

BİLGİSAYAR İLE
DEVRE TASARIMI SORUNUNA
TOPLU BİR BAKIŞ

YURDAKUL CEYHUN

UDK: 621.3.01 : 681.14

ÖZET

Bu bildiride bilgisayar yardımı ile özdevimsel devre tasarımı sorunu incelenerek bu konuda varılan aşamalar kısaca anlatılmıştır.

SUMMARY

in this paper a brief review of computer aided netvrork design in given.

1. GİRİŞ

Günümüzün insan yapısı yarattığı bilgisayar hemen her konuda olduğu gibi devre tasarımında da karşımıza çıkıvermekte. Çoğu kez bizlere yararı dokunan bilgisayar zaman zaman kullanana zarar da getirebiliyor. Bu yazıda bilgisayar desteği ile devre tasarımı sorununa topluca bakarak konunun kuramsal sorunlarına derinlenmesine inmeden bugüne dek bu konuda varılmış aşamaları kısaca özetleyeceğiz. Yazının sonunda ise bütün bu yararlarının yanı başında bilgisayarın bizlere getirebileceği zararlar üzerinde ilerde bu yolu denemek isteyenlere birkaç uyarıda bulunacağız. İlkın tasarımdan ne anladığımızı açıklamakla işe başlayalım.

2. DEVRE TASARIMI NEDİR ?

Devre kuramcısı olarak elimizde, öğelerin arasındaki bağlantılardan oluşan karmaşık bir dizge, başka bir deyişle bir N devresi, ve bu devreyi oluşturan imler kümesi vardır. Bu imlerin bir bölümünü uyarı imleri (e) diğer bir bölümünü ise tepke imleri (r) olarak iki alt kümeye ayırabiliriz. N devresi e ve r imleri arasında f işlevsel bağıntısını

Yurdakul Ceyhun, Y.Prof.Dr., ODTÜ

Elektrik Mühendisliği 219

6. teknik kongre

tanımlar. Simgesel olarak bu bağıntıyı

$$r \cdot f \cdot o \cdot e \quad (D)$$

diye gösterelim. Örneğin s-uzayında, doğrusal bir N devresi için

$$\begin{aligned} e &\Rightarrow E(s) \\ r &\cdot R(s) \\ f &\cdot F(s) \end{aligned}$$

olarak, Denklem (1)'i daha belirtik biçimde,

$$R(s) = F(s) E(s) \quad (2)$$

diye yazabiliriz. Denklem (2)'de F(s) nXm boyutunda ve öğeleri s karmaşık değişkeninin oransal işlevlerinden oluşan dikdörtgenel bir matristir. Ya da zaman uzayında bu N devresi

$$i(t) = Ax(t) + Be(t) \quad (3a)$$

$$r(t) = Cx(t) + De(t) \quad (3b)$$

denklem dizileri ile tanımlanabilir. Bu kez

$$\begin{aligned} e &- e(t) \\ r &- r(t) \\ f &- (A, B, C, D) \end{aligned}$$

dir. A, B, C ve D ise uygun boyutlarda gerçel matrislerdir. Her iki örnekten de görüldüğü gibi f kesinlikle e ve r imlerinden bağımsız, yalnızca N devresi tarafından tanımlanıyor. N devresinin tanımlayacağı f'nin bulunmasına kısaca, n devresinin çözümlenmesi diyeceğiz. Yapılan bir çözümlenme sonucu elde edilen f için verilen uyarı imleri sonucu, tepke imlerinin bulunmasına ise N devresinin çözümü diyeceğiz.

Şimdi çözümlenme işlemine tersten bakalım. Verilen bir f için (başka bir deyişle bir uyarı-tepke im çifti için) çözümlendiğinde bu f'yi tanımlayacak bir N devresi var mıdır, ve eğer varsa bu devre nedir? İşte bu soruna bireşim ya da f'nin N devresi ile gerçekleştirimi diyoruz.

