

22176

KARADENİZ TEKNİK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ELEKTRONİK MÜHENDİSLİĞİ PROGRAMI

GÜÇ ELEKTRONİĞİNDE KULLANILAN DEVRELERİN

BİLGİSAYAR YARDIMI İLE ÇÖZÜMLENMESİ

Elektronik müh. Melike Özlem UZUN

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde

"Elektronik Yüksek Mühendisi"

Ünvanının verilmesi için kabul edilen tezdır.

Tezin Enstitüye verildiği tarih: 21 Aralık 1992

Tezin sözlü savunma tarihi : 12 Şubat 1993

Tez danışmanı : Doç.Dr. Güven ÖNBİLGİN

Jüri üyesi : Doç.Dr. Osman TONYALI

Jüri üyesi : Doç.Dr. Cemil GÜRÜNLÜ

Enstitü Müdürü : Prof.Dr. Temel SAVASKAN

ARALIK - 1992

TRABZON

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

ÖNSÖZ

Bu tez çalışması, hem elektrik hem de elektronik mühendisliği kapsamına girdiğinden, her iki alanda çalışma yapan arkadaşlara katkıları olacağını umuyorum. Bu konudaki çalışmalarımı yönlendiren danışmanım Doç.Dr. Güven ÖNBİLGİN'e teşekkür ederim.

Melike Özlem UZUN

Trabzon, Aralık-1992

İÇİNDEKİLER:

ÖNSÖZ	II
ÖZET	VII
SUMMARY	VIII
SEMBOLLER	IX
BÖLÜM 1 GİRİŞ	1
1.1.LAPLACE DÖNÜŞÜMÜ-BİRİM BASAMAK İSLEVİ	1
1.2.FOURIER SERİSİ YAKLAŞIMI	3
1.3.DURUM DEĞİŞKENLERİ YAKLAŞIMI	3
1.3.1.DURUM DENKLEMLERİNİN ELDE EDİLMESİ	4
1.3.2.DURUM DENKLEMLERİNİN SAYISAL YÖNTEMLE ÇÖZÜMÜ..	5
1.3.3.YÖNTEMİN UYGULANMASI	6
1.4.ÖNEMSİZ HARMONİKLERLE ÇÖZÜMLEME YAKLAŞIMI	7
1.5.SCR DEVRELERİNİN SAYISAL HESABINDA PEKİŞTİRİLMİŞ ALGORİTMALARIN UYGULANMASI	7
1.6.ORTALAMA ALMALI "AVERAGING" YAKLAŞIM	9
1.7.KARMAŞIK DEĞİŞKEN ÇÖZÜMLEME YAKLAŞIMI	10

1.8.UZAY (PARK) VEKTÖRÜ İLE ÇÖZÜMLEME YAKLAŞIMI	10
1.9.YARI İLETKENLERİN İDEAL ANAHTAR GİBİ DÜŞÜNÜLMESİ. 11	
BÖLÜM 2 R,I,E ELEMANLARI İÇEREN DEVRELER İÇİN BİLGİ- SAYARLA BİR ÇÖZÜMLEME MODELİ	12
2.1.RNET PROGRAMI	12
2.2.BİR OMİK DEVRENİN ÇÖZÜMLENMESİ	14
2.3.DEĞERLENDİRME	16
2.4.RNET PROGRAMININ YAPISAL ÖZELLİKLERİ	16
2.4.1.PROGRAM GİRİŞİNİN OLUŞTURULMASI	17
BÖLÜM 3 L,C ELEMANLI DEVRELERİN MODELLENMESİ VE BİLGİ- SAYARLA BİR ÇÖZÜMLEME MODELİ	19
3.1.COMPANION MODEL	19
3.1.1.BACKWARD EULER YÖNTEMİ	19
3.1.2.SİĞA İÇİN COMPANION MODEL	20
3.1.3.ENDÜKTANS İÇİN COMPANION MODEL	21
3.2.ALÇAK GEÇİREN SÜZGEÇ İÇİN COMPANION MODEL	23
3.3.BENZER BİR DEVRE İÇİN COMPANION MODEL	25
3.4.ENDÜKTANS VE SİĞA İÇEREN PROGRAM	26
BÖLÜM 4 DİYOT İÇEREN DEVRELERİN MODELLENMESİ VE BİGİ- SAYARLA BİR ÇÖZÜMLEME MODELİ	28
4.1.DİYOT MODELLEMESİ	28

4.1.1. ANAHTAR YAKLAŞIMI İLE MODELLEME	28
4.1.2. BAĞIMSIZ KAYNAK YAKLAŞIMI İLE MODELLEME	29
4.2. SİNÜZOIDAL GİRİŞ OLUŞTURULMASI	31
4.3. DİYOT İÇEREN PROGRAM	32
4.4. PROGRAM ÇIKIŞI	33
4.4.1. GRAFİK ÇIKIŞ	33
4.5. AKIŞ DİYAGRAMI	35
BÖLÜM 5 UYGULAMA	36
5.1. RNET PROGRAMININ UYGULAMALARI	36
5.1.1. KARIŞIK KAYNAKLI UYGULAMA	36
5.1.2. YALNIZ AKIM KAYNAKLARI İÇEREN UYGULAMA	38
5.1.3. YALNIZ GERİLİM KAYNAKLARI İÇEREN UYGULAMA ...	40
5.2. COMPANION MODELLEME İLE GENİŞLETİLMİŞ PROGRAMIN UYGULAMALARI	42
5.2.1. SİĞA YÜKLÜ UYGULAMA	42
5.2.2. KARMASIK YÜKLÜ UYGULAMA	43
5.3. DİYOT İÇEREN PROGRAMIN UYGULAMALARI	45
5.3.1. BİR YARIM DALGA DOĞRULTUCU DEVRE İÇİN UYGULAMA	45
5.3.2. İNDÜKTİF YÜKLÜ BİR YARIM DALGA DOĞRULTUCU DEVRE İÇİN UYGULAMA	47
5.3.3. BİR TAM DALGA DOĞRULTUCU DEVRE İÇİN UYGULAMA.	49
BÖLÜM 6 TARTIŞMA	51
KAYNAKLAR	53

EKLER	55
EK-A:R,E,I ELEMANLARI İÇEREN DEVRELER İÇİN RNET PROGRAMI	55
EK-B:L,C ELEMANLARIDA İÇEREN DEVRELER İÇİN PROGRAM....	70
EK-C:DIYOT-TRİSTOR ELEMANLARIDA İÇEREN DEVRELER İÇİN PROGRAM	87
ÖZGEÇMİŞ	109



OZET

Elektrik - Elektronik devrelerinde birçok çözümlenme yöntemi geliştirilmiştir. Her birinin bir diğerine üstünlükleri vardır. Bu çalışmada yapılan işe, INCIDENCE MATRİS kurulması yöntemini içeren, çözümlenme yönteminin, bilgisayar yardımıyla kurulması ve geliştirilmesidir. Bu yöntem önce en basit devre üzerine kurulmuş, daha sonra adım adım geliştirilmeye geçilmiştir. Geliştirilmeye geçilirken şimdiye kadar kurulan bilgisayar destekli çalışmaların dışında bir yol izlenmeye özen gösterilmiştir.

Çalışmaya yalnız direnç, akım kaynağı ve gerilim kaynağı içeren RNET bilgisayar programının geliştirilmesi ile başlanmıştır. Daha sonra adım adım, indüktans, sığa, diyot içeren daha karmaşık devrelerin çözümüne geçilmiş ve çeşitli programlara üstünlükleri gözlenmiştir.

SUMMARY

Various analysis techniques have been developed for electrical and electronical circuit. Each technique has advantages over the others. A computer analysis programme is developed in this thesis. The program includes incidence matrix analysis technique together with L,C and switching semiconductor elements.

First, a computer program known as RNET is generalized. RNET includes only resistors, voltage sources and current sources.

Next, the companion model concept is employed to be included in the RNET as sources and resistors, but time dependent.

The last step in the program development has been to realise the model for the switching semiconductor elements, namely the diode and the thyristor.

The program developed has been described and applied to several practical cases. Consequently discussions of results and proposals for further work has been given.

SEMBOLLER

$V_{OUT}(t)$	-Zamana göre deęişen çıkış gerilimi
w	-Açıçal hız
f	-Frekans
$u(t)$	-Birim basamak fonksiyonu
R	-Devredeki direnç
L	-Devredeki endüktans
C	-Devredeki sığa
D	-Devredeki diyot
T	-Devredeki tristör
$L[...]$	-Laplace dönüşümü
E, V	-Devredeki bağımsız gerilim kaynağı
I, J	-Devredeki bağımsız akım kaynağı
X	-Durum deęişkenleri vektörü
U	-Kaynaklara ilişkin vektör
Y	-Çıkış deęişkenleri vektörü
A, B, C, D	-Katsayı matrisleri
A ucu	-Anot ucu
K ucu	-Katot ucu
G ucu	-Kapı ucu
S_I, S_V, S_G, S_{PC}	-İdeal anahtarlar
$V_{eş}$	-Eşdeğer gerilim kaynağı
$R_{eş}$	-Eşdeğer direnç
A_a	-Incidence matris
A	-İndirgenmiş incidence matris
N sayısı	-Devredeki toplam düğüm sayısı
B sayısı	-Devredeki toplam dal sayısı
V sayısı	-Devredeki toplam gerilim kaynağı sayısı
C sayısı	-Devredeki toplam akım kaynağı sayısı
R sayısı	-Devredeki toplam direnç sayısı

GR	-Referans düğüm numarası
x	-Dizge durumu
v (t)	-Dizge girişı
x' (t)	-Dizge durumunun türevi
DT	-Küçük bir zaman artımı
I _L , V _L	-Endüktans akım ve gerilimi
I _C , V _C	-Sığa akım ve gerilimi
R _{comp}	-Companion modeldeki eşdeğer direnç
I _{comp}	-Companion modeldeki eşdeğer akım kaynağı



BOLÜM 1

GİRİŞ

Güç elektroniği devreleri aktif ve pasif birçok eleman içerir. Bu devrelerin çözülmesi, sistemi tanımlayan eşitliklerin karmaşıklığı ve doğrusal olmamasında dolayı oldukça güçtür. Bu amaçla bir çok yöntem geliştirilmiştir. Bu yöntemlerin üstünlükleri tartışılabilir.

1.1.LAPLACE DÖNÜŞÜMÜ-BİRİM BASAMAK İŞLEVI YAKLAŞIMI:

Birçok yarı iletken devre Laplace dönüşümlerinin doğrudan uygulanmasıyla çözümlenmiştir [1]. Önce $V_{out}(t)$ yük gerilimi belirlenir ve bilinen Laplace dönüşüm bağıntıları kullanılarak $V_{out}(t)$ 'nin Laplace dönüşümü elde edilir. $V_{out}(s)$ yardımıyla $I_{out}(s)$ yük akımı eşitliği hazırlanır ve ters Laplace dönüşümü uygulanarak $i_{out}(t)$ yük akımına ulaşılır.

Takeuchi ; Laplace dönüşümlerinin bir doğrudan uygulaması ile güç elektroniği devrelerini çözümlenmiştir [2]. Bu yöntemin bir örneği olarak, bir tam dalga doğrultucu devre ve bir SCR beslemeli indüksiyon motor sistemini çözümlenmiştir.

Uygulama olarak, seri LR yüküne sahip ve $E_m \sin(\omega t)$ sinüzoidal kaynağı olan bir tamdalga doğrultucu için :

$$V_{out}(t) = \sum_{n=0}^{\infty} \left[u\left(t - \frac{n\pi}{\omega}\right) - u\left(t - \frac{(n+1)\pi}{\omega}\right) \right] \cdot E_m \sin(\omega t - n\pi) \quad \dots (1.1)$$

Burada $u(t)$ birim basamak işlevidir.

$\sin(\omega t - n\pi) = -\sin(\omega t - n\pi - \pi)$ eşitliğini (1.1)'e uygularsak :

$$V_{out}(t) = \sum_{n=0}^{\infty} E_m \left[u\left(t - \frac{n\pi}{\omega}\right) \cdot \sin\left[\omega\left(t - \frac{n\pi}{\omega}\right)\right] + u\left(t - \frac{(n+1)\pi}{\omega}\right) \sin\left[\omega\left(t - \frac{(n+1)\pi}{\omega}\right)\right] \right] \dots (1.2)$$

Burada:

$$L[u(t) \cdot \sin(at)] = \frac{a}{a^2 + s^2} \dots (1.3)$$

$$L[u(t-a) \cdot f(t-a)] = e^{-as} \cdot L[f(t)] \dots (1.4)$$

Laplace bağıntıları kullanılırsa, eşitlik (1.2) için:

$$V_{out}(s) = \sum_{n=0}^{\infty} \frac{w \cdot E_m}{s^2 + w^2} \left[e^{-(n\pi/w)s} + e^{-[(n+1)\pi/w]s} \right] \dots (1.5)$$

yazılabilir. Ayrıca devreden $V_{out}(s)$ için :

$$V_{out}(s) = (R + sL) \cdot I_{out}(s) \text{ yazılabilir.}$$

Buradan :

$$I_{out}(s) = \frac{1}{L} \cdot \frac{E_0(s)}{s + R/L} = \frac{E_m}{L} \sum_{n=0}^{\infty} \frac{w}{(s + R/L) \cdot (s^2 + w^2)} \cdot \left[e^{-(n\pi/w)s} + e^{-[(n+1)\pi/w]s} \right] \dots (1.6)$$

Eşitlik (1.4) 'deki bağıntı kullanılarak:

$$L^{-1} \left[\frac{w}{(s + R/L) \cdot (s^2 + w^2)} \right] = w \cdot (L^2/Z^2) \cdot e^{-(R/L)t} \cdot t + (L/Z) \cdot \sin(\omega t + \theta)$$

Burada : $Z^2 = R^2 + w^2 \cdot L^2$ ve $\theta = \tan^{-1}(w \cdot L/R)$ dir.

eşitlik (1.6)'nın ters dönüşümü yapılırsa

$$i_{out}(t) = (e_m/L) \cdot \sum_{n=0}^{\infty} \{ u(t - (n\pi/w)) \cdot [(w \cdot L^2/Z^2) \cdot e^{-[(R/L) \cdot (t - n\pi)/w]} + (L/Z) \cdot \sin(wt - n\pi - \theta)] + u(t - [(n+1) \cdot \pi/w] \cdot [(w \cdot L^2/Z^2) \cdot e^{-[(R/L) \cdot (t - [(n+1) \cdot \pi]/w)} - (L/Z) \cdot \sin(wt - n\pi - \theta)] \} \dots (1.7)$$

elde edilmiş olur.

1.2. FOURIER SERİSİ YAKLAŞIMI :

Şayet yük doğrusal düşünülebiliyorsa ve dönüştürücü tarafından yüke uygulanan gerilimde biliniyorsa ,yük akımını belirlemede Fourier serileri kullanılmıştır [1]. Seriler içindeki her bir özel salınımdaki yükün eşdeğer devresi ve yük gerilimini temsil eden Fourier serileri bulunarak çözümlenmeye başlanmıştır. Bu yöntem gerilim şeklinde bir belirsizlik derecesi olan devrelere uygulanmıştır.

Jacovides ; SCR beslemeli indüksiyon motorun çözümü için Fourier serileri yöntemini kullanmıştır [2]. Burada anah-tarlama devresini doğrusal gibi düşünmüştür. Dönüştürücü tarafından yüke uygulanan gerilimi tahmin etmiş, süperpozisyon ilkelerini kullanarak yük akımlarını belirlemiştir.

1.3. DURUM DEĞİŞKENLERİ YAKLAŞIMI :

Durum denklemi yaklaşımı ile geliştirilen yöntemler daha sistematiktir [1][3].

Nayak ve Hoft 'da ; durum değişkeni yaklaşımını kullanmıştır [2]. Algoritmalarını; tek faz indüksiyon motoru ve SCR beslemeli 3-faz indüksiyon motora uygulamışlardır.

1.3.1 DURUM DENKLEMLERİNİN ELDE EDİLMESİ :

Akım ve gerilim kaynakları, direnç, sığa ve endüktanslardan oluşan bir elektrik devresinin matematiksel modellenmesi "durum denklemleri" yöntemiyle gerçekleştirilir. Bu yöntemde devrenin uygun ağacı belirlenir. Bunun için bütün bağımsız gerilim kaynakları dal, bütün bağımsız akım kaynakları giriş, sığa elemanları olabildiğince dal, endüktans elemanları olabildiğince giriş ve direnç elemanları da uygun ağaca göre dal veya giriş olarak seçilirler.

Bu model kısaca :

$$X' = A * X + B * U$$

$$Y = C * X + D * U \quad \dots (1.8)$$

Olarak gösterilirler. Burada :

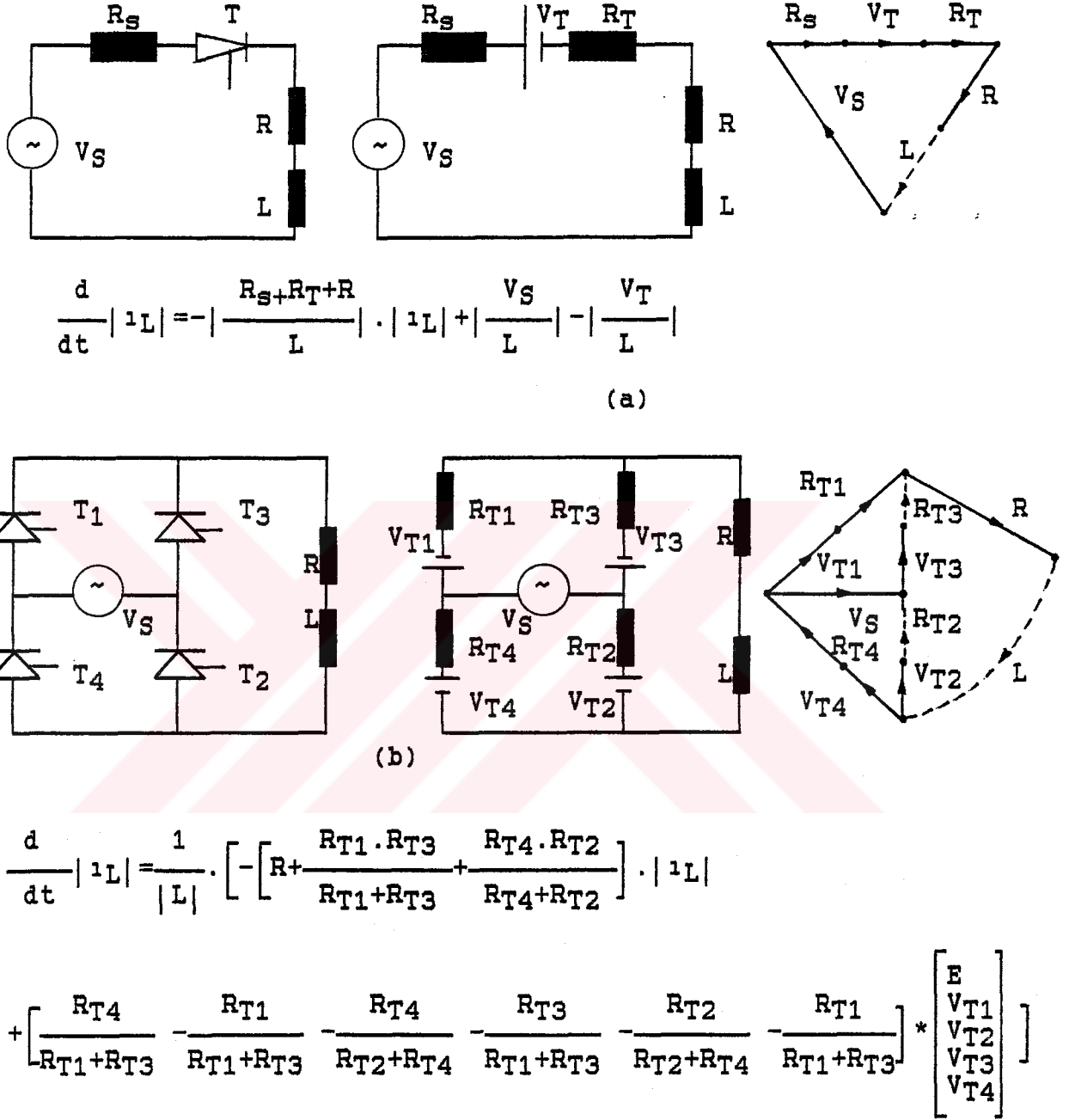
X: Durum değişkenleri vektörü

U: Devredeki kaynaklara ilişkin vektör.

Y: Çıkış değişkenleri vektörüdür.

A, B, C, D matrisleri ise devre topolojisi ile ilişkili akım - gerilim bağıntılarından elde edilen katsayı matrisleri olarak tanımlanırlar.

Bu katsayı matrisleri basit yapıdaki devrelerde elle elde edilebilirler Şekil 1.1.a'da tek yollu (yarım dalga) bir doğrultucu devre ile bir köprü (tam dalga) doğrultucu devrenin bu yoldan elde edilmiş durum denklemleri modelleri görülmektedir.



Sekil 1.1 Doğrultucu devrelerin durum modelleri ve denklemleri.

(a) Yarı dalga doğrultucu devre.

(b) Tam dalga doğrultucu devre.

1.3.2 DURUM DENKLEMLERİNİN SAYISAL YÖNTEMLE ÇÖZÜMÜ :

Bağıntı (1.8)'den de görüleceği üzere durum denklemi modelinin belli bir zaman süresinde çözümü bir diferansiyel

denklem takımının sayısal çözümünü gerektirir. Bu çözüm 4.dereceden Runge Kutta yada Runge Kutta - Merson yöntemleri ile elde edilebilir. Ancak programlama ve biçimlendirme açısından kolaylık getirebileceği düşünülen bir yaklaşımda, durum denklemi modelinin diferansiyel denklem takımını bir fark denklemi takımına dönüştürmek ve hesap yapılan süre boyunca herbir küçük hesap adımından diğerine bu ardışıl işlemlerle fark denklemini kullanmaktır. $X'=A*X+B*U$ diferansiyel denklem takımı bu açıdan yeniden biçimlendirildiğinde :

$$X(n+1)=X(n)+Dt. [A(n).X(n)+B(n).U(n)] \dots (1.9)$$

Bağıntısı elde edilir.

Ayrıca diferansiyel denklemlerin çözümü sırasında hesap adımının denetim altına alınmasında A matrisinin en büyük özdeğerinin sürekli izlenmesiyle gerçekleştirilebilmiştir.

1.3.3 YONTEMİN UYGULAMASI :

Bilgisayarda durum denklemi modeli ile güç elektroniği devrelerinin çözümünü gerçekleştirecek bir programın ana hatları aşağıda verilmiştir:

1- Devreye ilişkin topolojik verilerin verilmesi, devre elemanlarının değerleri, diyot ve tristorlerin iletim ve tıkama yönünde direnç değerleri, kırılma gerilimleri, tutma akımı değerleri, devredeki tristörlerin tetikleme darbelerine ilişkin zamanlama bilgileri, çözümün istendiği süre ve bu sürede çözümden kullanılacak temel zaman aralığının verilmesi.

2- Devre topolojik bilgilerine dayalı A,B,C,D matris elemanlarının oluşturulması.

3- 1 'de sözü edilen yöntemlerden biri kullanılarak Dt hesap adımları ile her hesap adımında devredeki diyot ve tristorlerin modelinin o andaki akım, gerilim ve darbe durumlarına göre A,B,C,D katsayı matrislerinin değerleri ile sayısal çözüm gerçekleşmesi.

Ancak fark denklemleri yaklaşımında karmaşık devrelerde daha küçük hesap adımları kullanma zorunluluğu, bu yöntemin programlama kolaylığını dengelemektedir.

1.4. ÖNEMSİZ HARMONİKLERLE ÇÖZÜMLEME YAKLAŞIMI :

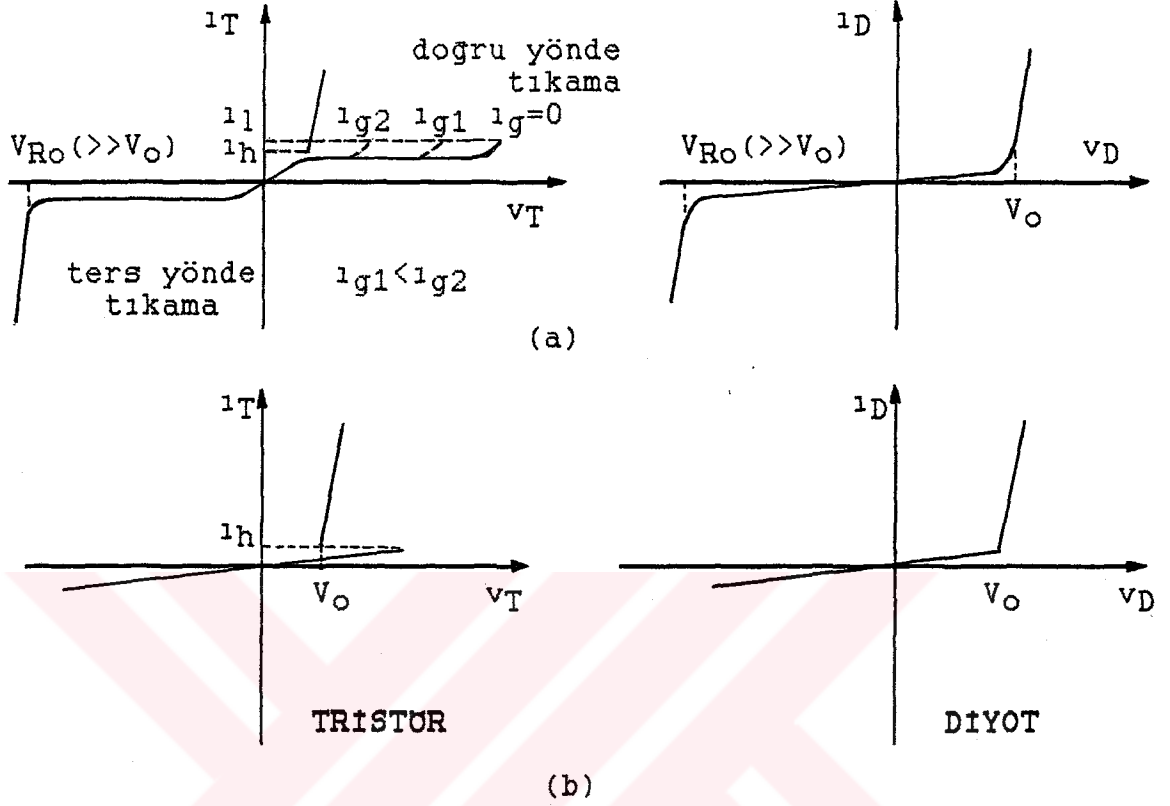
Bazı anahtar devrelerinde, dönüştürücüler tarafında üretilen bütün önemsiz harmoniklerle, yeterli kesinlik kazanmış ve sadeleştirilmiş bir çözüm elde edilebilmiştir [1].

