

**YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**STEP MOTOR KULLANILARAK LASER POINTER'IN  
YÖN KONTROLÜ**

**Elek. Müh. Mustafa Gürkan AYDENİZ**

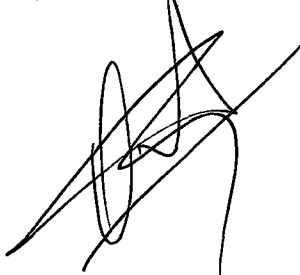
**F.B.E. Elektrik Mühendisliği Ana Bilim Dalı Elektrik Mühendisliği Programında Hazırlanan**

**YÜKSEK LİSANS TEZİ**

03.07.1998

**Tez Danışmanı : Yrd. Doç. Dr. İbrahim ŞENOL**

Prof. O. AYBAR



İbrahim ŞENOL  
Ahmeti

Y. Doç. Dr. Zehra Yumurtacı  
Ahmeti

79112

**İSTANBUL, 1998**

TEZİN  
KAYIT NO: 79112  
TARİHİ: 03.07.1998

79112



# İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ .....	iv
ŞEKİL LİSTESİ .....	v
ÇİZELGE LİSTESİ .....	vii
ÖZET .....	viii
ABSTRACT .....	ix
1. GİRİŞ.....	1
2. STEP MOTORLAR.....	2
2.1. Step Motorların Tanımı, Sınıflandırılması, Kullanım Alanları ve Avantajları.....	2
2.2. Step Motor Çeşitleri.....	5
2.2.1. Değişken relüktanslı step motor.....	5
2.2.2. Sabit mıknatıslı step motor.....	17
2.2.3. Hibrid step motor.....	21
3. STEP MOTORLARA AİT ÖNEMLİ BÜYÜKLÜKLER.....	27
3.1. Çözünürlük.....	27
3.2. Doğruluk.....	27
3.3. Tutma Momenti.....	28
3.4. Tek Adım Cevabı.....	30
3.5. Sürekli Rejim Maksimum Yük Momenti Eğrisi.....	32
3.6. Kalkışta Maksimum Yük Momenti Eğrisi.....	34
4. ELEKTROMANYETİK TEORİSİ, STEP MOTORLARIN YAPILARI VE MOMENT ÜRETMELERİ.....	36
4.1. Değişken Relüktanslı Step Motorlarda Statik Moment Oluşum.....	36
4.1.1. Sonsuz manyetik geçirgenlik durumu.....	36
4.1.2. Sabit manyetik geçirgenlik durumu.....	42
4.1.3. Manyetik doyuma ulaşma durumu.....	44
4.2. Hibrid Step Motorlarda Moment Üretilmesi.....	46
4.2.1. Analitik yaklaşım.....	47
4.2.2. Moment üretiminde sürekli mıknatısın etkisi.....	48
4.2.3. Sabit moment.....	50
4.3. Dış Yapısı, Sayısı ve Kutup Düzeni.....	51
4.3.1. Dış yapısı.....	52
4.3.2. Faz, dış ve adım sayısı ilişkileri.....	54

5.	STEP MOTORUN SÜRÜLÜŞÜ.....	57
5.1.	Tek Yönlü Sürüş.....	58
5.2.	Çift Yönlü Sürüş.....	59
5.3.	R/L Sürüşü.....	61
5.4.	Çift Seviyeli Sürüş.....	62
5.5.	Kıyıcı Regüleli Sürüş.....	62
6.	YAPILAN UYGULAMANIN YAZILIMI.....	64
6.1.	LP.pas.....	64
6.2.	Screen.pas.....	70
6.3.	Keyboard.pas.....	87
6.4.	TypesLP.pas.....	92
6.5.	StpProcs.pas.....	94
6.6.	MCtrl.pas.....	103
6.7.	StrOp.pas.....	115
6.8.	IO.pas.....	120
7.	SONUÇLAR.....	129
	KAYNAKLAR .....	130
	EKLER .....	131
	ÖZGEÇMİŞ .....	132

## SİMGE LİSTESİ

A	Stator birinci fazı
B	Stator ikinci fazı
$B$	Manyetik Akı (T)
$B_g$	Boşluktaki manyetik Akı (T)
d	Diş derinliği (m)
$f$	Kuvvet (N)
g	Hava boşluğu genişliği (m)
H	Manyetik alan (A/m)
i,I	Akım (A)
J	Yük momenti (kgm <sup>2</sup> )
$K_T$	Moment sabiti (Nm/Arad)
L	Endüktans (H)
m	Faz sayısı
n	Sarım sayısı
$N_r$	Rotor diş sayısı
$N_s$	Stator diş sayısı
p	Kutup çifti sayısı
q	Faz başına diş sayısı
r,R	Direnç ( $\Omega$ )
S1,..S4	Anahtarlar
T	Moment (Nm)
$T_A^+$	A fazı pozitif uyarılma tutma momenti (Nm)
$T_A^-$	A fazı negatif uyarılma tutma momenti (Nm)
$T_B^+$	B fazı pozitif uyarılma tutma momenti (Nm)
$T_B^-$	B fazı negatif uyarılma tutma momenti (Nm)
T1,..T8	Transistörler
t	Zaman (sn)
v,V	Gerilim (V)
w	Diş genişliği
$W_m$	manyetik enerji (J)
x	alınan yol (m)
$\alpha$	Zaman sabiti (s)
$\beta$	Zaman sabiti (s)
$\theta, \Theta$	Dönüş açısı (rad)
$\lambda$	Diş yolu (rad)
$\mu$	Manyetik geçirgenlik (H/m)
$\xi$	Elektriksel olarak dönüş açısı (rad)
$\rho$	Moment açısı (rad)
$\tau$	Moment (Nm)
$\phi, \Phi$	Manyetik akı (Tm <sup>2</sup> )
$\psi, \Psi$	Kaçak akı (Tm <sup>2</sup> )
$\omega, \Omega$	Açısal hız 1/rad.sn
$\omega_n$	Doğal frekans (1/sn)

## ŞEKİL LİSTESİ

	Sayfa
Şekil 2.1.	İki fazlı step motorun birinci fazının enerjilendirilmesi ile oluşan manyetik alan . . . . . 5
Şekil 2.2.	İki fazlı step motorun ikinci fazının enerjilendirilmesi ile oluşan manyetik alan . . . . . 6
Şekil 2.3.	İki fazlı step motorun iki fazının da enerjilendirilmesi ile oluşan manyetik alan . . . . . 7
Şekil 2.4.	İki fazlı step motorun iki fazının enerjilendirilmesi ile rotorun konumu . . . . . 7
Şekil 2.5.	İki fazlı step motorun tutma momentinin rotor pozisyonuna göre değişimi . . . . . 9
Şekil 2.6.	İki fazlı step motorun fazların ters olarak enerjilendirilmesi . . . . . 10
Şekil 2.7.	İki fazlı step motorun fazlarının sıra ile enerjilendirilmesinde statorun alacağı konumlar . . . . . 11
Şekil 2.8.	İki fazlı step motorun yarım adım çalışması durumunda birinci adım . . . 13
Şekil 2.9.	İki fazlı step motorun yarım adım çalışması durumunda ikinci adım . . . 13
Şekil 2.10.	İki fazlı step motorun yarım adım çalışması durumunda üçüncü adım . . . . . 14
Şekil 2.11.	İki fazlı step motorun yarım adım çalışması durumunda ilk 8 adımdaki rotor konumları . . . . . 15
Şekil 2.12.	İki fazlı sabit mıknatıslı step motor . . . . . 17
Şekil 2.13.	İki fazlı sabit mıknatıslı step motorda rotorun iki aynı stator kutbu arasındaki konumu . . . . . 18
Şekil 2.14.	İki fazlı sabit mıknatıslı step motorda senkronlama momentinin rotor pozisyonuna göre değişimi . . . . . 19
Şekil 2.15.	İki fazlı sabit mıknatıslı step motorda uygulanan momentin rotor Pozisyonuna göre değişimi . . . . . 20
Şekil 2.16.	İki fazlı sabit mıknatıslı step motorda momentlerin rotor Pozisyonuna göre değişimleri . . . . . 20
Şekil 2.17.	İki fazlı hibrid step motorun stator ve rotoru . . . . . 22
Şekil 2.18.	İki fazlı hibrid step motorun rotor kesiti . . . . . 22
Şekil 2.19.	Statoru ve rotoru çok dişli hibrid step motor . . . . . 24
Şekil 2.20.	Hibrid tip step motor rotoru . . . . . 25
Şekil 2.21.	Hibrid tip step motor statoru . . . . . 25
Şekil 3.1.	Faz akımının çeşitli değerleri için tutma momenti/rotor konumu eğrileri . . . . . 28
Şekil 3.2.	Hibrid motorun tutma momenti/rotor konumu eğrileri . . . . . 29
Şekil 3.3.	Step motorun tek adım cevabı . . . . . 30
Şekil 3.4.	Doğal frekansa yakın hızlarda step motorun davranışı . . . . . 32
Şekil 3.5.	Sürekli rejimde maksimum yük momenti/hız eğrisi . . . . . 33
Şekil 3.6.	Sürekli rejimde maksimum yük momenti ve kalkışta maksimum yük momenti eğrileri..... 35
Şekil 4.1.	Elektromanyetik alan içindeki demir parçası . . . . . 36
Şekil 4.2.	Step motor prensibinin açıklanabileceği basit bir model . . . . . 37
Şekil 4.3.	Kaplanmış alan . . . . . 39

Şekil 4.4.	At süresinde rotor dişinin manyetik alan tarafından $\Delta x$ kadarlık hareketi . . . . .	40
Şekil 4.5.	Sistemin manyetik enerji ve koenerji dağılımı . . . . .	46
Şekil 4.6.	İki fazlı hibrid step motor modeli . . . . .	47
Şekil 4.7.	Rotorda ve statorda aynı sayıda diş bulunan yapı . . . . .	52
Şekil 4.8.	Rotorda ve statorda farklı sayıda diş bulunan yapı . . . . .	52
Şekil 4.9.	Bir hibrid tip step motorun diş yapısı . . . . .	53
Şekil 5.1.	Tipik bir step motor sürüş sisteminin şematik düzenlemesi . . . . .	57
Şekil 5.2.	Step motor tek yönlü tipik sürüş düzeni . . . . .	58
Şekil 5.3.	İki fazlı step motor için tipik iki yönlü sürüş düzeni . . . . .	59
Şekil 5.4.	Tek veya çift yönlü sürüşlerin step motorun stator sargı bağlantıları ile bağlanabilme düzeni . . . . .	60
Şekil 5.5.	R/L sürüşünde stator sargılarının ortak hattına seri olacak uygun dirençlerin bağlanması . . . . .	61
Şekil 5.6.	Değişik gerilim değerlerindeki endüktif devredeki akım-zaman değişimi . . . . .	62
Şekil 5.7.	Kıyıcı regüleli sürüşte akım ve gerilimin zamana göre değişimi . . . . .	63



## ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2.1. 1. adımdan 5. adıma kadar stator sargılarının anahtarlama sırası . . . . .	12
Çizelge 2.2. Yarım modda çalışma için anahtarlama sıraları . . . . .	16





## ÖZET

Step motorun elektrik enerjisini adım mekanik hareketine dönüştürebilmesi ve bu hareketin pozisyon denetimi konularında gerekli oluşu bu motorun tercih edilen bir elektrik motoru olmasına neden olmuştur. Bu motorların dijital işaretlerle uyumlu çalışabilmesi ve geri besleme devresine ihtiyaç duymaması da tercih sebepleridir. Bu çalışmanın konusu ise laser pointer'ın ışınının xy eksenindeki hareketinin bilgisayar tuşları ile elde edilmesidir. Temel olarak bir yön kontrolü çalışmasıdır.

Çalışmada öncelikle step motorlar hakkında detaylı bilgiler verilerek, çeşitleri, çalışma prensipleri, avantajları ve kullanım alanları incelenmiştir. Üçüncü bölümde step motora ait önemli büyüklükler incelenmiş, dördüncü bölümde ise elektromanyetik teorisi ve step motorların yapıları ve moment oluşumları üzerinde durulmuştur. Beşinci bölümde bu motorların sürülüş metotları sınıflandırılmış ve bu metotlar açıklanmıştır. Altıncı bölümde yapılan yön kontrolü çalışmasında kullanılan iki adet hibrid tip step motorun sürülüşü için gerekli olan dijital işaretlerin elde edildiği bilgisayar yazılımı yer almaktadır. Ekler kısmında ise bu uygulamada kullanılan motorların sürücü devresi verilmiştir.

Sonuçlar kısmında bu çalışmanın biraz daha geliştirilmesi ile endüstride bir çok çalışmada kullanılabileceği ve step motor kullanılmasının nedeni hakkında bilgiler verilmiştir. Bu motorlara alternatif olarak küçük güçteki fırçasız doğru akım motorlarının da kullanılabileceği ifade edilmiştir.

## **ABSTRACT**

Capability transforming electrical energy to step mechanical energy and necessity of this motion in position control make the stepper motor preferable. And also, it is harmonious working with digital signals and not need any feed back circuits are made into account to preference. The main topic of this study is controlling of laser pointer's ray in XY axes with keyboard. Mainly it is study of way control.

In the thesis, firstly giving detailed information about stepper motors, kinds of stepper motors, running principles of them, advantages and their using area are scrutinized. In the third chapter, important values of stepper motors are examined and then, electromagnetic theory and structure of stepper motors and their torque production are presented in chapter four. In the fifth chapter driving methods of these motors are classified and explained afterwards. There is software to get digital signals for driving two hybrid stepper motors in applied way control in the sixth chapter. Driving circuits of this application is given in appendix 1.

In final part of this study, it is presented that with a little improvment of this work this can be available for many of industrial application and why we have chosen stepper motor. Also it is pointed out that alternatively brushless d.c. motors can be used in this example.

## 1. GİRİŞ

Dijital elektrik sinyallerini mekanik harekete çeviren elektrik motoru olarak tanımlanan step motorlar günümüzde yaygın olarak kullanılmaktadır. Bilgisayarlarda, robotlarda, seri yazıcılarda, XY çizicilerde, oyuncak makinalarda, daktilolarda istenilen nümerik kontrollü hareketler step motorlarla gerçekleştirilmektedir.

Hassas mekanik hareketin istenildiği yerlerde step motor kullanılmasının temel nedenleri ise güvenilir ayar elemanı olması, iyi pozisyon alması, dijital ayar sinyalleri ile çalıştırılabilmesi, ayar frekansı ile orantılı olan devir sayısının geniş ayar sahası içerisinde değiştirilebilmesi, ayar frekansı değiştirilmediği sürece sabit devir sayısında çalışması, geri besleme ünitelerine ihtiyaç duymaması, rotor dururken yüksek tutma momenti vermesi, yüksek derecede teknik güvenilirlikle uzun süre kullanılabilmesi ve alternatif çözümlerle karşılaştırıldığında daha ucuz olmasıdır.

Günümüzde güçleri miliwattlardan birkaç yüz watta kadar, moment değerleri 15 Nm' ye kadar ve mekanik olarak  $0.78^\circ$  den  $90^\circ$ 'ye kadar açısal hareket yapabilen step motorlar üretilmektedir.

Değişken relüktanslı, sabit mıknatıslı ve hibrid tip olarak üç gruba ayrılan step motorlar tez içerisinde detaylı olarak anlatılmış ve tezin uygulamasında Laser Pointer'in yön kontrolü için iki adet hibrid tip step motor kullanılmıştır.

## 2. STEP MOTORLAR

Step motor, diğer geleneksel elektrik motorlarından oldukça farklı adından da anlaşılacağı gibi hareketini adım adım yapabilen bir elektrik motorudur. Bu sebepten konum ve pozisyon kontrollerinin çok hassas olması gereken yerlerde vazgeçilemez bir motor tipi olarak diğer elektrik motorlarının önünde yer almaktadır. Asenkron motorlar ile karakteristik yönünden tamamen farklı olan step motor, DC motorlar ile bazı yönlerden aynı özelliklere sahiptir. Step motorlar bazı Türkçe literatürlerde adım motoru, yabancı literatürlerde de stepper veya stepping motor olarak da yer almaktadır.

Bu bölümde bu motorların tanımı, sınıflandırılması, kullanım alanları, avantajları ve günümüzde kullanılan en yaygın step motor çeşitlerine değinilmiştir.

### 2.1. Step Motorların Tanımı, Sınıflandırılması, Kullanım Alanları ve Avantajları

Step motor, sargılarının uygun kombinasyonda enerjilendirilmesi ile milinde dönüş açısı (adım hareketi) şeklindeki mekanik güç dönüşümünü sağlayan bir elektrik makinasıdır.

1920'li yıllarda step motorun sürülüşü özel olarak geliştirilmiş el ile yönlendirilen bir düzen ile DC kaynaktan yapılmaktaydı. 60'lı yıllara gelindiğinde hızla gelişmekte olan dijital kontrol step motorların kontrol tekniklerini de değişime zorlamıştır. Yine bu tarihlerde değişken relüktanslı motorların yanı sıra rotorda kalıcı mıknatıslar kullanılarak gerçekleştirilen step motorlar da kullanılmaya başlanmıştır. Step motorların dizaynı ve kontrolündeki en hızlı gelişmeler ve yenilikler 70'li yıllarda meydana gelmiştir. Bu yıllarda dört fazlı 1.8° adım açılı, dört fazlı 2°, 2.5° ve 5° adım açılı hibrid motorlar, 7.5° adım açılı sabit mıknatıslı motorlar ve üç ve dört fazlı değişken relüktanslı motorlar üzerinde son derece etkili değişimler meydana gelmiştir. Günümüzde ise step motorun sürülüşü, elektronik bir devre, mikroişlemci veya bilgisayar tarafından yapılmaktadır.

Step motor dinamik açıdan incelendiğinde, her bir adımında mekanik olarak bir denge noktasında olduğu görülmektedir. Motor, hareketinde kendisinin ve yükün atalet momentlerinin toplamı olan bir moment ile hareketi gerçekleştirir ve yeni bir denge

noktasına ulaşır. Başka bir deyişle “x” denge noktasından “x + adım açısı” olan yeni bir denge noktasına gelmiş olur. Motor bu hareketi sırasında boşta ise veya yük ataletinde bir düşüş meydana gelirse rotor yeni denge noktası civarında osilasyona girer. Bu olay yüksek hızlarda step motorun adım kaçırmaya veya stabil olmayan kontrol dışı hareket etmesine neden olur.

Uygulamalarda step motor kullanılırken motor sargılarının endüktansına ve momente dikkat edilmesi gerekir. Aşırı moment motorun kontrol dışı hareket etmesine neden olacağından moment değeri belirlenirken motorun maksimum momentinin %70'i ve hatta bu değerden biraz daha küçük bir moment değeri alınmalıdır.

Step motorlarda sınıflandırma, yapıları ve çalışma biçimleri bakımından yapılabilir. Aslında bir step motorun yapısı ve türü, motorun sargılarında enerji yok iken rotorun davranışı ile belirlenmektedir. Step motorlar, yapıları bakımından Değişken Relüktanslı ve Sabit Mıknatıslı Rotorlu olmak üzere iki ana gruba ayrılırlar. Ayrıca iki motor türünün de ortak özelliklerine sahip fakat kontrol açısından sabit mıknatıslı bir step motor olarak değerlendirilebilen üçüncü bir tip de Hibrid step motorlardır. Sabit mıknatıslı bir step motorda, stator sargılarında enerji yok iken, rotoru nominal moment veya daha küçük bir moment ile zorlandığında hareket etmez. Değişken relüktanslı step motorda ise rotor serbestçe hareket etmektedir.

Step motorlar bağlantı şekline göre de iki gruba ayrılırlar. Bunlar Unipolar (Tek Kutuplu) ve Bipolar (Çift Kutuplu) olarak isimlendirilirler. Unipolar step motorlar iki ana sargıya sahiptirler ve motordan altı uç dışarı çıkarılmıştır. Bunlardan dördü iki sargıya ait olan uçlardır. Diğer iki uç ise bu sargıların orta noktalarından alınmış olan uçlardır. Bipolar step motorlar da iki ana sargıya sahiptir ama dışarıya dört uç yani sadece faz sargılarının uçları çıkartılmıştır.

Step motorlar genellikle hassas konum kontrolünün gerektiği yerlerde kullanılmaktadır. Robot uygulamalarında, servo sistemlerde, xy koordinat işaretleyicilerinde, 3 eksen tarafından hareket eden nümerik kontrollü torna tezgahlarında, floppy disk sürücülerinde, şerit sürücü, şerit okuyucu ve şerit iz kontrol edicilerde, işaret basma makinalarında, uydu

antenlerinin yön kontrolünde, seri yazıcı, kelime işlem sistemlerinde, faks makinalarında, araba saatlerinde, ölçme aletlerinde kağıt sürme düzenlerinde, teleks makinalarında, imalat makinaların programlanmasında ve çeşitli uzay çalışmalarında step motor vazgeçilemez bir elektrik motoru olarak yerini almıştır.

Geleneksel elektrik motorlarına göre step motorlarının en önde gelen avantajları, sürücü devresinin maliyetinin düşük olması ve motorun oldukça küçük yapıya sahip olmasıdır. Diğer avantajları ise şöyle sıralayabiliriz:

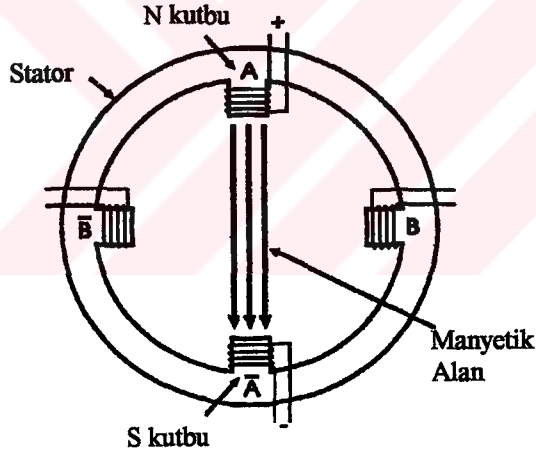
1. Veriler sayısal olarak ele alınmaktadır. Sayısal tüm devreler hem ucuz hem de çıkış hareketinin sayısal olması istenmesi halinde step motorları ideal bir çözümdür.
2. Ucuz mikroişlemci ile kontrol ve işlem rahatlıkla otomatize edilebilmektedir.
3. Bir açık çevrim sisteminde step motoru ile hız kontrolü ve doğru konuma ulaşmak mümkündür. Böylece alışılmış karasızlık problemleri önlenmiş olur.
4. Bir sabit mıknatıslı bir step motorda, aralıklı işlemlerde güç harcaması hareketsiz zaman sürecine indirgenir.
5. Adım motorlu kontrol sistemi için dizayn yöntemi son derece basit ve ucuzdur.

## 2.2. Step Motor Çeşitleri

### 2.2.1. Değişken relüktanslı step motor

Bu tip step motorda çalışma, stator ve rotor arasındaki hava aralığını minimum tutacak olan minimum relüktans durumlarının sağlanmasıyla makinenin stator ve rotorun dizaynına göre değişir.

Stator alanının uyarılması ile ( bir manyetik alan oluşturabilmek için d.c. akım akıtılarak enerjilendirilmesi ile ) rotor ileriye doğru çekilir ve sonra bu pozisyonlardan bir tanesine kilitlenir. Manyetik akı maksimum olarak akmaya başlayacaktır. Stator alan uyarılması stator alanını taşımak için değiştirildiğinde rotor minimum relüktansın yeni bir pozisyonuna taşınacaktır. Bu sefer step motor, açılmal rotor pozisyonu veya uyarılma akımında yapılacak çoklu bir değişim ile dönen bir hız sağlanarak çalıştırılabilir.



Şekil 2.1 İki fazlı step motorun birinci fazının enerjilendirilmesi ile oluşan manyetik alan

Rotor ve stator yapıları genellikle şekillendirilmiş nüve saçlarından imal edilirler. Stator duran kutuplar üzerinde alan sargılarına sahiptir.

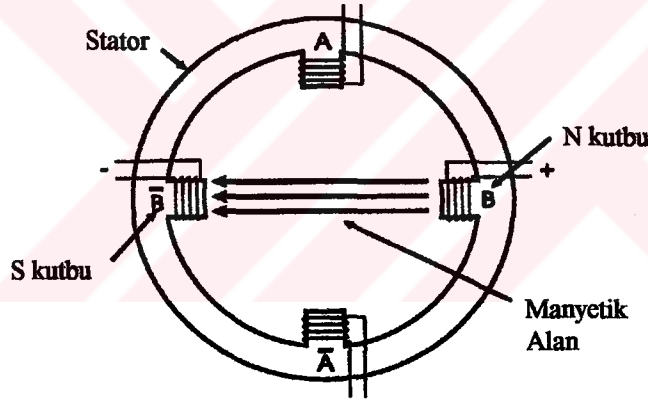
Şekil 2.1 de gösterildiği gibi statoru iki fazlı olarak göz önünde bulunduralım.

Stator üzerinde dağıtılmış elektriksel olarak izoleli sargılar ( A sargısı ve B sargısı ) bulunan dört sabit kutuplar ile yapraklı göbekli boyunduruktan oluşur.

Sargılar, doğru akım A sargısından geçtiğinde A kutbu bir N kutbu ve A- kutbu bir S kutbu oluşturacak şekilde düzenlenir ve manyetik alan gösterildiği gibi söylenebilir. Benzeri şekilde pozitif d.c. akım B sargısından geçtiğinde beklendiği gibi B ve B- de bir N ve S kutbu oluşturur.

Sargılarda ayrı ayrı veya beraber olarak akımı değiştirmek beklenen alanın polaritesini değiştirecektir.

D.C. akımın sadece B sargısından akıtılması ile oluşan manyetik alan Şekil 2.2 de gösterildiği gibidir.



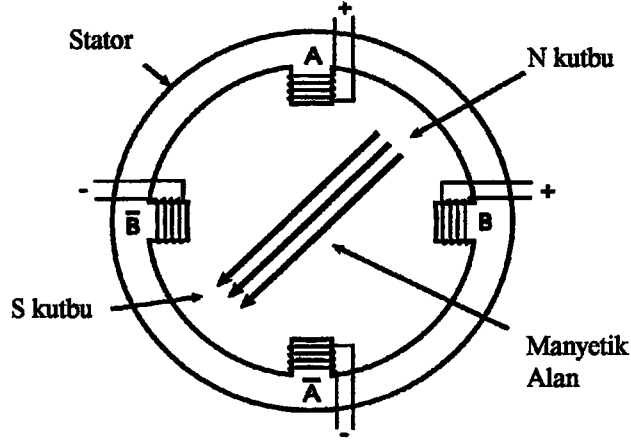
Şekil 2.2 İki fazlı step motorun ikinci fazının enerjilendirilmesi ile oluşan manyetik alan

Pozitif d.c. akımın aynı anda iki stator sargısından da ( A ve B ) akıtığımızda üretilen iki bağımsız manyetik alanlar karşılıklı olarak birbirini etkilerler ve birleşerek Şekil 2.3'de görüldüğü gibi tek bir sonuç alan meydana getirirler.

İki sargıdan da ayrı ayrı veya beraber hiç bir akım akıtılmadığında statorun merkezine yerleştirilmiş yumuşak demirli rotor harici bir güç uygulanmadığı sürece boşta kalacaktır.



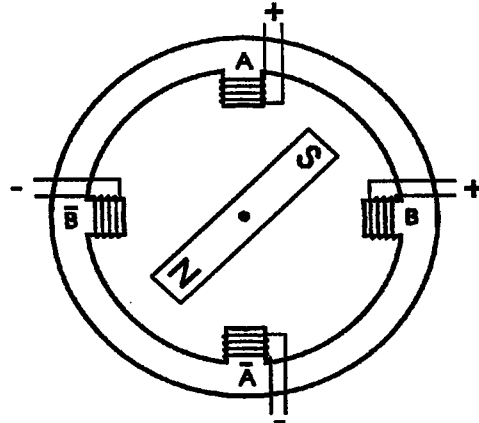
Çünkü rotor hiç bir doğal manyetik niteliğe sahip değildir ve sonuç olarak rotor ile uyarımsız stator arasında hiç bir karşılıklı etki oluşmayacaktır.



Şekil 2.3 İki fazlı step motorun iki fazında enerjilendirilmesi ile oluşan manyetik alan

Bununla beraber eğer pozitif d.c. akım A ve B sargılarından akıtılırsa, üretilen sonuç manyetik alan ( Şekil 2.3 ) rotorun endüklenmiş bir mıknatıs olmasına neden olacaktır. Rotor stator alan kutuplarına doğru farklı kutupların çekmesi aynı kutupların itmesi prensibinden dolayı etkilenecektir.

