

75219

PID KONTROLÖR VE PLC UYGULAMASI

YÜKSEK LİSANS TEZİ

Müh. Hatice Nevin HURMA

75219

Tezin Enstitüye Verildiği Tarih : 5 Haziran 1998

Tezin Savunulduğu Tarih : 26 Haziran 1998

Tez Danışmanı : Prof. Dr. Atilla BİR

Diğer Jüri Üyeleri : Doç. Dr. İbrahim EKSİN

Doç. Dr. Salman KURTULAN

İbrahim Eksin
Salman Kurtulan

HAZİRAN 1998

ÖNSÖZ

Bu tezin hazırlanmasında yardımcı olan hocam Sayın Prof. Dr. Atilla BİR'e ve tez çalışmamda bana yol gösteren Sayın Doç. Dr. Salman KURTULAN'a teşekkürlerimi sunmayı bir borç bilirim.

H. Nevin HURMA

Haziran 1998



İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	ii
İÇİNDEKİLER	iii
SEMBOL LİSTESİ	v
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	vii
ÖZET	viii
SUMMARY	ix
BÖLÜM 1 GİRİŞ	1
1.1 Tasarım kriterleri	2
1.2 Kontrolör yapıları	4
1.3 Tasarımda ana prensipler	5
1.3.1 Zaman tanım bölgesi davranışı	6
1.3.2 Frekans tanım bölgesi davranışı	7
BÖLÜM 2 KONTROL SİSTEMLERİ TASARIMI	11
2.1 Oransal Kontrolör	11
2.2 PI Kontrolör	12
2.3 PD Kontrolör	15
2.4 PID Kontrolör	17
2.5 Ziegler-Nichols Kuralları	18
2.6 İntegral Yığılması	19
BÖLÜM 3 DİJİTAL PID KONTROLÖR	21
3.1 Z Düzlemi tasarım kriterleri	23
3.2 Dijital Kontrol Sistemi	28
3.2.1 Dijital PD Kontrolör	28
3.2.2 Dijital PI Kontrolör	29
3.2.3 Kutup-Sıfır Silinmesi	30
3.2.4 Dijital PID Kontrolör	30
BÖLÜM 4 PLC'LERİN İÇ YAPISI	32
4.1 S7 200 PLC'lerine genel bakış	32
4.2 S7 200 PLC Bellek Tipleri	33
4.2.1 Program Belleği	34
4.2.2 Veri Belleği	34

4.2.3 Parametre Belleği	34
4.3 PLC Programının Yürütülmesi	35
4.4 S7 200 PLC Veri Alanı	36
4.5 Veri Belleğine Giriş	40
BÖLÜM 5 PLC'LERİN PROGRAMLANMASI	42
5.1 PLC Programlanmasına giriş	42
5.2 Temel Lojik komutları	43
5.3 Set ve Reset İşlemleri	45
5.4 Zamanlayıcılar	46
5.5 Sayıcılar	46
5.6 Karşılaştırma İşlemleri	48
5.7 Aritmetik İşlemler	48
5.8 Artırma/Azaltma/Tersleme İşlemleri	48
5.9 Veri Transferi Komutları	50
5.10 Organizasyon Komutları	52
5.11 Kesme işlemi	52
5.11.1 Zamana bağlı kesmeler	54
5.12 Dönüştürme işlemi	55
BÖLÜM 6 PID KONTROLÖR VE KONTROL EDİLEN SİSTEM	56
6.1 PLC programı	56
6.2 Kontrol edilen sistem	60
6.3 PID Parametrelerinin hesaplanması	63
6.3.1 PI Kontrolör	63
6.3.2 PID Kontrolör	64
SONUÇLAR ÖNERİLER	66
KAYNAKLAR	67
EKLER	68
ÖZGEÇMİŞ	72

SEMBOL LİSTESİ

ω	: Frekans
ω_g	: Kazanç sınırı payı frekansı
ω_m	: Ziegler-Nichols için osilasyon frekansı
ω_n	: Doğal frekans
ω_p	: Faz sınırı payı frekansı
ω_r	: Rezonans frekansı
BG	: Bant Genişliği
C_D	: Türev terim katsayısı
C_I	: İntegral terim katsayısı
e	: Hata işareti
e_{ss}	: Kararlı hal hatası
G_c	: Kontrolör
G_p	: Kontrol edilen sistem
K_D	: PID Kontrolörün türev kazancı
K_I	: PID Kontrolörün integral kazancı
K_m	: Ziegler-Nichols için osilasyon kazancı
K_m	: Ziegler-Nichols için osilasyon yapan minimum periyot
K_p	: PID Kontrolörün oransal kazancı
K_{pu}	: Ziegler-Nichols için osilasyon yapan minimum kazanç
L	: Maksimum eğimli eğrinin zaman eksenini kestiği nokta
M_{Dn}	: Türevsel terim
M_{In}	: İntegral terim
M_{n-1}	: Biriken toplam integral değeri (bias)
M_{pn}	: Oransal terim
M_r	: Rezonans tepesi
r	: Ayar (referans) değeri
R	: Cevap eğrisinin en yüksek eğimi
T	: Örnekleme zamanı
T_d	: Gecikme zamanı
T_D	: Türev zamanı
T_I	: İntegral zamanı
T_r	: Yükselme zamanı
T_s	: Yerleşme zamanı
u	: Kontrol işareti
y	: Kontrol edilen değişken, çıkış vektörü
ϕ_m	: Faz sınırı payı açısı
ζ	: Amortisman oranı

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 Basit bir kontrol sistemi blok diyagramı	2
Şekil 1.2 Kontrol edilen sistem	2
Şekil 1.3 Çeşitli kontrolör yapıları	5
Şekil 1.4 Zaman tanım bölgesi birim basamak cevap eğrisi	7
Şekil 1.5 Frekans tanım bölgesi cevap eğrisi	8
Şekil 1.6 Kutupsal koordinatlarda gösterilim	9
Şekil 2.1 Oransal Kontrolörün birim basamak cevabı ve blok diyagramı	11
Şekil 2.2 I ve PI Kontrolörlerin birim basamak cevabı ve blok diyagramı	12
Şekil 2.3 PI Kontrolörlü kontrol sistemi	12
Şekil 2.4 PD Kontrolörlü kontrol sistemi	16
Şekil 2.5 PD ve PID Kontrolörlerin birim basamak cevabı ve blok diyagramı	16
Şekil 2.6 Ziegler-Nichols için açık çevrim cevabı	19
Şekil 2.7 İntegral yığılması	20
Şekil 3.1 Örnekleme frekansının s ve z düzlemindeki etkisi	23
Şekil 3.2 Sabit sönüm faktörü ve sabit frekans	25
Şekil 3.3 Sabit sönüm oranı	26
Şekil 3.4 Değişik aşım değerlerinde basamak yanıt eğrisi ve % aşım-ç ilişkisi	26
Şekil 3.5 S ve z-düzlemindeki baskın ve baskın olmayan kutuplar	27
Şekil 3.6 Seri dijital kontrolörlü kontrol sistemi	28
Şekil 4.1 Simatic S7 200 belleği	33
Şekil 4.2 PLC tarama çevrimi	35
Şekil 4.3 PLC giriş ve çıkış görüntü bellekleri	36
Şekil 4.4 S7-200 PLC veri alanı	37
Şekil 4.5 Simatic S7 CPU212 PLC veri belleği	38
Şekil 4.6 Analog giriş ve çıkışlar	39
Şekil 4.7 İki ve tek kutuplu analog giriş bilgisi formatı	40
Şekil 4.8 Analog çıkış akım ve gerilim bilgisi formatı	40
Şekil 4.9 Bit adresleme	41
Şekil 4.10 Byte,word,doubleword adresleme	41
Şekil 5.1 Yükleme,ve,veya,çıkış komutları	44
Şekil 5.2 LPS, LRD, LPP komutları	44
Şekil 5.3 NOT, EU, ED komutları	45
Şekil 6.1 PID kontrolörünün gerçekleşmesi	56
Şekil 6.2 Ana program ve alt programın akış şeması	58
Şekil 6.3 Kesme altprogramın akış şeması	59
Şekil 6.4 PCS 327 Proses kontrol simülatörü bağlantı şeması	62
Şekil 6.5 Kontrol edilen sistemin kök eğrisi	63
Şekil 6.6 PI kontrolörlü sistemin kök eğrisi ve sistemin birim basamak cevabı	64
Şekil 6.7 Kutup yok etmeli PI kontrolörlü sistemin kök eğrisi	64
Şekil 6.8 PID kontrolörlü sistemin kök eğrisi	65

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 1.1 Kontrol sistem kriterleri	10
Tablo 2.1 Ziegler-Nichols Kuralları	19
Tablo 3.1 Çeşitli s ve z düzlemi kök yerleri ve birim basamak cevapları	24
Tablo 4.1 Simatic S7 212/214 özellikleri	32
Tablo 5.1 Set ve reset komutları	45
Tablo 5.2 Zamanlayıcı komutları	47
Tablo 5.3 Sayıcı komutları	47
Tablo 5.4 Karşılaştırma işlem komutları	49
Tablo 5.5 Çarpma ve bölme işlemi	49
Tablo 5.6 Toplama ve çıkarma işlemi	50
Tablo 5.7 Artırma, azaltma ve tersleme komutları	50
Tablo 5.8 Veri transferi komutları	51
Tablo 5.9 Organizasyon komutları	52
Tablo 5.10 Kesme komutları	55
Tablo 6.1 PLC’de hesaplamaların yapıldığı veri alanları	58

ÖZET

Bu çalışmada otomatik kontrol sisteminin temel kavramları üzerinde durulmuş, PID kontrolörün yapısı, zaman ve frekans bölgesinde tasarımı açıklanmıştır. PID kontrolörün gerçekleştirileceği Simatic S7-200 PLC üzerinde durulmuş, iç yapısı , programlama komut ve yöntemleri gösterilmiştir. PLC’de PID kontrolörü algoritması verilmiş ve PID kontrolörün PLC’de nasıl gerçekleştirileceği anlatılmıştır.

PID kontrolör parametre ayarlanması ile her türlü sisteme adapte edildiğinden sık kullanılan bir kontrolördür. Analog PID kontrolörlerde önce tasarım yapılır, sonra donanım oluşturulur. Fakat dijital kontrolörler yazılım ile oluşturulduğundan böyle bir sorun yoktur, parametreler yazılımdaki ufak oynamalarla kolaylıkla değiştirilebilir. Yalnız integral yığılması denilen kontrolördeki integral teriminin büyümesi ile ilgili olan sorun mevcuttur. Çıkış sınırlı olduğundan bu sorun ortaya çıkar. Bu da, her çevrim sırasında integral teriminin sınırlandırılması ile ve ayar değerinin tam skala değil de yarısı dolaylarında seçilmesi ile önlenabilir.

SUMMARY

THE PID CONTROLLER AND PLC APPLICATIONS

The controller compares the reference variable or set point (r) with controlled variable(y) and generates a signal (u) for driving the final control element. In the steady state, the output variable of the controller remains at a fixed value.

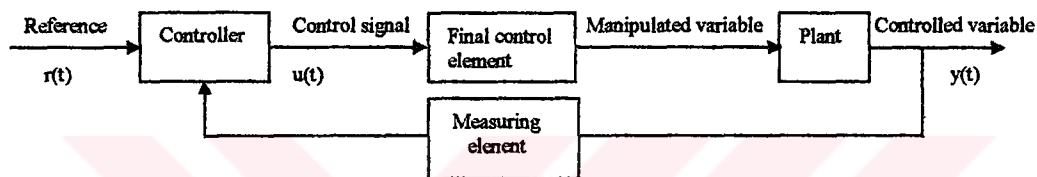


Figure 1 A basic block diagram of control systems

The P controller produces an output variable u proportional to the system deviation e . Relationship is as follows: [1]

$$u = K_c \cdot (r - y) = K_c \cdot e$$

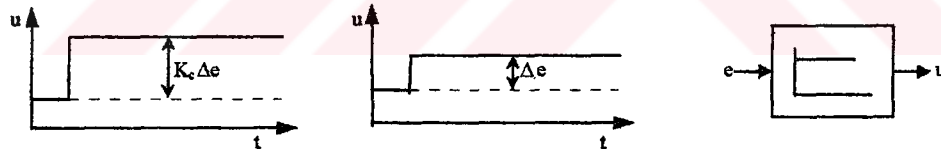


Figure 2 Step response and block diagram of a P controller

The P controller reacts very quickly to a system error because its output variable acts instantaneously on the final control element. However, P controller can not eliminate disturbances completely or correct the controlled variable y to the set point r . A steady state system deviation or offset always remains.

The I controller produces control signals proportional to the integral of the error and gives an increasing signal which eliminate the offset as long as the error is nonzero. The integral action coefficient K_I indicates by how many units the output variable of the I controller changes during the period Δt when the system deviation Δe is present at the input. The integrating time can be written as $T_I = 1 / K_I$.

The instantaneous response of proportional-control action and the eliminating offset properties of integral-controller action can be combined. The step response of the proportional-plus-integral action (PI) controller has a proportional action

component with an integral action component superimposed. Integral time is the time required for the integral action signal to equal that the proportional term if a constant error exist.

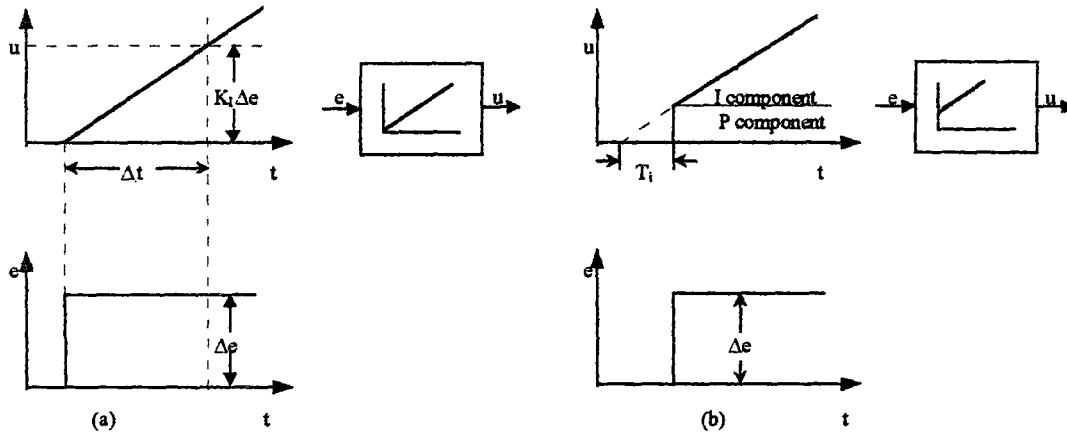


Figure 3 Step response and block diagram of I and PI controllers

PD controller has a strong effect on the controller output variable for a certain period of time when there are rapid changes in the controlled variable. PD controller produces signal according to the derivative of the system deviation changes. T_D is the derivative action time or rate time which governs the magnitude of D component.

The PID controller is the combination of proportional-integral-derivative actions and has the advantages and disadvantages of those. The step response of the PID controller is governed by three adjustable values as T_D , K_c , T_I . Analog PID controllers are as below;

$$G_c(s) = K_c \left[1 + \frac{1}{T_I s} + T_D s \right]$$

The digital implementation of the PID controller by using forward rectangle approximation can be written as; [2]

$$G(z) = K_c \left[1 + \frac{T}{T_I} \cdot \frac{z}{z-1} + \frac{T_D}{T} \cdot \frac{(z-1)}{z} \right]$$

$$G_c(z) = \frac{(K_c + K_D/T + T K_I) z^2 - (K_c + 2K_D/T) z + K_D/T}{z(z-1)}$$

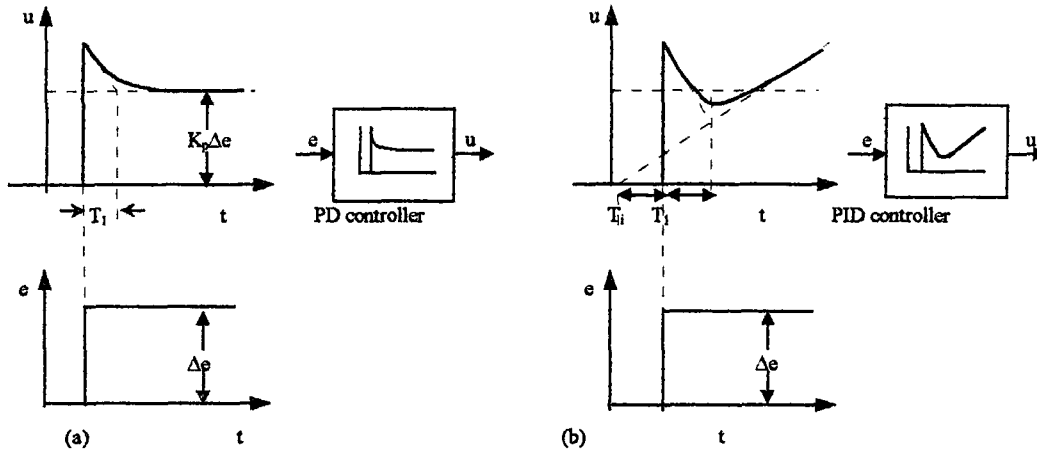


Figure 4 Step response and block diagram of PD and PID controllers

Because three parameters must be adjusted in the design of PID controllers, root locus and bode design techniques are usually not used directly. Ziegler and Nichols developed a method for tuning a PID controller, which is based on a simple stability analysis. First, set $K_D = K_I = 0$, and then increase the proportional gain until the system just oscillates (i.e., closed loop poles on the $j\omega$ axis). The proportional gain is then multiplied by 0.6, and the other two gains are calculated as [3]

$$K_c = 0.6 K_m \quad K_D = \frac{K_c \pi}{4 \omega_m} \quad K_I = \frac{K_c \omega_m}{\pi}$$

where K_m is gain at which the proportional system oscillates, and ω_m is the oscillation frequency. Note that this technique does not design to any specification. Rather, Ziegler and Nichols found that this design procedure provided “good” behavior for process controllers. Years of experience by process control engineers have indicated that it is indeed a good technique

Figure 1 shows the block diagram containing the basic elements of a single-loop continuous-data control system. The controller is usually an electronic circuit that operates on an analog signal and outputs the same type of signal. The advantage of the analog controller is that the system operates in real time and is capable of a very high bandwidth. It is equivalent to having an infinite sampling frequency, so that the controller is effective at all times. The disadvantages of the analog controller are that its elements are usually hard-wired, so that their characteristics are fixed, making it more difficult to make design changes, and component aging and sensitivity to environmental changes can be quite severe. Analog components are also more susceptible to noise problems [4].

However, digital components are less susceptible to aging and environmental variations. They are less sensitive to noise and disturbance. Digital processors are more compact and lightweight. The costs of digital controllers are going down. They allow more flexibility in programming: changing a design does not require an alteration in the hardware. They are more reliable and provide improved sensitivity to parameter variations.

Digital control systems are not without disadvantages when compared with the analog systems. Limitations on computing speed and signal resolution due to the finite wordlength of the digital processor. In contrast, analog controller operate in real time, and the resolution is theoretically infinite. The finite wordlength of the digital processor often translates into system instability in the form of limit cycles in closed-loop systems. The limitation of computing speed causes time delays in the control loop which may cause instability in closed-loop systems.

HANDLING PID ON THE PLC

A PID controller adjust the value of its output to try and balance the value of the proses variable to a given “set point”. To calculate the output value for a given instance, the controller finds the value of three different terms (using its user defined *Sample time*, *Gain*, *Rate* and *Reset* values along with the calculated *Error* value): a Proportional term, an Integral term and a Derivative term [5].

$$\text{Output} = M_{Pn} + M_{In} + M_{Dn}$$

The *sample rate* is the cycle time at which the PID loop recalculates the output. The *gain* controls the sensitivity of the output calculation by affecting the influence of all three terms. The *reset* time is a time which is used to increase or decrease the influence of the Integral term in the equation. Finally, the *rate* value is used to control the influence of the Derivative term in the equation. Each of these values needs to be preset by the user before the PID controller starts.

If the user does not want integral action (no I in the PID calculation), then a value of infinity should be specified for the integral time. If the user does not want derivative action (no D in the PID calculation), then a value of zero should be specified for the derivative time.

Error is figured as the difference between the normalized values of the setpoint and the proses variable. The controller calculates this value in three steps. The first two steps are to change both the set point and the proses variable into values that are based on a “0” to “1” (normalized) scale, this is done using the formula;

$$\begin{aligned} \text{Set point} & : SP = \text{raw_SP} / \text{max_val} \\ \text{Proses value} & : PV = \text{raw_PV} / \text{max_val} \end{aligned}$$

In the above formula, the raw_SP and raw_PV values are the actual values that come into the controller and the max_val term is the maximum value that either can take on. In our condition, the max_val is 32.752 After these two values have been calculated, the error term is figured as follows;

$$\text{Error} = SP - PV$$

The proportional term, M_{Pn} , is calculated using the following equation;

$$M_{Pn} = \text{Gain} * \text{Error}$$

The integral term, M_{In} , is calculated using the following equation;

$$M_{In} = K_c \sum_{1}^n e \ T / T_I + M_{initial} = (K_I / T_I) . e_n + M_{n-1}$$

$$M_{In} = C_I * \text{Error} + \text{Bias}$$

C_I is the coefficient of the integral term and is calculated from the *reset*

$$C_I = \text{Gain} * \text{sample} / \text{reset}$$

The larger *reset* value is, the less impact the integral term will have on the output, while larger *sample* times give it a bigger influence (sample time is also affects the derivative term, which will be explained later). The bias term represents (technically speaking) the area under the curve of a graph plotting the Error versus time. Abstractly, however, the *bias* value (ideally) grows to an output level that keeps the system stable, letting the proportional and derivative terms handle any small fluctuations.

The derivative formula for a given time n is calculated with the following equation;

$$M_{Dn} = C_D * (\text{Error}_{n-1} - \text{Error}_n)$$

$$C_D = \text{Gain} * \text{rate} / \text{sample}$$

In the derivative term, the *sample* time is indirectly proportional to the derivative component, while the *rate* is directly proportional.

Because of system inertia, the output value could take on a value greater than 100% or less than 0% (for some systems -100%). In response to this, the PID program implements Output Clamping. If the output is greater than 100%, then it is clamped to 100%. If the output falls lower than 0% (-100%), then it is held to 0% (-100%).

The only problem left to solve lies with the *bias* portion of the integral term. When the output for a system remains at 100% for a long period of time, the integral sum (which the *bias* term represents) can grow to extremely large values. This means that when the variable starts responding, the bias term will be keeping the calculated output well over 100% until it can be wound down. This generally results in the output swinging wildly from one limit to the other, but can be avoided using bias clamping. There are a few different types of bias clamping and there are two different conditions which cause bias blamping to occur;

$$\text{If Output} > 1 \quad \text{bias} = 1 - (M_{Pn} + M_{Dn})$$

$$\text{If Output} < 0 \quad \text{bias} = - (M_{Pn} + M_{Dn})$$

Program sequence is that;

- Read in the proses variable
- Set the set point
- Adjust the scale for the proses value and set point inputs
- Adjust the reset, rate, gain and sample time values

- Calculate the output
- Write the output
- Adjust the scale for the output

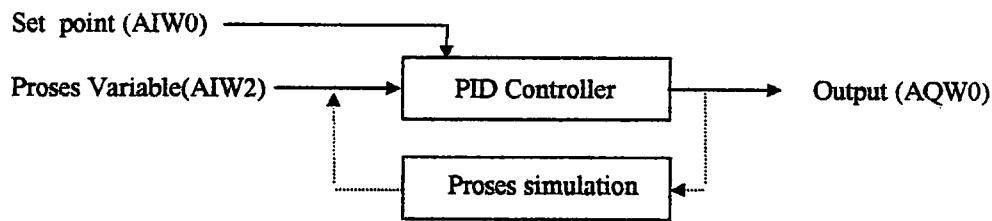


Figure 5

The flow chart of the PID controller as shown below,

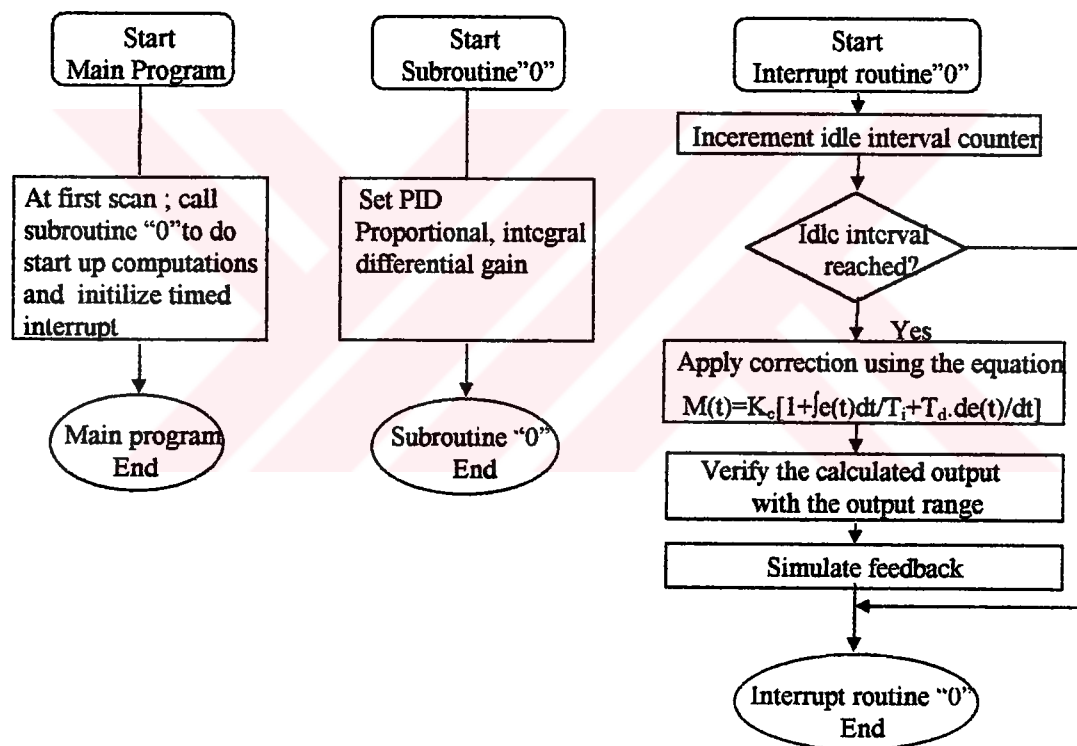


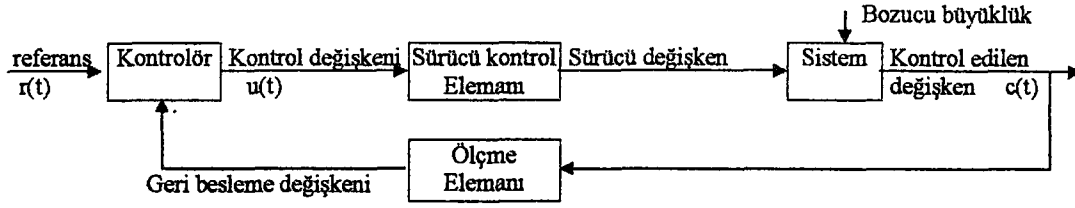
Figure 6 The flow chart of the PID controller

BÖLÜM 1

GİRİŞ

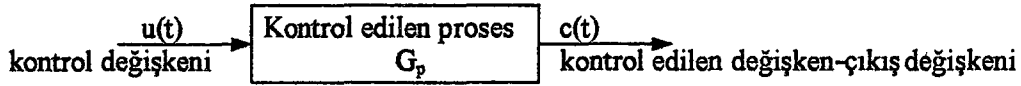
Otomatik kontrolün amacı sistemin değişken büyüklüklerini arzu edilen değerlerde tutarak kararlı bir çalışma ortamı yaratmaktır. Bir otomatik kontrol sisteminde giriş büyüklükleri değiştikçe çıkış büyüklüklerinin de mümkün olduğu kadar kısa bir sürede ayar değerlerine ulaşması ve bozucu büyüklükler tarafından meydana gelen değişimlerin sistem tarafından kendi kendine giderilmesi arzu edilir. Eğer kurulu bir düzende bu özellikler sağlanmıyorsa, sisteme kontrol cihazı eklenerek arzu edilen koşulların gerçekleşmesine çalışılır. Bir kontrol sisteminin başarısı şu üç özelliği gerçekleştirme oranı ile ölçülür; sistemde meydana gelen herhangi bir bozucu büyüklükten sonra normal çalışma şartlarından minimum sapma, bozulan çalışma şartlarını en kısa zamanda normale dönüştürmek, çalışma şartlarında meydana gelen değişimlerden ötürü sistemde minimum kalıcı hatanın oluşması [6].