Çıkış dalga şekline yayılan Fourier serilerinin bir terimini alıkoyarak bu çözümleme yapılmıştır. Harmoniklerin büyük bir kısmı yükü etkilemediğinde bu sadeleştirme kullanışlı olmuştur.

Krause ve Lipo 'da aynı yöntemi kullanmışlar ve çözümle- rinde yeterli bir doğruluk elde etmişlerdir [2].

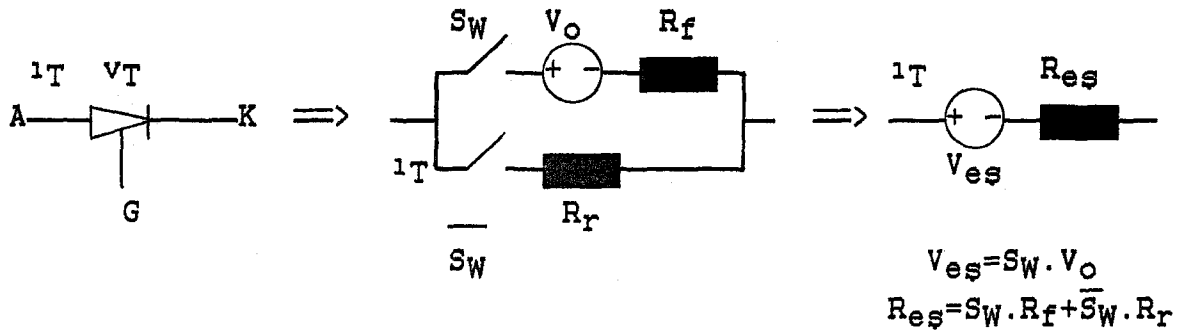
1.5. SCR DEVRELERİNİN SAYISAL HESABINDA PEKİŞTİRİLMİŞ ALGORİTMALARIN UYGULAMASI :

Tristör ve diyotların çalışmalarını simgeleyen v-i özegrileri Şekil 1.2.a'da verilmiştir. Bu özegrileri doğrusal parçalara bölerek Şekil 1.2.b'deki yaklaşım özegrileri elde etmek ve modellerini oluşturmak mümkün olmaktadır [1][3][4].



Sekil 1.2-Tristor ve Diyot özgeçirilerine yaklaşım.
 (a) Tristor ve diyot v-i özgeçirileri.
 (b) Tristor ve diyot yaklaşık özgeçirileri.

Tristorün doğrusal parçalı özgeçirileri yaklaşımı şekil.3'de verilen eşdeğer devreyi oluşturmanın temelini oluşturmuştur. Bu eşdeğer devrede ana işlev SI, SV, SG, SPC, SW, $\bar{S}W$ mantık anahtarlarının, bir devrenin çözümünde herhangi bir anda nasıl değerlendirileceğinde kendini gösterir.



Sekil 1.3-Tristor eşdeğer devresi (yaklaşık model).

Bu modelde anahtarların deęerleri şöyle saptanabilir:

$SI=1$ $i_T > i_h$ için $SI=0$ $i < i_R$ için.

$SV=1$ $v_T > v_0$ için $SV=0$ $v_T < v_0$ için.

$SG=1$ kapıda darbe olması durumunda, yoksa=0.

$SPC=1$ bir önceki hesap adımında tristör iletimde ise, yoksa = 0.

Tristör için $SW=(SV+SI).(SPC+SG)$, diyot için $SW=(SV+SI)$ olarak alınmaktadır. Böylece belli bir zaman aralığı boyunca tristör ya da diyot eşdeęer devresi elemanlarının deęerleri, o zaman adımındaki akım, gerilim, (tristör ise) kapıda tetikleme darbesinin var olup olmadığı ve bir önceki zaman aralığında tristör ya da diyot'un iletimde olup olmadığına baęlı olarak saptanabilir. Ayrıca bu modelde V_0 doğru yönde kırılım geriliminin gözardı edilmesi durumunda diyot ya da tristörün iletim yönünde çok küçük deęerli bir direnç ve tıkama yönünde çok büyük deęerli bir direnç ile modellendięi ortaya çıkar.

1.6 ORTALAMA ALMALI "AVERAGING" YAKLAŞIM :

Bu yöntemde; anahtarlama devrelerinin küçük sinyal tepkeleri, alçak salınım çözümlenmeleri için kullanışlıdır. [1] Bu yöntem genellikle bir anahtarlama salınımı ile çalışan anahtarlama cihazlarına uygulanabilmektedir. Şimdiye kadar bu yöntem dc-dc çeviricilerinde etkili bir şekilde kullanılmıştır. Fakat bununla sınırlı bir yöntem değildir. Ortalama alma yönteminin sonuçlandırdığı dizge, doğrusal veya doğrusal olmayan bir dizge olabilir. Bu yöntemi temel olarak alan çözümlemede, standard uygulanabilir yöntemlerle ilerlenebilir. (Laplace dönüşümü, durum deęişkenleri, vs. gibi) [1].

Wester ve Middlebrook ile Iliddlebrook ve Cuk, bu yöntemi kullanmışlardır [2].

1.7 KARMAŞIK DEĞİŞKEN ÇÖZÜMLEME YAKLAŞIMI :

Dizgede var olan simetriden ötürü 3-faz indüksiyon motorların çözümlemesini basitleştirmek için bu yöntem uygulanmıştır. Hem birinci hem ikinci derece doğrusal dizgeye sorun olan kaynak dönüştürücü gerilimini azaltmıştır. Buna yakın çözümler ya elle veya bir basit bilgisayar hesabıyla çözümlenebilmektedir [1].

Novotny ; dizgenin simetrisinden dolayı bir çevirici beslemeli 3-faz indüksiyon motorun çözümü için bu yöntemi kullanmıştır [2].

1.8 UZAY (PARK) VEKTÖRÜYLE ÇÖZÜMLEME YAKLAŞIMI :

Dönüştürücülerin dönen simetrisinden ve ac makinaların onlara olan bağlantısından dolayı park vektörler verilmiştir. Gerilim ve akımlar park vektörlerle gösterilmiştir. Bu yöntemle özel diferansiyel eşitlikler doğrudan doğruya hazırlanabilir [1][5]. Park vektör:

$$f(t) = \frac{2}{3} [f_a(t) + a \cdot f_b(t) + a^2 \cdot f_c(t)] \dots (1.10)$$

Şeklinde tanımlanmıştır. Burada; $a = \exp(j \cdot 2\pi/3) \dots (1.11)$

f_a, f_b, f_c ; bir 3-faz dizgenin ani faz değerleridir.

Ters Park ilişkileri:

$$f_a(t) = \text{Re}[f(t)] + f_0(t)$$

$$f_b(t) = \text{Re}[a^2 \cdot f(t)] + f_0(t)$$

$$f_c(t) = \text{Re}[a \cdot f(t)] + f_0(t) \dots (1.12) \text{ Şeklinde verilirler.}$$

Burada $f_0(t)$ sıfır sıra miktarıdır.

Ve; $f_0(t) = (1/3) \cdot [f_a(t) + f_b(t) + f_c(t)] \dots (1.13)$
şeklinde verilmiştir.

Böylece Park vektör dönüşümü 3-faz dizgeyi, bir tek faz karmaşık değişken dizgeye indirgemıştır.

Jordan ve Devan ; park vektör çözüm yöntemini, 3-faz köprü çeviricilerinin çözümünde kullanmışlardır. Kalman; aynı yöntemi 3-faz line-commutated köprü dönüştürücülerinin çözümünde kullanmıştır [2].

1.9.YARI İLETKENLERİN İDEAL ANAHTAR GİBİ DÜŞÜNÜLMESİ:

Martl ve Winograd çalışmalarında yarı iletken cihazları ideal anahtar gibi düşünmüşlerdir [2][6].

Devredeki diyot elemanı iletme geçtiğinde, çok küçük bir dirençle seri bağlı ideal bir anahtar (yani yaklaşıklıkla kısa devre bir ideal anahtar) olarak düşünülmüştür. Tıkama konumunda ise çok büyük bir direnç ve (açık konumda) seri bağlı ideal bir anahtar (yani yaklaşıklıkla açık devre bir ideal anahtar) olarak düşünülmüştür. tristördede aynı benzetim kullanılmıştır [1][7].

Bütün bu özetle verilen çözüm yöntemlerinden başka; anahtarlama işlev çözümü[1], Veri örnekleme modellemesi[1], Boolean cebiri[1] kullanılarak çözüm,...v.b. gibi daha birçok çözüm yöntemi vardır. Bunların üstünlükleri kullanım alanına göre farklı olup tartışılabilir.

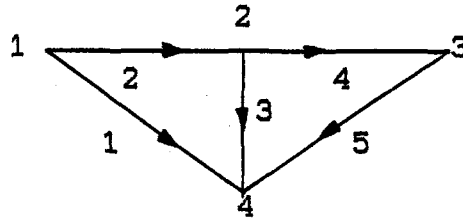
BÖLÜM 2

R,I,E ELEMANLARI İÇEREN DEVRELER İÇİN BİLGİSAYARLA BİR ÇÖZÜMLEME MODELİ

2.1.RNET PROGRAMI :

Bu program girişine verilen devreye ilişkin incidence matrisi oluşturarak çözüme giden bir yapıya sahiptir [8][9].

Incidence matris: Bir devreye ilişkin çizilen ağaçtaki her düğüme ilişkin Kirchhoff'un akım yasası yazılır. Bu denklemler matris formuna getirildiğinde incidence matris oluşur.Örneğin; Şekil 2.1.'deki ağaç için bu denilenler yapılırsa:



$$J = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \end{bmatrix}$$

J ;Dal akım vektörü

Şekil 2.1 Bir devre ağacı ve dal akım vektörü.

1. Düğüm için ; $J_1+J_2=0$
2. Düğüm için ; $-J_2+J_3+J_4=0$
3. Düğüm için ; $-J_4+J_5=0$
4. Düğüm için ; $-J_1-J_3-J_5=0$

Buradan:

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & 0 & -1 \end{bmatrix}}_{A_a} * \begin{bmatrix} J1 \\ J2 \\ J3 \\ J4 \\ J5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Burada oluşan A_a matrisi devrenin incidence matrisidir. Bu devrede 4 düğümü referans düğüm alınıp A_a matrisinden çıkarılırsa, indirgenmiş incidence matris denilen A matrisi oluşur.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{matrix} \rightarrow 1 \\ \rightarrow 2 \\ \rightarrow 3 \end{matrix}$$

Dallar 1 2 3 4 5

Buradanda şu yazılabilir:

$$A * J = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} * \begin{bmatrix} J1 \\ J2 \\ J3 \\ J4 \\ J5 \end{bmatrix} = 0 \quad \dots (2.1)$$

Düğüm gerilimlerini içeren e vektörü ile dal gerilimlerini içeren v vektörü arasındaki bağıntı:

$$v = A^T \cdot e \quad \dots (2.2) \quad \text{şeklinde verilmiştir.}$$

Aynı devre için:

$$v = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} * \begin{bmatrix} e1 \\ e2 \\ e3 \end{bmatrix}$$

Yani: $v1=e1$
 $v2=e1-e2$
 $v3=e2$
 $v4=e2-e3$ ve $v5=e3$ yazılabilir.

2.2. BİR OMİK DEVRENİN ÇÖZÜMLENMESİ [9] :

$$A \cdot J = 0, \quad v = A^T \cdot e \quad \text{olduğuna göre} \quad v^T \cdot J = 0 \quad \dots (2.3)$$

$$J = G \cdot v + J_S - G \cdot v_S \quad \dots (2.4)$$

Burada :

G : dal kondüktans matrisi

J_S : bağımsız akım kaynakları vektörü

v_S : bağımsız gerilim kaynakları vektörü olarak verilmiştir.

$$A \cdot G \cdot A^T \cdot e + A \cdot J_S - A \cdot G \cdot v_S = 0 \quad \text{veya}$$

$$A \cdot G \cdot A^T \cdot e = A \cdot G \cdot v_S - A \cdot J_S \quad \dots (2.5)$$

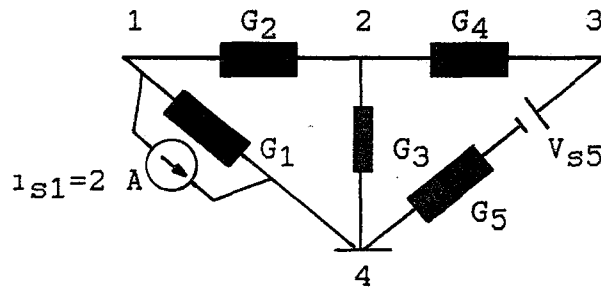
$$Y_n \triangleq A \cdot G \cdot A^T \quad \dots (2.6)$$

olup Y_n : düğüm admitans matrisidir.

$$i_S = A \cdot G \cdot v_S - A \cdot J_S \quad \dots (2.7)$$

Buradanda : $Y_n \cdot e = i_S \quad \dots (2.8)$ yazılabilir.

Bu anlatılanlar bir örnekle pekiştirilirse [9], Şekil 2.2 için :



$G_1 = 2$ mhos
 $G_2 = 1$ mhos
 $G_3 = 3$ mhos
 $G_4 = 1$ mhos
 $G_5 = 1$ mhos

Şekil 2.2 Bir omik devre.

$$A \cdot J = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} * \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$v = A^T * e = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

$$J = G \cdot v + J_S - G \cdot v_S$$

$$\begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Örneğin burada : $J_5 = v_5 - 1$ olduğu görünür.

$$Y_n * e = i_s$$

$$Y_n = A \cdot G \cdot A^T = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$$i_s = A \cdot G \cdot v_S - A \cdot J_S = \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix} \quad \text{Böylece :}$$

$$\begin{bmatrix} 3 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 2 \end{bmatrix} * \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix}$$

$$e = Y_n^{-1} \cdot i_s = \frac{1}{25} \cdot \begin{bmatrix} 9 & 2 & 1 \\ 2 & 6 & 3 \\ 1 & 3 & 14 \end{bmatrix} * \begin{bmatrix} -2 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{25} \cdot \begin{bmatrix} -17 \\ -1 \\ 12 \end{bmatrix}$$

$$v = A^T \cdot e = \frac{1}{25} \cdot \begin{bmatrix} -17 \\ -16 \\ -1 \\ -13 \\ 12 \end{bmatrix}$$

$$J = G \cdot v + J_S - G \cdot v_S = \frac{1}{25} \cdot \begin{bmatrix} 16 \\ -16 \\ -3 \\ -13 \\ -13 \end{bmatrix}$$

Böylece devredeki her elemana ait akım ve gerilim değerleri bulunmuş olur.

2.3. DEĞERLENDİRME:

RNET programı bu yapının en esnek şekilde bilgisayara aktarılmasıyla oluşmuştur [10]. Programın en büyük eksikliği yalnız direnç, akım ve gerilim kaynağı içeren devrelere uygulanabilir olmasıdır. Bir güç elektroniği devresi ise diyot, tristör gibi doğrusal olmayan elemanlarla, indüktans ve sığa gibi enerji depolayan elemanlar içermektedir. Programa bu elemanların da dahil edilmesi için çözümleme yöntemleri kullanılmıştır. Diyot, tristör gibi elemanlar için giriş bölümünde anlatılan anahtarlama yöntemi, indüktans ve sığa içinse bölüm-3'de anlatılan companion model kullanılmıştır.

2.4. RNET PROGRAMININ YAPISAL ÖZELLİKLERİ:

Bu program bağımsız akım ve gerilim kaynakları ve direnç içeren devreler için geliştirilmiştir [10]. Çıkışında her elemanın akımını ve üzerinde düşen gerilimi verir. Ayrıca eğer programa referans düğüm (toprak) tanıtılmamışsa, en çok elemanın bağlı bulunduğu düğümü referans düğüm seçer ve bu düğüm numarasını da çıkışta referans olarak belirtir.

2.4.1.PROGRAM GİRİŞİNİN OLUŞTURULMASI:

N: Devredeki toplam düğüm sayısı

B: Devredeki toplam dal sayısı

V: Devredeki toplam bağımsız gerilim kaynağı sayısı

C: Devredeki toplam bağımsız akım kaynağı sayısı

R: Devredeki toplam direnç sayısı

GR: Referans düğümün numarası, bu sıfır olarak verildiğinde en fazla dal bağlı olan düğüm referans düğüm seçiliyor.

Bu adımdan sonra elemanlar:

- 1- Gerilim kaynakları
- 2- Akım kaynakları
- 3- Dirençler

Sırasına uyularak düğüm numaraları ile belirten programa girilirler. Yalnız gerilim ve akım kaynaklarındaki yön tanımlaması şekil-2.3'deki kurala uygun olarak yapılır:

	1. düğüm verisi	2. düğüm verisi	kaynak değeri
01 ———— (+) ———— (6V) ———— (02)	01	02	+6.0 V
07 ———— (+) ———— (12V) ———— (11)	11	07	-12.0 V
14 ———— (←) ———— (2A) ———— (02)	02	14	+2.0 A
19 ———— (←) ———— (3A) ———— (18)	19	18	-3.0 A

Şekil 2.3 RNET Programının girişi için devre elemanlarının gösterilimi.

RNET programını geliştirerek, düğüm, dal, ...vs. sayılarını girmeden yalnız elemanların bağlı oldukları düğüm ve eleman değerleri girilerek bir program girişi oluşturulmuştur. Yani programa; yine aynı sıra ile elemanları tanıtmak (bağlı oldukları düğüm adları ile birlikte) ve varsa referans düğümün numarasını belirtmek yeterlidir.

Bu program Ek-A'da verilmiştir.



BÖLÜM 3

L,C ELMANLI DEVRELERİN MODELLENMESİ VE BİLGİSAYARLA BİR ÇÖZÜMLEME MODELİ:

Bölüm 2'de anlatıldığı üzere bir güç elektroniği devresi çözümlenirken devrede bulunan indüktans, sığa (gibi enerji depolayan elemanlar), diyot ve tristör (gibi doğrusal olmayan elemanlar) gibi elmanlar bir takım sorunlar yaratırlar. Bu sorunu çözümlmek için en uygun yöntemlerden biri Bölüm 1'de 1.3 kısmında anlatılan "Durum Denklemleri Yöntemi" dir [3]. Bu yaklaşıma, şimdiye kadar yapılan birçok çözümlleme programında başvurulmuştur. Bu tez çalışmasında bundan farklı olan bir çözüme gidilmiş ve aşağıda anlatılan modelleme yöntemleri kullanılmıştır.

3.1.COMPANION MODEL :

3.1.1.BACKWARD EULER YÖNTEMİ :

Bu yöntemde eski nokta kDT yerine yeni nokta $(k+1)DT$ 'de türev alınmış ve şu bağıntı kullanılmıştır:

$$X(t+DT)=X(t)+DT.X'(t+DT).....(3.1)$$

Burada görülüyorki, $((k+1).DT)$ 'deki türevin değeri belirtilmelidir. Buda $(k+1)DT$ 'deki durumun bilinmesini gerektiriyor [11].

Bir yol; 'Forward Euler' yöntemi ile $X((k+1)DT)$ için bir değer hesaplanmasıdır. Sonra bu değer kullanılarak

yukarıdaki türevin değeri hesaplanır.

$$x'(t) = a \cdot x(t) + b \cdot v(t)$$

$x(t)$: Dizge durumu

$v(t)$: Dizge girişi

Buna Backward Euler bağıntısı uygulanırsa bağıntı :

$$X(t+DT) = X(t) + DT \cdot X'(t+DT) \quad , \quad t = k \cdot DT$$

$$x[(k+1) \cdot DT] = x(k \cdot DT) + DT \cdot x'[(k+1) \cdot DT]$$

$$= x(k \cdot DT) + DT \cdot \{ a \cdot x[(k+1) \cdot DT] + b \cdot v[(k+1) \cdot DT] \}$$

$$x(k \cdot DT) + b \cdot DT \cdot v[(k+1) \cdot DT]$$

$$x[(k+1) \cdot DT] = \frac{\quad}{1 - a \cdot DT}$$

a ; negatif olsun, her bir pozitif DT için ;

$$\frac{1}{1 - a \cdot DT} < 1 \quad \text{olur.}$$

Backward Euler bağıntısı companion modeller kurulurken kullanılacaktır [11].

3.1.2. SİĞA İÇİN COMPANION MODEL :

$$\text{Sıganın gerilimi için : } V'_c = \frac{I_c}{C} \quad \dots (3.3) \quad \text{yazılır.}$$

Backward Euler bağıntısı : $X(t+DT) = X(t) + DT \cdot X'(t+DT)$

Bu bağıntı sığaya uygulanırsa :

$$V_C[(k+1).DT] = V_C(k.DT) + DT \cdot V_C'[(k+1).DT]$$

$$V_C[(k+1).DT] = V_C(k.DT) + \left\{ \frac{DT}{C} \cdot I_C[(k+1).DT] \right\} \dots (3.4)$$

Burada :

$$I_C[(k+1).DT] = \left(\frac{C}{DT} \right) \cdot V_C[(k+1).DT] - \left(\frac{C}{DT} \right) \cdot V_C(k.DT) \dots (3.5)$$

3.1.3. İNDÜKTANS İÇİN COMPANION MODEL :

$$\text{Endüktans akımı için : } I_L' = \frac{V_L}{L} \dots (3.6) \text{ yazılır.}$$

Backward Euler bağıntısı uygulanırsa ;

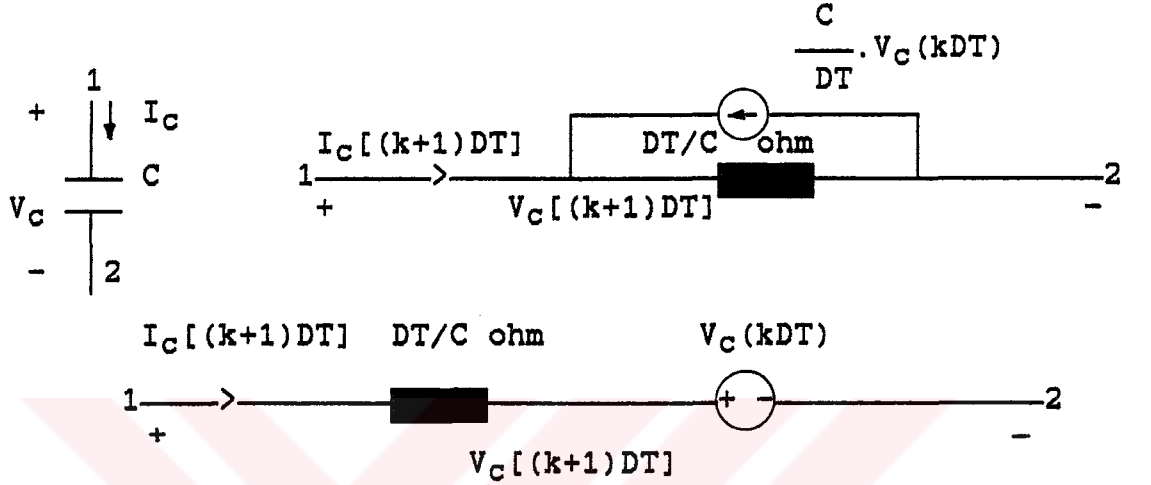
$$I_L[(k+1).DT] = I_L(k.DT) + DT \cdot I_L'[(k+1).DT]$$

$$I_L[(k+1).DT] = I_L(k.DT) + \frac{DT}{L} \cdot V_L[(k+1).DT] \dots (3.7)$$

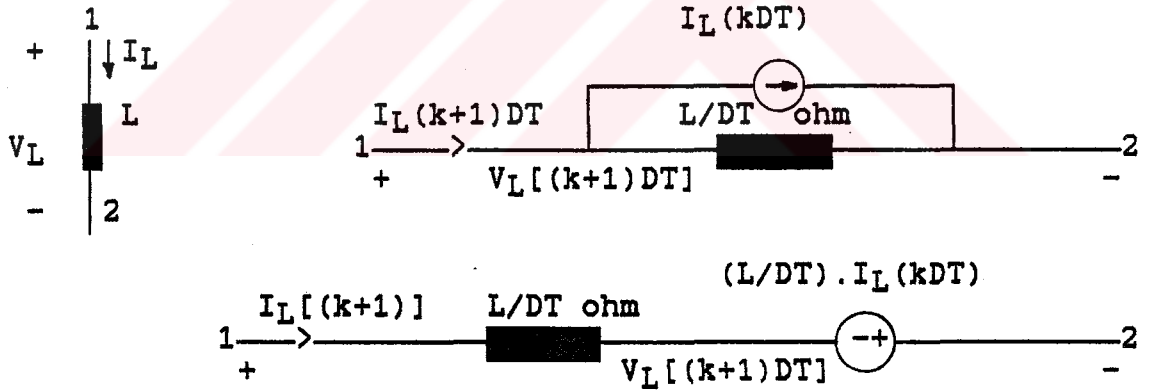
$$V_L[(k+1).DT] = \left(\frac{L}{DT} \right) \cdot I_L[(k+1).DT] - \left(\frac{L}{DT} \right) \cdot I_L(k.DT) \dots (3.8)$$

Elde edilen bu bağıntılardan ; bağıntı (3.5) yardımıyla Şekil 3.1.a 'daki sığa paralel eşdeğeri, bağıntı (3.4) yardımıyla Şekil 3.1.a 'daki sığa seri eşdeğeri elde edilmiş olur.

Aynı şekilde; bağıntı (3.7) yardımıyla Şekil 3.1.b 'deki endüktans paralel eşdeğeri, bağıntı (3.8) yardımıyla da Şekil 3.1.b 'deki endüktans seri eşdeğeri elde edilmiş olur. [11]



(a) Sığa paralel ve seri Companion modeli.



(b) Endüktans paralel ve seri Companion modeli.

Şekil 3.1 Backward Euler bağıntısı ile Companion modellerin kurulması.