Kısa bir periyottan sonra ( atalet, sürtünme ve osilasyon etkilerinden dolayı ) rotor kendi kendine manyetik alanla Şekil 2.4 de gösterildiği gibi hiza alacaktır.



Şekil 2.4 İki fazlı step motorun iki fazının enerjilendirilmesi ile rotorun konumu

Şekil 2.4 de yumuşak demirli rotor üzerinde belirtilen N ve S kutupları stator alanından dolayı endüklenmiştir. Stator alanının polaritesi hemen rotor polaritesine karşı yönde olan kutupla bitişik hale gelir.

Bu durumda rotordaki tork sıfırdır. Rotor ve stator arasında bir manyetik kuvvet oluşur ama bu direkt olarak rotor ile aynı hizadadır ve dönen moment veya tork yoktur.

Uygun tork ve rotor kuvveti arasındaki bu ilişkiyi açıklayabilmek için manyetiki kuvveti, rotor ile statoru direkt olarak birleştiren gergin bir yay gibi düşünebiliriz. Yay kuvveti radyal olarak etkilemektedir ve rotoru durgun halinden harekete geçirecek kesin bir dönel kuvvet sağlamamaktadır. Bununla beraber eğer rotor harici bir kuvvetle sapıyorsa ( yay yandan bir kuvvet sarf edecektir.) rotordaki bir tork veya dönen moment orijinal sapma kuvvetine ters yöndedir. Sapmadaki artış depolanan torkda bir artışa neden olacaktır.

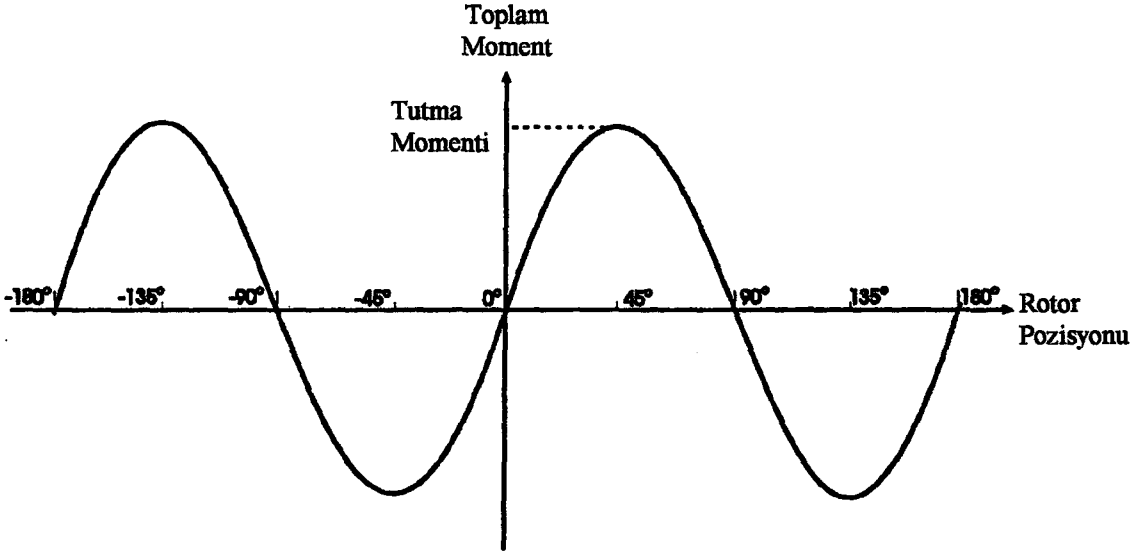
Rotoru etkileyen harici bir kuvvet, rotoru kuvvet ile aynı doğrultuda hareket ettirme eğilimi gösterecektir. Stator ve rotorun arasındaki kuvvetin büyüklüğü hava aralığının artmasından dolayı açılma yer değişim ile azalacaktır. Bununla beraber orijinal pozisyona dönmeyi deneyen depolanmış bir tork ters yönde oluşabilir. Bu tork Relüktans Torqu olarak isimlendirilir ve motorun değişken relüktansından dolayı ortaya çıkmıştır.

Şekil 2.5 iki fazlı bir sistem için rotor pozisyonuna karşı torkların toplamının karakteristiğini göstermektedir.

Kullanılan standart, eğer toplanılan tork rotoru saat yönünde döndürecek şekilde ise pozitif ve saat yönüne ters yön için negatiftir. Gösterilen tork rotoru pozisyonun dışına taşıyan harici kuvvetin torkudur ve bu tork rotorda kendi kendine oluşmaz. Rotor torkuda aynı genlikte olacaktır fakat ters polaritede.

Pozitif rotor sapmasıyla ( saat yönünde ) torkun pozitif olarak maksimum  $45^\circ$  ye kadar çıkıp sonra  $90^\circ$  ile sıfıra düşeceği görülebilir. Dalga şekli sinüzoidal olarak göz önüne alınabilir.

Tam karakteristiğın şekli aşağıdaki gibi açıklanabilir.



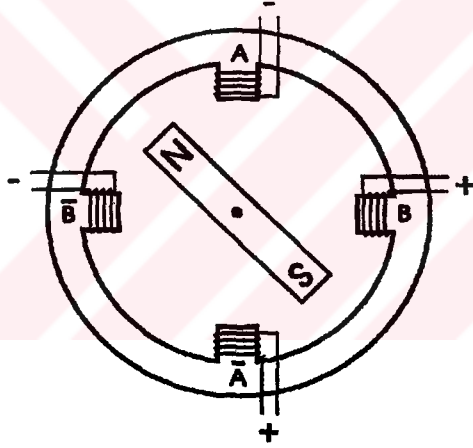
Şekil 2.5 İki fazlı step motorun tutma momentinin, rotor pozisyonuna göre değişimi

1.  $0^\circ$  de, rotor ile stator arasında sadece manyetik kuvvet mevcuttur. Tork yok.
2.  $0^\circ$  ile  $45^\circ$  arasında, rotoru yer değiştirmek için ihtiyaç olunan tork maksimuma doğru artar.
3.  $45^\circ$  de, Artan tork maksimumdur ve Tutma torku olarak isimlendirilir. Bu değerin üzerinde ilave edilen bir yük rotorun pozisyon dışına çıkmasına neden olur.
4.  $45^\circ$  ile  $90^\circ$  arasında, tork sıfıra doğru iner. Rotor ise tork diğer  $90^\circ$  de iptal olana kadar iki dengeli durum arasında eşit olarak etkilenir.
5.  $90^\circ$  ile  $180^\circ$  arasında, rotoru sonraki dengeli pozisyonuna çekilmekten korumak için engellemek gerekir. Bu torkun negatif değeri içindir.
6.  $135^\circ$  de, rotor sonraki durgun pozisyona doğru etkinin maksimum kuvveti ile davranır. Aynı zamanda bu da Tutma Torkudur ve  $45^\circ$  dekine ters yöndedir.
7.  $135^\circ$  ile  $180^\circ$  arasında, tork sıfıra doğru azalır. Rotor ise sonraki dengeli durum ile hizalı hale gelir.
8.  $180^\circ$  de, rotor bir kez daha sonraki dengeli pozisyon içinde stator alanı ile hizalanır. Döndürülmüş rotorda endüklenen mıknatıslanmanın polaritesi ile hiç bir tork üretilmez.

Negatif rotor pozisyonları için karakteristiğın şekli, 0°'den 180°'ye, aynı şekilde açıklanabilir.

Tutma torkunun önemi bunun maksimum tork oluşu ve bu maksimum torkun uyarılmış stator ile sabit rotora uygulaması ve halen daha bu pozisyonun üzerinde kontrol kurulmasıdır. Eğer uygulanan yük tutma torku değerine eşit veya bu değerden büyük bir değer ise rotor yük tarafından sürülebilir veya senkronizasyon kaybolur. Eğer harici yük tutma torkunun değerini azaltsaydı, rotor ilk önce stator alanına doğru etkilenecek ve dengeli pozisyonlardan bir tanesinde tutulu kalacaktı.

Sonra, akımın polaritesi B sargısında aynı yani pozitif iken ama A sargısında akım ters olarak değiştirilmiş yani negatif iken durumu göz önüne alalım. Durum Şekil 2.6 daki gibidir.

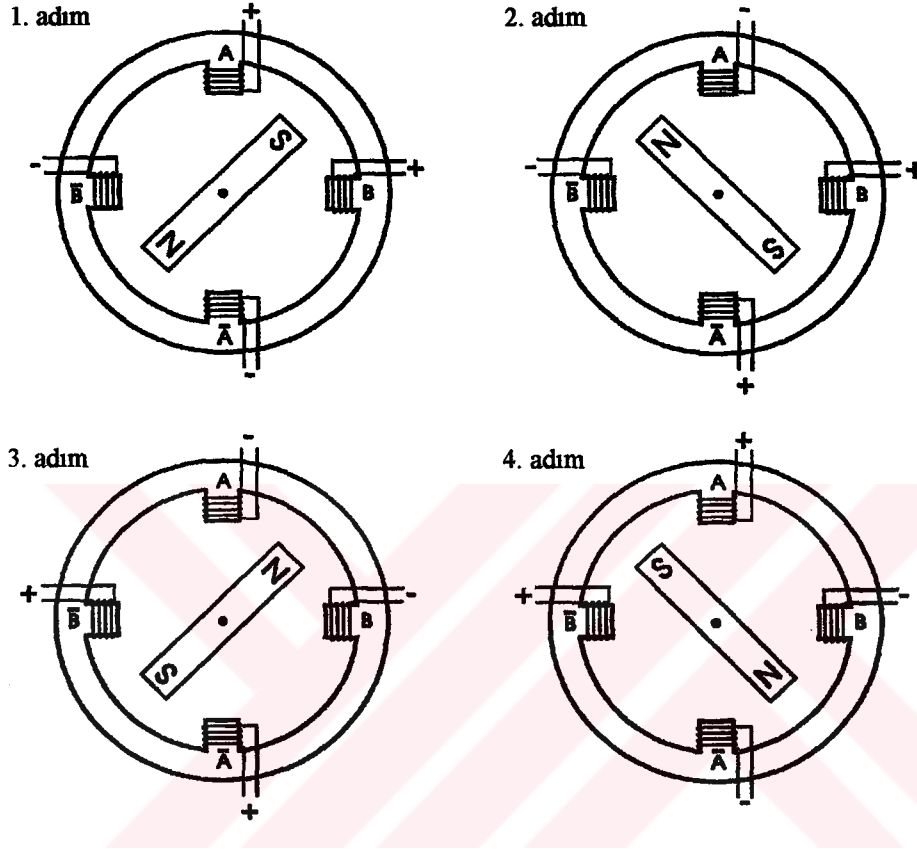


Şekil 2.6 İki fazlı step motorun fazların ters olarak enerjilendirilmesi

Rotor, stator sargı ve akımlarından oluşan sonuç manyetik alan ile hizalmıştır. Bu belirtiler gözlemciye rotorun 90°'lik saat yönünde dönüşünü göstermektedir.

Şekil 2.7 stator sargılarından akıtılan akımın 4 durumu altında rotorun pozisyonunu göstermektedir.

1. ve 2. adımlar önceden tanımlanmış Şekil 2.5 ve 2.6'dır. 3. Ve 4. Adımlar sonraki durumlardır ve A ve B sargılarının bir kez daha değiştirilmesi ile oluşur. Her adımda rotor saat yönünde  $90^\circ$  ile ilerlemektedir.



Şekil 2.7 İki fazlı step motorun fazlarının sıra ile enerjilendirilmesinde statorun alacağı konumlar

Eğer 1. adım ve 4. adımlardan sonra tekrarlınsaydı ( A ve B sargılarından pozitif akım akıtmak ile ) rotor 1. adımda ( Şekil 2.4 ve 2.7 ) gösterilen orijinal pozisyonuna gidecekti. Adımların tam sırası rotorda tamamlanmış tam bir devre (  $360^\circ$  ) sebep olmuştur.

Uyarma akımlarının dönüşlerinin sırasının aynı şekilde tekrarı rotora devam eden saat yönünde bir dönüş kazandıracaktır. Bu dönüş yapılan akım değişiklikleri ve hız akımlarının anahtarlama oranları ile yönetilmektedir.

Tanımlanmış sıralı anahtarlama, stator akımlarının stator sargılarına doğru eşli değiştirildiğinde, TAM ADIM işletme olarak isimlendirilir veya step motor TAM ADIM MOD' unda da çalıştırılıyor denir.

Ayrıca şu da göz önüne alınmalıdır. Bütün zamanlardaki her iki sarıdan da geçen akımlar her iki yön içinde akıtılabilir. Buna ÇİFT FAZ UYARMA adı verilir ve yüksek güç performansı istenilen yerlerde yaygın olan step motor uyarma tipidir.

Çizelge2.1., 1.adımdan 5.adıma kadar akımın anahtarlama sırasını göstermektedir. 5.adım aslında 1.adımın tekrarıdır.

Çizelge 2.1. 1. Adımdan 5. Adıma kadar stator sargılarının anahtarlama sırası

Adım	İleri Sıra	Ters Sıra
1.adım	$\bar{A} B$	$\bar{A} B$
2.adım	$A B$	$\bar{A} \bar{B}$
3.adım	$A \bar{B}$	$A \bar{B}$
4.adım	$\bar{A} \bar{B}$	$A B$
5.(1.)adım	$\bar{A} B$	$\bar{A} B$

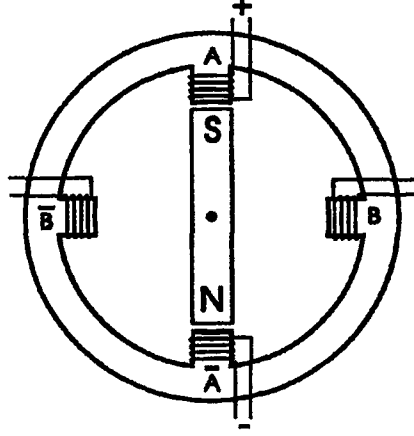
Bundan dolayı, eğer sargılardan geçirilen akımlar tabloda gösterilen ileri sırada ise rotorda aynı yönde dönecektir. İki fazlı stator için rotorun adım açısı her 5 adımda tam bir devri veren  $90^\circ$  olmalıdır. ( Rotor pozisyonunda 4 değişiklik olmaktadır.)

Eğer ters sıra kullanıldı ise ( aslında ileri sıradaki gibi aynı sayı ve akım durumlarına sahiptir ama bunlar ters sıradadır.) rotor ters yönde ama halen aynı adım açısı ve hızı ile hareket edecektir.

Tam adım modunda rotorun her adımda yaptığı açı ADIM AÇISI diye adlandırılır. Minimum adım açısı faz sayısının ve/veya stator kutup sayısının arttırılması ile sağlanabilir.

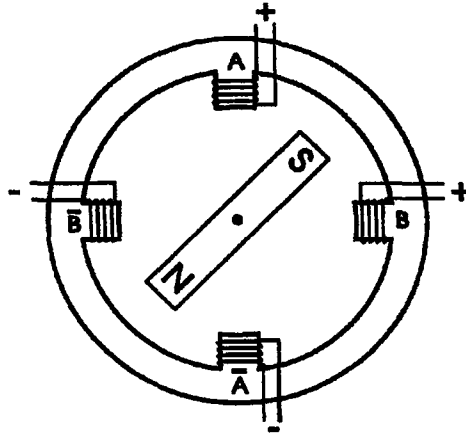
Step motorda pozisyonel çözümün daha da iyileştirilmesi YARIM ADIM MOD' unda çalışma ile sağlanabilir.

İlgilenilen durum Şekil 2.8 de gösterilmiştir.



Şekil 2.8 İki fazlı step motorun yarım adım çalışması durumunda birinci adım

Rotor önceden verilmiş olan aynı prensip ile A sargısından geçen pozitif d.c. akımın oluşturduğu alan ile kendi kendine hizalanmıştır. Bu sırada B sargısından hiç bir akım akıtılmamaktadır.

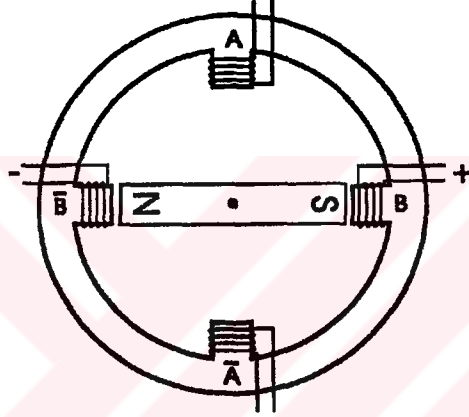


Şekil 2.9 İki fazlı step motorun yarım adım çalışması durumunda ikinci adım

A sargısındaki akım durumu devam ederken B sargısını da pozitif bir akım ile enerjilendirirsek rotor yeni bir manyetik alan ile hiza alacaktır. Bu şekil 2.9 da gösterilmiştir.

Etki rotorun önceki gibi saat yönünde dönüşüne neden olacaktır ama azaltılmış bir  $45^\circ$  lik adım açısı ile. ( Tam adım işletme için  $90^\circ$  lik adım açısına karşılıktır.)

Akım dağılışındaki diğer bir değişiklik, B sargısındaki akım aynı iken A sargısındaki akımı sıfır yapmak, rotorun takip ettiği manyetik alanın biraz daha saat yönünde dönüşü ile yeni bir pozisyon almasına neden olur. Bu Şekil 2.10 da gösterilmiştir.



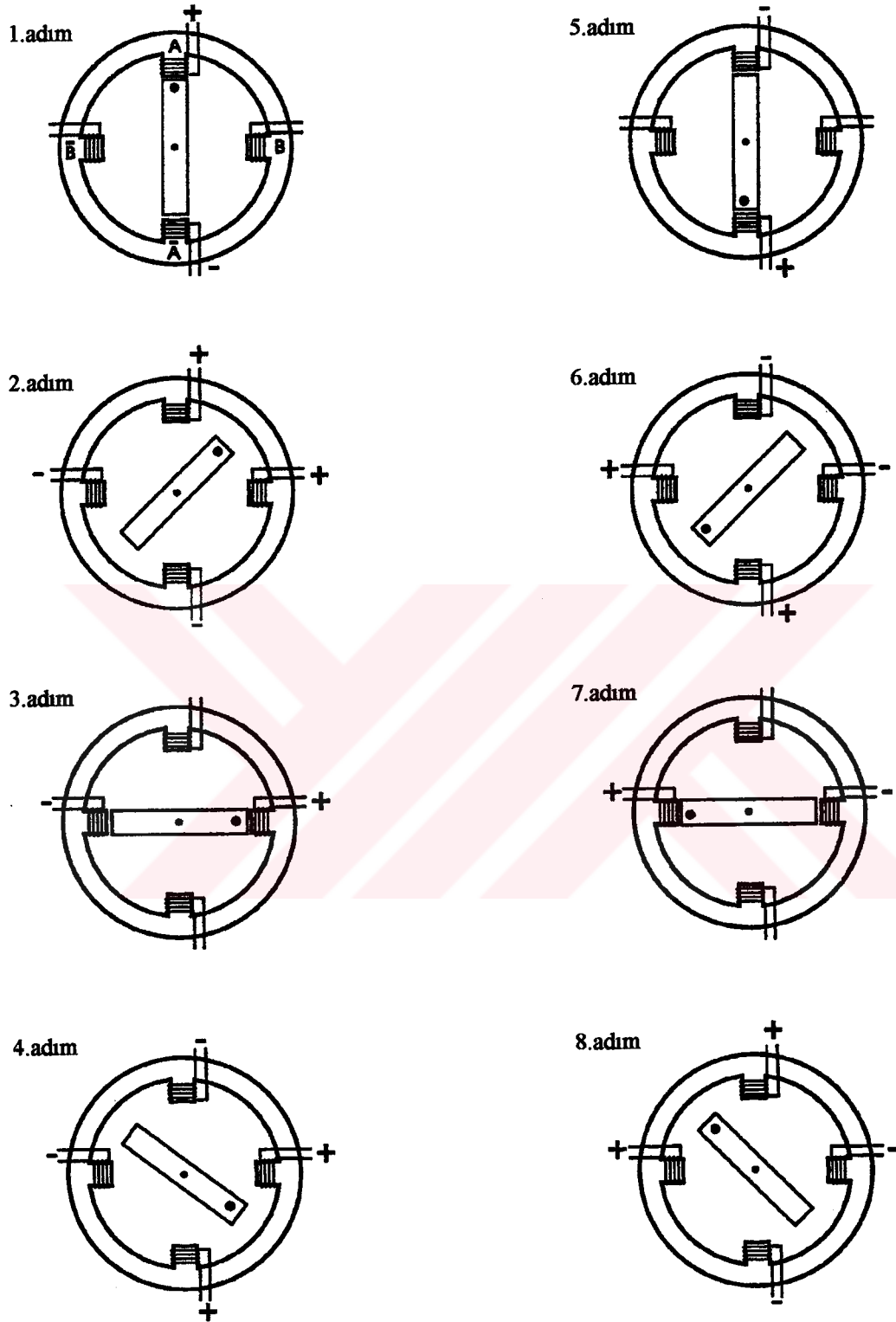
Şekil 2.10 İki fazlı step motorun yarım adım çalışması durumunda üçüncü adım

Yarım adım modda bir kez daha yapılan akım değişiminin etkisi rotorun  $45^\circ$  ile saat yönünde biraz daha dönüşüne neden olacaktır.

Şekil 2.11 muhtemel sekiz adımı, akımların akışını göstererek stator akım sırasında ve onların polaritelerini oluşturulan rotorun pozisyonu ile özetlemiştir. Sıradaki son adımın ( 9.adım) 1.adımın bir tekrarı olduğuna dikkat edilmelidir.

Böylece eğer stator akımının değişim sırası bu şekilde takip edilirse rotor  $45^\circ$  lik adımlar içinde  $360^\circ$  ile dönecektir. Benzeri olarak sırayı değiştirdiğimizde dönüş de ters yönde olacaktır ki bu yön saat yönüne ters olan yöndür.





Şekil 2.11. İki fazlı step motorun yarım adım çalışması durumunda ilk 8 adımda rotor konumları

Ayrıca eğer sıra periyodik olarak tekrarlanırsa rotor 360°'den fazla döndürülmüş olup devir üretilmeye başlanacaktır. Bir hızda sürekli bir dönüş, anahtarlama oranı ve ayrıca rotor ve statorun konstrüksiyonu ve dizaynı ile kontrol edilir.

Çizelge 2.2 yarım adım mod için anahtarlama sıralarını göstermektedir.

Çizelge 2.2. Yarım adım modda çalışma için anahtarlama sıraları

Adım	İleri Sıra	Ters Sıra
1.adım	$\bar{A} B$	$\bar{A} B$
2.adım	$- B$	$\bar{A} -$
3.adım	$A B$	$\bar{A} \bar{B}$
4.adım	$A -$	$- \bar{B}$
5.adım	$A \bar{B}$	$A \bar{B}$
6.adım	$- \bar{B}$	$A -$
7.adım	$\bar{A} \bar{B}$	$A B$
8.adım	$\bar{A} -$	$- B$
9.(1.)adım	$\bar{A} B$	$\bar{A} \bar{B}$

Tam adım modda olduğu gibi tablodaki son adım ilk adımla tamamiyle aynıdır ve sıranın değişimi rotorun dönüş yönünü değiştirecektir.

Çalıştırma bakımından adımların değişimi sırasında motor yarım adım modda çalıştırılırken sürenin yarısı için sargıların sadece birinden akım akıtılmaktadır. Bu ise TEK FAZLI UYARMA ile isimlendirilir ve bu step motoru besleyen elektriksel gücü sınırlar ve bundan elde edilen mekanik gücü azaltır. Bundan dolayı çalışma modunun seçimi ( yarım veya tam adım ) çıkış güç seviyesindeki azalma ile geliştirilmiş pozisyon seçimi için ihtiyacı dengelemeyi gerektirir.

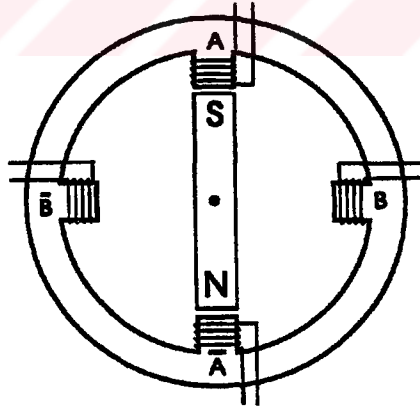
Değişken relüktanslı step motor küçük step açıları sağlayabilmek açısından uygundur genellikle sınır  $1^\circ$ 'den  $60^\circ$ 'ye kadardır. Onlar ayrıca düşük ataletle sahiptirler ve geçici durumlara çabuk cevap verirler. Yüksek adım oranlarında da çalışmaya elverişlidirler. Dezavantajları ise senkronlama torkunun olmayışı ve doğal sönümü olmayışı bundan dolayı da performansının osilasyon ve rezonansdan etkilenişidir.

### 2.2.2. Sabit mıknatıslı step motor

Sabit mıknatıslı step motor rotoruna yerleştirilmiş sabit bir mıknatısa sahiptir. Rotor torku iki etkinin kombinasyonudur.

- i. Manyetik kuvvet ve tork stator sargılarının uyarılmasından dolayıdır. Bu işlemler önceden anlatılmış olan değişken relüktanslı step motor ile aynıdır. Tek fark rotor sabit olarak mıknatıslanmıştır.
- ii. Stator ile rotor arasındaki manyetik kuvvetler /torklar statorun çıkıklığı ve rotordaki sabit mıknatıstan dolayıdır.

Şekil 2.12 de gösterilen iki fazlı sabit mıknatıslı step motora göz atalım.



Şekil 2.12 İki fazlı sabit mıknatıslı step motor

Rotorun N ve S kutupları rotorun yapısında varolduğundan dolayı hazırırlar.

Sargıların hiç birinden uyarılma akımı akıtılmadığı halde rotor en yakın stator kutbu ile hizalı hale gelecektir ( Burada A-A'). Rotorun herhangi kutbundan bir tanesi ile hizalı hale

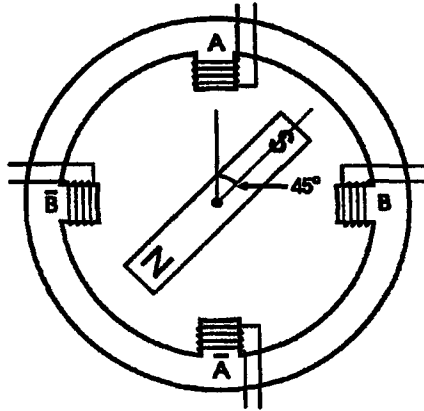
gelebileceği de bilinmelidir ve bu da ilk çevirmedeki veya bir önceki kullanımdan sonraki, rotorun serbest halde stator kutbuna en yakın olduğu yerdir. Hatta bu hizalama(kilitleme) yataklardaki sürtünme gibi içsel yükler veya herhangi bir harici yükün üstesinden gelmede uygun kuvvetlerdir.

Değişken relüktanslı step motor için verilen aynı sebepler için ( etkinin manyetik kuvveti sadece stator ve rotor arasında mevcuttur ) gösterilen pozisyonda rotorda hiç bir moment oluşmaz.

Eğer şu anda rotoru saat yönünde döndürmek amacıyla dışardan bir kuvvet rotora uygulanırsa, moment ortaya çıkar ve açısal yer değişim ile maksimuma çıkar. Bu ise Relüktans Momentidir.

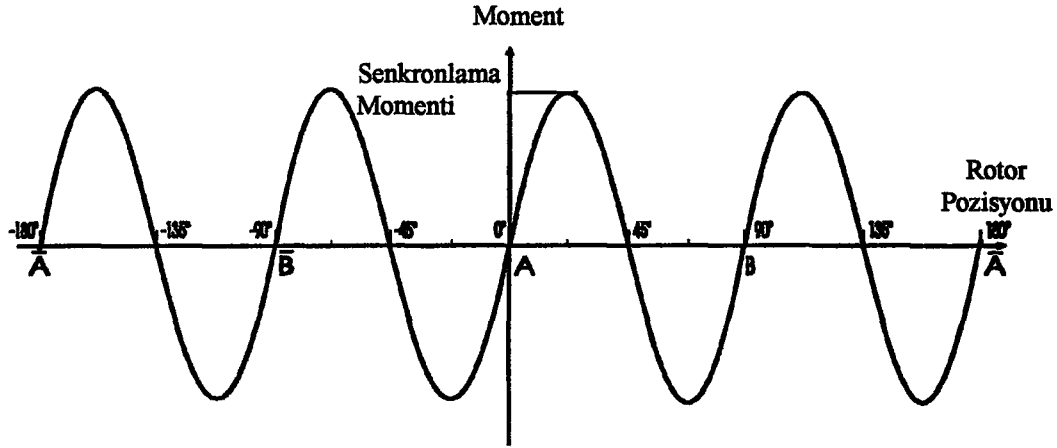
0 ile  $45^\circ$  arasında, üretilen moment sinüsoidal olarak  $22,5^\circ$  'de maximuma çıkar ve sonra 0'a iner.

