

**İSTANBUL TEKNİK ÜNİVERSİTESİ ★ ELEKTRİK -ELEKTRONİK FAKÜLTESİ**

**KONTROL KUTUPLARI İLE KARAKTERİZE EDİLEN PROSESLERDE  
MİKROBİLGİSAYAR İLE  
ADAPTİF KONTROL**

**( DOKTORA TEZİ )**

**Y.Müh. Füsun TUNALI**

**Tezin Enstitüye Verildiği Tarih : 22. Mart.1983**

**Tez Savunmasının Yapıldığı Tarih : 2. Ağustos.1983**

**Doktorayı Yöneten Öğretim Üyesi : Prof. M. Münir ÜLGÜR**

**Diğer Jüri Üyeleri : Prof. Dr. Nadir YÜCEL**

**: Doç Dr. Atilla BİR**

**İSTANBUL TEKNİK ÜNİVERSİTESİ  
ELEKTRİK-ELEKTRONİK FAKÜLTESİ  
OFSET BASKI ATÖLYESİ**

**EKİM, 1983**

**Y. G.  
Yükseköğretim Kurulu  
Dokümantasyon Merkezi**

2156

Tez çalışmalarım süresince değerli destek ve katkılarını  
esirgemeyen hocam ve yöneticim Sayın Prof. M. Münir ÜLGÜR 'e ve  
tezimin araştırma ve uygulama aşamalarında son derece yararlı  
teşvik ve yardımlarından dolayı Sayın Doç.Dr. Eşref ADALI'ya  
teşekkürü bir borç bilirim.

Füsun TUNALI

## İÇİNDEKİLER

	Özet	III
	Summary	VI
	Semboller	XII
I. BÖLÜM	GİRİŞ	1
2. BÖLÜM	YÖNTEMİN TANITILMASI	10
2.1	Tanımlar	10
2.2	İkinci dereceden sistemler için zaman domaini ve frekans domaini kriterleri	15
2.2.1	Zaman domaini	16
2.2.2	Frekans domaini	19
2.2.3	Zaman domaini kriterleri ile frekans domaini kriterleri arasındaki ilişkiler	21
2.2.4	Sonuçlar	23
2.2.5	Test sinyali	24
2.3	Adaptasyon algoritması	25
2.3.1	Kontrolör	26
2.3.2	Adaptasyon çevrimi	28
2.3.2.1	Giriş aşaması	29
2.3.2.2	Belirleme aşaması	30
2.3.2.3	Karar verme aşaması	31
2.3.2.4	Değiştirme aşaması	33
3. BÖLÜM	SİMÜLASYON	41
3.1	Simülasyonda kullanılan bilgisayar sistemi.	41
3.2	Örnekleme zamanı ve dizge seçimi	43
3.3	Kontrolör Simülasyonu ve (P,I,D) etkilerine ilişkin deneyler	48
3.3.1	Kontrolör simülasyonu	48
3.3.2	(P,I,D) etkileri	50
3.4	Öğrenme aşaması için başlangıç artımlarının seçimi ve doğruluk deneyleri	53
3.4.1	$\Delta P_o$ seçimi	53
3.4.2	$\Delta I_o$ seçimi	55
3.4.3	$\Delta D_o$ seçimi	56

		II
3.5	Adaptasyon algoritmasına ilişkin simülasyon çalışmaları	58
3.5.1	Programın tanıtılması	58
3.5.1.1	Test alt programı	59
3.5.2	Simülasyon örneği (I)	60
3.5.3	Simülasyon örneği (II)	68
4. BÖLÜM	GERCEKLEME	75
4.1	Mikrobilgisayarın ve bilgi aktarma aygıtının seçimi	75
4.2	Mikrobilgisayar ve Bilgi Aktarma Aygıtı	77
4.2.1	Mikrobilgisayar	77
4.2.2	Bilgi Aktarma aygıtı	78
4.2.3	Bilgi aktarma protokolu	78
4.3	Kontrolör yazılımı	80
4.4	Bilgisayar aritmetiği	81
4.4.1	Bilgi aktarma için kullanılan veri kalıbı	82
4.4.2	P katsayısı ile ilgili işlemler	83
4.4.3	I katsayısı ile ilgili işlemler	87
4.4.4	D katsayısı ile ilgili işlemler	89
4.5	Örnekleme zamanının seçimi	91
4.6	Adaptasyon algoritması yazılımı	92
4.7	Deneyler ve sonuçları	92
4.8	Kullanılan donanım	96
5. BÖLÜM	SONUCLAR	97
6. BÖLÜM	KAYNAKLAR	98
7. BÖLÜM	EKLER	
	Özgeçmiş	

## MİKROBİLGİSAYARLA ADAPTİF KONTROL

## ( ÖZET )

Mikrobilgisayar ile adaptif kontrolü amaçlayan bu çalışmaya yön veren görüşler ;

- Günümüzde adaptif kontrol uygulamalarında amaç, endüstriyel alanda adaptif tekniklerin kolaylıkla kullanılabilirliğini sağlayacak basit ve ucuz yollar bulmaktır. Bu amaca hizmet edebilecek bir adaptasyon algoritmasının geliştirilmesi,
- Aynı zamanda bu algoritmanın, mikrobilgisayarın yazılım ve donanımına uygun olacak biçimde geliştirilmesi,
- Algoritmayı en uygun ve en etkin biçimde yorumlayacak ve hızlı çalışabilecek bilgisayar programının yazılması,
- Yine bu algoritmaya uygun düşecek mikrobilgisayar ve çevre birimlerinden kurulu donanımın oluşturulması,

biçiminde sıralanabilir.

Bu görüşlerden kalkarak, bu tezde gerçekleştirilen algoritmanın ana hatları aşağıdaki gibi verilebilir:

- Sistem kurulduğunda, kontrolör sistemi, kendisinden istenen davranışı verecek biçime getirecek yetenektir.
- Zamanla sistem parametrelerinin değişmesi ya da bozucu etkisi sonucunda değişen sistem davranışını sistemden istenen davranış biçimine getirmek üzere kontrolör parametreleri devamlı olarak ayarlanır.

- Sistemden istenen davranış, sistemin frekans band karakteristiği ya da buna eşdeğer olan zaman domenî davranışı ile karakterize edilebilmektedir. Sistemin davranışı ise uygun seçilecek bir test sinyaline cevabı izlenerek değerlendirilir ve bunun için dizge karakteristiklerinin tam olarak bilinmesine gerek yoktur.
- Bütün bu işlemleri yerine getirmek üzere bir mikrobilgisayar ve gerekli çevre donanımı kullanılmaktadır.

Algoritmanın oluşturulması için öncelikle, sistemin davranışını belirleyecek kriterin saptanması ve buna göre kontrolör seçiminin nasıl yapılacağı incelenmiş, adaptasyonun gerçekleşmesi için kontrolör katsayılarının nasıl yenileneceği belirlenmiştir.

Kontrolör parametrelerinin sistemin davranışına etkinliğini saptamak için algoritmada bir öğrenme aşaması bulunmaktadır.

Öğrenme aşamasından sonra, sistem kendi haline bırakılır, ancak bu sırada sürekli olarak sistem çıkışı, test sinyali uygulanarak gözlenir. Gözlem sonuçları istenenden farklı olursa, öğrenme aşamasında edinilmiş bilgiler kullanılarak adaptif kontrolörün yeni parametreleri saptanır ve sistem bu parametrelerle çalışır.

Sistem parametrelerindeki değişmelerin bir adaptasyon işlemi sırasında çok az değişeceği bilindiğine göre, adaptasyonun ilk denemede sağlanacağı açıktır. Ancak sistem parametreleri üzerindeki değişmeler büyük olursa, ilk denemede istenen sonuca erişilemez. Sonuca erişmek üzere denemelere devam edilir. Ancak bu denemelerde ilk atılan adım en büyüktür ve sonuca yaklaştıkça adımlar küçülecektir.

Tezde yukarıda anlatılan ilkelere göre çalışan adaptif kontrolör önce bir melez bilgisayarda simüle edilmiş ve karşılaşılabilecek tüm olasılıklar için irdelenmiş ve olumlu sonuç alınmıştır.

Simülasyonla doğruluğu saptanan adaptif kontrolör mikrobilgisayarla gerçekleştirilmiştir. Gereklili olan bilgi aktarma aygıtı sisteme eklenmiştir. Gerçekleme sırasında donanım olarak endüstri standartlarına bağlı kalınmıştır. Algoritma için gerekli olan yazılım gerek çalışmanın hızını artırmak ve gerekse bilgisayar aritmetiğinin sorunlarını irdelleyebilmek açısından makina dilinde gerçekleştirilmiştir.

Mikrobilgisayar ile gerçekleştirdiğimiz adaptif kontrolör çeşitli durumlar için denenmiş ve beklenen sonuçlar alınmıştır.

## SUMMARY

## ADAPTIVE CONTROLLER WITH MICROCOMPUTER

The objective of this work which intends to use a microcomputer for adaptive control might be explain as follows :

- The needs of industrial application in the field of adaptive control today, is to use a cheap and easy to use adaptive controller which can easily handle adaptive techniques. An adaptive control algorithm which is suitable the above consideration has been developed.
- The algorithm, in the mean time, has been developed according to hardware and software requirement of a microcomputer.
- During the implementation of algorithm with a microcomputer, the following matters were also considered : 1<sup>o</sup> increase the operation speed of controller ; 2<sup>o</sup> an effective programming technique.

In the first step of adaptive control, process must be identified by adaptive controller. There are several methods may be used for this purpose either deterministic or stochastic. In this work a deterministic approach has been chosen and it is based on the frequency response of a system.

Since the measuring of the frequency response of a system is not easy, two time domain characteristics which are correspond to frequency response were used and they are the overshoot and the peak time. Variation of the peak time and the magnitude of overshoot indicate the change of system frequency response.



The peak time and overshoot can be controlled by a PID controller both directions in order to make desired values.

The adaptive control algorithm which is presented in this work uses the above principle and consist of following features :

- When an adaptive control system has been designed, it can properly work and the adaptive controller is capable to handle given process in order to establish the required response from the system.
- System start to run after setting of coefficient (P,I,D) of controller. And this mode of controller is represented as (1,0,0) which means P=1, I=0 and D=0 .
- The next step is dedicated for learning of process. In this step P,I,D are incremented as low as the process is not disturbed. The values of incrementatin of PID are also important question as far as applications are concerned. In this research,  $\Delta P_o$ ,  $\Delta I_o$  and  $\Delta D_o$  are calculated in respect to system frequency response. When  $\Delta P_o$ ,  $\Delta I_o$  and  $\Delta D_o$  are added to the previous values, the new values willbe as below :

$$P = P + \Delta P_o$$

$$I = I + \Delta I_o$$

$$D = D + \Delta D_o$$

The overshoot and peak time of system for the new PID will be respectively as follows ;

$$A = A + \Delta A_o$$

$$T_A = T_A + \Delta T_{A_o}$$

The relationship between causes and effects,

$$\text{RATIO } P_A = \frac{A - A_i}{\Delta P_o}$$

$$\text{RATIO } P_{T_A} = \frac{T_{A_i} - T_A}{\Delta P_o}$$

$$\text{RATIO } I_A = \frac{A - A_i}{\Delta I_o}$$

$$\text{RATIO } D_A = \frac{A_i - A}{\Delta D_o}$$

where  $A_i$  and  $T_{A_i}$  are required values, are calculated. These relations will be used in adaptation steps.

- After learning process, the system is in the free running mode. During this mode the response of system continuously observed and measured its peak time and overshoot whether or not in the given tolerance.

- If one of them is not in the tolerance, adaptive controller switches to adaptation mode. The above table gives the possible causes for adaptation process (TABLE I). In this table,

$A_ =$  : Overshoot is equal to desired overshoot  $A_i$ ,

$A \uparrow$  : Overshoot is higher than  $A_i$ ,

$A \downarrow$  : Overshoot is lower than  $A_i$ ,

$T_{A_ =}$  : Peak time is equal to desired peak time  $T_{A_i}$ ,

$T_{A \uparrow}$  : Peak time is more than  $T_{A_i}$ ,

$T_{A \downarrow}$  : Peak time is less than  $T_{A_i}$ ,

	$A =$	$A \uparrow$	$A \downarrow$
$T_{A=}$	1° REQUIRED RESPONSE	2° decrease I (I↓) or increase D (D↑)	3° increase I (I↑) or decrease D (D↓)
$T_{A \uparrow}$	<p>Firstly, increase P (P↑)</p> <p>then, calculate the expected value of A, jump to either mode of 1°, 2°, or 3° according to result</p>		
$T_{A \downarrow}$	<p>Firstly, decrease P (P↓)</p> <p>then, calculate the expected value of A and jump to either mode of 1°, 2°, or 3° according to result</p>		

TABLE I

As we see in the above table, P has the priority over I and D.

In order to reach to required values, the adaptation process might be repeated a few time. But a new try give a better result then it was before.

The required incrementation of P,I,D are calculated according to the learning step of algorithm and used the following equations :

$$\Delta P = \frac{\Delta T_A}{\text{RATIO } P_{T_A}}$$

$$A_{\text{exp}} = \Delta P \cdot \text{RATIO } P_A + A \quad (A_{\text{exp}} : \text{Expected value of } A)$$

$$\Delta A_{\text{exp}} = A_i - A_{\text{exp}}$$

$$\Delta D = \frac{\Delta A}{\text{RATIO } D_A} P \quad \text{or} \quad \Delta D = \frac{\Delta A_{\text{exp}}}{\text{RATIO } D_A} P$$

$$\Delta I = \frac{\Delta A}{\text{RATIO } I_A} P \quad \text{or} \quad \Delta I = \frac{\Delta A_{\text{exp}}}{\text{RATIO } I_A} P$$

The second part of this thesis is dedicated for simulation of the proposed adaptive control algorithm; For simulation of the algorithm, an Hybrid Computer System (EAI-500) was employed.

During the simulation work, the adaptive control algorithm which is drawn by this research was tested for all possible situation as well as the other important features of the research such as sampling time, not-disturbed incrementation of P,I,D coefficients in the learning mode. Some test example are given in the thesis.

The last section of the study is dedicated for implementation of algorithm by means of a microcomputer and data acquisition device.

This section is concentrated on these three topics; 1° Sampling time, 2° Computer arithmetic , 3° Digital noise.

**Sampling Time :** In applications, the sampling time must be chosen neither short nor long, in contrast to theoretical result. For the sampling time calculation,

$$T_s = \frac{\sqrt{2}}{\Omega_3} E_{\text{max}}$$

equation was choosen. Here ,  $E_{\text{max}}$  stand for maximum error that can be accepted and  $\Omega_3$  is frequency band width with respect to - 3 dB.

If the sampling time is taken longer then what this equation says, it gives us sampling error as we know.

If the sampling time is taken shorter, it causes a digital noise error and needs to much operation which means to much time for operation.

**Computer Arithmetic :** In order to create a fast and low digital noise software for proposed adaptive control algorithm, the microcomputer was programmed with machine language. All necessary arithmetic operation programs are written in fixed point arithmetic format. And mostly use 16 - or 32 bit arithmetic operation.

**Digital Noise :** When an analog signal is converted into digital number, quantization and resolution are well known problems. In addition to these, especially, multiplication and division of two numbers bring out error. These all errors are called as digital noise. In this research all of these problems are studied and try to keep the digital noise as low as possible.

The proposed adaptive controller has been designed base on Motorola M 6800 microcomputer and used its family components, As data acquisition device Burr-Brown ADC and DAC (MP 7208-7216-7104) which is 12-bit converter, is used.

The result of realization of adaptive controller which is presented in this thesis satisfied the theoretical and practical results.

## SEM BOLLER

$C(s)$	: Sistem cevabının Laplace dönüşüğü
$R(s)$	: Giriş işaretinin Laplace dönüşüğü
$c(t)$	: Sistem cevabı
$\zeta$	: Sönüm katsayısı
$\omega_n$	: Doğal frekans
$A$	: Aşım , aşım miktarı
$T_A$	: Aşıma erişme zamanı, aşım zamanı ya da tepe zamanı
$T_d$	: Gecikme zamanı
$T_r$	: Yükselme zamanı
$T_s$	: Yerleşme zamanı
$M_p$	: Çınlama genliği
$\omega_p$	: Çınlama frekansı
$\Omega_3$	: -3 dB kriterine göre frekans band genişliği
$\omega_A$	: Salınım frekansı
$\Omega_k$	: Kuramsal band genişliği
$P$	: PID kontrolörün oransal katsayısı
$I$	: PID kontrolörün integral katsayısı
$D$	: PID kontrolörün türev katsayısı
$T_o$	: Örnekleme zamanı
$E_k$	: Kuantalama yanılığı
$K$	: Dizgenin ileri yol kazancı



**1.** bölüm

**GİRİŞ**

Adaptif kontrol, dinamik özellikleri çeşitli nedenlerle değişen dizgenin ; devamlı olarak kendisini gözlediği ve bu gözlemlere dayanarak kontrol sisteminin kendisini devamlı ve otomatik tasarladığı kontrol türü olarak tanımlanmıştır (1).

Adaptif kontrol alanında bilinen en eski çalışma örneği 1950 yıllarında Caldwell 'in gerçeklediği pnömatik adaptif regülatördür (2). Bu çalışma, basit regülatörlerin otomatik ayarlanması amacıyla yapılmıştır. Bu yıllarda adaptif kontrol alanındaki çalışmalar oldukça yoğunlaşmıştı ve çoğunlukla yüksek performanslı uçaklar için otomatik pilotların gerçekleştirilmesi amacını güdüyordu. Bu dönemde yapılan çalışmalara örnek olarak ; Anderson, Buland ve Cooper sistemi (Aeronutronic Auto-Pilot) , Osder sistemi, Marx sistemi verilebilir (1,3). Bütün bu çalışmalar, model referans adaptif kontrol, özayarlı adaptif kontrol gibi bir çok adaptif kontrol tekniğinin temelini oluşturmaktadır. Ancak bu yıllarda yapılan araştırmalar, donanım ve teori yetersizliği ile fazla ileri gidememiştir.

1960 lı yıllarda, kontrol alanında adaptif kontrolün gelişmesine de etki edecek bir çok kuramsal çalışma yapılmıştır. Durum uzayı ve kararlılık teorisinin gelişimi, stokastik kontrol teorisinde önemli adımların atılması, optimal kontrol teorisinin gelişmesi adaptif sistemlerin anlaşılmasını önemli ölçüde arttırmıştır. Aynı zamanda bilgisayar teknolojisinin gelişimi, analog ve dijital bilgisayarların kontrol sistemlerinde gerçek zaman elemanı olarak kullanılabilmesi düşüncesini getirmiştir. Bu düşünce adaptif kontrol için önemli bir adım olmuştur.

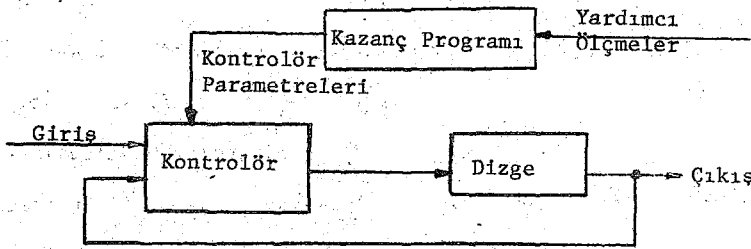
Adaptif kontrol için ilk çalışmalar olarak adlandırabileceğimiz bu kuşağı, bir çok adaptasyon modelinin geliştirildiği, ancak özlü bir tanımın ve sınıflandırmanın yapılamadığı, uygulamalar için teknolojinin yetersiz kaldığı ilk kuşak olarak tanımlayabiliriz.



1970 li yıllarda adaptif kontrole ilgi tekrar artmıştır. Bu yıllara kadar kontrol teorisindeki gelişmeler, daha önce teklif edilmiş olan bir çok adaptif kontrol modelinin anlaşılmasını ve geliştirilmesini sağlamıştır. Bu dönemdeki temel çalışmanın, öğrenme ve adaptif kontrole ilişkin bir çok modelin tek bir ortak çatı altında toplanabileceğini gösteren Tsypkin tarafından yapıldığı kabul edilmektedir (2). Aynı zamanda sistem belirlenmesi (identifikasyon) ve parametre kestirimi (estimasyon) konularındaki gelişmeler adaptif kontrolü anlamada ana rolü oynamıştır. Bu gün bu alanda yapılan çalışmaların üç ana grupta yürütülmekte olduğunu söyleyebiliriz:

#### 1) KAZANÇ PROGRAMLAMA

Bazı durumlarda, dizge dinamiğindeki değişikliklere bağlantılı olan yardımcı değişkenler bulmak mümkündür. Buna göre kontrolör parametrelerini önceden yapılacak bir programa göre değiştirerek adaptasyon sağlanabilir. Bu yöntemle ilişkin blok diyagram Şekil 1.1 deki gibidir (2).



Şekil 1.1 Kazanç Programlamalı Adaptif Kontrol.

Bu yaklaşımda, çalışma ortamının koşullarını içeren bir çok çalışma noktası için kontrolör parametrelerini belirleme biçiminde tasarım yapılır. Kararlılık ve davranış simülasyonla sınıanır. Gerekirse çalışma koşullarının sayısı arttırılır. Bu tür sistemlerin tasarımında ana problem uygun

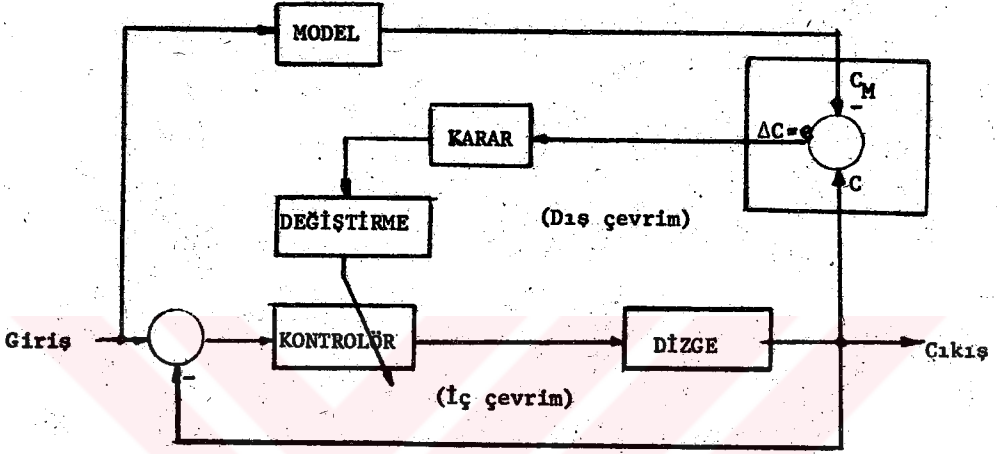
programlama deęişkenlerini bulmaktır. Bu ise doğal olarak ele alınan dizgenin fiziksel yapısına ilişkin bilgiye dayanır.

Bazı araştırmacılar tarafından, önprogramlı adaptif kontrol (3) ya da açık çevrimli adaptif kontrol (4) olarak adlandırılan modelleri de aynı sınıfta toplamak mümkündür.

Şekil 1.1 den de görülebileceęi gibi adaptasyon açısından, kazanç programlamalı model açık çevrimlidir ve doğal olarak açık çevrimin sakıncalarını taşır. Doğru olmayan bir programı düzeltecek hiçbir geri besleme yoktur. Bu yöntemin bir dięer sakıncası, tasarım için çok zaman harcamasıdır. Kontrolör parametrelerinin bir çok çalışma koşulu için belirlenmesi ve davranışın simülasyonlarla sınanması gerekmektedir. Yöntemin üstünlüğü ise dizgedeki deęişikliklere karşı kontrolör parametrelerinin çok çabuk deęiştirilebilmesidir. Hızlılık için sınırlandırma, deęişiklikleri belirtecek yardımcı ölçmelerin hızından gelecektir. Bu model güdümlü mermilerde, özellikle uçuş kontrol sistemlerindeki parametre deęişimlerini gidermek için kullanılan en üstün tekniktir. Kaynak 2 de ayrıca Taylor Instruments tarafından kazanç programlamalı endüstriyel kontrolörlerin pazarlanmakta olduğu da belirtilmektedir..

## 2) MODEL REFERANS ADAPTİF SİSTEMLER (MRAS)

Bu sistemler için verilen genel şema Şekil 1.2 deki gibidir. Bu model gerçekte servo sistemler için geliştirilmiştir. Sistemden istenen özellikler bir referans model anlamında verilir ve bu model ideal olarak dizge çıkışının nasıl olacağını gösterir. Burada referans modelin kontrol sisteminin bir parçası olduğuna dikkat edilmelidir. Sistemin iki çevrimden oluştuğu söylenebilir. İç çevrim alışıl gelmiş dizge ve kontrolörden oluşmuş bir kontrol çevrimidir. Kontrolör parametreleri model çıkışı ile dizge çı-



Şekil 1.2 Model Referans Adaptif Kontrol

kısa arasındaki fark küçük olacak şekilde dış çevrim tarafından ayarlanır. Burada ana problem bu farkı sıfıra getirecek kararlı bir sistemin elde edilebileceği ayarlama düzenini belirlemektir. Aşağıdaki ayarlama düzeni, orijinal MRAS de kullanılan MIT- kuralı olarak adlandırılan düzendir :

$$\frac{d\theta}{dt} = -k \epsilon \quad (1.1)$$

Burada ;  $\theta$  bileşenleri ayarlanabilen parametreler,  $\epsilon$  model hatası,  $\text{grad}_{\theta}\epsilon$  duyarlık türevleri ve  $k$  ise adaptasyon hızını belirleyen bir parametredir. Buna göre ayarlama düzeni, duyarlık türevlerini hesaplayacak bir lineer filtre , bir çarpıcı ve bir integratörden oluşacaktır. Bu düzen bir çok adaptif sistem için tipik bir düzendir.

MIT-kuralına benzer olarak, MRAS için kararlılık teorisi kullanılarak elde edilmiş bir çok ayarlama düzeni teklif edilmiştir. (1.1) deki duyarlık türevleri yerine bir çok fonksiyonlar getirilmiştir.

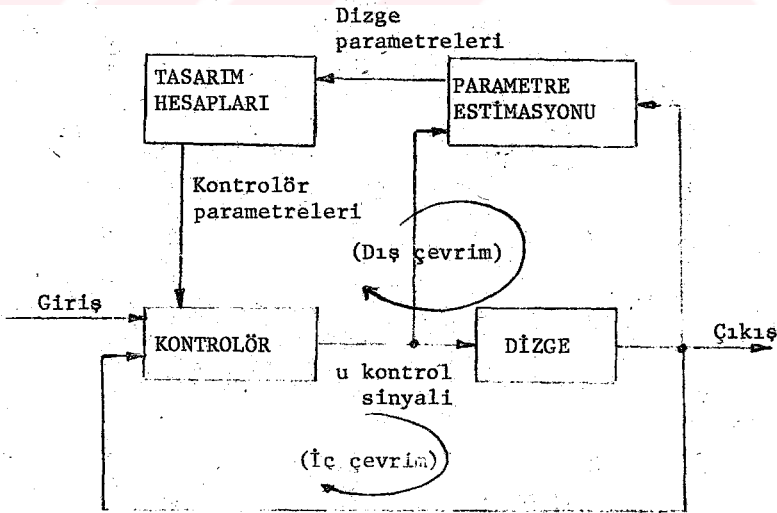
MRAS orijinal olarak Whitaker tarafından teklif edilmiştir (1,2).

Daha sonraki çalışmalar Parks(5), Monopoli(6) ve Landau(7) tarafından yapılmıştır. Kaynak 7 de, MRAS için yapılan çalışmalar sınıflandırılmış, tabloları ve modelleri verilmiştir. Aynı zamanda bu konuya ilişkin olarak 253 literatür sistematik olarak verilmektedir.

### 3) ÖZAYARLI ADAPTİF SİSTEMLER

Adaptif kontrol alanındaki üçüncü temel sınıf olan özayarlı adaptif sistemler olarak adlandırdığımız bu guruba, Åstrom (2) tarafından tanımlanan STR (self tuning regulators) ve diğer birçok araştırmacı tarafından tanımlanan öz-optimizasyonlu adaptif kontrol sistemlerini (self-optimizations adaptive control systems) toplamak mümkündür.

