

25625

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PARÇA-PARÇA DOĞRUSAL VEYA DIŞARDAN KONTROLLU
ELEMANLAR İÇEREN DEVRELERİN PERİYODİK
ÇÖZÜMLERİNİN BULUNMASI

Saadetdin HERDEM

DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

**F.C. YÜSEKÖĞRETİM KURULU
DOKÜMAN YAYIN MERKEZİ**

1993

ELAZIĞ

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PARÇA-PARÇA DOĞRUSAL VEYA DIŞARDAN KONTROLLU
ELEMANLAR İÇEREN DEVRELERİN PERİYODİK
ÇÖZÜMLERİNİN BULUNMASI

Saadeddin HERDEM

DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

Bu Tez, Tarihinde Aşağıda Belirtilen Jüri Tarafından
Oybirliği / Oyçokluğu ile Başarılı / Başarısız Olarak
Kabul Edilmiştir.

(İmza)

(İmza)

(İmza)

III

ÖZET

Doktora Tezi

PARÇA-PARÇA DOĞRUSAL VEYA DIŞARDAN KONTROLLU
ELEMANLAR İÇEREN DEVRELERİN PERİYODİK
ÇÖZÜMLERİNİN BULUNMASI

Saadetdin HERDEM

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği

Anabilim Dalı

1993, Sayfa: 153

Bu tezde, periyodik olarak uyarıldıkları zaman periyodik olarak davranış gösteren doğrusal olmayan devrelerin periyodik sürekli hal çözümlerinin bulunması için bir yöntem sunulmuştur.

Sunulan yöntem, devrede yer alan doğrusal olmayan elemanların yerine, parça-parça doğrusallık yaklaşımı kullanılarak doğrusal RLC elemanları, sabit gerilim ve akım kaynakları ve periyodik olarak çalışan anahtarlardan oluşan eşdeğerlerinin göz önüne alınmasına dayanmaktadır. Yöntem uygulanırken, ilk olarak doğrusal olmayan elemanların sürekli halde parça-parça doğrusallık yaklaşımı yapılan karakteristiklerindeki hangi doğrusal bölgelere girdikleri ve bu bölgelerde yaklaşık ne kadar süreyle kaldıkları belirlenmektedir. Bunun için geçici rejim analizi yapılmaktadır. Daha sonra bu bilgilere göre sürekli hal çözümü

IV

yapılarak yaklaşık olarak bulunmuş olan süreler iteratif bir şekilde istenilen hassasiyete kadar düzeltilmekte ve sonuçta devrenin sürekli hal çözümü bulunmaktadır. Çözüm işlemlerinde sayısal integrasyon veya optimizasyon teknikleri yerine doğrudan kesin sonuçları veren formüllerden yararlanıldığından, sonuçlar kısa sürede alınmaktadır.

Bütün bu hesapları gerçekleştirmek için PGEDAN isimli bir bilgisayar programı hazırlanmış ve yöntemin özelliklerini ortaya koyan değişik örnekler üzerinde uygulanmıştır.

ANAHTAR KELİMELEER: Doğrusal olmayan devreler, parça-parça doğrusal devreler, periyodik olarak çalışan anahtarlar içeren devreler, sürekli hal çözümü, tam çözüm.

V

ABSTRACT

Ph. D. Thesis

THE PERIODICAL SOLUTION OF NETWORKS
CONTAINING PIECEWISE LINEAR OR
CONTROLLED COMPONENTS

Saadetdin HERDEM

Firat University

Graduate School of Naturel and Applied Sciences

Department of Electrical and Electronics Engineering

1993, Page: 153

In this thesis, a new method is presented to find the periodic steady-state solutions of nonlinear networks which yield periodic steady-state responses when excited by periodic inputs.

Instead of the nonlinear components in the network, the presented method is based upon the treatment of their equivalent representations which are obtained by piecewise linear approximation and composed of linear RLC components, constant voltage and current sources and periodically operating switches. During the application of the method, first, the regions in their piecewise linear characteristics and the approximate durations for which the elements stay in these regions under steady-state conditions are determined. For this, transient analysis is applied. Later, using these data and applying the steady-state analysis, the time durations which are found approximately are corrected iteratively with in a desired

VI

accuracy, and finally the steady-state solution of the network is obtained. In the analysis, instead of numerical integration and optimization techniques explicit exact formulas are used, therefore the results are attained in a short time.

To perform these computations a computer program named PGEDAN is prepared and it is applied on several examples exposing the properties of the method.

KEY WORDS: Nonlinear networks, piecewise linear networks, networks containing periodically operated switches, steady-state analysis, exact analysis.



VII

TEŞEKKÜR

Bu tez çalışması sırasında her türlü yardımlarını benden esirgemeyen ve bulunduğumuz şehirler arasındaki uzak mesafeye rağmen yaptığı fedakarlıklarla çalışmamın aksamadan yürümesini sağlayan Tez Yöneticim Sayın Prof. Dr. Muhammet KÖKSAL' a çok teşekkür eder şükranlarımı sunarım.

Ayrıca çalışmalarım süresince gerek yardım eden gerekse manevi yönden destek olan bölümdeki arkadaşlarıma da teşekkür ederim.

Son olarak, oldukça uzun zaman ayırmayı ve sıkı bir tempoyu gerektiren bu çalışma süresince gösterdikleri sabır ve desteklerinden dolayı eşime, kızıma ve oğluma da çok teşekkür ederim.

VIII

İÇİNDEKİLER

ÖZET	I
ABSTRACT	III
TEŞEKKÜR	V
İÇİNDEKİLER	VI
ŞEKİLLER LİSTESİ	IX
SİMGELER LİSTESİ	XIII
1. GİRİŞ	1
2. DOĞRUSAL OLMAYAN DEVRELERİN PERİYODİK ÇÖZÜMLERİ	9
2.1. Giriş	9
2.2. Zamanla Değişen Doğrusal Devreler	9
2.3. Periyodik Olarak Zamanla Değişen Doğrusal Devreler	11
2.4. Parça Parça Doğrusal veya Dışardan Kontrollü Elemanlar İçeren Devreler	14
2.4.1. Modelleme	15
2.4.2. Eleman ve sistem durumları	23
2.4.3. Durum ve çıkış denklemleri	25
2.4.4. Anahtarlama denklemleri	30
2.4.5. Kontrol denklemleri	32
2.5. Parça Parça Doğrusal veya Dışardan Kontrollü Elemanlar İçeren Devrelerin Sürekli Hal Çözümü	35

IX

2.5.1. Devrenin sürekli hal davranışının belirlenmesi	36
2.5.2. Sürekli hal çözümü	41
3. BİLGİSAYAR PROGRAMI	47
3.1. Giriş	47
3.2. Akış Şeması	47
3.3. Ana Program	49
3.4. Kullanıcı Programları	52
3.5. TAMCOZ Altprogramı	54
3.5.1. TAMCOZ' un değişken isimleri	54
3.5.2. TAMCOZ' un altprogramları	56
3.6. APEN1 Altprogramı	57
3.6.1. APEN1' in değişken isimleri	58
3.6.2. APEN1' in altprogramları	60
3.7. Programla İlgili Değerlendirmeler	62
4. ÖRNEKLER	63
4.1. Giriş	63
4.2. Örnek 1	64
4.3. Örnek 2	70
4.4. Örnek 3	83
4.5. Örnek 4	94
4.6. Örnek 5	101
4.7. Örnek 6	110
4.8. Örnek 7	116
5. SONUÇLAR	125

X

KAYNAKLAR	128
EK - A PGEDAN PROGRAMININ LİSTESİ	134
EK - B ÖRNEK 1 VE ÖRNEK 7' YE İLİŞKİN KULLANICI PROGRAMLARININ LİSTELERİ	151



XI

ŞEKİLLER VE TABLOLAR

Şekil 2.1	: İdeal anahtar a) açık durumda b) kapalı durumda	16
Şekil 2.2	: İdeal olmayan bir anahtarın eşdeğer devresi	17
Şekil 2.3	: a) Tristör, b) eşdeğeri, c) periyodik olarak çalışan anahtar olarak anahtarlama fonksiyonu.....	18
Şekil 2.4	: (2.20)' deki denklemi temsil eden devre	19
Şekil 2.5	: (2.21)' de verilen denklemi sağlayan devre	19
Şekil 2.6	: a) Diyot karakteristiği ve yapılan yaklaşım, b),c) eşdeğer devreler	20
Şekil 2.7	: Güç diyotunun eşdeğer devresi	21
Şekil 2.8	: Doğrusal olmayan bir indüktans ve B-H karakteristiği	22
Şekil 2.9	: Doğrusal olmayan indüktansın eşdeğeri ve λ -i karakteristiği ..	23
Şekil 2.10	: a) Doğrusal olmayan elemanlar içeren bir devre, b) Doğrusal olmayan elemanların karakteristikleri	25
Şekil 2.11	: Sistemin durum değişimleri için notasyon	26
Şekil 2.12	: n. anahtarlama periyodu için notasyon	37
Şekil 2.13	: Geçiş zamanlarının belirlenmesi	42
Şekil 2.14	: n. anahtarlama periyodu için notasyon	44
Şekil 3.1	: PEGEDAN programının akış şeması	48
Şekil 3.2	: PGEDAN tarafından kullanılan altprogramlar	51
Şekil 3.3	: TAMCOZ tarafından kullanılan altprogramlar	57
Şekil 3.4	: APEN1 tarafından kullanılan altprogramlar	61
Şekil 4.1	: Örnek 1' e ilişkin devre	65
Şekil 4.2	: Sistemin 1. durumu için eşdeğer devre	66
Şekil 4.3	: Sistemin 2. durumu için eşdeğer devre	66

XII

Şekil 4.4	: Tristör akımının (i_L) değişimi	68
Şekil 4.5	: Kondansatör akımının (i_C) değişimi	69
Şekil 4.6	: Kondansatör ve indüktans gerilimlerinin (v_C ve v_L) değişimleri	69
Şekil 4.7	: Tristör geriliminin (v_T) değişimi	70
Şekil 4.8	: Örnek 2'ye ilişkin devre	71
Şekil 4.9	: Sistemin 1. durumu için eşdeğer devre	72
Şekil 4.10	: Sistemin 2. durumu için eşdeğer devre	73
Şekil 4.11	: Armatür akımının değişimi	78
Şekil 4.12	: Açısal hızın değişimi	79
Şekil 4.13	: D_3 diyotu ve kondansatör üzerindeki gerilimlerin değişimleri .	79
Şekil 4.14	: D_3 diyotunun akımının değişimi	80
Şekil 4.15	: Armatür akımının değişimi	81
Şekil 4.16	: Açısal hızın değişimi	81
Şekil 4.17	: D_3 diyotu ve kondansatör üzerindeki gerilimlerin değişimleri .	82
Şekil 4.18	: D_3 diyotunun akımının değişimi	82
Şekil 4.19	: a) Örnek 3' e ilişkin elektrik devresi b) Doğrusal olmayan direncin karakteristiği	84
Şekil 4.20	: Doğrusal olmayan elemanın eşdeğer devresi	85
Şekil 4.21	: Doğrusal olmayan eleman için yeni eşdeğer devre	86
Şekil 4.22	: Sistemin 1. durumu için eşdeğer devre	87
Şekil 4.23	: Sistemin 2. durumu için eşdeğer devre	87
Şekil 4.24	: Giriş, kondansatör ve doğrusal olmayan eleman gerilimlerinin değişimi	91
Şekil 4.25	: İndüktans akımının değişimi	92
Şekil 4.26	: Giriş, kondansatör ve doğrusal olmayan eleman gerilimlerinin değişimi	93

XIII

Şekil 4.27 : İndüktans akımının değişimi	93
Şekil 4.28 : Örnek 4' e ilişkin devre	95
Şekil 4.29 : Doğrusal olmayan direncin karakteristiği ve eşdeğer devresi ..	95
Şekil 4.30 : Sistemin 1. durumu için eşdeğer devre	96
Şekil 4.31 : Sistemin 2. durumu için eşdeğer devre	96
Şekil 4.32 : Kondansatör geriliminin değişimi	99
Şekil 4.33 : Doğrusal olmayan direnç akımının değişimi	99
Şekil 4.34 : Diyot geriliminin değişimi	100
Şekil 4.35 : Diyot akımının değişimi	100
Şekil 4.36 : Örnek 5' de göz önüne alınan devre	101
Şekil 4.37 : Doğrusal olmayan direncin karakteristiği ve eşdeğer devresi ..	102
Şekil 4.38 : Sistemin 1. durumu için eşdeğer devre	102
Şekil 4.39 : Sistemin 2. durumu için eşdeğer devre	103
Şekil 4.40 : Kondansatör geriliminin değişimi	108
Şekil 4.41 : Tristör geriliminin değişimi	108
Şekil 4.42 : İndüktans akımının değişimi	109
Şekil 4.43 : Tristör akımının değişimi	109
Şekil 4.44 : Diyot geriliminin değişimi	110
Şekil 4.45 : Örnek 6' ya ilişkin devre	111
Şekil 4.46 : Doğrusal olmayan indüktansın karakteristiği ve eşdeğer devresi	111
Şekil 4.47 : Sistemin 1. durumu için eşdeğer devre	112
Şekil 4.48 : Sistemin 2. durumu için eşdeğer devre	112
Şekil 4.49 : Giriş gerilimi ve indüktans akımının değişimleri	115
Şekil 4.50 : Giriş gerilimi ve indüktans akımının değişimleri	116
Şekil 4.51 : Örnek 7' ye ilişkin devre	117
Şekil 4.52 : Sistemin 1. durumu için eşdeğer devre	117

XIV

Şekil 4.53	: Sistemin 2. durumu için eşdeğer devre	118
Şekil 4.54	: İndüktans (yük) akımının değişimi	121
Şekil 4.55	: Yük uçlarındaki gerilimin değişimi	121
Şekil 4.56	: T_1 tristörünün uçlarındaki gerilimin değişimi	122
Şekil 4.57	: İndüktans (yük) akımının değişimi	123
Şekil 4.58	: Yük uçlarındaki gerilimin değişimi	123
Şekil 4.59	: T_1 tristörünün uçlarındaki gerilimin değişimi	124
Tablo 3.1	: TAMCOZ altprogramının bazı değişkenleri	55
Tablo 3.2	: APEN1 altprogramının bazı değişkenleri	59



SİMGELER

- A, B : Durum denklemlerine ilişkin katsayı matrisleri
 α, β : Zamanın fonksiyonu olan katsayılar
 $a(t)$: Anahtarlama fonksiyonu
 C, D : Çıkış denklemlerine ilişkin katsayı matrisleri
 C : Kapasitans
 Δt : Tam çözümde kullanılan çözüm adımı
 Δt_2 : Geçiş zamanlarının doğrulanmasındaki çözüm adımı
 F, G : Anahtarlama denklemlerine ilişkin katsayı matrisleri
 ϕ : Durum geçiş matrisi
 f : Frekans
 γ : Elemanlara dışardan yapılan kontrollerin sayısı
 g : İletkenlik
 η, σ : Geçiş zamanlarının gerçek değerleri
 η^1 : Geçiş zamanlarının hesaplanan değerleri
 η^0 : Geçiş zamanlarının yaklaşık değerleri
 i : Akım
 L : İndüktans
 μ : Manyetik permeabilite
 R : Direnç
 τ : Aralık (interval) süreleri
 $u(t)$: Giriş (kaynak) fonksiyonları
 w : Açısal frekans
 x : Durum değişkenleri vektörü
 y : Çıkış büyüklükleri vektörü

1. GİRİŞ

Doğrusal olmayan elemanlar uzun zamandan beri yer aldıkları sistemlerin analizlerini karmaşıklarıtmaktadırlar. Sistemlerde veya devrelerde (burada ve sonraki bölümlerde devre ve sistem kelimeleri aynı anlamda kullanılacaktır) eskiden beri yer alan bu tür elemanlara son yıllarda çok sayıda elektronik ve güç elektroniği devre elemanı eklenmiştir. İlk başlarda güç elektroniği devrelerinde sadece lambalar yer alırken yarı iletken teknolojisindeki gelişmelere paralel olarak önce büyük güçlü diyotlar ve daha sonra tristörler oldukça yaygın olarak kullanılmaya başlanmıştır. Buna örnek olarak, doğru akım motorlarının AC şebekelerden beslenmesine imkan veren kontrolsuz, yarı kontrollu ve kontrollu doğrultucu devreleri, bir DC gerilimden değişik değerlerde DC gerilimler elde edilebilen DC kıyıcı devreleri, aynı şekilde AC gerilimden değişik değerlerde AC gerilimler elde edilebilen AC kıyıcılar, DC gerilimden değişik frekanslarda AC gerilimlerin elde edildiği inverter devreleri ve bir AC gerilimden değişik frekanslarda AC gerilimlerin elde edildiği periyot dönüştürücü devreleri gibi uygulamalar verilebilir (Pearman 1980).

İndüksiyon motorlarının hız kontrolü için Yoshida, Mohri ve Yoshino yüksek kazançlı darbe tetiklemeli güç transistörlerini (PTPT: Puls-Triggered Power Transistor) kullanmışlardır (Yoshida vd. 1986). Aynı şekilde indüksiyon motorlarının hız kontrolünde Addoweesh, mikro işlemcilerle elde ettiği işaretleri sürücü devreler aracılığı ile güç transistörlerinden oluşan inverter devrelerine uygulamıştır (Addoweesh vd. 1989). Bu inverterlerde kullanılan darlington güç transistörleri oldukça hızlı oldukları için açma ve kapama kayıpları düşüktür.

Yer aldıkları sistemlerde daha çok anahtar olarak kullanılan güç elektroniği devre elemanlarının anahtarlama davranışlarının kayıplar açısından

iyileştirilmesi (Herdem 1987) ve anahtarlama frekanslarının artırılması (Bonkowski 1986) gibi birçok çalışma yapılmıştır.

Yarı iletken teknolojisinin daha da gelişmesiyle güç elektroniği devrelerinde kullanılan diyot, tristör, triyak, GTO-tristör (Gate Turn Off-Thyristor) ve güç transistörlerine ek olarak büyük güçleri çok hızlı anahtarlayabilen güç MOSFET'leri, MegaFET'ler ve IGBT (Insulated Gate Bipolar Transistor)'ler üretilmeye başlanmıştır (Harris 19894).

Karakteristikleri doğrusal olmayan güç elektroniği devre elemanlarını, hızlı anahtarlama, küçük bir güçle kontrol edebilme, küçük boyut, sessiz çalışma, düşük maliyet ve uzun ömür gibi çeşitli özellikleri vazgeçilemez hale getirmiştir. Dolayısıyla bu elemanların veya genel olarak doğrusal olmayan elemanların yer aldıkları sistemlerin analizi daha fazla önem kazanmıştır ve bunun için çok sayıda yöntem geliştirilmiştir. Burada özellikle sürekli hal analizi üzerinde durulacaktır.

Doğrusal olmayan sistemler doğrusal sistemlere göre oldukça farklıdır. Örneğin sinüzoidal uyarıya karşı davranışları, sistemin başlangıç şartlarına, uygulanan sinyalin genliğine, frekansına ve fazına bağlı olarak çok farklılıklar gösterebilir. Hatta bunların davranışları her zaman periyodik de olmayabilir (Chua vd. 1982). Bu çalışmada sadece periyodik fonksiyonlarla uyrıldıkları zaman sürekli hal çözümleri de periyodik olan sistemler ele alınmıştır. Dolayısıyla literatürde yer alan doğrusal olmayan sistemlerin sürekli hal çözümleri ile ilgili yöntemler kısaca özetlenecektir.

Doğrusal olmayan bir sistemin periyodik sürekli hal çözümünü bulmak için en basit yol sistemi tanımlayan denklemlerin geçici tepkeleri sönüncüye kadar nümerik integrasyonla çözülmesidir. Bu yöntem "Brute force" (Kaba kuvvet) olarak adlandırılır ve geçici tepkeleri çabuk sönen sistemlerde doğru ve güvenilir sonuçlar verir. Fakat geçici tepkelerinin sönüm zamanı uzun olan sistemlerde sürekli hal tepkesine ulaşıncaya kadar çok fazla nümerik integrasyon gerektiği için

bilgisayar zamanı açısından uygun değildir. Bu nedenle başka analiz yöntemleri de geliştirilmiştir.

Doğrusal olmayan sistemler eğer doğrusal ve doğrusal olmayan parçalara bölünebiliyorsa ve doğrusal olmayan kısım daha küçükse o zaman sinyallerin Fourier serileri ile gösterimini kullanan Harmonik denge metodu kullanılabilir (Nakhla ve Vlach 1976; Filicori vd. 1979). Serinin katsayıları sistemi tanımlayan denklemleri minimum hata ile sağlayacak şekilde belirlenir. Bunun için herhangi bir optimizasyon algoritmasından yararlanılabilir. Çok sayıda doğrusal olmayan eleman içeren veya büyük boyutlu sistemlerde optimize edilecek katsayıların sayısı arttığı için optimizasyon algoritmaları yetersiz kalmakta ve bilgisayar zamanı aşırı derecede uzamaktadır. Ancak benzer bir yaklaşım spektral analiz yöntemi kullanılarak Tohumoğlu tarafından kullanılmış ve optimizasyon veya nümerik integral kullanılmadan iteratif olarak Fourier serilerinin katsayılarının bulunabileceği gösterilmiştir (Tohumoğlu 1993).

Üçüncü bir yöntem de "Shooting methods" olarak bilinen yöntemdir (Aprille ve Branin 1977 ; Nakhla ve Branin 1977 ; Skelboe 1980). Bu yöntemin esası iki noktalı sınır değeri problemini çözmektir. Bunun için üç yaklaşım vardır ve üçü de bir periyot sonraki başlangıç şartlarını belirlemek için uygun bir algoritma kullanır. Bunlardan birincisi nümerik integrasyonla her iterasyon adımında Jakobian matrisinin hesabını gerektiren Newton metodudur. Bu metotta herhangi bir t_0 anındaki başlangıç değerleri ile t_0+T anındaki değerler arasındaki fark en aza indirilmeye çalışılır (Aprille ve Trick 1972). Bu gruptaki ikinci metod nümerik integrasyonla Gradyent vektörünün hesaplanmasıdır. Bu yöntemde problem, optimize edilecek parametreler olarak başlangıç değerlerinin seçildiği bir doğrusal olmayan programlama problemi gibi formüle edilir (Nakhla ve Branin 1977). Üçüncü bir yol Ekstrapolasyon yöntemidir. Bu yöntemde sistemi tanımlayan denklemler çeşitli ekstrapolasyon algoritmalarının uygulanabilmesi için nümerik

olarak integre edilmekte ve bir dizi ilk koşul vektörleri elde edilmektedir (Skelboe 1980). Daha sonra çeşitli ekstrapolasyonlar kullanılarak dizinin gerçek ilk koşullara yakınsaması hızlandırılmaktadır.

Bu son üç metodun karşılaştırılması, (Fattouh vd. 1980) ve (Skelboe 1982)' de yapılmış ve hızlı periyodik sürekli hal metodları olarak nitelendirilmişlerdir.

Literatürde güç elektroniği devrelerinin analizi birçok yazar tarafından ele alınmıştır. Bunlardan bazıları uzun bilgisayar zamanına gerek duyulduğu için genel metodlarla uğraşmayıp özel devrelerin çözümü ile uğraşmışlardır. Singh ve Kohli kıyıcıyla beslenen dışardan uyarımlı bir dc motoru analiz etmişlerdir (Singh ve Kohli 1983). Racz, Baka, Lupan ve Miklas yarı periyodik olan çok parametrelili sistemlerin periyodik sürekli hal şartlarından sapmalarını analiz etmişler, fakat bu sistemlerin sürekli hal çözümünü bulmak için genel bir metod vermemişlerdir (Racz vd. 1969).

Matsu, Othashi, Thunehiro ve Chellamuthu tristörlü devrelerde durum geçiş yaklaşımını kullanmışlardır (Matsu vd. 1979 ; Chellamuthu ve Sastry 1981). Bu yaklaşımda tristörün açık veya kapalı olması durumlarına geçiş zamanlarının tahmini değerlerini doğrulamak için özel bir algoritma kullanılmıştır. Ancak bu sadece kıyıcı ile beslenen bir dc motorun analizine uygulanmıştır. Milias, Zacharias, Hatziaioniu ve Galanos güç elektroniği devrelerini ideal anahtarlar ve doğrusal elemanlardan oluşan eşdeğer devrelerle modelleyerek anahtarların durumlarına göre sistemin durum denklemlerini düzenleyen bir algoritma sunmuşlardır (Milias vd. 1987). Bu durum denklemlerinin uygun nümerik tekniklerle çözülerek güç elektroniği devrelerinin geçici hal davranışlarının analiz edilebileceğini belirtmişlerdir.

Murakami genel olarak anahtarlar ve doğrusal RLC elemanlarından oluşan devrelerin formülasyonu ve çözümü için bir metod vermiştir (Murakami 1987).

Kapalı anahtarları gerilim kaynağı ve açık anahtarları da akım kaynağı kabul ederek durum uzayı formülasyonu kullanmıştır. Anahtarların konum değiştirdiği anlardaki süreksizlikler için Distribüsyon mantığıyla Heaviside ve Drac delta fonksiyonlarını kullanmıştır. Ayrıca anahtarların kontrol edilebilirliği ve gözlenebilirliğini de incelemiştir. Ancak Murakami' nin bu çalışmalarının büyük bir kısmı yaklaşık on yıl önce Köksal tarafından yapılmış ve yayınlanmıştır (Köksal 1975 ; Köksal ve Tokad 1976 ; Köksal ve Tokad 1977).

Huang ve Liu parça parça doğrusal sistemlerin çözümü için basit bir algoritma sunmuşlardır (Huang ve Liu 1989). Thevenin / Norton eşdeğer devreleri için tanımladıkları yeni bir teoreme dayalı olarak geliştirdikleri algoritmayı kullanmışlardır. Opal ve Vlach periyodik olarak anahtarlanan doğrusal devrelerin frekans domeni analizini ele almışlardır (Opal ve Vlach 1989). Nümerik Laplas dönüşümü kullanarak çok doğru sonuçlar elde etmişlerdir.

Peryodik olarak uyarıldıkları zaman sürekli hal davranışları periyodik olan doğrusal olmayan sistemlere her zamankinden farklı olarak ilk yaklaşım Köksal tarafından yapılmıştır (Köksal 1984b ; Köksal 1984c). Onun yaklaşımında bu şekildeki sistemler periyodik olarak zamanla değişen sistemler olarak göz önüne alınmıştır. Bu yaklaşım ne şutlama metodundaki gibi nümerik integrasyona ne de harmonik denge metodundaki gibi optimizasyona gerek duymaz. Doğrudan periyodik olarak zamanla değişen sistemler için geliştirilen analiz yöntemleri kullanılır. Peryodik uyarı olarak da sadece sinüzoidal uyarılar değil Fourier dizisi ile gösterimlenebilen bütün periyodik uyarılar göz önüne alınır.

Güç elektroniği devrelerinin analizi için Türkbeyler, Köksal' ın yukarıda bahsolunan metodundan yararlanarak bir yaklaşımda bulunmuştur (Türkbeyler 1987). Doğrusal olmayan elemanların parça parça doğrusal elemanlar olarak göz önüne alındığı bu yaklaşımda doğrusal olmayan elemanlar, doğrusal elemanlar ve periyodik olarak çalışan anahtarlardan oluşan eşdeğer devrelerle modellenmiştir.

Her doğrusal olmayan elemanın birden fazla farklı duruma sahip olduğu ve bu durumlarda kalma sürelerinin sistem cevabı tarafından belirlendiği kabul edilmiştir. Ana peryot sistemin zamanla değişmeyen doğrusal sistem olarak kaldığı alt aralıklara bölünerek analiz gerçekleştirilmektedir.

Bu yaklaşımın literatürdeki diğer metodlara göre en iyi yanı nümerik integrasyon kullanmadan doğrudan sürekli hal çözümünü elde etmesidir. Bunun için APEN olarak adlandırılan bir bilgisayar programı hazırlanmıştır. Peryodik olarak zamanla değişen sistemlerin analizi için hazırlanmış olan genel bilgisayar programı MAINLN' in bazı kısımlarının düzenlenmesi ve buna ek olarak bilinmeyen aralık sürelerinin belirlenmesi ve çok değişkenli (çok girişli - çok çıkışlı) sistemlerin de analizi gibi yeniliklerin eklenmesi ile APEN programı oluşmuştur.

Bu son yaklaşımda da bazı eksiklikler bulunmaktadır. Analizi yapılacak olan sistemin kaç durumunun bulunduğu ve bu durumlara hangi sırayla girerek bir peryottaki aralıkları oluşturduğunun kesin olarak ve aralıklarda kalma sürelerinin de yaklaşık olarak bilinmesi gerekmektedir. Ayrıca sistemin bulunacağı bütün aralıklarda aynı sayıda durum değişkenine sahip olacağı kabul edilmiştir. Bazı özel uygulamalarda yukardaki ön bilgiler tahminen belirlenebilirse de biraz karmaşık sistemlerde veya aynı sistemin değişik parametrelerle analizinde bu mümkün değildir.

Güç elektroniği devrelerinin veya genel olarak parça parça doğrusal veya dışardan kontrollü elemanların yer aldığı devrelerin sürekli hal analizi için bu tezde yer alan metod, yukarıda bahsolunan metodu da kapsayan genel bir uygulamadır. Bir kısmı APEN' in genelleştirilmiş halinden oluşan PGEDAN isimli bir bilgisayar programı hazırlanmıştır. Bu program, önce analizi yapılan sistemin sürekli halde girdiği durumları ve bu durumlara geçiş zamanlarının yaklaşık değerlerini belirlemekte, daha sonra geçiş zamanlarının gerçek değerlerini belirleyerek sürekli hal çözümünü bulmaktadır. Sürekli haldeki sistem durumları

bulunurken sistemi tanımlayan denklemlerin tam (geçici hal + sürekli hal) çözümünü yapılmaktadır. Bu çözümde de nümerik integrasyon kullanılmayıp her t değeri için çözüm doğrudan elde edilmiştir. Dolayısıyla büyük çözüm adımları kullanılarak çok kısa sürede sonuç elde edilebilmektedir.

Sistemi tanımlayan durum ve çıkış denklemlerinde kaynak fonksiyonlarının türevlerinin yer alması veya her aralıkta farklı sayıda durum değişkeni bulunması gibi hallerde de kullanılabilir şekilde genel bir bilgisayar programı olan PGEDAN üçüncü bölümde detaylı olarak anlatılmıştır.

Tezle ilgili bu kısa bilgilerden sonra tezin alt bölümleriyle ilgili olarak şu açıklamalar yapılabilir:

İkinci bölümde, önce doğrusal olmayan sistemler, zamanla değişen doğrusal sistemler ve periyodik olarak zamanla değişen sistemler hakkındaki temel bilgiler özet olarak verilmiştir. Sonra parça parça doğrusal veya dışardan kontrollü elemanlar içeren devrelerin periyodik çözümlerinin bulunması için gerekli olan işlemler anlatılmış ve bu işlemlerle ilgili formülasyon verilmiştir. Ayrıca söz konusu elemanların modellenmesi de bu bölümde verilmiştir.

Üçüncü bölümde, hazırlanan bilgisayar programı tanıtılmıştır. Genel açıklamalar ve akış şemasından sonra kullanılan altprogramlar, kullanıcı programları ve bunların değişkenleri anlatılmıştır. Bu bölümün sonunda programın hafıza gereksinimi, çalışma süresi gibi bilgiler verilmiştir.

Dördüncü bölümde, geliştirilen yöntemi çeşitli yönleriyle ortaya koyabilmek için değişik örnekler verilmiştir. Yenilikleri ve bazı özellikleri açıkça gösterebilmek için örnek sayısı özellikle fazla seçilmiştir. Bazı örnekler değişik modelleme veya değişik sistem parametreleriyle tekrar çözülerek sonuçlar karşılaştırılmıştır. İlk üç örnek daha önce literatürde yer alan örnekler arasından seçilmiş ve elde edilen sonuçlar doğrulanmıştır.

Beşinci bölümde, tezle ilgili sonuçlar özetle verilmiştir. Yani, bu yöntemin diğer yöntemlere göre üstünlüklerinden, yöntemin uygulanması sırasındaki sınırlamalardan ve daha sonra yapılacak çalışmalara ışık tutacak olan tavsiyelerden bahsedilmiştir.

Tezin sonunda ise hazırlanan bilgisayar programı ve dataların bilgisayara girilmesi açısından bir fikir vermesi için de birinci ve yedinci örneklere ilişkin kullanıcı programları ek olarak verilmiştir.



2. DOĐRUSAL OLMAYAN DEVRELERİN PERİYODİK ÇÖZÜMLERİ

2.1. Giriş

Oldukça kapsamlı ve karmaşık olan doğrusal olmayan sistemlerin burada tamamının incelenmesi mümkün olmadığı gibi aslında bu çalışmanın amacının da dışındadır. Bu nedenle sadece, periyodik olarak uyarıldıkları zaman kalıcı tepkeleri de periyodik olan doğrusal olmayan sistemler incelenecektir. Bu tür sistemlerin periyodik sürekli hal çözümlerinin bulunması için literatürde yer alan bazı metodlar birinci bölümde özetlenmişti ve bunların içerisinde Köksal'ın metodunun oldukça önemli olduğu belirtilmişti.

Bu çalışmada ele alınan sistemler ve uyarı fonksiyonları da Köksal'ın metodunun uygulanabileceği şekildedir. Bu nedenle konular fazla dallandırılmadan adı geçen metoda temel teşkil eden zamanla değişen ve periyodik olarak zamanla değişen doğrusal devrelerin genel bir özeti verilecektir. Daha sonra da parça parça doğrusal veya dışardan kontrollü elemanlar içeren devreler incelenecektir.

2.2. Zamanla Değişen Doğrusal Devreler

Periyodik olarak zamanla değişen doğrusal devre kavramının daha iyi anlaşılabilmesi için ilk olarak zamanla değişen doğrusal devrelere kısaca değinmek gerekmektedir. Zamanla değişen bir devrede parametreler zamanın fonksiyonudur. Bu şekildeki devrelerin en basit tipi aşağıdaki gibi m . dereceden lineer diferansiyel denklemle ifade edilir.

$$\left[\alpha_m(t) \frac{d^m}{dt^m} + \alpha_{m-1}(t) \frac{d^{m-1}}{dt^{m-1}} + \dots + \alpha_1(t) \frac{d}{dt} + \alpha_0(t) \right] u(t) = \left[\beta_m(t) \frac{d^m}{dt^m} + \dots + \beta_1(t) \frac{d}{dt} + \beta_0(t) \right] y(t) \quad (2.1)$$

Burada, $u(t)$ ve $y(t)$ giriş ve çıkış işaretlerini gösterir. α' lar ve β' lar zamanın sürekli fonksiyonudurlar (Nacaroğlu 1989). Eğer sistemde birden fazla giriş ve çıkış varsa o zaman birden fazla yüksek mertebeden diferansiyel denklem olacaktır. Bu da birçok zorluklar getirecektir. Buna rağmen daha geniş kapsamlı uygulamalara imkan vermek için bu çalışmada çok girişli, çok çıkışlı durum gözönüne alınmıştır.

Çoğu zaman giriş ve çıkış değişkenlerini birbirine bağlayan bir diferansiyel denklem yazılması tercih edilirse de sistemin durum uzay tanımı daha elverişlidir. Durum uzayında hem birinci dereceden diferansiyel denklemler olan durum denklemleri takımı hem de çıkış denklemleri takımı kullanılır. Böylece N tane birinci dereceden doğrusal diferansiyel denklem içeren durum denklemleri ve çıkış denklemleri,

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \quad (2.2)$$

$$y(t) = C(t)x(t) + D(t)u(t) \quad (2.3)$$

şeklinde yazılır. Burada $x(t)$, N elemanlı durum vektörü, $u(t)$, N_f elemanlı giriş vektörü ve $y(t)$, m elemanlı çıkış vektörüdür. $A(t)$, $B(t)$, $C(t)$ ve $D(t)$, tüm t değerleri için parça parça sürekli olacağı farzedilen skaler matrislerdir. Bu matrislerin boyutları sırasıyla $N \times N$, $N \times m$ ve $m \times N_f$ kadardır. Kısaca bu denklemler $R = [A(\cdot), B(\cdot), C(\cdot), D(\cdot)]$ sistem gösterimi olarak gösterilebilirler.

Başlangıç durumu $x(t_0)$ ve $t \geq t_0$ için (2.2) denkleminin tam çözümü,

$$x(t) = \Phi(t - t_0) x(t_0) + \int_{t_0}^t \Phi(t, \tau) B(\tau) u(\tau) d\tau \quad (2.4)$$

ile verilir. Burada $\Phi(t, \tau)$, $A(t)$ ' ye ilişkin durum geçiş matrisidir (Desoer 1970). Bir $x(t_0)$ başlangıç durumu, $u(t)$ girişi ve tüm $t \geq t_0$ için R sisteminin tam çözümü, denklem (2.4)' ü denklem (2.3)' de yerine yazarak

$$y(t) = C(t) \Phi(t - t_0) x(t_0) + C(t) \int_{t_0}^t \Phi(t, \tau) B(\tau) u(\tau) d\tau + D(t) u(t) \quad (2.5)$$

şeklinde bulunur. Burada sağ taraftaki ilk terim sıfır giriş cevabı ve ikinci terim sistemin sıfır durum cevabıdır. Bu cevaplar homojen ve özel çözüm olarak da adlandırılır. Durum geçiş matrisi bazı özel hallerde,

$$\Phi(t, \tau) = \exp \int_{\tau}^t A(\tau) d\tau \quad (2.6)$$

şeklinde ifade edilir. $\Phi(t, \tau)$ genelde analitik olarak çözülemez. Hangi durumlarda analitik çözümlerin varlığı ile ilgili detaylı bilgi literatürde bulunabilir (Wu 1980a ; Wu 1980b). Ancak A sabit bir matris ise (2.6) her zaman geçerlidir.

Zamanla değişen doğrusal devrelerle ilgili buraya kadar verilen temel kavramlar ileriki bölümler için yeterlidir. Bundan sonra bu çalışmada esas alınan periyodik olarak zamanla değişen devreler incelenecektir.

2.3. Periyodik Olarak Zamanla Değişen Doğrusal Devreler

Periyodik olarak zamanla değişen doğrusal devreler, periyodik katsayılı doğrusal diferansiyel denklemlerle karakterize edilirler. Zamanla değişen doğrusal

devreleri temsil eden (2.1) diferansiyel denkleminde $\alpha(t)$ ve $\beta(t)$ katsayıları, (2.2) ve (2.3) durum uzayı gösterimindeki $A(t)$, $B(t)$, $C(t)$ ve $D(t)$ katsayı matrisleri ana periyot T ile periyodiktirler. Daha açık olarak, $\alpha(t+T) = \alpha(t)$, $\beta(t+T) = \beta(t)$, $A(t+T) = A(t)$, $B(t+T) = B(t)$, $C(t+T) = C(t)$ ve $D(t+T) = D(t)$ eşitlikleri bütün t değerleri için geçerlidir. Denklem (2.2)'nin durum geçiş matrisi,

$$\Phi(t, \tau) = P(t, \tau) e^{M(t-\tau)} \quad (2.7)$$

olarak yazılabilir. Burada $P(t, \tau)$, T periyoduyla periyodik bir matris; M ise, aşağıdaki şekilde tanımlanan bir sabit matris olup,

$$M = \frac{1}{T} \ln[\Phi(T, 0)] \quad (2.8)$$

ve $P(t, \tau) = I$ 'dir (Zadeh 1961).

$$X(t) = P(t, 0) z(t) \quad (2.9)$$

dönüşümü ile yeni durum uzay gösterimi,

$$\dot{z}(t) = Mz(t) + \bar{B}u(t) \quad (2.10)$$

$$y(t) = \bar{C}(t)z(t) + \bar{D}(t)u(t) \quad (2.11)$$

şeklinde elde edilir. Buradaki yeni $\bar{B}(t)$, $\bar{C}(t)$ ve $\bar{D}(t)$ matrisleri hala T periyodu ile periyodiktirler ve

$$\bar{B}(t) = P^{-1}(t, \tau) B(t) = \sum_{k=-\infty}^{\infty} B_k e^{j\omega_0 k t} \quad (2.12)$$

$$\bar{C}(t) = C(t) p(t, \tau) = \sum_{l=-\infty}^{\infty} C_l e^{-j\omega_0 l t} \quad (2.13)$$

$$\bar{D}(t) = D(t) = \sum_{v=-\infty}^{\infty} D_v e^{j\omega_0 v t} \quad (2.14)$$

şeklinde ifade edilir. Bu denklemlerde $w_0=2\pi/T$ olup B_k, C_l, D_v sırayla $\bar{B}(t), \bar{C}(t), \bar{D}(t)$ periyodik matrislerinin kompleks Fourier serisi katsayılarını gösterirler. Bu sonuçlar şunu gösteriyor ki: Herhangi bir periyodik sistem, periyodik giriş çarpanları ($\bar{B}(t)$) ve periyodik çıkış çarpanları ($\bar{C}(t), \bar{D}(t)$) ile sabit bir sistem (M matrisi) olarak gösterilebilir (Köksal 1975). Bu sonuç pratik uygulamalar için çok önemlidir. (2.5),(2.6) sonuçları ve (2.10), (2.11) eşdeğer sistem gösterimi kullanılarak periyodik sistemin $x(t_0=0)=x_0$ başlangıç koşulu ve bir $u(t) = u e^{pt}$, ($p = jw$) girişi için toplam cevabı,

$$y(t) = \bar{C}(t) e^{Mt} x_0 + \bar{C}(t) e^{Mt} \int_0^t e^{-M\tau} \bar{B}(\tau) U e^{p\tau} d\tau + \bar{D}(t) U e^{pt} \quad (2.15)$$

olarak elde edilir. M' nin tüm özdeğerlerinin farklı olduğu kabulü ile e^{Mt} ,

$$e^{Mt} = \sum_{i=1}^N M_i e^{\lambda_i t} \quad (2.16)$$

olarak yazılabilir. Burada λ_i ' ler M' nin özdeğerleri M_i de $(sI-M)^{-1}$ 'in λ_i kutbundaki kalıntısıdır. Denklem (2.14) ve (2.16), (2.15)' de kullanılarak

$$y(t) = \left(\sum_{l=-\infty}^{\infty} C_l e^{-jw_0 l t} \right) \sum_{i=1}^N M_i \left[x_0 - \left(\sum_{k=-\infty}^{\infty} \frac{B_k}{p - \lambda_i + jw_0 k} \right) u \right] e^{\lambda_i t} + \left[\left(\sum_{i=1}^N \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \frac{C_l M_i B_k}{p - \lambda_i + jw_0 k} e^{jw_0(k-l)t} \right) + \sum_{v=-\infty}^{\infty} D_v e^{jw_0 v t} \right] U e^{pt} \quad (2.17)$$

toplam çözüm sonucu elde edilir. Sıfır giriş ve sıfır durum cevapları (2.17)' de $U=0$ ve $x_0=0$ alınarak belirlenebilir.

Bu çalışmada sürekli hal çözümleri üzerinde durulduğundan sistemler asimtotik olarak kararlı kabul edilecektir (sıfır giriş cevapları $t \rightarrow \infty$ halinde sıfır

olacak). Bunun için (2.17) denkleminde λ_i ' lerin reel kısımlarının negatif olmasının yeterli olduğu görülür. Ayrıca aynı denklemden kalıcı çözümün,

$$y(t) = H(p,t) U e^{pt} \quad (2.18)$$

şeklinde olduğu da görülür. Burada,

$$H(p,t) = \sum_{v=-\infty}^{\infty} \left[\sum_{i=1}^N \sum_{k=-\infty}^{\infty} \frac{C_{k-v} M_i B_k}{p - \lambda_i + j \omega_0 k} + D_v \right] e^{jv\omega_0 t} = \sum_{v=-\infty}^{\infty} H_v(p) e^{jv\omega_0 t} \quad (2.19)$$

nin (zamanla değişen transfer fonksiyonu) t zamanına göre periyodik olması önemlidir. Bu husus, $p=j\omega$ giriş frekansındaki bir uyarının çıkışta $p+jv\omega_0$, $v=0, +1, +2, \dots$ şeklinde frekans bileşenlerine ayrıştığını ve her çıkış bileşeninin giriş bileşeni ile arasındaki transfer fonksiyonunun $H_v(p)$ olduğunu ortaya çıkarır.

(2.18) ve (2.19) denklemleri ile ifade edilen yukardaki açıklamalar periyodik olarak zamanla değişen doğrusal sistemlerin en önemli özelliklerini oluşturur.

Buraya kadar zamanla değişen ve periyodik olarak zamanla değişen doğrusal devreler hakkındaki temel teori gözden geçirildi. Bundan sonra, parça parça doğrusal veya dışardan kontrollü elemanlar içeren devreler incelenecektir.

2.4. Parça Parça Doğrusal veya Dışardan Kontrollü Elemanlar İçeren Devreler

Parça Parça doğrusal veya dışardan kontrollü elemanlar içeren devreler aslında doğrusal olmayan devrelerdir. Ancak bu çalışmada ele alınan doğrusal

olmayan devreler, periyodik olarak uyarıldıkları zaman sürekli halde periyodik olarak çıkış verirler. Bu şekildeki devrelerin, periyodik olarak zamanla değişen doğrusal devreler gibi analiz edilebileceği ve bu yaklaşımın da ilk olarak Köksal (1984b) tarafından kullanıldığı önceki bölümlerde anlatılmıştı.

Devrede yer alan parça parça doğrusal elemanlar, sadece anahtarlar ve doğrusal elemanlar kullanılarak modellenebilirler. Anahtarların bazılarının açık bazılarının kapalı olmalarıyla oluşan her bir kombinasyon, parça parça doğrusal elemanların herbir doğrusal parçasına karşılık gelir. Devrenin tamamının davranışı periyodik olduğu için bu yaklaşımla devrede yer alacak anahtarların çalışmaları da periyodik olacaktır. Bu nedenle söz konusu devreler, periyodik olarak çalışan anahtarlar içeren devreler gibi analiz edilebilirler. Bu tip devrelerin analizi için ise gerekli tüm teorik çalışmalar hemen hemen yapılmış gibidir (Köksal ve Tokad 1977 ; Köksal ve Tokad 1976). Ayrıca, bu metodla parça parça doğrusallık özelliği göstermeyen değişimlere sahip doğrusal olmayan devrelerin analizi de belirli hata sınırları içerisinde mümkün olmaktadır.

2.4.1. Modelleme

Kapalı durumda küçük bir direnç, açık durumda büyük bir direnç gibi iki farklı değer gösteren iki uçlu elemana anahtar denir. Bu dirençler, anahtarın ileri yön ve geri yön dirençleri olarak da adlandırılırlar. Bir a anahtarının çalışması $a(t)$ anahtarlama fonksiyonu ile tanımlanabilir.

$$a(t) = \begin{cases} 1 & a \text{ kapalı ise} \\ 0 & a \text{ açık ise} \end{cases}$$

Bu anahtarın tersi olarak çalışan başka bir \bar{a} anahtarının anahtarlama fonksiyonu olan $\bar{a}(t)$ ise

$$\bar{a}(t) = 1 - a(t)$$

şeklinde tanımlanır. Eğer anahtarlama fonksiyonu T periyodu ile periyodik ise bu anahtara Periyodik Olarak Çalışan Anahtar denilir (Köksal 1975). O zaman anahtarlama fonksiyonu için,

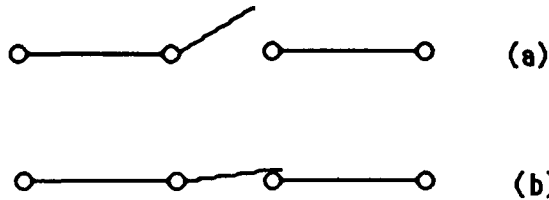
$$a(t) = a(t + T)$$

yazılabilir. Bir anahtarın kapalı durumdaki direnci sıfır ve açık durumdaki direnci sonsuz ise böyle bir anahtara İdeal Anahtar denilir (Köksal 1975). İdeal bir anahtarın şematik gösterimi Şekil 2.1'de verilmiştir.

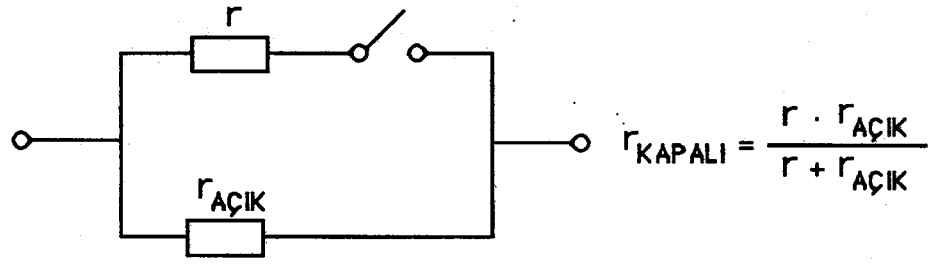
İdeal anahtarlardan başka bir de ideal olmayan anahtarlar tanımlanabilir. İdeal olmayan anahtarlar bir ideal anahtar ve iki direnç yardımıyla simüle edilebilirler. Şekil 2.2' de eşdeğer devresi verilen ideal olmayan bir anahtar,

$$r(t) = r_{KAPALI} \cdot a(t) + r_{AÇIK} \cdot \bar{a}(t)$$

direnci ile zamanla değişen bir dirence eşdeğerdir.