Şekil 2.13'de gösterilen  $45^\circ$  lik yer değişimde rotor tam olarak iki stator kutbunun yarı yolundadır ( ortasındadır.) İki kutba da olan etki eşit ama terstir ve oluşan net rotor momentini iptal eder.



Şekil 2.13 İki fazlı sabit mıknatıslı step motorda rotorun aynı iki stator kutbu arasındaki konumu 45 ile  $90^\circ$  arasında, rotor B kutbu ile hizalı halde kalabilmek için bir direnç göstermeyi talep eder. Böylece rotor tarafında üretilen moment açısal yer değişiminin bu sınırlarında negatiftir.

Şekil 2.14 rotorun  $\pm 180^\circ$ 'sini yani tam olarak  $360^\circ$  için rotora ters olarak uygulanmış moment karakteristiğini göstermektedir .



Şekil 2.14 İki fazlı sabit mıknatıslı step motorda senkronlama momentinin, rotor pozisyonuna göre değişimi

Uyarmsız durumdaki pozisyonda bir değişikliğe neden olmayan rotora uygulanmış maksimum momente senkronlama momenti (Bir mekanizmanın parçasını kilitleyen moment) adı verilir. Bu moment step motorun uygulamalarında önemli bir yer alır.

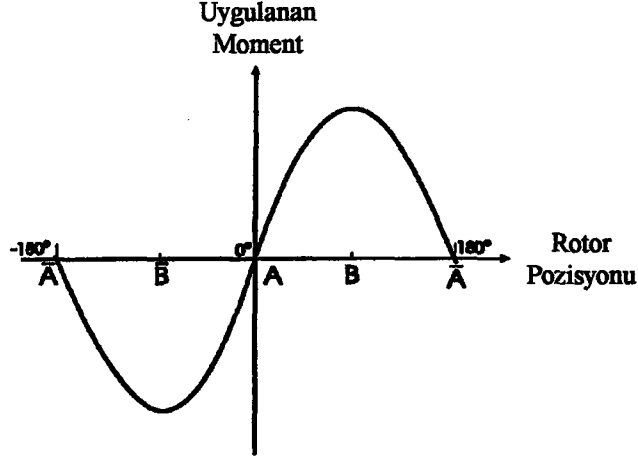
Eğer bir d.c. akım A sargısından şu anda akarsa, rotor oluşan manyetik alanla hizalanır. Bu ise Şekil 2.12 ' de gösterilen rotor pozisyonu ile aynıdır. Bu pozisyonda rotordaki N kutbu statordaki S kutbu ile karşılıklı olarak hizalanır.

Bu pozisyonda rotordan oluşan moment sıfırdır ve zaten aynı sebepler için daha önceden incelenmişti.

Eğer önceki gibi harici bir yük uygulanırsa, rotor kuvvet ile aynı yönde sapacaktır ve ters moment oluşur. Bu ise 'Uyarma Momenti' adını alır.

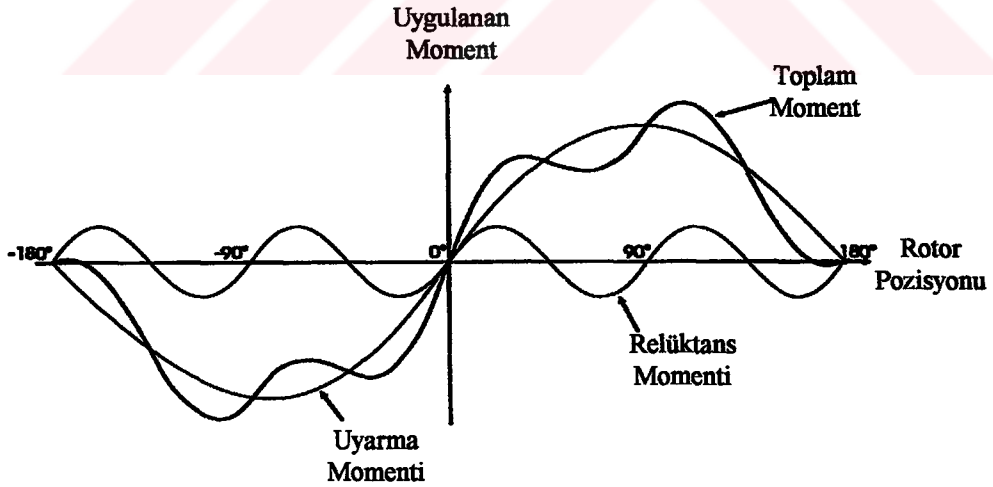
Rotor pozisyonuna karşı uygulanan momentin karakteristiği Şekil 2.15 'de verilmiştir .

$0^\circ$  ile  $90^\circ$  arasında , moment maximuma doğru pozitif olarak artar ve sonra  $180^\circ$ 'de sıfıra geri döner .



Şekil 2.15 İki fazlı sabit mıknatıslı step motorda uygulanan momentin, rotor pozisyonu göre değişimi

$180^\circ$ 'de ( A ) rotor momenti 0 olmasına rağmen bu oldukça kararsız bir durumdur ( benzeri olarak manyetik kutupların itiş ) ve uygulanmış kuvvet alındığında rotor hareket için zorlanmıştır. Şimdiye kadar relüktans momentini ve uyarma momentini ayrı ayrı gözden geçirildi. Pratikte her etkininde katkıda bulunduğu, rotorun oluşturduğu moment tek bir toplam momenttir.



Şekil 2.16 İki fazlı sabit mıknatıslı step motorda momentlerin, rotor pozisyonuna göre değişimleri

Şekil 2.16 sabit mıknatıslı step motorun birleştirilmiş relüktans ve uyarma momenti karakteristiklerini göstermektedir .

Çalışma bakımından sabit mıknatıslı step motor değişken relüktanslıdaki gibi stator uyarmasının değişimi ile cevap verir. Bu ise stator ve rotor arasında kurulan çekici ve itici manyetik kuvvetlerin kullanımı ile rotor kendisi hizalanır yani stator akımının değişimi ile ortaya çıkan stator alan pozisyonu değişimi ile Şekil 2.7 ve Şekil 2.11'e bakıldığında tam adım ve yarım adım modlarında her biri ayrı ayrı rotor pozisyonlarının tam sırası görülmektedir.

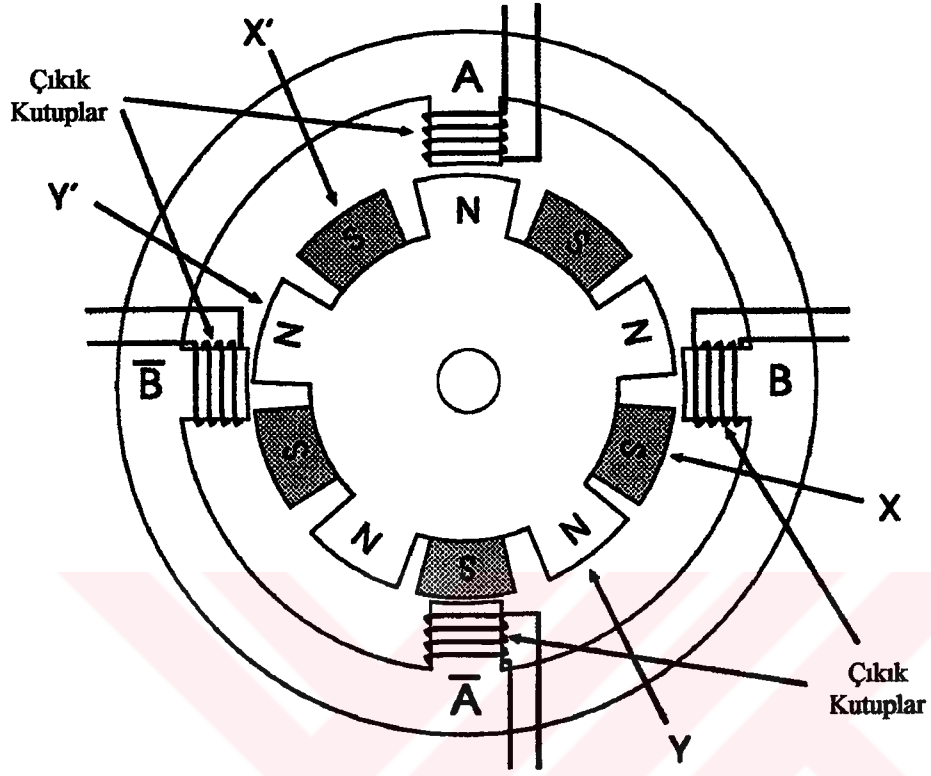
Sabit mıknatıslı step motorlar genellikle büyük adım açıları ile uygundur ( tipik olarak  $90^\circ$  ve  $120^\circ$  ) ve bu birçok uygulamalar için pozisyonel doğruluğu ciddi bir şekilde sınırlamaktadır. Onlar senkronlama momentine, iyi sönüm karakteristiğine ve eşdeğer bir değişken relüktanslı step motordan daha yüksek momente sahiptirler.

### 2.2.3. Hibrid Step Motor

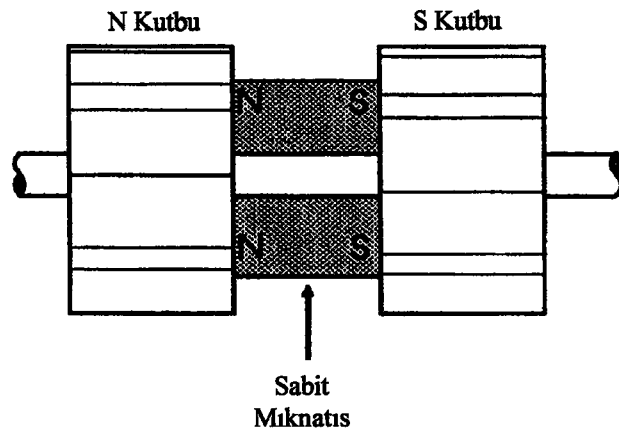
Hibrid step motorun rotoru daha önceden açıklanmış olan sabit mıknatıslı ile değişken relüktanslı tiplerin karakteristiklerinin birleşimi için yerleştirilmiş sabit bir mıknatıs ile şekillendirilmiş bir manyetik yola sahiptir.

Şekil 2.17 tipik bir iki fazlı Hibrid step motorun stator ve rotorunu göstermektedir.

Makinenin rotoru iki tane şekillendirilmiş yumuşak demirli kutup parçacıklarından oluşmaktadır. Her bir parçacık 5 dişli olup yarım dişlik bir derece ile kaydırılmıştır. Bir sabit kutupta bu parçacıklar arasına sıkıştırılmıştır. Bu Şekil 2.18 'de görülmektedir .



Şekil 2.17 İki fazlı hibrid step motor stator ve rotoru



Şekil 2.18 İki fazlı hibrid step motorun rotor kesiti



Her iki sargıdan da hiç bir akım akmadığında, rotor ve stator arasındaki paralel manyetik yollar yüzünden senkronlama momenti rotoru bir pozisyonda tutar.

Tam adım modunda A ve B sargılarından geçirilen pozitif d.c. akımlar sonuç olarak meydana gelen bir manyetik alan oluştururlar. Bu alan ise Şekil 2.17’de gösterilen pozisyonda rotoru tutan kuvvetleri güçlendirir.

Eğer B sargısındaki akım değiştirilmediği halde A sargısından geçen akım ters çevrilirse, stator alanı  $90^\circ$  ile saat yönünde ilerler. N ve S kutuplu rotor dişleri B ve A<sup>-</sup> (Şekil 2.17 ‘de X ve Y ) arasındaki yarı yolda yeni bir N kutbu pozisyonuna kurulur. S kutbu çekilirken N kutbu ( Y ) alan tarafından itilir. Benzeri olarak rotorun diğer tarafında olan X<sup>1</sup> ve Y<sup>1</sup> rotor dişleri itilir ve her biri ayrı ayrı yeni alana çekilirler.

Birleşmiş etki rotorun saat yönünde dişin derecesinin  $\frac{1}{4}$  ‘ ü ile dönmesine neden olur. Yani  $72^\circ/4$  veya  $18^\circ$ .

Tam adım modda iken stator uyarılmasında yapılan başarılı değişiklikler rotorun  $18^\circ$  ’lik adımlarla dönmesine neden olacaktır. Yarım adı çalışmada adım açısı  $9^\circ$  olacaktır.

Bilinen bir step motoru için adım açısı stator ve rotor dişlerinin derecelerinin farkından hesaplanabilir. Şekil 2.7 ve Şekil 2.8’ de gösterilen stator ve rotor konfigürasyonuna sahip bir step motor için,

$$\text{Stator dişi derecesi} = 360^\circ / \text{Diş sayısı} = 360^\circ / 4 = 90^\circ$$

$$\text{Rotor dişi derecesi} = 360^\circ / \text{Diş sayısı} = 360^\circ / 5 = 72^\circ$$

$$\text{Adım açısı} = \text{Diş derecelerindeki fark} = 90^\circ - 72^\circ = 18^\circ$$

Böylece 4 stator kutbu ve 5 rotor kutbuna sahip bir step motorun minimum rotor pozisyonu  $18^\circ$ ’dir. Bu ayrıca rotorun tam adım modda her 20 darbeye ve yarım adım mod için 40 darbeye tam bir devre sahip olacağı anlamına gelmektedir.

Herhangi bir bilinen rotor pozisyonunda rotordaki moment, bir çift kutuptan daha fazla kutupların oluşturduğu hem uyarma ve hem de relüktans etkilerinin bir sonucu olan momenttir.

Step motorun pozisyon ihtiyacının arttırılması için kutupların sayısının arttırılması gerektiği açıktır. Açısal yönden daha fazla gelişme, pratikte makinalarda rotor üzerindeki dişli sayılarının artışı ile mümkün kılınabilmektedir.

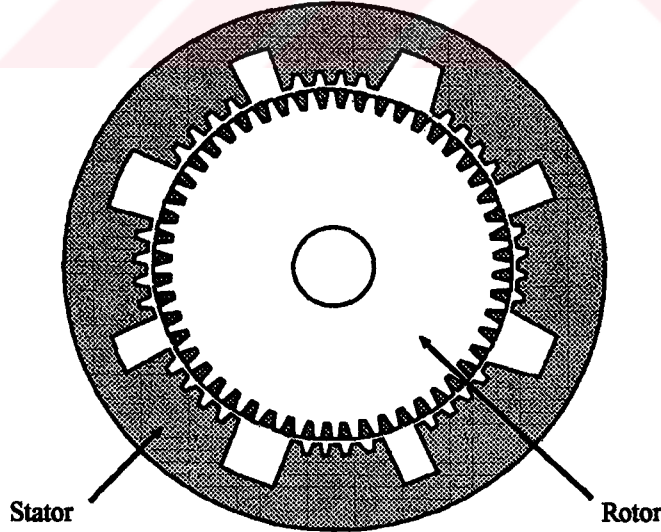
Şekil 2.19 rotor yüzeyleri bir çok dişe sahip olan endüstriyel bir hibrid step motor tipinin tipik bir stator/rotor düzenlemesini göstermektedir. Stator kutbu yüzeyleri de ayrıca eşit olmayan diş sayısına sahip olacak şekilde düzenlenmiştir.

50 stator dişi ve 40 rotor dişi ile basit adım açısı,

$$\text{Stator diş derecesi} = 360^\circ / 50 = 7.2^\circ$$

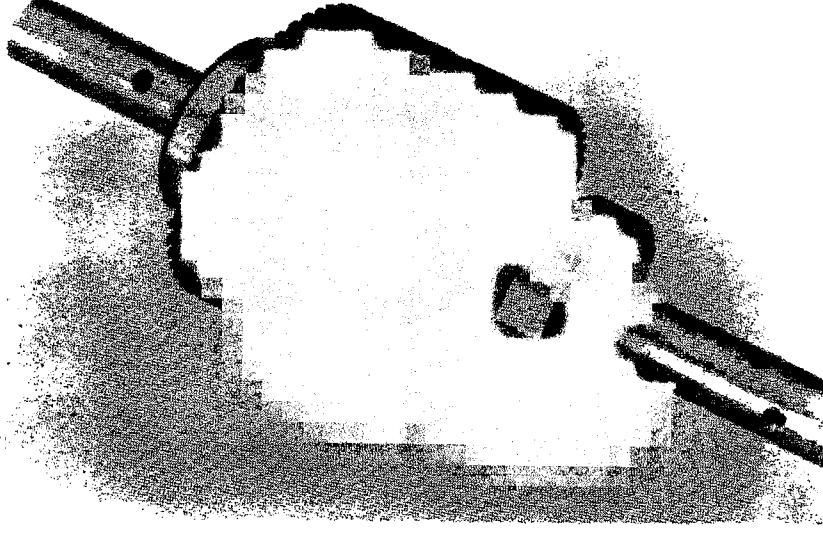
$$\text{Rotor diş derecesi} = 360^\circ / 40 = 9^\circ$$

$$\text{Adım açısı} = 9^\circ - 7.2^\circ = 1.8^\circ$$

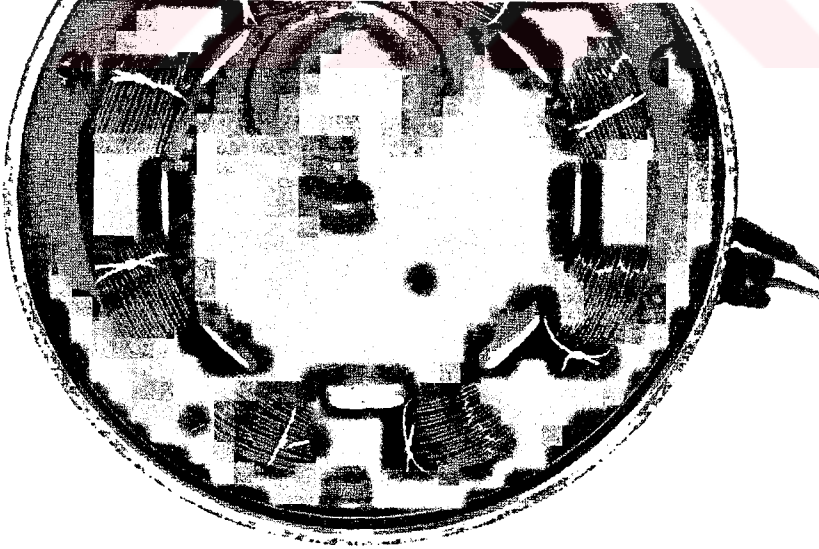


Şekil 2.19 Statoru ve rotoru çok dişli hibrid step motor

Bu step motorun adım açısı  $1.8^\circ$  ve  $360^\circ$ 'lik tam bir devri sağlayabilmek için 200 adıma tam adım modda veya 400 adıma yarım adım modda gereksinim duymaktadır.



Şekil 2.20. Hibrid tip step motor rotoru



Şekil 2.21. Hibrid tip step motor statoru

Genel olarak hibrid step motorları yaklaşık  $5^{\circ}$ 'ye kadar küçük adım açıları sağlamaktadır. Bu yüzden onlar çok yüksek açısal kararlılığa ve senkronlama momentinin de sağlanması ile ilave bir avantaja sahiptirler. Benzer boyutlardaki değişken relüktanslı veya sabit mıknatıslı step motorlarla karşılaştırıldığında, hibrid step motor daha büyük senkronlama, tutma ve dinamik momente sahiptir. Ayrıca hızlı adımlama oranı ile çalıştırılma kabiliyetine de sahiptir.



### 3. STEP MOTORLARA AİT ÖNEMLİ BÜYÜKLÜKLER

Step motorlarını incelemeye başlamadan önce bu motorlara ait bazı davranış karakteristiklerini ve terminolojiyi incelemek yararlı olacaktır. Buna; step motorlara ait büyüklüklerin ve bunların ifadesinde kullanılan terimlerin klasik motorlara ait büyüklüklerden ve terimlerden farklı olması nedeniyle gerek duyulmuştur.

#### 3.1. Çözünürlük

Çözünürlük, bir devirdeki adım sayısı veya dönen motorlar için adım açısı(derece), lineer motorlar için ise adım uzunluğu olarak tanımlanır. Sabit üretim sırasında tespit edilen bir büyüklüktür.

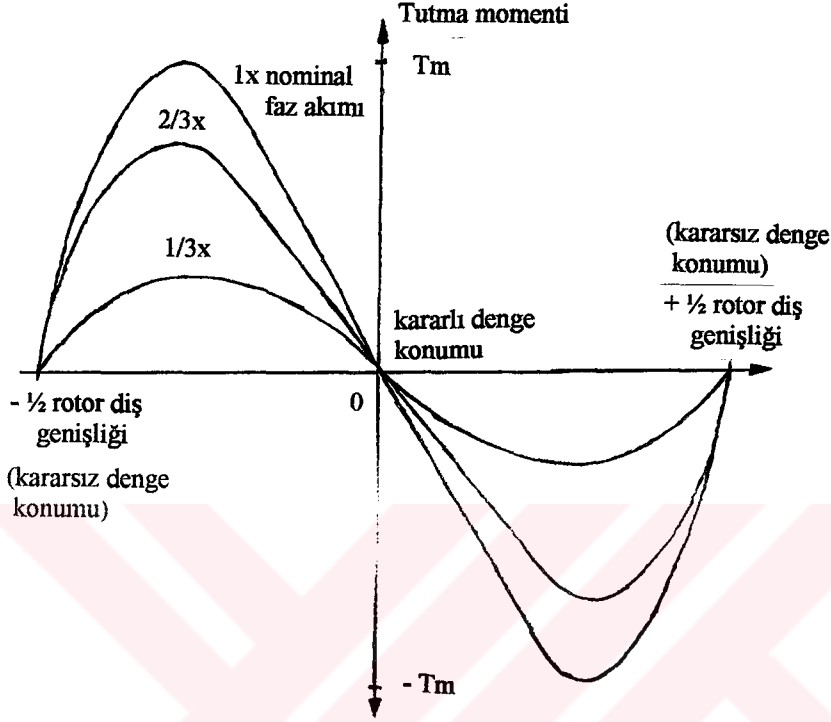
Bir step motorun adım büyüklüğü, çeşitli kontrol düzenleri ile değiştirilebilir. Yarım adım çalışmada adım büyüklüğü nominal değerinin(çözünürlüğünün) yarısına indirilir. Bu yöntem sırasıyla tek faz, iki faz, tek faz uyarımlarla uygulanabilir. Ayrıca motorun sahip olduğu faz sayısına göre yarım adım çalışma için değişik uyarma şekilleri saptanır. Bir diğer yöntem ise bir faz uyarılmış durumdayken diğer fazlardan akıtılacak akımların seviyelerinin kontrol edilmesiyle adım büyüklüğünün çeşitli değerlerde tutulmasıdır.

#### 3.2. Doğruluk

Bir step motorun adım konumları, tasarım ve üretim sırasında bir araya getirilen bir çok parçanın boyutlarıyla belirlenir. Bu parçaların boyutlarındaki toleranslar ve dahili sürtünmeler adımların nominal denge konumlarında da toleranslara neden olurlar. Bu adım motoru doğruluğu isimlendirilir ve belli bir konumdaki maksimum açısal hatanın nominal tek adım değerinin yüzdesi olarak ifade edilmiş halidir. Örneğin, 24 adımlık bir step motorda maksimum  $\pm 0.25$  derecelik hata,  $\pm \%1.67$  'lik adım açısı hatasını karşılık düşer. Bu hata her yeni adımla artmaz. Bütün yük koşullarında oluşur ve sistematiktir. Klasik step motorlarda  $\pm \%1$  ile  $\pm \%5$  arasında değişmektedir. Sürtünme momenti veya kuvveti nedeniyle oluşan konum hataları bu doğrulukla ilgisi olmayan, daha az veya çok olabilen rasgele hatalardır. Ancak her iki tip hata toplanarak sistemin hatası elde edilir.

### 3.3 Tutma Momenti

Tutma momenti bir motorun en temel karakteristiğidir. Tutma momenti eğrisi motorun ürettiği tutma momentinin rotor konumuna göre değişimini veren eğridir. Şekil3.1'de örnek bir tutma momenti eğrisi görülmektedir.



Şekil 3.1 Faz akımının çeşitli değerleri için tutma momenti/rotor konumu eğrileri

Eğrinin orijini, motorun bir fazını uyarılmış olduğu durumda rotorun kararlı adım konumuna karşılık düşer. Bu eğri rotor adım pozisyonundan uzaklaştırılırsa motorda indüklenenecek; rotoru, sıfır momentli adım pozisyonuna geri getirmeye çalışan momentin(tutma momenti) yönünü ve miktarını verir.

Tutma momenti eğrisi, motorun tüm rotor pozisyonları ve statik uyarma koşullarındaki ani momentini tam olarak tanımlamak için gereklidir. Tüm diğer moment karakteristikleri (statik veya dinamik) bu tutma momenti eğrisi baz alınarak elde edilebilirler.

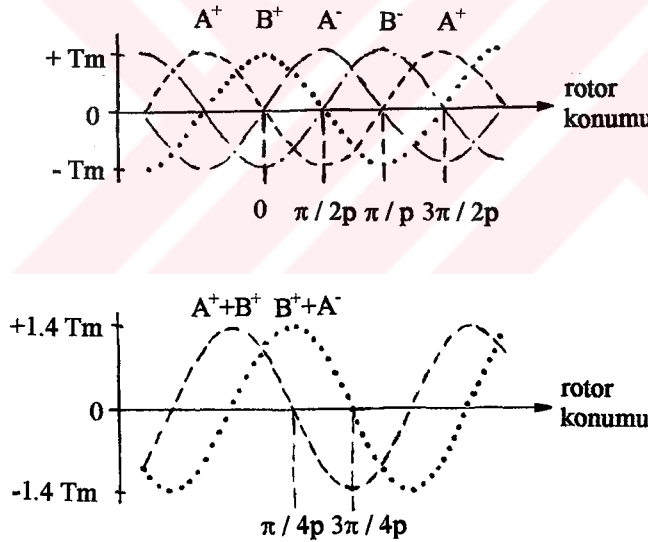
Statik moment/rotor pozisyonu karakteristiği, bir rotor diş genişliği periyoduyla tekrarlanır. Rotorun bir diş genişliği kadar yer değiştirmesi demek, stator ve rotor dişlerinin tekrar karşı karşıya gelmesi demektir. Yarım diş genişliği dönme konumunda rotor dişleri, stator

dişlerinin tam ortasında kalacağından kararsız denge konumu oluşur. Bu nedenle rotor ancak yarım diş genişliğinden az yer değiştirmiş ise gerçek adım pozisyonuna geri dönebilir. Daha büyük yer değiştirmelerde rotor ve stator dişleri istenen adım pozisyonundan bir veya bir kaç rotor diş genişliği uzaklıktaki kararlı denge konumunda karşılıklı gelebilirler. Tutma momenti eğrisinin genliği uyarma akımıyla yakın ilgilidir.