Tipik bir kontrol sistemi şekil 1.1'deki gibi gösterilebilir. Kontrol edilen sistem, devamlı olarak bir bozucu büyüklüğün veya sistemin dışında meydana gelen bazı etkenlerin tesiri altındadır. Söz konusu büyüklüğe karşı düzeltme yapabilmek için çıkış büyüklüğünün yani kontrol edilen değişkenin ölçülmesi gerekir. Bu iş ölçme elemanı tarafından yapılır. Kontrol cihazı ölçme elemanından kontrol edilen değişkenin gerçek değerini alır ve sistemin ne kadar iyi çalıştığına karar verir. Kontrol organı üzerinde istenen değerin ayar edildiği bir kısım vardır. Eğer herhangi bir fark söz konusu ise kontrol organı düzeltme hareketini başlatacak olan kontrol işaretini üretir. Ayar değeri elle sabit bir değere ayarlanabileceği gibi, otomatik olarak kontrol sistemi tarafından şartlara göre değiştirildiği durumlar da olabilir. Sürücü kontrol elemanı aldığı kontrol işaretinin değerine göre gerekli ayarlamaları yapan elemandır. Sürücü değişken, kontrol edilen değişkende istenen düzeltmeyi sağlamak üzere otomatik kontrol sistemini etkiler.



Şekil 1.1 Basit bir kontrol sistemi blok diyagramı

Şekil 1.2’de blok diyagramla gösterilen bir proses için kontrol sistem tasarımı aşağıdaki üç aşamada gerçekleştirilir [7].



Şekil 1.2 Kontrol edilen sistem

- 1) Sistemin neyi, nasıl yapması gerektiğinin belirlenmesi (tasarım kriterleri)
- 2) Kontrolör yada kompanzatörün kontrol edilen prosese nasıl bağlanacağını belirlenmesi
- 3) Tasarım amaçlarına ulaşmak için kontrolör parametre değerlerinin nasıl seçileceğidir.

1.1 TASARIM KRİTERLERİ

Tasarım kriterleri her uygulama için farklıdır ve görelî kararlılık, kararlı hal hatası, geçici cevap ve frekans cevabı kriterlerini içerir. Bazı uygulamalarda parametre değişimlerine karşı duyarlılık (dayanıklılık, bozucu etkileri giderme) gibi ek işlevlerin de bulunması istenir.

Lineer kontrol sistemleri tasarımı hem zaman hem de frekans tanım bölgesinde gerçekleştirilebilir. Örneğin kararlı hal hatası genellikle birim basamak, rampa ve parabolik giriş için tasarlanır ve uygulayıcı bakımından hatanın elde edilmesi zaman tanım bölgesinde daha kolaydır. Maksimum aşım, yükselme zamanı, yerleşme zamanı gibi diğer kriterler birim basamak giriş için geliştirilmiştir ve zaman tanım bölgesinde tasarım sırasında kullanılır. Görelî kararlılık, aynı zamanda kazanç payı, faz payı ve M_r tepe değeri ile de ölçülür. Bunlar tipik frekans tanım bölgesi

kriterleridir ve Bode diyagramı, Kazanç-faz diyagramı, Nichols abağı ile kullanılır.

Tasarımın zaman yada frekans tanım bölgesinde yapılması tasarımcının seçimine bağlıdır. Bir çok durumda maksimum aşım, yükselme zamanı ve yerleşme zamanı kriterleri sistem davranışının sınır değişkenleri olarak kullanılır. Deneyimsiz tasarımcılar için frekans tanım bölgesi kriterlerinin zaman tanım bölgesi kriterlerine etkilerinin nasıl olduğunu kestirmek, kazanç ve faz payı, rezonans tepesi gibi frekans tanım bölgesi kriterlerinin gerçek sistem davranışı ile ilişkisini anlamak zordur. Örneğin, 20 dB'lik kazanç payı maksimum aşımın %10'dan düşük olmasını garantiler. Maksimum aşımın %5'den küçük ve yerleşme zamanının 0.01 sn.'den küçük olması gerektiği daha anlaşılırdır. 60° 'lik faz payı ve 1.1'den küçük M_r 'nin sistem davranışına ne getireceği pek açık değildir.

Tarihsel açıdan bakarsak, lineer kontrol sistemlerinin tasarımı ilkin Bode diyagramı, Nyquist diyagramı, Kazanç-faz diyagramı ve Nichols abağı gibi frekans tanım bölgesinde kullanılan grafiksel yöntemlerden yararlanarak geliştirilmiştir. Bu yöntemlerin üstünlüğü; etraflı bir çizim gerektirmeksizin yaklaşık çizilebilmesidir. Bu nedenle tasarımcı kazanç-faz payı ve M_r gibi frekans tanım bölgesi kriterlerini kullanarak tasarımı gerçekleştirir. Yüksek mertebeden sistemler önemli bir sorun oluşturmaz.

Frekans tanım bölgesinde belli bir kontrolör tipi için deneme yanılmayı en aza indirecek tasarım işlemleri mevcuttur. Frekans tanım bölgesi yöntemleri gürültü etkisini giderme, sistem duyarlılığına ve proses tasarımına başka açıdan bakmak için kullanılabilir.

Zaman tanım bölgesi kullanılarak yapılan genel tasarım işlemlerini, ikinci mertebeden yüksek sistemler için kullanmak zordur. Yükselme, gecikme, yerleşme ve maksimum aşım gibi zaman tanım bölgesi kriterleri ile tasarım yalnızca ikinci mertebeden yada ikinci mertebeden sistemlere benzeştirilen sistemler için analitik olarak uygulanabilir.

Günümüzde bilgisayarlar yardımıyla kontrol sistem tasarımı çok kolay ve az zamanda gerçekleştirilmektedir. Modern bilgisayar yazılımları ile; tasarımcı birkaç dakika içinde zaman tanım bölgesi kriterlerini kullanarak bir çok tasarım yapılabilir. Bu durum grafiksel tasarımın kolaylığına sahip olan frekans tanım bölgesi tasarımının önemini büyük ölçüde azaltır.

1.2 KONTROLÖR YAPILARI

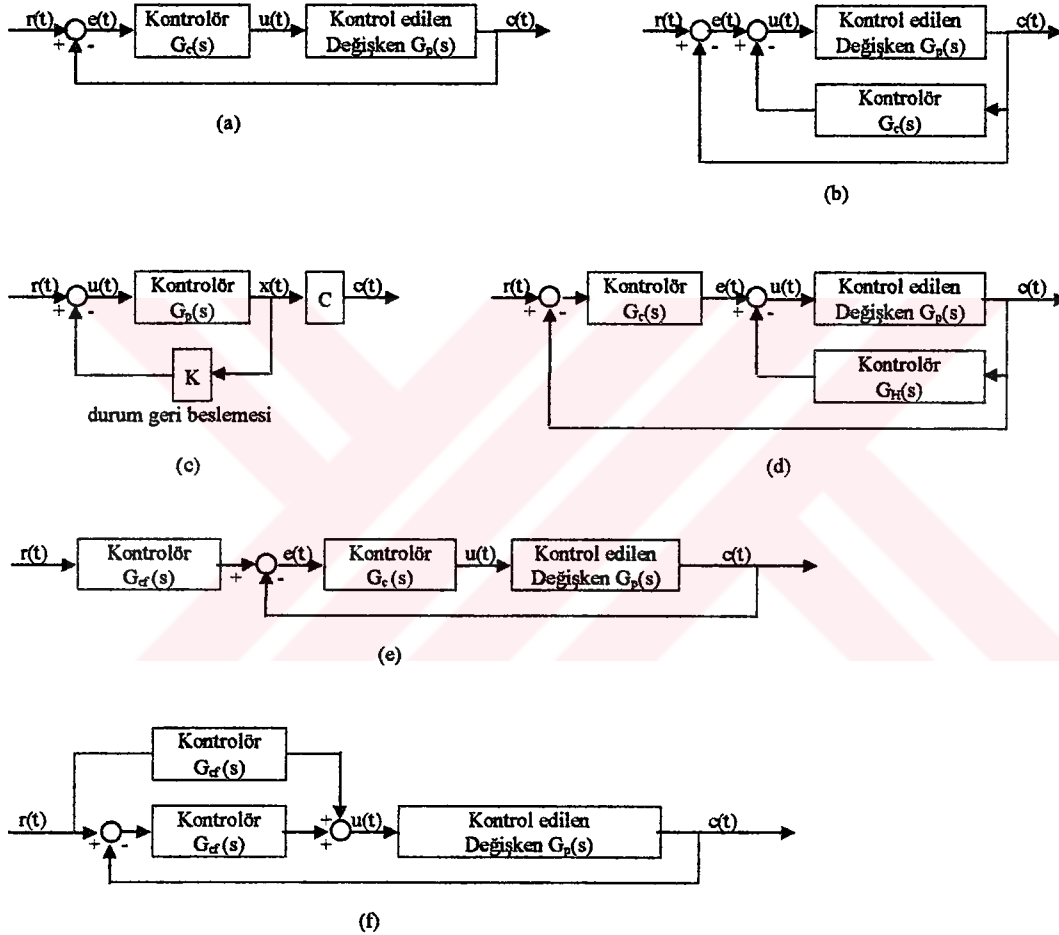
Genel olarak lineer sistemlerin dinamiği şekil 1.2’de verilen blok diyagram ile gösterilir. Tasarımın amacı $c(t)$ ile gösterilen kontrol edilen değişkenin istenen şekilde davranmasını sağlamaktır. Asıl problem tasarım kriterlerinin tümünü sağlayacak olan $u(t)$ kontrol işaretini belirlemektir.

Şekil 1.3 en çok kullanılan kontrol sistem kompanzator yapılarını göstermektedir. Bunlar kısaca açıklanacak olursa; Seri kompanzasyon (şek.1.3a) kontrolörün kontrol edilen proses ile seri olmasıdır. Geri beslemeli kompanzasyon’da (şek. 1.2b) kontrolör negatif geri besleme yoluna yerleştirilmiştir. Durum geri beslemeli kompanzasyon (şek.1.3c) sabit bir kazanç ile durum değişkeninin geri beslenmesi ile kontrol işareti üreten bir sistemi gösterir. Bu durum geri beslemesi olarak bilinir. Geri beslemeli kontrolde problem, yüksek mertebeden sistemlerde bulunan birçok durum değişkeninin geri beslenmesi için bir çok algılayıcıya (dönüştürücüye) ihtiyaç duyulmasıdır. Geri beslemeli kontrolörün gerçekleştirilmesi pahalı ve pratikte uygulanamaz olabilir. Düşük mertebeden sistemler için bile; bir çok durum değişkenleri doğrudan erişilebilir değildir ve çıkış değişkeninden ölçüm ile durum değişkenlerinin saptanması için gözlemleyici veya kestirimci gerekli olabilir.

Şekil 1.3a,b,c ’de gösterilen kompanzasyon şemaları bir kontrolörlü sistemleridir. Bir kontrolörlü sistemlerin dezavantajı gerçekleştirilebilen davranış kriterlerinin sınırlı olmasıdır. Örneğin sistem belli bir göreceli kararlılığa ulaşmak için tasarlanacaksa, parametre değişimlerine karşı duyarlılığı zayıf olabilir yada karakteristik denklemin kökleri belli bir göreceli sönümü sağlamak için seçilmişse birim basamağa karşı maksimum aşım kapalı çevrim transfer fonksiyonunun sıfırlarına bağlı olarak çok büyük olabilir. Şekil 1.3d,e,f iki kontrollü sistemleri gösterir. Seri Geri Beslemeli Kompanzasyon (şek.1.3e), bir geri besleme ve bir seri kontrolörden oluşan sistemdir. İleri Kompanzasyon’da (şek.1.3e/f) $G_{cf}(s)$ ileri besleme kontrolörü ileri yolunda $G_c(s)$ kontrolörü olan kapalı çevrim sistemi ile seri olarak yerleştirilmiştir. Şekil 1.3.f ’de $G_{cf}(s)$ ileri besleme kontrolörü ileri yol ile paralel yerleştirilmiştir. İleri beslemeli kompanzasyonun üstünlüğü; $G_{cf}(s)$ kapalı çevrimin içinde değildir. Böylece özgün sistemin karakteristik denkleminin köklerine etki etmez. $G_{cf}(s)$ ’ın kutup ve sıfırları kontrol edilen kapalı çevrim sisteminin transfer

fonksiyonunun kutup ve sıfırlarını iptal etmek veya kutup ve sıfır eklemek için kullanılır.

Yukarıda bahsedilen kompanzasyon çeşitlerinin içinde en çok kullanılanı prosese uygulanan işaretin türev, integral ve oransal kombinasyonundan oluşan bir işaret uygulayan PID kontrolördür. Kolay gerçekleştirilebilir ve zaman tanım bölgesinde kolay görüntülenebilir olduğundan, PID kontrolörler daha çok zaman tanım bölgesi yöntemleri ile tasarlanırlar.



Şekil 1.3 Çeşitli kontrolör yapıları

1.3 TASARIMDA ANA PRENSİPLER

Kontrolör yapısı seçildikten sonra, tasarımcı istenen özellikleri sağlayacak kontrolör tipini seçmelidir. Bir kontrol sistem tasarımında bir çok çözüm bulunabilir. Pratikte tasarım özelliklerini sağlayan en basit kontrolör seçilir. Kontrolör ne kadar

kompleks olursa, o kadar pahalı, az güvenilir ve tasarımı daha zor olur. Belli bir uygulama için belli bir kontrolör seçmek geçmişteki deneyimlere göre bazen de sezgisel olarak mümkün olabilir. Sonuç olarak tasarım işinde yeni olanlar en uygun çözümü bulmakta zorlanırlar.

Kontrolör seçildikten sonra, sıra kontrolör parametre değerlerini seçmeye gelir. Bu parametreler kontrolörün transfer fonksiyonunun katsayılarıdır. Temel tasarım yaklaşımı şudur; ayrı ayrı parametrelerin tasarım özelliklerine ve sistem davranışına nasıl bir etkisinin olduğu tayin edilir. Bu bilgiyi temel alarak tüm tasarım özelliklerini karşılayacak şekilde parametre seçimi yapılır.

Kontrolör parametreleri genellikle birbirini etkiler ve tasarım şartlarına karşı yönde etkide bulunur. Örneğin bir parametre ile maksimum aşım şartı sağlanırsa, yükselme zamanını düzeltmek için başka bir parametrenin ayarlanması ile maksimum aşım şartı bozulabilir. Tasarım şartları ne kadar fazla ise kontrolör parametresi o kadar çoğalır ve o kadar kompleks tasarım prosesi meydana gelir.

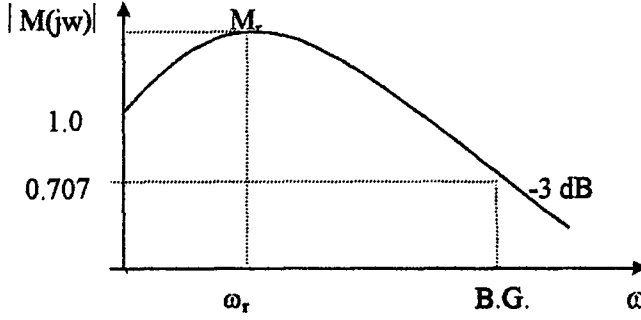
Zaman yada frekans tanım bölgesinde tasarım yaparken temel tasarım kural ve yöntemlerini ele almak önemlidir. Zaman tanım bölgesinde tasarım s-düzleminde ve köklerin yer eğrisine dayanarak yapılır. Frekans tanım bölgesinde ise kapalı çevrim transfer fonksiyonunun kazanç ve fazının hesaplanması temel alınır. Tasarım için kılavuz olan frekans-zaman tanım bölgesi kriterlerini inceleyelim.

1.3.1 ZAMAN TANIM BÖLGESİ DAVRANIŞI

Zaman tanım bölgesi davranışı birim basamak cevaba karşın incelenir. Bir ikinci mertebeden birim geribeslemeli bir sistemin zaman tanım bölgesi cevabının tipik bir şekli aşağıda verilmiştir.

$$c(t) = 1 - \frac{1}{(1-\zeta^2)} e^{-\zeta\omega_n t} \sin(\omega_n \sqrt{1-\zeta^2} t + \cos^{-1}\zeta) \quad (2.1)$$

Bir otomatik kontrol sistemine basamak şeklinde bir giriş uygulandığı takdirde ortalama çıkışın nihai değerinin yarısına erişinceye kadar geçen zamana gecikme zamanı (T_d) denir. Birim basamak giriş cevap eğrisinin %10 ile %90'ı arasındaki zamana yükselme zamanı (T_r) denir. Birim basamak giriş cevap eğrisinin son değerinin %x bölgesine, bundan sonra değişimi bu bölge içerisinde kalmak koşulu ile ilk girdiği ana kadar geçen zamana yerleşme zamanı (T_s) denir. Genelde $x=\%5$ yada



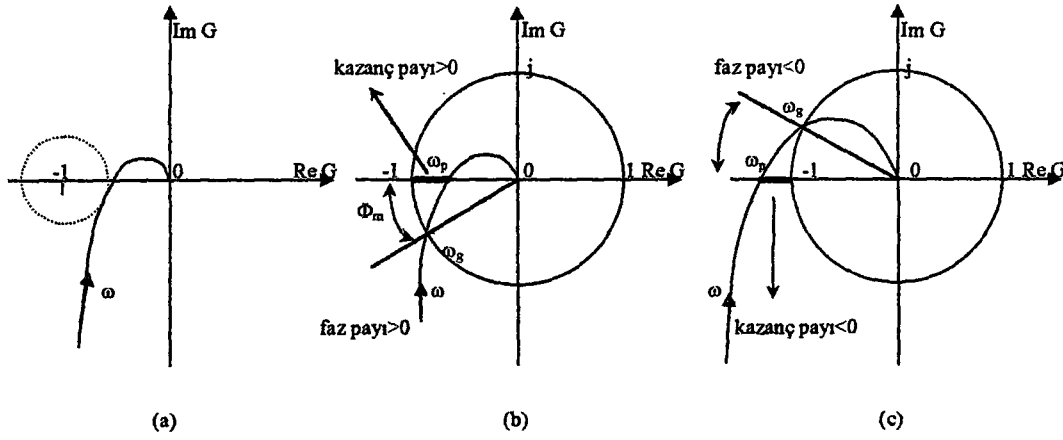
Şekil 1.5 Frekans tanım bölgesi cevap eğrisi

M_r tepe değeri, görel kararlılık kriteri olarak tanınır, bu değer frekans cevabı eğrisinde maksimum değere karşı düşer ve boyutsuz bir değerdir. Büyük M_r tepe değeri zaman tanım bölgesinde büyük aşım değerini verir. Frekans tanım bölgesi davranışında M_r için uygun değer aralığı $M_r = 1.2 - 1.5$ 'dur. M_r tepesinin meydana geldiği frekansa ω_r , rezonans frekansı adı verilir. BG bant genişliği, sıfır frekansı ile normalize edilmiş kapalı çevrim transfer fonksiyonunun büyüklüğünün 0.707 birim veya -3 dB olduğu frekans arasındaki rad/s. cinsinden frekans tanım bölgesidir. Bant genişliği büyük olan kapalı çevrim kontrol sistemlerine yüksek frekanslı gürültü yada bozucu işaretler sistem tarafından iletilir. Bu tür sistemler hızlı ve büyük aşımlı olur. Bant genişliği küçük olan sistemler ise küçük aşım değerli fakat yavaş davranışlı olur. Kazanç payı, sistemin görel kararlılığını belirler. Açık çevrim transfer fonksiyonu $G(j\omega)H(j\omega)$ eğrisi şekil 1.6' da gösterilen kutupsal biçimde çizilsin. Eğri $(-1,0)$ noktasından ne kadar uzak olursa sistem o kadar kararlı olur. Bu da şekil 1.6a.'da gösterilen merkezi $(-1,0)$ olan Nyquist eğrisine teğet olan çemberin yarıçapı ile ölçülür. Bu şekilde hesaplama, zor olacağından şekil 1.6b ve c'deki incelemeler dikkate alınır. Şekildeki ω_p kesim noktası (-1) noktasından ne kadar merkeze yakın olursa (Nyquist kararlılık kriterinden) sistem o kadar kararlı olur. Bu durumda $\overline{\omega_g}$ uzaklığı görel kararlılığın bir ölçütüdür. Kazanç sınırı payı aşağıdaki gibi tanımlanır.

$$20 \log_{10} \frac{1}{|G(j\omega_c)H(j\omega_c)|} \text{ dB} \quad (2.3)$$

kazanç sınırı payının meydana geldiği frekansa ω_p kazanç payı frekansı ismi verilir. Kararlı sistemler için kazanç payının pozitif olması gerekir. Faz payı da, görel kararlılığın bir ölçütüdür. $G(j\omega)H(j\omega)$ yer eğrisinin birim daireyi kestiği nokta ω_g ise, $\overrightarrow{\omega_g}$ vektörü ile negatif reel eksen arasında kalan açıya faz sınırı payı açısı (ϕ_m) denir. $\angle G(j\omega_g)H(j\omega_g)$ saat yönünde ölçülmelidir. Kararlı sistemler için; [7]

$$\phi_m = 180^\circ - \angle G(j\omega_g)H(j\omega_g) \quad (2.4)$$



Şekil 1.6 Kutupsal koordinatlarda gösterilim

Kazanç ve faz payının büyük olması kapalı çevrim kontrol sisteminin daha kararlı olmasını sağlar, bu da zaman tanım bölgesi davranışının daha yavaş olması demektir. Şekil 1.6.b’de olduğu gibi eğri birim daireyi üçüncü çeyrek bölgede keserse faz sınırı payı açısı pozitifdir. Kararsız sistemlerde ; Kazanç payı ve faz payı negatiftir. Pratikte kazanç payı 10...20 dB arasında , faz payı ise $+30^\circ \dots +45^\circ$ arasında seçilir. Bu değerlerle kapalı çevrim kontrol sisteminin kararlılığı sağlanmış olur ve zaman tanım bölgesi davranışı da tatmin edici hale gelir [9].

Kontrolör tasarımın amacı; tüm sistemin zaman tanım bölgesi kriterlerinin karşılanmasıdır. Tasarım frekans tanım bölgesinde yapıldığından sistemin kapalı çevrim transfer fonksiyonu $T(s)$ kriterleri frekans tanım bölgesine çevrilmelidir. Açık çevrim transfer fonksiyonunun $G_1(s) = G_c(s).G_p(s)$ Bode diyagramı çizilir. İstenen kriterleri karşılayacak şekilde $G_c(s)$ kompanseörü bulunur. Daha sonra tüm sistemin bu kriterleri karşılayıp karşılamadığı kontrol edilir. $G_c(s)$ ’in aranması genelde deneme-sınama ile olur. İşe basit bir kompanseörden başlanır, eğer başarılı olunmazsa daha karmaşıklarına geçilir. Kazanç ekleyen/çıkarıcı, faz geciktiren, faz ilerleten olmak üzere üç tip kompanseör vardır. PI kontrolör faz-geriletmeli kompanseörün sıfırının orijinde olması, PD kontrolör ise faz ilerletmeli kompanseörün sıfırının sonsuzda olması özel halidir. Tablo 1.1’de zaman ve frekans tanım bölgesi kriterlerinin karşılıkları verilmiştir [2].

Sistemin sağ yarı düzlemde açık çevrim kutbunun ve sıfırının olmadığı kabulü ile sistemin fazı, kazanç eğrisinin asimptotlarının eğimleri ile tahmin edilebilir. Eğer eğim 20 dB/decad ise faz -90° ye, 40 dB/decad ise faz -180° ye, 60 dB/decad ise faz -270° ye yaklaşır. Bu özellikten yola çıkarak, tasarımda ω_g frekansında asimptotun -20 dB'lik eğime sahip olması sağlanmalıdır, bu şekilde faz payı her durumda pozitif ve sistem kararlı olur.

Tablo 1.1 Kontrol sistem kriterleri

Zaman tanım bölgesi		Frekans tanım bölgesi	
	Tüm sistem	Tüm sistem	Çevrim transfer fonksiyonu
Sürekli hal	Konum hatası Hız hatası	$ 1-T(0) $ $T(0)=1$ ise $ T'(0) $	$ 1/(1+K_p) $ $ 1/K_v $
Geçici hal	Aşım Yükselme zamanı Yerleşme zamanı	Tepe değeri Band genişliği Yüksek frekans kazancı	Kazanç ve faz payı Faz sınırı payı frekansı ω_g Yüksek frekans kazancı

Genel olarak zaman ve frekans tanım bölgesi kriterleri özetlenirse aşağıdaki kısa açıklayıcı bilgiler ortaya çıkar.

1-Kapalı çevrim transfer fonksiyonunun karmaşık eşlenik kutupları basamak cevabın sönümsüz olmasına yol açar. Eğer sistem kutupları gerçel ise basamak cevabı sönümlüdür. Ancak kapalı çevrim transfer fonksiyonunun sıfırları, sistem sönümlü olsa dahi aşım neden olabilir.

2- S-düzleminde sıfıra en yakın olan kutuplar sistem cevabında baskın olurlar, sistem çabuk cevap verir ve bant genişliği büyüktür. Diğer orijine uzak kutupların etkileri hemen geçici cevapta kaybolduğundan etkileri az olur.

3- Transfer fonksiyonunun birbirine çok yakın sıfır ve kutupların sistem yanıtına etkisi çok küçük olur.

4- Zaman ve frekans tanım bölgesi kriterleri birbirleri ile ilişkilidir. Yükselme zamanı ile bant genişliği ters orantılıdır.

BÖLÜM 2

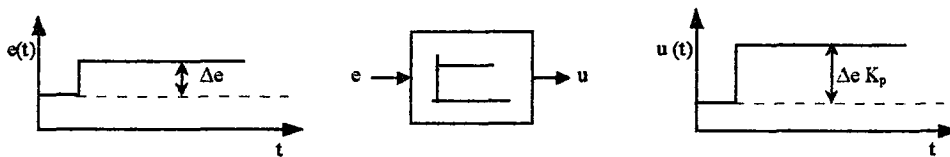
PID KONTROLÖR TASARIMI

PID kontrolör oransal, integral ve türev elemanlarının toplamından oluşur. Her üç kısım kullanılabileceği gibi sadece bir yada iki kısmın da kullanılması yeterli olabilir. Bu şekilde P, I, PI, PD, PID kontrolör olarak ayırabiliriz. Kontrolörlere toplayıcı, çıkarıcı, yükseltici, zayıflatıcı, türev ve integral alıcı gibi elemanlar ilave edilebilir. Tasarımcı hangi elemanların hangi oranda kullanmak gerektiğini ve nasıl bağlanması uygun olduğunu tasarlamalıdır. Uygulamada en iyi bilinen kontrolör PID kontrolördür. PID kontrolörün türev ve integral kısmının sisteme ayrı ayrı etkileri vardır [7].

