

38961

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MİKROİŞLEMCI-KONTROLLU ANAHTARLANMIŞ-KAPASİTÖR
FİLTRELERİNİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Melih Cevdet İNCE

DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

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

(İmza)

(İmza)

(İmza)

Danışman

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

ÖZET

Doktora Tezi

MİKROİŞLEMÇİ - KONTROLLU ANAHTARLANMIŞ - KAPASİTÖR
FİLTRELERİNİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Melih Cevdet İNCE

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik -Elektronik Mühendisliği

Anabilim Dalı

1994, Sayfa: 154

Bu tezde, N-yollu filtreler ve uygulamaları olan darbantlı yüksek kalite faktörlü bant geçiren süzgeç ve tarak tipi filtreler üzerinde çalışılmıştır. Anahtarlanmış kapasitörlü N-yollu süzgecin tasarım problemine çözüm bulmak için, anahtar fonksiyonu ile filtrenin transfer karakteristiği arasındaki kesin benzerlik kullanılmıştır.

Farklı yaklaşımlar kullanarak farklı bilgisayar programları hazırlanmıştır. OPTI bilgisayar programında amaç fonksiyonunu optimize etmek için penaltı-fonksiyonu tekniği kullanılmıştır. Daha genel bir program olan NIKOP 'ta amaç fonksiyonunu optimize etmek için eşlenik gradyant yöntemi kullanılmıştır. GENKDI bilgisayar programıyla anahtarlama periyodunun aralık sayısına bağlı olarak mümkün olan bütün kombinasyonların

beklenen gerekleřtirme řartlarını saęlayıp saęlamadıęı kontrol edilmiřtir. Bu bilgisayar programlarının getirdięi yenilikleri gstermek iin farklı rnekler zerinde alıřılmıř ve nceden bilinen zmlerden daha iyi zmler elde edilmiřtir. Elde edilen bazı zmler mikroilemci-kontrollu anahtarlanmıř-kapasitr filtresi kullanılarak deneysel olarak kontrol edilmiřtir.

Bu alıřmada ayrıca darbe-geřiřlik modlasyonlu inverter kullanarak asenkron motorun kontrol edilmesinde, doęrultucu ve inverter arasındaki devrenin trnn etkileri incelenmiřtir. Bazen zel durumlarda ideal anahtarlama řeklinin src ve motor devrelerinin parametrelerine baęlı olduęu gsterilmiřtir.

ANAHTAR KELİMELER: Anahtarlanmıř-kapasitrler, N-yollu szgeler, elektronik olarak ayarlanabilen bant-geiren filtreler, tarak tipi filtreler, darbe geřiřlik modlasyonlu inverter.

SUMMARY**Ph. D. Thesis****DESIGN AND REALIZATION OF MICROPROCESSOR-CONTROLLED
SWITCHED-CAPACITOR FILTERS****Melih Cevdet İNCE****Fırat Üniversitesi****Graduate School of Natural and Applied Sciences****Department of Electrical and Electronic Engineering****1995, Page: 154**

In this thesis, the N-path filters and their applications to realize narrow-band, high-Q band pass filters as well as comb filters are investigated. To find a solution of design problem of switched-capacitor N-path filter, an exact analogy is used between key function and transfer characteristics of filter.

By using different approaches different computer programs are prepared, and their results are compared with each other's. In the computer program OPTI, to optimize the objective function, penalty-function technique is used. In the more general program NIKOP, conjugated gradient method is used to optimize the objective function. By the computer program GENKDI, depending on the number of the switching period, all possible combinations are tested as to satisfy the expected realizability conditions. To demonstrate

the innovations of these computer programs, different examples are worked out and results more satisfactory than the previously known ones are obtained. Some of these results are tested by using a microprocessor-controlled switched-capacitor filter.

Apart from the above, the effects of the type of the network between rectifier and inverter in the control of asynchronous motor by using pulse-width modulation are investigated in this thesis. It is shown that in some special cases the ideal switching shape depends on the parameters of the rectifier and motor circuits.

KEY WORD: Switched capacitors, N-path filters, electronically adjustable band-pass filters, comb filters, pulse-width modulated inverter.



TEŐEKKÜR

Bu alıőmanın ortaya ıkmasında yönetici ve yönlendirici olarak her zaman desteklerini saęlayan tez yöneticim Sayın Prof.Dr.Muhammet Köksal'a teőekkür ederim.

Ayrıca alıőmalarım süresince gerek yardım eden gerekse manevi yönden destek olan bölümdeki arkadaşlarıma, tezin deneysel kısımlarındaki yardımları için Sayın Yard.Do.Dr.Ahmet Arslan'a teőekkür ederim.

Son olarak, sabırla bana katlanan ve beni destekleyen aileme de minnet ve őükranlarımı sunarım.



İÇİNDEKİLER

ÖZET.....	III
SUMMARY.....	V
TEŞEKKÜR.....	VII
İÇİNDEKİLER.....	VIII
ŞEKİLLER LİSTESİ.....	XII
TABLolar LİSTESİ.....	XVI
SİMGELER LİSTESİ.....	XVII
1. GİRİŞ.....	1
2. GENEL N-YOLLU FİLTRELER.....	6
2.1. Giriş.....	6
2.2. Zamanla Değişen Sistem Fonksiyonu	7
2.3. Transfer Fonksiyonu.....	10
2.4. N- Yollu Filtre Uygulamaları.....	12
2.5. Komütatörlü Devreler.....	15

3. ANAHTARLANMIŞ N-YOLLU FİLTRELERDE TRANSFER FONKSİYONUNUN KONTROL EDİLİŞİ.....	24
3.1. Giriş.....	24
3.2. Modülasyon Veya Anahtar Fonksiyonu İle Transfer Fonksiyonu Arasındaki İlişki.....	24
3.3. Optimizasyon Probleminin Kuruluşu.....	26
3.4. OPTI Programının Tanıtımı.....	35
3.4.1. GTETA altprogramı ve değişken isimleri	38
3.4.2. AAI altprogramı ve değişken isimleri	38
3.4.3. GROBJ, NGROBJ altprogramları ve değişken isimleri	38
3.4.4. DUSAR altprogramı ve değişken isimleri	39
3.5. OPTI Programı İle Çalışılan Örnekler	40
3.6. NIKOP Programının Tanıtımı	42
3.6.1. GIR ve BASL altprogramları ve değişken isimleri	44
3.6.2. SUBF altprogramı ve değişken isimleri.....	44
3.6.3. SIRALA ve FARKLIF altprogramları ve değişken isimleri	45
3.6.4. GH altprogramı ve değişken isimleri	46
3.6.5. OBJECT ve NRML altprogramları ve değişken isimleri	47
3.6.6. OP altprogramı ve değişken isimleri	47
3.7. NIKOP Programı ile Çalışılan Örnekler	48
4. GENKDI PROGRAMININ TANITIMI.....	56
4.1. Giriş.....	56
4.2. Ana Program GENKDI ve Değişken İsimleri.....	57
4.3. YAZG Altprogramı.....	61
4.4. TAMAR Altprogramı.....	62

4.5.	CAK Altprogramı ve Değişken İsimleri	62
4.6.	ARORT Altprogramı ve Değişken İsimleri.....	63
4.7.	GH Altprogramı ve Değişken İsimleri.....	63
4.8.	NORMZ Altprogramı ve Değişken İsimleri.....	65
4.9.	OBJECT Altprogramı ve Değişken İsimleri.....	65
4.10.	YAZH Altprogramı.....	66
4.11.	DEGER Altprogramı ve Değişken İsimleri.....	66
4.12.	YAZI Altprogramı ve Değişken İsimleri.....	66
4.13.	ETRR Altprogramı.....	67
4.14.	YAZIE Altprogramı ve Değişken İsimleri.....	68
4.15.	DGETR Altprogramı ve Değişken İsimleri.....	68
4.16.	GENKDI Prgramı İle Bulunan Bazı Çözümler.....	69
5.	GERÇEKLEŞTİRİLEN DEVRE.....	78
5.1.	Giriş.....	78
5.2.	Önerilen Yöntem.....	78
5.3.	Mikrobilgisayarda Çalıştırılacak Programın Elde Edilişi.....	84
5.4.	Gerçekleştirme ile ilgili değerlendirmeler	91
6.	DARBE GENİŞLİK MODÜLASYONLU İNVERTERLE ASENKRON MOTORUN KONTROLUNDA ARA DEVRENİN ETKİSİNİN İNCELENMESİ.....	93
6.1.	Giriş.....	93
6.2.	Bulunmuş Olan Bir Çözüm ve Analizi.....	94
6.3.	Akım Ara Devresinin İncelenmesi	100
6.4.	Karma Ara Devresinin İncelenmesi	104
6.5.	Gerilim Ara Devresinin İncelenmesi	107
6.6.	Yüke Paralel Kondansatörün Etkisi	109

7. SONUÇLAR.....	114
KAYNAKLAR.....	116
EK-1 OPTİ BİLGİSAYAR PROGRAMININ LİSTESİ.....	120
EK-2 NIKOP BİLGİSAYAR PROGRAMININ LİSTESİ.....	129
EK-3 GENKDI BİLGİSAYAR PROGRAMININ LİSTESİ.....	134
EK-4 N-YOLLU FİLTRELERİN TRANSFER FONKSİYONUNUN HESAPLANABİLMESİ İÇİN KULLANICININ YAZMASI GEREKEN ALTPROGRAMLAR	143
EK-5 6802 MIKROBİLGİSAYARINDA PROGRAM YAZILIMINDA KULLANILAN BAZI KOMUTLAR VE İŞLEMELERİ İÇİN GEREKEN SÜRELER	144
EK-6 MIKRO BİLGİSAYAR PROGRAMININ LİSTESİ.....	145
EK-7 İKİNCİ ARALIK DÜZENİNDE $N=2$, $M=3$, $N_Y=7$ VE 11, 12, 18, 24, 25, 46, 47, 53, 59, 60 ANAHTAR FONKSİYONU İÇİN MIKRO PROGRAMININ ÇIKTISI.....	149
EK-8 BÖLÜM 6'DAKİ DEVRELERDE YÜK GERİLİMİ VE AKIMINI HESAPLAMAK İÇİN MAINLN PROGRAMINDA KULLANICININ YAZMASI GEREKEN ALTPROGRAMLAR	150
EK-9 DISCRETE FOURIER ANALİZİ YAPAN D BİLGİSAYAR PROGRAMININ LİSTESİ.....	152

ŞEKİLLER

Şekil 2.1	: N-yollu bir süzgecin blok çizimi.....	7
Şekil 2.2	: N-yollu bir filtrenin transfer özellikleri a) giriş, b) çıkış işaretinin frekans spektrumları.....	10
Şekil 2.3	: a) Alt devrelerin, b)N-yollu dar bantlı yüksek Q'lu bant geçiren bir süzgecin transfer fonksiyonu.....	13
Şekil 2.4	: a) Alt devrelerin, b)N-yollu tarak tipi bir süzgecin transfer fonksiyonu...	15
Şekil 2.5	: a) Anahtarlarla gerçekleştirilmiş bir N-yollu süzgecin blok çizimi	19
Şekil 2.6	: a) İlkel bir N-yollu dur-geç filtre, b) anahtarlama fonksiyonu, c) transfer fonksiyonu.....	20
Şekil 3.1	: a) I/2 tek, b) I/2 çift durumları için anahtar fonksiyonları.....	29
Şekil 3.2	: a) N-yollu süzgecin devre şeması, b) TÇA sayısının azaltılması, c) çeyrek peryotta 1.yolun çalıştığı aralıklar	31
Şekil 3.3	: $R=5 \text{ K}\Omega$, $C=0.2 \text{ }\mu\text{F}$, 1000 Hz anahtarlama frekansı ve Şekil 3.2 c'deki anahtarlama fonksiyonu için Şekil 3.3 a veya b'deki devrenin transfer fonksiyonunun değişimi.....	33
Şekil 3.4	: OPTI programının akış şeması.....	37
Şekil 3.5	: a) $N=3, M=7, NY=15$, b) $N=4, M=9, NY=19$ için OPTI programının bulduğu anahtar fonksiyonlarının $R=5 \text{ K}\Omega$, $C=0.2 \text{ }\mu\text{F}$ için Şekil 3.2 b'de verilen filtrede ürettikleri transfer fonksiyonları.....	42
Şekil 3.6	: NIKOP ana programı tarafından kullanılan altprogramlar ve çağırma ilişkileri.....	43
Şekil 3.7	: NIKOP bilgisayar programının genel akış şeması.....	46

Şekil 3.8	: Anahtar fonksiyonu olarak. a) +10, -19, b) -10, +19, c) -10, -29, +48, +64 d) 13, -26, 51, 64, 89, -102, 127, 140 aralıklarının kullanılması durumunda elde edilen seyrek dişli tarak tipi karakteristikler.....	50
Şekil 3.9	: Her bir yolun bir kez devreye alındığı klasik uygulamayla elde edilen 19-yollu süzgeç karakteristiği.....	51
Şekil 3.10	: a) -6, 17, -28, 39 ve NY=11, b) -14, 41, -68, 95 ve NY=27 için elde edilen transfer karakteristiği.....	52
Şekil 3.11	: a) -8, -15, 30, 37, -52, -59, 74, 81 b) 8, -15, 30, -37, 52, -59, 74, -81, NY=11 ve $f_0=1000$ Hz için elde edilen transfer karakteristikleri.....	53
Şekil 3.12	: a) -10, 27, -28, 35, -44, 67, -76, 83, -90, 99, NY=11 b) -15, 44, -73, 102, -131, 160, -189, 218, -247, 276, NY=29 ve $f_0=1000$ Hz için elde edilen transfer karakteristikleri.....	53
Şekil 3.13	: a) 15,-36,-54,-67,83,-110,-118,139,-152,181 anahtarlama fonksiyonu için 19-yollu süzgeç transfer karakteristiği.....	55
Şekil 4.1	: GENKDI programının akış şeması.....	59
Şekil 4.2	: GENKDI tarafından kullanılan altprogramlar ve çağırma ilişkileri.....	60
Şekil 4.3	: Şekil 3.4'deki çözüme alternatif olarak a) birinci aralık düzeninde, b) ikinci aralık düzeninde bulunan çözümler için transfer fonksiyonlarının değişimi.....	70
Şekil 4.4	: N=3, NY=15, M=7 ve ikinci düzen için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği, a) 20, b) 15, c) 10, d) 5 dB zayıflama referansları için.....	72
Şekil 4.5	: N=3, NY=15, M=7 ve birinci düzen için Şekil 4.4'deki transfer karakteristiklerini üreten anahtarlama fonksiyonlarına alternatif olarak bulunan fonksiyonları kullanarak elde edilen süzgeç karakteristiği.....	73
Şekil 4.6	: N=4, NY=19, M=9 için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği.....	74

Şekil 4.7	: N=3, NY=19, M=9 için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği.....	75
Şekil 4.8	: N=3, NY=23, M=11 için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği.....	76
Şekil 4.9	: N=3, NY=9, M=5 için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği.....	76
Şekil 5.1	: 7 -yollu anahtarlanmış kapasitörlü süzgeç devresi.....	79
Şekil 5.2	: K kontrol girişiyle birlikte ters çeviren anahtar (TÇA).....	79
Şekil 5.3	: HEF 4066' nın a) fonksiyonel, b) bacak, c) şematik diyagramları.....	81
Şekil 5.4	: NOR kapıları ile gerçekleştirilen dış osilatör devresi.....	82
Şekil 5.5	: Dış osilatörün 6802 mikroişlemcisine bağlantısı.....	83
Şekil 5.6	: Düzeltme ve kuvvetlendirme devresi.....	83
Şekil 5.7	: Birinci yolun çalıştığı aralıklar.....	84
Şekil 6.1	: Evirici kutup geriliminin genel şekli.....	94
Şekil 6.2	: Ele alınan çözüm için evirici kutup geriliminin şekli.....	95
Şekil 6.3	: a) Sistemin tek fazlı blok gösterimi, b) Karma ara devre ile sistemin açık gösterimi.....	96
Şekil 6.4	: a) Akım ara devreli, b) karma ara devreli, c) akım ara devreli ve yük tarafında kondansatör, d) gerilim ara devreli, e) gerilim ara devreli ve yük tarafında kondansatör.....	97
Şekil 6.5	: Şekil 6.2 'de gösterilen gerilimin harmonik spekturumu.....	100
Şekil 6.6	: a) Şekil 6.2'deki gerilimin $R_y = 3 \Omega$, $L_y = 0.015$ H'lik yükten 50 Hz'de akıttığı akımın şekli, b) harmonik spekturumu.....	101
Şekil 6.7	: a) $R_s=0.5\Omega$, $L_s=0$ H, $R_y=3\Omega$, $L_y=0.015$ H için yük gerilimi, b) gerilimin harmonik spekturumu, c) yük akımı, d) akımın harmonik spekturumu.....	102
Şekil 6.8	: a) $R_s=0$, $L_s=0.0025$ H için aynı yükün üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu.....	103

- Şekil 6.9 : a) $R_S=0$, $L_S=0.1$ H için aynı yükün üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu..... 104
- Şekil 6.10 : a) $R_S=0.1$, $L_S=0.1$ H, $C=15.852$ μ F için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu..... 105
- Şekil 6.11 : a) $R_S=0.1$, $L_S=0.1$ H, $C=776.795$ μ F için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu..... 106
- Şekil 6.12 : a) $R_S=0.1$, $L_S=0.1$ H için aynı yükün üzerindeki gerilim, b) gerilimin harmonik spekturunu, c) yük akımı(süreksiz), d) yük akımının harmonik spekturumu..... 107
- Şekil 6.13 : a) $R_S=0$, $L_S=0$ için 30 Hz'de aynı yükten geçen akımın şekli, b) akımın harmonik spekturumu 108
- Şekil 6.14 : a) $R_S=1\Omega$, $R_Y=3\Omega$, $L_Y=0.015$ H ve $C=1000$ μ F için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu..... 109
- Şekil 6.15 : a) $R_S=0.1\Omega$, $L_S=0.1$ H, $R_Y=3\Omega$, $L_Y=0.015$ H ve $C=776.795$ μ F için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu..... 110
- Şekil 6.16 : a) 30 Hz çalışma frekansında Şekil 6.15'deki değerler için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu..... 111
- Şekil 6.17 : a) $R_S=0.1\Omega$, $L_S=0.1$ H, $R_Y=3\Omega$, $L_Y=0.015$ H ve $C=15.852$ μ F için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu..... 112
- Şekil 6.18 : a) $R_S=0.5\Omega$, $R_Y=3\Omega$, $L_Y=0.015$ H ve $C=1000$ μ F için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu. 113

TABLOLAR

Tablo 3.1	: İlk periyot boyunca bütün anahtarların çalıştığı aralıklar.....	34
Tablo 5.2	: HEF 4066B'nin bazı önemli tipik karakteristikleri.....	81
Tablo 5.3	: Her bir yoldaki anahtarların çalıştığı aralıklar.....	85
Tablo 5.4	: Mikrobilgisayarın çıkış portuna yazdırılacak ikili kodlanmış 16 tabanlı sayıların elde edilişi.....	86
Tablo 5.5	: A portunu çıkış olarak tanımlayan program.....	87
Tablo 5.6	: Her bir aralıkta çıkış portunun alacağı değeri doğrudan yükleyen program.....	87
Tablo 5.7	: Tablo 5.6'daki programın aralık sürelerinin eşitlenmiş olduğu program...	88
Tablo 5.8	: Ana program.....	89
Tablo 5.9	: Ana programa gecikme altprogramının eklenmesi.....	90
Tablo 6.1	: Şekil 6.4'deki devrelerin TÇA konumuna göre durum ve çıkış denklemlerindeki katsayı matrisleri.....	98
Tablo EK-5	: 6802 mikrobilgisayarında program yazılımında kullanılan bazı komutlar ve işlemleri için gereken süreler.....	144

SİMGELER

$H(s)$: Transfer fonksiyonu.
$u(t)$: Giriş (uyarı) fonksiyonu.
$y(t)$: Çıkış (cevap) fonksiyonu.
T	: Peryot.
f	: Frekans.
$p(\cdot), q(\cdot)$: Modülasyon fonksiyonları.
R	: Direnç.
$SI(t)$: Anahtarlama fonksiyonu.
$k(\theta)$: İmpulslar cinsinden anahtarlama fonksiyonu.
C	: Kapasitans.
C_n	: Fourier katsayıları.
L	: İndüktans.
ϕ	: Amaç (object) fonksiyonu.

1. GİRİŞ

Sayısal işaretlerin işlendiği devreler bir yana bırakılırsa, periyodik olarak çalışan anahtar (POÇA) ları içeren devreler üç ana bölümde toplanabilir.

1. Yüksek frekanslarda çalışan periyodik anahtarları içeren devreler.
2. N-yollu devreler.
3. Özsalınumlu geçiş devreleri.

Bu bölümlerden ilkinde örnek olarak dur-geç filtre (süzgeç)ler ve yavaşla-geç süzgeçler verilebilir (Köksal, 1979). Günümüz elektroniğinde zamanla değişmeyen transfer fonksiyonlarının gerçekleştirilmesi yüksek frekanslarda çalışan periyodik anahtarları içeren doğrusal (lineer) devrelerle de yapılabilmektedir (Trick, 1973). Böyle anahtarlar kullanılmasıyla zamanla değişmeyen bir devrenin dinamiği bir bölümüyle yada tümüyle değiştirilebilir. Bu olgu, uygulamada, çevre şartlarına duyarsız tümleşik devrelerin gerçekleştirilmesinde etkin şekilde kullanılır (Sun ve Frish, 1968). Ayrıca devre içlerindeki yerlerine ve çalışma biçimine bağlı olarak, bu anahtarlar bazı devre elemanlarının etken değerini değiştirirler ve böylece iyi bir elektronik ayarlama ve tam bir frekans ölçeklemesi sağlarlar. Tümleşik devre tasarımının göze çarpan zorluklarından birisi de büyük değerli direnç ve kapasitörlerin yapımı, diğer bir deyişle büyük zaman sabitlerinin gerçekleştirilmesidir. Bu zorluğu gidermek için çeşitli yöntemler kullanılmıştır. İlk kez Lin (Lin, Davis ve Sun, 1966) tarafından ortaya atılan bir yol, her dirence seri bir anahtar bağlamaktır. Bu durumda etken direnç, anahtar periyodunun, anahtarın bir periyot boyunca kapalı kalma süresine oranı kadar artar. Tümleşik devrelerde anahtarlama yöntemi ile direnç çarpılması Sun ve Frish tarafından ayrıntılı olarak incelenmiştir. Daha sonra dirençlere ek olarak, jirator iletkenliklerinin, bağımlı kaynaklar ve eksi empedans çeviricileri gibi elemanların geçiş katsayılarında çarpılabileceğini gösterdiler ve bunu

deneysel olarak uyguladılar (Hirano ve Nishimura, 1972). Periyodik anahtarlama kullanılarak gerçekleştirilen bu çarpım sayesinde transfer fonksiyonunda sürekli bir ayarlama sağlanmış olur. Hızlı çalışan periyodik anahtarların önemli bir başka uygulaması da çok kanallı transfer fonksiyonlarının gerçekleştirilmesinde görülür (Kraehler, 1969). Bu uygulamada ise yükselteçler ve bu tür anahtarları içeren aktif RC devrelerindeki diğer elemanlar zaman paylaşmalı olarak her kanal tarafından ortaklaşa kullanılır (Bruton ve Pederson, 1972). Fakat bu paylaşım her kanala ilişkin transfer fonksiyonlarının ayarlama alanını kısıtlar.

N-yollu süzgeçler ise ilk defa 1960 yıllarında Franks ve Sandberg tarafından bulunmuş ve bunların dar bantlı yüksek kalite faktör (Q) lü bant geçiren filtrelerin, tarak tipi filtrelerin ve zaman gecikmeli devrelerin tasarımında nasıl kullanılabileceği belirtilmiştir (Franks ve Sandberg, 1960). Bu yolla gerçekleştirilen bir dar bantlı bant geçiren süzgecin pasif(edilgin) LC elemanlar veya aktif (etkin) RC devreleri kullanarak gerçekleştirilen benzer süzgeçlerle karşılaştırıldığında aşağıdaki üstünlükleri olduğu görülmüştür (Glaser ve Halkias, 1969).

1. Bant genişliğinin ve merkez frekansının elektronik olarak ayarlanabilir olması,
2. Yüksek Q 'sünün olması,
3. Oldukça kararlı olması,
4. En önemli parametrelerinin eleman değerlerine bağlı olmaması.

Yukarıda üçüncü bölüm olarak verilen ve zaman paylaşmalı sistemlerde çok yaygın uygulaması olan ve kesikli zaman işaretleri ile sürekli zaman işaretlerinin birbirlerine çevrilmesinde kullanılan özsalınımlı geçiş devreleri bu çalışmanın kapsamı dışında tutulmuştur. Bu çalışmada yukarıda ikinci bölüm olan N-yollu devreler ele alınmıştır. Esas ağırlık ise böyle devrelerin periyodik olarak çalışan anahtarlar kullanılarak gerçekleştirilmesiyle ortaya çıkan komütatörlü devrelere verilmiştir. Özellikle komütatörlü devrelerin iki farklı uygulamasını oluşturan dar bantlı, yüksek Q 'lü bant geçiren filtrelerle tarak tipi süzgeçlerin tasarımı ile ilgilenilmiştir.

Modülasyon tipi N-yollu filtrelerin transfer fonksiyonunun, modülasyon fonksiyonlarının katsayılarına bağımlılığı (Franks ve Dann, 1970; Glaser, Halkias ve Meadows, 1969; Franks ve Sandberg, 1960) açıkça ortaya koyulmuştur.

Periyodik olarak çalışan anahtarları içeren doğrusal devrelerin analizi ve sentezi bunların uygulamaları ile birlikte incelenmiş böyle sistemlerin genel bir teorisi ve uygulaması verilerek, POÇA içeren devrelerin formulasyonu ve çözümü için bilgisayar programlamasına uygun yeni yöntemler sunulmuştur (Köksal, 1975). Bir periyot içinde her yolu bir kaç defa çalıştırarak ve ters çeviren anahtarlar (TÇA) (change-over switches) kullanarak, anahtarlı N-yollu filtrelerin band geçiren filtre olarak daha kullanışlı bir uygulaması gösterilmiştir. Bu uygulamada kullanılması gereken 3 değerli (-1, 0, +1) modülasyon teorisi anlatılmıştır. Bu çalışmadan sonra yine Köksal tarafından (Köksal, 1984) periyodik olarak çalışan anahtarları içeren lineer sistemlerin genel çözümlerini yapan tam bir analiz programı (MAINLN) tanıtılmıştır. Bu programla herhangi bir periyodik olarak çalışan anahtarlanmış devre, bir periyottaki bölümlerin sayısına bakılmaksızın kolaylıkla analiz edilebilir. Burada temel fikir periyodik olarak zamanla değişen parametrelere adımlar halinde, zamanla değişen fonksiyonlar yaklaşımını yapmaktır.

Anahtarlanmış kapasitörlü N-yollu süzgeçlerde (AKNYS) transfer fonksiyonunun kontrol edilmesini gösterirken MAINLN isimli bilgisayar programının getirdiği kolaylıklar da gözönüne alınarak klasik uygulama durumlarının (Fetweis, 1971; Wupper, 1971 (a); Wupper, 1971 (b); Wupper, 1974; Sun ve Frish, 1969; Heinlein, 1971; Langer, 1968; Ryken, 1970) aksine her bir periyot boyunca tek bir darbe yerine, anahtarlama fonksiyonunda darbe sıraları kullanılabilir. Bu görüşle Köksal tarafından 9-yollu, dar bantlı, yüksek Q'lu, bant geçiren anahtarlanmış kapasitörlü bir filtre tasarlanmıştır. MAINLN'le bu süzgecin analizi yapılarak 4.5 kHz'den küçük frekanslarda geçerli olan ve 1000 Hz'lik anahtarlama frekansına yerleşmiş merkez frekansı olan transfer fonksiyonu elde edilmiştir. Burada tasarım için temel görüşler ortaya konulmakla birlikte prosedür sistematik ve genel bir hale getirilmemiştir. Bu görüşler ışığında, AKNYS devresinden istenilen karakteristiği verecek

anahtarlama fonksiyonunun elde edilmesi için genel bir yol araştırılmıştır. Köksal tarafından işaret edildiği yönde ve problemin ele alınış biçimlerine göre bilgisayar programları yazılmış ve bunların sonuçları karşılaştırılmıştır.

İkinci bölümde N-yollu filtreler hakkında genel bilgiler verilmiştir. Bu bölümde zamanla periyodik olarak değişen sistemlerin bir bölümü olan N-yollu süzgeçler tanıtılmış ve bu süzgeçler yardımıyla zamanla değişmeyen transfer fonksiyonunun nasıl gerçekleştirildiği gösterilmiştir. Ayrıca N-yollu devrelerin periyodik çalışan anahtarlar kullanılarak yapımı ele alınmış, bunlarla ilgili uygulamalar belirtilmiştir. Uygulamaların sonucunda N-yollu süzgeçlerin aynı nitelikte özellikleri veren pasif LC elemanlar veya aktif RC devreleri kullanarak gerçekleştirilenlere göre daha kararlı ve ucuz, boyutlarının ise (özellikle zaman geciktiren devre olarak kullanıldığında) küçük olduğu görülmüştür. Ancak buraya dek sözü edilen iyi yanlarına karşın şunları da belirtmekte yarar vardır. Bu tip süzgeçlerin yapımı bindiriciler veya anahtarlar gibi değişik elemanların kullanımını gerektirmektedir. Bu elemanlar tümleşik devre olarak yapılabildiğinden bu bir sorun değildir, fakat bu elemanları kontrol edecek modülatör veya anahtarlama fonksiyonlarını üretmek için ek devrelere ihtiyaç duyulur. Ayrıca dar bantlı bant geçiren süzgeç olarak uygulamada tepe frekansı için geniş bir ayarlama alanı isteniyorsa bu fonksiyonlar ve ek devreler daha karmaşık biçimler alırlar.

Üçüncü bölümde ise anahtarlanmış kapasitörlü N-yollu süzgeçlerde transfer fonksiyonunun kontrol edilişi dayandığı bağıntılar verilerek anlatılmış, önce AKNYS devresinden sadece anahtarlama fonksiyonunun frekansına yerleşmiş merkez frekansı olan dar bantlı, yüksek kalite faktörlü bant geçiren filtre transfer fonksiyonunun elde edilmesi amaçlanarak gerekli prosedür açıklanmıştır. Bu prosedüre göre çalışan ve tasarımı, lineer olmayan denklemleri kullanarak yazılan amaç (object) fonksiyonunun optimizasyonu olarak ele alan bilgisayar programı OPTI tanıtılmıştır. OPTI programı ile bazı örnekler çalışıldıktan sonra, yine bir optimizasyon programı olan ve bazı sınırlandırmalar gevşetilerek AKNYS devresinden en genel halde istenen transfer fonksiyonunu veren bir

bilgisayar programı, NIKOP tanıtılmıştır. Bu programlarla bulunan bazı sonuçlar, birbirleriyle ve Bölüm 4'deki GENKDI ile karşılaştırmalı olarak verilmiştir.

Dördüncü bölümde, önce, üçüncü bölümdeki sonuçlar gözönüne alınarak ve buradaki şartlar genişletilmeden, istenen transfer fonksiyonunun üretilmesini sağlayacak olan anahtarlama fonksiyonunu bulan öcekilerden farklı bir bilgisayar programı tanıtılmıştır. GENKDI adı verilen bu program, tasarımı yapılan süzgecin seçilmesi gereken yol sayısı ve bir periyotta bir yolun kaç defa devreye alınacağına bağlı olarak olabilecek bütün kombinasyonları tek tek deneyerek istenen şartları sağlayıp sağlamadığını kontrolunu yapar.

Beşinci bölümde, önce, mikrobilgisayar kullanarak gerçekleştirilmiş olan AKNYS devresi ve bu devreyle merkez frekansının elektronik olarak nasıl değiştirildiği anlatılmış, sonra OPTI, NIKOP veya GENKDI ismi verilen bilgisayar programları ile bulunan çözümlerin bu devrede çalıştırılması ile alınan sonuçlar karşılaştırılmıştır. Bu bölümün kısımlarında anlatıldığı gibi, bazı kısıtlamalardan dolayı uygulamada 7-yollu süzgeçle yetinilmiş ancak bu biçimdeki tasarım için öneriler yapılmıştır. Bu bölümde kullanılan bir küçük bilgisayar programı (MIKRO) ayrıntılı olarak anlatılmamış, ancak listesi ve bir uygulama için çıktısı ek olarak verilmiştir.

Altıncı bölümde bu tezdeki konularla yakınlığı gözönüne alınarak, darbe genişlik modülasyonlu inverterle bir asenkron motorun kontrolü için yapılan çalışmada ara devrenin etkisi incelenmiştir. Bu inceleme sırasında, tezin diğer kısımlarında da kullanılan bazı bilgisayar programlarında kullanıcının yazması gereken altprogramların listesi ek olarak verilmiştir.

Yedinci bölümde, teze ilgili sonuçlar özetle verilmiştir. Yapılanların getirdiği kolaylıklardan, uygulamadaki sınırlamalardan ve daha sonra yapılacak çalışmalara ışık tutacak tavsiyelerden bahsedilmiştir.

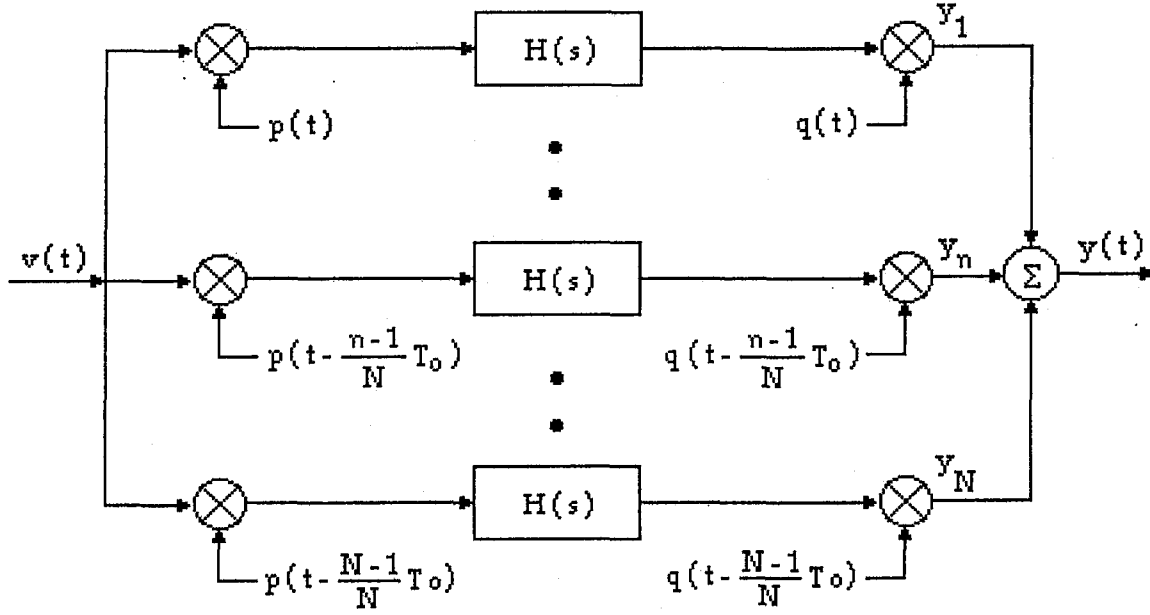
Tezin sonunda ise hazırlanan bilgisayar programların listeleri ve tezin izlenmesini kolaylaştıracak ekler verilmiştir.

2. GENEL N-YOLLU FİLTRELER

2. 1. Giriş

Bant genişliği ve merkez frekansı elektronik olarak ayarlanabilir, yüksek kalite faktörlü, kararlı ve en önemli parametreleri devrenin eleman değerlerine duyarlı bant geçiren filtre elde etmek için kullanılan bir N-yollu süzgecin blok çizimi Şekil 2.1'de verilmiştir.

Görüldüğü gibi N-yollu bir süzgeç N tane birbirinin aynı alt devreden oluşur. $H(s)$ bu alt devrelerin transfer fonksiyonu (geçiş işlevi) nu gösterir. Her alt devre bir giriş modülatörü (bindiricisi) tarafından uyarılır ve çıkışı da diğer bir modülatöre ulaşır. Giriş modülatörlerinin hepsi tek bir işaret $u(t)$ ile uyarılır. Çıkış bindiricileri ise bir toplayıcıya bağlandığından tek bir çıkış işareti $y(t)$ üretir. Tüm modülatörler periyotları aynı (T_0) olan fonksiyonlarla sürülür. Önemli olan nokta giriş ve çıkış bindiricilerini süren $p(.)$ ve $q(.)$ işlevlerinin ardışık bindiricilere T_0/N gibi bir zaman gecikmesiyle ($2\pi/N$ kadar bir evre farkıyla) uygulanmış olmasıdır. Bu gerçek aşağıda görüleceği gibi N-yollu bir süzgecin çalışma esasını oluşturur.



Şekil 2.1 N-yollu bir süzgecin blok çizimi

2.2. Zamanla Değişen Sistem Fonksiyonu

Genelde N-yollu süzgeçler, zamanla değişen doğrusal sistemlerin bir bölümüdür. Bu nedenle N-yollu bir süzgecin $u(t) = U e^{st}$ (w gerçel frekans ise $s = jw$) üssel girdisine olan kalıcı tepkisi

$$Y_{ss}(t) = H(s, t) U e^{st} \quad (2.1)$$

denklemlerle belirlenir (Zadeh, 1950). Burada

$$H(s, t) = \frac{Y_{ss}(t)}{u(t)} \Big|_{u(t) = U e^{st}} \quad (2.2)$$

olup süzgecin zamanla değişen sistem (devre) fonksiyonudur. Dikkat edilirse zamanla değişen sistem fonksiyonunun tanımı zamanla değişmeyen ve doğrusal devreler için kullanılan transfer fonksiyonunun tanımı gibidir. Tek fark sistem fonksiyonunun daha genel

olup, kompleks frekansa (s) ek olarak zamana (t) da bağlı olmasıdır. Zamanla T_0 periyodu ile değişen bir sistemin sistem fonksiyonu $H(s,t)$ 'de t zamanına göre T_0 periyodu ile periyodiktir (Darlington, 1955). Onun için böyle bir sistemin sistem fonksiyonunun

$$H(s, t) = \sum_{m=-\infty}^{\infty} H_m(s) e^{j\omega_0 m t} \quad , \quad \omega_0 = \frac{2\pi}{T_0} \quad (2.3)$$

biçiminde Fourier dizisi açılımı vardır. Bu açılım zamanla periyodik olarak değişen sistemlerin transfer özelliklerini tanımakta çok yararlıdır. Örneğin girişi $u(t) = U e^{j\omega t}$ olan zamanla periyodik olarak değişen bir sistemin çıkışı, denklem 2.1 ve 2.3'den

$$y(t) = H(j\omega, t) u e^{j\omega t} = \sum_{m=-\infty}^{\infty} H_m(j\omega) u e^{j(\omega + m\omega_0) t} \quad (2.4)$$

olarak bulunur. Buradan açıkça görülmektedir ki, giriş bir tek frekansı (ω) içerdiği halde çıkış bir çok frekansı ($\omega + m\omega_0$, $m = 0, \mp 1, \mp 2, \dots$) içermektedir. Bu frekanslardan yalnız bir tanesi giriş işaretinin frekansı olan ω 'ya eşittir. Diğerleri ise bu frekanstan $m\omega_0$ ($m = 0, \mp 1, \mp 2, \dots$) uzaklıkta bulunan ve ω_0 aralıklarıyla görülen görüntü frekanslarıdır. $H_m(s)$ Fourier katsayısı da giriş işareti ile çıkış işaretinin $\omega + m\omega_0$ frekansındaki bileşeni arasındaki zamanla değişmeyen transfer fonksiyonudur. Şekil 2.1'de görünen N-yollu süzgecin zamanla değişen sistem fonksiyonunu bulmak için p ve q fonksiyonlarının Fourier açılımlarından başlanır.

$$p(t) = \sum_{k=-\infty}^{\infty} P_k e^{j\omega_0 k t} \quad (2.5)$$

$$q(t) = \sum_{l=-\infty}^{\infty} Q_l e^{j\omega_0 l t}$$

Doğrusal devre kuramı kullanılarak $u(t) = U e^{st}$ için $y_n(t)$ bulunur. Sonra $y(t) = y_1 + y_2 + \dots + y_N$ ve 2.1, 2.2 denklemlerinden $H(s,t)$ elde edilir. Bulunan sonuç, 2.4 denklemiyle karşılaştırıldığında sistem fonksiyonunun Fourier katsayılarının,

$$H_m(s) = \sum_{k=-\infty}^{\infty} Q_{m-k} P_k H(s+jw_0 k) \sum_{n=1}^N e^{-j \frac{n-1}{N} 2\pi m} \quad (2.6)$$

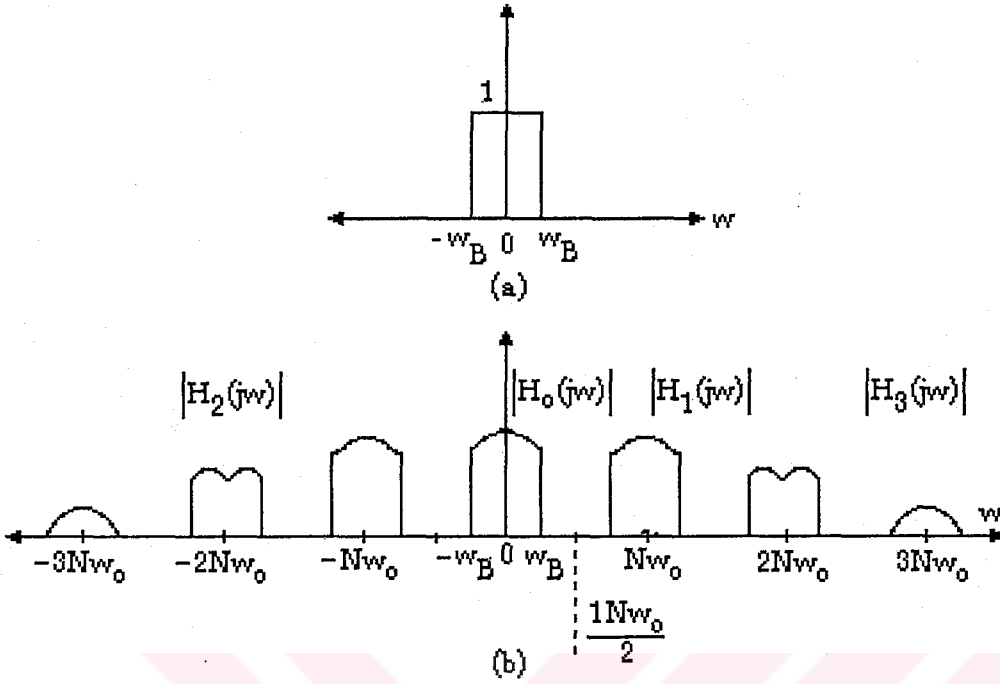
olduğu görülür. Bu denklemde ikinci toplam $m = lN$ ($l = 0, \pm 1, \pm 2, \dots$) için N 'ye eşit olup m 'nin diğer değerleri için sıfırdır. Başka bir deyişle,

$$H_m(s) = \begin{cases} N \sum_{k=-\infty}^{\infty} Q_{lN-k} P_k H(s+jw_0 k) & : m=lN, l=0, \pm 1, \pm 2, \dots \\ 0 & : \text{diğer } m\text{'lerin için} \end{cases} \quad (2.7)$$

Yukarıdaki sonuç N -yollu bir süzgecin çalışma ilkesini açıklaması bakımından çok önemlidir. $H_m(s)$, $e^{j\omega t}$ girişini çıkışının $(w+m\omega_0)$ görüntü frekansındaki bileşinine bağlayan transfer fonksiyonu olduğundan açıkça görülmektedir ki çıkışın w ile $w+N\omega_0$ frekansları arasında beklenen bileşenleri kaybolmuştur. 2.3 ve 2.7 denklemlerinde anlaşılacağı gibi zamanla değişen sistem fonksiyonu $H(s,t)$ zamana göre T_0 periyodu ile değil T_0/N periyodu ile değişmektedir. Diğer bir deyişle w frekansında bir giriş işareti çıkışta frekansları yalnız $w+l(N\omega_0)$, $l = 0, \pm 1, \pm 2, \dots$, olan görüntü bileşenleri yaratır. Diğer görüntü bileşenleri alt devrelerin birbirinin aynı ve modölatörlerin düzenli, eşit zaman farklarıyla sürülmesi nedeniyle kaybolmuşlardır. Bu nedenle uygulamada, zaman farkları arasındaki uyum ve alt devreler arasındaki benzerlik oldukça gereklidir.

Birbirine eşit ve düzenli zaman farklarını gerçekleştirebilmek bazı tip $p(\cdot)$ ve $q(\cdot)$ modülasyon fonksiyonları için daha kolay olmaktadır. Bunlara yeri geldikçe değinilecektir. Alt devreler arasındaki benzerliğin gerçekleştirilmesi sorunu ise sözde N -yollu süzgeçlerde (pseudo- N -path filters) çözülmüştür (Fetweis ve Wupper, 1971). Bu çeşit süzgeçler bu tezin kapsamı dışında tutulmuştur.

Yukarıdaki satırlarda anlatılmaya çalışılan N -yollu bir süzgecin transfer özelliklerini daha iyi kavramak için Şekil 2.2'yi incelemek gerekir.



Şekil 2.2 N-yollu bir filtrenin transfer özellikleri a) giriş, b) çıkış işaretinin frekans spektrumları

Bu şekilde giriş işaretininin bant sınırlı beyaz bir gürültü olduğu varsayılarak çıkış işaretinin frekans spektrumu çizilmiştir. Görüldüğü gibi giriş işaretinin spektrumu çıkışa her görüntü bileşeni için değişik biçimde yansımakta ve bu değişiklik o görüntü bileşenine karşı gelen Fourier katsayısıyla oluşmaktadır.

2.3. Transfer Fonksiyonu

N-yollu yollu bir süzgeçten doğrusal ve zamanla değişmeyen bir devre olarak yararlanmak için iki değişik yaklaşım vardır.

1. Bant sınırlı modülatör fonksiyonları kullanmak: Eğer $p(\cdot)$ ve $q(\cdot)$ modülatör fonksiyonlarının frekans bantları, l 'nin $N/2$ 'den büyük veya eşit değerleri için $P=Q=0$ ise 2.6 denkleminle belirlenmiş olan Fourier katsayıları m 'nin sıfırdan farklı değerleri için hep sıfırdır. Dolayısıyla çıkışta yalnız, frekansı giriş işaretinin frekansına eşit bir işaret görülür. 2.3 denkleminde de zamanla değişen sistem fonksiyonu $H(s,t)$ 'nin zamanla değişmeyen bir transfer fonksiyonun, $H_0(s)$ 'ye eşit olduğu ortaya çıkar. Bunun için zamanla periyodik olarak değişen N -yollu süzgeç, transfer fonksiyonu

$$H_0(s) = \sum_{k=-\infty}^{\infty} Q_{-k} P_k H(s+j\omega_0 k) \quad (2.8)$$

olan zamanla değişmeyen bir devreye eşdeğer olur. Bundan ötürü $H_0(s)$ N -yollu bir süzgecin transfer fonksiyonu olarak tanımlanır. Bant sınırlı modülatör fonksiyonları olarak ilk aklı gelen sinüs eğrisi biçimli fonksiyonlardır. Ancak bu tip fonksiyonların düzenli ve eşit zaman farklarıyla kaydırılması bir sorun olduğundan ya basamak tipi fonksiyonlar tercih edilir (Glasier ve Halkias, 1969), ya da ilerideki kısımlarda anlatılacak olan komütatörlü devreler kullanılır.

2. Bant sınırlı giriş ve çıkış işaretleri kullanmak : Eğer modülatör fonksiyonları bant sınırlı değilse, zamanla değişen sistem fonksiyonu $H(s,t)$ 'nin Şekil 2.2'de görülen Fourier katsayıları genel olarak sıfırdan farklıdır. Bu durumda farklı görüntü bileşenlerine ilişkin çıkışların çakışmaması için giriş frekansının bant genişliği olan ω_B 'nin $N\omega_0/2$ 'den az olması gereklidir. Bu nedenle N -yollu süzgeçler N 'yi gerektiği kadar büyük seçmekle istenildiği kadar geniş bir frekans bandında (0 ile $N\omega_0/2$ arasında) kullanılmak üzere tasarlanabilir. Tasarımı bitmiş ve gerçekleştirilmiş N -yollu bir filtre, eğer giriş işareti frekansı $N\omega_0/2$ 'den büyük bileşenleri içeriyorsa, kesim frekansı $N\omega_0/2$ olan alçak geçiren diğer bir süzgecin ardına bağlanmalıdır. Diğer taraftan genellikle çıkış işaretinin yalnız giriş işareti frekansında olan bileşeni istendiğinden diğer frekanslarda bulunan görüntü bileşenleri de kesim sıklığı $N\omega_0/2$ olan başka bir alçak geçiren süzgeçle atılır. Bu ikinci süzgeçte N -yollu süzgecin ardına bağlanır. Böylece iki alçak geçiren süzgecin arasında

çalışan N-yollu süzgecin transfer özellikleri transfer fonksiyonu $H_0(s)$ olan ve zamanla değişmeyen bir devreninkinin aynısı olur. Bu nedenle $H_0(s)$, bu koşullar altında çalışan N-yollu bir süzgecin transfer fonksiyonu olarak tanımlanır.

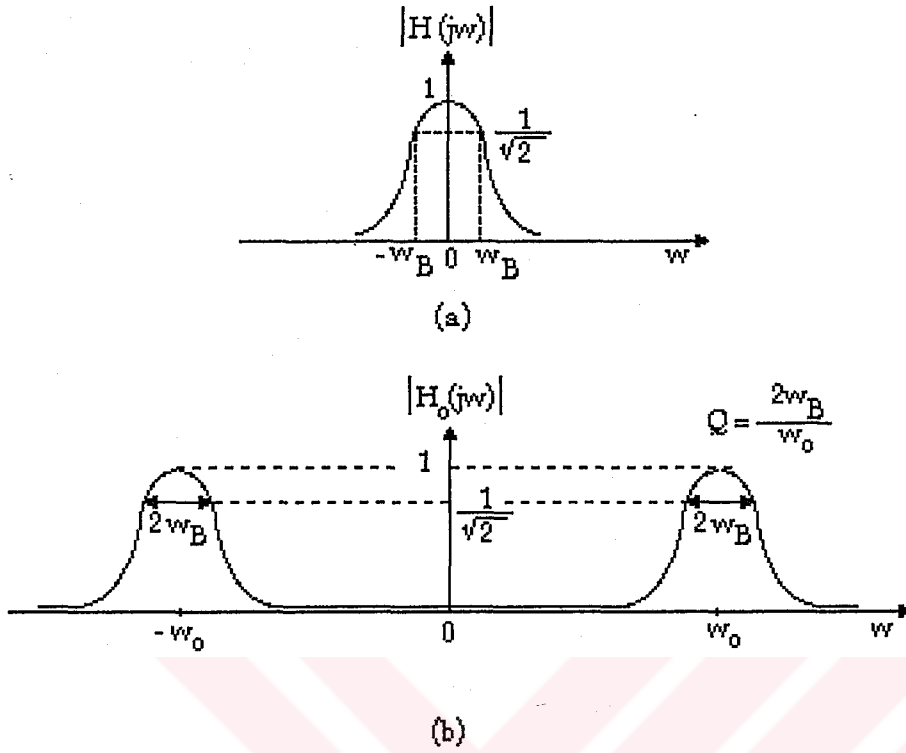
2. 4. N - Yollu Filtre Uygulamaları

2.8 denklemleri, N-yollu bir süzgecin transfer fonksiyonu olan $H_0(s)$ 'nin bu süzgecin alt devrelerine ilişkin $H(s)$ transfer fonksiyonunun $\omega_0 k$ ($k = 0, \pm 1, \pm 2, \dots$) frekanslarına kaydırılıp $Q_k P_k$ ağırlıklarıyla çarpıldıktan sonra toplanmalarıyla bulunduğunu gösterir. $p(\cdot)$ ve $q(\cdot)$ modülatör fonksiyonlarının Fourier katsayılarına bağlı olarak $H_0(s)$ transfer fonksiyonu çeşitli özellikler gösterebilir.

1. $Q_{-1} P_1 = Q_1 P_{-1} = 1/N$ ve $k \neq 1$ için $Q_{-k} P_k = 0$ ise

$$H_0(s) = H(s + j\omega_0) + H(s - j\omega_0) \quad : \quad s = j\omega \quad (2.9)$$

Şekil 2.3'de $H(s)$ ve $H_0(s)$ alçak geçiren bir alt devre için gösterilmiştir. Bu şekilde veya doğrudan doğruya 2.9 denkleminde anlaşılacağı gibi alt devrelerin alçak geçiren süzgeçler olmasına rağmen, N-yollu devrenin kendisi bir bant geçiren süzgeçtir. Bu bant geçiren süzgecin merkez frekansı ω_0 olup alt devrelerin parametrelerine bağlı olmadığından oldukça kararlıdır. Ayrıca süzgecin Q'su $Q = \omega_0 / 2\omega_B$ olduğundan devre elemanlarının değerlerini değiştirmeden ω_0 'ı çoğaltmakla istenildiği kadar büyük yapılabilmektedir. Böylece N-yollu devre yapısıyla yüksek Q'lu, dar bantlı, merkez frekansı ayarlanabilir ve oldukça kararlı süzgeçlerin gerçekleştirilmesi mümkün olmaktadır.



Şekil 2.3 a) Alt devrelerin, b)N-yollu dar bantlı yüksek Q'lu bant geçiren bir süzgecin transfer fonksiyonları

Eğer alt devreler, RC alçak geçiren süzgeç yerine birer RC yüksek geçiren süzgeç olarak seçilirse, o zaman $H(s)$ yüksek geçiren bir karakteristik gösteriyorken $H_o(s)$ 'in bant söndüren bir karakteristiği olacaktır. Böylece oluşan N-yollu süzgeç belirli bir frekans bandında değişmeyen, ancak f_0 frekansında ve bu frekansın katlarında çentikleri bulunan bir taraktipi çentik süzgeç (comb-notch filter) oluşturur. Periyodik olmayan işaretler periyodik bir gürültüyle karıştığında, uygun bir tarak tipi çentik süzgeç kullanılarak işaret üzerinde önemli olmayan bir bozulma karşılığında, bu işaret periyodik gürültüden arındırılabilir (Temes ve Mitra,1973), ancak çentiklerin çok dar olması gerekmektedir. Özel bir örnek olarak Hareketli Hedef Gösterici (HHG) (moving-target-indicator,MTI) radarı verilebilir. Burada, çentik tipi tarak süzgeç durgun hedeflerden dönen zamana göre periyodik işaretleri bastırır(Urkowitz,1957). Bu durumda çentikler, HHG'nin tarayıcı anteninin dönüş

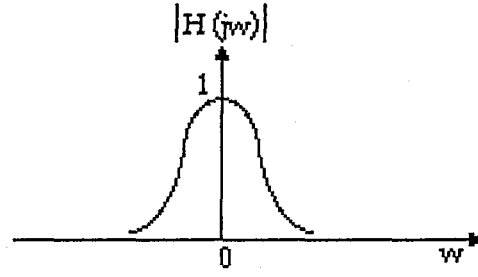
frekansının katlarına yerleştirilmelidirler. Tarak-tipi çentik süzgeçlerle ilgili daha fazla bilgi (Franks, 1973) ve (Acompara, 1966)'dan edinilebilir.

