

**GLOBAL OPTİMİZASYON YÖNTEMİ İLE
ASENKRON MOTOR HIZ KONTROLU****YÜKSEK LİSANS TEZİ****MÜH. MURAT ATA DEMİRÖZ****Tezin Enstitüye Verildiği Tarih : 3 Şubat 1992****Tezin Savunulduğu Tarih : 21 Şubat 1992****Tez Danışmanı : Doç. Dr. İbrahim EKŞİN****Diğer Jüri Üyeleri : Prof. Dr. M. Kemal SARIOĞLU****Doç. Dr. Metin GÖKAŞAN****ŞUBAT 1992****Y.Ü. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

ÖNSÖZ

Bu tez çalışmasını bana vererek, tüm çalışma boyunca zevkli ve rahat bir ortamda beni bilgileri ile destekleyip, pratik çalışmanın gerçekleştirilebilmesi için gerekli donanımın sağlanması konusunda gösterdiği yardımlardan dolayı hocam, sayın Doç.Dr.İbrahim EKŞİN'e sonsuz teşekkürlerimi sunarım. Çalışmam süresince, bilgilerini benimle içtenlikle paylaşan, sayın Doç.Dr. Metin GÖKAŞAN'a ve değerli dostum Y.Müh. Ogün TİĞLİ'ya çok teşekkür ederim. Optimizasyon programları üzerindeki çalışmalarım esnasında yaptığı katkılardan dolayı Y.Müh.Aylin T. USTA'ya da teşekkür ederim. Tüm öğrencilik yaşamım boyunca beni, maddi ve manevi olarak sabırla destekleyen babam, Yunus DEMİRÖZ'e de burada teşekkür etmek isterim. Tüm çalışma süresi boyunca bana sabırla dayanıp, sevgisi ile destek veren yaşam dostum Nalan AKÇAKAYA'ya da yürek dolusu sevgi ve teşekkürlerimi sunarım.

Üsküdar, Şubat 1992

Murat Ata DEMİRÖZ

İÇİNDEKİLER

SEMBOL LİSTESİ	v
ŞEKİL LİSTESİ	xii
TABLO LİSTESİ	xiv
ÖZET	xv
SUMMARY	xvi
BÖLÜM 1. GİRİŞ	1
BÖLÜM 2. GLOBAL OPTİMİZASYON YÖNTEMLERİ	9
2.1. Kontrollü Raslantısal Arama	10
2.2. Brownsal Hareket Temelli Raslantısal arama.	11
2.3. Yeni (Karma) Global Arama Yöntemi	13
BÖLÜM 3. GEÇİCİ REJİM DAVRANIŞ ÖLÇÜTLERİ	16
3.1. Birinci Dereceden Sistemler ve Geçici Rejim Davranış Ölçütleri.....	16
3.2. İkinci Dereceden Sistemler ve Geçici Rejim Davranış Ölçütleri.....	18
BÖLÜM 4. KAPALI ÇEVİRİM DİJİTAL KONTROL SİSTEMLERİ	24
4.1. Kapalı Çevrim Dijital Kontrol Sistemi	24
4.2. Uygulamada Kullanılan PID Kontrolörünün t , s , z , Tanım Bölgelerindeki Matematiksel Bağlantıları	25
BÖLÜM 5. KONTROL EDİLECEK SİSTEM OLARAK ASENKRON MOTOR.	36
5.1. Matematiksel Modelin Elde Edilmesi	39
5.2. Asenkron Makinaya PARK Dönüşümünün uygulaması ve d - q Modeli	42
5.2.1. Rotoru Sargılı Asenkron Makinanın Matematiksel Modelinin Akılar Kullanılarak Elde Edilmesi	50
5.3. Birime İndirgeme (Normalizasyon)	51
BÖLÜM 6. ASENKRON MOTOR SÜRÜCÜLERİ VE ÖZELLİKLERİ	57
6.1. Doğrudan Frekans Çeviricileri	58
6.2. Aradevrelili Frekans Çeviriciler	59
6.2.1. Akım Aradevrelili Frekans Çeviricileri	59
6.2.2. Gerilim Aradevrelili Frekans Çeviricileri .	61
6.2.2.1. Değişken Gerilim Aradevrelili Frekans Çeviricileri (DGAFC).....	61
6.2.2.2. Sabit Gerilim Aradevrelili Frekans Çeviricileri (SGAFC)	63

BÖLÜM 7. ASENKRON MAKİNANIN SKALER KONTROLU	65
7.1. Doğru Gerilim Aradevrelili Frekans Çevirici Kullanılarak, Gerilim/Frekans Oranının Sabit Tutulduğu Sürücü Sistemleri	66
7.2. Değişken Genlik ve Frekanslı Kaynak ile Beslenen Asenkron Makinaya İlişkin Matematiksel Bağlantılar	69
BÖLÜM 8. UYGULAMADA KULLANILAN SİSTEMİN TANITIMI	77
8.1. Uygulamada Kullanılan Düzeneğin Bolk Diyagramı	77
8.2. Uygulamada Kullanılan Asenkron Makinanın Teknik Verileri	78
8.3. Uygulamada Kullanılan Sürücünün Teknik ve Özellikleri	79
8.4. Uygulamada Kullanılan Giriş-Çıkış Kartının Teknik Verileri	81
BÖLÜM 9. GLOBAL ARAMA YAPILACAK BÖLGENİN SINIRLARININ ZEIGLER-NICHOLS YÖNTEMİNDEN YARARLANARAK BELİRLENMESİ	82
BÖLÜM 10. SONUÇLAR VE ÖNERİLER	86
KAYNAKLAR	111
EKLER	113
ÖZGEÇMİŞ	153

SEMBOL LİSTESİ

a	: Basamak yanıtı eğrisine çizilen maksimum eğimli teğetin kazanç eksenini kestiği nokta
A	: Φ 'nin argümanı
AA	: Fonksiyon değerlerinin saklandığı dizinin adı
$A_{\min}$	: Φ 'nin argümanının minimum değeri
β	: f_s / f_n
C_1	: t_r 'nin ceza katsayısı
C_2	: p_o 'nun ceza katsayısı
C_3	: e_{ss} 'nin ceza katsayısı
C_4	: g.d.'nin ceza katsayısı
$D(s)$	: Kontrolörün s-domenindeki transfer fonksiyonu
$D(z)$	: Kontrolörün z-domenindeki transfer fonksiyonu
ΔA	: Yamuk yöntemi ile integrasyonda 0-T aralığındaki alanın değeri
ΔA_1	: İleri yol integrasyonu ile bulunan alan
ΔA_{11}	: Öngörü operatörü ile integrasyonda bulunan alan
$E(s)$	: s-domeninde hata işareti
$e(t)$	: t-domeninde hata işareti
$E(z)$	: z-domeninde hata işareti
e_1	: T. örnekleme anına ilişkin hata işareti
e_2	: (T-1). örnekleme anına ilişkin hata işareti
e_3	: (T-2). örnekleme anına ilişkin hata işareti

E_{RO}	: Rotoru sargılı asenkron makinada, rotor uçları açık iken endüklenen gerilim
e_{ss}	: Kararlı hal hatası
$\vec{E}_s$	: Statorda endüklenen gerilim vektörü
$E[f(X)]$	: l_{sc} doğru parçası üzerindeki X noktasında, Brownsal hareket süreci olarak modellenen fonksiyonun beklenen değeri
$f(X_s)$	: Brownsal hareket temelli aramada, X_s noktasındaki fonksiyon değeri
$f(X_c)$	: Brownsal hareket temelli aramada, X_c noktasındaki fonksiyon değeri
$\bar{F}$	: Frekansın birim değeri
Φ	: $f_m^*(X)$ 'i K_m kadar geçme olasılığının hesaplandığı olasılık fonksiyonunun, alt fonksiyonu
Φ_{as}	: Stator a fazı akısı
Φ_{bs}	: Stator b fazı akısı
Φ_{cs}	: Stator c fazı akısı
Φ_{ar}	: Rotor a fazı akısı
Φ_{br}	: Rotor b fazı akısı
Φ_{cr}	: Rotor c fazı akısı
Φ_m	: Faydalı hava aralığı akısı
Φ_{md}	: Faydalı hava aralığı akısı d-ekseni bileşeni
Φ_{mq}	: Faydalı hava aralığı akısı q-ekseni bileşeni
Φ_{sd}	: Stator akısı d-ekseni bileşeni
Φ_{sq}	: Stator akısı q-ekseni bileşeni
Φ_{rd}	: Rotor akısı d-ekseni bileşeni
Φ_{rq}	: Rotor akısı q-ekseni bileşeni
Φ_{ON}	: Hava aralığı akısı anma değeri
$\bar{\Phi}_{sd}$	: Stator d-fazı akısı birim değeri
$\bar{\Phi}_{sq}$	: Stator q-fazı akısı birim değeri
$\bar{\Phi}_{rd}$	: Rotor d-fazı akısı birim değeri
$\bar{\Phi}_{rq}$	: Rotor q-fazı akısı birim değeri

f_L	: AA dizgesindeki N adet nokta içinde, en küçük değerli nokta
$f_m^*(X)$	: m ölçüm sonunda bulunan fonksiyonun maksimum değeri
f_{mil}	: Asenkron motorun mil frekansı (devir/s)
f_N	: Asenkron motorun anma frekansı (1/s)
f_{0N}	: Asenkron motorun anma frekansı
f_r	: Rotorda endüklenen gerilimin frekansı
$f_{r_{max}}$	: Rotorda endüklenen gerilimin maksimum frekansı
f_s	: Stator uygulanan gerilimin frekansı
f_p	: Yeni deneme noktası P'de, fonksiyonun değeri
$g.d.$	: Hızın referans değere otururken yaptığı geri dönüşün % cinsinden değeri
$\bar{i}$	: Akımın birim değeri
i_{as}	: Stator a-fazı akımı
i_{bs}	: Stator b-fazı akımı
i_{cs}	: Stator c-fazı akımı
i_{ar}	: Rotor a-fazı akımı
i_{br}	: Rotor b-fazı akımı
i_{cr}	: Rotor c-fazı akımı
$\bar{i}_\mu$	: Mıknatıslama akımı
I_N	: Stator faz akımı anma değeri
I_{Nr}	: Rotor akımı anma değeri
i_{sd}	: Stator d-fazı akımı
i_{sq}	: Stator q-fazı akımı
i_{rd}	: Rotor d-fazı akımı
i_{rq}	: Rotor q-fazı akımı
$\bar{i}_{sd}$	: Stator d-fazı akımının birim değeri
$\bar{i}_{sq}$	: Stator q-fazı akımının birim değeri
$\bar{i}_{rd}$	: Rotor d-fazı akımının birim değeri
$\bar{i}_{rq}$	: Rotor q-fazı akımının birim değeri
$\vec{i}_s$	: Stator akımı vektörü
$\vec{i}_r$	: Rotor akımı vektörü
$[I_s]$	: Stator akımları matrisi
$[I_r]$	: Rotor akımları matrisi

J	: Eylemsizlik momenti
ξ	: 2. dereceden sisteme ilişkin sönüm faktörü
k_c	: Sistemin kritik kazancı
K_p	: Oransal kontrolör parametresi
K_I	: İntegral kontrolör parametresi
K_D	: Türevsel kontrolör parametresi
K_s	: Stator sargı faktörü
K_r	: Rotor sargı faktörü
L	: Basamak yanıtı eğrisine çizilen maksimum eğimli teğetin zaman eksenini kestiği nokta
λ	: X noktasının X_s 'e olan uzaklığı
λ^*	: Φ 'nin argümanı A 'yı minimum yapan l_{st} doğrusunda, X noktasının X_s 'e olan uzaklığı
$[L_s]$	: Stator endüktans matrisi
$[L_r]$	: Rotor endüktans matrisi
L_s	: Stator özendüktansı
L_r	: Rotor özendüktansı
L_{so}	: Stator kaçak özendüktansı
L_{ro}	: Rotor kaçak özendüktansı
L_r^I	: Statora indirgenmiş rotor özendüktansı
L_{ro}^I	: Statora indirgenmiş rotor kaçak özendüktansı
$[L_{sx}(\theta)]$	: Stator ve rotora ilişkin endüktans matrisi
l_{st}	: X_s ve X_r arasındaki doğru parçasının uzunluğu
m_o	: Asenkron makinada endüklenen moment
$\bar{m}_o$	: Birime indirgenmiş endüklenen moment
M_{oN}	: Endüklenen momentin anma değeri
M_m	: Faydalı endüktans
M_N	: Asenkron makinanın milinden verdiği momentin anma değeri
m_s	: Stator faz sayısı
m_r	: Rotor faz sayısı

m_y	: Yk momenti
$\bar{m}_y$	: Yk momentinin birim deęeri
$[M_{sr}]$	: Stator ve rotor sargıları aras karşıt zendklem matrisi
$[M_{rs}]$	: Rotor ve stator sargıları aras karşıt zendklem matrisi
N	: AA dizgesinin eleman sayısı
$\bar{n}$	: Birime indirgenmiř hız
n_N	: Hızın anma deęeri
$\bar{n}_N$	: Birime indirgenmiř hızın anma deęeri
N_{0N}	: Bořtaki anma hız
N_s	: Stator bir fazındaki toplam sarım sayısı
N_r	: Rotor bir fazındaki toplam sarım sayısı
n_s	: Senkron devir hız
ω	: Asenkron motor milinin asal hız
ω_s	: Stator asal frekansı
ω_r	: Rotor gerilimin asal frekansı
Ω_{1N}	: Asal frekansın anma deęeri
p	: Makina ift kutup sayısı
P	: Yeni deneme noktas
P_1	: e_1 'in katsayısı
P_2	: e_2 'nin katsayısı
P_3	: e_3 'n katsayısı
$\bar{P}_a$	: Birime indirgenmiř g
PI	: Ama lt
P_N	: Asenkron motorun milinden verdięi gcn anma deęeri
$\bar{P}_n$	: Birime indirgenmiř gcn anma deęeri
p_o	: Ařm
P_{sN}	: Asenkron motorun elektriksel gcnn anma deęeri
q	: Yeni global arama yntenindeki alt dizge sayısı
R_r'	: Statora indirgenmiř rotor direnci
$[R_s]$	: Stator direnleri matrisi
$[R_r]$	: Rotor direnleri matrisi

$\bar{r}_r$	: Statora indirgendikten sonra birime indirgenmiş rotor direnci
s	: Kayma
σ	: Kaçak faktörü
t_c	: Kritik periyot
t_d	: Gecikme süresi
θ	: Stator a-fazı ile rotor a-fazı arasındaki konuma ilişkin açı
T_h	: Yolalma zaman sabiti
T_i	: İntegral kontrolör zaman sabiti
T_d	: Türevsel kontrolör zaman sabiti
T_N	: Zaman sabiti ($1/\Omega_{1N}$)
t_p	: Tepe süresi
T_p	: Sistemin geçici rejim salınımlarının periyodu
t_r	: Yükselme süresi
t_s	: Yerleşme süresi
$\bar{u}$	: Dönüştürme oranı
$U(s)$	: s-omeninde, kontrolör çıkış işareti
$u(t)$	: t-omeninde, kontrolör çıkış işareti
$U(z)$	: z-omeninde, kontrolör çıkış işareti
uk_1	: T. örnekleme anına ilişkin kontrol işareti
uk_2	: (T-1). örnekleme anına ilişkin kontrol işareti
v	: İlk arama bölgesi
$VAR[f(x)]$	: $f(X)$ 'in, X noktasındaki varyansı
$\bar{v}$	: Birime indirgenmiş gerilim
v_{as}	: Stator a-fazı gerilimi
v_{bs}	: Stator b-fazı gerilimi
v_{cs}	: Stator c-fazı gerilimi
v_{ar}	: Rotor a-fazı gerilimi
v_{br}	: Rotor b-fazı gerilimi
v_{cr}	: Rotor c-fazı gerilimi
v_{sd}	: Stator d-fazı gerilimi
v_{sq}	: Stator q-fazı gerilimi
v_{rd}	: Rotor d-fazı gerilimi

V_{rq}	: Rotor q-fazı gerilimi
V_{rd}'	: İndirgenmiş rotor d-fazı gerilimi
V_{rq}'	: İndirgenmiş rotor q-fazı gerilimi
V_c	: Sistem çıkışındaki gerilim
V_g	: Sistem girişindeki gerilim
V_N	: Stator gerilimi anma değeri
$V_{N_{max}}$	: Stator anma geriliminin maksimum değeri
V_{rN}	: Rotor gerilimi anma değeri
$[V_s]$	: Stator gerilimleri matrisi
$[V_r]$	: Rotor gerilimleri matrisi
$\vec{V}_s$	: Stator gerilim vektörü
$\vec{V}_r$	: Statora indirgenmiş rotor gerilim vektörü
$X(t)$	: Gauss olasılık sürecinin örneklenmiş fonksiyonu olarak modellenebilecek fonksiyon
$\bar{X}$	: Birime indirgenmiş endüktif reaktans
$x_c(t)$	: Sistem çıkış durum değişkeni
$x_g(t)$	: Sistem giriş durum değişkeni
X_m	: Faydalı özendüktif reaktans
X_N^*	: N adet ölçüm sonunda $X(t)$ 'nin maksimum değeri
X_s	: Stator özendüktif reaktansı
X_r	: Rotor özendüktif reaktansı
X_{sg}	: Stator kaçak özendüktif reaktansı
X_{rg}	: Statora indirgenmiş rotor kaçak özendüktif reaktansı
X_s	: n-boyutlu uzayın 2^n köşegenini birbirlerine bağlayan doğruların başlangıç noktaları
X_t	: n-boyutlu uzayın 2^n köşegenini birbirlerine bağlayan doğruların bitiş noktaları
Z_N	: Reaktans ve direncin temel değeri

ŞEKİL LİSTESİ

Şekil 3.1.	Birinci dereceden sistem, RL seri devresi..	16
Şekil 3.2.	Birinci dereceden sistemin basamak giriş işaretine verdiği yanıt	17
Şekil 3.3.	İkinci dereceden sistem, RLC seri devresi..	18
Şekil 3.4.	İkinci dereceden sistemin basamak giriş işaretine verdiği yanıt ve ölçütleri	22
Şekil 4.1.	Kapalı çevrim dijital kontrol sistemi blok diyagramı	24
Şekil 4.2.	Paralel PID kontrolörü blok diyagramı	25
Şekil 4.3.	a) Hata işareti $e(t)$ 'ye basamak yaklaşım, b) İleri yol integrasyonu blok diyagramı...	27
Şekil 4.4.	a) Hata işareti $e(t)$ 'ye öngörü operatörü ile basamak biçimli yaklaşım, b) Öngörü operatörü ile integrasyonun blok diyagramı	28
Şekil 4.5.	Yamuk yöntemi ile integrasyonun blok diyagramı	29
Şekil 4.6.	a) Poligonal tutuculu integratörün blok diyagramı b) Poligonal tutucu grafik yapısı c) Poligonal tutucunun blok diyagramı	30 31
Şekil 5.1.	Stator ve rotoru üç fazlı asenkron makinanın a,b,c eksen takımındaki eşdeğer devresi ...	38
Şekil 5.2.	Rotoru sargılı asenkron makinanın d-q eksen takımındaki eşdeğer devresi	47
Şekil 5.3.	Asenkron makinanın sürekli rejim eşdeğer devresi	49
Şekil 6.1.	Doğrudan frekans çevirici ilke şeması	58
Şekil 6.2.	Akım aradevrelili frekans çevirici ilke şeması	60
Şekil 6.3.	Değişken gerilim aradevrelili frekans çeviricileri ilke şeması	62
Şekil 6.4.	Sabit gerilim aradevrelili frekans çevirici ilke şeması	64
Şekil 7.1.	Değişken genlik ve frekansla beslenen asenkron motorun moment-hız özeğrileri	66
Şekil 7.2.	Açık çevrim V_g/f_g kontrolü ilke şeması ...	67
Şekil 7.3.	Kapalı çevrim V_g/f_g kontrolü ilke şeması ..	68
Şekil 7.4.	Değişken frekanslı kaynaktan beslenen asenkron makinanın T eşdeğer devresi	70
Şekil 7.5.	Rotor yanından stator yanına bakıldığında görülen Thevenin eşdeğer devresi	73
Şekil 8.1.	Uygulamadaki sistemin blok diyagramı	76
Şekil 9.1.	Zeigler-Nichols basamak yanıtı yönteminde incelenen basamak yanıt özeğrisi	83

Şekil 10.1.	$n_r = 1000$ dev/dak'da YÜKSÜZ durumda, sistemin kritik kazanç değerinde yaptığı osilasyonun grafiği	87
Şekil 10.2.	$n_r = 1500$ dev/dak'da, istenmeyen salınımı da içeren hız özeğrisi	90
Şekil 10.3.	$n_r = 1500$ dev/dak'da YÜKSÜZ durumda kontrol işareti ve hız grafikleri	91
Şekil 10.4.	$n_r = 1250$ ve $n_r = 1000$ dev/dak' da YÜKSÜZ durumda kontrol işareti ve hız grafikleri	92
Şekil 10.5.	$n_r = 750$ ve $n_r = 500$ dev/dak'da YÜKSÜZ durumda kontrol işareti ve hız grafikleri	93
Şekil 10.6.	$n_r = 1500$ dev/dak'da %50 YÜKLÜ durumda kontrol işareti ve hız grafikleri	94
Şekil 10.7.	$n_r = 1250$ ve $n_r = 1000$ dev/dak'da %50 YÜKLÜ durumda, kontrol işareti ve hız grafikleri	95
Şekil 10.8.	$n_r = 750$ ve $n_r = 500$ dev/dak'da %50 YÜKLÜ durumda, kontrol işareti ve hız grafikleri	96
Şekil 10.9.	$n_r = 1500$ dev/dak'da %100 YÜKLÜ durumda, kontrol işareti ve hız grafikleri	97
Şekil 10.10.	$n_r = 1250$ ve $n_r = 1000$ dev/dak'da %100 YÜKLÜ durumda, kontrol işareti ve hız grafikleri	98
Şekil 10.11.	$n_r = 750$ ve $n_r = 500$ dev/dak'da %100 YÜKLÜ durumda, kontrol işareti ve hız grafikleri	99
Şekil 10.12.	$n_r = 1375$ dev/dak'da yüksüz durumda enterpolasyon ile bulunan optimum parametrelerde, kontrol işareti ve hız grafikleri	101
Şekil 10.13.	$n_r = 1125$ ve $n_r = 875$ dev/dak'da yüksüz durumda enterpolasyon ile bulunan optimum parametrelerinde, kontrol işareti ve hız grafikleri	102
Şekil 10.14.	$n_r = 1500$ dev/dak'da, %25 yüklü durumda kontrol işareti ve hız grafikleri	103
Şekil 10.15.	$n_r = 1250$ ve $n_r = 1000$ dev/dak'da, %25 yüklü durumda kontrol işareti ve hız grafikleri	104
Şekil 10.16.	$n_r = 1500$ dev/dak'da, %75 yüklü durumda kontrol işareti ve hız grafikleri	105
Şekil 10.17.	$n_r = 1250$ ve $n_r = 1000$ dev/dak'da, %75 yüklü durumda kontrol işareti ve hız grafikleri	106
Şekil 10.18.	Sistemin $n_r = 1500$ dev/dak'da anma yükü ile ani olarak yüklenip, yük atmasına ilişkin kontrol işareti ve hız grafikleri	107
Şekil 10.19.	Sistemin $n_r = 1250$ ve $n_r = 1000$ dev/dak'da anma yükü ile ani olarak yüklenip, yük atmasına ilişkin kontrol işareti ve hız grafikleri	108
Şekil 10.20.	Sistemin $n_r = 750$ ve $n_r = 500$ dev/dak'da anma yükü ile ani olarak yüklenip, yük atmasına ilişkin kontrol işareti ve hız grafikleri	109

TABLO LİSTESİ

Tablo 5.1. Fiziksel büyüklüklerin birime indirgenmesi.....	52
Tablo 9.1. Zeigler-Nichols basamak yanıtı yöntemine göre önerilen PID kontrolörü parametreleri	83
Tablo 9.2. Zeigler-Nichols frekans yanıtı yöntemine göre önerilen PID kontrolörü parametreleri	84
Tablo 10.1. Çeşitli referans hız değerleri için kritik kazanç, kritik periyot ve kontrolör parametreleri üst sınır değerleri.....	88
Tablo 10.2. Sistemin YÜKSÜZ durumda çalışmasına ilişkin, PID kontrolör parametrelerinin optimumdeğerleri .	91
Tablo 10.3. Sistemin %50 YÜKLE çalışmasına ilişkin, PID kontrolör parametrelerinin optimum değerleri	94
Tablo 10.4. Sistemin %100 YÜKLE çalışmasına ilişkin, PID kontrolör parametrelerinin optimum değerleri	97
Tablo 10.5. Sistemin Ara noktalarda yüksüz çalışmasında, enterpolasyon ile bulunan optimum PID kontrolörü parametreleri	101
Tablo 10.6. Sistemin %25 Ara yükle çalışmasında, enterpolasyon ile bulunan optimum PID kontrolörü parametreleri	103
Tablo 10.7. Sistemin %75 Ara yükle çalışmasında, enterpolasyon ile bulunan optimum PID kontrolörü parametreleri	105

ÖZET

Bu yüksek lisans tezi çalışmasında, doğru gerilim aradevrelili frekans çevirici (DGAFÇ) ile skaler kontrol yöntemine göre sürülen, rotoru sargılı (Bilezikli) asenkron motorun hız kontrolü gerçekleştirilmiştir. Sürücü önünde yer alan kontrolör, dijital olarak, PID yapısında oluşturulmuştur. Çalışmada hedeflenen temel amaç, önceden belirlenen bir "Amaç Ölçütü"ne göre, PID kontrolör katsayılarını optimum olarak belirleyip, asenkron motorun istenen (referans) hıza ulaşana kadar, geçici rejimini kontrol etmektedir.

Optimum PID kontrolör katsayıları, Yeni Global Arama Yöntemi'ni kullanan bir program yardımı ile, önceden belirlenen sınırlar içinde, aranmaktadır. Kontrolör parametrelerinin optimum değerlerinin arandığı bölgenin sınırları ise, algoritması Nichols-Zeigler yönteminden yararlanılarak oluşturulmuş bir program yardımı ile belirlenmektedir.

Kullanılan "Yeni Global Arama Yöntemi", fiziksel sistemin kontrolü için gerekli verileri, sistem çıkış büyüklüklerinden alarak aramayı gerçekleştirdiğinden, sistemin matematiksel modeli ve transfer fonksiyonundan tamamen bağımsız olarak çalışır. Bu nedenle matematiksel modeli kolayca oluşturulabilen doğrusal (Lineer) sistemlere olduğu gibi, matematiksel modeli ve transfer fonksiyonu kolayca elde edilemeyen, doğrusal olmayan (non-lineer) sistemlere de kolayca uygulanabilir. Yapılan işlem: Sistem çıkış büyüklüklerini (konum, hız, akım) girdi olarak bilgisayara aktarıp, tanımlanan Amaç Ölçütü'ne göre kontrolör katsayılarını belirleyici aramayı yapmak ve bir sonraki adımda yeni katsayılarla göre oluşturulmuş kontrol işaretini, fiziksel sisteme tekrar uygulamak şeklinde özetlenebilir.

Çalışmanın sonunda bulunan optimum PID kontrolör parametrelerinin, sistemin geçici rejim davranışını istenilen şekilde kontrol edebildiği gözlenmiş olup, sonuçlar, çeşitli hız referansları için, grafiklerde sunulmuştur. Ayrıca değişik hız referansları ve yüklenme durumları için elde edilen optimum PID kontrolör katsayılarından oluşan tablolar da verilmiştir.

SUMMARY

SPEED CONTROL OF AN INDUCTION MACHINE USING WITH NEW GLOBAL OPTIMIZATION METHOD

The most commonly used machine in electrical drives is induction machine. Its better reliability, low maintenance cost, cheaper price are the main reason why it is preferred in many applications. However its application was limited by the complexity of its control which arises because of the variable-frequency supply, ac signals processing and the complex dynamics of the machine. The recent developments in power electronics have solved the variable-frequency supply problem by adequate frequency converters. On the other hand the implementation of microprocessors in the digital control circuits has introduced a wide scope of possibilities to overcome the signal processing.

In this work, the system which consist of an induction machine and its variable frequency drive, is controlled by the digital PID controller. The optimum values of digital PID controller parameters were found by the New Global Optimization Method, according to the described Performance Index. The Performance Index for the system is to be a weighted sum of the risetime (t_r), percent overshoot (po), steady state error (e_{ss}).

$$P.I. = -(C_1 \cdot t_r + C_2 \cdot po + C_3 \cdot e_{ss})$$

As we all know, the problem of the finding the best solution to a problem is very important for all the fields of engineering and science. If we define the problem in terms of mathematical representation such that the measure of the performance index is given by f , which is a real valued, non-linear function of n parameters, $X = (X_1, X_2, X_3, \dots, X_n)$. It will become simpler. Then the problem becomes minimizing or maximizing the function f . There is no real different between minimizing or maximizing since the maximum of $f(x)$ is the minimum of $-f(x)$.

In general the choice of which search method to use is common in dealing with the optimization problems since so many method exist, each with its own set of advantages and disadvantages.

The most difficult problems which require a search routine are the problems require a global search taking into account the entire solution space, not simply a small part of it and they are very common when dealing with real applications.

For search methods for solving the problems which have many local extrema the following attributes would be useful.

1. Derivatives of the function are not required. The search must be able to operate on functions which may not have derivatives at some points, may have discrete values and may have large flat areas. ($\text{grad}(f(x))=0$)
2. Very few parameters, each of which can be easily chosen. Minimum priori knowledge of the function is needed.
3. The method must be able to look beyond the local properties of the function to establish the global maximum over a set of bounded search variables.

In this study, a new global search method was applied to the system. The method uses W.L. PRICE's Controlled Random Search, B.E. STUCKMAN's Global Optimization Using Brownian Motion For Multivariable Functions. This method had been developed by Olcay BOZ.

In W.L. PRICE's Controlled Random Search, we are given a function of n variables and an initial search domain V which is defined by giving randomly over V . These points and their function values are stored in array AA . At each iteration a new trail point P is selected randomly. from a certain set of P satisfies the constraints. The function is evaluated at P and the function value f_p is compared with f_l . L is the point which has the lowest function value of the L points presently stored. If $f_p > f_l$ the point L is replaced in AA by P . If either P fails to satisfy the constraints of $f_p > f_l$ or it is not in the search boundaries then the trail is discarded and a fresh point is chosen from the potential trail set. algorithm goes on like that. As the algorithm proceeds the current set of N stored points tends to cluster around the maxima.

In 1963, H.J. KUSHNER proposed a search algorithm based upon the assumption that an unknown function could be modelled as a sample function of Gaussian Random Process- specifically a Brownian Motion Process.

He showed that the expected value of the unknown function $X(t)$ conditioned on all the measurements taken is a piece wise linear approximation of $X(t)$ itself. By using these two properties he was able to calculate the location of the best point to evaluate next.

The location of the next guess is obtained by finding the point with the maximum probability that $X(t)$ exceeds X_m^* (the maximum $X(t)$ found after m measurements) by some positive constant K_m or in other words

$$prob(X(t) \geq X_m^*(t) + K_m) = 1 - \Phi\left(\frac{X_m^*(t) + K_m - E[f(X)]}{\sqrt{VAR[f(X)]}}\right)$$

Φ : Cumulative normal density function.

This probability can be maximized by minimizing the quantity below:

$$A = \frac{(X_m^*(t) + K_m - E[X(t)])^2}{VAR[X(t)]}$$

from this the minimum value of A and its corresponding value of t can be found in closed form over an interval between two sample points. The values of A_{min} for each interval then compared and the minimum is taken as the best choice for next guess.

If we summarise the overall method,

1. Evaluate the function at its end points.
2. Find the $E[f(X)]$ and the variance for the segments for each successive points. (At the beginning we have only one segment)
3. Find A_{min} for each segmet.
4. Find the function value at t which correspond to the t of A_{min} .
5. Form two new segments around the new measurements.

B.E.STUCKMAN developed n-dimensional model of the Kushner's method. Search strategy is as follows:

1. Evaluate the function at the 2^n vertices of the n -dimensional space.
2. Construct the line segments l_{st} which connects X_s to X_t
3. Suppose that the measurements of the function can be modelled as Brownian Motion Process.
4. Take into account only the points along the line segmets. The expected value can be given as

$$E[f(X)] = f(X_s) + \lambda \frac{f(X_t) - f(X_s)}{|l_{st}|}$$

X a point on l_{st} , λ is the X 's distance from X_s . The conditional variance is

$$VAR[f(X)] = c \cdot \lambda \cdot \frac{|l_{st}| - \lambda}{|l_{st}|}$$

C is the mean square variation of the curve along l_{st} . Next we will find the probability that the unknown function will exceed $f_m^*(X)$, the largest value of the function after m guesses by K_m .

$$prob(f(X) \geq f_m^*(X) + K_m) = 1 - \Phi\left(\frac{f_m^*(X) + K_m - E[f(X)]}{\sqrt{VAR[f(X)]}}\right)$$

We can maximize the above quantity by minimizing

$$A = \frac{(f_m^*(t) + K_m - E[X(t)])^2}{VAR[X(t)]}$$

at the minimum point if we solve for λ^* ,

$$A = \frac{f_m^*(X) + K_m - f(X_s) - \lambda \cdot \frac{f(X_t) - f(X_s)}{|l_{st}|}}{c \cdot \lambda \cdot \frac{|l_{st}| - \lambda}{|l_{st}|}}$$

$$\lambda^* = \frac{(f_m^*(X) + K_m - f(X_s)) \cdot |I_{st}|}{2 \cdot f_m^*(X) + 2 \cdot K_m - f(X_s) - f(X_t)}$$

λ^* is independent of C. The maximum value of A is given as.

$$A_{\min} = \frac{4}{|I_{st}|} [(f_m^*(X) + K_m - f(X_s)) \cdot (f_m^*(X) + K_m - f(X_t))]$$

We find the line segment with the lowest $A_{\min}$ and the location of the best candidate for the next guess X_{m+1} by λ^* . X_{m+1} has the highest probability of exceeding the largest value found by K_m .

5. The function is evaluated at X_{m+1} and we continue in the same iterative manner.

The segment containing the new point is divided into two and the segments connecting the new point to the m-2 points are added. The value of $A_{\min}$ is found for each line segments including the newest ones. search is terminated after a predetermined number of points.

In the new hybrid method which had been developed by Olcay BOZ, proposed we are given a function of N bounded variables and an initial search domain is defined by using the limits of each variable. N trial points are chosen from the grids if the degree of function is more then five than there will be too much initial points and this will make the search very complicated. The function is evaluated at each trial point and the position and the function value corresponding to each point are stored in a table AA. Standart deviation of the table values of the points stored in AA are found and this is used for finding K_m . Line segments which connects each set of two points are constructed. Here we have two ways to go on,

1. To use the whole table for constructing line segments. In this way all the points in the table AA are connected with another point in the table by a line segments. We will have $m \cdot (m-1)/2$ line segments. (m: number of points stored in array AA)

2. To devide the table into sub-tables and construct line segments for each sub-table separately. In this way the table AA is divided into a predetermined number of sub-tables and each point in a subtable is connected with the other points. There will be no line segments between the points in different subtables. We will have $q(m/q)(m/q-1)/2$ line segments.

For each line segments the point that maximizes the probability that the function value will exceed $f_m^*(x)$. (The largest value of the function value in the table AA at that time.) by K_m is found. This is found by maximizing the probability

$$\text{prob}(f(X) \geq f_m^*(X) + K_m) = 1 - \Phi\left(\frac{f_m^*(X) + K_m - E[f(X)]}{\sqrt{\text{VAR}[f(X)]}}\right)$$

Since Φ is a monotonically increasing function, we can maximize the probability by minimizing the simpler quantity.

$$A = \frac{(f_m^*(t) + K_m - E[X(t)])^2}{\text{VAR}[X(t)]}$$

If we solve λ^* , the location of $A_{\min}$ yields,

$$\lambda^* = \frac{(f_m^*(X) + K_m - f(X_s)) \cdot |I_{st}|}{2 \cdot f_m^*(X) + 2 \cdot K_m - f(X_s) - f(X_t)}$$

λ is the distance of the newly found point from X_s . Function value of this point (P) is compared with the point (L) of the lowest function value in the array AA. If not the new point is discarded and another one is found on another line segment. After changing a point in the table lowest and highest function values in the array AA. are compared if they are identical to a predetermined accuracy we stop and our global maximum is the point with the highest function value in the table. If not we go on processing line segments after finding the new value of K_m and $f_m^*(x)$. If stopping criteria is not reached we contract line segments again in the table AA and iteration goes on in this manner.

For not restricting the search to the points near a local maxima we have to cover the entire search space for global searching. We can cover the entire search space by extending the line segments to the boundaries. By extending the line segments to the boundaries we can find points apart from each other. This decreases the probability of getting close to a local maximum.

BÖLÜM 1

GİRİŞ

Bilindiği gibi asenkron motorlar, mekanik yapılarının basitliği nedeni ile ucuz, hafif, bakım giderleri düşük ve yüksek güvenilirliğe sahip makinalardır. Temel mekanik yapılarına göre, Sincap Kafesli ve Rotoru Sargılı olmak üzere iki sınıfa ayrılırlar. Sincap kafesli asenkron motorların rotor devreleri, alüminyum çubuklar ve kısadevre halkalarından oluşup, oldukça basit bir mekanik yapıdadır. Bu nedenle günümüzde, küçük ve orta ölçekli sanayi kuruluşlarında %100'e varan bir kullanım hacmine sahiptir. Doğru akım makinalarındaki fırça-kollektör düzeneği gibi bir yapısı da olmadığından, çalışması esnasında ark oluşmaz. Bu nedenle de rafineri, yakıt depoları vb. yanıcı, parlayıcı ve patlayıcı ortamlar ile tozlu, kirli, nemli ve ıslak yerlerde güvenle kullanılabilirler.

Rotoru sargılı asenkron makinalarda ise rotor devresine, bilezikler üzerinden, elektriksel olarak ulaşma olanağı vardır. Rotora dışardan eklenen direnç düzeneği yada bu direnç düzeneği ile aynı işlevi yerine getirecek güç elektroniği devreleri yardımı ile rotoru sargılı asenkron motorda, yüksek kalkış momenti sağlanıp, yolalma akımı sınırlandırılır. Bu özelliği nedeni ile de büyük eylemsizlik momentine sahip, öğütme sistemleri, vinç ve kreyinler, yüksek güçlü karıştırıcılar, pompa ve havalandırma sistemleri gibi yüklerin tahrik edilmesinde yoğun olarak kullanılır. Çok geniş bir güç aralığı bölgesinde (0.5 kW'dan MW'lara kadar) üretilip, kullanılma olanağı vardır.

Ancak, asenkron makina, basit mekanik yapısına karşılık, oldukça karmaşık bir matematiksel yapıya sahiptir. Matematiksel olarak ortaya çıkan sistem, lineer olamayan (nonlineer) yapıdadır. Bu nedenle de asenkron makinanın hız ve akım kontrolü, oldukça karmaşık kontrol düzeneklerini gerektirir. Asenkron motorların kontrol edilebilme zorluğundan ve karmaşıklığından dolayı hız ve konum kontrolü gereken yerlerde uzunca bir süre doğru akım makinaları kullanılmıştır. Bilindiği gibi doğru akım makinaları lineer sistemler olup, konum ve hız kontroluna oldukça elverişli yapıdadırlar. Bu nedenle de kontrol düzenekleri basit ve ucuzdur. Ancak doğru akım makina sistemlerinin kuruluş ve bakım giderleri oldukça yüksektir. Mekanik yapıları, asenkron motorlara göre çok daha karmaşık olup, her ortamda kolayca kullanılamazlar.

Asenkron makinanın döner alan hızı,

$$n_s = 60 \frac{f_s}{p} \quad (1.1)$$

bağıntısında da görüleceği gibi, f_s statora uygulanan gerilimin frekansına ve makinanın p çift kutup sayısına bağlıdır.

$$n_s = n_r + n \quad (1.2)$$

bağıntısı uyarınca da rotorun hızı (n) ile rotorda oluşan döner alanın hızı (n_r) toplamı senkron döner alan hızını (n_s) verir. Böylece statora uygulanan gerilimin frekansının değiştirilmesi ile senkron döner alan hızı, dolayısı ile de rotor hızı değiştirilebilir.

Ancak asenkron motorun stator gerilimini anma değerinde sabit tutup, frekansını değiştirdiğimizde, frekansın anma değerinin altına inmesi halinde, döner alan ve rotor hızı azalacak, buna karşılıklı stator akımı artacaktır. Benzer olarak stator gerilimi frekansını anma değerinin üzerine çıkardığımızda ise, sabit stator gerilimi altında,

stator akımı azalacaktır. Bunun nedeni, statora ve rotora ilişkin özendüktif reaktansların ve kaçak reaktansların, stator gerilimi frekansına bağlı olarak değişmeleridir. Bu nedenle sadece stator gerilimi frekansının değişimi ile asenkron makinanın hızını kontrol etmek, tercih edilmeyen bir yöntemdir.

Sabit stator gerilimi ve frekansı altında, asenkron motorun hız kontrolünün bir diğer şekli de, çift kutup sayısının (p) değiştirilmesidir. Ancak bu işlem için özel stator sargı düzeneği gereklidir. Bu yöntem ile yapılan hız denetimi kademeli olup, ancak iki farklı hız değerine ulaşılmasını sağlar. Kademe değişimlerinde akım sıçramaları oluşur. Bu nedenle pek kullanışlı bir yöntem değildir.

Bilindiği gibi asenkron motorun bir faza indirgenmiş eşdeğer devresinden giderek, endüklenen moment ifadesi, üç fazlı sistemler için,

$$M_e = 3 \cdot \frac{1}{\frac{2 \cdot \pi \cdot n_s}{60}} \cdot \frac{R'_r}{s} \cdot \frac{V_s^2}{(R_s + \frac{R'_r}{s})^2 + (X_{sv} + X'_{rg})^2} \quad (1.3)$$

bağıntısı ile verilir. Statora uygulanan gerilimin frekansı sabit tutulduğunda, M_e , endüklenen moment, stator geriliminin genliğinin karesi ile orantılı değişmektedir. Bu yöntemde hız kontrol aralığı $n_d < n < n_s$ dir. Burada n_d devrilme momenti noktasındaki hız'dır. Ancak bu yöntemin temel amacı asenkron motorun hızını değil, hızlanmasını kontrol etmektir. Durma halindeki motorun, stator geriliminin genliği, oto transformatör veya alternatif gerilim kısıyıcısı ile yavaş yavaş yükseltilerek, motorun yumuşak yolalması sağlanır. Taşıyıcı (konveyör) bant, bobin sarma ve dokuma makinası gibi, yumuşak yolalması gerekli yüklerin tahrikinde kullanılır.

Yukarıda sözü geçen ve sakıncaları belirtilen kontrol yöntemlerinin sakıncalarını ortadan kaldıran ve asenkron motorun isteğe en uygun kontrolunu sağlayan yöntem, SKALER kontrol yöntemidir. Bu yöntemde statorda endüklenen gerilimin, statora uygulanan gerilimin frekansına oranı sabit kalacak biçimde, endüklenen gerilim ve frekans, birlikte değiştirilir. Uygulamalarda ise, statorda endüklenen gerilim yerine, stator uç gerilimi değiştirilir. Stator uç gerilimi, stator sargı direnci ve kaçak reaktansı üzerinde düşen gerilim ihmal edilirse, statorda endüklenen gerilime eşittir. Böylece, statordan rotora geçen hava aralığı akısı Φ_m sabit tutulur. Akının sabit kalması ile, rotorda endüklenen momentin, stator gerilimi frekansının anma değerinin altındaki değerleri için, devrilme momenti noktasının sabit kalması sağlanır. Bunun fiziksel anlamı şudur: Anma değerinin altındaki farklı stator gerilimi frekansları ve mil hızları için, yüke aktarılan momentin değeri aynı kalır. Buna asenkron motorun moment-hız özeğrisi üzerinde sabit moment bölgesinde çalışması adı verilir. Ancak bu noktada bir küçük ayrıntı vardır. V_s 'nin anma değerinin çok altına indiği noktalarda, I_s sabit olduğundan, R_s üzerinde düşen gerilim artık ihmal edilemez düzeye gelir. Bu da endüklenen gerilimin çok küçülmesine ve Φ_m hava aralığı akısının azalmasına neden olur. Bu nedenle belli bir frekans değerinin altında, $V_s/f_s = \text{sabit}$ ilkesi uygulanamaz. Bu durumda V_s sabit tutulup, frekans düşürülür. Buna asenkron motorun skaler kontrolunda IR kompanzasyonu adı verilir.

Asenkron motora uygulanan gerilimin genliği anma değerinin üzerine çıkarılamaz. Çünkü aşırı akımlardan dolayı, sargı yalıtkanlarının zarar görmesi söz konusudur. Ancak frekans anma değerinin üzerine çıkarılabilir. Bu durumda frekans için üst sınırı, motorun mekanik yapısı belirleyecektir. Sincap kafesli asenkron motorlar için $f_{Smax} = 2 \cdot f_{SN}$ alınabilir. Yani stator geriliminin frekansı anma değerinin iki katına dek çıkarılabilir. Sincap

kafesli asenkron motorun, rotoru ve yatakları bu hıza dek dayanabilir. Ancak rotoru sargılı asenkron makina için bu sınır, $f_{smax} = 1.5 \cdot f_{SN}$ olarak yani, frekansın anma değerinin 1.5 katı olarak bulunur. Bunun üzerindeki frekanslar için rotorun ulaşacağı hızlarda, merkezkaç kuvvet etkisi ile rotor sargıları oluklarından oynayıp, stator oluklarına sürtünerek zarar görebilir. Asenkron motorun, stator anma gerilimi altında, anma frekansının üzerindeki noktalarda çalışmasına, moment-hız özeğrisi üzerinde, sabit güç bölgesinde çalışma adı verilir. Skalar kontrol yöntemi ile ilgili bu noktaya dek anlatılanlardan anlaşılacağı gibi, asenkron motorun kontrolü için değişken genlik ve frekanslı, alternatif gerilim kaynağına gereksinim vardır.