2.1 ORANSAL KONTROLÖR

K sabit kazançlı kontrol sistemleri kontrolör çıkışındaki işaret, girişindeki işarete sabit bir oran ile bağıntılı olduğundan oransal kontrol olarak bilinir. Lineer olmayan sistemlere birim basamak giriş uygulandığında çıkış sabit bir yere yaklaşırsa sistem kararlı kabul edilir ve sistem K_p ile kontrol edilebilir, K_p arttıkça cevap hızı artar, fakat geri beslemeli sistem kararsız hale gelebilir. Oransal kontrolde çıkış hiçbir zaman istenen seviyeye ulaşmaz, kararlı hal hatası oluşur. Kontrol elemanının %100 çıkış vermesi için gerekli olan hatadaki % değişim oransal bant olarak tanımlanır. Oransal bandı küçültmek kontrol sistem kazancını artırmak anlamına gelmektedir [2].

$$u(t) = K_p e(t) \quad \text{veya} \quad U(s) = K_p E(s) \quad K_p = 100 / \% \text{ oransal bant} \quad (2.1)$$

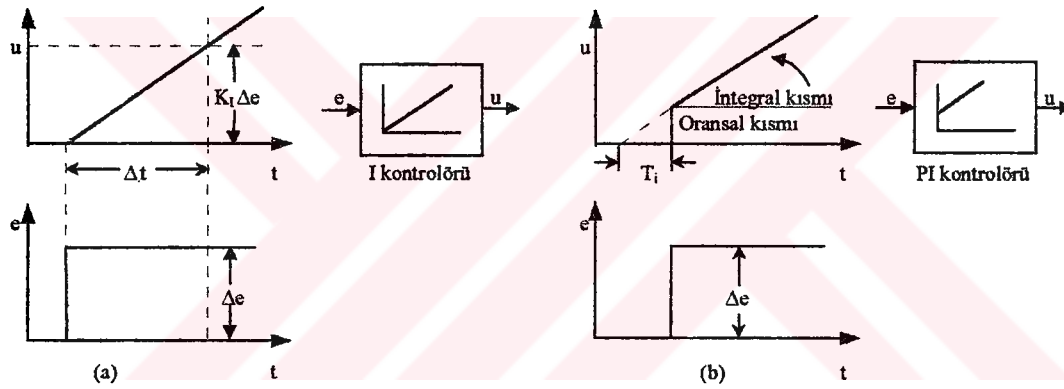


Şekil 2.1 Oransal kontrolörün birim basamak cevabı ve blok diyagramı

2.2 PI KONTROLÖR

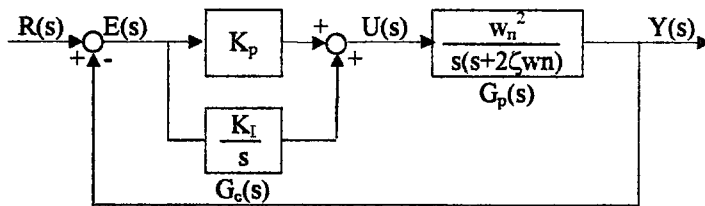
Oransal kontrolde kararlı hal hatası kontrol işaretinin değişmediği denge noktasına erişince oluşur. Kontrolör eğer hata sıfır olmadığı sürece bir artan işaret üretirse, kararlı hal hatası giderilebilir. Bu integral kontrolörün prensibidir. İntegral kontrolör sistemin tipini artırır, fakat geçici cevabı iyileştirecek yönde etkilemez. Bu nedenle oransal kontrolör ile beraber kullanılması yararlı olur. PID kontrolörün integral kısmı kontrolör girişinin zamana göre integrali ile orantılı işaret üretir. Şekil 2.3 PI kontrolörlü ikinci mertebeden bir sistemi gösterir. PI kontrolörün transfer fonksiyonu ise aşağıdaki gibidir [1,7].

$$G_p(s) = K_p + \frac{K_I}{s} \quad (2.2)$$



Şekil 2.2 I ve PI Kontrolörlerin birim basamak cevabı ve blok diyagramı

T_I , integral teriminin oransal terime ulaşması için gereken zaman olarak tanımlanır. Oransal kazanç integral terimi kullanıldığında azaltılmalıdır. Integral terimi kararlı hal hatasına hemen etkilemez, fakat salınımlar yaparak hatayı sıfıra indirir.



Şekil 2.3 PI kontrollü kontrol sistemi

Şekil 2.3'te verilen PI kontrolörlü sistemde açık çevrim transfer fonksiyonu;

$$G(s) = G_c(s).G_p(s) = \frac{w_n^2(K_p s + K_I)}{s^2(s + 2\zeta w_n)} \quad (2.3)$$

şeklindedir. PI kontrolörü ileri yönlü transfer fonksiyonuna $s = K_I/K_p$ 'de bir sıfır $s=0$ 'da bir kutup ekler. Böylece sistem tipi bir artar. Özgün sistemin kararlı hal hatası bir derece düzelir. Örneğin bir giriş için e_{ss} sabit ise PI kontrolör kullanıldığında kararlı hal hatasını sıfıra düşürür. Denklem 2.3'te transfer fonksiyonu verilen sistemde rampa giriş için $e_{ss} = 0$ olur, fakat sistem üçüncü mertebeden olduğundan özgün ikinci mertebeden sisteme göre kararlılığı daha azdır ve K_p , K_I değerleri uygun seçilmezse kararsız hale gelir.

PI kontrollü birinci mertebeden sistemlerde K_p değeri önemlidir, çünkü K_p rampa hatası sabiti, doğrudan K_p ile orantılıdır ve böylece rampa girişi uygulandığında kararlı hal hatasının genliği K_p ile ters orantılı olur. Diğer taraftan K_p çok büyük seçilirse sistem kararsız olabilir. Benzer şekilde sıfır tipi sistemlerde, basamak girişe ait kararlı hal hatası, K_p ile ters orantılıdır. PI kontrolör ile bir tipi sistem iki tipi sisteme dönüştürüldüğünde K_p kararlı hal hatasına etki etmez, rampa giriş için kararlı hal hatası hep sıfır olur. Bundan sonra yapılacak şey geçici cevabın istenen gibi olması için K_p ve K_I 'ya uygun değerler vermektir.

PI kontrolörün zaman tanım bölgesinde incelenmesi : $G_c(s)$ ' in sıfırının yeri uygun seçildiğinde kararlı hal hatasının yanı sıra sönümü de iyileştirilebilir. PI kontrolörler alçak geçiren filtreler olduklarından kompanze edilmiş sistem daha uzun yükselme ve yerleşme zamanına sahiptir. PI kontrolörün tasarım yöntemi; $s = K_I/K_p$ sıfırını diğer baskın kutuplardan uzak ve orijine çok yakın seçmektir. PI kontrolör tasarımında $s = K_I/K_p$ sıfırı $G(s)$ 'in en yakın kutbuna göre çok küçük seçilmelidir. Böylece köklerin yer eğrisinde orijinin yakınlarında küçük bir daire oluşur, fakat genel şekli kompanze edilmemiş sisteme çok benzer, böylece kontrol edilen sistemin karakteristik denklem köklerinin yeri fazla değiştirilmeden kriterlerin uygulanmasını sağlar. Kararlı hal hatası kriterini sağlayacak şekilde K_I parametresi saptanır ve istenen sönüm oranını(ζ) veren K_p seçilir. K_I/K_p sıfıra çok yakın olduğundan transfer fonksiyonunda ihmal edilebilir. Yükselme ve yerleşme zamanını minimum yapmak için K_p baskın kutuplar karmaşık-eşlenik olacak şekilde seçilir.

PI kontrolörün frekans tanım bölgesinde incelenmesi : PI kontrolörün transfer fonksiyonu frekans tanım bölgesinde incelemek için şöyle yazılır.

$$G_c(s) = K_p + \frac{K_I}{s} = \frac{K_I[1+(K_p/K_I)s]}{s} \quad (2.4)$$

İstenen bir faz payını gerçeklemek için PI kontrolör tasarımı şöyledir;

- 1- Kompanze edilmemiş sistemin ileri yol transfer fonksiyonu $G_p(s)$ 'in Bode diyagramı kararlı hal hatasını istenen değere getiren çevrim kazancı hesaba katılarak çizilir.
- 2- Kompanze edilmemiş sistemin faz ve kazanç payı Bode diyagramından bulunur. Belli bir faz payına ulaşmak için bu faz payına ilişkin yeni faz sınırı payı frekansı ω_g ' Bode diyagramından bulunur. İstenen faz payını gerçekleştirilebilmesi için seçilen yeni ω_g ' frekansında kompanze edilmiş transfer fonksiyonun genlik eğrisi x-ekseninde 0 dB'den geçmesi gerekir.
- 3- Kompanze edilmemiş transfer fonksiyonu genlik eğrisinin ω_g ' yeni kazanç payı frekansında 0 dB'e indirilmesi için, PI kontrolörün ω_g ' frekansındaki kazancına eşit miktarda zayıflatma yapması sağlanmalıdır. Bu amaçla K_p değeri aşağıdaki gibi bulunur.

$$|G_p(j\omega_g)|_{dB} = -20 \cdot \log_{10} K_p \text{ dB} \quad K_p < 1 \quad (2.5)$$

$$K_p = 10^{-|G_p(j\omega_g)|_{dB} / 20} \quad K_p < 1 \quad (2.6)$$

K_p değeri saptandıktan sonra tasarımın tamamlanması için uygun bir K_I değerinin seçilmesi kalır. Bu noktaya kadar $G_c(j\omega)$ 'nın genliğinin düşürülmesi ile ω_g faz payı frekansının ω_g olarak değiştirildiği halde, özgün fazın PI kontrolör ile değişmediğini kabul ettik. Bu mümkün değildir, PI kontrolörün zayıflatma özelliği faz payına zarar verecek faz geriletmesi yapar. Eğer $\omega = K_I/K_p$ köşe frekansı ω_g ' nün çok altında yerleştirilirse, ω_g civarında PI kontrolörün faz gecikmesi kompanze edilmiş sistemin fazında ihmal edilebilir düzeyde az etki eder. Diğer taraftan K_I/K_p değeri çok küçük seçilmemelidir aksi taktirde sistemin bant genişliği çok küçük

olacaktır, bu da yükselme ve yerleşme zamanının çok uzun olmasına yol açacaktır. Genel olarak $K_I/K_p = \omega_g / 10$ rad/sn. seçilmelidir.

4- Kompanze edilmiş sistemin Bode diyagramı istenen özelliklerin sağlanıp sağlanmadığını görmek için sınanır.

5- K_p ve K_I değerleri bulunur.

Eğer kontrol edilen sistem $G_p(s)$ sıfır tipinde ise K_I rampa hatasını düzeltmek için seçilir ve geriye sadece K_p parametresinin seçimi kalır. Belli bir K_p değer aralığı için faz payı, kazanç payı, M_r ve kapalı çevrimin bant genişliği hesaplanarak en uygun K_p değeri seçilir.

Yukarıdaki açıklamaların ışığında uygun seçilmiş PI kontrolörünün üstünlük ve sakıncalarını şöyle özetleyebiliriz; maksimum aşımı azaltır, sönümü iyileştirir, yükselme zamanını artırır, BG bant genişliğini azaltır, kazanç payı, faz payı ve M_r 'yi iyileştirir, yüksek frekanslı gürültüleri filtre eder.

2.3 PD KONTROLÖR

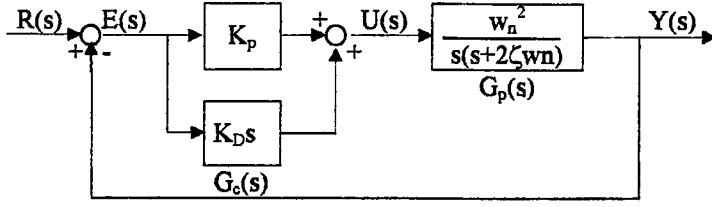
İntegral kısmı hata kalmadığı halde bile kontrol işareti üretmeye devam eder, bunun sonucu olarak sistemde salınımlar oluşur. Salınımların önüne geçmek için kontrolör hatanın sifıra yaklaştığı iletilmelidir, bu hatanın türevi alınarak yapılabilir. Türev ancak oransal kontrolörle beraber kullanılabilir. Şekil 2.4 ikinci mertebeden bir sistemin $G_p(s)$, geri beslemeli kontrol sistemini göstermektedir. Seri kontrolör $G_c(s)$ ise PD kontrolörüdür [7].

$$G_p(s) = \frac{w_n^2}{s(s+2\zeta w_n)} \quad (2.7)$$

$$G_c(s) = K_p + K_D s \quad (2.8)$$

Proses uygulanan kontrol işareti;

$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt} \quad (2.9)$$

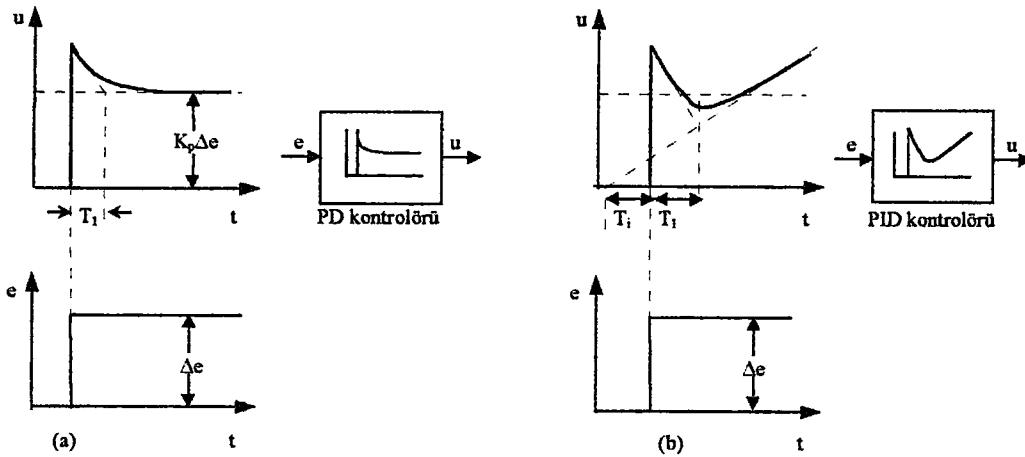


Şekil 2.4 PD kontrolörlü kontrol sistemi

Sistemin açık çevrim transfer fonksiyonu;

$$G(s) = \frac{Y(s)}{E(s)} = G_c(s) \cdot G_p(s) = \frac{w_n^2 (K_p + K_D s)}{s(s + 2\zeta w_n)} \quad (2.10)$$

PD kontrolör zaman tanım bölgesinde düşünülürse; $de(t)/dt$ e(t) hata işaretinin eğimini ifade ettiğinden, aslında sezgisel bir kontroldür. Eğimi bilmesi nedeniyle kontrolör hata yönünü sezer ve sistemi daha iyi kontrol etme olanağına erişir. Lineer sistemlerde basamak cevaba ilişkin $e(t)$ ya da $y(t)$ 'nin eğimi büyük olursa yüksek aşım oluşur. Türevsel kontrol $e(t)$ 'nin ani eğimini ölçer, yüksek bir aşım olacağını tahmin eder ve aşırı aşım meydana gelmeden önce uygun bir düzeltme gayretinde bulunur. PD kontrolör kararlı hal hatası zamanla değişiyorsa sistemin hatasına etkide bulunur, eğer kararlı hal hatası zamanda sabit ise $de(t)/dt = 0$ olur ve kontrolörün türevsel kısmı sisteme herhangi bir girişte bulunmaz. (2.10) bağıntısından da görüleceği üzere bu kontrolör sistem tipini değiştirmez, ancak geçici hal hatasını düzenler. Şekil 2.5'te PI ve PID kontrolörlerin birim basamak cevabı verilmiştir. T_1 teorik olarak sıfır kabul edilmesine rağmen kontrolörün performansına göre değişir. PI ve PID kontrolör çıkışları; $u(t) = K_p \cdot e + K_D \cdot de/dt$, $u(t) = K_p \cdot e + K_D \cdot de/dt + K_I \int e \cdot dt$ şeklindedir.



Şekil 2.5 PD ve PID kontrolörleri birim basamak cevabı ve blok diyagramı

Frekans tanım bölgesi analizinde PD kontrolör Bode diyagramının kolay çizilebilmesi için aşağıdaki şekilde yazılır.

$$G_c(s) = K_p + K_D s = K_p \left(1 + \frac{K_D}{K_p} s \right) \quad (2.11)$$

PD kontrolör yüksek geçiren kontrolördür. Faz ilerletmeli özelliği ile kontrol sisteminin faz payını iyileştirir. Fakat PD kontrolörünün genlik karakteristiği, ω_g faz payı frekansını yüksek değere çeker. K_p / K_D köşe frekansı değiştirilerek sönümü iyileştirecek optimum çözüm bulunur

PD kontrolörü genel olarak sönümü iyileştirir, maksimum aşımı azaltır, yükselme ve yerleşme zamanını düşürür. Bant genişliğini artırır, Kazanç payı, faz payı ve M_r 'yi iyileştirir. Yüksek frekanslarda gürültüyü ön plana çıkarabilir, az sönümlü ve kararsız sistemlerde etkili değildir. PD kontrolör zamanla değişiyorsa kararlı hal hatalarına etki eder.

2.4 PID KONTROLÖR

Önceki incelemelerde PD kontrolörün sisteme sönüm eklediğini fakat kararlı hal hatasını etkilemediğine değinildi. PI kontrolör görece kararlılığı ve aynı zamanda kararlı hal hatasını iyileştirir, fakat yükselme zamanını artırır. Bu durum, PI ve PD kontrolörün iyi özelliklerine sahip olan PID kontrolör kullanımını doğurur. PID kontrolörünü tasarlarken PD kısmının PI kısmı ile seri bağlandığını farz edelim. Bu taktirde;

$$G_c(s) = K_p + K_D s + \frac{K_I}{s} = (1 + K_{D1} s) \left(K_{p2} + \frac{K_{I2}}{s} \right) \quad (2.12)$$

$$K_p = K_{p2} + K_{D1} K_{I2} \quad (2.13)$$

$$K_D = K_{D1} K_{p2} \quad (2.14)$$

$$K_I = K_{I2} \quad (2.15)$$

Üç parametre yeterli olduğundan PD kısmının oransal sabiti “1” seçilmiştir. Tasarımda ilk olarak sadece PD kontrolörünün etki ettiği düşünülür. İstenen görece kararlılığa ulaşmak için K_{D1} değeri seçilir. Zaman tanım bölgesinde görece kararlılık

maksimum aşım frekans tanım bölgesinde ise faz payı ile ölçülür. K_{I2} ve K_{p2} parametreleri diğer gereksinimleri karşılamak üzere seçilir. Başka bir yöntemde ise ilk olarak PI kontrolör ihtiyaca göre tasarlanır, PD kısmı sona bırakılır.

2.5 ZIEGLER – NICHOLS KURALLARI

Bazı sistemlerde transfer fonksiyon modelini saptamadaki zorluk, optimum kontrolör kazanç değerlerini hesaplamanın deneysel yolunu bulmaya itmiştir. En çok kullanılan yöntem Ziegler-Nichols yöntemi, iki ayrı yöntemden oluşur. İlk yöntem sistemin açık çevrim basamak cevabına, ikincisi kontrolör ile sistemin deneysel olarak incelenmesi ve sonuçlarına ihtiyaç duyar. Sistemin modeli geliştirilse dahi bu yöntemi kullanmak oldukça yararlıdır. Ziegler-Nichols, birim basamak cevap ile IAE kriterini kullanarak, kurallarına göre tasarlanmış kontrolörlerin salınımlı ancak ikinci aşım birinci aşımın %25'inden küçük olacak şekilde, yeterli derecede sönümlü yanıt verdiğini belirlemişlerdir. Buna çeyrek söndürme kriteri denir [2].

Uygulanan kriter;

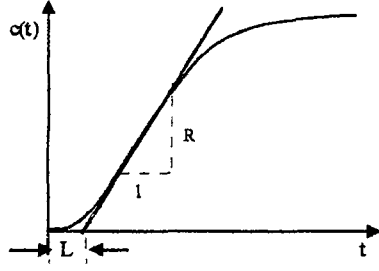
$$J = \int_0^{\infty} |e(t)| dt = \int_0^{\infty} |r(t) - y(t)| dt \quad (2.16)$$

şeklindedir.

İlk yöntem proses yanıt yöntemidir. Bir çok proses şekil 2.6'daki gibi açık çevrim birim basamak cevabına sahiptir. R, cevap eğrisinin en yüksek eğimi, L ise bu maksimum eğimli eğrinin zaman eksenini kestiği noktadır. Birinci ve ikinci mertebeden lineer sistemler pozitif L değeri vermezler, dolayısıyla bu yöntem uygulanamaz. Üçüncü ve daha yüksek mertebeden yeterli sönümlü lineer sistemler için Tablo 2.1'de verilen değerler kullanılabilir.

Kapalı çevrim (ultimate cycle) yöntemi, kontrolörlü proseslere uygulanır. Oransal hariç tüm kontrol tipleri sıfırlanır ve oransal kazanç en düşükten başlayarak sistem sönümsüz osilasyon yapana kadar artırılır, elde edilen kazanç K_{pu} , periyoda P_u denilirse tablo 2.1'de bu parametrelere göre Ziegler-Nichols tavsiyeleri verilmiştir. Oransal kazanç PI kontrolörde P kontrolördekinden az, PID kontrolde çok seçilmiştir. Çünkü kontrolörün integral kısmı sistemin tipini artırır ve kararsızlığa sürükleyebilir, bu nedenle düşük kazanç gereklidir. Öte yandan kontrolörün türev

kısmı sistemi kararlılığa götürdüğünden kazanç büyük seçilebilir. Bu seçimlerden sonra mutlaka ince ayar yapılmalıdır.



Şekil 2.6 Ziegler Nichols için açık çevrim cevabı

Tablo 2.1 Ziegler-Nichols kuralları

Kontrolör tipi	Açık çevrim yöntemi	Kapalı çevrim yöntemi
P kontrol	$K_p=1/RL$	$K_p=0.5K_{pu}$
PI kontrol	$K_p=0.9/RL$ $T_i=3.3L$	$K_p=0.45K_{pu}$ $T_i=0.83P_u$
PID kontrol	$K_p=1.2/RL$ $T_i=2L$ $T_D=0.5 L$	$K_p=0.6K_{pu}$ $T_i=0.5P_u$ $T_D=0.125P_u$

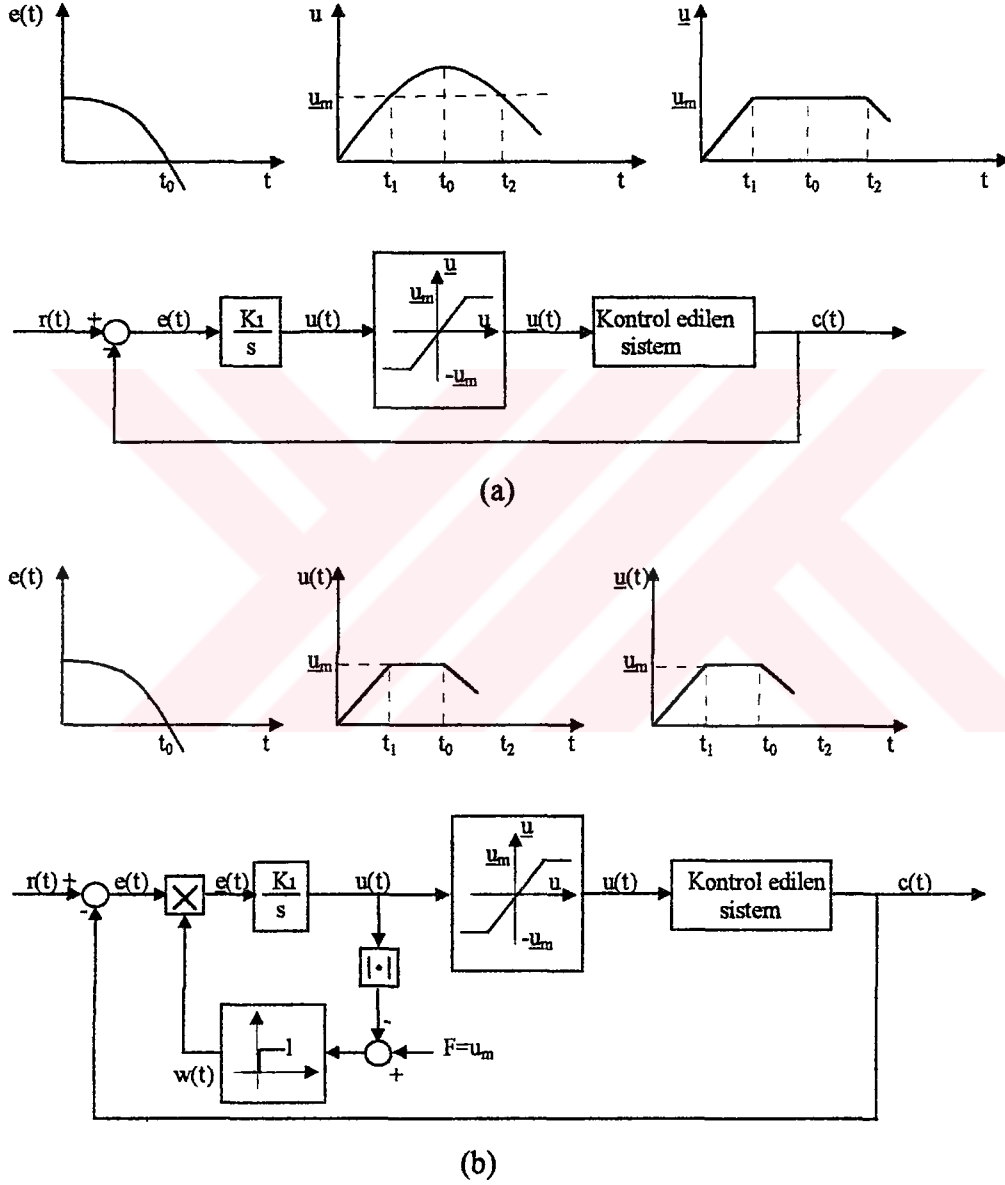
2.6 İNTEGRAL YIĞILMASI

İntegral kontrolör giriş işaretinin integralini alır. $u(t)=K_I \int e(t)dt$. Birçok hareketli sistem ve sürücü elemanda doyum bulunur. Örneğin şekil 2.7.a'daki gibi u_m doyumlu valf ve integral kontrollü bir sistemin bulunduğunu ve $e(t)$ hata işaretinin de şekildeki gibi olduğunu varsayalım. İlgili kontrolör çıkışı ve valf çıkışı şekilde gösterilmiştir [2].

İdealde, hata t_0 anında pozitiften negatife geçtiğinde aynı etkinin yine t_0 anında $u(t)$ 'de de belirmesi gerekir. Fakat integrasyon nedeniyle, $u(t)$ $t=t_0$ anında u_m değerinin üstündedir ve u_m değerine ulaşması $t=t_2$ anında olur. Bu nedenle, hatanın $u(t)$ 'deki etkisi ancak t_2-t_0 kadar bir gecikmeden sonra t_2 anında belirir. Sonuç olarak integrasyondan dolayı $u(t)$ işareti doyma seviyesinin üzerine çıkar ve $u(t)$ 'ye etki etmesi içinde bu seviyenin altına inmesi gereklidir. Bu olaya integrasyon yığılması adı verilir. $e(t)$ hatası işaret değiştirmedeği sürece $u(t)$ artmaya devam ettiğinden $c(t)$ kontrol edilen büyüklüğün cevap eğrisinde büyük aşımalar oluşur.

Eğer $e(t)$ hatası küçük ve hemen sıfıra yaklaşıyorsa, integral yığılması oluşmayabilir. İntegral yığılmasının geribeslemeli sistemde oluşması istenmez. Bunu önlemenin bir yolu; $u(t)$ doymaya ulaştığında integral alıcısı devreden çıkarmaktır.

Şekil 2.7.b’de integral alıcının girişi $e(t) = e(t) \cdot w(t)$ ’dir. Burada $|u(t)| \leq u_m$ iken $w(t)=1$, $|u(t)| > u_m$ iken $w(t)=0$ ’dır. Bilgisayar ile kontrolde bu sistem yazılım ile sağlanır. İntegral alıcının çıkışı doyma seviyesine ulaştığında devreden çıkarılır, böylece $u(t)$ doyma seviyesinin üstüne çıkmayacaktır. Dolayısıyla t_0 anındaki hata işareti değişimi anında $u(t)$ ’ye etki etmesi sağlanmış ve integral yığılması önlenmiş olur.



Şekil 2.7 İntegral Yığılması

BÖLÜM 3

DİJİTAL PID KONTROLÖR

Dijital kontrol sistem tasarımı sürekli kontrol sistem tasarımı ile prensipte benzeşir. Kontrol edilen sistem genelde aynıdır sadece kontrolör örneklenmiş prosese yada dijital bilgiye göre tasarlanır. Bu durumda da sürekli kontrol sisteminde olduğu gibi zaman ve frekans tanım bölgesinde inceleme yapılabilir. S-tanım bölgesindeki tasarım kriterleri z-tanım bölgesine uyarlanabilir, Tasarım bilgisayar programları ile en az sınımayla gerçekleştirilebilir [7] .