2. $Q_k P_k = 1/N$ ise

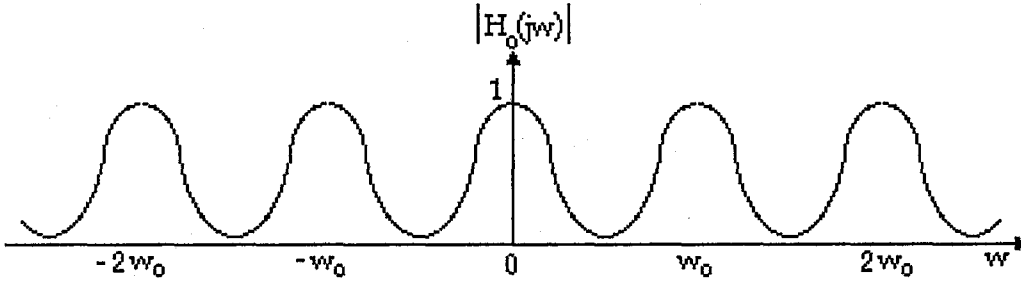
$$H_0(s) = \sum_{k=-\infty}^{\infty} H(s+jw_0 k) \quad , \quad s=jw \quad (2.10)$$

Şekil 2.4'de $H(s)$ ve $H_0(s)$ alçak geçiren bir alt devre için gösterilmiştir. Bu şekilden anlaşılacağı gibi yine alt devrelerin alçak geçiren süzgeçler olmasına karşı N-yollu devrenin kendisi tarak tipi bir süzgeci (comb filter) gerçekleştirmektedir. Tarak tipi transfer fonksiyonunun varlığı ise ancak bant genişliği w_B 'nin $w_0/2$ den küçük veya eşit değerlerinde mümkündür. Böyle süzgeçler de arka plandaki periyodik olmayan bir gürültüden periyodik işareti seçmek için kullanılırlar.

Yukarıdaki açıklamalardan sonra anlaşılıyor ki tarak tipi periyodik süzgeç özelliği yalnız zaman gecikmeli elemanların veya sayısal filtrelerin kullanılması ile değil N-yollu süzgeçlerle de elde edilebilir. Üstelik N-yollu devrelerin kullanılması çoğu zaman kararlılık ve bu boyut açısından daha uygun olur. N-yollu devreler tarak tipi çentik süzgeçlerin, zaman geciktiren elemanların, frekans modülatörleri ve daha bir çok elektronik sistemin gerçekleştirilmesinde de kullanılır (Franks ve Dann,1973; Franks ve Dann,1970; Chang, Declaris ve Newcomb,1973).



(a)



(b)

Şekil 2.4 a) Alt devrelerin, b)N-yollu tarak tipi bir süzgecin transfer fonksiyonları

2. 5. Komütatörlü Devreler

Şekil 2.1'de görülen N-yollu bir süzgecin gerçekleştirilebilmesi için gerçek bir çarpma işlemi yapabilen modülatörlere ihtiyaç vardır. Günümüz elektroniklerinde böyle modülatörlerin tümleşik örneksel çarpıcılar kullanılarak yapımı mümkün olmaktadır (Bilotti, 1968). Ancak bu çarpıcılar kullanıldıklarında $p(\cdot)$ ve $q(\cdot)$ fonksiyonlarının gereken transfer özelliklerini elde edecek biçimde üretilmesi ve çoğu zaman modülatörlerin kendileri maddi sorun olmaktadır. Diğer taraftan modülatörler yerine periyodik olarak çalışan anahtarlar (POÇA) kullanılabilir. Anahtarların FET 'ler kullanılarak kolayca gerçekleştirilmesi (Gulbenk, 1964, Shipley, 1964, Orfei ve Pallottino, 1972) ve özen

gerektirmeyen kare dalgalarla bile normal çalışabilmeleri N-yollu süzgeçlerin çarpıcılar yerine periyodik anahtarlar kullanılarak yapımını daha yaygın kılar.

Şekil 2.5'de modülatörler yerine anahtarlar kullanılarak yapılmış N-yollu bir süzgecin blok çizimi gösterilmiştir. $S(t)$ fonksiyonu birinci yolda bulunan anahtarların çalışma durumunu gösterir. $S(t)=0$ ise anahtarlar açık, $S(t)=1$ ise anahtarlar kapalı demektir. Dikkat edilirse ikinci yolda bulunan anahtarların durumu birinci yolda bulunanlarınkini T_0 / N gibi bir zaman gecikmesiyle kovalar. Aynı bağıntı diğer yollara ilişkin anahtarlar arasında da vardır. Diğer taraftan $S(t)$ öyle bir fonksiyondur ki herhangi bir anda en çok yalnız bir alt devre giriş ve çıkış arasına bağlı kalabilir. Örneğin t anında giriş ve çıkış arasında yalnız birinci alt devre bulunuyorsa $t_1 + T_0 / N$ anında yalnız ikincisi, $t_1 + 2T_0 / N$ anında yalnız üçüncüsü, ..., $t_1 + (N-1)T_0 / N$ anında da sonuncusu bulunur. Böylece her alt devre giriş ve çıkış arasına sırasıyla bağlanır. Bu yüzden böyle devrelere komütatörlü devreler denir. Yalnız burada kullanılan ve elektronik olarak gerçekleştirilen komütatörler elektrik makinalarında çok yaygın olan mekanik komütatörlerle karıştırılmamalıdır.

Aslında komütatörlü devreler, Şekil 2.5'de görülenden çok daha geneldir. Bunların en genelden en özele doğru güzel bir sıralaması (Köksal, 1975)'de yapılmıştır. Şekil 2.5'de görülenden daha genel bir komütatörlü devreye örnek olarak şu devreler verilebilir.

1. Çıkış anahtarlarını kontrol eden fonksiyon $S'(t)$, giriş anahtarlarını kontrol eden $S(t)$ den farklıdır. Bu durumda her alt devre 4 farklı durumda bulunabilir ; yalnız giriş anahtarı kapalı, yalnız çıkış anahtarı kapalı, her iki anahtar kapalı, her iki anahtar açık.

2. Transfer fonksiyonları $H(s)$ ile belirtilen tek girişli ve tek çıkışlı (çift kapılı), alt devreler çok girişli ve çok çıkışlı (çok kapılı) dır. Bu durumda giriş ve çıkış anahtarlarının da çok kapılı olması gerekir.

3. Alt devrelerin kendileri de zamanla periyodik olarak değişir. Bu durumda $H(s)$ transfer fonksiyonu $H(s,t)$ zamanla değişen sistem fonksiyonuna çevrilir.

Yukarıda gösterilen genel durumların hiç birinde değişik yollarda bulunan alt devreler arasında etkileşim yoktur. Bu ise $S(t)$ ve $S'(t)$ fonksiyonlarını belirli bir şekilde seçmekle sağlanır.

Komütatörlü devreler, periyodik olarak çalışan anahtarları içeren doğrusal devrelerin özel bir bölümü olduğundan, böyle devreler için geliştirilmiş yöntemlerle çözümlenebilir (Köksal ve Tokad, 1977; Köksal ve Tokad, 1976). Ancak yalnız komütatörlü devreler için geliştirilmiş çözümlene yöntemleri de vardır (Sun ve Frish, 1969). Şu önemli noktaları da sıralamakta fayda vardır. Komütatörlü bir devre olarak gerçekleştirilmiş N-yollu bir süzgecin çalışma ilkesi modülatörler kullanarak gerçekleştirilen N-yollu bir süzgecininki ile aynıdır. Zamanla değişen sistem fonksiyonları aynı özellikleri gösterir ve transfer fonksiyonu $H_0(s)$ aynı şekilde tanımlanır. Aradaki tek fark, komütatörlü süzgeçte zamana bağlı olmayan transfer fonksiyonu $H_0(s)$ 'in 2.3 kısmında belirtilen yöntemlerden yalnız ikincisi ile elde edilebilir olmasıdır. Komütatörler kesikli çalıştıklarından bant sınırlı modülatör fonksiyonlarının görevini yüklenemezler. Komütatörlü devrelerin en yaygın uygulaması N-yollu dur-geç süzgeçlerdir. Bu süzgeçlerin genel yapısı Şekil 2.5'dekinin aynıdır. Yalnız $H(s)$ transfer fonksiyonu ile gösterilen alt devrelerin her biri bir dur-geç süzgeçtir (Köksal, 1976). Bu alt devreler girişle çıkış arasında bağlandıklarında geç durumunda diğer zamanlarda ise dur durumunda kalırlar. Bundan dolayı $H(s)$ yerine $H(s/d)$ alınır, d anahtarların çalışma çarpanı (duty cycle) olup, geç zamanının T_0 periyoduna oranıdır .

N-yollu dur-geç süzgeçler en güzel şekilde Fetweis tarafından incelenmiştir (Fetweis, 1969). Fetweis'in sonuçları diğer araştırmacılarınkinden hem oldukça farklı hem de tasarım yönünden o derece daha yararlıdır. Fakat bu sonuçlar bir periyot boyunca her alt devresi yalnız bir kere giriş ile çıkış arasına bağlanan bir N-yollu dur-geç süzgeç için geçerlidir. Böyle bir durumda $S(t)$ anahtarlama fonksiyonu Şekil 2.6.b'de olduğu gibidir.

Fetweis'in sonuçlarına göre yukarıda adı geçen süzgeç için 2.7 denklemi,

$$H_m(s) = \begin{cases} \sum_{k=-\infty}^{\infty} S_{1N,k} H\left(\frac{s+jw_0 k}{d}\right) & : m=1N, l=0, \pm 1, \pm 2, \dots \\ 0 & : \text{diğer m'ler için} \end{cases} \quad (2.11a)$$

şeklini alır. Burada $S_{1N,k}$; $Q_{1N-k} P_k$ yerine geçmekte olup,

$$S_{1N,k} = Nd \frac{\sinh\left[\frac{(1-d)T_0}{2} s + jk\pi\right] \sinh\left[\frac{(1-d)T_0}{2} (s - j1Nw_0 \frac{d}{1-d}) + jk\pi\right] e^{-jw_0 lNd}}{\left[\frac{(1-d)T_0}{2} s + jk\pi\right] \left[\frac{(1-d)T_0}{2} (s - j1Nw_0 \frac{d}{1-d}) + jk\pi\right]} \quad (2.11.b)$$

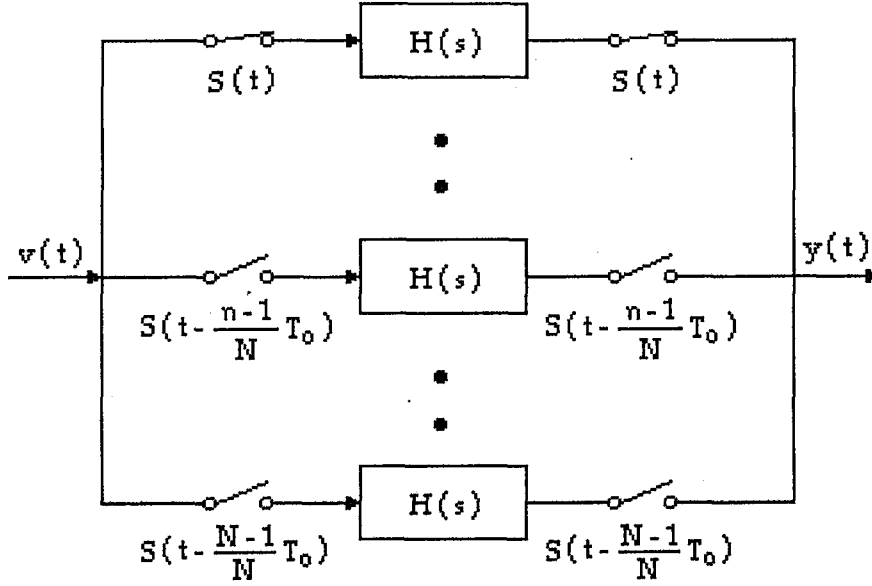
biçiminde yazılabilir. Özel olarak $m=0$ için yukarıda $H_m(s)$ için yazılmış olan denklem ,

$$H_0(s) = N \sum_{k=-\infty}^{\infty} d \left[\frac{\sin\left[(1-d) \frac{w}{w_0} + k\pi\right]}{\left[(1-d) \frac{w}{w_0} + k\pi\right]} \right]^2 H\left(\frac{s+jw_0 k}{d}\right) \quad (2.12)$$

denklemine dönüşür.

2.12 denklemi d parametresine bağlı olduğundan N -yollu dur-geç süzgeçler (Köksal, 1976)'de anlatılmış olan dur-geç süzgeçler gibi, d parametresi ile ayarlanabilir transfer fonksiyonlarını oluştururlar. Ancak ayarlama alanı dur-geç süzgeçlerde olduğu gibi geniş değildir. Çünkü d 'nin alabileceği en yüksek değer, her anahtarın kapalı kalma zamanı en fazla T_0/N olabileceğinden, $1/N$ olup yol sayısı ile sınırlanmıştır.

Eğer N -yollu süzgeci oluşturan alt devrelerin her biri kesim frekansı $Nw_0/2$ den küçük alçak geçiren süzgeç ise, 2.12 denklemi tarak tipi bir süzgeç özelliği verir.

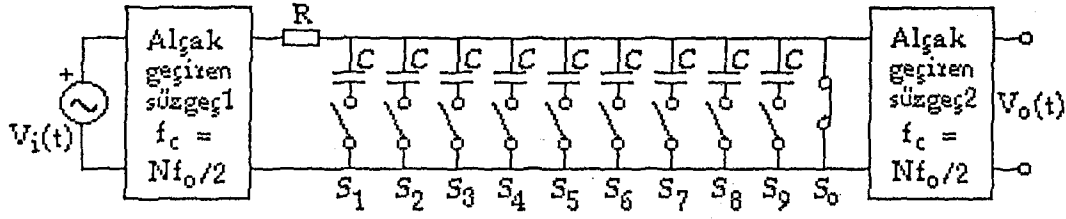


Şekil 2.5 Anahtarlarla gerçekleştirilmiş bir N-yollu süzgecin blok çizimi

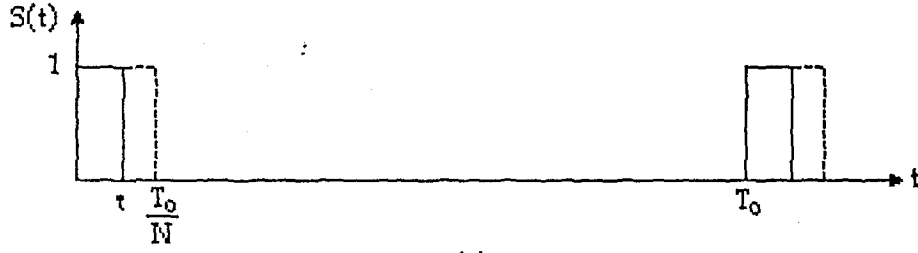
Bu özelliğin, N-yollu bir süzgecin transfer fonksiyonunun tanımındaki gerçeklerden dolayı, yalnız $0 \leq w < Nw_0/2$ bandı için geçerli olduğu hatırlanmalıdır. Böyle bir süzgece örnek olarak Şekil 2.6.a'da gösterilen dur-geç süzgeç verilebilir. Bu örnekte her alt devre transfer fonksiyonu,

$$H(s) = (1 / RC) / (s + 1 / RC) \quad (2.13)$$

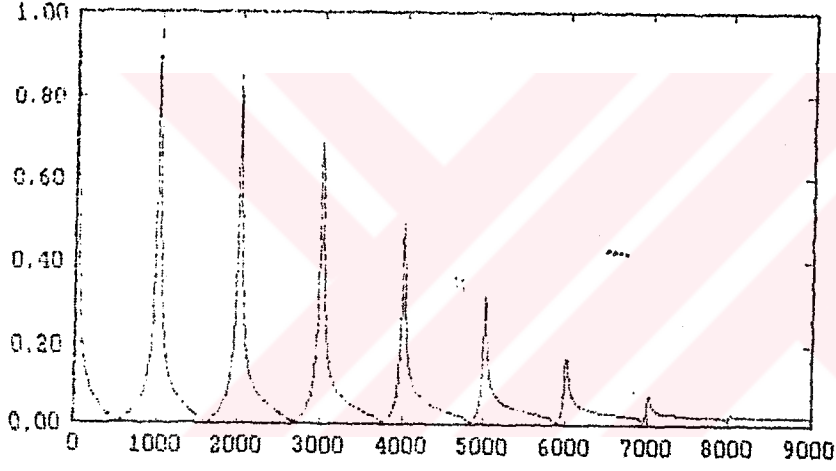
olan yalın bir RC alçak geçiren süzgeç olarak seçilmiştir. Özenle incelendiğinde Şekil 2.5'de gösterilen her yola ilişkin bir çift anahtarın (giriş ve çıkış anahtarları) görevini tek bir anahtarın yaptığı görülür. Ayrıca bu anahtar RC alt devresini bir dur-geç süzgeç olarak çalıştırma görevini de yüklenmiştir. Diğer taraftan R direnci her alt devrede zaman paylaşmalı olarak kullanılmıştır. S_0 anahtarı da çıkış işaretini alt devrelerden hiç biri çıkışa bağlı değilse sıfırlamak için konulmuştur.



(a)



(b)



(c)

Şekil 2.6 a) İkel bir N-yollu dur-geç filtre, b) anahtarlama fonksiyonu, c) transfer fonksiyonu

Yukarıdaki açıklamaların ışığında Şekil 2.6.a'da görülen N-yollu dur-geç süzgeçte bazı elemanların çok yönlü görevler yüklendiği görülmektedir. Bu ise her alt devrenin yalnız bir RC süzgeç olmasından ileri gelmektedir. Bundan dolayı her ne kadar bu süzgeç Şekil 2.5'de gösterilen anahtarlı bir N-yollu süzgecin (komütatörlü süzgeç) genel yapısından epeyce farklı görünmekteyse de, ilke olarak bu yapının tüm özelliklerini göstermektedir.

Şekil 2.6.a'daki N-yollu dur-geç süzgecin kazancı $H_0(s)$ tranfer fonksiyonunun büyüklüğü olarak Şekil 2.6.c'de çizilmiştir. Bu karakteristik daha önce değinildiği ve Şekil 2.6.a'da gösterildiği gibi, N-yollu süzgeç, kesim frekanslar $Nf_0/2$ olan iki alçak geçiren süzgecin arasına yerleştirildiğinde geçerlidir. Elbette süzgecin frekans eksenindeki çalışma alanı N'yi istenildiği kadar büyültmekle genişletilebilir. Şekil 2.6.c'de görüldüğü gibi tepeler, kf_0 frekanslarına, $k = 0, \mp 1, \mp 2, \dots, \mp N/2$ [eğer N tek sayı ise $(N-1)/2$], yerleşmiş olup şekil olarak $H(s)$ 'in özelliğini andırmaktadır. Yalnız frekans d ile kazanç da 2.12 denkleminde görülen Sinx/x fonksiyonuyla ölçeklenmiştir. Her ne kadar bu fonksiyon frekansa bağlıysada, tepelerin yerleştiği frekanslarda oldukça düz olduğundan her tepenin şekil ve bant genişliği $H(j\omega/d)$ 'ninkiyle aynıdır.

Şekil 2.6.c'deki karakteristikte tepelerin yükseklikleri frekans büyüdükçe azalmakta ise de N yeterince büyük alındığında böyle bir süzgeç geniş bir frekans bandında tarak tipi ve periyodik bir karakteristik gösterir. Bu tip karakteristiği olan süzgeçler beyaz bir gürültüyle karışmış periyodik işaretleri sezme için kullanılır. N-yollu süzgeçlerin genel ilkesini benimseyen böyle bir süzgeç de Wupper tarafından tasarlanmıştır (Wupper, 1972). Kovatacımı (bucked-brigate) denilen bir çift kapılı eleman kullanılarak gerçekleştirilen bu yapıda alt devreler arasında tam bir benzerlik sağlandığından daha önce sözü edilen ve istenmeyen görüntü bileşenlerini gerçekten yok etmek olanağı doğmuştur. Ayrıca ideal olmayan anahtarların kullanımından doğan alt devreler arasındaki girişim sorunu çözülmüştür. Bütün bunlardan dolayı deneysel sonuçlar teorik olanlarla hemen hemen aynıdır. Bu özellikleri taşıyan süzgeçler daha önce de belirtildiği gibi "Sözde N-yollu

Süzgeçler" olup, bunlar hakkında daha geniş bilgi (Fetweis ve Wupper,1971) ve (Wupper, 1972)'ye ek olarak (Bilotti,1968)'den elde edinilebilir.

Şimdi N-yollu dur-geç süzgeçlerin en yaygın uygulamalarından birisi olan dar bantlı bant geçiren süzgeç olarak kullanımını incelemenin tam sırasıdır. Bu olanak Şekil 2.6.a'da görülen süzgecin çıkışındaki alçak geçiren süzgeci geçirme bandı $f_0/2 < f < 3f_0$ olan bant geçiren bir süzgeçle değiştirmekle doğar. Bu şekilden de görüleceği gibi böylece oluşan süzgeç $Nf_0/2$ frekansıyla bant sınırlı olan her çeşit giriş işareti için bant geçiren bir süzgeç ödevi görür. Bu süzgecin tepe frekansı anahtarlama frekansı olan f_0 'a eşit olup bant genişliği de her biri alçak geçiren bir süzgeç olan alt devrelerin kesim frekansı ve anahtarların çalışma çalışma çarpanı d'ye bağlıdır. Onun için hem tepe frekansı hem de bant genişliği sırayla anahtarlama frekansı f_0 ve anahtarlama çarpanı d ile ayarlanabilmektedir. Yalnız tepe frekansının ayarlama alanı çıkış süzgecinin kesim frekansıyla kısıtlanmıştır; eğer bant genişliği yeterince dar ise tepe frekansı en fazla 1:2 oranında değiştirilebilir.

Yukarıda tanıtılan bant geçiren süzgecin en önemli özelliği bu süzgecin Q'sunun çok yüksek olmasıdır. Gerçekten de alt devreler olarak Şekil 2.6.a'daki gibi yalın RC alçak geçiren süzgeçlerin kullanıldığı varsayımıyla,

$$Q = \frac{RCf_0}{d} \geq RCNf_0 \quad (2.14)$$

bulunur. Onun için yüksek bir Q çeşitli şekillerde kolayca elde edilebilir (Ryken, 1970).

Darbantlı (yüksek Q'lu), kararlı, ayarlanabilir, yalın elemanlarla gerçekleştirilebilir olmasına karşın, bir N-yollu dur-geç devrenin bant geçiren bir süzgeç olarak kullanıldığında iki sakıncası vardır. Birincisi çıkışta geçirme bandı $f_0/2 < f < 3f_0$ olan yardımcı bir bant geçiren süzgeç kullanılması, ikincisi de bunun bir sonucu olarak tepe frekansının ayarlama alanındaki kısıtlanmadır (yukarıda da belirtildiği gibi tepe frekansı en fazla 1:2 oranında değiştirilebilir). Fakat yukarıdaki sakıncaları önleyecek yöntemlerde geliştirilmemiş değildir. (Köksal, 1975; Köksal ve Tokad, 1976; Köksal, 1975 *)'de gösterilmiştir ki Şekil 2.6.c'deki karakteristikte görülen ve $0, 2f_0, 3f_0, \dots, Jf_0$ ($J \leq N/2$) frekanslarında yerleşmiş

bulunan tepeleri hepten yok etmek veya belirli bir yüksekliđin altına dűşürmek imkanı vardır. Bu ise N-yollu sűzgecin her alt devresini bir periyot boyunca birden fazla defa alıřtırmak (giriř ile ıkıř arasına bađlamak) ve ters eviren anahtarlar (TA) (change-over switches) kullanmakla olur. Sonuta ıkıřta kullanılan yardımcı bant geiren sűzge, kesim frekansı J'ye bađlı olan alak geiren bir sűzgele deđiřtirilebilir. Bylece ve eđer $J=N/2$ alınırsa (N tek sayı ise $j= (N-1)/2$) ayarlama alanı geirme bandının yeterince dar olduđu durumlarda 1:2'den 1:N/2'ye ıkarılabilir. Bu paragrafta anlatılan zellikleri sađlayan filtre yapısı ve tasarımı Blm 3 'de ayrıntılı biimde anlatılmıřtır.



3. ANAHTARLANMIŞ N-YOLLU FİLTRELERDE TRANSFER FONKSİYONUNUN KONTROL EDİLiŞİ, OPTI VE NIKOP PROGRAMLARI

3.1. Giriş

Bu bölümde literatürde N-yollu süzgeçlerin transfer fonksiyonunun kontrol edilişi konusundaki belli başlı çalışmalardan hareket ederek, modülatörler yerine anahtarların kullanılmış olduğu anahtarlanmış N-yollu süzgeçlerin transfer fonksiyonunun kontrol edilmesine sistematik bir yaklaşım yapılmıştır. Hazırlanan bilgisayar programları OPTI ve NIKOP bu bölümün son kısımlarında tanıtılmış ve bazı örnekler verilmiştir.

3.2. Modülasyon Veya Anahtarlama Fonksiyonu İle Transfer Fonksiyonu Arasındaki İlişki

Bir N-yollu filtrenin transfer fonksiyonu Bölüm 2 de tanımlandı. Denklem 2.8'deki bağıntı ile modülasyon tipi N-yollu filtrelerin transfer karakteristiğinin değişmesi önerilen bir yolla mümkündür. Bu bağıntı, transfer fonksiyonunun, modülasyon fonksiyonlarının Fourier katsayılarına bağımlılığını açıkça gösterir (Franks ve Dann, 1970, Glasier ve Halkias, 1969, Franks ve Sandberg,1960). Bununla beraber anahtarlanmış N-yollu filtrede bu aşağıdaki şekilde kuvvetle sınırlandırılmıştır:

1.) Transfer fonksiyonu ve anahtarlama fonksiyonları arasında benzer ilişkinin olmayışı,

2.) Anahtarlama fonksiyonlarının kesikli deęişimleri yani sadece 0 ve 1 deęerlerini almaları,

3.) Periyodik olarak alıřan anahtarlar ieren devrelerin analiz tekniklerinin yetersizlięi,

nedeniyle bir N-yollu anahtarlanmış filtrenin transfer fonksiyonunun kontrolunda uzun zaman sadece anahtarlama fonksiyonlarının darbe-peryot ve darbe-geńişlik deęişimi kullanılıyordu. Bu üç sınırlandırmanın olmasına rağmen, ařağıdaki üç gerek bir N-yollu anahtarlanmış filtreden arzu edilen cevabın elde edilmesinde yardımcı olabilir.

1.) Bir modülasyon N-yollu filtresindeki modülatörlerle, bir anahtarlanmış N-yollu filtredeki anahtarlar arasında konum olarak kesin bir benzerlik vardır. Bu benzerlik Denklem 2.8'dekine benzer bir ilişkinin varlığını ummamıza neden olur. Böylece yukarıda verilen ilk sınırlandırma etkinliğini yitirir.

2.) Basit iki uçlu anahtarlar yerine Ters Çeviren Anahtar (TA) kullanılması +1 ve 0 deęerine ilave olarak -1 deęerini de alan bir anahtarlama fonksiyonunun tanımlanmasını mümkün kılar. Bu, direk baęlantı veya hi baęlantı olmaması durumundan bařka TA'nın iki kapısının ters baęlantısı ile olur. Böylece, basit anahtarların yanı sıra TA'nın da kullanılmasıyla ikinci sınırlandırma da serbest bırakılmış olabilir.

3.) MAINLN isimli bilgisayar programıyla herhangi bir (POA) ieren devrenin analizi bir periyottaki bölümlerin sayısına bakılmaksızın kolaylıkla analiz edilebilir. Böylece analitik hi bir zorlukla karşılaşılmadan her bir yol bir periyot boyunca birden fazla sayıda alıřmaya alınabilir. Artık klasik uygulama durumlarının aksine bir periyot boyunca tek bir darbe yerine anahtarlama fonksiyonunda darbe sıralarına izin verilebilir. Böylece, yukarıda sözü edilen son sınırlandırmanın üstesinden de anahtarları süren devrede bazı deęişiklikler yapılarak gelinebilir.

Gerekten TA'larla birleřtirilmiş bir N-yollu süzge yapılmıř (Pule, 1971) ve deneysel olarak doęrulanmıřtır. Bu filtrede ift harmoniklere yerleřmiř olan transfer fonksiyonunun piklerinin kazancı sıfıra indirilmiřtir. Bu filtre dur-ge tip olup klasik filtrelerden farklı

ilk uygulama olarak görülür ve yukarıdaki üç gerçeğe yakından ilgilidir. Madem ki TÇA çok fazla permütasyonda çalışabilir, o halde her bir anahtarlama kombinasyonu N-yollu filtrede başka karakteristikler üretmelidir. Pule, bunu belirtmekle beraber, cevap karakteristikleri ile anahtarlama işlemi arasındaki ilişkiyi kurmak için tam bir prosedürün olmadığına da işaret etmiştir. Yukarıdaki üç gerçeğin ışığında Pule'nin başlangıcı Köksal tarafından genişletilmiştir.

Şekil 2.5'deki N-yollu filtrenin anahtarlanmış biçimini ele alalım, burada anahtarların TÇA olduğunu varsayalım, böylece anahtarlama fonksiyonu,

$$\sum_{n=1}^N SI\left(t - \frac{n-1}{N}T_0\right) = 0 \text{ veya } 1 \text{ veya } -1 \quad (3.1.a)$$

$$\sum_{n=1}^N \left| SI\left(t - \frac{n-1}{N}T_0\right) \right| = 0 \text{ veya } 1 \quad (3.1.b)$$

şeklinindedir. Denklem 3.1.b'da verilen ifade herhangi bir anda en fazla bir yolun çalışmada olduğunu gösterir. Gerçekten $SI(t)$ pozitif ve negatif darbe sıralarından oluşur ve bir periyot boyunca en azından $(T_0 - T_0/N)$ süresince sıfırdır.

Sınırlanmış çalışma frekansı bölgesinde N-yollu süzgecin transfer özelliklerini belirleyen modülasyon fonksiyonlarının düşük dereceli katsayılarıdır. Bu gerçek ve yukarıda sözü edilen benzerlik aracılığıyla pratik amaçlar için $SI(t)$ 'nin Fourier serisi açılımındaki düşük dereceli Fourier katsayılarını istediğimiz gibi seçebileceğimiz bir yol araştıracağız.

3.3. Optimizasyon Probleminin Kuruluşu

Başlangıç olarak anahtar fonksiyonu dediğimiz $k(\theta)$ 'yı tanımlayalım.

$$k(\theta) = 2\pi \sum_{i=1}^I a_i \delta(\theta - \theta_i) \quad 0 \leq \theta \leq 2\pi \quad \text{için} \\ = k(\theta + 2\pi) \quad \text{tüm } \theta\text{'lar için} \quad (3.2.a)$$

Burada,

$$\theta = \frac{2\pi}{T_0} t ; 0 < \theta_i < \theta_{i+1} < 2\pi, i = 1, 2, \dots, I \quad \text{için} \quad (3.2.b)$$

$$a_i = \pm 1 \quad (3.2.c)$$

dir. Yani $k(\theta)$ her periyotta I darbelerinin bir sırasının oluşturduğu bir periyodik fonksiyondur. Anahtar fonksiyonu olan $k(\theta)$ 'nın Fourier katsayılarının mutlak değerlerinin karesi,

$$|C_n|^2 = \sum_{i=1}^I a_i^2 + \sum_{i=1}^I \sum_{j=i+1}^I 2a_i a_j \cos n(\theta_j - \theta_i), \quad n = 0, 1, 2, \dots \quad (3.3)$$

ile verilir.

Açıklayıcı bir örnek olarak, merkez frekansı anahtarlama frekansına yerleştirilmiş band geçiren filtre olarak kullanılacak bir anahtarlanmış N -yollu süzgecin tasarımını ele alalım. İsteriz ki,

$$|C_n|^2 = \text{bir maksimum değer, } n = 1 \text{ için} \quad (3.4.a)$$

$$= 0, \quad n = 0, 2, 3, 4, \dots, J (\geq 2) \quad (3.4.b)$$

olsun. Burada, J , band geçiren süzgecin arzu edilen frekans bölgesinde çalışmasına izin verecek kadar büyüktür. $J \leq N / 2$ olup yol sayısı ile sınırlıdır. O halde darbelerin olması gereken yerler olan θ_i açılarını ve yönleri olan a_i leri bulma problemi bir optimizasyon problemi olur ve sadece,

$$I \geq \begin{cases} J & \text{eğer } J \text{ çiftse} \\ J+1 & \text{eğer } J \text{ tekse} \end{cases}$$

çözülebilir. Bu problem Denklem 3.3'de verilen katsayıların karelerinin farkının ortalaması alınarak ve J indisinin öngördüğü küme üzerinde katsayıların uygun ölçeklenmiş biçimi seçilerek bir fonksiyonelin minimizasyonu için daha uygun olarak kurulabilir. Bir benzer problem daha önce incelenmişse de (Franks ve Dann, 1970) Denklem 3.1'deki sınırlandırmaların görülmediği ve problemin oldukça basitleştiği modülasyon N-yollu süzgeçle uğraşmışlardır.

Aşağıdaki satırlarda, yukarıda kurulan optimizasyon problemi bazı kabüllerin işe sokulması ile gevşetilmiştir. Denklem 3.3, $n=0$ için,

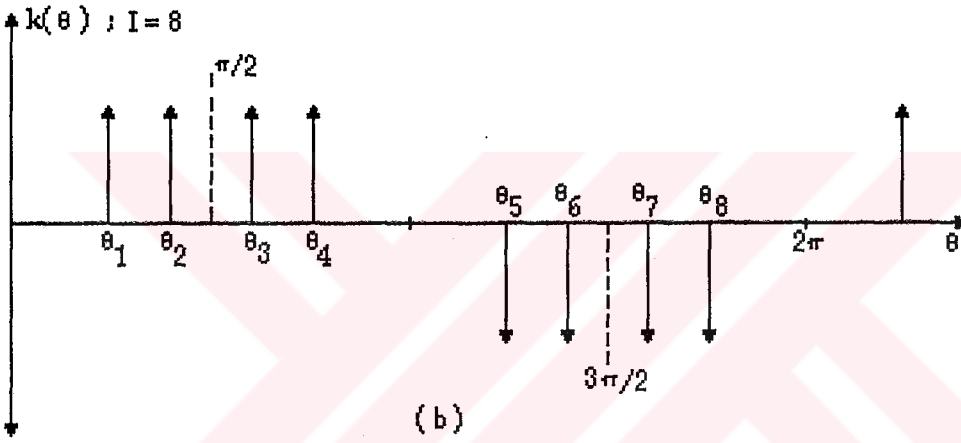
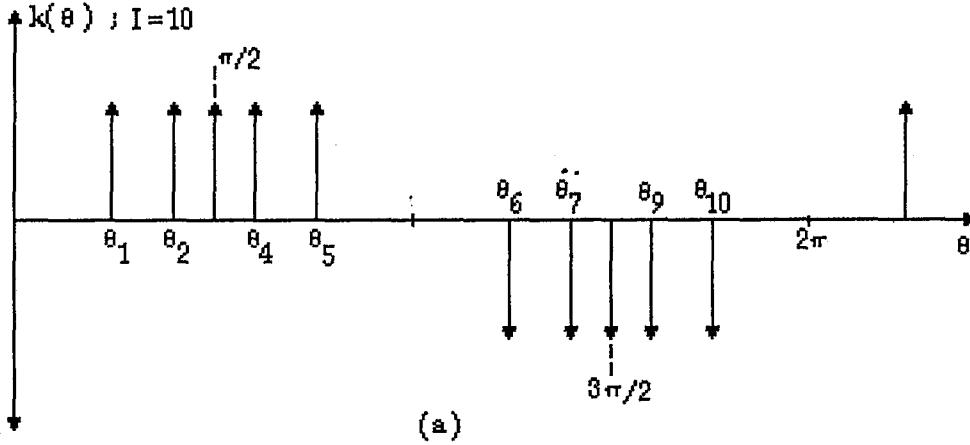
$$|C_0|^2 = \left(\sum_{i=1}^I a_i \right)^2$$

olarak yazılabilir, bu yüzden $C_0 = 0$ yapmak için I'yı çift seçmeliyiz çünkü a_i 'lerin yarısı 1, diğer yarısı -1 olmalıdır. Diğer yandan Denklem 3.4'ü sağlamak için yani, $|C_1|$ i maksimize etmek için I/2 pozitif darbelerin, kalan I/2 negatif darbeler tarafından izlenmesi veya tersi olmalıdır. Dahası Fourier analizinden biliyoruz ki eğer $k(\theta)$ çeyrek dalga simetrisine sahipse tüm çift harmonikler görünmez. Bu nedenlerle $k(\theta)$ anahtar fonksiyonu için Şekil 3.1'de gösterildiği gibi bir darbe sırası kabul edeceğiz. Böyle dalga şekilleri için Fourier açılımında çift harmoniklerin görünmediği ve sadece sinüslü terimlerin görüldüğüne göre, yukarıdaki optimizasyon problemini aşağıdaki şekilde yeniden formüle edebiliriz.

$0 < \theta_1 < \theta_2 < \dots < \theta_L < \pi/2$ olan açılar bulunacak, öyleki,

$$\sum_{i=1}^L \sin \theta_i + \frac{1}{2} : \text{maksimize edilmiş olacak ve} \quad (3.5.a)$$

$$\sum_{i=1}^L \sin \theta_i + (-1)^{\frac{n-1}{2}} \frac{1}{2} : 0 \quad n=3,5,7,\dots,J \text{ sadece tek tam sayılar} \quad (3.5.b)$$



Şekil 3.1 a) $I/2$ tek, b) $I/2$ çift durumları için anahtar fonksiyonları

Burada, $I/2$ Şekil 3.1.a'daki gibi tek farzedilmiştir ve $L = \frac{I}{4} - \frac{1}{2}$ dir, aksi takdirde, $I/2$ çifttir olup Şekil 3.1.b'deki gibidir. Zorla yüklenen bu simetri nedeniyle, gerekmediği halde J 'den büyük tüm çift harmonikler sıfır yapılır.

Darbelerin gerekli sayısı I , denklem 3.5 için çözüldüğünde, Denklem 3.4 için çözüldüğünden yaklaşık iki kere daha büyüktür. Gerçekten, $I/2$ ' nin tek olduğu durum için $I \geq 2J$ ve $I/2$ ' nin çift olduğu durum için $I \geq 2J - 2$ dir.

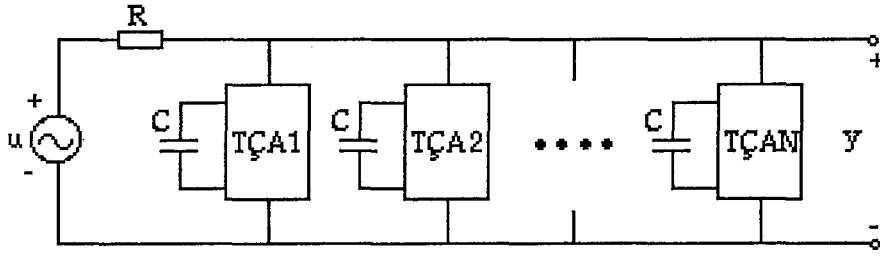
Optimizasyon probleminin belirlenmesinden sonraki adım anahtarlama fonksiyonunun hesaplanmasıdır. Bu, anahtarlama fonksiyonu $SI(t)$ ile $k(\theta)$ 'nin darbeleri arasındaki analogi kullanılarak yapılır. Denklem 3.2.c nedeniyle, $SI(t)$ 'deki tüm darbeler için aynı darbe genişliği öngörülmelidir. Bu, aynı zamanda, anahtarları sürececek olan elektronik devrenin yapımı için de bir kolaylık getirecektir. Bant genişliğini yeteri kadar dar tutarak ve $t_i = \frac{\theta_i}{2\pi} T_c$ daki her darbeyi a_i ile gösterilen yönde yerleştirerek denklem 3.1'i sağlamak mümkündür. Bununla beraber, maksimum güç transferi ve band genişliğinin darbe genişliği ile geniş bir sahada ayarı için darbe genişliği $\frac{T_0}{NI}$, mümkün olduğu kadar büyük seçilmelidir. Bunu sağlamak için her bir darbe orjinal olarak belirlenen yeri hafifce sağa veya sola kaydırılabilir. Ancak bu işlem sırasında ve genel olarak aşağıdaki şartlara uyulmalıdır.

1.) Her bir darbe ana periyot T_0 'ın NI eşit aralıkları arasından sadece bir aralığı işgal etmelidir.

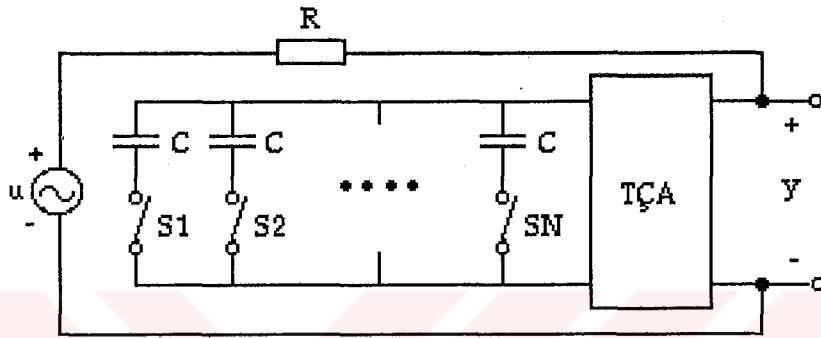
2.) Herhangibir aralıkta $[SI(t - \frac{n-1}{N} T_0) ; n=1, 2, \dots, N]$ fonksiyonlarından sadece biri sıfırdan farklı olmalıdır.

Böylece Denklem 3.1 sağlanmış olur. Bu yolla, tüm anahtarlama dalga şekilleri $NI f_0$ frekanslı ortak bir clock (saat)'dan elde edilebilir. $k(\theta)$ 'daki impulsların $SI(t)$ 'deki darbeler tarafından gösterilişi ve kaydırma işlemi nedeniyle $SI(t)$ 'nin Fourier katsayıları $k(\theta)$ 'ninkilerin sağladığı şartları kabaca sağlar. Bir sonuç olarak, $SI(t)$ 'nin J . derecenin üstündeki tek harmonikleri tam olarak yok edilemeyebilir. Böylece tüm prosedür, anahtarları süren devrenin karmaşıklığı, bant geçiren süzgecin transfer fonksiyonundaki toleranslar, darbe genişliği ile band genişliğinin ayar sahası, ana periyot ile merkez frekansının ayarlanması gibi konuları içeren kompleks bir genel optimizasyon problemi halini alır.

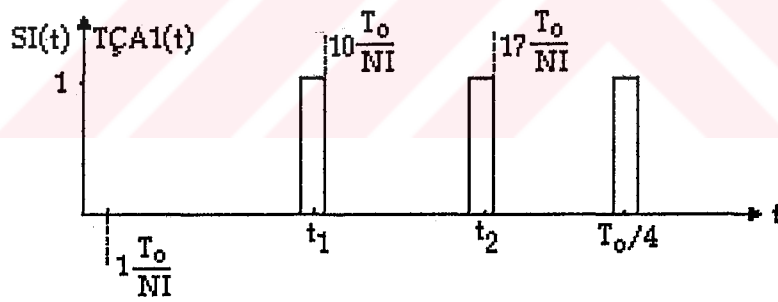
Yukarıdaki prosedürün ışığında, Şekil 3.2 a'da şeması, c'de anahtarlama fonksiyonu verilen bir N -yollu dur-geç filtre tasarlanmış ve frekans karakteristiği hesaplanmıştır.



(a)



(b)



(c)

Şekil 3.2 a) N-yollu süzgecin devre şeması, b) TÇA sayısının azaltılması, c) çeyrek periyotta 1. yolun çalıştığı aralıklar

Şekil 3.2. b'de gösterilen devre Şekil 3.2.a ile özdeş olup TÇA'ların sayısı 9'dan 1'e indirilmiştir. Bu süzgecin $R=5\text{ K}\Omega$, $C=0.2\text{ }\mu\text{F}$, 1000 Hz anahtarlama frekansı değerleri ve Şekil 3.2 c'deki anahtarlama fonksiyonu için transfer fonksiyonunun değişimi Şekil 3.3'de çizilmiştir.

Bu filtre 9 yol gerektirir ve $f_0=1000\text{ Hz}$ olmak üzere $(N/2)f_0$ 'den küçük frekanslarda bant geçiren süzgeç karakteristiği elde etmek için kullanılır. Bu yüzden $J=3$ seçebiliriz, yine de 4.5 kHz'den küçük frekans bölgesinde 5. harmoniğe yerleşen pikin muhtemel etkisini önlemek için $J=5$ seçelim. Şekil 3.1.a'da olduğu gibi bir çeyrek dalga simetrisi olduğunu varsayarak $I=10$ ($\geq 2J$) alabiliriz. Buradan $L=2$ olarak hesaplanır. Denklem 3.5 ise aşağıdaki biçimi alır.

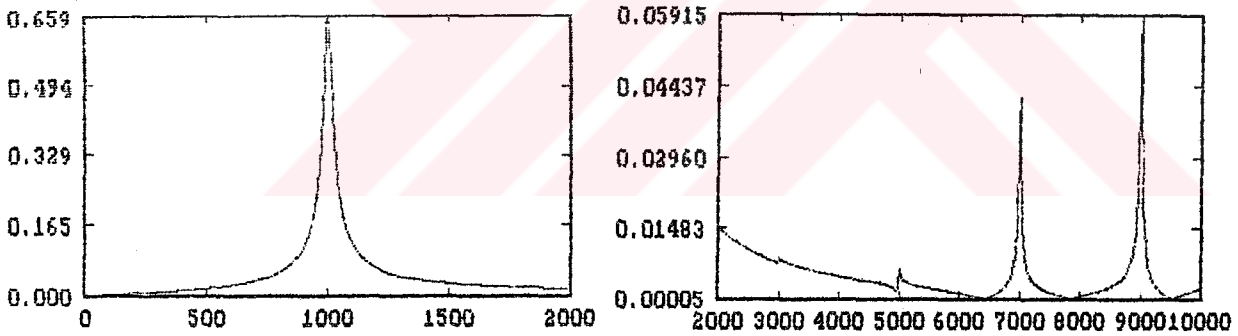
$$\begin{aligned}\sin 3\theta_1 + \sin 3\theta_2 - \frac{1}{2} &= 0 \\ \sin 5\theta_1 + \sin 5\theta_2 + \frac{1}{2} &= 0\end{aligned}$$

Bu denklemlerin çözümü bir bilgisayar programı kullanarak bulunabilir, örneğin en dik iniş yöntemi (steepest descent) kullanılarak ve başlangıç değerleri olarak $\theta_1=0.5$ ve $\theta_2=1.2$ tahmin edilerek bulunan sonuç $\theta_1=0.6626634$ (37.9678 derece) ve $\theta_2=1.1895628$ (68.1569 derece) şeklinde bulunmuştur. Böylece ilk iki impulsın yerleri ,

$$\begin{aligned}t_1 &= \frac{\theta_1}{2\pi} T_0 = 9.5 \frac{T_0}{NI} \\ t_2 &= \frac{\theta_2}{2\pi} T_0 = 16.9 \frac{T_0}{NI}\end{aligned}$$

olarak hesaplanır. Bu değerler, Şekil 3.2.c'de gösterildiği gibi ana periyot T_0 'ın ilk çeyreğinin 10. ve 17. aralıklarında gösterildi. Eğer $SI(t)$ 'nin birinci ve ikinci pulslarını 10. ve 17. aralıklara yerleştirirsek Tablo 3.1 den her aralık boyunca yalnız bir yolun uyarıldığını gözleriz, yani; denklem 3.1.b seçilmiş $SI(t)$ tarafından sağlanır. Yine de ikinci darbenin merkezinin tam olarak $16.9(T_0/Ni)$ değil $16.5(T_0/Ni)$ 'ya yerleştirilmesinin bir sonucu olarak ve $k(\theta)$ 'daki impulsların $SI(t)$ 'deki darbelerle değiştirilmiş olmasından $SI(t)$ fonksiyonu $k(\theta)$ 'dan biraz farklı Fourier açılımına sahip olacaktır. Bu nedenle Şekil 3.3'de

gösterildiği gibi 3. ve 5. harmoniklerdeki piklerin tam olarak bastırılması beklenmemelidir. Yine de bu pikler merkez frekansındaki pikten 38.27 ve 41.61 dB daha zayıf olup pratikte belirgin olarak bir olumsuzluğa yol açmaz. Tasarım prosedürünün esnekliğini göstermek için aşağıdaki durumu gözönüne alalım. Eğer Şekil 3.2.c'de gösterildiği gibi her biri t_1 ve t_2 yi içine alan iki aralığa yerleştirilen darbelerle Denklem 3.1'i sağlamayan bir anahtarlama fonksiyonu üretilseydi ne olurdu? Böyle durumlarda t_1 veya t_2 den birini diğerine yaklaştırmaya çalışabiliriz. Bunun karakteristiği çok fazla değiştirmedeği gözlenmiştir. Örneğin $SI(t)$ 'nin ikinci pulsunun 19. aralığa yerleştirilmesi bant geçiren filtrenin merkez frekansındaki pikten 34.75 ve 24.80 dB daha zayıf 3. ve 5. harmonik genliklerine yol açacaktır. Pratikte bu değerler hala geçirme bandı dışında bastırılacak frekans için yeteri kadar küçüktür.



Şekil 3.3 $R=5 \text{ K}\Omega$, $C=0.2 \text{ }\mu\text{F}$, 1000 Hz anahtarlama frekansı ve Şekil 3.2 c'deki anahtarlama fonksiyonu için Şekil 3.2 a veya b'deki devrenin transfer fonksiyonunun değişimi

Tablo 3.1 İlk periyot boyunca bütün anahtarların çalıştığı aralıklar (altları koyu çizgili olanlar pozitif, diğerleri negatif)

Yol	Çalıştığı Aralık										
1	10	17	23	29	36	55	62	68	74	81	
2	1	20	27	33	39	46	65	72	78	84	
3	4	11	30	37	43	49	56	75	82	88	
4	2	8	14	21	40	47	53	59	66	85	
5	5	12	18	24	31	50	57	63	69	76	
6	15	22	28	34	41	60	67	73	78	86	
7	6	25	32	38	44	51	70	77	83	89	
8	3	9	16	35	42	48	54	61	80	87	
9	7	13	19	26	45	52	58	64	71	90	

Tablo 3.1'de gösterilen sonuç (Köksal, 1975)'de defalarca deneme yapılarak bulunmuş bir çözümdür. Burada anlatılan prosedür bu tezde bilgisayar programı haline getirilmiştir. OPTI adı verilen ilk programda çeyrek periyot içindeki impulsların sayısı değişebilmekte ancak yönleri Şekil 3.1 a veya b'deki gibi olabilmektedir. Bu sınırlandırmanın kaldırıldığı program ise NIKOP adı verilen optimizasyon programıdır.

Açıların yerleri belirlendikten sonra bu açıların düştüğü aralıkların çözüm olabilmesi için, her bir aralıkta sadece bir yolun çalışıyor olması gerekir. Tablo 3.1'de verilen çözüm bu şartı sağlar. Benzer şekilde yukarıdaki satırlarda kısaca incelenen ve birinci yolun çalıştığı aralıkları 10, 19, 23, 27, 36, 55, 64, 68, 72, 81 olan çözümde bu şartı sağlar. Daha sonraki bölümlerde kısaca çakışma kontrolü olarak adlandırılan bu inceleme bu bölümde anlatılan OPTI ve NIKOP bilgisayar programlarında olduğu gibi Bölüm 4'de anlatılan GENKDI adlı Fortran bilgisayar programı içerisinde CAK adı verilen bir alt programda yapılmaktadır. Ayrıca simetri gözetilmeden benzer denemeleri yapan bir başka alt programla (ETRR)

burada verilen çözümden daha iyi bir çözüm ve bu çözüm kullanılarak fitreden elde edilen transfer fonksiyonunun değişimi Bölüm 4'ün sonunda verilmiştir.

3.4 . OPTI Programının Tanıtımı

Bu bölümün önceki kısımlarında verilen bağıntıları esas alarak, AKNYS'den istenilen transfer fonksiyonunu üretecek olan anahtarlama fonksiyonunu belirlemek amacıyla OPTI adı verilen bir Fortran bilgisayar programı hazırlandı. Bu program, biri fonksiyon olmak üzere 15 altprogram kullanır. Bunlardan GIR, giriş bilgilerinin bazılarının klavyeden girilip bazılarının hesaplandığı, YBAS, YA ve YS ise sırasıyla başlangıç, ara ve sonuç değerlerini yazan altprogramlardır. TAMAR, CAK, NORMZ ve DGETR alt programları Bölüm 4'de GENKDI bilgisayar programı içinde aynı değişken isimleriyle tanıtılmıştır. Bunlardan DGETR altprogramı GENKDI içinde kullanılırken sonuç civarındaki sınırlı bölgede çeyrek ve yarı dalga simetrisine bakmaksızın yeni aralık değerleri verirken, OPTI programında bu simetriye sahip olan aralık kombinasyonu değerlerini verir. OPTI bilgisayar programının listesi EK-1'de verilmiştir. Sonuç bilgilerini değerlendirirken, bulunan aralıkların anlamlı Fourier katsayılarının hesabında kullanılan ve Bölüm 4'de discrete Fourierle karşılaştırılan SB2A altprogramı da bu liste içersindedir. Fonksiyon altprogramı olan OBJE Denklem 3.6'da verilen ve maksimize edilecek olan ϕ değerini hesaplar.

$$\phi(\theta) = G_1(\theta) - AK_j \sum_{i=3}^M [(G_i(\theta))^2 - (a_i)^2]^2 \quad (3.6)$$

Burada, G_1 birinci harmonik genliğini, a_i 'ler 3'den M'e kadar bastırılacak harmoniklerin alması gereken küçük genlik değerlerini gösterir. Bu haliyle object fonksiyonun analitik türevide kolayca alınabilir. Burada θ bir vektör olup boyutu kullanılan

impuls sayısı ile belirlidir. AK'nın artan deęerleri için bu fonksiyonun maksimizasyonu sonucu olarak impulsların yerleřtirileceęi açđ deęerleri bulunmuř olur.

OPTI programının akıř řemasđ Şekil 3.4'de çizilmiř ve bazı altprogramların ayrıntısına girmeye gerek görülmemiřtir. Kullanıcıya sorulan bazı deęiřkenlerin açıklamaları ařaęıda verilmiřtir.

KSE : 1 deęeri için Şekil 3.3 b'deki, 2 deęeri için a'daki yerleřtirme düzeninin kullanıldıęını gösteren tam sayıdır.

N : Çeyrek peryot içinde kullanılan impuls sayısını gösteren tamsayıdır.

M : Bastırılmak istenen harmoniklerin sonuncusunun numarasıdır.

NY : AKNYS'in yol sayısıdır.

