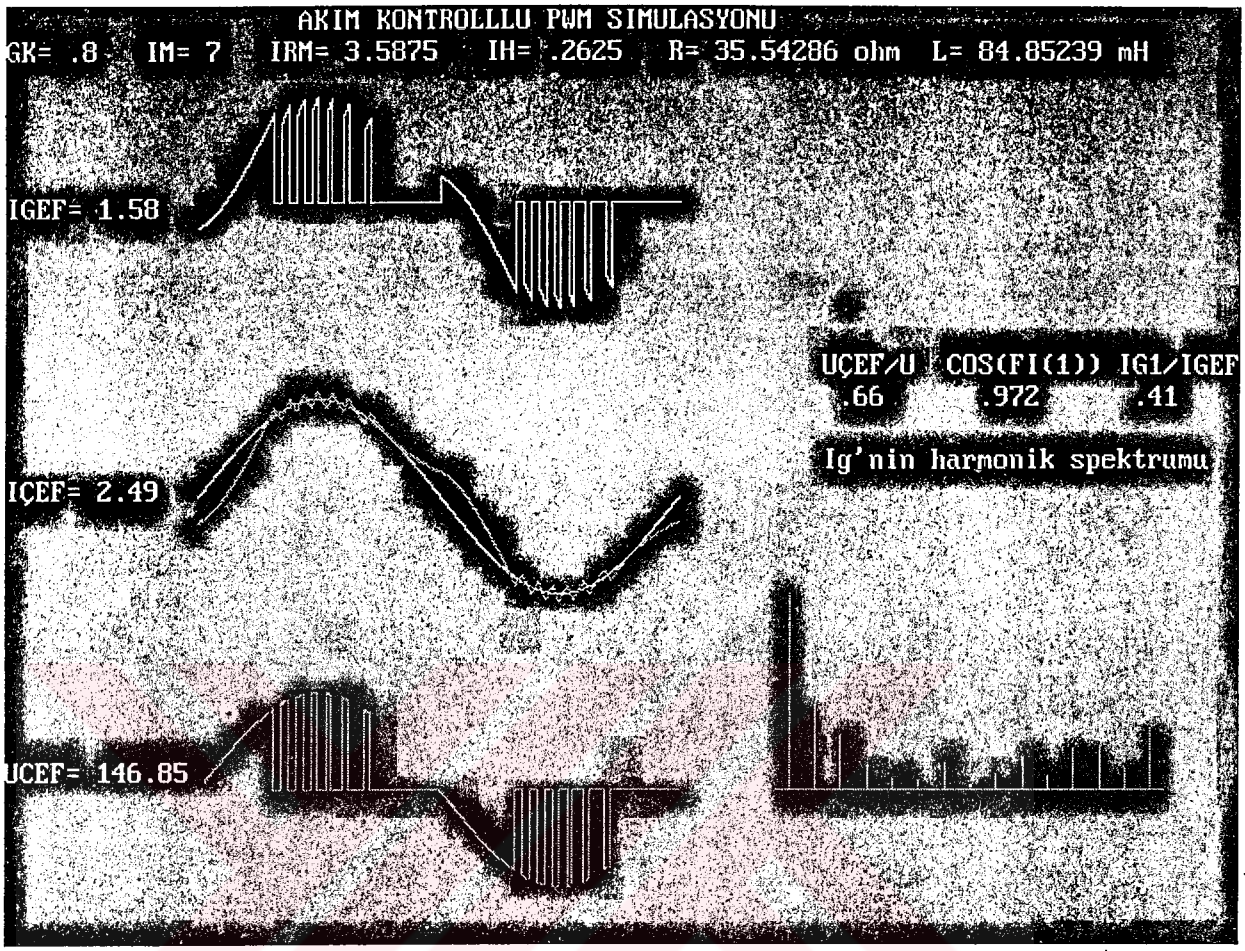
Simülasyon programından IH = %15 I_{YM} değeri ve değişik referans akımları için elde edilen yük akımı, şebeke akımı ve yük geriliminin dalga şekilleri, temel bileşen güç katsayısı TBGK=$\cos(\varphi_1)$, girişteki toplam güç katsayısı TGK= $(I_1/I).\cos(\varphi_1)$ için I_1/I değeri, ve UÇEF/U Şekil 2.2, Şekil 2.3, Şekil 2.4, Şekil 2.5'te verilmiştir.



Şekil 2.2. 1 A ' lik referans akım için I_g, $I_ç$ ve $U_ç$ değişimleri.

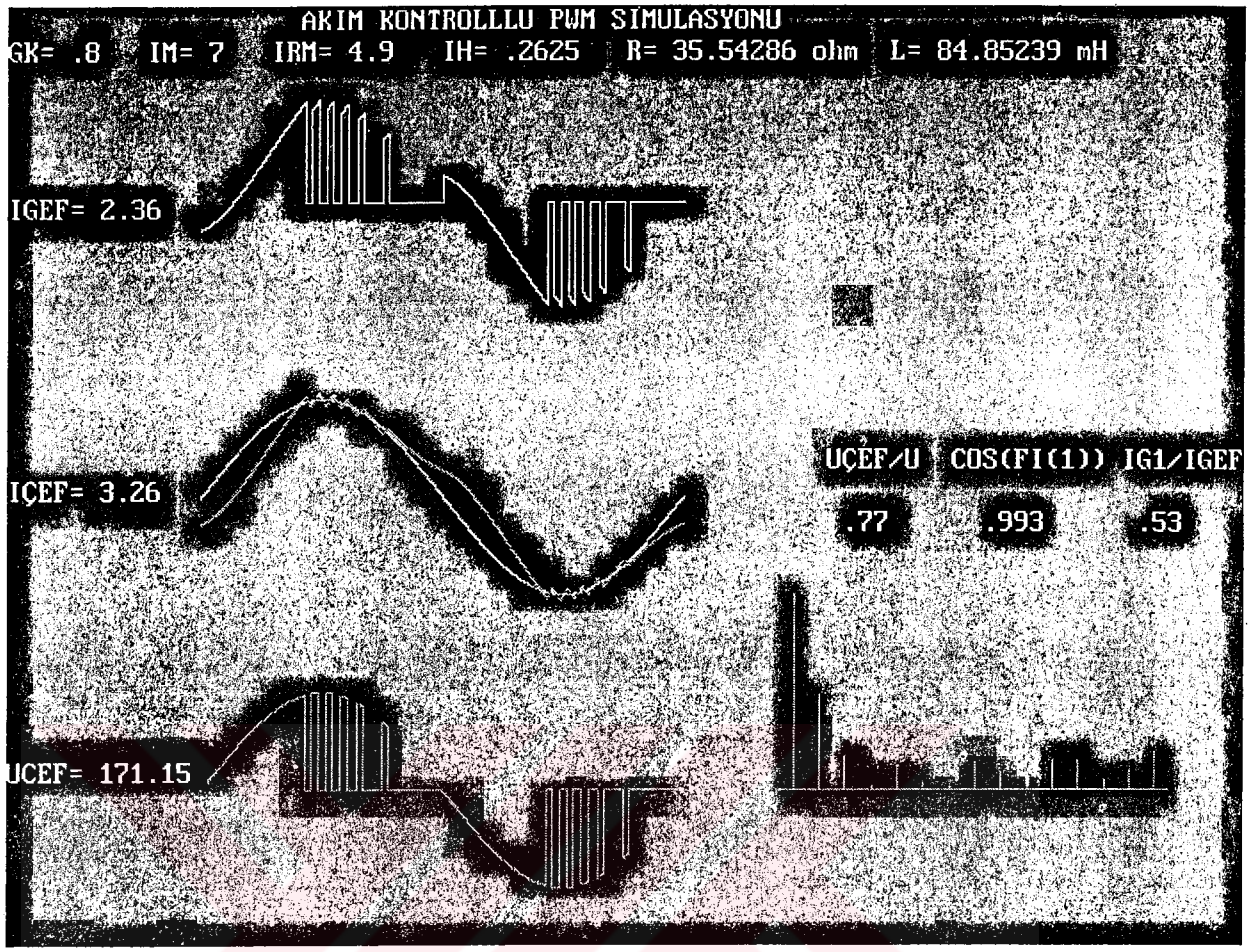
1.07 A'lik referans akım için I_{GEF} = 0.45 A , $U_{ÇEF}$ = 94 V, TBGK=cos(φ_1) = 0.95

I_1/I = 0.17 ve TGK= (I_1/I).cos(φ_1) = 0.17x0.95 olmaktadır.



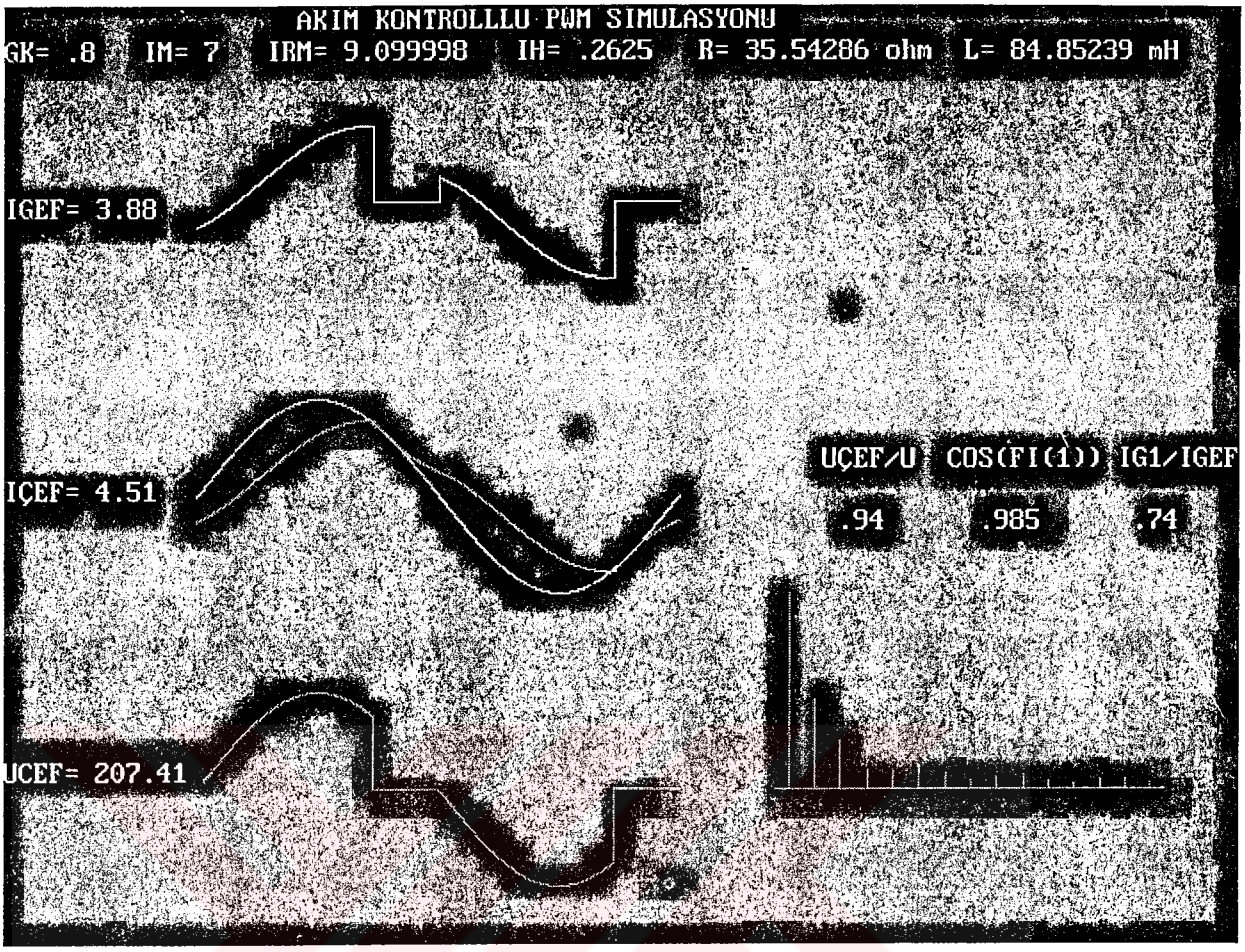
Şekil 2.3. 2.5 A'lik referans akım için I_g, $I_ç$ ve $U_ç$ değişimleri.

2.5 A'lik referans akım için I_{GEF} = 1.5 A, $U_{ÇEF}$ = 146 V, TBGK=cos(φ_1) = 0.972, I_1/I = 0.41 ve TGK= (I_1/I).cos(φ_1) = 0.41 x 0.972 olmaktadır.



Şekil 2.4. 3.2 A'lik referans akım için I_g, $I_ç$ ve $U_ç$ değişimleri.

3.2 A'lik referans akım için I_{GEF} = 2.36 A, $U_{ÇEF}$ = 171.15 V , TBGK=cos(φ_1) = 0.993, I_1/I = 0.53 ve TGK= (I_1/I).cos(φ_1) = 0.53 x 0.99 olmaktadır.



Şekil 2.5. 4.5 A'lik referans akım için I_g, $I_ç$ ve $U_ç$ değişimleri.

4.5 A'lik referans için I_{GEF} = 3.88 A, $U_{ÇEF}$ = 207 V, TBGK=cos(φ_1) = 0.985, I_1/I = 0.71

TGK= $(I_1/I).\cos(\varphi_1)$ = 0.71 x 0.985 olmaktadır.

Sonuç olarak akım kontrollu asimetrik PWM ac kıyıcı devresinde şebeke güç katsayısının 1'e oldukça yakın olduğu ve çıkış harmoniklerinin kabul edilebilir bir seviyede kaldığı görülmektedir. Akım kontrollu asimetrik PWM ac kıyıcı devresi pratik olarak uygulanabilirlik ve giriş güç katsayısı açısından, diğer PWM metodlarına tercih edilebilmektedir.