Bir motorun tüm fazlarına ait tutma momenti eğrileri birbirlerine göre faz farkları olan eğriler şeklinde aynı akım ekseninde gösterilirse, belli bir anda motor momentini arttırmak için iki veya daha fazın birlikte uyarılması gerektiği görülecektir. Şekil3.1.'de iki fazlı hibrid motorun tutma momenti eğrileri görülmektedir. Fazların pozitif ve negatif uyarılma durumları için tutma momenti ifadeleri;

$$T_A^+ = -T_m \sin(p \theta) \quad (3.1.)$$

$$T_A^- = -T_m \sin(p \theta - \pi) \quad (3.2.)$$



Şekil 3.2. Hibrid motorun tutma momenti/rotor konumu eğrileri

(a) Tek fazlı uyarım durumu

(b) İki fazlı uyarım durumu

$$T_B^+ = -T_m \sin(p \theta - \pi/2) \quad (3.3.)$$

$$T_B^- = -T_m \sin(p \theta - 3\pi/2) \quad (3.4.)$$

iki faz uyarılma durumunda ise,

$$T_A^+ T_B^+ = T_A^+ + T_B^+ = T_B^+ = -2T_m \sin(p\theta - \pi/4) \cos(\pi/4) \\ -1.4 T_m \sin(p\theta - \pi/4) \quad (3.5.)$$

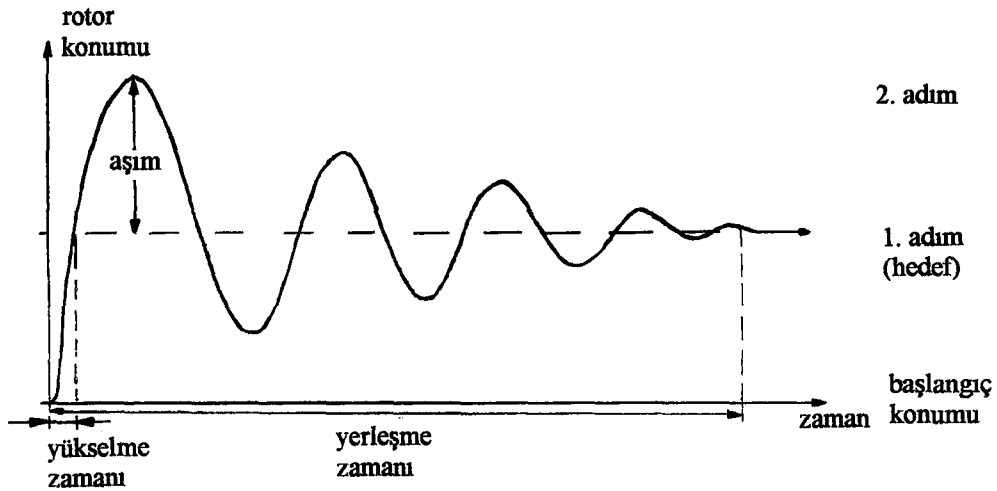
şeklinde elde edilir.

Görüldüğü gibi eğrinin genliği 1.4 katına çıkmıştır. Bu nedenle yüksek moment gerektiren uygulamalarda iki faz uyarma diye tanınan uyarma şekli kullanılarak motor çalıştırılır.

Tez çalışması, iki adet hibrid adım motoru üzerinde yapıldığından örnek olarak bu adım motoru alınmıştır. Daha çok sayıda faza sahip adım motorları için tutma momenti karakteristiğine göre daha çok sayıda faz aynı anda uyarılarak moment artırılabilir.

#### 3.4. Tek Adım Cevabı;

Motor fazlarından biri uyarılmış durumda ise, motor kararlı adım durumundadır. Bu fazın uyarılması kaldırılıp bir diğer faz uyarılırsa motor bir adım atacaktır. Rotor konumunun zamana göre bu değişimi tek adım cevabı olarak tanımlanır. Tek adım cevabı, motorun adım hareketinin hızını, cevabını aşım ve salınım miktarını, adım açısının hassaslığını veren önemli bir karakteristiktir. Şekil 3.3'de tipik bir tek adım cevabı görülmektedir.



Şekil 3.3. Step motorun tek adım cevabı



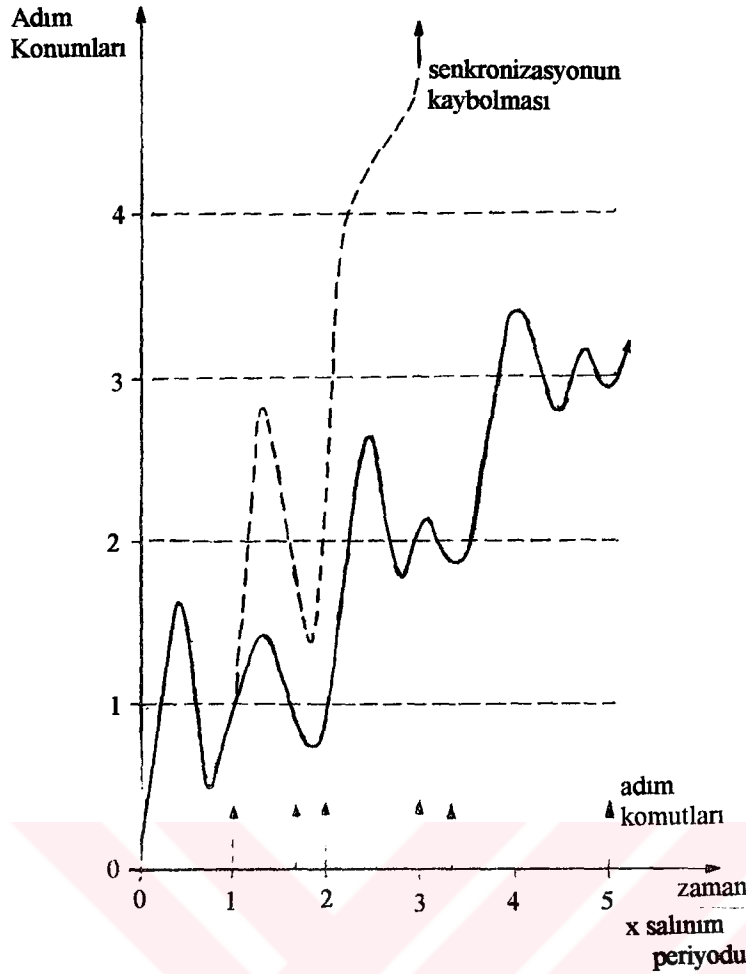
Tutma momenti eğrisi bu cevaptaki aşım ve osilasyonun sebebini açıklamaktadır. Yeni bir faz uyarıldığında, rotor ve stator dişleri arasında oluşan moment, rotoru yeni karalı duruma doğru çekecektir.

Bu moment rotor adım konumuna gelene kadar devam edecek rotor bu konumu geçip aşım yapınca da ters yönde bir moment olarak rotora etki edecektir. Rotor, adım konumundan tekrar geriye geçince bu moment tekrar rotora etki edecek ve bu hareket rotorun kinetik enerjisinin tümü kaybolana kadar devam edecektir.

Bu cevaba ait önemli bir büyüklük olan yükselme zamanı, motorun istenen adım konumuna ilk ulaştığı andır. Rotor bu noktaya ulaştığında hızı maksimum değerinde olacağından, sistem hedefi aşacaktır. Bu ilk aşımın genliği, adım büyüklüğünün yüzdesi olarak ifade edilir ve yüzde aşım olarak tanımlanır. Step motorlarda bu değer genellikle %80-90 mertebelerine çıkar. Yerleşme süresi ise sistem osilasyonunun hedefinin  $\pm\%5$  'ine ilk girdiği an olarak tanımlanır ve bir kaç yüz milisaniye sürebilir.

Hız gerektiren uygulamalarda büyük aşım ve salınımlı cevap bir dezavantajdır. Örneğin, bir yazıcıda her yeni karakterin basımından önce motorun istenen konuma yerleşmesini beklemek gerekeceğinden bu sistemin yavaşlamasına sebep olacaktır.

Tek adımın önemli olmadığı, arka arkaya bir kaç adımın atılmasının gerektiği, hızlı sistemlerde ise motorun ikinci adım konumuna oturmasını beklemeden ikinci adım atılabilir. Şekil 3.4.'den de anlaşılacağı üzere adımlama hızı, denge konumu civarındaki rotor osilasyonunun doğal frekansına eşit ise, uyarma darbesi-adım hareketi senkronizasyonu bozulacak ve rotor istenenden bir kaç adım daha fazla atacaktır.



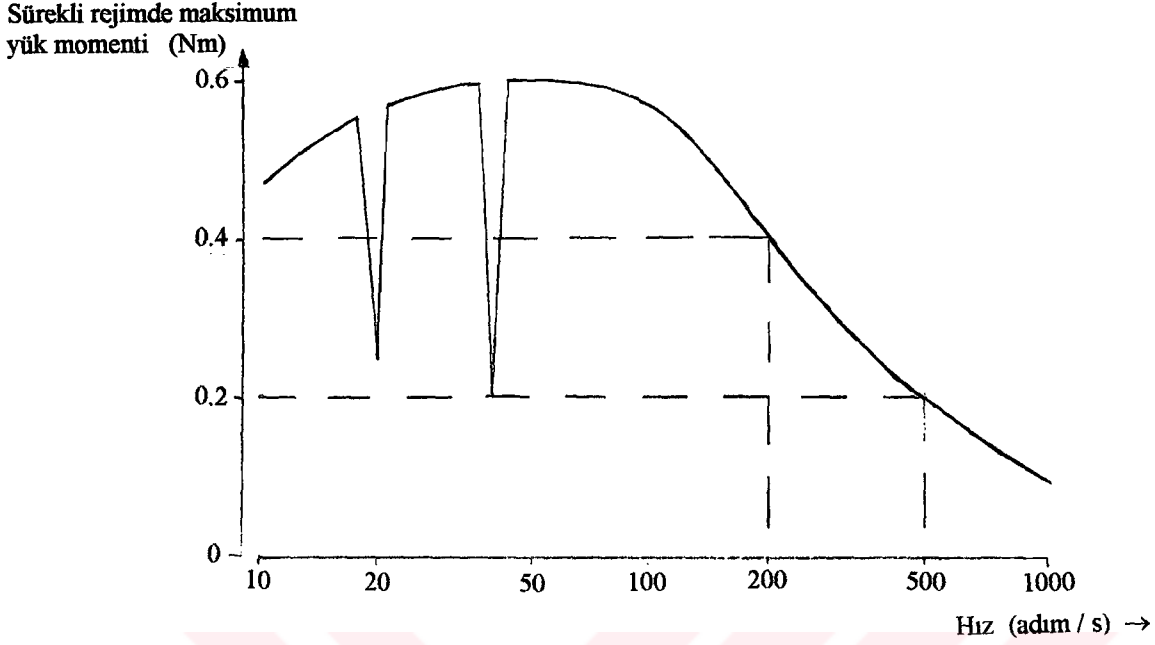
Şekil 3.4. Doğal frekansa yakın hızlarda step motorunun davranışı

Bu davranışın nedeni, ikinci adım darbesi geldiğinde rotor hızının maksimum değerinde olması ve salınım genliğinin yarım diş genişliğinden büyük olmasıdır. Bu durumun rotor davranışına etkisi, sürekli rejimde maksimum yük momentinin incelenmesi sırasında ele alınacaktır.

Görülüyor ki adım motorundan maksimum performans alabilmek için tek adım cevabındaki aşım ve salınımların azaltılması, yerleşme süresinin kısaltılması gerekmektedir. Bu nedenle tek adım cevabının iyileştirilmesi, adım motorlarının kontrolünde çok büyük bir öneme sahiptir.

### 3.5. Sürekli Rejimde Maksimum Yük Momenti Eğrisi;

Bir step motor, mekanik yükün pozisyonunu bir kaç adımlık bir hareketin sonunda değiştirebilmek için kullanılacaksa, motorun hızlanma, yavaşlama veya sabit hızda dönme durumlarında ne kadar moment indükleyeceğinin bilinmesi gerekir.



Şekil 3.5. Sürekli rejimde maksimum yük momenti/hız eğrisi

Sürekli rejimde maksimum yük momenti/hız eğrisi, herhangi bir sabit dönüş hızında, rotor hareketinin giriş darbe dizisiyle olan senkronizasyonunu bozmadan ve rotorun durmasına neden olmadan sürekli halde motor miline uygulanabilecek maksimum yük momentini verir. Bu moment aynı zamanda, söz konusu hızda motorda indüklenebilecek maksimum moment anlamına da gelir. Bu eğri oluşturulurken, eğrinin her bir noktası için, motor ilgili hıza ulaşana kadar boşta hızlandırılır. Daha sonra motora uygulanan yük momenti giderek artırılarak, motorun durmasına yada giriş darbe dizisiyle olan senkronizasyonun bozulmasına neden olmayan en yüksek yük momentini saptanır. Bu momente, o hız için sürekli rejimde maksimum hız momenti denir. Belli bir yük momentini için, motorun durmasına yada giriş darbe dizisiyle olan senkronizasyonun bozulmasına neden olmadan çalışabilecek en yüksek hız da, maksimum sürekli rejim hızı olarak isimlendirilir.

Klasik motorlarda bu eğriye karşılık gelebilecek bir karakteristik yoktur. Maksimum yük momenti eğrisi, çalışma noktalarını göstermez, bir transfer fonksiyonu eğrisi de değildir. Sadece çalışma bölgesini sınırlar. Bu eğrinin sınırladığı bölge içinde herhangi bir noktada

motor giriş darbelerini kaybetmeden ve durma tehlikesi olmadan ilgili hız ve momenti ile çalışır. Sınırlar dışına çıkıldığında motorun davranışı için aynı şey söylenemez.

100 adım/saniyeye kadar olan düşük hızlarda faz akımı sargılarda nominal değerine kadar kolaylıkla yükselebildiğinden, indüklenen moment de büyük olmaktadır. Bu durumda maksimum moment, tutma momenti eğrilerinden elde edilebilir. Adım hızı yükseldikçe sargı akımları nominal değerlerine yükselemeden ikinci adım darbesi geldiğinden, moment, düşük hızlardaki moment kadar yüksek olmamakta, motor hızlandıkça üretebileceği maksimum moment azalmaktadır. Fazlarda akımın çabuk yükselmesini sağlayacak sürücü devrelerle yüksek hızlarda motorda indüklenerek momenti arttırmak mümkündür.

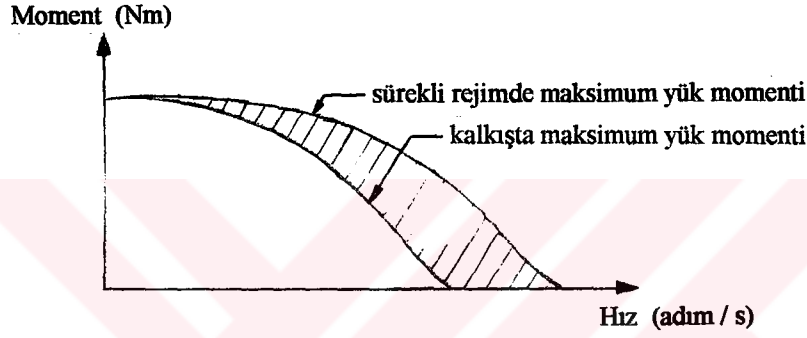
Bu eğrideki iki derin çukur, tüm step motorları sistemlerinde mekanik rezonansın etkileridir. Tek adım cevabının incelenmesinde de anlatıldığı gibi, adım hızı mekanik rezonansın doğal frekansa veya katlarına eşit ise veya bir başka deyişle ikinci adım darbesi geldiğinde rotor, birinci adıma ait kararlı konumunu geçmiş ve pozitif hıza sahip ise yeni darbenin gelişini senkronizasyonun kaybolmasına neden olur. Rotorun alacağı konum önceden kestirilemez. Rotor gidip yerleşeceği böyle bir konumdan sonra tekrar harekete geçmek için yeni konumuna ait uyarma darbesinin sürücü lojik'i tarafından gönderilmesini bekler. Bu durumda, motorun bir süre durmasına ve momentin düşmesine sebep olur.

Mekanik salınımın sebep olduğu bu istenmeyen davranış, motorların salınımlarının azaltılması bir ölçüde olsa giderilebilmektedir.

### 3.6. Kalkışta Maksimum Yük Momenti Eğrisi;

Özellikle açık çevrim çalışmada, duran bir sistemi istenen pozisyona getirebilmek için; motorlara uygulanan uyarma darbelerinin motor tarafından hiç kaçırılmadan takip edilmesini sağlamak çok önemlidir. Ancak gelen uyarma darbelerinin sıklığı motorun miline bağlı yükü sıfır hızından itibaren kaldırıp hızlandırmasına izin vermeyebilir. Bu yüzden step motorları için kalkışta maksimum yük momenti eğrileri tanımlanır.

Bu eğrilerde yatay eksen motorun adımlama hızına yani motora uygulanacak giriş darbelerinin frekansına karşılık düşerken, dikey eksen de bu hızlarda motorun giriş darbeleriyle olan senkronizasyonunu bozmadan harekete geçirebileceği duran yükü gösterir. Bu eğrilerin her biri belli bir eylemsizlik için tanımlanır ve sadece o eylemsizlik için geçerlidir. Belli bir yükü, uyarma darbelerinin hiç birini kaybetmeksizin üzere kullanılacak maksimum hıza maksimum kalkış hızı denir. Bu hız aynı yük için maksimum sürekli rejim hızından çok daha düşüktür.



Şekil 3.6. Sürekli rejimde maksimum yük momenti ve kalkışta maksimum yük momenti eğrileri

Kalkışta maksimum yük momenti değerleri de tutma momenti eğrilerinden elde edilebilmektedirler. Bu moment değerleri de motorun sürülme şekline son derece bağlıdır. Akımın hızlı yükselebildiği sistemlerde tutma momenti de hızla artacağından ve adım konumuna erken yerleşeceğinden tüm sistemin hızlandırılabilmesi mümkündür.

Bu bölümde incelenen tüm büyüklüklerde de görüldüğü gibi, step motorun performansını arttırmak için adım cevabının iyileştirilmesi gerekmektedir. Bu sorun bu güne kadar genellikle mekanik yolla veya elektronik katlarla ya da basit prensipli özel sürücü devrelerle çözümlenmeye çalışılmıştır.

#### 4. ELEKTROMANYETİK TEORİSİ, STEP MOTORLARIN YAPILARI VE MOMENT ÜRETMELERİ

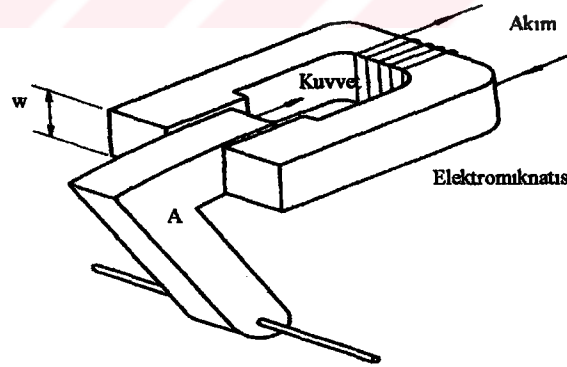
Bir önceki bölümde step motorların tipi yapıları, çalışma prensipleri avantajları ve kullanım alanları üzerinde duruldu. Bu bölümde ise ayrıntılı olarak elektromanyetik teorisi ve step motorların moment üretmeleri elektrodinamik yaklaşımla incelenecektir.

##### 4.1. Değişken Relüktanslı Tip Step Motorda Statik Moment Oluşumu

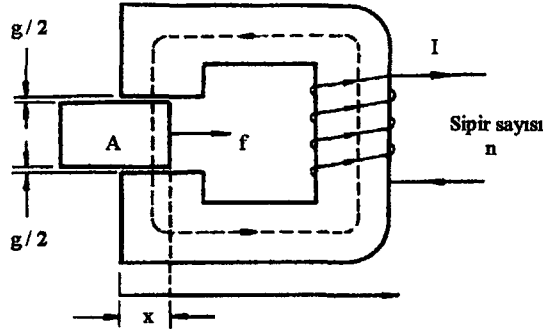
Bir elektrik motorunda moment oluşumu bir çok yöntem yardımıyla açıklanabilir. Biz bu bölümde step motorun devre parametrelerine uygun olan manyetik enerji ve koenerji prensibine dayanan yöntem ile moment oluşumunu açıklanacaktır. Öncelikle ideal olarak rotor ve statorun sonsuz manyetik geçirgenliğe sahip oldukları düşünülmüş adım-adım rotor ve statorun manyetik doyuma ulaşmaları durumu dikkate alınmıştır.

##### 4.1.1. Sonsuz manyetik geçirgenlik durumu;

Şekil 4.1.'de görülen düzenek tarafından üretilen manyetik alan nedeniyle hareket eğiliminde olan demir parçasının davranışını açıklamak üzere şekil 4.2.'deki model kullanılabilir.



Şekil 4.1. Elektromanyetik alan içindeki demir parçası



Şekil 4.2. Step motor prensibinin açıklanabileceği basit bir model

Şekil 4.2.'de görüldüğü gibi  $n$  sarımlı bobinden geçen  $I$  akımı manyetik akıyı meydana getirirken, demir parçasını  $x$  yönünde hareket ettirecek bir  $f$  kuvvetinin varlığından söz edilebilir. Burada demir parçası step motorun rotor dişi,  $n$  sarımlı bobinin üzerinde bulunduğu yapı ise step motorun statoru olarak düşünülebilir. Öncelikle hava boşluğunda meydana gelen manyetik akıyı tariflemek gerekirse Ampere yasasının bilinmesi gerekir.

**Ampere Yasası;** Birimi metre başına Ampere olan  $H$  vektörü ile birimi tesla olan  $B$  vektörü arasında;

$$B = \mu H \quad (4.1.)$$

Bağıntısı bulunduğu bilinmektedir. Ampere yasası  $H$  vektörü ile ilgilidir. Ampere' in deneylere dayanarak elde ettiği yasanın ifadesi aşağıdaki gibidir.

$H$  vektörünün kapalı bir eğri boyunca hesaplanan eğrisel integrali, integrasyon yolunun çevrelediği toplam akıma eşittir.

Yukarıdaki ifadenin matematiksel notasyonu Şekil 4.2.'deki devre için yapılacak olursa;

$$\int H dl = nI \quad (4.2.)$$

bağıntısı yazılabilir. Bu eşitliğin sol tarafı;

$$\int H dl = H_g \left(\frac{g}{2}\right) + H_g \left(\frac{g}{2}\right) + H_i = H_g g + H_i l \quad (4.3.)$$

yukarıdaki gibi yazılabilir. Burada  $H_g$  : boşluklardaki manyetik alan yoğunluğu,  $H_i$  : sarımın üzerinde bulunduğu metalin içindeki manyetik alan yoğunluğu,  $l$  : devredeki toplam yol olarak yer almaktadır.

Eğer malzemenin manyetik geçirgenliği çok yüksek olursa (4.1.) bağıntısından hareketle malzemenin içindeki manyetik alan yoğunluğunun ( $H_i$ ) sıfır olacağı söylenebilir. Bu durumda (4.3.) bağıntısından;

$$H_g = nI/g \quad (4.4.)$$

yazılabilir. Hava boşluğundaki akı yoğunluğu ise;

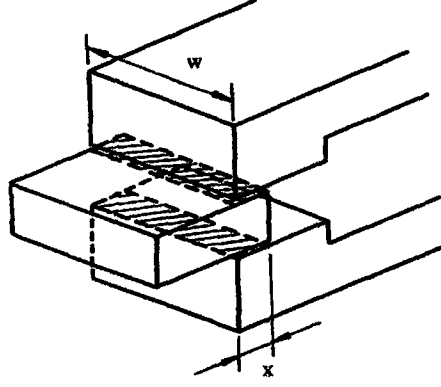
$$B_g = \mu_0 nI/g \quad (4.5.)$$

olarak tariflenebilir. Şekil 4.3.'de görülen düzenekte demir parçasının manyetik alana dik olarak giren kısmının genişliğinin  $w$  ve manyetik alan içindeki kısmının  $x$  olduğunu düşünerek manyetik alan tarafından kapsanmış kısmının alanını  $wx$  olduğunu söyleyebiliriz. (4.4.) bağıntısı bu alan ile çarpılırsa;

$$\phi = wx\mu_0 nI/g \quad (4.6.)$$

manyetik akı miktarı bulunur.





Şekil 4.3. Kaplanmış alan

Bu arada bir bobindeki sargı sayısı ile bu sargıdan geçen kuvvet çizgilerinin sayısının çarpımı;

$$\psi = n\phi = wx\mu_0 n^2 I/g \quad (4.7)$$

şeklinde verilebilir. Bu ifadeye dayanarak  $\Delta t$  süresince  $\Delta x$  kadar bir hareket olduğunu düşünerek akı bağıntısındaki artış aşağıdaki gibi olur.

$$\Delta\psi = wx\mu_0 n^2 I/g \Delta x \quad (4.8)$$

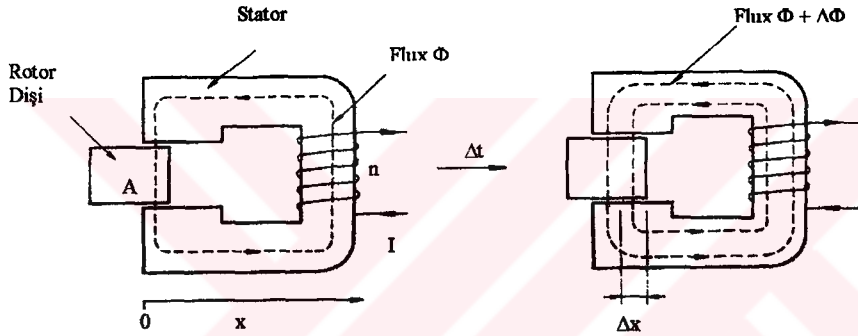
Bununla birlikte  $\Delta x$  kadarlık hareket sonucu endüklenen e.m.k.'nin değeri de aşağıdaki gibidir.

$$e = -\Delta\psi/\Delta t = -wx\mu_0 n^2 I \Delta x/g\Delta t \quad (4.9)$$

yukarıdaki – işareti endüklenen gerilimin akıma ters yönlü olduğunu göstermektedir. Söz konusu I akımını bir kaynak tarafından sağlandığından  $\Delta t$  süresince bu kaynağın yapmış olduğu iş;

$$\Delta P_i = I |e| \Delta t = (wx\mu_0 n^2 I^2 / g) \Delta x \quad (4.10.)$$

şeklinde ifade edilebilir. Yukarıdaki bağıntılarda bobinin direnci hesapları kolaylaştırmak için çok küçük olduğu düşünülmüştür.



Şekil 4.4.  $\Delta t$  süresinde rotor dişinin manyetik alan tarafından  $\Delta x$  kadarlık hareketi

Kaynak tarafından yapılan işin bir kısmı mekaniki harekete dönüşmekte, bir kısmı ise hava boşluğunda manyetik alan enerjisi olarak harcanmaktadır. Bu kaybolan enerjideki artış miktarı;

$$\begin{aligned} \Delta W_m &= 1/2 B_g^2 / \mu_0 x (\text{Hava boşluğundaki artış}) \\ &= B_g^2 g w \Delta x / 2 \mu_0 \end{aligned} \quad (4.11.)$$

(4.11.) bağıntısından hareketle  $\Delta P_i$  gücünün yarısının mekanik enerjiye dönüştüğünün yarısının da hava boşluğunda manyetik enerji şeklinde açığa çıktığını söyleyebiliriz. Mekanik olarak yapılan iş;

$$f\Delta x = \mathbf{B}_g^2 g w \Delta x / 2\mu_0 \quad (4.12.)$$

şeklinde ifade edilebilir ve her iki taraftaki  $\Delta x$ ' ler sadeleştirilirse;

$$\Delta x = \mathbf{B}_g^2 g w / 2\mu_0 \quad (4.13.)$$

bağıntısı elde edilir ki  $\mathbf{B}_g$ ' nin değeri (4.5.)'da yerine yazılırsa;

$$f = w \mu_0 n^2 I^2 / 2g \quad (4.14.)$$

elde edilebilir. Diğer taraftan hava boşluğundaki manyetik enerji;

$$W_m = \mathbf{B}_g^2 g w \Delta x / 2\mu_0 \quad (4.15.)$$

şeklinde yazılabilir. (4.13). ve (4.15.) bağıntıları karşılaştırılırsa;

$$f = dW_m/dx \quad (4.16.)$$

eşitliği elde edilir. Bu durumda en önemli nokta bütün bu ifadelerde ve temel olarak demir parçasının yer değiştirmesi sırasında I akımının sabit kaldığının varsayılmış olmasıdır. Bu sebepten 4.16. eşitliği;

$$f = (\partial W_m / \partial x)_{I=\text{sabit}} \quad (4.17.)$$

şeklinde yazılmalıdır. Bu bağıntı bobin direncinin sıfır olmadığı bir başka deyişle genel durum için geçerlidir. Diğer taraftan manyetik akımında sabit kaldığı varsayılarak bir başka bağıntıda elde edilebilir fakat (4.17.) bağıntısı step motorların incelenmesinde kullanmak için daha uygundur.

#### 4.1.2. Sabit manyetik geçirgenlik durumu;

Yukarıda bahsedilen ve stator yapısının sonsuz manyetik geçirgenliği olduğu durumda manyetik alan sadece hava boşluğunda meydana gelir ve bu tür bir olayın yukarıda yapıldığı gibi matematiksel incelemesi oldukça basittir. Diğer taraftan statoru oluşturan malzemenin sabit bir manyetik geçirgenliğe sahip olması durumunda ise manyetik alan sadece hava boşluğunda değil stator da ortaya çıkar. Bu gibi durumları elektromanyetik teori ile incelemek kolay değildir. Bu nedenle bu bölümde rotora hareket sağlayan kuvvetin ifadesi devre parametrelerine bağlı olarak ve statoru oluşturan malzemenin manyetik geçirgenliğinin manyetik alanın bir fonksiyonu olmadığı göz önüne alınarak verilecektir.