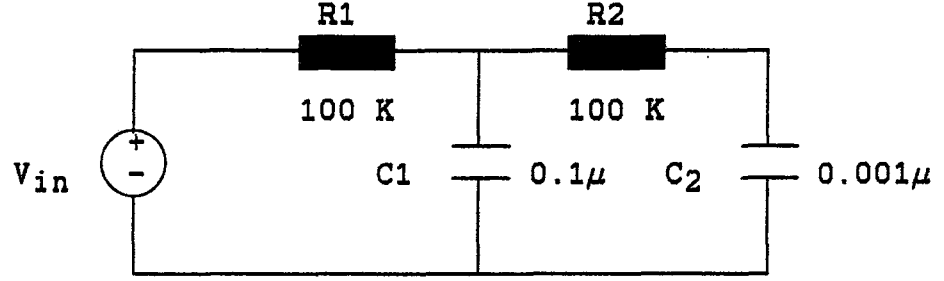
(a) Sığa paralel ve seri Companion modeli.

(b) Endüktans paralel ve seri Companion modeli.

Bu yöntem tez çalışmasında etkin bir şekilde kullanılacağından konuya açıklık getirecek şekilde iki uygulama verilebilir. İlk uygulamada sığanın, diğerinde ise endüktansın Companion modellenmesi açıklanmıştır.

3.2.ALÇAK GEÇİREN SÜZGEÇ İÇİN COMPANION MODEL KURULMASI :

Şekil (3.2)'deki alçak geçiren süzgeç 'in companion eşdeğerini çıkartmak için :



Şekil 3.2 Alçak geçiren süzgeç.

$$DT=0.1 \text{ msn} = 10^{-4} \text{ sn}$$

C_1 sığası için :

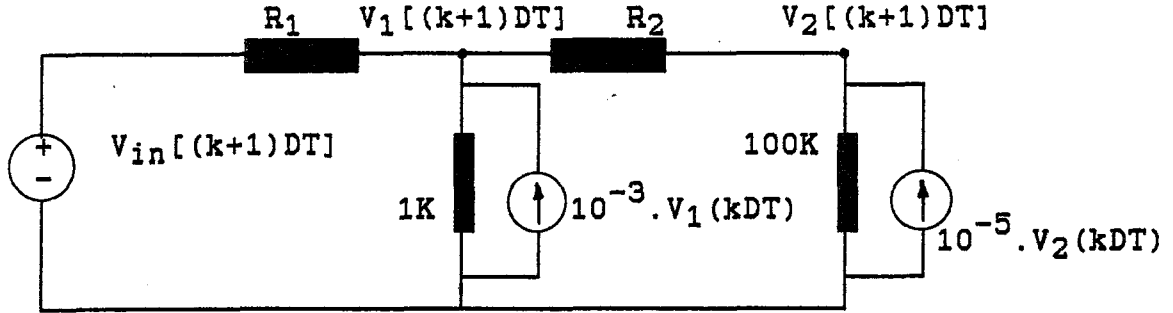
$$R_{\text{comp1}} = \frac{DT}{C_1} = \frac{10^{-4}}{10^{-7}} = 10^3 \text{ ohms} = 1 \text{ Kohms}$$

$$I_{\text{comp1}}(k.DT) = \frac{V_1(k.DT)}{DT/C_1} = 10^{-3} \cdot V_1(k.DT)$$

C_2 sığası için :

$$R_{\text{comp2}} = \frac{DT}{C_2} = \frac{10^{-4}}{10^{-9}} = 10^5 \text{ ohms} = 100 \text{ Kohms}$$

$$I_{\text{comp2}}(k.DT) = \frac{V_2(k.DT)}{DT/C_2} = 10^{-5} \cdot V_2(k.DT)$$



Şekil 3.3 Alçak geçiren süzgeç'in companion eşdeğeri.

1 Dügümü için :

$$\frac{V_1((k+1)DT) - V_{in}((k+1)DT)}{100K} + \frac{V_1((k+1)DT) - V_2((k+1)DT)}{100K} + \frac{V_1((k+1)DT)}{1K} = 10^{-3} \cdot V_1(kDT)$$

2 Dügümü için :

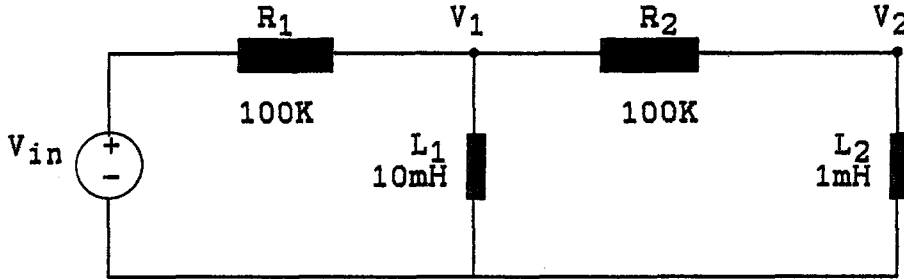
$$\frac{V_2((k+1)DT) - V_1((k+1)DT)}{100K} + \frac{V_2((k+1)DT)}{100K} = 10^{-5} \cdot V_2(kDT)$$

$$\begin{bmatrix} 1.02 & -0.01 \\ -0.01 & 0.02 \end{bmatrix} * \begin{bmatrix} V_1((k+1)DT) \\ V_2((k+1)DT) \end{bmatrix} = \begin{bmatrix} 0.01V_{in}((k+1)DT) + V_1(kDT) \\ 0.01V_2(kDT) \end{bmatrix}$$

$$\begin{bmatrix} V_1((k+1)DT) \\ V_2((k+1)DT) \end{bmatrix} = \frac{1}{0.0203} * \begin{bmatrix} 0.02 & 0.01 \\ 0.01 & 1.02 \end{bmatrix} * \begin{bmatrix} 0.01V_{in}((k+1)DT) + V_1(kDT) \\ 0.01V_2(kDT) \end{bmatrix}$$

Devre bilgileri bu duruma sokulduktan sonra bunun programı rahatlıkla yazılabilir.

3.3.BENZER BİR DEVRE İÇİN COMPANION MODEL KURULMASI :



Şekil 3.4 Companion Modeli kurulacak olan bir devre.

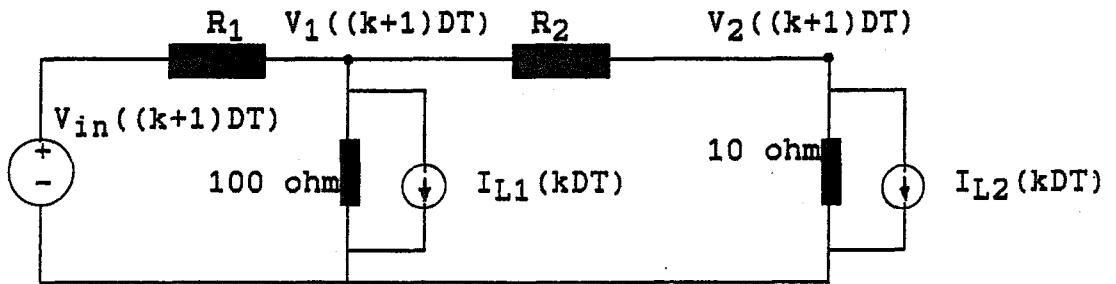
$$DT = 0.1 \text{ msn}$$

L_1 endüktansı için :

$$R_{\text{comp1}} = \frac{L_1}{DT} = \frac{10 \cdot 10^{-3}}{10^{-4}} = 100 \text{ ohms}$$

L_2 endüktansı için :

$$R_{\text{comp2}} = \frac{L_2}{DT} = \frac{1 \cdot 10^{-3}}{10^{-4}} = 10 \text{ ohms}$$



Şekil 3.5 şekil 3.4'deki devrenin companion eşdeğeri.

1 Dügümü için:

$$\frac{V_1((k+1)DT) - V_{in}((k+1)DT)}{100K} + \frac{V_1((k+1)DT) - V_2((k+1)DT)}{100K} + \frac{V_1((k+1)DT)}{100} = -I_{L1}(kDT)$$

2 Dügümü için :

$$\frac{V_2((k+1)DT) - V_1((k+1)DT)}{100K} + \frac{V_2((k+1)DT)}{10} = -I_{L2}(kDT)$$

$$1002V_1((k+1)DT) - V_2((k+1)DT) = V_{in}((k+1)DT) - I_{L1}(kDT) \cdot 10^5$$

$$10001V_2((k+1)DT) - V_1((k+1)DT) = -10^5 \cdot I_{L2}(kDT)$$

$$\begin{bmatrix} 10.02 & -0.01 \\ -0.01 & 100.01 \end{bmatrix} * \begin{bmatrix} V_1((k+1)DT) \\ V_2((k+1)DT) \end{bmatrix} = \begin{bmatrix} 0.01V_{in}((k+1)DT) - 10^3 I_{L1}(kDT) \\ -10^3 \cdot I_{L2}(kDT) \end{bmatrix}$$

Bu devre bilgileri de bu forma sokulduktan sonra bunun programıda rahatlıkla yazılabilir.

3.4.ENDÜKTANS VE SIĞA İÇEREN PROGRAM :

Yukarıda anlatılan ve Ek A 'de verilen programa :

DEVREDE ENDÜKTANS/SIĞA VARMI ? (Y/N)

Sorusu ile girilmiştir. Tabi programda siğa ve endüktansların companion modelleri oluşturulacağından yukarıdaki sorunun yanıtı YES olarak girilirse program companion model hazırlayacağı DT zaman dilimlerinin uzunluğunu ve bunu ne kadar zaman tekrarlayacağına ilişkin

TIME toplam zamanı ister. Bunlar verildikten sonra:

1- Gerilim kaynakları

2- Dirençler

3- Sığalar

4- Endüktanslar

5- Akım kaynakları

sırasına uyularak devre elemanları programa verilir. sığa ve endüktans elemanları girilirken ekrana:

BAŞLANGIÇ DEĞERİ NEDİR?

sorusu gelir. Elemanın bir başlangıç değeri varsa bu değer girilir yoksa sıfır değer girilir.

Bu program Ek-B'de verilmiştir.

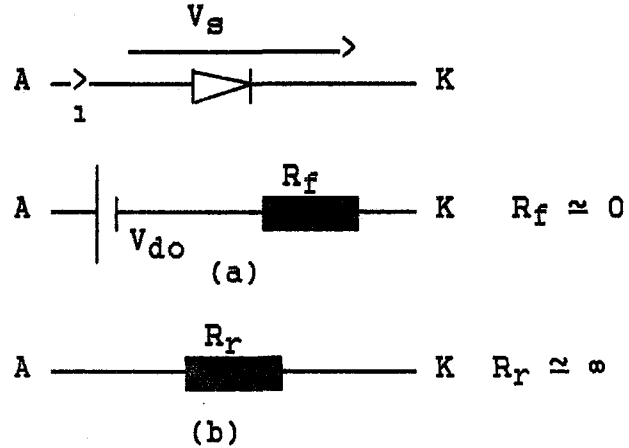
BÖLÜM 4

DIYOT İÇEREN DEVRELERİN MODELLENMESİ VE BİLGISAYARLA BİR ÇÖZÜMLEME MODELİ:

4.1. DIYOT MODELLEMESİ :

4.1.1. ANAHTAR YAKLAŞIMI İLE MODELLEME:

Güç elektroniği devrelerini çözümlerken diyot elemanının doğrusal olmamasından kaynaklanan sorunlarla karşılaşmıştır. Buraya kadar yapılan çalışmada karşılaşılan her tür elemanın (L, C gibi) direnc, akım kaynağı veya gerilim kaynağı eşdeğerleri oluşturulabilmektedir. Diyot için de yapılacak yaklaşımla bu kurala uyulacaktır. Bölüm 1'deki (1.5) ve (1.9)'da verilen çözümlene yaklaşımlarında diyot elemanı bir anahtar gibi düşünülmüş ve böylece doğrusal olmamasından ötürü gelen sorunlardan uzaklaşmıştır [2][4][6][14].



Sekil 4.1 Bir diyot elemanı.
(a) Diyot elemanı iletimde iken, elektronik eşdeğer devresi.
(b) Diyot elemanı tıkmada iken, elektronik eşdeğer devresi.

1.Durum :

Eğer diyot elemanı üzerindeki gerilim V için:

$V > V_{do}$ durumu söz konusu ise, diyot elemanı ilettime geçer. bu durumdaki eşdeğer devresi şekil (4.1.a)'da verilmiştir.

2.Durum :

Eğer diyot elemanı içinden akan i akımı için :

$i < 0$ durumu söz konusu ise, diyot elemanı tıkamaya geçer. Bu durumdaki eşdeğer devresi şekil (4.1.b)'de verilmiştir.

Burada $R_f = 0$ ve $R_r = \infty$ yaklaşımını yapmak yanlış olmaz. Böylece yukarıda anlatılan her iki durum göz önüne alınarak diyot elemanı için ideal anahtar yaklaşımı yapılabilir.

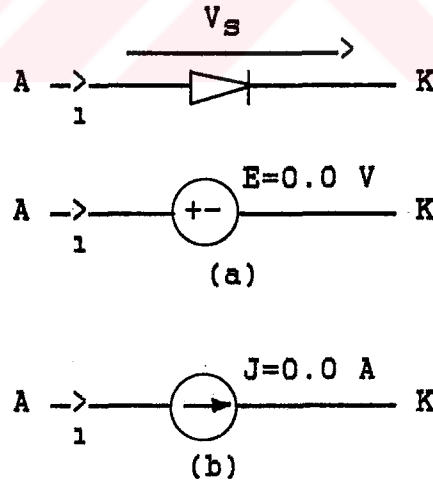
4.1.2.BAĞIMSIZ KAYNAK YAKLAŞIMI İLE MODELLEME :

Düşünülen modelleme, diyot tıkamada iken değeri 0 Amper olan bir akım kaynağını, eşdeğer devre olarak kullanmak ve diyot iletimde iken değeri 0 Volt olan bir gerilim kaynağını, eşdeğer devre olarak kullanmaktır. Başlangıç anında yani devre yani açıldığında, devre yapısındaki diyotların tümü değeri 0 Amper olan akım kaynakları şeklinde tanımlanmışlardır. Bunun nedeni devredeki kaynakların henüz açılmamış olması ve diyotların tıkalı kabul edilmesidir. Buna göre ilk DT hesap adımı sonrası diyot düğümleri arasındaki gerilim, diyot eşik gerilimi ile karşılaştırılacak, eğer büyükse diyot ilettime girecek ve programda 0 Volt değerli gerilim kaynağı olarak tanımlanacaktır. Bu karşılaştırma sonucu olumsuzsa diyot aynı eşdeğerle tanımlanır. Her DT adımında bu karşılaştırma yapılır. ilettime geçen diyotlar için de her DT adımı için "diyottan akan akım sıfırın altına

düştümü ?" karşılaştırması yapılır. Bu şart doğru bulunduğunda diyot yeniden tıkamada olacak ve değeri 0 Amper olan akım kaynağı olarak eşdeğerlendirilecektir.

Burada tamamen diyotlardan oluşan çevrelerin bulunduğu devreler sorun yaratmıştır. Çünkü diyotlar devrenin uygun ağaç yapısında tamamen gerilim kaynakları içeren çevrimler oluşturabilmektedir. Bunun önüne geçebilmek için, diyot verileri her girildiğinde program otomatik olarak, diyot düğümlerine paralel 2 Mohm değerinde direnç elemanları yerleştirir. Zaten bu dirençler pratikte diyot elemanının oluşturduğu kaçak akımları simgelerler.

Programın yapısı geliştirilirken diyotlar için bu modelleme kullanılmıştır. Bu yapılar Şekil (4.2) 'de gösterilmiştir.



Şekil 4.2 Bir diyot elemanı.

- (a) Diyot elemanı iletimde iken, elektronik eşdeğer devresi.
- (b) Diyot elemanı tıkamada iken, elektronik eşdeğer devresi.

4.2. SÜNİZOIDAL GİRİŞ OLUŞTURULMASI :

Çalışmanın bu aşamasına kadar devre girişleri DC düzenindeydi. Çünkü bu çalışmaya temel alınan RNET programı bu düzenleme ile geliştirilmisti. Bu aşamada program girişine yapılan yeni eklemelerle, devre girişlerinin sünizoidal düzende seçilebilmesi sağlanmış oldu.

Örneğin programa bir gerilim kaynağı girişi yapılmış olsun. Bu aşamada ekranda :

A - DOĞRU GERİLİM

B - SİNÜS

C - COSİNÜS

SEÇİNİZ (A/B/C) ?

sorusu gelmektedir. Seçim A ise program bundan önceki bölümlerde sunulan ve EK-A ile EK-B 'de örneklenen düzene göre devam etmektedir.

Seçim B yada C ise ekrana :

FREKANSI GİRİN ?

isteği gelmektedir. Bu veri girildiğinde ise program, her DT adımı için devre girişlerinin hangi degerde olduğunu hesaplayarak devam etmektedir.

Devrede sünizoidal akım kaynağı varsa, bu kaynağın programa girişide, yukarıda anlatılan gerilim kaynağı girişi biçimindedir. Aynı sorulara bu kez akım kaynağı için yanıt verilir.

4.3. DIYOT İÇEREN PROGRAM :

Bölüm 3'de anlatılan ve Ek-B 'de verilen programa:

DEVREDE DIYOT VARMI? (Y/N)

sorusu ile girilmiştir. Bu soru YES diye yanıtlanırsa yine toplam zaman girilir ve :

1- Gerilim kaynakları

2- Dirençler

3- Sığalar

4- İndüktanslar

5- Akım kaynakları

6- Diyotlar

sırasına uyularak devre elemanları programa verilir.

Bu program da Ek-C'de verilmiştir.

4.4. PROGRAM ÇIKIŞI :

Dal Nosu - Elemanın cinsi - Eleman üzerindeki - Elemandan akan
gerilim akım

Referans düğüm: ...

Seklinde bir çıkış hazırlanır. Ayrıca devredeki diyot elemanlarının adlarının yanına (parantez içinde) programa verilen TIME toplam zamanı sonunda elemanın bulunduğu durum yazılmıştır. Yani eleman TIME zamanı sonunda iletimde ise ;

DIYOT (İLETİMDE)

tıkama durumunda ise;

DIYOT (TIKAMADA)

sekinde bir çıkış hazırlanır.

4.4.1. GRAFİK ÇIKIŞ :

Program sonuçlandığında ve sayısal sonuçlar oluştuğunda, ekrana aşağıdaki gibi bir soru gelir :

AYRICA GRAFİK SONUÇ İSTENİYORMU ? (Y/N)

Bu soru NO diye yanıtlanırsa program durur. Eger YES diye yanıtlanırsa ekranda bir başka soru belirir :

A-ELEMAN İÇİNDEN AKAN AKIMIN GRAFİĞİ

B-ELEMAN ÜZERİNDEKİ GERİLİMİN GRAFİĞİ

SEÇİNİZ (A/B) ?

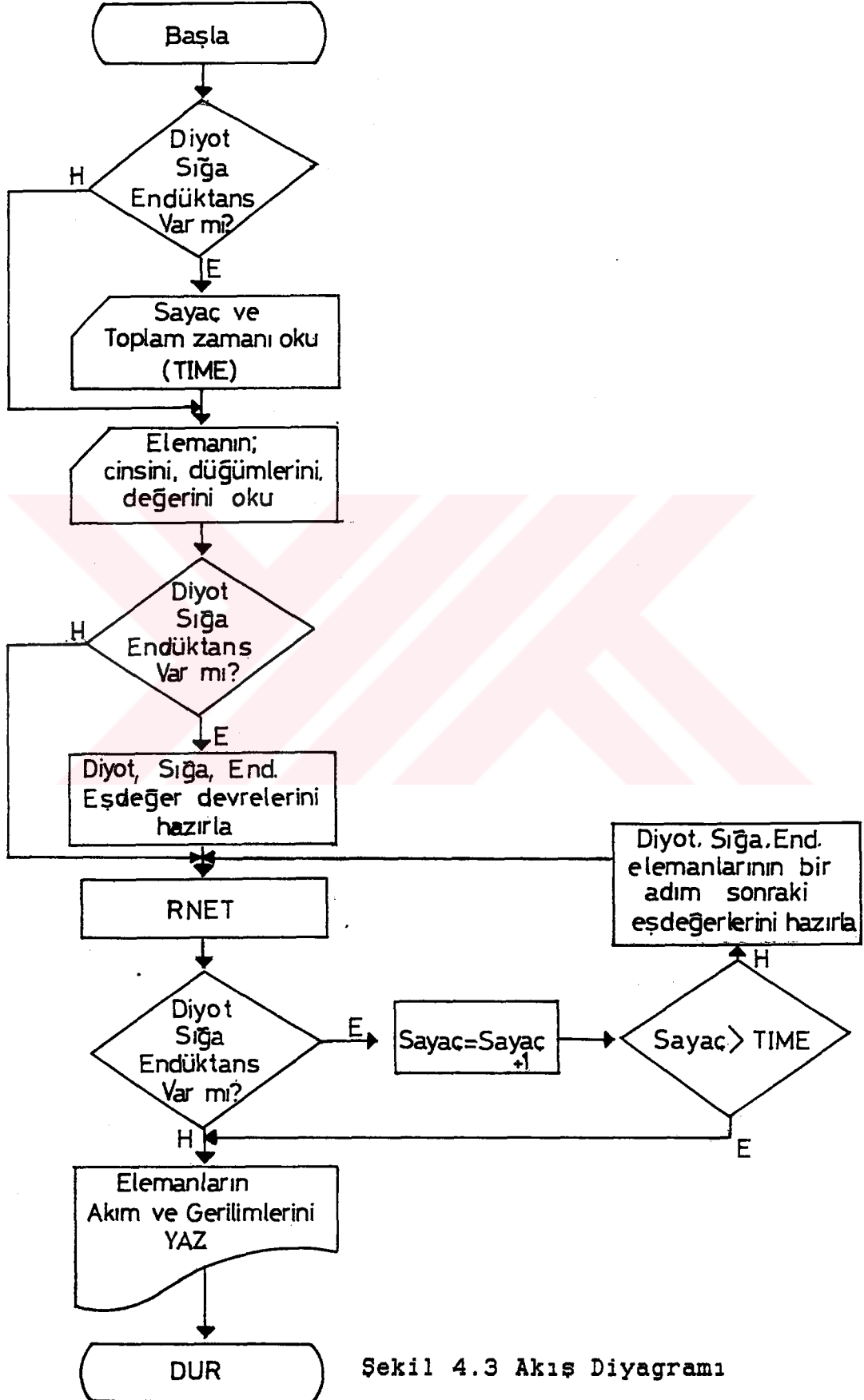
Bu sorunun yanıtı A ve B seçeneklerinden biri tercih edilerek girilir. Bu işlemin ardından aşağıdaki gibi bir istek gelir :

GRAFİĞİNİ GÖRMEK İSTEDİĞİNİZ ELEMANIN NOSUNU GIRIN

Eleman nosu verilerin programa giriliş sırasına göre belirlenir. Bu sayı yazılıp ENTER'e basıldığında istenen grafik ekranda oluşur. Grafik çizildikten sonra, altına tekrar yukarıdaki sorular birer birer gelmeye başlar. Böylece istenirse devrede bulunan her eleman için, program yeniden koşturulmadan gerilim ve akım grafikleri çizdirilebilir.



4.5. AKIŞ DIYAGRAMI [12][13]:



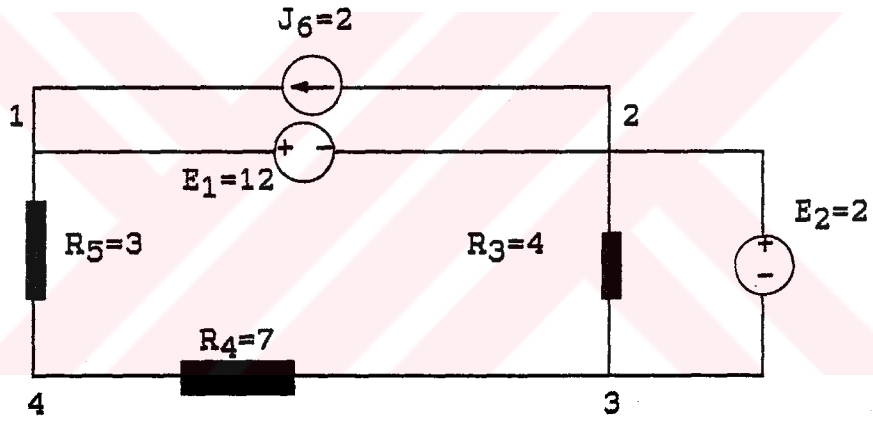
Şekil 4.3 Akış Diyagramı

BÖLÜM 5

UYGULAMA

5.1.RNET PROGRAMININ UYGULAMALARI :

5.1.1.KARIŞIK KAYNAKLI UYGULAMA :



Şekil 5.1 Akım ve Gerilim kaynaklarını bir arada içeren devre.

Bilgisayardan alınan çıkış :

DÜĞÜMLER G.KAYNAKLARI DİRENÇLER A.KAYNAKLARI

1 - 2 -12.000

2 - 3 2.000

2 - 3 4.000

3 - 4 7.000

4 - 1 3.000

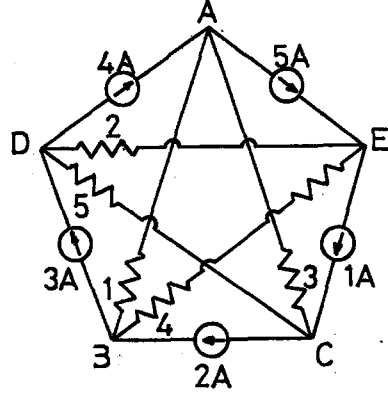
2 - 1 2.000

N= 4 B= 6 R= 3 V= 2 J= 1

TOPRAK= 2

ELEMAN NOSU	ELM.ÜZERİNDEKİ GERİLİM	ELM.AKAN AKIM
1 (GERİLİM KAYNAĞI)	-1.20000E+01 VOLT	3.00000E+00 AMP
2 (GERİLİM KAYNAĞI)	2.00000E+00 VOLT	5.00000E-01 AMP
3 (DİRENÇ)	2.00000E+00 VOLT	5.00000E-01 AMP
4 (DİRENÇ)	7.00000E+00 VOLT	1.00000E+00 AMP
5 (DİRENÇ)	3.00000E+00 VOLT	1.00000E+00 AMP
6 (AKIM KAYNAĞI)	1.20000E+01 VOLT	2.00000E+00 AMP

5.1.2.YALNIZ AKIM KAYNAKLARI İÇEREN UYGULAMA :



Şekil 5.2 Akım kaynakları içeren bir devre.

