



**EMO**



**KTÜ**



**TÜBİTAK**

# ÖNSÖZ

Giderek gelenekselleşen Elektrik Mühendisliği Ulusal Kongrelerinin beşincisinde Trabzon'da buluşuyoruz. EMO ile KTÜ Elektrik-Elektronik Mühendisliği Bölümü'nün işbirliği ve TÜBİTAK'ın kalkışıyla gerçekleşmekle olan Kongremizin başarılı ve verimli geçmesi umudundayız. Kongre sonuçlarından kıvanç duymak istiyoruz.

Kongre'ye, bugün-? kadar yapılmış çalışmalar ve yayınlanmış duyurulardan da anıncı:nc<:ği gibi, bilinen yöntemlerin yanı sıra gelecek yıllara deneyim aktarabilecek yeni yaklaşırlar uygulanmaya çalışılmıştır. Bildin' özetlerinin değerlendirilmesine katılan uzman sayısının sistematik olarak artıınınası, değerlendirme biçiminin daha da nesnelleştirilmesi, biidiü kitabında yeni yazım ve sunuş biçimlerinin oluşturulması gibi teknik gelişmelerin dışında ilginç olacağı sanılan panellerle günce! sorunların irdelenmesi ve yöresel öğelerle sosyal etkinliklere renk kalması amaçlanmıştır.

Kongrenin hazırlık ve düzenleme çalışmalarında bazı aksaklıklar olmuştur. Öncelikle kongre katarının olması gerekenden daha geç alınabilmiş olması, özel değerlendirme sürecinin posta trafiğinin çok yoğun olduğu bayram dönemlerine rastlaması İem Yürütme Kurulu'nu İem de Kongre'ye katılmak isteyenleri zor durumda bırakmıştır.

Kongrenin düzenlenmesi sırasında edinilen deneyimler ışığında sorunları çözücü ilkesel önerilerin ortaya konması yararlı olacaktır. Bunları kısaca sıralayabiliriz. Örneğin 6. Kongre'nin ya da kısaca EMUK'95'm nerede ve ne zaman yapılacağını şimdiden kararlaştırmak gerekmektedir. Bundan sonra Konferans olarak adlandırılması daha uygun olacak Kongre için süekli ya da uzun süre görevli bir 'Ulusal Düzenleme Kurulu'nun oluşturulması ve bu Kurul'un temel ilkesel karar ve yöntemleri üretmesi daha elverişli olacaktır. Kongre'nin yapılacağı konumdaki işleri ise 'Yerel Düzenleme Kurulu' üstlenmelidir. 'Bilimsel Değerlendirme Kurulu'nun da ayrıntılı bir sınıflandırma ve nitelik belirlenmesi ile bir kere oluşturulması, yalnızca gelişen koşullara göre güncelleştirilmesi düşünülebilir.

CMUK, böylesi bir yapılaşma ile daha sağlıklı, zaman planlaması daha verimli bir konferansa dönüşecektir kanısındayız. Örneğin bu durumda bildiri tam metninin de değerlendirme ve denetim sürecine girmeleri olanaklı kılınacak, şu ana kadar ancak Yürütme Kurulu'nun uyumlu olarak bilincine varabildiği teknik sorunlar ortadan kalkacaktır. Konferansda da içerik ve düzey açısından belirli bir iyileştirme sağlanabilecektir. Bunu en yakında, EMUK'95'de gerçekleşmiş olarak görmek dileyordüyük.

Bilindiği gibi Kongremiz Elektrik, Elektronik-Haberleşme, Kontrol ve Bilgisayar Sistemleri alanlarında bilimse!- teknolojik özgün katkıların tartışılıp değerlendirilmesi ile araştırma, geliştirme, uygulama ve eğitim süreçlerindeki kişi ve kuruluşların birbiriyle doğrudan iletişimini sağlamayı amaçlamaktadır. Ayrıca sosyal yakınlıkla ve dayanışmaya da

katkıda bulunmaktadır. Ancak Kongre ve onunla birlikte oluşturulan sergi/fuvarın çok değerli bir 'Mesleki Eğitim ve Geliştirme' aracı olduđu bilincinin kiři ve kurumlarda dafia çok yerleşmesi için çaba gösterme geređi de ortaya çıkmaktadır.

Kongrenin gerçekleşmesini sađlayan, hazırlık ve düzenlemeleri üstlenen KTÜ, EMO ve TÜBİTAK'a, oluşturulmuş olan kurulların üyelerine, ayrıca burada adlarını saymakla bitmeyecek kiři ve kamu - özel - akademik nitelikli kuruluşlara, yardım ve katkıları nedeniyle, Kongre'nin yararlı sonuçlarını paylaşacak olan topluluđumuz adına teşekkürlerimizi sunmak isteriz.

Kongremizin başarılı ve verimli bir biçimde gerçekleşmesi, ülkemiz için bilimsel - teknolojik kazanımlar üretmesi dileđiyle Yürütme Kurulu olarak saygılarımızı iletiriz.

Doç. Dr. Güven ÖNBİLGİN  
Yürütme Kurulu Başkanı

# ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

## YÜRÜTME KURULU

Güven ÖNBİLGİN (K1U)  
Yakup AYDIN (EMO) Sefa AKPINAR (KTU)  
Canan TOKER (ODTÜ) Kaya DOZOKLAR (EMO)  
Hasan ÜİNCER (KTU) A.Oğuz SOYSAL (IU)  
Abdul İlah SEZGİN (KTU) İrfan SENLİK (EMO)  
Kenan SOYKAN (EMO) Y.Nuri SEVGEN (EMO)

## DANIŞMA KURULU

Rasim ALDEMİR (BARMEK) Mehmet KESİM (Anadolu U)  
Teoman ALPTURK (TMMOB) Macit MUTAF (EMO)  
Ahmet ALTINEL (TEK) Erdinç ÖZKAN (PTT)  
İbrahim ATALI (EMO) Kamil SOGUKPINAR (TETSAN)  
Malik AVİRAL (ELİMKO) Sedat SİSBOT (METRONİK)  
Emir BİRGÜN (EMO) Atıf URAL (Kocaeli U.)  
Sıtkı CİĞDEM (EMO) İ. Ata YİĞİT (EMO)  
R. Can ERKÖK (ABB) Fikret YÜCEL (TELETAS)  
Bülent ERTAN (ODTÜ) Hamit SERBEST (CU)  
Uğur ERTAN (BARMEK) Canan TOKER (ODTÜ)  
İsa GÜNGÖR (EMO) Nusret YUKSELER (ITU)  
Ersin KAYA (Kaynak) Kemal ÖZMEHMET (DEU)  
Okyay KAYNAK (Boğaziçi U)

## SOSYAL ETKİNLİKLER KURULU

Y. Nuri SEVGEN (EMO)  
Necla ÇORUH (PTT) Hatice SEZGİN (KTU)  
Esen ÖNKİBAR (TEK) Yusuf TANDOĞAN (PTT)  
Abdullah SEZGİN (KTU) Ömer K. YALCIN (TELSER)

## SEKRETERLİK HİZMETLERİ

Necmi İKİNCİ (EMO) Elmas SARI (EMO)

BİLİMSEL DEĞERLENDİRME KURULU

Cevdet ACAR (ITU)  
İne i AKKAYA (ITU)  
A.Sefa AKPINAR (KTU)  
Ayhan ALTINTAS (BiI.U)  
Fuat AfvıDAY (ITU)  
Fahrettin ARSLAN (IU)  
Murat ASKAR (ODTÜ)  
Abdu I lah ATALAR (Bi I .U)  
Sel im AY (YTU)  
Ümit AYGOLU (ITU)  
Atalay BARKANA (Anadolu U)  
Mehmet BAYRAK (Selçuk U)  
Atilla BİR (ITU)  
Gal ip CANSEVER (YTU)  
Kenan DANIŞMAN (Erciyes U)  
Ahmet DERVISOGLU (ITU)  
Hasan D INCER (KTU)  
M.Sezai D INCER (Gazi U)  
Gün sel DURUSOY (İTU)  
Nadia ERDOĞAN (ITU)  
Aydan ERKMEN (OÜTU)  
İsmet ERKMEN (ODTÜ)  
H.Büİent ERTAN (ODTÜ)  
Selçuk GEÇİM (Hacettepe U)  
Cem GÖKNAR (İTU)  
Remzi GULGUN (YTU)  
Filiz GUNES (YTU)  
İrfan GÜNEY (Marmara U)  
Fikret GÜRGEN (Boğaziçi U)  
Fuat GURLEYEN (İTU)  
Cemi I GURUNLU (KTU)  
Nurdan GUZELBEYOĞLU (İTU)  
Emre HARMANCI (İTU)  
Altuğ İFTAR (Anadolu U)  
Kemal İNAN (ODTÜ)  
Asım KASAPOĞLU (YTU)  
Adnan KAYPMAS (İTU)  
Ahmet H. KAYRAN (İTU)  
Mehmet KESİM (Anadolu U)  
Erol KOCAOGLAN (ODTÜ)  
Muhammet KOXSAL (İnönü U)  
Hayrettin KÖYMEN (Bil. U)  
Hakan KUNTMAN (İIU)  
Tamer KUTMAN (ITU)  
Duran LEBLEBİCİ (ITU)  
Kevork MARDİKYAN (ITU)  
A. Faik MERGEN (ITU)  
Avni MORÜUL (Boğaziçi U)  
Güven ÖNBİLGİN (KTU)  
BUİent ÖRENCİK (İTU)  
BUİent ÖZGUC (Bi I .U)  
A.Büİent ÖZGÜLER (Bi I .U)  
Yı Imaz ÖZKAN (ITU)  
Muzaffer ÖZKAYA (ITU)  
Kemal ÖZMEHMET (DEU)  
Osman PALAMUTCUOGLU (İTU)  
Erdal PANAYIRCI (İTU)  
Ha I it PASTACI (YTU)  
Ahmet RUMELİ (ODTÜ)  
Bülent SANKUR (Boğaziçi U)  
M.Kemal SARIOGLU (IT"U)  
Müzeyyen SARITAS (Gazi U)  
A.Hami t SERBEST (Çil)  
Osman SEVAİOGLU (ODTÜ)  
A.Oğuz SOYSAL (IU)  
Taner SENGÖR (YTU)  
Emi n TACER (İTU)  
Nesrin TARKAN (İTU)  
Mehmet TOLUN (ODTÜ)  
Osman TONYALI (KTU)  
Ersin TULUNAY (ODTÜ)  
Nejat TUNCAY (İTU)  
At ı f URAL (KocaeI i U)  
Alper URAZ (Hacettepe U)  
Gökhan UZGÖREN (IU)  
Yi I d ı r im UCTUĞ (ODTÜ)  
Asaf VAROL (Fırat U)  
Sıddık B. YARMAN (IU)  
Mümtaz YILMAZ (KTU)  
Melek YÜCEL (OÜTU)  
Nusret YUKSELER (İTU)  
Selma YUNCU (Gazi U)

## ÖĞRENCİ DERS ALMA FORMLARININ BİLGİSAYAR KONTROLLÜ OPTİK OKUYUCU İLE DEĞERLENDİRİLMESİ

Sabri ÇELİK

Osman COŞKUN

Ferhat DALDABAN

Erciyes Üniversitesi Mühendislik Fakültesi, Elektronik Bölümü Kayseri

### ÖZET:

Öğrencilerin aldıkları dersleri, Ders Alma Formlarından, ders listelerine geçebilme işlemini hızlandırmak için bilgisayar kontrollü bir optik okuyucu dizayn edilmiş ve çalıştırılmıştır.

Özel olarak hazırlanan bir forma öğrenciler aldıkları dersleri işaretlemektedirler. Bu formlarda işaretlenen dersler **CNY 70 Optocoupler** elemanı kullanılarak yapılan optik okuyucuda okunarak bilgisayar tarafından değerlendirilmekte ve ana bilgisayar sistemine transfer edilmektedir. Daha sonrada istenilen derse ait sınıf listesi anında alınabilmektedir.

### 1. GİRİŞ:

Ders geçme sisteminin uygulandığı üniversitelerde öğrenciler, her dönem için almak istedikleri dersleri Ders Alma Formlarına işlemekte ve işlenen bu dersler formlardan ders listelerine geçilerek ilgili öğretim elemanına gönderilmektedir. Ders alma listelerinin hazırlanması işlemi, 2 memur görevlendirilen ve ortalama 1000 öğrencisi olan bir fakültede en az 10 gün sürmektedir.

Yapılan bu çalışmayla öğrencilere normal Ders Alma Formlarının yanında birde **Optik Okuyucu Ders Alma Formu** doldurtulmaktadır. Bu formda öğrenci numaraları ve dersler kodlanmıştır. Optik okuyucu üzerine konulan formlardaki kurşun kalemle boyanmış yerler, CNY 70 tarafından yayılan ışığı yuttuğu için çıkışta hiçbir gerilim oluşmamaktadır. Diğer yerler ışığı yansıttığı için transistör iletme geçecektir. İşte transistörlerin bu şekilde iletimde olması veya olmaması durumu, bilgisayara takılan bir giriş/çıkış kartı vasıtasıyla okunarak, öğrenci numarası ve aldığı ders kodları oluşturulan bir dosyaya kayıt edilmektedir. Sistemde 80286 mikro işlemci kullanan bir AT bilgisayar kullanılmaktadır. Bu bilgisayara 144 adet giriş/çıkış kanalı bulunan bir kart takılmıştır. Bu kartın çıkışları doğrudan optik okuyucuya bağlanmıştır. BASIC dilinde yapılan bir programla da bu karta gelen sinyaller değerlendirilerek, programa daha önceden girilen ders isimleri ve kodlarıyla karşılaştırılma yapılmaktadır. Yapılan bu karşılaştırma sonucunda öğrencinin aldığı ders kodları o öğrencinin eğitim gördüğü bölümün **Sequettinal** formda oluşturulan dosyasına kayıt edilmektedir. Bu bilgiler daha sonra bir transfer programı tarafından çok kullanıcı **Btrieve** dosya formatında indeks dosya haline getirilmektedir. Bu indeks dosya çok kullanıcı Novell İşletim Sistemi yüklü olan ana bilgisayar sistemine aktarılmakta ve COBOL dili ile yazılan **Öğrenci Otomasyon Programı** nda kullanılmaktadır.

2. ELEKTRONİK DONANIM

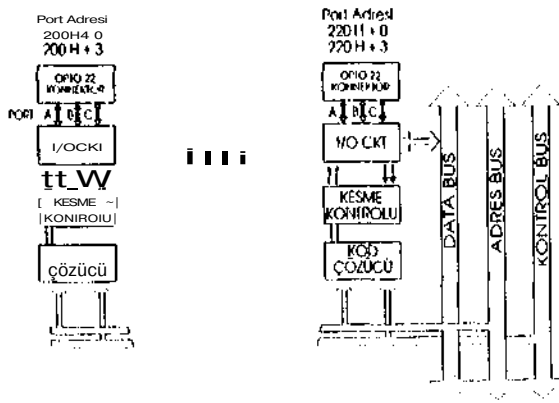
### 2.1. Bilgisayar

Sistemde kullanılan bilgisayar 80286 mikro işlemci kullanan AT bir bilgisayardır. Bilgisayarda Hercules ekran kartı, 40 MB'lık bir Hard Disk ve 1.44 inçlik bir Floppy Disket üniteleri bulunmaktadır. Bilgisayarın genişleme yuvasına takılan giriş / çıkış kartından data bus'a gelen bilgi Hard diskte depolandıktan sonra transfer programıyla

ana bilgisayara aktarılmaktadır [1].

## 2.2. Giriş - Çıkış Kartı

Bilgisayarın genişleme yuvasına takılan 8 bitlik bu kartla 144 adet giriş/çıkış hattının kontrolü yapılabilmektedir. Bölümlerin 4 sınıf olmasından hareketle her sınıf için maximum 24 adet ders seçme imkanı tanınmıştır. Toplam 4 sınıf için 96 adet giriş kullanılmaktadır. Öğrenci numarasını tesbit içinde ayrıca 40 adet giriş kullanılmaktadır. Bir girişte okumayı onaylatmak için kullanılmaktadır. Giriş/Çıkış kartı 200H adresinden başlayıp 24 Byte'lık hafıza bölgesinde tanımlanmaktadır. Bu adreslerin okutulmasıyla öğrencilerin aldığı dersler tesbit edilmektedir. Şekil - 1 'de giriş çıkış kartı iç yapısı blok diyagramı görülmektedir [2].



Şekil - 1. Giriş / Çıkış Kartı İç Yapısı

## 2.3. Optokuplör Kartı

136 adet CNY 70 optokuplör elemanı bir plaket üzerine uygun bir şekilde monte edilmiştir. Monte edilen bu sekile uygun olarak dizayn edilen optik okuyucu ders alma formu şekli Şekil 2'de görülmektedir. Optokuplör girişindeki led diyot bir direnç vasıtasıyla + 5V 'luk bir kaynağa irtibatlandırılmıştır (Şekil 3). Optik okuyucu ders alma formu cihaz üzerine konulduğunda, led diyodun yaydığı ışık

kalemle hazırlanmış yere denk geldiğinde ışığı yutmakta ve yansıma olmamaktadır. Bu anda optokuplör içindeki transistor yalıtımdadır ve giriş/çıkış kartına lojik 1 seviyesinde bir sinyal gitmektedir. Karalanmış olan yerlerden ise yansıma olacağı için transistor iletime geçmekte ve giriş/çıkış kartına lojik 0 sinyali ulaşmaktadır [3]. Optik okuyucu ders alma formunda karalanması gereken dairelerin yarısı bile karalanırsa sonuçta doğru okuma yapılabilmektedir. Transistörün kollektoruna bağlı olan direnç değerinin değiştirilmesiyle okuma hassasiyeti değiştirilebilmektedir [4].

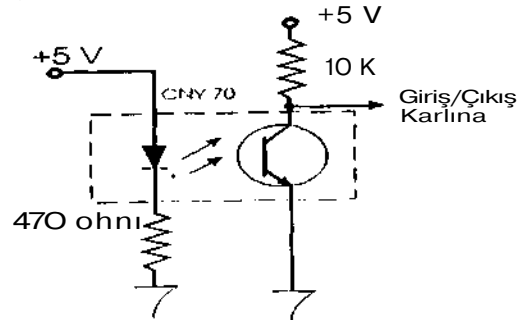
UKuvaa ONrtmMtTMI MOHMD! «UKMU(01.TUJ  
KLU TBONIK BA UMO DKBS ALMA FORMU

MRENCİNİN:  
AOI SOYADI: \_\_\_\_\_  
NUMARASI: \_\_\_\_\_  
BÖÜMÜ: \_\_\_\_\_  
SINIFI: \_\_\_\_\_  
İMZA: \_\_\_\_\_

0	1	2	3	4	5	6	7	8	9
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

MATEMATİK	İLKİM	İSTANBUL	İSTANBUL
B W .V	İLÜK M	İSTANBUL	İSTANBUL
< > &EX.GVY TİM.	EM AL. TED	BUI DEV LM	İSTANBUL
M 11 MAL29-ÇİLOL	İM S	İOKSEYOSH	m 111 İKAR.ÖL.ÖNİŞ
BÖL.ÜLWİ	Em...	027	İM 108 İSTANBUL
m a İTİ.ÖL.C.LAM	EH 11	El	m 101 Mutmon
İSTANBUL	İSTANBUL	İSTANBUL	İM 107 İSTANBUL
İSTANBUL	İSTANBUL	İSTANBUL	İM 106 İSTANBUL
İSTANBUL	İSTANBUL	İSTANBUL	İM 105 İSTANBUL
İSTANBUL	İSTANBUL	İSTANBUL	İM 104 İSTANBUL
İSTANBUL	İSTANBUL	İSTANBUL	İM 103 İSTANBUL
İSTANBUL	İSTANBUL	İSTANBUL	İM 102 İSTANBUL
İSTANBUL	İSTANBUL	İSTANBUL	İM 101 İSTANBUL

Şekil -2 : Optik Okuyucu Ders Alma Formu



Şekil - 3 : Optokuplör Bağlantı Şeması

### 3. BİLGİSAYAR PROGRAMI:

#### 3.1. Programlama Esasları:

Sistemin çalışmasını sağlayan program BASIC diliyle yazılmıştır [5]. Optokuplörlerden gelen lojik bilgi port adreslerinden doğrudan okunmaktadır. 144 Bitlik bilginin okunması işlemi yaklaşık 100 ms sürmektedir. Okuma işlemi yapılır yapılmaz kağıttaki karalı olan yerlerin görüntüsü ekrana yansımaktadır. Yani kağıt, optik okuyucu üzerindeyken kontrol imkanı olmaktadır. Şekil - 4 ' de optik okuyucu tarafından okunan bir formun ekrandaki görüntüsü görülmektedir. Okunan bu değerler her bölüm için ayrı bir dosyada toplanmaktadır. Dosyada toplanan bu ders kodlarına öğrenci numarası, yıl dönem gibi diğer bilgiler ilave edilmektedir.

ERCİYES ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ	00000
OPTİK OKUYUCU DERS ALMA FORMU GİRİŞ EKRANI	0000 1
	00>0 2
0000<00000000000000000000000000000000	0000 3
00000000<<0000<>>00000000000000000000	0<00 4
00000000000000000000000000000000000000	0000 5
00<0<00000<0000>00000000000000000000	000> 6
Öğrenci No: 6247	•000 7
Devam ? (E/H)	0000 8
	0000 9

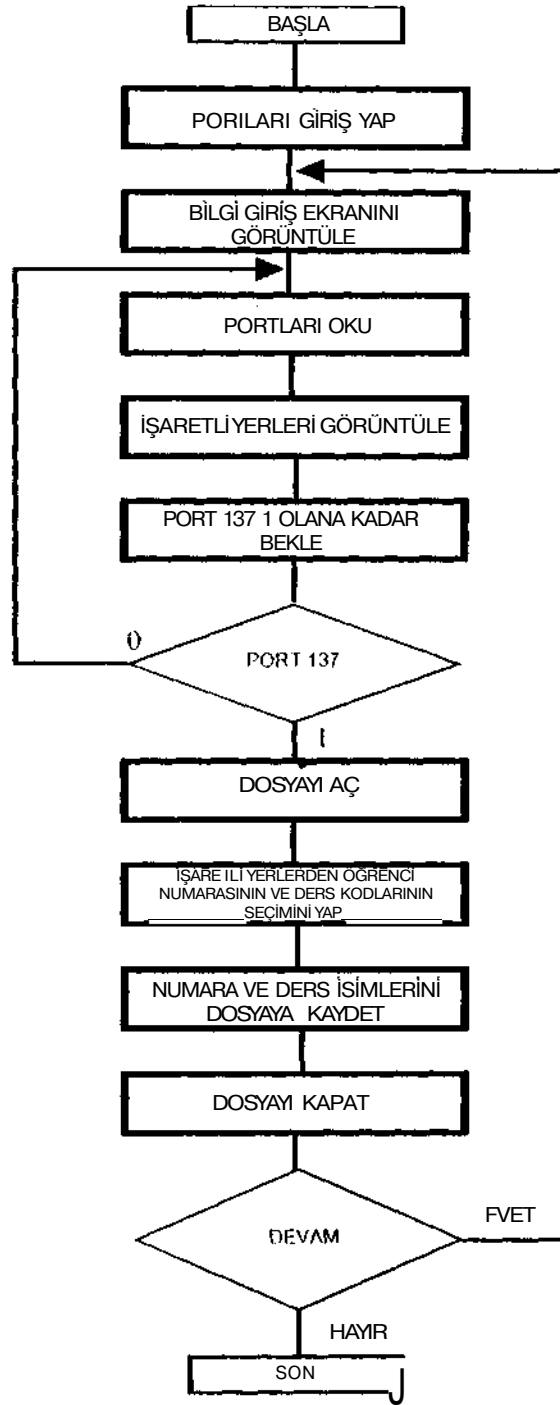
Şekil 4: Optik Okuyucu tarafından okunan bir formun ekrandaki görüntüsü

#### 3.2. Okuma Programı:

BASIC diliyle yapılan ve optik okuyucudan gelen bilgileri değerlendiren programa ait akış diyagramı Şekil - 5 ' te görülmektedir.

Programın çalışması şu şekildedir.

