

YILDIZ TEKNİK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

Mikro. Kul. Adım Mot. ile kap.
Çev. kon. kont.

Yüksek Lisans Tezi

Mehmet Yakut

1991

YILDIZ TEKNİK ÜNİVERSİTESİ
KÜTÜPHANE DOKÜMANTASYON
DAİRE BAŞKANLIĞI

R 152
160

Kot	:	
Alındığı Yer	:	Fen Bil. Ens.
Tarih	:	10.4.95
Fatura	:	
Fiyatı	:	177.000,-
Ayniyat No	:	1-6
Kayıt No	:	50989
UDC	:	
Ek	:	

YILDIZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İÇİNDEKİLER

ÖZET

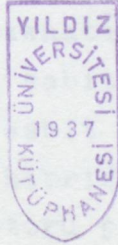
ABSTRACT

BÖLÜM 1

GİRİŞ

BÖLÜM 2

ADIM MOTORLARI

MİKRODENETLEYİCİ KULLANARAK**ADIM MOTORLARI İLE****KAPALI ÇEVİRİM KONUM KONTROLU****YÜKSEK LİSANS TEZİ****TEZİ YÖNETEN: PROF.DR. ATILLA ATAMAN****TEZİ HAZIRLAYAN: MEHMET YAKUT**

İÇİNDEKİLER

ÖZET	V
ABSTRACT	VII
BÖLÜM 1: Giriş	
BÖLÜM 2: ADIM MOTORLARI	
2.1. Adım motoru tanımı	3
2.2. Adım motoru tipleri	4
2.2.1 Selonoid-çark tipi adım motoru	4
2.2.2 Değişken relüktanslı adım motoru	5
2.2.2.1 Çok parçalı statorlu VR adım motoru	5
2.2.2.2 Tek parçalı statorlu VR adım motoru	8
2.2.3 Sabit mıknatıslı adım motoru	13
2.2.3.1 Statoru sabit mıknatıslı adım motoru ...	13
2.2.3.2 Rotoru sabit mıknatıslı adım motoru	14
2.2.4 Karma (hibrid) adım motoru	15
2.3. Adım motoru performans karakteristikleri	21
2.3.1 Adım duyarlılığı (çözümleme)	21
2.3.2 Tek adım cevabı	21
2.3.3 Duyarlılık	22
2.3.4 Statik tutma kuvveti	23
2.3.5 Dinamik sürekli hız motor kuvveti	24
2.3.6 Dinamik hızlanma kuvveti	25
2.4. Adım motoru avantaj ve dezavantajları ..	26
2.4.1 Adım motoru avantajları	26
2.4.2 Adım motoru dezavantajları	26

BÖLÜM 3: ADIM MOTOR SÜRÜCÜ DEVRELERİ VE SIRALAMA LOJİĞİ

3.1	Sıralama lojigi	28
3.2	Adım motor sürücü devreleri	33

BÖLÜM 4: MİKROİŞLEMCİ İLE ADIM MOTOR HIZ DENETİMİ

4.1	Açık çevrim denetimi	38
4.2	Kapalı çevrim denetimi	42
4.2.1	Kapalı çevrim denetimde konum kodlayıcı	42
4.2.2	Rotor dönüş yönünün belirlenmesi	44
4.2.3	Adım onaylama yöntemi ile kapalı çevrim denetimi	46
4.2.4	Kapalı çevrim denetimde uç açısı ve önemi	52
4.2.5	Uç açısı seçerek kapalı çevrim denetimi	53
4.2.6	Zaman gecikmelerini kullanarak uç açısının seçimi	58
4.3	Gelişmiş kapalı çevrim denetim yöntemleri	60
4.3.1	Uç açısı hesaplanması ve yüke uyumlu sabit hız denetimi	60
4.3.2	Sayısal denetim	63
4.3.3	Sayısal PID denetleyici	63
4.3.4	Sayısal PID denetim fikrinin adım motoruna uygulanması	66
4.3.5	Uç adım motoru için PID denetim algoritması	70

BÖLÜM 5: ADIM MOTOR DENETİMİNDE HPC
MİKRODENETLEYİCİ KULLANILMASI

5.1.	HPC'nin genel tanımı	77
5.2.	HPC blok diyagramı ve genel özellikleri	79
5.3.	HPC 16040 yazıcı yapısı ve programlama modeli	79
5.4.	HPC adresleme modları	82
5.4.1	Akümülatörün kaynak olduğu adresleme modları	82
5.4.2	Belleğin doğrudan hedef olduğu adresleme modları	83
5.5.	Etkin komut kodu kullanımı	84
5.5.1	Tek byte'lık komutlar	84
5.5.2	Bit işleme komutları	85
5.6	HPC komut kümesi	86
5.6.1	Aritmetik işlem komutları	86
5.6.2	Bellek değiştirme komutları	87
5.6.3	Bit düzenleme komutları	87
5.6.4	Bellek aktarma komutları	87
5.6.5	Yazıcıya ivedi yükleme komutları	88
5.6.6	Akümülatör ve C komutları	89
5.6.7	Dallanma ve kontrol komutları	90
5.7	HPC donanım birimleri ve yonga bacaklarının tanımlanması	91
5.7.1	Giriş/Çıkış uçları	91
5.8	HPC zamanlayıcıları	95
5.8.1	HPC'de zamanlayıcı işlemleri	96
5.8.2	HPC zamanlayıcı çıkışları	98
5.8.3	Zamanlayıcı yazıcıları	99
5.9	HPC kesme hizmetleri	102
5.9.1	Kesme hizmetlerinin yürütülmesi	103
5.9.2	Kesme kontrol yazıcıları	103

BÖLÜM 6:	SONUÇ	105
----------	-------------	-----

EKLER	107
-------------	-----

KAYNAKLAR	110
-----------------	-----

ÖZGEÇMİŞ	111
----------------	-----

ÖZET

20. yüzyılın karakteristiğini çizen en önemli gelişme, yarıiletken teknolojisindeki hızlı gelişme sonucu ortaya çıkan bilgisayarlar ve robot sistemleridir.

Bu nedenle çalışmamda çeşitli bilgisayar çevrebirimlerinde ve robot sistemlerinde yaygın olarak kullanılan adım motorları ve denetim yöntemleri ele alınmıştır. Adım motorları gelişim sırası ile incelenerek, temel motor karakteristikleri, avantaj ve dezavantajları burada verilmiştir.

Basit ve temel sürücü devreleri kullanılarak, devre elemanlarının seçim kriterleri ve devrelerin birbirlerine göre üstünlükleri verilmiştir.

Denetim yöntemleri incelenirken öncelikle açık çevrim denetimi irdelenerek konuya giriş yapılmıştır. Daha sonra konum kodlayıcı konusu ve bu elemanın kapalı çevrim denetiminde nasıl kullanılacağı anlatılmıştır. Kapalı çevrim denetiminde önce sabit uç açılı denetim yöntemi ele alınmış yönteme ait sakıncalar ortaya konmuştur. Bu sakıncalar nedeniyle üzerinde çalışılan sistemin ihtiyacı doğrultusunda, daha gelişmiş kapalı çevrim denetim yöntemleri araştırılmıştır. Bu araştırmalar sonucunda PID (oransal-integral-türevsel) denetim yöntemine karar verilmiş ve uygun bir algoritma geliştirilmiştir. Geliştirilen PID denetim algoritmasını yürütecek HPC mikrodenetleyici assembler programı yazılmıştır. Bu işlem tamamlandıktan sonra HPC'ye üstünlük kazandıran zamanlayıcılar gözönünde bulundurularak, 3 adet adım motorunu denetleyecek PID denetim algoritması geliştirilmiştir.

VI

Bu çalışma sonunda elde edilen sistem, erişim hızının yüksek ve konumlama hatasının minimum olması istenen bir robot düzeneğinde kullanılabilir. Bu çalışmada yapılanların, konu üzerinde çalışacaklara faydalı olmasını dilerim.

In this study, stepping motors widely used in computer peripherals and robot systems and their driving and control methods are investigated. This investigation is made with regard to stepping motor history. Advantages and disadvantages of step motor and basic motor characteristics are presented.

Criteria for choosing the circuit elements are explained by using simple circuitries and also advantages and disadvantages of these circuitries are included.

As an introduction to the control methods of stepping motor, first open loop control is presented. Position encoder and its use in closed loop control is then introduced. In closed loop control section, constant lead angle control is represented first, then its drawbacks are listed. As a result of these drawbacks, advanced closed loop control methods are investigated, for the system requirements. Finally, PID (proportional-integral-derivative) control method is designed for the stepping motor and an algorithm is developed. MFC microcontroller assembler program is adapted to this algorithm. Considering MFC powerful timer and interrupt sources, an assembler program is developed for the PID control of three stepping motors.

The system described here can be used in a robot system, which requires high acceleration and low positioning error. I hope this study will be helpful for later studies.

VII

ABSTRACT

As a result of the technological development, the most important advances which characterize the twentieth century are computers and robot systems.

In this study, stepping motors widely used in computer peripherals and robot systems and their driving and control methods are investigated. This investigation is made with regard to stepping motor history. Advantages and disadvantages of step motor and basic motor characteristics are presented.

Criterion for choosing the circuit elements are explained by using simple circuitries and also advantages and disadvantages of these circuitries are included.

As an introduction to the control methods of stepping motor, first open loop control is presented. Position encoder and its use in closed loop control is then introduced. In closed loop control section, constant lead angle control is represented first, then its drawbacks are listed. As a result of these drawbacks, advanced closed loop control methods are investigated, for the system requirements. Finally, PID (proportional-integral-derivative) control method is decided for the stepping motor and an algorithm is developed. HPC microcontroller assembler program is adapted to this algorithm. Considering HPC powerful timers and interrupt sources, an assembler program is developed for the PID control of three stepping motors.

The system described here can be used in a robot system, which requires high acceleration and low positioning error. I hope this study will be helpful for later studies.

BÖLÜM 1

GİRİŞ:

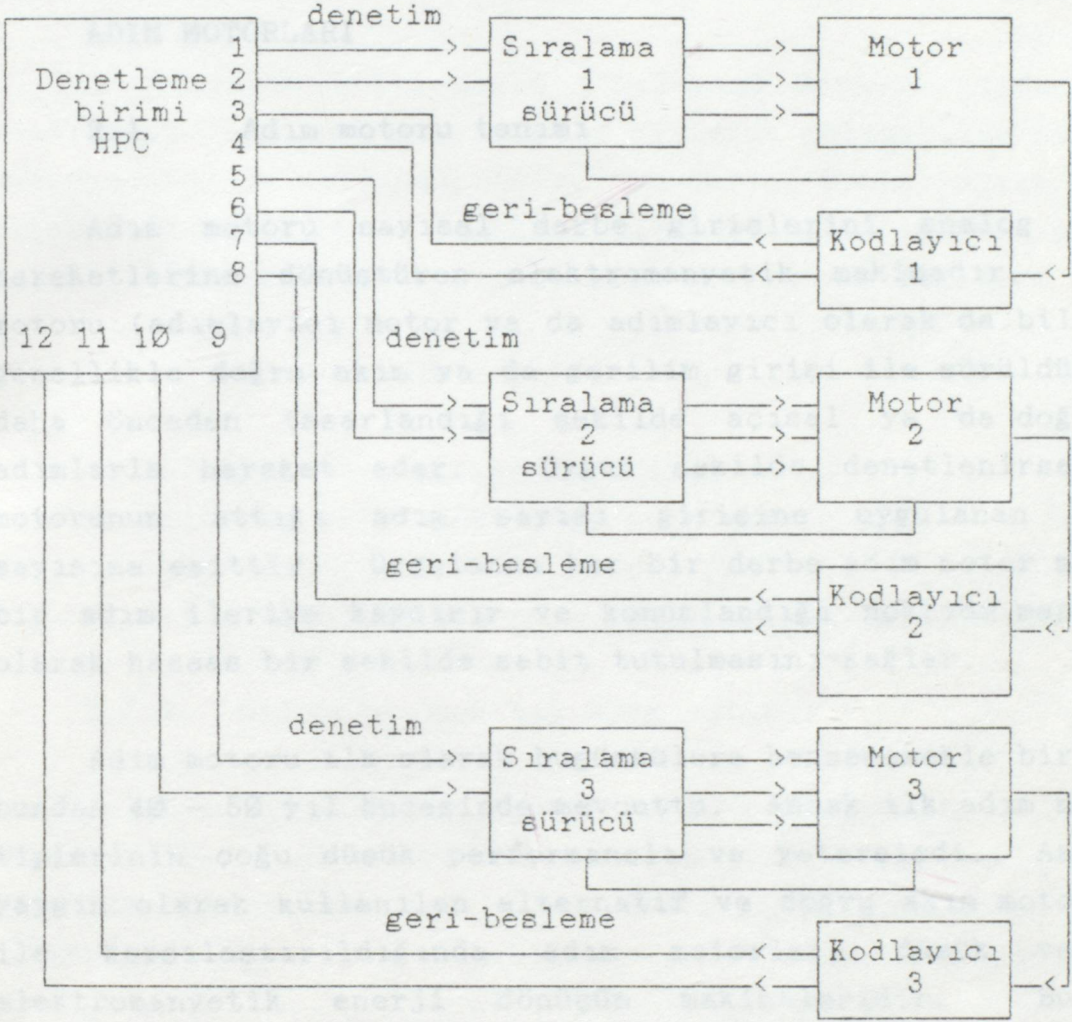
Bilgisayarlar ve robot sistemleri gelişen teknolojinin sonucu olarak günlük hayatta yaygın olarak kullanılır hale gelmiştir. Kullanımın yaygınlaşması insan ihtiyaçlarının sınırlarını genişleterek ve çalışan sistemlerden daha yüksek hız, verim ve duyarlılık gereksinimini daha da arttırmıştır.

Bilgisayar sisteminde, adım motoru ve denetim yöntemi bilgisayar sisteminin bilgi erişim hızını, erişilen bilginin doğruluğunu, kullanılan sürücü devresi ise aynı sistemin verimini doğrudan etkilemektedir. Eğer adım motoru bir robot sisteminde kullanılıyorsa kullanılan adım motoru sistemin bir noktadan diğer bir noktaya erişim hızını ve erişilen noktanın doğruluğunu etkileyecektir. Benzer şekilde adım motorunun denetim yöntemi de sonucu doğrudan etkileyecektir.

Adım motorları normalde açık çevrim denetimine uygun bir yapıya sahiptir. Ancak özellikle yükün dinamik olarak değiştiği robot sistemlerinde adım motorunun kapalı çevrimle denetlenmesi gereklidir. Kapalı çevrim denetimi ile hem sistemin konumlama hatası en düşük düzeyde tutulur hem de erişim hızı daha yüksektir.

Şekil 1.1,deki blok diyagramda üç adet adım motoru için kapalı çevrim denetim sistemi görülmektedir. Tasarlanan sistemde HPC mikrodenetleyicisi her bir adım motoru için yön ve adımlama bilgisi üretmekte ve motorların konum bilgilerini açısız kodlayıcılardan okumaktadır. Motorlara ait yön bilgileri zamanlayıcı çıkışları kullanılarak elde edilmekte ve kodlayıcılardan okunan konum bilgileri kesme girişlerinde değerlendirilmektedir. Motorlar için denetim yöntemi olarak PID (oransal-integral-türevsel) kullanılmaktadır. HPC'de mevcut zamanlayıcılar ve aritmetik veri işleme komutları PID

denetim algoritmasını geliştirmede kolaylık sağlamaktadır.



Şekil 1.1 Tasarlanan sistem blok diyagramı

BÖLÜM 2

ADIM MOTORLARI

2.1. Adım motoru tanımı

Adım motoru sayısal darbe girişlerini analog çıkış hareketlerine dönüştüren elektromanyetik makinadır. Adım motoru (adımlayıcı motor ya da adımlayıcı olarak da bilinir) genellikle doğru akım ya da gerilim girişi ile sürüldüğünde daha önceden tasarlandığı şekilde açısal ya da doğrusal adımlarla hareket eder. Uygun şekilde denetlenirse adım motorunun attığı adım sayısı girişine uygulanan darbe sayısına eşittir. Uygulanan her bir darbe adım motor milini bir adım ileriye kaydırır ve konumlandığı noktada manyetik olarak hassas bir şekilde sabit tutulmasını sağlar.

2.2. Belkoid-park tip adım motoru

Adım motoru ilk olarak bugünlüklerle benzememekle birlikte bundan 40 - 50 yıl öncesinde mevcuttu. Ancak ilk adım motoru tiplerinin çoğu düşük performanslı ve yetersizdi. Aslında yaygın olarak kullanılan alternatif ve doğru akım motorları ile karşılaştırıldığında adım motorları düşük verimli elektromanyetik enerji dönüşüm makineleridir. Bununla birlikte sayısal sistemlerde ve mikrobilgisayarlar da sağlanan büyük gelişmeler adım motoru kullanımının yaygınlaşmasında en önemli etken olmuştur. Günümüzde adım motoru endüstride birçok kontrol sisteminde kullanılmaktadır. Kullanım yerlerine örnek olarak bilgisayar çevre birimleri (yazıcı, kart okuyucu, disk sürücü ve bellek erişim düzenekleri) sayılabilir. Ayrıca adım motoru nümerik kontrol sistemlerinde uzay endüstrisinde ve robot eklemlerinde de kullanılmaktadır. Adım motoru, sürücü ve rotorun birlikte belirleyeceği en yüksek hıza kadar verilen giriş darbe hızıyla her iki yönde de denetlenebilir. Çabuk ivmelenme, yavaşlama ve durma yeteneği nedeniyle adım motoru, artımlı hareket denetiminde dur-kalk işlemleri için en uygun

elemandır. Adım motoru düşük rotor ataleti, kaymasız ve birikimsiz konumlama yeteneği ile karakterize edilebilir.

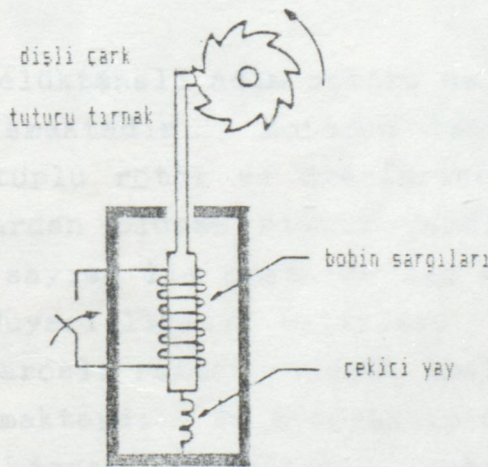
Adım motoru belli darbe dizilerini verilen hızda takip edebildiği için, değişik senkron hızlarda çalışabilen senkron motor olarak da kullanılabilir. Bu duruma örnek; düşük hızlı alternatif akım (a.a.) senkron makinaya olan ihtiyaç sonucu geliştirilen, sabit mıknatıslı adım motoru verilebilir.

2.2. Adım motoru tipleri

- a) Selonoid-çark tip adım motoru (VR)
- b) Değişken relüktanslı tip adım motoru (VR)
- c) Sabit mıknatıslı adım motoru (PM)
- d) Hibrit (karma) tip adım motoru (senkron endüktör)

2.2.1. Selonoid-çark tip adım motoru

Çalışma prensibi açısından adım motorlarının en basit türüdür. Bu motor yaylı bir mekanizma ile tekrar yerine geri dönebilen bir bobin nüvesi, bu nüveye bağlı bir tırnak ve bu tırnak yardımıyla kilitlenebilen bir dişli çarktan oluşur.



Şekil 2.1. Selonoid-çark tip adım motoru

Bobin enerjilendiğinde, nüveye mille tutturulan tırnak, çarkın dişlerinden birine takılarak, çarkın bir diş adımı kadar dönmesine neden olur. Bobinin enerjisi kesildiğinde yay yardımıyla nüve tekrar eski konumunu alır, böylece tırnak mekanizması da bir sonraki oyuğa oturur. Sonuç olarak adım motoru bir adım ilerlemiştir. Şekil 2.1.'de görüldüğü gibi bu tip adım motoru yalnızca bir yönde hareket yeteneğine sahiptir. Ancak diş sayısı ikiye çıkarılıp tırnaklar da özel bir mekanizma ile bağlanırsa iki yönlü hareket de elde edilebilir.

2.2.2. Değişken relüktanslı adım motoru (VR)

Endüstride en yaygın kullanılan adım motoru türü değişken relüktanslı adım motorudur. Bu motorlar da temel olarak iki kısımda incelenebilir.

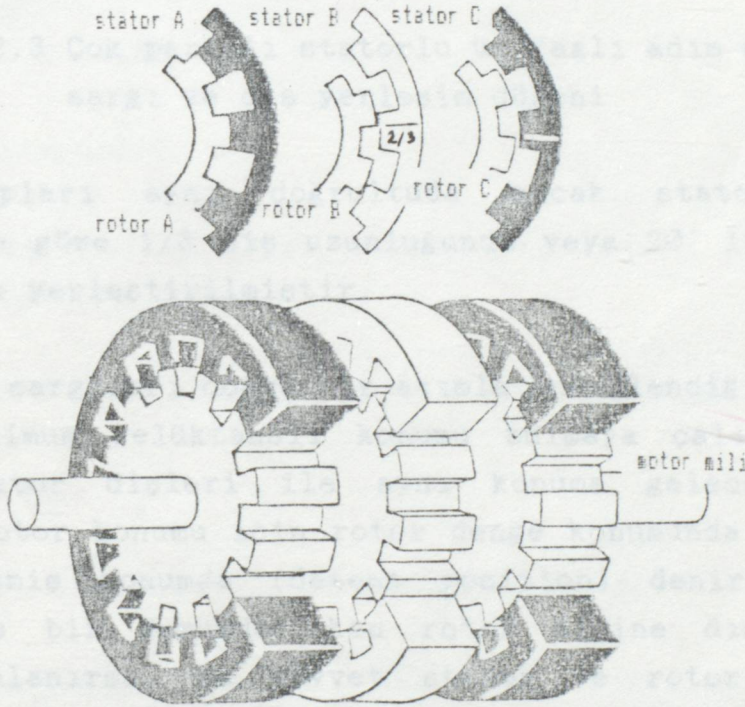
- a) Çok parçalı statorlu VR adım motoru
- b) Tek parçalı statorlu VR adım motoru

VR adım motorunda rotor dairesel yada doğrusal olabilir.

2.2.2.1. Çok parçalı statorlu VR adım motoru

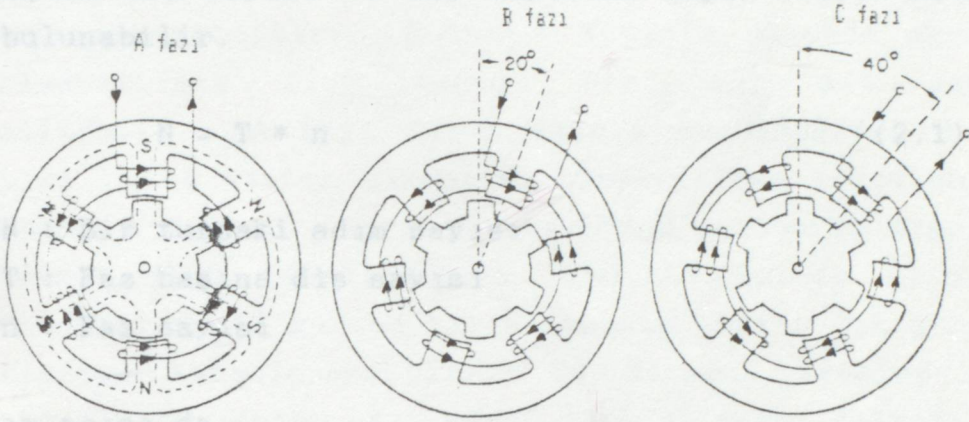
Değişken relüktanslı adım motoru da bir bakıma selonoid-çark gibi çalışmaktadır. Motorun temel yapısında yumuşak demirli çok kutuplu rotor ve üzerlerine bobin sarılmış ince plakalı kutuplardan oluşan stator vardır. Rotor ve stator üzerindeki diş sayısı ile statorun faz sayısı adım motorunun bir adımlık duyarlılığını belirler. Önceleri VR adım motorları çok parçalı rotorlu olarak imal edilmekte ve yaygın olarak kullanılmaktaydı. Bu motorların çoğu da üç ya da dört fazlı olarak tasarlanmaktaydı. Daha sonra çok parçalı statorlu adım motorlarının yüksek maliyetlerinden dolayı, tek parçalı statorlu adım motorları yaygınlaşmaya başlamıştır.

Üç fazlı çok statorlu adım motoru yapısı itibariyle, aynı mile monte edilmiş birbirinden elektriki olarak bağımsız stator ve rotorlardan oluşmaktadır. Stator ve rotor bölümlerinden herbiri eşit diş sayısına ve eşit büyüklükteki diş oyuklarına sahiptir. Stator bölümleri mekaniki olarak birbirine bağlıdır ancak manyetik ve elektriki açıdan yalıtılmıştır. Rotor parçaları da ortak mil üzerine monte edilmiş ancak manyetik olarak yalıtılmıştır. Rotorun döndürülebilmesi için rotor veya stator parçaları birbirlerinden belirli miktarlarda ilerde ve geride yerleştirilmiştir.



Şekil 2.2 Çok parçalı statorlu adım motoru diş yerleşme düzeni

Bu durum Şekil 2.2.'de açıklanmaya çalışılmıştır. Buna göre üç fazlı adım motoru için stator parçaları birbirinden $1/3$ diş uzunluğu kadar döndürülmüş olarak yerleştirilmelidir. Anlatım kolaylığı açısından Şekil 2.3.'te üç fazlı çok parçalı statorlu ve herbir statorda altı adet diş bulunan adım motoru sargı ve kutup düzeni gösterilmiştir. Şekilde



Sekil 2.3 Çok parçalı statorlu üç fazlı adım motoru için sargı ve diş yerleşim düzeni

rotor kutupları aynı doğrultuda ancak stator dişleri birbirlerine göre $1/3$ diş uzunluğunda veya 20° lik açılarla kaydırılarak yerleştirilmiştir.

A fazı sargıları doğru bir akımla enerjlendiğinde, rotor dişleri minimum relüktanslı konumu bulmaya çalışacak ve A fazının stator dişleri ile aynı konuma gelecektir. Bu şartlarda rotor konumu için rotor denge konumunda yada rotor dişe yerleşmiş konumda (detent position) denir. A fazı enerjilenmiş bir durumda iken rotor miline dışarıdan bir kuvvet uygulanırsa, bu kuvvet stator ve rotor dişlerinin denge konumunu bozacak ve buna karşı denge durumunu koruyabilmek için, karşı yönde bir kuvvet oluşacaktır. Bu kuvvet motorun statik tutma kuvveti (static holding torque) olarak bilinir. Kuvvet değeri, stator ve rotor dişlerindeki kaymanın miktarına bağlı olarak değişir. Dışarıdan uygulanan kuvvet arttıkça, tutma kuvvetinin tepe değerine ulaşılincaya kadar konumdaki bozulma artar. Bu noktadan sonra artan konum bozulmasıyla birlikte, statik tutma kuvveti de tekrar sıfıra ulaşincaya kadar azalmaya başlar.

Çok parçalı statorlu motor için bir turdaki adım sayısı faz başına diş sayısı ve faz sayısına bağlı olarak aşağıdaki gibi bulunabilir.

$$N = T * n \quad (2.1)$$

N : Bir turdaki adım sayısı

T : Faz başına diş sayısı

n : Faz sayısı

$$R = 360 / N \quad (2.2)$$

ile bulunabilir.

Şekil 2.3. te verilen örnek adım motoru için $n = 3$ ve $T = 6$ dır. Bu bilgilere göre

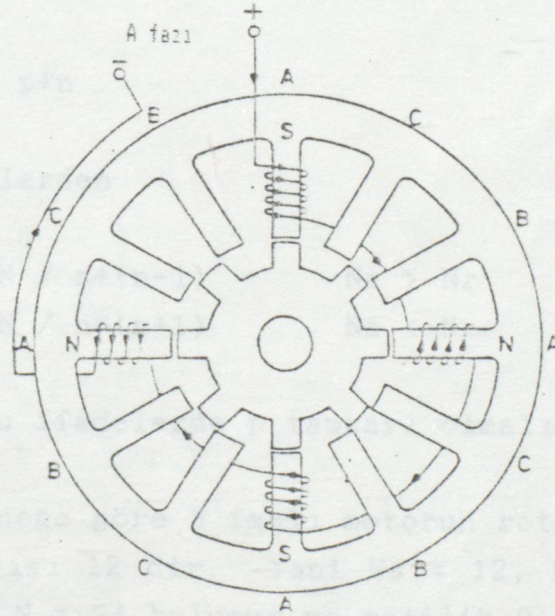
$$N = 3 * 6 = 18 \text{ adım/tur ve } R = 360 / 18 = 20 \text{ derece/adım}$$

olarak bulunur.

2.2.2.2. Tek parçalı statorlu VR adım motoru

Tek parçalı statorlu VR adım motorları ölçüleri açısından daha küçük hacimli ve yapımları açısından daha basittir. Ancak bu tip motorlarda sadece belli adım açıları elde edilebilmektedir. Tek parçalı statorlu VR adım motoru rotor ve statorda sadece belli diş gurubunun kilitlenmesiyle çalışır. Bu tür motor için aynı zamanda rotor ve stator üzerindeki diş sayısının da farklı olması gerekmektedir. Rotor ve stator diş sayılarının gerçekleştirilebilir kombinasyonları kullanılarak tek parçalı statorlu VR adım motoru için 1,8 ile 30 derece arasındaki adım açılarını elde etmek mümkündür.

Şekil 2.4'de üç fazlı VR adım motoru diyagramı görülmüyor. Örnek motorda stator 12 diş sayısına, rotor ise 8 diş sayısına sahiptir. Motor 4 kutuplu düzende sarılarak gerçekleştirilmiş olup sadece bir faza ait sargılar gösterilmiştir. A fazı doğru akımla sürüldüğünde 4 rotor çıkıntısı ile 4 stator çıkıntısı birbirlerine kenetlenirler. Bu durumda A fazının enerjisi kesilip B fazı sürülürse rotor saat dönüş yönü doğrultusunda (CW) 15 derece dönecek ve rotorun diğer 4 diş gurubu ile B fazının stator diş gurupları birbirlerine kenetlenecektir. Bu durumun tersine C fazı sürülmüş olsaydı rotor saat dönüş yönünün tersi doğrultuda 15 derece dönmüş olacaktı.



Şekil 2.4 Üç fazlı tek parçalı statorlu adım motoru

Tek parçalı statorlu adım motorunda stator ve rotor dişleri arasında aşağıdaki ilişki vardır. İcin tek fazlı tek

$$N_s = N_r \pm p \quad (2.3)$$

N_s : Stator diş sayısı

N_r : Rotor diş sayısı

p : Faz başına stator diş sayısı

N sayısı rotorun bir devrindeki adım sayısı ise derece olarak adım açısı

$$R = 360^\circ / N \quad (2.4)$$

ile bulunur. Bunlara göre

$$N = | N_s * N_r / (N_s - N_r) | \quad (2.5)$$

n faz sayısı ise

$$N_s = p * n \quad (2.6)$$

dir ve bu eşitliklerden

$$p = N / n * (n-1) \quad N_s > N_r \quad (2.7)$$

$$p = N / n * (n+1) \quad N_s < N_r \quad (2.8)$$

olarak bulunur, bu ifadelerde p tamsayı olmalıdır.