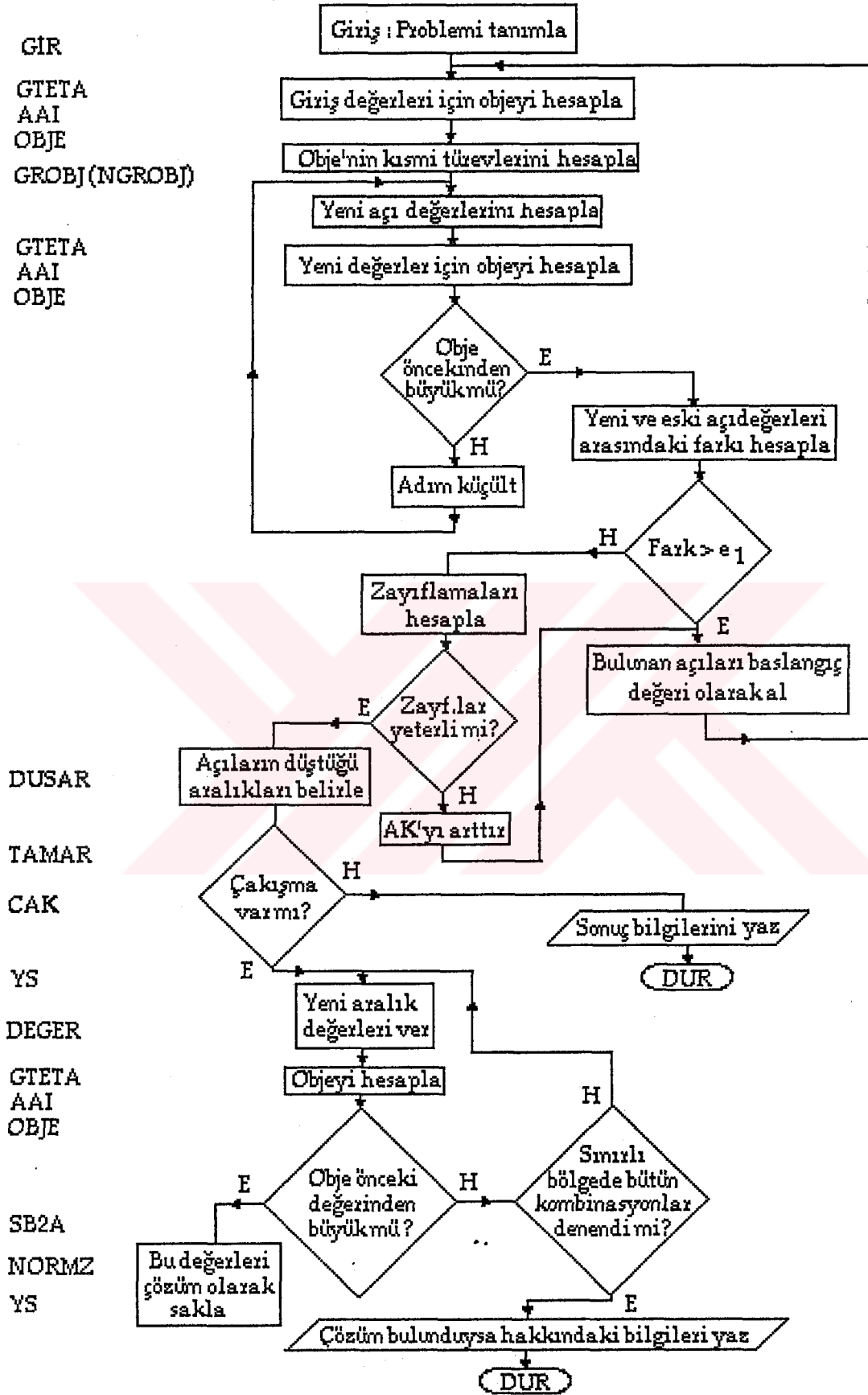
B : M'e kadar bastırılacak olan harmoniklerde istenen zayıflamaların dB olarak deęerlerini içeren vektördür.

AK : 3.6'daki baęıntıda aynı isimle gösterilen deęiřkenin bařlangıç deęeridir.

AKAT : AK'nın arttırılması için kullanılan katsayıdır.

K1 : Çözüm etrafında tarama yapılacak bölgeyi sınırlayan çift sayıdır.

ZREF : Tarama yapılan bölgede, bastırılmıř olması gereken harmoniklerde yeterli görülen zayıflamaları içeren vektördür.



Şekil 3.4 OPTI programının akış şeması

3.4 .1. GTETA altprogramı ve deęişken isimleri

Bu altprogram KSE, N, M, TETA deęişkenlerinin deęerlerini giriş olarak alır ve Denklem 3. 5'deki baęıntılar üzerinden harmonik genliklerini hesaplar. Gerçekte bu deęerler tam olarak Fourier serisindeki katsayılar olmamakla birlikte çarpan olarak gelecek katsayının optimizasyon programının işleminde bir etkisi yoktur. Daha önce açıklanmayan deęişken ismi olan,

TETA : impulsların bulunduğu açların radyan olarak deęerini içeren vektördür.

3.4 .2. AAI altprogramı ve deęişken isimleri

Bu altprogram M, G ve B'nin verilen deęerleri için küçük olmasını istediğimiz harmonik genliklerinin istediğimiz zayıflama deęerini sağlamak için eşit olması gereken ve Denklem 3.6'da a_i ile gösterilen deęerleri hesaplar. Bu altprogramda a_i 'ler A ismi verilen bir vektörle gösterilmiştir.

3.4 .3. GROBJ , NGROBJ altprogramları ve deęişken isimleri

GROBJ altprogramın deęişken isimleri N, M, TETA, G, GG, A, AK, GOBJ, DELTA'dır. Bu deęişkenlerden daha önce açıklanmayanlar aşağıda verilmiştir.

GG : Denklem 3.6'da AK ile çarpım halinde görülen kısmın N tane açığı için kısmi türevlerini içeren vektördür.

GOBJ : Denklem 3.6'da verilen fonksiyonun N tane açığı için kısmi türevlerini içeren vektördür.

DELTA: Eşlenik doğrultunun hesabında kullanılan ve kısmi türevlerin karelerinin toplamının kare kökü olan reel değişkendir.

GROBJ, object fonksiyonunun açılara göre kısmi türevlerini analitik olarak hesaplar. Bu değerler kullanılarak objeyi artıracak yönde yeni açığı değerleri bulunur. Türevlerin numerik olarak hesaplandığı altprogram **NGROBJ** 'da EK-1' de OPTI bilgisayar programı ile verilmiştir. Bu altprogram **GTETA**, **AAI** ve **OBJE** altprogramlarını çağırarak her bir açığı verilen küçük bir artışla object fonksiyonunun değişim miktarını hesaplar ve kısmi türevleri belirler. Bu altprogramın değişken isimleri N, M, TETA, KSE, G, A, B, AK, GOBJ ve DELTA'dır. GROBJ veya NGROBJ altprogramlarından herhangi birinin kullanımı programın sonuçlarında bir farka yol açmaz.

3.4.4 . DUSAR altprogramı ve değişken isimleri

Bu altprogram bulunan açığı değerlerinin hangi aralıklar içinde kaldığını belirler. Değişken isimleri KSE, N, TETA, NY, AR olup ilk dört değişkenin değerine göre sonuncusu olan AR belirlenir.

AR : Çeyrek peryot içindeki TETA'ların düştüğü aralıkları içeren vektördür.

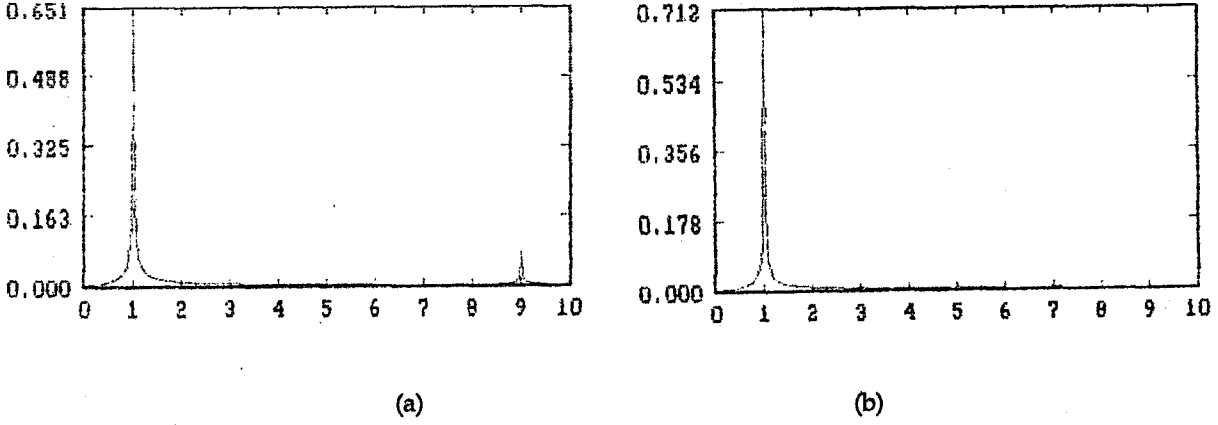
3.5. OPTI Programı İle Çalışılan Örnekler

Çeyrek peryot içindeki impuls sayısına bağlı olarak bastırılacak harmonik sayısı belirlidir. OPTI bilgisayar programına Düzen=1, N=2, M=5, NY=11 girilerek 3. ve 5. harmoniklerde 100 dB zayıflama istenirse bulunan açılar, derece olarak 42.124 ve 78.05'dir. Bu açıların çeyrek peryot içinde düştüğü aralıklar ise 11. ve 20. aralıklardır. Diğer yollarla çakışma söz konusu olmadığından program, 11,20,25,34,55,64,69,78 aralıklarını çözüm olarak yazar. Benzer şekilde aynı aralık yerleştirme düzeninde N=3, M=7, NY=15 için bulunan açılar 34.118, 63.191 ve 78.317 derecedir. Bu açılar 18., 32. ve 40. aralıklara düşer. Çakışma olmadığından bu aralıkların etrafı araştırılmaz ve program bu aralıkları çözüm olarak yazar. N=4, M=9, NY=19 için bulunan açılar 29.274, 54.863, 65.153, 88.883 derece ve içinde kaldıkları aralıklar 25,46,55,76 'dır. Çakışma olduğu için K1=2 verilerek bu aralıkların etrafında program tarafından yapılan 54 kombinasyon denemesi yapılır ve sonuç olarak 25, 46, 55, 76 aralıkları bulunur.

İkinci aralık (veya impuls) yerleştirme düzeni daha avantajlıdır ve daha yüksek zayıflama değerlerine ulaşılabilir. Bu düzende N=2, M=5, NY=11 için bulunan açılar 38.118 ve 68.232 derecedir. Daha yüksek zayıflama değerleri için açılar Denklem 3.5'in çözümü olan 37.967 ve 68.156 derece olarak bulunur. Her iki durumda da açıların düştüğü aralıklar 12, 21 ve yeri sabit olan aralık 28'dir. Bu aralıklar için çakışma yoktur ve çözüm olarak alınabilir. N=3, M=7, NY=15 için bu aralık düzeninde bulunan açılar 31.844, 57.621, 72.982 derecedir. Düşükleri aralıklar 19, 34, 43'dür. 90 derece orta.nokta olacak şekilde yerleştirilen aralık 53. aralık olup bu kombinasyon içinde çakışma söz konusu olmadığından çözüm olarak yazılır. N=4, M=9, NY=19 ve 40 dB zayıflama için OPTI programıyla bulunan açı değerleri 35.777, 53.597, 66.759, 78.464 derecedir. Bu açıların düştüğü aralıklar olan 34, 51, 64, 75, 86 için yapılan çakışma kontrolünün sonucu olumsuzdur ve denenen 81 kombinasyondan sonra 35, 52, 63, 76, 86 aralıkları çözüm olarak bulunur. N=5, M=11, NY=23 ve 100 dB zayıflama değerleri

giriş olarak verilirse bulunan açılar 25.000, 46.828, 54.483, 73.830, 75.710 derecedir. Bu açılar içinde kaldığı aralıklar olan 36, 66, 77, 104, 107, 127 için yapılan çakışma kontrolü sonucu da olumsuz olduğundan bu aralıklar etrafındaki 243 kombinasyon denenerek 36, 67, 76, 104, 127 çözümü bulunmuştur. $N=6$, $M=13$, $NY=27$ için bulunan açılar 20.967, 40.107, 49.557, 66.777, 66.795 derecedir. Bu açılar düştüğü aralıklar olan 41, 79, 97, 131, 131, 162, 176 incelendiğinde aynı aralığa düşen iki açı olduğu görülür. Bu durumda program aralıkları 41, 79, 97, 130, 131, 162, 176 şeklinde düzenler. Bu aralık kombinasyonu için çakışma söz konusudur ve $K1=2$ değeri için de çözüm yoktur. $K1=4$ değeri için bu aralıkların etrafındaki 9375 kombinasyon denenerek 43, 78, 96, 129, 133, 162, 176 sonucu çözüm olarak bulunur. Son olarak $N=7$, $M=15$, $NY=31$ için OPTI programıyla bulunan açı değerleri 18.386, 35.244, 45.819, 59.330, 61.989, 78.402, 78.402 derece olarak bulunur. 6. ve 7. açılar düştüğü aralıklar yeniden düzenlenerek 48, 92, 119, 154, 161, 202, 203, 233 aralıkları için çakışma kontrolü yapılır. Çakışma olduğu için $K1=2$ değeriyle bu aralıkların etrafındaki 48, 91, 119, 154, 161, 201, 204, 233 aralıkları yukardakiler gibi 20 dB referans zayıflama kullanılarak bulunmuş olur.

Bu sonuçlar Bölüm 4'deki GENKDI örnekleriyle karşılaştırılabilir, örneğin burada ikinci aralık düzeninde $N=3$, $M=7$, $NY=15$ için bulunan 19, 34, 43, 53 çözümünün uygulanması ile elde edilecek olan filtre transfer karakteristiği Şekil 3.5 (a) ile aynı problemin Bölüm 4'de GENKDI programının 20 dB referans zayıflama değeri için bulduğu 19, 36, 44, 53 çözümünün aynı süzgeçte üretecekleri transfer karakteristikleri, Şekil 4.5 (a) arasında belirgin bir fark yoktur. Benzer şekilde OPTI programıyla ikinci aralık düzeni kullanarak $N=4$, $M=9$, $NY=19$ ve 40 dB zayıflama için bulunan 34, 51, 64, 75, 86 aralıklarından oluşan anahtar fonksiyonunun, aynı eleman değerleri için aynı süzgeçten elde edilen transfer fonksiyonunun değişimi olan Şekil 3.5 (b)'nin, Bölüm 4'de GENKDI'nin bulduğu 30, 55, 56, 76, 86 aralıklarının anahtar fonksiyonu olarak kullanılmasıyla elde edilen ve Şekil 4.7'de verilen karakteristiğe göre merkez frekansında biraz daha yüksek kazançtan başka belirgin bir farkı yoktur.



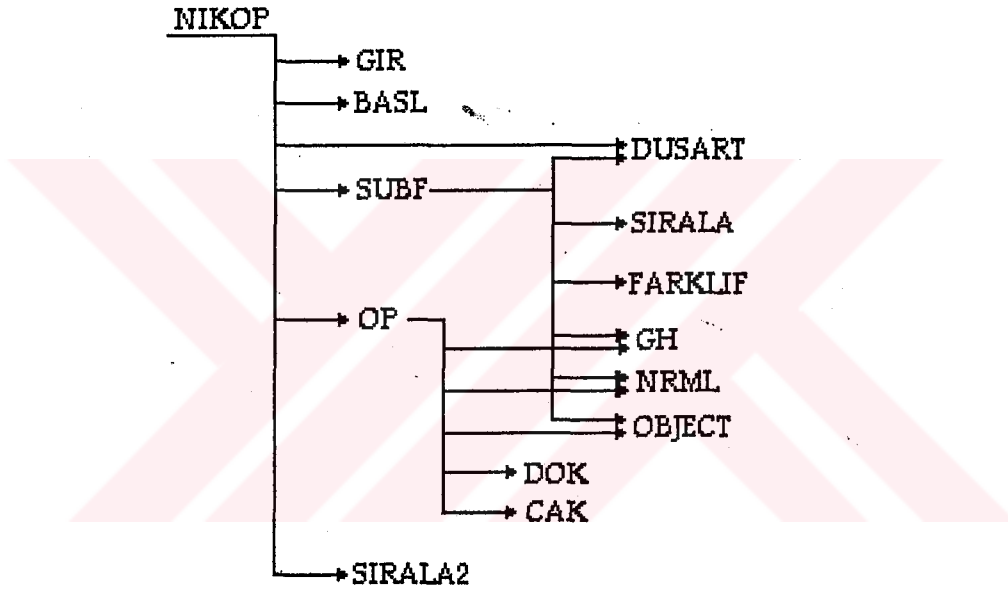
Şekil 3.5 a) $N=3$, $M=7$, $NY=15$, b) $N=4$, $M=9$, $NY=19$ için OPTI programının bulduğu anahtar fonksiyonlarının $R=5K\Omega$, $C=0.2\mu F$ için Şekil 3.2 b'de verilen filtrede ürettikleri transfer fonksiyonları

Yukarıda verilenlerden farklı olarak örneğin, ikinci aralık düzeninde $N=3$, $M=7$ yerine 9, $NY=19$ ve 30 dB zayıflama için GENKDI ve OPTI bilgisayar programlarıyla elde edilen sonuçlar aynı olup 30, 45, 57, 67 aralıklarıdır.

3.6. NIKOP Programının Tanıtımı

Bu program 13 tane alt program kullanır. Bu altprogramlar ve çağırma ilişkileri Şekil 3.6'da gösterilmiştir. Altprogramlardan CAK diğer iki programdaki gibi çakışma kontrolü yapar. DUSART alt programı da diğer programlardaki DUSAR gibidir, farkı sadece bir peryot boyunca açıların düştüğü aralıkları belirlemektir. Bu altprogramlar tekrardan kaçınmak için anlatılmayacaktır. DOK altprogramı ise OP altprogramı tarafından çağırılan ve ara sonuçları yazan altprogramdır. Bu altprogramında ayrıntısına gerek görülmemiştir. NIKOP programı çalıştırıldığında GIR altprogramı kullanıcıya KIS, M, NY değerlerini sorar. Bu değişken isimleri diğer programlarda açıklandığı gibidir. Programa M

ile verilen harmoniğe kadar harmonik genliklerinin, aralarında en büyük olanına göre normalize edilmiş değerleri girilir. Belirlenen işaret kombinasyonlarının her biri için normalize edilmiş genlik değerleri üzerinden hesaplanan OBJECT fonksiyonunun optimizasyonu yapılır. Yeni açı değerlerinin hesaplanması için eşlenik gradyant yöntemi kullanılır ve kısmi türevler numerik olarak belirlenir. NIKOP bilgisayar programının genel bir akış şeması ayrıntıya girilmeden Şekil 3.7'de gösterilmiştir. NIKOP programının listesi EK-2'de verilmiştir.



Şekil 3.6 NIKOP ana programı tarafından kullanılan altprogramlar ve çağırma ilişkileri

3.6.1. GIR ve BASL altprogramları ve deęişken isimleri

GIR altprogramının deęişken isimleri KIS, M, HISTB, NY'dir. Bu deęişken isimleri ařaęıda açıklanmıştır.

KIS : Bir peryot içindeki impuls sayısıdır.

M : Kontrol edilecek harmonik numarasıdır. Örneęin $M=3$ için ortalama deęer, 1., 2., 3. harmonik genlikleri kontrol edilecek demektir.

HISTB: N-yollu süzgecin transfer fonksiyonunda anahtarlama fonksiyonunun harmoniklerinin frekanslarına karşı düşen frekanslarda almasını istedięimiz normalize genlik deęerleridir. Ortalama deęerde dahil olduęundan HISTB, $M+1$ boyutunda bir vektördür.

NY : Yol sayısıdır.

BASL altprogramı KIS deęişkeninin deęeriyle sayısı verilen impulsları bir peryot içine simetrik olarak yayar. Deęişken isimleri KIS ve TETA olup,

TETA: Radyan olarak açıları gösteren vektördür.

3.6.2. SUBF altprogramı ve deęişken isimleri

SUBF altprogramı bu impulsların yönlerinin olabilecek bütün kombinasyonları için farklı harmonik spektrumunu oluşturan işaret kombinasyonlarını belirler. Deęişken isimleri KIS, BM, TETA, A, B, KJ, ABN, ABNK, ZABNK, ABNKB, HIST, KJB, KJB1, KJB2, OB, OBF, KJF,IF olup daha önce açıklanmayanlar ařaęıda verilmiştir.

BM : Satır sayısı 2^{KIS} , sütun sayısı KIS olan ve bütün işaret kombinasyonlarının belirleneceği matristir.

KJ : BM matrisinden faydalanılarak KIS sayısı kadar impulsın seçilmiş yönlerini içeren vektördür.

OB : Her bir işaret kombinasyonu için hesaplanan object değerlerini içeren ve BM'nın satır sayısı boyutunda olan vektördür.

OBF : Her bir işaret kombinasyonu için hesaplanan ve diğerlerinden farklı olan object değerlerini içeren vektördür.

KJB : Her biri için objenin hesaplandığı ve BM matrisindeki simetri nedeniyle aynı spekturumu verecek olan kısmın ayıklandığı, satır sayısı 2^{KIS-1} , sütun sayısı KIS olan işaret kombinasyonu matrisidir.

Diğer değişken isimleri ilgili altprogramlarda açıklanacaktır.

3.6.3. SIRALA ve FARKLIF altprogramları ve değişken isimleri

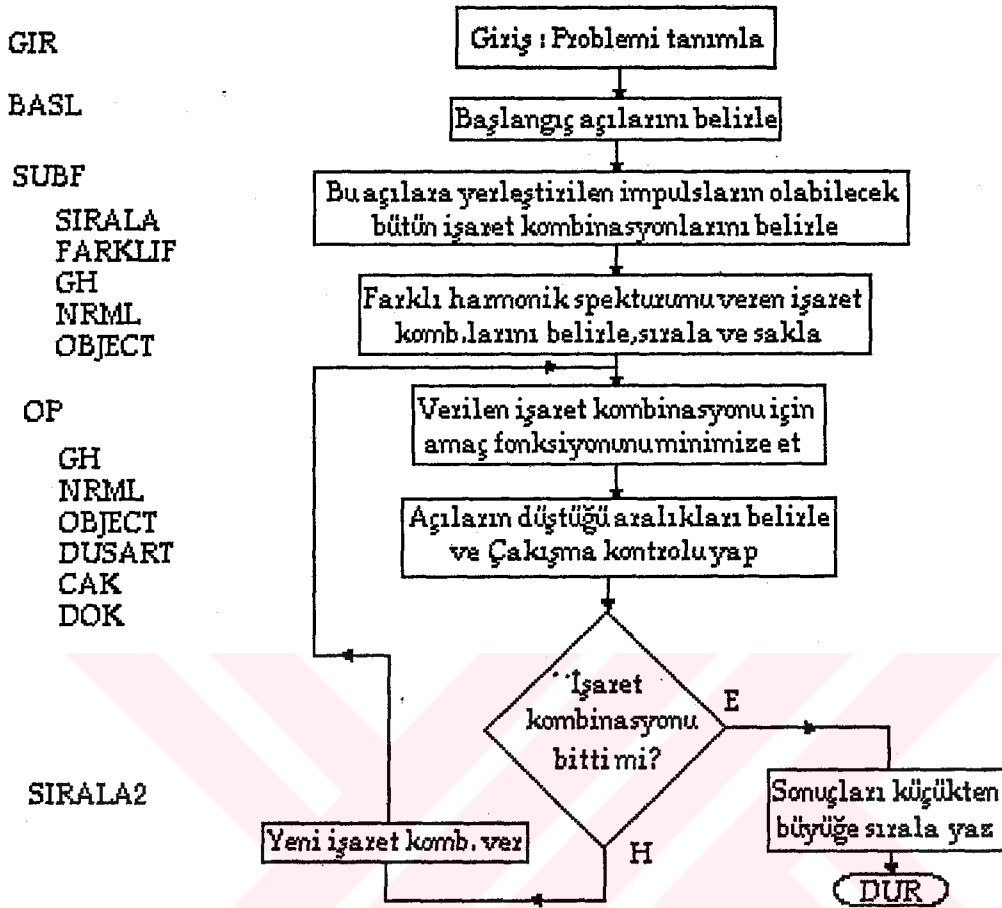
SUBF altprogramı içinde kullanılan altprogramlardır. Bunlardan SIRALA altprogramının değişken isimleri NSAT, OB, KJB, KIS, KJB1, KJB2 olup

NSAT : BM matrisinin satır sayısıdır.

KJB1, KJB2 : OB değerlerinin küçükten büyüğe sıralanması yapılırken işaret kombinasyonlarının değerlerini tutan vektöryel değişkenlerdir.

FARKLIF altprogramının değişken isimleri N, S, OBF, KIS, KJB, KJF, IF'dir. Bunlardan daha önce açıklanmayan sonuncusu,

IF : Farklı obje veya başka bir deyişle farklı harmonik spekturumları olan işaret kombinasyonlarının sayısıdır.



Şekil 3.7 NIKOP bilgisayar programının genel akış şeması

3.6.4. GH altprogramı ve değişken isimleri

Denklem 3.5'de verilen bağıntılar esas alınarak genelleştirilmiş ve impulsların işaretleride göz önüne alınarak, ortalama değerde dahil olmak üzere tüm harmonikleri hesaplayacak hale getirilmiştir. Bu bağıntıları kullanan altprogram GH ileriki bölümlerde de kullanılan discrete Fourier analizine taban teşkil eder. Bu altprogramın değişken isimleri KIS, NY, M, TETA, A, B, AB, KJ, ORT'dur.

A : Harmoniğin sinüslü terimlerinin katsayılarını,

B : Kosinüslü terimlerinin katsayılarını,

AB : Harmonik genliklerinin mutlak değerlerini içeren M boyutundaki vektörlerdir.

ORT : İşaret kombinasyonundan hesaplanan ortalama değerdir.

3.6.5. OBJECT ve NRML altprogramları ve değişken isimleri

NRML altprogramının değişken isimleri M, A, B, ABN, ABNK, ZABNK'dır. Bunlardan daha önce açıklanmayan,

ABN : Anahtarlama fonksiyonunun normalize edilmiş harmonik genliklerini,

ABNK : Anahtarlama fonksiyonunun normalize edilmiş harmonik genliklerinden hesaplanan ve filtre karakteristiğinde anahtarlama fonksiyonunun normalize edilmiş harmonik genlikleri değerlerini içeren vektördür. **

ZABNK : Nonmaliz edilmiş harmonik genliklerinin dB olarak zayıflama değerlerini içeren vektördür.

OBJECT altprogramının değişken isimleri M, HIST, ABNK, OBJ'dır. Bunlardan,

OBJ : İstenen normalize genlik değerleri olan HIST ve ABNK'nın farklarının kareleri toplamı şeklinde belirlenen amaç fonksiyonun değerini gösteren reel sayıdır.

3.6.6. OP altprogramı ve değişken isimleri

Bu altprogram her bir farklı harmonik spektrumunu üreten işaret kombinasyonu için başlangıç açısı değerlerini alır ve eşlenik gradyant yöntemi kullanarak amaç fonksiyonunu

minimize eder. Eşlenik doğrultunun belirlenmesi için kısmi türevler sayısal olarak hesaplanır. Değişken isimleri KIS, IF, TETA, TETAB, KJ, KJF, M, A, B, AB, ABN, ABNK, ZABNK, HIST, OBJ, EK, ISRT, GOBJ, AR, KY, OBB, NY'dir. Bunlardan daha önce açıklanmayan,

EK : Her bir aç için amaç fonksiyonunun kısmi türevleri hesaplanırken objeyi küçültecek yönde eski açya eklenecek veya çıkarılacak aç değerini içeren vektördür.

ISRT : EK değişkenlerinin yönlerini içeren vektördür.

GROBJ : Amaç fonksiyonunun sayısal olarak hesaplanan kısmi türevlerini içeren vektördür.

AR : Açların bir peryot boyunca düştükleri aralık değerlerini içeren vektördür.

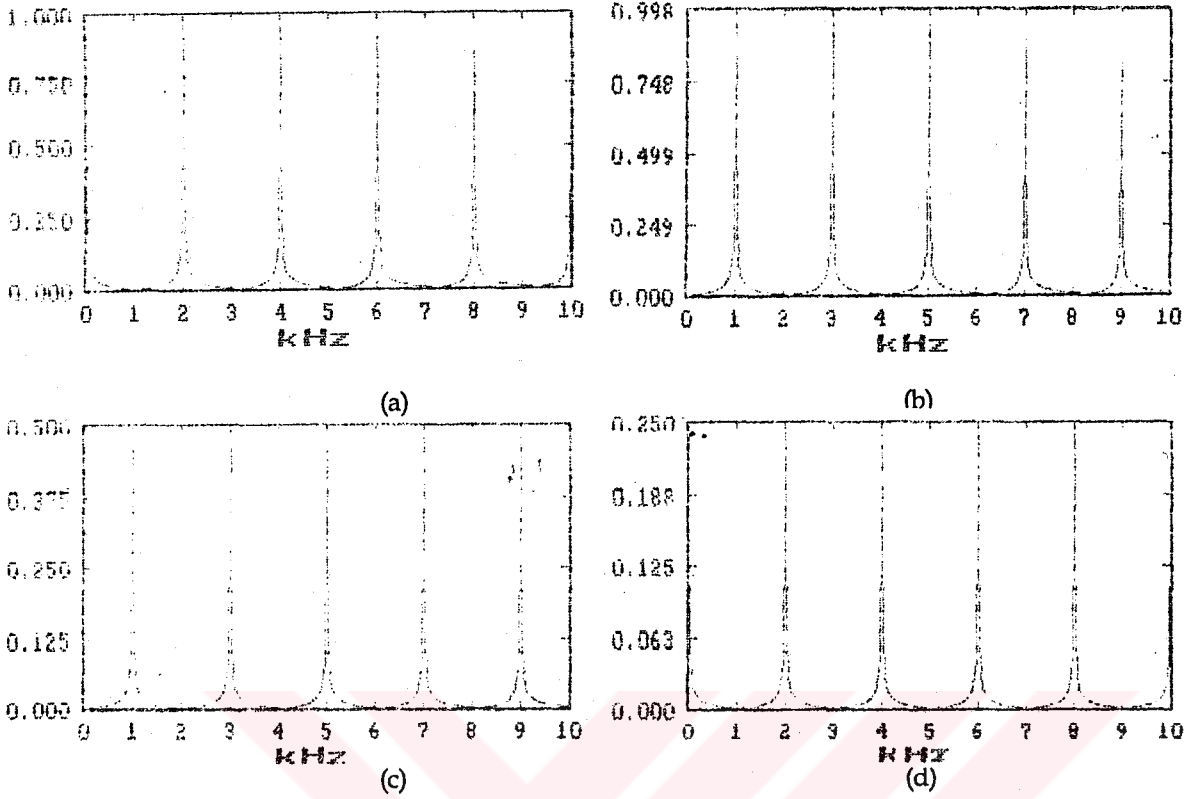
KY : Çakışma kontrolunda kullanılan ve diğer yolların çalıştığı aralıkları içeren vektördür.

3.7. NIKOP Programı ile Çalışılan Örnekler

Bu genel programla bazı giriş değerleri için OPTI ve GENKDI programlarının buldukları sonuçlar aynen veya yaklaşık olarak bulunur. Ayrıca diğer programlarla araştırılmayan bazı özel sonuçlar elde edilmiştir. İlk örnek olarak bir anahtarlama peryodunda iki impulsın olduğu ve 9. harmoniğe kadar normalize genlik değerleri olarak 1,0,1,0,1,0,1,0,1,0 şartını sağlayan veya buna yaklaşan bir çözüm arayalım. Yol sayısı olarak 19 verildiğinden program bulunduğu açları 10. ve 29. aralıklara yerleştirir. Her iki aralığın yönüde aynıdır ve çakışma yoktur. Bu anahtar fonksiyonu Örnekler için hep kullanılan $R=5 K\Omega$, $C=0.2 \mu F$ ve 1000 Hz anahtarlama frekansı için Şekil 3.8 a'daki karakteristiği üretir. Tarak tipi olan bu süzgecin karakteristiğindeki tepeler anahtarlama frekansının çift katlarına yerleşmiştir. Bu seyrek dişli tarak karakteristiği Şekil 3.9'da karakteristiği

verilen ve her bir yolun bir peryot boyunca bir kez devreye alındığı 19 -yollu klasik tarak filtre ile karşılaştırılırsa, her iki karakteristiğinde geçerli olduğu 8.5 kHz'e kadar tepelerin kazançlarındaki göreceli düşmenin normal tarak tipinde daha fazla olduğu görülür. Benzer şekilde NIKOP programından normalize genlik değerleri olarak bu defa 0,1,0,1,0,1,0,1,0,1 istenirse bulunan aralıklar yine 10. ve 19. aralıklardır ancak işaretleri birbirinin tersidir. Sonuç olarak elde edilen süzgeç karakteristiği, Şekil 3.8 b'de gösterildiği gibi anahtarlama frekansının tek katlarına yerleşmiş seyrek dişli tarak filtresidir.

Eğer tarak tipi karakteristiktan beklenen, artan frekanslarda tepelerin kazançların göreceli olarak az değişmesi ise böyle bir durum, bütün tepeler de daha az genlik değeri fakat göreceli zayıflama farkının azaldığı bir karakteristik olan Şekil 3.8 c' de gösterilmiştir. Bu transfer fonksiyonu NIKOP programından $KIS=4$, $NY=19$, $M=9$ ve normalize harmonik genliklerinin 9. harmoniğe kadar sırasıyla 0,1,0,1,0,1,0,1,0,1 değerlerini sağlamak üzere bulunan ilk ikisinin negatif son ikisinin pozitif (veye tersi) işaretli olduğu 10,29,48,67 anahtar fonksiyonu için elde edilmiştir. Eğer $KIS=8$ verilerek bu defa çift harmoniklere yerleşmiş pikleri olan bir karakteristik elde etmek istersek 13,26,51,64,89,102,127,140 aralıklarının anahtar fonksiyonu olması durumu için elde edilen süzgeç karakteristiği Şekil 3.8 d incelenirse yol sayısı az olmasına rağmen karakteristiğin geçerli olduğu frekans bandında göreceli zayıflamalar hemen hemen ortadan kalkmıştır. Ancak bu defada geçirme frekanslarındaki genlik değerleri düşüktür.



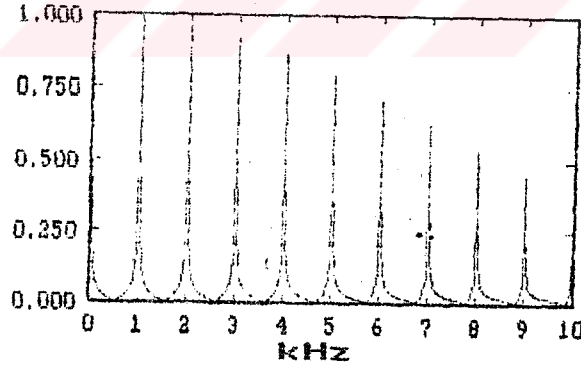
Şekil 3.8 Anahtar fonksiyonu olarak (a) +10, -19, (b) -10, +19, (c) -10, -29, +48, +64 (d) 13, -26, 51, 64, 89, -102, 127, 140 aralıklarının kullanılması durumunda elde edilen seyrek dişli tarak tipi karakteristikler

Bir başka uygulama olarak programdan $KIS=4$, $M=5$, $NY=11$ ve 5.harmoniğe kadar sırasıyla 0, 0, 1, 0, 0, 0 normalize genliklerini sağlamasını veya başka bir deyişle 2. harmoniği en büyük, diğerlerini küçük yapmasını istersek, bulunan sonuç, -, +, -, + işaret kombinasyonundaki 6, 17, 28, 39 aralıklarıdır. Gerçekten de bu aralıklar anahtarlama periyodunun 2. harmoniğinin tepelerinde yerleşmiştir ve önceden görülebilen bir sonuçtur. Bu anahtar fonksiyonu ile süzgeçten elde edilen karakteristik Şekil 3.10. a'da gösterilmiştir. Bu anahtar fonksiyonunun 2, 6, 10, 14,... den başka harmonikleri sıfırdır, böylece yol sayısı yeteri kadar büyük tutularak, örneğin $NY=27$ için -14, 41, -68, 95 aralıklarının anahtar fonksiyonu olması durumunda 13.5 kHz'e kadar geçerli olan daha seyrek dişli ve tepelerdeki

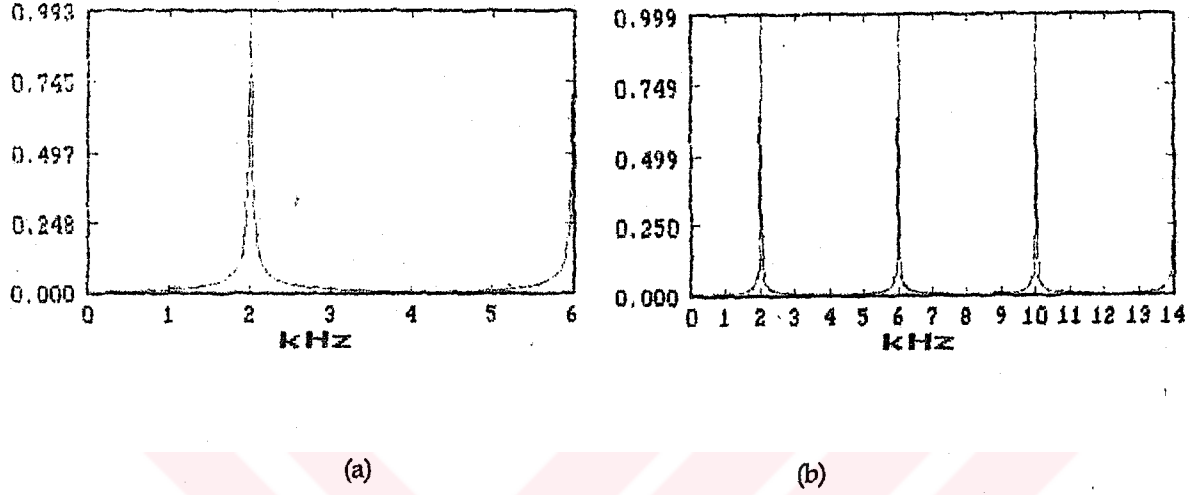
kazançların hemen hemen aynı olduğu Şekil 3.10 b'deki süzgeç karakteristiği elde edilmiş olur.

NIKOP programında KIS=8 vermek , OPTI ve GENKDI programlarında 1.aralık yerleştirme düzeninde N=2 durumuna karşı düşer. NY=11 ve istenen normalize harmonik genlikleri 0, 1, 0, 0, 0, 0 olarak girilirse GENKDI ve OPTI programlarıyla da bulunan ve açılının 11,20,25,34,55,64,69,78 aralıklarına yerleştirilmiş olduğu çözüm bulunur. Bu aralıkların ilk dördü pozitif, son dördü negatiftir. Bu fonksiyon kullanılarak süzgeçten elde edilen bant geçiren karakteristiğın merkez frekansında kazanç 0.693 olarak hesaplanmıştır.

Benzer şekilde 2. harmonik genliğinin büyük diğerlerinin küçük olduğu anahtar fonksiyonu aranırsa -,-,+,,+,-,-,+,,+ biçimindeki işaret kombinasyonu ile 8,15,30,37,52,59,74,81 aralıkları bulunur. Bu fonksiyonla elde edilen filtre karakteristiği Şekil 3.11 a'da gösterildiği gibidir. Aynı problem 4. harmoniğın büyük olması için girilirse önceden de görülebileceği gibi aynı aralıklar bu defa , +, -, +, -, +, -, +, - işaretleriyle çözüm olarak bulunur ve Şekil 3.11 b'deki karakteristiği elde etmemizi sağlar.

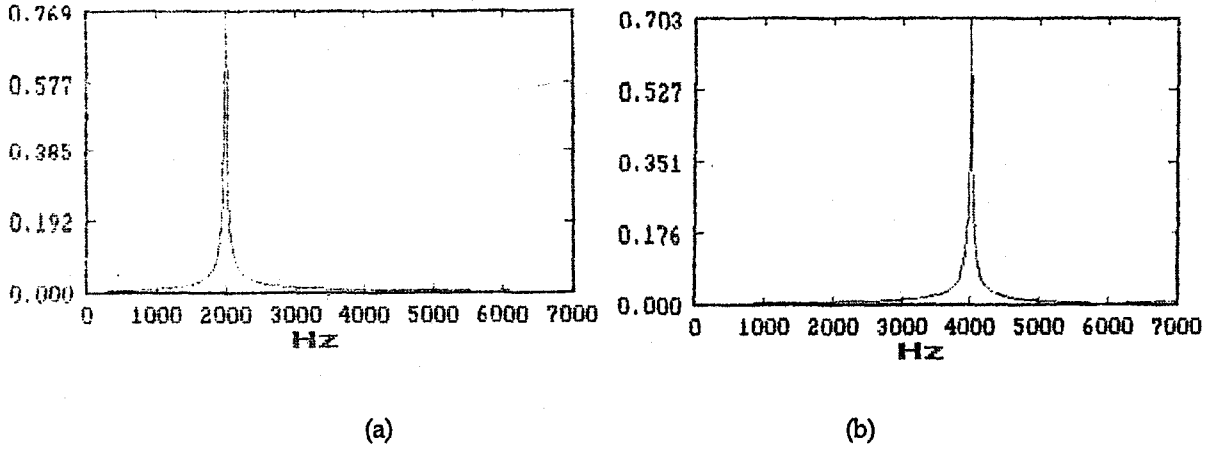


Şekil 3. 9 Her bir yolun bir kez devreye alındığı klasik uygulamayla elde edilen 19-yollu süzgeç karakteristiği



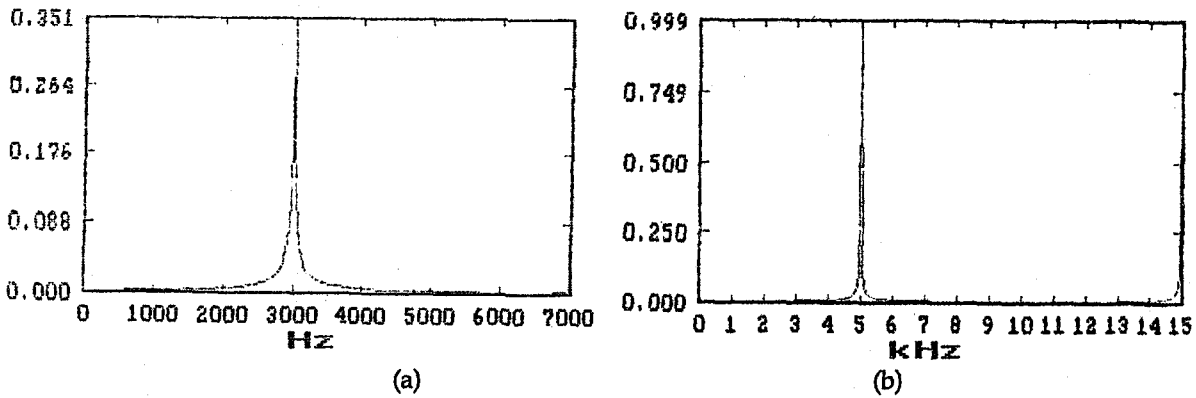
Şekil 3. 10 (a) -6, 17, -28, 39 ve $NY=11$, (b) -14, 41, -68, 95 ve $NY=27$ için elde edilen transfer karakteristikleri

Benzer araştırma $KIS=10$, $M=5$, $NY=11$ için yapılırsa, beşinci harmoniğe kadar istenen normalize genlik değerleri 0, 1, 0, 0, 0, 0 için bulunan en iyi çözüm, diğer programlarla da bulunan ve ilk beş aralığın pozitif, son beşinin negatif olduğu 12,21,28,35,44,67,76,83,90,99 simetrik aralıklarıdır. Elde edilen süzgeç karakteristiğinin anahtarlama frekansına yerleşmiş olan merkez frekansındaki genlik değeri 0.6603'dür. 3. harmoniğin en büyük diğerlerinin küçük olduğu çözüm 10,27,28,35,44,67,76,83,90,99 aralıklarıdır ve işaretler sırasıyla - ve + şeklinde tekrarlanır.



Şekil 3. 11 (a) -8, -15, 30, 37, -52, -59, 74, 81 (b) 8, -15, 30, -37, 52, -59, 74, -81 , NY=11 ve $f_o=1000$ Hz için elde edilen transfer karakteristikleri

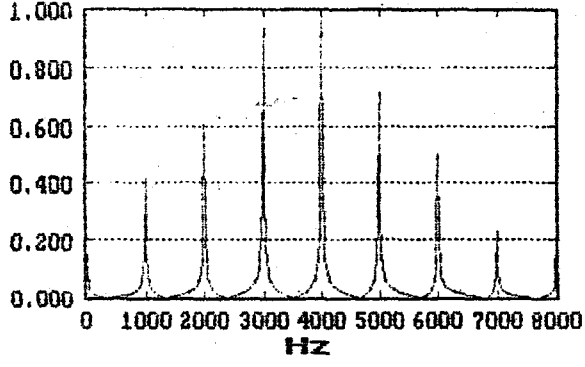
Bu anahtar fonksiyonu ile 11-yollu süzgeçten elde edilen karakteristik Şekil 3.12 a'da gösterilmiştir. Benzer şekilde 5. harmoniği büyük yapan aralıklar -6, 17, -28, 39, -50, 61, -72, 83, -94, 105 şeklindedir. Bu çözüm aslında impulsların ilk çeyrek periyotta 18, 54 ve 90 dereceye yerleşmiş olduğu özel bir durumdur ve bu fonksiyonun 5, 15, 25, 35, ... numaralı harmoniklerinden başka bütün harmoniklerinin değeri sıfırdır.



Şekil 3. 12 (a) -10, 27, -28, 35, -44, 67, -76, 83, -90, 99 , NY=11 (b) -15, 44, -73, 102, -131, 160, -189, 218, -247, 276 , NY=29 ve $f_o=1000$ Hz için elde edilen transfer karakteristikleri

Yol sayısı yeteri kadar yüksek tutularak örneğin, $NY=29$ için ilk çeyrek periyottaki aralıkların $-15, 44, -73$ olduğu anahtar fonksiyonuyla elde edilen süzgeç karakteristiği Şekil 3.12 b'de gösterilmiştir

Harmonik genliklerinden normalize değer olarak 0 ile 1 arasında herhangi bir sayı istenebilir. Örneğin $KIS=10$, $M=8$ ve $NY=19$ için 8. harmoniğe kadar 0.2, 0.4, 0.6, 0.8, 1, 0.8, 0.6, 0.4, 0.2 istenirse NIKOP bilgisayar programı tarafından bulunan açılış yerleri ve yönleri $28.96, -66.98, -103.51, -123.71, 158.00, -209.77, -222.96, 261.51, -285.12, 339.35$ derecedir. Bu açılışların düştüğü aralıklar ve yönleri $16, -36, -55, -66, 84, -111, -118, 139, -151, 180$ olup bu fonksiyonun filtrede kullanılmasıyla üreteceği karakteristik istenen değerlere çok yakın olabileceken, çakışma nedeniyle çözüm olarak kullanılamaz. Bu durumda çözüm etrafında çakışma olmayan aralık kombinasyonları için tarama yapılabilir. Böylece bulunan $15, -36, -54, -67, 83, -110, -118, 139, -152, 181$ aralıklarından oluşan anahtar fonksiyonunun, örnekler için hep aynı değerlerle kullanılmakta olan filtreye uygulanmasıyla alacağımız karakteristik Şekil 3.13 de gösterilmiştir. Çakışma nedeniyle vazgeçmek durumunda kaldığımız anahtar fonksiyonunun, süzgecin transfer fonksiyonunda anahtarlama frekansının harmoniklerine karşı düşen frekanslarda, istenen normalize genlik değerlerine çok iyi yaklaşmasına rağmen, çözüm etrafı taranarak bulunan çözümde istenen değerlerden biraz uzaklaşmıştır. Çünkü çözüm olarak ele alınan aralıkların orta noktaları $27.47, 67.26, 101.36, 126.00, 156.31, 207.47, 222.63, 262.42, 287.05, 342.00$ derece olup orijinal yerlerinden farklıdır.



Şekil 3.13 15, -36, -54, -67, 83, -110, -118, 139, -152, 181 anahtarlama fonksiyonu için 19-yollu süzgeç transfer karakteristiği

4. GENKDI PROGRAMININ TANITIMI

4.1. Giriş

Tezin önceki bölümlerinde anlatıldığı gibi, özellikle Bölüm 3'ün sonlarındaki Tablo 3.1'den anlaşılın, bir periyot içinde pulsların bulunduğu aralıkların, çözüm olabilmesi için hem yollar arasında çakışma olmaması şartını hemde bastırılmak istenen harmoniklerde yeteri kadar zayıflamanın olması şartını birlikte sağlaması gerektiğidir. Olabilecek bütün aralık kombinasyonları için bu şartların araştırılması aranılan çözümü bulmak için bir yöntem olabilir. Aralıklar çeyrek periyot içinde iki farklı biçimde düzenlenebilir. Birinci düzende yeri sabit olan aralık yoktur ve son aralığın bitiş noktası 90 derecedir. İkinci düzende ise son aralığın yeri sabit olup orta noktası 90 derece olacak şekilde yerleştirilmiştir. Çeyrek periyot içinde kullanılacak aralık sayısı, düzeni ve yol sayısı bilindiğine göre denenecek aralık kombinasyonu sayısı T, aşağıdaki bağıntıyla hesaplanabilir.

$$T = \sum_{k_1=1}^{\mu-n} \sum_{k_2=k_1+1}^{\mu-n+1} \sum_{k_3=k_2+1}^{\mu-n+1} \dots \sum_{k_n=k_{n-1}+1}^{\mu} \quad (4.1)$$

Bu bağıntıdaki n, çeyrek periyot içindeki serbest aralık sayısıdır. Diğer değişken olan μ aşağıdaki bağıntı ile hesaplanır.

$$\mu = (NY)(KIS)/4 + K \quad (4.2)$$

Burada, NY yol sayısı, KIS ise bir periyotta kullanılan aralık sayısı olup ikinci düzen kullanıldığında $4N+2$, birinci düzen kullanıldığında $4N$ 'dir. Benzer şekilde K, ikinci düzen için 0,5, birinci düzen için 0'dır. Araştırılan çözümden beklenen, birinci harmoniğin büyük,

diğer harmoniklerin küçük olmasıdır. Bu nedenle yukarıda ikinci düzen olarak adlandırılan aralık yerleştirme biçimi daha avantajlı görülmektedir.

Bu işlemleri gerçekleştirmek üzere GENKDI isimli Fortran bilgisayar programı hazırlandı. Olabilecek bütün aralık kombinasyonları için çakışma kontrolunu ve istenen zayıflamaların sağlanıp sağlanmadığının kontrolunu yaparak şartları sağlayan aralık kombinasyonlarını, bu aralıklar için harmonik genliklerini, zayıflamalarını ve normalize değerlerini yazan program ana altprogram şeklinde hazırlanmıştır. Biri fonksiyon alt programı olmak üzere 14 altprogram kullanır. Bu altprogramlar tanıtıldıktan sonra GENKDI ile bulunan bazı sonuçlar verilmiş ve elde edilen dar bantlı bant geçiren filtrelerin karakteristikleri çizilmiştir.

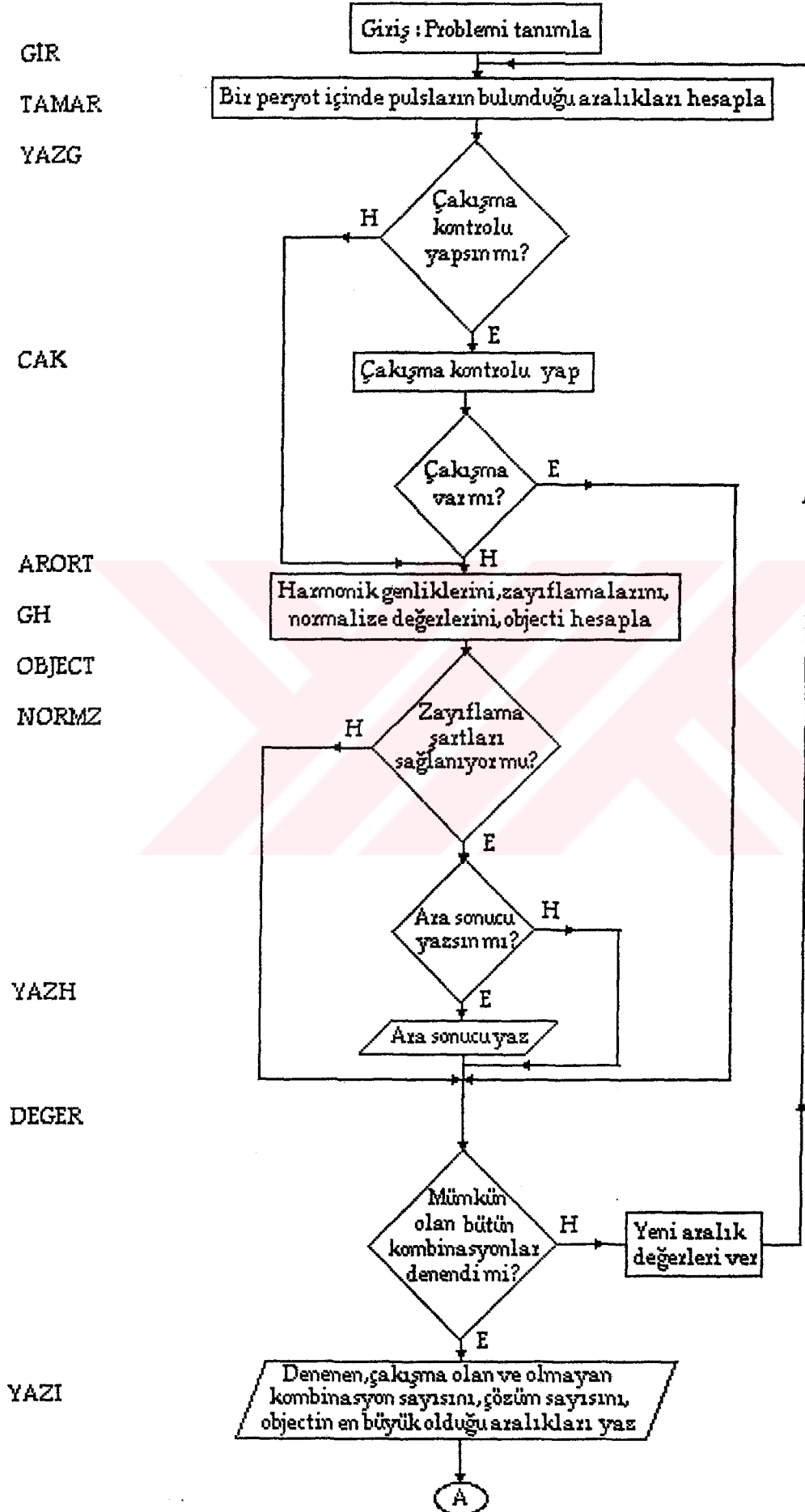
4.2. Ana Program GENKDI ve Değişken İsimleri

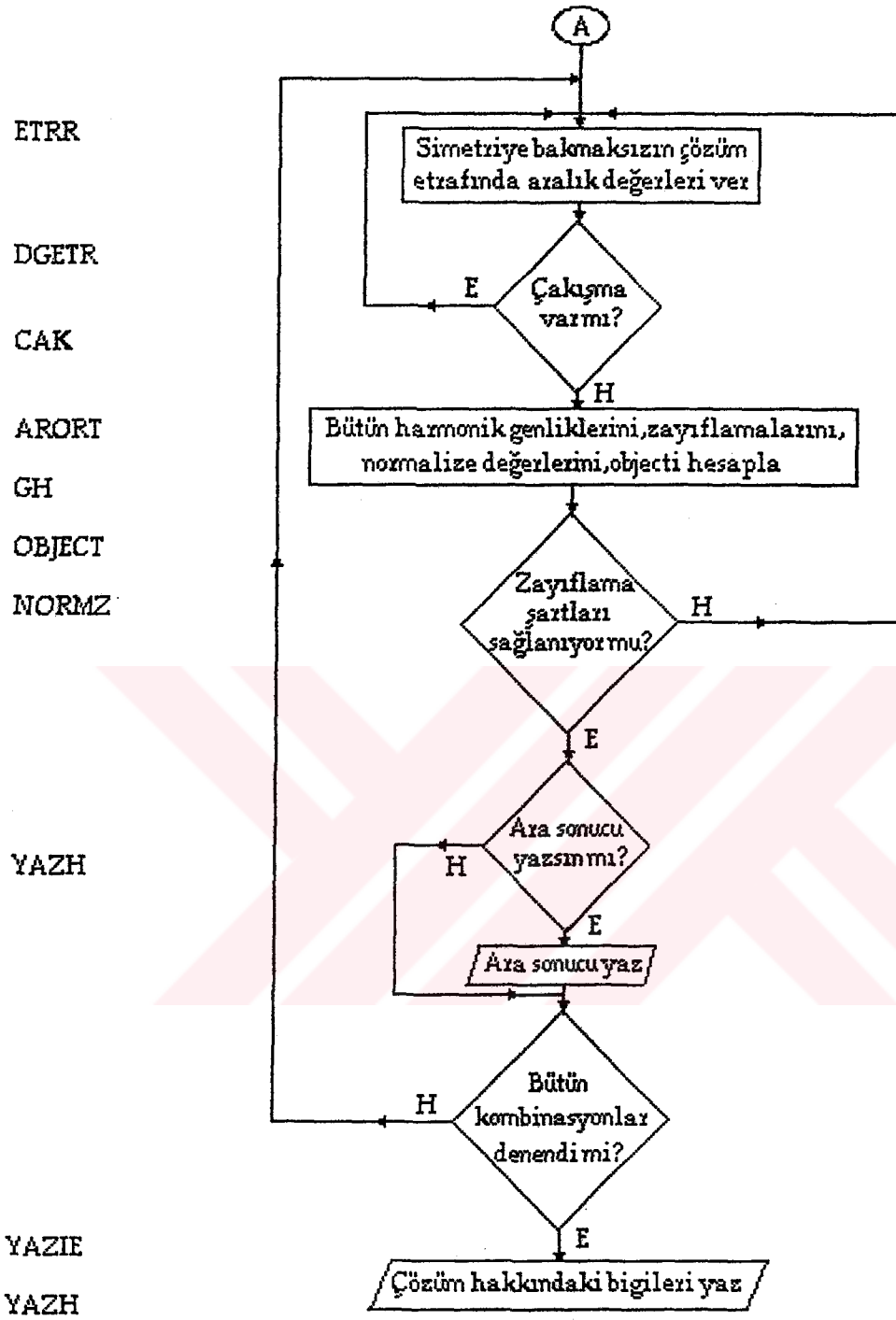
GENKDI programı tarafından üretilecek sonuçların başka bir programa aktarılabilmesi için bu program bir ana altprogram olarak hazırlanmıştır. Programın çalıştırılabilmesi için aşağıdaki çağırın programın kullanıcı tarafından yazılması gerekir.