Birinci adımda giriş/çıkış kartının bütün portları giriş yapılacak şekilde programlanmaktadır. İkinci adımda ekranda, bilgigiriş ekranı görüntülenmektedir. Okuma işlemi yapılırken öğrenci numarasının doğru olup olmadığının kontrol edilmesi için okunan bu numara ekranda görüntülenmektedir. Doğru olan okuma sonunda onay anahtarına



Şekil - 5 Program Akış Diyagramı



basılarak ders bilgileri bilgisayara geçirilir. Daha sonra optik okuyucudan gelen portlardaki değerler okutulmuş yerler ekrana bir kare şeklinde yansıtılmaktadır. 137. giriş (onay anahtarı) lojik 1 olduğu anda okuma işlemi döngüsü dışına çıkılmakta ve bilgilerin yazılacağı dosya açılmaktadır. Noktalar erlere denk gelen ders isimleri aynı formdan okunan öğrenci numarasıyla birlikte açılan bu dosyaya yazılmaktadır. Bu işlemden sonra dosya kapatılmakta ve yeni bir giriş yapıp yapılmayacağına dair **Evet/Hayır** onay istenmektedir. **E** tuşuna basılırsa program başlangıç noktasına geri dönmektedir. **H** tuşuna basıldığında ise program sona ermektedir.

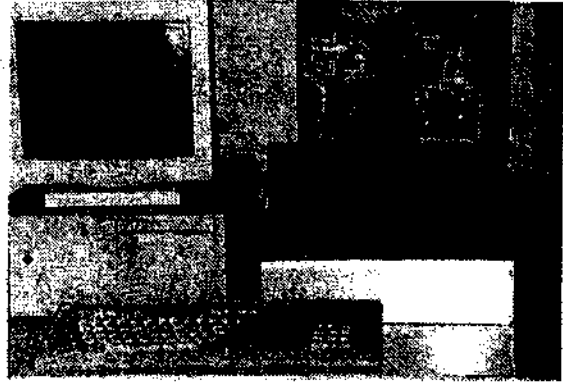
### 3.3. Transfer Programı:

Ders alma formlarının teker teker okutulması işlemi bitirildiği anda, okuma programının oluşturduğu sıralı dosya bu transfer programıyla index dosya haline dönüştürülmektedir. Ana bilgisayar sisteminde gerekli bilgilerin alınabilmesi için COBOL diliyle **Öğrenci Otomasyon Programı** yazılmıştır. Bu programda öğrencilerin aldığı dersler optik okuyucudan okunduğu gibi teker teker kodlarıyla beraber bilgisayar klavyesinden de girilebilmektedir. Bu bilgilerin değişik formatlarda listelenebilmesi için index dosya formatı kullanılmıştır. Aynı zamanda bu dosya çok kullanıcı ortamında da kullanılabilir.

### 4. SONUÇ:

Yapılan bu sistemle Erciyes Üniversitesi Mühendislik Fakültesi Elektronik ve Makina Mühendisliği Bölümlerinde öğretim gören bini aşkın öğrencinin aldığı dersler, bir gün gibi bir sürede optik okuyucu tarafından değerlendirilmekte ve daha sonra listeler halinde alınmaktadır. Optik okuyucuyu daha verimli çalıştırabilmek için bilgisayar programında yapılacak bir değişiklikle transfer programının ortadan kaldırılmasına çalışılacak ve optik okuyucu ana bilgisayar

sistemine doğrudan bağlanacaktır.



Şekil - 6 :Optik Okuyucu Ünitesi Resmi

### KAYNAKLAR:

- [1] Murray, S, Richard, L.S. , " The IBM PC ", Addison - Wesley Publishing Company, 1990.
- [2] The PC Labcards Users Manuel, Advantech Co. Ltd. Taiwan Feb. 1990.
- [3] "Digital Design" ; Prentice / Hall International, Inc. 1984.
- [4] Computer Systems Architecture, M. Morris Mano Prentice-Hall. Inc. New Jersey 1982.
- [5] Kaypro Computers, " GW Basic, User's Guide", Kaypro Computers, 1988.



**Sabri Çelik,**

1961 yılında Kayseri'de doğdu. 1978'de Endüstri Meslek Lisesini, 1979 yılında da Kayseri Lisesini bitirdi. 1983'de Erciyes Üniversitesi

Mühendislik Fakültesinde Lisans, 1983'de Uludağ Üniversitesi Fen Bilimleri Enstitüsünde Yüksek Lisans, 1992 yılında ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

# DAĞITIR KONTROL SİSTEMLERİ İÇİN PRATİK BİR UYGULAMA

E. Burçin Sinanoğlu Kayhan Erciycş

Ege Üniversitesi Bilgisayar Mühendisliđi Bölümü  
35100 Bornova-İZMİR  
obl8950@lrearn, bilker@trcam

## ÖZET

Gerçek-zamanlı dağıtık bir kontrol sistemi için kullanılabilir bir bilgisayar sisteminin, donanımdan bağımsız yazılımı bu çalışmada oluşturulmuştur. Bu yazılım, gerçek-zamanlı dağıtık işletim sisteminin çekirdeđi [2] ve dağıtık sistem süreçleri arasında iletişimi sağlayan iletişim yöneticisinden [1] oluşmaktadır. İletişim yöneticisi High-level Data Link Control (HDLC) protokolünün alt küme işlevlerini gerçekleştirmektedir. Önceki bir çalışmada, ana denetimi olarak çalışan kişisel bilgisayar için bir de kullanıcı arayüzü tasarlanmıştır. [1] Dağıtık sistemin benzeşimi kişisel bilgisayar ortamında yapılarak test ortamı sağlanmıştır.

## 1. Giriş

Yaniletken teknolojisindeki gelişmeler, daha önce bir anabilgisayarı yapabileceđi hız ve hacimdeki işlerin mikroişletimci tabanlı sistemler tarafından gerçekleştirilmesine yol açmıştır. Bu sebepler; günümüzde yönetim ve genel veri-tabanı olarak kullanılan bir ana bilgisayar ve, buna bađlı orta seviyede iş yürütecek ve en alttaki veri toplama/işleme birimleri arasındaki iletişimi sağlayacak birkaç minibilgisayardan oluşan klasik sistemleri yerine iş istasyonları/kişisel bilgisayarların yerel bir iletişim ađı ile birleştirilmesinden oluşan dağıtık sistemler önem kazanmıştır, (bkz. Şekil 1)

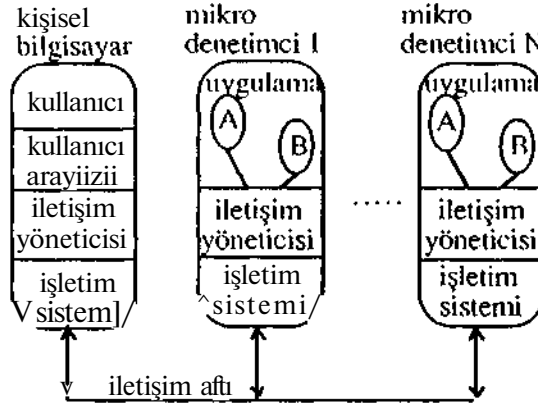
Bahsedilen dağıtık sistemlerin yazılımı, donanımdaki gelişmelere paralel olarak ilerleyememiş ve bilgisayar alanında önemli araştırmalarından birini oluşturmuştur. Dağıtık sistemler yazılımında önemli araştırmalardan biri, tek işlemci üzerinde çalışabilen ve teorisi bir hayli olmuştur olan işletim sistemlerinin, dağıtık işletim sistemi olarak gerçekleştirilmesidir. Dağıtık işletim sistemlerini belirleyen en önemli özellik, tüm sistem kaynaklarını uygulamaya tek bir sistem gibi göstermesidir.

Dağıtık sistemlerin kullanımı günümüzde belirli bir işe adanmış gömülü (embedded) sistemlerde de yaygınlaşmıştır. Fabrikalardaki süreç denetim, nükleer enerji, petro-kimya sistemleri gibi örneklenebilecek bu sistemlerde bir işin doğru yapılması ile birlikte zamanında yapılması da önemlidir. Bir işin zamanında yapılmamasının ise mal ve can kaybına sebep olabileceđi bu sistemler, gerçek-zamanlı sistemler olarak bilinirler.

258

Donanımdaki gelişmeler kişisel bilgisayar ve mikrodenetimcilerin birçok alanda kullanımına olanak vermiştir. Ortalama bir bilgeyc. sahip ufak bir fabrika, iş-akış kontrolünü sağlayabilmek ve verimliliđi arttırabilinck için tanımlanacak sisteme benzer bir tasarımı gerçekleştirebilir, (bkz. Şekil 1)

Bu sistemdeki kişisel bilgisayar, mikrodenetimciler üzerinde çalışan süreçleri, elle monitör ve kontrol etmeye yarayan bir araç olarak düşünülebilir. Kullanıcı istekleri menü yardımıyla belirlenir. Ayrıca herhangi bir anda, mikrodenetimcilerden herhangi biri bir kesinti gönderebilir. Son isteđe ait durum ve elde edilen deđerler kişisel bilgisayar ekranına yansıtılır. Kontrol mekanizması, girişleri klavyeden alarak limit deđerleri üzerinde deđişiklik yapmaktadır.



Şekil 1. Sistem Konfigurasyonu

## 2. Dağıtık Kontrol Sistemi

Kişisel bilgisayar ve mikrodenetimciler, sistemin düzgün çalışabilmesi için bir şekilde iletişime girmelidir. Bu yüzden donanımdan bağımsız bir protokola ihtiyaç vardır. Bunun yanı sıra, iletişim protokolü hata ve akış kontrolünü de sağlamalıdır. HDLC protokolü yukarıda belirtilen özelliklere uygunluđu, kolay kullanımı ve herkes tarafından bilinir olması nedeniyle seçilmiştir.

Tüm mikrodenetimcilerce ait parametreler ve deđerler her zaman ekranda gösterilmektedir. Bu sayede kullanıcı elde edilen deđerlerdeki deđişimleri kolayca gözlereyebilmektedir. Alarm durumu, monitör isteđi olmaksızın ekranda Şekil 2'deki gibi gösterilir.

ELEKTRİK MÜHENDİSLİĐİ 5. ULUSAL KONGRESİ

da Erciyes Üniversitesi Fen bilimleri Enstitüsünde Doktora eğitimini tamamladı. Halen Erciyes Üniversitesi Mühendislik Fakültesinde Öğretim Görevlisi olarak görev yapan Çelik iki çocuk babası olup İngilizce bilmektedir.



**Osman Coşkun,**

14.10.1962 tarihinde Yozgat ili Çayıralan ilçesi Aşağı Tekke Köyünde doğdu. 1979'da İstanbul Boğaziçi (B.K.Ç) Lisesini bitirdikten sonra, 1983'de

Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Bölümünde lisans, 1989'da İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsünde Yüksek lisans ve 1992'de Erciyes Üniversitesi Fen Bilimleri Enstitüsünde Doktora öğrenimini tamamladı. Halen Erciyes Üniversitesi Mühendislik Fakültesi Elektronik Bölümünde öğretim üyesi olarak görev yapmakta olan Coşkun dört çocuk babası olup İngilizce bilmektedir.



**Ferhat Daldaban,**

1959 yılında Kayseri'nin Pınarbaşı ilçesinde doğdu. İlk, orta, lise tahsilini Kayseri'de tamamladı. 1981 yılında K.T.Ü. Elektronik ve Haberleşme

Müh. Bölümünden mezun oldu. 1988 yılında Uludağ Üniversitesi Fen Bilimleri Enstitüsünde yüksek lisans eğitimini bitirdi. 1992 yılında Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektronik Anabilim dalında Doktora eğitimini tamamladı. Halen Erciyes Üniversitesi Mühendislik Fakültesinde öğretim üyesi olarak çalışmakta olan Daldaban 2 çocuk babası olup İngilizce bilmektedir.

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Kullanıcı, birinci öğeyi seçip ekle edilen değerleri izleyebilir veya ikinci öğeyi seçip bir parametrenin alt ve üst sınırlarını değiştirebilir. Bir monitör isteği seçilen mikrodenetimciye gönderildiğinde tüm parametrelere ait son değerler geri döndürülür ve hemen ekrana yazdırılır. Kontrol isteği ise bir parametrenin limitlerini değiştirmek için gönderilir. Bu isteğe karşılık bir cevap gelmez ama değiştirilen limit değerleri ekranda gösterilir. Herşey ekranda tek sayfa halinde listelendiği için kullanıcı sayfalar arasında dolaşmak zorunda kalmamaktadır.

#### MASTERMAINMENU

MICROCONTROLLER NUMBER = 1  
MICROCONTROLLER STATUS = R

	SET	UP	LOW	OBTAINED	
PARAMETERS	UNIT	POINT	LIMIT	LIMIT	VALUE ALRM
1. PRESSURE	mbar	5	6	4	7 HIGH
2. DENSITY	g/cm	5	9	1	4

MICROCONTROLLER NUMBER = 2  
MICROCONTROLLER STATUS = R

	SET	UP	LOW	OBTAINED	
PARAMETERS	UNIT	POINT	LIMIT	LIMIT	VALUE ALRM
1. TEMP.	C	120	152	74	125
2. REVOL.	ipm	1300	1700	1000	1100 LOW

1. Request for MONİTÖR
2. Request for CONTROL
3. EXIT

Şekil 2. Ekran Forman

### 3. Çekirdek Tasarımı

Bir bilgisayar sistemi üzerinde kullanıcı programlarını kontrol ve idare eden programlar topluluğuna işletim sistemi denir. İşletim sistemi donanım ile kullanıcı programları arasında bir arayüz görevi görürken kullanıcı programları arasındaki iletişimi de sağlar. Sistem üzerindeki süreçlerin çalışmasını sağlayan temel işlevler ise işletim sisteminin çekirdeğini meydana getirir.



Şekil 3. Çekirdek Yapısı

Bu çalışmada, dağıtık gerçek-zamanlı çok-zamanlı ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

bir çekirdek (asarımı gerçekleştirilmiş ve bu programın her düğümde kopyalanmasıyla işletimin sağlanması tasarlanmıştır, işletim sistemi, süreç modelini esas alarak ve hiyerarşik bir yapı içerisinde oluşturulmuştur. 12]

### 3.1. Sistem Nesneleri

İşletim sisteminde kullanılanın ve sistemin verimliliğini önemli ölçüde etkileyecek olan veri yapıları şunlardır: süreç kontrol bloğu, semafor, posta kutusu, tampon, aygıt kontrol bloğu ve sistem tablosu. Gerçek-zamanlı uygulamalarda oluşması istenmeyen ek yükü ortadan kaldırmak için lüm bellek işlemlerinin statik olarak gerçekleştirildiği bu sistemde ilk olarak veri yapıları bağlı listeler haline getirilir. Bu işlemin amacı kullanılan ve kullanım dışı öğeleri belirleyebilmektir.

#### Süreç Kontrol Bloğu

Temel eleman olarak süreci ele alan bu sistemde bir sürece ait lüm bilginin tutulduğu veri yapısı süreç kontrol bloğudur. Süreç konumu bloğu, süreci işletim sistemi açısından tanımlı hale getirir. İşletim sistemi bu sayede bir süreçle ilgili olarak işe kaldığı yerden başlayabilme imkanına sahip olur. Bu veri yapısının içinde başlıca şu alanlar vardır:

- sürecin kimliği,
- sürecin konumu,
- sürecin önceliği,
- sürecin bellek alanını belirleyen işaretçiler,
- kullanılan kaynakları belirleyen işaretçiler,
- kayıtçı saklama alanı.

#### Semafor

Bir sistemdeki süreçler izolasyon halinde bulunmazlar. Bir bakıma, işbirliği içinde belirli bir amaca yönelik olarak çalışırlar. Bir başka açıdan ise sınırlı olan kaynakları ele geçirmek için yarışmaktadırlar. Bu iki durum da süreçler arasında hızlı bir iletişimin gerekliliğini ortaya koyar.

Süreçlerarası iletişime en önemli katkı (Dijkstra, 1965) semafor olgusunun ve temel işlevlerinin (P ve V) ortaya alınmasıydı. Gerçekte ilkinin haricinde sadece P ve V operasyonlarına tabi tutulabilen pozitif bir tamsayıdan ibaret olan semafor yapısı bir kilit işlevi görür. Sistem kaynaklarının kontrolünü ve korunmasını sağlar.

#### Posta Kutusu

Posta kutuları süreç ve mesajlar için bir buluşma noktası özelliğini taşırlar. Bu veri yapısı, üzerinde mesajların veya süreçlerin beklemesine olanak sağlamalıdır.

#### Tampon

Ana bellekle genellikle süreçlerarası iletişim esnasında veri tutmak için kullanılan sahalar tampon denir. Bellek yönetimi havuz ve tampon yapılarıyla sağlanır. Bir havuz içerisinde kullanıcı tarafından

belirlenen u/unlukla ve sayıda tampon saha bulunur. Her havuz bir semafor tarafından korunur. İlgili semafor havuzdaki tampon sayısı ile ilklendir. Böylece, havuzda boş tampon saha kalmadığı zaman istekle bulunan sürecin bekletilmesi mümkün olur ve havuza bir tampon saha döndüğünde bekleyen ilk sürece verilir.

### Aygıt Kontrol tiloğu

Sistemde; her aygıt için oluşturulan bu veri yapısı; aygıt özellikleri, aygıt sürücüsünün bekleyeceği ve sinyallycccği semaforlar, alınan/gönderilen bilgilerin tampon saha adresleri gibi bilgileri içermekle ve kullanıcı ile aygıt arasındaki veri alışverişi ve senkronizasyon için kullanılmaktadır.

### Sistem Tablosu

Sistem tablosu, işletim sistemi için referans teşkil edebilecek tüm verileri bünyesinde loplar. Sistem ilk yüklendiğinde oluşturulan kullanım dışı veri yapı listelerinin başlangıç ve biliş adresleri sistem tablosuna kaydedilir.

### Algoritmalar

Yukarıda belirtilen veri yapılarından posta kutusu, tampon ile semafor kuyruklar ve süreç kontrol blokları bağlı listeler halinde tuludur. Sistemde her veri yapısı için kullanılan ve kullanılmayan öğeleri kılmak üzere iki farklı üsle oluşturulur. Kuyruklar için klasik ekleme ve çıkarma algoritmaları (enq/cuc ve dcq/cuc) kullanılmaktadır. Yani, eklemeler sona yapılırken çıkarmalar baş taraftan gerçekleştirilmekle böylece ilk gelen ilk çıkar mantığı işletilmektedir. Fakat süreçlerin öncelik ve bekleme zamanı gibi kriterlere göre sıralanmasını gerektiren uygulamalar için farklı algoritmalar geliştirilmiştir. Sıralı ekleme işlemleri sayesinde üslenin ilk elemanın sıradaki elemanı olması sağlanmıştır. Öncelikleri dikkate alarak oluşundan bir listenin ilk elemanı en yüksek önceliğe sahip olacağından, önceliğin belirleyici olduğu iş-dağılımı algoritmalarının işletimi için listedeki ilk elemanı almak yeterli olacaktır. Ayrıca, bekleme zamanları kümülatif olarak tutulduğundan her zaman günlemesi gerektiğinde Kim süreçler için değil sadece listedeki ilk süreç için işlem yapılacaktır.

### 3.2. Zamanlayıcı

Zaman'ayıcı, bilgisayar sisteminde merkezi işlem biriminin takvimini belirler. Başka bir deyişle, sistem kaynaklarının en önemlisinin en iyi ve adil şekilde paylaşımını sağlar. Ayrıca üretilen işi maksimize etmeyi, ck-yükü minimuma indirmeyi, kaynak kullanımını dengelemeyi, sonsuz ertelemeyi önlemeyi ve öncelikleri kullanmayı amaçlar.

Zamanlayıcı, çalışmakta olan süreç devam edemediği veya merkezi işlemci biriminin daha iyi kullanılması olasığında devreye girer. Bu durumlar detaylı olarak aşağıdaki gibi belirtilebilir:

- bir sürecin statüsünü değiştiren bir kesintiden sonra,
- çalışmakta olan sürecin bir bekleme zamanı boyunca devam edememesi üzerine,

-bir semafor üzrciinde gerçekleştirilen ve başarısızlıkla sonuçlanan bir P işleminden sonra sürecin bloke olması sebebiyle.

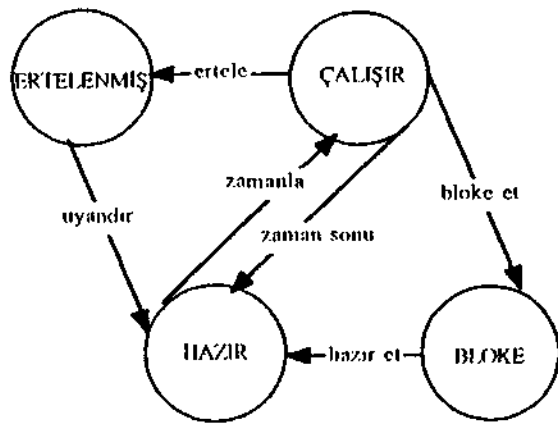
Sistemdeki tüm süreçler, işlev ve statiiicli indeki farklılıklardan dolayı üç ayrı kategoride toplanmıştır: sistem, gerçek-zamanlı, geri-plan. Çünkü süreçler arasındaki farklar değişik zamanlama algoritmalarının kullanımını gerektirmektedir. Uygulamada bu olay süreç kontrol bloklarının farklı kriterlere göre sıralanmasıyla gerçekleştirilir.

İler süreç bir kez başlayıp sonlanana kadar çalışamayabilir. O anda kullanıma açık olmayan bir kaynağı beklemek zorunda olduğu için veya bir mesajın gelmesini beklediği için merkezi işlemciyi kaybedebilir. Çalışması kesilen veya durdurulan bir sürecin işine kaldığı yerden devam edebilmesi için o anki durumu ile ilgili tüm verilerin saklanması gerekir. Sürece tekrar merkezi işlemci zamanı tanındığında saklanan veriler geri yüklenir ve işleme kalınan yerden devam edilir.

Kesintiler, dış dünya ile bilgisayar arasındaki iletişimin temelini oluştururlar. Gerçek-zamanlı sistemlerde genellikle dış dünyadan kesintiler yoluyla alınan veriler işlenir. Bu sebeple, bir süreç iki giriş noktası vardır. Kesinti seviyesinde veriler alınır, normal seviyede işlenirler. Bir süreç iki seviyenin birinde uyanık olabilir.

### 3.3. Süreç Konum Yönetimi

Sistemdeki varlığı süresince bir süreç farklı konumlarda bulunur. Birçok değişik olay bir sürecin konum değiştirmesine neden olabilir. Çalışır durumda olan bir süreç çalışma zamanı dolduğu için ya da ihtiyacı olan kaynakları ele geçiremediği için, beklemekle olan bir süreç, bekleme zamanı dolduğu için konum değiştirebilir. İler şekilde durum değişikliği işletim sisteminin tarafından kaydedilmeli ve gerekli düzenlemeler yapılmalıdır. Örneğin, bloke olmuş bir süreç bu durumdan kurtulduğunda hazır listesine eklenmeli ve zamanlayıcı bu sürecin çalışma önceliğini kontrol etmelidir.



Şekil 4. Süreç Konum Şeması

Bir süreç eğer merkezi işlem biriminin kullanım ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

hakkına sahipse o süreç ÇALIŞIR konuludur. Eğer bir süreç, mümkün olsa incike/i işlem birimini kullanabilecek durumda ise HAZIR konumdadır. Eğer süreç, bir semafor kuyruğunda sistem kaynaklarından biri için bekliyorsa RI.ÖKE konumdadır. Son olarak, eğer süreç belli bir zaman süresi boyunca beklemek zorunda ise ERTELENMİŞ konumdadır.

Süreç konumları arasındaki geçişleri sağlayan Kını yordamların ortak özelliği gerçekleşen konum değişikliğini ilgili süreç kontrol bloğuna kaydetmeleridir. Zamanla yordamı, HAZIR süreçlerin bulunduğu listeleri belirlenen sıraya göre tarar ve bulunan ilk sürece ait bilgileri sisteme yükledikten sonra kontrolü bu sürece bırakır.