3. AKIM KONTROLLU PWM AC KIYICI İLE ASENKRON MOTOR KONTROLU

3.1. ASENKRON MOTORLAR

Asenkron motorlar, stator ve rotor olarak adlandırılan iki kısımdan meydana gelir. Stator sabit kısım olup, motorun gövdesini oluşturur. Rotor ise hareketli kısımdır. Statorda daima döner alanı oluşturan sargılar bulunup, rotorda sargı bulunmayabilir. Stator sargılarına uygulanan gerilim, senkron hız ile dönen bir döner alan oluşturur. Döner alan, hava aralığında sinüsoidal bir akı meydana getirir. Bu akının sargılarda endüklediği elektromotor kuvveti rotordan I_r akımının akmasına neden olur. I_r rotor akımı ve ϕ döner alanının oluşturduğu T_d döndürme momenti ile rotor w_m hızıyla döner. Senkron açısal hız,

$$w_S = \frac{2w}{p} \tag{3.1}$$

olarak ifade edilir. Burada p kutup sayısı ve w rad/s olarak kaynak frekansıdır. Eğer stator gerilimi $V_S = \sqrt{2}.V_S \cdot \sin wt$ ise, rotorda (3.2) formülü ile ifade edilen bir akı oluşur.

$$\varnothing(t) = \varnothing_m . \cos(w_m t + \delta - w_S t) \tag{3.2}$$

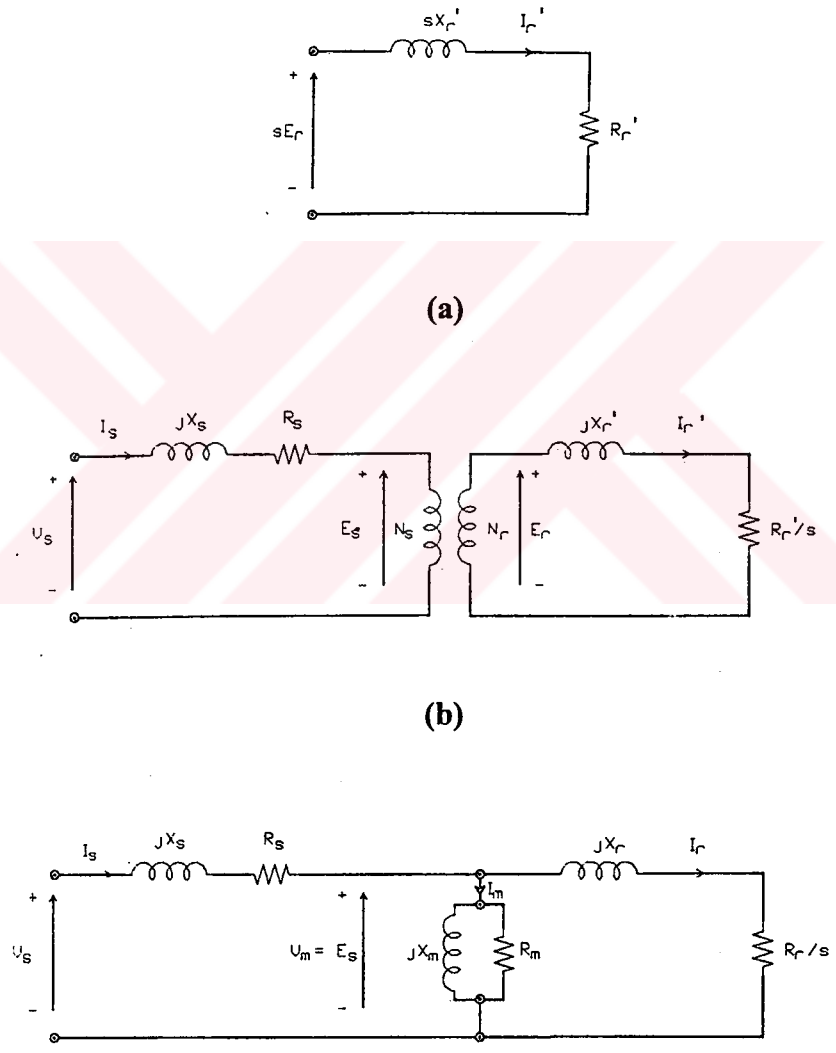
Rotor sargılarının bir fazında endüklenen gerilim (3.3) nolu formülde verildiği gibidir.

$$\begin{aligned} e_r = N_r \frac{d\Phi}{dt} &= N_r \frac{d}{dt}\left[\Phi_m \cos\left(w_m t + \delta - w_S t\right)\right] \\ &= -N_r \Phi_m \left(w_S - w_m\right) \sin\left[\left(w_S - w_m\right)t - \delta\right] \\ &= -sE_m \sin\left(sw_S t - \delta\right) \\ &= -s\sqrt{2}E_r \sin\left(sw_S t - \delta\right) \end{aligned} \tag{3.3}$$

Burada N_r her bir rotor fazındaki sarım sayısı, w_m rotorun açısal hızı, δ rotorun bağıl pozisyonu ve E_r rotorun bir fazında endüklenen gerilimin efektif değeridir. s kayma ifadesi (3.4) nolu formülde verilmiştir.

$$s = \frac{w_S - w_m}{w_S} \tag{3.4}$$

Asenkron motorun eşdeğer devresi Şekil 3.1'de gösterilmektedir.



Şekil 3.1. Asenkron motorların devre modeli.

(a) Rotor devresi. (b) Stator ve rotor devresi. (c) Eşdeğer devre.

Rotorun bir fazının eşdeğer devresi Şekil 3.1a 'da gösterilmiştir. Burada R_r' ' rotor sargılarının bir fazının direnci, X_r' ' rotorun bir fazının kaynak frekansındaki reaktansı, E_r ise hız sıfır (s=1) olduğunda endüklenen faz geriliminin efektif değerini gösterir. Rotor akımı;

$$I_r' = \frac{sE_r}{R_r' + jsX_r'} = \frac{E_r}{R_r'/s + jX_r'} \tag{3.5}$$

olarak verilir. r indisi rotor sargılarını göstermektedir.

Asenkron motorun bir fazının devre modeli Şekil 3.1b'de gösterilmiştir. Burada, R_S ve X_S stator sargılarının bir fazının direnci ve reaktansıdır. Statora indirgenen bütün değerler ile tüm devre modeli Şekil 3.1c'de gösterilmiştir. Burada R_m uyarma kayıplarına neden olan direnç, X_m ise mıknatıslama reaktansıdır. Stator demir kayıpları kaynak bağlandığında oluşacaktır. Rotor demir kayıpları ise kaymaya bağlıdır. Motorun dönmesiyle $P_{S,V}$ sürtünme ve vantilasyon kayıpları ortaya çıkar.

3.1.1. ASENKRON MOTORUN KARAKTERİSTİKLERİ

I_r rotor akımı ve I_S stator akımı Şekil 3.1c'de gösterilen devre modelinden bulunabilir. I_r ve I_S bilinirse üç fazlı motorun güç ve moment ifadeleri aşağıdaki gibi elde edilir.

Stator bakır kayıpları,

$$P_{su} = 3.I_s^2.R_s \tag{3.6}$$

Rotor bakır kayıpları,

$$P_{ru} = 3.I_r^2.R_r \tag{3.7}$$

Demir kayıpları,

$$P_c = \frac{3V_m^2}{R_m} \cong \frac{3V_s^2}{R_m} \tag{3.8}$$

Döner alan gücü,

$$P_g = 3I_r^2 \frac{R_r}{w_s} \tag{3.9}$$

Mekanik güç,

$$P_d = P_g - P_{ru} = \frac{3I_r^2 R_r}{s}(1-s) \tag{3.10}$$

$$= P_s(1-s) \tag{3.11}$$

Mil momenti,

$$T_d = \frac{P_d}{w_m} \tag{3.12}$$

$$= \frac{P_g(1-s)}{w_s(1-s)} = \frac{P_g}{w_s} \tag{3.12a}$$

Şebekeden çekilen elektriki güç,

$$P_i = 3V_s.I_s.\cos\theta_m \tag{3.13}$$

$$P_i = P_c + P_{su} + P_g \tag{3.13a}$$

Burada θ_m, V_s. ve I_s arasındaki açıdır. Çıkış gücü $P_o = P_d - P_{s,v}$ olarak bulunur.

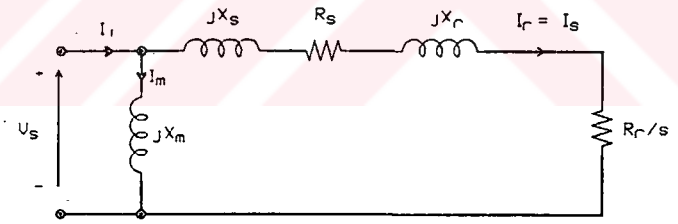
Verim;

$$\eta = \frac{P_o}{P_i} = \frac{P_d - P_{s,v}}{P_c + P_{su} + P_g} \tag{3.14}$$

Eğer $P_g >> (P_c + P_{su})$ ve $P_d >> P_{s,v}$ olursa,verim takriben şu hale gelir;

$$\eta \cong \frac{P_d}{P_g} = \frac{P_g\,(1-s)}{P_g} = (1-s) \tag{3.14a}$$

X_m'nin değeri normalde R_m'den çok büyüktür. Bu nedenle hesaplamaları kolaylaştırmak için R_m devre modelinden kaldırılabilir. Eğer $X_m^2 >> (R_s^2 + X_s^2)$ olursa, bu durumda $V_s \cong V_m$ olur ve X_m sola kaydırılabilir. Bu durum Şekil 3.2 'de gösterilmiştir;



Şekil 3.2. Bir faz için yaklaşık eşdeğer devre.

Bu durumda motorun giriş empedansı,

$$Z_i = \frac{-X_m(X_s + X_r) + jX_m\left(R_s + \dfrac{R_r}{s}\right)}{R_s + \dfrac{R_r}{s} + j\left(X_m + X_s + X_r\right)} \tag{3.15}$$

Motorun güç faktörü,

$$\theta_m = \pi - tg^{-1}\frac{R_s + \frac{R_r}{s}}{X_s + X_r} + tg^{-1}\frac{X_m + X_s + X_r}{R_s + \frac{R_r}{s}} \qquad (3.16)$$

Şekil 3.2 'den, efektif rotor akımı;

$$I_r = \frac{V_s}{\left[\left(R_s + \frac{R_r}{s}\right)^2 + \left(X_s + X_r\right)^2\right]^{1/2}} \qquad (3.17)$$

I_r ifadesi, $P_g = 3I_r^2\frac{R_r}{s}$ ifadesinde yerine konup, elde edilen yeni P_g (3.12a) eşitliğinde kullanılırsa moment şu şekilde yazılabilir;

$$T_d = \frac{3R_r V_s^2}{sw_s\left[\left(R_s + \frac{R_r}{s}\right)^2 + \left(X_s + X_r\right)^2\right]} \qquad (3.18)$$

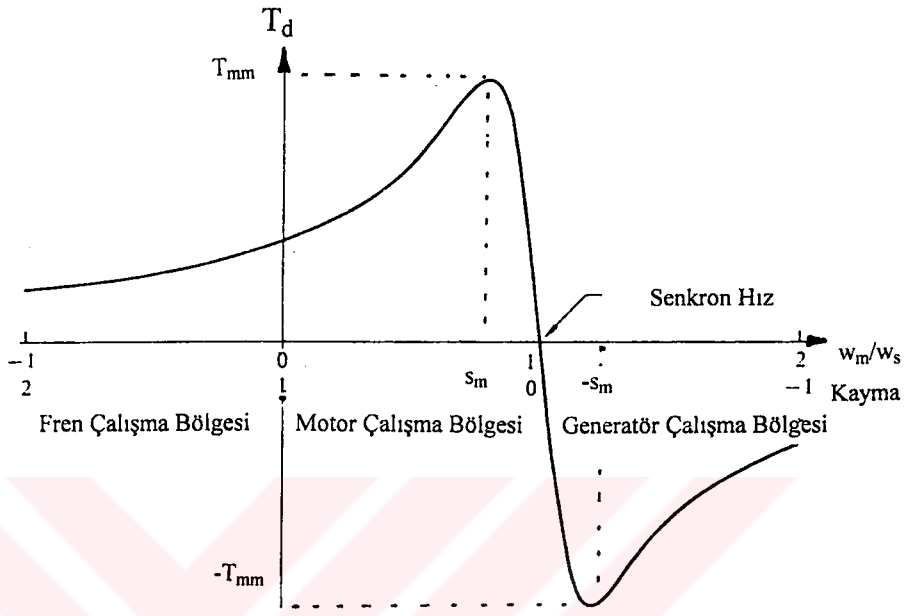
Motor sabit gerilim ve frekans ile beslenirse, motor milindeki moment kaymanın bir fonksiyonudur ve (3.18) nolu eşitikten elde edilebilir. Momentin kayma veya hızın fonksiyonu olarak değişimi Şekil 3.3'de gösterilmiştir. Burada üç çalışma bölgesi bulunmaktadır.

1) Motor çalışma bölgesi : $0 \le s \le 1$

2) Generatör çalışma bölgesi : $s < 0$

3) Frenleme bölgesi : $1 \le s \le 2$

Motor çalışma bölgesinde rotor, stator döner alanı ile aynı yönde döner. Kayma arttıkça momentte artar. Hava aralığı akısı ise sabit kalır. Moment s = s_m 'de maksimum değerine ulaştıktan sonra, hava aralığı akısının azalmasından dolayı kaymada oluşan artışla moment azalır.



Şekil 3.3. Moment - hız karakteristiği.

Generatör çalışma bölgesinde w_m, senkron hız w_s'ten daha büyüktür. w_m ve w_s aynı yönde ve kayma negatiftir. Bu yüzden R_r/s negatiftir. Yani güç milden alınarak rotor devresine verilmekte ve motor generatör olarak çalışmaktadır. Motor gücü kaynağa geri verir. Moment-hız karakteristiği motor çalışma bölgesine benzer fakat momentin değeri negatiftir.

Fren çalışma bölgesinde ise hız, döner alanın dönme yönü ile ters yöndedir. Kayma 1'den büyüktür. Eğer motor, motor çalışma bölgesinde çalışırken stator döner alanının yönünü değiştirmek için kaynağın uçları değiştirilirse, bu durum meydana gelebilir. Moment döner alan ile aynı yönde olduğundan, hareket yönüne karşı koyar ve frenleme momenti ortaya çıkar. s >1 olduğundan motor akımı yüksek olacak, fakat mil momenti düşük olacaktır. Frenlemeden dolayı oluşan enerji motorun aşırı ısınmasına neden olabilir bu yüzden harcanması gerekir. Normalde bu tip frenleme tavsiye edilmez.

Başlangıçta motor hızı $w_m = 0$ ve $s = 1$'dir. Kalkış momenti (3.18)'deki T_d ifadesinde $s = 1$ yazarak bulunabilir.

$$T_s = \frac{3R_r V_s^2}{w_s\left[\left(R_s + R_r\right)^2 + \left(X_s + X_r\right)^2\right]} \tag{3.19}$$

Maksimum moment için kayma s_m, $\frac{dT_d}{d_s} = 0$ alınarak bulunur.

$$s_m = \mp \frac{R_r}{\left[R_s^2 + \left(X_s + X_r\right)^2\right]^{1/2}} \tag{3.20}$$

(3.18)'deki T_d ifadesinde $s = s_m$ yazarak motor çalışma esnasındaki maksimum moment bulunabilir.

$$T_{mm} = \frac{3V_s^2}{2w_s\left[R_s + \sqrt{R_s^2 + \left(X_s + X_r\right)^2}\right]} \tag{3.21}$$

Generatör çalışma bölgesindeki maksimum moment (regenerative torque) ise (3.18)'de $s = - s_m$ yazarak bulunabilir.

$$T_{mr} = \frac{3V_s^2}{2w_s\left[-R_s + \sqrt{R_s^2 + \left(X_s + X_r\right)^2}\right]} \tag{3.22}$$

R_s diğer devre empedanslarına göre küçük olduğu kabul edilirse (özellikle 1 kW'dan büyük güçteki motorlarda), yukarıdaki eşitlikler aşağıdaki gibi sadeleşir.

$$T_d = \frac{3R_r V_s^2}{sw_s\left[\left(\frac{R_r}{s}\right)^2 + \left(X_s + X_r\right)^2\right]} \tag{3.23}$$

$$T_s = \frac{3R_r V_s^2}{w_s\left[R_r{}^2 + \left(X_s + X_r\right)^2\right]} \tag{3.24}$$

$$s_m = \mp\frac{R_r}{X_s + X_r} \tag{3.25}$$

$$T_{mm} = -T_{mr} = \frac{3V_s^2}{2w_s\left(X_s + X_r\right)} \tag{3.26}$$

(3.23) ve (3.24) eşitlikleri (3.26)'ya göre normalize edilirse (3.27) ve (3.28) ifadeleri elde edilir.

$$\frac{T_d}{T_{mm}} = \frac{2R_r(X_s + X_r)}{s\left[\left(\frac{R_r}{s}\right)^2 + \left(X_s + X_r\right)^2\right]} = \frac{2s.s_m}{s_m^2 + s^2} \tag{3.27}$$

$$\frac{T_s}{T_{mm}} = \frac{2R_r\left(X_s + X_r\right)^2}{R_r{}^2 + \left(X_s + X_r\right)^2} = \frac{2s_m}{s_s^2 + 1} \tag{3.28}$$

Eğer $s<1$ ve $s^2 << s_m{}^2$ ise (3.27) ifadesi aşağıdaki gibi olur,

$$\frac{T_d}{T_{mm}} = \frac{2s.}{s_m} = \frac{2(w_s + w_m)}{s_m} \tag{3.29}$$

Hız, momentin bir fonksiyonu olarak (3.30) nolu eşitlikte verilmiştir.

$$w_m = w_s\left(1 - \frac{s_m}{T_{mm}} T_d\right) \tag{3.30}$$

(3.29) ve (3.30) nolu denklemler incelendiğinde, motor düşük kayma ile çalıştığında momentin kayma ile orantılı olacağı ve hız arttıkça momentin azalacağı anlaşılmaktadır.