Bugün parametre seçimi yada kontrolör gerçekleştirilmesi sadece yazılım ile oluşturulan dijital kontrolörlerin kullanımı yaygın hale gelmiştir. Dijital kontrolör dinamiği z-transfer fonksiyonu yada fark denklemleri ile tanımlanabilir. Sürekli sistemlerde PID kontrolörleri (2.12) denklemi ile ifade edilir. K_p oransal elemanı K_p sabit kazanç olarak dijitale çevrilir. Dijital bilgisayar ve işlemciler sonlu kelime uzunluğuna sahip olduklarından K_p sonsuz değerini alamaz [7] .

$f(t)$ fonksiyonunun $t=kT$ anında zamana göre türevi, geri dikdörtgen fark kuralı ile, $t=kT$ ve $t=(k-1)T$ zamanında ölçülen değerleri kullanılarak

$$\left. \frac{df(t)}{dt} \right|_{t=kT} = \frac{1}{T} \left[f(kT) - f[(k-1)T] \right] \quad (3.1)$$

şeklinde yazılabilir.

z-transfer fonksiyonunu bulmak için her iki tarafın z dönüşümü alınır:

$$Z \left(\left. \frac{df(t)}{dt} \right|_{t=kT} \right) = \frac{1}{T} (1-z^{-1})F(z) = \frac{z-1}{Tz} F(z) \quad (3.2)$$

Buna göre dijital türev alıcının z-transfer fonksiyonu;

$$G_D(z) = K_D \frac{z-1}{Tz} \quad (3.3)$$

şeklindedir. K_D türev kontrolörünün oransal sabitini ifade eder. z yerine e^{sT} yerleştirilirse örnekleme periyodu sıfıra yaklaşırken $G_D(z)$ 'in sürekli türevsel kontrol transfer fonksiyonu olan K_D 'e yaklaştığı görülebilir. Örnekleme periyodunun seçimi çok önemlidir. Sürekli bilgi dijital bilgiye çevrilirken çok bilgi kaybı olmayacak şekilde T değeri yeteri kadar küçük seçilmelidir. Örnekleme zamanının oldukça küçük seçilmesi sistem davranışı bakımından mükemmeldir, fakat bu durumda dijital bilgisayarın işlem sayısı artar, daha iyi donanıma sahip olması gerekebilir. Bu yüzden optimum örnekleme zamanı belirlenmelidir.

İntegral kontrolörün dijitale dönüştürecek birkaç sayısal integrasyon kuralı vardır. Bunlar yamuk, ileri ve geri dikdörtgen integrasyondur. Aşağıda sırasıyla yamuk, ileri ve geri dikdörtgen integrasyonu ile elde edilen transfer fonksiyonları sırasıyla denklem (3.5), (3.7) ve (3.9)'da verilmiştir.

$$u(kT) = u(k-1)T + \frac{T}{2} [f(kT) + f[(k-1)T]] \quad (3.4)$$

$$G_I(z) = K_I \frac{U(z)}{F(z)} = \frac{K_I T(z+1)}{2(z-1)} \quad (3.5)$$

$$u(kT) = u[(k-1)T] + T f(kT) \quad (3.6)$$

$$G_I(z) = K_I \frac{U(z)}{F(z)} = \frac{K_I Tz}{z-1} \quad (3.7)$$

$$u(kT) = u[(k-1)T] + T f[(k-1)T] \quad (3.8)$$

$$G_I(z) = K_I \frac{U(z)}{F(z)} = \frac{K_I T}{z-1} \quad (3.9)$$

Yukarıda değinilen oransal, türevsel ve integral işlemler birleştirilerek dijital PID kontrolör modellenebilir, transfer fonksiyonu sırasıyla yamuk, ileri ve geri dikdörtgen integrasyonu için üç ayrı şekilde verilmiştir.

$$G_c(z) = \frac{(K_p + T K_I/2 + K_D/T)z^2 + (T K_I/2 - K_p - 2K_D/T)z + K_D/T}{z(z-1)} \quad (3.10)$$

$$G_c(z) = \frac{(K_p + K_D/T + T K_I)z^2 - (K_p + 2K_D/T)z + K_D/T}{z(z-1)} \quad (3.11)$$

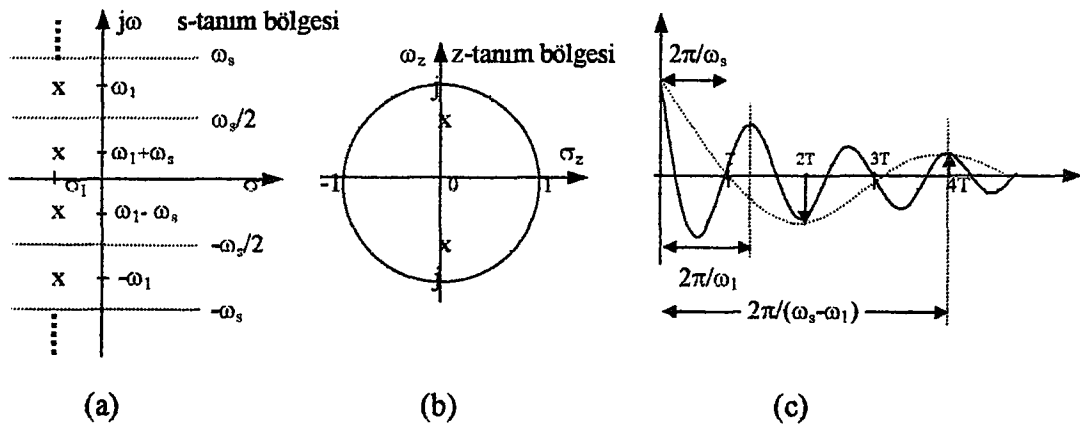
$$G_c(z) = \frac{(K_p + K_D/T)z^2 + (T K_I - K_p - 2K_D/T)z + K_D/T}{z(z-1)} \quad (3.12)$$

3.1 Z-DÜZLEMİ TASARIM KRİTERLERİ

Ayrık sistem sürekli sistemden farklı olarak, cevap işaretini belli örnekleme zamanlarında ölçer ve her örnekleme zamanında kontrol işareti üretir. Bu zaman içinde işaretin değişmediği varsayılır. Örnekleme periyodu ve sistemin zaman sabiti-ne bağlı olarak ayrık sistemin doğruluğu değişir. Diğer bir deyişle gerçek $c(t)$ çıkışı ile örneklenmiş sinyal $c^*(t)$ arasında büyük farklılık oluşur. Daha önce sürekli

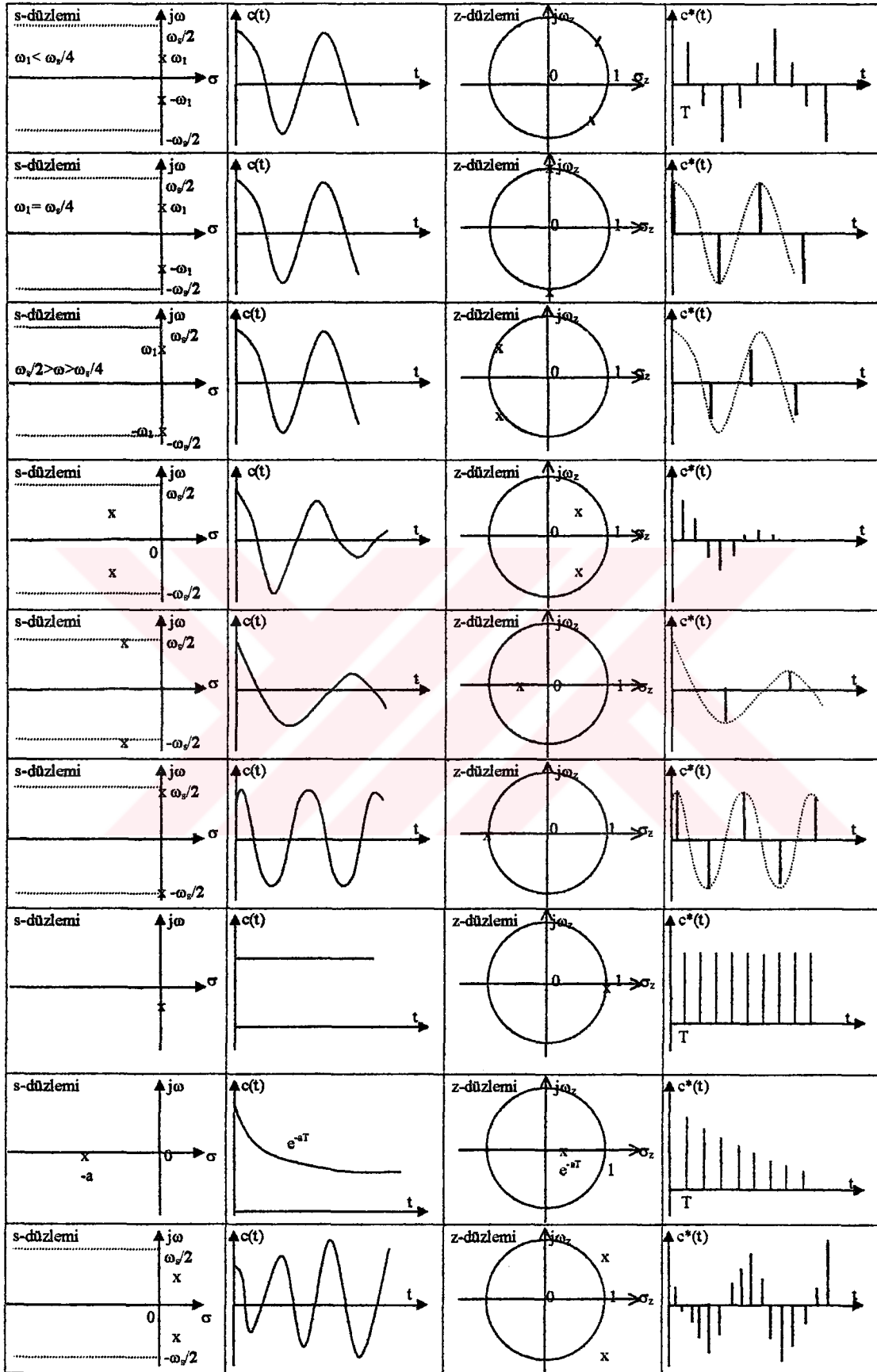
oranı ve doğal frekans gibi tasarım kriterleri dijital kontrolörlerde de aynıdır [10] .

Sürekli sistemlerde, K kazancını artırmak sönüm oranını, kararlı hal hatasını düşürür, maksimum aşımı, ω_n doğal frekansı, band genişliğini artırır. Ayrık zamanda ise sabit K değeri için T'yi artırmak sistemi kararsızlığa sürükler, çok küçük örnekleme zamanı ise dijital bilgisayarın daha hızlı çalışması, daha fazla hesap yapması ve daha kompleks donanıma sahip olmasını gerektirir. Örnekleme frekansı $\omega_s < 2\omega_1$ seçilirse, örnekleme işlemi s düzleminde $s = \sigma_1 \pm j\omega_1 + jn\omega_s$ ($n = \pm 1, \pm 2, \dots$) olmak üzere sonsuz sayıda kutup üretir. Şekil 3.1'de gösterildiği gibi örnekleme işlemi kutupları $\omega_s/2 < \omega < \pm\omega/2$ aralığı içine sokar, öyle ki $s = \sigma_1 \pm j(\omega_s - \omega_1)$ kutbu olan orijinal bir sistem ile aynı etkiyi gösterir. T örnekleme periyodunu artırmak ω_s örnekleme frekansını artırmak anlamına geldiğinden, T ne kadar küçültülürse s-düzleminin sol yarı eğrisinin o kadar fazla kısmı z-düzlemine intikal ettirilir ve sistem daha kararlı olur. Tablo 3.1 'de s ve z düzlemindeki kutup yerleşimine ilişkin cevap eğrileri görülebilir [4].



Şekil 3.1 Örnekleme frekansının s ve z-tanım bölgesindeki etkisi

Tablo 3.1 Çeşitli s ve z düzlemi kök yerleri ve birim basamak cevapları



Ayrık sistemde; önemli performans kriterlerinden olan kararlı hal hatasını birim basamak giriş durumunda hesaplamak için son değer teoremi kullanılırsa [10] ;

$$E(z) = \frac{R(z)}{1+D(z)G(z)} = \frac{z}{z-1} \cdot \frac{1}{1+D(z)G(z)} \quad (3.13)$$

$$e(\infty) = \lim_{z \rightarrow 1} (z-1) \frac{z}{z-1} \frac{1}{1+D(z)G(z)} = \frac{1}{1+D(1)G(1)} = \frac{1}{1+K_p} \quad (3.14)$$

“0” tipi sistem için konum sabiti

$$K_p = D(1)G(1) \quad (3.15)$$

olarak elde edilir. Sisteme rampa giriş uygulandığı durumda;

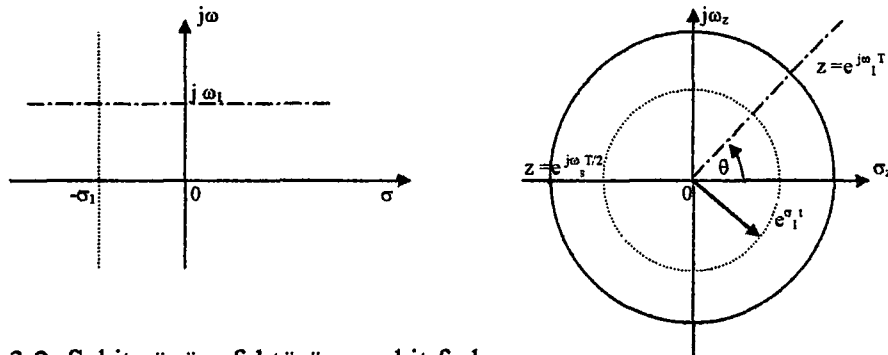
$$R(z) = Tz/(z-1)^2$$

$$e(\infty) = \lim_{z \rightarrow 1} (z-1)E(z) = \frac{1}{K_v} \quad (3.16)$$

$$K_v = \lim_{z \rightarrow 1} \frac{(z-1)(1+D(z)G(z))}{Tz} \quad (3.17)$$

Kararlı hal hatası kapalı çevrim kutupları $z=1$ 'e yaklaşırken, kapalı çevrim sıfırları $z=0$ 'a yaklaşırken azalır.

Kontrolör tasarımında görelî kararlılık problemi karakteristik denklem kökleri yerinin sabit sönüm faktörü ve oranına göre incelenmesinden ibarettir. s-düzlemindeki sabit sönüm faktörü ($s=\sigma_1$) eğrisi z düzleminde $z=0$ merkezli $e^{\sigma_1 T}$ yarıçaplı çembere karşılık gelir. Yine s-düzlemindeki $\omega = \omega_1$ yatay eğrisi $z=e^{j\omega_1 T}$ olmak üzere aşım $\theta = \omega_1 T$ radyan olan bir fazördür. Eğer sönüm faktörü maksimum değeri (σ_1) isteniyorsa minimum zaman sabiti $1/\sigma_1$ demektir. z-düzleminde karakteristik denklemin köklerinin $|z| = e^{\sigma_1 T}$ ile birim daire arasında olması gereklidir [4] .

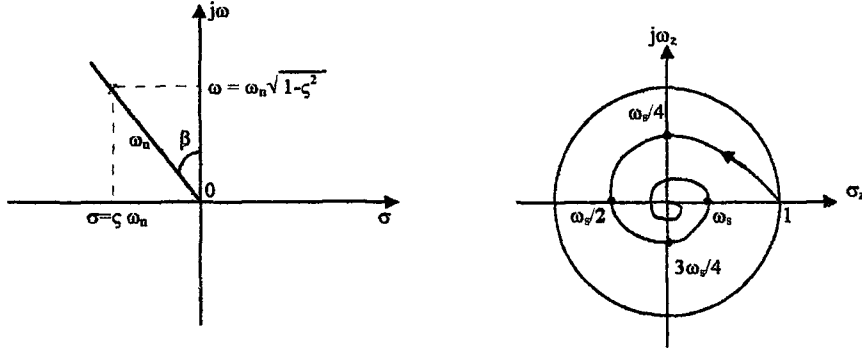


Şekil 3.2 Sabit sönüm faktörü ve sabit frekans

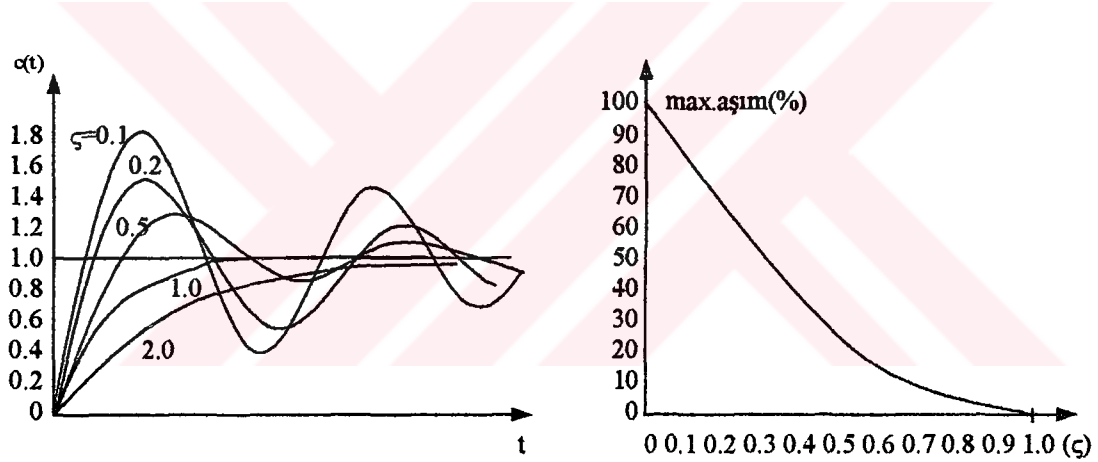
Kontrol sistem tasarımı genellikle sistem transfer fonksiyonu baskın kutuplarının sönüm oranı ($\zeta=0.707$) olarak istenir. s-düzlemindeki sabit sönüm oranı z-düzleminde bir logaritmik spiral oluşturur.

$$s = -\omega \tan \beta + j\omega \quad \beta = \sin^{-1} \zeta \quad T = 2\pi/\omega_s \quad (3.18)$$

$$z = e^{Ts} = e^{T(-\omega \tan \beta + j\omega)} = e^{-2\pi \omega \tan \beta / \omega_s} \angle 2\pi \omega / \omega_s \quad (3.19)$$



Şekil 3.3 Sabit sönüm oranı

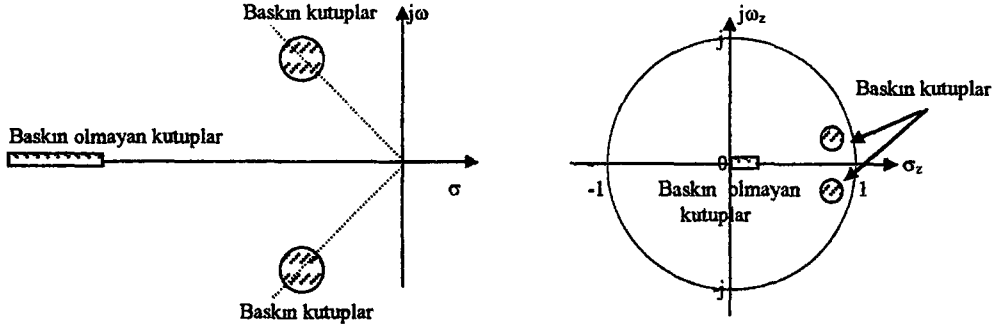


Şekil 3.4 Değişik aşım değerlerinde basamak yanıt eğrisi ve % aşım- (ζ) ilişkisi

İkinci mertebeden sistemler uygulamada çok seyrek görülmesine rağmen bir çok davranış parametreleri, yüksek mertebeden sistemlerin analizi zor olduğundan, ikinci mertebeden sistemlere göre tanımlanırlar. Yüksek mertebeden sistemlerdeki kutupların bazıları geçici cevaba etkileri ihmal edilecek şekilde yerleştirilirse bu sistemler ikinci mertebeden sistem gibi düşünülebilir. Şekil 3.5'te s ve z tanım bölgesinde baskın olan ve olmayan kutuplarının yerleri gösterilmiştir.

s-düzleminde baskın kutuplar $\zeta=0.707$ doğrusu üzerindedir. z-düzleminde ise birim daire içinde $z=0$ civarlarındadır. z-düzlemindeki baskın olmayan kutuplar

sıfırın yakınındadır, çünkü $s \rightarrow \infty$ için $z \rightarrow 0$ 'dır. Negatif reel eksenenden kaçınılmalıdır, çünkü salınımına neden olacaktır.



Şekil 3.5 s ve z-düzlemindeki baskın ve baskın olmayan kutuplar

s-tanım bölgesinde baskın olmayan kutuplar geçici cevapta ihmal edilebilir. z-tanım bölgesinde ise bu kökler sadece maksimum aşım ve sönümde ihmal edilir, ancak kapalı çevrim transfer fonksiyonunun $z=0$ civarında kutup sayısının fazlalığı cevap eğrisinde zaman gecikmesine yol açar. Örneğin kararlı bir sistemin transfer fonksiyonuna $z=0$ 'da iki kutup eklenirse maksimum aşım olmaz, fakat cevap eğrisine $2T$ kadar ek bir gecikme gelir. Yüksek mertebeden sistemleri basitleştirmek için bu baskın olmayan kutupları kaldırmak yerine bunları $z=0$ 'da göstermek uygundur.

z-düzleminde birim daire içindeki karmaşık kökler, cevap eğrisinin zamanla sönen salınımlı olmasına neden olur. Genelde karmaşık kökler birim daireye yaklaşırsa, salınım daha az sönümlü olur. Bu olay kapalı çevrim transfer fonksiyonunun kutup-sıfır yerleşimi ile maksimum aşım ve $T_{\max}$ arasındaki ilişkiyi saptamak bakımından yararlı olabilir.

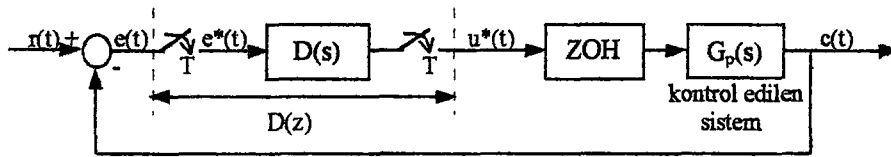
$$\text{Maksimum aşım} = 1 + \exp(-\zeta\pi/\sqrt{1-\zeta^2}) \quad (3.20)$$

$$T_{\max} = \pi / (\omega_n \sqrt{1-\zeta^2}) \quad (3.21)$$

İkinci mertebeden yüksek sistemlerde yukarıdaki ilişkiyi kurmak zordur. Fakat sistem bir çift baskın karmaşık kutup ve diğerleri baskın olmayan kutuplar olarak düzenlenirse bu kutupların etkileri ihmal edilebileceğinden inceleme kolaylaşır.

3.2 DİJİTAL KONTROL SİSTEMİ

Dijital kontrol sistemi tasarımı sürekli sistem tasarımı ile aynı şekilde olduğu söylenebilir. Çıkışın belli performans kriterlerine göre davranması için proses kontrolü yapılır. Dijital sistemde birkaç yapı mevcuttur. Geri besleme bilgisi dijital çevirici ile alınıp dijital kontrolöre uygulanabilir; diğer taraftan dijital işaret ZOH sıfır dereceden tutucu ile sürekli kontrolöre de uygulanabilir. Sürekli veya dijital kontrolörler negatif geri besleme yolunda da kullanılabilirler. Bu tezde PLC ile gerçekleştirilen PID kontrolör; seri kontrolörlü geleneksel dijital kontrol sistemleri gibi kullanılmamıştır[4].



Şekil 3.6 Seri dijital kontrolörlü dijital kontrol sistemi

Oransal, integral ve türev kontrolörlerin birleşiminden oluşan PID kontrolörün sırasıyla yamuk, ileri dikdörtgen ve geri dikdörtgen integrasyonu ile dijital olarak gerçekleştirilmesi denklem (3.10,11,12)'deki gibidir.

3.2.1 DİJİTAL PD KONTROLÖR

Tasarım gereksinimlerine bağlı olarak çoğunlukla PD kontrolör kullanılır. Kontrol edilen sistemin "1" tipi olduğu durumlarda birim basamak giriş için sürekli hal hatası sıfır olacağından $z=1$ 'de kutup koymaya gerek yoktur. PD kontrolör yeterli olabilir. $K_I = 0$ seçilerek PD kontrolör elde edilir. PD kontrolörün $z=0$ 'da kutbu ve pozitif gerçel eksende birim daire içinde $z = K_D / (K_p T + K_D)$ 'da sıfırı vardır [4] .

$$D(z) = \frac{(K_p T + K_D) z - K_D}{Tz} \quad (3.22)$$

PD kontrolör faz ilerletmeli kontrolörün kutbun $p=0$ 'da bulunması özel halidir. Dijital kontrolörün sistemin kararlı hal performansına etkide bulunmaması için K_p ,

$$\lim_{z \rightarrow 1} D(z) = 1 \quad (3.23)$$

koşulu sağlanacak şekilde seçilebilir. Daha sonra K_D değeri geçici cevabı iyileştirmek için uygun değere ayarlanır. Faz ilerletmeli kontrolörde z_1 ve p_1 birim daire içinde seçilir ve genellikle K_p değeri aşağıdaki gibi seçilir.

$$D(z) = K_p \frac{z - z_1}{z - p_1} \quad |z_1| > |p_1| \quad (3.24)$$

$$K_p = \frac{1 - p_1}{1 - z_1} \quad (3.25)$$

3.2.2 DİJİTAL PI KONTROLÖR

Sistem “0” tipinde ise kararlı hal hatasını sıfırlamak için PI kontrolör kullanmak uygundur. Eğer iyi tasarlanırsa sistem tipini artırarak kararlı hal performansını iyileştirir hem de görelî kararlılığı artırır. PI kontrolör faz geriletmeli kontrolörün özel bir şeklidir, $z=1$ ’de kutbu ve $z = (K_p - K_I T) / K_p$ ’de sıfırı vardır. $z=1$ seçilmesiyle $K_v = \infty$ olur ve kararlı halde K_p ’nin önemi kalmaz böylece K_p ’nin seçimi ile görelî kararlılık düzeltilir. Geri ve ileri dikdörtgen ve yamuk iterasyonuna göre ayrık PI kontrolör aşağıdaki gibidir [4].

$$D(z) = \frac{z - 1}{z - 1} \quad (3.26)$$

$$D(z) = \frac{(K_p + K_I T) z - K_p}{z - 1} \quad (3.27)$$

$$D(z) = \frac{(2K_p + K_I T)z + (K_I T - 2K_p)}{2(z - 1)} \quad (3.28)$$

Faz geriletici kontrolör transfer fonksiyonu ;

$$D(z) = K_p \frac{z - z_1}{z - p_1} \quad |z_1| < |p_1| \quad (3.29)$$

z_1 ve p_1 ’in her ikisi de $z=1$ ’e çok yakın seçilirse buna “faz geriletmeli dipol kontrolör” denir. Faz geriletmeli kontrolörler kontrol çevrim kazancı K_p ’yi azaltırlar. Faz geriletmeli dipol kontrolör ile kompanze edilmiş sistemin açık çevrim transfer fonksiyonu $D(z) G_{ho} G_p(z) \cong K_p G_{ho} G_p(z)$ gibi yaklaşık gösterilebilir. Dipol $z=1$ civarında küçük bir çevrim oluşturur. Böylece kompanze edilmiş sistemin kök eğrisi

kompanze edilmiş sistemin kazanı büyük çıkar ki, istenen de budur. Faz geriletici dipol kontrolör tasarımı izlenecek yol;

- Hata kriterinden K değeri seçilir. Bu K değerine K_{eski} denir.
- Kompanze edilmemiş sistemin kök eğrisi çizilir.
- Kök eğrisinden görelî kararlılık kriterini sağlayan K değeri bulunur. Bu değer K_{yeni} olarak tanımlanır.
- Kontrolör kazancı $K_p = K_{yeni} / K_{eski}$ olarak bulunur.
- $z_1=1$ 'e yakın seçilir.
- $p_1=1 - K_p (1 - z_1)$ olarak bulunur.

PI kontrolörde de bazı durumlara kontrolör sıfırı $z=1$ 'e çok yakın seçilerek dipol özelliği sağlanır.