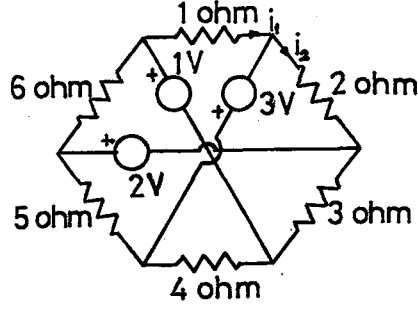
Bilgisayardan alınan çıkış :

DÜĞÜMLER	G.KAYNAKLAR	DİRENÇLER	A.KAYNAKLARI
2 - 5		1.000	
1 - 3		2.000	
2 - 4		3.000	
3 - 5		4.000	
1 - 4		5.000	
1 - 2			4.000
2 - 3			5.000
3 - 4			1.000
4 - 5			2.000
1 - 5			-3.000

$$N= 5 \quad B=10 \quad R= 5 \quad V= 0 \quad J= 5$$

ELEMAN NOSU	ELM.ÜZERİNDEKİ GERİLİM	ELM.AKAN AKIM
1 (DIRENÇ)	-1.00000E+00 VOLT	-1.00000E+00 AMP
2 (DIRENÇ)	-4.00000E+00 VOLT	-2.00000E+00 AMP
3 (DIRENÇ)	0.00000E+00 VOLT	0.00000E+00 AMP
4 (DIRENÇ)	8.00000E+00 VOLT	2.00000E+00 AMP
5 (DIRENÇ)	5.00000E+00 VOLT	1.00000E+00 AMP
6 (AKIM KAYNAĞI)	5.00000E+00 VOLT	4.00000E+00 AMP
7 (AKIM KAYNAĞI)	-9.00000E+00 VOLT	5.00000E+00 AMP
8 (AKIM KAYNAĞI)	9.00000E+00 VOLT	1.00000E+00 AMP
9 (AKIM KAYNAĞI)	-1.00000E+00 VOLT	2.00000E+00 AMP
10 (AKIM KAYNAĞI)	4.00000E+00 VOLT	-3.00000E+00 AMP

5.1.3.YALNIZ GERİLİM KAYNAKLARI İÇEREN UYGULAMA :



Şekil 5.3 Gerilim kaynakları içeren bir devre.

Bilgisayardan alınan çıkış :

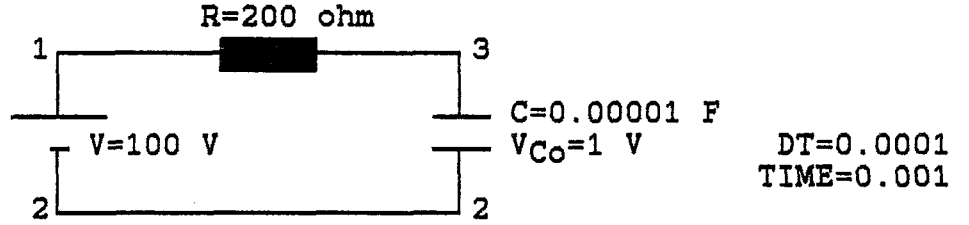
DÜĞÜMLER	G.KAYNAKLARI	DİRENÇLER	A.KAYNAKLARI
1 - 4	1.000		
6 - 3	2.000		
5 - 2	3.000		
1 - 2		1.000	
2 - 3		2.000	
3 - 4		3.000	
4 - 5		4.000	
5 - 6		5.000	
6 - 1		6.000	

$$N= 6 \quad B= 9 \quad R= 6 \quad V= 3 \quad J= 0$$

ELEMAN	NOSU	ELM.ÜZERİNDEKİ GERİLİM	ELM.AKAN AKIM
1	(GERİLİM KAYNAĞI)	1.00000E+00 VOLT	-7.79279E-01 AMP
2	(GERİLİM KAYNAĞI)	2.00000E+00 VOLT	2.25225E-02 AMP
3	(GERİLİM KAYNAĞI)	3.00000E+00 VOLT	-9.59459E-01 AMP
4	(DİRENÇ)	9.54955E-01 VOLT	9.54955E-01 AMP
5	(DİRENÇ)	-9.00900E-03 VOLT	-4.50450E-03 AMP
6	(DİRENÇ)	5.40540E-02 VOLT	1.80180E-02 AMP
7	(DİRENÇ)	-3.04504E+00 VOLT	-7.61261E-01 AMP
8	(DİRENÇ)	9.90991E-01 VOLT	1.98198E-01 AMP
9	(DİRENÇ)	1.05405E+00 VOLT	1.75676E-01 AMP

5.2.COMPANION MODELLEME İLE GENİŞLETİLMİŞ PROGRAMIN UYGULAMALARI

5.2.1.SİĞA YÜKLÜ UYGULAMA :



Şekil 5.4.Kapasitif yüklü bir devre.

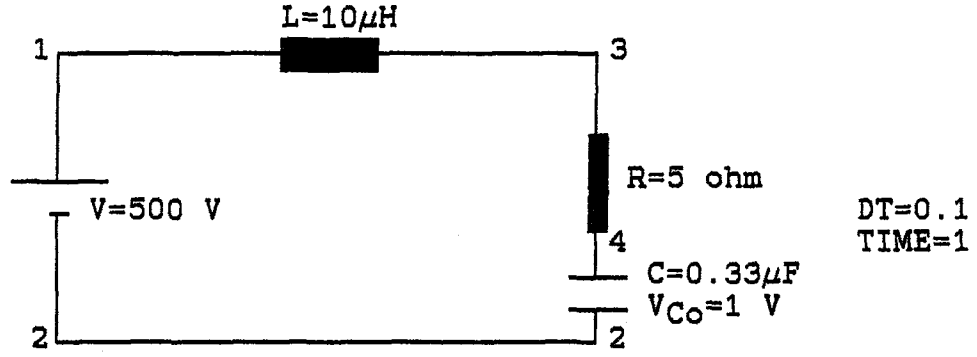
Bilgisayardan alınan çıkış :

DÜĞÜMLER	G.KAYNAKLARI	DİRENÇLER	A.KAYNAKLARI
1 - 2	100.000		
1 - 3		200.000	
DÜĞÜMLER	SİĞALAR	İNDÜKTANSLAR	İLK DEĞER
3- 2	0.00001		1.00

N= 3 B= 3 R= 1 V= 1 SİĞA= 1 TOPRAK= 1

ELEMAN NOSU	ELM.ÜZERİNDEKİ GERİLİM	ELM.AKAN AKIM
1(GERİLİM KAYNAĞI)	1.00000E+02 VOLT	-2.89417E-01 AMP
2(DİRENÇ)	5.78832E+01 VOLT	2.89416E-01 AMP
3(SİĞA)	4.21168E+01 VOLT	2.89417E-01 AMP

5.2.2. KARMAŞIK YÜKLÜ UYGULAMA :



Şekil 5.5 Karmaşık yüklü bir devre.

Bilgisayardan alınan çıkış :

DÜĞÜMLER	G.KAYNAKLARI	DİRENÇLER	A.KAYNAKLARI
1 - 4	500.000		
2 - 3		5.000	

DÜĞÜMLER	SİGALAR	İNDÜKTANSLAR	İLK DEĞER
3- 4	0.0000003300		1.00
1- 2		0.0000001	2.00

N= 4 B= 4 R= 1 V= 1 J= 0 SIGA= 1 IND= 1

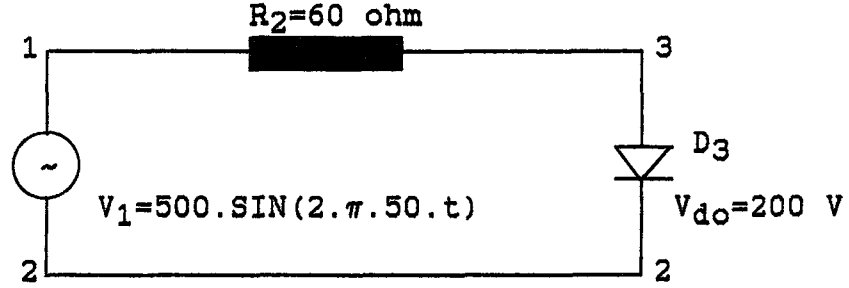
TOPRAK= 1

ELEMAN NOSU	ELM.ÜZERİNDEKİ GERİLİM	ELM.AKAN AKIM
1(GERİLİM KAYNAĞI)	5.00000E+02 VOLT	6.66631E-11 AMP
2(DİRENÇ)	-3.33316E-10 VOLT	-6.66631E-11 AMP
3(SİĞA)	4.99666E+02 VOLT	-6.66631E-11 AMP
4(İNDÜKTANS)	0.00000E+00 VOLT	-6.66631E-11 AMP



5.3.DIYOT İÇEREN PROGRAMIN UYGULAMALARI :

5.3.1.BİR YARIM DALGA DOGRULTUCU DEVRE İÇİN UYGULAMA :



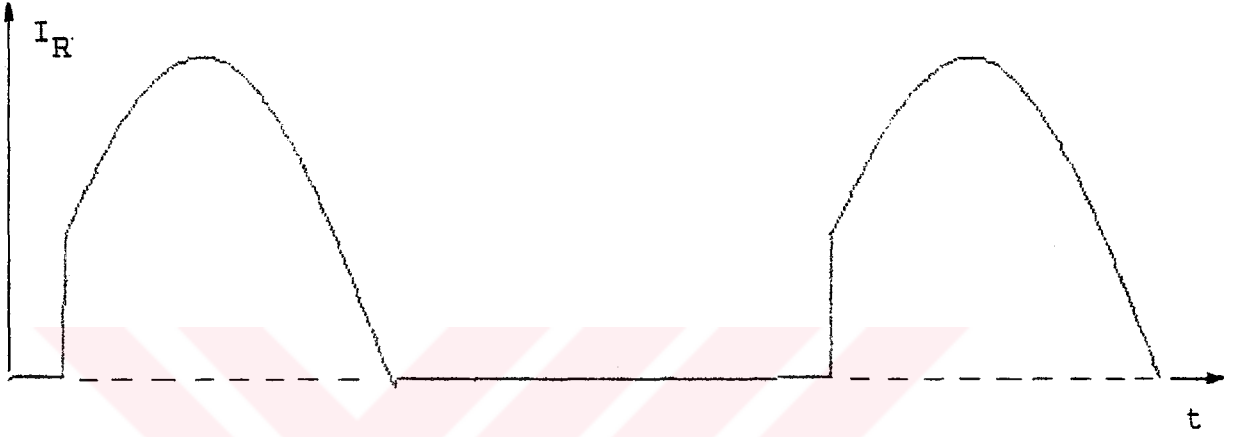
Şekil 5.6 Yarım Dalga Doğrultucu.

Bu uygulamada; devre verileri, $DT=0.0001$ zaman dilimi ve $TIME=0.03$ toplam zamanı ile programa girilir. Program çıkışında; her DT adımı için, devreye ait elemanlara ilişkin akım ve gerilim değerleri hesaplanmıştır. $TIME$ toplam zamanı sonunda aşağıdaki sonuç hesaplanmıştır:

ELEMAN NOSU	ELM.ÜZERİNDEKİ GERİLİM	ELM.AKAN AKIM
1(GERİLİM KAYNAĞI)	6.66380E-03 VOLT	-1.11063E-04 A
2(DİRENÇ)	6.66380E-03 VOLT	1.11063E-04 A
3(DIYOT(İLETİMDE))	1.11063E-10 VOLT	1.11063E-04 A

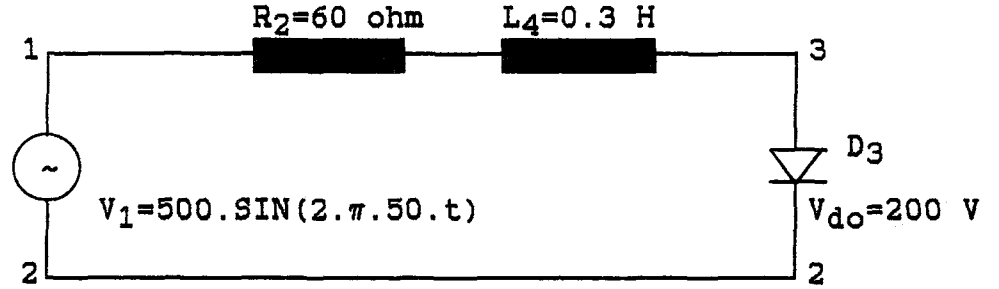
Ayrıca Bölüm-4'ün 4.4.1. alt bölümünde anlatıldığı gibi programdan grafik sonuç istenmiştir. İstenirse devredeki her bir eleman için, akım yada gerilimlerinin zamana göre grafikleri çizdirilebilir.

Aşağıda devreden akan akımın zamana göre grafiği çizdirilmiştir. Bu grafikten de hazırlanan programın doğruluğuna karar verilebilir.



Şekil 5.7 Yarım dalga doğrultucu devre içinden akan akımın, zamana göre şekli.

5.3.2. İNDÜKTİF YÜKLÜ BİR YARIM DALGA DOĞRULTUCU DEVRE İÇİN UYGULAMA :



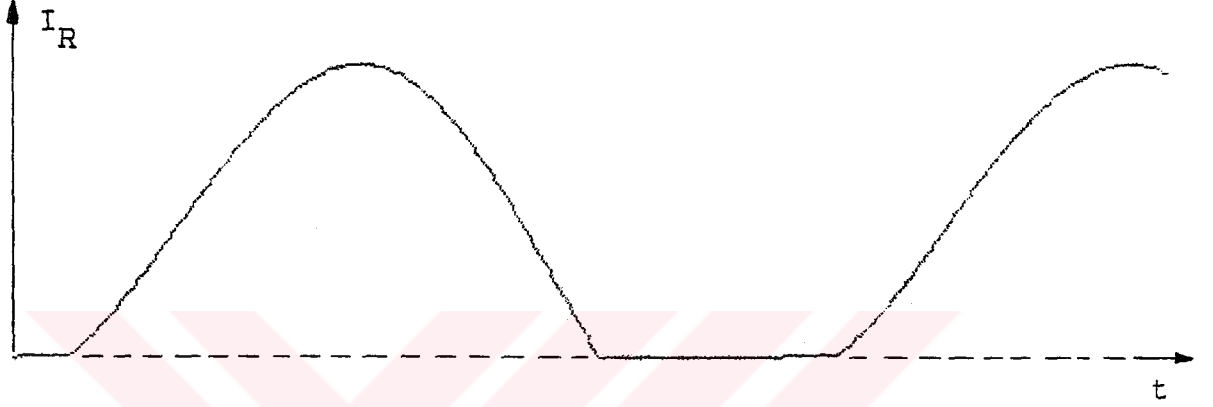
Şekil 5.8 Endüktif Yüklü Yarım Dalga Doğrultucu.

Bu uygulamada; devre verileri, $DT=0.0001$ zaman dilimi ve $TIME=0.03$ toplam zamanı ile programa girilir. Program çıkışında; her DT adımı için, devreye ait elemanlara ilişkin akım ve gerilim değerleri hesaplanmıştır. $TIME$ toplam zamanı sonunda aşağıdaki sonuç hesaplanmıştır:

ELEMAN NOSU	ELM. ÜZERİNDEKİ GERİLİM	ELM. AKAN AKIM
1 (GERİLİM KAYNAĞI)	6.66380E-03 VOLT	-7.34497E+00 A
2 (DİRENC)	1.46899E+02 VOLT	7.34497E+00 A
3 (DİYOT (İLETİMDE))	7.34497E-06 VOLT	7.34497E+00 A
4 (İNDÜKTANS)	-1.46893E+02 VOLT	-4.89642E-02 A

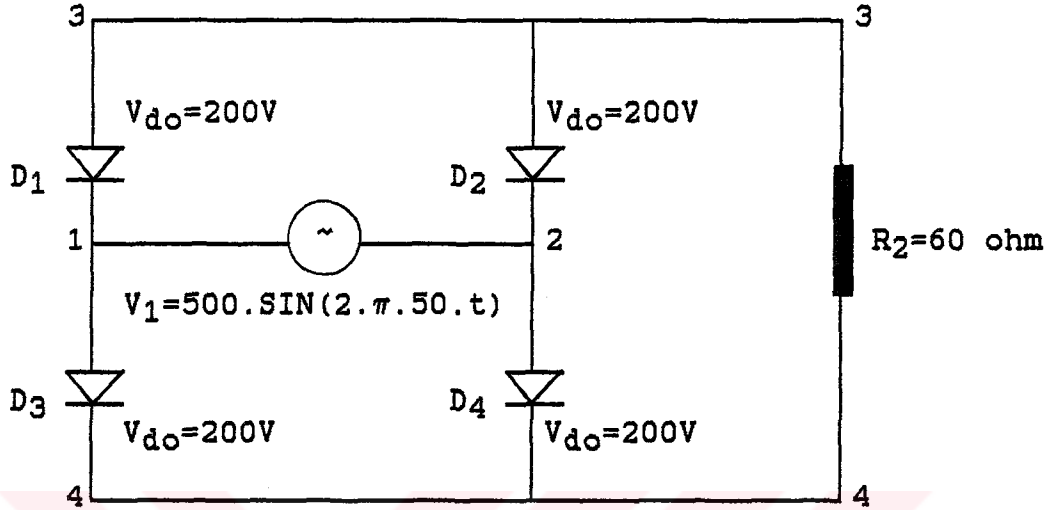
Ayrıca Bölüm-4'ün 4.4.1. alt bölümünde anlatıldığı gibi programdan grafik sonuç istenmiştir. İstenirse devredeki her bir eleman için, akım yada gerilimlerinin zamana göre grafikleri çizdirilebilir.

Aşağıda devreden akan akımın zamana göre grafiği çizdirilmiştir. Burada da devrenin doğru çözümlendiği izlenilebilir.



Şekil 5.9 İndüktif yüklü yarım dalga doğrultucu devre için den akan akımın, zamana göre şekli.

5.3.3. BİR TAM DALGA DOĞRULTUCU DEVRE İÇİN UYGULAMA :



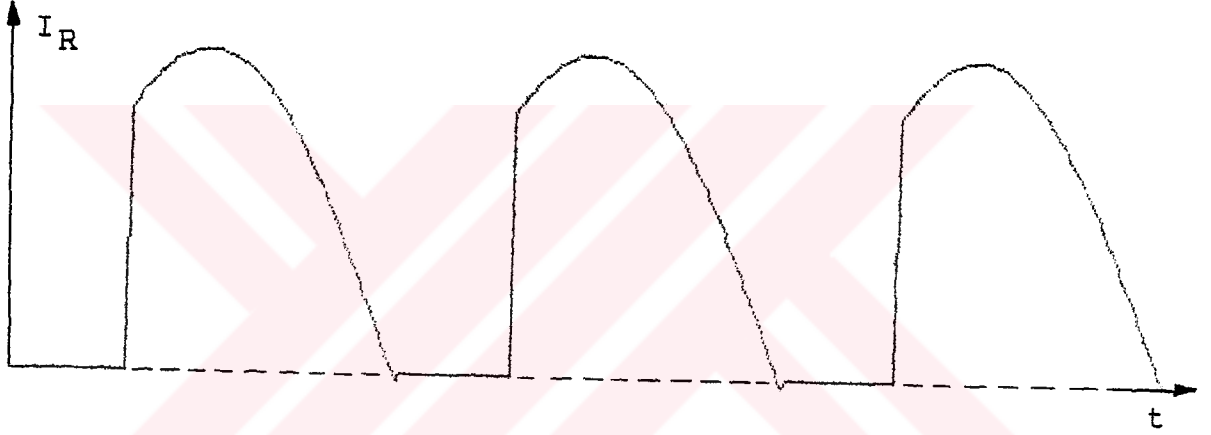
Şekil 5.10 Tam Dalga Doğrultucu.

Bu uygulamada; devre verileri, $DT=0.0001$ zaman dilimi ve $TIME=0.03$ toplam zamanı ile programa girilir. Program çıkışında; her DT adımı için, devreye ait elemanlara ilişkin akım ve gerilim değerleri hesaplanmıştır. $TIME$ toplam zamanı sonunda aşağıdaki sonuç hesaplanmıştır:

ELEMAN NOSU	ELM.ÜZERİNDEKİ GERİLİM	ELM.AKAN AKIM
1 (GERİLİM KAYNAĞI)	6.66380E-03 VOLT	-1.11077E-04 A
2 (DİRENÇ)	6.66380E-03 VOLT	1.11063E-04 A
3 (DİYOT (İLETİMDE))	6.66380E-15 VOLT	6.66380E-09 A
4 (DİYOT (TIKAMADA))	-6.66380E-03 VOLT	-6.66380E-09 A
5 (DİYOT (TIKAMADA))	-6.66380E-03 VOLT	-6.66380E-09 A
6 (DİYOT (İLETİMDE))	1.11070E-10 VOLT	1.11070E-04 A

Ayrıca Bölüm-4'ün 4.4.1. alt bölümünde anlatıldığı gibi programdan grafik sonuç istenmiştir. İstenirse devredeki her bir eleman için, akım yada gerilimlerinin zamana göre grafikleri çizdirilebilir.

Aşağıda devredeki direnç elemanı içinden akan akım çizdirilmiştir.



Şekil 5.11 Tam dalga doğrultucu devredeki direnç yükü içinden akan akımın, zamana göre şekli.

BÖLÜM 6

TARTIŞMA

Bu programda hazırlanan Companion model ile indüktans ve sığalar devrede direnç ve akım kaynağı şeklinde ve de diyot ve tristör elemanların anahtar şeklinde yer almışlardır. Program bir önceki adıma bakarak indüktans ve sığa eşdeğer elemanlarının değerlerini belirler, diyot ve tristörlerinde açık ya da kapalı durumda olacağına karar verir. Bu yapılırken devre girişinde verilen kaynakların yalnız DC olması zorunluluğu da kaldırılmıştır. Yani programa eklenen bir kısım ile de bağımsız kaynak girişleri DC - AC esnekliğine kavuşmuşlardır.

Hazırlanan bu programla basit yapıdaki devrelerden daha karmaşık devrelere doğru adım adım çıkılmış ve program yapısı daha esnekleştirilmiştir. Karmaşık devrelerdeki üstünlüğü ve veri girişinin daha basit yapıda olması bu programı RNET programına üstün kılmıştır.

Programın kullanımındaki en büyük yararlarından biri de sudur; veri girişinde bir hata olduğu takdirde, bu hatanın cinsini ekrana yazarak uyarır. Yani :

- 1- Yalnız gerilim kaynaklarından oluşan bir çevre olması
- 2- Dügümlerin eksik girilmesi,
- 3- Dalların eksik girilmesi,

- 4- Yalnız akım ve gerilim kaynakları olan bir devre girilmesi,
- 5- Direnci olmayan bir devre girilmesi,
- 6- Dal sayısına oranla fazla düğüm girilmesi,
- 7- Yalnız akım kaynaklarının bağlı olduğu düğüm bulunması,
- 8- Devre yapısının kopuk girilmesi,

gibi hallerde, bu hata ekrana yazılarak, kullanıcı uyarılmaktadır. Programa girilen veri tam bir elektronik devre özelliğinde olana kadar bu uyarı yapılır.

KAYNAKLAR:

1. Hoft, R. G. And Casteel, J. B., Power Electronic Circuit Analysis Tecniques, 2nd IFAC Symp. On Cont. In Power Electronics And Electrical Drives, Vol.2, 1977 ,987-1024.
2. Turanlı, M. , Güç Elektronığı Devrelerini Çözümlmek İçin Bir Sayısal Bilgisayar Programı, Yüksek Lisans Tezi, O.D.T.Ü. Elektrik Mühendisliği Bölümü, Ankara, 1980.
3. Önbilgin, G. ve Şenlik, I. ,Güç Elektronığı Devrelerinin Bilgisayar Destekli Tasarımı. Güç Elektronığı Semineri, Kısım.10, 10(1989), 264-277.
4. Kutman, T. , Aplication of Stiffly Stable Algoritims In The Digital Computation of The SCR Circuits, 2th IFAC Symp. On Cont. In Power Electronics And Electrical Drives, Vol.1, 1978, 15-18.
5. Racz, I. , Analysis of Three - Phase Semiconductor Connections By Park-Vectors , 2th IFAC Symp. On Cont. In Power Electronics And Electrical Drives, Vol.1, 1978 , 37-45.
6. Boulos, F. And Glaize, C. ,Variable Topology: A New Approach In C.A.D., Second European Conference On Power Electronics And Applications: EPE. Vol.2, 9(1987),767-771.
7. Szczeany,R. And Gruszczynski,P. ,Modelling and Simulation of Converter Systems, Second European Conference On Power Electronics And Applications: EPE. Vol.2, 9(1987),789-794.

8. Karni, S., Analysis of Electrical Networks, John Wiley & Sons Co., New York, 1986.
9. Desoer, C. A. And Kuh, E. S., Basic Circuit Theory, McGraw-Hill Book Co., New York, 1969.
10. Wing, O. , Circuit Theory With Computer Methods, Holt-Rinehart and Winston Co., New York, 1972.
11. Mastascusa, E. J. , Computer-Assisted Network and System Analysis , John Wiley And Sons Co., Canada, 1988.
12. Coffetti, A. , Petrecca, G. And Savini, A. , A Digital Simulation Of Converter Circuits, 2th IFAC Symp. On Cont. In Power Electronics And Electrical Drives, Vol.1, 1978 , 7-14.
13. Foch, H., Reboulet, C. And Schonek, J. , A General Digital Computer Simulation Programme For Thyristor Static Converters (Programme Sagso) Application Examples, 2th IFAC Symp. On Cont. In Power Electronics And Electrical Drives, Vol.1, 1978 , 1-6.
14. Hui, S.Y.R. And Christopoulos C. , A Discrete Approach To The Modelling of Power Switching Networks, 20th Annual IEEE Power Electronics Specialists Conference. Vol.1, 6(1989), 130-137.