CALL GENKDI (Argümanlar)

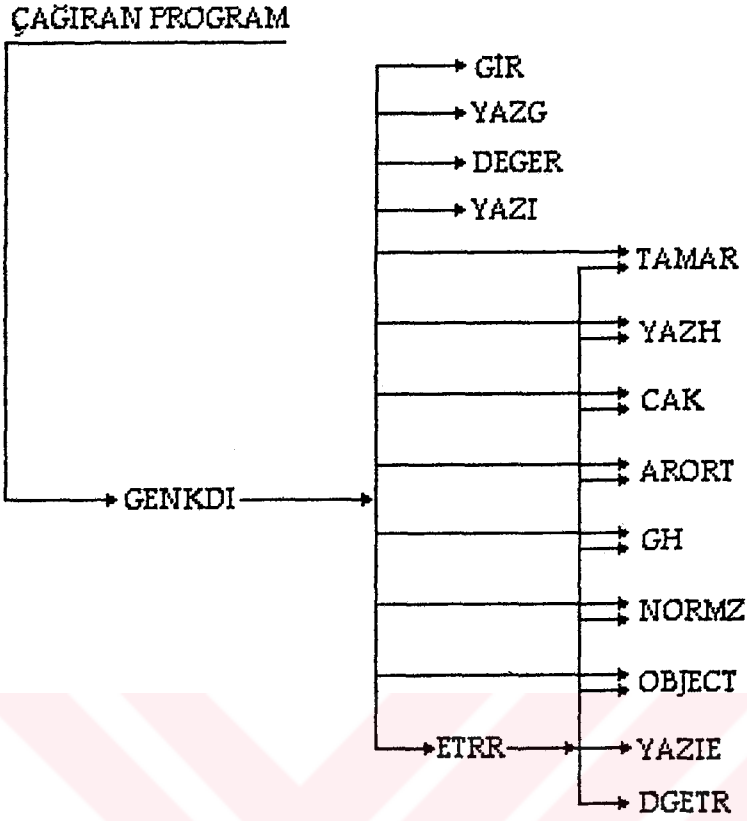
STOP

END





Şekil 4.1 GENKDI programının akış şeması



Şekil 4.2 GENKDI tarafından kullanılan altprogramlar ve çağırma ilişkileri

GENKDI programının altprogramları ile ilgili detaylara girmeden önce bu programın genel bir akış şeması Şekil 4.1 'de verilmiştir.

GENKDI tarafından kullanılan altprogramlar ve çağırma ilişkileri Şekil 4.2'deki şemadan açıkça görülmektedir. Şimdi bu programda kullanılan ve kullanıcı tarafından değerleri klavyeden girilen değişkenler açıklanacaktır.

KSE : Önceki bölümde anlatılan yöntemde, çeyrek ve yarı dalga simetrisine sahip olan ve Bölüm 3'de Şekil 3.1 a ve b'de verilen bu bölümün giriş kısmında açıklanan aralık yerleştirme düzenlerini tanımlayan tamsayıdır. KSE'nin 1 değeri $\pi / 2$ de aralık olmayan (1. düzen), KSE'nin 2 değeri ise $\pi / 2$ de , $\pi / 2$ orta nokta olacak şekilde o aralığa bir darbe yerleştirilmiş olan (2. düzen) durumları gösterir.

N : Çeyrek periyot içindeki serbest aralık sayısıdır.

NY : N- yollu süzgecin yol sayısıdır.

M : Bastırılmak istenen yada başka bir deyişle, genliğinin küçük olmasını istediğimiz en büyük harmonik numarasıdır. M, tek sayı olup 3'den M'e kadar tek harmoniklerin gözönüne alınacağını gösterir.

ZREF : 3'den M'e kadar tek harmoniklerin, ana harmonik bileşeninin genlik değerine göre normalize edilmiş değerlerinin sağlaması gereken referans zayıflamaları içeren vektördür.

LCP : Çeyrek ve yarı dalga simetrisi olan aralık kombinasyonlarının her biri için diğer yollar gözetilerek çakışma kontrolunun yapılıp yapılmayacağını gösteren sayıdır. LCP, 1 verilirse çakışma kontrolü yapılır, 0 verilirse çakışma kontrolü yapılmaz.

LY : Zayıflama referanslarını sağlayan ara çözümlerin yazdırılıp yazdırılmayacağını gösteren tamsayıdır. LY , 1 verilirse ara çözümler yazılır, 0 verilirse yazılmaz.

K1 : Bulunan çeyrek ve yarı dalga simetrisine sahip çözümün etrafında taranacak bölgeyi sınırlayan çift tamsayıdır.

ZREFE : Simetri gözetilmeksizin araştırılan bölgede 2'den M'e kadar bütün harmoniklerin sağlaması istenen zayıflama değerlerini içeren vektördür.

AR : Çeyrek periyot içinde pulsların bulunduğu başlangıç aralıklarını içeren vektördür. Bu değerler program içerisinde kendiliğinden belirlenir.

Şekil 4.1'den görüldüğü gibi GENKDI'nın ilk altprogramı GIR olup yukarıda açıklanan değişkenlerin değerlerini kullanıcının girmesini sağlar.

4.3 YAZG Altprogramı

Programın hangi giriş değerleri için çalıştırıldığını belirtmek üzere GIR altprogramı ile girilen değişkenlerin değerlerini yazar.

4.4 TAMAR Altprogramı

Çeyrek periyot içinde verilen aralıkları, bir periyodun bölündüğü aralık sayısını, çeyrek ve yarı dalga simetrisi olan şekli gözönüne alarak bir periyot içindeki pulsların bulunduğu aralıkları belirler. Bu altprograma aktarılan değişkenler (N,NY,AR,KSE) olup yukarıda açıklanmıştır. Burada AR bir periyot içindeki aralıkları gösteren bir vektöre dönüşür. Bu altprogramın kendi içinde kullandığı vektöryel olmayan değişkenlerin açıklanmasına gerek görülmemiştir.

4.5. CAK Altprogramı ve Değişken İsimleri

TAMAR altprogramı bu altprograma bir periyot boyunca 1. yolun çalıştığı aralık değerleri olarak ele alınan kombinasyonu aktarır. Bu kombinasyon, yol sayısı ve bir periyodun bölündüğü aralık sayısı gözönüne alınarak kaydırılır ve diğer yolların çalıştığı aralıklar bulunur. Ele alınan her bir yolla, diğer yolların herhangi biri aynı aralıkta çalışıyorsa bu durumda çakışmanın olduğu belirlenir. Aksi halde çakışma yoktur ve bu kombinasyon zayıflama şartlarını da sağlıyorsa çözüm olabilir.

Bu altprogramda kullanılan argümanlar (KIS,NY,AR,KY,KC) olup daha önce açıklanmamış olanları aşağıda verilmiştir.

KIS : Bir periyot içindeki aralık sayısını gösteren tamsayıdır.

KY : Bütün yolların çalıştığı aralıkları içeren vektördür.

KC : Bu altprogramın işlemesi sonucu 0 veya 1 değerini alan tamsayıdır. KC=0 ise bu aralık kombinasyonu için çakışma yok, 1 ise çakışma-var demektir.

4.6. ARORT Altprogramı ve Değişken İsimleri

TAMAR altprogramı ile belirlenen aralık kombinasyonu için bu aralıkların orta noktalarının değerlerini hesaplar.

Bu altprogramın argümanları (KIS,NY,AR,TETA) olup ilk üçü daha önce açıklanmıştır.

TETA : Bir peryot boyunca verilen aralıkların orta noktalarının radyan olarak değerlerini içeren vektördür.

4.7. GH Altprogramı ve Değişken İsimleri

Bölüm 3'de verilen yaklaşım ve bağıntılar üzerinden ele alınan aralık kombinasyonunun harmonik genliklerini hesaplar. Bu altprogramda ele alınan her bir aralık kombinasyonu için Fourier açılımındaki katsayılar verilen harmonik numarasına kadar hesaplanır. Bölüm 3'de 3.5 a ve b 'de çeyrek peryot için yazılmış olan denklemler bütün harmonikleri ve bütün peryodu kapsayacak şekilde genişletilmiştir. Böylece çeyrek ve yarı dalga simetrisi olması şartına bakılmaksızın aynı yaklaşımla, herhangi bir aralık kombinasyonu için harmonik genlikleri hesaplanabilir. Denklem 3.5 a ve b'deki ifadeler esas alındığında açılar, ele alınan aralık kombinasyonundaki her bir aralığın orta noktası olarak kabul edilmiştir. Ancak daha doğru bir hesaplama aralıkların başladığı ve bittiği açı değerleri kullanılarak yapılabilir. Böyle bir program yazılmış ve listesi Ek***de verilmiştir. Aralık sayısının küçük olduğu durumlarda bu aralıkların orta noktalarını olarak fonksiyonu çok iyi bir yaklaşımla tanımlamak mümkün olmadığından, discrete Fourier analizinde olduğu gibi, harmonik genlikleri çok doğru olarak hesaplanamaz. Aralık

sayısının yeteri kadar büyük olduğu durumlarda her iki altprogramla hesaplanan harmonik genliklerinin değerleri yaklaşık eşittir.

Sayısal örnek olarak önce bir periyodun 10 eşit aralığa bölüdüğü ve ilk beşinin pozitif, kalanların ise negatif olduğu durumu ele alalım. GH altprogramı ile hesaplanan 1., 3. ve 5. harmonik genlikleri sırasıyla 1.29442, 0.49442 ve 0.400000 olarak bulunur. Gerçekte ele alınan şekil bir kare dalga olup SB2A altprogramı ile hesaplanan aynı numaralı harmonik genlikleri, yine sırasıyla, 1.27324, 0.424413 ve 0.25464 dür. Aralık sayısının yeteri kadar büyük olduğu duruma örnek olarak bir periyodun 90 aralığa bölüdüğü ve Bölüm 3 Tablo 3.1'de çözüm olarak verilen 10, 17, 23, 29, 36, 55, 62, 68, 74, 81 aralıklarını ele alalım. Benzer şekilde ilk beş aralığın pozitif, kalanların negatif olduğu bu aralıklar için GH altprogramı ile hesaplanan aynı numaralı harmonik genlikleri sırasıyla 0.18037, 0.00929 ve 0.01543 iken SB2A altprogramı kullanılarak hesaplananlar 0.18033, 0.009274 ve 0.01535' dir. Yani aralık sayısının yeteri kadar büyük olduğu durumda harmonik genliklerinin GH altprogramı ile hesaplanması, başka bir deyişle Fourier açılımının yapılması büyük hatalara yol açmaz.

Bu altprogramın değişkenleri (NY,M,KIS,TETA,A,B,AB,MF) olup ilk üçü daha önce açıklanmıştır.

A : Fourier açılımında kosinüslü terimlerin katsayılarını gösteren vektördür.

B : Fourier açılımında sinüslü terimlerin katsayılarını gösteren vektördür.

AB : Fourier açılımında harmonik bileşenlerinin genliklerini gösteren vektördür.

MF : 1 veya 2 değerlerini alabilen bir tamsayıdır. MF=1 ise bütün harmonikler, MF=2 ise sadece tek harmonikler hesaplanır. Böylece gereksiz hesaplamadan kaçınılarak zaman tasarrufu sağlanmış olur.

4.8. NORMZ Altprogramı ve Değişken İsimleri

NORMZ altprogramı GH altprogramı tarafından hesaplanan harmonik genliklerinin en büyük genliğe sahip olana göre normalize edilmiş değerlerini ve bunlar üzerinden dB olarak zayıflamalarını hesaplar.

Bu altprogramın değişken isimleri (M,AB,ABN,ZYF,MF) dir. Bunlardan 3. ve 4. sü aşağıda açıklanmıştır.

ABN : Harmonik genliklerinin normalize değerlerini içeren vektördür.

ZYF : Harmoniklerdeki zayıflama değerlerini içeren vektördür.

4.9. OBJECT Altprogramı ve Değişken İsimleri

En iyi çözümü seçmek amacıyla düzenlenen bir parametreyi hesaplayan fonksiyon altprogramıdır. Büyük bir OBJ için ana harmonik genliğinin büyük, bastırılmak istenen harmonik genliklerinin ise verilen referansa yakın veya daha fazla zayıflamayı sağlamış olması gerekir.

Bu altprogramda kullanılan değişkenler (M,ZREF,ZYF,AB,OBJ,MF) olup, daha önce açıklanmayan,

OBJ : En iyi çözümü seçmek için kullanılan ve yukarıdaki değişkenlere bağlı olan bir parametredir.

4.10. YAZH Altprogramı

İstenirse ara çözümleri yazan bu altprogramın değişkenleri (KIS,AR,M,AB,ABN,OBJ,ZYF,LCP,MF) olup daha önce açıklanmıştır.

4.11. DEGER Altprogramı ve Değişken İsimleri

Çeyrek periyot içinde olabilecek aralık kombinasyonu değerini verir. Daha önce açıklanan N ve AR 'dan başka diğer iki değişkeni aşağıda verilmiştir.

KNT : 0 ve 1 değerlerini alabilen bir tamsayıdır. KNT=1 ise bütün kombinasyon değerleri verilmiş demektir. KNT=0 ise yeni aralık kombinasyonu değerleri verilir.

ARK : Bir periyottaki aralık sayısı ARS olmak üzere $(ARS/4) + 1$ ile belirlenen bir tamsayıdır.

4.12. YAZI Altprogramı ve Değişken İsimleri

Aşağıda verilen değişkenlerin değerlerini yazar.

AKMBS : Denenen aralık kombinasyonu sayısıdır.

CAKS : Çakışma kontrolü yapılmışsa çakışma olmayan aralık kombinasyonu sayısıdır.

SAY : Çakışma kontrolü yapılmışsa çakışma olmayan ve referans zayıflamaları sağlayan, çakışma kontrolü yapılmamışsa sadece referans zayıflamaları sağlayan aralık kombinasyonu sayısıdır.

OBJMAX : En iyi çözüm için OBJ 'nın aldığı değerdir.

4.13. ETRR Altprogramı

Seçime bağlı olarak, çakışma kontrolü yaparak veya yapmadan bulunan çeyrek ve yarı dalga simetrisi olan çözüm etrafında simetriyi sağlamayan aralık kombinasyonları da dahil olacak şekilde, K1 ile sınırlı bölge içerisinde yaptığı tarama ile en büyük OBJ değerini veren ve çözüm olan aralık kombinasyonunu seçer. Simetri gözetmeksizin çözüm etrafı K1 ile sınırlanan bölgede taranırken denenen kombinasyon sayısı, $(K1+1)^{KIS}$ bağıntısı ile belirlendiğinden KIS'nin büyük değerleri için çalışması çok zaman alır. Örneğin, çift sayı olan K1'in 2, KIS'nin 10 değeri için denenecek kombinasyon sayısı 59049 iken K1'in aynı, KIS'nin 22 değerini aldığı durum için 10^{10} 'lar mertebesine ulaşır. Bu durum göz önüne alınarak program, ETRR altprogramının çalıştırılıp çalıştırılmayacağını kullanıcıya soracak şekilde hazırlanmıştır.

ETRR altprogramı Şekil 4.2'den görülebileceği gibi 9 altprogram kullanır. Bunlardan sadece ETRR tarafından kullanılan YAZIE ve DGETR altprogramları hakkında açıklamalar aşağıdaki kısımlarda verilmiştir.

Bu altprogram daha önce açıklaması yapılan tüm değişken isimlerini kullanır. Bunların dışında kalan değişken isimlerinin açıklaması aşağıdaki altprogramların değişken isimlerinde verilmiştir.

4.14. YAZIE Altprogramı ve Değişken İsimleri

Sadece ETRR altprogramının kullandığı bir altprogramdır. Aşağıda açıklaması verilen değişkenlerin aldığı değerleri yazar.

AKMBG : Çözüm etrafı sınırlı bir bölgede araştırılırken denenen aralık kombinasyonu sayıdır.

AKCY : Bu bölgede denenen kombinasyonlardan çakışma olmayanların sayısıdır.

SAY : Bu sınırlı bölgede hem çakışma olmayan hemde 2 den M'e kadar bütün harmoniklerde istenen ZREFE zayıflamalarını sağlayan kombinasyon sayısıdır.

4.15. DGETR Altprogramı ve Değişken İsimleri

Sadece ETRR altprogramının kullanılan ve çözüm etrafında araştırılacak aralık değerlerini veren altprogramdır.

Bu altprogramın değişken isimleri (KIS,AR,ARG,K1,L) olup daha önce açıklanmayanlar aşağıda verilmiştir.

ARG : Simetrik çözümden K1 ile bir periyot için hesaplanan aralık değerlerini içeren vektördür.

ARM : En iyi çözümü veren aralıkları gösteren vektördür.

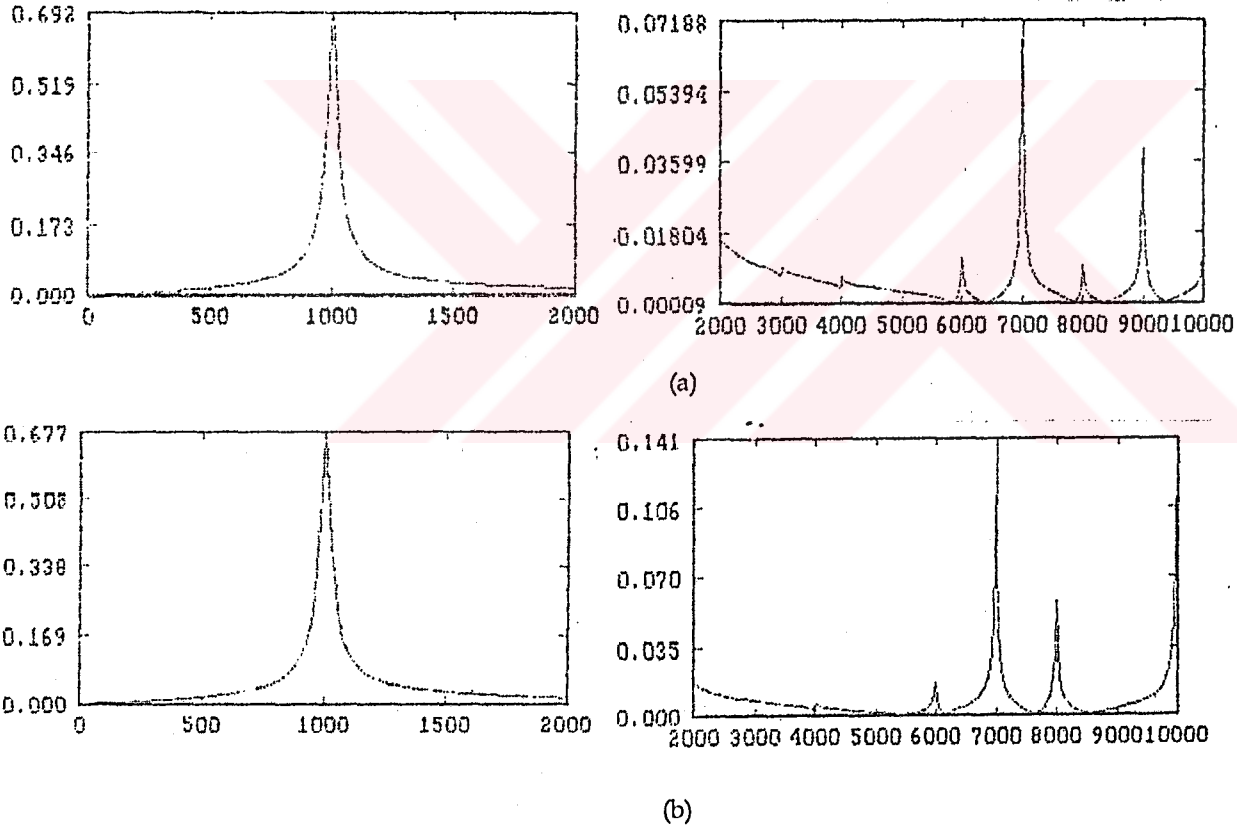
L : Çözüm etrafındaki sınırlı bölgede denenecek kombinasyonun bitip bitmediğini gösteren tamsayıdır. L = 0 ise yeni değerler verilir, L = 1 ise bütün kombinasyonlar denenmiş demektir.

4.16. GENKDI Programı ile Bulunan Bazı Çözümler

İlk örnek olarak Bölüm 3'de anlatılan ve açılı yerleri bulunduktan sonra, açılımların düştüğü aralıkların çözüm olup olmayacağını tekrar tekrar yapılan deneme ve iyi bir sezgi ile belirlendiği problemi ele alalım (Köksal, 1975). Gerçekte, 2. aralık yerleştirme düzeninde çeyrek periyot içinde 2 tane serbest aralık ($N=2$) kullanarak, (4.1)'den hesaplanan $T=231$ aralık kombinasyonu için, 3. ve 5. harmonik genliklerinde 20 dB zayıflama şartını ve çakışma olmaması şartını birlikte sağlayan sadece 2 çözüm vardır. Bunlardan birincisi, 2,15,23,31,44,47,60,68,76,89, ikincisi, 10,17,23,29,36,55,62,68,74,81 aralıklarıdır. Birinci çözüm için GENKDI programı içinde hesaplanan 1., 3. ve 5. harmonik genlikleri sırasıyla 0.12911, 0.00768 ve 0.00536 olup, 3. ve 5. harmonik zayıflamaları 1. harmoniğe göre 24.50 ve 27.63 dB'dir. Benzer şekilde 2. çözüm için 1. harmonik genliği 0.18037, 3. ve 5. harmonik genlikleri ise 0.00929 ve 0.01543 olup 25.76 ve 21.35 dB zayıflama değerindedir. Görüldüğü gibi ikinci çözüm 1. harmoniğinin büyüklüğü nedeniyle tercih edilmelidir. Bu bant geçiren süzgecin 4.5 kHz'e kadar geçerli olan karakteristiği Bölüm 3, Şekil 3.4'de $R=5$ k Ω , $C=0.2$ μ F için çizilmiştir. Bu karakteristikte 3. ve 5. harmonikler için zayıflama değerleri dB olarak 38.27 ve 41.61'dir. Alt devrelerin RC çarpımını 10 kat artırırsak veya RC aynı değerde iken anahtarlama frekansını 10 kat artırırsak bu durumda 3. ve 5. harmonik genliklerindeki zayıflamalar 50.75 ve 42.77 dB olarak bulunur. Bu değerler yukarıda ikinci çözüm olarak verilen aralık kombinasyonu için GENKDI programı içinde hesaplanan zayıflama değerlerinin iki katıdır. Başka bir deyişle anahtarlama fonksiyonunun normalize edilmiş harmonik genliklerinin kareleri, süzgecin sağlayabileceği en yüksek zayıflamaların (anahtarlama frekansının yeterince yüksek, alçak geçiren alt devrelerin yeterince düşük kesme frekansının) olduğu durumda, karakteristiğinin o harmonik frekanslarındaki normalize genlik değerine eşittir. Sayısal örnek olarak alt devrelerin $R=50$ k Ω , $C=0.2$ μ F olan alçak geçiren filtrenin 10 kHz anahtarlama frekansında çalıştırılmasıyla elde edilen

karakteristiğın 3. ve 5. harmonik frekanslarındaki normalize edilmiş genlik değerleri sırasıyla 0.00264757 ve 0.00725198 olduğu görülür. Anahtarlama fonksiyonunun SB2A altprogramı ile hesaplanan aynı numaralı harmoniklerinin normalize değerleri 0.00264487 ve 0.00725183'dür. Bu gerçek Bölüm 2 , Denklem (2.7)'de ifade edilmiştir.

GENKDI programıyla bu çözümden daha iyi bir çözüm araştırılmış ve çakışma kontrolü yapılmadan bulunan en iyi çözümün etrafı ETRR alt programında $K1=2$ verilerek 5. harmoniğe kadar tüm harmoniklerde en az 20 dB zayıflamayı sağlayan ve çakışma olmayan 11,19,24,28,36,55,62,67,73,80 aralıkları bulunmuştur. Bu anahtarlama fonksiyonu ve $R=5\text{ k}\Omega$, $C=0.2\text{ }\mu\text{F}$ değerleri için Bölüm 3, Şekil 3.3'de gösterilen filtrenin transfer fonksiyonunun değişimi, MAINLN programıyla hesaplatılarak Şekil 4.3 a'da gösterilmiştir.

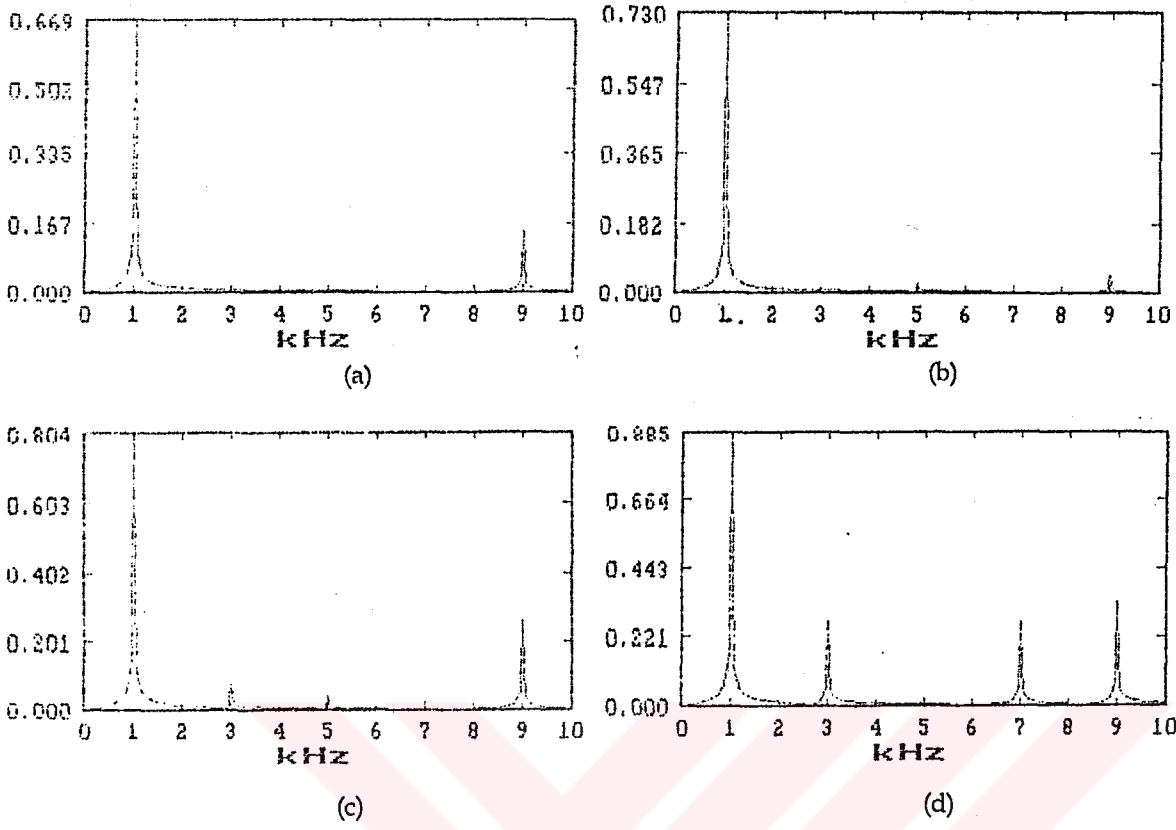


Şekil 4.3 Şekil 3.4 'deki çözüme alternatif olarak a) birinci aralık düzeninde, b) ikinci aralık düzeninde bulunan çözümler için transfer fonksiyonlarının değişimi

Şekil 4.3 b'de gösterilen karakteristik ise aynı problem için birinci aralık düzeni kullanıldığında bulunan 10,17,22,29,44,51,56,63 çözümünün ürettiğidir. Şekil 4.3 b'deki karakteristik bir periyot içinde 8 aralık kullanılarak elde edildiği halde 3. ve 5. harmonik frekanslarında 39.37 ve 51.97 dB zayıflama sağlar.

Şekil 4.3 a ve b'den görüldüğü gibi çeyrek ve yarı dalga simetrisinden vazgeçildiği için çift numaralı harmoniklerde de pikler ortaya çıkmıştır. Yinede a'daki transfer fonksiyonunun genliği merkez frekansı olan 1000 Hz'de Şekil 3.3'dekinden %5 daha büyük değer aldığından ve üstelik bu bant geçiren süzgeç karakteristiğide önceki bölümlerde anlatıldığı gibi 4.5 Khz'e kadar geçerli olduğundan Şekil 3.3 ve Şekil 4.3 b'dekine tercih edilmelidir.

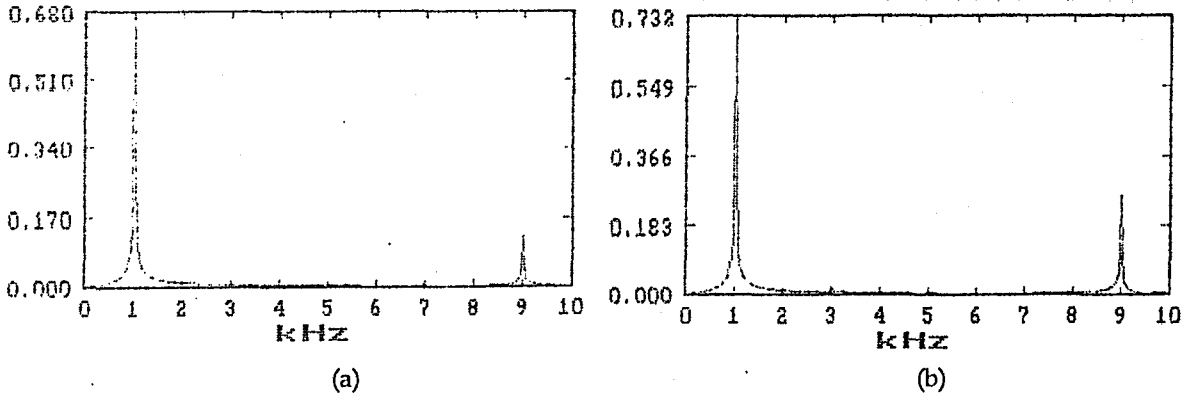
İkinci örnek olarak $N=3$, $NY=15$, $M=7$, 20 dB zayıflama değerleri için GENKDI çalıştırılırsa denenen $T=22$ 100 aralık kombinasyonundan 3375'inde çakışma olmadığı ve bunların 6 tanesinin bu zayıflama değerlerini sağladığı ve bunların içinden en uygun olan aralıkların 19,36,44,53,62,70,87,124,141,149,158,167,175,192 olduğu görülür. Bu anahtarlama fonksiyonu için önceki örnekte kullanılan eleman değerleri ve anahtarlama frekansı için MAINLN programı ile belirlenen transfer fonksiyonu (7.5 kHz'e kadar geçerli) Şekil 4.4 a'da gösterilmiştir. Bu karakteristikin 3, 5 ve 7 kHz'deki zayıflamaları sırasıyla 40.08, 40.96 ve 59.51 dB'dir. Benzer şekilde 15 dB zayıflama şartı ile bulunan anahtarlama fonksiyonu 24,37,43,53,63,69,82,129,142,148,158,168,174,187 aralıklarıdır. Bu fonksiyon kullanılarak elde edilen ve Şekil 4.4 b'de gösterilen süzgeç karakteristiğinde aynı frekanslar için sırasıyla 34.40, 30.21 ve 54.95 dB zayıflama vardır. Bu zayıflamaların yeterli olduğu düşünülürse 1 kHz'deki kazancın daha fazla olması gözönüne alınarak b'deki çözüm tercih edilir. Benzer işlemler 10 dB ve 5 dB zayıflama referansları verilerek yapılmış ve sırasıyla bulunan anahtarlama fonksiyonları 31, 35, 48, 53, 58, 71, 75, 136, 140, 153, 158, 163, 176, 180 ve 38,41,43,53,63,65,68,143,146,148,158,168,170,173 için Şekil 4.4 c ve d'de süzgeç karakteristikleri çizilmiştir.



Şekil 4.4 $N=3$, $NY=15$, $M=7$ ve ikinci düzen için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristikleri, a) 20, b)15, c) 10, d) 5 dB zayıflama referansları için.

Özellikle son karakteristikten belirgin olarak gözleendiği gibi, merkez frekansında kazanç artışı sağlanmış olmasına rağmen, 3 ve 7 kHz'deki pikler kabul edilemeyecek kadar büyümüştür.

Üçüncü örnek olarak, yukarıdakine alternatif olabilecek çözümleri bu defa birinci aralık düzenini kullanarak araştıralım. $N=3$, $M=7$, $NY=15$ ve 20 dB zayıflama referansları için bulunan anahtarlama fonksiyonu 19,32,40,51,59,72,109,122,130,141,149,162 aralıklarıdır. Yine 7.5 kHz'e kadar geçerli olan ve MAINLN programı ile hesaplatılan süzgeç karakteristiği Şekil 4.5 a'da gösterilmiştir.

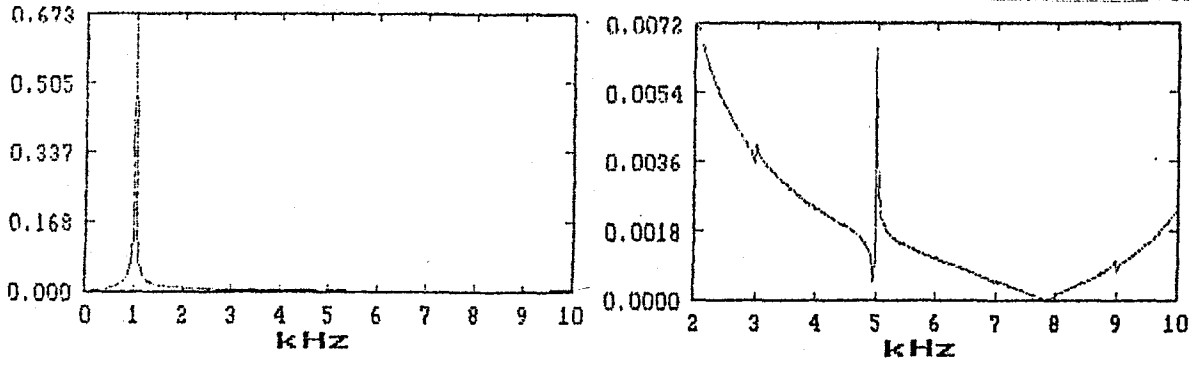


Şekil 4. 5 $N=3$, $NY= 15$, $M=7$ ve birinci düzen için Şekil 4.4'deki transfer karakteristiklerini üreten anahtarlama fonksiyonlarına alternatif olarak bulunan fonksiyonlarını kullanarak elde edilen süzgeç karakteristiği

Şekil 4.4 a'daki karakteristikle karşılaştırıldığında 1 kHz'de sağladığı daha yüksek kazanç , 3, 5 ve 7 kHz'de sırasıyla 43.00, 41.45 ve 42.13 dB gibi yüksek zayıflamalara sahip olduğu görülür.

Anahtarlama fonksiyonunun daha az sayıda aralıkla ifade edilmesi, pratikte daha az anahtarlama kaybı demek olduğundan Şekil 4.5 a'yı oluşturan anahtarlama fonksiyonu, Şekil 4.4 a'yı oluşturan anahtarlama fonksiyonuna tercih edilebilir. Şekil 4.4 b'de birinci aralık düzeni kullanılarak bulunan, 3, 5 ve 7 kHz'de sırasıyla, 31.52, 40.21 ve 54.56 dB zayıflamalarla Şekil 4.4 b'ye alternatif olabilecek süzgeç karakteristiği gösterilmiştir.

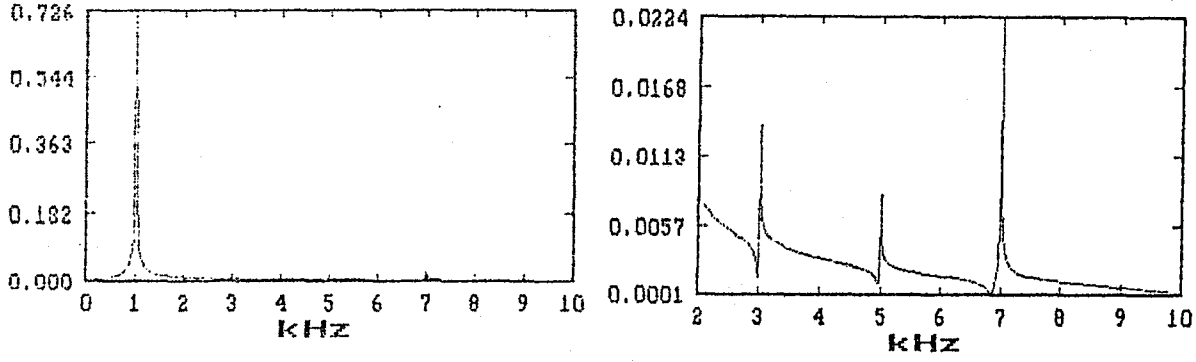
Dördüncü örnek olarak $N=4$, $NY= 19$, $M=9$, 20 dB zayıflama değerleri için GENKDI çalıştırılırsa, $T=2\ 024\ 785$ aralık kombinasyonunun denenmesi için gereken zaman 80386 prosesörü ve 80387 koprocesörü olan bir kişisel bilgisayarda 1 saat 43 dakikadır. Denenen aralık kombinasyonlarının 76 tanesi çakışma ve zayıflama şartlarının sağlar. En iyi çözüm, aralıkların 30,55,56,76,86,96,116,117,142,201,226,227,247,257,267,287,288,313'de olduğu durumdur. Bu anahtarlama fonksiyonu için 19-yollu süzgecin üreteceği karakteristik Şekil 4.6'da gösterilmiştir. Bu transfer fonksiyonunun 3., 5., 7., ve 9. harmonik frekanslarındaki zayıflamaları sırasıyla 44.65, 40.26, 64.08 ve 56.86 dB olarak hesaplanmıştır.



Şekil 4. 6 $N=4$, $NY= 19$, $M=9$ için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği

Beşinci örnek olarak $N=3$, $NY= 19$, $M=9$, 15 dB zayıflama değerleri için GENKDI çalıştırılırsa, $T=45760$ aralık kombinasyonu denenerek, ikinci aralık yerleştirme düzeninde birinci yolun çalıştığı aralıklar, 30,45,57,67,77,89,104,163,178,190,200,210,222,237 olarak bulunur. Bu anahtarlama fonksiyonu kullanılarak analizi yapılan süzgecin 8.5 kHz'e kadar geçerli olan karakteristiği Şekil 4.7'de gösterilmiştir. Bu şekil, Şekil 4.6 ile karşılaştırıldığında merkez frekansında sağladığı daha yüksek kazanç ve bastırılması istediğimiz frekanslarda (3 Khz'de 34.40 dB, 5 kHz'de 38.93 dB, 7 kHz'de 30.20 dB, 9 kHz'de 59.74 dB) yeterli zayıflamalarla, daha az anahtarlama ile ulaşılmış bir sonuç olarak öncesine tercih edilebilir.

Altıncı örnek olarak yukarıda bulduğumuz sonuçtan cesaret alarak, $N=3$, $NY=23$ ve $M=11$ için çözüm aranması halinde $T=82160$ aralık kombinasyonu denenerek en uygun anahtarlama fonksiyonu olarak, 31,49,65,81,97,113,131,192,210,226,242,258,274,292 bulunur. Alt devrelerin aynı R ve C değerleri için 11.5 kHz'e kadar geçerli olan filtre transfer fonksiyonu Şekil 4.8'de gösterilmiştir.

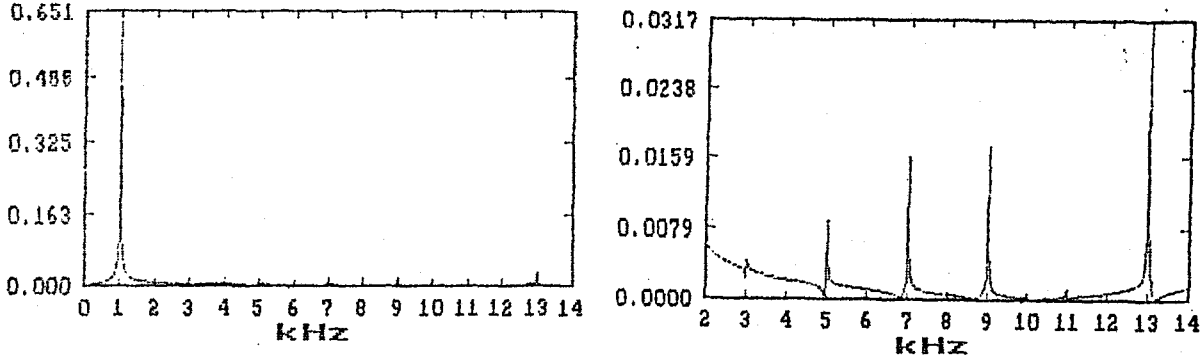


Şekil 4. 7 $N=3$, $NY= 19$, $M=9$ için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği

Şekil 4.7'nin sağ tarafında ölçek farkı ile ayrıntılı olarak incelenen ve yüksek zayıflamaların istendiği frekans bölgesinde, en az zayıflama 7. harmonik frekansına karşı düşen 7 kHz'de ve 30.20 dB değerindedir.

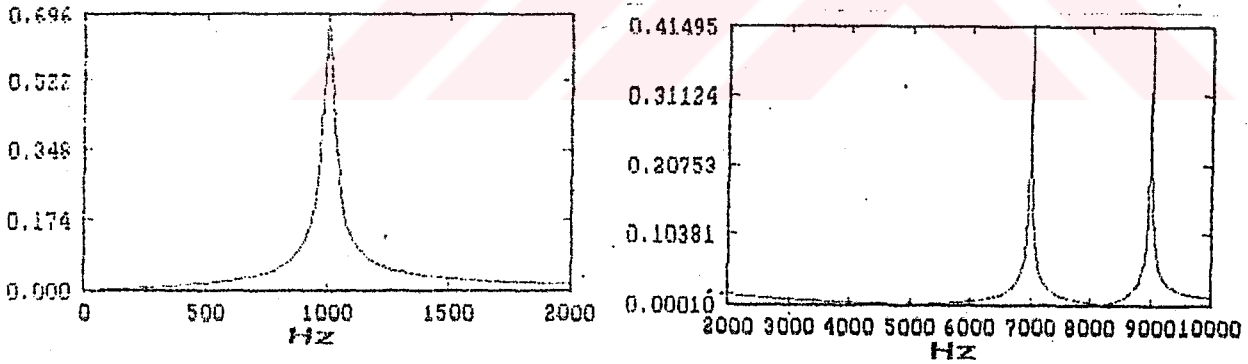
Benzer şekilde aşağıdaki karakteristiğin geçerli olduğu bölgede en az zayıflama 9 kHz'de ve 31.64 dB değerindedir. Bu değerler, bir bant geçiren filtrenin durdurma bölgesi için yeterli zayıflama değerleridir.

Böylece 11.harmoniğe kadar genlikleri bastırmak için bir periyot içinde 22 aralık yerine, 14 aralıkla da yeterli çözüme ulaşılabileceğini göstermiş olduk.



Şekil 4. 8 $N=3$, $NY=23$, $M=11$ için bulunan anahtarlama fonksiyonunu kullanarak elde edilen süzgeç karakteristiği

Anahtarlama fonksiyonunun daha az aralık içermesi anahtarlama kayıplarını azaltacak ve merkez frekansının yerleştirileceği frekansın ayarlanmasında bazı kolaylıklar sağlayacaktır.



Şekil 4. 9 $N=3$, $NY=9$, $M=5$ ve ikinci düzende bulunan anahtarlama fonksiyonunun kullanılmasıyla elde edilen süzgeç karakteristiği

Yedinci ve son örnek olarak, bu defa çeyrek peryot içinde gereğinden fazla aralık kullanarak, yani $N=3$, $M=5$, $NY=9$ ve ikinci aralık düzeni için GENKDI programını çalıştıralım. Birinci yolun çalıştığı aralıklar olarak kabul edilen anahtarlama fonksiyonu, $T=4495$ denemeden sonra 16,17,29,32,35,47,48,79,80,92,95,98,110,111 olarak bulunur.

Böylece elde edilen ve 4.5 kHz'e kadar geçerli olan karakteristik Şekil 4.9'da gösterilmiştir. Bu şekil ile Şekil 4.3 a karşılaştırılırsa gereksiz yere anahtarlama sayısını arttırmanın belirgin bir üstünlük sağlamadığı görülecektir.



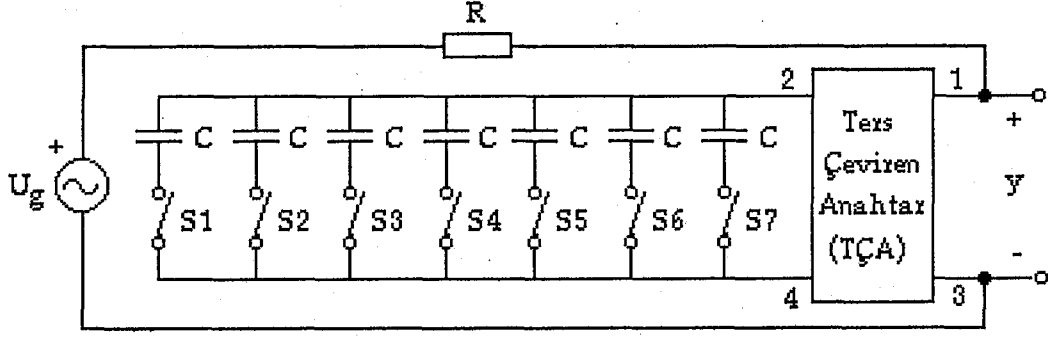
5. GERÇEKLEŐTİRİLEN DEVRE

5.1. Giriő

Bir MC6802 mikroişlemcili mikrobilgisayar kullanarak anahtarlanmış kapasitörlü 7 - yollu süzgeç devresinin anahtarlarını süren işaretler üretildi. Ancak bir mikrobilgisayarı sadece bu iş için çalıştırmak pratikte ekonomik olmayacağı için gerekli öneriler yapıldı. Burada bir amaçta pratik uygulama yaparak tasarımı yapılan filtrenin teorik sonuçlarının elde edilip edilemeyeceğini irdelemektir. Bu nedenle kolaylıkla temin edilebilen bir mikrobilgisayarla anahtarlanmış kapasitörlü N-yollu bir süzgecin nasıl gerçekleştirileceği ve bunun piyasa tipinin nasıl olabileceği konusunda fikirler öne sürüldü.

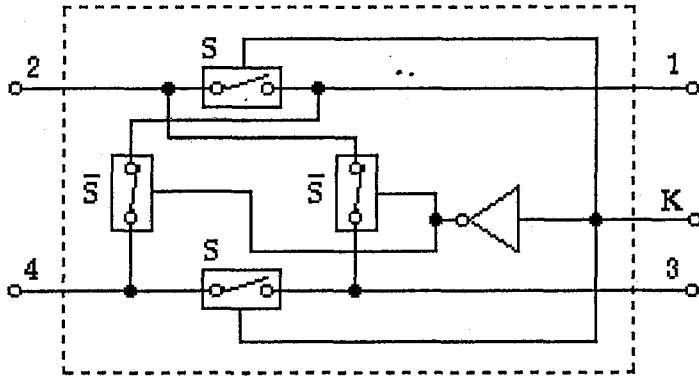
5.2. Önerilen Yöntem

Önerilen yöntemde, mikrobilgisayarın 8, 16 veya 32 bitlik olmasına bağılı olarak anahtarlanmış kapasitörlü N-yollu süzgecin yol sayısı sırasıyla, en çok 7, 15 veya 31 olabilir. Kullanılan mikrobilgisayarın 8 bitlik A ve B olarak adlandırılan iki çıkış portu vardır. Bu iki portta çıkış olarak tanımlansa bile ikisinede aynı anda bilgi yazdırmak mümkün olmayacağından, örneğin 15 -yollu AKNYS devresinin teoriyle uyumlu olarak çalıştırılması kullanılan mikrobilgisayarla mümkün değildir. Bu nedenle portlardan biri seçilerek Şekil 5.1 de verilen 7-yollu AKNYS devresi gerçekleştirildi.



Şekil 5.1 7-yollu anahtarlanmış kapasitörlü süzgeç devresi

Gerçekte bu mikrobilgisayarın PIA (Peripheral Interface Adapter)'sının uçlarına Latch(kilit) eklenerek bütün portlara bilginin gelmesi beklenebilir ve böylece Latch enable ile portlara aynı anda bilgi yazdırmak mümkün olur. Bu işlemler yapılırsa, ilerideki kısımlarda anlatılan mikrobilgisayar programının çalışması port sayısının artırılması ve Latch kullanımı nedeniyle bir miktar yavaşlarsada çıkış bilgisi istenilen sayıda birleştirilmiş portlara aynı anda yazılabilir. Kullanılan mikrobilgisayarda bu donanım olmadığından gerçekleştirilen N-yollu filtrenin yol sayısı 7 ile sınırlı tutulmuştur.



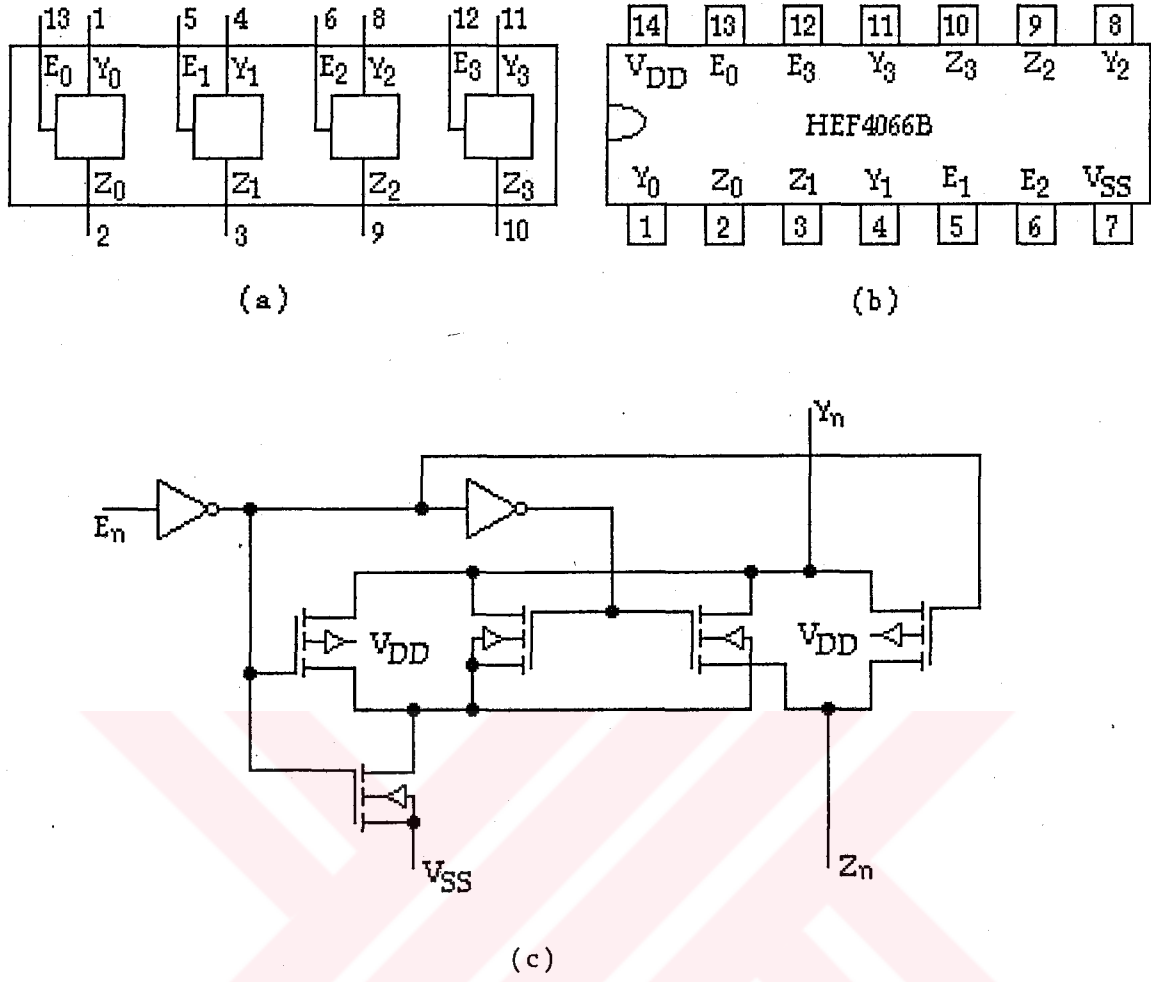
Şekil 5.2 K kontrol girişiyle birlikte ters çeviren anahtar (TÇA)

Mikrobilgisayarın A portundan alınan 8 çıkışın 7 tanesi yolların herbirisindeki anahtarları, kalan 8. si ters çeviren anahtarı (TÇA) kontrol etmek için kullanıldı. Şekil 5.2 de kontrol bilgisi girişi K ile birlikte TÇA'nın aldığı yapı verilmiştir. Bu yapıdaki K'nın ters çevrilme işlemi devrenin sadeliğini korumak amacıyla bir direnç ve bir anahtar kullanılarak gerçekleştirildi.

5 tane anahtarla oluşturulan TÇA'nın aldığı durumlar, K kontrol girişine bağlı olarak, Tablo 5.1 de verilmiştir. TÇA'nın kullanılmasıyla klasik yapıda 7-yollu AKNYS için gereken 7 direnç 1 e , 14 anahtar 12 ye indirilmiştir. Anahtar olarak herbiri 4 tane çift yönlü anahtar içeren CMOS HEF4066B entegresi kullanıldı. Şekil 5.3 de a-fonksiyonel, b- bacak, c- şematik diyagramları verilen entegrenin bazı önemli tipik karakteristik değerleri Tablo 5.2 de verilmiştir.