Şekildeki örneğe göre 3 fazlı motorun rotor diş sayısı 8 ve stator diş sayısı 12 dir. Yani $N_s = 12$, $N_r = 8$ ve $n = 3$ tür. Bunlara göre $N = 24$ bulunur ve eşitlik 2.7'ye göre

$$p = 24 / 3 * 2 = 4 \text{ bulunur}$$

Tek parçalı statorlu adım motoru eşitlikleri kullanılarak elde edilebilecek en büyük diş açısı 30° derecedir. Bir dönüşteki adım sayısı N 'in minimum olması için tek fazlı tek dişe sahip motorun dönemeyeceği düşünülerek, p için

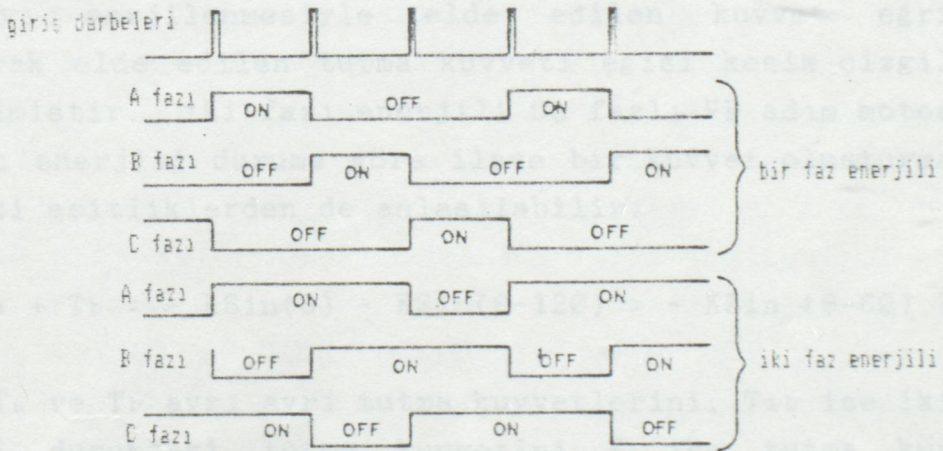
seçilebilecek en küçük tamsayı 2 olarak belirlenir. Buna göre $N = 2n(n-1)$ olarak alınır ve iki yönlü dönme için faz sayısı n en küçük 3 olabilir. Bunlara göre

$$N = 2 * 3 * 2 = 12 \text{ [adım/tur] ve}$$

en büyük adım açısı da

$$R = 360^\circ / 12 = 30^\circ \text{ olarak elde edilir.}$$

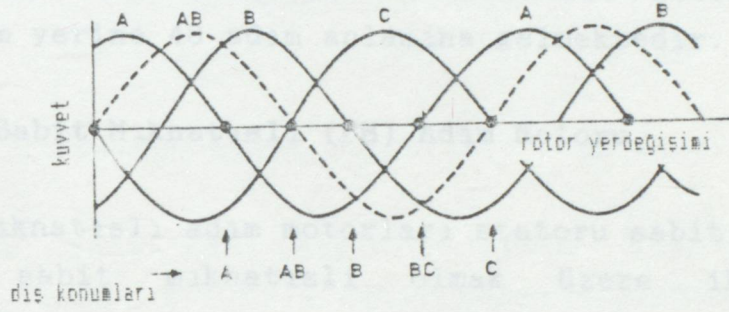
Şekil 2.5'te üç fazlı VR adım motorunun bir fazı enerjili ve iki fazı enerjili zaman diyagramları görülmektedir. Dörtten daha çok faz sayısına sahip adım motorlarında bir kerede iki ya da daha fazla fazın enerjilenmesi yaygın olarak kullanılır. Bununla birlikte üç ya da dört fazlı motor için pratikte enerjilenebilecek max faz sayısı ikidir.



Şekil 2.5 Üç fazlı adım motoru için tek fazı enerjili ve iki fazı enerjili zamanlama diyagramları

Prensip olarak iki fazı enerjilenmiş 3 fazlı VR adım

motorunda tüm motor sargılarına uygulanan akımın aynı olduğu varsayılarak, tek fazlı enerjilenmiş üç fazlı VR adım motoruna göre elde edilebilecek max tutma kuvvetinde herhangi bir değişme olmaz. Şekil 2.6'da üç fazlı VR adım motoru için düz çizgilerle ideal ve statik tutma kuvveti eğrileri gösterilmektedir.



Şekil 2.6 Üç fazlı adım motoru için ideal statik tutma kuvveti eğrileri

A ve B fazları aynı anda enerjilendiğinde, A ve B fazlarının ayrı ayrı enerjilenmesiyle elde edilen kuvvet eğrileri toplanarak elde edilen tutma kuvveti eğrisi kesik çizgilerle gösterilmiştir. İki fazlı enerjili üç fazlı VR adım motorunun tek fazlı enerjili duruma göre ilave bir kuvvet oluşturmadığı aşağıdaki eşitliklerden de anlaşılabılır.

$$T_{ab} = T_a + T_b = -K \sin(\theta) - K \sin(\theta - 120^\circ) = -K \sin(\theta - 60^\circ) \quad (2.9)$$

Burada T_a ve T_b ayrı ayrı tutma kuvvetlerini, T_{ab} ise iki faz enerjili durumdaki tutma kuvvetini K ise tutma kuvveti sabitini göstermektedir. Burada dikkat edilmesi gereken önemli bir diğer nokta ise iki fazın enerjilenmesi durumu motorun adım açısını etkilememektedir. Ancak iki fazlı enerjili durumda motorun gerçek dış yerleşme konumları, tek fazın enerjili olduğu duruma göre yarım adım açısı kadar ileri ya da geri kaymıştır. Şekil 2.6 dikkatle incelenirse

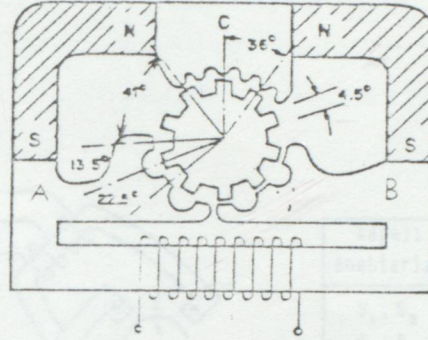
başka bir nokta daha göze çarpar. Fazlar A,AB,B,BC,C ve CA sırasıyla enejilendiğinde yarım-adım modu denilen anahtarlama yöntemi elde edilmektedir. Şekil 2.6'da görüldüğü gibi yarım-adım modunda motorun doğal diş yerleşme konumları arasında ilave kuvvet eğrileri oluşmaktadır. Sonuç olarak motorun hareket halindeki kuvvet değişimleri daha yumuşak olacaktır. Bunun diğer bir sonucu da motor adım açısının iki kat düşürülmüş olmasıdır, bu da örnekteki motor için bir turda 24 adım yerine 48 adım anlamına gelmektedir.

2.2.3. Sabit Mıknatıslı (PM) Adım Motoru

Sabit mıknatıslı adım motorları statoru sabit mıknatıslı ve rotoru sabit mıknatıslı olmak üzere iki grupta incelenebilir.

2.2.3.1. Statoru sabit mıknatıslı adım motoru

Adından da anlaşılacağı gibi bu motor statora sabit mıknatısların yerleştirilmesiyle oluşturulur. Şekil 2.7'de görüldüğü gibi statordaki sabit mıknatıslardan çıkan manyetik akı yumuşak demir üzerine oyuklar (dişler) açılarak oluşturulan kutup parçaları üzerinden geçerek yine yumuşak demirli rotor üzerinden devreyi tamamlar. Devre üzerinden akan bu manyetik akı rotoru etkileştirir. Şekil 2.7'de görüldüğü gibi rotor üzerine on adet diş yerleştirilmiştir, yani diş adımı uzunluğu 36 derecedir. Stator üzerinde ise üç adet diş gurubu görülmektedir. A ve B diş gurupları da rotorda olduğu gibi 36 derecelik adım açılına sahiptir. A ve B'deki dişler öyle yerleştirilmiştir ki bunlardan A veya B gurubu rotor dişleri ile aynı hizaya geldiğinde diğer gurup rotor dişlerinden tamamen ayrılmaktadır. C diş gurubunda ise adım uzunluğu 18 derecedir. Stator sargılarını +V,Ø,-V,Ø sırasıyla enerjilediğimizde rotor saat dönüş yönünde 18 derecelik adımlarla dönecektir.



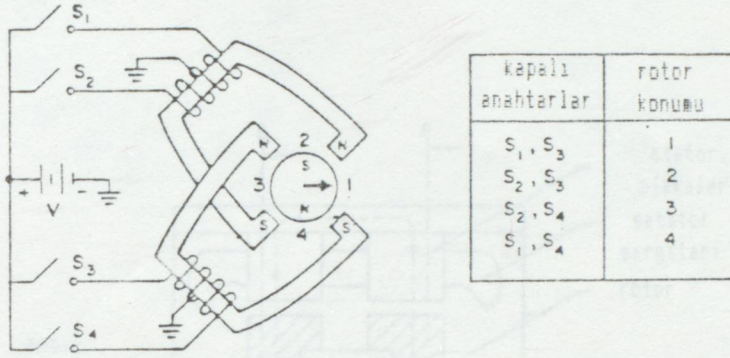
Şekil 2.7 Statoru sabit mıknatıslı adım motoru

Yukarıda anlatılan statoru sabit mıknatıslı adım motorunun en önemli avantajı motor sargıları enerjilenmediği durumda bile oldukça yüksek tutma kuvvetine sahip olmasıdır. Ayrıca yapımının basit olması da ilave bir avantaj olarak sayılabilir. Ancak motorun sadece bir yönde dönebiliyor olması ve motordan elde edilebilecek hızın çok düşük olması en önemli dezavantajlarıdır.

2.2.3.2. Rotoru sabit mıknatıslı adım motoru

Bu tip motorlar da yapımlarının basitliği ve ucuz olmaları ile karakterize edilebilirler. Sabit mıknatıslı adım motorlarının 2-8 faz ve 2-24 kutup sayısına sahip farklı tipleri mevcuttur.

Rotoru sabit mıknatıslı adım motorunda rotor enine açısız dilimler halinde mıknatıslanmış kutuplara sahiptir. Şekil 2.8'de çift kutuplu çift statorlu en basit rotoru sabit mıknatıslı adım motoru görülmektedir. Şekilde verilen tablodaki sırayla sargılar enerjilendiğinde rotor saat dönüş yönü doğrultusunda yada tersi doğrultuda 90 derecelik adımlarla dönecektir. Şekil 2.8'deki motorun sürülmesi yöntemi tek yönlü sürme olarak bilinir. Bu şekilde



Şekil 2.8 Rotoru sabit mıknatıslı adım motoru

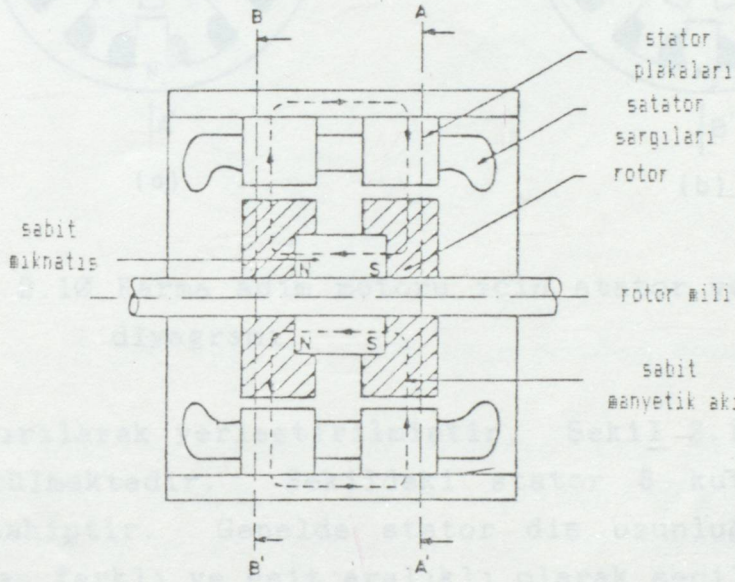
isimlendirmenin nedeni kullanılan kaynağın tek polariteli olmasındandır. Ancak motor sargıları dışarıda paralel hale getirilebilecek şekilde, uçları dışarı çıkarılırsa, çift yönlü sürme işlemi de gerçekleştirilebilir. Bu durumda motorun adım atmasını sağlamak için sargı akımlarının yönünü değiştirmek gerekmektedir.

2.2.4. Karma (hibrid) adım motorları (senkron endüksiyon makinaları)

Rotoru sabit mıknatıslı diğer bir adım motoru da karma adım motoru ya da senkron endüksiyon makinası olarak bilinir. Karma (hibrid) terimi motor hem sabit mıknatıslı hem de değişken relüktanslı adım motoru prensibiyle çalıştığı için kullanılmaktadır.

Bu motor tipi ilk olarak düşük hız uygulamaları için alternatif akım iki fazlı senkron makina şeklinde tasarlanmıştır. Daha sonra bu motorun doğru akımla sürülerek adım motoru olarak da kullanılabileceği görülmüştür. Yüksek

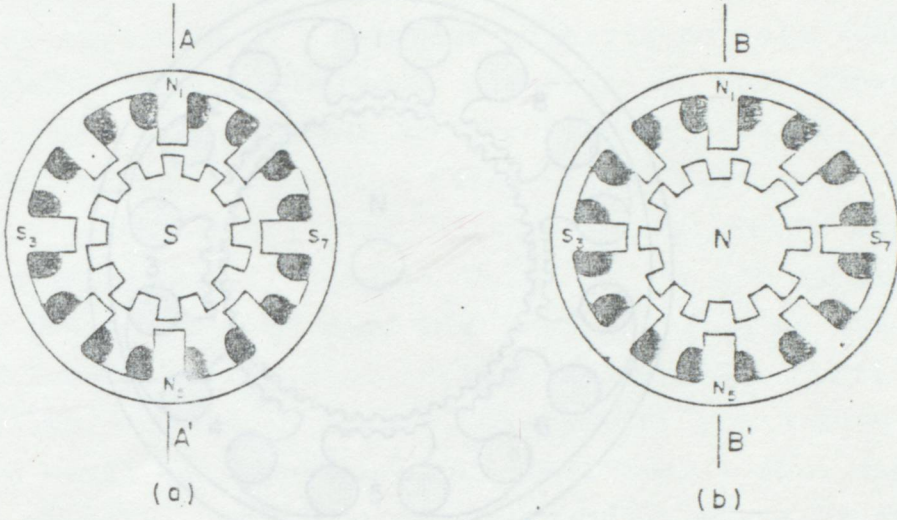
verimi ve diğer avantajları nedeniyle karma adım motoru endüstride değişken relüktanslı adım motoru ile birlikte çok yaygın olarak kullanılan adım motoru tipidir. Karma adım motorlarının yaygın olarak kullanılan adım açısı 1.8 derecedir (200 [adım/tur]).



Şekil 2.9 Karma adım motoru kesit görünüşü

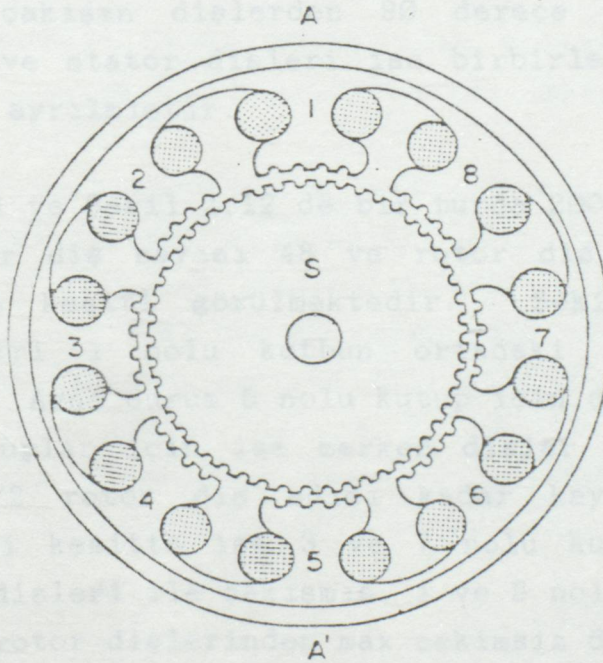
Motorun çalışma prensibi ve yapımı Şekil 2.9 ve Şekil 2.10'da görülmektedir. Rotor eksenini doğrultusunda, tek yönlü manyetik alan oluşturmak üzere, konumlanmış sabit mıknatıslar vardır. (Prensip olarak aynı alan, stator parçaları arasında rotora göre doğru açıda yerleştirilen sabit bir bobinle de elde edilebilir).

Şekil 2.9'da sabit mıknatısın manyetik akı yolları görülmektedir. Bu motorda rotor iki kısımdan oluşmaktadır. Bu kısımlardan biri kuzey (N) diğeri güney (S) olacak şekilde manyetik olarak kutuplanmıştır. Herbir kısımdaki rotor dişleri eşit aralıklı olarak yerleştirilmiştir. Ancak heriki kısımdaki rotor dişleri birbirlerinden 1/2 rotor diş adımı



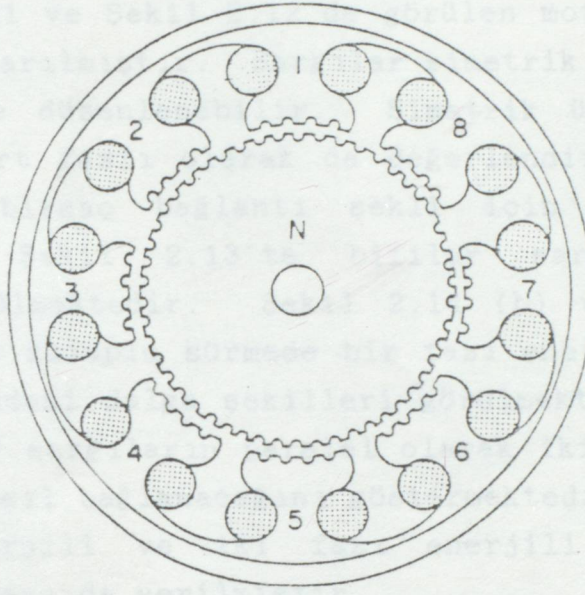
Şekil 2.10 Karma adım motoru için stator ve rotor kutup diyagramı

kadar kaydırılarak yerleştirilmiştir. Şekil 2.10'da motorun kesiti görülmektedir. Şekildeki stator 8 kutup yada diş sayısına sahiptir. Genelde stator diş uzunluğu rotor diş uzunluğundan farklı ve eşit aralıklı olarak seçilir.



Şekil 2.11 200 adım/tur'luk adım motoru kesiti (S kutbu)

B



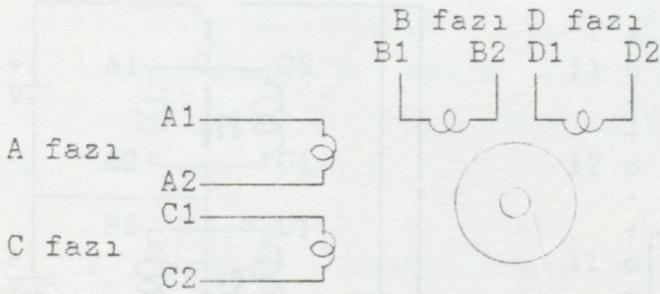
B'

Şekil 2.12 200 adım/tur'luk adım motoru kesiti (N kutbu)

Karma adım motorunun en önemli özelliği Şekil 2.10 incelenerek açıklanabilir. Rotor ve stator diş uzunlukları o şekilde ayarlanır ki birbirlerine göre 180 derecelik açılarda bulunan iki stator dişi ile sadece iki rotor dişi çakışır. Aynı zamanda çakışan dişlerden 90 derece farklı açılarda bulunan rotor ve stator dişleri ise birbirlerinden manyetik olarak tamamen ayrılmıştır.

Şekil 2.11 ve Şekil 2.12'de bir turda 200 adım atan (1.8 derece), stator diş sayısı 48 ve rotor diş sayısı 50 olan adım motorunun kesiti görülmektedir. Şekil 2.11'de rotor dişlerinden biri 1 nolu kutbun ortadaki dişi ile aynı doğrultudadır. Aynı durum 5 nolu kutup için de geçerlidir, 3 ve 7 nolu kutuplar için ise merkez dişler en yakın rotor dişlerinden 1/2 rotor diş adımı kadar kaymış konumdadır. Şekil 2.12'deki kesitte ise 3 ve 7 nolu kutupların merkez dişleri rotor dişleri ile çakışmış, 1 ve 5 nolu kutplarda ise merkez dişler rotor dişlerinden max çekimsiz durumda yani 1/2 adım uzakta konumlanmıştır.

Şekil 2.11 ve Şekil 2.12'de görülen motorun 8 kutbu iki fazlı olarak sarılmıştır. Sargılar simetrik üç uçlu yada tek bobin şeklinde düzenlenebilir. Simetrik üç uçlu düzenleme için motor dört fazlı olarak da değerlendirilebilir. Çünkü mümkün olan birkaç bağlantı şekli için dört ayrı sargı mevcuttur. Şekil 2.13'te bifilar sargıların şematik diyagramı görülmektedir. Şekil 2.14 (b) ve (c)'de bifilar sargıların tek kutuplu sürmede bir fazı enerjili ve iki fazı enerjili durumdaki dalga şekilleri görülmektedir. Şekil 2.14 (d)'de bifilar sargıların paralel olarak iki yönlü sürülmesi işlemi için nasıl bağlanacağını göstermektedir. Aynı zamanda bir fazı enerjili ve iki fazı enerjili sürme için faz enerjileme sırası da verilmiştir.



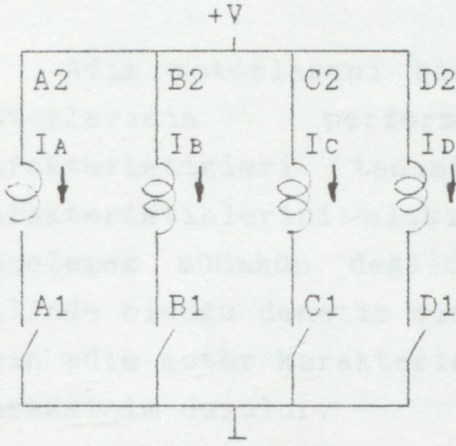
Şekil 2.13 Dört fazlı karma adım motoru için bifilar (ortak ucu olmayan) sargı diyagramı

Şekil 2.11'deki karma adım motoruna ait stator aslında değişken relüktanslı (VR) adım motoru için de aynıdır. Yani aynı stator kullanılarak değiştirilebilir rotor ile hem değişken relüktanslı (VR) hem de karma adım motoru elde etmek mümkündür. Daha önce de belirtildiği gibi karma adım motoru senkron indüksiyon motoru olarak da bilinir. Karma adım motoru iki fazlı sinüsoidal işaret ile sürüldüğünde motor senkron makina gibi çalışacaktır. Motor hızı,

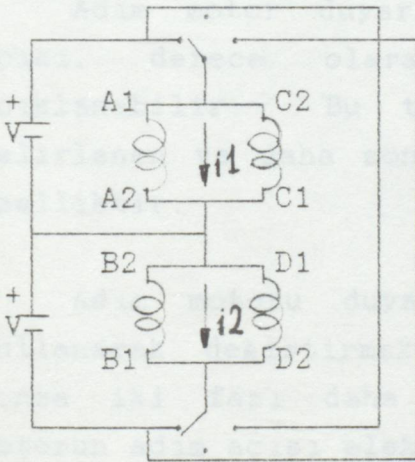
$$S = 50 \times \frac{1}{p} \times \frac{1}{n_r}$$

$$(2.10)$$

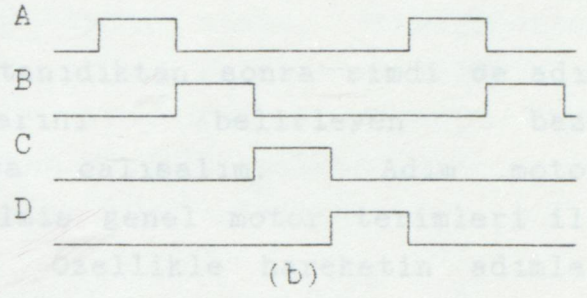
VR düzenlenebilir. Burada S, senkron hız, p, frekans, n_r rotor döngü sayısıdır. Motor döngü sayısı 50 olan bir motor için, frekans 50 Hz ise, 72 döngü başına senkron hız elde edilir.



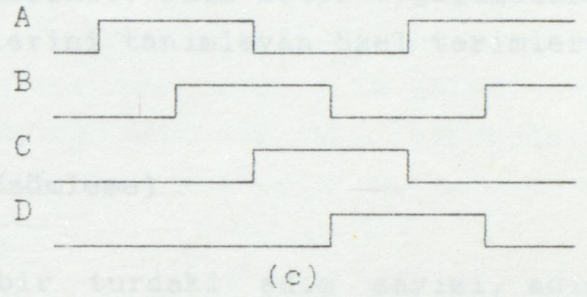
(a)



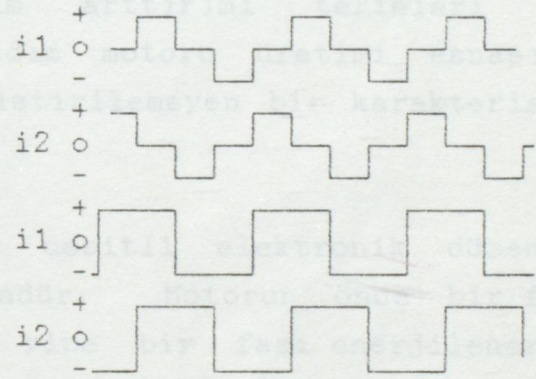
(d)



(b)



(c)



Şekil 2.14 (a) Tek yönlü sürme için sargı bağlantı düzeni

(b) Bir fazı enerjili zamanlama diyagramı

(c) İki fazı enerjili zamanlama diyagramı

(d) İki yönlü sürme için sargıların paralel bağlantı düzeni ve iki yönlü sürmeye ait zamanlama diyagramları

$$S = 60 f / N_r$$

$$(2.10)$$

ile bulunabilir. Burada S; senkron hız, f; frekans, N_r ; rotor diş sayısıdır. Rotor diş sayısı 50 olan bir motor için, frekans 60 Hz ise, 72 devir/dakikalık senkron hız elde edilir.

2.3. Adım motoru performans karakteristikleri

Adım motorlarını biraz tanıdıktan sonra şimdi de adım motorlarının performanslarını belirleyen bazı karakteristikleri tanımlamaya çalışalım. Adım motor karakteristiklerini alışlagelmiş genel motor terimleri ile incelemek mümkün değildir. Özellikle hareketin adımlar halinde olduğu denetim sistemlerinde, adım motor uygulamaları için adım motor karakteristiklerini tanımlayan özel terimlere gereksinim duyulur.

2.3.1. Adım duyarlığı (çözümleme)

Adım motor duyarlığı, bir turdaki adım sayısı, adım açısı, derece olarak adım arttırımı terimleri ile açıklanabilir. Bu terim adım motoru üretimi esnasında belirlenen ve daha sonra değiştirilemeyen bir karakteristik özelliiktir.

Adım motoru duyarlığını çeşitli elektronik düzenler kullanarak değiştirmek mümkündür. Motorun önce bir fazı sonra iki fazı daha sonra yine bir fazı enerjilenerek, motorun adım açısı elektronik olarak ikiye bölünebilir.

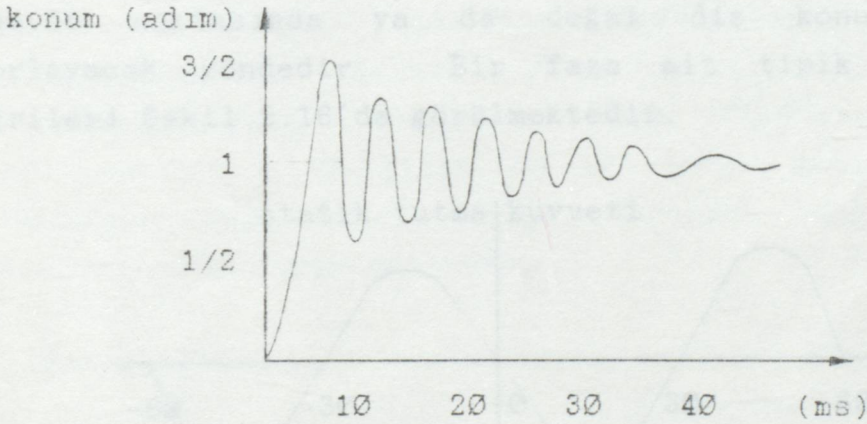
Aynı zamanda adım açısı; motorun bir fazı enerjili iken diğer fazın enerji oranı değiştirilerek de ayarlanabilir. Ancak bu işlemin duyarlı olarak yapılabilmesi için motorun özelliklerinin çok iyi bilinmesi gerekmektedir.

2.3.2. Tek adım cevabı

Motor sargılarından herhangi biri enerjilenmiş, yani motor doğal diş konumlarından birine yerleşmiş durumda iken, enerjili sargının enerjisi kesilip bir sonraki sargı enerjilenirse motor bir adım ileriye gidecektir. Motorun uzun zaman aralıkları ile bu şekilde uygulanan darbelere karşı

vereceği cevap tek adım cevabı olarak tanımlanır. Tek adım cevabı motorun bir adımı ne kadar sürede atabileceğini gösterdiği için önemli adım motor karakteristiklerinden biridir. Tek adım cevabı ile aynı zamanda motorun dişe yerleşmesi esnasında ortaya çıkan sönümlü salınım süresini ve motorun adım duyarlılığını da tespit etmek mümkündür.

Adım motorları genellikle motor fazına uygulanan anahtarlama işareti ile birlikte salınımlı olarak cevap verirler. Bu durumda salınımları azaltmak ya da engellemek için zayıflatma ya da söndürme elemanlarına gereksinim duyulur. Şekil 2.15'te tipik adım motoru için tek adım cevabı görülmektedir.



Şekil 2.15 Adım motoru için tipik tek adım cevabı

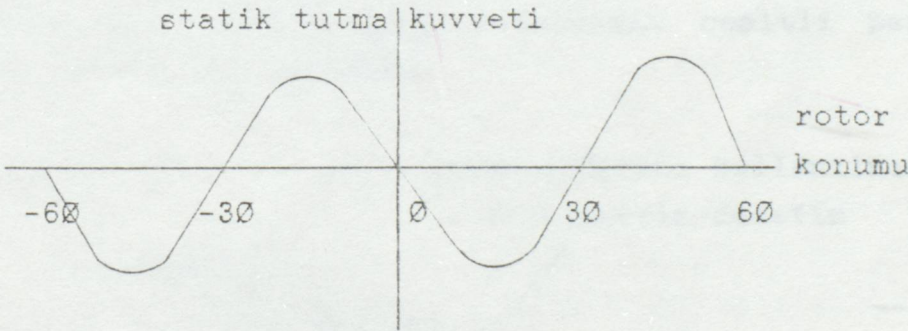
2.3.3. Duyarlılık (hassasiyet)

Adım motorunda adım konumları tasarım ve üretim esnasında birleştirilen bazı parçaların boyutları ile belirlenir. Bu parçaların boyutlarında olabilecek hatalar, motorun adım konumlarında da belli miktar hatalara neden olur. Ortaya çıkan bu adım hatası, adım motorunun duyarlılığı olarak bilinir ve herhangi bir adım konumunda, maksimum açısal hatanın normal adım açısına yüzde oranı ile ifade edilir. Normalde adım motorları için açısal duyarlık $\pm 5'$ ler

civarındadır ve bu oran özel amaçlara göre $\pm 1\%$ 'lere kadar düşürülebilir. Adım motorunun yapısından gelen hatalar adım atma işlemleri ile birlikte birikmez. Aynı zamanda sürtünmeli yükler de ilave, ancak birikmeyen konum hatalarına neden olur. Tabii bu hata motora uygulanan yükteki sürtünmeden kaynaklandığı için, motor duyarlılığı ile ilgili değildir.

2.3.4. Statik tutma kuvveti

Statik tutma kuvveti adım motorunun en temel karakteristiklerinden biridir. Tutma ve bırakma kuvveti eğrileri rotorun konum değiştirmesine karşılık motorun tutma ve bırakma kuvvetini gösterir. Tutma kuvveti rotoru sıfır kuvvet noktasında ya da doğal dış konumunda tutmaya zorlayacak yöndedir. Bir faza ait tipik tutma kuvveti eğrileri Şekil 2.16'da görülmektedir.