Bu gurubun temel özelliği, parametre kestirimi ile dizgenin belirlenmesi ve elde edilen değerlere dayanarak optimum kontrolör parametrelerinin tasarlanmasıdır. Bu sisteme ilişkin blok diyagram Şekil 1.3 te verilmiştir.



Şekil 1.3 Özayarlı Adaptif Kontrol

Burada da tüm sistem iki çevrimden oluşmaktadır. İç çevrim dizgeyi ve parametreleri ayarlanabilen kontrolörden oluşmuştur. Kontrolör parametreleri dış çevrim tarafından ayarlanır. Dış çevrim ise bir rekürsif parametre estimatörü bloğunu ve bir tasarım hesapları bloğunu içerir. Özayarlı adaptif sistemler, gerek parametre estimatörü gerekse tasarım hesaplayıcısı yönünden oldukça esnektir. Şimdiye kadar tasarım tekniği olarak, kutup yerleştirmesi, faz ve kazanç payları, minimum varyans kontrolü gibi bir çok tasarım tekniği düşünülmüştür. Parametre estimatörü için de stokastik yaklaşım, minimum kareler, genişletilmiş ve genelleştirilmiş minimum kareler, genişletilmiş Kalman filtresi ve maksimum Likelihood yöntemleri üzerinde en çok çalışma yapılmış yöntemlerdir. Kaynak 4 de özayarlı adaptasyon için uygun teknikler verilmiştir.

Bu grupta ilk çalışma Kalman tarafından yapılmıştır (1,2). Kalman özel amaçlı bilgisayar gerçekleyerek kontrolör görevini bu bilgisayara yaptırmıştır. Åstrom ve Wittenmark (8) tarafından yapılan çalışmalar bu konunun temel çalışmalarıdır. Kurz ve diğerleri çalışmalarında, ele alınan dizgeye uygun olarak seçilmesi gereken, bir belirleme yöntemi ve bir kontrol kuralı ile adaptasyonun sağlanabileceğini göstermiş ve örnek olarak altı farklı adaptasyon algoritması incelemişlerdir (9).

Bütün bu çalışmalardan yararlanarak söylenebilir ki, adaptif kontrol için kurulmuş tam bir teori yoktur. Böyle olunca da adaptif kontrolü gerçeklemeye çalışırken ele alınan probleme özgü olarak önceden sezgisel olarak çözümlenmesi gereken bir çok sorun vardır.

Daha önce de belirtildiği gibi adaptif kontrolün ilk kuşağında yapılan çalışmalarda analog donanım kullanılmıştır. Bu ise zaten bilinen kontrol yöntemlerine göre karmaşık olan adaptif kontrolün gerçekleşmesinde daha da büyük sorunlar oluşturmakta idi.

Son zamanlarda bilgisayar teknolojisinin de hızla ilerlemesi özellikle mikroşlemcilerin çıkması, adaptif kontrole da bir başka boyut kazandırmıştır. Analog düzenlerle gerçeklemede ortaya çıkan bir çok sorun bu sayede giderilmiştir. Bilinen adaptasyon teknikleri mikrobilgisayar kullanarak gerçekleştirilmeye çalışılmış ve birçok yöntem teklif edilmiştir(10,11,12). Geçtiğimiz son on yılda çoğunlukla üniversitelerde olmak üzere adaptif kontrol konusunda çok miktarda simulasyon ve laboratuvar denemeleri yapılmıştır.

Günümüzde adaptif kontrol uygulamalarında amaç, endüstriyel alanda adaptif tekniklerin kolaylıkla kullanılabilirliğini sağlayacak basit ve ucuz yollar bulmaktır.

Çalışmamızda bu amaca hizmet edebilecek bir adaptif kontrol yöntemi geliştirilmiştir. Adaptif kontrolde amaç, sistem dinamiğindeki ve bozuculardaki değişikliklere adapte olabilecek şekilde kontrolörün otomatik ayarlanmasını sağlamaktır. O halde başlangıç noktası, bir kurulu düzen ve parametreleri ayarlanabilir bir kontrolörden oluşmuş, bilinen bir geri beslemeli kontrol çevrimidir. Ana problem ise kurulu düzen dinamiğindeki değişiklikleri belirleyebilmek ve buna göre kontrolör parametrelerini değiştirecek uygun bir yol bulmaktır.

Bu tezde yaptığımız çalışmanın ana hatları aşağıdaki gibi verilebilir :

- 1<sup>o</sup> Sistem kurulduğunda, kontrolör, sistemi kendisinden istenen davranışı verecek biçime getirecek yetenektedir.
- 2<sup>o</sup> Sistem parametrelerinin değişmesi ya da bozucu etkisi sonucunda istenen davranışı sürdürecekt biçimde kontrolör parametreleri devamlı olarak ayarlanır (Adaptif ayarlama) .

- 3° Sistemden istenen davranış, sistemin frekans-band karakteristiği ya da buna eşdeğer olan zaman domeni davranışı ile verilebilmektedir. Sistemin davranışı ise uygun olarak seçilecek bir test sinyaline cevabı izlenerek değerlendirilir ve bunun için dizge karakteristiklerinin tam olarak bilinmesine gerek yoktur.
- 4° Bütün bu işlemleri yerine getirmek üzere bir mikrobilgisayar ve gerekli çevre donanımı kullanılacaktır.

Kaynak 2 de, Åstrom yeni tip adaptif regülatörlerin nasıl olması gerektiğini anlatmaktadır. Åstrom, genellikle adaptif regülatör denildiğinde üzerinde hiçbir ayar düğmesi bulunmayan mutlak bir kara kutunun kastedildiğini, ancak çok özel uygulamalar dışında bunun mümkün olamayacağını söylemektedir. Gerçekten de kontrol isteklerinin çok farklı olacağı düşünülürse böyle genel amaçlı kara kutunun gerçekleşmesi düşünülemez. Bu durumda Åstrom, bu kara kutuya kapalı çevrimli sistemden istenen davranışı gösterecek bir takım ayar düğmelerinin konulabileceğini belirtmektedir. İstenen davranışı gösterecek büyüklük olarak, band genişliğinin, faz ve kazanç paylarının seçilebileceğini örnek olarak vermiştir ve yeni tip regülatörlerin bu biçimde gerçekleşmesinin istendiğini ifade etmektedir.

Yukarıda çalışmamızın ana hatlarına bakıldığında gerçeklediğimiz bu adaptif kontrol yönteminin Åstrom'un belirttiği ve gerçekleştirilebileceğini söylediği yeni tip regülatörlere bir örnek olacağı açıkça görülmektedir.

Bir başka açıdan düşünüldüğünde, gerçekleştirdiğimiz yöntem, adaptif kontrolün ilk kuşağının karşılaştığı donanım güçlüklerini günümüz teknolojisi yardımıyla yenen, pratik uygulamaya yönelik bir yöntemdir. Yöntem, pratik düşünceye dayanarak gerçekleştirilmiş olması nedeniyle, daha geniş bir uygulama alanı bulabilecektir.

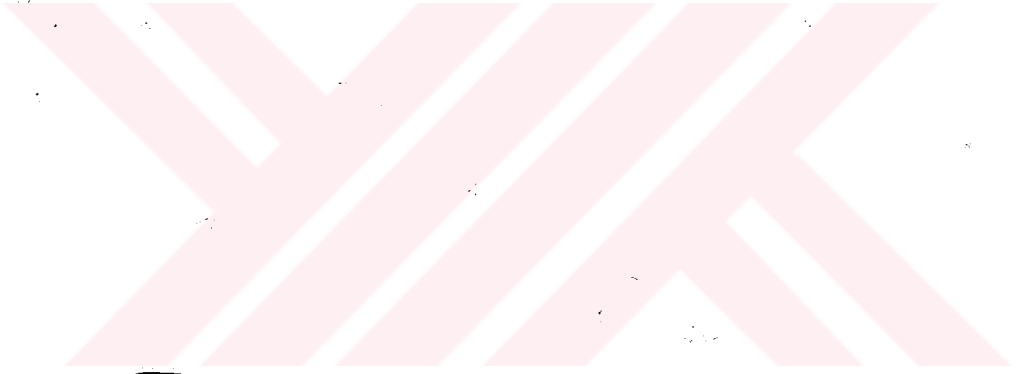
Yöntemin tanıtılmasına ayrılan ikinci bölümde, öncelikle çalışmamızda esas olan bazı temel kavramlar ve tanımlar verilecek ve daha sonra geliştirilen adaptif kontrol sistemi ayrıntılı bir biçimde tanıtılacaktır.

Üçüncü bölümde, yöntemin uygulandığı simülasyon çalışmaları ve elde edilen deney sonuçları verilecektir.

Yöntemin gerçekleştirilebilirliğinin gösterildiği simülasyon çalışmalarına dayanarak, böyle bir adaptif kontrol sisteminin mikrobilgisayar ve çevre donanımı ile gerçekleştirilmesi, karşılaşılan güçlükler ve çözümleri dördüncü bölümde verilmiştir.

Çalışmanın bütün adımlarında elde edilen özellikle yenilik getiren sonuçlar, toplu bir biçimde sonuç bölümünde verilecektir.





**2.** bölüm

# **YÖNTEMİN TANITILMASI**

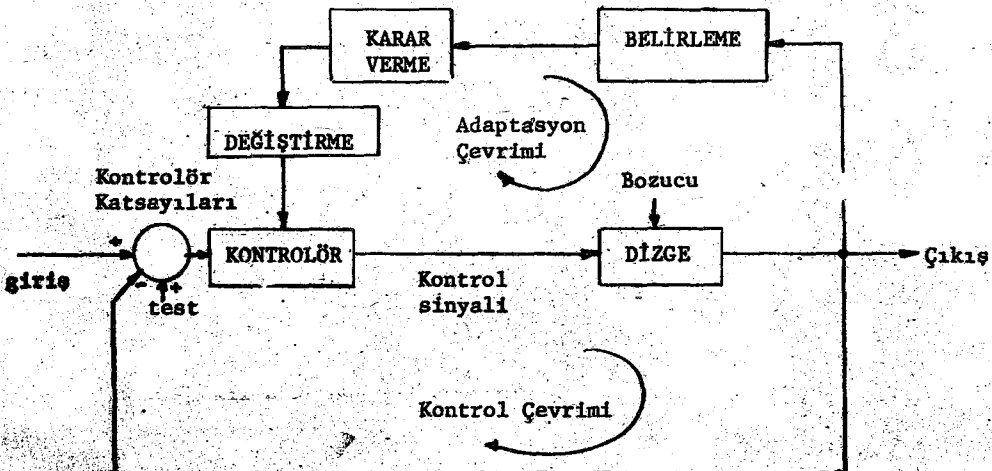
## 2.1 TANIMLAR

Giriş bölümünde ana hatları ile tanıtilan çalışmamızda ele alınan problem aşağıdaki biçimde verilebilir :

Belirli bir kontrol görevini yerine getirmek üzere gerçekleştirilmiş bir fiziksel düzen kurulmuştur. Bu düzen alışlagelmiş kontrol sistemlerine benzer olarak dizge ve kontrolörden oluşmuş negatif geri beslemeli tek giriş ve tek çıkışlı bir otomatik kontrol sistemidir. Buna göre sistemin normal çalışması olarak adlandıracağımız bu çalışma, düzenin gerekli kontrol işlemini yerine getirmesi sürecidir. Bu amaçla sisteme uygun bir giriş sinyali uygulanmakta ve istenilen çıkış sinyali elde edilmektedir.

Ancak, ele alınan bu düzende dizge özellikleri çevresel nedenlerden ötürü değişmektedir ve bu durumda istenilen çıkış elde edilememektedir. Bu yüzden dizgenin , bu değişiklikler giderilecek biçimde adaptif olarak kontrol edilmesi gerekmektedir.

Buna göre tasarlanacak sistemin işlemsel blok diyagramı Şekil 2.1 deki gibi olacaktır.



Şekil 2.1 İşlemsel blok diyagram

Blok diyagramda görülen işlemler ve büyüklükler aşağıdaki biçimde kısaca tanımlanabilir :

**Dizge :** Belirli bir kontrol görevini yerine getirmek üzere kurulmuş, ancak çevresel nedenlerden ötürü özellikleri değişen, bu yüzden de adaptif olarak kontrol edilmesi gereken düzendir.

**Kontrolör :** Adaptif kontrolün gerçekleştirilmesi için, dizge değişikliklerini giderebilecek şekilde parametreleri ayarlanabilen bir kontrolördür.

**Kontrol Çevrimi :** Dizge ve kontrolörü içeren, sistemin normal çalışmasını sağlayan çevrimdir.

**Adaptasyon Çevrimi :** Adaptif kontrolü yerine getirmek üzere kurulu düzene ek olarak getirilmiş bir çevrimdir. Burada; dizge çıkışının sürekli olarak izlenmesi, dizge dinamiğinde değişiklik olup olmadığının saptanması, elde edilen bilgilerden yararlanarak kontrolör parametrelerinde gerekli değişikliklerin yapılması işlemleri gerçekleştirilir.

**Giriş Sinyali :** Sistemin normal çalışmasını yerine getirmesi için uygulanan sinyaldir.

**Test Sinyali :** Dizgede oluşacak değişikliklerin saptanması amacıyla sistemin giriş sinyaline ek olarak uygulanan sinyaldir. Test sinyalinin ne şekilde seçildiği Bölüm 2.2.5'de verilmiştir. Şimdilik test sinyalinin sistemin normal çalışmasını etkilemeyecek biçimde olduğunu söylemek yeterli olacaktır.

Adaptasyon çevriminde görülen belirleme, karar verme ve değiştirme işlem blokları da kısaca şu şekilde tanımlanabilir :

**B e l i r l e m e :** Sisteme uygulanacak test sinyaline cevabının izlenerek davranışının saptanması işlemidir.

**K a r a r V e r m e :** Sistemin davranışının istenilen biçimde olup olmadığının sınılanması ve buna göre kontrolör parametrelerinin değiştirilip değiştirilmeyeceğine karar verilmesidir.

**D e ğ i Ő t i r m e :** Test sonucunda elde edilen bilgilere ve daha önceden elde edilmiş olan bilgilere dayanarak kontrolör parametrelerinin yeni değerlerinin hesaplanması ve yenilenmesi işlemleridir.

Bu temel işlemlere dayanarak adaptasyon çevrimi için bir tanım verilebilir:

**Tanınım 1: Adaptasyon Çevrimi:**

*Adaptif kontrol amacıyla gerçekleştirilen ve adaptasyonun gerektirdiği belirleme, karar verme ve değiştirme temel işlemlerinin yapıldığı çevrimdir.*

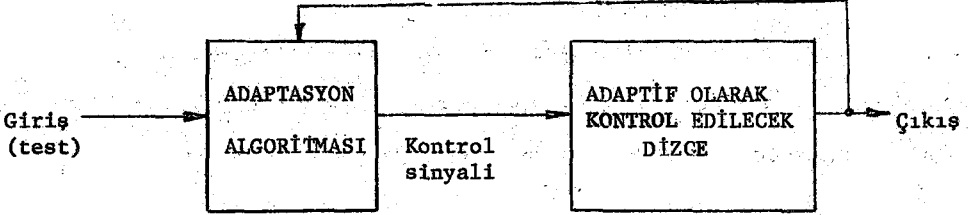
Burada problem, sadece adaptasyonun gerçekleşmesi olarak ele alındığında, sistemin normal çalışmasının göz önüne alınmasına gerek yoktur. Bu düşünce ile giriş sinyalini kaldırmak mümkündür. Bu durumda sisteme bir tek test sinyali uygulanmakta ve buna karşılık elde edilen sistem cevabı da sistemin davranışını belirleyebileceğimiz bir çıkış sinyali olacaktır.

Daha önce de belirtildiği gibi adaptasyon için gereken tüm işlemler bir mikrobilgisayar sistemi ile gerçekleştirilecektir. Bu bakımdan Şekil 2.1 de gösterilen parametreleri ayarlanabilen kontrolörün de bir dijital kontrolör olarak aynı mikrobilgisayar sisteminden yararlanarak gerçekleştirilmesinin uygun olacağını hemen söylemek mümkündür. Dijital kontrolörün tartışması daha ileride tekrar ele alınacaktır.

Yöntemin tanıtılmasına açıklık getirmek amacıyla ve mikrobilgisayarın kullanıldığını göz önüne alarak burada bazı tanımların ve blok diyagram gösterilimlerinin getirilmesinde yarar vardır.

### Tanım 2: Adaptasyon Algoritması

Adaptasyon çevrimi yazılımı ile dijital kontrolör yazılımını içeren algoritmadır. Buna göre sistemin blok diyagramı Şekil 2.2 deki gibi verilebilir.

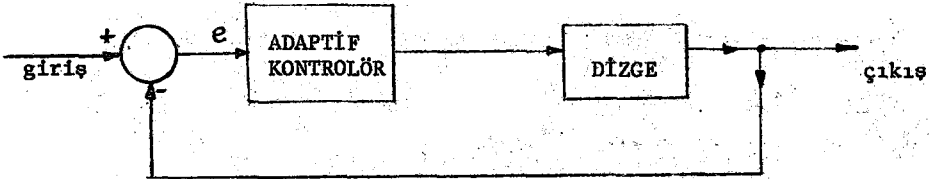


Şekil 2.2 Adaptasyon Algoritması

### Tanım 3: Adaptif Kontrolör

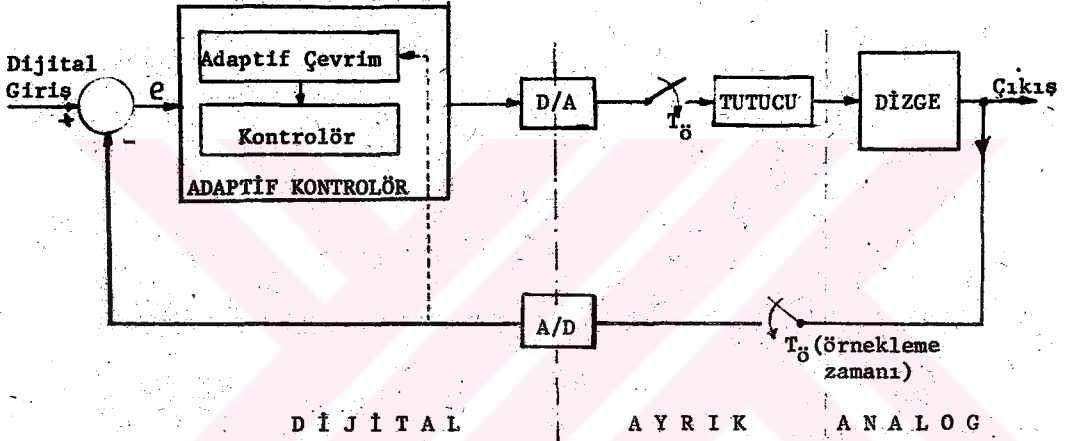
Özellikleri zamanla değişen sistemin, istenilen cevabı vermesi için gerekli kontrol sinyalini üretecek kontrolördür. Adaptif kontrolörün iki görevi vardır. Birincisi dizge için alışılmış kontrol işlevini sağlamak ikincisi ise dizgenin özellikleri değiştiğinde istenileni sağlayacak biçimde yapısını değiştirmek, yani adaptasyonu gerçekleştirmek.

Adaptif kontrolör tanımı bu şekilde yapıldıktan sonra sistem modelini alışılmış otomatik kontrol sistemi olarak vermek mümkündür (Şekil 2.3),



Şekil 2.3. Adaptif kontrolör

Adaptif kontrolörün dijital bilgisayarda gerçekleştirilmesi nedeniyle Şekil 2.3 de sinyal açısından var olan çelişkiyi gidermek için Şekil 2.4 deki gibi bir blok diyagram gösterilimi uygun olacaktır. Bu blok diyagram dijital kontrol sistemlerinin incelenmesinde kullanılan genel blok diyagram gösterilimine göre verilmiştir (13).



Şekil 2.4 İşaret yönünden adaptif kontrol sisteminin blok diyagramı

Burada sisteme, daha önce de belirtildiği gibi bir dijital giriş (test) sinyali uygulanmaktadır. Dijital bilgisayarda gerekli kontrol sinyali üretilmekte ve analog sinyale dönüştürülmektedir. Analog olarak elde edilen kontrol sinyali dizgeye uygulanmakta ve dizge çıkışı tekrar dijital sinyale dönüştürülerek çevrim tamamlanmaktadır. Bu haliyle bilinen bir dijital kontrol sistemi ile aynı olan sistemimizdeki tek fark adaptif kontrolör bloğundaki adaptif çevrimdir. Bu çevrim dizge çıkışını izleyerek gerekiyorsa dijital kontrolörün katsayılarını değiştirerek istenilen sistem cevabını sağlayacaktır.

Adaptasyon için, dizgeye özgü en uygun yolun seçilmesi, mümkün olan yeterli ve etkin en kısa yöntemin kullanılması adaptasyonun hızlılığı açısından önemli bir sorundur. Dizgenin davranış özellikleri ve kontrol bilgisinden yararlanarak öncelikle, dizge dinamiğinin hangi karakteristiklerinin bilinmesi, ölçmelerin hangi doğrulukta ve sıklıkta yapılması gerektiği soruları cevaplandırılmalı ve bunun ardından da aşağıdaki sorulara cevap verilmelidir :

- Dizge dinamiğini karakterize etmek için mümkün yollar nelerdir ?
- Bu karakteristikleri belirlemek için uygulanması gereken işaretler nelerdir ?
- Bu işaretlerden elde edilen bilgi ne şekilde işlenecektir ?
- Önceden verilebilecek doğruluk nedir ?

Belirleme problemini mümkün olduğu kadar basite indirgemek adaptasyonun doğal olarak getireceği karmaşıklığı artırmamak bakımından önemlidir.

Çalışmamızda da öncelikle bu sorulara cevap aranmıştır. Bu bakımdan ilk olarak dizge dinamiğini karakterize edebilecek büyüklüklerin ele alınması yerinde olacaktır.

## 2.2 İKİNCİ DERECEDEKİ SİSTEMLER İÇİN ZAMAN DOMENİ VE FREKANS DOMENİ KRİTERLERİ ;

Bu bölümde, çalışmamızda gerekli olabilecek, ikinci dereceden sistemlere ilişkin bazı zaman ve frekans domeni kriterleri ele alınacaktır. Ele alınan bu kriterlerin kısaca tanımları, matematiksel ifadeleri verilecek, daha sonra aralarındaki ilişkiler elde edilmeye çalışılacaktır.

## 2.2.1 ZAMAN DOMENİ

Bilindiği gibi, ikinci dereceden bir sistemin transfer fonksiyonu,

$$G_k(s) = \frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (2.1)$$

olup, impuls cevabı (Şekil 2.5a);

$$g_k(t) = \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin \omega_n \sqrt{1-\zeta^2} t \quad (2.2)$$

biçimindedir ve birim basamak girişine cevabı da (Şekil 2.5b) ;

$$c(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin(\omega_n \sqrt{1-\zeta^2} t + \psi) \quad (2.3)$$

dir. Burada,

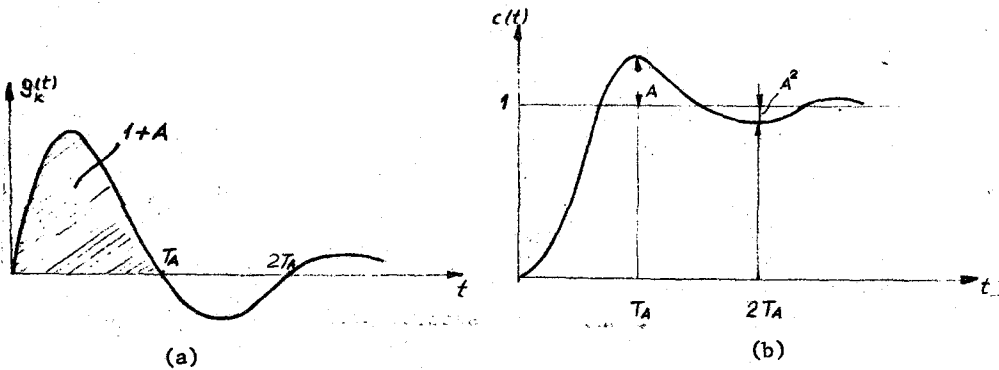
$\zeta$  : Sönüm katsayısı,

$\omega_n$  : Doğal frekans,

ve

$$\psi = \arctg \sqrt{1-\zeta^2} / \zeta$$

dır.



Şekil 2.5 İkinci dereceden bir sistemin,  
a) İmpuls cevabı  
b) Birim basamak cevabı



İkinci dereceden sistemler için tanımlanmış olan bir çok zaman domeni kriteri vardır. Bu kriterler ikinci dereceden sistemin birim basamak cevabı üzerinde tanımlanmış büyüklükler olup; Aşım, Tepe zamanı ya da Aşım Zamanı, Gecikme Zamanı, Yükselme Zamanı, Yerleşme Zamanı gibi kriterlerdir (14). Bu büyüklüklerin sistemin sönüm katsayısı ve doğal frekansı ile ilişkileri aşağıdaki gibidir :

1° Aşım (A) ve Aşım Zamanı ( $T_A$ ) :

Şekil 2.5b den görüldüğü gibi  $c(t)$ ,  $r(t) = u(t)$  giriş işareti etrafında salınım yapan bir işarettir. Bu salınımın tepe ve çökme yaptığı anlar (2.3) ten yararlanılarak,  $n = 1, 2, 3, \dots$  için;

$$T_n = \frac{n\pi}{\omega_n \sqrt{1-\zeta^2}} \quad (2.4)$$

olacaktır. Bu anlara ilişkin çıkış değerleri ise,

$$c_n = c(T_n) = 1 - (-1)^n e^{-n\pi\zeta/\sqrt{1-\zeta^2}} \quad (2.5)$$

biçimindedir.  $n = 1$  için,

$$T_1 = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \quad (2.6)$$

ve

$$c_1 = 1 + e^{-\pi\zeta/\sqrt{1-\zeta^2}} \quad (2.7)$$

olup çıkış işaretinin maksimum değeridir. Buna göre Aşım,  $A$ ;

$$A \triangleq 1 - c_1 = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \quad (2.8)$$

ve Aşım Zamanı (ya da Tepe Zamanı)  $T_A$  ;

$$T_A \triangleq \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \quad (2.9)$$

ya da

$$\omega_A \triangleq \omega_n \sqrt{1 - \zeta^2} \quad (2.10)$$

alınırsa  $T_A = \pi/\omega_A$  olacaktır. A ve  $T_A$  ya bağlı olarak c(t) cevabının tepe ve çökme yaptığı anlar ve değerler ise aşağıdaki gibi verilebilir.

Zaman	$T_A$	$2T_A$	$3T_A$	$4T_A$	$5T_A$	.....
Tepe/Çökme	1. Tepe	1. Çökme	2. Tepe	2. Çökme	3. Tepe	.....
Çıkış Değeri	$1 + A$	$1 - A^2$	$1 + A^3$	$1 - A^4$	$1 + A^5$	.....

2° G e c i k m e Z a m a n ı ( $T_d$ ) :

Çıkış işaretinin, birim basamak girişinin 0,5 değerine eriştiği zaman olarak tanımlanan gecikme zamanının yaklaşık değeri,

$$T_d \approx \frac{1 + 0,7 \zeta}{\omega_n} \quad (2.11)$$

olarak verilmektedir (15).