Tablo 5.1 TÇA'nın alabileceği durumlar

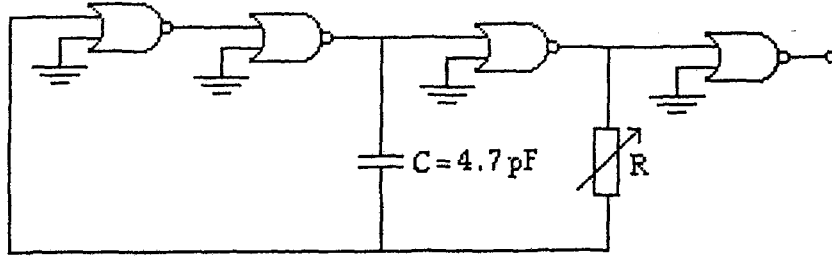
K	S	\bar{S}	Anahtarlanan Uçlar
1	Kapalı	Açık	2o—o1 4o—o3
0	Açık	Kapalı	4o—o1 2o—o3



Şekil 5.3. HEF 4066' nin a) fonksiyonel, b) bacak, c) şematik diyagramları

Tablo 5.2. HEF 4066B ' nin bazı önemli tipik karakteristikleri

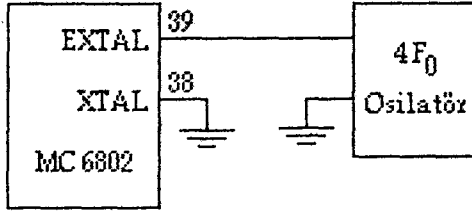
Herbir anahtarın harcayabileceği maksimum güç	$P_{\max} = 100 \text{ mW}$
Anahtarın kapalı olduğu durumdaki direnci $E_n = V_{DD}$, $V_{DD} = 5 \text{ V}$ iken	$R_{on} = 115 \text{ ohm}$
Anahtarın açık olduğu durumda kaçak akım $E_n = V_{SS}$, $V_{DD} = 5 \text{ V}$ iken	$I_{0Z} = \text{çok küçük}$
Anahtarın açık olduğu durumda kaçak akım $E_n = V_{SS}$, $V_{DD} = 15 \text{ V}$ iken	$I_{0Z} = 200 \text{ nA}$



Şekil 5.4 NOR kapılarıyla gerçekleştirilen dış osilatör devresi

Önerilen yöntem gereği, mikrobilgisayarın clock(saat) frekansının ayarlanabilir olması gerektiğinden 6802 mikroişlemcisinin sabit kristal osilatörü yerine bir dış, ayarlanabilir osilatör gerçekleştirildi. 6802 mikroişlemcisinin 1 MHz lik saat frekansı, 1 MHz lik kristalin 4 MHz likten daha pahalı olması nedeniyle, 4 MHz lik işareti 4'e bölen bir devrenin eklenmesiyle elde edilmiştir. Bu nedenle dışarıdan bağlanacak osilatörün frekansı 4 MHz olmalıdır. Bu frekans değerine CMOS kapılar kullanarak ulaşabiliriz. Şekil 5.4 de NOR kapıları kullanılarak gerçekleştirilen ayarlı osilatör devresi verilmiştir.

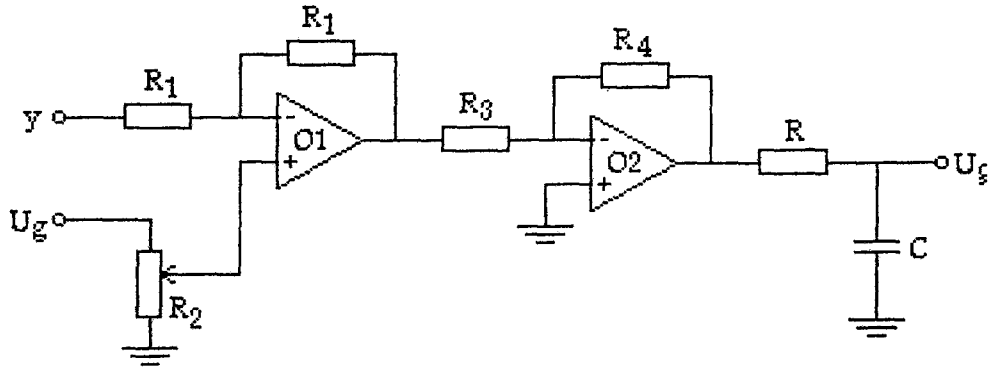
Bu osilatörle elde edilen saat darbeleri çok düzgün kare dalgalar olmasa bile 6802 mikroişlemcisindeki 4 e bölücü devre çıkışında işaret, mikrobilgisayarın kendi kristal osilatörünün sağladığı 1 MHz lik saat darbeleri kadar düzgündür. Dört tane iki girişli NOR kapısı içeren bir CMOS 4001 entegresi ile yapılan osilatörün MC 6802 mikroişlemcisine bağlantısı Şekil 5.5 de verilmiştir. Direnç değeri değiştirilerek mikroişlemcinin programı işletme hızı ve buna bağlı olarak band geçiren süzgecin merkez frekansı istenilen yere kaydırılabilir.



Şekil 5.5 Dış osilatörün 6802 mikroişlemcisine bağlantısı

Şekil 5.1 deki devrede R direncinin yeteri kadar büyük alınmaması durumunda anahtarların ideal olmaması nedeniyle band geçiren süzgecin y çıkışı, geçirilmeyen frekans değerlerinde tam sıfır olamaz. Bu istenmeyen küçük gerilim şekil 5.6 da gösterilen O1 işlemsel yükseltecinin pozitif girişine bağlı olan R2 potansiyometresi ayarlanarak bastırılabilir.

O2 işlemsel yükselteci ile de R4 ayarlı direnci değiştirilerek kazanç ayarı yapılabilir. Bu sayede çıkışa koyulacak olan birinci dereceden alçak geçiren süzgecin, çıkışı yükleme sorununun üstesinden kolayca gelinmiş olur.



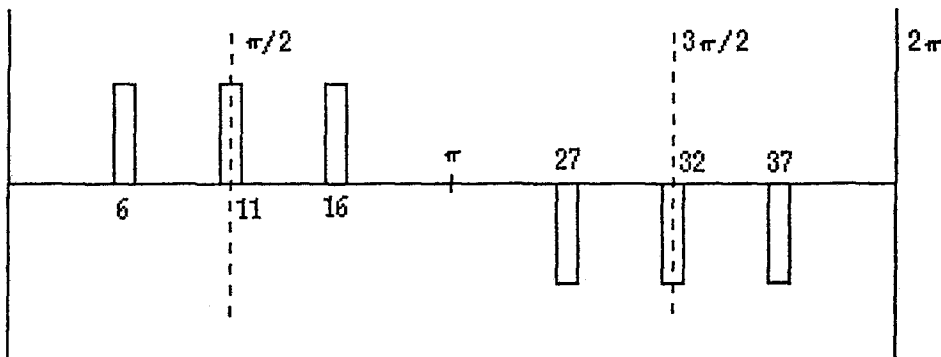
Şekil 5.6 Düzeltme ve kuvvetlendirme devresi

Önceden belirlenmiş bir çözüm için yazılan 7-yollu süzgeç programı çalıştırılmadan önce küçük bir program çalıştırılarak, süzgeç devresindeki bütün anahtarlar açık devre edilir. Bu durumda giriş frekansı tasarlanan merkez frekansı civarına getirilerek şekil 5.6 daki R_4 direnci ile kazanç ayarı (istenirse 1) yapılabilir.

5.3. Mikrobilgisayarda Çalıştırılacak Programın Elde Edilişi

Önceki kısımda, çeyrek periyottaki serbest puls sayısı 1, yol sayısı 7 için bulunan 3. harmoniğin bastırıldığı şekil 5.7 çözümünü ele alalım.

Bir periyot 42 aralığa bölünmüştür ve 1. yolun çalıştığı aralıklardan hareketle diğer 6 yolun çalıştığı aralıklar ($42/7=6$) 6 aralık kaydırılarak belirlenebilir. Tablo 5.3 de altı çizgili olarak gösterilen aralıklarda TÇA'nın K bilgisi 1, çizgili olmayanlarda 0 alınır. Veya terside geçerlidir. Tablo 5.3 den herbir aralıkta hangi yolun çalıştığı ve TÇA'nın durumu gözetilerek Tablo 5.4 düzenlenebilir.



Şekil 5.7 Birinci yolun çalıştığı aralıklar

Tablo 5.3 Her bir yoldaki anahtarların çalıştığı aralıklar

Yol	Çalıştığı Aralık					
1	<u>6</u>	<u>11</u>	<u>16</u>	27	32	37
2	1	<u>12</u>	<u>17</u>	<u>22</u>	33	38
3	2	7	<u>18</u>	<u>23</u>	<u>28</u>	39
4	3	8	13	<u>24</u>	<u>29</u>	<u>34</u>
5	9	14	19	<u>30</u>	<u>35</u>	<u>40</u>
6	<u>4</u>	15	20	25	<u>36</u>	<u>41</u>
7	<u>5</u>	<u>10</u>	21	26	31	<u>42</u>

Tablo 5.4'den mikrobilgisayarın 8 bitlik çıkış portunun alacağı değerler ve bu değerleri porta yüklemek için yazdırılması gereken ikili kodlanmış 16 tabanlı sayılar açıkça görülmektedir.

Tablo 5.5 A portunu çıkış olarak tanımlayan program

4F	CLRA	Kontrol kayıtlarını sıfırla
b7 E001	STAA \$E001	Veri yolu kayıtlarını seç
43	COMA	A portunu çıkış olarak seç
b7 E000	STAA \$E000	
86 04	LDA A £\$04	
b7 E001	STAA \$E001	

Önce A veya B portu çıkış olarak tanımlanmalıdır. Programın bu kısmı Tablo 5.5 de verilmiştir.

Tablo 5.4 de elde edilen sayıların girilmesi birkaç şekilde olabilir. Tablo 5.6 da verilen şekilde girilmesi halinde programın yazılması uzun ve bellekte kaplayacağı alan fazla olacaktır. Ancak bu durumda bir periyodun tamamlanması saat periyodunun $(42+7+3)=297$ katı süre alacak, diğerleriyle karşılaştırıldığında saat frekansının değiştirilmesiyle merkez frekansının değişim aralığı en geniş olanı elde edilecektir. Son satırdaki JUMP komutu nedeniyle aralık süreleri son aralığa kadar saat periyodunun 7 katı olmasına rağmen, son aralıkta 10 katı süre alacaktır. Bu sürelerin hesabında yardımcı olan ve bazı saçılmış komutların tükettiği süreleri gösteren bir Tablo, EK-5'de verilmiştir.

Tablo 5.6 Her bir aralıkta çıkış portunun alacağı değeri doğrudan yükleyen program

****	86 04	LDA A £\$02	A akümülatörüne (02) ₁₆ sayısını yükle
	b7 E000	STAA \$E000	A akümülatörünün değerini A portuna yükle
Bu kısım aralık sayısı kadar yazılacak			
7E ****		JMP	**** adresine git

Bu simetrisizlik, aralık sayısı yeteri kadar büyük olduğundan süzgecin çalışmasında ihmal edilebilecek bir bozulmaya yol açar. Yinede bu etki tamamen yok edilmek istenirse Tablo 5.7 de gösterildiği gibi uygun satır aralarına programın çalışmasını değiştirmeyen, ancak simetri sağlamak açısından JUMP komutu ile aynı zamanı tüketen bir komut (örneğin LDAB, opcode d6 00) girilebilir.

Bu durumda programın yazılımı daha çok yer tutar ve bir periyodun tamamlanması ($42 \cdot 10 = 420$) saat periyodunun 420 katı süre alır. Bu süreler anahtarlama frekansını, dolayısıyla band geçiren süzgecin merkez frekansını belirler. 1 MHz lik saat frekansında Tablo 5.6 daki program, merkez frekansı 3367 Hz, Tablo 5.7 deki program merkez frekansı 2380.9 Hz olan band geçiren süzgeçleri oluşturacaktır.

Yazılacak programın bellekte az yer tutması istenirse Tablo 5.5 den sonra Tablo 5.8 de verilen ana program girilebilir.

Tablo 5.7 Tablo 5.6' daki programın aralık sürelerinin eşitlenmiş olduğu program

****	86 04	LDAA E\$02	A akümülatörüne (02) ₁₆ sayısını yükle
	b7 E000	STAA \$E000	A akümülatörünün değerini A portuna yükle
	d6 00	LDAB E\$00	B akümülatörüne (00) 16 sayısını yükle
	Bu kısım aralık sayısı kadar yazılacak		
	86 C0	b7 E000	A akümülatörünün değerini A portuna yükle
	7E ****	JMP	**** adresine git

Tablo 5.8 Ana program

* CE (****) ¹	LDX #\$(****) ¹	(****) ¹ adresini başlangıç olarak index registere bildir
A6 00	LDAA \$00	A akümülatörüne index registerin gösterdiği adresteki değeri yükle
08	INX	Index registerin değerini 1 arttır
b7 E0 00	STAA \$E0 00	A akümülatörünün değerini A portuna yükle
8C (****) ²	CPX	Index registerin değerini (****) ² ₁₆ sayısıyla karşılaştır
26 F5	BNE	Sıfır değilse (10) ₁₀ geri git
7E *	JMP	* adresine git

Ana programda (****)¹ ve (****)² ile verilen adresler arasında Tablo 5.4 den bulunan ikili kodlanmış 16 tabanlı sayılar girilmelidir. Ana programda bu iki değişiklik yapılarak değişik çözümler çalıştırılabilir. Şekil 5.7 de verilen çözüm için Tablo 5.8 deki programın bir periyodu tamamlaması saat periyodunun $(42 \cdot (5+4+5+3+4)) = 42 \cdot 21 = 882 + 6 = 888$ 888 katı süre alır. Bu da 1 MHz lik saat frekansında süzgecin merkez frekansının 1126.1 Hz olması demektir. Herbir aralık saat periyodunun 21 katı süre alırken son aralık JMP ve LDX in getirdiği simetrisizlik nedeniyle 27 kat süre alır. Bu yazılım şeklinde simetrisizlik düzeltilemezse süzgecin çalışmasını çok etkilemez.

Mikrobilgisayarı çok düşük frekanslarda çalıştırmadan, merkez frekansı düşük süzgeç elde etmeye uygun ve bellekte az yer tutarak farklı programların çalıştırılmasına izin veren bir program Tablo 5.8 deki ana programa bir gecikme alt programı ekleyerek elde edilebilir. Eğer mikrobilgisayarda Timer (zamanlayıcı) donanımı varsa bu gecikme programının yapacağı işi register'e timer'in sayacağı ve bu süre boyunca programın geciktirileceği uygun bir sayı yazarak sağlanabilir.

Tablo 5.9. Ana programa gecikme alt programının eklenmesi

****	CE(****) ¹	LDX #\$(****) ¹	
	A6 00	LDAA \$00	
	08	INX	
	b7 E0 00	STAA \$E0 00	
	bd(****) ³	JSR	(****) ³ adresindeki alt programa git
	8C(****) ²	CFX	
	26 F2	BNE	Sıfır değilse (13) ₁₀ geri git
	7E ****	JMP	
gecikme alt programı			
(****) ³	86 (**)	LDAA \$(**)	A akümülatörüne (**) ₁₆ sayısını yükle
	4A	DECA	A akümülatörünün değerini 1 azalt
	26 Fd	BNE	Sıfır değilse (2) ₁₀ geri git
	39	RTS	Alt programdan dön

Bu mikrobilgisayarda Timer olmadığından yazılan ve Tablo 5.9 da verilen bu gecikme alt programındaki (**)=01 iken şekil 5.7 çözümünün bir periyodunun tamamlanması $(42*(5+4+5+9+2+2+4+5+3+4))=42*43=1806+6=1812$ saat periyodunun 1812 katı süre alacak ve 1 MHz lik saat frekansında merkez frekansı 551.8 Hz olacaktır.

Burada da herbir aralık saat periyodunun 43 katı süre alırken, son aralık 49 kat süre alır alır. Tablo 5.9 da verilen alt programda (**) sayısı 01, 02, 03,... verilerek, yazılım sayesinde süzgecin merkez frekansını kademeli olarak kaydırmak mümkündür. Aradaki frekans değerlerine yine dış osilatörün frekansı değiştirilerek ulaşılabilir.

Mikrobilgisayara Tablo 5.6, 5.7, 5.8 veya 5.9 şeklinde girilmek üzere Tablo 5.4 de belirlenen ikili kodlanmış 16 tabanlı sayıları elde etmek için MIKRO adlı bilgisayar programı hazırlandı ve listesi EK-6'de verildi. Programa giriş bilgisi olarak çeyrek periyottaki serbest puls sayısı, yol sayısı, çeyrek periyotta pulsların bulunduğu aralıklar (başka bir deyişle, 1.yolun çeyrek periyot içinde çalıştığı aralıklar) verilerek , yol sayısı 7 ve

daha küçük, 8-15 ve 15-31 için 8, 16, 32 bitlik mikrobilgisayarların portlarına yazdırılacak uygun sayıları verir.

5.4. Gerçekleştirme ile ilgili değerlendirmeler

N-yollu anahtarlanmış-kapasitörlü bant geçiren bir süzgeçin mikro-işlemci kontrollü olarak nasıl gerçekleştirileceği anlatılırken örnek olarak seçilen 7-yollu süzgeç test edildi. Değişik anahtarlama frekanslarında analiz sonuçlarının ölçülen sonuçlarla uyum içinde olduğu görüldü. Ayrıca ikinci aralık düzeninde ve çeyrek periyotta 2 serbest aç kullanarak 3. harmoniğin bastırıldığı çözüm olan ve anahtar fonksiyonunun 11,12,18,24,25,-46,-47,-53,-59,-60 aralıklarından oluştuğu bant geçiren filtre uygulamasında test edilerek teoriyle uyumlu sonuçlar ölçüldü. Yine 7-yollu olan bu filtrenin çalışması için mikrobilgisayara yazılması gereken program bilgilerini veren MİKRO programı çıktısı EK- 7'da verilmiştir.

Yukarıda anlatıldığı gibi dar bantlı yüksek kalite faktörlü bant geçiren filtre elde etmek için, merkez frekansı kolayca ayarlanabilir olsada bir mikrobilgisayarın bu işe tahsis edilmesi gerekli değildir. AKNYS devresi, mikroişlemci, çevre birimleri ve bellek ünitelerinin bir tümleşik devre içine yerleştirilmesi uygulamada büyük kolaylıklar sağlayacaktır. Buna örnek olarak genel mikrokontrolciler ve özel amaçlarla (örneğin telekomünikasyon) üretilmiş mikrokontrolcileri göstermek mümkündür. Bu elemanlarda mikroişlemci, bellek, seri ve paralel çevre birimleri ve timer 'lar tümleşik devre haline getirilerek bir minimum mikrobilgisayar oluşturulmuştur.

Ayrıca, böyle bir mikrobilgisayara birden fazla AKNYS devresi koyularak bunların paralel olarak çalıştırılması da mümkün olabilir. Bütün bunlar süzgeç çalışma frekansından çok yüksek frekanslarda çalışan mikroişlemci gerektirecektir. Mikrobilgisayarın çok mikroişlemcili olması ve her bir mikroişlemcinin paralel çalışacak AKNYS'leri ayrı ayrı kontrol etmesi de şimdi çok pahalıya mal olmasına rağmen ileride ucuz gerçekleştirilebilecek bir tümdevre olabilir.



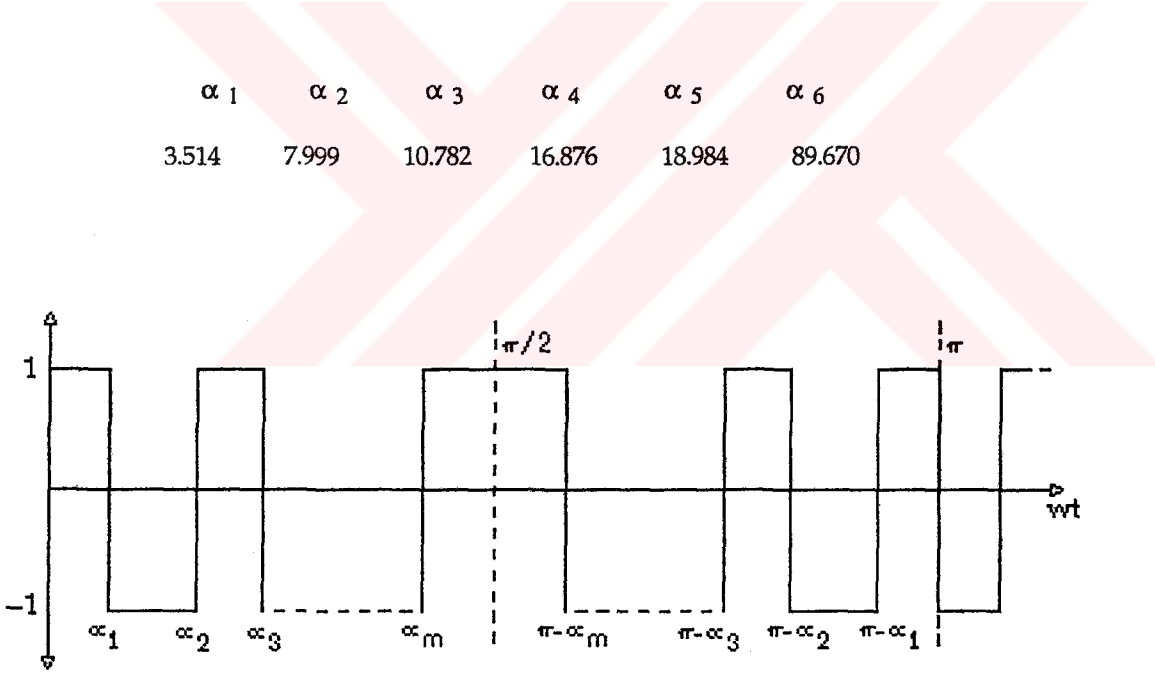
6. DARBE GENİŞLİK MODÜLASYONLU İNVERTERLE ASENKRON MOTORUN KONTROLUNDA ARA DEVRENİN ETKİSİNİN İNCELENMESİ

6.1. Giriş

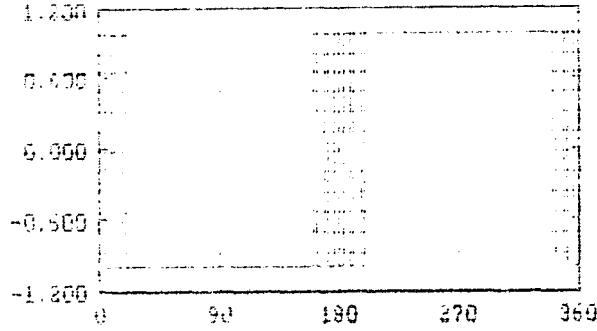
Asenkron motorun kontrolü amacıyla yapılan bir çalışmada (Tatar, 1994) Darbe Genişlik Modülasyonlu (DGM), inverter (evirici) çıkış geriliminin istenmeyen harmoniklerinin elendiği ve ana harmonik bileşenin en büyük yapıldığı anahtarlama açıları bulunmuştur. Bu çalışmada harmonik eliminasyonu yöntemleri tartışılmış, toplam harmonik distorsiyonu gözönüne alınarak yapılan optimizasyon yöntemiyle bulunan çözümler teorik ve uygulama sonuçlarıyla verilmiştir. Bulunan en iyi çözümün kontrol edilen motora uygulanması ile motordan alınan performansın beklendiği gibi sürücünün harmonik bileşenlerine bağlı olduğu gözlenmiştir. Adı geçen çalışma bir yönüyle benzerlik gösterdiğinden ve devrenin tek fazlı durumu, izleyen kısımlarda anlatıldığı gibi ele alındığında bu tezdeki diğer devreler gibi MAINLN'le kolaylıkla analiz edilebildiğinden ara devrenin etkisini irdelemek amacıyla bu tezin kapsamına alınmıştır.

6.2. Bulunmuş Olan Bir Çözüm ve Analizi

Yarım ve çeyrek dalga simetrisine sahip, tek fazlı evirici kutup gerilimi genel olarak Şekil 6.1'deki gibidir. Bu genel şekilde görülen α_1, α_3 gibi tek sayı indisli anahtarlama açıları negatife geçiş, α_2, α_4 gibi çift sayı indisli anahtarlama açıları pozitive geçiş açıları olarak tanımlanır. Bu işaretin de önceden belirtildiği gibi, simetriden dolayı Fourier açılımında sadece tek sayılı sinüslü bileşenler vardır. Bu çalışmada bulunan çözümlerden birini ele alalım. Çeyrek periyot içinde 6 açı yeri belirlenerek 5., 7., 11., 13. harmonik bileşenlerinin bastırıldığı ve ana harmonik bileşenin en büyük olduğu çözüm aşağıda verilmiştir.



Şekil 6.1 Evirici kutup geriliminin genel şekli

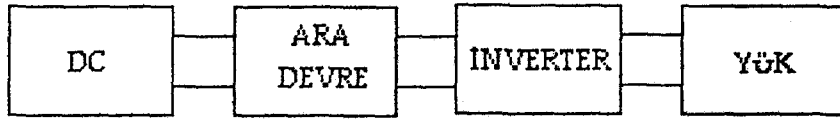


Şekil 6.2 Ele alınan çözüm için evirici kutup geriliminin şekli

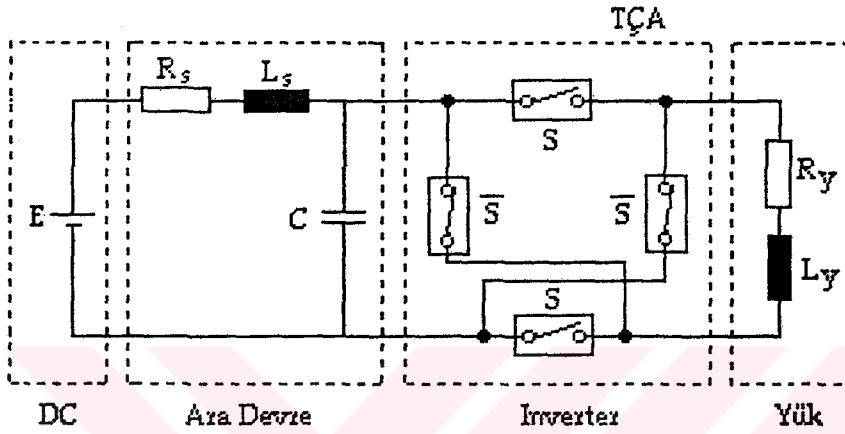
Burada α 'lar derece olarak verilmiştir. Adı geçen çalışmada bu çözümlerin bulunduğu yöntemler ve karşılaştırmaları yapılmıştır. Burada amaç bu ve bunlar gibi çözümlerde gerilim veya akım ara devresinin etkisini ortaya koymaktır. Motorun davranışında esas etkili olan motordan geçen akımdır. Bununla birlikte MAINLN kullanılarak hem yük üzerindeki gerilimin hemde yükten geçen akımın verilen frekans için, sürekli haldeki değişimi bir periyot boyunca çizdirilebilir. Bu periyodik akım veya gerilimin DISCRFY adı verilen bir Fortran bilgisayar diliyle hazırlanmış bir programla ayrık (discrete) Fourier analizi yapılır. Elde edilen harmonik genliklerinden, yüzde Toplam Harmonik Distorsiyonu (THD) aşağıdaki gibi hesaplanır.

$$\% \text{ THD} = 100 \frac{\sqrt{\sum_{i=2}^M H_i^2}}{H_1} \quad (6.1)$$

Burada, H_i 'ler akım veya gerilimin harmonik bileşenlerinin genlikleri, M ise hesaba katılan son harmoniğin numarasıdır.



(a)



(b)

Şekil 6.3 a) Sistemin tek fazlı blok gösterimi

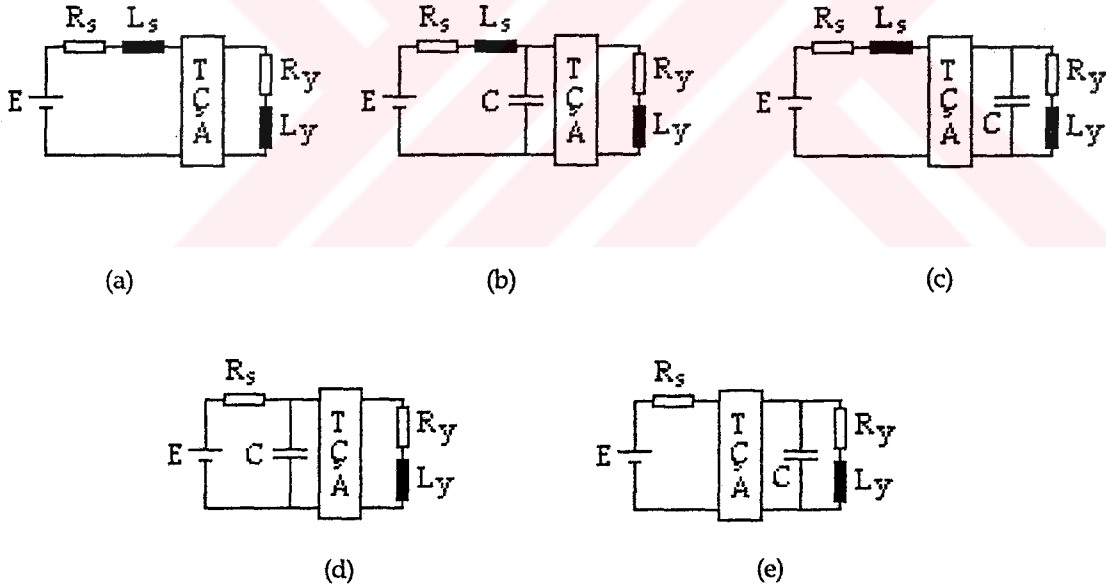
b) Karma ara devre ile sistemin açık gösterimi.

Ele alınan çözüm 3 fazlı, 3 telli sistem için bulunmuş ve böyle sistemlerde 3 ve 3'ün katları olan harmonik akımlarının akmadığı belirtilmiştir. Bu nedenle yüzde THD'nin hesabında i 'nin üç veya üçün katı olduğu harmonikler kullanılmamıştır.

Eğer yük olarak motor sargısı bir direnç ve bir indüktansla, inverter TÇA ile ve ara devre birer direnç, indüktans ve kondansatörle modellenirse, Şekil 6.3.a'da blok olarak gösterilen devre, Şekil 6.3.b'deki gibi ele alınabilir. Burada, E ile gösterdiğimiz doğru gerilim kaynağı aslında tek fazlı doğrultucu çıkışı olup, dalgalılığı olan bir doğru gerilimdir. Ancak burada araştırılan ara devrenin varlığı ve yapısının yük gerilimi ve

akımına etkisi olduğundan çok iyi filtre edilmiş bir doğru gerilim olarak kabul edilmiştir. Ayrıca, motor olarak düşünülen yükün motorda endüklenen emk'yı gözönüne almadan doğrusal elemanlar olan seri bir direç ve indüktansla modellenmesi yetersiz olmakla birlikte, devre bu haliyle kolaylıkla analiz edilebileceğinden tercih edilmiştir. Motorun sürekli ve geçici durum davranışının incelenmesi için iyi bir benzetim, hava aralığının üniform olduğu, magnetik devrenin doğrusal davrandığı, statör ve rotor sargı dirençlerinde sıcaklık ve frekansla değişim olmadığı gibi kabüllerle (Tatar, 1994)'de verilmiştir.

Şekil 6.3.b'de verilen modelde, Şekil 6.2 'de verilen gerilimin pozitif olduğu sürelerde S anahtarları kapalı, \bar{S} anahtarları açık (1.konum), gerilim negatif olduğu sürelerde S anahtarları açık, \bar{S} anahtarları kapalıdır (2.konum). Böylece Şekil 6.2 'den de açıkça görülebileceği gibi bir periyot boyunca devrenin girdiği $K=26$ durum için durum ve çıkış denklemleri yazılabilir.



Şekil 6.4 a) Akım ara devreli, b) karma ara devreli, c) akım ara devreli ve yük tarafında kondansatör, d) gerilim ara devreli, e) gerilim ara devreli ve yük tarafında kondansatör

Tablo 6.1 Şekil 6.4'deki devrelerin TÇA konumuna göre durum ve çıkış denklemlerindeki katsayı matrisleri

Devre	TÇA konum	A	B	Çıkış V_y		Çıkış I_y	
				C	D	C	D
Şekil 6.4 (a)	1	$-\frac{R_s+R_y}{L_s+L_y}$	$\frac{1}{L_s+L_y}$	$\frac{L_sR_y-L_yR_s}{L_y+L_s}$	$\frac{L_y}{L_y+L_s}$	1	0
	2	"	*-1	"	*-1	"	"
(b)	1	$0 \quad \frac{1}{C} \quad \frac{-1}{C}$ $\frac{-1}{L_s} \quad \frac{-R_s}{L_s} \quad 0$ $\frac{1}{L_y} \quad 0 \quad \frac{-R_y}{L_y}$	0 $\frac{1}{L_s}$ 0	1 0 0	0	0 0 1	0
	2	" " *-1 " " " *-1 " "	" " "	*-1 " "	"	" " " "	"
(c)	1	$0 \quad \frac{1}{C} \quad \frac{-1}{C}$ $\frac{-1}{L_s} \quad \frac{-R_s}{L_s} \quad 0$ $\frac{1}{L_y} \quad 0 \quad \frac{-R_y}{L_y}$	0 $\frac{1}{L_s}$ 0	1 0 0	0	0 0 1	0
	2	" *-1 " *-1 " " " " "	" " "	" " "	"	" " " "	"
(d)	1	$\frac{-1}{R_sC} \quad \frac{-1}{C}$ $\frac{1}{L_y} \quad \frac{-R_y}{L_y}$	$\frac{1}{R_sC}$ 0	1 0	0	0 1	0
	2	" *-1 " *-1 " "	" "	*-1 "	"	" " "	"
(e)	1	$\frac{-1}{R_sC} \quad \frac{-1}{C}$ $\frac{1}{L_y} \quad \frac{-R_y}{L_y}$	$\frac{1}{R_sC}$ 0	1 0	0	0 1	0
	2	" " " " " "	*-1 "	" "	"	" " "	"

Ele alınan devre biçimleri Şekil 6.4'de, analizleri için MAINLN'e girilmesi gereken durum ve çıkış denklemlerindeki katsayı matrisleri Tablo 6.1'de topluca verilmiştir.

Tablo 6.1'de, her bir devrenin durum ve çıkış denklemlerindeki katsayı matrislerinin elemanları için TÇA'nın 2. konumunda kullanılan (·) işareti 1. konumdaki değerini koruyan matris elemanlarını, (* -1) ise işaret değiştiren elemanları göstermek amacıyla kullanılmıştır.

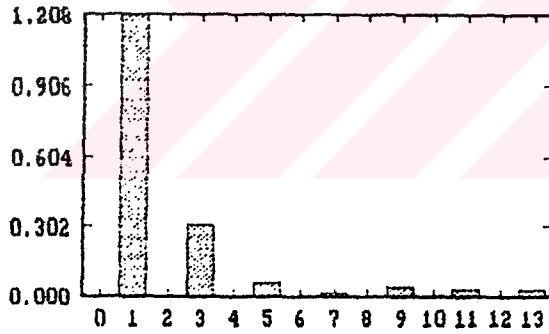
Şekil 6.4.a' da verilen devrede durum değişkeni olan indüktans akımında süreksizlik söz konusudur. Bu durumda programa anahtarlama anında geçerli olan anahtarlama denklemindeki katsayı matrisleri de girilmelidir. TÇA' nın konum değiştirme anlarında geçerli olan bu matrisler,

$$F = (L_y - L_s) / (L_y + L_s) , \quad G = 0 \quad (6.2)$$

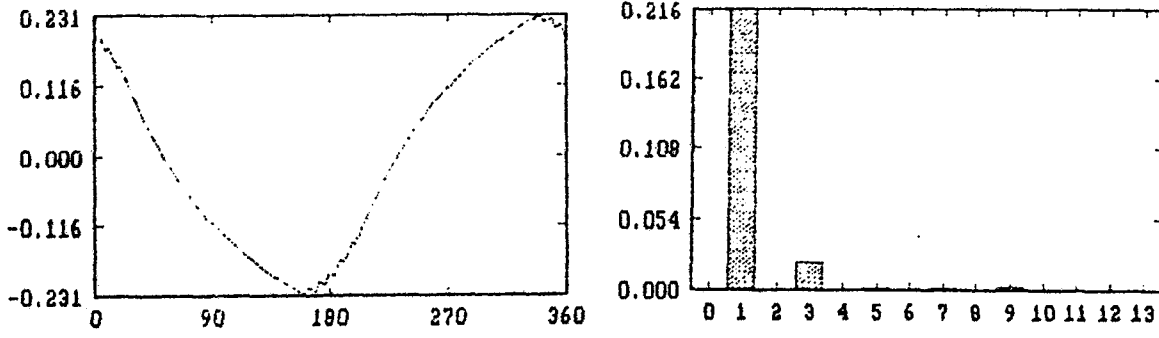
şekindedir. Tablo 6.1'de verilen diğer devreler için süreksizlik söz konusu değildir. MAINLN adlı bilgisayar programıyla bu devrelerin analizi için, kullanıcı tarafından bazı alt programların düzenlenmesi gerekir. Tekrardan kaçınmak amacıyla sadece Şekil 6.4.a devresi için bu altprogramlar, değişken isimlerinden vektöryel olanlarının minimum boyutlarıyla, EK-8'de verilmiştir. Bu alt programlar Şekil 6.4.a'daki devrenin, Tablo 6.1 ve Denklem 6.2'de verilenler aracılığıyla çeyrek periyot içinde değişik anahtarlama açıları ve farklı frekanslar için sürekli hal analizini mümkün kılar. Bu programın bir periyot içinde 500 nokta için yük geriliminin veya akımının değeri olan çıktısını giriş olarak alan discrete Fourier programının (D.FOR) listesi ise EK-9'de verilmiştir. Bu programın verileri de daha önce olduğu gibi ASEASY adlı programla grafik olarak değerlendirilmiştir.

6.3. Akım Ara Devresinin İncelenmesi

Ara devrenin akım ara devresi olması durumunu gösteren Şekil 6.4.a devresinde ara devre elemanları olan R_s ve L_s 'in sıfır değerleri için, Şekil 6.2'de verilen kutup gerilimi yük üzerinde görülür. Bu gerilimin harmonik spektrumu Şekil 6.5'de verilmiştir. Şekil 6.5 için Denklem (6.1)'den hesaplanan %THD = 5.542310 dur. (Kare dalga için bu değer 27.3206 dır.) Bu gerilimin $R_y = 3 \Omega$, $L_y = 0.015$ H'lik yükten 50 Hz çalışma frekansında akıttığı akımın şekli ve harmonikleri Şekil 6.6.a ve b'de verilmiştir. Bu harmonik genlikleri için %THD=0.7323166 olup inverterin çalışma frekansına yani, bu frekansta yük empedansı değerine bağlı olarak değişir. (Aynı frekans ve yük empedansı değerinde kare dalga için bu değer 5.422634'dür.)

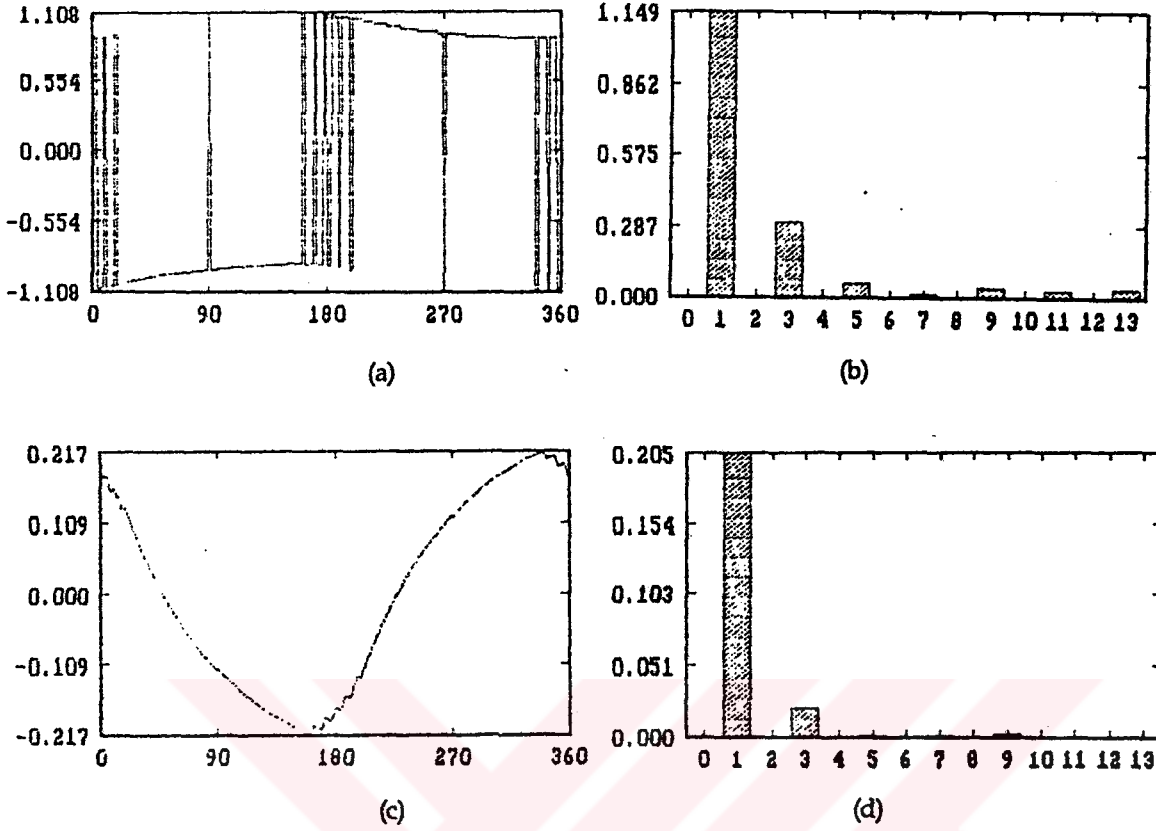


Şekil 6.5 Şekil 6.2'de gösterilen gerilimin harmonik spektrumu



Şekil 6.6 a) Şekil 6.2'deki gerilimin $R_y=3 \Omega$, $L_y=0.015$ H'lik yükten 50 Hz'de akıttığı akımın şekli, b)harmonik spekturumu

Görüldüğü gibi istenmeyen harmonikleri bastırılmış olan gerilimi omik-endüktif bir yüke uygulamak, yük empedansının yüksek frekanslı akım bileşenlerine gösterdiği büyük direnç nedeniyle akımın THD'sinde kendiliğinden bir avantaj sağlar. Bu avantaj çalışma frekansının 50 Hz'e yakın değerlerinde daha belirgindir. Yani R_s ve L_s 'in sıfır değeri için anahtarlama fonksiyonu olan kutup gerilimi yük üzerinde görülür ve düşük frekanslarda akımın şekli gerilimin şekline yaklaşacağından, %THD'si geriliminkine (bu çözüm için 5.542310 değerine) yaklaşır. Daha yüksek frekanslarda akımın harmonik distorsiyonu daha küçük değerler alır. Ara devrede $L_s=0$ iken R_s in sıfırdan farklı değerleri için yük üzerindeki gerilimin veya yükten geçen akımın harmonikleri, $R_s=0$, $L_s=0$ durumu için bilinen gerilim harmoniklerinden, devrenin bir gerilim bölücü olarak ele alınmasıyla hesaplanabilir. EK-9'da verilen D isimli programın içinde GERBOL alt programı bu amaçla yazılmıştır. Şekil 6.7.a, b, c, d ' de $R_s=0.5 \Omega$, $L_s=0$ H, $R_y=3 \Omega$, $L_y=0.015$ H için 50 Hz'de programların çalıştırılmasıyla elde edilen gerilim ve akımın şekilleri verilmiştir. Gerilim şeklindeki bozulma nedeniyle gerilim için %THD, 5.818500'a akım için 0.7675999'a yükselmiştir.

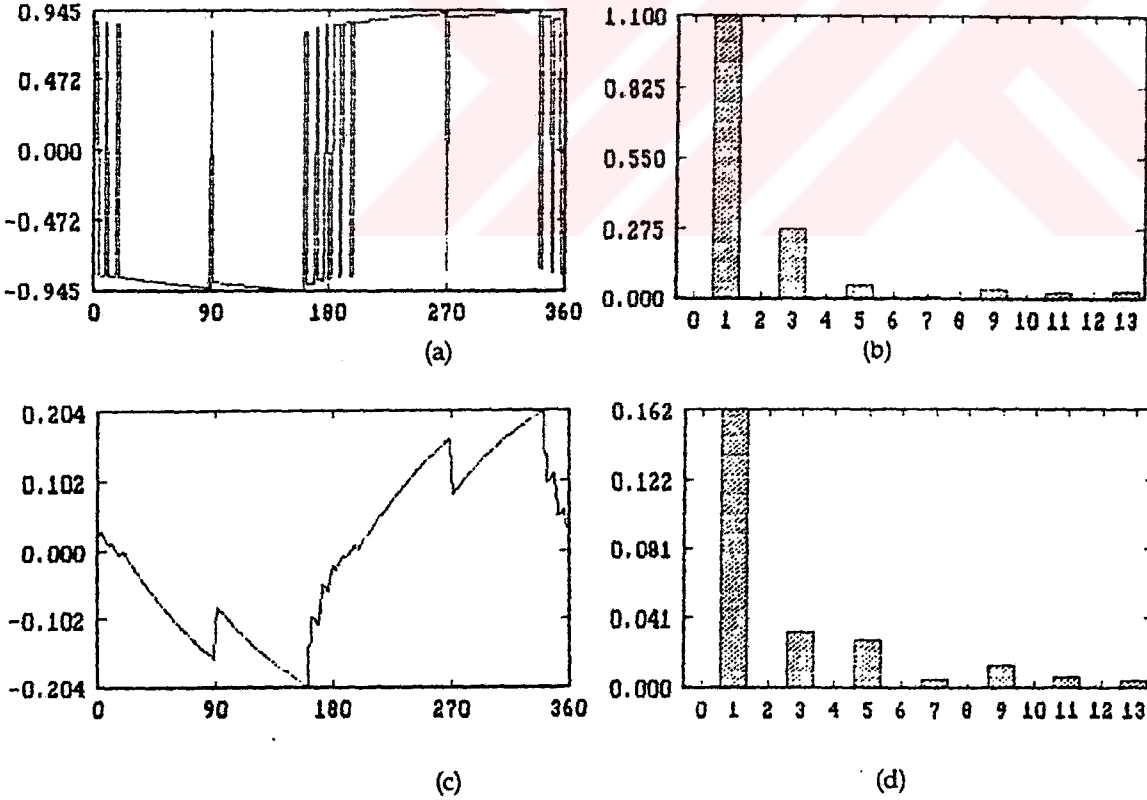


Şekil 6.7 a) $R_s=0.5 \Omega$, $L_s=0$ H, $R_y=3 \Omega$, $L_y=0.015$ H için yük gerilimi, b) gerilimin harmonik spektrumu, c) yük akımı, d) akımın harmonik spektrumu

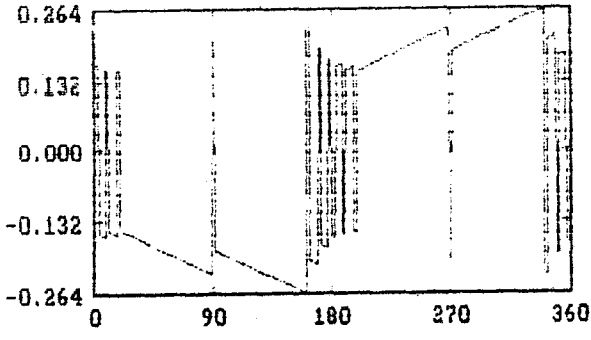
Aynı devrede bu defa $R_s=0$, $L_s=0.0025$ H için benzer işlemler yapılarak bulunan yük gerilimi, harmonik genlikleri, yük akımı ve harmonik genlikleri Şekil 6.8.a,b,c,d'de verilmiştir. Şekil 6.8.a'daki gerilim için %THD=5.278176 iken, c'deki akım için %THD=17.6016'ya çıkmıştır. Akım ara devresine benzetilen bu durum için akımda ortaya çıkan süreksizlik Şekil 6.8.c'de görüldüğü gibi bastırılmak istenen harmoniklerin tekrar ortaya çıkmasına neden olur. Ayrıca, bu durumda yukarıda GERBOL alt programıyla yapılan hesaplamalar geçersizdir. Yani devreyi gerilim bölücü olarak düşünüp, R_s ve L_s 'in sıfır değerleri için önceden hesaplanmış olan gerilim veya akım harmoniklerinden, L_s 'in sıfırdan farklı değeri için yük gerilimi veya akımının harmonikleri hesaplanamaz.

Benzer şekilde yük üzerindeki gerilimin harmoniklerini, yükün o harmonik frekansındaki empedans değerine bölerek yük akımının harmonikleride önceki gibi hesaplanamaz. Yük üzerindeki gerilimin harmonikleriyle yükten geçen akımın harmonikleri arasında böyle bir ilişkinin kurulabilmesi ancak akımdaki belirli süreksizlikler için hesaplanan Fourier katsayılarının gerilim için hesaplanan katsayılarla cebirsel olarak toplanması durumunda mümkündür.

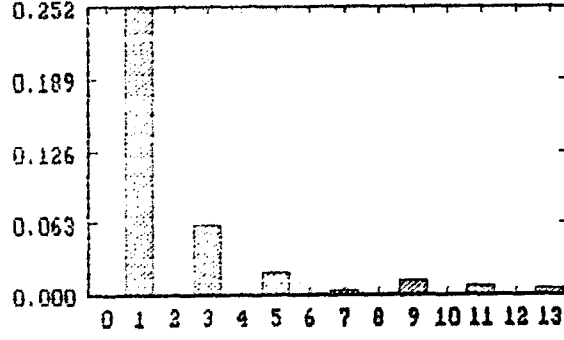
Akım ara devresinde kullanılan $L_S=0.0025$ H'lik indüktans yerine daha gerçekçi bir değer olan $L_S=0.1$ H'lik bir indüktans koymamız durumunda elde edilen şekiller Şekil 6.9.a, b, c, d'de verilmiştir. Şekil 6.9.c'de gösterilen yük akımının distorsiyonu 20.8767 olup çok olumsuzdur. Sonuç olarak burada ele aldığımız modelde akım ara devresinin kullanılmasının akımın harmonik distorsiyonunu arttırdığını görüyoruz.



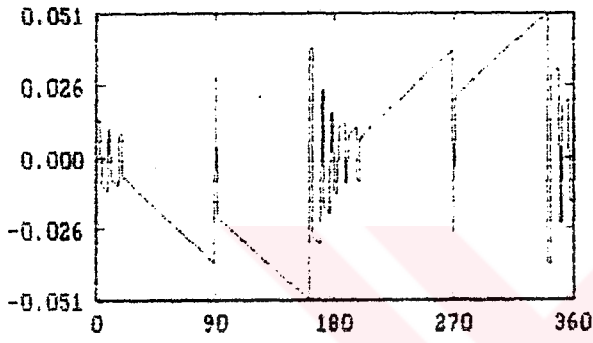
Şekil 6.8 a) $R_S=0$, $L_S=0.0025$ H için aynı yükün üzerindeki gerilim, b) gerilimin harmonik spektrumunu, c) yük akımı, d) yük akımının harmonik spektrumunu



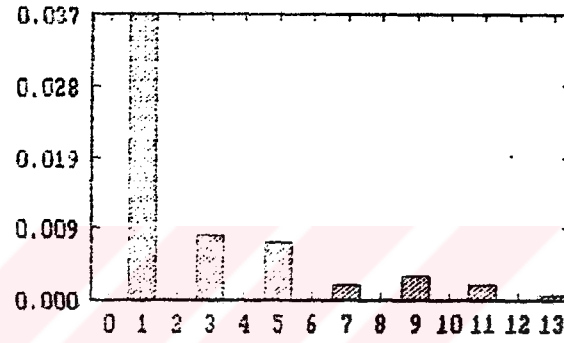
(a)



(b)



(c)

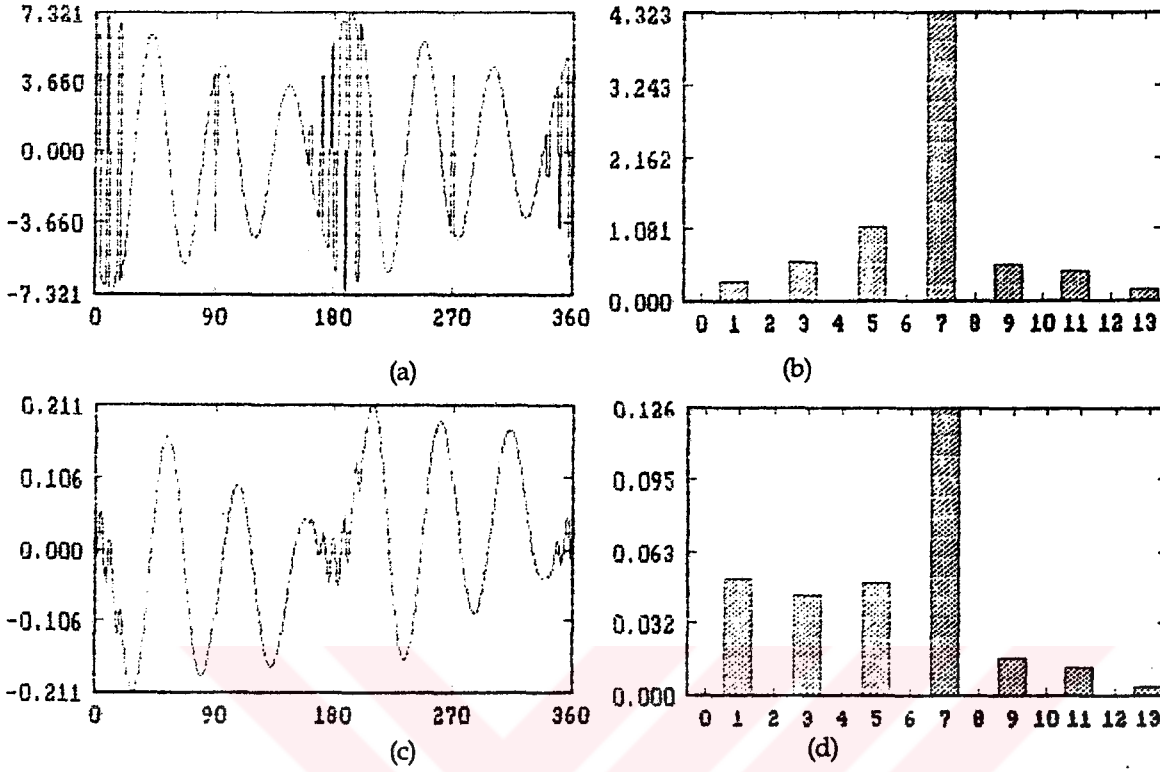


(d)

Şekil 6.9 a) $R_S=0$, $L_S=0,1$ H için aynı yükün üzerindeki gerilim, b) gerilimin harmonik spektrumunu, c) yük akımı, d) yük akımının harmonik spektrumunu

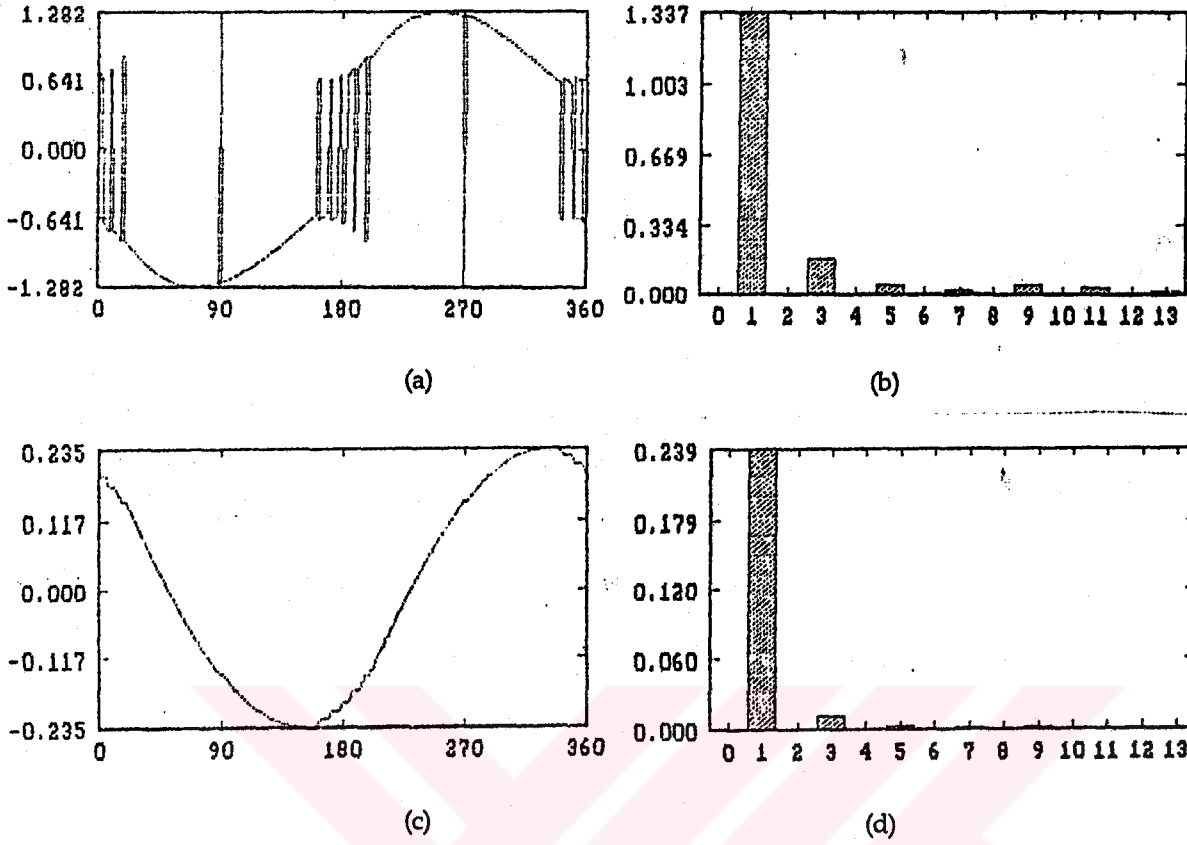
6.4. Karma Ara Devrenin İncelenmesi

Şekil 6.4.b'de gösterilen devre, Tablo 6.1'deki denklemleri kullanarak yukarıdakilere benzer şekilde $R_S=0,1\Omega$, $L_S=0,1$ H, $R_Y=3\Omega$, $L_Y=0,015$ H $C=15,852$ μ F değerleri için analiz edilerek elde edilen sonuçlar Şekil 6.10.a, b, c, d'de verilmiştir.