Zaman sonu yordamı, saat kesinti yordamı tarafından aklı ve edilir. Bir süreç için belirlenen çalışma zamanı sonlandığında arık merkezi işlem birimi başka bir sürecin kullanımına verilmelidir. Çok-görevli bir işletim sisteminin kullanıldığı bir uygulamada diğer süreçler beklerken bir süreç sonsuza kadar çalışamaz. Sürece bir seferde ayrılan süre sonlandığında HAZIR kuyruğundaki yeni bir süreç zamanlayıcı tarafından ÇALIŞIR konuma getirilir. Her süreç için farklı bir değer alabilecek olan çalışma süresi yaratma aşamasında belirlenir ve süreç kontrol bloğuna kaydedilir.

Bir semafor üzerinde başarısız bir P operasyonu gerçekleştiğinde işlemi başlatan süreç istediğini elde edememiş demektir ve BLOKE edilerek semafor üzerinde "I" iletilmesi (kuyruklanması) gerekir. BLOKE edilmeden yapılabilecek P işlemi sayısını belirleyen semafor değeri, bu durumda sıfırdır.

Hazır et yordamı, süreç kontrol blok adresi verilen herhangi bir süreci hazır duruma getirebilir. Yordam, sürecin ait olduğu grubu belirleyerek uygun algoritmanın yardımıyla hazır listesine yerleştirir. Hazır el yordamı bir sürecin ilk varoluşu aşamasında da kullanılır. Süreç kontrol bloğuna gerekli bilgiler kaydedildikten sonra yapı ilgili grubun HAZIR süreçler listesine eklenir. Uyandırma işleminde ise ilgili süreç ERTELENMİŞ süreçlerin tutulduğu listeden çıkarılır ve daha sonra hazır el yordamı vasıtasıyla HAZIR konuma getirilir.

Çalışmakta olan süreç kendini belirli bir süre için uyulmak isteyebilir. Relidendi bekleme zamanını kriler alınarak süreç kuyruklanır. Sistemin saat kesinti yordamı, zaman geldiğinde bir kesinti ile durumu işletim sistemine bildirecektir.

#### *Saat kesimi yordamı*

Zamanla ilgili belirlemeleri yapabilmek için sistemde bir saat yordamı devamlı olarak çalışmalıdır. Bu yordam her saat sinyalinde devreye girerek gerekli kontrolleri ve işlemleri gerçekleştirmelidir. Yordamın başlangıç adresi kesinti vektör tablosunda istenen kesiniye karşılık gelen sahaya yerleştirilmiştir. Bir kesinti geldiğinde, yani bir birim zaman geçtiğinde, çalışmakta olan sürecin kalan çalışma süresi ve bekleme listesinin başındaki sürecin kalan bekleme zamanı bir birim azaltılır. Azaltma işleminden sonra bu değerlerin

**ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ**

sıfır olup olmadığı kontrol edilir. Rğcı ilk değer sıfır ise zaman sonu yordamı, ikinci değer sıfır ise uyandır yordamı aktif edilir.

#### 3.4. Siireçle'aitisi İletişim ve Senkroni/aşYon

Bilgisayar sistemi içerisinde varolan süreçler izolasyon içerisinde hareket etmezler. Bir bakıma ortak bir amacı gerçekleştirmek, kullanıcı işlerini çalıştırmak, için yardımlaşır. Ama aynı zamanda merkezi işlem birimi, bellek, giriş/çıkış aygıtları gibi sınırlı sistem kaynakları için yarışmaktadırlar. Yardımlaşma ve yarışma kavramları, bir iletişimin gerekliliğini ortaya koymaktadır. Karşılıklı dışlama, kilitleme ve senkronizasyon; iletişimin önem kazandığı alanlardır.

Tasarlanan sistemde, süreçler arasında senkronizasyonu sağlamak için semafor yapıları ve iletişimi gerçekleştirmek için posla kutuları kullanılmıştır.

#### *Semafor Mekanizması*

Gerçek-zamanlı sistemlerde, A ve B gibi iki süreci senkronize etmek için üzerinde sadece P ve V işlemlerinin yapılabildiği bir değişken kullanılır. Bu iki işlem sayesinde süreçlerin belirli olayların oluşumunu beklemesi sağlanabilir. Böylece, süreçler arasında göreceli bir senkronizasyon sağlanmış olur.

Semafor değeri sıfır ise P işlemi ilgili sürecin bloke edilmesine neden olur. Eğer semafor sıfırdan farklı bir değere sahipse bu işlem sonucunda istek yerine getirilebilecek demektir. V işlemi ise başarısız bir P işlemi sonucunda kuyruklanmış süreçlerden ilkinin isteğinin yerine getirilmesine olanak verir. Eğer kuyruk boş ise semafor değeri bir artırılır. Anlaşıldığı gibi, her semafora bağlı bir liste vardır. Bu listelerde bloke edilmiş süreçler uyandırılma kadar tutulur.

#### *Posla Kutusu Mekanizması*

Süreçlerin veri alışverişinde bulunabilmesi için aralarında bir iletişim mekanizması olması gerekir. Doğrudan isimlendirme için gönderenin ve alıcının isimlerinin mesaj içerisinde bulunması gerekir. Dolaylı iletişim yöntemlerinde ise bağlantı noktaları ve posla kutuları kullanılır. Tasarlanan sistemde ikinci yöntem kullanılmıştır.

Çalışan süreç, herhangi bir posta kutusuna bir mesaj gönderdiğinde posla kutusunda bekleyen ilk süreç bu mesajı alacaktır. Eğer mesaj gönderilen posta kutusunda mesaj bekleyen süreç yoksa o zaman mesaj kuyruklanacaktır. Çalışan süreç, herhangi bir posta kutusundan bir mesaj almak istediğinde kuyruktaki ilk mesajı alacaktır. Eğer mesaj yoksa kuyruklanarak bir mesaj gelmesini bekleyecektir. Yani bir posta kutusunda süreçler de mesajlar da bekleyebilecektir. Ama aynı anda hem süreçlerin hem mesajların beklemesi kullanılan muhtemelen mümkün değildir.

#### 3.5. Bellek Yönetimi

Tasarlanan işletim sisteminde, dinamik bellek

yönetiminin beraberinde getireceği ek yükü ortadan kaldırmak amacıyla statik yerleşim uygulanmıştır. Bu nedenle, sistemin ilklcnmcsi aşamasında gerekli tüm bellek alanları ayırdır ve ilerideki kullanımı sağlamak için bağlı listeler oluşturulur. İşletim sisteminde kullanılan veri yapılarının hepsi aynı şekilde ilklcnir ve listelerin ilk ve son elemanlarının adresleri sislem tablosuna kaydedilir.

Değişik sayı ve büyüklükteki tampon sahalarını tutmak için tampon havuzları oluşturulmuştur. Bir tampon havuzunun içindeki tampon sahalarının büyüklüğü sabit tutulmuştur. Bunun dışındaki tüm kriterler kullanıcıya bırakılmıştır. Kullanıcının kullanımına açılan bellek sahası içinde kaç adet, hangi büyüklükte tampon havuzu oluşlurulacağı parcmclıc geçişleri ile belirlenir. Uygulamanın özelliklerine göre tek bir bütün olarak kullanım sağlanabileceği gibi bellek sahasını mantıksal parçalara ayırmak ta mümkündür.

### 3.6. (İriş/Çıkış Yönetimi

Herhangibir işletim sisteminin yerine getirmesi gereken işlevlerden biri de giriş/çıkış aygıtlarının yönelimidir. Bilgisayar için tasarlanmış disk, teyp gibi birçok giriş/çıkış ünitesi ile bilgisayar arasındaki iletişim seri veya paralel, senkron veya asenkron olarak gerçekleşebilir. Bu tip mekanik giriş/çıkış aygıtlarını genellikle kontrolcü adı verilen tümleşik devreler kontrol eder. İşleyici ile kontrolcü arasındaki en basit iletişim, kontrolcünün yazmaçlarına işleyicinin bilgi yazması yoluyla bilgi transferini sağlamaktır. Bir giriş/çıkış işleminin bitip bilmediğini anlamının bir yolu işleyicinin aygıtı periyodik veya sürekli olarak kontrol etmesidir. Diğer bir yöntem ise aygıtın işlem bittiğinde kesinti göndermesidir.

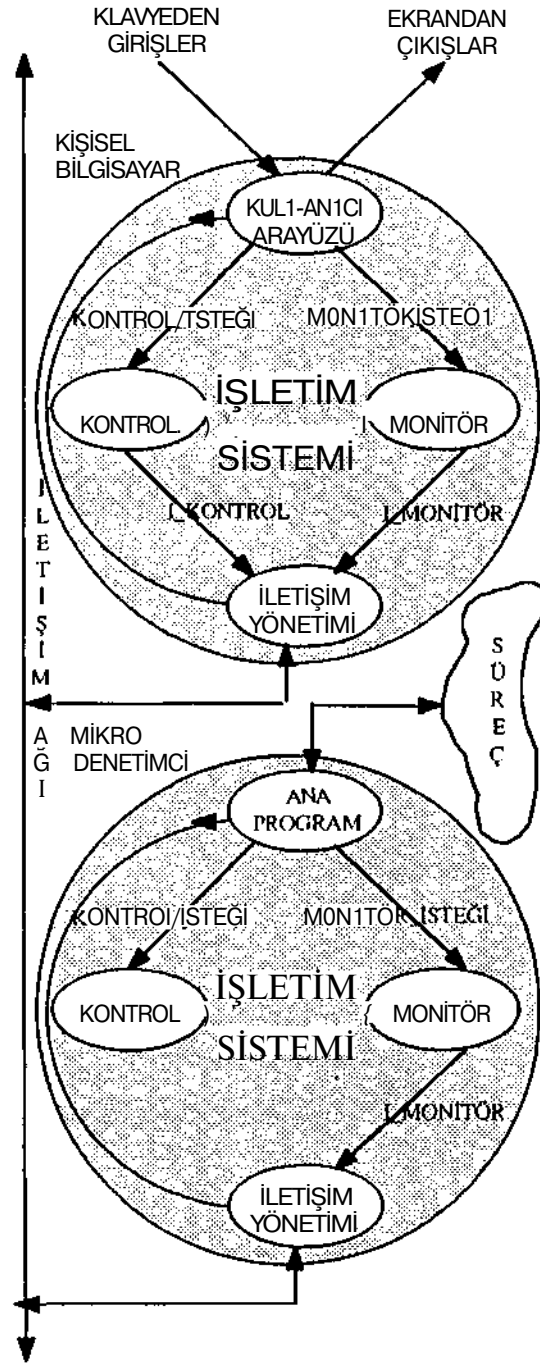
Tasarlanan giriş/çıkış mekanizmasında, aygıt sürücülerini ile iletişim, aygıt kontrol bloğu yoluyla gerçekleşir. Aygıt sürücüsü, örneğin disk sürücüsü, aygıt kontrol bloğu semaforunda sürekli beklemededir. Genel bir giriş/çıkış kullanıcı yordamı, giriş/çıkış istek bloğu adlı veri yapısına gerekli bilgileri koyup aygıt kontrol bloğundaki semaforu sinyaller. Bundan sonraki alt seviye giriş/çıkış işlevleri bu semafora beklemekte olan aygıt sürücüsüne kalmıştır.

### 4. Sonuçlar

Bu çalışmada, dağıtık işlem kontrol sistemi uygulaması kişisel bilgisayar ve mikrodenetimcilerin tümleşimi ile gerçekleştirilmiştir. Küçük bir fabrikanın üretim hattı için ucuz ve çok yönlü olarak düşünülen bu tip sistemlerin kullanımının ve kurulumunun az bir gayret gerektireceği düşünülmektedir.

Çalışma mikrodncnlimci için dağıtık gerçek-zamanlı çekirdek tasarımını da içermektedir. Bu çekirdek; süreç, giriş/çıkış ve kesinti yönetimi gibi alt düzey sistem işlevlerini gerçekleştirmektedir. İşletim sistemi ve iletişim ortamı net olarak tanımlanmış bir sislem kullanıcıya leslim edilebilecektir. Bu sistemin üstüne her değişik uygulama, kullanıcı tarafından, uygulama süreçleri olarak işletim sistemine arayüz

sağlayacaktır. Dolayısıyla her düğümde bir kopyası çalışan bir işletim sistemi, iletişim yöneticisi ve sistem programları bulunacaktır. Bu tasarım homojen, bakımı kolay yapılabilen ve hatalardan arınmış bir sistemin gerçekleşmesini sağlayacaktır ve birbirinden çok farklı uygulamalar için kullanılabilir.



Sekil 5. Sistemin Mantıksal Görüntüsü  
ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Kayıtlık hır

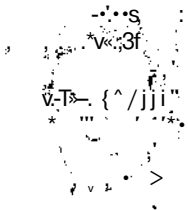
1. Tuğlular, T, Yıldırım, Ş, Sinanoğlu, E. B., Freiyçş, K, Çankaya, H. C, "Kişisel Bilgisayar Tabanlı Bir Dağılık Kontrol Sistemi", Kgc Ünivcrsilcsi Bilgisayar Araştırma ve Uygulama Merkc/i Dergisi, ("ili 14, Sayı 119"1 .

2. Sinanoğlu, F. B., "A Real-time Müllilasking Dislribüitel Operating System Kernel", Fge Üniversitesi Mühendislik Fakültesi Bilgisayar Müllindisliğı Bölümü Bilirine Tezi, 1993.

#### Ya/nılara İlişkin Bilgi



SİNANOGLU, F. Burçin, 1971, Mersin doğumlu. İlk ve orta öğrenimini İzmir'de tamamladı. 1989-93 döneminde Fge Üniversitesi Bilgisayar Mühendisliğı Bölümündeki Lisans eğilimi yapıldı. İşletim sistemleri teorisi ve gciçck-/anlanlı süreç kontrol sistemleri ilgi alanları içindedir.



F. RCİYŞ, Kayhan, 1956, İzmir doğumlu. 1979'da Manchesler Üniversitesinden Lisans, 1983'ic Salford Üniversitesinde Yüksek Lisans ve 1989'ıla Fge Üniversitesi 'nden doktora aldı. İTİ' Araştırma Laboratuvarında kanal transfer teçizati, Tc.c.laş İlmiasında modem ve Alcacl Portekiz, firmasında ISUN üzerine çalıştı. Halen yardımcı doçent olarak Fge

Üniversitesi'ndc görevlidir. Dağılık işletim sistemleri, paralel bilgi işleme, gciçck-zamanlı sistemler, bilgisayar ağları ilgi alanları içindedir.



# Bilgisayar Ağlarının Topolojik Tasarımında Değişik Teknikler

Cem F.rsoy

Bilgisayar Mühendisliği Bölümü  
Boğaziçi Üniversitesi, 80815 Bebek, İstanbul

## Özet

Bilgisayar ağlarının topolojik tasarımı zor bir katımsal eniyileme problemidir. Çözümü kolaylaştırmak için, bu problem daha basit alt-problemlere bölünür. Alt-problemlerin çözümünde kullanılan klasik yöntemler, doğrusal programlama ve dinamik programlamadır. Fakat ağdaki düğüm sayısı arttıkça, kabul edilebilir bir zamanda eniyi çözümün bulunması imkansız olmakla ve buluşsal (heuristic) yöntemlerin kullanılması yeğlenmektedir. Bu tip problemlerde, rassal-buluşsal yöntemlerin kullanılması çok iyi sonuçlar vermiştir. Kullanılan rassal buluşsal yöntemlerden bazıları, tavlama benzetimi (simulated annealing), genetik algoritmalar ve rassal sinir ağlarıdır. Bu bildiride bilgisayar ağlarının topolojik tasarımı ile ilgili genel problemler ve yerel alan ağlarının birbirlerine bağlanmalarında ortaya çıkan bazı özel problemler tanıtılacak ve yukarıda değinilen çözüm yolları karşılaştırılacaktır.

## 1 Giriş

Son yıllarda, bilgisayarların işyerlerinin ayrılmaz bir parçası haline gelmesi ile birlikte, yerel alan ağlarının (LAN - Local Area Network) kullanımı hızla yayılmıştır. Günümüzde, orta büyüklükte işyerlerinde bile birkaç LAN bulunması olağan sayılmaktadır. Bu işyerleri arasındaki coğrafi uzaklıklar ve yerel ağların kapasitelerindeki sınırlamalar, bu ağların birbirlerine metropoliten ve geniş alan ağları (MAN-WAN) üzerinden bağlanmasını gerektirmektedir. LAN-MAN ağlarının tasarımında değişik amaç işlevleri ve kısıtlar kullanılabilir: İletişim gecikmesi, maliyet, güvenilirlik, ve veri güvenliği bunların en önemlileridir.

Bu ağların topolojik tasarımı zor bir katımsal eniyileme problemidir. Çözümü kolaylaştırmak için, bu problem daha basit alt-problemlere bölünür. Alt-problemlerin çözümünde kullanılan

klasik yöntemler, doğrusal programlama ve dinamik programlamadır. Fakat ağdaki düğüm sayısı arttıkça, kabul edilebilir bir zamanda eniyi çözümün bulunması olanaksız olmakta ve buluşsal yöntemlerin kullanılması yeğlenmektedir [1]. Bu tip problemlerde, rassal-buluşsal yöntemlerin kullanılması çok iyi sonuçlar vermiştir [2]. [3]. Kullanılan rassal buluşsal yöntemlerden bazıları, tavlama benzetimi [4], genetik algoritmalar [5] ve rassal sinir ağlarıdır [6].

Bildirinin 2. Bölümü'nde bilgisayar ağlarının topolojik tasarımı ile ilgili genel ve özel problemler tanıtılmış, 3. Bölümü'nde rassal buluşsal teknikleri de içeren değişik çözüm yolları karşılaştırılmış, sonuçlar ise Bölüm 4'de verilmiştir.

## 2 Ağların Topolojik Tasarımı

Bilgisayar haberleşmesinde, ağların topolojik tasarımı problemleri sık sık karşımıza çıkar. Bu problemler birbirleriyle belli miktarlarda veri iletişimde bulunmak isteyen uç kullanıcıların (düğüm) belirli işlerler sağlanacak şekilde birbirlerine bağlanması için verilecek kararlar dizisinin bulunmasıdır. Genellikle, bu kararlar sonucunda belirli bir kriterin eniyilenmesi ve belirli sınırlamaların içinde kalınması beklenir [1], [2], [3], [4], [5], [6], [7], [8], [9]. Bu kriter toplam maliyetin ucuzluğu olabileceği gibi, ortalama paket gecikmesi veya toplam kapasite gibi ağın verimi ile ilgili de olabilir. Sınırlamalar ise, iki düğüm arasında en çok izin verilen paket gecikmesi, bağlantıların kapasiteleri veya ağın güvenilirliği ile ilgili olabilir. Verilecek kararlar hangi düğümlerin direk olarak bağlanacağı, bu bağlantıların kapasiteleri, veri paketlerinin hangi yolları seçeceği (yol-alama) gibi değişik tiplerde olup birbirleri ile etkileşim içindedir. Bu nedenle, ağ tasarımı problemleri karmaşık ve zordur. Karmaşıklıklarını azaltmak için, birbirlerinden oldukça bağımsız alt problemlere bölünmeye çalışılarak, kademeli bir yaklaşım izlenir [1], [9].

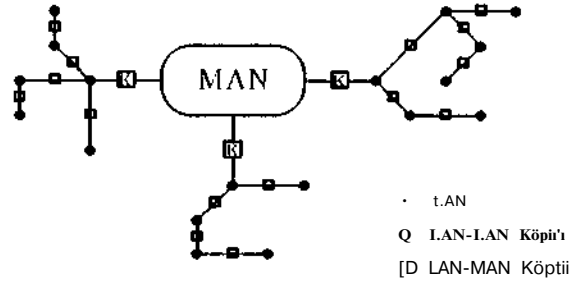
ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Genellikle, bu alt problemler dahi çok zor olup, çok hesap yoğun çözüm teknikleri gerektirebilirler. Bu teknikler Bölüm 3'de sıralanmıştır. Bu bölümün devamında, rassal buluşal tekniklerin başarılı olduğu iki özel topoloji tasarımı problemi tanıtılacaktır.

## 2.1 LAN-MAN Ağlarının Tasarımı

Devamlı artan yerel ağ kullanıcı sayıları bu ağların kapasite ve coğrafi sınırlamalarını aşabilmek için birbirlerine bağlanmaları gereksinimini doğurmuştur. Değişik bağlantı yolları arasında, köprüler sadelikleri ve hızları nedeniyle büyük bir ilgi uyandırmışlardır [10]. Köprüler Açık Sistemler Arabağlaşımı (OSI) modelinin en alt iki katmanında (fiziksel ve bağ) çalışarak, hem coğrafi hem de kapasite sınırlamalarını aşmak için kullanılan en basit geçit birimleridir. Köprüler veri paketlerini varış adreslerine göre süzerek, trafik yoğunluğunun gereksiz yere her yerde artmasını engelledikleri için, ağın verimini artırır ve ortalama paket gecikmesini azaltırlar. Görevlerinin sınırlı olması nedeniyle hem karmaşıklık, hem de fiyat bakımından görece ucuz ve hızlıdır. Değişik yerel ağları birbirine bağlamak için değişik tipte köprüler kullanılır. IEEE 802 standartlar grubu saydam köprüleri standart haline getirmiştir. Bu köprüler CSMA-CD tipi yerel ağları birbirine bağlamak için yaygın olarak kullanılmaktadır. Saydam köprülerin en büyük avantajı, yerel ağları bağlamak için kullanıcıların katkısına gereksinim duymamalarıdır. Yani, kullanıcılar kendi ağlarındaki mi yoksa başka bir ağdaki mi bir kullanıcı ile konuştuklarını fark etmemektedirler. Saydam köprüler iki kullanıcı arasında birden fazla yolun aynı anda bulunmasına izin vermedikleri için, yerel ağların bu tip köprülerle bağlanmasından oluşan genişletilmiş ağların topolojilerinin tüm ağı geren bir ağaç olması gerekmektedir. Tüm ağı geren ve islenen özelliklerdeki ağacı bulmak zor olduğu için, bu problem de birbirinden oldukça bağımsız alt-problemlere bölünerek, daha küçük geren ağaçların birleştirilmesi problemine dönüştürülmektedir [2]. Şekil 1 birbirlerine bir metropoliten alan ağı üzerinden bağlanmış birçok yerel ağdan oluşan genişletilmiş bir ağı göstermektedir. Bu örnekle yerel ağlar üç alt gruba ayrılmıştır. Gerçek bir uygulamada, bu gruplar değişik yerlerdeki binalara veya değişik alt çalışma gruplarına karşılık gelebilir. Buna en iyi örneklerden biri, birden fazla kampüsü olan ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

üniversitelerde, aynı kampüsteki yerel ağları birbirlerine bağladıktan sonra, kampüsleri bir MAN üzerinden bağlamaktır.



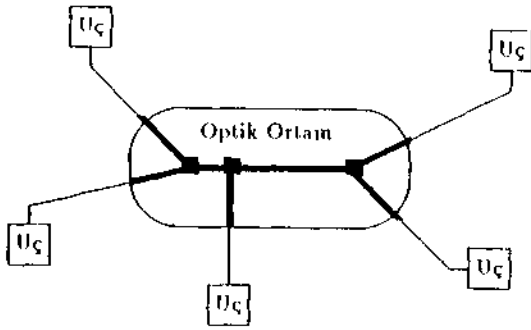
Şekil 1: Örnek bir LAN-MAN ağı.

Varolan çözüm tekniklerinin bu tip topolojik tasarım problemlerine uygulanabilmesi için, önce bilgisayar ağının modellenmesi gerekir. Bu problemde, yerel ağlar bir çizgenin düğümleri, köprüler ise bağlar olarak modellenir. Daha sonra, geren ağaç seçimi bir katınsal eniyileme problemi olarak tanımlanabilir. Bu problemin modellenmesi ve formülasyonu [2]'de detaylı olarak anlatılmıştır. Geren ağaç seçiminde kullanılacak amaç işlevi ortalama paket gecikmesi olduğunda, bağ ve düğümlerin gecikme modellerinin de seçilmesi gerekir. En az ortalama paket gecikmeli geren ağın seçiminde verilecek kararlar dizisi hangi yerel ağların birbirlerine direkt olarak bağlanmasını belirleyecektir. Bulunacak ağaç üzerindeki akışların düğüm ve bağların kapasitelerini aşmama kısıtı problemi zorlaştırmakta ve geleneksel çözüm teknikleri yetersiz kalmaktadır. Bu nedenle, problemin çözümü için, tavlama benzetimi [2], rassal sinir ağları, genetik algoritmalar gibi tekniklerden yararlanılmaktadır. Bu teknikler 3. Bölüm'de tanıtılacaktır.