Şekil 2.1 ideal anahtar a) açık durumda b) kapalı durumda

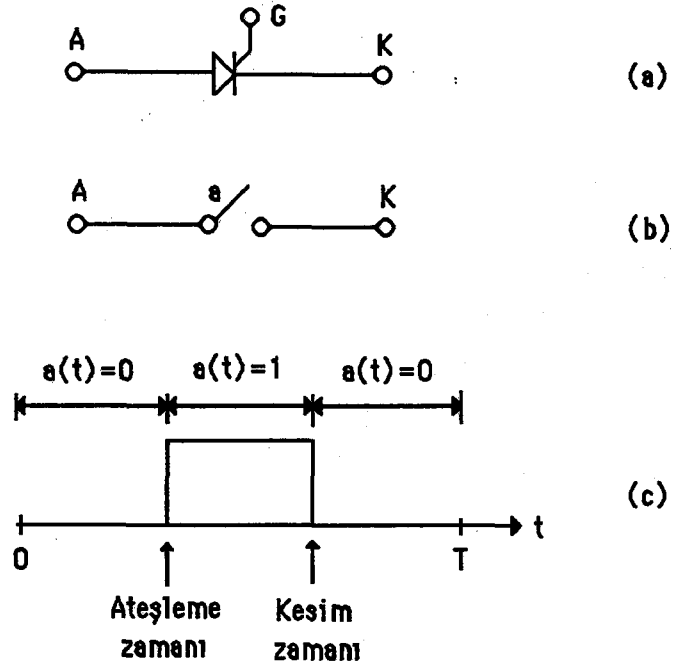


Şekil 2.2. İdeal olmayan bir anahtarın eşdeğer devresi

Pratikte kullanılan anahtarlar, burada tanımlananlardan ne ideal olan ne de ideal olmayan anahtarlara tam olarak benzemezler. Pratik anahtarlar için, anahtarın bir durumdan diğer duruma geçme zamanı sıfır değildir. Bu nedenle her zaman için AÇIK ve KAPALI durumlarına ek olarak bir de geçiş durumu gözönüne alınmalıdır. Geçiş durumları ile ilgili olarak Yükselme ve Düşme zamanları tanımlanır. Yarı iletken anahtarlar için sıfır olmayan geçiş zamanlarını temsil etmek üzere ideal olmayan anahtarın eşdeğer devresinde açık durum direncine paralel olarak küçük bir kapasite bağlanabilir (Nacaroğlu 1989).

Uygulamada ikiden daha fazla durumlu veya ikiden fazla uçlu anahtarlarla da karşılaşılabilir. Bu anahtarlar, sadece iki ucu bulunan ve uygun anahtarlama fonksiyonuna sahip olan anahtar devreleri yardımıyla simüle edilebilirler (Köksal 1975).

Bu çalışmada parça parça doğrusal elemanlar olarak daha çok güç elektroniği devrelerinde kullanılan yarı iletken elemanlar göz önüne alınmıştır. Bu nedenle yarı iletken elemanların modellenmesine de değinilecektir. Diyot, Tristör, GTO Tristör (Gate Turn-Off Tristör), ... gibi güç elektroniği elemanlarının akım-gerilim karakteristikleri doğrusal değildir. Bu elemanların yer aldıkları devrelerin periyodik olarak çalışan anahtarlar içeren devreler gibi analiz edilebilmeleri için bu elemanların yerine eşdeğer devrelerinin göz önüne alınması gerekir. Bir güç diyotu veya tristör için basit bir eşdeğer olarak Şekil 2.3' deki gibi ideal anahtar kullanılabilir.



Şekil 2.3 a) Tristör, b) eşdeğeri, c) periyodik olarak çalışan anahtar olarak anahtarlama fonksiyonu

Yarı iletken elemanlar sadece anahtarlar tarafından temsil edilebilecekleri gibi RLC elemanları ve sabit akım ve gerilim kaynaklarından oluşan eşdeğer devreler tarafından da temsil edilebilirler.

Yarı iletken elemanların iletme geçme ve kesime gitme zamanları ihmal edilebiliyorsa bu elemanların reaktif etkileri çok önemli değildir. Dolayısıyla yarı iletken elemanların eşdeğer devrelerini elde etme için i-v karakteristiklerinde parça-parça doğrusallık yaklaşımı kullanılabilir. Zaten yarı iletken elemanların çoğu parça-parça doğrusallık yaklaşımına çok uygun olan i-v karakteristiklerine sahiptirler. Bu yaklaşımla elde edilecek eşdeğer devreler basit bir anahtardan daha doğru sonuçlar verir. Karakteristiklerdeki her doğrusal bölge için eğrinin denklemini aşağıdaki gibi yazılabilir.

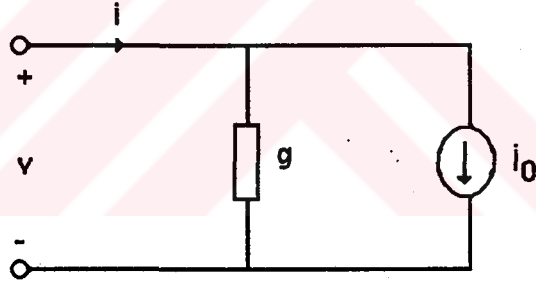
$$i = g v + I_0 \quad (2.20)$$

(2.20) denklemini sağlayacak olan eşdeğer devre Şekil 2.4' de verilmiştir. Bu denklem eşdeğer olarak,

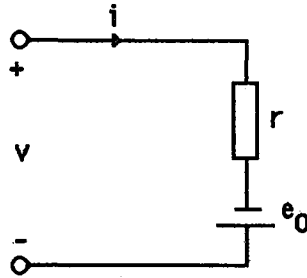
$$v = \frac{i - I_0}{g} = r i - e_0 \quad (2.21)$$

şeklinde de yazılabilir. Burada $r=1/g$ ve $e_0=I_0/g'$ dir. (2.21) denklemini sağlayan eşdeğer devre de Şekil 2.5' de verilmiştir.

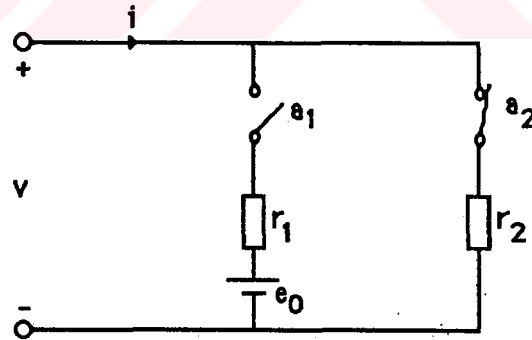
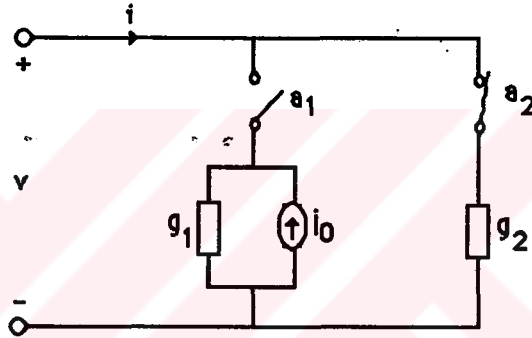
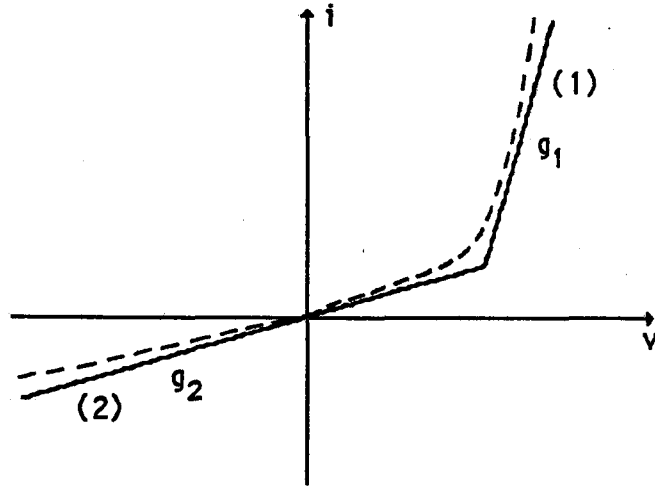
Parça parça doğrusallık yaklaşımı metodu ile eşdeğer devre bulunmasına örnek olarak Şekil 2.6a' daki güç diyotu karakteristiği alınabilir. Kesik çizgilerle gösterilen eğri, gerçek karakteristiği göstermektedir. Şekil 2.6 b, c' de ise, eşdeğer devreler görülmektedir. 1. bölgede a_1 kapalı ve a_2 açıktır. 2. bölgede anahtarlar konum değiştirirler. Devre parametreleri g_1 , g_2 ve I_0 , birleştirilen doğru parçalarının i -ekseni ile kesiştiği noktaları ve eğimlerinden kolayca belirlenebilir. r_1 , r_2 , ve e_0 ise sırayla $r_1=1/g_1$, $r_2=1/g_2$ ve $e_0=I_0/g_1$ eşitlikleriyle belirlenirler.



Şekil 2.4 (2.20)' deki denklemi temsil eden devre



Şekil 2.5 (2.21)' de verilen denklemi sağlayan devre



Şekil 2.6 a) Diyot karakteristiği ve yapılan yaklaşım, b),c) eşdeğer devreler

Yarı iletken elemanların transient davranışlarının önemli olduğu durumlarda iletme geçme ve kesime gitme zamanları ihmal edilemeyebilir. Buna hem uyarma sinyallerinin periyotlarına hem de devrenin zaman sabitlerine göre

karar verilir. Eğer yarı iletkenle ilgili zaman değerleri, devreyle ilgili zamanlar yanında ihmal edilemiyorsa o zaman yukarıda verilen eşdeğer devreler yeterli olmayabilirler. Bu durumda eşdeğer devrelerde kapasitans ve indüktans gibi reaktif elemanlar da yer almalıdır. Bu şekildeki eşdeğer devreler bazı yazarlar tarafından geliştirilmiştir ve literatürde bulunabilirler (Williams 1977 ; Sorker ve Mukhopadhyay 1982 ; Mukhopadhyay ve Mitra 1980 ; Gray ve Searle 1969 ; Rajagopalan 1987). Bir fikir vermesi açısından Şekil 2.7' de güç diyotu için bu şekilde oluşturulmuş bir eşdeğer devre verilmiştir.

Doğrusal olmayan reaktif elemanların doğrusal elemanlar ve anahtarlar kullanılarak modellenmesi de, cebirsel elemanların direnç, anahtar ve sabit kaynaklar kullanarak modellenmesine benzer şekilde mümkün olmaktadır. Buna örnek olarak Şeki 2.8' de karakteristiği verilen doğrusal olmayan bir indüktansın eşdeğeri gösterilebilir.

Şekil 2.8' deki karakteristikte,

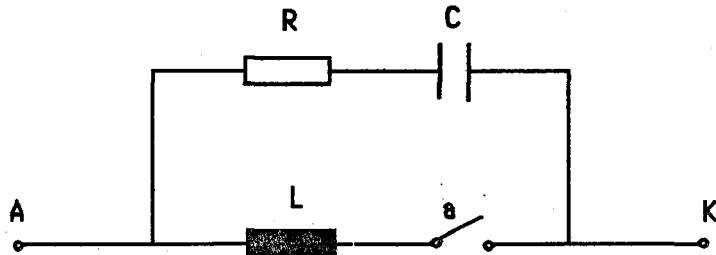
1. durum için $B = \mu_1 H + B_0$

2. durum için $B = \mu_2 H$

3. durum için $B = \mu_1 H - B_0$

yazılabilir. Bu ifadelerden ve

$$\Phi = B A , \quad \lambda = N \Phi = N B A , \quad H = (L / \mu N A) I$$



Şekil 2.7 Güç diyotunun eşdeğer devresi

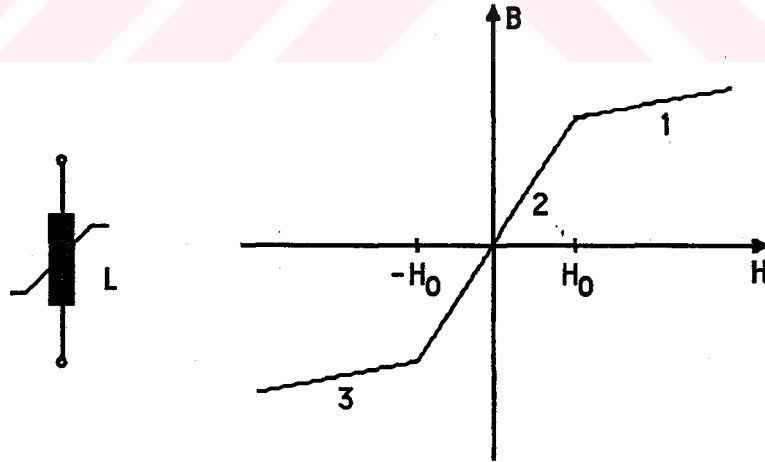
bağıntılarından yararlanılarak,

$$v = \frac{d\lambda}{dt} = N A \frac{dB}{dt} = N A \begin{cases} \mu_1 \frac{dH}{dt} \\ \mu_2 \frac{dH}{dt} \\ \mu_1 \frac{dH}{dt} \end{cases} = \begin{cases} L_1 \frac{di}{dt} \\ L_2 \frac{di}{dt} \\ L_1 \frac{di}{dt} \end{cases} \quad (2.22)$$

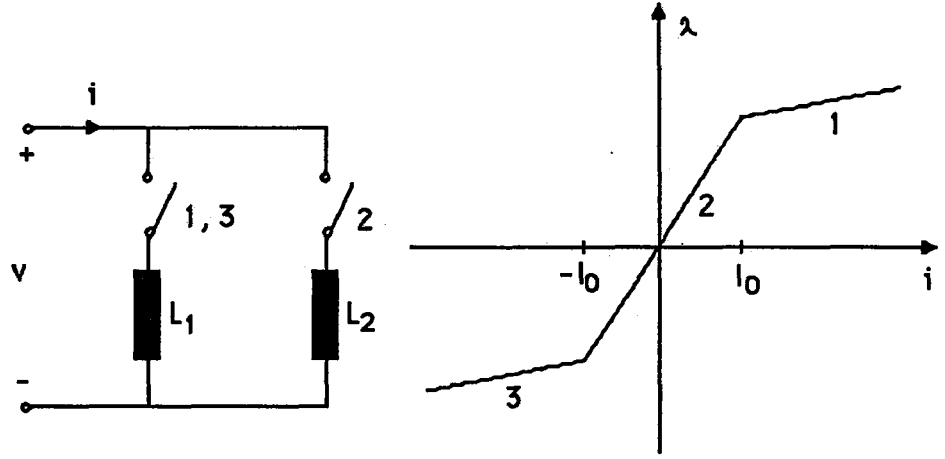
ifadesi elde edilir. Bu ifadelerin ışığında Şekil 2.8' deki doğrusal olmayan indüktansın doğrusal indüktanslar ve anahtarlardan oluşan eşdeğeri Şekil 2.9' daki gibi elde edilir. Şekil 2.9' da aynı zamanda eşdeğer devrenin karakteristiği de verilmiştir. Bu karakteristikte,

1. durum için , $\lambda = L_1 i + \lambda_0$
2. durum için , $\lambda = L_2 i$
3. durum için , $\lambda = L_1 - \lambda_0$

yazılabilir.



Şekil 2.8 Doğrusal olmayan bir indüktans ve B-H karakteristiği



Şekil 2.9 Doğrusal olmayan indüktansın eşdeğeri ve λ -i karakteristiği

2.4.2. Eleman ve sistem durumları

Doğrusal olmayan elemanların yer aldığı devrelerdeki doğrusal olmayan elemanların karakteristiklerine parça parça doğrusallık yaklaşımı yapıldığı yukarda anlatılmıştı. Bu yaklaşımla elde edilen yeni karakteristiklerdeki farklı eğimlere sahip olan her bir doğru parçası o eleman için bir eleman durumuna karşılık gelmektedir. Devredeki her eleman için belirlenen bu eleman durumlarının değişik kombinasyonlarla bir araya gelmesiyle de sistem durumları oluşmaktadır. N_c , devrede yer alan parça parça doğrusal elemanların sayısını, ve \bar{c}_i de her parça parça doğrusal elemanın eleman durumu sayısını göstermek üzere sistemin girebileceği durumların sayısı S_m , en fazla

$$S_m = \prod_{i=1}^{N_c} \bar{c}_i = \bar{c}_1 \cdot \bar{c}_2 \dots \bar{c}_{N_c} \quad (2.23)$$

olarak bulunur. Bu denklemden de anlaşılacağı gibi herhangi bir elemanın bir durumu diğer elemanların bütün durumlarıyla bir arada olabilecek şekildeki

kombinasyonların tamamı değişik sistem durumları sayısı için en yüksek ihtimali oluşturmaktadır. Ancak değişik elemanlar sistemi oluşturacak şekilde birbirlerine bağlandığında, elemanların girebileceği durumlar birbirleriyle ilişkili olmak zorundadır. Dolayısıyla fiziksel olarak sistemin girmesi mümkün olabilecek durum sayısı (S) S_m ' den az olabilir. Sistemdeki bağımsız kaynakların ve sistem parametrelerinin seçilmiş özel değerlerine bağlı olarak sistem S sistem durumundan yalnızca bazılarına girebilir. Bunun nedeni, sistemin sürekli hal davranışının henüz bilinmiyor olmasıdır. Başlangıçta fiziksel olarak mümkün olan bütün ihtimaller (S) göz önüne alınarak işlemlere başlanmaktadır. Daha sonra çözüm işlemleri devam ederken her çözüm adımından sonra elemanların durumları ayrı ayrı belirlenmekte ve bu eleman durumlarına karşılık gelen sistem durumu belirlenmektedir. Bu şekilde devam eden işlemler sonunda sürekli halde girilen sistem durumları belirlenir ve sürekli hal çözümüne geçilir.

Eleman durumlarına göre sistemin durumunun belirlenmesinde

$$W = [w_{i,j}] \in N^{S \cdot N_c}$$

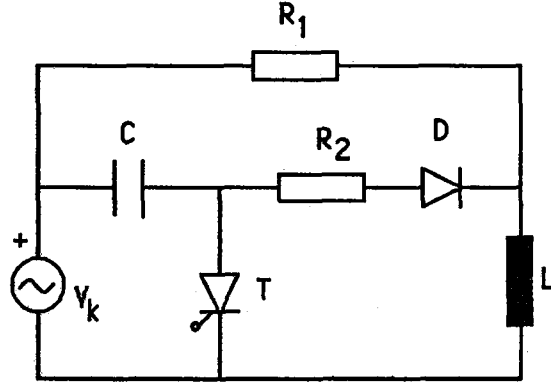
matrisi kullanılır. Burada $w_{i,j}$, i. sistem durumu için j. parça parça doğrusal elemanın eleman durumuna (1. durum, 2. durum gibi) eşittir.

Eleman ve sistem durumlarının daha net olarak izahını yapabilmek için Şekil 2.10 da gösterilen örnek ele alınabilir. Bu devredeki parça parça doğrusal elemanlardan T Tristörü 1. eleman ve D Diyotu da 2. eleman olarak kabul edilirse eleman ve sistem durumları aşağıdaki gibi yazılabilir.

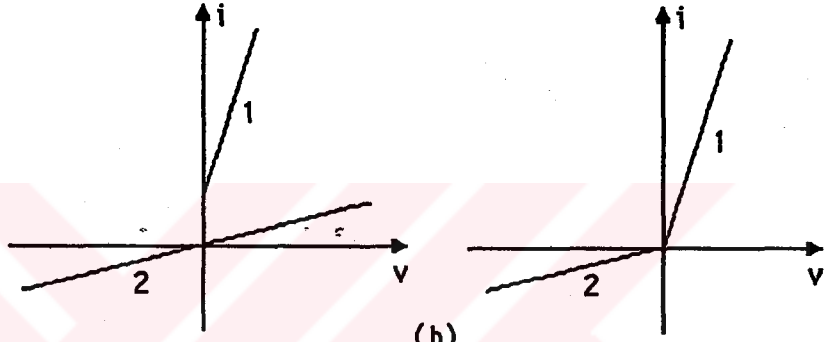
$$\bar{c}_1 = 2, \quad \bar{c}_2 = 2$$

$$S = 2 \cdot 2 = 4$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{bmatrix}$$



(a)



(b)

Şekil 2.10 a) Doğrusal olmayan elemanlar içeren bir devre,
b) Doğrusal olmayan elemanların karakteristikleri

2.4.3. Durum ve çıkış denklemleri

Peryodik olarak zamanla değişen doğrusal devrelerde zamanla değişen elemanlar sadece anahtarlar ise bu devrelerin durum ve çıkış denklemlerini bulmak, anahtarların konumlarının değişmediği her bir durum için zamanla değişmeyen doğrusal devrelerin durum ve çıkış denklemlerini bulmakla aynıdır. Bu metodlar da literatürde geniş biçimde yer almıştır (Desoer 1970 ; Dervişoğlu 1971). Ancak sistemin bir durumdan diğer duruma geçmesi halinde durum değişkenlerinin yeni değerlerini tanımlayan anahtarlama denklemleri için bu

referanslarda verilen teori yeterli değildir. Özellikle anahtarlama denklemleri de söz konusu olduğunda durum ve çıkış denklemlerine ilaveten anahtarlama denklemlerini bulmak için en uygun metod Köksal (Köksal 1975 ; Köksal ve Tokad 1977) tarafından geliştirilmiş ve Nacaroğlu (1989) tarafından programlanmıştır. Genel oluşu ve bilgisayar programına uygunluğu nedeniyle aşağıda bu yöntemin sonuçları sunulmuştur.

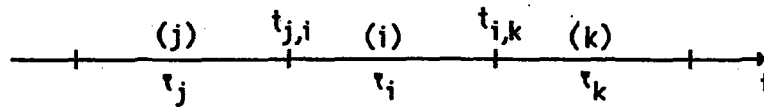
Peryodik olarak çalışan anahtarlar içeren doğrusal bir devrenin bir durumdan diğer duruma geçişi Şekil 2.11' de gösterilen zaman süresi ile anlatılmaya çalışılmıştır. Burada $t_{j,i}$ ve $t_{i,k}$ devrenin durum değiştirdiği (en az bir anahtarın konum değiştirdiği) anları ve τ_j , τ_i ve τ_k da sırayla devrenin i., j. ve k. durumlarda kaldığı süreleri göstermektedir.

τ_j süresince geçerli olan durum ve çıkış denklemleri,

$$\begin{bmatrix} \dot{x}_i(t) \\ y_i(t) \end{bmatrix} = \begin{bmatrix} A_i \\ C_i \end{bmatrix} x_i(t) + \sum_{l=0}^{j_i} \begin{bmatrix} B_{i,l} \\ D_{i,l} \end{bmatrix} \frac{d^l}{dt^l} u(t) \quad (2.24)$$

şeklindedir. Burada j_i , i. durum için giriş fonksiyonlarının (u , uyarma fonksiyonlarının) türevlerinin en yüksek mertebesi ve $y_i(t)$ de, çıkış vektörü y ' nin ($t_{j,i}$, $t_{i,k}$) aralığındaki değeridir. Çıkış vektörünün boyutu bütün durumlar için m' ye eşittir.

Durum vektörü x_i ' nin boyutu ise her durum için farklı olabilir. Devrede mümkün olan en yüksek toplam durum değişkeni sayısı N_x olmak üzere, \bar{N} vektörü,



Şekil 2.11 Sistemin durum değişimleri için notasyon

her durumdaki durum deęişkeni sayısını gösterir ve \bar{N} ' nin elemanlarının deęerleri en fazla N_x ' e eřit olabilir, yani

$$\bar{N} = [\bar{n}_i] \in \mathbb{R}^S, \quad i = 1, 2, \dots, S; \quad [\bar{n}_i] \leq N_x$$

řeklinde dir. Devrenin girebileceęi her durumda, durum deęişkenlerinin sayılarından bařka bir de bunların numaralarının (hangileri olduklarının) bilinmesi gerekmektedir. Bu numaralar N^x matrisinde depolanmıřtır:

$$N^x = [n_{i,j}^x] \in \mathbb{R}^{S \times N_x}, \quad i = 1, 2, \dots, S; \quad j = 1, 2, \dots, N_x$$

$n_{i,j}^x$, $j=1, 2, \dots, n_i$ için i . durumdaki j . durum deęişkeninin numarasıdır ve $j = n_i+1, \dots, N_x$ için 0' dir.

Uyarma fonksiyonlarının sayısı N_f olmak üzere frekans deęerleri \bar{F} vektöründe yer almaktadır ve

$$\bar{F} = [\bar{f}_i] \in \mathbb{R}^{N_f}, \quad i = 1, 2, \dots, N_f$$

dir. Bu kaynakların genlik deęerleri vektörü

$$U = [u_i] \in \mathbb{C}^{N_f}, \quad i = 1, 2, \dots, N_f$$

olup, her kaynak fonksiyonu

$$u_i(t) = U_i e^{j2\pi f_i t} = U_i e^{p_i t}, \quad p_i = j2\pi f_i$$

řeklinde ifade edilir.

Her kaynak fonksiyonu için ayrı bir frekans kabul edilmiřtir. Bir kaynak fonksiyonu birden fazla frekans içeriyorsa, bu kaynak fonksiyonu içerdieęi frekans sayısı kadar kaynak fonksiyonunun toplamı olarak yazılabildięinden bu kaynak doğrusallık özellięi kullanılarak o sayıda kaynaęa eřdeęer kılınabilir. Dolayısıyla yapılan varsayım herhangi bir özelleřtirmeye neden olmaz. Kaynak fonksiyonları arasında sabit kaynakların yer alması veya devredeki doğrusal olmayan elemanların

eşdeğerlerinden dolayı sabit kaynakların devreye eklenmesi durumunda bu sabit kaynakların herbirisi farklı frekansta birer kaynakmış gibi kabul edilmiştir. Bu kaynakların üstel olarak ifade edilebilmeleri için de kosinüs fonksiyonuyla değiştikleri ve frekanslarının 10^{-10} Hz gibi çok küçük değerler olduğu varsayılmıştır.

Yukardaki açıklamaların ışığında (2.24) denklemindeki $u(t)$ ' nin yerine

$$[u_1(t) \ u_2(t) \ \dots \ u_{N_f}(t)] = [U_1 e^{p_1 t} \ \dots \ U_{N_f} e^{p_{N_f} t}]$$

yazılırsa

$$\begin{bmatrix} \dot{x}_i(t) \\ y_i(t) \end{bmatrix} = \begin{bmatrix} A_i \\ C_i \end{bmatrix} x_i(t) + \sum_{k=1}^{N_f} \begin{bmatrix} B_{i,k} \\ D_{i,k} \end{bmatrix} u_k(t) \quad (2.25)$$

denklemini elde edilir. Burada,

$$B_{i,k} = \sum_{l=0}^{j-1} p_k^{(l)} B_{i,k,l}$$

$$D_{i,k} = \sum_{l=0}^{j-1} p_k^{(l)} D_{i,k,l}$$

şeklinde olup $B_{i,k,l}$ ($D_{i,k,l}$) (2.24)' deki $B_{i,l}$ ($D_{i,l}$) matrisinin k . kolonunu gösterir. Birer vektör olan $B_{i,k}$ ve $D_{i,k}$ 'ların herbirisi bir kolon olacak şekilde B_i ve D_i matrisleri oluşturulduğunda (2.25) denklemini,

$$\begin{bmatrix} \dot{x}_i(t) \\ y_i(t) \end{bmatrix} = \begin{bmatrix} A_i \\ C_i \end{bmatrix} x_i(t) + \begin{bmatrix} B_i \\ D_i \end{bmatrix} u(t) \quad (2.26)$$

şekline getirilmiş olur. Dolayısıyla her durum için $2N_f$ tane vektör yerine 2 tane alt matris elde edilmiş olur.

A_i , B_i , C_i ve D_i , i . durumda geçerli olan katsayı matrisleridir. Bu matrislerin boyutları için,

$$A_i = [a_{j,k}] \in R^{\bar{n}_i \cdot \bar{n}_i}$$

$$B_i = [b_{j,k}] \in C^{\bar{n}_i \cdot N_f}$$

$$C_i = [c_{j,k}] \in R^{m \cdot \bar{n}_i}$$

$$D_i = [d_{j,k}] \in C^{m \cdot N_f}$$

yazılabilir. Bilgisayar programında az sayıda tanım kullanılabilmesi için bütün A_i matrisleri tek bir vektör (A vektörü) altında isimlendirilmiştir. A vektörü, $i=1$ ' den başlamak üzere A_i matrislerinin kolon kolon bir vektör şeklinde dizilmesiyle oluşmuştur. A_i matrisinin j,k . elemanı A vektörünün

$$\sum_{l=1}^{i-1} \bar{n}_l \cdot \bar{n}_l + (k-1) \bar{n}_i + j.$$

elemanı olarak alınmıştır.

Benzer şekilde B_i , C_i , D_i ve N^x matrisleri de B , C , D ve N vektörleri şeklinde gösterilmiştir. B_i , C_i , D_i ve N^x ' in j,k . elemanları,

$$B' \text{ nin } \sum_{l=1}^{i-1} \bar{n}_l \cdot N_f + (k-1) \bar{n}_i + j.$$

$$C' \text{ nin } \sum_{l=1}^{i-1} m \cdot \bar{n}_l + (k-1) m + j.$$

$$D' \text{ nin } (i-1) m \cdot N_f + (k-1) m + j.$$

$$N' \text{ nin } (k-1) S + j.$$

elemanları olarak alınmıştır.

2.4.4. Anahtarlama denklemleri

Sistemin bir durumdan diğereine geçtiği anlarda durum değişkenlerinde oluşan süreksizlikler anahtarlama denklemleri yardımıyla sorun olmaktan çıkarılır. Yük sakınımı denklemleri kullanılarak bu sorunun ortadan kaldırılması için bazı yöntemler geliştirilmiştir (Fahmy vd. 1987 ; Tanaka ve Mori 1982 ; Wellekens 1982). Ancak bu yöntemler etkili değildir ve büyük çaplı uygulamalara imkan vermezler. Anahtarlama denklemleri için aynı zamanda durum ve çıkış denklemlerinin elde edilmesini de içeren ve bilgisayar programına uygun olan en genel ve sistematik yöntemin Köksal (Köksal 1975 ; Köksal ve Tokad 1977) tarafından verildiği belirtilmiştir. Şimdi bu yöntemin anahtarlama denklemleri ile ilgili kısmına kısaca değinilecektir.

Şekil 2.11' deki herhangi bir $t_{i,k}$ anına ilişkin anahtarlama denklemi aşağıdaki üç şart kabul edilerek bulunur:

i) $u(t)$ ve $u_{i+1}(t)$, $t_{i,k}$ komşuluğunda parça parça süreklidirler.

ii) $u(t)$ ve $u_{i+1}(t)$ nin, $t_{i,k}$ ' nin komşuluğunda $j_{i+1}-1$ defa diferansiyeli alınabilir ($t_{i,k}$ hariç)

iii) $t = t_{i,k}$ anında kapanan bir anahtar sonsuz akım çekemez ve $t=t_{i,k}$ anında açılan bir anahtar uçlarında sonsuz gerilim üretmez.

İlk iki şart girişle ilgilidir. Sonuncusu ise her ne kadar bütün devrenin davranışını ilgilendirirse de giriş uyarısı altında özellikle anahtarları ilgilendirir (Köksal ve Tokad 1977).

$t_{i,k}$ ' da $j_i+j_{i+1}-1$ kere diferansiyeli alınabilir özellikteki bir giriş için anahtarlama denklemi, i sistemin bulunduğu durum ve k sistemin geçtiği yeni durum olmak üzere,

$$X_k(t_{i,k}^+) = F_{i,k} X_i(t_{i,k}^-) + \sum_{v=0}^{j_i+j_{i+1}-1} \delta^{(j_{i+1})} G_{i,k} u^{(v)}(t_{i,k}) \quad (2.27)$$

olarak yazılabilir. Burada,

$$\delta(j_{i+1}) = \begin{cases} 0 & j_{i+1} = 0 \text{ ise} \\ 1 & j_{i+1} \geq 1 \text{ ise} \end{cases}$$

olarak tanımlanır. Daha önce (2.25), (2.26) denklemlerinin elde edilmesinde olduğu gibi giriş fonksiyonlarının üstel ve farklı frekanslarda olmaları durumunda, yani $u_i = U_i e^{p_i t}$ için (2.27) denklemi,

$$x_k(t_{i+1}, k) = F_{i,k} x_i(t_i, k) + \sum_{l=0}^{N_f} G_{i,k,l} U_l e^{p_l t_i, k} \quad (2.28)$$

olarak yazılır. Burada $G_{i,k,l}$ bir kolon vektörü olup

$$G_{i,k,l} = \sum_{v=0}^{j_i + j_{i+1} \delta(j_{i+1})} p_l^v G_{i,k,l}$$

ifadesiyle belirlenir. Bu ifadede $\sqrt{G_{i,k,l}}$, $\sqrt{G_{i,k}}$ matrisinin l . kolonunu gösterir. $G_{i,k,l}$ vektörleri bir $G_{i,k}$ matrisinin kolonları olacak şekilde yazıldığında (2.28) denklemi,

$$x_k(t_{i+1}, k) = F_{i,k} x_i(t_i, k) + G_{i,k} u(t_i, k) \quad (2.29)$$

$$F_{i,k} = [f_{j,l}] \in R^{\bar{n}_k \times \bar{n}_i}$$

$$G_{i,k} = [g_{j,l}] \in R^{\bar{n}_k \times N_f}$$

$$i, k = 1, 2, \dots, S \quad ; \quad i \neq k$$

şeklini alır. Bu denklem, anahtarlama hemen sonraki durum değişkenlerinin değerlerinin, anahtarlama hemen önceki durum değişkenlerinin ve anahtarlama anındaki giriş fonksiyonlarının değerlerine doğrusal olarak bağlı olduğunu gösterir.

S değişik duruma girebilen bir sistemde bir durumdan herhangi bir başka duruma geçişlerin sayısı $S(S-1)$ tanedir. Dolayısıyla bu sistem $S(S-1)$ tane

anahtarlama denklemine sahip olacaktır. Bu, anahtarlama denklemlerindeki katsayı matrislerinin (F, G) toplam boyutları hakkında fikir vermesi açısından önemlidir.

Daha önce de uygulandığı gibi bilgisayar programında çok fazla tanım kullanmamak için $F_{i,k}$ ve $G_{i,k}$ matrislerinin tamamı F ve G vektörlerinde toplanmıştır. $F_{i,k}$ ve $G_{i,k}$ matrislerinin j,l. elemanları,

$$F' \text{ nin } \sum_{m=1}^{i-1} \sum_{p=1}^S \bar{n}_m \cdot \bar{n}_p + \sum_{m=1}^{k-1} \bar{n}_i \cdot \bar{n}_m + (I-1) \bar{n}_{k+j} .$$

$$G' \text{ nin } (I-1) \sum_{m=1}^S \bar{n}_m \cdot N_f + \sum_{m=1}^{k-1} \bar{n}_m \cdot N_f + (I-1) \bar{n}_{k+j} .$$

elemanları olarak alınmışlardır.

2.4.5. Kontrol denklemleri

Parça parça doğrusal elemanların karakteristiklerindeki her bir doğrusal parça, bir eleman durumu ve bu elemanların değişik durumlarının kombinasyonları da sistem durumu olarak tanımlanmıştır. Sistemin hangi sistem durumunda olduğunun belirlenebilmesi için elemanların ayrı ayrı eleman durumlarının belirlenmesi gerekiyordu. Örneğin Şekil 2.6' da verilen güç diyotu için bu elemanın durumu,

$$e_i = \begin{cases} 1 & v \geq V_0 \text{ ve } i \geq I_0 \text{ ise} \\ 2 & -v \geq -V_0 \text{ ve } -i \geq -I_0 \text{ ise} \end{cases}$$

şeklinde belirlenir. Diyotun kontrol denklemi olarak bilinen bu ifadede v ve i diyota ilişkin uç büyüklükleri olup e diyotun durumunu gösterir. Bir elemanın durumunu belirlemek için en az bir eşitsizliğin olduğu varsayımıyla, değişik durumlarda

bulunabilecek tüm doğrusal olmayan elemanlara ilişkin kontrol denklemleri (burada denklem, eşitsizlikleri de kapsar anlamda kullanılmıştır) topluca matris formda,

$$L_{i,j}^c \cdot y(t) \geq R_{i,j}^c \quad (2.30)$$

biçiminde yazılabilir. Bu yazılış bilgisayar programlamasına uygun olup

$$L_{i,j}^c = \begin{bmatrix} c \\ l_{k,l} \end{bmatrix} \in R^{n_{i,j}^d \cdot m}$$

matrisi değerleri durum değişimine neden olan uç büyüklüklerine (y vektörü ile gösterilmiştir) ilişkin katsayıları,

$$R_{i,j}^c = \begin{bmatrix} c \\ r_k \end{bmatrix} \in R^{n_{i,j}^d}$$

vektörü de uç büyüklüklerinin durum değişimine neden olan referans değerlerini gösterir. Burada $n_{i,j}^d$, i . elemanın j . durmu için kontrol denklemlerinin sayısını göstermektedir ve elemanların durumlarına göre denklem sayısını gösteren N^d matrisinin elemanıdır. Daha önce tanımlandığı gibi c_i , i . elemanın durum sayısı ve N_c eleman sayısı olmak üzere N^d için,

$$N^d = \begin{bmatrix} d \\ n_{i,j} \end{bmatrix} \in N^{N_c \cdot \max \bar{c}_k}$$

$$n_{i,j}^d = 0, \quad j = \bar{c}_j + 1, \dots, \max \bar{c}_k$$

yazılabilir. Kontrol denklemlerinin bir eleman durumu için birden fazla olması halinde bunların birbirine VE ile mi yoksa VEYA ile mi bağlı olduğunu gösterebilmek için Z_i matrisi tanımlanmıştır. i . elemana ilişkin bilgileri içeren Z_i matrisinde, j eleman durumunu göstermek üzere,

$$z_{j,k} = \begin{cases} 0 & (k+1). \text{ denkleme yok ise} \\ 1 & k. \text{ denkleme } (k+1). \text{ denkleme VE ile bağılı ise} \\ 2 & k. \text{ denkleme } (k+1). \text{ denkleme VEYA ile bağılı ise} \end{cases}$$

$$Z_i = [z_{j,k}] \in N^{\bar{c}_i \cdot \max n_{j,i}^d}$$

$$z_{j,k} = 0, \quad k = n_j^d + 1, \dots, \max n_{j,i}^d$$

olarak tanımlanabilir.

Yukardaki notasyon ışığında daha önce örnek verilen güç diyotunun i. elemanı olduğu farzedilerek kontrol denklemleri birinci durum için $L^{c_{i,1}} y(t) \geq R^{c_{i,1}}$ 'e karşılık,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} \geq \begin{bmatrix} V_0 \\ I_0 \end{bmatrix}$$

ve ikinci durum için $L^{c_{i,2}} y(t) \geq R^{c_{i,2}}$ 'ye karşılık,

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ i \end{bmatrix} \geq \begin{bmatrix} -V_0 \\ -I_0 \end{bmatrix}$$

şeklinde dir.

Sadece parça parça doğrusal olan elemanlar için buraya kadar yapılan tanımlamalar yeterlidir. Ancak bu elemanlar aynı zamanda dışarıdan da kontrol ediliyorsa o zaman eleman durumlarının belirlenmesinde bunun da dikkate alınması gerekir. Çünkü eleman durumlarını sadece hesaplanan büyüklükler (y) değil, dışarıdan yapılan etki de belirleyecektir. Bu etkilerin zamanlarının bilinmesi ve kontrol denklemlerinin yanı sıra bu zaman değerlerinin de kontrol edilmesi gerekir. Bunun için öncelikle elemanların hangi durumlarının zamanla ilgili hangi durumlarının ilgisiz olduğunu gösteren bir N^z matrisi,

$$N^z = [n_{i,j}^z] \in N^{N_c \cdot \max \bar{c}_k}$$

$$n_{i,j}^z = \begin{cases} 0 & \text{i. elemanın j. durumu zamanla ilgili değilse} \\ n & \text{i. elemanın j. durumu bir periyotta n kez kontrol ediliyorsa} \end{cases}$$

$$n_{i,j}^z = 0, j = \bar{c}_1 + 1, \dots, \max \bar{c}_k$$

ifadeleriyle tanımlanmıştır. Zamanla ilgili olan eleman durumlarına ilişkin zaman değerleri de aşağıda tanımlanan Z^z matrisinde depolanmıştır.

$$Z^z = \begin{bmatrix} z \\ z_i \end{bmatrix} \in R^\gamma$$

$$\gamma = \sum_{j=1}^{N_c} \sum_{k=1}^{\max \bar{c}_i} n_{j,k}^z$$

Burada z_i^z 'ler ($i=1, 2, \dots, \gamma$) dışardan kontrol edilen elemanlara kontrolün yapıldığı zaman değerleridir. Bu değerlerin sıralanışı, N^z matrisi kolon kolon tarandığında karşılaşılabilecek olan sıfırdan farklı elemanların sıralarıyla aynıdır.

Daha önce yapıldığı gibi bilgisayar programında az sayıda tanım kullanılabilmesi için bu bölümde tanımlanan matrisler ($L^{c_{i,j}}, Z_i$) de birer vektör olarak düzenlenmiştir. Aynı zamanda bir elemanla ilgili olan vektörler ($R^{c_{i,j}}$) bir araya toplanarak tüm elemanlarla ilgili vektörler haline getirilmişlerdir.

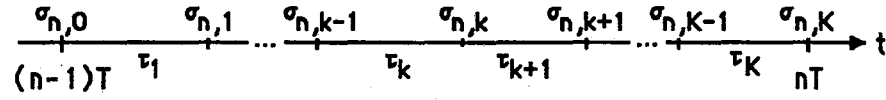
2.5. Parça-Parça Doğrusal veya Dışardan Kontrollü Elemanlar İçeren Devrelerin Sürekli Hal Çözümü

Geliştirilen yöntemin en önemli kısmını oluşturan çözüm iki aşamadan oluşmaktadır. İlk olarak sistemin geçici hal çözümünü de içeren tam çözümden yararlanılarak sistemin girdiği durumlar ve durumlar arası geçiş zamanları

yaklaşık olarak belirlenmektedir. Daha sonra durumlar arası geçiş zamanları tam olarak belirlenerek sürekli hal çözümüne geçilmektedir. Birbirinden farklı olan bu iki çözüm adımlarından birincisi devrenin sürekli hal davranışının belirlenmesi başlığı altında ve ikincisi de sürekli hal çözümü başlığı altında ayrı ayrı incelenecektir.

2.5.1. Devrenin sürekli hal davranışının belirlenmesi

S değişik durumu bulunan bir sistemin, sürekli halde bu S değişik durumdan hangilerine, hangi sırayla ve hangi zamanlarda girdiği başlangıçta belli değildir. Oysaki sürekli hal çözümünün yapılabilmesi için bunların bilinmesi gerekmektedir. Bu durumlar; sistemde yer alan parça-parça doğrusal veya dışarıdan kontrollü elemanlar tarafından belirlenmektedir. Bundan dolayı önce bu elemanların durumlarının belirlenmesi daha sonra bu eleman durumlarına göre sistem durumunun belirlenmesi gerekmektedir. Bu elemanlara dışardan kontrolün ne zaman yapılacağı zaten bilinmektedir. Öylese sorun, bunların sistem tarafından belirlenen uç büyüklüklerine göre hangi bölgede bulduklarının belirlenmesidir. Bunun için de sistemi tanımlayan durum ve çıkış denklemlerinin çözülmesi gerekmektedir. Aşağıda sırayla incelenecek olan bu adımların sonucunda sistemin durumlar açısından sürekli hal davranışı belirlenmiş olacaktır. Girebileceği S değişik durumu bulunan sistem, sürekli halde bunlardan bazılarında hiç girmeyebilir, bazılarında bir kez ve bazılarında da bir kezden fazla girebilir. Sürekli halde girilen durumlar aralık (interval) olarak adlandırılır. Toplam aralık sayısı K olmak üzere K, S' den büyük veya küçük olabileceği gibi S' e eşit de olabilir. Sürekli haldeki n. periyot için notasyon Şekil 2.12' de verilmiştir.



Şekil 2.12 n. anahtarlama periyodu için notasyon

Sürekli haldeki k. aralıkta sistem s. durumda ise s. duruma ait bütün bilgiler (A_s, B_s, C_s, \dots) k. aralığa aktarılır (A_k, B_k, C_k, \dots). Aynı şekilde (k+1). aralıkta sistem (s+1). duruma geçiyorsa s. durumdan (s+1). duruma geçiş için kullanılan anahtarlama matrisleri ($F_{s,s+1}, G_{s,s+1}$) de k. aralıktan (k+1). aralığa geçiş için kullanılacak olan anahtarlama matrisleri (F_k, G_k) olarak aktarılır. Bütün bunlar gösteriyor ki sistemin başlangıçtaki durumu için yapılan tanımlamalarla sürekli hal davranışı için yapılan tanımlamalar birbirinden farklıdır.

Bölüm 2.4.3' de tanımlanan (2.26)' daki durum denkleminin $t_{j,i} \leq t < t_{j,k}$ aralığında $u_k(t) = U_k e^{p_k t}$ için çözümü,

$$x_i(t) = e^{A_i(t-t_{j,i})} x_i(t_{j,i}) + \sum_{k=1}^{N_f} \int_{t_{j,i}}^t e^{A_i(t-\tau)} B_{i,k} u_k(\tau) d\tau \quad (2.31)$$

$$x_i(t) = e^{A_i(t-t_{j,i})} x_i(t_{j,i}) + \sum_{k=1}^{N_f} \int_{t_{j,i}}^t e^{A_i(t-\tau)} B_{i,k} U_k e^{p_k \tau} d\tau$$

$$x_i(t) = e^{A_i(t-t_{j,i})} x_i(t_{j,i}) + \sum_{k=1}^{N_f} \int_{t_{j,i}}^t e^{A_i t} e^{-(A_i - p_k) \tau} B_{i,k} U_k d\tau$$

$$x_i(t) = e^{A_i(t-t_{j,i})} x_i(t_{j,i}) + \sum_{k=1}^{N_f} e^{A_i t} (A_i - p_k)^{-1} \left(e^{-(A_i - p_k) t} - e^{-(A_i - p_k) t_{j,i}} \right) B_{i,k} U_k$$

$$x_i(t) = e^{A_i(t-t_{j,i})} x_i(t_{j,i}) + \sum_{k=1}^{N_f} \left[(A_i - p_k)^{-1} e^{p_k t} - (A_i - p_k)^{-1} e^{p_k t_{j,i}} e^{A_i(t-t_{j,i})} \right] B_{i,k} U_k$$

olarak elde edilir. Bu çözüm yukarda belirtildiği gibi $t_{j,i} \leq t < t_{i,k}$ zaman aralığında geçerlidir. Başlangıçta sistemin hangi durumlara, hangi sırayla ve hangi zamanlarda girdiği belli değildir. Dolayısıyla $t_{j,i}$, $t_{i,k}$ zaman değerleri ve τ_i zaman süreleri bilinmemektedir. Bundan dolayı çözüme başlanırken sistemin birinci durumda olduğu kabul edilerek ve $t_{j,i} = 0$ alınarak başlanır. Daha sonra zaman Δt adımıyla artırılarak çözüme devam edilir. Her çözüm adımından sonra,

$$y_i(t) = C_i x_i(t) + \sum_{k=1}^{N_i} D_{i,k} U_k e^{P_k t} \quad (2.32)$$

çıkış denklemi kullanılarak çıkış büyüklüklerinin değerleri elde edilir. Bu değerlerden yararlanılarak sistemin yeni durumu belirlenir. Eğer durumda değişme yoksa Δt adımıyla aynı işlemlere devam edilir. Durum değişimi varsa o zaman eski ve yeni duruma uygun olan anahtarlama denklemi kullanılarak yeni durumdaki durum değişkenlerinin başlangıç değerleri belirlenir.

Tam çözümde kullanılan zaman adımı Δt sayısal çözümleme yöntemlerinde olduğu gibi çok küçük bir değer değildir. Sistemin durumlarda kalma sürelerinden büyük olmamak şartıyla başka bir sınırlama yoktur. Bu nedenle çözüm işlemleri çok kısa sürede sonuçlandırılabilir.