Asenkron motorların moment ve hızı aşağıdaki metodlar ile kontrol edilmektedir.

1) Stator gerilim kontrolu.
2) Rotor gerilim kontrolu.
3) Frekans kontrolu.
4) Stator gerilim ve frekans kontrolu.
5) Stator akım kontrolu.
6) Gerilim, akım ve frekans kontrolu.

Moment - hız ihtiyacını karşılamak için genellikle gerilim, akım ve frekans kontrolu birlikte kullanılır. Bu tezde sadece stator gerilim kontrolu incelenecektir.

3.1.2. STATOR GERİLİMİNİ DEĞİŞTİREREK YAPILAN ASENKRON MOTOR HIZ KONTROLU

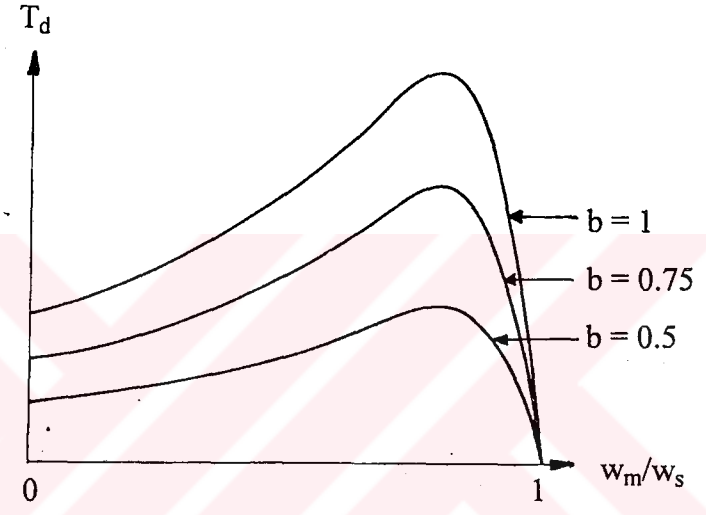
Eşitlik (3.18) 'de momentin stator geriliminin karesi ile doğru orantılı olduğu, ve stator gerilimindeki azalma ile hızın da azalacağı görülmektedir. Motorun besleme gerilimi $b.V_S$ olduğunda, moment ifadesi eşitlik (3.18)'den aşağıdaki gibi elde edilir.

$$T_d = \frac{3R_r(bV_s)^2}{sw_s\left[\left(R_s + \frac{R_r}{s}\right)^2 + (X_s + X_r)^2\right]}, \qquad b \le 1 \tag{3.31}$$

Farklı b değerleri için moment - hız karakteristikleri Şekil 3.4'te gösterilmiştir. Yük doğrusu ile kesişme noktaları kararlı çalışma noktalarını gösterir. Herhangi bir magnetik devrede endüklenen gerilim, akı ve frekansla orantılıdır. Hava aralığındaki akının efektif değeri aşağıdaki gibi ifade edilebilir.

$$V_a = b.V_s = K_m.w\,\Phi \qquad \text{veya}$$

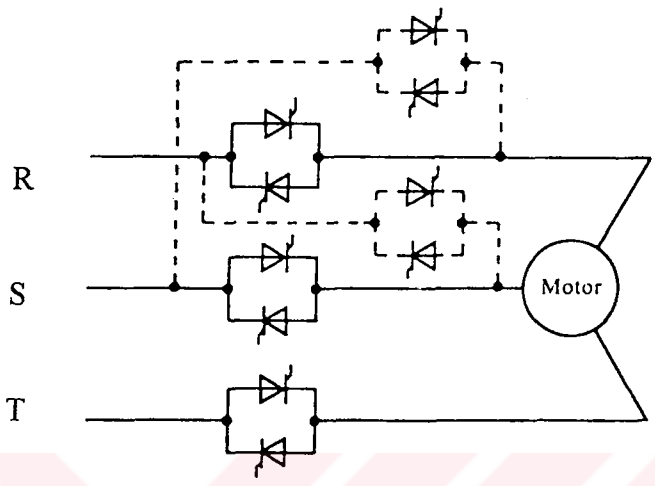
$$\Phi = \frac{V_a}{K_m w} = \frac{b.V_s}{K_m w} \tag{3.31}$$



Şekil 3.4. Stator gerilimi değiştirildiğinde moment - hız karakteristikleri.

Burada K_m sabittir ve stator sargısındaki sarım sayısına bağlıdır. Stator gerilimi azaltıldığında hava aralığındaki akı ve moment azalır. Hız kontrol aralığı, maksimum momenti veren s_m kayma değerine bağlıdır. Düşük kaymalı motorlar için, hız kontrol aralığı çok dardır. Bu şekilde yapılan gerilim kontrolü sabit momentli yükler için uygun değildir. Düşük kalkış momentine, küçük kayma değerlerinde ve dar bir hız aralığında çalışmaya ihtiyaç duyan uygulamalar için kullanılabilmektedir. Stator gerilimi AC gerilim ayarlayıcıları (AC Kıyıcı) ve PWM inverterler ile kontrol edilebilmektedir. Bununla beraber, hızın dar bir aralıkta değişmesinden dolayı gerilim kontrolünü sağlamak için AC gerilim ayarlayıcıları kullanılır. Klasik AC kıyıcılar oldukça basittir. Bununla beraber harmonik muhtevası yüksektir ve devrenin giriş güç faktörü düşüktür.

Klasik AC kıyıcılar genellikle düşük güç uygulamalarında, fan, pompa, havalandırma cihazları gibi kalkış momentinin düşük olduğu yerlerde kullanılır. Ayrıca yüksek güçlü asenkron motorlarda kalkış akımını sınırlamak için de kullanılmaktadır.



Şekil 3.5. Faz kesme metoduyla stator gerilimini değiştirerek asenkron motorda hız kontrolu.

3.2. PID KONTROLU

PID kontrol modu oran, integral ve türev kontrol modlarının kombinasyonudur. PID kontrolör aynı zamanda üç modlu kontrolör olarak da bilinmektedir. PWM AC kıyıcı ile yapılan asenkron motor hız kontrolunda, motor hızının istenilen değere hızlı ulaşması ve motorun yüklenmesi durumunda hızın değişmemesi için uygun bir kontrol yapılması gerekmektedir. Sistemin cevap verme süresinin iyileştirilmesi ve uzun süreli kararlılığının sağlanması için PID kontrolun kullanılması uygun olmaktadır.

PID kontrol yapabilmek için öncelikle kontrol edilen değişkendeki hatanın tespit edilmesi gerekir. Hata sistemin o anda bulunduğu konum ile belirli bir zaman içerisinde olması istenen konum arasındaki işaretli farkdır. İstenilen pozisyondan gerçek pozisyonun çıkartılması ile bulunur. Bu tezde incelenen motor kontrol uygulamasında hata, istenilen hızdan gerçek hızın çıkartılması ile bulunmaktadır. Bu fark daha sonra oran, integral ve türev terimlerinin sistemin çıkışına nasıl etki edeceğini belirlemede kullanılır.

Oran terimi, hatanın oran terimi kazancı ile çarpılmasıyla bulunur. Bu terim olması istenen çalışmadan uzaklıkla orantılı olarak çıkışı arttırır. Oran terimi arttıkça sistemin izleme hatası düşer. Oran kazancı büyükse, sistemin kararlılığını arttırmak için daha fazla integral ve diferansiyel terim kazançlarını eklemek gerekir.

İntegral modu büyük yük değişimlerinin yol açtığı oransal kaymayı ortadan kaldırmak için kullanılır. İntegral terimi hatanın o ana kadar toplanmasıyla elde edilir. Büyük bir değişiklik oluşturabilmek için çok küçük bir hatayı değerlendirebilmektedir. Digital bir sistemde bu, her bir örnekleme peryodunda akümülatöre bir sayı eklenerek yapılır. Dolayısıyla bir süre sonra ortaya çıkan küçük bir hata, sistemi harekete geçirebilecek bir çıkış oluşturabilen yeterli büyüklükte bir sayı üretebilecektir. Hatanın zaman içerisindeki toplamı bir katsayı ile çarpılır. Bu terim sistemin uzun süreli doğruluğunun iyi olmasını sağlar. Bu amaç için bir miktar yeterli olup aşırısı osilasyonlara sebep olur.

Türev modu özellikle sistemde ani yük değişimleri varsa daha uygundur. Türev terimi hatanın bir önceki çevrimden itibaren ne kadar değiştiğini inceler. Bunu eski hatadan yeni hatayı çıkararak yapar.

İdeal üç modlu kontrolör eşitlik (3.32)'de tanımlanmaktadır.

$$v = P.\,e + P.\,I.\int_0^t e.\,dt + P.\,D.\,\frac{de}{dt} + v_o \qquad (3.32)$$

Pratik PID kontrolörü türev kısıtlama terimi de içermektedir. Bu kontrolörün zaman domenindeki, frekans domenindeki ifadeleri ve transfer fonksiyonu (3.33), (3.34) ve (3.35)'te verilmiştir.

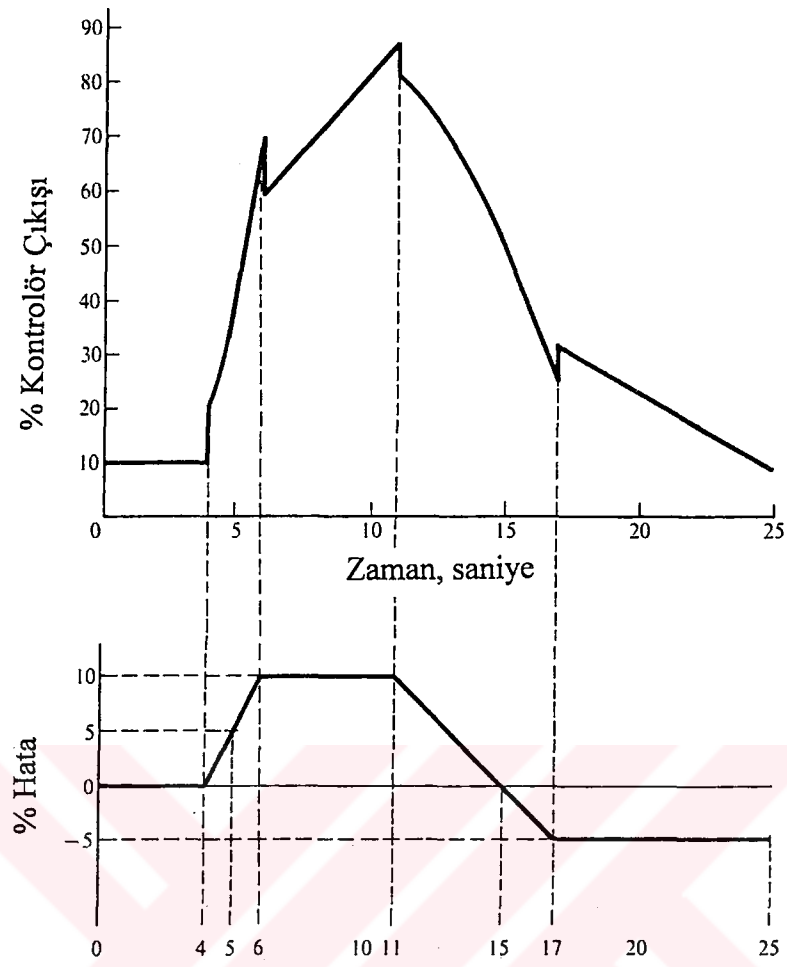
$$v = P.\,e + P.\,I.\int_0^t e.\,dt + P.\,D.\,\frac{de}{dt} - \alpha.D.\frac{de}{dt} + v_o \qquad (3.33)$$

$$V = P.\,E + P.\,\frac{I}{S}.\,E + P.\,D.\,s.\,E - \alpha.\,D.\,s.\,V \qquad (3.34)$$

$$\frac{V}{E} = P.\,\frac{I + s + D.\,s^2}{s + \alpha.\,D.\,s^2} \qquad (3.35)$$

Hata değişimi bilinen ve parametreleri aşağıda verilen bir PID kontrolörün çıkışının nasıl değiştiğini inceleyelim. P = 4.3, I = 1 / 7 1/s, D = 0.5 s, v_0 = % 10. Çizim yapılırken türev kısıtlama terimi ihmal edilmiştir.

PID kontrolörün katsayıları (3.33) nolu denklemde yerine konulursa kontrolör çıkışının ifadesi $v = 4{,}3\,.\,e + 0{,}614\int_0^t e.\,dt + 2{,}15\,\frac{de}{dt} + 10$ olarak bulunur. Kontrol çıkışının hataya göre zamana bağlı değişimi Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Kontrol çıkışının hataya göre zamana bağlı değişimi.

3.2.1. ANALOG PID KONTROLU

Elektronik analog kontrolör hata algılama ve kontrol bölümü olmak üzere iki ana bölümden oluşur. Hata istenilen değer (Set Point = SP)'den kontrol edilen değişkenin ölçülen değeri (c_m = measured value of controlled variable) çıkartılarak bulunur. Analog PID kontrolörün şeması Şekil 3.7'de gösterilmiştir. Kontrolörün transfer fonksiyonu aşağıda verilmiştir.

$$\frac{V}{E} = P.\left[\frac{1+s}{s}\right].\left[\frac{1+D.s}{1+\alpha.D.s}\right] \qquad (3.36)$$

Devrede PID katsayılarını direnç ve kondansatörler belirlemektedir. Katsayılar aşağıdaki eşitlikler yardımıyla bulunabilir.

$$P = \frac{R_i}{R_1 + R_d}$$

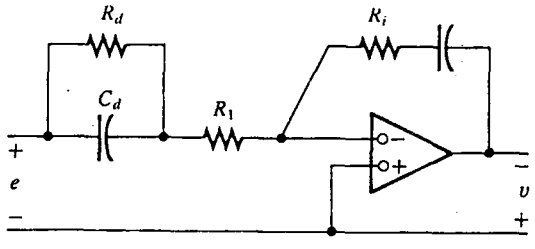
$$\alpha = \frac{R_1}{R_1 + R_d} \qquad (3.37)$$

$$I = \frac{1}{R_i . C_i}$$

$$D = R_d . C_d$$

Kontrolör parametreleri P=4, I=1/7 s^{-1}, D=0.5s, α=0.1 (türev kısıtlama terimi) olan analog PID kontrolör tasarlamak için C_i=10 µF (seçilirse) R_i=700 kΩ, R_1=17,5 kΩ, C_d=3,2 µF, R_d=157,5 kΩ bulunmaktadır. Bu durumda kontrolörün transfer fonksiyonu,

$$\frac{V}{E} = \frac{1 + 7{,}5s + 3{,}5s^2}{7s + 0{,}35s^2}$$ olarak bulunmaktadır.



Şekil 3.7. Analog PID Kontrolörün Devre Şeması.

3.2.2. DİJİTAL PID KONTROLU

Endüstriyel kontrol sistemlerinde mikroişlemci-tabanlı dijital kontrolörler yaygın olarak kullanılmaktadır. Dijital kontrolörlerin popüler olması için birçok sebeb mevcuttur. Mikroişlemciler, adaptif kendiliğinden ayarlama, çok değişkenli kontrol ve uzman sistemler gibi gelişmiş özellikler sağlamaktadır. Ayrıca mikroişlemcinin bir ağ üzerinden haberleşmesi dijital kontrolörün kendini kabul ettirmesinin başka bir nedenidir. Kapalı çevrim kontrolunda kullanılan dijital kontrolörler genellikle PI, PD veya PID kontrol modlarını gerçekleştirirler. Bu kısımda dijital PID kontrolör incelenecektir.