## 2.2 Çok-hoplmalı Işıksal Ağların Tasarımı

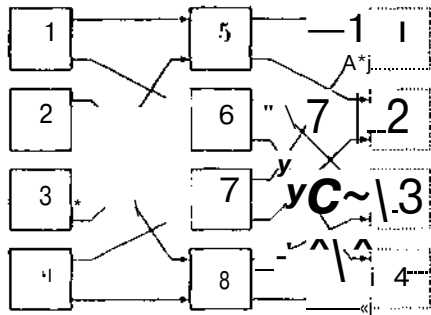
Optik liflerin çok büyük bant genişliği potansiyeli olması ışıksal haberleşme tekniklerine olan ilgiyi çok artırdı. Yüksek hızlara elvermesi nedeniyle, bu tekniklerin en ümit verici olanlarından biri de, dalgaboyu bölüşümlü çoğullamaya dayanan çok-hoplmalı ışıksal ağlardır [11]. Bu ağ mimarisinde, bütün haberleşme uçları. Şekil 2'de görüldüğü gibi, aynı optik ortama bağlıdır. Bu ortam yol, ağaç, yıldız şeklinde değişik topolojilerde olabilir. Tüm uçlar aynı ortama bağlı olmasına rağmen, dalgaboyu bölüşümlü çoğulların aynı anda birden fazla kanaldan

haberleşmeye izin verir. Her uç az sayıda gönderici ve alıcıya sahiptir. İki uç arasındaki her bir katınla bir dalgaboyu ayrılmıştır. Gönderici ve alıcı sayılarındaki sınırlamadan dolayı, tüm uçlar arasında direk bir kanal yoktur. Bu nedenle, veri paketleri gidecekleri uçlara bir dizi ara düğümden hoplayarak varırlar. Bu mimari, yol-atama açısından klasik sakla-ve-ilet lipi paket analılarlamalı ağlara benzemektedir. Değişik dalgaboylarının değişik uç çiftleri arasındaki haberleşmeye ayrılması mantıksal bir bağlantı şeması belirler. Bu mantıksal topoloji ağın fiziksel topolojisinden bağımsızdır ve başka bir dizi dalgaboyu ataması ile değiştirilebilir.



Şekil 2: Işıksal ağlar için, bir fiziksel topoloji.

İlk olarak önerilen mantıksal topoloji, ShuffINet [11], Şekil 3'de gösterilmiştir. Bu şekildeki her bir tek yönlü bağ değişik bir dalgaboyuna karşılık gelmektedir. Eisenberg ve Mchravari, düzgün olmayan trafik isterleri olduğunda, bu mantıksal topolojinin beceriminin düştüğünü gösterdiler [12]. Bunun üzerine değişik mantıksal topolojiler ve tasarım teknikleri önerildi. Örneğin, Sivarajan ve Ramasvami De Bruijn çizgelerini mantıksal topoloji olarak kullanarak çok daha fazla sayıda ucun birbirlerine bağlanabileceğini gösterdiler [13].



Şekil 3: ShuffINet mantıksal bağlantı şeması.

Mantıksal topolojileri cazipleştiren nedenlerden biride yavaşça sıklık uyumlaması yapılabilen lazer ve optik alıcıların son zamanlarda ulaşılabilir hale gelmesidir. Bu bileşenlerin varlığı mantıksal topolojilerin değişen trafik isterlerine veya aksama durumlarına uyarlanabilir olmasını sağlamıştır. Fakat, bu bileşenlerin şimdilik yavaş olması ani uyarlamalara izin vermemektedir. Labourdette ve Acampora bu tip bileşenler kullanarak tekrar düzenlenebilir mantıksal topoloji tasarımı üzerine bir problem tanımlamışlardır [M]. Bu problem bir mantıksal bağlantı şemasının bulunması (dalgaboyu ataması) ve yol-alama kararlarının verilmesinden oluşmaktadır. Amaç kullanım oranı en yüksek olan bağın bu oranını enküçültmeye çalışmaktır. Bu ekip problemi dalgaboyu atama ve yol-alama olarak iki alt-probleme ayırdıktan sonra doğrusal programlama ve buluşsal teknikler kullanarak çözmeye çalışmışlardır. Önce eniyi mantıksal bağlantı şemasını bulmaya çalışmış, sonra da bu bulunan topoloji üzerinde en iyi yol-atamayı başarmaya çalışmışlardır. Biz aynı probleme tavlama benzetimi yoluyla çözümler aradık ve bahsi geçen iki aşamalı teknikle eş iyilikte veya daha iyi çözümler bulduk [15].

Daha önce de belirtildiği gibi, bu problemde amaç en büyük bağ kullanım oranını enküçültmektir. Bu amaca ulaşmak için verilecek kararlar dizisi hangi uçların direk bağlanacağı ve veri paketlerine hangi yolların atanacağından oluşmaktadır. En iyi karar dizisi problem kısıtlarına uygun olarak bulunmalıdır. Bu problem kısıtları ise: yine bağ ve uç akış kapasiteleri ve her bir uçla bulunan toplam gönderici ve alıcı sayılarındadır. Ayrıca yol-atama konusunda da kısıtlar olabilir. Örneğin, aynı uç çiftleri arasındaki veri paketlerinin hep aynı veya değişik yollar kullanabilmesi gibi. Biz en kısa yol ve en iyi paylaşılabilir akışlar yöntemleri için iki ayrı çözüm grubu oluşturduk. En kısa yollar sanal devre tipi uygulamalara elverişlidir. Çünkü, aynı gönderici alıcı arasındaki veri paketleri aynı ve yolu izlerler. Ayrıca, en kısa yolların hesaplanması daha kolaydır ancak eniyilenmemişlerdir. Her ne kadar daha hesap yoğun olsalarda eniyi paylaşılabilir akışlar başarımlı artırırlar. Biz bu tip akışların eniyilerini bulabilmek için Akış Saptırması (Flow Deviation) [16] tekniğini kullandık. Bu problemin formülasyonu oldukça zor bir karışık-tamsayı eniyileme problemi ortaya çıkarmaktadır [15].

Tamsayı kararlar mantıksal bağlantıları, tamsayı ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

olmayanlar ise akış miktarlarını belirlemektedir. Bu kararlar birbirlerini etkilemekte ve amaç işlevini eniyileycen topoloji ve yolların bulunması çok zorlaşmaktadır. Çözüm uzayı boyutunun uç sayısına üstel olarak bağlı olması nedeniyle tüm olasılıkların kabul edilebilir bir zamanda denenmesi olanaksızlaşmaktadır. Problemin çözümü için, doğrusal ve dinamik programlama gibi geleneksel eniyileme tekniklerinin çok uzun süreler gerektirmesi nedeniyle, 'eniye' yakın sonuçlar veren ve 3. Bölüm'de tanıtılan buluşsal tekniklerden yararlanılmaktadır.

### 3 Çözüm Teknikleri

Topolojik tasarım problemlerinin hemen hemen tümünün çözümü çok zordur. Bu problemlere eniyi çözümlerin polinomsal bir karmaşıklıkta olan algoritmalarla bulunması olası değildir [17]. Geleneksel matematiksel programlama teknikleri [17] problemin bazı özel hallerine kesin sonuçlar bulmak için kullanılabilir. Bu tekniklerin en yaygını doğrusal programlamadır. Her ne kadar güçlü olsa da amaç işlevlerinin sürekli olmaması bu tekniğin uygulanabilirliğini azaltmaktadır.

Pekçok tasarım değişkeninin lambayı olması nedeniyle, problem tamsayı programlama teknikleri gerektirmektedir [17]. Bu tekniklerin bazıları: Tamsayı kısıtlarını yok sayarak yapılan doğrusal programlama gevşetmesi, çeşitli kısıtları yok sayarak yapılan Lagrange gevşetmesi, tüm çözüm uzayı yerine daha akılcı bir eniyi arama yolu olan dallan-ve-sınırla tekniği ve verilecek kararlar dizisini terslen bulmaya çalışan dinamik programlama teknikleridir [17]. Üstel bir karmaşıklığa sahip olsalar da, bu teknikler özellikle küçük ve orta boy problemler için yararlıdır.

Buluşsal teknikler problem hakkındaki özel bilgi ve varsayımlara dayanırlar ve çok nadiren en iyi çözümü garanti edebilirler. Ancak, çok hızlı olmaları nedeniyle, çabukça bir iyi olurlu çözüm bulmakta yararlıdırlar. En-tipik buluşsal algoritmalar ağgözlü olanlardır. Bu algoritmalar her aşamada o an için en iyi olan karan seçerler. Diğer bir buluşsal teknik ise benzer bir problem için verimli çalışan bir algoritmayı kullanmaktır. Buna bir örnek: kapasite kısıtlarına uyan bir en iyi geren ağ bulabilmek için, geleneksel geren ağ algoritmalarını bu kısıtları da gözönüne alacak şekilde değiştirmektir. Bazı başarılı buluşsal algoritmalar [j]'de tanıtılmıştır.

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Önemli bir buluşsal teknik grubu yerel arama algoritmalarıdır. Bu algoritmalar olurlu bir topolojiden başlarlar ve bu topolojinin komşuları arasında daha iyi çözümler araştırırlar. Komşular birbirlerinden küçük değişiklikler yapılarak elde edilebilirler. Geleneksel yerel arama algoritmaları ağgözlüdür ve sadece daha iyi çözümleri kabul ederler. Bütün komşular arasında daha iyi bir çözüm bulamayınca kadar aramalarına devam ederler. Çok basit olmalarına rağmen, bu konudaki değişik eniyileme problemlerinin çözümünde başarılı olmuşlardır [17]. En büyük dezavantajları yerel eniyilere takılmalarıdır. Bu sorun bazı değişikliklerle aşılabılır. Bunun bir yolu, tavlama benzetiminde olduğu gibi arasıra daha kötü çözümleri de kabul elmekdir[4].

Tavlama benzetimi olurlu bir topolojiden başlayarak, komşular arasında araştırmaya devanı eder. Yeni bir komşu daha iyiye her zaman kabul edilir ve arlık onun komşuları araştırılır. Amaç işlevi açısından daha pahalı olan komşular ise belirli bir olasılıkla kabul edilirler. Bu olasılık iki topoloji arasındaki maliyet farkı ile ters orantılıdır ve bir kontrol parametresi kullanılarak gittikçe düşürülür. İlk başlarda algoritma çok daha rassal davranarak hemen hemen her türlü yeni çözümü kabul eder. Sonlara doğru ise rassallık çok azalarak, sadece daha iyi çözümleri kabul eden gerekirci bir yol izlenir. Yerel aramaya komşular arasında daha iyi bir çözüm olmadığına inanılınca son verilir. Tavlama benzetiminin detayları [2] ve [4]'de verilmiştir. Bu tekniğin başarımında komşu seçiminin ve kontrol parametresini değiştirme hızının çok önemi vardır, örneğin, LAN-MAN ağlarının tasarımında gereken geren ağaçların komşularını üretirken, varolan bağlardan biri ortadan kaldırılmış ve geriye kalan iki alt ağaç rassal olarak seçilen başka bir bağ ile birleştirilmiştir. Çok-hoplmalı ışksal ağlarda ise, iki ayrı bağ seçilerek, bu bağların uçları çapraz olarak değiştirilmiştir. Bu komşu ürcümc teknikleri her iki problem içinde iyi sonuçlar vermiştir [2], [15]. Komşular başka bir yolla üretildiğinde, algoritmanın izleyeceği yollar değişeceği için, seçilen topolojiler ve sonuçlar da değişecektir. Tavlama benzetimi, rassallığı nedeniyle, bazı çözümlerin üzerinden birden fazla kez geçebilir veya çözüm uzayının gereksiz olduğu bilinen köşelerini de arayabilir. Çözüm uzayının belli kısımlarının yasak bölge (tabu) sayılması ve buraların aranmaması yoluyla, aramanın verimini arttırmaya dayanan tekniğe de yasak (tabu) arama adı verilir [18].

Yerel eniyilcre takılma sorununa çözüm getiren tekniklerden biri de genetik algoritmalarıdır. Bu teknik kısaca, canlılarda çoğalma ve güçlü olanın yaşaması prensiplerine dayanarak çözüm uzayında yerli arama yapmak olarak tanımlanabilir [5]. Oldukça hızlı bir teknik olup, topoloji tasarımı probleminin bazı özel hallerine iyi çözümler bulabilirler [3].

Katımsal eniyilme problemi olarak formüle edilmiş topolojik tasarımı problemlerini çözmenin bir yolu da Hopfield tipi yapay sinir ağıları kullanmaktır [6]. Bu ağılar yüksek hızları ve paralel uygulamalara elverişlilikleri nedeniyle tercih edilmektedirler. Ancak, amaç işlevinin oldukça özel bir çokkrimli olmasını gerektirdikleri için, uygulama alanları sınırlıdır. Her ne kadar ilk olarak önerildiğinde gerekirci olsalar da, daha sonradan Gauss yakınaları gibi rassal sinir ağıları da önerilmiştir [19]. Yine, bu yolla yerel eniyilcre takılma önlenmeye çalışılmıştır.

## Sonuç

Veri iltişimindeki arlan isterler gittikçe karmaşıklaşan ağılar gerektirmektedir. Bunun sonucu olarak, bilgisayar ağılarının topolojik tasarımı ile ilgili problemlerle daha sık karşılaşılacaktır. Bu problemlerin hemen hemen tümü çözümleri oldukça zor katımsal eniyileme problemleridir. Bunlara iki örnek: yerel ağıların köprülerle bağlanmasında karşımıza çıkan enküçük gecikmeli gereken ağaç seçimi ve çok-hoplama ışıklı ağılarda karşılaşılan mantıksal topoloji problemleridir. Bu problemlerin normal boyutta olanları için bile tüm çözümlerin araştırılması olanaksızdır. Doğrusal ve tamsayı programlama gibi geleneksel teknikler ise çok uzun süreler gerektirmektedirler. Yerli arama gibi, gerekirci buluşsal algoritmaların çözümlerinin de istenilen iyilikte olmaması nedeniyle, rassal buluşsal teknikler kullanılmaktadır. Bu tekniklerden (avlama benzetimi, genetik algoritmalar ve rassal sinir ağıları topoloji tasarımı probleminin bazı özel durumlarına, kabul edilebilir zamanlarda, oldukça iyi çözümler buldukları için umul vericidirler.

## Kaynakça

1. R.R. Boorslyn ve H. Frank, "Latge-scalc neluork lopolical oplimization", IEEE Trans. on Commuicalins, pp. 48-60, January 1977.
2. C. Ersoy ve S.S. Panvar, "Topological design of intereonneed LAN-MAN networks", Proceedings of the IEEE INFOCOM, Milano, İtalya, May 1992.
3. A.İ. Oyman ve C. Ersoy, "Solving concntrator localion problems using genetic algorilhms", MELECON,94'e sunuldu.
4. S. Kirkpatrick, C. D. Gelatl ve M.P. Vecchi, "Optimization by simulated annealing", Science, pp. 671-680. May 1983.
5. D.E. -Goldbcrg, " Gcncllc algorilhms in search, optimizalion, and machine learning", Addison-Wesley, 1989.
6. J.J Hopfield ve D.W. Tank, " 'Neural' compıtalion of decisions in optimization problems", biological Cybcnelics, 141-152, 1985.
7. B. Gavislı, "Topological design of centralized compuler nctvorks", Networks, vol. 12, pp. 355-377, 1982.
8. A. Kerslienbaum, "Telecommunications nctwork design algorilhms", Mc Graw-Hill, 1993.
9. D. Bortsckas ve R. Gallager, "Data Nctvorks, 2nd Ed.", Prcnllice-Hall, 1992.
10. "Köprü ve yol-atayıcılar üzerine özel sayı", IEEE NctUvork, January 1988.
11. M.G. Hluclıy ve M.J. Karol, "Slurfncl: An applicalion of generalized perfcccl shufflcs to nullihop lighlwavc ncluorks". Proceedings of the IEEE INFOCOM, pp. 379-390. 1988.

12. M. Eisenberg ve N. Mehravari, "Performance of the multichannel multihop lightwave network under nonuniform traffic", IEEE JSAC, pp. 1063-1077, August 1988.
13. K. Sivarajan ve R. Ramaswani, "Multihop lightwave networks based on De Bruijn graphs", Proceedings of the IEEE INFOCOM, pp. 1001-1011, 1991.
14. J.P. Labourdette ve A.S. Acampora, "Rearrangeable multihop lightwave networks", IEEE Tr. on Communications, pp. 1223-1230, August 1991.
15. C. Ersoy ve S.S. Panjvar, "Topological design of multihop lightwave networks", Proceedings of the IEEE GLOBECOM, 1993.
16. L. Fratta, M. Gerla ve L. Kleinrock, "The Flow Deviation method: An approach to store-and-forward communication network design", Networks, pp. 97-133, 1973.
17. C. H. Papadimitriou ve K. Steiglitz, "Combinatorial algorithms - Algorithms and complexity", Prentice-Hall, 1982.
18. F. Glover, E. Taillard ve D. de Werra, "A user's guide to tabu search", Annals of Operations Research, Vol. 41, Nos. 1-4, 1993.
19. Y. Akiyama, A. Yamashita, M. Kajinra ve H. Aiso, "Combinatorial optimization with Gaussian machines", International Joint Conference on Neural Networks Bildiri Kitabı, pp. 1:533-540, 1989.

Cem Ersoy 28 Eylül 1963'te Ankara'da doğdu. Sırasıyla, 1984 ve 1986 yıllarında, Boğaziçi Üniversitesi, Elektrik Mühendisliği Bölümü'nden Lisans ve Yüksek Lisans derecelerini aldı. 1992 yılında, Brooklyn, New York, A.B.D.'deki Polytechnic University'den aynı alanda doktora derecesini aldı. 1984 ve 1986 yılları arasında, İstanbul'daki Netaş A.Ş.'de Araştırma ve Geliştirme Bölümü'nde çalıştı. 1987 ve 1992 yılları arasında, Polytechnic University'de Araştırma Asistanı olarak çalıştı. 1992 yılından itibaren, Boğaziçi Üniversitesi, Bilgisayar Mühendisliği Bölümü'nde Yardımcı Doçent olarak çalışmaktadır. İlgili alanları: bilgisayar ağları, katımsal eniyileme ve sayısal sinyal işlemedir.

# NÖRAL AĞ ALGORİTMALARI KULLANILARAK ALFABETİK DESENLERİN BİLGİSAYARA ÖĞRETİLMESİ VE GİRİŞ BİLGİSİNİ ARTIRMANIN EĞİTME ÜZERİNDEKİ ETKİLERİ

Adem KALINLI

Erciyes Üniv. M. Y.O. KAYSERİ

Zahide YILBAŞ

Erciyes Üniv. Müh. Fak. Elektronik Bölümü KAYSERİ

Ahmet ÖZEK

Erciyes Üniv. Müh. Fak. Elektronik Bölümü KAYSERİ

Mehmet ALTUNER

Erciyes Üniv. Müh. Fak. Elektronik Bölümü KAYSERİ

## ÖZET

Nöral Ağlar, sahip oldukları birçok farklı özelliklerinden dolayı sadece bilgisayar ile uğraşan bilim adamları için değil, psikologlar, fizyologlar ve algılama bilimi ile ilgilenen bilim adamları için de çok cazip bir araştırma sahasıdır. Beynin sahip olduğu özellikleri elde edebilmek için, insanlar çeşitli öğrenme algoritmaları üzerinde çalışmaktadırlar. Bu algoritmaların en iyi bilineni ve en yaygın olarak kullanılanı, geri-yayılım (back-propagation) algoritmasıdır. Özellikle desen (karakter) tanıma problemlerinde, diğer nöral ağ örneklerinden çok daha iyidir. Bu çalışmada, geri-yayılım algoritması kullanılarak alfabetik desenleri öğrenecek ve tanıyacak uygun bir nöral ağ oluşturulmuş ve bir bilgisayar programı geliştirilerek eksik desenleri tanımada başarılı neticeler elde edilmiştir. Alfabetik desenleri daha fazla giriş bilgisi ile tanımlamanın, eğitime üzerindeki etkileri incelenmiştir.

## 1. GİRİŞ

Nöral ağlar, beynin sinirsel ağ yapısını taklit eden matematiksel bir sistem üzerine kurulmuştur. Bir nöral ağ, herbiri genellikle sabitleştirilmiş sınırlı hesaplama kapasitesi olan basit işlem elemanlarının toplamından oluşur. Ağdaki her bir işlem elemanı, sinirsel yapıdaki tek başına nöron'a karşılık gelir. Her bir işlem elemanı, belirli sayıdaki basit sayısal bilgileri girdi olarak almakta ve basit sayısal bilgi olarak çıktı üretmektedir. Birçok işlem

elemanı çıkışı, diğer işlem elemanları için girdi olarak kullanılarak geniş nöral ağlar meydana getirilir [1].

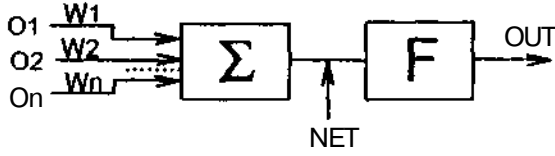
Nöral ağlar, bilinen mantıkla programlanamazlar. Bu ağların programlanmasında, çeşitli ağ algoritmaları kullanılır. Bunlardan en yaygın kullanılan ikisi, geri-yayılım (back-propagation) ve karşı yayılım (counter-propagation) ağ algoritmalarıdır. Programcı ağı, giriş ve ona karşılık gelen uygun çıkış verisini sağlayarak programlar. Ağ, işlem elemanlarını birbirine bağlayan ağırlıkları bir matematiksel model içerisinde değiştirerek öğrenir. Öğrenme sağlandığında, ağ hızlı bir şekilde giriş değerlerine karşılık gelen doğru çıkış değerlerini verir. Genel olarak, nöral uygulamalar işlemlerin çok karışık olduğu ve mantık adımları halinde programın kolaylıkla yazılmadığı durumlarda uygun ve hızlı çözümler vermektedir. İyi eğitilmiş uzman kişilerin bile yanılıya düşebildiği konularda, başarılı sonuçlar alınabilmektedir.

Geleneksel algoritmik teknikleri kullanarak, belirli türlerdeki problemleri çözmek zordur. Özellikle bir algoritmik yolla özelleştirilemeyen ve veri girdileri güvenilir olmayan endüstriyel problemler, mevcuttur. Bu tür problemlerde, verinin insanlar tarafından giriş olarak değil de doğrudan dış ortamdan gelmesine çok sık taslanır (ses veya görüntü gibi). Nöral şebeke tekniklerinin problemlerin çözümünde önemli yer tutması, bu tür problemlere duyulan ilgiyi artırmış ve birçok araştırmacı bu konuya yönelmiştir [2].

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

## 2. GERİ YAYILIM ÖĞRENME ALGORİT-MASI

Şekil 1 de, gen yayılım noral ağları için kullanılan temel nöron blok diyagramı görülmektedir.