3.2.3 KUTUP-SIFIR SİLİNMESİ

S ve z-düzleminde kontrol sistem tasarımı yaygın yöntem kontrol edilen sistemin istenmeyen kutup ve sıfırlarının kontrolör kutup ve sıfırları ile silinerek yok edilmesidir. Yeni açık çevrim kutup ve sıfırları daha avantajlı yerlerde eklenir ve tasarım kriterleri gerçekleştirilir. Daha çok kontrol edilen değişkenin karmaşık kutupları yok edilir, en kolay kontrol stratejisidir. Fakat her tatmin edice sonuca ulaşamaz. Eğer istenmeyen kutuplar birim daireye çok yakınsa kutup-sıfır silinmesi tam gerçekleşmemesi halinde sistem kararsızlığa sürüklenebilir. Birim daire içinde olmak koşuluyla, kutup ve sıfır tam birbirini üzerinde olmasa dahi kapalı çevrim kutbu sıfıra çok yakın olacağından geçici cevaba etkisi ihmal edilebilir düzeyde olacaktır [4].

3.2.4 DİJİTAL PID KONTROLÖR

Dijital PID kontrolör z düzlemi kök eğrisine $z=0$ ve $z=1$ 'de iki adet kutup ve bu iki kutbun ortasında birim dairenin sağ tarafında iki adet sıfır ekler. Küçük K değerlerinde kapalı çevrim kutbu $z=1$ 'e çok yakın olduğundan aşım çok büyük olur.

İlkin PI kontrolör tasarımı yapıyor gibi istenen K_v hata sabiti değerinden, K_I hesaplanır, sıfırların yerleri belirlenir ve istenen kapalı çevrim kutuplarına ulaşmak için uygun K_p ve K_D değerleri bulunur. Dijital PID kontrolörün transfer fonksiyonu denklem (3.10,11,12)'deki gibidir.



BÖLÜM 4

PLC'LERİN İÇ YAPISI

Tüm PLC'lerin temel yapısı ve işletim sistemi genel olarak birbirinin aynıdır. Bu bölümde Simatic S7 200 PLC'sinin bellek yapısı, program işletimi gibi konular ele alınacaktır.

4.1 S7-200 PLC'LERİNE GENEL BAKIŞ

S7 200 PLC ailesi CPU 212 ve CPU 214 olmak üzere iki çeşittir, merkezi işlemci ünitesi, mevcut giriş ve çıkışları ile kompakt bir PLC'dir. Bunların özellikleri;

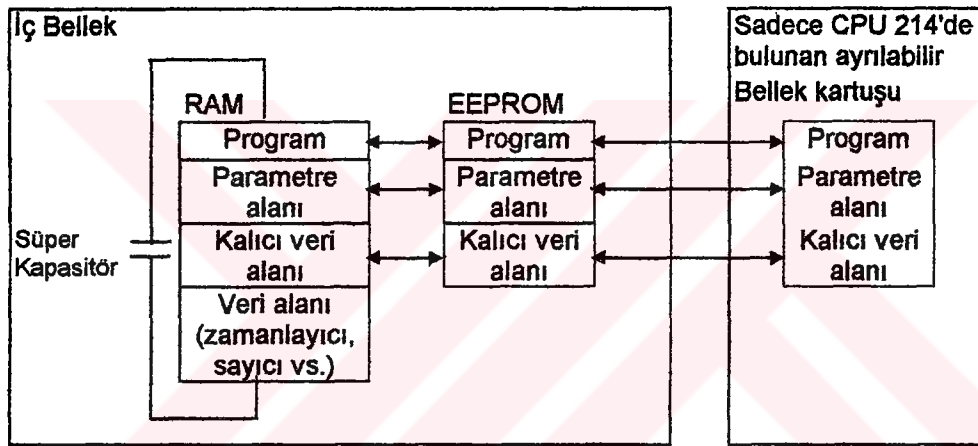
Tablo 4.1 Simatic S7 212 / 214 özellikleri

	CPU 212	CPU 214
Program belleği	512 kelime	2048 kelime
Veri belleği	512 kelime	2048 kelime
CPU üzerindeki G/Ç sayısı	8 giriş / 6 çıkış	14 giriş / 10 çıkış
Zamanlayıcı	64 (2 adet 1ms. 8 adet 10ms. 54 adet 100ms.)	128(4adet 1ms. 16adet 10ms. 108 adet 100ms.)
Sayıcı	64(48 yukarı sayıcı, 16 yukarı/aşağı sayıcı)	128 (96 yukarı sayıcı, 32 yukarı/aşağı sayıcı)
İç bellek biti	128	256
Özel bellek biti	368	688
Dört matematik fonksiyonu	evet	evet
Kesme kapasitesi	1 yükselen/düşen kenar, 1 zamana bağlı kesme, 1 yük.hızlı sayıcı kesmesi	4 yükselen/düşen kenar, 2 zamana bağlı kesme, 7 yüksek hızlı sayıcı kesmesi, 2 darbe dizisi kesmesi
Yüksek hızlı sayıcı	1(2 kHz.)	1 (2 kHz.) 2 (7 kHz.)
Enerji kesildiğinde veri belleği destekleme süresi	50 saat	190 saat
Komut işletim süresi	1.3 ms.	0.8 ms.

CPU 212 ve CPU 214 PLC'lerinin DC çıkış, AC çıkış ve röle çıkışlı olmak üzere üç modeli de mevcuttur. CPU 212 küçük çaplı uygulamalarda kullanılabilir ve iki adet genişleme modülü takılmasına izin verebilir. CPU 214'ün ise daha büyük çaplı uygulamalarda kullanılması mümkündür ve yedi adet genişleme modülü takılmasına izin verebilir. CPU 214'de seçimsel bellek kartuşu, 2 analog potansiyometre ayarı (PLC içinde dijital değere dönüştürülür) ve gerçek zaman saati mevcuttur.

4.2 S7-200 PLC BELLEK TİPLERİ

S7-200 PLC'sinde bellek üç alana ayrılır; program belleği, veri belleği, parametre belleği. Şekil 4.1'de S7-200 PLC'sinin belleğinin sembolik şekli verilmiştir. Bunları ayrıntılı olarak incelersek;



Şekil 4.1 S7 200 Belleği

PLC'lerde genelde programın, verilerin, sabitlerin vs. bilgilerin saklanması için RAM, EPROM, EEPROM entegre devreleri kullanılmaktadır. RAM (Random Access Memory), kolaylıkla programlanabilen, içeriği değiştirilebilen veya silinebilen bellek ortamıdır. RAM'in içeriği enerji ile desteklenmediği zaman silinir. Bu yüzden genelde RAM'e destek sağlayacak bir pil kullanılır. EPROM (Erasable Programmable Read Only Memory), programlandıktan sonra sadece okunabilen ve pil gerektirmeyen bellek saklama ortamıdır. Bu nedenle, değişken olmayan kullanıcı programlarının saklanmasında yararlanır. EPROM içeriğini silmek için ultraviyole ışınması altında bir süre bekletmek gerekir. EEPROM (Electrically Erasable Programmable Read Only Memory), silinmesi için ultraviyole ışın yerine DC gerilim yeterli olan EPROM tipidir.

4.2.1 PROGRAM BELLEĞİ

İstenilen kontrol sistemini gerçeklemek için gerekli olan komut listesini içerir. Program aynı zamanda yazılabilir ve okunabilir olan kalıcı bellekte de saklanır. PLC'nin enerjisi kesildiğinde program kaybolmaz, çünkü pil yada herhangi bir güç kaynağına gerek kalmaksızın program kalıcı bellekte saklı kalır. Simatic sistemlerinde her bir komut bellekte 2 veya 4 byte yer kaplar. Mikroişlemci bellekteki programın komutlarını sırayla birer birer işler ve işlemesi gereken komutu (genellikle 2 byte sonraki komut) akılda tutmak için bir sayıcı kullanır.

Program belleği, LAD merdiven diyagramı yada STL deyim listesini içinde bulundurur. Programlama cihazı ile program belleği içeriği değiştirilebilir. Program belleği OB1 (Organizasyon Belleği 1) olarak tanımlanır ve iç kalıcı bellekte saklanır. CPU 214'te takip çıkarılabilen OB1 ve sürekli saklanması gerekli bilgilerin yer aldığı seçimsel bellek kartuşu mevcuttur. Program belleğinin büyüklüğü CPU 212'de 512 kelime, CPU 214'de 2048 kelime'dir.

4.2.2 VERİ BELLEĞİ

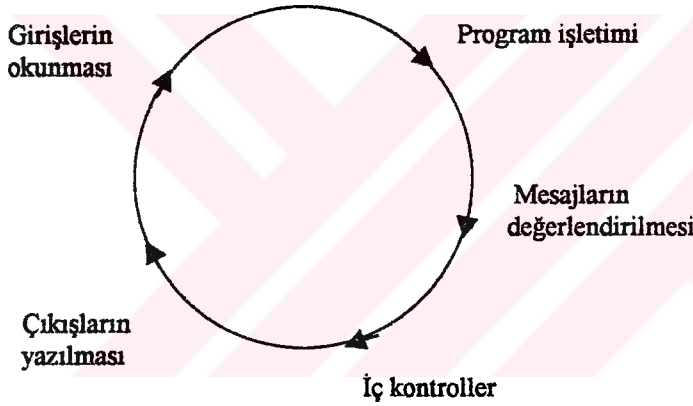
Kontrol programının işletilmesi sırasında PLC'nin çalışma alanı olarak kullandığı bellek kısmıdır. Veri belleği hesaplamalar, ara sonuçların geçici olarak saklanması, reçetelerde kullanılan sabitler ve kontrol parametre sabitleri için gerekli bellek alanlarını içerir. Veri belleği içine program kontrolünde girilebilir. Veri belleğinin bir kısmı kalıcı bellekte (yazılabilir/okunabilir) saklanır. PLC'nin enerjisi kesildiğinde kalıcı bellekte saklanan bilgi kaybolmaz. Bu nedenle, sabitler ve diğer bilgiler kalıcı bellekte saklanabilir. Kalıcı bellek program belleği gibi aynı karakteristiğe sahiptir. Veri belleğinin diğer kısmı RAM sıradan erişimli bellekte saklanır. RAM'deki bilgiler enerji kesildikten sonra kısıtlı bir zaman içinde pil yardımıyla saklı kalır. Bu zaman sonunda silinir.

4.2.3 PARAMETRE BELLEĞİ

Şifre, istasyon adresi, kalıcı bilgiler gibi yapısal parametre seçimlerinin üretici firma tarafından atanan yada sonradan kullanıcının değiştirdiği değerlerinin saklandığı yerdir. Parametre belleğinin içeriği program belleği ile aynı karakteristiğe sahiptir ve kalıcı bellekte saklanır.

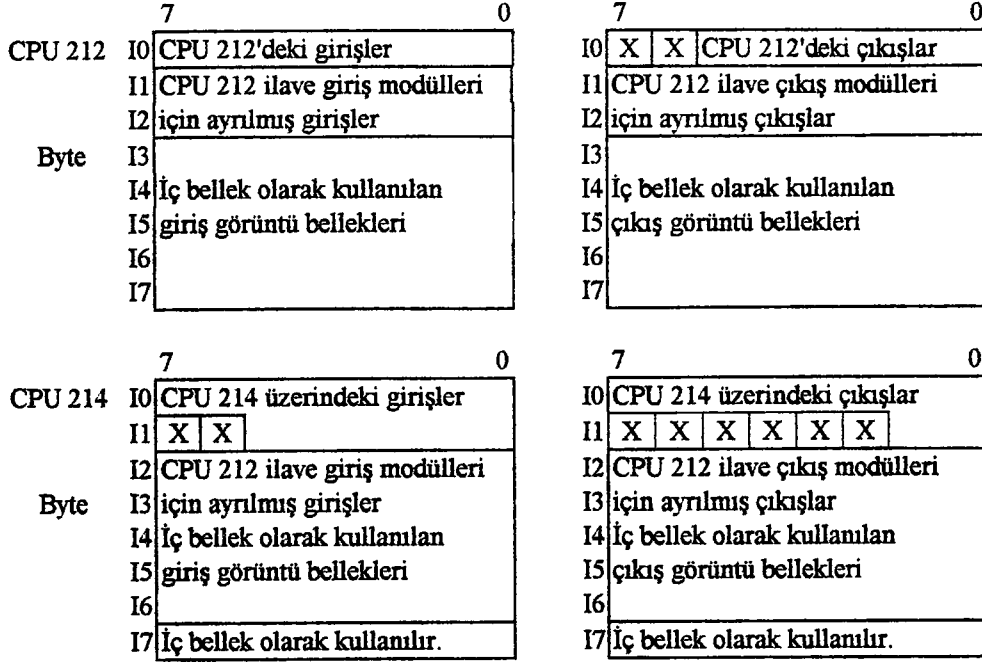
4.3 PLC PROGRAMININ YÜRÜTÜLMESİ

Program “tarama” diye tabir ettiğimiz bir döngü içinde yürütülür. Bunlar; Girişleri okuma, Programı işletme, Mesajların işletimi, İç kontroller, Çıkışların yazılmasıdır. Bu işlevler bir düzen ve sıra içinde yerine getirilirler. Ek olarak kullanıcı kesmeleri önceliklerine göre işletilirler. Kesme işlevi her taramada düzenli olarak yapılmaz, kesme olayı meydana geldiğinde yürütülür. Eğer kesme kullanılacaksa, her kesme olayı için program blokları programın bir parçası olarak kaydedilir. PLC analog giriş ve çıkışları otomatik olarak tarama çevriminde güncelleştiremez. Program içinde doğrudan analog giriş ve çıkışlara ulaşmak mümkündür. Şekil 4.2 bir tarama çevrimini gösterir.



Şekil 4.2 PLC tarama çevrimi

Girişleri okuma: Her tarama başında öncelikle giriş bitleri okunur ve bunlar giriş görüntü belleklerine yazılır. Giriş görüntü bellekleri I0 - I7, Q0 - Q7 olmak üzere 8 byte büyüklüğündedir. Bunlardan I0.0-I0.7 CPU 212 üzerine mevcut bulunan girişler içindir. I1.0-I1.7 ve I2.0-2.7 ek girişler için ayrılmıştır. I3-I7 byte'ları ise program içinde iç bellek bitleri olarak kullanılabilir. Q0.0-Q0.5 çıkışları CPU üzerinde mevcuttur. Q0.6, Q0.7 çıkış bitleri kullanılmaz. Q1.0-Q1.7 ve Q2.0-Q2.7 ek çıkışlar için ayrılmıştır. Q3,Q7 byte'ları iç bellek biti olarak kullanılabilir. CPU 212 ve 214 için giriş ve çıkış görüntü bellekleri şekil 4.3'te verilmiştir. Şekilde X ile gösterilen bitler kullanılmayan bitlerdir.



Şekil 4.3 S7 200 PLC giriş ve çıkış görüntü bellekleri

Programı işletme: PLC programın işletimine ilk komuttan başlar, son komuta kadar sürdürür. Immediate I/O (derhal G/Ç) ister program ister kesme işlevinin gerçekleşmesi sırasında giriş ve çıkışlara anında ulaşmayı içerir.

Mesajların işletimi: Bu tarama çevrimi diliminde PLC haberleşme kapısından aldığı mesajları değerlendirir.

İç kontroller : PLC program belleği, giriş çıkış modülleri vs. kontrolünün periyodik olarak yapılmasını içerir.

Çıkışların yazılması : Çıkış görüntü belleğindeki değerler çıkış modüllerine yazılır ve bir tarama çevrimi tamamlanmış olur. CPU 212 ve 214 için giriş ve çıkış görüntü bellekleri şekil 4.3'te verilmiştir.

4.4 S7 - 200 PLC VERİ ALANI

Veri belleği ve veri nesneleri olarak iki kısma ayrılabilir.

Veri Belleği : Değişken bellek (V bellek) , giriş görüntü belleği(I), çıkış görüntü belleği(Q), iç bellek bitleri(M), özel bellek bitlerinden (SM) oluşur. Bunlar yazılabilir ve okunabilir (SMB'nin yalnızca okunur tipleri hariç) bellek alanlarıdır ve bit, byte, word, doubleword olarak erişilebilir.

Veri nesneleri : Sayıcının saydığı zaman değeri gibi bilgilerin saklandığı bellek alanlarıdır. Zamanlayıcı, sayıcı, analog giriş, analog çıkış, akümülatör ve yüksek hızlı sayıcı verilerinden oluşur. RAM veri bellek ve nesnelerinin depolanmasını sağlar. CPU212'de 50 saat kadar CPU214'de 190 saat kadar üzerindeki bilgiyi kaybolmadan saklayabilir. Analog giriş, analog çıkış ve yüksek hızlı sayıcı verileri RAM'den ziyade kendi modülleri içinde saklanırlar. Şekil 4.4'te veri alanının sembolik gösterilimi, şekil 4.5'te ise CPU 212 için veri bellek ve nesneleri gösterilmiştir.

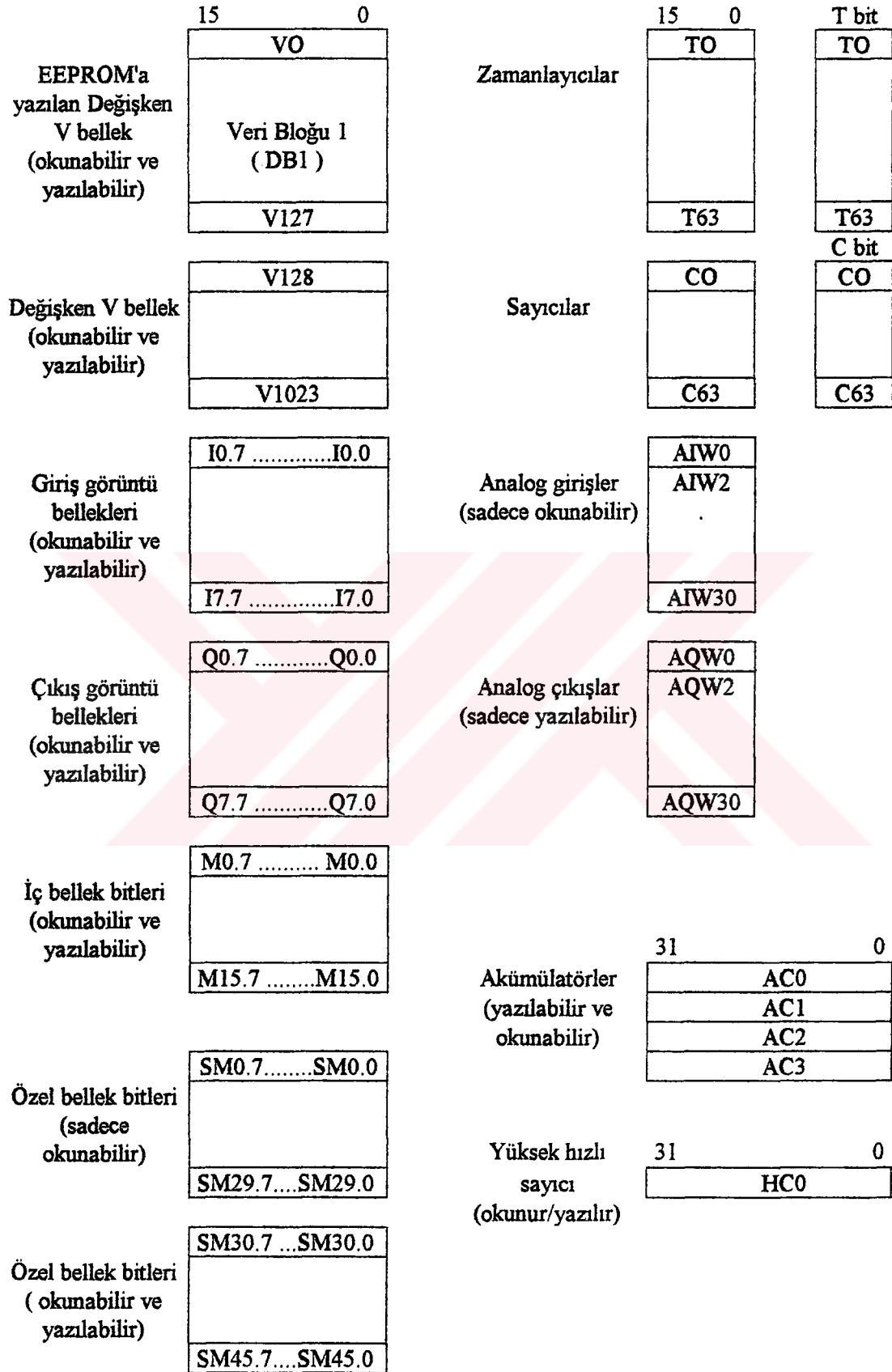


Şekil 4.4 S7 200 PLC veri alanı

Değişken bellek (V bellek); iki kısma ayrılır, her iki kısım da RAM'de saklanır, okunabilir ve yazılabilir. V belleğinin birinci kısmı DB1 veri bloğu olarak tanımlanır. CPU 212'de ilk 128 byte'lık CPU 214'de ilk 512 byte'lık kısımdır. Bu kısım aynı zamanda EEPROM'a da yüklenir. V belleğinin ikinci kısmı sadece RAM'de saklanır. EEPROM'a yüklenmez.

Giriş Ve Çıkış Görüntü Bellekleri; PLC her taramanın başında girişlerin durumunu giriş görüntü belleklerine kopyalar. Programın işletimi sırasında giriş görüntü belleklerinden alınan giriş bilgileri değerlendirilir. Tarama süresince gerçek girişler değişse bile giriş görüntü bellekleri nazara alındığından etkisi olmaz, ancak diğer taramanın başında işleme sokulurlar. Program işletimi sırasında girişlerin durumuna göre olması gereken çıkış bilgileri çıkış görüntü belleklerine yazılır, taramanın sonunda çıkış görüntü belleklerinin içeriği çıkışa aktarılır. Bu bir döngü halinde devam eder. Giriş görüntü bellekleri kullanmasının üç ana nedeni :

a - Tarama başında tüm girişleri örneklemek, bu değerleri sabit tutarak program komutlarını gerçeklemek ve program sonunda da çıkışları güncellemek sistemde kararlılığı sağlar.



Şekil 4.5 Simatic S7 CPU 212 PLC veri belleği

b - Girişleri giriş görüntü belleklerinden okumak, gerçek girişleri okumaktan daha kısa zaman aldığından tarama süresi kısadır.

c - Gerçek girişler bit olarak değerlendirilebilir, fakat giriş görüntü bellekleri bit, byte, word, doubleword olarak işleme sokulabilir. Bu da işletimde ek bir esneklik sağlar. Ayrıca giriş ve çıkış görüntü bellekleri giriş ve çıkışlardan daha fazla sayıda mevcuttur, bu da kullanılmayan giriş ve çıkışların görüntü belleklerinin program içinde iç bellek biti olarak kullanılmasını sağlar.

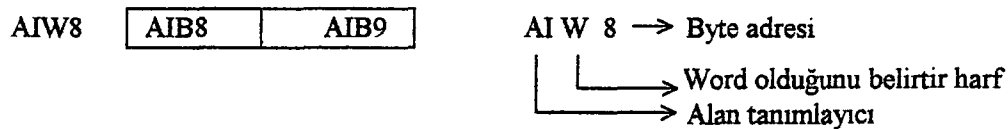
Immediate I/O (Derhal G/Ç); Program işletimi sırasında gerçek giriş ve çıkışlara doğrudan erişmek için kullanılır. Giriş noktasına doğrudan ulaşımında giriş görüntü belleği değişmez, gerçek giriş okunur. Çıkış noktasına doğrudan ulaşımında derhal çıkış görüntü belleğinde ilgili çıkış biti değiştirilerek çıkışa anında ulaşım sağlanır.

İç Bellek Biti (M); Çıkışa aktarılmayan program içinde ara çıkış olarak kullanılan bitlerdir. İç bellek bitleri genelde bit olarak kullanıldığı gibi byte, word, doubleword olarak da kullanılabilir.

Özel Bellek Bitleri (SM); PLC sistemi ve program arasında haberleşmeyi sağlayan ve bir takım fonksiyonları da içeren bitlerdir.

Zamanlayıcı Ve Sayıcılar; Zamanlayıcılar, zaman değerini sayan cihazlardır. 1ms,10ms,100ms hassasiyetli zamanlayıcılar mevcuttur. Sayıcılar, sayma girişindeki her yükselen kenarı sayan cihazlardır.

Analog Giriş Ve Çıkışlar; Analog modüller gerilim, sıcaklık gibi gerçek analog büyüklükleri bir dijital kelime değerine dönüştürür. Analog modüller giriş modülü, çıkış modülü ve giriş/çıkış modülü olabilir. AI ve AO analog modülünden okunur ya da yazılabilen 16 bitlik word değerleridir, analog girişler yalnız okunabilir, analog çıkışlar yalnız yazılabilir AI analog girişi, AO analog çıkışı belirtir ve word değeri olduğunu belirten W eki ile birlikte AIW, AOW olarak tanımlanır. AIW,AOW 0'dan 30'a kadar ikişerli aralıklarla tanımlıdır. (AIW0,AIW2,AIW4,...)



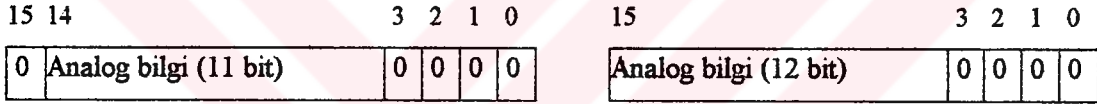
Şekil 4.6 Analog giriş ve çıkışlar

Analog giriş modülüne çeşitli değer aralıklarında gerilim ve akım değerleri uygulanabilir. Örneğin tez uygulamasında kullanılan EM235 3 analog giriş/1 analog çıkışlı modüle 0-50mV, 1-100mV, 0-500mV, 0-1V, 0-5V, 0-10V ve 0-20mA giriş olarak uygulanabilir. Hangi girişin uygulanacağı ve girişin işaretli-işaretsiz olup olmadığı modüldeki dip switch ile seçilir. İşaretli analog giriş bilgisinde en anlamlı bitin sıfır olması pozitif analog değer olduğunu belirtir. İşaretsiz bilgide sağdan üç bit, işaretli bilgide sağdan dört bit okunmaz.



Şekil 4.7 İki kutuplu ve tek kutuplu analog giriş bilgisi formatı

Analog çıkış $\pm 10V$ veya 0-20mA olabilir. Akım bilgisi 11 bit, gerilim bilgisi 12 bit'ten ibarettir ve her iki bilgi için sağdan dört bit sıfır olarak kaydedilir.



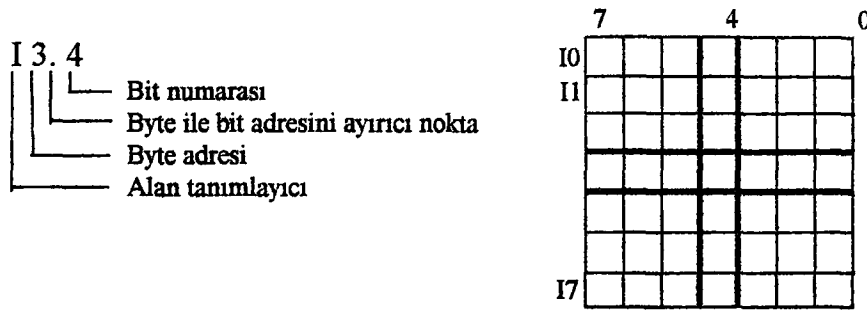
Şekil 4.8 Analog çıkış akım ve gerilim bilgisi formatı

Akümülatörler ; Akümülatörler bellek gibi okunabilir ve yazılabilen byte'lardır. Örneğin alt-programlara parametre geçirmede komutları parametrelendirmede kullanılır. Bir kesme durumunda kesme işleminden önce akümülatör üzerindeki bilgi PLC tarafından saklanır ve kesme sonucunda bu bilgi tekrar akümülatöre yüklenir.

Yüksek Hızlı Sayıcı; PLC tarama süresinden daha hızlı sayma yapmak amacıyla kullanılır. CPU212'de bir adet, CPU214'de üç adet yüksek hızlı sayıcı mevcuttur. Yüksek hızlı sayıcılar işaretli 32 bitlik tamsayı sayma değerine sahiptir.