3° Y ü k s e l m e Z a m a n ı ( $T_r$ ) :

Yükselme zamanı, çıkış işaretinin girişin 0,1 değerine eriştiği an ile 0,9 değerine ulaştığı an arasındaki zaman aralığı ya da çıkışın girişin 0,5 değerine eriştiği noktadan çizilen teğetin, teğet-altı olarak tanımlanmaktadır. Gecikme zamanına benzer olarak yükselme zamanı için de matematiksel ifade bulmak oldukça zordur. Bu bakımdan yaklaşık olarak ifade edilmektedir (15). Yükselme zamanı hakkında bir fikir oluşturmak üzere (2.3) ifadesinde  $\omega_n = 1$  alınarak,  $0 < \zeta < 1$  aralığında çeşitli  $\zeta$  değerleri için, her iki tanıma göre yükselme zamanı bir bilgisayar programı ile hesaplanmıştır. Çıkış işaretinin 0,1 değerine eriştiği an ile 0,9 değerine eriştiği an arasındaki zaman farkı tanımına göre bulunan değerlerden yararlanılarak  $T_r$  için üçüncü dereceden bir yaklaşıklık hesaplanarak,

$$T_r \approx \frac{1,9 \zeta^3 - 0,65 \zeta^2 + 1,16 \zeta + 0,97}{\omega_n} \quad (2.12)$$

bulunmuştur.

Ek 2.1 de ,  $0 < \zeta < 1$  için her iki tanıma göre bulunan  $T_r$  değerleri ile (2.12) ile verdiğimiz yaklaşıklıkta ve Kaynak 15 tarafından verilen ikinci ve birinci dereceden yaklaşık ifadelerden hesaplanan  $T_r$  değerleri verilmiştir.

4<sup>o</sup> Yerleşme Zamanı ( $T_s$ ) :

Yerleşme zamanı, çıkış işaretinin yerleşik durumuna eriştiğinin kabul edildiği zaman olup, yaklaşık olarak ifade edilmektedir.

$$T_s \approx 3 / \zeta \omega_n \quad (\% \pm 5 \text{ kriteri}) \quad (2.13)$$

ya da

$$T_s \approx 4 / \zeta \omega_n \quad (\% \pm 2 \text{ kriteri}) \quad (2.14)$$

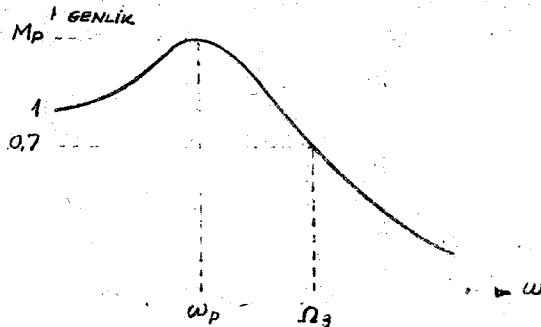
biçiminde verilmektedir (15).

## 2.2.2 FREKANS DOMENİ

(2.1) ifadesinde  $s \rightarrow j\omega$  konulduğunda,

$$G_k(j\omega) = \frac{1}{(1 - \omega^2 / \omega_n^2) + j 2\zeta\omega / \omega_n} \quad (2.15)$$

frekans domenine ilişkin transfer fonksiyonu elde edilir. Buna göre ikinci dereceden sistemin genlik cevabı ise genel olarak Şekil 2.6 daki gibidir.



Şekil 2.6 İkinci dereceden sistemin genlik cevabı

Şekilde gösterilen büyüklükler, frekans domeninde kullanılan kriterlerden olup,

- $M_p$  : Kapalı çevrim frekans genlik eğrisinin maksimum değeri (Çınlama genliği)
- $\omega_p$  : Kapalı çevrim frekans genlik eğrisinin maksimum değerini aldığı frekans, (Çınlama frekansı)
- $\Omega_3$  : Kapalı çevrim frekans genlik eğrisinde, genliğin 0,707 değerine (logaritmik koordinatlarda -3 dB) düştüğü frekans, (Band genişliği)

olarak tanımlanmaktadır. Bu kriterlerin de zaman domeni kriterlerine benzer biçimde, sistemin sönüm katsayısı ve doğal frekansı ile ilişkileri verilebilir.

1° Çınlama Genliği ( $M_p$ ) ve Çınlama Frekansı ( $\omega_p$ ) Çınlama genliğinin ve frekansının sistemin sönüm katsayısı ve doğal frekansı cinsinden karşılıkları  $0 < \zeta < 0,707$  için,

$$M_p = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad (2.16a)$$

ve

$$\omega_p = \omega_n \sqrt{1-2\zeta^2} \quad (2.16b)$$

biçimindedir ( $\zeta$ ) ve  $\zeta > 0,707$  için  $M_p = 1$  dir.

2° Band Genişliği ( $\Omega_3$ ) :

(2.15) ifadesinde, transfer fonksiyonunun genliğinin 0,707 değerine eşitlenmesiyle,

$$\Omega_3 = \omega_n \sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}} \quad (2.17)$$

elde edilir.

### 2.2.3 ZAMAN DOMENİ KRİTERLERİ İLE FREKANS DOMENİ KRİTERLERİ ARASINDAKİ İLİŞKİLER

Frekans ve zaman domenî kriterleri çok sayıda olmasına karşın, Bölüm 2.2.1 ve 2.2.2 de sadece çalışmamızda gerekecek büyüklükler ele alınmıştır. Bu bölümde ise daha önce verilen büyüklüklerin bir kısmı için aralarındaki ilişkiler verilecektir.

a)  $\omega_p - \omega_A$  ilişkisi :

(2.10) ve (2.16b) den yararlanılarak,

$$\omega_p^2 + \omega_n^2 = 2 \omega_A^2 \quad (2.18)$$

ilişkisi elde edilebilir.

b)  $A - M_p$  ilişkisi :

(2.16a) ve (2.8) den gerekli işlemler yapılarak,

$$M_p = \frac{\pi^2 + \ln^2 A}{2 \pi \ln A} \quad (2.19)$$

ya da tersine olarak,

$$A = e^{-\pi M_p (1 - \sqrt{1 - 1/M_p^2})} \quad (2.20)$$

bulunur.

c)  $T_r - \Omega_3$  ilişkisi :

Bölüm 2.2.1 den görüldüğü gibi, yükselme zamanı yaklaşık olarak ifade edilmektedir. Bu nedenle  $T_r - \Omega_3$  ilişkisi yukarıda verilenlere benzer biçimde bulunamaz. Ancak, bir ideal alçak geçiren filtrenin band genişliği  $\Omega_k$  ise, bu filtre için yükselme süresinin (0,5 noktasından çizilen teğetin, teğet-altı kriterine göre) :

$$T_r = \frac{\pi}{\Omega_k} \quad (2.21)$$

şeklinde olduğu gösterilmiştir (16).

Ele aldığımız sistemin de bir alçak geçiren filtre karakteristiğinde olduğu göz önüne alınırsa, (2.21) bağıntısına benzer bir ilişkinin olacağı hemen söylenebilir. Burada sorun, ideal filtrenin  $\Omega_k$  kuramsal band genişliği yerine, bizim sistemimizde hangi büyüklüğün alınacağıdır.

$$T_r' = \frac{\pi}{\Omega_3} \quad (2.22)$$

aldığımızda, Tablo 2.1 de çeşitli  $\zeta$  değerleri için hesaplanan  $\Omega_3$  değerlerini (2.22) de yerine koyarak bulacağımız  $T_r'$  değerlerini, Tablo E1 de bulunan  $T_r$  değerleri ile karşılaştırırsak,

$$T_r' > T_r$$

olduğunu görürüz. Buna göre,

$$T_r < \frac{\pi}{\Omega_3} \quad (2.23)$$

ilişkisi geçerli olacaktır. Demek ki gerçek band genişliğini bildiğimiz bir sistemin basamak cevabında yükselme süresi için bir üst sınır vermemiz mümkün olacaktır.

Elde edilen bütün bu ilişkiler hakkında bir fikir oluşturmak için,  $0 < \zeta < 0,707$  aralığında çeşitli  $\zeta$  değerleri için;  $A$ ,  $M_p$ ,  $\omega_A/\omega_n$ ,  $\omega_p/\omega_n$ ,  $\Omega_3/\omega_n$  değerleri, Ek 2.2 de verilmiştir.

## 2.2.4 SONUÇLAR

Şimdiye kadar yapılan incelemelerden görüldüğü gibi, zaman domeni kriterlerinden sadece Aşım ve Aşım Zamanı sistem parametreleri olan  $\zeta$  ve  $\omega_n$  cinsinden tam olarak ifade edilebilmekte, diğer zaman domeni kriterleri ise ancak yaklaşık olarak bu büyüklükler cinsinden ifade edilebilmektedir.

Benzer biçimde band genişliği, çınlama frekansı ve çınlama genliği için de matematiksel ilişkiler verilebilmektedir.

Bölüm 2.2.3 ten görüldüğü gibi zaman domeni-frekans domeni geçişi mümkün olmaktadır. İkinci dereceden sistemler için verilen kriterlerin ve aralarındaki ilişkilerin sonuçlarını aşağıdaki gibi verebiliriz :

1° Aşım ve çınlama genliği sadece  $\zeta$ 'nin fonksiyonudur ve artan  $\zeta$  değerleri için azalma gösterirler [(2.8) ve (2.16)]. Aşım ya da çınlama genliğinden biri bilindiğinde diğerine geçiş mümkündür [(2.19) ve (2.20)] :

$0,4 < \zeta < 0,707$  için  $M_p \approx 1 + A$  olduğu söylenebilir [ Tablo E 2.2 ve Kaynak 17 ].

2°  $\omega_A$ ,  $\omega_p$  ve  $\Omega_3$ ; doğal frekans  $\omega_n$  ile doğrudan orantılıdır [(2.10), (2.16b), (2.17)]. Bir başka deyişle,  $\omega_n$  arttıkça aşım zamanı  $T_A$  azalır, buna karşılık çınlama frekansı ve band genişliği artar.  $\omega_n$  sabit tutulduğunda ise  $\zeta$  arttığında  $\omega_A$ ,  $\omega_p$  ve  $\Omega_3$  azalır [ Tablo E 2.2 ].

3°  $\omega_n$  sabit tutulduğunda,  $T_r$  yükselme süresi band genişliğinin tersine olarak  $\zeta$  arttıkça artış gösterir [(2.12) ve Tablo E 2.1].

4° Yükselme süresi band genişliği ile ters orantılıdır [(2.22) ve (2.23)].

Sonuç olarak sistemin basamak cevabında aşım ve aşım zamanının , frekans cevabında  $M_p$  ve  $\omega_p$  nin ölçülmesi ile sistem parametreleri, yani  $\zeta$  ve  $\omega_n$  belirlenebilecektir.

Frekans domeninde ölçmelerin çok daha uzun süre alacağı ve ölçme güçlükleri düşünülürse zaman domeninde çalışmanın uygunluğu açıktır. Ancak sistemden istenenler frekans karakteristiği ile verilmiş ise bu karakteris- tiklerden zaman domenine geçiş mümkündür.

Eğer adaptif olarak kontrol edilecek dizgenin  $\zeta$  ve  $\omega_n$  değerleri- nin belirlenmesi, sistemi karakterize etmek için yeterli ise, adaptasyon için ölçülmesi gereken büyüklükler de ortaya çıkmış olacaktır.

Bir başka deyişle, adaptif olarak kontrol edilecek dizgenin basamak cevabının, ikinci dereceden salınımlı bir davranış olması isteniyor ise, bu durumda aşım ve aşım zamanının ölçülmesi, istenen davranışı oluşturacak biçimde kontrolör parametrelerinin ayarlanması için yeterli olacaktır.

Adaptasyon yönteminin geliştirilmesinde, şimdiye kadar elde ettiğimi- z sonuçlar esas alınmıştır. Ancak, algoritma oluşturulmasında elde edi- len matematiksel ifadelerin kullanılması yoluna gidilmemiş, sadece bu ifa- delerden çıkarılan kavramlardan yararlanılmıştır.

Bundan sonraki bölümlerde, adaptasyon algoritması tanıtılmaya çalı- şılacaktır. Bu tanıtım sırasında da yukarıda elde edilen sonuçlardan nasıl yararlandığımız da gösterilecektir.

## 2.2.5 TEST SINYALI

Yukarıdaki sonuçlardan anlaşıldığı gibi algoritmamızda gerekli olan aşım ve aşım zamanının ölçülmesi için test sinyali basamak biçiminde olma- lıdır. Bu nedenle uygulamamızda basamak fonksiyonu eşdeğeri olan ve siste-



min normal çalışmasına etkisinin integrali sıfır olan kare dalga sinyali kullanılacaktır. Kare dalganın genliği de yine sistemin normal çalışmasını etkilemeyecek ancak gürültüden ayırt edilebilecek seviyede seçilecektir.

### 2.3 ADAPTASYON ALGORİTMASI

Adaptasyon algoritmasını oluşturabilmemiz için öncelikle adaptif olarak kontrol edilecek dizge ve davranışı hakkındaki isteklerin belirlenmesi gerekmektedir. Çalışmamızda ele aldığımız dizge aşağıda verildiği gibidir:

- Dizge verildiğinde istenen davranışa uygun olarak tasarlanmıştır. Ancak zamanla parametre değişmesi ya da çevre koşullarının değişmesi davranış üzerine etkimektedir, bu nedenle adaptif olarak kontrol edilmesi gereklidir.
- Dizgeden istenen davranış, ikinci dereceden salınımlı bir davranıştır. Sistem başlangıçta belirlenen bu davranış biçimini sabit tutacak şekilde kontrol edilecektir.

Buna göre, Bölüm 2 .1 de tanımladığımız Adaptasyon Algoritmasının oluşturulması gerekmektedir. Adaptasyon algoritmasının, adaptasyon çevrimi ve kontrolör yazılımını içerdiğini söylemiştik. Adaptasyon çevriminde yapılacak işlemlerin, kontrolörün yapısına bağlı olacağı göz önüne alınırsa, öncelikle seçilen kontrolörün tanıtılmasında daha sonra da adaptasyon çevriminin temel işlemlerinin nasıl gerçekleştirileceğinin anlatılmasında yarar vardır.

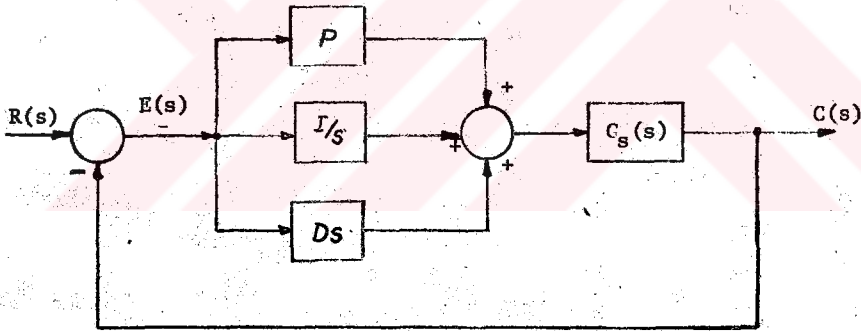
## 2.3.1 KONTROLÖR

Parametre optimizasyonlu kontrolörler içinde en çok kullanılanları P(Oransal), PI(Oransal +İntegral), PD(Oransal +Türevsel) ve PID(Oransal+ İntegral +Türevsel) türleri olduğu için çalışmamızda bu kontrolörler ele alınmıştır.

Bilindiği gibi PID kontrolörlü, tek-giriş tek-çıkışlı sürekli bir kontrol sistemi Şekil 2.7 deki gibi olup, kontrolörün giriş-çıkış ilişkisi;

$$u(t) = P e(t) + I \int_0^t e(\tau) dt + D \frac{de(t)}{dt} \quad (2.24)$$

biçiminde yazılabilir (18).



Şekil 2.7 PID kontrolörlü bir kontrol sistemi

Kontrolörün transfer fonksiyonu ise (2.24 a) daki gibi olacaktır.

$$G_c(s) = \frac{U(s)}{E(s)} = P + \frac{I}{s} + D s \quad (2.24a)$$

Daha önce de belirttiğimiz gibi kontrolör, bir mikrobilgisayar sistemi ile gerçekleştirilecektir. Bu bakımdan (2.24) ifadesinin ayrıklaştırılması gerekmektedir. Küçük örnekleme zamanı  $T_s$  için (2.24) ilişkisi doğrudan doğruya ayrıklaştırma ile bir diferans denklemine dönüştürülebilir. Bunun

için türev elemanı yerine birinci dereceden bir fark alma ve integral yerine bir toplama işlemi getirilir. Sürekli integrasyonun ayırık yaklaşıklığı için dikdörtgen- ya da trapez yaklaşıklığı kullanılabilir ( ).

Dikdörtgen yaklaşıklığı kullanılarak (2.24) ifadesi ayrıklaştırılırsa,

$$u(k) = P e(k) + I T_{\delta} \sum_{i=0}^k e(i) + \frac{D}{T_{\delta}} [e(k) - e(k-1)] \quad (2.25)$$

$$u(t) = u(k) ; kT_{\delta} \leq t \leq (k+1)T_{\delta}$$

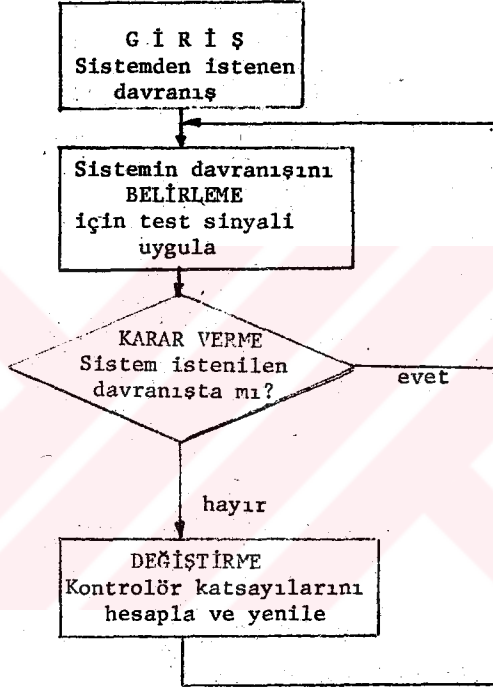
elde edilir.

(2.24) ya da (2.25) ten görüleceği gibi, PID için verilen bu ilişkiden yararlanarak, P, PI, PD kontrolörlerine de geçmek mümkündür. Örneğin,  $P \neq 0$ ,  $I \neq 0$  ve  $D = 0$  için PI kontrolör etkisi elde edilecektir. O halde (2.25) ile verdiğimiz kontrolör gerçekte P, PI, PD ve PID olmak üzere dört tip kontrolörü simgelemektedir. Adaptasyon çevriminde hangi tip kontrolör etkisi gerektiğine karar verildiğinde, P, I ve D katsayılarına uygun değerler verilerek istenen kontrolör elde edilmiş olur.

Yapılan çalışmada ana problem, adaptasyon için kontrolör katsayılarının seçimi olduğu için burada, örnekleme zamanının yeterince küçük olduğu kabul edilmiş ve (2.25) ile verilen kontrolörün çalışmalarımız için yeterli olduğu varsayılmıştır. Ayırık PID kontrolörü için örnekleme zamanının büyüklüğüne ve çeşitli yaklaşıklıklara göre bir çok gerçekleştirme türü Kaynak 4 de verilmiştir.

## 2.3.2 ADAPTASYON ÇEVİRİMİ

Adaptasyon çevriminde yapılması gereken işlemler göz önüne alındığında, Şekil 2.8 deki gibi bir akış diyagramı verilebilir.



Şekil 2.8 Adaptasyon Çevrimine ilişkin akış diyagramı

Akış diyagramından görüldüğü gibi, sistemden istenen davranış giriş büyüklüğüdür. Sisteme test sinyali uygulanarak o andaki davranışı saptanacak (Belirleme) ve girişteki bilgi ile karşılaştırılacaktır (karar verme). Eğer sistem istenilen biçimde davranmıyorsa kontrolör katsayıları yeniden hesaplanarak yenilenir ve tekrar belirleme işlemine geçilir.

Akış diyagramında gösterilen her blok aşağıda tek tek ele alınarak anlamları ve blok içinde yapılan işlemler açıklanmıştır.

### 2.3.2.1 GİRİŞ AŞAMASI

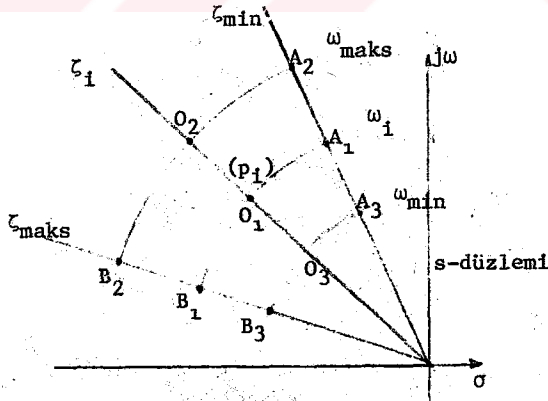
Adaptif çevrim yazılımında giriş büyüklüğü olarak gösterdiğimiz, aslında bizim sistemden istediğimiz davranışı simgelemektedir. Bölüm başında istenen davranışın ikinci derece benzeri bir davranış olduğunu belirtmiştik. Bu ise dizgenin matematiksel modelinin yaklaşık olarak (2.1) transfer fonksiyonu ile verilebileceğini, bir başka deyişle dizgenin,

$$P_{i1,2} = -\zeta \omega_n \pm j \omega_n \sqrt{1 - \zeta^2} \quad (2.26)$$

$(0 < \zeta < 1)$

kontrol kutupları ile karakterize edilebileceğini göstermektedir.

Sistemin istenen davranışta olduğu durumda  $P_i$  kontrol kutuplarına ilişkin sönüm katsayısı  $\zeta_i$  ve doğal frekansı  $\omega_{ni}$  ile gösterildiğine göre sistemde oluşacak değişiklikleri, sönüm katsayısı ve doğal frekans cinsinden ifade etmek mümkündür. Kontrol kutuplarının değişimi,  $P_i$  etrafındaki bir bölgede, Şekil 2.9 da gösterildiği gibidir.



Şekil 2.9 Kontrol kutuplarının değişim bölgesi

(2.8) ve (2.9) ifadelerinden görüleceği gibi,  $\zeta$  ve  $\omega_n$  in değişimi sistemin basamak cevabında Aşım ve Aşım Zamanının değişmesi biçiminde göz-

lenebilir. O halde adaptif çevrimin girişi; İstenen Aşım  $A_1$  ve İstenen Aşım Zamanı  $T_{A_1}$  olacaktır.

### 2.3.2.2 BELİRLEME AŞAMASI

Belirleme işlemi, sistemin basamak cevabının izlenerek, aşım ve aşım zamanının ölçülmesi biçiminde gerçekleştirilecektir. Şekil 2.4 ten görüldüğü gibi  $c(t)$  çıkış işaretinin örnek değerleri  $c(kT_0)$  elde edilmektedir.  $A$  ve  $T_A$  değerleri bu örneklerden yararlanılarak bulunur. Bunun için yazılacak program parçası, bu örneklerden en büyüğünün aranması ve bulunan değer kaçınıcı örnek olduğunun saptanması biçiminde olacaktır. Örnekleme zamanı  $T_0$  olduğuna göre ve  $k_1$  'inci örnek en büyük çıkış değerini göstermekte ise,

$$T_A^* = k_1 T_0 \quad (2.27)$$

$$A^* = c(k_1 T_0) - 1 \quad (2.28)$$

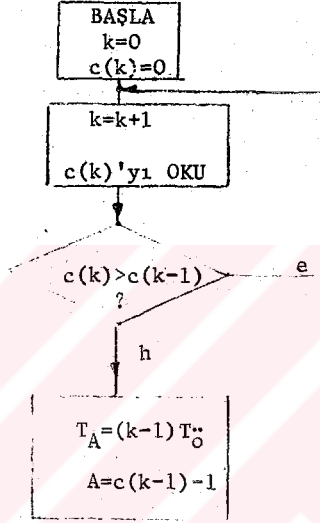
biçiminde hesaplanacak  $T_A^*$  ve  $A^*$  değerleri, yaklaşık olarak aşım ve aşım zamanı olacaktır. Bulunan değerlerin gerçek değerlere yakınlığının örnekleme zamanına bağlı olduğu açıktır. Örnekleme zamanının etkisi ve seçimi ayrı bir bölümde tartışılacağından, burada seçilen örnekleme zamanının yeterli sıklıkta olduğu düşüncesiyle,

$$A \approx A^*$$

$$\text{ve } T_A \approx T_A^*$$

olduğu kabul edilecektir.  $A$  ve  $T_A$  nın ölçülmesi, yani belirleme işlemine ilişkin akış diyagramı Şekil 2,10 da verilmiştir.

Burada dikkat edilirse, belirleme işlemi yapılırken ayrıca ölçmeler yapılmamakta; kontrol çevriminde zaten var olan çıkış örneklerinden yararlanılmaktadır.



Şekil 2.10 Belirleme işlemine ilişkin akış diyagramı

### 2.3.2.3 KARAR VERME AŞAMASI

Belirleme işlemi tamamlandıktan sonra, sistemin istenen davranışta olup olmadığına karar verilir. Bu ise bir karşılaştırma işlemidir.

$$D1 \triangleq | A - A_1 | \quad (2.29)$$

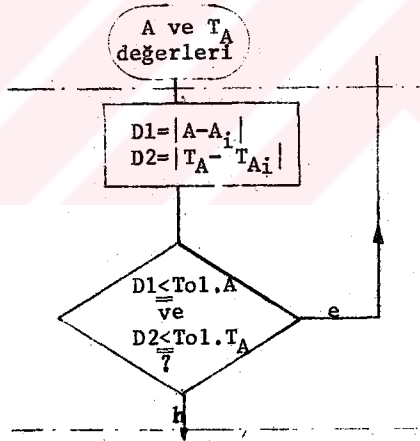
$$D2 \triangleq | T_A - T_{A_1} | \quad (2.30)$$

olmak üzere , sistemin istenen davranışta olması  $D1$  ve  $D2$  nin önceden

belirlenecek tolerans sınırları içinde kalmasına karşılık düşecektir. Yani aşım için verilecek tolerans  $Tol.A$  ve aşım zamanı için verilecek tolerans  $Tol.T_A$  ile gösterilirse,

$$( D1 \leq Tol.A \text{ VE } D2 \leq Tol.T_A ) \quad (2.31)$$

koşulu sağlandığında, sistemin istenen davranışta olduğu kabul edilecektir. Bir başka deyişle (2.31) koşulu sağlandığında, sistem adapte edilmiş sayılacaktır. Buna göre karar verme bloğu için de Şekil 2.11 deki gibi bir akış diyagramı verebiliriz.



Şekil 2.11 Karar Verme işlemine ilişkin akış diyagramı



## 2.3.2.4 DEĞİŞTİRME AŞAMASI

Bu aşamada, ölçülen aşım ve aşım zamanının, istenen  $A_1$  ve  $T_{A_1}$  değerlerine göre durumu göz önüne alınarak, sisteme getirilmesi gereken kontrolör tipine karar verilerek katsayıları hesaplanır.

Kontrolör katsayılarının değiştirilmesi için ilk akla gelen, katsayılara küçük artırımlar vermek ve elde edilen sonuçlara göre iteratif olarak katsayı değiştirmelerine devam ederek sistemi istenen davranışa getirmektir.

Ancak hemen görülebilir ki, iterasyon sayısı, sistemdeki değişiklik miktarına, ayarlanacak parametre sayısına ve seçilen adımın uzunluğuna bağlıdır. Örneğin, R.N. Clark ve E.E. Gould'un verdiği bir adaptasyon modelinde dört parametre için iterasyon sayısı ortalama 18'dir (19). Amaç adaptasyon olduğunda optimum noktanın bulunmasının mümkün olduğu kadar hızlı olması istenecektir. Geliştirilen algoritmada, başlangıçta kontrolör katsayılarının sistem davranışına etkinliği saptanmakta ve katsayı yenilemede öğrenilen bilgilerden yararlanılarak adım uzunluğu hesaplanmaktadır. Bir diğer özellik ise katsayılardan hangilerinin öncelikle değiştirileceğine karar verilmesidir. Bunun için öncelikle Bölüm 2.3.1 de verdiğimiz kontrolörün (P,I,D) katsayılarının dizge üzerine etkilerini belirlememiz gerekmektedir. Aşağıda Şekil 2.7 ile verdiğimiz kontrol sisteminde, dizge transfer fonksiyonu,

$$G_s(s) = \frac{K}{s(s+a)} \quad (2.32)$$

biçiminde seçilerek, (P,I,D) katsayılarının etkileri incelenmiştir.

