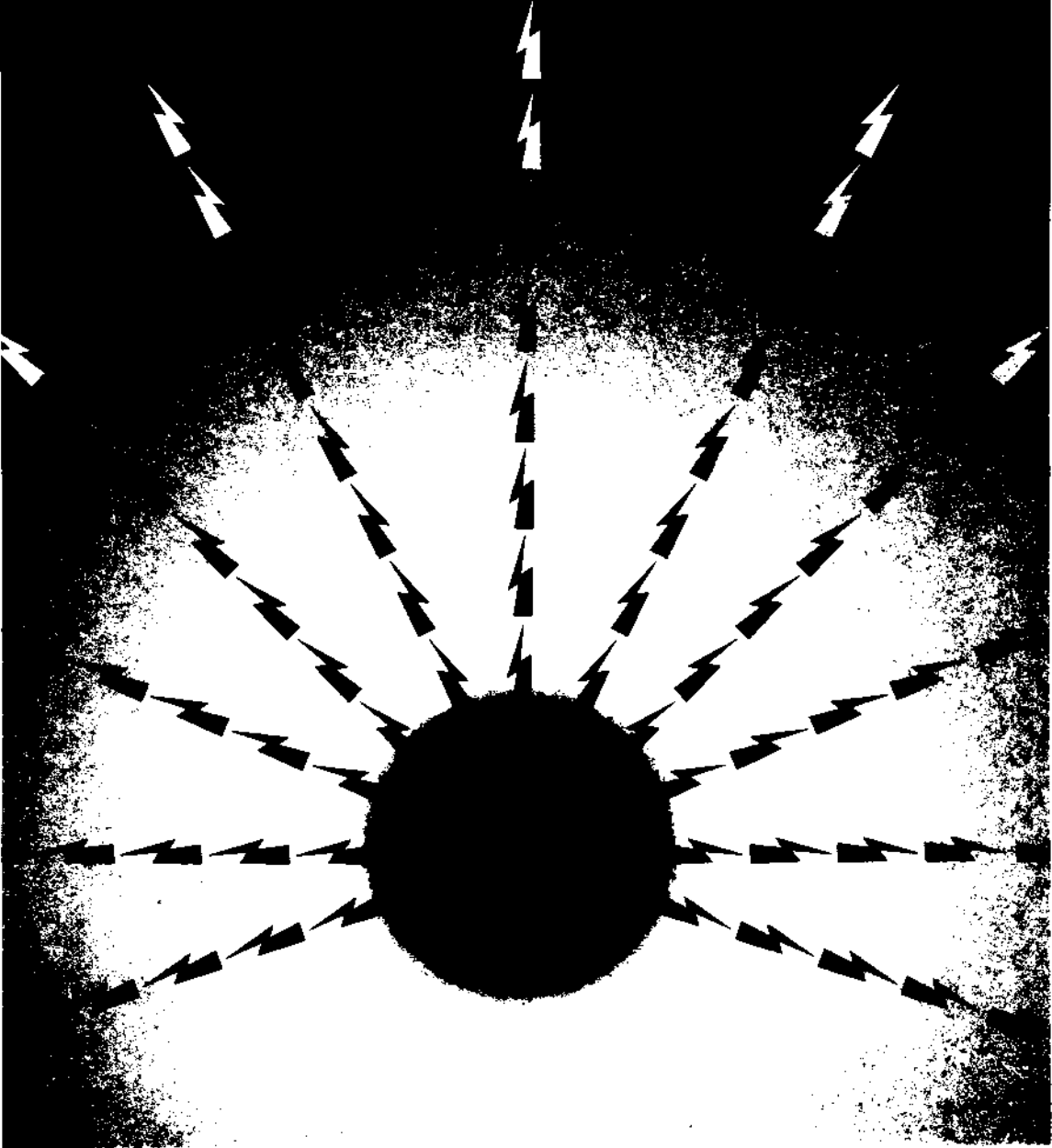


TMMOB ELEKTRİK MÜHENDİSLERİ ODASI

ELEKTRİK - ELEKTRONİK BİLGİSAYAR MÜHENDİSLİĞİ 7. ULUSAL KONGRESİ



TMMOB
ELEKTRİK MÜHENDİSLERİ ODASI
ANKARA ŞUBESİ



ODTÜ
ELEKTRİK -ELEKTRONİK
MÜHENDİSLİĞİ BÖLÜMÜ



TÜBİTAK

ÖNSÖZ

TBMMO Elektrik Mühendisleri Odası Elektrik-Elektronik-Bilgisayar Mühendisliği 7. Ulusal Kongresini ve Sergisini Orta Doğu Teknik Üniversitesi'nde gerçekleştirmiş olmaktan onur ve sevinç duymaktayız. Üniversite olarak kongreye ikinci kez evsahipliği yapmamız bizi fazlasıyla mutlu etmiştir, ama mutluluğumuz asıl geçen süre içinde Odamızın, meslek yaşamımızın ve Üniversitemizin ne kadar gelişmiş olduğunu gözlemekten kaynaklanmaktadır.

Gerçekten de ilgi alanlarımızın çeşitlenmesi, bu alanlarda belh bir beceriye ulaşılmış olması, eskiden güçlü olduğumuz dallarda da gücümüzün sürmesi Elektrik-Elektronik ve Bilgisayar Mühendislerimizin ülke genelinde giderek daha fazla söz sahibi olmaları olgusunu yaratmaktadır. Bireysel basanlarımızın kurumlarımızı da ülke ekonomisi ve gelişmesi bakımından güçlendirmekte olduğu açıktır. Nitekim bu sektörlerde faaliyet gösteren kuruluş sayısı hızla artmaktadır. Bu sayısal gelişmenin nitelik bakımından da aynı hızla sürdüğünü görmek sevindiricidir. Kongremiz ve sergimiz bunun en somut kanıtını oluşturmaktadır.

2000Mİ yılların Türkiye'sinin ihtiyaçlarını yakahyabilmek için daha çok şeyler yapılması gerekmektedir. Endüstri-Eğitim Kurumları ve Meslek Odaları arasındaki iletişim ve karşılıklı etkileşimi güçlendirmek gerekmektedir. Bu geçmişe oranla daha sevindirici bir düzeyde sürüyor da olsa henüz gelişmiş ülkelerdeki başarılı örneklerin uzağındadır. Önümüzdeki yıllarda bu konuda daha fazla çabaya ihtiyaç vardır.

Tüm katılımcılara Kongre ve Sergimize vermiş oldukları güç için teşekkür ediyorum. Sizleri Üniversitemizde görmeyi kıvançla selamlıyor saygılarımı sunuyorum.

Prof. Dr. Fatik Canatan
Yürütme Kurulu Başkanı

ELEKTRİK-ELEKTRONİK-BİLGİSAYAR MÜHENDİSLİĞİ
7. ULUSAL KONGRESİ

YÜRÜTME KURULU

Fatih CANATAN (Başkan, ODTÜ)

M. Mete BULUT (ODTÜ)
Cengiz BEŞİKÇİ (ODTÜ)
Gönül SAYAN (ODTÜ)
Cemil ARIKAN (TÜBİTAK)
M. Hacim KAMOY (ASELSAN)
Hüseyin ARABUL (BARMEK)
Aydın GÜRPINAR (ENERSİS)

M. Asım RASAN (EMO)
Cengiz GÖLTAŞ (EMO)
H. Ali YİĞİT (EMO)
Kubilay ÖZBEK (EMO)
M. Sıtkı Çiğdem (EMO)
Funda BAŞARAN (EMO)
Mustafa ÖZTÜRK (EMO)

EDİTÖRLER

Fatih CANATAN

Mehmet Mete BULUT

BÜYÜK ÖLÇEKLİ BİLGİSAYAR AĞLARINDA YÖNLENDİRME KONTROLÜ TASARIMI

Aydın AYBAR
Osmangazi Üniversitesi
Elektrik-Elektronik Mühendisliği Bölümü
ESKİŞEHİR
aydin@mmf.mm.anadolu.edu.tr

Altuğ İFTAR
Anadolu Üniversitesi
Elektrik-Elektronik Mühendisliği Bölümü
ESKİŞEHİR
altug@mmf.mm.,,anadolu.edu.tr

ABSTRACT:

A decentralized dynamic routing controller algorithm is developed for large-scale computer networks. Most important step of this algorithm is calculating a feedback matrix. The feedback matrix calculation is explained in this paper.

I. GİRİŞ

Bilgisayar ağları, bilgisayarların birbirleriyle bilgi alış-verişi yapması için oluşturulmuşlardır. Bir bilgisayar ağı birbirlerine linklerle bağlanmış nodlardan oluşur. Ağ dışından (kullanıcılardan) ve/veya ağ içindeki diğer nodlardan mesaj alarak, bu mesajları erişmeleri gereken nodlara veya bu nodlara eriştirilmek üzere ağ içindeki diğer nodlara aktarabilen her bilgisayar veya bilgisayar gurubu bir nodu oluşturur. Bu nodlar arasında mesaj ileten bağlantılar ise link olarak adlandırılır.

Bilgisayar ağlardaki en önemli problem yönlendirme problemidir. Mesajların ağa girdikleri noda 'kaynak nodu', erişmek istedikleri noda 'varış nodu' denir. Yönlendirme problemi, bir mesajın kaynak nodundan varış noduna hangi linkler ve nodlar üzerinden iletileceğinin belirlenmesidir. Yönlendirme problemini çözmek için çeşitli algoritmalar önerilmiştir. Daha önce yapılan çalışmalar ve önerilen yönlendirme algoritmalarının genel bir özeti [1]'de verilmiştir.

Daha önce yapılan çalışmalardan [2]'de örtüşen kısmında bir tek nod olan ve iki altağdan meydana gelen bilgisayar ağları için öncelikle içerme prensibi uygulanmış ve bu prensipden yararlanılarak bu topolojiye sahip bilgisayar ağları için dışmerkezli dinamik bir yönlendirme kontrolü tasarlanmıştır. Burada içerme prensibi, Ikeda ve Şiljak [3] tarafından ortaya atılan geliştirilmiş sistem ile özgün sistem arasındaki ilişkilerden biridir. İçerme prensibi, bilgisayar ağları için ortaya konularak, geliştirilmiş bilgisayar ağının, özgün bilgisayar ağını içerdiği bulunmuştur. Bu ilişkiden yararlanılarak, özgün bilgisayar ağı için yönlendirme kontrolü tasarlanmıştır [2]. Ataşlar ve İftar [4] tarafından yapılan çalışmada ise ikiden fazla altağa sahip ve/veya örtüşen kısmında birden fazla nod bulunan büyük ölçekli bilgisayar ağları için bir dışmerkezli dinamik yönlendirme algoritması geliştirilmiştir. [4]'de önerilen yönlendirme stratejisi her bir noddan diğer

nodlara bilgi alış-verişi gerektirmeksizin yapılabilmesi açısından dışmerkezlidir. Ancak bu yönlendirme kontrolü tasarımı, her bir mesaj için mesajın kaynak ve varış nodunun yanı sıra, mesajın bulunduğu altağdaki yerel kaynak ve yerel varış nodu ile bulunduğu altağdan yönlendirileceği bir sonraki altağın bilinmesi gerekir. Bu algoritmada yapılan işlem sayısı ve her bir mesaj için bilinmesi gereken bilgi miktarı fazladır.

Bu çalışmada, büyük ölçekli bilgisayar ağları için daha az hesaplama, gerektiren ve her bir mesaj için bilinmesi gereken bilgi miktarının daha az olduğu bir yönlendirme kontrolü tasarlanmaya çalışılmıştır.

Kullanılan Notasyon:

z : Geliştirilmiş ağdaki z noduna karşılık gelen ortak nodu,
 y : Geliştirilmiş ağdaki y noduna karşılık gelen ortak nodu,
 i : Özgün ağdaki i noduna karşılık gelen ayrıştırılmış nodu,
 L : Mesajın kaynak nodunun bulunduğu altağı,
 K : Mesajın varış nodunun bulunduğu altağı,
 S_r : Özgün bilgisayar ağının S_r altağındaki nodların oluşturduğu kümeyi,
 $f(wx)$: Geliştirilmiş bilgisayar ağında, X altağına gönderilmek üzere, W altağı içindeki varış nodu z olan mesajların, y nodundan W altağına giriş oranını [4],
 $G(j)$: j nodunun elemanı olduğu altağların indislerinin oluşturduğu kümeyi,
 $U_{s_i}(i)$: S_i altağında, i nodunun akış-yukarı nodlarının oluşturduğu kümeyi,
 $V_{s_i}(i)$: S_i altağında, i nodunun akış-aşağı nodlarının oluşturduğu kümeyi,
 $D(S_i)$: Geliştirilmiş bilgisayar ağında, S_i altağının akış-aşağı altağlarının oluşturduğu kümeyi,
 $D(i)$: Özgün bilgisayar ağında i nodunun akış-aşağı nodlarının oluşturduğu kümeyi,
 $U(S_k)$: Geliştirilmiş bilgisayar ağında, S_k altağının akış-yukarı altağlarının oluşturduğu kümeyi,
 $L(S_r, X)$: Geliştirilmiş bilgisayar ağında X altağından gelen mesajın, S_r altağına giriş yapabileceği ayrıştırılmış nodların sayısını,
 n_i : özgün bilgisayar ağında i nodunun akış-aşağı nodlarının sayısını,
 $A(S, X)$: Geliştirilmiş bilgisayar ağında, X altağının

dan gelen mesajın, 5 altağma giriş yapabileceği nodların kümesini.

$\langle \Delta \rangle (E, Y)$: Genleştirilmiş bilgisayar ağında, 5 altağmdan Y altagina giden mesajların çıkış yapabileceği nodların kümesini,

$eritti$: S_r altağında, 5j. altağma gönderilmek istenen mesajlar için i nodunun yerel varış nodu olma olasılığını (yaklaşık olarak),

$$\hat{\gamma}_r(i) \triangleq \frac{f_{(S_r, S_k)}^i}{\sum_{V \in \mathcal{I}(S_r)} \sum_{b \in \mathcal{L}(S_r, V)} \sum_{h \in \mathcal{F}(S_r, S_k)} f_{(S_r, S_k)}^h} b$$

$\gamma_r(0)$: S_r altağında olduğu kabul edilen i nodundaki bir mesajın, yerel varış nodunun i nodu olma olasılığı (yaklaşık olarak),

$$\gamma_r(i) \triangleq \frac{\theta_{(S_r)}^{y,i}}{\sum_{j \in \mathcal{I}(S_r, i)} \sum_{Y \in \mathcal{P}(S_r)} \sum_{z \in \mathcal{F}(S_r, Y)} \theta_{(S_r)}^{y,z}}$$

$Arf(U)$: Yerel varış nodu i nodu olan mesajın, S_r akağından S_k akağına gönderilme olasılığını (yaklaşık olarak),

$$\Delta_{r,k}(i) \triangleq \frac{\sigma_r^k(i) \theta_{S_r, S_k}^{L,K}}{\sum_{S_i \in \mathcal{P}(S_r)} \sigma_r^i(i) \theta_{S_r, S_i}^{L,K}}$$

$P^{\wedge}(i)$: i nodundaki bir mesajın S_r altağında bulunma olasılığını (yaklaşık olarak),

$$P_S^{\wedge}(i) \triangleq \frac{\sum_{S \in \mathcal{I}(S_r, i)} \theta_{S, S_r}^{L,K}}{\sum_{S \in \mathcal{I}(S_r)} \theta_{S, S_r}^{L,K}}$$

göstermektedir.

II. BİLGİSAYAR AĞININ DİNAMİK MODELİ ve YÖNLENDİRME KONTROLÜ

Bu çalışmada, N tane noddan oluşan bir bilgisayar ağı için İftar ve Davison [5] tarafından geliştirilmiş olan ağ dinamiği modeli ve kontrol stratejisi kullanılmıştır. Bu N tane nod, birbirlerine linklerle (linkler tek yönlüdür) mesaj gönderebilecek şekilde bağlanmıştır. Mesaj varış noduna gelir gelmez ağdan ayrılır. Mesajlar varış noduna gidebilmek için, kaynak nodunda ve daha sonra ulaştıkları ara nodlarda sıraya girerler ve buldukları nodun alt nodlarına gönderilirler. Bir mesajın herhangi bir noddan bir alt noda gönderilmesi sırasında işlenmesinden kaynaklanan bir gecikme olur. Geliştirilen ağ dinamiği modeli işlem gecikmelerini kapsayacak şekilde tasarlanmıştır (bkz. [5]).

Bilgisayar ağının dinamik denklemleri, t zaman değişkenini göstermek üzere, her bir i, I, k ($i \wedge k$ ve $/ - \mathcal{L} k$) için aşağıdaki şekilde yazılabilir:

$$q_i^{lk}(t+1) = q_i^{lk}(t) + g_i^{lk}(t) - \sum_{j \in \mathcal{P}(i)} u_{ij}^{lk}(t) \quad (1a)$$

$$m_i^{lk}(t+1) = m_i^{lk}(t) + g_i^{lk}(t - r_i^{lk}) - \sum_{j \in \mathcal{P}(i)} u_{ij}^{lk}(t) \quad (16)$$

$$g_i^{lk}(t) = \delta_{i,l} f_i^k(t) + \sum_{j \in \mathcal{I}(i)} u_{ij}^{lk}(t) \quad (1e)$$

Burada, $\delta_{ij} = \begin{cases} 1, & i = l \text{ ise} \\ 0, & i \neq l \text{ ise} \end{cases}$, q_i^k , i nodundaki kaynak nodu $/$, varış nodu k olan mesajların toplam sıra uzunluğunu, m_i^{lk} , i nodunda işlenmiş, sıra bekleyen, kaynak nodu $/$, varış nodu k olan mesajların sıra uzunluğunu, ff varış nodu k olan mesajların, i nodundan ağa giriş oranını, u_{ij}^{lk} , i nodundan j noduna gönderilen, kaynak nodu $/$, varış nodu k olan mesajların $1-j$ linki boyunca akış oranını, g_i^k , ağ dışından ve/veya i nodunun akış-yukarı nodlarından i noduna giriş yapan, kaynak nodu $/$, varış nodu k olan mesajların toplam akış oranlarını, $P(i)$ ve $U(i)$ ise i nodunun, sırasıyla, akış-aşağı ve akış-yukarı nodlarını göstermektedir, $\mathcal{C} \triangleq \{ \langle \mathcal{C}^* \rangle \}$ $N(N-1)^2$ boyutlu, toplam sıra uzunluğu vektörünü, $m \triangleq \{ m_i^k \}$, $N(N-1)^2$ boyutlu, işlenmiş mesajların sıra uzunluğu vektörünü, $/ \triangleq \{ /, * \}$, $N(N-1)$ boyutlu, dışarıdan noda giren mesajların akış oranı vektörünü, $u \triangleq \{ u_i^j \}$, $L(N-1)^2$ boyutlu, linkler boyunca mesaj akış oranları vektörünü ve $g \triangleq \{ g_i^k \}$, $N(N-1)^2$ boyutlu, nodlara giren mesajların toplam akış oranları vektörünü göstermektedir.

Bilgisayar ağına uygulanacak yönlendirme kontrollerini ise aşağıdaki gibi gösterebiliriz [5]:

$$v(t) = O[m(t) + \sum_{r=1}^{r_{max}} T_r D_r g(t-r)] \quad (2)$$

$$u(t) = T(t)v(t) \quad (3)$$

Burada $T(t)$, $u_{i,r}^{lk}$ 'ye karşılık gelen elemanı

$$T_{ij}(t) = \min \left(1, \frac{\hat{c}_{ij} - j}{\sum_{k \in \mathcal{N}^*} \sum_{l \in \mathcal{N}^k} v_{lj}^k(t)} \right)$$

şeklinde olan bir köşegen matrisi, D_r ($r = 1, 2, \dots, r_{max}$), $T = r^k$ ise m_{ij}^{lk} 'ye karşılık gelen köşegen elemanı 1 olan, diğer tüm elemanları 0 olan bir köşegen matrisi ve c_{ij} ise $i-j$ linkinin kapasitesini göstermektedir. Önerilen kontrol stratejisinde, tek tasarım parametresi 0 ile gösterilen geri besleme matrisidir, istenilen kontrolün sağlanması için bu matrisin aşağıda verilmiş olan şartları sağlaması gerekmektedir [5].

- Ş1) 0'nın tüm elemanları negatif olmayan gerçek sayılar olmalıdır.
- Ş2) 0_{ij}^{lk} , \mathcal{C} 'nin elemanlarını göstermek üzere $u = Qq$ ise $j \in \mathcal{L}(i)$, $u_{ij}^{lk} = 9^k q^k$ şeklinde olmalıdır. Bir başka deyişle, 6 dışmerkezli bir geri besleme matrisi olarak elde edilmelidir.
- Ş3) $\sum_{j \in \mathcal{P}(i)} \theta_{ij}^{lk} = 1$ olmalıdır.

Ş4) Her bir I için, her bir j_0 'dan her bir f 'ya $(\hat{c} \wedge \{i_0, l\}) \in \mathcal{J}_{i_0, i_1, \dots, i_m}^f > 0$ olmak üzere $i_0, i_1, \dots, i_m = k$ şeklinde bir yol olması gerekir.

III. ÖRTÜŞMELİ AYRIŞTIRMA ve GENLEŞTİRME

Büyük ölçekli bilgisayar ağlarında merkezi yönlendirme algoritması uygulamak zor, hatta çoğu zaman imkansız olmaktadır. Tasarlanan yönlendirme kontrolündeki geri besleme matrisinin boyutu $(0 : L(N - 1)^2 \times N(N - 1)^2)$, bilgisayar ağındaki nod ve link sayısına bağlıdır. Bilgisayar ağındaki nod ve link sayısı arttıkça geri besleme matrisini hesaplamak güçleşmektedir. Bundan dolayı büyük ölçekli bilgisayar ağları için yönlendirme kontrolünü dışmerkezli olarak tasarlamak merkezi olarak tasarlamaktan daha kolay olabilmektedir. Yönlendirme kontrolünün dışmerkezli olarak tasarlanabilmesi için uygulanabilecek bir yaklaşım verilen bilgisayar ağını örtüşmeli olarak ayrıştırmaktır. Örtüşmeli olarak ayrıştırılan bilgisayar ağına, genleştirme [6] uygulanarak örtüşmeli altağların ayrışık olarak elde edilmesi mümkündür. Genleştirilen bilgisayar ağında, altağlar birbirinden ayrışık olarak ortaya çıkacağından, ayrışık her bir altağ için yerel olarak bir geri besleme matrisi bulunabilir ve her bir altağ için birer yönlendirme kontrolü tasarlanabilir. Özgün bilgisayar ağının (ele alman ağ) yönlendirme kontrolü de, ayrışık olarak ortaya çıkan altağların geri besleme matrislerinden yararlanılarak hesaplanabilir [2].

Büyük ölçekli bilgisayar ağlarının, örtüşen altağların belirlemek için bugüne kadar genel bir sistematik yöntem geliştirilememiştir. Ancak belirli topolojiye sahip bilgisayar ağları için belirli çözümler yapılabilmektedir. Büyük ölçekli bilgisayar ağları genelde, belli dinamik parçalar (örtüşen kısım) üzerinden kuvvetli bağlarla, bunlar dışında zayıf bağlarla bağlı noddan oluşmaktadır. Kuvvetli bağlarla bağlı dinamik parçalar, bir veya birkaç noddan meydana gelebilir. Örtüşen kısımda yer alan nod veya nodlar birkaç altağın elemanı olabilir. Büyük ölçekli bilgisayar ağının altağları, ortak nod veya nodlar hariç, birbirleriyle bağlantıları olmayacak şekilde noddan meydana gelmelidirler. Bilgisayar ağının örtüşen kısmında yer alan nod veya nodlar altağların ortak elemanlarıdır. Ayrıca büyük ölçekli bilgisayar ağlarını altağlara ayrıştırırken bilgisayar ağının coğrafi konumu ve fiziksel bağlantıları da göz önünde tutulabilir.

Genleştirilmiş ağ, örtüşen kısımda bulunan noddan ve varsa linklerin örtüşen her bir altağda tekrarlanması sonucu elde edilebilmektedir. Tekrarlanan bu nodlar arasında (bir altağdan diğerine uzanan) sonsuz kapasiteli olduğu varsayılan iki link (her biri bir yönde) olmakta, ayrıca örtüşen kısımdaki nodlarla diğer nodlar arasındaki linkler, bir altağdan diğerine uzanmamak koşuluyla korunmaktadır [7].

IV. BÜYÜK ÖLÇEKLİ BİLGİSAYAR AĞLARININ YÖNLENDİRME KONTROLÜNÜN BULUNMASI

Bu çalışmada, büyük ölçekli bilgisayar ağları için bir yönlendirme kontrolü algoritması geliştirilmeye çalışılmıştır. Önerilen yönlendirme algoritmasının geliştirilme aşamaları:

- Bilgisayar ağına örtüşmeli ayrıştırma uygulanması ve ağın genleştirilmesi.
- Her bir altağ için geri besleme matrislerinin bulunması.
- Ayrışık durumda olan her bir altağın bir nod gibi kabul edilmesi ile oluşan ağ için geri besleme matrisinin bulunması.
- Bulunan geri besleme matrislerinden yararlanılarak, özgün bilgisayar ağı için bir geri besleme matrisinin bulunması.
- Bulunan bu matristen yararlanılarak özgün bilgisayar ağı için yönlendirme kontrolünün bulunması.

şeklinde dir.

Örtüşmeli ayrıştırma uygulanan büyük ölçekli bilgisayar ağı [7]'de anlatıldığı şekilde genleştirilerek, altağlarının ayrışık olarak ortaya çıkması sağlanır. Elde edilen her bir altağ için [5]'de anlatıldığı şekilde bir geri besleme matrisi (\hat{C}_s^o) ve [4]'de anlatıldığı şekilde, ayrışık durumda olan her bir altağın bir nod gibi kabul edilmesi ile oluşan ağ için geri besleme matrisi (\hat{O}) bulunur.