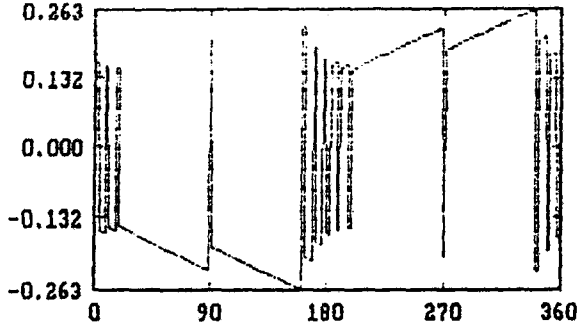
Şekil 6.10 a) $R_s=0.1$, $L_s=0.1$ H $C=15.852$ μ F için yük üzerindeki gerilim, b) gerilimin harmonik spektrumu, c) yük akımı, d) yük akımının harmonik spektrumu

Diğer elemanlar aynı değerde iken kondansatörün 776.795 μ F değeri için benzer çizimler Şekil 6.11 a, b, c, d'de gösterilmiştir. Şekillerin incelenmesinden de anlaşılacağı gibi gerilim ve akımda çizdirilmiş oldukları frekansın, $C=776.795$ μ F olduğu devrede 1, $C=15.852$ μ F olduğu devrede ise 7 katı frekansta bir salınım söz konusudur. Gerçektende paralel L_s , L_y ve C 'nin oluşturduğu devrelerin $C=776.95$ μ F ve $C=15.852$ μ F için rezonans frekanları, sırasıyla 50 ve 350 Hz'dir. Bu osilasyon nedeniyle kondansatörün 776.795 μ F olduğu devrede çok büyük olumsuzluk olmadığı halde (akım için THD=0.8639), diğer devrede gerilim ve akımın 7. harmonikleri büyüyerek, (6.1)'den hesaplanan distorsiyonları gerilim için 1540.50, akım için 270.428 gibi çok olumsuz değerlere yükseltmiştir.

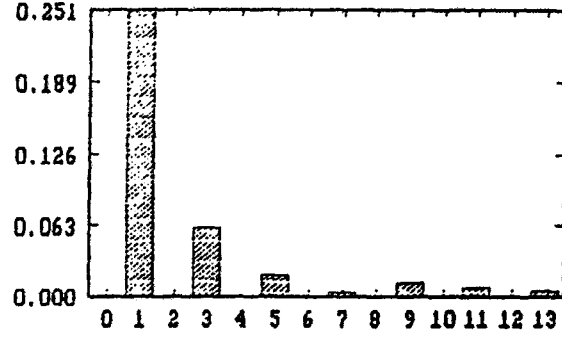


Şekil 6.11 a) $R_S=0.1$, $L_S=0,1$ H $C= 776.795 \mu F$ için yük üzerindeki gerilim, b) gerilimin harmonik spektrumunu, c) yük akımı, d) yük akımının harmonik spektrumunu

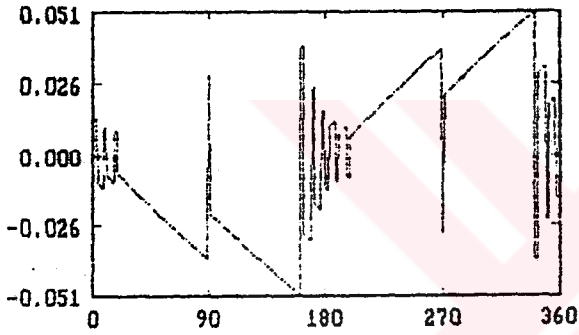
Şekil 6.4 a'da verilen devrenin aynı değerler için elde edilen sonuçları Şekil 6.12 a, b, c, d'de verilmiştir. Kondansatörün ilavesiyle süreksizlik ortadan kalktığı için Şekil 6.12 deki gerilim ve akım THD'leri sırasıyla 8.3801 ve 20.8623 'den Şekil 6.11 deki değerler olan 3.9865 ve 0.8639'a düşmüştür. Ancak rezonans frekansının 7.harmoniğe yerleşmiş olduğu Şekil 6.10' da görüldüğü gibi gerilim ve akım için yine sırasıyla THD'ler 1540.50 ve 270.428 olan çok olumsuz değerlere yükselmiştir.



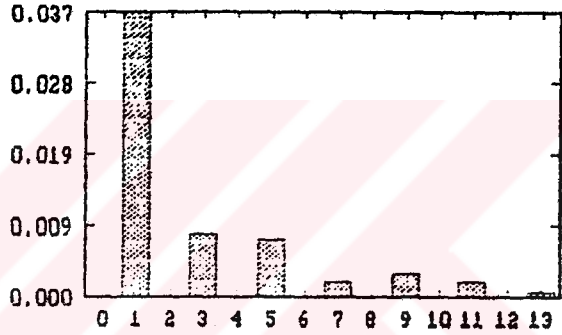
(a)



(b)



(c)



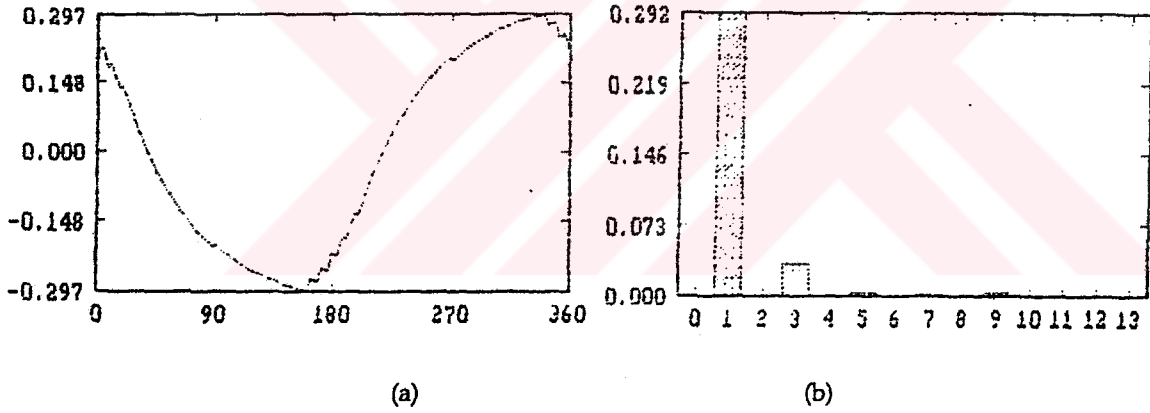
(d)

Şekil 6.12 a) $R_S=0.1$, $L_S=0,1$ H için aynı yükün üzerindeki gerilim, b) gerilimin harmonik spektrumunu, c) yük akımı (süresiz), d) yük akımının harmonik spektrumunu

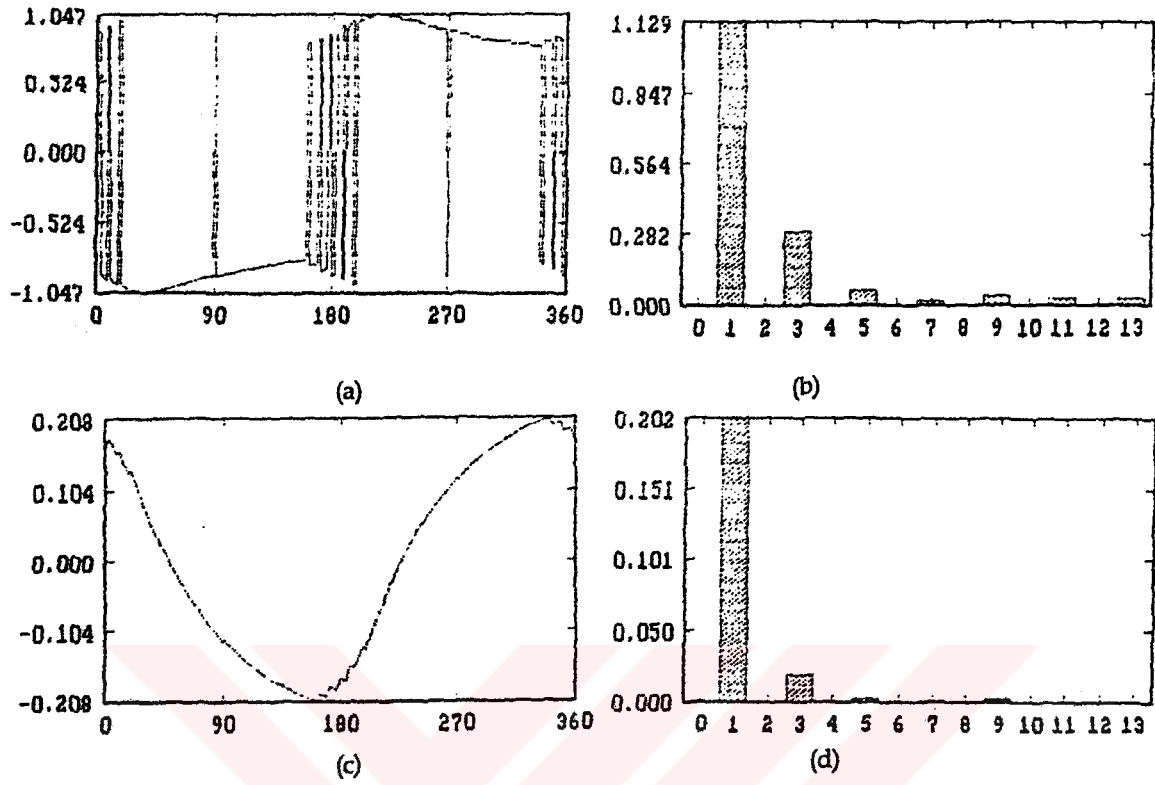
6.5. Gerilim Ara Devresinin İncelenmesi

Şekil 6.4 d'de gösterilen devre, doğrultucu çıkışı dalgalılığı olmayan bir DC olarak kabul edilmesine rağmen, doğrultucu ile kondansatör arasındaki direncin, genel olarak gerilim ara devresinin etkisini inceleme imkanı sağlar. R_S 'in sıfır değeri için bu devre ile

Şekil 6.4 a'da gösterilen devrede R_S ve L_S 'in sıfır olduğu durumun bir farkı yoktur. Bu durum çok iyi bir gerilim ara devresinin varlığına karşı düşer ve düşük gerilim ve akım harmonik distorsiyonlarını sağlar. Gerçektende akım distorsiyonu bakımından en elverişli ara devre gerilim ara devresi olup, yükün omik-endüktif yapısı nedeniyle ilave bir avantaja sahiptir. Çalışma frekansının 30 Hz olması durumunda aynı yük için yük akımının şekli ve harmonik spekturumu Şekil 6.13 a ve b'de verilmiştir. Bu durum içinde yük üzerindeki gerilim Şekil 6.2'de 50 Hz çalışma frekansı için çizilen gerilim gibidir. Bu frekans için akımın harmonik distorsiyonu 50 Hz'de 0.7323166 olan değerinden 0.889772 değerine yükselmiştir. Şekil 6.4 d'de gösterilen devre, $R_S=1 \Omega$, $R_Y=3 \Omega$, $L_Y=0.015 \text{ H}$ ve $C=1000 \mu \text{ F}$ değerleri için analiz edilerek elde edilen sonuçlar Şekil 6.14'de topluca verilmiştir. Burada gerilim ve akım için %THD'ları sırasıyla 5.9433 ve 0.88730 iken $R_S=0.1 \Omega$ için 5.6703 ve 0.75028 değerine düşer.



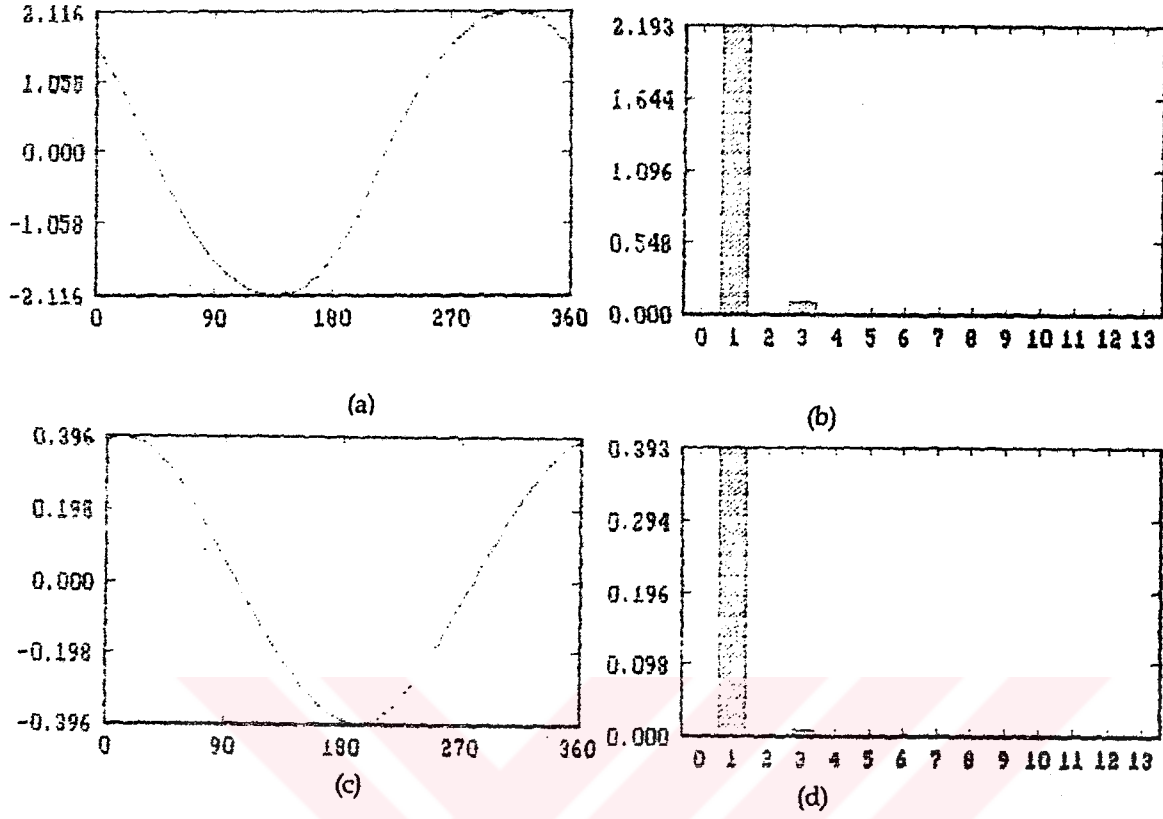
Şekil 6.13 a) $R_S=0$, $L_S=0$ için 30 Hz'de aynı yükten geçen akımın şekli, b) akımın harmonik spekturumu



Şekil 6.14 a) $R_S=1 \Omega$, $R_Y=3 \Omega$, $L_Y=0.015 \text{ H}$ ve $C=1000 \mu \text{ F}$ için yük üzerindeki gerilim, b) gerilimin harmonik spektrumunu, c) yük akımı, d) yük akımının harmonik spektrumunu

6.6. Yüke Paralel Kondansatörün Etkisi

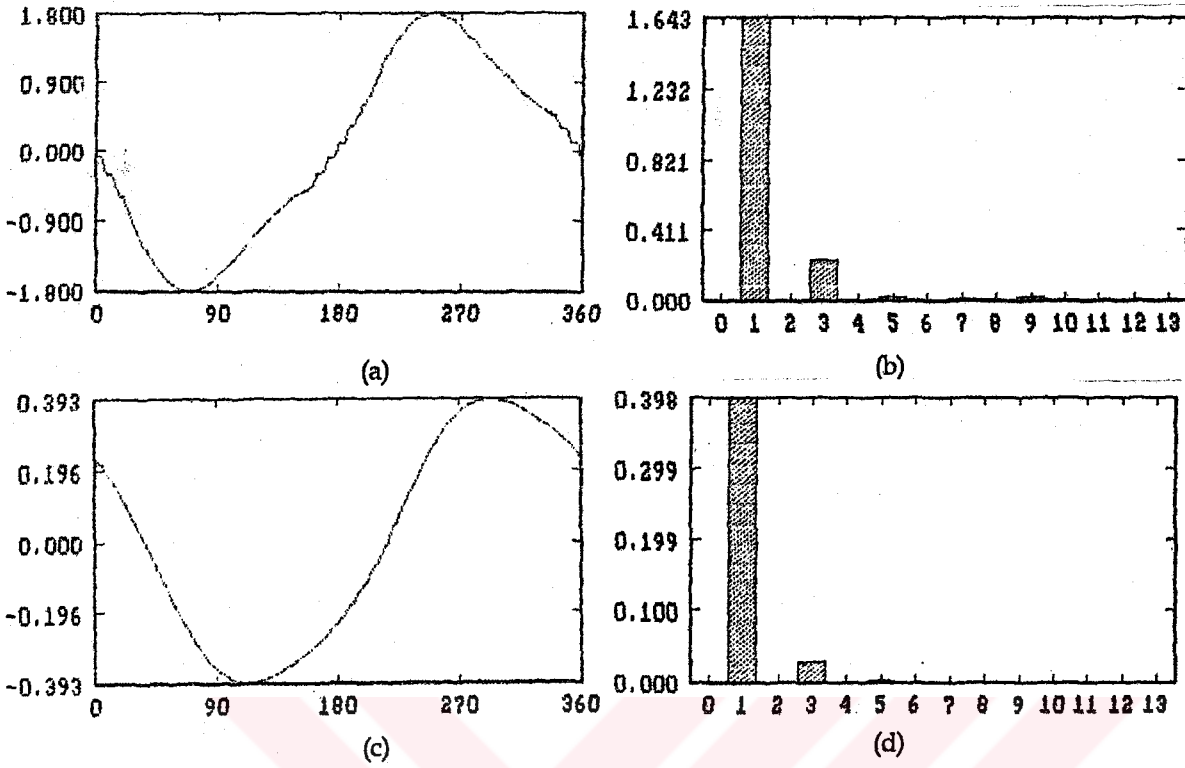
Şekil 6.4 b'de gösterilen devrede ara devre akım ara devresi olup b'deki devrede olduğu gibi L_S , L_Y ve C 'nin belirlediği rezonans frekansının etkisi söz konusudur. $R_S=0.1 \Omega$, $L_S=0.1 \text{ H}$ $C=776.795 \mu \text{ F}$ için elde edilen sonuçlar Şekil 6.15'de gösterilmiştir. Gerilim ve akım için (6.2)'den hesaplanan %THD'ları sırasıyla 0.26106 ve 0.0539 gibi oldukça düşük değerler olup yukarda aynı sayısal değerler için incelenen karma ara devreli (Şekil 6.11) den daha iyidir. Bu eleman değerleri için rezonans frekansı 50 Hz 'dir ve Şekil 6.15'deki sonuçlar da bu çalışma frekansı için elde edilmiştir.



Şekil 6.15 a) $R_S=0.1 \Omega$, $L_S=0,1 \text{ H}$, $R_Y=3 \Omega$, $L_Y=0.015 \text{ H}$ ve $C=776.795 \mu\text{F}$ için yük üzerindeki gerilim, b) gerilimin harmonik spektrumu, c) yük akımı, d) yük akımının harmonik spektrumu

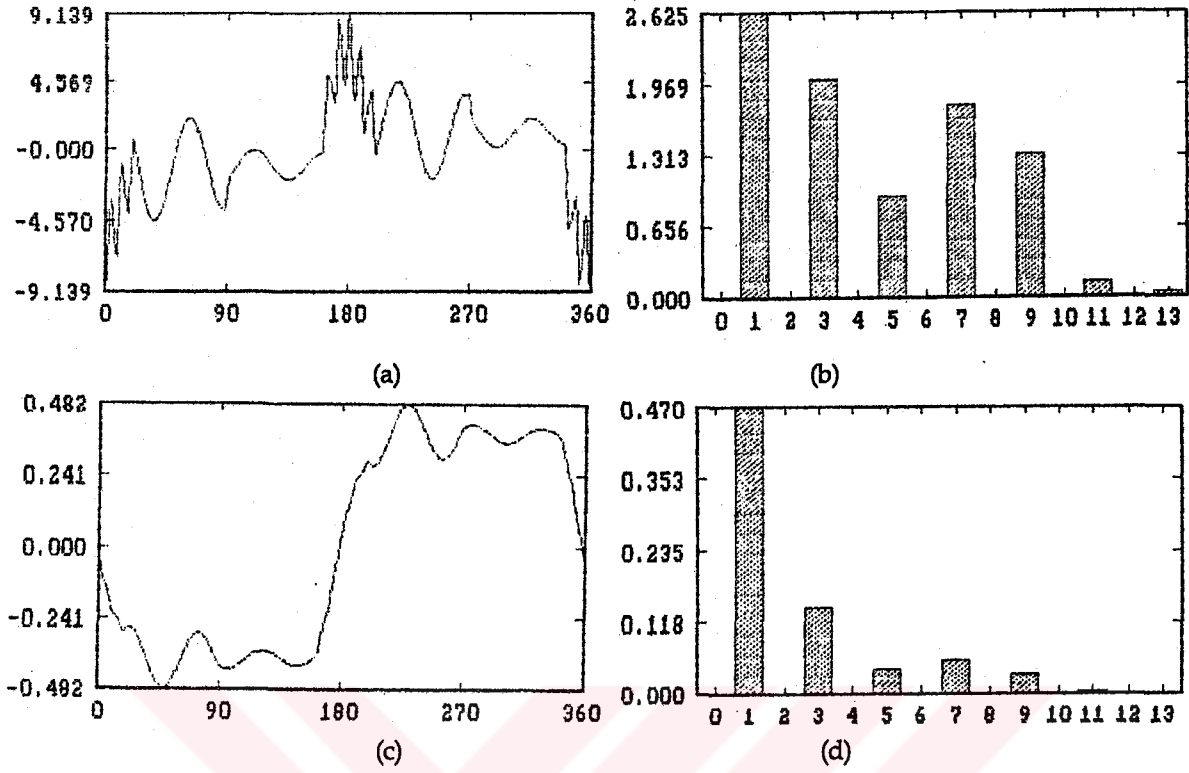
İnverterin frekansının daha düşük değerlerinde, örneğin 30 Hz'de bu durum biraz daha kötüleşecektir (Şekil 6.16). 30 Hz'de gerilim ve akım şekilleri sinüs biçiminden biraz daha uzaklaştıklarından %THD'ları 1.0096 ve 0.2735'e yükselmiştir.

Görülüyor ki akım ara devresindeki indüktans ile kullanılan tek yükün indüktansı göz önüne alınarak, rezonans frekansı en yüksek çalışma frekansına eşit olacak şekilde değeri belirlenen bir kondansatörün yüke paralel bağlanması ile yük akımının %THD'nu iyileştirilebilir. Ancak burada dikkat edilmesi gereken, kondansatörün çalışma frekansının en yüksek değerinde çok küçük reaktans değeri almayacak bir değerde olmasıdır. Burada söylenenlere uyulursa her çalışma frekansında yüke paralel kondansatörün etkisi farklı miktarlarda fakat olumlu olur.



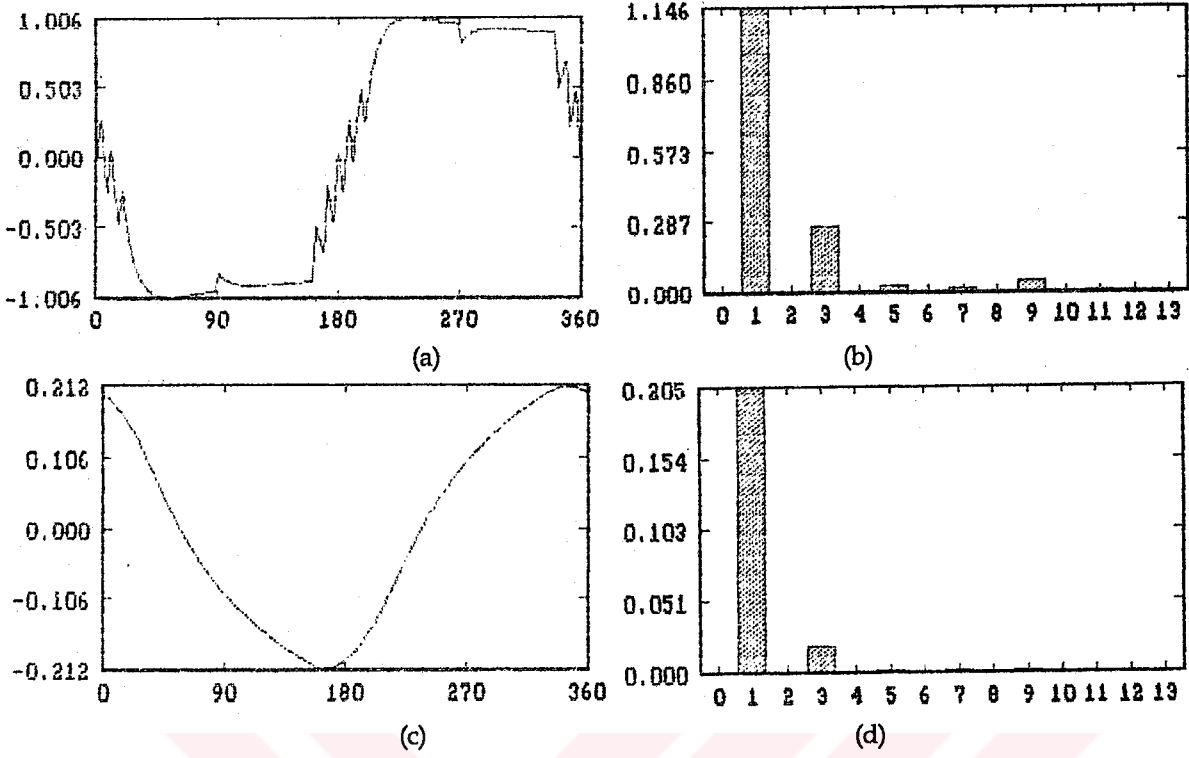
Şekil 6.16 a) 30 Hz çalışma frekansında Şekil 6.15'deki değerler için yük üzerindeki gerilim, b) gerilimin harmonik spektrumu, c) yük akımı, d) yük akımının harmonik spektrumu

Kondansatör değerinin $15.852 \mu\text{F}$ olduğu durum için yük gerilimi ve akımının 50 Hz'de bir periyot boyunca sürekli haldeki değişimleri ve harmonik spektrumları Şekil 6.17'de verilmiştir. Bu eleman değerleri için rezonans frekansı 350 Hz olduğu için bu defa etki olumsuzdur ve gerilim ve akım için %THD'ları 76.5057 ve 14.1268 değerlerine yükselmiştir. Bununla birlikte bu durum aynı eleman değerleri için Şekil 6.4 b'deki devrenin sonuçları olan Şekil 6.10 'dan daha kötü değildir.



Şekil 6.17 a) $R_s=0.1 \Omega$, $L_s=0,1 \text{ H}$, $R_y=3 \Omega$, $L_y=0.015 \text{ H}$ ve $C=15.852 \mu\text{F}$ için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu

Şekil 6.4 e'de gösterilen devrede R_s 'in sıfır değeri için durum değişkeni olan kondansatör geriliminde süreksizlik söz konusudur ve analiz için anahtarlama denkleminin yazılması gerekir. Burada çok iyi gerilim ara devresi ile yük tarafındaki kondansatör arasında pratikte bir küçük direnç olacağı kabulüyle, yük tarafına bağlanan kondansatörün etkisi incelenmiştir.



Şekil 6.18 a) $R_s=0.5 \Omega$, $R_y=3 \Omega$, $L_y=0.015 \text{ H}$ ve $C=1000 \mu \text{ F}$ için yük üzerindeki gerilim, b) gerilimin harmonik spekturumu, c) yük akımı, d) yük akımının harmonik spekturumu

Şekil 6.7'de gösterilenlerle karşılaştırmak amacıyla $R_s=0.5 \Omega$, $R_y=3 \Omega$, $L_y=0.015 \text{ H}$ ve $C=1000 \mu \text{ F}$ için elde edilen sonuçlar Şekil 6.18'de topluca gösterilmiştir. Bu şekilde gösterilen gerilim ve akım %THD'ları sırasıyla 2.74310 ve 0.58643 olup Şekil 6.7'dekilerden düşüktür. Daha düşük çalışma frekansları için bu distorsiyonlarda bir miktar artış olacaktır. Örneğin 30 Hz'de benzer analizler sonucu bulunan gerilim ve akım %THD'leri 5.8185 ve 0.86142 değerlerine yükselir.

7. SONUÇLAR

Bu tezde N-yollu süzgeçlerle gerçekleştirilen dar bantlı yüksek kalite faktörlü bant geçiren süzgeçlerle, tarak tipi süzgeçlerin tasarım prosedürü geliştirilmiştir. Filtreden beklenen transfer fonksiyonu karakteristiğinin, N-yollu filtrenin anahtarlarını süren fonksiyonla aralarındaki kesin benzerlikten hareket ederek uygun anahtarlama fonksiyonunu hesaplayacak bilgisayar programları hazırlanmıştır. Hazırlanan bilgisayar programları çeşitli örneklerle çalıştırılmış ve sonuçları birbirleriyle karşılaştırılmıştır.

Bu programlardan OPTI, merkez frekansı anahtarlama frekansına yerleşmiş olan dar bantlı bant geçiren filtre karakteristiği üretecek anahtarlama fonksiyonlarını bulur. NIKOP programında simetri ve işaret sınırlandırmaları gevşetildiğinden daha geneldir. NIKOP programıyla da merkez frekansı anahtarlama frekansına yerleşmiş dar bantlı filtre için en uygun ve OPTI programının bulduğu sonuç bulursa da OPTI, aynı problemin çözümünde çok daha hızlıdır. GENKDI bilgisayar programı ise ilgili bölümde ayrıntıları ile anlatıldığı gibi aralık kombinasyonları için tek tek denemeler yaptığından, aralık sayısının büyük olduğu problemlerde çok uzun zaman alır. Bu programlarla önceden bilinenlerden daha iyi sonuçlar elde edilmiştir.

NIKOP programıyla bulunan anahtarlama fonksiyonunun birinci harmoniğinden başkası da maksimize edilebileceğinden elde edilen dar bantlı bant geçiren süzgecin merkez frekansı bu maksimize edilen harmoniğin frekansına da yerleştirilebilir. Bu programla, artan frekanslarda piklerin genliklerinde daha az göreceli zayıflamanın olduğu seyrek dişli tarak tipi karakteristiklerde üretmek mümkündür. Bölüm 3'ün sonunda Şekil 3.13'de verilen karakteristiğin ne amaçla bir filtreden beklendiği tartışılabilir.

İlgili bölüm de ayrıntısıyla anlatıldığı gibi gerçekleştirilen 7-yollu bir anahtarlanmış kapasitör devresidir. Bununla birlikte test edilen uygulamalardan tatmin edici sonuçlar alınmıştır.

DGM'lu inverterle asenkron motorun kontrolunda ara devrelerin motor akımının harmonik distorsiyonuna etkileri araştırılmıştır. İrdemeler her bölümün içinde yapıldığından tekrardan kaçınmak için burada kısa cümlelerle yetinilmiştir.



KAYNAKLAR

BRUTON, L. T. ve PEDERSON, R. T., (1972). Time-Multiplexed Active Filters. *J. Solid-State Circuits.*, Vol. SC-7.

DARLINGTON, S., (1955). In Introduction to Time Variable Networks. *Proc. of the Midwest Symp. on Circuit Analysis.* Illinois Üniversitesi, Illinois, s. 5.1-5.25.

FETWEIS, A., (1971). Theory of Stop-Go N-Path Filters. *AEO*, Band.25.173-180.

FRANKS, L. E. ve DANN, L. D., (1970). Implementation of N-Path Filters Using Polarity Reversal Modulation. *Canadian Symp. on Comm.* Montreal. Canada.

FRANKS, L. E. ve SANDBERG, I. W., (1960). An Alternative Approach to the Realization of Network Transfer Functions: The N-path Filter. *Bell Syst. Tech.*3. Vol.39. 1321-1350.

HIRANO, K. ve NISHIMURA, S., (1972). Active RC Filters Containing Periodically Operated Switches. *IEEE Trans. Circuits Theory.*, Vol. CT-19, 253-260.

GLASER, A. B., HALKIAS, C. C. ve MEADOWS, H. E., (1969). A Tunable, Band Width-Adjustable Solid-State Filters. *J. Franklin Inst.*, Vol.288,83-98.

KAehler, J. A., (1969). Periodic-Switched Filter Networks-A Means of Amplifying and Varying Transfer Functions. *IEEE J. Solid-State Circuits.*, Vol. SC-4, 225-230.

KÖKSAL, M., (1979). N-Yollu Süzgeçler. TMMOB Elektrik Mühendisliği., Sayı. 265-266, 408-420.

KÖKSAL, M., (1975). Analysis and Applications of Linear Circuits Containing Periodically Operated Switches, Doktora Tezi, ODTÜ, Ankara.

KÖKSAL, M., (1984). A computer Program for the General solutions of Periodically Time-Varying Systems : MAINLN, Technical Report No: GEEE/CAS-84/2, ODTÜ, Gaziantep.

KÖKSAL, M. (1976). Dur-Geç Süzgeçler, TMMOB Elektrik Mühendisliği., Sayı. 237, 463-466.

LIN, H. C., DAVIS, W. R. ve SUN, Y., (1966). Active and Passive Multiplication of RLC Time Constants for Subaudio Frequency Integrated Filters. IEEE Region 6-th Ann. Conf., Vol. 1, 1-9.

PULE, C., (1971). An Odd and an Even N-Path Filter. Int. J. Electronics., Vol.30. No.3. 227-232.

SUN, Y. ve FRISH, I. T., (1968). Resistance Multiplication in Integrated Circuits by Means of Switching. IEEE Trans. Circuits Theory., Vol. CT-15, 184-192.

HENNING, H. H., (1969). A 96 Channel PCM Channel Bank. IEEE Int.Conf. on Communications. 34.17-34.22.

LANGER, E., (1968). Ein Neuartiges N-Pfad-Filter mit Zwei Konjugiert Komplexen Polpaaren. Frequenz 22. 90-95.

RYKEN, M., (1970). Unusual Filter Concept Promises High-Q, Low Frequency IC Device.

Electronic Design, Vol. 21, No.17: 21-24.

SUN, Y. ve FRISH, I. T., (1969). A General Theory of Commutated Networks. IEEE Trans.

Circuit Theory. Vol. CT-16. 502-508.

TATAR, Y., (1994). Sayısal İşaret İşlemci Temelli Darbe Genişlik Modülasyonlu Frekans

Eviricisi ile Sincap Kafesli Asenkron Motor Kontrolu için İşaret Üretimi. Doktora Tezi,
Fırat Üniversitesi, Elazığ.

TEMES, G. C. VE MİTRA, S. K., (1973). Modern Filter Theory and Design. A Wiley-

Interscience Publication.USA.

TRICK, T. N., (1973). An Equivalence Between Fast Periyodic-Switched Systems and Time-

Invariant Systems. IEEE Trans. Circuits Theory., Vol. CT-20, 324-325.

URKOWITZ, H., (1957). Analysis and Synthesis of Delay Line Periodic Filters. IRE

Trans. Circuit Theory. Vol.CT-4.41-53.

WUPPER, H., (1971). Low Frequency Comb Filter Containing a Novel Integrated

Device. Network and Signal Theory. 230-231. (a).

WUPPER, H., (1971). Low Frequency Comb Filter Containing a Novel Integrated

Device. Institut für Informationstechnik Ruhr-Universität Bochum. D-463 Bochum, F.R.
Germany. (b).

WUPPER, H., (1974). A Modified N-Path Filter Suited for Practical Realization. **IEEE Trans. Circuits and Systems**. Vol. CAS-21.



EK-1 OPTI BILGISA YAR PROGRAMININ LISTESI

```

DIMENSION TETA0(90),TETA1(90),G0(90),G1(90),GOBJ(90),A(90),GC(90)
& ZYF(90),B(90),BS(90),U1(90),U2(90),KY(90),ZREF(90)
INTEGER AR(90),ARG(90),ARM(90),ARS,ARK
OPEN(6,FILE='OPTI.YAZ',STATUS='OLD')
CALL GIR(KSE,N,M,NY,B,AK,AKAT,TT,KSK,K1,ZREF)
KW=0
LSAY=0
OBJRM=-1.E20
EPS=1.E-2
E1=1
MAKSIT=100000
ITER=1
PI=3.141592
KIS=4*N+(KSE/2)**2
ARS=NY*KIS
ARK=ARS/4+1
U=2*PI/(NY*KIS)
DO 39 I=1,N
39 TETA0(N+1-I)=1.57079-I*U
KS=0
8 CALL GTETA(KSE,N,M,TETA0,G0)
CALL AAI(M,G0,B,A)
OBJ0=OBJE(M,G0,A,AK)
IF(ITER.GT.1)GO TO 43
DO 5 I=3,M,2
5 ZYF(I)=20*ALOG10((G0(I)**2/G0(I)**2))
CALL YBAS(N,M,B,ZYF,TETA0,OBJ0,G0,AK,AKAT)
43 CONTINUE
KS=KS+1
IF(KS.NE.KSK)GO TO 351
DO 6 I=3,M,2
6 ZYF(I)=20*ALOG10((G0(I)**2/G0(I)**2))
CALL YA(N,M,TETA0,G0,ZYF,OBJ0,ITER,B,AK,AKAT)
KS=0
951 CONTINUE
T=TT
CALL GROBJ(N,M,TETA0,G0,GG,A,AK,GOBJ,DELTA)
C ANALITIK TUREV YERINE ASAG.SAYISAL TUREV ALAN NGR OBJ KULL.BILIR
C CALL NGR OBJ(N,M,TETA0,KSE,G0,A,B,AK,GOBJ,DELTA)
10 DO 15 I=1,N
15 TETA1(I)=TETA0(I)+(T/DELTA)*GOBJ(I)
CALL GTETA(KSE,N,M,TETA1,G1)
CALL AAI(M,G1,B,A)
OBJ1=OBJE(M,G1,A,AK)
IF(OBJ1.GT.OBJ0) GO TO 20
T=T/2
IF(T.LT.1.E-35) GO TO 998
GO TO 10
20 DO 25 I=1,N
25 IF(ABS(TETA1(I)-TETA0(I)).GT.EPS) GO TO 40
DO 325 I=3,M,2
325 ZYF(I)=20*ALOG10((G0(I)**2/G0(I)**2))
DO 470 I=3,M,2
470 IF(ZYF(I).LT.(B(I)-E1))GO TO 555
GO TO 999
555 AK=AK*AKAT
40 IF(ITER.GT.MAKSIT) GO TO 99
ITER=ITER+1
DO 50 I=1,N
50 TETA0(I)=TETA1(I)
GO TO 8
998 WRITE(*,*)T;1.E-35 OLDUGUNDAN BITIRDIM !!!!!!!!!!!!!!!
WRITE(6,*)T;1.E-35 OLDUGUNDAN BITIRDIM !!!!!!!!!!!!!!!
GO TO 999
99 WRITE(*,*)ITER BUYUK MAKSIT OLDUGUNDAN BITIRDIM !!!!!!!!!!!!!!!
WRITE(6,*)ITER BUYUK MAKSIT OLDUGUNDAN BITIRDIM !!!!!!!!!!!!!!!
999 CONTINUE
DO 53 I=3,M,2
53 ZYF(I)=20*ALOG10((G0(I)**2/G0(I)**2))
CALL YS(N,M,ITER,TETA0,OBJ0,G0,ZYF,B,AK,AKAT,T)
CALL DUSAR(KSE,N,TETA0,NY,ARG)
WRITE(6,*)'BULUNAN ACILARIN DUSTUGU ARALIKLAR'
WRITE(*,*)'BULUNAN ACILARIN DUSTUGU ARALIKLAR'
DO 510 I=1,N+(KSE/2)
WRITE(*,*)'ARG('I,')='ARG(I)
510 WRITE(6,*)'ARG('I,')='ARG(I)
WRITE(6,*)
CALL TAMAR(N,NY,ARG,KSE)
CALL CAK(KIS,NY,ARG,KY,KC1)
IF(KC1.EQ.0)GO TO 1252
DO 511 I=1,N
511 ARG(I)=ARG(I)-K1/2
1252 DO 512 I=1,N+1
512 AR(I)=ARG(I)
IF(KC1.EQ.0)GO TO 1253
122 CONTINUE
DO 793 I=1,N
793 IF(AR(I).GE.AR(I+1).OR.AR(I).LE.0)GO TO 534
KW=KW+1
GO TO 533
534 CONTINUE
GO TO 102

```



```

533 CALL TAMAR(N,NY,AR,KSE)
CALL CAK(KIS,NY,AR,KY,KC)
IF(KC.EQ.0)GO TO 9990
IF(KC.EQ.1)GO TO 102
9990 CONTINUE
DO 1592 I=1,N
1592 TETA0(I)=AR(I)*U-U/2
CALL GTETA(KSE,N,M,TETA0,G0)
CALL AAI(M,G0,B,A)
OBJR=OBJER(M,G0,ZREF)
IF(OBJR.LT.OBJRM)GO TO 102
OBJRM=OBJR
LSAY=LSAY+1
1253 DO 846 I=1,N+KSE/2
846 ARM(I)=AR(I)
IF(KC1.EQ.0)GO TO 1254
102 CALL DGETR(N,AR,ARG,K1,L)
IF(L.EQ.1)GO TO 122
IF(LSAY.LT.1)GO TO 3333
1254 CONTINUE
CALL TAMAR(N,NY,ARM,KSE)
WRITE(*,*)'SONUC BIR PERYOT BOYUNCA ARALIKLAR'
WRITE(6,*)'SONUC BIR PERYOT BOYUNCA ARALIKLAR'
DO 18 I=1,KIS
WRITE(*,*)I,' INCI PULSIN BULUNDUGU ARALIK= ',ARM(I)
18 WRITE(6,*)I,' INCI PULSIN BULUNDUGU ARALIK= ',ARM(I)
DO 592 I=1,KIS
592 TETA0(I)=ARM(I)*U-U/2
WRITE(*,*)'IMPULSLAR BU CAK.OLMAYAN ARALIKLARIN ORTASINA KOYULUR'
WRITE(6,*)'IMPULSLAR BU CAK.OLMAYAN ARALIKLARIN ORTASINA KOYULUR'
WRITE(6,*)'BIR PERYOT BOYUNCA ACILAR'
DO 221 I=1,KIS
TETA0D=TETA0(I)*180/3.141592
WRITE(*,*)'TETA0('I,')=',TETA0(I),'R',TETA0D,'D'
221 WRITE(6,*)'TETA0('I,')=',TETA0(I),'R',TETA0D,'D'
CALL GTETA(KSE,N,M,TETA0,G0)
WRITE(6,*)'GTETA IMPULSLAR UZERINDEN HARMONIK VE ZAYIFLAMALAR'
WRITE(*,*)'GTETA IMPULSLAR UZERINDEN HARMONIK VE ZAYIFLAMALAR'
CALL NORMZ(M,G0,ABN,ZYFN)
WRITE(*,*)'OBJRM=',OBJRM
WRITE(6,*)'OBJRM=',OBJRM
WRITE(6,*)'.....'
CALL SB2A(KSE,N,NY,ARM,U1,U2,BS,M)
WRITE(6,*)'SB2A UZERINDEN HARMONIK VE ZAYIFLAMALAR'
WRITE(*,*)'SB2A UZERINDEN HARMONIK VE ZAYIFLAMALAR'
CALL NORMZ(M,BS,BSN,ZYFN)
IF(KC1.EQ.0)GO TO 1255
GO TO 9494
3333 WRITE(*,*)'COZUM BULUNAMADI'
WRITE(6,*)'COZUM BULUNAMADI'
9494 WRITE(6,*)'KW=',KW,' KMBS DENENDI K1=',K1
WRITE(*,*)'KW=',KW,' KMBS DENENDI K1=',K1
STOP
1255 WRITE(6,*)'BULUNAN ACILARIN DUST.ARALIKLARDA CAK.OLMADIGINDAN'
WRITE(6,*)'KW YE BAGLI OLARAK KMBSLAR DENENMEDI.SONUC....'
WRITE(*,*)'BULUNAN ACILARIN DUST.ARALIKLARDA CAK.OLMADIGINDAN'
WRITE(6,*)'KW YE BAGLI OLARAK KMBSLAR DENENMEDI.SONUC....'
STOP
END
C*****
SUBROUTINE GIR(KSE,N,M,NY,B,AK,AKAT,TT,KSK,K1,ZREF)
DIMENSION B(1),ZREF(1)
WRITE(*,*)'DUZEN=1,PI/2 DE PULS YOK ?'
WRITE(*,*)'DUZEN=2,PI/2 DE PULS VAR ?'
READ(*,*)KSE
WRITE(*,*)'N DEGERINI GIRINIZ'
READ(*,*)N
WRITE(*,*)'M DEGERINI GIRINIZ'
READ(*,*)M
WRITE(*,*)'NY YOLSAYISI DEGERINI GIR'
READ(*,*)NY
DO 1 I=3,M,2
WRITE(*,*)'B('I,') DB OLARAK ZAYIFLAMA DEGERINI GIRINIZ'
1 READ(*,*)B(I)
WRITE(*,*)'AK DEGERINI GIRINIZ'
READ(*,*)AK
WRITE(*,*)'AKAT DEGERINI GIRINIZ:2,10..'
READ(*,*)AKAT
TT=1
WRITE(*,*)'KSK DEGERINI GIR KAC ADIMDA BIR YAZSIN:20,30..'
READ(*,*)KSK
WRITE(*,*)'KI DEGERINI GIRINIZ'
READ(*,*)K1
DO 86 I=3,M,2
WRITE(*,*)'ZREF('I,')=? dB'
86 READ(*,*)ZREF(I)
RETURN
END
C*****
SUBROUTINE YBAS(N,M,B,ZYF,TETA0,OBJ0,G0,AK,AKAT,ZREF)
DIMENSION B(1),ZYF(1),TETA0(1),G0(1),ZREF(1)
WRITE(6,*)'BASLANGIC DEGERLER'
DO 17 I=1,N

```

```

TETA0D=TETA0(I)*180/3.141592
17 WRITE(6,*)'TETA0('I,')=',TETA0(I),'R=',TETA0D,'D'
WRITE(6,*)'OBJ0=',OBJ0
DO 735 I=1,M,2
735 WRITE(6,*)'G0('I,')=',G0(I)
DO 731 I=3,M,2
731 WRITE(6,*)'IST.ZYF=',B(I),' SU AN ZYF=',ZYF(I),'DB'
WRITE(6,*)' AK=',AK,' AKAT=',AKAT
WRITE(6,*)'COZUM ETRAFI TARANIRKEN REFERANS ZAYIFLAMALAR'
DO 86 I=3,M,2
86 WRITE(6,*)'ZREF('I,')=? dB'
WRITE(6,*)'.....'
RETURN
END
C-----
SUBROUTINE YA(N,M,TETA0,G0,ZYF,OBJ0,ITER,B,AK,AKAT)
DIMENSION TETA0(1),G0(1),B(1),ZYF(1)
WRITE(*,*)'ARA SONUCLAR SADECE EKRANA YAZILIR.'
DO 12 I=1,N
TETA0D=TETA0(I)*180/3.141592
12 WRITE(*,*)'TETA0('I,')=',TETA0(I),'R=',TETA0D,'D'
WRITE(*,*)'OBJ0=',OBJ0,' ITER=',ITER
DO 535 I=1,M,2
535 WRITE(*,*)'G0('I,')=',G0(I)
DO 531 I=3,M,2
531 WRITE(*,*)'IST.ZYF=',B(I),' SU AN ZYF=',ZYF(I),'DB'
WRITE(*,*)' AK=',AK,' AKAT=',AKAT
WRITE(*,*)'.....'
RETURN
END
C-----
SUBROUTINE YS(N,M,ITER,TETA0,OBJ0,G0,ZYF,B,AK,AKAT,T)
DIMENSION TETA0(1),G0(1),B(1),ZYF(1)
WRITE(6,*)'BULUNAN DEGERLER'
WRITE(*,*)'BULUNAN DEGERLER'
WRITE(*,*)'ITER=',ITER
WRITE(6,*)'ITER=',ITER
DO 112 I=1,N
TETA0D=TETA0(I)*180/3.141592
WRITE(6,*)'TETA0('I,')=',TETA0(I),'R=',TETA0D,'D'
112 WRITE(*,*)'TETA0('I,')=',TETA0(I),'R=',TETA0D,'D'
WRITE(6,*)'OBJ0=',OBJ0
WRITE(*,*)'OBJ0=',OBJ0
DO 135 I=1,M,2
WRITE(6,*)'G0('I,')=',G0(I)
135 WRITE(*,*)'G0('I,')=',G0(I)
DO 131 I=3,M,2
WRITE(*,*)'IST.ZYF=',B(I),' SU AN ZYF=',ZYF(I),'DB'
131 WRITE(6,*)'IST.ZYF=',B(I),' SU AN ZYF=',ZYF(I),'DB'
WRITE(6,*)' AK=',AK,' AKAT=',AKAT,' T=',T
WRITE(*,*)' AK=',AK,' AKAT=',AKAT,' T=',T
WRITE(*,*)'.....'
WRITE(6,*)'.....'
RETURN
END
C-----
SUBROUTINE GTETA(KSE,N,M,TETA,G)
DIMENSION TETA(1),G(1)
DO 10 I=1,M
G(2*I-1)=0
SAB=(-1)**(I-1)*0.5
IF(KSE.EQ.1)SAB=0.
DO 20 J=1,N
20 G(2*I-1)=G(2*I-1)+SIN((2*I-1)*TETA(J))
10 G(2*I-1)=G(2*I-1)+SAB
RETURN
END
C-----
SUBROUTINE AAI(M,G,B,A)
DIMENSION G(1),B(1),A(1)
DO 45 I=3,M,2
45 A(I)=G(I)/SQRT(10**(B(I)/20))
RETURN
END
C-----
FUNCTION OBJE(M,G,A,AK)
DIMENSION G(1),A(1)
GA=0
DO 50 I=3,M,2
50 GA=GA+(G(I)**2-A(I)**2)**2
OBJE=G(1)-AK*GA
RETURN
END
C-----
FUNCTION OBJER(M,G,ZREF)
DIMENSION G(1),ZREF(1)
GA=0
DO 50 I=3,M,2
Z=20*A LOG10(ABS(G(1)/G(I)))
K=1
IF(Z.GT.ZREF(I))K=0
50 GA=GA+K*ABS(G(I))
OBJER=G(1)-GA

```

```
RETURN
END
```

```
C-----
SUBROUTINE GROBJ(N,M,TETA,G,GG,A,AK,GOBJ,DELTA)
DIMENSION TETA(1),G(1),A(1),GOBJ(1),GG(1)
DO 20 I=1,N
  GG(I)=0
DO 20 J=3,M,2
  GG(I)=GG(I)+4*(G(J)**2-A(J)**2)*G(J)*(J)*COS(J**TETA(I))
DO 30 I=1,N
  GOBJ(I)=COS(TETA(I))-AK*GG(I)
  DEL=0
DO 40 I=1,N
  DEL=DEL+GOBJ(I)**2
DELTA=SQRT(DEL)
RETURN
END
```

```
C-----
SUBROUTINE NGROBJ(N,M,TETA,KSE,G,A,B,AK,GOBJ,DELTA)
DIMENSION TETA(1),GOBJ(1),A(1),B(1)
DEL=0.0
DO 1 I=1,N
  EK=1.E-4
  EK=EK*ABS(TETA(I))
  CALL GTETA(KSE,N,M,TETA,G)
  CALL AA1(M,G,B,A)
  OBJI=OBJE(M,G,A,AK)
  TETA(I)=TETA(I)+EK
  CALL GTETA(KSE,N,M,TETA,G)
  CALL AA1(M,G,B,A)
  OBJIS=OBJE(M,G,A,AK)
  GOBJ(I)=(OBJIS-OBJI)/EK
  DEL=DEL+GOBJ(I)**2
  TETA(I)=TETA(I)-EK
1 CONTINUE
DELTA=SQRT(DEL)
RETURN
END
```

```
C-----
SUBROUTINE DUSAR(KSE,N,TETA,NY,AR)
DIMENSION TETA(1)
INTEGER AR(1)
PI=3.141592
KIS=4*N+(KSE/2)**2
IF(KSE.EQ.2)TETA(N+1)=PI/2
DO 92 I=1,N+KSE/2
  92 WRITE(6,*)TETA(I),'='?,TETA(I)
  U=2*PI/(NY*KIS)
  DO 2 J=1,NY*KIS/4+1
  DO 2 J=1,N+KSE/2
  IF((TETA(J).GT.(I-1)*U).AND.(TETA(J).LE.I*U))GO TO 7
  GO TO 2
  7 AR(J)=I
  2 CONTINUE
  DO 93 I=1,N+KSE/2
  WRITE(*,*)AR(I),'='?,AR(I)
  93 WRITE(6,*)AR(I),'='?,AR(I)
  111 DO 100 I=1,N+KSE/2-1
  IF(AR(I).GE.AR(I+1).AND.AR(1).EQ.1.AND.AR(I)+1.LT.AR(N+KSE/2))
  &AR(I+1)=AR(I)+1
  100 IF(AR(I).GE.AR(I+1).AND.AR(N+KSE/2).EQ.NY*KIS/4+1)AR(I)=AR(I)-1
  DO 101 I=1,N+KSE/2-1
  101 IF(AR(I).GE.AR(I+1))GO TO 111
  WRITE(6,*)'AYNI ARALIGA DUSEN ACILAR VARSA DUZELTILDI'
  WRITE(*,*)'AYNI ARALIGA DUSEN ACILAR VARSA DUZELTILDI'
  DO 94 I=1,N+KSE/2
  WRITE(*,*)AR(I),'='?,AR(I)
  94 WRITE(6,*)AR(I),'='?,AR(I)
RETURN
END
```

```
C-----
SUBROUTINE TAMAR(N,NY,AR,KSE)
INTEGER AR(1),ARS,ARK
KIS=4*N+(KSE/2)**2
ARS=NY*KIS
ARK=ARS/4+1
IF(KSE.EQ.2)AR(N+1)=ARK
IF(KSE.EQ.1)KH1=1
IF(KSE.EQ.2)KH1=0
DO 1 I=N+KSE/2+1,KIS/2
  1 AR(I)=ARK-KH1+(ARK-AR(KIS/2+1-1))
DO 2 I=KIS/2+1,KIS
  2 AR(I)=AR(I-KIS/2)+ARS/2
RETURN
END
```

```
C-----
SUBROUTINE CAK(KIS,NY,AR,KY,KC)
DIMENSION KY(1)
INTEGER AR(1),ARS
ARS=NY*KIS
DO 3 I3=1,NY-1
DO 1 I1=1,KIS
  KY(I1)=AR(I1)+I3*(ARS/NY)
```

```

IF(KY(11).GT.ARS)KY(11)=KY(11)-ARS
DO 2 I2=1,KIS
IF(AR(I2).EQ.KY(I1))GO TO 33
2 CONTINUE
GO TO 1
33 KC=1
GO TO 999
1 CONTINUE
3 CONTINUE
KC=0
999 RETURN
END
-----
SUBROUTINE SB2A(KSE,N,NY,AR,U1,U2,BS,NT)
DIMENSION BS(1),U1(1),U2(1)
INTEGER AR(1)
KIS=4*N+(KSE/2)*2
PI=3.141592
U=2*PI/(NY*KIS)
DO 292 I=1,KIS
U1(I)=(AR(I)-1)*U
292 U2(I)=AR(I)*U
WRITE(*,*)'CIZGI SPEKTRUMU ICIN'
WRITE(*,*)'ZAIS2 DEN BEKLENE ZAYIFLAMALAR'
WRITE(6,*)'CIZGI SPEKTRUMU ICIN'
WRITE(6,*)'ZAIS2 DEN BEKLENE ZAYIFLAMALAR'
DO 121 I=1,NT,2
BS(I)=0.
DO 1021 J=1,KIS
BB=-1
IF(J.LE.KIS/2)BB=1
1021 BS(I)=BS(I)-BB*(1/(1*PI))*(COS(I*U2(J))-COS(I*U1(J)))
Z=-20.*ALOG10((BS(1)**2/BS(I)**2))
WRITE(*,*)I,BS(I),Z
121 WRITE(6,*)I,BS(I),Z
WRITE(6,*)
RETURN
END
-----
SUBROUTINE NORMZ(NT,AB,ABN,ZYFN)
DIMENSION AB(1)
WRITE(*,*)'NORMALIZE'
WRITE(6,*)'NORMALIZE'
ABMAX=0.0
DO 1 I=1,NT,2
IF(ABS(AB(I)).LT.1.E-35)AB(I)=1.E-35
1 IF(ABS(AB(I)).GT.ABMAX)ABMAX=ABS(AB(I))
DO 2 I=1,NT,2
ABN=ABS(AB(I))/ABMAX
ZYFN=20.*ALOG10(ABN)
WRITE(*,*)I,ABN,ZYFN
2 WRITE(6,*)I,ABN,ZYFN
RETURN
END
-----
SUBROUTINE DGETR(N,AR,ARG,K1,L)
INTEGER AR(1),ARG(1)
L=0
IF(N.EQ.0)GO TO 666
AR(N)=AR(N)+1
IF(AR(N).LE.ARG(N)+K1)GO TO 122
IF(N.EQ.1)GO TO 666
AR(N-1)=AR(N-1)+1
AR(N)=ARG(N)
IF(AR(N-1).LE.ARG(N-1)+K1)GO TO 122
IF(N.EQ.2)GO TO 666
AR(N-2)=AR(N-2)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
IF(AR(N-2).LE.ARG(N-2)+K1)GO TO 122
IF(N.EQ.3)GO TO 666
AR(N-3)=AR(N-3)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
IF(AR(N-3).LE.ARG(N-3)+K1)GO TO 122
IF(N.EQ.4)GO TO 666
AR(N-4)=AR(N-4)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
IF(AR(N-4).LE.ARG(N-4)+K1)GO TO 122
IF(N.EQ.5)GO TO 666
AR(N-5)=AR(N-5)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
IF(AR(N-5).LE.ARG(N-5)+K1)GO TO 122
IF(N.EQ.6)GO TO 666
AR(N-6)=AR(N-6)+1