1° P y a d a K a z a n ç E t k i s i :

(2.24a) transfer fonksiyonunda  $I=0$  ve  $D=0$  alınırsa, P kontrolörün transfer fonksiyonu,

$$G_p(s) = P \quad (2.33)$$

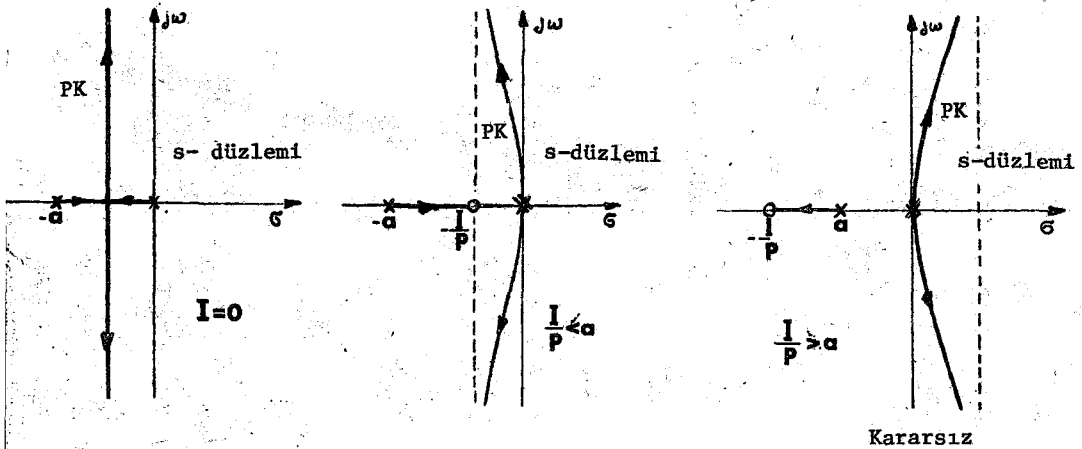
biçiminde olacaktır. Bu ise kontrol sisteminde ileri yola konan bir kazanç elemanı olup, bilindiği gibi kazancın artması  $\zeta$  'yı küçültür,  $\omega_n$  'i artırır. Dolayısıyla aşımı artıracak ve aşım zamanını azaltacaktır.

2° I ya da İntegral Etkisi :

(2.32) de,  $D=0$  alınırsa,

$$G_{PI}(s) = P + \frac{I}{s} = P \frac{s + \frac{I}{P}}{s} \quad (2.34)$$

biçiminde PI tipi kontrolör elde edilecektir. Görüldüğü gibi PI, sisteme  $s_k = 0$  da bir kutup (integral etkisi),  $s_o = -I/P$  de bir sıfır getirmektedir.  $s_o$  'ın yeri I katsayısı ile değiştirilerek sisteme getirilen integral etkisi kontrol edilebilmektedir. I katsayısının artması integral etkisini artıracığından,  $\zeta$  küçülerek aşım artacaktır. İntegral etkisinin yerleşik durum yanılığını düzeltici yönde olmasına karşılık, sistemi daha kararsız kılacağı açıktır. Ele aldığımız dizgeye, İntegral etkisini Şekil 2.12 deki geometrik yer eğrisi ile göstermek mümkündür.



Şekil 2.12 İntegral etkisi

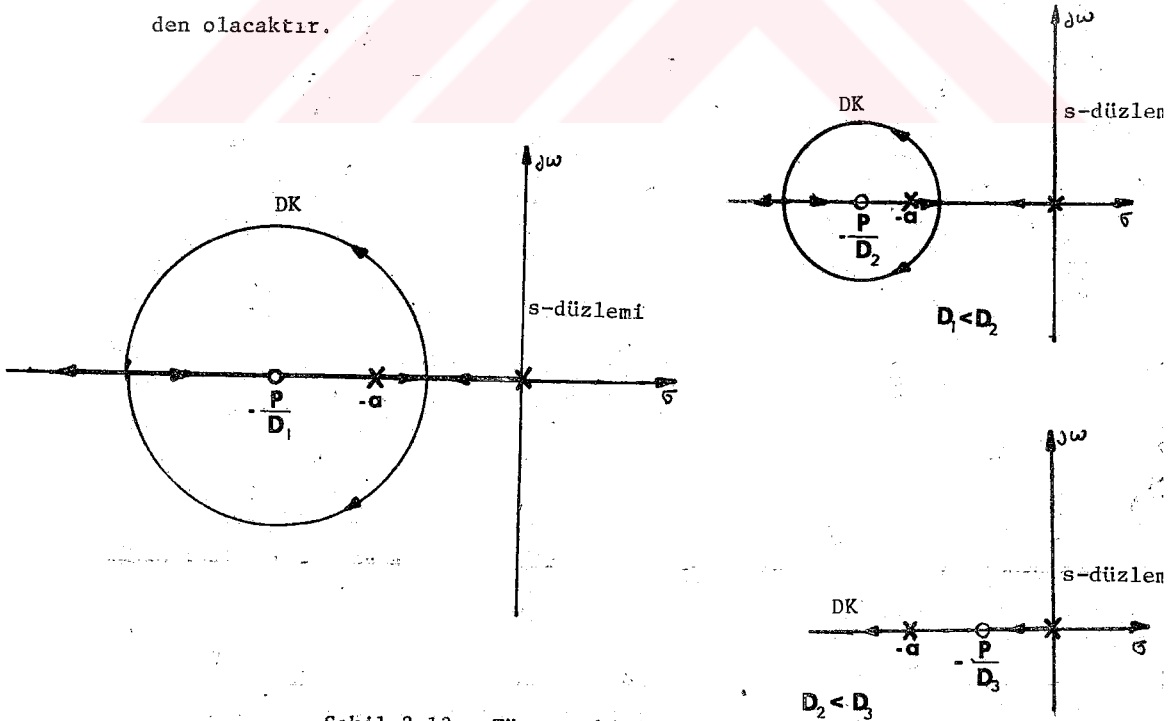
Şekilden görüldüğü gibi,  $s_0$  'in açık çevrim kutbundan ötede olması halinde sistem kararsız olmaktadır. Ancak gerçekleştirilen algoritma gereği böyle bir duruma gelinmeyecektir.

3° D ya da Türev Etkisi :

(2.32) de  $I=0$  alınarak,

$$G_{PD}(s) = P + D s \quad (2.35)$$

olacaktır. Burada sisteme bir sıfır,  $s_0 = -P/D$  eklenmektedir. Bu durumda sistemin band genişliği artar, sistem cevabı hızlanır, aşımı azalır, kararlılığı artar. D katsayısının etkisi, Şekil 2.13 deki gibi gösterilebilir. Burada integralin kararsızlık yönündeki etkisine karşılık olarak, D katsayısının çok artması, sistem davranışının birinci dereceden olmasına neden olacaktır.



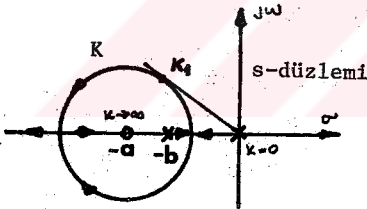
Şekil 2.13. Türev etkisi

Bu incelemeden yararlanarak edineceğimiz sonuçlar şöyledir:

P katsayısı A ve  $T_A$  üzerine doğrudan etkilidir. I ve D katsayılarının etkileri ise birbirinin tersi biçiminde ve daha çok aşım üzerine olmaktadır. Bu nedenle değiştirme algoritması öncelikle P katsayısı üzerinde çalışacak ve bunun ardından I ya da D nin gerekliliğini saptayarak uygun I ya da D değerini bulacaktır.

Yukarıdaki incelemeden edindiğimiz P,I,D etkileri uygulamada çokça rasladığımız dizgeler için geçerlidir. Ancak bu etkilerin yönü, bazı dizgeler için çalışma noktasına bağlı olarak değişebilir. Örneğin,

$G_S(s) = \frac{K(s+a)}{s(s+b)}$  biçiminde bir dizge ele alındığında, Şekil 2.14 de verilen geometrik yer eğrisinden görüleceği gibi, kazancın belli bir değerinden sonra kazanç artışı söylediğimiz tersine, aşımı azaltma yönünde etki edecektir.



$K < K_1$  için kazanç artışı aşımı artırır.  
(Umulan etki)  
 $K > K_1$  için kazanç artışı aşımı azaltır.  
(Ters yönde etki)

Şekil 2.14  $G_S(s) = \frac{K(s+a)}{s(s+b)}$  için geometrik yer eğrisi

Ancak bu tür aykırılıklar, algoritmamızda göz önüne alınarak şu şekilde çözümlenmiştir:

Adaptasyonun ilk adımında yani öğrenme sürecinde, P,I ve D nin yapacağı etkilerin bilindiği varsayılır. Etkinin yönü ise bu aşamada yapılacak test sonuçlarına göre belirlenir. Eğer umulan etki yönünde olmayan bir sonuç alınır, algoritma bunu göz önüne alarak değerlendirme yapar. Aşağıda algoritmada yapılacak işler ve katsayı hesaplanması adım adım verilmiştir.

## 1. ADIM :

Başlangıçta sistemin istenen biçimde davrandığı bilinmektedir. Bu durumdaki kontrolör katsayılarının normalize değerleri  $P=1$ ,  $I=0$  ve  $D=0$  olarak alınacaktır. Bu adımda sisteme basamak girişi (Test sinyali) uygulanarak istenen aşım  $A_1$  ve istenen aşım zamanı  $T_{A1}$  ölçülür (GİRİŞ).

## 2. ADIM : (Öğrenme Aşamasında yapılacak işlemler)

Bu adımda, kontrolör katsayılarının  $A$  ve  $T_A$  üzerine etkileri ve etki yönleri saptanır. Bunun için  $P, I, D$  katsayılarına sırayla küçük artımlar verilerek  $A$  ve  $T_A$  değerleri ölçülür.

### P ETKİSİNİ ÖĞRENME :

- (1)  $(1 + \Delta P_0, 0, 0)$  için Test uygulanır,  $A$  ve  $T_A$  belirlenir.

Bu değerler  $\Delta P_0$  artımına karşılık etkiyi verecektir.

- (2)  $|A - A_1| \leq Tol.A$  VEYA  $|T_A - T_{A1}| \leq Tol.T_A$  ise  $P_0$  artımı yetersizdir.  $\Delta P_0 \leftarrow \Delta P_0 + \Delta P_0$  alınarak (1)'e dönülür.

- (3) Çözlenebilir bir etki elde edildiğinde,

$$ORAN P_A = \frac{A - A_1}{\Delta P_0} \quad (2.36)$$

$$ORAN P_{T_A} = \frac{T_{A1} - T_A}{\Delta P_0} \quad (2.37)$$

oranları hesaplanır.

- (4)  $\Delta P_0 > 0$  için  $A > A_1$  ve  $T_A < T_{A1}$  ise dizge umulan davranıştır. Aksi halde etki ters yönde değerlendirilir.

### I ETKİSİNİ ÖĞRENME :

- (1)  $(1, \Delta I_0, 0)$  için Test uygulanır,  $A$  ve  $T_A$  belirlenir.

Bu değerler  $\Delta I_0$  artımına karşılık etkiyi verecektir.

(2)  $|A - A_1| \leq Tol.A$  ise  $\Delta I_0$  artımı yetersizdir.  $\Delta I_0 + \Delta I_0 + \Delta I_0$  alınarak (1)'e dönülür.

(3) Gözlenebilir bir etki elde edildiğinde,

$$ORAN \ I_A = \frac{A - A_1}{\Delta I_0} \quad (2.38)$$

hesaplanır.

(4)  $\Delta I_0 > 0$  için  $A > A_1$  ise dizge umulan davranıştaadır. Aksi halde etki ters yönde değerlendirilir.

#### D ETKİSİNİ ÖĞRENME :

(1)  $(1,0, \Delta D_0)$  için Test uygulanır, A ve  $T_A$  belirlenir.

Bu değerler  $\Delta D_0$  artımına karşılık etkiyi verecektir.

(2)  $|A - A_1| \leq Tol.A$  ise  $\Delta D_0$  artımı yetersizdir.  $\Delta D_0 + \Delta D_0 + \Delta D_0$  alınarak (1)'e dönülür.

(3) Gözlenebilir bir etki elde edildiğinde,

$$ORAN \ D_A = \frac{A_1 - A}{\Delta D_0} \quad (2.39)$$

hesaplanır.

(4)  $\Delta D_0 > 0$  için  $A < A_1$  ise dizge umulan davranıştaadır. Aksi halde etki ters yönde değerlendirilir.

Öğrenme işlemleri bu şekilde tamamlandıktan sonra, katsayılar tekrar

$(1,0,0)$  konumuna getirilir.

#### 3. A D İ M :

Sisteme tekrar belirleme ve karar verme işlemleri için Test uygulanır.

(2.31) koşulu sağlandığı sürece 3. Adım tekrarlanır. Koşulun sağlanmaması

durumunda, değiştirme işlemlerine geçilir ve aşağıdaki biçimde kontrolör

katsayıları hesaplanarak yenilenir.

#### 4. A D I M : ( Değişirme aşamasında yapılacak işlemler)

Dizgedeki değişmelerin sonucunda  $A$  ve  $T_A$  üzerinde izlenebilecek mümkün değişiklikleri gidermek üzere algoritma, katsayı değiştirmelerini Tablo 2.2 deki biçimde yapacaktır.

Mimkün Durumlar	$A_{\underline{}}$	$A_{\uparrow}$	$A_{\downarrow}$
$T_A =$	İSTENEN DURUM	İntegral azalt ( $I_{\downarrow}$ ) ya da Türev artır ( $D_{\uparrow}$ )	İntegral artır ( $I_{\uparrow}$ ) ya da Türev azalt ( $D_{\downarrow}$ )
$T_A \uparrow$	Kazanç artır ( $P_{\uparrow}$ ) Buna göre Beklenen Aşım değerini hesaplayarak birinci satırda uygun sütuna geç		
$T_A \downarrow$	Kazanç azalt ( $P_{\downarrow}$ ) Buna göre Beklenen Aşım değerini hesaplayarak birinci satırda uygun sütuna geç		

Tablo 2.2 Değişirme işlemleri

Tabloda,  $A_{\underline{}}$  sembolü aşımın istenen değerde olduğunu,  
 $A_{\uparrow}$  sembolü aşımın istenenden fazla olduğunu,  
 $A_{\downarrow}$  sembolü aşımın istenenden az olduğunu,  
 $T_{A_{\underline{}}}$  sembolü aşım zamanının istenen değerde olduğunu,  
 $T_{A_{\uparrow}}$  sembolü aşım zamanının istenenden fazla olduğunu,  
 $T_{A_{\downarrow}}$  sembolü aşım zamanının istenenden az olduğunu göstermektedir.

Algoritma öncelikle istenen aşım zamanı değerini yani  $T_{A_{\underline{}}}$  konumunu elde etmeye çalışacaktır. Örneğin, Test sonucunda tabloda ( $A_{\downarrow}$ ,  $T_{A_{\downarrow}}$ ) durumuna gelinmiş ise, öncelikle  $P$  katsayısı azaltılacaktır. Kazancın azaltılmasına karşılık, aşım da bir değişiklik olacağı açıktır. Bu bakımdan bu kazanç değişiminin yapacağı aşım etkisi öngörülerek, Beklenen Aşım  $A_{BEK}$

hesaplanmalıdır.  $A_{BEK}$  değeri,  $A_1$  ile karşılaştırılır, tablonun birinci satırında, karşılaştırmadan elde edilen sonuca göre, uygun sütundaki işlemler yapılarak, gerekiyorsa  $I$  ya da  $D$  değişimleri hesaplanır.

Katsayılar hesaplandıktan sonra 3. Adım 'a geçilecektir.

$P, A_{BEK}, I$  ve  $D$  büyüklüklerinin hesaplanması :

Algoritma;  $P_0, I_0, D_0$  bir önceki katsayı değerlerini göstermek üzere,  $P, I$  ve  $D$  'nin yeni değerlerini,

$$P = P_0 + \Delta P \quad (2.40)$$

$$I = I_0 + \Delta I \quad (2.41)$$

$$D = D_0 + \Delta D \quad (2.42)$$

biçiminde hesaplayacaktır.  $\Delta P, \Delta I, \Delta D$  değerleri ise Tablo 2.2 de gösterildiği gibi eğer gerekli bulunuyorsa hesaplanacak aksi halde sıfır olarak alınacaktır. Bu artımların hesaplanmasında aşağıdaki formüller kullanılacaktır:

$$\Delta T_A = T_A - T_{A_1} \quad (2.43)$$

$$\Delta A = A_1 - A \quad (2.44)$$

olmak üzere,

$$\Delta P = \frac{\Delta T_A}{\text{ORAN } P_{T_A}} \quad (2.45)$$

$$A_{BEK} = \Delta P \cdot \text{ORAN } P_A + A \quad (2.46)$$

$$\Delta A_{BEK} = A_1 - A_{BEK} \quad (2.47)$$

$$\Delta D = \frac{\Delta A}{\text{ORAN } D_A} P \quad \text{ya da} \quad \Delta D = \frac{\Delta A_{BEK}}{\text{ORAN } D_A} P \quad (2.48)$$

$$\Delta I = \frac{\Delta A}{\text{ORAN } I_A} P \quad \text{ya da} \quad \Delta I = \frac{\Delta A_{BEK}}{\text{ORAN } I_A} P \quad (2.49)$$

olarak gerekli değişimler hesaplanacaktır.





**3** . bölüm

**SİMÜLASYON**

Bu bölümde, daha önceki bölümde tanıtıma çalıştığımız adaptasyon yöntemine ilişkin simülasyon çalışmaları verilecek ve elde edilen sonuçlar tartışılacaktır. Öncelikle simülasyon çalışmasında kullandığımız bilgisayar sisteminin teknik özellikleri kısaca verilecektir. Ayrıca bu bilgisayar sisteminin çalışmamıza getirdiği sınırlandırmalar ortaya konularak örnekleme zamanının ve adaptif olarak kontrol edilecek dizge örneğinin nasıl seçildiği anlatılacaktır.

Daha sonra Bölüm 2.3.2 de verilen, kontrolörün P,I,D katsayılarının dizgeye etkilerine ilişkin deneyler, öğrenme aşamasında söz konusu olan başlangıç artımları  $\Delta P_o$ ,  $\Delta I_o$ ,  $\Delta D_o$ 'ın hesaplanması ve doğruluk deneyleri verilecektir.

Bölüm sonunda ise tanıtılan algoritmanın işleyişi ve programlanmasına ilişkin özellikler ve bu algoritma kullanılarak yapılan simülasyon sonuçları yer alacaktır.

### 3.1 SİMÜLASYONDA KULLANILAN BİLGİSAYAR SİSTEMİ

Simülasyon çalışmalarımızı gerçekleştirdiğimiz bilgisayar, EAI-500 Melez-Bilgisayar sistemidir. Şekil 2.4 ile verdiğimiz adaptif kontrol sistemi modelinde, dizge Analog Bilgisayarda, adaptif kontrolör ise Dijital Bilgisayarda simüle edilmiştir. Kullandığımız bu Melez-Bilgisayar sisteminin teknik özellikleri kısaca aşağıdaki gibidir (20,21)

## 1° Dijital Bilgisayar : (EAI PACER 100)

## A- Donanım Özellikleri :

Sözcük Uzunluğu : 16 bit  
Bellek Kapasitesi : 32 K x 16 bit  
Saat Frekansı : 5 MHz

## B- Yazılım Özellikleri :

FORTRAN IV  
HOI (Hytran Operating Interpreter)  
ASSEMBLER

## 2° Analog Bilgisayar : (EAI PACER 500)

## A- Elemanlar :

İşlem Devreleri  
İntegratörler  
Katsayı Vericiler  
Çarpma Devreleri  
Lojik Elemanlar  
Komperatörler  
Limiterler  
İzle/Tut devreleri  
Analog Anahtar Devreleri

B- Doğruluk :  $10^{-4}$  volt mertebesinde

## 3° Ara Birimler :

A/D ve D/A Çeviriciler : 16 bit BCD -  $\pm 10$  volt

#### 4° B i l g i A k t a r m a :

##### Dijital Bilgisayardan - Analog Bilgisayara (YAZMA İŞLEMİ):

Mevcut sistemde, dijital Makinadan analog makinaya Bilgi gönderilmesi, FORTRAN ya da ASSEMBLER dili çalışma ile analog makinadaki RDAC çıkışı yardımıyla gerçekleştirilmektedir.

##### Analog Bilgisayardan - Dijital Bilgisayara (OKUMA İŞLEMİ):

Okuma işlemine ilişkin tarafımızdan yapılan program Ek 3.1 de verilmiştir. Bilgi aktarma ya da bir analog değerın dijital bilgisayar tarafından okunup tekrar analog bilgisayara aktarılması için geçen süre,  $T_{oy} \approx 150$  ms olarak ölçülmüştür.

### 3.2 ÖRNEKLEME ZAMANI VE DİZGE SECİMİ

Yukarıda belirttiğimiz gibi, örnekleme zamanının, dijital makinanın gecikmesinin ihmal edilebileceği mertebede seçilmesi gerekmektedir. Kontrolör yazılımının işletim süresinin yaklaşık 5 ms mertebesinde olduğu düşünülürse, örnekleme zamanının,  $T_{oy} \approx 150$  ms yanında büyük olmasını sağlamak yeterli olacaktır. Kabul edilebilir bir değer olarak örnekleme zamanı için

$$T_0 = 1 \text{ s}$$

seçilmiştir.

Bu teknik sınırlamalar, simüle edilecek dizgenin seçimini de etkilemiştir. Yapılan bir çalışmada örnekleme zamanı ile dizgenin  $\Omega_3$  band genişliği arasında,

$$T_0 = \frac{\sqrt{2}}{\Omega_3} Y_{\max} \quad (3.1)$$

ilişkisi verilmiştir (22). Burada,  $Y_{\max}$ , katlanılabilecek en büyük örnek-  
leme yanılığını göstermektedir.

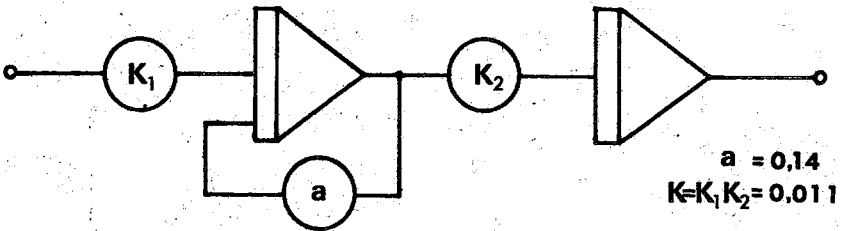
Örnekleme zamanı 1 s seçildiğine göre, katlanılabilecek en büyük yanılga % 10 alınır (3.1) den,

$$\Omega = \frac{\sqrt{2}}{1} 0,1 = 0,141$$

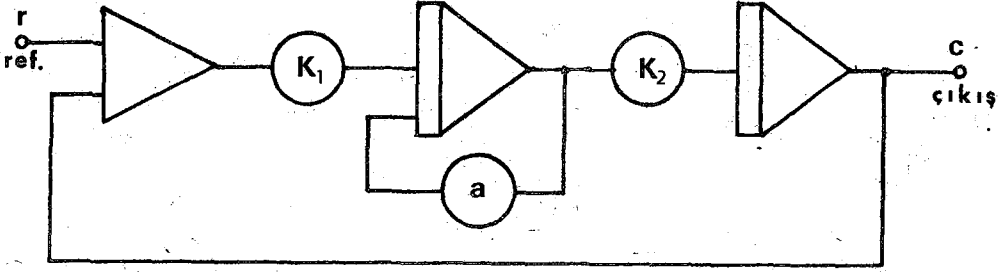
olarak bulunur. Buna göre, simülasyon çalışması için ele alınacak dizgenin band genişliğinin bu mertebede olması gerekmektedir. Örnek dizge olarak, transfer fonksiyonu (2,32) ile verilen biçimde bir dizge seçilmiştir. Kontrolör katsayılarının (1,0,0) olduğu durumda, istenen davranışı veren dizge katsayıları,  $K = 0,011$  ve  $a = 0,14$  olarak seçilmiştir. Bu durumda dizge transfer fonksiyonu,

$$G_s(s) = \frac{0,011}{s(s + 0,14)} \quad (3,2)$$

biçiminde olacaktır. Bu transfer fonksiyonunun analog makinada simülasyonu ya da analog programı, Şekil 3.1 de verilmiştir. Şekil 3.2 de ise, Şekil 2.7 de verdiğimiz sürekli kontrol sisteminde, kontrolör katsayıları (1,0,0) için, yani kontrolörsüz sürekli durum için birim basamak girişine karşılık kapalı çevrim cevabının elde edileceği analog program verilmiştir.



Şekil 3.1  $G_s(s)$  için analog program



Şekil 3.2 Kontrolörsüz sürekli sistem için analog program

Kontrolörsüz sürekli sistemin birim basamak cevabına ilişkin büyüklükler Bölüm 2.2 de verilen ilişkiler kullanılarak hesaplanırsa, istenen davranış için,

$$\omega_n = 0,1056 \text{ r/s}$$

$$\zeta = 0,6627$$

$$A = 0,062$$

$$T_A = 39,71 \text{ s}$$

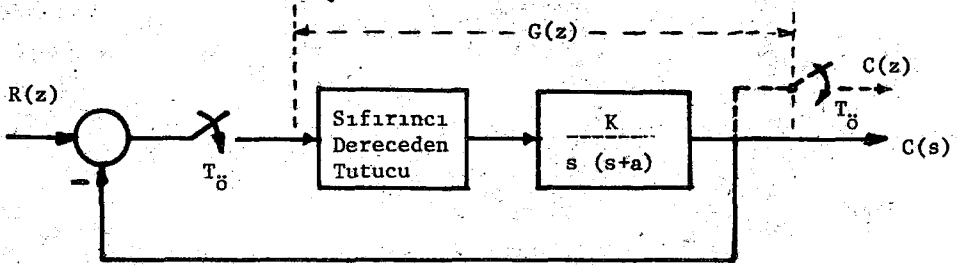
$$\omega_A = 0,08 \text{ r/s}$$

$$\Omega_3 = 0,1636 \text{ r/s}$$

(3.3)

bulunur.

Bölüm 2.1 de de belirttiğimiz gibi, çalışmada test sinyali (giriş: birim basamak sinyali), giriş ile çıkışın karşılaştırılması ve kontrolör, dijital bilgisayarda simüle edileceğinden, katsayıların (1,0,0) durumu için yani, kontrolörsüz ayırık durum için Şekil 3.3 deki gibi bir ayırık, matematiksel model verilebilir.