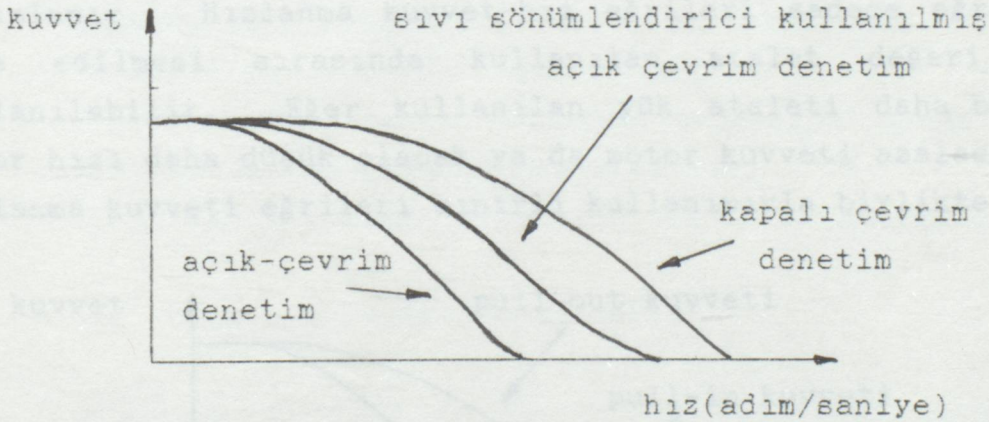
Şekil 2.16 Bir faza ait adım motoru tipik statik tutma kuvveti eğrisi

Tutma kuvveti eğrileri rotor konumu ve sargı enerjilenmesinin tüm statik şartları altında, motorun anlık kuvvetini belirlemek için gerekli tüm bilgileri içinde bulundurur. Diğer tüm kuvvet karakteristikleri (statik ya da dinamik) bu tutma kuvveti eğrilerine bağlı olarak ortaya çıkmaktadır.

2.3.5. Dinamik sürekli hız motor kuvveti (pull-out)

Giriş darbe dizisine bağlı olarak, rotor senkronizasyonunu bozmadan, adım motorunun o anki hız değerinde, motora yük olarak uygulanabilecek maksimum sürtünme kuvvetine motorun dinamik kuvveti adı verilir.

Adım motoru dinamik kuvvet eğrileri normal motorların kuvvet-hız eğrilerinden oldukça farklıdır. Adım motoru dinamik kuvvet eğrileri adım motoru için özel bir çalışma noktası belirtmez ve doğru akım motorlarında olduğu gibi motora ait bir transfer fonksiyonu bağıntısı da göstermez. Bu eğriler temel olarak, motorun isteğe uygun şekilde çalışabileceği, kuvvet-hız eğrilerini göstermektedir. Bu eğrilerin dışına çıkılmak istendiğinde motor artık çalışmayacaktır. Şekil 2.17'de çeşitli çalışma koşulları altındaki adım motoruna ait kuvvet-hız eğrileri görülmektedir. Şekilden de görüldüğü gibi motor sürülme tekniğine bağlı olarak adım motorundan çeşitli performans eğrileri elde etmek mümkündür.



Şekil 2.17 Adım motoru dinamik kuvvet eğrileri

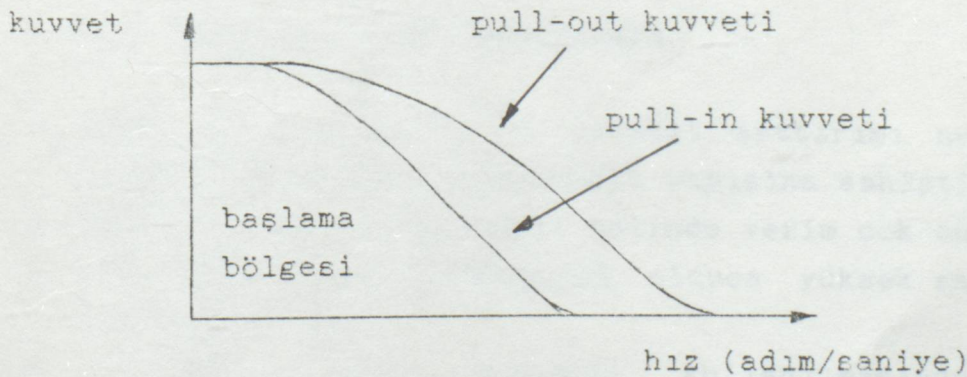
Şekil 2.17'den görüldüğü gibi rotor dönmüyorken en yüksek çıkış kuvveti elde edilmektedir. Giriş darbe hızı arttığında sargı endüktansları, faz akımlarının sürekli durumdaki değerlerine ulaşmasını engellemekte ve motor

kuvveti düşmektedir. Genelde, motor, geri besleme olmaksızın bir darbe dizisi ile sürüldüğünde, motora ait dinamik kuvvet hızla düşmektedir. Motor miline sıvı akışkanlığı yardımıyla elde edilen atalet sonunda oluşan söndürme elemanının bağlanmasıyla, salınım önlediği için motor kuvvet-hız karakteristiği iyileşmektedir. Denetleme çevrimini bir geribesleme yolu üzerinden kapatarak motor kuvvet-hız karakteristiği daha da iyileştirilebilir. Kapalı çevrimde sıvı sönümlendiricinin kayıpları da faydalı kuvvete dönüşür.

Açık çevrim denetiminde motor sürekli olarak maksimum kuvvet sınırlarının altında çalıştırılmalıdır. Bu işlem motorun hareketi esnasında adım kaybetmemesi ya da durmaması için gereklidir. Ancak kapalı çevrim denetiminde motor, maksimum dinamik kuvvetine yakın değerlerde çalıştırılabilir.

2.3.6. Dinamik hızlanma kuvveti (pull-in)

Bu kuvvet de motorun kalkışı esnasında, adım kaybetmeksizin, verilen darbe hızında, motora uygulanabilecek, en yüksek yük sürtünme kuvveti olarak tanımlanır. Hızlanma kuvvet-hız eğrileri sadece eğrilerin elde edilmesi sırasında kullanılan atalet değeri için kullanılabilir. Eğer kullanılan yük ataleti daha büyükse motor hızı daha düşük olacak ya da motor kuvveti azalacaktır. Hızlanma kuvveti eğrileri sınırlı kullanımıyla birlikte elde



Sekil 2.18 Dinamik kuvvet eğrileri arasındaki ilişki

edilmeleri de zaman alan işlemleri gerektirmektedir.

Şekil 2.18'de adım motoruna ait tipik pull-out ve pull-in kuvvet eğrileri ve bunlar arasındaki ilişki görülmektedir. Pull-out ve pull-in kuvvet eğrileri arasında kalan bölge, ölü bölge olarak adlandırılır. Bu bölge doğru tasarımlanmış bir motor için kararsız çalışma bölgesini gösterir. Bu bölgede çalıştırıldığında motor adım kaybetme eğilimi gösterir ve durur.

2.4. Adım motoru avantajları ve dezavantajları

2.4.1. Adım motoru avantajları

- * Geri besleme olmaksızın, kararlılık sorunu çıkarmadan, açık çevrim denetimi ile denetlenebilir.
- * Konumlama hataları adım sayısının artmasıyla birikmez.
- * Sayısal giriş işaretleri ile çalıştığı için sayısal devrelerle denetimi çok kolaydır.
- * Mekaniki yapıları basit olup az bir bakımla hatta hiç bakıma ihtiyaç göstermeden kullanılabilir.
- * Çalışma şartlarındaki kirlilikten etkilenmez.
- * Gerektiğinde kolayca soğutulabilir.
- * Herhangi bir zarar görmeden defalarca durup kalkma kabiliyeti ile oldukça dayanıklıdır.
- * Yapısının basitliği sebebiyle sağlam ve güvenilirdir.

2.4.2. Adım motoru dezavantajları

- * Sabit adım açısı ya da hareket arttırımı nedeniyle esnek olmayan adım çözünürlüğü yapısına sahiptir.
- * Normal sürücü kullanılması halinde verim çok düşüktür.
- * Tek adım cevabı açısından oldukça yüksek salınımlı davranır.
- * Ataleti büyük yüklerle kullanılmasında çeşitli kısıtlamalar vardır.

BÖLÜM 3

ADIM MOTOR SÜRÜCÜ DEVRELERİ VE SIRALAMA LOJİĞİ

Adım motor kullanılan düzenlerde, kullanılan güç sürücü devresi ve hatta sıralama lojiğinin çalışma şekli adım motor performansını etkileyen önemli etkenlerdir. Adım motoruna ait en yüksek ivmelenme hızı, dinamik çekme kuvveti, yavaşlama ve durma yeteneği gibi performans karakteristiklerinin yanı sıra motor ve sürücünün güç tüketimi ile birlikte verimi de kullanılan sürücü devrenin türü ile doğrudan ilişkilidir.

Adım motor sıralama lojiği temel olarak uygulanan yön ve adım darbelerine uygun olarak fazların sürülme sırasını belirleyen darbe dizisini üretir.

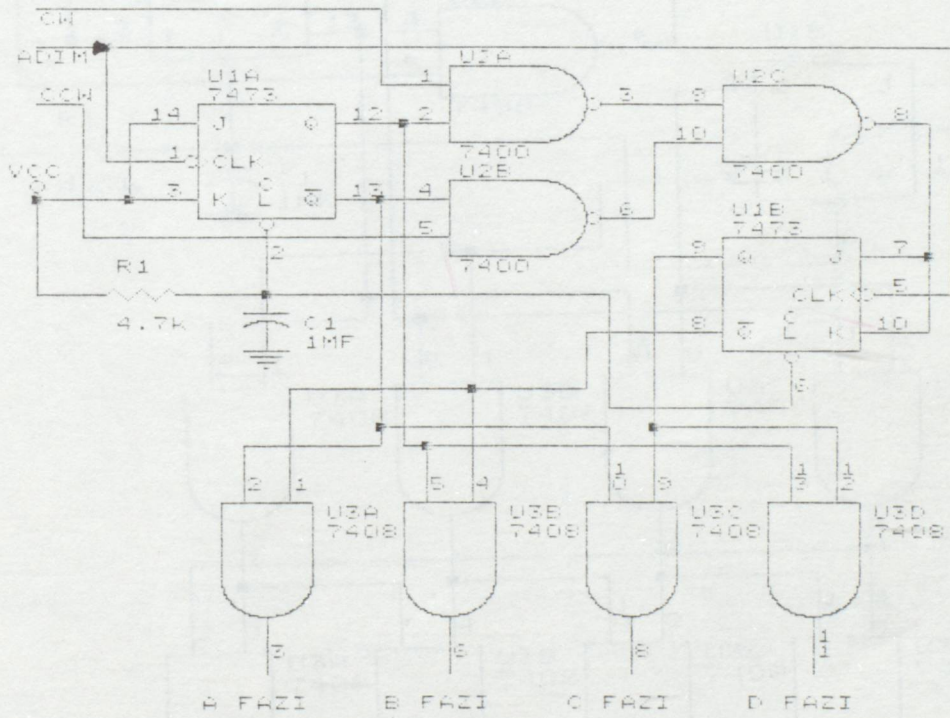
Güç sürücülerinde ise sıralama lojiğinden alınan bu darbeler motor sargılarını sürebilecek güç seviyesine yükseltilir. Güç sürücü devrelerinde fazların aşırı sürülmesini önlemek için akım sınırlama devreleri ve fazların enerjilerinin kesilmesi esnasında oluşacak yüksek gerilimleri sınırlamak için bastırma devreleri kullanılır.

3.1. Sıralama Lojiği

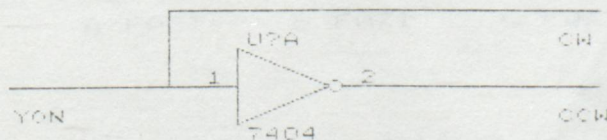
Sıralama devreleri genelde uygulanan yön bilgisine bağlı olarak adım motoru faz sürücülerine bir halka sayıcı üzerinden darbe dizileri üretir. Halka sayıcı motor faz sayısı kadar çıkış katına ve konumuna sahiptir. Sayıcıya darbe uygulanmasıyla yön bilgisine bağlı olarak çıkış katlarının birindeki aktif seviye sağa ya da sola kayarak fazların sürülme sırasını değiştirir ve motorun adım

bağlanmış RC elemanlarıyla devreye yeni enerji verilmesi durumunda sayıcının 00 konumundan başlaması sağlanır. Şekil 3.1. (b)'de bu sayıcıya ait zamanlama diyagramı verilmiştir. Şekilden de görüldüğü gibi faz enerjileme sırası A,B,C,D,A'dır.

Şekil 3.1(a)daki sıralama lojiğinde iki flip-flop arasına sayma yönünü değiştiren basit lojik kapı devresi ekleyerek kolayca iki yönlü sıralama lojiği devresi elde edilir. Şekil 3.2. (a)'daki devre Şekil 3.1. (a)'daki devrenin çift yönlü sıralama lojiğine dönüştürülmüş şeklidir. Şekildeki devrede YÖN girişi "1" iken saat dönüş yönünde (CW) "0" iken saat tersi yönde dönme elde edilir.



Şekil 3.2 (a) iki yönlü dört fazlı sıralama lojiği



Şekil 3.2 Tek uç ile yön denetimi

Şekil 3.1. ve Şekil 3.2.'deki devreler bir kerede sadece bir fazın enerjilendiği sürme işaretlelerini üretmektedirler. Enerji tüketimi fazla olsa bile, iki fazı enerjili sürme adım motorları için tercih edilen durumdur. İki fazı enerjili durumda motor adım açısı değişmemekle birlikte motor performansında kayda değer değişimler olmaktadır. Buna örnek olarak tek adım cevabı oldukça iyileşmektedir.

Şekil 3.3 iki yönlü dört fazlı, iki faz enerjili sıralama lojiği

Şekil 3.4.(a)'da minimum elemanla gerçekleştirilmiş iki yönlü ve iki fazın enerjili olduğu sıralama lojiği devresi görülmektedir. Devrede YÖN ucunda "1" varken saat yönünde (CW) dönme, "0" varken saat tersi yönde (CCW) dönme elde edilir.

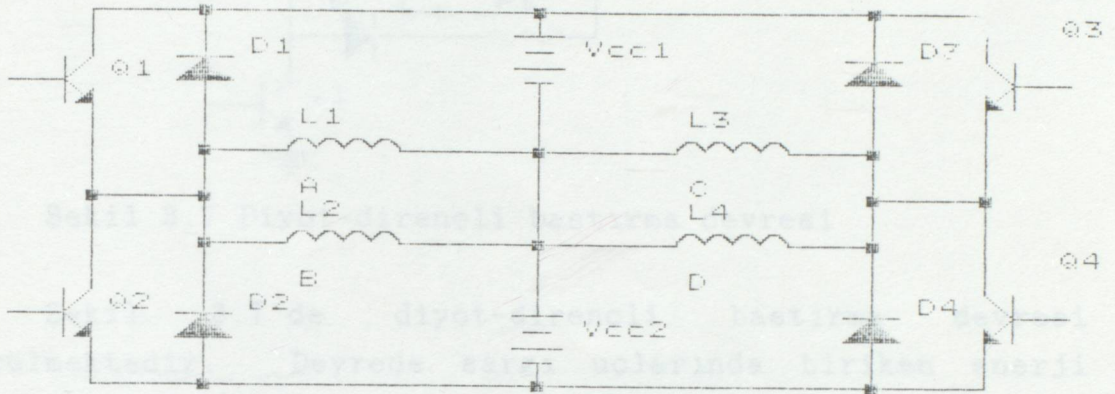
3.3 Adım Motor Sürücü Devreleri

Adım motorları genellikle alternatif akımların doğrultulmasıyla çalışan doğru akım kaynaklarından sürülür. Yarıiletken devre elemanlarının anahtar olarak çalıştığı sürücü devreler kullanılarak, adım motoru fazları, kare dalga şeklindeki akım darbeleri ile enerjilenir. Yarıiletken anahtarları sürecekle işaretler sayısal entegre devrelerden ya da mikroişlemcilerden elde edilerek yeterli seviyeye yükseltilir.

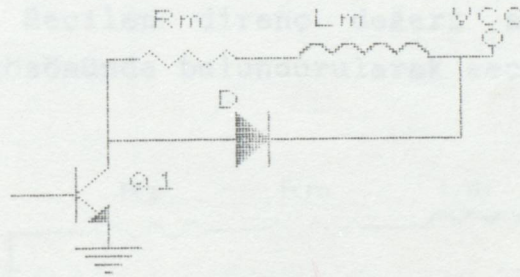
Değişken relüktanslı adım motorunda akan akımın polaritesi üretilen kuvveti deęiştirmedeęi için basit tek yönlü sürücü devresi bu motoru sürmek için yeterlidir. Genel olarak herbir faza seri bir transistör bağlanır ve transistörler açılıp kapanarak fazlara uygun enerji sağlanmış olur. Sürme işlemi gerçekleştirilirken sargı uçlarına serbest geçiş diyotları bağlanmalıdır. Serbest geçiş diyodu, sürücü transistör ile sargı enerjisi kesildiğinde, transistörü oluşacak aşırı ters endüktif gerilimlerden korur.

Şekil 3.5'te dört fazlı sabit mıknatıslı adım motoru için çift yönlü sürücü devre görülmektedir. Burada iki fazın enerjili olduğu sürme yöntemi kullanılmıştır. Devreyi süren lojik devre yine iki fazın enerjili olduğu tek yönlü sürücü devre için de kullanılabilir.

Şekil 3.6'da diyotlu bastırma devresi görülmektedir. Bu devre kullanılabilecek en basit bastırma devresidir. Fazın enerjili olduğu durumda diyot ters kutuplanmıştır ve



Şekil 3.5 Sabit mıknatıslı dört fazlı motor için iki yönlü güç sürücü devresi

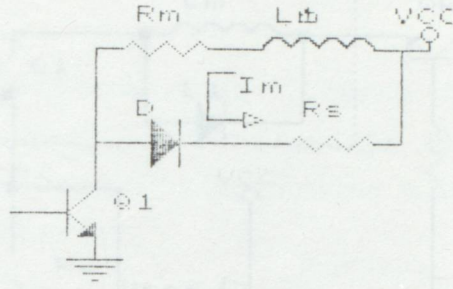


Şekil 3.6 Diyotlu bastırma devresi

açık devre etkisi gösterir. Fazın enerjisi kesildiğinde, sargı uçlarındaki gerilimin yönü değişir ve bu durumda diyot doğru yönde kutuplanarak akım için kolay bir akış yolu teşkil eder.

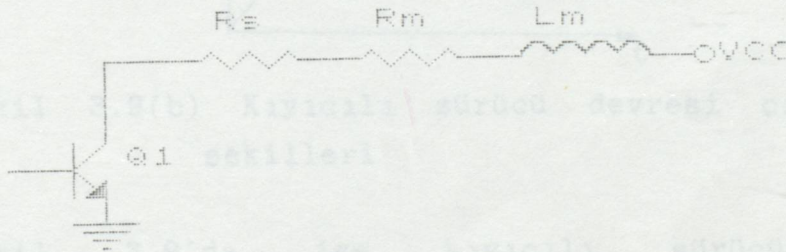
Transistörün kollektör emetör gerilimi de besleme geriliminin çok az üzerinde bir değerde tutulur.

Fazın enerjisi kesildiğinde sargılarda biriken enerji R_m sargı direnci ve serbest geçiş diyodu düz yön direnci üzerinde harcanacaktır. Diyot direnci ihmal edilirse sargı enerjisi boşalma süresi L_m/R_m ile belirlenir. Bu süre yüksek hızlar için oldukça uzundur ve motor performansını kötü yönde etkiler. Zaman sabitini kısaltmanın kolay bir yolu diyotlu geri besleme yoluna seri bir direnç ilave etmektir.



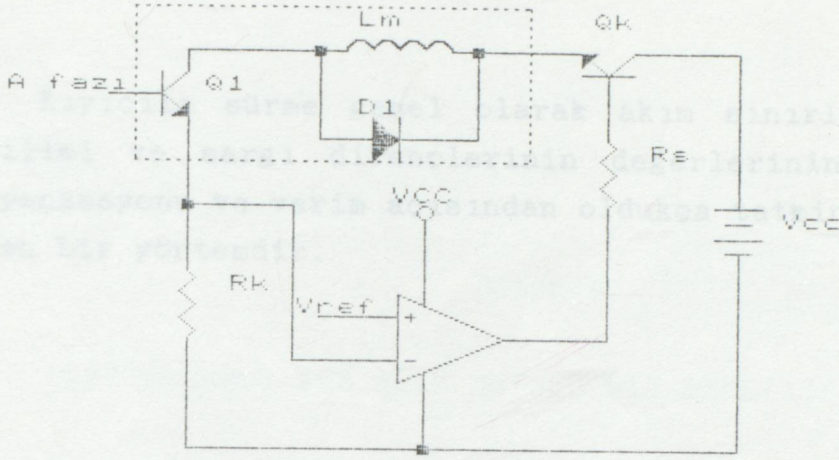
Şekil 3.7 Diyot-dirençli bastırma devresi

Şekil 3.7'de diyot-dirençli bastırma devresi görülmektedir. Devrede sargı uçlarında biriken enerji olabildiğince hızlı bir şekilde R_s direnci uçlarında harcanmaktadır. Devrenin yeni zaman sabiti $L_m/(R_s+R_m)$ 'dir. Ancak bu durumda kullanılabilecek transistörün kollektör emetör sınır gerilimi kullanılan R_s direncine bağlı olarak artmıştır. Seçilen direnç değeri sargılardan akan en yüksek akım gözönünde bulundurularak seçilmelidir.

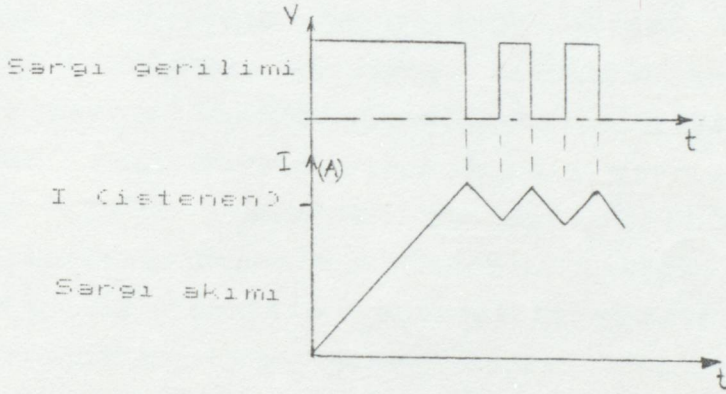


Şekil 3.8 Seri dirençli sürücü devresi

Şekil 3.8'de seri dirençli sürücü devresi görülmektedir. Bilindiği gibi adım motorunu yüksek hızlara çıkarmak için sargı akımlarının yükselme hızını arttırmak gerekmektedir. Bu işlem için zaman sabiti kısaltılmalıdır. Normal olarak sargı akımının zaman sabitini sargı endüktansı L_m ile sargı direnci R_m belirlemektedir. Eğer sargıya seri bir direnç eklenirse zaman sabiti kısaltılmış olacaktır. Aynı zamanda bu işlem bastırma devresinin karakteristiğini de düzeltebilir. Ancak bu devrenin de önemli bir dezavantajı devrede fazladan güç kaybına neden olmasıdır.



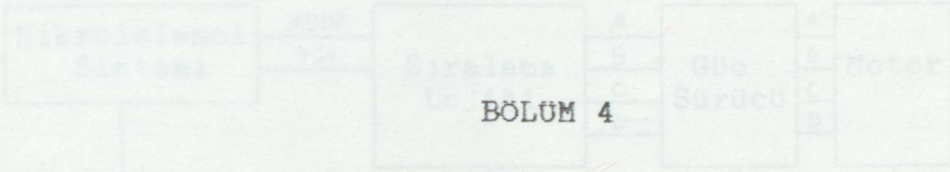
Şekil 3.9(a) Kıyıcılı sürücü devresi



Şekil 3.9(b) Kıyıcılı sürücü devresi çıkış dalga şekilleri

Şekil 3.9'da ise kıyıcılı sürücü devresi görülmektedir. Kıyıcı yöntemi de adım motorundan yüksek performans elde etmek için kullanılabilecek en uygun yöntemlerden biridir. Şekilde sargılardan yalnızca bir tanesi çizilmiştir. Herhangi bir motor fazı enerjilendiğinde Qk kıyıcı transistörü yüksek değerli besleme gerilimini açıp kapayarak sargılara uygular. Yüksek gerilim kaynağı sargılardaki normal akım yükselme hızını birkaç kat arttırır. Devrede karşılaştırmacı akım sezici dirençteki gerilimle daha önceden belirlenerek referans girişine uygulana gerilimi karşılaştırır. Motor akımı istenen değeri aşmış ise Qk kesime sokulur, akım değeri tekrar istenen değer altına düştüğünde Qk yenide doyuma sürülür. Akımın bu şekilde denetlenmesi sonucu ortalama sabit bir sürme akım değeri elde edilir.

Karakteristik Anıları: asılmalıdır. Bu durum motorun adım karakterine yada kazanmasına neden olabilir.



BÖLÜM 4

MİKROİŞLEMCI İLE ADIM MOTORU HIZ DENETİMİ

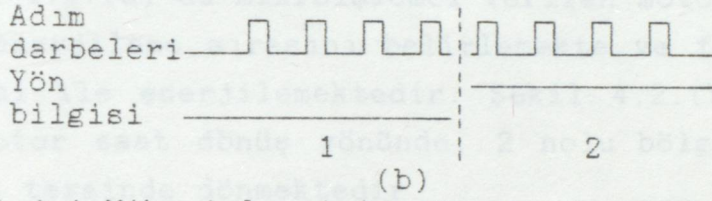
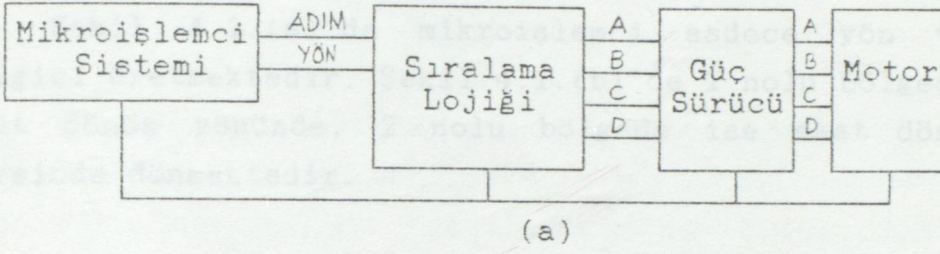
Adım motorlara kapalı çevrimle denetlenebileceği gibi, bazı koşullar sağlanarak açık çevrimle de denetlenebilir ve diğer motorların denetiminde ortaya çıkan açık çevrim denetim hataları görülmez. Açıklamak istersek adım motoru açık çevrim denetiminde dikkatli olunursa konumlama hatası olmaz. Açık çevrim olsun kapalı çevrim olsun çok çeşitli adım motor denetim yöntemi geliştirmek mümkündür. Kullanılan denetim yöntemine bağlı olarak seçilen elektronik denetim devresinin şekli belli kalıplara oturtulmakta ve daha sonra değiştirilmesi yeniden tasarlanmasından daha güç olmaktadır.

Ancak adım motor denetiminde mikroişlemci kullanılması bu sorunu büyük ölçüde ortadan kaldırmakta ve denetim olayına daha esnek bir yapı kazandırmaktadır. Denetim yöntemi değiştirilmek istendiğinde sabit yapıllı denetim devreleri ile uğraşmak yerine, mikroişlemci sisteminde yazılan programı değiştirmek daha kolaydır.

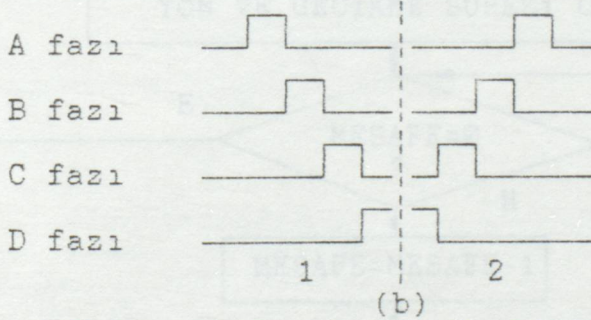
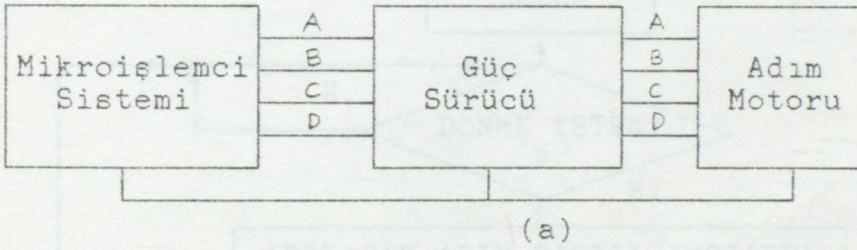
4.1 Açık Çevrim Denetimi

Adım motoru için kullanılabilecek en basit yöntem açık çevrim denetimidir. Bu yöntemde mikroişlemci sistemi adım motoru için gerekli darbe dizisini ve gerekli gecikme sürelerini üretir. Bu işlemleri yaparken de motordan herhangi bir geribesleme bilgisi almaz. Ancak darbe dizisi oluşturulurken daha önce adım motorundan elde edilen

karakteristik sınırları aşılmamalıdır. Bu durum motorun adım kaybetmesine yada kazanmasına neden olabilir.



Şekil 4.1 Mikroişlemcinin yön ve adım bilgisi ürettiği denetim devresi blok diyagramı



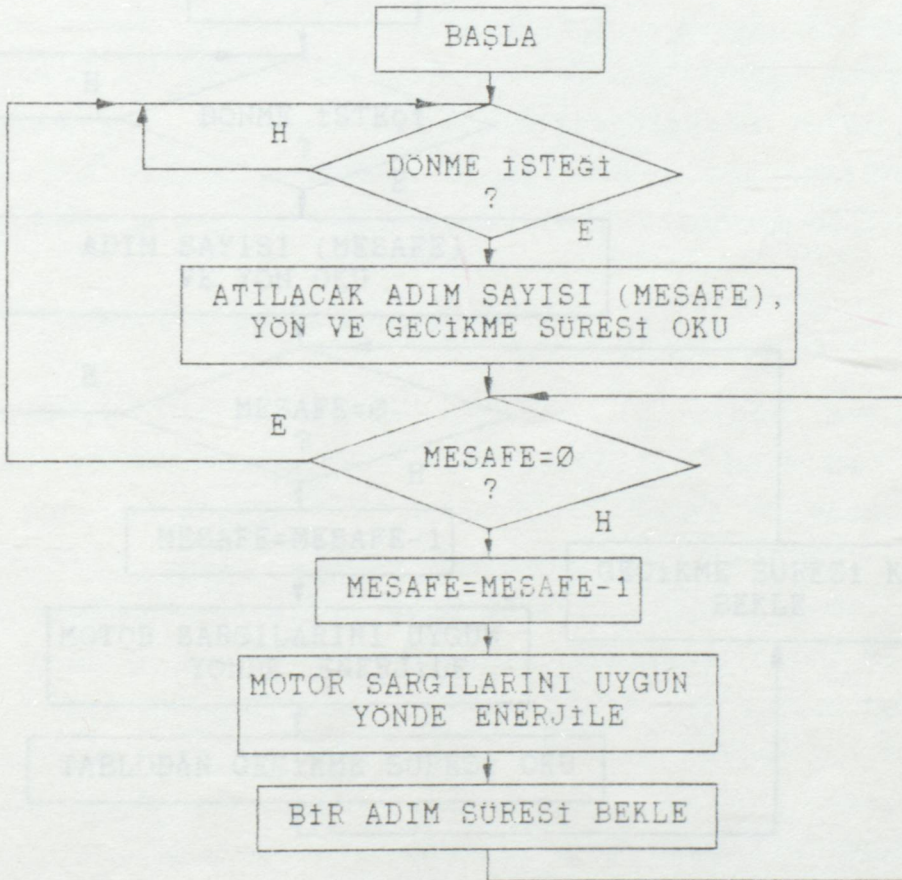
Şekil 4.2 Mikroişlemcinin yön bilgisine uygun olarak darbe dizisi ürettiği denetim devresi blok diyagramı

Mikroişlemci ile adım ve yön bilgisi ya doğrudan lojik sıralama devresine uygulanır yada mikroişlemci yön bilgisine bağlı olarak adım motoru için gerekli sıralama

işaretleri de mikroişlemci tarafından üretilerek sürücü devreye uygulanır. Şekil 4.1 ve Şekil 4.2 bu durumları göstermektedir.