Dijital kontrolör kontrol edilen değişkeni Δt örnekleme zamanı olarak adlandırılan zaman aralıkları ile ayrılmış belirli zamanlarda ölçmektedir. Kontrol edilen değişkenin herbir örneğini (veya ölçümünü) mikrobilgisayara giriş olarak verebilmek için ikilik düzende sayıya dönüştürmek gerekir. Mikrobilgisayar, ölçülen değişkenin herbir örneğini istenilen çalışma değerinden çıkartarak hata örneklerine ait bir küme oluşturur.

e1 = sp - cm1	birinci hata örneği
e2= sp - cm2	ikinci hata örneği
e3= sp - cm3	üçüncü hata örneği
.	
.	
.	
en= sp - cmn	şu andaki hata örneği

Herbir hata örneği hesaplandıktan sonra dijital PID kontrolörü PID algoritmasını takip ederek, e1, e2, ... en hata örneklerine göre kontrolör çıkışını hesaplar. PID kontrol algoritması konumsal veya artış türünde olabilir.

Konumsal PID algoritması hata işaretlerini kullanarak sistemin konumunu (v_n), mesela valf pozisyonunu, belirler. Eşitlik (3.38)'de konumsal algoritmanın basitleştirilmiş ifadesi verilmektedir.

$$v_n = P.\, e_n + P.\, I.\, \Delta t.\, \sum_{j=1}^{n} e_j + P.D.\frac{\Delta e_n}{\Delta t} \qquad (3.38)$$

v_n = sistemin şu andaki konumu (% olarak)

P = kontrolör kazancı

e_n = şu andaki hata örneği (% olarak)

Δt = örnekleme zamanı, saniye

I = integral işlemi katsayısı, 1/saniye

D = türev işlemi zaman sabiti, saniye

$\Delta e_n = e_n - e_{n-1}$ = hata işaretindeki değişme olarak tanımlanmaktadır.

Artış türündeki PID algoritması step motor gibi çıkışları artan sistemler için daha uygundur. Konumsal algoritma daha doğaldır. Eğer örnekleme zamanı Δt, integral işlemi zaman sabitinden (1 / I) çok daha küçük ise, konumsal algoritma analog kontrolöre benzer bir sonuç verecektir.

Konumsal PID algoritmasının akış diyagramı Şekil 3.8'de verilmiştir. Akış diyagramında;

cmn= kontrol edilen değişkenin n. örneği,

sp = (set point) istenilen çalışma noktası,

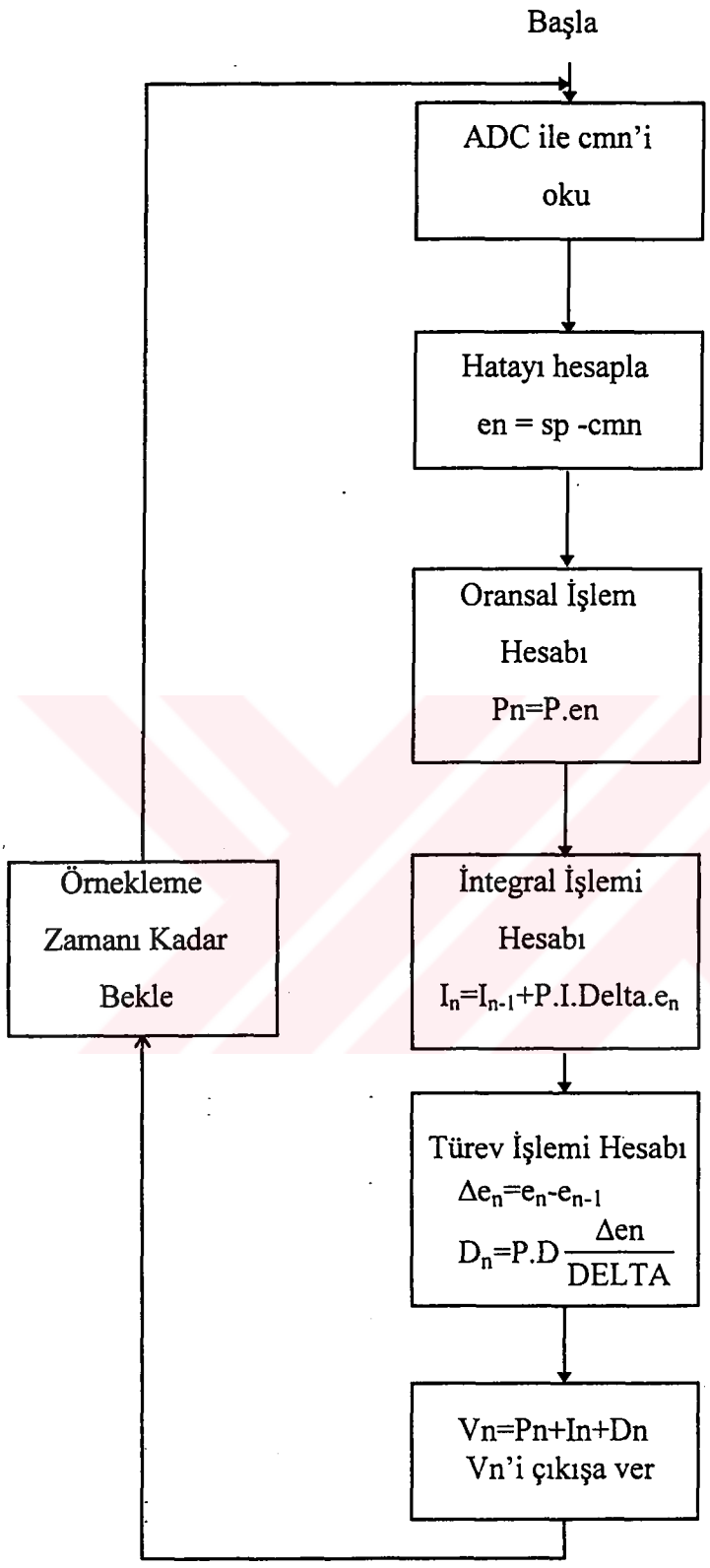
en = n. hata örneği,

In = integral modu işlemi,

Pn = oran modu işlemi,

Dn = türev modu işlemi,

DELTA=örnekleme zamanı olarak kısaltılmıştır.



Şekil 3.8. Konumsal PID Kontrol Algoritması

4. MİKRODENETLEYİCİ KULLANILARAK YAPILAN PWM AC KIYICI İLE ASENKRON MOTOR KONTROLU UYGULAMASI

4.1. GİRİŞ

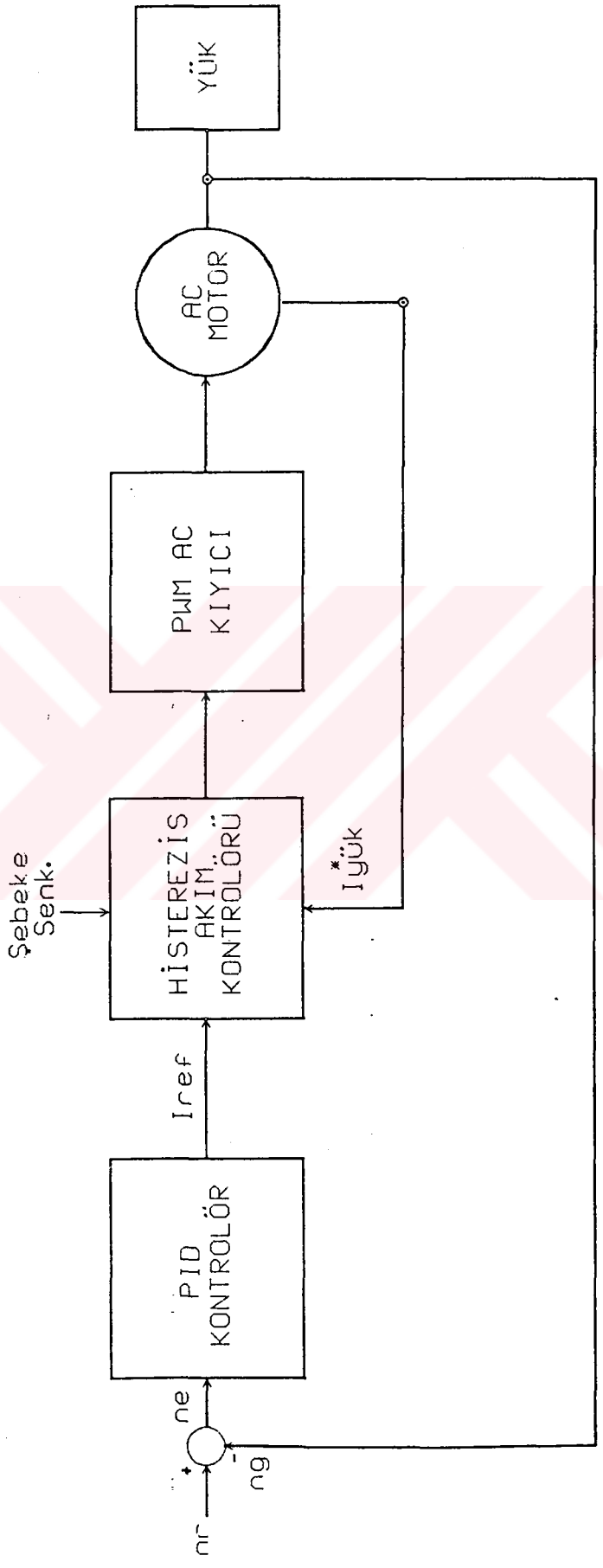
Bu bölümde mikrodenetleyici kullanılarak yapılan PWM AC kıyıcı ile gerçekleştirilen asenkron motor kontrol sistemi tanıtılmaktadır. Kontrol sisteminin genel blok diyagramı Şekil 4.1'de görülmektedir. Sistem motorun istenilen hızda dönmesini sağlamak üzere kapalı çevrim içerisinde motor hızını ve yük akımını değerlendirerek motor için gerekli kontrol sinyalini üretmektedir. Kontrol sistemi ile, motorun istenilen hıza ulaşma süresinin kısa olması ve yük değişimleri ile motor hızının değişmemesi hedeflenmiştir.

Sisteme giriş olarak n_r referans hızı girilmektedir. Motorun milinden okunan n_g hızı ile hız geribeslemesi yapılmaktadır. Referans hız ile geribesleme hızı arasındaki fark sistemdeki n_e hız hatasını vermektedir.

$$n_e = n_r - n_g \tag{4.1}$$

n_e hız hatası PID kontrolör ile değerlendirilmekte, hatanın zamana göre değişimine bağlı olarak kontrolör çıkışında I_{ref} referans akımı üretilmektedir. PID kontrolör motor hızının istenilen referans hıza hızlı bir şekilde ulaşmasını ve yükteki ani değişimleri karşılayabilmesini sağlayacak gerekli çıkışı üretmektedir. PID kontrolörün P, I ve D kazançları sistem ile uyum içerisinde olacak şekilde seçilmiştir.

PID kontrolör çıkışında üretilen referans akım, yükten alınan I_y^* örnek yük akımı ve şebekeye ait bilgi histerezis akım kontrolörü tarafından değerlendirilmektedir. Histerezis akım kontrolörü yük akımının seçilen referans akım etrafında belli bir band içerisinde kalmasını sağlamakta, şebekeye ait bilgiyi de kullanarak güç devresine ait elemanlardan hangilerinin tetikleneceğine karar vererek PWM AC kıyıcı devresini kontrol etmektedir. PWM AC kıyıcı devresinin çıkışı PWM şeklinde değişen bir gerilimdir. Dolayısıyla kontrol sisteminin çıkışı motor uçlarındaki gerilimin efektif değerini kontrol etmektedir.



Şekil 4.1. Kontrol sisteminin genel blok diyagramı.

Motor durmakta iken motorun istenilen hıza ulaşıncaya kadar kontrol sisteminin nasıl çalıştığını ve kontrol işaretlerinin nasıl değiştiğini sayısal bir örnek üzerinde inceleyelim. Sistemin girişindeki referans hız n_r = 1000 d/d olsun. Motorun başlangıç hızı n_g = 0'dır. Hızdaki hata hesaplanarak motorun hızlanması gerektiğine karar verilmektedir. Başlangıçta hızdaki hata en büyük değerini almaktadır. Motorun istenilen hıza en kısa zamanda ulaşabilmesi için PID kontrolör çıkışında I_{ref} akımı %100 seviyesinde olacaktır. Motorun I_{ref} akımının maksimum seviyesinde akım çekebilmesi için motor gerilimi otomatik olarak histerezis akım kontrolörü ve PWM ac kıyıcı devresi tarafından belirlenmektedir. Eğer akım kontrolü yapılmasaydı ve şebeke gerilimi doğrudan asenkron motora uygulansaydı motor kalkış anında nominal akımının 6-8 katı bir akım çekecekti. Buradaki kontrol sistemi ile motor aşırı akım çekmeden, kontrol sistemi tarafından belirlenen referans akımının maksimum değeri ile kalkmakta dolayısıyla bu şekilde kalkış yapabilmesi için motor uçlarında olması gereken gerilimin efektif değeri başlangıçta düşük bir seviyede tutulmaktadır. Kontrol sisteminde referans akımın maksimum değeri motorun nominal akımının 2 katı olarak seçilmiştir. n_g=0 iken motor nominalin 2 katı akımla kalkış yapmakta ve motorun uçlarındaki gerilimin efektif değeri 30 V olmaktadır. Motor hızlandıkça hızla orantılı olarak motorda bir gerilim endüklenecek ve hızdaki hata azalacaktır. Motor hızı 300 d/d'ya ulaşmış olsun ve motorda hızla orantılı olarak endüklenen gerilim 25 V olsun. Hata bir önceki duruma göre daha düşük bir değerdedir. PID kontrolör, çıkışındaki referans akımı, hatanın azalmaya başlayarak belli bir değerin altına düşmesiyle uygun bir şekilde azaltarak hata sıfır olduğunda referans akımı yükün ihtiyacı olan akım değerine getirmeye çalışacaktır. Motor hızı 300 d/d'ya yükseldiğinde PID kontrolör çıkışındaki referans akımın % 90 seviyesine düştüğünü kabul edelim. Bu durumda motor uçlarında oluşan gerilim, hızla orantılı olarak motor uçlarında endüklenen gerilim ile motordan geçen akımla orantılı olarak oluşan gerilim düşümünün toplamı olmalıdır. Bu gerilim otomatik olarak PWM ac kıyıcı tarafından sağlanacaktır. Son olarak motorun istenilen 1000 d/d hız değerine ulaştığını düşünelim. Motorun bu hız değerinde kararlı bir şekilde dönebilmesi için motor akımının yükün çektiği akımı karşılaması gerekir.