Şekil 4.3.'de verilen modelde akı bağıntısı bobin endüktansı cinsinden;

$$\psi = LI \quad (4.18.)$$

şeklinde ifade edilebilir. Bu durumda sistemdeki manyetik enerji;

$$W_m = \frac{1}{2} L I^2 \quad (4.19.)$$

bağıntısıyla verilir ve bu durumda demir parçası  $\Delta t$  süresi boyunca  $\Delta x$  kadar hareket ederse bobin endüktansında ( $L$ )  $\Delta L$  kadarlık bir artış olduğu söylenebilir. Aynı şekilde bobinde endüklenen e.m.k;

$$e = \Delta\psi/\Delta t = \Delta(LI)/\Delta t \quad (4.20.)$$

Eğer güç kaynağı bir akım kaynağı ise metal parçanın hareketi sırasında  $I$  akımını sabit olarak sağlıyorsa yukarıdaki eşitlik;

$$e = -I\Delta L/\Delta t \quad (4.21.)$$

şeklinde ifade edilebilir. Bu durumda kaynak tarafında endüklenen e.m.k.'ya eşit fakat ters yönlü olduğundan kaynak tarafından yapılan iş;

$$\Delta P_i = I |e| \Delta t = I^2 \Delta L \quad (4.22.)$$

şeklinde yazılabilir. Diğer taraftan manyetik enerji miktarındaki artış ise;

$$\Delta W_m = \frac{1}{2} I^2 \Delta L \quad (4.23.)$$

aynı şekilde 2.22. ve 2.23. bağıntıları karşılaştırılırsa kaynak tarafından yapılan işin yarısının manyetik enerjiye dönüştüğü görülür. Diğer yarı bölüm ise mekanik enerjiye dönüşmektedir ve aşağıdaki gibi ifade edilebilir;

$$\Delta P_o = f \Delta x = \frac{1}{2} I^2 \Delta L \quad (4.24)$$

ve bu eşitlikten kuvvet bağıntısı aşağıdaki gibi bulunur.

$$f = \frac{1}{2} I^2 \Delta L / \Delta x \quad (4.25.)$$

Yukarıdaki hesaplamalarda bobin direncinin sıfır olduğu ve kaynağın bir akım kaynağı olduğu düşünülmüştür. Fakat son olarak bulunan (4.25.) bağıntısı genel durumlara uygulanabilir. Bu durumda demir parçası üzerinde etkili olan kuvvet bobin endüktansını arttıracak yönde olacaktır.

#### 4.1.3. Manyetik doyuma ulaşma durumu;

Bütün step motorların karakteristik olarak statorlarının manyetik doyuma ulaşma durumları söz konusudur. Eğer bir step motor lineer B/H karakteristik bölgesinde çalışmak üzere tasarlandıysa bu motorun üretebileceği birim moment çok küçük olacağından söz konusu bu motor pratik uygulamalarda kullanılamayacak kadar büyük yapıya sahip olacaktır. Bu nedenle manyetik doyumu göz önüne almayan teori pratik değildir. Statordaki manyetik doyuma dikkate alan bir moment teorisi şu şekilde verilmiştir. (*Stepping motors and Their Microprocessor Control, T.Kenjo p.p.72,74*)

Şekil 4.3.'de verilen modeli temel alarak enerji dönüşümü incelenirse ki faz akımının oluşturduğu manyetik alanın yarattığı  $f$  kuvvetinin etkisiyle rotor  $x_0$ ' dan  $x_0 + \Delta x$  konumuna hareket etmektedir, manyetik akının  $(\psi)$   $x$  ve  $I$  akımına bağlı olduğu söylenebilir ve  $\psi(x, i)$  şeklinde ifade edilir. Eğer hareket sırasında  $I$  akımının sabit kaldığı düşünülürse güç kaynağı tarafından  $\Delta t$  zamanında yapılan  $\Delta P_i$  işi;

$$\Delta P_i = I v \Delta t = I \Delta \psi / \Delta t = I \Delta \psi \quad (4.26.)$$

yukarıdaki gibi verilebilir. Diğer taraftan  $\Delta t$  süresi boyunca rotor tarafında yapılan mekanik iş;

$$\Delta P_o = f \Delta x \quad (4.27.)$$

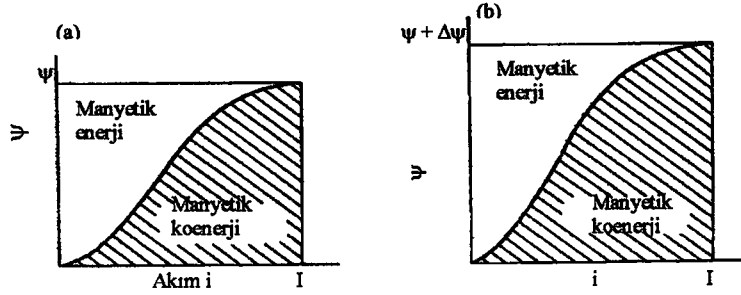
şeklide olur.  $\Delta x$  kadarlık bir hareket sonucu sistemin mekanik enerjisindeki artış miktarı;

$$\Delta W_m = \int_0^{\psi + \Delta \psi} i d\psi(x_0 + \Delta x, i) - \int_0^{\psi} i d\psi(x_0, i) \quad (4.28.)$$

şeklinde ifade edilebilir. Bu eşitlikte  $i$  akımı 0'dan  $I$  değerine kadar değişen bir değişken olarak dikkate alınmaktadır. Yukarıdaki ifade incelenecek olursa ilk terim rotorun  $x = x_0 + \Delta x$  konumunda iken sahip olduğu manyetik enerji ve ikinci terim ise rotorun  $x_0$  konumunda iken sahip olduğu bir başka deyişle başlangıç manyetik enerjisidir. (4.28.) bağıntısındaki her iki teriminde integral değerleri hesaplanacak olursa sistemin manyetik enerjisi;

$$\Delta W_m = I\Delta\psi - \int_0^I \psi(x, i) di \quad (4.29.)$$

şeklinde bulunur ve sistemin rotorun her iki konumundaki manyetik enerji ve koenerjileri Şekil 4.5. olduğu gibi gösterilebilir.



Şekil 4.5. Sistemin manyetik enerji ve koenerji dağılımı

(4.26.) bağıntısında verildiği gibi yukarıdaki ifadenin ilk terimi güç kaynağı tarafından yapılan iş miktarıdır. Buradan hareketle yukarıdaki ifade basitleştirilebilir ve statoru n sarımdan ibaret bir step motor için moment ifadesi sonuç olarak şu şekilde verilebilir.

$$T = \partial / \partial \theta \sum_{i=1}^n \int_0^i \psi(\theta, i) di_i \quad (4.30.)$$

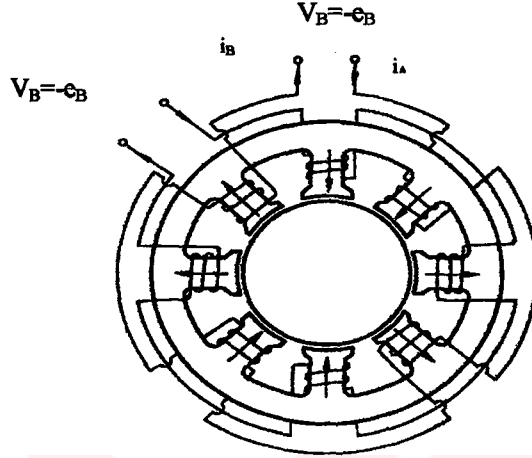
#### 4.2. Hibrid Step Motorda Moment Üretilmesi

Bu bölümde hibrid step motorda moment üretilmesi incelenecektir. Değişken relüktanslı step motorun moment oluşturmasına ek olarak bu tip step motorlarda sürekli mıknatıslığın etkisi de göz önünde bulundurulacaktır.



#### 4.2.1. Analitik yaklaşım

Şekil 4.6.'daki kutup yapısına sahip iki kutuplu hibrid motor üzerinde durulacaktır. Şekil 4.6' da rotor dişleri çizilmemiştir. Stator sargıları; 1, 3, 5,7, birinci fazı 2, 4, 6, 8, ikinci fazı oluşturacak şekilde seri olarak bağlanmışlardır.



Şekil 4.6. İki fazlı hibrid step motor modeli

Yapacağımız analizi kolaylaştırması açısından, sargı dirençlerinin etkisi, fuko akımları, karşılıklı endüktans, histerisiz ve darbe momentleri ihmal edilmiştir. Ayrıca motor manyetik devresinin lineer olduğu varsayılmıştır.

Temel enerji dönüşüm yasasından başlayacak olursak,

$$\text{Elektrik Enerjisi} = \text{Mekanik Enerji} + \text{Manyetik Devredeki Enerji} \quad (4.31)$$

Bu teorinin devamında aşağıdaki denklemler için şu açıklamalar göz önünde bulundurulmalıdır.

- (i) Küçük harfler ile ifade edilen semboller, zamana bağlıdır.
- (ii) Büyük harfler ile ifade edilen semboller, Laplace dönüşümlüdür ve sabittir.
- (iii) Zamana bağlı olarak değişen moment  $\tau$  ile ifade edilmiştir, bunu  $t$  ile karıştırmamak gerekir.

Böylece (4.31) denklemini şu şekilde yazılabilir,

$$-(e_A i_A + e_B i_B) = \tau \frac{d\theta}{dt} + \frac{d}{dt} \left( \frac{1}{2} i_A^2 L_A + \frac{1}{2} i_B^2 L_B \right) \quad (4.32.)$$

Manyetik devrenin lineer olduğu ve iki faz arasındaki karşılıklı endüktansın ihmal edildiğini kabul ettiğimiz için, momenti iki bileşene ayırabiliriz,

$$\tau = \tau_A + \tau_B \quad (4.33.)$$

Böylece;

$$-e_A i_A = \tau_A \frac{d\theta}{dt} + \frac{1}{2} \frac{d}{dt} i_A^2 L_A \quad (4.34.)$$

$$-e_B i_B = \tau_{BA} \frac{d\theta}{dt} + \frac{1}{2} \frac{d}{dt} i_B^2 L_B \quad (4.35.)$$

olur.

#### 4.2.2. Moment üretiminde sürekli mıknatısın etkisi

Her bir faz için uç gerilimi, faz sargılarında sürekli mıknatıslık akısı tarafından üretilen gerilim ve faz endüktansından geçen akımın oluşturduğu gerilim olarak iki bileşenin toplamı şeklinde ifade edilir.

A fazı için (4.34) denklemini aşağıdaki gibi yazılabilir;

$$-(e_{gA} + e_{LA})i_A = \tau_A \frac{d\theta}{dt} + \frac{1}{2} \frac{d}{dt} i_A^2 L_A \quad (4.36.)$$

Burada  $e_{LA}$  A fazından geçen akımın endüklediği gerilim,

$$e_{LA} = -\frac{d}{dt}(i_A L_A) \quad (4.37.)$$

olarak verilir ve bu denklem (4.36)' da yerine yazılırsa;

$$-e_{gA}i_A + i_A \frac{d}{dt}(i_A L_A) = \tau_A \frac{d\theta}{dt} + \frac{1}{2} \frac{d}{dt} i_A^2 L_A \quad (4.38.)$$

$$i_A \frac{d}{dt}(i_A L_A) - \frac{1}{2} \frac{d}{dt} i_A^2 L_A = i_A^2 \frac{dL_A}{dt} + L_A i_A \frac{di_A}{dt} - \frac{1}{2} L_A \frac{di_A^2}{dt} - \frac{1}{2} i_A^2 \frac{dL_A}{dt} \quad (4.39.)$$

ikinci ve üçüncü terim birbirini götürür ve;

$$\frac{1}{2} i_A^2 \frac{dL_A}{dt} = \frac{1}{2} i_A^2 \frac{dL_A}{d\theta} \frac{d\theta}{dt} \quad (4.40.)$$

Bu (4.36)'da yerine yazılırsa A fazının momenti;

$$\tau_A = -\frac{e_{gA}i_A}{\theta} + \frac{1}{2} i_A^2 \frac{dL_A}{d\theta} \quad \text{ve} \quad \theta = \frac{d\theta}{dt} \quad (4.41.)$$

(4.41.) denkleminin sağ tarafındaki ikinci terim, rotor konumunun değişimi ile oluşan momenti temsil eder ki bu da değişken relüktanslı motorun moment üretme prensibidir. Tipik hibrid step motorlarda, faz endüktansından dolayı oluşan moment kalıcı momente göre değişiminin çok düşük olması sebebiyle ihmal edilebilir.

$$\tau = -(e_{gA}i_A + e_{gB}i_B)/\theta \quad (4.42.)$$

her bir fazda endüklenen gerilim;

$$e_{gA} = n(-\phi_1 + \phi_3 - \phi_5 + \phi_7) \quad (4.43.)$$

$$e_{gB} = n(-\phi_2 + \phi_4 - \phi_6 + \phi_8) \quad (4.44.)$$

Burada  $n$ ; kutuplardaki sarım sayısını,  $\phi_k$ ; Kutupların sabit manyetik akısını ifade etmektedir.  $e_g$  'nin değişiminin biraz harmonik içeren sinüs dalga değişimine benzediğini biliyoruz. Eğer harmonikli bileşenler ihmal edilirse ;

$$e_{gA} = \omega C \cos(\omega t - \rho) \quad (4.45.)$$

$$e_{gB} = \omega C \cos(\omega t - \rho) \quad (4.46.)$$

olur ki, burada  $C$ ; motorun boyutlarına ve sarım sayısına bağlı bir sabittir,  $\rho$ ; radyal olarak faz açısıdır. Buradaki açısal frekans  $\omega$ , açısal hız ve rotor diş sayısına bağlıdır.

$$\omega = N_r \theta \quad (4.47.)$$

#### 4.2.3. Sabit moment

Her bir fazdaki akım ve endüklenen gerilimin sinüsoidal dalga şekline sahip olduğu ve aynı fazda oldukları kabul edildiğinde ;

$$i_A = -I_M \sin \omega t \quad (4.48)$$

$$i_B = +I_M \cos \omega t \quad (4.49)$$

(4.47.), (4.48.), (4.49) ifadelerini (4.42.) eşitliğinde yerine yazarsak;

$$\tau = -\frac{\omega C I_M}{\theta} [\sin(\omega t - \rho) \cos \omega t - \cos(\omega t - \rho) \sin \omega t] \quad (4.50.)$$

$$\tau = C N_r I_M \sin \rho$$

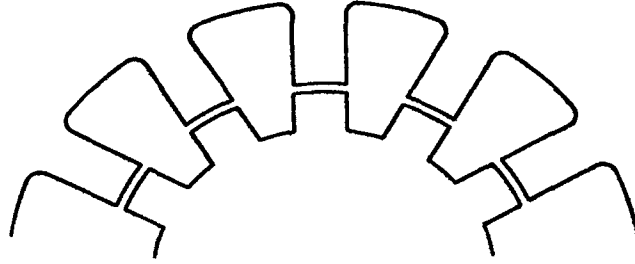
Yük momenti ile dengeli olmak zorunda olan motor momenti, “sin $\rho$ ” ile doğru orantılı olduğu için “ $\rho$ ” moment açısı veya yük açısı olarak adlandırılır. Eğer sabit mıknatısın endüklediği gerilim ve akımın her ikisi de sinüsoidal dalga şekline sahip iseler hibrid motorda moment sıçramaları olur. Bir çok uygulamada akım dalga şekli sinüsoidal değişimden farklı olduğundan moment sıçrama bileşenleri büyük olur. Ayrıca şunu da belirtmek gerekir ki rotor mıknatısındaki harmonikler sebebiyle uygulanan akım sinüsoidal değişime sahip olmasa da bu moment sıçramaları olacaktır.

#### 4.3. Diş Yapısı, Sayısı ve Kutup Düzeni

Step motorların geleneksel motorlardan çok farklı yapıya sahip oldukları hemen görülmektedir. Bir motorun moment üretebilmesi en önemli karakteristik yapısı dişlerine bağlıdır. Diğer tüm dönen veya lineer hareket eden motorlarda moment üretimi için diş yapısı gerekli değildir. Fakat yinede bu tip motorlarda uygulanan diş yapısının amacı stator ve rotor arasındaki reluktansı en aza indirmektedir. Step motorlarda ise moment üretimi için hem rotorda hemde statorda diş bulunması temel olarak gereklidir. Bu bölümde step motorlardaki diş yapısı, sayısı, ve bunların ilişkileri incelenecektir.

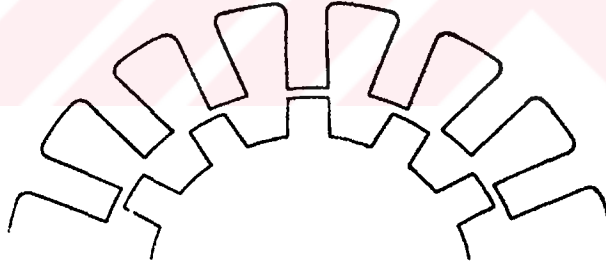
### 4.3.1. Diş yapısı

Değişik tipteki motorların diş yapıları üç temel grupta toplanabilir. Birinci grup şekil 4.7'de görüldüğü üzere stator ve rotorda eşit sayıda diş bulundurur. Bu tip yapı çok kütleli step motorlarda bulunan yapıdır.



Şekil 4.7. Rotorda ve statorda aynı sayıda diş bulunan yapı

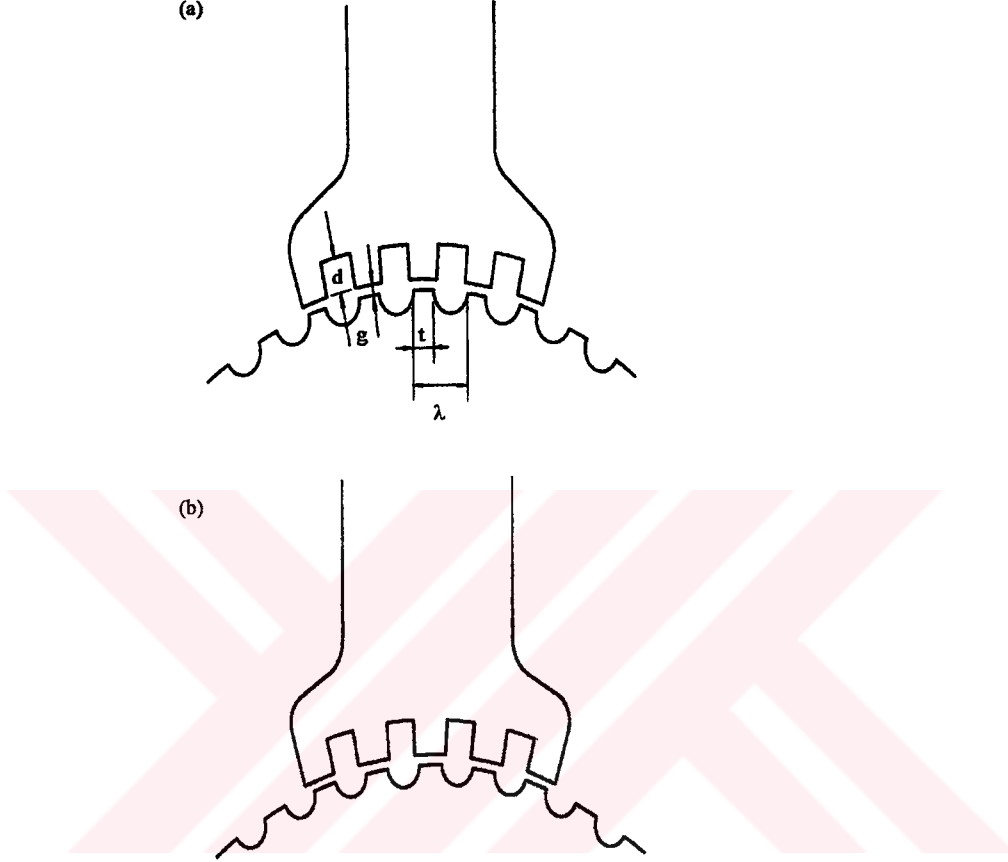
Bu tip yapıda dişler aynı anda enerjilendirilir ve enerjileri kesilir. İkinci tip yapı şekil 4.8.'de görüldüğü gibidir.



Şekil 4.8. Rotorda ve statorda farklı sayıda diş bulunan yapı

Bu tip yapılar genellikle tek kütleli değişken relüktanslı tip ve büyük adım açılı step motorlarda kullanılan türlerdir ve tür makinalarda dişler aynı anda enerjilendirilmezler. Son olarak üçüncü tip diş yapısı şekil (4.8.)'de verildiği gibidir. Bu yapıda stator dişleri stator kutupları altında gruplandırılmışken rotor dişleri homojen bir şekilde dağıtılmıştır. Bu tip yapı tek kütleli değişken relüktanslı tip ve küçük adım açılı step motorlarda

kullanılmaktadır. Bununla birlikte hibrid tip step motorlarda rotor ve stator diş sayısında çok küçük bir fark vardır. Örneğin, statorda 50 diş var iken rotorda 48 diş bulunur.



Şekil 4.9. Bir hibrid tip step motorun diş yapısı (a) rotor ve stator diş sayıları farklı (b) rotor ve stator diş sayıları aynı

Bir step motorun dizaynı sırasında diş/slot oranı karar verilmesi gereken en önemli unsurdur. Çünkü bu faktör bir step motorun statik moment karakteristikleri ile yakından ilgilidir. Aynı zamanda diş/slot oranı bir step motorun dinamik karakteristiklerinde belirleyici bir rol oynar çünkü bu oran her bir fazın endüktansını belirler.

Step motorlar çok değişken yapı gruplarına sahip olmalarına rağmen değişken relüktanslı tip step motorlarda moment üretimi Şekil 4.6'de gösterildiği gibi basit manyetik yapıya dayandırılabilir. Bu tür diş yapısına sahip motorlarda doyuma girmeyi de göz önüne alarak statik moment oluşumunu açıklayan bir teori ortaya koymuşlardır. Bu çalışmaya göre mümkün olan en büyük moment en küçük adım açısında ortaya çıkabilir. Ve optimum diş-

genişlik / diş-sayısı oranı ( $t/\lambda$ ) teorik olarak 0.42 dir. Yine bu teoriye göre pratik uygulamalarda bu oran 0.38 ile 0.48 arasında değişmektedir.

Modern değişken relüktanslı tip ve hibrid step motorlar aşağıda belirtilen genel karakteristiklere sahiptirler.

1. Diş-genişlik / Diş-sayısı oranı ( $t/\lambda$ ) 0.5 civarındadır.
2. Statordaki slot derinliği  $d$  yaklaşık olarak diş genişliğinin yarısı kadardır.
3. Rotor için diş-genişlik / diş-sayısı oranı ( $t/\lambda$ ) 0.38 ile 0.45 arasındadır.
4. Rotor için slot yapısı yarı daireseldir, stator için ise dikdörtgen veya yarı daireseldir.
5. Hava boşluğunun genişliği  $g$  mümkün olduğu kadar küçük olmalıdır. Yaklaşık olarak 0.02-0.05 arası.

#### 4.3.2. Faz, diş ve adım sayısı ilişkileri

$m$  fazlı rotorunda  $N_r$  adet diş bulunan bir motor için bir hareketteki adım sayısı şu genel ifade ile verilir.

$$S = m N_r \quad (4.51.)$$

Bu bağıntı bir fazın veya iki fazın devrede olduğu uygulamalarda doğrudur. Fakat yarı-adım sürme durumunda bu ifade aşağıdaki gibi olmalıdır.

$$S = 2m N_r \quad (4.52.)$$



$m$  fazlı bir step motorda herhangi bir fazdan anahtarlama başladığını düşünelim. Anahtarlama sıralamasının bir turu tamamlandığında yine ilk enerjilenen faza sıra geldiğinde rotor  $m$  adımlık bir hareket yapmış ve rotor bir diş genişliği kadar bir yol kat etmiştir. Motora bir adım genişliğinde hareket kazandırabilmek için  $m$  adet darbeye ihtiyaç varsa motorun bir devrini tamamlaması için  $mN_r$  adet darbeye gerek vardır. Şekil 4.8.'de verilen türde rotor ve statorun farklı sayıda dişe sahip oldukları durumda yukarıdakine benze şekilde bir eşitlik şu şekilde verilebilir.

$$q = |N_r - N_s| \quad (4.53.)$$

Bu bağıntıda  $N_r$  : rotor diş sayısı

$N_s$  : stator diş sayısı

$$q = N_s/m \quad (4.54.)$$

faz başına stator diş sayısı ifade edilebilir. (4.51.) ve (4.54.) bağıntılarından  $m$  değerini elemine edersek

$$S = N_r N_s/q \quad (4.55.)$$

$$S = N_r N_s / |N_r - N_s| \quad (4.56.)$$

Bulunabilir. Daha genel olarak (4.53.), (4.54.), (4.56.) bağıntılarından hareketle;

$$S = m(m+1)q \quad N_r > N_s \quad (4.57.)$$

$$S = m(m-1)q \quad N_r < N_s \quad (4.58.)$$

Bu tür motorlar için en geniş adım açısının  $30^\circ$  olduğunu şu şekilde gösterebiliriz.  $q$ ' nun alabileceği en küçük değer bir rotoru hareket ettirmek için en azından iki kutup gerekeceğinden 2 olabilir. En küçük mümkün  $m$  değeri ise 3' tür. Bu durumda 4.58. eşitliğinden;

$$S = 3 \times 2 \times 2 = 12$$

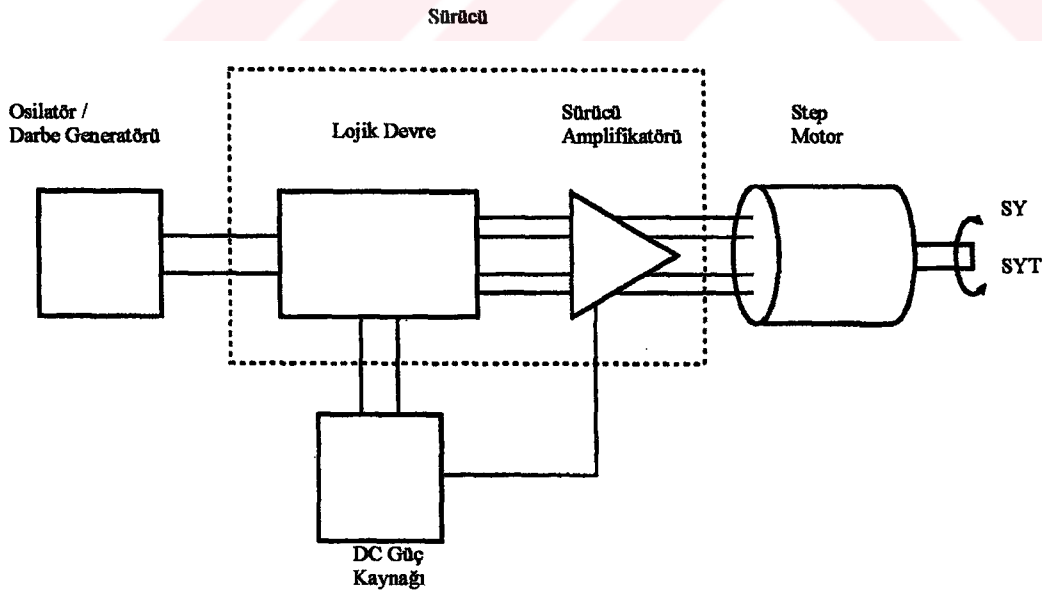
$$\theta = 360/S = 360/12 = 30^\circ \text{ olarak bulunur.}$$

## 5. STEP MOTORUN SÜRÜLÜŞÜ

Sürekli çalışma modunda da tek adımlı çalışma modunda da step motorun stator sargıları enerjilemek için özel bir tipte elektronik sürüş devresine ihtiyaçları vardır. Stator sargularından geçen akımı geçici veya periyodik olarak anahtarlamaktır.

Burada step motor sürüşünün bir kaç değişik tipleri vardır. Ama bunların hepsi aynı özelliğe ve işlevliğe sahiptir.

- İstenilen performansı elde edebilmek için giriş komutlarını kesen lojik bir sistem (örneğin dönüş yönü).
- İstenilen adımlama oranını elde etmek için bir darbe generatörü veya osilatörü.
- Motor ve sürüşün doğruluğundan emin olmak için koruma ve görüntüleme sistemi.
- Step motoru sürmek için gerekli enerji seviyesini sağlayan bir güç sürüş amplifikatörü.