```

```

AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
IF(AR(N-6).LE.ARG(N-6)+K1)GO TO 122
IF(N.EQ.7)GO TO 666
AR(N-7)=AR(N-7)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
IF(AR(N-7).LE.ARG(N-7)+K1)GO TO 122
IF(N.EQ.8)GO TO 666
AR(N-8)=AR(N-8)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
IF(AR(N-8).LE.ARG(N-8)+K1)GO TO 122
IF(N.EQ.9)GO TO 666
AR(N-9)=AR(N-9)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
IF(AR(N-9).LE.ARG(N-9)+K1)GO TO 122
IF(N.EQ.10)GO TO 666
AR(N-10)=AR(N-10)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
IF(AR(N-10).LE.ARG(N-10)+K1)GO TO 122
IF(N.EQ.11)GO TO 666
AR(N-11)=AR(N-11)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
IF(AR(N-11).LE.ARG(N-11)+K1)GO TO 122
IF(N.EQ.12)GO TO 666
AR(N-12)=AR(N-12)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
IF(AR(N-12).LE.ARG(N-12)+K1)GO TO 122
IF(N.EQ.13)GO TO 666
AR(N-13)=AR(N-13)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)

```

```

AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
IF(AR(N-13).LE.ARG(N-13)+K1)GO TO 122
IF(N.EQ.14)GO TO 666
AR(N-14)=AR(N-14)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
IF(AR(N-14).LE.ARG(N-14)+K1)GO TO 122
IF(N.EQ.15)GO TO 666
AR(N-15)=AR(N-15)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
IF(AR(N-15).LE.ARG(N-15)+K1)GO TO 122
IF(N.EQ.16)GO TO 666
AR(N-16)=AR(N-16)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
IF(AR(N-16).LE.ARG(N-16)+K1)GO TO 122
IF(N.EQ.17)GO TO 666
AR(N-17)=AR(N-17)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
IF(AR(N-17).LE.ARG(N-17)+K1)GO TO 122
IF(N.EQ.18)GO TO 666
AR(N-18)=AR(N-18)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)

```

```

AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
IF(AR(N-18).LE.ARG(N-18)+K1)GO TO 122
IF(N.EQ.19)GO TO 666
AR(N-19)=AR(N-19)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
IF(AR(N-19).LE.ARG(N-19)+K1)GO TO 122
IF(N.EQ.20)GO TO 666
AR(N-20)=AR(N-20)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
AR(N-19)=ARG(N-19)
IF(AR(N-20).LE.ARG(N-20)+K1)GO TO 122
IF(N.EQ.21)GO TO 666
AR(N-21)=AR(N-21)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
AR(N-19)=ARG(N-19)
AR(N-20)=ARG(N-20)
IF(AR(N-21).LE.ARG(N-21)+K1)GO TO 122
IF(N.EQ.22)GO TO 666
AR(N-22)=AR(N-22)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)

```

```
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
AR(N-19)=ARG(N-19)
AR(N-20)=ARG(N-20)
AR(N-21)=ARG(N-21)
IF(AR(N-22).LE.ARG(N-22)+K1)GO TO 122
IF(N.EQ.23)GO TO 666
122 L=1
666 RETURN
END
```



EK-2 NIKOP BILGISAYAR PROGRAMININ LİSTESİ

```

DIMENSION TETA(90),A(90),B(90),AB(90),KJ(90),ABN(90),ABNK(90)
&,ZABNK(90),HIST(90),ISRT(90),KY(90),GOBJ(90),EK(90),TETAB(900)
&,OB(1024),HISTB(90),OBB(1024),OBF(1024),KJF(1024,10)
&,ABNBK(1024,10),KJB(1024,12),KJB1(1024,10),KJB2(1024,10)
INTEGER BM(1024,10),AR(90)
OPEN(6,FILE='NIKOP.YAZ',STATUS='OLD')
CALL GIR(KIS,M,HISTB,NY)
CALL BASL(KIS,TETAB)
WRITE(*,*)'VERILEN ACILARIN DUSTUGU ARALIKLAR'
WRITE(6,*)'VERILEN ACILARIN DUSTUGU ARALIKLAR'
CALL DUSART(KIS,NY,TETAB,AR)
CALL SUBP(KIS,BM,TETAB,A,B,AB,KJ,ABN,ABNK,ZABNK,ABNBK,HIST,KJB,
&KJB1,KJB2,OB,OBF,KJF,IF)
DO 2309 I=0,M
2309 HIST(I)=HISTB(I)
CALL OP(KIS,IF,TETA,TETAB,KJ,KJF,M,A,B,AB,ABN,ABNK,ZABNK,HIST,OBJ
&,EK,ISRT,GOBJ,AR,KY,OBB,NY)
CALL SIRALA2(IF,OBB)
STOP
END
C*****
SUBROUTINE GIR(KIS,M,HISTB,NY)
DIMENSION HISTB(1)
WRITE(*,*)'KIS DEGERINI GIR'
READ(*,*)KIS
WRITE(*,*)'M DEGERINI GIR ; VEYA ESIT NGH OLMALI'
READ(*,*)M
WRITE(*,*)'TRANSFER FONKSIYONUNDAN ISTENEN NORML DEGERLER'
DO 343 I=0,M
WRITE(*,*)I,'INCI HARM.DE ISTENEN NORML DEGER=? =,; 1'
READ(*,*)HISTB(I)
343 WRITE(6,*)I,'INCI HARM.DE ISTENEN NORML DEGER=? =,; 1',HISTB(I)
WRITE(*,*)'NY YOLSAYISI DEGERINI GIR'
READ(*,*)NY
RETURN
END
C*****
SUBROUTINE OP(KIS,IF,TETA,TETAB,KJ,KJF,M,A,B,AB,ABN,ABNK,ZABNK,
&HIST,OBJ,EK,ISRT,GOBJ,AR,KY,OBB,NY)
DIMENSION TETA(1),TETAB(1),KJ(1),A(1),B(1),AB(1),ABN(1),
&ABNK(1),ZABNK(1),HIST(1),EK(1),ISRT(1),GOBJ(1),KY(1),OBB(1)
&,KJF(1024,10)
INTEGER AR(1)
IO=0
PI=3.141592
U=2*PI/(NY*KIS)
DELT1=1.E-5
DELT2V=.5
WRITE(6,*)'DELT1=?',DELT1
WRITE(6,*)'DELT2V=?',DELT2V
DO 7575 IIF=1,IF
DO 1 I=1,KIS
TETA(I)=TETAB(I)
KJ(I)=KJF(IIF,I)
1 WRITE(6,*)I,' INCI PULSIN YONU=?',KJ(I)
CALL GH(KIS,M,TETA,A,B,AB,KJ,ORT)
CALL NRML(M,A,B,AB,ABN,ABNK,ZABNK)
CALL OBJECT(M,HIST,ABNK,OBJ)
222 CONTINUE
DELT2=DELT2V
CALL GH(KIS,M,TETA,A,B,AB,KJ,ORT)
CALL NRML(M,A,B,AB,ABN,ABNK,ZABNK)
CALL OBJECT(M,HIST,ABNK,OBJ)
OBJB=OBJ
DELTA=0
DO 555 I=1,KIS
TETA(I)=TETA(I)+DELT1
CALL GH(KIS,M,TETA,A,B,AB,KJ,ORT)
CALL NRML(M,A,B,AB,ABN,ABNK,ZABNK)
CALL OBJECT(M,HIST,ABNK,OBJ)
GOBJ(I)=ABS(OBJB-OBJ)/OBJB
IF(GOBJ(I).LT.1.E-20)GOBJ(I)=1.E-20
DELTA=DELTA+GOBJ(I)
ISRT(I)=1
IF(OBJ.GT.OBJB)ISRT(I)=-1
TETA(I)=TETA(I)-DELT1
555 CONTINUE
500 CONTINUE
DO 556 I=1,KIS
EK(I)=ISRT(I)*DELT2*GOBJ(I)/DELTA
556 TETA(I)=TETA(I)+EK(I)
DO 5661 I=1,KIS
5661 IF(ABS(EK(I)).GT.1.E-3)GO TO 3434
GO TO 225
3434 CONTINUE
CALL GH(KIS,M,TETA,A,B,AB,KJ,ORT)
CALL NRML(M,A,B,AB,ABN,ABNK,ZABNK)
CALL OBJECT(M,HIST,ABNK,OBJ)
OBJS=OBJ
IF(OBJS.GE.OBJB)GO TO 224
OBES=ABS(OBJB-OBJS)
WRITE(*,*)'OBJB='.OBJB,' OBES='.OBES

```

```

GO TO 222
224 CONTINUE
DO 256 I=1,KIS
256 TETA(I)=TETA(I)-EK(I)
DELT2=DELT2/2
GO TO 500
225 CONTINUE
DO 211 I=1,KIS
211 TETA(I)=TETA(I)-EK(I)
CALL GH(KIS,M,TETA,A,B,AB,KJ,ORT)
CALL NRML(M,A,B,AB,ABN,ABNK,ZABNK)
CALL OBJECT(M,HIST,ABNK,OBJ)
IO=IO+1
OBB(IO)=OBJ
CALL DOK(KIS,TETA,ORT,A,B,AB,IO,M,ABN,ABNK,ZABNK,OBJ)
WRITE(*,*)'BULUNAN AÇILARIN DUSTUGU ARALIKLAR'
WRITE(6,*)'BULUNAN AÇILARIN DUSTUGU ARALIKLAR'
CALL DUSART(KIS,NY,TETA,AR)
DO 592 I=1,KIS
592 TETA(I)=AR(I)*U-U/2
CALL GH(KIS,M,TETA,A,B,AB,KJ,ORT)
CALL NRML(M,A,B,AB,ABN,ABNK,ZABNK)
CALL OBJECT(M,HIST,ABNK,OBJ)
WRITE(*,*)'BU ARALIKLARIN ORTASI=TETA ALARAK DOKUM:.....'
WRITE(6,*)'BU ARALIKLARIN ORTASI=TETA ALARAK DOKUM:.....'
DO 967 I=1,KIS
TETAD=TETA(I)*180/PI
WRITE(*,*)'TETA('I,')=',TETA(I),'R',' TETAD=',TETAD,'D'
967 WRITE(6,*)'TETA('I,')=',TETA(I),'R',' TETAD=',TETAD,'D'
WRITE(*,*)'ORT=',ORT
WRITE(6,*)'ORT=',ORT
WRITE(*,*)'1.SUTUN HARMONIK NUMARASI'
WRITE(6,*)'1.SUTUN HARMONIK NUMARASI'
WRITE(*,*)'2.SUTUN A LAR COSINUSLU TERIMLERIN KATSAYILARI'
WRITE(6,*)'2.SUTUN A LAR COSINUSLU TERIMLERIN KATSAYILARI'
WRITE(*,*)'3.SUTUN B LER SINUSLU TERIMLERIN KATSAYILARI'
WRITE(6,*)'3.SUTUN B LER SINUSLU TERIMLERIN KATSAYILARI'
WRITE(*,*)'AB(0)=ABS(ORT)*2=',AB(0)
WRITE(6,*)'
IO=0
A(0)=0
B(0)=0
AB(0)=ABS(ORT)*2
WRITE(*,*)IO,A(0),B(0),AB(0)
WRITE(6,*)IO,A(0),B(0),AB(0)
DO 9305 I=1,M
WRITE(6,*)I,A(I),B(I),AB(I)
9305 WRITE(*,*)I,A(I),B(I),AB(I)
WRITE(6,*)'
WRITE(*,*)'IKINCI SUTUN GTS9755.FOR UN NORM DEGERLERI ABN LER'
WRITE(6,*)'IKINCI SUTUN GTS9755.FOR UN NORM DEGERLERI ABN LER'
WRITE(*,*)'UCUNCU SUTUN TRF.FONK.DAN BEKLENEEN NORM DEGERLER ABNK'
WRITE(6,*)'UCUNCU SUTUN TRF.FONK.DAN BEKLENEEN NORM DEGERLER ABNK'
WRITE(*,*)'DORDUNCU SUTUN TRF.FONK.DAN BEKL NORM DEGLR ZAYFL,db'
WRITE(6,*)'DORDUNCU SUTUN TRF.FONK.DAN BEKL NORM DEGLR ZAYFL,db'
WRITE(6,*)'
DO 9445 I=0,M
WRITE(*,*)I,ABN(I),ABNK(I),ZABNK(I)
WRITE(6,*)I,ABN(I),ABNK(I),ZABNK(I)
9445 CONTINUE
WRITE(*,*)'OBJ=',OBJ
WRITE(6,*)'OBJ=',OBJ
CALL CAK(KIS,NY,AR,KY,KC)
7575 CONTINUE
RETURN
END
C-----
SUBROUTINE BASL(KIS,TETA)
DIMENSION TETA(1)
PI=3.141592
AKIS=KIS
KSE=1
IF(AKIS/4.GT.KIS/4)KSE=2
N=(KIS-(KSE/2)*2)/4
A=PI/(2*(N+1))
DO 1 I=1,N+KSE/2
1 TETA(I)=A*I
DO 5023 I=1,N+1
DO 5023 I=1,N+KSE/2+1/KSE
5023 TETA(N+KSE/2+1)=PI/2+(PI/2-TETA(N+1/KSE+KSE/2-1))
DO 5024 I=1,2*N+KSE/2
5024 TETA(2*N+KSE/2+1)=PI+TETA(I)
WRITE(6,*)'BIR PERYOTTAKI PULS SAYISI=',KIS
WRITE(*,*)'BIR PERYOTTAKI PULS SAYISI=',KIS
DO 2 I=1,KIS
WRITE(*,*)'TETA('I,')=',TETA(I)
2 WRITE(6,*)'TETA('I,')=',TETA(I)
WRITE(6,*)'
WRITE(*,*)'
RETURN
END
C-----
SUBROUTINE DOK(KIS,TETA,ORT,A,B,AB,IO,M,ABN,ABNK,ZABNK,OBJ)

```

```

DIMENSION TETA(1),A(1),B(1),AB(1),ABN(1),ABNK(1),ZABNK(1)
PI=3.141592
WRITE(*,*)'BULUNAN SONUCUN AYRINTILI DOKUMU:.....'
WRITE(6,*)'BULUNAN SONUCUN AYRINTILI DOKUMU:.....'
DO 567 I=1,KIS
TETAD=TETA(I)*180/PI
WRITE(*,*)TETA('I,')=,TETA(I),'R, ' TETAD=,TETAD,'D'
567 WRITE(6,*)TETA('I,')=,TETA(I),'R, ' TETAD=,TETAD,'D'
WRITE(*,*)'ORT=' ,ORT
WRITE(6,*)'ORT=' ,ORT
WRITE(*,*)'1.SUTUN HARMONIK NUMARASI'
WRITE(6,*)'1.SUTUN HARMONIK NUMARASI'
WRITE(*,*)'2.SUTUN A LAR COSINUSLU TERIMLERIN KATSAYILARI'
WRITE(6,*)'2.SUTUN A LAR COSINUSLU TERIMLERIN KATSAYILARI'
WRITE(*,*)'3.SUTUN B LER SINUSLU TERIMLERIN KATSAYILARI'
WRITE(6,*)'3.SUTUN B LER SINUSLU TERIMLERIN KATSAYILARI'
WRITE(6,*)'AB(0)=ABS(ORT)**2=,AB(0)
WRITE(6,*)'
IO=0
A(0)=0
B(0)=0
AB(0)=ABS(ORT)**2
WRITE(*,*)IO,A(IO),B(IO),AB(IO)
WRITE(6,*)IO,A(IO),B(IO),AB(IO)
DO 7305 I=1,M
WRITE(6,*)I,A(I),B(I),AB(I)
7305 WRITE(*,*)I,A(I),B(I),AB(I)
WRITE(6,*)'
WRITE(*,*)'IKINCI SUTUN GTS97S5.FOR UN NORM DEGERLERI ABN LER'
WRITE(6,*)'IKINCI SUTUN GTS97S5.FOR UN NORM DEGERLERI ABN LER'
WRITE(*,*)'UCUNCU SUTUN TRF.FONK.DAN BEKLENEEN NORM DEGERLER ABNK'
WRITE(6,*)'UCUNCU SUTUN TRF.FONK.DAN BEKLENEEN NORM DEGERLER ABNK'
WRITE(*,*)'DORDUNCU SUTUN TRF.FONK.DAN BEKL NORM DEGLR ZAYFL,db'
WRITE(6,*)'DORDUNCU SUTUN TRF.FONK.DAN BEKL NORM DEGLR ZAYFL,db'
WRITE(6,*)'
DO 7445 I=0,M
WRITE(*,*)I,ABN(I),ABNK(I),ZABNK(I)
WRITE(6,*)I,ABN(I),ABNK(I),ZABNK(I)
7445 CONTINUE
WRITE(*,*)'OBJ=' ,OBJ
WRITE(6,*)'OBJ=' ,OBJ
RETURN
END
C*****
SUBROUTINE GH(KIS,M,TETA,A,B,AB,KJ,ORT)
DIMENSION TETA(1),A(1),B(1),AB(1),KJ(1)
PI=3.141592
NS=KIS
ORT=0.
DO 592 I=1,KIS
592 ORT=ORT+KJ(I)
ORT=ORT/NS
DO 10 I=1,M
A(I)=0.
B(I)=0.
DO 11 J=1,KIS
A(I)=A(I)+(2./NS)*KJ(J)*COS(I*TETA(J))
11 B(I)=B(I)+(2./NS)*KJ(J)*SIN(I*TETA(J))
10 AB(I)=SQRT(A(I)**2+B(I)**2)
AB(0)=ABS(ORT)**2
RETURN
END
C*****
SUBROUTINE NRML(M,A,B,AB,ABN,ABNK,ZABNK)
DIMENSION A(1),B(1),AB(1),ABN(1),ABNK(1),ZABNK(1)
ABNRM=1.E-30
DO 665 I=0,M
665 IF(AB(I).GT.ABNRM)ABNRM=AB(I)
DO 7445 I=0,M
ABN(I)=AB(I)/ABNRM
IF(ABN(I).LE.1.E-30)ABN(I)=1.E-30
ABNK(I)=ABN(I)**2
IF(ABNK(I).LE.1.E-30)ABNK(I)=1.E-30
ZABNK(I)=20*ALOG10(1./ABNK(I))
7445 CONTINUE
RETURN
END
C*****
SUBROUTINE OBJECT(M,HIST,ABNK,OBJ)
DIMENSION HIST(1),ABNK(1)
OBJ=0
DO 2 I=0,M
2 OBJ=OBJ+(HIST(I)-ABNK(I))**2
RETURN
END
C*****
SUBROUTINE DUSART(KIS,NY,TETA,AR)
DIMENSION TETA(1)
INTEGER AR(1)
PI=3.141592
U=2*PI/(NY*KIS)
30 DO 32 I=1,NY*KIS
32 IF(TETA(I).GT.2*PI)TETA(I)=TETA(I)-2*PI

```

```

      DO 33 I=1,NY*KIS
      33 IF(TETA(I).GT.2*PI)GO TO 30
      301 DO 321 I=1,NY*KIS
      321 IF(TETA(I).LT.0)TETA(I)=TETA(I)+2*PI
            DO 331 I=1,NY*KIS
      331 IF(TETA(I).LT.0)GO TO 301
            DO 23 I1=1,NY*KIS
            DO 23 J=1,KIS
            IF((TETA(J).GT.(I1-1)*U).AND.(TETA(J).LE.I1*U))GO TO 73
            GO TO 23
      73 AR(J)=I1
      23 CONTINUE
            DO 81 I=1,KIS
            WRITE(*,*)AR('I,')=',AR(I)
      81 WRITE(6,*)AR('I,')=',AR(I)
            RETURN
            END
C-----
      SUBROUTINE CAK(KIS,NY,AR,KY,KC)
      DIMENSION KY(1)
      INTEGER AR(1),ARS
      ARS=NY*KIS
      DO 3 I3=1,NY-1
      DO 1 I1=1,KIS
      ISRT=1
      IF(AR(I1).LT.0)ISRT=-1
      KY(I1)=ABS(AR(I1))+I3*KIS
      IF(ABS(KY(I1)).GT.ARS)KY(I1)=ABS(KY(I1))-ARS
      KY(I1)=KY(I1)*ISRT
      DO 2 I2=1,KIS
      IF(ABS(AR(I2)).EQ.ABS(KY(I1)))GO TO 33
      2 CONTINUE
      GO TO 1
      33 KC=1
      WRITE(*,*)'CAKISMA VAR'
      WRITE(6,*)'CAKISMA VAR'
      WRITE(*,*)'VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV'
      WRITE(6,*)'VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV'
      GO TO 999
      1 CONTINUE
      3 CONTINUE
      KC=0
      WRITE(*,*)'CAKISMA YOK'
      WRITE(6,*)'CAKISMA YOK'
      WRITE(*,*)'VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV'
      WRITE(6,*)'VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV'
      999 RETURN
      END
C-----
      SUBROUTINE SUBF(KIS,BM,TETA,A,B,AB,KJ,ABN,ABNK,ZABNK,ABNKB,HIST,
      &KJB,KJB1,KJB2,OB,OF,KJF,IF)
      DIMENSION TETA(1),A(1),B(1),AB(1),KJ(1),ABN(1),ABNK(1),ZABNK(1),
      &HIST(1),KJB1(1,1),KJB2(1,1),OF(900),KJF(1024,KIS)
      &,OB(2**(KIS-1)),KJB(2**(KIS-1),KIS),ABNK(2**(KIS-1),8)
      INTEGER BM(2**KIS,KIS)
      M=KIS/2
      DO 313 I=0,M
      313 HIST(I)=(I+1)**10/(M+1)
      NSAT=2**(KIS-1)
      NSUT=KIS
      DO 44 J=1,KIS
      DO 44 I=1,2**J
      BM(I,J)=1
      IF(I.GT.(2**J)/2)BM(I,J)=-1
      WRITE(*,*)'KONTROL',J,I
      44 CONTINUE
      DO 11 J=1,KIS
      DO 1 I=(2**J)+1,2**(KIS-1)
      1 BM(I,J)=BM(I-(2**J),J)
      11 CONTINUE
      DO 76 I=1,NSAT
      DO 77 J=1,NSUT
      77 KJ(J)=BM(I,J)
      CALL GH(KIS,M,TETA,A,B,AB,KJ,ORT)
      CALL NRML(M,A,B,AB,ABN,ABNK,ZABNK)
      CALL OBJECT(M,HIST,ABNK,OBJ)
      OBJ=OBJ
      DO 449 IB=1,KIS
      KJB(I,IB)=KJ(IB)
      449 CONTINUE
      DO 454 IB=0,M
      454 ABNK(I,IB)=ABNK(IB)
      76 CONTINUE
      CALL SIRALA(NSAT,OB,KJB,KIS,KJB1,KJB2)
      CALL FARKLIF(NSAT,OB,OF,KIS,KJB,KJF,IF)
      WRITE(6,*)'SUBF SONUCLARI BASLANGICI'
      WRITE(*,*)'FARKLI DEGER SAYISI=IF=',IF
      WRITE(6,*)'FARKLI DEGER SAYISI=IF=',IF
      DO 243 I=1,IF
      DO 12 II=1,KIS
      WRITE(*,*)'KJF('I,','II,')=',KJF(I,II)
      12 WRITE(6,*)'KJF('I,','II,')=',KJF(I,II)
      WRITE(*,*)'OBJ('I,')=',OBJ(I)

```

```

WRITE(6,*)'OBF('I,')='OBF(I)
WRITE(*,*)
243 WRITE(6,*)
WRITE(6,*)'SUBF SONUCLARI SONU'
RETURN
END
C*****
SUBROUTINE SIRALA(N,S,KJB,KIS,KJB1,KJB2)
DIMENSION KJB1(1,1),KJB2(1,1)
&,S(N),KJB(N,KIS)
WRITE(*,*)'SIRALIYOR'
100 DO 10 I=1,N-1
IF(S(I).LE.S(I+1))GO TO 11
B1=S(I+1)
B2=S(I)
DO 74 IJ=1,KIS
KJB1(I,IJ)=KJB(I+1,IJ)
74 KJB2(I,IJ)=KJB(I,IJ)
S(I)=B1
S(I+1)=B2
DO 78 IJ=1,KIS
KJB(I,IJ)=KJB1(I,IJ)
78 KJB(I+1,IJ)=KJB2(I,IJ)
GO TO 10
11 CONTINUE
DO 75 II=1,KIS
75 KJB(I,II)=KJB(I,II)
10 CONTINUE
DO 20 I=1,N-1
20 IF(S(I).GT.S(I+1))GO TO 100
WRITE(*,*)'KUCUKTEN BUYUGE SIRALANMIS SAYILAR'
DO 31 I=1,N
DO 32 J=1,KIS
32 WRITE(*,*)'KJB('I,','J,')='KJB(I,J)
31 WRITE(*,*)'OBJ('I,')='S(I)
RETURN
END
C*****
SUBROUTINE FARKLIF(N,S,OBF,KIS,KJB,KJF,IF)
DIMENSION S(1),OBF(1),KJB(N,KIS),KJF(1024,KIS)
C WRITE(*,*)'EPS=?'
C READ(*,*)EPS
EPS=0.1
IF=1
OBF(IF)=S(IF)
DO 44 I=1,KIS
44 KJF(IF,I)=KJB(IF,I)
DO 21 I=1,N
21 IF(S(I).LT.1.E-30)S(I)=1.E-30
DO 1 I=2,N
FARK=ABS(S(I)-S(I-1))
IF(FARK.LE.EPS)GO TO 1
IF=IF+1
OBF(IF)=S(I)
DO 45 I5=1,KIS
45 KJF(IF,I5)=KJB(I,I5)
1 CONTINUE
RETURN
END
C*****
SUBROUTINE SIRALA2(IF,OBB)
DIMENSION OBB(1)
WRITE(*,*)IF,'TANE OPT SONUCU BULUNAN OBB LARI SIRALIYOR'
WRITE(6,*)IF,'TANE OPT SONUCU BULUNAN OBB LARI SIRALIYOR'
100 DO 10 I=1,IF-1
IF(OBB(I).LE.OBB(I+1))GO TO 10
B1=OBB(I+1)
B2=OBB(I)
OBB(I)=B1
OBB(I+1)=B2
10 CONTINUE
DO 20 I=1,IF-1
20 IF(OBB(I).GT.OBB(I+1))GO TO 100
WRITE(*,*)'KUCUKTEN BUYUGE SIRALANMIS SAYILAR'
DO 31 I=1,IF
WRITE(6,*)'OBB('I,')='OBB(I)
31 WRITE(*,*)'OBB('I,')='OBB(I)
RETURN
END

```

EK-3 GENKDI BILGISAYAR PROGRAMININ LISTESİ

```

DIMENSION KY(90),ZREF(90),ZYF(90),A(90),B(90),AB(90),ABN(90)
&,ZREFE(90),TETA(90)
INTEGER AR(90),ARM(90),ARG(90)
OPEN(6,FILE='GENKDI.YAZ',STATUS='OLD')
CALL GENKDI(KY,ZREF,ZYF,A,B,AB,ABN,AR,ARM,ARG,ZREFE,TETA)
STOP
END
-----
SUBROUTINE GENKDI(KY,ZREF,ZYF,A,B,AB,ABN,AR,ARM,ARG,ZREFE,TETA)
DIMENSION KY(1),ZREF(1),ZYF(1),A(1),B(1),AB(1),ABN(1),ZREFE(1)
&,TETA(1)
INTEGER AR(1),ARM(1),ARG(1),ARS,ARK
C NMAX=9
SAY=0
CAKS=0
OBJMAX=-1.E20
KNT=0
MF=2
AKMBS=1
CALL GIR(KSE,N,NY,M,ZREF,LCP,AR,LY,K1,ZREFE)
KIS=4*N+(KSE/2)**2
ARS=NY*KIS
ARK=ARS/4+1
IF(KSE.EQ.2)AR(N+1)=ARK
CALL YAZG(KSE,N,NY,M,ZREF,LCP,KIS,ARS,LY,K1,ZREFE)
122 CONTINUE
CALL TAMAR(N,NY,AR,KSE)
IF(LCP.EQ.0)GO TO 435
CALL CAK(KIS,NY,AR,KY,KC)
IF(KC.EQ.1)GO TO 38
CAKS=CAKS+1
435 CONTINUE
CALL ARORT(KIS,NY,AR,TETA)
CALL GH(NY,M,KIS,TETA,A,B,AB,MF)
CALL NORMZ(M,AB,ABN,ZYF,MF)
DO 67 I=3,M,2
67 IF(ZYF(I).LT.ZREF(I))GO TO 999
SAY=SAY+1
OBJ=OBJECT(M,ZREF,ZYF,AB,MF)
IF(OBJ.LT.OBJMAX)GO TO 999
C IF(OBJ.GT.OBJMAX)OBJMAX=OBJ
OBJMAX=OBJ
DO 2001 I=1,KIS
2001 ARM(I)=AR(I)
IF(LY.EQ.0)GO TO 474
CALL YAZH(KIS,AR,M,AB,ABN,OBJ,ZYF,LCP,MF)
474 CONTINUE
GO TO 999
38 CONTINUE
999 CALL DEGER(N,AR,KNT,ARK)
IF(KNT.EQ.1)GO TO 909
AKMBS=AKMBS+1
GO TO 122
909 CONTINUE
IF(SAY.NE.0)GO TO 889
CALL YAZI(AKMBS,CAKS,SAY,OBJMAX,LCP)
WRITE(*,*)'COZUM BULUNAMADI'
WRITE(6,*)'COZUM BULUNAMADI'
GO TO 1222
889 CONTINUE
WRITE(*,*)'SONUC'
WRITE(6,*)'SONUC'
CALL YAZI(AKMBS,CAKS,SAY,OBJMAX,LCP)
CALL ARORT(KIS,NY,ARM,TETA)
CALL GH(NY,M,KIS,TETA,A,B,AB,MF)
CALL NORMZ(M,AB,ABN,ZYF,MF)
OBJ=OBJECT(M,ZREF,ZYF,AB,MF)
CALL YAZH(KIS,ARM,M,AB,ABN,OBJ,ZYF,LCP,MF)
IF(K1.EQ.0)GO TO 1222
MF=1
CALL ETRR(N,KSE,M,ZREFE,K1,NY,ARG,AR,ARM,A,B,AB,ABN,ZYF,KY,LY,
&TETA,MF)
1222 RETURN
END
-----
SUBROUTINE TAMAR(N,NY,AR,KSE)
INTEGER AR(1),ARK,ARS
KIS=4*N+(KSE/2)**2
ARS=NY*KIS
ARK=ARS/4+1
IF(KSE.EQ.2)AR(N+1)=ARK
IF(KSE.EQ.1)KH1=1
IF(KSE.EQ.2)KH1=0
DO 1 I=N+KSE/2+1,KIS/2
1 AR(I)=ARK-KH1+(ARK-AR(KIS/2+1-I))
DO 2 I=KIS/2+1,KIS
2 AR(I)=AR(I-KIS/2)+ARS/2
RETURN
END
-----
SUBROUTINE CAK(KIS,NY,AR,KY,KC)
DIMENSION KY(1)

```

```

INTEGER AR(1),ARS
ARS=NY*KIS
DO 3 I3=1,NY-1
DO 1 I1=1,KIS
KY(I1)=AR(I1)+I3*(ARS/NY)
IF(KY(I1).GT.ARS)KY(I1)=KY(I1)-ARS
DO 2 I2=1,KIS
IF(AR(I2).EQ.KY(I1))GO TO 33
2 CONTINUE
GO TO 1
33 KC=1
C KC=1 CAKISMA VAR
GO TO 999
1 CONTINUE
3 CONTINUE
KC=0
C KC=0 CAKISMA YOK
999 RETURN
END
C-----
SUBROUTINE ARORT(KIS,NY,AR,TETA)
DIMENSION TETA(1)
INTEGER AR(1)
ARS=NY*KIS
PI=3.141592
GEN=2*PI/ARS
DO 1 I=1,KIS
1 TETA(I)=AR(I)*GEN-GEN/2
RETURN
END
C-----
SUBROUTINE GH(NY,M,KIS,TETA,A,B,AB,MF)
DIMENSION TETA(1),A(1),B(1),AB(1)
NS=KIS*NY
C A COS.LU TERIMLERIN KATSAYILARI
C B SIN.LU TERIMLERIN KATSAYILARI
C AB(I)=SQRT(A(I)**2+B(I)**2)
DO 10 I=1,M,MF
A(I)=0.
B(I)=0.
DO 11 J=1,KIS
KJ=1
IF(J.GT.KIS/2)KJ=-1
A(I)=A(I)+(2./NS)*KJ*COS(I*TETA(J))
11 B(I)=B(I)+(2./NS)*KJ*SIN(I*TETA(J))
10 AB(I)=SQRT(A(I)**2+B(I)**2)
RETURN
END
C-----
SUBROUTINE NORMZ(M,AB,ABN,ZYF,MF)
DIMENSION AB(1),ABN(1),ZYF(1)
ABMAX=0.0
DO 1 I=1,M,MF
IF(ABS(AB(I)).LT.1.E-35)AB(I)=1.E-35
1 IF(ABS(AB(I)).GT.ABMAX)ABMAX=ABS(AB(I))
DO 2 I=1,M,MF
ABN(I)=ABS(AB(I))/ABMAX
2 ZYF(I)=-20*ALOG10(ABN(I))
RETURN
END
C-----
FUNCTION OBJECT(M,ZREF,ZYF,AB,MF)
DIMENSION ZREF(1),ZYF(1),AB(1)
OBJECT=AB(1)
MFB=3
IF(MF.EQ.1)MFB=2
DO 2 I=MFB,M,MF
KZ=1
IF(ZYF(I).GT.ZREF(I))KZ=0
2 OBJECT=OBJECT-ABS(AB(I))*KZ
RETURN
END
C-----
C N=9
SUBROUTINE DEGER(N,AR,KNT,ARK)
INTEGER AR(1),ARK
IF(N.EQ.0)GO TO 5
AR(N)=AR(N)+1
IF(AR(N).LE.ARK-1)RETURN
IF(N.EQ.1)GO TO 5
AR(N-1)=AR(N-1)+1
AR(N)=AR(N-1)+1
IF(AR(N-1).LE.ARK-2)RETURN
IF(N.EQ.2)GO TO 5
AR(N-2)=AR(N-2)+1
AR(N-1)=AR(N-2)+1
AR(N)=AR(N-1)+1
IF(AR(N-2).LE.ARK-3)RETURN
IF(N.EQ.3)GO TO 5
AR(N-3)=AR(N-3)+1
AR(N-2)=AR(N-3)+1
AR(N-1)=AR(N-2)+1
AR(N)=AR(N-1)+1

```



```

IF(AR(N-3).LE.ARK-4)RETURN
IF(N.EQ.4)GO TO 5
AR(N-4)=AR(N-4)+1
AR(N-3)=AR(N-4)+1
AR(N-2)=AR(N-3)+1
AR(N-1)=AR(N-2)+1
AR(N)=AR(N-1)+1
IF(AR(N-4).LE.ARK-5)RETURN
IF(N.EQ.5)GO TO 5
AR(N-5)=AR(N-5)+1
AR(N-4)=AR(N-5)+1
AR(N-3)=AR(N-4)+1
AR(N-2)=AR(N-3)+1
AR(N-1)=AR(N-2)+1
AR(N)=AR(N-1)+1
IF(AR(N-5).LE.ARK-6)RETURN
IF(N.EQ.6)GO TO 5
AR(N-6)=AR(N-6)+1
AR(N-5)=AR(N-6)+1
AR(N-4)=AR(N-5)+1
AR(N-3)=AR(N-4)+1
AR(N-2)=AR(N-3)+1
AR(N-1)=AR(N-2)+1
AR(N)=AR(N-1)+1
IF(AR(N-6).LE.ARK-7)RETURN
IF(N.EQ.7)GO TO 5
AR(N-7)=AR(N-7)+1
AR(N-6)=AR(N-7)+1
AR(N-5)=AR(N-6)+1
AR(N-4)=AR(N-5)+1
AR(N-3)=AR(N-4)+1
AR(N-2)=AR(N-3)+1
AR(N-1)=AR(N-2)+1
AR(N)=AR(N-1)+1
IF(AR(N-7).LE.ARK-8)RETURN
IF(N.EQ.8)GO TO 5
AR(N-8)=AR(N-8)+1
AR(N-7)=AR(N-8)+1
AR(N-6)=AR(N-7)+1
AR(N-5)=AR(N-6)+1
AR(N-4)=AR(N-5)+1
AR(N-3)=AR(N-4)+1
AR(N-2)=AR(N-3)+1
AR(N-1)=AR(N-2)+1
AR(N)=AR(N-1)+1
IF(AR(N-8).LE.ARK-9)RETURN
IF(N.EQ.9)GO TO 5
5 KNT=1
RETURN
END
C-----
SUBROUTINE GIR(KSE,N,NY,M,ZREF,LCP,AR,LY,K1,ZREFE)
DIMENSION ZREF(1),ZREFE(1)
INTEGER AR(1)
WRITE(*,*)'DUZEN1 PI/2 DE ARALIK YOK'
WRITE(*,*)'DUZEN2 PI/2 DE ARALIK VAR'
WRITE(*,*)'DUZEN1=1, DUZEN2=2 ??'
READ(*,*)KSE
WRITE(*,*)'N DEGERINI GIR MAX 9'
READ(*,*)N
C NMAX=9
IF(N.GT.9)GO TO 43
GO TO 44
43 WRITE(*,*)'1/4 PERYOTTAKI PULS SAYISI 9 DAN BUYUK OLAMAZ'
RETURN
44 WRITE(*,*)'YOL SAYISI NY DAIMA TEK SAYI 1,3,5...GIR '
READ(*,*)NY
WRITE(*,*)'M DEGERINI GIR'
READ(*,*)M
DO 842 I=3,M,2
WRITE(*,*)ZREF('I,')=? DB'
842 READ(*,*)ZREF(I)
WRITE(*,*)'CAK.KONT.YAP=1,YAPMA=0'
READ(*,*)LCP
DO 91 I=1,N
91 AR(I)=I
WRITE(*,*)'ARA SONUCLARI YAZ=1 YAZMA=0'
READ(*,*)LY
WRITE(*,*)'BULUNAN COZUM ETRAFI BUTUN HARMONIKLER ICIN TARANACAK'
WRITE(*,*)'K1 DEGERINI GIRINIZ 2,4,6..'
WRITE(*,*)'ISTENMIYORSA 0 GIRINIZ'
READ(*,*)K1
IF(K1.EQ.0)GO TO 89
DO 88 I=2,M
WRITE(*,*)ZREFE('I,')=?'
88 READ(*,*)ZREFE(I)
89 RETURN
END
C-----
SUBROUTINE YAZG(KSE,N,NY,M,ZREF,LCP,KIS,ARS,LY,K1,ZREFE)
DIMENSION ZREF(1),ZREFE(1)
INTEGER ARS
WRITE(6,86)KSE

```



```

WRITE(*,86)KSE
86 FORMAT(' DUZEN',I2,' ICIN GENKDI CIKTISI')
WRITE(6,87)N
WRITE(*,87)N
87 FORMAT(' CEYREK PERYOTTAKI PULS SAYISI=',I3)
WRITE(6,88)KIS
WRITE(*,88)KIS
88 FORMAT(' BIR PERYOTTAKI PULS SAYISI  =',I3)
WRITE(6,89)NY
WRITE(*,89)NY
89 FORMAT(' YOL SAYISI                =',I3)
WRITE(6,90)ARS
WRITE(*,90)ARS
90 FORMAT(' BIR PERYOTTAKI ARALIK SAYISI =',I3)
DO 392 I=3,M,2
WRITE(*,21)I,ZREF(I)
392 WRITE(6,21)I,ZREF(I)
21 FORMAT(' ZREF(',I2,')                =',F5.2)
IF(K1.EQ.0)WRITE(*,*)'COZUM ETRAFI TARANMAYACAK'
IF(K1.EQ.0)WRITE(6,*)'COZUM ETRAFI TARANMAYACAK'
IF(K1.EQ.0)GO TO 179
WRITE(*,66)K1
WRITE(6,66)K1
66 FORMAT(' K1=',I2,' ICIN COZUM ETRAFI TARANACAK')
WRITE(*,*)'TARAMA SIRASINDAKI ZAYIFLAMA REFERANSLARI'
WRITE(6,*)'TARAMA SIRASINDAKI ZAYIFLAMA REFERANSLARI'
DO 393 I=2,M
WRITE(*,121)I,ZREFE(I)
393 WRITE(6,121)I,ZREFE(I)
121 FORMAT(' ZREFE(',I2,')                =',F5.2)
179 IF(LY.EQ.0)WRITE(*,*)'ARA SONUCLAR YAZILMAYACAK'
IF(LY.EQ.0)WRITE(6,*)'ARA SONUCLAR YAZILMAYACAK'
IF(LY.EQ.1)WRITE(*,*)'ARA SONUCLAR YAZILACAK'
IF(LY.EQ.1)WRITE(6,*)'ARA SONUCLAR YAZILACAK'
IF(LCP.EQ.1)WRITE(6,48)
IF(LCP.EQ.1)WRITE(*,48)
IF(LCP.EQ.0)WRITE(6,49)
IF(LCP.EQ.0)WRITE(*,49)
48 FORMAT(' CAKISMA KONTROLU YAPILACAK.....')
49 FORMAT(' CAKISMA KONTROLU YAPILMAYACAK.....')
RETURN
END
C*****
SUBROUTINE YAZH(KIS,AR,M,AB,ABN,OBJ,ZYF,LCP,MF)
DIMENSION AB(1),ABN(1),ZYF(1)
INTEGER AR(1)
DO 846 I=1,KIS
WRITE(*,1000)I,AR(I)
846 WRITE(6,1000)I,AR(I)
1000 FORMAT(' AR(',I3,') =',I3)
DO 735 I=1,M,MF
WRITE(*,1001)I,AB(I),I,ABN(I)
735 WRITE(6,1001)I,AB(I),I,ABN(I)
1001 FORMAT(' AB(',I3,') =',F10.7,'.....ABN(',I3,')=',F10.7)
DO 731 I=1,M,MF
WRITE(*,1004)I,ZYF(I)
731 WRITE(6,1004)I,ZYF(I)
1004 FORMAT(' ZYF(',I3,')=',F7.3,' DB')
WRITE(*,1002)OBJ
WRITE(6,1002)OBJ
1002 FORMAT(' OBJ =',F10.7)
IF(LCP.EQ.0)GO TO 687
WRITE(6,*)'CAKISMA YOK'
WRITE(*,*)'CAKISMA YOK'
GO TO 643
687 WRITE(*,*)'CAK.KONT.YAPILMADI'
WRITE(6,*)'CAK.KONT.YAPILMADI'
643 WRITE(*,*)'.....'
WRITE(6,*)'.....'
RETURN
END
C*****
SUBROUTINE YAZI(AKMBS,CAKS,SAY,OBJMAX,LCP)
WRITE(6,1005)AKMBS
WRITE(*,1005)AKMBS
1005 FORMAT(' DENENEN KOMBINASYON SAYISI AKMBS                =',E10.5)
IF(LCP.EQ.0)GO TO 607
WRITE(*,1006)CAKS
WRITE(6,1006)CAKS
1006 FORMAT(' CAKISMA OLMAYAN KOMBINASYON SAYISI CAKS        =',E10.5)
WRITE(6,1007)SAY
WRITE(*,1007)SAY
1007 FORMAT(' CAKISMA OLMAYAN VE ZREFI SAGLAYAN KMB SAYISI SAY='E10.5)
GO TO 1999
607 WRITE(6,1008)SAY
WRITE(*,1008)SAY
1008 FORMAT(' CAK.KONT.YAPILMADAN ZREFI SAGLAYAN KOMB.SAY. SAY='E10.5)
1999 WRITE(6,1009)OBJMAX
WRITE(*,1009)OBJMAX
1009 FORMAT(' OBJMAX=',F10.7)
RETURN
END
C*****

```

```

SUBROUTINE YAZIE(K1,AKMBG,AKMBS,AKCY,SAY,AKMBF)
WRITE(*,*)'ETRR SONUCLARI'
WRITE(6,*)'ETRR SONUCLARI'
WRITE(*,2340)AKMBG,AKCY,SAY
WRITE(6,2340)AKMBG,AKCY,SAY
2340 FORMAT(' DENENEN',E10.5,' KOMB.DAN',E10.5,' UNDE CAK.YOK BUNUN',
&/,E10.5,' U ZREFE SARTINI SAGLIYOR')
WRITE(*,*)'CAK YOK AKCY=',AKCY,'GECRL AKMBG=',AKMBG,' DEN AKMBS=',
&AKMBS
WRITE(6,*)'CAK YOK AKCY=',AKCY,'GECRL AKMBG=',AKMBG,' DEN AKMBS=',
&AKMBS
C AKMBF=(K1+1)**KIS FORMULUNDEN HESAPLANAN KMBS
WRITE(*,*)'AKMBF=(K1+1)**KIS FORMULUNDEN HESAPLANAN AKMBS=',AKMBF
WRITE(6,*)'AKMBF=(K1+1)**KIS FORMULUNDEN HESAPLANAN AKMBS=',AKMBF
RETURN
END
C-----
SUBROUTINE ETRR(N,KSE,M,ZREF,K1,NY,ARG,AR,ARM,A,B,AB,ABN,ZYF,KY
&,LY,TETA,MF)
DIMENSION A(1),B(1),AB(1),ABN(1),ZREF(1),ZYF(1),KY(1),TETA(1)
INTEGER AR(1),ARG(1),ARS,ARM(1)
KIS=4*N+(KSE/2)**2
LCP=1
CALL TAMAR(N,NY,ARM,KSE)
DO 400 I=1,KIS
400 ARG(I)=ARM(I)
SAY=0
ARS=NY*KIS
OBJMAX=-.1E20
PI=3.141592
U=2*PI/ARS
AKMBG=0
AKCY=0
AKMBS=1
DO 511 I=1,KIS
511 ARG(I)=ARG(I)-K1/2
DO 512 I=1,KIS
512 AR(I)=ARG(I)
122 CONTINUE
IF(MF.EQ.1)GO TO 51
CALL TAMAR(N,NY,AR,KSE)
51 CONTINUE
IF(AR(1).LE.0.OR.AR(KIS).GT.ARS)GO TO 123
DO 199 I=1,KIS-1
199 IF(AR(I+1).LE.AR(I))GO TO 123
AKMBG=AKMBG+1
CALL CAK(KIS,NY,AR,KY,KC)
IF(KC.EQ.1)GO TO 123
AKCY=AKCY+1
CALL ARORT(KIS,NY,AR,TETA)
CALL GH(NY,M,KIS,TETA,A,B,AB,MF)
CALL NORMZ(M,AB,ABN,ZYF,MF)
DO 459 I=2,M
459 IF(ZYF(I).LT.ZREF(I))GO TO 123
OBJ=OBJECT(M,ZREF,ZYF,AB,MF)
SAY=SAY+1
IF(LY.EQ.0)GO TO 1799
WRITE(*,*)' ETRR ARA SONUCLAR'
WRITE(6,*)' ETRR ARA SONUCLAR'
CALL YAZH(KIS,AR,M,AB,ABN,OBJ,ZYF,LCP,MF)
1799 CONTINUE
C IF(OBJ.LT.OBJMAX)GO TO 123
OBJMAX=OBJ
DO 575 I=1,KIS
575 ARM(I)=AR(I)
123 KK=N
IF(MF.EQ.1)KK=KIS
CALL DGETR(KK,AR,ARG,K1,L)
IF(L.EQ.0)GO TO 999
AKMBS=AKMBS+1
GO TO 122
999 CONTINUE
IF(SAY.NE.0)GO TO 888
AKMBF=(K1+1)**KK
CALL YAZIE(K1,AKMBG,AKMBS,AKCY,SAY,AKMBF)
WRITE(*,*)'COZUM BULUNAMADI'
WRITE(6,*)'COZUM BULUNAMADI'
GO TO 1111
888 CONTINUE
C AKMBF=(K1+1)**KK FORMULUNDEN HESAPLANAN KMBS
AKMBF=(K1+1)**KK
CALL ARORT(KIS,NY,ARM,TETA)
CALL GH(NY,M,KIS,TETA,A,B,AB,MF)
CALL NORMZ(M,AB,ABN,ZYF,MF)
OBJ=OBJECT(M,ZREF,ZYF,AB,MF)
CALL YAZIE(K1,AKMBG,AKMBS,AKCY,SAY,AKMBF)
CALL YAZH(KIS,ARM,M,AB,ABN,OBJ,ZYF,LCP,MF)
1111 RETURN
END
C-----
SUBROUTINE DGETR(N,AR,ARG,K1,L)
INTEGER AR(1),ARG(1)
L=0

```

```

IF(N.EQ.0)GO TO 666
AR(N)=AR(N)+1
IF(AR(N).LE.ARG(N)+K1)GO TO 122
IF(N.EQ.1)GO TO 666
AR(N-1)=AR(N-1)+1
AR(N)=ARG(N)
IF(AR(N-1).LE.ARG(N-1)+K1)GO TO 122
IF(N.EQ.2)GO TO 666
AR(N-2)=AR(N-2)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
IF(AR(N-2).LE.ARG(N-2)+K1)GO TO 122
IF(N.EQ.3)GO TO 666
AR(N-3)=AR(N-3)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
IF(AR(N-3).LE.ARG(N-3)+K1)GO TO 122
IF(N.EQ.4)GO TO 666
AR(N-4)=AR(N-4)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
IF(AR(N-4).LE.ARG(N-4)+K1)GO TO 122
IF(N.EQ.5)GO TO 666
AR(N-5)=AR(N-5)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
IF(AR(N-5).LE.ARG(N-5)+K1)GO TO 122
IF(N.EQ.6)GO TO 666
AR(N-6)=AR(N-6)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
IF(AR(N-6).LE.ARG(N-6)+K1)GO TO 122
IF(N.EQ.7)GO TO 666
AR(N-7)=AR(N-7)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
IF(AR(N-7).LE.ARG(N-7)+K1)GO TO 122
IF(N.EQ.8)GO TO 666
AR(N-8)=AR(N-8)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
IF(AR(N-8).LE.ARG(N-8)+K1)GO TO 122
IF(N.EQ.9)GO TO 666
AR(N-9)=AR(N-9)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
IF(AR(N-9).LE.ARG(N-9)+K1)GO TO 122
IF(N.EQ.10)GO TO 666
AR(N-10)=AR(N-10)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
IF(AR(N-10).LE.ARG(N-10)+K1)GO TO 122
IF(N.EQ.11)GO TO 666
AR(N-11)=AR(N-11)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)

```

```

AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
IF(AR(N-11).LE.ARG(N-11)+K1)GO TO 122
IF(N.EQ.12)GO TO 666
AR(N-12)=AR(N-12)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
IF(AR(N-12).LE.ARG(N-12)+K1)GO TO 122
IF(N.EQ.13)GO TO 666
AR(N-13)=AR(N-13)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
IF(AR(N-13).LE.ARG(N-13)+K1)GO TO 122
IF(N.EQ.14)GO TO 666
AR(N-14)=AR(N-14)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
IF(AR(N-14).LE.ARG(N-14)+K1)GO TO 122
IF(N.EQ.15)GO TO 666
AR(N-15)=AR(N-15)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
IF(AR(N-15).LE.ARG(N-15)+K1)GO TO 122
IF(N.EQ.16)GO TO 666
AR(N-16)=AR(N-16)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)

```

```

IF(AR(N-16).LE.ARG(N-16)+K1)GO TO 122
IF(N.EQ.17)GO TO 666
AR(N-17)=AR(N-17)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
IF(AR(N-17).LE.ARG(N-17)+K1)GO TO 122
IF(N.EQ.18)GO TO 666
AR(N-18)=AR(N-18)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
IF(AR(N-18).LE.ARG(N-18)+K1)GO TO 122
IF(N.EQ.19)GO TO 666
AR(N-19)=AR(N-19)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
IF(AR(N-19).LE.ARG(N-19)+K1)GO TO 122
IF(N.EQ.20)GO TO 666
AR(N-20)=AR(N-20)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
AR(N-19)=ARG(N-19)
IF(AR(N-20).LE.ARG(N-20)+K1)GO TO 122
IF(N.EQ.21)GO TO 666
AR(N-21)=AR(N-21)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)

```

```
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
AR(N-19)=ARG(N-19)
AR(N-20)=ARG(N-20)
IF(AR(N-21),LE.ARG(N-21)+K1)GO TO 122
IF(N.EQ.22)GO TO 666
AR(N-22)=AR(N-22)+1
AR(N)=ARG(N)
AR(N-1)=ARG(N-1)
AR(N-2)=ARG(N-2)
AR(N-3)=ARG(N-3)
AR(N-4)=ARG(N-4)
AR(N-5)=ARG(N-5)
AR(N-6)=ARG(N-6)
AR(N-7)=ARG(N-7)
AR(N-8)=ARG(N-8)
AR(N-9)=ARG(N-9)
AR(N-10)=ARG(N-10)
AR(N-11)=ARG(N-11)
AR(N-12)=ARG(N-12)
AR(N-13)=ARG(N-13)
AR(N-14)=ARG(N-14)
AR(N-15)=ARG(N-15)
AR(N-16)=ARG(N-16)
AR(N-17)=ARG(N-17)
AR(N-18)=ARG(N-18)
AR(N-19)=ARG(N-19)
AR(N-20)=ARG(N-20)
AR(N-21)=ARG(N-21)
IF(AR(N-22),LE.ARG(N-22)+K1)GO TO 122
IF(N.EQ.23)GO TO 666
122 L=1
666 RETURN
END
```