Çözümlemenin genellikle birik sonuç vermesine karşılık, bireşimin sonucu hiçbir kez birik değildir. Örneğin, kR ve (1-k)R değerinde iki direncin seri bağlanmasından doğan bir devrenin, k'nın bütün değerleri için tanımlayacağı f'ler

$$f = R \quad (4)$$

dir. Bu da Denklem (4)'de verilen f'yi sonsuz sayıda N'nin gerçekleştirebileceğini gösterir. İlkin önemsiz belki de yararsız gibi gözükken bireşimin birik olmaması, temelde çok önemli başka bir sorunu ortaya koyar. Şöyle ki, eğer bireşim sonucu birik olsaydı, istenen f'yi gerçekleştiren N devresi bulunduktan sonra yapabileceğimiz başka bir iş kalmazdı. Oysa, görüyoruz ki elimizde istenen f'yi gerçekleştirebilecek sonsuz öğeli bir $\{N_0, N_1, N_2, \dots, N\}$ devreler kümesi var. O zaman bu kümeden öyle bir N = N^ devresi seçelim ki, bu devre yalnız ereğimiz olan f'yi gerçekleştirmekten öteye bizim isteğimize kalmış başka ereklere de sağlasın. Bu yeni sorun daha biçimsel olarak şöyle ortaya konabilir.

Gerçekleştirimi istenen N devresine ilişkin bir tasarım göstergesi

$$I = I(x_1, x_2, \dots, x_n) \quad (5)$$

tanımlayalım. Burada Xf'ler bizim isteğimize kalmış tasarım ereklere olsun. Örneğin,

x_1 * N'de endüktans olmayacak.

x_2 * N'deki dirençlerin değeri R_0 dan küçük olacak,

x_3 * N'de olabildiğince az sayıda öğe bulunacak,

x_4 * Tepke imleri r, N'deki öğelerin değerlerinde görülebilecek küçük oynamalara çok duyar olmayacak, vb.

Hem fin gerçekleştirimi hem de I'nın isterlerinin sağlanması genellikle sorunun çözümünü olarak dışı bırakabilir. Çözüme bir esneklik getirmek için

$$I \text{ s } ||f_1 - f|| \quad (6)$$

diye tanımlanan ve f'ye bir f yaklaşıklıkta içinde ilişik diğer f'leri düşünelim. Başka bir deyişle, sorunumuz I'nın isterlerini olabildiğince sağlayacak N^ devresini bulmak yerine, f'ye bir Z yaklaşıklıkta içinde kalan f^'yi sağlayacak ve de I'nın tüm isterlerini olabildiğince sağlayacak Nj^ devresini bulmak olsun. İşte biz bu N^ devresinin bulunmasına devre tasarımı diyoruz. Demek ki devre tasarımı bize bireşim sonucu bulacağımız devreler içinde, belli bir tasarım göstergesine göre en iyi devreyi vermektedir. Bu en iyi devrenin bulunması, genellikle çok sayıda başka devrelerin türetilmesini, bulunan bu devrelerin her birinin çözümlenmesini, tasarım göstergesinin belli değişikliklere göre türevinin alınmasını vb. gibi işlemleri gerektirecektir, tşte bu nedenlerden ötürü, böyle bir sorunu elle çözmek-olanaklarımızın dışına taşar ve bilgisayarın bir destek, bir yardımcı olarak kullanılması kaçınılmaz bir gerek olarak karşımıza çıkar.

Şimdi bilgisayarla devre çözümlenmesinin nasıl yapıldığını ve bu konuda varılan aşamaları kısaca anlatalım.