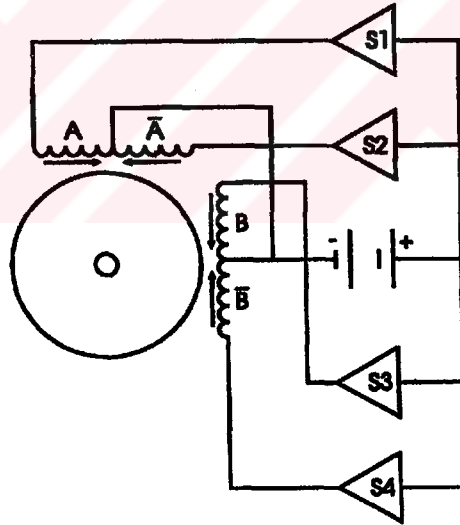
Şekil5.1. Tipik bir step motor sürüş sisteminin şematik düzenlemesi.

Takip eden kısımlarda en yaygın kullanılan step motor sürüşlerinin prensipleri detaylı olarak incelenecektir.

### 5.1. Tek Yönlü Sürüş

Adından da anlaşılacağı gibi tek yönlü sürüş akımın stator sargılarının her kısmından tek bir yönde akmasına izin verir. Bununla beraber dönen bir manyetik alan üretmek için ( her bir sargıda pozitif ve negatif akım akışını sağlamak için ), akımın sıra ile anahtarlanması gerekmektedir. Bir yolla bu potansiyelin üstesinden gelebilmek amacıyla stator sargıları sarılmalı ve bağlanmalıdır.

Bir tek yönlü sürüşte çalıştırmak için step motor 4 stator sargısının eşdeğeri ile donatılmalıdır. İki fazlı bir step motor için bu iki stator sargısının da ortasından yapılacak bir bağlantı ile gerçekleştirilebilir. Şekil 5.2. stator sargısı orta uçlu bir iki fazlı step motorun tek yönlü sürüş bağlantısının tipik bir düzenlemesini göstermektedir.



Şekil 5.2. Step motor tek yönlü tipik sürüş düzeni

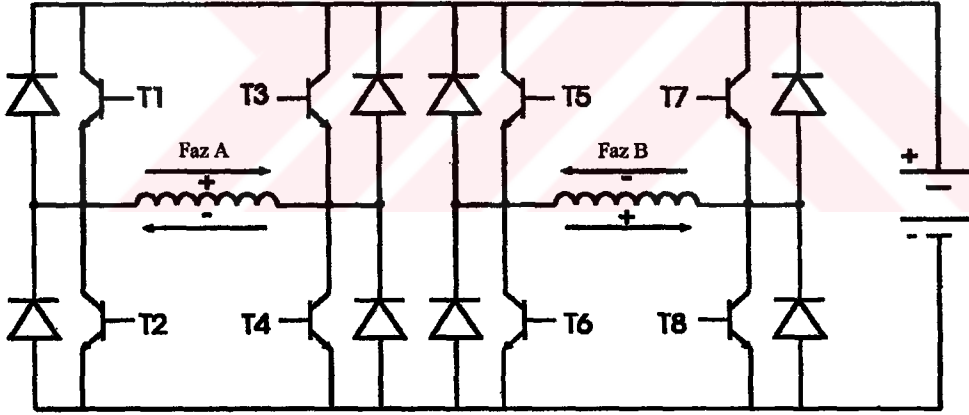
Anahtarlar mekanik veya elektronik şekilde ya tam açık yada tam kapalı olarak yapılabilirler. Her bağımsız anahtar her bir adımda istenilen akım akışını sağlamak için kullanılır. İsmen A veya A' ve/veya B veya B' aynı anda A ile A' 'den aynı şekilde B ve B' den akım akıtmanın mümkün olmayacağını görmek önemlidir.

Sadece A sargısından pozitif bir akım akıtmak için S1 anahtarı kapalı olmalıdır. Diğer üç anahtar açık kalmalıdır. Eğer aynı zamanda S2 de kapalı olsaydı oluşan iki alan birbirine ters yönde olacak ve motorun performansının fayda sağlamada hiç bir yardımı olamayacaktır. S1, S2, S3, S4'ün seçilerek anahtarlanması tam adım ve yarım adım mod sıralarının oluşturulmasını sağlayacaktır.

Tipik olarak bir tek yönlü sürüşten beslemede, eğer 4 stator sargısı da ayrı ise step motor 8 kabloya eğer ikişerli gruplarla bağlanırsa 6 kabloya ( Şekil 5.2.'de ), eğer ortak uçlar içten bağlı ise 5 kabloya sahip olacaklardır. Şekil 5.4. değişik sürüş tipleri için sargı konfigürasyonlarını özetlemektedir .

## 5.2. Çift Yönlü Sürüş

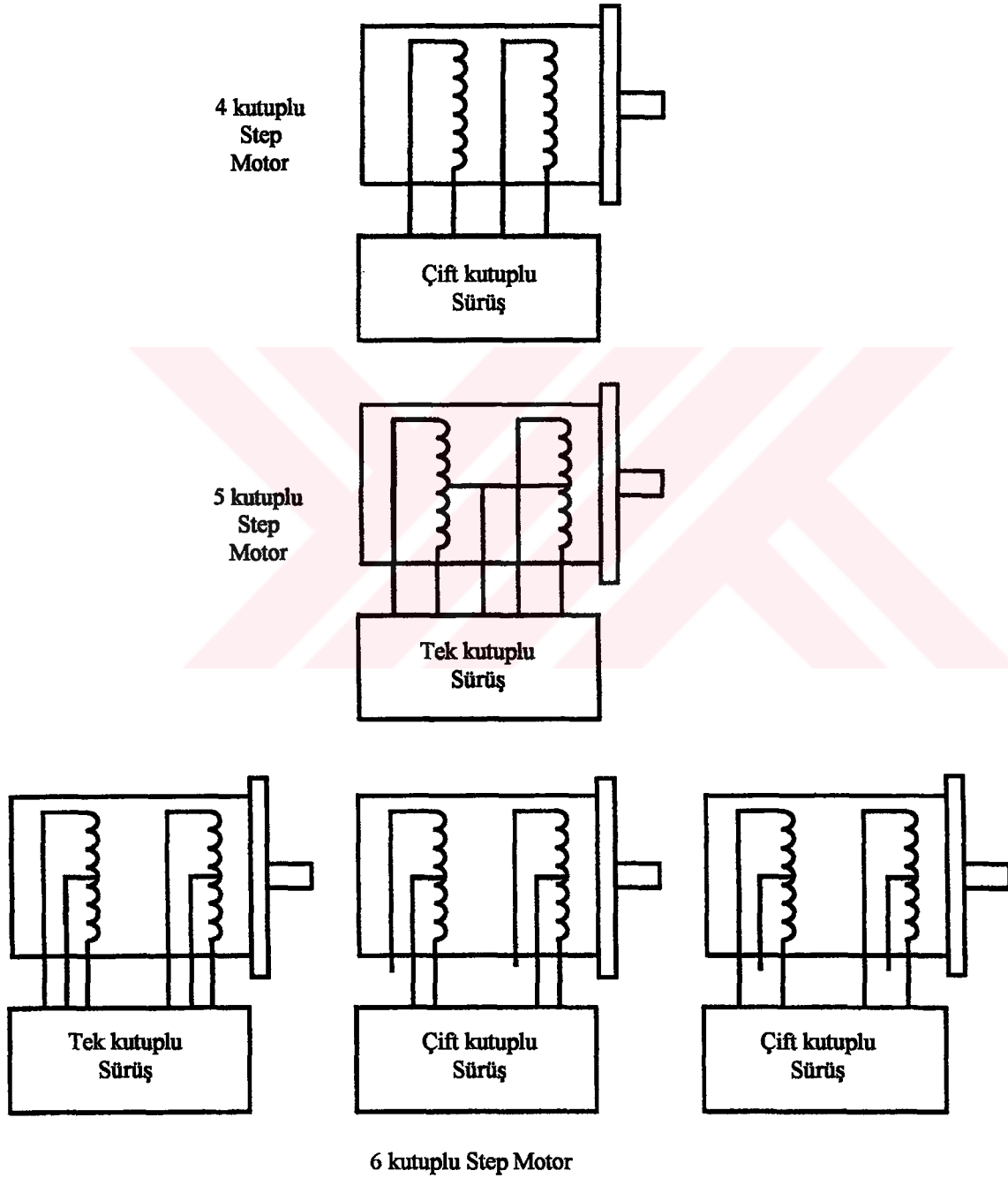
Bir iki yönlü sürüş ile, çıkış polaritesi akımın her iki yönde de akıtılması sağlanarak değiştirilebilir. Step motor iki efektif sargıya sahip olmalıdır ve sadece 4 kablo ile çalıştırılabilir.



Şekil 5.3 İki fazlı step motor için tipik iki yönlü sürüş düzeni.

Transistörlerin ( T1-T8 ) fonksiyonları her bir stator sargısından geçen pozitif ve negatif akım akışının sırasını istenildiği gibi sağlamaktadır. A fazından pozitif akımın geçmesi için T1 ve T4 transistörlerinin iletimde olabilmeleri için tetiklenmeleri gerekmektedir. A fazında negatif akım için T3 ve T2 transistörleri anahtarlanmalıdır. Böylece tek d.c güç kaynağı A fazından her iki yönde de akımın akmasını sağlayabilmektedir. Benzeri olarak pozitif ve negatif akımların B fazından da akması sağlanabilir.

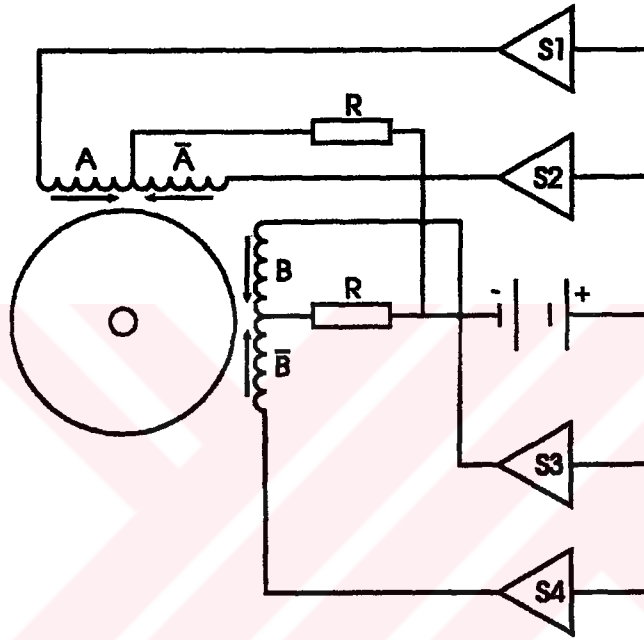
Sürekli çalışma için transistör anahtarlamalarının doğruluğu ve güvenilirlik kontrolü gereklidir. Bu Şekil 5.1. 'de gösterilen tipin lojik devresi ile elde edilir. Her bir adım sırasında sargıların tamamının kullanımı yani tek yönlü sürüşteki gibi sargının yarısıyla sınırlama olmamasından dolayı çift yönlü sürüş step motorun çalışmasını algılamasını sağlar.



Şekil 5.4. Tek veya çift yönlü sürüşlerin step motorun stator sargı bağlantıları ile bağlanabilme düzeni

### 5.3. R/L Sürüşü

Önceden açıkladığımız tek yönlü ve çift yönlü sürüşler için akım, istenilen stator sargısına uygulanan bir gerilimin uygulanması ile akıyor diye kabul edilmişti. Pratikte standart bir kaynaktan elde edilen akımın anahtarlanması yüksek frekansları yüksekçe olan endüktif sargılardan sertçe etkilenecektir. Bu dezavantajın üstesinden gelmek için pratikteki step motor sürüşlerine bu etkileri minimize edecek şekilde teknikler ilave edilir.



Şekil 5.5. R/L sürüşünde stator sargılarının ortak hattına seri olacak uygun dirençlerin bağlanması.

Dirençlerin fonksiyonu artan gerilimin stator sargılarına ilave edilmesine izin verir. Amaç step motorun performansını arttırmaktır ( özellikle yüksek adım oranlarında ). Direncin (R) değeri şu formülden hesaplanabilir,

$$R = ( V_s - V_m ) / I \quad (5.1.)$$

Burada;

$V_s$  : Kaynak gerilimi

$V_m$  : Nominal motor gerilimi

$I$  : Nominal motor akımı

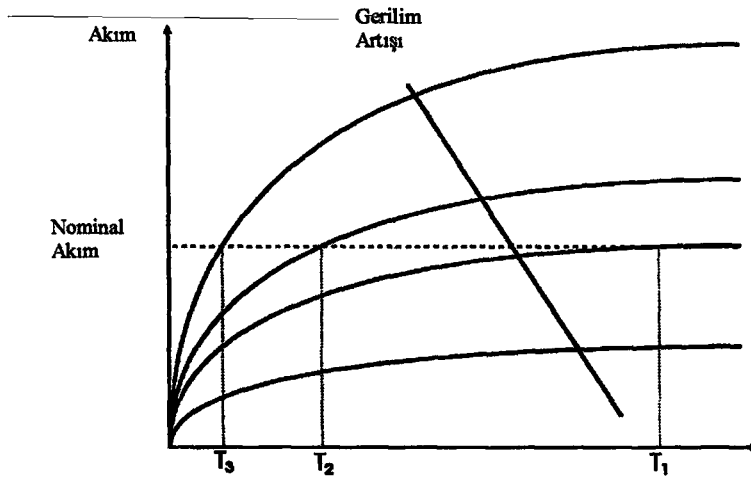
Bu sürüş sistemi çok basittir ama dezavantajlardan dolayı istenilmez. Verim düşüktür çünkü seri dirençler üzerinde çok büyük güç harcanmaktadır.

#### 5.4. Çift Seviyeli Sürüş

Çift seviyeli sürüş istenilen çıkışı elde etmek için dikkatlice seçilmiş yüksek ve alçak gerilimlere sahip iki güç kaynağının kullanımı ile gerçekleştirilir. Her adımın başlangıcında yüksek gerilim stator sargularına uygulanır ve akım sargının doğal endüktifliğinden dolayı üstel olarak değişecektir. Stator akımı nominal değere ulaştığında yüksek gerilim kesilir ve hiç beklemeden alçak gerilim kaynağı ile akımın nominal değerde kalması sağlanır. Darbenin sonunda alçak gerilimde kesilir.

#### 5.5. Kısıyıcı Regüleli Sürüş

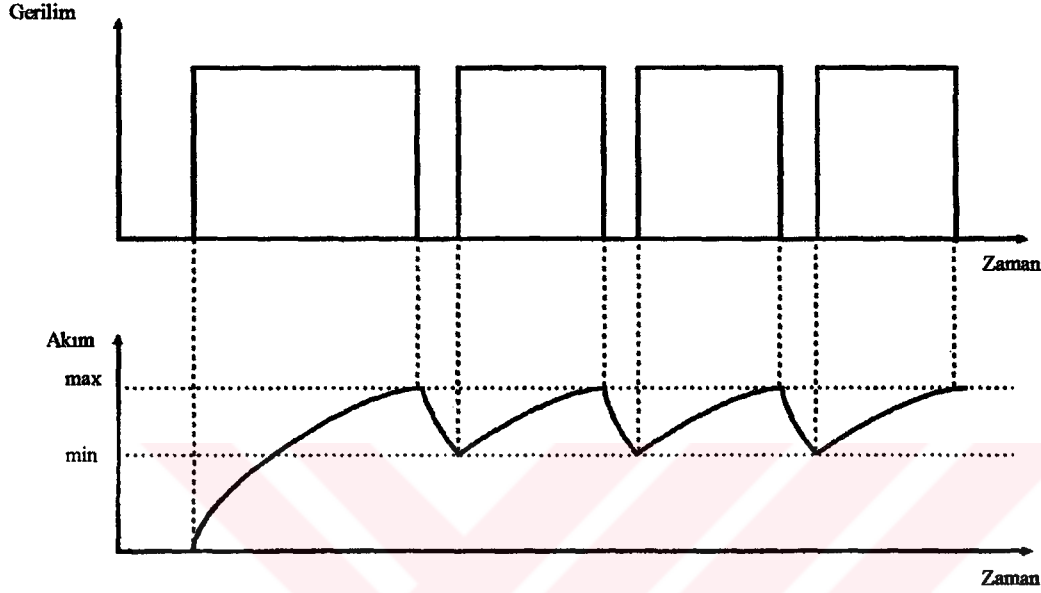
Kısıyıcı regüleli sürüş daha karışık ve verimli step motor sürüşüdür ve step motorun çalışma performansını arttırmak için kullanılır. Bu ise her değişimde stator sargularından akan akımın oranını artırma ile sağlanır. Şekil 5.6. değişik gerilimler ve endüktif devre için zamana karşı akımın karakteristiğini göstermektedir. Uygulanan gerilimin yükseldiği görülebilir. Akım kısa bir süre içinde nominal değerine ulaşır bu ise zaman sabitinin göz ardı edilmesidir. Motor üzerindeki etki uygulanan elektrik gücünün ortalama seviyesini arttırmaktadır ve bu da yükü süren mekaniki çıkış gücünü benzeri olarak arttıracaktır.



Şekil 5.6. Değişik gerilim değerlerindeki endüktif devredeki akım-zaman değişimi



Kıyıcı regüleli sürüşte bir yüksek gerilim kaynağı motorun nominal geriliminin 10 ila 20 katı kadardır ve bu kaynak başlangıçta stator sargılarına uygulanır ve müsaade edilen akım sargının endüktansından dolayı üstel olarak artar. Bir kıyıcı sürüşünün çıkış gerilim ve akımının dalga şekilleri Şekil 5.7.'de gösterilmiştir .



Şekil 5.7. Kıyıcı regüleli sürüşte akım ve gerilimin zamana göre değişimi

Rotorun takip ettiği döner bir manyetik alan üretebilmek için bu gerilim ve akımın dalga şekilleri stator sargılarına tam veya yarım adım mod da uygulanır.

Her adım darbesi periyodunun başında sargılardan geçen akım çabucak oluşacaktır. Akım önceden belirlenmiş bir seviyeye ulaştığında gerilim kesilir ve akımın sarım direnci ve endüktansından dolayı motor sargılarında dolaşmasına izin verilir. Bu akım bir maksimum değere kadar yavaşça sönecektir ve bu seviyeye geldiğinde makineye gerilim tekrar uygulanır ve akım tekrar maksimum seviyesine çıkar. Bu uygulama periyot boyunca devam eder ve bu yüzden sargılardan geçen akımın ortalama değeri makina' nın kullanımı için uygundur.

Kıyıcı regüleli sürüşün çalışması doğal olarak kalkış ve duruş uygulamalarında çok uygundur. Çünkü yüksek gerilim ve akım rotorun ivmelenmesi için uygundur.

## 6. YAPILAN UYGULAMANIN YAZILIMI

Bu bölümde sunulan yazılım ileride yapılacak farklı uygulamaların yazılımlarının da ilave edilebileceği şekilde Turbo Pascal programında hazırlanmıştır.

### 6.1. LP.pas

```
program LP;
```

```
uses Crt,Screen,KeyBoard,TypesLP,StpProcs,MCtrl,StrOP;
```

```
const
```

```
  ProgInfo : string[30] = 'Laser Pointer Controller 1.0';
```

```
  Mx : byte = 5;
```

```
  My : byte = 4;
```

```
  MenuSay = 5;
```

```
  Menu : array[1..MenuSay] of string[20] =
```

```
    (' Otomatik çalışma ',
```

```
     ' Manuel çalışma ',
```

```
     ' Program Hakkında ',
```

```
     ' Program Ayarları ',
```

```
     ' Programdan çıkış ');
```

```
  MSayac : byte = 1;
```

```
procedure About;
```

```
const
```

```
  XPos : byte = 20;
```

```
  YPos : byte = 8;
```

```
var
```

```
  Ekran : TEkran;
```

```
  Atr : byte;
```

```
  Ch : char;
```

```
begin
```

```

EkraniSakla(Ekran);
Atr := White+16*Blue;
KutuYap(XPos, YPos, 40, 6, Ince, Atr);
Atr := LightGray+16*Blue;
YazXY(XPos+10, YPos, ' PROGRAM HAKKINDA ', Atr);
YazXY(XPos+5, YPos+2, ProgInfo, Atr);
YazXY(XPos+3, YPos+4, 'Mustafa G.Aydeniz  İstanbul 1998', Atr);
StatusLine(#17#196#217':Devam');
repeat
  if KeyPressed then
    begin
      Ch := ReadKey;
      if Ch in [kbEsc, kbEnter] then Break;
    end;
until False;
EkraniGeriAl(Ekran);
end;

procedure DoneScreen;
begin
  TextAttr := Normal;
  ClrScr;
  CursorOn;
end;

{ ----- Ayarlari goster ----- }

procedure ShowSetup;
const
  XPos : byte = 9;
  YPos : byte = 17;
var

```

```

    Atr : byte;
begin
    Atr := White+16*Blue;
    KutuYap(XPos, YPos, 60, 5, Ince, White+16*Blue);
    YazXY(XPos+3, YPos+1, '1.Motor', Atr);
    YazXY(XPos+3+30, YPos+1, '2.Motor', Atr);

    Atr := LightGray+16*Blue;
    YazXY(XPos+3, YPos+2, 'Kalkış katsayısı :'+StrLF(Setup.xGecik, 4), Atr);
    YazXY(XPos+3, YPos+3, 'Rampa sayısı   :'+StrLF(Setup.xRampSay, 4), Atr);
    YazXY(XPos+3, YPos+4, 'Rampa katsayısı  :'+StrLF(Setup.xRampKat, 4), Atr);
    YazXY(XPos+3+30, YPos+2, 'Kalkış katsayısı :'+StrLF(Setup.yGecik, 4), Atr);
    YazXY(XPos+3+30, YPos+3, 'Rampa sayısı   :'+StrLF(Setup.yRampSay, 4), Atr);
    YazXY(XPos+3+30, YPos+4, 'Rampa katsayısı  :'+StrLF(Setup.yRampKat, 4), Atr);
end;

{ ----- }

procedure InitScreen;
var
    Atr : byte;
begin
    Atr := Black+16*Cyan;
    SetVideoMode(3);
    CursorOff;
    UseFontYenix;
    ClearUstLine;
    YazXY(25, 1, ProgInfo, Atr);
    MasaYap(Cyan+16*Blue);
    StatusLine(' ');
end;

```

```

procedure DrawMainMenu;
const
    MenuName : string[8] = 'ANA MENÜ';
var
    Atr : byte;
    I : byte;
begin
    Atr := White+16*Blue;
    KutuYap(Mx,My,SizeOf(Menu[1]),MenuSay+1,Ince,Atr);
    Atr := LightGray+16*Blue;
    YazXY(Mx+((SizeOf(Menu[1])-Length(MenuName)) div 2),My,MenuName,Atr);
    for I := 1 to MenuSay do YazXY(Mx+1,My+I,Menu[I],Atr);
    Atr := White+16*Red;
    YazXY(Mx+1,My+MSayac,Menu[MSayac],Atr);
    StatusLine(#24#25:'Y"nlendir '#17#196#217':Se#');
end;

{ ---- Ana Menu ---- }

procedure MenuAl;
var
    Ch : char;
begin
    YazXY(Mx+1,My+MSayac,Menu[MSayac],White+16*Red);
    repeat
        if OzelChar(Ch) then
            begin
                YazXY(Mx+1,My+MSayac,Menu[MSayac],LightGray+16*Blue);
                case Ch of
                    kbUp, kbLeft : begin
                        Dec(MSayac);
                        if MSayac <= 0 then MSayac := MenuSay;

```

```

        end;
    kbDown, kbRight : begin
        Inc(MSayac);
        if MSayac > MenuSay then MSayac := 1;
        end;
    kbAltX      : begin
        DoneScreen;
        Halt;
        end;

    end;
    YazXY(Mx+1,My+MSayac,Menu[MSayac],White+16*Red);
    end;
until Ch = kbEnter;
YazXY(Mx+1,My+MSayac,Menu[MSayac],LightGray+16*Blue);
case MSayac of
    1 : Tasdiklendi('Henüz proje aşamasında. ');
    2 : RunManualCtrl;
    3 : About;
    4 : begin
        RunSetup;
        ShowSetup;
        end;
    5 : if Tasdiklendi('Programdan çıkmak istediğinize emin misiniz?') then
        begin
            DoneScreen;
            Halt;
        end;
    end;
end;

begin
    InitScreen;

```

```
Tasdiklendi('Bu program windows altından çalıştırılmamalıdır.');
```

```
if not LoadSetup then
```

```
  begin
```

```
    HataMesaji('Program ayarları okunamadı. Yeni ayar dosyası oluşturulacak.');
```

```
    if not SaveSetup then HataMesaji('Program ayarları kaydedilemiyor...');
```

```
  end;
```

```
ShowSetup;
```

```
About;
```

```
DrawMainMenu;
```

```
repeat
```

```
  MenuAl;
```

```
until False;
```

```
end.
```



## 6.2. Screen.pas

```
{ $A-,B-,D-,E-,F-,G-,I-,L-,N-,O-,P-,Q-,R-,S-,T-,V-,X- }
{ $M 8192,0,0 }
```

```
unit Screen;
```

```
interface
```

```
const
```

```
    Normal = 7;
```

```
    Ters   = 112;
```

```
    Ince   = 1;
```

```
    Kalin  = 2;
```

```
type
```

```
    TEkran = array[1..4096] of byte;
```

```
procedure SetVideoMode(Mode : byte);
```

```
function GetVideoMode:byte;
```

```
procedure SetActivePage(PageNo : byte);
```

```
function GetActivePage:byte;
```

```
procedure SetCursorMode(CrsrMode : word);
```

```
function GetCursorMode:word;
```

```
procedure CursorOn;
```

```
procedure CursorOff;
```

```
procedure Locate(X,Y : byte);
```

```
function GetCursorXPos:byte;
```

```
function GetCursorYPos:byte;
```

```
procedure YazXY(X,Y : byte; Msg:string; Atr : byte);
```

```
procedure EkraniSakla(Ekran : TEkran);
```

```
procedure EkraniGeriAl(var Ekran : TEkran);
```

```
function IsVga:boolean;
```



```

function IsRenkli:boolean;
function InColorMode:boolean;
procedure StatusLine(Msg : string);
procedure GetFont(FontBuffer : array of byte);
procedure GetBiosFont;
procedure UseFontYenix;
procedure Fill(X,Y,W,H,C,Atr :byte);
procedure InitWindow(X,Y,W,H,Atr : byte);
procedure MasaYap(Atr : byte);
procedure KutuCiz(X,Y,W,H,Tip,Atr : byte);
procedure KutuYap(X,Y,W,H,Tip,Atr : byte);
procedure GolgeYap(X,Y,W,H : byte);
procedure HataMesaji(Msg : string);
function Tasdiklendi(Msg : string):boolean;
procedure ClearUstLine;
procedure EkranIAc;
procedure EkranIKapa;

