

66585

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**ALÇAK GEÇİREN VE BAND GEÇİREN AKTİF OTA-C
FİLTRELERİNİN QUAD VE KASKAD
YÖNTEMİ İLE TASARIMI**

YÜKSEK LİSANS TEZİ

Müh. Ömer ORAK

Tezin Enstitüye Verildiği Tarih : 20 Ocak 1997

Tezin Savunulduğu Tarih : 6 Şubat 1997

Tez Danışmanı : Prof. Dr. Ali Nur GÖNÜLEREN

Diğer Jüri Üyeleri : Prof. Dr. Cevdet ACAR

Prof. Dr. Fuat ANDAY

ŞUBAT 1997

ÖNSÖZ

Bu tezin hazırlanması sırasında yapıcı uyarıları ile çalışmalarına yön veren ve yakın ilgi gösteren değerli hocam Sayın Prof. Dr. Ali Nur GÖNÜLEREN'e teşekkürlerimi sunarım.

Ocak 1997

Ömer Orak



İÇİNDEKİLER

	Sayfa No
ÖNSÖZ.....	ii
SEMBOL LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	vii
TABLO LİSTESİ.....	ix
ÖZET.....	x
SUMMARY.....	xi
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. ALÇAK GEÇİREN VE BAND GEÇİREN FİLTRELERİN TASARIMI.....	5
2.1. Filtrelerin Sınıflandırılması.....	5
2.2. Yaklaşıklık Tipleri.....	5
2.3. Alçak Geçiren Filtre Transfer Fonksiyonunun Elde Edilmesi.....	15
2.3.1. Prototip alçak geçiren-alçak geçiren frekans dönüşümü.....	15
2.4. Band Geçiren Filtre Transfer Fonksiyonunun Elde Edilmesi.....	21
2.4.1. Prototip alçak geçiren-band geçiren frekans dönüşümü.....	22
BÖLÜM 3. QUAD YÖNTEMİ.....	24
3.1. Quad Devre Yapısı.....	24
3.1.1. Quad devresinin gerçekleştirilmesi.....	26
3.1.2. Quad bloğu geribesleme katsayısının belirlenmesi.....	28
3.2. İkinci Dereceden Transfer Fonksiyonundan Alçak Geçiren Band Geçiren Dönüşümü Yardımıyla Elde Edilen 4.Dereceden Band Geçiren Filtrelerin Tasarımı.....	30
3.3. Yüksek Dereceden Filtrelerin Quad Yöntemi İle Tasarımı.....	32
3.3.1. Sıfır-kutup düzenlemesi.....	33
BÖLÜM 4. ALÇAK GEÇİREN VE BAND GEÇİREN FİLTRELERİN QUAD VE KASKAD TASARIM DENKLEMLERİNİN ELDE EDİLMESİ.....	35
4.1. Quad Tasarım Yöntemi Kullanılarak Alçak Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması.....	35

4.2.	Kaskad Tasarım Yöntemi Kullanılarak Alçak Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması.....	42
4.3.	Elde Edilen Alçak Geçiren Filtre Transfer Fonksiyonu İle Quad ve Kaskad Tasarım.....	47
4.4.	Quad Tasarım Yöntemi Kullanılarak Band Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması.....	49
4.5.	Kaskad Tasarım Yöntemi Kullanılarak Band Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması.....	52
4.6.	Elde Edilen Band Geçiren Filtre Transfer Fonksiyonu İle Quad ve Kaskad Tasarım.....	55
BÖLÜM 5.	OTA MODELLERİ VE AKTİF OTA-C BİQUADRATİK DEVRELERİ.....	57
5.1.	Giriş.....	57
5.2.	OTA Elemanı ve Modelleri.....	58
5.2.1.	İdeal OTA modeli.....	58
5.2.2.	Gerçek OTA modeli.....	58
5.2.3.	Bipolar OTA.....	59
5.2.4.	CMOS OTA.....	60
5.3.	Literatürde Verilmiş Olan Aktif OTA-C Biquadratik Devreleri.....	63
BÖLÜM 6.	YÜKSEK DERECEDEN QUAD VE KASKAD FİLTRELERİNİN DİNAMİK DEĞİŞİMİ	75
6.1.	Quad ve Kaskad Yönteminde Dinamik Değişimin İyileştirilmesi..	75
6.2.	Kazanç Katsayılarının Belirlenmesi ve Biquadratik Alt Devrelerin Gerçeklenmesi.....	83
6.2.1.	Doyma durumuna göre kazanç katsayılarının bulunması.....	83
6.2.2.	Biquadratik altdevrelerin OTA-C elemanlarıyla gerçekleştirilmesi.....	86
BÖLÜM 7.	OTA-C QUAD VE KASKAD FİLTRELERİNİN MAKSİMUM GİRİŞ GERİLİMİNİN HESAPLANMASI.....	88
7.1.	Genel Kavramlar.....	88
7.2.	Yüksek Dereceden Butterworth, Chebyshev ve Eliptik Quad/ Kaskad Filtrelerinin Maksimum Giriş Geriliminin Hesaplanması...	90