Güç elektroniği elemanlarının, günümüzdeki kadar çok gelişmediği dönemlerde, böylesi bir kaynak elde etmek oldukça sorun olmuştur. Doğru akım motorları (Ward-Leonard grubu) ve senkron generatörden oluşan bir yapı ile değişken genlik ve frekanslı bir kaynak elde etmek mümkündür. Ancak böylesi bir sistemin kuruluş maliyeti ve bakım giderlerinin ne kadar yüksek olacağı açıkça bellidir. Son yirmi yıl içinde güç elektroniği ve bilgisayar teknolojisindeki büyük gelişmeler sayesinde asenkron motorların kontrolü sorun olmaktan çıkmıştır. İlerleyen bölümlerde daha da ayrıntılı anlatılacağı gibi 'Sabit Gerilim Aradevreli Frekans Çeviriciler'le (SGAFC), günümüzde asenkron motorun kontrolü kolaylıkla başarılmaktadır.

Asenkron makinanın bilinen bir diğer özelliği ise, moment-hız özeğrisinde, lineer çalışma bölgesi içinde kalındığı sürece, yüklendikçe mil hızının çok fazla düşmemesidir. Ancak bu, motor yüklendikçe hızı sabit kalıyor demek değildir. Bilindiği gibi asenkron motorların boşta çalışmadaki mil hızları, senkron devir hızına oldukça yakındır. Anma yükü ile yüklendiği durumda ise, senkron devir hızının %4 - %8 altına düşer. Duyarlı hız kontrolünün gerektiği kağıt, demir-çelik, dokuma ve cam fabrikaları

gibi kuruluřlarda, her eřit yklenme durumu iin, mil hızının sabit kalması istenebilir. Bu durumda asenkron motorun kapalı evrim kontrol dzenleri ile kontrol edilmesi gerekir. Hıza iliřkin bilgi, motor miline baėlı dairesel kodlayıcı (encoder) veya takogeneratr zerinden alınır. İstenen (referans) deėer ile mil hızı arasındaki fark, hata iřareti olarak belirlenir. Hata iřareti kontrolrde iřlenerek, uygun kontrol iřareti elde edilir. Bu kontrol iřareti srcye uygulanıp, eřitli yklenme durumları iin, asenkron motorun mil hızının istenen deėerde sabit kalması saėlanır.

Gnmz piyasa uygulamalarında, kapalı evrim konum, hız ve akım kontrolunda, kontrolr olarak PI (Oransal + integral) veya PID (Oransal + integral + Trevsel) tipi kontrolrler kullanılır. Uzun yıllar boyunca bu tip kontrolrler analog olarak retilmiřtir. Gnmzde geliřen bilgisayar ve mikroiřlemci teknolojisi ile, bu tip kontrolrler diėital olarak gereklenmektedir. Diėital kontrolr kartları, analog-diėital ve diėital-analog dnřtrcleri, silinip-yazılabilir bellekleri, diėital giriř-ıkıř birimleri ve mikroiřlemcisi ile birlikte tek bir kart halinde retilabilmektedir. Bylece pratik kullanıma son derece yatkın hale gelmiřlerdir.

Gnmz piyasa uygulamalarında, PID kontrolrler iin en uygun parametrelerin belirlenmesi, genellikle deneme-yanılma yntemi ile yapılmaktadır. Bu yntemle raslantısal olarak parametreler belirlenmeye alıřılır. K_p , K_I , K_D katsayıları deneme-yanılama řeklinde, eřitli deėerlere raslantısal olarak ayarlanıp, sistemin kendinden beklenen kriterlere yakın bir noktada alıřması saėlanmaya alıřılır.

Bu yksek lisans tezinde yapılan alıřma, asenkron motor ve srcsnden oluřan bir sistemin nnde bulunan PID kontrolrnn, K_p , K_I , K_D parametrelerinin optimum

değerlerinin, bilimsel bir yöntemle, bulunmasını içermektedir. Parametrelerin en uygun değerleri belli sınırlar içinde, Yeni Global Arama Yöntemi kullanılarak bulunmaktadır. Global aramanın yapılacağı bölgenin sınırları ise, Nichols-Zeigler yönteminden yararlanılarak geliştirilen bir yöntemle belirlenmektedir. Bu sınırlar içinde, belirlenen bir Amaç Ölçütü'nün minimum yapılmasını sağlayacak, en uygun parametre değerleri bulunmaktadır. Bulunan parametre değerleri ile asenkron motor, boşta ve çeşitli yüklenme durumları için, istenen hız değerine, minimum yükselme zamanında, aşısız ve kararlı hal hatası olmaksızın ulaştırılabilmektedir.

Geliştirilen global arama yönteminin en belirgin özelliği, kontrol edilen sistemin matematiksel modelinden tamamen bağımsız çalışmasıdır. Böylece lineer bir sistem olan doğru akım makinasına uygulanabildiği gibi [1], lineer olamayan (nonlinear) asenkron makinaya da kolaylıkla uygulanabilir. Yöntemin çalışmasındaki temel ilke, sistemin çıkış büyüklüklerini (konum, hız, akım) girdi olarak alıp, bunları işleyerek uygun kontrol işaretini üretip, sisteme tekrar uygulamaktır.

Ancak bu noktada önemli bir sorunla karşılaşılmaktadır. Asenkron makinanın lineer olmayan (non-linear) matematiksel yapısında dolayı, sistem üzerinde herhangi bir çalışma noktası için bulunacak optimum değerler, bir başka çalışma noktası için, optimum olma koşulunu sağlamamaktadır. Bu nedenle sistem üzerinde ölçümler yapılarak, çeşitli çalışma noktaları için ayrı ayrı optimum değerlerin, bulunması gerekmektedir. Ancak tüm çalışma noktaları için de ölçümler yapmak pratik bir çözüm yolu olmamaktadır. Bu nedenle belli noktalarda sistem üzerinde ölçümler yaptıktan sonra, istenen ara çalışma noktaları için optimum değerler, lineer enterpolasyon yardımı ile bulunabilmektedir.

Yöntemin tek sakıncası, çevrim-dışı (off-line) çalışan bir yapıda olup, en uygun sonuca ulaşana kadar, ard arda çok sayıda deneme yapılmasını gerektirmesidir. İkinci bölümde, kullanılan global arama yönteminin çalışma ilkeleri ve matematiksel yapısı detaylı olarak açıklanmıştır. Üçüncü bölümde, birinci ve ikinci dereceden sistemlerin durum denklemlerinin elde edilmesi anlatılıp, bu sistemlerin basamak giriş işareti için verdikleri yanıt üzerinde geçici ve sürekli rejim ölçütlerinin tanımları yapılmıştır. Dördüncü bölümde, tezde kullanılan PID kontrolör algoritmasının t , s ve z domenindeki matematiksel bağıntıları verilmiştir. Beşinci bölümde kontrol edilen sistem olarak asenkron motorun matematiksel modeli, a - b - c koordinat eksen takımıyla elde edilmiştir. Sonra bu bağıntılara PARK dönüşümü uygulanıp, d - q dönen koordinat sisteminde, kontrol açısından çok daha uygun bir yapıya ulaştırılmıştır. d - q eksen takımıyla asenkron makina bağıntıları, son olarak da birime indirgenerek, kontrol açısından çok daha yalın yapıya ulaştırılmıştır. Altıncı bölümde asenkron motor sürücü devrelerinin çeşitleri ve yapıları anlatılmıştır. Yedinci bölümde asenkron makina SKALER kontrolün uygulanması ve V_s/f_s oranının sabit tutulması durumunda devrilme momentinin nasıl sabit kaldığının ispatı verilmiştir. Sekizinci bölümde, pratik çalışmanın yapıldığı sistemin elemanları ayrı ayrı tanıtılmıştır. Dokuzuncu bölüm, global arama bölgesi sınırlarının belirlenmesinde yararlanılan Zeigler-Nichols yönteminin tanıtımını içermektedir. Son bölümde ise, asenkron motor ve sürücüsünden oluşan sistemi kontrol eden PID kontrolörü katsayılarının optimum değerleri, sistemin çeşitli çalışma noktaları için tablolar halinde verilmiştir. Sistem üzerinde yapılabilecek geliştirmelere ilişkin öneriler de bu bölümün sonunda sunulmuştur.

BÖLÜM 2

GLOBAL OPTİMİZASYON YÖNTEMLERİ

Giriş bölümünde de belirtildiği gibi, bir sistemin belli bir çalışma bölgesi içinde, amaca en uygun biçimde çalıştırılmasını sağlamak, mühendisliğin temel sorunlarından biridir. Bu yüksek lisans tezi çalışmasında, asenkron motor ve sürücüsünden oluşan bir sistemin önünde bulunan dijital PID kontrolörünün parametreleri, sistemin çeşitli çalışma noktaları için optimum belirlenmiştir. Bu amaçla sistemin geçici rejim kriterleri olan yükselme zamanı (t_r), aşım (po) ve kararlı hal hatası (e_{ss}) ile bunların ağırlık katsayılarından oluşan (c_1, c_2, c_3) bir Amaç Ölçütü (Performance Index) tanımlanmıştır.

$$P.I. = -(c_1.t_r + c_2.po + c_3.e_{ss}) \quad (2.1)$$

Temel amaç, bu geçici rejim kriterlerini dolayısı ile de amaç ölçütünü minimum yapan, PID kontrolörü parametrelerinin (K_p, K_I, K_D) bulunmasıdır.

Geliştirilen Yeni Global Arama Yöntemi [2], tanımlanan bir fonksiyonun, belli sınırlar içinde global maksimumunu bulmaya yöneliktir. Çalışmanın temel amacı tanımlanan amaç ölçütünün global minimumunu bulmak olduğu için, amaç ölçütü $f(x) = -f(x)$ yapısında oluşturularak, programın bulduğu global maksimumun, gerçek sistemin amaç ölçütünün global minimumu olması sağlanmıştır.

Optimizasyonda bir problemin, belirlenen arama alanı içinde birçok lokal minimum yada lokal maksimumu bulunabilir. Böylesi durumlarda problemin çözümü global

arama yöntemlerindedir. Global arama yöntemleri gerçek sistemlerin sorunlarına da pratik çözümler getirir. Gerçek uygulamalarda kullanılacak olan arama yönteminin şu özellikleri taşıması gerekir:

1. Amaç ölçütü fonksiyonunun türevlerine gerek duymamalı.
2. Hepside seçilebilir az sayıda parametre ile fonksiyon hakkında çok az bilgiye gereksinim olmalı.
3. Arama yöntemi, fonksiyonun bölgesel (lokal) yapısının ötesine de bakabilmeli, lokal minimum yada lokal maksimum noktalarına yakınsama olasılığı az olmalıdır.

Bu amaçla aşağıda, yeni global arama yönteminin dayandığı, Kontrollü Raslantısal Arama ve Brownsal Hareket Temelli Raslantısal Arama yöntemleri kısaca aktarılacaktır. Ardından da Yeni Global Arama Yöntemi anlatılacaktır.

2.1. KONTROLLU RASLANTISAL ARAMA YÖNTEMİ

Bu yöntemde n değişkenli bir fonksiyon ve her değişkene ilişkin sınır değerleri ile belirlenen bir V ilk arama bölgesinin verilmesi gerekir. V 'de raslantısal olarak N adet nokta belirlenir. Bu noktalar ve bunların fonksiyon değerleri bir AA dizgesinde saklanır. Yöntemin her bir ötelemesinde, yeni bir deneme noktası P , belirli deneme noktaları arasından olası olarak seçilir. Bu seçilen nokta, sınır değerlerinin içinde kalıyorsa, fonksiyon bu noktada değerlendirilir. P noktasındaki fonksiyonun değeri f_p , o anda N adet saklanan nokta içindeki en küçük değer olan f_l ile karşılaştırılır. Eğer $f_p > f_l$ ise AA dizgesinde L noktası, P ile yer değiştirilir. Eğer $f_p \leq f_l$ ise veya P noktası sınır değerlerin dışında ise bu noktadan vazgeçilip seçim kümesi içinden yeni bir nokta seçilir. Bu şekilde, algoritma ile N adet noktanın maksimum değer etrafında toplanmasını sağlanır.

2.2. BROWNSAL HAREKET TEMELLİ RASLANTISAL ARAMA

H.J.KUSHNER [3], bir fonksiyonun, Gauss olasılık sürecinin örneklenmiş fonksiyonu olarak modellenebileceği bir arama yöntemi önermiştir. $X(t)$ fonksiyonunun 0 ana kadar alınan değerlerine bağlı olarak belirlenen, beklenen değerinin, aynı $X(t)$ 'nin X_1 'i (N adet ölçüm sonunda elde edilen $X(t)$ 'nin enbüyük değeri) en yüksek olasılıkla geçme olasılığıdır.

B.STUCKMAN [4], [5], [6], KUSHNER'in tek boyutlu uzay için yaptığı bu çalışmayı, n boyutlu uzay için geliştirmiştir. n boyut için de tüm değişkenler sınırlıdır.

Önerilen n-boyutlu arama yönteminin temel esasları şöyle sıralanabilir:

1. Fonksiyon n boyutlu uzayın, 2^n köşegeninde değerlendirilir.
2. X_s noktası ile X_t noktalarını birleştiren doğru parçaları oluşturulur.
3. Fonksiyon ölçümlerinin, Brownsal Hareket Süreci olarak modellenebileceği varsayımı ile, l_{st} doğru parçası üzerindeki noktalar gözönüne alınarak, fonksiyonun beklenen değeri için,

$$E[f(X)] = f(X_s) + \lambda \frac{f(X_t) - f(X_s)}{|l_{st}|} \quad (2.1)$$

bağıntısı yazılabilir. Burada X , l_{st} üzerinde bir nokta ve λ ise X 'in X_s 'e olan uzaklığıdır.

Varyans ise,

$$VAR[f(X)] = c \cdot \lambda \cdot \frac{|I_{st}| - \lambda}{|I_{st}|} \quad (2.3)$$

şeklinde yazılabilir. Bu bilgiler ışığında yeni belirlenecek X noktasının, $f_m^*(x)$ 'i (m ölçüm sonunda bulunan, fonksiyonun maksimum değeri), K_m kadar geçme olasılığı,

$$prob(f(X) \geq f_m^*(X) + K_m) = 1 - \Phi\left(\frac{f_m^*(X) + K_m - E[f(X)]}{\sqrt{VAR[f(X)]}}\right) \quad (2.4)$$

bağıntısı ile bulunur.

Bu olasılığın maksimum olduğu noktayı, I_{st} doğru parçası boyunca ararsak, fonksiyonun yeni değerlendirilme noktasını bulabiliriz. (2.4) bağıntısında Φ monoton artan bir fonksiyondur. Bu nedenle argümanının minimum değerinin bulunması, Φ 'yi minimum yapar. Bu da X noktasındaki fonksiyon değerinin, $f_m^*(x)$ 'i, K_m kadar geçme olasılığının maksimum olması demektir.

$$A = \frac{(f_m^*(X) + K_m - E[f(X)])^2}{VAR[f(X)]} \quad (2.5)$$

bağıntısının, λ^* için minimum değeri bulunursa, beklenen değer $E[f(x)]$ 'in değeri, A 'da yerine konularak,

$$A = \frac{f_m^*(X) + K_m - f(X_s) - \lambda \cdot \frac{f(X_t) - f(X_s)}{|I_{st}|}}{c \cdot \lambda \cdot \frac{|I_{st}| - \lambda}{|I_{st}|}} \quad (2.6)$$

$$\lambda^* = \frac{(f_m^*(X) + K_m - f(X_s)) \cdot |I_{st}|}{2 \cdot f_m^*(X) + 2 \cdot K_m - f(X_s) - f(X_t)} \quad (2.7)$$

$$A_{\min} = \frac{4}{|I_{st}|} [(f_m^*(X) + K_m - f(X_s)) \cdot (f_m^*(X) + K_m - f(X_t))] \quad (2.8)$$

bağıntıları elde edilir. $A_{\min}$ 'i minimum olan doğru parçası üzerinde bulunan nokta, yeni değerlendirme noktamız olur.

4. Fonksiyon X_{m+1} 'de hesaplanır ve ardışıl olarak algoritmaya devam edilir.

Yeni belirlenen nokta yardımı ile, bu noktanın üzerinde bulunduğu doğru parçası ikiye bölünür. Yeni noktayı, diğer noktalara bağlayan doğru parçaları, eski doğrulara eklenir. Arama önceden belirlenmiş sayıda ötelemeden sonra durdurulur.

2.3. YENİ GLOBAL OPTİMİZASYON YÖNTEMİ

O.BOZ [2] tarafından geliştirilen bu yöntemde, n değişkenli bir fonksiyon üzerinde çalışılır. Arama bölgesi, herbir değişken için belirlenen sınırlar ile tanımlıdır. Arama bölgesinde N adet deneme noktası raslantısal olarak seçilir. Buna seçenek olarak da deneme noktaları, arama bölgesini belirli sayıda alt bölgeye ve bu bölgeleri de noktasal arama yapacak şekilde, belirgin M adet noktaya bölerek, bunların içinden N adet nokta da seçilebilir. ($M \geq N$) Seçilen deneme noktalarındaki fonksiyon değerleri bulunur ve bu deneme noktaları ile bu noktalardaki fonksiyon değerleri bir AA dizgesinde saklanır. Dizgedeki fonksiyon değerlerinden yararlanarak standart sapma bulunur. Bulunan standart sapma'dan K_m değerinin hesaplanır. (K_m değerinin hesaplanmasında daha farklı yöntemler de kullanılabilir.) Dizgedeki her iki noktayı birbirleri ile birleştiren doğru parçaları oluşturulur. Burada iki ayrı yöntem uygulanabilir. Birinci yöntem, doğru parçalarını oluştururken tüm dizgeyi kullanmaktır. Bu durumda, N , AA dizgesindeki nokta sayısı olmak üzere

$$\frac{N \cdot (N - 1)}{2} \quad (2.9)$$

adet doğru parçası elde edilir. İkinci yöntemde, AA dizgesi belli sayıda alt dizgeye bölünerek, doğru parçaları sadece alt dizgedeki noktalar arasında oluşturulur. Değişik alt dizgeler arasındaki noktalar arasında doğru parçaları yoktur. Bu durumda q alt dizge sayısı ve N de AA dizgesindeki nokta sayısı olmak üzere,

$$\frac{q}{2} \left(\frac{N}{q} \right) \left(\frac{N}{q} - 1 \right) \quad (2.10)$$

adet doğru elde edilmiş olur.

İlk yöntemde daha çok belleğe ve işlemci zamanına gereksinim duyulur. Fakat daha doğru bir global maksimum bulma olasılığımız vardır.

Her fonksiyon için, fonksiyonun o ana kadar bulunan maksimum değeri $f_m^*(X)$ 'i K_m kadar geçmesi olasılığını, maksimuma ulaştıran nokta, (2.4) bağıntısı ile bulunur.

Φ monoton artan yada azalan bir fonksiyon olduğundan, yukarıdaki olasılığı (2.5) bağıntısının minimum olduğu yeri bularak da maksimum yapabiliriz.

Eğer, λ için çözüm yaparsak, A_{min} 'in yeri bize (2.7) bağıntısındaki λ değerini verecektir.

λ, yeni bulunan noktanın, doğru parçasının başlangıç noktasından olan uzaklığıdır. Yeni bulunan P noktasındaki fonksiyon değeri f_p , AA dizgesindeki minimum fonksiyon değeri olan f_L ile karşılaştırılır. Eğer $f_p > f_L$ ise, L noktası dizgeden çıkarılıp yerine P noktası atanır. Eğer $f_p \leq f_L$ ise nokta dikkate alınmaz ve diğer doğruların taranmasına geçilir. P ile L yer değiştirildiğinde, AA dizgesindeki maksimum fonksiyon değeri f_L ile minimum

fonksiyon değeri f_L karşılaştırılır Eğer aralarındaki önceden belirlenen bir değerin (durma kriteri) altında ise öteleme durdurulur ve AA dizgesindeki en büyük değer global maksimum olur. Eğer durma kriteri sağlanmamış ise diğer doğru parçalarında aynı aramalar yapılarak ötelemeye devam edilir. Tüm doğrular tarandıktan sonra durma kriteri hala sağlanmadiysa, yeni oluşan AA dizgesindeki noktalar arasında doğru parçaları oluşturulur ve ötelemeye yukarıdaki şekilde devam edilir.

Her ötelemeden sonra AA dizgesi, dizgedeki en yüksek fonksiyon değerli noktaya yakınsamaya çalışacaktır. Tüm doğru parçalarında arama yapıldıktan sonra eğer durma kriteri sağlanmamış ise yeni oluşan AA dizgesinde noktalar arasında doğru parçaları oluşturarak aramaya devam edilir. Fakat her ötelemeden sonra arama bölgesi küçülecektir. Bu nedenle global maksimum, arama bölgesi dışında kalabilir ve bulunamayabilir. Bu nedenle başlangıçta verilen bölgenin tümünde arama yapılması gerekir. Bunu yapmak içinde aramayı, global ve lokal olarak iki kısımda incelemek gerekir. Global aramada doğru parçaları, sınırları kesene dek uzatılır. Böylece arama, tüm bölgeye yayılmış olup, belli bir alana sıkıştırılmamış olur. Eğer tüm aramaları global yaparsak yakınsama çok yavaş olur. Yakınsamayı hızlandırmak için lokal aramadan yararlanılır. Lokal aramada, dizgedeki noktalar arasındaki doğru parçaları normal şekilde oluşturulur.

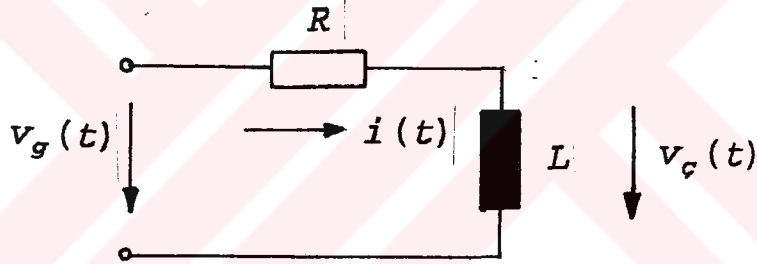
Aramaya global olarak başlayıp bir kaç ötelemeden sonra (Dizgedeki noktalardan en az birinin global maksimum yakınlarında olduğunu kabul ederek) lokal aramaya geçilerek, arama hızlandırılabilir. Ya da global arama ile yerel arama dönüşümlü olarak kullanılır.

BÖLÜM 3

GEÇİCİ REJİM DAVRANIŞ ÖLÇÜTLERİ

3.1. BİRİNCİ DERECEDEKİ SİSTEMLERİN GEÇİCİ REJİM DAVRANIŞ ÖLÇÜTLERİ

Birinci dereceden sisteme örnek olarak, bir doğru akım makinasının uyarma sargısı devresi örnek verilebilir. RL seri devresi şeklindeki uyarma sargısı devresi, tipik olarak birinci dereceden sistem davranışı örneğini gösterir.[7]



Şekil 3.1. Birinci dereceden sisteme örnek, RL seri devresi.

Akım - gerilim bağıntısı,

$$v_g(t) = R \cdot i(t) + L \cdot \frac{d}{dt} i(t) \quad (3.1)$$

birinci dereceden diferansiyel denklem olarak düzenlendiğinde,

$$\frac{v_g(t)}{R} = i(t) + \frac{L}{R} \frac{d}{dt} i(t) \quad (3.2)$$

yapısına ulaşılır. Burada,

$$\tau = \frac{L}{R} \text{ [s]} \quad (3.3)$$

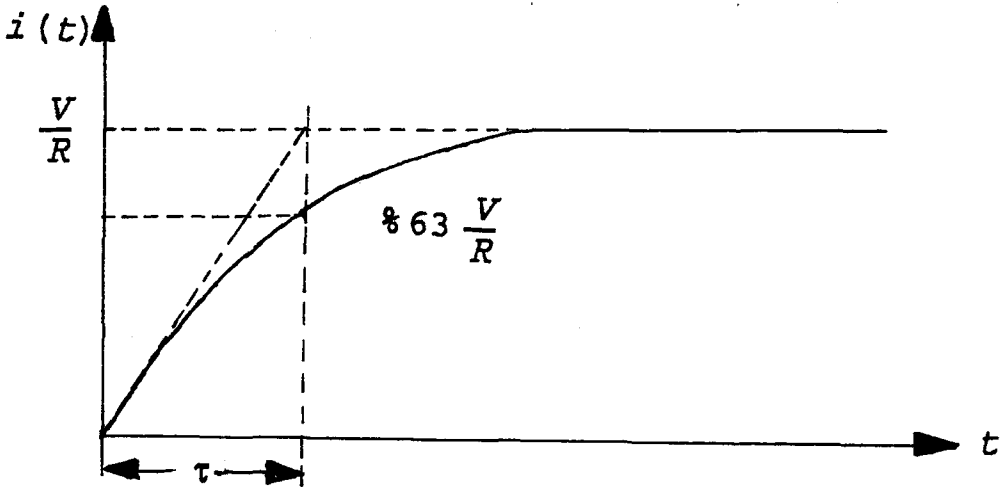
saniye boyutunda, zaman sabiti olarak adlandırılır. Genel olarak,

$$\frac{v_g(t)}{R} = i(t) + \tau \frac{d}{dt} i(t) \quad (3.4)$$

bağıntısı yazılabilir. Basamak giriş işaretine ilişkin sistem yanıtı, (3.4) diferansiyel denkleminin tam çözümünden,

$$i(t) = \frac{V_g}{R} (1 - e^{-\frac{t}{\tau}}) \quad (3.5)$$

şeklinde elde edilir. Birinci dereceden sistemlerin birim basamak giriş işaretine verdikleri yanıt, şekil 3.2. görülmektedir.



Şekil 3.2. Birinci dereceden sistemlerin basamak giriş işaretine verdikleri yanıt.

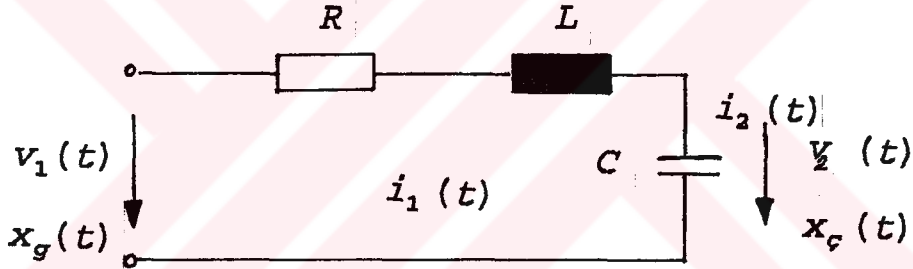
Birinci dereceden sistemin s-domeni transfer fonksiyonu

$$F(s) = \frac{K_P}{1 + s\tau} \quad (3.6)$$

şeklinde tanımlanır.

3.2.İKİNCİ DERECEDEKİ SİSTEMLERİN GEÇİCİ REJİM DAVRANIŞ ÖLÇÜTLERİ

İkinci dereceden sisteme örnek olarak aşağıdaki RLC seri devresi ele alınabilir.



Şekil 3.3. İkinci dereceden sisteme örnek olarak, RLC seri devresi

Sistemin giriş çıkış işaretleri gerilim boyutunda ele alınıp, incelendiğinde, sistemin diferansiyel denklemi ($i_2 = 0$ koşulu altında)

$$x_g(t) = v_1(t) = R \cdot i_1(t) + L \frac{d}{dt} i_1(t) + \frac{1}{C} \int_0^t i_1(t) dt \quad (3.7)$$

şeklinde elde edilir.

Çıkış büyüklüğü olarak, C kondansatörü üzerindeki gerilim alınırsa,

$$x_c(t) = v_2(t) = \frac{1}{C} \int_0^t i_1(t) dt \quad (3.8)$$

bağıntısına ulaşılır. Buradan

$$\frac{d}{dt} x_c(t) = \frac{d}{dt} v_2(t) = \frac{i_1(t)}{C} \quad (3.9)$$

$$i_1(t) = C \frac{d}{dt} v_2(t) = C \frac{d}{dt} x_c(t) \quad (3.10)$$

bağıntıları elde edilip, (3.7)'de yerine yazıldığında,

$$x_c(t) = v_1(t) = RC \frac{d}{dt} v_2(t) + LC \frac{d^2}{dt^2} v_2(t) + v_2(t) \quad (3.11)$$

şeklinde ikinci dereceden diferansiyel denklem elde edilir. Burada

$$LC = \tau^2 \quad (3.12)$$

olarak tanımlandığında ikinci dereceden sistemin zaman sabiti olarak

$$\tau = \sqrt{LC} \quad (3.13)$$

bağıntısı tanımlanır.

$$RC = 2\xi\tau \quad (3.14)$$

biçiminde tanımlandığında, sönüm faktörü,

$$\xi = \frac{R}{2} \sqrt{\frac{C}{L}} \quad (3.15)$$

bağıntısı ile belirlenen, boyutsuz bir büyüklüktür.

Böylece ikinci dereceden sistemin diferansiyel denklemi

$$\frac{d^2}{dt^2}x_c(t)\tau^2 + \frac{d}{dt}x_c(t)2\xi\tau + x_c(t) = x_g(t) \quad (3.16)$$

şeklinde elde edilir. Bu durumda $t \rightarrow \infty$ için,

$$k_p = \frac{x_c(t)}{X_g} \quad (3.17)$$

sistem kazancı olarak tanımlanırsa, (3.16) denkleminin basamak giriş için

$$x_g(t) = \begin{cases} 0 & t < 0 \\ X_g & t \geq 0 \end{cases} \quad (3.18)$$

tam çözümü,

$$x_c(t) = X_g k_p [1 - e^{-(\frac{\xi}{\tau})t} (\cos(\frac{t}{\tau}\sqrt{1-\xi^2}) + \frac{\xi}{\sqrt{1-\xi^2}} \sin(\frac{t}{\tau}\sqrt{1-\xi^2}))] \quad (3.19)$$

$$x_c(t) = X_g k_p \left[1 - \frac{e^{-\left(\frac{\xi}{\tau}\right)t}}{\sqrt{1-\xi^2}} \sin\left(\frac{t}{\tau} \sqrt{1-\xi^2} + \arccos \xi\right) \right] \quad (3.20)$$

bağıntısı bulunur. Çıkış ifadesi düzenlendiğinde ise,

$$x_c(t) = X_g k_p \cdot \left[1 - e^{-\left(\frac{\xi}{\tau}\right)t} \cdot \left(\cos \omega t + \frac{\xi}{\omega \tau} \sin \omega t \right) \right] \quad (3.21)$$

$$x_c(t) = X_g k_p \cdot \left[1 - \frac{e^{-\left(\frac{\xi}{\tau}\right)t}}{\omega \tau} \sin(\omega t + \arccos \xi) \right] \quad (3.22)$$

bağıntısı elde edilir. Burada sistemin doğal açısal frekansı olarak

$$\omega_n = \frac{\sqrt{1-\xi^2}}{\tau} \quad (3.23)$$

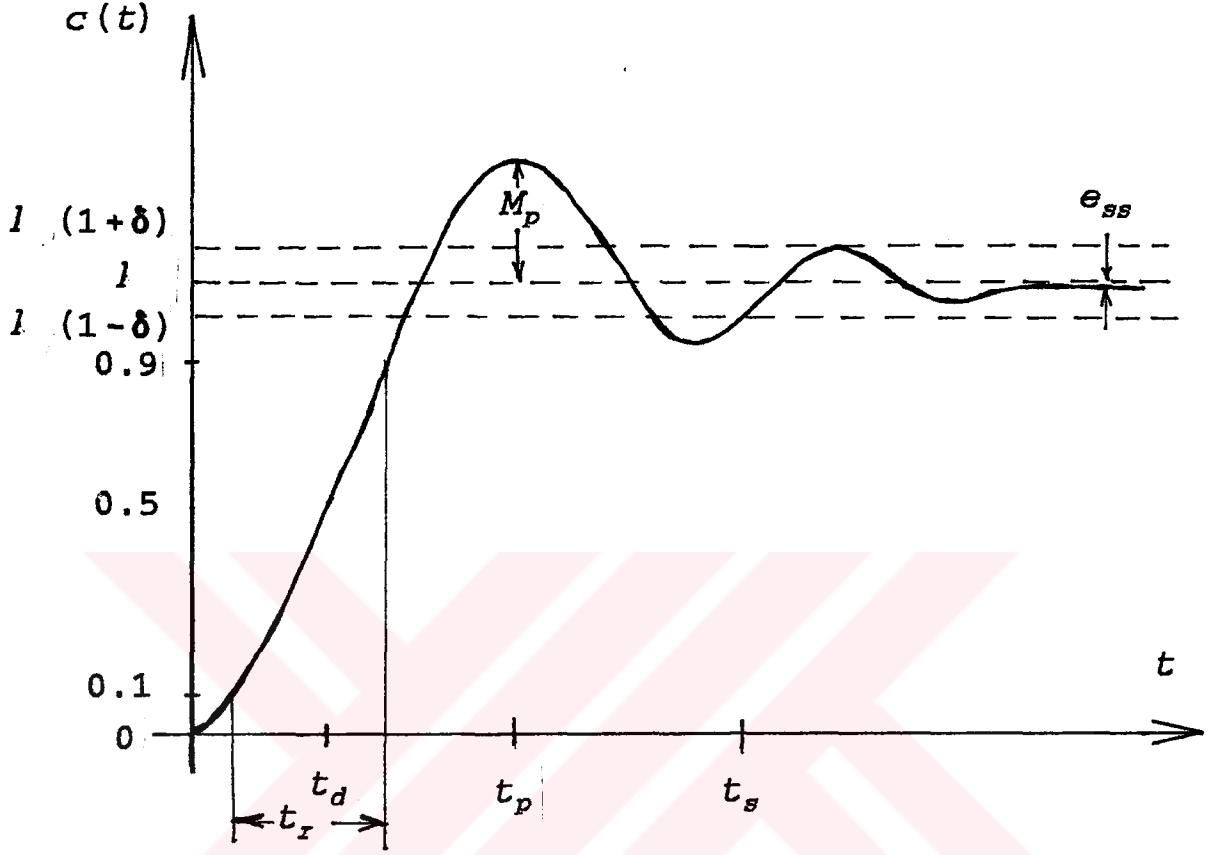
şeklinde tanımlanır.

s-domeni transfer fonksiyonu ise,

$$F(s) = \frac{X_c(s)}{X_g(s)} = \frac{k_p \cdot \omega_n^2}{s^2 + 2\xi \omega_n s + \omega_n^2} \quad (3.24)$$

bağıntısı ile elde edilir.

Birim basamak giriş için ikinci dereceden bir sistemin verdiği yanıt ve ölçütleri, sönüm faktörü $0 < \xi < +1$ durumu için, şekil 3.3.'de gösterilmiştir.



Şekil 3.4. Birim basamak giriş işareti için, ikinci dereceden sistemin yanıtı ve ölçütleri

Bu eğride kontrol açısından gerekli ölçütler aşağıdaki gibi tanımlanabilirler.

Gecikme Süresi (t_d) : Sistem yanıtının, varış değerinin %50'sine ulaşma süresi.

Yükselme Süresi (t_r) : Sistem yanıtının, varış değerinin %10'dan %90'ına ulaşma süresi. (İkinci bir tanım olarak da, t_d noktasında yanıt eğrisine çizilen teğetinin, 0 ile 1 arasında kalan süresi.

Tepe Süresi (t_p) : Yanıt eğrisinin maksimum değerine ulaşma süresi.

Aşım (PO) : Yanıt eğrisinin maksimum değeri ile varış değeri arasındaki farkın, yüzde türünden değeri.

$$PO = \%M_p = \frac{\text{Maksimum değer} - \text{varış değeri}}{\text{varış değeri}} . 100 \quad (3.25)$$

$$PO = \%M_p = \frac{c(t_p) - c(\infty)}{c(\infty)} . 100 \quad (3.26)$$

Yerleşme Süresi (t_s) : Yanıt eğrisinin, varış değerinin belli bir $\% \delta$ sınır değerinin içine bir daha hiç çıkmamak üzere girdiği ilk süredir.

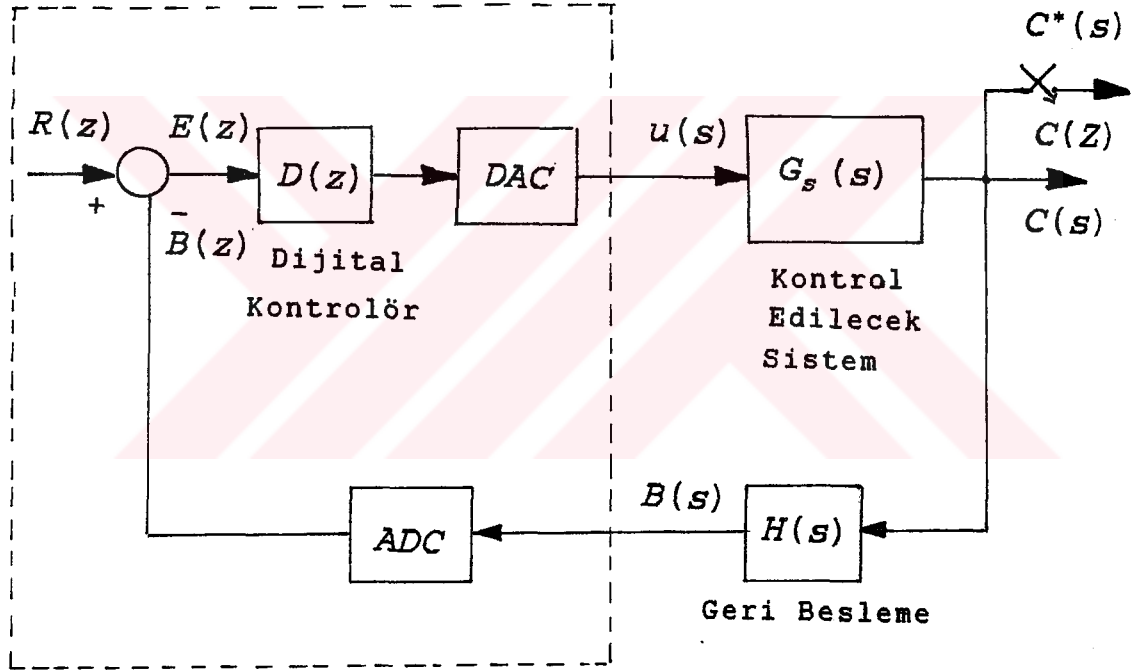
Kararlı Hal Hatası (e_{ss}) : $t \rightarrow \infty$ için yanıt eğrisi ile varış değeri arasındaki fark.

BÖLÜM 4

KAPALI ÇEVİRİM DİJİTAL KONTROL SİSTEMLERİ

4.1 KAPALI ÇEVİRİM DİJİTAL KONTROL SİSTEMİ

Kapalı çevrim dijital kontrol sisteminin kontrola yönelik blok diyagramı aşağıdaki gibidir.



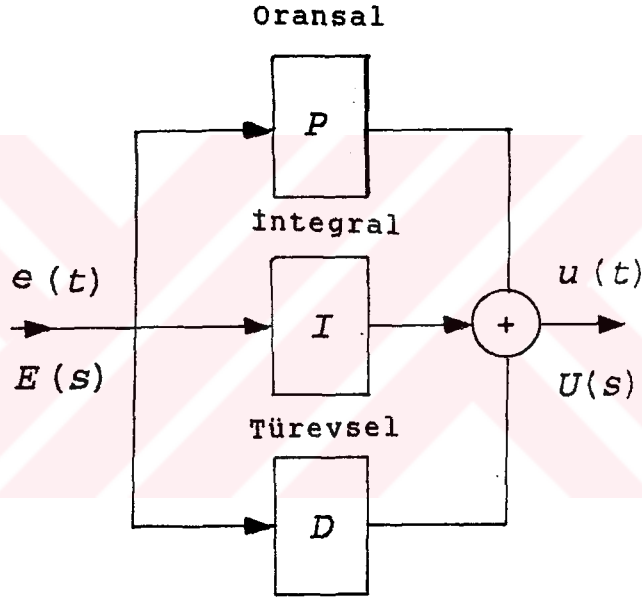
Mikrokontrolör veya (PC)

Şekil 4.1 Kapalı çevrim dijital kontrol sistemi blok diyagramı

Yukarıdaki blok diyagramda, kesikli çizgi içinde gösterilen bloklar dijital kontrolör kartı üzerinde yer alabilir. Ya da bilgisayar ve giriş-çıkış birimleri yardımı ile oluşturulabilir.

4.2. UYGULAMADA KULLANILAN DİJİTAL PID KONTROLÖRÜNÜN t , s VE z TANIM BÖLGELERİNDEKİ MATEMATİKSEL BAĞINTILARI

Uygulamada kullanılan PID kontrolörü, oransal(P), integral(I) ve türevsel(D) kontrol yapan blokların paralel yapıda bağlandıkları varsayılararak oluşturulmuştur. Dijital PID yapısı oluşturulurken, kontrolörün zaman (t) domeni bağıntısından yola çıkılarak, s domeni ve ardından z domeni bağıntıları elde edilmiştir. z domeni bağıntıları doğrudan programlama yöntemi ile turbo PASCAL dilinde programlanıp, "Yeni Global Arama Yöntemi" programı içine katılanmıştır.



Şekil 4.2. Paralel yapıdaki PID kontrolörü blok diyagramı

Paralel yapıdaki PID kontrolörünün, zaman (t) tanım bölgesi içinde aşağıdaki bağıntıyı gerçeklediği varsayılır.

$$u(t) = K_p \cdot e(t) + K_I \cdot \int_0^t e(t) \cdot dt + K_D \cdot \frac{d}{dt} e(t) \quad (4.1)$$

Burada Laplace dönüşümü ile s tanım bölgesi için aynı bağıntı, ilk koşulları sıfır alarak,

$$U(s) = K_P \cdot E(s) + \frac{K_I}{s} \cdot E(s) + K_D \cdot s \cdot E(s) \quad (4.2)$$

şeklinde elde edilir. Buradan dijital kontrolörün s domen transfer fonksiyonu,

$$D(s) = \frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D \cdot s$$

$$D(s) = \frac{K_D \cdot s^2 + K_P \cdot s + K_I}{s} \quad (4.3)$$

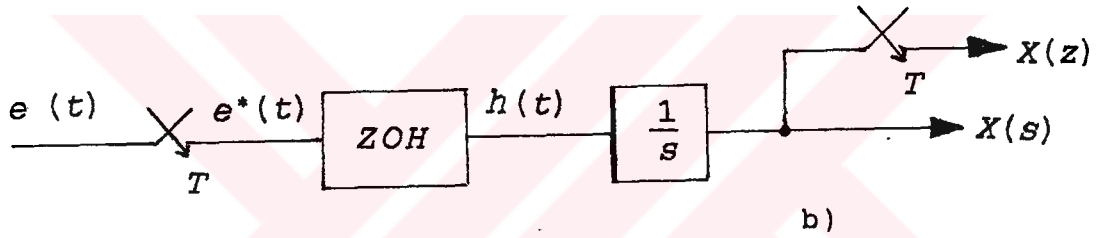
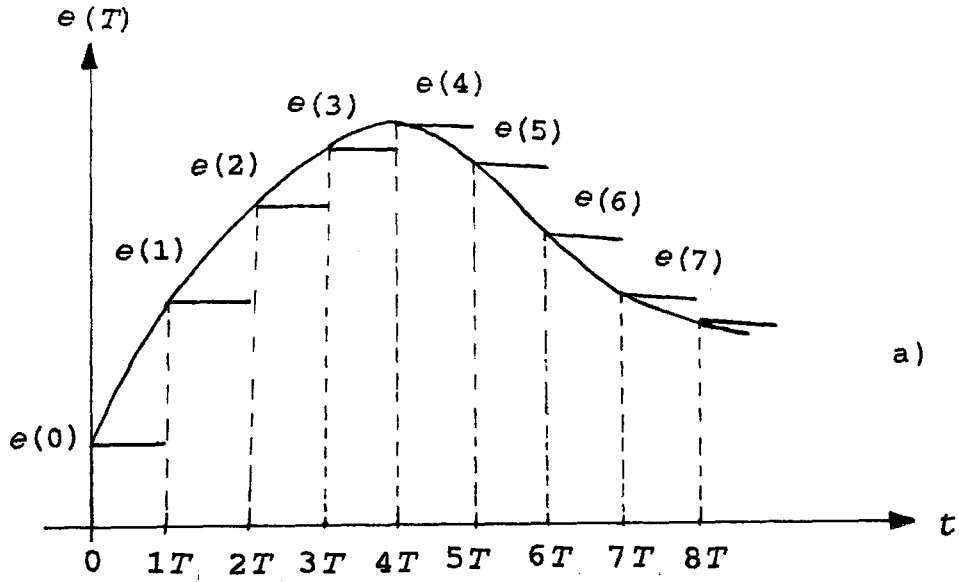
olarak bulunur.