Bilgisayar ağının geri besleme matrisinin (\hat{O}) bulunması:

Bilgisayar ağının \hat{O} geri besleme matrisinin elemanları, \hat{O} , $f(s, s')$ ve \hat{C}_s^o ifadeleri kullanılarak aşağıdaki şekilde bulunmuştur.

i) Mesajın, kaynak ve varış nodu aynı altağda tanımlanmış ise ($i, k \in S_j$),

$$a) \quad \hat{O}_{ij}^{tk} = \begin{cases} \theta_{(S_j^o)}^{tk} & j \in S_j^o \text{ ise} \\ 0 & j \notin S_j^o \text{ ise} \end{cases}$$

$$b) \quad \hat{O}_{ij}^{tk} = \delta_{ij}^k = 1, \quad \forall j \in \hat{P}(i)$$

ii) Mesajın kaynak ve varış nodu farklı altağlarda tanımlanmış ise ($i \in S_1^o, j \in S_2^o$ ve $k \in S_2^o, k < l \in S_1^o$),

$$a) \quad \hat{O}_{ij}^{tk} = \begin{cases} \sum_{x \in \hat{P}(i)} \theta_{(S_1^o)}^{tx} & \dots \\ 0 & j \notin S_1^o \text{ ise} \end{cases}$$

$$b) \quad \hat{O}_{ij}^{tk} = \delta_{ij}^k, \quad i \in S_1^o,$$

b1) i nodu ortak nod değilse,

$$\hat{\theta}_{S_1, Y}^{S_1, S_2} = \mathbf{E}_{Y \in \hat{D}(S_1)} \mathbf{E}_{z \in \Phi(S_1, Y) \setminus \{i\}} \frac{f_{(S_1, Y)}^z}{\sum_{h \in \Phi(S_1, Y) \setminus \{i\}} f_{(S_1, Y)}^h} \theta_{(S_1^*)_{ij}}^{iz}, \forall j \in \hat{V}(i)$$

b2) i nodu ortak nod ise,

$$\hat{\theta}_{S_1, Y}^{S_1, S_2} = \mathbf{E}_{Y \in \hat{D}(S_1)} \mathbf{E}_{z \in \Phi(S_1, Y) \setminus \{i\}} \left[\theta_{S_1, Y}^{S_1, S_2} (1 - \gamma_1(0)) - \frac{f_{(S_1, Y)}^z}{\sum_{h \in \Phi(S_1, Y) \setminus \{i\}} f_{(S_1, Y)}^h} \theta_{(S_1^*)_{ij}}^{iz} \right] + \mathbf{E}_{\alpha \in G(i) \cap G(j) \setminus \{1\}} \mathbf{E}_{Y \in \hat{D}(S_\alpha)} \mathbf{E}_{z \in \Phi(S_\alpha, Y) \setminus \{i\}} \left[\theta_{S_\alpha, Y}^{S_1, S_2} \gamma_1(i) \Delta_{1\alpha}(i) - \frac{f_{(S_\alpha, Y)}^z}{\sum_{h \in \Phi(S_\alpha, Y) \setminus \{i\}} f_{(S_\alpha, Y)}^h} \theta_{(S_\alpha^*)_{ij}}^{iz} \right], \forall i \in \hat{D}(z)$$

Burada, $\hat{\theta}_{S_\alpha, Y}^{S_1, S_2} = \left[\begin{matrix} \cdot \\ \cdot \\ \cdot \end{matrix} \right]$, $\forall \alpha \in G(i) \cap G(j)$ ise diğer durumlarda

c) $i \notin \hat{S}_j$ ve $i \in \hat{S}_j$ olmak üzere,

e1) i nodu ortak nod değilse ($i \notin \hat{S}_j$),

$$\hat{\theta}_{S_3, Y}^{S_3, S_2} = \mathbf{E}_{X \in \hat{U}(S_3)} \mathbf{E}_{Y \in \lambda(S_3, X)} \mathbf{E}_{Y \in \hat{D}(S_3)} \mathbf{E}_{z \in \Phi(S_3, Y) \setminus \{i\}} \frac{f_{(S_3, Y)}^z}{\sum_{V \in \hat{U}(S_3)} \sum_{b \in \lambda(S_3, V)} \sum_{h \in \Phi(S_3, Y) \setminus \{i\}} f_{(S_3, Y)}^h} \theta_{(S_3^*)_{ij}}^{zy}, \forall j \in \hat{V}(i)$$

e2) i nodu ortak nod ise,

$$\hat{\theta}_{S_3, Y}^{S_3, S_2} = \sum_{P \in G(i) \cap G(j)} \left\{ \sum_{X \in \hat{U}(S_\beta)} \sum_{Y \in \lambda(S_\beta, X)} \sum_{Y \in \hat{D}(S_\beta)} \sum_{z \in \Phi(S_\beta, Y) \setminus \{i\}} \theta_{S_\beta, Y}^{S_3, S_2} \left[(1 - \gamma_\beta(i)) P_S^\beta(i) \right] \left[\frac{f_{(S_\beta, Y)}^z}{\mathbf{E}_{V \in \hat{U}(S_\beta)} \sum_{b \in \lambda(S_\beta, V)} \sum_{h \in \Phi(S_\beta, Y) \setminus \{i\}} f_{(S_\beta, Y)}^h} \theta_{(S_\beta^*)_{ij}}^{zy} + \mathbf{E}_{\alpha \in G(i) \cap G(j) \setminus \{0\}} \mathbf{E}_{Y \in \hat{D}(S_\alpha)} \mathbf{E}_{z \in \Phi(S_\alpha, Y) \setminus \{i\}} \left[\theta_{S_\alpha, Y}^{S_3, S_2} \gamma_\beta(i) \Delta_{\beta\alpha}(i) P_S^\beta(i) \right] \left[\frac{f_{(S_\alpha, Y)}^z}{\sum_{h \in \Phi(S_\alpha, Y) \setminus \{i\}} f_{(S_\alpha, Y)}^h} \theta_{(S_\alpha^*)_{ij}}^{zy} \right] \right], \forall i \in \hat{V}(i)$$

Teorem: Yukarıdaki şekilde bulunan geri besleme matrisi (0) makalenin II. bölümde gösterilen Ş1 - Ş4 şartlarını sağlar.

İspat: Teoremin ispatı yer darlığı nedeniyle verilememiştir. Yazarlarla irtibat kurularak teoremin ispatı temin edilebilir.

Büyük ölçekli bilgisayar ağının geri besleme matrisi (0) bulunduktan sonra, [5]'de bahsedildiği gibi bir yönlendirme kontrolü kullanılır. Sonuç olarak büyük ölçekli bilgisayar ağları için dışmerkezli bir yönlendirme kontrolü bulunur.

V. SONUÇ

Bu çalışmada, örtüşen kısımlarında bir veya daha fazla ortak nod olan ve iki veya daha fazla altağa sahip bilgisayar ağlar; için [4]'de geliştirilen yönlendirme kontrolü algoritması yerine, daha az hesaplama gerektiren ve her bir mesaj için bilinmesi gereken bilgi miktarının daha az olduğu (burada her bir mesaj için kaynak ve varış nodunun bilinmesi yeterlidir) bir yönlendirme

kontrolü algoritması geliştirilmiştir. Özgün bilgisayar ağı için bir geri besleme matrisi hesaplanarak, [5]'de bahsedildiği gibi bir yönlendirme kontrolü kullanılmıştır. Böylece büyük ölçekli bilgisayar ağları için dışmerkezli bir yönlendirme kontrolü bulunmuş olur.

VI. KAYNAKÇA

- [1] A. Aybar, "Bilgisayar ağlarında örtüşmeli ayrıştırma ve dinamik yönlendirme kontrolü," Yüksek Lisans Tezi, Osmangazi Üniversitesi, Ağustos 1995.
- [2] A. Aybar ve A. İftar, "Bilgisayar ağlarında içerme prensibinden yararlanılarak yönlendirme kontrolü tasarımı," *Birinci Bilgisayar Ağları Sempozyumu Kitabı*, s. 106-115, İstanbul, Türkiye, Mayıs 1996.
- [3] M. Ikeda ve D. D. Siljak, "Overlapping decentralized control with input, state, and output inclusion," *Control Theory and Advanced Technology*, c. 2, s. 155-172, 1986.

- [4] B. Ataşlar ve A. İftar, "Decentralized routing controller design for networks with more than two overlapping subnetworks," *Proceeding of IEEE International Conference on System, Man, and Cybernetics*, s. 2960-2965, Vancouver, BC, Canada, 1995.

- [5] A. İftar ve E. J. Davison, "A decentralized discrete-time controller for dynamic routing," *Proceedings of the IEEE Conference on Decision and Control*, s. 1362-1366, Honolulu, HI, Aralık 1990.

- [6] A. İftar, "Overlapping decentralized dynamic optimal control," *International Journal of Control*, c. 58, s. 187-209, 1993.

- [7] A. Aybar, A. İftar, ve M. B. Uzman, "Decentralized routing control using overlapping decompositions," *Proceedings of the Mediterranean Electrotechnical Conference*, s. 671-674, Antalya, Türkiye, Nisan 1994.



Controller Area Network (CAN) Uyumlu Arabirim Elemanları Tasarımı ve Uygulaması

H. EKİZ¹, A. KUTLU², İ. ERTÜRK³, ET. POWNER³

¹ Sakarya Üniversitesi, Sakarya, Türkiye,

² S. Demirel Üniversitesi, Isparta, Türkiye,

³ The University of Sussex Brighton BN1 9QT, U.K.

ABSTRACT

The Controller Area Network (CAN) is a serial bus with high speed, high reliability, and low cost for distributed real time control applications. The CAN is a desirable, cheap solution for networks in industrial environments [1], but there are limits both on the maximum length of a single CAN bus and on the number of nodes that may be connected together. A solution is to divide the CAN network into segments and connect them using bridges. Bridges are high performance devices that are used to interconnect LAN s at the Logical Link Control (LLC) or Medium Access Control (MAC) level in the protocol hierarchy.

Unlike many serial communication protocols the CAN message contains no information relating to the destination and source addresses. Instead the message contains an identifier which indicates the type of information contained in the message. Because of this feature of the CAN message it is not possible to use traditional address based bridges to connect CAN segments.

The aim of this study is not only to design and implement a bridge to connect CAN segments based on CAN protocol features but also to do performance analysis of the designed bridge. In the following, a brief overview about the CAN protocol and bridges is presented and the designed system is explained in detail and is simulated. In addition, the results obtained from the performance analysis of the bridge with different loads are presented.

1. Giriş

Bilgisayar ağları terimi; kaynakların paylaşımını sağlamak amacıyla bilgisayar, mikrokontroler ve diğer elemanların (yazıcı, dosyalar, bellekler, v.b.), daha çok bilgiye ulaşabilmek, daha fazla elemanı kontrol edebilmek ve daha fazla işlem gücü üretebilmek için, hatlar ve arabileşim elemanları aracılığıyla birleştirilmesi şeklinde tanımlanabilir. Birlikte bağlanan elemanlar bir ofiste, bir bina içerisinde, bir arabada olabileceği gibi bir ülkeye dağılmış şekilde veya farklı ülkelerde de bulunabilir. Elemanları birbirleri ile bağlamada kullanılan

karakteristikler 'protokol' olarak adlandırılır. Şu anda birbirinden farklı özelliklere sahip çok sayıda haberleşme protokolü bulunmaktadır; Ethernet, ATM, CAN, Token Ring, TCP IP, ISDN bunlardan bazılarıdır.

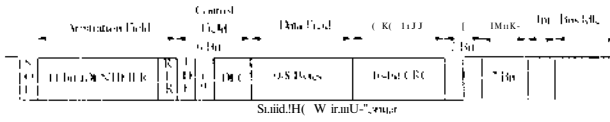
Bilgisayar ağlarında karşılaşılan problemlerden birisi, farklı özelliklere sahip sistemler arasındaki haberleşmedir. Bunun anlamı her yeni protokolle beraber yeni arabirim elemanlarına ihtiyaç duyulacağıdır. Kısa zaman önce kullanıma sunulan ve hızla kullanım alanı genişleyen *Controller Area Network* (CAN) ile birlikte hem CAN sistemlerini birbirlerine bağlayacak, hem de CAN ile diğer sistemlerin haberleşmesini sağlayacak arabirim elemanlarına ihtiyaç ortaya çıkmıştır. Bu çalışmanın amacı CAN sistemlerini birbirlerine bağlayacak arabirim elemanları (şu anda piyasada bulunan arabirim elemanları CAN protokolünün özellikleri nedeni ile kullanılamamaktadır) tasarımı ve uygulamasını yapmaktır. Bu amaçla takip eden bölümlerde önce CAN ve bilgisayar ağları ile ilgili bir özet verilmiş; daha sonra CAN/CAN arabirim elemanının tasarımı ve uygulaması ile ilgili ayrıntılar açıklanmıştır.

2. Control Area Network (CAN)

CAN, gerçek zamanlı kontrol uygulamalarında kullanılan yüksek hız, yüksek güvenilirlik, ve düşük fiyatlı, seri bir haberleşme protokolüdür. Şekil 1'de çatı (frame) formatı görülen bu protokol, başlangıçta otomobillerdeki elektronik kontrol sistemlerinin tek bir hat üzerinden birbirleri ile haberleşmesi için kullanılmasına rağmen, kısa zaman içerisinde farklı yerlerde uygulama alanı bulmuştur. PLC kontrollü üretim sistemleri, robotik, tıbbi aletler, lazerler, uzay uçuş simülatörleri, endüstrideki CAN temelli sistemler, gemi otomasyon sistemleri, şu andaki kullanım alanlarından sadece birkaçıdır [1,2].

Otomobiller için tasarlanmış olması ve yüksek verimi nedeni ile kısa süre içerisinde özellikle sanayide kullanılan kontrol sistemlerinde aranan bir sistem haline gelmesi [3], beraberinde yeni problemleri de getirmiştir. Bu problemlerden birisi, sanayideki

uygulamalarında ortaya çıkan CAN taşıtı uzunluğu, bir diğeri kullanılan diğeri sistemlerle haberleşme uyumsuzluğudur.



ŞEKİL 1. CAN çatı formatı

ilk problemin çözümü, CAN temelli sistemleri kısımlara ayırmak (segmented) ve bu sistemleri arabirim (interworking) elemanları (repeater, bridge, router v.b.) olarak adlandırılan elemanlardan uygun olanlar yardımıyla birleştirmektir. [2]'de CAN temelli sistemler için uygun arabirim elemanının 'bridge' olduğu gösterilmiştir.

ikinci problemin çözümü, sistemler arasında uyumsuzluğu ortadan kaldıracak 'bridge' veya 'router' kullanmaktır. Bu problemin çözümü ile ilgili ayrıntılı bilgi [4] ve [5]'de bulunabilir.

3. Bilgisayar Ağları Arabirim Elemanları ve Bridge'ler

Bilgisayar Ağlarının yayılım alanını genişletmek veya farklı protokole sahip sistemleri bir arada kullanmak için ağ arabirim elemanları kullanılır. Yaygın olarak kullanılan 4 çeşit arabirim elemanı vardır: Repeater, Bridge, Router, Gateway.

Birbirleri ile aynı bölgede bulunan ve genel olarak aynı karakteristiğe sahip ağları bağlamada bridge kullanılırken [6], uzak mesafelerde ve çok farklı özelliklere sahip sistemler arasında router veya gateway tercih edilir. Bu çalışmada CAN sistemlerinin yayıldığı alanı genişletmek amacıyla CAN protokolü özelliklerine uygun bridge kullanımı önerilmekte ve bu amaçla CAN/CAN bridge tasarımı açıklanmaktadır.

Kullanıldıkları yerlere ve yaptıkları işlere göre doğrudan aktaran (pass-through) bridge'ler ve uyumlaştırıcı bridge'ler (translation bridges) olarak iki gruba ayrılan bridge'ler, genel olarak Şekil 2'de özetlenen işlemleri gerçekleştirirler. Bridge tarafından gerçekleştirilen işlemler 3 aşamada özetlenebilir.

- Sistemlerden gelen bütün mesajların ilgili birim tarafından alınması,
- Gelen mesajlardaki tanıtım bilgileri ile kendi veritabanı tablosunda (database table) bulunan bilgilerin karşılaştırılması,
- Karşılaştırma sonucuna göre alınan mesajın (doğrudan veya uyumlaştırma işleminden sonra) ilgili sistemlere gönderilmesi (forwarding) veya mesajın silinmesi (discarding).

4. CAN / CAN Bridge Tasarımı

Bridge tasarımında en önemli faktörlerden birisi mesajların yönlendirilmesinde kullanılacak bilginin seçimidir. CAN mesajı çatışma (collision) temelli seri bir haberleşme protokolü olmasına rağmen, diğeri seri haberleşme sistemlerinden farklı olarak mesaj içerisinde kaynak ve hedef adresleri ile ilgili herhangi bir bilgi yoktur (Şekil 1). Bunun anlamı; piyasada bulunan ve kaynak adres / hedef adres / ağ numarası bilgilerinden birisini kullanan bridge'lerin CAN sistemlerinde kullanılmayacağıdır [7].

CAN mesajı yalnızca mesajı tanımlamak amacıyla kullanılan tanıtıcı bilgisine sahiptir (arbitration field) ve bu bilgi CAN protokolündeki mesaj önceliğini çözümlenmek amacıyla da kullanılır [8]. Tasarlanan sistemde mesaj tanıtıcı bilginin yönlendirme kararları için kullanılması önerilmektedir.

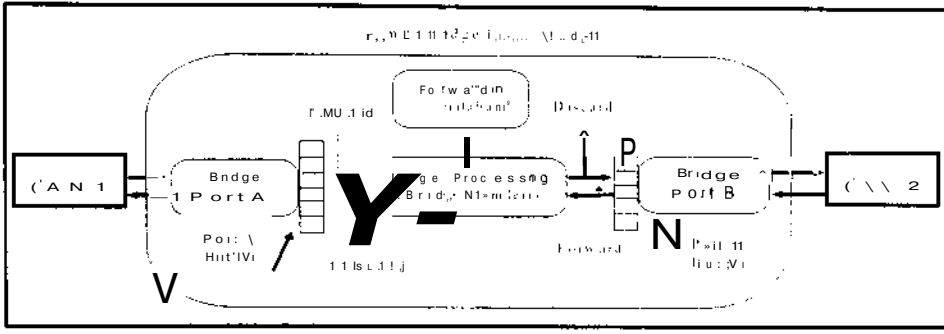
CAN / CAN bridge tasarımında karşılaşılan diğeri bir problem, CAN sistemlerinde kullanılan onaylama (acknowledgement) işlemidir. Bu problem; CAN chip'inki ilgili yazıçaları ayarlamak süretiyle çözümlenebileceği önerilmektedir [7].

Tasarlanan sistemde; CAN sistemlerinden alınan bütün mesajlar, ilgili CAN ünitesi tarafından, karşılaştırma işlemi yapılmak üzere bir geçici bellek (buffer) de depolanmaktadır. Depolanan mesajlar ile bridge veritabanında (ortak bellekte) bulunan bilgiler adres işleme ünitesi (AİÜ) tarafından karşılaştırılır. Karşılaştırma sonucunda mesajın bridge'in diğeri tarafında bulunan CAN sistemine iletilmesi gerektiğine karar verirse, mesaj ilgili CAN ünitesine gönderilir. Mesajın ilgili veritabanında bulunmaması durumunda mesaj silinir.

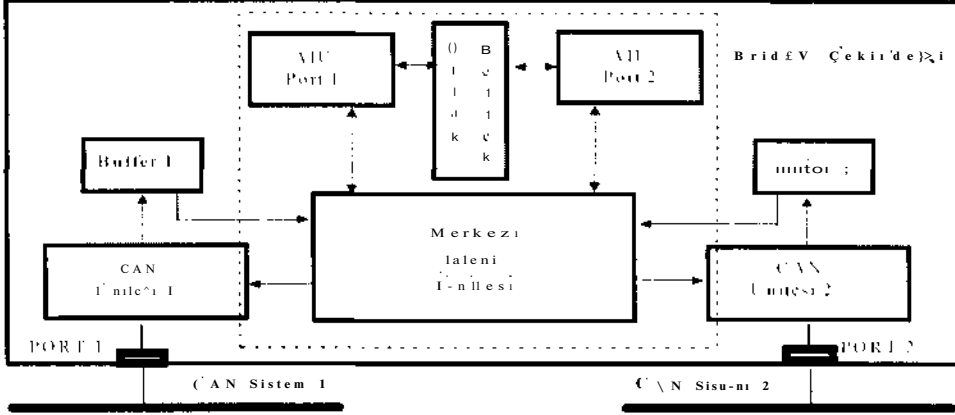
Tasarlanan sistemde mesajlar üzerinde yapılan işlemlerin büyük bir kısmı adres işleme ünitesi tarafından gerçekleştirilir. Mesajlardan karşılaştırılacak kısımların ayrıştırılması ve karşılaştırma işlemleri [7]' de ayrıntılı olarak açıklanmıştır.

5. Tasarlanan Bridge'in Uygulaması

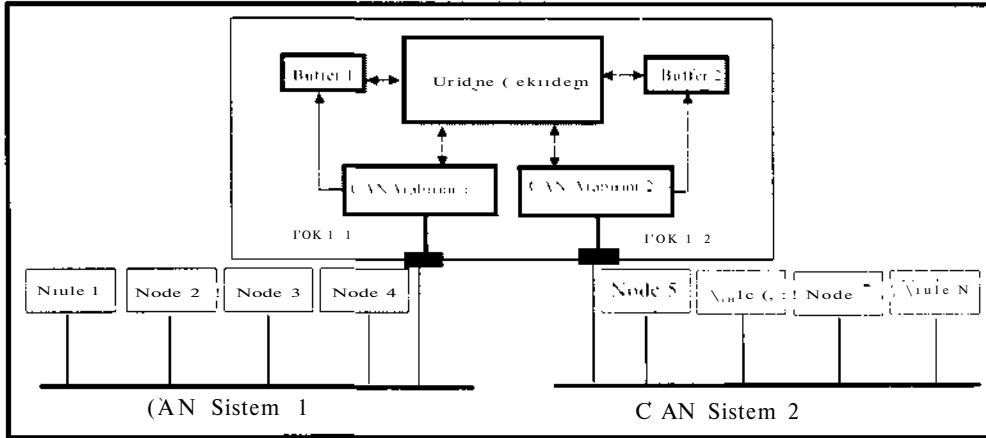
Tasarlanan bridge uygulama sonuçlarını göstermek amacıyla Şekil 4' de gösterildiği biçimde bir sisteme yerleştirilmiştir. Bridge'in bağlandığı CAN sistemleri her birisi lokal ve uzak (remote) mesaj gönderme kapasitesine sahip 4 kullanıcı / istasyon'dan (node) oluşmaktadır. Tasarlanan bridge uygulama devresi, simülasyon tekniğinden faydalanılarak gerçekleştirilmiş ve belirli bir senaryoya göre oluşturulan mesaj trafiği altında sistem tepkisi incelenmiştir.



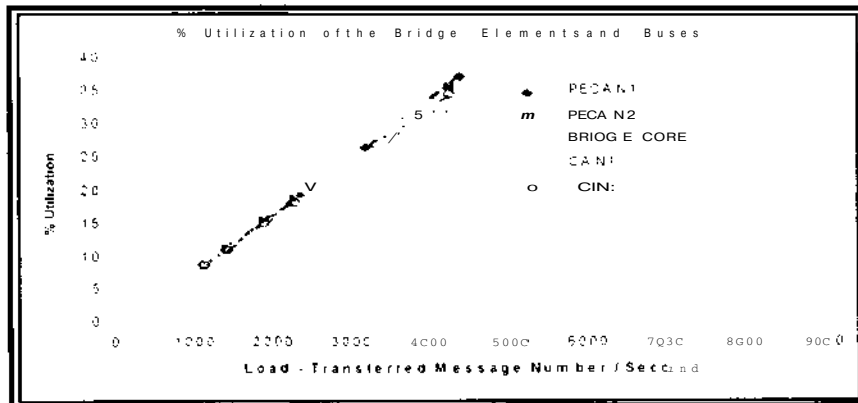
ŞEKİL 2. Temel bridge işlem modeli



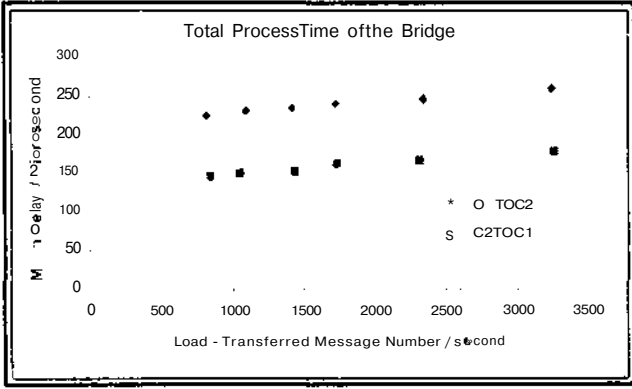
ŞEKİL 3. Tasarlanan CAN / CAN bridge blok şeması



ŞEKİL 4. Bridge yardımıyla bağlanmış CAN sistemleri



ŞEKİL 5. Bridge elemanları ve taşıtlar kullanım oranları



ŞEKİL 6. Toplam bridge işlem zamanı

Uygulaması yapılan senaryoda, CAN sistemleri arasında %70 yerel (local) %30 uzak sistemlere olacak şekilde bir mesaj akışı sağlanmış, bridge elemanları ve taşıtların Şekil 5'de görülen oranlarda kullanıldığı (utilization) bulunmuştur. Şekilden görüleceği üzere, CAN üniteleri (PECAN), bridge çekirdeği (bridge core) ve CAN taşıtlarında kullanım oranı kabul edilebilir sınırlar içerisinde [9].

Yukarıda bahsedilen % 70 yerel %30 uzak mesaj trafiği altında, sistemden geçen mesajların bridge içerisinde toplam harcadıkları zaman, saniyede gönderilen mesaj sayısına bağlı olarak Şekil 6'da görülmektedir. Uygulaması yapılan bridge içerisinde harcanan zaman, gerçek zamanlı kontrol sistemlerinde kabul edilen sınırlar içerisinde [9].

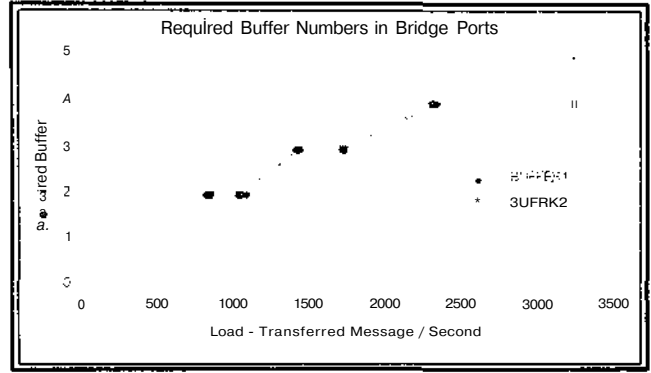
Bridge uygulamalarında en önemli faktörlerden birisi gerekli bellek miktarıdır. Uygulanan senaryoya göre bridge için gerekli geçici bellek (buffer) miktarı Şekil 7'de sunulmuştur. Şekilden görüldüğü gibi gerekli bellek miktarı da kabul edilebilir sınırlar içerisinde [9].

6. Sonuç ve Değerlendirme

Yapılan bu çalışmada, CAN sistemlerinin sanayide kullanımında ile ortaya çıkan problemler incelenmiş ve bu problemlerin çözümü açıklanmıştır. CAN sistemlerinin daha geniş alanlarda kullanılabilmesi için uygun çözümün CAN sistemlerini bridge'ler yardımıyla birleştirmek olduğu ifade edilmiştir. Piyasada bulunan bridge'lerin CAN sistemleri arasında kullanılamaması nedeniyle, CAN protokolü özelliklerine uygun yeni bir bridge tasarımı yapılmıştır. Son kısımda, tasarlanan bridge iki CAN sisteminin bağlanması amacıyla kullanılmış ve belirli bir senaryoya göre elde edilen sonuçlar sunulmuştur.

Kaynaklar

1. Lawrenz W., "World wide status of CAN-Present and Future", 2nd International CAN Conference Proceedings, London, 1995.



ŞEKİL 7. Gerekli bellek sayısı (buffer sayısı *108 Bits)

2. H. Ekiz, A. Kutlu, E.T. Powner, "Performance Analysis of CAN in Bridged systems", Proceedings of II Communication Symposium, June 1995, Manchester, UK .

3. Mclaughlin R., 'CAN Controlling from cars to X-rays', IEE Networking May 95, UK.

4. H. Ekiz, A. Kutlu, M. D. Baba, E. T. Powner, "Design and Implementation of a CAN / Ethernet Bridge", Proceedings of 3rd International CAN Conference, pp. 11/17-11/26, 1-2 October 1996, Paris, France.

5. H. Ekiz, A. Kutlu, M.D. Baba, ET. Powner, "Performance Analysis of a Ethernet / CAN Bridge", IEEE SICON'97 Conference, 14-17 April 1997, Singapore.

6. Held G., Internetworking LANs and WANs, Wiley communication Technology, England, 1993.

7. H. Ekiz, A. Kutlu, E. T. Powner, "Design and Implementation of a CAN/CAN Bridge", Proceedings of IEEE ISPAN'96 Conference, pp. 507-513, 12-14 June 1996, Beijing, China.

8. Controller Area Network (CAN), LAN in vehicle communication protocol, SAE J1583 Mar90, SEA Information Report, pg. 20. 226-248.

9. H. Ekiz, A. Kutlu, M. D. Baba, E. T. Powner, "Performance Analysis of a CAN / CAN Bridge", Proceedings of IEEE ICNP'96 Conference, pp. 181.188, 29 October-1 November 1996, Ohio, USA.

ODTÜ BİLGİSAYAR AĞI ÜZERİNDE ÖLÇMELER VE BİR SİMÜLASYON ÇALIŞMASI

NUSRET YILMAZ¹, F.RÜYAL ERGÜL
Orta Doğu Teknik Üniversitesi
Elektrik ve Elektronik Mühendisliği Bölümü
06531 - Ankara

ABSTRACT

In order to assess the performance of present METU-NET and the performance of network when traffic grows in future, a study has been done to determine the traffic parameters of network using a special software tool "Sniffer". Based on the results of Sniffer measurements and physical parameters of back-bone network subsystems, a simulation is done to estimate the network performance when network traffic is increased. The obtained results indicate that present network was not overloaded at the time of investigations and has capacity for handling traffic load increases up to 200 percent with acceptable delay levels.