7.2.1. Alçak geçiren ve band geçiren Butterworth ve Chebyshev	
filtrelerinin maksimum giriş geriliminin bulunması.....	91
7.2.2. Alçak geçiren ve band geçiren Eliptik filtrelerin	
maksimum giriş geriliminin bulunması.....	94
SONUÇLAR VE ÖNERİLER.....	97
KAYNAKLAR.....	99
EK A.....	102
EK B.....	121
ÖZGEÇMİŞ	174



SEMBOL LİSTESİ

ε	: Salınım faktörü
N	: Filtre derecesi
p_k	: Prototip alçak geçiren filtrenin kutupları
w_p	: Alçak geçiren filtre geçirme bandı açısal frekansı
A_i, B_i	: Prototip alçak geçiren filtre transfer fonksiyonunun katsayıları
n_x	: Band geçiren filtre quad bloğu sayısı
t_x	: Alçak geçiren filtre quad bloğu sayısı
w_o	: Band geçiren filtre merkez frekansı
B	: Band geçiren filtre band genişliği
f_p	: Alçak geçiren filtre geçirme bandı frekansı
f_a	: Alçak geçiren filtre söndürme bandı frekansı
A_p	: Alçak geçiren filtre için maksimum geçirme bandı zayıflatması
A_a	: Alçak geçiren filtre için minimum söndürme bandı zayıflatması
f_{a1}	: Band geçiren filtre için alt söndürme bandı frekansı
f_{p1}	: Band geçiren filtre için alt geçirme bandı frekansı
f_{p2}	: Band geçiren filtre için üst geçirme bandı frekansı
f_{a2}	: Band geçiren filtre için üst söndürme bandı frekansı
$Q_1, \dots, Q_N$	: Quad blokları iyilik faktörleri
$f_1, \dots, f_N$	: Kazanç değerleri atanmamış quad blokların geri besleme katsayıları
$f'_1, \dots, f'_N$	: Kazanç değerleri atanmış quad blokların geri besleme katsayıları
$T_{Qi}(s)$	: Quad blokları transfer fonksiyonu
$T_{ki}(s)$	: Kaskad blokları transfer fonksiyonu
$T_{1i}(s), T_{2i}(s)$	: Quad bloklarını oluşturan biquadratik devrelerin transfer fonksiyonları
$C_N(\Omega)$	: N.dereceden Chebyshev polinomu
H_o, T_a, T_b	: Quad ve kaskad devrelerine ilişkin alt devreler
T_x	: H_o, T_a, T_b alt devrelerinden oluşan blok