PID kontrolörünün z domenindeki bağıntılarını elde etmek için önce integral alıcı blok incelenecektir. Burada hata işareti $e(t)$ 'ye basamak biçimli yaklaşımda bulunulup, ileri yol integrasyonu uygulanır.[8][9]

$$\frac{X(z)}{E(z)} = Z\left[\frac{1 - e^{-Ts}}{s} \cdot \frac{1}{s}\right]$$

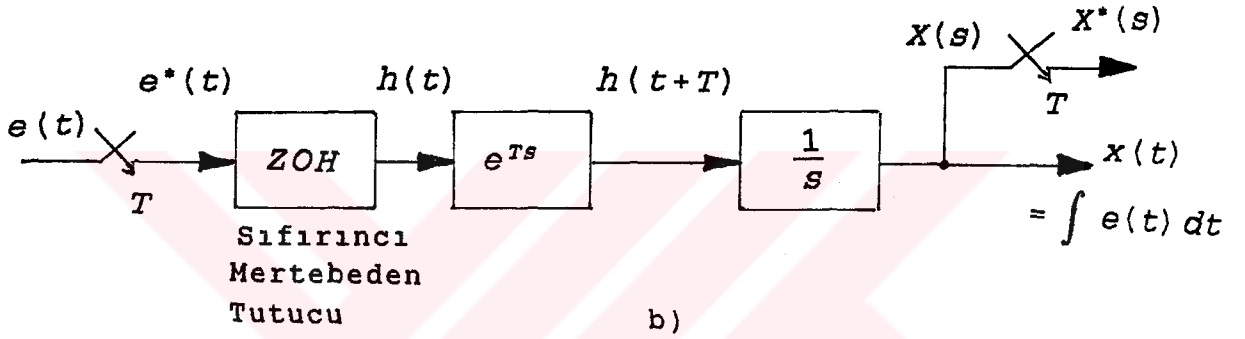
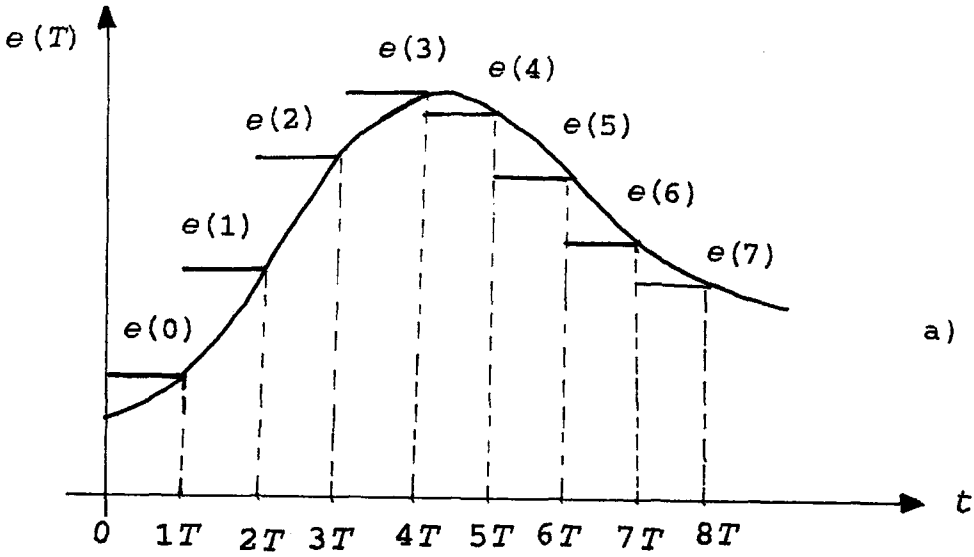
$$\frac{X(z)}{E(z)} = \frac{T}{(z - 1)} \quad (4.4)$$

Şeklinde ileri yol integratörünün transfer fonksiyonu elde edilir



Şekil 4.3 a) Hata işareti $e(t)$ 'ye basamak biçimli yaklaşım. b) ileri yol integrasyonu blok diyagramı

Ayrık integrasyon işlemi ÖNGÖRÜ operatörü kullanılarak da gerçekleştirilebilir. Bilindiği gibi öngörü operatörü fiziksel olarak gerçekleştirilemeyen bir yapıdır. Ancak matematiksel bağıntılara katkılандığında elde edilen ifadeler, fiziksel olarak gerçekleştirilebilir yapıda olabilir.



Şekil 4.4 a) Hata işareti $e(t)$ 'ye, öngörü operatörü ile basamak biçimli yaklaşım b) Öngörü operatörü ile integrasyonun blok diyagramı

Öngörü operatörü ile ayırık integrasyon yapısının transfer fonksiyonu,

$$\frac{X(z)}{E(z)} = Z \left[\frac{1 - e^{-Ts}}{s} \cdot e^{Ts} \cdot \frac{1}{s} \right]$$

$$\frac{X(z)}{E(z)} = \frac{T \cdot z}{(z - 1)} \quad (4.5)$$

şeklinde elde edilebilir. Görüldüğü gibi (4.5) bağıntısı fiziksel olarak gerçekleştirilebilir bir yapıdadır.

Ayrık integrasyon işlemini YAMUK yöntemi ile yaparsak, yamuk alanı ΔA 'nın 0 - T arası değeri,

$$\Delta A = \frac{e(0) + e(T)}{2} \cdot T$$

$$\Delta A = \frac{1}{2} [e(0) \cdot T + e(T) \cdot T] \quad (4.6)$$

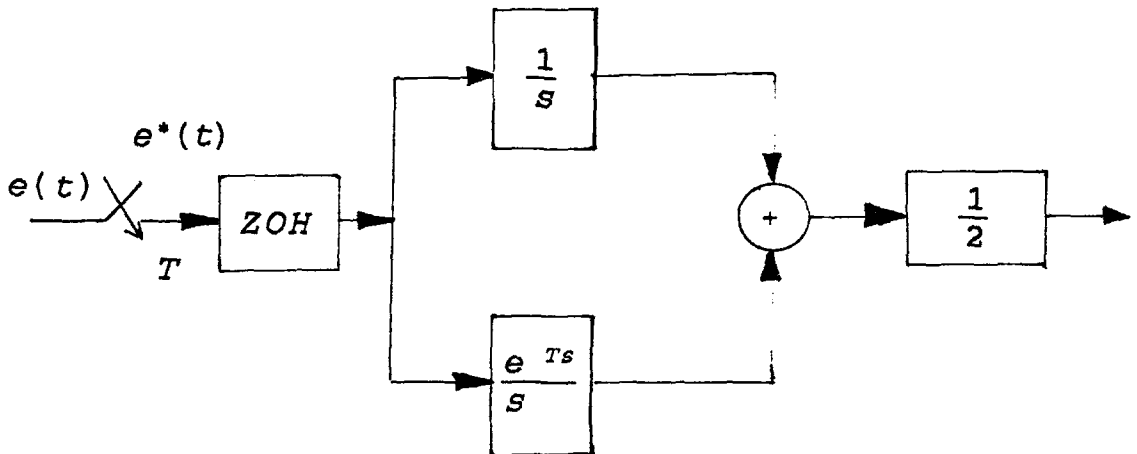
olarak bulunur. Burada

$$\Delta A_I = e(0) \cdot T \quad (4.7)$$

alanı, ileri yol integrasyon yöntemi ile

$$\Delta A_{II} = e(T) \cdot T \quad (4.8)$$

alanı ise öngörü operatörü ile integrasyon yöntemiyle hesaplanır.



Şekil 4.5 Yamuk yöntemi ile integrasyonun blok diyagramı

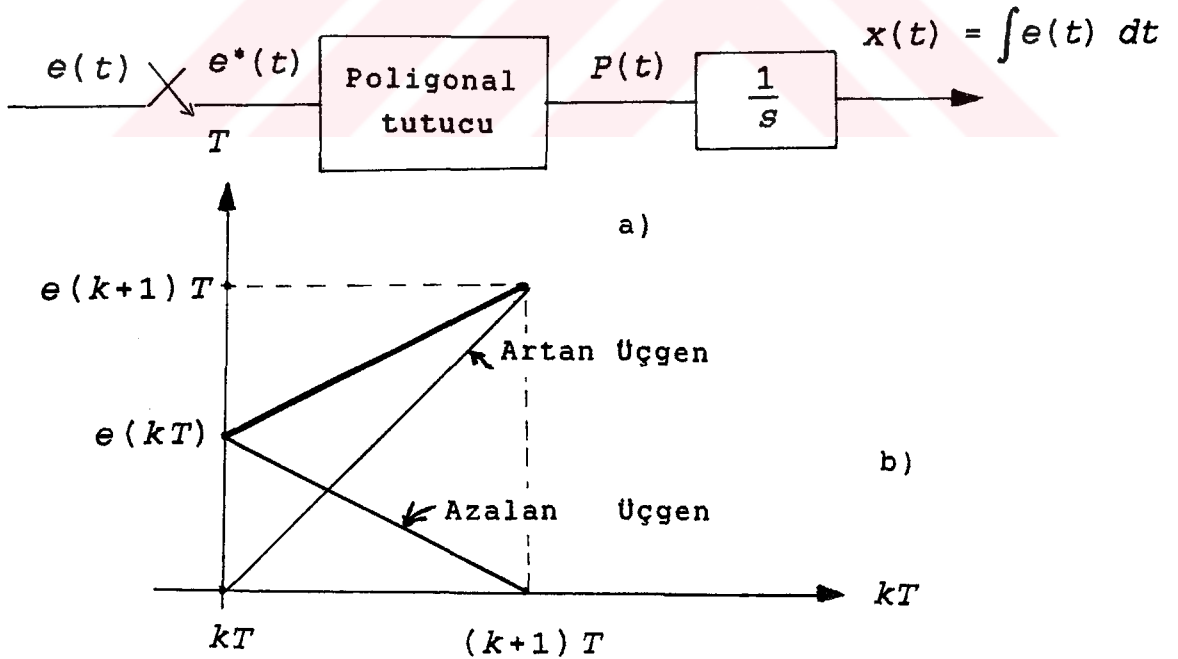
$$\frac{U(s)}{E(s)} = \frac{1}{2} \left[\frac{1 - e^{-Ts}}{s} \right] \left[\frac{1}{s} + \frac{e^{-Ts}}{s} \right]$$

Yamuk yöntemi ile integral alan yapının toplam transfer fonksiyonu,

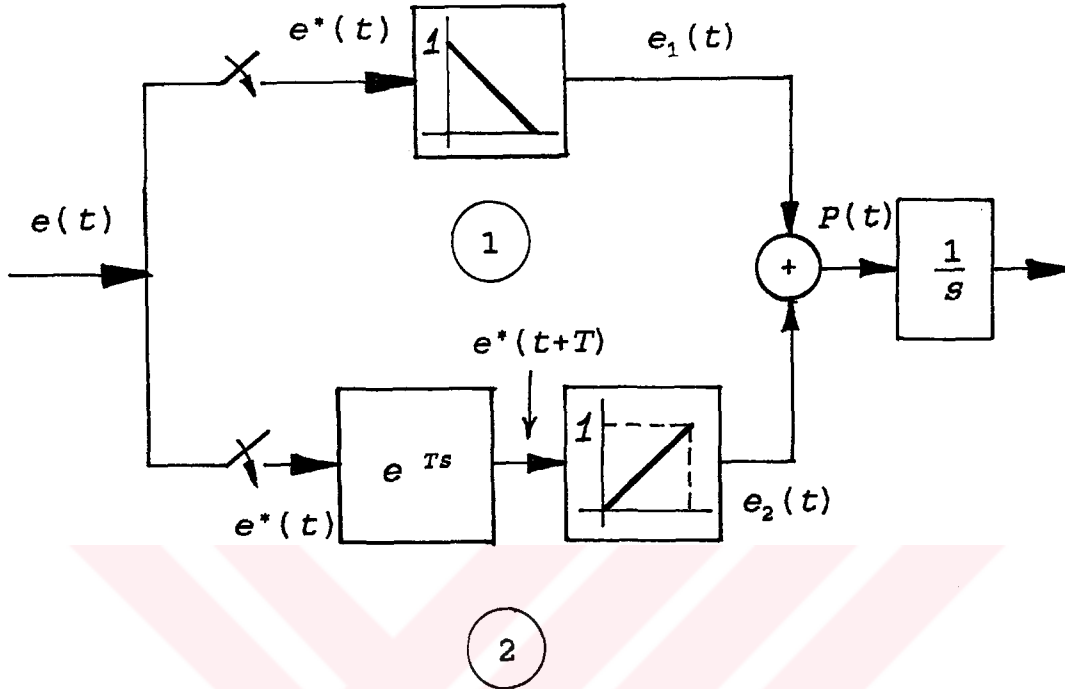
$$\frac{U(s)}{E(s)} = \frac{T(z + 1)}{2(z - 1)} \quad (4.9)$$

şeklinde elde edilir. Hata işaretine yamuk yöntemine göre yaklaşımda bulunduğu için, ileri yol integrasyonu veya öngörü operatörü ile integrasyon yöntemlerine göre daha doğru sonuç verir.

Ayrık integrasyonun işleminde bir diğer yöntem de tutucu'nun, sıfıncı mertebe değil, birinci mertebe yapılmasıdır.



Şekil 4.6. a) Poligonal tutuculu integratörün blok diyagramı. b) Poligonal tutucunun grafik yapısı



Şekil 4.6. c) Poligonal tutucunun blok diyagramı

$e(kT)$ ve $e[(k+1)T]$ örnekleme anlarına ilişkin işaretlerden, azalan ve artan üçgenler yaklaşımı ile poligonal işaret elde edilebilir. Buna göre azalan üçgen yapısı için transfer fonksiyonu,

$$e_1(t) = e^*(t) \cdot \left[\frac{-t-T}{T} \cdot u(t) + \frac{t-T}{T} \cdot u(t-T) \right]$$

$$\frac{E_1(s)}{E^*(s)} = -\frac{1}{s^2 T} + \frac{1}{s} + \frac{e^{-Ts}}{s^2 T} \quad (4.10)$$

şeklinde elde edilir.

Artan üçgen yapısı için transfer fonksiyonu,

$$e_2(t) = e^*(t+T) \left[\frac{t}{T} u(t) - \left[\frac{t-T}{T} + 1 \right] u(t-T) \right]$$

$$\frac{E_2(s)}{E^*(s)} = e^{Ts} \cdot \left[\frac{1}{s^2 T} - \frac{e^{-Ts}}{s^2 T} - \frac{e^{-Ts}}{s} \right]$$

$$\frac{E_2(s)}{E^*(s)} = \frac{e^{Ts}}{s^2 T} - \frac{1}{s^2 T} - \frac{1}{s} \quad (4.11)$$

şeklinde elde edilir.

(4.10) ve (4.11) bağıntıları toplanıp, poligonal tutucunun transfer fonksiyonu,

$$\frac{P(s)}{E^*(s)} = \frac{e^{Ts} + e^{-Ts} - 2}{s^2 T} \quad (4.12)$$

olarak bulunur.

Poligonal tutucunun çıkışındaki işaret integratöre uygulanıp, tüm yapının transfer fonksiyonu aşağıdaki gibi elde edilir.

$$\frac{X(z)}{E(z)} = K_I \frac{T \cdot (z+1)}{2 \cdot (z-1)} \quad (4.13)$$

Görüldüğü gibi yamuk yöntemi ile işlem yapan integratör ve poligonal tutuculu integratör aynı ayrık transfer fonksiyonunu vermektedir.

Ayrık türev ifadesi ise,

$$\left. \frac{de(t)}{dt} \right|_{t=kT} \approx \frac{e(kT) - e(k-1)T}{T} \quad (4.14)$$

bağıntısı ile hesaplanır. Türev alıcı bloğun transfer fonksiyonu olarak

$$\frac{X(z)}{E(z)} = K_D \frac{(z-1)}{T \cdot z} \quad (4.15)$$

bağıntısı elde edilir.

Böylece PID kontrolörünün z tanım bölgesindeki transfer fonksiyonu olarak da

$$\frac{U(z)}{E(z)} = K_P + K_I \frac{T \cdot (z+1)}{2 \cdot (z-1)} + K_D \frac{(z-1)}{T \cdot z} \quad (4.16)$$

bağıntısı elde edilir. Buradan direk programlama yöntemi ile

$$\frac{U(z)}{E(z)} = \frac{K_P T z^2 (z-1) + T z K_I T (z+1) + 2 (z-1) K_D (z-1)}{2 (z-1) T z}$$

$$\frac{U(z)}{E(z)} = \frac{(2 K_D + 2 K_P T + K_I T^2) z^2 + (K_I T^2 - 2 K_P T - 4 K_D) z + 2 K_D}{2 T z^2 - 2 T z} \quad (4.17)$$

(4.17) bağıntısının pay ve paydasını z^{-2} ile çarparak

$$\frac{U(z)}{E(z)} = \frac{(2K_D + 2K_P T + K_I T^2) + (K_I T^2 - 2K_P T - 4K_D) z^{-1} + 2K_D z^{-2}}{2T - 2Tz^{-1}} \quad (4.18)$$

ifadesine ulaşılır. Buradan da

$$\begin{aligned} & (2K_D + 2K_P T + K_I T^2) E(z) + (K_I T^2 - 2K_P T - 4K_D) z^{-1} E(z) + 2K_D z^{-2} E(z) \\ & = 2TU(z) - 2Tz^{-1}U(z) \end{aligned} \quad (4.19)$$

bağıntısı elde edilir. Burada

$$E(z) = e_1 \quad : \quad T. \text{ örneklemeye ilişkin hata işareti,}$$

$$z^{-1} \cdot E(z) = e_2 \quad : \quad (T-1). \text{ (Bir önceki) örneklemeye ilişkin hata,}$$

$$z^{-2} \cdot E(z) = e_3 \quad : \quad (T-2). \text{ (İki önceki) örneklemeye ilişkin hata işareti,}$$

$$U(z) = uk_1 \quad : \quad T. \text{ örneklemeye ilişkin kontrol işareti,}$$

$$z^{-1} \cdot U(z) = uk_2 \quad : \quad (T-1). \text{ örneklemeye ilişkin kontrol işaretleri olmak üzere ve}$$

$$P_1 = \frac{2K_D + 2K_P T + K_I T^2}{2T}$$

$$P_2 = \frac{K_I T^2 - 2K_P T - 4K_D}{2T}$$

$$P_3 = \frac{K_d}{T}$$

tanımlamaları ile, her T örnekleme anı için PID algoritmasının oluşturduğu kontrol işareti,

$$uk_1 = P_1 \cdot e_1 + P_2 \cdot e_2 + P_3 \cdot e_3 + uk_2 \quad (4.20)$$

bağıntısı ile edle edilir.

(4.20) bağıntısı uyarınca elde edilen PID kontrolör algoritması, EK-C'de sunulan TURBO PASCAL dilindeki programda gerçekleştirilmiştir. Bu program yardımı ile girilen verilen K_p , K_i , K_d , PID kontrolörü katsayılarına bağlı olarak, sistem performansı belirlenip, amaç ölçütü hesaplanmaktadır.

BÖLÜM 5

KONTROL EDİLECEK SİSTEM OLARAK ASENKRON MOTOR

Rotoru sargılı (Bilezikli) asenkron makinada da diğer tüm elektrik makinalarında olduğu gibi, bir duran kısım (Stator), bir de dönen kısım (Rotor) vardır. Rotoru sargılı asenkron makinalarda stator sargıları bir veya üç fazlı olarak üretilirler. Statoru tek fazlı asenkron makinalar düşük güçlü tahrik düzeneklerinde kullanılır. Tez de modeli verilen sistem, üç fazlı rotoru sargılı asenkron makinedir. Bu makinanın rotoru da statoru gibi üç fazlı sargılardan oluşmuştur. Asenkron makinanın modellenmesindeki temel amaçların başında makina bir takım fiziksel büyüklükler uygulamadan önce makina modelinden yola çıkılarak oluşturulan simülasyon programları yardımı ile oluşabilecek fiziksel durumların, bilgisayar yardımı ile ekranda gözlenmesi gelir. Böylece tasarlanan sistemde oluşabilecek sorunlar önceden belirlenebilir ve önlemleri alınabilir. Yine makina modelinden yararlanarak, uygulanacak kontrol işaretinin türetilmesi de sağlanabilir. Bu nedenle makinanın iyi bir şekilde modellenmesi oldukça önemlidir. Geliştirilen modelin, makinanın fiziksel davranışını iyi yansıtabilmesi ve kontrol algoritmalarına uygulanabilecek türden, basit yapıda olması arzu edilir. Bu nedenle modelin minimum düzeyde varsayım ile oluşturulması gerekir. Bu varsayımlar modeli, gerçek fiziksel sistemin yapısından uzaklaştırmamalıdır. Verilecek olan rotoru sargılı asenkron makina modeli, sistemin hem geçici hem de sürekli rejiminin incelenmesini sağlar. Ayrıca giriş işaretlerinin sinüsoidal yapıdan farklı olması durumlarında da sistemin analizini sağlar.

Modelin elde edilmesinde yapılan varsayımlar:

a) Stator sargıları, stator çevresine düzgün olarak yayılmıştır. Hava aralığında oluşan akı sinüsoidal biçimdedir.

b) Üç fazlı stator sargıları çevreye, 120° 'lik elektriksel açı ile düzgün biçimde yayılmışlardır.

c) Doyma ve dış etkileri ihmal edilmiştir.

d) Magnetik kısımların geçirgenliği sonsuz varsayılmıştır.

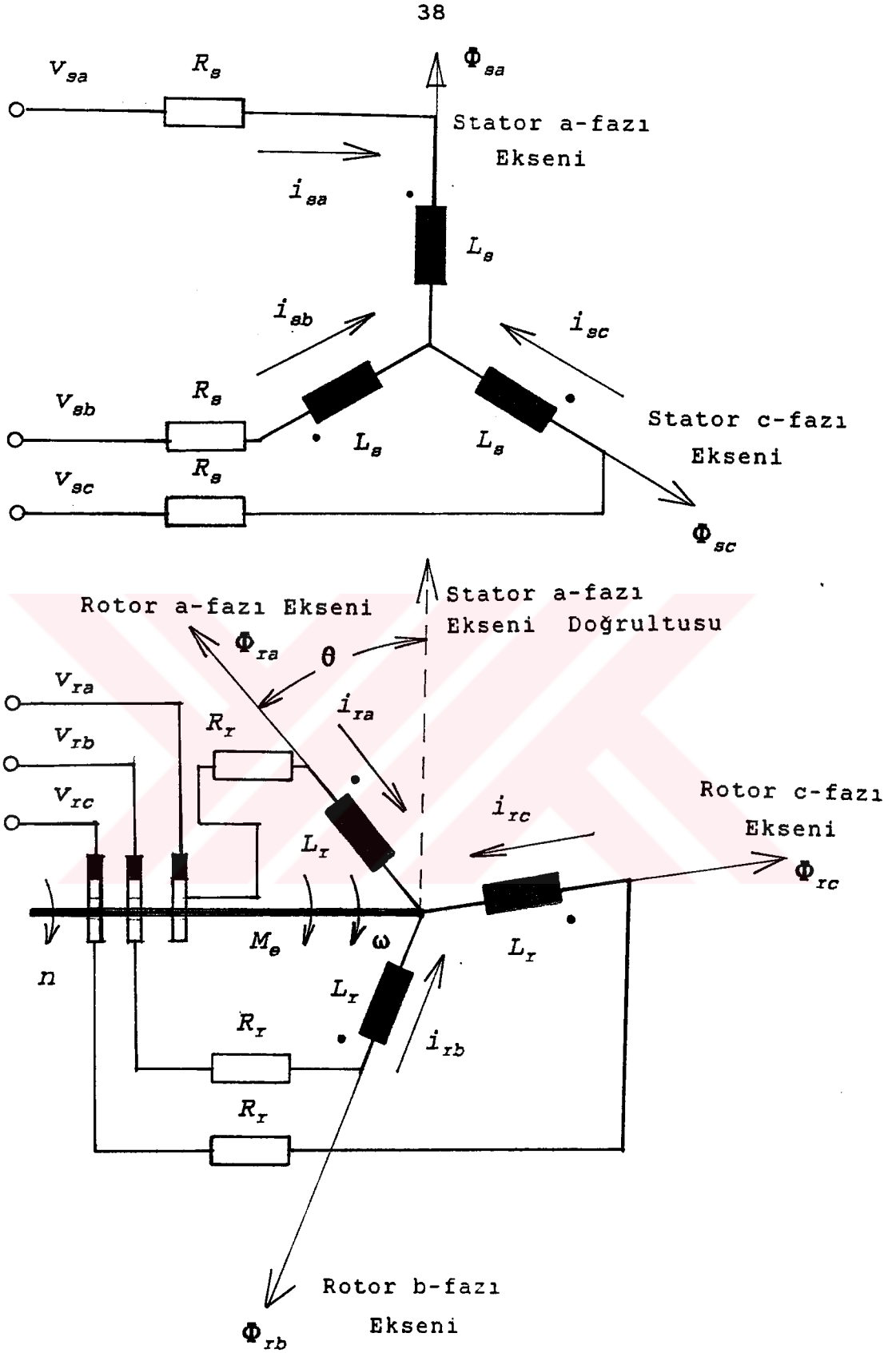
e) Histerezis ve fuko kayıpları ihmal edilmiştir.

f) Akım yığılması (deri olayı) ihmal edilmiştir.

g) Rotor sargıları, rotor eksenine göre simetrik yayılmışlardır.

h) Dirençler ve endüktansların sıcaklıktan bağımsız oldukları varsayılmıştır.

Bu varsayımlar ışığı altında statoru ve rotoru üç fazlı asenkron makinanın bağlantı şeması Şekil 5-1' de verilmiştir.



Şekil 5.1. Stator ve Rotoru üç fazlı (Bilezikli) asenkron makinanın a-b-c eksen takımındaki eşdeğer devresi

5.1. MATEMATİKSEL MODELİN ELDE EDİLMESİ:

Rotoru sargılı asenkron makinanın, üç'er fazlı stator ve rotoruna ilişkin toplam 6 adet elektriksel denklemi ile mekanik yana ilişkin 1 adet denklemi olup, toplam 7 adet denklemi vardır. Söz konusu modelde yer alan stator ve rotora ilişkin direnç ve endüktans matrisleri $[R_s]$, $[R_r]$, $[L_s]$

$$[R_s] = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \quad [R_r] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix} \quad (5.1)$$

$$[L_{sx}(\theta)] = \begin{bmatrix} [L_s] & [M_{sx}] \\ [M_{rx}] & [L_r] \end{bmatrix}, \quad [M_{sx}] = [M_{rx}] \quad (5.2)$$

$$[L_s] = \begin{bmatrix} L_s & M_{ss} & M_{ss} \\ M_{ss} & L_s & M_{ss} \\ M_{ss} & M_{ss} & L_s \end{bmatrix}, \quad [L_r] = \begin{bmatrix} L_r & M_{rr} & M_{rr} \\ M_{rr} & L_r & M_{rr} \\ M_{rr} & M_{rr} & L_r \end{bmatrix} \quad (5.3)$$

$$[M_{sx}] = M_{sx} \cdot \begin{bmatrix} \cos(p\theta) & \cos(p\theta + \frac{2\pi}{3}) & \cos(p\theta - \frac{2\pi}{3}) \\ \cos(p\theta - \frac{2\pi}{3}) & \cos(p\theta) & \cos(p\theta + \frac{2\pi}{3}) \\ \cos(p\theta + \frac{2\pi}{3}) & \cos(p\theta - \frac{2\pi}{3}) & \cos(p\theta) \end{bmatrix} \quad (5.4)$$

biçiminde tanımlanabilir. Akım ve gerilim ifadeleri

$$[I_s] = \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix}, \quad [V_s] = \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix}, \quad [I_r] = \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix}, \quad [V_r] = \begin{bmatrix} v_{ar} \\ v_{br} \\ v_{cr} \end{bmatrix} \quad (5.5)$$

Akı ifadeleri,

$$[\Phi_s] = \begin{bmatrix} \Phi_{as} \\ \Phi_{bs} \\ \Phi_{cs} \end{bmatrix}, \quad [\Phi_r] = \begin{bmatrix} \Phi_{ar} \\ \Phi_{br} \\ \Phi_{cr} \end{bmatrix} \quad (5.6)$$

Makinanın Akı-Akım bağıntıları,

$$[\Phi_s] = [L_s] \cdot [I_s] + [M_{sr}] \cdot [I_r] \quad (5.7)$$

$$[\Phi_r] = [L_r] \cdot [I_r] + [M_{rs}] \cdot [I_s] \quad (5.8)$$

biçiminde verilirse, rotoru sargılı asenkron makinanın genel matematiksel modeli, statora ilişkin olarak,

$$[V_s] = [R_s] \cdot [I_s] + \frac{d}{dt} [\Phi_s] \quad (5.9a)$$

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} = \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_s & 0 \\ 0 & 0 & R_s \end{bmatrix} \cdot \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \Phi_{as} \\ \Phi_{bs} \\ \Phi_{cs} \end{bmatrix} \quad (5.9b)$$

rotor yanına ilişkin olarak da,

$$[V_r] = [R_r] \cdot [I_r] + \frac{d}{dt} [\Phi_r] \quad (5.10a)$$

$$\begin{bmatrix} v_{ar} \\ v_{br} \\ v_{cr} \end{bmatrix} = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix} \cdot \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \Phi_{ar} \\ \Phi_{br} \\ \Phi_{cr} \end{bmatrix} \quad (5.10b)$$

momente ilişkin olarak ise

$$m_e = \frac{1}{2} \left[[I_s]^T \quad [I_r]^T \right] \cdot \begin{bmatrix} 0 & \frac{\partial M_{sr}}{\partial \theta} \\ \frac{\partial M_{rs}}{\partial \theta} & 0 \end{bmatrix} \cdot \begin{bmatrix} [I_s] \\ [I_r] \end{bmatrix} = m_y + J \cdot \frac{d\omega}{dt} \quad (5.11)$$

bağıntısı yazılabilir.

Yukarıdaki denklemlerde, 's' indisliiler statora, 'r' indisliiler ise rotora ilişkin büyüklüklerdir.

R_s : Stator bir faz sargı direnci

L_s : Stator bir faz özendüklemi

M_{ss} : Stator iki faz sargısı arası karşıt endüklem

R_r : Rotor bir faz sargı direnci

L_r : Rotor bir faz sargısı özendüklemi

M_{rr} : Rotor iki faz sargısı arası karşıt endüklem

M_{sr} : Rotor ve Stator sargıları arası karşıt endüklem

θ : Stator 'a' fazı ile rotor 'a' fazı arasındaki konuma ilişkin geometrik açı

a,b,c : Stator ve rotor fazları

p : Makina çift kutup sayısı

ω : Rotorun açısal frekansı

M_y : Yük momentı

Rotoru sargılı asenkron makina için 3 adet statora , 3 adet rotora ve 1 adet de mekanik yana ilişkin olmak üzere

toplam 7 adet diferansiyel denklem elde edilmiştir. Ancak elde edilen model, kontrole elverişli yapıda değildir. Ayrıca rotor ve stator arasındaki karşıt endüklem (M_{sr}) rotor konumunun geometrik açısı ile değişmesi, parametreleri rotor konumuna göre değişken olan bir modeli ortaya çıkarır. Bu sakıncaları ortadan kaldırmak için bir takım dönüşümler geliştirilmiştir. Bu dönüşümler ile karmaşık olan denklem sistemleri yalınlaştırılıp, sabit katsayılı yapıya ulaştırılabilir. Uygulanabilecek dönüşümler içinde en uygun olanı denklem sistemini, dönen koordinat sisteminde oluşturmaktır. Bu şekilde denklem sisteminde yer alan değişken parametreler sabit kaldığı gibi, elde edilen durum büyüklükleri de doğru akım büyüklüklerine dönüşür. Ayrıca bu yapıda modeli temsil eden denklem takımı da oldukça yalınlaşmaktadır. Bu yapı kontrole oldukça uygun bir yapı olup, asenkron makinanın vektörel kontrolunda da bu modelden yararlanılmaktadır.

5.2 ASENKRON MAKİNAYA PARK DÖNÜŞÜMÜNÜN UYGULANMASI VE d-q MODELİ

(5.9a) ve (5.10a) bağıntılarını matrisel yapıda yeniden yazarsak,

$$[V_{sx}] = [R_{sx}] \cdot [I_{sx}] + \frac{d}{dt} [L_{sx}] \cdot [I_{sx}] \quad (5.12)$$

$$\begin{bmatrix} [V_s] \\ [V_r] \end{bmatrix} = \begin{bmatrix} [R_s] & 0 \\ 0 & [R_r] \end{bmatrix} \cdot \begin{bmatrix} [I_s] \\ [I_r] \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} [L_s] & [M_{sr}] \\ [M_{rs}] & [L_r] \end{bmatrix} \cdot \begin{bmatrix} [I_s] \\ [I_r] \end{bmatrix} \quad (5.12a)$$

şeklinde elde edilir. Burada R_{sr} ve L_{sr} matrisleri simetrik olduklarından, bunları ortogonal matrisler yardımı ile sabit katsayılı ve köşegenel hale getirme olanağı vardır. Bu amaçla normalize edilmiş PARK DÖNÜŞÜMÜ kullanılabilir. Park dönüşümü yardımı ile a-b-c koordinat düzleminden,

dönen eksen koordinat düzlemi olan d-q-0 'a doğrudan geçilebilir. Dönüşüm için [11],[12]

$$\sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta_s & \cos(\theta_s - \frac{2\pi}{3}) & \cos(\theta_s + \frac{2\pi}{3}) \\ -\sin \theta_s & \sin(\theta_s - \frac{2\pi}{3}) & \sin(\theta_s + \frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} i_{as} \\ i_{bs} \\ i_{cs} \end{bmatrix} = \begin{bmatrix} i_{ds} \\ i_{qs} \\ i_{0s} \end{bmatrix} \quad (5.13)$$

bağıntısından yararlanılır. Burada

$$\theta_s = \omega_s \cdot t \quad (5.14)$$

olarak verilir. Stator yanı dönüşümü için yukarıda verilen bağıntının benzeri, rotor yanı dönüşümü için de verilebilir.

$$\sqrt{\frac{2}{3}} \begin{bmatrix} \cos \theta_r & \cos(\theta_r - \frac{2\pi}{3}) & \cos(\theta_r + \frac{2\pi}{3}) \\ -\sin \theta_r & \sin(\theta_r - \frac{2\pi}{3}) & \sin(\theta_r + \frac{2\pi}{3}) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} i_{ar} \\ i_{br} \\ i_{cr} \end{bmatrix} = \begin{bmatrix} i_{dr} \\ i_{qr} \\ i_{0r} \end{bmatrix} \quad (5.15)$$

Burada da

$$\theta_r = \omega_r \cdot t \quad (5.16)$$

olarak tanımlanır. Rotoru sargılı asenkron makina simetrik bir sistem olduğundan '0' bileşeni gözönüne alınmayabilir.

(5.9a) ve (5.10a) bağıntılarına PARK dönüşümü uygulanması sonucunda, stator için,

$$\begin{bmatrix} v_{sd} \\ v_{sq} \end{bmatrix} = \begin{bmatrix} R_s & 0 \\ 0 & R_s \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \begin{bmatrix} L_s & 0 \\ 0 & L_s \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \frac{3}{2} M_{sx} \cdot \omega_s \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \omega_s \cdot L_s \cdot \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \frac{3}{2} \cdot M_{sx} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} \quad (5.17)$$

Rotor için de,

$$\begin{bmatrix} v_{rd} \\ v_{rq} \end{bmatrix} = \begin{bmatrix} R_r & 0 \\ 0 & R_r \end{bmatrix} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \begin{bmatrix} L_r & 0 \\ 0 & L_r \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \frac{3}{2} M_{sx} \cdot \omega_r \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \omega_r \cdot L_r \cdot \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \frac{3}{2} \cdot M_{sx} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} \quad (5.18)$$

bağıntıları elde edilir. Endüklenen moment'e ilişkin olarak

$$M_e = \frac{3}{2} \cdot P \cdot M_{sx} \cdot (i_{sq} \cdot i_{rd} - i_{sd} \cdot i_{rq}) \quad (5.19)$$

bağıntısı elde edilir.

Elde edilen d-q modelini daha kullanışlı hale getirmek için rotor büyüklükleri statora indirgenecektir. indirgemedede kullanılacak indirgeme katsayısı $\bar{u}$,

$$\bar{u} = \frac{K_s \cdot N_r}{K_r \cdot N_s} \quad (5.20)$$

şeklinde tanımlanmaktadır. Burada

K_s ve K_r stator ve rotora ilişkin sargı faktörleri,

N_s ve N_r stator ve rotorun bir fazındaki toplam sarım sayısıdır.

indirgeme akımlar için yapıldığında,

$$i'_{rx} = \frac{1}{\tilde{U}} \cdot i_{dx} , \quad i'_{rq} = \frac{1}{\tilde{U}} i_{rq} \quad (5.21)$$

empedanslar için yapıldığında ise

$$L'_r = \tilde{U}^2 \cdot L_r , \quad R'_r = \tilde{U}^2 \cdot R_r , \quad M_m = \tilde{U} \cdot \frac{3}{2} M_{sr} \quad (5.22)$$

eşitlikleri elde edilir. Bu eşitliklerden yola çıkarak elde edilen akı-akım bağıntıları,

$$\Phi_{sd} = L_s \cdot i_{sd} + M_m \cdot i'_{rd} = L_{s\sigma} \cdot i_{sd} + M_m \cdot (i'_{rd} + i_{sd}) \quad (5.23)$$

$$\Phi'_{rd} = L'_r \cdot i'_{rd} + M_m \cdot i_{sd} = L'_{r\sigma} \cdot i'_{rd} + M_m \cdot (i'_{rd} + i_{sd}) \quad (5.24)$$

$$\Phi_{sq} = L_s \cdot i_{sq} + M_m \cdot i'_{rq} = L_{s\sigma} \cdot i_{sq} + M_m \cdot (i'_{rq} + i_{sq}) \quad (5.25)$$

$$\Phi'_{rq} = L'_r \cdot i'_{rq} + M_m \cdot i_{sq} = L'_{r\sigma} \cdot i'_{rq} + M_m \cdot (i'_{rq} + i_{sq}) \quad (5.26)$$

$$\Phi_{md} = M_m \cdot (i'_{rd} + i_{sd}) \quad (5.27)$$

$$\Phi_{mq} = M_m \cdot (i'_{rq} + i_{sq}) \quad (5.28)$$

$$\Phi_{sd} = L_s \cdot i_{sd} + \Phi_{md} \quad (5.29)$$

$$\Phi'_{rd} = L'_r \cdot i'_{rd} + \Phi_{md} \quad (5.30)$$

$$\Phi_{sq} = L_s \cdot i_{sq} + \Phi_{mq} \quad (5.31)$$

$$\Phi'_{rq} = L'_r \cdot i'_{rq} + \Phi_{mq} \quad (5.32)$$

indirgeme katsayısı (5.20) ve indirgenmiş büyüklüklerle (5.21), (5.22) modele ilişkin (5.17), (5.18) ve (5.19) bağıntıları yeniden düzenlenerek,

$$v_{sd} = R_s \cdot i_{sd} - \omega_s \cdot (L_s \cdot i_{sq} + M_m \cdot i'_{rq}) + L_s \frac{d}{dt} i_{sd} + M_m \frac{d}{dt} i'_{rd} \quad (5.33)$$

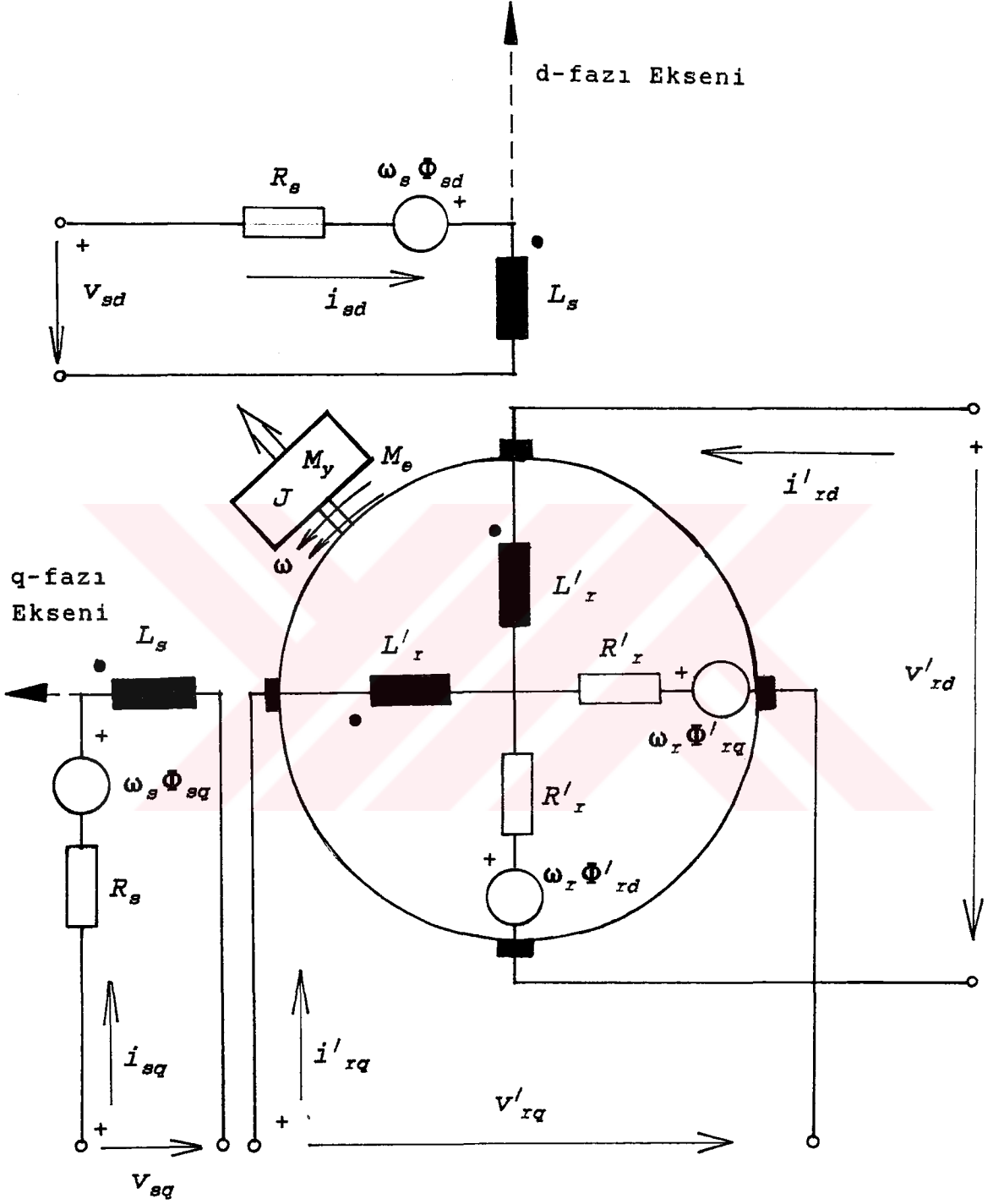
$$v_{sq} = R_s \cdot i_{sq} + \omega_s \cdot (L_s \cdot i_{sd} + M_m \cdot i'_{rd}) + L_s \frac{d}{dt} i_{sq} + M_m \frac{d}{dt} i'_{rq} \quad (5.34)$$

$$v'_{rd} = R'_r \cdot i'_{rd} - \omega_r \cdot (L'_r \cdot i'_{rq} + M_m \cdot i_{sq}) + L'_r \frac{d}{dt} i'_{rd} + M_m \frac{d}{dt} i_{sd} \quad (5.35)$$

$$v'_{rq} = R'_r \cdot i'_{rq} + \omega_r \cdot (L'_r \cdot i'_{rd} + M_m \cdot i_{sd}) + L'_r \frac{d}{dt} i'_{rq} + M_m \frac{d}{dt} i_{sq} \quad (5.36)$$

$$M_e = p \cdot M_m \cdot (i_{sq} \cdot i'_{rd} - i'_{rq} \cdot i_{sd}) \quad (5.37)$$

(5.33) - (5.37) arası bağıntılarla elde edilen modele ilişkin eşdeğer devre Şekil 5.2'de verilmiştir.



Şekil 5.2 Rotoru sargılı asenkron makinanın d-q eksen takımındaki eşdeğer devresi

(5.23)-(5.37) denklemlerinde yer alan durum ve kontrol büyüklükleri, doğru akım ve gerilim biçiminde büyüklüklerdir. d-q eksen takımındaki bu büyüklükleri sürekli sinüzoidal çalışma halinde vektörel olarak yazabiliriz.

Stator yanına ilişkin olarak,

$$\vec{v}_s = v_{sd} + jv_{sq} , \quad \vec{i}_s = i_{sd} + ji_{sq} \quad (5.38)$$

$$\vec{v}_s = R_s \cdot \vec{i}_s + j\omega_s (L_s \cdot \vec{i}_s + M_m \cdot \vec{i}_r') + L_s \frac{d}{dt} \vec{i}_s + M_m \frac{d}{dt} \vec{i}_r' \quad (5.40)$$

Rotor yanına ilişkin olarak da,

$$\vec{v}_r' = v_{rd}' + jv_{rq}' , \quad \vec{i}_r' = i_{rd}' + ji_{rq}' \quad (5.39)$$

$$\vec{v}_r' = R_r' \cdot \vec{i}_r' + j\omega_r (L_r' \cdot \vec{i}_r' + M_m \cdot \vec{i}_s) + L_r' \frac{d}{dt} \vec{i}_r' + M_m \frac{d}{dt} \vec{i}_s \quad (5.41)$$

bağıntıları elde edilir.