```

implementation

```
uses NewFont,Crt;
```

const

```
ScrSeg : word = $B800;
```

```
procedure CalcScrSeg;assembler;
```

asm

```
call GetVideoMode
```

```
cmp al,7
```

```
jz @B000
```

```
jmp @Son
```

```
@B000: mov ax,0B000h
```

```
mov ScrSeg,ax
```

```
@Son:
```

```
end;
```

```
function CalcOfs(X,Y: byte):word;assembler;
```

```
asm
```

```
xor ax,ax
```

```
mov al,Y
```

```
dec al
```

```
mov bx,160
```

```
mul bx
```

```
xor bx,bx
```

```
mov bl,X
```

```
dec bl
```

```
shl bx,1
```

```
add ax,bx
```

```
end;
```

```
procedure Fill(X,Y,W,H,C,Atr :byte);assembler;
```

```
asm
```

```
mov al,X
```

```
push ax
```

```
mov al,Y
```

```
push ax
```

```
call CalcOfs
```

```
mov di,ax
```

```
mov ax,ScrSeg
```

```
mov es,ax
```

```
xor bx,bx
```

```
mov bh,H
```

```
inc bh
```

```
mov dx,di
```

```
@Loop: xor cx,cx
        mov cl,W
        inc cl
        push di
@Yatay: mov al,C
        mov es:[di],al
        mov al,Atr
        mov es:[di+1],al
        add di,2
        loop @Yatay
        pop di
        add di,160
        inc bl
        cmp bl,bh
        jne @Loop
end;
```

```
procedure SetVideoMode(Mode : byte);assembler;
asm
    mov ah,00h
    mov al,Mode
    int 10h
    call CalcScrSeg
end;
```

```
function GetVideoMode:byte;assembler;
asm
    mov ah,0fh
    int 10h
end;
```

```
procedure SetActivePage(PageNo : byte);assembler;
```

```
asm
    mov ah,05h
    mov al,PageNo
    int 10h
end;
```

```
function GetActivePage:byte;assembler;
```

```
asm
    mov ah,0fh
    int 10h
    mov al,bh
end;
```

```
procedure SetCursorMode(CrsrMode : word);assembler;
```

```
asm
    mov ah,01h
    mov cx,CrsrMode
    int 10h
end;
```

```
function GetCursorMode:word;assembler;
```

```
asm
    call GetActivePage
    mov bh,al
    mov ah,03h
    int 10h
    mov ax,cx
end;
```

```
procedure CursorOn;assembler;
```

```
asm
    call GetCursorMode
```

```
    and ah,10011111b { on }  
    push ax  
    call SetCursorMode  
end;
```

```
procedure CursorOff;assembler;
```

```
asm
```

```
    call GetCursorMode  
    and ah,10011111b { off }  
    or ah,00100000b  
    push ax  
    call SetCursorMode  
end;
```

```
procedure Locate(X,Y : byte);assembler;
```

```
asm
```

```
    call GetActivePage  
    mov bh,al  
    mov dh,Y  
    mov dl,X  
    dec dh  
    dec dl  
    mov ah,02h  
    int 10h  
end;
```

```
function GetCursorXPos:byte;assembler;
```

```
asm
```

```
    call GetActivePage  
    mov bh,al  
    mov ah,03h  
    int 10h
```

```
    mov al,dl
    inc al
end;
```

```
function GetCursorYPos:byte;assembler;
```

```
asm
```

```
    call GetActivePage
    mov bh,al
    mov ah,03h
    int 10h
    mov al,dh
    inc al
```

```
end;
```

```
procedure YazXY(X,Y : byte; Msg:string; Atr : byte);assembler;
```

```
asm
```

```
    mov al,X
    push ax
    mov al,Y
    push ax
    call CalcOfs
    mov si,ax
    push ds
    mov ax,ScrSeg
    mov ds,ax
    les di,Msg
    xor ax,ax
    mov al,es:[di]
    mov cx,ax
    inc di
```

```
@Yaz:  mov al,es:[di]
```

```
    mov ds:[si],al
```

```
    mov al,Atr
    mov ds:[si+1],al
    add si,2
    inc di
    loop @Yaz
    pop ds
end;
```

```
procedure EkraniSakla;assembler; { (Ekran : TEkran) }
```

```
asm
```

```
    push ds
    mov ax,ScrSeg
    mov ds,ax
    mov si,0
    mov cx,4096
    les di,Ekran
    cld
    rep movsb
    pop ds
end;
```

```
procedure EkraniGeriAl;assembler; { (var Ekran : TEkran) }
```

```
asm
```

```
    push ds
    mov ax,ScrSeg
    mov es,ax
    mov di,0
    mov cx,4096
    lds si,Ekran
    cld
    rep movsb
    pop ds
```

end;

function InColorMode:boolean;

begin

InColorMode := (ScrSeg = \$B800);

end;

function IsVga:boolean;assembler;

asm

mov ah,12h

mov al,00h

mov bl,36h

int 10h

mov ah,0

cmp al,12h

jnz @NotVga

inc ah

@NotVga:

end;

function IsRenkli:boolean;assembler;

asm

xor ax,ax

push ds

mov ds,ax

mov si,1040h

mov al,[si]

test al,00010000b

jz @True

xor ax,ax

jmp @Son

@True: mov ax,1



```
@Son : pop ds  
end;
```

```
procedure StatusLine(Msg : string);  
begin  
  Fill(1,25,79,0,32,Black+16*Cyan);  
  YazXY(3,25,Msg,Black+16*Cyan);  
end;
```

```
procedure GetFont(FontBuffer : array of byte);assembler;
```

```
asm
```

```
  call IsVga  
  and al,al  
  jz @Quit  
  push bp  
  les bp,FontBuffer  
  mov cx,0100h { 256 karakter }  
  mov bx,1000h  
  mov dx,0  
  mov ax,1100h  
  int 10h  
  pop bp
```

```
@Quit:
```

```
end;
```

```
procedure GetBiosFont;assembler;
```

```
asm
```

```
  call IsVga  
  and al,al  
  jz @Quit  
  push bp  
  mov ax,1130h
```

```
    mov bh,06h
    int 10h
    mov cx,0100h { 256 karakter }
    mov bx,1000h
    mov dx,0
    mov ax,1100h
    int 10h
    pop bp
@Quit:
end;

procedure UseFontYenix;
begin
    if IsVga then GetFont(FontYenix);
end;

procedure InitWindow(X, Y, W, H, Atr : byte);
begin
    Fill(X, Y, W, H, 32, Atr);
end;

procedure MasaYap(Atr : byte);
begin
    Fill(1, 2, 79, 23, 178, Atr);
end;

procedure GolgeYap(X, Y, W, H : byte); assembler;
asm
    xor ax, ax
    mov al, X
    add al, 2
    push ax
```

```
xor ax,ax
mov al,Y
add al,H
inc al
push ax
call CalcOfs
inc ax
mov di,ax
mov ax,ScrSeg
mov es,ax
xor cx,cx
mov cl,W
inc cx
@Alt: mov al,00001000b { Golgeleme katsayisi }
      mov es:[di],al
      add di,2
      loop @Alt
{ --- }
mov al,X
add al,W
inc al
push ax
mov al,Y
inc al
push ax
call CalcOfs
inc ax
mov di,ax
mov ax,ScrSeg
mov es,ax
xor cx,cx
mov cl,H
```

```

@Yan: push di
      mov al,00001000b { Golgeleme katsayisi }
      mov es:[di],al
      add di,2
      mov es:[di],al
      add di,2
      pop di
      add di,160
      loop @Yan
end;

```

```

procedure KutuCiz(X,Y,W,H,Tip,Atr: byte);

```

```

const

```

```

  Frame : array[1..2,1..6] of char =
    ( ('Ú', '¿', 'À', 'Ù', 'Ä', '³'),
      ('É', '»', 'È', '¼', 'Í', 'º') );

```

```

var

```

```

  I : byte;

```

```

begin

```

```

  YazXY(X,Y,Frame[Tip,1],Atr);

```

```

  YazXY(X+W,Y,Frame[Tip,2],Atr);

```

```

  YazXY(X,Y+H,Frame[Tip,3],Atr);

```

```

  YazXY(X+W,Y+H,Frame[Tip,4],Atr);

```

```

  Fill(X+1,Y,W-2,0,byte(Frame[Tip,5]),Atr);

```

```

  Fill(X+1,Y+H,W-2,0,byte(Frame[Tip,5]),Atr);

```

```

  for I:=Y+1 to Y+H-1 do begin

```

```

    YazXY(X,I,Frame[Tip,6],Atr);

```

```

    YazXY(X+W,I,Frame[Tip,6],Atr);

```

```

  end;

```

```

end;

```

```

procedure KutuYap(X,Y,W,H,Tip,Atr : byte);

```

```

begin
  InitWindow(X,Y,W,H,Atr);
  KutuCiz(X,Y,W,H,Tip,Atr);
  GolgeYap(X,Y,W,H);
end;

procedure HataMesaji(Msg:string);
const
  YPos : byte = 8;
var
  Ekran : TEkran;
  Ch   : char;
  Atr  : byte;
  S    : string;
  Len  : byte;
begin
  EkraniSakla(Ekran);
  StatusLine('Devam etmek için ENTER"e basın. ');
  if IsRenkli then Atr := 79 else Atr := Ters;
  Len := Length(Msg);
  if Len < 40 then Len := 40;
  KutuYap((70-Len) div 2, YPos, Len+10, 6, Ince, Atr);
  if IsRenkli then Atr := 79 else Atr := Ters;
  YazXY(((70-Len) div 2)+5, YPos+2, Msg, Atr);
  FillChar(S, SizeOf(S), #196);
  S[0] := Chr(Len+11);
  S[1] := #195;
  S[Len+11] := #180;
  YazXY((70-Len) div 2, YPos+4, S, Atr);
  if IsRenkli then Atr := Yellow+16*Red else Atr := Ters;
  YazXY(((70-Len) div 2)+5, YPos+5, 'Enter:Tamam', Atr);
  if IsRenkli then Atr := 79 else Atr := Ters;

```

```
YazXY(36,YPos,' Hata ',Atr);
```

```
repeat
```

```
  Ch := ReadKey;
```

```
until Ch = #13;
```

```
EkraniGeriAl(Ekran);
```

```
end;
```

```
function Tasdiklendi(Msg : string):boolean;
```

```
const
```

```
  YPos : byte = 8;
```

```
var
```

```
  Ekran : TEkran;
```

```
  Ch   : char;
```

```
  Atr  : byte;
```

```
  S    : string;
```

```
  Len  : byte;
```

```
begin
```

```
  Tasdiklendi := False;
```

```
  EkraniSakla(Ekran);
```

```
  StatusLine('Kullanıcının kararı bekleniyor...');
```

```
  if IsRenkli then Atr := 79 else Atr := Ters;
```

```
  Len := Length(Msg);
```

```
  if Len < 40 then Len := 40;
```

```
  KutuYap((70-Len) div 2,YPos,Len+10,6,Ince,Atr);
```

```
  if IsRenkli then Atr := 79 else Atr := Ters;
```

```
  YazXY(((70-Len) div 2)+5,YPos+2,Msg,Atr);
```

```
  FillChar(S,SizeOf(S),#196);
```

```
  S[0] := Chr(Len+11);
```

```
  S[1] := #195;
```

```
  S[Len+11] := #180;
```

```
  YazXY((70-Len) div 2,YPos+4,S,Atr);
```

```
  if IsRenkli then Atr := Yellow+16*Red else Atr := Ters;
```

```

YazXY(((70-Len) div 2)+5, YPos+5, 'Enter:Tamam', Atr);
YazXY(((70-Len) div 2)+Len-5, YPos+5, 'Esc:Vazgeç', Atr);
if IsRenkli then Atr := 79 else Atr := Ters;
YazXY(36, YPos, ' Onay ', Atr);
repeat
  Ch := ReadKey;
until Ch in [#27, #13];
if Ch = #13 then Tasdiklendi := True;
EkraniGeriAl(Ekran);
end;

```

```

procedure ClearUstLine;
begin
  Fill(1,1,79,1,32, White+16*Cyan);
end;

```

```

procedure EkraniAc; assembler;
asm
  cli
  mov dx, 3c0h
  mov al, 20h { Ac }
  out dx, al
  sti
end;

```

```

procedure EkraniKapa; assembler;
asm
  cli
  mov dx, 3c0h
  mov al, 00h { Kapa }
  out dx, al
  sti

```

end;

begin

  CalcScrSeg;

end.





### 6.3. Keyboard.pas

```
unit KeyBoard;
```

```
interface
```

```
const
```

```
    kbF1      = #59;
```

```
    kbF2      = #60;
```

```
    kbF3      = #61;
```

```
    kbF4      = #62;
```

```
    kbF5      = #63;
```

```
    kbF6      = #64;
```

```
    kbF7      = #65;
```

```
    kbF8      = #66;
```

```
    kbF9      = #67;
```

```
    kbF10     = #68;
```

```
    kbCtrlF1  = #94;
```

```
    kbCtrlF2  = #95;
```

```
    kbCtrlF3  = #96;
```

```
    kbCtrlF4  = #97;
```

```
    kbCtrlF5  = #98;
```

```
    kbCtrlF6  = #99;
```

```
    kbCtrlF7  = #100;
```

```
    kbCtrlF8  = #101;
```

```
    kbCtrlF9  = #102;
```

```
    kbCtrlF10 = #103;
```

```
    kbShiftF1 = #84;
```

```
    kbShiftF2 = #85;
```

```
    kbShiftF3 = #86;
```

```
    kbShiftF4 = #87;
```

kbShiftF5 = #88;  
kbShiftF6 = #89;  
kbShiftF7 = #90;  
kbShiftF8 = #91;  
kbShiftF9 = #92;  
kbShiftF10 = #93;

kbAltF1 = #104;  
kbAltF2 = #105;  
kbAltF3 = #106;  
kbAltF4 = #107;  
kbAltF5 = #108;  
kbAltF6 = #109;  
kbAltF7 = #110;  
kbAltF8 = #111;  
kbAltF9 = #112;  
kbAltF10 = #113;

kbAlt1 = #120;  
kbAlt2 = #121;  
kbAlt3 = #122;  
kbAlt4 = #123;  
kbAlt5 = #124;  
kbAlt6 = #125;  
kbAlt7 = #126;  
kbAlt8 = #127;  
kbAlt9 = #128;  
kbAlt0 = #129;

kbAltA = #30;  
kbAltB = #48;  
kbAltC = #46;

kbAltD = #32;  
kbAltE = #18;  
kbAltF = #33;  
kbAltG = #34;  
kbAltH = #35;  
kbAltI = #23;  
kbAltJ = #36;  
kbAltK = #37;  
kbAltL = #38;  
kbAltM = #50;  
kbAltN = #49;  
kbAltO = #24;  
kbAltP = #25;  
kbAltQ = #16;  
kbAltR = #19;  
kbAltS = #31;  
kbAltT = #20;  
kbAltU = #22;  
kbAltW = #17;  
kbAltX = #45;  
kbAltV = #47;  
kbAltY = #21;  
kbAltZ = #44;

kbEnter = #13; { 1 byte }  
kbEsc = #27; { 1 byte }  
kbSpace = #32; { 1 byte }  
kbBkSpace = #8; { 1 byte }  
kbTab = #9; { 1 byte }  
kbCtrlEnter = #10; { 1 byte }  
kbUp = #72;  
kbDown = #80;

```

kbLeft    = #75;
kbRight   = #77;
kbHome    = #71;
kbEnd     = #79;
kbPgUp    = #73;
kbPgDn    = #81;
kbCtrlLeft = #115;
kbCtrlRight = #116;
kbCtrlHome = #119;
kbCtrlEnd  = #117;
kbCtrlPgUp = #132;
kbCtrlPgDn = #118;
kbIns     = #82;
kbDel     = #83;
kbShiftIns = #5;
kbShiftDel = #7;
kbCtrlIns  = #4;
kbCtrlDel  = #6;

```

```

function OzelChar(var Kod:char):boolean;

```

```

implementation

```

```

uses Crt;

```

```

function OzelChar(var Kod:char):boolean;

```

```

var

```

```

    Ch:Char;

```

```

begin

```

```

    OzelChar:=False;

```

```

    Ch:=ReadKey;

```

```

    if Ch=#0 then begin

```

```
Ch:=ReadKey;  
OzelChar:=True;  
end;  
Kod:=Ch;  
end;  
  
end.
```



## 6.4. TypesLP.pas

```
unit TypesLP;
```

```
interface
```

```
type
```

```
  TSetup = record
```

```
    xGecik,yGecik   : word;
```

```
    xRampSay,yRampSay : word;
```

```
    xRampKat,yRampKat : word;
```

```
    BaseAdr        : word
```

```
  end;
```

```
const
```

```
  Setup : TSetup = (xGecik   : 2500;
```

```
    yGecik   : 2500;
```

```
    xRampSay : 10;
```

```
    yRampSay : 10;
```

```
    xRampKat : 100;
```

```
    yRampKat : 100;
```

```
    BaseAdr  : $378);
```

```
  SetupFName : string[10] = 'LP.Ini';
```

```
function LoadSetup:boolean;
```

```
function SaveSetup:boolean;
```

```
implementation
```

```
function LoadSetup:boolean;
```

```
var
```

```
  F      : file;
```

```
    NumRead : word;
begin
  LoadSetup := False;
  Assign(F,SetupFName);
  {$I-}
  Reset(F,1);
  if IOResult <> 0 then Exit;
  BlockRead(F,Setup,SizeOf(Setup),NumRead);
  Close(F);
  {$I+}
  LoadSetup := True;
end;
```

```
function SaveSetup:boolean;
var
  F      : file;
  NumWrite : word;
begin
  SaveSetup := False;
  Assign(F,SetupFName);
  {$I-}
  ReWrite(F,1);
  if IOResult <> 0 then Exit;
  BlockWrite(F,Setup,SizeOf(Setup),NumWrite);
  Close(F);
  {$I+}
  SaveSetup := True;
end;

end.
```

## 6.5. StpProcs.pas

```
unit StpProcs;
```

```
interface
```

```
procedure RunSetup;
```

```
implementation
```

```
uses Crt,Screen,KeyBoard,TypesLP,StrOp,IO;
```

```
const
```

```
  Mx = 32;
```

```
  My = 4;
```

```
  XPos = Mx;
```

```
  YPos = My;
```

```
  MenuSay = 2;
```

```
  Menu : array[1..MenuSay] of string[18] =
```

```
    (' Rampa ayarları ',
```

```
     ' Port ayarı    ');
```

```
  MSayac : byte = 1;
```

```
procedure RampSetup;
```

```
{ }const
```

```
{ }  MaxItem = 6;
```

```
{ }procedure DrawWin;
```

```
{ }var
```

```
{ }  Atr : byte;
```

```
{ }begin
```

```
{ }  Atr := White+16*Blue;
```

```
{ }  KutuYap(XPos,YPos,43,7,Ince,Atr);
```



```

{} Atr := LightGray+16*Blue;
{} YazXY(XPos+13,YPos,'RAMPA AYARLARI',Atr);
{} StatusLine(#24#25':Yönlendir #17#196#217':Seç Esc:Ayarlar Menüü');
{}end;

{}procedure WriteValues;
{}const
{} Atr : word = LightGray+16*Blue;
{}begin
{} YazXY(XPos+1,YPos+1,' 1.Motor kalkış katsayısı '+StrLF(Setup.xGecik,4)+' tick
',Atr);
{} YazXY(XPos+1,YPos+2,' 1.Motor rampa sayısı '+StrLF(Setup.xRampSay,4)+'
tick ',Atr);
{} YazXY(XPos+1,YPos+3,' 1.Motor rampa katsayısı '+StrLF(Setup.xRampKat,4)+'
tick ',Atr);
{} YazXY(XPos+1,YPos+4,' 2.Motor kalkış katsayısı '+StrLF(Setup.yGecik,4)+' tick
',Atr);
{} YazXY(XPos+1,YPos+5,' 2.Motor rampa sayısı '+StrLF(Setup.yRampSay,4)+'
tick ',Atr);
{} YazXY(XPos+1,YPos+6,' 2.Motor rampa katsayısı '+StrLF(Setup.yRampKat,4)+'
tick ',Atr);
{}end;

{}procedure MarkSelected(Selector: byte);
{}const
{} Atr : word = White+16*Red;
{}begin
{} case Selector of
{} 1 : YazXY(XPos+1,YPos+1,' 1.Motor kalkış katsayısı '+StrLF(Setup.xGecik,4)+'
tick ',Atr);
{} 2 : YazXY(XPos+1,YPos+2,' 1.Motor rampa sayısı
'+StrLF(Setup.xRampSay,4)+' tick ',Atr);

```

```

{} 3 : YazXY(XPos+1,YPos+3,' 1.Motor rampa katsayısı
'+StrLF(Setup.xRampKat,4)+' tick ',Atr);
{} 4 : YazXY(XPos+1,YPos+4,' 2.Motor kalkış katsayısı    '+StrLF(Setup.yGecik,4)+'
tick ',Atr);
{} 5 : YazXY(XPos+1,YPos+5,' 2.Motor rampa sayısı
'+StrLF(Setup.yRampSay,4)+' tick ',Atr);
{} 6 : YazXY(XPos+1,YPos+6,' 2.Motor rampa katsayısı
'+StrLF(Setup.yRampKat,4)+' tick ',Atr);
{} end;
{}end;

{}procedure GetSelectedValue(Selector: byte);
{}const
{}  Atr : word = White+16*Red;
{}var
{} Ekran : TEkran;
{}begin
{} EkraniSakla(Ekran);
{} CursorOn;
{} StatusLine('Değer girişi...');
{} case Selector of
{} 1 : Setup.xGecik := ReadWord(Setup.xGecik,4,XPos+33,YPos+1,Atr);
{} 2 : Setup.xRampSay := ReadWord(Setup.xRampSay,4,XPos+33,YPos+2,Atr);
{} 3 : Setup.xRampKat := ReadWord(Setup.xRampKat,4,XPos+33,YPos+3,Atr);
{} 4 : Setup.yGecik := ReadWord(Setup.yGecik,4,XPos+33,YPos+4,Atr);
{} 5 : Setup.yRampSay := ReadWord(Setup.yRampSay,4,XPos+33,YPos+5,Atr);
{} 6 : Setup.yRampKat := ReadWord(Setup.yRampKat,4,XPos+33,YPos+6,Atr);
{} end;
{} EkraniGeriAl(Ekran);
{} CursorOff;
{}end;

```

```

var
  Ekran  : TEkran;
  Ch     : char;
  Selector : byte;
begin
  EkraniSakla(Ekran);
  DrawWin;
  WriteValues;
  Selector := 1;
  MarkSelected(Selector);
repeat
  Ch := ReadKey;
  case Ch of
    kbEsc : begin
      EkraniGeriAl(Ekran);
      Exit;
      end;
    kbEnter: begin
      WriteValues;
      GetSelectedValue(Selector);
      MarkSelected(Selector);
      end;
    #0    : begin
      Ch := ReadKey;
      case Ch of
        kbUp,kbLeft  : if Selector > 1 then Dec(Selector);
        kbDown,kbRight : if Selector < MaxItem then Inc(Selector);
      end;
      end;
  end;
  WriteValues;
  MarkSelected(Selector);

```

```
until False;
end;
```

```
{-----}
```

```
procedure PortSetup;
```

```
{}const
```

```
{} MaxItem = 3;
```

```
{}procedure DrawWin;
```

```
{}var
```

```
{} Atr : byte;
```

```
{}begin
```

```
{} Atr := White+16*Blue;
```

```
{} KutuYap(XPos,YPos,29,6,Ince,Atr);
```

```
{} Atr := LightGray+16*Blue;
```

```
{} YazXY(XPos+9,YPos,'PORT AYARI',Atr);
```

```
{} StatusLine(#24#25':Yönlendir '#17#196#217':Seç Esc:Ayarlar Menüsü');
```

```
{}end;
```

```
{}procedure WriteValues;
```

```
{}const
```

```
{} Atr : word = LightGray+16*Blue;
```

```
{}begin
```

```
{} YazXY(XPos+1,YPos+1,' Printer Port at $0378 ',Atr);
```

```
{} YazXY(XPos+1,YPos+2,' Printer Port at $0278 ',Atr);
```

```
{} YazXY(XPos+1,YPos+3,' Printer Port at $03BC ',Atr);
```

```
{} YazXY(XPos+1,YPos+5,' Seçili address: $'+HexWord(Setup.BaseAdr),Atr);
```

```
{}end;
```

```
{}procedure MarkSelected(Selector: byte);
```

```
{}const
```

```

{}  Atr : word = White+16*Red;
{}begin
{} case Selector of
{} 1 : YazXY(XPos+1,YPos+1,' Printer Port at $0378 ',Atr);
{} 2 : YazXY(XPos+1,YPos+2,' Printer Port at $0278 ',Atr);
{} 3 : YazXY(XPos+1,YPos+3,' Printer Port at $03BC ',Atr);
{} end;
{}end;

```

var

Ekran : TEkran;

Ch : char;

Selector : byte;

begin

EkraniSakla(Ekran);

DrawWin;

WriteValues;

Selector := 1;

MarkSelected(Selector);

repeat

Ch := ReadKey;

case Ch of

kbEsc : begin

EkraniGeriAl(Ekran);

Exit;

end;

kbEnter: begin

case Selector of

1: Setup.BaseAdr := \$378;

2: Setup.BaseAdr := \$278;

3: Setup.BaseAdr := \$3BC;

end;

```

    WriteValues;
    MarkSelected(Selector);
end;
#0 : begin
    Ch := ReadKey;
    case Ch of
        kbUp, kbLeft : if Selector > 1 then Dec(Selector);
        kbDown, kbRight : if Selector < MaxItem then Inc(Selector);
    end;
end;
end;
WriteValues;
MarkSelected(Selector);
until False;
end;
{ ----- }

procedure DrawStpMenu;
const
    MenuName : string[8] = 'AYARLAR';
var
    Atr : byte;
    I : byte;
begin
    Atr := White+16*Blue;
    KutuYap(Mx, My, SizeOf(Menu[1]), MenuSay+1, Ince, Atr);
    Atr := LightGray+16*Blue;
    YazXY(Mx+((SizeOf(Menu[1])-Length(MenuName)) div 2), My, MenuName, Atr);
    for I := 1 to MenuSay do YazXY(Mx+1, My+I, Menu[I], Atr);
    Atr := White+16*Red;
    YazXY(Mx+1, My+MSayac, Menu[MSayac], Atr);

```

```
StatusLine(#24#25':Yönlendir '#17#196#217':Seç Esc:Ana Menü');
end;
```

```
function MenuAl(var Secim: byte):boolean;
var
  Ch : char;
begin
  MenuAl := False;
  YazXY(Mx+1,My+MSayac,Menu[MSayac],White+16*Red);
  repeat
    if OzelChar(Ch) then
      begin
        YazXY(Mx+1,My+MSayac,Menu[MSayac],LightGray+16*Blue);
        case Ch of
          kbUp, kbLeft : begin
            Dec(MSayac);
            if MSayac <= 0 then MSayac := MenuSay;
            end;
          kbDown, kbRight : begin
            Inc(MSayac);
            if MSayac > MenuSay then MSayac := 1;
            end;
        end;
        YazXY(Mx+1,My+MSayac,Menu[MSayac],White+16*Red);
      end;
    until Ch in [kbEnter,kbEsc];
    YazXY(Mx+1,My+MSayac,Menu[MSayac],LightGray+16*Blue);
    Secim := MSayac;
    if Ch = kbEsc then Exit;
    MenuAl := True;
  end;
```

```
procedure RunSetup;
var
  Ekran : TEkran;
  Secim : byte;
begin
  EkraniSakla(Ekran);
  DrawStpMenu;
  repeat
    if not MenuAl(Secim) then
      begin
        if not SaveSetup then HataMesaji('Program ayarları kaydedilemiyor...');
        EkraniGeriAl(Ekran);
        Exit;
      end;
    case Secim of
      1 : RampSetup;
      2 : PortSetup;
    end;
  until False;
end;

end.
```



## 6.6. MCtrl.pas

```
unit MCtrl;

interface

procedure RunManualCtrl;

implementation

uses Crt,Screen,KeyBoard,TypesLP,StrOp,IO;

const
    SMArray : array[1..8] of byte = (1,3,2,6,4,12,8,9);
    XI : byte = 1;
    YI : byte = 1;

procedure NOP;
var
    I : word;
begin
    for I := 0 to 250 do;
end;

procedure ResetTimer;
begin
    Port[$43] := $34;
    NOP;
    Port[$40] := 0;
    NOP;
    Port[$40] := 0;
    NOP;
end;
```

```

function ReadTimer:word;assembler;
asm
  cli
  mov dx,$40    { Data port for timer }
  mov al,dh     { 0 to latch counter 0 }
  out $43,al   { Latch timer }
  jmp @0       { Null jump }
@0: in  al,dx   { Timer chip LSB }
  jmp @1       { Null jump }
@1: mov cl,al   { Save in CL }
  in  al,dx    { Timer chip MSB }
  mov ah,al
  mov al,cl
  not ax
  sti
end;

```

```

function Sur(X,Y: integer):longint;

```

```

var

```

```

  OrjX,OrjY    : word;
  XTimer,YTimer : longint;
  XGecik,YGecik : word;
  Time        : longint;
  T           : record
    W1,W2 : word;
  end absolute Time;
  A        : word;
  XArti,YArti : boolean;
  OldIMR   : byte;

```

```

begin

```

```

  XArti := X < 0;

```

```

YArti := Y < 0;
X := Abs(X);
Y := Abs(Y);
{ ----- }
ResetTimer;
OldIMR := Port[$21];
Port[$21] := $FF;
OrjX := X;
OrjY := Y;
XTimer := 0;
YTimer := 0;
XGecik := Setup.xGecik;
YGecik := Setup.yGecik;
Time := 0;
repeat
  A := ReadTimer;

  if T.W1 > A then Inc(T.W2);

  T.W1 := A;

  if (X = 0) and (Y = 0) then Break;

  if X <> 0 then
    if Time - XTimer > XGecik then
      begin
        if XArti then
          begin
            Inc(XI);
            if XI > 8 then XI := 1;
          end
        else