$x_i(t)$ durum vektörünün boyutu, anahtarlama ile devre kofigürasyonundaki değişimler nedeniyle her i için farklı olabilir. Bundan dolayı her durum değişiminde durum, çıkış ve anahtarlama denklemlerinin katsayı matrislerinin boyutları da değişir. Programlama açısından karmaşık olmasına rağmen bilgisayarda hesap zamanı ve bellek kullanımı açısından avantaj sağlayabilmek için her durumda o duruma ait durum vektörünün boyutuna uygun boyutlar kullanılmıştır.

Çözüm işlemlerine başlanırken sistemin birinci durumda olduğu kabul edilmiş ve bu duruma ilişkin durum ve çıkış denklemleri kullanılarak çözüme başlanmıştır. Çözümün her adımından sonra parça-parça doğrusal veya dışardan

kontrollü elemanların durumları kontrol denklemleri kullanılarak ayrı ayrı belirlenir. Daha sonra elemanların mevcut durumlarına karşılık gelen sistem durumu belirlenir. Belirlenen yeni sistem durumu eskisiyle aynı ise bu işlemlere devam edilir. Eğer yeni sistem durumu öncekinden farklı ise o zaman eski ve yeni duruma uygun olan anahtarlama denklemi kullanılarak yeni durumdaki durum değişkenlerinin başlangıç değerleri belirlenir ve bu yeni duruma ilişkin durum ve çıkış denklemleri seçilerek çözüme devam edilir. Her durum değişimi olduğunda girilen yeni durumların numaraları bir vektörde toplanır. Her periyot sonunda o periyottaki durumların bir önceki periyottaki durumlarla aynı olup olmadığı kontrol edilir. Sistem sürekli hale geldiği zaman her periyotta aynı durumlar yer alacaktır. Bu sağlandığı zaman çözüm işlemine son verilir ve en son periyot için depolanan durumlar, sistemin sürekli haldeki girdiği durumlar olarak alınır.

Eleman durumları belirlenirken elemanların özelliklerine göre yazılan kontrol denklemlerinden ve zaman vektörlerinden yararlanır. Parça-parça doğrusal bir elemanın hangi durumda olduğunu o elemanın uç geriliminin ve akımının değerleri belirler. Eğer bu eleman dışardan kontrollü bir elemansa bu değerlerin yanı sıra dışardan yapılan etkinin zamanı da eleman durumunu belirleyen bir etken olacaktır. Bu nedenle, eleman durumu belirlenirken kontrol denklemlerinin yanı sıra N^Z matrisinde gösterilen zamanların Z^Z vektöründeki zaman değerleri de kontrol edilir. Herhangi bir durumu zamana bağımlı olan elemanlarda bazen, kontrol denklemlerine göre o duruma giriyor gözükse bile o durumla ilgili zaman değerine ulaşılmadıkça geçiş gerçekleşmeyecektir (Tristör veya Transistörün iletme geçmesi gibi). Bazen de kontrol denklemlerine göre durum değişimi yok gibi gözükse bile o durumla ilgili zaman değerine ulaşıldığı için geçiş gerçekleşir (Transistörün kesime gitmesi gibi). Bunların birbirinden ayrımı, zaman değerinin hangi duruma ilişkin olduğuna bağlıdır.

İki durumdan fazla durumu bulunan bazı elemanlarda bir durumdan başka bir duruma doğrudan geçiş mümkün olmayabilir. Yani bazen elemanların özelliklerine göre öyle olur ki birden fazla duruma ait kontrol denklemlerindeki şartlar sağlanabilir. Gerçekte ise bu durumlardan sadece birine girilebilir. İşte bu gibi durumları belirleyebilmek için Geçiş Matrisi tanımlanmıştır. j ve i eleman durumlarını göstermek üzere i . elemana ilişkin \bar{G}_i geçiş matrisi,

$$\bar{G}_i = [g_{j,i}] \in N^{\bar{n}_i \cdot \bar{n}_i}$$

$$g_{j,i} = \begin{cases} 1 & j. \text{ durumdan } i. \text{ duruma geçebiliyorsa} \\ 0 & j. \text{ durumdan } i. \text{ duruma geçemiyorsa} \end{cases}$$

şeklinde tanımlanır.

Yukarda anlatılanların ışığında bütün elemanların durumları ayrı ayrı belirlendikten sonra bu eleman durumlarına karşılık gelen sistem durumları belirlenir. Bunun için eleman durumlarına göre sistem durumunu gösteren W Eleman Durumu-Sistem Durumu matrisinden yararlanılır.

Sistemin sürekli haldeki durumları belirlenirken bir periyot boyunca girilen durumlar bir vektörde depolanmaktaydı. Bu işlem devam ederken, aynı zamanda durum değişimlerinin olduğu zaman değerleri de başka bir vektörde depolanır. Her periyot için zaman değerleri sıfırdan başlamak üzere alınır. Yani n . periyotta işlem yapılırken bir t anında durum değişimi olmuşsa, T ana periyodu göstermek üzere depolanacak $t_{n,i}$ değeri,

$$t_{n,i} = t - (n-1) T$$

olarak hesaplanır. Böylece zaman değerlerinin periyot bazında kalması sağlanmış olur. Her periyotta yeni değerler depolandığı için durumların belirlenmesi işlemine son verildiği andaki en son değerler yaklaşık geçiş zamanları olarak alınır. Yaklaşık ifadesinin kullanılması, çözüm için kullanılan Δt zaman adımının yeterince küçük

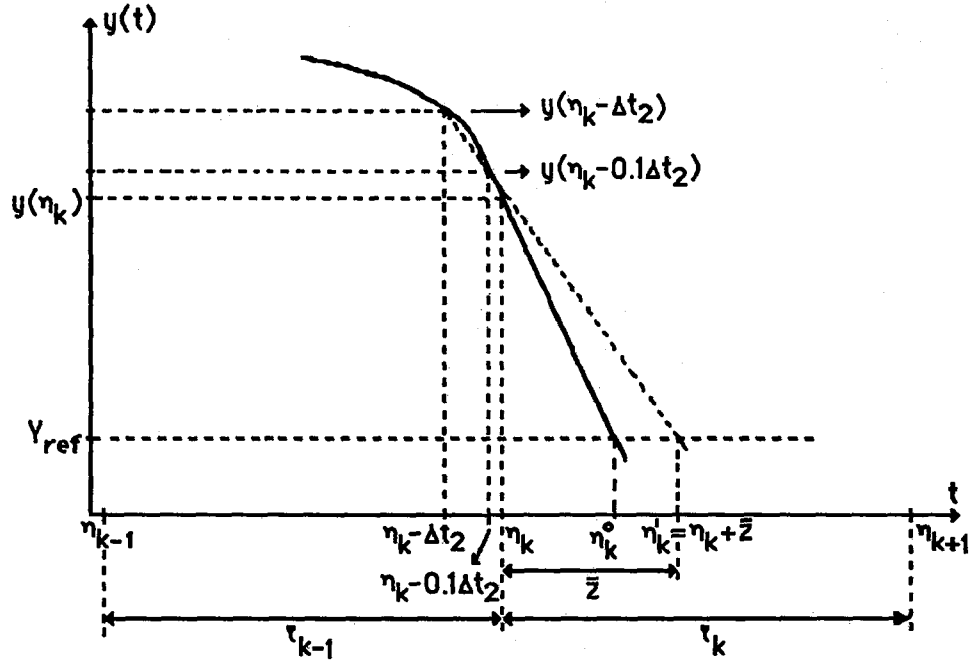
olmamasındandır. Bu zamanların gerçek değerleri daha sonra başka bir metodla kısa sürede bulunabileceği için bu aşamada küçük Δt adımı kullanılarak bilgisayardaki hesap zamanının artırılması gereksiz görülmüştür.

2.5.2. Sürekli hal çözümü

Parça-parça lineerlik yaklaşımı kullanılan sistemlerin sürekli hal çözümlerinin elde edilebilmesi için lineer aralıkların sayısı K 'nin ve geçiş zamanları $\eta_1, \eta_2, \dots, \eta_k$ 'ların bilinmesi gerekmektedir. Aralık sayısı ve aralıkların sırasıyla geçiş zamanlarının yaklaşık değerleri yukarıda elde edilmiştir. Zamanların gerçek değerlerinin belirlenebilmesi için Şekil 2.13' de gösterilen yöntemden yararlanılacaktır.

Sistemin k . aralıktan $(k+1)$. aralığa geçiş zamanını Şekil 2.13' de görüldüğü gibi $y(t)$ 'nin değişimi belirliyor olsun. Geçiş zamanının gerçek değeri, $y(t)$ 'nin y_{ref} referans değeri ile kesiştiği noktaya karşılık gelen η_k^0 'dır. Halbuki yaklaşık olarak belirlenen geçiş zamanı η_k bu noktaya uzaktır. Öyleyse η_k 'nin η_k^0 'a doğru kaydırılması gerekmektedir. Bunun için (2.33) ve (2.34)'den yararlanılacaktır. Bu eşitliklerden görüleceği gibi geçiş anına kadar sadece $\eta_k - 0,1\Delta t_2$ ve $\eta_k - \Delta t_2$ 'deki $y(t)$ değerleri hesaplanmaktadır.

Daha sonra Newton-Raphson metodu kullanılarak geçiş zamanları daha hassas olarak belirlenmektedir (Türkbeyler 1987). Şekilden de görüleceği gibi yeni geçiş zamanı,



Şekil 2.13 Geçiş zamanlarının belirlenmesi

$$\eta_k^0 \approx \eta_k^1 = \eta_k + \bar{z} \quad (2.33)$$

olarak alınır. Burada,

$$\bar{z} = \Delta t_2 \frac{y(\eta_k - 0,1 \Delta t_2) - 0,9 y_{ref} - 0,1 y(\eta_k - \Delta t_2)}{y(\eta_k - \Delta t_2) - y(\eta_k - 0,1 \Delta t_2)} \quad (2.34)$$

şeklinde ve Δt_2 , $y(t)$ ' nin değerine ve eğimine uygun olabilecek küçük bir zaman parçasıdır. Ana periyot T ' nin $1/10000$ ' i olarak alınabilir (Türkbeyler 1987).

Yukarda k. geçiş zamanı için anlatılan düzeltme işlemi, kesin olarak bilinmeyen tüm η_k ' lar için tekrarlanır. Bulunan yeni geçiş zamanları (η_k^1 ' lar) göz önüne alınarak sistemin sürekli hal analizi yeniden yapılır.

Bu işlemler itiratif olarak tekrarlanarak her geçiş zamanındaki ilgili $y(t)$ ile y_{ref} arasındaki farkın daha önce belirlenen Δy değerinden daha küçük olması sağlanır. Bu fark, kabul edilebilir bir hataya karşılık geldiği için bu aşamadan sonra elde edilen η_k^1 değerleri geçiş zamanları olarak kabul edilir.

Her iterasyonda yeni geçiş zamanları belirlendikten sonra bu zamanlara göre aralıkların birbirine taşma yapıp yapmadığının kontrol edilmesi gerekmektedir. Yani,

$$\eta_0^1 < \eta_1^1 < \dots < \eta_K^1 < T \quad (2.35)$$

şartları sağlanmalıdır. Eğer taşma oluyorsa (2.33) eşitliği,

$$\eta_k^1 = \eta_k + \frac{\bar{z}}{s} \quad (2.36)$$

şeklinde düzeltilmelidir. Burada \bar{s} bir emniyet faktörüdür ve $\bar{s} > 1$ dir. Seçilen bir \bar{s} için hala aralıklar üst üste geliyorsa program tarafından otomatik olarak \bar{s} iki ile çarpılarak $\eta_k^1 - \eta_k$ azaltılır. Yani,

$$\eta_k^1 = \eta_k + \frac{\bar{z}}{s_n}, \quad s_n = 2^n \cdot \bar{s}, \quad n = 1, 2, \dots \quad (2.37)$$

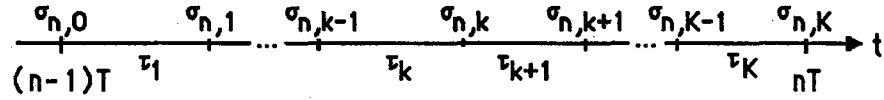
ile son değer bulunmuş olur. Her iterasyondan sonra emniyet faktörü s_n başlangıç değeri olan $s_n = \bar{s}$ ' e indirgenir (Türkbeyler 1987).

Hem bilinmeyen zamanların belirlenmesi sırasındaki çözüm işlemlerinde hem de bir periyot boyunca toplam sonuçların elde edilmesi sırasındaki çözüm işlemlerinde Köksal tarafından türetilen sürekli hal analizi sonuçları kullanılmıştır. Bununla ilgili geniş bilgi (Köksal 1975 ; Köksal 1984a)'da bulunabilir. Şimdi kullanılan bu sonuçlar kısaca verilecektir.

Çözümü yapılacak olan durum, çıkış ve anahtarlama denklemleri, verilmeden önce bu denklemlerin daha iyi yorumlanabilmesi için n. anahtarlama periyoduna ilişkin notasyon Şekil 2.14' de tekrar verilmiştir.

n. periyot için geçerli olan durum, çıkış ve anahtarlama denklemleri,

$$\dot{x}_{n,k}(t) = A_k x_{n,k}(t) + B_k u(t) \quad (2.38)$$



Şekil 2.14 n. anahtarlama periyodu için notasyon

$$y_{n,k}(t) = C_k x_{n,k}(t) + D_k u(t) \quad (2.39)$$

$$x_{n,k+1}(t_{n,k}^+) = F_k x_{n,k}(t_{n,k}^-) + G_k u(t_{n,k}) \quad (2.40)$$

$t_{n,k-1} < t < t_{n,k}$ ve $k = 1, 2, \dots, K$ için

şeklindedir. $u(t) = U e^{pt}$ için (2.38) denkleminin $t_{n,k-1} < t < t_{n,k}$ aralığındaki çözümü,

$$x_{n,k}(t) = e^{A_k(t-t_{n,k-1})} \left[x_{n,k}(t_{n,k-1}^+) + (A_k - pI)^{-1} B_k U e^{p(t_{n,k-1}-t)} \right] - (A_k - pI)^{-1} B_k U e^{pt} \quad (2.41)$$

şeklinde elde edilir. Burada $t = t_{n,k}$ alınarak,

$$x_{n,k}(t_{n,k}^-) = e^{A_k \tau_k} \left[x_{n,k}(t_{n,k-1}^+) + (A_k - pI)^{-1} (1 - e^{(pI - A_k) \tau_k}) B_k U e^{p t_{n,k-1}} \right] \quad (2.42)$$

şeklinde yazılır. Denklem 2.42 ve denklem 2.40 K defa ardışıl olarak kullanıldığında,

$$x_{n+1,k} \left[((n+1)T + \eta_k^+) \right] = M_k x_{n,k} (nT + \eta_k^+) + H_k U e^{p(nT + \eta_k)} \quad (2.43)$$

elde edilir. Burada $k=1, 2, \dots, K$ ve η_k, M_k, H_k ,

$$\eta_k = \begin{cases} 0 & k=1 \text{ için} \\ \tau_1 + \tau_2 + \dots + \tau_{k-1} & k=2, 3, \dots, K \text{ için} \end{cases}$$

$$\beta_{k,j} = \begin{cases} \eta_{j+T} & j = 1, 2, \dots, k-1 \text{ için} \\ \eta_j & j = k, k+1, \dots, K \text{ için} \end{cases}$$

$$\varnothing_k = F_k e^{A_k \tau_k}$$

$$N_{k,j} = \begin{cases} \varnothing_K \varnothing_{K-1} \dots \varnothing_{j+1} & k=1 ; j=1, 2, \dots, K-1 \text{ için} \\ \varnothing_{k-1} \varnothing_{k-2} \dots \varnothing_1 & k=2, 3, \dots, K ; j=K \text{ için} \\ \varnothing_{k-1} \varnothing_{k-2} \dots \varnothing_{j+1} & k=3, 4, \dots, K ; j=1, 2, \dots, k-2 \text{ için} \\ N_{k,K} N_{1,j} & k=2, 3, \dots, K-1 ; j=k, k+1, \dots, K-1 \text{ için} \\ I & k=1 ; j=K \text{ ve } k=2, 3, \dots, K ; j=k-1 \text{ için} \end{cases}$$

$$M_k = N_{k,k} \varnothing_k$$

$$H_k = e^{-p \eta_k} \sum_{j=1}^k N_{k,j} \left[F_j ((A_j - pI)^{-1} (e^{A_j \tau_j} - I e^{p \tau_j}) B_j e^{p(B_{k,j})} + G_j e^{p(B_{k,j} + \tau_k)} \right]$$

$$P_k = \left[J_k + (A_k - pI)^{-1} B_k \right] e^{p \eta_k}$$

$$J_k = \left(e^{pT} I - M_k \right)^{-1} H_k$$

olarak tanımlanırlar. Bu ifadelerden yararlanılarak sürekli hal zaman domeni çözümü,

$$y(t) = H(p, t) U \cdot e^{pt} \quad (2.44)$$

olarak elde edilir. Burada,

$$H(p, t) = C_k \left[e^{A_k(t-\eta_k)} p_k e^{-pt} - (A_k - pI) B_k \right] + D_k \quad (2.45)$$

$$\eta_k < t < \eta_{k+1} \text{ için}$$

şeklindedir (Köksal 1975 ; Köksal 1984a).

Giriş fonksiyonlarının birden fazla olması durumunda daha önce tam çözüm yapılırken uygulandığı gibi yukarıdaki sonuçlara da süperpozisyon uygulanır.

Yukardaki denklemleri kullanarak, periyodik olarak çalışan anahtarlar içeren lineer devrelerin çözümünü elde eden bir bilgisayar programı Köksal tarafından hazırlanmıştır (Köksal 1984a). MAINLN olarak adlandırılan ve periyodik olarak zamanla değişen sistemlerin genel amaçlı analizini yapan bu program, Türkbeyler (Küpeli) tarafından güç elektroniği devrelerinin analizini yapmak için düzenlenmiştir. APEN olarak adlandırılan bu program, MAINLN' in özel bir uygulaması olarak kabul edilebilir.



3. BİLGİSAYAR PROGRAMI

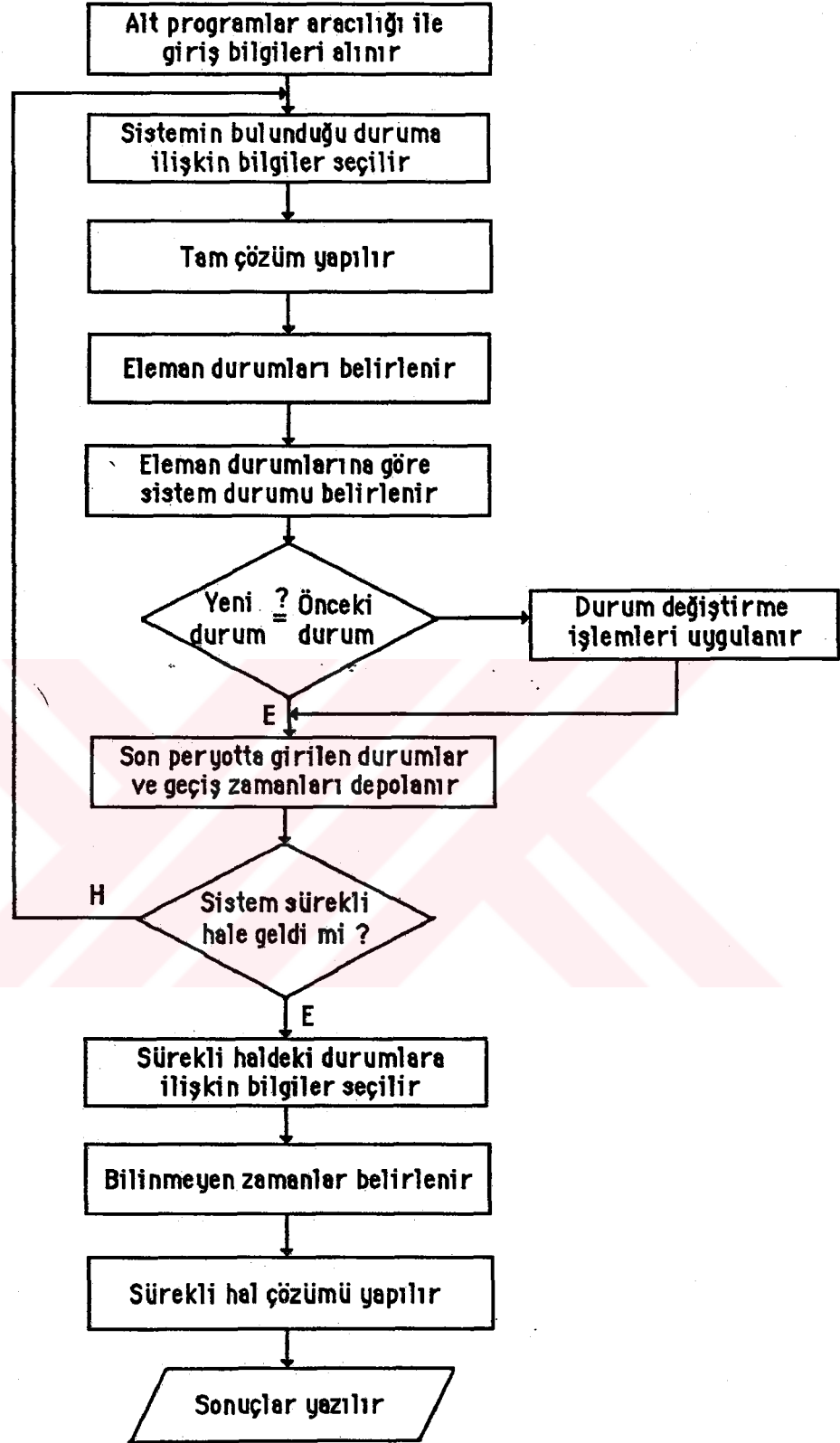
3.1. Giriş

Yukardaki bölümlerde anlatılan işlemleri gerçekleştirmek üzere bir bilgisayar programı hazırlanmış ve ek olarak tezin sonunda verilmiştir. Periyodik olarak çalışan güç elektroniği devrelerinin, veya daha genel olarak parça parça doğrusallık özelliği gösteren doğrusal olmayan devrelerin, analizini gerçekleştirecek olan bu program (PGEDAN), temel olarak iki ana altprogramdan oluşmaktadır. Bunlardan birincisi tam çözümü gerçekleştiren TAMCOZ altprogramı ve ikincisi de sürekli hal çözümünü gerçekleştiren APEN1 altprogramıdır.

TAMCOZ, sürekli haldeki aralıkları (interval) bulmak amacıyla durum denklemlerini ve kontrol denklemlerini de göz önüne alarak hassas olmayan geçici hal ve sürekli hal çözümlerini bulmaktadır. APEN1 altprogramı ise daha önce bahsolunan APEN programının genelleştirilmesi ile elde edilmiştir ve doğrudan sürekli hal çözümünü bulmaktadır. Bu altprogramlarla ilgili detaylı bilgi sonraki bölümlerde verilecektir.

3.2. Akış Şeması

Hazırlanan programla ilgili açıklamalardan önce programın mantığının anlaşılabilmesi için fazla detaylı olmayan bir akış şeması Şekil. 3.1' de verilmiştir. Şekilde görülen bazı bloklar çok uzun program parçalarına karşılık



Şekil 3.1 PEGEDAN programının akış şeması

gelmektedir. Bazı önemli ana altprogramlar ayrı başlıklar altında incelendiği için, burada şekli karmaşıkleştirmek gereksiz görülmüştür.

Akış şemasında görülen işlemlerin bir kısmı doğrudan PGEDAN içerisinde gerçekleştirilirken bir kısmı da çağrılan altprogramlar aracılığı ile gerçekleştirilmektedir. Bu nedenle önce PGEDAN ve daha sonra bazı altprogramlarla kullanıcı programları sırayla incelenecektir.

3.3. Ana Program

Ana program başlığı altında açıklanacak olan PGEDAN programı aslında bir ana altprogram olarak hazırlanmıştır. Bu program tarafından kullanılan dataları üretebilecek başka bir programının bu programı bir altprogram gibi çağırarak kullanılabilmesi için bu yol tercih edilmiştir. PGEDAN, ilk olarak kullanıcı tarafından yazılan alt programları çağırarak giriş bilgilerini alır. Sonra sistemin bulunduğu duruma (başlangıçta 1. durum kabul ediliyor) ilişkin bilgiler seçilir ve tam çözüm için TAMCOZ alt programı çağırılır. Daha sonra elde edilen çözüm sonuçlarına göre eleman durumları ve bu durumlara göre de sistem durumu belirlenir. Sistem durumunda değişme olduğunda girilen yeni durum ve bu andaki zaman değeri ayrı vektörlerde depolanır. Her periyodun sonunda o periyottaki durumların bir önceki periyottaki durumlarla aynı olup olmadığı kontrol edilir. Giriş bilgileri arasında yer alan ve değeri girilmediği zaman otomatik olarak 4 seçilen IVS'nin değeri kadar üst üste periyotlardaki durumlar aynı kalırsa veya bir önceki periyotta elde edilen yaklaşık geçiş zamanları ile son periyotta elde edilen yaklaşık geçiş zamanları arasındaki farklar ana periyodun 1/100' ünden daha küçük olursa sistemin sürekli hale eriştiği kabul edilir. Son olarak, elde edilen bu

durumlara ilişkin bilgiler giriş bilgileri arasından seçilerek sürekli hal çözümü için APEN1 alt programı çağrılır. Kullanıcı alt programları, TAMCOZ ve APEN1' e ek olarak bir de ILKD alt programı yer almaktadır. Her durum değişiminden sonra ILKD çağrılarak anahtarlama denklemleri aracılığı ile girilen yeni durumdaki durum vektörünün başlangıç değerleri hesaplanır.

PGEDAN ve diğer alt programlar arasındaki ilişkiyi daha açık şekilde ortaya koyabilmek için Şekil 3.2' deki şema hazırlanmıştır.

Programda kullanılan değişkenler ilgili alt programlarda açıklanacağı için burada altprogramlarda kullanılmayıp sadece ana programda kullanılan değişkenler açıklanacaktır.

CN: Doğrusal olmayan eleman sayısı

CSN: Doğrusal olmayan elemanların durum sayılarını gösteren ve CN boyutunda olan vektör

NCE: Doğrusal olmayan elemanların herbir durumu için kaç tane kontrol denklemi oluşunu gösteren matris

NZV: Doğrusal olmayan elemanların her bir durumlarının zamana bağlı olup olmadığını gösteren matris

ZV: Zamana bağlı olan eleman durumlarına ilişkin zaman değerlerinin girildiği vektör

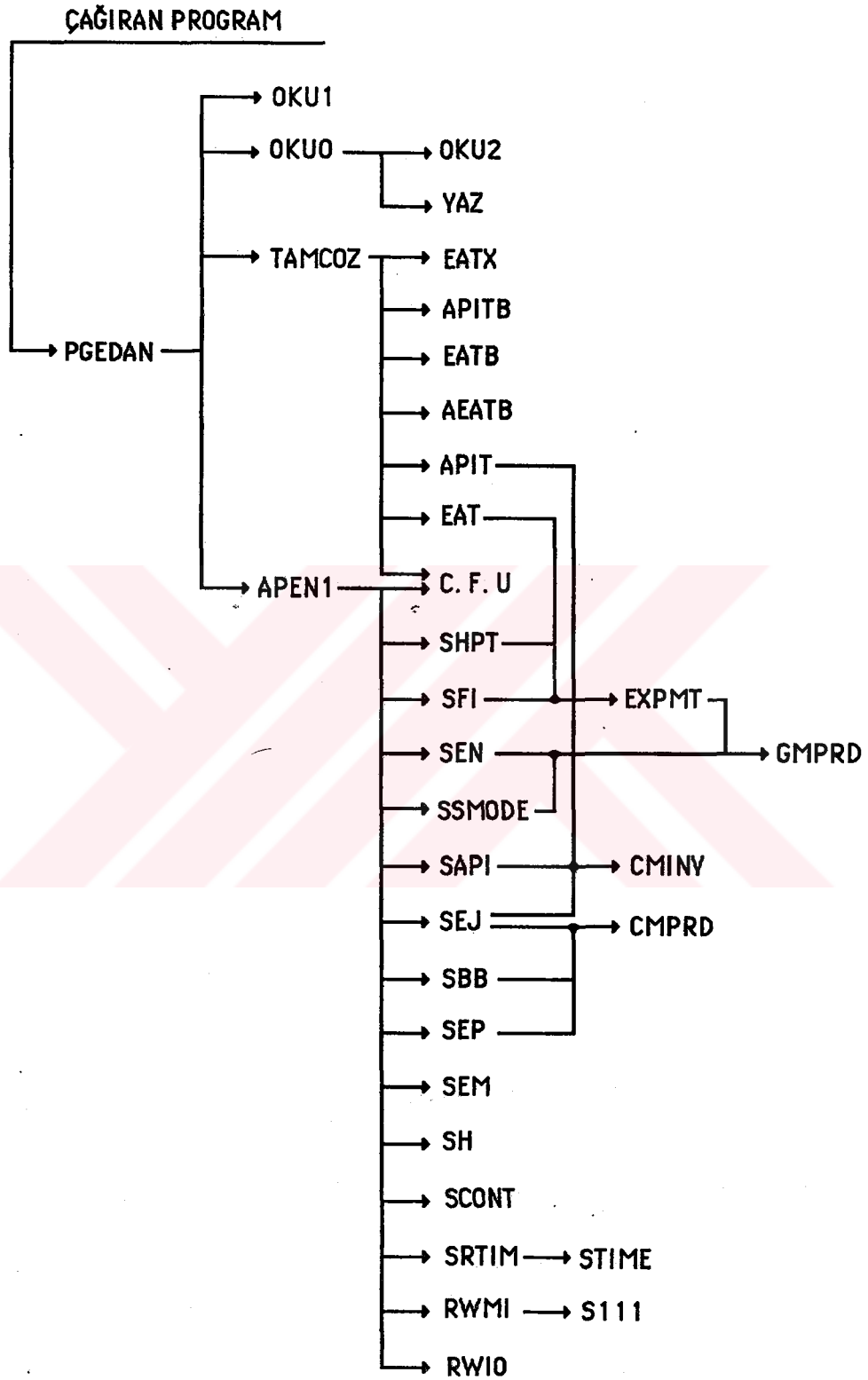
IVS: Sistemin sürekli hale geldiğine karar verebilmek için aynı durumların üst üste kaç periyot tekrar edilmesi gerektiğini gösteren değişken

NDT: Genel çözümde kullanılan zaman adımı DT' nin ana periyodu kaç bölerek elde edileceğini gösteren değişken

NSX: Eleman durumları için durum değişkeni numaralarını gösteren matris

TTO: Sistemin ana periyodunu gösteren değişken

CS: Bütün doğrusal olmayan elemanların buldukları durumları gösteren CN boyutlu vektör



Şekil 3.2 PGEDAN tarafından kullanılan altprogramlar

SSCS: Lineer olmayan elemanların durumlarına göre sistem durumunu gösteren matris

DKM: Kontrol denklemlerinin katsayılar matrisi

DRM: Kontrol denklemlerinin referans değerlerinin girildiği vektör

DKM ve DRM'yi değerlendirirken kontrol denklemlerinin $DKM \cdot y \geq DRM$ formunda olduğu gözönünde bulundurulmalıdır.

3.4. Kullanıcı Programları

Daha önce de belirtildiği gibi kendisine data üreten başka programlar tarafından da kullanılabilmesi için PGEDAN bir ana altprogram olarak hazırlanmıştı. Bu nedenle programı kullanmak isteyen kullanıcıların aşağıdaki çağırın programı yazmaları gerekmektedir.

CALL PGEDAN

STOP

END

Çözümü yapılacak olan sistemi tanımlayan bilgileri programa giriş bilgileri olarak girebilmek için kullanıcıların yukarıdaki çağırın programa ek olarak aşağıda verilen iki altprogramla bir fonksiyon altprogramını da yazmaları gerekmektedir.

SUBROUTINE OKU1 (SSN, NSV, NOV, NT, NF, NKT, NCE, CN, CSN, D1, NDT,
IVS, NZV, ZV, TTO, ICV, DELY, YREF)

SUBROUTINE OKU2 (A, B, C, D, F, G, DKM, DRM, GDM, SSCS, CS, FRQ, X0, T0,
NSX)

COMPLEX FUNCTION U(JJ)

İlk iki altprogramın ikinci satırları tip belirtme deyimleri olmalıdır. Bu satırlarda alt programlardaki argümanların vektörel boyutları belirtilmelidir. Ancak bu vektörlerin gerçek boyutları ana programda tanımlandığı için burada, NSV(1), A(1), B(1),... şeklinde yazılabilir. Argümanların tipleri de daha sonra verilecek olan tablolar yardımıyla belirlenebilir. Bu argümanlardan bazıları aşağıda açıklanmıştır.

SSN: Sistemin girebileceği durumların sayısı

NSV: Bütün durumlardaki durum değişkeni sayısını gösteren vektör

NOV: Çıkış büyüklüklerinin sayısı

D1: Bir matrisin üstel fonksiyonu seriye açma yöntemi ile hesaplanırken terimler, son eklenen terimin mutlak normunun önceki toplamının mutlak normuna oranı D1'den büyük veya eşit oluncaya kadar toplanır. Bu işlemler EXPMT alt programında gerçekleştirilir. Matris, ilk olarak mutlak normu, 1' den küçük veya eşit olana kadar her defasında 2' ye bölünür. Daha sonra alışılmış Maclaurin serisi açılımıyla üstel fonksiyon bulunur. Son olarak sonuç, orijinal matrisin 2'ye bölündüğü sayı kadar arka arkaya kendi kendisiyle çarpılır. Böylece orijinal matrisin üstel fonksiyonu hesaplanmış olur (Köksal 1984a).

NT: Sürekli hal çözümü yapılırken bir periyotta kaç kez çıkış büyüklüklerinin hesaplanacağını gösteren değişken

NF: Farklı frekansa sahip girişlerin sayısı

NKT: Kaynak fonksiyonlarının türev mertebesi

T0: Tam çözümün başlatılacağı zaman değeridir ve genellikle sıfır olarak girilir. Daha sonra program içerisinde değişebilir.

Kullanıcı programları arasında yer alan fonksiyon altprogramı ile ilgili olarak da şu açıklamalar yapılabilir: Daha önce de belirtildiği gibi sistemin giriş fonksiyonu,

$$u_i(t) = U_i e^{p_i t}, \quad p_i = j \omega_i \quad \text{ve} \quad i=1,2, \dots, NF$$

şeklindeydi. $e^{p_i t}$ değeri daha önce çözümün içerisine dahil edildiği için fonksiyon altprogramında U_i değerleri girilir. Kompleks bir büyüklük olan bu değerler girilirken esas geçerli olan girişin $u_i(t)$ ' nin reel kısmı olduğu yani $u_{ai}(t)$ geçerli olan girişi göstermek üzere,

$$u_{ai}(t) = \text{Re} \left[u_i(p) e^{p_i t} \right]$$

olduğu göz önünde bulundurulmalıdır. Zaten bu gerçek nedeniyle çıkış olarak, elde edilen çözümlerin reel kısımları alınmaktadır.

3.5. TAMCOZ Altprogramı

Sistemin sürekli hal davranışının belirlenmesinde önemli bir adım olan genel çözümün bulunması TAMCOZ altprogramı yardımıyla gerçekleştirilir. Verilen t zaman değerleri için bölüm 2.4.3' de verilen denklemler çözülerek çıkış büyüklükleri hesaplanır. Her durumda o duruma ilişkin bilgiler kullanılır ve o duruma girildiği andaki zaman değeri de çözümün başladığı zaman olarak alınır. Bir çok altprogramı olan TAMCOZ' un alt programları ve argümanları alt bölümlerde açıklanacaktır.

3.5.1. TAMCOZ' un değişken isimleri

Bu altprogramda kullanılan bazı değişkenler Tablo 3 .1' de verilmiştir.

Tablo 3.1 TAMCOZ altprogramının bazı değişkenleri

Değişken ismi	Değişken tipi	Değişken boyutu	İlgili altprogram	Teorideki notasyon
A	R	$\sum_{I=1}^{NSI} NSY(I).NSY(I)$	---	A_i
B	C	$\sum_{I=1}^{NSI} NSY(I).NF$	---	B_i
C	R	$\sum_{I=1}^{NSI} NSV(I).M$	---	C_i
D	C	$M.NF.NSI$	---	D_i
F	R	$\sum_{I=1}^{NSI} \sum_{J=1}^{NSI} NSY(I).NSY(J)$	---	$F_{i,i+1}$
G	R	$S. \sum_{I=1}^{NSI} NSY(I).NF$	---	$G_{i,i+1}$
NSV	I	NSI	---	n_i
FRQ	R	NF	---	N_f
FIC	R	$NSY(I).NSY(I)$	EAT	$e^{A_i(t-t_{n,i-1})}$
FIB	R	$NSY(I)$	EATB	$e^{A_i(t-t_{n,i-1})} . B_i$
FIX	R	$NSY(I)$	EATX	$e^{A_i(t-t_{n,i-1})} . x_{n,i}(t_{n,i-1})$
YT	R	M	TAMCOZ	$Y(t)$
FREQ	C	NF	---	p_i
API	C	$NSY(I).NSY(I)$	APIT	$(A_i - p_i)^{-1}$
APIB	C	$NSY(I)$	APITB	$(A_i - p_i)^{-1} . B_i$
AFIB	C	$NSY(I)$	AEATB	$(A_i - p_i)^{-1} . e^{A_i(t-t_{n,i-1})} . B_i$
PUAB	C	$NSY(I)$	TAMCOZ	$U . e^{pt} . (A_i - p_i)^{-1} . B_i$
PUAF	C	$NSY(I)$	TAMCOZ	$U . e^{pt_{n,i-1}} . (A_i - p_i)^{-1} . e^{A_i(t-t_{n,i-1})} . B_i$
DUP	C	M	TAMCOZ	$D_i e^{pt} U$
XO	R	$NSY(I)$	ILKD	$x_{n,i}(t_{n,i-1})$
BJ	R	$NSY(I)$	---	$B_{i,j}$
DJ	R	M	---	$D_{i,j}$
HX1	C	$NSY(I)$	TAMCOZ	$PUAF + PUAB$
HX	C	$NSY(I)$	TAMCOZ	$FIX + HXI$
CHX	C	M	TAMCOZ	$C_i . HX$

Tablonun birinci kolonunda deęişken isimleri, ikinci kolonunda deęişken tipleri, üçüncü kolonunda vektör olarak boyutları, dördüncü kolonunda hesaplandıkları altprogram isimleri ve son kolonunda da teoride kullanılan notasyonlar gösterilmiştir.

Deęişkenlerin bir kısmı kullanıcı altprogramları aracılığı ile girilen bir kısmı da TAMCOZ'un alt programlarında hesaplanan deęişkenlerdir. Tabloda verilen deęişkenlerin dışında tanımlanması gereken bazı deęişkenlerde şunlardır:

K: Yapılan çözümün hangi sistem durumu için yapıldığını gösteren deęişken

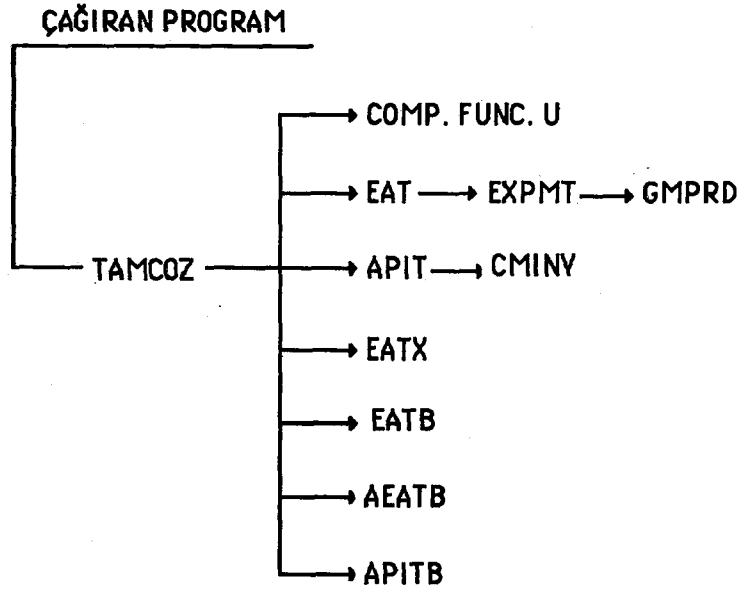
T: Çözüm için geçerli olan zaman deęeri

O, P, Q,R, IR: Ara işlemlerde kullanılan yardımcı vektörlerdir.

3.5.2. TAMCOZ' un altprogramları

Tam çözümü bulan TAMCOZ altprogramı dokuz altprogram ve bir fonksiyon altprogramından oluşmuştur. Bu altprogramlardan bazıları APEN1 altprogramı ile ortak kullanılmaktadır. Bunlar, matris işlemlerini gerçekleştiren altprogramlarla kullanıcı tarafından yazılan fonksiyon altprogramıdır. Altprogramların birbirleriyle ve çağırılan programla olan ilişkilerini gösteren bir şema Şekil 3.3' te verilmiştir.

Önemli olan bazı deęişkenlerin hangi altprogramlarda hesaplandığı Tablo 3.1' de verilmiştir. Dolayısıyla EAT, EATX, EATB, APIT ve APITB gibi altprogramlarda hangi deęişkenlerin hesaplandığı bu tablodan görülebilir. COMPLEX FUNCTION U altprogramı da kullanıcı programları arasında anlatılmıştı. Diğer altprogramlardan GMPRD, genel matris çarpımı yapan, CMINV kompleks matrisin inversini alan ve EXPMT bir matrisin üstel fonksiyonunu bulan altprogramlardır.



Şekil 3.3 TAMCOZ tarafından kullanılan altprogramlar

3.6. APEN1 Altprogramı

Sistemin durumlar bakımından sürekli hal davranışı belirlendikten sonra sürekli hal çözümü APEN1 altprogramı yardımıyla elde edilir. Bu altprogram yeni hazırlanmamıştır. Gelişimi şöyle özetlenebilir: İlk olarak anahtarlar içeren lineer devrelerin analizi için MAINLN programı hazırlanmıştır. Daha sonra bu program Türkbeyler (Küpeli) tarafından güç elektroniği devrelerinin analizini yapmak için düzenlenmiştir. Yapılan bu düzenleme ile elde edilen APEN programı, her aralığında aynı sayıda durum değişkeni olan devreleri çözebilecek durumda idi. Bu durumda, bütün A_k ' ların boyutları aynı ve bütün $F_{i,j}$ ' ler kare matris şeklinde oluyordu. Bunların da programlama açısından ne kadar kolaylık sağlayacağı açıktır. Ancak bu çalışmada her aralıkta farklı sayıda durum değişkeni olabileceği göz önüne alındığı için daha önce anlatılan program kısımları buna göre hazırlanmıştı. O zaman bu kısmın da aynı şekilde olması gerekiyordu. Bunun için APEN programı, her aralıkta

farklı sayıda durum değişkeni olduğu zaman da çözüm yapılabilecek şekilde geliştirilmiştir. Elde edilen bu yeni program, altprogram olarak kullanılmış ve APEN1 olarak isimlendirilmiştir. Genel durumda bütün A_k ' ların farklı boyutlarda alacağı ve $F_{i,j}$ ' lerin kare formun dışında dikdörtgen şeklinde de olacağı düşünülürse yapılan geliştirmenin önemi daha iyi anlaşılabilir. Özellikle $N_{k,j}$ ' lerin hesaplanmasında çok zorluklarla karşılaşmıştır. APEN1'in kullandığı değişkenler ve altprogramlar alt bölümlerde açıklanacaktır.

3.6.1. APEN1' in değişken isimleri

Bu altprogramda kullanılan değişkenlerin bazıları diğer altprogramların değişkenleri ile aynı olduğu için daha önce açıklanmışlardı. Bunların dışında kalan ve vektör şeklinde olan değişkenler Tablo 3.2' de açıklanmıştır. Tabloda verilmeyen değişkenlerden bazıları da şunlardır:

T0: Sistemin ana periyodu

K: Sürekli haldeki aralıkların sayısı

NRT: Kesin değeri bilinmeyen geçiş zamanlarının sayısı

O, P, Q, R, IQ, IR ve OPQ: Ara işlemlerde kullanılan yardımcı vektörler

CO, CP, CQ, CB1 ve CB2: Ara işlemlerde kullanılan yardımcı kompleks

vektörlerdir.

Tablo incelenirken değişkenlerin boyutları ile ilgili sütunda,

$$\sum_{i=1}^K NSV(i+1) \cdot NSV(i)$$

toplamında $NSV(K+1) = NSV(1)$ olduğu dikkate alınmalıdır.

Tablo 3.2 APEN1 altprogramının bazı değişkenleri

Değişken ismi	Değişken tipi	Değişken boyutu	İlgili altprogram	Teorideki notasyon
ETA	R	K	—	η_k
T	R	K	APEN1	τ_k
TIMEV	R	NT	SRTIM	t
FI	R	$\sum_{I=1}^K NSY(I+1) \cdot NSY(I)$	SFI	$F_k e^{A_k \tau_k}$
EN	R	$\sum_{I=1}^K \sum_{J=1}^K NSY(I) \cdot NSY(J+1)$	SEN	$N_{k,j}$
EM	R	$\sum_{J=1}^K NSY(J) \cdot NSY(J)$	SEM	M_k
API	R	$\sum_{I=1}^K NSY(I+1)$	SAPI	$(A_k - pI)^{-1}$
BB	R	$\sum_{I=1}^K NSY(I+1)$	SBB	$API(FI - F_k e^{p \tau_k}) B_k - G_k e^{p \tau_k}$
TRANSF	C	M.K	SHPT	H(p,t)
H	C	$\sum_{I=1}^K NSY(I)$	SH	H_k
EJ	C	$\sum_{I=1}^K NSY(I)$	SEJ	J_k
EP	C	$\sum_{I=1}^K NSY(I)$	SHPT	P_k
DEPO	R	M.NT	APEN1	y(t)
ICY	R	NRT	APEN1	—
IVRT	R	NRT	APEN1	—
YREF	R	NRT	APEN1	y_{ref}
DELY	R	NRT	APEN1	Δy

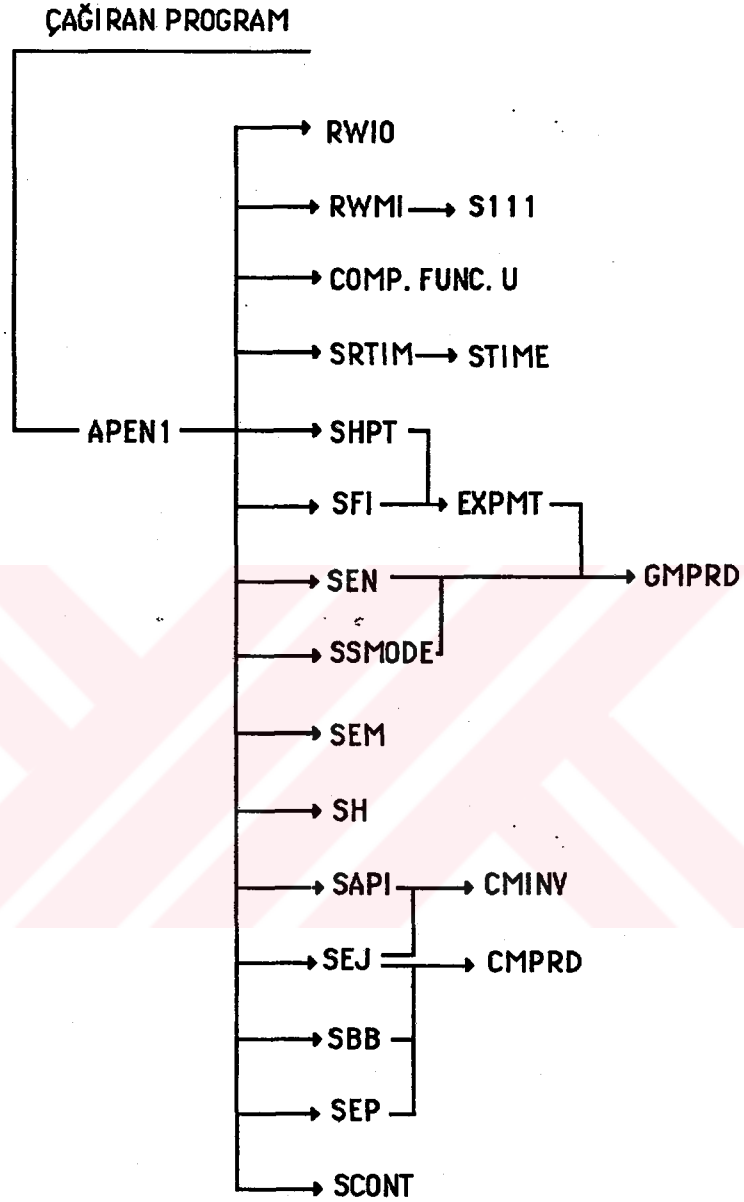
3.6.2. APEN1'in altprogramları

Sürekli hal çözümünü yapan APEN1 altprogramı birçok altprogramdan oluşmaktadır. Bu altprogramlardan bazıları TAMCOZ tarafından da kullanılmaktadır. Çağırılan program ve altprogramlar arasındaki ilişkileri gösteren bir şema Şekil 3.4.' de verilmiştir.