Sürekli rejimde, makinanın elektriksel yanına ilişkin olarak (5.40) ve (5.41) denklemleri ve aşağıdaki bağıntılar kullanılarak, rotoru sargılı asenkron makinanın, sürekli rejim modeli ve eşdeğer devresi elde edilebilir.

$$\omega_r = s \cdot \omega_s , \quad L_s = L_{s0} + M_m , \quad L_r' = L_{r0}' + M_m \quad (5.42)$$

$$\omega_s \cdot L_s = X_s , \quad \omega_s \cdot L_r' = X_r' \quad (5.43)$$

$$\omega_s \cdot L_{s0} = X_{s0} , \quad \omega_s \cdot L_{r0}' = X_{r0}' \quad (5.44)$$

$$\omega_s \cdot M_m = X_m \quad (5.45)$$

Mıknatıslama akımı,

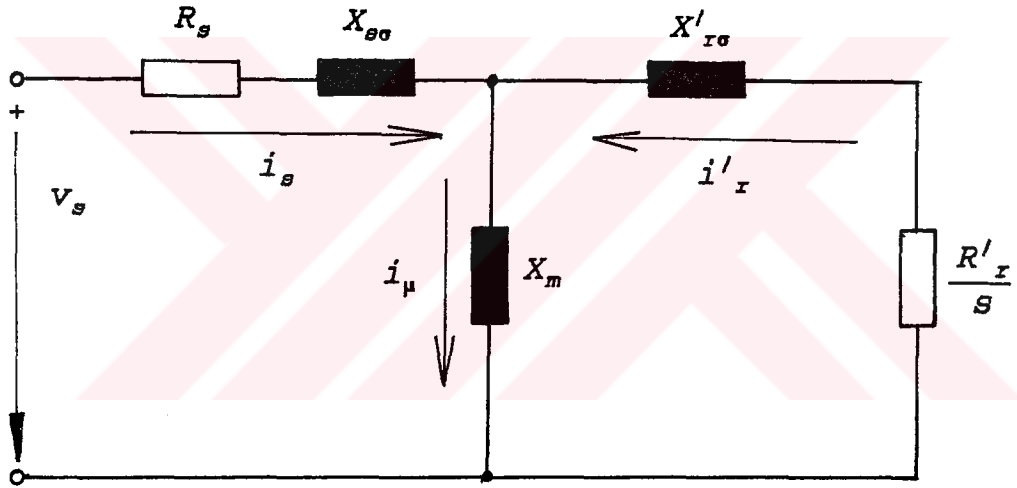
$$\vec{I}_\mu = \vec{I}_s + \vec{I}_r' \quad (5.46)$$

olarak tanımlandığında,

$$\vec{V}_s = R_s \cdot \vec{I}_s + j(X_{ss} \cdot \vec{I}_s + X_m \cdot \vec{I}_\mu) \quad (5.47)$$

$$\vec{V}_r' = \frac{R_r'}{s} \cdot \vec{I}_r' + j(X_{rs}' \cdot \vec{I}_r' + X_m \cdot \vec{I}_\mu) \quad (5.48)$$

bağıntıları elde edilir.



Şekil 5.3 Rotoru sargılı asenkron makinanın sürekli rejim eşdeğer devresi

Sürekli rejim eşdeğer devresinde yer alan parametreler kısa devre, boşa çalışma ve yavaşlama deneyleri yardımı ile ölçülebilir. Böylece makinanın gerçek büyüklükleri ile ölçülebilir büyüklükleri arasında bir ilişki elde edilmiş olur.

5.2.1 ROTORU SARGILI ASENKRON MAKİNANIN MATEMATİKSEL MODELİNİN AKILAR KULLANILARAK ELDE EDİLMESİ

Rotoru sargılı asenkron makinanın akı-akım bağıntıları kullanılarak, matematiksel model yeniden düzenlenirse,

$$v_{sd} = R_s \cdot i_{sd} - \omega_s \cdot \Phi_{sq} + \frac{d}{dt} \Phi_{sd} \quad (5.49)$$

$$v_{sq} = R_s \cdot i_{sq} + \omega_s \cdot \Phi_{sd} + \frac{d}{dt} \Phi_{sq} \quad (5.50)$$

$$v'_{rd} = R'_r \cdot i'_{rd} - \omega_r \cdot \Phi'_{rq} + \frac{d}{dt} \Phi'_{rd} \quad (5.51)$$

$$v'_{rq} = R'_r \cdot i'_{rq} + \omega_r \cdot \Phi'_{rd} + \frac{d}{dt} \Phi'_{rq} \quad (5.52)$$

$$M_e = P \cdot \frac{M_m}{L'_r} \cdot (i_{sq} \cdot \Phi'_{rd} - i_{sd} \cdot \Phi'_{rq}) \quad (5.53)$$

eşitlikleri elde edilir.

5.3 BİRİME İNDİRGEME (NORMALİZASYON)

Birime indirgemedede, aşağıda tanımlamaları yapılacak olan anma (nominal) büyüklükleri temel (baz) olarak alınacaktır. Birime indirgeme ile model daha işlevsel bir yapıya ulaşacaktır.[6]

Tablo 5.1 Fiziksel Büyüklüklerin Birime İndirgenmesi

Fiziksel Büyüklükler	Gerçek Büyüklükler	Temel Büyüklükler	Birime İndirgenmiş Büyüklükler
Gerilim	V	$\sqrt{2} V_N$	$\bar{V} = \frac{V}{\sqrt{2} V_N}$
Akım	I	$\sqrt{2} I_N$	$\bar{I} = \frac{I}{\sqrt{2} I_N}$
Reaktans Direnç	$X = \omega L$ R	$Z_N = \frac{V_N}{I_N}$	$\bar{X} = \frac{X}{Z_N}$ $\bar{I} = \frac{R}{Z_N}$
Açısal Frekans	$\omega = 2 \pi n$	$\Omega_{1N} = 100 \pi$	$\bar{f} = \frac{\omega}{\Omega_{1N}}$ $= \frac{f}{f_N}$

Hız	n	$N_{0N} = \frac{\Omega_{1N}}{2 \pi p}$	$\bar{n} = \frac{n}{N_{0N}}$
Frekans	f	$N_{0N} = \frac{\Omega_{1N}}{2 \pi p}$	$\bar{f} = \frac{f}{F_N}$
Moment	m_e m_y	$M_{eN} = \frac{P_N}{2 \pi N_N}$	$\bar{m}_e = \frac{m_e}{M_{eN}}$ $\bar{m}_y = \frac{m_y}{M_{eN}}$
Güç	p_a	$P_{sN} = 3 V_N I_N$	$\bar{p}_a = \frac{p_a}{P_{sN}}$
Akı	Φ	$\Phi_{0N} = \frac{\sqrt{2} V_L}{\Omega_{1N}}$	$\bar{\Phi} = \frac{\Phi}{\Phi_{0N}}$

(5.49)-(5.53) bağıntılarına birime indirgeme işlemi uygulanırsa,

$$\bar{V}_{sd} = \bar{I}_s \cdot \bar{I}_{sd} + T_N \frac{d}{dt} \bar{\Phi}_{sd} - \bar{F}_s \cdot \bar{\Phi}_{sq} \quad (5.54)$$

$$\bar{V}_{sq} = \bar{I}_s \cdot \bar{I}_{sq} + T_N \frac{d}{dt} \bar{\Phi}_{sq} + \bar{F}_s \cdot \bar{\Phi}_{sd} \quad (5.55)$$

$$\bar{V}_{xd} = \bar{I}_x \cdot \bar{I}_{xd} + T_N \frac{d}{dt} \bar{\Phi}_{xd} - \bar{F}_x \cdot \bar{\Phi}_{xq} \quad (5.56)$$

$$\bar{V}_{xq} = \bar{I}_x \cdot \bar{I}_{xq} + T_N \frac{d}{dt} \bar{\Phi}_{xq} + \bar{F}_x \cdot \bar{\Phi}_{xd} \quad (5.57)$$

$$\bar{m}_e = k_m \cdot X_m \cdot (\bar{I}_{xd} \cdot \bar{I}_{sq} - \bar{I}_{xq} \cdot \bar{I}_{sd}) \quad (5.58)$$

bağıntıları elde edilir.

Yukarıdaki bağıntılarda kullanılan tanımlar:

$$X_r = \frac{\Omega_{1N} \cdot L'_r}{Z_N} , \quad X_s = \frac{\Omega_{1N} \cdot L_s}{Z_N} , \quad X_m = \frac{\Omega_{1N} \cdot M_m}{Z_N} \quad (5.59)$$

$$k_m = \frac{2}{3} \cdot \frac{\bar{n}_N}{\bar{p}_N} , \quad T_N = \frac{1}{\Omega_{1N}} = \frac{1}{100 \cdot \pi} \quad (5.60)$$

Birime indirgenmiş anma hızı,

$$\bar{n}_N = \frac{n_N}{N_{CN}} \quad (5.61)$$

birime indirgenmiş anma mekanik gücü,

$$\bar{P}_N = \frac{P_N}{P_{SN}} \quad (5.62)$$

olarak tanımlanır.

Akı,akım bağıntıları birime indirgenirse,

$$\bar{\Phi}_{sd} = x_s \cdot \bar{i}_{sd} + x_m \cdot \bar{i}_{rd} \quad (5.63)$$

$$\bar{\Phi}_{sq} = x_s \cdot \bar{i}_{sq} + x_m \cdot \bar{i}_{rq} \quad (5.64)$$

$$\bar{\Phi}_{rd} = x_r \cdot \bar{i}_{rd} + x_m \cdot \bar{i}_{sd} \quad (5.65)$$

$$\bar{\Phi}_{rq} = x_r \cdot \bar{i}_{rq} + x_m \cdot \bar{i}_{sq} \quad (5.66)$$

biçiminde denklemler elde edilir. Son olarak hareket denklemleri de birime indirgenirse,

$$\bar{m}_e = \bar{m}_y + T_h \cdot \frac{d}{dt} \bar{n} \quad (5.67)$$

bağıntısına ulaşılır. T_h , yolalma zaman sabiti olup,

$$T_h = \frac{2 \cdot \pi \cdot J \cdot N_{ON}}{M_{eN}} \quad (5.58)$$

şeklinde tanımlanır.

d-q koordinatlarında, akımları, akılar cinsinden yazarsak,

$$\bar{I}_{sd} = \frac{1}{\sigma \cdot X_s} \cdot \bar{\Phi}_{sd} - \frac{X_m}{\sigma \cdot X_s \cdot X_r} \bar{\Phi}_{rd} \quad (5.69)$$

$$\bar{I}_{sq} = \frac{1}{\sigma \cdot X_s} \cdot \bar{\Phi}_{sq} - \frac{X_m}{\sigma \cdot X_s \cdot X_r} \bar{\Phi}_{rq} \quad (5.70)$$

$$\bar{I}_{rd} = \frac{1}{\sigma \cdot X_r} \cdot \bar{\Phi}_{rd} - \frac{X_m}{\sigma \cdot X_s \cdot X_r} \bar{\Phi}_{sd} \quad (5.71)$$

$$\bar{I}_{rq} = \frac{1}{\sigma \cdot X_r} \cdot \bar{\Phi}_{rq} - \frac{X_m}{\sigma \cdot X_s \cdot X_r} \bar{\Phi}_{sq} \quad (5.72)$$

Bu ifadelerde,

$$\sigma = 1 - \frac{X_m^2}{X_s \cdot X_r} \quad (5.73)$$

kaçak faktörüdür. Moment ve hareket bağıntısı,

$$\bar{m}_e = k_m \cdot X_m \cdot (\bar{I}_{sq} \cdot \bar{I}_{rd} - \bar{I}_{rq} \cdot \bar{I}_{sd}) = \bar{m}_y + T_h \frac{d}{dt} \bar{n} \quad (5.74)$$

şeklinde tanımlanır.

Rotoru sargılı asenkron motorun, d-q dönen koordinat sisteminde, rotoru statora indirgenmiş ve birime indirgenmiş büyüklükler cinsinden, modelinin durum denklemleri aşağıdaki gibidir. Bu denklemlerde parametreler zamanla değişmemektedir. Model, 4 adet elektriksel yana ilişkin, 1 adet de mekanik yana ilişkin olmak üzere toplam 5 adet, birinci mertebeden, sabit katsayılı diferansiyel denklem sisteminden oluşmaktadır.

$$\frac{d}{dt} \bar{\Phi}_{sd} = \frac{1}{T_N} (-\bar{I}_s \cdot \bar{I}_{sd} + \bar{F}_s \cdot \bar{\Phi}_{sq} + \bar{V}_{sd}) \quad (5.75)$$

$$\frac{d}{dt} \bar{\Phi}_{sq} = \frac{1}{T_N} (-\bar{I}_s \cdot \bar{I}_{sq} - \bar{F}_s \cdot \bar{\Phi}_{sd} + \bar{V}_{sq}) \quad (5.76)$$

$$\frac{d}{dt} \bar{\Phi}_{rd} = \frac{1}{T_N} (-\bar{I}_r \cdot \bar{I}_{rd} + \bar{F}_r \cdot \bar{\Phi}_{rq}) \quad (5.77)$$

$$\frac{d}{dt} \bar{\Phi}_{rq} = \frac{1}{T_N} (-\bar{I}_r \cdot \bar{I}_{rq} - \bar{F}_r \cdot \bar{\Phi}_{rd}) \quad (5.78)$$

$$\frac{d}{dt} \bar{n} = \frac{1}{T_h} [k_m \cdot x_m \cdot (i_{sq} \cdot i_{rd} - i_{rq} \cdot i_{sd}) - \bar{m}_y] \quad (5.79)$$

Bu modelde giriş büyüklükleri $\bar{V}_{sd}$, $\bar{V}_{sq}$, $\bar{F}_s$, durum değişkenleri ise, akılar olup, $\bar{\Phi}_{sd}$, $\bar{\Phi}_{sq}$, $\bar{\Phi}_{rd}$, $\bar{\Phi}_{rq}$ dur.

(5.69) - (5.72) arası akım bağıntıları ile (5.75) - (5.79) arası akı bağıntılarını eşzamanlı olarak, dördüncü mertebeden Runge-Kutta yöntemi ile çözerek, rotoru sargılı asenkron makinanın benzetişimini (simülasyonunu) sağlayan program, Turbo PASCAL programlama dilinde, EK-1'de sunulmuştur. Bu program yardımı ile giriş büyüklükleri değiştirilerek, rotoru sargılı asenkron makinanın geçici ve sürekli rejim davranışları gözlemlenebilir.

BÖLÜM 6

ASENKRON MOTOR SÜRÜCÜLERİ VE ÖZELLİKLERİ

Günümüz teknolojisinin güç elektroniği elemanları ve mikrobilgisayarlara getirdiği yeniliklerle, teorik olarak geliştirilen asenkron motor kontrol yöntemleri, pratik olarak da gerçekleştirilmekte ve uygulanmaktadır. Uygulanacak kontrol teorisi ne olursa olsun, istenen kontrol işaretinin asenkron makineye uygulanabilmesi için uygun bir sürücü devreye her zaman gereksinim vardır. Sürücüler, uygulanacak kontrol teorisine göre üretilen kontrol işaretini, en yüksek performans ile motora uygulayabilmelidir. Bu nedenle kullanılacak sürücüler asenkron makineyi çok geniş bir aralıkta kontrol edebilmelidir. Bu amaçla çeşitli sürücü devre tasarımları geliştirilmiş olup, her tip sürücünün bir diğerine göre üstünlükleri ve sakıncaları bulunmaktadır.

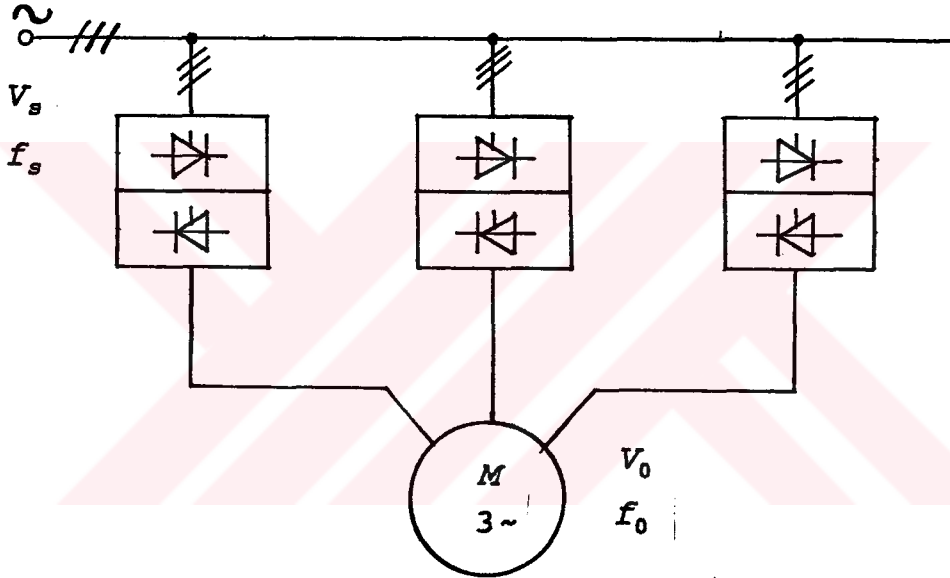
Asenkron makinenin kontrolünde uygulanabilecek en iyi yöntemlerden biri stator gerilimi frekansı ile stator gerilimi genliğinin birlikte değiştirilmesidir. Bu durumda makine anma devrine kadar maksimum momenti ile yüklenebilir ve geniş bir aralıkta kontrol sağlanabilir.

Asenkron makine kontrolünde kullanılan frekans çeviricileri başlıca iki sınıfta toplanabilir;

- a) Doğrudan frekans çeviriciler.
- b) Aradevrelili frekans çeviriciler.

6.1. DOĞRUDAN FREKANS ÇEVİRİCİLER

Doğrudan frekans çeviriciler ile şebeke geriliminden, değişken genlik ve frekansa gerilim elde edilir [13]. Genellikle yarım yada tam dalga doğrultucular birbirlerine ters paralel bağlanarak kullanılırlar. Yarım dalga doğrultucularda 18, tam dalga doğrultucularda ise 36 adet tristör bulunur. Güç ve kontrol devrelerinin üretim maliyetleri çok yüksek olduğundan, çok büyük güçlü asenkron makinelerin (> 1 MW) kontrolunda kullanılır.



Şekil 6.1. Doğrudan frekans çevirici ilke şeması

Çıkış dalga şeklindeki bozukluklar nedeni ile motora uygulanan gerilimin frekansının alt sınırı $f_0 \leq \frac{1}{3} f_s$ ($\sim 17-20 \text{ Hz}$) olarak belirlenir. İstendiğinde $f_0 = 0.5 \text{ Hz}$ 'e kadar inilebilir. Şebeke frekansının üzerine çıkma olanağı olmadığından asenkron motor anma devrinin üzerindeki hızlarda döndürülemez. Düşük hızlı tahrik sistemlerinde kullanılır. Uygulamada doğrudan frekans çeviriciler, çimento fabrikalarındaki soğutma sistemleri ve döner fırınların (Klinker) tahrik edilmesinde kullanılır.

6.2. ARADEVRELİ FREKANS ÇEVİRİCİLER.

Genel anlamda aradevrelİ frekans çeviricileri başlıca üç kısımdan oluşur.

1) Girişte, şebeke gerilimini doğrultan kontrollu veya kontrolsuz doğrultucu,

2) Doğrultucuya bağılı pasif filtre elemanlarından oluşan aradevre,

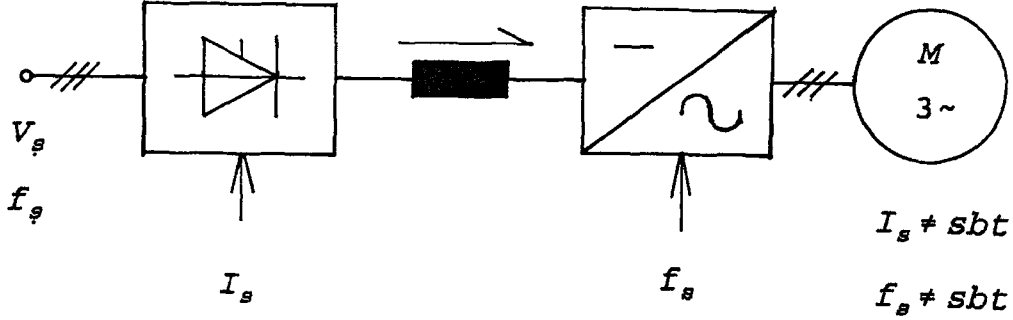
3) Aradevre üzerinden beslenen evirici.

Aradevrelİ frekans çeviricileri aradevrenin şekline göre başlıca iki kısma ayrılır;

- a) Akım aradevrelİ frekans çeviricileri,
- b) Gerilim aradevrelİ frekans çeviricileri.

6.2.1. AKIM ARADEVRELİ FREKANS ÇEVİRİCİLERİ (AAFÇ)

Akım aradevrelİ frekans çeviricilerinde, akımın genliğini sabit tutabilmek için girişte kontrollu bir doğrultucu ve aradevrede büyük değerli bir seri endüktansdan yararlanılır [13]. Aradevre çıkışına ise akımın frekansının kontrol edildiğı bir evirici bağlanır. Eviricide klasik tristörler kullanıldığında ardışıl komitasyon uygulanır ve yardımcı tristörlere gerek duyulmaz. Akımı sabit tutmak için kullanılan kontrollu doğrultucu evirme modunda da çalıştırılabilir. Böylece asenkron motor frenleme yaptığında yada generatör modunda çalıştırıldığında akım yönün aynı kalması koşulu ile aradevre gerilimi yön değiştirebilir. Bu durumda aradevrede biriken enerji evirme modunda çalıştırılan kontrollu doğrultucu üzerinden şebekeye aktarılabilir. Ayrıca eşlenik paralel kontrollu doğrultucular kullanılarak akım aradevrelİ frekans çeviriciler yardımı ile dört bölgede asenkron makinanın kontrolu yapılabilir.



Şekil 6.2. Akım aradevrelili frekans çevirici ilke şeması.

Çıkış akımındaki harmonikler ek kayıplara ve özellikle düşük hızlarda moment dalgalanmalarına neden olmaktadır. Harmonikleri azaltmak için evirici darbe genişlik modülasyonlu çalıştırılabilir. Ancak akımın, moment oluşumunda doğrudan etkili olması, gerilim aradevrelili frekans çeviricilerine göre bu sistemin önemli sakıncasını oluşturur.

Akım aradevrelili frekans çeviricisinde eviricinin çalışması tamamen yüke bağlıdır. Yük akımı arttığında komutasyon süresi azalır, yük akımı azaldığında komutasyon süresi artar. Komutasyon süresi aynı zamanda asenkron makinanın kaçak reaktanslarında biriken enerjiye de bağlıdır. Komutasyon esnasında asenkron makinanın sargılarında gerilim sıçramaları oluşur. Kaçak reaktanslar büyüdükçe sargı uçlarındaki gerilim sıçramaları da artar. Gerilim sıçramaları ise sargı yalıtkanlarını zorlar. Bu tip sürücü üzerinden beslenen asenkron makinalarda yalıtım sınıfı (sargı tellerinin emaye kalınlığı) yüksek seçilmelidir.

Akım aradevrelili frekans çeviricilerin, büyük değeri aradevre filtre elemanı nedeni ile dinamik performansları düşüktür. Yüksüz olarak çalıştırılmazlar. Aynı anda birden fazla motoru paralel olarak kontrol edemezler. Yani genel amaçlı bir çevirici tipi değildir.

6.2.2. GERİLİM ARADEVRELİ FREKANS ÇEVİRİCİLERİ (GAFC)

Gerilim aradevrelili frekans çeviricilerde aradevre endüktansı değeri daha düşük olup, gerilimin sabit tutulması için, eviriciye paralel, bir de filtre kondansatörü yer alır. Komutasyon olayı yükten bağımsız olduğundan boşa da çalıştırılabilir ve birden çok motoru kontrol etmeye uygun yapıdadır. Aradevre geriliminin yapısına göre iki türdür:

- a) Değişken gerilim aradevrelili frekans çevirici, (DGAFC)
- b) Sabit gerilim aradevrelili frekans çevirici (SGAFC)

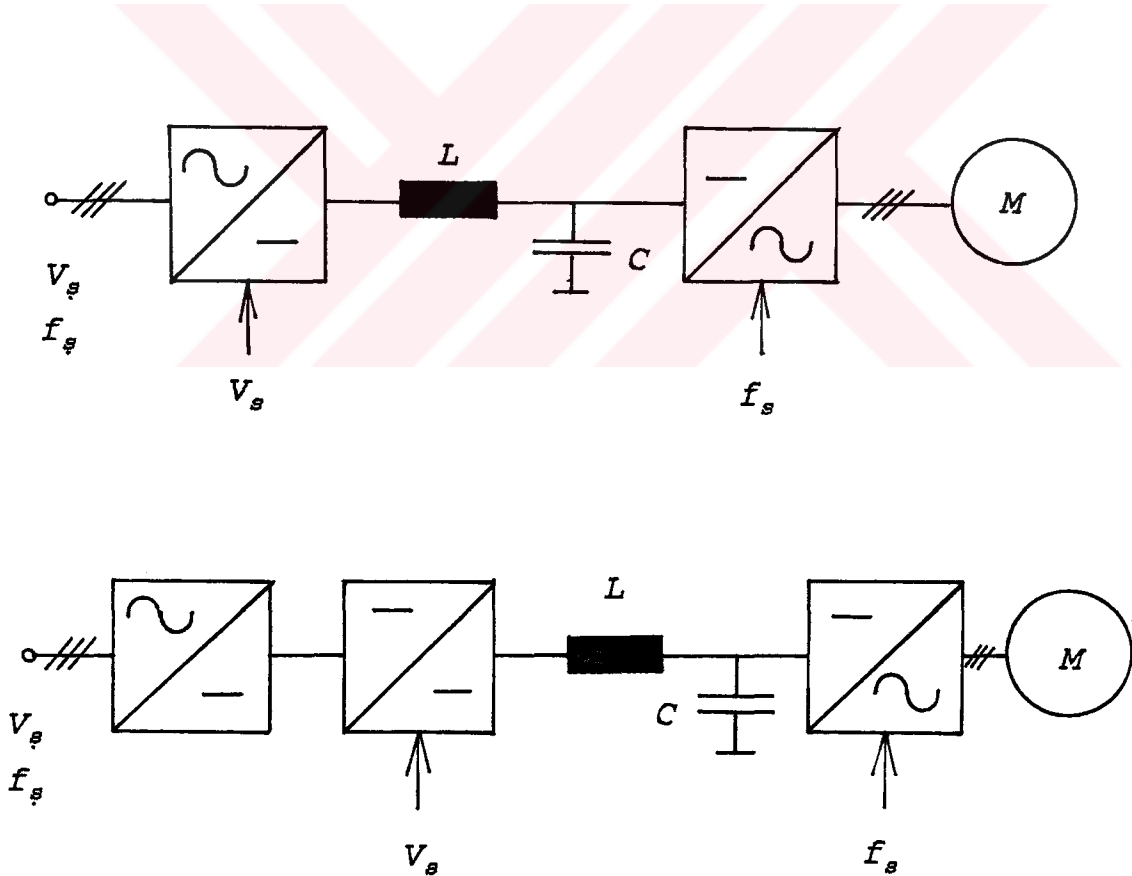
6.2.2.1. DEĞİŞKEN GERİLİM ARADEVRELİ FREKANS ÇEVİRİCİ

Değişken gerilim aradevrelili frekans çeviricilerde aradevre gerilimi iki türlü elde edilebilir:

- 1) Girişte kontrollü doğrultucu kullanılarak,
- 2) Girişte kontrolsüz doğrultucu kullanılıp, aradevre gerilimi, doğru gerilim kıyıcı ile değiştirilerek.

Her iki yöntemde de aradevre gerilimi geniş bir aralıkta değiştiğinden, harmonik distorsiyon da geniş bir aralıkta değişir. Maksimum distorsiyonun oluştuğu aradevre gerilimini de süzebilmek için aradevre kondansatörü büyük seçilmelidir. Bu da büyük zaman sabiti demek olduğundan sistem dinamiğini yavaşlatır. Eviricinin klasik tristörler kullanılarak gerçekleştirilmesi durumunda komutasyon olayı, akım aradevreliliye göre çok daha zordur. Komutasyon için yardımcı tristörlere de gerek duyulur ve kontrol devresi

karmaşıklılaşır. Evirici çıkışında elde edilen 120° 'lik kare dalga biçimli blok , 2 ve 3'ün katı harmonikler yer almadığı halde yine de büyük değerli harmonik gerilimleri içerir. Bunun sonucu düşük hız değerlerinde ısınma ve moment salınımları artar. Yük akımında oluşan tepelerden ötürü eviricinin büyük boyutlandırılması gerekir. Girişte kontrollü doğrultucu kullanıldığında güç faktörü düşük olur. Aradevredeki kondansatörün büyüklüğü nedeni ile sistem performansı azalır. Girişte kontrolsüz doğrultucu kullanılıp, aradevre gerilimi doğru gerilim kıyıcı ile değiştirildiğinde güç faktörü 1'e yaklaşır. Ayrıca kıyıcı frekansının yüksek olması halinde, aradevre gerilimindeki harmonikler azalacağından daha düşük değerli kondansatör kullanılıp sistem performansı yükseltilebilir.



Şekil 6.3. Değişken gerilim aradevreli frekans çeviricileri ilke şemaları

6.2.2.1. SABİT GERİLİM ARADEVRELİ FREKANS ÇEVİRİCİLER

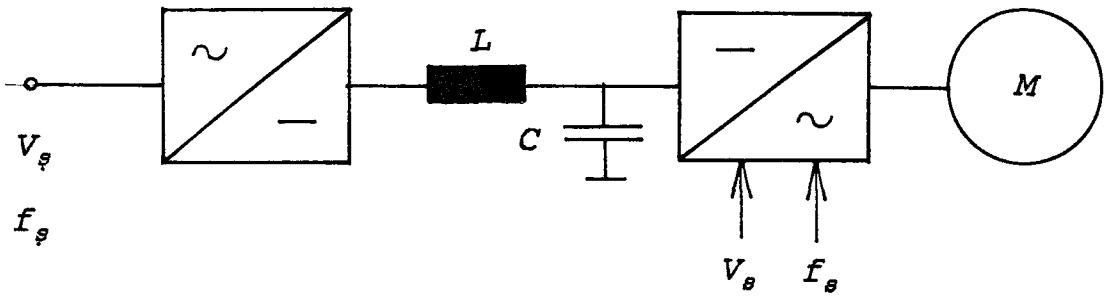
Bu tip frekans çeviricilerinde aradevre gerilimi sabittir. Şebeke yanında kontrolsuz doğrultucu kullanılır. Böylece hem güç faktörü iyileştirilmiş olur hemde daha küçük süzme elemanlarına gerek duyulur. Aradevre geriliminin sabit olması harmoniklerin de azalmasını sağlar. Sabit aradevre gerilimi, hem genliğin hemde frekansın denetlendiği evirici üzerinden asenkron makinaya aktarılır. Evircinin aynı anda hem genliği hemde frekansı kontrol edebilmesi Darbe Genişlik Modülasyonu (PWM - Pulse Width Modulation) ile sağlanır. Darbe genişlik modülasyonu'nda darbe sayısının artması ve darbelerin uygun noktalarda üretilmesi, bastırılan harmonik sayılarının artmasını sağlayacaktır. Ancak artan darbe sayısı, anahtarlama elemanlarının daha hızlı çalışmalarını gerektirecektir. Bu da yüksek hızlı anahtarlama elemanlarının kullanılması demektir. Gelişen güç elektroniği teknolojisi, bu tip hızlı kullanıma yönelik, anahtarlama kayıpları düşük olan bipolar tranzistör, güç mosfet'leri ve IGBT (Yalıtılmış kapılı Bipolar Tranzistör) elemanlarını küçük ve orta güçler için kullanıma sunmuştur. Büyük güçlü sistemler için anahtarlama hızı daha düşük olan tristör ve GTO (Kapıdan Tıkanabilir Tristör) kullanımı sürmektedir.

Darbe Genişlik Modülasyonu, harmonik eliminasyonuna dayanan yöntemler ile günümüzde yoğun olarak kullanılır. Harmonik eliminasyonunda darbelerin başlangıç ve bitiş açıları, yani darbelerin süreleri ile darbeler arası süreler, optimum kontrol ilkelerine göre önceden belirlenir. Açıların uygun belirlenmesi ile hem harmonikler hemde gerilimin genliği kontrol edilebilir. Harmoniklerin uygun şekilde bastırılması ile motora uygulanan gerilim ve dolayısı ile akım sinüzoidal yapıya yaklaştırılır. Böylece kayıplar ve moment salınımları en aza indirilmiş olur. Darbe genişlik modülasyonunun üretilmesi için çeşitli firmalar tarafından piyasaya sürülmüş lojik tümdevreler

bulunmaktadır. PHILIPS'in HEF-4752 tümdevresi asenkron makinanın skaler kontroluna yönelik ($v_s / f_s = \text{sabit}$) darbe genişlik modülasyonlu kontrol işaretini üç fazlı olarak, evirici anahtarlama elemanlarına uygulamaya hazır yapıda üretmektedir. Bunun dışında mikroişlemci temelli DGM üretmeye yönelik sistemler de yoğun olarak kullanılmaktadır. SIEMENS'in SAB-8051 mikroişlemcisi ve SLE-4520 tümdevre çifti de mikroişlemci temelli DGM üretiminde yoğun kullanım alanına sahiptir.

Sürücüde amaç, asenkron makınayı olabildiğince sinüsoidal biçime yakın akımla beslemektir. Bu amaçla evirici, darbe genişlik modülasyonu yanında, motor akımı sinüzoidal bir akım referansını izleyebilecek şekilde de anahtarlatabilir.

SGAFÇ'de aradevre gerilimi sabit olduğundan şebeke geriliminin yanında akü grubu üzerinden de beslenebilir.



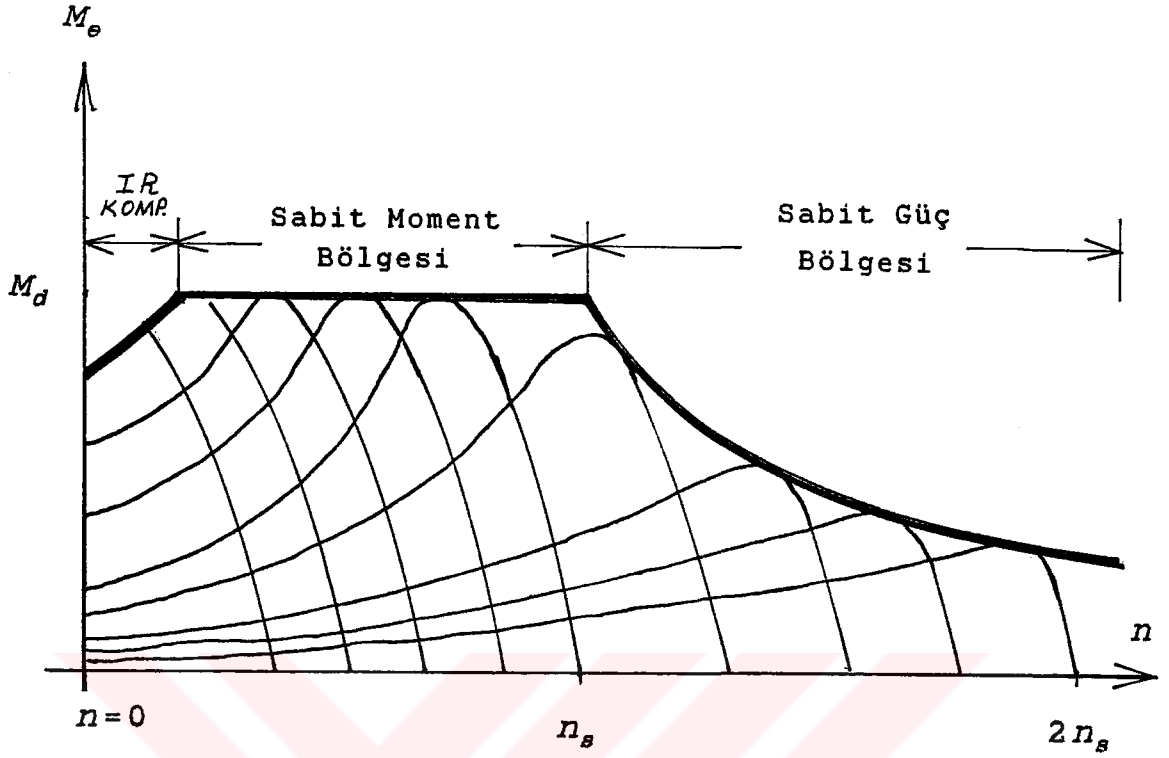
Şekil 6.4. Sabit gerilim aradevreli frekans çevirici ilke şeması

BÖLÜM 7

ASENKRON MAKİNANIN SKALER KONTROLU

Tez çalışmasında pratik olarak gerçekleştirilen hız kontrolunda, sabit gerilim aradevrelili sürücü, skaler kontrol yöntemine göre, girişine gelen kontrol işaretine bağlı olarak, üç fazlı, darbe genişlik modülasyonlu gerilimi asenkron motora uygulamaktadır. Skaler kontrol yöntemi doğru gerilim aradevrelili frekans çeviriciler veya doğru akım aradevrelili frekans çeviriciler kullanılarak asenkron makınaya uygulanabilir. Skaler kontrolde kontrolu yapılan değişkenin sadece modülü ile ilgilenilip, fazı gözönüne alınmaz. Kontrol edilen büyüklüklere ilişkin referans ve geribesleme işaretleri doğru gerilimlerdir. Skaler kontrolde motora uygulanan stator geriliminin genliği ile frekansları oranı sabit kalacak şekilde, gerilim ve frekans birlikte değiştirilir. Böylece frekansın anma değerinin altındaki çalışma noktaları için, statordan rotora geçen hava aralığı akısı Φ_m sabit kalacağından, motorda endüklenen momentin devrilme noktaları sabit kalır. Φ_m 'nin sabit kalması rotorda endüklenen gerilimin frekansı f_r 'nin , ve rotor akımı I_r 'nin de sabit kalmasını sağlar. Buna asenkron motorun moment-hız özeğrisi üzerinde, sabit moment bölgesinde çalışması adı verilir.

Frekansın anma değerinin üzerine çıkarılması halinde, stator gerilimi anma değerinin üzerine çıkarılamaz. Bu durumda artık $\vec{V}_s / f_s = \text{sabit ilkesi geçerli değildir}$. Bu tip çalışmada milden alınan güç sabit kalmak üzere, mil hızı artar, moment azalır. Buna asenkron motorun sabit güç bölgesinde çalışması adı verilir.

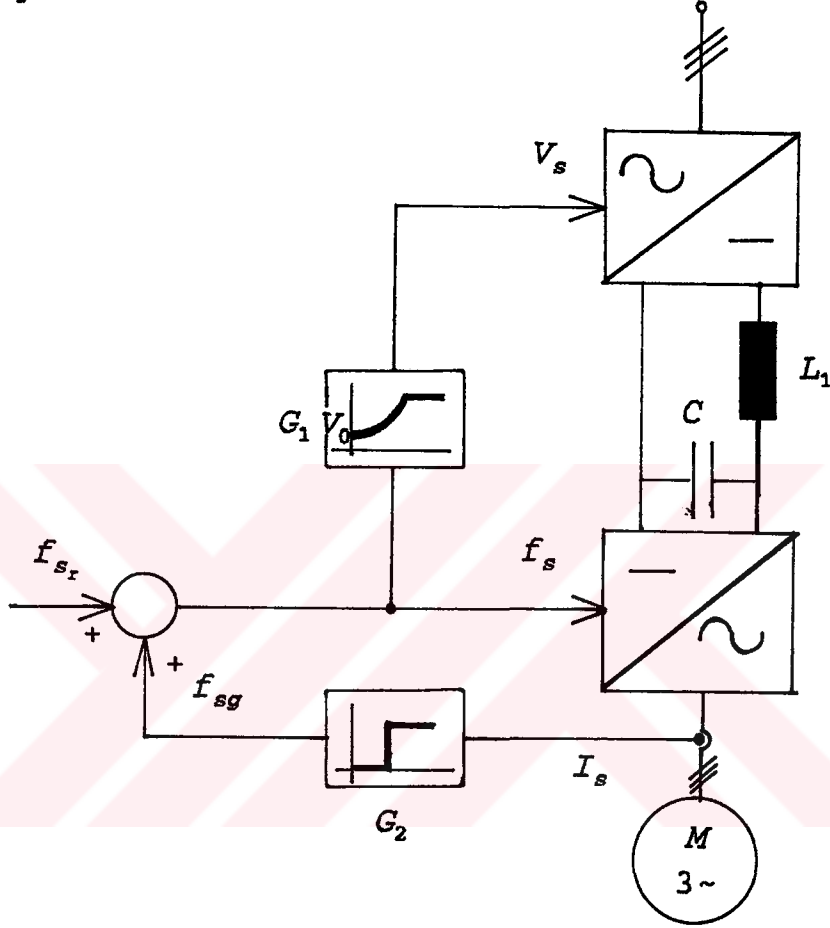


Şekil 7.1. Değişken genlik ve frekanslı kaynaktan beslenen asenkron motorun moment-hız özeğrileri

7.1. DOĞRU GERİLİM ARADEVRELİ FREKANS ÇEVİRİCİ (DGAFC) KULLANILARAK, GERİLİM / FREKANS ORANININ SABİT TUTULDUĞU SÜRÜCÜ SİSTEMLERİ

Açık çevrim skaler kontrola ilişkin sürücü devre şebeke yanında bir veya üç fazlı kontrollü doğrultucu, L ve C filtre elemanlarından oluşan aradevre ve evirici'den oluşur [13]. Eviriciye uygulanan kontrol işareti frekans referansından gelmektedir. Gerilimin genliği için kontrol- lu doğrultucuya, kontrol işareti, frekans referansının belli bir katı olacak şekilde uygulanmaktadır. Frekansın ve dolayısı ile hızın çok düştüğü çalışma noktalarında statora uygulanan gerilim de çok düşecektir. Bu durumda stator direnci üzerindeki kayıplar artık ihmal edilemez bir

noktaya gelecektir. Bu nedenle yardımcı bir V_0 gerilimi kontrollü doğrultucunun kontrol işaretine eklenerek stator kayıpları kompanse edilir. Böylece anma hava aralığı akısı ve maksimum moment, hızın yaklaşık sıfır olduğu noktaya kadar sağlanabilir.

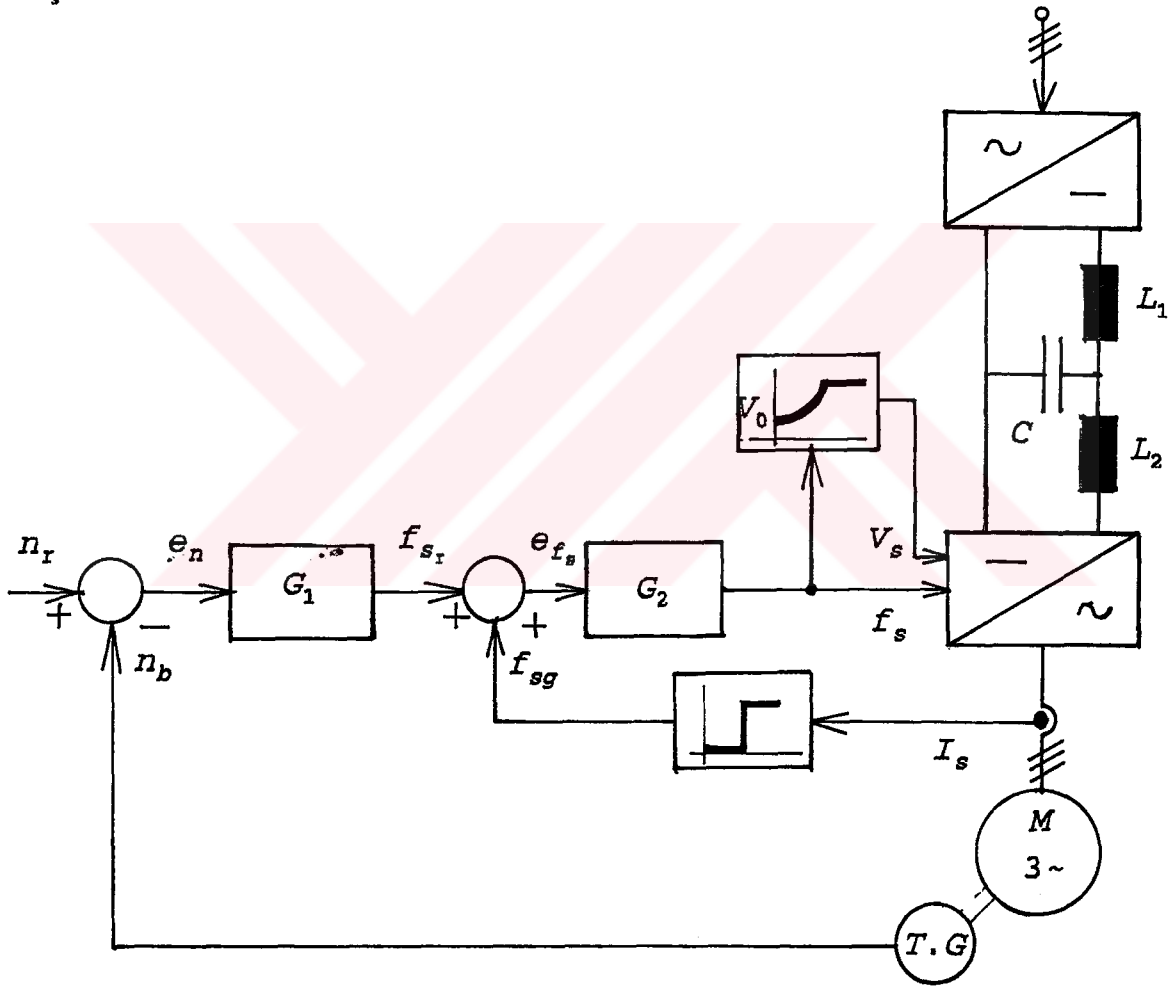


Şekil 7.2. Açık çevrim $\vec{V}_s/f_s$ kontrolü ilke şeması

Sürekli rejimde çalışmada eğer yük momentini artarsa, hız azalacak ve kayma büyüyecektir. Artan yük momentine göre sistem yeni çalışma noktasında dengeye gelecektir. Bu tip açık çevrim çalışmada, frekans referansı ani olarak (basamak biçiminde) arttırılıp, azaltılırsa motor devrilme momentini noktasını aşar ve kararsız çalışma bölgesine geçebilir. Bunu önlemek için akımdan geribesleme alınarak, frekans referansındaki basamak biçimli artış ve azalışlar olduğunda, evirici kontrol işaretinin daha yavaş biçimde

değişmesi sağlanabilir. Böylece motorun, frekans değişimlerini, kararlı çalışma bölgesinin dışına çıkmadan izlemesi sağlanabilir.