3. BİLGİSAYARLA DEVRE ÇÖZÜMLEMESİ

Devre çözümlenmesi için geliştirilen yöntemler devrelerin direnç, sığaç, bağımlı kaynaklar vb. gibi düşüncel öğelerden oluştuğu **sayılıtısma dayanır**. Oysa biz böyle soyut öğelerle değil **transistor vb.** gibi somut öğelerden oluşan devrelerin **taaarımları** ile ilgilimiz. Bu nedenle, yapılacak ilk 19 somut öğelerin soyut öğeler türünden biçimlenmesini yapmaktır. Yine sayısal işlemlerde hem kolaylık sağlamak için hem de bilgisayarın içinde işlemler sonucu oluşabilecek yanılıgıları önlemek için belirli ölçeklemelerin yapılması gerekebilir. Bütün bu ön işlemler yapıldıktan sonra ortaya somut bir

6. teknik kongre

devrenin soyut öğelerden oluşmuş eşdeğeri çıkar, ve sorun bu devrenin çözülmesidir. Burada başka bir özelliği açıklayalım. Devre kuramında biz yalnız düşüncel transformatörlerden oluşan devrelerin çözümü gibi gerçekte karşılaşmayacağımız düşüncel sorunlarla da ilgileniriz. Böyle soyut devrelerle ilgilenince de karşımıza bir devrenin durum denklemleri vb. yazılabilir mi ya da bir devrenin çözümü var mı gibi çoğunlukla soyut sorunlar çıkar. Devre tasarımı söz konusu olunca, sonunda somut olarak bir devre tasarlayacağımız için ilgilendiğimiz ve de ilgileneceğimiz tüm devreler için her tür denklemlerin yazılabileceğini ve de bunların çözülebileceğini en baştan varsayalım. Bu ön açıklamalardan sonra, şimdi ilk sorunumuza dönelim. Çözümleme için kullanılan yöntemleri dokumsal ve sayısal olarak iki kümede toplayabiliriz. Bu yöntemlerin ana özelliklerini sırasıyla özetleyelim.

3.1 Devre Çözümlemesi İçin Dokumsal Yöntemler

Bir devrenin çözülmesi için belli bir takım belirtenlerin ve alt belirtenlerin bulunması gerekir. Oysa çizge kuramından [1,2] bu belirten ya da altbelirtenlerinin değerlerinin o devrenin çizgesinde ya da bu çizgeye ilişik başka çizgelerdeki her bir ağaç ya da tümler ağaçtaki admitans ya da empedansların çarpımlarının bütün ağaç ya da tümler ağaçlara göre toplamlarına eşit olduğunu biliyoruz [3]. Demek ki bilgisayarda verilen bir çizge için bütün ağaçları bulan bir altyordam kullanarak istenen devre çözümlenebilir. Bu yöntemin üstün bir yanı, sayısal bir işleme pek gerek olmadığı için FORTRAN dili gibi sayısal işlemlere dönük bir dil kullanılsa bile istenen sonuçların bilgisayarın yeteneklerini zorlamadan simgesel olarak elde edilebilmesidir. Yine bu nedenden rasgele bir değişkene göre istenen bir denklemin türevi de kolaylıkla alınabilir. Yalnız bu yöntemin en büyük sakıncası, çizgedeki ayırıt sayısı arttıkça çözüm süresinin giderek çok uzun zaman almasıdır, örneğin, bu yöntemle çalışan CALAHAN [4], 15-20 ayırıtlı bir devreyi 1 dakikada, 25-30 ayırıtlı bir devreyi ise 1 saate yakın bir sürede çözümlenebilmektedir.

3.2 Devre Çözümlemesi İçin Sayısal Yöntemler

Zaman uzayında durum denklemleri ve s-uzayında düğüm, çevre denklemleri vb. gibi herhangi bir yöntem incelendiğinde bu yöntemlerin devre öğelerinin sayısal değerleri kullanıldığı sürece bilgisayara kolaylıkla uygulanabileceği görülür [5]. Yalnız incelenen devreler birkaç yüz düğümlü ve birkaç bin ayırıtlı ise bu yöntemlerin doğrudan doğruya uygulanması bilgisayarsa bellek yetersizlikleri gibi bazı sakıncalar doğurabilir. Bu sakıncaları yok etmek için değişik yöntemler uygulanabiliriz. Örneğin TAP [6], NET-1 [7], ECAP [9] gibi programlar, ilk kez Kron'un önerdiği, devreyi altdevrelere parçalayıp sonra yeniden

birleştirme yolunu kullanır [10]. Böylesine büyük devrelerin çözülmesinde başvuru başka yolların başında ise alt üst ayrışımını kullanmak [11] ya da elde edilen matrislerin seyreltikliğinden yararlanmak [12] gibi değişik sayısal çözümleme yöntemleri gelmektedir.