1-Giriş:

Orta-Doğu Teknik Üniversitesi yerleşke yerel iletim ağı METU-NET, IEEE 802.5 standartına uygun Token-Ring türü bir yerel iletim ağıdır (LAN-Local Area Network) ve IBM firması tarafından kurulmuştur. 16 Mbps iletim hızında çalışmakta olan METU-NET ağı, yerleşke içinde binalar arası mesafelerin fazla olması nedeniyle bir ana halka ve iki yan halkadan oluşturulmuştur. Ağı meydana getiren temel birimler olarak Çoklu Erişim Ünitesi MAU (Multistation Access Unit), Yerel Ağ Köprüleri (LAN Bridges) ve fiber optik dönüştürücüler sayılabilir. Ağ içinde, sekiz port'u düğümlerin iletim ortamına erişimlerini sağlamak ve iki port'u da MAU'ların birbirlerine bağlanmasını sağlamak için ayrılmış 16 adet IBM 8228 tipinde MAU bulunmaktadır. İki adet IBM 8209 LAN Bridge ana halka ile yan halkaların köprü bağlantısını gerçekleştirmek için kullanılmaktadır. IBM 8220 fiber optik dönüştürücüler ise, ağın daha geniş bir alana yayılabilmesi için MAU'ların birbirine fiber optik kablo üzerinden bağlanmasında kullanılmaktadır. İletim ortamına bağlı birbirinden farklı özelliklere sahip düğümler iletim ortamına MAU'lar üzerinden bağlıdır ve düğümlerin toplam sayısı 95'dir.

METU-Net'in kapasitesinin ne ölçüde kullanıldığının belirlenmesi ve muhtemel trafik yükü artışları durumunda performansının değerlendirilmesi amacıyla bu çalışma yapılmıştır. Yerel ağa bağlı düğüm noktalarının trafik değerlerinin ölçülmesi için Sniffer ağ analizi yazılımı kullanılmış, elde edilen bulgulara dayanılarak yapılan bir simülasyon çalışması sonucunda da sistemin muhtemel yük artışı durumundaki başarımı belirlenmiştir.

2-Sniffer Network Analyzer ve Veri Toplama Çalışmaları:

iletim ortamına bağlı yaklaşık 35 bölüm, idari birimler, merkezler, enstitüler, çeşitli bölüm ve merkezlerin server'ları gibi özellikleri ve trafik yükleri birbirinden farklı pek çok düğüm bağlıdır. Bu düğümlerin ürettikleri trafik çok değişik istatistiksel özellikler göstermektedir. METU-NET üzerinde yapılacak herhangi bir çalışmanın gerçekçi olması için, istatistiksel bilgilerin güvenilir olması gerekmektedir. Bu bilgilere dayandırılacak trafik modelleri güvenilir olacaktır ve elde edilecek sonuçlara duyulacak güven de aynı oranda artacaktır. Gerçekçi veri elde etmek için doğrudan ağ üzerinde ölçü alma gereği duyulmuş. Bilgişlem Daire Başkanlığından destek alınarak istenilen ölçmeler yapılmıştır. Düğümler tarafından üretilen trafiği incelemek ve düğümler ile ilgili istatistiksel verileri toplamak amacı ile Bilgişlem Daire Başkanlığında bulunan "Sniffer Network Analyzer" yazılımı kullanılmıştır. Bu yazılım, bağlı olduğu iletim ortamının aktivitesini gözlemleyebilen, kaydedebilen ve görüntüleyebilen bir yazılımdır. Yazılım, iki temel konumda çalışmaktadır. Bunlar "analiz" ve "monitör" konumlarıdır. Analiz konumunda çalıştırıldığı zaman, diagnostik çalışmalar yapılabilir. Monitör konumunda çalıştırıldığı zaman, yazılım bağlı olduğu ortamı dinleyen pasif bir gözlemcidir, ortamın trafiğine katkıda bulunmaz. Monitör konumunda

* Devlet Hava Meydanları İşletmesi Genel Müdürlüğü, Ankara



çalışırken iki temel işlev görür. Bunlar algılama ve görüntüleme işlevleridir. Algılama sırasında kullanıcı tarafından belirtilecek parametreler doğrultusunda veriler filtrelenerek kaydedilir. Görüntüleme sırasında kaydedilen veriler sınıflandırılır, yorumlanır ve görüntülenir.

Sniffer Network Analyzer kullanılarak METU-NET üzerinden iki farklı veri grubu toplanmıştır, ilk önce, bir hafta boyunca ve her gün ikişer saatlik ortalamalar alınarak 24 saat boyunca kesintisiz veri toplanmıştır. Daha sonra, günün en yoğun trafik değerinin yaşandığı iki saatlik zaman diliminde daha detaylı bilgi edinmek için 10'ar dakikalık ortalamalar alınarak veri toplanmıştır. Kaydedilen Sniffer Network Analyzer verileri grafiksel olarak ifade edilmiştir. Bu grafiklere örnek olarak incelemenin yapıldığı haftanın en yoğun trafiğinin yaşandığı 29 Mart 1996 Cuma gününün değerleri Şekil-1'de gösterilmiştir. Ancak, burada verilen değerlerin anlık trafik değerleri değil ikişer saatlik ortalama değerler olduğuna dikkat edilmelidir. Daha kısa zaman dilimleri temel alınarak yapılacak incelemede anlık olarak yaşanan daha yoğun trafik değerlerinin etkilerinin kolaylıkla görüleceği açıktır. Şekil-2'de 10 Mayıs 1996 Cuma gününde, günün en yoğun saat dilimi olan 10:00 ile 12:00 yaşanan trafik değerlerinin 10'ar dakikalık ortalama değerleri görülmektedir, iki saatlik ortalama değer yaklaşık 2.2 Mbps olduğu halde 10'ar dakikalık ortalama değerlerin 3 Mbps'e kadar çıkabildiği gözlenmektedir. Pratik nedenlerden dolayı ortalama alma süreleri 10 dakika ile sınırlanmıştır. Trafiğin gerçek tepe değerlerinin görülmesi için çok daha kısa ortalama alma sürelerine ihtiyaç vardır. Network performansının belirlenmesinde hem anlık tepe değerleri hemde ortalama değerler gözönüne alınmak zorunda olduğu için bu ölçmeler önemlidir. Pratik nedenlerden dolayı 10 dakikadan daha kısa sürelerde ölçü alınmamıştır.

Beklendiği gibi, ölçmeler sonucunda METU-NET'e bağlı düğümlerin ürettikleri trafiğin yoğunluğu ve dağılımı açısından birbirlerinden oldukça farklı özellikler taşıdığı gözlemlenmiştir. Örneğin Sniffer Network Analyzer'in çıktıklarına göre 29 Mart 1996 Cuma günü saat 10:00 ile 12:00 arasında network'e bağlı tüm düğümün ürettikleri trafik miktarlarının kümülatif değerleri karşılaştırılınca en fazla veri ileten düğüm lardan "Cisco AGS'nin trafik miktarının, en az veri üreten düğümlerden "IBM 293137"nin

trafik miktarına oranının yaklaşık 1 000 000 olduğu gözlenmektedir. Bu durum, METU-NET gibi klasik ve paylaşımlı yerel ağlarda kullanıcılar için adanmış band ayrılmadığından, bir bölüm kullanıcı için veri iletim ortamının kullanımını, gereksiz gecikmeler yaratarak verimsiz yapacaktır.

3-METU-NET Simülasyonu

Simülasyonun amacı, METU-NET'in kullanıcı sayısının artması ve yeni servislerin sağlanmaya başlanması ile birlikte ortaya çıkan gereksinimler karşısında performansının değerlendirilmesinin yapılmasıdır. Simülasyon sırasında incelenen temel performans kriteri, üretilen trafik içinde gönderilmeden önce bekletilen trafiğin yüzdesidir. Simülasyon yazılımı Turbo Pascal 6.0 kullanılarak yazılmıştır. Yazılım girdileri METU-NET'e bağlı her düğüm için normal çalışma koşullarında beklenen trafik değerleridir. Her düğüm tarafından yaratılan trafik bir rastgele değişken (Poisson) kullanılarak temsil edilmektedir. Her düğüme bağlı kullanıcı sayısının büyük olduğu göz önüne alınarak, her kullanıcı tarafından yaratılan Poisson dağılımının bir Gaussian dağılımı ile yaklaşımının yapılabileceği kabul edilmiştir. Yazılım, ayrıca fiziki hat uzunluklarını ve diğer gecikmelerini ayrı ayrı göz önüne alarak bir token'ın ağ içinde bir kere dönmesi için gereken süreyi de hesaba katmakta, ve üretilen belirli uzunluktaki veri boğunun gönderilme süresini oldukça gerçekçi biçimde hesaplayabilmektedir.

Yazılım çıktıları, simülasyon süresince her bir düğümün ürettiği ve ortalama ilettiği trafik değerleri, iletilebilen trafiğin üretilen trafiğe oranı, ağ üzerinde iletilen trafik, kullanıcılar tarafından üretilen toplam trafik değerleri, simülasyon süresi ve bu süre boyunca token dönüş zamanı (Token Rotation Time) gibi bilgilerdir.

Simülasyon yazılımı, Sniffer yazılımı tarafından belirlenen trafik değerlerini normal değerler olarak kabul ederek, bu değerlere göre %50, %100, %150 ve %200 oranında artırılmış beş değer için koşturulmuştur. Normal değerler, Sniffer tarafından yapılan bir haftalık ölçmeler sırasında en yoğun gün olan 29 Mart 1996 Cuma günü bulunan trafik değerleridir. Şekil-3, artan trafiğin ortaya çıkarttığı gecikmeyi; toplam üretilen verilerden iletilmeden önce bekletilenlerin üretilen verilere

olan oranı cinsinden vermektedir. Şekil-3 sonuçlarına göre, üretilen toplam trafiğin yaklaşık %10'luk bölümü, trafik yaklaşık %240 oranında artarsa gecikmeli olarak iletilmektedir. Daha küçük oranlarda «trafik artışı kayda değer gecikmeler yaratmamaktadır. Normal trafik şartları altında "ağ doğrudan iletimi" (Network Through-put) 3.238 Mbps olarak bulunmuştur. Klasik ağ yapılarında 16 Mbps çalışma hızının yaklaşık 10 Mbps ağ doğrudan iletimine karşılık olduğu düşünülürse, METU-NET için kapasitesinin oldukça altında çalıştığı söylenebilir.

Yapılan ölçüler uzun dönem ortalamaları gösterdiği için, özellikle token-ring türü yapılarda, bulunan ağ doğrudan iletimi değerleri tam ve kesin bir şekilde ağ kapasitesini ve ağ performansını belirleyemez. Trafiğin katlanarak artması durumunda, tepe değer büyüklüğüne bağlı olarak, gecikmeler kullanıcılar için artacaktır. Diğer bir deyişle, doğrusal olarak artan trafiğe karşılık olarak ağ doğrudan iletimi doğrusal artmayacaktır, bir doyma etkisi gözlenecektir. Bu etkiyi görebilmek için, ağ doğrudan iletimi artan trafik oranına göre hesaplanmış ve Şekil-4'de sonuçlar verilmiştir. Şekil-4'e göre, doğrusal olarak artan trafiğe rağmen ağ doğrudan iletimi doğrusal artmamakta. beklendiği gibi daha yavaş artmaktadır. Dolayısıyla, ağ ihtiyacı tam olarak karşılamamakta. düğümlerde kuyruklar oluşmakta ve üretilen trafiğin bir bölümü gecikmeli olarak gönderilebilmektedir. Şekil-4 sonuçlarına göre, METU-NET trafiği, ölçmelerin yapıldığı Mart 1996 içerisinde henüz doyma etkisinin gözlenebileceği noktaya ulaşmamış bulunmaktadır.

4-Sonuçlar:

Elde edilen Sniffer sonuçları ve yapılan simülasyon çalışmaları, çalışmaların yapıldığı tarihte METU-NET'in kapasitesinin oldukça altında kullanılmakta olduğu, özellikle 10:00 ve 18:00 saatleri dışında kullanımın çok düşük olduğunu göstermektedir. Göze çarpan temel özellikler, genelde düğümler arası trafiğin çok az olması, buna karşılık Internet trafiği gibi bazı nedenlerden dolayı ağ kullanımında büyük asimetrliler görülmesi. düğümler

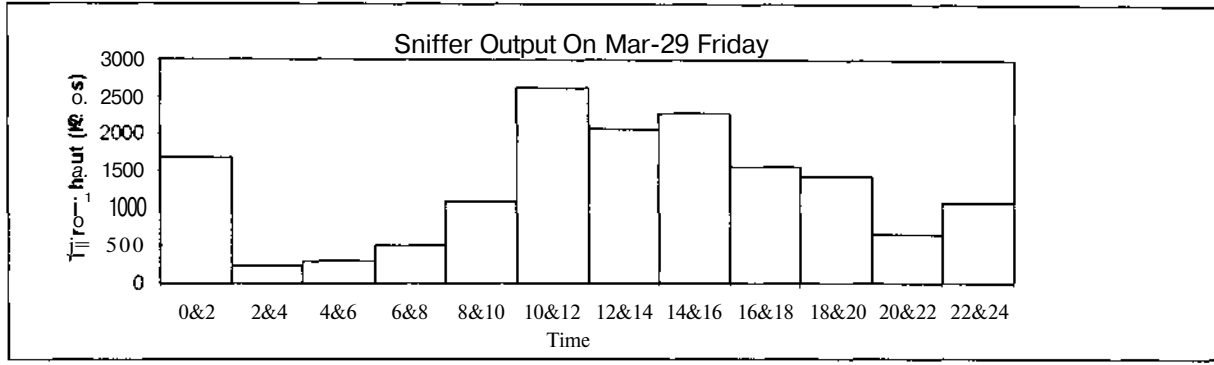
tarafından üretilen trafiğin 1:10³ gibi çok büyük oranlarda değişiklikler göstermekte olmasıdır. Görülen bir diğer olumsuzluk ise. ağ köprüleri üzerinde gözlenen yığılmalardır.

Bütün bu olumsuz etkilere rağmen. METU-NET için aşırı bir yüklenme gözlenmemiş, toplam trafik değerinin en az %200 artma durumunda bile çok ciddi gecikme problemleri ile karşılaşılmayacağı sonucuna varılmıştır. METU-NET için asıl dar boğaz, multimedya uygulamaları gibi geniş band isteyen uygulamaların ağ üzerinden yapılması durumunda ortaya çıkacaktır. Bu durum, Token-Ring türü paylaşımlı ortamlarda kaçınılmazdır fakat METU-NET içerisinde bu ihtiyacın ne zaman ortaya çıkacağı çalışmanın yapıldığı tarihte belli değildi.

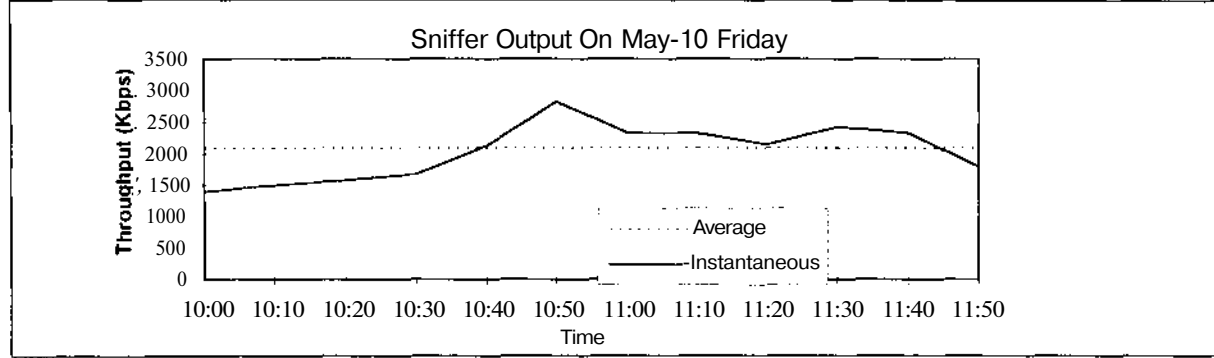
Yapılan çalışma sonuçlarının iyileştirilmesi ve METU-NET performansının daha büyük duyarlılıkla bulunması mümkündür. Bunun yapılabilmesi için daha geniş kapsamlı veri toplama ve özellikle toplanan verilerin 10 dakikaya göre çok daha kısa süreler içinde alınan ortalama değerlerden oluşması gerekecektir. Bu yapıldığı takdirde, simülasyon sonuçları daha detaylı olacak. METU-NET performansı da daha iyi değerlendirilmiş olacaktır.

5- Referanslar

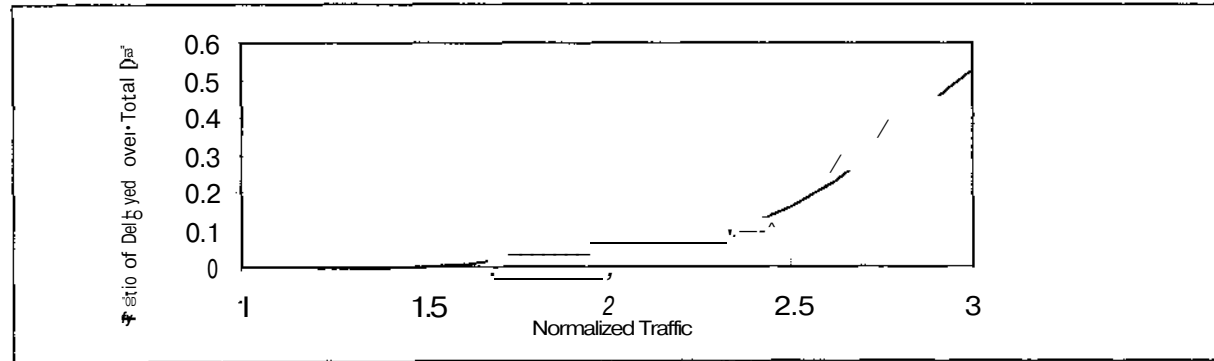
- 1- Nusret Yılmaz, Performance Evaluation of METU Local Area Network and Comparison with an ATM Based Network. M.Sc. Thesisi. September 1996, METU Ankara.
- 2- Gilbert Held, Token Ring Networks. Characteristics. Operation, Construction and Management. John-Wiley and Sons. 1994.
- 3- Jesper Nilhausen, Token Ring Networks. Prentice Hall 1990.
- 4- Martin dePrycker, Asynchronous Transfer Mode, Ellis Horwood LTD, 1993
- 5- Sniffer Network Analyzer, Operatör and Users Manuals. Vol 4,5,6. Network General Corp. 1996.



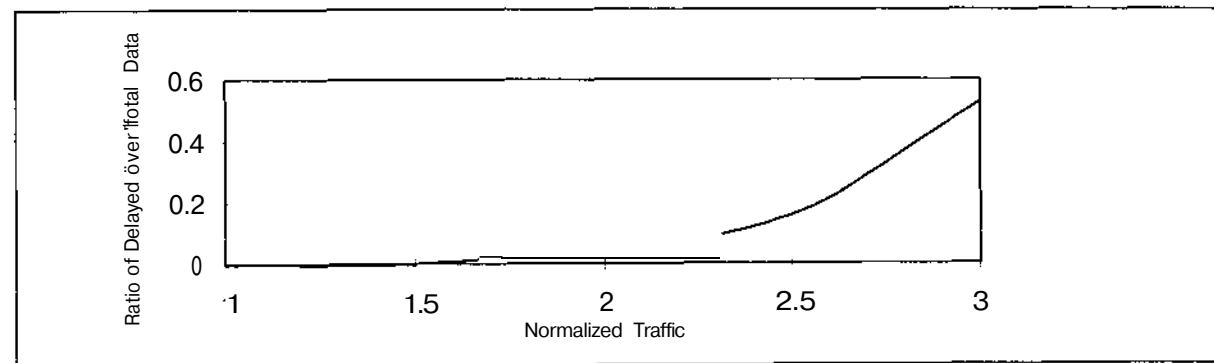
Şekil-1 METU-NET 24 Saatlik Trafik Değerleri (2 Saatlik ortalama değerler)



Şekil-2 METU-NET 24 Saatlik Trafik Değerleri (10 dakikalık ortalama değerler)



Şekil-3 METU-NET Trafikinde Gecikmenin Trafik Artışından Etkilenmesi



Şekil-4 METU-NET Ağ Doğrudan Kullanımının Artan Trafikçe Bağlı Değişmesi.

X.25 YÖNLENDİRİCİ VLSİ TÜMDEVRE TASARIMI

O.V. ERDAĞ, Prof. Dr. M. AŞKAR
TÜBİTAK BİLTEN
ODTÜ KAMPUSU
EE BİNASI D BLOK
06531 - ANKARA

ABSTRACT

This article mainly concerns the design of a packet router VLSI chip which will be used in X.25 packet switching communication networks. It was aimed to introduce a new point of view for the packet router concept and to show that it is possible to combine all the functions of packet switching on a single chip. This feature provides ease of the use to the users while simplifying the system design. The chip has been designed and synthesized totally using VHDL. Therefore, it will be easy to adopt the design for the future requirements, to the new technologies. Since data memory is integrated into the chip, it is not possible to extract and see the data flowing through the router.

Paket anahtarlama iletişim şebekelerinde kullanılmak üzere, X.25 standardına uyumlu bir paket yönlendirici yonga tasarımı gerçekleştirilmiştir. Yapılan çalışmanın amacı, halen kullanılmakta olan ve çok geniş bir yelpaze içinde çeşitliliğe sahip paket yönlendiricilere daha farklı bir bakış açısı ile yaklaşmak ve kullanıcılar için yeni seçenekler sunabilmektir. Aynı zamanda, yönlendirme işlevlerinin tamamını tek bir yonga üzerinde toplayarak X.25 paket veri ağları için gündeme gelmemiş olan *yönlendirme tümdevresi* kavramını tartışmaya açmaktadır.

GENEL ÖZELLİKLER

Tasarımı yapılan yonga, CCITT X.25 standardının [1] ikinci katmanı olan veri bağı katmanı (Data Link Layer) için tanımlanan ve çerçeve yapısı Şekil 1'de görülen LABP (Link Access Procedure - Balanced) ile tamamen uyumludur. Bunun yanısıra, yönlendirme işlevini gerçekleştirebilmek için, üçüncü katman olan ağ katmanına (Network Layer) ilişkin işlevlerin bir kısmını da içerir. İkinci katmanın bütünüyle gerçekleştirilmesi nedeniyle, yönlendirici olarak kullanılmasının yanısıra, istenildiğinde, X.25 iletişim denetleyici olarak da kullanılabilir.

Literatürde çeşitli paket yönlendiricilere rastlanmakla birlikte [2], [3], [4] X.25 için tasarlanmış

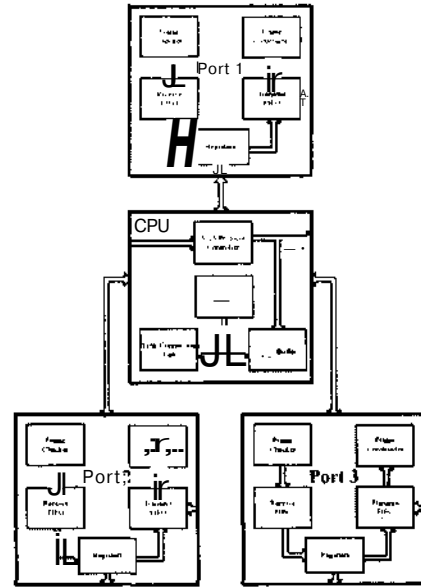
olan ilk yonga olma özelliğini taşıyan bu yönlendirici, çift yönlü (full-duplex) olarak çalışan üç porttan oluşmaktadır. Hiyerarşik yönlendirme (hierarchical routing) algoritmasının uygulandığı yönlendirici, bu yöntem sayesinde ülkemizde de halen kullanılmakta olan TURPAK paket bilgi şebekesine de uyum sağlayabilmektedir.

BAYRAK[8 Bit]	ADRESİS Biti	KOMUT[5 Bit]	BU Gİ[259Byte]	FCS[8 Bit]	BAYRAK[8 Bit]
---------------	--------------	--------------	----------------	------------	---------------

Şekil 1. LABP Çerçevesi.

X.25 standardı için tanımlanmış olan çalışma modlarında Temel Mod'a uygun olarak çalışmaktadır. *Single Link Procedure* uygulayan yonga, portları üzerinden toplam olarak 8 adet bağlantı kurabilme özelliğine sahiptir. Bilgi taşıma bölgesinde 256 byte uzunluğunda bilgi bulduran paketleri kullanabilen yonga, 8 bitlik arabirimi aracılığıyla da dış dünya ile iletişimini gerçekleştirir.

Şekil 2'de genel yapısı gösterilen yönlendirici, temel olarak iki birimden oluşmaktadır: *i.)* Alıcı-Verici Birim, *ii.)* Merkezi İşlem Birimi. Bu birimlerin yapısı aşağıdaki bölümlerde ayrıntılarıyla açıklanmıştır.



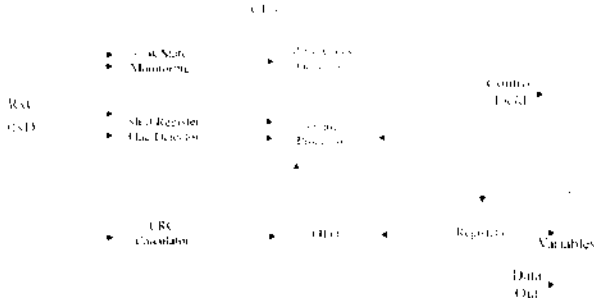
Şekil - 2. X.25 Yönlendirici Blok Diyagramı

ALICI-VERİCİ BİRİMİ

Her üç port üzerine yerleştirilmiş olan Alıcı/Verici Birimleri, bağlı oldukları port üzerindeki veri akışını sağlarlar. Port üzerinde kurulu bulunan bağdan (link) gelen bilgiler bu birim tarafından karşılanılır ve paketin yönlendirilme süreci de başlatılmış olur. Hatta gönderilecek veri de bu birim üzerinden yönlendiriciyi terkeder. Alıcı/Verici Birimi, adından da anlaşılacağı gibi , iki alt birimden oluşmaktadır: *i.)* Alıcı, *ii.)* Verici.

ALICI

Şekil 3'te işlevsel yapısı ve birimler arası bağlantıları gösterilmiş olan Alıcı, yönlendiriciye gelen verilerin karşılandığı, LAPB için tanımlanmış olan denetimlerin yapıldığı yerdir.



Şekil - 3. Alıcı

Alıcıya gelen paketlerin öncelikle geçerlilikleri denetlenir. Alınan paketin X.25'e uygunluğuna karar verildikten sonra, taşıma sırasında paket içindeki verilerde oluşabilecek hataların belirlenebilmesi için CRC (Cyclic Redundancy Check) hesaplanarak paket içinde FCŞ(Frame Check Şequence - Çerçeve Denetim Dizisi) alanına yerleştirilmiş olan değerle karşılaştırılır. Hata çıkması durumunda Merkezi İşlem Birimi uyarılarak, paketin alındığı noktaya gereken karşılığın verilmesi sağlanır.

Alıcı üzerinde, gelen paketlerin tipinin belirlenmesi ve paket içinde yeralan X.25 standardına ilişkin bilgilerin çıkartılabilmesi sırasında kullanılmak üzere, 24 byte büyüklüğünde bir FIFO türü bellek bulunmaktadır. Ayrıca, alıcı üzerinde kurulmuş bulunan bağlantı ile ilgili, V(S) (*Send State Variable*), V(R) (*Receive State Variable*), N(S) (*Send Sequence Number*) ve N(R) (*Receive Sequence Number*) gibi değişkenlerle port adresi, yönlendiricinin bulunduğu hiyerarşik konumu belirten düzey bilgisi yeralır. Alıcı üzerinde tutulan değişkenlerin listesi Tablo 1'de görülebilir.