Açık çevrim kontrolde şebeke gerilimdeki dalgalanmalar, evirici çıkış gerilimine de yansiyarak hava aralığı akısında dalgalanmalara neden olup, moment ve hız salınımları yaratabilmektedir. Yine açık çevrim kontrolde motorun yük alma ve yük atma durumlarında hızının değişmesi engellenemez. Bu tip sakıncaların önüne geçmek için kapalı çevrim kontrollü sürücüler oluşturulmuştur.



Şekil 7.3. Kapalı çevrim $\vec{V}_s/f_s$ kontrolü ilke şeması

Kapalı çevrim kontrolde sürücü, sabit gerilim aradevrelili frekans çevirici olabilir [3]. Motora uygulanacak gerilimin genliği ve frekansı Darbe Genişlik Modülasyonu

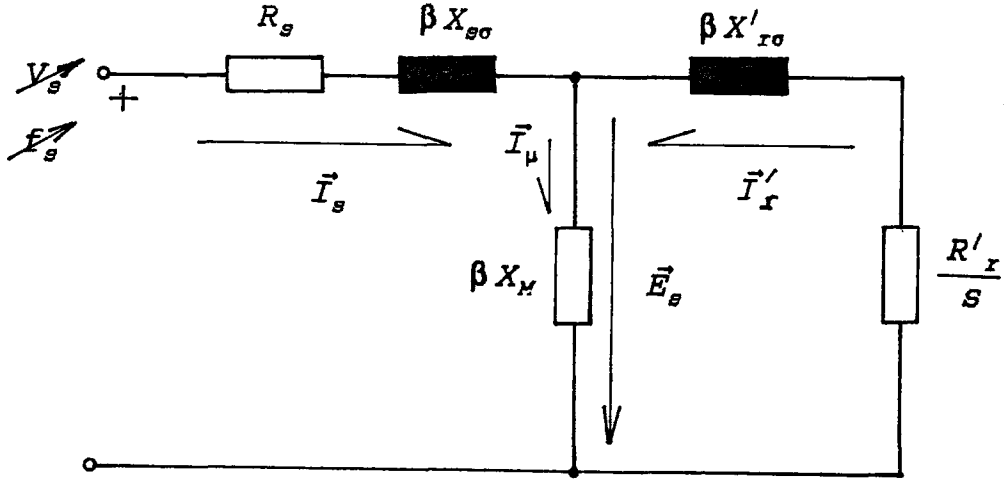
ile evirici üzerinden kontrol edilebilir. Hıza ilişkin geribesleme işareti takogeneratör ve döner kodlayıcı (encoder) üzerinden alınıp, referans değer ile karşılaştırılarak elde edilen hata işareti G_1 kontrolörüne uygulanır. G_1 kontrolörü çıkışındaki işaret frekans referansı olarak alınıp, akımdan gelen geribesleme işareti ile toplanarak G_2 kontrolörüne aktarılır. G_2 çıkışında motora uygulanacak frekansa ilişkin kontrol işareti elde edilmektedir. Bundan aynı zamanda gerilim için kontrol işareti de, düşük frekanslardaki stator kayıpları kompanzatörünü de içeren $\vec{V}_s/f_s$ fonksiyon generatörü bloğu üzerinden elde edilir. G_1 ve G_2 PI tipi kontrolörler olabilir. Şekil 7.3.'deki ilke şemada da görüldüğü gibi kapalı çevrin kontrolde motor akımından da geri besleme alınmaktadır. Böylece hız referansındaki basamak biçimli artış ve azalmalarda, frekans kontrol işaretinin referansı daha yavaş izlemesi sağlanıp, motorun devrilme momentini aşarak, kararsız çalışma bölgesine geçmesi önlenir. Yani rotor hızının, moment-hız özeğrisindeki değişimleri, lineer çalışma bölgesi içinde kalarak izlemesi sağlanır. Stator akımından alınan geribesleme ile aynı zamanda, sürücü sistemi ve motorun, aşırı yüklenme ve aşırı akımlara karşı korunması da sağlanır.

7.2. DEĞİŞKEN GENLİK VE FREKANSLI KAYNAKLA BESLENEN ASENKRON MAKİNAYA İLİŞKİN MATEMATİKSEL BAĞINTILAR

Asenkron makinanın sürekli sinüzoidal halde bir faza ve rotoru statora indirgenmiş T eşdeğer devresinin değişken frekanslı kaynaktan beslenmesi durumu için yeniden çizilmiş hali, şekil 7.4.'de verilmiştir.

Bu eşdeğer devreden yararlanarak, asenkron makinaya skaler kontrol uygulandığında, devrilme momentinin nasıl sabit kaldığının ispatı aşağıda sunulacaktır. Bu amaçla gözönüne alınması gerekli ilk nokta, asenkron makinaya uygulanan gerilim frekansının değişmesi ile, özendüktif

reaktanslarının değişeceğiştir. Özendüktif reaktansların frekansa bağlı değişimi, β katsayısı ile çarpılarak modellenenabilir.



Şekil 7.4. Değişken frekanslı kaynaktan beslenen asenkron makinanın T eşdeğer devresi

$$\beta = \frac{f_s}{f_N} \quad (7.1)$$

Burada,

f_s : Statora uygulanan gerilimin frekansı,

f_N : Stator anma frekansıdır.

n : Rotor (mil) hızı,

n_s : Statora uygulanan frekansa bağlı senkron devir hızı olmak üzere, kayma,

$$s = \frac{n_s - n}{n_s} \quad (7.2)$$

ifadesi ile tanımlanır.

p : Çift kutup sayısı,

f_{mil} : Rotor (mil) frekansı,

f_r : Rotorda endüklenen gerilimin frekansı olmak üzere

$$n_s = \frac{60 \cdot f_s}{p} \quad n_r = \frac{60 \cdot f_{mil}}{p} \quad (7.3)$$

$$s = \frac{n_s - n}{n_s} = \frac{\frac{60 \cdot f_s}{p} - \frac{60 \cdot f_{mil}}{p}}{\frac{60 \cdot f_s}{p}} = \frac{f_s - f_{mil}}{f_s}$$

$$f_r = s \cdot f_s \quad (7.4)$$

bağıntıları elde edilir.

Şekil 7.4.'deki eşdeğer devrede yer alan X_{ss} , X_{rs} , X_M özendüktif reaktansları, $f_N = 50$ Hz anma frekansı için tanımlanmaktadır.

$$X_{ss} = 2\pi f_N L_{ss} \quad (7.5a)$$

$$X'_{rs} = 2\pi f_N L'_{rs} \quad (7.5a)$$

$$X_M = 2\pi f_N L_M \quad (7.5c)$$

Statorda endüklenen gerilim için,

$$\vec{E}_s = \vec{I}_r \left[\frac{R'_r}{s} + j\beta X'_{rs} \right] \quad (7.6)$$

yazılıp, (7.1) ve (7.4) bağıntıları (7.6)'da yerine konulduğunda

$$\vec{E}_s = \vec{I}_r \left[\frac{R_r \cdot f_s}{f_r} + j \frac{f_s}{f_N} X_{ss} \right] \quad (7.7)$$

bağıntısı elde edilir.

Hava aralığı akısı Φ_m ,

$$\Phi_m = \frac{E_s}{4.44 \cdot N_1 \cdot f_s} \quad (7.8)$$

$K = \frac{1}{4.44 \cdot N_1}$ olarak tanımlandığında

$$\Phi_m = K \cdot \frac{E_s}{f_s} \quad (7.9)$$

bağıntısına ulaşılır. (7.7) ifadesi, (7.9)'a benzer yapıya getirildiğinde

$$\frac{\vec{E}_s}{f_s} = \vec{I}_r' \left[\frac{R_r'}{f_r} + j \frac{X_{rg}}{f_N} \right] \quad (7.10)$$

(7.5b) tanımını uyarınca,

$$\frac{\vec{E}_s}{f_s} = \vec{I}_r' \left[\frac{R_r'}{f_r} + j 2 \pi L_{rg}' \right] \quad (7.11)$$

ifadesi elde edilir.

(7.8) ve (7.9) ifadelerinde de görüldüğü gibi hava aralığı akısı Φ_m 'nin sabit kalabilmesi, $\vec{E}_s/f_s$ oranının sabit kalmasına bağlıdır. $\vec{E}_s/f_s$ oranı sabit olduğunda, (7.11) uyarınca, $\vec{I}_r$ rotor akımının ve f_r , rotorda endüklenen gerilim frekansının da sabit kalması sağlanır.

Hava aralığından rotora aktarılan güç, m_s stator faz sayısı olmak üzere

$$P_g = m_s \cdot (I_r')^2 \cdot \frac{R_r'}{s} \quad (7.12)$$

endüklenen moment ifadesi ise,

$$M_g = \frac{P_g}{\omega_s} \quad (7.13)$$

bağıntıları ile elde edilir. Stator gerilimi açısal frekansı uygulanan frekansa bağlı olarak,

$$\omega_s = 2\pi \frac{n_s}{60} \quad (7.14)$$

(7.3) ve (7.14) ifadeleri yerine yazılarak,

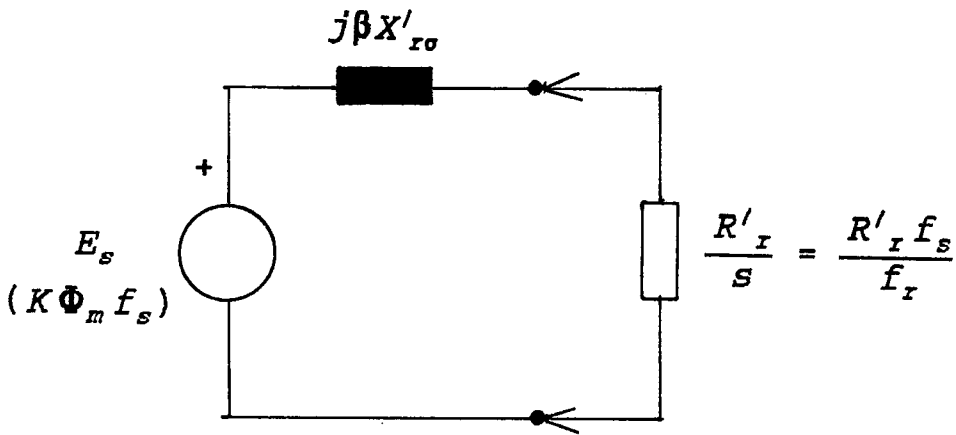
$$\omega_s = 2\pi \frac{f_s}{p} \quad (7.15)$$

şeklinde elde edilir. (7.12) ve (7.15) bağıntıları (7.13)'de yerine yazılarak, endüklenen moment ifadesi için

$$M_e = m_s \frac{(I'_r)^2 \cdot R'_r}{\frac{2\pi}{p} \cdot f_r} \quad (7.16)$$

bağıntısı elde edilir.

Endüklenen momentin maksimum değeri (devrilme momenti), eşdeğerden devrede statordan rotora maksimum güç aktarımı koşulu, Thevenin eşdeğerinden giderek bulunabilir.



Şekil 7.5. Rotor yanından stator yanına bakıldığında görülen Thevenin eşdeğer devresi

Maksimum güç aktarımı teoremine göre, bir kaynaktan bir yüke maksimum güç aktarımı koşulu, kaynak ve yük empedanslarının birbirlerine eşit olmalarıdır. Buna göre

$$\frac{R'_r}{s} = \beta X'_{r0} \quad (7.17)$$

şeklinde kaynak ve yük empedansları eşitliği yazılabilir. Buradan, rotorda endüklenen gerilimin frekansının maksimum değeri için

$$f_{r_{\max}} = f_N \cdot \frac{R'_r}{X'_{r0}} \quad (7.18)$$

bağıntısı elde edilir. Görüldüğü gibi maksimum rotor gerilimi frekansı, statora uygulanan gerilimin frekansından ve rotor hızından bağımsız olup, anma frekansı ve rotor parametrelerine (R_r , X_{r0}) bağlıdır. $f_{r_{\max}}$ devrilme momenti noktasında rotorda endüklenen gerilimin frekansıdır. Rotor akımı için,

$$I'_r = \frac{E_s}{\sqrt{\left(\frac{R'_r}{s}\right)^2 + (X'_{r0})^2}} \quad (7.19)$$

bağıntısı yazılabilir. (7.9) ve (7.17) bağıntılarını (7.19)'da kullanarak,

$$(I'_r)^2 = \frac{k \cdot \Phi_m \cdot f_s}{\beta^2 \cdot (X'_{r0})^2} \quad (7.20)$$

yapısında, rotor akımının karesel bağıntısı elde edilir. (7.17) ve (7.20) bağıntıları (7.15)'de yerine yazılarak devrilme moment bağıntısı,

$$M_{dev} = m_s \frac{(k \cdot \Phi_m)^2 \cdot f_N}{\frac{4\pi}{p} \cdot X'_{r0}} \quad (7.21)$$

şeklinde elde edilir.

Hava aralığı akısı Φ_m 'nin sabit kalması koşulu altında, devrilme de momenti sabit kalmaktadır. Bu koşul aynı zamanda $f_{r_{max}}$ sabit kalmasından dolayı, devrilme kaymasının da sabit kalmasını sağlar.

Ancak pratik uygulamalarda, statorda endüklenen gerilim $\vec{E}_s$ 'nin değiştirilmesi oldukça güçtür. Bu nedenle uç gerilimi V_s 'nin değişimi sağlanır. E_s/f_s oranı yerine V_s/f_s oranı sabit tutulur. Bu durumda stator kayıplarını gözönüne almak gerekir.

$$\vec{V}_s = \vec{E}_s + \vec{I}_s (R_s + jX_{s\sigma}) \quad (7.22)$$

bağıntısı yazılıp (7.1), (7.5a) ve (7.9) bağıntıları (7.22)'de yerlerine yazılırsa,

$$\vec{V}_s = k\Phi_m f_s + \vec{I}_s (R_s + j2\pi f_s L_{s\sigma}) \quad (7.23)$$

yapısına ulaşılır. Burada

$$\frac{\vec{V}_s}{f_s} = k\Phi_m + \vec{I}_s \left(\frac{R_s}{f_s} + j2\pi L_{s\sigma} \right) \quad (7.24)$$

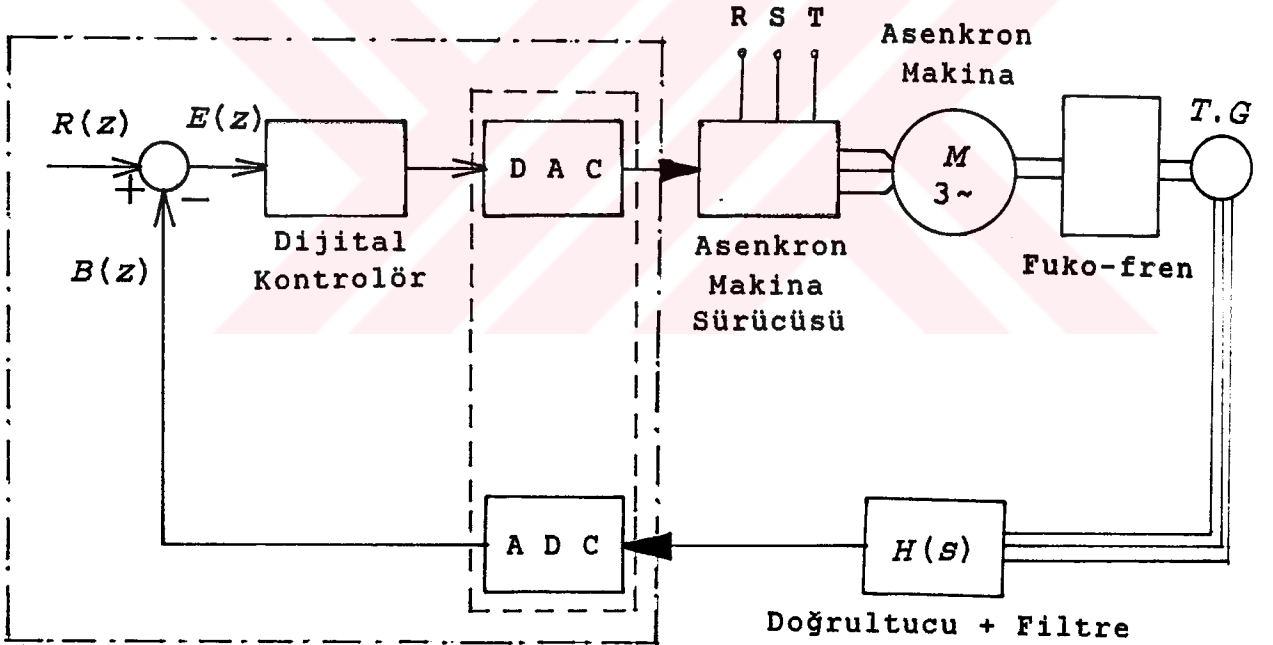
düzenlemesi yapılabilir. $\vec{V}_s/f_s$ oranı sabit tutulduğunda, f_s 'nin büyük olduğu çalışma noktalarında, R_s/f_s oranı pek önemli değildir. Ancak f_s küçüldükçe bu oran önem kazanır ve hava aralığı akısı Φ_m 'nin azalmasına neden olur. Bu nedenle belli bir f_s değerinin altında $\vec{V}_s/f_s$ =sabit ilkesi uygulanamaz. Bu f_s değerinin altında, V_s belli bir sınır değerinde sabit tutulup, f_s azaltılır. Hatta düşük hız değerlerinde ve ilk yolalma anında daha büyük yolalma momenti elde etmek için V_s arttırılabilir. Buna skaler kontrolde IR kompanzasyonu adı verilir.

BÖLÜM 8

UYGULAMADA KULLANILAN SİSTEMİN TANITIMI

Bu bölümde tez çalışmasının gerçekleştirildiği sistemin ayrıntılı tanıtımı yapılacaktır. Bu amaçla ilk önce sistemin ayrıntılı blok diyagramı verilip ardından sistemde yer alan tüm birimler ayrı ayrı tanıtılacaktır.

8.1. UYGULAMADA KULLANILAN SİSTEMİN BLOK DİYAGRAMI



Şekil 8.1. Uygulamada kullanılan sistemin blok diyagramı

Kontrol edilcek sistem olarak, rotoru sargılı (Bilezikli) asenkron makina ve sürücüsü ele alınıp bu ikiliye tek bir sistem gözü ile bakılmıştır. Uygulanada bilezikli asenkron makinanın rotor sargı uçları kısadevre yapılmış

olup sincap kafesli asenkron makina gibi davranması sağlanmıştır. Kişisel bilgisayarda (PC) turbo Pascal dilinde yazılan PID kontrolör algoritması, dijital PID kontrolör işlevini yerine getirmiştir. Bu kontrolör algoritması, optimum katsayılarının bulunması amacı ile bir altprogram olarak Yeni Global Arama programının içine yerleştirilmiştir.

$e(z)$ hata işaretine bağlı kontrolör çıkış işareti $uk(z)$, Dijital-Analog Dönüştürücü (DAC)'ye uygulanıp sürekli hale getirilir. Sürekli kontrol işareti doğru gerilim olarak 0-10V arasında değişmektedir. Bu işaret doğrudan sürücünün kontrol girişine uygulanır. Sürücü, gelen kontrol işaretine bağlı, frekansı 0.5 - 60 Hz ve genliği 50 - 380 V arasında değişen Darbe Genişlik Modülasyonlu gerilimi asenkron motorun statoruna uygular.

Asenkron motor, miline bağlı EDDY akımı freni (Dinamofren) ile anma mil gücü ve momentine kadar yüklenebilir. Dinamofren'in motora gösterdiği yük, hızın karesi ile orantılı değişmektedir.

Aynı mile bağlı takogeneratör üzerinden hızla ilişkin geribesleme bilgisi, üç fazlı alternatif gerilim olarak alınır. Takogeneratör anma yükü ile yüklenmesi halinde tamamen hızla doğru orantılı gerilim üretmektedir. Elde edilen üç fazlı alternatif hız bilgisi, üç fazlı tamdalga köprü doğrultucu ile doğrultulur ve filtre edilir. Doğrultucunun çıkışına bağlı gerilim bölücü üzerinden, hızın sıfır ve maksimum değerlerine karşılık doğru gerilim şeklinde 0 - 10 V arasında değişen geribesleme işareti elde edilir. Bu işaret bilgisayar ortamına aktarılmak üzere Analog-Dijital Dönüştürücü (ADC)'ye uygulanır. ADC çıkışında elde dijital hız geribeslemesi bilgisi, PID algoritması içinde işlenmek üzere, toplama noktası işlevini gören program satırında, hata işaretinin üretiminde kullanılır.

8.2. UYGULAMADA KULLANILAN ASENKRON MAKİNANIN TEKNİK VERİLERİ

Kontrol edilen sistemde yer alan SIEMENS marka rotoru sargılı (Bilezikli) asenkron makinanın plaka değerleri aşağıda verilmiştir.

Stator devresine ilişkin olarak:

	Yıldız	Üçgen
U_N	: 380 V	220 V
I_N	: 9.9 A	17.1 A
f_N	: 50 Hz	
P_N	: 4 kW	
M_N	: 27.1 Nm	
n_N	: 1410 dev/dak	

Rotor devresine ilişkin olarak:

(Rotor sargıları yıldız bağlıdır)

U_{Nr}	: 130 V
I_{Nr}	: 21 A
E_{x0}	: 130 V (Rotor uçları açık durumda endüklenen gerilim)

Asenkron makina parametreleri ise aşağıdaki gibidir. Bu parametreler belirlenirken, stator ve rotora ilişkin direnç ve endüktans değerleri RLC-metre ile ölçülmüştür. Diğer parametreler ise boşa çalışma, kısa devre ve yavaşlama deneyleri ile belirlenmiştir.

R_s	: 1.07 Ω
L_s	: 54.18 mH
R_r	: 0.52 Ω
L_r	: 10.73 mH
X_{rs}	: 2.406 Ω
X_{ss}	: 2.406 Ω
R_{fe}	: 322.42 Ω
X_M	: 39.57 Ω
X_s	: 17 Ω

$$\begin{aligned}
X_r &: 3.371 \Omega \\
\tau &: 12.5 \text{ s} \\
\beta &: 0.00646 \text{ Nms} \\
J &: 0.0872 \text{ kgm}^2
\end{aligned}$$

Asenkron makinayı yüklemek için kullanılan dinamofren 220 V - 4.5 A (max) doğru gerilimle beslenip, 3000 dev/dak'da 7.5 kW, 1500 dev/dak'da 6 kW 'lık aktif yükle sistemi yükleyebilir.

Hız geribeslemesinin alındığı takogeneratör ise 1000 dev/dak anma hızında, 6 VA anma yükünde, 30V fazarası gerilim üretebilen bir sistemdir. Maksimum 10000 dev/dak kadar mekanik olarak dayanıp, ölçme yapılabilir. Anma yükü ile yüklü olduğu sürece çok doğru bir lineerliğe sahiptir.

8.3. UYGULAMADA KULLANILAN ASENKRON MAKİNA SÜRÜCÜSÜNÜN TEKNİK VERİLERİ VE ÖZELLİKLERİ

Kullanılan sürücü ABB (Asea Brown Boveri) firmasına ait SAMI ministar 12 MB4 tipi cihazdır. Cihazın plaka değerleri aşağıdaki gibidir.

Lineer ve karesel yük için (IP20 / IP54)

$$\begin{aligned}
U_N &: 380 \text{ V} \\
U_{Nmax} &: 415 \text{ V} \\
I_N &: 17 \text{ A} \\
P_N &: 7.5 \text{ kW}
\end{aligned}$$

Karesel yük (Pompa ve Fanlar için)

$$\begin{aligned}
U_N &: 380 \text{ V} \\
U_{Nmax} &: 415 \text{ V} \\
I_N &: 24 \text{ A} \\
P_N &: 11 \text{ kW}
\end{aligned}$$

Cihaz giriş gerilimi frekansı 48 - 63 Hz arasında olup, güç faktörü yaklaşık 1 'dir. Çıkış gerilimi genliği üç fazlı olarak, sıfır ile anma değeri U_N arasında, çıkış frekansı ise 0.5 ile 120 Hz arasında değiştirilebilir.

Cihaz, sabit gerilim aradevrelı frekans çevirici tipindedir. Eviricinin çıkış geriliminin genlięi ve frekansı Darbe Geniřlik Modölasyonu (DGM) ile skaler kontrol yöntemine göre kontrol edilmektedir. Eviricide anahtarlama elemanı olarak Bipolar Güç tranzistörü kullanılmış olup, anahtarlama frekansı yaklaşık 2.5 kHz dolayındadır. DGM ile sistemin diyagnostik kontrolleri, koruma sistemleri ve izleme (monitoring) işlemlerinin tümü, SIEMENS SAB 80635 N mikroişlemcisi denetimi ile sağlanmış tır. DGM işareti, verilen kontrol işaretime baęlı olarak, mikroişlemci tarafından, eprom'da bulunan Look-Up Table'dan okunarak, sürücü elemanlar üzerinden eviricinin anahtarlama tranzistörlerine gönderilmektedir.

Cihazın minimum ve maksimum frekansı, verilen hız değerine ulaşma süresi (acceleration time) ve yavaşlama süresi (deceleration time), aşırı akım koruma sınırı, sabit hız değeri ile çok sayıda koruma ve frenlemeye yönelik işlevler, dijital olarak programlanabilmektedir. Buna göre IR kompanzasyonu %0 - %100 arasında, V_s/f_s oranı lineer veya karesel olarak, sabit moment bölgesinden sabit güç bölgesine geçiş frekansı (Field Weaking Point) 30-120 Hz arasında, 1.5 Hz'in altındaki frekanslarda rotoru bloke edecek doğru gerilim enjeksiyonu, generatör çalışma ve frenlemede aradevre geriliminin yükselmesini önleyecek dinamik frenleme gibi işlevlerin, dijital olarak ön ayarlanabilme olanağı vardır. Cihazın ön yüzünde bulunan dijital gösterge sistemi ile çalışma anında motora uygulanan gerilimin frekansı, referans frekanslar, referans frekanslar arası fark, % cinsinden motor akımı, momentı, aktif gücü ve aradevre gerilimi ile soęutucu sıcaklığı doğrudan izlenebilir.

8.4. UYGULAMADA KULLANILAN GİRİŞ-ÇIKIŞ KARTININ TEKNİK VERİLERİ VE ÖZELLİKLERİ

Kontrol edilen sistemle, bilgisayar ortamı arasında arabağlantıyı sağlayan giriş-çıkış kartı, PC-Lab-Card firmasının PCL 818 tipi, ayarlanabilir kazançlı, yüksek performanslı veri aktarma kartıdır. Kart bilgisayar slotlarına doğrudan bağlanabilir yapıda olup üzerinde 16 girişli analog-dijital dönüştürücü, iki adet dijital-analog dönüştürücü, 16 bit'lik dijital giriş-çıkış birimi ile 16 bit'lik programlanabilir zamanlayıcı-sayıcı ünitesi vardır.

Kart üzerindeki analog-dijital dönüştürücünün girişleri, 16 adet tek'li, 8 adet diferansiyel (ikili) yapıda olup çeşitli giriş genlik aralığına sahiptir. Çevirme tipi 'ardışıl yaklaşımlı' olup, çözünürlüğü 12 bit'dir. 100 kHz (10 μ s) örnekleme frekansında çalışıp, yazılımla, kart üzerinden programlı veya dışardan tetiklenebilir. Veri aktarımı programın içine olduğu gibi doğrudan belleğe de (DMA) yapılabilir.

Dijital-analog dönüştürücü iki kanallı olup, 12 bit çözünürlüğe sahiptir. Referans ve çıkış gerilimleri 0-10V (DC) ile +/- 10V (AC) arasında olabilir. Çevirme tipi 12 bit monolitik çarpma olup, +/-0.5 bit lineerliğe sahiptir. Çevirme süresi 5 μ s'dir.

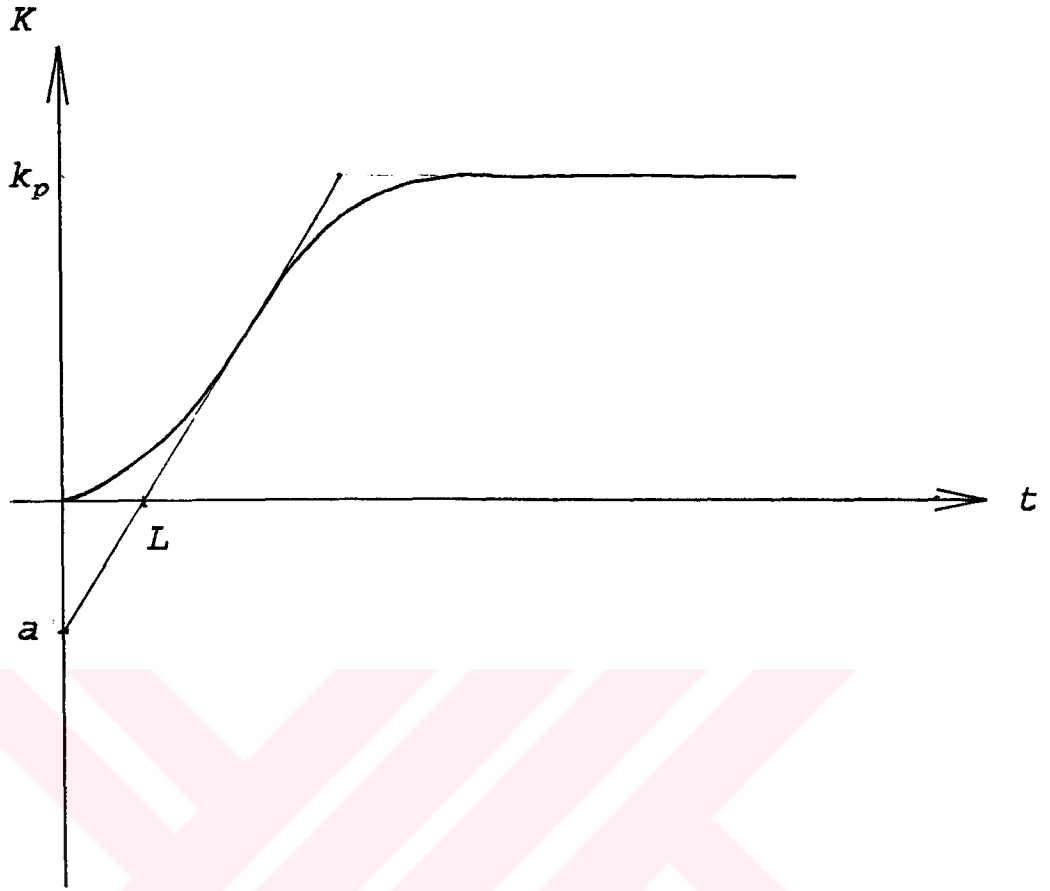
Programlanabilir zamanlayıcı/sayıcı, Intel 8254'dür. Sayıcılar, 3 kanallı 16 bit olup, ikisi programlanabilir pacer olarak ayrılmış olup, biri de kullanıcıya ayrılmıştır. Pacer çıkışı 0.00023 Hz (71 dakika/darbe) ile 2.5 MHz arasında ayarlanabilir. Ayrıca kesme ve doğrudan belleğe erişim için kanalları vardır. Uygulamada kullanılan kişisel bilgisayar ise 80386 mikroişlemcili 25 MHz işletim hızına sahiptir. Ayrıca matematik işlemler ve kayan nokta (floating point) aritmetiğini hızlandırmak amacı ile 80387 (coprocessor) ile desteklenmiştir.

BÖLÜM 9

GLOBAL ARAMANIN YAPILACAĞI BÖLGENİN SINIRLARININ ZEIGLER-NICHOLS YÖNTEMİNDEN YARARLANARAK BELİRLENMESİ

Zeigler ve Nichols 1942 yılında [14] lineer sistemlerin kontrolunda kullanılan PID kontrolör parametrelerinin belirlenmesi için iki yöntem sunmuşlardır. Sundukları yöntemler, çeşitli düzeltmelerle, bugüne dek yaygın olarak kullanılagelmıştır. Bu yöntemler "Zeigler-Nichols Basamak Yanıtı" yöntemi ile "Zeigler-Nichols Frekans Yanıtı" yöntemleridir.[5]

Basamak yanıtı yönteminde, açık çevrim kontrollu olarak sistemin basamak girişine verdiği yanıt üzerinde değerlendirme yapılarak PID kontrolör parametreleri belirlenmektedir. Yanıt eğrisine yükselme zamanı içinde çizilen maksimum eğimli teğetin zaman ve kazanç eksenlerini kestiği noktalardan a ve L parametreleri belirlenir. Şekil 9.1. de bu parametrelerin nasıl bulunacağı belirtilmiştir. Zeigler-Nichols, bu parametrelerden yararlanarak, P, PI, PID tipi kontrolör parametrelerinin bulunmasını sağlayan deneysel formüller geliştirmişlerdir. Tablo 9.1.'de verilen bağıntılarda P tipi oransal kontrolörün kazancı K ile, I tipi kontrolörün zaman sabiti T_i ile ve D tipi türevsel kontrolörün zaman sabiti T_d ile verilmiştir. T_p kapalı çevrimli kontrol sisteminin, geçici rejimde yaptığı salınımların periyodudur.



Şekil 9.1. Zeigler-Nichols Basamak Yanıtı Yöntemindeki Basamak Yanıt Özeğrisi

Tablo 9.1. Zeigler-Nichols Basamak Yanıtı Yöntemine göre önerilen PID kontrolör parametreleri

Kontrolör	K	T_i	T_d	T_p
P	$1/a$			$4L$
PI	$0.9 / a$	$3L$		$5.7L$
PID	$1.2 / a$	$2L$	$L/2$	$3.4L$

Zeigler-Nichols Frekans Yanıtı yöntemi, sistem dinamiği ile, çok basit biçimde, belirlenmektedir. Bu yöntem, sistemin transfer fonksiyonunun Nyquist diyagramında, negatif gerçekte eksenle kesiştiği noktanın bulunmasına dayanmaktadır. Bu nokta " k_c " kritik kazanç (Ultimate Gain) ve " t_c " kritik periyot (Ultimate period) olarak tanımlanan iki parametre ile belirlenmektedir. Transfer fonksiyonu kolayca elde edilemeyen sistemlerde bu noktaları bulmak için, P kontrolörünün kazancı kararlılık sınırına kadar arttırılır. Sistemi kararlılık sınırında osilasyona sokan P kontrolörün kazanç değeri k_c olarak belirlenir. Bu kazanç değeriindeki osilasyonun periyodu ise t_c 'yi verir. PID kontrolörünün parametreleri k_c ve t_c 'ye bağlı basit deneysel formüller olarak tablo 9.2.'de verilmiştir.

Tablo 9.2 Zeigler-Nichols Frekans Yanıtı yöntemine göre önerilen PID kontrolör parametreleri

Kontrolör	K	T_I	T_D	T_P
P	$0.5k_c$			t_c
PI	$0.4k_c$	$0.8t_c$		$1.4t_c$
PID	$0.6k_c$	$0.5t_c$	$0.12t_c$	$0.85t_c$

Tablo.9.1. ve Tablo 9.2.'de önerilen K , T_I ve T_D değerleri Zeigler ve Nichols lineer sistemler üzerine uzun süreli yaptıkları deneysel çalışmalar sonunda ortaya çıkarılmış deneysel formüllerdir. Bunların belirlenmesinde, sistemin geçici rejimi esnasında yaptığı salınımların genlikleri arasındaki fark (genlik sönüm faktörü) $1/4$ ve buna karşı düşen bağıl sönüm faktörünün $\xi=0.22$ olması esas alınmıştır. Bu kriter geçen zaman içinde çeşitli şekillerde yeniden düzenlenmiştir.

Yeni Global Arama Yönteminde sınır değerlerin belirlenmesinde Zeigler-Nichols'un frekans yanıtı yönteminde yararlanılmıştır. Bilindiği gibi asenkron makina ve sürücüsünden oluşan sistemin basit anlamda bir transfer fonksiyonunu elde etmeğe olanak yoktur. Bu nedenle frekans yanıtı yöntemi kullanılıp, sistemin çeşitli çalışma noktaları için k_c ve t_c değerleri belirlenmiştir. Yukarıda da sözedildiği gibi Zeigler ve Nichols, PID parametrelerini transfer fonksiyonu elde edilebilen lineer sistemler için ve $\xi=0.22$ bağıl sönüm kriterine göre vermiştir. Tezde üzerinde çalışılan sistem nonlineer yapıda olduğundan, lineer sistemler için önerilen parametreler istenen sonuca ulaşılmasını sağlayamamıştır. Ancak önerilen parametre değerlerinin iki katı, global aramanın gerçekleştirileceği bölgenin sınırlarının belirlenmesi için çok iyi bir yaklaşım getirmektedir. Böylece çeşitli hızlardaki çalışma noktaları için, alt sınır değeri olarak sıfır, üst sınır değeri olarak da Zeigler-Nichols yöntemindeki k_c ve t_c değerlerinden elde edilen PID parametrelerinin iki katı seçilerek belirlenen sınırlar içinde, asenkron motor ve sürücüsünden oluşan sistemi kontrol eden dijital PID kontrolörü parametrelerinin optimum değerleri, önceden belirlenen Amaç Ölçütü'ne göre, Yeni Global Arama Yöntemi ile bulunmuştur.

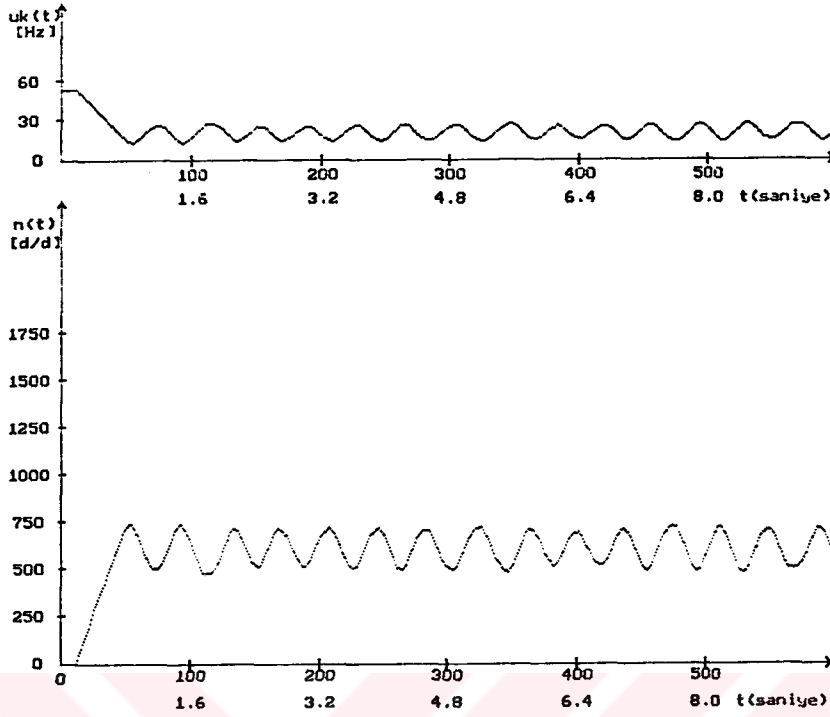
BÖLÜM 10

SONUÇLAR VE ÖNERİLER

Pratik olarak gerçekleştirilen çalışmada temel amaç, asenkron motorun, belirlenen Amaç Ölçütüne göre, geçici ve sürekli rejim altında hız kontrolunu gerçekleştirmektir. Bu amaçla da belirlenen Amaç Ölçütünü minimum yapan sistem performansını sağlayacak PID kontrolörü parametrelerinin en uygun (optimum) değerleri, çeşitli hız referansları ve yüklenme durumları için belirlenmiştir. Kontrolör parametrelerinin sınırdeğerleri bölüm 9'da anlatılan Zeigler-Nichols yönteminden yararlanarak bulunmuştur. Bu yöntemde, lineer sistemler ve sistemin geçici rejimi esnasında yaptığı ilk iki salınımın genlikleri arasındaki sönüm oranının 1/4 ve bu göre hesaplanan $\xi=0.22$ bağıl sönüm kriterine göre önerilen kontrolör parametrelerinin iki katı, sınır değer olarak seçilmiştir. Bu amaçla ilk önce çeşitli hız referansları için sistemin kritik kazanç ve kritik periyodu belirlenmiştir. Bu değerler ve tablo 9.2. deki bağıntılara göre K , T_I ve T_D belirlenip,

$$K_P = K \quad K_I = \frac{1}{T_I} \quad K_D = T_D \quad (10.1)$$

bağıntıları ile bulunan K_P , K_I ve K_D değerlerinin iki katı bu parametrelerin üst sınır değerleri olarak seçilmiştir. Alt sınır değerleri ise tüm parametreler için sıfırdır. Sistem üzerinde yapılan ölçümlerde K_P ve K_D sıfır alınıp sadece K_I 'nin değeri arttırılarak, sistemin osilasyona başladığı ilk kazanç değeri, kritik kazanç k_c , osilasyonun periyodu ise kritik periyod t_c olarak alınmıştır. Şekil 10.1'de $n_r = 1000$ dev/dak için sistemin kritik kazanç değerinde yaptığı osilasyon görülmektedir.



Şekil 10.1. $n_r = 1000$ dev/dak için yüksüz durumda sistemin kritik kazanç değerinde yaptığı osilasyonun grafiği

Bu hız referansı için kritik kazanç, $k_c = 3.85$ ve kritik periyot, $t_c = 0.64$ s olarak bulunmuştur. Tablo 9.2.'deki PID katsayı değerlerinden,

$$K_p = 0.6 * 3.85 = 2.31$$

$$T_I = 0.5 * 0.64 = 0.32 \text{ s}$$

$$T_D = 0.12 * 0.64 = 0.0768 \text{ s}$$

olarak belirlenip,

$$K_I = 1 / T_I = 3.125$$

$$K_D = T_D = 0.462$$

şeklinde bulunur. Buna göre $n_r = 1000$ dev/dak hız referansı için yüksüz durumda, arama bölgesinin sınırları, yukarıda elde edilen K_p , K_I , K_D değerlerinin iki katı, üst sınır değer olarak alındığında

$$0 < K_p < 4.62$$

$$0 < K_I < 6.25$$

$$0 < K_D < 0.924$$

olarak belirlenirler. Sistem üzerinde yapılan çok sayıdaki ölçümler sonucunda, 500,750,1250 ve 1500 dev/dak'lık hız

referansları için bulunan kritik kazanç ve kritik periyot değerleri aşağıdaki tabloda sunulmuştur. Yüksüz durumda elde edilen k_c ve t_c değerlerinin yüklü durumdaki değerleri de araştırılmıştır. Ancak çalışmanın yapıldığı laboratuarda bulunan yüklerin, (Dinamo frenler) oluşturdıkları moment hızla karesel orantılı olarak değiştiğinden, sistem osilasyona girdiği zaman, değişen hız durumlarında, sabit yüklerle yüklenememiştir. Bu durumda da yapılan ölçümler sağlıklı sonuçlar vermemiştir. Ancak yüksüz çalışma durumunda belirlenen sınır değerler, yüklü çalışma durumları için de tam olarak yeterli olmuştur.

Tablo 10.1. Çeşitli referans hız değerleri için kritik kazanç, kritik periyot ve kontrolör parametrelerinin üst sınır değerleri

İstenen Hız (dev/dak)	Kritik Kazanç (k_c)	Kritik periyot t_c [s]	Üst sınır değerler
1500	9	0.71	$K_p : 10.8$ $K_I : 5.63$ $K_D : 0.17$
1250	3.85	0.64	$K_p : 4.31$ $K_I : 6.25$ $K_D : 0.153$
1000	3.85	0.64	$K_p : 4.31$ $K_I : 6.25$ $K_D : 0.153$
750	4.5	0.64	$K_p : 4.31$ $K_I : 6.25$ $K_D : 0.153$
500	4.3	0.592	$K_p : 5.16$ $K_I : 6.75$ $K_D : 0.142$

Sınır değerlerin belirlenmesinin ardından tanımlanan Amaç Ölçütünü minimum yapacak kontrolör parametrelerinin optimum değerleri bulunmuştur. Bu amaçla Yeni Global Arama Yöntemi ile K_p , K_I ve K_D parametrelerine bağlı olarak minimum değeri bulunacak olan AMAÇ ÖLÇÜTÜ'nün (P.I.), tanımı yapılmıştır. İlk durumda Amaç Ölçütü sistemin basamak girişe verdiği yanıt eğrisinden yararlanarak belirlenen t_r (Yükselme süresi), p.o. (Aşım), e_{ss} (Kararlı Hal Hatası), ölçütlerine bağlı olarak

$$P.I. = C_1 \cdot t_r + C_2 \cdot po + C_3 \cdot e_{ss} \quad (9.1)$$

şeklinde tanımlanmıştır. Burada C_1 , C_2 ve C_3 katsayıları CEZA Katsayıları'dır. En iyi yanıt eğrisine ulaşılması durumunda po ve e_{ss} yaklaşık sıfır olup, t_r 'de minimum değerine ulaşmalıdır. Böylesi yanıt eğrisinin elde edilmesini sağlayan K_p , K_I , K_D değerleri, P.I.'i minimum yapacak ve aranılan optimum değerler olacaklardır.