EKLER :

EK-A:R,E,I ELEMANLARI İÇEREN DEVRELER İÇİN RNET PROGRAMI

INTEGER N,B,R,V,C,T,GROUND,CR,TR

REAL JJ,MA,MB,IR

DIMENSION ALPHA(9)

DIMENSION TBTREE(49,49)

DIMENSION APRIME(20,50)

DIMENSION FMT(7)

DIMENSION TBJE(20,20),TBRCE(20,49)

DIMENSION YE(49,49),YO(49,49),L(20),M(20)

DIMENSION A(20,50),ATREE(19,49),ACHORD(19,49),BTREE(49,49),

1 AARRAY(2500),NODEI(50),NODEJ(50),RR(50),E(50),JJ(50),

2 INDEX(50),RTREE(50),RCHORD(50),GTREE(50),GCHORD(500),VR(50),

3 IR(50),TEMP(50),X(19,50),XX(19,49),MA(19,49),MB(19,49),

4 XT(19,49),XD(19,49),XE(19,49),XF(49,49),XG(49,49),BJE(19,49)

5 ,BRCE(19,49),BRCRT(19,49),BJRT(19,49),SUM(50),SUM1(50),

6 SUM2(50),SUM3(50),SUM4(50),TBRCRT(49,49),TBJRT(49,49)

CHARACTER*1 TYPE(50)

OPEN(UNIT=7,FILE='SONUC1',STATUS='NEW')

C

```
8 WRITE(6,*) '          SIRAYA UYUN :          1.GERILIM KAYNAKLARI '
  WRITE(6,*) '                                          2.DIRENÇLER '
  WRITE(6,*) '                                          5.AKIM KAYNAKLARI '
  EPSI=1.E-12
  R=0
  C=0
  B=0
  V=0
  KAT=1
```

C

```
      K=1
100  WRITE(6,*) 'TYPE '
      READ(5,102) TYPE(K)
102  FORMAT(A1)
      IF(TYPE(K).EQ.'Z') GO TO 159
      WRITE(6,*) 'NODEI  NODEJ  VALUE'
      READ*,NODEI(K),NODEJ(K),VAL
159  IF(TYPE(K).EQ.'V') THEN
      V=V+1
      B=B+1
      E(K)=VAL
      ELSE IF (TYPE(K).EQ.'R') THEN
      R=R+1
      RR(K)=VAL
      B=B+1
      ELSE IF (TYPE(K).EQ.'J') THEN
      B=B+1
      C=C+1
      JJ(K)=VAL
      ELSE IF (TYPE(K).EQ.'Z') THEN
      GO TO 152
      ELSE
      GO TO 9012
```

```

END IF
K=K+1
GO TO 100
C
152 N=0
DO 157 K=1,B
IF(N.GT.NODEI(K)) THEN
N=N
ELSE
N=NODEI(K)
END IF
IF(N.GT.NODEJ(K)) THEN
N=N
ELSE
N=NODEJ(K)
END IF
157 CONTINUE
IF(N.EQ.0) GO TO 30000
WRITE(6,*) 'TOPRAK?'
READ*,GROUND
IF(V.EQ.0) GO TO 2
1 IF(C.EQ.0) GO TO 4
GO TO 6
2 WRITE(6,3)
3 FORMAT(15X,'HIÇ GERİLİM KAYNAĞI YOK.')
GO TO 1
4 WRITE(6,5)
5 FORMAT(15X,'HIÇ AKIM KAYNAĞI YOK.')
6 IF(N.LT.2) GO TO 9002
IF(B.LT.2) GO TO 9004
IF(R.EQ.0) GO TO 9006
IF(B.LT.N) GO TO 9008
NM=N-1
NB=B-(N-1)
TR=NM-V
KV=V+1
CR=R-(NM-V)
NP=N+1
KVR=V+R
C
WRITE(7,*) 'DÜĞÜMLER G.KAYNAKLARI DİRENÇLER A.KAYKAKLARI'
DO 153 K=1,B
IF(TYPE(K).EQ.'V') THEN
154 WRITE(7,154) NODEI(K),NODEJ(K),E(K)
FORMAT(1X,I3,'-',I3,1X,F12.3)
ELSE IF(TYPE(K).EQ.'R') THEN
155 WRITE(7,155) NODEI(K),NODEJ(K),RR(K)
FORMAT(1X,I3,'-',I3,11X,F12.3)
ELSE IF (TYPE(K).EQ.'J') THEN
158 WRITE(7,158) NODEI(K),NODEJ(K),JJ(K)
FORMAT(1X,I3,'-',I3,24X,F12.3)
END IF

```

```

153 CONTINUE
C
WRITE(7,*) '*****'
WRITE(7,161)N,B,R,V,C
161 FORMAT(2X,'N=',I2,2X,'B=',I2,2X,'R=',I2,2X,'V=',I2,2X,'J=',I2)
C
C*****
C
C INCIDENCE MATRIS A(I,K)'NIN HAZIRLANMASI
C
321 DO 300 K=1,B
INDEX(K)=K
DO 300 I=1,N
IF(NODEI(K).EQ.I) GO TO 10
IF(NODEJ(K).EQ.I) GO TO 20
A(I,K)=0.0
GO TO 300
10 A(I,K)=1.0
GO TO 300
20 A(I,K)=-1.0
300 CONTINUE
C
DO 360 I=1,N
CHECK=0.0
DO 350 K=1,B
IF(A(I,K).EQ.EPSI) GO TO 350
CHECK=CHECK+1.0
350 CONTINUE
IF(CHECK.LT.2.0) GO TO 9010
360 CONTINUE
IF(GROUND.NE.0) GO TO 30
BIG=0.0
DO 400 I=1,N
SUM(I)=0.0
DO 500 K=1,B
500 SUM(I)=ABS(A(I,K))+SUM(I)
IF(SUM(I).LE.BIG) GO TO 400
BIG=SUM(I)
GROUND=I
400 CONTINUE
WRITE(7,987) GROUND
987 FORMAT(5X,'TOPRAK=',I3)
C
30 DO 600 K=1,B
DO 600 I=1,NM
IF(I.GE.GROUND) GO TO 35
A(I,K)=A(I,K)
GO TO 600
35 IP=I+1
A(I,K)=A(IP,K)
600 CONTINUE
DO 650 I=1,NM

```

```

DO 650 J=1,B
650 APRIME(I,J)=A(I,J)
      CALL TRIANG(A,NM,B,INDEX)
C
      KVR=V+R
      DO 805 K=1,NM
      IF(INDEX(K).GT.KVR) GO TO 9014
805 CONTINUE
      DO 810 K=N,B
      IF(INDEX(K).LE.V) GO TO 9000
810 CONTINUE
C
      DO 820 J=1,NM
      IF(A(J,J).EQ.EPSI) GO TO 9012
820 CONTINUE
C
      DO 800 K=1,B
      DO 800 I=1,NM
800 A(I,K)=APRIME(I,INDEX(K))
C
      DO 69 K=1,NM
      DO 69 I=1,NM
      ATREE(I,K)=A(I,K)
69 CONTINUE
      DO 1600 K=1,NB
      DO 1600 I=1,NM
      KH=K+NM
      ACHORD(I,K)=A(I,KH)
1600 CONTINUE
C
      DO 88 J=1,NM
      DO 88 I=1,NM
      NU=I+NM*(J-1)
      AARRAY(NU)=ATREE(I,J)
88 CONTINUE
      CALL MINV(AARRAY,NM,D,L,M)
      DO 25 J=1,NM
      DO 25 I=1,NM
      NU=I+NM*(J-1)
      ATREE(I,J)=AARRAY(NU)
25 CONTINUE
C
      DO 1660 I=1,NM
      DO 1660 J=1,NB
      BTREE(I,J)=0.0
      DO 1665 K=1,NM
1665 BTREE(I,J)=-ATREE(I,K)*ACHORD(K,J)+BTREE(I,J)
1660 CONTINUE
      IF(TR.EQ.0) GO TO 6000
      IF(CR.EQ.0) GO TO 6600
      GO TO 1644
C

```

```

C
6000 DO 6100 I=1,R
      IPV=I+V
6100 GCHORD(I)=1.0/RR(INDEX(IPV))
C
      DO 6200 I=1,NM
      VR(INDEX(I))=E(I)
      DO 6200 J=1,NB
6200  TBTREE(J,I)=BTREE(I,J)
C
      DO 6300 I=1,NB
      IPV=I+V
      VR(INDEX(IPV))=0.0
      DO 6225 J=1,NM
6225  VR(INDEX(IPV))=TBTREE(I,J)*E(J)+VR(INDEX(IPV))
      IF(I.LE.R) GO TO 6250
      GO TO 6300
6250  IR(INDEX(IPV))=GCHORD(I)*VR(INDEX(IPV))
6300  CONTINUE
      IF(C.EQ.0) GO TO 6450
      DO 6400 K=1,C
      KVR=K+V+R
6400  IR(INDEX(KVR))=JJ(KVR)
C
6450  DO 6500 J=1,NM
      IR(INDEX(J))=0
      DO 6500 I=1,NB
      IPV=I+V
6500  IR(INDEX(J))=BTREE(J,I)*IR(INDEX(IPV))+IR(INDEX(J))
      GO TO 10000
C
6600  DO 6650 I=1,R
      IPV=I+V
6650  RTREE(I)=RR(INDEX(IPV))
      DO 6700 K=1,NB
      KVR=K+V+R
      IR(INDEX(KVR))=JJ(KVR)
      DO 6700 I=1,NM
6700  TBTREE(K,I)=BTREE(I,K)
C
      DO 6800 I=1,NM
      IPV=I+V
      IR(INDEX(I))=0.0
      DO 6725 K=1,C
      KVR=K+V+R
6725  IR(INDEX(I))=BTREE(I,K)*JJ(KVR)+IR(INDEX(I))
      IF(I.GT.V) GO TO 6750
      GO TO 6800
6750  VR(INDEX(IPV))=RTREE(I-V)*IR(INDEX(IPV))
6800  CONTINUE
      IF(V.EQ.0) GO TO 6875
C

```

```

DO 6850 I=1,V
6850 VR(INDEX(I))=E(I)
C
6875 DO 6900 K=1,NB
      KVR=K+V+R
      VR(INDEX(KVR))=0.0
      DO 6900 J=1,NM
6900 VR(INDEX(KVR))=TBTREE(K,J)*VR(INDEX(J))+VR(INDEX(KVR))
      GO TO 10000
C
1644 DO 1700 I=1,NM
      DO 1700 J=1,NB
      A(I,J)=BTREE(I,J)
1700 CONTINUE
      DO 1705 I=1,NM
      DO 1705 J=1,NB
1705 BTREE(J,I)=A(I,J)
C
      IF(V.EQ.0) GO TO 1850
      DO 1800 I=1,CR
      DO 1800 K=1,V
      BRCE(I,K)=BTREE(I,K)
1800 CONTINUE
1850 DO 1900 I=1,CR
      DO 1900 K=1,TR
      JKV=K+V
      BRCRT(I,K)=BTREE(I,JKV)
1900 CONTINUE
      IF(C.EQ.0.0) GO TO 2197
      IF(V.EQ.0) GO TO 2001
      MCR=CR+1
      JCR=CR+C
      DO 2000 I=1,C
      DO 2000 K=1,V
      KP=I+CR
      BJE(I,K)=BTREE(KP,K)
2000 CONTINUE
2001 DO 2100 I=1,C
      DO 2100 K=1,TR
      IT=I+CR
      KV=K+V
      BJRT(I,K)=BTREE(IT,KV)
2100 CONTINUE
C
2197 DO 2200 I=1,TR
      IPV=I+V
      RTREE(I)=RR(INDEX(IPV))
      GTREE(I)=1.0/RR(INDEX(IPV))
2200 CONTINUE
      DO 2300 I=1,CR
      INX=I+NM
      RCHORD(I)=RR(INDEX(INX))

```

```

GCHORD(I)=1.0/RR(INDEX(INX))
2300 CONTINUE
C
IF(C.EQ.0) GO TO 2401
DO 2400 I=1,C
DO 2400 K=1,TR
2400 TBJRT(K,I)=BJRT(I,K)
C
2401 DO 2500 I=1,CR
DO 2500 K=1,TR
2500 TBJRT(K,I)=BRCRT(I,K)
C
C
X=RTREE*TBJRT
DO 2800 I=1,TR
DO 2800 J=1,CR
X(I,J)=0
DO 2900 K=1,TR
IF(I.NE.K) GO TO 2900
X(I,J)=RTREE(I)*TBJRT(K,J)+X(I,J)
2900 CONTINUE
2800 CONTINUE
C
C
XX=RTREE*TBJRT*GCHORD
DO 3000 I=1,TR
DO 3000 J=1,CR
XX(I,J)=0
DO 3100 K=1,CR
IF(J.NE.K) GO TO 3100
XX(I,J)=X(I,K)*GCHORD(K)+XX(I,J)
3100 CONTINUE
3000 CONTINUE
C
C
M31=MB
DO 3400 J=1,CR
DO 3400 I=1,CR
MB(I,J)=0
DO 3500 K=1,TR
3500 MB(I,J)=BRCRT(I,K)*XX(K,J)+MB(I,J)
3400 CONTINUE
DO 3600 K=1,CR
DO 3600 I=1,CR
IF(I.EQ.K) GO TO 95
MB(I,K)=MB(I,K)
GO TO 3600
95 MB(I,K)=MB(I,K)+1
3600 CONTINUE
DO 74 I=1,CR
DO 74 J=1,CR
NU=I+CR*(J-1)
74 AARRAY(NU)=MB(I,J)
CALL MINV(AARRAY,CR,D,L,M)
DO 76 I=1,CR

```

```

DO 76 J=1,CR
  NU=I+CR*(J-1)
76 MB(I,J)=AARRAY(NU)
C
  IF(C.EQ.0) GO TO 4750
DO 4200 I=1,TR
DO 4200 J=1,C
  XE(I,J)=0
DO 4300 K=1,TR
  IF(I.NE.K) GO TO 4300
  XE(I,J)=RTREE(K)*TBJRT(K,J)+XE(I,J)
4300 CONTINUE
4200 CONTINUE
C
DO 4400 I=1,CR
DO 4400 J=1,C
  XF(I,J)=0
DO 4500 K=1,TR
4500 XF(I,J)=BRCRT(I,K)*XE(K,J)+XF(I,J)
4400 CONTINUE
C
C
  XG=MB*XF
  IF(C.EQ.0) GO TO 4750
DO 4600 I=1,CR
DO 4600 J=1,C
  XG(I,J)=0
DO 4700 K=1,CR
4700 XG(I,J)=MB(I,K)*XF(K,J)+XG(I,J)
4600 CONTINUE
C
4750 IF(V.EQ.0) GO TO 4880
DO 4637 I=1,CR
DO 4637 J=1,V
  YE(I,J)=0.0
DO 4638 K=1,CR
4638 YE(I,J)=MB(I,K)*BRCE(K,J)+YE(I,J)
4637 CONTINUE
4880 DO 4800 I=1,CR
  SUM1(I)=0
  IF(C.EQ.0) GO TO 4901
DO 4900 K=1,C
  KRV=K+R+V
4900 SUM1(I)=XG(I,K)*JJ(KRV)+SUM1(I)
4901 SUM2(I)=0.0
  IF(V.EQ.0) GO TO 4870
DO 5000 K=1,V
5000 SUM2(I)=YE(I,K)*E(K)+SUM2(I)
4870 INX=I+NM
  VR(INDEX(INX))=-SUM1(I)-SUM2(I)
5400 IR(INDEX(INX))=GCHORD(I)*VR(INDEX(INX))
4800 CONTINUE
C
M68=MA

```



```

C      MA=RTREE*TBR CRT*GCHORD*BRCRT
      DO 3200 J=1,TR
      DO 3200 I=1,TR
      MA(I,J)=0
      DO 3300 K=1,CR
3300  MA(I,J)=XX(I,K)*BRCRT(K,J)+MA(I,J)
3200  CONTINUE
      DO 3700 K=1,TR
      DO 3700 I=1,TR
      IF(I.EQ.K) GO TO 85
      MA(I,K)=MA(I,K)
      GO TO 3700
85    MA(I,K)=MA(I,K)+1
3700  CONTINUE
      DO 49 I=1,TR
      DO 49 J=1,TR
      NU=I+TR*(J-1)
49    AARRAY(NU)=MA(I,J)
      CALL MINV(AARRAY,TR,D,L,M)
      DO 65 I=1,TR
      DO 65 J=1,TR
      NU=I+TR*(J-1)
65    MA(I,J)=AARRAY(NU)
C
C      XT(MA)-1*RTREE*TBR CRT*GCHORD
      DO 3800 I=1,TR
      DO 3800 J=1,CR
      XT(I,J)=0
      DO 3900 K=1,TR
3900  XT(I,J)=MA(I,K)*XX(K,J)+XT(I,J)
3800  CONTINUE
C
      IF(V.EQ.0) GO TO 4041
      DO 4000 I=1,TR
      DO 4000 J=1,V
      XD(I,J)=0
      DO 4100 K=1,CR
4100  XD(I,J)=XT(I,K)*BRCE(K,J)+XD(I,J)
4000  CONTINUE
C
C      YO=MA*XE
4041  IF(C.EQ.0) GO TO 5150
      DO 4037 I=1,TR
      DO 4037 J=1,C
      YO(I,J)=0.0
      DO 4039 K=1,TR
4039  YO(I,J)=MA(I,K)*XE(K,J)+YO(I,J)
4037  CONTINUE
C
5150  DO 5100 I=1,TR
      SUM3(I)=0.0
      IF(C.EQ.0) GO TO 5201

```

```

DO 5200 K=1,C
KRV=K+R+V
5200 SUM3(I)=YO(I,K)*JJ(KRV)+SUM3(I)
5201 SUM4(I)=0.0
IF(V.EQ.0) GO TO 5310
DO 5300 K=1,V
5300 SUM4(I)=XD(I,K)*E(K)+SUM4(I)
5310 IPV=I+V
VR(INDEX(IPV))=SUM3(I)-SUM4(I)
5500 IR(INDEX(IPV))=GTREE(I)*VR(INDEX(IPV))
5100 CONTINUE
C
C IF(C.EQ.0) GO TO 5480
DO 5430 K=1,C
SUM3(K)=0.0
DO 5430 I=1,TR
IPV=I+V
5430 SUM3(K)=-BJRT(K,I)*VR(INDEX(IPV))+SUM3(K)
DO 5450 I=1,C
SUM4(I)=0.0
IF(V.EQ.0) GO TO 5450
DO 5451 J=1,V
5451 SUM4(I)=-BJE(I,J)*E(J)+SUM4(I)
5450 CONTINUE
DO 5490 K=1,C
KRV=K+R+V
5490 VR(INDEX(KRV))=SUM3(K)+SUM4(K)
C
5480 IF(V.EQ.0) GO TO 5665
DO 5600 I=1,V
5600 VR(INDEX(I))=E(I)
C
IF(C.EQ.0) GO TO 5640
DO 5630 J=1,V
SUM1(J)=0.0
DO 5630 K=1,C
TBJE(J,K)=BJE(K,J)
KRV=K+R+V
5630 SUM1(J)=TBJE(J,K)*JJ(KRV)+SUM1(J)
5640 DO 5650 J=1,V
SUM2(J)=0.0
DO 5650 I=1,CR
TBRCE(J,I)=BRCE(I,J)
INX=I+NM
5650 SUM2(J)=TBRCE(J,I)*IR(INDEX(INX))+SUM2(J)
DO 5660 I=1,V
5660 IR(INDEX(I))=SUM1(I)+SUM2(I)
5665 IF(C.EQ.0) GO TO 10000
C
DO 5670 K=1,C
KRV=K+R+V
5670 IR(INDEX(KRV))=JJ(KRV)

```

```

GO TO 10000
C
9000 WRITE(6,9001)
9001 FORMAT(5X,'**HATA** G.KAYNAKLARINDAN OLUŞAN ÇEVİRİM VAR.')
```

GO TO 1102

```

9002 WRITE(6,9003)
9003 FORMAT(5X,'**HATA**DEVREDE ENAZ İKİ DÜĞÜM BULUNMALIDIR.'
1,/9X,'KONTROL EDİN.YALNIŞ GİRİŞ YAPILMIŞ OLABİLİR.')
```

GO TO 1102

```

9004 WRITE(6,9005)
9005 FORMAT(5X,'**HATA**DEVREDE ENAZ İKİ DAL BULUNMALIDIR.'
1,/9X,'KONTROL EDİN.YALNIŞ GİRİŞ YAPILMIŞ OLABİLİR.')
```

GO TO 1102

```

9006 WRITE(6,9007)
9007 FORMAT(5X,'**HATA** OMİK BİR DEVRE İÇİN DİRENÇ OLMALI.'
1,/9X,'GİRİŞİ KONTROL EDİN.')
```

GO TO 1102

```

9008 WRITE(6,9009)
9009 FORMAT(5X,'**HATA**DAL SAYISINA ORANLA FAZLA DÜĞÜM VAR.')
```

GO TO 1102

```

9010 WRITE(6,9011) I
9011 FORMAT(5X,'**HATA** DÜĞÜM',I3,1X,'BOŞTA KALMIŞ.')
```

GO TO 1102

```

9012 WRITE(6,9013)
9013 FORMAT('**HATA** PROGRAM,DEVRE ÜZERİNDE UYGUN AĞAC ',
1/9X,'BELİRLEYEMİYOR.GİRİŞİ KONTROL EDİN.')
```

GO TO 1102

```

9014 WRITE(6,9015)
9015 FORMAT(5X,'**HATA**YALNIZ AKIM KAYNAKLARININ BAĞLI',
1/9X,'OLDUĞU DÜĞÜM VAR.')
```

GO TO 1102

C

C

```

10000 KAL=0
KOF=1
WRITE(7,10001)
10001 FORMAT(6X,' NUMARA ',7X,'ELEMEN ÜZERİNDEKİ VOLT',
17X,'ELEMANDAN AKAN AKIM ')
```

C

```

11111 IF(KAL.EQ.B) GO TO 1102
IF(TYPE(KOF).EQ.'V') GO TO 11000
IF(TYPE(KOF).EQ.'R') GO TO 11002
IF(TYPE(KOF).EQ.'J') GO TO 11004
11000 KAL=KAL+1
WRITE(7,11001) KAL,VR(KOF),IR(KOF)
11001 FORMAT(1X,I3,3X,'( GER. KAYNAGI )',6X,1PE12.5,1X,
1'VOLT',10X,1PE12.5,1X,'AMP')
KOF=KOF+1
GO TO 11111
11002 KAL=KAL+1
WRITE(7,11003) KAL,VR(KOF),IR(KOF)
```

```

11003 FORMAT(1X,I3,3X,'( DİRENÇ )',12X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP')
      KOF=KOF+1
      GO TO 11111
11004 KAL=KAL+1
      WRITE(7,11005) KAL,VR(KOF),IR(KOF)
11005 FORMAT(1X,I3,3X,'( AKIM KAYNAĞI )',6X,1PE12.5,1X,
1'VOLT',10X,1PE12.5,1X,'AMP')
      KOF=KOF+1
      GO TO 11111

```

```

C
1102 WRITE(6,30008)
30008 FORMAT(/////15X,'BU, PROGRAMIN SONUDUR. ')
      GO TO 8
30000 CLOSE(UNIT=7)
      STOP
      END

```

C*****

```

      SUBROUTINE TRIANG(A,N,M,INDEX)
      DIMENSION A(20,50),INDEX(50)

```

```

C
C
      DO 1000 LL=1,N
      IMP=LL+1
100 IF( A(LL,LL).EQ.0.0) GO TO 400
C
      DO 300 I=IMP,N
      IF( A(I,LL).EQ.0.0) GO TO 300
      DUMMY=A(LL,LL)/A(I,LL)
      DO 200 K=LL,M
200 A(I,K)=DUMMY*A(I,K)-A(LL,K)
300 CONTINUE
      GO TO 1000
C
400 IF(LL.EQ.N) GO TO 800
      DO 700 I=IMP,N
      IF( A(I,LL).EQ.0.0) GO TO 600
      DO 500 K=LL,M
      TEMP=A(I,K)
      A(I,K)=A(LL,K)
      A(LL,K)=TEMP
500 CONTINUE
      GO TO 100
600 IF(I.EQ.N) GO TO 800
700 CONTINUE
C
800 IF(LL.EQ.M) GO TO 1000
      DO 900 J=IMP,M
      IF( A(LL,J).EQ.0.0) GO TO 900
      DO 850 IT=1,N
      TEMP=A(IT,J)

```

```

      A(IT,J)=A(IT,LL)
      A(IT,LL)=TEMP
C 850 CONTINUE
C
      KEEP=INDEX(LL)
      INDEX(LL)=INDEX(J)
      INDEX(J)=KEEP
      GO TO 100
C 900 CONTINUE
C 1000 CONTINUE
      RETURN
      END
C*****
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(2500),L(20),M(20)
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
10 IF( ABS(BIGA)-ABS(A(IJ))) 15,20,20
15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
20 CONTINUE
C
      J=L(K)
      IF(J-K) 35,35,25
25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
30 A(JI)=HOLD
C
35 I=M(K)
      IF(I-K) 45,45,38
38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)

```

```

      A(JK)=A(JI)
C 40 A(JI)=HOLD
      45 IF(BIGA) 48,46,48
      46 D=0.0
      RETURN
      48 DO 55 I=1,N
      IF(I-K) 50,55,50
      50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
      55 CONTINUE
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
      60 IF(J-K) 62,65,62
      62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
      65 CONTINUE
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
      70 A(KJ)=A(KJ)/BIGA
      75 CONTINUE
C
      D=D*BIGA
C
      A(KK)=1.0/BIGA
      80 CONTINUE
C
      K=N
      100 K=K-1
      IF(K) 150,150,105
      105 I=L(K)
      IF(I-K) 120,120,108
      108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
      110 A(JI)=HOLD
      120 J=M(K)
      IF(J-K) 100,100,125
      125 KI=K-N

```

```
DO 130 I=1,N
KI=KI+N
HOLD=A(KI)
JI=KI-K+J
A(KI)=-A(JI)
130 A(JI)=HOLD
GO TO 100
150 RETURN
END
```



EK-B:L,C ELEMANLARIDA İÇEREN DEVRELER İÇİN PROGRAM

INTEGER N,B,R,V,C,T,GROUND,CR,TR,VLT,CC,RL,RC,SYC1,SYC2
1,SYC3,SYC4,D,DNM

REAL JJ,MA,MB,IR,JJJ

DIMENSION ALPHA(9)

DIMENSION TBTREE(49,49),JJJ(50),NDI(30),NDJ(30)

DIMENSION APRIME(20,50),NDEI(20),NDEJ(20),VODD(20)

DIMENSION FMT(7),VL(30),DNODEI(30),DNODEJ(30)

DIMENSION TBJE(20,20),TBRCE(20,49),VO(20)

DIMENSION YE(49,49),YO(49,49),L(20),M(20),VOD(50)

DIMENSION A(20,50),ATREE(19,49),ACHORD(19,49),BTREE(49,49),

1 AARRAY(2500),NODEI(50),NODEJ(50),RR(50),E(50),JJ(50),

2 INDEX(50),RTREE(50),RCHORD(50),GTREE(50),GCHORD(500),VR(50),

3 IR(50),TEMP(50),X(19,50),XX(19,49),MA(19,49),MB(19,49),

4 XT(19,49),XD(19,49),XE(19,49),XF(49,49),XG(49,49),BJE(19,49)

5 ,BRCE(19,49),BRCRT(19,49),BJRT(19,49),SUM(50),SUM1(50),

6 SUM2(50),SUM3(50),SUM4(50),TBRCRT(49,49),TBJRT(49,49)

CHARACTER*1 TYPE(50),COMP,DTYPE(50),TYP(50)

OPEN(UNIT=7,FILE='SONUC2',STATUS='NEW')

C

8 WRITE(6,*) ' SIRAYA UYUN : 1.GERİLİM KAYNAKLARI '

WRITE(6,*) ' 2.DİRENÇLER '

WRITE(6,*) ' 3.SIGALAR '

WRITE(6,*) ' 4.İNDÜKTANSLAR '

WRITE(6,*) ' 5.AKIM KAYNAKLARI '

EPSI=1.E-12

R=0

C=0

B=0

V=0

D=0

RC=0

RL=0

CC=1

KAT=1

VLT=1

DNM=1

WRITE(6,*) 'DEVREDE SIĞA/İNDÜKTANS VARMI(Y/N)?'