APEN1 altprogramı genelleştirilmeden önce giriş bilgilerini READO, SINPT ve STIME gibi altprogramlar ve U kompleks fonksiyon altprogramı yardımı ile alıyordu. Yeni durumda ise giriş bilgileri doğrudan çağırılan program (PGEDAN) tarafından aktarılmaktadır. Sadece U kompleks fonksiyonu eskisi gibi muhafaza edilmiş, diğer altprogramlardan READO ve SINPT iptal edilmiştir. STIME altprogramı ise verilen T0 ve NT değerlerine göre TIMEV değerlerini hesaplayan bir altprogram olarak kullanılmıştır.

Önceki alt bölümde verilen Tablo 3.2 yardımıyla hangi altprogramda hangi değişkenlerin hesaplandığı belirlenebilir. Matris işlemleri yapan altprogramlar daha önce TAMCOZ' ün altprogramları arasında kısaca açıklanmışlardı. Diğer altprogramlardan SCONT, bir peryot boyunca sıralanan aralıkların birbiri üzerine taşıp taşımadığını kontrol eder. Eğer bir taşma varsa bunu ortadan kaldırmak için gerekli işlemleri yapar.

SSMODE altprogramı ise verilen sistemin durumlarının ayrı ayrı asimtotik kararlılık testini yapmak için kullanılır. Bunun için, M' nin $k=2^n$. kuvveti olan M^k nin mutlak normu birden küçük veya bire eşit oluncaya kadar hesaplanır. Eğer 2^{INF} e kadar mutlak norm 1' in altına düşerse M istenilen özelliklere sahip demektir (Nacaroğlu 1989). O zaman işlemlere son verilir ve o duruma ilişkin kararlılık testi sonucu olarak "1" yazılır. Tersine ise sonuç "0" yazılır.



Şekil 3.4 APEN1 tarafından kullanılan altprogramlar

3.7. Programla İlgili Değerlendirmeler

Hazırlanan bilgisayar programı, başlangıçta birçok datanın bilindiği kabul edilerek hazırlanmıştır. Program tek başına düşünülüşü zaman bu bir eksiklik gibi görülebilir. Ama bu program tek başına değil, bu çalışmaya paralel olarak devam eden ve Demir tarafından yapılan (Demir (Devam ediyor)) çalışmanın sonucunda elde edilecek olan bilgisayar programı ile birlikte bir bütünlük oluşturacaktır. Demir tarafından gerçekleştirilen program, analizi yapılacak olan sistemle ilgili durum, çıkış, anahtarlama ve kontrol denklemlerini elde edecektir. Bu çalışma sonucunda oluşturulan program ise elde edilen bu datalara göre yukarı bölümlerde anlatılan analiz işlemlerini gerçekleştirecektir. Dolayısıyla gerekli olan dataların bilindiği kabul edilerek ve bir altprogramla alınabilecek şekilde program hazırlanmıştır.

Hazırlanan programın kullanılan bilgisayarla ilgili bazı değerleri de şu şekilde özetlenebilir:

Program FORTRAN 77 programlama dili ile yazılmıştır. Programın kullanıcı programları dışındaki hafıza ihtiyacı 44282 B' tır. Kullanıcı programlarının uzunluğu da girilen datalara bağlı olmakla birlikte ortalama 2425 B' tır. Örnekler çözülrken programın derlenmesi ve çalıştırılması 80386-33 mikroişlemcili ve matematiksel işlemciye sahip olan bir kişisel bilgisayarda gerçekleştirilmiştir. Programı derlemek için FTN77/386 fortran derleyici programı kullanılmış ve derleme işlemi ortalama 12.74 saniye sürmüştür. Derleme işleminden sonra programın çalışma zamanı da ortalama 15.15 saniye sürmektedir.

Aşağıda verilecek olan örnekler programın özelliklerinin daha iyi anlaşılmasını sağlayacaktır. Zaten bu nedenle örnek sayısı fazla tutulmuştur.

4. ÖRNEKLER

4.1. Giriş

Bu bölümde, geliştirilen yöntemin ve hazırlanan programın özelliklerini ortaya koyabilmek için değişik örnekler verilmiştir. Elde edilen sonuçları daha önce elde edilen sonuçlarla karşılaştırabilmek için bazı örnekler özellikle literatürde yer alan örnekler arasından seçilmiştir.

İlk üç örnek daha önce literatürde değişik kaynaklarda yer almıştır. Bu örnekler hem literatürdeki parametrelerle çözümlenerek aynı sonuçların elde edildiği gösterilmiş hem de değişik parametrelerle çözümlenerek bulunan yeni sonuçlar yorumlanmıştır. Özellikle üçüncü örnekte iki ayrı modelleme uygulanarak aynı sonuçlar elde edilmiştir.

Dördüncü örnek, bazı durumlarında durum değişkeni bulunmayan devrelerin çözümünü göstermek için seçilmiştir. Ayrıca bu örnek durum veya çıkış denklemlerinde kaynak fonksiyonlarının türevlerinin de yer alması halindeki işlemleri de aydınlatır.

Beşinci örnekte sürekli hale ulaşabilmesi için uzun süre gereken sistemlere örnek olabilecek bir devre ele alınmıştır. Bu devrenin sürekli hal çözümü sonuçlarının sayısal integrasyon yöntemlerine göre ne kadar kısa sürede bulunduğu gösterilmiştir.

Altıncı örnekte karakteristikleri histerisizli olan doğrusal olmayan elemanlar içeren devrelere örnek olacak bir devrenin çözümü gerçekleştirilmiştir. Özellikle histerisiz etkisinin gösterilebilmesi için devre mümkün olduğu kadar basit seçilmiştir.

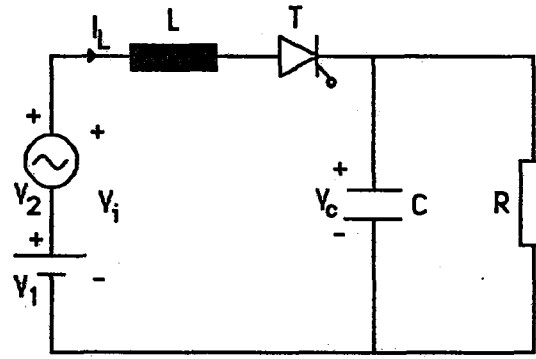
Son örnekte ise üç fazlı, kontrollü doğrultucu (O3) devresi ile beslenen RL yükü analiz edilmiştir. Farklı RL değerleri için devrenin sürekli hal davranışının nasıl değiştiği açıkça ortaya konulmuştur.

Bu örneklerle ilgili geniş bilgiler ve çözüm sonuçları aşağıda verilmiştir.

4.2. Örnek 1

Bu örnekte ele alınan Şekil 4.1 deki devre daha önce literatürde yer almıştır (Türkbeyler 1987). Devrede doğrusal olmayan eleman olarak bir tristör bulunmaktadır. Bu tristör bir ideal anahtar olarak modellendiği zaman devre iki ayrı duruma sahip olur. Anahtarın açık veya kapalı olmasına karşılık gelen bu durumların bir periyot boyunca nasıl sıralanacakları ve ne kadar süreyle devam edecekleri belli değildir. Belli olan tek zaman değeri tristörün tetiklendiği andır. Bu anda iletme geçen tristör akımının sıfır olmasıyla doğal olarak kesime gidecektir. Literatürdeki çözümde bu zamanın da yaklaşık olarak bilindiği kabul edilmişti. Bu kabul, değişik devre parametreleri ve değişik tetikleme zamanları için farklı bilgilerin önceden bilinmesini gerektirmektedir. Burada ise önce bu değerler bulunmakta ve daha sonra bu değerlere göre çözüm yapılmaktadır. Dolayısıyla her zaman için önceden bir tahminde bulunmaksızın çözüm işlemi gerçekleştirilebilmektedir. Ayrıca orada sistemin birinci durumunda tek durum değişkeni olduğu halde iki durum değişkeni olduğu kabul edilmiştir. Çünkü program her durumda aynı sayıda durum değişkeni olacağı kabul edilerek hazırlanmıştır. Burada ise genelleştirme yapıldığı için gerçeğe uygun şekilde tek durum değişkeni alınarak çözüm yapılmış ve aynı sonuçlar elde edilmiştir.

Devrenin girebileceği durumlara ilişkin bilgiler aşağıda verilmiştir:



$$\begin{aligned} V_1 &= 10 \text{ V} \\ V_2 &= 10 \sin(\omega t + \pi/6) \\ R &= 1 \text{ k}\Omega \\ C &= 22 \text{ nF} \\ L &= 20 \text{ mH} \\ f_2 &= 5 \text{ kHz} \end{aligned}$$

Şekil 4.1 Örnek 1' e ilişkin devre

1. Durum (T tristörü kesimde):

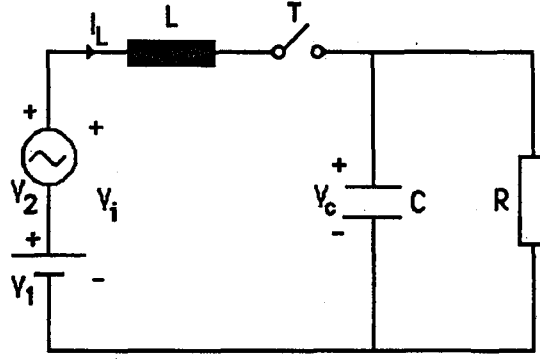
$$\dot{v}_c = [-1/RC] v_c + [0 \quad 0] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (4.1)$$

$$\begin{bmatrix} \dot{i}_L \\ v_T \\ v_L \\ v_C \\ \dot{i}_C \\ v_i \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ -1/R \\ 0 \end{bmatrix} v_c + \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (4.2)$$

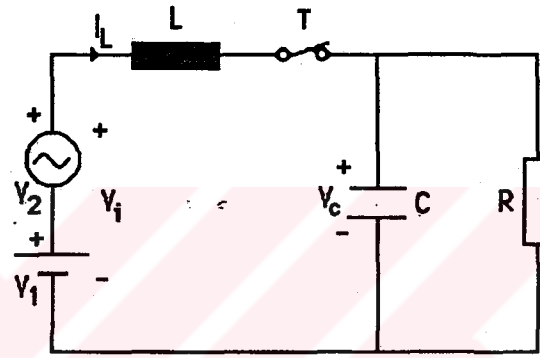
2. Durum (T tristörü iletimde):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_c \end{bmatrix} = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/RC \end{bmatrix} \begin{bmatrix} i_L \\ v_c \end{bmatrix} + \begin{bmatrix} 1/L & 1/L \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (4.3)$$

$$\begin{bmatrix} \dot{i}_L \\ v_T \\ v_L \\ v_C \\ \dot{i}_C \\ v_i \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & -1/R \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_c \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (4.4)$$



Şekil 4.2 Sistemin 1. durumu için eşdeğer devre



Şekil 4.3 Sistemin 2. durumu için eşdeğer devre

Sistemin yukarıda belirtilen durumlardan birinden diğerine geçerkenki davranışı genel olarak (2.28)' de gösterilen anahtarlama denklemleri ile belirlenir. Bu denklemlere ilişkin katsayı matrisleri (4.5)' de verilmiştir.

$$F_{1,2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad G_{1,2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (4.5)$$

$$F_{2,1} = [0 \quad 1] \quad G_{2,1} = [0 \quad 0]$$

Devrenin durumlarını belirleyen tristör için aşağıdaki kontrol denklemleri kullanılmıştır. Hangi duruma ilişkin kontrol denklemleri sağlanıyorsa ona göre

tristörün dolayısıyla devrenin durumu belirlenmiştir. Bu denklemlerde y , (4.2) ve (4.4) denklemlerindeki çıkış büyüklükleri vektörüdür.

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad z_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Bu örnek için yapılan tanımlamalar ve daha önce 3. bölümde yapılmış olan açıklamalar ışığında, durum ve çıkış denklemleri ile anahtarlama ve kontrol denklemlerine ilaveten aşağıdaki datalar da bilgisayara girilir.

CN=1

CSN(1)=2

SSN=2

NSV=[1 2]

NOV=6

NF=2

TT0=0.2E-3

FRQ=[0.1E-10 5000]

NZV=[0 1]

ZV=[0.01E-3]

NT=400

NCE=[2 2]

NDT=50

x0=[0 0]

ICV=[1]

DELY=[0.0001]

YREF=[0]

$$NSX = \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}$$

$$SSCS = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$GDM = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Programın birinci kısmında, ana periyodun 1/50 ' si kadar bir hassasiyetle yapılan çözümle sistemin bir periyot boyunca girdiği durumlar ve bu durumlara geçiş zamanlarının yaklaşık değerleri şu şekilde belirlenmiştir:

Sistem durumu

Yaklaşık geçiş zamanı

1

$\eta_1=0.0$

2

$\eta_2=0.1E-4$

1

$\eta_3=0.12E-3$

Programın ikinci kısmında ilk olarak bu yaklaşık geiş zamanlarının gerek deęerleri ana peryodun 1/10000' i kadar bir hassasiyetle belirlenmiřtir.

Bu deęerler,

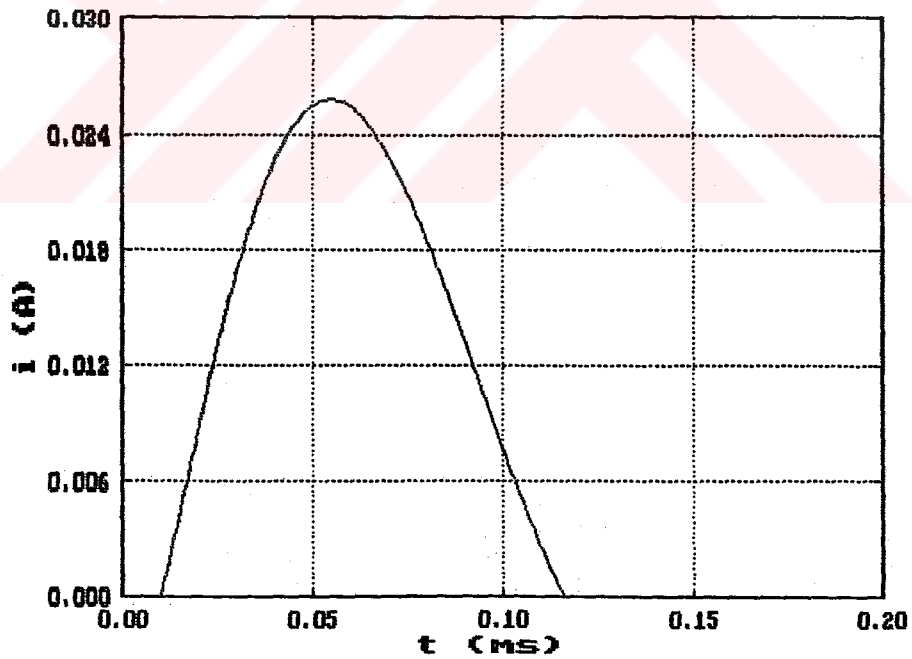
0.0

0.1E-4

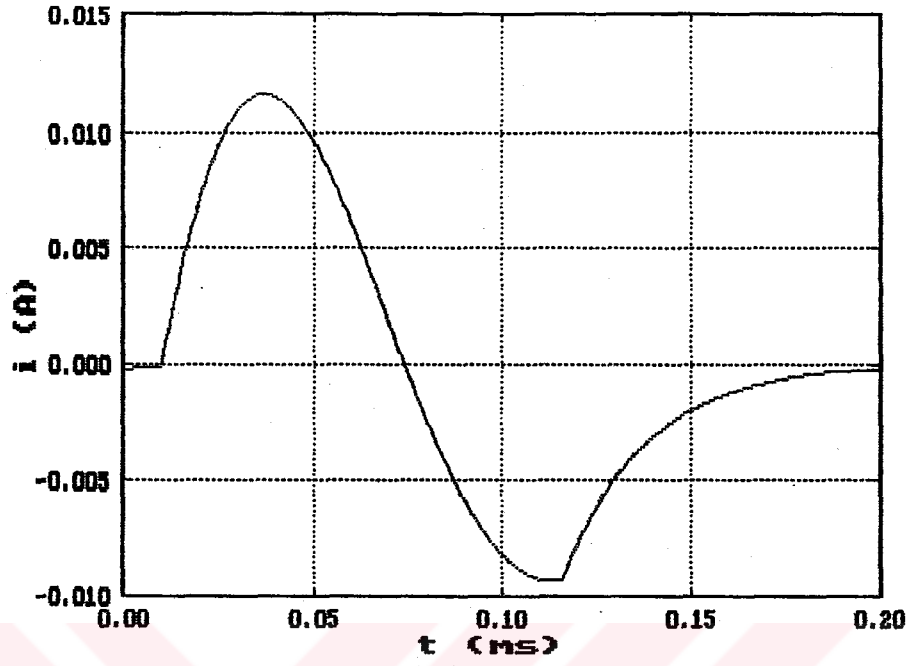
1.1604E-4

řeklinde bulunmuřtur. Daha sonra bu deęerlere gre de sistemin srekli hal gzm elde edilmiřtir. Elde edilen bu sonular řekil 4.4, 4.5, 4.6 ve 4.7' de verilmiřtir.

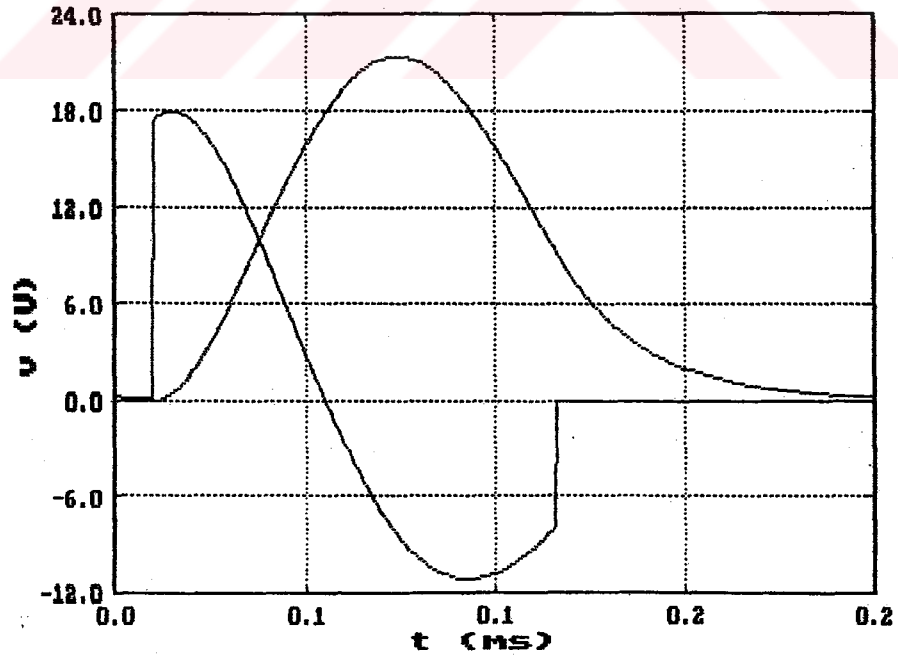
Bu sonular literatrdeki sonularla aynıdır.



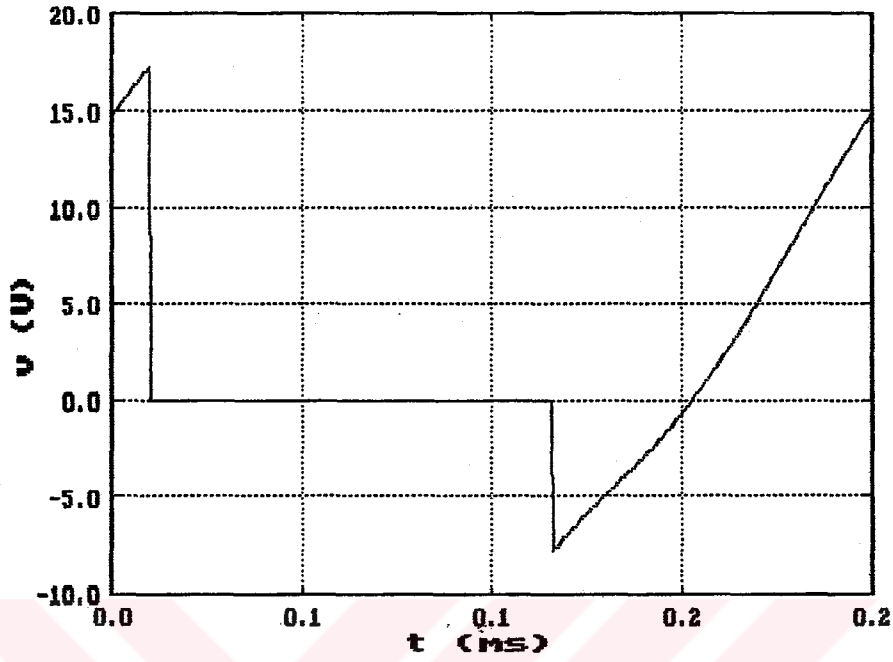
řekil 4.4 Tristor akımının (i_L) deęiřimi



Şekil 4.5 Kondansatör akımının (i_C) değişimi



Şekil 4.6 Kondansatör ve indüktans gerilimlerinin (v_C ve v_L) değişimleri



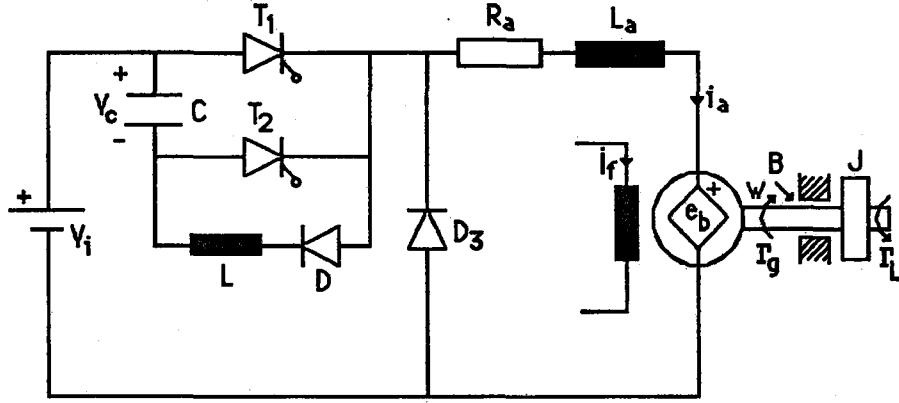
Şekil 4.7 Tristör geriliminin (v_T) değişimi

4.3. Örnek 2

Bu örnekte kıyıcı ile kontrol edilen dışardan uyarımlı bir dc motor analiz edilmiştir. Şekil 4.8' de görülen bu motor daha önce literatürde de yer almıştır (Türkbeyler 1987 ; Kılıcı ve Köksal 1987a ; Kılıcı ve Köksal 1987b). Böyle bir motorun mekaniksel davranışı,

$$J \frac{d\omega}{dt} + B\omega + \Gamma_L = K i_a = \Gamma_g \quad (4.6)$$

ile tanımlanır. Bu denklemdaki ve şekildeki terminoloji ve/veya kullanılan değerler aşağıdaki gibi özetlenebilir:



Şekil 4.8 Örnek 2'ye ilişkin devre

B: Sürtünme katsayısı (0.0029 Nm/rad/s)

J: Atalet katsayısı (0.0028 Kg.m²)

K: Emk (veya moment) sabiti (1.34 V/rad/s)

Γ_L : Yük momenti (0.0 Nm)

Γ_g : Üretilen moment (Nm)

V_i : Kaynak gerilimi (220 V)

i_a : Armatür akımı (A)

i_f : Uyartım akımı (A)

L_a : Armatür indüktansı (0.0726 H)

R_a : Armatür direnci (5.27 ohm)

TT0: Kıyıcının çalışma periyodu (s)

Analizi yapılan bu sistemde dört adet doğrusal olmayan eleman yer almaktadır. Ancak bunlardan D diyodu dikkate alınmayacaktır. Çünkü, kapasitenin yükü boşalırken T_1 , D ve L elemanları kayıpsız bir LC çevrimi oluşturmaktadırlar. Bu boşalma olayı da bilinmektedir. Bu nedenle D diyodu için yeni bir durum tanımlaması yapmak gereksizdir (Türkbeyler 1987). Geriye kalan üç elemanın her birisi iki ayrı durumda bulunabileceği için sistemin girebileceği değişik durum sayısı en fazla sekiz olabilir ($S_m=8$). Ancak bu durumlardan iki tanesi fiziksel

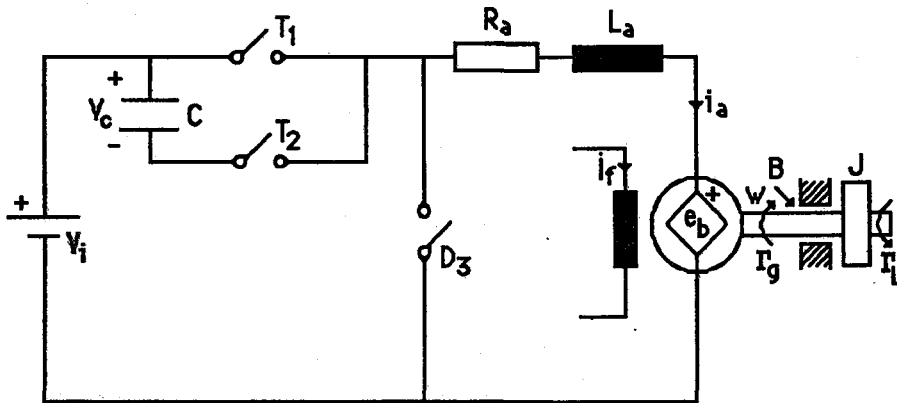
olarak gerilim kaynağını kısa devre yaptığı için mümkün olmadığından sistemin durum sayısı $S=6$ olmaktadır. Bu durumlara ilişkin şekiller ve denklemler aşağıda verilmiştir.

1. Durum (T_1, T_2, D_3 açık):

Bütün elemanların kesimde olduğu bu durum, sıfır akım aralığı olarak bilinir. Bu aralığa ilişkin eşdeğer devre Şekil 4.9' da verilmiştir. Bu devreye ilişkin durum ve çıkış denklemleri aşağıdaki gibidir.

$$\begin{bmatrix} \dot{i}_a \\ \dot{w} \\ \dot{v}_c \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -B/J & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ -\Gamma_L/J \\ 0 \end{bmatrix} [1] \quad (4.7)$$

$$\begin{bmatrix} i_a \\ w \\ v_c \\ v_3 \\ i_3 \\ v_1 \\ i_1 \\ v_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -K & 0 \\ 0 & 0 & 0 \\ 0 & -K & 0 \\ 0 & 0 & 0 \\ 0 & -K & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 220 \\ 0 \\ 220 \\ 0 \end{bmatrix} [1] \quad (4.8)$$



Şekil 4.9 Sistemin 1. durumu için eşdeğer devre

2. Durum (T_1 , T_2 açık, D_3 kapalı):

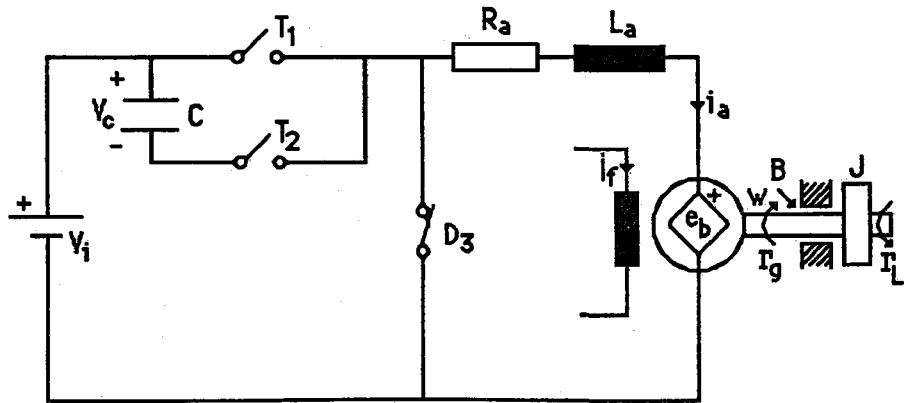
Şekil 4.10' da eşdeğer devresi verilen bu duruma (serbest dolaşım aralığı olarak bilinir) ilişkin denklemler (4.9) ve (4.10)' daki gibidir.

$$\begin{bmatrix} \dot{i}_a \\ \dot{w} \\ \dot{v}_c \end{bmatrix} = \begin{bmatrix} -R_a/L_a & -K/L_a & 0 \\ K/J & -B/J & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ -\Gamma_L/J \\ 0 \end{bmatrix} [1] \quad (4.9)$$

$$\begin{bmatrix} i_a \\ w \\ v_c \\ v_3 \\ i_3 \\ v_1 \\ i_1 \\ v_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 220 \\ 0 \\ 220 \\ 0 \end{bmatrix} [1] \quad (4.10)$$

3. Durum (T_1 , D_3 açık, T_2 kapalı):

Bu durum, komütasyon aralığı olarak bilinir. Bu ve bundan sonraki durumlarda devre şeması verilmeden sadece durum ve çıkış denklemleri verilecektir.



Şekil 4.10 Sistemin 2. durumu için eşdeğer devre

$$\begin{bmatrix} \dot{i}_a \\ \dot{w} \\ \dot{v}_c \end{bmatrix} = \begin{bmatrix} -R_a/L_a & -K/L_a & -1/L_a \\ K/J & -B/J & 0 \\ 1/C & 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 220/L_a \\ -\Gamma/J \\ 0 \end{bmatrix} [1] \quad (4.11)$$

$$\begin{bmatrix} i_a \\ w \\ v_c \\ v_3 \\ i_3 \\ v_1 \\ i_1 \\ v_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -220 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [1] \quad (4.12)$$

4. Durum (T_1 açık, T_2 , D_3 kapalı):

$$\begin{bmatrix} \dot{i}_a \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -R_a/L_a & -K/L_a \\ K/J & -B/J \end{bmatrix} \begin{bmatrix} i_L \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ -\Gamma/J \end{bmatrix} [1] \quad (4.13)$$

$$\begin{bmatrix} i_a \\ w \\ v_c \\ v_3 \\ i_3 \\ v_1 \\ i_1 \\ v_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 220 \\ 0 \\ 0 \\ 220 \\ 0 \\ 0 \\ 0 \end{bmatrix} [1] \quad (4.14)$$

5. Durum (T_2 , D_3 açık, T_1 kapalı):

Bu durum iş aralığı olarak bilinir.

$$\begin{bmatrix} \dot{i}_a \\ \dot{w} \\ \dot{v}_c \end{bmatrix} = \begin{bmatrix} -R_a/L_a & -K/L_a & 0 \\ K/J & -B/J & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 220/L_a \\ -\Gamma/J \\ 0 \end{bmatrix} [1] \quad (4.15)$$

$$\begin{bmatrix} i_a \\ w \\ v_c \\ v_3 \\ i_3 \\ v_1 \\ i_1 \\ v_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ w \\ v_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -220 \\ 0 \\ 0 \\ 0 \\ 220 \\ 0 \end{bmatrix} [1] \quad (4.16)$$

6. Durum (D_3 açık, T_1, T_2 kapalı):

$$\begin{bmatrix} \dot{i}_a \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -R_a/L_a & -K/L_a \\ K/J & -B/J \end{bmatrix} \begin{bmatrix} i_a \\ w \end{bmatrix} + \begin{bmatrix} 220/L_a \\ -\Gamma/J \end{bmatrix} [1] \quad (4.17)$$

$$\begin{bmatrix} i_a \\ w \\ v_c \\ v_3 \\ i_3 \\ v_1 \\ i_1 \\ v_2 \\ i_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -220 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [1] \quad (4.18)$$

Sistemin yukarıda belirtilen durumlarından birinden diğerine geçerkenki davranışı genel olarak (2.28)' de gösterilen anahtarlama denklemleri ile belirlenir. Bu denklemlere ilişkin katsayı matrisleri aşağıda verilmiştir.

$$F_{1,2} = F_{1,3} = F_{2,1} = F_{2,3} = F_{3,1} = F_{3,2} = F_{3,5} = F_{5,1} = F_{5,2} = F_{5,3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$F_{1,4} = F_{1,6} = F_{2,4} = F_{2,6} = F_{3,4} = F_{3,6} = F_{5,4} = F_{5,6} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$F_{4,1} = F_{4,2} = F_{4,3} = F_{4,5} = F_{6,1} = F_{6,2} = F_{6,3} = F_{6,5} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$F_{1,5} = F_{2,5} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad F_{4,6} = F_{6,4} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$G_{1,2} = G_{1,3} = G_{2,1} = G_{3,1} = G_{3,2} = G_{3,5} = G_{5,1} = G_{5,2} \\ = G_{5,2} = G_{4,1} = G_{4,2} = G_{4,3} = G_{4,5} = G_{6,1} = G_{6,2} = G_{6,3} = G_{6,5} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$G_{1,4} = G_{1,6} = G_{2,4} = G_{2,6} = G_{3,4} = G_{3,6} = G_{5,4} = G_{5,6} = G_{4,6} = G_{6,4} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$G_{1,5} = G_{2,5} = \begin{bmatrix} 0 \\ 0 \\ -2 \ 2 \ 0 \end{bmatrix}$$

Sistemin yukarda tanımlanan durumlardan hangisinde olduğunun belirlenmesinde aşağıdaki kontrol denklemlerinden yararlanır. Bu kontrol denklemleri yardımıyla elemanların durumları, daha sonra verilecek olan eleman durumu sistem durumu matrisi yardımıyla da sistem durumu belirlenir.

1. Eleman:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

2. Eleman:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

3. Eleman:

$$\begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$z_1 = z_2 = z_3 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Bu örnek için yapılan tanımlamalar ve daha önce 3. bölümde yapılmış olan açıklamalar ışığında, durum ve çıkış denklemleri ile anahtarlama ve kontrol denklemlerine ilaveten aşağıdaki datalar da bilgisayara girilmiştir.

$$CN=3$$

$$CSN=[2 \ 2 \ 2]$$

$$SSN=6$$

$$NSV=[3 \ 3 \ 3 \ 2 \ 3 \ 2]$$

$$NOV=9$$

$$NF=1$$

$$TT0=5.4.E-3$$

$$FRQ=[0.1E-10]$$

$$NZV=[0 \ 1 \ 0 \ 1 \ 0 \ 0]$$

$$ZV=[0.0 \ 0.01E-3]$$

$$NDT=50$$

$$NT=400$$

$$ICV=[4 \ 5]$$

$$DELY(4)=0.1$$

$$DELY(5)=0.0001$$

$$YREF=[0 \ 0]$$

$$GDM_1 = GDM_2 = GDM_3 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$NCE = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 2 \end{bmatrix}$$

$$x0 = \begin{bmatrix} 0 \\ 0 \\ 220 \end{bmatrix}$$

$$SSCS = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}$$

$$NSX = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 0 \\ 1 & 2 & 3 \\ 1 & 2 & 0 \end{bmatrix}$$

Yukarda verilen denklemler ve datalara göre programın birinci kısmı, sistemin sürekli halde girdiği durumları ve yaklaşık geçiş zamanlarını ana periyodun 1/50' si kadar bir hassasiyetle aşağıdaki gibi belirlemiştir.

Sistem durumu

Yaklaşık geçiş zamanı

5

$$\eta_1=0.0$$

3

$$\eta_2=0.2E-2$$

2

$$\eta_3=0.24E-2$$

1

 $\eta_4=0.41E-2$

Bu durumlar bir periyot boyunca sıralanarak dört aralık oluşturmaktadırlar. Bunlardan birincisi iş aralığı, ikincisi komütasyon (aktarım) aralığı, üçüncüsü serbest dolaşım aralığı ve dördüncüsü de sıfır akım aralığıdır.

Programın ikinci kısmında, yukarıda yaklaşık değerleri bulunan geçiş zamanlarının gerçek değerleri ana periyodun $1/10000$ ' i kadar hassasiyetle belirlenmiştir. Bu değerler,

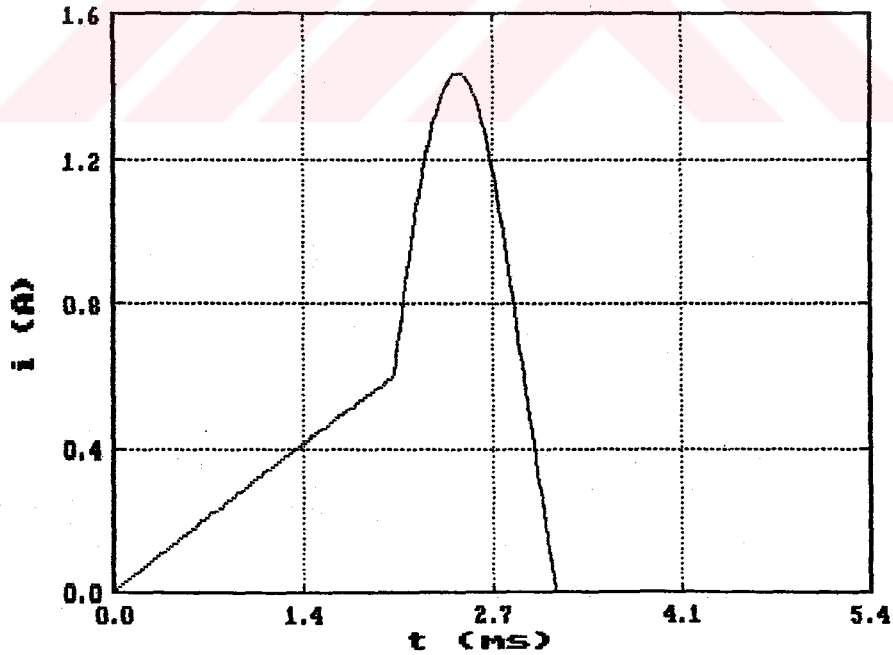
0.0

0.2E-2

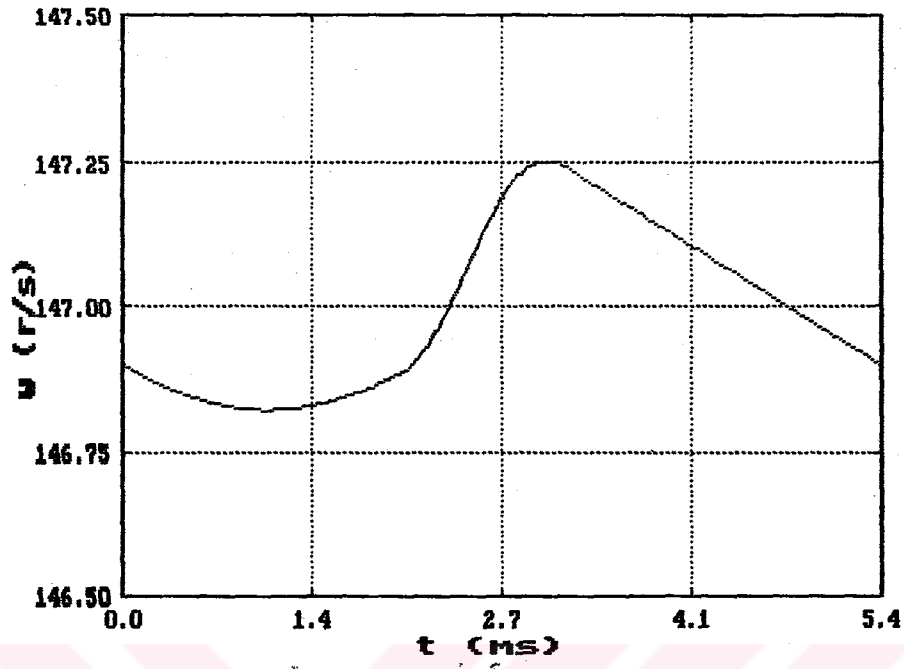
0.2818E-2

0.314E-2

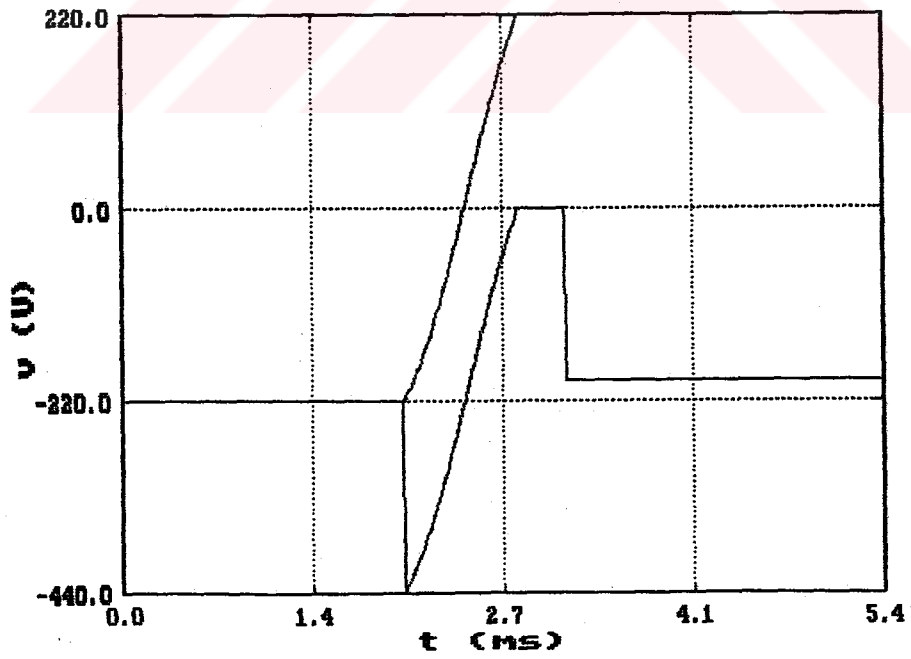
şeklindedir. Daha sonra bu değerlere göre bir periyot boyunca 400 değer için sürekli hal çözümü hesaplanmıştır. Hesaplanan sonuçlara göre çizilen grafikler Şekil 4.11, 4.12, 4.13 ve 4.14' de gösterilmiştir.

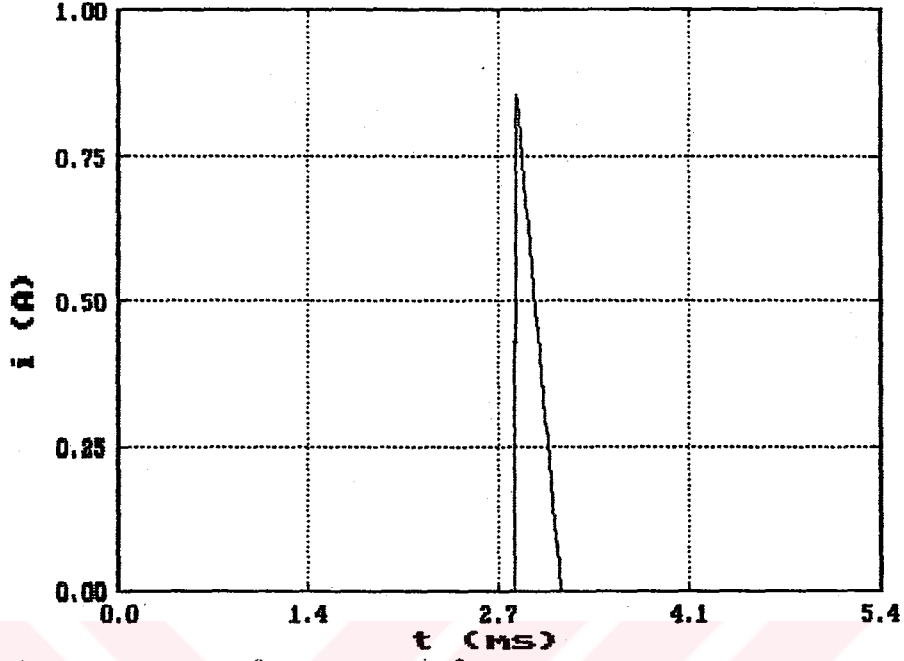


Şekil 4.11 Armatür akımının değişimi



Şekil 4.12 Açısal hızın değişimi

Şekil 4.13 D₃ diyotu ve kondansatör üzerindeki gerilimlerin değişimleri



Şekil 4.14 D₃ diyotunun akımının değişimi

Bir sistemde değişik parametreler için girilen sistem durumlarının değişebileceğini gösterebilmek için aynı örnek sadece aşağıdaki datalar değiştirilerek tekrar çözülmüştür.

$$L_a=0.1089 \text{ H}$$

$$B=0.05 \text{ Nm/rad/s}$$

$$ZV=[0.0 \quad 0.2E-3]$$

Yeni datalar için sistem durumları ve yaklaşık geçiş zamanları,

Sistem durumu	Yaklaşık geçiş zamanı
5	$\eta_1=0.0$
3	$\eta_2=0.2E-3$
2	$\eta_3=0.813E-3$

şeklinde bulunmuştur. Görüldüğü gibi sistem sadece üç değişik duruma girmektedir.

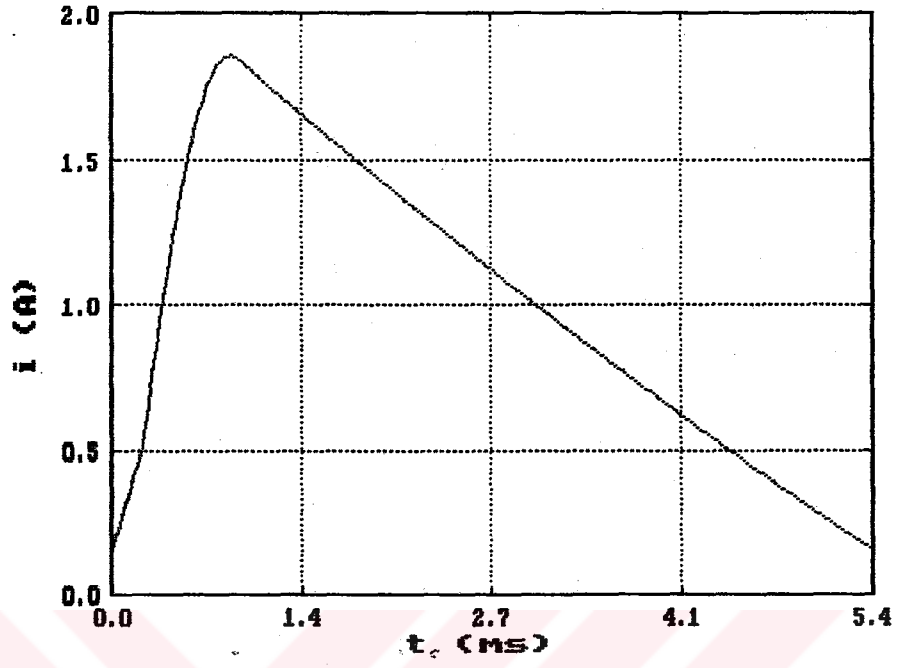
Bu durumlara ilişkin geçiş zamanları programın ikinci kısmında,

$$0.0$$

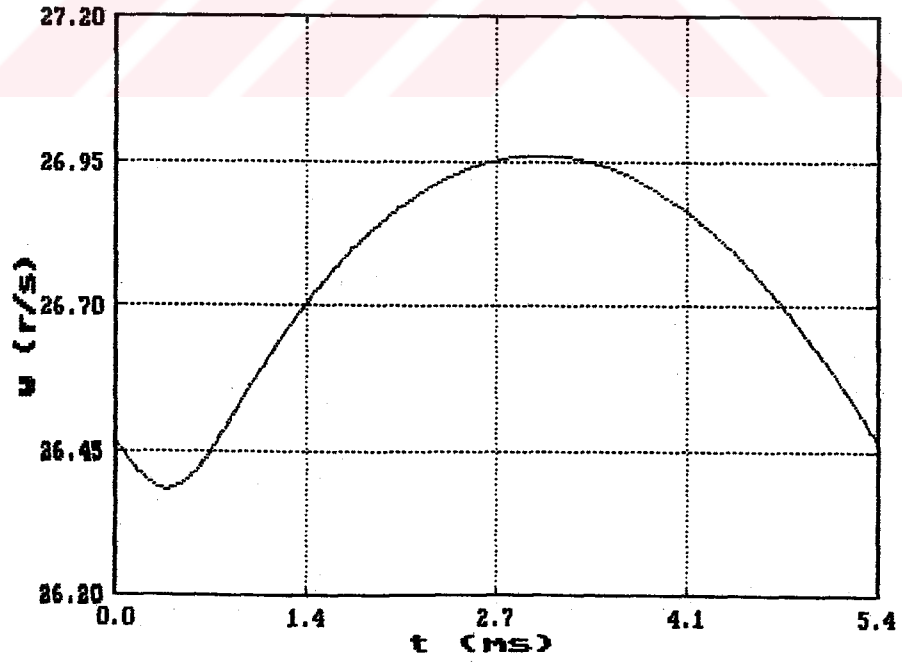
$$0.2E-3$$

$$0.8012E-3$$

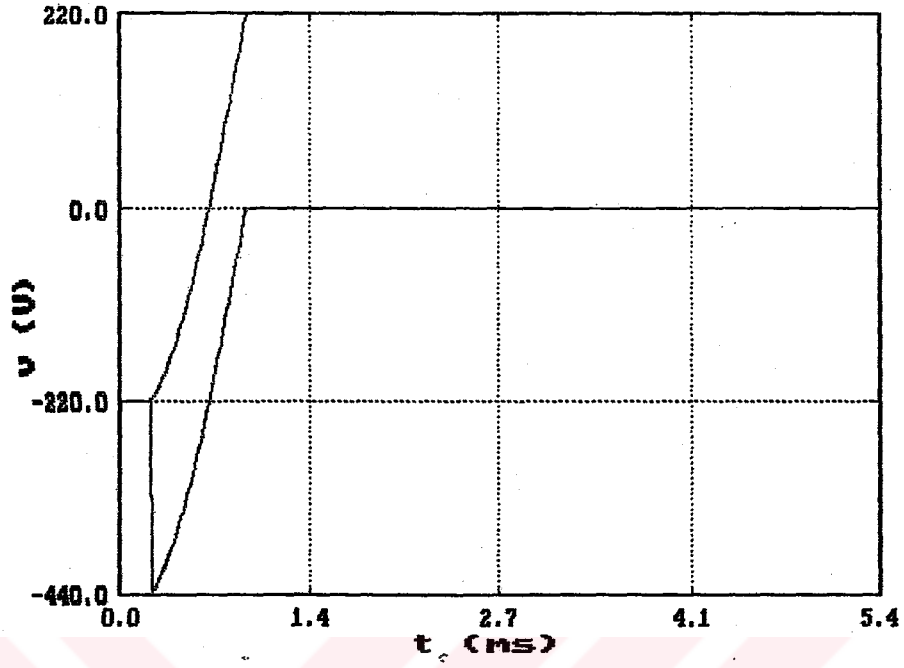
şeklinde bulunmuştur. Bu değerlere göre elde edilen çıkış büyüklüklerinin değişimleri Şekil 4.15, 4.16, 4.17 ve 4.18' de verilmiştir.