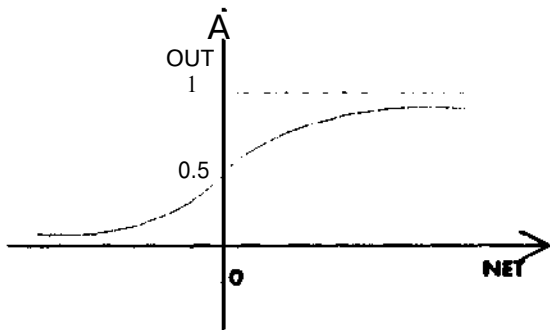
Şekil 1. Aktivasyon fonksiyonlu yapay nöron

Giriş seti, önceki seviyeden veya dışarıdan uygulanır. Bu girişlerin herbiri, ilgili ağırlıkla çarpılır ve bu çarpımların toplamı hesaplanır. Bu toplam değer, NET olarak adlandırılır ve ağda herbir nöron için ayrı ayrı hesaplanır.

$$NET = O_1W_1 + O_2W_2 + \dots + O_nW_n = \sum_{i=1}^n O_iW_i \quad (1)$$

NET hesaplandıktan sonra, aktivasyon fonksiyonu F 'e uygulanır. Bu suretle, çıkış sinyali OUT elde edilir. Geri-yayılım algoritmasında genellikle Şekil 2'de görülen "Sigmoid Aktivasyon Fonksiyonu" kullanılır

$$OUT = \frac{1}{(1 + e^{-NET})} \quad (2)$$

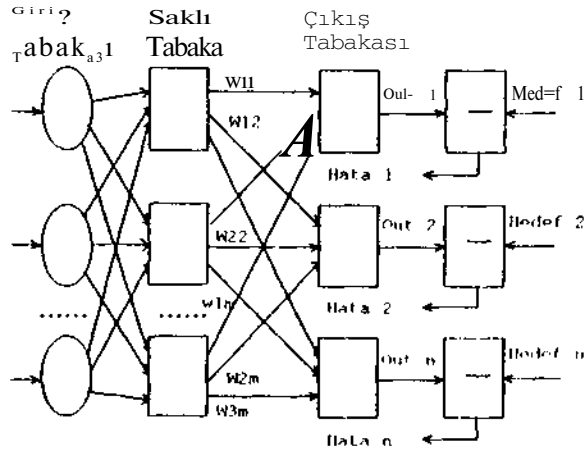


Şekil 2. Sigmoid aktivasyon fonksiyonu

Sigmoid fonksiyonu, çıkışın OUT sıfır (0) ile bir (1) arasında olmasını sağlar. Bu yolla, büyük sinyaller şebekeye uygunlaştırılabilir. Geri yayılım ile öğrenme için elverişli çok tabakalı bir ağ. Şekil 3'de görülmektedir.

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

$$\delta = F(NET) \times (HEDEF - OUT) \quad (3)$$



Şekil 3. İki-tabakalı geri yayılım ağı

Daire şeklinde gösterilen ilk nöron seli, girişlerin toplanmasını sağlamaya yardımcı olur. Geri yayılım ağları, çok tabakalıdır. Her bir tabakadaki işlem elemanı, bir sonraki tabakanın bütün işlem elemanlarına bağlanabilir. Nöral ağ, üç tabakadan oluşmuştur. Bunlar sırasıyla, giriş tabakası (input layer), saklı tabaka (hidden layer) ve çıkış tabakasıdır (output layer) [3].

öğrenmenin başlangıcında, bir giriş setine karşı, hedeflenen çıkış seti verilmektedir. Benzer şekilde öğrenme devam ederken farklı desenlerin öğretilmesi için, farklı değerleri tanımlayan her bir giriş setine karşı arzu edilen çıkış seti de verilmektedir. Bir giriş (input) ve o girişe karşı gelen (hedeflenen; target) çıkış öğrenme çifti olarak adlandırılır. Öğretilcek desenlere ait bütün öğrenme çiftlerinin oluşturduğu guruba ise, öğrenme seti denir [4].

Bir geri yayılım ağındaki öğrenme, aşağıdaki işlem adımlarını gerektirir:

1. Ağ girişine giriş vektörü uygulandığında öğrenme setinden gelecek öğrenme çiftinin seçimi.
2. Ağ çıkışının hesabı.
3. Arzu edilen hedef çıkış ve hesaplanan çıkış arasındaki hatanın hesabı.
4. Hatayı azaltacak bir yolla ağırlıkların avarlanması



Çıkış tabakası, hata değeri bastırma fonksiyonu  $F(\text{NET})$ 'in türeviyle doğru orantılıdır. Bu sonuç öğrenme yeteneğinin iyileştirilmesi için, öğrenme oran katsayısı olarak adlandırılan  $r$  ile çarpılır.  $r$  tipik olarak 0.01-1.0 arasında değişir. Her bir işlemci eleman için hesaplanan hata değerleri, ağırlıkların ayarlanmasında kullanılır. Saklı tabakadaki her bir nöron için hala değeri, şu şekilde hesaplanır.

$$\delta = F'(\text{NET}) \sum \delta_k W_{jk} \quad (4)$$

Aşağıda ağırlıkların ayarlanmasında kullanılan eşitlikler görülmektedir.

$$\Delta W_{ij} = \eta \delta_j W_{ij} \quad (5)$$

$$W(n+1) = W(n) + \Delta W \quad (6)$$

Her bir desene ait bilgiler girişe uygulandıkça, yukarıdaki işlem adımları, eğitime sayısı kadar tekrar edilir [5].

### 3. ALFABETİK DESENLERİN TANITILMASI İÇİN OLUŞTURULAN GERİ YAYILIM NÖRAL AĞI

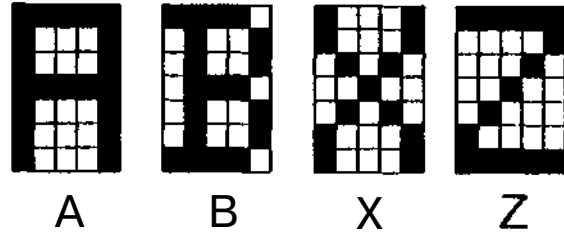
7x5 Matris formundaki 35 kareye yerleştirilen 26 temel alfabetik karakteri öğrenecek ve tanıyacak bir geri yayılım nöral ağı, tasarlanmıştır. Şekil 4'de, bazı alfabetik karakterlerin yerleştirilme düzeni gösterilmiştir. Burada veri olarak; dolu kareler bir (1), boş kareler sıfır (0) olarak kabul edilmiştir. Her bir alfabetik karakter 35 kare içerisine yerleştirildiğinden, giriş tabakasında 35 işlem elemanı ve tanınması istenen karakter sayısından dolayı da çıkış tabakasında 26 işlem elemanı bulunmasına karar verilmiştir [6].

Nöral ağ çıkışında 26 işlemci elemanı bulunduğundan, nöral ağ'a bir giriş deseni verildiğinde çıkışta 26 değer üretilir. Hangi girişe karşılık, istenilen değerlerin çıkışta arzu edildiği ise kod çözücüye benzer bir sistemle verilir.

Saklı tabakada yer alacak işlem elemanı sayısını belirlemede kullanılan belirli bir metod bulunmamasıyla birlikte, yapılan

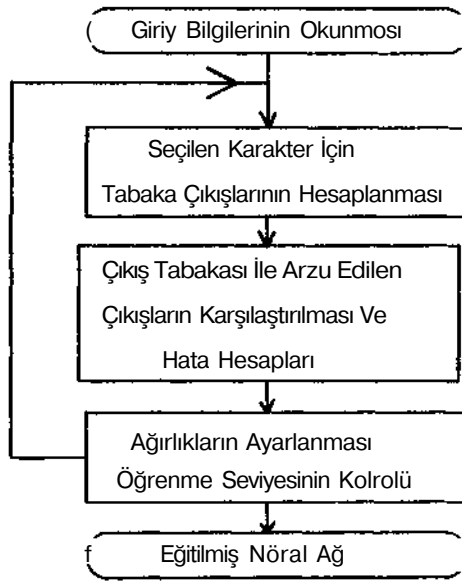
272

araştırmalarda genel olarak en uygun sonuçlar giriş ve çıkış tabakaları eleman sayıları arasındaki değerlerde elde edilmiştir. Yapılan denemelerden sonra, saklı tabakada 26 işlem elemanı kullanılmasının en uygun sonuçları verdiği görülmüş ve (35-26-26) şeklinde bir geri yayılım nöral ağı oluşturulmuştur. Bu ağda, giriş-saklı tabakaları arasında 910, saklı-çıkış tabakaları arasında ise 676 bağlantı bulunmaktadır [6].



Şekil 4. Alfabetik karakterlerin yerleştirilme düzeni

Bu model esas alınarak alfabetik desenlerin ağa öğretilmesi için, BASIC programlama dilinde bir program geliştirilmiştir. Şekil 5'de yazılan programın işlem akışı, blok olarak verilmiştir.

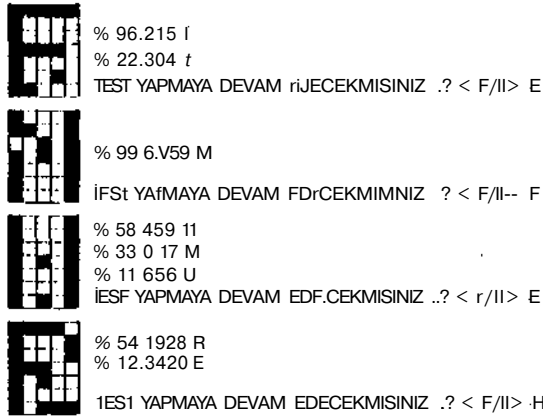


Şekil 5. İşlem akışı blok şeması

Programda ilk desene ait 35 veri alınır ve ilgili işlemlere tabi tutularak bütün işlemci elemanların katıldığı bir işlemden sonra çıkışlar elde edilir. Daha sonra, elde edilmek

istenilen veriden hedeflenen çıkışlar okunur ve gerçek çıkışlarla karşılaştırılarak her bir işlemci için hatalar hesaplanır. Başlangıçta bütün ağırlıklar, rasgele atanmıştır. Rasgele atanan ağırlık değerleri, hesaplanan hata değerlerine göre yeniden düzenlenir ve dosyaya kaydedilir. Bundan sonra yeni giriş bilgileri okunduğunda, yeni ağırlık değerleri ile işleme labi tutulur. Şebekenin işlemleri her tekrarlayışında, ağırlık değerleri, sürekli olarak yeniden düzenlenir. Bu olayın tekrarlanmasına, "KATMIT" adı verilir. 26 desenin herbiri için işlemler birkez yapıldıktan sonra yeniden ilk karaktere dönülür ve işlemlere eğitime süresi boyunca devam edilir.

Şekil 6'da, verilen bazı eksik desenler ve ağırlık bu desenleri tanıma yüzdesi görülmektedir.

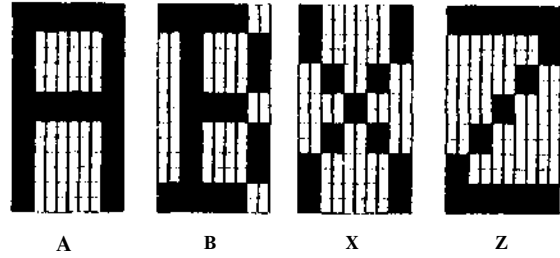


Şekil 6. Ağa verilen eksik desenler ve ağırlık tanıma yüzdesi

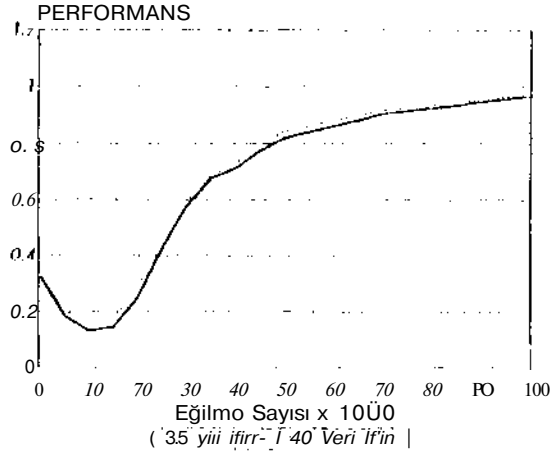
#### 4. NÖRAL AĞIN GİRİŞ TABAKASINDA YER ALAN İŞLEM ELEMANI SAYISININ ARTIRILMASI

Alfabetik karakterlerin giriş verisini artırmak için, Mx10 matris formunda 140 kareden oluşan yeni bir veri hazırlanmıştır (Şekil 7). Deneme yanılma yoluyla saklı tabakada 26 işlem elemanı bulunmasının en iyi sonuçları verdiği görülerek (140-26-26), formunda bir geri yayılım nöral ağı oluşturulmuştur. Oluşturulan ağ, tasarlanan desenlere göre eğitilerek elde edilen sonuçlar diğer nöral ağ ile karşılaştırılmıştır. Oluşturulan her iki nöral ağ, 100.000 eğitim işlemine tabi tutulmuş ve ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRE-2,

elde edilen performans grafiği Şekil 8'de verilmiştir [6].



Şekil 7. Alfabetik desenlerin bölünerek giriş verisinin artırılması



Şekil 8. Öğrenme işlemi sırasında performans değişimi

#### 5. SONUÇ

Geliştirilen geri yayılım ağına 26 alfabetik desenin %95.5'e varan seviyelerde öğretilmesi başarılmıştır. Ağı tanıma seviyesi, farklı desenler için az da olsa değişim göstermiştir. Bu durum, desenlerin yapısından kaynaklanmaktadır. Ağı saklı tabaka yapısında bulunan işlem elemanı sayısı, öğrenmeyi direkt olarak etkilemektedir. Saklı tabakadaki işlem elemanı sayısının uygun olmaması, öğrenme yeteneğini düşürmekle ve öğrenme zamanını artırmaktadır. Deseni tanımlayan veri sayısının artırılması giriş tabakasındaki işlem elemanı sayısını artırmakta, fakat saklı tabakadaki işlem elemanı sayısı üzerinde bir etkisi olmamaktadır. Giriş ve çıkış tabakalarındaki işlem elemanı sayılan, ağı eğitilecek desen sayısına bağılı olarak belirlenmekte, saklı

tabakadaki işlem elemanı sayısı için ise kesin bir kısıtlı bulunmamaktadır. Yapılan çalışmalarda saklı tabakadaki işlem elemanı sayısının, giriş ve çıkış tabakaları işlem elemanı sayıları arasında bir değerde olması gerektiği bulunmuştur. Ağ öğrenme işlemine başladıktan sonra kısa bir sürede %99'lara varan bir öğrenme seviyesine ulaşmakta ve bu seviyeden sonra ağı öğrenme yeteneğinde iyileşme daha yavaş olmaktadır.

#### KAYNAKLAR

- [1] YILBAŞ, Zahide, "Nöral Şebekeler ve Analizleri", E.Ü. Fen Bilimleri Ens., Yük. Lisans Tezi, Ocak 1990, Kayseri.
- [2] DAYHOFF, JUDITH E., "Neural Networks Architectures", New York, 1990.
- [3] AMAM S., "Characteristics of Fully Connected Random Nets of Analog Neuron-like Elements", IEEE Transaction on Systems, Man and Cybernetics Vol. SMC-2, No 5, pp. 643-667, 1972.
- [4] HIRAI, Y., "A Model of Human Associative Processor", IEEE Transaction on Systems Man and Cybernetics, Vol. SMC-13, No 5, pp. 853-860, 1983.
- [5] NVASSERMAN, PHILIP D., "Neural Computing Theory and Practice", 1989, New York.
- [6] KALINLI, ADEM, "Bir Geri yayılım Nöral Ağında Giriş Tabakasındaki İşlemci Elemanı Sayısının Artırmanın Performans, Eğitim Zamanı ve Saklı Tabakadaki Nöron Sayısı Üzerindeki Etkileri", E.Ü. Fen Bilimleri Ens., Yük. Lisans Tezi, Şubat 1993, Kayseri.



#### ADEM KAUNI.1

1968 yılında Sivas'da doğdu. Lisans eğitimi F.Ü. Müh. Fak. Elektronik Bölümü'nde yaptı. Yüksek lisans eğitimi 1992'de F.Ü. Fen Bilimleri Enstitüsünde tamamladı. Nöral şebekeli ve bilgisayar işlenileni üzerinde çalışmalarını sürdüren Kalıntı.

E.Ü. Bilgi İşlem Merkezinde Uzman, ve AI). Erciyes M.Y.O'nda Öğretim Görevlisi olarak görev yaptı. Halen E.Ü. M.Y.O'nda Öğretim Görevlisi olarak görev yapmaktadır.



#### ZAHİDE YILBAŞ

1962 yılında Kayseri'de doğdu. E.Ü. Müh. Fak. Elektronik Bölümü'nden mezun oldu. Yüksek lisans eğitimi 1990'da E.Ü. Fen Bilimleri Enstitüsü Elektronik ABD'nde tamamladı. Doktora eğitimi ise E.Ü. Fen Bilimleri Enstitüsü Elektronik

ABD'nde 1992 yılında tamamlandı. Çalışmalarını Nöral şebekeler üzerinde sürdüren Yılbaş. halen F.Ü. Müh. Fak. Elektronik Bölümü'nde Uzman olarak görev yapmaktadır.



#### AHMET ÖZEK

1966 yılında Eskişehir'de doğdu. 1988 yılında E.Ü. Müh. Fak. Elektronik Bölümü'nden mezun oldu. Yüksek lisans eğitimi 1991'de E.Ü. Fen Bilimleri Enstitüsü Elektronik ABD'nde tamamladı. Çalışmalarını Opto-elektronik.

Laser, Bilgisayar ve Bilgisayar kontrollü sistemler üzerinde sürdürmektedir. Halen E.Ü. Müh. Fak. Elektronik Bölümünde Araştırma Görevlisi olarak görev yapmaktadır.



#### MEHMET ALTUNER

1958 yılında Kayseri'nin Talaş ilçesinde doğdu. Yüksek Lisans eğitimi 1987'de Uludağ Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği ABD'nde. Doktora eğitimi de 1992'de Erciyes Üniversitesi Fen

Bilimleri Enst. Elektronik Mühendisliği ABD'nde tamamladı. Halen E.Ü. Müh. Fak. Elektronik Bölümü'nde öğretim üyesi olarak görev yapmaktadır.

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Ünlüakın A. Uğur, Gürgen Fikret S.

Boğaziçi Üniversitesi  
Bilgisayar Mühendisliği Bölümü

80815 Bebek İstanbul

### Özet:

Bu çalışmada, Japonca'nın birbirinden zor ayırt edilebilen altı sesbirimi üzerinde çalışılarak, klasik metotlardan En Yakın Komşu, Kanonik Diskriminasyon Analizi metotlarıyla, Hata Geri Yayıma algoritmasıyla eğitilmiş tek gizli katmanlı Yapay Nöron Ağlarının sınıflandırma becerileri karşılaştırmalı olarak test edilmiştir. Sonuçta, Yapay Nöron Ağlarının, diğer metotlara göre daha iyi bir sınıflandırma yeteneğine sahip oldukları gözlemlenmiştir.

### 1. Giriş:

Yapay nöron ağları, yeni eğitim algoritmalarının ve hızlı bilgisayarların son zamanlardaki hızlı gelişmelerine paralel olarak yeniden önemli bir araştırma konusu haline gelmiştir [1]. Basit hesaplama elemanlarının birleşmesinden oluşan ağlar, özellikle birden fazla kısıtın aynı anda sağlanmasını gerektiren tanıma ve optimizasyon problemleri için umut vadediticidir. Tanımda, bilgisayar bir girdiyi çeşitli özelliklerini gözönüne alarak kendisine en yakın özelliklere sahip olan sınıfa ilişikler. Yani tanıma işlemi aslında, nesnelere sınıflandırmadan çok farklı bir işlem değildir. Tanıma işlemleri - görme, ses ve doğal dil işleme gibi- çok sayıda değişik ve belirsizliğe sahip olduğundan, klasik seri programlama teknikleriyle genelde yeterli başarı sağlanamamaktadır. Yapay nöron ağları ise sinir sisteminin yapısına benzer bir yapı teşkil ettiklerinden, tanıma işlemleri için umut verici bir yapı oluşturmaktadırlar.

Sesbirimleri, sesin en temel dil birimleridir. Ses tanımda, hem tanıma işleminin hafıza gereksinimini azalttıkları için hem de geniş dağarcıktaki ses tanımayı kolaylaştırdıklarından dolayı sık sık referans birimleri olarak kullanılırlar.

Sesbirim sınıflandırma, ses tanımanın en alt seviyedeki işlemlerinden birisi olup, sesbirimleri arasındaki sınırların iyi tanımlanmamış ve bulanık olması dolayısıyla zor bir işlemdir. Bu çalışmada, Japonca'nın birbirinden zor ayırt edilebilen altı sesbirimi /b, d, g, m, n, N/ üzerinde çalışılmıştır. Ses 12 KHz'te örneklenmiş olup, 10 ms'lik zaman aralığında ilk 16 Mel-skalasındaki ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

spektral katsayılarla temsil edilmiştir. Her sesbirim vektörü 7 çerçeveden oluşmuş olup toplam ses analiz aralığı 70 ms'dir ve vektör 112 elemandır.

Sınıflandırmanın başarısının tespiti ve tanıma performansının belirlenmesi amacıyla üç ayrı veri setinden oluşan bir veritabanı kullanılmıştır. İlk iki veri setindeki vektörler, izole okunmuş kelimelerden alınmış olup, biri kullanılan teknikleri eğitmek, diğeri ise eğitilmiş tekniklerin, üzerinde çalışılmamış yeni vektörleri sınıflandırmadaki başarılarını sınamak amacıyla kullanılmıştır. Her bir sesbirim sınıfında 100 örnek olup, toplam veri seti büyüklüğü her bir veri seti için 600 vektördür. Bir üçüncü veri seti ise, yavaş hızla okunmuş cümlelerdeki kelimelerden alınmış olup, yine tekniklerin, üzerinde eğitilmedikleri vektörleri sınıflandırmadaki başarılarını tespit etmek amacıyla kullanılmıştır. Sınama amacıyla kullanılan bu veri setlerinden ilki Veri Seti 1, ikincisi ise Veri Seti 2 olarak adlandırılmıştır.

Klasik sınıflandırma metotlarından En Yakın Komşu (EYK) ve Kanonik Diskriminasyon Analizi (KDA) metotlarıyla, Hata Geri Yayıma algoritmasıyla eğitilmiş tek gizli katmanlı Yapay Nöron Ağlarının (YNA) sınıflandırma becerileri ölçülmüş ve sonuçlar karşılaştırmalı olarak sunulmuştur.

### 2. Metotlar:

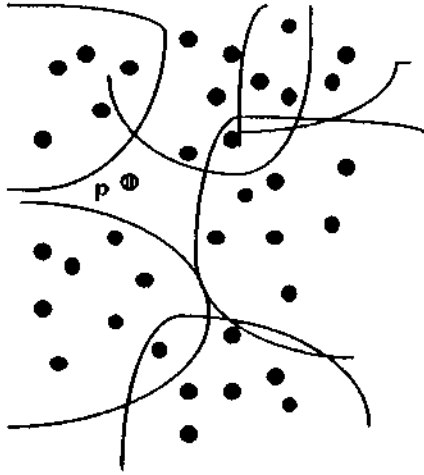
Sınıflandırma problemlerini en genel anlamda aşağıdaki şekilde ifade edebiliriz: m farklı grup olup her biri  $n_1, n_2, \dots, n_m$  elemandan oluşmaktadır. Grubun her elemanı ise p tane özeliğe sahip bir vektördür:  $x_1, x_2, \dots, x_p$ .

Özelde, elimizdeki problemde ise, sınıf sayısı altı olup  $m=6$ 'dır. Her grupta 100 eleman vardır ve dolayısıyla  $n_1=n_2=\dots=n_m=100$ 'dür. Her eleman ise 112 alt elemandan oluşmuştur ve  $p=112$ 'dir. İzleyen analizler, yukarıda verilen değerler yerlerine konularak yapılmışlardır.

## 2.1. En Yakın Komşu Metodu:

Bu metot, istatistiksel örüntü tanımının en klasik metotlarından birisi olup, sınama setinin örüntü vektörlerini, eğitim setindeki en yakın komşularının sınıflarına ilişkilendirir. Metot, örnek vektörlerin istatistiksel dağılımından bağımsız olup, yalnızca en yakın komşunun sınıfına göre bir sınıflandırma yapar.

Bu metotta, sınama setinin vektörleri teker teker alınarak, eğitim setinin her bir elemanına olan uzaklıkları ölçülür. En çok kullanılan mesafe ölçüsü Euclidean uzaklığı olsa da herhangi bir diğer ölçü de kullanılabilir. Sınıflandırılacak vektör, eğitim setindeki kendisine en yakın elemanın sınıfından sayılır (Figüre 1).



Figüre 1. En Yakın Komşu Metodunda p vektörü, kendisine en yakın vektörün sınıfına ilişkilendirilir.