1 numaralı yazmaçta saklanan *X.25 Protokol Durumu* değişkeni, protokolün işleyişi sırasında gelinen noktayı göstermektedir. 14 farklı durumun tanımlandığı bu değişken, Merkezi İşlem Birimi tarafından belirlenir ve her bir port için farklı değerler alabilir. Bu değişken Merkezi İşlem Birimi içinde açıklanmaktadır.

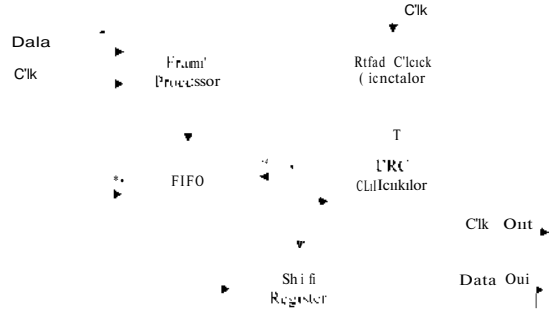
Tutucu içinde yeralan en önemli değişkenlerden birisi de onaltıncı yazmaçta saklanan *Düzye* bilgisidir. Değeri 0 ile 3 arasında değişir. "0", yönlendiricinin ülke girişine yerleştirildiğini ve paket ağları arasındaki iletişimi yönlendirmek üzere kullanılacağını belirtir. "1" şehirlerarası yönlendirmeleri yaparken "2" şehir içinde, data santralleri arasına yerleştirilmiş olan bir yönlendiriciyi tanımlar. Son olarak, "3"üncü düzeye yerleştirilmiş olan yönlendirici, santral içinde, data paket ağı aboneleri arasındaki paket yönlendirme işlemlerinin yapılacağını belirtir.

Tablo 1. Yazmaç İçerikleri.

Yazmaç	İçeriği
1	X.25 protokol durumu.
2	Linkin fiziksel durumu.
3	Gelen paketin tipi.
4	Alınan FRMR paketinin bilgi bölümü.
5	Alınan FRMR paketinin bilgi bölümü.
6	Alınan FRMR paketinin bilgi bölümü.
7	V(S)veV(R)
8	N(S)and N(R)
9	Port Adres 14-13
10	Port Adres 12-11
11	Port Adres 10-9
12	Port Adres 8-7
13	Port Adres 6-5
14	Port Adres 4-3
15	Port Adres 2-1
16	Düzye

VERİCİ

Yönlendiriciden çıkacak bilgilerin alıcı tarafa gönderilmesini sağlayan verici, temelde alıcıya benzemekle birlikte beş bloktan oluşan daha basit bir yapısı vardır (Şekil 4).



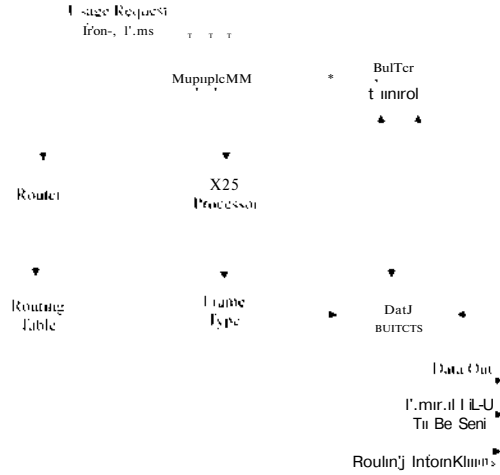
Şekil - 4. Verici

Vericinin en önemli görevlerinden biri CRC hesabıdır. Gönderilecek olan çerçevenin, yönlendirme işlemleri sonrasında, alınan çerçeveden farklı bir duruma gelmesi nedeniyle CRC hesabının yeniden

yapılması gerekir. Merkezi İşlem Birimi'nden alınan veriler doğrultusunda, *Frame Processor* ve 6 byte'lık *FIFO* yardımı ile oluşturulan yeni çerçeve için yeniden hesaplanan CRC. çerçevenin *FCS* alanına yerleştirilir. Bu işlemlerin yapılması sırasında ve özellikle bilgi byte'larının tampon belleklerden okunması sırasında gereken saat sinyalleri de verici içinde üretilir. Gönderime hazır duruma getirilen, ancak 8 bitlik paralel veri olarak hazırlanmış olan çerçeve, seri bilgiye çevrilerek iletim hattı üzerinden alıcı tarafa gönderilir.

MERKEZİ İŞLEM BİRİMİ

Yönlendirme işleminin gerçekleştirildiği ve X.25 protokolüyle ilgili işlemlerin yapıldığı birim olan Merkezi İşlem Birimi, yönlendiricinin en önemli parçasıdır (Şekil 5). Girişine yerleştirilmiş olan *Multiplexor*, portlardan gelen kullanım isteklerini değerlendirerek bu birimin sırayla, her üç port tarafından ortaklaşa kullanılmasını sağlar. Portların, port numarası ile doğru orantılı öncelikleri vardır. Bu durumda 1 numaralı port, en yüksek önceliğe sahiptir,



Şekil - 5. Merkezi İşlem Birimi

Alınan çerçevelerin değerlendirilerek yapılması gereken işlemlere ve gönderilecek paket tipine karar verme işlemleri yönlendirici içinde, *X.25 Processor* tarafından gerçekleştirilir. Bir durum makinası olarak yapılmış olan bu birim, toplam 14 durum üzerinde değerlendirme yapar ve karar verir. Bir sonraki basamakta bulunulacak durum, alınan paket tipi de gözönüne alınarak belirlenir. Daha önce Alıcı'nın tanımlanması sırasında da sözü edilen durumlar Tablo 2'de listelenmiştir.

Merkezi İşlem Biriminin bir diğer önemli işlevi de yönlendirme işlemini gerçekleştirmek ve alınan pakelerin, ilgili portlar üzerinden gönderilmeden önce tutuldukları tampon belleklerinin denetimini sağlamaktır.

Hierarşik yönlendirmenin esas alındığı bu tümdevrede X.121 [5] adresleme standardı ve TURPAK üzerinde kullanılmakta olan numaralama sistemi temel alınmıştır. X.121 standardında 14 basamak olarak tanımlanmış olan adres bilgisinin ilk 4 basamağı ITU-T tarafından belirlenmiş ve her ülke için bir sayı verilmiştir. Kalan 10 basamağın kullanımı paket ağ işleten kuruluşların kullanımına bırakılmıştır. TURPAK'ın üzerine kurulmuş bulunduğu DNS [6] paket santrallerinde ise halen Türkiye ve ABD'de kullanılmakta olan numaralandırma sistemine eş bir numaralama planı uygulanmaktadır.

Tablo 2. Durum Listesi.

Durum	Tanımı
S1	Bağlantı kesik
S2	Bağ kurma.
S3	Çerçeve reddi.
S4	Bağlantı kesme isteği.
S5	Bilgi aktarımı.
S6	REJ çerçevesi gönderildi.
S7	1 çerçevesi için onay bekleniyor.
S8	Yönlendirici meşgul.
S9	Öteki uçtaki aygıt meşgul.
S10	Yönlendirici ve öteki uçtaki aygıt meşgul.
S11	S7 ve S8
S12	S7veS9
S13	S7veS10
S14	REJ çerçevesi gönderildi öteki uçtaki aygıt meşgul.

Yukarıda açıklanan numaralama sistemi doğrultusunda, bir X.25 adresi aşağıdaki gibi yazılabilir:

$$P + \text{DNIC} + \text{DNPA} + \text{DCO} + \text{EPN}$$

Yukarıda gösterilmiş olan numaralama sisteminde P (prefix) ITU-T tarafından X.121 içinde tanımlanmış olan uluslararası arama kodu olan 'O' gösterirken DNIC (Data Network Identification Code) ITU-T'nin her bir paket ağ şebekesi için atamış olduğu uluslararası ağ tanıma kodunu göstermektedir ve 4 basamaktan oluşmaktadır. DNPA (Data Numbering Plan Area), DCO (Data Central Office Code) ve EPN (End Point Number) ise paket ağ işletmecisi tarafından atanan numaralardır. Telefon sistemlerindeki alan koduna karşılık gelen DNPA 3 basamaklıdır. Şehir içi santral kodlarına karşılık gelen DCO da 3 basamaklı iken santral içindeki abone numarasını belirten EPN 4 basamaklıdır.

Alıcı birim içindeki tutucular içinde yer alan L tutucusunun içine yazılan değer, yönlendiricinin yukarıda açıklanmış olan adresleme planı içinde yerleştirildiği sıradüzensel noktayı belirtir. L yazmacı

içme yazılan bu değer "0" en yüksek sıradüzensel noktayı belirtir. Bu durumda yönlendirici ülke yada paket ağ girişlerine yerleştirilmişlerdir ve yönlendirme işlemi DNIC değen dikkate alınarak yapılır.

İkinci düzey şehir yada alan girişleridir ve L değeri "1"dir. Yönlendirme işlemi temel olarak DNPA değeri ile gerçekleştirilir. L'nın "2" değerini aldığı üçüncü düzey, paket ağ santralleri arasında yer alır ve DCO yönlende dikkate alınan adres bölümü durumuna gelir. Dördüncü ve son düzeyde L "3" olurken yönlendirici işlemlerini EPN'İ dikkate alarak yapar.

Yönlendirme işlemi tamamlandıktan sonra, karar verilen çıkış portu çerçeve içindeki LCN ile birlikte Yönlendirme Tablosu'na yazılır. X.25 paketlerinin tamamında adres bilgisi olmadığı için adres bilgisi taşımayan paketler için LCN'ler yönlendirme için kullanılır.

Yönlendirme işlemi tamamlanan paketlerin tampon belleğe yazımı da bitirdikten sonra çerçeve içindeki LCN, yönlendiricinin LCN'leri ile değiştirilir. Yönlendirici 8 adet bağlantı kurabildiği için bu LCN'ler sabit olarak seçilmişlerdir. Alınan çerçeve hangi bağlantı numarası ile geliyorsa ilgili LCN, tampon bellek üzerindeki çerçevenin LCN bölgesine yazılır. Bağlantı numaralarına göre belirlenmiş olan LCN'ler Tablo 3'te listelenmiştir.

Tablo 3. Mantıksal Kanal Numarası.

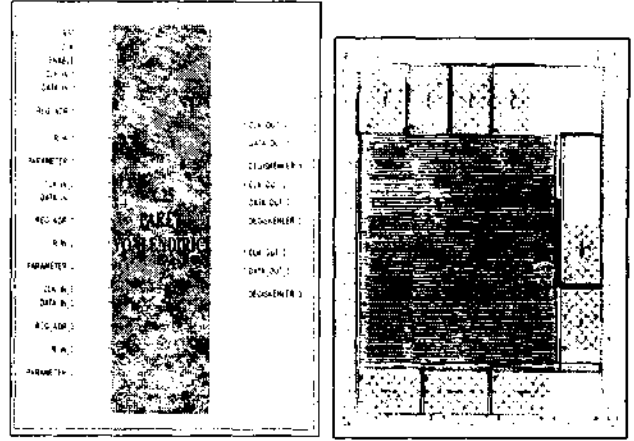
Bağ	Yeni Mantıksal Kanal											
L1	0	1	0	0	0	1	0	0	0	1	0	0
L2	0	1	0	0	0	1	0	1	0	1	0	1
L3	0	1	0	0	0	1	1	0	0	1	1	0
L4	0	1	0	0	0	1	1	1	0	1	1	1
L5	0	1	0	0	1	0	0	0	1	0	0	0
L6	0	1	0	0	1	0	0	1	1	0	0	1
L7	0	1	0	0	1	0	1	0	1	0	1	0
L8	0	1	0	0	1	0	1	1	1	0	1	1

Yönlendiriciye gelen bilgi paketlerinin, yönlendirme işlemlerinin bitmesini ve gönderilecek olan portun boşalmasını beklediği yer olan tampon bellek üç ana parçaya ayrılmıştır. Her bir parça bir giriş portuna aittir. Diğer bir deyişle her port, aldığı bilgi paketlerini kendisine ayrılmış olan tampon bellek bölgesine yazar. Portlar için ayrılan bölgeler, 256 byte'lık 3 bilgi paketini saklayabilecek büyüklüktedir. Her bir port için ayrılan tampon bellek alanının toplam uzunluğu, paketin başına eklenmiş olan LCN ve PTI gibi bilgilerle birlikte 777 byte'tır. Toplam tampon bellek büyüklüğü ise 2331 byte'tır.

SONUÇ

Bütünüyle VHDL ile tasarlanan yonganın tasarımı sırasında, Synopsys Inc. tarafından üretilen VHDL araçları kullanılmıştır. ES2 0.7 um CMOS

teknolojisinin kullanıldığı ve 71 mm² alan kaplayan yonganın serimi, CADENCE Inc. tarafından üretilen ilgili yazılımlar aracılığıyla gerçekleştirildi. 75 pinden oluşan yonganın genel görünümü ve serimi Şekil 6'da verilmiştir. Yapılan literatür çalışması sırasında X.25 protokolü için tasarlanan bir yönlendirici yongaya rastlanmaması ve bütünüyle VHDL ile tasarlanması nedeniyle Dünya'da kavram ve tasarım olarak bir ilk olma özelliği taşıyan, üzerinden akan bilginin yonga üzerindeki geçici belleklerde saklanması nedeniyle veri akışında tam bir güvenlik sağlayan yonga ile ilgili daha ayrıntılı bilgi [7]'de bulunabilir.



(a) Genel Görünüm.

(b) Serim.

Şekil - 6

KAYNAKÇA

- 1.) CC/TT Recommendations X.1-X.32, Blue Book, VolumeVMI, Fascicle VIII.2, 1988.
- 2.) K.Bolding,S.C.Cheung, ve diğerleri,"The Chaos router Chip:Design And Implementation of an Adaptive Router", *Proceedings of VLSI'93*, IFIP 1993, s. 311-320.
- 3.) G.A.Boughton, *Artic Routing Chip*, MIT Laborator for Computer Sciences, Computation Structures Group Memo, 7 Mart 1994.
- 4.) A.Mu.J.Larson, ve diğerleri,"A 9.6 Gbyte/s Thorougthput Plesiochronous Routing Chip", *COMPCON'96*, 25-28 Ocak 1996, Santa' Clara, ABD.
- 5.) CCITT Recommendations X.40-X.181, Blue Book, VolumeVMI, Fascicle VIII.3, 1988.
- 6.) Northern Telecom Ltd., *DPN Data Networking System Reference Handbook*, 1986.
- 7.) O.V.ERDAĞ, *The Design Of An X.25 ftouter VLSI Chip*, Elektrik-Elektronik Mühendisliği Master Tezi, Temmuz 1997, ODTÜ, Ankara.

Lokal Hesablama Şebekelerinde İnformasiyanın Mehviliyinin Şahlanmasının Bezi Meseleleri

E. A. BALIYEV, S. H. ALIYEB, M. M. GENBEROV
Azerbaycan Teknik Üniversitesi
BAKİ

If we pay attention to the fact of numerous of netv/orks v/e can say that the keeping of state, public and private secrets is an actual problem. The operation systems using in netv/orks have some possibilities connected with keeping of date at present. But these possibilities not always satisfy users.

The first stage of the offered method of the problem of the keeping and sending safety of the date if connected with the identify of terminal users. In order to identify the users it is used passwords (the keeping . passages). The structure of the identification file, composed by the netv/orks administrater. consist of the identification number, the keeping code, the type of the allowing transaction and the categoru of the offer right. To enter the system at first the user has to gaher the Identification number. then the keeping code. To carry out the conerete transaction of the keeping code it is important to define the user's authority right. The following stoge of the pointed keeping method foresees the organization of the cruptographic system. The carrying out of the operations of the codes uncovering in the sending and receiving process can gNe the possibility to organize the higtion degree of the keeping of the date. The effica of the pointed method is defined by the complication of the codification algorithm connected with the codification key.

Son zamanlar lokal hesablama şebekeleri (LHŞ) verilenlerin irnai, şahlanması, ötürülmesi ve s. sistemlerinin ayrılmaz bir hissesi hesab olunur. Diğer terefdən layihelendirmenin avtomatlaşdırılması, müxtəlif sənaye sahələri üzrə idarəetmə sistemləri (AIS), roboto-tehniki sistemləri, bank işləri, kağızsız sənədlər ahırımın həyata keçirilməsi, elektron poçtası və s. sistemlərində müasir LHŞ-leri əsas rol oynadığından xüsusi əhəmiyyət kəsb edir.

Kompüterlerin şebekələrə qoşulan abonent və stansiyaların tündən verilenlərin ötürülmə mühitindən və sairə əsli olaraq hal-hazırda müxtəlif növlü LHŞ-ler fəaliyyət göstərir.

Birinci qrupa küllü sayda istifadəçilərə yönləndirilmiş 2400-19200 bod sürət ilə 100-500 m məsafədə informasiya ahırımın ötürülməsini təmin edən. olduğca kiçik dəyərə malik verilenlərin ötürülmə

şebəkəsi (VÖŞ) sisteminin köməyi ilə əsasən fərdi kompüterləri özündə birləşdirən LHŞ-ler aid olunur.

İkinci qrupa fərdi kompüterlərdən başqa mikroprosessor komplektlərini birləşdirən. layiheləndirmənin avtomatlaşdırılması vasitələri avadanlıqları, sənədli informasiya emalı. o cümlədən də elektron poçtası vasitələrinə sazlanmış LHŞ-ler aid olunur.

Üçüncü qrupa mikroprosessor komplekslərini. mikro kompüterləri. elmi-tədqiqat işlərinin layiheləndirilməsinin avtomatlaşdırılmasında istifadə olunan orta sinif bilgisayarları, çevik avtomatlaşdırılmış istehsal modulunu, o cümlədən roboto-tehnikanın istifadəsi ilə mürekkəb istehsal proseslərinin idarə olunmasını özündə birləşdirən LHŞ-ler aid olunur.

Dördüncü qrupa daha mürekkəb idarə sistemlərində, yeni nəhəng istehsal sahələri. xalq təsərrüfatının müxtəlif sahələrində istifadə olunan evvelki LHŞ-lərin əsas elementlərini özündə birləşdirən bütün mövcud kompüterləri özündə birləşdirən LHŞ-ler aid olunur. Bu qrup LHŞ- lərdə yüksək dəyərə malik 10 Km məsafədə 10-50 Mbod sürətli ilə informasiya ahırımın ötürülməsini təmin edən müxtəlif VÖŞ-lərdən istifadə olunur. Belə LHŞ- lər vilayət, rayon və böyük şəhərin lokal şebekələrindən fərgənir.

Qeyd olunanları nəzərə almagla LHŞ-lərin işlənməsinin bezi tərəflərini əsas tutmaqda, yəni verilenlərin gəbulu, emalı, ötürülməsi, saxlanması. paylanması, və s. kimi məsələlərə müasir kompüterlərin evez edilmez rolü olduğundan INTEL 80386, 80486 MP-leri əsasında yaradılmış LHŞ-ler xüsusi əhəmiyyət kəsb edir.

Kompüter şebekələrinin istifadəçilərinin çox saylı olması faktı nəzərə alınarsa onlarda dövlət, içtimai və şəxsi şiflərin təhlükəsizliyi və mühafizəsi problemi öz aktualığı ilə fərgənir. Kompüter şebekələrində təhlükəsizlik məsələsi ilk növbədə verilenlərin təsadüfi və ya gərəzli açıqlanması, həmçinin onların icazəsiz dəyişdirilməsi və ya legv edilməsi ilə bağlıdır. Bu məsələlərin həlli üçün aşadək mümkün variantları göstərmək olar.

- şebəkənin istifadəçiləri ondan istifadə etməzden evvel identifikasiya olunmalıdırlar:

- şebekenin administratoru (şebekenin işletmesine icaze veren şexs) istifadəçilerin faaliyyətinin səlahiyyətli olub-olmadığına nəzarət etməlidir;
- istifadəçilerin ixtiyarsız istifadəsindən qorunmaq üçün verilənlər, aparatlar və programlar təminatı kilidlərlə (açarlarla)-bağlılığa (privacy locks) malik olmalıdırlar;
- verilənlər bərpə oluna bilən, yoxlanıla bilən və pis məqsədlərdən qoruna bilən olmalıdır;
- verilənlərin ötürülməsi konfidensial olub səhvlərdən mühafizə olunmalıdır.

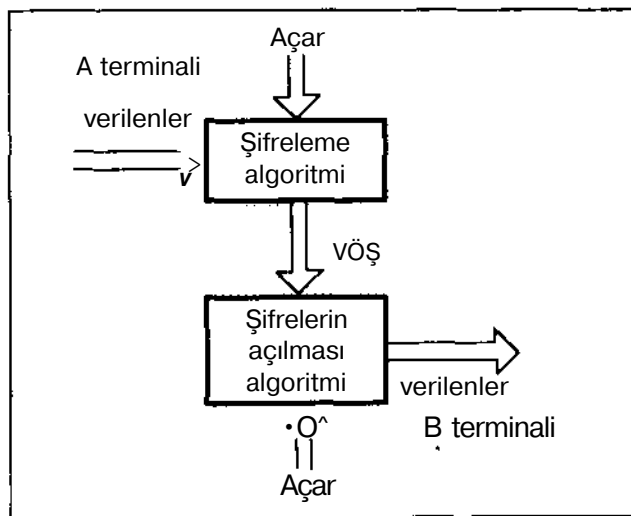
Qeyd etmək lazımdır ki, hal-hazırda istismarda olan istər lokal, istərsə də global kompüter şəbəkələrində istifadə olunan şəbəkə emeliyyat sistemləri verilənlərin mühafizəsi ilə əlaqəli müəyyən imkanlara malikdirlər. Lakin bu imkanlar istifadəçiləri heç də həmişə qane etmir. Yuxarıda göstərilən məsələlərin hər biri ayn-aynı zamanda böyük bir problem olduğundan tədricən olunan məqsəddə mühafizə məsələsinin həllinin program və aparat vasitələrindən birgə istifadəsinin bir variantına baxılır.

Verilənlərin saxlanması və ötürülməsinin təhlükəsizliyi məsələsinin təklif olunan usulünün ilkin mərhələsi terminal istifadəçilərin identifikasiyası ilə bağlıdır. İstifadəçilərin identifikasiyası üçün mühafizə kodlarından-parollardan istifadə olunur.

İstifadəçinin identifikasiya edən nömrə	mühafizə kodu (parol)	icazə verilən tranzaksiya tipləri	müraciət hüquqı ategoriyası
---	-----------------------	-----------------------------------	-----------------------------

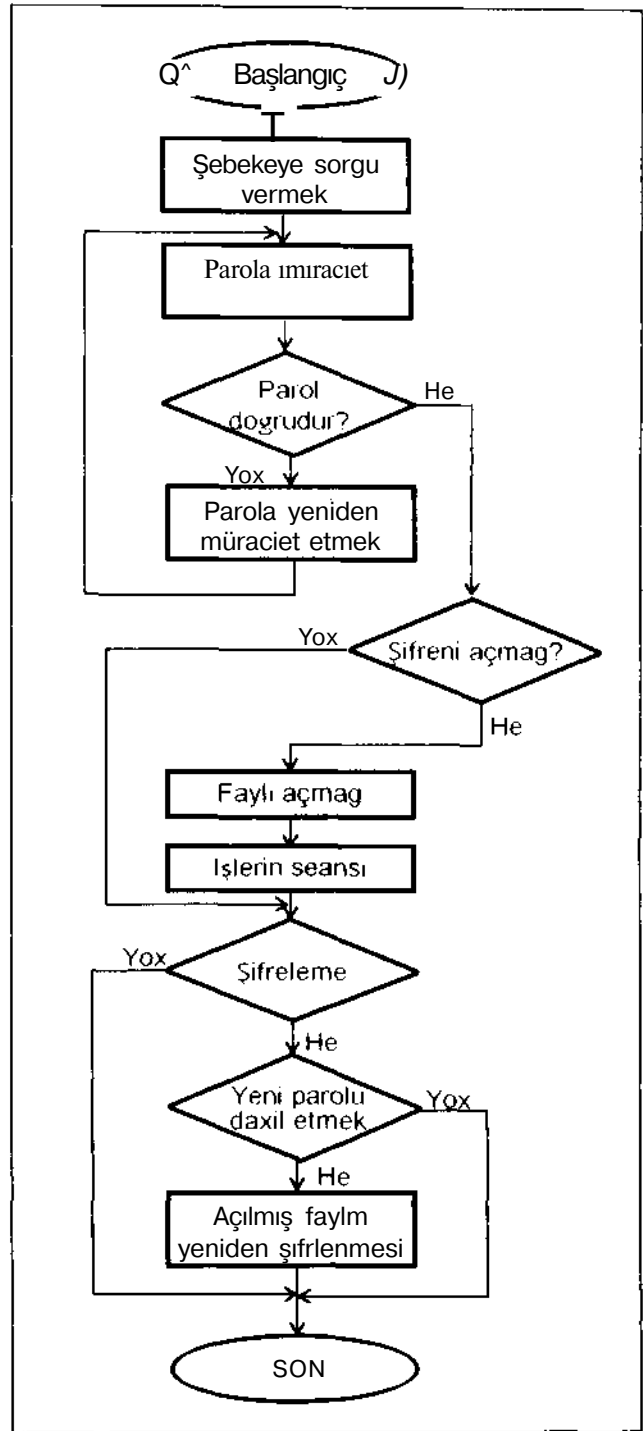
Şəkil 1. "İdentifikasiya faylının strukturu"

Sadə halda məsafəli müraciət serverlərinə, müəyyən kategoriyalı serverlərə və lazım gələrsə məsafəli müraciət serverinin bütün terminallarına eyni parol verilir.



Şəkil 2 "Kriptografiya sisteminin strukturu"

Verilənlərin mühafizəsi səviyyəsinin artırılması məqsədi ilə yaxşı olar ki, hər bir terminal-istifadəçi öz mühafizə koduna malik olsun.



Şəkil 3 "Şəbəkədə informasiya məxfiliyinin təmin olunması algoritminin sxemi"

Sisteme daxil olmaq üçün istifadəçi əvvəlcə identifikasiya nömrəsini, sonra isə mühafizə kodunu yığmalıdır. Kompüter daxil edilən kodun doğruluğunu və tranzaksiyaya yerinə yetirmək üçün istifadəçinin səlahiyyəti olub-olmadığını yoxlayır. Bu

meqsədlə şəbəkənin administratorunun təşkil etdiyi identifikasiya fayhının strukturu Şəkil: 1-də göstərilmişdir

Mühafizə kodunun yoxlanması nəticəsində müraciət hüququ kategoriyası müəyyənləşdirilir ki, bu da icazə verilən fealiyyət dairesini təyin edir.

Baxılan mühafizə usulünün növbəti mərhələsi isə kriptografiya sisteminin təşkilini nəzərdə tutur. Verilənlərin ötürülməsi və qəbulu prosesində şifrələmə/şifrələrin açılması əməliyyatlarının aparılması verilənlərin mühafizəsinin daha yüksək səviyyəsinin təşkilinə imkan verir. Kriptografiya sisteminin ümumi strukturu Şəkil: 2-də göstərilmişdir

Verilənlərin mühafizəsinin aparat üsuluna aid olan bu mərhələnin reallaşdırılması mikroprosessorların və ya xüsusi şifrələmə mikro sxeminin istifadə olunması tələb edir. Təklif olunan üsulun effektivliyi şifrələmə açarından asılı olan şifrələmə algoritminin mürəkkəbliyi ilə müəyyən olunur. Açar kimi bitlər və

ya simvollar ardıcılığından istifadə olunur ki, onların da mürəkkəbliyi dərəcəsi məxfiliyə olan tələbatdan asılı olur. İstər parolların, istərsə də açarlar açarların tez-tez dəyişdirilməsi verilənlərin mühafizəsinə artırmağa imkan verən faktorlardan sayıla bilər.

İnformasiyanın məxfiliyinin təmin olunması algoritmi işlənilərkən (Şəkil: 3) şifrələmə istifadə olunma üsulları təyin olunma növləri yeni fərdi yaxud qruplaşma şəklində müəyyən olunur. Bunlar hamısı şəbəkələrin istifadəçilərindən asılıdır.