ŞEKİL LİSTESİ

		<u>Sayfa No</u>
Şekil 2.1	AG filtre karakteristikleri	6
Şekil 2.2	YG filtre karakteristikleri	6
Şekil 2.3	BG filtre karakteristikleri	7
Şekil 2.4	BS filtre karakteristikleri	7
Şekil 3.1	Quad devre yapısı	24
Şekil 3.2	Yüksek dereceden quad devre topolojisi	32
Şekil 4.1	Yüksek dereceden alçak geçiren quad filtresi blok diyagramı	47
Şekil 4.2	Yüksek dereceden alçak geçiren kaskad filtresi blok diyagramı	48
Şekil 4.3	Yüksek dereceden band geçiren quad filtresi blok diyagramı	55
Şekil 4.4	Yüksek dereceden band geçiren kaskad filtresi blok diyagramı	55
Şekil 5.1	OTA devre sembolü	58
Şekil 5.2	İdeal eşdeğer devre	58
Şekil 5.3	İdeal olmayan OTA eşdeğer devresi	59
Şekil 5.4	Bipolar OTA'nın iç yapısı	59
Şekil 5.5	Simetrik CMOS OTA yapısı	61
Şekil 5.6	V_s 'nin g_m ile değişimi	62
Şekil 5.7	I_s 'nin g_m ile değişimi	63
Şekil 5.8	Tablo 5.2'deki transfer fonksiyonlarını gerçekleyen genel OTA-C biquad devresi	64
Şekil 5.9	Temel OTA-C yapıları	66
Şekil 5.10	Genel OTA-C biquad	67
Şekil 5.11	$G(s)$ 'e karşı düşen işaret akış diyagramı	68
Şekil 5.12	$G(s)$ 'i gerçekleyen OTA-C devresi	68
Şekil 5.13	$G(s)$ 'i gerçekleyen minimum OTA-C biquad devresi	70
Şekil 5.14	Genel amaçlı bir OTA-C biquad devresi	73
Şekil 6.1	Yüksek dereceden alçak geçiren quad filtresine ilişkin kritik gerilimler	76
Şekil 6.2	Yüksek dereceden alçak geçiren kaskad filtresine ilişkin kritik gerilimler	76

Şekil 6.3	Yüksek dereceden band geçiren quad filtresine ilişkin kritik gerilimler	76
Şekil 6.4	Yüksek dereceden band geçiren kaskad filtresine ilişkin kritik gerilimler	77
Şekil 6.5	H_0 kazançlı giriş bloğuna ilişkin alt devre	77
Şekil 6.6.a	T_a bloğuna ilişkin RC devresi	77
Şekil 6.6.b	OTA'larla direnç gerçekleştirilmesi	78
Şekil 6.6.c	Eşdeğer devre	78
Şekil 6.6.d	T_a bloğuna ilişkin OTA-C devresi	79
Şekil 6.7.a	Butterworth ve Chebyshev yaklaşıklık için alçak geçiren OTA-C quad/kaskad devresindeki T_b bloğuna ilişkin alt devre	79
Şekil 6.7.b	Eliptik yaklaşıklık için alçak geçiren OTA-C quad/kaskad devresindeki T_b bloğuna ilişkin alt devre	79
Şekil 6.8	Band geçiren OTA-C quad/kaskad devresindeki T_b bloğuna ilişkin alt devre	80
Şekil 6.9.a	Butterworth ve Chebyshev filtre için alçak geçiren OTA-C quad alt devresi	80
Şekil 6.9.b	Butterworth ve Chebyshev filtre için band geçiren OTA-C quad alt devresi	80
Şekil 6.9.c	Eliptik filtre için alçak geçiren /band geçiren quad alt devresi	81
Şekil 6.10	Kazanç değerleri atanmış alçak geçiren quad filtresi	84
Şekil 6.11	Kazanç değerleri atanmış alçak geçiren kaskad filtresi	84
Şekil 6.12	Kazanç değerleri atanmış band geçiren quad filtresi	84
Şekil 6.13	Kazanç değerleri atanmış band geçiren kaskad filtresi	84
Şekil A.1	Örnek 1'deki alçak geçiren Butterworth yaklaşıklı filtreye ilişkin frekans karakteristiği	105
Şekil A.2	Örnek 2'deki alçak geçiren Chebyshev yaklaşıklı filtreye ilişkin frekans karakteristiği	109
Şekil A.3	Örnek 4'deki band geçiren Butterworth filtresine ilişkin frekans karakteristiği	115
Şekil A.4	Örnek 5'deki band geçiren Chebyshev filtresine ilişkin frekans karakteristiği	118