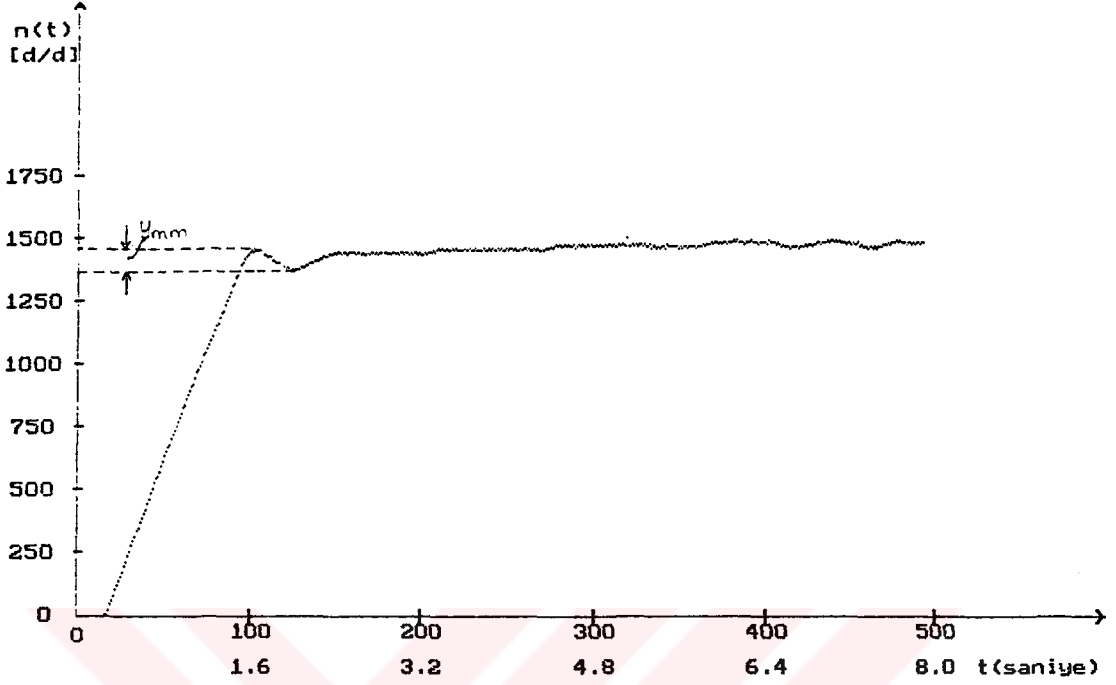
Yapılan çalışmalar esnasında sistemin belirli kontrolör katsayılarına karşılık, ilginç bir yanıt eğrisi verdiği gözlenmiştir. Şekil 10.2.de görülen bu yanıt tipinde sistem aşım ve kararlı hal hatası göstermemekte ancak istenen değere otururken küçük bir salınım yapmaktadır. Bu salının (10.2)'de tanımlanan kriterler ile Amaç Ölçütü'ne katkılanamamaktadır. Bu istenmeyen salınım etkisinden kurtulmak için, geri besleme işareti üzerinde salının miktarı ölçülüp, referansa bağlı % cinsinden değeri belirlenerek, P.I. bağıntısına katkılanır. Böylece sistemin yaptığı salınım

$$gd = \frac{Y_{mm}}{rf} \cdot 100 \quad (10.3)$$

şeklinde ölçülüp, Amaç Ölçütü

$$P.I. = -(C_1 \cdot t_r + C_2 \cdot po + C_3 \cdot e_{ss} + C_4 \cdot g.d.) \quad (10.4)$$

şeklinde yeniden tanımlanmıştır.

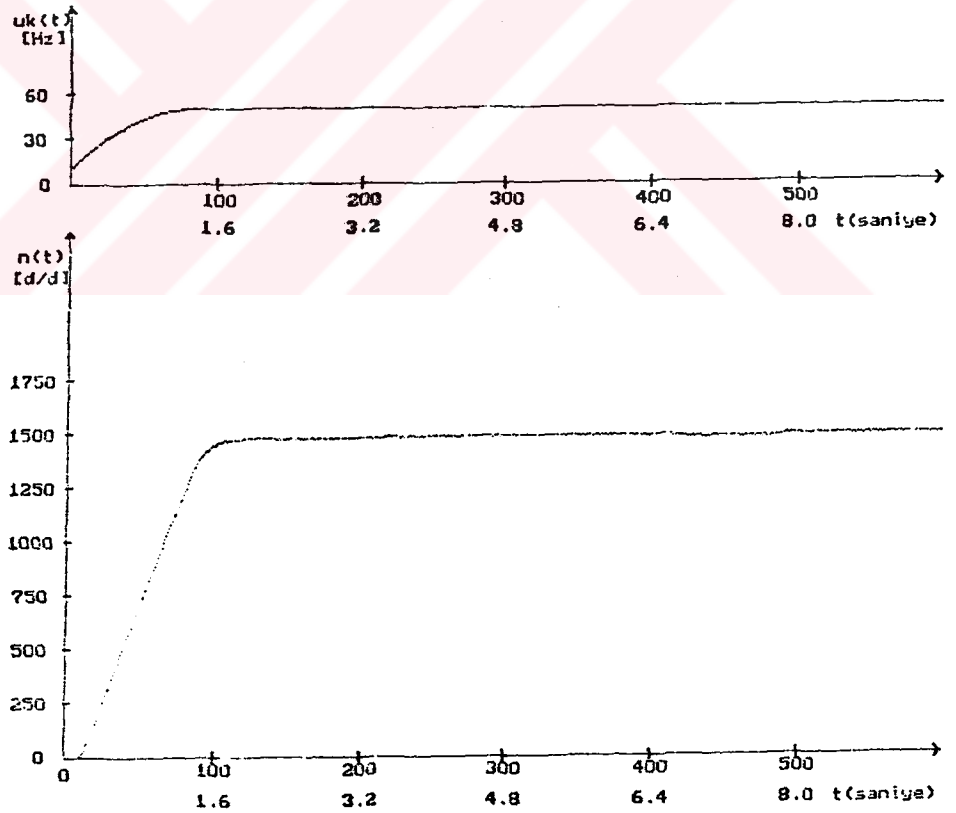


Şekil 10.2. Sistemin kararlı duruma otururken yaptığı istenmeyen salınımı içeren hız grafiği

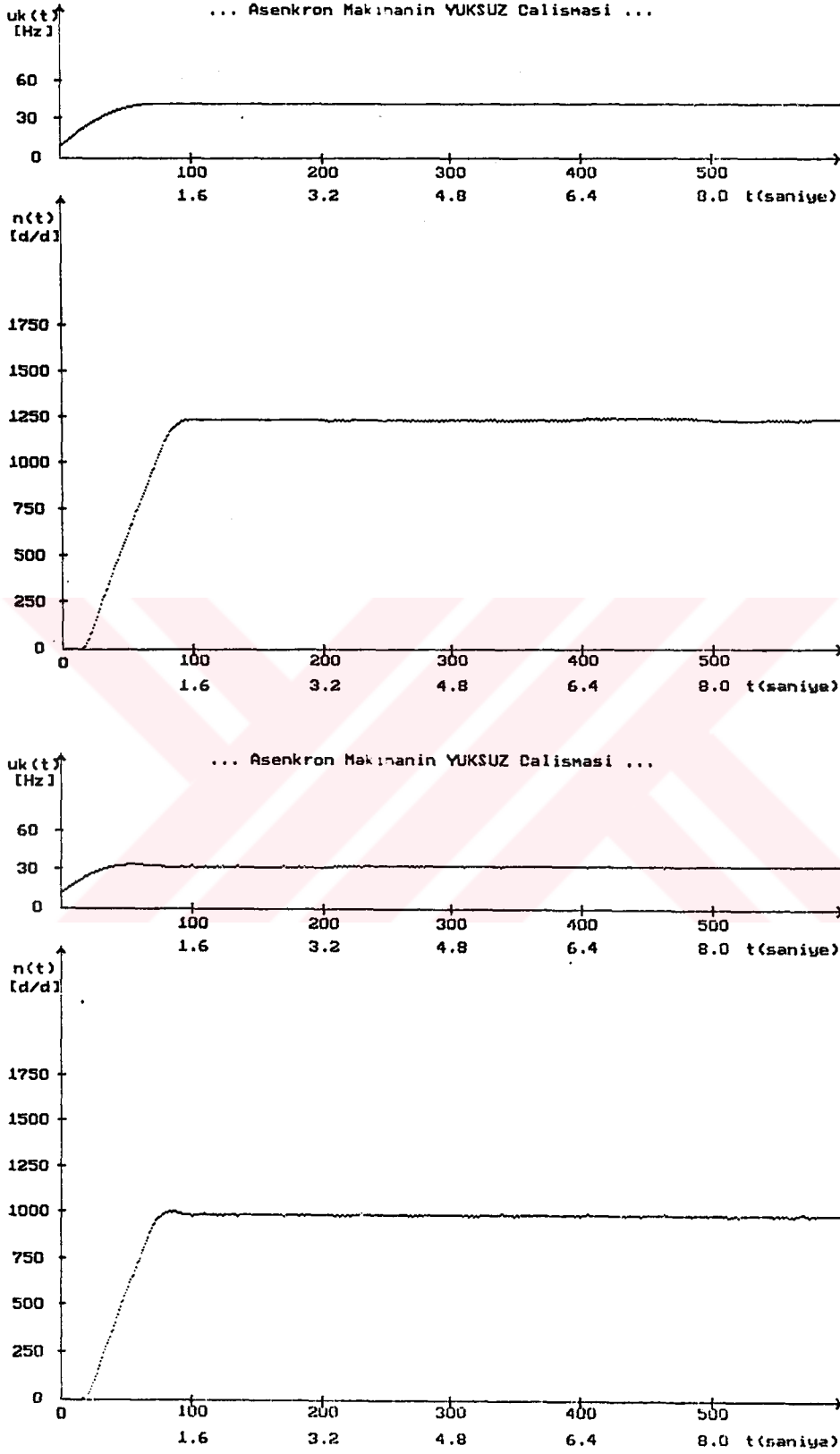
Yukarıda da sözedildiği gibi C_1 , C_2 , C_3 , C_4 ceza katsayılarıdır. Bu katsayılar belirlenmesinde başlıca iki yöntem izlenir. İlk yöntemde, geçici rejim ölçütleri aynı miktarda etki ile P.I.'e katkılansın istenebilir. Bu durumda P.I.'in bileşenlerinin herbiri, yaklaşık aynı nicel değeri alabilmesi için uygun ceza katsayıları değerleri belirlenir. İkinci yöntemde ise hangi geçici rejim ölçütü daha öncelikli ise yani daha küçük olması isteniyorsa onun katsayısı daha büyük seçilir. Böylece, o ölçütde oluşan artımlar daha büyük ölçüde P.I.'e yansıtılmış olur. Yapılan çalışmada $C_1 = 10$, $C_2 = 30$, $C_3 = 50$, $C_4 = 100$ seçilmişlerdir. Bu ceza katsayıları ve (10.4)'de tanımlanan Amaç Ölçütü'ne göre sistemin değişik çalışma noktalarına ilişkin en uygun PID kontrolör parametreleri aşağıdaki tablolarda verilip bu çalışma noktalarına ilişkin hız ve kontrol işaretinin zamanla değişim grafikleri sunulmuştur.

Tablo 10.2. Sistemin YÜKSÜZ durumda çalışmasına ilişkin PID kontrolör parametrelerinin optimum değerleri

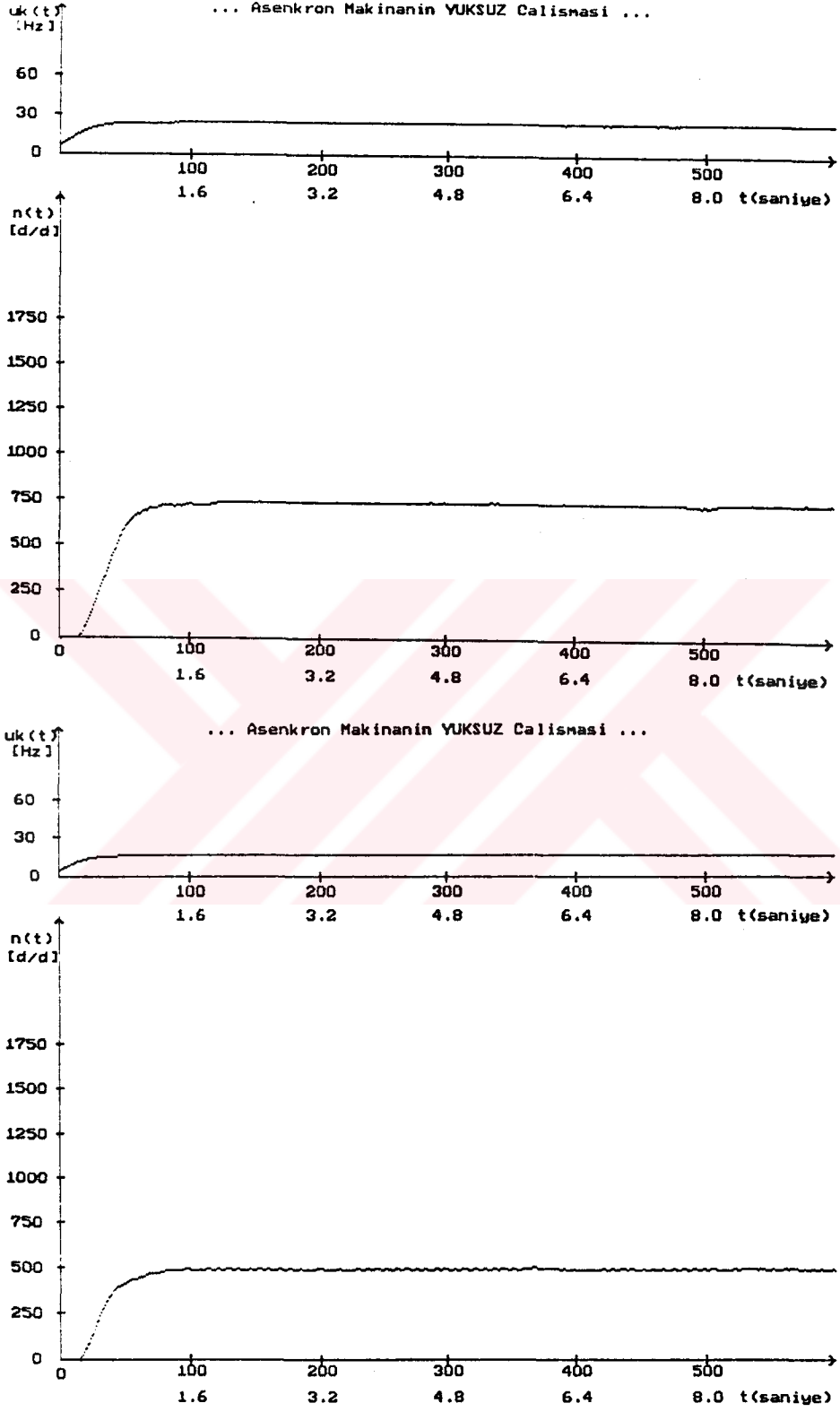
İstenen Hız (dev/dak)	K_p	K_I	K_D
1500	0.482	2.835	0.004
1250	0.77	2.867	0.01
1000	0.84	3.33	0.011
750	0.7	3.57	0.012
500	0.827	4.71	0.012



Şekil 10.3. $n_r = 1500$ dev/dak 'da YÜKSÜZ çalışma durumunda kontrol işareti ve Hız grafikleri.



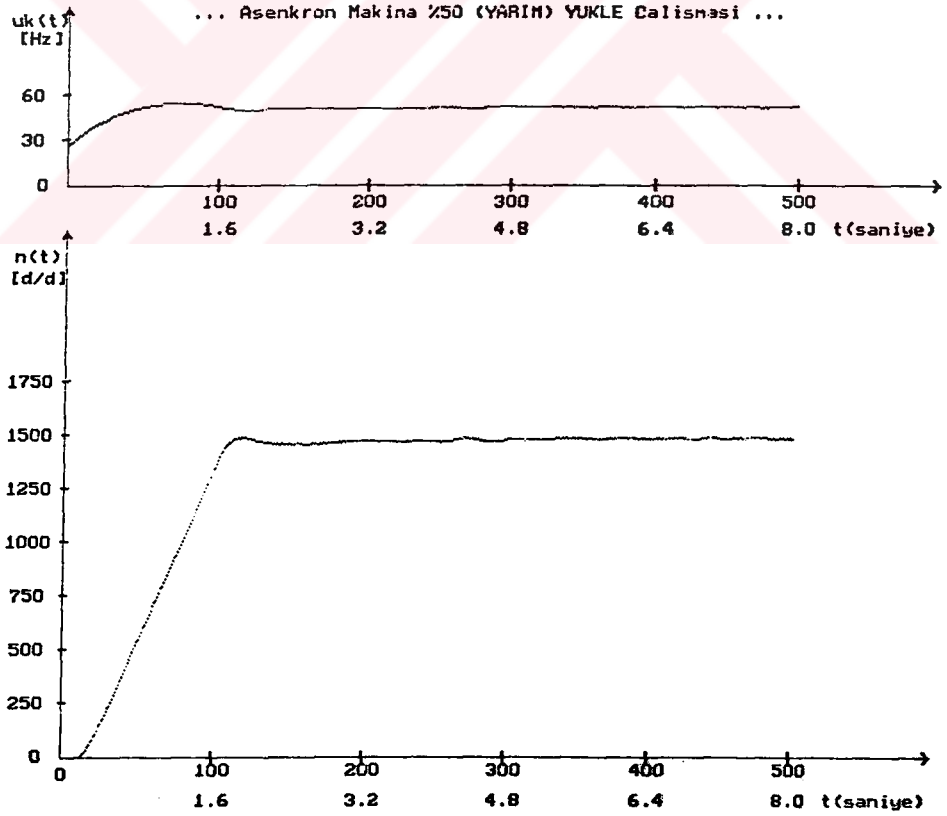
Şekil 10.4. $n_1 = 1250$ dev/dak ve $n_1 = 1000$ dev/dak'da YÜKSÜZ çalışma durumunda, kontrol işareti ve Hız grafikleri.



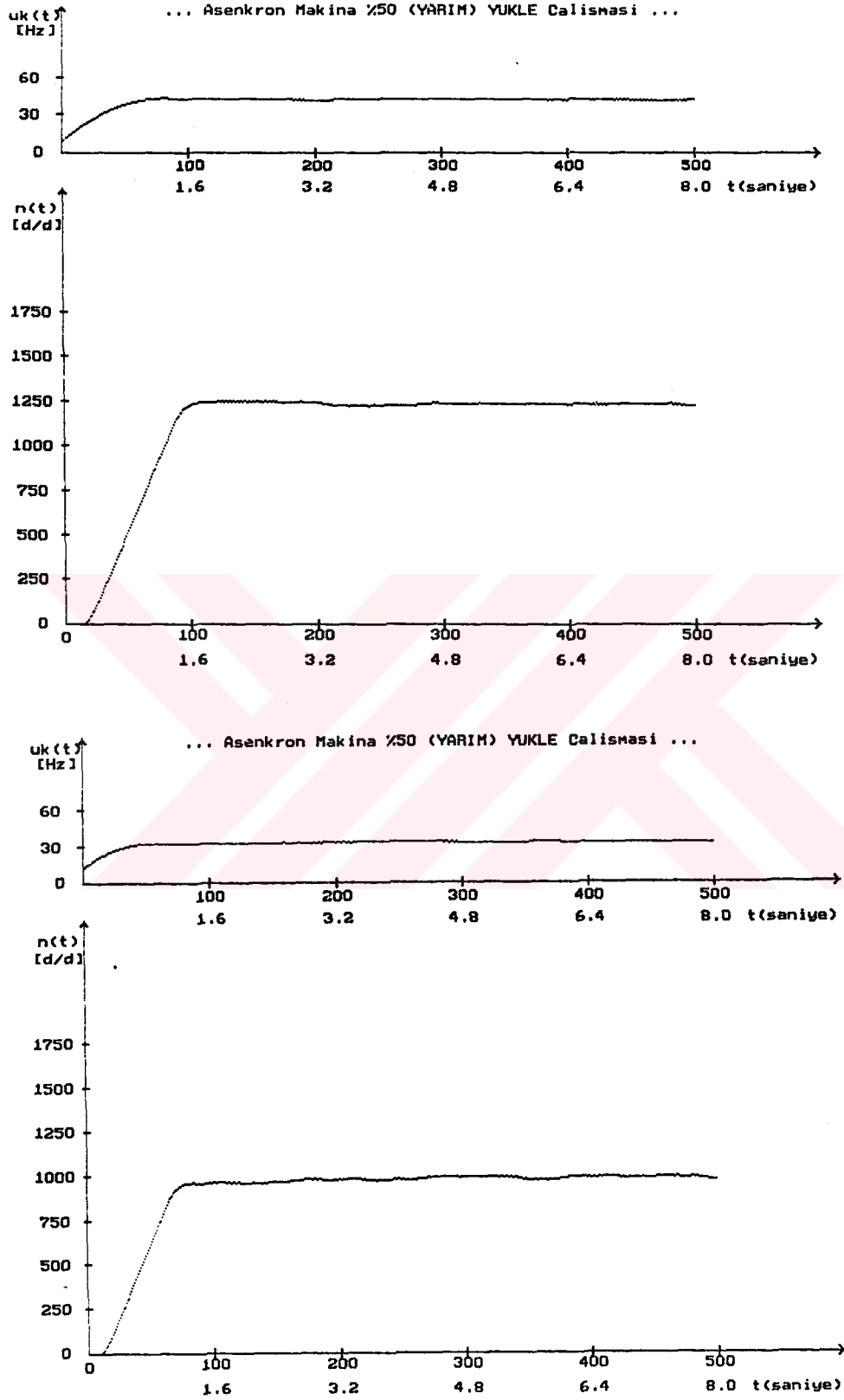
Şekil 10.5. $n_r = 750$ dev/dak ve $n_r = 500$ dev/dak'da YÜKSÜZ çalışma durumunda, kontrol işareti ve Hız grafikleri.

Tablo 10.3. Sistemin %50 YÜKLE çalışmasına ilişkin,
PID kontrolör parametrelerinin optimum değerleri

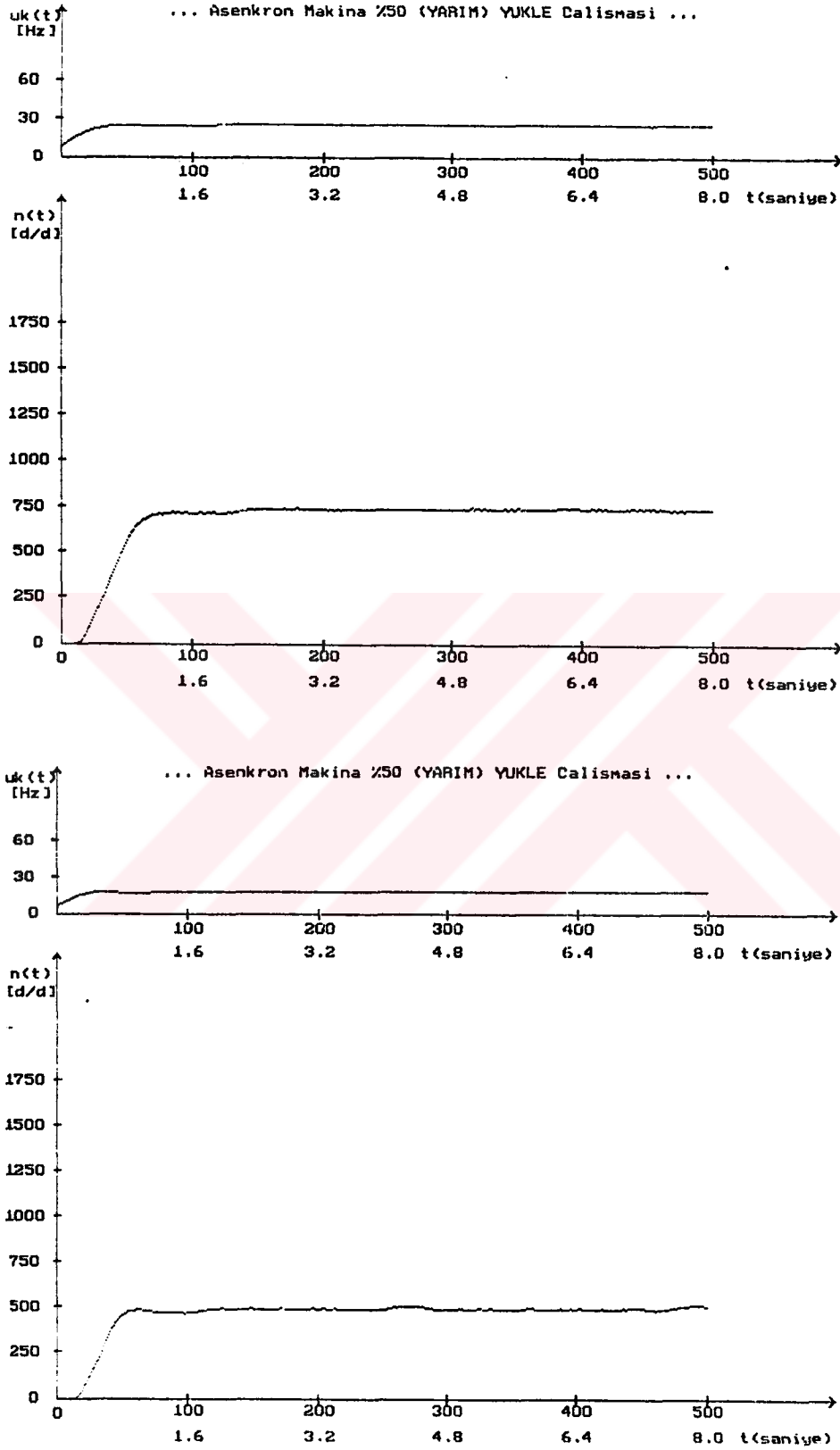
İstenen Hız (dev/dak)	K_p	K_I	K_D
1500	1.11	2.27	0.015
1250	1.066	2.834	0.00752
1000	1.07	3.337	0
750	0.98	3.635	0.008
500	1.202	5.052	0.01315



Şekil 10.6. $n_r = 1500$ dev/dak 'da, %50 YÜKLÜ çalışma durumunda kontrol işareti ve hız grafikleri.



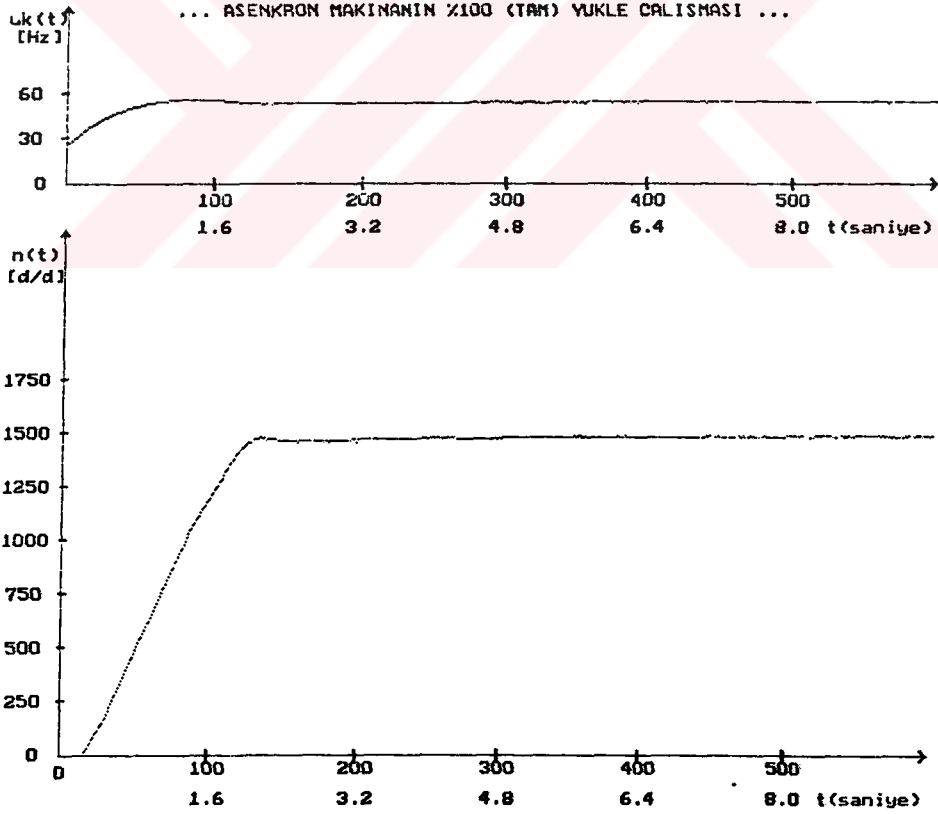
Şekil 10.7. $n_r = 1250$ dev/dak ve $n_r = 1000$ dev/dak'da %50 YÜKLÜ çalışmada, kontrol işareti ve hız grafikleri.



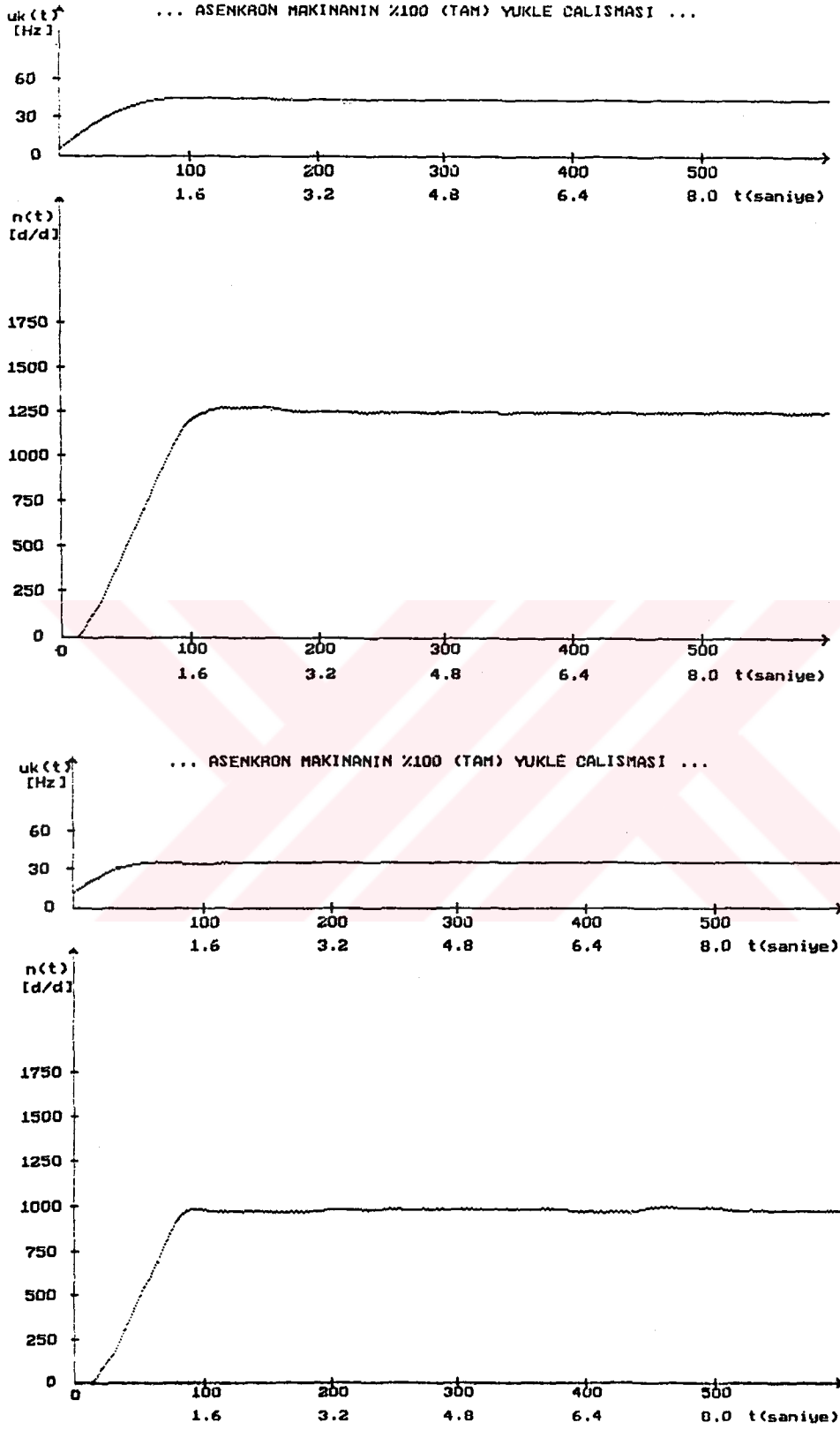
Şekil 10.8. $n_r = 750$ dev/dak ve $n_r = 500$ dev/dak'da %50 YÜKLÜ çalışmada kontrol işareti ve Hız grafikleri.

Tablo 10.4. Sistemin %100 YÜKLE çalışmasına ilişkin,
PID kontrolör parametrelerinin optimum değerleri

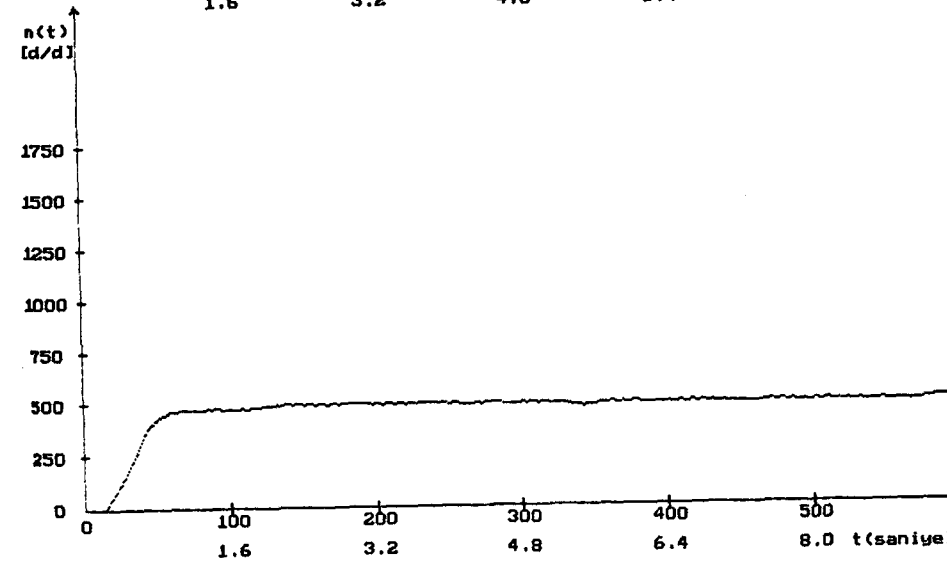
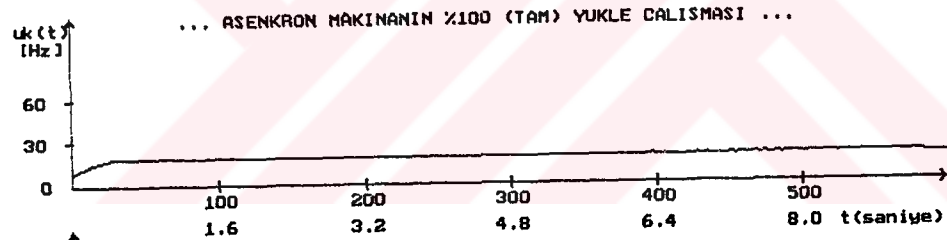
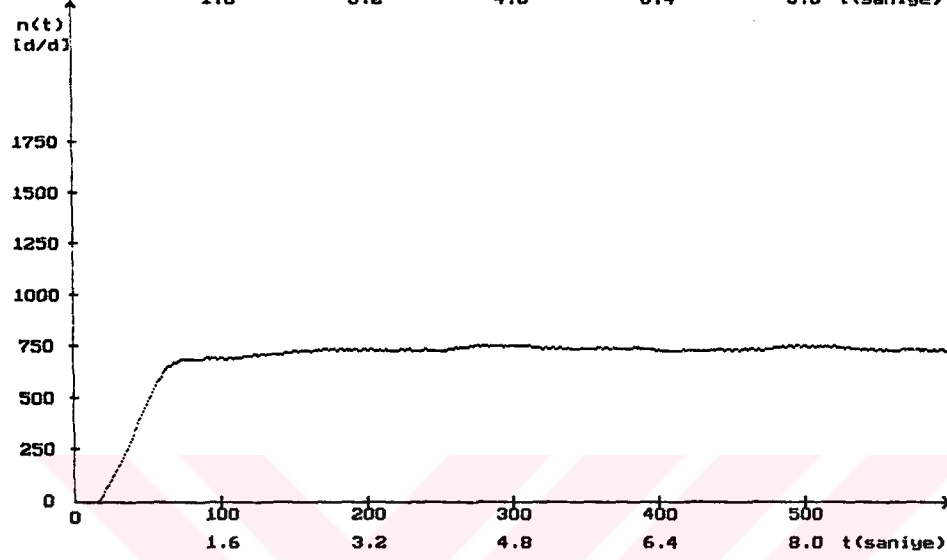
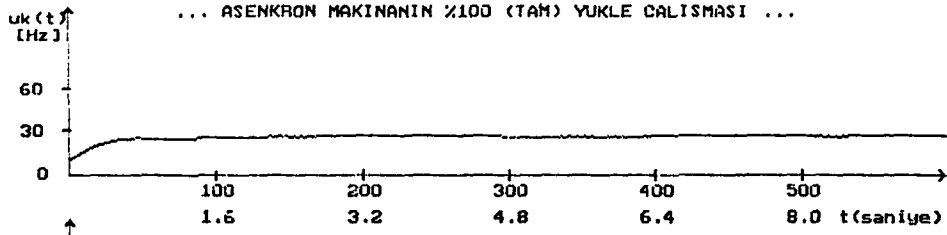
İstenen Hız (dev/dak)	K_p	K_I	K_D
1500	1.33	2.33	0.011
1250	1.615	2.55	0.00646
1000	0.848	3.068	0.0053
750	1.481	3.586	0.0148
500	1.27	4.754	0.00205



Şekil 10.9. $n_r = 1500$ dev/dak ve %100 YÜKLÜ çalışma durumunda kontrol işareti ve hız grafikleri.



Şekil 10.10. $n_r = 1250$ dev/dak ve $n_r = 1000$ dev/dak'da
%100 YÜKLÜ çalışmada, kontrol işareti ve Hız grafikleri.



Şekil 10.11. $n_r = 750$ dev/dak ve $n_r = 500$ dev/dak'da %100 YÜKLÜ çalışmada kontrol işareti ve hız grafikleri.

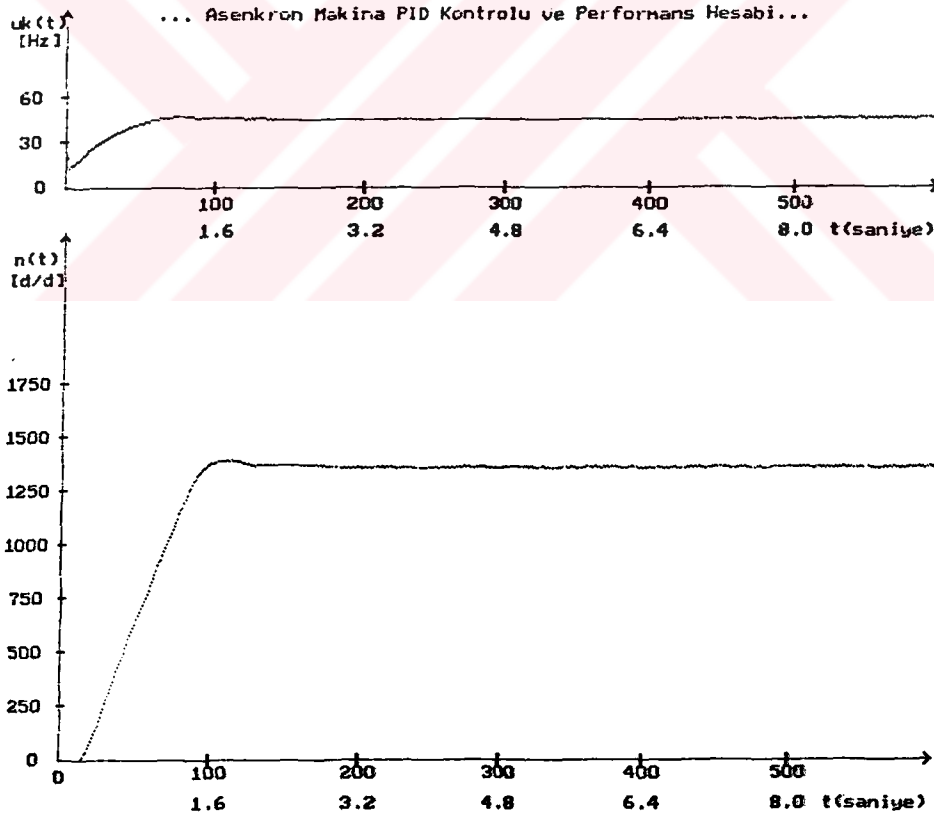
Bilindiği gibi asenkron makina, lineer olmayan (non-linear) bir sistemdir. Bu nedenle de belli bir çalışma noktası için belirlenen PID kontrolör katsayıları, bir başka çalışma noktası için optimum değer olmaktan çıkar. Eğer üzerinde çalışılan sistem, lineer yapıdaki doğru akım motoru olsa idi, bulunacak tek bir grup optimum PID kontrolör katsayıları, tüm farklı hız referansları ve yüklenme durumları için de en uygun değerler olacaktı. Doğaldır ki bu durum, doğru akım motorunun parametrelerinin zamanla değişmediği varsayımı altında geçerlidir. Ancak bu durum asenkron makina için söz konusu değildir. Bundan dolayı bu çalışmada, asenkron makina ve sürücüsünden oluşan sistem üzerinde beş değişik referans hız değeri ve üç ayrı yüklenme durumu için optimum değerler ayrı ayrı belirlenmiştir. Grafiklerden de görüldüğü gibi, optimum olarak belirlenen PID kontrolörü parametreleri, sisteme uygulandığında, sistemin geçici rejimi başarılı bir şekilde kontrol altına alınabilmektedir. Ancak, kolaylıkla tahmin edilebileceği gibi, pratik uygulamada, sistem üzerinde her farklı çalışma noktası için ölçümler yapmak uygun bir çözüm yolu değildir. Bu nedenle, sadece ölçüm yapılan noktalara ilişkin optimum değerlerden yola çıkarak, ara çalışma noktaları için de optimum değerlerin belirlenmesi amacı ile lineer enterpolasyon yöntemi kullanılmıştır. Bu yöntem yardımı ile ölçüm yapılan çalışma noktalarına ilişkin PID kontrolör katsayıları arasında

$$K_{ARA} = K_{ALT} + \left[\frac{K_{ALT} - K_{ÜST}}{n_{ALT} - n_{üst}} \right] \cdot [n_{ARA} - n_{ALT}] \quad (10.5)$$

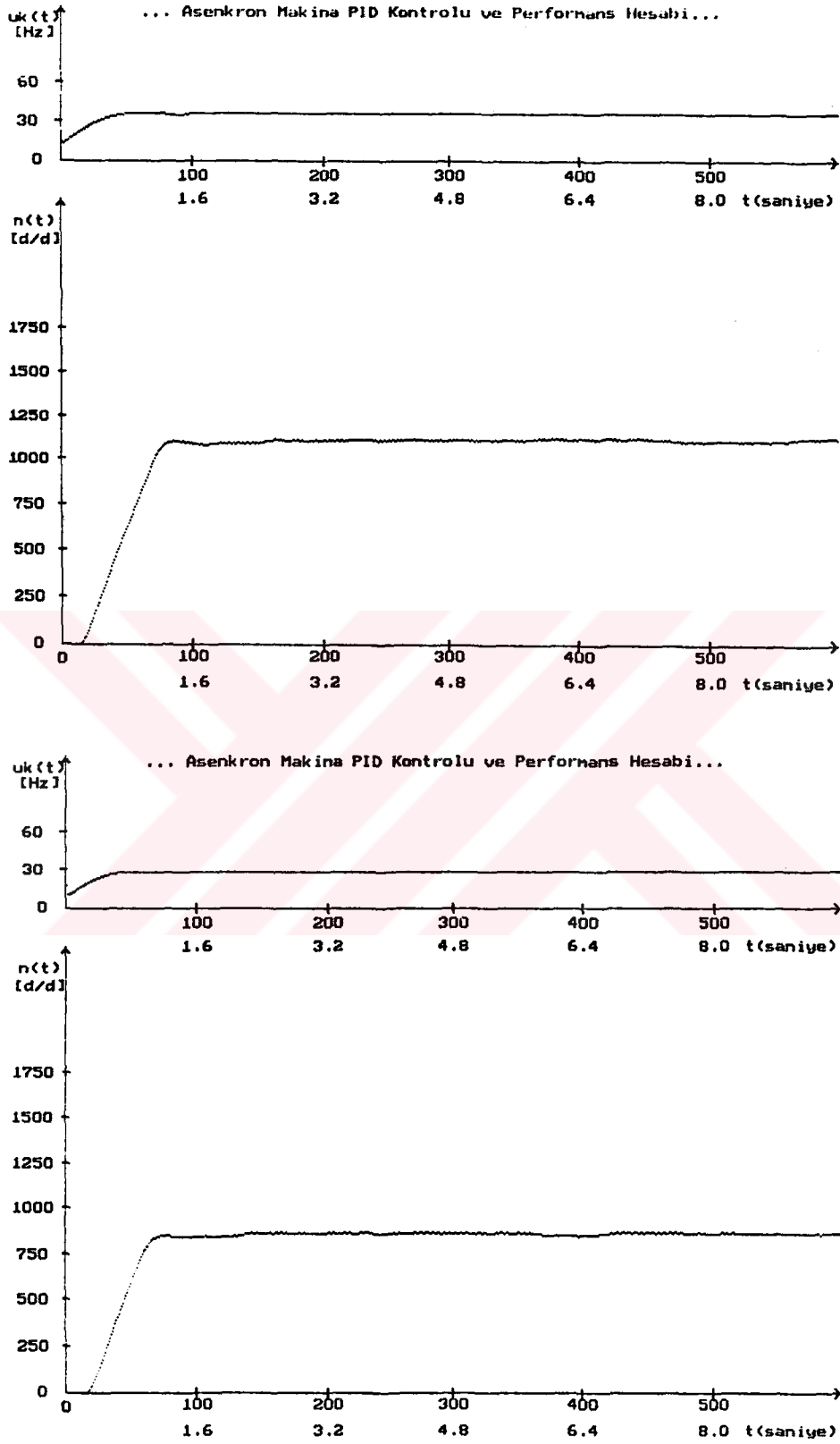
bağıntısı uyarınca, lineer doğrular oluşturulup, ara çalışma noktalarına ilişkin optimum değerler hesaplanabilir. Ara çalışma noktalarına ait, enterpolasyon yardımı ile bulunan optimum PID kontrolörü katsayıları Tablo 10.5 'de ve bu değerlere ilişkin sistem yanıtlarının yer aldığı kontrol işareti ve hız grafikleri de ilerleyen sayfalarda sunulmuştur.