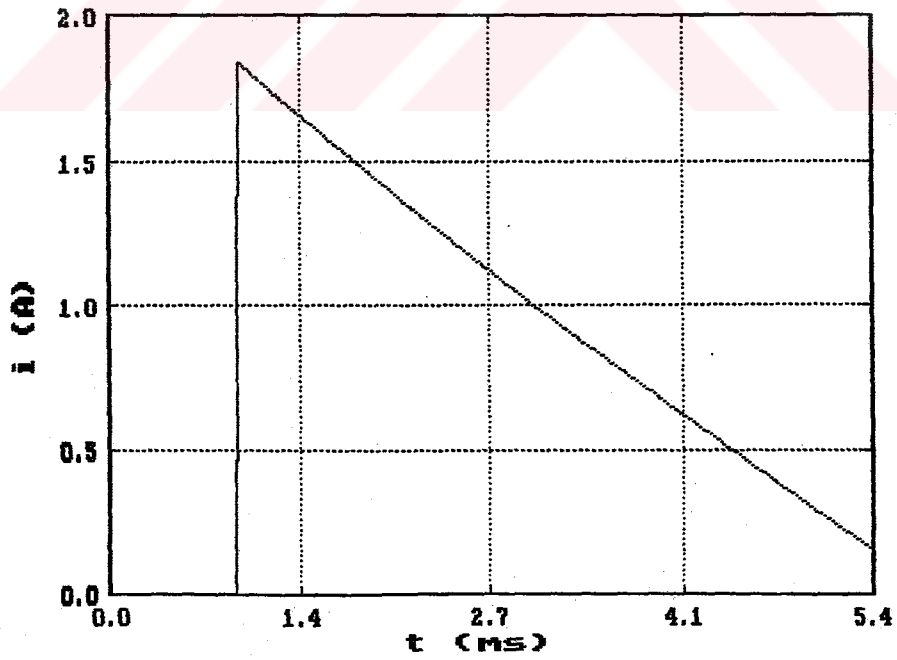
Şekil 4.15 Armatür akımının değişimi



Şekil 4.16 Açısal hızın değişimi



Şekil 4.17 D₃ diyotu ve kondansatör üzerindeki gerilimlerin değişimleri



Şekil 4.18 D₃ diyotunun akımının değişimi

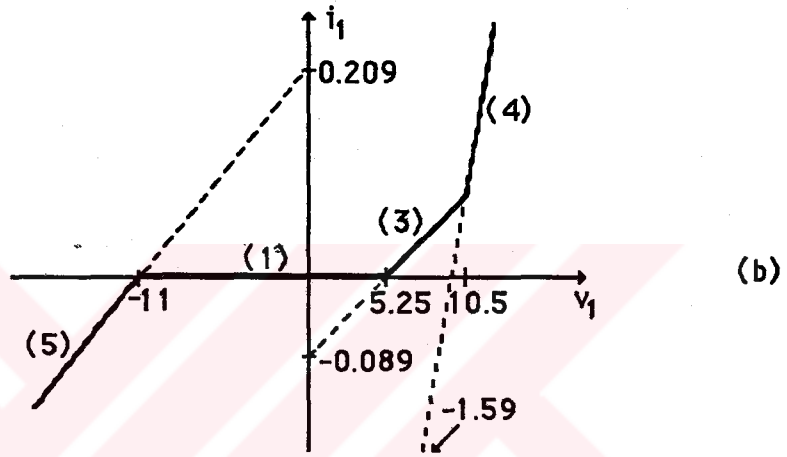
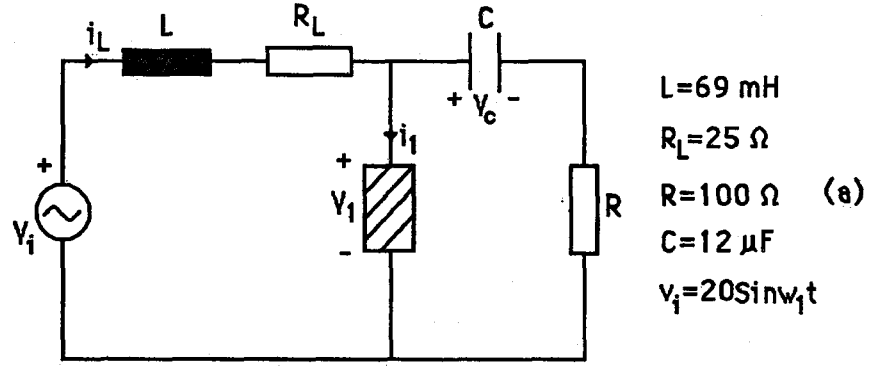
4.4. Örnek 3

Bu örnekte analiz edilen devre de daha önce literatürde yer almıştır (Türkbeyler 1987 ; Köksal ve Türkbeyler 1987 ; Türkbeyler ve Köksal 1988). Adı geçen kaynaklarda bu sistemin sürekli hal çözümü bulunmuştur. Ancak bu çözüm yapılırken sistemin sürekli halde bir periyot boyunca girdiği durumların kesin olarak ve bu durumlara geçiş zamanlarının da yaklaşık olarak bilindiği kabul edilmiştir. Sistemde yer alan elemanların veya kaynak geriliminin değişik değerleri için girilen sistem durumları ve geçiş zamanları değişebilecektir. Dolayısıyla literatürde yapılan kabuller her zaman geçerli olamayacaktır.

Burada aynı örnek değişik modeller ve değişik kaynak gerilimi değerleri için çözümler yapılarak sonuçlar karşılaştırılmıştır. Programın birinci kısmı tarafından belirlenen durumlar ve yaklaşık geçiş zamanlarına göre ikinci kısmı tarafından sürekli hal çözümü elde edildiğinden, literatürde verilen çözüm için zorunlu olan başlangıçtaki tahmini değerlere gerek kalmamakta ve böylece daha genel bir analiz yapılmış olmaktadır.

Analizi yapılan sistem Şekil 4.19a' da verilen ve doğrusal olmayan bir direnç içeren elektrik devresidir. Bu devrede yer alan doğrusal olmayan direncin akım-gerilim karakteristiği Şekil 4.19b' deki gibidir.

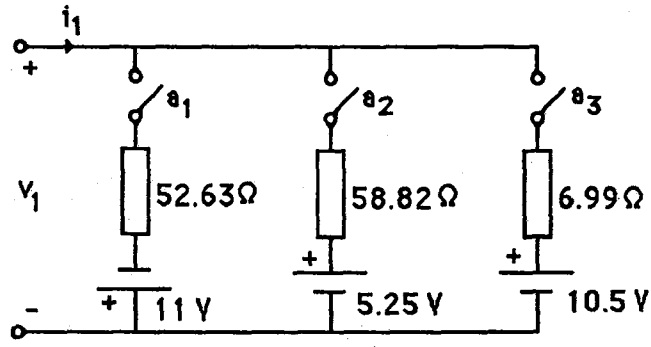
Devrede yer alan doğrusal olmayan eleman Şekil 4.20' de verilen eşdeğer devre ile gösterilebilir. Anahtarlar, dirençler ve gerilim kaynaklarından oluşan bu eşdeğer devrede anahtarların değişik kombinasyonlarla açılıp kapanmalarıyla oluşacak karakteristikler birleştirildiğinde, doğrusal olmayan eleman karakteristiğinin aynısı elde edilir. Dolayısıyla Şekil 4.19a ile verilen sistemin analizinin yapılması yerine, bu sistemdeki doğrusal olmayan elemana karşılık Şekil 4.20' deki eşdeğer devre bağlanmak suretiyle elde edilen yeni sistemin analizi yapılabilir.



Şekil 4.19 a) Örnek 3' e ilişkin elektrik devresi

b) Doğrusal olmayan direncin karakteristiği

Şekil 4.19b' de karakteristiği verilen elemanı modelleyen eşdeğer devrede (Şekil 4.20), anahtarların değişik kombinasyonlarla açılıp kapanmalarıyla, 8 farklı durumun elde edileceği düşünülebilir. Ancak bu eleman, karakteristiğinden de görülebileceği gibi fiziksel olarak dört durumda bulunabilir. Bu durumlar karakteristik üzerindeki 1, 3, 4, 5 ile gösterilen doğrusal bölgelere karşılık düşmektedirler. Buradan anlaşılmaktadır ki, anahtarlar birbirinden bağımsız olarak hareket edemezler.



Şekil 4.20 Doğrusal olmayan elemanın eşdeğer devresi

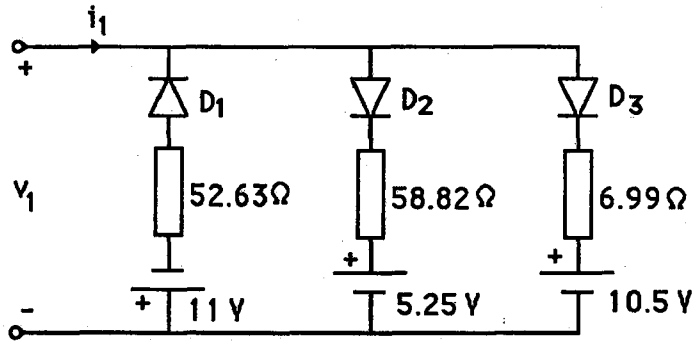
Anahtarlar birbirinden bağımsız olarak hareket edebilselerdi anahtar pozisyonlarına göre eleman durumları,

1. durum : a_1, a_2, a_3 açık
2. durum : a_1, a_2 açık, a_3 kapalı
3. durum : a_1, a_3 açık, a_2 kapalı
4. durum : a_1 açık, a_2, a_3 kapalı
5. durum : a_2, a_3 açık, a_1 kapalı
6. durum : a_2 açık, a_1, a_3 kapalı
7. durum : a_3 açık, a_1, a_2 kapalı
8. durum : a_1, a_2, a_3 kapalı

şeklinde olurdu.

Gerçekten anılan elemanın daha uygun ve yalnız dört duruma girebilen bir eşdeğer devresi ideal diyotlar kullanılarak Şekil 4.21' deki gibi gösterilebilir. Bu devrede, örneğin, D_1 kapalı iken D_2 ve D_3 kapalı olamazlar. Dolayısıyla fiziksel olarak sadece dört durum gerçekleşebilir.

Her ne kadar fiziksel olarak dört farklı durumun olabileceği gerekirse de, burada daha önceki uygulamadan farklı olarak başlangıçta sekiz ($=S_m$) farklı durum gözönüne alınmıştır. Çünkü mümkün olmayan durumlar zaten diyotların



Şekil 4.21 Doğrusal olmayan eleman için yeni eşdeğer devre

kontrol denklemleri vasıtasıyla hiçbir zaman erişilemeyecek durumlar olacak ve sonuçta değişik eleman ve kaynak değerlerine göre sistemin girebileceği durum sayısı dört ($=S$) olarak ortaya çıkacaktır.

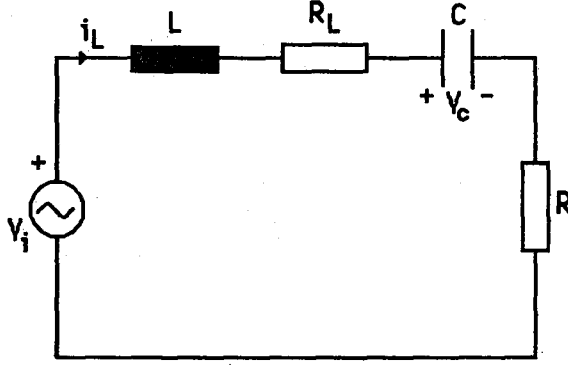
Gereksiz gibi görülebilecek bu uygulama, fiziksel olarak girebileceği durum sayısı önceden kestirilemeyen sistemlerin analizi sırasında da geliştirilen yöntemin ne denli olumlu sonuçlar verdiğini gösterebilmek için özellikle yapılmıştır.

Devrede yer alan diyotlardan herbirisi iki ayrı durumda (iletim, kesim) bulunabileceği için sistemin girebileceği durumların maksimum sayısı sekiz olur (1, 2, 3, 4, 5, 6, 7, 8 durumları).

Yukarda bahsolunan sekiz duruma ilişkin durum ve çıkış denklemleri aşağıda verilmiştir.

1. Durum (D_1, D_2, D_3 açık):

Bu durumda devre Şekil 4.22' deki şekle dönüşür. Bu devre için durum ve çıkış denklemleri kolayca (4.19) ve (4.20)' deki gibi bulunur.



Şekil 4.22 Sistemin 1. durumu için eşdeğer devre

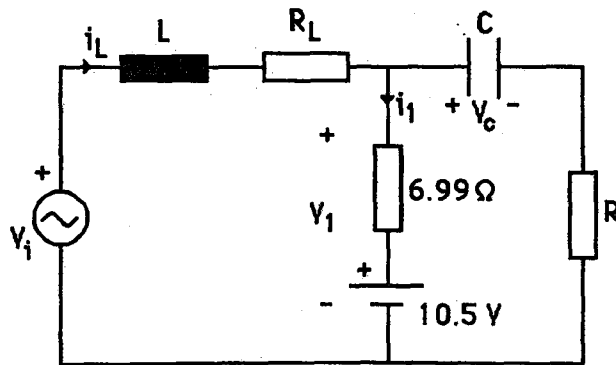
$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} 0 & 83333 \\ -14.49 & -1812 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 14.49 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.19)$$

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 1 & 100 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.20)$$

2. Durum (D_1 , D_2 açık, D_3 kapalı):

Bu durumda devre Şekil 4.23' deki gibi olup, durum ve çıkış denklemleri (4.21) ve (4.22)' deki gibi bulunur.

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -774.9 & 5484 \\ -0.94 & -456.93 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 8136.6 \\ 14.49 & -142.27 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.21)$$



Şekil 4.23 Sistemin 2. durumu için eşdeğer devre

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 0.065 & 6.5 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 9.81 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.22)$$

3. Durum (D_1, D_3 açık, D_2 kapalı):

Bu ve bundan sonraki durumlarda devre şeması verilmeden sadece durum ve çıkış denklemleri verilecektir.

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -525.62 & 30871 \\ -5.375 & -898.96 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 2759.5 \\ 14.49 & -47.85 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.23)$$

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 0.37 & 37.04 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 3.3 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.24)$$

4. Durum (D_1 açık, D_2, D_3 kapalı):

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -800 & 4933.3 \\ -0.867 & -447.5 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 7944 \\ 14.49 & -135.27 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.25)$$

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 0.0588 & 5.88 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 9.33 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.26)$$

5. Durum (D_2, D_3 açık, D_1 kapalı):

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -546.27 & 28738.33 \\ -5 & -862 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & -6008.97 \\ 14.49 & 104.39 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.27)$$

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 0.34 & 34.48 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & -7.2 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.28)$$

6. Durum (D_2 açık, D_1, D_3 kapalı):

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -810.37 & 4862.2 \\ -0.87 & -446.5 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 6458.65 \\ 14.49 & -108.55 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.29)$$

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 0.058 & 5.81 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 7.5 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.30)$$

7. Durum (D_3 açık, D_1, D_2 kapalı):

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -659.95 & 18118.52 \\ -3.19 & -677.37 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & -2197.63 \\ 14.49 & 37.63 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.31)$$

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 0.217 & 21.74 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & -2.61 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.32)$$

8. Durum (D_1, D_2, D_3 kapalı):

$$\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -781.51 & 4405.58 \\ -0.77 & -438.89 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 6102.54 \\ 14.49 & -105.78 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.33)$$

$$\begin{bmatrix} v_1 \\ v_C \\ i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 0.053 & 5.28 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 0.947 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.34)$$

Durumlar arası geçiş matrislerine gelince, değişik durumlarda hep aynı durum değişkenleri olduğundan ve bunlar da sürekli olduğundan bütün $F_{i,j}$ ' ler (2×2) birim matris ve bütün $G_{i,j}$ ' ler (2×2) sıfır matrislerdir. Eşdeğer devrede yer alan diyotlara ilişkin kontrol denklemleri de aşağıdaki gibi yazılmıştır.

1. Eleman:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} [y] \geq [-11]$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix} [y] \geq [11]$$

2. Eleman:

$$\begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix} [y] \geq [-5.25]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} [y] \geq [5.25]$$

3. Eleman:

$$\begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix} [y] \geq [-10.5]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} [y] \geq [10.5]$$

$$z_1 = z_2 = z_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Yukardaki denklemlere ek olarak aşağıdaki datalar da bilgisayara girilmiştir.

$$CN=3$$

$$CSN=[2 \ 2 \ 2]$$

$$SSN=8$$

$$NOV=4$$

$$NF=2$$

$$NDT=50$$

$$TT0=0.02$$

$$FRQ=[50 \ 0.1E-10]$$

$$NZV=[0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

$$ZV=[0.0]$$

$$DELY(1)=0.05$$

$$NT=400$$

$$ICV=[1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$NSV=[2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2 \ 2]$$

$$YREF=[5.25 \ 10.5 \ 10.5 \ 5.25 \ -11 \ 11]$$

$$GDM_1 = GDM_2 = GDM_3 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$NCE = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

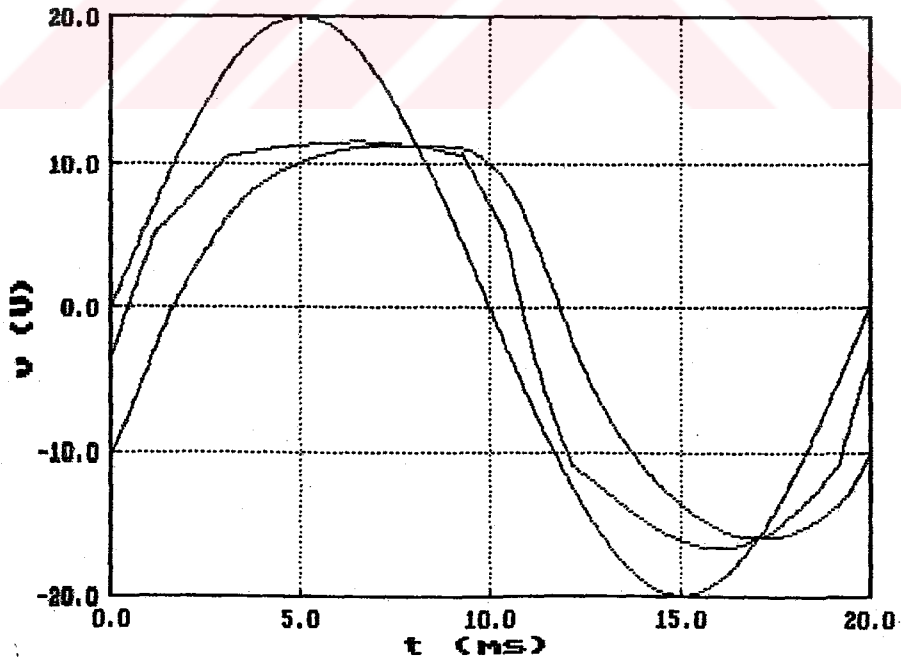
$$SSCS = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix}$$

$$NSX = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix}$$

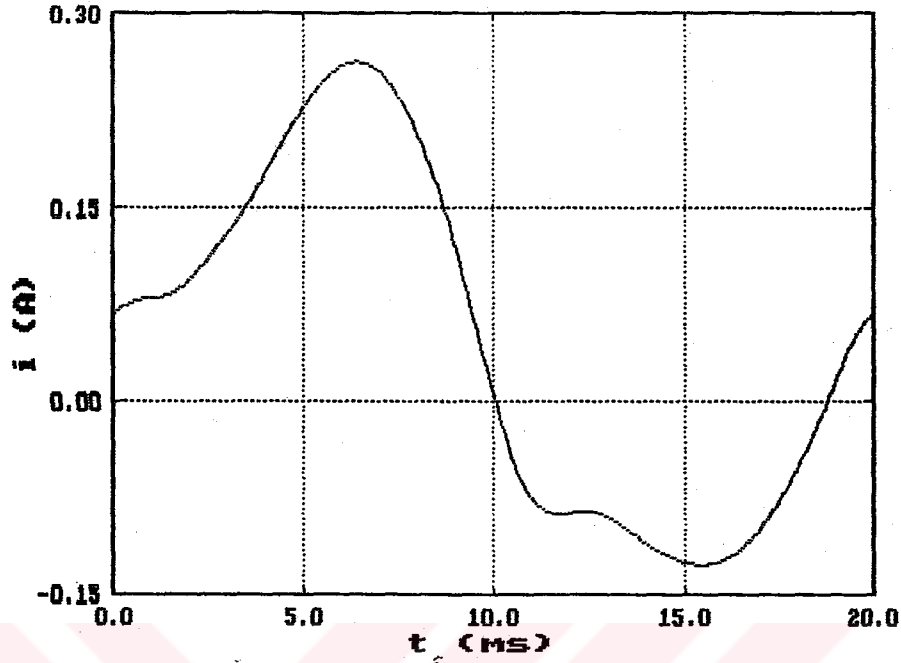
Yukarıda verilen denklemler ve datalar kullanılarak sistemin sürekli halde girdiği durumlar ve yaklaşık geçiş zamanları şu şekilde belirlenmiştir.

Sistem durumu	Yaklaşık geçiş zamanı
1	$\eta_1=0.0$
3	$\eta_2=0.16E-2$
4	$\eta_3=0.2E-2$
3	$\eta_4=0.96E-2$
1	$\eta_5=0.108E-1$
5	$\eta_6=0.124E-1$
1	$\eta_7=0.192E-1$

Bu durumlara ve yaklaşık geçiş zamanlarına göre programın ikinci kısmında yapılan sürekli hal çözümü sonuçlarına göre çizilen grafikler Şekil 4.24 ve 4.25' de verilmiştir.



Şekil 4.24 Giriş, kondansatör ve doğrusal olmayan eleman gerilimlerinin değişimi



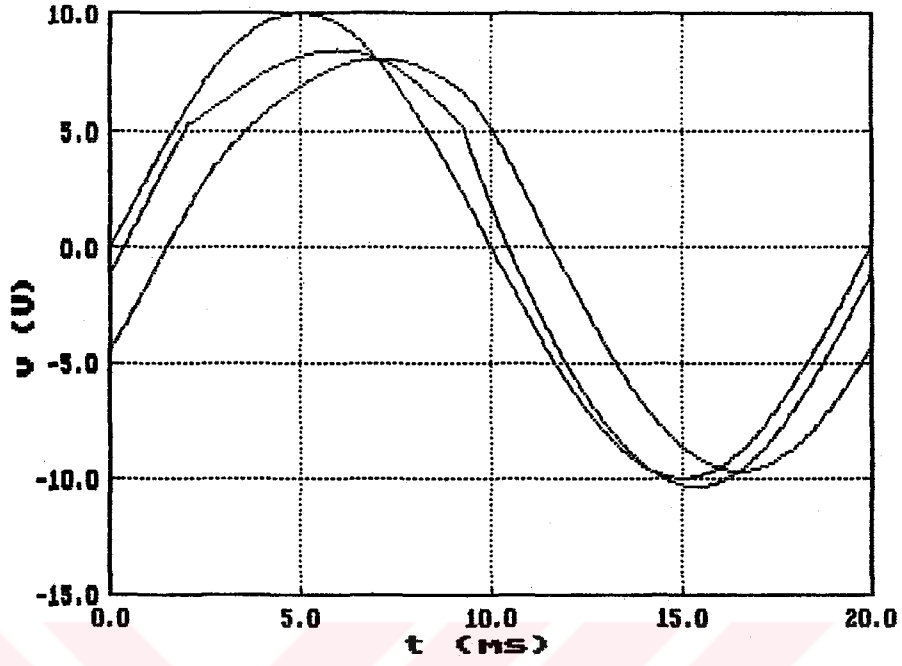
Şekil 4.25 İndüktans akımının değişimi

Aynı sistemde bütün datalar aynı kalmak üzere sadece gerilim kaynağının değeri değiştirilerek çözüm yapıldığında daha önce de bahsedildiği gibi sistemin girdiği durumlar ve geçiş zamanları değişmektedir. $v_i=10\sin\omega_1 t$ için yapılan çözüm sonucunda girilen sistem durumları ve geçiş zamanları,

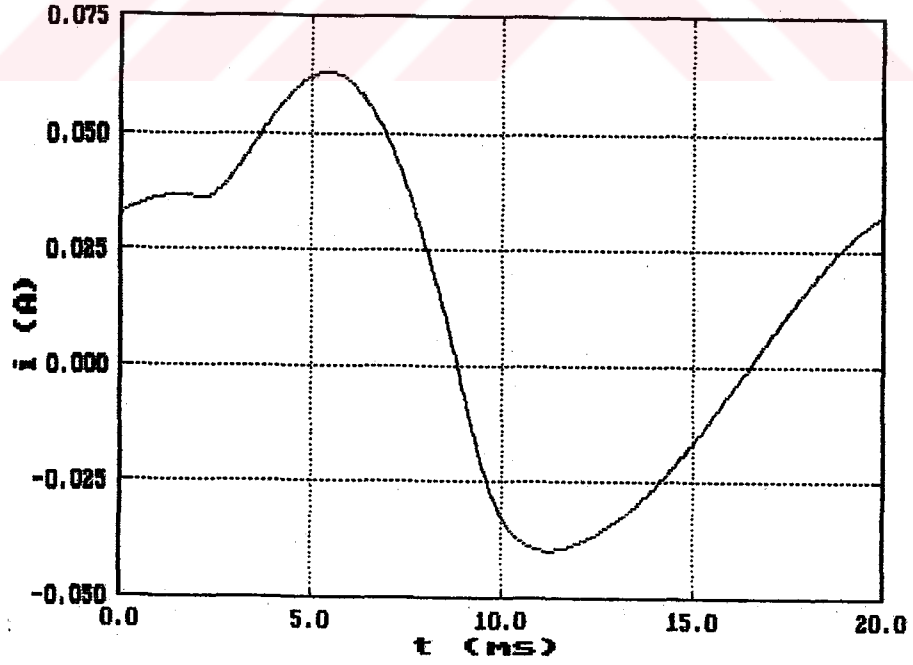
Sistem durumu	Yaklaşık geçiş zamanı
1	$\eta_1=0.0$
3	$\eta_2=0.24E-2$
1	$\eta_3=0.92E-2$

şeklinde bulunmuştur.

Bu durumlara ve geçiş zamanlarına göre yapılan sürekli hal çözümü sonuçları gerçek geçiş zamanlarının değerleri de gösterilerek Şekil 4.26 ve 4.27' de verilmiştir.



Şekil 4.26 Giriş, kondansatör ve doğrusal olmayan eleman gerilimlerinin değişimi



Şekil 4.27 İndüktans akımının değişimi

Bu örnek deęişik bir modelleme ile tekrar çözümlenerek aynı sonuçlar elde edilmiştir. Yukarıda doğrusal olmayan eleman, üç adet diyot, kaynaklar ve dirençlerle modellenmişti. Dolayısıyla sistemde sanki üç ayrı eleman varmış gibi sekiz ayrı sistem durumu oluşmuştu. Yeni modellemede ise doğrudan doğrusal olmayan elemanın kendisi göz önüne alınmış ve karakteristiğindeki her doğrusal parça bir durum olarak alınmıştır. Böylece sistemde dört durumlu tek doğrusal olmayan eleman yer aldığı için sistemin girebileceği durum sayısı da dört olmuştur. Bu durumların eski modellemedeki hangi sistem durumlarına karşılık geldiğinin belirtilmesi ile yetinilecektir.

Yeni sistem durumu

Eski sistem durumu

1

4

2

3

3

1

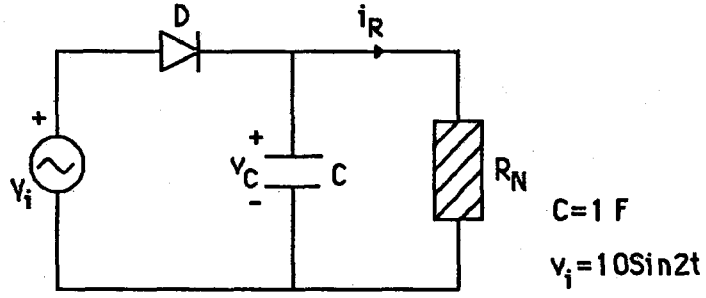
4

5

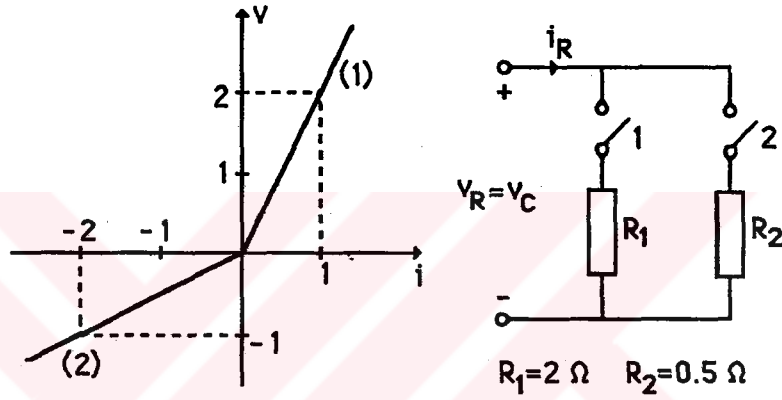
Girilen datalar ve kullanılan denklemler karşılık geldikleri durumları ile aynı ve sonuçlar da önceki sonuçlarla aynıdır. Bu nedenle aynı şeylerin tekrarı gereksiz görülmüştür.

4.5. Örnek 4

Bu örnekte Şekil 4.28' de verilen ve bir diyotla bir doğrusal olmayan direnç ihtiva eden devre analiz edilmiştir. Devrede yer alan doğrusal olmayan direncin karakteristiği ve eşdeğer devresi Şekil 4.29' da verilmiştir. Bu elemanın ve diyotun ikişer durumları olduğu için sistemin girebileceği durum sayısı dört olmaktadır. Bu durumlara ilişkin devreler ve denklemler aşağıda verilmiştir.



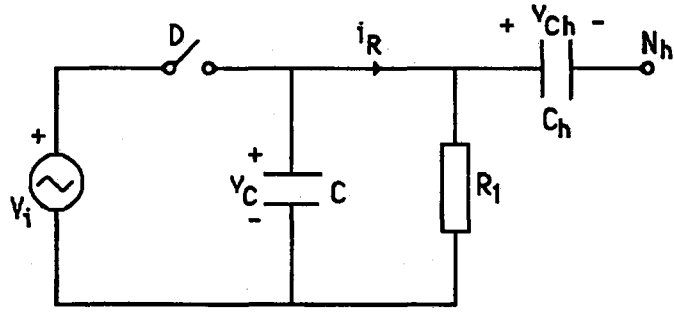
Şekil 4.28 Örnek 4' e ilişkin devre



Şekil 4.29 Doğrusal olmayan direncin karakteristiği ve eşdeğer devresi

Bilgisayar programı hazırlanırken sistemde en az bir durum değişkeninin olacağı kabul edilmiştir. Halbuki sistemde yer alan anahtarların pozisyonlarına göre bazı durumlarda durum değişkeni olmayabilir (örneğin bu örnekteki diyotun iletimde olduğu durumlar). Bu durumlarda programa yeni kontrol komutları ekleyerek programı karmaşıktırmamak için devreye hayali bir kapasitans ve hayali bir düğüm noktası eklenmiştir. Bu ekleme, devrenin yapısında herhangi bir değişikliğe neden olmamaktadır. Sadece bir durum değişkeni olarak gözüktüğü için devrede her zaman en az bir durum değişkeni olmasını garanti etmektedir.

1. Durum (D_1 açık, R_N 1. bölgede):



Şekil 4.30 Sistemin 1. durumu için eşdeğer devre

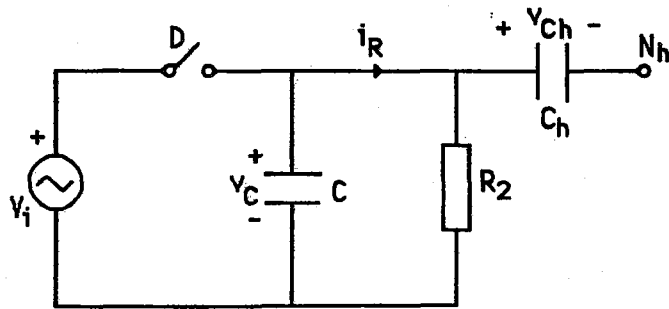
$$\begin{bmatrix} \dot{v}_{Ch} \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -1/R_1 C \end{bmatrix} \begin{bmatrix} v_{Ch} \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [v_i] \quad (4.35)$$

$$\begin{bmatrix} v_C \\ i_R \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1/R_1 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{Ch} \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} [v_i] \quad (4.36)$$

2. Durum (D_1 açık, R_N 2. bölgede):

$$\begin{bmatrix} \dot{v}_{Ch} \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -1/R_2 C \end{bmatrix} \begin{bmatrix} v_{Ch} \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [v_i] \quad (4.37)$$

$$\begin{bmatrix} v_C \\ i_R \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1/R_2 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{Ch} \\ v_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} [v_i] \quad (4.38)$$



Şekil 4.31 Sistemin 2. durumu için eşdeğer devre

3. Durum (D_1 kapalı, R_N 1. bölgede):

$$\begin{bmatrix} \dot{v}_{ch} \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} v_{ch} + \begin{bmatrix} 0 \end{bmatrix} v_i \quad (4.39)$$

$$\begin{bmatrix} v_C \\ i_R \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} v_{ch} + \begin{bmatrix} 1 \\ 1/R_1 \\ 0 \\ 1/R_1 \end{bmatrix} v_i + \begin{bmatrix} 0 \\ 0 \\ 0 \\ C \end{bmatrix} \dot{v}_i \quad (4.40)$$

4. Durum (D_1 kapalı, R_N 2. bölgede):

$$\begin{bmatrix} \dot{v}_{ch} \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} v_{ch} + \begin{bmatrix} 0 \end{bmatrix} v_i \quad (4.41)$$

$$\begin{bmatrix} v_C \\ i_R \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} v_{ch} + \begin{bmatrix} 1 \\ 1/R_2 \\ 0 \\ 1/R_2 \end{bmatrix} v_i + \begin{bmatrix} 0 \\ 0 \\ 0 \\ C \end{bmatrix} \dot{v}_i \quad (4.42)$$

Anahtarlama denklemlerinin katsayı matrisleri ve elemanlara ilişkin kontrol denklemleri aşağıdaki gibidir.

$$F_{1,3} = F_{1,4} = F_{2,3} = F_{2,4} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$F_{3,4} = F_{4,3} = \begin{bmatrix} 1 \end{bmatrix}$$

$$F_{1,2} = F_{2,1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$F_{3,1} = F_{3,2} = F_{4,1} = F_{4,2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$G_{1,3} = G_{1,4} = G_{2,3} = G_{2,4} = G_{3,4} = G_{4,3} = \begin{bmatrix} 0 \end{bmatrix}$$

$$G_{3,1} = G_{3,2} = G_{4,1} = G_{4,2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$G_{1,2} = G_{2,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

1. Eleman:

$$\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

2. Eleman:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$z_1 = z_2 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Yukarıdaki denklemlere ek olarak aşağıdaki datalar da bilgisayara girilmiştir.

$$CN=2$$

$$CSN=[2 \ 2]$$

$$SSN=4$$

$$NOV=4$$

$$NF=1$$

$$NDT=50$$

$$TT0=\pi$$

$$FRQ=[1/\pi]$$

$$NZV=[0 \ 0 \ 0 \ 0]$$

$$ZV=[0.0]$$

$$NSV=[2 \ 2 \ 1 \ 1]$$

$$NT=400$$

$$ICV=[3 \ 4]$$

$$DELY(3)=0.001$$

$$DELY(4)=0.001$$

$$YREF=[0 \ 0]$$

$$GDM_1 = GDM_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

$$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$NCE = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

$$SSCS = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 1 \\ 2 & 2 \end{bmatrix}$$

$$NSX = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Yukarda verilen denklemler ve datalar kullanılarak sistemin sürekli halde girdiği durumlar ve yaklaşık geçiş zamanları şu şekilde belirlenmiştir.

Sistem durumu

Yaklaşık geçiş zamanı

1

$$\eta_1=0.0$$

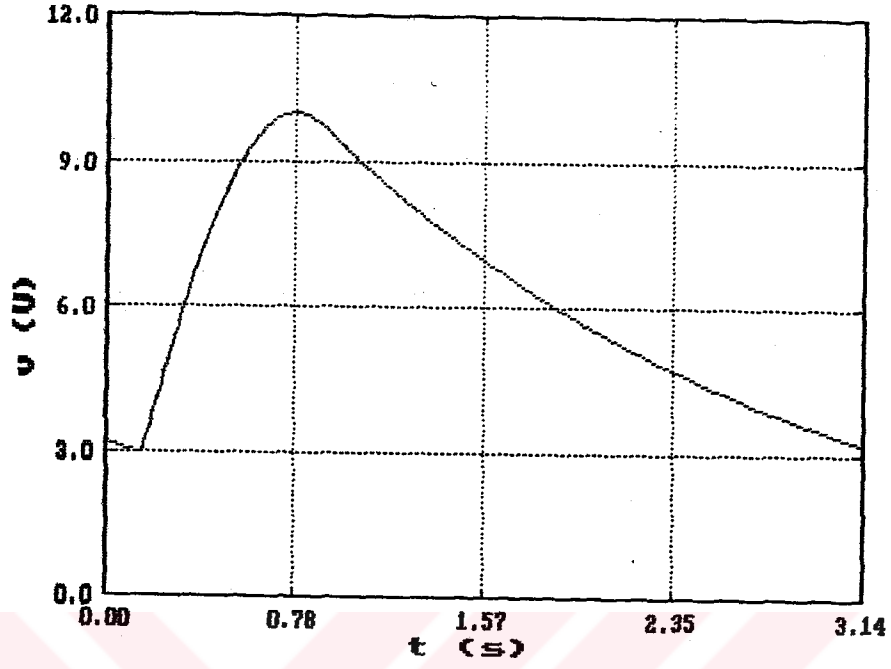
3

$$\eta_2=0.44$$

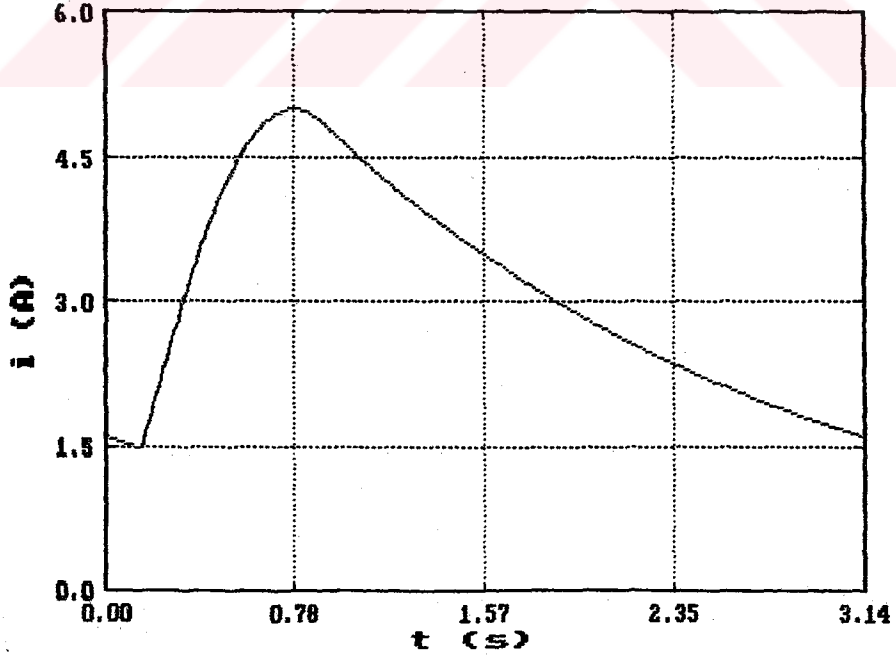
1

$$\eta_3=0.911$$

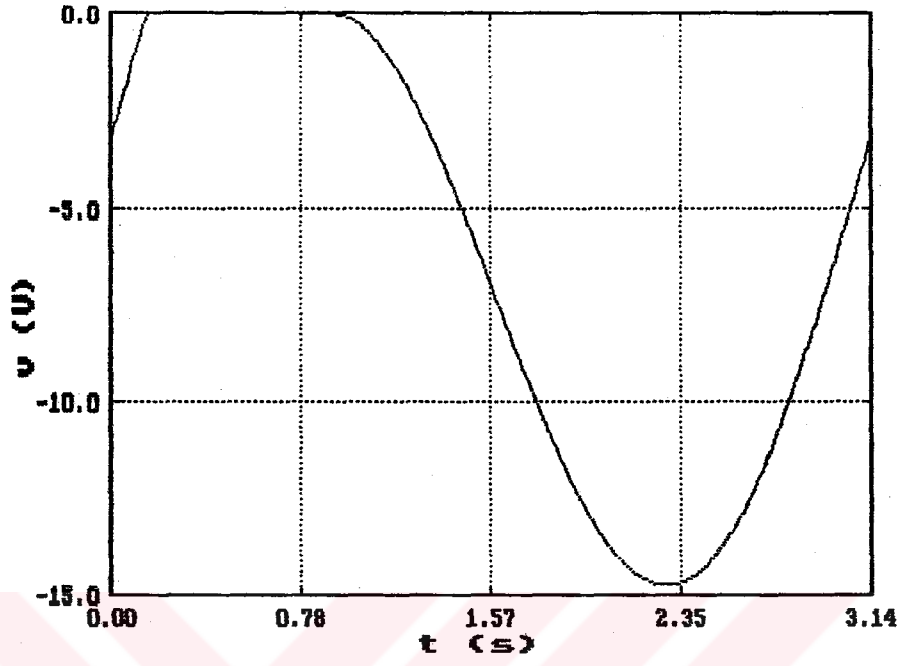
Bu durumlara ve yaklaşık geçiş zamanlarına göre programın ikinci kısmında yapılan sürekli hal çözümü sonuçları Şekil 4.32, 4.33, 4.34 ve 4.35' de verilmiştir.



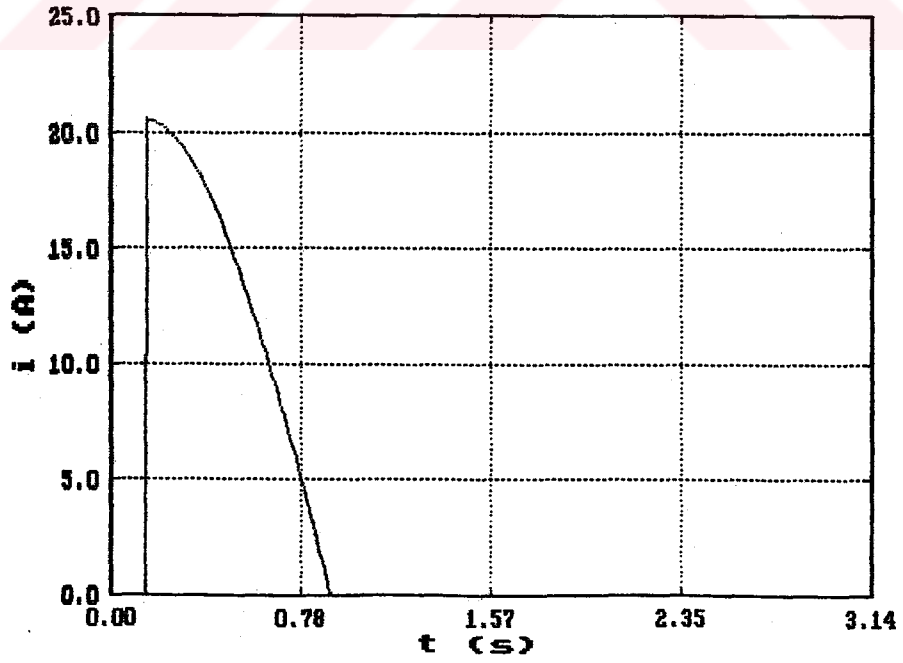
Şekil 4.32 Kondansatör geriliminin değişimi



Şekil 4.33 Doğrusal olmayan direnç akımının değişimi



Şekil 4.34 Diyot geriliminin değişimi



Şekil 4.35 Diyot akımının değişimi

4.6. Örnek 5

Bu örnekte Şekil 4.36' da verilen devrenin analizi yapılmıştır. Devrede, 30 derecede tetiklenen bir tristör, bir diyot ve doğrusal olmayan bir direnç yer almaktadır. Doğrusal olmayan direncin karakteristiği ve eşdeğer devresi Şekil 4.37' de verilmiştir. Şekilden de görüldüğü gibi bu elemanın iki durumu vardır. Tristör ve diyotun da ikiye durumları olduğu için sistemin toplam durum sayısı sekiz olmaktadır. Bu durumlara ilişkin eşdeğer devreler ve denklemler aşağıda verilmiştir.