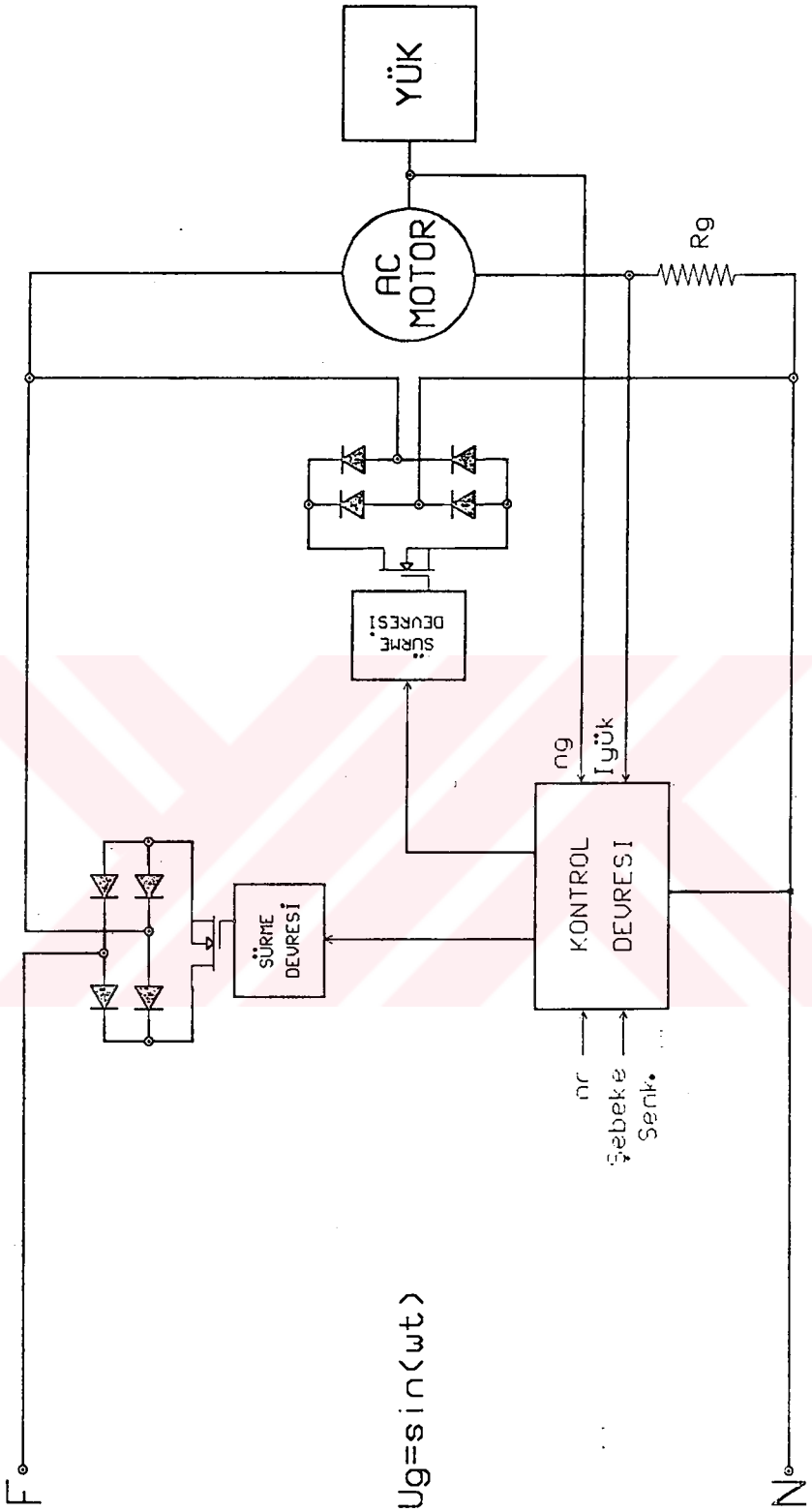
Yükün 1000 d/d hızda çektiği akım, referans akımın maksimum değerinin % 20 'si olsun. Bu durumda PID kontrolör hatanın sıfır olduğunu görerek referans akımı % 20 seviyesine düşürecektir. Motor uçlarında oluşan yeni gerilim artık motorun istenilen hızda dönmesini sağlayacak gerilimdir. Uygulamada PID kontrolörün hatanın sıfıra düştüğünü görür görmez akımı istenilen değere getirmiş olması düşük bir ihtimaldir. Dolayısıyla hata bir süre negatif ve bir süre de pozitif değerlerde dolaşarak sönümlü olarak sıfıra inecek sistem kararlı çalışma noktasına bir süre sonra oturacaktır.

Motorun kararlı çalışma noktasında çalışmasını sürdürebilmesi için, istenilen referans hızdaki yük momentini karşılayacak akımın motordan geçmesi ve motorun istenilen referans hızda dönebileceği bir gerilimle beslenmesi gerekmektedir. Kararlı çalışma akım ve geriliminin sadece bir değeri için mevcuttur. PWM AC kıyıcı ile asenkron motorun stator geriliminin efektif değerini değiştirerek yapılan kontrol, vantilatör karakteristikli yükler için uygundur. Bu özellikteki yüklerde moment $K.n^2$ ile orantılı olarak değişmekte, hızdaki bir değişme momentin , dolaylı olarak akımın karesel olarak artmasına veya azalmasına neden olmaktadır. Hızın bir miktar değişmesi akımda büyük bir değişiklik ihtiyacı ortaya çıkarmaktadır. Kontrol sisteminin yükteki ani değişimlere de iyi cevap verebilmesi için herhangi bir kararlı çalışma noktasındaki akım ve gerilim değerinden, başka bir kararlı çalışma noktasındaki akım ve gerilim değerlerine geçişini uygun bir şekilde sağlamak üzere PID kontrolördeki P, I ve D katsayılarının dikkatli bir şekilde seçilmesi gerektiği görülmektedir.

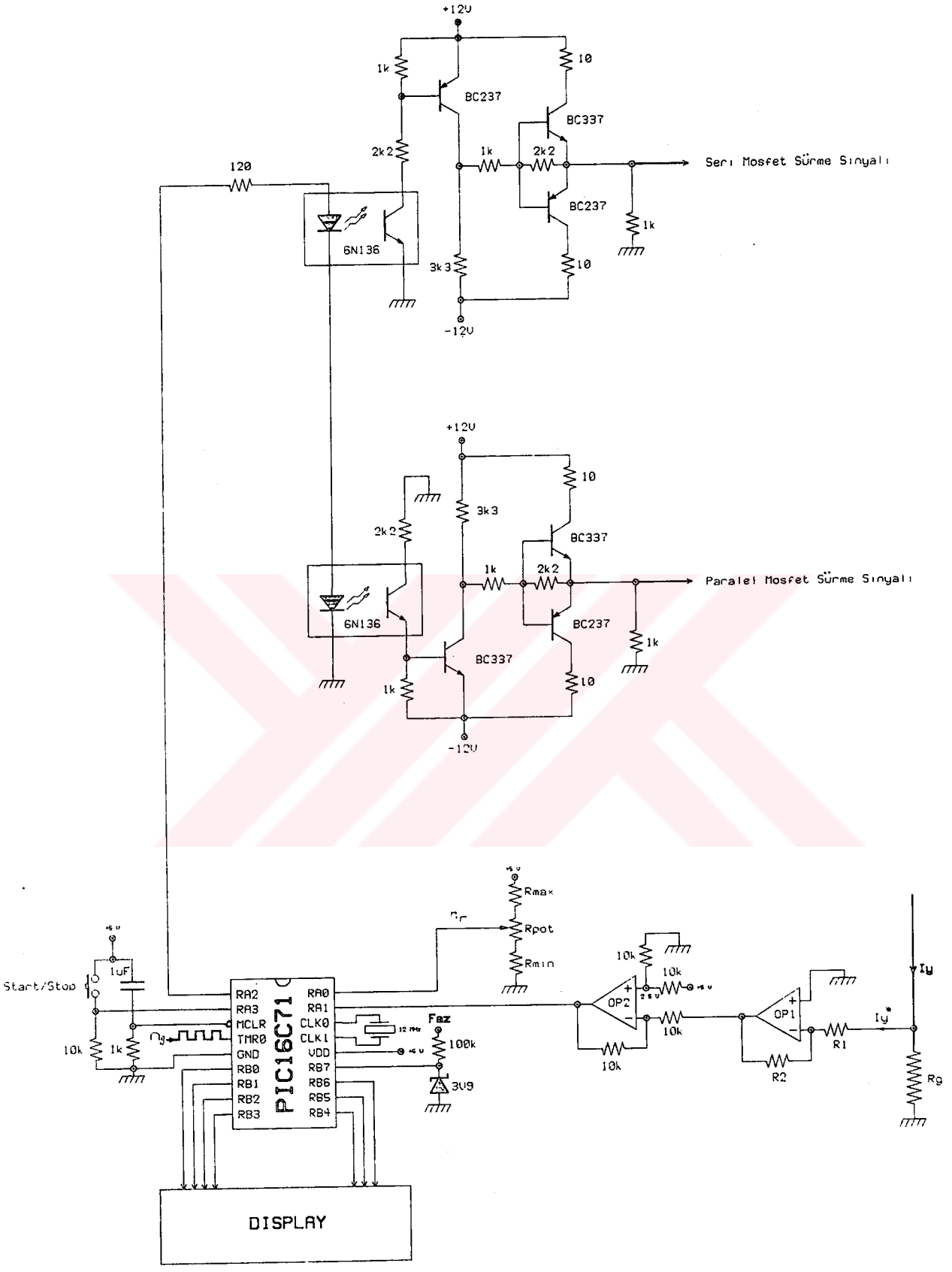
4.2. UYGULAMA DEVRESİ

Laboratuvarda gerçekleştirilen uygulama devresinin blok şeması Şekil 4.2'de görülmektedir. PWM ac kıyıcı devrelerinde iki grup vardır. Birinci grup yüke verilen gücü kontrol eder. İkinci grup yük akımının sürekliliğini sağlar. Yük akımı yön değiştirdiğinden, her grup için pozitif alternansta ve negatif alternansta akımı geçirecek ters paralel bağlı iki kontrollu eleman kullanmak gerekir. Bu durumda, MOSFET kullanılarak yapılan tek fazlı PWM ac kıyıcıda dört adet mosfet kullanmak gerekecektir. Herbir mosfet için +Vcc, -Vcc ve GND çıkışı veren bağımsız sürme kaynağı ve sürme devresi gerektiğine göre sadece mosfetler için dört adet trafo ve dört adet sürme devresi kullanmak gerekecektir. Eleman sayısını düşürmek ve kontrolu daha sade bir şekilde yapabilmek için ters paralel bağlı iki mosfet yerine bir diyot köprüsü ile bir adet mosfet kullanılmıştır. Bu durumda devrede kullanılan toplam mosfet sayısı ikiye düşmekte dolayısıyla iki adet bağımsız sürme kaynağı ve iki adet sürme devresi kullanılmaktadır. Kontrol devresi için de +Vcc, -Vcc ve GND çıkışı veren bir sürme kaynağına ihtiyaç duyulmuştur. Kontrol devresinin GND seviyesi şebekenin nötrüne bağlanarak yük akımını opto eleman kullanmadan doğrudan okuma imkanı sağlanmıştır.

Kontrol devresine, istenilen n_r referans hızı, kontrol edilen büyüklükler; $I_{yük}$ motor akımı örneği ve n_g motor geribesleme hızı ve mosfetlerin anahtarlama anlarını şebekeye uygun olarak seçebilmek için şebeke senkronizasyon işareti girilmektedir. Kontrol devresinin çıkışları hangi kolun iletime gireceğini belirleyen işaretleri mosfetlerin sürme devrelerine göndermektedir. Sürme devrelerine gönderilen sürme işaretleri her iki mosfetin aynı anda iletimde kalmasına sebeb olmamalıdır. Bunun için kilitlemeli sürme düzenleri kullanılmaktadır. Motor milinden hız geribeslemesi almak için, üzerinde 40 adet yarık bulunan optik kodlayıcı kullanılmıştır. Kodlayıcının bir tarafında ışık yayan bir diyot, diğer tarafında opto transistör bulunmaktadır. Opto transistörden elde edilen çıkış, transistörlü bir devre ile keskinleştirilerek kare dalga olarak kontrol devresine verilmiştir. Belli bir süre içindeki kare dalgalar sayılarak n_g geribesleme hız bilgisi elde edilmektedir. Uygulama devresine ait şema Şekil 4.3'te verilmiştir.



Şekil 4.2. Uygulama devresinin blok şeması.



Şekil 4.3. Uygulama devresi.

Akım kontrollu PWM ac kıyıcı ile gerçekleştirilen asenkron motor hız kontrolu devresinde, kontrol PIC16C71 denetleyicisi tarafından yapılmaktadır. Mikrodenetleyiciye 12 MHz'lik kristal bağlanmıştır. Bir komutun yürütülmesi $T = 4/12^6 = 333$ ns ' de tamamlanmaktadır. $\overline{MCLR}$ bacağına sıfır seviyesinin gelmesi ile mikrodenetleyici reset vektöründen yani program belleğinin 00h adresinden itibaren çalışmaya başlamaktadır. Kesme geldiğinde mikrodenetleyicinin ilgili kesme hizmet programına dallanabilmesi için, 04h kesme adresine gerekli dallanma komutu yerleştirilmiştir.

Mikrodenetleyicinin RA0 ve RA1 bacağı analog giriş, RA2 bacağı dijital çıkış, RA3 bacağı dijital giriş olarak tanımlanmıştır. Mikrodenetleyicinin RB7 bacağı herbir değişimde kesme algılayacak şekilde giriş olarak tanımlanmıştır. RB0-RB6 gösterge sürmek üzere çıkış olarak tanımlanmıştır.

Referans hız RA0 bacağından, yük akımı RA1 bacağından okunmakta ve ADC ile dijital sayılara dönüştürülmektedir. Yük akımının Rg direncinin üzerinde oluşturduğu gerilim uygun bir katsayı ile çarpılarak ve 2.5 V ile toplanarak RA1 bacağına verilmiştir. Burada hedeflenen, yükten maksimum akım geçtiğinde RA1 uçlarındaki gerilimin 0 ile 5 V arasında kalması ve bu gerilimin ekseninin 2.5 V olmasıdır. RA3 bacağına bağlı Start/Stop tuşuna basıldığında sistem çalıştırılmakta, tekrar basıldığında durdurulmaktadır. RA2 bacağından verilen sürme sinyali seri ve paralel mosfetlere ait sürme devrelerindeki optik elemanların girişindeki diyotlara verilmektedir. Diyotlar seri olarak bağlanarak bir koldaki mosfet iletime girerken diğer koldaki mosfetin kesime gitmesi sağlanmıştır. Böylece her iki elemanın aynı anda iletimde kalma riski ortadan kaldırılmıştır. Mosfetin hızlı iletime girmesi ve iletimden çıkışının hızlı olması için ±12V' luk kaynak kullanılmış ve mosfet sürme akımı transistörlerle yükseltilmiştir. RA2 bacağından verilebilecek sürme akımı maksimum 20 mA'dir. Bu akım sürme devrelerinin girişindeki diyotları doğrudan sürmek için yeterli olmaktadır. RB7 bacağına şebeke ile senkronizasyonu sağlayan bir işaret verilmiştir.

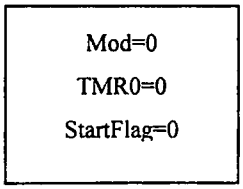
4.3. PIC16C71 MİKRODENETLEYİCİSİ İÇİN YAZILAN PROGRAM

Programın akış diyagramı Şekil 4.4'te verilmiştir. Programın çalışması şu şekildedir. Her 10 ms'de bir TMR0 ile motor hızı okunmaktadır. ROM'da saklanan sinüs tablosu Iref olarak kullanılmaktadır. 256 farklı sinüs referansı, Iref'in Mod değişkeninde tutulan bir sayı ile çarpılmasıyla elde edilmektedir. Başlangıçta Mod ve TMR0 sıfır değerindedir. Start tuşuna ait StartFlag bayrağı da sıfırdır. Şebekenin her sıfır noktasında bir kesme oluşturularak RB7 bacağından şebeke bilgisi alınmaktadır.