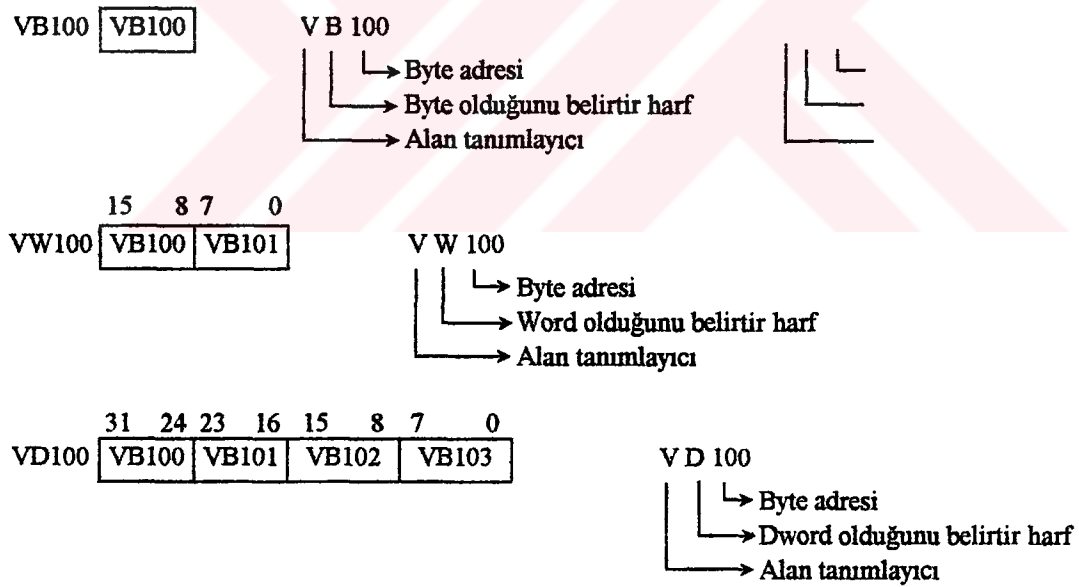
4.5 VERİ BELLEĞİNE GİRİŞ

Veri belleği 5 kısma ayrılır; I,Q,M,SM,V . Erişmek istenen verinin adresini belirtmek lazımdır. Belleğe bit, byte, word, doubleword olarak adresleme yapmak mümkündür. Bir bite ulaşmak için o bitin adresi tanımlanmalıdır. Adresleme yapılırken önce alan tanımlayıcı, ardından byte adresi, nokta, bit adresi (0..7) yazılır.



Şekil 4.9 Bit adresleme

Byte, word ve double word adresleri de aynı bit adresi gibi tanımlanır. Önce alan tanımlayıcı, sonra büyüklüğünü belirten harf, ardından byte adresi gelir. Byte adreslerken byte olduğunu belirtir harfin ardından byte numarası yazılır. Word iki byte'dan oluşur, dolayısıyla word adreslerken ardışıl iki adet byte adreslenmiş olur. Örneğin VW100 yazıldığında VW100'ün yüksek anlamlı byte'ı VB100 düşük anlamlı byte'ı VB101'dir. Dword dört byte'dan oluşur ve adreslerken dört adet ardışıl byte adreslenmiş olur. Şekil 4.10'da adres tanımlamalarının nasıl yapıldığı gösterilmiştir.



Şekil 4.10 Byte, word, doubleword adresleme

BÖLÜM 5

PLC'LERİN PROGRAMLANMASI

Programlanabilir Lojik Kontrolörler endüstriyel bir tesis veya makinanın önceden yazılmış bir programa ve durum değişikliğine göre kontrolünü yapan mikroişlemci bazlı bir kontrol sistemidir. Kontaklı kumanda devrelerinde sistem kablo bağlantılarıyla yapılmakta ve işin başında tüm sistemin kararlaştırılması ve uygulanması gerekmektedir. Sonradan yapılacak bir değişiklik oldukça zor ve uzun sürecektir. Fakat PLC sistemlerinde istenilen mantık kablolarla değil programlama suretiyle olacağından oldukça pratiktir. PLC'ler için geliştirilmiş olan programlama dilleri, kontaktörlü ve röleli kumanda devrelerinin tasarımı ile ilgilenen kişilerin kolayca anlayıp uygulayabileceği biçimdedir. Genel olarak üç çeşit programlama yönteminden söz edilebilir.

- Deyim listesi ile programlama (Statement list, Instruction list)
- Merdiven diyagramı ile programlama
- Diğer programlama yöntemleri (Grafcet, lojik kapı sembolleri, lojik akış veya sembolik dille programlama)

Bu programlama yöntemlerinden deyim listesi ve merdiven diyagramı ile programlama genellikle el programlayıcılarında kullanılır. Kişisel bilgisayarlarda her üç yöntemi de kullanmak mümkündür. Bu bölümde Simatic S7 200 PLC komutları açıklanacaktır.

5.1 PLC KOMUTLARINA GİRİŞ

Simatic S7-200 PLC'sinde merdiven diyagramı (LAD) ve deyim listesi (STL) ile program yapmak mümkündür.

Merdiven diyagramı : Endüstriyel kumanda sisteminde kullanılan elemanların çizimine benzer olarak oluşturulan bir grafik dildir. Merdiven diyagramı ile programlamada; lojik akışı gösteren basit elemanlar kullanılır.

Kontaklar : Kapalı olduğunda güç akışına izin veren anahtar kontağıdır. Normalde açık (—|—) ve normalde kapalı (—|/—)

Bobin : Kontaktör veya röleyi ifade eder.

Kutular : Kutuya güç akışı geldiğinde çeşitli fonksiyonları gösteren sembollerdir. Zamanlayıcı, sayıcı, matematik fonksiyonları vs. için kutular mevcuttur.

Bobin ve kutular merdiven diyagramında sağa dayalı olmalıdır. Güç soldaki çizgiden sağdakine doğru akar. Sol çizgi enerjili iletkeni sağ ise nötr iletkeni temsil eder. Step7 Micro Software yazılımında sağ çizgi gösterilmez. Güç sol çizgiden kapalı kontaklar üzerinden bobin yada kutulara iletilir.

Deyim Listesi : PLC fonksiyonlarının kısaltmalarının oluşturduğu programlama dilidir. Program merdiven diyagramında yazılmışsa, programlama cihazı bu programı deyim listesine çevirir. Deyim listesi ile program yapmak için öncelikle S7-200 PLC'sinin 9 bit derinlik ve 1 bit genişlikli yığın kullanımını öğrenmek gereklidir. Tüm işlemler genellikle yığının ilk biti yada 1. ve 2. biti ile yapılır.

5.2 TEMEL LOJİK KOMUTLAR

Kontaklı kumanda devresindeki seri ve paralel bağlı kontaklar yerine kullanabileceğimiz temel lojik komutlar aşağıda verilmiştir. Bu komutların lojik yığını nasıl kullandıkları da anlatılmıştır.

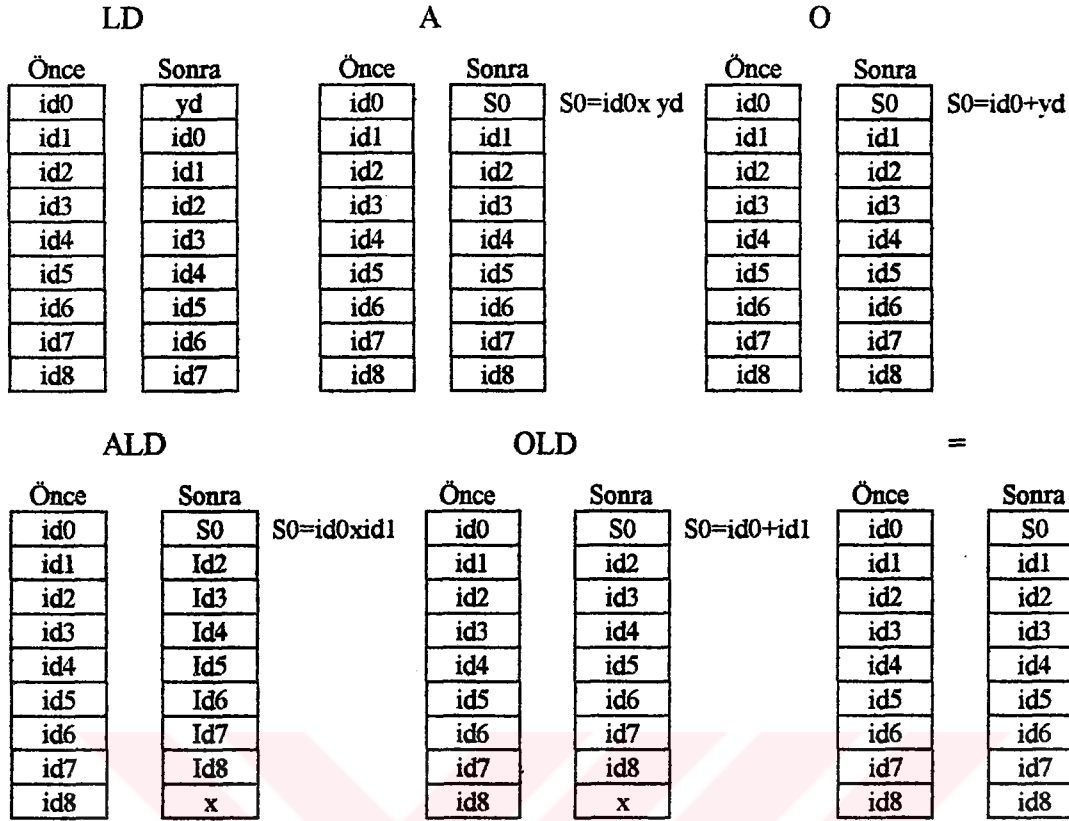
LD (LOAD): Yığının en üst seviyesine operandın değerini yükler. Yandaki şekilde; (yd) yığına yüklenen yeni değer (id0-id8) yığındaki ilk değerler olarak gösterilmiştir.

AND,OR: Operandın değeri ile yığının üst seviyesine ve / veya işlemi uygular, yine yığının üst seviyesine yazar.

ALD,OLD: Yığının ilk iki seviyesine ve / veya işlemi uygular, sonucu yığının üst seviyesine yazar.

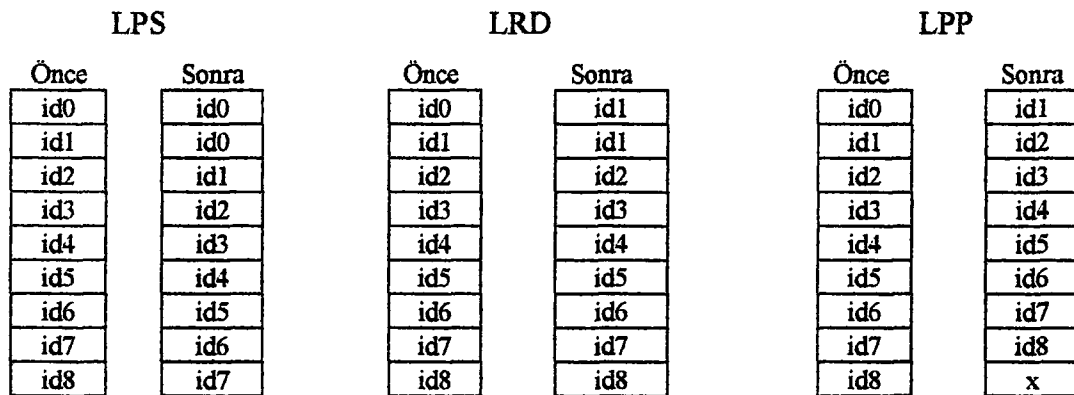
ÇIKIŞ (=): Yığının en üst seviyesindeki değer belirtilen çıkış bitine kopyalanır ve yığının tüm seviyeleri değişmez.

YÜKLEME, VE, VEYA ve ÇIKIŞ komutlarının yığını nasıl etkiledikleri şekil 5.1'de gösterilmiştir.



Şekil 5.1 Yükleme, ve, veya, çıkış komutları

LPS,LRD,LPP : Bu komutlar yığın üzerinde özel işlemler yaparlar. Merdiven diyagramında programlanamazlar. LPS (Lojic Push) komutu ; yığının üst seviyesini ikinci seviyesine kopyalar ve diğer seviyeleri bir geri iter. LRD (Lojic Read) komutu; yığının ikinci seviyesini birinci seviyesine kopyalar fakat diğer seviyelerde değişiklik, itme yada geri çekme yapmaz. LPP (Lojic Pop) komutu ; yığının üst seviyesindeki bilgiyi atar ve tüm seviyelerdeki bilgileri bir geri çeker. Sekizinci seviye anlamsız olarak kalır. Bu komutların yığını nasıl etkiledikleri şekil 5.2’de gösterilmiştir.



Şekil 5.2 LPS, LRD, LPP komutları

NOT, EU, ED : Bu komutlar yine yığın üzerinde özel işlemler uygular. NOT, yığının üst seviyesinin lojik olarak değilini alır ve aynı yere yükler. EU, her taramada 0'dan 1'e geçişi kontrol eder ve bu geçiş olduğunda yığının üst seviyesi "1" olur. ED, her taramada 1'dan 0'e geçişi kontrol eder ve bu geçiş olduğunda yığının üst seviyesi "1" olur. Bu komutların yığını nasıl etkiledikleri şekil 5.3'te gösterilmiştir.

NOT		EU		ED	
Önce	Sonra	Önce	Sonra	Önce	Sonra
Id0	id0	id0	id1	id0	id1
Id1	id0	id1	id1	id1	id2
id2	id1	id2	id2	id2	id3
id3	id2	id3	id3	id3	id4
id4	id3	id4	id4	id4	id5
id5	id4	id5	id5	id5	id6
id6	id5	id6	id6	id6	id7
id7	id6	id7	id7	id7	id8
id8	id7	id8	id8	id8	x

Şekil 5.3 NOT, EU, ED komutları

Tablo 5.1 Set ve reset komutları

KOMUT		TANIM	OPERANDLAR
LAD	STL		
n ——()	= n	Yığının en üst noktasındaki bilgi "n" bitine aktarılır.	<u>n (bit):</u> I,Q,M,SM,T,C,V
S_BIT N ——(S)	S S_BIT,N	S-BIT 'den başlayarak N adet biti "1" yapar.	<u>S-BIT(bit):</u> I,Q,M,SM,T,C,V
S_BIT N ——(R)	R S_BIT,N	S-BIT 'den başlayarak N adet biti "0" yapar. Eğer S-BIT zamanlayıcı yada sayıcı olarak tanımlanmışsa, hem zamanlayıcı/ sayıcı biti hem de zamanlayıcı/sayıcı değeri resetlenir.	<u>N (byte):</u> IB,QB,MB,SMB, VB,AC,sabit, *VD,*AC
n ——(I)	= I n	Yığının en üst noktasındaki bilgi derhal "n" çıkışına aktarılır.	<u>n(bit):</u> Q
S_BIT N ——(S_I)	SI S_BIT,N	S-BIT 'den başlayarak N adet çıkışı derhal "1" yapar.	<u>S-BIT(bit):</u> Q
S_BIT N ——(R_I)	RI S_BIT,N	S-BIT 'den başlayarak N adet çıkışı derhal "0" yapar.	<u>N (byte):</u> IB,QB,MB,SMB,VB, AC,sabit,*VD,*AC

5.3 SET VE RESET İŞLEMLERİ

Set ve Reset deyimleri yetkilendirildiğinde ilgili çıkışı veya "n" adet ardışıl çıkış dizisini "1" veya "0" yapar. "n" 1-255 arasında bir değer olabilir. Set komutu ile çıkış "1" yapılıncaya set girişindeki sinyal "0" olsa dahi reset girişindeki sinyal "1" olana kadar çıkış değişmez. Set ve reset komutlarına "T" teriminin eklenmesi ile çıkış yada çıkış

dizileri derhal “1” yada “0” yapılır. Bunun anlamı çıkışın durumu hem çıkış görüntü belleklerine hem de gerçek çıkış noktalarına anında taramanın sonunu beklemeksizin aktarılır. Tablo 5.1’de merdiven diyagramı ve deyim listesinde set ve reset komutlarının gösterilimi ve açıklanması verilmiştir.

5.4 ZAMANLAYICILAR

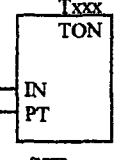
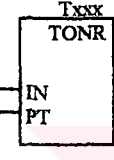
Simatic S7-200 PLC’inde TON ve TONR olarak adlandırılan iki çeşit zamanlayıcı vardır. Bu iki zamanlayıcı tablo 5.2’de açıklanmıştır. TON zamanlayıcısı IN girişi lojik “1” olduğu sürece sayar, diğer durumda resetlenir. Ayrıca reset komutu ile de resetlemek mümkündür. TONR zamanlayıcısı IN etkin olduğu süre boyunca sayar ve IN girişi etkin değilken kaldığı zaman değerini hafızaya alır, IN girişi tekrar lojik “1” olduğunda kaldığı yerden saymaya devam eder. TONR zamanlayıcısını resetlemek için reset komutunu kullanmak gerekir. Reset komutu ile hem Txxx zamanlayıcı biti hem de zaman değeri resetlenir.

Zamanlayıcılar 1ms, 10ms, 100ms lik olmak üzere 3 çeşittir ve sırasıyla 32,767 s, 327,67 s, 3276,7 s maksimum değere sahiplerdir. PT zaman ayar değerleri şöyle belirlenir; İstenen zaman ayar değeri = PT x (1ms/10ms/100ms zamanlayıcısı) Örneğin 10ms.’lik zamanlayıcıda 50 sabiti verilirse $50 \times 10\text{ms} = 500\text{ms}$ ayarlanmış demektir. 1ms’lik zamanlayıcılar programın neresinde olursa olsun her 1ms’de güncelleştirilir. Dolayısıyla bir program taraması boyunca bir çok kere güncelleme yapılır. Bu neden-den ötürü program işletimi sırasında bir tarama boyunca zamanlayıcının zaman değeri sabit olması garantilenmez. 10ms’lik zamanlayıcı program taramasının en başın-da otomatik olarak güncelleştirilir ve bir tarama boyunca zaman değeri sabit kalır.

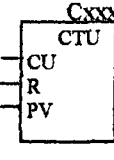
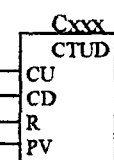
5.5 SAYICILAR

Sayıcılar, kumanda edilen sisteme ilişkin bir takım olayların sayılması ve değerlendirilmesi amacıyla kullanılır. Bu amaçla geleneksel kumanda devrelerinde özel elektronik yada elektromekanik düzenler kullanılır. PLC’lerde bu işlem program komutları ile sağlanır. Simatic S7 PLC’inde CTU yukarı sayma, CTUD yukarı/aşağı sayma olarak iki adet sayıcı mevcuttur. Bunların çalışma şekli, her iki dilde nasıl yazıldığı tablo 5.3’te verilmiştir.

Tablo 5.2 Zamanlayıcı komutları

KOMUT	TANIM	OPERAND												
LAD  STL TON Txxx,PT	Açmada gecikmeli (TON) zamanlayıcısı IN girişi lojik "1" olunca maksimum değere kadar zamanı sayar,zaman PT ile ayarlanmış zaman değerine eşit veya büyük olduğunda zamanlayıcı biti Txxx="1" olur. IN girişindeki sinyal kalkarsa zamanlayıcı resetlenir.Zaman maksimum değere ulaştığında sayma işlemi durur. Deyim listesinde yığının en üst seviyesi "1" olduğunda zamanlayıcı çalışır. Txxx >=PT olduğunda zamanlayıcı biti Txxx = 1 olur.Yığının en üst seviyesi "0" olduğunda zamanlayıcı resetlenir. <table border="1"> <thead> <tr> <th></th><th>CPU 212/214</th><th>CPU 214</th></tr> </thead> <tbody> <tr> <td>1ms</td><td>T32</td><td>T96</td></tr> <tr> <td>10ms</td><td>T33-T36</td><td>T97-T100</td></tr> <tr> <td>100ms</td><td>T37-T63</td><td>T101-T127</td></tr> </tbody> </table>		CPU 212/214	CPU 214	1ms	T32	T96	10ms	T33-T36	T97-T100	100ms	T37-T63	T101-T127	<u>Txxx</u> (212) T32-T63 (214) T32-T63 T96-T127 <u>PT</u>: VW, T,C, IW, QW, MW , SMW, AC, AIW, Sabit, *VD, *AC
	CPU 212/214	CPU 214												
1ms	T32	T96												
10ms	T33-T36	T97-T100												
100ms	T37-T63	T101-T127												
LAD  STL TONR Txxx,PT	Hafızalı açmada gecikmeli (TONR) zamanlayıcısı IN girişi etkinken maksimum değere kadar zamanı sayar, zaman PT ile ayarlanmış değerine eşit veya büyük olunca zamanlayıcı biti Txxx = "1" olur. IN girişindeki sinyal kalkarsa yada zaman maksimum değere erişirse sayma işlemi durur, fakat resetlenmez. IN girişi tekrar etkin olduğunda saymaya kaldığı yerden devam eder. Deyim listesinde, yığının en üst seviyesi "1" iken zamanlayıcı max. değere kadar sayar. Txxx >=PT olduğunda zamanlayıcı biti Txxx = "1" olur.Yığının en üst seviyesi "0" iken ve zaman maksimum değere ulaşınca sayma işlemi durur. <table border="1"> <thead> <tr> <th></th><th>CPU 212/214</th><th>CPU 214</th></tr> </thead> <tbody> <tr> <td>1ms</td><td>T0</td><td>T64</td></tr> <tr> <td>10ms</td><td>T1-T4</td><td>T65-T66</td></tr> <tr> <td>100ms</td><td>T5-T31</td><td>T67-T95</td></tr> </tbody> </table>		CPU 212/214	CPU 214	1ms	T0	T64	10ms	T1-T4	T65-T66	100ms	T5-T31	T67-T95	<u>Txxx</u> : (212) T0-T31 (214) T0-T31 T64-T95 <u>PT</u>: VW, T, C IW, QW, MW SMW, AC,AIW Sabit,*VD, *AC
	CPU 212/214	CPU 214												
1ms	T0	T64												
10ms	T1-T4	T65-T66												
100ms	T5-T31	T67-T95												

Tablo 5.3 Sayıcı Komutları

KOMUT	TANIM	OPERANDLAR
LAD  STL CTU Cxxx, PV	Yukarı sayma (CTU) sayıcısı CU girişindeki sinyalin her yükselen kenarında değerini bir artırır. Eğer Cxxx >= PV (ayarlanan değer) olursa Cxxx sayıcı biti "1" olur. Reset girişi yetkilendirilirse resetlenir. Resetlenmediği sürece sayma işlemi max. değere (32.767) kadar kadar devam eder. CTU CU girişinin yükselen kenarını sayar. CU girişi yığının 2. seviyesine, reset girişi yığının üst seviyesine yüklenir. Cxxx >= PV olduğunda Cxxx = 1 olur.	<u>Cxxx</u> (212) C0-C47 (214) C0-C47 C80-C127 <u>PV</u> VW, T, C IW, QW, MW SMW, AC,AIW Sabit, *VD, *AC
LAD  STL CTUD Cxxx,PV	Yukarı/aşağı (CTUD) sayıcısı CU girişinin yükselen kenarında yukarı doğru, CD girişinin yükselen kenarında aşağıya doğru sayar. Cxxx >= PV olduğunda sayıcı biti Cxxx = "1" olur. Maksimum değer 32.767 'ye ulaşınca yukarı saymayı, minimum değer -32.768 'e ulaşınca aşağıya saymayı durdurur. Reset girişi yetkilendirilirse sayıcı resetlenir. CU girişindeki değer yığının 3. seviyesine CD girişindeki değer yığının 2. seviyesine reset girişindeki değer de yığının üst seviyesine yüklenir.	<u>Cxxx</u> (212) C48-C63 (214) C48-C79 <u>PV</u> VW, T, C IW, QW, MW SMW, AC,AIW Sabit, *VD, *AC

5.6 KARŞILAŞTIRMA İŞLEMLERİ

Byte, word ve dword değerleri arasında karşılaştırma işlemini gerçekleştirmek için karşılaştırma kontakları kullanılır. Eşitlik, büyük-eşitlik ve küçük-eşitlik karşılaştırmalarını gerçekleştirmek mümkündür. Eşit değillik, küçüklük ve büyüklük karşılaştırmaları, mevcut olan eşitlik, büyük-eşitlik ve küçük-eşitlik komutlarına NOT komutu eklenerek oluşturulabilir. Örneğin, 50 sabiti ile VW100'ün içeriğine eşit değillik karşılaştırması yapılacaksa deyim listesi şöyle yazılır;

LDW= VW100,50

NOT

Karşılaştırma tipine bağlı olarak elde edilen "1" ve "0" karşılaştırma sonucu ya akümülatöre yüklenir ya da "ve" "veya" işlemlerine tabi tutulabilir.

Byte karşılaştırması işaretsizdir. (7FFF > 8000)

Word,doubleword ve reel karşılaştırması işaretlidir. (7FFFFFFF > 80000000)

5.7 ARİTMETİK İŞLEMLER

Toplama, çıkarma, çarpma ve bölme işlemlerini deyim listesi ve merdiven diyagramı ile programlama mümkündür. Çarpma işleminde 16 bitlik iki tamsayı 32 bitlik sonuç doğurur. Bölme işleminde bölen, bölünen, bölümün tamsayı kısmı ve bölümün ondalık kısmı 16 bitlik değerlerdir. Bölümün tamsayısı yüksek anlamlı 16 bit'e ondalığı düşük anlamlı 16 bit'e yüklenir. Aritmetik işlem komutlarından çarpma ve bölme tablo 5.5 'te, toplama ve çıkarma tablo 5.6'da verilmiştir.

5.8 ARTIRMA/AZALTMA/TERSLEME İŞLEMLERİ

Artırma azaltma komutları ile word veya doubleword değerler bir artırılır veya bir azaltılır. Tersleme komutu operandın lojik olarak değerini alır. Herhangi lojik işlemin sonucu "0" ise SM1.0 = "1" olur. Merdiven diyagramında programlarken OUT çıkışı IN olarak tanımlanırsa, daha az bellek harcanmış olur ve daha az STL deyim listesi komutları oluşturulur. Bu komutların merdiven diyagramı ve deyim listesinde gösterilimi tablo 5.7'de verilmiştir.

Tablo 5.4 Karşılaştırma işlem komutları

KOMUT(LAD/STL)		TANIM	OPERANDLAR
$\text{---} \mid \text{---} = \text{B} \mid \text{---}$ n1 n2	LDI = n1,n2 AI = n1,n2 OI = n1,n2	Byte karşılaştırma kontağı n1=n2 iken kapanır. LDB, AB,OB deyimleri n1=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	<u>n1,n2(byte) :</u> VB,IB
$\text{---} \mid \text{---} >= \text{B} \mid \text{---}$ n1 n2	LDI >= n1,n2 AI >= n1,n2 OI >= n1,n2	Byte karşılaştırma kontağı n1>=n2 iken kapanır. LDB, AB,OB deyimleri n1>=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	QB,MB SMB, AC, Sabit,*VD, *AC,SB
$\text{---} \mid \text{---} <= \text{B} \mid \text{---}$ n1 n2	LDW <= n1,n2 AW <= n1,n2 OW <= n1,n2	Byte karşılaştırma kontağı n1<=n2 iken kapanır. LDB, AB,OB deyimleri n1<=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	
$\text{---} \mid \text{---} = \text{I} \mid \text{---}$ n1 n2	LDW = n1,n2 AW = n1,n2 OW = n1,n2	Integer karşılaştırma kontağı n1=n2 iken kapanır. LDB, AB,OB deyimleri n1=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	<u>n1,n2(word) :</u> VW,T,C,IW, QW, MW, SMW, AC, AIW,sabit, *VD,*AC SW
$\text{---} \mid \text{---} >= \text{I} \mid \text{---}$ n1 n2	LDW >= n1,n2 AW >= n1,n2 OW >= n1,n2	Integer karşılaştırma kontağı n1>=n2 iken kapanır. LDB,AB,OB deyimleri n1>=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	
$\text{---} \mid \text{---} <= \text{I} \mid \text{---}$ n1 n2	LDW <= n1,n2 AW <= n1,n2 OW <= n1,n2	Integer karşılaştırma kontağı n1<=n2 iken kapanır. LDB,AB,OB deyimleri n1<=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	
$\text{---} \mid \text{---} = \text{D} \mid \text{---}$ n1 n2	LDD = n1,n2 AD = n1,n2 OD = n1,n2	Doubleword karşılaştırma kontağı n1=n2 iken kapanır. LDB,AB,OB deyimleri n1=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	<u>n1,n2(dword):</u> VD,ID,QD, MD, SMD,AC, HC,sabit, *VD,*AC,SD
$\text{---} \mid \text{---} >= \text{D} \mid \text{---}$ n1 n2	LDD >= n1,n2 AD >= n1,n2 OD >= n1,n2	Doubleword karşılaştırma kontağı n1>=n2 iken kapanır. LDB,AB,OB deyimleri n1>=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	
$\text{---} \mid \text{---} <= \text{D} \mid \text{---}$ n1 n2	LDD <= n1,n2 AD <= n1,n2 OD <= n1,n2	Doubleword karşılaştırma kontağı n1<=n2 iken kapanır. LDB,AB,OB deyimleri n1<=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	
$\text{---} \mid \text{---} = \text{R} \mid \text{---}$ n1 n2	LDR = n1,n2 AR = n1,n2 OR = n1,n2	Reel karşılaştırma kontağı n1=n2 iken kapanır. LDB,AB,OB deyimleri n1=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	<u>n1,n2(reel):</u> VD,ID,QD, MD, SMD,AC, HC,sabit, *VD,*AC, SD
$\text{---} \mid \text{---} >= \text{R} \mid \text{---}$ n1 n2	LDR >= n1,n2 AR >= n1,n2 OR >= n1,n2	Reel karşılaştırma kontağı n1>=n2 iken kapanır. LDB,AB,OB deyimleri n1>=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	
$\text{---} \mid \text{---} <= \text{R} \mid \text{---}$ n1 n2	LDR <= n1,n2 AR <= n1,n2 OR <= n1,n2	Reel karşılaştırma kontağı n1<=n2 iken kapanır. LDB,AB,OB deyimleri n1<=n2 olduğunda yığının en üst seviyesini "1" yapar, VE yada VEYA işlemi uygular.	