```

```

begin
  Dec(XI);
  if XI < 1 then XI := 8;
end;
Port[Setup.BaseAdr] := SArray[XI] or (SArray[YI] shl 4);
XTimer := Time;
Dec(X);
if OrjX > Setup.xRampSay then
  begin
    if X < Setup.xRampSay then Inc(XGecik,Setup.xRampKat);
    if X > (OrjX - Setup.xRampSay) then Dec(XGecik,Setup.xRampKat);
  end;
end;

if Y <> 0 then
  if Time - YTimer > YGecik then
    begin
      if YArti then
        begin
          Inc(YI);
          if YI > 8 then YI := 1;
        end
      else
        begin
          Dec(YI);
          if YI < 1 then YI := 8;
        end;
      Port[Setup.BaseAdr] := SArray[XI] or (SArray[YI] shl 4);
      YTimer := Time;
      Dec(Y);
      if OrjY > Setup.yRampSay then
        begin

```

```

    if Y < Setup.yRampSay then Inc(YGecik,Setup.yRampKat);
    if Y > (OrjY - Setup.yRampSay) then Dec(YGecik,Setup.yRampKat);
    end;
end;

until False;
Port[$21] := OldIMR;
Sur := Time;
end;

procedure PortBosalt;
begin
    Port[Setup.BaseAdr] := 0;
end;

procedure PortTut;
begin
    Port[Setup.BaseAdr] := SArray[XI] or (SArray[YI] shl 4);
end;

{ ----- }

procedure RunAdimCtrl;
const
    XPos : byte = 16;
    YPos : byte = 9;
    MaxItem = 3;
    X : integer = 0;
    Y : integer = 0;

{}procedure DrawWin;
{}var

```

```

{} Atr : byte;
{}begin
{} Atr := White+16*Blue;
{} KutuYap(XPos, YPos, 43, 5, Ince, Atr);
{} Atr := LightGray+16*Blue;
{} YazXY(XPos+13, YPos, 'ADIM SAYISI GİRİŞİ', Atr);
{} StatusLine(#24#25':Yönlendir '#17#196#217':Seç Esc:Kontrol Menüü');
{}end;

```

```

{}procedure WriteValues;
{}const
{}  Atr : word = LightGray+16*Blue;
{}begin
{} YazXY(XPos+1, YPos+1, ' 1.Motorun atacağı adım sayısı: '+StrLF(X,4), Atr);
{} YazXY(XPos+1, YPos+2, ' 2.Motorun atacağı adım sayısı: '+StrLF(Y,4), Atr);
{} YazXY(XPos+1, YPos+4, ' Konumlan ', Atr);
{}end;

```

```

{}procedure MarkSelected(Selector: byte);
{}const
{}  Atr : word = White+16*Red;
{}begin
{} case Selector of
{} 1 : YazXY(XPos+1, YPos+1, ' 1.Motorun atacağı adım sayısı: '+StrLF(X,4), Atr);
{} 2 : YazXY(XPos+1, YPos+2, ' 2.Motorun atacağı adım sayısı: '+StrLF(Y,4), Atr);
{} 3 : YazXY(XPos+1, YPos+4, ' Konumlan ', Atr);
{} end;
{}end;

```

```

{}procedure GetSelectedValue(Selector: byte);
{}const
{}  Atr : word = White+16*Red;

```

```

{}var
{} Ekran : TEkran;
{} XStr,YStr : string;
{} Code   : integer;
{}begin
{} EkraniSakla(Ekran);
{} CursorOn;
{} StatusLine('Değer girişi...');
{} case Selector of
{} 1 : X := ReadWord(X,4,XPos+38,YPos+1,Atr);
{} 2 : Y := ReadWord(Y,4,XPos+38,YPos+2,Atr);
{} end;
{} EkraniGeriAl(Ekran);
{} CursorOff;
{}end;

```

```

var
  Ekran   : TEkran;
  Ch      : char;
  Selector : byte;
begin
  EkraniSakla(Ekran);
  DrawWin;
  WriteValues;
  Selector := 1;
  MarkSelected(Selector);
  repeat
    Ch := ReadKey;
    case Ch of
      kbEsc : begin
        EkraniGeriAl(Ekran);
        Exit;

```

```

    end;
kbEnter: begin
    WriteValues;
    if Selector in [1..2] then GetSelectedValue(Selector) else Sur(X, Y);
    MarkSelected(Selector);
    end;
#0 : begin
    Ch := ReadKey;
    case Ch of
        kbUp, kbLeft : if Selector > 1 then Dec(Selector);
        kbDown, kbRight : if Selector < MaxItem then Inc(Selector);
    end;
    end;
end;
WriteValues;
MarkSelected(Selector);
until False;
end;
{ ----- }

```

```

procedure RunTusCtrl;
const
    XPos : byte = 15;
    YPos : byte = 6;
var
    Ekran : TEkran;
    Ch : char;
    Atr : byte;
begin
    EkraniSakla(Ekran);
    StatusLine('Esc:Kontrol Menus');

```



```

KutuYap(XPos,YPos,40,9,Ince,White+16*Blue);
Atr := LightGray+16*Blue;
YazXY(XPos+10,YPos,'OK TUŞLARIYLA KONTROL',Atr);

YazXY(XPos+2,YPos+1,['Sağ ok ]   sağa yavaş',Atr);
YazXY(XPos+2,YPos+2,['Sol ok  ]   sola yavaş',Atr);
YazXY(XPos+2,YPos+3,['Yukarı ok]   yukarı yavaş',Atr);
YazXY(XPos+2,YPos+4,['Aşağı ok ]   aşağı yavaş',Atr);
YazXY(XPos+2,YPos+5,['Home   ]   sağa hızlı',Atr);
YazXY(XPos+2,YPos+6,['End    ]   sola hızlı',Atr);
YazXY(XPos+2,YPos+7,['Page Up ]   yukarı hızlı',Atr);
YazXY(XPos+2,YPos+8,['Page Dn ]   aşağı hızlı',Atr);

repeat
if KeyPressed then
begin
Ch := ReadKey;
case Ch of
kbEsc : begin
EkraniGeriAl(Ekran);
Break;
end;
#0 : begin
Ch := ReadKey;
case Ch of
kbUp   : Sur(1,0);
kbDown : Sur(-1,0);
kbLeft : Sur(0,1);
kbRight: Sur(0,-1);
kbHome : Sur(0,20);
kbEnd  : Sur(0,-20);
kbPgUp : Sur(20,0);
kbPgDn : Sur(-20,0);

```

```

        end;
    end;
end;
end;
until False;
end;

```

```

{ ----- }

```

```

const

```

```

    Mx = 32;
    My = 4;
    MenuSay = 2;
    Menu : array[1..MenuSay] of string[25] =
        (' Ok tuşlarıyla kontrol ',
        ' Adım sayısı girişi ');
    MSayac : byte = 1;

```

```

procedure DrawCtrlMenu;

```

```

const

```

```

    MenuName : string[8] = 'KONTROL';
    XPos = Mx;
    YPos = My;

```

```

var

```

```

    Atr : byte;
    I : byte;

```

```

begin

```

```

    Atr := White+16*Blue;
    KutuYap(Mx,My,SizeOf(Menu[1]),MenuSay+1,Ince,Atr);
    Atr := LightGray+16*Blue;
    YazXY(Mx+((SizeOf(Menu[1])-Length(MenuName)) div 2),My,MenuName,Atr);
    for I := 1 to MenuSay do YazXY(Mx+1,My+I,Menu[I],Atr);

```

```

Atr := White+16*Red;
YazXY(Mx+1,My+MSayac,Menu[MSayac],Atr);
StatusLine(#24#25':Yönlendir '#17#196#217':Seç Esc:Ana Menü');
end;

```

```

function MenuAl(var Secim: byte):boolean;
var
  Ch : char;
begin
  MenuAl := False;
  YazXY(Mx+1,My+MSayac,Menu[MSayac],White+16*Red);
  repeat
    if OzelChar(Ch) then
      begin
        YazXY(Mx+1,My+MSayac,Menu[MSayac],LightGray+16*Blue);
        case Ch of
          kbUp, kbLeft  : begin
              Dec(MSayac);
              if MSayac <= 0 then MSayac := MenuSay;
            end;
          kbDown, kbRight : begin
              Inc(MSayac);
              if MSayac > MenuSay then MSayac := 1;
            end;
        end;
        YazXY(Mx+1,My+MSayac,Menu[MSayac],White+16*Red);
      end;
    until Ch in [kbEnter,kbEsc];
    YazXY(Mx+1,My+MSayac,Menu[MSayac],LightGray+16*Blue);
    Secim := MSayac;
    if Ch = kbEsc then Exit;
  MenuAl := True;

```

```
end;

procedure RunManualCtrl;
var
  Ekran : TEkran;
  Secim : byte;
begin
  EkraniSakla(Ekran);
  DrawCtrlMenu;
  repeat
    if not MenuAl(Secim) then
      begin
        EkraniGeriAl(Ekran);
        Exit;
      end;
    case Secim of
      1 : RunTusCtrl;
      2 : RunAdimCtrl;
    end;
  until False;
end;

end.
```

## 6.7. StrOp.pas

```
unit StrOp;
```

```
interface
```

```
function IntToStr(I : longInt):string;
function HexByte(Sayi : byte):string;
function HexWord(Sayi : word):string;
function HexLongInt(Sayi : longint):string;
function BinByte(Sayi: byte):string;
function IsHarf(C : char):boolean;
function IsKucukHarf(C : char):boolean;
function IsBuyukHarf(C : char):boolean;
function IsRakam(C : char):boolean;
function StrL (L: longint): string;
function StrLF (L: longint; Field: integer): string;
function StrR (R: real): string;
function StrRF (R: real; Field: integer): string;
function StrRFD (R: real; Field,Decimals: integer): string;
function Sifirli(L:longint;Sifir:byte):string;
```

```
implementation
```

```
function IntToStr(I : longint):string;
var
  S : string;
begin
  Str(I, S);
  IntToStr := S;
end;

function HexByte(Sayi : byte):string;
```

```

const
  HexRakam : array[0..15] of char =
    '0123456789ABCDEF';
begin
  HexByte := HexRakam[Sayi shr 4] + HexRakam[Sayi and 15];
end;

function HexWord(Sayi : word):string;
begin
  HexWord := HexByte(Sayi shr 8) + HexByte(Sayi and 255);
end;

function HexLongInt(Sayi : longint):string;
begin
  HexLongInt := HexWord(Sayi shr 16) + HexWord(Sayi and 65535);
end;

function BinByte(Sayi: byte):string;
var
  Temp : string[8];
  I : byte;
begin
  FillChar(Temp,SizeOf(Temp),'0');
  Temp[0] := #8;
  if (Sayi and 1) <> 0 then Temp[8] := '1';
  if (Sayi and 2) <> 0 then Temp[7] := '1';
  if (Sayi and 4) <> 0 then Temp[6] := '1';
  if (Sayi and 8) <> 0 then Temp[5] := '1';
  if (Sayi and 16) <> 0 then Temp[4] := '1';
  if (Sayi and 32) <> 0 then Temp[3] := '1';
  if (Sayi and 64) <> 0 then Temp[2] := '1';
  if (Sayi and 128) <> 0 then Temp[1] := '1';

```

```
BinByte := Temp;
end;
```

```
function IsKucukHarf(C : char):boolean;
begin
  if Byte(C) in [97..122] then IsKucukHarf := True
  else IsKucukHarf := False;
end;
```

```
function IsBuyukHarf(C : char):boolean;
begin
  if Byte(C) in [65..90] then IsBuyukHarf := True
  else IsBuyukHarf := False;
end;
```

```
function IsRakam(C : char):boolean;
begin
  if Byte(C) in [48..57] then IsRakam := True
  else IsRakam := False;
end;
```

```
function IsHarf(C : char):boolean;
begin
  if ( IsKucukHarf(C) or IsBuyukHarf(C) ) then
    IsHarf := True else IsHarf := False;
end;
```

```
function StrL; { (L: longint): string; }
var Result: ^string;
begin
  Inline(      { Typical code: }
    $89/$EC/   { mov sp,bp   ; Drop Result }
  )
```

```

$16/      { push ss      ; Result segment }
$FF/$76/$0A);      { push [bp+$0A] ; Result offset ($08 for near) }
str (L,Result^);
end;

```

```

function StrLF; { (L: longint; Field: integer): string; }
var Result: ^string;
begin
  Inline ($89/$EC/$16/$FF/$76/$0C);
  str (L:Field,Result^);
end;

```

```

function StrR; { (R: real): string; }
var Result: ^string;
begin
  Inline ($89/$EC/$16/$FF/$76/$0C);
  str (R,Result^);
end;

```

```

function StrRF; { (R: real; Field: integer): string; }
var Result: ^string;
begin
  Inline ($89/$EC/$16/$FF/$76/$0E);
  str (R:Field,Result^);
end;

```


```

function StrRFD; { (R: real; Field,Decimals: integer): string; }
var Result: ^string;
begin
  Inline ($89/$EC/$16/$FF/$76/$10);
  str (R:Field:Decimals,Result^);
end;

```



```
function Sifirli; { (L:longint;Sifir:byte):string; }  
var  
  S : string;  
  S1 : string;  
begin  
  FillChar(S,Sifir+1,'0');  
  S1 := IntToStr(L);  
  Insert(S1,S,Sifir-Byte(S1[0])+1);  
  S[0] := Chr(Sifir);  
  Sifirli := S;  
end;  
  
end.
```



## 6.8. IO.pas

```
unit IO;
```

```
interface
```

```
function ReadUpStr(Prompt: string; Len,XPos,YPos,Atr: byte):string;
```

```
function ReadNumStr(Prompt: string; Len, XPos, YPos, Atr : byte):string;
```

```
function ReadWord(Prompt: word; Len,XPos,YPos,Atr : byte):word;
```

```
implementation
```

```
uses Crt,Screen,KeyBoard,StrOp;
```

```
function ReadNumStr(Prompt : string; Len, XPos, YPos, Atr : byte):string;
```

```
const
```

```
    F : char = '±'; {#177}
```

```
var
```

```
    Temp : string;
```

```
    X : byte;
```

```
    Ch : char;
```

```
    I : byte;
```

```
procedure Display;
```

```
begin
```

```
    Temp[0] := Chr(Len);
```

```
    YazXY(XPos,YPos,Temp,Atr);
```

```
    if X >= XPos+Len then X := XPos+Len;
```

```
    if X <= XPos then X := XPos;
```

```
    GotoXY(X,YPos);
```

```
end;
```

```

begin
  FillChar(Temp,Len+1,F);
  for I := 1 to Length(Prompt) do if Prompt[I] = #32 then Prompt[I] := F;
  Insert(Prompt,Temp,1);
  X := XPos+Pos(F,Prompt)-1;
  Display;
repeat
  Ch := ReadKey;
  if Ch in [kbEnter, kbEsc] then
    begin
      for I := 1 to Len do
        if Temp[I] = F then Temp[I] := #32;
      ReadNumStr := Temp;
      Exit;
    end;
  if Ch = #0 then
    begin
      Ch := ReadKey;
      case Ch of
        kbLeft  : begin
                      Dec(X);
                      Display;
                    end;
        kbRight : begin
                      Inc(X);
                      Display;
                    end;
        kbHome  : begin
                      X := XPos;

```

```

        Display;
    end;
kbEnd  : begin
        X := XPos+Len;
        Display;
    end;
kbDel  : begin
        if X-XPos <> Len then
            begin
                Delete(Temp,X-XPos+1,1);
                Temp[Len] := F;
            end;
        Display;
    end;
end;
end
else if Ch = kbBkSpace then
    begin
        Temp[X-XPos] := F;
        Dec(X);
        Display;
    end
else if Ch in [#48..#58] then
    begin
        Insert(Ch, Temp, X-XPos+1);
        Inc(X);
        Display;
    end;
until False;
end;

```

```
function ReadWord(Prompt: word; Len, XPos, YPos, Atr : byte):word;
```

```
const
```

```
  F : char = '±'; {#177}
```

```
var
```

```
  Temp : string;
```

```
  X : byte;
```

```
  Ch : char;
```

```
  I : byte;
```

```
  S : string;
```

```
  Code : integer;
```

```
  W : word;
```

```
procedure Display;
```

```
begin
```

```
  Temp[0] := Chr(Len);
```

```
  YazXY(XPos, YPos, Temp, Atr);
```

```
  if X >= XPos+Len then X := XPos+Len;
```

```
  if X <= XPos then X := XPos;
```

```
  GotoXY(X, YPos);
```

```
end;
```

```
begin
```

```
  S := IntToStr(Prompt);
```

```
  FillChar(Temp, Len+1, F);
```

```
  Insert(S, Temp, 1);
```

```
  X := XPos+Length(S);
```

```
  Display;
```

```
repeat
```

```
  Ch := ReadKey;
```

```
if Ch in [kbEnter, kbEsc] then
```

```
  begin
```

```
    I := 1;
```

```
    repeat
```

```
      if Temp[I] = F then
```

```
        begin
```

```
          Temp[0] := Chr(I-1);
```

```
          Break;
```

```
        end;
```

```
        Inc(I);
```

```
      until I >= Len+1;
```

```
      Val(Temp, W, Code);
```

```
      ReadWord := W;
```

```
      Exit;
```

```
    end;
```

```
if Ch = #0 then
```

```
  begin
```

```
    Ch := ReadKey;
```

```
    case Ch of
```

```
      kbLeft  : begin
```

```
        Dec(X);
```

```
        Display;
```

```
      end;
```

```
      kbRight : begin
```

```
        Inc(X);
```

```
        Display;
```

```
      end;
```

```
      kbHome  : begin
```

```
        X := XPos;
```

```
        Display;
```

```

        end;
kbEnd    : begin
        X := XPos+Len;
        Display;
        end;
kbDel    : begin
        if X-XPos <> Len then
        begin
            Delete(Temp,X-XPos+1,1);
            Temp[Len] := F;
        end;
        Display;
        end;
end;
end
else if Ch = kbBkSpace then
begin
    Temp[X-XPos] := F;
    Dec(X);
    Display;
end
else if Ch in [#48..#58] then
begin
    Insert(Ch, Temp, X-XPos+1);
    Inc(X);
    Display;
end;
until False;
end;

```

```

function ReadUpStr(Prompt: string; Len,XPos,YPos,Atr: byte):string;
const
    F : char = '±'; {#177}
var
    Temp : string;
    X   : byte;
    Ch  : char;
    I   : byte;

procedure Display;
begin
    Temp[0] := Chr(Len);
    YazXY(XPos,YPos,Temp,Atr);
    if X >= XPos+Len then X := XPos+Len;
    if X <= XPos then X := XPos;
    GotoXY(X,YPos);
end;

begin
    FillChar(Temp,Len+1,F);
    for I := 1 to Length(Prompt) do if Prompt[I] = #32 then Prompt[I] := F;
    Insert(Prompt,Temp,1);
    X := XPos+Pos(F,Prompt)-1;
    Display;
repeat
    Ch := UpCase(ReadKey);
    if Ch in [kbEnter, kbEsc] then
        begin
            for I := 1 to Len do
                if Temp[I] = F then Temp[I] := #32;

```



```
ReadUpStr := Temp;
Exit;
end;
if Ch = #0 then
begin
Ch := ReadKey;
case Ch of
kbLeft  : begin
          Dec(X);
          Display;
        end;
kbRight : begin
          Inc(X);
          Display;
        end;
kbHome  : begin
          X := XPos;
          Display;
        end;
kbEnd   : begin
          X := XPos+Len;
          Display;
        end;
kbDel   : begin
          if X-XPos < Len then
            begin
              Delete(Temp,X-XPos+1,1);
              Temp[Len] := F;
            end;
          Display;
        end;
end;
```

```
        end;  
    end;  
end  
else if Ch = kbBkSpace then  
    begin  
        Temp[X-XPos] := F;  
        Dec(X);  
        Display;  
    end  
else {if IsBuyukHarf(Ch) then}  
    begin  
        Insert(Ch, Temp, X-XPos+1);  
        Inc(X);  
        Display;  
    end;  
until False;  
end;  
end.
```

## 7. SONUÇLAR

Sistemlerin kontrolü bilimler arası bir konu olup tüm mühendislik alanlarını ilgilendirmektedir. Bu nedenle kontrol sistemleri farklı üretim yapan işletmelerde bir çok makine, elektrik, elektronik, bilgisayar, vb. mühendislerini yakından ilgilendirmektedir. Kontrol sistemlerinin konstrüksiyonu ve donanımı daha çok elektrik, elektronik ve makine mühendislerini ilgilendirmektedir. Hızla gelişen kontrol konusu günümüzde sınırsızca büyüyen bir potansiyel olarak karşımıza çıkmaktadır. Kontrol sistemleri içerisinde bilgisayarların da kullanımı bu konuyu daha da geniş kapsamlı hale getirmiştir.

Genel amaçlı herhangi bir kontrol projesinde mühendislik açısından yapılamayacak uygulama yoktur. Fakat bu durum proje mühendislerinin istedikleri gibi hareket edebilecekleri anlamına gelmemektedir. Genel olarak baktığımızda kararların alınışı müşterilerin istekleri doğrultusunda yapılmalı ve hatta müşterilerin kullanmakta olduğu iş gücünün kapasitesi ve işletmenin ekonomi politikası göz önünde bulundurulmalıdır.

Yukarıda açıklandığı üzere bu tezin konusunu oluşturan çalışmanın tasarım ve uygulama denemelerinde hemen hemen her konuda yukarıdaki kısıtlayıcı etkilere rastlanmıştır. Bu bağlamda bu çalışmanın bir çok alternatifinin olması düşünülemez. Fakat yetenekli ve yaratıcı mühendislerin yapabilecekleri sürücü devresi ve mekaniki aksam tasarımı ve ergonomisi çeşitli alternatifler oluşturabilir.

Aynı zamanda çalışmada kullanımı uygun görülen step motorların yerine başka tür motorlar (firçasız doğru akım motoru ) da düşünülmesi muhtemeldir. Bu durumda donanım ve yazılımın değişmesi gerekecektir.

Yukarıda belirtilen tüm yöntem ve düşünceler ile kullanılması muhtemel diğer alternatifler çalışmanın ilk dönemlerinde düşünülmüş ( step motorun fiyatının daha uygun oluşu, istenilen sürülüşün daha kolay elde edilişi, mekaniki aksamda ağırlığı fazla olmasından dolayı rulman kullanılmayışı vs.) fakat sistemin hem maliyet hemde kullanılabilirlik ve verimlilik açısından bu çalışmada kullanılan sistemin en uygun olduğu sonucuna varılmıştır.

**KAYNAKLAR**

AustinHughes, 1993, "ELECTRIC MOTORS AND DRIVES",  
Newnes.: p.p.257,290

Denis O'Kelly, 1991, "PERFORMANCE AND CONTROL OF ELECTRICAL  
MACHINES", Cambridge University press.:p.p.285,310

Douglas W. Jones, "Stepping Motor Control Software", Internet Adres;  
*<http://dvt07.fagmed.uit.no/fag/step/software.html>*

Douglas W.Jones, "A Worked Stepping Motor Example", Internet Adres;  
*<http://dvt07.fagmed.uit.no/fag/step/circuits.html>*

Muhammad H. RASHID, 1993, "POWER ELECTRONICS CIRCUITS, DEVICES AND  
APPLICATIONS", Prentice Hall Int. Ed.:p.p.477,493

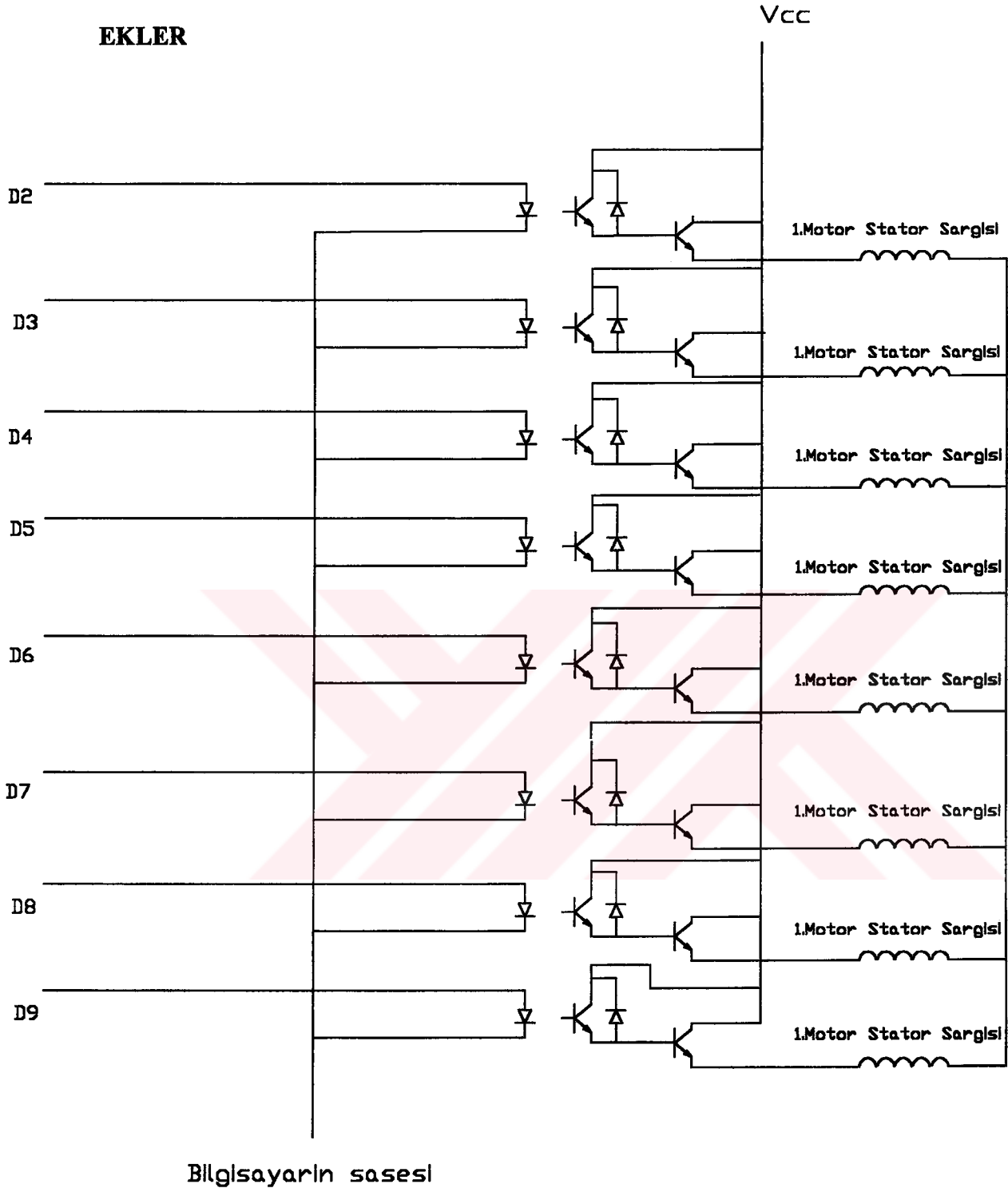
Prof.Y.Müh. Adnan ERGENELİ, 1988, "ELEKTROMAGNETİK ALANTEORİSİ II.  
KİTAP", p.p.69-71

Takashi KENJO, 1983, "STEPPING MOTORS AND THEIR MICROPROCESSOR  
CONTROL", Cambridge Pub.:p.p.1,87

Takashi KENJO, 1990, " POWER ELECTRONICS FOR THE MICROPROCESSOR  
AGE", Oxford University Press : p.p.170,199

TECQUIPMENT User's Manual, 1995,  
Tequipment Pub.:p.p.2-1, 2-28

## EKLER



Ek. 1. Kullanılan sürücü devresi

**ÖZGEÇMİŞ**

<b>Doğum tarihi</b>	24. 02. 1973	
<b>Doğum yeri</b>	Giresun	
<b>Lise</b>	1987-1991	Maçka Teknik Lisesi Elektrik Bölümü
<b>Lisans</b>	1991-1995	Yıldız Teknik Üniversitesi Elektrik-Elektronik Fak. Elektrik Mühendisliği Bölümü
<b>Yüksek Lisans</b>	1995-1998	Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik Mühendisliği Ana Bilim Dalı, Elektrik Mühendisliği Programı
<b>Çalıştığı Kurum</b>	1995-Devam	Y.T.Ü. Elektrik Müh. Bölümü Araştırma Görevlisi

Yıldız Teknik Üniversitesi  
Elektrik Mühendisliği Bölümü