Tablo 10.5. Sistemin Ara noktalarda, YÜKSÜZ çalışmasında, Enterpolasyon ile bulunan optimum PID kontrolör parametreleri

İstenen Hız (dev/dak)	K_p	K_I	K_D
1375	0.626	2.8511	0.007
1125	0.805	3.1	0.00105
875	0.77	3.45	0.0115
625	0.7635	4.135	0.012



Şekil 10.12. $n_r = 1375$ dev/dak'da YÜKSÜZ çalışmada Enterpolasyon ile bulunan optimum PID kontrolör parametrelerinde, sistemin kontrol işareti ve hız grafikleri

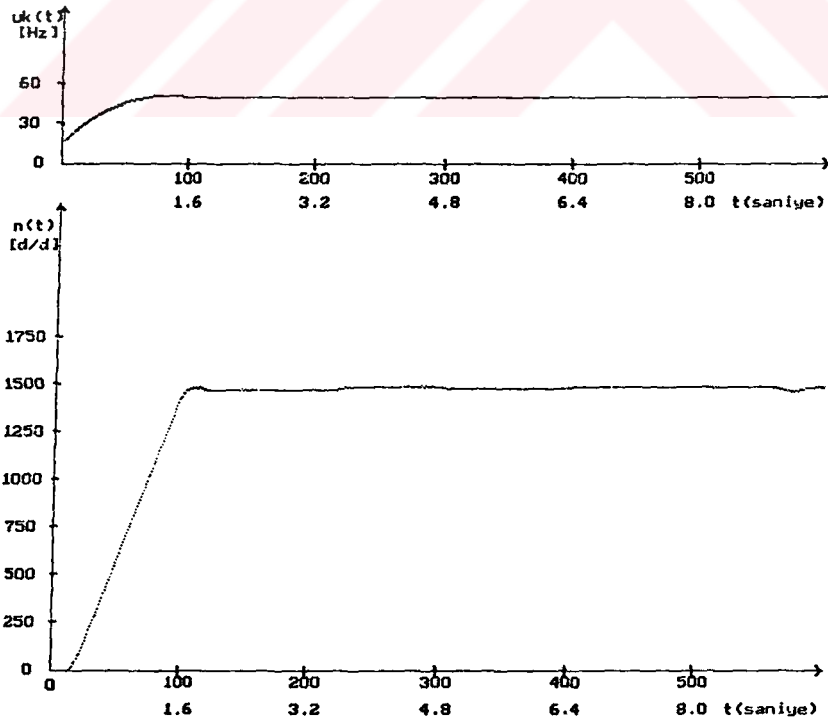


Şekil 10.13. $n_r = 1125$ dev/dak'da ve $n_r = 875$ dev/dak'daki YÜKSÜZ çalışmada, Enterpolasyon ile bulunan optimum PID kontrolör parametrelerinde, sistemin yanıt grafikleri

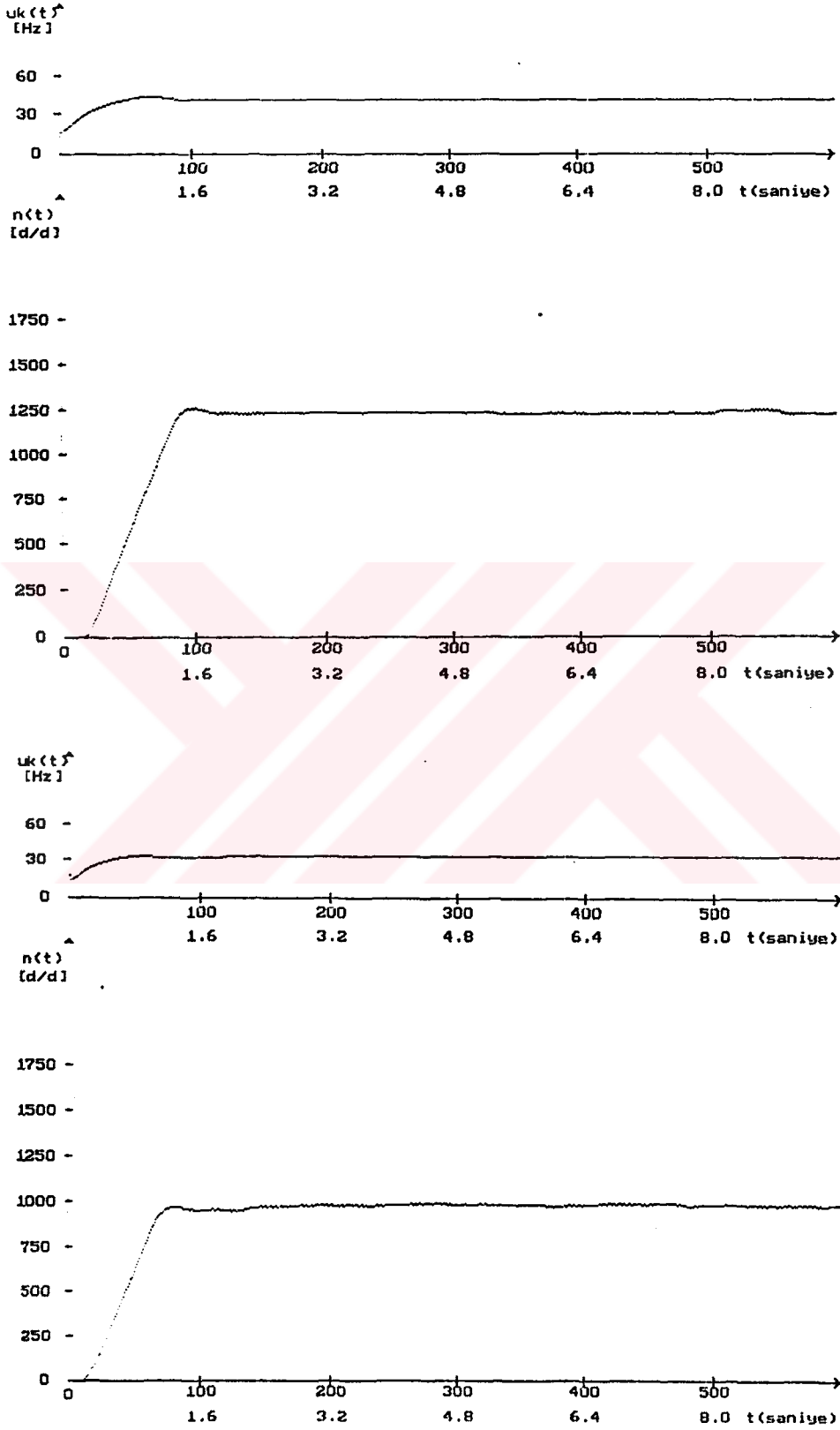
Lineer Enterpolasyon işlemi ara yüklenme durumları için de uygulanabilir. %25 ve %75 ara yüklenme durumlarına ilişkin olarak enterpolasyon ile hesaplanan optimum PID kontrolörü parametrelerine ilişkin tablolar ve sistemin yanıtları aşağıda sunulmuştur.

Tablo 10.6. Sistemin %25 ARA YÜKLE çalışmasında, Enterpolasy ile bulunan, optimum PID kontrolör parametreleri

İstenen Hız (dev/dak)	K_p	K_I	K_D
1500	0.796	2.55	0.0095
1250	0.918	2.85	0.01375
1000	0.955	3.333	0.0055
750	0.84	3.6025	0.01
500	1.0145	4.876	0.0125



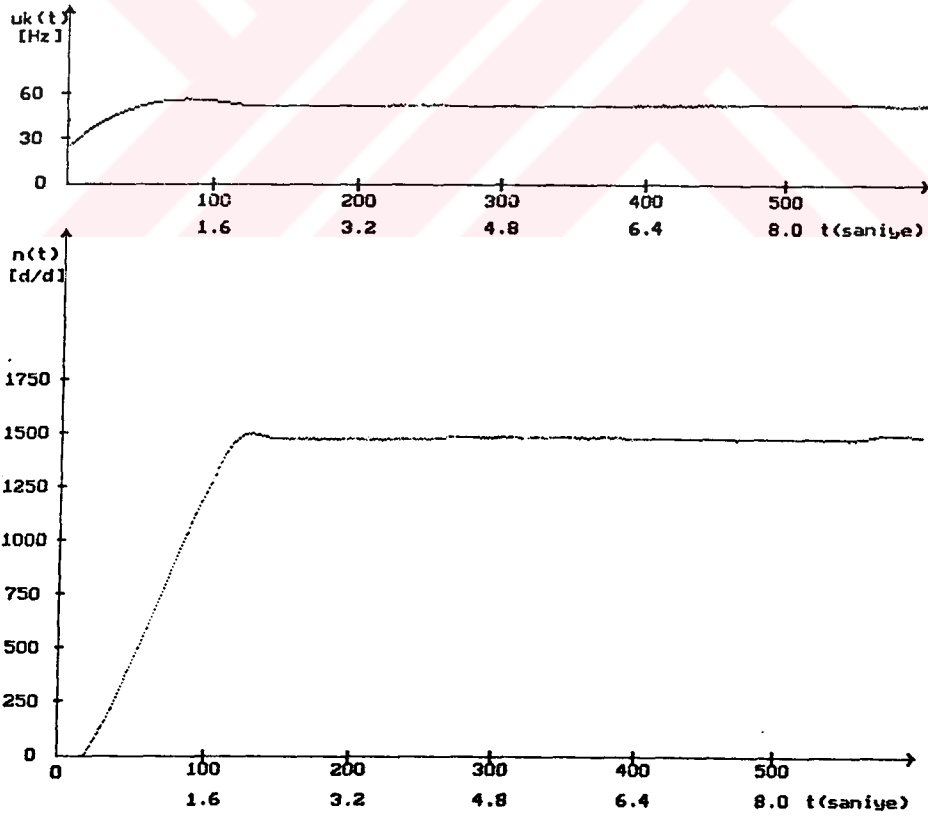
Şekil 10.14. $n_r = 1500$ dev/dak 'da %25 YÜKLÜ çalışma durumunda kontrol işareti ve hız grafikleri.



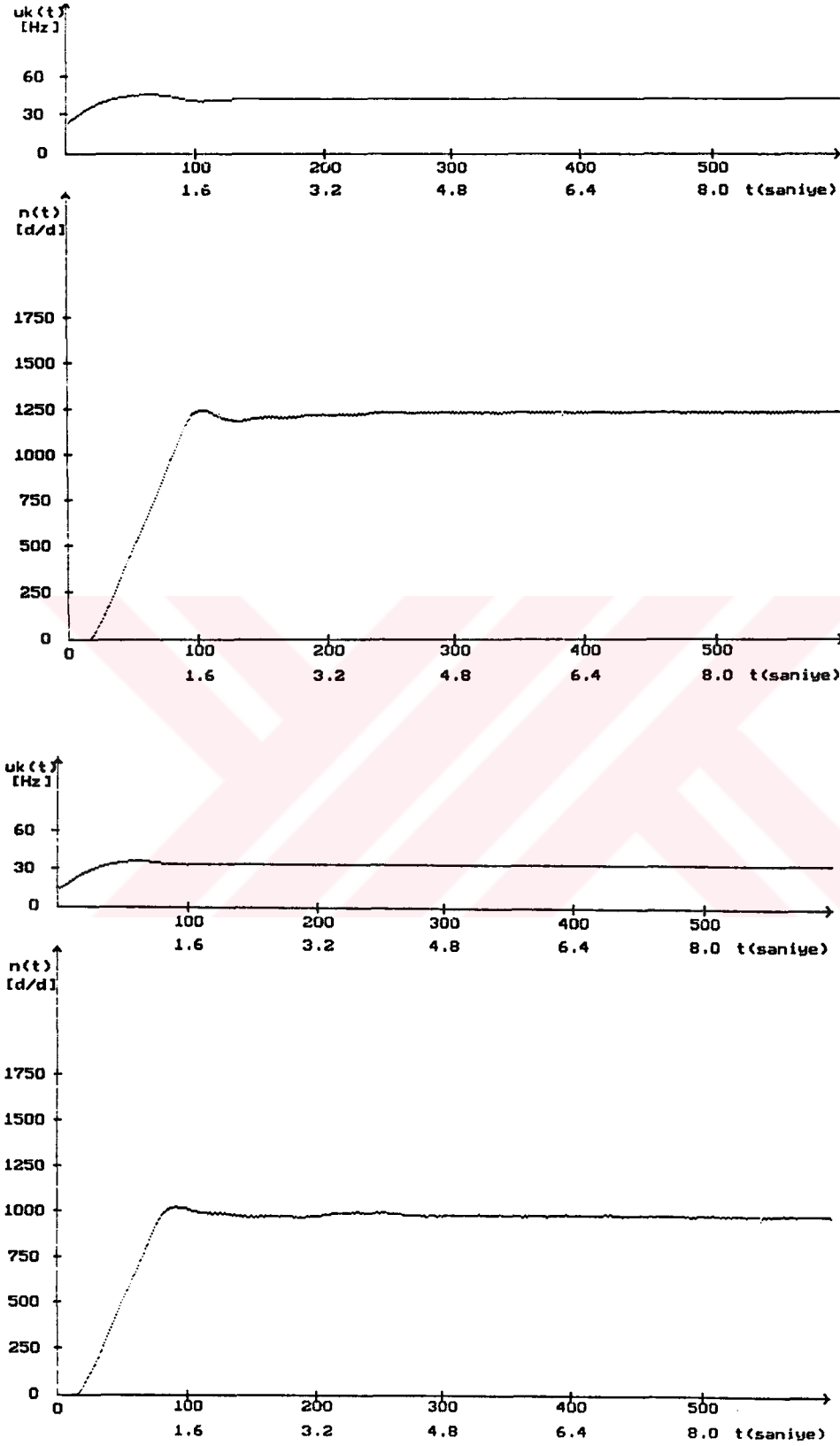
Şekil 10.15. $n_r = 1250$ dev/dak ve $n_r = 1000$ dev/dak'da %25 YÜKLÜ çalışmada, kontrol işareti ve hız grafikleri.

Tablo 10.7. Sistemin %75 ARA YÜKLE çalışmasında enterpolasyon ile bulunan, optimum PID kontrolör parametreleri

İstenen Hız (dev/dak)	K_p	K_I	K_D
1500	1.22	2.3	0.013
1250	1.3405	2.692	0.007
1000	0.959	3.2025	0.00265
750	1.2305	3.6105	0.0114
500	1.236	4.903	0.00796

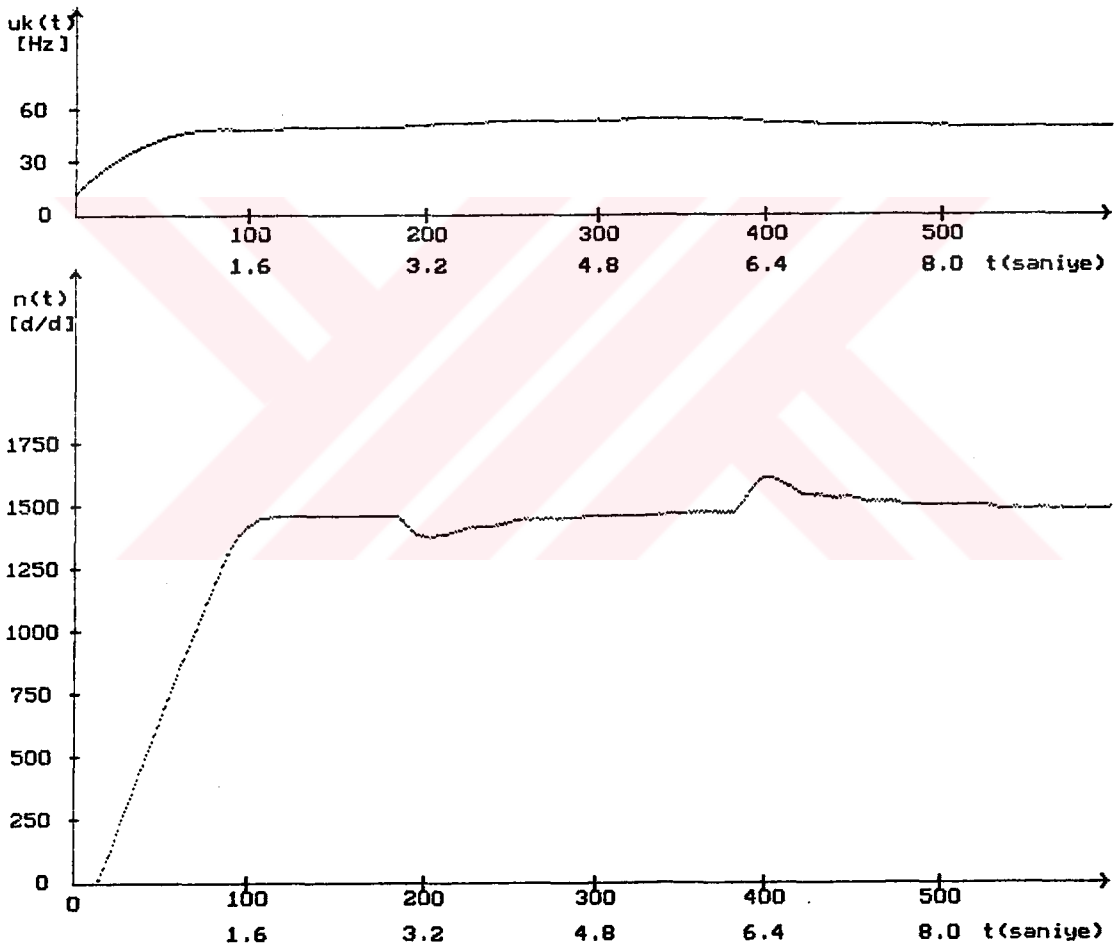


Şekil 10.16. $n_r = 1500$ dev/dak 'da %75 YÜKLÜ çalışma durumunda, kontrol işareti ve hız grafikleri.

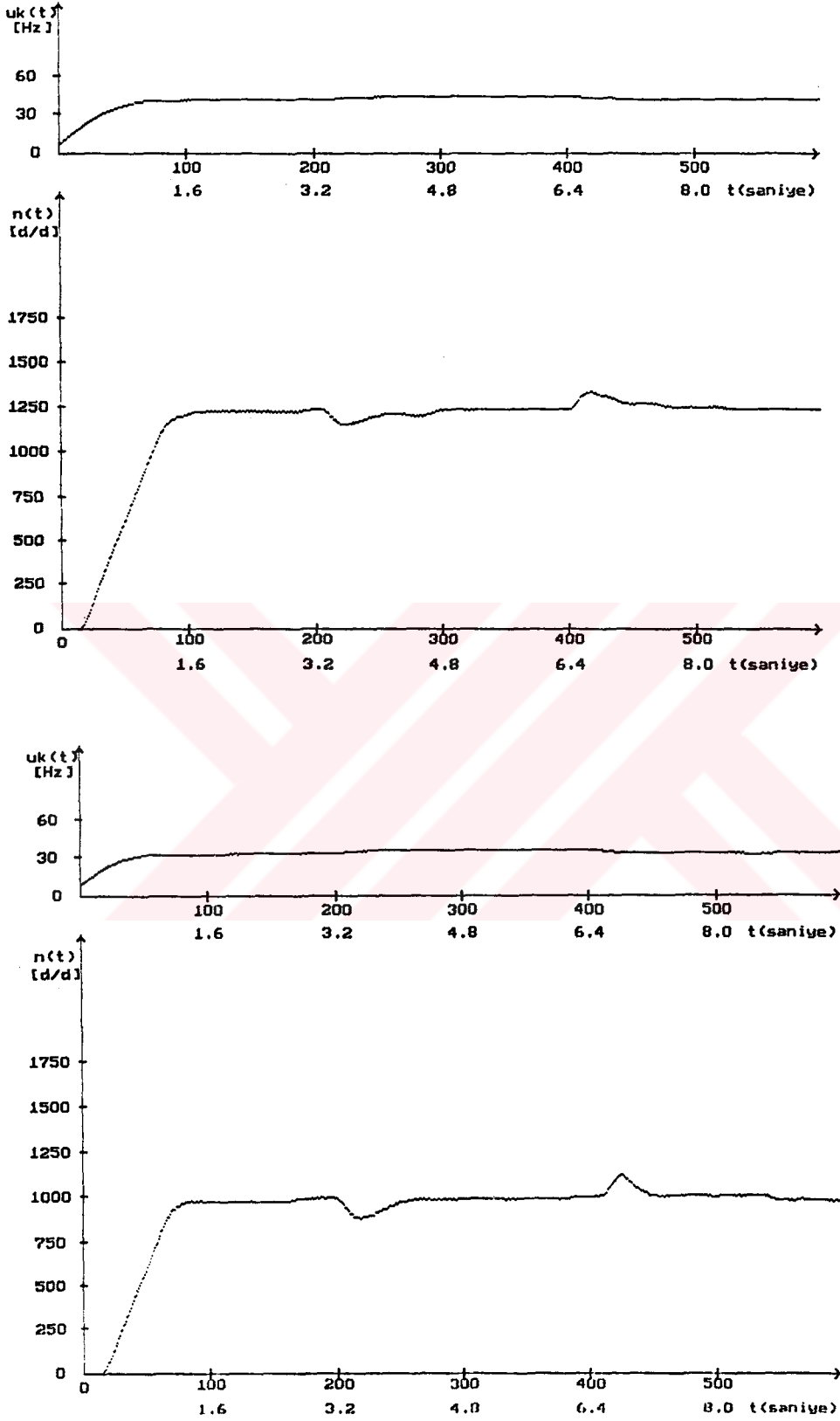


Şekil 10.17. $n_r = 1250$ dev/dak ve $n_r = 1000$ dev/dak'da %75 YÜKLÜ çalışmada kontrol işareti ve hız grafikleri.

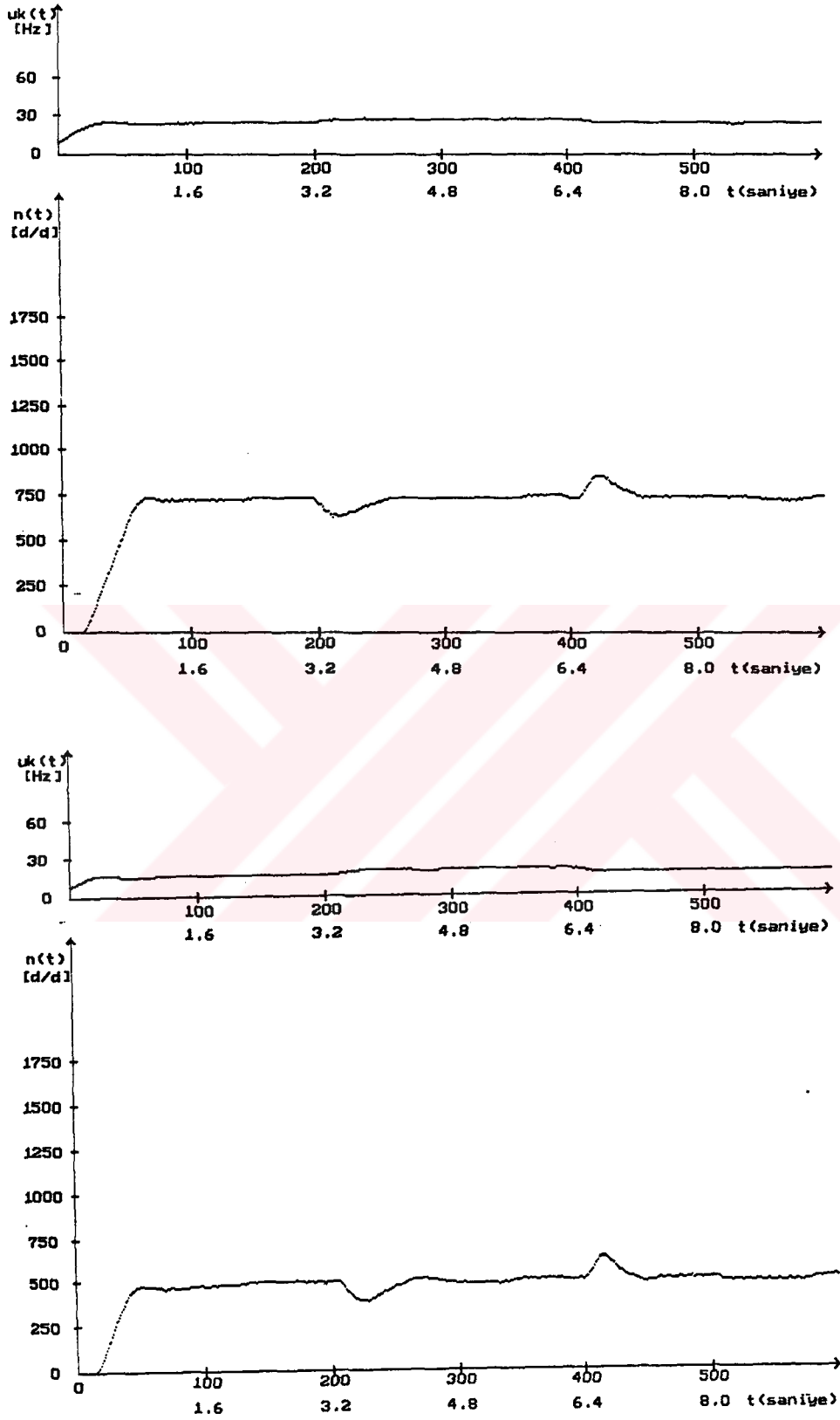
Sistemin yüksüz durumda çalışma noktalarına ilişkin olarak ölçümle bulunan, optimum PID kontrolörü katsayılarında, motorun, geçici rejimini tamamlayıp sürekli hale oturduğunda, Anma Yüğü ile ani olarak yüklenmesi ve sonra bu yükün ani olarak atılması durumlarında sistem yanıtlarını içeren grafikler aşağıda sunulmuştur.



Şekil 10.18. Sistemin, $n_r = 1500$ dev/dak'da ANMA YÜKÜ ile ani olarak yüklenip, ani yük atılması durumlarında, kontrol işareti ve Hız grafikleri.



Şekil 10.19. Sistemin, $n_i = 1250$ dev/dak'da ve $n_i = 1000$ dev/dak'da ANMA YÜKÜ ile ani olarak yüklenip, ani yük atılması durumlarında, kontrol işareti ve hız grafikleri.



Şekil 10.20. Sistemin, $n_r = 750$ dev/dak'da ve $n_r = 500$ dev/dak'da ANMA YÜKÜ ile ani olarak yüklenip, ani yük atılması durumlarında kontrol işareti ve hız grafikleri.

Böylece dijital PID kontrolörü, sürücü ve asenkron motordan oluşan sistem, asenkron motor ile tahrik edilen sistemlere rahatlıkla uygulanabilir. Bu çalışmanın devamı olarak, kontrol sistemine, kişisel bilgisayar yerine bir mikrokontrolör eklenmesi düşünülmektedir. Kişisel bilgisayar ile sistem üzerinde çeşitli çalışma noktaları için optimum PID kontrolörü parametreleri, Yeni Global Optimizasyon programıyla, çevrim dışı (off-line) ölçümler sonucu belirlenip, yukarıdakilerine benzer yapıda tablolar oluşturulabilir. Sonra, bu tablolar bir mikrokontrolörün belleğine yüklenebilir. Sistem üzerinde optimum değerleri belirlemek için yapılan ölçümler bittikten sonra kişisel bilgisayara gereksinim kalmayacak, onun yerini, dijital PID kontrolör algoritması ve optimum PID kontrolör parametreleri değerlerinin yüklendiği mikrokontrolör alabilecektir. Sisteme verilen hız referansı v moment algılayıcıları ile belirlenebilecek yüklenme durumları, mikroişlemcinin belleğindeki tablo değerlerine doğrudan karşılık geliyorsa tablo değerleri, doğrudan kullanılabilir. Referans değerler, tablo değerlerine karşılık gelmiyorsa, bulunduğu aranın sınır değerleri arasında lineer enterpolasyon yapılarak optimum değerler belirlenip, dijital PID kontrolörü parametreleri olarak atanarak, kontrolör algoritmasının içine yerleştirilecektir. Böylece yapılan çalışma ile elde edilen sonuçlar, piyasadaki pratik uygulamalar için rahatlıkla kullanılabilir bir yapıya ulaştırılmış olacaktır.

KAYNAKLAR

- [1] TIĞLI, O., Global Optimizasyon Yöntemi Kullanılan Bir Doğru Akım Motorunun Hız Kontrolünün Gerçeklenmesi, Yüksek Lisans Tezi, İTÜ, 1990
- [2] BOZ, O., A New Global Optimization Technique and Application To System Design, Yüksek Lisans Tezi, İTÜ 1990
- [3] KUSHNER, H., J., A New Method of Location the Maximum of an Arbitrary Multipeak Curve In the Presence of Noise, Proceedings of the Joint Automatic Control Conference ,1963
- [4] STUCKMAN, B., E., Global Optimization Using Brownian Motion for Multivariable Functions, TR-02-87, ESE, 01 Oakland University Press, 1986
- [5] STUCKMAN, B., E., LAURSEN N., W., Modern Control System Design Using Global Search, Proceeding of the 8th Meeting of the Coordinate Group on Modern Control Theory U.S. Army, 1987
- [6] STUCKMAN, B., E., STUCKMAN, P., L., LILLY, J., Finding the Best optimal Control Using Multivariable Stochastic Global Search, Proceeding of the 31st Midwest Symposium on Circuits & Systems, St.Louis Missouri, 9-12 August 1988.

- [7] FRÖHR, F., OTTENBURGER, F., Introduction to
Electronic Control Engineering, Siemens &
Aktiengesellschaft Heyden & Son Ltd.
London, 1982
- [8] KUO, B., Digital Control Systems, SRL Publishing
Company, 1977
- [9] DORF, R., C., Modern Control Systems, Addison Wesley
Publishing Company, 1974
- [10] GÖKAŞAN, M., Sincap Kafesli Asenkron Makinalarda
Modern Kontrol Yöntemlerinin Uygulanması,
Doktora Tezi, İTÜ, 1989
- [11] SARIOĞLU, M., K., Asenkron Makinalar, Çağlayan
Kitabevi, 1983
- [12] SARIOĞLU, M., K., Dynamics of Electrical Machines,
Classnotes for EE 497 MKS, University of
Illinois, Urbana IL 61801
- [13] BOSE, B., K., Power Electronic and AC Drives,
Prentice Hall, 1986
- [14] Zeigler, J., G., Nichols, N., B., Optimum Settings
of Automatic Controllers, Trans. ASME, 64,
759-768 (1942)
- [15] ÅSTRÖM, K., J., Automatic Tuning of PID Controllers,
Instrument Society of America 1988

EK 1

program Murat Ata 1 ;

{ Bu Program, Asenkron Makinanın, Değişken Genlik ve Frekanslı Kaynaktan Beslenmesi Durumu için Simülasyonunu Vermektedir...}

USES CRT,GRAPH;

VAR

FISDB,FISQB,FIRDB,FIRQB	: REAL;
FISDB1,FISQB1,FIRDB1,FIRQB1	: REAL;
FISDB2,FISQB2,FIRDB2,FIRQB2	: REAL;
FISDB3,FISQB3,FIRDB3,FIRQB3	: REAL;
FISDB4,FISQB4,FIRDB4,FIRQB4	: REAL;
ISDB,ISQB,IRDB,IRQB,ISB,IRB	: REAL;
NB,NB1,NB2,NB3,NB4	: REAL;
P1,P2,P3,P4,P5,Q1,Q2,Q3,Q4,Q5	: REAL;
R1,R2,R3,R4,R5,S1,S2,S3,S4,S5	: REAL;
N,NSS,T,VSS,VT,RSB,RRB,XKSB,XKRB,XM,XMB,XSB	: REAL;
XRB,FR,FSB,FRB,IT,HIZ,HIZ1,ZNT,VS,VSD,VSQ	: REAL;
VSDB,VSQB,PP,MYB,TN,MEN,KM,OM1N,OMSB,OMS,OMR	: REAL;
L,NONT,PSNT,FIT,SIGMA,K1,K2,K3,K4,K6,XX,F,FS	: REAL;
TH,MENT,S	: REAL;
I,kk	: INTEGER;
Gd, Gm,YENI,x	: INTEGER;
Color	: WORD;

CONST

```

VN=220;
RS=1.93;
RR=1.45;
XKS=3;
XKR=3.8;
XMM=59;
NON=23.8534;
IIN=6.7;
PN=3000;
J=0.0218;
H=0.0016;
P=2;
XS=62;
XR=62.8;
PI=3.14159;
FT=50;

```

LABEL

basla;

```

{ ***** }
{ ***** EKRANA EKSEN TAKIMINI ÇİZEN ALTPROGRAM ***** }
{ ***** }

```

```
PROCEDURE ciz;
```

```
begin
```

```

  Gd := vga; gm:= vgahi; InitGraph(Gd, Gm, '');
  Color := GetMaxColor;
  setttextjustify(centertext,centertext);
  setttextstyle(smallfont,horizdir,2);

```

```
  outtextxy(320,4,' ... Asenkron Makina Simulasyonu ...');
```

```

  line(40,0,40,150);
  line(40,420,639,420);
  line(40,100,639,100);
  line(40,130,40,420);

```

```

  outtextxy(40,5,'^');      outtextxy(20,8,'Me(t)');
  outtextxy(610,110,'t(ms)'); outtextxy(40,130,'^');
  outtextxy(610,430,'t(ms)'); outtextxy(20,140,'n(t)');
  outtextxy(635,101,'>');   outtextxy(635,421,'>');
  outtextxy(20,152,'[d/d]'); outtextxy(22,100,'0');
  outtextxy(22,420,'0');    outtextxy(40,430,'0');
  outtextxy(15,52,'1');     outtextxy(15,77,'0.5');
  outtextxy(38,50,'-');     outtextxy(38,75,'-');
  outtextxy(15,180,'1500');  outtextxy(15,220,'1250');
  outtextxy(15,260,'1000');  outtextxy(15,300,'750');
  outtextxy(15,340,'500');   outtextxy(15,380,'250');
  outtextxy(40,180,'-');     outtextxy(40,220,'-');
  outtextxy(40,260,'-');     outtextxy(40,300,'-');
  outtextxy(40,340,'-');     outtextxy(40,380,'-');

```

```

  kk:=0;i:=49;
  while kk< 500 do
  begin
    outtextxy(140+kk,110,chr(i)+'00');
    outtextxy(140+kk,430,chr(i)+'00');
    outtextxy(140+kk,421,'|');
    outtextxy(140+kk,101,'|');
    kk:=kk+100;
    i:=i+1;
  end;

```

```

end;
end;
```

```
BEGIN
```

```
basla:
```

```
  CLRSCR;
```

```

WRITELN ( '          YUK MOMENTININ BIRIM DEGERI: ' );
READLN (MYB);
WRITELN ( '          STATOR(Faz-notr) GERILIMININ GERCEK
DEGERI: ' );
READLN (VSS);
WRITELN ( '          STATOR GERILIMI FREKANSININ GERCEK
DEGERI: ' );
READLN (F);

```

```

FS:=F;
NSS:=F/P;
VS:=SQRT(2)*VSS;
VT:=SQRT(2)*VN;
IT:=SQRT(2)*IIN;
ZNT:=VT/IT;

```

```

TN:=1/(2*50*PI);
MEN:=PN/(2*PI*NON);
TH:=(2*PI*J*NSS)/MEN;
KM:=1.32;
OM1N:=2*F*PI;
OMSB:=(2*PI*NSS)/OM1N;
OMS:=2*PI*NSS;
L:=F/FT;
XM:=L*XMM;

```

{Birim Değerlerin Hesabı.....}

```

XKSB:=XKS/ZNT;
XKRB:=XKR/ZNT;
RSB:=RS/ZNT;
RRB:=RR/ZNT;
XMB:=XM/ZNT;
XSB:=(L*XS)/ZNT;
XRB:=(L*XR)/ZNT;
SIGMA:=1-(XMB*XMB/(XRB*XSB));
K1:=1/SIGMA;
K2:=XMB/(SIGMA*XSB*XRB);
K3:=1/TN;
K4:=1/TH;
K6:=1/6;

```

```

NONT:=OM1N/(2*PI*P);
PSNT:=3*VN*IIN;
MENT:=MEN;
FIT:=VN*SQRT(2)/OM1N;
VSDB:=VS/VT;
VSQB:=0;
FSB:=F/FT;

```

```
{ ... Baslangıç Koşulları ... }
```

```
FISDB:=0;FISQB:=0;FIRDB:=0;FIRQB:=0;
FISDB4:=0;FISQB4:=0;FIRDB4:=0;FIRQB4:=0;
ISDB:=0;ISQB:=0;IRDB:=0;IRQB:=0;
NB:=0;N:=0;
```

```
{ ... Makina Denklemlerinin Çözümü ... }
```

```
ciz;
```

```
FOR I:=1 TO 550 DO
BEGIN {1}
```

```
  N:=NB*NONT;
  OMR:=2*PI*N;
  S:=(NSS-N)/NSS;
  FR:=S*FS;
  FRB:=FR/FT;
```

```
{ ... Diferansiyel Denklem Takımının Çözümü ... }
```

```
P1:=H*((K3)*(-RSB*ISDB+FSB*FISQB+VSDB));
P2:=H*((K3)*(-RSB*ISQB-FSB*FISDB+VSQB));
P3:=H*((K3)*(-RRB*IRDB+FRB*FIRQB));
P4:=H*((K3)*(-RRB*IRQB-FRB*FIRDB));
P5:=H*((K4)*(KM*K2*(FISQB*FIRDB-FIRQB*FISDB)
-MYB*NB*NB-0.01*NB));
```

```
FISDB1:=FISDB+P1*0.5;
FISQB1:=FISQB+P2*0.5;
FIRDB1:=FIRDB+P3*0.5;
FIRQB1:=FIRQB+P4*0.5;
NB1:=NB+P5*0.5;
```

```
Q1:=H*((K3)*(-RSB*ISDB+FSB*FISQB1+VSDB));
Q2:=H*((K3)*(-RSB*ISQB-FSB*FISDB1+VSQB));
Q3:=H*((K3)*(-RRB*IRDB+FRB*FIRQB1));
Q4:=H*((K3)*(-RRB*IRQB-FRB*FIRDB1));
Q5:=H*((K4)*(KM*K2*(FISQB1*FIRDB1-FISDB1*FIRQB1)
-MYB*NB*NB-0.01*NB));
```

```
FISDB2:=FISDB+Q1*0.5;
FISQB2:=FISQB+Q2*0.5;
FIRDB2:=FIRDB+Q3*0.5;
FIRQB2:=FIRQB+Q4*0.5;
NB2:=NB+Q5*0.5;
```

```

R1:=H*((K3)*(-RSB*ISDB+FSB*FISQB2+VSDB));
R2:=H*((K3)*(-RSB*ISQB-FSB*FISDB2+VSQB));
R3:=H*((K3)*(-RRB*IRDB+FRB*FIRQB2));
R4:=H*((K3)*(-RRB*IRQB-FRB*FIRDB2));
R5:=H*((K4)*(KM*K2*(FISQB2*FIRDB2-FISDB2*FIRQB2)
-MYB*NB*NB-0.01*NB));

```

```

FISDB3:=FISDB+R1;
FISQB3:=FISQB+R2;
FIRDB3:=FIRDB+R3;
FIRQB3:=FIRQB+R4;
NB4:=NB+R5;

```

```

S1:=H*((K3)*(-RSB*ISDB+FSB*FISQB3+VSDB));
S2:=H*((K3)*(-RSB*ISQB-FSB*FISDB3+VSQB));
S3:=H*((K3)*(-RRB*IRDB+FRB*FIRQB3));
S4:=H*((K3)*(-RRB*IRQB-FRB*FIRDB3));
S5:=H*((K4)*(KM*K2*(FISQB3*FIRDB3-FISDB3*FIRQB3)
-MYB*NB*NB-0.01*NB));

```

```

{ ... Akıların Hesabı ... }

```

```

FISDB4:=FISDB4+(K6)*(P1+Q1+Q1+R1+R1+S1);
FISQB4:=FISQB4+(K6)*(P2+Q2+Q2+R2+R2+S2);
FIRDB4:=FIRDB4+(K6)*(P3+Q3+Q3+R3+R3+S3);
FIRQB4:=FIRQB4+(K6)*(P4+Q4+Q4+R4+R4+S4);
NB4:=NB4+(K6)*(P5+Q5+Q5+R5+R5+S5);

```

```

IF NB4 <= 0 THEN NB4:=0;

```

```

XX:=KM*K2*(FISQB4*FIRDB4-FISDB4*FIRQB4);

```

```

FISDB:=FISDB4;
FISQB:=FISQB4;
FIRDB:=FIRDB4;
FIRQB:=FIRQB4;
NB:=NB4;

```

```

{ ... Akımların Hesabı ... }

```

```

ISDB:=(K1*(1/XSB))*FISDB-(K2)*FIRDB;
ISQB:=(K1*(1/XSB))*FISQB-(K2)*FIRQB;
IRDB:=(K1*(1/XRB))*FIRDB-(K2)*FISDB;
IRQB:=(K1*(1/XRB))*FIRQB-(K2)*FISQB;
ISB:=SQRT(ISDB*ISDB+ISQB*ISQB);
IRB:=SQRT(IRDB*IRDB+IRQB*IRQB);

```

```

x:=round(2*i);
putpixel (40+x,420-ROUND(nb*240*L),color);
PUTPIXEL (40+x,100-ROUND(XX*40),COLOR);

```

```

END; {1}

      HIZ:=N*60;
      HIZ1:=NSS*60;
{YENI:=READKEY;}READLN;
CLOSEGRAPH;
CLRSCR;

WRITELN ('    VS=',VSS:9:1,' (V)');
WRITELN (' ');
WRITELN ('    FS=',F:5:1,' (Hz)');
WRITELN (' ');
WRITELN ('    MYB=',MYB:6:2);
WRITELN (' ');
WRITELN (' ');
WRITELN ('    S=',S:9:4,' ');
WRITELN ('    NS=',HIZ1:9:4,' (d/d)');
WRITELN ('    N=',HIZ:9:4,' (d/d)');
WRITELN (' ');
WRITELN (' ');
WRITELN ('    ISB=',ISB:9:6);
WRITELN (' ');
WRITELN ('    IRB=',IRB:9:6);
WRITELN (' ');
WRITELN ('    Me(t)=' ,XX:9:6);

READLN;
goto basla;

END.