READ(5,101) COMP

101

FORMAT(A1)

IF(COMP.EQ.'Y') THEN

WRITE(6,*) 'ZAMAN ARALIĞI(DT) TOPLAM ZAMAN(TIME)'

READ*,DT,TIME

SAYAC=DT

END IF

C

K=1

100

WRITE(6,*) 'TYPE '

READ(5,102) TYPE(K)

102

FORMAT(A1)

IF(TYPE(K).EQ.'Z') GO TO 159

WRITE(6,*) 'NODEI NODEJ VALUE'


```

159 READ*,NODEI(K),NODEJ(K),VAL
    IF(TYPE(K).EQ.'V') THEN
      V=V+1
      B=B+1
      E(K)=VAL
    ELSE IF (TYPE(K).EQ.'R') THEN
      R=R+1
      RR(K)=VAL
      B=B+1
    ELSE IF (TYPE(K).EQ.'C') THEN
      WRITE(6,*)'BASLANGIÇ DEGERİ NEDİR?'
      READ*,VO(K)
      B=B+2
      RC=RC+1
      R=R+1
      RR(K)=DT/VAL
      DTYPE(CC)='C'
      JJJ(CC)=-1.*VO(K)*(VAL/DT)
      DNODEI(CC)=NODEI(K)
      DNODEJ(CC)=NODEJ(K)
      CC=CC+1
      C=CC-1
    ELSE IF (TYPE(K).EQ.'L') THEN
      WRITE(6,*)'BASLANGIÇ DEGERİ NEDİR?'
      READ*,JJJ(CC)
      VO(K)=JJJ(CC)
      B=B+2
      RL=RL+1
      R=R+1
      RR(K)=VAL/DT
      DTYPE(CC)='L'
      DNODEI(CC)=NODEI(K)
      DNODEJ(CC)=NODEJ(K)
      CC=CC+1
      C=CC-1
    ELSE IF (TYPE(K).EQ.'J') THEN
      B=B+1
      C=C+1
      VL(KAT)=VAL
    ELSE IF (TYPE(K).EQ.'Z') THEN
      GO TO 152
    ELSE
      GO TO 9012
    END IF
    K=K+1
    GO TO 100
152 IF(COMP.EQ.'Y') THEN
      CC=CC-1
      K=K-(KAT-1)
      DO 60606 I=1,CC
        TYPE(K)=DTYPE(I)
        JJ(K)=JJJ(I)

```

```

        NODEI(K)=DNODEI(I)
        NODEJ(K)=DNODEJ(I)
60606  K=K+1
        DO 70707 I=1,(KAT-1)
        TYPE(K)=TYP(I)
        JJ(K)=VL(I)
        NODEI(K)=NDI(I)
        NODEJ(K)=NDJ(I)
        VOD(K)=VODD(I)
70707  K=K+1
        END IF
C
C      WRITE(6,*) '*****'
C
        N=0
        DO 157 K=1,B
        IF(N.GT.NODEI(K)) THEN
        N=N
        ELSE
        N=NODEI(K)
        END IF
        IF(N.GT.NODEJ(K)) THEN
        N=N
        ELSE
        N=NODEJ(K)
        END IF
157    CONTINUE
        IF(N.EQ.0) GO TO 30000
        WRITE(6,*) 'TOPRAK?'
        READ*,GROUND
        IF(V.EQ.0) GO TO 2
1      IF(C.EQ.0) GO TO 4
        GO TO 6
2      WRITE(6,3)
3      FORMAT(15X,'HIÇ GERİLİM KAYNAĞI YOK')
        GO TO 1
4      WRITE(6,5)
5      FORMAT(15X,'HIÇ AKIM KAYNAĞI YOK')
6      IF(N.LT.2) GO TO 9002
        IF(B.LT.2) GO TO 9004
        IF(R.EQ.0) GO TO 9006
        IF(B.LT.N) GO TO 9008
        NM=N-1
        NB=B-(N-1)
        TR=NM-V
        KV=V+1
        CR=R-(NM-V)
        NP=N+1
        KVR=V+R
C
        WRITE(7,*) 'DÜĞÜMLER G.KAYNAKLARI DİRENÇLER A.KAYNAKLARI'
        DO 153 K=1,B

```

```

IF (TYPE(K).EQ.'V') THEN
WRITE(7,154) NODEI(K),NODEJ(K),E(K)
154 FORMAT(1X,I3,'-',I3,1X,F12.3)
ELSE IF (TYPE(K).EQ.'R') THEN
WRITE(7,155) NODEI(K),NODEJ(K),RR(K)
155 FORMAT(1X,I3,'-',I3,11X,F12.3)
ELSE IF (TYPE(K).EQ.'J') THEN
WRITE(7,158) NODEI(K),NODEJ(K),JJ(K)
158 FORMAT(1X,I3,'-',I3,24X,F12.3)
END IF
153 CONTINUE

```

C

```

MRC=RC
MRL=RL
WRITE(7,*) 'DÜĞÜM          SİĞALAR      İNDÜKTANSLAR      İLK          '
WRITE(7,*) '                                           DEĞER          '
DO 193 K=1,B
IF (TYPE(K).EQ.'C') THEN
IF (MRC.EQ.0) GO TO 193
VOM=DT/RR(K)
81 WRITE(7,81) NODEI(K),NODEJ(K),VOM,VO(K)
FORMAT(1X,I3,'-',I2,2X,F12.10,15X,F4.2)
MRC=MRC-1
ELSE IF (TYPE(K).EQ.'L') THEN
IF (MRL.EQ.0) GO TO 193
VOM=RR(K)*DT
83 WRITE(7,83) NODEI(K),NODEJ(K),VOM,VO(K)
FORMAT(1X,I3,'-',I2,14X,F12.7,3X,F4.2)
MRL=MRL-1
END IF
193 CONTINUE
WRITE(7,*) '*****'
WRITE(7,161)N,B,R,V,C,RC,RL
161 FORMAT(2X,'N=',I2,2X,'B=',I2,2X,'R=',I2,2X,'V=',I2,2X,'J=',I2,
12X,'SİGA=',I2,2X,'İND=',I2)

```

C

```

KASK=V+R+1
KAPK=V+R+RC+1
KAJK=V+R+RC+RL+1
KSSK=KASK
KPPK=KAPK
KKK=KASK
KLL=KAPK

```

C

C*****

C

C

INCIDENCE MATRİS A(I,K)'NİN HAZIRLANMASI

C

321

```

DO 300 K=1,B
INDEX(K)=K
DO 300 I=1,N
IF (NODEI(K).EQ.I) GO TO 10

```

```

        IF(NODEJ(K).EQ.I) GO TO 20
        A(I,K)=0.0
        GO TO 300
10     A(I,K)=1.0
        GO TO 300
20     A(I,K)=-1.0
300    CONTINUE
C
        DO 360 I=1,N
        CHECK=0.0
        DO 350 K=1,B
        IF(A(I,K).EQ.EPSI) GO TO 350
        CHECK=CHECK+1.0
350    CONTINUE
        IF(CHECK.LT.2.0) GO TO 9010
360    CONTINUE
C
        IF(GROUND.NE.0) GO TO 30
        BIG=0.0
        DO 400 I=1,N
        SUM(I)=0.0
        DO 500 K=1,B
500    SUM(I)=ABS(A(I,K))+SUM(I)
        IF(SUM(I).LE.BIG) GO TO 400
        BIG=SUM(I)
        GROUND=I
400    CONTINUE
        WRITE(7,987) GROUND
987    FORMAT(5X,'GROUND=',I3)
C
        DO 30 K=1,B
        DO 600 I=1,NM
        IF(I.GE.GROUND) GO TO 35
        A(I,K)=A(I,K)
        GO TO 600
35     IP=I+1
        A(I,K)=A(IP,K)
600    CONTINUE
        DO 650 I=1,NM
        DO 650 J=1,B
650    APRIME(I,J)=A(I,J)
        CALL TRIANG(A,NM,B,INDEX)
C
        KVR=V+R
        DO 805 K=1,NM
        IF(INDEX(K).GT.KVR) GO TO 9014
805    CONTINUE
        DO 810 K=N,B
        IF(INDEX(K).LE.V) GO TO 9000
810    CONTINUE
C
        DO 820 J=1,NM

```

```

      IF(A(J,J).EQ.EPSI) GO TO 9012
820  CONTINUE
C
      DO 800 K=1,B
      DO 800 I=1,NM
800  A(I,K)=APRIME(I,INDEX(K))
C
      DO 69 K=1,NM
      DO 69 I=1,NM
      ATREE(I,K)=A(I,K)
-----
69  CONTINUE
      DO 1600 K=1,NB
      DO 1600 I=1,NM
      KH=K+NM
      ACHORD(I,K)=A(I,KH)
1600 CONTINUE
C
      DO 88 J=1,NM
      DO 88 I=1,NM
      NU=I+NM*(J-1)
      AARRAY(NU)=ATREE(I,J)
88  CONTINUE
      CALL MINV(AARRAY,NM,D,L,M)
      DO 25 J=1,NM
      DO 25 I=1,NM
      NU=I+NM*(J-1)
      ATREE(I,J)=AARRAY(NU)
25  CONTINUE
C
      DO 1660 I=1,NM
      DO 1660 J=1,NB
      BTREE(I,J)=0.0
      DO 1665 K=1,NM
1665 BTREE(I,J)=-ATREE(I,K)*ACHORD(K,J)+BTREE(I,J)
1660 CONTINUE
      IF(TR.EQ.0) GO TO 6000
      IF(CR.EQ.0) GO TO 6600
      GO TO 1644
C
6000 DO 6100 I=1,R
      IPV=I+V
6100 GCHORD(I)=1.0/RR(INDEX(IPV))
C
      DO 6200 I=1,NM
      VR(INDEX(I))=E(I)
      DO 6200 J=1,NB
6200 TBTREE(J,I)=BTREE(I,J)
C
      DO 6300 I=1,NB
      IPV=I+V
      VR(INDEX(IPV))=0.0
      DO 6225 J=1,NM

```

```

6225 VR(INDEX(IPV))=TBTREE(I,J)*E(J)+VR(INDEX(IPV))
      IF(I.LE.R) GO TO 6250
      GO TO 6300
6250 IR(INDEX(IPV))=GCHORD(I)*VR(INDEX(IPV))
6300 CONTINUE
      IF(C.EQ.0) GO TO 6450
      DO 6400 K=1,C
      KVR=K+V+R
6400 IR(INDEX(KVR))=JJ(KVR)
C
6450 DO 6500 J=1,NM
      IR(INDEX(J))=0
      DO 6500 I=1,NB
      IPV=I+V
6500 IR(INDEX(J))=BTREE(J,I)*IR(INDEX(IPV))+IR(INDEX(J))
      GO TO 10000
C
C
6600 DO 6650 I=1,R
      IPV=I+V
6650 RTREE(I)=RR(INDEX(IPV))
      DO 6700 K=1,NB
      KVR=K+V+R
      IR(INDEX(KVR))=JJ(KVR)
      DO 6700 I=1,NM
6700 TBTREE(K,I)=BTREE(I,K)
C
      DO 6800 I=1,NM
      IPV=I+V
      IR(INDEX(I))=0.0
      DO 6725 K=1,C
      KVR=K+V+R
6725 IR(INDEX(I))=BTREE(I,K)*JJ(KVR)+IR(INDEX(I))
      IF(I.GT.V) GO TO 6750
      GO TO 6800
6750 VR(INDEX(IPV))=RTREE(I-V)*IR(INDEX(IPV))
6800 CONTINUE
      IF(V.EQ.0) GO TO 6875
C
      DO 6850 I=1,V
6850 VR(INDEX(I))=E(I)
C
6875 DO 6900 K=1,NB
      KVR=K+V+R
      VR(INDEX(KVR))=0.0
      DO 6900 J=1,NM
6900 VR(INDEX(KVR))=TBTREE(K,J)*VR(INDEX(J))+VR(INDEX(KVR))
      GO TO 10000
C
1644 DO 1700 I=1,NM
      DO 1700 J=1,NB
      A(I,J)=BTREE(I,J)

```

```

1700 CONTINUE
      DO 1705 I=1,NM
      DO 1705 J=1,NB
1705 BTREE(J,I)=A(I,J)
C
      IF(V.EQ.0) GO TO 1850
      DO 1800 I=1,CR
      DO 1800 K=1,V
      BRCE(I,K)=BTREE(I,K)
1800 CONTINUE
1850 DO 1900 I=1,CR
      DO 1900 K=1,TR
      JKV=K+V
      BRCRT(I,K)=BTREE(I,JKV)
1900 CONTINUE
      IF(C.EQ.0.0) GO TO 2197
      IF(V.EQ.0) GO TO 2001
      MCR=CR+1
      JCR=CR+C
      DO 2000 I=1,C
      DO 2000 K=1,V
      KP=I+CR
      BJE(I,K)=BTREE(KP,K)
2000 CONTINUE
2001 DO 2100 I=1,C
      DO 2100 K=1,TR
      IT=I+CR
      KV=K+V
      BJRT(I,K)=BTREE(IT,KV)
2100 CONTINUE
C
2197 DO 2200 I=1,TR
      IPV=I+V
      RTREE(I)=RR(INDEX(IPV))
      GTREE(I)=1.0/RR(INDEX(IPV))
2200 CONTINUE
      DO 2300 I=1,CR
      INX=I+NM
      RCHORD(I)=RR(INDEX(INX))
      GCHORD(I)=1.0/RR(INDEX(INX))
2300 CONTINUE
C
      IF(C.EQ.0) GO TO 2401
      DO 2400 I=1,C
      DO 2400 K=1,TR
2400 TBJRT(K,I)=BJRT(I,K)
C
2401 DO 2500 I=1,CR
      DO 2500 K=1,TR
2500 TBRCRT(K,I)=BRCRT(I,K)
C
C V3=(1+BRCRT*RTREE*TBRCRT*GCHORD)-(BRCRT*RTREE*TBJRT)JJ-(BRCE)E

```

```

C V6=(1+RTREE*TBR CRT*GCHORD*BRCRT)-(RTREE*TBJRT)JJ
C   -(RTREE*TBR CRT*GCHORD)
C     X=RTREE*TBR CRT
      DO 2800 I=1,TR
      DO 2800 J=1,CR
      X(I,J)=0
      DO 2900 K=1,TR
      IF(I.NE.K) GO TO 2900
      X(I,J)=RTREE(I)*TBR CRT(K,J)+X(I,J)
2900 CONTINUE
2800 CONTINUE
C
C     XX=RTREE*TBR CRT*GCHORD
      DO 3000 I=1,TR
      DO 3000 J=1,CR
      XX(I,J)=0
      DO 3100 K=1,CR
      IF(J.NE.K) GO TO 3100
      XX(I,J)=X(I,K)*GCHORD(K)+XX(I,J)
3100 CONTINUE
3000 CONTINUE
C
C     M31=MB
      DO 3400 J=1,CR
      DO 3400 I=1,CR
      MB(I,J)=0
      DO 3500 K=1,TR
3500 MB(I,J)=BRCRT(I,K)*XX(K,J)+MB(I,J)
3400 CONTINUE
      DO 3600 K=1,CR
      DO 3600 I=1,CR
      IF(I.EQ.K) GO TO 95
      MB(I,K)=MB(I,K)
      GO TO 3600
95 MB(I,K)=MB(I,K)+1
3600 CONTINUE
      DO 74 I=1,CR
      DO 74 J=1,CR
      NU=I+CR*(J-1)
74 AARRAY(NU)=MB(I,J)
      CALL MINV(AARRAY,CR,D,L,M)
      DO 76 I=1,CR
      DO 76 J=1,CR
      NU=I+CR*(J-1)
76 MB(I,J)=AARRAY(NU)
C
      IF(C.EQ.0) GO TO 4750
      DO 4200 I=1,TR
      DO 4200 J=1,C
      XE(I,J)=0
      DO 4300 K=1,TR
      IF(I.NE.K) GO TO 4300

```



```

      XE(I,J)=RTREE(K)*TBJRT(K,J)+XE(I,J)
4300 CONTINUE
4200 CONTINUE
C
      DO 4400 I=1,CR
      DO 4400 J=1,C
      XF(I,J)=0
      DO 4500 K=1,TR
4500 XF(I,J)=BRCRT(I,K)*XE(K,J)+XF(I,J)
4400 CONTINUE
-----
C
C      XG=MB*XF
      IF(C.EQ.0) GO TO 4750
      DO 4600 I=1,CR
      DO 4600 J=1,C
      XG(I,J)=0
      DO 4700 K=1,CR
4700 XG(I,J)=MB(I,K)*XF(K,J)+XG(I,J)
4600 CONTINUE
C
4750 IF(V.EQ.0) GO TO 4880
      DO 4637 I=1,CR
      DO 4637 J=1,V
      YE(I,J)=0.0
      DO 4638 K=1,CR
4638 YE(I,J)=MB(I,K)*BRCE(K,J)+YE(I,J)
4637 CONTINUE
4880 DO 4800 I=1,CR
      SUM1(I)=0
      IF(C.EQ.0) GO TO 4901
      DO 4900 K=1,C
      KRV=K+R+V
4900 SUM1(I)=XG(I,K)*JJ(KRV)+SUM1(I)
4901 SUM2(I)=0.0
      IF(V.EQ.0) GO TO 4870
      DO 5000 K=1,V
5000 SUM2(I)=YE(I,K)*E(K)+SUM2(I)
4870 INX=I+NM
      VR(INDEX(INX))=-SUM1(I)-SUM2(I)
5400 IR(INDEX(INX))=GCHORD(I)*VR(INDEX(INX))
4800 CONTINUE
C
      M68=MA
C
      MA=RTREE*TBRCRT*GCHORD*BRCRT
      DO 3200 J=1,TR
      DO 3200 I=1,TR
      MA(I,J)=0
      DO 3300 K=1,CR
3300 MA(I,J)=XX(I,K)*BRCRT(K,J)+MA(I,J)
3200 CONTINUE
      DO 3700 K=1,TR
      DO 3700 I=1,TR
      IF(I.EQ.K) GO TO 85

```

```

      MA(I,K)=MA(I,K)
      GO TO 3700
85    MA(I,K)=MA(I,K)+1
3700  CONTINUE
      DO 49 I=1,TR
      DO 49 J=1,TR
      NU=I+TR*(J-1)
49    AARRAY(NU)=MA(I,J)
      CALL MINV(AARRAY,TR,D,L,M)
      DO 65 I=1,TR
      DO 65 J=1,TR
      NU=I+TR*(J-1)
65    MA(I,J)=AARRAY(NU)
C
C    XT(MA)-1*RTREE*TBR CRT*GCHORD
      DO 3800 I=1,TR
      DO 3800 J=1,CR
      XT(I,J)=0
      DO 3900 K=1,TR
3900  XT(I,J)=MA(I,K)*XX(K,J)+XT(I,J)
3800  CONTINUE
C
      IF(V.EQ.0) GO TO 4041
      DO 4000 I=1,TR
      DO 4000 J=1,V
      XD(I,J)=0
      DO 4100 K=1,CR
4100  XD(I,J)=XT(I,K)*BRCE(K,J)+XD(I,J)
4000  CONTINUE
C    YO=MA*XE
4041  IF(C.EQ.0) GO TO 5150
      DO 4037 I=1,TR
      DO 4037 J=1,C
      YO(I,J)=0.0
      DO 4039 K=1,TR
4039  YO(I,J)=MA(I,K)*XE(K,J)+YO(I,J)
4037  CONTINUE
C
5150  DO 5100 I=1,TR
      SUM3(I)=0.0
      IF(C.EQ.0) GO TO 5201
      DO 5200 K=1,C
      KRV=K+R+V
5200  SUM3(I)=YO(I,K)*JJ(KRV)+SUM3(I)
5201  SUM4(I)=0.0
      IF(V.EQ.0) GO TO 5310
      DO 5300 K=1,V
5300  SUM4(I)=XD(I,K)*E(K)+SUM4(I)
5310  IPV=I+V
      VR(INDEX(IPV))=SUM3(I)-SUM4(I)
5500  IR(INDEX(IPV))=GTREE(I)*VR(INDEX(IPV))
5100  CONTINUE

```

```

C
C   IF(C.EQ.0) GO TO 5480
      DO 5430 K=1,C
      SUM3(K)=0.0
      DO 5430 I=1,TR
      IPV=I+V
5430  SUM3(K)=-BJRT(K,I)*VR(INDEX(IPV))+SUM3(K)
      DO 5450 I=1,C
      SUM4(I)=0.0
      IF (V.EQ.0) GO TO 5450
      DO 5451 J=1,V
5451  SUM4(I)=-BJE(I,J)*E(J)+SUM4(I)
5450  CONTINUE
      DO 5490 K=1,C
      KRV=K+R+V
5490  VR(INDEX(KRV))=SUM3(K)+SUM4(K)
C
5480  IF(V.EQ.0) GO TO 5665
      DO 5600 I=1,V
5600  VR(INDEX(I))=E(I)
C
      IF(C.EQ.0) GO TO 5640
      DO 5630 J=1,V
      SUM1(J)=0.0
      DO 5630 K=1,C
      TBJE(J,K)=BJE(K,J)
      KRV=K+R+V
5630  SUM1(J)=TBJE(J,K)*JJ(KRV)+SUM1(J)
5640  DO 5650 J=1,V
      SUM2(J)=0.0
      DO 5650 I=1,CR
      TBRCE(J,I)=BRCE(I,J)
      INX=I+NM
5650  SUM2(J)=TBRCE(J,I)*IR(INDEX(INX))+SUM2(J)
      DO 5660 I=1,V
5660  IR(INDEX(I))=SUM1(I)+SUM2(I)
5665  IF(C.EQ.0) GO TO 10000
C
      DO 5670 K=1,C
      KRV=K+R+V
5670  IR(INDEX(KRV))=JJ(KRV)
      GO TO 10000
C
9000  WRITE(6,9001)
9001  FORMAT(5X,'**HATA** G.KAYNAKLARINDAN OLUŞAN ÇEVİRİM VAR')
      GO TO 1102
9002  WRITE(6,9003)
9003  FORMAT(5X,'**HATA** DEVREDE ENAZ İKİ DÜĞÜM BULUNMALIDIR, '
1, /9X, 'KONTROL EDİN.YALNIŞ GİRİŞ YAPILMIŞ OLABİLİR. ')
      GO TO 1102
9004  WRITE(6,9005)
9005  FORMAT(5X,'**HATA** DEVREDE ENAZ İKİ DAL BULUNMALIDIR. '

```

```

1./9X, 'KONTROL EDİN. YALNİŞ GİRİŞ YAPILMIŞ OLABİLİR.')
GO TO 1102
9006 WRITE(6,9007)
9007 FORMAT(5X, '**HATA** OMİK BİR DEVRE İÇİN DİRENÇ OLMALI.'
1./9X, 'GİRİŞİ KONTROL EDİN.')
GO TO 1102
9008 WRITE(6,9009)
9009 FORMAT(5X, '**HATA**DAL SAYISINA ORANLA FAZLA DÜĞÜM VAR.')
```

```

GO TO 1102
9010 WRITE(6,9011) I
9011 FORMAT(5X, '**HATA** DÜĞÜM', I3, 1X, 'BOŞTA KALMIŞ')
```

```

GO TO 1102
9012 WRITE(6,9013)
9013 FORMAT('**HATA** PROGRAM UYGUN AĞAÇ BELİRLEYEMİYOR.',
1/9X, 'GİRİŞİ KONTROL EDİN.')
GO TO 1102
9014 WRITE(6,9015)
9015 FORMAT(5X, '**HATA** YALNIZ A.KAYNAKLARININ BAĞLI',
1/9X, 'OLDUĞU DÜĞÜM VAR.')
```

```

GO TO 1102
C
10000 KASK=KKK
KAPK=KLL
SYC1=RC
SYC2=RL
SYC3=RC
SYC4=RL
IF (COMP.EQ. 'Y') THEN
IF (SAYAC.GT.TIME) GO TO 90999
DO 41 I=1,B
IF (TYPE(I).EQ. 'C') THEN
IF (SYC1.EQ.0) GO TO 41
JJ(KASK)=-VR(I)*(1./RR(I))
KASK=KASK+1
SYC1=SYC1-1
ELSE IF (TYPE(I).EQ. 'L') THEN
IF (SYC2.EQ.0) GO TO 41
JJ(KAPK)=IR(I)+JJ(KAPK)
KAPK=KAPK+1
SYC2=SYC2-1
END IF
41 CONTINUE
SAYAC=SAYAC+DT
GO TO 321
END IF
C
90999 B=B-(RC+RL)
KAL=0
KOF=1
WRITE(7,10001)
10001 FORMAT(6X, ' NUMARA ', 7X, 'ELEMEN ÜZERİNDEKİ VOLT',
17X, 'ELEMANDAN AKAN AKIM ')