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 5.1 Şekil 5.5’de kullanılan tranzistorlar için tipik değerler	62
Tablo 5.2 Şekil 5.8’deki devreye ilişkin çeşitli koşullara göre elde edilen filtre tipleri	65
Tablo 5.3 Şekil 5.10’daki biquada ilişkin değişik koşullar ve filtre tipleri	67
Tablo 5.4 Şekil 5.11’deki işaret akış grafi kullanılarak oluşturulan OTA-C filtreleri	71
Tablo 5.5 Şekil 5.14’deki biquad için çeşitli koşullara göre elde edilen filtre tipleri	73
Tablo 6.1 Butterworth, Chebyshev ve Eliptik türden quad/kaskad filtresine ilişkin T_a bloğunun ve temel biquad devrelerinin eleman değerleri	87
Tablo A.1 Örnek 1’e ilişkin kazanç katsayıları	103
Tablo A.2 Örnek 1’e ilişkin eleman değerleri	104
Tablo A.3 Örnek 2’deki filtreye ilişkin kazanç katsayıları	106
Tablo A.4.a Örnek 2’ye ilişkin alt blok eleman değerleri	108
Tablo A.4.b Örnek 2’ye ilişkin quad blokları eleman değerleri	108
Tablo A.5 Örnek 3’e ilişkin kazanç katsayıları	110
Tablo A.6 Örnek 3’e ilişkin eleman değerleri	112
Tablo A.7 Örnek 4’e ilişkin kazanç katsayıları	113
Tablo A.8 Örnek 4’e ilişkin eleman değerleri	114
Tablo A.9 Örnek 5’e ilişkin kazanç katsayıları	116
Tablo A.10 Örnek 5’e ilişkin eleman değerleri	117
Tablo A.11 Örnek 6’ya ilişkin kazanç katsayıları	119

ÖZET

OTA (Operational Transconductance Amplifier) monolitik tümleştirme tekniği ile kolayca gerçekleştirilebilir, elektronik olarak ayarlanabilir ve tasarım kolaylığı sağlar. Yüksek frekans performansları işlemsel kuvvetlendiricilerden çok daha iyidir. OTA elemanı ile oluşturulan filtrelerde OTA iletkenlik kazancı tasarım parametresi olarak kullanılabilir. Ayrıca OTA, CMOS teknolojisine uygun bir elemandır. Bir ucu topraklı kapasite elemanı da CMOS teknolojisi ile kolayca gerçekleştirilebilir ve OTA ile beraber uyumlu bir eleman çifti oluşturur. Filtre tasarımında ,son yıllarda OTA ve kapasite elemanları sayılan bu nedenlerden dolayı, işlemsel kuvvetlendirici ve RC elemanlarından çok daha tercih edilir duruma gelmiştir.

Bu çalışmada, sadece OTA ve kapasite elemanları kullanılarak, quad ve kaskad tasarım yöntemiyle, yüksek dereceden aktif filtre tasarımı amaçlanmıştır. Filtreler ve yaklaşıklık tipleri kısaca ele alındıktan sonra alçak geçiren ve band geçiren filtre transfer fonksiyonları Butterworth, Chebyshev ve Eliptik yaklaşıklık tipleri için ayrı ayrı elde edilmiştir. Daha sonra quad yöntemi genel bir biçimde anlatılmıştır.

Genel quad yöntemi yardımıyla, alçak geçiren ve band geçiren filtreler için quad ve kaskad denklemlerinin bulunması ayrıntılı bir biçimde ele alındıktan sonra OTA-C elemanlarıyla oluşturulan biquadratik devreler incelenerek, kullanılan alt devrelere ilişkin kazanç değerleri devrenin dinamik davranışını iyileştirecek biçimde belirlenmektedir. Ayrıca tasarlanan OTA-C filtrelerinin girişine uygulanabilecek maksimum gerilim değerinin hesaplanması için izlenecek yol anlatılmıştır.

Ek A'da çeşitli tasarım örnekleri ve bu örneklerde elde edilen filtreler için frekans cevabı karakteristikleri Micro Cap analiz programı yardımıyla verilmektedir. Ek B'de ise alçak geçiren ve band geçiren filtrelerin tasarımını quad ve kaskad yöntemi ile gerçekleştiren bir bilgisayar programı verilmektedir.