İstifadə olunan ədəbiyyat:

1. Mənsita İ. "Mikro EHM-lərin aparat vasitələri" Moskva, Mir 1988 (Rus dilində)
2. Kato M., Tokoro E. "EHM şəbəkələrinin yaradılması", Moskva, Mir 1988 (Rus dilində)
3. Korrobis Dj. D. BASE III PLUS-da programlaşma Moskva, 1991 (Rus dilində)
4. Bəliyev E. A., Paşayev İ. S., Şirinov E. İ., Genbarov M. M. "LHŞ-nin işlənməsinin bəzi məsələləri" Bakı, AZTU (Rus dilində)

Tekstil Dokuma İşlemleri için bir LAN ve Barkodlu Üretim Takip Sisteminin Geliştirilmesi

Haluk GÜMÜŞKAYA, Yusuf ULCAY*

TÜBİTAK MAM, Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü
P.K. 21, 41470, Gebze, Kocaeli

* Uludağ Üniversitesi, Tekstil Mühendisliği Bölümü, Görükle, Bursa
email: haluk@mam.gov.tr, ulcay@mf40.bim.uludag.edu.tr

ABSTRACT

This paper outlines the construction of a LAN (Local Area Network) system which forms the basis of a knowledge-based textile CIM (Computer Integrated Manufacturing) system, especially a monitoring and management system for textile weaving processes. In the development of the network, hardware choices and programming methodologies; work-in process tracking with barcode and the design of a future intelligent management system that will effectively utilize the distributed data bases in the mill are taken into consideration. The network software has a scaleable, 32-bit high-performance Client/Server architecture running under Windows NT.

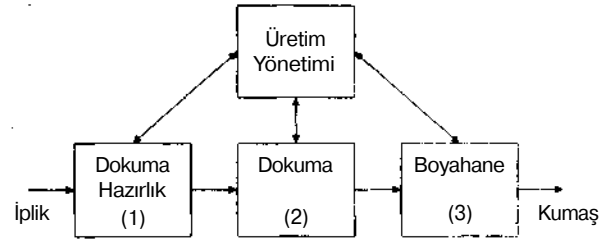
1. Giriş

CIM'in (Computer Integrated Manufacturing) bir ögesi olan ve genelde MRP II (Manufacturing Resource Planning) diye adlandırılan yazılımlar, bir şirketin genel yönetimine, fabrika yönetimine ve üretim yönetimine destek vermesi amacıyla kullanılır. CIM'in diğer bir ögesi olan SCADA (Supervisory Control and Data Acquisition) ve benzeri sistemler ise, proses gözleme (monitoring), kontrol etme ve doğrudan proseslerle bağlantıyı sağlarlar [1], [2]. Bugün bilhassa ülkemizde tekstilde üretim takibine yönelik CIM çözümleri çok azdır. Yurt dışında bazı çalışmalar bulunmakla beraber, ticari değerlerinden ve gizliliklerinden dolayı bunların akademik ortama yansımaları yok denecek kadar azdır [3], [4]. Bu çalışmada çerçevesi «çizilmeye çalışılan bir Üretim Takip Sistemi (ÜTS) ile, tekstile yönelik MRP II ve SCADA türü yazılımlarda bulunması gereken bazı önemli prensipler belirtilmektedir. Bu makalede, TÜBİTAK MAM ve Uludağ Üniversitesi Tekstil Mühendisliği Bölümleri ile Bursa'da bir tekstil fabrikasının ortaklaşa çalışmaları olan, *Tekstil Üretim Takip Sistemi (TÜTS)* projesinin dokumaya yönelik kısmı (DÜTS—Dokuma Üretim Takip Sistemi) sunulmaktadır. Bu proje ile hedeflenen, dokuma hazırlık, dokuma ve boyahane işlemlerinin tamamı için bir TÜTS'nin geliştirilmesidir.

2. Tekstil Üretim Takibine Yönelik bir LAN

Üretimde verimlilik için üretim aşamalarının, işçilik, malzeme ve ekipmanın takibi ve kontrolü gerekmektedir. ideal bir ÜTS, üretim ve işçilik zamanını dakikası dakikasına takip ve rapor eder. Yine tezgahların işleme/durma, arıza/bekleme, üretime yeniden başlama bilgilerini anında tespit edip performansları hakkında kesin ve doğru bilgileri istatistiksel dökümler halinde sunar. ÜTS'de ayrıca üretim için tüketilen hammadde, miktar ve cins olarak anında belirlenir. Üretim takibinde işçilikten, malzemeye ve kullanılan hammaddeye kadar kapsamlı bilgilerin, ÜTS sayesinde maliyet muhasebesi bölümüne anında ve doğru olarak aktarılması istenir. Böylelikle, üretilirken tüketilen vakit ve nakitin boyutları hakkında en doğru bilgi kazanılmış ve verim ölçülmüş olur. Bu çalışmada yukarıda belirtilen, bir ÜTS'den genel olarak istenenler göz önüne alınarak, tekstile yönelik bir ÜTS düşünülmüştür.

Dokuma hazırlık, dokuma ve boyahane bölümlerine sahip birçok tekstil fabrikası için, tekstil işlemleri Şekil 1'deki blok diyagram ile gösterilebilir.



ŞEKİL: 1 Tekstil kumaş üretimindeki işlemler.

Fabrikanın girişi iplik olup son ürün ise kumaştır. Üretim Yönetimi, fabrika yönetim işlerinin yapıldığı yerdir ve üç tekstil işleminden ayrı bir binada yer alır. Ayrıca bu üç tekstil işlemi, genelde ayrı binalarda olur ve aynı fabrikada bulunmayabilirler de. Yani, bir firma dokumahaneye sahip olup boyama işlerini başka bir yerde yaptırabilir. Bu çalışmadaki fabrika, bu üç bölüme sahip olup bunlar birbirine yakın ayrı binalarda bulunmaktadır.

Bu çalışmada, iplik ambarı, çözgü ve haşıl işlemleri, *Dokuma Hazırlık* bölümünde; dokuma işlemi, çığ kumaş hata kontrolü ve çığ kumaş ambarındaki işlemler ise *Dokuma* bölümünde ele alınmaktadır. Dokuma Hazırlık ve Dokuma bölümlerindeki işlemlerin detaylandırılmış iş akışı ve bu işlemleri gözlemek için düşünülen LAN sistemi Şekil 2'de görülmektedir. Aşağıdaki bölümlerde sırasıyla, yukarıda belirtilen işlemler ve bunları takip etmek amacıyla önerilen LAN altında çalışan, barkod tabanlı ÜTS'deki bilgisayarlı iş noktaları tanıtılacaktır.

Bir dokuma fabrikasında üç tür genel stok bulunur: Hammadde (iplik ve haşıl işlemi için kimyasal maddeler), süreçteki iş (*Work In Process -WIP*) (çözgü ve haşıl sonrası leventlerdeki iplik, dokuma sonrası çığ kumaş topları) ve boyamadan önceki ürün (hata kontrolünden sonraki çığ kumaş topları). Dokuma fabrikasındaki stok ve WIP kontrolü için barkodlu sistem kullanılmaktadır [5]. Şekil 2'de görülen 6 bilgisayar terminali (PC1-PC6) ile belirtilen noktalarda, herbir bilgisayara bağlı barkod okuyucu ve yazıcı bulunur. Bu sistemdeki işlemlerin büyük bir kısmı, yarı-otomatik olup kullanıcının el ile müdahalesini gerektirir. Bunun nedeni, bazı işlemlerin tam-otomatik hale getirilememeleri; aynı zamanda, geliştirilen sistemin, ucuz, pratik ve birçok fabrikaya uygun ve esnek olmasının hedeflenmesidir.

3. Barkodlu İş Emri

Barkod etiketleri ile İplik Ambarında depolanan ipliklerin üretime girmeleri için verilen iş emrinde, yapılacak işlemleri belirten barkod bulunur (Şekil 2'deki A noktası). Her işe verilecek tek bir barkod bilgisi ile, ürün takibi kolaylaşmakta ve hızlanmaktadır. Bir işin, üretimin hangi aşamasında olduğu öğrenilmek isteniyorsa, sadece barkodlu iş emrindeki barkod bilgisi taranır ve bu bilgiye hızlı bir şekilde ulaşılır.

İş emrine herbir işlemde sonra, yapılan işlem kodu, işçinin ID numarası, var ise hata kodu (işlem, iplik, kumaş ve makine gibi), uzunluk ve ağırlık gibi bilgiler, yapılan işlemlere göre, barkod olarak veya el ile, eklenir ve bir sonraki iş noktasına iş aktarılır. A noktasından sonraki B, C ve D noktalarında, iş emrine yeni barkod bilgileri, sırasıyla PC2, PC4 ve PC5 terminallerinden eklenir. Yani, iş emri ile verilen ilk barkoda, 3 noktada yeni barkodların eklenmesiyle, toplam 4 barkod bilgisi, işin bitme noktası olan C'de iş emri kağıdında bulunur. En son D noktasında ise, iş emrindeki 4 barkod bilgisi taranır ve Çığ Kumaş Ambarına gidecek partiye (veya her bir kumaş topunun üzerine) tek bir barkod bilgisi basılır.

4. İplik Ambarı

Dokuma Hazırlıkta ilk bölüm olan iplik Ambarında, firmalardan gelen iplik paketleri, bir işleme girmeden önce stoklanır. Dışarıdan gelen hammadde iplik paketleri stoğa alınırken, barkod yazıcı ve el ile bilgisayar kaydı yapılır.

Bilgisayarlı ve barkodlu bir ÜTS'de, önce ambarların fiziksel durumu ve firmanın yapısına göre mal giriş/çıkış işlemleri ile iş akışının çok iyi düşünülmesi ve ona göre sistemin tasarlanması gerekir. Ambarın fiziksel durumunun ve stok hareketlerinin düzenlenmesi ve bir standarda uygun hale getirilmesi, bilgisayarlı ve barkodlu sistemin güvenilirliği açısından çok önemlidir. Böyle bir ambar stoğundaki değişiklikler ve iş akışı çok hızlı ve güvenilir bir şekilde takip edilebilir. Ambardaki barkodlu etiketlerle depolanmış paketlerin sayımı, RF tekniğiyle çalışan taşınabilir el terminaleri ile hızlı ve güvenilir bir şekilde yapılır. Bu sayede, mal sayımı günlük bir işlem haline gelmekte ve bilgisayar kayıtları her zaman o anki stok durumunu tam olarak göstermektedir. Bu yüzden, yıllık envanter sayımı için uzun zaman ayırmaya gerek kalmaz.

5. Çözgü ve Haşıl

Barkodlu iş emrine göre, iplik kutuları iplik Ambarından alınıp Çözgü işlemine girer. Çözgü'de hangi bobinlere (levent) hangi ipliklerden ne kadar sarıldığı bilgisayar ile takip edilmektedir. Çözgüden geri iade edilmiş iplikler, iplik Ambarı terminalinden (PC1) takip edilir. Bu ipliklerin bazıları tekrar kullanılabilir (pamuklu gibi), bazıları ise kullanılamaz durumdadır. Benzeri bilgilerin bilgisayarda olması ile fire hesaplamaları yapılabilecektir.

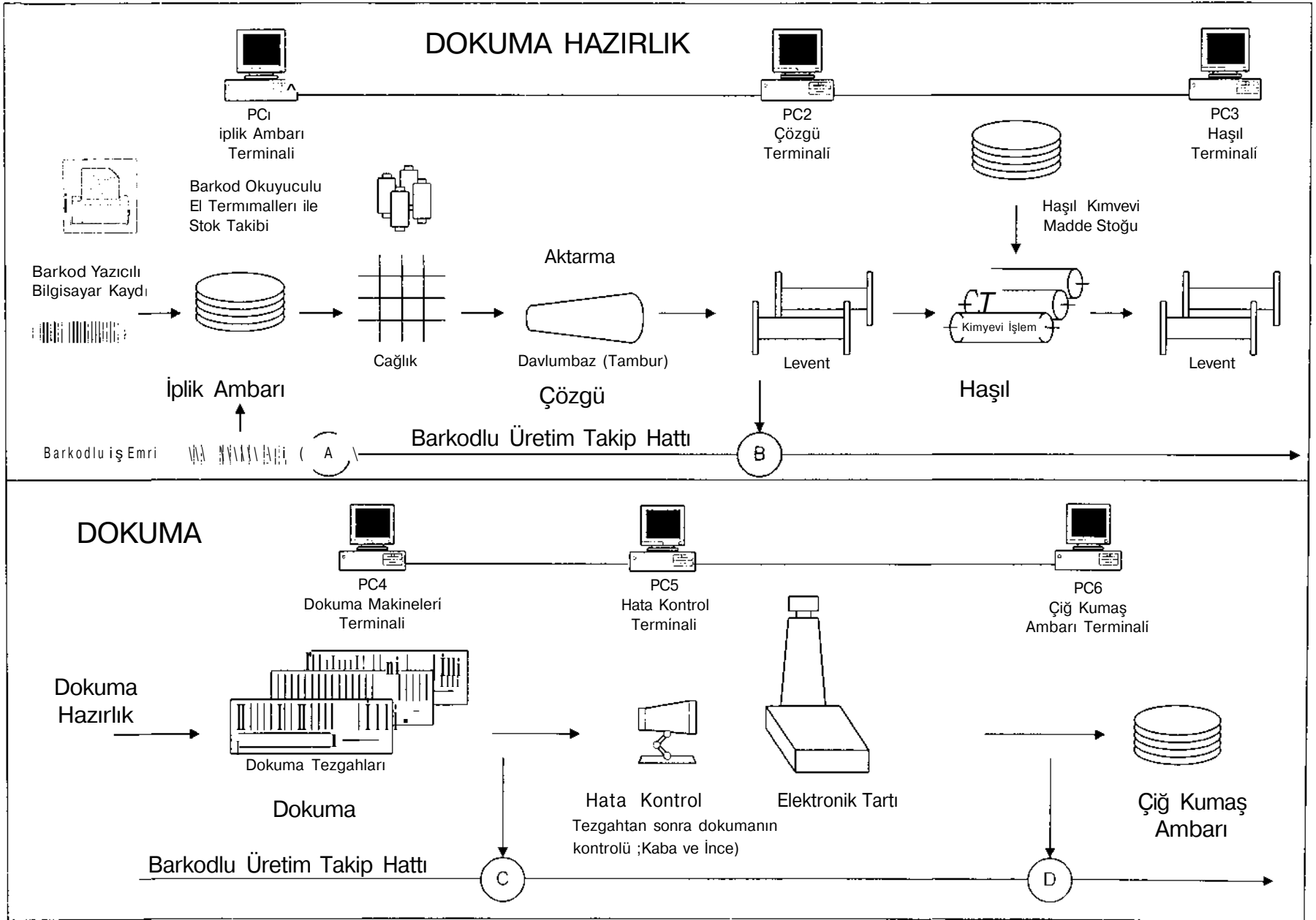
Çözgüde, iplik kopması ile bağlanması arasındaki geçen zamanlar verimlilik hesaplarında kullanılacağı için, Çözgü boyunca, iş kesinti zamanları bilgisayara elektronik bir yol ile okunmaktadır. Verimlilikte önceden belirlenen hedeflere erişmek için bilgisayar kullanılabilir. Örneğin, Çözgü verimliliğinde %85 hedefinin üstündeki işler için, işçilere mükafat, altında olanlar için uyarı verilebilir. Bu sayede çalışanların iş durumları takip edilerek randımanlarının artırılması sağlanabilir.

Çözgüden sonra, leventlerdeki ipliklerin, kimyasal sıvılardan geçirilerek tekrar leventlere sarılma işlemi olan Haşıl takip edilmekte ve maliyeti hesaplanmaktadır. Hangi bobinler ve iplikler haşıl işleminden geçti gibi bilgiler bilgisayarda bulunur. Çözgü işlemi bazı ipliklere uygulanmayacağı ve isteğe de bağlı olabileceği için, iş emrinde bu işleme ayrılan bir barkod alanına gerek duyulmamıştır, iş emrinde Haşıla ilgili yere, gerekli bilgilerin el ile girilmesi uygun görülmüş ve bu ayrıca maliyeti de düşürmüştür.

6. Tezgahlarda Dokuma

Haşıldan sonra, iplikler dokuma tezgahlarına gelir ve burada dokunur. Şekil 2'de gözükten PC4, dokuma makinelerinden veri toplayan bir bilgisayar terminaline veya birden çok terminale karşı gelir.

Fabrikada bulunan tekstil makinelerinin hancı cihazlarla haberleşmesinde standart bir protokole sahip olma durumunda özel SCADA çözümler bulunur [2]. Standart bir protokole sahip olmamaları



ŞEKİL: 2 Dokuma Hazırlık ve Dokuma işlemleri bilgisayarlı ve barkodlu ÜTS blok diyagramı

durumunda, eğer makinelerin protokolleri kullanım kitaplarında bulunuyorsa, bu cihazlar için protokol yazmak mümkündür, fakat bu pahalı bir yöntemdir. Makinelerinin herhangi bir haberleşme özelliğinin bulunmaması veya protokol dokümanlarına erişilememesi durumunda, DÜTS'ne gerekli bilgilerin toplanabilmesi amacıyla PLC (*Programmable Logic Control/er*)/VME (*Versa Modular Euro bus*) veya RTU (*Remote Terminal Unit*) kullanılabilir. Bu bahsedilen yöntemler pahalı oldukları için, projenin ilk aşamasında, dokuma makinelerinin verileri el ile toplanmaktadır. Dokuma makinesini kullanan işçi veya makinelerden sorumlu bir teknisyen, her yeni bir işe başlangıçta ve iş sonunda, makineye karşı gelen bilgisayar terminalinde çalışan programa, gerekli verileri (tezgah numarası, işçinin ID'si, lot numarası, dokunan kumaşın metresi, dokuma tarihi, renk kodu gibi bilgileri) el ile girer. Dokuma bittikten sonra ise, iş emrine, yapılan işlemlerle ilgili bir barkod bilgisi basar. Her bir bilgisayar terminalinde çalışan program, bir grup makine veya bir dokuma makinesinin verisini toplar ve LAN'daki bir üst düzey makineye aktarır.

7. Hata Kontrol ve Çiğ Kumaş Ambarı

Üretilen dokumaların fiziki kaba hata kontrolü el ile yapılır. Kaba kontrolden geçen dokumalar, yine el ile ince nokta hata kontrolünden geçer. Çiğ kumaşlar uzunluk ve ağırlık bilgileri ile bilgisayara girilir. Hata kontrolünden geçen ve elektronik tartı ile tartılan dokumalar. Çiğ Kumaş Ambarında boyahaneye gitmek üzere stokların iş emrine, bu son D noktasında, Hata Kontrol barkod bilgisi eklenir ve son olarak, iş emrindeki basılı 4 barkod bilgisi taranır ve Çiğ Kumaş Ambarına gidecek partiye (veya her bir kumaş topunun üzerine) tek bir barkod bilgisi basılır.

8. Üretim Planlama

Yukarıda anlatılan yöntemlerle, dokuma fabrikasına ait aşağıdaki bilgiler toplanmaktadır:

- Üretilen kumaş miktarları, kalite bilgileri
- Kusurlu kumaş miktarları, fireler, yeniden işlem gerektiren parçalar
- Üretimin verimliliği (kullanılan malzeme miktarları, toplam makine duruş zamanları, nedenleri gibi)
- Üretilen ürünlerin kodlanması ile geriye dönük bilgi edinme (üretim hattı ile ilgili verilerin bu ürünler ile ilişkilendirilmesi)
- İş ilerlemelerinin barkodlu sistem ile merkezi takibi
- Hazırlık ve üretim zamanları, atıl zamanlar
- Personel çalışma saatleri, randımanları
- Malzeme ihtiyaçları
- Dokuma tezgah arızaları ve bakım bilgileri
- İstatistik amaçlı bilgiler

Veri tabanında bulunan yukarıdaki bilgilerle, gant şemaları, iş akış animasyonu ve diğer grafik yöntemlerle üretim planlaması yapılabilecektir. Sistemden her zaman alınacak bu bilgiler doğrultusunda, yöneticiler ve mühendisler önemli ve çok değerli bir zaman tasarrufu sağlayacaktır. Bu

sistem, malzeme ihtiyaçlarını en hızlı ve en iyi bir şekilde belirleyerek, hammadde, yarı mamul ve mamul stok devir hızını arttıracak, işletme sermaye ihtiyacını mümkün olduğu kadar azaltacak ve müşteriye en uygun tarihte malı teslim etmeyi hızlandıracaktır. Üretim planlamanın orta ve uzun vadeli amacı ise, yeni kapasite artırımlarına zamanında karar verilmesine yardımcı olmaktır. Ayrıca bu sistem sayesinde, çeşitli senaryolar üretip çeşitli stratejiler geliştirmek ve bunların olası sonuçlarını incelemek mümkün olacaktır.

9. Sonuç

Bu çalışmada tanıtılan DÜTS ile üretim hattının kalitesi devamlı gözlemlendiği ve kaydedildiği için bu sistem ISO 9000 için çok iyi bir temel teşkil etmekte olup yukarıda belirtilen faydaları sağlayacaktır. Dokuma hatalarının bilgisayar ile sürekli toplanması bize istatistik olarak önemli veriler sağlamaktadır. Bu verilerden, ve ayrıca çalışan işçi ve mühendis uzmanların görüşlerinden hareketle, kumaş hatalarını sınıflandırmaya ve bulunmasına yardım etmeye yönelik bir uzman sistem geliştirilmiştir [6].

Fabrika verilerinin güvenilir bir şekilde toplanması, saklanması ve daha sonra hızlı bir şekilde erişilip kullanılması, gelişen donanım ve yazılım teknolojilerinden çok etkilenmektedir. DÜTS en son teknolojileri kullanmaktadır. Pentium tabanlı PC'lerde çalışan DÜTS yazılımı. Windows NT altında çalışan, ölçeklenebilir, 32-bit yüksek-performans *Client/Server* yapısına sahiptir. Programlar Delphi ortamında, nesneye yönelik Pascal ile geliştirilmekte olup InterBase veritabanım kullanmaktadır [7].

Kaynaklar

1. H. Sumen, "SCADA ve MRP II Sistemleri" *Otomasyon*. Sayı:54, Sayfa: 100-102. 1996
2. E. Saraç, "Üretim Hattı ve SCADA" *Otomasyon* Sayı:55, Sayfa:142-145. 1997.
3. A. Arakawa, M. Ono, (Editör M. Acar). "Development of a LAN System for Weaving Factories", *Mechatronic Design in Textile Engineering*, NATO ASI Series E Applied Sciences-Vol. 279, pp. 231-238 1995
4. S. Jayaraman, (Editör M. Acar), "Computer-Aided Design and Manufacturing A Textile-Apparel Perspective", *Mechatronic Design in Textile Engineering*, NATO ASI Series E Applied Sciences-Vol 279, pp 239-269. 1995
5. *The Barcode Book*. Roger C Palmer Helmers Publishing, İne, 1991.
6. A. Babaev, H. Gümüşkaya, Y. Uicay "Development of an Expert System for Diagnosis of Textile Fabric Defects". *6th Turkish Symposium on Artificial Intelligence and Netini! Nei\|-orks (TAINN'97)*. May 21-23, 1997, Ankara
7. *Delphi 2.0 C/S Suite User Guide* BD'and International, İne, 1996.

Mantıksal Ağ Analizi İçin Elektronik Çizelge Yazılımları

Hidayet Tunç ŞİMŞEK, Erol KOCAOĞLAN
Orta Doğu Teknik Üniversitesi
Elektrik ve Elektronik Mühendisliği Bölümü
06531 -ANKARA

e-posta: simsek@srdc.metu.edu.tr, erol-kocaoglan@metu.edu.tr

ABSTRACT

This paper presents a feasible method for analyzing logic networks with spreadsheets. Solutions are presented for combinatorial, synchronous and asynchronous sequential circuits, with the emphasis being on the latter. The methods presented here yield a very useful tool which may be used by students to develop an understanding of complicated digital design concepts. The primary advantage of our method is that timing considerations such as races in asynchronous sequential circuits are affiliated into the analysis. The validity of the methods presented in this paper are verified within the scope of experimental results. The methods presented in this paper do not require a pre-requisite in programming, they are cost effective and they can readily be implemented by undergraduate students.

I. GİRİŞ

Bu makale mantıksal ağların elektronik çizelgeler (spreadsheets) ile olurlu analizlerini konu almaktadır. Esas olarak katımsal, eşzamanlı ve eşzamansız ardışıl devrelerin çözümleri ve detaylı analizleri ele alınmaktadır. Önerilen yöntemler sonucunda lisans düzeyinde karmaşık sayısal kavramlarla baş edebilecek bir araç elde edilmektedir. Önerilmekte olan çözümler elde edilmiş deneysel sonuçlar vasıtası ile doğrulanmıştır.

Elektronik çizelgeler özellikle muhasebe uygulamalarında eşi olmayan algoritmik ve yineleyici hesaplama yazılımlarıdır. Temel unsurları girdi değişikliklerine anında tepki vermektir (*what-if simulation*). Bu sayede girdi-çıkı çözümlenmesi sorgulama vasıtası ile yapılabilmektedir. Sonuç olarak x girdisinin değeri a değil de b olsa z çıktısının değeri ne olurdu sorusuna cevap alabilmekteyiz. Buna ek olarak modern elektronik çizelgeler görsel yazılım modülleri ile gelmektedir. Bu modüller sayesinde olay güdümlü benzetimler yaratmak mümkündür. Bu iki temel özellik sayesinde mühendislik uygulamalarında çok sık karşılaştığımız algoritmik ve zaman alıcı yöntemleri ve hesapları özışler kılmak mümkündür.

Elektronik çizelgeler, sırası ile Huelsman, [HUE84], ve El-Hajj ve Kabalan, [HAK91], tarafından *IEEE Trans. On Education* dergisinde yayınlanan makalelerinde mantıksal ağ uygulamalarında yer almıştır. Ancak, önerdikleri yöntemler kullanıcı arayüzünü ve mantıksal devrelerin zaman dayanaklı sorunlarına değinmemiştir. Bundan dolayı, yöntemlerinde katımsal devrelerde "hazard", eşzamansız ardışıl devrelerde "race" ve gecikme-kusur bulgulama, [BDK92], gibi ileri düzey mantıksal dizge sorunlarına çözüm getirilmemiştir. Bu tür konular zor olmasına rağmen lisans düzeyindeki öğrencilerin mantıksal ağlara vakıf olmasında büyük katkılarda bulunur.

Bu makalede mantıksal ağlara benzetim kurmak için MicroSoft (MS) Excel ve görsel bir programlama dili olan Visual Basic ile yeni bir yöntem sunulmaktadır. Sunulmakta olan yöntemin avantajları şunlardır:

- Katımsal, eşzamanlı ve eşzamansız ardışıl devrelerin analizinde zamanlama ile ilgili sorunlar göz önünde bulundurulmaktadır.
- Grafik kullanıcı arayüzü sayesinde mantıksal ağların devre şemaları kolaylıkla girilebilmektedir.
- Hata ayıklama (debugging) ve doğrulama işlemleri adım-adım simülasyon sayesinde kolay bir şekilde gerçekleştirilebilir.
- Visual Basic yazılımlarını geliştirmek fazla programcılık tecrübesi gerektirmediğinden lisans düzeyindeki öğrenciler tarafından bu ve benzeri yazılımlar geliştirilebilir.