Şekil 4.1.(a)'da mikroişlemci sadece yön ve adım bilgisi üretmektedir. Şekil 4.1.(b)'de 1 nolu bölgede motor saat dönüş yönünde, 2 nolu bölgede ise saat dönüş yönü tersinde dönmektedir.

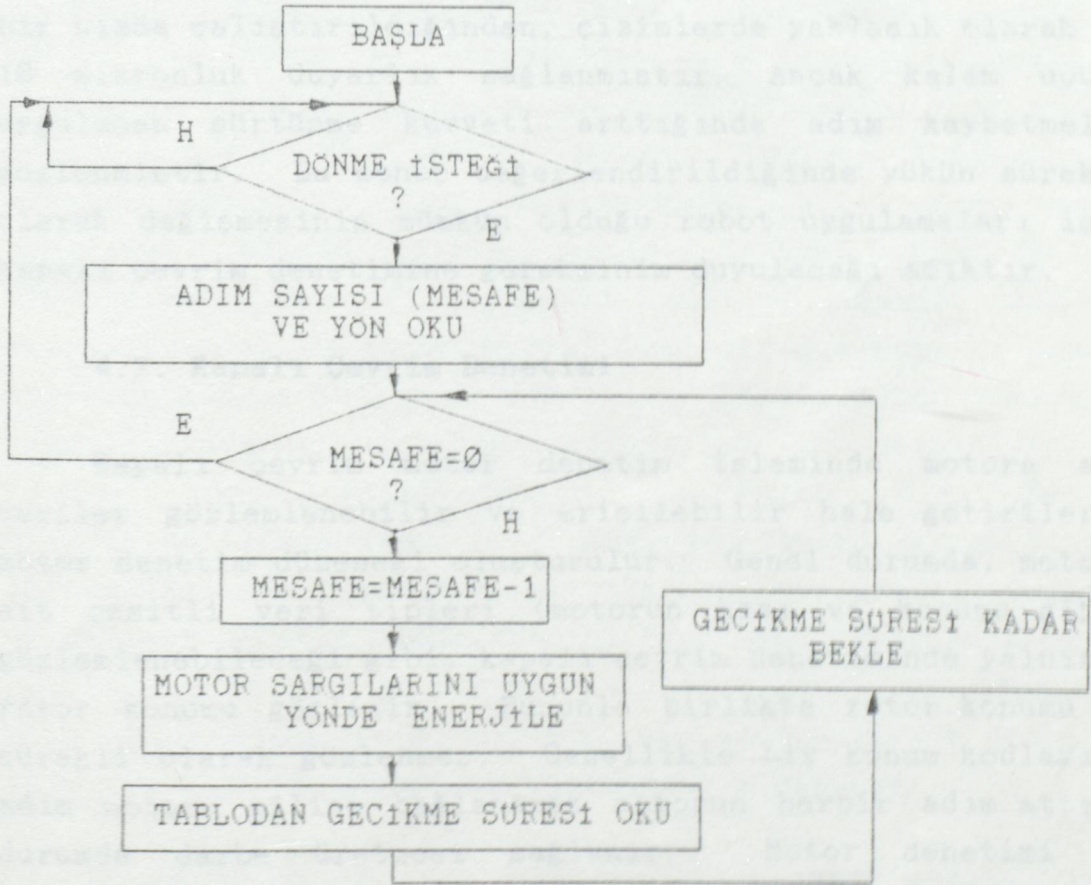
Şekil 4.2.(a)'da mikroişlemci verilen motor dönüş yönü için faz enerjileme sırasını belirlemekte ve fazları uygun T gecikmesi ile enerjilemektedir. Şekil 4.2.(b)'de 1 nolu bölgede motor saat dönüş yönünde, 2 nolu bölgede ise saat dönüş yönü tersinde dönmektedir.



Şekil 4.3 Sabit hız açık çevrim denetim algoritması

Mikroişlemci ile adım motorunu sürmek için ilk olarak atılmak istenen adım sayısı bilinmelidir. Bu adımların yönü de bilinirse motor kolayca denetlenebilir. Denetime ait algoritma Şekil 4.3'te verilmiştir.

Kullanılan bu yöntemde motor sabit hızda dönmektedir. İlk anda motor durmakta olduğu için, normal olarak bu sabit hız, motorun kabul edebileceği en küçük hız değeri olacaktır. Bu sakıncayı ortadan kaldırmak için, atılacak adım sayısına bağlı olarak, motorun hızlandırılıp yavaşlatıldığı açık çevrim denetim yöntemi kullanılmalıdır.



Şekil 4.4 Zaman gecikmesi denetimi ile ivmelenme, sabit hız, yavaşlama algoritması

Hızın denetlendiği bu yöntemde motora bir ivmelenme algoritması uygulanır. Motora herbir adım attırıldığında, adım atma süresi T , küçültülerek motorun hızlanması sağlanır. Kullanılan yöntemde motor 8-10 adım atma işlemi sonunda tatminkar bir hız değerine ulaşmaktadır. Tabii motor hızlandırıldığına göre ancak belli adım sayısında durdurulabilecektir. Öyle ise kalan adım miktarı hızlanmada kullanılan adım sayısının altına düşerse motor yavaşlatılmaya başlanmalıdır. Bu yöntemle ilişkin algoritma Şekil 4.4'te görülmektedir. Açık çevrimde hızlanma ve yavaşlama algoritmaları kullanılarak gerçekleştirilen proje çalışmasında elde edilen çizim örnekleri EK1'de verilmiştir. Burada motor adım kaybetmeye neden olmayacak kadar düşük bir hızda çalıştırıldığından, çizimlerde yaklaşık olarak 10 mikronluk duyarlık sağlanmıştır. Ancak kalem ucuna uygulanan sürtünme kuvveti arttığında adım kaybetmeler gözlenmiştir. Bu sonuç değerlendirildiğinde yükün sürekli olarak değişmesinin mümkün olduğu robot uygulamaları için kapalı çevrim denetimine gereksinim duyulacağı açıktır.

4.2. Kapalı Çevrim Denetimi

Kapalı çevrim motor denetim işleminde motora ait veriler gözlemlenebilir ve erişilebilir hale getirilerek motor denetim düzeneği oluşturulur. Genel durumda, motora ait çeşitli veri tipleri (motorun hızı ve konumu gibi) gözlemlenebileceği gibi, kapalı çevrim denetiminde yalnızca rotor konumu gözlenir. Bununla birlikte rotor konumu da sürekli olarak gözlenmez. Genellikle bir konum kodlayıcı adım motoru miline bağlanarak motorun herbir adım attığı durumda darbe üretmesi sağlanır. Motor denetimi bu darbelerin denetleyici düzenek ve yazılım yardımıyla gözlenmesi sonucu gerçekleştirilir.

4.2.1. Kapalı Çevrim Denetiminde Konum Kodlayıcı

Burada öncelikle konum kodlayıcının nasıl bir işaret

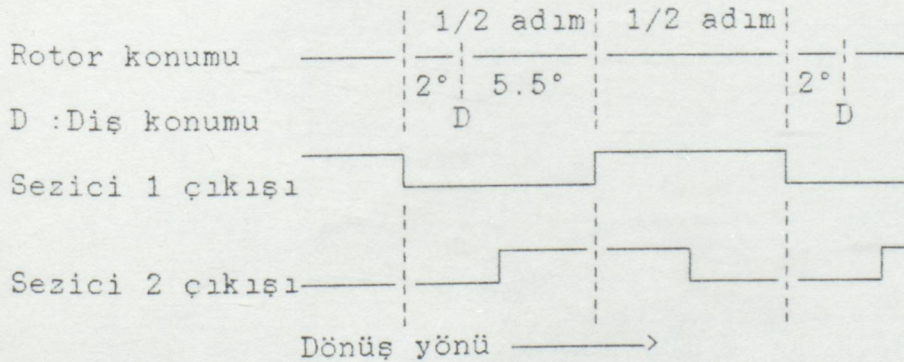
ürettiğinin ve bu işaretlerin mikroişlemciye nasıl gönderileceğini ele almak gerekmektedir.

Konum kodlayıcı disk öncelikle rotor miline monte edilir ve çıkışında "0" ya da "1" lojik seviye vardır. Rotor döndükçe kodlayıcı disk de dönerek sezici elemanların (Transducer) yarıklardan, yani aydınlık-karanlık veya karanlık-aydınlık bölgelerden geçişinde, 1-0 veya 0-1 geçişleri üretmesine neden olur. Konum kodlayıcı, her bir motor adımının oluşması esnasında tam sayıda geçişler üretecek şekilde tasarlanır. Kodlayıcıların çoğu her bir motor adımına karşılık tek bir 1-0 geçişi oluşturacak şekilde tasarlanırlar. (Tabii bir sonraki geçiş bir öncekinin tersi olarak 0-1 geçişi olacaktır). Genelde kullanılan durum ve mikroişlemcilerin sınırlı hesaplama süreleri nedeniyle bundan sonra konum kodlayıcı için her bir adımda bir tek darbe üreten konum kodlayıcılar anlaşılacaktır. Eğer kodlayıcıdan bir motor adımı esnasında birden fazla geçiş darbesi üretilseydi, mikroişlemci daha fazla sayıda bilgiyi işlemek zorunda kalacaktı. Bu da mikroişlemci için oldukça değerli zamanı gereksiz olarak kullanmak demektir. Yüksek motor hızlarına ulaşıldığında her bir adım ortalama 1ms'lik zaman dilimleri içinde gerçekleştirilecektir. Bugün bile mevcut mikroişlemcilerin çoğu özellikle denetlenen motor sayısı ikinin üzerinde ise motora ait tüm konum bilgisini okuyacak ve gerekli denetim algoritmasını yürütecek uygun hıza sahip değildir. 8 bitlik mikroişlemcilerin çoğu ise, tek bir adım motoru için bile, duyarlı bir kapalı çevrim denetimi için, bu hıza ve hesaplama yeteneğine sahip değildir.

Konum kodlayıcı 1-0 geçişlerinin meydana geldiği uzay durumları ayarlanabilecek şekilde tasarlanır, öyle ki rotor doğal dış konumuna yerleştikten önce veya sonra özel bir açıda 1-0 geçişi sağlanacak şekilde konum ayarlanır. Genellikle konum kodlayıcılarda 1-0 geçişi ile 0-1 geçişi arasında yarım adımlık fark vardır.

4.2.2. Rotor Dönüş Yönünün Belirlenmesi

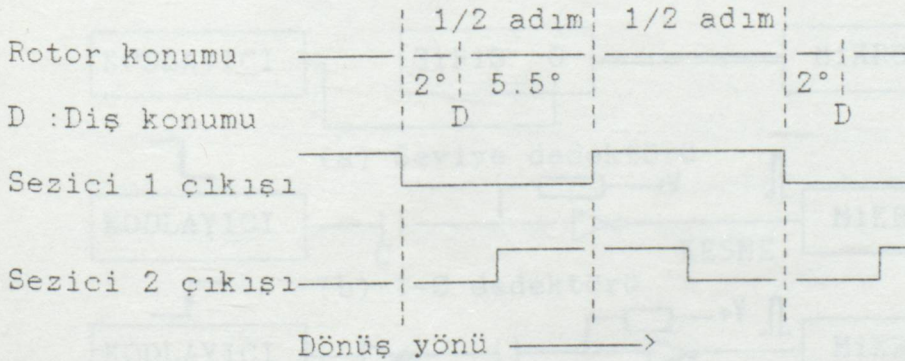
Rotor dönüş yönünü belirlemek için genellikle ikinci bir sezici eleman (transducer) diğerine dik uzayda yada diğer bir anahtarlama açısında yerleştirilir. Şekil 4.5'te birbirlerine göre 90 derece faz açısı ile yerleştirilmiş iki sezici elemana ait çıkış dalga şekilleri görülmektedir. Burada kullanılan motorun 15 derecelik adımlara sahip olduğu ve bir turda 24 adım attığı varsayılmıştır. Buna göre sezici elemanların herbiri rotor dönüşünün her 15 derecesine karşılık bir adet 1-0 geçişi üretir. Şekil 4.5'te birinci sezici elemanın 1-0 geçişinin motor gösterilen yönde dönerken, rotor doğal dış konumuna yerleşmeden 2 derece önce meydana geldiği varsayılmıştır. (Rotor yönü saat yönü doğrultusu (CW) kabul edilebilir.) Eğer rotor diğer yönde dönüyor olsaydı (CCW) birinci sezici elemanın 1-0 geçişi rotorun doğal dış konumuna yerleşmesinden 5.5 derece önce gerçekleşecekti. Rotor yönünü belirleyebilmek için ikinci bir sezici eleman şekildeki gibi birinci sezici elemanın 1-0 geçişlerinin uzay durum olarak ortasında geçiş üretecek şekilde yerleştirilmiştir. Yönün nasıl belirlendiğini anlamak için rotorun ilk önce Şekil 4.5'te gösterilen 1 nolu konumun solunda olduğunu varsayalım. 1 nolu konumda birinci elemanda 1-0 geçişi olur ve bu anda ikinci eleman lojik "0"



Şekil 4.5 Sezicilerin dik olarak yerleştirildiği kodlayıcı çıkış dalga şekilleri

4.2.2. Rotor Dönüş Yönünün Belirlenmesi

Rotor dönüş yönünü belirlemek için genellikle ikinci bir sezici eleman (transducer) diğerine dik uzayda yada diğer bir anahtarlama açısında yerleştirilir. Şekil 4.5'te birbirlerine göre 90 derece faz açısı ile yerleştirilmiş iki sezici elemana ait çıkış dalga şekilleri görülmektedir. Burada kullanılan motorun 15 derecelik adımlara sahip olduğu ve bir turda 24 adım attığı varsayılmıştır. Buna göre sezici elemanların herbiri rotor dönüşünün her 15 derecesine karşılık bir adet 1-0 geçişi üretir. Şekil 4.5'te birinci sezici elemanın 1-0 geçişinin motor gösterilen yönde dönerken, rotor doğal dış konumuna yerleşmeden 2 derece önce meydana geldiği varsayılmıştır. (Rotor yönü saat yönü doğrultusu (CW) kabul edilebilir.) Eğer rotor diğer yönde dönüyor olsaydı (CCW) birinci sezici elemanın 1-0 geçişi rotorun doğal dış konumuna yerleşmesinden 5.5 derece önce gerçekleşecekti. Rotor yönünü belirleyebilmek için ikinci bir sezici eleman şekildeki gibi birinci sezici elemanın 1-0 geçişlerinin uzay durum olarak ortasında geçiş üretecek şekilde yerleştirilmiştir. Yönün nasıl belirlendiğini anlamak için rotorun ilk önce Şekil 4.5'te gösterilen 1 nolu konumun solunda olduğunu varsayalım. 1 nolu konumda birinci elemanda 1-0 geçişi olur ve bu anda ikinci eleman lojik "0"



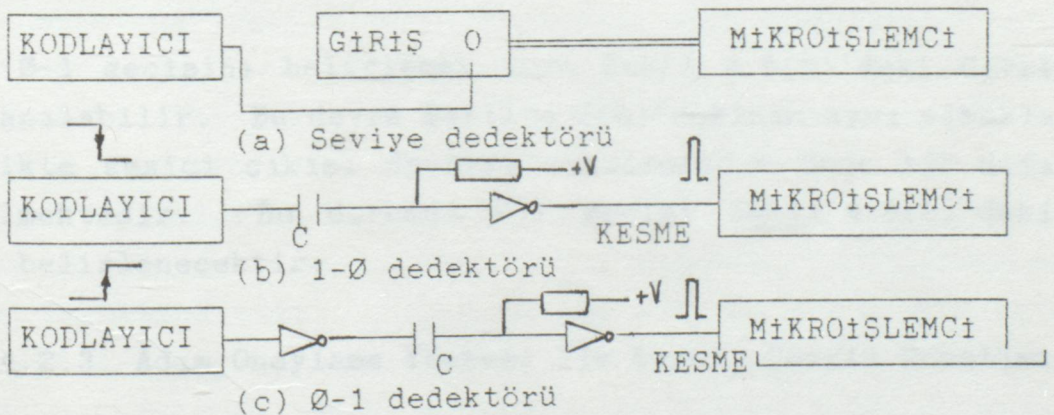
Şekil 4.5 Sezicilerin dik olarak yerleştirildiği kodlayıcı çıkış dalga şekilleri

çıkışına sahiptir. Şimdi de rotorun 2 nolu konumun solunda olduğunu ve sola doğru döndüğünü varsayalım. 2 nolu konumda birinci elemanda 1-0 geçişi olur ve bu anda ikinci eleman lojik "1" çıkışına sahiptir. Böyle bir çalışma şekline göre motor dönüş yönünü belirlemek mümkündür. Bunun için önce birinci elemanın 1-0 geçişi beklenir, geçiş meydana geldiğinde ikinci elemanın çıkışı okunur. Bu durumda okunan seviye "0" ise rotor dönüş yönü saat dönüş yönü CW ve okunan seviye "1" ise rotor dönüş yönü saat dönüş yönü tersi CCW olarak tespit edilir. Tabii birinci elemanın 1-0 geçişleri yerine 0-1 geçişleri izlenirse olayın aksi düşünüülerek motor dönüş yönü için karar verilebilir.

Yukarıdaki örnekten de anlaşılabileceği gibi yön belirleme işleminde hem lojik seviyelerin (1 veya 0) hem de lojik geçişlerin (1-0 veya 0-1) belirlenmesi gerekmektedir.

Bu işaretlerin sezilmesinde ve mikroişlemciye aktarılmasında kullanılan tekniklerin bazıları Şekil 4.6'da gösterilmiştir.

Şekil 4.6(a)'da sezici eleman çıkışı giriş birimlerinin bir bitine uygulanır. Mikroişlemci girişin içeriğini okuyup sezici elemanın durumunu belirleyebilir.



Şekil 4.6 Kodlayıcıların mikroişlemciye bağlanması

Şekil 4.6(b)'de 1-Ø geçişlerinin nasıl belirleneceği gösterilmiştir. Şekilde görüldüğü gibi sezici eleman kapasite üzerinden bir evirici ile bağlanmıştır. Aynı zamanda evirici girişi bir direnç ile lojik "1" seviyeye çekilmiştir. 1-Ø geçişi meydana geldiğinde kapasitenin solundaki gerilim "Ø" seviyeye çekilir. Kapasite uçlarındaki gerilim birden değişemeyeceği için kapasitenin sağındaki gerilim değeride "Ø" seviyeye çekilecektir. Yani evirici girişi 1-Ø geçişi gördüğünde çıkışta Ø-1 geçişi görülecektir. Daha sonra kapasite RC zaman sabiti ile tekrar V gerilimine dolacağı için, belli bir süre sonra evirici girişi tekrar lojik "1" seviyeye ulaşacaktır. Buna bağlı olarak da çıkış tekrar "Ø" seviyeye düşecektir. Böylece sezici elemenda 1-Ø geçişine karşılık evirici çıkışında Ø-1-Ø geçiş dizisi üretilecektir.

Bu işaret mikroişlemci kesme girişine uygulanabilir veya bir D flip-flop ile tutularak mikroişlemci için uygun bir zamanda okunacak şekilde saklanır. Bu devrede Ø-1 geçişleri dikkate alınamamaktadır. Bu aslında evirici girişinin başlangıçta lojik "1" seviyede tutulmasından kaynaklanmaktadır. Yani Ø-1 geçişi oluştuğunda kapasite fazladan bir V gerilimine doldurulmak istenir. Ancak bu gerilim evirici için hala lojik "1" seviye anlamına gelmektedir. Öncekinden daha yüksek olan bu giriş gerilimi RC zaman sabiti ile belirlenen oranda azalacaktır ve V değerine ulaşacaktır.

Ø-1 geçişini belirlemek için Şekil 4.6(c)'deki devre kullanılabilir. Bu devre Şekil 4.6(b)'dekinin aynı olmakla birlikte sezici çıkışı devreye uygulanmadan önce bir defa evrilmektedir. Bu durumda Ø-1 geçişi Şekil 4.6(b)'deki gibi belirlenecektir.

4.2.3. Adım Onaylama Yöntemi ile Kapalı Çevrim Denetimi

Adım motorlarının açık çevrim denetiminde en önemli problem senkronizasyonun kaybolması olasılığıdır. Senkronizasyon

kaybı adım atma sıralamasının sonunda n fazlı motor için $n*N$ adım kazancı olarak ortaya çıkan durum (N tamsayı) veya motor sabit hızla bir noktadan diğer bir noktaya hareket ederken bazı adımların kaybolması olarak ortaya çıkan durum şeklinde açıklanabilir. (Yani motor hareket etmez ancak denetim devresi bu durumun farkında değildir). Aynı zamanda

Motorun adım kaybetmemesi veya adım kazanmaması için hatta geçici senkronizasyon kaybı bile olmaması için en uygun yöntem kapalı çevrim denetim yöntemi olarak bilinen adım onaylama yöntemidir. Adım onaylama yöntemi temel olarak rotor tarafından atılan herbir adımın tespit edilmesi ve sayılması şeklinde açıklanabilir. Bu işlem de rotor miline bağlanan konum kadlayıcı çıkış işaretlerinin gözlenmesiyle gerçekleşir. Adım atma işleminin herhangi bir aşamasında beklenen adım sayısı ile bu sayım sonucu karşılaştırılabilir.

Adım sayısı ile sayım sonucu aynı ise adım sayısı kabul edilir ve olayların akışının normal olduğuna karar verilir. Tabii bu durumda adım atma işlemlerine devam edilebilir. Bununla birlikte sayım sonucu ile denetleyici değerleri uyum içinde değilse, ortada bir sorun olduğuna karar verilir ve bu durumun düzeltilmesi için gerekli işlemler yapılır.

Adım onaylama işlemini açıklamak için önce sadece adım sıralaması sonunda ortaya çıkacak adım atma hataları ile ilgilendiğimizi varsayalım. Buna göre motorun hızlanma ve sabit hızla ilerleme fazlarında adım kaybetmediğini yani motorun durmadığını varsaymış oluyoruz. Varsayımımıza göre motor durma ya da yavaşlama fazı esnasında adım kazanmakta veya kaybetmektedir. Sonuç olarak da, en son durum için istenen konumda bulunmayacaktır. Bu varsayıma göre motorun adım hatalarının adım sıralama işleminin sonuna kadar kontrol edilmesine gerek yoktur. Böylece düzeltme işlemi adım atma işleminin sonuna kadar gerçekleştirilmeyecek

ancak adım atma işlemi bittikten sonra devreye girecektir. Bu tip adım onaylamaya ilişkin programa ait akış diyagramı Şekil 4.7'de verilmiştir.

Kolaylık sağlaması açısından arzu edilen dönüş yönünün saat dönüş yönü olduğunu kabul edelim. Aynı zamanda darbeler arasındaki zamanlamanın program içinde belirlendiği ve her defasında tekrar tanımlanmasına gerek olmadığını varsayalım. Bu durumda kullanıcı arzu edilen dönüş yönünü ve gidilecek mesafeyi belirtmekte ve hareketi başlatmak için kesme butonuna basmaktadır. Mikroişlemci açık çevrim denetiminde olduğu gibi fazları enerjiler, ancak kodlayıcıdan gelen bilgilere göre, atılan adım sayısına karar verir. Katedildiği belirlenen mesafe, asıl açık çevrim sıralamasının tamamlanmasından sonra, arzu edilen mesafe ile karşılaştırılır. Eğer elde edilen değerler uyuşmuyorsa bir düzeltme işlemi gerekmektedir. Şimdi Şekil 4.7'de verilen akış diyagramının detaylı olarak neleri gerçekleştirdiğini inceleyelim.

Başlangıçta önkoşullama komutlarından sonra program başla adresi civarında sonsuz bir bekleme çevrimine girer. Kesme butonuna basıldığında mikroişlemci kesme altprogramına dallanacaktır. Bu altprogram ana programı başlatmaktadır, burada kesme kabulü yeniden işleme sokulur. Böylece bundan sonra kodlayıcılardan gelen kesme istekleri dikkate alınacaktır. Programda ilk önce gidilecek mesafe ve motor için dönüş yönü okunur.

Ana programın çalışması esnasında kodlayıcılardan rastgele zamanlarda kesmeler gelmektedir. Bu kesmelerin nasıl işlendiğinin anlaşılması oldukça önemlidir. Aşağıda anlatılanlarda sezici elemanların Şekil 4.5'teki gibi verleştirildiği varsayılmıştır.

Rotor ilk olarak doğal dış konumlarından birine yerleşmiştir. Bu durumda birinci sezici eleman ve ikinci

sezici eleman çıkışında lojik "0" değeri vardır. Mikroişlemciye birinci elemanın herbir 0-1 ve 1-0 geçişinde kesme gelecektir. Motor ilk olarak dönmeye başladığında mikroişlemciye 1-0 geçişli kesme gelecektir. Eğer rotor normal dönüyorsa, ikinci elemanda birinci elemanın 0-1 geçişi için lojik "1" seviye görülecektir. Bundan sonra birinci elemanda "1" seviye olduğundan, bir sonraki geçiş 1-0 geçişi olacaktır. Şimdi iki farklı rotor dönme hareketi olasılığı vardır. Eğer rotor saat yönünde dönüyorsa birinci elemanın 1-0 geçişinde ikinci elemanda "0" seviye görülecektir. Bununla birlikte rotor senkronizasyonu bozularak rotor tekrar eski dış konumuna dönerse ikinci elemanda, birinci elemanın 1-0 geçişi için "1" seviye olacaktır. Verilen rotor dış konumu için osilasyon oluşması yani tekrar aynı dış üzerine yerleşilmesi durumu, birinci elemanın 1-0 yada 0-1 geçişlerinde, ikinci elemanın çıkışında sürekli aynı seviyenin görülmesiyle karakterize edilebilir.

Saat dönüş yönünde adım atma işleminde iki farklı geçiş esnasında ikinci elemanda farklı seviyeler olacaktır. Yani 1-0 geçişinde "0" değeri olacaktır. Saat yönü tersi adım atma işleminde de birinci elemanın iki farklı geçişi için ikinci elemanda iki farklı seviye olacaktır, yani 1-0 geçişinde saat yönünde dönmeyen tersi olarak "1" seviye olacaktır. Yapılan bu analize göre rotor dönüş yönünün belirlenmesi için aşağıdaki işlemler yapılabilir.

Birinci elemanın 0-1 geçişinde ikinci elemanın değeri okunur ve saklanır. Daha sonra meydana gelecek 1-0 geçişinde ikinci eleman tekrar okunur. Bu durumda ikinci elemanın eski değeri ile yeni değeri aynı ise, osilasyon ortaya çıkmış demektir ve motor dış konumunu değiştirmeyerek adım atmamıştır. Bu durumda 1-0 ve 0-1 geçişleri sayılmaz. Bununla birlikte ikinci elemandan okunan yeni değer eski değerden farklı ise, motorun bir adım atması sağlanmıştır. Atılan bu adım saat yönünde yada

saat tersi yönde olabilir. Bunu belirlemek ikinci elemanın yeni okunan değerine bakılır. Bu değer "1" ise motor saat dönüş yönü aksinde (CCW) bir adım atmıştır, eğer değer "0" ise motor saat dönüş yönünde bir adım atmıştır. Böylece duruma göre onaylanan adım sayacı değerine "1" eklenir veya çıkarılır. Bu işlem sırasıyla atılan toplam adım sayısı tespit edilebilir.

Sonuç olarak alınan yol, toplam alınması gerekli uzunluğa eşit değilse, bir düzeltme işlemi gerekmektedir. Onaylanan adım sayısından gereken adım sayısının çıkarılması sonucu, elde edilen sayı negatif ise, motorun adım kaybettiği anlaşılır, yani motora saat dönüş yönünde adım attırılmalıdır.

Bu işlem herbirinde bir saat dönüş yönü darbe uygulanan 100 ms'lik bekleme çevrimleri kullanılarak elde gerçekleştirilebilir. 100 ms süre motor için beklenebilecek en büyük süredir, yani motor bu süre içinde kesinlikle bir adım atabilmektedir. Bu işlemde de herbir adım attırma işleminden sonra, motorun adım atıp atmadığının test edilmesi gerekir. Test işlemi kesme sonucunda motorun bir adım atması beklenerek sağlanabilir. Programda onaylanan adım sayısı atılmak istenen adım sayısına eşit olduğunda ana programa dönülür. Eğer tespit edilen adım sayısı atılmak istenen adım sayısından fazla ise, motor fazladan adım atmış yani adım kazanmış demektir. Bu durumda ise saat dönüş yönü tersi altprogramına gidilerek onaylanan ve arzu edilen adım sayıları eşitleninceye kadar program çalışır. Adım onaylama yöntemi ile geçici senkronizasyon bozuklukları belirlenebilmekte ve düzeltilebilmektedir.

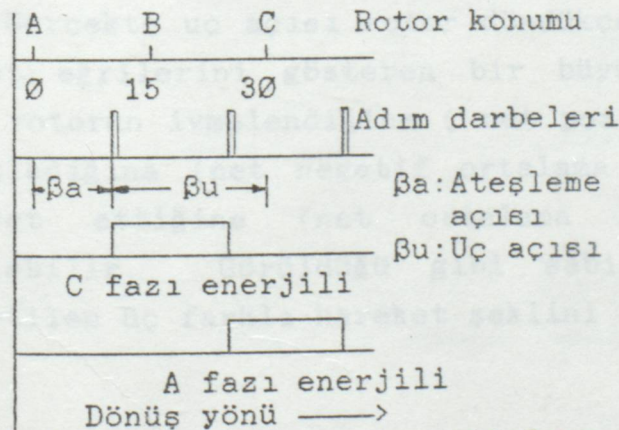
Görüldüğü gibi yukarıda anlatılan yöntemde senkronizasyon kaybı ancak faz anahtarlama sırası bittikten sonra tespit edilmektedir. Ancak onaylanan adım sayısı ile istenen adım sayısının adım atma işleminin herhangi bir anında kontrol edilebileceği açıktır. Örnek olarak adım

kaybetme motorun ivmelenme fazı esnasında ortaya çıkarsa, atılan adım sayısı ile beklenen adım sayısı arasındaki fark gittikçe büyüyecektir. Bu nedenle önceden yapılan bir kontrol işlemi bu sorunu ortadan kaldıracak yada motorun yeniden ivmelenmesi sağlanacaktır. Belki de motora daha düşük eğimli bir ivmelenme eğrisi uygulanacaktır. Sonuç olarak adım onaylama yöntemi ile çok farklı hatalı adım belirleme ve düzeltme algoritmaları geliştirilebilir.