SUMMARY

DESIGN OF LOWPASS AND BANDPASS ACTIVE OTA-C FILTERS USING QUAD AND CASCADE METHODS

In electronic circuits, filters are widely used. Filters are circuits which select the electrical signals according to their frequencies. At the beginning of this century, filters were realized using resistors, inductors and capacitors as basic components. In filter realization, inductors caused a lot of problems. These problems is that inductors are very large components and their production and tuning are difficult. In the 1950s, active networks began to take the place of inductors. So, designers begun to use the active-RC filters.

Operational amplifiers have been used in the active filter design for many years, because they are developed and cheap circuit components. The operational amplifiers have not a good frequency performances. This is caused to come out more suitable gain devices such as OTAs.

High frequency behaviour of Operational Transconductance Amplifiers (OTAs) is better than operational amplifiers (OPAMPs).

Furthermore, OTAs provide linear electronic tunability of its transfer gain. For this reason, filter circuits that are realized by using OTAs are more useful than operational amplifiers.

The use of circuits constructed with operational transconductance amplifiers and capacitors is potentially advantageous for the realization of high-frequency continuous-time monolithic analogue linear and nonlinear systems. OTA and grounded capacitors are easy to implement in all IC technologies. Grounded capacitors are natural for the simplest single-polynomial process and can absorb many

capacitive shunt parasitics. Grounded capacitors are the cheapest linear capacitor in integrated circuits. A grounded capacitor can be implemented as input capacitance of MOS transistor.

Several different design methods for high order active filters have been reported in literature to realize the transfer function of the form

$$T(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}, \quad m < n \quad (1)$$

Those methods can be classified into five categories.

- 1- The direct method
- 2- Cascade realization
- 3- Inductance simulation
- 4- Frequency dependent negative resistor (FDNR)
- 5- Multiloop feedback topologies (MF)

The sensitivity behaviour of the direct method is not good. The obtained circuits are very sensitive to component variations.

Cascade and MF topologies are widely used in filter design. Cascade design is commonly used in industry because of the simplicity of the design and ease of tuning. The sensitivity performance of the cascade design is good in stop band, but poor in the pass band.

MF topologies utilize subsection of biquad blocks and have lower sensitivity over the pass band than those of cascade design

Inductance simulation method is expensive and the realized circuits are not modular. In other words, different designs required very different circuit structures. The main aim of inductance simulation and FDNR method is to realize inductorless filter circuits. The sensitivity performance of these designs depend on the quality of the operational amplifiers used.

In this study, design procedures for high order lowpass and bandpass OTA-C filters are presented using the configuration of quad. Quad configuration, which is a special application of Generalized Multiloop-feedback Filters, has many advantages.

Filter circuits that are realized using quad configuration are less sensitive than ones that are realized using the other methods. Quad configuration is general and can be applied to high order symmetrical or unsymmetrical filters. It is easy to tune. It is economic and modular.

The realization of quad configuration is easier than all other methods except the cascade method. It does not require feedforward and summation circuits.

Using the low sensitivity performance of the quad configuration and the advantages of the OTA-C devices yield suitable filters for many applications. In the thesis, design of lowpass and bandpass active OTA-C filters using quad and cascade method is realized for Butterworth, Chebyshev, Elliptic approximations. OTA-C cascade realizations can be easily obtained using the same procedure, because the cascade realization is a special case of the quad configuration.

In section 2, filter classifications and filter approximations types are examined. Transfer functions of lowpass and bandpass filters are obtained for every filter approximation. While this work is realized, firstly the transfer function of prototype lowpas filter is found by using the given properties and then frequency transformations according to the filter type are made.