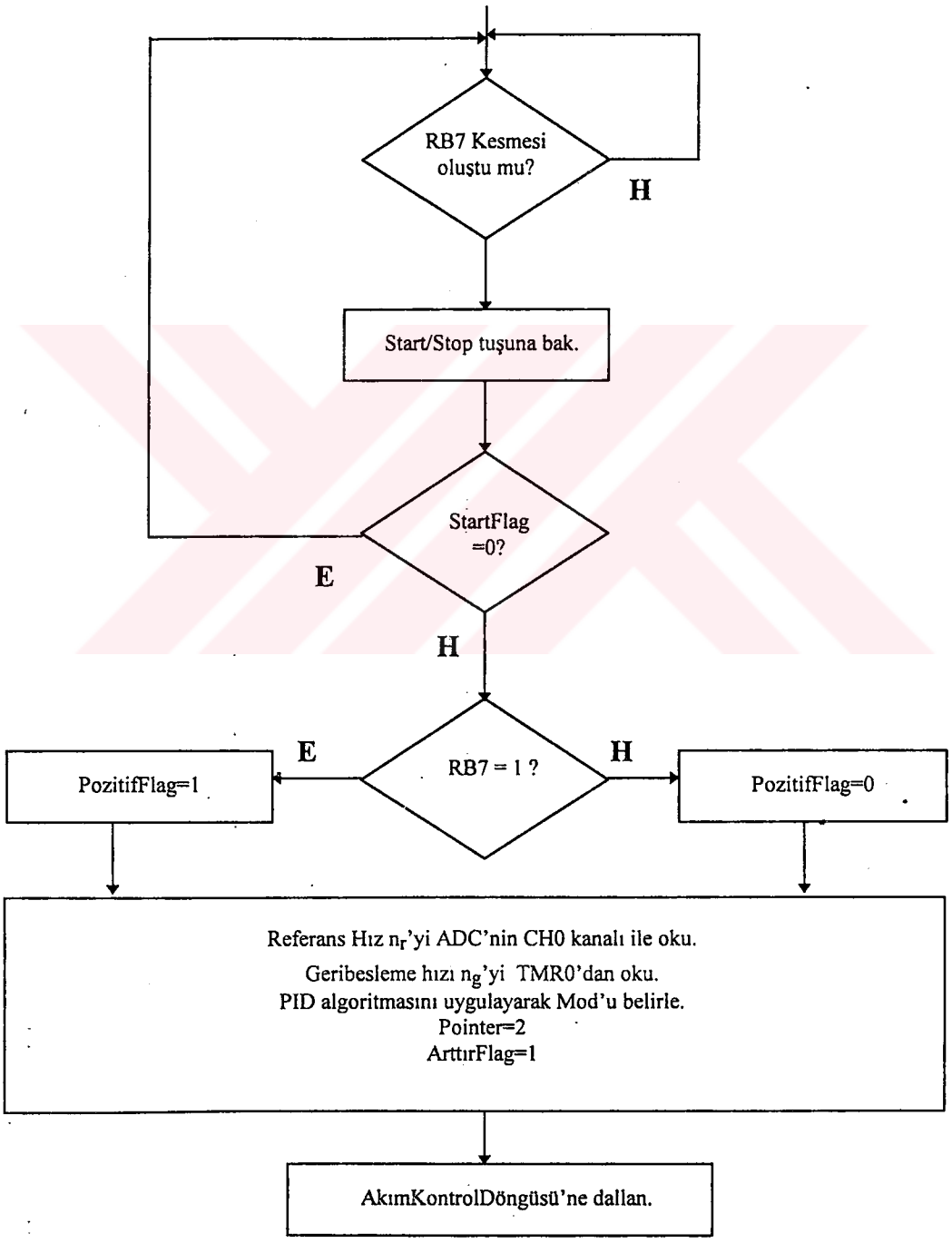
Start tuşuna basıldığında şebekenin sıfır noktasında program başlayacaktır. Pointer değişkeni şebekenin sıfır noktasında 0 ile yüklenmektedir. PozitifFlag şebekenin pozitif alternansında 1, negatif alternansta 0 yapılır. İlgili referans hız ADC'nin CH0 kanalından alınarak TMR0 ile okunan geribesleme hızı ile karşılaştırılır ve hataya göre PID algoritması ile uygun Mod değeri belirlenir. Mod değişkeninin dolayısıyla Iref referans akımının belli değerler arasında kalması için alt ve üst limitler konulmuştur. Hız karşılaştırılması ve PID algoritmasının Mod değerini belirlemesi şebekenin sıfır noktasından itibaren 200µs sürmektedir. Bundan sonra program akım kontrol döngüsüne girmektedir. Yük akımı her 100µs'de A/D dönüştürücü ile okunarak Iyük değişkenine yüklenmektedir. Her bir örnekleme sonunda Pointer değişkeni bir attırılarak sinüs tablosundan ilgili sinüs değeri alınmaktadır. Sinüs tablosundan alınan değer Pointer ve Mod tarafından belirlenir. Pointer k. sinüs örneğini, Mod ise sinüs referansının genliğini belirlemektedir. Yük akımı her 100µs'de bir örneklendiğine göre bir peryotta 200 adet sinüs örneği kullanmak gerekecektir. 128 farklı sinüs referansını saklamak üzere her bir örnek için bir byte kullanılırsa 128x200=25.600 byte kullanmak gerekecektir. Kullanılan mikrodenetleyici ise 1024 byte'lık bellek içermektedir. Dışarıdan bellek bağlanmadan bu problemin çözülebilmesi için sinüs referansının sadece en büyük değeri 0 ile $\pi/2$ arasında depo edilmiş diğer bütün değerler bu tablodan elde edilmiştir. Örneklenen yük akımı negatif değerler almaması gerektiğinden 2.5 V ile toplanmıştır. Pozitif alternansta yük akımı 2.5 V'un üzerinde olduğunda referans akım Ymax/2 değeri ile toplanarak elde edilmektedir. Negatif alternansta ise Ymax/2 - Iref

değeri referansa atanmaktadır. Yük akımının seçilen histerezis bantının içerisinde kalabilmesi için pozitif ve negatif alternansta tetiklenmesi gereken mosfetlerin belirlenmesi akış diyagramında gösterilen P_Kontrol ve N_Kontrol altprogramları ile sağlanmıştır. Seçilen sinüs referansının IH/2 kadar altına düşülmesi veya IH/2 kadar üstüne çıkılması durumunda anahtarlar konum değiştirmektedir. Bu durum pozitif ve negatif alternanslarda farklı karşılaştırmalar ile sağlanmıştır. Pozitif alternansta akım artmakta iken referans akımdan IH/2 kadar büyük olması durumunda SeriMOS 0 yapılır. PORTA'nın 2. biti olarak tanımlanmış olan SeriMOS 1 yapıldığında seri mosfet iletime girerek paralel mosfet kesime gitmekte, 0 yapıldığında seri mosfet kesime giderek paralel mosfet iletime girmektedir. Pozitif alternansta akım azalmakta iken referans akımdan IH/2 kadar küçük olması durumunda SeriMOS 1 yapılır. Negatif alternansta ise akım negatif yönde artmakta iken referans akımın IH/2 kadar altına düşerse SeriMOS 0 yapılır. Negatif alternansta akım azalmakta iken, yani eksene yaklaşmakta iken referans akımın IH/2 kadar üstüne çıkarsa SeriMOS 1 yapılmaktadır. Pointer değişkeni 100 değerini aştığında sıfır noktasına yaklaşılmış olduğu anlaşılarak kesme hizmet programına gidilmekte ve şebekenin sıfır noktasının başlaması beklenmektedir. Stop tuşuna basılmadıysa her sıfır noktasının başlangıcında ArttırFlag 1 yapılarak SeriMOS'a 1 atanmaktadır.

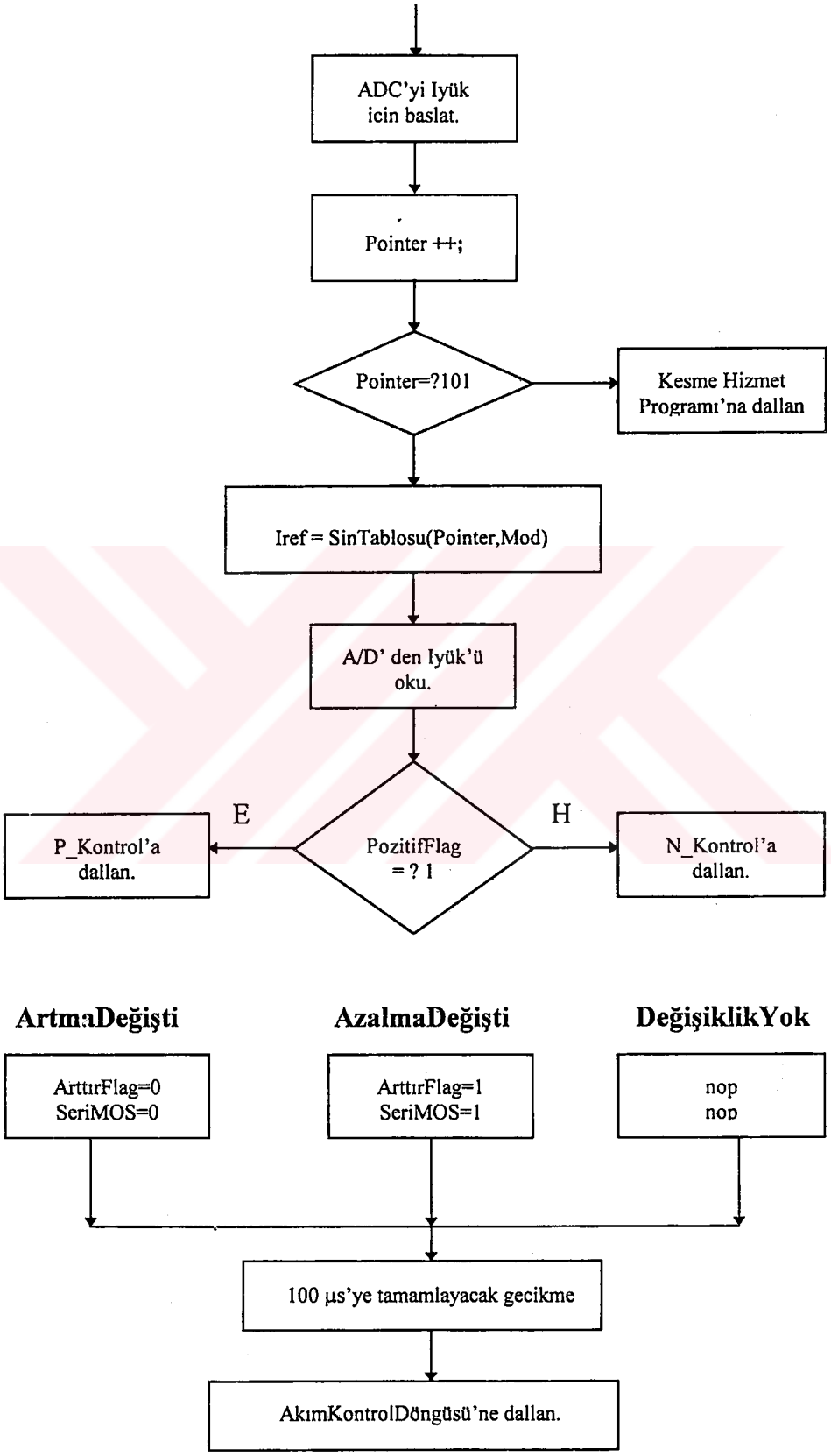
Başlangıç Değerlerinin Atanması

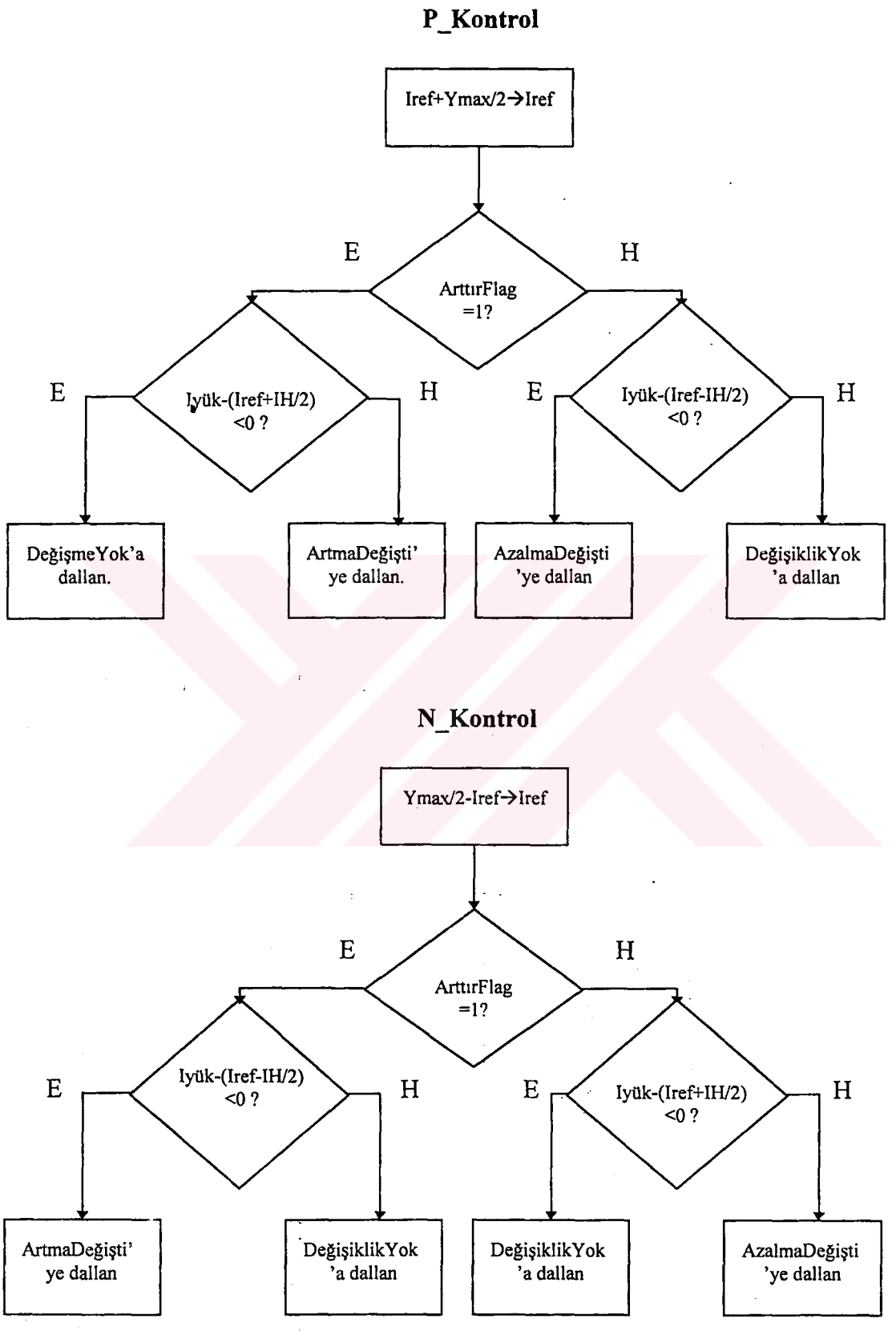


Kesme Hizmet Programı



Akım Kontrol Döngüsü





Şekil 4.4. PIC16C71 için geliştirilen programın akış diyagramı.