Metot, daha matematiksel bir ifadeyle aşağıdaki gibi ifade edilebilir: p vektörünün sınıflandırılacağı varsayalım. Elimizdeki veritabanındaki toplam vektör sayısı 600'dür. Eğitim setindeki her vektörün sınıfı bilinmekte ve sınıfla vektör arasındaki ilişki şu şekilde ifade edilmektedir:  $C(X_j) = j$  ( $j \in \{1, 6\}$ ). Yine  $dist()$  adında, iki vektör arasındaki uzaklığı veren bir fonksiyon olduğunu varsayalım. O halde En Yakın Komşu metodu şu şekilde ifade edilebilir:

$$C(p) = C(\text{inin}(\text{dist}(p, X_j))) \text{ i c } [1, 600] \quad (2.1)$$

## 2.2. Kanonik Diskriminasyon Analizi:

Bazen sınıfların birbirinden daha iyi ayrılmasını sağlayabilmek amacıyla elde edilen vektör elemanlarının doğrusal kombinasyonlarından oluşan yeni değişkenler tanımlamak yararlıdır.

olabilir [2], Eğer ortalama değer, gruptan gruba önemli bir değişiklik gösterirken aynı grup içerisinde mümkün olduğunca sabit kalıyorsa, sınıflar bu yeni değişkenleri kullanarak birbirinden daha iyi ayrılabilirler.

Bu yöntem kullanıldığında, yeni elde edilen doğrusal kombinasyonların sayısı, analiz edilen vektörlerin eleman sayısının veya mevcut sınıf sayısının daha küçük olanı kadardır. Bu yeni kombinasyon vektörlerine diskriminant fonksiyonları denir:

$$Z_j = 3,7 x_j + a_2 X_2 + \dots + a_{112} X_{112} \quad (2.2)$$

Kanonik diskriminant fonksiyonunun katsayılarını (ajiller) bulmak bir özdeğer bulma problemi. Örneğiçi matris, V, ve Toplam matris, T, hesaplanır:

$$t_{sk} = \sum_{i=1}^6 X_i (X_{ij_s} - X_{j_s}) (X_{ij_k} - X_{j_k}) \quad (2.3)$$

$$V_{sk} = \sum_{i=1}^6 L_i (X_{ij_s} - X_{j_s}) (X_{ij_k} - X_{j_k}) \quad (2.4)$$

Burada  $t_{s,k}$  ve  $v_{s,k}$  sırayla T ve V matrislerinin s'inci satırdaki ve k'inci kolondaki elemanlarıdır.  $x_{ij}$ ,  $x_j$  elemanının j sınıfındaki i'nci elemanını belirtir.  $X_j$ ,  $x_j$ 'nin aynı grup içindeki ortalama değerini verirken,  $x_i$  tüm gruplar içindeki ortalama değerini verir.

Bu iki matris kullanılarak Örneklerarası matris, B, hesaplanır:

$$B = T - V \quad (2.5)$$

Daha sonra  $V^{-1} \cdot B$  matrisinin özdeğer ve özvektörleri bulunur. Tüm vektörler yeni bulunan vektör transformasyonundan geçirilerek yeni küçük boyutlu vektörler elde edilir. Yeni vektörler, hem orijinal bilgiyi tutarlar hem de sınıfları daha iyi ayırırlar.

## 2.3. Yapay Nöron Ağları:

Yapay nöron ağları, birbirine bağlı ve perseptron diye de adlandırılan basit hesaplama elemanlarının paralel çalışmalarına ve sinir sisteminin işleyişini taklide dayanan, yeni yeni geniş amaçlı kullanılmaya başlanan bir modeldir [3]. Model, basit hesaplama elemanları kullanmasına rağmen çok sayıda elemanın paralel çalışmaları sonucunda çok iyi bir performans gösterir.

Çok katmanlı perseptron, her girdi vektörü için bir çıktı vektörü yaratır. Çalışmamızda, girdi vektörü ( $x = (x_1, x_2, \dots, x_n)$ ) 112 elemanlı ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

olup, çıktı vektörü (  $y = (y_1, y_2, \dots, y_6)$  ) ise altı elemanlıdır (Her sesbirim bir elemanla gösterilmektedir). Hata geri yayma tekniğiyle eğitilen ağlar, hesaplama elemanları arasındaki bağlantı ağırlıklarını (  $w = (w_1, w_2, \dots, w^)$  ) ayarlayarak hedeflenen (  $d = (d_1, \dots, d_6)$  ) ve ortaya çıkan çıktı vektörleri,  $y$ , arasındaki farkı en aza indirmeye çalışır (Figüre 2).

Parametrik olarak ifade edilirse

$$y = O(x, w) \quad (2.6)$$

Burada  $O()$  kullanılan ağ yapısıdır.  $O'$ 'yı bir uzaklık ölçme birimi olarak kabul edersek, hedef fonksiyon şu şekilde ifade edilebilir:

$$E_p = \sum_{j=1}^6 O(dp_j - yp_j) \quad (2.7)$$

$$E = \sum_{j=1}^{600} E_j \quad (2.8)$$

Burada  $dp_j$  ve  $yp_j$  sunulan  $p$  vektörü için hedeflenen ve ortaya çıkan çıktı vektörleridir.  $E_p$  sadece bir vektörün ağa sunulduğunda ortaya çıkmış hata,  $E$  ise tüm vektörlerin ağa sunulmaları sonucu ortaya çıkmış toplam hatadır. Hata geri yayma metodu, gradient descent metodu uygulayarak ağı ağırlıklarını değiştirir:

$$w_{ji}^{yem} = w_{ji}^{eski} + \Delta p w_{ji}^{eski} + \alpha \Delta p w_{ji}^{eski} \quad (2.9)$$

Buradaki  $\alpha$ , momentum terimi olup salınımı engelleyici bir özellik taşır.

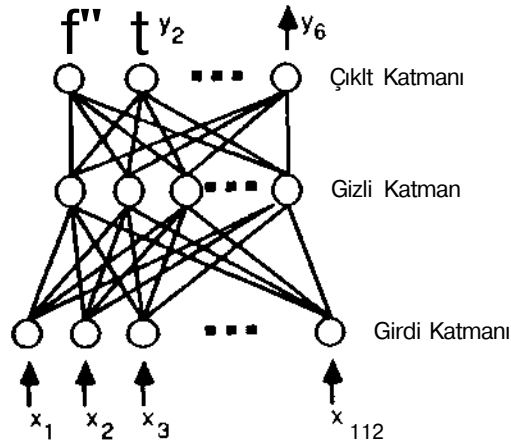


Figure 2. Çalışmamızda Kullanılan Tipik bir Yapay Nöron Ağı

### 3. Deneysel ve Elde Edilen Sonuçlar:

1. En Yakın Komşu metodu kullanılarak sınıflandırma yapılmış ve aşağıdaki sonuçlar elde edilmiştir.

Tablo 1. En Yakın Komşu Metodunun Sonuçları

EYK	Doğru Sınıf. (%)	
	Veri Seti 1	Veri Seti 2
	90.2	67.3

Sadece en yakın komşuya bakarak ilk veri setindeki vektörlerin %90.2'si, ikinci veri setindeki vektörlerin ise %67.3'ü doğru sınıflandırılmıştır. Metot çok basit ve süratli olduğu için referans olarak kabul edilmiş ve diğer metotlar ancak onun performansının üzerine çıkabildiğinde başarılı olarak kabul edilmişlerdir.

2. Kanonik Diskriminasyon Analizi metodu kullanılarak yapılan sınıflandırma sonucu elde edilen yeni veriabanı üzerinde çalışılmıştır. Yeni özellik vektörlerinin boyutları altı elemandan dört elemana azaltılarak, sınıflandırmadaki başarıları tespit edilmiştir.

Tablo 2. Kanonik Diskriminasyon Analizinin Sonuçları

KDA	Doğru Sınıf. (%)		
	6 özellik	5 özellik	4 özellik
Veri Seti 1	89.8	89.8	87.5
Veri Seti 2	52.1	52.3	51.8

Elde edilen doğru sınıflandırma oranları, En Yakın Komşu metoduyla elde edilenden daha az gerçekleşmiştir. Veri Seti 1'de, yakın sonuçlar alınmış olmasına rağmen, Veri Seti 2'deki sınıflandırma performansı çok düşmüştür. Analiz, boyut sayısını düşürmüş olmasına rağmen, aynı zamanda önemli ve ayırt edici bazı özelliklerin de yok olmasına neden olarak, sınıflandırma performansını düşürmüştür. Kullanılan özellik sayısı azaltıldıkça, doğru sınıflandırma oranının düştüğü tespit edilmiş ve en iyi sınıflandırmanın beş özellikli durum için olduğu saptanmıştır.

3. Hedef fonksiyon olarak ortalama karesel hata alınarak, tek gizli katmanlı ve 16, 32 ve 48 gizli düğümlü ağlar kullanılarak, yapay nöron ağlarının sınıflandırma performansları ölçülmüştür.

Tablo 3. Yapay Nöron Ağı Sonuçları

YNA	Doğru Sınıf. (%)		
	16 Gizli Düğüm	32 Gizli Düğüm	48 Gizli Düğüm
Eğitim Seti	100.0	100.0	100.0
Veri Seti 1	94.7	95.2	95.6
Veri Seti 2	68.0	70.7	68.5

Yapay nöron ağı kullanılarak yapılan sınıflandırmanın diğer iki metottan çok daha iyi sonuçlar verdiği gözlenmiştir. Eğitim seti üzerinde tam sınıflandırma sağlanırken, birinci ve ikinci veri setleri üzerinde elde edilen sonuçlar, diğer iki metot kullanılarak elde edilen sonuçlardan çok daha iyidir. En iyi ağ yapısı, 32 gizli düğümü olmaktadır (Veri Seti 1 ve 2'deki sonuçların ortalaması en yüksek olanı). Yapay nöron ağı ve farklı algoritmalar kullanılarak yine aynı veritabanı üzerinde yapılan çalışmalar, [4], [51], [6], [7] numaralı referanslarda bulunabilir.

4. KDA yöntemiyle boyutları küçültülmüş yeni özellik vektörleri kullanılarak, gruplar arası sınırlar yapay nöron ağı kullanılarak oluşturulmuştur. Tek gizli katmanlı ve sekiz ve 16 gizli düğümü bulunan ağlar kullanılarak, nöron ağlarının sınıflandırmayı nasıl etkilediği sınırlanmıştır. Bu sayede hem ağların eğitim süresi kısaltılmış hem de sadece KDA kullanılarak yapılan sınıflandırmaya göre daha iyi bir genelleştirme elde edilmiştir.

Tablo 4. KDA kullanılarak Boyutları Küçültülmüş Vektörlerin YNA ile Sınıflandırılması

KDA + YNA	Doğru Sınıf. (%)	
	8 Gizli Düğüm	16 Gizli Düğüm
Eğitim Seti	98.8	98.3
Veri Seti 1	91.0	91.2
Veri Seti 2	46.6	60.5

Bu metodun sadece KDA'ya göre üstünlüğü, veri setleri üzerinde sınıflandırma yapılırken, en yakın komşuya bakmak yerine, nöron ağlarıyla çizilen sınırlara bakılmasıdır. Bu sayede, belirli bir esneklik sağlanmış ve 16 gizli düğümlü ağ için oldukça iyi bir performans yakalanmıştır.

#### 4. Sonuçlar:

En Yakın Komşu metodunun, Kanonik Diskriminasyon Analizi metodundan daha iyi genelleştirme yapabildiği, ancak Yapay Nöron Ağlarının her ikisinden de daha iyi bir

sınıflandırma performansına sahip olduğu gözlenmiştir. KDA kullanılarak boyutları küçültülmüş özellik vektörleriyle eğitilmiş ağların, belirli ağ yapıları için, KDA'dan daha iyi sınıflandırma yapabildiği, ancak hiçbir boyut indirgeme işlemi yapılmadan eğitilmiş ağlardan daha kötü bir performans sergilediği görülmüştür. Sonuçta, Yapay Nöron Ağlarının, sınıflandırma problemleri için oldukça iyi bir performans gösterdikleri; Veri Seti 1 üzerinde %95.6, Veri Seti 2 üzerinde ise %70.7 doğru sınıflandırma oranına ulaşabildikleri gözlenmiştir.

#### Teşekkür:

Japonca sesbirim veritabanı NTT Human Interface Laboratories, Speech and Acoustic Lab. tarafından sağlanmıştır. Tüm çalışanlarına ve bilhassa Dr. Sadaoki Furui'ye çok teşekkür ederiz. Bu çalışma, TÜBİTAK tarafından desteklenmiştir (EEEAG-53).

#### Kaynaklar:

1. Rumelhart, D.E., and McClelland, *Parallel Distributed Processing*, Vol. 1, MIT Press, Cambridge, MA, 1986
2. Manly, Brian F.J., *Multivariate Statistical Methods A Primer*, Chapman and Hall, Bristol, 1986
3. Hush, D.R., and Horne, B.G., "Progress in Supervised Neural Networks," *IEEE Signal Processing Magazine*, pp. 8-39, Jan 1993
4. Ünlüakın, A.U., "A Comparative Study of Artificial Neural Networks for Phoneme Recognition", MS Thesis, Department of Computer Engineering, Boğaziçi University, 1993
5. Aratma, S., "Memory-based Function Approximation Using Neural Networks," MS Thesis, Department of Computer Engineering, Boğaziçi University, 1993
6. Ünlüakın, A.U., Alpaydın, R., Gürgen, F., Alpaydın, E. (submitted) "Distributed and Local Neural Classifiers for Phoneme Recognition," *International Symposium on Computer and Information Sciences 8*, 1993
7. Gürgen, F.S., Aikawa, K., and Shikano, K., "Phoneme Recognition with Neural Networks Using a Novel Fuzzy Training Algorithm," *Proceedings of IEEE International Joint Conference on Neural Networks*, Singapore, pp. 572-577, 1991

## Yazarlar:



A. Uğur Ünlüakın 1970 yılında doğdu. Boğaziçi Üniversitesi Bilgisayar Mühendisliği bölümünden 1991'de lisans, 1993'te yüksek lisans dereceleri aldı. 1991 yılından bu yana aynı bölümde araştırma görevlisi olarak çalışmaktadır.

İlgi alanları yapay nöron ağları, ses ve örüntü tanımadır.



Fikret Gürgen 1960 yılında doğdu. İTÜ Elektronik ve Haberleşme Mühendisliği Bölümünden 1980 ve 1982'de lisans ve yüksek lisans dereceleri aldı. Ohio State Üniversitesinden MS derecesi (1986) ve Akron Üniversitesi'nden Elektrik Mühendisliği doktora derecesi (1989) aldı. 1989-91 yılları arasında Japonya'da Nippon Telegraph and Telephone (NTT) şirketinde araştırmacı olarak çalıştı. 1991 yılında doçent unvanı aldı. Halen Boğaziçi Üniversitesi Bilgisayar Mühendisliği bölümünde çalışmaktadır.

İlgi alanları yapay nöron ağları,, konuşma işareti (ve genel işaret) işleme, örüntü tanıma, haberleşme, donanım konularından oluşmaktadır.





birimlerinin birbirleriyle iletişim kurma ve olabildiğince yüksek hızlarda iletişim kurma gereksinimlerinin giderek artması, konunun lirajı önemini giderek arttırmış ve bunun sonucunda modemler, iletişim kuramındaki gelişmelerin hızla uygulamaya konulduğu bir alan olmuştur. (ferçeklen ile tasarlanan modemler için yüzyılın ortalarında yaklaşık 1(1 bil/sn olan iletişim hızları, Ungerboeck'in kafes kodlamalı modülasyon tekniği sayesinde, bugün telefon kanal sığasına oldukça yakın bir değer olan 10200 bil/sn'ye kadar çıkmıştır.

Veri iletim hızını 10000 bil/sn'ye çıkararak ve kafes kodlamalı modülasyon tekniğine dayalı modemlerle T2 CROSS modülasyonu kullanılmaktadır. T2-CROSS modülasyonuna uygun, 1/2 oranlı kodlayıcı tasarımlarının en önemlilerinden biri Ungerboeck'in [2] önerdiği doğrusal kodlardır. Modemler için T2 tarafından V X2 standartları olarak kabul edilen kodlayıcı ise Ungerboeck'in yönlendirilmesine benzer şekilde Vei [2] tarafından tasarlanmıştır ve VU oranlı doğrusal olmayan bir katlamalı kodlayıcı ile kanal işaretlerinin 90°, 180° ve 270°'lik faz belirsizliklerine karşı bir farksal kodlayıcı içermektedir. Bu sistem yardımıyla kodlama hızı 1/2-QAM'e göre 1/2'lik bir kazanç sağlanmaktadır.

Ungerboeck tipi kodların sistemin hala hasarlımın iyileştirmesi gerçeğine karşı, alıcı tarafından algılanması kullanan kodçözücülerin, alınan işaret hakkında yaptıkları karşılaştırma sayısı, yani "kodçözme karmaşıklığı" oldukça yüksektir. Oysa, kodçözme karmaşıklığının azaltılması, kodçözücü tasarımının basitleşmesini sağlayacaktır. Bu nedenle son yıllardaki çalışmalar, yüksek kodlanın kazançlı fakal düşük kodçözme karmaşıklıklı sistemlerin tasarımına yönelmiştir. Bu tür sistemlerin tasarımı, "çok düzeyli kodlama/çok aşamalı kodçözme" tekniğine dayanmaktadır.

(çok düzeyli kodlama tekniğinde temel düşünce, modülasyonlu işaretin birden fazla kodlanın aralından belirlenmesi ve alıcı kısmında da çok aşamalı olarak çözülmesidir. İhini ve Hirakaun'un [1] önerdiği ve HIS (hnti İllrukma Sclnur) olarak anılan yapıda, modülasyonlu işaret her kodlayıcının çıkış bitlerinden bir tanesi göz önüne alınarak belirlenir. Yani, modülasyonlu işaret kaç bit ile simgeleniyorsa, kodlamayı sayısı da o kadardır. Verici kısmındaki her kodlamayı için alıcı kısmında ayrı bir kodçözücü vardır ve daha üst düzeydeki kodçözücüler daha alt düzeydeki kodçözücülerin çıkışlarını bir yan bilgi olarak kullanırlar. Helli bir kodlamayı sistemi için çok aşamalı kodçözme işleminin hala başarımı, en büyük başarımlı kodçözme işlemine göre biraz dalın kötüdür. İlinin karşılık, çok düzeyli kodlama ile sağladığı uyumluluk ve böylece kodçözme karmaşıklığını azaltması nedeniyle önerilmektedir. (çok düzeyli kodlamasının ilk önerildiği zamanlarda, sistemde blok kodlar ve seri kararlı kodçözücüler kullanıldığından önemli oranda bir kodlama kazancı elde edilememiştir. Ancak daha sonra, Yarıgıçlı ve İhini [1], katlamalı kodlar ve yumuşak ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

kararlı kodçözücüler kullanarak ve düzeylere ait karresel serbest Öklid uzaklığı ih' o düzeydeki katlanırdı kodun serbest İllumming uzaklığı arasında doğrusal bir ilişki olmasından yararlanarak, yüksek kodlama kazançlı sistemler tasarlanmıştır ve ayrıca bu sistemlerin oldukça düşük kodçözme karmaşıklığına sahip olduklarını göstermişlerdir.

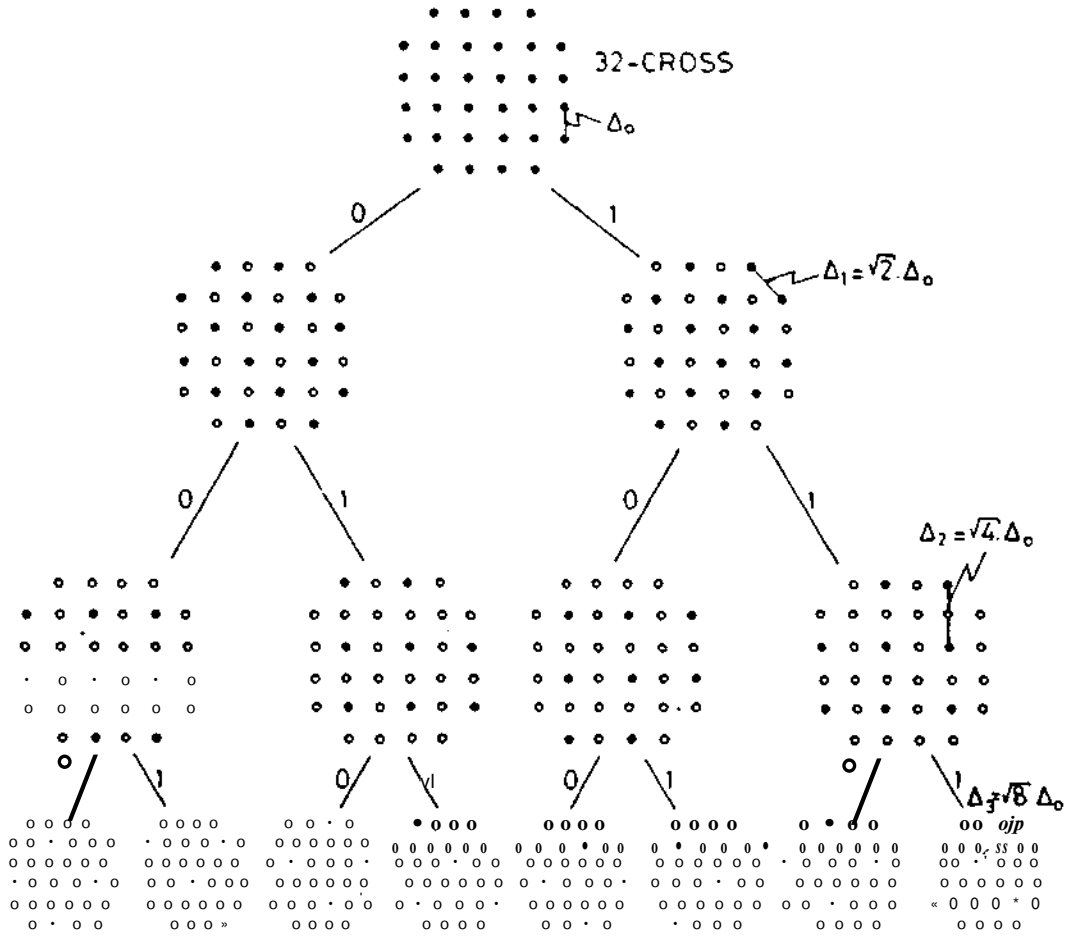
(çok düzeyli kodlayıcılar ilk kez L'ottie-Taylor [7] tarafından geliştirilmiş ve Ungerboeck kodları dahil, birçok kodun aslında çok düzeyli kodların özel bir biçimi olduğu gösterilmiştir. L'ottie-Taylor'un eleştirilirdi kodlar ile İllumming uzaklığı önerdiği kodların tümü çok düzeyli olmalarına karşın, aralarında kodlama işlemi açısından farklılık bulunmaktadır. (çok düzeyli kodlara ilişkin ayrıntılı hata başarımı analizi ilk kez Kolman, Zehavi ve Shamai [8] tarafından yapılmıştır.

Hu çalışmada V X2 standartları verilen 1/2 bil/sn modülasyonu T2-CROSS işaret kümesine dayalı kodlayıcı-modülasyon yapıları değiştirilerek ve çok düzeyli kodlama tekniği kullanılarak, yine 1/2 oranlı yeni yapılar önerilmiştir. (çok düzeyli yapının kodçözme karmaşıklığını azaltmasına ek olarak, kodçözme karmaşıklığını daha da düşürmek için yüksek kodlama oranlarında klasik katlamalı kodlar yerine, boşluklu katlamalı kodlar kullanılmıştır. Hu tür kodlar, genellikle 1/2 gibi düşük oranlı bir katlamalı kodlama yapılarındaki bazı bitlerin bir silme matrisine göre periyodik olarak silinmesi yoluyla oluşturulurlar. Sonuçta ehle edilen sistemlerin yüksek kodlama kazançlarına ve oldukça düşük kodçözme karmaşıklıklarına sahip oldukları görülmüştür. Ayrıca, önerilen çok düzeyli bir sistemin hala başarımlı analizi yapılmış ve Ungerboeck'in tasarladığı 1/2 oranlı kodlamayı ile karşılaştırılmıştır.