In section 3, quad method are generally considered. Basic quad topology contain two second order biquadratic blocks and a feedback path as shown in Fig.1. The transfer function of Fig.1 can be expressed as,

$$T_Q(s) = \frac{T_1(s)T_2(s)}{1 + fT_1(s)T_2(s)} \quad (2)$$

where,

$$T_i(s) = \frac{k_i(s^2 + (w_{zi}/Q_{zi})s + w_{zi}^2)}{(s^2 + (w_{pi}/Q_{pi})s + w_{pi}^2)}, \quad i = 1, 2 \quad (3)$$

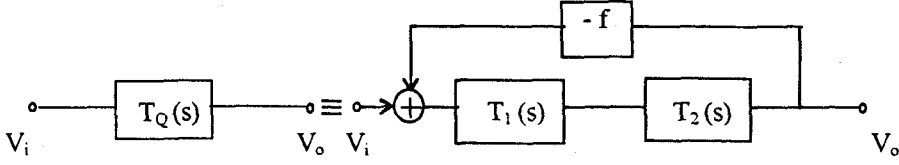


Fig 1. Quad topology

In section 4, the method which should be followed to obtain quad and cascade equations for lowpass and bandpass filter whose transfer functions are determined in section 2 is given in details.

The resulting $T(s)$ filter transfer function is factored into 4th order transfer functions. With the assumption that radial frequencies and quality factors are equal to each other biquadratic transfer functions and feedback coefficients are determined by using the equation (3)

In section 5, OTA component and its models are considered. OTA is basically a voltage controlled current source as seen from Figure 2.

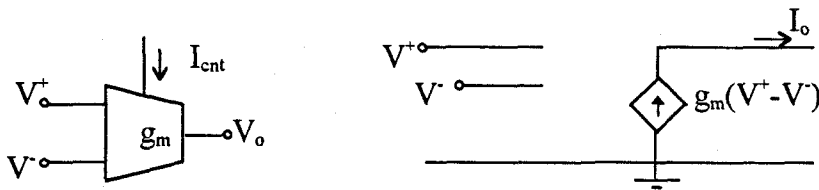


Figure 2. OTA a) Circuit symbol b) Equivalent circuit of ideal OTA

The transconductance gain, g_m , is assumed proportional to I_{cntl} . The proportionality constant k is dependent on temperature, device geometry and the process. The output current is given by,

$$I_o = g_m(V^+ - V^-) \quad (4)$$

The filter may not be operate properly because many practical OTA have some problems. The problems are finite input and output impedance's.

Bipolar OTAs can not operate linearly for the input voltages greater than certain values. In other words, they have limited dynamic range. Bipolar and the symmetrical CMOS-OTAs are also given in section 5. Apart from OTA models , four different biquadratic circuit structures is given by comparing with each other.

In section 6, each quad blocks are realized by using the known dynamic range consideration of quad filters. Pole-zero pairing and ordering of quad blocks have significant effect on the dynamic range of a high order filter. Determination and distribution of gain factors of the biquads are also very important to maximize the dynamic range. Therefore, gain factors of each biquadratic subcircuits and feedback factors are determined in the way to improve the dynamic behaviour of the active filter. In addition to those, the values of OTA and capacitor components that will be used in the filter design are calculated with the aid of the resulting design equations.

The maximum input voltage level is very important to not causing clipping and slew-rate limiting. The output voltage and current of any OTA should not exceed the saturation values for the linear operation of the OTA-C filter. In section 7, the maximum input voltage levels of lowpass and bandpass filters will be determined considering this point.

In the Appendix A, Examples of lowpass and bandpass filters and frequency characteristics are given using Micro Cap Analysis program.

In the Appendix B, a computer program that uses QUICKBASIC language is given. This program realizes the design of lowpass and bandpass active OTA-C filters by using one of the three approximation.

BÖLÜM 1

GİRİŞ

Elektronik ve haberleşme sistemlerinde filtre devreleri geniş bir kullanım alanına sahiptir. Filtreler, elektriksel işaretleri frekanslarına göre seçen ve işleyen devrelerdir. Daha önceleri çoğu filtrelerde temel eleman olarak direnç, kondansatör ve endüktans kullanılmaktaydı. Endüktans elemanlarının boyutlarının büyük olması, maliyetlerinin fazla olması, üretimlerinin ve ayarlarının zor olması sorun oluşturmaktaydı. Teknolojinin gelişimi ile endüktans elemanlarının yerini aktif elemanlar almaya başlamıştır. Böylece işlemsel kuvvetlendiricilerden oluşmuş aktif - RC filtreleri kullanılmaya başlamıştır.