```

```

C
11111 IF(KAL.EQ.B) GO TO 1102
      IF(TYPE(KOF).EQ.'V') GO TO 11000
      IF(TYPE(KOF).EQ.'R') GO TO 11002
      IF(TYPE(KOF).EQ.'C') GO TO 20202
      IF(TYPE(KOF).EQ.'L') GO TO 30303
      IF(TYPE(KOF).EQ.'J') GO TO 11004
11000 KAL=KAL+1
      WRITE(7,11001) KAL,VR(KOF),IR(KOF)
11001 FORMAT(1X,I3,3X,'(GER. KAYNAGI )',6X,1PE12.5,1X,
1 'VOLT',10X,1PE12.5,1X,'AMP')
      KOF=KOF+1
      GO TO 11111
11002 KAL=KAL+1
      WRITE(7,11003) KAL,VR(KOF),IR(KOF)
11003 FORMAT(1X,I3,3X,'( DİRENC )',12X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP')
      KOF=KOF+1
      GO TO 11111
20202 IF(SYC3.EQ.0) GO TO 54321
      KAL=KAL+1
      IR(KOF)=IR(KOF)+IR(KSSK)
      WRITE(7,20201) KAL,VR(KOF),IR(KOF)
20201 FORMAT(1X,I3,3X,'( SİGA )',11X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP')
      KSSK=KSSK+1
      SYC3=SYC3-1
54321 KOF=KOF+1
      GO TO 11111
30303 IF(SYC4.EQ.0) GO TO 76543
      KAL=KAL+1
      IR(KOF)=IR(KOF)+IR(KPPK)
      WRITE(7,30301) KAL,VR(KOF),IR(KOF)
30301 FORMAT(1X,I3,3X,'( İNDÜKTANS )',12X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP')
      KPPK=KPPK+1
      SYC4=SYC4-1
76543 KOF=KOF+1
      GO TO 11111
11004 KAL=KAL+1
      WRITE(7,11005) KAL,VR(KAJK),IR(KAJK)
11005 FORMAT(1X,I3,3X,'( AKIM KAYNAGI )',6X,1PE12.5,1X,
1 'VOLT',10X,1PE12.5,1X,'AMP')
      KAJK=KAJK+1
      KOF=KOF+1
      GO TO 11111
C
1102 WRITE(6,30008)
30008 FORMAT(/////15X,'BU, PROGRAMIN SONUDUR.')
      GO TO 8
30000 CLOSE(UNIT=7)

```

```

      STOP
      END
C*****
      SUBROUTINE TRIANG(A,N,M,INDEX)
      DIMENSION A(20,50),INDEX(50)
C
C
      DO 1000 LL=1,N
      IMP=LL+1
100 IF( A(LL,LL).EQ.0.0) GO TO 400
C
      DO 300 I=IMP,N
      IF( A(I,LL).EQ.0.0) GO TO 300
      DUMMY=A(LL,LL)/A(I,LL)
      DO 200 K=LL,M
200 A(I,K)=DUMMY*A(I,K)-A(LL,K)
300 CONTINUE
      GO TO 1000
C
400 IF(LL.EQ.N) GO TO 800
      DO 700 I=IMP,N
      IF (A(I,LL).EQ.0.0) GO TO 600
      DO 500 K=LL,M
      TEMP=A(I,K)
      A(I,K)=A(LL,K)
      A(LL,K)=TEMP
500 CONTINUE
      GO TO 100
600 IF(I.EQ.N) GO TO 800
700 CONTINUE
C
800 IF(LL.EQ.M) GO TO 1000
      DO 900 J=IMP,M
      IF( A(LL,J).EQ.0.0) GO TO 900
      DO 850 IT=1,N
      TEMP=A(IT,J)
      A(IT,J)=A(IT,LL)
      A(IT,LL)=TEMP
850 CONTINUE
C
      KEEP=INDEX(LL)
      INDEX(LL)=INDEX(J)
      INDEX(J)=KEEP
      GO TO 100
900 CONTINUE
1000 CONTINUE
      RETURN
      END
C*****

```

```

SUBROUTINE MINV(A,N,D,L,M)
DIMENSION A(2500),L(20),M(20)
C
D=1.0
NK=-N
DO 80 K=1,N
NK=NK+N
L(K)=K
M(K)=K
KK=NK+K
BIGA=A(KK)
DO 20 J=K,N
IZ=N*(J-1)
DO 20 I=K,N
IJ=IZ+I
10 IF(ABS(BIGA)-ABS(A(IJ))) 15,20,20
15 BIGA=A(IJ)
L(K)=I
M(K)=J
20 CONTINUE
C
J=L(K)
IF(J-K) 35,35,25
25 KI=K-N
DO 30 I=1,N
KI=KI+N
HOLD=-A(KI)
JI=KI-K+J
A(KI)=A(JI)
30 A(JI)=HOLD
C
35 I=M(K)
IF(I-K) 45,45,38
38 JP=N*(I-1)
DO 40 J=1,N
JK=NK+J
JI=JP+J
HOLD=-A(JK)
A(JK)=A(JI)
40 A(JI)=HOLD
C
45 IF(BIGA) 48,46,48
46 D=0.0
RETURN
48 DO 55 I=1,N
IF(I-K) 50,55,50
50 IK=NK+I
A(IK)=A(IK)/(-BIGA)
55 CONTINUE
C
DO 65 I=1,N
IK=NK+I

```

```

HOLD=A(IK)
IJ=I-N
DO 65 J=1,N
IJ=IJ+N
IF(I-K) 60,65,60
60 IF(J-K) 62,65,62
62 KJ=IJ-I+K
A(IJ)=HOLD*A(KJ)+A(IJ)
65 CONTINUE

```

C

```

KJ=K-N
DO 75 J=1,N
KJ=KJ+N
IF(J-K) 70,75,70
70 A(KJ)=A(KJ)/BIGA
75 CONTINUE

```

C

```
D=D*BIGA
```

C

```

A(KK)=1.0/BIGA
80 CONTINUE

```

C

```

K=N
100 K=K-1
IF(K) 150,150,105
105 I=L(K)
IF(I-K) 120,120,108
108 JQ=N*(K-1)
JR=N*(I-1)
DO 110 J=1,N
JK=JQ+J
HOLD=A(JK)
JI=JR+J
A(JK)=-A(JI)
110 A(JI)=HOLD
120 J=M(K)
IF(J-K) 100,100,125
125 KI=K-N
DO 130 I=1,N
KI=KI+N
HOLD=A(KI)
JI=KI-K+J
A(KI)=-A(JI)
130 A(JI)=HOLD
GO TO 100
150 RETURN
END

```


EK-C:DIYOT-TRİSTÖR ELEMANLARIDA İÇEREN DEVRELER İÇİN PROGRAM.

C
C

PROGRAMIN KAREKTER VERİLERİ BÜYÜK HARFLE GİRİLMELİDİR !

```
include 'sys$library:gksdefs.for'  
integer ws_id,num_points  
INTEGER N,B,R,V,C,T,GROUND,CR,TR,VLT,CC,RL,RC,SYC1,SYC2  
INTEGER SYC3,SYC4,D,DNM,CY  
REAL JJ,MA,MB,IR,JJJ,IRIND,IRCAP  
DIMENSION ALPHA(9),OMEGA(50),TPD(50),TPDJ(50),OMEGAJ(50)  
DIMENSION TBTREE(49,49),JJJ(50),NDI(30),NDJ(30)  
DIMENSION APRIME(20,50),NDEI(20),NDEJ(20),VODD(20)  
DIMENSION FMT(7),VL(30),DNODEI(30),DNODEJ(30),SEEV(50)  
DIMENSION TBJE(20,20),TBRCE(20,49),VO(20),IRCAP(50),IRIND(50)  
DIMENSION YE(49,49),YO(49,49),L(20),M(20),VOD(50),SEEVJ(50)  
DIMENSION A(20,50),ATREE(19,49),ACHORD(19,49),BTREE(49,49),  
1 AARRAY(2500),NODEI(50),NODEJ(50),RR(50),E(50),JJ(50),  
2 INDEX(50),RTREE(50),RCHORD(50),GTREE(50),GCHORD(500),VR(50),  
3 IR(50),TEMP(50),X(19,50),XX(19,49),MA(19,49),MB(19,49),  
4 XT(19,49),XD(19,49),XE(19,49),XF(49,49),XG(49,49),BJE(19,49)  
5 ,BRCE(19,49),BRCRT(19,49),BJRT(19,49),SUM(50),SUM1(50),  
6 SUM2(50),SUM3(50),SUM4(50),TBRCRT(49,49),TBJRT(49,49),  
7 txvar(2000),tyvar(2000),tyvarv(2000,50),tyvari(2000,50)  
CHARACTER*1 TYPE(50),COMP,DTYPE(50),TYP(50),LIN,SCNV(30)  
CHARACTER*1 SCNJ(30),SCNJJ(30),SCCJ,SECIM,GRAF  
data ws id/1/  
OPEN(UNIT=7,FILE='SOR',STATUS='NEW')
```

C

```
8 WRITE(6,*) ' SIRAYA UYUN : 1.GERİLİM KAYNAKLARI '  
WRITE(6,*) ' 2.DİRENÇLER '  
WRITE(6,*) ' 3.DİYOTLAR '  
WRITE(6,*) ' 4.SİĞALAR '  
WRITE(6,*) ' 5.ENDÜKTANSLAR '  
WRITE(6,*) ' 6.AKİM KAYNAKLARI '  
EPSI=1.E-12  
DT=0.0  
TIME=0.0  
R=0  
C=0  
B=0  
V=0  
D=0  
RC=0  
RL=0  
CC=1  
KAT=1  
VLT=1  
DNM=1  
IDS=0  
IRS=0  
CY=0
```

```

PI=4.0*ATAN(1.0)
LSTOP=1
IDNM=1
SAYAC=0
isayac=0
NANIK=1
WRITE(6,*) 'DEVREDE SIĞA/İNDÜKTANS VARMI (Y/N)?'
READ(5,101) COMP
WRITE(6,*) 'DEVREDE DİYOT VARMI (Y/N)?'
READ(5,101) LIN
101  FORMAT(A1)
     IF (COMP.EQ.'Y'.OR.LIN.EQ.'Y') THEN
WRITE(6,*) 'ZAMAN ARALIĞI (DT)      TOPLAM ZAMAN (TIME)'
READ*,DT,TIME
NANIK=6
END IF

C
K=1
100  WRITE(6,*) 'TYPE '
     READ(5,102) TYPE(K)
102  FORMAT(A1)
     IF (TYPE(K).EQ.'Z') GO TO 159
WRITE(6,*) 'NODEİ  NODEJ  VALUE'
READ*,NODEİ(K),NODEJ(K),VAL
159  IF (TYPE(K).EQ.'V') THEN
     V=V+1
     B=B+1
     SEEV(K)=VAL
WRITE(6,*) 'A-DOĞRU GERİLİM'
WRITE(6,*) 'B-SİNÜS'
WRITE(6,*) 'C-COSİNÜS      SEÇİNİZ (A/B/C)?'
READ(5,187) SCNV(K)
187  FORMAT(A1)
     IF (SCNV(K).EQ.'A') THEN
E(K)=VAL
ELSE
WRITE(6,*) 'FREKANSI GİR?'
READ*,FREK
OMEGA(K)=2.*PI*FREK
TPD(K)=VAL
     IF (SCNV(K).EQ.'B') THEN
E(K)=VAL*SIN(OMEGA(K)*SAYAC)
     ELSE IF (SCNV(K).EQ.'C') THEN
E(K)=VAL*COS(OMEGA(K)*SAYAC)
     END IF
END IF
ELSE IF (TYPE(K).EQ.'R') THEN
R=R+1
RR(K)=VAL
B=B+1
IRS=IRS+1
ELSE IF (TYPE(K).EQ.'D') THEN

```

```

WRITE(6,*) 'EŞİK GERİLİMİNİ GIR'
READ*,VOD(K)
R=R+1
RR(K)=1.E+6
B=B+1
IDS=IDS+1
ELSE IF (TYPE(K).EQ.'C') THEN
WRITE(6,*) 'BASLANGIÇ DEĞERİ NEDİR?'
READ*,VO(K)
B=B+2
RC=RC+1
R=R+1
RR(K)=DT/VAL
DTYPE(CC)='C'
JJJ(CC)=-1.*VO(K)*(VAL/DT)
DNODEI(CC)=NODEI(K)
DNODEJ(CC)=NODEJ(K)
CC=CC+1
C=CC-1
ELSE IF (TYPE(K).EQ.'L') THEN
WRITE(6,*) 'BASLANGIÇ DEĞERİ NEDİR?'
READ*,JJJ(CC)
VO(K)=JJJ(CC)
B=B+2
RL=RL+1
R=R+1
RR(K)=VAL/DT
DTYPE(CC)='L'
DNODEI(CC)=NODEI(K)
DNODEJ(CC)=NODEJ(K)
CC=CC+1
C=CC-1
ELSE IF (TYPE(K).EQ.'J') THEN
B=B+1
C=C+1
CY=CY+1
SEEVJ(KAT)=VAL
WRITE(6,*) 'A-DOGRU AKIM'
WRITE(6,*) 'B-SİNÜS'
WRITE(6,*) 'C-COSİNÜS          SEÇİNİZ(A/B/C)?'
READ(5,888) SCCJ
888 FORMAT(A1)
IF(SCCJ.EQ.'A') THEN
VL(KAT)=VAL
ELSE
WRITE(6,*) 'FREKANSI GIR?'
READ*,FREK
OMEGAJ(KAT)=2.*PI*FREK
TPDJ(KAT)=VAL
SCNJJ(KAT)=SCCJ
IF(SCCJ.EQ.'B') THEN
VL(KAT)=VAL*SIN(OMEGAJ(KAT))*SAYAC)

```

```

ELSE IF(SCCJ.EQ.'C') THEN
  VL(KAT)=VAL*COS(OMEGAJ(KAT)*SAYAC)
END IF
END IF
TYP(KAT)=TYPE(K)
NDI(KAT)=NODEI(K)
NDJ(KAT)=NODEJ(K)
KAT=KAT+1
ELSE IF (TYPE(K).EQ.'Z') THEN
GO TO 152
ELSE
GO TO 9012
END IF
K=K+1
GO TO 100
152 IF(COMP.EQ.'Y') THEN
CC=CC-1
K=K-(KAT-1)
DO 60606 I=1,CC
TYPE(K)=DTYPE(I)
JJ(K)=JJJ(I)
NODEI(K)=DNODEI(I)
NODEJ(K)=DNODEJ(I)
60606 K=K+1
DO 70707 I=1,(KAT-1)
SEEV(K)=SEEVJ(I)
TYPE(K)=TYP(I)
JJ(K)=VL(I)
NODEI(K)=NDI(I)
NODEJ(K)=NDJ(I)
SCNV(K)=SCNJJ(I)
IF(SCNJJ(I).EQ.'B'.OR.SCNJJ(I).EQ.'C') THEN
TPD(K)=TPDJ(I)
OMEGA(K)=OMEGAJ(I)
END IF
70707 K=K+1
ELSE
KAV=K-1
K=K-(KAT-1)
DO 80808 I=K,KAV
JJ(I)=VL(VLT)
SEEV(I)=SEEVJ(VLT)
TYPE(I)=TYP(VLT)
NODEI(I)=NDI(VLT)
NODEJ(I)=NDJ(VLT)
SCNV(I)=SCNJJ(VLT)
IF(SCNJJ(VLT).EQ.'B'.OR.SCNJJ(VLT).EQ.'C') THEN
TPD(I)=TPDJ(VLT)
OMEGA(I)=OMEGAJ(VLT)
END IF
80808 VLT=VLT+1
END IF

```

C

```

N=0
DO 157 K=1,B
IF(N.GT.NODEI(K)) THEN
N=N
ELSE
N=NODEI(K)
END IF
IF(N.GT.NODEJ(K)) THEN
N=N
ELSE
N=NODEJ(K)
END IF
157 CONTINUE
IF(N.EQ.0) GO TO 30000
WRITE(6,*) 'TOPRAK?'
READ*,GROUND
IF(V.EQ.0) GO TO 2
1 IF(C.EQ.0) GO TO 4
GO TO 6
2 WRITE(6,3)
3 FORMAT(15X,'HIÇ GERİLİM KAYNAĞI YOK. ')
GO TO 1
4 WRITE(6,5)
5 FORMAT(15X,'HIÇ AKIM KAYNAĞI YOK. ')
6 IF(N.LT.2) GO TO 9002
IF(B.LT.2) GO TO 9004
IF(R.EQ.0) GO TO 9006
IF(B.LT.N) GO TO 9008
NM=N-1
NB=B-(N-1)
TR=NM-V
KV=V+1
CR=R-(NM-V)
NP=N+1
KVR=V+R

C
WRITE(7,30701) DT,TIME
30701 FORMAT(//,8X,'DT =',F12.10,8X,'TIME =',F12.6,/)
WRITE(7,*) '*****'
WRITE(7,*) ' DÜĞÜMLER          G.KAYNAKLARI          DİRENÇLER
1      A.KAYNAKLARI '
DO 153 K=1,B
IF(TYPE(K).EQ.'V') THEN
IF(SCNV(K).EQ.'A') THEN
WRITE(7,154) NODEI(K),NODEJ(K),E(K)
154 FORMAT(1X,I3,'-',I3,6X,F12.3)
ELSE IF(SCNV(K).EQ.'B') THEN
WRITE(7,15401) NODEI(K),NODEJ(K),SEEV(K),OMEGA(K)
15401 FORMAT(1X,I3,'-',I3,6X,F5.1,'SIN',F5.1,'t')
ELSE IF(SCNV(K).EQ.'C') THEN
WRITE(7,15402) NODEI(K),NODEJ(K),SEEV(K),OMEGA(K)

```

```

15402  FORMAT(1X,I3,'-',I3,6X,F5.1,'COS',F5.1,'t')
      END IF
      ELSE IF (TYPE(K).EQ.'R') THEN
      WRITE(7,155) NODEI(K),NODEJ(K),RR(K)
155    FORMAT(1X,I3,'-',I3,17X,F12.3)
      ELSE IF (TYPE(K).EQ.'J') THEN
      IF(SCNV(K).EQ.'A') THEN
      WRITE(7,158) NODEI(K),NODEJ(K),JJ(K)
158    FORMAT(1X,I3,'-',I3,35X,F12.3)
      ELSE IF (SCNV(K).EQ.'B') THEN
      WRITE(7,15407) NODEI(K),NODEJ(K),SEEV(K),OMEGA(K)
15407  FORMAT(1X,I3,'-',I3,35X,F5.1,'SIN',F5.1,'t')
      ELSE IF (SCNV(K).EQ.'C') THEN
      WRITE(7,15408) NODEI(K),NODEJ(K),SEEV(K),OMEGA(K)
15408  FORMAT(1X,I3,'-',I3,35X,F5.1,'COS',F5.1,'t')
      END IF
      END IF
153    CONTINUE
C
      MRC=RC
      MRL=RL
      WRITE(7,*) '*****'
      WRITE(7,*) '
      WRITE(7,*) 'DÜĞÜMLER      SİĞALAR      İNDÜKTANSLAR      İLK      EŞİK'
      WRITE(7,*) '
      WRITE(7,*) '
      DEĞER      VOLT'
      DO 193 K=1,(B+RL+RC)
      IF (TYPE(K).EQ.'C') THEN
      IF (MRC.EQ.0) GO TO 193
      VOM=DT/RR(K)
      WRITE(7,81) NODEI(K),NODEJ(K),VOM,VO(K)
81    FORMAT(1X,I3,'-',I2,2X,F12.10,15X,F4.2)
      MRC=MRC-1
      ELSE IF (TYPE(K).EQ.'L') THEN
      IF (MRL.EQ.0) GO TO 193
      VOM=RR(K)*DT
      WRITE(7,83) NODEI(K),NODEJ(K),VOM,VO(K)
83    FORMAT(1X,I3,'-',I2,14X,F12.7,3X,F4.2)
      MRL=MRL-1
      ELSE IF (TYPE(K).EQ.'D') THEN
      WRITE(7,87) NODEI(K),NODEJ(K),VOD(K)
87    FORMAT(1X,I3,'-',I2,36X,F6.2)
      END IF
193    CONTINUE
      WRITE(7,*) '*****'
      IBB=B-(RL+RC)
      ICC=C-(RL+RC)
      WRITE(7,161)N,IBB,IRS,V,ICC,RC,RL,IDS,GROUND
161  FORMAT(2X,'N=',I2,2X,'B=',I2,2X,'R=',I2,2X,'V=',I2,2X,'J=',I2,
12X,'SİĞA=',I2,2X,'İND=',I2,2X,'DİYOT=',I2,2X,'TOPRAK=',I2,
1'.NODE')
      WRITE(7,*) '*****'

```

C

```

KASK=V+R+1
KAPK=V+R+RC+1
KAJK=V+R+RC+RL+1
KSSK=KASK
KPPK=KAPK
KKK=KASK
KLL=KAPK

```

C

C*****

C

```

321  isayac=isayac+1
      txvar(isayac)=sayac
      IF(LSTOP.GT.1) THEN
      DO 40008 K=1,B
      IF (SCNV(K).EQ.'B') THEN
          IF (TYPE(K).EQ.'V') THEN
              E(K)=TPD(K)*SIN(OMEGA(K)*SAYAC)
          ELSE
              JJ(K)=TPD(K)*SIN(OMEGA(K)*SAYAC)
          END IF
      ELSE IF (SCNV(K).EQ.'C') THEN
          IF (TYPE(K).EQ.'V') THEN
              E(K)=TPD(K)*COS(OMEGA(K)*SAYAC)
          ELSE
              JJ(K)=TPD(K)*COS(OMEGA(K)*SAYAC)
          END IF
      END IF
40008 CONTINUE
      END IF

```

C

```

      IF(IDNM.GT.4.AND.LIN.EQ.'Y') THEN
      K=0
4321  K=K+1
          IF(K.GT.B) GO TO 32455
          IF(TYPE(K).EQ.'D') THEN
              IF(VR(K).GT.VOD(K)) THEN
                  RR(K)=1.E-6
                  TYPE(K)='A'
              END IF
          ELSE IF (TYPE(K).EQ.'A') THEN
              IF(IR(K).LT.0.0) THEN
                  RR(K)=1.E+6
                  TYPE(K)='D'
              END IF
          END IF
          GO TO 4321
      END IF

```

C

```

32455  IDNM=10
      LSTOP=5

```

C

C INCIDENCE MATRIS A(I,K)'NIN HAZIRLANMASI

```

DO 300 K=1,B
INDEX(K)=K
DO 300 I=1,N
IF(NODEI(K).EQ.I) GO TO 10
IF(NODEJ(K).EQ.I) GO TO 20
A(I,K)=0.0
GO TO 300
10 A(I,K)=1.0
GO TO 300
20 A(I,K)=-1.0
300 CONTINUE
C
DO 360 I=1,N
CHECK=0.0
DO 350 K=1,B
IF(A(I,K).EQ.EPSI) GO TO 350
CHECK=CHECK+1.0
350 CONTINUE
IF(CHECK.LT.2.0) GO TO 9010
360 CONTINUE
C
IF(GROUND.NE.0) GO TO 30
BIG=0.0
DO 400 I=1,N
SUM(I)=0.0
DO 500 K=1,B
500 SUM(I)=ABS(A(I,K))+SUM(I)
IF(SUM(I).LE.BIG) GO TO 400
BIG=SUM(I)
GROUND=I
400 CONTINUE
WRITE(7,987) GROUND
987 FORMAT(5X,'TOPRAK=',I3)
C
30 DO 600 K=1,B
DO 600 I=1,NM
IF(I.GE.GROUND) GO TO 35
A(I,K)=A(I,K)
GO TO 600
35 IP=I+1
A(I,K)=A(IP,K)
600 CONTINUE
DO 650 I=1,NM
DO 650 J=1,B
650 APRIME(I,J)=A(I,J)
CALL TRIANG(A,NM,B,INDEX)
C
KVR=V+R
DO 805 K=1,NM
IF(INDEX(K).GT.KVR) GO TO 9014
805 CONTINUE
DO 810 K=N,B

```



```

      IF(INDEX(K).LE.V) GO TO 9000
810  CONTINUE
C
      DO 820 J=1,NM
      IF(A(J,J).EQ.EPSI) GO TO 9012
820  CONTINUE
C
      DO 800 K=1,B
      DO 800 I=1,NM
800  A(I,K)=APRIME(I,INDEX(K))
C
      DO 69 K=1,NM
      DO 69 I=1,NM
      ATREE(I,K)=A(I,K)
69  CONTINUE
      DO 1600 K=1,NB
      DO 1600 I=1,NM
      KH=K+NM
      ACHORD(I,K)=A(I,KH)
1600 CONTINUE
C
      DO 88 J=1,NM
      DO 88 I=1,NM
      NU=I+NM*(J-1)
      AARRAY(NU)=ATREE(I,J)
88  CONTINUE
      CALL MINV(AARRAY,NM,D,L,M)
      DO 25 J=1,NM
      DO 25 I=1,NM
      NU=I+NM*(J-1)
      ATREE(I,J)=AARRAY(NU)
25  CONTINUE
C
      DO 1660 I=1,NM
      DO 1660 J=1,NB
      BTREE(I,J)=0.0
      DO 1665 K=1,NM
1665 BTREE(I,J)=-ATREE(I,K)*ACHORD(K,J)+BTREE(I,J)
1660 CONTINUE
      IF(TR.EQ.0) GO TO 6000
      IF(CR.EQ.0) GO TO 6600
      GO TO 1644
C
6000 DO 6100 I=1,R
      IPV=I+V
6100 GCHORD(I)=1.0/RR(INDEX(IPV))
C
      DO 6200 I=1,NM
      VR(INDEX(I))=E(I)
      DO 6200 J=1,NB
6200 TBTREE(J,I)=BTREE(I,J)
C