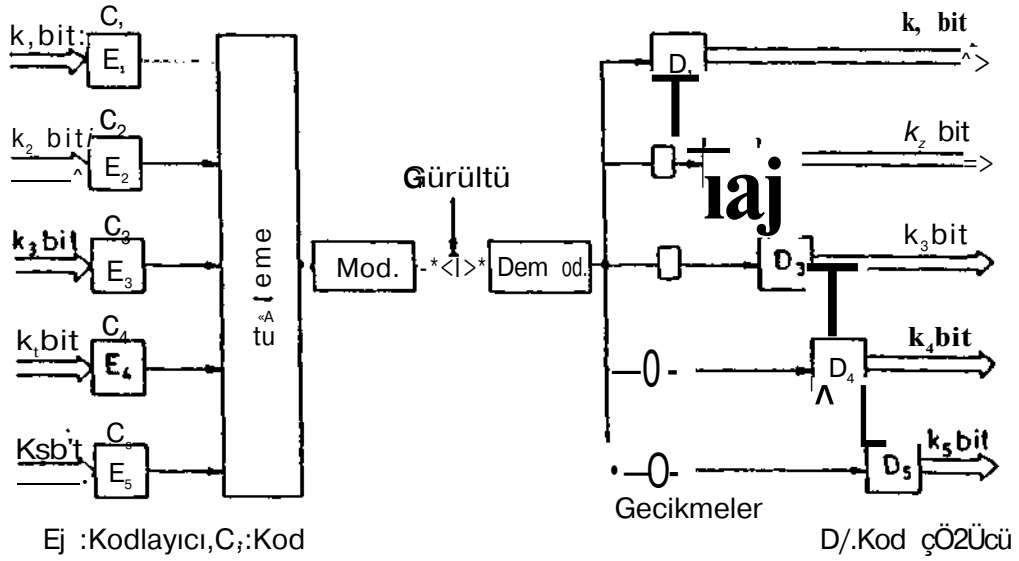
## 2 32-CROSS Modülasyonu İçin Tasarlanan Çok Düzeyli Sistemler

12-CROSS işaret kümesi Şekil 1'deki gibi bölmelelenebilir [1]. Şekildeki bölmeleme son iki düzey için ikiye bölerek devan eder. A<sub>0</sub>, A<sub>1</sub>, A<sub>2</sub> bölmeleme düzeyinde kodlamalar arasındaki en küçük Öklid uzaklığını göstermek üzere (i = 1, 2, ..., 2<sup>i</sup> - 1), sırasıyla A<sub>0</sub>, A<sub>1</sub> = √2A<sub>0</sub>, A<sub>2</sub> = √2A<sub>1</sub>, A<sub>3</sub> = √2A<sub>2</sub>, A<sub>4</sub> = √2A<sub>3</sub> değerlerini alır. 12-CROSS modülasyonu 1/2 işaret-te sahip olduğu için, çok düzeyli sistemde U adet kodlamayı (U<sup>i</sup>) vardır ve i. düzeydeki katlamalı kod (A<sub>i</sub>) ile gösterilir. Kle alınan çok düzeyli sisteme ilişkin blok diyagram Şekil 2'de görülmektedir. (çok kodunun serbest İllumming uzaklığı 'hm' i' olarak sistemin serbest Öklid uzaklığı

$$d_{min} = \min \{ \sqrt{2}^i h_m, A_1, \dots \} \quad (1)$$



Şekil 1. 32 - CROSS modülasyonu için küme bölmelmesi.



Şekil 2. 32 - CROSS modülasyonu için 5 düzeyli sistemlerin blok diyagramı.

ile verilir. (', kodunun hızı  $R$  /  $r$ ,  $r$  /  $N$  ise, tüm sistemin kod hızı

$$R = \int JU$$

biçimindedir. Katlamalı kodlayıcılar ve alıcı kısmında da Viterbi kodçözücü kullanılması durumunda, (', kodunu çözen /), kodçözücüsünün kodçözme karmaşıklığı [fi]

$$/., = 2^{K_i} \frac{(2^{L_i} - 1)}{n_i} \quad (3)$$

olarak tanımlanır. Kurada  $A_i$ , (', kodunun sınırlı uzunluğu, yani bellek elemanı sayısıdır. Eğer kodun kafes yapısında paralel geçiş varsa (II) ifadesi bir üst sınır olarak alınabilir. Yüksek kodlama oranlı klasik katlamalı kodlar yerine hoşluktu katlamalı kodlar kullanılması durumunda, <., /> oranlı hu tür 'f'J' koduna ilişkin kodçözme karmaşıklığı,  $i$ . düzey için (S),

$$L_i = 2 \frac{...}{7} \quad (4)$$

ile verilir, /, / için toplamı, tüm sistemin kodçözme karmaşıklığı /.'yi verir. Kodlamasız düzeylerin kodçözme karmaşıklığı sıfırdır.

Tablo 1. M-CROSS için 4) «nnh 1 düzeyli kodlayıcılar

Siftrn	C,	R,	K,	$d_{in}$	$d_{e}/d_s$	AKK [TBI]	AKK-ü«7 [dB]	L	
CR-1	C <sub>1</sub>	1/3	1	1	1	4.77	0.77	S	
	C <sub>2</sub>	3/6	7	5	1				
	C <sub>3</sub>	11/12	7	2	1				
	C <sub>4</sub>	(•)	-	1	1				
CR-2	C <sub>1</sub>	1/1	1	1	6	4.77	0.77	10	
	C <sub>2</sub>	3/4	7	3					1
	C <sub>3</sub>	3/4	7	3					1
	C <sub>4</sub>	(i)	-	1					1
C/1-3	C <sub>1</sub>	1/1	1	1	7	.114	1.44	K.SS	
	C <sub>2</sub>	3/4	3	2					1
	C <sub>3</sub>	3/4	3	2					1
	C <sub>4</sub>	(o)	-	1					1
CR-»	C <sub>1</sub>	1/2	4	7	7	5.44	1.44	17	
	C <sub>2</sub>	3/4	3	4					1
	C <sub>3</sub>	3/4	2	3					1
	C <sub>4</sub>	(1)	-	1					1
Cff-5	C <sub>1</sub>	1/1	1	1	«	8.02	202	İt	
	C <sub>2</sub>	J/4	4	4					1
	C <sub>3</sub>	U/12	1	2					1
	C <sub>4</sub>	(d)	-	1					1
CR-1	C <sub>1</sub>	1/2	5	5	8	«02	202	21.6«	
	C <sub>2</sub>	2/3	3	4					1
	C <sub>3</sub>	2/3	2	2					1
	C <sub>4</sub>	(1)	-	1					1
CR-1	C <sub>1</sub>	1/2	5	5	8	«02	2.02	25	
	C <sub>2</sub>	3/1	3	4					1
	C <sub>3</sub>	V4	2	3					1
	C <sub>4</sub>	(D)	-	1					1

32-CROSS modülasyonu için tasarlanan yeni sistemler Tablo 1'de verilmiştir. Tabloda, AKK, kodlamasız li-C^AM'e göre asimptolik kodlama kazancı; ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

Ah l' - Vllg. ise llljrrrherhoer'ın !!2 ('IU)SS için tasarladığı 1/5 oranlı, S dırıunlu koda \2\ K^m< kazancı; ar-tıjını RÖsl.erniektedir. [•', kodlamalı :12-(IU)SS için; /•?', kodlaması/ M» QAM için ortalama işaret enerjilerini v'k ''/';, ~ Aj';, kodlamasız Ki-CJAM için kar'sıl serliest Öklid uzaklığını j;cisleniek üzere asimptotik kodlama kazancı [1],

$$AKK [dB] = 10 \log \left( \frac{J \pm LJS^{\wedge}}{r's l^{\wedge} MI} \right) \quad (5)$$

hağmtısıyla hesaplanabilir. Ayrıca, /•', //?', = '2 olduğıu kolaylıkla gösterilebilir.

'nhlo l'leki ('/? - 5 sistemine ilişkin serbest Öklid uzaklığının  $d/p = \sqrt{RA}$  oluluğı (1)dein görölmektedir. Hııııı göre (7f-5 sistemi, Ungerboer'ını koduna göre 'l il l i asimptolik kodlama kazancı sağlamaktadır. (2)dein, bu sistem için /fıııı değışmediğı kolayca görölmektedir İler düzey için kodçözme karmaşıklığı /., = i.:.l. L., ~.ii, /s., ~.107, /., ~.0, L., = 0 olmak üzere, topi;:m karmaşıklık L = İldir. Oysa, Ungerboer'ın tipi \jTı oranlı kodun kodçözme karmaşıklığı daha yüksektir, (•erçekl.eri de, 2^\* dimimin, L/u oranlı ve kafes diyagramında paralel geçiş olan kallamalı kodlara ilişkin kodçözme karmaşıklığı

$$/., r- /? (•'/' - D) \cdot 2^{V2' - D} \quad (6)$$

bağıntısı ile hesaplanabilir. /?, kafes diyagramında lüm durululara ge|et farklı altkume sayısını ve k'. kodlanarak geçen giriş bili sayısını göstermektedir. Kuradan, sözü edilen llingerhoeck kodu için karmaşıklık /., = .18 olarak bulunur. ('örüldüğü gibi bu değ, CII - T) sisteminin kodçözme karmaşıklığından oldukça yüksektir, 'nhlo İde verilen diğer sistemler için de benzer karşılaştırmalar yapılabilir.

Kaşarım ölçütü olarak asimptolik kodlama kazancının alınması, ancak yüksek işaret/gürültü oranları için sağlıklı olmaktadır. Kı büyük olasılıklı hata olayına dayalı bu ölçül, düşük işaret/gürültü oranlarında daha küçük olasılıklı hata olaylarının da etkin olması nedeniyle anlamını yitirmektedir, İlıı yüzden, kodlavırılaitn aklarının işlevleri yardımıyla hata olasılık üst sınırlarının hesaplanması yoluna gidilmektedir

(•»]•

### 3 Hata Başarım Analizi

(,ok düzeyli kodların çok aşamalı kodçö/titüüne ilişkin analitik hata başarım analizi, kanal gürültüsünün sıfır ortalamalı ve A' /2 varyanslı beyaz (Jnıss gürültüsü olduğu varsayımı altında yapılmıştır. Ayrıca kodçözürülerı taşıdıkları yan bilgilerin her zaman doğru olduğu varsayılmıştır. İtu durumda i. düzeydeki kodçöziicüye ait. bit hata olasılığı üst. sınırı

$$P_{e,i} < U_n \sqrt{\frac{d_{jE_i}^2 E_s}{2N_n}} \exp\left(-\frac{d_{jE_i}^2 E_s}{4N_n}\right) \frac{\partial T_i(W^{(i)}, D)}{di} \quad (7)$$

ile verilebilir [10]. Muratla  $d_{ij} = \sqrt{A_{ij}^2}$ ,  $i, j$  düzeydeki karesel serbest Öklid uzaklığını göstermektedir. Ayrıca  $Q(i) = (\sqrt{2n}) \exp(-i^2/2n)$ , Gauss hata işlevi olup türev alındıktan sonra  $i$  yerine 1 konulmaktadır.  $C_i$  katlamalı kotluna ilişkin aktarım işlevi

$$T_i(W^{(i)}, t) = \sum_{j=1}^{\infty} \sum_{k=1}^{\infty} a_{jk} [W^{(i)}]^{j+k} \quad (8)$$

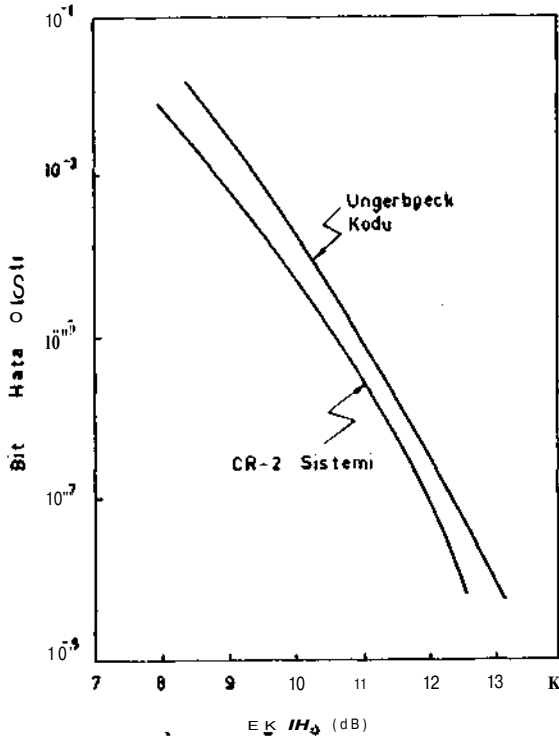
olarak yazılabilir.  $i, j, k$  kafes yapısındaki  $i$  uzaklıklı yol çifti sayısıdır. Genel olarak,  $2^N$  elemanlı bir işaret uzayı için,  $i$  düzeydeki kodçözücünün "ilgilendiği" işaret kümesi  $2^{N-i+1}$  elemanlıdır. Bu küme  $\{s, \dots, (\cdot)\}$  ve  $\{s, \dots, (\cdot)\}$  biçiminde gösterilebilen iki ayrık altkümenin birleşiminden oluşmuş olarak düşünülebilir ( $m^{(i)} = 0, 2^i, \dots, 2^N - 2^i$ ,  $m^{(i)} = mC + i^{i-1}$  ve  $i = 1, 2, \dots, M$ ), iler bit için, en büyük olasılıklı hata, iletilen kanal işareti yerine en yakın kanal işaretinin seçilmesiyle oluşur. Eğer  $A^{(i)}$   $\{s, \dots, (\cdot)\}$  ile  $A_j$   $i$  uzaklıktaki toplam işaret sayısını gösterirse, hata ağırlığı

$$W^{(i)} = \frac{A^{(i)}}{2^{M-i}} \exp(-A^2 / E_b / 4 A^2) \quad (9)$$

olarak alınabilir. Toplam bit hata olasılığı ise [8]

$$P_b = \frac{\sum_{i=1}^M R_i P_{bi}}{\sum_{i=1}^M R_i} \quad (10)$$

biçimindedir. 32-CROSS için  $M = 5$ 'tir ve  $4^5 = 52$ ,



Şekil 3.  $Cli-2$  sisteminin ve Ungerboeck kodlu için bit hata olasılığı  $E_b/N_0$  sınırları.

$4^5 = 52$ ,  $4^{(5)} = 10$ ,  $A^5 = 3$  ve  $4^{(5)} = 1$  olduğu gösterilebilir.  $Cli-2$  sisteminin ve 8 ilünlü Ungerboeck kodunun bit hata olasılığı üst sınır eğrileri karşılaştırılmıştır. Dırarla,  $E_b$  bit başına düşen ortalama enerjiyi göstermektedir. Şekilden varılan önemli bir sonuç, 32-CROSS modülasyonu için çok tüzeyli kodlamanın kodçözme karmaşıklığını oldukça azaltmasının yanında, hata başarımı açısından da Ungerboeck koduna göre üstünlük sağladığıdır. Gerçekten de,  $Cli-2$  sistemi için bit hata olasılığı üst sınırının  $10^{-6}$  değeri,  $E_b/N_0 \sim 11.3$  dB iken elde edilmektedir. Oysa, Ungerboeck kodu, aynı değeri, daha yüksek bir işaret/gürültü oranında, yani  $k_t/V_0 = 11.7$  dB iken almaktadır.

## 4 Sonuç

32-CROSS modülasyonu için, çok tüzeyli kodlama/çok aşamalı kodçözme tekniğine dayalı, 4/5 oranlı sistemler tasarlanmıştır. İler düzey için uygun kodlama oranları belirlendikten sonra, kodlama kazancını yükseltmek ve kodçözme karmaşıklığını azaltmak amacıyla, belli bir kotlama oranında serbest Hamming uzaklığı maksimum ve belli bir kullanım uzaklığımla sınırlı uzunluğu miliminin olan katlamalı kodlayıcılar kullanılmıştır. Ayrıca, kodçözme karmaşıklığı daha tta düşürmek için yüksek kodlama oranlarında boşlukta katlamalı kodlar kullanılmıştır. Tasarlanan sistemler kotlama kazancı, kodçözme karmaşıklığı ve bit hata basanını açısından incelenmiştir. Sonuçla; alıcı kısımda birden fazla kodçözücü kullanmak, çok aşamalı kodçözme işleminde kotlun çözülmesini bir miktar geciktirse de, önerilen sistemlerin Ungerboeck'in 1/5 oranlı ve 8 ilünlü kotluna göre 2  $(//<$  kadar daha fazla bit ktlbilimsiz 1(-)QAM'e göre  $dH$  kadar kodlama kazançları sağlattıkları gösterilmiştir. Diğer yandan, kodçözme karmaşıklıklarında 18'den 8'e varan düşüşler elde edilmiştir.

## 5 Kaynaklar

- [1] (i. Ungerboeck, "Channel coding with multi-level/phase signals", *IEEE Trans. Inform. Theory*, vol. IT-28, pp.55-07, January 1982.
- [2] (j. Ungerboeck, "Trellis coded modulation with trellis terminated signal sets-Part I: aul 2", *IEEE Commun. Mag.*, vol.25, no.2, pp5-21, February 1987.
- [3] "A family of 2-wire duplex modems operating at data signalling rates of up to 9600 bit/s, for use on the general switched telephone network on leased telephone-type circuits", (*CI'V Iecomneuduhon V.32*, Malaga-Torremolmus. 1984.

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

- [1] L.F. Wei, "Isolationally invariant convolutional channel coding with expanded signal space", *IEEE Trans. Selected Areas in Commun.*, vol.SAC-2, pp.072-686, September 1984.
- [5] H. Imai and S. Hirakawa, "A new multilevel coding scheme using error-correcting codes", *IEEE Trans. Inform. Theory*, vol.IT-23, no.3, pp.371-377, May 1977.
- [6] K. Yamaguchi and H. Imai, "Highly reliable multilevel channel coding system using binary convolutional codes", *Electronic Letters*, vol.23, no.18, pp.939-941, August 1987.
- [7] G.J. Pottie and D.L. Taylor, "Multilevel codes based on partitioning", *IEEE Trans. Inform. Theory*, vol.IT-35, no.1, pp.87-98, January 1989.
- [8] Y. Kofman, E. Zehavi, and S. Shitrit (Shitrit), "Information analysis of a multilevel coded modulation system", *IEEE Trans.*, no.751, Technion, Israel, 1990.
- [9] Y. Yasuni, K. Kasahara, and Y. Hirata, "Highly reliable punctured convolutional codes for soft decision Viterbi decoding", *IEEE Trans. Inform. Theory*, vol.COM-32, pp.315-319, March 1984.
- [10] K. Zehavi and J.K. Wolf, "On the performance evaluation of trellis codes", *IEEE Trans. Inform. Theory*, vol.IT-33, pp.191-202, March 1987.



Ümit AYDIN, mühendis, yüksek lisans ve doktora derecelerini sırasıyla 1978, 1981 ve 1989'da İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi'nden aldı. Yıldız Üniversitesi'nde 1980-1980 yılları arasında araştırma görevlisi, 1986-1989 yılları arasında öğretim görevlisi olarak çalıştı. 1989-1992 yılları arasında İTÜ Elektrik-Elektronik Fakültesi Haberleşme Anabilim Dalı'nda yardımcı doçent, olarak görev yaptı. 1992'de doçent oldu. İletişim kuramı ve iletişim kuramı konularında dersler vermekte ve halen kodlama ve modülasyon sistemleri ile ilgili araştırmalar yapmaktadır.



İbrahim ALTUNBAŞ, 1907 yılında İsparta'nın Sütçüler ilçesinde doğdu. İlk ve orta öğrenimini Sütçüler'de tamamladı. 1984 yılında girdiği İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi'nde, 1988 yılında Elektronik ve Haberleşme mühendisi olarak mezun oldu. 1992 yılında aynı üniversitede yüksek lisans eğitimini tamamlayarak yüksek mühendis unvanı aldı ve doktora öğrenimine başladı. Aynı zamanda İ.T.Ü. Elektrik-Elektronik Fakültesi Haberleşme Anabilim Dalı'nda araştırma görevlisi olan İbrahim Altunbaş, halen iletişim sistemlerinde kodlama ve modülasyon konuları üzerine araştırmalar yapmaktadır.

# MİKROBİLGİSAYAR KONTROL İH Vİ ZSLRİ BAĞLAN TILI

## C;KNKL(İRİŞ-(İKİŞ İŞİ.LMC İ KARTI

Yrd Doç. Di Doğün İBKAHİM  
Yüküm Dūānı Üniversitesi, Bilgisayar Mühendisliği Bölümü, 1.celkoça

### ÖZET

\ lıknışleuuc iler ve mikrobilgisayarlar gurumumuzuk hemen hemen bulun elektronik kontrol \c olşine eıha-lannda kullanılmaktadır .Hu çtdışmada, tek yon^u mikrubılı^ısın^tır kullanarak tasannu y.erçikleştinlmış olun iv eçk miktarda Kiriş ve çıkışıl Mı/H/ olun bir işlemei kurumlun bahsedilmişin'. Kurt, herban^ıbir model biliiisiyvtını R.SJ.İ2 sen oltinik ballanır ve bılaşım^ıunlan almış olduđu komutlara t^>^v kendi içiriş ve çıkış porlarını kontrol eder w bu şekilde bıh\ısa\ıum ilış dunva ile irtibatı sını^ıaimiş olur

### 1. GİRİŞ

Bilgisayarı destekli otomasyon yuinninü/o'e lıllı;is.i>amı en yaygın uygınlamaları ana.sıııladı Sıcaklık kontrolü, basınç kontrolü, İn/ konııoluı gibi iııomas^non konulan artık bilgisayarı ktıllanarak Napiılmakladı Ne şekilde bilgisayar olınsa olsun, bu gibi uygınlamalaula bilgisayarın içeiisine gırış ve ^ııl.^.l.m konııol edecek kaillaı lakılmakladı Hu kailli,m genelde çok ınıklıkta paralel gırış ^e paralel çıkıklar ılıma elmeklediler . Bu gibi kartlar bilgisa>anın BUS sistemine bağlamıı ve bundan dolayı kullanılacak olan bilgisayarııı elektronik şapışıına tamamen bağımlı kalırlar. Örneğin. bu inim bilgisayar için kullı,ıunlan bir gıı^ı-çıkı^ kailinin haşk.ı model bıı bilgisasıda kullanılması u m lı ik n n değıldı Bilgisayarı üıelen lınn.at.tıı ulan bu sıkı bağımlılıktan dolayı bilgisayarı gırış,-çıkı> karırlarının fiyatları da oldukça yüksektii

İllı yı/nm/da. lıei çeşil bilgisa>arıda kullanılabilecek \c s.ı/ıı tatalından lasannıı gerçekleştiril ilnıp olan genel bıı gırış-çıkı^ kanından bahsedilmişin Mu kan. lıeııhangıbu t)ilgisayarı m sen illelışim poiııuuı l>ai;lanıp bıııı.sayarıdan aldıđı komutları sa>esinde kendi paralel gıı.Ş ve çıkınlanın konııol edei

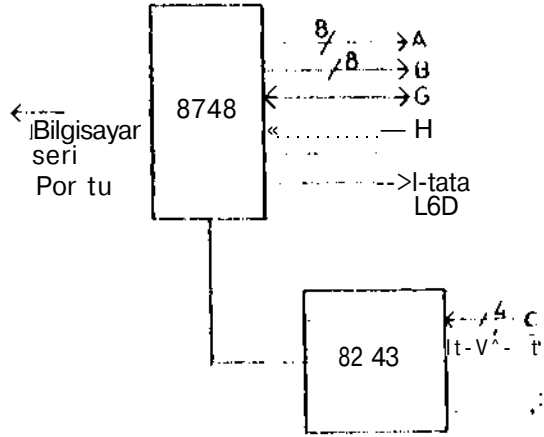
Böylece, bağlanmış okluğu bilgisayarı a gırış-çıkış poııııan sađlamış olur.

Kailin en önemli ö/elliđi ve gelirmiş okluğu yenilik ise lıerııangıbir bilgisayar ile çalışabilir olması ve kullandıđı bilgisayarın elektronik yapısına hiç bağımlı olmayışıdır.

### 2. (İRİŞ-K.İ.UİŞ-(İKİŞ KAIMİNIN YAPISI

Hu şa/ımı/da bahsetmiş okluđumu/, geii^I gırış-çıkış kain 87-İS modeli (/I/) lek yonga bir mikrobilgisayar kullanmak gerçekleştirilinişli Kailin blok şeması şekil I de gıııleniınışlii

N7 18 oklukça popüler S bııık lek yonga bu mikrobilgisayar olup bu yonga ıı/ciıııdu I K HI^ROM program halı/ası, ol Byte RAM bııııı halı/ası, /arıııııılayıcı/sayıcı ve gırış-çıkış poıııııı bulunakladı S7.18 ile bıııııkıe, K2-H gırış-çıkış >ongası (M/) kullamılmışlır. Bu yonga, karını lıııııııı gırış ve çıkış sayısını arıııııııılır.