PIC16C71 mikrodenetleyicisi için yazılan program aşağıda verilmiştir.

```
; Akım Kontrollu PWM ile Motor Kontrolu. Copyright A. Faruk Bakan 1997
; Bu programda hız geribeslemesi alınarak ve motor akımına göre PWM gerilimi
; üretilerek motor kontrol edilmiştir. Referans hız potansiyometre ile ADC'nin CH0
; kanalı ile, motor akımı her 100 µs'de bir CH1 kanalı ile ölçülmüştür. Motorun hızı
; her 10 ms'de bir TMR0 ile okunmaktadır.

; RA0 (CH0) ← Referans hız (0-255)
; RA1 (CH1) ← Iyük (0-255)
; RA2 → SeriMOS
; RA3 ← Start/Stop tuşu


        LIST P=16C71, F=INHX8M

Flags       equ   12h       ; Bayrakları tutuyor.

Iyük        equ   14h       ; ADC ile ölçülüyor.
Iref        equ   15h       ;  tabloda tutulan sinüs referansı
Mod         equ   16h       ; 256 farklı Iref değerini belirler.
Counter     equ   17h       ;  hız okumak için kullanılıyor.
Speed       equ   18h       ;  referans hız .
Pointer     equ   19h       ; tablodan sinüs örneklerini almak üzere tutuluyor.
temp        equ   1Fh       ; geçici hesaplamalar için kullanılıyor.
hata        equ   1Ah
OranTer     equ   1Bh       ; OranTer=Hata/OranKat
OranKat                     equ   1Ch
IntTop      equ   1Dh       ; IntTop=IntTop+hata
IntTer      equ   1Eh       ; IntTer=IntTop/IntKat
IntKat      equ   1Fh
AccA        equ   20h       ; 8 bit işaretli aritmetik için kullanılyor.
AccB        equ   21h       ; 8 bit işaretli aritmetik için kullanılyor.
mulcnd      equ   22h       ; 8 bit multiplicand
mulplr      equ   23h       ; 8 bit multiplier
H_byte      equ   24h       ; High byte of the 16 bit result
L_byte      equ   25h       ; Low byte of the 16 bit result
count       equ   26h       ; loop counter
Same                        equ   1

StartFlag   equ   7 ; Flags<7>
OldStartValue equ 6 ; Flags<6>
PozitifFlag equ   5 ; Flags<5>
ArttırFlag  equ   4 ; Flags<4>   ; Arttır = 1 ise Seri_MOS=1
```

```
SeriMOS     equ   2 ; PORTA<2>
Start       equ   3 ; PORTA<3>

IH          equ   .10  ; histerizis bandı IH = 20
YMAX        equ   .255
SpeedH      equ   .1   ; Hızdaki sapma.

            include"p16c71.inc"

            org   0
            goto  Initialization

            org   4
            goto  ServiceInt

;*****************************************************************
;                        Başlangıç Değerlerinin Atanması
;*****************************************************************

Initialization

InitCounter bsf   STATUS,RP0          ; bölüm 1 seçiliyor.
            bsf   OPTION_REG,T0CS     ; 1 TMR0 saat kaynağı harici
            bsf   OPTION_REG,T0SE     ; TMR0 düşen kenarında artıyor.
            bsf   OPTION_REG,PSA      ; 1 PSA is WDT'ye atanıyor.
            bcf   INTCON,T0IE         ; TMR0 kesmesini iptal et.

InitPortA   bsf   STATUS,RP0          ; bölüm 1 seçiliyor.
            movlw B'00000010'         ; RA0-RA1 Analog giriş olarak seçiliyor.
            movwf ADCON1              ; RA2-RA3 Dijital Çıkış olarak seçiliyor.
            movlw B'11111011'         ; PORTA<0,1,3> giriş
            movwf TRISA               ; PORTA<2> çıkış
            bcf   STATUS,RP0          ; bölüm 0'ı seçiliyor.

InitPortB   bsf   STATUS,RP0          ;Bölüm 1'i seç.
            movlw B'10000000'         ;RB7 giriş. Diğerleri çıkış
            movwf TRISB               ;
            bcf   OPTION_REG,NOT_RBPU ; pull up aktif yapılıyor.
            bcf   STATUS,RP0          ; Bölüm 0'ı seç.
            clrf  PORTB               ; PORTB'yi sıfırla
            bcf   INTCON,RBIE         ; kesmeyi iptal et.
            movf  PORTB,w             ; PortB'yi oku.
            bcf   INTCON,RBIF         ; bayrağı temizle.
            bsf   INTCON,RBIE         ; kesmeyi mümkün kıl.
            bsf   INTCON,GIE          ; global bitini mümkün kıl.
```

```
InitFSR     movlw  .0          ; Değişkenlere ilk değerleri atanıyor.
            movwf  Mod
            clrf   Counter
            clrf   TMR0
            bcf    Flags,StartFlag
            bcf    Flags,OldStartValue

;*****************************************************************************
;                       Service Interrupt
;*****************************************************************************

ServiceInt  bcf    PORTA,SeriMOS   ; SeriMOSFET kesimde
            btfss  INTCON,RBIF     ; RB değişmesi ile kesme oluştu mu?
            goto   ServiceInt      ; alternansın başında başlamak için
                                   ; kesme gelene kadar bekle.

ServiceRB   movf   PORTB,w         ;portu oku.
            bcf    INTCON,RBIF     ;bayrağı temizle.
            andlw  B'10000000'     ;RB7'yi w'ya al.
            btfss  STATUS,Z
            goto   Set_Pozitif     ; w=1 (Z=0) , PozitifFlag=1
            goto   Clr_Pozitif     ; w=0 (Z=1) , PozitifFlag=0
Set_Pozitif bsf    Flags,PozitifFlag
            goto   RefHızOku
Clr_Pozitif bcf    Flags,PozitifFlag
            goto   RefHızOku

RefHızOku   ; Referans hız ADC'nin CH1 kanalı ile okunuyor. (0-255)
            movlw  B'00001001'     ; Fosc/2 , ch1  , ADON=1
            movwf  ADCON0          ; turn on A/D
            bsf    ADCON0,2        ; GO=1
ADCHızLoop  btfsc  ADCON0,2        ; DONE=0 mı?
            goto   ADCHızLoop
            movf   ADRES,w         ; w=ADC_result
            movwf  Speed           ; Referans hız alındı.

; TMR0 her 10 ms'de 1 kontrol ediliyor.
; Motor max. devirde 10 ms'de 140 darbe üretiyor.

            movf   TMR0,w          ; Gerçek Hız TMR0'dan okunuyor.
            movwf  temp            ; gerçek hız temp'de tutuluyor.
            clrf   TMR0
```

```
HataHes     movf   nger,w       ; w = nref-nger
            subwf  nref,w       ; Çıkarma sonucu negatif ise C=0
            btfsc  STATUS,C     ; >= 0 ise C=1 olur.
            goto   HataPoz
            goto   HataNeg

HataPoz     movwf  hata         ; hata = nref - nger
            bcf    Flags,HataFlag ; hata pozitif. Bayrak 0
            goto   Exit_HataHes  ; motor istenilen hıza ulaşmamış.

HataNeg     movf   nref,w       ; hata negatif.
            subwf  nger,w       ; motor hızı istenilen hızdan büyük.
            movwf  hata         ; hata = nger-nref
            bsf    Flags,HataFlag ; Hata bayrağı negatif için 1
Exit_HataHes

            movlw  HIGH PIDHesabı
            movwf  PCLATH
            goto   PIDHesabı

ModTamam    nop

HızTamam    nop

StartKontrol  movf   PORTA,w
            movwf  temp         ; PORTA--->temp'e yükleniyor.
            btfss  temp,Start   ; temp<3> (PORTA<3>)  = 1 ise basıldı.
            goto   StartNotPressed
            btfsc  Flags,OldStartValue
            goto   CheckStartFlag ; Daha önceden basılı tutulmuş.
            bsf    Flags,OldStartValue
            btfss  Flags,StartFlag
            goto   SetStartFlag
            goto   ClrStartFlag
SetStartFlag  bsf    Flags,StartFlag
            goto   CheckStartFlag
ClrStartFlag  bcf    Flags,StartFlag
            goto   CheckStartFlag
StartNotPressed
            bcf    Flags,OldStartValue
CheckStartFlag
            btfss  Flags,StartFlag      ; StartFlag = 0 ise SeriMosfet'i
            goto   ServiceInt           ; iletime sokma, ServiceInt'e git.
```

```
MOSFET_Settings
            bsf     Flags,ArttırFlag     ; Arttır = 1
            bsf     PORTA,SeriMOS        ; Seri_Mosfeti iletime geçir.

            clrf    Pointer
            bsf     INTCON,GIE

            movlw   .20          ; 100 mikro sn'ye tamamlayacak kadar gecikme.
            movwf   temp
Delay100Tamam
            decfsz  temp
            goto    Delay100Tamam
            goto    Current_Loop

;*****************************************************************************
;                   Akım_Döngüsü
;*****************************************************************************

Current_Loop
            movlw   B'00000001' ; Fosc/2 , ch0  , ADON=1
            movwf   ADCON0      ; turn on A/D
            bsf     ADCON0,2    ; GO=1 Iyük'ün okunması için ADC başlatılıyor.

            incf    Pointer,f
            movf    Pointer,w
            xorlw   .101
            btfsc   STATUS,Z    ; Pointer = 101 olduysa ServiceInt'e git.
            goto    ServiceInt  ; Şebeke sıfır olana kadar bekle.


            ; ***** Iref(Pointer,Mod) alınıyor. *****
            goto    Sin_Table_Pointer_Mod


Kontrol     movf    ADRES,w     ; w=ADC_RESULT
            movwf   Iyük        ; Iyük alınıyor..

            btfss   Flags,PozitifFlag
            goto    N_Kontrol
            goto    P_Kontrol

P_Kontrol   movlw   YMAX/2
            addwf   Iref        ; Iref = Iref + 255/2
            nop
            nop
```

```
                nop
                nop
                btfss   Flags,ArttırFlag
                goto    P_Azalma
                goto    P_Artma
P_Artma         movlw   IH/2
                addwf   Iref,w
                subwf   Iyük            ; Iyük - (Iref+IH/2) >= 0 ise C=1 değişti.
                                        ;                    < 0    C=0 değişiklik _yok
                btfss   STATUS,C
                goto    No_Change
                goto    Artma_Change
P_Azalma        movlw   IH/2
                subwf   Iref,w
                subwf   Iyük            ; Iyük - (Iref-IH/2) <  0 ise C=0 değişti.
                                        ;                    >= 0    C=1 değişiklik _yok
                btfss   STATUS,C
                goto    Azalma_Change
                goto    No_Change


N_Kontrol       movf    Iref,w
                movwf   temp
                movlw   YMAX/2
                movwf   Iref
                movf    temp,w
                subwf   Iref            ; Iref = 255/2 - Iref
                btfss   Flags,ArttırFlag
                goto    N_Azalma
                goto    N_Artma
N_Artma         movlw   IH/2
                subwf   Iref,w
                subwf   Iyük            ; Iyük - (Iref-IH/2) >= 0 ise C=1 değişiklik yok.
                                        ;                    < 0    C=0    değişti.
                btfss   STATUS,C
                goto    Artma_Change
                goto    No_Change
N_Azalma        movlw   IH/2
                addwf   Iref,w
                subwf   Iyük            ; Iyük - (Iref-IH/2) <  0 ise C=0 değişiklik yok
                                        ;                    >= 0    C=1 değişti.
                btfss   STATUS,C
                goto    No_Change
                goto    Azalma_Change
```

```
Artma_Change
            bcf     Flags,ArttırFlag
            bcf     PORTA,SeriMOS
            goto    Delay
Azalma_Change
            bsf     Flags,ArttırFlag
            bsf     PORTA,SeriMOS
            goto    Delay
No_Change   nop
            nop
            nop
            goto    Delay

Delay       movlw   .10   ; 100 mikro sn'ye tamamlayacak kadar gecikme.
            movwf   temp
Delay1      decfsz  temp
            goto    Delay1

Tamam       goto    Current_Loop

;*****************************************************************************
;                   Iref(Pointer,Mod) alınarak
;                   Kontrol'a geri dönülüyor.
;*****************************************************************************

            org   200h

Sin_Table_Pointer_Mod
            movlw   .50
            subwf    Pointer,w     ; Pointer-50 >= 0 ise C=1
            btfss    STATUS,C
            goto     Table_Pointer
            movlw   .100
            movwf   temp
            movf    Pointer,w
            subwf   temp,f        ; temp = 100-Pointer
            goto    Pointer_OK
Table_Pointer
            movf    Pointer,w     ; temp = Pointer
            movwf   temp
Pointer_OK  movf    temp,w

            call    Sin_256_Table
            movwf   mulcnd        ; Sin_Table(k)--->Sin_256'ya (mulcnd) yükleniyor.
            movf    Mod,w
```

```
            movwf  mulplr

mpy_S       clrf    H_byte
            clrf    L_byte
            movlw   8
            movwf   count
            movf    mulcnd,w
            bcf     STATUS,C    ; Clear the carry bit in the status Reg.
Mloop       rrf     mulplr
            btfsc   STATUS,C
            addwf   H_byte,Same
            rrf     H_byte,Same
            rrf     L_byte,Same
            decfsz  count
            goto    Mloop       ; Multiplier loop

            bcf     STATUS,C    ; 1   /2
            rrf     H_byte
            rrf     L_byte
            bcf     STATUS,C    ; 2   /4
            rrf     H_byte
            rrf     L_byte
            bcf     STATUS,C    ; 3   /8
            rrf     H_byte
            rrf     L_byte
            bcf     STATUS,C    ; 4   /16
            rrf     H_byte
            rrf     L_byte
            bcf     STATUS,C    ; 5   /32
            rrf     H_byte
            rrf     L_byte
            bcf     STATUS,C    ; 6   /64
            rrf     H_byte
            rrf     L_byte
            bcf     STATUS,C    ; 7   /128
            rrf     H_byte
            rrf     L_byte,w
            bcf     STATUS,C    ; 8   /256
            rrf     H_byte
            rrf     L_byte,w
            movwf   Iref

            goto    Kontrol
```

```
Sin_256_Table
        addwf  PCL
        retlw 0
        retlw .4
        retlw .8
        retlw .11
        retlw .15
        retlw .19
        retlw .23
        retlw .27
        retlw .31
        retlw .35
        retlw .39
        retlw .43
        retlw .46
        retlw .50
        retlw .54
        retlw .57
        retlw .61
        retlw .64
        retlw .68
        retlw .71
        retlw .74
        retlw .78
        retlw .81
        retlw .84
        retlw .87
        retlw .90
        retlw .92
        retlw .95
        retlw .98
        retlw .100
        retlw .103
        retlw .105
        retlw .107
        retlw .109
        retlw .111
        retlw .113
        retlw .115
        retlw .117
        retlw .118
        retlw .119
        retlw .121
        retlw .122
        retlw .123
```

```
                retlw .124
                retlw .125
                retlw .125
                retlw .126
                retlw .126
                retlw .127
                retlw .127
                retlw .127

;**************** P I ve D terimlerinin hesaplanarak Mod'un bulunması****

PIDHesabı       org    300h

OranHes         movf   OranKat,w
                movwf  temp
                movf   hata,w
                movwf  OranTer
OranHesLoop     bcf    STATUS,C          ; 2
                rrf    OranTer           ; hatayı ikiye bölüyoruz.
                decfsz temp
                goto   OranHesLoop       ; OranTer = Hata / 2 ^ OranKatsayısı
                                         ; OranTerimi + veya - olabilir.

;*************** Integral terimi hesabı **********************

                clrf   temp
                btfsc  Flags,HataFlag  ; Sayı0'a ait işaret. O ise sayı pozitif.
                bsf    temp,0          ; temp'e alınıyor.
                btfsc  Flags,IntTopFlag ; Sayı1'e ait işaret. O ise sayı pozitif.
                bsf    temp,1  ; temp'e alınıyor.
                movf   hata,w ;
                movwf  AccB    ; Sayı0-->AccB'ye yükleniyor
                movf   IntTop,w ;
                movwf  AccA    ; Sayı1-->AccA'ya yükleniyor.

                call   S_Toplama

                bcf    Flags,IntTopFlag
                btfsc  temp,2
                bsf    Flags,IntTopFlag
                movf   AccA,w
                movwf  IntTop

                goto   IntHes
```