II. ELEKTRONİK ÇİZELGELER VE GÖRSEL PROGRAMLA İLE BOOLE İŞLEMLERİ

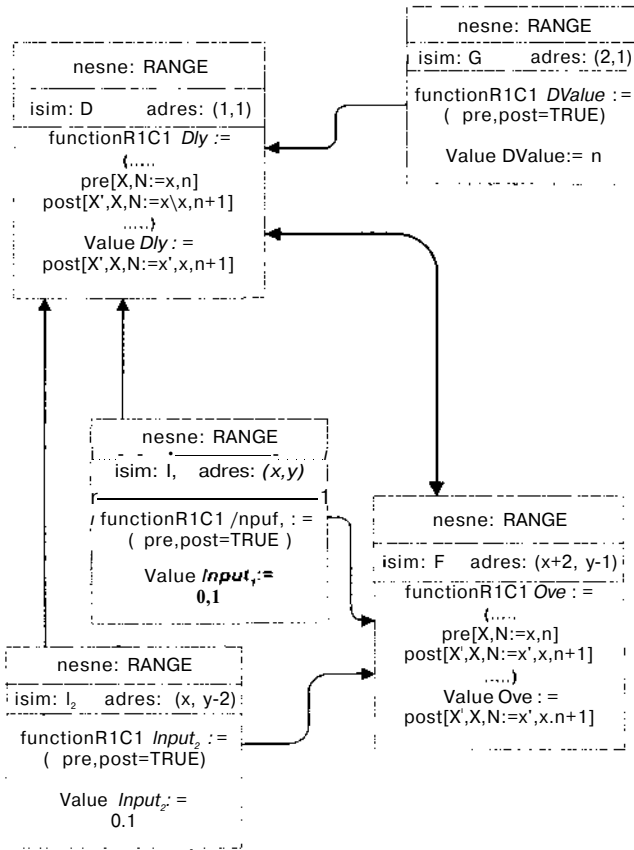
Görsel programlama son zamanlarda popülerlik kazanan basit bir programlama tekniğidir, [VB95], MS-Excel de bulunan Visual Basic programlama modülleri sayesinde görsel programlama nesnelere ve elektronik çizelgenin otomatik hesaplama özelliği ile mantıksal ağ devreleri simüle edilebilmektedir. Görsel programlama tekniğinde nesnelere yöntemlerinden (object methods) ve özelliklerinden (object properties) faydalanarak uygulamalar üretilir. Mantıksal devre çözümlenmesi için gerekli olan nesne aşağıda görülmekte olan "Range" nesnesidir ve bu nesnenin "FormulaR1d" özelliğidir.

Set $x/Object =$
 $Worksheets("sheet1").Range(cell1,cell2)$
 $x/Object.FormulaR1C1 = x/R1C1StyleFormula$

Tanım 1 $\mathcal{E}=\{1,0\}$ kümesi olsun. **Pre** serbest değişkenlerini $X\text{ç}1$, $N\text{ç}Z_+$ dan alan bir öneri (proposition) olsun. **Post** serbest değişkenlerini $X'\text{ç}1$, $X\text{ç}1$, $N\text{ç}Z_+$ dan alan bir öneri olsun. $F=(\text{Pre}, \text{Post})$ bir yordam olsun.

Eğer $VneN \text{ } -, (-, Pre[X'=x, N=n])VPosf | X'=x', X=x, N=n+1]$ önerisi doğru ise F yordamına otomatik hesaplama özelliği olan bir yordam denir.

"Range" nesnesinin "FormulaR1C1" özelliği otomatik hesaplama özelliği olan bir yordamdır. Şekil 1 de gösterilmekte olan "Range" nesnesinin etkileşim modeli, [DOD96], bir Boole işlevinin görsel modellemesidir. Burada "FormulaR1C1" özelliğini oluşturan *pre* ve *post* önerilen OVE (Olumsuz VE) Boole işlemi gerçekleştirmektedir. Elektronik çizelgeler esas olarak nesnelerin yöntemlerinin ve özelliklerinin birbirleri ile $VneZ_+$ anında otomatik olarak haberleşmesini sağlar. Haberleşme esansında oluşturulmuş olan *pre* ve *post* önerilerine gerekli olan serbest değişkenleri iletilir. Bu serbest değişkenler "Range" nesnesinin "Value" yöntemi ile elde edilmektedir. Bu şekilde *pre* ve *post* önerilerinin gerçekleştirdiği OVE işlemi otomatikleştirilir.



ŞEKİL1 Nesne etkileşim modeli

OVE işlemini pratikte gerçekleştiren Visual Basic yordamları Şekil 2 de gösterilmiştir. Bu şekildeki *Dly* fonksiyonu sanal saat işlemini görmektedir. *Dly* yordamının döndürdüğü değer *Ove* yordamının yerine getirdiği OVE işlemini düzenlemektedir. Burada dikkat edilmesi gerek nokta kullanılan "Range" nesnelerinin dizin adresleridir (x,y). Kullanılan "Range" nesnelerin Elektronik Çizelgenin üzerindeki görelî dizin adresleri Şekil 1 de verilmiştir. "Range" nesnesinin "Offset" yöntemi ile başka bir "Range" nesnesinin görelî adresi elde edilir.

```
Function Ove(Input1,Input2,
            DelayValue, IterationCount)
```

```
    If (IterationCount = DelayValue) Then
        Ove = 1
        If (input1 = 1) And (Input2 = 1) Then
            Ove = 0
        EndIf
    Else
        Ove = Input1.Offset(-1, 2)
    EndIf
```

```
End Function
```

```
Function D/y(Input1, Input2, DelayValue)
```

```
    If (Input1 = 1) And (Input2 = 1) Then
        If (Input1.Offset(-1, 2) = 1)Then
            Ove = DelayValue. Offset(1,C» + 1
        Else
            Ove = 0
        End If
    Else
        If (Input1.Offset(-1, 2) = 0) Then
            Ove = DelayValue. Offset(LO) + 1
        Else
            Ove = 0
        EndIf
    EndIf
```

```
End Function
```

ŞEKİL2 Visual Basic *Ove* ve *Dly* Yordamları

III. OLE İLE KULLANICI ARAYÜZÜ

OLE (Objects Linking and Embedding) sayesinde kullanıcı arayüzünde simgeler kolaylıkla yaratılır. Bunun için MS-Excel'in çizim nesnelere - daire, kare, üçgen vb. - kullanılır. Öncelikle devre elemanlarını temsil edecek simgeler tasarlanır ve bu tasarım Şekil 3 de gösterildiği gibi otomatikleştirilir. Burada *OveKapsısınıOluştur* yordamı yarım daire ve düz çizgiler kullanarak ekrana bir OVE simgesi çizer. Sonra, bu simgenin kapsadığı "Range" nesnelerini gerekli *Ove* ve *Dly* yordamları ile bağdaştırır

Tanım 2 Arcs, Lines ve Circles "Range" nesnesinin çizim özellikleri olsun ve CE{Arcs,Lines,Circles} olsun, 11,12 ve G "Range" nesnesi olsun. F,D otomatik hesaplama özelliği olan birer yordam olsun, $S=(I_1^s, I_2^s, F_s, D_s, G_s, C_s^k)$ bir Boole simgesidir, 11, 12 simgenin girdilerini temsil eder. G simgenin fiziksel gecikmesini temsil eder. F simgenin Boole işlevini yerine getiren bir yordamdır. D simgenin gecikmesini yerine getiren bir yordamdır.

Sub *OveKapısımOluştur* ()

GateNo = GateNo + 1

With ActiveWindow.ActiveCell

coor1 = Left
coor2 = Top
coor3 = Left + .Width
coor4 = Top + (3/2) * .Height
coor5 = Top + 3 * .Height
ovaltop = 0.2 * .Width
ovalheight = 0.4 * .Height

formula_F = "Ove(xIStyleAddress)"
formula_D = "DIy(IStyleAddress)"

ActiveCell.FormulaR1C1 = formula_F
ActiveSheet.Cells(2, GateNo).FormulaR1C1
= formula_D

End With

With ActiveSheet

.Arcs.Add(coor1,coor2,coor3,coor4)
.Arcs.Add(coor1,coor5,coor3,coor4)
.Lines.Add(coor1,coor2,coor1,coor5)
.Ovals.Add(coor3,ovaltop,ovalwidth,ovalheight)

End With

End Sub

ŞEKİL: 3 Visual Basic *OveKapısımOluştur* Yordamı

IV. MANTIKSAL AĞ OLUŞUMU

Mantıksal ağ analizi Boole işlevi ve devre simgelerinin devre yapısına uygun olarak etkileşimiyle yapılır. Burada esas olan herhangi bir simgenin çıktısını diğer simgenin girdisine uygulanmasıdır. Bunu gerçekleştirmek için MS-Excel *kes/kopyala/yapıştır* özelliği kullanılır. Bir S simgesinin çıktısını temsil eden "FormularICr nesne özelliği kopyalanır ve diğer simgenin girdisini temsil eden "Range" nesnesine yapıştırılır Bunun da ötesinde, bu sanal bağlantı MS-Excel'in çizim nesneleriyle temsil edilebilir.

Tanım 3 S bir Boole simgesi olsun. $M = \{ S \mid s, teM$ ise ve s nin çıkışı t yi etkiliyor ise $I_t := F_s \}$ bir mantıksal ağdır.

V. ÖRNEK: Eşzamansız Ardışık Mantık Devresi

Şekil 4 de "race" durumu içeren eşzamansız ardışık bir mantık devresi görülmektedir, [MAN91]. Devre $X1^2Y3Y6 = 1^0 \cdot 1$ toplam kararlı durumunda iken X2 girdisi 0 -> 1 değişimi gösterirse devrenin bir sonraki durumunda kararsızlık gözlenir:

- 3. Kapının gecikmesi fazla ise: $x-\backslash x2Y3Y6 = 1101$
- 6. Kapının gecikmesi fazla ise: $x-\backslash x2Y3yQ = 1110$

Devrenin değişim tablosu her iki durum için tablo 1(a) ve 1(b) de gösterilmiştir. Bu örnekte 6. Kapı (y_6) her zaman mantıksal 0 olarak sonuçlanmaktadır. Tablo 1(a) y_6 daki 1^0 geçişinin uzun bir gecikmeli olduğu durumun değişim tablosudur. Bu gecikmeden dolayı y_3 0^1 geçişini yapabilmektedir. Tablo 1(b) aksi durumun değişim tablosudur. Burada y_6 daki 1^0 anlık geçişi y_3 ün mantıksal 0 değerinde kalmasına sebep olmaktadır.

TABLO: 1 Şekil 4 teki devrenin değişim tablosu
(a) 3. Kapının gecikmesi fazla
(b) 6. Kapının gecikmesi fazla

Dürüm	*1*2	y1 -y6	11	11	13	14	75	76
0	10	110101	0/1	0/1	0/1	0/1	0/1	0/3
1	11	100111	0/1	1/1	0/1	0/1	1/1	0/3
2	11	101111	0/1	0/1	1/1	0/1	0/1	1/3
3	11	001111	1/1	0/1	0/1	0/1	0/1	2/3
4	11	001110	0/1	0/1	0/1	0/1	0/1	3/3
5	11	011110	0/1	1/1	0/1	0/1	0/1	0/3
6	11	011110	0/1	0/1	0/1	0/1	0/1	0/3

(a)

Dürüm	*7*2	y7 -y6	11	11	73	14	75	76
0	10	110101	0/1	0/1	0/3	0/1	0/1	0/1
1	11	100111	0/1	1/1	0/3	0/1	1/1	0/1
2	11	100110	0/1	0/1	1/3	0/1	0/1	1/1
3	11	110110	0/1	11	2/3	0/1	0/1	0/1
4	11	110110	0/1	0/1	0/3	0/1	0/1	0/1

(b)

VI. SONUÇ

Bu makalede önerilen yöntemler çeşitli problemler üzerinde denenmiş ve doğrulanmıştır. Buna ek olarak, bahsi geçen yöntemler hiç bir programlama alt yapısı gerektirmez. Bu sebepten dolayı lisans düzeyindeki öğrenciler tarafından bu ve benzeri tip yazılımlar geliştirmek mümkündür.

Microsoft Excel - Örnek

File Edit View Insert Format Tools Data Window Help

gaitn_i: =my'nand'gap^PQr's.p7r:-P|r/.F:3:3)

1 1 1 1 1 3 <-Gecikme Değerleri
0 0 0 0 0 0 <-Dürüm

T1 T? T3 T4 T5 T6

Comments:

In this example, there is a critical race from the total stable state = 1. 2v3v6=1001 when x2 is changed from 0 to 1.

The possible results are: 1. 2v3v6 = 1101 if nand2 has a larger delay
2. 1110 if nand4 has a larger delay

State Transition Table:

y3y6	x1x2=00	01	11	10
00	MM	MM	MM	MM
01	MM	MM	MM	MM
11	MM	MM	MM	MM
10	MM	MM	MM	MM

Ready Calculate Sum=1 NUM

ŞEKİL. 4 Eşzamansız Ardışık Mantık Devresi

KAYNAKÇA

[HUE84] L. P. Huelsman. "Electrical engineering applications of microcomputer spreadsheet analysis programs," *IEEE Trans. Educ.* vol. E-27, pp. 86-92. 1984.

[HAK91] A. El-Hajj, K. Y. Kabalan, "A spreadsheet simulation of logic networks," *IEEE Trans. Educ.*, vol. 34, pp. 43-46, 1991.

[BDK92] M. J. Bryan, S. Devadas, K. Keutzer. "Necessary and sufficient conditions for hazard free robust transistor stuck-open fault testability in

multilevel circuits," *IEEE Trans. Computer-Aided Design.* vol. 11, pp. 800-803, June 1992.

[VB95] *Microsoft Excel / Visual Basic Programmer's Guide.* Microsoft Press, 1995.

[DOD96] M. Dodani, "Object-oriented methodologies in practice: The Big Picture," *Journal of Object-Oriented Programming*, vol. 9, pp. 26-29, April 1996.

[MAN91] M. Morris Mano. *Digital Design*, 2nd ed., Prentice-Hall, 1991

AYNI-TÜRLÜ BOOLE FONKSİYONLARIN TANINMASI

Vasif NABIYEV , Rifat YAZICI.

Karadeniz Teknik Üniversitesi,
Bilgisayar Müh. Bölümü,
61080-TRABZON

ABSTRACT

This paper deals with an implementation of Boolean functions of the same types with a common logic Circuit. The BF's are classified into groups according to their structures so that each group shares the same switching logic. Identification of BF types is an important research subject and with wide practical application.

To identify a BF type, the meth vector of the BF is used as a pointer to the table containing some important knowledge about its functions.

The number of all the 4-variable functions, which is equal to $(2^4)^{2^4} = 65536$, can be reduced to 402 different types of functions. This reduction shows that it is necessary to build up an effective algorithm for identification of the BF types. In this study, the meth properties of BF's of the same type are investigated to efficiently and easily determine the BF types.

1. GİRİŞ

Teknik mantıkta Boole Fonksiyonlarının (BF) benzerliğinin belirlenmesi büyük önem taşımaktadır. BF kümesini aşağıda verilen üç açıdan sınıflandırmak mümkündür.

1. Değişkenlerin yerine terslerinin koyulması
2. Değişkenlerin yer-değiştirilmesi
3. Fonksiyonların yerine terslerinin koyulması

Yukarıdaki ölçütü gözönüne alarak yapılacak çevirmelere aynı türe çevirme denir. Aynı mantık devresini gerektiren Boole fonksiyonları kümesindeki her fonksiyon aynı özellikleri taşımaktadır. Bu yüzden, bu ortak özelliklerin belirlenerek aynı türlü fonksiyonların verilen bir fonksiyonlar kümesinden seçilmesi , çok-fonksiyonlu bir mantık sisteminin

donanımsal gerçekleştirmesini daha basit ve ucuz yapacaktır.

4-değişkenli BF türleri Akers S.B. tarafından belirlenmiş ve bunlara içerdikleri terim sayısına göre Harvard numaraları verilmiştir. Ama verilen bir fonksiyonun bu türlerden hangisine karşı düştüğünü gösterecek bir yöntem henüz bulunamamıştır.

Bu çalışmada 4-değişkenli BF türleri belirlenmeye çalışılmıştır. Bu amaçla türlerin metrik özelliklerini gösteren karakteristik vektörler oluşturularak yeni bir sınıflandırma algoritması geliştirilmiştir. Bu algoritma genel olmasına rağmen, bilgisayarda işlem kolaylığı sağlasın diye değişken sayısı dört olarak sınırlandırılmıştır.

2. AYNI-TÜRLÜ BOOLE FONKSİYONLARIN METRİK KARAKTERİSTİK VEKTÖRÜ $\forall \mu$

Konunun daha kolay izlenebilmesi için aşağıdaki kavramların verilmesi yararlı olacaktır.

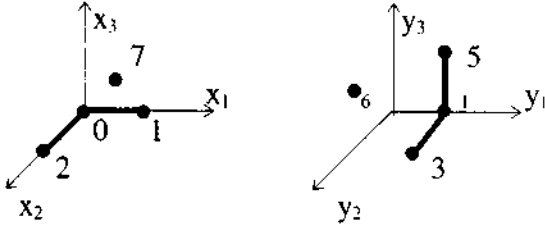
Tanım. $f(x_1, x_2, \dots, x_n)$ fonksiyonundaki değişkenlerin yeri değiştirildiğinde ve bazı değişkenlerin tersi alındığında elde edilen $g(x_1, x_2, \dots, x_n)$ fonksiyonuna, $f(x_1, x_2, \dots, x_n)$ fonksiyonu ile aynı-türlü olan BF'nu denir.

Örneğin $f(x_1, x_2, x_3)$ ile $g(y_1, y_2, y_3)$ Boole fonksiyonlarının aynı-türlü olabilmesi için karşılıklı

olarak $y_1 \Leftrightarrow x_1, y_2 \Leftrightarrow x_2, y_3 \Leftrightarrow x_3$ dönüşümlerinin yapılabilmesi gereklidir (şekil 1).

$$f(x_1, x_2, x_3) = \overline{x_1 x_2 x_3} \vee x_1 x_2 x_3 \vee x_1 x_2 \overline{x_3} \vee x_1 \overline{x_2} x_3$$

$$g(y_1, y_2, y_3) = y_1 y_2 y_3 \vee y_1 y_2 \overline{y_3} \vee y_1 \overline{y_2} y_3 \vee y_1 \overline{y_2} \overline{y_3}$$



ŞEKİL 1. $f(x_1, x_2, x_3)$ ve $g(y_1, y_2, y_3)$ Boole fonksiyonlarının uzayda gösterilişi

$f(x_1, x_2, \dots, x_n)$ fonksiyonunun terimleri sayısına BF'nun rankı denir. Aynı-türe dönüştürmede birinin terimleri diğerinin terimlerine dönüştüğünden dolayı aynı-türlü fonksiyonların rankları eşittir. Rankların eşitliği fonksiyonların aynı türlülüğü için gerek koşul olduğundan, rankların eşit olmaması BF'nun farklı olması için yeter koşuldur.

Tanım 2. n-boyutlu uzayın N ve M noktaları arasındaki Hamming uzaklığı, M noktasını N noktasına götürmek için gereken farklı değerli değişkenlerin sayısına denir.

n-boyutlu Boole uzayının noktaları arasındaki maksimum uzaklık n'ye, minimum uzaklık ise 0'a eşittir. Aralarındaki uzaklık 1'e eşit olan noktalar (terimler) komşu noktalar olarak adlandırılır. Verilen a noktasından uzaklığı $p(a, /3) \leq k$ koşulunu gerçekleyen (3 noktaları kümesine a noktasının k civarı denir. Görüldüğü gibi a ve (3 noktalarının /< civarlarının arakesitinin boş olması için gerek ve yeter koşul

$p(a, P) \leq k$ olmasıdır. Fonksiyonları Boole uzayında varsayarak metrik vektör kavramı verilebilir.

Tanım 3. n-değişkenli fonksiyonun Boole uzayında bulunan noktaları arasındaki eşit Hamming uzaklıkları sayılarının oluşturduğu vektöre metrik vektör (V_n) denir.

Teorem 1. Her bir tür, n-boyutlu uzayda n-boyutlu metrik vektörle karakterize edilir.

Teorem 2. V_n metrik vektöründeki j_i eleman değerlerinin toplamı sabit olup

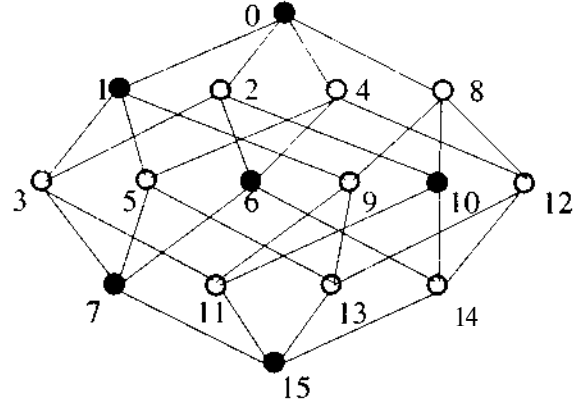
$$\sum_{j=1}^n f_i(k_j, k_s) = R(f)(R(f)-1)/2$$

bağıntısı ile verilebilir. Burada f_i vektörün i elemanının değerini ve $R(/)$ ise Boole türünün rankını gösterir.

Bu metrik vektöre dayanan tanıma yöntemi, $f(x_1, x_2, \dots, x_n) = \bigwedge_{i=1}^n \mu_i$ değerli BF'nun n-boyutlu küpü.)

köşeleri arasındaki uzaklıklarının dağılımına göre yapılabilir (şekil 2). Örnek için Harvard numarası 53 olan $f_{53}(x_1, x_2, x_3, x_4) = \bigwedge_{i=1}^4 V(0, 1, 6, 10, 15)$ Boole türünü gözönüne alalım. Bu tür $V(f)$ Boole altuzayının yardımı ile aşağıdaki şekilde belirlenebilir.

$$T(f) = (\{0, 1, 6, 10, 15\}, \{0, 1\}, \{6, 10, 15\})$$



ŞEKİL 2. $T(f_{53})$ altuzayının noktalarının hiperküpte gösterilişi

Burada Hamming uzaklıkları :

$$H(0, 1) = 1,$$

$$H(0, 6) = H(0, 10) = H(6, 10) = H(6, 15) = H(10, 15) = 2,$$

$$H(1, 6) = H(1, 10) = H(1, 15) = 3,$$

$$H(0, 15) = 4$$

değerlerini alır.

TABLO 1. $T(f_{53})$ altuzayının Hamming uzaklıkları

10	2			
6	2	2		
1	3	3	3	
0	4	2	2	1
	15	10	6	1

j_i uzaklıklara ($i = 1, w$) göre sıralandığında $V(f)$ metrik karakteristik vektörü aşağıdaki biçimde elde edilir.

$$V_n = (1, 5, 3, 1)$$

Fonksiyonun rankının $R=5$ olduğu gözönüne alındığında teorem 2'den metrik vektörün eleman değerlerinin toplamı.

$$\sum_{i=1}^4 \mu_i = 1 + 5 + 3 + 1 = 10$$

veya

$$R(O(R(O-1)/2)=(5*4)/2$$

olarak bulunur.

Hamming uzaklıkları toplamının sabitliğinden, bilgisayar uygulamasında türlerin tanınmasını kolaylaştıran aşağıdaki önemli sonuç elde edilir.

Sonuç 1. n-boyutlu uzayda bir tür bu türün karakteristik vektöründen herhangi bir elemanın atılmasıyla elde edilen (n-1)-boyutlu vektörle tanınabilir.

/ uzaklıklarının Q/ maksimum değerlerinin hesaplanması için aşağıdaki bağıntı yazılabilir.

$$Q_i = \binom{n}{i} * 2^{n-1}, \quad i = \overline{1, \dots, n}$$

Burada / , metrik vektördeki hanelerin sırasını ve n ise Boole uzayının boyutunu gösterir.

Özel olarak, n=4 alındığında karakteristik metrik vektör aşağıdaki sınır değerlerini alır.

V	32	48	32	8
i	1	2	3	4

Bu yorumlardan maksimum uzunlukların toplam sayısının aşağıdaki şekilde hesaplanabileceği sonucuna varılır.

$$S_d \cdot \prod_{i=1}^n 2^i = \binom{2^n}{2}$$

BF'un karakteristik metrik vektörüne bakılarak, türlere karşı düşen BF kümesini önemli derecede daraltmak mümkündür.

2. AYNI-TÜRLÜ BF'ların SEVİYE-İLİŞKİ KARAKTERİSTİK VEKTÖRÜ V_m

Özel durumlarda seviye-ilişki vektörünü kullanmak daha elverişlidir. Eğer bütün BF ve değişkenleri için bir koordinat sistemi seçilirse, aynı-türlü Boole fonksiyonları farklı durumları olan aynı biçimli geometrik şekillerin benzerleri olacaktır. O zaman aynı türlü fonksiyonlara bir mantıksal biçimin farklı koordinat sistemlerindeki yazılımı gibi bakılabilir. Boole türü ise bu biçimin sabit bir uyarlaması olarak görülebilir. Buradan, aynı BF'un hiperküpteki gösteriminde köşelerin birim civarındaki ilişkiler sayısının eşit olduğu anlaşılır.

Tanım 4. Seviye-ilişki karakteristiği (n+1)-boyutlu bir vektör olup , bu vektörün her / elemanı hiperküptün köşelerinin birim civarı için S(V) ilişki derecesini belirtir.

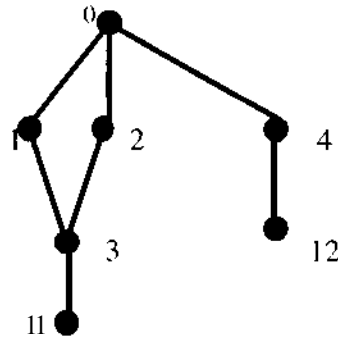
Teorem 3. Seviye-ilişki vektörünün elemanlarının toplamı sabit olup fonksiyonun rankına. her elemanın maksimum değeri ise değişkenler sayısına eşittir.

$$\sum_{i=1}^n S_i = R, \quad S_{\max(i)} = n$$

Örnek olarak 121 Harvard numaralı Boole türü gözönüne alınsın ve $F(f_{121})$ Boole altuzayı yardımı ile belirlendiği varsayılınsın.

$$f_{121}(X_1, X_2, X_3, X_4) \equiv 1 = V(0, 1, 2, 3, 4, 11, 12)$$

$$n=4, \quad R=7$$



ŞEKİL 3. $f_{121}(X_1, X_2, X_3, X_4)$ fonksiyonunun ilişki ağacı

$r(f-|2|)$ altuzayının V_i birim köşelerinin $S(V_i)$ ilişki dağılımından (n+1)-boyutlu V_m seviye-ilişki vektörü elde edilir.

$$T(f) = (\{0, 1, 2, 3, 4, 11, 12\}, \{0, 1\}, \{0, 2\}, \{0, 4\}, \{1, 3\}, \{2, 3\}, \{4, 12\}, \{3, 11\})$$

$$\begin{aligned} S(0) &= S(3) = 3, \\ S(1) &= S(2) = S(4) = 2, \\ S(11) &= S(12) = 1. \end{aligned}$$

Buradan tanım 3'e dayanarak seviye-ilişki

vektörü V_m aşağıdaki gibi verilebilir.

$$V_m = 0, 2, 3, 2, 0$$

Sonuç 2. Elemanların toplamının sabitliğinden dolayı, n-boyutlu uzayda seviye-ilişki

karakteristik vektörü n -boyutlu bir vektörle tanımlanabilir.

Böylece, Boole fonksiyonları, Boole türlerin karakteristik vektörlerine göre sınıflandırılarak Harvard sıralamasından farklı olan yeni bir Türkiye - Azerbaycan sıralaması elde edilmiş oldu.

4. SONUÇLAR

Bu çalışmada Boole fonksiyonlarının tanınması amacı ile onların metrik özellikleri araştırılmıştır. Bu özellikler

- Boole uzayında iki nokta arasındaki Hamming uzaklığına ve
- Boole uzaydaki noktaların ilişki derecesine

dayandırılmıştır.

Sonuç olarak n -boyutlu metrik ve $(n+1)$ -boyutlu seviye-ilişki vektörleri elde edilmiş, ve bu vektörlerin özellikleri araştırılarak 4-değişkenli Boole türlerinin terimler sayısına dayalı Harvard sıralanmasından farklı olan ve her türe erişimi sağlayan daha etkin bir algoritmaya dayanan Türkiye-Azerbaycan sıralaması geliştirilmiştir.

KAYNAKLAR

1. Muroga S. *Logical Design and Switching Theory*. Toronto, 1989, 624p.
2. Davido J.-P., Dechamp M., Thouse A. *Discrete and Switching Function*. New York, Mc Graw - HM, 1977.
3. Gorbatov V.A. *Switching Schemes and Graphs (Rus)*, Moscow, 1993
4. Nabiyev V.V. *Design of Logic network in Bases of Structure Algebraic (Rus)*, International Symposium 'Logic Management', Moscow, 1991.