```

```

DO 6300 I=1,NB
IPV=I+V
VR(INDEX(IPV))=0.0
DO 6225 J=1,NM
6225 VR(INDEX(IPV))=TBTREE(I,J)*E(J)+VR(INDEX(IPV))
IF(I.LE.R) GO TO 6250
GO TO 6300
6250 IR(INDEX(IPV))=GCHORD(I)*VR(INDEX(IPV))
6300 CONTINUE
IF(C.EQ.0) GO TO 6450
DO 6400 K=1,C
KVR=K+V+R
6400 IR(INDEX(KVR))=JJ(KVR)
C
6450 DO 6500 J=1,NM
IR(INDEX(J))=0
DO 6500 I=1,NB
IPV=I+V
6500 IR(INDEX(J))=BTREE(J,I)*IR(INDEX(IPV))+IR(INDEX(J))
GO TO 10000
C
6600 DO 6650 I=1,R
IPV=I+V
6650 RTREE(I)=RR(INDEX(IPV))
DO 6700 K=1,NB
KVR=K+V+R
IR(INDEX(KVR))=JJ(KVR)
DO 6700 I=1,NM
6700 TBTREE(K,I)=BTREE(I,K)
C
DO 6800 I=1,NM
IPV=I+V
IR(INDEX(I))=0.0
DO 6725 K=1,C
KVR=K+V+R
6725 IR(INDEX(I))=BTREE(I,K)*JJ(KVR)+IR(INDEX(I))
IF(I.GT.V) GO TO 6750
GO TO 6800
6750 VR(INDEX(IPV))=RTREE(I-V)*IR(INDEX(IPV))
6800 CONTINUE
IF(V.EQ.0) GO TO 6875
C
DO 6850 I=1,V
6850 VR(INDEX(I))=E(I)
C
6875 DO 6900 K=1,NB
KVR=K+V+R
VR(INDEX(KVR))=0.0
DO 6900 J=1,NM
6900 VR(INDEX(KVR))=TBTREE(K,J)*VR(INDEX(J))+VR(INDEX(KVR))
GO TO 10000
C

```

```

1644 DO 1700 I=1,NM
      DO 1700 J=1,NB
      A(I,J)=BTREE(I,J)
1700 CONTINUE
      DO 1705 I=1,NM
      DO 1705 J=1,NB
1705 BTREE(J,I)=A(I,J)
C
      IF(V.EQ.0) GO TO 1850
      DO 1800 I=1,CR
      DO 1800 K=1,V
      BRCE(I,K)=BTREE(I,K)
1800 CONTINUE
1850 DO 1900 I=1,CR
      DO 1900 K=1,TR
      JKV=K+V
      BRCRT(I,K)=BTREE(I,JKV)
1900 CONTINUE
      IF(C.EQ.0.0) GO TO 2197
      IF(V.EQ.0) GO TO 2001
      MCR=CR+1
      JCR=CR+C
      DO 2000 I=1,C
      DO 2000 K=1,V
      KP=I+CR
      BJE(I,K)=BTREE(KP,K)
2000 CONTINUE
2001 DO 2100 I=1,C
      DO 2100 K=1,TR
      IT=I+CR
      KV=K+V
      BJRT(I,K)=BTREE(IT,KV)
2100 CONTINUE
C
2197 DO 2200 I=1,TR
      IPV=I+V
      RTREE(I)=RR(INDEX(IPV))
      GTREE(I)=1.0/RR(INDEX(IPV))
2200 CONTINUE
      DO 2300 I=1,CR
      INX=I+NM
      RCHORD(I)=RR(INDEX(INX))
      GCHORD(I)=1.0/RR(INDEX(INX))
2300 CONTINUE
C
      IF(C.EQ.0) GO TO 2401
      DO 2400 I=1,C
      DO 2400 K=1,TR
2400 TBJRT(K,I)=BJRT(I,K)
C
2401 DO 2500 I=1,CR
      DO 2500 K=1,TR

```

```

2500  TBCRT(K,I)=BRCRT(I,K)
C
C      X=RTREE*TBCRT
      DO 2800 I=1,TR
      DO 2800 J=1,CR
      X(I,J)=0
      DO 2900 K=1,TR
      IF(I.NE.K) GO TO 2900
      X(I,J)=RTREE(I)*TBCRT(K,J)+X(I,J)
2900  CONTINUE
2800  CONTINUE
C
C      XX=RTREE*TBCRT*GCHORD
      DO 3000 I=1,TR
      DO 3000 J=1,CR
      XX(I,J)=0
      DO 3100 K=1,CR
      IF(J.NE.K) GO TO 3100
      XX(I,J)=X(I,K)*GCHORD(K)+XX(I,J)
3100  CONTINUE
3000  CONTINUE
C
C      M31=MB
      DO 3400 J=1,CR
      DO 3400 I=1,CR
      MB(I,J)=0
      DO 3500 K=1,TR
3500  MB(I,J)=BRCRT(I,K)*XX(K,J)+MB(I,J)
3400  CONTINUE
      DO 3600 K=1,CR
      DO 3600 I=1,CR
      IF(I.EQ.K) GO TO 95
      MB(I,K)=MB(I,K)
      GO TO 3600
      95  MB(I,K)=MB(I,K)+1
3600  CONTINUE
      DO 74 I=1,CR
      DO 74 J=1,CR
      NU=I+CR*(J-1)
      74  AARRAY(NU)=MB(I,J)
           CALL MINV(AARRAY,CR,D,L,M)
      DO 76 I=1,CR
      DO 76 J=1,CR
      NU=I+CR*(J-1)
      76  MB(I,J)=AARRAY(NU)
C
      IF(C.EQ.0) GO TO 4750
      DO 4200 I=1,TR
      DO 4200 J=1,C
      XE(I,J)=0
      DO 4300 K=1,TR
      IF(I.NE.K) GO TO 4300

```

```

      XE(I, J)=RTREE(K)*TBJRT(K, J)+XE(I, J)
4300 CONTINUE
4200 CONTINUE
C
      DO 4400 I=1, CR
      DO 4400 J=1, C
      XF(I, J)=0
      DO 4500 K=1, TR
4500 XF(I, J)=BRCRT(I, K)*XE(K, J)+XF(I, J)
4400 CONTINUE
-----
C
C      XG=MB*XF
      IF(C.EQ.0) GO TO 4750
      DO 4600 I=1, CR
      DO 4600 J=1, C
      XG(I, J)=0
      DO 4700 K=1, CR
4700 XG(I, J)=MB(I, K)*XF(K, J)+XG(I, J)
4600 CONTINUE
C
4750 IF(V.EQ.0) GO TO 4880
      DO 4637 I=1, CR
      DO 4637 J=1, V
      YE(I, J)=0.0
      DO 4638 K=1, CR
4638 YE(I, J)=MB(I, K)*BRCE(K, J)+YE(I, J)
4637 CONTINUE
4880 DO 4800 I=1, CR
      SUM1(I)=0
      IF(C.EQ.0) GO TO 4901
      DO 4900 K=1, C
      KRV=K+R+V
4900 SUM1(I)=XG(I, K)*JJ(KRV)+SUM1(I)
4901 SUM2(I)=0.0
      IF(V.EQ.0) GO TO 4870
      DO 5000 K=1, V
5000 SUM2(I)=YE(I, K)*E(K)+SUM2(I)
4870 INX=I+NM
      VR(INDEX(INX))=-SUM1(I)-SUM2(I)
5400 IR(INDEX(INX))=GCHORD(I)*VR(INDEX(INX))
4800 CONTINUE
C      M68=MA
C      MA=RTREE*TBR CRT*GCHORD*BRCRT
      DO 3200 J=1, TR
      DO 3200 I=1, TR
      MA(I, J)=0
      DO 3300 K=1, CR
3300 MA(I, J)=XX(I, K)*BRCRT(K, J)+MA(I, J)
3200 CONTINUE
      DO 3700 K=1, TR
      DO 3700 I=1, TR
      IF(I.EQ.K) GO TO 85

```

```

      MA(I,K)=MA(I,K)
      GO TO 3700
85    MA(I,K)=MA(I,K)+1
3700  CONTINUE
      DO 49 I=1,TR
      DO 49 J=1,TR
      NU=I+TR*(J-1)
49    AARRAY(NU)=MA(I,J)
      CALL MINV(AARRAY,TR,D,L,M)
      DO 65 I=1,TR
      DO 65 J=1,TR
      NU=I+TR*(J-1)
65    MA(I,J)=AARRAY(NU)
C
C    XT(MA)-1*RTREE*TBR CRT*GCHORD
      DO 3800 I=1,TR
      DO 3800 J=1,CR
      XT(I,J)=0
      DO 3900 K=1,TR
3900  XT(I,J)=MA(I,K)*XX(K,J)+XT(I,J)
3800  CONTINUE
C
      IF(V.EQ.0) GO TO 4041
      DO 4000 I=1,TR
      DO 4000 J=1,V
      XD(I,J)=0
      DO 4100 K=1,CR
4100  XD(I,J)=XT(I,K)*BRCE(K,J)+XD(I,J)
4000  CONTINUE
C    YO=MA*XE
4041  IF(C.EQ.0) GO TO 5150
      DO 4037 I=1,TR
      DO 4037 J=1,C
      YO(I,J)=0.0
      DO 4039 K=1,TR
4039  YO(I,J)=MA(I,K)*XE(K,J)+YO(I,J)
4037  CONTINUE
C
5150  DO 5100 I=1,TR
      SUM3(I)=0.0
      IF(C.EQ.0) GO TO 5201
      DO 5200 K=1,C
      KRV=K+R+V
5200  SUM3(I)=YO(I,K)*JJ(KRV)+SUM3(I)
5201  SUM4(I)=0.0
      IF(V.EQ.0) GO TO 5310
      DO 5300 K=1,V
5300  SUM4(I)=XD(I,K)*E(K)+SUM4(I)
5310  IPV=I+V
      VR(INDEX(IPV))=SUM3(I)-SUM4(I)
5500  IR(INDEX(IPV))=GTREE(I)*VR(INDEX(IPV))
5100  CONTINUE

```

```

C
C   IF(C.EQ.0) GO TO 5480
      DO 5430 K=1,C
      SUM3(K)=0.0
      DO 5430 I=1,TR
      IPV=I+V
5430  SUM3(K)=-BJRT(K,I)*VR(INDEX(IPV))+SUM3(K)
      DO 5450 I=1,C
      SUM4(I)=0.0
      IF (V.EQ.0) GO TO 5450
      DO 5451 J=1,V
5451  SUM4(I)=-BJE(I,J)*E(J)+SUM4(I)
5450  CONTINUE
      DO 5490 K=1,C
      KRV=K+R+V
5490  VR(INDEX(KRV))=SUM3(K)+SUM4(K)
C
5480  IF(V.EQ.0) GO TO 5665
      DO 5600 I=1,V
5600  VR(INDEX(I))=E(I)
C
      IF(C.EQ.0) GO TO 5640
      DO 5630 J=1,V
      SUM1(J)=0.0
      DO 5630 K=1,C
      TBJE(J,K)=BJE(K,J)
      KRV=K+R+V
5630  SUM1(J)=TBJE(J,K)*JJ(KRV)+SUM1(J)
5640  DO 5650 J=1,V
      SUM2(J)=0.0
      DO 5650 I=1,CR
      TBRCE(J,I)=BRCE(I,J)
      INX=I+NM
5650  SUM2(J)=TBRCE(J,I)*IR(INDEX(INX))+SUM2(J)
      DO 5660 I=1,V
5660  IR(INDEX(I))=SUM1(I)+SUM2(I)
5665  IF(C.EQ.0) GO TO 10000
C
      DO 5670 K=1,C
      KRV=K+R+V
5670  IR(INDEX(KRV))=JJ(KRV)
      GO TO 10000
C
9000  WRITE(6,9001)
9001  FORMAT(5X,'**HATA**G.KAYNAKLARINDAN OLUŞAN ÇEVİRİM VAR. ')
      GO TO 1102
9002  WRITE(6,9003)
9003  FORMAT(5X,'**HATA**DEVREDE ENAZ İKİ DÜĞÜM BULUNMALIDIR, '
1, /9X, 'KONTROL EDİN.GİRİŞ YALNIŞ GİRİLMİŞ OLABİLİR. ')
      GO TO 1102
9004  WRITE(6,9005)

```

```

9005 FORMAT(5X,'**HATA**DEVREDE ENAZ İKİ DAL BULUNMALIDIR,'
1./9X,'KONTROL EDİN. GİRİŞ YALNIZ GİRİLMİŞ OLABİLİR.')
GO TO 1102
9006 WRITE(6,9007)
9007 FORMAT(5X,'**HATA**OMİK BİR DEVRE İÇİN DİRENÇ OLMALI.'
1./9X,'GİRİŞİ KONTROL EDİN.')
GO TO 1102
9008 WRITE(6,9009)
9009 FORMAT(5X,'**HATA**DAL SAYISINA ORANLA FAZLA DÜĞÜM VAR')
GO TO 1102
9010 WRITE(6,9011) I
9011 FORMAT(5X,'**HATA** DÜĞÜM',I3,1X,'BOŞTA KALMIŞ.')
GO TO 1102
9012 WRITE(6,9013)
9013 FORMAT('**HATA** PROGRAM DEVRE ÜZERİNDE UYGUN AĞAÇ'
1/9X,' BELİRLEYEMİYOR. GİRİŞİ KONTROL EDİN.')
GO TO 1102
9014 WRITE(6,9015)
9015 FORMAT(5X,'**HATA** YALNIZ AKIM KAYNAKLARININ BAĞLI')
1/9X,' OLDUĞU DÜĞÜM VAR.')
GO TO 1102
C
10000 KASK=KKK
KAPK=KLL
SYC1=RC
SYC2=RL
SYC3=RC
SYC4=RL
IF (COMP.EQ.'Y') THEN
IF (SAYAC.GT.TIME) GO TO 90999
DO 41 I=1,(B+RL+RC)
IF (TYPE(I).EQ.'C') THEN
IF (SYC1.EQ.0) GO TO 41
JJ(KASK)=-VR(I)*(1./RR(I))
KASK=KASK+1
SYC1=SYC1-1
ELSE IF (TYPE(I).EQ.'L') THEN
IF (SYC2.EQ.0) GO TO 41
JJ(KAPK)=IR(I)+JJ(KAPK)
KAPK=KAPK+1
SYC2=SYC2-1
END IF
41 CONTINUE
END IF
C
90999 B=B-(RC+RL)
KAL=0
KOF=1
WRITE(7,10001)
10001 FORMAT(6X,' NUMARA ',7X,'ELEMEN ÜZERİNDEKİ VOLT',
17X,' ELEMANDAN AKAN AKIM')
C

```



```

11111 IF(KAL.EQ.B) GO TO 19331
      IF(TYPE(KOF).EQ.'V') GO TO 11000
      IF(TYPE(KOF).EQ.'R') GO TO 11002
      IF(TYPE(KOF).EQ.'D') GO TO 40404
      IF(TYPE(KOF).EQ.'A') GO TO 50505
      IF(TYPE(KOF).EQ.'C') GO TO 20202
      IF(TYPE(KOF).EQ.'L') GO TO 30303
      IF(TYPE(KOF).EQ.'J') GO TO 11004
11000 KAL=KAL+1
      WRITE(7,11001) KAL,VR(KOF),IR(KOF)
11001 FORMAT(1X,I3,3X,'( GER. KAYNAGI )',6X,1PE12.5,1X,
1 'VOLT',10X,1PE12.5,1X,'AMP',//)
      tyvarv(isayac,KAL)=VR(KOF)
      tyvari(isayac,KAL)=IR(KOF)
      KOF=KOF+1
      GO TO 11111
11002 KAL=KAL+1
      WRITE(7,11003) KAL,VR(KOF),IR(KOF)
11003 FORMAT(1X,I3,3X,'( DIRENC )',12X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP',//)
      tyvarv(isayac,KAL)=VR(KOF)
      tyvari(isayac,KAL)=IR(KOF)
      KOF=KOF+1
      GO TO 11111
20202 IF(SYC3.EQ.0) GO TO 54321
      KAL=KAL+1
      IRCAP(KOF)=IR(KOF)+IR(KSSK)
      WRITE(7,20201) KAL,VR(KOF),IRCAP(KOF)
20201 FORMAT(1X,I3,3X,'( SIGA )',11X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP',//)
      tyvarv(isayac,KAL)=VR(KOF)
      tyvari(isayac,KAL)=IRCAP(KOF)
      KSSK=KSSK+1
      SYC3=SYC3-1
54321 KOF=KOF+1
      GO TO 11111
30303 IF(SYC4.EQ.0) GO TO 76543
      KAL=KAL+1
      IRIND(KOF)=IR(KOF)+IR(KPPK)
      WRITE(7,30301) KAL,VR(KOF),IRIND(KOF)
30301 FORMAT(1X,I3,3X,'( INDUKTANS )',12X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP',//)
      tyvarv(isayac,KAL)=VR(KOF)
      tyvari(isayac,KAL)=IRIND(KOF)
      KPPK=KPPK+1
      SYC4=SYC4-1
76543 KOF=KOF+1
      GO TO 11111
11004 KAL=KAL+1
      WRITE(7,11005) KAL,VR(KAJK),IR(KAJK)
11005 FORMAT(1X,I3,3X,'( AKIM KAYNAGI )',6X,1PE12.5,1X,
1 'VOLT',10X,1PE12.5,1X,'AMP',//)

```

```

KAJK=KAJK+1
KOF=KOF+1
GO TO 11111

```

C

```

40404 KAL=KAL+1
WRITE(7,16161) KAL,VR(KOF),IR(KOF)
16161 FORMAT(1X,I3,3X,'(DIYOT(TIKAMADA))',6X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP',/)
tyvarv(isayac,KAL)=VR(KOF)
tyvari(isayac,KAL)=IR(KOF)
KOF=KOF+1
GO TO 11111

```

C

1

30

30

C

```

50505 KAL=KAL+1
WRITE(7,17171) KAL,VR(KOF),IR(KOF)
17171 FORMAT(1X,I3,3X,'(DIYOT(ILETİMDE))',6X,1PE12.5,1X,'VOLT',
110X,1PE12.5,1X,'AMP',/)
tyvarv(isayac,KAL)=VR(KOF)
tyvari(isayac,KAL)=IR(KOF)
KOF=KOF+1
GO TO 11111

```

C*

C

C

```

19331 WRITE(7,*) '*****'
IF(NANIK.LT.3) GO TO 1102
SAYAC=SAYAC+DT
IF(SAYAC.GT.TIME) GO TO 18305
B=B+(RC+RL)
GO TO 321

```

C

C

```

18305 WRITE(6,8176)
8176 FORMAT(/////,15X,'AYRICA GRAFIK SONUÇ İSTENİYORMU (Y/N)?')
READ(5,2609) GRAF
2609 FORMAT(A1)

```

C

```

IF(GRAF.EQ.'N') GO TO 1102
WRITE(6,*) 'A-ELEMAN İÇİNDEN AKAN AKIMIN GRAFIĞI'
WRITE(6,*) 'A-ELEMAN ÜZERİNDEKİ GERİLİMİN GRAFIĞI'
WRITE(6,*) ' SEÇİNİZ (A/B)?'
READ(5,17128) SECIM
17128 FORMAT (A1)
WRITE(6,*) 'GRAFIĞINI GÖRMEK İSTEDİĞİNİZ ELEMANIN NOSUNU GIR'
READ(5,8076) KEKNO
8076 FORMAT(I2)

```

C

```

DO 7923 I=1,isayac
IF(SECIM.EQ.'A') THEN
tyvar(I)=tyvari(I,KEKNO)
ELSE IF(SECIM.EQ.'B') THEN
tyvar(i)=tyvarv(I,KEKNO)
END IF

```

C

```

7923 CONTINUE

```

C

```

18307 call gks$open_gks('sys$error:')

```

```

      KAJK=KAJK+1
      KOF=KOF+1
      GO TO 11111
C
40404 KAL=KAL+1
      WRITE(7,16161) KAL,VR(KOF),IR(KOF)
16161 FORMAT(1X,I3,3X,' (DIYOT(TIKAMADA)) ',6X,1PE12.5,1X,' VOLT',
      110X,1PE12.5,1X,' AMP',//)
      tyvarv(isayac,KAL)=VR(KOF)
      tyvari(isayac,KAL)=IR(KOF)
      KOF=KOF+1
      GO TO 11111
C
50505 KAL=KAL+1
      WRITE(7,17171) KAL,VR(KOF),IR(KOF)
17171 FORMAT(1X,I3,3X,' (DIYOT(ILETIMDE)) ',6X,1PE12.5,1X,' VOLT',
      110X,1PE12.5,1X,' AMP',//)
      tyvarv(isayac,KAL)=VR(KOF)
      tyvari(isayac,KAL)=IR(KOF)
      KOF=KOF+1
      GO TO 11111
C
19331 WRITE(7,*) '*****'
      IF(NANIK.LT.3) GO TO 1102
      SAYAC=SAYAC+DT
      IF(SAYAC.GT.TIME) GO TO 18305
      B=B+(RC+RL)
      GO TO 321
C
18305 WRITE(6,8176)
8176  FORMAT(/////,.15X,' AYRICA GRAFIK SONUC ISTENIYORMU (Y/N)?')
      READ(5,2609) GRAF
2609  FORMAT(A1)
      IF(GRAF.EQ.'N') GO TO 1102
      WRITE(6,*) 'A-ELEMAN ICINDEN AKAN AKIMIN GRAFIGI'
      WRITE(6,*) 'A-ELEMAN UZERINDEKI GERILIMIN GRAFIGI'
      WRITE(6,*) '          SECINIZ (A/B)?'
      READ(5,17128) SECIM
17128 FORMAT (A1)
      WRITE(6,*) 'GRAFIGINI GORMEK ISTEDIGINIZ ELEMANIN NOSUNU GIR'
      READ(5,8076) KEKNO
8076  FORMAT(I2)
C
      DO 7923 I=1,isayac
      IF(SECIM.EQ.'A') THEN
      tyvar(I)=tyvari(I,KEKNO)
      ELSE IF(SECIM.EQ.'B') THEN
      tyvar(i)=tyvarv(I,KEKNO)
      END IF
7923  CONTINUE
C
18307 call gks$open_gks('sys$error:')

```

```

call gks$open_ws(ws_id,gks$k_coind_default,gks$k_vt240)
call gks$activate_ws(ws_id)
call gks$set_window(1,0.0,0.04,-20.0,20.0)
call gks$set_VIEWPORT(1,0.0,1.0,0.0,0.5)
CALL GKS$SELECT_XFORM(1)
call gks$polyline(isayac,txvar,tyvar)
call gks$close_seg()
call gks$deactivate_ws(ws_id)
call gks$close_ws(ws_id)
call gks$close_gks()
go to 18305

```

```

C
1102 WRITE(6,30008)
30008 FORMAT(/////15X,'BU, PROGRAMIN SONUDUR.')
30000 CLOSE(UNIT=7)
STOP
END

```

```

C*****

```

```

SUBROUTINE TRIANG(A,N,M,INDEX)
DIMENSION A(20,50),INDEX(50)

```

```

C
DO 1000 LL=1,N
IMP=LL+1
100 IF( A(LL,LL).EQ.0.0) GO TO 400
C
DO 300 I=IMP,N
IF( A(I,LL).EQ.0.0) GO TO 300
DUMMY=A(LL,LL)/A(I,LL)
DO 200 K=LL,M
200 A(I,K)=DUMMY*A(I,K)-A(LL,K)
300 CONTINUE
GO TO 1000
C
400 IF(LL.EQ.N) GO TO 800
DO 700 I=IMP,N
IF (A(I,LL).EQ.0.0) GO TO 600
DO 500 K=LL,M
TEMP=A(I,K)
A(I,K)=A(LL,K)
A(LL,K)=TEMP
500 CONTINUE
GO TO 100
600 IF(I.EQ.N) GO TO 800
700 CONTINUE
C
800 IF(LL.EQ.M) GO TO 1000
DO 900 J=IMP,M
IF( A(LL,J).EQ.0.0) GO TO 900
DO 850 IT=1,N
TEMP=A(IT,J)
A(IT,J)=A(IT,LL)

```

```

      A(IT,LL)=TEMP
850 CONTINUE
C
      KEEP=INDEX(LL)
      INDEX(LL)=INDEX(J)
      INDEX(J)=KEEP
      GO TO 100
900 CONTINUE
1000 CONTINUE
      RETURN
      END
C*****

      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(2500),L(20),M(20)
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
10 IF( ABS(BIGA)-ABS(A(IJ))) 15,20,20
15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
20 CONTINUE
C
      J=L(K)
      IF(J-K) 35,35,25
25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
30 A(JI)=HOLD
C
35 I=M(K)
      IF(I-K) 45,45,38
38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)

```

```

40 A(JI)=HOLD
C
45 IF(BIGA) 48,46,48
46 D=0.0
   RETURN
48 DO 55 I=1,N
   IF(I-K) 50,55,50
50 IK=NK+I
   A(IK)=A(IK)/(-BIGA)
55 CONTINUE
C
   DO 65 I=1,N
   IK=NK+I
   HOLD=A(IK)
   IJ=I-N
   DO 65 J=1,N
   IJ=IJ+N
   IF(I-K) 60,65,60
60 IF(J-K) 62,65,62
62 KJ=IJ-I+K
   A(IJ)=HOLD*A(KJ)+A(IJ)
65 CONTINUE
C
   KJ=K-N
   DO 75 J=1,N
   KJ=KJ+N
   IF(J-K) 70,75,70
70 A(KJ)=A(KJ)/BIGA
75 CONTINUE
C
   D=D*BIGA
C
   A(KK)=1.0/BIGA
80 CONTINUE
C
   K=N
100 K=K-1
   IF(K) 150,150,105
105 I=L(K)
   IF(I-K) 120,120,108
108 JQ=N*(K-1)
   JR=N*(I-1)
   DO 110 J=1,N
   JK=JQ+J
   HOLD=A(JK)
   JI=JR+J
   A(JK)=-A(JI)
110 A(JI)=HOLD
120 J=M(K)
   IF(J-K) 100,100,125
125 KI=K-N
   DO 130 I=1,N

```

```
KI=KI+N  
HOLD=A(KI)  
JI=KI-K+J  
A(KI)=-A(JI)  
130 A(JI)=HOLD  
GO TO 100  
150 RETURN  
END
```



ÖZGEÇMİŞ

1966 yılında İzmit/Gölcük'te doğdu. İlkokulu İzmir/Urla Şehit Kemal İlkokulu ve İçel Kuvai Milliye İlkokullarında, ortaokulu İçel Atatürk Lisesi orta kısmı ve Trabzon Kanuni ortaokullarında, liseyi Trabzon Lisesinde tamamladı. 1983 yılında KTÜ Mühendislik-Mimarlık Fakültesi Elektrik-Elektronik Bölümünde öğrenime başladı ve 1988 yılında Elektronik Mühendisi ünvanı ile mezun oldu. 1989 yılında KTÜ Fen Bilimleri Enstitüsü Elektronik Mühendisliği bölümünde yüksek lisans öğrenimine başladı. Bu arada 1988 yılında Trabzon PTT Baş Müdürlüğü'nde şebeke mühendisi olarak göreve başladı. 1990 yılında PTT'deki görevinden istifa etti.