Şekil 3.3 Kontrolörsüz ayırık sistem modeli

Bu durumda, sıfırıncı dereceden tutucu kullanılarak  $G(z)$  hesaplanırsa,

$$G(z) = Z \left\{ \frac{1 - e^{-sT_0}}{s} \cdot \frac{K}{s(s+a)} \right\} = \frac{K}{a} \left[ \frac{T_0}{z-1} - \frac{1 - e^{-aT_0}}{a(z - e^{-aT_0})} \right] \quad (3.4)$$

bulunur. Ele alınan sistem ve  $T_0 = 1$  s için istenen durumda çıkış işareti hesaplanırsa,

$$C(z) = \frac{0,00533 z^2 + 0,00508 z}{z^3 - 2,864 z^2 + 2,7385 z - 0,6744} \quad (3.5)$$

olacaktır.(13). (3.5) ifadesinden çıkışın maksimum değeri hesaplanırsa,

$$c(39T_0) \Big|_{\text{maks}} = 1,07516$$

olarak bulunur, buna göre aşım ve aşım zamanı da,

$$A = 0,07516 \quad \text{ve} \quad T_A = 39 \text{ s}$$

olacaktır. Bilgisayar simülasyonu sonucunda ise, maksimum değer  $k = 34$  üncü

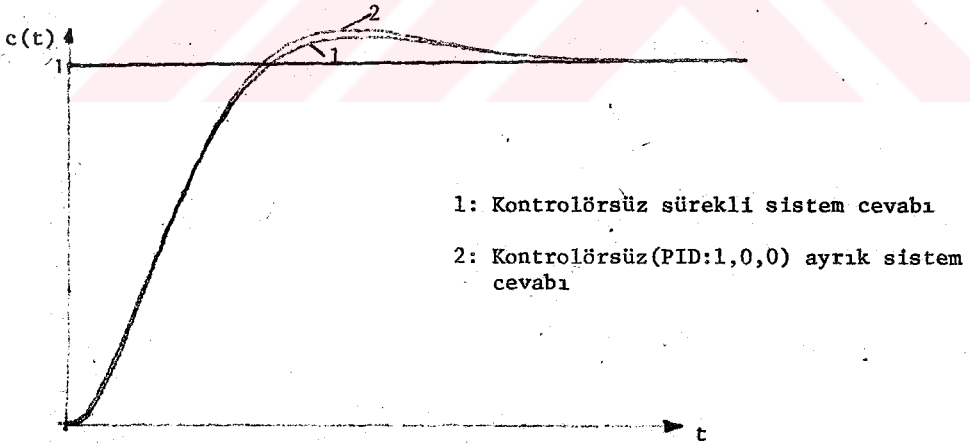
örnekte  $c_{\text{maks}} = 1,081$  olarak ölçülmüştür. (3.4) ifadesinde  $T_0 = 1,15$  s alınır ve  $C(z)$  hesaplanırsa,

$$C(z) = \frac{0,00699 z^2 + 0,0066 z}{z^3 - 2,844 z^2 + 2,702 z - 0,8579} \quad (3.6)$$

olur ve maksimum değer  $k=34$  üncü örnekte,  $c_{maks} = 1,078$  olarak bulunur.

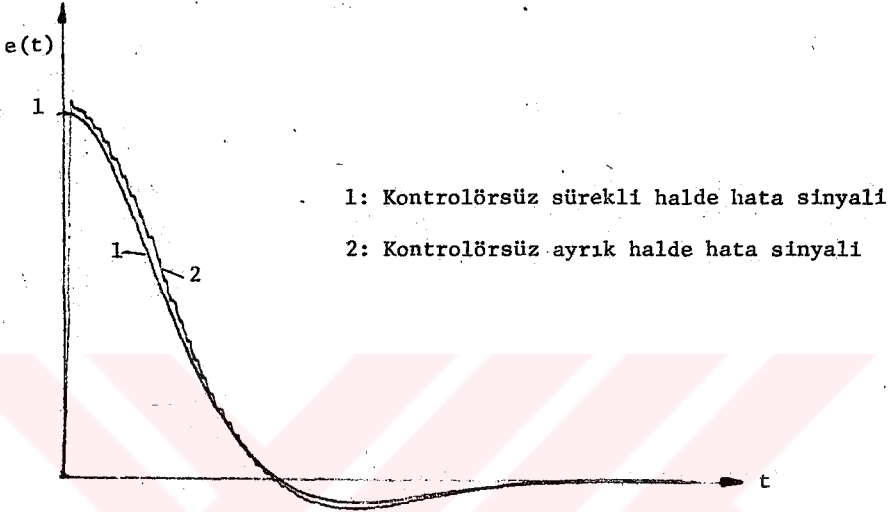
Bu ise sistemin matematiksel modelinde,  $T_g = 1,15$  s alınması ve bilgisayar gecikmesi dediğimiz 150 ms nin de göz önüne alınması gerektiğini göstermektedir. Şekil 3.4 ve Şekil 3.5 te sırasıyla, sürekli ve ayrık simülasyon için, dizge istenen durumda iken sistem cevabı ve hata sinyallerine ilişkin çizimler verilmiştir.

Şekilden de görüleceği gibi seçilen örnekleme zamanı, yeterince küçük olarak kabul edilebilecektir.



Şekil 3.4 Sürekli ve ayırık simülasyon sonucunda elde edilen sistem cevabı çizimleri



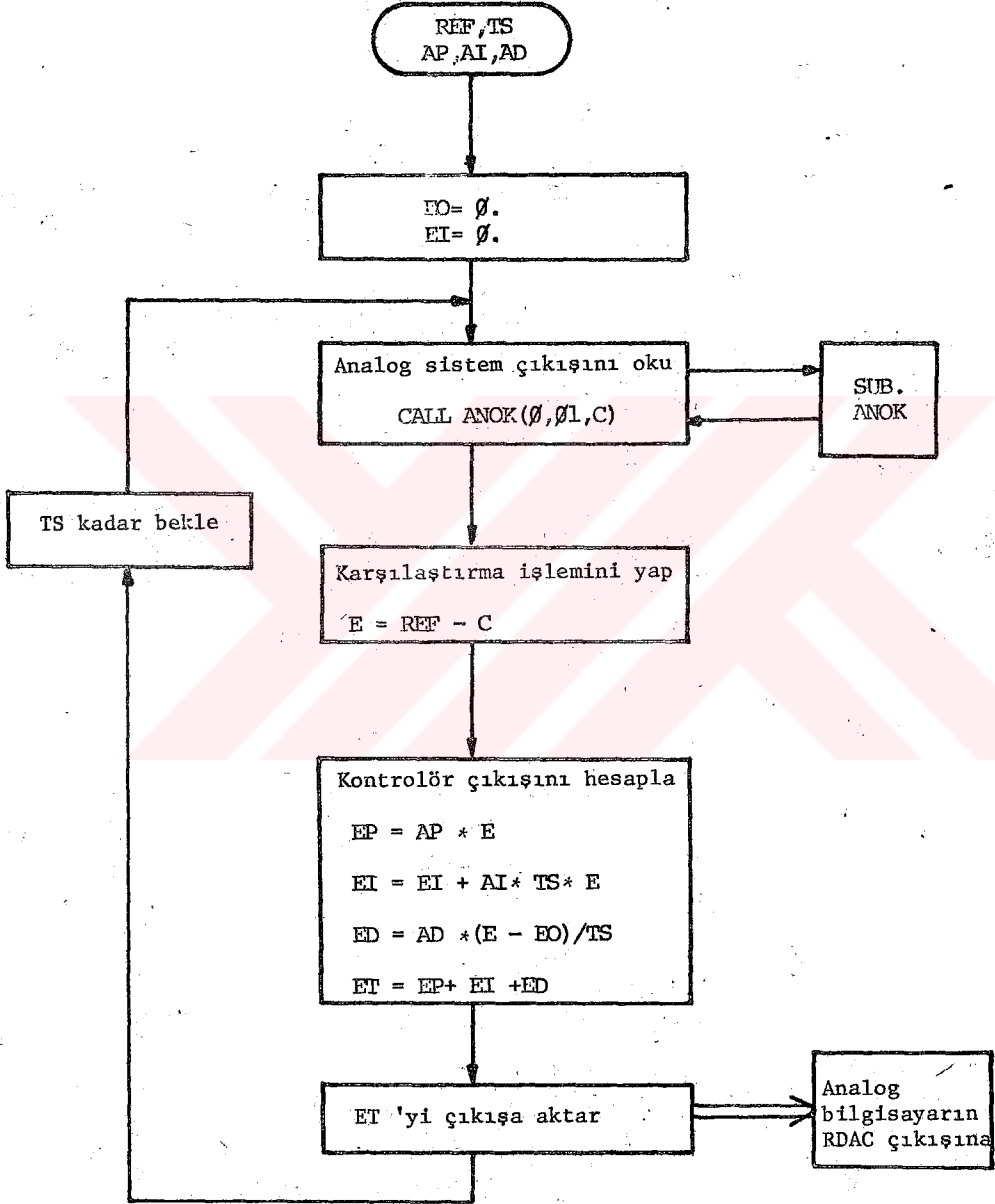


Şekil 3.5 Sürekli ve ayırık simülasyon sonucunda elde edilen hata sinyalleri çizimleri

### 3.3 KONTROLÖR SİMÜLASYONU VE (P,I,D) ETKİLERİNE İLİŞKİN DENEYLER

#### 3.3.1 KONTROLÖR SİMÜLASYONU

Örnekleme zamanının ele alınan sistem için yeterli küçüklükte olduğu görüldüğünden, kontrolör (2.25) ifadesinde verilen biçimde ayrıklaştırılarak dijital bilgisayarda simüle edilmiştir. Giriş işareti, girişle çıkışın karşılaştırılması ve kontrolöre ilişkin olarak FORTRAN IV dilinde yazılan DEJİTAL PID KONTROLÖR (Bkz. Ek 3.2) programına ilişkin akış diyagramı Şekil 3.6 da verilmiştir.



Şekil 3.6 . Dijital Bilgisayarda gerçekleştirilen işlemlere ilişkin akış diyagramı.

Akış diyagramında gösterilen büyüklükler, programda kullanılan değişkenler olup, (2.25) ifadesi de göz önüne alınırsa, bu değişkenlerin anlamları aşağıdaki gibidir :

- C : Analog sistem çıkışı  
 REF : r, test sinyali, birim basamak girişi (R:sabit)  
 TS :  $T_g$  örnekleme zamanı  
 AP : Kontrolörün P katsayısı  
 AI : Kontrolörün I katsayısı  
 AD : Kontrolörün D katsayısı  
 EO :  $e(k-1)$ , hatanın ya da kontrolör girişinin bir önceki örnek değeri  
 E :  $e(k)$ , o andaki hata ya da o andaki kontrolör girişinin örnek değeri  
 ET :  $u(k)$ , o andaki kontrolör çıkışı  
 ANOK : Analog değeri okuma  
 RDAC : Analog değeri yazma

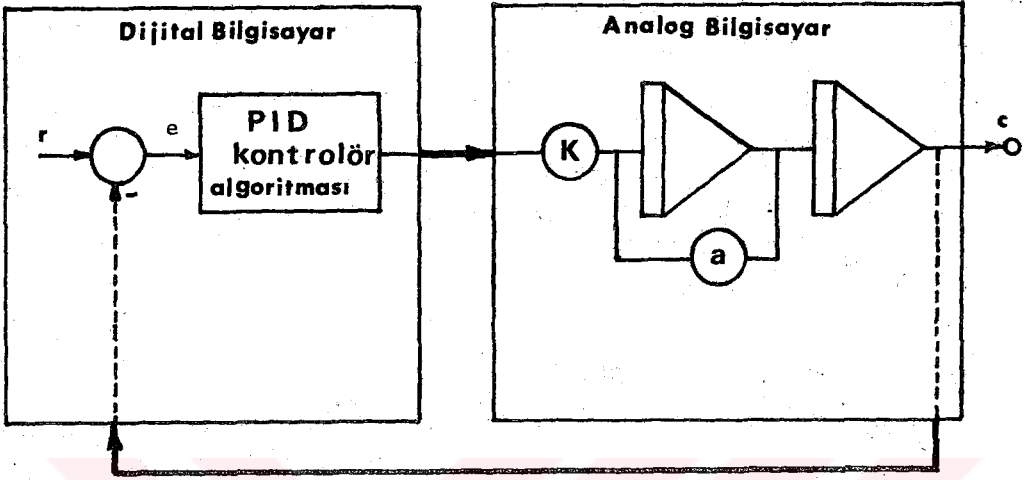
Gerçeklenen bu melez simülasyonda örnekleme zamanını ayarlamak için, dijital bilgisayar arşivinde bulunan geciktirme altprogramından yararlanılmıştır. (QSDLYR).

Kontrol sistemine ilişkin melez simülasyon modelimizi Şekil 3.7 deki biçimde verebiliriz.

### 3.3.2 P, I, D ETKİLERİ

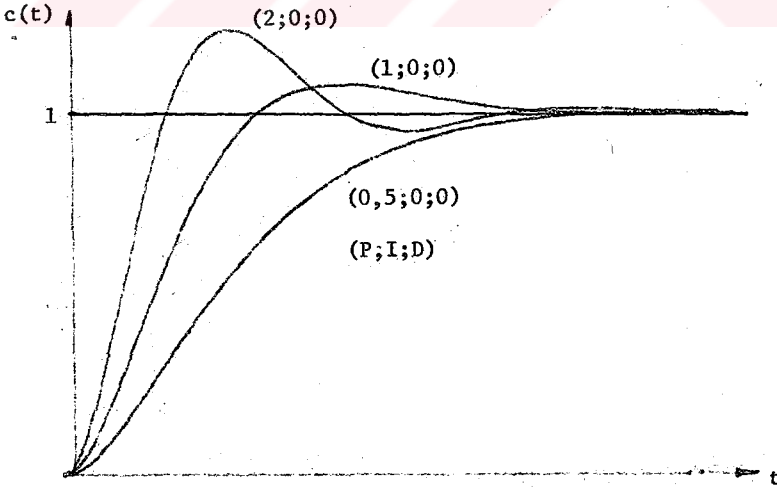
Simülasyon çalışmalarında birinci grup deneyler, Bölüm 2.3.2.4 te verdiğimiz P, I, D katsayılarının etkilerinin incelenmesi ve doğrulanması deneyleridir.

Bunun için, dizge istenen durumda iken (K ve a katsayıları sabit) sırası ile P, I ve D katsayılarına değerler verilerek çıkış işareti izlenmiş ve bir X-Y çiziciye çizdirilmiştir. Şekil 3.8 a,b ve c de sırası ile P, I ve D

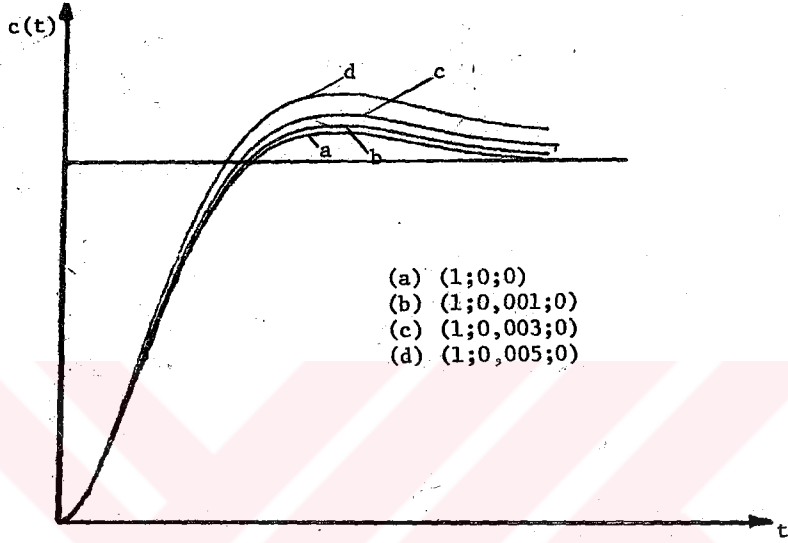


Şekil 3.7 Melez simülasyon modeli

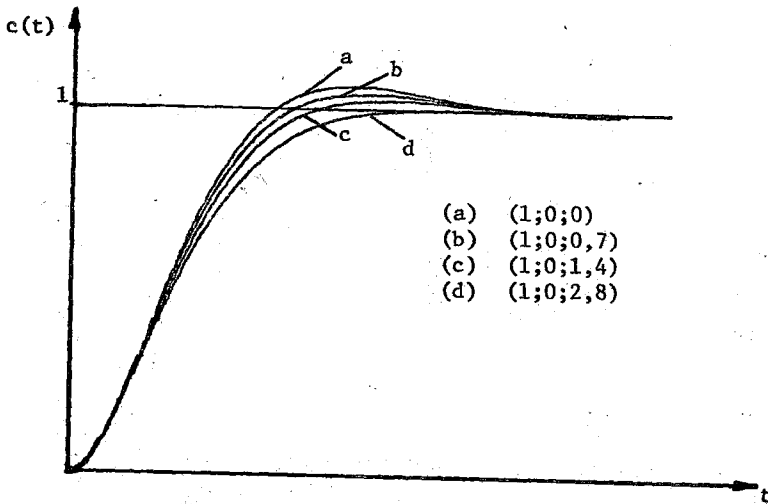
için seçilen bir kaç örnek çizim verilmiştir. Şekillerden görüleceği gibi başlangıçta verdiğimiz etki yönleri doğru olmaktadır.



Şekil 3.8a P etkisi deney sonuçları



Şekil 3.8b I etkisi deney sonuçları



Şekil 3.8c D etkisi deney sonuçları

### 3.4. ÖĞRENME AŞAMASI İÇİN BAŞLANGIÇ ARTIMLARININ SECİMİ VE DOĞRULUK DENEYLERİ

Bölüm 2.3.2.4 te açıklandığı gibi algoritma, öğrenme aşamasında (2. Adım) P, I, D katsayılarının etkilerini verecek oranları hesaplamaktadır. Bu hesapların yapılabilmesi için,  $\Delta P_0$ ,  $\Delta I_0$  ve  $\Delta D_0$  başlangıç artımlarının belirlenmesi gerekmektedir. Çalışmamızda bu değerler rasgele küçük artımlar olarak seçilmemiş, sistem davranışı üzerine minimum ancak gözlenebilir etki verecek şekilde, istenen duruma ilişkin aşım zamanı değerinden yararlanılarak hesaplanmasına çalışılmıştır. Bu bölümde  $\Delta P_0$ ,  $\Delta I_0$  ve  $\Delta D_0$  seçiminin nasıl yapılacağı ve seçime ilişkin doğruluk deneylerinin sonuçları verilecektir.

#### 3.4.1 $\Delta P_0$ 'IN SECİMİ

Açık çevrim transfer fonksiyonu,

$$G_A(s) = \frac{\omega_n^2}{s(s + 2\zeta\omega_n)} \quad (3.7)$$

olan birim geri beslemeli bir kontrol sisteminde, açık çevrim genlik eğrisinin 0 dB eksenini kesim frekansı  $\omega_k$  ile  $\omega_A$  karşılaştırıldığında  $\zeta < 0,816$  için  $\omega_k < \omega_A$  olup,  $\zeta$ 'nin küçük değerleri için bu iki değer birbirine çok yakındır (Bkz. Ek 3.3).  $P=1$  ya da  $20 \log P = 0$  dB için kesim frekansı  $\omega_k$  ve aşım zamanı  $T_A$  ise, P değerine 1 dB 'lik bir artım verildiğinde ( $P=1,122$ ), genlik eğrisinin -40 dB/dekad eğimli olduğu düşünülürse, yeni kesim frekansı  $\omega_k^1$ ,

$$\omega_k^1 = 1,059 \omega_k$$

olacaktır. Eğer  $\omega_k \approx \omega_A$  olduğu varsayılırsa, kazançtaki 1 db lik artım,

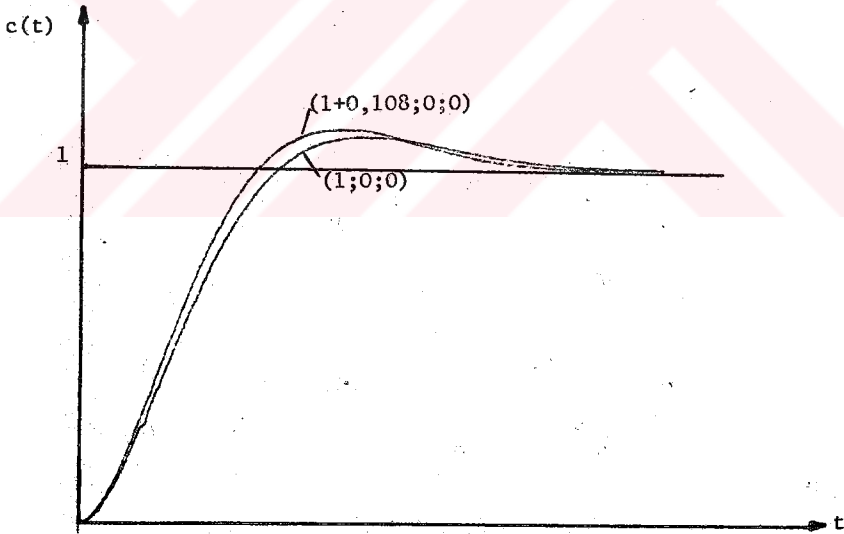
aşım zamanı üzerine,

$$T'_A = \frac{\pi}{1,059 \omega_k} = 0,944 T_A \text{ ya da } T'_A - T_A = 0,056$$

şeklinde etki edecektir. O halde  $T_A$  üzerinde %5 lik bir değişim oluşturabilmek için gerekli kazanç, yukarıdaki tersine işlem yapılırsa,

$P = 1,108$  olarak bulunur. Buna göre  $\Delta P_o = 0,108$  olarak seçilmelidir.

Şekil 3.9 da  $\Delta P_o = 0,108$  için yapılan deney sonucunda elde edilen sistem cevabına ilişkin çizim verilmiştir. Deney sonucunda yeterli etki elde edilmiş ve bu nedenle adaptasyon algoritmasında da  $\Delta P_o = 0,1$  olarak seçilmiştir.



Şekil 3.9. Öğrenme aşaması için  $\Delta P_o$  seçimine ilişkin deney sonucu

### 3.4.2 $\Delta I_0$ 'IN SECİMİ

$\Delta I_0$  'in seçilmesinde de,  $\Delta P_0$  seçimine benzer olarak,  $\Delta I_0$  'in dizgenin frekans cevabına etkisi göz önüne alınmıştır. Bilindiği gibi, integral elemanın  $P = 1$  için sisteme getireceği faz,

$$\Phi_I = -90^\circ + \text{Arctg} \frac{\omega}{T} \quad (3.8)$$

biçimindedir. Bu elemanın sisteme eklenmesi sonucu  $I$  değerine bağlı olarak tüm sistemin faz payı azalacak dolayısıyla çınılama genliği  $M_p$  artacaktır. Integral elemanına ilişkin sıfırın kırılma frekansı  $I$  olduğuna göre,  $I$ , dizgenin  $\omega_a = a$  daki kutbunun yaklaşık iki dekad altında seçildiğinde  $\omega = \omega_a$  frekansındaki faz etkisi  $-0,6^\circ$  dir. Buna göre minimum etki için,

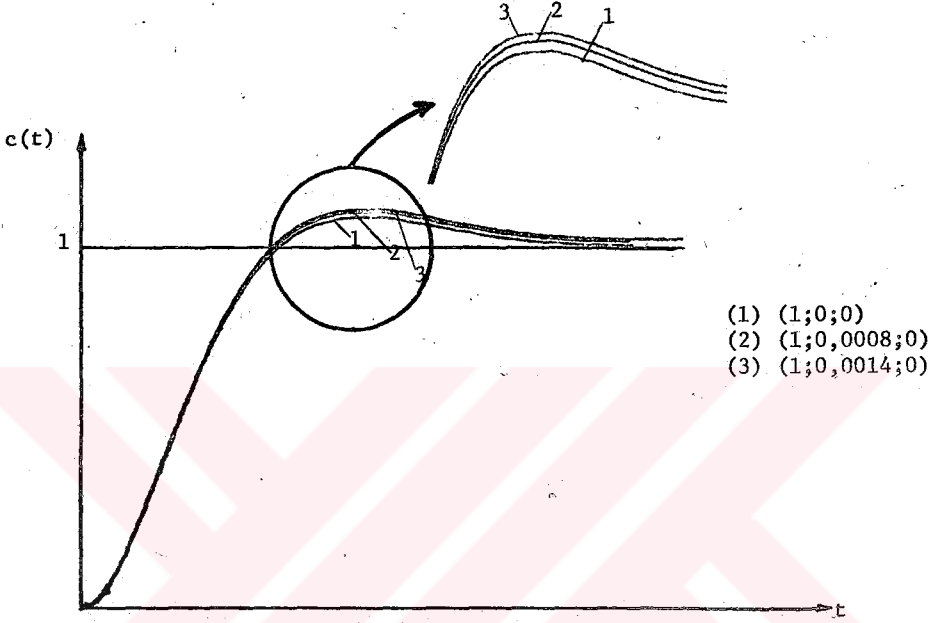
$$\Delta I_0 = 0,01 \omega_a \quad (3.9)$$

seçilmelidir. Burada da  $\omega_A$  değerinden ( $T_A$  aşım zamanından) yararlanmak istersek, (3.7) sistemi için  $\zeta > 0,42$  için  $\omega_k < 2\zeta \omega_n$  olduğundan (BKZ. Ek 3.3) (3.9) ifadesinde  $\omega_a$  yerine  $\omega_k$  ya da  $\omega_A$  kullanılması mümkündür:

$$\Delta I_0 = 0,01 \frac{\pi}{T_A} \quad (3.10)$$

olacaktır. (3.2) ile verdiğimiz dizge için,  $\omega_a = 0,14$  r/s ve  $\omega_A = \pi/T_A = \pi/39.1 = 0,08$  r/s dir. Şekil 3.10 da,  $\Delta I_0$  için her iki değerden yararlanılarak bulunan artımlar verilmiş, yapılan deney sonuçları çizdirilmiştir. Şekilden de görüldüğü gibi  $\omega_A$  dan yararlanılarak verilen artımın etkisi daha az olmaktadır. Bu ise sistem çalışmasını bozmamak yönüyle olumlu bir seçim olacağını göstermektedir.





Şekil 3.10 Öğrenme aşaması için  $\Delta I_0$  seçimine ilişkin deney sonucu

### 3.4.3 $\Delta D_0$ 'IN SECİMİ

$P = -1$  için türev elemanının fazı,

$$\phi_D = \text{Arctg } D \omega \quad (3.11)$$

biçiminde olup bu elemanın sisteme eklenmesi ile  $D$  katsayısına bağlı olarak tüm sistemin faz payı artacak dolayısıyla  $M_p$  azalacaktır. Türev elemanının kırılma frekansı  $\omega_D = 1/D$  olduğuna göre, minimum etki için, integralin tersine olarak,

$$\omega_D = 100 \omega_a \quad (3.12a)$$

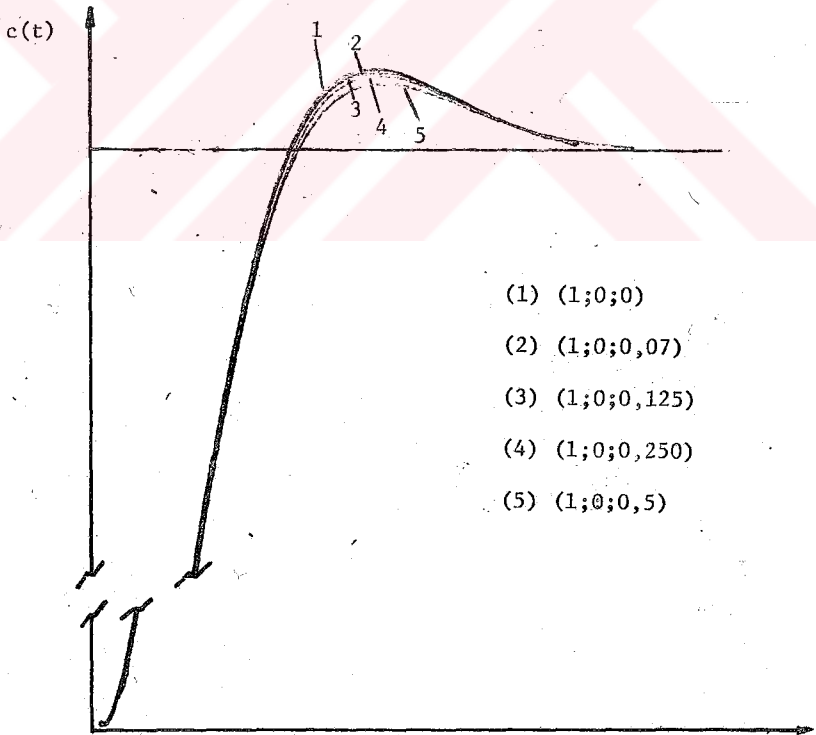
ya da

$$\Delta D_0 = 1/(100 \omega_a) \quad (3.12b)$$

olarak  $\omega_a$ 'nın iki dekad üstünde seçilmelidir. Burada da  $\omega_A$  değerini kullanırsak,

$$\Delta D_o = 1/(100 \omega_A) = 0,01 \frac{T}{\pi} \quad (3.13)$$

olacaktır.  $\Delta D_o$ 'ın seçilmesine ilişkin doğruluk deneyinde,  $\Delta D_o$ 'a sırasıyla, 0,01/0,14 ve 0,01/0,08 değerleri verilmiştir. Deney sonucunda elde edilen etkilerin çok küçük olduğu gözlenmiştir. Bu nedenle  $\Delta D_o$ 'ın 2 ve 4 katı değerleri için de deney tekrarlanmıştır. Deney sonuçlarına ilişkin farkın görülebilir olması için çıkış sinyalleri mümkün olduğu kadar büyük genlikli çizdirilmiş ve bu nedenle de Şekil 3.11 de sadece çıkışın tepe yaptığı bölge gösterilmiştir.