4.2.4 Kapalı Çevrim Denetiminde Uç Açısı ve Önemi

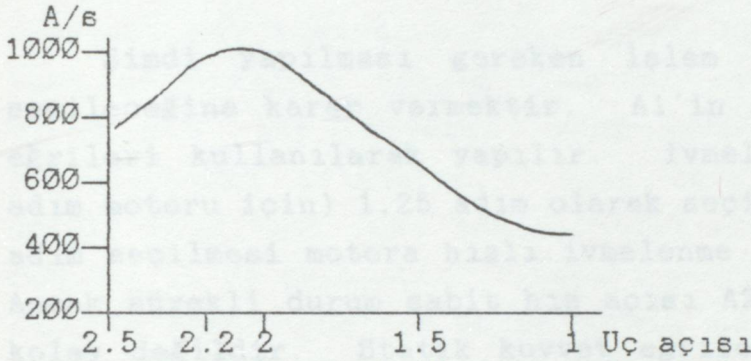
Adım motoru için ateşleme açısı dendiğinde kodlayıcıdan darbe geldiği andaki konum ile fazların enerjilendiği konum arasındaki mesafe akla gelmektedir. Uç açısı kavramı da ateşleme açısı ile ilişkili olduğu halde, konum kodlayıcının kullanılmasına bağlı değildir. Kısaca uç açısı dendiğinde, ilişkin fazın enerjilendiği andaki rotor konumu ile rotorun enerjilenen faza ait denge konumuna geldiği (doğal dış konumu) nokta arasındaki açısal mesafe anlaşılır. Bu durum Şekil 4.8'de daha açık görülmektedir.

Şekilden de görüldüğü gibi uç açısı C sargısının enerjilendiği andaki rotor konumundan, rotor dişlerinin C fazına ait dişlerle çakışacağı konuma kadar olan mesafedir. Şekilde açıkça görüldüğü gibi sargıların enerjileme



Şekil 4.8. Uç açısı ve ateşleme açısı

işleminin, rotor doğal diş yerleşme konumlarında iken yapılması şartı yoktur. Adım motoru kapalı çevrim denetiminde, uç açısının önemini göstermek için, Şekil 4.9,daki sürekli durum hızı ile uç açısı arasındaki ilişkiyi veren tipik eğriden faydalanılabilir. Şekilde görüldüğü gibi,



Şekil 4.9. Tipik Uç açısı-Sürekli durum hız eğrisi

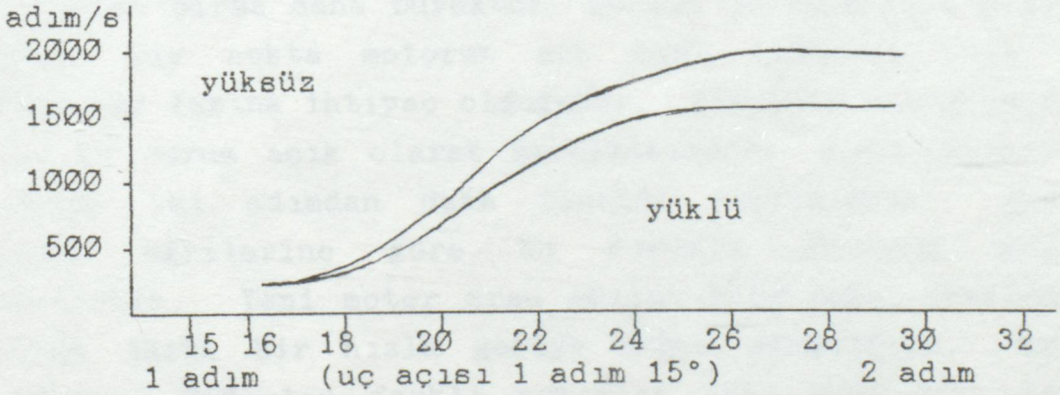
adım motoru en yüksek hıza 2.2 adımlık uç açısında ulaşmaktadır. 2 adımdan daha büyük uç açıları ise negatif ateşleme açısı demektir. Negatif ateşleme açılarında motorun ters yönde dönmesini önlemek için ivmelenme işleminden sonra darbe ejeksiyonu yapılmalıdır. Darbe enjeksiyonu, belli bir hız değerine ulaşıldıktan sonra motora peşpeşe iki adım darbesi uygulayarak yapılır.

4.2.5. Uç Açısı Seçerek Kapalı Çevrim Denetimi

Gerçekte uç açısı rotor döndükçe rotorun karşılaştığı kuvvet eğrilerini gösteren bir büyüklüktür. Uç açısına göre rotorun ivmelendiğine (net pozitif ortalama kuvvet), yavaşladığına (net negatif ortalama kuvvet), sabit hızla hareket ettiğine (net ortalama sıfır kuvvet) karar verilebilir. Görüldüğü gibi sabit uç açısı yukarıda belirtilen üç farklı hareket şeklini ortaya koymaktadır.

Böylelikle sabit uç açısı ivmelenme için kullanılırken, aynı zamanda frenleme işlemi için de kullanılabilir. Uç açısının ivmelenme açısından frenleme açısına geçirilmesi elektronik olarak darbe enjeksiyonu metodu ile aynı fazın peşpeşe enerjilenmesi sonucu gerçekleştirilir. Bu işlemden sonra tekrar sezici eleman 1 gözlenmeye başlanır. Artık A1 açısı sabit frenleme açısıdır.

Şimdi yapılması gereken işlem A1 ve A2'nin nasıl seçileceğine karar vermektir. A1'in seçimi statik kuvvet eğrileri kullanılarak yapılır. ivmelenme açısı (üçfazlı adım motoru için) 1.25 adım olarak seçilebilir. A1'in 1.25 adım seçilmesi motora hızlı ivmelenme yeteneği kazandırır. Ancak sürekli durum sabit hız açısı A2'nin seçimi A1 kadar kolay değildir. Statik kuvvet eğrilerinin dinamik kuvvet eğrilerinden farklı olmasına neden olan bobinlerin endüktans etkileri, adım motor modellerinin lineer olmaması ve analiz işlemlerinin çok olması nedeniyle A2'yi teorik yollardan hesaplamak hemem hemen mümkün değildir. Pratikte motorun son ortalama hızına karşılık uç açısının değişim eğrisi çizilerek, bu çizimlerden A2'nin yaklaşık tahmini yoluna gidilir. Şekil 4.10'da örnek bir eğri görülmektedir.



Şekil 4.10 Adım motoru için tipik hız-uç açısı eğrileri (Yükli ve yüksüz durumda)

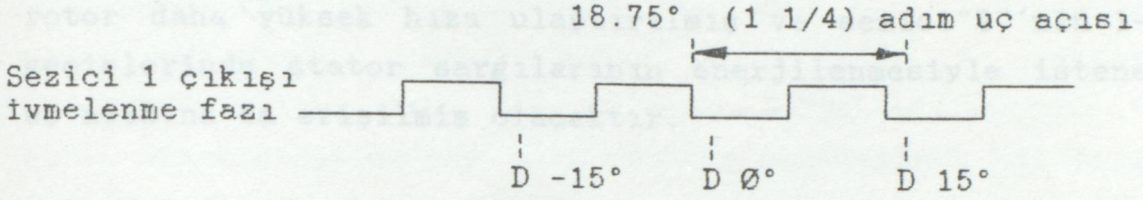
Bu yöntemdeki en önemli sorun, Şekil 4.10'dan da anlaşılacağı gibi, motora bağlanan yükün motor ortalama hızını etkileyerek zayıflatmasıdır. Rotorla bağlanan ilave bir yük değeri için, sabit uç açısı ile motor, ortalama net kuvvetin sıfır olduğu hız değerine kadar ivmelenenecektir. Bu anda yük değerinde bir değişme olursa sistemin dengesi bozulacak ve sistem yeniden denge durumuna ulaşmak için ortalama net kuvveti sıfırlayınca kadar yavaşlayacaktır. Buna göre kapalı çevrim denetimi ve sabit uç açısı ile denetlenen bir motorun yükü arttırıldığında motorun yavaşlayacağı anlaşılır. Açık çevrimde bilindiği gibi motora uygulanan yüke bağlı olarak motor ya senkron hızda dönecek ya da duracaktır. Kapalı çevrim sabit uç açısı ile denetim yönteminde ise motor durmaz ancak değeri bilinmeyen ve yüke bağlı olarak değişen bir hızda döner.

Uç açısı seçiminde A2'nin nasıl seçileceğini inceleyelim. Örnek olarak üç fazlı 15 derecelik bir motorun istenen son hızının yeterince büyük olduğunu varsayalım. Şekil 4.10'daki çizim kullanılarak motor için sürekli durum uç açısı kabaca 32 derece elde edilir. Şekilden de görüldüğü gibi kullanılacak uç açısı iki motor adımından biraz daha büyüktür. Burada unutulmaması gereken önemli bir nokta motorun son hıza erişmesi için bir ivmelenme fazına ihtiyaç olduğudur. İlk anda motor durduğu için bu durum açık olarak görülmektedir. Aynı zamanda uç açısında iki adımdan daha ileride seçilmiştir. Statik kuvvet eğrilerine göre bu durumda oluşacak kuvvet negatiftir. Yani motor arzu edilen hıza doğru ivmelenmek yerine düşük bir hızla geriye doğru dönecektir. Şekil 4.10'daki değerler farklı motorlar için deneysel olarak elde edilebilir. Tahmin edileceği gibi, tüm akım yükselme etkileri, elde edilen eğrileri etkilemektedir. Bilindiği gibi statik kuvvet eğrilerinde akımların dinamik değişimleri görülememektedir. Sonuçta statik analizden yola çıkarak 32 derecelik uç açısında en yüksek hıza

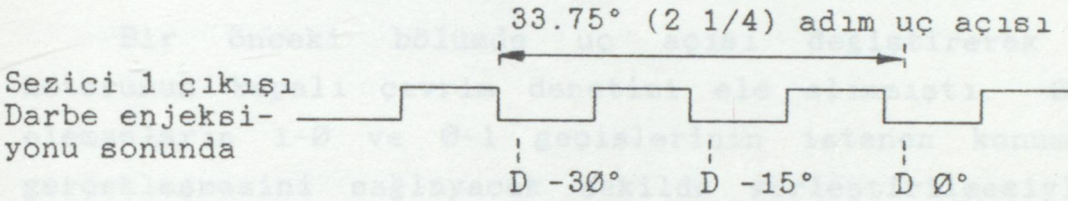
erişilebileceği yorumu yapılamaz. Bunun en önemli nedeni statik analizde akım yükselme hızlarına ait faydalı bilgi bulunmamasıdır. Ancak düşük hızlar için statik kuvvet eğrileri tamamen geçerlidir ve zaten motorun harekete başlarken büyük uç açılarından başlanması uygun değildir. Yapılacak en uygun yaklaşım harekete daha düşük bir uç açısı ile başlamak (örneğin 18.75 derece) ve hız yeteri kadar yükseltildikten sonra uç açısının 32 dereceye çıkarılmasıdır.

Yukarıda yapılan analizi de hatırlayarak, uç açısı yöntemi için sezici bilgilerini kullanmaya yönelik denetim algoritmasında Şekil 4.11.'deki dalga şekilleri kullanılabilir. Şekil 4.11(a)'ya göre sezici "1", 1-0 geçişi rotor dış konumundan 3.75 derece önce gerçekleşecek şekilde yerleştirilmiştir. İlk olarak rotorun Şekil 4.11(a)'daki gibi -15 derecelik açıda beklediğini varsayalım. Bu nokta her 15 derecede bir yerleştirilmiş doğal dış konumlarından herhangi biridir. Şimdi de rotor sağa doğru dönecek şekilde fazların enerjilendiğini varsayalım. Bu durum için faz enerjileme açısı 18.75 dereceye karşılık gelmektedir. Rotor sağa doğru dönmeye başlamıştır. Aynı şekilde sezici "1" in her 1-0 geçişinde fazların enerjilenmesiyle uç açısı A1 18.75 derecede sabit olacaktır. Bu işlemlerde sadece ilk adım sezicilerden bir geçiş gelmeden önce gerçekleştirilmektedir. İlk faz enerjilenmesinden sonra diğer tüm ivmelendirme adımları sezici "1" in her 1-0 geçişinde gerçekleşir.

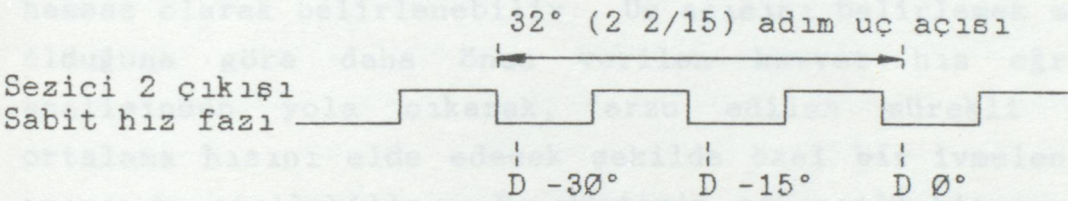
Rotor belli bir süre ivmelendirildikten sonra, daha yüksek sürekli durum ortalama hızlarına erişmek için rotor uç açısı 32 dereceye anahtarlanır. Bu işlem Şekil 4.11(b)'de görüldüğü gibi darbe enjeksiyonu ile gerçekleştirilir. Bu durumu rotor 18.75 derecelik uç açısında iken her zamanki tek bir faz enerjilemesi yerine peşpeşe iki faz enerjilemesi şeklinde açıklayabiliriz. Bu işlem Şekil 4.11(b)'de görüldüğü gibi uç açısını elektronik



(a) İvmelenme fazında



(b) Darbe enjeksiyonu sonunda



(c) Sabit hız fazında

Şekil 4.11 Sabit uç açısı seçerek kapalı çevrimde çeşitli uç açılarının elde edilmesi

olarak 33.75 dereceye (2 1/4 adım) kaydırır. Rotor sağa doğru dönmeye devam eder ve biraz sonra doğal diş konumlarından 32 derece ileride bulunur. Sezici "2", 1-0 geçişi doğal diş konumlarından 2 derece ileride oluşacak şekilde yerleştirilmiştir. Dolayısı ile sezici "2"nin 1-0 geçişi her bir doğal diş konumundan 32 derece ileride gerçekleşecektir. Şekil 4.11(c)'de bu durum gösterilmiştir. Buna göre sezici "2" izlenerek rotor ucunun 32 derece olduğu anlar belirlenebilir. Faz enerjileme işlemleri sezici "2"nin 1-0 geçişlerinde yapılırsa 32 derecelik uç açısı elde edilmiş olacaktır. Fazların sezici "1"e göre enerjilenmelerinden sezici "2"ye göre enerjilenmeleriyle

rotor daha yüksek hıza ulaştırılmış ve sezici "2" nin 1-0 geçişlerinde stator sargılarının enerjilenmesiyle istenen uç açısına da erişilmiş olacaktır.

4.2.5. Zaman Gecikmelerini Kullanarak Uç Açısının Seçimi

Bir önceki bölümde uç açısı değiştirerek adım motorunun kapalı çevrim denetimi ele alınmıştı. Sezici elemanların 1-0 ve 0-1 geçişlerinin istenen konumlarda gerçekleşmesini sağlayacak şekilde yerleştirilmesiyle uç açısı seçilmiş olur. Sezici elemanların geçişleri gözlenerek her geçişte fazların enerjilenmesiyle uç açısı hassas olarak belirlenebilir. Uç açısını belirlemek mümkün olduğuna göre daha önce verilen kuvvet hız eğrileri analizinden yola çıkarak, arzu edilen sürekli durum ortalama hızını elde edecek şekilde özel bir ivmelenme uç açısı da seçilebilir. Bu yöntemin sakıncalı bir yanı ise iki sezici elemanın geçişlerine karşılık gelen elde edilebilir uç açısı sayısının iki ile sınırlanmış olmasıdır. Örneğin bir önceki örnekte sadece 18.75 ve 32 derecelik uç açıları elde edilebilir. Böyle bir sakıncanın önüne geçebilmek için kullanılabilecek uygun yöntemlerden biri değişken zaman gecikmeleri kullanmaktır.

Yöntemin püf noktası ise sezici elemanlardan geçiş kenarlarının elde edilmesinden belli bir zaman gecikmesi sonra fazların enerjilenmesinin sağlanmasıdır. Bu zaman gecikmesi esnasında motor dönmeye devam edecektir, zaman gecikmesi sonunda motor sargıları enerjilenecek, ancak bu anda rotor yeni bir konumda olacaktır. Dolayısıyla rotorun gerçek uç açısı, sezici eleman kenar geçişlerinden hemen sonra fazların enerjilendiği duruma göre daha kısaltılmış olacaktır. Farklı zaman gecikmeleri kullanılarak sezici elemanlardan elde edilen zaman dilimlerinden çok daha küçük aralıklı zaman gecikmeleri elde etmek mümkündür.

4.3. Çalıştırılabilen Kapalı Çevrim Denetim Yöntemleri

Bu yöntemi gerçekleştirmek için önce rotor, durmasına neden olmayacak, yeteri kadar büyük bir uç açısında ivmelendirilir. Öyle bir uç açısı elde edilmelidir ki sıfır zaman gecikmesinde bile rotorun hareketinde bir sorun ortaya çıkmamalıdır. Motorun en yüksek hıza nazaran daha düşük bir hızda çalışması istenirse sezici elemanların kenar geçişleri tespit edildikten sonra sıfırdan farklı bir zaman gecikmesi kullanılır. Bu işlem Şekil 4.12'de açıklanmıştır.

Burada gerçekte uç açısının değil de zaman gecikmesinin doğrudan denetlendiği unutulmamalıdır. Farklı hızlarda motor aynı miktardaki zaman gecikmesi için farklı miktarda yol katedecektir. Dolayısıyla verilen zaman gecikmesi, geçiş kenarının dedekte edildiği andaki rotor hızına bağlı olarak, farklı uç açılarına karşılık gelecektir. Bununla birlikte sürekli halde bu algoritma bir uç açısı sezici gibi davranır. Yani sabit bir zaman gecikmesi ve sabit bir uç açısı (kenar geçişine karşılık gelen) için bu yöntem kararlı sürekli durum uç açısına neden olacaktır. Mikroişlemci sistemlerinde çeşitli zaman gecikmeleri bir dizi içinde saklanabilir ve rotor döndükçe bu dizinin farklı elemanları kullanılabilir. Bu şekilde tüm sürekli durum hızlarına erişmek mümkündür.

Bu yöntemin mikroişlemci ile gerçekleştirilmesi oldukça açıktır. Yeteri kadar büyük bir uç açısına ulaşıncaya kadar motor bir programla hızlandırılır. Örneğin 32 dereceye kadar ve bu açıda hiç bir zaman gecikmesi kullanılmaksızın motor en yüksek hızda çalıştırılabilir. Daha düşük bir hıza erişilmek istenirse zaman gecikmesi yöntemi kullanılır. Her adım atma işleminden sonra belli bir süre gecikme sağlanarak daha düşük hızlar elde edilir.

4.3. Geliştirilmiş Kapalı Çevrim Denetim Yöntemleri

Bundan önceki bölümlerde konum kodlayıcı ele alınmış ve konum kodlayıcının ürettiği işaretler açıklanmıştı. Arzu edilen miktarda adım atıldığının belirlenmesini sağlayan yöntemler açıklanmış, böylece geçici adım kaybedilmesi sorunu ortadan kaldırılmıştır. Daha sonra sürekli durum uç açısını belirlemeye dayanan yöntemler incelenmiştir. Uç açısı rotorun karşılaştığı zorluğu gösterdiği için, bu yöntem kullanıcıya motorun davranışına göre denetim imkanı verir. Bununla birlikte uç açısının seçilmesi tekniğinde bile sabit uç açısı ile denetim yapıldığından adım motoru yüke duyarlıdır. Yani uç açısı sabit tutulup motor yükü değiştirilirse motor hızı da değişir.

Bu bölümde adım motorlarının mikroişlemci kullanılarak denetlenmesi için daha geliştirilmiş teknikler ele alınacaktır. Ele alınacak ilk yöntem, uygulanan yüke uyum sağlayan sabit hız denetleme yöntemi olacaktır. Bu işlem gerçekleştirilirken aynı zamanda denetim yönteminin mikroişlemcilere uygun denetleme algoritmaları ile uyuşması da dikkate alınacaktır. Diğer denetim tekniği bir öncekinden tamamen farklıdır. Bu teknikte stator faz akımları mikroişlemci tarafından denetlenmektedir. Tabii bu iki denetim yöntemi mikroişlemci tabanlı denetim yöntemlerinin tümü anlamına gelmemelidir. Burada esas amaç mikroişlemciler gibi duyarlı elemanların karmaşık denetim işlemlerinde ve geliştirilmiş denetim yöntemlerinde nasıl kullanılacağını göstermektir.

4.3.1. Uç Açısı Hesabı ve Yüke Uyumlu Sabit Hız Denetimi

Bir önceki bölümde uç açısı seçimi ile sabit hız denetimi ele alınmıştı. Belirtildiği gibi sabit uç açısı ile denetim yapıldığında motorun yüklenmesiyle birlikte

hızı da düşmektedir. Bununla birlikte motorun hızının düştüğünün kolayca tespit edilebileceği de açıktır. Öyleyse yüklenme durumunda motor hızını düzenlemek için kullanılacak geri besleme yönteminde motor hızının parametre olarak kullanılıp kullanılamayacağı sorusu sorulabilir. Yük değişmesiyle birlikte uç açısını da dinamik olarak değiştirmek mümkün olabilir. Buna benzer bir yöntem uç açısı hesabı olarak bilinir. Bu yöntemde ilk olarak yapılacak işlem belirlenir (rotorun verilen hızda dönmesi gibi) ve işlemi gerçekleştirmek üzere uç açısını hesaplayan bir algoritma ile uç açısı bulunur. Bu yöntemde uç açısı önceden seçilmez. Denetleyici tarafından işlemin gereğini sağlayacak uç açısı değil de sadece yapılacak işlem bilinmektedir.

Uç açısı fikrini açıklamak için yapılacak özel işlem üzerinde düşünmek gerekmektedir. Burada yapılacak işlem motorun sürekli durum hızını istenen değerde tutacak bir algoritma geliştirmektir. Daha sonra bu yöntemin sabit yük değişmeleine karşılık yüke uyumlu olması sağlanmalıdır. Yani motor yükü sabit farklı bir yük haline geldiğinde motor hızı bunun ardından tekrar istenen hız değerine ulaşmalıdır. Bu işlemde gerçekleştirilecek son aşama mikroişlemci sistemindeki tüm kısıtlamaları dikkate alarak yöntemin mikroişlemci sistemine adaptasyonudur. İşlemler yapılırken çeşitli sorunlarla karşılaşılacağı muhtemeldir.

Bu sorunların ilki, adım motoru için mevcut olan basit bir model bulunmamasıdır. Mevcut modeller ise genellikle birçok eşitlik ihtiva etmekte, ya da motor modelini doğrusal olmaktan tamamen uzaklaştıran, diferansiyel eşitlikler ile motor modellenenbilmektedir. Doğrusal olan motor modelleri ise, motor performansını duyarlı olarak tahmin etmeye olanak tanımayacak kadar basittir.

Bu problemi çözmek için bir yöntem belli anahtar parametreleri, ya da motor davranışlarını tanımlamaya

çalışmak ve diğer tüm etkileri ihmal etmektir. Yani motorun detaylı mekanik davranışı tamamen anlaşılmassa bile belli parametreleri kontrol edilerek motor davranışını denetlemek mümkündür. Adım motoru denetim meselesi ele alınacak olursa, anahtar parametrenin rotor uç açısı olduğu görülür. Statik eğri analizinde incelendiği gibi sabit uç açısı sabit sürekli durum hızına neden olacaktır. Buna göre Şekil 4.9'da görüldüğü gibi uç açısı arttırıldıkça sürekli durum hızı da artacaktır. Tabii ki bu durum belli bir uç açısı sınırı içinde geçerlidir. Çok büyük uç açıları motorun yavaşlamasına neden olacaktır. Bunun nedeni büyük uç açıları için rotorun negatif kuvvet bölgesine düşmesidir. Bölüm sonunda verilen örnekte 32 derecelik uç açısı yaklaşık en büyük uç açısidir. (bu açı motorun çalıştırılabileceği en yüksek hızı verir)

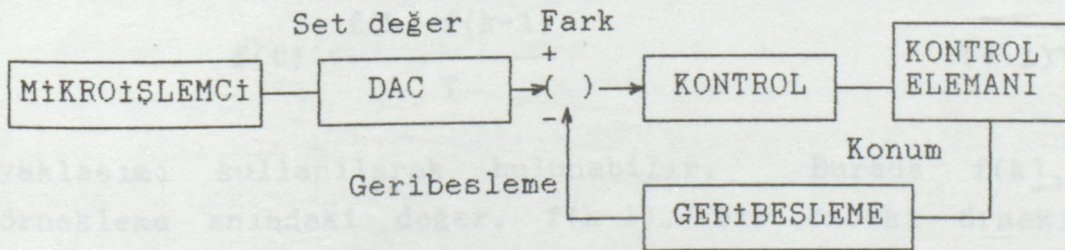
Elde edilen bu sonuçlar ışığında, rotorun uç açısı dinamik olarak denetlenebilirse, sürekli durum hızının da denetlenebileceği söylenebilir. Uç açısının zaman gecikmesi yardımıyla seçilmesi yöntemi uç açısının denetimi için bir yaklaşım sağlar. Bu yaklaşımda rotor 32 derecelik uç açısı konumlarında iken sezici elemanlardan 1-0 geçişleri olacak şekilde sezicilerin yerleştirilmelerini gerektirir. Faz enerjilenmesi her bir kenar sezildiğinde gerçekleştirilirse, motor en yüksek sürekli hal hızına ulaşacaktır. Hızın denetlenebilmesi için sezici elemanların 1-0 geçişleri ile fazların enerjilenmeleri arasına bir zaman gecikmesi yerleştirilir. Bu durumda zaman gecikmesinin denetlenmesiyle dolaylı olarak rotorun uç açısı da denetlenmiş olacaktır.

Hızın denetlenmesinde karşılaşılan bir sonraki sorun ise zaman gecikmesini ayarlamak için hangi kontrol yöntemi ya da algoritmasının kullanılacağıdır. Kontrol yönteminden öncelikle bilinmeyen sabit bir yük durumunda rotorun istenen hızda dönmesini sağlaması istendiği hatırlanırsa; Bu durumda en uygun denetim yöntemi (bu durumlarda yaygın

olarak kullanılan bir denetim olan) PID (oransal-integral-farksal) denetim yöntemidir. Konunun kolayca anlaşılabilmesi açısından kısaca PID denetleyicinin nasıl çalıştığı ve PID denetim fikrinin adım motor denetim probleminde nasıl uygulanacağını açıklamak yerinde olacaktır.

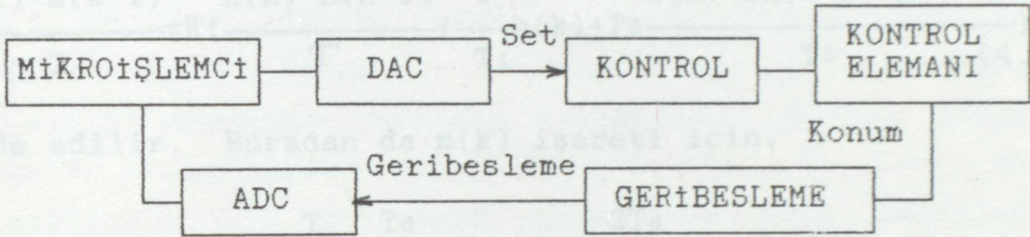
4.3.2 Sayısal Denetim

Bir robot sisteminde, elektronik konum denetim birimlerinin sürülebilmesi için hesaplayıcı sistemin önce istenen makina koordinatlarını ve buna bağlı olarak gerekli set-noktası değerlerini hesaplaması gerekmektedir. Bu set-noktası değerleri denetim sisteminin istenen makina koordinatları için sürülmesini temin ederler. Set-noktası değerleri sayısal değerler olup, denetim sistemine sayısal_analog dönüştürücüler (DAC) yardımıyla aktarılırlar. Bu işlem Şekil 4.13 (a)'da gösterilmiştir. Robot dinamiğinin doğrusal olmaması nedeniyle, robot denetim sisteminin sürekli cevabının, analitik olarak tahmin edilmesi mümkün değildir. Dolayısıyla tüm hareketlilik sınırları dahilinde denetim sisteminin performansını iyileştirmek için sadece set-noktası değerleri değil aynı zamanda denetleyici sistemin



Şekil 4.13 (a) Sürekli denetim sistemi için sayısal set noktasının analog değere dönüştürülmesi

parametreleri de değiştirilmek istenebilir. Bazı sürekli denetim sistemlerinde bu mümkün olmakla birlikte, daha pratik ve duyarlı bir çözüm şekli de bir mikroişlemciyi denetim sisteminin bir parçası haline getirmektir. Bu durum Şekil 4.13 (b)'de gösterilmiştir. Mikroişlemcinin geribesleme işaretini örnekleyebilmesi için sisteme analog-sayısal dönüştürücünün (ADC) eklenmesi gerekmektedir.



Şekil 4.13 (b) Sayısal set noktası ile konum kontrolunda geribesleme çevriminde mikroişlemci kullanılması yöntemi

4.3.3 Sayısal PID Denetleyici

PID denetleyiciyi taklit etmek için kullanılacak denetim algoritmasının geliştirilmesi aynı zamanda sayısal PID denetim prensibini de açıklamaktadır. Zamanın sürekli değişen bir $f(t)$ fonksiyonunun türevi,

$$f(t)' = \frac{f(k) - f(k-1)}{T} \quad (4.1)$$

yaklaşımı kullanılarak bulunabilir. Burada $f(k)$, T örnekleme anındaki değer, $f(k-1)$, bir önceki örnekleme değeri ve T , örnekleme aralığıdır. PID denetim algoritması,

$$m(s) = K \left(1 + \frac{1}{Tis} + T \right) h(s) \quad (4.2)$$

kullanılarak gerçekleştirilir. Yukarıdaki eşitlik her iki yanı S ile çarpılarak tekrar yazılırsa,

$$Sm(s) = K \left(Sh(s) + \frac{1}{T_i} h(s) + T_d S^2 h(s) \right)$$

ve $f'(t)$ 'nin S transformunun $Sf(s)$ olduğu hatırlanırsa

$$\frac{m(k) - m(k-1)}{T} = K \left\{ \frac{h(k) - h(k-1)}{T} + \frac{1}{T_i} h(k) + T_d \frac{h(k) - 2h(k-1) + h(k-2)}{T^2} \right\} \quad \dots (4.3)$$

elde edilir. Buradan da $m(k)$ işareti için,

$$m(k) = m(k-1) + K \left\{ \left[1 + \frac{T}{T_i} + \frac{T_d}{T} \right] h(k) - \left(1 + \frac{2T_d}{T} \right) h(k-1) + T_d h(k-2) \right\}$$

$$m(k) = m(k-1) + A_1 h(k) - A_2 h(k-1) + A_3 h(k-2) \quad \dots (4.4)$$

fark eşitliği elde edilir. Burada $m(k)$ sisteme uygulanan kontrol değerini göstermektedir ve $h(k)$ hatasını hesaplamak için kullanılan fark eşitliği, aşağıdaki şekildedir.

$$h(k) = s(k) - \ddot{o}(k) \quad (4.5)$$

Burada $s(k)$; k anındaki set değeri, $\ddot{o}(k)$; k anındaki geri besleme değeridir. Yukarıdaki son iki eşitlik, her bir örnekleme aralığında çözülüp denetim işareti m^* değerinin hesaplanmasını göstermektedir. Mikroişlemci için önceki çıkış değerinin $m(k-1)$ ve son iki hata değerlerinin $h(k-1)$ ve $h(k-2)$ saklanması gerekmektedir.

Bu işlemlerde fark eşitliği çıkarılırken türev için yapılan yaklaşım unutulmamalıdır. Örnekleme frekansı veri işleme hızı ile sınırlandırıldığı için örnekleme aralığı kısaltıldıkça sayısal denetleyicinin performansı sürekli denetim eşdeğerine daha fazla yaklaşacaktır.