Eğer doğrusal olmayan devrelerin çözülmesi isteniyorsa durum denklemlerinin kullanılması kaçınılmaz tek çıkar yoldur. Yalnız bu durumda hangi değişkenlerin durum değişkeni olarak kullanılacağı sorununun yanıtı birik değildir. Örneğin bu tür devrelerde akım/gerilim ya da yük/akı değişkenlerinin bir bölümü durum değişkenleri olarak seçilebilir. Chua-Rohrer ve Desoer-Katzenelson-Stern algoritmaları durum değişkenlerinin seçimi için kullanılan belli başlı yöntemlerdir [13]. Elde edilen denklemlerin çözümü ise sayısal çözümlemenin bilinen yöntemleri ile sağlanabilir.

Genellikle durum denklemlerinin kurulması ilkesine dayalı devre çözümlemesi yapan programlar arasında POTTLE [14], CIRCUS [15], SCEPTRE [16], PREDICT [17], TAG, STRAP, TRAC [18], CIRCAL [19], ve AEDNET [20] sayılabilir. Doğrusal olmayan devrelerin özel bir altkümü olan çok değerli dirençsel devrelerin çözülmesinde MECA [21] oldukça yetenekli programlardan biridir. Bu arada özellikle Kron'un yöntemine dayanarak elektrik devrelerinin çözümlenme ilkeleri ile mekanik ve yapısal devreleri çözen DYANA [22] ve STRESS [23] programlarından da söz edilebilir.

4. ÖZDEVİMSSEL DEVRE TAŞARIMI

Buraya kadar tasarımın tanımını ve bilgisayarla devre çözümlemesi olanaklarını inceledik. Şimdi, bilgisayarın insan denetimi olmadan nasıl özdevimsel devre tasarımı yapabileceğini inceleyelim. Genellikle I tasarım göstergesi devre duyarlıklarının [24] bir işlevi ol atak seçilebilir. I'nın en iyi değerinin saptanabilmesi ise I'nin türevlerinin bulunmasını gerektirir [25]. Bu nedenlerden ötürü devre tasarımı sürekli türevler olarak ve belli bir matematiksel yöntemi izleyerek en iyi N devresinin bulunması olarak düşünülebilir. Eğer dokumsal yöntemler kullanılıyorsa türev almak bilgisayar için bir sorun değildir. Ama sayısal çözümlemelerde türev alma başlı başına bir sorun olarak karşımıza çıkar. Bu sorun arakarsılı devre kavramını kullanarak başka bir devrenin sayısal çözümlemesine indirgenebilir [26].

En iyi devrenin bulunmasında izlenebilecek bir başka yol da eşdeğer devre kavramını kullanmaktır [27, 28]. Hangi yol izlenirse izlensin ilk başlangıç devresi N_0 ise yine bilgisayar tarafından özdevimsel olarak bireştirilebilir [29]. Bütün bunlardan görüyoruz ki bilgisayar ilk başlangıç devresinin bireşimini yaptıktan sonra ya arakarsılı devreyi çözümlenerek ya da eşdeğer devreler türetmek tasarımı özdevimsel olarak gerçekleştirebilir.