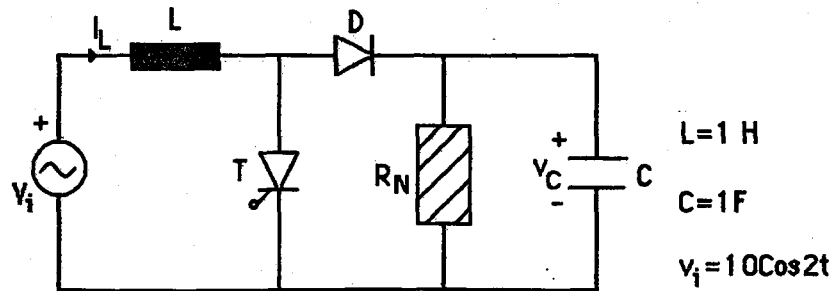
1.Durum (T ve D açık, R_N 1. bölgede):

$$[\dot{v}_C] = [-1/R_1C][v_C] + [0 \ 0] \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.43)$$

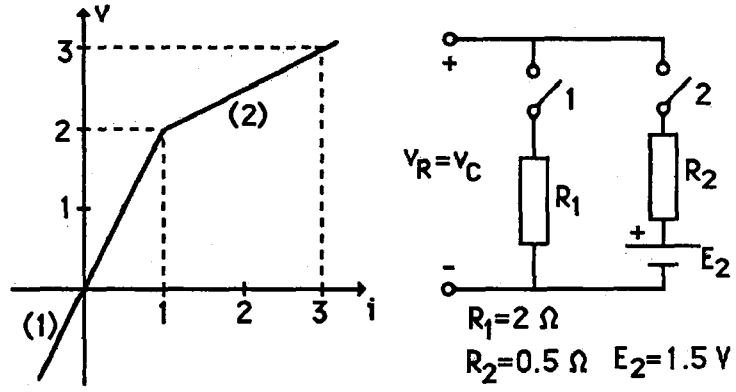
$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} [v_C] + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.44)$$

2.Durum (T ve D açık, R_N 2. bölgede):

$$[\dot{v}_C] = [-1/R_2C][v_C] + [0 \ E_d/R_2C] \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.45)$$



Şekil 4.36 Örnek 5'de göz önüne alınan devre

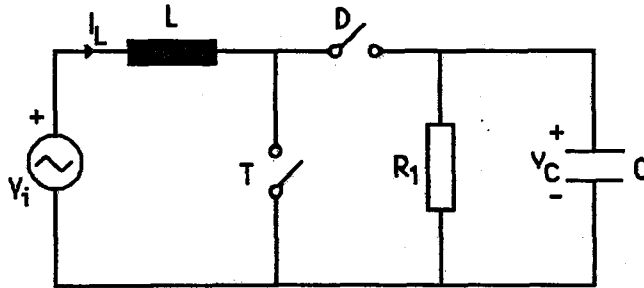


Şekil 4.37 Doğrusal olmayan direncin karakteristiği ve eşdeğer devresi

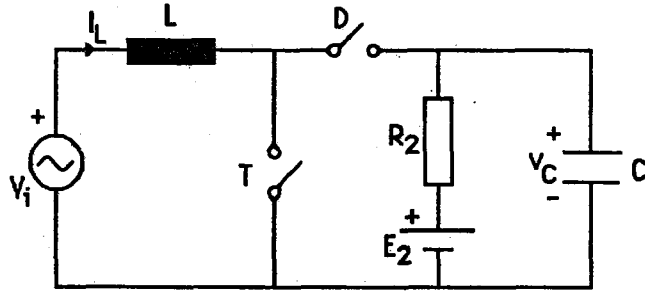
$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} [v_C] + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.46)$$

3. Durum (T açık, D kapalı ve R_N 1. bölgede):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/R_1 C \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 1/L & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.47)$$



Şekil 4.38 Sistemin 1. durumu için eşdeğer devre



Şekil 4.39 Sistemin 2. durumu için eşdeğer devre

$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.48)$$

4.Durum (T açık, D kapalı ve R_N 2. bölgede):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} 0 & -1/L \\ 1/C & -1/R_2C \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 1/L & 0 \\ 0 & E_2/R_2C \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.49)$$

$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.50)$$

5.Durum (D açık, T kapalı ve R_N 1. bölgede):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -1/R_1C \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 1/L & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.51)$$

$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.52)$$

6.Durum (D açık, T kapalı ve R_N 2. bölgede):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_C \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & -1/R_2C \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 1/L & 0 \\ 0 & E_2/R_2C \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.53)$$

$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.54)$$

7.Durum (T ve D kapalı ve R_N 1. bölgede):

$$\dot{i}_L = [0]i_L + [1/L \ 0] \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.55)$$

$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} i_L + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.56)$$

8.Durum (T ve D kapalı ve R_N 2. bölgede):

$$\dot{i}_L = [0]i_L + [1/L \ 0] \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.57)$$

$$\begin{bmatrix} v_C \\ i_L \\ v_T \\ i_T \\ v_D \\ i_D \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} i_L + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & E_2/R_2 \\ 0 & 0 \\ 0 & -E_2/R_2 \end{bmatrix} \begin{bmatrix} v_i \\ 1 \end{bmatrix} \quad (4.58)$$

Sistemin yukarıda belirtilen durumlarından birinden diğerine geçerken kullanılan anahtarlama denklemlerine ilişkin katsayı matrisleri aşağıda verilmiştir.

$$F_{1,2}=F_{2,1}=F_{7,8}=F_{8,7}=[1]$$

$$F_{1,7}=F_{1,8}=F_{2,7}=F_{2,8}=F_{7,1}=F_{7,2}=F_{8,1}=F_{8,2}=[0]$$

$$F_{3,1}=F_{3,2}=F_{4,1}=F_{4,2}=F_{5,1}=F_{5,2}=[0 \quad 1]$$

$$F_{3,7}=F_{3,8}=F_{4,7}=F_{4,8}=F_{5,7}=F_{5,8}=[1 \quad 0]$$

$$F_{7,3}=F_{7,4}=F_{7,5}=F_{7,6}=F_{8,3}=F_{8,4}=F_{8,5}=F_{8,6}=\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$F_{1,3}=F_{1,4}=F_{1,5}=F_{1,6}=F_{2,3}=F_{2,4}=F_{2,5}=F_{2,6}=\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$F_{3,4}=F_{3,5}=F_{3,6}=F_{4,3}=F_{4,5}=F_{4,6}=F_{5,3}=F_{5,4}=F_{5,6}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Bütün $G_{i,j}$ matrisleri $F_{i,j}$ ' lerle uygun boyutlarda sıfır matrislerdir. Bilgisayar programına bu değerler girilmediği zaman otomatik olarak sıfır seçildiği için burada yazılmaları gereksiz görülmüştür.

Sistemin yukarıda tanımlanan durumlardan hangisinde olduğunun belirlenmesinde aşağıdaki kontrol denklemlerinden yararlanılmıştır.

1. Eleman:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

2. Eleman:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} [y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

3. Eleman:

$$[-1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0][y] \geq [-2]$$

$$[1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0][y] \geq [2]$$

$$z_1 = z_2 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad z_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Durum ve çıkış denklemleri ile anahtarlama ve kontrol denklemlerine ilaveten aşağıdaki datalar da bilgisayara girilmiştir.

$$\begin{array}{lll} \text{CN}=3 & \text{CSN}=[2 \ 2 \ 2] & \text{SSN}=8 \\ \text{NOV}=9 & \text{NSV}=[1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 1 \ 1] & \text{NF}=2 \\ \text{TTO}=\pi & \text{FRQ}=[1/\pi \ 0.1\text{E}-10] & \text{NZV}=[0 \ 1 \ 0 \ 0 \ 0 \ 0] \\ \text{ZV}=[\pi/12] & \text{NDT}=100 & \text{NT}=400 \\ \text{ICV}=[4 \ 5] & \text{DELY}(4)=0.0001 & \text{DELY}(5)=0.0001 \\ \text{YREF}=[0.001 \ 0.01] & & \end{array}$$

$$\text{GDM}_1 = \text{GDM}_2 = \text{GDM}_3 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{NCE} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 1 & 1 \end{bmatrix}$$

$$\text{SSCS} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \end{bmatrix}$$

$$\text{NSX} = \begin{bmatrix} 2 & 0 \\ 2 & 0 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Yukarıda verilen denklemler ve datalara göre programın birinci kısmı, sistemin sürekli halde girdiği durumları ve bu durumlara yaklaşık geçiş zamanlarını ana periyodun 1/100' ü kadar bir hassasiyetle aşağıdaki gibi belirlemiştir:

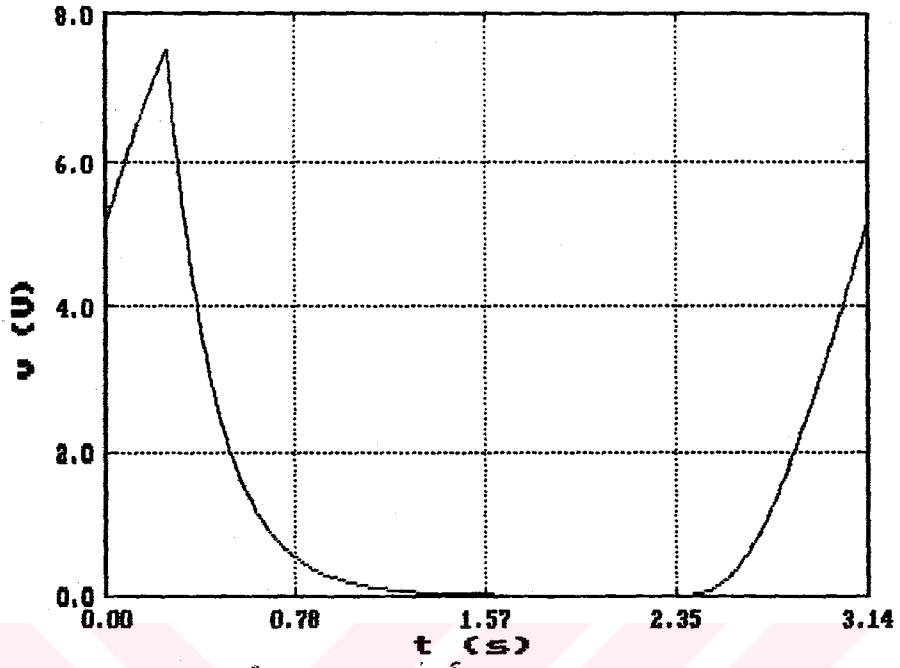
Sistem durumu	Yaklaşık geçiş zamanı
3	$\eta_1=0.0$
5	$\eta_2=0.252$
1	$\eta_3=0.22E+1$
3	$\eta_4=0.306E+1$

Programın ikinci kısmında yukarıda yaklaşık değerleri bulunan geçiş zamanlarının gerçek değerleri ana periyodun 1/10000 ' i kadar bir hassasiyetle belirlenmiştir. Bu değerler,

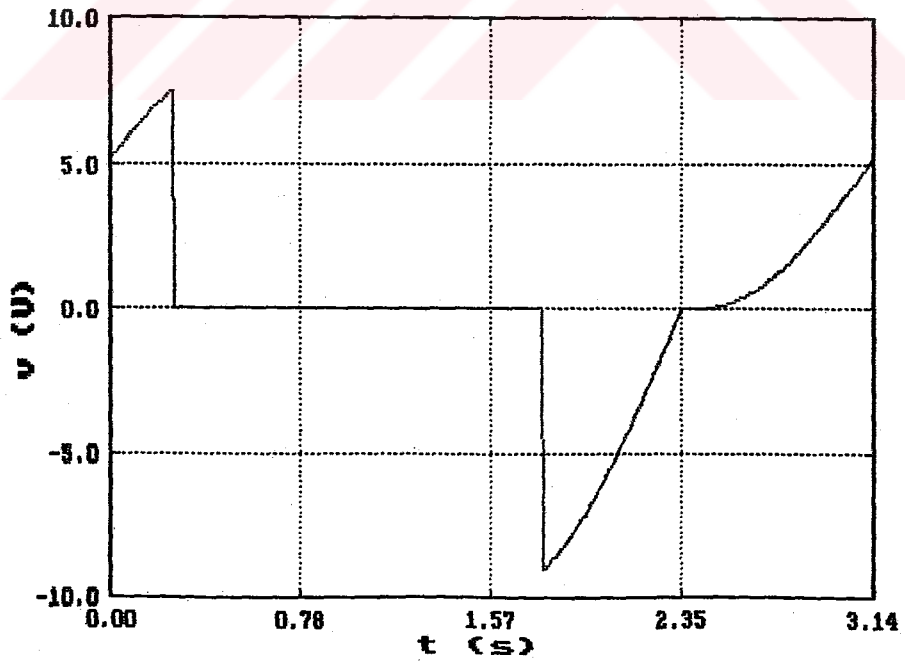
0.0 0.252 0.1783E+1 0.2357E+1

şeklinde dir. Burada özellikle dikkat edilmesi gereken nokta, geçiş zamanlarının yaklaşık değerleri ile gerçek değerleri arasındaki farktır. Ele alınan sistemin geçici hal davranışı çok uzun sürdüğü için programın birinci kısmının sonunda sistem girdiği durumlar bakımından sürekli hale gelmiş olmasına karşılık geçiş zamanları bakımından henüz sürekli hale gelmemiştir. Bu durum burada kullanılan yöntem için bir sorun teşkil etmemektedir. Çünkü programın ikinci kısmında geçiş zamanlarının gerçek değerleri çok kısa sürede belirlenmektedir. Yaklaşık zaman değerleri de periyodun 1/100' ü kadarlık bir çözüm adımı ile yaklaşık 15 periyotluk çözüm sonunda bulunmaktadır. Eğer aynı sistem sayısal integrasyon yöntemlerinden birisi ile çözümlenerek sürekli hal çözümü elde edilmek istenseydi, hem çok küçük çözüm adımı kullanılacaktı hem de sistem sürekli hale ulaşınca kadar buradakinin birkaç misli periyot boyunca çözüme devam edilecekti. Bunun için gereken bilgisayar zamanının buradaki zamanla kıyaslanamayacak kadar büyük olacağı açıkça ortadadır.

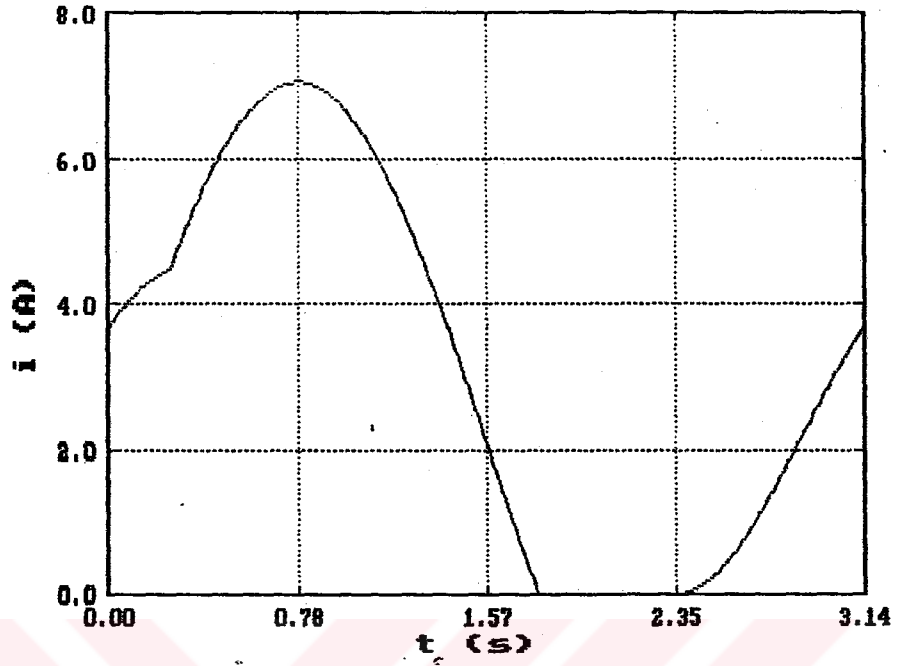
Yukarıdaki değerlere göre bir periyot boyunca 400 zaman değeri için sürekli hal çözümü hesaplanmıştır. Hesaplanan sonuçlara göre çizilen grafikler Şekil 4.40, 4.41, 4.42, 4.43 ve 4.44' te gösterilmiştir.



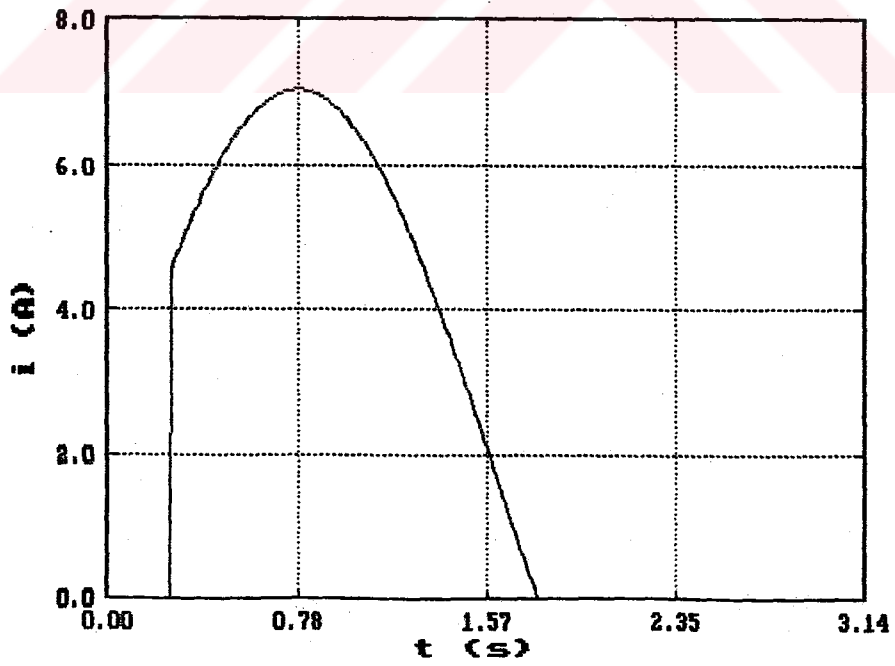
Şekil 4.40 Kondansatör geriliminin değişimi



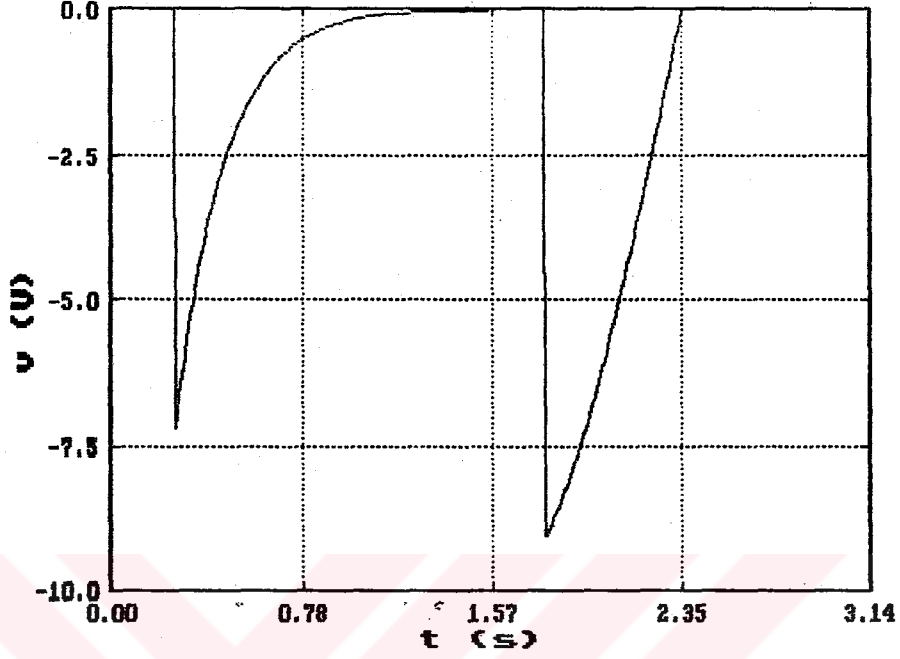
Şekil 4.41 Tristör geriliminin değişimi



Şekil 4.42 İndüktans akımının değişimi



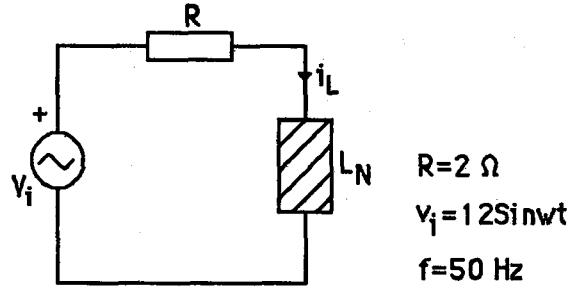
Şekil 4.43 Tristör akımının değişimi



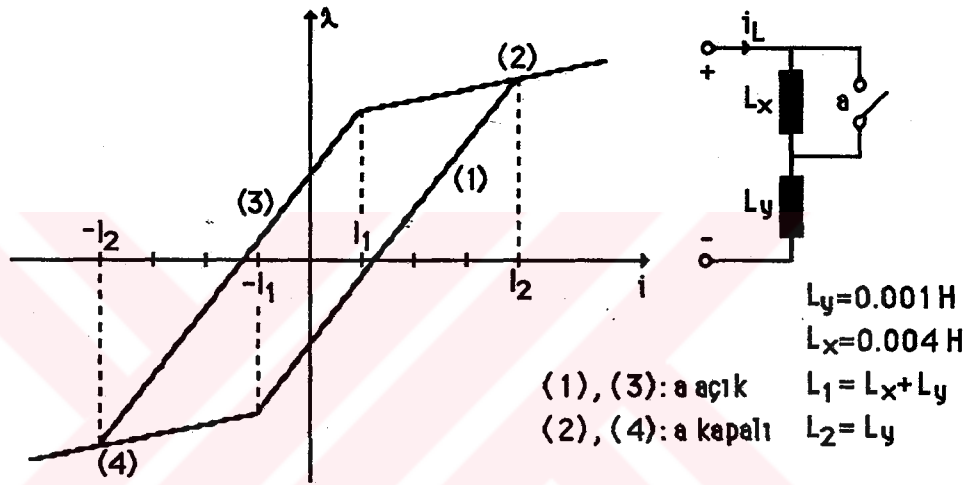
Şekil 4.44 Diyot geriliminin değişimi

4.7. Örnek 6

Buraya kadar verilen örneklerde dışardan kontrollü veya karakteristiği doğrusal olmayan elemanlar içeren devreler ele alınmıştır. Bu örnekte ise karakteristiklerinde histerisiz bulunan elemanlara örnek olmak üzere doğrusal olmayan ve histerisiz bulunan bir indüktansın yer aldığı elektrik devresi analiz edilmiştir. Şekil 4.45' te verilen bu devre, elde edilen çözüm sonuçlarında histerisiz etkisinin başka detaylarla karışmadan açıkça görülebilmesi için özellikle basit olarak seçilmiştir. Devrede yer alan doğrusal olmayan indüktansın karakteristiği ve eşdeğer devresi Şekil 4.46' da verilmiştir.



Şekil 4.45 Örnek 6' ya ilişkin devre



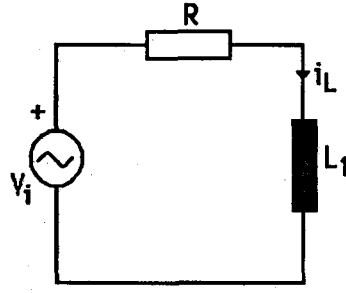
Şekil 4.46 Doğrusal olmayan indüktansın karakteristiği ve eşdeğer devresi

Devredeki doğrusal olmayan tek eleman indüktans olduğu için bunun durumları doğrudan sistemin durumları olmaktadır. Bu durumlara ilişkin denklemler ve datalar aşağıda verilmiştir.

1. Durum (L_N 1. bölgede):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} -R/L_1 \\ 1/L_1 \end{bmatrix} \begin{bmatrix} i_L \\ v_i \end{bmatrix} \quad (4.59)$$

$$\begin{bmatrix} i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} i_L + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v_i \quad (4.60)$$



Şekil 4.47 Sistemin 1. durumu için eşdeğer devre

2. Durum (L_N 2. bölgede):

$$\begin{bmatrix} i_L \end{bmatrix} = \begin{bmatrix} -R/L_2 \end{bmatrix} \begin{bmatrix} i_L \end{bmatrix} + \begin{bmatrix} 1/L_2 \end{bmatrix} \begin{bmatrix} v_i \end{bmatrix} \quad (4.61)$$

$$\begin{bmatrix} i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} i_L \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} v_i \end{bmatrix} \quad (4.62)$$

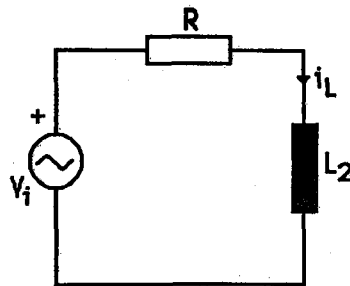
3. Durum (L_N 3. bölgede):

$$\begin{bmatrix} i_L \end{bmatrix} = \begin{bmatrix} -R/L_1 \end{bmatrix} \begin{bmatrix} i_L \end{bmatrix} + \begin{bmatrix} 1/L_1 \end{bmatrix} \begin{bmatrix} v_i \end{bmatrix} \quad (4.63)$$

$$\begin{bmatrix} i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} i_L \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} v_i \end{bmatrix} \quad (4.64)$$

4. Durum (L_N 4. bölgede):

$$\begin{bmatrix} i_L \end{bmatrix} = \begin{bmatrix} -R/L_2 \end{bmatrix} \begin{bmatrix} i_L \end{bmatrix} + \begin{bmatrix} 1/L_2 \end{bmatrix} \begin{bmatrix} v_i \end{bmatrix} \quad (4.65)$$



Şekil 4.48 Sistemin 2. durumu için eşdeğer devre

$$\begin{bmatrix} i_L \\ v_i \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} [i_L] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [v_i] \quad (4.66)$$

Sistem yukarıda belirtilen durumlardan birinden diğerine geçerken kullanılan anahtarlama denklemlerinin katsayı matrisleri, sistemde süreksizlik olmadığı için bütün $F_{i,j}$ ' ler 1 ve bütün $G_{i,j}$ ' ler 0 şeklindedir.

Sistemin durumlarının belirlenmesinde aşağıdaki kontrol denklemlerinden yararlanılmıştır.

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} -l_1 \\ -l_2 \end{bmatrix}$$

$$[1 \ 0] [y] \geq [l_1]$$

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} [y] \geq \begin{bmatrix} -l_2 \\ -l_1 \end{bmatrix}$$

$$[-1 \ 0] [y] \geq [l_1]$$

$$z_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Bu örnek için yapılan tanımlamalar ve daha önce 3. bölümde yapılmış olan açıklamalar ışığında, durum ve çıkış denklemleri ile anahtarlama ve kontrol denklemlerine ilaveten aşağıdaki datalar da bilgisayara girilmiştir.

$$CN=1$$

$$CSN=[4]$$

$$SSN=4$$

$$NSV=[1 \ 1 \ 1 \ 1]$$

$$NOV=2$$

$$NF=1$$

$$TT0=0.02$$

$$FRQ=[50]$$

$$NZV=[0 \ 0 \ 0 \ 0]$$

$$ZV=[0.0]$$

$$NT=400$$

$$NCE=[2 \ 1 \ 2 \ 1]$$

$$NDT=100$$

$$x0=[0]$$

$$ICV=[1 \ 1 \ 1]$$

$$DELY(1)=0.01$$

$$YREF=[4 \ 1 \ -4]$$

$$NSX = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$SSCS = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$GDM = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Programın birinci kısmında, ana periyodun 1/100 ' ü kadar bir hassasiyetle yapılan çözümle sistemin bir periyot boyunca girdiği durumlar ve bu durumlara geçiş zamanlarının yaklaşık değerleri şu şekilde belirlenmiştir:

Sistem durumu	Yaklaşık geçiş zamanı
1	$\eta_1=0.0$
2	$\eta_2=0.52E-2$
3	$\eta_3=0.1E-1$
4	$\eta_4=0.152E-1$

Programın ikinci kısmında ilk olarak bu yaklaşık geçiş zamanlarının gerçek değerleri ana periyodun 1/10000 ' i kadar bir hassasiyetle belirlenerek,

0.0 0.5028E-2 0.9957E-2 0.1503E-1

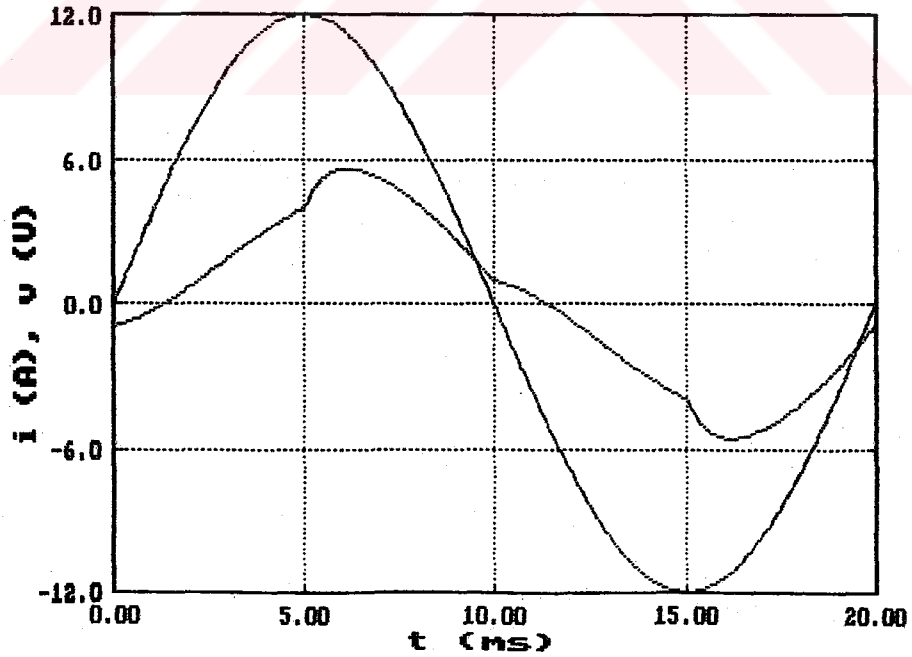
şeklinde bulunmuştur. Daha sonra bu değerlere göre sistemin sürekli hal çözümü yapılmış ve elde edilen çözüm sonuçlarına göre çizilen grafikler Şekil 4.49' da verilmiştir.

Şekil 4.49' da verilen indüktans akımı incelendiğinde histeresiz etkisi açıkça görülmektedir. İndüktans, η_2 anına kadar karakteristiğindeki 1. bölgede olduğu için eşdeğeri büyük bir indüktans değerine karşılık gelmektedir ve bu nedenle içerisinde geçen akımın eğimi küçüktür. η_2 anında 1. bölgeden 2. bölgeye geçtiği için eşdeğer indüktans değeri küçülmektedir. Bu anda akımın değeri ani olarak değişmeyeceği için direncin uçlarındaki gerilim de ani olarak değişmeyecektir. Dolayısıyla indüktans kendi gerilimini o an için aynı değerde tutmaya çalışacaktır. Bunun için de akımının eğimini artırmaktadır. η_3 anında ise bu olayın tersi gerçekleştiği için yani küçük indüktans değerinden büyük indüktans değerine geçildiği için akımın eğimi tekrar azalmaktadır. Burada dikkat edilmesi gereken nokta, 2. bölgeye girildiği andaki akım değeri ile 2. bölgeden çıkıldığı andaki akım değerleri birbirinden farklıdır. Aynı şey diğer bölgelerde de görülmektedir.

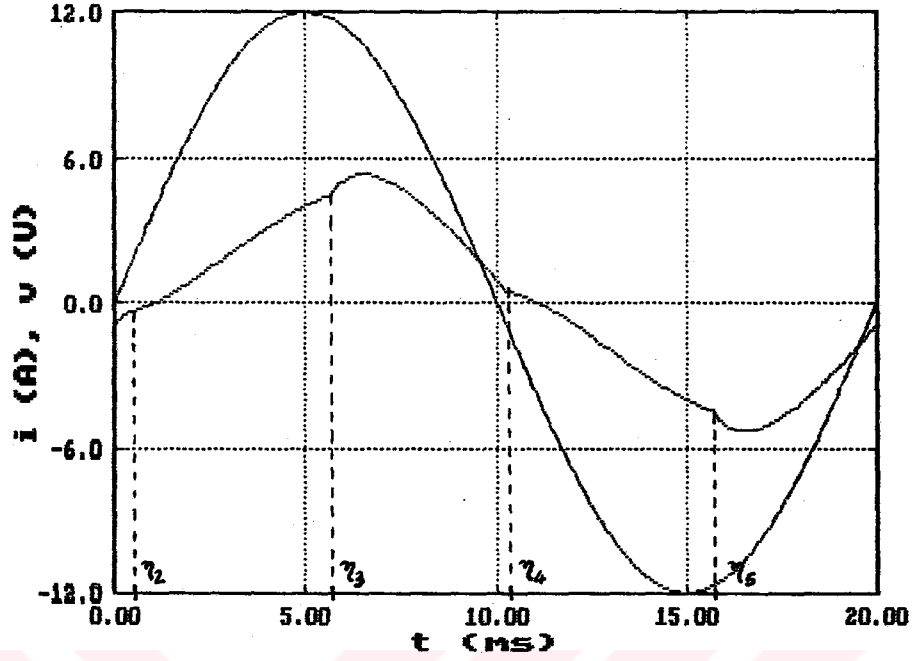
Şekil 4.45 ile verilen devre için aynı gösterimler geçerli olmak üzere değişik bir karakteristiğe karşılık gelecek şekilde, sadece Şekil 4.46' daki I_1 ve I_2 değerleri, sırayla 0.5 A ve 4.5 A alınarak çözüm işlemleri tekrarlanmıştır. Bu çözüm sonucunda sistem durumları ve yaklaşık geçiş zamanları,

Sistem durumu	Yaklaşık geçiş zamanı
4	$\eta_1=0.0$
1	$\eta_2=0.8E-3$
2	$\eta_3=0.56E-2$
3	$\eta_4=0.104E-1$
4	$\eta_4=0.156E-1$

şeklinde bulunmuştur. Böylece elde edilen sürekli hal çözümleri için çizilen grafikler gerçek geçiş zamanları (η' lar) da gösterilerek Şekil 4.50' de verilmiştir.



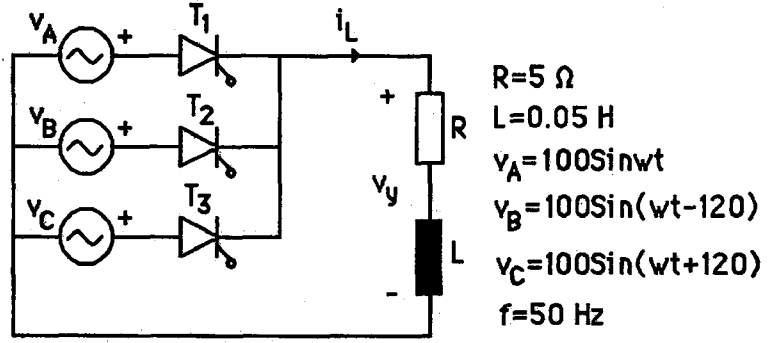
Şekil 4.49 Giriş gerilimi ve indüktans akımının değişimleri



Şekil 4.50 Giriş gerilimi ve indüktans akımlarının değişimleri

4.8. Örnek 7

Bu örnekte üç fazlı sistemlere örnek olması için Şekil 4.51' de verilen doğrultucu devresi analiz edilmiştir. Devrede yer alan tristörlerin birer uçları ortak olduğu için herbirisini bir eleman olarak almak yerine üçü tek eleman gibi alınabilir. Bu eleman, tristörlerin hepsinin açık olması ve teker teker kapalı olmaları durumlarına göre dört duruma sahip olur. Sistemdeki tek eleman bu olduğu için sistemin girebileceği durum sayısı da dört olmaktadır. Bu durumlara ilişkin eşdeğer devreler ve denklemler aşağıda verilmiştir.

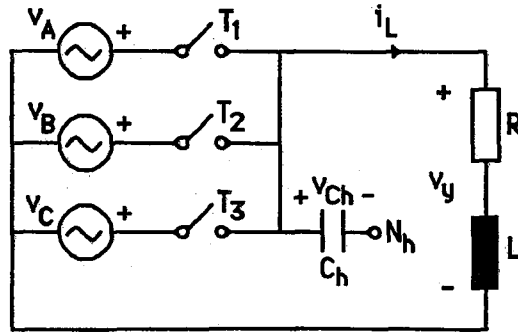


Şekil 4.51 Örnek 7' ye ilişkin devre

Bilgisayar programı hazırlanırken sistemde en az bir durum değişkeninin olacağı kabul edildiği için 4. örnekte olduğu gibi devreye hayali bir kapasitans ve hayali bir düğüm noktası eklenmiştir. Bu ekleme, devrenin yapısında herhangi bir değişikliğe neden olmamaktadır. Sadece bir durum değişkeni olarak gözüktüğü için devrede her zaman en az bir durum değişkeni olmasını garanti etmektedir.

1. Durum (T_1 , T_2 ve T_3 açık):

$$\begin{bmatrix} \dot{v}_a \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} v_a \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.67)$$



Şekil 4.52 Sistemin 1. durumu için eşdeğer devre

$$\begin{bmatrix} i_L \\ v_y \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [v_{ch}] + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.68)$$

2. Durum (T_2 ve T_3 açık, T_1 kapalı):

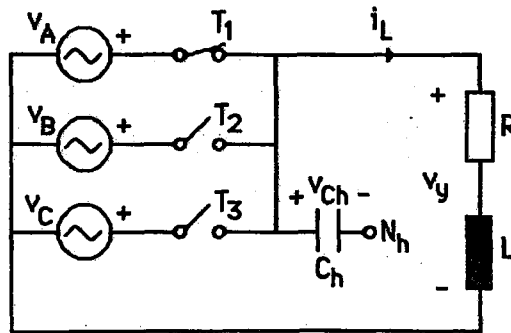
$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_{ch} \end{bmatrix} = \begin{bmatrix} -R/L & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_{ch} \end{bmatrix} + \begin{bmatrix} 1/L & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.69)$$

$$\begin{bmatrix} i_L \\ v_y \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_{ch} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.70)$$

3. Durum (T_1 ve T_3 açık, T_2 kapalı):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_{ch} \end{bmatrix} = \begin{bmatrix} -R/L & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_{ch} \end{bmatrix} + \begin{bmatrix} 0 & 1/L & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.71)$$

$$\begin{bmatrix} i_L \\ v_y \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_{ch} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.72)$$



Şekil 4.53 Sistemin 2. durumu için eşdeğer devre

4. Durum (T_1 ve T_2 açık, T_3 kapalı):

$$\begin{bmatrix} \dot{i}_L \\ \dot{v}_{Ch} \end{bmatrix} = \begin{bmatrix} -R/L & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_{Ch} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1/L \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.73)$$

$$\begin{bmatrix} i_L \\ v_y \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ v_{Ch} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_C \end{bmatrix} \quad (4.74)$$

Anahtarlama denklemlerinin katsayı matrisleri aşağıdaki gibidir.

$$F_{2,1}=F_{3,1}=F_{4,1}=[0 \quad 1]$$

$$F_{2,3}=F_{2,4}=F_{3,2}=F_{3,4}=F_{4,2}=F_{4,3}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad F_{1,2}=F_{1,3}=F_{1,4}=\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$G_{2,1}=G_{3,1}=G_{4,1}=[0 \quad 0 \quad 0]$$

$$G_{1,2}=G_{1,3}=G_{1,4}=G_{2,3}=G_{2,4}=G_{3,2}=G_{3,4}=G_{4,2}=G_{4,3}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Durumların belirlenmesinde aşağıda verilen kontrol denklemlerinden yararlanılmıştır.

$$[-1 \quad 0 \quad 0 \quad 0 \quad 0][y] \geq [0]$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}[y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}[y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}[y] \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$z_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}$$

Yukardaki denklemlere ek olarak aşağıdaki datalar da bilgisayara girilmiştir.

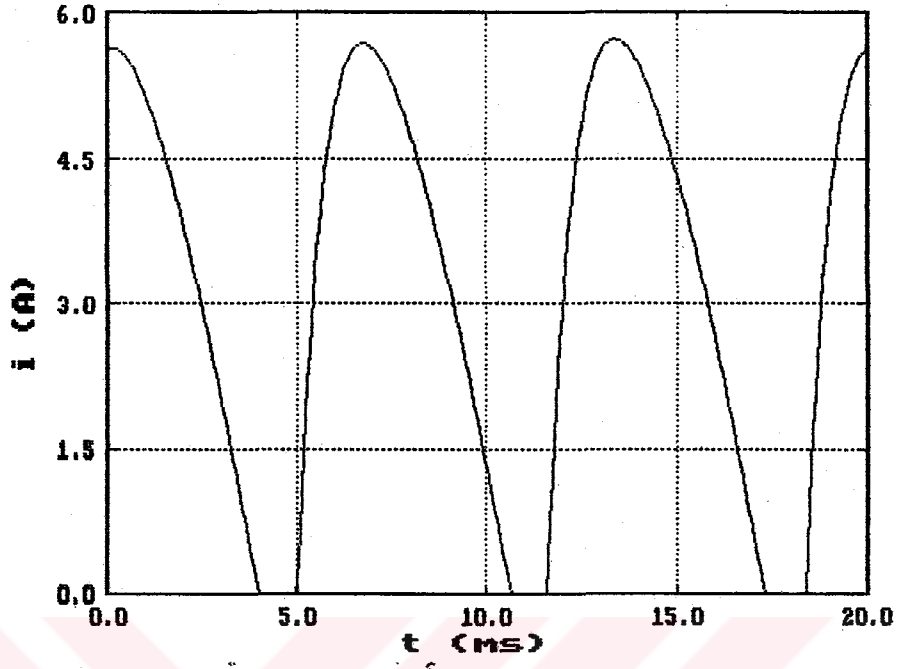
$$\begin{array}{lll}
 \text{CN}=1 & \text{CSN}=[4] & \text{SSN}=4 \\
 \text{NOV}=5 & \text{NF}=3 & \text{NDT}=100 \\
 \text{TT0}=0.02 & \text{FRQ}=[50 \ 50 \ 50] & \text{NZV}=[0 \ 1 \ 1 \ 1] \\
 \text{NCE}=[1 \ 2 \ 2 \ 2] & \text{NSV}=[1 \ 2 \ 2 \ 2] & \text{NT}=400 \\
 \text{ICV}=[1 \ 1 \ 1] & \text{DELY}(1)=0.0001 & \text{YREF}=[0 \ 0 \ 0] \\
 \text{ZV}=[0.005 \ 0.011666 \ 0.018332] & &
 \end{array}$$

$$x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{SSCS} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \quad \text{NSX} = \begin{bmatrix} 2 & 0 \\ 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{bmatrix}$$

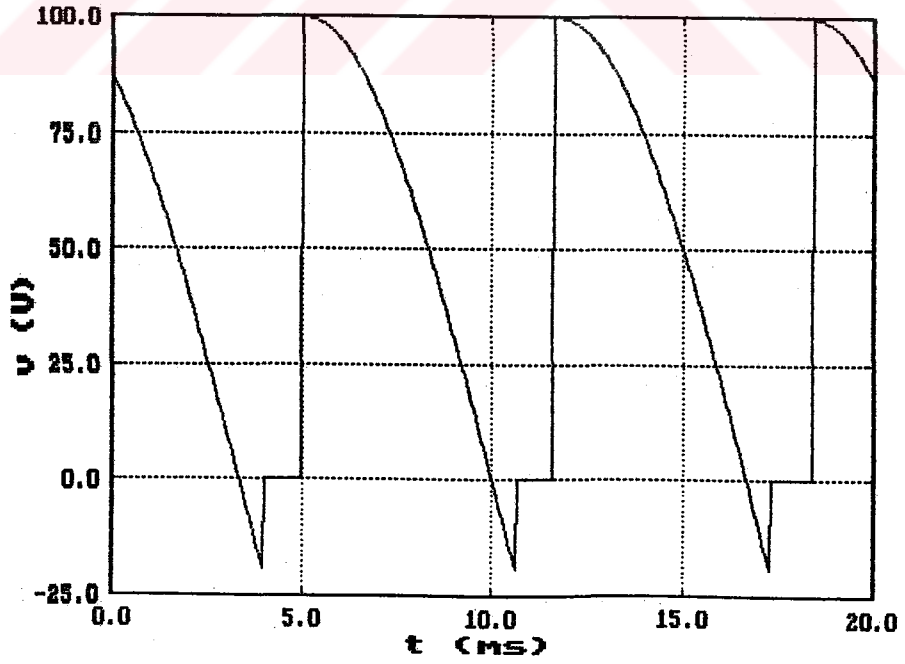
Yukarda verilen denklemler ve datalar kullanılarak sistemin sürekli halde girdiği durumlar ve yaklaşık geçiş zamanları aşağıdaki gibi belirlenmiştir. Belirlenen değerlere dikkat edilirse tristörlerin teteklenme zamanları ile aynı olduğu görülür. Çünkü bütün geçişler dışardan müdahale ile olmaktadır ve bunların zamanı da kullanıcı tarafından belirlenmektedir. Dolayısıyla bu zamanlar aynı zamanda gerçek geçiş zamanları olmaktadır.

Sistem durumu	Yaklaşık geçiş zamanı
4	$\eta_1=0.0$
2	$\eta_2=0.005$
3	$\eta_3=0.011666$
4	$\eta_4=0.018332$

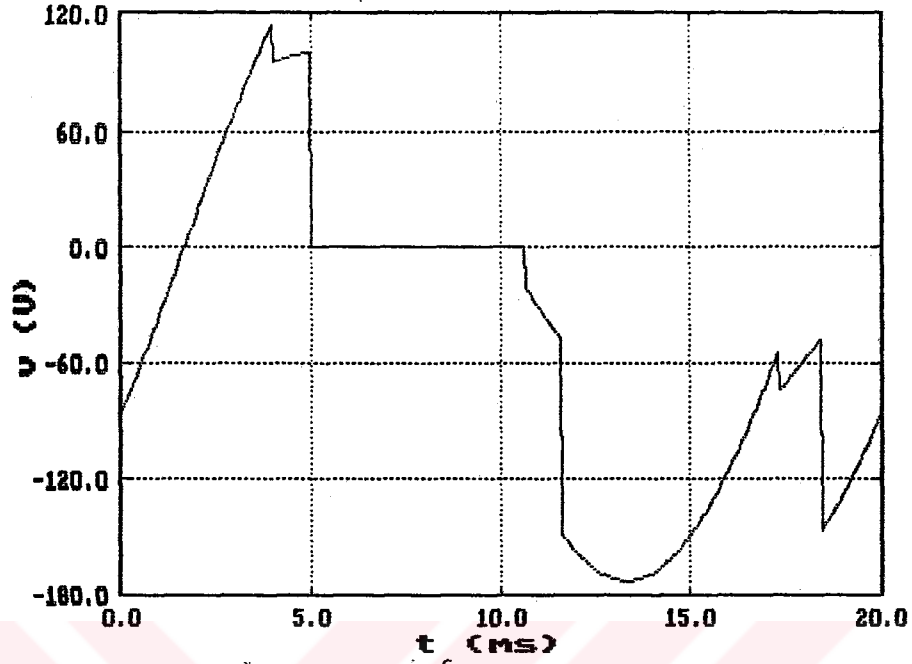
Bu durumlar ve yaklaşık geçiş zamanları için programın ikinci kısmında yapılan sürekli hal çözümü sonuçlarına göre çizilen grafikler Şekil 4.54, 4.55 ve 4.56' da verilmiştir. Tristörlerin gerilimleri, prensip olarak birbirine benzediği için sadece T_1 ' in geriliminin değişimi verilmiştir.