Şekil 3.11 Öğrenme aşaması için  $\Delta D_o$  seçimine ilişkin deney sonuçları

### 3.5 ADAPTASYON ALGORİTMASINA İLİŞKİN SİMÜLASYON CALIŞMALARI

Şimdiye kadarki bölümlerde, analog bilgisayarda dizge simülasyonunun, dijital bilgisayarda kontrolör simülasyonunun giderek bir dijital kontrollu sisteme ilişkin melez simülasyonun nasıl gerçekleştirileceğini ve 2. Bölümde verdiğimiz algoritma için gerekli ön hazırlıkları ve deneylerinin sonuçlarını vermiş bulunmaktayız.

Burada ise adaptif kontrol için tanıtılan algoritmanın bir bütün halinde işleyişi gösterilecektir. Bu nedenle öncelikle algoritmaya ilişkin olarak yazılan program tanıtılacak daha sonra örneklerle programın çalışması ve elde edilen sonuçlar verilecektir.

#### 3.5.1 PROGRAMIN TANITILMASI

Adaptif kontrol için yazılan program Ek 3.4 te verilmiştir. Programın işleyişi Bölüm 2.3.2.4 te verdiğimiz adımlara göre yapılmaktadır. Ancak burada kısaca programın akışını vermek yararlı olacaktır.

Daha önce belirttiğimiz gibi dizge analog bilgisayarda simüle edilmiştir. Başlangıçta kontrolör katsayıları  $(1,0,0)$  iken analog programda katsayılar istenen davranışı verecek biçimde ayarlanır. Bundan sonra program çalıştırılır. Algoritma gereği, sisteme birim basamak girişi (TEST) uygulanır, istenen değerler belirlenir (1.ADİM). Bundan sonraki adım öğrenme aşaması olup, sisteme test uygulanarak, kontrolör katsayılarının etkilerini verecek olan oran değerleri hesaplanır (2.ADİM). Öğrenme aşaması tamamlandıktan sonra, tekrar test uygulanır (3.ADİM), elde edilen sonuçlar (Belirleme) başlangıçta saptanan istenen değerlerle karşılaştırılır (karar verme), sistem istenen davranışta ise tekrar test uygulamaya devam edilir. Dizgede değişiklik oluşturulması, analog programda katsayılar değiştirile-

ek sağlanır. Bu ise sistem istenen davranışa geldikten sonra (adaptasyon tamamlandıktan sonra) katsayı potansiyometreleri değiştirilerek yapılacaktır. Eğer sistem çıkışında hissedilebilir bir değişiklik yapılmış ise, algoritma '4.ADIM' a geçerek adaptasyon işlemlerine başlayacaktır. İstenen davranış elde edilene kadar 3. ve 4. adımlar tekrarlanacaktır. Adaptasyon tamamlandıktan sonra da program gereği test uygulanmasına devam edilecektir. Eğer bu arada dizgede değişiklik yapılmışsa, algoritma yeniden adaptasyona başlayacak ve işlemler benzer biçimde sürüp gidecektir.

Bu işlem akışına göre çalışan programda görülen değişkenlerin ve deyimlerin bir kısmına ilişkin açıklamalar Ek 3.6 da verilmiştir. Burada sadece programda görülen TEST alt programının tanıtılmasında yarar vardır.

### 3.5.1.1 TEST ALT PROGRAMI

Bu alt program, daha önce anlatılan deneylerimizde kullanılan dijital PID kontrolör (Ek 3.2) programından yararlanılarak düzenlenmiştir.

SUBROUTINE TEST (AP, AI, AD, GCKM, CD, TD) alt programının giriş/çıkış değişkenleri (argüman listesinde verilen) aşağıdaki gibi açıklanabilir:

Giriş Değişkenleri :

- AP : Kontrolörün P katsayısı
- AI : Kontrolörün I katsayısı
- AD : Kontrolörün D katsayısı
- GCKM : Örnekleme zamanı (Bkz. Bölüm III.3.1)

Çıkış Değişkenleri :

- CD : Çıkış işaretinin maksimum değeri
- TD : Çıkışın maksimum değerine ulaştığı örnek sayısı (aşım erişme zamanı)

Programda çıkış değişkenleri olan CD ve TD daha önce Bölüm 2.3.2.2 de belirttiğimiz biçimde saptanmaktadır. Alt programda CD ve TD belirlendikten sonra ana programa dönülür. Alt program Ek 3.5 te verilmiştir.

### 3.5.2 SİMÜLASYON ÖRNEĞİ (I)

Gerçeklenen programın işleyişini göstermek amacıyla ,ilk örneğimiz daha önce seçtiğimiz (2.32) ile verilen dizgeye ilişkindir. Burada K kazanıcı sabit tutulacak ve 'a'parametresi değiştirilerek algoritmanın adaptasyonu gerçekleştirilmesi izlenecektir. 'a'parametresinin değişiminin kontrolörsüz halde sistem cevabına etkisini göstermek üzere, Şekil 3.3 teki analog program kullanılarak, kontrolörsüz sürekli halde, 'a'katsayısına '0' dan '0,24' e kadar çeşitli değerler verildiğinde elde edilen çıkışlara ilişkin çizimler Şekil 3.12 de verilmiştir.

'a' katsayısına (analog programda (01) potansiyometresi) istenen durum için '0,14' değeri verilmiş[(3.2)transfer fonksiyonu], program işletilerek, daha sonra katsayı değiştirilerek her seferinde adaptasyon izlenmiş, elde edilen sonuçlar ve çizimler aşağıda açıklamalı olarak adım adım verilmiştir.

#### 1° Giriş ve Öğrenme Aşaması

$$a = 0,14$$

$$K = 0,1095 \times 0,1019 = 0,11 \quad \text{için yapılan ilk test sonucunda, program}$$

çıkışında,

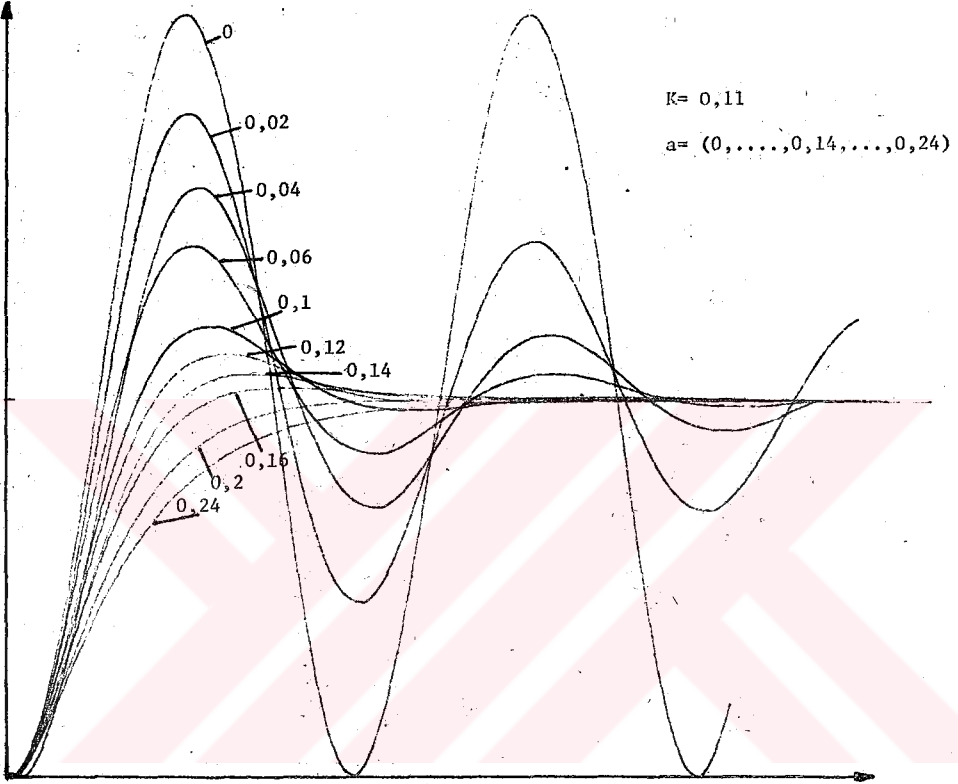
```
*****
REF= .1000 TOLC= .0020 TOLT= .0000 TS=
1000.0
```

İSTENEN DEGERLER:

ÇIKIŞ= .108099

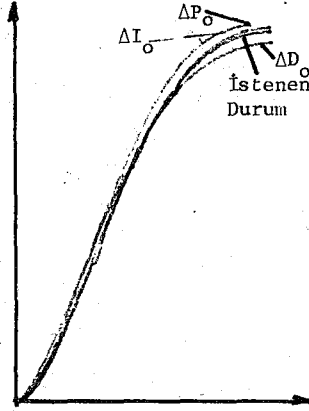
TEPE ZAMANI= 34.

```
*****
```



Şekil 3.12 'a' parametresinin değişimine karşılık  
sistem çıkışı değişimi

yazılmaktadır. Burada ÇIKIS ile istenen maksimum çıkış değeri, TEPE ZAMANI ile de bu maksimum değeri aldığı örnek sayısı gösterilmiştir. TOLC mutlak aşım toleransını, TOLCT de mutlak aşım zamanı toleransını göstermektedir. Program çıkışı olarak bu değerler yazıldıktan sonra öğrenme aşamasına geçilecektir. İstenen durum ve öğrenme aşamasına ilişkin çizimler Şekil 3.13 te verilmiştir.



Şekil 3.13 İstenen durum ve öğrenmeye ilişkin çizimler

Bundan sonra tekrar test uygulamasına geçilmiş ve istenen değerlerde tolerans dışı bir değişme olmadığından çıkışta,

P= 1.00000  
I= .000000  
D= .000000  
C= .108199  
T= 34.

biçiminde P,I,D katsayıları, çıkışın maksimum değeri C, ve aşım zamanı T yazılmıştır. Bu formatta bir çıkış görüldüğünde, sistemin adapte edildiği ya da henüz sistem çıkışında istenen değerlerden toleranslar dışında sapmanın olmadığı anlaşılacaktır.

## 2° A d a p t a s y o n A ş a m a s ı

Adaptasyon aşamasına geçişi sağlamak için, analog programda 'a' katsayısına '0.12' değeri verilmiş, bu durumda test uygulandıktan sonra, program çıkışında,

ADAPTASYON BASLANGICI:  
(P,I,D): 1.00000 .000000 .000000 C= .113999 T= 31.

yazılmıştır. Bu format da, sistem çıkışının istenen değerden saptığını yani bir parametre değişikliğinin olduğunu ve adaptasyona başlanacağını göstermektedir. Elde edilen sonuçlardan ve öğrenmeden edinilen bilgilerden yararlanarak kontrolör katsayıları değiştirilecek ve test uygulanacaktır. Bu işlemler de, her katsayı yenileme ve elde edilen sonuçlar adaptasyon tamamlanana kadar,

#### ÖZETLENEN İŞLEMLERİ:

P(I,D):	.850000	.000000	.905069	C= .107399	T= 37.
P(I,D):	.999999	.000000	1.65565	C= .107799	T= 32.

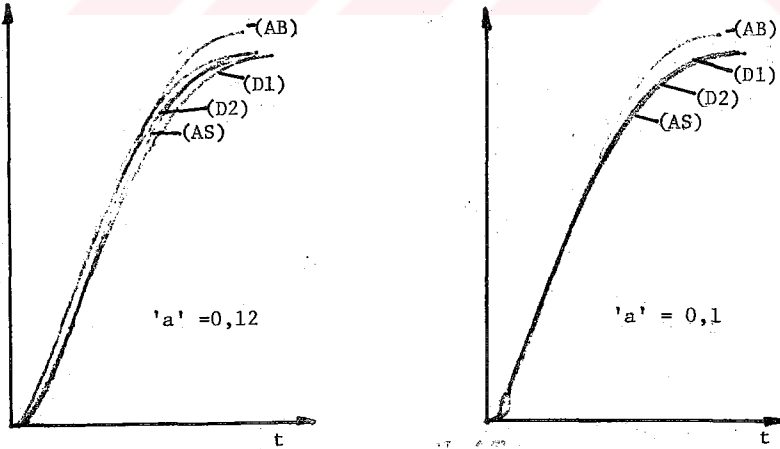
formatında yazılacaktır. Adaptasyon tamamlandığında ise, çıkışta,

#### ADAPTASYON SONU:

P= .939999  
I= .000000  
D= 1.65565

C= .106699      T= 34.

yazılacaktır. Buradan da adaptasyonun tamamlandığı anlaşılmakta ve o andaki kontrolör katsayılarına verilen değerlerle bunlara karşılık elde edilen sonuçlar çıkışta yazılmaktadır. Şekil 3.14a da bu aşamaya ilişkin çizimler



Şekil 3.14 a) 'a' = 0,12 için adaptasyon aşaması

b) 'a' = 0,1 " " "



verilmiştir. Şekillerde görülen kısaltmaların açıklaması ise şöyledir,

- (AB) : adaptasyon başlangıcını,  
 (D1) : 1. Değiştirme sonucunu,  
 (D2) : 2. " " ,  
 .....  
 (Dn) : n. " " ,  
 (AS) : Adaptasyon sonunu göstermektedir.

Bundan sonraki aşamalarda, 'a' katsayısı değiştirilerek sonuçlar alınmış ve çizimler elde edilmiştir. Aşağıda bu deneylerden örnekler verilmiştir:

A) 'a' katsayısı '0,12' den '0,1' e değiştirildiğinde (Şekil 3.14b)

<b>ADAPTASYON BASLANGICI:</b>				
(P,I,D): .939999	.000000	1.65565	C= .112399	T= 31.
<b>DEĞİŞTİRME İSLEMLERİ:</b>				
1.				
(P,I,D): .850000	.000000	2.44202	C= .107699	T= 33.
2.				
(P,I,D): .820000	.000000	2.44202	C= .107099	T= 35.
<b>ADAPTASYON SONU:</b>				
P= .834999				
I= .000000	C= .107399		T= 34.	
D= 2.44202				

B) '0,1' den '0,08' e (Şekil 3.15 a)

<b>ADAPTASYON BASLANGICI:</b>				
(P,I,D): .834999	.000000	2.44202	C= .113899	T= 31.
<b>DEĞİŞTİRME İSLEMLERİ:</b>				
1.				
(P,I,D): .789999	.000000	3.79344	C= .107899	T= 32.
2.				
(P,I,D): .759999	.000000	3.79344	C= .107199	T= 33.
<b>ADAPTASYON SONU:</b>				
P= .730000				
I= .000000	C= .106500		T= 34.	
D= 3.79344				

C) '0,08' den '0,06' ya (Şekil 3.15 b)

ADAPTASYON BASLANGICI:  
(P,I,D): .730000 .000000 3.79344 C= .113699 T= 31.

DEĞİŞTİRME İŞLEMLERİ:  
1.  
(P,I,D): .640000 .000000 4.57545 C= .108299 T= 33.

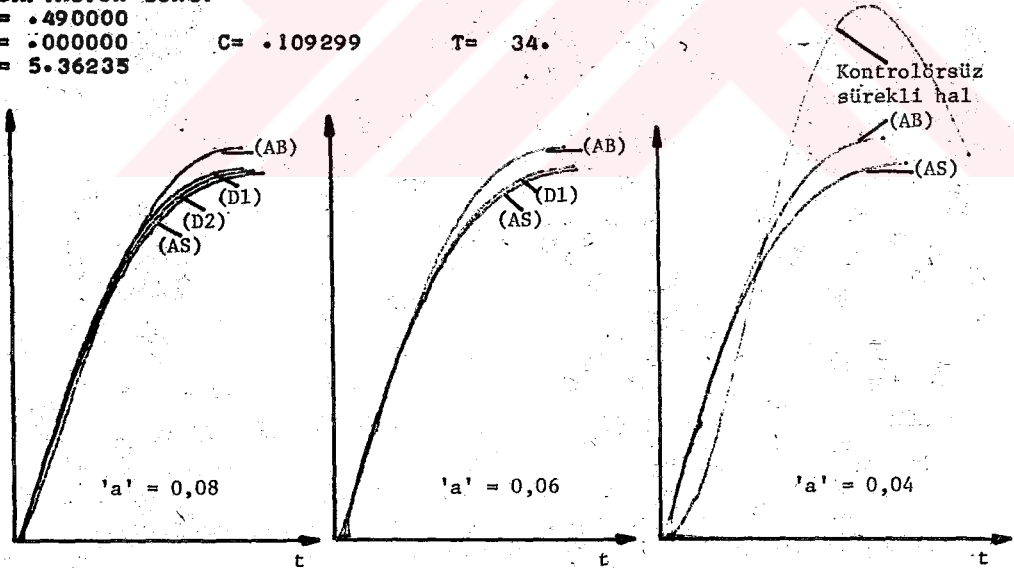
ADAPTASYON SONU:  
P= .610000  
I= .000000 C= .107399 T= 34.  
D= 4.57545

D) '0,06' dan '0,04' e (Şekil 3.15 c)

(Bu şekilde ayrıca, karşılaştırma için, kontrolörsüz sürekli çıkışa ilişkin çizim de verilmiştir)

ADAPTASYON BASLANGICI:  
(P,I,D): .610000 .000000 4.57545 C= .116299 T= 30.

ADAPTASYON SONU:  
P= .490000  
I= .000000 C= .109299 T= 34.  
D= 5.36235



(a)

(b)

(c)

Şekil 3.15 a) 'a' = 0,08 için adaptasyon aşaması

b) 'a' = 0,06 " " "

c) 'a' = 0,04 " " "

Benzer biçimde 'a' katsayısı değiştirilerek deney tekrarlanmıştır. Katsayı değeri tekrar artırılarak deneylere devam edilmiştir ve her seferinde adaptasyonun sağlandığı görülmüştür. Katsayının artırılması yönündeki deneylere ilişkin birkaç örnek aşağıda verilmiştir.

E) '0,12' den '0,14' e ( Şekil 3.16 a)

ADAPTASYON BASLANGICI:  
(P,I,D): .926302 .000000 1.44331 C= .103099 T= 38

DEĞİSTİRME İSLEMLERİ:  
1.  
(P,I,D): .995438 .000000 .476943 C= .106599 T= 34

ADAPTASYON SONU:  
P= 1.01272  
I= .000000 C= .106899 T= 34.  
D= .476943

F) '0,14' ten '0,16' ya (Şekil 3.16' b)

ADAPTASYON BASLANGICI:  
(P,I,D): 1.01272 .000000 .476943 C= .103199 T= 34

DEĞİSTİRME İSLEMLERİ:  
1.  
(P,I,D): 1.09914 .000000 .000000 C= .105399 T= 34

ADAPTASYON SONU:  
P= 1.11642  
I= .206317E-02 C= .107799 T= 34.  
D= .000000

C) '0,16' dan '0,18' e (Şekil 3.16 c)

ADAPTASYON BASLANGICI:  
(P,I,D): 1.11642 .206317E-02 .000000 C= .104799 T= 34

DEĞİSTİRME İSLEMLERİ:  
1.  
(P,I,D): 1.18556 .000000 .000000 C= .103199 T= 34

2.  
(P,I,D): 1.20284 .231551E-02 .000000 C= .105699 T= 34

3.  
(P,I,D): 1.22012 .231551E-02 .000000 C= .105899 T= 34

4.  
(P,I,D): 1.25469      .231551E-02      .000000      C= .106099      T= 34.

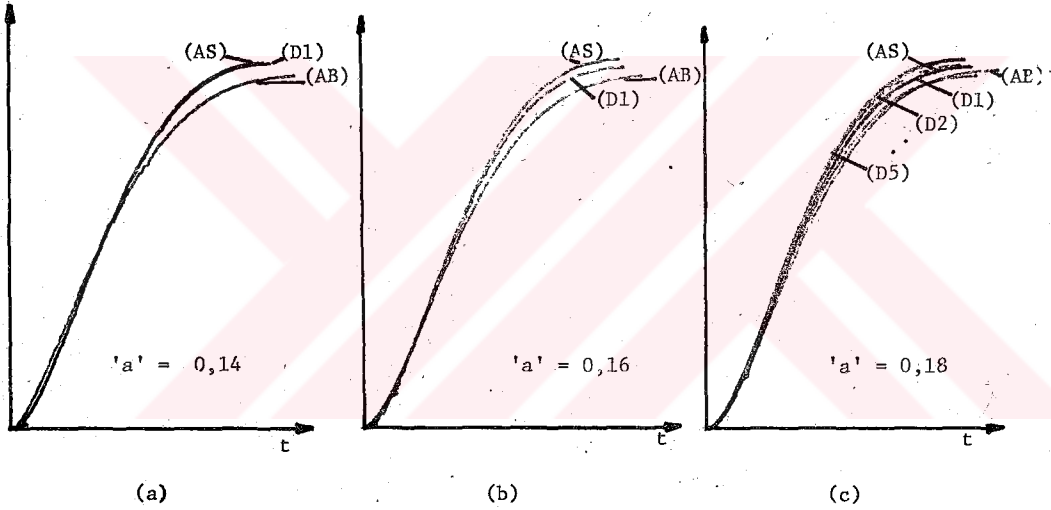
5.  
(P,I,D): 1.25469      .424776E-02      .000000      C= .107899      T= 35.

## ADAPTASYON SONU:

P= 1.27197

I= .424776E-02      C= .107899      T= 34.

D= .000000

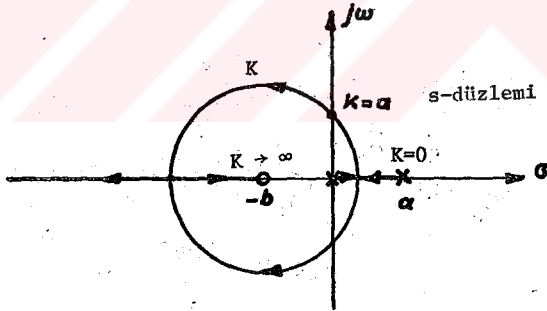
Şekil 3.16 a)  $'a' = 0,14$  için adaptasyon aşamasıb)  $'a' = 0,16$  " " "c)  $'a' = 0,18$  " " "

### 3.5.3 SİMÜLASYON ÖRNEĞİ (II)

İlk örnek üzerine yapılan deneylerden görüldüğü gibi algoritmamız adaptasyonu sağlamaktadır. Burada vereceğimiz ikinci örnek, Bölüm 2.3.2.4 te bahsettiğimiz gibi umulanın tersine davranış gösteren bir dizgeye ilişkindir. Bu tür bir dizge için seçilen transfer fonksiyonu,

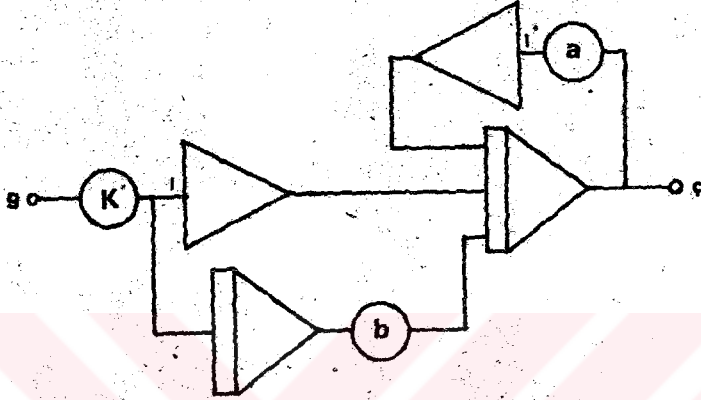
$$G_{II}(s) = \frac{K(s+b)}{s(s-a)} \quad (3.14)$$

biçiminde olup, bu transfer fonksiyonu için, sürekli halde geometrik yer eğrisi Şekil 3.17'deki gibidir. Bu sistemde çıkışın kararlı olduğu bölgede ( $K > a$ ), geometrik yer eğrisinden görüleceği gibi kazanç artışı aşımı azaltacak yönde etki edecektir.

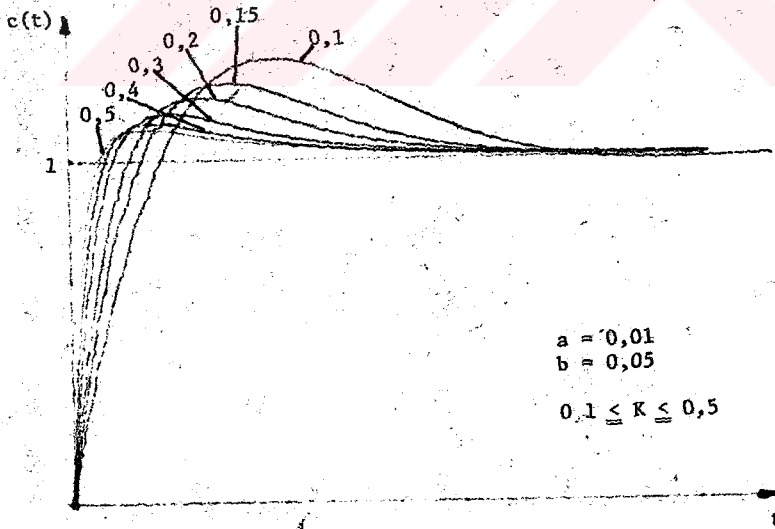


Şekil 3.17 (3.14) transfer fonksiyonuna ilişkin geometrik yer eğrisi

Bu dizgeye ilişkin analog program ise, Şekil 3.18 de verilmiştir. Şekil 3.19 da ise, ( $a = 0,01$  ;  $b = 0,05$  ;  $K = 0,1 \dots\dots 0,5$  değişken ) için kontrolör-süz sürekli hale ilişkin, sistemin birim basamak cevabı çizdirilmiştir. Şekilden görüldüğü gibi, kazanç artışı aşımı ve aşım zamanını azaltacak yönde etmektedir.



Şekil 3.18  $G_{II}(s)$  ' e ilişkin analog program



Şekil 3.19  $G_{II}(s)$  'in kazanç değişimine göre kontrolörsüz sürekli halde birim basamak girişine cevabı

Aşağıda bu dizge için yapılan simülasyon çalışmalarının bazılarının sonuçları verilmiştir.

**DENEY 1 : K KAZANCININ DEĞİŞMESİNE KARŞILIK ADAPTASYON**

İstenen durum için seçilen katsayı değerleri,

$$K = 0,2$$

$$a = 0,01$$

$$b = 0,05$$

olup, bu durumda program çalıştırıldığında aşağıdaki sonuçlar elde edilmiştir:

1° Giriş ve Öğrenme Aşaması :

\*\*\*\*\*  
REF= .1000 TOLC= .0500 TOLT= 1.0000 TS= 1000.0

ISTENEN DEGERLER:  
CIKIS= .120199 TEPE ZAMANI= 16.

\*\*\*\*\*  
P= 1.00000  
I= .000000 C= .120199 T= 16.  
D= .000000

2° Adaptasyon Aşaması :

A) 'K' katsayısı '0,2' den '0,25' e değiştirildiğinde

ADAPTASYON BASLANGICI:  
(P,I,D): 1.00000 .000000 .000000 C= .117399 T= 14.

ADAPTASYON SONU:  
P= .800000  
I= .000000 C= .120399 T= 16.  
D= .000000

B) '0,3' ten '0,4' e

ADAPTASYON BASLANGICI:  
(P,I,D): .699999 .000000 .000000 C= .116299 T= 12.

DEGISTIRME ISLEMLERI:

1.  
(P,I,D): .300000 .000000 .000000 C= .129999 T= 23.

2.  
(P,I,D): .999999 .000000 .000000 C= .112599 T= 9.

3.  
(P,I,D): .649999 .000000 .000000 C= .117299 T= 13.

4.  
(P,I,D): .387499 .000000 .000000 C= .125099 T= 19.