Tablo 5.5 Çarpma ve bölme işlemleri

KOMUT (LAD/STL)		TANIM	OPERANDLAR
MUL $\text{---} \mid \text{---} \text{EN} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN1} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN2} \text{OUT} \mid \text{---}$ MUL IN1,OUT DIV $\text{---} \mid \text{---} \text{EN} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN1} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN2} \text{OUT} \mid \text{---}$ DIV IN1,OUT		16 bitlik tamsayılarda çarpma ve bölme işlemi yapar. Bölmede 32 bitlik çıkışın 16 bitlik düşük anlamlı kısmı bölümü, 16 bitlik yüksek anlamlı kısmı kalanı verir. Mer.diyag: IN1*IN2=OUT, IN1/IN2=OUT Dey.lis: IN1*OUT=OUT, OUT/IN1=OUT	<u>IN1,IN2:</u> VW,T,C,IW, QW,MW,SMW,AC, AIW,Sabit,*VD, AC,SW <u>OUT:</u> VD,ID,QD, MD, SMD,AC,*VD,*AC, D
MUL_R $\text{---} \mid \text{---} \text{EN} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN1} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN2} \text{OUT} \mid \text{---}$ *R IN1,OUT DIV_R $\text{---} \mid \text{---} \text{EN} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN1} \mid \text{---}$ $\text{---} \mid \text{---} \text{IN2} \text{OUT} \mid \text{---}$ /R IN1,OUT		32 bitlik reel sayılarda Çarpma ve bölme işlemi yapar. Merdiven diyagramında IN1*IN2=OUT IN1/IN2=OUT Deyim listesinde IN1*OUT=OUT OUT/IN1=OUT	<u>IN1,IN2:</u> VD,ID,QD,MD, SMD,AC,Sabit,*VD, *AC,SD <u>OUT:</u> VD,ID,QD,MD, SMD, AC,*VD,*AC, SD

Tablo 5.6 Toplama ve çıkarma işlemleri

KOMUT (LAD/STL)		TANIM	OPERANDLAR
ADD_I EN IN1 IN2 OUT +I IN1,OUT SUB_I EN IN1 IN2 OUT -I IN1,OUT		16 bitlik tamsayılarla toplama ve çıkarma işlemi yapar. Merdiven diyagramında $IN1+IN2=OUT$ $IN1-IN2=OUT$ Deyim listesinde $IN1+OUT=OUT$ $OUT-IN1=OUT$	<u>IN1,IN2:</u> VW,T,C,IW, QW,MW,SMW,AC,AIWS abit,*VD,*AC,SW <u>OUT:</u> VW,T,C,IW,QW, MW, SMW, AC,*VD,*AC,SW
ADD_DI EN IN1 IN2 OUT +D IN1,OUT SUB_DI EN IN1 IN2 OUT -D IN1,OUT		32 bitlik tamsayılarla toplama ve çıkarma işlemi yapar. Merdiven diyagramında $IN1+IN2=OUT$ $IN1-IN2=OUT$ Deyim listesinde $IN1+OUT=OUT$ $OUT-IN1=OUT$	<u>IN1,IN2:</u> VD,ID,QD,MD SMD,AC,HC,Sabit,*VD,*AC,SD <u>OUT:</u> VD,ID,QD, MD, SMD, AC,*VD,*AC, SD
ADD_R EN IN1 IN2 OUT +R IN1,OUT SUB_R EN IN1 IN2 OUT -R IN1,OUT		32 bitlik reel sayılarda toplama ve çıkarma işlemi yapar. Merdiven diyagramında $IN1+IN2=OUT$ $IN1-IN2=OUT$ Deyim listesinde $IN1+OUT=OUT$ $OUT-IN1=OUT$	<u>IN1,IN2:</u> VD,ID,QD,MD SMD,AC,HC,Sabit,*VD,*AC,SD <u>OUT:</u> VD,ID,QD, MD, SMD, AC,*VD,*AC, SD

Tablo 5.7 Artırma, azaltma ve tersleme komutları

KOMUT (LAD / STL)			TANIM	OPERANDLAR
INC_B EN IN OUT INCB IN DEC_B EN IN OUT DECB IN INV_B EN IN OUT INVB IN			IN giriş değerine INCW "1" ekler, DECW "1" çıkarır, INVW tersine çevirir ve sonucu merdiven diyagramında çıkışa, deyim listesinde IN girişine yazar.	<u>IN :</u> VB,IB,QB,MB, SMB, AC, AIW, sabit,*VD,*AC,SB <u>OUT :</u> VB,IB,QB,MB,SMB, AC,*VD,*AC,SB
INC_W EN IN OUT INCW IN DEC_W EN IN OUT DECW IN INV_W EN IN OUT INVW IN			IN giriş değerine INCW "1" ekler, DECW "1" çıkarır, INVW tersine çevirir ve sonucu merdiven diyagramında çıkışa, deyim listesinde IN girişine yazar.	<u>IN:</u> VW,T,C,IW, QW, MW,SMW,AC, AIW, sabit,*VD,*AC,SW <u>OUT :</u> VW,T,C,IW QW,MW,SMW,AC,*VD,*AC,SW
INC_DW EN IN OUT INCW IN DEC_DW EN IN OUT DECW IN INV_DW EN IN OUT INVW IN			IN giriş değerine INCW "1" ekler, DECW "1" çıkarır, INVW tersine çevirir ve sonucu merdiven diyagramında çıkışa, deyim listesinde IN girişine yazar.	<u>IN :</u> VD,ID,QD,MD, SMD, AC,HC,sabit *VD,*AC,SD <u>OUT :</u> VD,ID,QD, MD,SMD,AC,*VD,*AC,SD

5.9 VERİ TRANSFERİ KOMUTLARI

Byte, word, dword değerlerini bir yerden başka bir yere kopyalamaya yarar. Swap komutu ile bir word değerinin yüksek anlamlı byte'ı ile düşük anlamlı byte'ı yer

değiştirilir. Blok komutları tek bir komutla bir dizi işlem yapmayı sağlayan komutlardır. Bir kaynak veri dizisinden başka bir veri dizisine kopyalama yapar. Fill komutu ise; belleğin bir kısmını belirtilen bir word değeri ile doldurur.

Tablo 5.8 Veri transferi komutları

KOMUT(LAD/STL)		TANIM	OPERANDLAR
MOV_B EN IN OUT	MOVB IN,OUT	EN girişi "1" olunca IN girişindeki byte değeri OUT çıkışına aktarır. Giriş byte'ı değişikliğe uğramaz.	<u>IN</u> : VB,IB,QB,MB,SMB, (byte) AC,sabit,*VD,*AC <u>OUT</u> : VB,IB,QB,MB,SMB, (byte) AC,*VD,*AC
MOV_W EN IN OUT	MOVW IN,OUT	EN girişi "1" olunca IN girişindeki word değeri OUT çıkışına aktarır. Giriş word'ü değişikliğe uğramaz.	<u>IN</u> : VW,T,C,IW,QW,MW,SMW (word) AC,AIW,sabit, *VD,*AC <u>OUT</u> : VW,T,C,IW,QW,SMW (word)MW,AC,AQW,*VD,*AC
MOV_DW EN IN OUT	MOVD IN,OUT	EN girişi "1" olunca IN girişindeki doubleword değeri OUT çıkışına aktarır. Giriş doubleword değeri değişikliğe uğramaz.	<u>IN</u> : VD,ID,QD,MD,SMD, (dword) AC,HC,sabit,*VD,*AC &VB,&IB,&QB,&MB,&T,&C <u>OUT</u> : VD,ID,QD,MD, (dword) SMD,AC,*VD,*AC
MOV_R EN IN OUT	MOVR IN,OUT	EN girişi "1" olunca IN girişindeki 32 bitlik reel sayıyı OUT doubleword çıkışına aktarır. Giriş reel sayısı bu işlemle değişikliğe uğramaz.	<u>IN</u> : VD,ID,QD,MD, SMD, AC,Sabit, *VD,*AC, SD <u>OUT</u> : VD,ID,QD,MD, SMD, AC,*VD,*AC,SD
SWAP EN IN OUT	SWAP IN	EN girişi "1" olunca IN girişindeki word değerinin düşük anlamlı byte'ıyla yüksek anlamlı byte'ının yerlerini değiştirir.	<u>IN</u> : VW,T,C,IW,QW,MW (word) SMW,AC,*VD,*AC
BLKMOV_B EN IN N OUT	BMB IN,OUT,N	EN girişi "1" olunca IN girişindeki byte değeri ile bundan sonraki N tane byte değeri OUT çıkışındaki byte değerinden başlayarak N tane çıkışa kopyalar. (N=1...255)	<u>IN</u> : VB,IB,QB,MB,SMB, (byte) *VD,*AC <u>OUT</u> : VB,IB,QB,MB,SMB, (byte) AC,*VD,*AC <u>N</u> : VB,IB,QB,MB,SMB, (byte) AC,sabit,*VD,*AC
BLKMOV_W EN IN N OUT	BMW IN,OUT,N	EN girişi "1" olunca IN girişindeki word değeri ile bundan sonraki N tane word değeri OUT çıkışındaki word değerinden başlayarak N tane çıkışa kopyalar. (N = 1...255)	<u>IN</u> : VW,T,C,IW,QW,MW (word) SMW,AIW,*VD,*AC <u>OUT</u> : VW,T,C,IW,QW,MW (word) SMW,AQW,*VD,*AC <u>N</u> : VB,IB,QB,MB,SMB, (byte) AC,sabit,*VD,*AC
FILL_N EN IN N OUT	FILL IN,OUT,N	EN girişi "1" olunca OUT çıkışındaki word değerinden başlayarak N tane çıkış word'ünün hepsine IN girişindeki word değerini kopyalar.	<u>IN</u> : VW,T,C,IW,QW,MW,SMW (word) AIW,sabit, *VD,*AC <u>OUT</u> : VW,T,C,IW,QW,MW (word) SMW,AQW,*VD,*AC <u>N</u> : VB,IB,QB,MB,SMB, (byte) AC,sabit,*VD,*AC

(alıcı/verici), 7 yüksek hızlı sayıcı kesmesi (HSC0 için $CV=P_V$, HSC1 ve HSC2 için yön değişimi, dış reset, $CV=P_V$), 2 darbe katar çıkışı (PTO) kesmesi (PTO0, PTO1)

Kesme işlemi başlatılmadan yığının üst seviyesi “1” geri kalanlar “0” olarak düzenlenir. Bobin veya işlem kutuları kontak kullanmadan doğrudan sol çizgiye bağlanabilir. Deyim listesinde ise LD komutu kullanılmayabilir. Kesme üstünlük sırası: haberleşme, I/O kesmeleri (HSC ve darbe katar çıkış kesmeleri dahil) ve zamana bağlı kesme’dir.

Aynı üstünlük sırasındaki kesme işlemlerinde ilk gelen kesme ilk yapılır mantığı mevcuttur. Aynı anda iki kesme altprogramı işletilemez. Bir kesme işletilirken başka bir kesme olayı ortaya çıktığında, sırada bekler ve süregelen kesme işlemi bittiğinde işletilir. Genel olarak; sıranın tutabileceğinden daha fazla kesme oluşabilir. Bu durumda sıra taşma bellek bitleri (kaybedilen kesme tipine göre) sistemde düzenlenir.

Kontak, bobin ve akümülatör lojigi kesmelerden etkilendiğinden sistem lojik yığını, akümülatörü, akümülatör ve komut işletim durumunu belirten özel bellek bitlerini (SM) saklar ve kesme işlemi bitiminde tekrar yükler. Böylece kesme altprogramının yürütülmesi sebebiyle ana programın karışıklığa uğraması önlenir.

PLC çalışma moduna alındığında, kesmeler yetkilendirilmemiştir, “ENI - Bütün kesmeler etkin” komutu ile tüm kesmelerin işletimine izin verilebilir. “DISI - Bütün kesmeler etkin değil” komutu işletildikten sonra oluşan kesmeler sıraya sokulur, fakat kesme işleminin gerçekleşmesine izin verilmez.

Kesme işlemini gerçekleştirmeden önce kesme olayı ile bu olay gerçekleştiğinde işletilmesi istenen kesme altprogramı arasında ilişki oluşturmak gereklidir. Bunun için kesme olayı numarası ve kesme altprogramı etiket numarası ATTACH komutu ile ilişkilendirilir. Birden çok kesme olayı bir kesme altprogramı ile ilişkilendirilebilir, fakat bir kesme olayı birden fazla kesme altprogramı ile ilişkilendirilemez. Böyle bir durumda, bir kesme olayı oluştuğunda sadece en son ilişkilendirilen kesme altprogramı işletilir. Bir kesme olayı ile kesme altprogramı ilişkilendirildiğinde, o kesme otomatik olarak yetkilendirilir. Eğer “DISI” komutu ile tüm kesmelerin işletilmesine son verilirse oluşan her kesme sıraya girer ve tekrar “ENI” komutu ile işletilmesine izin verilene dek sırada bekler. DETACH komutu ile; kesme olayı ve kesme altprogramı arasındaki ilişki bozularak kesmelerin etkinlikleri sona erer.

Her kesme altprogramı bir etiketle tanıtılır. Kesme altprogramı, program etiketi ile koşulsuz sonlandırma komutu arasında yazılır. Programdan koşullu sonlandırma ile de çıkılabilir. Fakat koşulsuz sonlandırma daima gereklidir. Kesme altprogramlarının mümkün olduğu kadar kısa olması, ana programın işletilmesini aksatmaması bakımından gereklidir. Eğer bu yapılmaz ise ana programda istenilmeyen aksamalar doğurabilir. En kısa en iyidir mantığı geçerlidir. Kesme programları ana programın bitiminden sonra yazılır. Kesme altprogramı içinde DISI, ENI, CALL, HDEF, FOR/NET ve END komutları kullanılamaz.

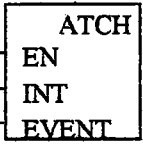
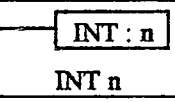
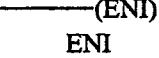
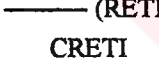
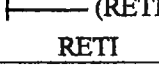
5.11.1 ZAMANA BAĞLI KESMELER;

CPU 212'de 1 adet CPU 214'de 2 adet zamana bağlı kesme mevcuttur. Belli zaman aralığında bir işlem tekrarlanmak istenirse, zamana bağlı kesme kullanılabilir. Altprogramın işletilme süresi olarak 5ms - 255ms. arasında tamsayı değerler verilebilir. Bu işletilme süresini "0. zamana bağlı kesme" için SMB34'e , "1. zamana bağlı kesme" için SMB35'e yazmak gereklidir. Genellikle zamana bağlı kesmeler analog girişlerin belli aralıklarda örneklenmesini sağlamak için kullanılır.

Bir kesme altprogramı zamana bağlı kesme olayı ile ilişkilendiğinde zamana bağlı kesme etkin olur. Altprogramın işletilme süresini değiştirmek için önce SMB34 ve SMB35 deki değerin düzeltilmesi, ardından kesme altprogramının zamana bağlı kesme olayı ile tekrar ilişkilendirilmesi gereklidir. Tekrar ilişkilendirme sırasında zamana bağlı kesme fonksiyonu önceki ilişkilendirmeden kalan zaman değerlerinin tümünü siler ve sistem yeni değer ile kesme altprogramını işletmeye başlar.

Zamana bağlı kesme yetkilendirildiğinde belirli zaman aralıklarında ilişkilendirilmiş kesme altprogramını sürekli olarak işletir. "Çalışma" modundan çıkılırsa yada DETACH komutu kullanılırsa zamana bağlı kesme altprogramının işletimi sona erdirilir. Eğer "ENI" komutu işletilirse zamana bağlı kesmeler yine yetkilendirilir. Zamana bağlı kesmelerin işletimine izin verilmeyen süre boyunca oluşan zamana bağlı kesmeler sıra dolana yada kesmelerin işletimine izin verilene kadar sıraya girer. 0. ve 1. zamana bağlı kesmelerin olay numaraları sırasıyla 10 ve 11'dir. Merdiven diyagramı ve deyim listesinde kesme altprogramı komutlarının gösterilimi ve açıklaması tablo 5.10'da yer almaktadır.

Tablo 5.10 Kesme komutları

KOMUT(LAD/STL)	TANIM	OPERANDLAR
 ATCH INT,EVENT	ATTACH Kesme ilişkilendirme komutu EVENT girişindeki kodlanmış kesme olayı ile INT girişindeki kodlanmış kesme altprogramını ilişkilendirir ve kesme olayını yetkilendirir.	INT (CPU 212): 0 - 31 (byte) (CPU 214): 0 - 127 EVENT (CPU 212): 0,1,8-10,12 (byte) (CPU 214): 0 - 20
 DTCH EVENT	DTCH kesme ilişkisini koparma komutu bir kesme olayının (EVENT) diğer bütün kesme programları ile ilişkisini keser ve kesme olayının işletilmesine son verir.	EVENT (CPU 212): 0,1,8-10,12 (byte) (CPU 214): 0 - 20
 INT n	"n" Kesme altprogramının başlangıcını belirtir.	n: (CPU 212) : 0 - 31 (word) (CPU 214) : 0 - 127
 ENI	Bütün ilişkilenmiş kesmelerin işletimine izin verilir.	yok
 DISI	Bütün ilişkilenmiş kesmelerinin işletimine son verilir.	yok
 CRETI	Bir önceki lojik işlem sonucuna göre koşullu olarak kesmeden dönüşü sağlar.	yok
 RETI	Koşulsuz kesmeden dönüşü sağlar. Her kesmenin sonuna konmalıdır.	yok

5.12 DÖNÜŞTÜRME İŞLEMLERİ

Verileri bir formattan başka bir formata dönüştürmeyi sağlar.

ATH IN,OUT,LEN : ASCII'den 16'lı sisteme dönüştürür.

HTA IN,OUT,LEN : 16'lı sistem'den ASCII'ye dönüştürür.

SEG IN,OUT : 16'lı sistemde yazılı olan karakteri 7 segmentli display için dönüştürür.

BCD I OUT : İkili-onlu kodlu sayıyı tamsayıya çevirir.

I BCD OUT : Tamsayıyı ikili-onlu kodlu sayıya çevirir.

DI_REAL IN,OUT : IN girişindeki 32 bitlik işaretli tamsayıyı 32 bitlik reel sayı olarak OUT çıkışına yazar.

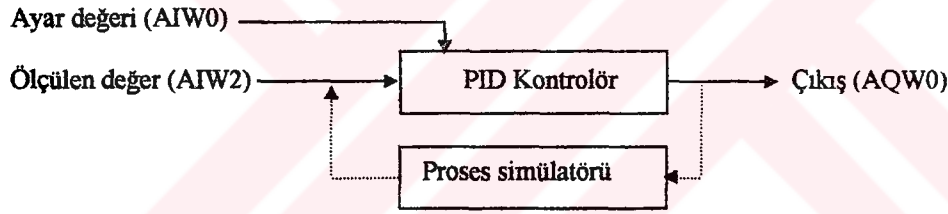
TRUNCH IN,OUT : IN girişindeki 32 bitlik reel sayıyı 32 bitlik işaretli tamsayı olarak OUT çıkışına yazar.

BÖLÜM 6

KONTROLÖR VE KONTROL EDİLEN SİSTEM

6.1 PLC PROGRAMI

PID kontrolör, çıkış değerini, verilen ayar değerine getirmeye çalışır. Her bir örnekleme periyodunda oransal, integral ve türev olmak üzere üç terimden oluşan çıkış değerini hesaplar. İstenen örnekleme zamanında bu hesapları yapılabilmesi için PLC’de zamanlanmış kesme oluşturulur. PID kontrolörünün gerçekleştirilmesi aşağıdaki gibidir.



Şekil 6.1 PID kontrolörünün gerçekleştirilmesi

PID kontrolün oransal terimi, hesaplanan hata değeri ile verilen kazancın çarpımından ibarettir:

$$m_p[kT] = K_c e[kT] \quad (6.1)$$

$$M_{Pn} = K_c \cdot e \quad (6.2)$$

İleri dikdörtgen iterasyonu ile elde edilmiş (3.9) integral kontrolör denklemi aşağıdaki şekilde düzenlenirse;

$$G_I(z) = \frac{M_I(z)}{E(z)} = \frac{K_I T}{1 - z^{-1}} \quad (6.3)$$

$$M_I(z) (1 - z^{-1}) = K_I T E(z) \quad (6.4)$$

$$m_i[kT] - m_i[(k-1)T] = K_I T e[kT] \quad (6.5)$$

$$m_i[kT] = (K_c T / T_I) e[kT] + m_i[(k-1)T] \quad \text{ve} \quad K_I = K_c / T_I \quad (6.6)$$

İntegral terimi sürekli zamanda $e(t)$ hata işaretinin integrali alınarak bulunurken, ayrık zamanda hatanın toplamı ile elde edilir. İntegral teriminin etkisini belirleyen sabit T_I integral zamanıdır, integral teriminin etkide bulunması istenmezse T_I değeri olabildiğince büyük seçilmelidir.

$$M_{In} = K_c T / T_I \cdot \sum_1^n e + M_{ilk} = K_I \cdot T \cdot e_n + M_{n-1} \quad (6.7)$$

Denklem (3.3)'te verilen türev kontrolörü dijital olarak gerçekleştirilmesi için aşağıdaki şekilde düzenlenir:

$$G_D(z) = \frac{M_D(z)}{E(z)} = \frac{K_D}{T} (1 - z^{-1}) \quad (6.8)$$

$$M_D(z) = (K_D / T) (1 - z^{-1}) E(z) \quad (6.9)$$

$$m_D[kT] = (K_D / T) \{ e[kT] - e[(k-1)T] \} \quad (6.10)$$

$$m_D[kT] = (K_c T_D / T) \{ e[kT] - e[(k-1)T] \} \quad \text{ve} \quad K_D = K_c T_D \quad (6.11)$$

Türev terimi sürekli zamanda $e(t)$ hata işaretinin türevi alınarak bulunurken, ayrık zamanda hata $e(kT)$ ile bir önceki hatanın $e[(k-1)T]$ farkı alınarak elde edilir. Türev teriminin etkisini belirleyen sabit T_D türev zamanıdır ve T_D sıfır seçilerek türev etkisi kaldırılabilir:

$$M_{Dn} = (K_c \cdot T_D / T) (e_n - e_{n-1}) = (K_D / T) (e_n - e_{n-1}) \quad (6.12)$$

PID kontrolörünün çıkışı oransal, integral ve türevsel kontrolörlerin toplamı olduğundan,

$$\text{Çıkış} = M_{Pn} + M_{In} + M_{Dn} \quad (6.13)$$

Ayar değerinden ölçülen değer çıkarılarak hata bulunur ve yukarıdaki üç terim ayrı ayrı hesaplanarak çıkış değeri saptanır. Geriye integral yığılması problemini ortadan kaldırmak kalır. Bunun için şu yöntem kullanılır;

Eğer kontrolör çıkışı "1" den büyükse $M_{n-1} = 1 - (M_P + M_D)$ olarak düzeltilir.

Eğer kontrolör çıkışı "-1" den küçükse $M_{n-1} = -1 - (M_P + M_D)$ olarak düzeltilir.

Bir başka deyişle; çıkış “1” den büyükse M_{n-1} , M_P , M_D toplamı “1” olacak şekilde M_{n-1} ayarlanır. çıkış “-1” den küçükse M_{n-1} , M_P , M_D toplamı “-1” olacak şekilde M_{n-1} ayarlanır. Böylece M_{n-1} “-1” ve “1” değeri arasında sınırlandırılmış olur.

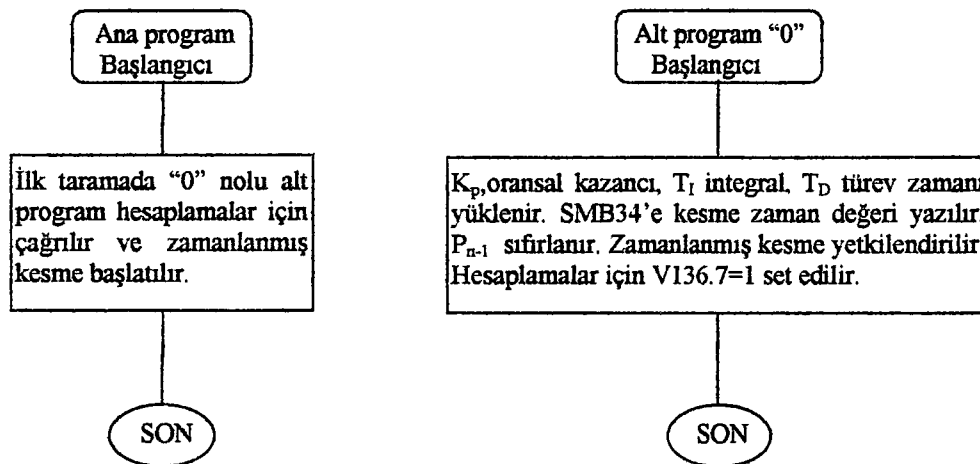
PLC programı bir alt program ve bir zamanlanmış kesmeden ibarettir. Ana program sadece ilk taramada zamanlanmış kesmenin işletimine izin verilmesi, belleğin silinmesi gibi bir takım başlangıç işlemlerini yapmak üzere alt programı çağırır. Bundan sonra sadece zamanlanmış kesme her T zamanında çalışır.

PID programında kullanılan V-hafızalar ve bunların hangi bilgileri saklamak için kullanıldığı aşağıda belirtilmiştir.