5. ÇİZİMSEL VERİ ÇÖZÜMLEMESİ İLE DEVRE TASARIMI

Özdevimsel tasarımda işlem başladıktan sonra kullanıcı tasarım süreci içindeki tüm denetimini yitirmektedir. Bunun doğuracağı sakıncaları önlemek ve tasarım sürecine esneklik getirmek için bilgisayar ile kullanıcı arasında bir iletişim kurulması istenebilir. Verimin düşürülmemesi için en iyi iletişim ortamı bugünkü düzenlerde KIT (Katot Işınlı Tüp) göstergesi olarak saptanmıştır. Kullanıcı, KIT göstergesi üzerine bir ışık kalemi ile devreyi çizerek bilgisayara yapması gerekli işlemleri bildirir ve bilgisayar bu işlemler sonucu ortaya çıkan devreyi kullanıcıya yine KIT göstergesi üzerine çizer. Böylelikle kullanıcı ve bilgisayar arasında verimli bir iletişim ortamı çıkmış olur. Bilgisayara KIT göstergesi üzerinde çizimsel olarak bildirilen verilerin işlenebilmesi için bu çizimlerin ilk önce bilgisayarca çözümlenmesi gerekir. Bu çözümlenmeyi yapabilecek bir çok program arasında örneğin GINA ya da CALLIGRAPH [30] gösterilebilir.

6. DİĞER UYGULAMA YERLERİ

Yukarıda ana ayrıntıları ile açıkladığımız toplu öğelerden oluşan devrelerin tasarımından öteye bilgisayar diğer tür devrelerin de tasarımı sırasında yine bizlere yardımcı olabilmektedir. Örneğin, basma devrelerdeki karmaşık tel bağlantıları için en uygun yolun bulunması gibi. PUZZLE [31] ya da ACCEL [32] bu tür tasarım sorunlarını çözmede oldukça yetenekli programlardır.

Başka bir örnek ise CAMP programıdır [33]. Bu program, tümleşik devrelerin yapımında gerekli ana maskaların bilgisayar ile tasarımı yapılmasına olanak sağlar.

"Devre" sözcüğünden anladığımızı yalnız direnç, sığaç gibi öğelerden oluşmuş düzenlerle kısıtlamaz ve daha geniş anlamıyla "örü" leri düşünürsek, bu güne dek geliştirdiğimiz yöntemlerin kolaylıkla daha geniş kapsamlı sorunlara da uygulanabileceğini görürüz [34]. Örneğin bir iletişim örüsünün tasarımı ya da bir demiryolu ağının tasarımı yine devre ve çizge kuramının olanakları ve bilgisayarın desteği ile yapılabilir.

7. SONUÇ

Bu kısa irdelememizde devre kuramının en son ve de en ilginç sorununa kısaca değindik. Konu tüm sorunlarını bile ortaya koymasını tamamlamamış, emekleme aşamasında henüz. Gerek devre kuramının olanaklarını, gerek matematik ve gerekse bilgisayarın donanım ve yazılım olanaklarını zorlamakta ve her geçen gün yeni buluşlara yeni uygulamalara yol açmakta. Kısaca bir araştırmacı için hem kuramsal hem de uygulama açısından sonsuz olanaklar sağlamakta. Bir mühendis olarak ise bizleri ister istemez hayli etkileyebilir. Örneğin, bir süzgeç tasarımını gerçekleştirmek için neden eski yöntemlerle uğraşalım, neden bir bilgisayar düzeni ile bu sorunu hemen çözmeyelim? Belki her mühendisin aklından geçebilir bu soru. İşte bu nedenden yazının en başında bilgisayarın bize getirebileceği yarar ve zararlara değineceğiz demistik. Konuyu kapatmadan önce böylesine aceleci bir tutuma ilişkin bir kaç uyarıda bulunmak gerekli olacaktır. Yanıtlamamız gereken ilk soru kaç tasarım yapacağımız ve bu tasarımların ile de bilgisayarla yapılmasının ne derece zorunlu olduğunu. Çok küçük bir bilgisayar düzeni, kursak bile, bunun için ödeyeceğimiz para, bu düzende tasarılanacak devrelerden elde edeceğimiz kazancı aşabilir. Biz yalnız yenilik ve de kolaylık diye gereksiz bir yatırıma mı girişiyoruz? Bu sorun kesinlikle açıklığa kavuşturulmalıdır. Bu düzenler genellikle üniversite ve araştırma kuruluşlarında özel erekler için geliştirilmiştir. Çoğu kez genel ereklerle yapılmış bir bilgisayar düzeninde çalışmaya-bilir. Bir an için isterlerimizi karşılayacak bir programı satın aldığımızı düşünelim. Böylesine hazır programlarda ilerde karşımıza çıkabilecek sorunları çözmek için yapmak isteyeceğimiz en küçük bir değişiklik bile olanağımız dışında kalabilir. Yanıtlamamız gerekli ikinci soru da işte budur. Bu ve benzeri nedenlerden dış ülkelerde de gördüğümüz gibi bilgisayar desteği ile tasarım yapmak isteyen kuruluşlar çoğunlukla kendi düzenlerini kendileri gerçekleştirmektedir. Örneğin, TBTA' da böyle bir yol izlemektedir [35].