4.3.4 Sayısal PID Denetleyici Fikrinin Adım Motoruna Uygulanması

Adım motoru hızını denetlerken, zaman gecikmesini otomatik olarak değiştirmek için, belli bir hata işareti gerekmektedir. Bu hata işareti aşağıdaki yöntemle elde edilir. İlk olarak istenen sürekli durum hızı belirlenir yani böylece adımlar arası istenen zaman gecikmesi de belirlenmiş olmaktadır. Adımlar arası istenen süre T_{is} ($T_{is}:1/S$) sistemin referans girişi'dir. Sistem çıkışı da adımlar arası ölçülen gerçek süre T_o 'dür. T_o süresi sezici elemanların 1-Ø geçişleri arasındaki süre hesaplanarak ölçülebilir. (1-Ø geçişleri birbirlerinden "1" adım farkla oluşmaktadır). Sezici elemanların 1-Ø geçişleri arasında yürütülen komutların toplam yürütme süreleri kaydedilerek bu süre kolayca hesaplanabilir. Ya da her bir geçiş esnasında bir sayıcı ya da zamanlayıcı sıfırlanıp her 1-Ø geçişi oluştuğuna zamanlayıcı değeri okunarak da geçen süreyi ölçmek mümkündür. Geçişler arası toplam komut yürütme süresi ölçülen süre T_o 'ye eşittir. Bu durumda hata işareti $h=T_{is}-T_o$ farkı ile kolayca elde edilir.

Şimdi de hata işaretinden gecikmeyi hesaplamaya çalışalım. İstenilen adımlar arası zaman gecikmesinin T_{is} olarak belirlendiği düşünelim ve ölçülen sürenin T_{is} 'den daha büyük olarak T_o şeklinde ölçüldüğünü varsayalım. Bu durum Şekil 4.14(a)'da açıklanmaya çalışılmıştır. Hata işareti ($T_{is}-T_o$) olarak belirlendiği için bu durumda oluşan hata pozitifdir. Kolayca anlaşılacağı gibi bu durumda hatayı ortadan kaldırmak için rotorun dönüşü yavaşlatılmalı yani adımlar arası zaman gecikmesi arttırılmalıdır. Bu işlemi yapabilmek için de rotor uç açısı küçültülmelidir. Burada görüldüğü gibi pozitif bir hatayı düzeltmek için zaman gecikmesindeki değişim de pozitif olmalıdır. (Yani zaman gecikmesi arttırılmalıdır). Negatif hata için ise incelenmesi gereken durum Şekil 4.14(b)'de görülmektedir.

Şekilde görüldüğü gibi ölçülen süre oldukça uzundur. Adımlar arası zamanın azaltılması yani rotor dönüş hızının arttırılması gerekmektedir. Hızı arttırmak için de uç açısı küçültülmelidir. Kolayca görüldüğü gibi negatif hatayı düzeltmek için zaman gecikmesinde negatif bir değişim yapılmalıdır (yani gecikme süresi kısaltılmalıdır). Alınan tüm bu sonuçlara göre hatanın doğru bir şekilde ortadan kaldırılması için zaman gecikmesini ayarlarken hata ile doğru orantılı bir işaret kullanılabilecektir.

Mikroişlemci sisteminde zaman gecikmesini ayarlamak için, toplam zamanı küçük zaman dilimlerine kuantalamak gerekmektedir. Zaman gecikmesi oluşturulurken, beklenilmesi gerekli süre içinde kullanılması gereken zaman dilimi sayısı belirlenir. Örneğin zamanın 1 μs 'lik dilimlere bölündüğü varsayılırsa, bu durumda gecikme N sayısı ile ifade edilecek ve N gecikmesinden $1*N$ μs 'lik gecikme anlaşılabilecektir. Bu yöntemdeki kuantalamadan,

$$\begin{array}{l} \text{Tis istenen} \\ \hline \text{Tö ölçülen} \\ \hline \end{array} \quad h = T_{is} - T_{ö} > 0$$

Bu durumda zaman gecikmesi arttırılmalıdır, sonuç olarak uç açısı kısalmış ve hız küçülür.

(a)

$$\begin{array}{l} \text{Tis istenen} \\ \hline \text{Tö ölçülen} \\ \hline \end{array} \quad h = T_{is} - T_{ö} < 0$$

Bu durumda zaman gecikmesi arttırılmalıdır, sonuç olarak uç açısı uzar ve hız artar.

(b)

Şekil 4.14 Hata işaretine bağlı olarak zaman gecikmesinin ayarlanması

kolayca anlaşılacağı gibi sadece 1 μs 'nin katları olan gecikmeleri elde etmek mümkündür. HPC'de işlemci 17 MHz hızla çalışırken zamanlayıcı birimler CLK/16 yani yaklaşık 1MHz hızla çalışmaktadır. Yani HPC ile denetlenebilecek en kısa zaman dilimi 1 μs 'dir.

Şimdi de hata teriminin nasıl belirleneceğini ve zaman gecikmesini ayarlama nasıl kullanılacağını ele alalım. Denetlenebilen en kısa zaman birimi 1 μs olduğuna göre N μs 'lik zaman gecikmesi elde etmek için ilgili zamanlayıcıya N sayısının girilmesi gerekmektedir.

İstenen hız örnek olarak 800 adım/sn ise adımlar arası istenen zaman gecikmesinin (yani motor hızının) 1250 μs olması yani N sayısının 1250 olmasını gerektirir. Yine örnek olarak kodlayıcıdan gelen kesmelerle motorun iki adımı arasında ölçülen süre T_0 1100 μs olsun. Buna göre hata $T_{is} - T_0 = 1250 - 1100 = 150 \mu s$ zaman dilimine karşılık gelir. Şimdi bu hata terimini kullanabilmek için motorun denetiminde kullanılan zaman gecikmesine 1 birim eklendiğinde adımlar arası ölçülen zamanın da 1 birim artacağını varsaymak gerekir.

Seçilen sayılara göre motorun istenen hızın üzerinde olduğu görülmektedir. Buna göre motor denetiminde nominal gecikme değeri N arttırılarak adımlar arası süre uzatılmalıdır. Gerçekte N'deki artmayı hata miktarı kadar yani 150 birim yaparsak bir sonraki adımda motor hızı da 150 birim artmaz. Bunu hızlandırmak için PID denetimin oransal kazanç katsayısı K_p 1'den farklı 2,3,4 olarak seçilebilir. Kullanılacak gecikme değeri $(N(n))$ bir önceki adımda kullanılan nominal gecikme $(N(n-1))$ ile oransal kazanç (K_p) hata $(h(n))$ çarpımının toplanmasıyla elde edilir. (Burada bir önceki adımdaki hatanın sıfır olduğu varsayılmıştır). Sonuç olarak n'inci adımdaki denetim değeri

$$N(n) = N(n-1) + K_p \cdot h(n)$$

Ancak hata işaretindeki değişimler hesaplanıp ve denetim kullanılarak bulunur. Aslında oransal kazanç ile çarpılacak hata işareti $h(n)$ ile $h(n-1)$ farkıdır. Aksi halde sürekli durum hatalarında kısa zamanda saturasyon olayı ortaya çıkacaktır. Denetim işleminde sürekli durum hatalarının önlenmesi aynı zamanda integral teriminin gerekliliğini de ortaya koymaktadır. Sürekli durum hatalarını önlemek için n ve $n-1$ 'inci hataları kullanarak integral kontrol terimi

$$K_i \cdot (1/T_i) \cdot (h(n) + h(n-1)) / 2$$

şeklinde elde edilir. integral teriminde çarpan olarak gelen integral zaman sabiti T_i de integral kazanç sabiti K_i içinde kullanılarak hesaplamalarda kolaylık sağlanabilir. Denetim işleminde integral terimi, denetlenen değişkende meydana gelen sürekli durum hatalarını düzenlemekte ve hata varolduğu sürece, denetim değeri de integral işlemine bağlı olarak değişmektedir. Böylece eğer sürekli durum hatası sistemi doyuma götürecek kadar büyük değilse, sistem belli bir süre sonra kararlı duruma ulaşacaktır.

integral kazanç katsayısı gereğinden büyük seçilirse kararlı duruma ulaşması salınımlı olacaktır, çünkü denetim değerinde sistemin cevap verme hızından daha büyük değişimler ortaya çıkmaktadır. Denge duruma erişmek isteyen ya da denge durumundaki bir sistemde hata teriminde ani değişimler ortaya çıkarsa denetim teriminin de ani olarak değiştirilmesi gerekecektir. Ancak sistem denetim değerinin değişim hızına cevap verecek kadar hızlı değildir. Bu nedenle denetim teriminin değişim hızını belli aralıklarda sınırlamak ya da yavaşlatmak gerekir. Bu işlemin gerçekleştirilmesi hata teriminin türevi alınarak gerçekleştirilir. Türev terimi denetim terimindeki ani değişimleri bastırarak sistemin cevap verebileceği sınırlarda tutacaktır. Tabii ki burada türev teriminin kazanç katsayısı da dikkatli seçilmelidir. K_p , K_i , K_d

terimleri mikroişlemci sisteminde tamsayı olmalıdır. Ancak hata işaretindeki değişimler hesaplama ve denetim sınırlarını zorlayacak kadar büyük olursa terimleri kesirli seçmek gerekir. Bunu gerçekleştirmek için 1 μ s'lik zaman dilimi 4,8,16,... μ s'lik zaman dilimlerine genişletilerek, dolaylı yoldan kesirli terimler elde edilmiş olur.

4.3.5 Üç Adım Motoru için PID Denetim Algoritması

Geliştirilen algoritma yardımı ile üç adım motoru aynı anda PID olarak denetlenebilmektedir. Kullanılan adım motorları ile yaklaşık 1000 adım/sn hızlara erişilmiştir. PID denetim algoritmasının, yeni denetim değerini üretmeye hızı yetmediği durumlarda bile, bir önceki kontrol değeri kullanılarak, motorun hız kaybetmesi önlenmiştir. Algoritmanın açıklanması aşağıdaki gibidir.

Program akışında ilk olarak HPC çevrebirimleri, yazıcıları, zamanlayıcıları ve kesme vektörleri koşullanır. Motorlara ait değişkenler ilk değerlerine set edilir. Her motorun hızının ölçülmesinde kullanılan kesme vektörleri ayrı ayrı koşullanır. HPC'de kesme hizmetlerinin birbirlerine göre öncelikleri olduğundan motorlara 1,2,3 öncelik sırasıyla hizmet verilir. Herbir motor için farklı vektörlerde yerleşmiş eşdeğer kesme altprogramları kullanılır. Daha sonra motorlara ait hız, adım sayısı gibi değişkenler okunur. Adım sayısı 0'dan büyükse motora ilk adım darbesi uygulanır. Bu işlemden önce zamanlayıcılar yeteri kadar büyük geri sayma değerlerine set edilmiş ve durma modunda olmalıdır. Bundan sonra zamanlayıcılar tekrar normal çalışma moduna alınır. Tüm ön koşullama işlemleri tamamlandıktan sonra genel kesme izni (GIE) verilir. Motor1 öncelikli olduğu için işaretçi önce motor1 değişkenlerini işaretler ve DENETİM programı yürütülmeye başlanır. DENETİM adlı altprogram her üç motor için de kullanılır. Bu programa gelirken sadece işaretçinin işaretlediği değişkenler değişmektedir.

DENETİM programı başlangıcında motor durum bayrağı test edilir, eğer motor durma modunda ise hiçbir işlem yapılmadan geri dönülür ve bir sonraki motora ait işlemlerin yürütülmesine başlanır. Motor durma modunda değilse motorun bir önceki adımda hesaplanan değerleri kullanıp kullanmadığı test edilir. Bir önceki adımda hesaplanan değerler henüz kullanılmadıysa, bu işlemlerin gerçekleşmesini beklemek üzere ana programın akışına dönülür. Motor için hesaplanan değer işlenmişse, yani motor kesme hizmeti görmüşse bundan sonra yeni denetim değerleri üretmek gerekmektedir. Bu nedenle motorun hangi hareket fazında olduğu belirlenmelidir. Motor ivmelenme, yavaşlama ya da sabit hız (PID) fazında olabilir. Motor ivmelenme fazında ise motorun istenen hıza erişip erişmediği test edilir. Motor istenen hıza erişmişse hareket fazı PID olarak değiştirilir ve PID işlemine başlanır. Aksi halde motor hızı arttırılmaya devam edilir. Motor istene hıza erişmiş ve belli bir süre bu hızda hareket ederek istenen sayıda adım atmışsa artık motor yavaşlama fazına geçirilmelidir. Bu işlem kesme altprogramında yapılır. Yavaşlama fazında kalan adım sayısına bağlı olarak durulabilecek zaman gecikmeleri seçilir. Kalan adım sayısı 1'den büyükse kullanılan zaman gecikmesi arttırılarak motorun durması kolaylaştırılır 1'e eşitse motor durma fazına alınır ve son adımı atması beklenmek üzere ana program akışına geri dönülür.

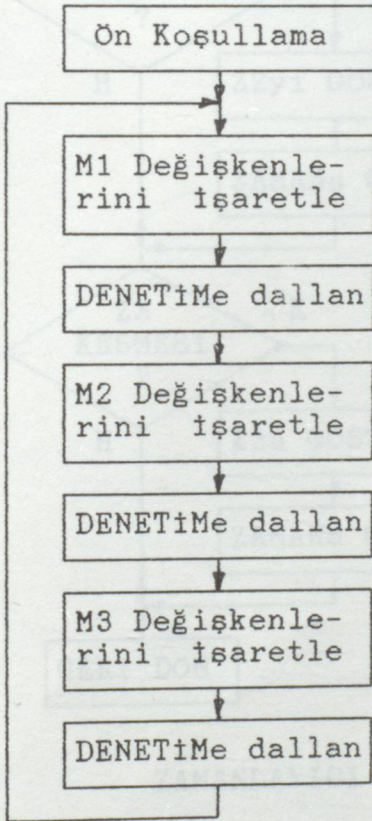
PID programında ise kesme programında ölçülen hız değeri ile istenen hız değerinin farkı alınarak hata değeri belirlenir. HATA0 değeri A1 katsayısı ile çarpılarak SAK1 gözüne kaydedilir. Bir önceki adım hatası HATA1 ile A2 değeri çarpılarak SAK2 gözüne ve HATA2 değeri A3 ile çarpılarak SAK3 gözüne yazılır. Hesaplanan bu çarpım değerleri tek tek toplanarak bir önceki çevrimdeki kontrol değerine (KONTROL1) ilave edilmesiyle yeni kontrol değeri (KONTROL0) elde edilir. Son adım olarak hata değerleri güncellenmelidir. Güncelleme işleminden sonra ana program akışına geri dönülür.

Yukarıdaki işlemler herbir motor için aynıdır ve ön koşullama işleminde motorların ilk adım darbeleri verildiği için artık konum kodlayıcılardan kesme gelebilecektir.

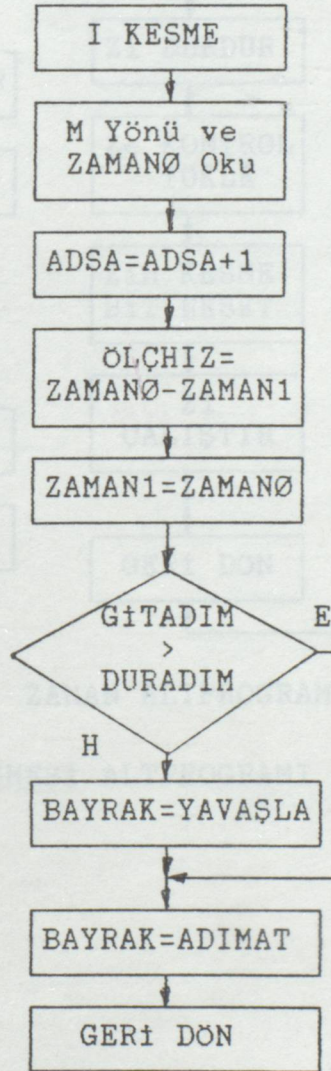
Kesme alt programında önce motora ait yön bilgisi okunur bundan sonra 17Mhz/16 hızıyla serbest çalışan 16 bitlik zamanlayıcı değeri okunur. Okunan bu değer ZAMANØ değeridir ve bir önceki kesmede okunan ZAMAN1 değeri çıkarılarak ölçülen hız değeri elde edilir. Bir sonraki kesme de kullanılmak üzere ZAMANØ değeri ZAMAN1 değerine atanır. Kalan adım sayısı durulabilecek max adım sayısından daha büyükse normal çalışma işlemine devam edilir. Aksi halde motor durum bayrağı yavaşlama moduna getirilerek bundan sonraki adımlarda motorun yavaşlamaya başlaması sağlanır. Adım sayacı bir arttırılarak bayrak gözü üzerinde adımın onaylandığı belirtilmesinden sonra kesme altprogramından geri dönlür.

Normal durumlarda kodlayıcıdan gelen kesme istekleri zamanlayıcı kesmesinden önce gerçekleşir. Yani zamanlayıcı kesmesi daha önce gelmişse motor ya hiç dönmüyor ya da yük çok büyük olduğundan istenmiyecek kadar yavaş dönüyordur. Bu durumda hata ortaya çıkmıştır. HPC'de herbir kesme bacağı için ayrı kesme vektörü olmasına karşılık zamanlayıcılara ait kesme vektörü tektir. Bu nedenle üç motorun zamanlayıcıları ile ilgili kesmelerin aynı kesme hizmet programında işlenmesi ve bu program içinde hangi zamanlayıcıdan kesme isteği gelmişse ona kesme hizmeti verilmelidir. Zamanlayıcı kesmesi geldiğinde motor1 öncelikli olduğu için önce zamanlayıcı1'den kesme isteği gelip gelmediğine bakılır. Kesme isteği gelmişse motor1'e ve zamanlayıcı1 'e ait değişkenler işaretlenerek herbir zamanlayıcı için ortak olan, ZAMAN alt programına gidilir. ZAMAN altprogramında ilk olarak bir önceki adımın onaylanıp onaylanmadığı yani motorun kesme hizmeti görüp görmediği test edilir. Adım onaylanmadıysa yukarıda da belirtildiği gibi hata otraya çıkmıştır ve uygun işlemin yapılacağı bir

altprograma dallanılır. Normal durumda ise motora bir adım darbesi uygulanır ve kesme gelen zamanlayıcı durdurulur. Durdurma işleminden sonra zamanlayıcıya hazır bulunan KONTROLØ değeri yüklenerek zamanlayıcının kesme geldiğini gösteren biti kesme hizmeti yürütüldüğü için temizlenir. Zamanlayıcının çalışması tekrar başlatılarak kesmeden geri dönülür. Geri dönme işlemi zamanlayıcılardan çoklu kesme gelebileceği gözönünde bulundurularak tekrar zamanlayıcı kesme programına yapılmaktadır. Benzer işlemler diğer zamanlayıcılar için de yapılarak işlemlere devam edilir.



ANA PROGRAM BLOĞU



KESME PROGRAMI

M1: Motor1
M2: Motor2
M3: Motor3

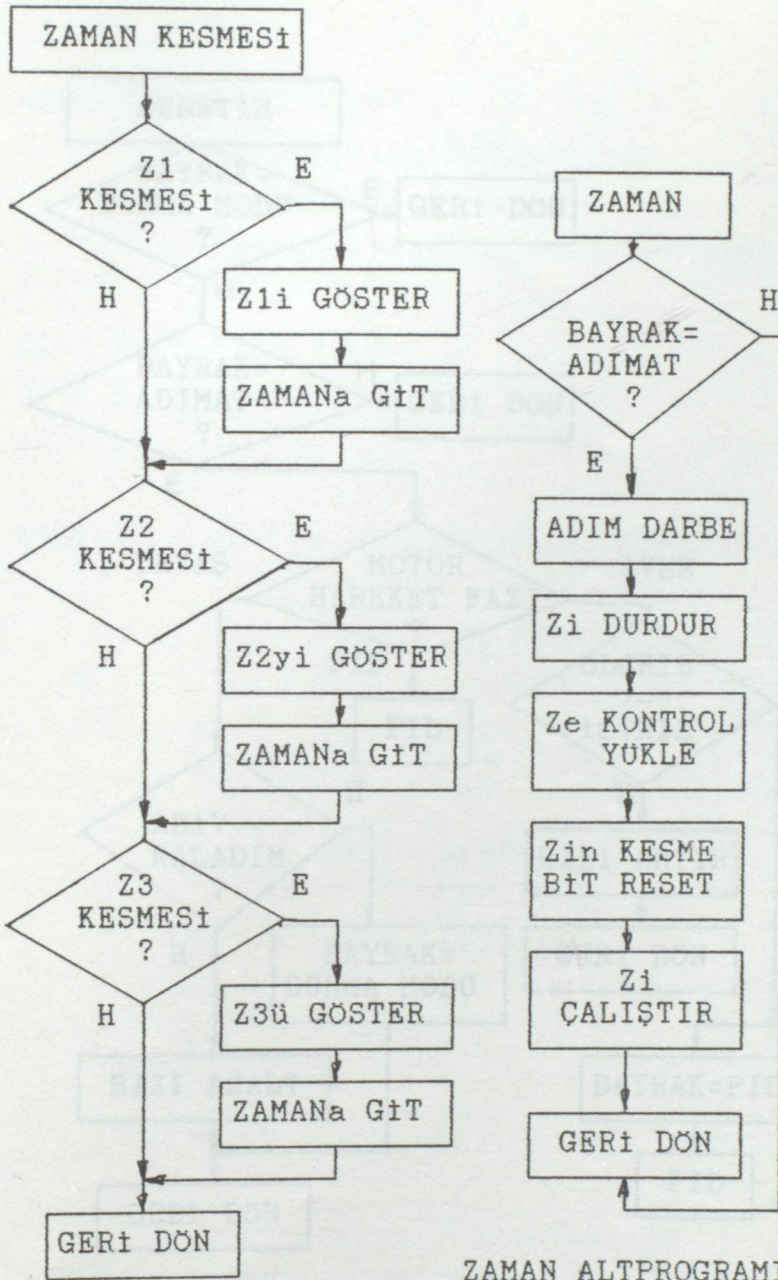
ZAMANØ: Kesme anındaki serbest çalışan zamanlayıcı değeri
ZAMAN1: Bir önceki kesmedeki zamanlayıcı değeri

ÖLÇHİZ: Ölçülen hız değeri

ADSA: Sayılan adım değeri

GİTADIM: Gidilecek adım sayısı (kalan adım)
DURADIM: Durulabilecek adım sayısı

BAYRAK: Motora ait durumu gösteren 8 bitlik yazıcı

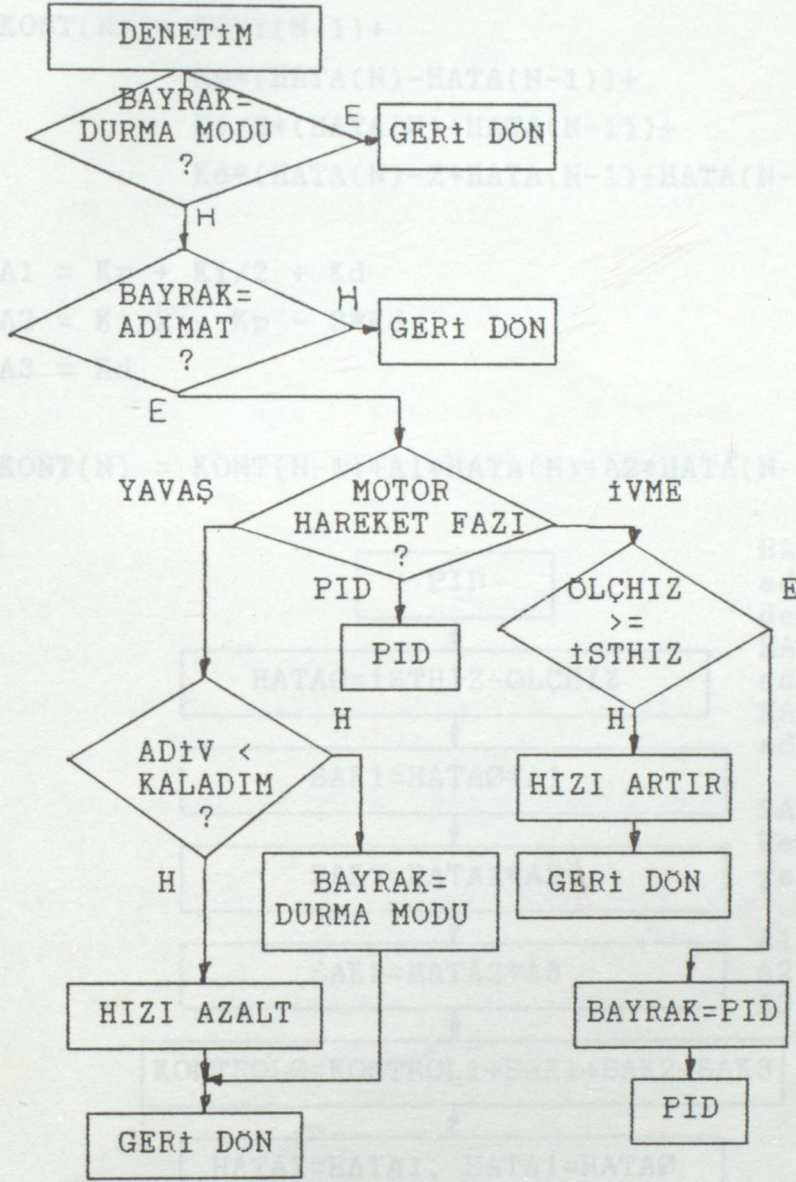


Z1: Zamanlayıcı1
 Z2: Zamanlayıcı2
 Z3: Zamanlayıcı3

ADIMAT: Bayrak yazıcısındaki motorun bir önceki adımı attığını gösteren bit

ZAMAN ALTPROGRAMI

ZAMANLAYICI KESMESİ ALTPROGRAMI AKIŞI



İSTHİZ: Erişilmek istenen hız

ADİV: ivmelenme adımı sayısı

HERBİR MOTORA AİT DENETİM ALTPROGRAMI AKIŞI

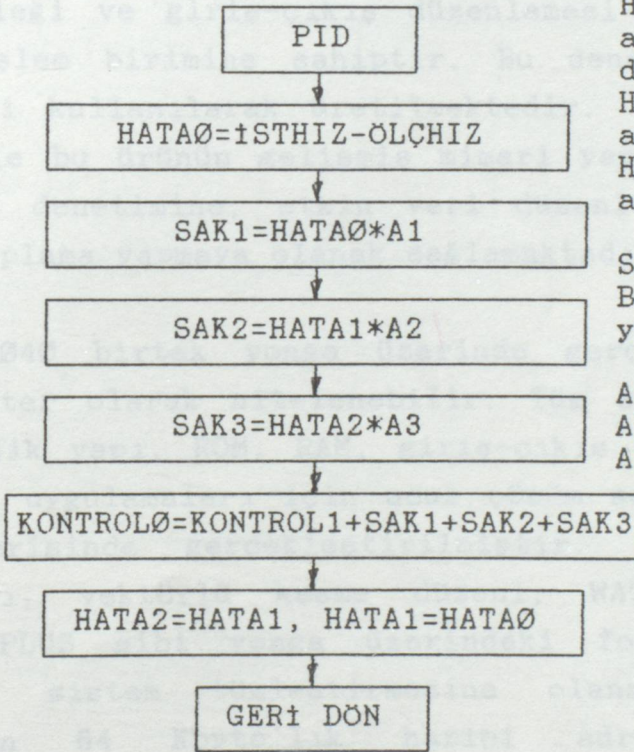
$$\begin{aligned}
 KONT(N) &= KONT(N-1) + \\
 &Kp*(HATA(N)-HATA(N-1)) + \\
 &Ki/2*(HATA(N)+HATA(N-1)) + \\
 &Kd*(HATA(N)-2*HATA(N-1)+HATA(N-2))
 \end{aligned}$$

$$A1 = Kp + Ki/2 + Kd$$

$$A2 = Ki/2 - Kp - 2*Kd$$

$$A3 = Kd$$

$$KONT(N) = KONT(N-1) + A1*HATA(N) + A2*HATA(N-1) + A3*HATA(N-2)$$



HATAØ: Atılacak
adıma ait hata
değeri

HATA1: Bir önceki
adımdaki hata değeri
HATA2: İki önceki
adımdaki hata değeri

SAK1, SAK2, SAK3:
Bellekte herhangi
yaz-boz gözleri

A1: $Kp + Ki/2 + Kd$
A2: $Ki/2 - Kp - 2Kd$
A3: Kd

DENETİME AİT PID ALTPROGRAMI

BÖLÜM 5

HPC MİKRODENETLEYİCİSİ

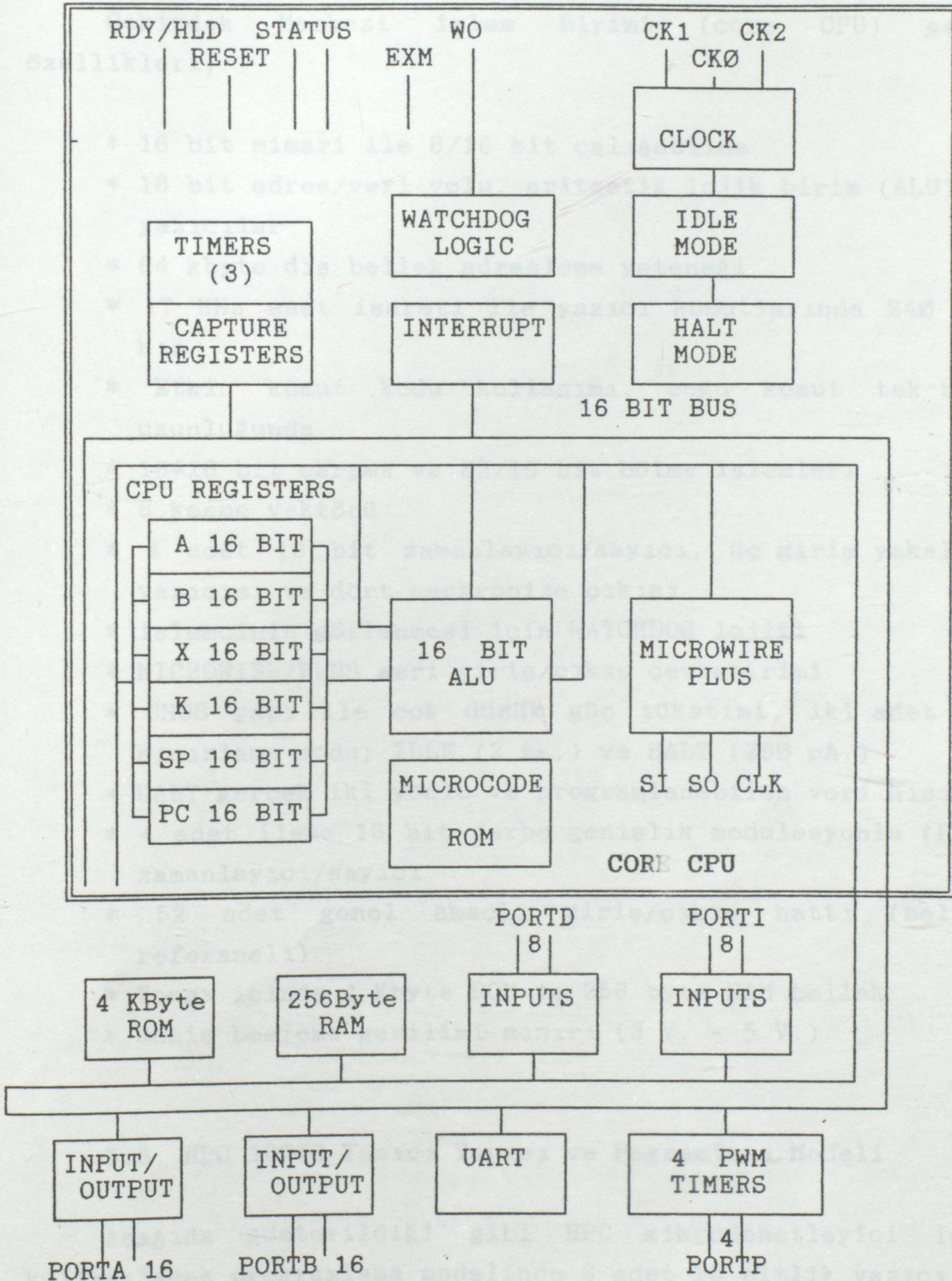
5.1 HPC Genel Tanımı

HPC16040 HPC yüksek performanslı mikrodenetleyici ailesinin ilk elemanlarından biridir. Bu denetleyici ailesinin her bir elemanı çeşitli uygulamalarda kullanılmak üzere belleği ve giriş-çıkış düzenlemesi aynı olan çekidek merkezi işlem birimine sahiptir. Bu denetleyici mikroCMOS teknolojisi kullanılarak üretilmektedir. Kullanılan üretim tekniği ile bu ürünün gelişmiş mimari yapısı hızlı ve esnek giriş-çıkış denetimine, etkin veri düzenlemesine ve yüksek hızlı hesaplama yapmaya olanak sağlamaktadır.