Görüntü İşleme İçin Hızlı Bulanık Kohonen Topaklama Ağı Algoritması

Mehmet BULUT *, Derya DEMİR**, Hamdi ATMACA

* Dumlupınar Üniversitesi, Elektrik-Elektronik Mühendisliği, KÜTAHYA
Tel : +90-274-2242074; e-mail: bulut@ana.cc.yildiz.edu.tr

** Tübitak-MAM Bilişim Teknolojileri Enstitüsü., PK. 21, 4170 Gebze-Kocaeli
Tel: +90-262-6412300; fax : +90-262-6412309, e-mail: derya@yunus.mam.tubitak.gov.tr

ABSTRACT

In this study, Fuzzy Kohonen Clustering Networks (FKCN) algorithm is proposed for segmentation at image processing. But this algorithm for iterative operation will be very time consuming. Therefore, attempts have been taken in order to make the FKCN algorithm realistically useful for image segmentation. In this work, a new FKCN algorithm developed based on image histogram. Theoretical analysis and experimental results show that computing time decreased and proposed FKCN was successful algorithm for image segmentation.

Keywords : Kohonen Clustering Network, Fuzzy C-means, Image segmentation

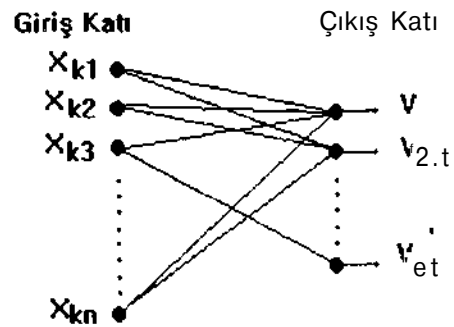
1. Giriş

Görüntü bölütlemenin amacı, görüntüyü özel bir uygulama alanına göre anlamlı bölgelere ayırmaktır. Bu anlamlı bölgeler birbirleri ile örtüşmemeli ve kendi içlerinde belirleyici bir özelliğe göre süreklilik göstermelidir. Bölütleme sonucu ortaya çıkan bu bölgeler bölüt (segment) adını alır. Topaklama (clustering) analizi görüntü işlemenin bölütleme işlemi ile aynı özellikler taşır. Bundan dolayı topaklama analizinde kullanılan yöntemler görüntü bölütlemeye de kullanılabilir. Görüntü bölütlemeye belirleyici özellik olarak grilik seviyesi, doku özelliği, renk bilgisi, çizgилilik ve süreklilik alınabilir. Bu çalışmada grilik seviyesi kullanılmıştır. Görüntü bölütleme işlemi topak analizi olarak ele alınabilir. Bu durumda bir görüntüdeki her bir nesne ve arkaplan birer topak gösterirler. Topaklama algoritmaları görüntülere uygulanarak bölütlenmeleri sağlanabilir. Topaklama algoritmaları, topaklara örüntü düzenleyerek örüntüler arasında ki etkileşmeyi belirlemeye çalışır [1]. öyleki bir topaktaki örüntüler, farklı topaklara ait olan örüntülerden daha fazla

birbirine benzemektedirler. Bu promleme birçok klasik yaklaşım Kohonen,Bezdek, Duda ve Hart, Tou ve Gonzales tarafından ortaya atılmışlardır[^],3,4].

2. Kohonen topaklama Ağı (KCN)

Kohonen topaklama ağları, iteratif olarak kesin topaklama (hard clustering) için en iyi ağırlıklar kümesini bulan öğretmensiz (unsupervised) hücrelerdir. KCN yapı olarak Şekil 1.'de gösterildiği gibi bir giriş veya besleme (fanout) katı ve bir çıkış veya yarış (competitive) katı olmak üzere iki kat içermektedir. Her çıkış düğümü bir protipe veya ona iliştilirilen bir ağırlık vektörüne sahiptir. Bu ağırlık vektörü öğrenme esnasında ayarlanır. Verilen bir giriş vektörü için çıkış katındaki düğümler arasında bir yarış olur ve girişten minimum uzaklığa sahip olan ve kazanan (winner) olarak adlandırılan ağırlık vektörü güncelleştirilir. Bunun için de önceden tanımlanan komşuluk kümesi ve ağırlıkları ayarlanır. Bu işlem ağırlık vektörü kararlı olana kadar devam eder. Bu yöntemde bitiş işlemi için zamanla azalan bir öğrenme hızı tanımlanmalıdır. Ayrıca yine zamanla azalan güncel komşuluk tanımlanmalıdır [2],



Şekil. 1 Kohonen Topaklama Ağı Yapısı

Kohonen Topaklama Ağlarının (KCN), bazı problemleri vardır. Ağda "bitiş" problemi oluşmaktadır. KCN'ler heuristik işlemler olduklarından dolayı 'bitiş' (termination), verilerin veya herhangi bir işlem modelinin optimizasyonu üzerinde temelli değildir. Son ağırlık vektörleri genellikle giriş dizisine bağlıdır; farklı başlangıç şartlarında genellikle farklı sonuçlar üretilir. KCN algoritmalarında öğrenme hızı (learning rate), komşuluk güncelleştirme büyüklüğü (size of reighborhood) öğrenme esnasında değiştirilerek bir ver: kümesinden diğerine değişen 'kullanılabilir' sonuçların üretilmesi sağlanır.

$$U_{ik,t} = \frac{z_{ik}^2}{z_{ik}^2 + \sum_{j=1}^c z_{ij}^2} \quad (1)$$

Denk. (1) için ana Uya ihtiyaç duyulmamakta ve komşuluk güncelleştirilmesi KCN'deki gibi yapılmaktadır.

Bezdek ve Pal'in bu çalışmasında ise KCN ve FCM algoritmalarının tam birleştirilmesi, Kohonen güncelleştirilmesi için öğrenme hızı tanımlanması ile gösterilmektedir.

$$a_{ik,t} = \left(U_{ik,t} \right)^{m_t} \quad m_t = \frac{(m_0 - 1)}{t \max} \quad (2)$$

Burada $U_{ik,t}$ yukardaki gibi hesaplanmakta ve $m_t = m_0 - (m_0 - 1) \frac{t}{t_{max}}$, m_0 birden büyük pozitif bir sabitlerdir ve $m_t > 1$ 'dir. Burada t_{max} KCN için iterasyon sınırır

3. Bulanık Kohonen Topaklama Ağı (FKCN)

KCN topaklama, Bulanık C-ortalama (Fuzzy C-means) algoritması ile yakından ilgilidir. Bulanık C-ortalama (Fuzzy C-means) algoritması görüntü bölütleme için de kullanılabilir. FCM, görüntüdeki pikseller ile her bir C- topağının merkezi arasında bir ağırlıklı benzerlik ölçüsünü hesaplar. Bu benzerlik ölçütüne dayanan bir fonksiyonun iteratif olarak optimuma ulaşmasını kullanır. Bu fonksiyonun yerel bir extramumu görüntünün optimal bir topağını gösterir[6]. FCM algoritması optimize işlemler olduğundan bunların KCN ile birleştirilmesi bazı problemlerin ortadan kaldırılmasını sağlar. Hunsterberg ve Ajjimarangsee[7] bu yaklaşımı ilk olarak ele aldılar. Bezdek ve Pal'in bu çalışmasında [5], Bulanık Kohonen Topaklama Ağı (FKCN) algoritması olarak adlandırılacak yeni bir algoritma sunulmaktadır. öğrenme hızı (laerning rate) parametresi, FCM ile KCN kurallarının güncelleştirilmesi için bulanık üyelik fonksiyonuna dönüştürülmektedir. FKCN kendinden organizeli olması ve komşuluk güncelleştirme büyüklüğü, öğrenme esnasında otomatik olarak ayarlandığından FKCN algoritması genellikle FCM nesnel fonksiyonunun minimizasyon yaklaşımı ile bitmektedir. Bu yeni algoritma FCM algoritmasının Kohonen Topaklama algoritmasına uyarlanmış bir halini göstermektedir.

Yapılan çalışmalar, KCN ve FCM ile üretilen nümerik sonuçlar arasında yakın bir ilişki olduğunu göstermektedir. Hunsterberg ve Ajjimarangsee KCN'de şu değişikliği yapmışlardır.

$$V_{k,t} = V_{k,t-1} + u_{ik,t} (x_k - V_{k,t-1}) \quad (1)$$

Burada $U_{ik,t}$, üyelik derecesini göstermekte ve şu şekilde hesaplanmaktadır.

Bulanık Kohonen Topaklama Algoritması

Adım 1. C, || *jj ve t: > 0 olan parametreler seçilir

Adım 2. Başlangıç $V_0 = (V_{10}, V_{20}, \dots, V_{c0})$ değerleri, $m_0 > 0$ ve t_{max} iterasyon limiti seçilir,

Adım 3. $t=1,2,\dots,t_{max}$ için

(a) Bütün öğrenme hızları $\{u_{ik,t}\}$ hesaplanır.

$$\alpha_{ik,t} = \left(U_{ik,t} \right)^{m_t} \quad m_t = \frac{(m_0 - 1)}{t \max}$$

(b) Bütün ağırlık vektörleri $\{V_{k,t}\}$ güncelleştirilir

$$V_{k,t} = V_{k,t-1} + \frac{\sum_{i=1}^n \alpha_{ik,t} (x_i - V_{k,t-1})}{\sum_{i=1}^n \alpha_{ik,t}}$$

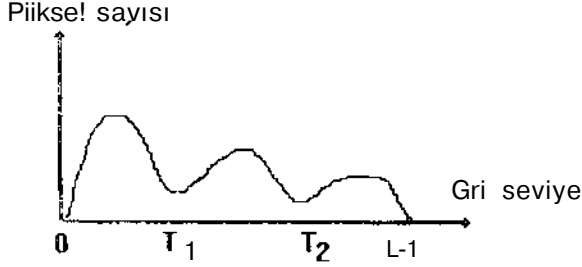
(c) $E_t = \sum_{k=1}^c \|V_{k,t} - V_{k,t-1}\|$ hesaplanır,

(d) Eğer $E_t < e$ dur, değilse $t = t + 1$ yap ve 3.adımın başına git.

4. Görüntü Histogramı

öte yandan ele alınan görüntüde arkaplan üzerinde birden fazla nesne olabilir Bu nesnelerin özelliklerine bağlı olarak yansıttıkları grilik düzeyleri farklı olabilir. Bu durumda görüntü histogramı ikiden fazla baskın tepeye sahip olabilecektir Örneğin Şekil

7 de arkaplan üzerinde iki nesne içeren bir görüntü histogramı görülmektedir. Buna bu görüntünün 3 topaktan (bölüt) oluştuğu açıkça görünmektedir. Bölütleme algoritmaları kullanılarak bu görüntüdeki arkaplan ve nesnelere birbirinden ayrıştırılabilir. Yani herbir piksel bulanık kümeler kullanılarak hangi bölüte daha yakın olduğu saptanabilir ve her bölüt homojen bir yapıya kavuşturulabilir.



Şekil 2. Arkaplanda iki nesneye sahip bir görüntünün histogramı

5. Hızlı FKCN Algoritması Geliştirilmesi ve Görüntü Ölütlemesinde Kullanılması

Görüntü boyutu (büyüklüğü) birçok uygulamalarda yeterince büyük olmaktadır. Aynı zamanda FKCN algoritmasının hesaplanması iteratif işlem gerektirdiğinden böyle büyük veri parçaları için hesaplama çok zaman almaktadır. Bundan dolayı FKCN algoritmasını görüntü bölütlemeye kullanışlı hale getirmek için işlemdeki n veri parçasının sınırlandırılması gerekmektedir[8]. Burada amaçlanan algoritma, işlem sayısını azaltmak için n veri parçası olarak görüntü uzayını yerine gri seviyeleri gösteren gri fonksiyonu kullanmaktadır. Algoritma gri seviyelerini gösteren histograma dayanmaktadır.

Gri seviyesi olarak negatif olmayan $G=\{0,1,2, \dots, L-1\}$ tanımlansın. Burada L-1 maksimum gri seviyesini göstermektedir. $M_1 \times N_1$ büyüklüğündeki bir görüntü için (i,j) noktasındaki gri seviye $f(i,j)$ ile tanımlanmaktadır, $1 < i < M_1, 1 < j < N_1$. Görüntüdeki k gri seviyesine sahip piksel sayısı $h(k)$ ile gösterilmektedir, $k=1,2, \dots, L-1$. $H(k)$, histogram olarak adlandırılmaktadır. Burada da nesnel fonksiyon kullanılmaktadır. Fakat görüntü yüzeyinde topak merkezi bulmak yerine, gri seviyesinde topak merkezi bulunmaktadır. Bunun için bir boyutlu koordinatta aşağıdaki ifade geliştirilmiştir.

$$V_{i,t} = V_{i,t-1} + \frac{\sum_{k=1}^L \alpha_{i,t}(k-V_{i,t-1})h(k)}{\sum_{s=1}^L \alpha_{i,t}(s)} \quad (3)$$

Hesaplama yöntemi FKCN algoritmasına benzerdir. Burada görülmektedir ki veri sayısı doğrudan $M_1 \times N_1$ 'den L-1'e azaltılmaktadır. Mesela 512×512 büyüklüğünde bir görüntü ve 8 bit gri seviye (veya $L=256$) için hesaplama zamanı $512 \times 512 / 256$ kez azalacaktır.

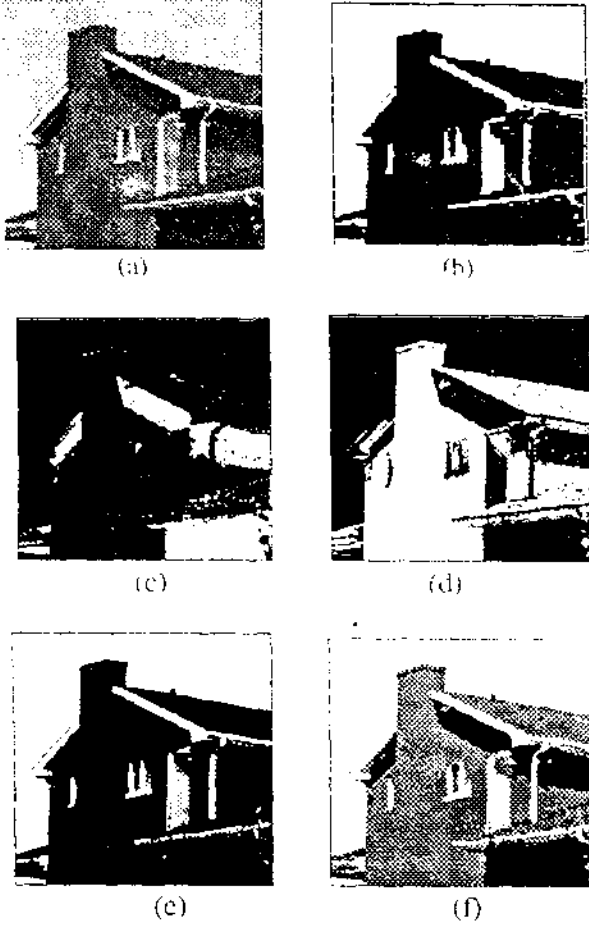
6. Sonuçlar

Bu çalışmada görüntü histogramı bilgisi kullanılarak FKCN algoritmasına görüntü bölütleme özelliği sağlanmıştır. Amaçlanan algoritma kısa bir zaman aralığında görüntüde yer alan herbir pikselin topaktaki üyelik değerlerini hesaplamakta ve görüntü bölütlenmektedir. Dahası geliştirilen ağ ikiden fazla bölüt ile de çalışabilmektedir. Algoritmadaki O sabiti topak sayısına karşılık gelmektedir.

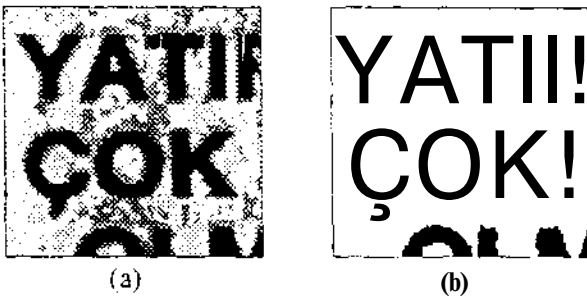
Geliştirilen algoritma çeşitli görüntülere uygulanmış ve başarılı sonuçlar alınmıştır. Algoritma House, Gazete, Lenna ve Kameraman görüntülerine uygulanmış ve sonuçlar Şekil 3-6'de verilmektedir. Sonuçlar $m=2$ ve $d=0.02$ değerleri için alınmıştır.

Kaynakça :

- [1]. Pal R. N., Pal S. K, A Review on Image Segmentation Techniques, Pattern Recognition, vol.26, no.9, pp 1277-1294, 1993.
- [2]. T. , Kohonen, Self-Organization and Associative memory, 3rd Edn. Berlin, 1989
- [3]. J.C.Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, NewYork, 1981
- [4]. J. Tou, R. Gonzales, Pattern Recognition Principles, Addison-Wesley, Massachusetts,1974
- [5]. Eric Chen-Kuo Tsao, Jamec C Bezdek, Nikhil R. Pal, Fuzzy Kohonen Clustering Networks, , Pattern Recognition, vol. 27, no. 5, pp 754-764, 1994
- [6]. T. L. Huntsberger, C. L Jacobs, R L Cannon. Iterative Fuzzy Image Segmentation, Pattern Recognition, vol 8, no. 2, pp. 131-138, 1985
- [7]. T. Huntsberger, P. Ajjimarangsee, Parallel Self Organizing Feature Maps For Unsupervised Pattern Recognition, Int. J. Gen. Syst. 16, pp 357-372, 1989
- [8]. H.Atmaca, M.Bulut, D.Demir , Histogram Based Fuzzy Kohonen Clustering Networks For Image Segmentation, 1996 International Conference on image Processing (ICIP'96), Switzerland, 1996



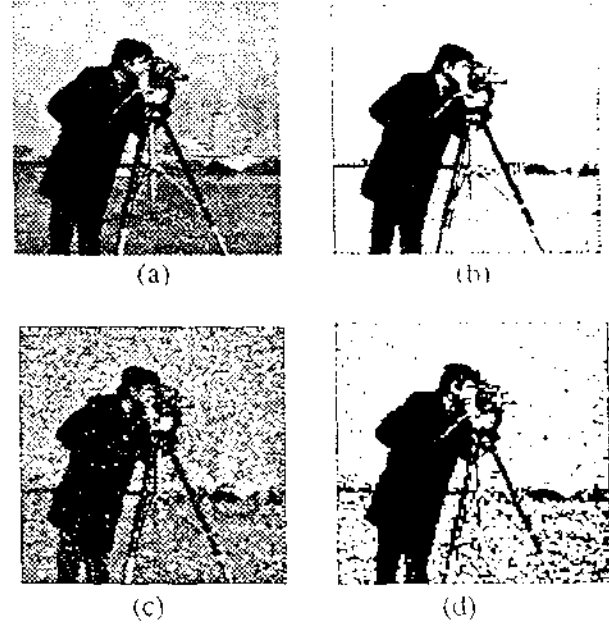
Şekil 3. (a) Orijinal House görüntüsü (b) İki topak ($c=2$) için bölütlenmiş görüntü (c),(d),(e) Üç topak için gri seviyelerin bulanık üyelik gösterimi, (f) histogram temelli FKCN kullanılarak üç topak için clusters composed from (c,d,e) deki topakların birleşimi olan üç topaklı ($c=3$) bölütlenmiş görüntü



Şekil 4. (a) Orjinal Gazete görüntüsü (b) ($c=2$) için histogram temelli FKCN ile bölütlenmiş görüntü



Şekil 5. (a) Orijinal Lenna görüntüsü; (b) iki topak ($c=2$) için bölütlenmiş görüntü; (c) üç topak ($c=3$) için bölütlenmiş görüntü ve (d) histogram temelli FKCN kullanılarak beş topak ($c=5$) için bölütlenmiş görüntü



Şekil 6. (a) Orjinal Kameraman görüntüsü (b) ($c=3$) için histogram temelli FKCN ile bölütlenmiş görüntü (c) Gürültü eklenmiş görüntü (d) Gürültülü görüntünün ($c=3$) için histogram temelli FKCN ile bölütlenmiş şekli

İKİ BOYUTLU KENARLARDAN ÜÇ BOYUTLU YÜZEYLERİN ALGILANMASI

F. ÜÇÜNCÜ, N. İKİZLER, C. YILMAZER

K.T.Ü. Müh.-Mim.Fak.

Elektronik. Müh. Bölümü

ABSTRACT

Inference of 3-D shape from 2-D contours in a single image is an important problem in machine vision.

In this study, classes of techniques proposed in the past were surveyed and a critical analysis was provided and the comparison of the results of this study with the results of the previous studies was done.

Two kind of symmetries in figures, which are known as paralel and skew symmetries, give significant informatior, about surface shape for a variety of objects. In this study, the constraints imposed by these symmetries were derived and using these constraints to infer 3-D shape was shown.

The zero Gaussian curvature (ZGC) was discussed in depth and for various ZGC surfaces, the results on the recovery of surface orientation were shown.

1. Giriş

Bilgisayar görmesinde, önemli problemlerden biri de iki boyutlu olarak algılanan şekil içindeki kenarlardan üç boyutlu cismin oluşturulmasıdır. Bu konu, şeklin tanınması için gereklidir. Bu yöntemde kullanılan asıl bilgi, iki boyutlu kenarlar arasında mevcut olan paralel ve çarpık simetridir. Bu simetrilere bazı sınırlamalar getirilerek bu sınırlamalar altında üç boyutlu görüntünün elde edilmesine çalışılmıştır.

Orta seviyeli görmenin ana amaçlarından biri cisimdeki nesnelerin yüzeylerinin kısmi yönlendirmelerini bulmaktır. Buradaki temel zorluk, görüntünün üç boyutlu cismin iki boyutlu yansıması olmasıdır. Bu nedenle üç boyutlu yapı bazı kabuller yapılmadan elde edilemez.

2. İki Boyutlu Kenarlardan Üç Boyutlu Yüzeylerin Algılanması

Yöntem, temiz ve kapalı sınır çizgilerinin verildiğini kabul eder. Bu nedenle burada yüzey işaretlerinden nesne kenar çizgilerinin ayrıştırılması ya da diğer algılanabilir gruplama işlemleri açıklanmayacaktır.

Bu çalışmada, ortografik yansıtma tekniği kullanılmıştır. Yüzeylerin yönlendirmelerini (normalleri ile verilen) göstermek için gradyan uzayı kullanılmıştır. $ax+by+cz+d=0$ düzleminin normali olan $N, N=(a,b,c)$ vektörü ile gösterilmiştir. Bu ifade $(p,q,1)$ şeklinde yazılabilir ki burada $p=a/c$ ve $q=b/c$ 'dir. Burada (p,q) , her noktası üç boyutlu düzlemin normaline karşı düşen iki boyutlu uzay olarak tanımlanabilir.

2.1. Simetrilere ve Niteliksel Sonuçlar

Simetrilere, şeklin algılanmasında önemli bir rol oynar. Burada, paralel ve çarpık simetri olmak üzere iki çeşit simetri tanımlanmış ve yüzeylerin oluşturulmasında nasıl kullanılabilecekleri açıklanmıştır.

Eğrilerin simetrik olabilmeleri için, iki eğri arasında belirli noktasal uygunlukların olması gerekmektedir. Eğriler üzerinde uygun gelen noktaları birleştiren çizgilere simetri çizgileri, bu çizgilerin orta noktalarına simetri eksenini ve simetriyi oluşturan eğrilere de simetri eğrileri adı verilmektedir.

A. Paralel Simetri

Yay uzunluğu olan s parametresine bağlı $X_1(s)=(x_1(s),y_1(s))$ eğrilerini $(1=1,2,..)$ ele alalım. $\theta_1(s)=\arctan((dy_1(s)/ds)/(dx_1(s)/ds))$ olsun. Eğer $X_1(s)$ ve $X_2(s)$ arasında aşağıda belirtildiği şekilde bir $f(s)$ uygunluk fonksiyonu varsa bu iki eğri paraleldir denir.

$$\theta_1(s) = \theta_2(f(s))$$

Burada s 'in her değerine karşı $X_1(s)$ ve $X_2(s)$ tanımlı ve $f(s)$ sürekli ve türevi alınabilir bir fonksiyon olmalıdır. Bu tanımlı kullanarak iki eğri arasındaki simetriyi hesaplamak, $f(s)$ fonksiyonunu kestirime dayanır. Bu durumun kullanışlı bir özel hali, $f(s)$ 'in doğrusal fonksiyon olarak sınırlandırılmış halidir. Bu durumda simetri şartı aşağıdaki şekli alır:

$$O_1(s) = O_2(as + b)$$

Burada, a ve b sabitlerdir.

B. Çarpık Simetri

Bu tür simetride, simetri eksenini doğrusal ve simetri çizgileri simetri eksenine aynı açıyı yapıyor olmalıdır. Simetri çizgilerinin, simetri eksenine ortogonal olduğu durumlarda $X_1(s)$ ve $X_2(s)$ eğrilerinin çarpık simetri olma koşulu:

$$K_1(s) = -K_2(s + b)$$

Burada $K(s)$ bükülme ve b bir sabittir. $u_1(s)$, simetri çizgileri ile bu çizgilerin i eğrisi ile yaptığı açı olsun. İki eğrinin çarpık simetri olması için gerek şart:

$$K_1(s) \sin(\alpha_1(s)) = -K_2(s + b) \sin(\alpha_2(s + b))$$

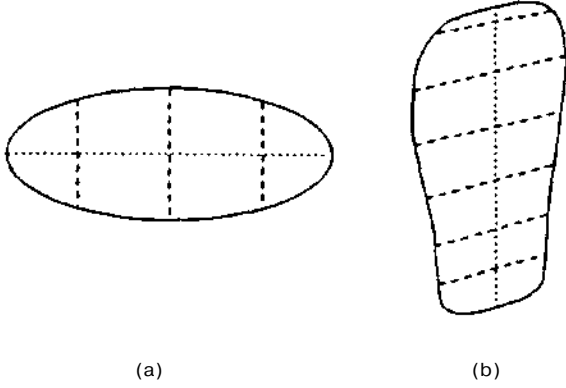
Yukarıdaki şartlar sadece eğriler için geçerlidir, çünkü doğrular için bükülme sıfır olacaktır. Çizgiler için, çarpık simetrisinin ilk tanımı şu şekilde açıklanabilir: Verilen çizgilerin uçlarını birleştiren diğer iki çizgi paralel ise, verilen iki çizgi çarpık simetriktir denir. Bu durumda iki yeni çizgi, simetri çizgileri olur ve bu çizgilerin orta noktalarını birleştiren çizgi de simetri eksenini olur.

C. Genel Bakış Açısı:

Bir cisim, görüntünün algılanabilir özelliği, bakış yönünün zayıf değişimleri altında korunuyorsa (sabit kalıyorsa), genel bakış noktasından görüntülenmiştir denir. Özel olarak, ilgilendiğimiz algılanabilir özellikler, düz çizgiler, çizgilerin paralellığı ve eğrilerin simetrisidir. Burada, bağımsız olarak tek yüzeyler incelenecektir.

D. Tek Çarpık Simetrlili Yüzeyler

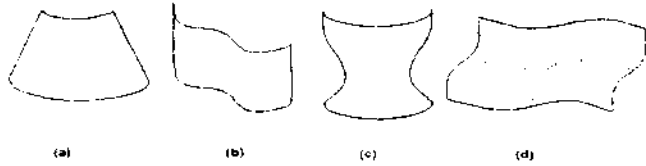
Tek bir çarpık simetri, yüzeyin bütün kenar çizgilerini kapsar. Tek çarpık simetrlili bütün yüzeyler genellikle düzlemsel olarak algılanır. Şekil 1.a'daki elips, merkezinden geçen simetri eksenini etrafındaki iki çarpık simetrlili eğri olarak tanımlanabilir. Bu durumda bütün simetrliler, elipsin sınırları üzerindeki bütün noktaları kapsar[1].