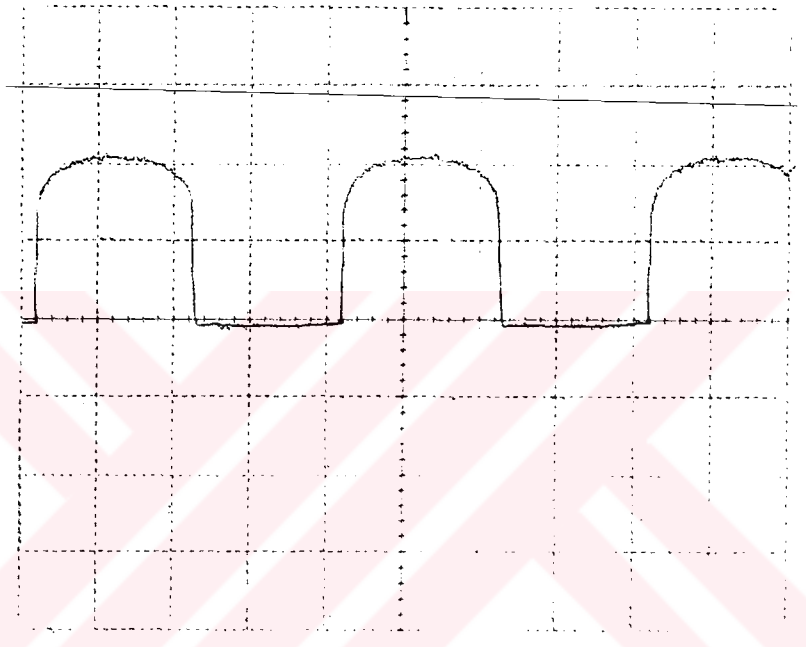
```
S_Toplama
;*****************************************************************
; temp<1>=AccAflag, temp<0>=AccBflag
; Sayı0+Sayı1-->Sayı1,   AccB+AccA-->AccB, işaret temp<2>'ten alınacak.

              movf   temp,w
              addwf  PCL
              goto   PP     ; 00 PP
              goto   PN     ; 01 PN
              goto   NP     ; 10 NP
              goto   NN     ; 11 NN

;************** AccA (+), AccB (-)

PN          movf   AccB,w
              subwf  AccA,w
              btfsc  STATUS,C        ; çıkarma sonucu negatifse C=0 olur.
              goto   AccAP_B_AccBN    ; AccA>AccBN
              goto   AccAP_K_AccBN    ; AccBN>AccA

AccAP_B_AccBN  movf   AccB,w
              subwf  AccA        ; AccA = AccA - AccB
              bcf    temp,2
              return

AccAP_K_AccBN  movf   AccA,w
              subwf  AccB,w
              movwf  AccA        ; AccA =  AccB -AccA
              bsf    temp,2
              return

;************** AccA (-), AccB (+)

NP          movf   AccB,w
              subwf  AccA,w
              btfsc  STATUS,C
              goto   AccAN_B_AccBP    ; AccA>AccBP
              goto   AccAN_K_AccBP    ; AccBP>AccA

AccAN_B_AccBP  movf   AccB,w
              subwf  AccA        ; AccA = AccA - AccB
              bsf    temp,2
              return
```

```
AccAN_K_AccBP  movf   AccA,w
               subwf  AccB,w
               movwf  AccA       ; AccA =  AccB -AccA
               bcf    temp,2
               return

NN             bsf    temp,2
               goto   PPNN
PP             bcf    temp,2
               goto   PPNN
PPNN           movf   AccB,w
               addwf  AccA
               btfss  STATUS,C      ; Taşma varsa 255 ile sınırlandır.
               return
               movlw  .255
               movwf  AccA
               return

;*********************************************************************

IntHes         movf   IntKat,w
               movwf  temp
               movf   IntTop,w
               movwf  IntTer
IntHesLoop     bcf    STATUS,C      ; 2
               rrf    IntTer        ; IntTop'ı ikiye bölüyoruz.
               decfsz temp
               goto   IntHesLoop    ; IntTer = IntTop / 2 ^ IntegKatsayısı
                                    ; IntTerimi + veya - olabilir. Hata ile aynı işarette.

;*********************************************************************
; P+I-->P'ye  yazılıyor.

               clrf   temp
               btfsc  Flags,IntTopFlag ; IntTer'e ait işaret.
               bsf    temp,0        ; temp'e alınıyor.
               btfsc  Flags,HataFlag  ; OranTer'e ait işaret.
               bsf    temp,1  ; temp'e alınıyor.
               movf   IntTer,w ;
               movwf  AccB    ; Sayı0-->AccB'ye yükleniyor
               movf   OranTer,w ;
               movwf  AccA    ; Sayı1-->AccA'ya yükleniyor.

               call   S_Toplama
```

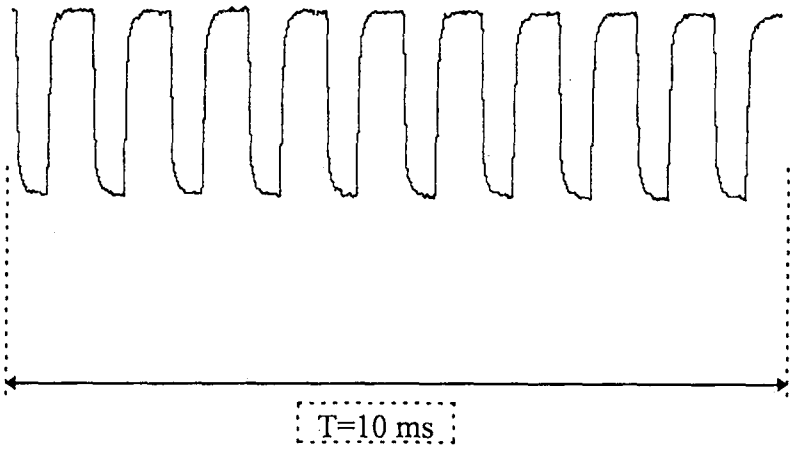
```
            bcf   Flags,HataFlag
            btfsc  temp,2
            bsf   Flags,HataFlag
            movf   AccA,w
            movwf  OranTer              ; (+/-)OranTer = (+/-)OranTer + (+/-)IntTer
                                        ;HataFlag      HataFlag     IntTopFlag


;*************** Mod Hesabı ***********************************************

            btfsc  Flags,HataFlag  ; OranTer'e ait işaret pozitif ise atla.
            goto   OranTerN_ModP   ; Oran Terimi Negatif, Mod Pozitif

            movf   OranTer,w
            addwf  Mod
            btfss  STATUS,C        ; Taşma varsa 255 ile sınırlandır.
            goto   Exit_PIDHesabı
            movlw  .255
            movwf  Mod
            goto   Exit_PIDHesabı

OranTerN_ModP  movf   OranTer,w
            subwf  Mod,w           ; Mod = Mod-OranTer
            btfsc  STATUS,C        ; çıkarma sonucu negatifse C=0 olur.
            goto   ModP_B_OranTerN ; Mod>OranTer
            movlw  .1
            movwf  Mod
            goto   Exit_PIDHesabı

ModP_B_OranTerN
            movf   OranTer,w
            subwf  Mod         ; Mod = Mod - OranTer
            goto   Exit_PIDHesabı

Exit_PIDHesabı
            movlw  HIGH ModTamam
            movwf  PCLATH
            goto   ModTamam

END
```

4.4. UYGULAMA DEVRESİNDEN ALINAN SONUÇLAR

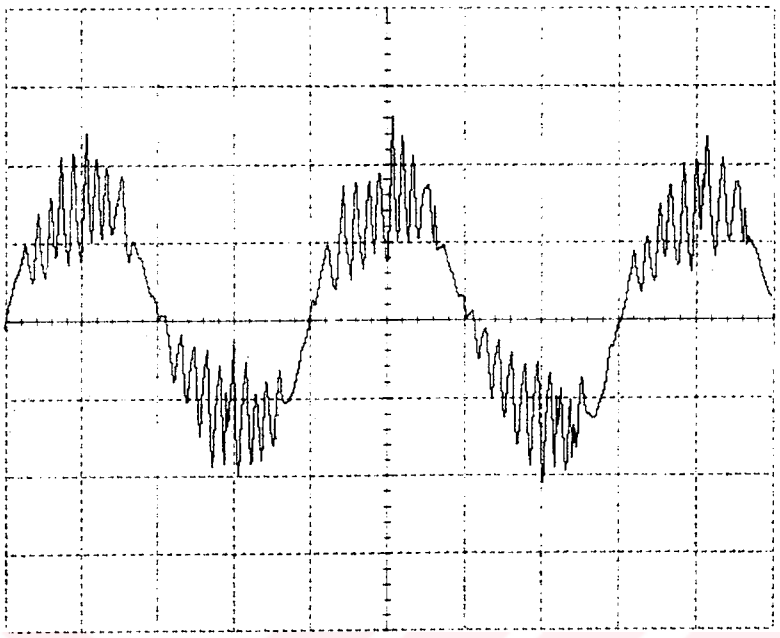
Uygulama devresinden alınan sonuçlar aşağıda verilmiştir. Şebeke ile senkronizasyon amacıyla mikrodenetleyicinin RB7 bacağına verilen işaretin değişimi Şekil 4.5'te verilmiştir. Mikrodenetleyici'nin TMR0 girişine verilen geribesleme hız bilgisine ait değişim Şekil 4.6'da verilmiştir. Optik kodlayıcının 40 adet yarığı mevcuttur. Dolayısıyla motor bir devirde 40 adet darbe üretmektedir. Şekil 4.6'da 10 ms'de 10 darbe mevcut olduğuna göre geribesleme hızı n_g=1500 d/d' dır. Bu devir için motordan geçen akımın değişimi Şekil 4.7'de, motor gerilimi Şekil 4.8'de verilmektedir.



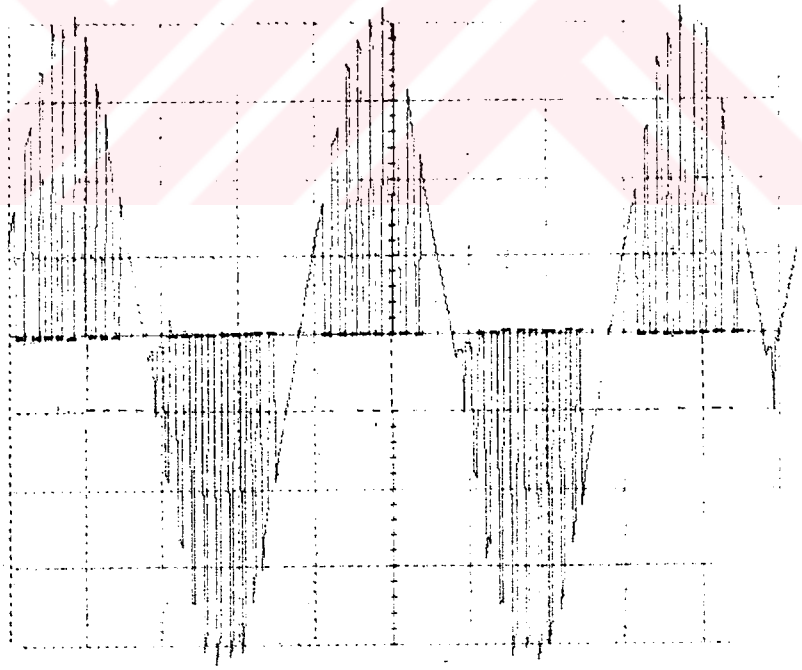
Şekil 4.5. Mikrodenetleyicinin RB7 bacağına verilen şebeke senkronizasyon işareti.



Şekil 4.6. Motor hızı 1500 d/d olduğunda motor milinden alınan geribesleme işareti.



Şekil 4.7. Motor akımının değişimi.



Şekil 4.8. Motor'a uygulanan geriliminin değişimi.

SONUÇLAR

AC kıyıcılarda, faz kontrol tekniğinde meydana gelen mahzurların düzeltilmesi amacıyla PWM teknikleri kullanılmaktadır. Bu tezde incelenen akım kontrollu asimetrik PWM AC kıyıcı, faz kontrol açısına göre kontrol yapan klasik AC kıyıcıların şebeke ve yük tarafında oluşturduğu olumsuzlukları ortadan kaldırmaktadır. Akım kontrollu asimetrik PWM AC kıyıcının şebeke güç katsayısının oldukça iyi olduğu ve çıkış harmoniklerinin kabul edilebilir bir seviyede kaldığı tespit edilmiştir. Ayrıca akım kontrollu asimetrik PWM AC kıyıcı devresi pratik olarak uygulanabilirlik ve giriş güç katsayısı açısından, diğer PWM metodlarına tercih edilebilmektedir.

PWM AC kıyıcı ile yapılan asenkron motor hız kontrol sisteminde, motor hızının istenilen değere hızlı ulaşması ve motorun yüklenmesi durumunda hızın değişmemesi için kontrolun kapalı çevrim içerisinde yapılması gerekmektedir. Sistemin cevap verme süresinin iyileştirilmesi ve uzun süreli kararlılığının sağlanması için PID kontrolun kullanılması uygun olmaktadır.

Laboratuvarda gerçekleştirilen akım kontrollu PWM AC kıyıcı devresi ile yapılan asenkron motor hız kontrolunda, motorun oldukça geniş bir hız aralığında kararlı bir şekilde çalıştığı ve yüklenmelere karşı iyi cevap verdiği gözlenmiştir. Kontrol aralığı boyunca şebeke güç faktörü açısından sistemin çalışmasının oldukça iyi olduğu tespit edilmiştir.

Kontrolun dijital olarak yapılması sayesinde analog kontrolda karşılaşılan problemler ortadan kalkmıştır. Ayrıca, sistemin kullanım amacına ve kullanıldığı yere bağlı olarak istenilen değişiklikler programda kolayca yapılabilmektedir.

Sistem kontrolu için seçilen mikrodenetleyicinin, imkanlarının tamamına yakınının kullanılması, mikrodenetleyicinin uygun seçildiğini göstermektedir. Bazı dijital sistemlerde mikrodenetleyici ile beraber kullanılması gereken ADC, ROM gibi elemanlar da ortadan kalkmış, kontrol tek bir mikrodenetleyici ile gerçekleştirilerek kontrol devresinde kullanılan eleman sayısı ez aza indirgenmiştir. Böylece endüstriyel olarak kulanılabilir ekonomik, bir devre elde edilmiştir.

KAYNAKLAR

[1] Bateson, Robert N., 1989, Introduction to Control System Technology, Third Edition, Merrill Publishing Company.

[2] Bose, B.K., 1986, Power Electronics and AC Drives, Prentice-Hall.

[3] Choe, G., Wallace, A.K., Park, M., "An Improved PWM Technique for AC Choppers.", October 1989, IEEE Transactions on Power Electronics.

[4] Nishimura, T., Nakkaoka, M., Maruhashi, T., ,"Reduction of Vibration and Acoustic Noise in Induction Motor Driven by Three Phase PWM AC Chopper Using Static Induction Transistors.", July 1989, IEEE Transactions On Power Electronics.

[5] Jang, D., Choe, G., "Improvement of Input Power Factor in AC Choppers Using Assymetrical PWM Technique.", April 1995, IEEE Transactions on Industrial Electronics.

[6] Rashid, M. H., 1993, "Power Electronics", Second Edition, Prentice Hall.

[7] PIC16/17 Microcontroller Data Book, 1995, Microchip.

ÖZGEÇMİŞ

Doğum Tarihi	08.09.1972
Doğum Yeri	İstanbul
1978-1983	Beykoz Gümüşsuyu İlkokulu
1983-1986	Beykoz Ziya Ünsel Ortaokulu
1986-1990	Haydarpaşa Anadolu Teknik Lisesi Elektronik Bölümü
1990-1994	Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü
1995	Yıldız Teknik Üniversitesi Elektrik Mühendisliği Bölümü Elektrik Makinaları Anabilim Dalı'nda Araştırma Görevlisi
1996-	Yüksek Lisans