ADAPTASYON SONU:

P= .518749  
I= .000000 C= .120299 T= 16.  
D= .000000

C) '0,5' ten '0,6' ya

ADAPTASYON BASLANGICI:  
(P,I,D): .387499 .000000 .000000 C= .119499 T= 14.

ADAPTASYON SONU:

P= .299999  
I= .000000 C= .123499 T= 17.  
D= .000000

D) '0,4' ten '0,3' e

ADAPTASYON BASLANGICI:  
(P,I,D): .562499 .000000 .000000 C= .123199 T= 18.

ADAPTASYON SONU:

P= .649999  
I= .000000 C= .120899 T= 16.  
D= .000000



E) '0,2' den '0,15' e (Şekil 3.20 a)

ADAPTASYON BASLANGICI:  
(P,I,D): .912499 .000000 .000000 C= .126299 T= 21.

DEĞİSTİRME İSLEMLERİ:  
1.  
(P,I,D): 1.13124 .000000 .000000 C= .122699 T= 18.

ADAPTASYON SONU:  
P= 1.21874  
I= .000000 C= .121599 T= 17.  
D= .000000

F) '0,15' ten '0,25' e (Şekil 3.20 b)

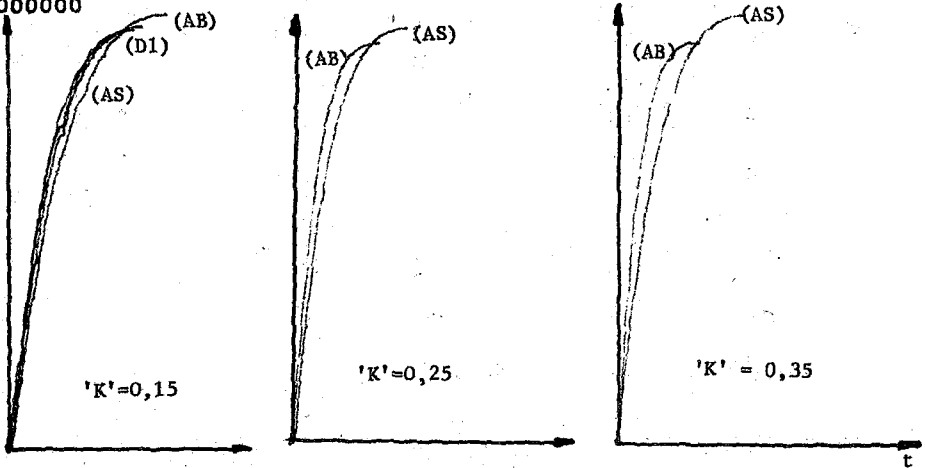
ADAPTASYON BASLANGICI:  
(P,I,D): 1.21874 .000000 .000000 C= .115599 T= 12.

ADAPTASYON SONU:  
P= .868749  
I= .000000 C= .120099 T= 15.  
D= .000000

G) '0,25' ten '0,35' e (Şekil 3.20 c)

ADAPTASYON BASLANGICI:  
(P,I,D): .868749 .000000 .000000 C= .116399 T= 12.

ADAPTASYON SONU:  
P= .518749  
I= .000000 C= .124199 T= 17.  
D= .000000



Şekil 3.20 a) 'K' = 0,15 için adaptasyon aşaması  
b) 'K' = 0,25 " " "  
c) 'K' = 0,35 " " "

## DENEY 2: b KATSAYISININ DEĞİŞMESİNE KARŞILIK ADAPTASYON

İstenen durum için katsayılar Deney 1 deki gibi seçilmiştir.

## 1° Giriş ve Öğrenme Aşaması :

\*\*\*\*\*  
 REF= .1000 TOLC= .0050 TOLT= 1.0000 TS= 1000.0

## İSTENEN DEĞERLER:

CIKIS= .121999 TEFE ZAMANI= 16.

\*\*\*\*\*

P= 1.00000 C= .121699 T= 16.  
 I= .000000  
 D= .000000

## 2° Adaptasyon Aşaması :

A) 'b' katsayısı '0,05' ten '0,035' e değiştirildiğinde

ADAPTASYON BASLANGICI:  
 (P,I,D): 1.00000 .000000 .000000 C= .117299 T= 18.

## DEĞİŞTİRME İSLEMLERİ:

1.  
 (P,I,D): 1.19999 .465359E-01 .000000 C= .123199 T= 13.

## ADAPTASYON SONU:

P= .899999  
 I= .200281E-01 C= .125000 T= 17.  
 D= .000000

B) '0,06' dan '0,07' ye

ADAPTASYON BASLANGICI:  
 (P,I,D): 1.02857 .000000 .000000 C= .124399 T= 14.

## DEĞİŞTİRME İSLEMLERİ:

1.  
 (P,I,D): .774601 .000000 1.07968 C= .123299 T= 19.

## ADAPTASYON SONU:

P= 1.15555  
 I= .000000 C= .118499 T= 15.  
 D= 1.07968

C) '0,08' den '0,11' e

ADAPTASYON BASLANGICI:

(P,I,D): .965077 .000000 1.07968 C= .124100 T= 14.

ADAPTASYON SONU:

P= .774601

I= .000000 C= .126799 T= 16.

D= 1.07968

D) '0,11' den '0,09' a

ADAPTASYON BASLANGICI:

(P,I,D): .774601 .000000 1.07968 C= .125099 T= 18.

ADAPTASYON SONU:

P= .965077

I= .000000 C= .122399 T= 16.

D= 1.07968

E) '0,09' dan '0,06' ya

ADAPTASYON BASLANGICI:

(P,I,D): .965077 .000000 1.07968 C= .119199 T= 18.

DEGISTIRME ISLEMLERİ:

1.

(P,I,D): 1.15555 .000000 .000000 C= .120599 T= 14.

ADAPTASYON SONU:

P= .965077

I= .000000 C= .123299 T= 16.

D= .000000



**4 . bölüm**

**GERÇEKLEME**

Bu bölümde, çalışmasını Bölüm 3 de göstermiş olduğumuz adaptasyon algoritmamızın, bir mikrobilgisayar sistemi ile nasıl gerçekleştirileceği ve bu gerçekleştirilmede karşılaşılan teknik sorunların çözümleri ele alınacaktır.

Bu nedenle öncelikle çalışmada kullanılan mikrobilgisayarın ve bilgi aktarma aygıtının teknik özellikleri kısaca verilecek, uygulamada seçilen bilgi aktarma protokolu tanıtılacaktır.

Yüksek seviyede dil kullanılmayan bir bilgisayarda dijital kontrolör yazılımının oluşturulmasında bazı ayrıcalıklar ve teknik sorunlar ortaya çıkmaktadır. Burada özellikle bu sorunları ortaya çıkarabilmek ve çözümlerini getirmek amaçlanmıştır. Örneğin, simülasyon çalışmasında düşünmediğimiz bir sorun, bilgisayar aritmetiğinin oluşturulması, gerçeklemeye ilişkin olan bu çalışmada temel sorunlardan biri olarak ortaya çıkmaktadır. Bu nedenle, bilgisayar aritmetiği, sorunları ve çözümleri bu bölümün konusudur.

Gerçeklemeye ayırdığımız bu bölümde, kullanılan donanım, adaptasyon algoritmasına ilişkin yazılım tanıtılacak, yapılan deneyler ve sonuçları verilecektir.

#### 4.1. MİKROBİLGİSAYARIN VE BİLGİ AKTARMA AYGITININ SEÇİMİ

Bir dijital kontrol sisteminde bilindiği gibi, sürekli bir değerin dijital biçime dönüştürülmesi ve tersine dijital bir bilginin de analog bir değere dönüştürülmesi söz konusudur (Bilgi Aktarma).

Sürekli bir değerin dijital biçime getirilmesinde bazı yanlışlıkların olacağı açıktır (23). Bu yanlışlıklardan önemli olan bir tanesi ise kuantalama yanlışlığıdır. Örnek olarak  $\pm 10$  Voltluk aralıkta değişen bir sürekli işaret,

a) 8 bitlik 2'ye t mleyen biimde alıřan bir AD ye uygulandıėında, elde edilecek veri deėerinde , kuantalama yanılıėı  $\epsilon_k$

$$\epsilon_k = 2^{-7} \text{ olup,}$$

olup,  $\pm 10$  volt iin belirsizlik,

$\pm 10 \cdot \epsilon_k = \pm 78$  mV olacaktır. AD giriřine uygulanan s rekli iřare -  
tin ortalama deėeri 5 V ise bu durumda yanılıėa,

$$\pm 78 \cdot 10^{-3} / 5 = \pm 1,56$$

olacaktır.

b) 12 bitlik bir AD kullanıldıėında ise kuantalama yanılıėı

$$\epsilon_k = 2^{-11}$$

ve benzer biimde  $\pm 10$  voltteki belirsizlik,

$$\pm 10 \cdot 2^{-11} = \pm 4,88 \text{ mV}$$

5 voltluk ortalama deėerdeki yanılıėa da,

$$\pm 4,88 \cdot 10^{-3} / 5 = \pm 0,096$$

olacaktır. Bu iki  rnekten g r ld ėu gibi bilgi aktarma aygıtının bit sayı-  
sının artması, doėruluėu artırmaktadır. Ancak bit sayısının doėruluėu ar-  
tırmasına karřılık g z  n ne alınması gereken řu noktalar vardır:

1<sup>o</sup> Bilgi aktarma aygıtının fiyatı bit sayısına baėlı olarak  stel  
bir biimde artmaktadır.  rneėin 12 bitlik bir AD, 8 bitlik bir AD nin  
15-20 katı fiyatında, 16 bitlik bir AD ise 8 bitliėin 100 katı fiyatın-  
dadır.

2<sup>o</sup> Bit sayısının artması bu veriyi iřleyecek bilgisayarın seėimi-  
ni de etkileyecektir. Bu da maliyet aısından  nemlidir. 16 bitlik bir bil-  
gisayar 8 bitlik bir bilgisayara g re 20-50 kat pahalıdır.

Ancak bilgisayarla kontrol gerçeklenmesinde endüstri standartları göz önüne alınırça, bilgi aktarma aygıtının en az 12 bitlik olması gerekmektedir.\*

Bu durum da dikkate alınarak 12 bitlik bilgi aktarma aygıtı seçilmiştir. Mikrobilgisayar ise 8 bitlik olarak seçilmiş ancak 12 bitlik olarak gelen bilgiyi işlemek üzere, 8 bitlik bilgisayar 16 ya da 32 bitlik veri işler biçimde yazılım olarak düzenlenmiştir. 8 bitlik bir makinada 16 ya da 32 bitlik işlem yapılması ise uzun ve karmaşık programlamayı gerektirmiştir. Bu ise sadece programcıyı zorlar görünmekte ise de programın işleme süresinin dijital kontrol için katlanabilir düzeyde olması gerekmektedir.

Bu durum da göz önüne alınmış ve sonuç olarak aşağıda teknik özellikleri kısaca verilen 8 bitlik bilgisayar ve 12 bitlik bilgi aktarma aygıtı çalışmalarımız için seçilmiştir.

## 4.2 MİKROBİLGİSAYAR VE BİLGİ AKTARMA AYGITI

### 4.2.1 MİKROBİLGİSAYAR (24)

Motorola MEK-6802 D5E

Merkezi İşlem Birimi (MİB) : Motorola M 6800

Sözcük Uzunluğu : 8 bit

Bellek Kapasitesi : 3K-byte

Saat Frekansı : 890 KHz

Programlama : Makina dilinde

\* Foxboro SPEC 200 , Honeywell TDC-2000 ve Fischer & Porter DCI-4000 sistemleri örnek olarak verilebilir.

## 4.2.2 BİLGİ AKTARMA AYGITI (25)

### A- ANALOG-DİJİTAL ÇEVİRİCİ (ADÇ) VE ÖRNEKLEME-TUTMA (Ö/T) DEVRESİ

Burr-Brown MP 7802 - 7216 , Motorola UNIBUS' ına uyumlu

Giriş Gerilimi :  $\pm 10$  V (aralığı seçilmiştir.)

Ayırıcılık : 12 bit, ikili  
En düşük anlamlı bit (EDB) karşılığı: 4,8828 mV

Çevirme Süresi: 33  $\mu$ s (  $\pm 10$  V kademesinde)

Dijital Çıkış: 2'ye tümleyen biçimde

Analog giriş Sayısı: 8 kanal simetrik- 16 kanal asimetrik

### B- DİJİTAL-ANALOG ÇEVİRİCİ (DAÇ) DEVRESİ

Burr-Brown MP 7104 , Motorola UNIBUS' ına uyumlu

Giriş Verisi: 12 bit ikili, 2'ye tümleyen

Ayırıcılık: 12 bit, ikili  
EDB karşılığı : 4,8828 mV

Analog Çıkış :  $\pm 10$  V (aralığı seçilmiştir)

Analog çıkış sayısı: 4 kanal simetrik ya da asimetrik seçilebilir

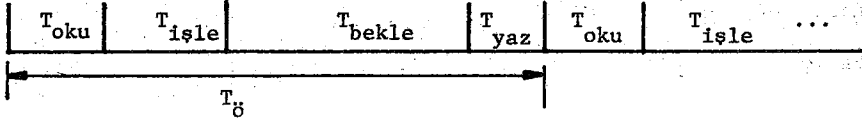
## 4.2.3 BİLGİ AKTARMA PROTOKOLU

Analog bir değer'in mikrobilgisayara aktarılması için i s t e m e l i T ü r 'den protokol kullanılmıştır (22). Bu protokol uyarınca, sürekli bir değişimin mikrobilgisayara aktarılması aşamaları şöyledir :

1° Örnekleme zamanının geldiğinin anlaşılması :

Uygulamamızda, gerçek zaman saatinin kullanılmasına gerek duyulmamıştır. Bunun yerine örnek alma süresi yazılım olarak gerçekleştirilmiştir. Örnek alma periyodu , iki örnek alma arasında bilgisayarda yapılan işlemler cinsinden şu şekilde belirlenmektedir :





Burada,

$T_{oku}$  : ADÇ 'nin çevirme süresi Verinin MİB' ne alınması için geçen süre olup  $\approx 40 \mu s$  (sabit) tir.

$T_{işle}$  : Alınan ya da okunan örnek değere bilgisayarda uygulanan aritmetik işlemlerin alacağı süre olup  $\approx 1 ms$  (sabit) dir.

$T_{bekle}$ : Örnekleme süresini denk getirebilmek için özellikle eklenmiş program parçasının (bir geciktirme çevrimi) alacağı süre olup, deneylerde  $\approx 7 ms$  olarak ayarlanmıştır.

$T_{yaz}$  : MİB' indeki verinin DAÇ çıkışına ulaşması için geçen süre olup  $< 10 \mu s$  (sabit) dir.

Görüldüğü gibi  $T_{bekle}$  süresi ayarlanarak, örnekleme zamanı değiştirilebilmekte, örnekleme zamanının geldiği ise yazılım olarak anlaşılmaktadır.

2° Örnek Alma :

Bu aşamada MİB, Ö/T devresine elindeki örneği TUT ve aynı zamanda ADÇ 'ye Ö/T çıkışındaki değeri ÇEVİR komutunu gönderir. Böylece analog işaretin dijital işarete çevrilmesi işlemi başlatılır.

3° Analog-Dijital Çevirme :

MİB 'nin isteği üzerine analog-dijital çevirme işlemi yukarı-

daki gibi başlar ve devam eder. Çevirme süresi giriş geriliminden bağımsız olduğu için, MİB bu sabit çevirme süresince bekleme durumuna sokulur.

4° Okuma :

Bu aşamada MİB, ADÇ çıkışındaki veriyi kendi içine alır.

Bilgisayarda bulunan bir verinin dış ortama aktarılması ise bu verinin doğrudan doğruya DAÇ girişine yazılmasıyla gerçekleşmektedir. DAÇ her zaman girişindeki veriyi analog büyüklüğe çevirmeye hazırdır ve çevirme süresi sadece propagasyon gecikmesi kadar olacaktır.

### 4.3 KONTROLÖR YAZILIMI

Simülasyon çalışmamıza benzer olarak (Bölüm 3.3.1 ) burada da (2.25) ilişkisi ile verilen kontrolör denklemi gerçekleştirilecektir. Şekil 3.6 da verdiğimiz akış diyagramı tekrar göz önüne alınırsa, burada da benzer bir akış söz konusudur. Kontrolöre ilişkin yazılımda gerekli aritmetik işlemler (toplama, çıkarma, çarpma ve bölme) burada oldukça büyük önem kazanmaktadır.

Yüksek seviyede dil kullanmayan bir bilgisayarda bu temel işlemlerin gerçekleşmesi hem de 8 bitlik bir makinada 16 ya da 32 bitlik veriler için gerçekleşmesi önemli bir sorundur. Uygulamamızda seçtiğimiz bilgisayarda yüksek seviyede dil kullanılmamış olması,

1° işleme süresini kısaltmak,

2° Bilgisayar aritmetiğini oluşturmak, getireceği sorunları ortaya çıkarmak ve çözümlerini araştırmak,

amacıyla tercih edilmiştir.

Bir başka deyişle, böyle bir mikrobilgisayar sistemi ile bir kontrol çevrimi kurmak için gerekli yazılım, bu yazılımda kullanılan toplama, çıkarma, çarpma ve bölme temel işlemlerini gerçekleyecek yazılımlara göre çok daha basit olmaktadır. Bu bakımdan bilgisayar aritmetiği olarak adlandıracağımız bu temel işlemlerin gerçekleşmesi çok daha önemli bir sorun olarak karşımıza çıkmakta, kullanılacak aritmetiğin doğruluğu tüm sistemin çalışmasına doğrudan etkili olmaktadır.

Bu nedenle öncelikle aşağıdaki paragraflarda çalışmamızda gerçeklediğimiz bilgisayar aritmetiği ele alınacaktır.

#### 4.4 BİLGİSAYAR ARİTMETİĞİ

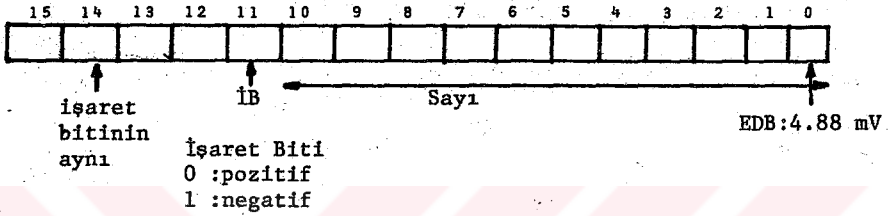
Bu çalışmada, adaptasyon algoritmamız için gerekli bilgisayar aritmetiğinin oluşturulabilmesi için öncelikle aşağıdaki sorulara cevap aranmıştır:

- ADÇ 'den gelen, analog değerın dijital karşılığı ne şekildedir?
- Kontrolör katsayılarının ondalık değerleri, alt ve üst sınırları nelerdir ?
- Kontrolör yazılımında ve adaptasyon yazılımında gereken aritmetik işlemler nelerdir ?

Bu soruların cevapları, bizim için gereken aritmetiği oluşturmamızı sağlayacaktır. Öncelikle ADÇ den gelen dijital verinin MİB 'ne alınmasındaki görünümü ele alınacak , daha sonra gerek kontrolörde gerek adaptasyon algoritmasında sırasıyla P katsayısı üzerine yapılan işlemler, I katsayısı üzerine yapılan işlemler , D katsayısı üzerine yapılan işlemler ele alınarak gerekli aritmetik oluşturulacaktır.

#### 4.4.1 BİLGİ AKTARMA İÇİN KULLANILAN VERİ KALIBI

ADÇ 'den gelen dijital verinin MİB 'ne alınmasındaki görünümü, Şekil 4.1 de gösterilen kalıpta olup 16 bitlik 2 'ye tümleyen biçimindedir.



Şekil 4.1 Analog değer karşılığı gösterilimi kalıbı  
(Kalıp I)

Örneğin, 16 sayı sisteminde bu kalıba göre,

En küçük pozitif gerilim	4,88 mV karşılığı	:	(0001) <sub>16</sub>
En büyük	" " 10 V	:	(07FF) <sub>16</sub>
En " negatif	" -4,88 mV	:	(FFFF) <sub>16</sub>
En küçük	" " -10 V	:	(8000) <sub>16</sub>

biçiminde olacaktır. Buna göre 5 V luk gerilim bilgisayarda (0400)<sub>16</sub> olarak gösterilmelidir.

Bilindiği gibi kontrol çevriminde çıkış işareti ile referansın karşılaştırılması işlemi dijital bilgisayarda yapmaktaydık. Bu durumda çıkış işareti ADÇ den Kalıp I 'de geleceğinden , referans gerilimi de aynı kalıpta yazılmalıdır. Bu durumda,

$$E = R - C$$

nin hesaplanması için 16 bitlik iki sayının çıkarılması programı oluşturulmuştur. Görüleceği gibi E değeri de Kalıp I ' de olacaktır.

## 4.4.2 P KATSAYISI İLE İLGİLİ İŞLEMLER

Kontrolörde ve adaptasyon algoritmasında P katsayısı ile ilgili işlemleri tekrar göz önüne alırsak,

a) P kontrolörde,

$$EP = P * E \quad (4.1)$$

b) Öğrenme aşamasında,

$$\Delta P_o = 0,1 \quad (4.2)$$

$$P = 1 + \Delta P_o \quad (4.3)$$

$$\text{ORAN } P_{T_A} = \frac{T_{A_i} - T_{A_o}}{\Delta P_o} \quad (4.4)$$

$$\text{ORAN } P_A = \frac{A_o - A_i}{\Delta P_o} \quad (4.5)$$

c) Adaptasyon aşamasında,

$$\Delta P = \frac{T_A - T_{A1}}{\text{ORAN } P_{T_A}} \quad (4.6)$$

$$P = P_{-1} + \Delta P \quad (P_{-1}; \text{ Bir önceki P değeri}) \quad (4.7)$$

$$A_{\text{BEK}} = \Delta P * \text{ORAN } P_A + A \quad (4.8)$$

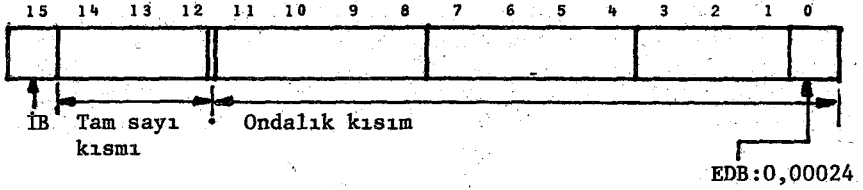
$$\Delta A_{\text{BEK}} = A - A_{\text{BEK}} \quad (4.9)$$

işlemleri yapılmaktadır. Sırasıyla bu işlemler ele alındığında,

(4.1) İŞLEMİ :

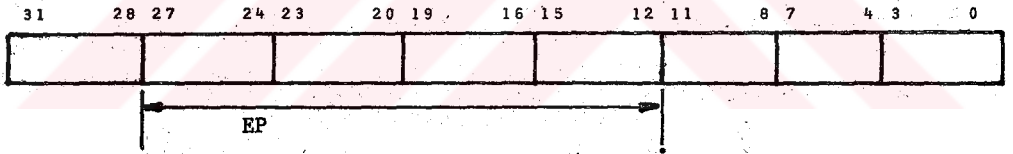
E değeri Kalıp I 'e göredir. Bilindiği gibi endüstriyel kontrolörlerde P katsayısı 0,01 - 2 aralığında değişebilmektedir (26). Bu değişim de göz önüne alınarak ve kullandığımız aritmetiğe uygun düşürebilmek amacı ile P katsayısı Şekil 4.2 ile verilen kalıpta (Kalıp II) seçilmiştir.

Buna göre P 'nin deęişim aralığı 0,00024 - 7 olacaktır.



Şekil 4.2 Sabit noktalı katsayı gösterilim kalıbı (Kalıp II)

(4.2) işlemleri görüldüğü gibi bir çarpma işlemidir. Bunun için 16 bitlik 2'ye tımleyen iki sayının çarpılması programı gereklidir. Çarpım sonucu 32 bit olarak elde edilmektedir. Sonucun 32 bitte görüntümü Şekil 4.3 teki biçimde olacaktır, EP değeri ise taralı olmayan bölgede bulunacaktır.



Şekil 4.3 32 bitlik çarpım sonucu gösterilimi

Böylece elde edilecek çarpım sonucunun kesme yanlışlığı (truncation error) bilgi aktarma aygıtının kuantalama yanlışlığı kadar olacaktır.

(4.2) ve (4.3) İŞLEMLERİ :

(4.2) nin verililişi de Kalıp I biçiminde olacaktır. 0,1 ondalık sayının Kalıp I 'e göre yazılmış 16 lık karşılığı  $(0.199)_{16}$  olacaktır.

(4.3) işlemleri için iki 16 bitlik 2'ye tımleyen sayının toplanması programı yazılmıştır. Bu durumda (4.3) işlemleri sonucu  $(1.199)_{16}$  olacaktır.

(4.4) ve (4.6) İŞLEMLERİ :

İşlem sayısını azaltmak amacıyla bu iki işlemler birlikte ele alınmış-

tir. (4.6) da (4.4) yerine konursa,

$$\Delta P = \frac{T_A - T_{A_i}}{T_{A_i} - T_{A_o}} \cdot \Delta P_o \quad (4.10)$$

olacaktır. Aşım zamanının  $T_A = k T_{\delta}$  biçiminde ölçüldüğü göz önüne alınırsa,

$$T_{A_i} = k_i T_{\delta}$$

$$T_{A_o} = k_o T_{\delta}$$

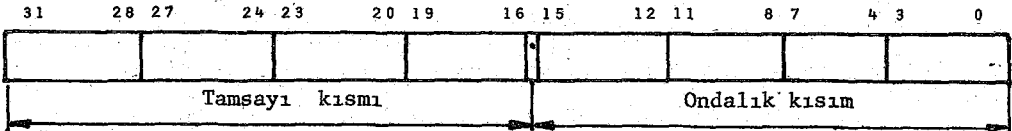
$$T_A = k T_{\delta}$$

(4.11)

değerleri (4.10) da yerine konulduğunda,

$$\Delta P = \frac{k - k_i}{k_i - k_o} \cdot \Delta P_o \quad (4.12)$$

olacaktır. Görüldüğü gibi burada bir bölme işlemi gerekmektedir. Bölme için yazılan program 16 bitlik iki sayının bölünmesi programı olup, sonuç 16 bit tamsayı, 16 bit ondalık kısım olarak 32 bitte Şekil 4.4 te verilen kalıpta elde edilmektedir.



Şekil 4.4 32 bitlik elde edilen bölme sonucu gösterilimi

(4.12) de gösterilen  $k$  katsayıları örnekleme zamanının uygun seçilmesi sonucu 1 byte'a ve farkları da 1 byte'a sığmaktadır. Bu nedenle bölme sonucunda elde edilecek değerler, dizge parametrelerinde olabilecek normal değişimler için, tamsayı kısmının bir dördlük içinde olması bekle-

nir. Ancak yazılan program beklenmeyen durumları (yani tamsayı kısmının bir dörtlükten fazla olması) giderici önlemleri içermektedir. Buna göre Şekil 4,3 te bölme sonucunun Kalıp II 'ye uygun kısmı alınarak işlemlere devam edilecektir. (4.12) de görülen çarpma işlemi ise yine aynı çarpma programı kullanılarak gerçekleştirilecektir.

(4.5) ve (4.8) İŞLEMLERİ :

(4.8) ifadesinde (4.5) ve (4.12) yerine konursa,

$$A_{BEK} = \frac{k - k_i}{k_i - k_o} (A_o - A_i) + A \quad (4.13)$$

olacaktır. Görüldüğü gibi  $A_{BEK}$  'nin hesaplanmasında daha önce bölme ile elde edilen sonuç (Kalıp II) ,  $(A_o - A_i)$  farkı (Kalıp I) ile çarpılacak ve A değeri ile toplanacaktır. Sonuçta elde edilecek  $A_{BEK}$  değeri yine Kalıp I'e uygun olacaktır.