HPC16040 birtek yonga üzerinde gerçekleştirilmiş bir mikrokompüter olarak nitelenebilir. Tüm sistem zamanlaması, dahili lojik yapı, ROM, RAM, giriş-çıkış birimleri , yüksek performans uygulamaları için ucuz çözüm sağlamak üzere aynı yonga içerisinde gerçekleştirilmiştir. UART, sekiz adet zamanlayıcı, vektörlü kesme düzeni, WATCHDOG lojiği ve MICROWIRE/PLUS gibi yonga üzerindeki fonksiyonlar yüksek yoğunluklu sistem tümleştirmesine olanak sağlamaktadır. HPC16040'ın 64 Kbyte'lık harici adresleme yeteneği, mikroişlemcilerle birlikte pahalı çevrebirimlerin kullanımını gerektiren uygulamalarda kullanılmasına olanak sağlar.

Kullanılan microCMOS teknolojisi çok düşük güç tüketimi ile kullanıcıya en uygun hız/güç çözümünü sunar. Mikrodenetleyicinin IDLE ve HALT modları ilave güç tasarrufu sağlamak amacıyla kullanılmıştır.

5.2. HPC Blok Diyagramı ve HPC Genel Özellikleri



Şekil 5.1 HPC BLOK DİYAGRAMI

5.2. HPC Blok Diyagramı ve HPC Genel Özellikleri;

Çekirdek Merkezi İşlem Birimi (core CPU) genel özellikleri;

- * 16 bit mimari ile 8/16 bit çalışabilme
- * 16 bit adres/veri yolu, aritmetik lojik birim (ALU) ve yazıcılar
- * 64 kbyte dış bellek adresleme yeteneği
- * 17 MHz saat işareti ile yazıcı komutlarında 240 ns. hız
- * Etkin komut kodu kullanımı, çoğu komut tek byte uzunluğunda
- * 16*16 bit çarpma ve 32/16 bit bölme işlemleri
- * 8 kesme vektörü
- * 4 adet 16 bit zamanlayıcı/sayıcı, üç giriş yakalama yazıcısı ve dört senkronize çıkışı
- * işlemcinin gözlenmesi için WATCHDOG lojiği
- * MICROWIRE/PLUS seri giriş/çıkış çevrebirimi
- * CMOS yapı ile çok düşük güç tüketimi, iki adet güç sınırlama modu; IDLE (2 mA.) ve HALT (250 µA.)
- * UART gerçek iki yönlü ve programlanabilen veri hızında
- * 4 adet ilave 16 bit darbe genişlik modülasyonlu (PWM) zamanlayıcı/sayıcı
- * 52 adet genel amaçlı giriş/çıkış hattı (bellek referanslı)
- * Yonga içinde 4 Kbyte ROM ve 256 byte RAM bellek
- * Geniş besleme gerilimi sınırı (3 V. - 5 V.)

5.3. HPC 16040 Yazıcı Yapısı ve Programlama Modeli

Aşağıda gösterildiği gibi HPC mikrodenetleyici için kullanılacak programlama modelinde 6 adet 16 bitlik yazıcı ve elde biti vardır. Tüm yazıcılar bellek haritası üzerinde de yerleşmiş olduğu için byte byte adreslenebilir ve erişilebilir durumdadır.

Akümülatör (A) yazıcısı 00C8 HEX adresinde,

16 bitlik A yazıcısı bir çok komut için genel hedef yazıcısıdır. Ancak bellekten belleğe adresleme modlarında genellikle mevcuttur.

(B) adres yazıcısı 00CC HEX adresinde,

Dolaylı adresleme için birincil adres yazıcısı 16 bitlik B yazıcısıdır. Bazı özel komutlar için B yazıcısı otomatik olarak bellekte tablolar veya diziler üzerinde hareket etmek üzere bir veya iki azaltılıp arttırılabilir. Aynı zamanda B yazıcısı, K yazıcısı ile karşılaştırılarak istenilen sınıra ulaşıldığında, çevrimin dışına çıkmak için bir atlama işlevi gerçekleştirmede kullanılır.

(X) adres yazıcısı 00CE HEX adresinde,

B yazıcısında olduğu gibi X adres yazıcısı da dolaylı adreslemeler için kullanılabilecek 16 bitlik yazıcıdır. Sınır testi olmamakla birlikte, B yazıcısı gibi otomatik olarak bir veya iki eksiltiilip arttırılabilir. X yazıcısı bir çevrim içinde B yazıcısı ile bağlantılı olarak kullanılırsa sadece üç adet tek bytelık komut ile bellekten belleğe blok aktarımı mümkün olur.

(K) sınırlama yazıcısı 00CA HEX adresinde,

K yazıcısı B yazıcısının otomatik eksiltiilip arttırılmasında kullanılan 16 bitlik yazıcıdır. Bu işlemi gerçekleştiren komutlarda B yazıcısı K yazıcısındaki adres değerini aştığında bir atlama işlemi gerçekleştirilir.

(SP) yığın gösterici 00C4 HEX adresinde,

Yığını adresleyen yığın gösterici 16 bitlik SP

yazıcısıdır. SP yazıcısı her bir PUSH komutu yada alt programa dallanma işleminde 2 arttırılır ve her bir POP yada RETURN komutundan sonra 2 azaltılır. Kesme programları da 16 bitlik geri dönüş adresini yığına atmaktadır. SP yazıcısı bellek haritasındaki adresinden okunabileceği gibi üzerine de herhangi bir sayı yazılarak yığın 16 bitlik bellek alanı üzerinde herhangi bir yere yerleştirilmiş olur ve yığın derinliği mevcut belleğin sınırlarına bağlıdır.

(PC) program sayacı 00C6 HEX adresinde,

16 bitlik PC yazıcısı 64K'lık adresleme sınırı içindeki program belleğini adresleyen sadece okunabilir bir yazıcıdır.

işlemci durum yazıcısındaki (PSW), (C) elde (carry) biti

Elde (C) biti PSW yazıcısındaki 5 nolu bit konumunda bulunur. Yapılan toplama veya çıkarma işleminden sonra meydana gelen elde taşmasını yada borç durumunu gösterir. Elde biti hem 8 bitlik hem de 16 bitlik işlemlerden etkilenecek şekilde kullanılabilir. Toplama komutlarından ortaya çıkabilecek işaretli taşmalarda elde biti lojik "1" olur ve çıkarma işlemlerinden gelebilecek borç durumlarında ise elde biti lojik "0" olur. Çok duyarlıklı toplama ve çıkarma komutlarında, bir önceki komuttan gelen elde yada borç durumu hesaba katılacak şekilde elde biti giriş olarak da kullanılabilir. MULT ve DIV komutları elde bitini daima sıfırlamaktadır. DIVD komutu ise taşma yada sıfıra bölme durumu ortaya çıkmadığı sürece elde bitini sırlar. SC ve RC komutları kullanılarak elde biti sıfırlanıp, "1" e set edilebilir, böylece elde biti genel amaçlı bir bayrak olarak da kullanılabilir.

5.4. HPC Adresleme Modları

5.4.1. Akümülatörün Kaynak Olduğu Adresleme Modları,

Yazıcı üzerinden dolaylı (register indirect) adresleme

Bu adreslem modu HPC için normal adresleme modudur. Bu modda kullanılan komutlar bir byte'lık komutlardır. Adres bilgisi bu mod için B yazıcısında (bazı komutlar için X yazıcısında) bulunur. Komuta ait işlem B yazıcısının işaret ettiği bellek adresi üzerinde gerçekleştirilir.

Doğrudan (direct) adresleme

Komutun kendisi ile birlikte işlem yapılacak bellek gözünü doğrudan işaretleyen 8 yada 16 bitlik adres alanını da içerir.

Dolaylı (indirect) adresleme

Komut 8 bitlik adres alanına sahiptir. Adreslenen 16 bitlik kelime işlem yapılacak bellek gözünü gösterir.

Sıralı (indexed) adresleme

Komut 8 bitlik adres bölgesini ve 8 yada 16 bitlik yerdeğiştirme alanını içerir. Adreslenen kelimenin içeriği yerdeğiştirme adresi ile toplanarak işlem yapılacak bellek gözünün adresi elde edilir.

ivedi (immediate) adresleme

Komut işlem alanı olarak kullanılan 8 yada 16 bitlik ivedi alanı kullanır.

Yazıcı dolaylı (register indirect) adresleme (otomatik arttırmalı yada azaltmalı)

işlem bölgesi X yazıcısı ile adreslenen bellek alanıdır. Bu modda X yazıcısı otomatik olarak azaltılır yada arttırılır (8 bit modu için "1", 16 bit modu için "2").

Yazıcı dolaylı (register indirect) şartlı atlama ile birlikte otomatik arttırmalı ve azaltmalı adresleme

Bu adresleme modunda işlem bölgesi B yazıcısı ile adreslenen bellek alanıdır. Bu mod B yazıcısını otomatik olarak azaltır veya arttırır (8 bit modu için "1", 16 bit modu için "2"). Bundan sonra B yazıcısı K yazıcısı ile karşılaştırılır. Eğer B'nin içeriği K'nın içeriğini aşarsa bir atlama durumu ortaya çıkar.

5.4.2. Belleğin Doğrudan Hedef Olduğu Adresleme Modları

Bellekten belleğe doğrudan adresleme (direct memory to direct memory)

Komut 8 yada 16 bitlik adres alanını içerir. Alanlardan biri doğrudan işlem yapılacak kaynak bölgeyi göstermektedir, diğer alan ise doğrudan işlem yapılacak hedef bölgeyi göstermektedir.

Doğrudan belleğe ivedi adresleme (immediate to direct memory)

Komut 8 yada 16 bitlik adres alanını ve 8 yada 16 bitlik ivedi veri alanını içerir. ivedi veri alanı işlem bölgesi ve doğrudan alan hedef bölgesidir.

B ve X yazıcılarını kullanarak çift yazıcı dolaylı adresleme (Double register indirect)

Bu adresleme modu sadece RESET, SET ve IF bit komutlarında kullanılır. 64 Kbyte'lık adres sınırı içindeki

Özel bir bit B ve X yazıcısı kullanılarak adreslenebilir. İşlenecek belleğin adresi B yazıcısına X yazıcısının yüksek anlamlı 13 biti eklenerek elde edilir. Bellek gözündeki işlenecek yada test edilecek özel bit ise X yazıcısının en düşük anlamlı 3 biti kullanılarak seçilir.

Yeni tasarıma ve program çevirileri için çok fonksiyonlu

Konu 5.5. Etkin Komut Kodu Kullanımı

Tek yonga üzerinde kurulan mikrodenetleyiciler için en önemli kriter komut kodu etkinliğidir. Komutların kodlanma tekniği ne kadar iyi olursa yonga üzerine yerleştirilebilecek özellikler o kadar fazla olur. Yonganın bellek ölçüleri sabit olduğundan kodlama etkin değilse yonganın birçok özelliği kullanılamayabilir yada programcı yonganın daha büyük ve daha pahalı tipini almak zorunda kalır. HPC tamamen kod etkinliğine sahip olacak şekilde tasarlanmıştır.

5.5.1 Tek Byte'lık Komutlar

HPC'deki komutların çoğu tek byte'lık komutlardan oluşmaktadır. Aynı zamanda HPC'de kodlamada etkinlik sağlayan iki adet komut türü vardır.

JP komutu bir byte'lık dallanma komutudur. Bu komutla gerçekleştirilebilecek dallanma miktarı ileriye veya geriye 32'dir ancak mikrodenetleyici uygulamalarında karşılaşılan çoğu dallanma işlemleri bu sınır içinde olmaktadır. Ama mikrodenetleyicilerin çoğu hatta 8 bitlik olanlar bile herhangi en kısa dallanma işlemi için bile 2 byte'lık komutlar kullanmaktadır.

JSRP bir bytelık alt program çağırma komutudur. Kullanıcı en çok kullandığı alt program adreslerinin 16 tanesini bir tablo şekline getirmekte ve bu işlemten sonra bu çağrılar sadece bir byte uzunluğunda olmaktadır. Diğer mikrodenetleyicilerin çoğunda alt program çağrıları 2 byte

hatta 3 byte uzunluğundadır.

Bunlara ek olarak HPC'de program belleğinde ileri veya geri 1k'lık altprogram çağrıları etkin bir şekilde gerçekleştirilebilmektedir.

Veri taşıma ve program çevrimleri için çok fonksiyonlu komutlar

HPC çoklu işlem yapabilen tek bytelık komutlara sahiptir. Örneğin tek bytelık XS komutu aşağıdaki işlemleri yapar.

* A yazıcısındaki veri ile B yazıcısının gösterdiği yerdeki veri yer değiştirilir.

* B yazıcısı arttırılır.

* B yazıcısı K yazıcısı ile karşılaştırılır.

* B yazıcısının içeriği K yazıcısının içeriğinden büyükse şartlı atlama üretilir.

Çok fonksiyonlu komutların önemi ardışıl bellek bölgelerinde çevrim oluşturulurken ve işlem bittiğinde çevrimden çıkılırken daha iyi anlaşılmaktadır.

5.5.1 Bit İşleme Komutları

Belleğin, giriş/çıkış yazıcılarının, herhangi bir biti tek byte'lık bit komutları kullanılarak değiştirilebilir ve test edilebilir. Bu bitler doğrudan yada dolaylı olarak adreslenebilir. Tüm yazıcılar ve giriş/çıkış bellek haritalı olduğundan etkili kontrol yapabilmek için özel bitlerin değiştirilmesi oldukça kolaydır. Bu işlemler yalnızca IRPD yazıcısı için geçerli değildir. IRPD yazıcısında yalnızca ivedi yükleme komutu bir biti yada bitleri değiştirebilir.

5.6. HPC Komut Kümesi

5.6.1. Aritmetik işlem Komutları

ADD: Toplama komutu

$MA + Meml \Rightarrow MA \text{ elde} \Rightarrow C$

ADC: Eldeli toplama komutu

$MA + Meml + C \Rightarrow MA \text{ elde} \Rightarrow C$

DADC: Eldeli ondalıklı toplama

$MA + Meml + C \Rightarrow MA(\text{ondalıklı}) \text{ elde} \Rightarrow C$

SUBC: Eldeli çıkarma komutu

$MA - Meml + C \Rightarrow MA \text{ elde} \Rightarrow C$

DSUBC: Eldeli ondalıklı çıkarma komutu

$MA - Meml + C \Rightarrow MA(\text{ondalıklı}) \text{ elde} \Rightarrow C$

MULT: İşaretsiz çarpma

$MA * Meml \Rightarrow MA \& X, \emptyset \Rightarrow K, \emptyset \Rightarrow C$

DIV: İşaretsiz bölme

$MA / Meml \Rightarrow MA, \text{kalan} \Rightarrow X, \emptyset \Rightarrow K, \emptyset \Rightarrow C$

IFEQ: Eşitlik testi

MA ile Meml'yi karşılaştır, eşitse bir sonraki komutu yürüt

IFGT: Büyüklük testi

MA ile Meml'yi karşılaştır MA > Meml ise bir sonraki komutu yürüt

AND: Lojik VE işlemi

$MA \text{ AND } Meml \Rightarrow MA$

OR: Lojik VEYA işlemi

MA OR Mem1 => MA

XOR: Lojik EXOR işlemi

MA EXOR Mem1 => MA

5.6.2. Bellek Değiştirme Komutları

INC: Belleği bir arttır

Mem + 1 => Mem

DECSZ: Belleği bir azalt, bellek sıfırsa bir sonraki komutu atla

Mem - 1 => Mem, Mem=0 ise bir sonraki komutu atla

5.6.3. Bit İşleme Komutları

SBIT: Bit set komutu

1 => Mem.bit (bit 0 ile 7 arası)

RBIT: Bit reset komutu

0 => Mem.bit (bit 0 ile 7 arası)

IFBIT: Bit test komutu

Mem.bit = 1 ise bir sonraki komutu yürüt

5.6.4. Bellek aktarma komutları

LD: Belleğe yükleme komutu

Mem1 => MA

LD: X'i azaltarak yada arttırarak belleğe yükleme komutu

Mem(X) => A, X±1 (ya da ±2) => X

ST: Belleğe yazma komutu

A => Mem

X: Yer deęiřtirme komutu

A <=> Mem yer deęiřtir

X: X'i azaltarak yada arttırarak yer deęiřtirme komutu

A <=> Mem(X), $X \pm 1$ (ya da ± 2) => X

PUSH: Belleęi yığına sakla

W => W(SP), $SP+2$ => SP

POP: Yığından belleęe sayı yükle

$SP-2$ => SP, W(SP) => W

LDS: B'yi azaltıp ya da arttırarak A'ya yükleme ve bir sonraki komutu řartlı olarak atlama

Mem(B) => A, $B \pm 1$ (ya da ± 2) => B ve B, K'dan küçük ya da büyükse bir sonraki komutu atla

XS: B'yi azaltarak ya da arttırarak belleęi yer deęiřtirme ve bir sonraki komutu řartlı olarak atlama

Mem(B) <=> A, $B \pm 1$ (ya da ± 2) => B

5.6.5. Yazıcıya ivedi yükleme komutları

LD A: A'ya ivedi yükle imm => A

LD B: B'ye ivedi yükle imm => B

LD K: K'ya ivedi yükle imm => K

LD X: X'e ivedi yükle imm => X

LD BK: B ve K'ya ivedi yükle imm => B, imm' => K

5.6.6 Akümülatör ve C komutları

CLR A: Akümülatörü sil

$\emptyset \Rightarrow A$

INC A: A'yı bir arttır

$A+1 \Rightarrow A$

DEC A: A'yı bir azalt

$A-1 \Rightarrow A$

COMP A: A'nın evriğini al

$A' \Rightarrow A$

SWAP A: A'nın dört bitlik parçalarını yer değiştir

$A15:12 \Leftarrow A11:8 \Leftarrow A7:4 \Leftarrow A3:\emptyset$

RRC A: A'yı elde üzerinden sağa döndür

$C \Rightarrow A15:.. \Rightarrow .. \Rightarrow A\emptyset \Rightarrow C$

RLC A: a'yı elde üzerinden sola döndür

$C \Leftarrow A15:.. \Leftarrow ... \Leftarrow A\emptyset \Leftarrow C$

SHR A: A'yı sağa ötele

$\emptyset \Rightarrow A15:.. \Rightarrow .. \Rightarrow A\emptyset \Rightarrow C$

SHL A: A'yı sola ötele

$C \Leftarrow A15:.. \Leftarrow .. \Leftarrow A\emptyset \Leftarrow \emptyset$

SET C: Elde bitini bir yap

$1 \Rightarrow C$

RESET C: Elde bitini sıfır yap

$\emptyset \Rightarrow C$

IF C: Elde biti bir mi

Elde biti bir ise bir sonraki komutu yürüt

IFN C: Elde biti sıfır mı komutu atla
Elde biti sıfır ise bir sonraki komutu yürüt

5.6.7. Dallanma ve Kontrol Komutları

JSRP: Tablodan alt programa dallan
PC => W(SP), SP+2 => SP, W(&tablo) => PC

JSR: Alt programa bağlı olarak dallan
PC => W(SP), SP+2 => SP, PC+rel& => PC, rel& = +1025 ile
-1023

JSRL: Uzak mesafeli alt programa bağlı olarak dallan
PC => W(SP), SP+2 => SP, PC+rel& => PC

JP: Yakın mesafeye bağlı dallan
PC+rel& => PC, rel& +32 ile -31

JMP: Bağlı dallan
PC+rel& => PC, rel& +257 ile -255

JMPL: Bağlı uzak dallan
PC+rel& => PC

JID: PC+A'ya dolaylı dallan
PC+A => PC ve Mem(PC)+PC => PC

JIDW:

NOP: işlem yapma
PC+1 => PC

RET: Alt programdan geri dön
SP-2 => SP, W(SP) => PC

RETS: Alt programdan geri dön ve bir sonraki komutu atla
 SP-2 => SP, W(SP) => PC ve komutu atla

RETI: Kesme programından geri dön
 SP-2 => SP, W(SP) => PC, tekrar kesmeye izin

5.7. HPC Donanım Birimleri ve Yonga Bacaklarının Tanımlanması

MOS transistörlerin kullanıldığı herhangi bir devre tasarımında olması gerektiği gibi HPC16083 mikrodenetleyicisinin lojik yapısı da elektrostatik boşalmalara karşı elektronik koruma devreleri içermektedir. Yonga üzerindeki tüm uçlarda diodlu koruma düzeni sağlanmıştır. HPC mikrodenetleyicisinde kullanıcı ortamına çıkan 68 uç bulunmaktadır. HPC kılıf yapısı Şekil 5. .. da görülmektedir.

5.7.1. Giriş/Çıkış Uçları

A0'dan A15'e kadar olan uçlar, herbirinin giriş ya da çıkış olarak programlanabileceği, veri yönü yazıcısını içeren (DIR A) 16 bitlik iki yönlü giriş/çıkış çevrebirimi Port A'yi oluşturur. UPI özelliği kullanılmak istendiğinde Port A, ana işlemcinin veri, durum ve komutları üzerinden aktarabileceği, bir veri yolu şeklini alır. HPC harici bir belleğe erişmek üzere çalıştırılmak istenirse Port A çoğullanmış adres/veri yolu olarak kullanılır.

B0'dan B15'e kadar olan uçlar da Port A'ya benzeyen yapıda 16 bitlik çift yönlü bir giriş/çıkış portu olan Port B'yi oluşturur. PortB veri yönü yazıcısı (DIR B) ve fonksiyon yazıcısı (BFUN) ile birlikte herbir ucun ayrı ayrı değişik fonksiyonları gerçekleştirmek üzere aşağıda gösterildiği gibi kullanılabilir.

B0:	TDX	UART veri çıkışı
B1:		
B2:	CKX	UART saat giriş ve çıkışı
B3:	T2IO	T2 zamanlayıcısı giriş/çıkış ucu
B4:	T3IO	T3 zamanlayıcısı giriş/çıkış ucu
B5:	SO	MICROWIRE/PLUS çıkışı
B6:	SK	MICROWIRE/PLUS saat giriş/çıkışı
B7:	HLDA	Tutma bilgilendirme çıkışı
B8:	TS0	Zamanlayıcı eşzaman çıkışı
B9:	TS1	Zamanlayıcı eşzaman çıkışı
B10:	UA0	UPI modu için adres 0 girişi
B11:	WRRDY	UPI modu için yazma hazır çıkışı
B12:		
B13:	TS2	Zamanlayıcı eşzaman çıkışı
B14:	TS3	Zamanlayıcı eşzaman çıkışı
B15:	RDRDY	UPI modu için okuma hazır çıkışı

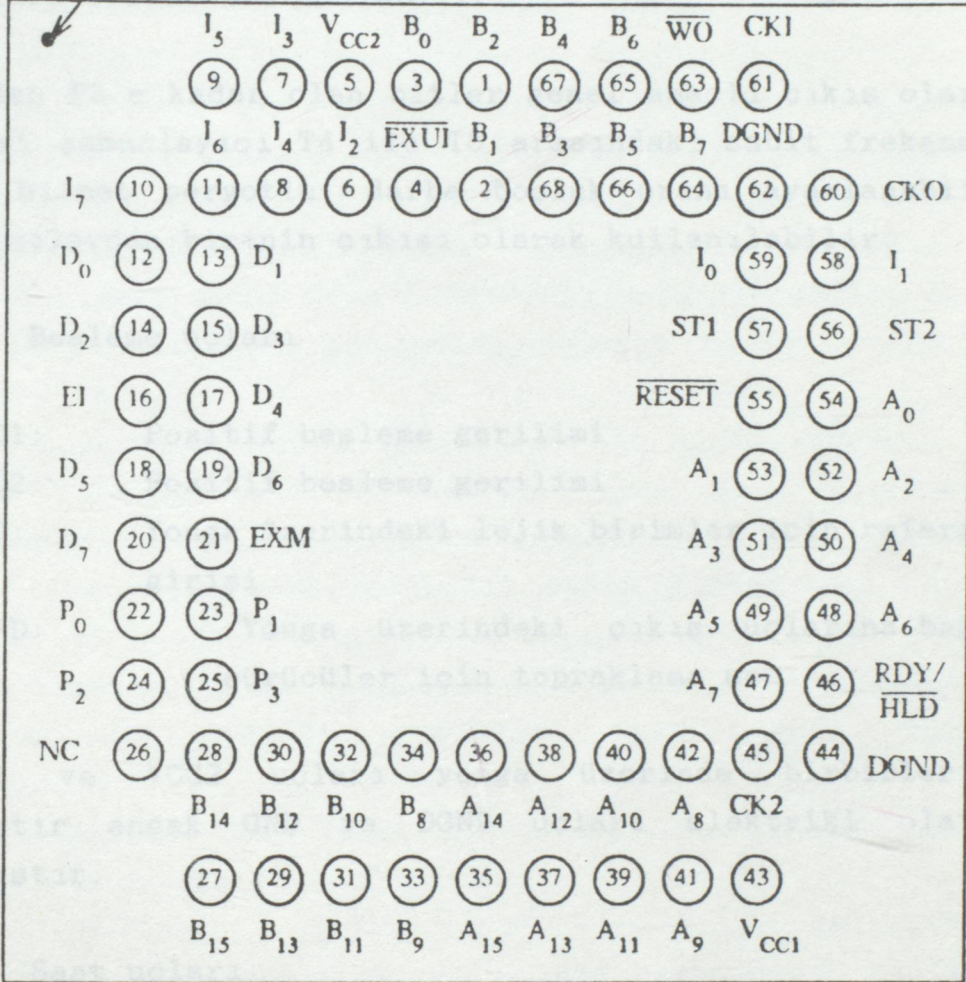
HPC harici belleğe erişmek üzere kullanıldığında Port B'nin dört biti aşağıdaki gibi kullanılır. (Bu modun seçilmesiyle adı geçen 4 bit için önceden seçilen fonksiyonlar iptaledilmiş olur).

B10:	ALE	Adres tutma izni çıkışı
B11:	WR	Yazma çıkışı
B12:	HBE	Yüksek anlamlı byte izin çıkışı
B15:	RD	Yazma çıkışı

I0'dan I7'ye kadar olan bitler genel amaçlı giriş çevrebirimi PortI ve aşağıdaki fonksiyonları sağlayan giriş uçları olarak kullanılabilir.

I0:		
I1:	NMI	Maskelenemiyen kesme girişi
I2:	INT2	Kesme 2 / Giriş yakalama 2 / UPI URD
I3:	INT3	Kesme 3 / Giriş yakalama 3 / UPI UWR
I4:	INT4	Kesme 4 / Giriş yakalama 4
I5:	SI	MICROWIRE veri girişi

MARKALAMA NOKTASI



GL-52-0-U

Şekil 5.2 HPC 16083-46083 için uç bağlantıları

Not: Aynı uç bağlantı gösterimi HPC MOLE emulatrör kablosu üstten görünüşü için de aynıdır.

I6: RDX UART veri girişı

I7:

D0'dan D7'ye kadar olan bitler genel amaçlı sayısal giriş uçları olarak kullanılabilir.

P0'dan P3'e kadar olan bitler genel amaçlı çıkış olarak ve herbiri zamanlayıcı T4 ile T8 arasındaki sabit frekanslı değişken hizmet peryotlu, darbe boşluk oranı ayarlanabilir zamanlayıcılardan birinin çıkışı olarak kullanılabilir.

Besleme uçları

VCC1: Pozitif besleme gerilimi

VCC2: Pozitif besleme gerilimi

GND: Yonga üzerindeki lojik birimler için referans girişı

DGND: Yonga üzerindeki çıkış uçlarına bağlı sürücüler için topraklama ucu

VCC1 ve VCC2 uçları yonga üzerinde birbirlerine bağlanmıştır ancak GND ve DGND uçları elektriki olarak yalıtılmıştır.

Saat uçları

CKI: Osilatör girişı

CKO: Osilatör çıkışı

CK2: Saat çıkışı (frekansı CKI/2)

CKI ve CKO uçları genellikle dışarıdan bir kristal uçlarına bağlanır.

Diğer uçlar

WO: Bu uçtaki lojik 0 seviye WATCHDOG lojik tarafından istenmeyen bir durumun ortaya çıktığını göstermektedir.

ST1: Bu uç mikrodenetleyicinin yeni komuta ait opkodun okunma çevriminde olduğunu gösterir.

T2: ALE'nin durumuna bağlı olarak mikroişlemcinin durumunu gösterir.

RESET: işlemcinin çalışmaya yeni baştan başlaması için aktif lojik 0 giriştir.

RDY/HLD: ENIR yazıcısının 0. bitinin durumuna bağlı olarak iki farklı şekilde kullanılır. Birinci kullanım şekli yavaş bellekler için veri yolu okuma çevrimini uzatmak amacıyla hazır (READY) giriştir. İkinci kullanım şekli ise doğrudan bellek erişimi için (DMA) veri yolunu boşaltan tutma (HOLD) isteği giriştir. Normalde işlemci bu ucu tutma ucu olarak yorumlar.

EXM: Dışarıdan bellek erişimine izin girişi ucudur. Bu uç yonga içindeki kalıcı belleği devre dışı bırakır ve işlemci dışarıdaki bir belleği adresleyecek şekilde çalışır.

EI: Dışarıdan maskelenebilir kesme girişi ucu. Aktif lojik 0 ya da 1 olacak şekilde ya da inen ya da çıkan kenara duyarlı olacak şekilde programlanabilir. Aynı zamanda 4. giriş yakalama (EICR) yazıcısı için giriş olarak kullanılabilir.

EXUI: Harici UART kesme giriş ucu. Seviye duyarlı ve aktif lojik 0 seviye ile çalışır.

NC: Bağlantı olamayan uçlar. Bu uçlar gelecekteki kullanımlar için ayrılmıştır. Bu uçlara herhangi bir bağlantı yapılmamalıdır.

5.8. HPC Zamanlayıcıları

HPC T0'dan T8'e kadar 9 adet 16 bit zamanlayıcıya sahiptir. T0 zamanlayıcısı üç adet yakalama yazıcısına (I2CR, I3CR, I4CR) sahip yukarı sayan bir sayıcıdır. (I3CR ve I2CR, T1 ve ona ait giriş yazıcısı R1 olarak da kullanılabilir.) T8 zamanlayıcısı her zaman T0 zamanlayıcısının değerine sahiptir. T8 dördüncü giriş yakalama yazıcısı EICR için zaman bazı oluşturur. T1'den T7'ye kadar olan zamanlayıcılar kendilerine ait taşmalardan

sonra yani deęerlerinin otomatik olarak yüklendięi R1'den R7'ye kadar yazıcıya sahip ařaęı sayan sayıcılardır. T2'den T7'ye kadar olan zamanlayıcıların herbirinin kendine ait çıkış işaretleri vardır (T2 aynı zamanda 5 çıkışı etkileyebilmektedir). T2 ve T3 zamanlayıcılarına dışarıdan iki ayrı saat girişı uygulanabilir. HPC'de T0 ve T8'dengeri kalan tüm zamanlayıcılar istenilen herhangi bir anda durdurulup tekrar çalıştırılabilir. Bu zamanlayıcılar ve onlara ait yazıcılar da aynı şekilde istenilen herhangi bir anda okunabilir. Zamanlayıcıların içerdii sayılar dışarıdan asenkron bir saat işaretiyle tetiklenmeleri durumunda bile yazılımla okunmaları esnasında deęerleri garanti edilmiştir. Zamanlayıcı giriş yazıcısı ya da zamanlayıcını kendisine yazılım tarafından yazılan her deęer zamanlayıcının çalışıp çalışmama durumunda olmasına bakılmaksızın geçerlidir. Zamanlayıcının kendisine yapılan yükleme ya da taşma sonucu meydana gelen yükleme ile zamanlayıcıya ait yazıcının deęeri deęişmez. Bir zamanlayıcının çalıştırılmasından sonra yazıcısının deęerinin sabit bırakılması zamanlayıcı taşmalarının, ilişkin yazıcının deęerine baęlı düzenli zaman aralıklarıyla gerçekleşmesine neden olur. Zamanlayıcının otomatik yüklenmesi işlemi 0000'dan FFFF'e taşmanın gerçekleştięi anda yapılır. Zamanlayıcılar kullanılarak n sayısına bölme yapmak için giriş yazıcısı n-1 sayısı ile yüklenmelidir.