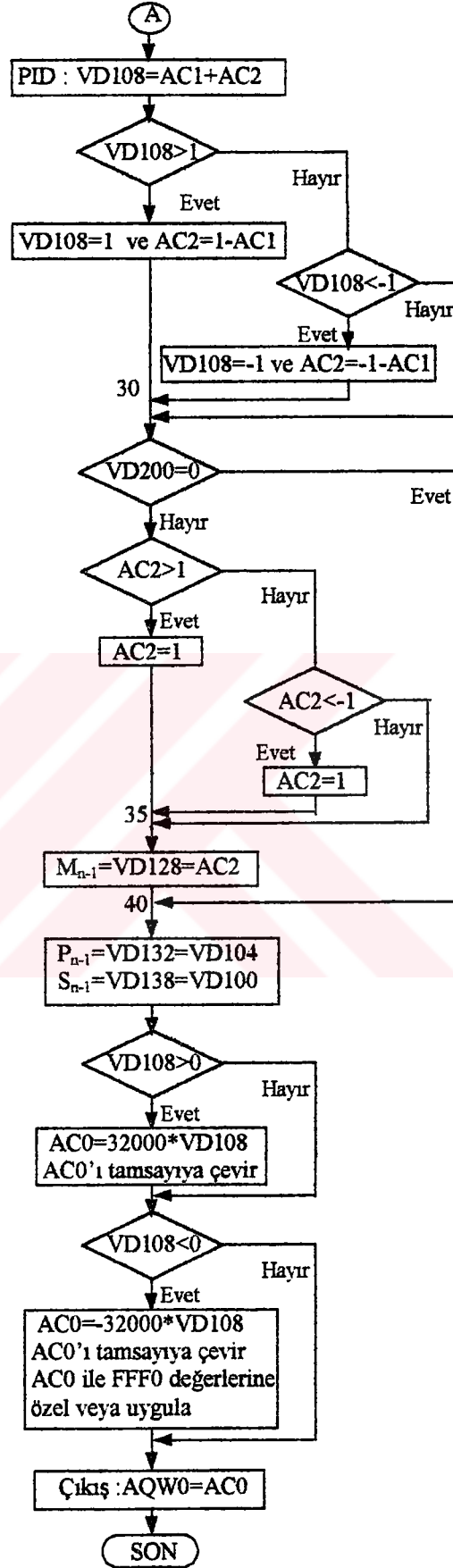
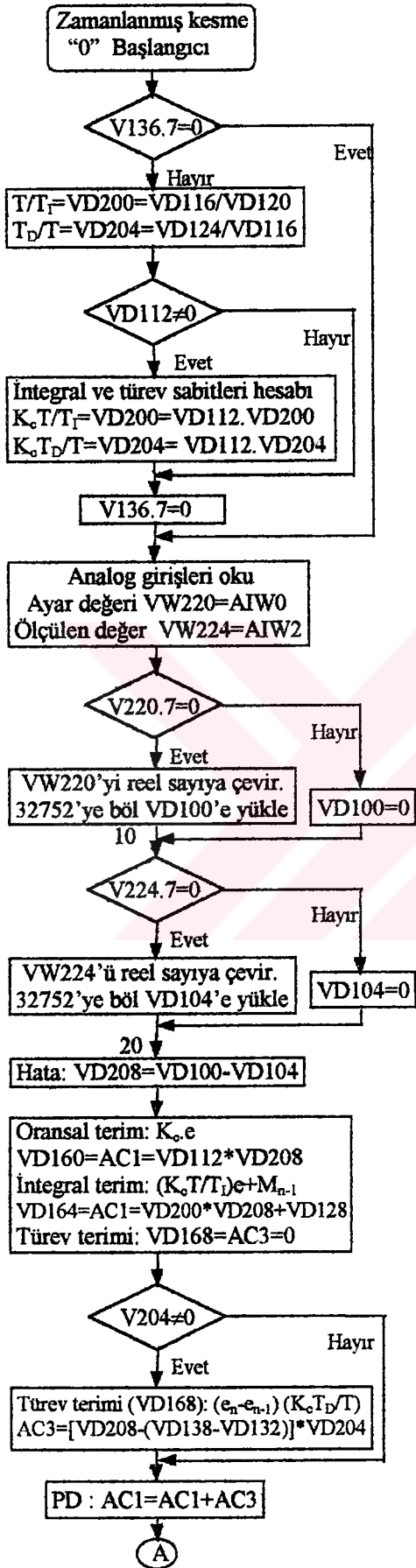
Tablo 6.1 PLC’de hesaplamaların yapıldığı veri alanları

V-hafıza	İçeriği	V-hafıza	İçeriği
VD100	Ayar değeri	VD142	K_c oransal kazanç
VD104	Ölçülen değer	VD146	T örnekleme zamanı
VD108	Kontrolör çıkışı	VD150	T_I integral zamanı
VD112	K_c oransal kazanç	VD154	T_D türev zamanı
VD116	T örnekleme zamanı	VD160	Oransal terim
VD120	T_I integral zamanı	VD164	İntegral terim
VD124	T_D türev zamanı	VD168	Türev terimi
VD128	Bias M_{n-1}	VD200	$K_c T / T_I$
VD132	Bir önceki ölçülen değer P_{n-1}	VD204	$K_c T_D / T$
VD138	Bir önceki ayar değeri S_{n-1}	VD208	hata

PID Programın ana program, alt program ve zamanlanmış kesme programı akış diyagramları ise şu şekildedir.



Şekil 6.2 Ana program ve alt programın akış şeması



Şekil 6.3

6.2 KONTROL EDİLEN SİSTEM

Proses kontrol simülatörü PCS 327, proses kontrol yöntemlerinin uygulanmasında kullanılmak üzere işlemsel kuvvetlendiricilerden oluşan cihazdır, kontrolör ve proses olmak üzere iki kısım mevcuttur. PCS327 Proses kontrol simülatörünün yavaş (1 s.) ve hızlı (10 ms.) olmak üzere iki çalışma hızı mevcuttur. Proses yada kontrolör dış cihazlar ile kullanılmak istendiğinde yavaş (slow) konumda çalışmak en uygunudur, bu konum kullanıcıya zaman sabitini ayarlama imkanı tanır. Yavaş konumunda; hızlı konumunda devrede olan kondansatöre paralel kondansatör eklenir. Bu ek kondansatörler lehim çubuğundan kolaylıkla küçük yada büyük değerleri ile değiştirilebilir.

Lineer olmayan fonksiyonlarıyla beraber çeşitli kontrolör yapılarının uygulanmasına izin veren bu prosesde, oransal, integral, türevsel, iki ve üç basamaklı kontrol modlarında çalışma yapılabilir. Simülatör işaretleri osiloskop ile gözlenebilir. Giriş ve çıkış işaretleri $\pm 10V$ 'dur.

Genel Özellikler

Giriş ve çıkış değer aralığı	$\pm 10V$
Çıkışta tam skala için minimum direnç yükü	5k Ω
Besleme gerilimi	220-250V 50/60Hz. yada 110-120V
İç besleme gerilimi	$\pm 15V$

Proses

Geciktirme zaman sabiti	10ms.(hızlı) 1s.(yavaş)
İntegral zaman sabiti	10ms.(hızlı) 1s.(yavaş)
Transfer geciktirmesi	10ms.(hızlı) 1s.(yavaş)

Kontrolör

Referans değer aralığı	$\pm 10V$
İntegral zamanı değer aralığı	250-5ms.(hızlı) 25-0.5s.(yavaş)
Türev zamanı değer aralığı	0-20ms.(hızlı) 0-2s.(yavaş)
Oransal bant değer aralığı	200% - 4%

Arabirim üniteleri

Gerilim - akım dönüştürücüsü (2 adet)	$(-10..+10)V \rightarrow (4..20)mA$
Akım - gerilim dönüştürücüsü (1 adet)	$(4..20)mA \rightarrow (+10..-10)V$

Bir basit geciktirme, anahtar ile integrale dönüştürülebilen iki adet geciktirme mevcuttur, bu elemanların zaman sabitleri yavaş/hızlı anahtarı ile 10ms. yada 1s.

olarak seçilebilir. Yine bir yavaş/hızlı anahtarı ile 10ms. yada 1s.'lik transfer geciktirmesi ve bir adet evirici vardır. Bu elemanlarla "0" , "1" ve "2" tiplerinde olmak üzere 12 değişik lineer proses karakteristikleri elde edilir. Her eleman polariteyi değiştirdiğinden gerektiğinde evirici kullanılır. Ek olarak giriş bozucusu ve yük bozucusu eklemek için iki adet toplama elemanı sağlanmıştır. Düzeltici eleman, motor elemanı, algılayıcı eleman, ölçme elemanı için donanımda hiçbir şey olmamasına rağmen uygulamada olması gereken yerleri belirtmek bakımından gösterilmiştir.

Kontrolör kısmında $\pm 10V$ aralığında ayarlanabilen referans, bir karşılaştırma elemanı, integral zamanı ayarlanan integral kontrolör, türev zamanı ayarlanan türevsel kontrolör, oransal bandı ayarlanan oransal kontrolör bulunmaktadır. Uygulamada lineer olmayan elemanlar çoktur. Bunları simüle etmek için tipik bazı lineer olmayan elemanlar prosese ilave edilmiştir

İki adet integral yada geciktirme olarak değiştirilebilen sistem hızlı konumunda $0.01 \mu F$ 'lık kondansatör ve geciktirme konumuna alındığında $1M\Omega$ 'luk direnç devrededir. Yavaş konumunda zaman sabiti 1s.'dir, bu değiştirilmek istenirse aşağıda hesaplanan değerdeki kondansatör ile değiştirilir:

$$C = \frac{T}{100} - 0.01\mu F \quad (6.14)$$

Örneğin $T=500ms.$ 'lik zaman sabiti elde etmek için $C=0.5-0.01=0.49\mu F$ 'lık kondansatör ile değiştirilmelidir.

Transfer geciktirmesinde, C3 ve C8 olmak üzere iki adet ek kondansatör vardır. $T=10 ms.$ 'lik gecikme için $0.047\mu F$ 'lık sabit kondansatör mevcuttur. $T ms.$ 'lik gecikme için eklenecek kondansatör aşağıda hesaplanan değerde olmalıdır.

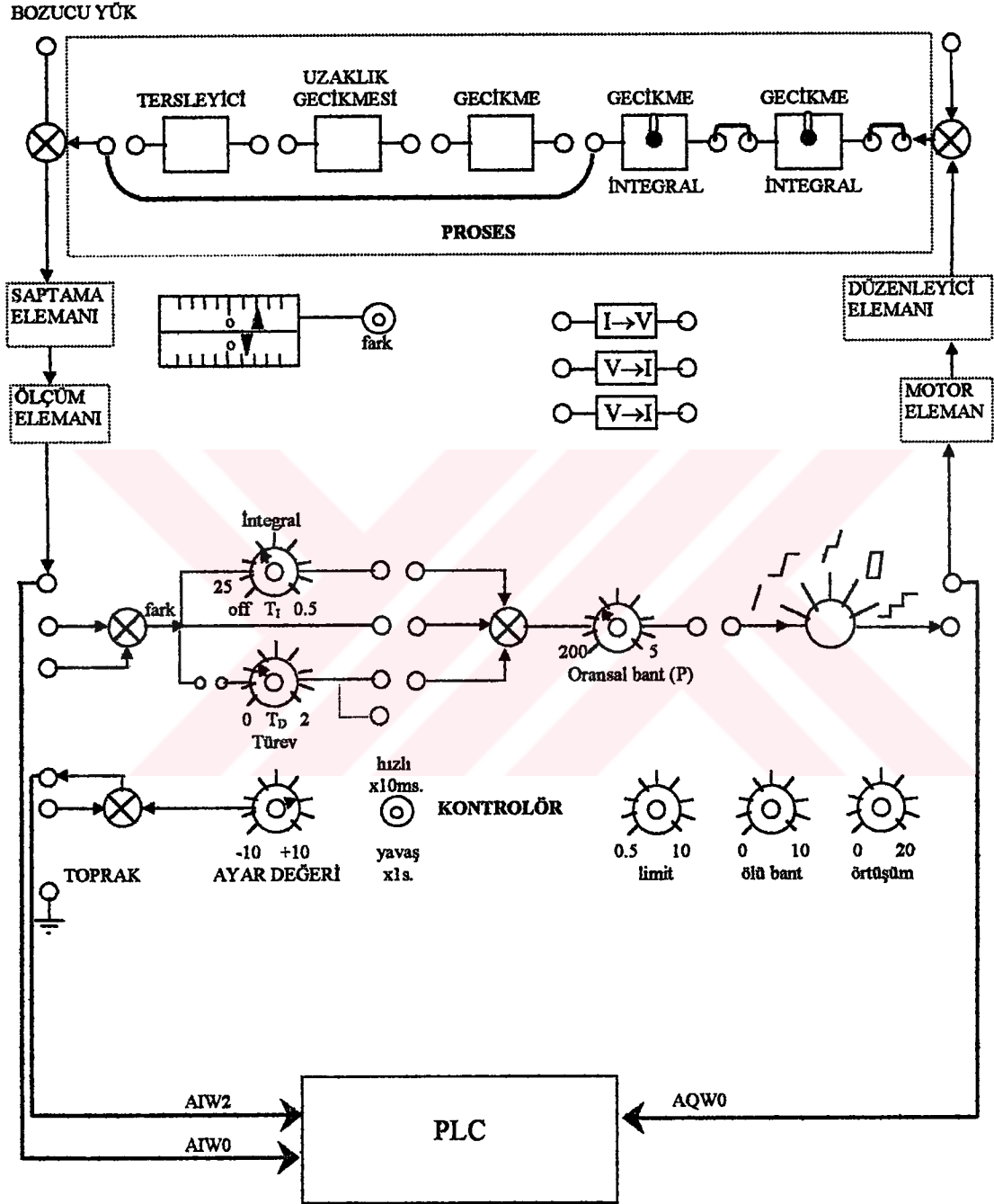
$$C = \frac{T}{213} - 0.047\mu F \quad (6.15)$$

Örneğin $T = 2000ms.$ 'lik gecikme elde etmek için $C = 9.389-0.047 = 9.342 \mu F$ 'lık kondansatör ile değiştirilmelidir.

Tezde iki adet geciktirme kullanılmıştır. Şu halde tezde kullanılan sistemin transfer fonksiyonu aşağıdaki gibidir.

$$G_p(s) = \frac{1}{(s+1)^2} \quad (6.16)$$

Denklem 6.16'daki sistemin PCS 327 proses kontrol simülöründe gerçekleştirilmesi ve kontrol edilen sistemin PLC'ye ne şekilde bağlandığı şekil 6.4'te gösterilmiştir.

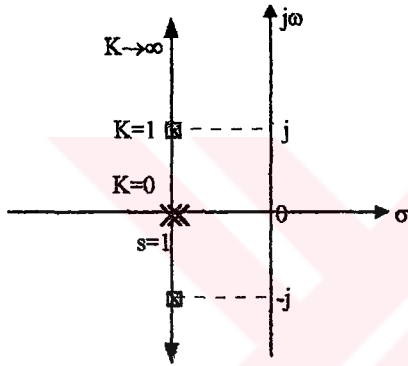


Şekil 6.4 PCS 327 Proses kontrol simülörü bağlantı şeması

6.3 PID PARAMETRELERİNİN HESAPLANMASI

PID parametrelerinin hesabı farklı şekillerde yapılabilir. Bunlardan birkaçına değineceğiz. Tasarım yöntemlerinden birisi sistemi küçük örnekleme zamanları için s - tanım bölgesinde düşünmek, tasarımı s - tanım bölgesinde yaparak parametreleri belirlemektir. Sistemin $s=1$ 'de iki adet kutbu vardır. Kontrolörsüz kapalı çevrim transfer fonksiyonu;

$$T(s) = \frac{G(s)}{1+G(s)} \quad G(s) = \frac{1}{(s+1)^2} \quad (6.17)$$



Kararlı hal hatası $s \rightarrow 0$ için $ess = 0.5$ 'dir.

Bu durumda sistem sadece K kazançlı oransal kontrolörle kontrol edilemez.

Sistem oldukça yavaş, ancak kararlıdır.

Şekil 6.5 Kontrol edilen sistemin kök eğrisi

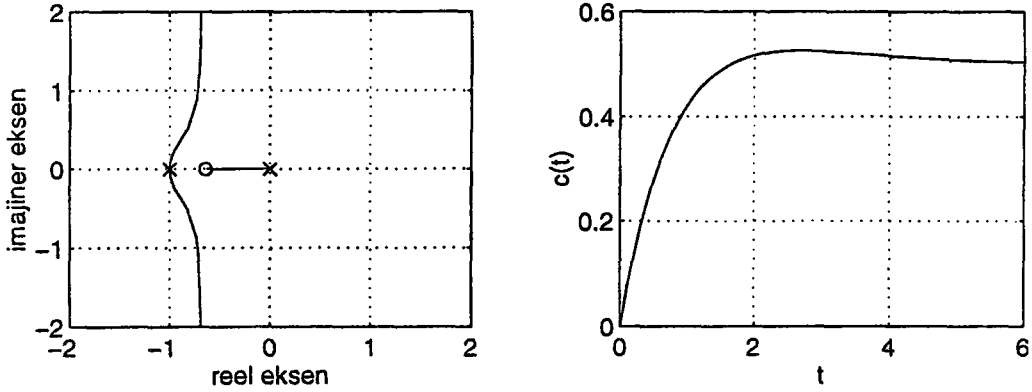
6.3.1 PI KONTROLÖR

PI kontrolörü s -tanım bölgesinde sisteme $z=0$ 'da kutup ve $s=K_I/K_p$ de bir sıfır ekler. Eklenecek sıfırın yeri ve kontrolör kazancının belirlenmesi gereklidir.

$$G_c(s) = \frac{K_p(s + K_I/K_p)}{s} \quad (6.18)$$

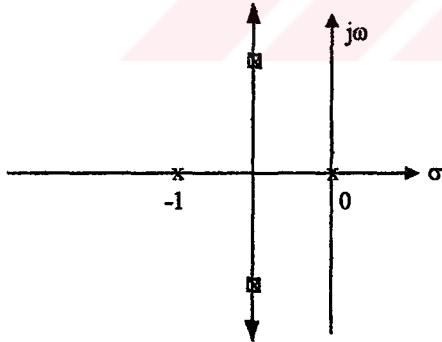
Kompanze edilmemiş sistemin kapalı çevrim kök eğrisinin değişmemesi istenirse $z = K_I/K_p$ reel ekseninde "0" ve "1" arasında seçilir ve K_p kontrolör kazancı ile kapalı çevrim kutupların yerleri belirlenir. Fakat bu sistemin davranışı reel eksenindeki kapalı çevrim kutbu yüzünden oldukça yavaş olacaktır.

Örneğin $z = K_I/K_c = 0.64$ seçilsin. $K_I = 0.64 \cdot K_p$ olacaktır. K_p değerinin artırılmasıyla kapalı çevrim kutupları uzaklaşır. $K_p = 1.1$ ve $K_I = 0.7$ için sistemin kök eğrisi ve kapalı çevrim kutupları şekil 6.6'daki gibi olur.



Şekil 6.6 PI kontrolörlü sistemin kök eğrisi ve sistemin birim basamak cevabı

Kontrolör sıfırı $s=1$ 'deki sistemin kutbunun birini silecek şekilde seçilirse kök eğrisi şekil 6.7'deki gibi olur. K_p kontrolör kazancı ile kapalı çevrim kutupların yerleri belirlenir. Fakat bu sistemin davranışı bir öncekinden daha hızlı olacaktır.



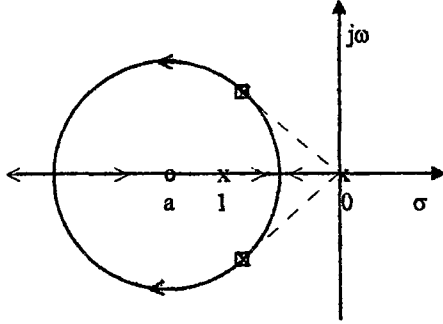
PI Kontrolör sıfırı $z=K_I/K_c=1$ seçilir. Bu durumda $s=-1$ 'deki katlı kutbun biri silinir. $K_I = K_c$ olacaktır. K_c değerinin arttırılması ile kapalı çevrim kutupları uzaklaşır. En uygun çözüm $K_I=K_c=0.8$ 'dir. Aşım %12, yerleşme zamanı 3.2 sn.'dir.

Şekil 6.7 Kutup yok etmeli PI kontrolörlü sistemin kök eğrisi

6.3.2 PID KONTROLÖR

Kontrolör sıfırı $s=1$ 'den daha uzağa yerleştirilirse, kök eğrisi aşağıdaki gibi olur. K_D kontrolör kazancı ile kapalı çevrim kutupların yerleri belirlenir. Fakat bu sistemin davranışı önceki iki sistemden daha hızlı olacaktır.

$$G_c(s) = \frac{K_D \{s^2 + (K_p/K_D)s + K_I/K_D\}}{s} \quad (6.19)$$



PID Kontrolör sıfırları $z = 1$ ve $z = a$ olarak seçilir. K_D değerinin artırılmasıyla kapalı çevrim kutupları sıfıra doğru ilerler. Bazı K_p, K_I, K_D parametreleri için sistemin birim basamak cevabı aşağıdaki gibidir.

Şekil 6.8 PID kontrolörlü sistemin kök eğrisi

PID kontrolör için yapılan deneyler sonucu aşağıda verilmiştir.

K_c	K_I	K_D	T_I	T_D	% aşım	t_s
0.45	0.6923	0.045	0.65	0.10	0	2.8
0.20	0.4	0.02	0.50	0.10	0	7
0.45	0.75	0.045	0.60	0.10	2	2.4
0.35	0.875	0.07	0.40	0.20	4	1.8
0.5	0.8333	0.05	0.60	0.10	4	2.2
0.45	0.8182	0.045	0.55	0.10	4	2.1
0.40	1.143	0.08	0.35	0.20	5	1.7

SONUÇLAR VE ÖNERİLER

Oransal, integral, ve türevsel kontrolörün toplamı olan PID kontrolörler, proses kontrolde çok yaygın kullanılmaktadır. Oransal kontrol, sistemin hatasını bir K sabiti ile çarparak kontrol işareti üretir ve mutlaka bir miktar kararlı hal hatası kalır. İntegral kontrol hatanın integralini aldığından kararlı hal hatasını yok eder. Türevsel kontrol ise, sistemin geçici cevabını düzeltir. Kontrol edilmek istenen sistem için kazanç, integral ve türev zamanı parametreleri bulunarak, kontrolü en iyi şekilde sağlanır.

Günümüzde PLC'lerin komut işleme ve dolayısıyla tarama hızı oldukça artmıştır. Bu nedenle tek bir PLC ile çok kompleks bir sistemin kontrolü yapılabilmektedir. Kontaklı kumanda devreleri ile birlikte PID gibi yazılımla gerçekleştirilebilen her türlü kontrolör mümkün olabilmektedir, yalnız PLC'nin giriş ve çıkışlarının sınırlı olması bazı problemler yaratır. PLC çıkışı ancak +10V -10V arasında olabilir, hesaplanan PID kontrolör çıkışı bu değerden yüksek olsa dahi sisteme verilemez.

Kontrolörün prostedeki değışiklere hızlı cevap verebilmesi için integral yığılması oluşmasının önlenmesi gereklidir. Bunun için çıkış sürekli kontrol edilir ve sınırlara erişildiğinde daha fazla yükselmesi ya da azalması engellenir. Bir diğer nokta ise, referans değerin maksimum ölçeğin yarısı dolaylarında seçilmesi ile integral yığılmasının bir ölçüde önüne geçilmesi sağlanmış olur.

KAYNAKLAR

- [1] CLOSED-LOOP CONTROL WITH SIMATIC S5 Basic Concepts, Siemens, 1989.
- [2] CHENG Chi-Tsong, Analog & Digital Control System Designs Transfer Functions State-space and Algebraic methods, Sounders College Publishing, 1993.
- [3] SHAHLIAN B., MICHAEL H., Control System Design Using Matlab, Prentice Hall, 1993.
- [4] BENJAMIN C.K., Digital Control Systems, Second Edition, , Sounders College Publishing, 1992.
- [5] SIMATIC S7-200 Tips Master Listing of Application Notes, SIEMENS A.G., 1996.
- [6] AKOĞLU Köksal, Temel Otomatik Kontrol Bilgileri, 4. Ulusal Otomatik Kontrol Sempozyumu, 11-13 kas. 1987 İst., İTÜ Elek. Elektronik Fak. Baskı Atölyesi.
- [7] BENJAMIN C.K., Automatic Control System , 7. Edition Prentice-Hall International Editions.
- [8] ÜLGÜR Münir, Otomatik Kontrol Sistemlerinde Davranış Kriterleri, İTÜ Elek. Elektronik Fak. Ofset Baskı Atölyesi, 1987.
- [9] SARIOĞLU Kemal, Otomatik Kontrol, İTÜ Elek. Elektronik Fak. Ofset Baskı Atölyesi, 1992.
- [10] FRANKLIN Gene.F, Digital Control Of Dynamic Systems, Addison-Wesley Publishing Company Inc., 1980.
- [11] SIMATIC STEP-7 Micro Programming Reference Manual, Siemens A.G., 1995.

EKLER

PID PROGRAMI

```
NETWORK //ANA PROGRAM
LD SM0.1
CALL 0
```

```
NETWORK
MEND
```

```
NETWORK //ALT PROGRAM
SBR 0
```

```
NETWORK //SPn,Kc,Ts,Ti,Td yüklenir,zamanlayıcı kesme aktiflenir.
LD SM0.1
FILL 0, VW100, 60
FILL 0, VW200, 60
MOVR 0.1000000, VD142
MOVR 0.05000000, VD146
MOVR 20.00000, VD150
MOVR 0.0000000, VD154
MOVB 50, SMB34
ATCH 0, 10
MOVR 0.0000000, VD132
S V136.7, 1
ENI
```

```
NETWORK
RET
```

```
NETWORK //ZAMANLAYICI KESMESİ
INT 0
```

```
NETWORK
LDN V136.7
JMP 0
```

```
NETWORK
LD SM0.0
MOVR VD142, VD112
MOVR VD146, VD116
MOVR VD150, VD120
MOVR VD154, VD124
MOVR VD116, VD200
/R VD120, VD200
MOVR VD124, VD204
```

```
/R VD116, VD204  
AR= 0.0000000, VD112  
NOT  
*R VD112, VD200  
*R VD112, VD204
```

```
NETWORK  
LD SM0.0  
R V136.7, 1
```

```
NETWORK  
LBL 0
```

```
NETWORK //Analog Giriş Değerlerini Okuma  
LD SM0.0  
MOVW AIW0, VW220  
MOVW AIW2, VW224
```

```
NETWORK  
LD V220.7  
MOVR 0.0000000, VD100  
JMP 10
```

```
NETWORK  
LDN V220.7  
XORD AC0, AC0  
MOVW VW220, AC0  
DTR AC0, AC0  
MOVR AC0, VD100  
/R 32752.00, VD100
```

```
NETWORK  
LBL 10
```

```
NETWORK  
LD V224.7  
MOVR 0.0000000, VD104  
JMP 20
```

```
NETWORK  
LDN V224.7  
XORD AC0, AC0  
MOVW VW224, AC0  
DTR AC0, AC0  
MOVR AC0, VD104  
/R 32752.00, VD104
```

```
NETWORK  
LBL 20
```

```
NETWORK  
LD SM0.0  
MOVR VD100, VD208  
-R VD104, VD208
```

```

MOVR VD112, AC1
*R   VD208, AC1
MOVR AC1, VD160
MOVR VD200, AC2
*R   VD208, AC2
+R   VD128, AC2
MOVR AC2, VD164
MOVR 0.0000000, AC3
MOVR 0.0000000, VD168

```

```

NETWORK
LDR= VD204, 0.0000000
NOT
MOVR VD138, AC3
-R   VD132, AC3
*R   -1.000000, AC3
+R   VD208, AC3
*R   VD204, AC3
MOVR AC3, VD168

```

```

NETWORK
LD   SM0.0
+R   AC3, AC1
MOVR AC1, VD108
+R   AC2, VD108
MOVR VD108, VD172

```

```

NETWORK
LDR<= VD108, 1.000000
NOT
MOVR 1.000000, VD108
MOVR 1.000000, AC2
-R   AC1, AC2
JMP  30

```

```

NETWORK
LDR>= VD108, -1.000000
NOT
MOVR -1.000000, VD108
MOVR -1.000000, AC2
-R   AC1, AC2

```

```

NETWORK
LBL  30

```

```

NETWORK
LDR= VD200, 0.0000000
JMP  40

```

```

NETWORK
LDR<= AC2, 1.000000
NOT
MOVR 1.000000, AC2
JMP  35

```

NETWORK
LDR>= AC2, -1.000000
NOT
MOVR -1.000000, AC2

NETWORK
LBL 35

NETWORK
LD SM0.0
MOVR AC2, VD128

NETWORK
LBL 40

NETWORK
LD SM0.0
MOVR VD104, VD132
MOVR VD100, VD138

NETWORK
LDR>= VD108, 0.0000000
MOVR 32000.00, AC0
*R VD108, AC0
TRUNC AC0, AC0

NETWORK
LDR>= VD108, 0.0000000
NOT
MOVR -32000.00, AC0
*R VD108, AC0
TRUNC AC0, AC0
XORD 16#FFF0, AC0

NETWORK // Analog çıkışı yazma
LD SM0.0
MOVW AC0, AQW0

NETWORK
RETI