(4.7) ve (4.9) İŞLEMLERİ :

Görülebileceği gibi , (4.7) işlemi Kalıp II 'ye uygun iki sayının toplanması, (4.9) işlemi ise Kalıp I ' e uygun iki sayının farkının alınması işlemleridir. Daha önce gerçekleştirilmiş olan 16 bitlik 2'ye tımleyen iki sayının toplanması ve çıkarılması programları bu işlemleri yapmak için kullanılacaktır.

Şimdiye kadar P katsayısı ile ilgili işlemler incelenirken, çalışmamızda gerekli olan aritmetik elde edilmiş oldu. Buna göre gerçekleştirilen alt programları bir kez daha özetlersek,

1) 16 bit 2'ye tımleyen iki sayının toplanması alt programı:

Sonuç : 16 bit 2'ye tımleyen

2) 16 bit 2'ye tımleyen iki sayının çıkarılması alt programı:

Sonuç : 16 bit 2'ye tımleyen



3) 16 bit 2'ye tmleyen iki sayının arpılması alt programı (27):  
(Booth Algoritması)

Sonuç : 32 bit 2'ye tmleyen

4) 16 bit 2'ye tmleyen iki sayının blnmesi alt programı

Sonuç : 32 bit (16 bit tamsayı, 16 bit ondalık) 2'ye tmleyen

eklindedir.

P katsayısı ile ilgili ilemlerde bir yandan bilgisayar aritmetiđi oluturulurken grldđ gibi bir yandan da yapılması gereken ilemler, kullanılan aritmetiđe uygun olacak, arpma ve blmelerde tama (overflow) olmayacak biimde dzenlendi. İntegral ve trev katsayıları ile yapılacak ilemlerde de benzer biimde dzenlemelere gidilecek ve kurulan aritmetiđin yeterliliđi sınıacaktır.

#### 4.4.3 I KATSAYISI İLE İLGİLİ İLEMLER

Kontrolrde ve adaptasyon algoritmasında I katsayısı ile ilgili olarak,

a) I kontrolrde,

$$EI = EI_{-1} + T_o * I * E \quad (4.14)$$

( $EI_{-1}$  : o andaki rneđe kadar elde edilen integral sonucu)

b) đrenme aamasında,

$$\Delta I_o = \frac{\pi}{T_{A_i}} \cdot 0,01 \quad (4.15)$$

$$\text{ORAN } I_A = \frac{A_o - A_i}{\Delta I_o} \quad (4.16)$$

c) Adaptasyon aşamasında,

$$\Delta I = \frac{A_i - A}{\text{ORAN } I_A} P \quad (4.17)$$

$$I = I_{-1} + \Delta I \quad (I_{-1}; \text{Bir önceki } I \text{ değeri}) \quad (4.18)$$

işlemleri yapılacaktır.

(4.14) İŞLEMİ :

(4.14) ifadesinde dikkat edilirse integral katsayısı daima  $T_0$  örnekleme zamanı ile çarpılmaktadır. Bu çarpımın Kalıp II 'ye uygun elde edilmesi ile (4.14) işleminin yapılması için mevcut aritmetik yeterli olacaktır. O halde integral ile ilgili işlemlerin hepsinde  $I$  katsayısı yerine  $(I \times T_0)$  çarpımını elde etmek gerekmektedir. Buna göre diğer işlemler aşağıdaki gibi düzenlenmelidir.

(4.15) İŞLEMİ :

(4.15) ifadesinde  $T_{A_i}$  yerine  $k_i \cdot T_0$  konur ve eşitliğin her iki tarafı  $T_0$  ile çarpılırsa,

$$T_0 \Delta I_0 = \frac{\pi}{k_i} 0,01 \quad (4.19)$$

elde edilir. (4.19) işlemi için mevcut bölme ve çarpma alt programları burada kullanılabilir.

(4.16) ve (4.17) İŞLEMLERİ :

(4.16) ilişkisi (4.17) de yerine konursa,

$$\Delta I = \frac{A_i - A}{A_0 - A_i} \Delta I_0 \cdot P \quad (4.20)$$

elde edilir. Her iki tarafın  $T_0$  ile çarpılması sonucunda,

$$(T_0 \Delta I) = \frac{A - A_i}{A_0 - A_i} P (T_0 \Delta I_0) \quad (4.21)$$

olacaktır. Burada işlem sırası, öncelikle bölme daha sonra P ile çarpma ve  $(T_0 \Delta I_0)$  ile çarpma şeklinde yapılacaktır.

(4.18) İŞLEMİ :

(4.18) ifadesi de  $T_0$  ile çarpıldığında,

$$T_0 I = T_0 I_{-1} + T_0 \Delta I \quad (4.22)$$

olacaktır. Toplama işlemi sonucu doğrudan doğruya (4.14) işleminde kullanıma hazırdır.

#### 4.4.4 D KATSAYISI İLE İLGİLİ İŞLEMLER

D katsayısı ile ilgili işlemler de,

a) D kontrolörde,

$$ED = (E - E_{-1}) * D / T_0 \quad (4.23)$$

b) Öğrenme aşamasında,

$$\Delta D_0 = \frac{T}{\pi} A \quad 0,01 \quad (4.24)$$

$$\text{ORAN } D_A = \frac{A_i - A_0}{\Delta D_0} \quad (4.25)$$

c) Adaptasyon aşamasında,

$$\Delta D = \frac{A_i - A}{\text{ORAN } D_A} P \quad (4.26)$$

$$D = D_{-1} + \Delta D \quad (4.27)$$

biçimindedir.

Dikkat edilirse burada, İntegral işlemlerinin tersine D katsayısının  $T_0$  örnekleme zamanına oranı gelmektedir. O halde türev ilgili işlemlerde daima  $D/T_0$  oranı elde edilmelidir. Bu durumda,

(4.23) İŞLEMİ :

$(D/T_0)$  Kalıp II 'ye uygun elde edildiğinde, bu işlem Kalıp I deki iki sayının çıkarılması ve sonucun Kalıp II 'deki bir sayı ile çarpılması işlemi olacaktır ki daha önce yapılan işlemler ile aynı biçimdedir.

(4.24) İŞLEMİ :

(4.15 ) in düzenlemesine benzer düşünce ile,

$$\frac{\Delta D_0}{T_0} = \frac{k_i}{\pi} 0,01 \quad (4.28)$$

elde edilir.

(4.25) ve (4.26) İŞLEMLERİ :

Bu işlemler de düzenlendiğinde,

$$\frac{\Delta D}{T_0} = \frac{A_i - A_0}{A_i - A_0} P \left( \frac{\Delta D}{T_0} \right) \quad (4.29)$$

olacaktır.

(4.27) İŞLEMİ :

(4.27) işlemi de,

$$\frac{D}{T_0} = \frac{D_{-1}}{T_0} + \frac{\Delta D_0}{T_0} \quad (4.30)$$

olarak gerçekleştirilecektir.

Görüldüğü gibi İntegral ve T ürev ile ilgili işlemler uygun düzenlendiğinde, kurduğumuz aritmetik yeterli olmaktadır.

## 4,5 ÖRNEKLEME ZAMANININ SEÇİLMESİ

Bölüm 3.2 de değinildiği üzere, örnekleme zamanının bilgi aktarma ve işleme sürelerinin yanında yeterince büyük olacak şekilde seçilmesi gerekmektedir. Öte yandan  $T_0$  nün büyük seçilmesi ise dizgenin yavaş olmasını gerektirecektir. Uygulamada, dijital kontrolörün olabildiğince hızlı dizgeleri kontrol edebilmesi istenecektir.

Uygulamamızda bilgi aktarma ve işleme süresi  $\approx 1$  ms olup, örnekleme zamanı  $T_0 = 10$  ms olarak seçilmiştir. Buna göre kontrol edilebilecek en hızlı dizgenin band genişliği de, (3.1) bağıntısına göre katlanılabilecek en büyük örnekleme yanılığısı % 1 seçildiğinde 1,41 r/s, % 10 seçildiğinde ise 14,1 r/s olacaktır. Görüldüğü gibi seçilen örnekleme zamanı uygulamalar için yeterli olacaktır.

Burada şu önemli noktayı belirtmek gerekmektedir; Dizgenin band genişliği daraldıkça ya da bir başka deyişle dizge yavaşladıkça örnekleme zamanının (3.1) bağıntısına göre değiştirilmesi gerekir. Tersine bir görüş olan, yavaş sistemden sık örnekleme ile daha iyi sonuç alınacağı görüşü, teorik olarak doğru olmakla beraber, bilgisayar aritmetiği bölümünde vurguladığımız gibi örnek sayısının artması algoritmamızda yer alan çarpma ve bölme işlemlerinde taşmaya neden olacaktır. Bu duruma getirilecek çözüm, bilgisayar aritmetiğinde kullanılan bit sayısının artırılmasıdır. Böyle bir çözüm işlem süresini ve programlamayı uzatarak yeni sorunlar doğuracaktır.

Sonuç olarak örnekleme zamanı, kontrol edilecek en hızlı dizgenin band genişliğinden (3.1) ilişkisinin verdiği üst sınıra eşit olarak alınmalı, bu sürenin bilgi aktarma ve bilgi işleme sürelerine göre 10-15 kat büyük olup olmadığı sınanmalıdır.

## 4.6 ADAPTASYON ALGORİTMASI YAZILIMI

Uygulama çalışmamızda gerçekleştirdiğimiz yazılım temelde simülasyon bölümünde verdiğimiz yazılıma benzerdir. Ayrıcalıklarını aşağıdaki gibi sıralamamız mümkündür :

- 1° Simülasyonda yüksek düzeyde bir dil (FORTRAN IV) kullanılmıştır. Buna karşılık burada M 6800 mikroişlemcisinin komut kümesi kullanılarak makina dilinde program yazılmıştır.
- 2° Yüksek düzeyde olan bir dilde aritmetik işlemler arşiv fonksiyonu biçiminde bulunmasına karşılık, burada aritmetik işlemler için tarafımızdan yazılan alt programlar kullanılmıştır.
- 3° Algoritmamızda yapılması gereken tüm aritmetik işlemler, Bölüm 4.4.1,2,3 ve 4 te açıklandığı biçimde gerçekleştirilmiştir.

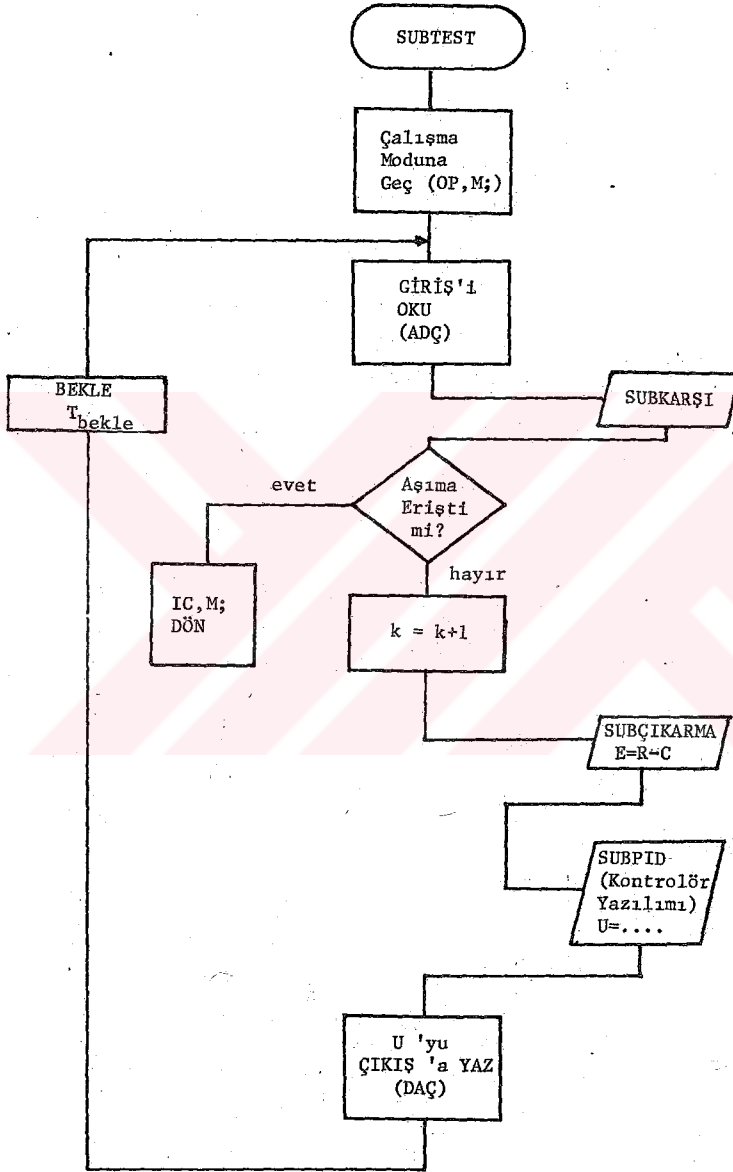
Algoritmaya ilişkin tüm program Ek 4.1 de verilmiştir. Programın daha kolay izlenebilmesi amacıyla makina kodları yanı sıra işlemi çağrıştıracak kısaltmalar da verilmiştir.

Bu bölümde sadece, ana programa ilişkin akış diyagramı (Şekil 4.5) ve simülasyon çalışmamıza benzer biçimde oluşturulan TEST alt programına ilişkin akış diyagramı (Şekil 4.6) verilmiştir.

## 4.7 DENEYLER VE SONUÇLARI

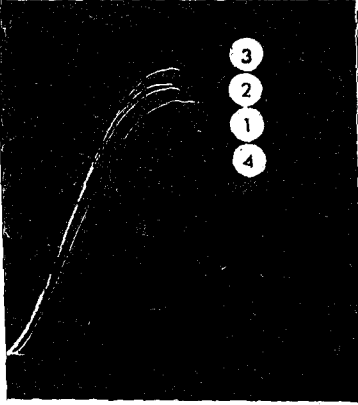
Bu bölümde, gerçekleştirdiğimiz yazılımın ve donanımın çalışmasını izlemek üzere yapılan deneylerin sonuçları verilecektir. Burada da simülasyon çalışmamızda olduğu gibi dizge analog bilgisayarda simüle edilmiştir. Dizgeye ilişkin program Şekil 3.2 deki gibidir. Yapılan deneylere ilişkin birkaç sonuç Şekil 4.7 a,b,c ve d de verilmiştir.





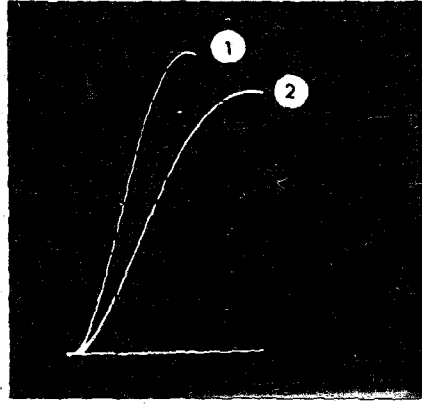
Şekil 4.6 TEST alt programına (SUBTEST) ilişkin akış diyagramı





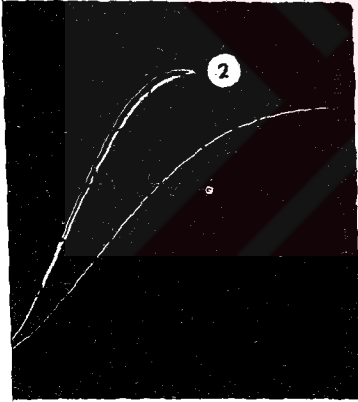
a) Öğrenme

- 1° İstenen durum
- 2° P etkisi
- 3° I etkisi
- 4° D etkisi



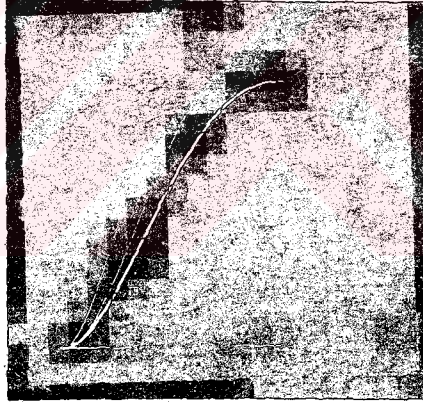
(b) Adaptasyon

- 1° Adaptasyon başlangıcı
- 2° Adaptasyon sonu



c) Adaptasyon

- 1° Adaptasyon başlangıcı
- 2° Adaptasyon sonu



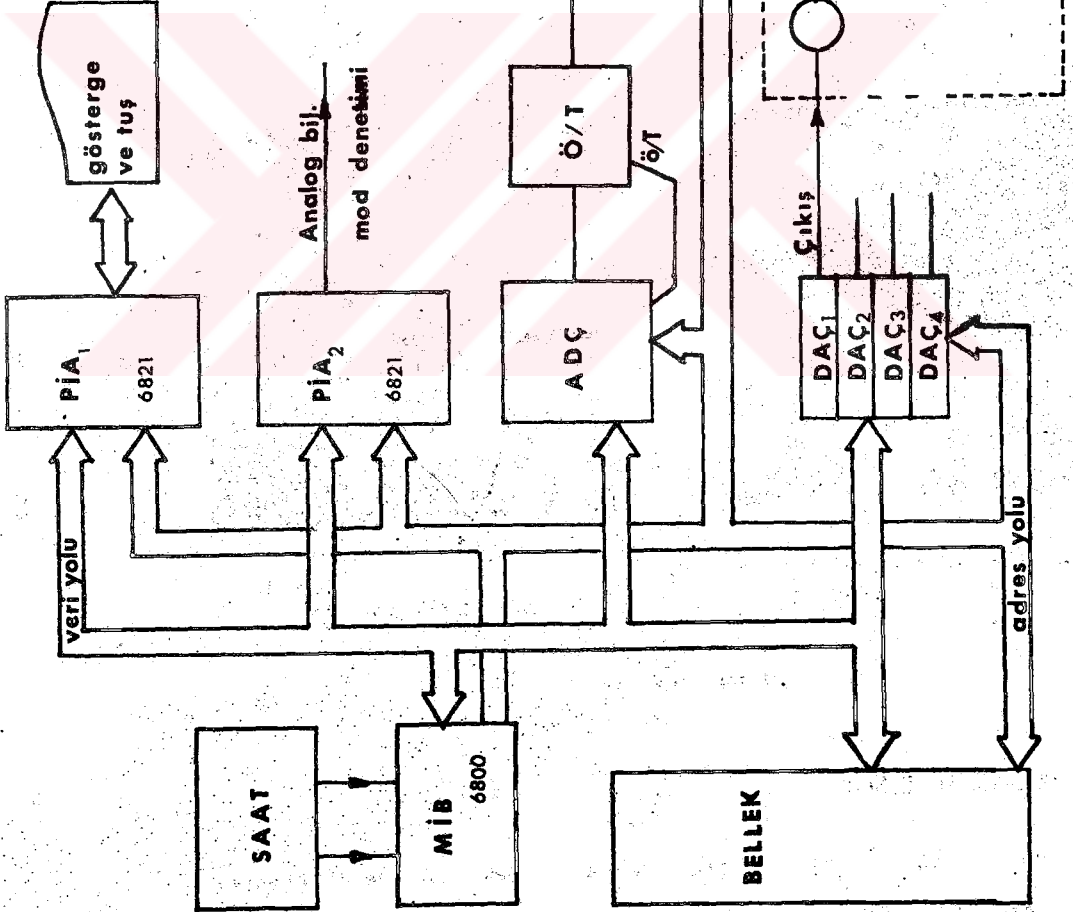
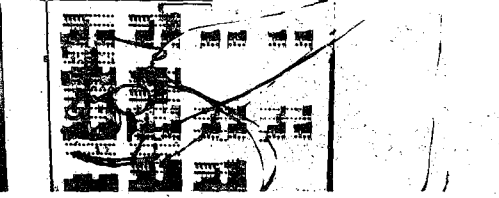
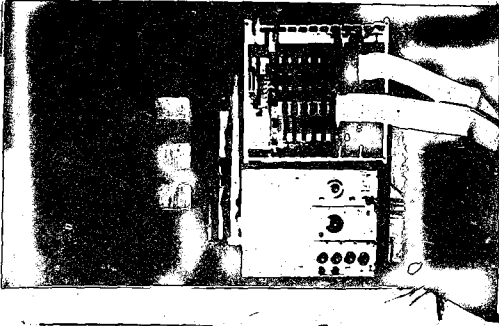
(d) Adaptasyon

- 1° Adaptasyon başlangıcı
- 2° Adaptasyon sonu

Şekil 4.7 Gerçeklenen adaptif kontrolöre ilişkin deney sonuçları

#### KULLANILAN DONANIM

Uygulama çalışmamızda kullanılan donanım Şekil 4.8 de verilmiştir. Bu donanımda görülen elemanlarz ilişkin teknik özellikler daha önce miştir.





**5.** bölüm

**SONUÇLAR**

Adaptif olarak kontrol edilecek dizge çıkışının aşım zamanı ve aşım değerini, dizgede zamanla olabilecek değişikliklere karşı sabit tutma ve bunu bir mikrobilgisayarla gerçekleştirme çalışmamızın getirdiği yeniliktir.

Adaptif kontrolör, sürekli olarak test sinyali uygulayarak dizge çıkışını izler ve bu izlemeden öğrendiği bilgilere göre adaptasyon işlemi- ni yerine getirir. Dolayısıyla bu tezde önerilen adaptif kontrolör, öğre- nerek çalışan bir adaptif kontrolördür.

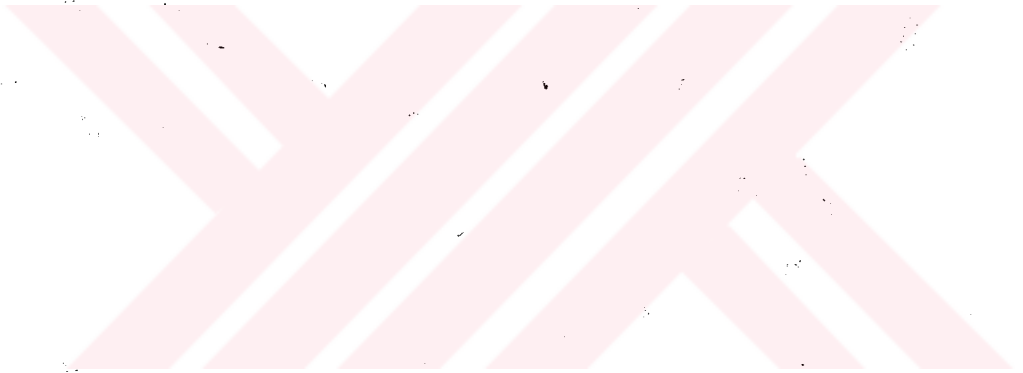
Sistem davranışının kontrolör parametrelerine duyarlılığını hesapla- yabilmek için kontrolör parametrelerine uygun artımlar verilerek sistem davranışı üzerine etkileri ölçülmüştür. Sistemin istenen davranışını boz- mayacak şekilde uygun parametre artımının nasıl hesaplanacağı gösterilmiş- tir.

Algoritmamız aşımı olan ikinci dereceden davranışlı dizgeler için doğrudan doğruya kullanılabilir. Ancak algoritmanın özünü oluşturan öğren- me nedeniyle genelde her türlü dizgeye uygulanabilir. Bu uygulamada önem- li olan dizgeyi karakterize edecek kriterin saptanmasıdır.

Yöntemin çalışabilirliği ve güvenilirliği simülasyon çalışmaları ile olabilecek tüm olasılıklar için denenmiş ve doğrulanmıştır.

Mikrobilgisayarla gerçekleştirdiğimiz bu adaptasyon işleminde; örnekle- me zamanının, bilgi aktarma aygıtının ve bilgisayar aritmetiğinin en uygun şekilde nasıl seçilmesi gerektiği ortaya konulmuş ve gerçekleştirme bu görüş- lerle yapılmıştır.

Gerçekleme 8 bitlik mikrobilgisayar kullanılarak ucuz ve güvenilir biçimde yapılmıştır. Bugünün ülkemiz fiyatları ile adaptif kontrolörün ma- liyeti 10.000.- TL civarındadır.



**6** ■ bölüm

**KAYNAKLAR**

- (1) MISHKIN, E. ; BRAUN, L., Jr. ; Adaptive Control Systems, McGraw-Hill Book Company, Inc., 1961.
- (2) ASTROM, K.J. ; Adaptive Control, Arab School of Science and Technology-4. Yaz Okulu (24. Ağustos-10 Eylül 1981), 1981, 7.1-7.21 .
- (3) EVELEIGH, V. W. ; Adaptive Control and Optimization Techniques, McGraw-Hill Inc., 1967.
- (4) ISERMANN, R. ; Digitale Regelsysteme, Springer-Verlag, Berlin/Heidelberg, 1977.
- (5) HANG, C.C. , PARKS, P.C. ; Comparative Studies of Model Reference Adaptive Control Systems, IEEE Vol. AC-18, No.5, Oct. (1973), 419-427.
- (6) MONOPOLI, R.V. ; Model Reference Adaptive Control with an Augmented Error Signal, IEEE Vol. AC-19, No.5, Oct. (1974), 474-484.
- (7) LANDAU, I.D. ; A Survey of Model Reference Adaptive Techniques, Automatica Vol.10 , (1974), 353-379.
- (8) ASTROM, K.J. , WITTENMARK, B. ; On Self Tuning Regulators, Automatica Vol.9 , (1973), 185-199.
- (9) KURZ, H. , ISERMANN, R., and SCHUMANN, R. ; Experimental Comparison and Application of Various Parameter Adaptive Control Algorithms, Automatica Vol.16, (1980), 117-133.
- (10) YAHAGI, T. ; A Method for Adaptive Control Using a Minicomputer, IEEE Vol. IECI-26, No.1, Feb. (1979), 16-21.
- (11) HUGUENIN, F., SCHAUFELBERGER, W. ; Model Reference Adaptive Gain Control Using Microcomputers, Digital Computer Applications to Process Control, Van Nauta Lemke, ed., IFAC and North-Holland Pub.Com. (1977), 497-504.
- (12) GHANDAKLY, A. ; Design of an Adaptive Control System Using an On-Line Minicomputer in Real Time, Arab School of Science and Technology-4. Yaz Okulu (24. Ağustos-10 Eylül) 1981, B.1-B.3 .
- (13) KUO, B.C. ; Digital Control Systems, Holt-Saunders, 1980.
- (14) ÜLGÜR, M.M. ; Otomatik Kontrol Sistemlerinde Davranış Kriterleri, Ders Notu, İTÜ Elektrik Fakültesi Ofset Baskı Atölyesi, 1973.
- (15) KUO, B.C. ; Automatic Control Systems, Prentice-Hall Inc., 1975.
- (16) BOWER, J.L., SCHULTHEISS, P.M. ; Introduction to the Design of Servomechanism, John Wiley & Sons Inc., 1958.
- (17) D'AZZO, J.J., HOUPIS, C.H. ; Linear Control System Analysis and Design, McGraw-Hill Inc., 1975.

- (18) AYLOR, J.H. , RAMEY, R., and COOK, G. ; Design and Application of A Microprocessor PID Predictor Controller, IEEE Vol. IECI-27, No.3, Aug. (1980), 133-137.
- (19) CARUTHERS, F.P., LEVENSTEIN, H. ; Adaptive Control Systems, Pergamon-Press, 1963.
- (20) EAI; PACER 100 Reference Handbook, Electronic Associates Inc., 1971.
- (21) EAI; PACER 500 Reference Handbook, Electronic Associates Inc., 1971.
- (22) ADALI, E. ; Dağıtılmış Bilgisayarlarla Denetim, Doçentlik Tezi, İTÜ Elektrik Fakültesi, 1980.
- (23) KATZ, P. ; Digital Control Using Microprocessors, Prentice-Hall Inc., 1981.
- (24) MOTOROLA ; MEK-6802 D5E(D1) Microcomputer Evolution Bord User's Manual, Motorola Inc., 1980.
- (25) BURR-BROWN ; Microcomputer Analog I/O Systems, Data Shield, Burr-Brown Corporation, USA, July 1979.
- (26) PHILIPS ; Industrial Automation, Philips Industrial Catalogue PCS-700, Philips Inc., Germany, 1979.
- (27) MOTOROLA ; M 6800 Microprocessor Applications Manual, Motorola Inc., Switzerland, 1975.