Şel. I (İıııı (iıi) İUı Karlı İtoK ŞemaM

İncelendi. İş-çıkış kablolu, tanımlanmış

Pvıl A S hıl sıkı,  
l'oil li X 1)11 çıkış

Poil ( -I bil gniş  
l'oil O -I bil gniş  
Poil i: -I hıl gniş  
l'iiii I' -I hıl gniş;

l'oil (i I hıl gniş  
l'oil 11 I hıl gniş

l'oil K 100 11/dock  
l'oil (i I bil gniş \e\ə  
2MH/doek

Hoylece, genel iş-çıkış kabil topları 16 çıkış, IS giriş ve iki tarafe de kade dalga üreten portları ililiva etmektedir.

Kail, bilgisayarın KS2 >2 seri portuna bağlamı Oala soma. bilgisa)arıda >a/lan bu portları ile kana seri olarak komut gönderili. Kail ise bu komutlara uygun olarak kendi gniş ve çıkış portlarına istenilen bilgiyi gönderili ve bilgisa)arı bu şekilde dış dünyaya ile lenisiu kın müş oliri.

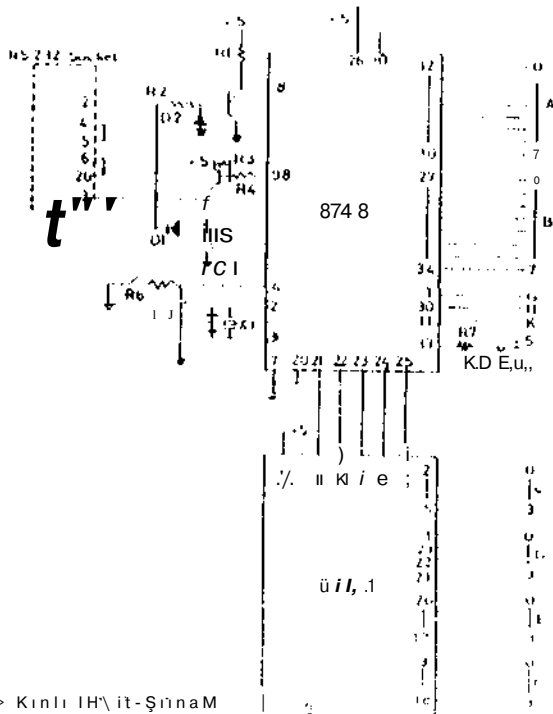
Şekil 1. tasannu gerçekliğin ilen genci c.iş-vıkış karlının devre şemasını gosiieuikle.lı lıuada. göüleeegi gibi, kailin mal oluş lhalım mımıkun olduđu kadar a/ tutmak Klıı pan,a seri,ınine ok

uiri-ur \e\ ilmişlu l )e\ lede is; is ola lak X7 IX uuk lı lık \onga bu iiki, bil; i. sa\ar, X' l i HHCı l' lıy ..ıkış \OIUMM. "SOS modeli \olla) legüladı, sonrasi \c bıkaı, lam: ile lıansislor buluuınakladı l'oil A \ e M dilek olaiak S7-1S \o ilgasını l >li \ı: l' l portları mullan aliminsin. (huş portları ( : l) i. \ e l' IM; K24 f songasının l' l. l'S, P(ı \e 17 portları mullan alınımışlıları l bılıık gniş portları <i \e lı, X7-X \ l'ii; isıını l' l'c lı çıkışları mullan alınımışlıları Yine asın şonj' anını l'oil . " alkişlan. X2l i yongasını l . mılrol lınek H.m kullarılmışlıları KS! i2 \ı; l' l' s'gual se\ iM. lııuu bıbuı ine donuşlm inek iyin iki lane lıansislor kullanil mişEri l i. in si. m T l. KS. ' i. ' sil' ilııuu l' l' sı \ı u- sınıe donu; t m m A\ m şı kılık lıansislor l' 2. X/ IS yongası mu l' l. l. , ıkıştıu KS .M 2 ses iscsine donuüşlıuu m

Kail u/c111ule ayrı ıra konini lıala iş. ü. 11 )d->) bulunmakla; lıı Kaila balalı konini gönderiliııuc bu işık >anaı \e kullanıu\_ıvı ika/, edeı

### 3. KOMUİL.AH

(ıenel gniş-çıkış kabil kon tul lanın AS( lı olaiak bađlı olduđu bilgisayarın KS2^2 sen portları mullan .ılı lı lı u m il lı lı kısa \e o/olup konuluılarda lie\ad\ imal (k< lalı,ıı) sa> sistemi kullarılmışlılı Ki muıl.ıi soçılınken. olonası\oıul, kullarılabılı lı lı. kumul yeşille i ine ö/el lıkke oncıı vci il inışı lı Kııııull.nd.ı 'tın .ı.ılır (rııııı^c-ılııııı) karaklıcı lcı l. ullanılabilılı lıkal bu karakleilcı kail laralından bu oııııı^ alınıma/



Şekil 2 (icıu-l (ıııı) <ıııı> Kılıı lıH\ it-şınnaM



Genci giriş-çıkış kartının kumullarını 3 bölümde incelemek mümkündür:

- i. (ikiş kumulları
- ii. Giriş komutları
- iii. Genel kumullar

Şimdi kısaca bu komutlara bir göz atalım.

### 3.1 ÇIKIŞ KOMUTLARI

Bu komutlar, genel giriş-çıkış kartı yoluyla bilgisayardan dışarıya kontrol sinyali göndermek için yaratılmışlardır. Mevcut çıkış komutları Tablo I de özetlenmiştir. Burada Port A yerine Port B de kullanılabilir.

A=nn	A<- nn
A=&nn	A<- A AND. nn
A=!nn	A<- A OR. nn
A=Xnn	A<- A XOR. m
A-Qnn	A<- A XiXNOR. m
A=.	A<- A
A=1	A<- AH 1
A=-	A<- A-1
A=/	A<- A/2
A=*	A<- A*2

Tablo I. (ikiş Kumulları

A=nn

Bu komul, kartın 8 bitlik A (veya B) portuna hexadecimal nn sayısını gönderir. Örneğin, A=FF komutu Port A'nın bütün çıkışlarını lojik 1 yapar.

A=&nn

Bu komul, Port A da bulunan 8 bitlik bilginin hexadecimal nn sayısı ile lojik AND mı alır ve çıkan neticeyi yine Port A ya gönderir. Bu komutun esas kullanılışı, bilhassa kontrol uygulamalarında belirli bir veya birden fazla çıkışı lojik 0 yapmak içindir.

A=!nn

Bu komul, Port A da bulunan 8 bitlik bilginin hexadecimal nn sayısı ile lojik OR mı alır ve çıkan neticeyi yine Port A ya gönderir. Bu komutun esas kullanılışı belirli bir veya birden fazla çıkışı lojik 1 yapmak içindir.

A=Xnn

Bu komul, Port A da bulunan X bitlik bilginin hexadecimal mı sayısı ile lojik XOR mı alır ve çıkan neticeyi yine Port A ya gönderir.

A-Qnn

Bu komul, A=Xnn komutuna benzer fakat burada KXiXNOR-NOU alır.

A=.

Bu komul, Port A da bulunan 8 bitlik bilgiyi lojik ters (complement) yapar. Örneğin, Port A da 10 gibi 8 bitlik bir bilgi olduğunu kabul edersek, A=. komulu Port A daki bilgiyi 10 inin tersi olan EF yapar.

A=+

Bu komul, Port A da bulunan 8 bitlik bilgiyi 1 artırır. Örneğin, Port A da 2F gibi bir bilgi olduğunu kabul edersek, A=+ komulu Port A daki bilgiyi 30 yapar.

A=-

Bu komul A=+ komutuna ben/er fakat burada bilgi 1 olarak eksiltir.

A=/

Bu komul A=+ komuluna ben/er fakat burada bilgi 1 basamak sağa kaydırılır (2 ye bölünür) ve soldaki bit 0 ile doldurulur.

A=\*

Bu komul A=/ komutuna benzer fakat burada bilgi 1 basamak sola kaydırılır (2 ile çarpılır) ve sağdaki bit 0 ile doldurulur.

### 3.2 GİRİŞ KOMUTLARI

Bu komutlar, genel giriş-çıkış kartı yolu ile bilgisayara bilgi aktarırlar. Mevcut giriş komutları Tablo 2 de özetlenmiştir. Burada Port V yerine Port D, li, F, ü ve I i kullanılabilir.

<b>I-C</b>	1 < C
<b>1-&amp;11C</b>	1 <.. C AND. II
<b>1-.C</b>	1 <- C

**Tablo 2. (iri\$ komutları**

### 1=C

Bu komut, kartın 4 bitlik C portundan hlight ile\adccimal sayı okur ve bilgisayara seri ASCII olarak gönderir.

### 1=&C

Bu komut, kartın 4 bitlik C portundan 1 digit hexadecimal sayı okur, bu sayının lıc\adecimal mın sayısını ile lojik AND) ını alır ve neticeyi seri olarak bilgisayara gönderir.

### 1=.C

Bu komut, kartın 4 bitlik C portundan bir sayı okur ve bu sayının tersini (complement) alıp neticeyi bilgisayara gönderir.

## 3.3 GENEL KOMUTLAR

Bu komutlar daha genel amaç için yaratılmış olup komut listesi Tablo 3 de verilmiştir.

L	A <- 0	B <- 0
H	A <- 1	B <- 1
GA	1 <- A	
GB	1 <- B	
T	1 <- Versiyon No	
C	G <- Clock	
R	Reset	

**Tablo 3. Genel Komutlar**

Komut L, bütün çıkışları (Port A ve Port B) lojik 0 yapar. Aynı şekilde, komut H bütün çıkışları lojik 1 yapar.

Komut GA, çıkış portu olan Port A daki mevcut bilgiyi okuyup bilgisayara aktarır. Aynı şekilde, komut GB Port B deki bilgiyi bilgisayara aktarır.

Komut T, genel giriş-çıkış kartının versiyon numarasını bilgisayara gönderir

Komut C, normalde 1 bitlik giriş portu olan Port 1 yi 2MHz/ kare dalga üreten çıkış portu yapar. Bu port clock olarak kullanılabilir

Komut R, bir önceden hatalı olarak verilen komutları dolayı kart ücünde mevcut olan konileri Ha ika/ ışığını (LHD) sönlendirip kaili yeni konileri almak için ilii/enleir

ELEKTRİK MÜHENDİSLİĞİ 5. ULUSAL KONGRESİ

## 4. ÖRNEK PROGRAM

Genel giriş-çıkış kartı IBM uyumlu kişisel bir bilgisayara bağlanmış ise, aşağıda verilen basit BASIC programı ile karta çıkış komutları gönderilebilir:

```

10 OPEN 'COM1:9600,N,8,2,RS,CS,DS' as ti I
20 DTS = " "
30 \VUHLI DTS<> ""
40 PRINT
50 INPUT "Gönderilecek Bilgi (00-FI-)";I3TSi
60 INPUT "Hangi Porta (A veya B)";PORTS$
70 PRINT « 1,PORTS;"-",DTS
80 WEND
90 CLOSE U 1

```

## 5. SONUÇ

Bu çalışmamızda bahsetmiş olduğumuz genel giriş-çıkış kartının bilgisayar destekli kontrol eğitiminde ve aynı zamanda endüstride birçok otomatik kontrol uygulamalarında kullanım alanları mevcuttur. Kartın en önemli özelliği herhangi bir bilgisayar ile kullanılabilir olması ve aynı zamanda oldukça ucuz olmasıdır. Kart, seri olarak bağlanmış olduğu bilgisayardan aldığı komutlar ile bilgisayara giriş ve çıkış portları sağlar.

Kartı daha kullanışlı bir hale getirmek için, daha ileri bir çalışma olarak kari ü/eiine dijital çevirici (D/A) ve analog çevirici (A/D) devreleri de ilave edilebilir.

## 6. KAYNAKLAR

- 1/. Embedded Controller Handbook, Volume 1, INTEL Corporation, Santa Clara, 1988
- 12/. D. İbrahim, Universal I/O Processor, Electronic Product Design, August 1989



Dr Doğan İbrahim 1954 yılında Lelkoş'a'da doğdu ve Lise öğrenimini Lefkoş'a'da tamamladıktan sonra 1975 yılında Salford Üniversitesi Elektronik Mühendisliği bölümünden mezun oldu. Daha sonra 1977 yılında Manchesler Üniversitesinde Olomalik Kontrol dalında yüksek lisans ve 1981

yılında (Uy Ütüveri şilesinden Doktora diplomaları aldı. Dr Doğan İbrahim İngiltere'de çeşitli kuruluşlarda u/un yıllar çalıştı ve 'Chartered Engineer' oldu. Halen Yakın Doğu Üniversitesi Uilgisayarı Bolüm Başkan olan Dr İbrahim'in ilgi alanları lobotik ve bilgisayar ileleclikli olomalik konuloktur





i	2 <sup>i</sup>	Z(i)
2	4	2
3	8	2
4	16	4
5	32	7
6	64	12
7	128	16
8	256	32
9	512	64
10	1024	128
11	2048	256
12	4096	512
13	8192	1024
14	16384	2048
15	32768	4096
16	65536	8192

Tablo-1. Değişken sayısına bağlı olarak Z(i)'nin değerleri

	i=2	i = 3	i = 4	i = 5	i = 6	i = 7
1	0	0	0	0	0	0
2	3	7	7	7	7	15
3			8	13	13	19
4			15	15	17	28
5				17	20	37
6				20	26	42
7				26	37	54
8					43	57
9					46	70
10					50	73
11					56	85
12					63	90
13						99
14						108
15						112
16						127

Tablo-2. Z(i) tane mintermi belirleyen sayılar

No	Giriş deę. sayısı x Çıkış fonk. sayısı	Min./Maks. minterm	PT7 T(s)	Açıklama
1	4x1	14	4/0.00	0,15 hariç bütün mint.
2	5x1	30	5/0.06	0,31 hariç bütün mint.
3	6x1	62	6/0.39	0,63 hariç bütün mint.
4	7x1	126	7/1.59	0,127 hariç bütün mint.
5	4x3	10/10	11/0.06	
6	4x5	10/10	15/0.00	
7	11x8	20/22	45/0.06	
8	11x8	48/50	98/0.22	
9	11x8	118/120	343/3.85	
<b>10</b>	15x15	20/24	49/0.05	
11	15x15	48/50	100/0.22	
12	15x15	118/120	416/3.90	

Tablo-3. İndirgeme sonuçları. Burada, Min./Maks. fonksiyondaki minterm sayısının alt ve üst sınırlarını, PT/T(s) indirgenmiş ifadedeki terim sayısını ve indirgeme süresini göstermektedir.

Test bilgisayarı 386 SX-25







ULTRA KALİTE İLE  
ULTRA GARANTİLİ ÜRETİM YAPMAK  
VE ULTRA DESTEK VERMEK...

2000'li yıllar için !,



yıl ücretsiz tamir, bakım ve ömür boyu sürekli yedek parça ve servis güvencesi.

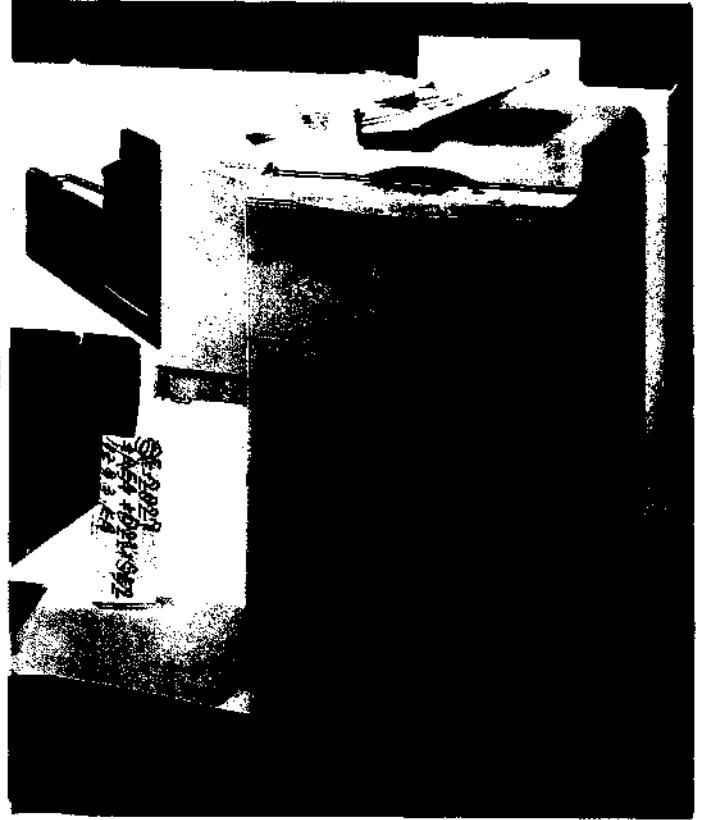
**ULTRA**  
*COMPUTER*



BİLGİ  
ELEKTRONİK  
TİC. VESAN. A.Ş.



# SHARP



## KAREL

**BÜRO MAKİNELERİ TİC. LTD.ŞTİ.**

Mrk : İki Nolu Gazipaşa Cad. no: 6/1-2

Tel. : (03)216740-260144 Fac 211211

Şube : Gazipaşa Cad. No: 10/A

: (03) 222983 Fax: (03) 211211

Tel. **61100 TRABZON**

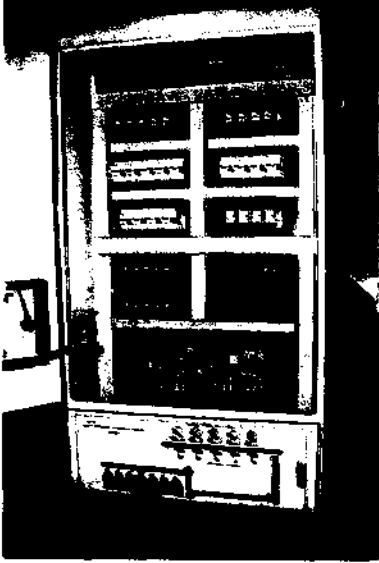
: Peüt Meydanı Cennet Çeşme Sok.

Şube No: 4 (Bakırcılar cami karşısı)

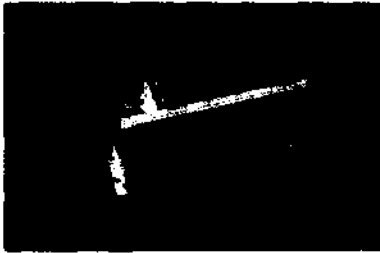
: (011)82078 Fax: 82078

Tel. **25200 ERZURUM**

# ELEKTRONİK SANAYİİNDE BURSA'DA GÜÇLÜ KURULUŞ



- Otomasyon ve Process Kontrol
  - AC/DC Motor Hız Kontrolü
- SÜREKLİ BAKIM,  
GÜÇLÜ HİZMET KADROSU.**



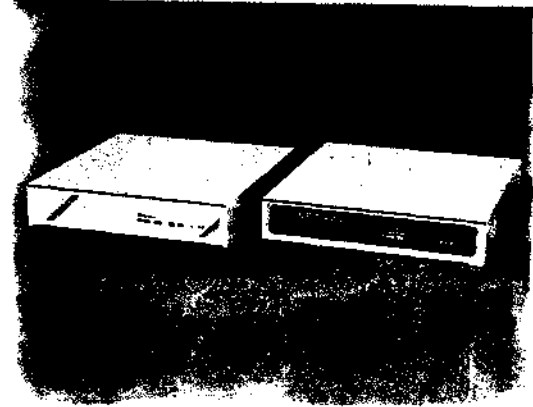
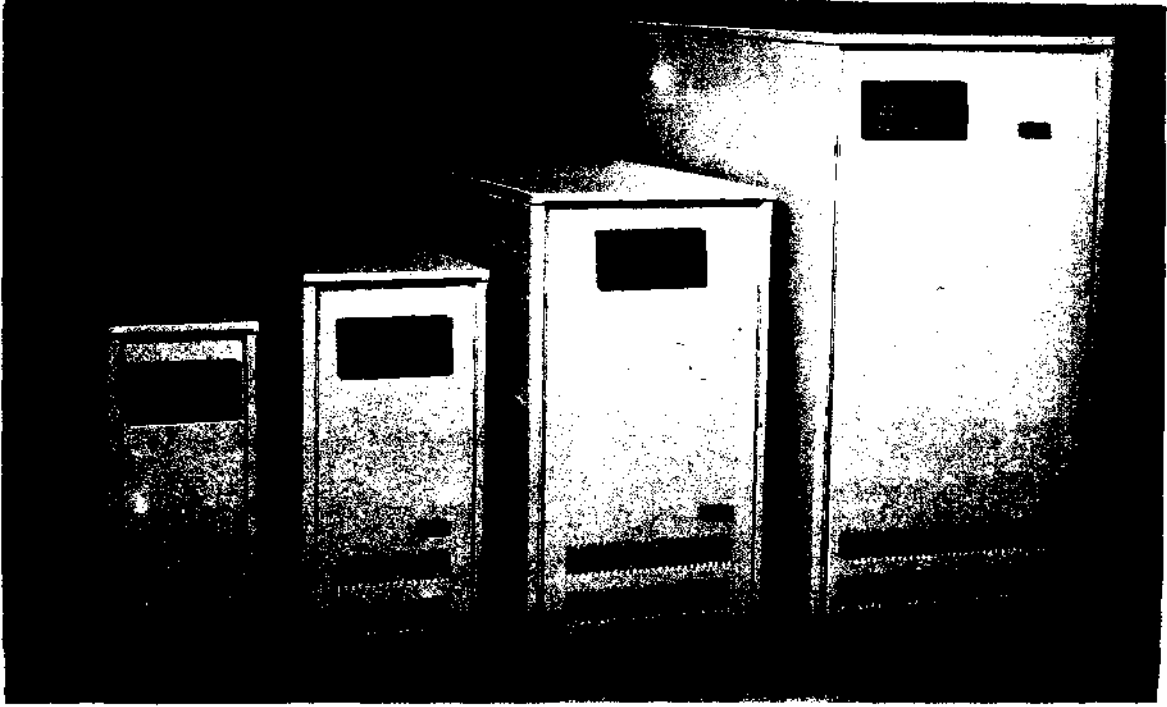
## ÜRETİM KONULARIMIZ :

- DC MOTOR HIZ KONTROL ÜNİTELERİ
- © AC MOTOR YOL VERİCİ
- © AC MOTOR HIZ KONTROL ÜNİTELERİ
- PLC+PC VE MİKROİŞLEMCİ UYGULAMALARI
- AKÜ ŞARJ CİHAZLARI
- © DİGİTAL/ANALOG KONTROL CİHAZLARI
- © İZOLASYON AMPLİFİYER
- HİDROLİK MOTOR VE ORANSAL VANA KONTROL
- © MÜHENDİSLİK VE TAAHHÜT HİZMETLERİ
- ◁ MALZEME SATIŞ

Merkez: Elmasbahçeler Sabun evi Sokak No. 17/A BURSA  
Tel: (224) 260 32 32 (3 Hat) - (224) 261 35 37  
Şube: Ahmet Hamdi Tanpınar Cad. Elektronik Pasajı No: 12  
Tel: (224) 221 51 67

# İNFORM KESİNTİSİZ GÜÇ KAYNAKLARI

500 watt - 30000 Watt arası güçlerde  
Türkiye çapında çalışan 3500 cihazlık güvence



## DiĞER ÜRÜNLERİMİZ

- INFORM SERVO ELEKTRONİK VOLTAJ REGÜLATÖRLERİ  
(1 KVA - 100 KVA Arası)
- İNDÜKSİYON ISITMA - ERİTME SİSTEMLERİ  
(6000 Hz. 200 KW.'a kadar)
- TRİSTÖR KONTROLLÜ KAPLAMA REDRESÖRLERİ  
(10000 Amper'e kadar)

## ÖZDİSAN ELEKTRONİK SANAYİİ

Galata Kulesi Sokak No:32 -34/3 - 36/18 Karaköy/İSTANBUL  
Tel: 243 40 34-251 29 41 - 252 07 14 - 252 08 84 Fax: 244 59 43