Şekil 4.54 İndüktans (yük) akımının değişimi



Şekil 4.55 Yük uçlarındaki gerilimin değişimi



Şekil 4.56 T₁ tristörünün uçlarındaki gerilimin değişimi

Aynı örnek, $R=15 \Omega$ ve $L=0.01$ H için tekrar çözümlenerek birinci kısımda

Sistem durumu

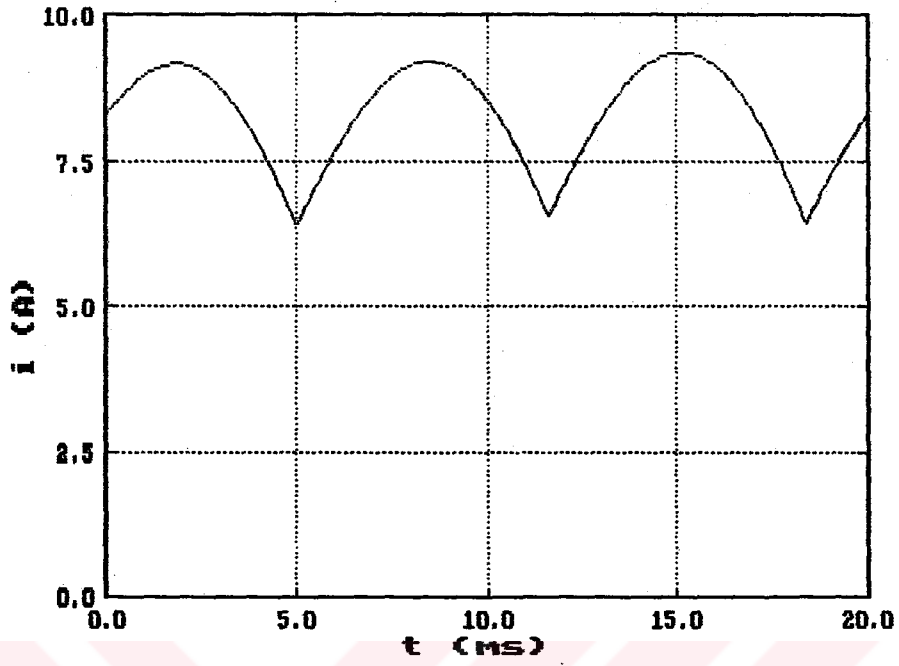
Yaklaşık geçiş zamanı

4	$\eta_1=0.0$
1	$\eta_2=0.4E-2$
2	$\eta_3=0.5E-2$
1	$\eta_4=0.108E-1$
3	$\eta_5=0.116E-1$
1	$\eta_6=0.174E-1$
4	$\eta_7=0.184E-1$

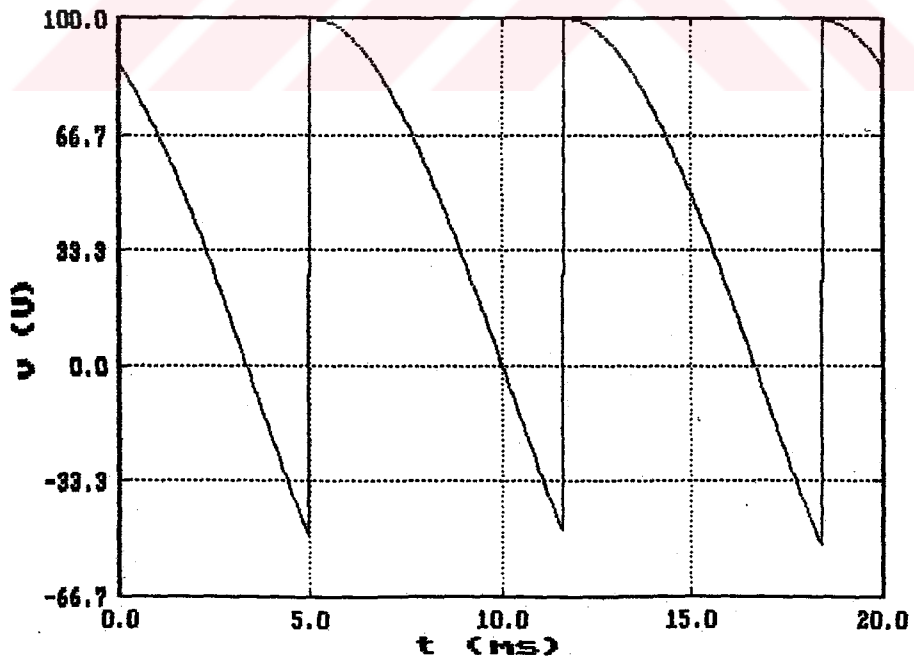
yaklaşık değerleri ve ikinci kısımda da bu geçiş zamanlarının gerçek değerleri,

0.0 0.3989E-2 0.5E-2 0.106E-1 0.116E-1 0.173E-1 0.184E-1

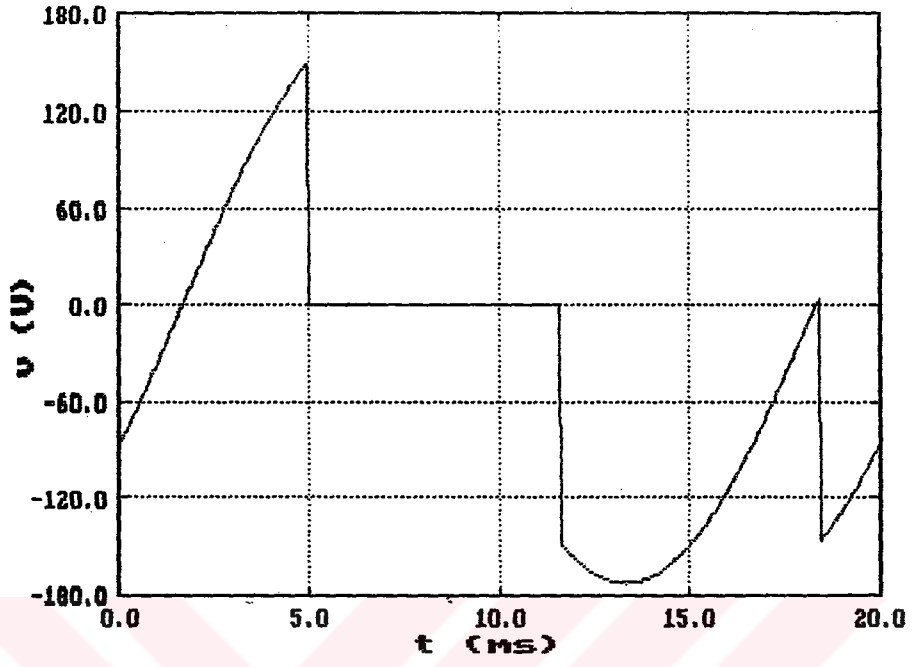
şeklinde belirlenmiştir. Bu değerlere ve yukarıdaki datalara göre yapılan sürekli hal çözümünün sonuçları Şekil 4.57, 4.58 ve 4.59' da verilmiştir.



Şekil 4.57 İndüktans (yük) akımının değişimi



Şekil 4.58 Yük uçlarındaki gerilimin değişimi



Şekil 4.59 T_1 tristörünün uçlarındaki gerilimin değişimi

5. SONUÇLAR

Bu tezde, parça parça doğrusal ve dışardan kontrollü elemanlar içeren sistemlerin sürekli hal çözümlerini elde etmek için genel bir metod verilmiştir. Bu metod için kullanılan algoritmanın tamamı bilgisayar programı olarak gerçekleştirilmiş ve çok sayıda örnek üzerinde test edilerek tatmin edici sonuçlar elde edilmiştir.

Verilen metodla parça parça doğrusallık yaklaşımı yapılabilecek karakteristiğe sahip olan elemanlar içeren herhangi bir devrenin sürekli hal analizi yapılabilmektedir. Metod, bilgisayarda çok kısa sürede doğru sonuçlar vermektedir. Çünkü, doğrusal olmayan devrelerin yerine bunların eşdeğerleri olacak şekilde periyodik olarak çalışan anahtarlar içeren devreler kullanılmakta ve bu devrelerin çözümleri de kesin çözüm sonuçlarını veren formüller yardımıyla yapılmaktadır. Dolayısıyla klasik metodlarda olduğu gibi Gradient vektörlerinin ve Jacobian matrisinin hesaplanmasını gerektiren optimizasyon algoritmalarının kullanılmasına veya nümerik integrasyonda olduğu gibi uzun zaman harcayan küçük adımlı çözümlere gerek kalmamaktadır. Bu metodun avantajları ve çekiciliği buradadır.

Parça parça doğrusallığın yanısıra periyodik olarak çalışan anahtarlardan da bahsedilince bu metodun özellikle güç elektroniği devrelerinin sürekli hal analizi için çok uygun olduğu açıkça görülebilir. Çünkü, bu devrelerde yer alan elemanlar parça parça doğrusallık yaklaşımına uygun olan karakteristiklere sahiptirler ve genellikle de periyodik şartlar altında çalışmaktadırlar.

Sürekli hal davranışları bilinmeyen sistemlerin analizi veya bir sistemin değişik parametreler için analizi bu metodla kolayca yapılabilmektedir. Çünkü kullanılan metodda önce, sistemin sürekli hal davranışı sistemin tam (geçici hal +

sürekli hal) çözümü doğrudan kesin sonuç veren formüller yardımıyla elde edilerek belirlenmekte ve daha sonra yine doğrudan kesin sonuç veren formüller yardımıyla sürekli hal çözümü bulunmaktadır.

Birden fazla girişe (kaynağa) sahip olan veya durum ve çıkış denklemlerinde kaynak fonksiyonlarının türevleri yer alan sistemlerin birden fazla çıkış büyüklüğü için çözümü yapılabilmektedir. Ayrıca periyodik olarak çalışan anahtarların pozisyonlarına göre oluşan sistem durumlarından herbirinde farklı sayıda durum değişkeni olduğu hallerde bu sayıya uygun boyutta katsayı matrisleri kullanılarak bilgisayar hafızasının verimli kullanılması sağlanmıştır.

Analizi yapılan sistemde karakteristiğinde histerisiz bulunan elemanların yer alması durumunda yapılan yeni tanımlama ve düzenlemeler yardımıyla bu durum çözüme yansıtılabilmektedir. Bunun bir uygulaması Örnek 6' da verilmiştir.

Yukarda söz edilen bu avantajlara rağmen bütün analiz tekniklerinde olduğu gibi bu metodun da bazı sınırlayıcı yanları vardır. Bunlardan birincisi, analizi yapılan sistemin periyodik girişe sahip olduğu ve bütün çıkışların sürekli hal cevaplarının periyodik olduğunun kabul ediliyor olmasıdır. Ama bu kabul metodun temelini oluşturduğu için vazgeçilemez bir sınırlamadır. İkinci olarak hazırlanan programın çok fazla giriş bilgisine gerek duymasıdır. Aslında bunun nedeni, bu tez çalışmasına paralel olarak yürütülen diğer bir çalışmada hazırlanan programdan elde edilecek olan formülasyon ile ilgili bilgileri kullanacak şekilde bu programın hazırlanmış olmasıdır.

Doğrusal olmayan sistemlerin yerine kullanılan ve periyodik olarak çalışan anahtarlar içeren doğrusal eşdeğer devrelerde çok sayıda anahtar bulunması halinde sistemin girebileceği durumların sayısı çok fazla artmaktadır. Dolayısıyla bu da çok sayıda durum, çıkış ve anahtarlama denklemi elde edilmesi anlamına gelmektedir. Bu denklemlerin yukarda bahsedildiği gibi başka bir program

yardımıyla elde edileceği göz önüne alınırsa, bu durum mevcut çalışmanın kapsamı içerisinde sadece bilgisayar zamanı açısından önemlidir.

Daha sonra yapılacak çalışmalar için şu noktalar önerilebilir:

Bilgisayarda kaçınılmaz olan sayısal hataları azaltmak için çift duyarlı aritmetik kullanılabilir.

Sistemin sürekli hal davranışı belirlenirken bütün çıkış büyüklüklerinin hesaplanması yerine sadece kontrol denklemlerinde kullanılan çıkış büyüklükleri hesaplanabilir. Bu husus çıkış denklemlerinin iki grup halinde (kontrol büyüklükleri ve gerçek çıkış büyüklükleri) düzenlemesi ile mümkündür.

Bilgisayar programı daha gelişmiş bir programlama diline dönüştürülerek daha kısa sürede çözüm elde edilebilir veya daha kullanışlı bir paket program haline getirilebilir.

KAYNAKLAR

- ADDOWEECH, K. E., SHEPHERD, W. ve HULLEY, L. N., 1989. Induction Motor Speed Control Using a Microprocessor-Based PWM Inverter, *IEEE Trans. on Industrial Electronics*, Vol. 36, No. 4: 516-522.
- APPRILLE, T. J. ve TRICK, T. N., 1972. Steady State Analysis of Nonlinear Circuits with Periodic Inputs, *Proc. IEEE*, Vol. 60: 108-114.
- BRONKOWSKI, R. L., 1986. A Technique for Increasing Power Transistor Switching Frequency, *IEEE Trans. on Industry Applications*, Vol. IA-22, No. 2:240-243.
- CHELLAMUTHU, C. ve SASTRY, V. V., 1981. Simulation of Chopper Fed DC Motor Drive System - power Oriented Approach, *Electrical Machines and Electromech.*, Vol. 6: 451-466.
- CHUA, L. O., HASLER, M. ve VERBURG, F., 1982. Dynamics of A Piecewise Linear Resonant Circuits, *IEEE Trans. on Circuits and Systems*, Vol. CAS-29:535-547.
- DEMİR, Y., (Devam ediyor). Lineer Olmayan Devrelerin Durum Denklemlerinin Değişik Bölgeler için Bulunması ve Bölgeler Arası Geçişler için Anahtarlama Denklemlerinin Hesaplanması, Doktora Tezi, Fırat Üniversitesi, Elazığ.
- DERVİŞOĞLU, A., 1971. State Equations and Initial Values in Active RLC Networks, *IEEE Trans. Circuits Theory (Corresp.)*, Vol. CT-18: 544-547.
- DESOER, C. A., 1970. Notes for a Second Course on Linear Systems, Van Nostrand Reinhold, New York.
- FAHMY, M. F., MAKKY, M. Y. ve DOSS, M. M., 1987. A Method for Frequency Domain Analysis of Switched Capacitor Filters, *IEEE Trans. Circuits Syst.*,

Vol. CAS-34, No. 8: 955-960.

FATTOUH, F. C., RUMIN, N. C. ve BLOSTEIN, M. L., 1980. Computational Techniques for The Periodic Steady-state Analysis of Large Nonlinear Circuits, **IEEE Trans. on Circuits and Systems**, Vol. CAS-27: 332-335.

FILICORI, F., MONACO, V. A. ve NALDI, C., 1979. Simulation and Design of Microwave Class - C Amplifiers Through Harmonic Analysis, **Proc. IEEE/MIT-S**, Orlando, USA.

GRAY, P. E. ve SEARLE, C. L., 1969. **Electronic Principles, Physics, Models and Circuits**, Wiley and Sons Ltd., New York.

HARRIS Semiconductor, 1989. **Power MOSFET' s**, Data Book, Harris Corp., USA.

HERDEM, S., 1987. **Tristör ve Güç Transistörlerinin Anahtarlama Davranışları**, Yüksek Lisans Tezi, Fırat Üniversitesi, Elazığ.

HUANG, Q. ve LIU, R.-W., 1989. A Simple Algorithm for Finding All Solutions of Piecewise-linear Networks, **IEEE Trans. on Circuits and Systems**, Vol. 36, No. 4: 600-609.

KILCI, T. ve KÖKSAL, M., 1987. Doğru akım Motorlarının Yarı İletken Sürücülerle Kontrolü ve Simülasyonu, **Elk. Müh. 2. Ulusal Kong. Bildiri Kitabı**, 2/2: 720-723, ODTÜ, Ankara, (a).

KILCI, T. ve KÖKSAL, M., 1987. Simulation and Control of DC Motors by Semiconductor Drivers, **Proc. Int. IASTED Symp. on Simulation, Modelling and Development**:139-142, Kahire, Mısır.

KÖKSAL, M. ve TOKAD, Y., 1976. On The Solution of Linear Circuits Containing Periodically Operated Switches, **Proc. European Conf on Circuits Theory and Design.**, Vol. 1: 77-82.

KÖKSAL, M. ve TOKAD, Y., 1977. State Space Formulation of Linear Circuits Containing Periodically Operated Switches, **Circuits Theory and Appl.**, Vol. 5: 155-170.

- KÖKSAL, M. ve TÜRKBEYLER. E., 1987. Peryodik Sinyaller İle Uyarılmış Lineer Olmayan Sistemlerin Analizi İçin Çabuk Yakınsayan Yeni Bir Metod, XI. Yöney. Araş. Kong. Bildiri Kitabı: XV-21 - XV-30, Marmara Ün., İstanbul.
- KÖKSAL, M., 1984. A Computer Program for The General Solutions of Periodically Time - varying Systems, ODTÜ Gaziantep Elekt. - Elektrn. Müh. Böl., Tek. Rap., No. GEEE/CAS-84/2, (a).
- KÖKSAL, M., 1984. Lineer Olmayan Sistemlerin Peryodik Çözümlerinin Bilgisayar Kullanarak Bulunması, 1. Ulusal Bilgisayar Destekli Tasarım Semp. Tebliğleri, 431-442, İzmir, (b).
- KÖKSAL, M., 1984. A Fast Convergent Method for The Analysis of Nonlinear Systems with Periodic Excitations, IEEE Int. Symp. on Circuit and Syst. Proc., Vol. 3-3:1362-1364, Montreal-Canada, (c).
- KÖKSAL. M., 1975. Analysis and Applications of linear Circuits Containing Periodically Operated Switches, Doktora Tezi, ODTÜ, Ankara.
- MATSU, N., OTHASHI, Y. ve THUNEHİRO, Y., 1979. Some Considerations on The Analysis of A State Transition Approach to Thyristor Circuits, Electrical Engineering in Japon, Vol. 99, No. 2:77-83.
- MILIAS, J., ZACHARIAS, TH., HATZIADONIU, C. ve GALANOS, G. D., 1987. An Algorihm for Transient Simulation of Power Electronics Systems, IEEE Trans. on Circuits and Systems, Vol. CAS-34, No. 8: 969-973.
- MUKHOPACHYAY, A. K. ve MITRA, T. K., 1980. A Matrix Modelling of Thyristor, J. L. E. (India) Part ET3, Vol. 62: 82.
- MURAKAMI, Y., 1987. A Method for The Formulation and Soluation of circuits Composed of Switches and Linear RLC Elements, IEEE Trans. on Circuits and Systems, Vol. CAS-34, No. 5: 496-509.

- NACAROĞLU, A., 1989. **Computer Oriented Analysis of Linear Circuits Containing Periodically Operated Switches ; Applications to Switching Capacitor Networks**, Doktora tezi, ODTÜ, Gaziantep.
- NAKHLA, M. S. ve BRANIN, Jr., 1977. Determining The Periodic Responce of Nonlinear Systems by A Gradient Method, *Int. J. Circuit Theory Apl.*, Vol. 5, No. 3: 255-273.
- NAKHLA, M. S. ve VLACH, J., 1976. A Piecewise Harmonic Balance Technique for Determination of Periodic Response of Nonlinear Systems, *IEEE Trans. on Circuits and Systems*, Vol. CAS-23: 85-91.
- OPAL, A. ve VLACH, J., 1989. Analysis and Sensitivity of Periodically Switched Linear Networks, *IEEE Trans. on Circuits and Systems*, Vol.36, No.4: 522-532.
- PEARMAN, R. A., 1980. **Power Electronics Solid State Motor Control**, Reston Publishing Co., Virginia, USA.
- RACZ, I., BORKA, J., LUPAN, K. ve MIKLOS, D., 1969. Quasi-periodic Dynamic Behaviour of Piecewise Linear Multy-parameter Systems, *Periodica Polytechnica*, Vol. 13, No. 3: 206-219.
- RAJAGOPALAN, V., 1987. **Computer-aided Analysis of Power Electronic Systems**, Marcel Dekker Inc., New York.
- SARKAR, G. ve MUKHOPACHYAY, A. K., 1982. The Modelling of A Thyristor for On-Off Conditions, *INT. J. Electronics*, Vol. 53, No. 3: 281-288.
- SINGH, S. N. ve KOHLI, D. R., 1983. State Variable Analysis of A Chopper Fed Seperately Excited DC Motor, *Electrical Machines and Power systems*, Vol. 8:45-55.
- SKELBOE, S., 1980. Computation of The Periodic Steady - state Responce of Nonlinear Systems by Extrapolation Methods, *IEEE Trans. on Circuits and Systems*, Vol. CAS - 27 : 161-175.

- SKELBOE, S., 1982. Conditions for Quadratic Convergence of Quick Periodic Steady-state Methods, *IEEE Trans. on Circuits and Systems*, Vol. CAS-29: 234-239.
- TANAKA, M. ve MORI, S., 1982. Topological Formulation for The Coefficient Matrices of State Equations for Switched Capacitor Networks, *IEEE Trans. Circuits Syst.*, Vol. CAS-29, No. 2: 106-115.
- TOHUMOĞLU, G., 1993. **Computation of Steady State Responses of Periodically Excited Nonlinear Systems by Extended Spectral Analysis**, Doktora Tezi, ODTÜ, Ankara.
- TÜRKBEYLER, E. ve KÖKSAL, M., 1988. Periodic Steady-state Solutions of Power Electronic Circuits, *Proc. Int. AMSE Conf. on Modelling and Simulation*, Vol. 2D: 93-104, İstanbul, Türkiye.
- TÜRKBEYLER, E., 1987. **A New Approach for The Analysis of Power Electronic Networks by Treating such Networks as Linear Circuits Containing POS**, Yüksek Lisans Tezi, ODTÜ, Gaziantep.
- WELLEKENS, C. J., 1982. Equivalence of Two Design of Bilinear Switched Capacitor Ladder Filters, *Electronic Letters*, Vol. 18, No. 6: 246-247.
- WILLIAMS, B. W., 1977. State Space Thyristor Computer Models, *Proc. of IEE*, Vol. 124, No. 9: 43-46.
- WU, M. Y., 1980. Solution of Certain Classes of Linear Time - varying Systems, *Int. J. Control*, Vol. 31, No. 1: 11-20, (a).
- WU, M. Y., 1980. Solvability and Representation of Linear Time-varying Systems, *Int. J. Control*, Vol. 31, No. 5: 937-945, (b).

YOSHIDA, Y., MOHRI, K. ve YOSHINO, K., 1986. PWM Inverter Using High-Gain Pulse-Triggered Power Transistors and a New PWM Control Method, **IEEE Trans. on Industrial Electronics**, Vol. IE-33, No. 2: 132-137.

ZADEH, L. A., 1961. Time - varying Networks I, **Proc. IRE**, Vol. 49, No.10: 1488-1503.



EK - A PGEDAN PROGRAMININ LİSTESİ

```

C   PERİYODİK OLARAK ÇALISAN GÜC ELEKTRONIKI DEVRELERİNİN ANALIZI
C   İÇİN BİLGİSAYAR PROGRAMI ..... PGEDAN
C
SUBROUTINE PGEDAN
DIMENSION FRQ(5),A(400),C(600),F(800),G(400),X0(20),CY(400),
IFC(80),NSV(50),C(60),HXI(80),FIB(80),FIX(80),YT(80),IR(80),
2P(80),C(80),DRM(200),NCE(80),NSX(80),DKM(600),STI(500),NZV(50),
3TV(80),DEK(80),NSVY(40),AY(300),FY(400),GY(200),IVRI(20),IVZ(20),
4DELY(20),YREF(20),ICV(20),ZV(30),R(80)
COMPLEX PW,API(50),U,HX(50),APIB(40),AFIB(40),CC(40),CC(40),CP,
IPUAF(40),PUAF(40),CHX(40),DUP(40),B(200),D(400),BJ(100),DJ(200),
2BY(200),DY(200)
INTEGER CSN(50),SSCS(80),CS(50),CN,SSN,GDM(600),V0(50),SIRA(500),
1VI(50),CSNT,ZVM
C   GİRİŞ BİLGİLERİNİN ALINMASI
CALL OKU1(SSN,NSV,NOV,NT,NF,NKT,NCE,CN,CSN,DI,NDT,IVS,NZV,ZV,TT0,
1ICV,DELY,YREF)
CALL OKU0(SSN,NSV,NOV,NF,NCE,CN,CSN,A,B,C,D,F,G,DKM,DRM,GDM,SSCS,
1FRQ,X0,CS,TO,NSX,NKT)
K=1
IK=1
SIRA(1)=1
STI(1)=0
T=TO
DT=TT0/NDT
IDT=1
NTT=1
NTTE=1
IV=0
IVI=1
NZVM=0
ZVM=0
DO 4 I=1,CN
4 NZVM=NZVM+CSN(I)
DO 3 I=1,NZVM
3 ZVM=ZVM+NZV(I)
1111 IF(T.LT.(NTT*TT0)) GOTO 47
T=NTT*TT0
NTT=NTT-I
C   GEÇİCİ HAL + SUREKLİ HAL ÇÖZÜMÜ
47 CALL TAMCOZ(K,DI,T,TO,NSV,NOV,X0,FRQ,NF,A,B,C,D,YT,HX,NSX)
C   ÇÖZÜM SONUÇLARINA GÖRE ELEMAN DURUMLARI
IGT=0
CSNT=0
TK=(NTT-1)*TT0
DO 5 I=1,CN
MOD=1
IF(LEQ 1) GOTO 58
CSNT=CSNT+CSN(I-1)
58 DO 8 J=1,CSN(I)
II=CSNT+J
IF(MOD.EQ.2) GOTO 59
IF(MOD.EQ.1.AND.NZV(IJ).NE.0) GOTO 48
GOTO 8
59 JD=1
NNRT=0
NNVT=0
DO 13 II=1,CN
DO 13 J1=1,CSN(II)
IF(II.GE.I.AND.J1.GE.J) GOTO 14
JD=JD-I
NNVT=NNVT+NCE(JD-1)*(NOV+1)
NNRT=NNRT+NCE(JD-1)
13 CONTINUE
14 KI=0
DO 15 I1=1,NCE(JD)
DO 15 J1=1,NOV
I1D=I1-(J1-1)*NCE(JD)+NNVT
IF(DKM(I1D).EQ.0.) GOTO 15
KI=KI-I
DEK(KI)=DKM(I1D)*YT(J1)
I2=KI+NNVT+NCE(JD)*NOV
IF(KI.EQ.1) GOTO 20
IF(DKM(I2-1-1)EQ.25.30)
20 IF(DEK(KI).GE.DRM(NNRT+KI)) GOTO 23
IF(DKM(I2).EQ.2.) GOTO 23
GOTO 17
25 IF(KONT.EQ.1.AND.DEK(KI).GE.DRM(NNRT+KI)) GOTO 23
GOTO 17
30 IF(KONT.EQ.1.OR.DEK(KI).GE.DRM(NNRT+KI)) GOTO 23
GOTO 17
23 IF(DKM(I2).EQ.0.)GOTO 50
KONT=1
15 CONTINUE
WRITE(6,*)DKM LERDE HATA OLABİLİR "
C   GEÇİŞ DURUM MATRİSİNİN KONTROLU "
50 IF(LEQ 1) GOTO 51

```

```

    IGT=IGT+CSN(I-1)*CSN(I-1)
51 IKT=(J-1)*CSN(I)+CSN(I)+IGT
    IF(GDM(IKT).EQ.1) GOTO 46
    GOTO 17
C   DISARDAN YAPILAN ETKININ KONTROLU
46 IF(MOD.EQ.1) GOTO 53
48 IF(MOD.EQ.2.AND.NZV(IJ).EQ.0) GOTO 53
    NZVT=0
    DO 62 KK=1,IJ-1
62 NZVT=NZVT+NZV(KK)
    DO 63 I1=1,NZV(IJ)
    I11=NZVT+I11
    ALIM=T-0.5*DT
    ULIM=T+0.5*DT
    ZAM=IK+ZV(I11J)
    IF(ZAM.GT.ALIM.AND.ZAM.LE.ULIM) GOTO 59
63 CONTINUE
    GOTO 8
17 KONT=0
8 CONTINUE
    IF(MOD.NE.1) GOTO 5
    MOD=2
    GOTO 58
231 WRITE(6,21)K,I,CS(I),T
21 FORMAT(/X,'OLUMSUZ',I2,' CS(',I2,')=',I3,I,X,'T=',E14.6/)
    WRITE(6,22)(YT(J),J=1,NOV)
22 FORMAT(/X,'YT=',6(E11.4,1X))
    GOTO 5
53 CS(I)=I
5 CONTINUE
C   ELEMAN DURUMLARINA GORE SISTEM DURUMU
SSN=1
    DO 40 I=1,CN
40 SSN=SSN*CSN(I)
    DO 55 I=1,SSN
    DO 60 J=1,CN
    IJ=I+(J-1)*SSN
    IF(CS(IJ).EQ.SSCS(IJ)) GOTO 60
    GOTO 55
60 CONTINUE
    GOTO 65
55 CONTINUE
    WRITE(6,70)
70 FORMAT(/' HIC BIR DURUMA GIRILMEDI. VERILERI KONTROL EDINIZ.'/)
    GOTO 66
65 IF(NTT.GT.NTTE) GOTO 561
    IF(L.EQ.K.AND.IDT.NE.2) GOTO 66
561 IF(IDT.EQ.1) GOTO 562
    IF(L.EQ.KG) GOTO 331
    DT=DT*50.
    IDT=I
    T=TG
    I=KG
    GOTO 563
562 IF(T-SITI(IK).GE.0.6*DT.AND.T-SITI(IK).LE.1.4*DT) GOTO 333
C   BIR PERYOTTAKI DURUMLAR VE GECIS ZAMANLARI
563 K0=K
    K=I
    IK=IK+1
    T0=T
    SITI(IK)=T
    SIRAI(IK)=I
    IF(NTT.EQ.NTTE) GOTO 81
    DO 77 I1=I,IK
    IIV=I1-(IIV-1)
    TV(IIV)=SITI(I1)-(NTT-2)*T10
    IF(TV(IIV).LT.DT) TV(IIV)=0.
77 V1(IIV)=SIRAI(I1)
    IVK=IK-IIV
    IIV=IK
    NTTE=NTT
    DO 78 I1=1,IVK
    IF(V1(I1).EQ.V0(I1)) GOTO 76
    IV=0
    GOTO 79
76 CONTINUE
    IV=IV+1
79 DO 76 I2=1,IVK
    V0(I2)=V1(I2)
78 V1(I2)=0
    IF(IV.GE.IVS) GOTO 555
81 IF(K.EQ.K0) GOTO 66
C   YENI DURUM ICIN X0'IN HESABI.....
    CALL ILKD(K,K0,T,NE,SSN,NSV,FRQ,G,F,HX,X0,NSX)
C   DT=DT0
66 T=T+DT
    GOTO 1111
333 TG=T
    T=T-DT
    DT=DT/50.
    KG=I
    IDT=2
    GOTO 66
331 IK=IK+1
    DT=DT*50.
    IDT=I
    GOTO 563
555 WRITE(6,52)
52 FORMAT(/2X,'SIRA',2X,'DURUM',3X,'ZAMAN')

```

```

DO 661 I=1,IK
WRITE(6,56)I,SIRA(I),SITI(I)
56 FORMAT(1X,I4,2X,I5,2X,E11.4)
661 CONTINUE
WRITE(6,92)
92 FORMAT(// ' SUREKLI HALDE          DURUMLARA GRES/'
1' GIRILEN DURUMLAR:          ZAMANLARI :/')
DO 93 J=1,IVK
93 WRITE(6,91)V0(J),TV(J)
91 FORMAT(7X,I3,3X,'.....',2X,E11.4)
C  DATALARIN SUREKLI HAL COZUMU ICIN HAZIRLANMASI
DO 115 I=1,IVK
115 NSVY(I)=NSV(V0(I))
NST=0
NSTT=0
KGP=0
KFP=0
DO 110 L=1,IVK
NSVT=0
NSVTT=0
KGTOP=0
KFTOP=0
DO 125 II=1,V0(L)-1
NSVT=NSVT+NSV(II)
125 NSVTT=NSVTT+NSV(II)*NSV(II)
IF(L.EQ.1) GOTO 118
NST=NST+NSVY(L-1)
NSTT=NSTT+NSVY(L-1)*NSVY(L-1)
118 NI=NSVY(L)
DO 201 II=1,NI
DO 251 I3=1,NF
II3=I1+(I3-1)*NI
II3Y=II3+NST*NF
II3L=II3+NSVT*NF
251 BY(II3Y)=B(II3L)
DO 201 I2=1,NI
II2=I1+(I2-1)*NI
II2Y=II2+NSTT
II2L=II2+NSVTT
201 AY(II2Y)=A(II2L)
DO 301 II=1,NOV
DO 351 I3=1,NF
II3=I1+(I3-1)*NOV
II3Y=II3+(L-1)*NOV*NF
II3L=II3+(V0(L)-1)*NOV*NF
351 DY(II3Y)=D(II3L)
DO 301 I2=1,NI
II2=I1+(I2-1)*NOV
II2Y=II2+NST*NOV
II2L=II2+NSVT*NOV
301 CY(II2Y)=C(II2L)
L1=L-1
IF(L.EQ.IVK) L1=1
N2=NSVY(L1)
IF(L.EQ.1) GOTO 381
KGP=KGP+NSVY(L)*NF
KFP=KFP+NSVY(L)*NSVY(L-1)
381 IF(V0(L).EQ.V0(L1)) GOTO 399
DO 401 II=1,SSN
DO 401 I2=1,SSN
IF(I1.EQ.I2) GOTO 401
IF(I1.EQ.V0(L).AND.I2.EQ.V0(L1)) GOTO 402
KGTOP=KGTOP+NSV(I2)*NF
KFTOP=KFTOP+NSV(I2)*NSV(II)
401 CONTINUE
WRITE(6,*)' F VE G LERIN AKTARILMASINDA HATA OLABILIR !!!'
399 DO 453 I1=1,N2
DO 503 I3=1,NF
II3=I1+(I3-1)*N2
II3Y=II3+KGP
503 GY(II3Y)=0.
DO 453 I2=1,NI
II2=I1+(I2-1)*N2
II2Y=II2+KFP
II1=I1+(I1-1)*N2
FY(II2Y)=0.
IF(II2.EQ.II1) FY(II2Y)=1.
453 CONTINUE
GOTO 110
402 DO 451 I1=1,N2
DO 501 I3=1,NF
II3=I1+(I3-1)*N2
II3Y=II3+KGP
II3L=II3+KGTOP
501 GY(II3Y)=G(II3L)
DO 451 I2=1,NI
II2=I1+(I2-1)*N2
II2Y=II2+KFP
II2L=II2+KFTOP
451 FY(II2Y)=F(II2L)
110 CONTINUE
IZ=0
DO 130 I=1,ZVM
DO 130 J=1,IVK
ALM=TV(J)-0.5*DT
ULM=TV(J)+0.5*DT
IF(ZV(I).GE.ALM.AND.ZV(I).LT.ULM) GOTO 132
GOTO 130
132 IZ=IZ+1

```

```

IVZ(IZ)=J
130 CONTINUE
JZ=0
IF(IVK.EQ.1) GOTO 127
DO 139 I=2,IVK
DO 141 J=1,IZ
IF(1.EQ.IVZ(J)) GOTO 139
141 CONTINUE
JZ=JZ+1
IVRI(JZ)=I
139 CONTINUE
127 NRT=JZ
CALL APENI(NSVY,TT0,IVK,NOV,NF,FRQ,TV,AY,BY,CY,DY,FY,GY,NRT,IVKI
1,ICV,YREF,DELY,NT)
RETURN
END
C
SUBROUTINE OKUD(SSN,NSV,NOV,NF,NCE,CN,CSN,A,B,C,D,F,G,DKM,DRM,GDM
1,SSCS,FRQ,X0,CS,TO,NSX,NKT)
DIMENSION A(1),C(1),F(1),G(1),DKM(1),DRM(1),NSX(1),NCE(1),NSV(1),
1FRQ(1),X0(1)
INTEGER GDM(1),SSCS(1),CS(1),CSN(1),CN,SSN
COMPLEX B(1),D(1),FREQ(10),PKUV
TO=0.
NSVT=0
NSVTT=0
NOVT=0
NSOVT=0
KGTOP=0
KFTOP=0
DO 9 I=1,CN
DO 9 II=1,CSN(I)
DO 9 I2=1,CSN(I)
IF(1.EQ.1) GOTO 8
II2=I2+(I2-1)*CSN(I)+(I-1)*CSN(I-1)*CSN(I-1)
GOTO 9
8 II2=I2+(I2-1)*CSN(I)
9 GDM(II2)=1
DO 13 L=1,SSN
IF(L.EQ.1) GOTO 11
NSVT=NSVT+NSV(L-1)
NSVTT=NSVTT+NSV(L-1)*NSV(L-1)
NOVT=NOVT+NOV
NSOVT=NSOVT+NSV(L-1)*NOV
11 DO 14 I2N=1,NSV(L)
DO 14 I2M=1,NSV(L)
II2M=I2M+(I2M-1)*NOV+NSOVT
14 C(II2M)=0.
DO 17 I11=1,NKT+1
DO 17 I3=1,NF
DO 112 II=1,NSV(L)
II13L=NSVT*NF+(I3-1)*NSV(L)+II
112 B(II13L)=(0.,0.)
DO 113 I1M=1,NOV
I1M18L=NOVT*NF+(I3-1)*NOV+I1M
17 D(I1M18L)=(0.,0.)
DO 13 II=1,NSV(L)
DO 13 I2=1,NSV(L)
II2L=I1+(I2-1)*NSV(L)+NSVTT
13 A(II2L)=0.
DO 24 L=1,NSVT+NSV(SSN)
DO 19 I1=1,SSN
DO 19 I2=1,SSN
IF(II.EQ.I2) GOTO 19
KGTOP=KGTOP+NSV(I2)*NF
KFTOP=KFTOP+NSV(I2)*NSV(II)
19 CONTINUE
DO 21 II=1,KGTOP
21 G(II)=0.
DO 23 I2=1,KFTOP
23 F(I2)=0.
NCOVT=0
NCST=0
NCET=0
DO 41 I=1,CN
41 NCST=NCST+CSN(I)
DO 39 ID=1,NCST
IF(ID.EQ.1) GOTO 51
NCET=NCET+NCE(ID-1)
NCOVT=NCOVT+NCE(ID-1)*(NOV+1)
51 DO 43 II=1,NCE(ID)
II1=II+NCET
DRM(II1)=0.
DO 43 I2=1,NOV+1
II2=II1+(I2-1)*NCE(ID)
II2L=II2+NCOVT
43 DKM(II2)=0.
39 CONTINUE
CALL OKUZ(A,B,C,D,F,G,DKM,DRM,GDM,SSCS,CS,FRQ,X0,TO,NSX)
IF(NKT.EQ.0) GOTO 115
NSVT=0
DO 110 KK=1,SSN
IF(KK.EQ.1) GOTO 125
NSVT=NSVT+NSV(KK)
125 PKUV=(1.,0.)
DO 110 IK=1,NKT
DO 110 IJ=1,NF
PIM=6.2831853*FRQ(IJ)

```



```

FREQ(I1)=CMPLX(0.,PIM)
PKUV=PKUV*FREQ(I1)
DO 140 I33=1,NSV(KIK)
ITT=I33+(I1-1)*NSV(KIK)
ITTT=ITT+NSVT*NF
140 B(ITT)=B(ITT)+PKUV*B(ITTT)
DO 145 I44=1,NOV
FIT=I44+(I1-1)*NOV
FITT=FIT+(KIK-1)*NOV*NF
145 D(ITT)=D(ITT)+PKUV*D(ITTT)
110 CONTINUE
115 CALL YAZ(NSV,SSN,NOV,NF,A,B,C,D,F,G,NCE,DKM,DRM,CN,CSN,SSCS)
RETURN
END

```

C

```

SUBROUTINE YAZ(NSV,SSN,NOV,NF,A,B,C,D,F,G,NCE,DKM,DRM,CN,CSN,
ISSCS)
DIMENSION A(1),C(1),F(1),G(1),NSV(1),DRM(1),DKM(1),NCE(1)
INTEGER CSN(1),SSCS(1),CN,SSN
COMPLEX B(1),D(1)
NSVT=0
NSVTT=0
NOVT=0
NSOVT=0
KGTOP=0
KFTOP=0
WRITE(6,7)
7 FORMAT(// ' SUREKLI HALDE GIRILEN DURUMLARIN VE BU DURUMLARDA'//
1' KALMA SURELERININ BELIRLENMESI :'/)
DO 8 I=1,SSN
IF(L.EQ.1) GOTO 9
NSVT=NSVT+NSV(L-1)
NSVTT=NSVTT+NSV(L-1)*NSV(L-1)
NOVT=NOVT+NOV
NSOVT=NSOVT+NSV(L-1)*NOV
NSOVT=NSOVT+NCE(L-1)*(NOV+1)
9 II=1+NSVTT
I2=1+(NSV(L-1)*NSV(L))
WRITE(6,10)I,(A(I),I=1,12,NSV(L))
10 FORMAT(//1X,'***** DURUM=,I2,' *****//1X,'A=',5(E1
11.4,1X))
IF(NSV(L).EQ.1) GO TO 17
DO 11 I=2,NSV(L)
I4=1+NSVTT
I5=I4+(NSV(L-1)*NSV(L))
11 WRITE(6,12) (A(IJ),IJ=I4,I5,NSV(L))
12 FORMAT(3X,5(E11.4,1X))
17 M1=1+NSOVT
M2=M1+(NSV(L-1)*NOV)
WRITE(6,15) (C(I),I=M1,M2,NOV)
15 FORMAT(//1X,'C=',5(E11.4,1X))
IF(NOV.EQ.1) GOTO 20
DO 18 I=2,NOV
M3=1+NSOVT
M4=M3+(NSV(L-1)*NOV)
18 WRITE(6,19) (C(IM),IM=M3,M4,NOV)
19 FORMAT(3X,5(E11.4,1X))
20 II=1+NSVT*NF
I22=II+(NF-1)*NSV(L)
WRITE(6,21) (B(I),I=1,1,I22,NSV(L))
21 FORMAT(//1X,'B=',5(E11.4,1X))
IF(NSV(L).EQ.1) GOTO 24
DO 22 I=2,NSV(L)
I4=1+NSVT*NF
I5=I4+(NF-1)*NSV(L)
22 WRITE(6,23) (B(IJ),IJ=I4,I5,NSV(L))
23 FORMAT(3X,5(E11.4,1X))
24 M1=1+NOVT*NF
M2=M1+(NF-1)*NOV
WRITE(6,26) (D(I),I=M1,M2,NOV)
26 FORMAT(//1X,'D=',5(E11.4,1X))
IF(NOV.EQ.1) GO TO 31
DO 29 I=2,NOV
M3=1+NOVT*NF
M4=M3+(NF-1)*NOV
29 WRITE(6,30) (D(IM),IM=M3,M4,NOV)
30 FORMAT(3X,5(E11.4,1X))
C F VE G LERIN YAZDIRILMASI.....
31 II=L
DO 40 I2=1,SSN
IF(I1.EQ.I2) GOTO 40
IN=1+KFTOP
IM=IN+(NSV(I1-1)*NSV(I2))
WRITE(6,13)I1,I2,(F(I),I=IN,IM,NSV(I2))
13 FORMAT(//1X,'DURUM,I2,1X,'-',1X,'DURUM,I2//' F=',5(E11.4,1X))
IF(NSV(I2).EQ.1) GOTO 16
DO 14 I=2,NSV(I2)
I4=1+KFTOP
I5=I4+(NSV(I1-1)*NSV(I2))
14 WRITE(6,42) (F(IJ),IJ=I4,I5,NSV(I2))
42 FORMAT(3X,5(E11.4,1X))
16 KFTOP=KFTOP+NSV(I1)*NSV(I2)
II=1+KGTOP
I22=II+(NF-1)*NSV(I2)
WRITE(6,25) (G(I),I=1,1,I22,NSV(I2))
25 FORMAT(//1X,'G=',5(E11.4,1X))
IF(NSV(I2).EQ.1) GOTO 27
DO 26 I=2,NSV(I2)
I4=1+KGTOP

```

```

IE=I4-(NF-1)*NSV(I2)
26 WRITE(6,43)(G(IJ),IJ=I4,IB,NSV(I2))
43 FORMAT(3X,5(E11.4,1X))
27 KGTOP=KGTOP+NSV(I2)*NF
40 CONTINUE
IF(L.LT.SSN) GOTO 47
NCOVT=0
SSN=1
IE=1
NCST=0
CSNT=0
NCET=0
DO 41 I=1,CN
41 SSN=SSN*CSN(I)
DO 411 I=1,CN
411 NCST=NCST+CSN(I)
DO 39 ID=1,NCST
IF(ID.EQ.1) GOTO 51
NCET=NCET+NCE(ID-1)
NCOVT=NCOVT+NCE(ID-1)*(NOV+I)
51 M1=1+NCOVT
M2=M1+NOV*NCE(ID)
IF(IE.EQ.1) GOTO 57
IF(ID.LE.CSNT) GOTO 58
57 WRITE(6,49)IE
49 FORMAT(3X,I3,' ELEMEN')
CSNT=CSNT+CSN(IE)
IE=IE+1
58 WRITE(6,32)(DKM(I),I=M1,M2,NCE(ID))
32 FORMAT(1X,'DKM=',8(F7.3,1X)/5X,8(F7.3,1X))
IF(NCE(ID).EQ.1) GOTO 35
DO 33 I=2,NCE(ID)
M1=1+NCOVT
M2=M1+NOV*NCE(ID)
33 WRITE(6,34)(DKM(IM),IM=M1,M2,NCE(ID))
34 FORMAT(5X,8(F7.3,1X)/5X,8(F7.3,1X))
35 M1=1+NCET
WRITE(6,36)DRM(M1)
36 FORMAT(1X,'DRM=',E11.4)
IF(NCE(ID).EQ.1) GOTO 39
DO 37 I=2,NCE(ID)
M1=1+NCET
37 WRITE(6,36)DRM(M1)
38 FORMAT(5X,E11.4)
39 CONTINUE
M2=1+(CN-1)*SSN
WRITE(6,44)(SSCS(I),I=1,M2,SSN)
44 FORMAT(1X,'SSCS=',6I6)
IF(SSN.EQ.1) GOTO 47
DO 46 I=2,SSN
M2=1+(CN-1)*SSN
46 WRITE(6,48)(SSCS(IM),IM=1,M2,SSN)
48 FORMAT(6X,6I6)
47 WRITE(6,'*')*****
8 CONTINUE
RETURN
END

```

```

C
C TAM COZUM YAPAN ALTPROGRAM
SUBROUTINE TAMCOZ(K,DI,T,TD,NSV,NOV,X0,FRQ,NF,A,B,C,D,YT,HX,NSX)
DIMENSION FRQ(1),A(1),C(1),XD(1),NSX(1),C(80),P(80),R(80),IR(80),
FIC(80),NSV(1),C(80),HX(80),FIB(80),FIX(80),YT(80)
COMPLEX FRQ(5),PW,API(80),U,HX(1),APIB(80),AFIB(80),CO(80),B(1),
ICP,PUAR(80),PUAF(80),CHX(80),DUP(80),CC(80),CFIB(80),D(1),BJ(80),
2DI(80)
NSVT=0
NSVTT=0
NOVT=0
NSOVT=0
IF(K.EQ.1) GOTO 1011
DO 1010 I=1,K-1
NSVT=NSVT+NSV(I)
NSVTT=NSVTT+NSV(I)*NSV(I)
NOVT=NOVT+NOV
1010 NSOVT=NSOVT+NSV(I)*NOV
1011 CALL EAT(NSV,K,DI,T,TD,A,FIC,NSVTT,Q,P,Q,R)
CALL EATX(K,NSV,X0,FIC,FIX,NSVT,NSX)
DO 1021 I=1,NSV(K)
1021 HX(I)=0
DO 1022 I=1,NOV
CHX(I)=0
YT(I)=0
1022 DUP(I)=0
DO 221 JJ=1,NF
PIM=2*(22./7.)*FRQ(JJ)
PW=CMPLX(0.,PIM)
DO 1001 I=1,NSV(K)
IJL=NSVT*NF+(JJ-1)*NSV(IQ)+I
IL=NSVT+I
1001 BJ(IL)=B(IJL)
DO 1003 I=1,NOV
IJL=NOVT*NF+(JJ-1)*NOV+I
IL=NOVT+I
1003 DJ(IL)=D(IJL)
CALL API(PW,API,NSV,K,A,NSVTT,CO,IR)
CALL APIB(APIB,API,BJ,NSV,K,NSVT)
CALL EATB(K,NSV,BJ,FIC,FIB,NSVT)
CALL AEATB(K,NSV,API,FIB,AFIB,CFIB)
CP=CEXP(PW*T)

```

```

CP0=CEXP(PW*TO)
DO 32 I=1,NSV(K)
PUAB(I)=AFIB(I)*CP*U(JJ)
PUAF(I)=AFIB(I)*CP0*U(JJ)
32 HX1(I)=HX1(I)+PUAF(I)-PUAB(I)
DO 1004 I=1,NOV
I1J=NOVT+I
1004 DUP(I)=DUP(I)+DJ(I1J)*CP*U(JJ)
221 CONTINUE
DO 33 I=1,NSV(K)
33 HX(I)=FDX(I)+HX1(I)
DO 1005 I1=1,NOV
DO 1005 I2=1,NSV(K)
I12=I1+(I2-1)*NOV
I12L=I12+NSOVT
1005 CHX(I1)=CHX(I1)+C(I12L)*HX(I2)
DO 1006 J1=1,NOV
1006 VTY(J1)=REAL(CHX(J1)+DUP(J1))
RETURN
END

C
SUBROUTINE AEATE(K,NSV,API,FIB,AFIB,CFIB)
COMPLEX API(1),AFIB(1),CFIB(1),FIB(1)
DIMENSION NSV(1)
DO 5 I1=1,NSV(K)
AFIB(I1)=(0.,0.)
DO 5 I2=1,NSV(K)
CFIB(I2)=CMPLX(FIB(I2),0.)
I12=I1+(I2-1)*NSV(K)
AFIB(I1)=AFIB(I1)+API(I12)*CFIB(I2)
5 IF(CABS(AFIB(I1)).LE.1.E-30) AFIB(I1)=0
RETURN
END

C
SUBROUTINE EATX(K,NSV,X0,FIC,FDX,NSVT,NSX)
DIMENSION NSV(1),X0(1),FIC(1),FDX(1),NSX(1)
DO 5 I1=1,NSV(K)
FDX(I1)=0.
DO 5 I2=1,NSV(K)
I12=I1+(I2-1)*NSV(K)
I21=NSVT+I2
I22=NSX(I21)
FDX(I1)=FDX(I1)+FIC(I12)*X0(I22)
5 IF(ABS(FDX(I1)).LE.1E-30) FDX(I1)=0.
RETURN
END

C
SUBROUTINE EATB(K,NSV,B,FIC,FIB,NSVT)
DIMENSION NSV(1),FIC(1)
COMPLEX B(1),FIB(1)
DO 5 I1=1,NSV(K)
FIB(I1)=0.
DO 5 I2=1,NSV(K)
I2L=2+NSVT
I12=I1+(I2-1)*NSV(K)
5 FIB(I1)=FIB(I1)+FIC(I12)*B(I2L)
RETURN
END

C
SUBROUTINE APIT(PW,API,NSV,K,A,NSVIT,CO,IR)
DIMENSION A(1),IR(1),NSV(1)
COMPLEX PW,API(1),CO(1),CP(40)
LIM=NSV(K)*NSV(K)
DO 131 I=1,LIM
131 API(I)=CMPLX(0.,0.)
DO 1 I3=1,NSV(K)
DO 1 I2=1,NSV(K)
I23=I2+(I3-1)*NSV(K)
I23L=I23+NSVIT
1 CO(I23)=CMPLX(A(I23L),0.)
DO 11 I4=1,NSV(K)
I44=I4+(I4-1)*NSV(K)
11 CO(I44)=CO(I44)-PW
N=NSV(K)
CALL CMINV(CO,CP,N,IR)
DO 2 I3=1,NSV(K)
DO 2 I2=1,NSV(K)
I23=I2+(I3-1)*NSV(K)
API(I23)=CP(I23)
2 IF(CABS(API(I23)).LE.1E-30) API(I23)=0
RETURN
END

C
SUBROUTINE APITB(APIB,APLB,NSV,K,NSVT)
DIMENSION NSV(1)
COMPLEX APIB(1),APL(1),B(1)
DO 5 I1=1,NSV(K)
APIB(I1)=(0.,0.)
DO 5 I2=1,NSV(K)
I2L=2+NSVT
I12=I1+(I2-1)*NSV(K)
5 APIB(I1)=APIB(I1)+APL(I12)*B(I2L)
RETURN
END

C
SUBROUTINE EAT(NSV,K,DI,T,TO,A,FIC,NSVIT,O,P,Q,R)
DIMENSION FIC(1),A(1),O(1),P(1),Q(1),R(1),NSV(1)
LIM=NSV(K)*NSV(K)
DO 131 I=1,LIM