Sonuç olarak,şunu diyebiliriz, Türkiye'nin içinde bulunduğu koşulları göz önüne getirdiğimizde tasarımlarında bilgisayarı kullanmaları zorunlu olan kuruluşların bir ya da bir kaçını birden yine Türkiye içindeki üniversitelerle işbirliği yaparak kendi sorunlarını çözecek düzenleri yine kendileri geliştirmelidir.

KAYNAKLAR

- fil Gei/hun, Y., "Çizge Kuramının Temelleri-I" Elektrik Mühendisliği, No:205-206, s.47-58, 1974
[2] Ceyhun, Y., "Çizge Kuramının Temelleri-II" Elektrik Mühendisliği, No:207, s.141-145, 1974

- [3] Mayeda, VI., "Graph Theory", Wiley Interscience, 1972
[4] Calahan, D.A., "Linear Network Analysis and Realization Digital Computer Programs: An Instruction Manual", Univ. of Illinois Bulletin, Vol.62, No:58

6. teknik kongre

- [5] Hüseyin, Ö., Y.Ceyhun, S.Penbeci, K.Abdullah: "Electrical System Analysis", Ders Notları, ODTÜ 1974
- [6] Brooks, N.G., H.S.Long: "A Program for Computing the Transient Katponse of Transistor Svitching Circuits-PE TAP", IBM Development Lab., Pough-keepsie, N.Y. Tech.Rept. 00.700, 1959
- [7] Branin, F.H.Jr.,: "DC and Transient Analysis of Networks Using a Digital Computer", İRE Int.Conv.Ref., Part 2., s.236-256, Mart 1962
- [8] Malmberg, A.F., F.L.Cornwell ve F.N.Hofer: "NET-1 Network Analysis Program", Los Alamos Scientific Lab., Los Alamos, N.Mex.Rept. LA-3119, 1964
- [9] "1620 Electronic Circuit Analysis Program ECAP", IBM Corp., Data Processing-Div., White Plains, N.Y., Application Program 1620-EE-02X.
- [10] Kron, C "A Set of Principles to Interconnect the Solutions of Physical Systems", J.Appl. Phys. Vol.24, s.965-980, 1953
- [11] Forsythe, G.E., C.B.Moler, "Computer Solution of Linear Algebraic Systems", Prentice-Hall, 1967
- [12] Proc. Sparse Matrix Symp., IBM Rep. RA-1, Mart 1969
- [13] Kuo, F.F., W.G.Magnuson Jr. : "Computer Oriented Circuit Design", Prentice-Hall 1969, Bölüm 4
- [14] Pottle, C., "Comprehensive Active Network Analysis by Digital Computer: A State Space Approach" Research Rep. EERL 59, Cornell University, Ithaca, New York, Electrical Eng. Research Lab.
- [15] Milliman, L.D., tt.A.Massena ve R.H.Dickhaut: "CIRCUS-A Digital Computer Program for Transient Analysis of Electronic Circuits-User's Guide", The Boeing Comp., Seattle, Wash., Harry Diamond Lab.Rept. AD-346-1, Ocak, 1967
- [16] Mathers, H.W., S.P.Stdore ve J.R.Sents, "Automated Digital Computer Program for Determining Responses of Electronic Circuits to Transient Nuclear Radiation (SCEPTRE)", Vol.1, IBM Space Guidance Center, Owego, N.Y., IBM File, 66-928-611, Şubat 1967