Şekil 1. Tek çarpık simetrlili yüzeylere örnek

E. İki Simetrlili Yüzeyler

iki simetrlili yüzeylerde, şeklin kenar çizgileri iki simetri ile örtülmüştür ve bu simetrlilerin en az biri paralel simetridir. Bu tür şekiller, yüzey şekli hakkında bize en fazla bilgiyi verenlerdir. Eğer iki simetriden biri düz simetri çizgilerine sahip çarpık simetri ise, yüzey, sıfır Gauss eğriliği yüzeyi olmalıdır. Aksi halde, eğer çarpık simetri eğrileri düz değil ise ya da eğik paralel simetri varsa, çift eğimli yüzey mevcut demektir (Şekil 2).



Şekil 2. İki simetrlili yüzeylere örnek

F. Sıfır Gauss Eğriliği (ZGC) Yüzeyleri

Sıfır Gauss eğriliği (ZGC) yüzeyleri, yüzeyin Gauss eğriliğinin (maksimum ve minimum eğriliklerin çarpımı) her yerde sıfır olduğu yüzeylerdir. Silindirikler ve koniler örnek verilebilir. Bu yüzeyler, bir parça kağıdı hiç kesmeden katlayarak ya da bükerek oluşturulabildikleri için, üretilebilir yüzeyler adını alırlar. Yuvarlama adı verilen ZGC yüzeyleri için

minimum bükülmenin çizgileri düzdür. Bu yuvarlamalar boyunca ZGC yüzeylerinin üzerine düz çizgiler yerleştirmek mümkündür.

Eğer paralel simetri doğrusal karşılık fonksiyonuna sahip ise, yüzey koniktir, eğer karşılık fonksiyonu birim ise, yüzey silindirikdir. ZGC yüzeylerinin yuvarlamaları ve çapraz bölgeleri şu şekilde anlaşılır: Yuvarlamalar, yüzeyin bükülmelerinin sıfır olduğu yerler boyunca olan çizgilerdir. Çapraz bölgeler, çaprazvari olan eğrilerdir, özellikle paralel simetri eğrileri olarak yansıyan eğrilerdir.

G. Yuvarlamaların Bulunması

ZGC yüzeyinin yönlendirilmesi yuvarlama boyunca değişmez. Bu nedenle, eğer yuvarlama üzerindeki tek bir noktada yüzeyin yönlendirilmesi bulunursa, bu, tüm yuvarlama boyunca geçerli olur.

2.2. Niceliksel Şekil Oluşturulması

Burada, niceliksel şekil oluşturulması için önerilen yöntem açıklanmıştır. Genellikle, ZGC yüzeyler üzerinde durulmuştur. Yöntem, aşağıdaki üç sınırlamanın kullanılmasına dayanır.

1) Eğimli paylaşılmış sınır sınırlaması {CSBC}: Bu sınırlama, kesişen iki yüzey üzerindeki noktaların yönlendirilmeleri arasında ilişki kurar[2]. Eğer iki yüzeyin kesişiminden oluşan eğrinin düzlemselliği kabul edilirse, daha basit bir sınırlama elde edilir.

ZGC yüzeyler için, paralel simetri eğrilerinin düzlemsel olduğu kabul edilmiştir.

2) iç yüzey sınırlaması {ISC}: iç yüzey sınırlaması, yüzeydeki komşu noktaların bağıl yönlendirmelerini sınırlar. ZGC yüzeyleri için, yüzeyin yuvarlamalarının görüntüsü, komşu noktaların yüzey yönlendirmelerini inşa etmek için kullanılır.

Geometrik olarak ISC aşağıdaki gibi açıklanabilir: Paralel simetri eksenini boyunca hareket edildikçe, yüzey yönlendirmesi, yuvarlamaların görüntüsüne ortogonal yönde (paralel simetri çizgileri), p-q düzlemi içinde hareket eder. Silindirik yüzeyler için, tüm yuvarlamalar birbirine paralel olduğu için ISC eğrisi, bir doğru şeklinde olur. Bu sınırlama, kenar çizgisi hakkında bir tahmine ihtiyaç duymaz.

3) ISC ve CSBC'nin Birleşimi

Eğer ZGC yüzeyi n adet nokta ile tanımlanırsa, 2n+2 adet bilinmeyen ortaya çıkar. Bunların 2n adeti yüzey yönlendirmeleri (p,q) için ve iki adeti de çapraz bölge düzlemi (p_c,q_c) içindir. Bu sınırlama bize n tane sınırlama eşitliği sağlar. CSBC, ISC ile birlikte kullanılırsa 2n-1 adet eşitlik elde edilir. Bundan dolayı, ZGC yüzeyi tanımlamak için üç derece özgürlük (bilinmeyen sayısının denklem sayısından üç fazla olması durumu) ortaya çıkar.

4) Ortogonalite sınırlaması (OC): Ortogonalite, paralel simetri eksenini ile paralel simetri çizgileri arasındadır. Bu, çarpık simetrlili düzlemsel dilimler elde etmek için yuvarlamalar boyunca yüzeyi

dilimlemeye ve bu dilimlerin Kanade'nin [3] çokkenarlıların analizinde olduğu gibi üç boyutlu uzayda ortogonal simetrik olduğunu kabul etmeye karşılık düşer Simetri çizgilerinin yüzeyin yuvarlamaları olduğu ZGC yüzeyleri için. ortogonalite sınırlaması, çapraz bölgelerin maksimum bükülme çizgileri boyunca olduğunu kabul eder.

2.3. Sınırlamaların Birleştirilmesi

Yukarıda açıklanan üç farklı sınırlama, n adet nokta için. $2n+2$ bilinmeyen üreten ((p.,q.) dahil) $3n-1$ adet sınırlama denklemi ortaya çıkarır. Bu. $n>3$ için denklem takımının sınırlama ötesi olduğunu gösterir.

Silindirik yüzey, yuvarlamalarının üç boyutlu uzayda birbirine paralel olduğu ZGC yüzeylerdir. Bu yüzey $X(u,v)=(x(u,v),y(u,v),z(u,v))$ ile ifade edilsin. Burada u . simetri eksenini boyunca ve v . yuvarlamalar boyuncadır. Simetri eksenini boyunca hareket edildikçe, simetri ekseninin tanjantı ile yatay arasındaki açı u(u) olsun u . sadece unun fonksiyonudur ve yuvarlama ile yatay arasındaki açının |i olmasını sağlar. Bütün yuvarlamalar paralel olduğu için \\ sabittir. |i 'yi ,i/2'ye eşitlemek için koordinat sistemini her zaman çevirebiliriz. Bu yöntemdeki asıl amaç. simetri eksenini boyunca (p(u).q(u)) yüzey yönlendirmesini hesaplamaktır. Çapraz bölge düzleminin yönlendirmesi (p.,q) olsun. Eğer p_c. q_c ve q_c'i sabitleştirirsek. u'nun bütün değerleri için (p(u).q.) yüzey yönlendirmesi sabit olur. Bu son sınırlama ortogonalite ile açıklanmıştır. Ortogonalite sınırlaması ile q 'yi değişken bırakarak p_c=0 ve q_c=-1/q_c elde edilir CSBC. ISC ve OC sınırlamaları silindirik yüzeyler için sağlanır ve q_c adını verdiğimiz serbestlik derecesi kalır

A. (p_c,q_c)'nin Kestirimi

Bahsedilen üç sınırlama çapraz bölge düzleminin yönlendirmesinin (p.,q) l: fonksiyonunun minimum çizgisi boyunca olması için sınırlandırılması gibi bir bağımsız değişken bırakır. Bu minimum çizgiyi hesaplamak çok zaman alır. Bunun yerine (p . q .) yi hesaplamak için aşağıdaki gradyan azalma algoritması kullanılır.

1) Çarpık simetri ekseninin yönünde olan ve p-q uzayında orijinden geçen bir l, başlangıç çizgisi seçilir (l=l).

2) Aşağıda açıklanan yöntemi kullanarak l çizgisi için (p.,q.) hesaplanır.

3) (p . q .) için l! fonksiyonunun değeri hesaplanır ve üstteki işlemi l çizgisinin rrtu derece yanındaki çizgiler için tekrarlayarak ve bu çizgiler için olan l' değerlerini karşılaştırarak (p_c.q_c)'nin Enin minimum çizgisi boyunca olup olmadığı kontrol edilir.

4) Eğer (p . q) L 'nin minimum çizgisi boyunca ise işlem sona erdirilir. Aksi halde l çizgisini E 'nin azalması yönünde sn açısı kadar çevirerek yeni bir çizgi seçilir ve ikinci adıma geri dönülür.

B. Verilen l Çizgisi İçin (p_c,q_c)'nin Hesaplanması

Koordinat sistemi, l çizgisi p-q düzleminin q eksenini ile kesişene kadar çevrilir. Dolayısıyla, p_c=0 ve q_c- bilinmeyen değerde olur.

q/yi sabitleştirmek için çapraz bölgenin şekli kullanılır Bu çalışmada matematiksel sınırlamalardan daha çok algılanabilir özelliklere dayanan bir yöntem önerilmektedir.

Yöntem, aşağıdaki insan algılamasına ilişkin gözlemlere dayanmaktadır.

1) Birleşik şekiller tercih edilir [4]

2) Çok düşük ve çok yüksek yana eğimler yerme orta seviyede yana eğimler tercih edilir

3) Algılanan yana eğimler için geniş sınırlı belirsizlik vardır.

Bu gözlemlere dayanılarak. q 'yi tespit etmek için iki adımlı bir yöntem önerilmektedir, ilk olarak q için bir değer kestirilir ve 45 derecelik bir eğimle güncelleştirilir. ilk kestirim için çapraz bölgeye bir elips uydurulmalıdır ve onu daire yapan bir yönlendirmeye geri yansıtılmalıdır.

C. q_c'nin İlk Kestirimi

q_c için ilk yaklaşımda elips uydurma işlemi kullanılmıştır. Çapraz bölge kenar çizgisine bir elips uydurulur ve q_c'nin ilk yaklaşımını elde etmek için. uydurulan elips olarak yansıtılması gereken (p...q_c) dairesinin yönlendirmesi. p-q düzlemi üzerindeki q eksenini üzerine yansıtılır. Buna q_c, adı verilir. Dairenin, görüntü düzlemine elips olarak yansıtmasını sağlayan iki adet (p_c.,q_c) değeri vardır. Katı cisim yorumu veren nokta seçilir

D. q_c'nin Güncelleştirilmesi

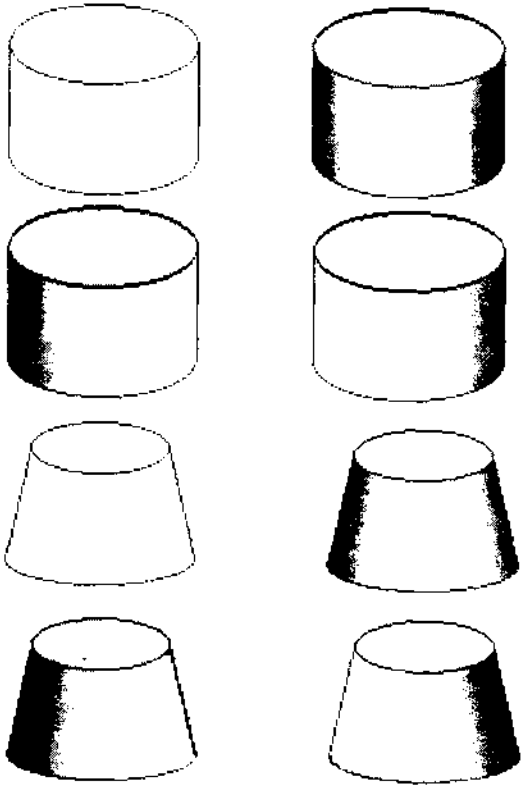
Bu güncelleştirme işleminin amacı, insanların orta seviyedeki yana eğimleri, çok az veya çok fazla yana eğime tercih etme davranışlarını taklit etmektir. En son q_c değerini aşağıdaki gibi elde etmek için q_c güncelleştirilir (q_c'yi dereceye çevirdikten sonra):

$$q_c = 45 + \lambda(q_c - 45)$$

Burada λ. [0,1] aralığında değişen bir güvenlik faktörüdür ve elipsin çapraz bölge eğrisine yaklaşıklığının fonksiyonudur. Denemeler göstermiştir ki elipse yaklaşım ne kadar iyi ise λ. değeri o kadar büyümektedir ve q_c 45 'ye yaklaşmaktadır ve daha az düzeltmeye ihtiyaç duymaktadır.

3. SONUÇLAR

Yukarıda söz edilen simetriklerin yardımıyla ve üretilen sınırlamalar kullanılarak yapay olarak oluşturulan kenar çizgilerine yöntem uygulanmış ve Şekil 3'deki sonuçlar elde edilmiştir



Şekil 3. Yapay kenarlar ve elde edilen üç boyutlu cisimler

Elde edilen sonuçlar, daha önce bu konu ile ilgili yapılan çalışmaların ve üretilen yöntemlerin sonuçlarıyla karşılaştırılmıştır.

Elde edilen sonuçlar şu şekildedir:

- 1) Sonuçlar, insan algılamasıyla uyumaktadır.
- 2) Üretilen sınırlamaların yardımıyla üç boyutlu yüzeylerin oluşturulmasında, diğer yöntemlere göre [5] daha az kabul gerekmektedir.

4. KAYNAKLAR

- 1) Ulupınar, F. ve Nevatia, R., Perception of 3-D Surfaces From 2-D Contours, IEEE Trans. Patt. Anal. Machine Intell., 15, 1 (1993), 3-18.
- 2) Mackworth, A. K., Interpreting Pictures of Polyhedral Scenes, Artif. Intell., 4, (1973), 121-137.
- 3) Kanade, T., Recovery of the Three Dimensional Shape of an Object From a Single View, Artif. Intell., 17, (1981), 409-460.
- 4) Brady, M. ve Yuille, A., An Extremum Principle For Shape From Contour, IEEE Trans. Patt. Anal. Machine Intell., PAMI-6, (1984), 288-301.
- 5) Stevens, K. A., The Visual Interpretations of Surface Contours, Artif. Intell., 17, (1981), 47-73.

FORMANT FREKANSLARIN SES TANIMADA KULLANILMASI

Nuri İKİZLER, Rifat YAZICI, Cengiz YILMAZER, Fatih ÜÇÜNCÜ
Karadeniz Teknik Üniversitesi
Elektrik ve Elektronik Mühendisliği Bölümü
61080-TRABZON

ABSTRACT

Speech signals are the most important fact for the communication. Human beings transmit their feelings, thoughts and desires through speech signals. These signals can be used to achieve a communication between human beings and the computers. If the computers recognize and execute the speech commands produced by human beings, this will allow the normal and handicapped people to make use of computer technology in large extent.

The speech recognition process can be named as speech recognition. Developing of the speech recognition systems is quite important for the handicapped people to control electronic equipments around them. Also speech recognition systems can be used in the electronic technologies.

The importance of the speech recognition systems can be seen from their large usage. So engineers have been investigated to create the best speech recognition system for years.

For this purpose, Artificial Neural Networks will be trained by the parameters that will be obtained from the analysis of the speech signals using Linear Predictive Coding technique. The results will be tested by new patterns supplied later. Finally the obtained speech recognition rate will be discussed in various respects.

1. Giriş

Bu çalışmada, değişik ortamlarda söylenmiş Türkçe ve İngilizce'deki kelimelerin Doğrusal Öngörümleme Kodlama yöntemi kullanılarak analizi yapılmış, analiz sonucu ses işaretinin belirleyici bir parametresi olan formant frekanslar bulunmuştur. Hatanın Geri Yayılımı ile öğrenme (Error-Back Propagation Training) yöntemi kullanılarak oluşturulan çok katmanlı bir Yapay Sinir Ağı, bulunan bu formant frekanslar ile eğitilmiştir. Daha sonra eğitilen Yapay Sinir Ağı, farklı ortamlarda yeniden alınan sesler ile test edildiğinde ses tanımada oldukça başarılı sonuçlar vermiştir.

2. Ses İşarelerinin D.Ö.K. İle Analizi

Yapılan çalışmada değişik seslerin 1. ve 2. formant frekansları, o sese ait belirleyici parametre olarak kullanılmıştır. Gürültü düzeyi farklı ortamlarda bir mikروفon tarafından alınan ses işaretlerine; örnekleme, kuvantalama, kodlama ve pencereleme işlemleri sırasıyla uygulanmıştır.

Ses işaretinin analizinde kullanılan birçok yöntem mevcuttur. Bu yöntemlerin en önemlilerinden biri Doğrusal Öngörümleme Kodlama yöntemidir. DÖK yönteminin temeli DÖK katsayılarının ($a_k, k=1,2,\dots,P$) belirlenmesidir. Bu katsayılar doğrudan ses işaretinden elde edilebileceği gibi, ses işaretine ilişkin özilişki fonksiyonu kullanılarak da elde edilebilir. Sesin dalga şeklindeki birbirini takip eden örnek değerler arasında yüksek bir ilişki vardır. Bu ilişki göz önüne alınarak, Doğrusal öngörümleme aşağıdaki şekilde kabul edilebilir. [1]

$$y_n = a_0 y_{n-1} + \alpha_1 y_{n-2} + \dots + \alpha_p y_{n-p} \quad (D)$$

Bu ilişki şu şekilde açıklanabilir. Sesin dalga şekli, p sayıdaki örnek değerlerin ağırlıklı toplamının, bir sabit olan a_0 ile çarpılması ile önceden tahmin edilebilir. Bu katsayılar, en küçük kareler ortalama öngörümleme hatası bir kriter alınarak, doğrusal öngörümleme katsayıları ($a=1,2,\dots,p$) olarak isimlendirilirler. Bu durumda bu katsayıların bulunma işlemine Doğrusal Öngörümleme Analizi adı verilir. Doğrusal Öngörümleme analizinde tüm-kutup sistemi kullanılır. Buradaki kutuplar ses tayfinin formantlarına karşı gelir.

Bir tüm-kutup sisteminde, sistemin cevap fonksiyonu $H(z)=1/A(z)$ ve $A(z)=0$ için z değerleri $H(z)$ 'nin kutuplarını verir. Bunun için doğrusal öngörümleme katsayıları a_i 'ler analizden elde edildikten sonra, aşağıdaki eşitlik kullanılarak kutuplar bulunabilir.

$$z^p + \alpha_1 z^{p-1} + \alpha_2 z^{p-2} + \dots + \alpha_{p-1} z + a_0 = 0 \quad (2)$$

(2) eşitliği P dereceden gerçek katsayıları olan bir ifadedir. Genelde $P/2$ sayıda kökü vardır. Çünkü kökler kompleks ve eşleniktir. Herhangi bir çift;

$$z_1 \sim r_1 e^{j\theta_1}, z_2 = r_1 e^{-j\theta_1} \quad (3)$$

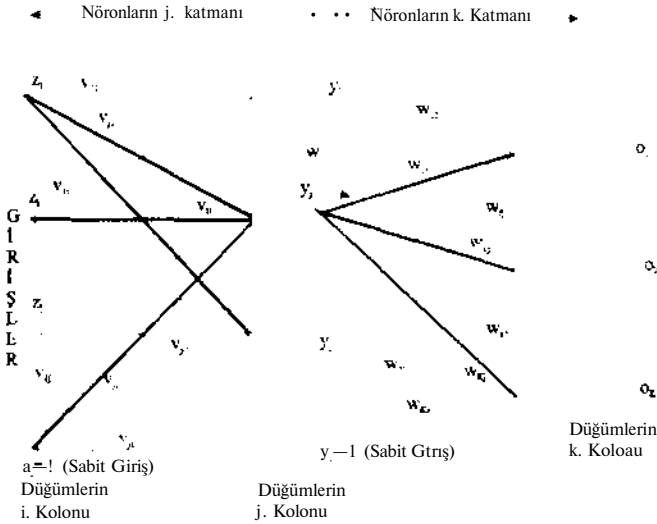
şeklinde yazılırsa, formant frekansların eşitliğindeki şekilde ifade edebilir.

$$f_1 = \frac{\omega_1}{2\pi} = \frac{1}{2\pi T} \cdot \arg(z_1) \text{ Hz} \quad (4)$$

şeklinde ifade edilebilir [1].

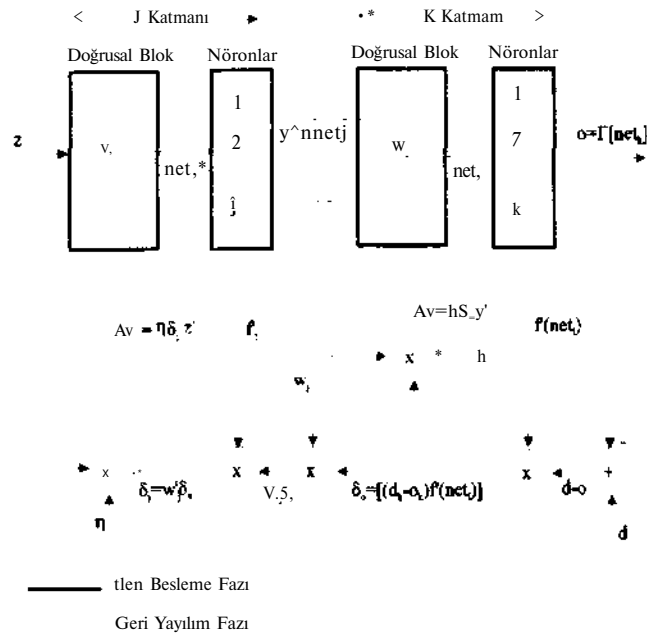
3. Yapay Sinir Ağı Yapısı ve Hatanın Geri Yayılımı ile Öğrenme

Şekil 1'de çalışmada kullanılan ileri beslemeli Yapay Sinir Ağı Yapısı verilmiştir [2]



Şekil 1 İki tane sürekli algılama katmanı ile katmanlanılmış ileri beslemeli ağ.

Şekil 2'de ise Hatanın geri yayılımı ile öğrenme (H.G.Y.İ.Ö.) ağ işlemi blok diyagram halinde verilmiştir ve bu diyagram Hatanın geri yayılımı ile öğrenme yöntemini bütünüyle açıklar. Ağ içindeki hatanın akuda burada görülebilir [2].



Şekil 2. Hatanın geri yayılımı ile öğrenmede, ileri ve geri sinyal akışını gösteren diyagram.

Değişik ortamlarda A kişisi için kaydedilen kelimelerin formant frekans değerleri giriş olarak Yapay Sinir Ağına verilmiştir. Bu değerlerden bazıları Tablo 1 de gösterilmiştir. Çıkış olarak ise kelimenin kendisi alınmıştır. Hatanın geri yayılımı ile öğrenme yöntemi uygulanarak ağırlık matrisi hesaplanmıştır. Daha sonra bu yapıya daha fazla sayıda kelime eklenmiştir. Yapay Sinir Ağına öğretilen kelimeler farklı ortamlarda yeniden kaydedilmiş ve çıkışın hangi kelimeye daha fazla yakınsadığı tespit edilmiştir [3]. Bu tespit sonucu elde edilen değerlerden bazıları Tablo 2. de gösterilmiştir.

Tablo 1 Farklı zamanlarda söylenmiş ses örneklerinin 1 ve 2 formant frekans değerleri

Ses örnekleri (1.2=3)	1. Formant Frekans (Hz)	2. Formant Frekans (Hz)
A sesi (1)	465.3	875.6
A sesi (2)	456.1	857.6
A sesi (3)	461.8	807.2
E sesi (1)	406.9	1653
E sesi (2)	430.1	1664
E sesi (3)	411	1657
I sesi (1)	245.1	1987
I sesi (2)	251.6	2025
i sesi (3)	275.2	1952
i sesi (1)	293.9	1044
I sesi (2)	300.5	1038
I sesi (3)	300	1050
Ü sesi (1)	275	1447
Ü sesi (2)	286.8	1416
Ü sesi (3)	286.9	1419
Ö sesi (1)	368.4	1370
Ö sesi (2)	348.3	1334
Ö sesi (3)	363	1342
O sesi (1)	367.6	748.2
O sesi (2)	375.1	736.1
O sesi (3)	385.6	738.2
U sesi (1)	274.4	571.1
U sesi (2)	270.4	577.8
U sesi (3)	269.3	574.4
Dir sesi (1)	301	1575
Dir sesi (2)	309.6	1585
Dir sesi (3)	313.9	1597
Help sesi (1)	389	1464
Help sesi (2)	393.5	1421
Help sesi (3)	390.3	1446
Mem sesi (1)	418.1	1478
Mem sesi (2)	435.5	1466
Mem sesi (3)	440.8	1473
Say sesi (1)	370.2	1723
Say sesi (2)	375.7	1705
Say sesi (3)	370	1708
We sesi (1)	308.7	1924
We sesi (2)	299.2	1938
We sesi (3)	287.4	1922

Tablo 2 F1 ve F2 formant frekanslarına göre eğitilen Yapay Sinir Ağı'nın daha sonra girilen ses örneklerini tanıma yüzdesi

Ses Örneği	Ses örneğinin Formant frekanslarında öğretilen sese göre sapma	Yapay Sinir Ağına Öğretilen kelime sayısı	H G Y l O de ; hata- nin sen değan	Ses tanı- ma yuz- desi (%)
a	20	8	0.0001	100
e	20	8	0.0001	100
h	20	8	0.0001	99
r	40	8	0.0001	93
ü	40	8	0.0001	93
ö	40	8	0.0001	100
o	20	8	0.0001	100
u	40	8	0.0001	100
dir	20	10	0.01	98
help	20	10	0.01	90
mem	40	10	0.01	100
say	40	10	0.01	100
we	20	10	0.01	98
are	20	10	0.01	98
you	20	10	0.01	96
am	40	10	0.01	100
me	20	10	0.01	100
we	20	15	0.02	100
are	20	15	0.02	97
you	20	15	0.02	95
am	20	15	0.02	95
go	20	15	0.02	99
The	20	15	0.02	85
Ishe	20	15	0.02	95
come	20	15	0.02	100
play	20	15	0.02	100
time	20	15	0.02	100
date	20	15	0.02	95
at	40	15	0.02	95

4, SONUÇLAR

Doğrusal Öngörümleme Kodlama analiz yöntemi ses işaretleri parametrelerinin doğru ve ekonomik gösterimine söz tanımada doğruluğun artırılmasına analiz hesap zamanının azalmasına olanak sağlayan bir yöntem olarak gözlenmiştir.

Formant frekans hesaplamalarında %5 kadar düşük bir oranda dahi olsa iraksak sonuçlara sebep olmuştur. Bu iraksamanın nedeni olarak yapılan ses kayıtlarının bozukluğu ortamdaki gürültü etkisi

yada ses analiz aralığının yanlış seçimi gösterilebilir.

Düşük bir yüzde ile görülen iraksamanın dışında birbirinin aynı dilbilimsel mesajı taşıyan ses işaretleri için sonuçların birbirine olan yakınsaması yapılan ses analizinin oldukça yüksek bir doğrulukta olduğunu göstermiştir.

Yapay Sinir Ağları kullanılarak yapılan ses tanıma işlemi oldukça başarılı sonuçlar vermiştir. Eğitilen Yapay Sinir Ağı işlem hızı bakımından oldukça hızlı olduğu halde eğitim süresinin tanınması gereken kelime sayısı arttıkça uzaması sistemin önemli bir dezavantajıdır. Yapay Sinir Ağına öğretilecek olan formant frekans değerlerinin sayısının kısıtlı olmasında tanıma oranında sistemdeki büyümeyle orantılı olarak bir düşmeye neden olmaktadır. Yapay Sinir Ağı'nın eğitimi tamamlandıktan sonra, çok kısa olan tanıma süresi, bu ses tanıma sisteminin oldukça önemli bir avantajıdır. Bir mikrofon, 486DX tabanlı bir bilgisayar ve 16 bitlik bir ses kartından oluşan donanım gereksinimi benzerlerinin yanında oldukça düşük bir maliyete sahiptir.

Kaynakça

- [1] Shuzo S., ve Kazuo N., Fundamentals of Speech Processing, Third Edition, Academic Press, NewYcrk, 1955.
- [2] Zarada J. M. Introduction of Artifical Neural Systems, Second Edition West Publishing Company St. Paul, 1990.
- [3] İkizler N. Doğrusal Öngörümleme Kodlama ve Yapay Sinir Ağı Yöntemlerinin Ses Tanımada Kullanılması, Yüksek Lisans Tezi, K.T.Ü. Mühendislik-Mimarlık Fakültesi, Trabzon, 1995.