```

```

131 FIC(I)=0.
DO 1 I=1,NSV(K)
DO 1 I=1,NSV(K)
II=I+(I-1)*NSV(K)
IIL=I+NSVT
1 Q(I)=A(IIL)*(T-T0)
N=NSV(K)
CALL EXPMT(Q,P,N,DI,Q,R)
DO 5 II=1,NSV(K)
DO 5 I2=1,NSV(K)
IIL2=I1+(I2-1)*NSV(K)
FIC(IIL2)=P(IIL2)
5 IF(ABS(FIC(IIL2))).LE.1E-30) FIC(IIL2)=0
RETURN
END

C
SUBROUTINE ILKD(KI,K0,T,NF,SSN,NSV,FRQ,G,F,HX,X0,NSX)
DIMENSION NSV(1),X0(1),F(1),G(1),X0F(40),X0G(40),FRQ(1),NSX(1),
IHXT(40)
COMPLEX PW,U,HX(1)
KFTOP=0
KGTOP=0
NSVT=0
IF(K0.EQ.1) GOTO 41
DO 42 I=1,K0-1
42 NSVT=NSVT+NSV(I)
41 DO 43 I=1,NSV(K0)
II=NSVT+I
I2=NSX(II)
43 HXT(I2)=REAL(HX(I))
DO 28 I=1,NSV(KI)
28 X0G(I)=0
DO 10 II=1,SSN
DO 10 I2=1,SSN
IF(I1.EQ.I2) GOTO 10
IF(I1.EQ.K0.AND.I2.EQ.KI) GOTO 15
KFTOP=KFTOP+NSV(II)*NSV(I2)
KGTOP=KGTOP+NSV(I2)*NF
10 CONTINUE
15 DO 20 II=1,NSV(KI)
X0F(II)=0
DO 20 I2=1,NSV(K0)
IIL2=KFTOP+II+(I2-1)*NSV(KI)
I21=NSVT+I2
I22=NSX(I21)
20 X0F(II)=X0F(II)+F(IIL2)*HXT(I22)
DO 35 II=1,NF
PIM=2*(22./7.)*FRQ(II)
PW=CMPLX(0.,PIM)
CP=CEXP(PW*T)
DO 35 I2=1,NSV(KI)
IIL2=KGTOP+(II-1)*NSV(KI)+I2
35 X0G(I2)=X0G(I2)+REAL(G(IIL2)*CP*U(II))
DO 21 I=1,NSV(KI)
21 X0(I)=X0F(I)+X0G(I)
RETURN
END

C
SUBROUTINE STIME(NT,TIMEV,TO)
DIMENSION TIMEV(1)
T1=0.0
T2=TO
DO 30 I=1,NT
30 TIMEV(I)=T1+(I-1)*(T2-T1)/(NT-1)
RETURN
END

C
SUREKLI HAL COZUMU YAPAN ALTPROGRAM
SUBROUTINE APENI(NSV,TO,K,M,NF,FRQ,ETA,A,B,C,D,F,G,NRT,IVRT,ICV,
1YREF,DELY,NT)
DIMENSION FRQ(1),TIMEV(500),T(20),ETA(20),A(1),C(1),G(80),
1,F(1),G(1),FIC(200),EI(400),EN(800),EM(400),O(80),ICV(1),
2F(80),C(80),R(80),IQ(100),IR(80),NSV(1),DEPO(10000),IVRT(1),
3YREF(1),DELY(1),OPQ(20)
COMPLEX FREQ(6),PW,API(80),BB(100),TRNSF(80),H(50),EJ(50),EF(50),
1U,CO(50),CP(50),CQ(50),CB1(50),CB2(50),ZFR,B(1),D(1),BJ(80),
2DJ(80)
INTEGER M
KI=K-1
DO 18 J=1,KI
T(J)=ETA(J+1)-ETA(J)
18 CONTINUE
T(K)=TO-ETA(K)
CALL REWIC(INF,DI,NSV,TO,K,M,NF)
CALL RWMI(NSV,K,M,NF,FRQ,FREQ,T,TO,A,B,C,D,F,G,DT,IFAC)
IFARI=0
1020 DO 15 J=1,NRT
J2=2*J
J21=J2-1
IVRTJ=IVRT(J)
TIMEV(J2)=ETA(IVRTJ)-DT/10
15 TIMEV(J21)=TIMEV(J2)-DT
C
YENI INTERVAL ZAMANI
NT=2*NRT
KI=K-1
DO 20 J=1,KI
T(J)=ETA(J+1)-ETA(J)
20 CONTINUE
T(K)=TO-ETA(K)

```

```

IF(IFARK.EQ.0) GOTO 19
WRITE(6,*)
WRITE(6,*) T=(T(I),I=1,K)
WRITE(6,*)
WRITE(6,*) ETA=(ETA(I),I=1,K)
CALL SRTIM(NI, TIMEV, TO)
19 CALL SFI(NSV, K, DI, T, A, F, FIC, FI, O, P, Q, R)
CALL SEN(NSV, K, FI, EN, O, P, Q)
CALL SEM(NSV, K, FI, EN, EM)
CALL SSMODE(NSV, K, INF, EM, O, P, IQ, IR)
DO 1000 J=1, NT
DO 1000 II=1, M
JII=(II-1)*NT+J
1000 DEPO(JII)=0.0
DO 221 JJ=1, NF
PW=FREQ(JJ)
NSVT=0
DO 1001 L=1, K
IF(L.EQ.1) GOTO 991
NSVT=NSVT+NSV(L-1)
991 DO 1001 L=1, NSV(L)
IIJL=I+(JJ-1)*NSV(L)+NSVT*NF
II=I+NSVT
1001 BJ(IL)=B(IIJL)
KGTOP=0
DO 1007 L=1, K
LI=L+1
IF(L.EQ.K) LI=1
IF(L.EQ.1) GOTO 992
KGTOP=KGTOP+NSV(L)
992 DO 1007 L=1, NSV(L)
IIJL=I+(JJ-1)*NSV(L)+KGTOP*NF
II=I+KGTOP
1007 GJ(IL)=G(IIJL)
DO 1003 L=1, K
DO 1003 I=1, M
IIJL=(L-1)*M*NF+(JJ-1)*M+I
II=(L-1)*M+I
1003 DJ(IL)=D(IIJL)
CALL SAP(PW, API, NSV, K, A, CO, CP, IR)
CALL SBB(PW, BB, API, NSV, K, T, BI, F, GJ, FIC, CO, CP, OQ, CBI, CB2)
DO 221 J=1, NT
KEI=K-1
DO 399 L=1, KEI
IF(TIMEV(J).GE.ETA(L).AND.TIMEV(J).LT.ETA(L+1)) GO TO 400
399 CONTINUE
L=K
400 N=NSV(L)
NSVT=0
NSVTI=0
NSOVI=0
IF(L.EQ.1) GOTO 401
DO 403 LN=2, L
NSVT=NSVT+NSV(LN-1)
NSVTI=NSVTI+NSV(LN-1)*NSV(LN-1)
403 NSOVI=NSOVI+M*NSV(LN-1)
401 CALL SH(L, PW, H, BB, NSV, K, TO, T, ETA, EN, CBI, CB2)
CALL SEP(L, PW, EJ, H, N, K, TO, EM, CO, CP, IR, NSVTI)
CALL SEP(L, PW, EP, EJ, APL, N, ETA, BI, CO, CP, OQ, NSVT, NSVTI)
CALL SHP(I, TRNSF, ML, PW, TIMEV(J), EP, API, NSV, DI, ETA, A, BI, C, DI, O, P, Q)
IR, CBI, NSVT, NSVTI, NSOVI)
ZAMAN=TIMEV(J)
DO 769 II=1, M
ZFR=ZAMAN*FREQ(JJ)
CBI(II)=CEXP(ZFR)
Q(II)=REAL(CBI(1))*TRNSF(II)*U(JJ)
JII=(II-1)*NT+J
769 DEPO(JII)=DEPO(JII)+Q(II)
221 CONTINUE
IF (IFARK.EQ.1) GO TO 1030
C GERCEK ZAMAN DEGERLEHINI BULMAK ICIN 'NEWTON RAPSON' METODU
IFARK=1
IIC=1
3000 DO 111 L=1, K
111 OPQ(L)=ETA(L)
IIC=IIC+1
DO 14 J=1, NRT
IVRTJ=IVRT(J)
J2=2*J
II=(ICV(J)-1)*NRT+J2
Y1=DEPO(II)
E=I-1
Y2=DEPO(I2)
ICVJ=ICV(J)
YR=YREF(J)
C WRITE(6,*) Y1=, Y1, ' YR=, YR
T1=TIMEV(I2)
T2=TIMEV(J2-1)
Z=DT*(Y1-0.9*YR-0.1*Y2)/(Y2-Y1)
FARK=ABS(YR-Y1)
IF(FARK.GE.DELY(ICVJ)) IFARK=0
14 OPQ(IVRTJ)=ETA(IVRTJ)+Z/IFAC
C YENI ITERASYON ICIN YENI ETA DEGERL...
CALL SCONT(K, OPQ, IFAC, IC, TO)
IF(IC.EQ.1) GO TO 3000
IFAC=IFAC/(2**IIC)
DO 112 L=1, K
112 ETA(L)=OPQ(L)
GO TO 1020

```

```

1030 WRITE(6,87)
87 FORMAT(/' SIRA ZAMAN(S)      CIKIS BUYUKLUKLERI...'
1' - - - - - ')
DO 1002 JJJ=1,NT
JM=(M-1)*NT+JJJ
MMI=M-4
IF(MLE,4) MMI=1
JMI=(M-MMI)*NT+JJJ
1002 WRITE(6,45) JJJ,TIMEV(JJJ),(DEPC(I),I=JJJ,JM,NT)
45 FORMAT(14,1X,E11.4,1X,5(E11.4,1X)/16X,5(E11.4,1X))
RETURN
END

C
SUBROUTINE SKTIM(NT,TIMEV,TO)
DIMENSION TIMEV(1)
TIMEV(1)=TO
WRITE(6,329)
329 FORMAT(/' ZAMANLA ILGLI TANIMLAMALAR:/'
9' NT : ZAMAN SAYISI '/'
4' TIMEV : ZAMAN VEKTORU '/')
CALL STIME(NT,TIMEV,TO)
WRITE(6,557) NT,(TIMEV(I),I=1,NT)
557 FORMAT(/' ZAMANLA ILGLI DEGERLER:/'
2' ZAMAN SAYISI :/'
3' NT =,I3/'
6' ZAMAN VEKTORU:/'
7' TIMEV(I) =/(16X,5(E11.4,1X))
RETURN
END

C
SUBROUTINE REWIC(INF,DI,NSV,TO,K,M,NF)
DIMENSION NSV(1)
INTEGER M
INF=6
DI=1E-10
WRITE(6,8)
8 FORMAT(///' SUREKLI HAL COZUMU :')
WRITE(6,10)
10 FORMAT(///' HESAPLAMALARDA KULLANILAN SABITLER:/'
1' INF: KARARLILIK TESTI ICIN, MATRIS**(2**INF)'IN NORMU; 1 ?/'
2' DI : USTEL MATRISDE ,(SON TERIMIN NORMU / ONCEKI TERIMIN NORMU);
3DI'//
4' BU ORNEGIN SABITLERI:/'
5' TO : SANIYE OLARAK ANA PERYOT/'
6' M : CIKIS BUYUKLUKLERININ SAYISI/'
7' K : BIR PERYOTTAKI INTERVAL SAYISI/'
8' NSV(I): DURUM DEGISKENI SAYISI/')
WRITE(6,11) INF,DI,TO,M,K
11 FORMAT(' BU ORNEK ICIN YUKARDAKI SABITLERIN DEGERLERI :/'
1' INF =',I2/' DI =',E10.4/' TO =',E10.4/'
2' M =',I2/' K =',I2)
DO 19 I=1,K
19 WRITE(6,13) I,NSV(I)
13 FORMAT(' NSV(',I2,') =',I2)
RETURN
END

C
SUBROUTINE FMMI(NSV,K,M,NF,FRQ,FREQ,T,TO,A,B,C,D,F,G,DT,IFAC)
DIMENSION A(1),C(1),F(1),G(1),FRQ(1),NSV(1),T(1)
COMPLEX FRQ(1),B(1),D(1)
INTEGER M
WRITE(6,33)
33 FORMAT(/'
1' FREKANSLA ILGLI TANIMLAMALAR:/'
4' NF : FREKANS SAYISI '/'
5' FRQ: FREKANS VEKTORU '/')
WRITE(6,86) NF
86 FORMAT(' FREKANSLA ILGLI DEGERLER:/' NF =,I2)
DO 83 I=1,NF
83 WRITE(6,87) I,FRQ(I)
87 FORMAT(' FRQ(',I2,') =,E11.4)
DT=TO/100000.
IFAC=2
DO 16 I=1,NF
PIM=6.2831853*FRQ(I)
16 FRQ(I)=CMPLX(0.,PIM)
NSVT=0
NSVTI=0
NSOVI=0
KGTOP=0
KFTOP=0
DO 8 I=1,K
LI=L+1
IF(L.EQ,K) LI=1
IF(L.EQ,1) GOTO 8
NSVT=NSVT+NSV(L-1)
NSVTI=NSVTI+NSV(L-1)*NSV(L-1)
NSOVI=NSOVI+NSV(L-1)*M
KFTOP=KFTOP+NSV(L-1)*NSV(L)
KGTOP=KGTOP+NSV(L)
8 CALL S111(L,NSV,K,M,NF,T,A,B,C,D,F,G,NSVT,NSVTI,NSOVI,KFTOP,KGTOP)
RETURN
END

C
SUBROUTINE S111(L,NSV,K,M,NF,T,A,B,C,D,F,G,NSVT,NSVTI,NSOVI,KFTOP
1,KGTOP)
DIMENSION T(1),A(1),C(1),F(1),G(1),NSV(1)
COMPLEX B(1),D(1)
LI=L+1

```

```

IF(L.EQ.K) L1=1
N=NSV(L)
I1=1+NSVTT
I2=1+(N-1)*N
WRITE(6,10)L,L,T(L),(A(I),I=1,I2,N)
10 FORMAT//1X,'*****INTERVAL=',I2,' *****'// 'T(',I2,')='
1,E11.4// ' A=',5(E11.4,1X))
IF(N.EQ.1) GO TO 17
DO 11 I=2,N
I4=1+NSVTT
I5=I4+(N-1)*N
11 WRITE(6,12) (A(IJL),IJL=I4,I5,N)
12 FORMAT(3X,5(E11.4,1X))
17 I1=1+KFTCP
I2=1+(N-1)*NSV(L1)
WRITE(6,13)(F(I),I=1,I2,NSV(L1))
13 FORMAT// ' F=',5(E11.4,1X))
IF(NSV(L1).EQ.1) GO TO 16
DO 14 I=2,NSV(L1)
I4=1+KFTCP
I5=I4+(N-1)*NSV(L1)
14 WRITE(6,12) (F(IJL),IJL=I4,I5,NSV(L1))
16 M1=1+NSOVT
M2=M1+(N-1)*M
WRITE(6,15) (C(I),I=M1,M2,M)
15 FORMAT// ' C=',5(E11.4,1X))
IF(M.EQ.1) GO TO 20
DO 18 I=2,M
M3=I+(I-1)*N*M
M5=M3+(N-1)*M
18 WRITE(6,19) (C(IM),IM=M3,M5,M)
19 FORMAT(3X,5(E11.4,1X))
20 I1=1+NSVT*NF
I2=I1+(NF-1)*N
WRITE(6,21)(B(I),I=I1,I2,N)
21 FORMAT// ' B=',5(E11.4,1X))
IF(N.EQ.1) GO TO 24
DO 22 I=2,N
I4=1+NSVT*NF
I5=I4+(NF-1)*N
22 WRITE(6,23)(B(IJL),IJL=I4,I5,N)
23 FORMAT(3X,5(E11.4,1X))
24 I1=1+KGTCP*NF
I2=I1+(NF-1)*NSV(L1)
WRITE(6,25)(G(I),I=I1,I2,NSV(L1))
25 FORMAT// ' G=',5(E11.4,1X))
IF(NSV(L1).EQ.1) GO TO 27
DO 26 I=2,NSV(L1)
I4=1+KGTCP*NF
I5=I4+(NF-1)*NSV(L1)
26 WRITE(6,23)(G(IJL),IJL=I4,I5,NSV(L1))
27 M1=I+(I-1)*NF*M
M2=M1+(NF-1)*M
WRITE(6,28)(D(I),I=M1,M2,M)
28 FORMAT// ' D=',5(E11.4,1X))
IF(M.EQ.1) GO TO 31
DO 29 I=2,M
M3=I+(I-1)*NF*M
M5=M3+(NF-1)*M
29 WRITE(6,30)(D(IM),IM=M3,M5,M)
30 FORMAT(3X,5(E11.4,1X))
31 RETURN
END

```

```

C
C   INTERWALLERIN HESABINDA 'IFAC' IN KONTROLU...
SUBROUTINE SCONT(K,OPQ,IFAC,IC,TO)
DIMENSION OPQ(1)
IC=0
DO 10 I=2,K
IF(OPQ(I)-OPQ(I-1))11,11,10
10 CONTINUE
IF(TO-OPQ(K))11,11,12
11 IFAC=2*IFAC
IC=1
WRITE(6,14)
14 FORMAT/
1 ' BIR INTERVAL BIR DIGERINE TASTI 'IFAC' DEGISTIRILDI.'
12 RETURN
END

C
C   HER K ICIN TBORIDEKI P' NIN HESABI...
SUBROUTINE SEP(L,PW,EP,EJ,API,N,ETA,B,CO,CP,CQ,NSVT,NSVTT)
COMPLEX PW,EP(1),EJ(1),API(1),CO(1),CP(1),CQ(1),B(1)
DIMENSION ETA(1)
DO 1 I1=1,N
DO 2 I2=1,N
I1I2=I1+(I2-1)*N
I1I2L=I1I2+NSVTT
2 CO(I1I2)=API(I1I2L)
I1L=I1+NSVT
1 CP(I1L)=B(I1L)
CALL CMPRI(CO,CP,CQ,N,N,1)
DO 3 I1=1,N
3 EP(I1)=(EJ(I1)+CQ(I1))*CEXP(PW*ETA(L))
RETURN
END

C
C   HER K ICIN J' NIN HESABI...
SUBROUTINE SEJ(L,PW,EJ,H,N,K,TO,EM,CO,CP,IR,NSVTT)

```



```

COMPLEX PW,EI(1),H(1),CO(1),CP(1)
DIMENSION EM(1),IR(1)
DO 22 I1=1,N
DO 22 I2=1,N
I12=I1+(I2-1)*N
I12L=I12+NSVTT
22 CO(I12)=CMPLX(-EM(I12L),0.)
DO 23 I1=1,N
I11=I1+(I1-1)*N
23 CO(I11)=CEXP(PW*TO)+CO(I11)
CALL CMINV(CO,CP,N,IR)
CALL CMPRD(CP,H,EI,N,N,1)
RETURN
END

C
C N(K,I) ' LERIN HESABI.
SUBROUTINE SEN(NSV,K,FL,EN,Q,P,Q)
DIMENSION FI(1),EN(1),C(1),P(1),Q(1),NSV(1)
DO 111 L=1,K
DO 111 J=1,K
IF(L.EQ.1.AND.J.NE.K) GO TO 10
IF(J.EQ.K.AND.L.NE.1) GO TO 20
IF(L.GT.2.AND.J.LE.(L-2)) GO TO 30
IF(L.NE.1.AND.L.NE.K.AND.J.GE.L.AND.J.LT.K)GO TO 40
DO 2 I=1,NSV(L)
DO 22 J1=1,NSV(L)
I1=I+(J1-1)*NSV(L)
22 Q(I1)=0
I=I+(I-1)*NSV(L)
2 Q(I)=1
GO TO 5
10 J1=J+1
FITOP=0
IF(J1.EQ.2) GOTO 333
DO 222 I=1,J1-2
222 FITOP=FITOP+NSV(I+1)*NSV(I)
333 DO 4 I1=J1,K
J11=J1+1
IF(J1.EQ.K) J11=1
FITOP=FITOP+NSV(I1)*NSV(I1-1)
DO 6 I2=1,NSV(J11)
DO 6 I3=1,NSV(J1)
I23=I2+(I3-1)*NSV(J11)
I23I1=I23+FITOP
6 Q(I23)=FI(I23I1)
IF(I1.NE.J1) GOTO 61
DO 66 I6=1,NSV(J1)
DO 67 I7=1,NSV(J1)
I6I7=I6+(I7-1)*NSV(J1)
67 P(I6I7)=0
I6I6=I6+(I6-1)*NSV(J1)
66 P(I6I6)=1
61 I1=I1+1
IF(I1.EQ.K) I11=1
N1=NSV(I11)
N2=NSV(I1)
N3=NSV(J1)
CALL GMPRD(Q,P,Q,N1,N2,N3)
DO 4 I2=1,N1
DO 4 I3=1,N3
I23=I2+(I3-1)*N1
4 P(I23)=Q(I23)
GO TO 5
20 LE=L-1
FITOP=0
DO 7 I1=1,LE
IF(I1.EQ.1) GOTO 77
FITOP=FITOP+NSV(I1)*NSV(I1-1)
77 DO 8 I2=1,NSV(I1+1)
DO 8 I3=1,NSV(I1)
I23=I2+(I3-1)*NSV(I1+1)
I23I1=I23+FITOP
8 Q(I23)=FI(I23I1)
IF(I1.NE.1) GOTO 13
DO 88 I8=1,NSV(1)
DO 89 I9=1,NSV(1)
I8I9=I8+(I9-1)*NSV(1)
89 P(I8I9)=0
I8I8=I8+(I8-1)*NSV(1)
88 P(I8I8)=1
13 N1=NSV(I1+1)
N2=NSV(I1)
N3=NSV(1)
CALL GMPRD(Q,P,Q,N1,N2,N3)
DO 7 I2=1,N1
DO 7 I3=1,N3
I23=I2+(I3-1)*N1
7 P(I23)=Q(I23)
GO TO 5
30 JA=J+1
LE=L-1
FITOP=0
IF(JA.EQ.2) GOTO 411
DO 422 I=1,JA-2
422 FITOP=FITOP+NSV(I+1)*NSV(I)
411 DO 9 I1=1,LE
FITOP=FITOP+NSV(I1)*NSV(I1-1)
DO 11 I5=1,NSV(I1+1)
DO 11 I6=1,NSV(I1)

```



```

I516=I5+(I6-1)*NSV(I1+1)
I51611=I516+FITOP
11 Q(I516)=F(I51611)
IF(I1.NE.JA) GOTO 12
DO 81 I6=1,NSV(JA)
DO 82 I9=1,NSV(JA)
I619=I6+(I9-1)*NSV(JA)
82 P(I619)=0
I618=I6+(I8-1)*NSV(JA)
81 P(I618)=0
12 N1=NSV(I1+1)
N2=NSV(I1)
N3=NSV(JA)
CALL GMPRI(O,P,Q,N1,N2,N3)
DO 9 I2=1,N1
DO 9 I3=1,N3
I2I3=I2+(I3-1)*N1
9 P(I2I3)=Q(I2I3)
GO TO 5
40 JA=J+1
FITOP=0
IF(JA.EQ.2) GOTO 433
DO 444 I=1,JA-2
444 FITOP=FITOP+NSV(I+1)*NSV(I)
433 DO 71 I1=JA,K
I1=I+1
IF(I1.EQ.K) I11=1
FITOP=FITOP+NSV(I1)*NSV(I1-1)
DO 17 I2=1,NSV(I11)
DO 17 I3=1,NSV(I11)
I2I3=I2+(I3-1)*NSV(I11)
I2I311=I2I3+FITOP
17 Q(I2I3)=F(I2I311)
IF(I1.NE.JA) GOTO 171
DO 172 I6=1,NSV(JA)
DO 173 I9=1,NSV(JA)
I6I9=I6+(I9-1)*NSV(JA)
173 P(I6I9)=0
I618=I6+(I8-1)*NSV(JA)
172 P(I618)=0
171 N1=NSV(I11)
N2=NSV(I1)
N3=NSV(JA)
CALL GMPRI(O,P,Q,N1,N2,N3)
DO 71 I2=1,N1
DO 71 I3=1,N3
I2I3=I2+(I3-1)*N1
71 P(I2I3)=Q(I2I3)
LE=L-1
FITOP=0
DO 131 I1=1,LE
IF(I1.EQ.1) GOTO 451
FITOP=FITOP+NSV(I1)*NSV(I1-1)
451 DO 14 I2=1,NSV(I1+1)
DO 14 I3=1,NSV(I1)
I2I3=I2+(I3-1)*NSV(I1+1)
I2I311=I2I3+FITOP
14 Q(I2I3)=F(I2I311)
N1=NSV(I1+1)
N2=NSV(I1)
N3=NSV(JA)
CALL GMPRI(O,P,Q,N1,N2,N3)
DO 131 I2=1,N1
DO 131 I3=1,N3
I2I3=I2+(I3-1)*N1
131 P(I2I3)=Q(I2I3)
5 J1=J+1
IF(J1.EQ.K) J1=1
LTOP=0
IF(L1.EQ.1) GOTO 72
DO 611 I1=2,L
611 LTOP=LTOP+NSV(I1-1)*NSV(J1)
72 JTOP=0
IF(J1.EQ.1) GOTO 51
DO 73 I=1,J-1
DO 73 I1=1,K
73 JTOP=JTOP+NSV(I1)*NSV(I+1)
51 DO 111 I1=1,NSV(L)
DO 111 I2=1,NSV(I1)
I1I2=I1+(I2-1)*NSV(L)
I1I2LJ=I1I2+LTOP+JTOP
EN(I1I2LJ)=Q(I1I2)
111 CONTINUE
RETURN
END
C
C F*EXP(A*T) NIN HESAB...
SUBROUTINE SFI(NSV,K,DI,T,A,F,FC,F1,O,P,Q,R)
DIMENSION T(1),A(1),F(1),FC(1),F1(1),O(1),P(1),
1Q(1),R(1),NSV(1)
NSVTT=0
NSVFT=0
FITOP=0
DO 3 L=1,K
IF(L.EQ.1) GOTO 7
NSVTT=NSVTT+NSV(L-1)*NSV(L-1)
FITOP=FITOP+NSV(L-1)*NSV(L)
7 DO 1 I=1,NSV(L)
DO 1 I1=1,NSV(L)

```

```

II=I+(I-1)*NSV(L)
II=J+NSVTT
1 Q(II)=A(II)*T(L)
N=NSV(L)
CALL EXPMT(O,P,N,DI,Q,R)
DO 5 II=1,NSV(L)
DO 6 I2=1,NSV(L)
II2=I1+(I2-1)*NSV(L)
II2L=II2+NSVTT
FIC(II2L)=F(II2)
5 F(II2L)=P(II2)
L1=L+1
IF(L.EQ.K) L1=1
IF(L.EQ.1) GOTO 8
NSVFT=NSVFT+NSV(L)*NSV(L-1)
8 DO 2 II=1,NSV(L1)
DO 2 I2=1,NSV(L)
II2=I1+(I2-1)*NSV(L1)
II2L=II2+FTTOP
F(II2L)=0.
DO 2 I3=1,NSV(L)
II3L=I1+(I3-1)*NSV(L1)+NSVFT
I32=I3+(I2-1)*NSV(L)
2 F(II2L)=F(II2L)+F(II3L)*P(I32)
3 CONTINUE
RETURN
END
C
M' NIN HESABL..
SUBROUTINE SEM(NSV,K,FI,EN,EM)
DIMENSION FI(1),EN(1),EM(1),NSV(1)
LTOP=0
LKTOP=0
NSVTT=0
DO 1 L=1,K
L1=L+1
IF(L.EQ.K) L1=1
IF(L.EQ.1) GOTO 8
NSVTT=NSVTT+NSV(L-1)*NSV(L-1)
LTOP=LTOP+NSV(L)*NSV(L-1)
LKTOP=0
DO 73 I4=1,L-1
DO 73 I5=1,K
73 LKTOP=LKTOP+NSV(I5)*NSV(I4+1)
8 DO 1 I=1,NSV(L)
DO 1 J=1,NSV(L)
II=I+(J-1)*NSV(L)+NSVTT
EM(II)=0.
DO 1 II=1,NSV(L1)
III=I+(II-1)*NSV(L)+LTOP+LKTOP
IIL=II+(I-1)*NSV(L1)+LTOP
1 EM(II)=EM(III)+EN(III)*F(II,II)
RETURN
END
C
(A-PI) ' NIN HESABL..
SUBROUTINE SAPI(PW,API,NSV,K,A,CO,CP,IR)
COMPLEX PW,API(1),CO(1),CP(1)
DIMENSION A(1),IR(1),NSV(1)
NSVTT=0
DO 2 L=1,K
IF(L.EQ.1) GOTO 15
NSVTT=NSVTT+NSV(L-1)*NSV(L-1)
15 DO 1 I3=1,NSV(L)
DO 1 I2=1,NSV(L)
I2I3=I2+(I3-1)*NSV(L)
I2I3L=I2I3+NSVTT
1 CO(I2I3)=CMPLX(A(I2I3L),0.)
DO 1 I4=1,NSV(L)
I4I4=I4+(I4-1)*NSV(L)
11 CO(I4I4)=CO(I4I4)*PW
N=NSV(L)
CALL CMINV(CO,CP,N,IR)
DO 2 I3=1,NSV(L)
DO 2 I2=1,NSV(L)
I2I3=I2+(I3-1)*NSV(L)
I2I3L=I2I3+NSVTT
2 API(I2I3L)=CP(I2I3)
RETURN
END
C
ASIMTOTIK KARARLIK TESTI..
SUBROUTINE SSMODE(NSV,K,INF,EM,Q,P,ISSM,IL)
DIMENSION EM(1),Q(1),P(1),ISSM(1),IL(1),NSV(1)
NSVTT=0
DO 7 L=1,K
IL(L)=1
ISSM(L)=1
IF(L.EQ.1) GOTO 15
NSVTT=NSVTT+NSV(L-1)*NSV(L-1)
15 DO 8 II=1,NSV(L)
DO 8 I2=1,NSV(L)
II2=I1+(I2-1)*NSV(L)
II2L=II2+NSVTT
8 Q(II2)=EM(II2L)
N=NSV(L)
DO 5 II=1,INF
CALL GMPRD(O,O,P,N,N,N)
ONORM=0.

```

```

DO 2 I=1,NSV(L)
DO 2 J=1,NSV(L)
II=I+(J-1)*NSV(L)
O(IJ)=P(IJ)
2 ONORM=ONORM+ABS(O(IJ))
IF(ONORM.LT.1.0) GO TO 7
5 CONTINUE
ISSM(L)=0
7 CONTINUE
WRITE(6,11) (II(L),L=1,K)
11 FORMAT/' ASIMTOTIK KARARLIK TESTININ SONUCU :/'
1' ISSM(L)=1 ISE ML ASIMTOTIK KARARLIDIR./
2' ISSM(L)=0 ISE ML ASIMTOTIK KARARLI DEGILDIR.//
3' L=,2013)
WRITE(6,12)(ISSM(L),L=1,K)
12 FORMAT' ISSM(L)=,2013)
RETURN
END
C
C   TEORIDEKI BJ ' NIN HESABI..
SUBROUTINE SBB(PW,BB,APL,NSV,K,T,B,F,G,FIC,CO,CP,CQ,CB1,CB2)
DIMENSION T(1),F(1),G(1),FIC(1),NSV(1)
COMPLEX PW,BB(1),API(1),CO(1),CP(1),CQ(1),CB1(1),CB2(1),B(1)
NSVT=0
NSVT=0
NSVFT=0
NSVGT=0
DO 4 J=1,K
IF(J.EQ.1) GOTO 12
NSVT=NSVT+NSV(J-1)
NSVTT=NSVTT+NSV(J-1)*NSV(J-1)
12 DO 2 II=1,NSV(J)
IIJ=II+NSVT
CB1(II)=P(IIJ)
DO 2 I2=1,NSV(J)
II2=II+(I2-1)*NSV(J)
II2J=II2+NSVTT
CP(II2)=API(II2J)
2 CQ(II2)=CMPLX(FIC(II2J),0.)
DO 3 II=1,NSV(J)
III=II+(II-1)*NSV(J)
3 CQ(III)=CQ(III).CEXP(PW*T(J))
N=NSV(J)
CALL CMPRD(CQ,CB1,CB2,N,N,1)
CALL CMPRD(CP,CB2,CB1,N,N,1)
J1=J+1
IF(J.EQ.K) J1=1
IF(J.EQ.1) GOTO 8
NSVFT=NSVFT+NSV(J)*NSV(J-1)
NSVGT=NSVGT+NSV(J)
8 DO 31 II=1,NSV(J1)
DO 31 I2=1,NSV(J)
II2=II+(I2-1)*NSV(J)
II2J=II2+NSVFT
31 CQ(II2)=CMPLX(F(II2J),0.)
N1=NSV(J1)
CALL CMPRD(CQ,CB1,CB2,N1,N,1)
CP(1)=CEXP(PW*T(J))
DO 32 II=1,NSV(J1)
IIJ=II+NSVGT
32 CB1(II)=CB2(II)+G(IIJ)*CP(1)
DO 4 II=1,NSV(J1)
IIJ=II+NSVGT
4 BB(IIJ)=CB1(II)
RETURN
END
C
C   H ' IN HESABI..
SUBROUTINE SHL(PW,H,BB,NSV,K,TO,T,ETA,EN,CB1,CB2)
COMPLEX PW,H(1),BB(1),CB1(1),CB2(1),D5
DIMENSION T(1),ETA(1),EN(1),NSV(1)
NSVT=0
DO 1 I=1,NSV(L)
1 H(I)=(0.,0.)
DO 6 J=1,K
D5=CEXP(PW*ETA(J))
J1=J+1
IF(J.EQ.K) J1=1
IF(J.EQ.1) GOTO 9
NSVT=NSVT+NSV(J)
9 DO 21 II=1,NSV(J1)
IIJ=II+NSVT
21 CB2(II)=D5*BB(IIJ)
IF(I.L.70,20,20)
70 D5=CEXP(PW*TO)
DO 22 II=1,NSV(J1)
22 CB2(II)=D5*CB2(II)
20 LTOP=0
IF(L.EQ.1) GOTO 72
DO 611 II=2,L
611 LTOP=LTOP+NSV(II-1)*NSV(J1)
72 JTOP=0
IF(J.EQ.1) GOTO 51
DO 73 I=1,J-1
DO 73 II=1,K
73 JTOP=JTOP+NSV(II)*NSV(I+1)
51 DO 30 II=1,NSV(L)
CB1(II)=(0.,0.)
DO 30 I2=1,NSV(J1)

```

```

      II2=I1+(I2-1)*NSV(L)
      II2LJ=II2+L*TOP+J*TOP
30  CBI(II)=CBI(II)+EN*(II2LJ)*CB2(I2)
      DO 8 I=1,NSV(L)
      8  H(I)=H(I)+CBI(I)
      DO 6 I=1,NSV(L)
      6  H(I)=CEXP(-PW*ETA(L))*H(I)
      RETURN
      END
C
C  H(P,T) ' NIN HESABL..
SUBROUTINE SHPT(TRNSF,M,L,PW,TM,EP,API,NSV,DI,ETA,A,B,C,D,O,P,Q
I,R,CBI,NSVT,NSVTT,NSOVT)
COMPLEX PW,EP(1),API(1),CBI(1),CC,TRNSF(1),B(1),D(1)
DIMENSION ETA(1),A(1),C(1),O(1),P(1),Q(1),R(1),NSV(1)
N=NSV(L)
NN=N*N
DO 20 I=1,NN
IL=I+NSVTT
20  O(I)=A(IL)*(TM-ETA(L))
      CALL EXPMT(O,P,N,DI,Q,R)
      CC=CEXP(-PW*TM)
      DO 21 I=1,N
      CBI(I)=(0.,0.)
      DO 21 J=1,N
      II=I+(J-1)*N
      IIL=J+NSVTT
      IJL=J+NSVT
21  CBI(I)=CBI(I)+P(II)*EP(J)*CC-API(IIL)*B(IJL)
      DO 22 II=1,M
      IIL=II+(L-1)*M
      TRNSF(II)=D(IIL)
      DO 22 I=1,N
      IIIL=II+(L-1)*M+NSOVT
22  TRNSF(II)=TRNSF(II)+O(IIIL)*CBI(I)
      RETURN
      END
C
C  DIGER ALTPROGRAMLAR
C
C  USTEL MATRIS HESABL..
SUBROUTINE EXPMT(Q,R,N,DI,Q,R)
DIMENSION O(1),P(1),Q(1),R(1)
NN=N*N
JE=0
30  AG=0.
      DO 31 I=1,NN
31  AG=AG+ABS(O(I))
      IF(AG.LE.1.) GO TO 41
      JE=JE+1
      B=2.0**JE
      DO 32 I=1,NN
32  O(I)=O(I)/2.0
      GO TO 30
41  DO 1 I=1,N
      DO 1 J=1,N
      II=I+(J-1)*N
      P(II)=0.
1  Q(II)=0.
      DO 2 I=1,N
      II=I+(I-1)*N
      P(II)=1.
2  Q(II)=1.
      IM=0
10  IM=IM+1
      CALL GMPRD(Q,O,R,N,N,N)
      PNORM=0.
      QNORM=0.
      DO 6 II=1,N
      DO 6 I2=1,N
      II2=I1+(I2-1)*N
      R(II2)=R(II2)/IM
      Q(II2)=R(II2)
      P(II2)=P(II2)+Q(II2)
      PNORM=PNORM+ABS(P(II2))
6  QNORM=QNORM+ABS(Q(II2))
      IF(PNORM.GE.(-0.1E-15).AND.PNORM.LE.0.1E-15) GO TO 10
      QF=QNORM/PNORM
      IF(QF.LE.D1) GO TO 20
      GO TO 10
20  IF(JE.EQ.0) RETURN
      DO 51 I=1,NN
51  O(I)=O(I)*B
      DO 53 JEM=1,JE
      DO 52 I=1,NN
      Q(I)=P(I)
52  R(I)=P(I)
      CALL GMPRD(Q,R,P,N,N,N)
53  CONTINUE
      RETURN
      END
C
C  KOMPLEKS MATRIS INVERSI HESABL..
SUBROUTINE CMINV(A,B,N,IR)
DIMENSION IR(1)
COMPLEX A(1),B(1)
DO 3 I=1,N
DO 3 J=1,N
II=I+(J-1)*N

```

```

3 B(IJ)=A(IJ)
  L=0
  DO 6 I=1,N
8 IR(I)=0
20 AX=0.0
  DO 14 I=1,N
    IF(I-IR(I))15,14,15
15 II=I+(I-1)*N
    AG=CABS(B(II))
    IF(AG-AX) 14,14,17
17 AX=AG
    K=I
14 CONTINUE
    KK=K+(K-1)*N
    B(KK)=1.0/B(KK)
    DO 1 I=1,N
      DO 1 J=1,N
        IF(L.EQ.K.OR.J.EQ.K) GO TO 1
        II=I+(I-1)*N
        IK=I+(K-1)*N
        KJ=K+(J-1)*N
        B(IJ)=B(II)*B(IK)*B(KK)*B(KJ)
1 CONTINUE
    DO 2 I=1,N
      IF(L.EQ.K) GO TO 2
      IK=I+(K-1)*N
      KI=K+(I-1)*N
      B(IK)=B(IK)*B(KK)
      B(KI)=B(KI)*B(KK)
2 CONTINUE
    IR(K)=K
    L=L-1
    IF(L.NE.N) GO TO 20
  RETURN
  END

```

```

C
C   KOMPLEKS MATRIS CARPIMI...
C   SUBROUTINE CMPRD(A,B,C,N,M,L)
C   COMPLEX A(1),B(1),C(1)
  DO 1 I=1,N
    DO 1 J=1,L
      II=I+(J-1)*N
      C(II)=(0.,0.)
      DO 1 K=1,M
        IK=I+(K-1)*N
        KJ=K+(J-1)*M
1 C(II)=C(II)+A(IK)*B(KJ)
  RETURN
  END

```

```

C
C   GENEL MATRIS CARPIMI...
C   SUBROUTINE GMPRD(A,B,C,N,M,L)
C   DIMENSION A(1),B(1),C(1)
  DO 1 I=1,N
    DO 1 J=1,L
      II=I+(J-1)*N
      C(II)=0.
      DO 1 K=1,M
        IK=I+(K-1)*N
        KJ=K+(J-1)*M
1 C(II)=C(II)+A(IK)*B(KJ)
  RETURN
  END

```

EK - B ÖRNEK 1 VE ÖRNEK 7 YE İLİŞKİN KULLANICI PROGRAMLARININ

LİSTELERİ

C ÖRNEK 1'E İLİŞKİN KULLANICI PROGRAMLARI

```
C
OPEN (6,FILE='CIKI.DAT',STATUS='NEW')
CALL PGEDAN
STOP
END
```

```
C
SUBROUTINE OKU1(SSN,NSV,NOV,NT,NF,NKT,NCE,CN,CSN,DI,NDT,IVS,NZV,
IZV,TTD,ICV,DELY,YREF)
DIMENSION NSV(1),NCE(1),NZV(1),ZV(1),ICV(1),DELY(1),YREF(1)
INTEGER CSN(1),CN
DI=1E-10
TTD=1./5000.
NDT=40
IVS=4
NT=400
NF=2
ICV(1)=1
DELY(1)=0.0001
YREF(1)=0.
SSN=2
NKT=0
NSV(1)=1
NSV(2)=2
NOV=3
NCE(1)=2
NCE(2)=2
CN=1
CSN(1)=2
NZV(1)=0
NZV(2)=1
ZV(1)=0.1E-04
RETURN
END
```

```
C
SUBROUTINE OKU2(A,B,C,D,F,G,DKM,DRM,GDM,SSCS,CS,FRQ,XD,TO,NSX)
DIMENSION A(1),C(1),F(1),G(1),DKM(1),DRM(1),NSX(1),FRQ(1),XD(1)
INTEGER GDM(1),SSCS(1),CS(1)
COMPLEX B(1),D(1)
TO=0.
NSX(1)=2
NSX(2)=1
NSX(3)=2
XD(1)=0.
XD(2)=0.
FRQ(1)=5000.
FRQ(2)=1E-10
A(1)=45000.
A(3)=45.E+06
A(4)=50.
A(5)=45000.
B(3)=(50.,0.)
B(5)=(50.,0.)
C(2)=1.
C(4)=1.
C(5)=0.001
C(7)=1.
C(11)=1.
C(15)=1.
C(16)=1.
C(17)=0.001
D(2)=(1.,0.)
D(6)=(1.,0.)
D(8)=(1.,0.)
D(12)=(1.,0.)
D(15)=(1.,0.)
D(18)=(1.,0.)
D(21)=(1.,0.)
D(24)=(1.,0.)
F(2)=1.
F(4)=1.
CS(1)=1.
DKM(1)=1.
DKM(4)=1.
DKM(13)=1.
DKM(15)=1.
DKM(18)=1.
DKM(27)=1.
SSCS(1)=1.
SSCS(2)=2
RETURN
END
```

```
C
COMPLEX FUNCTION U(JJ)
PI=3.14159285
A=10.*COS(PI/3.)
B=10.*SIN(PI/3.)
```

```

U=CMLPX(A,B)
IF(IJ.EQ.2) U=CMLPX(10.,0.)
RETURN
END

*****

C   ORNEK 7' YE ILISKIN KULLANICI PROGRAMLARI
C
OPEN (6,FILE='CIK7.DAT',STATUS='NEW')
CALL PGEDAN
STOP
END

C
SUBROUTINE OKU1(SSN,NSV,NOV,NT,NF,NKT,NCE,CN,CSN,DI,NDT,IVS,NZV,
1ZV,TIO,ICV,DELY,YREF)
DIMENSION NSV(1),NCE(1),NZV(1),ZV(1),ICV(1),DELY(1),YREF(1)
INTEGER,CSN(1),CN
DI=1E-10
NDT=100
IVS=5
NF=400
TIO=0.02
ICV(1)=1
ICV(2)=1
ICV(3)=1
DELY(1)=0.0001
YREF(1)=0.0
YREF(2)=0.0
YREF(3)=0.0
NF=3
SSN=4
NKT=0
NSV(1)=1
NSV(2)=1
NSV(3)=1
NSV(4)=1
NOV=5
NCE(1)=1
NCE(2)=2
NCE(3)=2
NCE(4)=2
CN=1
CSN(1)=4
NZV(1)=0
NZV(2)=1
NZV(3)=1
NZV(4)=1
ZV(1)=0.005
ZV(2)=0.011666
ZV(3)=0.016332
RETURN
END

C
SUBROUTINE OKU2(A,B,C,D,E,G,DKM,DRM,GDM,SSCS,CS,FRQ,X0,TU,NSX)
DIMENSION A(1),C(1),F(1),G(1),DKM(1),DRM(1),NSX(1),FRQ(1),X0(1)
COMPLEX B(1),D(1)
INTEGER, GDM(1),SSCS(1),CS(1)
TU=0.
NSX(1)=1
NSX(2)=1
NSX(3)=1
NSX(4)=1
A(1)=0.0
A(2)=1500.
A(3)=1500.
A(4)=1500.
B(4)=(100.,0.)
B(8)=(100.,0.)
B(12)=(100.,0.)
C(6)=1.
C(11)=1.
C(16)=1.
D(3)=(1.,0.)
D(9)=(1.,0.)
D(15)=(1.,0.)
D(17)=(1.,0.)
D(19)=-1.,0.)
D(20)=-1.,0.)
D(24)=(1.,0.)
D(30)=(1.,0.)
D(33)=(1.,0.)
D(37)=(1.,0.)
D(38)=-1.,0.)
D(40)=-1.,0.)
D(45)=(1.,0.)
D(46)=(1.,0.)
D(54)=(1.,0.)
D(57)=(1.,0.)
D(58)=-1.,0.)
D(59)=-1.,0.)
F(5)=1.
F(6)=1.
F(8)=1.
F(9)=1.
F(11)=1.
F(12)=1.
X0(1)=0.
FRQ(1)=50.

```

```
FRQ(2)=50.  
FRQ(3)=50.  
CS(1)=1  
DKM(1)=1.  
DKM(7)=1.  
DKM(12)=1.  
DKM(17)=1.  
DKM(19)=1.  
DKM(28)=1.  
DKM(29)=1.  
DKM(31)=1.  
DKM(40)=1.  
DKM(41)=1.  
SSCS(1)=1  
SSCS(2)=2  
SSCS(3)=3  
SSCS(4)=4  
RETURN  
END
```

```
C  
COMPLEX FUNCTION U(JJ)  
IF(JJ.NE.1) GOTO 10  
U=CMLPX(0.0,-100.0)  
GOTO 50  
10 A=0.886*100.  
B=0.5*100.  
IF(JJ.EQ.3) GOTO 20  
U=CMLPX(-A,B)  
GOTO 50  
20 U=CMLPX(A,B)  
50 RETURN  
END
```