```

EK 2

```
program Murat Ata 2 ;
```

```
{ Bu program sistem performansını ve objective function'i
hesaplar...}
{ Örnekleme Zamanının belirlenmesinde INTERRUPT Vektörleri
kullanılmıştır...}
```

```
USES crt,dos,graph;
```

```
CONST
```

```
    T_ornek = 16; { Örnekleme Zamanı "milisaniye" }
```

```
    tson = 600;
```

```
    rf_gercek =1000; { ... Gerçek Hız referansı ... }
```

```
VAR
```

```
    cik_say,ikinci_cik_say,in_say      : INTEGER;
    in_sa,t,ci,rfi,uk1i,kk,i,s{tson}  : INTEGER;
    hb,lb,uk1i,chno,xzx,yuk_zam       : INTEGER;
    dl,dh,fff,fs,a,xz,sure,k1         : INTEGER;
    oku_a1,yeni_oku_a1,oku_71         : INTEGER;
    oku_21,yeni_oku_21                : INTEGER;
    Gd, Gm                             : INTEGER;
```

```
    gb1,gb2,gb3,gb4,gb,gbg,gba       : REAL;
    kp,ki,kd,e1,e2,e3,uk1,uk2,tt      : REAL;
    rf,C,K,ksi,Wn,f,ymm,ww            : REAL;
    max_min_fark,a_sak,a_sakla,u_sak  : REAL;
    u_sakla, uk1_ust,usakla,asakla     : REAL;
```

```
    sakla : ARRAY[1..600,1..2] of REAL;
```

```
    say : ARRAY[1..600] of INTEGER;
```

```
    saat,dakika,saniye,salise         : WORD;
    Color                              : WORD;
    time_baslangic,time_yeni           : WORD;
```

```
    bayrak1,bayrak2,ikinci_cik        : BOOLEAN;
```

```
LABEL
```

```
    og1;
```

```

{ ***** }
{ ***** PID KONTROLÖR ALTPROGRAMI ***** }
{ ***** }

```

```

PROCEDURE PID (var kp,ki,kd,e1,e2,e3
               ,uk1,uk2,uk1_ust : real);

```

```

CONST

```

```

    Ualt      = 0;
    Uust      = 10000;

```

```

VAR

```

```

    p1,p2,p3  : REAL;

```

```

begin

```

```

    p1:=(2*kd+2*kp*tt+ki*tt*tt)/(2*tt);

```

```

    p2:=- (4*kd+2*kp*tt-ki*tt*tt)/(2*tt);

```

```

    p3:=kd/tt;

```

```

    { ...Kontrolör işareti ... }

```

```

    uk1:=p1*e1+p2*e2+p3*e3+uk2;

```

```

    e3:=e2; e2:=e1; uk2:=uk1;

```

```

    if uk1 < Uust then

```

```

        begin

```

```

            xzx:=1;

```

```

        end;

```

```

    if uk1 >= Uust then

```

```

        begin

```

```

            uk1:=Uust;

```

```

            if xzx = 1 then

```

```

                begin

```

```

                    uk1_ust := uk1_ust + 1;

```

```

                    xzx := 0;

```

```

                end;

```

```

            end;

```

```

            if uk1<=Ualt then uk1:=Ualt;

```

```

end;

```

```

{ ***** }
{ ***** EKRANA EKSEN TAKIMINI ÇİZEN ALTPROGRAM ***** }
{ ***** }

PROCEDURE  ciz;

VAR

san,hhb,llb : INTEGER;

begin

    Gd := vga;
    gm := vgahi;
    InitGraph(Gd, Gm, '');
    Color := GetMaxColor;
    setttextjustify(centertext,centertext);
    setttextstyle(smallfont,horizdir,2);

    outtextxy(320,4,' ... Asenkron Makina PID Kontrolu ve
    Performans Hesabi...');

    line(40,0,40,100);
    line(40,420,639,420);
    line(40,100,639,100);
    line(40,125,40,420);

    outtextxy(40,5,'^');    outtextxy(20,8,'uk(t)');
    outtextxy(600,125,'t(saniye)');    outtextxy(40,130,'^');
    outtextxy(600,445,'t(saniye)');    outtextxy(20,140,'n(t)');
    outtextxy(635,101,'>');    outtextxy(635,421,'>');
    outtextxy(20,152,'[d/d]');    outtextxy(20,18,'[Hz]');
    outtextxy(22,100,'0');    outtextxy(22,420,'0');
    outtextxy(40,430,'0');    outtextxy(15,50,'60');
    outtextxy(15,75,'30');    outtextxy(38,50,'-');
    outtextxy(38,75,'-');    outtextxy(15,210,'1750');
    outtextxy(40,210,'-');    outtextxy(15,240,'1500');
    outtextxy(15,270,'1250');    outtextxy(15,300,'1000');
    outtextxy(15,330,'750');    outtextxy(15,360,'500');
    outtextxy(15,390,'250');    outtextxy(40,240,'-');
    outtextxy(40,270,'-');    outtextxy(40,300,'-');
    outtextxy(40,330,'-');    outtextxy(40,360,'-');
    outtextxy(40,390,'-');

    kk:=0;i:=49;
    while kk< 500 do
    begin

        san := T_ornek * ((kk+100) div 100);
        hhb := san div 10 ;
        llb := san mod 10 ;

```

```

outtextxy(140+kk,125,chr(48+hhb)+chr(46)+chr(48+llb));
outtextxy(140+kk,445,chr(48+hhb)+chr(46)+chr(48+llb));

```

```

        outtextxy(140+kk,110,chr(i)+'00');
        outtextxy(140+kk,428,chr(i)+'00');
        outtextxy(140+kk,421,'|');
        outtextxy(140+kk,101,'|');
        kk:=kk+100;
        i:=i+1;
    end;
end;

```

```

INTERRUPT;

```

```

begin

```

```

    Bayrak2:=true;

```

```

    port[$70] := $C;
    oku_71 := port[$71];
    port[$A0] := $20;
    port[$20] := $20;

```

```

end;

```

```

{ ***** }
{ ***** ANA PROGRAM ***** }
{ ***** }

```

```

VAR

```

```

    c1,c2,c3,c4,ii,cx,izle,ft      : INTEGER;
    gb_sak                        : REAL;
    egim,uc,ucc,carp              : REAL;
    mvy,po,ess,obj_func          : REAL;
    gunce                        : TEXT;

```

```

LABEL

```

```

    son,devam,basla;

```

```

begin

```

```

    assign(gunce,'gunce');

```

```
PORT[$70] := $A;
PORT[$71] := $2A; { ... Örneklem Zamanını değiştirmek için
                    2F' e kadar değiştirilebilir... }
```

```
OKU_A1 := PORT[$A1];
YENI_OKU_A1 := OKU_A1 AND $FE;
PORT[$A1] := YENI_OKU_A1;
OKU_21 := PORT[$21];
YENI_OKU_21 := OKU_21 AND $FB;
PORT[$21] := YENI_OKU_21;
SETINTVEC($70,@ZAMANLAYICI);
PORT[$70] := $B;
PORT[$71] := $43;
PORT[$70] := $C;
OKU_71 := PORT[$71];
```

```
basla:
```

```
    c1 := 10;    { ... CEZA KATSAYILARI ... }
    c2 := 30;
    c3 := 50;
    c4 := 100;
```

```
    rf:=(rf_gercek*4095)/1765;
```

```
    c:=0;
```

```
    clrscr;
```

```
    tt := T_ornek / 1000;
```

```
ogl:
```

```
    time_yeni:=0;time_baslangic:=0;
    { bayrak1:=false;}
    a:=1; clrscr; yuk_zam:=0; mvy:=0; t:=0;
    s:=0; gb:=0;
```

```
{ ... Çıkış Portlarının sıfırlanması ... }
```

```
    port[$304]:=0;port[$305]:=0;
```

```
{ delay(1250);}
```

```
    port[$302]:=$00;    { ADC'nin Koşullandırılması }
```

```
clrscr;
```

```
{ .. PID Kontrolör Parametrelerinin Ekrandan Girilmesi..}
```

```
writeln('          ... Lütfen  kp,ki & kd degerlerini
giriniz ...');
    writeln('');
    writeln(' kp=?');
    readln(kp);
```

```

        writeln('');
        writeln('  ki=?');
        readln(ki);

        writeln('');
        writeln('  kd=?');
        readln(kd);

ciz;

    { ... Başlangıç Koşulları ... }

uk1:=0; e1:=0; e2:=0; e3:=0 ;
uk2:=0; gb:=0; c:=0; f:=0;
t:=0; yuk_zam:=0; ft:=0;

fs:=0;  gb_sak:=0.0;  u_sakla:=0.0;  a_sakla:=0.0;
usakla:=10000.0;  asakla:=10000.0;  a_sak  :=0.0;
u_sak:=0.0;  cik_say  := 0;  ikinci_cik  := false;
ikinci_cik_say := 1; max_min_fark :=0; gb:=0; gbb :=0;
uk1_ust:=0;

rewrite(gunce);

        settime(0,0,0,0);

REPEAT      { Döngünün Başlagıç noktası }

        dl:=0;
        dh:=0;

        port[$300]:=0;  { ... ADC'nin Çevrime Başlaması ...}

repeat      { ... ADC'nin çevrimi için bekleme ... }
        izle := port[$308];
        izle := izle AND $80;
        izle := izle shr 7;
        ft:=ft+1;
until ( izle = 0 );

{ ADC'nin Düşük Anlamalı Bitlerini Okunması}

        dl:=port[$300];

{ ADC'nin Yüksek Anlamalı Bitlerini Okunması}

        dh:=port[$301];

fff:=dl and $f0;
fff:=fff shr 4;

        gb:=dh*16+fff; { ... Geri Besleme işareti ... }

{gb:=dh*16;}
gbb:=gb;

```

```
{ ... hız değişiminin son min ve max değerlerinin bulunması...}
```

```
if gb > 1 then
  begin
```

```
    if gb > gb_sak then
      begin
```

```
        a_sakla := a_sak;
        u_sak   := gb;
        gb_sak  := gb;
        {cik_say :=0;}
        in_sa   :=0;
```

```
        if gb > (1.001*a_sakla) then
          begin
```

```
            if ikinci_cik then
              begin
```

```
                if asakla > a_sakla then
                  begin
                    asakla:=a_sakla;
```

```
                    if usakla > u_sakla then
                      begin
                        usakla := u_sakla;
                      end;
```

```
{outtextxy(320,175,' max_min_fark hesaplandi...');}
```

```
            end;
```

```
          end;
```

```
        end;
```

```
        ikinci_cik := False;
        goto devam;
```

```
    end;
```

```
    {cik_say :=1;}
    u_sakla := u_sak;
```

```
    a_sak := gb;
```

```
    gb_sak :=gb;
```

```
    in_sa := in_sa + 1;
```

```
    if in_sa > 3 then
```

```
      begin
```

```
        ikinci_cik := true;
```

```
      end;
```

```
  end;
```

```
devam:
```

```

if cik_say >= 1 then
  begin
    if in_say >= 1 then
      begin
        if u_sakla - a_sakla >= 0.1*rf then
          begin
            s:=s+1;

            if s <= 6 then
              begin
                in_say :=0;
                goto cizim;
              end;
            goto cizim;
          { goto yaz1;}
        end;
      end;
    end;
  end;

if gb < (0.9*rf) then {..rf*0.9 Bandını Oluşturan Yapı..}
  begin
    ii:=0;
    end;

if gb >= (0.9*rf) then
  begin
    if ii<=0 then
      begin
        yuk_zam:=t;
        ii:=ii+1;
      end;
    end;

if mvy <= gb then
  begin
    mvy:=gb;
    end;

if t>=tson-1 then
  begin
    po:=100*(mvy-rf)/rf;

    if po <= 2.0 then { ... ASIM ALT SINIRI...}
      begin
        po:=0;
      end;
    end;

cizim:

```

```

ci :=round(gb/19.5);
rfi :=round(rf/10);
uk1i:=round(uk1/200);
PutPixel(40+t,100-uk1i,Color);
PutPixel(40+t,420-ci,Color);

e1:=rf-gbb;          { Hata işareti }

pid(kp,ki,kd,e1,e2,e3,uk1,uk2,uk1_ust);

sakla[t,1] := uk1;
sakla[t,2] := gbb;

{ ... Açık Çevrim Kontrol ... }

    {uk1i:= round((rf_gercek*4095)/1500);}

{ ... Kapalı Çevrim Kontrol ... }

    uk1i:=round(uk1*4095/10000);

    hb:=uk1i div 16;
    lb:=uk1i mod 16;

{ ... Kontrol işaretinin Portlara Yüklemesi ... }

    port[$304]:=lb;
    port[$305]:=hb;

{ ... Örnekleme Süresinin Tamamlanması için Bekleme ... }

repeat
until (Bayrak2);
bayrak2 := false;

    t:=t+1;

UNTIL (t = tson);    { ... Döngünün Sonu ... }

repeat until keypressed;
closegraph;

{ ... Hız Bilgisi işaretlerinin Dosyaya Yükleneşi ... }
t:=0;

```

```

repeat
write(gunce,sakla[t,1]:10:3);
write(gunce,sakla[t,2]:10:3);
writeln(gunce);
t:=t+1;
until( t = 600);

{ ... Kararlı Hal Hatasının Hesaplanması ... }

    gba := (u_sakla + a_sakla)/2;
    ess:=100*abs(rf-gba)/rf;

    if ess <= 1 then
        begin
            ess := 0;
        end;

    if yuk_zam <= 0 then
        begin
            yuk_zam:=tson;
        end;

{ ... Hız Üzerindeki Salınımın Hesaplanması ... }

    if usakla < 1.1*rf then
        begin
            max_min_fark := (usakla - asakla);
        end;
    ymm := 100*(max_min_fark / rf);

{ ... Performans Index'in hesaplanması... }

    P.I. := c1*po + c2*yuk_zam + c3*ess + c4*ymm ;

gotoxy(10,3);
writeln(' ... SISTEM PERFORMASI ve OBJECTIVE FUNCTION
DEGERI ... ');

gotoxy(15,12);
writeln(' max_min_fark = ',max_min_fark:8:3);
gotoxy(15,14);
writeln(' ymm = ',ymm:8:3,' tt= ',tt:8:4);
gotoxy(10,10);
writeln('po = ',po:8:3,' yuk_zam = ',yuk_zam:8,' ess
= ',ess:8:3);
gotoxy(10,16);
writeln(' PERFORMANCE INDEX = ',P.I.:10:4,'
fsay=',ft);

```

```
gotoxy(10,7);
writeln('kp  =',kp:8:3,'          ki  =  ',ki:8:3,'          kd
        =',kd:8:3);

gotoxy(10,24);
writeln('... DEVAM  ETMEK  ICIN  ENTER  TUSUNA  BASINIZ
...');

readln;
clrscr;

goto og1;

end.
```



```
PROGRAM AS_PID_OPT(input, output);
```

```
{ Bu program, dijital PID kontrolörü parametrelerinin  
optimum değerlerini, öngörülen sınırlar içinde, Yeni Global  
Arama Yöntemi ile belirler.}
```

```
USES
```

```
  crt,dos,GRAPH;
```

```
CONST
```

```
  rf_real          = 1500;  
                    { ...reference value must be in RPM ... }  
  
  t_end            = 600;  
  
  t_sample         = 16;      { " Milisecond " }  
  
  stopping_factor  = 1;  
  
  max_deg_func     = 10;  
  deg_func         = 3;  
  num_sub_tbl      = 2;  
  num_pnt          = 10;  
  num_pnt_sub_tbl  = num_pnt DIV num_sub_tbl;  
  num_crs_pnt      = 30;  
  bnd_part         = 5;  
  pi               = 3.141592653589793;  
  n                = 3;  
  m                = 1;  
  h                = 0.0005;  
  max_num_samp     = 4000;  
  pt               = $380;
```

```
TYPE
```

```
  points           = RECORD  
    X      : ARRAY[1..max_deg_func] OF REAL;  
    func   : REAL;  
    inuse  : BOOLEAN;  
  END;  
  
  point           = ARRAY [1..num_pnt] OF points;  
  boundary_values = ARRAY [1..max_deg_func] OF  
    RECORD  
      Xl, Xf : REAL;  
    END;  
  
  vector1         = ARRAY[1..n] OF REAL;
```

VAR

```

global_maximum      : points;
bound_val           : boundary_values;
Xk0                 : vector1;

first               : file of real;
second              : file of real;

same_pnt,flag1,flag2 : BOOLEAN;

glob_local          : INTEGER;
func_eval           : INTEGER;
pnt_rep             : INTEGER;
pnt_eval            : INTEGER;
rand_num            : INTEGER;
i, j, k, kk         : INTEGER;
gb,mvy,t,ii,f,lb,hb,chno,sayi : INTEGER;
uk11,ci,uk11,rfi,tr : INTEGER;
dl,dh,fff,san,hhb,llb : INTEGER;
oku_a1,yeni_oku_a1,oku_71 : INTEGER;
oku_21,yeni_oku_21   : INTEGER;
izx,cik_say,in_say   : INTEGER;
ikinci_cik_say,s,ik_cik_say : INTEGER;
Gd, Gm,x             : INTEGER;

deg                 : REAL;
Amax_func           : REAL;
Amin_func           : REAL;
C1, C2, C3, C4      : REAL;
USAY,SAKLA,ALTSAY,ALTSAKLA : REAL;
GBB,SAY,gb_sak,a_sakla,u_sakla : REAL;
a_sak,u_sak,asakla,usakla : REAL;
max_min_fark,ymm     : REAL;
rf,c,cy,y,uk1,e1,e2,e3 : REAL;
uk2,p1,p2,p3,tt      : REAL;
dummy_rand          : REAL;

Color,hh,mm,s1,s2    : WORD;

```

```

{ ***** }

{ ***** PROCEDURE PID ***** }

{ ***** }

```

```
PROCEDURE PID(var kp,ki,kd,e1,e2,e3,uk1,uk2:real);
```

```
CONST
```

```

    U_up      = 0;
    Uust      = 10000;

```

```
VAR
```

```

    p1,p2,p3 : REAL;

```

```
begin
```

```
  p1:=(2*kd+2*kp*tt+ki*tt*tt)/(2*tt);
  p2:=- (4*kd+2*kp*tt-ki*tt*tt)/(2*tt);
  p3:=kd/tt;
```

```
    uk1:=p1*e1+p2*e2+p3*e3+uk2;
    e3:=e2;e2:=e1;uk2:=uk1;
```

```
    if uk1<=U_up then uk1:=U_up;
    if uk1>=Uust then uk1:=Uust;
```

```
    if uk1<= 0. then
      begin
        uk1:=0.;
      end;
    if uk1>=10000 then
      begin
        uk1:=10000;
      end;
```

```
end;
```

```
{ ***** }
```

```
{ ***** PROCEDURE ON SCREEN DRAWING ***** }
```

```
{ ***** }
```

```
PROCEDURE DRAW ;
```

```
VAR
```

```
san,hhb,llb : integer;
```

```
begin
```

```
  Gd := vga;
  gm := vgahi;
  InitGraph(Gd, Gm, '');
  Color := GetMaxColor;
  setttextjustify(centertext,centertext);
  setttextstyle(smallfont,horizdir,2);
```

```
  outtextxy(320,4,' ... UNLOADED working of Induction Machine
  ...');
```

```
    line(40,0,40,100);
    line(40,420,639,420);
    line(40,100,639,100);
    line(40,125,40,420);
```

```
  outtextxy(40,5,'^'); outtextxy(20,8,'uk(t)');
  outtextxy(600,125,'t(second)'); outtextxy(40,130,'^');
```

```

outtextxy(600,445,'t(second)');outtextxy(20,140,'n(t)');
outtextxy(635,101,'>');outtextxy(635,421,'>');
outtextxy(20,152,'[d/d]'); outtextxy(20,18,'[Hz]');
outtextxy(22,100,'0');    outtextxy(22,420,'0');
outtextxy(40,430,'0');    outtextxy(15,50,'60');
outtextxy(15,75,'30');    outtextxy(38,50,'-');
outtextxy(38,75,'-');    outtextxy(15,210,'1750');
outtextxy(40,210,'-');    outtextxy(15,240,'1500');
outtextxy(15,270,'1250'); outtextxy(15,300,'1000');
outtextxy(15,330,'750');  outtextxy(15,360,'500');
outtextxy(15,390,'250');  outtextxy(40,240,'-');
outtextxy(40,270,'-');    outtextxy(40,300,'-');
outtextxy(40,330,'-');    outtextxy(40,360,'-');
outtextxy(40,390,'-');

```

```

kk:=0;i:=49;
while kk< 500 do
begin

```

```

    san := t_sample * ((kk+100) div 100);
    hhb := san div 10 ;
    llb := san mod 10 ;

```

```

outtextxy(140+kk,125,chr(48+hhb)+chr(46)+chr(48+llb));
outtextxy(140+kk,445,chr(48+hhb)+chr(46)+chr(48+llb));

```

```

    outtextxy(140+kk,110,chr(i)+'00');
    outtextxy(140+kk,428,chr(i)+'00');
    outtextxy(140+kk,421,'|');
    outtextxy(140+kk,101,'|');
    kk:=kk+100;
    i:=i+1;

```

```

end;
end;

```

```

{ ***** }
{ ***** PROCEDURE FIND_PERFORMANCE ***** }
{ ***** }

```

```

PROCEDURE FIND_PERFORMANCE ( var

```

```

    KP, KI, KD,PO, tr, ESS,ymm : REAL;
    s1,s2 : word);

```

```

VAR

```

```

izle,in_sa : integer;
ikinci_cik : boolean;

```

```

LABEL
    final, drawing, writel, go_on, jump_on;

Begin

uk1:=0,e1:=0,e2:=0,ii:=0      ;
e3:=0,uk2:=0,t:=0,cy:=0,tr:=0;

chno:=0,mvy:=0;

flag1:=true;
flag2:=true;
gb_sak:=0., u_sakla:=0.0, a_sakla:=0.0, a_sak :=0.0;
u_sak:=0.0; cik_say := 0; ik_cik_say:=1,gb:=0, gbb :=0;
s:=0, izx:=0, ii:=0;

    port[$302]:=$00; { Conditng of ADC }

port[$304]:=0; port[$305]:=0; { Reset of output Ports }

delay(4500);

DRAW;

hh:=0; mm:=0; s1:=0; s2:=0,i:=0;

    { ----- Start of Loop ----- }

ikinci_cik := false;
ikinci_cik_say := 1;
max_min_fark :=0.0;
asakla:=10000;
usakla:=10000;
in_sa := 0;
cik_say := 0;

settime(0,0,0,0);

REPEAT

    dl:=0;
    dh:=0;

    port[$300]:=0;    { Start of conversion of ADC }

repeat
    trace := port[$308];
    trace := trace AND $80;
    trace := trace shr 7;
until ( trace = 0 );

```

```

dl:=port[$300]; { reading of ADC's low byte's latch }
dh:=port[$301]; { reading of ADC's High byte's latch }

fff:=dl and $f0;
fff:=fff shr 4;

gb:=dh*16+fff; { Feedback signal with low bytes}
{ gb:=dh*16;} { Feedback signal without low bytes}

gbb:=gb;

e1:=rf-gb; { ... Error signal...}

pid(kp,ki,kd,e1,e2,e3,uk1,uk2);

{ Finding the maximum value of feedback signal for percent
overshoot }

    if mvy <= gb then
    begin
        mvy:=gb;
    end;

    if gb < (0.9*rf) then
    begin
        ii:=0;
    end;

    if gb >= (0.9*rf) then
    begin
        if ii = 0 then
        begin
            tr:=t;
            ii:=ii+1;
        end;
    end;

    if gb > 10 then
    begin

        if gb > gb_sak then
        begin
            a_sakla := a_sak;
            u_sak   := gb;
            gb_sak  := gb;
            cik_say := 0;
            in_sa   := 0;
            in_say  := 1;

```

```

if gb > (1.001*a_sakla) then
begin
  if ikinci_cik then
  begin
    if asakla > a_sakla then
    begin
      asakla:=a_sakla;
      if usakla > u_sakla then
      begin
        usakla := u_sakla;
      end;
    end;
  end;
end;
end;

ikinci_cik := False;
goto go_on;

end;
  cik_say := 1;
  u_sakla := u_sak;
  a_sak := gb;
  gb_sak :=gb;
  in_sa := in_sa + 1;
  if in_sa > 2 then
  begin
    ikinci_cik := true;
  end;

  end;

go_on:

  if cik_say >= 1 then
  begin
    if in_say >= 1 then
    begin
      if u_sakla - a_sakla >= 0.15*rf then
      begin
        s:=s+1;
        if s <= 6 then
        begin
          in_say :=0;
          goto drawing;
        end;
        goto writel;
      end;
    end;
  end;
  end;

  end;

drawing:
  ci :=round(gb/19.5);
  rfi :=round(rf/20);
  uk1i:=round(uk1/200);

```

```

PutPixel(40+t,100-uk1i,Color);
PutPixel(40+t,420-ci,Color);

uk11:=round(4095*uk1/10000); { ... Control Signal ... }

hb:=uk11 div 16;
lb:=uk11 mod 16;

port[$304]:=lb; {Start of DAC for Control Signal output}
port[$305]:=hb;

repeat { ... Waiting for another Sampling ... }
until(flag2);
flag2:=False;

{ ... Calculation of Percent Overshoot ... }

if t>=t_end-1 then
begin
po:=100*(mvy-rf)/rf;
if po<= 2 then
begin
po:=0;
end;
end;

t:=t+1;
if t>=t_end then
begin
goto son;
end;

UNTIL (t = t_end);

WRITE1
outtextxy(320,150,' Finding - kp - ki - kd - values making
the system unstable ...');

izx:=1;
delay(2000);

final:    ess:=100*abs(rf-gb)/rf;

if ess <= 1.0 then
begin
ess:=0.0;
end;

if tr = 0 then
begin
tr:=t_end;
end;

```

```

        sayi:=sayi+1;
ii:=0;

    if usakla <= (1.1*rf) then
        begin
            max_min_fark := (usakla - asakla);
            outtextxy(320,175,' max_min_fark hesaplandi...');
        end;
        ymm := 100*(max_min_fark / rf);

CLOSEGRAPH;

Clrscr;

end;

INTERRUPT;

begin
    flag2:=true;

    PORT[$70] := $C;
    READ_71 := PORT[$71];
    PORT[$A0] := $20;
    PORT[$20] := $20;

end;

{ ***** }
{ ***** FUNCTION INT_RAND ***** }
{ ***** }

FUNCTION int_rand      ( seed : INTEGER):INTEGER;

BEGIN
    int_rand := round(seed*random(10000)/10000);
END;

{ ***** }
{ ***** FUNCTION REAL_RAND ***** }
{ ***** }

FUNCTION real_rand      ( seed : REAL):REAL;

BEGIN
    real_rand := seed*random(10000)/10000;
END;

```

```

{ ***** }
{ ***** PROCEDURE WRITE_RESULTS ***** }
{ ***** }

```

```

PROCEDURE WRITE_RESULTS( A : point,
                        Amax_func, Amin_func : REAL );
VAR
    i : integer;
    r:real;
BEGIN
    assign(second,'second');
    assign(first,'first');
    REWRITE (first);
    r:=say1;
    write(first,r);
    FOR i:= 1 to num_pnt DO
        BEGIN
            gotoxy(10,i+8);
            write ('A['',i:2,'']=( ', A[i].X[1]:10:5,' ', ' ',
                A[i].X[2]:10:5,' ', ' ',
                A[i].X[3]:10:5,' ') -> ',
                A[i].func:10:5 );
            WRITE(first, A[i].X[1]);
            WRITE(first, A[i].X[2]);
            WRITE(first, A[i].X[3]);
            WRITE(first, A[i].FUNC);
            IF A[i].func = Amax_func
            then
                write('**> max ')
            else
                IF A[i].func = Amin_func
                then
                    write('**> min ');
            writeln;
        END;
        close(first);
        REWRITE(second);
        RESET(first);
        read(first,r);
        write(second,r);
        FOR i:= 1 to num_pnt DO
            BEGIN
                READ(first,DEG);
                WRITE(second,DEG);
                READ(first,DEG);
                WRITE(second,DEG);
                READ(first,DEG);
                WRITE(second,DEG);
                READ(first,DEG);
                WRITE(second,DEG);
            END;
            close(first);
            close(second);
        END;
    END;

```

```

{ ***** }
{ ***** FUNCTION FUNC_VAL ***** }
{ ***** }

FUNCTION func_val( Vars : points):REAL;

VAR
  PO, tr, ess : REAL;

label
  show;

BEGIN

  find_performance(vars.X[1],vars.X[2],
vars.X[3],PO,tr,ess,ymm,s1,s2);

    if izx = 1 then
      begin
        func_val:=-20000.;
        vars.func:=-20000.;
        goto show;
      end;

  func_val := -(C1*PO + C2*tr + C3*ess + C4*ymm);
  func_eval := func_eval + 1;
  gotoxy(10, 3);
  vars.func := -(C1*PO + C2*tr + C3*ess + C4*ymm);
  clrscr;

SHOW:

  writeln('new point = ( ',
                                vars.X[1]:10:5,' , ',
                                vars.X[2]:10:5,' , ',
                                vars.X[3]:10:5,' ) -> ',
                                vars.func:10:5 );

  gotoxy(5,3);
  writeln('Number of tring = ',sayi);
  gotoxy(30,3);
  writeln(' % max_min_fark=',ymm:10:5);

  gotoxy(5,5);
  writeln('p.o. = ', po:10:5, ' tr = ', tr:10:5, ' ess
= ', ess:10:5);
  gotoxy(5,6);

END;

```

```

{ ***** }
{ ***** PROCEDURE FIND_MAX ***** }
{ ***** }

```

```

PROCEDURE FIND_MAX ( VAR max_func      : REAL;
                     VAR max_place     : INTEGER;
                     table              : point;
                     tbl_size          : INTEGER);

```

```

VAR

```

```

    i : INTEGER;

```

```

BEGIN

```

```

    max_func := -9.9E+35;

```

```

    FOR i:=1 TO tbl_size DO

```

```

        IF max_func < table[i].func

```

```

        THEN

```

```

            BEGIN

```

```

                max_func := table[i].func;

```

```

                max_place := i

```

```

            END;

```

```

END;

```

```

{ ***** }
{ ***** PROCEDURE FIND_MIN ***** }
{ ***** }

```

```

PROCEDURE FIND_MAX ( VAR min_func      : REAL;
                     VAR min_place     : INTEGER;
                     table              : point;
                     tbl_size          : INTEGER);

```

```

VAR

```

```

    i : INTEGER;

```

```

BEGIN

```

```

    min_func := 9.9E+35;

```

```

    FOR i:=1 TO tbl_size DO

```

```

        IF min_func > table[i].func

```

```

        THEN

```

```

            BEGIN

```

```

                min_func := table[i].func;

```

```

                min_place := i

```

```

            END;

```

```

END;

```

```

{ ***** }
{ ** PROCEDURE EXTEND_LINE_SEGMENTS_TO_BOUNDARIES ** }
{ ***** }

```

```

PROCEDURE EXTEND_LINE_SEGMENTS_TO_BOUNDARIES
  ( VAR X1_temp, X2_temp      : points;
    VAR X1_temp_func,
        X2_temp_func         : REAL;
        X1, X2               : points;
    VAR A                     : point;
    VAR minindex, maxindex    : integer;
        bnd                   : boundary_values);

```

```

VAR
  i,j                      : INTEGER;
  n, n_temp                : REAL;
  point_on_boundary        : BOOLEAN;
  same_pnt                  : BOOLEAN;

```

```

BEGIN

```

```

  n := 7E+35;

```

```

  point_on_boundary := FALSE;

```

```

  FOR i:=1 TO deg_func DO

```

```

    BEGIN

```

```

      FOR j:=1 TO deg_func DO

```

```

        IF (X2.X[j] = bnd[i].X1) OR (X2.X[j] = bnd[i].Xf)
        THEN

```

```

          BEGIN

```

```

            point_on_boundary := TRUE;

```

```

            EXIT;

```

```

          END;

```

```

        IF point_on_boundary

```

```

        THEN

```

```

          EXIT;

```

```

        END;

```

```

        IF NOT(point_on_boundary)

```

```

        THEN

```

```

          BEGIN

```

```

            IF NOT (X1.X[1] = X2.X[1])

```

```

            THEN

```

```

              BEGIN

```

```

                n := (bnd[1].X1 - X1.X[1]) / (X2.X[1] - X1.X[1]);

```

```

                IF (n < 0)

```

```

                THEN

```

```

                  BEGIN

```

```

                    n := (bnd[1].Xf - X1.X[1]) / (X2.X[1] - X1.X[1]);

```

```

                  END;

```

```

                END;

```

```

            FOR i:=2 TO deg_func DO

```

```

BEGIN
  IF NOT (X1.X[i] = X2.X[i])
  THEN
    BEGIN
      n_temp := (bnd[i].X1 - X1.X[i]) /
                (X2.X[i] - X1.X[i]);

      IF n_temp < 0
      THEN
        BEGIN
          n_temp := (bnd[i].Xf - X1.X[i]) /
                    (X2.X[i] - X1.X[i])
        END;
      IF n_temp < n
      THEN
        BEGIN
          n := n_temp
        END;
      END;
    END;
  IF NOT ((ABS(n-1) < 1E-16) OR (ABS(n) < 1-16))
  THEN
    BEGIN
      FOR i:=1 TO deg_func DO
        X2_temp.X[i] := n*X2.X[i] - (n-1)*X1.X[i];

        X2_temp.func := func_val(X2_temp);

        IF X2_temp.func > A[minindex].func
        THEN
          BEGIN
            A[minindex] := X2_temp;
            A[minindex].func := X2_temp.func;
            pnt_rep := pnt_rep + 1;
            IF X2_temp.func > Amax_func
            THEN
              BEGIN
                Amax_func := X2_temp.func;
                maxindex := minindex;
              END;
            find_min(Amin_func, minindex, A, num_pnt)
          END;
          write_results(A, Amax_func, Amin_func);
          pnt_eval := pnt_eval + 1;
        END
      ELSE
        IF (ABS(n-1) < 1E-16)
        THEN
          BEGIN
            FOR i:=1 TO deg_func DO
              X2_temp.X[i] := X2.X[i];

              X2_temp.func := X2.func;
            END
          ELSE

```

```

        BEGIN
            FOR i:=1 TO deg_func DO
                X2_temp.X[i] := X1.X[i];

                X2_temp.func:= X1.func;
            END;

        END
    ELSE
        X2_temp := X2;

n := 7E+35;

point_on_boundary := FALSE;

FOR i:=1 TO deg_func DO
    BEGIN
        FOR j:=1 TO deg_func DO
            IF (X1.X[j] = bnd[i].Xl) OR (X2.X[j] = bnd[i].Xf)
            THEN
                BEGIN
                    point_on_boundary := TRUE;
                    EXIT;
                END;

            IF point_on_boundary
            THEN
                EXIT;
            END;
            IF NOT(point_on_boundary)
            THEN
                BEGIN
                    IF NOT (X1.X[1] = X2.X[1])
                    THEN
                        BEGIN
                            n := (bnd[1].Xl -X2.X[1])/(X1.X[1] - X2.X[1]);

                            IF (n < 0)
                            THEN
                                BEGIN
                                    n := (bnd[1].Xf -X2.X[1])/(X1.X[1] - X2.X[1]);
                                END;
                            END;
                        END;
                    FOR i:=2 TO deg_func DO
                        BEGIN
                            IF NOT (X1.X[i] = X2.X[i])
                            THEN
                                BEGIN
                                    n_temp := (bnd[i].Xl -X2.X[i])/(X1.X[i] - X2.X[i]);
                                    IF n_temp < 0
                                    THEN
                                        BEGIN
                                            n_temp := (bnd[i].Xf -X2.X[i])/(X1.X[i] - X2.X[i])
                                        END;
                                    END;
                                END;
                            END;
                        END;
                    END;
                END;
            END;
        END;
    END;

```

```

        IF n_temp < n
        THEN
            BEGIN
                n := n_temp
            END;
        END;
    END;
    IF NOT ((ABS(n-1)<1E-16) OR (ABS(n)<1-16))
    THEN
        BEGIN
            FOR i:=1 TO deg_func DO
                X1_temp.X[i] := n*X1.X[i] - (n-1)*X2.X[i];

                X1_temp.func:= func_val(X1_temp);

                IF X1_temp.func > A[minindex].func
                THEN
                    BEGIN
                        A[minindex]      := X1_temp;
                        A[minindex].func := X1_temp.func;
                        pnt_rep           := pnt_rep + 1;
                        IF X1_temp.func > Amax_func
                        THEN
                            BEGIN
                                Amax_func := X1_temp.func;
                                maxindex  := minindex;
                            END;
                            find_min(Amin_func, minindex, A, num_pnt)
                        END;
                        write_results(A, Amax_func, Amin_func);

                        pnt_eval := pnt_eval + 1;
                    END
                ELSE
                    IF (ABS(n-1)<1E-16)
                    THEN
                        BEGIN
                            FOR i:=1 TO deg_func DO
                                X1_temp.X[i] := X1.X[i];

                                X1_temp.func:= X1.func;
                            END
                        ELSE
                            BEGIN
                                FOR i:=1 TO deg_func DO
                                    X1_temp.X[i] := X2.X[i];
                                    X1_temp.func:= X2.func;
                                END;
                            END
                        END
                    ELSE
                        X1_temp := X1;
                    END;
                END;
            END;
        END;
    END;

```

```

{ ***** }
{ *****  PROCEDURE FIND_NEW_POINTS ***** }
{ ***** }

```

```

PROCEDURE FIND_NEW_POINT (      Xs, Xt : points;
                               VAR Xnew  : points;
                               lamda    : REAL);

```

```

VAR

```

```

    length : REAL;
    k       : INTEGER;

```

```

BEGIN

```

```

    k       := 1;
    length  := 0;

```

```

    FOR k:=1 TO deg_func DO
        length := SQR(Xt.X[k] - Xs.X[k]) + length;

```

```

    length := SQR(length);

```

```

    FOR k:=1 TO deg_func DO
        Xnew.X[k] := Xs.X[k]+lamda*(Xt.X[k]-Xs.X[k])/length;
        Xnew.func := func_val(Xnew);

```

```

END;

```

```

{ ***** }
{ *****  PROCEDURE FIND_Km ***** }
{ ***** }

```

```

PROCEDURE FIND_Km ( VAR Km : REAL;
                    A   : point );

```

```

VAR

```

```

    A_func_tot      : REAL;
    A_func_mean     : REAL;
    sigma_square    : REAL;
    i                : INTEGER;

```

```

BEGIN

```

```

    A_func_tot := 0;
    FOR i:=1 TO num_pnt DO
        A_func_tot := A_func_tot + A[i].func;
    A_func_mean := A_func_tot/num_pnt;
    sigma_square := 0;
    FOR i:=1 TO num_pnt DO
        sigma_square := sigma_square +
                        SQR(A_func_mean - A[i].func);
    Km := SQR( sigma_square );

```

```

END;

```

```

{ ***** }
{ ***** PROCEDURE CONSTR_TABLE ***** }
{ ***** }

```

```

PROCEDURE CONSTR_TABLE (      bnd      : boundary_values;
                             VAR A      : point);

```

```

VAR

```

```

    k, i, j, s, l : INTEGER;

```

```

BEGIN

```

```

    FOR k:=1 TO num_pnt DO

```

```

        BEGIN

```

```

            FOR i:=1 TO deg_func DO

```

```

                A [ k ] . X [ i ]      : =
bnd[i].Xf+real_rand(bnd[i].Xl-bnd[i].Xf);

```

```

                A[k].func := func_val(A[k]);

```

```

                pnt_eval := pnt_eval + 1;

```

```

            END;

```

```

END;

```

```

{ ***** }
{ ***** PROCEDURE READ_TABLE ***** }
{ ***** }

```

```

PROCEDURE READ_TABLE (var a:point);

```

```

VAR

```

```

f_name      :string;

```

```

f           :FILE OF REAL;

```

```

r           :real;

```

```

i           :integer;

```

```

BEGIN

```

```

    Clrscr;

```

```

    gotoxy(20,10);

```

```

    write('THE NAME OF STARTING TABLE : ');

```

```

    gotoxy(30,12);

```

```

    write('1: first');

```

```

    gotoxy(30,14);

```

```

    write('2: second');

```

```

    gotoxy(27,17);

```

```

    write(' YOUR CHOISE :');

```

```

    read(i);

```

```

    if i=1 then begin

```

```

        assign(first,'first');

```

```

        reset(first);

```

```

        read(first,r);WRITELN(R:10:5);

```

```

        sayi:=round(r);

```

```

        for i:=1 to num_pnt do

```

```

            begin

```

```

        read(first,A[i].X[1]);
        read(first,A[i].X[2]);
        Read(first,A[i].X[3]);
        read(first,A[i].func);
    end;
    close(first);
end
else begin
    assign(second,'second');
    reset(second);
    read(second,r);
    sayi:=round(r);
    for i:=1 to num_pnt do
    begin
        read(second,A[i].X[1]);
        read(second,A[i].X[2]);
        read(second,A[i].X[3]);
        read(second,A[i].func);
    end;
    close(second);
end;

clrscr;

end;

{ ***** }
{ ***  PROCEDURE HYB_PARTIAL_TBL_FIND_GLOBAL_MAX *** }
{ ***** }

PROCEDURE hyb_partial_tbl_find_global_max
(   bnd           : boundary_values;
    VAR global_max : points);

VAR

    i, j, k, l           : INTEGER;
    maxindex, minindex   : INTEGER;
    m,p                  : INTEGER;
    lst_length           : REAL;
    Km                   : REAL;
    lamda,kont           : REAL;
    A                    : point;
    AA                   : point;
    Al, Am               : points;
    Xnew                 : points;

LABEL
    100,101,102,103,104;

BEGIN

```

```

gotoxy(25,12);
write('2: READ THE STARTING TABLE FROM THE DIRECTORY ');
gotoxy(25,10);
write('1: CONSTRUCT THE STARTING TABLE');
gotoxy(25,15);
write(' YOUR CHOISE :');
read(i);
if i=2 then read_table(a)
  else constr_table(bnd, A);

find_max(Amax_func, maxindex, A, num_pnt);
find_min(Amin_func, minindex, A, num_pnt);
find_Km(Km, A);

write_results(A, Amax_func, Amin_func);

kont:=(abs(Amin_func - Amax_func)/ abs(Amax_func))*100;

WHILE ABS(kont) > stopping_factor DO
BEGIN
  FOR i:= 1 to num_pnt do
    AA[i]:=A[i];

  glob_local := glob_local + 1;

  FOR k:=1 TO num_sub_tbl DO
  BEGIN
    FOR i:=1 TO num_pnt_sub_tbl DO
      FOR j:=i+1 TO num_pnt_sub_tbl DO
      BEGIN

        l := (k-1)*num_pnt_sub_tbl + i;
        m := (k-1)*num_pnt_sub_tbl + j;

        same_pnt := TRUE;

        FOR p:=1 TO deg_func DO
          IF NOT (AA[l].X[p]=AA[m].X[p])
          THEN
            BEGIN
              same_pnt := FALSE;
              GOTO 101;
            END;

        101: IF NOT same_pnt
        THEN
          BEGIN
            IF (glob_local MOD 2) = 0
            THEN
              BEGIN
                extend_line_segments_to_boundaries
                  (A1, Am,
                  AA[l].func, AA[m].func,

```

```

AA[l], AA[m],
A, minindex, maxindex,
bound_val);

same_pnt := TRUE;
FOR p:=1 TO deg_func DO
IF NOT (Al.X[p]=Am.X[p])
THEN
BEGIN
same_pnt := FALSE;
GOTO 102;
END;
102: END
ELSE
BEGIN
Al := AA[l];
Am := AA[m];
END;
END;

IF NOT same_pnt
THEN
BEGIN
lst_length := 0;
FOR p:=1 TO deg_func DO
lst_length := SQR(Al.X[p] - Am.X[p])
+ lst_length;
lst_length := SQR(lst_length);

lamda := (Amax_func + Km - Al.func)
*lst_length/(2*Amax_func
+ 2*Km - Al.func
- Am.func);

pnt_eval := pnt_eval + 1;
find_new_point (Al, Am, Xnew, lamda);

IF Xnew.func > Amin_func
THEN
BEGIN

pnt_rep := pnt_rep + 1;
A[minindex] := Xnew;

find_Km(Km, A);

IF Xnew.func > Amax_func
THEN
BEGIN
Amax_func := Xnew.func;
maxindex := minindex;
END;

```

```

        find_min(Amin_func, minindex, A, num_pnt);
    END;
    write_results(A, Amax_func, Amin_func);

    kont:=(abs(Amin_func      -      Amax_func)/
abs(Amax_func))*100;

    IF ((kont) < stopping_factor )
    THEN
        BEGIN
            global_max := A[maxindex];
            GOTO 100;
        END;
    END;
END;
END;
END;
END;
global_max := A[maxindex];
100:
END;

```

```

{ ***** }
{ ***** MAIN PROGRAM ***** }
{ ***** }

```

```

BEGIN

```

```

    RANDOMIZE;

```

```

PORT[$70] := $A;
PORT[$71] := $2A; { This value can be changed to 2F for
another sampling periods

```

```

READ_A1 := PORT[$A1];
NEW_READ_A1 := READ_A1 AND $FE;
PORT[$A1] := NEW_OKU_A1;
READ_21 := PORT[$21];
NEW_READ_21 := READ_21 AND $FB;
PORT[$21] := NEW_READ_21;
SETINTVEC($70,@TIMER);
PORT[$70] := $B;
PORT[$71] := $43;
PORT[$70] := $C;
READ_71 := PORT[$71];

```

```

rf:=(rf_real * 4095)/1765; { Referance Value for program}

```

```

tt:= t_sample / 1000; { Sampling Time for PID procedure }

```

```

clrscr;

glob_local := 0;
sayi:=0;

{ Boundary Values of parameters of PID Controller }

bound_val[1].Xf := 0;   bound_val[1].Xl := 4.65;
bound_val[2].Xf := 0;   bound_val[2].Xl := 14.5;
bound_val[3].Xf := 0;   bound_val[3].Xl := 0.35;

{ The Penalty Coefficients }

C1 := 30; C2 :=10; C3 := 50; C4 := 100;

{ Starting Conditions for variebles }

Xk0[1] := 0.5;
Xk0[2] := 2.5;
Xk0[3] := 0;

rand_num      := 0;
pnt_eval      := 0;
pnt_rep       := 0;
func_eval     := 0;

hyb_partial_tbl_find_global_max(      bound_val,
global_maximum);
readln;
writeln('points evaluated = ', pnt_eval);
writeln('points replaced = ', pnt_rep);
writeln('functions evaluated -> ', func_eval);
writeln('hybrid partial');
WRITELN(' the maximum point is
(',global_maximum.X[1]:20:20,
', ', global_maximum.X[2]:20:20 );
WRITELN(global_maximum.X[3]:20:20,
', ', global_maximum.X[4]:20:20 );
WRITELN('FUNCTION value at this point is ',
global_maximum.func:20:20);

readln;
END.

```

ÖZGEÇMİŞ

Murat Ata Demiröz 1965 yılında, Üsküdar'da doğdu. İlköğrenimini Sultantepe ilkokulu'nda, orta öğrenimini Özel Moda Koleji'nde yaptı. Lise öğrenimini Haydarpasa Endüstri Meslek Lisesi, Elektronik bölümünde tamamladı. 1982 yılında İTÜ, İnşaat Fakültesi, Çevre Mühendisliği bölümüne girdi. İki yıl bu bölümde okuduktan sonra 1984 yılında Yıldız Üniversitesi, Mühendislik Fakültesi, Elektrik Mühendisliği bölümüne girdi. 1985 yılında İTÜ Elektrik-Elektronik Fakültesi, Elektrik Mühendisliği bölümüne yatay geçiş yaptı. 1988 yılında bu bölümden mezun olup, İTÜ Fen Bilimleri Enstitüsü, Kontrol ve Bilgisayar Mühendisliği Anabilim Dalı, Kontrol ve Bilgisayar Mühendisliği programında Yüksek Lisans eğitimine başladı. 1991 yılı, Mayıs ayında İTÜ Elektrik-Elektronik Fakültesi, Kontrol ve Bilgisayar Mühendisliği Bölümü, Kontrol ve Kumanda Sistemleri Anabilim Dalı'na Araştırma Görevlisi olarak girdi. Halen bu görevini sürdürmektedir.