HPC'de tüm zamanlayıcılar üzerinden kesme üretmek mümkündür.

5.7.1. HPC'de Zamanlayıcı işlemleri

T0 CKI/16 hızıyla serbest olarak çalışır. Temel kullanım amacı yazılım ile zaman aralıklarının ölçülmesidir. Bununla birlikte T0 Watchdog lojik için referans süre ve güç tüketimi kısıtlama için Idle modu denetim işareti üretir. Bu nedenlerle T0'ı durdurmak mümkün deęildir. Bununla birlikte I4CR kullanılarak dolaylı yoldan içerięi okunabilir. T0'ın

kendisine ait çıkışa bağlı bir ucu yoktur ancak taşma anlarında kesme üretmek üzere programlanabilir. TMMODE yazıcısının 0-3 bitleri T0'ın taşma durumlarını gözlemek ve kesme üretimini kontrol etmek için kullanılır.

T0'ın kendisine ilişkin I2CR, I3CR ve I4CR olarak adlandırılan üç adet 16 bitlik yakalama yazıcısı vardır. Bu yazıcılar kendilerine karşılık gelen uçlardan kesme gelmesi durumunda T0'dan yüklenecek şekilde programlanabilir. Kesme için uygun kenar seçimi IRCD yazıcısı üzerinde gerçekleştirilir. Bu üç uç aynı zamanda seçilen kenarlarda kesme üretecek şekilde de programlanabilir. Bir yakalama işlemi gerçekleşmesi için ENIR yazıcısındaki uygun kesme izin biti (I2,I3,I4) set edilmelidir. (Ancak GIE bitinin set olması gerekli değildir.) Yakalama yazıcılarının ikisi I3CR ve I2CR sırasıyla T1 yazıcısı ve T1 giriş yazıcısı R1 olarak düzenlenebilir.

I4CR yakalama yazıcısı I4 ucundan bağımsız olarak T0'dan sürekli yüklenecek şekilde programlanabilir. Bu özellik T0'ın okunabilmesi için kullanılabilecek tek yöntemdir. T8 de T0 gibi yukarıya doğru sayar ve dördüncü giriş yakalama yazıcısı EICR için zaman bazı üretir. EI ucunda uygun kenar görüldükten sonra T8'deki değer EICR yakalama yazıcısına yüklenir. Bu nedenle dört giriş yakalama yazıcısının da (I2CR,I3CR,I4CR,EICR) I2,I3,I4 ve EI uçlarında uygun kenar görüldüğünde T0 zamanlayıcısından yüklendiği düşünülebilir.

T1 zamanlayıcısı ve ona ilişkin R1 yazıcısı T0 yakalam yazıcısı I3CR ve I2CR olarak programlanmadığı durumda mevcuttur. Bu düzenleme T0CON yazıcısındaki T0 ORG biti ile yapılır. T0 ORG "0" ise yakalama yazıcısı fonksiyonları, T0 ORG "1" ise zamanlayıcı fonksiyonları seçilmiş olur. T1 çalıştırıldığı durumlarda CKI/16 hızıyla geriye doğru sayar. T1'e ait bir çıkış ucu bulunmamakla birlikte T1'den taşma durumlarında kesme oluşması sağlanabilir. T1 TMMODE yazıcısının 4-7. bitleri ile kontrol edilir.

T2 ve T3 en esnek yapıdaki HPC zamanlayıcılarındandır. T2 ve T3 birbirlerinden bağımsız olarak CKI'ya ya da dışarıdan gelen saat işaretini DIVBY yazıcısında belirtilen 14 farklı orandan biriyle bölerek sayar. T2 aynı zamanda T3'ten gelen taşmalarla da tetiklenebilir. DIVBY yazıcısında yapılacak uygun bir seçimle T2 ve T3 sayıcı çiftleri bir adet 32 bitlik sayıcı (R3 yazıcısındaki değere bağlı olarak) oluşturabilir. T2 ve T3'e harici girişler (DIVBY yazıcısı ile seçilmişlerse) Port B'nin B3 ve B4 bitlerinden alınır ve zamanlayıcılar herbir düşen kenarda bir eksiltir. Bu uçlar hem normal giriş/çıkış hem de zamanlayıcı çıkışı olarak çalışabileceğinden programlanmaları esnasında BFUN ve DIRB yazıcıları ile diğer zamanlayıcı fonksiyonlarını seçen yazıcılara dikkat edilmelidir.

T2 TMMODE yazıcısının 8-11 bitleri ve T3 TMMODE yazıcısının 12-15 bitleri ile kontrol edilir.

T4'den T7'ye kadar olan zamanlayıcılar hep birlikte darbe genişlik modülasyonlu (PWM) zamanlayıcıları olarak adlandırılır. Bu zamanlayıcılar CKI/16 hızıyla geriye sayan zamanlayıcılardır ve 0000'dan FFFF değerine taşma oluştuğunda bunlara ilişkin yazıcılardan yeni değerleri yüklenir. Herbir zamanlayıcının kendine ait bir çıkış ucu vardır ve taşma durumunda kesme üretecek şekilde programlanabilir. T4'den T7'ye kadar olan zamanlayıcılar PWMODE yazıcısı üzerinden kontrol edilirler.

5.7.2. Zamanlayıcı Çıkışları

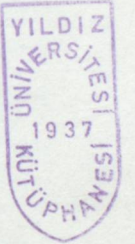
HPC'de T2'den T7'ye kadar olan zamanlayıcı birimlerinden dış dünyaya doğrudan işaretler alınabilir.

T2'nin 5 adet çıkış ucu vardır ve bunlar bağımsız olarak çıkış şeklinde seçilebilir. B3 (T2IO), B8 (TS0), B9 (TS1), B13 (TS2), B14 (TS3) olarak düzenlenebilir. TS0-TS3

uçları zamanlayıcı senkronize çıkışları olarak bilinir. T3 zamanlayıcısının sadece bir çıkış ucu B4(T3IO) vardır. T4'den T7'ye kadar olan zamanlayıcıların herbirine ait P0 (T4OUT)'dan P3(T7OUT)'a kadar birer çıkış ucu vardır.

Bu uçlardan herhangi biri zamanlayıcılara bağlı olarak seçildiğinde uçların lojik konumu herbir zamanlayıcı taşmasında durum değiştirir. Eğer uçlar normal çıkış olarak programlanırsa uçların sayısal durumu ilişkin çıkış yazıcısının değerini içerir.

Farklı zamanlayıcı çıkışları gerçek zamanda aynı anda değişmez. Bunun nedeni sayıcıların tam olarak eşzamanlı saymamalarından kaynaklanmaktadır. Zamanlayıcılar T0-T3 ve T4-T7 olmak üzere iki bağımsız grupta olarak ayrılmışlardır ve herbir zamanlayıcı CK2'nin iki artımında tazelenmektedir. Ancak TS0-TS3 ve T2IO uçları T2 zamanlayıcısı üzerinden çalıştıkları için tamamen eşzamanlı olarak değişmektedir.



5.7.3. Zamanlayıcı Yazıcıları

HPC zamanlayıcılarını programlamak için 14 adet 16 bitlik yazıcı vardır. Port P yazıcısının tüm bitleri reset anında sıfırlanır. Ancak diğer yazıcıların içerikleri reset anında değer değiştirmez ve besleme kaynağı yeni açıldığında değerleri tanımsızdır.

PORTB, DIRB ve BFUN yazıcıları T2 ve T3 zamanlayıcılarına dışarıdan saat girişlerini ve zamanlayıcılardan çıkışları kontrol eder. Bununla birlikte zamanlayıcılardan bağımsız fonksiyonları da vardır.

I4CR: I4 ucunun programlanan kenerında T0 zamanlayıcısından yükleme yapan 16 bitlik yakalama yazıcısıdır, 0180H adresinde yerleştirilmiştir. T0'ın sürekli

gözlenebilmesi için T0'dan sürekli yüklenecek şekilde de programlanabilir. Bu seçim TOCON yazıcısındaki C4F biti ile yapılır.

I3CR/T1: I3 ucunun programlanan kenarında T0'dan değer yükleyen 16 bitlik yakalama yazıcısıdır, 0182H adresinde yerleştirilmiştir. TOCON yazıcısındaki T0ORG biti set edildiğinde bu yazıcı T1 zamanlayıcısı olarak çalışır.

I2CR/R1: I2 ucunun programlanan kenarında T0'dan değer yükleyen 16 bitlik yakalama yazıcısıdır 0184H adresinde yerleştirilmiştir. T0CON yazıcısının T0ORG biti set edildiğinde T1 zamanlayıcısı için R1 giriş yazıcısı olarak kullanılır.

EICR: EI ucunun programlanan kenarında T8 zamanlayıcısından değer yükleyen 16 bitlik yakalama yazıcısıdır, 015EH adresinde yerleştirilmiştir.

R2: 0186H adresindeki 16 bitlik T2 giriş yazıcısıdır.

T2: 0188H adresindeki 16 bitlik T2 yazıcısıdır.

R3: 018AH adresindeki 16 bitlik T3 giriş yazıcısıdır.

T3: 018C adresindeki 16 bitlik T3 yazıcısı

DIVBY: T2, T3, UART ve MICROWIRE için saat işareti bölme oranlarını belirleyen yazıcıdır. 12-15 bitler T3 için, 8-11 bitler T2 için, 4-7 bitler UART için, 0-3 bitler MICROWIRE için saat bölme oranını 16 ile 131072 arasında ikinin katları olacak şekilde belirler.

T0CON: T0 düzenleme yazıcısı

T0C4F biti ile I4CR yazıcısını fonksiyonu belirlenir. "1" ise T0'dan sürekli yükleme, "0" ise I4'te uygun kenarda yükleme. T0ORG biti T0 yazıcılarını düzenler. "1" ise I2CR R1 ve I3CR T1 konumuna geçer, "0" ise I2CR I2'deki, I3CR I3'teki uygun kenarda T0'dan yüklenir.

EICON: EI ucundan gelecek kesmeleri kontrol etmede

kullanılır.

ACK biti EI bilgi biti herhangi bir kesme geldikten sonra bu bitin sıfırlanması ile IRPD yazıcısındaki EI biti sıfırlanır. MODE bit EI'daki kesmenin kenar ya da seviye duyarlı olacağını belirler.

POL biti MODE bitine bağlı olarak kenar ya da seviyenin yönünü belirler.

TMMODE: T0-T3 zamanlayıcılarını kontrol eden 0190H adresindeki 16 bitlik yazıcıdır.

PWMODE: T4-T7 zamanlayıcılarını kontrol eden 0150H adresindeki 16 bitlik yazıcıdır.

TIE biti ilişkin zamanlayıcıdan kesme iznine imkan sağlar. Bu bitin "1" olması kendisine ait zamanlayıcıdan kesme gelmesine izin verir.

TIP biti Zamanlayıcıdan gelen kesmenin göstergesidir. TIP ve TIE set olduğu sürece kesme isteği geçerli kalır. TIP biti yalnızca okunabilen bir bittir ancak kendisini ilgilendiren ACK bitine "1" yazılarak sıfırlanabilir. TIP biti TIE üzerinden kesme izni verilmiş olmasına bakılmaksızın zamanlayıcı taşmalarında "1" konumunu alır.

STP biti zamanlayıcıyı çalıştırıp durdurmak için kullanılır. "1" ise zamanlayıcı durur "0" ise sürekli çalışır.

ACK biti Zamanlayıcı kesme bilgi bitidir. Bu bite "1" yazılarak ilişkin TIP biti sıfırlanır. Bu bit yalnızca yazılabilir türdendir, okunduğunda sürekli "0" değeri okunur.

PORTP: Port P çıkış kontrol yazıcısıdır, T4'ten T7'ye kadar PWM zamanlayıcılarının herbirini ayrı ayrı programlamada kullanılır.

OUT biti Port P'nin bitlerinin çıkış seviyeleini değiştirir.

TEN biti Port P'nin OUT bitlerinden ya da zamanlayıcı taşmalarından etkileneceğini belirler. "1" seviye çıkış uçlarının zamanlayıcı taşmaları ile sürüleceğini gösterir ve OUT biti yazılımla sadece okunabilir duruma gelir. "0" seviye ise çıkış uçlarının OUT bitlerine bağlı olarak değişeceğini

gösterir.

5.8. HPC Kesme Hizmetleri

HPC'de sekiz değişik kaynaktan gelebilecek kesmeleri desteklemek üzere vektörlü kesme hizmet birimine sahiptir. Kesme kaynakları RESET, NMI, I2, I3, I4, EI, Zamanlayıcılar ve UART (EXUI)'dan herhangi biri olabilir.

Kesme hizmetlerinin yürütülmesi için aynı zamanda birden fazla kesme isteği gelmesi durumunda kesmeler arasında bir öncelik sırası belirlenmiştir. En yüksek öncelik sırasına sahip kesme hizmeti ilk önce yürütülür.

Kesme öncelik sırası

Vektör adresi Kesme kaynağı Öncelik sırası

FFFFE	RESET	Ø (en yüksek)
FFFFC	NMI	1
FFFFA	I2	2
FFF8	I3	3
FFF6	I4	4
FFF4	TIMER	6
FFF2	UART yada EXUI	7
FFF0	EI	8 (en düşük)

RESET ve NMI maskelenemeyen kesmelerdir. Diğer kesmeler ENIR yazıcısının ilişkin bitleri yardımıyla gerektiğinde izin verilip, gerektiğinde iptal edilebilir. Kesme kaynağına ait bitin set edilmesiyle kesme izni verilir, bitin reset edilmesi kesme iznini geçersiz kılar. HPC'de aynı zamanda maskelenebilir tüm kesme isteklerine aynı anda izin verilebilir. Bu işlem ENIR yazıcısındaki GIE biti yardımıyla gerçekleştirilir.

HPC'deki çevrebirimlere ait kesmeler bir kez oluştuklarında ya çevrebirime ait kesme hizmetinin yürütülünceye ya da kesme isteği çevrebirim üzerinde susturuluncaya kadar geçerli kalır.

5.8.1. Kesme Hizmetlerinin Yürütülmesi

Derhal yürütmeye giren RESET kesmesi hariç HPC'de diğer tüm kesmeler yürütülmekte olan komut tamamlandıktan sonra işleme sokulur.

Kesme isteğinin alınmasından sonra HPC ENIR yazıcısının CGIE bitini PSW yazıcısına kopyalar. Daha sonra GIE biti gelebilecek tüm maskelenebilir kesme isteklerin önlemek için reset edilir. Kesme hizmet programından geri dönüş adresi yığına atılır ve yığın gösterici (SP) iki arttırılır. Son olarak program sayacına kesme isteği gelen vektörün içeriği yüklenir ve kesme hizmet programı yürütmeye girer.

Kesme hizmet programı esnasında gelebilecek kesme istekleri GIE bitinin kesme hizmet programı içinde tekrar set edilmesiyle sağlanabilir aksi halde Yeni kesme istekleri ancak RETI komutu yürütüldükten sonra işlenebilecektir. RETI komutu ile aynı zamanda yığından kesilen programa geri dönüş adresi program sayacına yüklenir. PSW yazıcısındaki CGIE biti maskelenemeyen kesme hizmet programından geri dönüşte kullanılmaktadır. NMI altprogramında ana programa RETI komutu ile geri dönmek için PSW'deki CGIE bitinin "1" olduğu görülmelidir, eğer CGIE biti "0" ise sadece RET komutu ile ana programa geri dönülebilir.

5.8.2. Kesme Kontrol Yazıcıları

HPC kesme hizmetlerinin yürütülmesini 4 adet yazıcı ile gerçekleştirmektedir. Bunlar kesme izin yazıcısı (ENIR), kesme durum yazıcısı (IRCD), kesme düzenleme yazıcısı (EICON) ve kesme işaretleme yazıcısı (IRPD)'dir.

ENIR yazıcısı tüm maskelenebilen kesmelere ait kesme izin bitlerini ve genel kesme izin bitini GIE içinde bulundurur. GIE biti herhangi bir kesme hizmeti yürütülürken tüm maskelenebilir kesmeleri engellemek için kullanılmıştır.

IRCD yazıcısı I2, I3, I4, kesme kaynaklarından gelecek kesmelerin kenar tipini belirler. Seçilen bu kenar aynı zamanda zamanlayıcı bölümündeki ilgili yakalama yazıcısına yapılacak yüklemenin kenarını da belirler. IRCD yazıcısı aynı zamanda kesme hizmetleriyle ilgisi olmayan düzenleme bitlerini de içerir.

EICON yazıcısı EI ucundan gelebilecek kesmenin dört farklı tipini seçmekte kullanılır. Aynı zamanda IRPD yazıcısındaki EI bitini sıfırlamaya yarayan bilgi bitine (ACK) de sahiptir.

IRPD yazıcısı kesme isteği gelen kesme kaynaklarını gösterir. Bu yazıcıda her bir kesme vektörü için bir bit bulunur. Herhangi bir kesme isteği oluşması durumu ile IRPD yazıcısındaki kesme kaynağına ait bit "1" olur. IRPD yazıcısındaki kesme isteklerine ait bitler kesme isteklerine izin verilmiş olması durumuna bakılmaksızın "1" olurlar. I2, I3, I4'e ait, kenar tetiklemeli kesme isteklerinde, IRPD yazıcısında bunlara ait bitler, HPC kesme hizmet programına girer girmez sıfırlanırlar. IRPD'deki EI biti EI ucundaki kesme isteği ile birlikte set olur, ancak kesme hizmet programına girmekle bu bit sıfırlanmaz.

IRPD yazıcısındaki NMI, UART, TIMER, EI, μ WIREDONE bitleri yalnızca okunabilir bitlerdir. Çevre birimlere ait kesme hizmetleri yürütüldükten sonra ilgili IRPD bitleri sıfırlanır. INT2, INT3, INT3, INT4 bitleri yazılımla sadece "0" yazılabilecek şekilde düzenlenmiştir. Bu gözlemlere "1" yazılması hiçbir etki yapmaz, ancak "0" yazılması ilgili kesme hizmeti yürütülmüş etkisi uyandırır.

BÖLÜM 6

SONUÇ:

Düşük güçlü ancak duyarlı konumlama gerektiren robot uygulamalarında kullanılabilecek bu çalışmada 16 bitlik bir mikrodenetleyici kullanılarak üç adet adım motorunun hızının PID denetimi sağlanmıştır. Bilindiği gibi robot eksenlerinde, dikey doğrultudaki hareketlerde, kolun yükü değişmese bile, yükün konumundaki değişmeden dolayı motora binen gerçek yük ya da motora uygulanan moment dinamik olarak değişmektedir. Bu nedenle hızın ve konumlamanın önemli olduğu uygulamalarda genelde açık çevrimle denetlenen adım motorları için bile gelişmiş kapalı çevrim denetimine ihtiyaç duyulur. Bu ihtiyaçtan yola çıkarak adım motorlarının kapalı çevrim denetim yöntemleri ele alınmıştır. Sonuç olarak güçlü zamanlayıcılara sahip ve çoklu kesme hizmetlerine cevap verebilen bir mikrodenetleyici kullanılmıştır. Çözümün mikrodenetleyici ile halledilmesinde minimum donanım kullanılmasına özen gösterilmiştir. Tasarlanan düzende, konum kodlayıcılardan gelen konum bilgileri, kesme hizmeti programlarında işlenmekte ve zamanlayıcı birimlerden adım motorları için, hesaplanan uygun gecikme sürelerine ait, adımlama darbeleri üretilmektedir. Bu düzenek yardımıyla üç ekseninde adım motoru bulunan bir robot kolu sürülebilir.

Benzer şekilde nümerik kontrollu tezgahlarda da kesici ucu hareket ettiren düzeneklerde aynı yöntem kullanılabilir. Daha fazla eksenle çalışmak için, mikrodenetleyicide mevcut diğer kesme hizmetleri ve zamanlayıcılar da kullanılabilir. Ancak bu durumda PID hesaplama süresi uzamakta ve duyarlılığı düşüren yaklaşımları denetim programına ilave etmeyi gerektirmektedir.

EK-2'de adım onaylama yöntemi ile elde edilmiş değişken yük altındaki adım/zaman grafiği verilmiştir. Denetleme işleminde her adım atma olayının ardından belli bir zaman gecikmesi kullanılmıştır. Grafikte görüldüğü gibi motorun ortalama hızı yaklaşık 550 adım/saniye'dir. Motoru uygulanan yük etkisiyle motor hızında büyük ölçüde dalgalanmalar olduğu gözlenmektedir.

EK-3'te ise değişken yük altında PID denetim yöntemi ile elde edilmiş adım/zaman grafiği verilmiştir. Grafikten görüldüğü gibi motorun ortalama hızı 1100 adım/saniye civarındadır. Açıkça görüldüğü gibi motor hızındaki dalgalanmalar adım onaylama yöntemindekine oranla oldukça azalmıştır. Ortalama 0.9 ms adım hızı ile çalışan motor yaklaşık 0.1 ms adım dalgalanmaları ile hareket etmektedir.

Sonuç olarak adım motoru hızında gözle görülebilir bir iyileşme elde edilmiştir. Adım motoru daha uygun bir mekanik düzenek kullanılarak ve daha kaliteli konum kodlayıcı (yapısal hataları azaltılmış) kullanarak, hızda ortaya çıkan dalgalanmalar daha düşük düzeylere indirilebilir. Düzenek bu haliyle de açık çevrim ya da basit kapalı çevrim denetim yöntemlerinden iyi sonuç vermektedir.

PROFORMA EK-1/177

ABCDEFGHIJKLMNÖPRSTUVYZ,

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

ABCDEFGHIJKLMNÖPRSTUVYZ.

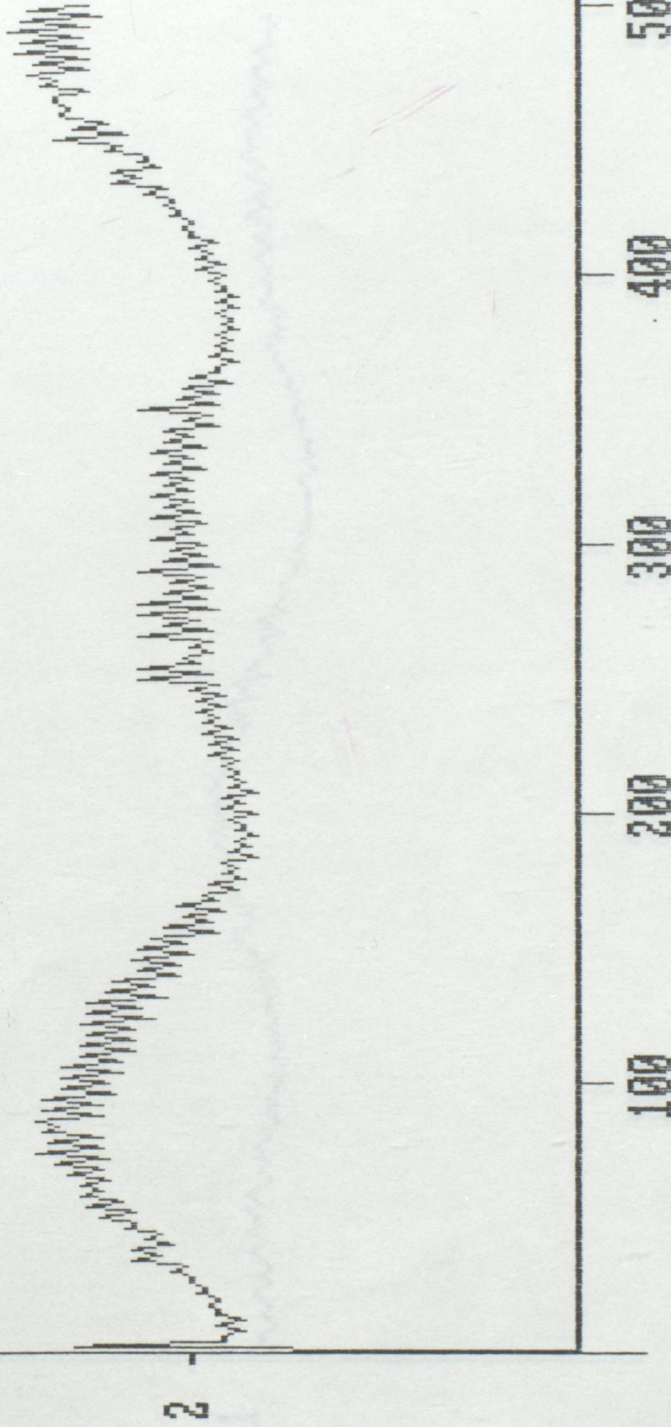
Acık çevrim denetimi kullanılarak gerçekleştirilmiş
iki eksenli konum kontrolunda elde edilmiş yazım
karakterleri örnekleri

step duration(ms)

4

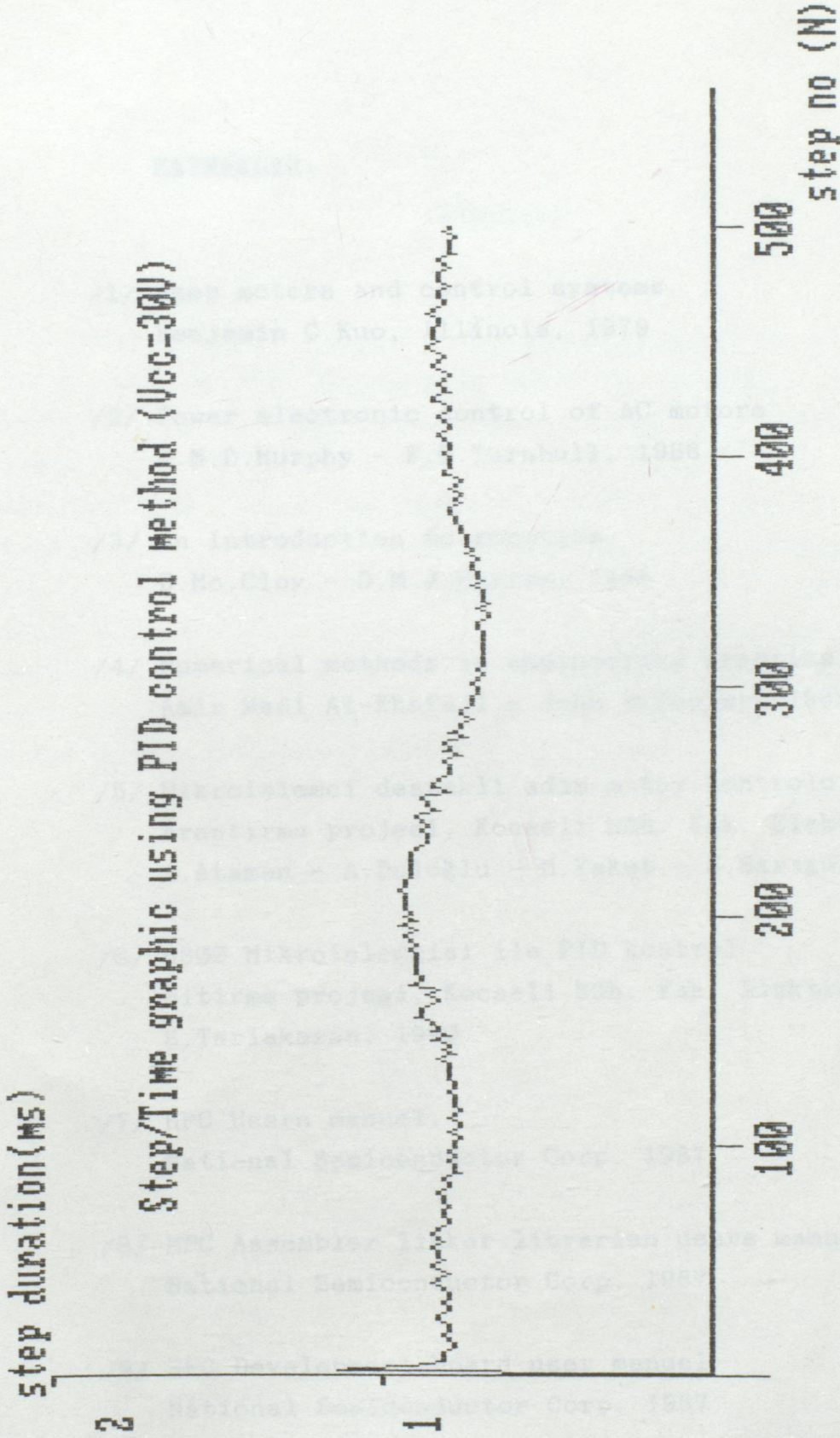
2

Step/Time graphic using step conf. method (Vcc=10V)



step no (N)

Zaman gecikmeleri kullanarak adım onaylama yöntemi ile elde edilmiş değişken yük altında adım motoru adım/zaman grafiği



PID Denetim yöntemi ile elde edilmiş değişken yük altında
adım motoru adım zaman grafiği

KAYNAKLAR:

- /1/ Step motors and control systems
Benjamin C.Kuo, Illinois, 1979
- /2/ Power electronic control of AC motors
J.M.D.Murphy - F.G.Turnbull, 1988
- /3/ An Introduction to robotics
D.Mc.Cloy - D.M.J.Harris, 1986
- /4/ Numerical methods in engineering practise
Amir Wadi Al-Khafaji - John R.Tooley, 1986
- /5/ Mikroislemci destekli adım motor kontrolu
Araştırma projesi, Kocaeli Müh. Fak. Elektronik Bl.
A.Ataman - A.Daloğlu - M.Yakut - E.Sarıgül, 1990
- /6/ 6800 Mikroislemcisi ile PID kontrol
Bitirme projesi, Kocaeli Müh. Fak. Elektronik Bl.
Ş.Tarlakazan, 1990
- /7/ HPC Users manuel,
National Semiconductor Corp. 1987
- /8/ HPC Assembler linker librarian users manuel,
National Semiconductor Corp. 1987
- /9/ HPC Development board user manuel
National Semiconductor Corp. 1987

ÖZGEÇMİŞ:

Mehmet YAKUT, 1966 yılında Karaabdülbaki'de (İZMİT) doğdu. İlk ve orta öğrenimini izmit'te tamamladı. izmit Endüstri Meslek Lisesi Elektronik Bölümü'nden 1983 yılında mezun oldu. 1983 ÖSYM sınavı ile Y.Ü. Kocaeli Müh. Fak. Elektrik Müh. Bölümüne girdi. 1984 ÖSYM sınavı ile aynı fakültenin Elektronik ve Haberleşme Müh. Bölümüne geçiş yaptı. Elektronik ve Haberleşme Müh. Bölümünden 1987 yılında mezun oldu ve aynı bölümde araştırma görevlisi olarak çalışmaya başladı. Halen Y.Ü. Fen Bilimleri Enst. Elektronik ve Haberleşme Bl. Elektronik anabilim dalında yüksek lisans programında çalışmalarına devam etmektedir.