- [17] "Automated Digital Computer Program for Determining Responses of Electronic Systems to Transient Nuclear Radiation (PREDICT)", Vol.II, IBM Space Guidance Center, Owego, N.Y.IBM File 64-521-5, Temmuz 1964
- [18] Faik, H., "Computer Programs for Circuit Design", Electro-Technology, Vol.77, s.54-57, Haziran 1966
- [19] Dertouzos, M.L., "CIRCAL: On line Circuit Design", Proc. IEEE Vol.55, s.637-654 Mayıs 1967
- [20] Katzenelson, J., "AEDNET: A simulator for nonlinear networks", Proc. IEEE, Vol.54, s.1536-1552, Kasım 1966
- [21] Chua, L.O., P.Medlock: "MECA-Multivalued Electronic Circuit Analysis Program", Purdue Univ. Tech. Rept. TR-EE-69-7 School of Elect. Eng. Nisan 1969
- [22] Olsztyn, J.J., T.L.Theodoroff: "GMR DYANA Programming Manuals I and II", General Motors Research Lab., VJarren, Mich., 1959
- [23] Fenves, S.J., "STRESS (Structural Engineering System Solver) a Computer Programming System for Structural Engineering Problems", M17, Cambridge Mass., Tech.Rept.T63-2, Haziran 1963
- [24] Ceyhun, Y., "Sensitivity Considerations in Electrical Networks", Ders Notları, ODTÜ Elektirik Müh.Böl., 1972
- [25] Fletcher, R., M.J.D.Powell: "A Rapidly Convergent Descent Method for Minimization", Brit. Computer J., Cilt 6 -s.163-168, 1963
- [26] Ceyhun, Y., "Sensitivity Invariants of Certain Class of Networks", Electr. Letts. Vol.7, No.3, s.85-86, 1971
- [27] Kançık, H., MS Tezi, ODTÜ Elk.Müh.Böl.1975
- [28] Ceyhun, Y., "Introduction to Equivalent Network Theory", Ders Notları, ODTÜ, Elk.Müh.Böl., 1972
- [29] Ceyhun, Y., A.Esen: "Computer Aided Network Synthesis for Educational Purposes", 4th Iranian Conf. on Elect. Eng. Pahlavi Univ., Shiraz, İnan, 12-16 Mayıs 1974
- »
- [30] Kuo, F.F., W.G.Magnuson Jr: "Computer Oriented Circuit Design", Prentice-Hall, 1969 Bölüm 10
- [31] Zane, R., D.A.tfilber: "PUZZLE: A Program for Computer Aided Design of Printed Circuit Art-vork", Lavrence Radiation Lab., UCRL 17814, Eylül 1967
- [32] Fisk, C.J., D.L.Isett: "ACCEL, Automated Circuit Card Etching Layout", Sandia Lab., Tech. Memo SC-TM-65-544, Ekim 1965
- [33] Mays, C.H., "Computer Aided Design for Large Scale Integration", Int.Solid State Circuits Conf. Digest, Şubat 1967
- [34] Frank, H., I.T.Frisch: "Communication, Transmission and Transportation Networks", Addison-Wesley, 1971
- «
- [35] Tüyeni, E., Y.Ceyhun: "Basamaklı Edilgin RLC Devrelerinin Çözülmesinde Yeni Olanaklar", EMO, Elektrik Mühendisliği, s.452-466, No.212, 1974