EK-4N-YOLLU FİLTRENİN TRANSFER FONKSİYONUNUN HESAPLANABİLME İÇİN MAINLN BİLGİSAYAR PROGRAMINDA KULLANICININ YAZMASI GEREKEN ALTPROGRAMLAR

```

C N-YOLLU SUZGECİN TRANSFER FONKSİYONUNU MAINLN LE HESAPLAMAK İÇİN
C KULLANICININ YAZMASI GEREKEN ALTPROGRAMLAR
  DIMENSION KV(900),RF(900),FRQV(900),NHV(900),TIMEV(900),XO(900),
  *,A(900),B(900),C(900),D(900),F(900),G(900),FIC(900),FI(900),EN(900
  *),EM(900),O(900),P(900),T(900),ETA(900),Q(900),R(900),S(900),V(900
  *),V(900),IQ(900),IR(900),OPQ(900),ON(900)
  COMPLEX FRQV(900),API(900),BB(900),TRNSF(900),H(900),EJ(900)
  *,EP(900),ER(900),APIB(900),CO(900),CP(900),CQ(900),CR(900),CB1(900
  *),CB2(900)
  OPEN(6,FILE='ZAIŞ2.YAZ',STATUS='OLD')
  OPEN(7,FILE='ZAIŞ2.WKS',STATUS='OLD')
  CALL MAINLN(KV,RF,FRQV,NHV,TIMEV,XO,T,ETA,A,B,C,D,F,G,FIC,FI,EN,
  *EM,O,P,Q,R,S,V,IQ,IR,OPQ,FRQV,API,BB,TRNSF,H,EJ,EP,ER,APIB,CO,CP,
  *CQ,CR,CB1,CB2,ON)
  STOP
  END

```

```

C-----
SUBROUTINE SINPT(TO,T,A,B,C,D,F,G,H,K,TT,DT,IYAZ,KAR)
DIMENSION A(1),B(1),C(1),D(1),F(1),G(1),T(1),KAR(1)
WRITE(*,*)'HZ OLARAK ANAHTARLAMA FREKANSI=?'
READ(*,*)FO
TO=1./FO
WRITE(*,*)'BİR PERİYOTTA Kİ PULS SAYISI=KİS=?'
READ(*,*)KİS
DO 1 I=1,KİS
  WRITE(*,*)I,' İNCİ PULSİN BULUNDUGU ARALIK=?'
1 READ(*,*)KAR(I)
  WRITE(*,*)'YOL SAYISI=NY=?'
  READ(*,*)NY
  WRITE(7,*)'KİS=',KİS
  WRITE(7,*)'NY=',NY
  DO 2 I=1,KİS
21 WRITE(7,*)I,' İNCİ PULSİN BULUNDUGU ARALIK=?',KAR(I)
    K=2*KİS+1
    KARS=NY*KİS
    IYAZ=1
    WRITE(*,*)'R=? OHM'
    READ(*,*)R
    WRITE(*,*)'C=? MİKRO FARAD'
    READ(*,*)CC
    CC=CC*1.E-6
    WRITE(7,*)'R=?',R
    WRITE(7,*)'CC=?',CC
    AS=1./(R*CC)
    DS=TO/KARS
    WRITE(7,*)'AS=',AS
    WRITE(7,*)'DS=',DS
    KAR(0)=0
    KAR(KİS+1)=KARS+1
    DO 22 I=0,KİS
    WRITE(7,*)'KAR(',I,')=',KAR(I)
22 WRITE(*,*)'KAR(',I,')=',KAR(I)
    DO 2 I=1,KİS+1
    T(2*I-1)=(KAR(I)-KAR(I-1))*DS
    DO 3 I=1,KİS
    T(2*I)=DS
    DO 30 I=1,KİS
    A(2*I-1)=0.
    B(2*I-1)=0.
30 C(2*I-1)=0.
    DO 31 I=1,KİS
    WRITE(*,*)I,' İNCİ PULSİN İŞARETİ=?'
    READ(*,*)L
    A(2*I)=-AS*ABS(L)
    B(2*I)=AS*L
31 C(2*I)=NY*L
    DO 33 I=1,K
    WRITE(*,*)'A(',I,')=',A(I)
    WRITE(7,*)'A(',I,')=',A(I)
    WRITE(*,*)'B(',I,')=',B(I)
    WRITE(7,*)'B(',I,')=',B(I)
    WRITE(*,*)'C(',I,')=',C(I)
    WRITE(7,*)'C(',I,')=',C(I)
    WRITE(*,*)'T(',I,')=',T(I)
33 WRITE(*,*)'T(',I,')=',T(I)
  RETURN
  END

```

```

C-----
SUBROUTINE SOTPT(NF,F1,F2,IF,RF,FRQV,NH,N1,N2,IH,NHV,NT,T1,T2,
*IT,TIMEV,XO)
DIMENSION FRQV(1),RF(1),NHV(1),TIMEV(1),XO(1)
WRITE(*,*)'F1=? Hz OLARAK'
READ(*,*)F1
IF(F1.EQ.0)F1=1.E-30
WRITE(*,*)'F2=? Hz OLARAK'
READ(*,*)F2
WRITE(*,*)'NF=?'
READ(*,*)NF
N2=0
IF=1
RETURN
END

```

```

C-----
COMPLEX FUNCTION UFUNC(P)
COMPLEX P
UFUNC=(1.,0.)
RETURN
END

```

EK- 5. 6802 MİKROBİLGİSAYARINDA PROGRAM YAZILIMINDA KULLANILAN
BAZI KOMUTLAR VE İŞLEMELERİ İÇİN GEREKEN SÜRELER

op code	Kısaltma Mnemonic	Periyot süresi Cycle time	Adresleme biçimi Adressing mode
86	LDAA	2	immediate
b7	STAA	5	extended
d6	LDAB	3	direct
7E	JMP	3	extended
CE	LDX	3	immediate
A6	LDAA	5	index
08	INX	4	inherent
8C	CPX	3	immediate
26	BNE	4	relative
Bd	JSR	9	extended
4A	DECA	2	inherent
39	RTS	5	inherent

EK-6 MIKRO BILGİSAYAR PROGRAMININ LİSTESİ

```

INTEGER AR(58),KY(31,58),H(1258)
OPEN(6,FILE='MIKRO.YAZ',STATUS='OLD')
WRITE(*,*)'MIKRO BILGISAYARIN HAFIZASINA YAZILACAK KODLARI VERİR'
WRITE(*,*)'DUZEN1 P1/2 DE ARALIK YOK'
WRITE(*,*)'DUZEN2 P1/2 DE ARALIK VAR'
WRITE(*,*)'DUZEN1=1, DUZEN2=2 ??'
READ(*,*)KSE
WRITE(*,*)'N DEGERINI GIR'
READ(*,*)N
WRITE(*,*)'YOL SAYISI NY DAIMA TEK SAYI 1,3,5...GIR'
READ(*,*)NY
DO 91 I=1,N
WRITE(*,*)'AR(I)=?'
91 READ(*,*)AR(I)
CALL MKRBS(N,AR,NY,KY,H,KIS,KSE)
STOP
END
C*****
SUBROUTINE MKRBS(N,AR,NY,KY,H,KIS,KSE)
INTEGER AR(4*N+2),KY(NY,4*N+2),H(NY*(4*N+2)),ARS,ARK
IF(KSE.EQ.1)KSS=0
IF(KSE.EQ.2)KSS=2
IF(KSE.EQ.1)KH1=1
IF(KSE.EQ.2)KH1=0
KIS=4*N+KSS
ARS=NY*KIS
ARK=ARS/4+1
IF(KSE.EQ.1)GO TO 270
AR(N+1)=ARK
270 DO 1 I=N+KSS/2+1,KIS/2
1 AR(I)=ARK-KH1+(ARK-AR(KIS/2+1-I))
DO 2 I=KIS/2+1,KIS
2 AR(I)=(AR(I-KIS/2)+ARS/2)*(-1)
WRITE(6,*)'DUZEN=',KSE
WRITE(*,*)'DUZEN=',KSE
WRITE(6,*)'CEYREK PERYOTTAKI PULS SAYISI=',N
WRITE(*,*)'CEYREK PERYOTTAKI PULS SAYISI=',N
WRITE(6,*)'BIR PERYOTTAKI PULS SAYISI=',KIS
WRITE(*,*)'BIR PERYOTTAKI PULS SAYISI=',KIS
WRITE(*,*)'YOL SAYISI=',NY
WRITE(6,*)'YOL SAYISI=',NY
WRITE(*,*)'BIR PERYOTTAKI ARALIK SAYISI=',ARS
WRITE(6,*)'BIR PERYOTTAKI ARALIK SAYISI=',ARS
WRITE(6,*)
WRITE(*,*)
DO 18 I=1,KIS
WRITE(*,*)I,' INCI PULSIN BULUNDUGU ARALIK= ',AR(I)
18 WRITE(6,*)I,' INCI PULSIN BULUNDUGU ARALIK= ',AR(I)
DO 3 J=1,NY
DO 3 I=1,KIS
KCARP=1
IF(AR(I).LT.0)KCARP=-1
KY(J,I)=ABS(AR(I))+J-1)*KIS
IF(KY(J,I).GT.NY*KIS)KY(J,I)=(KY(J,I)-NY*KIS)
KY(J,I)=KCARP*KY(J,I)
C WRITE(6,*) KY('J','I')=,KY(J,I)
C WRITE(6,*) KY('J','I')=,KY(J,I)
3 CONTINUE
DO 4 J=1,ARS
DO 4 I1=1,NY
DO 4 I2=1,KIS
IF(ABS(KY(I1,I2)).EQ.J.AND.KY(I1,I2).GT.0) H(J)=I1
IF(ABS(KY(I1,I2)).EQ.J.AND.KY(I1,I2).LT.0) H(J)=-I1
4 CONTINUE
DO 5 J=1,ARS
C WRITE(6,*)H('J')=,H(J)
5 WRITE(*,*)H('J')=,H(J)
C IF(NY.LT.7)GO TO 121
IF(NY.GT.7.and.NY.le.15)GO TO 122
IF(NY.GT.15.and.NY.le.31)GO TO 123
WRITE(6,*)'NY(,=)7 OLDUGUNDAN 8 BITLIK MIKROBILG.IN 8 LİK BİR POR
&TU YETERLİDİR'
WRITE(*,*)'1. SUTUN SIRA NO'
WRITE(*,*)'2. SUTUN A PORTUNUN DEGERI'
WRITE(*,*)'3. SUTUN MIKROBLA YAZILACAK İKİLİ KODLANMIS 16TABSAYI'
WRITE(6,*)'1. SUTUN SIRA NO'
WRITE(6,*)'2. SUTUN A PORTUNUN DEGERI'
WRITE(6,*)'3. SUTUN MIKROBLA YAZILACAK İKİLİ KODLANMIS 16TABSAYI'
WRITE(6,*)'BİRİNCİ SUTUN SIRA NO'
WRITE(6,*)'İKİNCİ SUTUN MIKROBILG. A PORTUNUN ALACAGI DEGER'
WRITE(6,*)'UCUNCU SUTUN BU DEGERİ VEREN İKİLİ KODLANMIS 16 TABANLI
& SAYI BU SAYI MIKRO BILG.A GIRILECEK'
DO 6 I=1,ARS
IF(H(I).EQ.-1)WRITE(*,*)I,' 00000001 ;01'
IF(H(I).EQ.-1)WRITE(6,*)I,' 00000001 ;01'
IF(H(I).EQ.1)WRITE(*,*)I,' 10000001 ;81'
IF(H(I).EQ.1)WRITE(6,*)I,' 10000001 ;81'
IF(H(I).EQ.-2)WRITE(*,*)I,' 00000010 ;02'
IF(H(I).EQ.-2)WRITE(6,*)I,' 00000010 ;02'
IF(H(I).EQ.2)WRITE(*,*)I,' 10000010 ;82'
IF(H(I).EQ.2)WRITE(6,*)I,' 10000010 ;82'
IF(H(I).EQ.-3)WRITE(*,*)I,' 00000100 ;04'

```

```

IF(H(I).EQ.-3)WRITE(6,*)I, 00000100 004'
IF(H(I).EQ.3)WRITE(*,*)I, 10000100 084'
IF(H(I).EQ.3)WRITE(6,*)I, 10000100 084'
IF(H(I).EQ.-4)WRITE(*,*)I, 00001000 008'
IF(H(I).EQ.-4)WRITE(6,*)I, 00001000 008'
IF(H(I).EQ.4)WRITE(*,*)I, 10001000 088'
IF(H(I).EQ.4)WRITE(6,*)I, 10001000 088'
IF(H(I).EQ.-5)WRITE(*,*)I, 00010000 010'
IF(H(I).EQ.-5)WRITE(6,*)I, 00010000 010'
IF(H(I).EQ.5)WRITE(*,*)I, 10010000 090'
IF(H(I).EQ.5)WRITE(6,*)I, 10010000 090'
IF(H(I).EQ.-6)WRITE(*,*)I, 00100000 020'
IF(H(I).EQ.-6)WRITE(6,*)I, 00100000 020'
IF(H(I).EQ.6)WRITE(*,*)I, 10100000 0A0'
IF(H(I).EQ.6)WRITE(6,*)I, 10100000 0A0'
IF(H(I).EQ.-7)WRITE(*,*)I, 01000000 040'
IF(H(I).EQ.-7)WRITE(6,*)I, 01000000 040'
IF(H(I).EQ.7)WRITE(*,*)I, 11000000 0C0'
IF(H(I).EQ.7)WRITE(6,*)I, 11000000 0C0'
6 CONTINUE
RETURN
122 WRITE(6,*)'NY 7 ILE 15 ARASINDA 16 BITLIK PORT GEREKIYOR'
WRITE(*,*)'NY 7 ILE 15 ARASINDA 16 BITLIK PORT GEREKIYOR'
WRITE(*,*)'1. SUTUN SIRA NO'
WRITE(*,*)'2. SUTUN A PORTUNUN DEGERI'
WRITE(*,*)'3. SUTUN MIKROBL.A YAZILACAK IKILI KODLANMIS 16TABSAYI'
WRITE(6,*)'1. SUTUN SIRA NO'
WRITE(6,*)'2. SUTUN A PORTUNUN DEGERI'
WRITE(6,*)'3. SUTUN MIKROBL.A YAZILACAK IKILI KODLANMIS 16TABSAYI'
DO 66 I=1,ARS
IF(H(I).EQ.-1)WRITE(*,*)I, 00000000 00000001 0001'
IF(H(I).EQ.-1)WRITE(6,*)I, 00000000 00000001 0001'
IF(H(I).EQ.1)WRITE(*,*)I, 10000000 00000001 8001'
IF(H(I).EQ.1)WRITE(6,*)I, 10000000 00000001 8001'
IF(H(I).EQ.-2)WRITE(*,*)I, 00000000 00000010 0002'
IF(H(I).EQ.-2)WRITE(6,*)I, 00000000 00000010 0002'
IF(H(I).EQ.2)WRITE(*,*)I, 10000000 00000010 8002'
IF(H(I).EQ.2)WRITE(6,*)I, 10000000 00000010 8002'
IF(H(I).EQ.-3)WRITE(*,*)I, 00000000 00000100 0004'
IF(H(I).EQ.-3)WRITE(6,*)I, 00000000 00000100 0004'
IF(H(I).EQ.3)WRITE(*,*)I, 10000000 00000100 8004'
IF(H(I).EQ.3)WRITE(6,*)I, 10000000 00000100 8004'
IF(H(I).EQ.-4)WRITE(*,*)I, 00000000 00001000 0008'
IF(H(I).EQ.-4)WRITE(6,*)I, 00000000 00001000 0008'
IF(H(I).EQ.4)WRITE(*,*)I, 10000000 00001000 8008'
IF(H(I).EQ.4)WRITE(6,*)I, 10000000 00001000 8008'
IF(H(I).EQ.-5)WRITE(*,*)I, 00000000 00010000 0010'
IF(H(I).EQ.-5)WRITE(6,*)I, 00000000 00010000 0010'
IF(H(I).EQ.5)WRITE(*,*)I, 10000000 00010000 8010'
IF(H(I).EQ.5)WRITE(6,*)I, 10000000 00010000 8010'
IF(H(I).EQ.-6)WRITE(*,*)I, 00000000 00100000 0020'
IF(H(I).EQ.-6)WRITE(6,*)I, 00000000 00100000 0020'
IF(H(I).EQ.6)WRITE(*,*)I, 10000000 00100000 8020'
IF(H(I).EQ.6)WRITE(6,*)I, 10000000 00100000 8020'
IF(H(I).EQ.-7)WRITE(*,*)I, 00000000 01000000 0040'
IF(H(I).EQ.-7)WRITE(6,*)I, 00000000 01000000 0040'
IF(H(I).EQ.7)WRITE(*,*)I, 10000000 01000000 8040'
IF(H(I).EQ.7)WRITE(6,*)I, 10000000 01000000 8040'
IF(H(I).EQ.-8)WRITE(*,*)I, 00000000 10000000 0080'
IF(H(I).EQ.-8)WRITE(6,*)I, 00000000 10000000 0080'
IF(H(I).EQ.8)WRITE(*,*)I, 10000000 10000000 8080'
IF(H(I).EQ.8)WRITE(6,*)I, 10000000 10000000 8080'
IF(H(I).EQ.-9)WRITE(*,*)I, 00000001 00000000 0100'
IF(H(I).EQ.-9)WRITE(6,*)I, 00000001 00000000 0100'
IF(H(I).EQ.9)WRITE(*,*)I, 10000001 00000000 8100'
IF(H(I).EQ.9)WRITE(6,*)I, 10000001 00000000 8100'
IF(H(I).EQ.-10)WRITE(*,*)I, 00000010 00000000 0200'
IF(H(I).EQ.-10)WRITE(6,*)I, 00000010 00000000 0200'
IF(H(I).EQ.10)WRITE(*,*)I, 10000010 00000000 8200'
IF(H(I).EQ.10)WRITE(6,*)I, 10000010 00000000 8200'
IF(H(I).EQ.-11)WRITE(*,*)I, 00000100 00000000 0400'
IF(H(I).EQ.-11)WRITE(6,*)I, 00000100 00000000 0400'
IF(H(I).EQ.11)WRITE(*,*)I, 10000100 00000000 8400'
IF(H(I).EQ.11)WRITE(6,*)I, 10000100 00000000 8400'
IF(H(I).EQ.-12)WRITE(*,*)I, 00001000 00000000 0800'
IF(H(I).EQ.-12)WRITE(6,*)I, 00001000 00000000 0800'
IF(H(I).EQ.12)WRITE(*,*)I, 10001000 00000000 8800'
IF(H(I).EQ.12)WRITE(6,*)I, 10001000 00000000 8800'
IF(H(I).EQ.-13)WRITE(*,*)I, 00010000 00000000 1000'
IF(H(I).EQ.-13)WRITE(6,*)I, 00010000 00000000 1000'
IF(H(I).EQ.13)WRITE(*,*)I, 10010000 00000000 9000'
IF(H(I).EQ.13)WRITE(6,*)I, 10010000 00000000 9000'
IF(H(I).EQ.-14)WRITE(*,*)I, 00100000 00000000 2000'
IF(H(I).EQ.-14)WRITE(6,*)I, 00100000 00000000 2000'
IF(H(I).EQ.14)WRITE(*,*)I, 10100000 00000000 A000'
IF(H(I).EQ.14)WRITE(6,*)I, 10100000 00000000 A000'
IF(H(I).EQ.-15)WRITE(*,*)I, 01000000 00000000 4000'
IF(H(I).EQ.-15)WRITE(6,*)I, 01000000 00000000 4000'
IF(H(I).EQ.15)WRITE(*,*)I, 11000000 00000000 C000'
IF(H(I).EQ.15)WRITE(6,*)I, 11000000 00000000 C000'
66 CONTINUE
RETURN
123 WRITE(6,*)'NY 15 ILE 31 ARASINDA 32 BITLIK PORT GEREKIYOR'

```

```

WRITE(*,*)'NY 15 ILE 31 ARASINDA 32 BITLIK PORT GEREKIYOR'
WRITE(*,*)'1. SUTUN SIRA NO'
WRITE(*,*)'2. SUTUN A PORTUNUN DEGERI'
WRITE(*,*)'3. SUTUN MIKROBLA YAZILACAK IKILI KODLANMIS 16TABESAYI'
WRITE(6,*)'1. SUTUN SIRA NO'
WRITE(6,*)'2. SUTUN A PORTUNUN DEGERI'
WRITE(6,*)'3. SUTUN MIKROBLA YAZILACAK IKILI KODLANMIS 16TABESAYI'
DO 666 I=1,ARS
  IF(H(I).EQ.-1)WRITE(6,*) I,';00000000 00000000 00000000 00000001;0
  &000 0001'
  IF(H(I).EQ.-2)WRITE(6,*) I,';00000000 00000000 00000000 00000010;0
  &000 0002'
  IF(H(I).EQ.-3)WRITE(6,*) I,';00000000 00000000 00000000 00000100;0
  &000 0004'
  IF(H(I).EQ.-4)WRITE(6,*) I,';00000000 00000000 00000000 00001000;0
  &000 0008'
  IF(H(I).EQ.-5)WRITE(6,*) I,';00000000 00000000 00000000 00010000;0
  &000 0010'
  IF(H(I).EQ.-6)WRITE(6,*) I,';00000000 00000000 00000000 00100000;0
  &000 0020'
  IF(H(I).EQ.-7)WRITE(6,*) I,';00000000 00000000 00000000 01000000;0
  &000 0040'
  IF(H(I).EQ.-8)WRITE(6,*) I,';00000000 00000000 00000000 10000000;0
  &000 0080'
  IF(H(I).EQ.-9)WRITE(6,*) I,';00000000 00000000 00000000 00000001;0
  &000 0100'
  IF(H(I).EQ.-10)WRITE(6,*) I,';00000000 00000000 00000010 00000000;0
  &000 0200'
  IF(H(I).EQ.-11)WRITE(6,*) I,';00000000 00000000 00000100 00000000;0
  &000 0400'
  IF(H(I).EQ.-12)WRITE(6,*) I,';00000000 00000000 00001000 00000000;0
  &000 0800'
  IF(H(I).EQ.-13)WRITE(6,*) I,';00000000 00000000 00010000 00000000;0
  &000 1000'
  IF(H(I).EQ.-14)WRITE(6,*) I,';00000000 00000000 00100000 00000000;0
  &000 2000'
  IF(H(I).EQ.-15)WRITE(6,*) I,';00000000 00000000 01000000 00000000;0
  &000 4000'
  IF(H(I).EQ.-16)WRITE(6,*) I,';00000000 00000000 10000000 00000000;0
  &000 8000'
  IF(H(I).EQ.-17)WRITE(6,*) I,';00000000 00000001 00000000 00000000;0
  &001 0000'
  IF(H(I).EQ.-18)WRITE(6,*) I,';00000000 00000010 00000000 00000000;0
  &002 0000'
  IF(H(I).EQ.-19)WRITE(6,*) I,';00000000 00000100 00000000 00000000;0
  &004 0000'
  IF(H(I).EQ.-20)WRITE(6,*) I,';00000000 00001000 00000000 00000000;0
  &008 0000'
  IF(H(I).EQ.-21)WRITE(6,*) I,';00000000 00010000 00000000 00000000;0
  &010 0000'
  IF(H(I).EQ.-22)WRITE(6,*) I,';00000000 00100000 00000000 00000000;0
  &020 0000'
  IF(H(I).EQ.-23)WRITE(6,*) I,';00000000 01000000 00000000 00000000;0
  &040 0000'
  IF(H(I).EQ.-24)WRITE(6,*) I,';00000000 10000000 00000000 00000000;0
  &080 0000'
  IF(H(I).EQ.-25)WRITE(6,*) I,';00000001 00000000 00000000 00000000;0
  &100 0000'
  IF(H(I).EQ.-26)WRITE(6,*) I,';00000010 00000000 00000000 00000000;0
  &200 0000'
  IF(H(I).EQ.-27)WRITE(6,*) I,';00000100 00000000 00000000 00000000;0
  &400 0000'
  IF(H(I).EQ.-28)WRITE(6,*) I,';00000100 00000000 00000000 00000000;0
  &800 0000'
  IF(H(I).EQ.-29)WRITE(6,*) I,';00001000 00000000 00000000 00000000;1
  &000 0000'
  IF(H(I).EQ.-30)WRITE(6,*) I,';00100000 00000000 00000000 00000000;2
  &000 0000'
  IF(H(I).EQ.-31)WRITE(6,*) I,';01000000 00000000 00000000 00000000;4
  &000 0000'
  IF(H(I).EQ.1)WRITE(6,*) I,';10000000 00000000 00000000 00000001;8
  &000 0001'
  IF(H(I).EQ.2)WRITE(6,*) I,';10000000 00000000 00000000 00000010;8
  &000 0002'
  IF(H(I).EQ.3)WRITE(6,*) I,';10000000 00000000 00000000 00000100;8
  &000 0004'
  IF(H(I).EQ.4)WRITE(6,*) I,';10000000 00000000 00000000 00001000;8
  &000 0008'
  IF(H(I).EQ.5)WRITE(6,*) I,';10000000 00000000 00000000 00010000;8
  &000 0010'
  IF(H(I).EQ.6)WRITE(6,*) I,';10000000 00000000 00000000 00100000;8
  &000 0020'
  IF(H(I).EQ.7)WRITE(6,*) I,';10000000 00000000 00000000 01000000;8
  &000 0040'
  IF(H(I).EQ.8)WRITE(6,*) I,';10000000 00000000 00000000 10000000;8
  &000 0080'
  IF(H(I).EQ.9)WRITE(6,*) I,';10000000 00000000 00000001 00000000;8
  &000 0100'
  IF(H(I).EQ.10)WRITE(6,*) I,';10000000 00000000 00000010 00000000;8
  &000 0200'
  IF(H(I).EQ.11)WRITE(6,*) I,';10000000 00000000 00000100 00000000;8
  &000 0400'
  IF(H(I).EQ.12)WRITE(6,*) I,';10000000 00000000 00001000 00000000;8

```

```
&000 0800'  
IF(H(I).EQ.13)WRITE(6,*)I, '10000000 00000000 00010000 00000000;8  
&000 1000'  
IF(H(I).EQ.14)WRITE(6,*)I, '10000000 00000000 00100000 00000000;8  
&000 2000'  
IF(H(I).EQ.15)WRITE(6,*)I, '10000000 00000000 01000000 00000000;8  
&000 4000'  
IF(H(I).EQ.16)WRITE(6,*)I, '10000000 00000000 10000000 00000000;8  
&000 8000'  
IF(H(I).EQ.17)WRITE(6,*)I, '10000000 00000001 00000000 00000000;8  
&001 0000'  
IF(H(I).EQ.18)WRITE(6,*)I, '10000000 00000010 00000000 00000000;8  
&002 0000'  
IF(H(I).EQ.19)WRITE(6,*)I, '10000000 00000100 00000000 00000000;8  
&004 0000'  
IF(H(I).EQ.20)WRITE(6,*)I, '10000000 00001000 00000000 00000000;8  
&008 0000'  
IF(H(I).EQ.21)WRITE(6,*)I, '10000000 00010000 00000000 00000000;8  
&010 0000'  
IF(H(I).EQ.22)WRITE(6,*)I, '10000000 00100000 00000000 00000000;8  
&020 0000'  
IF(H(I).EQ.23)WRITE(6,*)I, '10000000 01000000 00000000 00000000;8  
&040 0000'  
IF(H(I).EQ.24)WRITE(6,*)I, '10000000 10000000 00000000 00000000;8  
&080 0000'  
IF(H(I).EQ.25)WRITE(6,*)I, '10000001 00000000 00000000 00000000;8  
&100 0000'  
IF(H(I).EQ.26)WRITE(6,*)I, '10000010 00000000 00000000 00000000;8  
&200 0000'  
IF(H(I).EQ.27)WRITE(6,*)I, '10000100 00000000 00000000 00000000;8  
&400 0000'  
IF(H(I).EQ.28)WRITE(6,*)I, '10001000 00000000 00000000 00000000;8  
&800 0000'  
IF(H(I).EQ.29)WRITE(6,*)I, '10010000 00000000 00000000 00000000;9  
&000 0000'  
IF(H(I).EQ.30)WRITE(6,*)I, '10100000 00000000 00000000 00000000;A  
&000 0000'  
IF(H(I).EQ.31)WRITE(6,*)I, '11000000 00000000 00000000 00000000;C  
&000 0000'  
666 CONTINUE  
RETURN  
END
```

**EK-7 İKİNCİ ARALIK DÜZENİNDE N=2, M=3, NY=7 VE 11, 12, 18, 24, 25, 46, 47, 53, 59, 60
ANAHTAR FONKSİYONU İÇİN MİKRO PROGRAMININ ÇIKTISI**

DUZEN= 2
 CEYREK PERYOTTAKI PULS SAYISI= 2
 BİR PERYOTTAKI PULS SAYISI= 10
 YOL SAYISI= 7
 BİR PERYOTTAKI ARALIK SAYISI= 70
 1 İNCİ PULSİN BULUNDUGU ARALIK= 11
 2 İNCİ PULSİN BULUNDUGU ARALIK= 12
 3 İNCİ PULSİN BULUNDUGU ARALIK= 18
 4 İNCİ PULSİN BULUNDUGU ARALIK= 24
 5 İNCİ PULSİN BULUNDUGU ARALIK= 25
 6 İNCİ PULSİN BULUNDUGU ARALIK= -46
 7 İNCİ PULSİN BULUNDUGU ARALIK= -47
 8 İNCİ PULSİN BULUNDUGU ARALIK= -53
 9 İNCİ PULSİN BULUNDUGU ARALIK= -59
 10 İNCİ PULSİN BULUNDUGU ARALIK= -60

NY(i,=)7 OLDUGUNDAN 8 BITLİK MİKROBİLG.İN 8 LİK BİR PORTU YETERLİDİR

1. SUTUN SIRA NO

2. SUTUN A PORTUNUN DEGERI

3. SUTUN MİKROBLA YAZILACAK İKİLİ KODLANMIS 16TABSAYI

BİRİNCİ SUTUN SIRA NO

İKİNCİ SUTUN MİKROBİLG. A PORTUNUN ALACAGI DEGER

UCUNCU SUTUN BU DEGERI VEREN İKİLİ KODLANMIS 16 TABANLI SAYI BU SAYI MİKRO BİLG.A GIRILECEK

1 = 11000000 = C0
 2 = 11000000 = C0
 3 = 00000100 = 04
 4 = 10100000 = A0
 5 = 10100000 = A0
 6 = 00001000 = 08
 7 = 00001000 = 08
 8 = 11000000 = C0
 9 = 00000100 = 04
 10 = 00000100 = 04
 11 = 10000001 = 81
 12 = 10000001 = 81
 13 = 00001000 = 08
 14 = 11000000 = C0
 15 = 11000000 = C0
 16 = 00010000 = 10
 17 = 00010000 = 10
 18 = 10000001 = 81
 19 = 00001000 = 08
 20 = 00001000 = 08
 21 = 10000010 = 82
 22 = 10000010 = 82
 23 = 00010000 = 10
 24 = 10000001 = 81
 25 = 10000001 = 81
 26 = 00100000 = 20
 27 = 00100000 = 20
 28 = 10000010 = 82
 29 = 00010000 = 10
 30 = 00010000 = 10
 31 = 10000100 = 84
 32 = 10000100 = 84
 33 = 00100000 = 20
 34 = 10000010 = 82
 35 = 10000010 = 82
 36 = 01000000 = 40
 37 = 01000000 = 40
 38 = 10000100 = 84
 39 = 00100000 = 20
 40 = 00100000 = 20
 41 = 10001000 = 88
 42 = 10001000 = 88
 43 = 01000000 = 40
 44 = 10000100 = 84
 45 = 10000100 = 84
 46 = 00000001 = 01
 47 = 00000001 = 01
 48 = 10001000 = 88
 49 = 01000000 = 40
 50 = 01000000 = 40
 51 = 10010000 = 90
 52 = 10010000 = 90
 53 = 00000001 = 01
 54 = 10001000 = 88
 55 = 10001000 = 88
 56 = 00000010 = 02
 57 = 00000010 = 02
 58 = 10010000 = 90
 59 = 00000001 = 01
 60 = 00000001 = 01
 61 = 10100000 = A0
 62 = 10100000 = A0
 63 = 00000010 = 02
 64 = 10010000 = 90
 65 = 10010000 = 90
 66 = 00000100 = 04
 67 = 00000100 = 04
 68 = 10100000 = A0
 69 = 00000010 = 02
 70 = 00000010 = 02

EK-8 BÖLÜM 6'DAKI DEVRELERDE YÜK GERİLİMİ VE AKIMINI HESAPLAMAK İÇİN MAINLN PROGRAMINDA KULLANICININ YAZMASI GEREKEN ALTPROGRAMLAR

```
C BOLUM 6 DA MAINLN LE YUK GERILIMI VEYA AKIMINI HESAPLAMAK ICIN KULL.
C YAZACAGI ALTPROGRAMLAR,DEGISKENLER MINIMUM BOYUTLAR ILE GOSTERILDI
  DIMENSION KV(1),RF(1),FRQV(1),NHV(1),TIMEV(501),XO(1),T(26),ETA(
*26),A(26),B(26),C(26),D(26),F(26),G(26),FIC(26),FI(26),EN(676),E
*M(26),O(1),P(1),Q(1),R(1),S(1),V(1),IQ(26),IR(26),OPQ(501)
COMPLEX FREQV(1),API(26),BB(26),TRNSF(1),H(1),EJ(1),EP(1),ER(1),
*APIB(26),CO(1),CP(1),CQ(1),CR(1),CB1(1),CB2(1)
OPEN(6,FILE='Y.YAZ',STATUS='OLD')
OPEN(9,FILE='Y.FOU',STATUS='OLD')
CALL MAINLN(KV,RF,FRQV,NHV,TIMEV,XO,T,ETA,A,B,C,D,F,G,FIC,FI,EN,
*EM,O,P,Q,R,S,V,IQ,IR,OPQ,FREQV,API,BB,TRNSF,H,EJ,EP,ER,APIB,CO,CP,
*CQ,CR,CB1,CB2)
STOP
END
```

```
C*****
SUBROUTINE READO (INF,D1,N,ICONT,TO,NT0,KVD,KV,K,MI,MO)
  DIMENSION KV(1)
  WRITE(*,*)'ICONT=? 0 DUR.DEG.LERI SUREKSIZ F VE G LER VAR'
  WRITE(*,*)'ICONT=? 1 DRM.DEG.LERI SUREKLI F VE G LER YOK'
  WRITE(*,*)'LS=0 ISE SUREKLI ICONT=1 GIRINIZ'
  WRITE(*,*)'LS 0 DAN FARKLI ISE SUREKSIZ ICONT=0 GIRINIZ'
  READ(*,*)ICONT
  WRITE(9,*)ICONT
  WRITE(*,*)'F=? HZ'
  READ(*,*)F
  TO=1./F
  WRITE(9,*)F
  WRITE(*,*)'CEYREK PERYOTTAKI ANAHTARLAMA ACISI SAYISI=?'
  READ(*,*)KACI
  K=4*KACI+2
  IF(KACI.EQ.1)K=2
  MI=1
  MO=2
  WRITE(6,*)'READO DA VERILENLER.'
  IF(ICONT.EQ.0)WRITE(6,*)'ICONT=0 DURUM DEGISKENLERI SUREKSIZ'
  IF(ICONT.EQ.1)WRITE(6,*)'ICONT=1 DURUM DEGISKENLERI SUREKLI'
  WRITE(6,*)'TO=? SANIYE,TO
  WRITE(6,*)'K=?',K
  WRITE(6,*)'MI=?',MI
  WRITE(6,*)'MO=?',MO
  WRITE(6,*)'.....'
  RETURN
END
```

```
C*****
SUBROUTINE SINPT(T,A,B,C,D,F,G,H,K,TT,DT,IYAZ)
  DIMENSION A(1),B(1),C(1),D(1),F(1),G(1),T(1)
  WRITE(*,*)'F=? HZ'
  READ(*,*)FR
  TO=1./FR
  WRITE(*,*)'ICONT=0 ISE DURUM DEGISKENLERI SUREKSIZ ?'
  WRITE(*,*)'ICONT=1 ISE DURUM DEGISKENLERI SUREKLI ?'
  WRITE(*,*)'ICONT=?'
  READ(*,*)ICONT
  WRITE(*,*)'RS=? OHM'
  READ(*,*)RS
  IF(RS.LT.1.E-30)RS=1.E-30
  WRITE(9,*)RS
  WRITE(*,*)'LS=? HENRY'
  READ(*,*)ALS
  IF(ALS.LT.1.E-30)ALS=1.E-30
  WRITE(9,*)ALS
  WRITE(*,*)'RY=? OHM'
  READ(*,*)RY
  IF(RY.LT.1.E-30)RY=1.E-30
  WRITE(9,*)RY
  WRITE(*,*)'LY=? HENRY'
  READ(*,*)ALY
  IF(ALY.LT.1.E-30)ALY=1.E-30
  WRITE(9,*)ALY
  WRITE(6,*)'SINPT DE VERILENLER.'
  WRITE(6,*)'RS=? OHM',RS
  WRITE(6,*)'LS=? HENRY',ALS
  WRITE(6,*)'RY=? OHM',RY
  WRITE(6,*)'LY=? HENRY',ALY
  IYAZ=1
  WRITE(6,*)'IYAZ=?',IYAZ
  WRITE(*,*)'YUK GERILIMI HESAPLANACAKSA 0, YUK AKIMI ISE 1 GIR ?'
  READ(*,*)KGA
  IF(KGA.EQ.0)WRITE(6,*)'YUK GERILIMI HESAPLANACAK'
  IF(KGA.EQ.1)WRITE(6,*)'YUK AKIMI HESAPLANACAK'
  WRITE(9,*)KGA
  KACI=(K-2)/4
  T(0)=0.0
  ALF=0.0
  DO 133 I=1,KACI
  WRITE(*,*)'ALFA=? DERECE'
  READ(*,*)ALFA
  T(1)=(ALFA*TO/360)-(ALF*TO/360)
  ALF=ALFA
133 CONTINUE
  T(KACI+1)=2*(90-ALF)*TO/360
  DO 134 I=1,KACI
```

```

134 T(KACI+1+I)=T(KACI+1-1)
DO 135 I=1,2*KACI+1
135 T(2*KACI+1+I)=T(I)
DO 414 I=1,K
WRITE(6,*)T('I,')=? SANIYE',T(I)
414 WRITE(*,*)T('I,')=? SANIYE',T(I)
DO 31 I=1,K/2
A(2*I-1)=-(RY+RS)/(ALY+ALS)
B(2*I-1)=1./(ALY+ALS)
C(2*I-1)=(ALS*RY-RS*ALY)/(ALY+ALS)
D(2*I-1)=ALY/(ALY+ALS)
IF(KGA.EQ.1)C(2*I-1)=1
IF(KGA.EQ.1)D(2*I-1)=0
IF(ICONT.EQ.1)GO TO 31
F(2*I-1)=(ALY-ALS)/(ALY+ALS)
31 CONTINUE
DO 32 I=1,K/2
A(2*I)=-(RY+RS)/(ALY+ALS)
B(2*I)=-1./(ALY+ALS)
C(2*I)=(ALS*RY-RS*ALY)/(ALY+ALS)
D(2*I)=-ALY/(ALY+ALS)
IF(KGA.EQ.1)C(2*I)=1
IF(KGA.EQ.1)D(2*I)=0
IF(ICONT.EQ.1)GO TO 32
F(2*I)=(ALY-ALS)/(ALY+ALS)
32 CONTINUE
RETURN
END
C*****
SUBROUTINE SOTPT (NF,F1,F2,IF,RF,FRQV,NH,N1,N2,IH,NHV,NT,T1,T2,
*IT,TIMEV,XO)
DIMENSION FRQV(1),RF(1),NHV(1),TIMEV(1),XO(1)
WRITE(6,*)SOTPT DE VERILENLER.'
F2=1.0E-30
WRITE(6,*)F2=? HERTZ',F2
IT=1
WRITE(6,*)IT=?',IT
C WRITE(*,*)NT=?'
C READ(*,*)NT
NT=501
WRITE(6,*)NT=?',NT
WRITE(9,*)NT
RETURN
END
C*****
COMPLEX FUNCTION UFUNC(P)
COMPLEX P
UFUNC=(1.,0.)
RETURN
END

```


EK-9 DISCRETE FOURIER ANALİZİ YAPAN D BİLGİSAYAR PROGRAMININ LİSTESİ

```

DIMENSION TETA(501),A(250),B(250),AB(251),YG(501),AB0(251)
OPEN(4,FILE='KEMPO.OKU',STATUS='OLD')
OPEN(5,FILE='Y.FOU',STATUS='OLD')
OPEN(6,FILE='D.YAZ',STATUS='OLD')
CALL GIR(F,NT,RS,ALS,RY,ALY,KTC,KPI,ART,KGB,M,KN,KOL,KH,KGA)
CALL TTTYG(NT,M,TETA,YG,F,KGA)
CALL DIHAR(NT,M,TETA,A,B,AB,YG,ORT,F)
IF(KOL.EQ.0)GO TO 22
CALL KEMPO(ORT,AB,M)
22 CALL THDIS(AB,M,THD,KH)
CALL TCIZ(KTC,A,B,ORT,M,ART,KPI)
IF(KN.EQ.0)GO TO 23
CALL NORMZ(M,AB,ORT)
23 CONTINUE
IF(KGB.EQ.0)GO TO 999
CALL GERBOL(F,RS,ALS,RY,ALY,AB0,M,AB,ORT)
999 STOP
END
C*****
SUBROUTINE GIR(F,NT,RS,ALS,RY,ALY,KTC,KPI,ART,KGB,M,KN,KOL,KH,KGA)
PI=3.141592654
WRITE(6,*)'GIR'
WRITE(*,*)'TOPL.HARM.DIST.HESABINDA BUTUN HARM.LER ALINACA KSA 0'
WRITE(*,*)'3 VE KATLARI ALINMAYACA KSA 1 GIRINIZ'
READ(*,*)KH
WRITE(*,*)
WRITE(6,*)'Y.FOR(MAINLN) NIN URETTIGI Y.FOU FILE IN DAN'
WRITE(6,*)'OKUNAN DEGERLER'
READ(5,*)ICONT
IF(ICONT.EQ.0)WRITE(6,*)'ICONT=0 DURUM DEGİSKENLERI SUREKSİZ'
IF(ICONT.EQ.1)WRITE(6,*)'ICONT=1 DURUM DEGİSKENLERI SUREKLI'
READ(5,*)F
TO=1./F
WRITE(6,*)'TO=',TO,' SANIYE,' F='F,' Hz'
READ(5,*)NT
NT=NT-1
WRITE(6,*)'NT=',NT
READ(5,*)RS
WRITE(6,*)'RS=',RS,' OHM'
READ(5,*)ALS
WRITE(6,*)'LS=',ALS,' HENRY'
READ(5,*)RY
WRITE(6,*)'RY=',RY,' OHM'
READ(5,*)ALY
WRITE(6,*)'LY=',ALY,' HENRY'
READ(5,*)KGA
IF(KGA.EQ.0)WRITE(6,*)'YUK GERILIMI HESAPLANDI'
IF(KGA.EQ.1)WRITE(6,*)'YUK AKIMI HESAPLANDI'
WRITE(6,*)'KLAVYEDEN GIRILEN DEGERLER'
WRITE(*,*)'NORMALIZE ETSIN MI? KN=1 EVET,,KN=0 HAYIR'
READ(*,*)KN
WRITE(6,*)'NORMALIZE ETSIN MI? KN=1 EVET,,KN=0 HAYIR',KN
WRITE(*,*)'M İLE VERİLECEK KADAR HARMONİĞİN TOPLAMINDAN '
WRITE(*,*)'TEKRAR CİZSİN MI? KTC=1 EVET,,KTC=0 HAYIR'
READ(*,*)KTC
WRITE(6,*)'M İLE VERİLECEK KADAR HARMONİĞİN TOPLAMINDAN '
WRITE(6,*)'TEKRAR CİZSİN MI? KTC=1 EVET,,KTC=0 HAYIR',KTC
IF(KTC.EQ.0)GO TO 23
C WRITE(*,*)'KAC PI İCİN CİZSİN ?'
C READ(*,*)KPI
KPI=2
C WRITE(*,*)'ARTIM=? ?'
C READ(*,*)ART
ART=2*PI/(NT)
23 CONTINUE
WRITE(*,*)'GECERLİ OLAN HARMONİK SAYISI=NT/2',NT/2
WRITE(*,*)'KACINCI HARMONİĞE KADAR HES, M=?'
READ(*,*)M
WRITE(6,*)'GECERLİ OLAN HARMONİK SAYISI=NT/2',NT/2
WRITE(6,*)'KACINCI HARMONİĞE KADAR HES, M=?',M
WRITE(*,*)'KAYNAK TARA Fİ EMPEDANSI=0 İCİN YETW50S CALİSTIRILMİS'
WRITE(*,*)'OLMALI. KEMPO OLUSTURULSUN=1 OLUSTURULMASIN=0'
WRITE(*,*)'ONCEDEN OLUSTURULDUYSA 0 GIR'
READ(*,*)KOL
IF(KOL.EQ.1)WRITE(6,*)'KEMPO.OKU OLUSTURULACAK'
IF(KOL.EQ.0)WRITE(6,*)'KEMPO.OKU OLUSTURULMAYACAK'
IF(KOL.EQ.1)KGB=0
IF(KOL.EQ.1)GO TO 81
IF(KOL.EQ.0)KGB=1
IF(KOL.EQ.0)GO TO 81
WRITE(*,*)'GERBOL KULLANILSIN MI? EVET=1, HAYIR=0'
READ(*,*)KGB
81 WRITE(6,*)'GERBOL KULLANILSIN MI? EVET=1, HAYIR=0',KGB
WRITE(6,*)
RETURN
END
C*****
SUBROUTINE TTTYG(NT,M,TETA,YG,F,KGA)
DIMENSION TETA(1),YG(1)
YGM=0.0
PI=3.141592654
WRITE(6,*)'SUBR. TTTYG ÇIKTI Sİ'

```



```

WRITE(6,*)'1.SUTUN TIME SANIYE'
WRITE(6,*)'2.SUTUN TETA RADYAN'
WRITE(6,*)'3.SUTUN TETA DERECE'
IF(KGA.EQ.0)WRITE(6,*)'4.SUTUN YUK GERILIMI DEGERI YG'
IF(KGA.EQ.1)WRITE(6,*)'4.SUTUN YUK AKIMI DEGERI YG'
WRITE(6,*)'5.SUTUN 4.SUTUN UN NORMALIZE DEGERI'
DO 40 I=1,NT
  READ(5,*)YG(I)
40 IF(ABS(YG(I)).GT.YGM)YGM=ABS(YG(I))
DO 41 I=1,NT
  TIME=(I-1)/(NT*F)
  TETA(I)=TIME*2*PI*F
  TETAD=TETA(I)*180/PI
C  READ(5,*)YG(I)
  WRITE(*,*)TIME,TETA(I),TETAD,YG(I),YG(I)/YGM
41 WRITE(6,*)TIME,TETA(I),TETAD,YG(I),YG(I)/YGM
  WRITE(6,*)
  RETURN
  END
C*****
SUBROUTINE DIHAR(NT,M,TETA,A,B,AB,YG,ORT,F)
DIMENSION TETA(1),A(1),B(1),AB(1),YG(1)
PI=3.141592654
C A COS.LU TERIMLERIN KATSAYILARI
C B SIN.LU TERIMLERIN KATSAYILARI
C AB(I)=SQRT(A(I)**2+B(I)**2)
  WRITE(6,*)'SUBR. DIHAR CIKTISI'
  WRITE(*,*)'CIZGI SPEKTRUMU VE TOPLAM HARM.DISTORSIYONU.ICIN'
  WRITE(*,*)'1.SUTUN HARMONIK NUMARASI'
  WRITE(*,*)'2.SUTUN HARMONIK GENLIGI'
  WRITE(6,*)'CIZGI SPEKTRUMU VE TOPLAM HARM.DISTORSIYONU.ICIN'
  WRITE(6,*)'1.SUTUN HARMONIK NUMARASI'
  WRITE(6,*)'2.SUTUN HARMONIK GENLIGI'
  ORT=0.
  DO 41 I=1,NT
    C  READ(5,*)YG(I)
41 ORT=ORT+YG(I)
    ORT=ORT/NT
    DO 10 I=1,M
      A(I)=0.
      B(I)=0.
      DO 11 J=1,NT
        A(I)=A(I)+(2./NT)*YG(J)*COS(I*TETA(J))
11 B(I)=B(I)+(2./NT)*YG(J)*SIN(I*TETA(J))
10 AB(I)=SQRT(A(I)**2+B(I)**2)
      I=0
      WRITE(*,*)I,ABS(ORT)
      WRITE(6,*)I,ABS(ORT)
      DO 687 I=1,M
        WRITE(*,*)I,AB(I)
        WRITE(6,*)I,AB(I)
687 CONTINUE
      WRITE(6,*)
      RETURN
      END
C*****
SUBROUTINE KEMPO(ORT,AB,M)
DIMENSION AB(1)
C KEMPO.OKU YU OLUSTURMAK ICIN
  WRITE(4,*)ABS(ORT)
  DO 7305 I=1,M
    WRITE(4,*)AB(I)
7305 CONTINUE
  RETURN
  END
C*****
SUBROUTINE THDIS(AB,M,THD,KH)
DIMENSION AB(1)
THD=0.0
IF(KH.EQ.0)GO TO 66
DO 10 I=2,M
  BOS=1
C 3 VE 3 UN KATLARI OLAN HARMONIK AKIMLARI 3 FAZLI 3 TELLI SIST.DE YOK
  IF((BOS/3-I/3).LT.1.E-5)GO TO 10
  THD=THD+AB(I)**2
10 CONTINUE
  THD=SQRT(THD)/AB(1)
  GO TO 71
66 CONTINUE
DO 11 I=2,M
  THD=THD+AB(I)**2
11 CONTINUE
  THD=SQRT(THD)/AB(1)
71 CONTINUE
  WRITE(6,*)'SUBR. THDIS CIKTISI'
  WRITE(*,*)'3 VE 3 UN KATI OLAN HARM. AKIMLARI 3 FAZLI 3 TELLI'
  WRITE(*,*)'SIST.DE YOK'
  WRITE(6,*)'3 VE 3 UN KATI OLAN HARM. AKIMLARI 3 FAZLI 3 TELLI'
  WRITE(6,*)'SIST.DE YOK'
  IF(KH.EQ.0)WRITE(*,*)'BUTUN HARM.KULL.TH D HESAPLANDI'
  IF(KH.EQ.0)WRITE(6,*)'BUTUN HARM.KULL.TH D HESAPLANDI'
  IF(KH.EQ.1)WRITE(*,*)'3 VE KATLARI HARIC.TH D HESAPLANDI'
  IF(KH.EQ.1)WRITE(6,*)'3 VE KATLARI HARIC.TH D HESAPLANDI'

```

```

WRITE(*,*)'TOPLAM HARMONIK DISTORSİYONU=THD=',THD
WRITE(6,*)'TOPLAM HARMONIK DISTORSİYONU=THD=',THD
WRITE(6,*)
RETURN
END
C*****
SUBROUTINE GERBOL(F,RS,ALS,RY,ALY,ABO,M,AB,ORT)
DIMENSION ABO(1),AB(1)
WRITE(6,*)'SUBR. GERBOL CIKTISI'
PI=3.141592654
DO 546 I=0,M
546 READ(4,*)ABO(I)
WRITE(*,*)'1.SUTUN HARMONIK NUMARASI'
WRITE(*,*)'2.SUT.KAYN.EMP=0 ICIN ONCEDEN BELIRL.HARM.GENL.KEMPO'
WRITE(*,*)'3.SUTUN RS=',RS,' ICIN GERBOL LE BELIRL.HARM.GENLIGI'
WRITE(*,*)'GERBOL GECERLI ISE BU 3.SUTUN YUK.THDS IN 2.SUTUNUNA'
WRITE(*,*)'ESIT CIKAR'
WRITE(6,*)'1.SUTUN HARMONIK NUMARASI'
WRITE(6,*)'2.SUT.KAYN.EMP=0 ICIN ONCEDEN BELIRL.HARM.GENL.KEMPO'
WRITE(6,*)'3.SUTUN RS=',RS,' ICIN GERBOL LE BELIRL.HARM.GENLIGI'
WRITE(6,*)'GERBOL GECERLI ISE BU 3.SUTUN YUK.THDS IN 2.SUTUNUNA'
WRITE(6,*)'ESIT CIKAR'
I=0
ORT0=ORT
ORT=ORT0*RY/(RS+RY)
WRITE(*,*)I,ORT0,ORT
WRITE(6,*)I,ORT0,ORT
DO 5 I=1,M
XLS=2*PI*I*F*ALS
XLY=2*PI*I*F*ALY
AB(I)=(ABO(I)/SQRT((RS+RY)**2+(XLS+XLY)**2))*SQRT(RY**2+XLY**2)
WRITE(*,*)I,ABO(I),AB(I)
5 WRITE(6,*)I,ABO(I),AB(I)
WRITE(6,*)
RETURN
END
C*****
SUBROUTINE TCIZ(KTC,A,B,ORT,NT,ART,KPI)
DIMENSION A(1),B(1)
IF(KTC.EQ.0)GO TO 999
WRITE(*,*)'FOURIERLE TEKRAR ELDE EDILEN SEKLI CIZMEK ICIN'
WRITE(*,*)'1.SUTUN X EKSENI RADYAN'
WRITE(*,*)'2.SUTUN',NT,' TERIM TOPL.DAN ELDE EDILEN ISARET'
WRITE(6,*)'SUBR. TCIZ CIKTISI'
WRITE(6,*)'FOURIERLE TEKRAR ELDE EDILEN SEKLI CIZMEK ICIN'
WRITE(6,*)'1.SUTUN X EKSENI RADYAN'
WRITE(6,*)'2.SUTUN',NT,' TERIM TOPL.DAN ELDE EDILEN ISARET'
PI=3.141592654
X=0.
C HARMONIKLER
C CIZIM ICIN
5 FX=0.
DO 2 I=1,NT
2 FX=FX+A(I)*COS(I*X)+B(I)*SIN(I*X)
C VERILEN ISARETI HARMONIKLERIN TOPLAMINDAN ELDE ETTIM
FX=FX+ORT
WRITE(*,*)X,FX
WRITE(6,*)X,FX
X=X+ART
IF(X.GE.(KPI*PI))GO TO 999
GO TO 5
999 RETURN
END
C*****
SUBROUTINE NORMZ(M,AB,ORT)
DIMENSION AB(1)
IF(ABS(ORT).LT.1.E-30)ORT=1.E-30
ABMAX=ABS(ORT)
IMAX=0
DO 776 I=1,M
IF(AB(I).LT.1.E-30)AB(I)=1.E-30
IF(AB(I).GT.ABMAX)IMAX=I
776 IF(AB(I).GT.ABMAX)ABMAX=AB(I)
WRITE(*,77)IMAX,ABMAX
WRITE(6,77)IMAX,ABMAX
77 FORMAT(/,'SUBR.NORMZ CIKTISI,.....',/,',1.SUTUN HA
&RM NO',/,',2.SUTUN GENLIGI EN BUYUK OLAN',I5,', INCI HARMONIK=',F10
&.5,', ILE',/,', NORMALIZE EDILMIS GENLIK DEGERLERI',/,',3.SUTUN NORMA
&LIZE GENLIKLERIN dB OLARAK ZAYIFLAMA DEGERLERI',/)
I=0
ORTN=ABS(ORT)/ABMAX
ZABN=20.*ALOG10(ORTN)
WRITE(*,*)I,ORTN,ZABN
WRITE(6,*)I,ORTN,ZABN
DO 777 I=1,M
ABN=AB(I)/ABMAX
ZABN=20.*ALOG10(ABN)
WRITE(*,*)I,ABN,ZABN
777 WRITE(6,*)I,ABN,ZABN
WRITE(6,*)
RETURN
END

```