Aktif filtre tasarımında, işlemsel kuvvetlendiriciler oldukça iyi geliştirilmiş ve ucuz elemanlar oldukları için uzun zamandır kullanılmaktadır. Analog ve diğer işaret işleme devrelerinin aynı silikon kırımlık üzerinde tümleştirilmesi isteği ve işlemsel kuvvetlendiricilerle gerçekleştirilen filtrelerin davranışında yüksek frekanslarda bozulmalar olması OTA gibi daha uygun kazanç elemanlarının ortaya çıkmasına neden olmuştur [1].

OTA'ların yüksek frekans davranışlarının işlemsel kuvvetlendiricilerden çok daha iyi olmasından dolayı OTA elemanı kullanılarak gerçekleştirilen filtre devrelerinin daha yüksek frekanslarda çalışabilmesi mümkün olmaktadır. Ayrıca dışarıdan elektriksel olarak kontrol edilebilir iletkenlik parametresi nedeniyle OTA'lar daha kullanışlı devre elemanlarıdır. Filtre tasarımında OTA ve topraklı kapasite elemanlarının kullanılması tümleşik devre teknolojisi açısından avantaj sağlar. Bu avantajların başında OTA elemanının tümleştirme tekniği ile kolayca elde edilebilmesi gelmektedir. Diğer bir avantajı da topraklı kapasite elemanının tümleşik devre yapısında doğal bir eleman olması ve MOS tranzistorun giriş kapasitesi ile gerçekleştirilebilmesidir. Ayrıca

topraklı kapasite, paralel parazitik kapasitif etkileri yok edebilir ve tümleşik devrelerdeki en ucuz lineer kapasitedir [2].

Yüksek dereceden aktif - RC filtrelerinin gerçekleştirilmesi konusunda literatürde oldukça yoğun çalışmalar mevcuttur. s domenindeki ifadesi,

$$T(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}, \quad m < n \quad (1.1)$$

biçiminde verilen gerilim transfer fonksiyonunun aktif-RC devreleri ile gerçekleştirilmesi konusunda yapılan çalışmalar beş ana grup altında toplanabilir [3].

- 1- Direkt tasarım yöntemi
- 2- Endüktans benzeşim (simülasyonu) yöntemi
- 3- Kaskad biquadratik tasarım yöntemi
- 4- Frekansa bağlı negatif direnç yöntemi
- 5- Çok çevrimli geribesleme (Multiple - loop feedback (MF)) yöntemi

Bu yöntemlerden direkt tasarım yöntemi pratikte, filtre karakteristiğinin devre elemanlarındaki değişimlere karşı oldukça duyarlı olması nedeniyle hemen hemen hiç kullanılmamakta, sadece teorik olarak önem taşımaktadır. Endüktans benzeşim yönteminde yüksek dereceden filtrelerin gerçekleştirilmesi oldukça pahalıdır. Çünkü her bir L simülasyonu için değişik imalat gerekmektedir, yani devre modüler değildir. Kaskad biquadratik tasarım yönteminde ise filtrenin frekans cevabının geçirme bandında eleman değişimlerine karşı çok duyar olduğu, diğer frekans değerleri için ise duyarlığın çok az olduğu görülür. Frekansa bağlı negatif direnç yöntemi de endüktans benzeşim yöntemi gibi pahalı bir yöntemdir.

İşte quad tasarım yöntemi bu sayılan dezavantajları ortadan kaldıran uygulaması genel ve kolay, az duyarlıklı yeni bir tasarım yöntemidir. MF yönteminin özel bir hali olan quad tasarım yönteminde filtre cebirsel işlemlerle kolaylıkla tasarlanabilmektedir. Pratik açıdan devrenin gerçekleştirilmesinde kullanılan aktif devre elemanlarının teknolojik sınırlarını aşmayacak biçimde devre elemanları belirlenebilmektedir.

Tasarımda ileri besleme ve toplama devrelerine gerek duyulmamaktadır. Yöntem oldukça ekonomiktir. Filtre ayarı kolaydır.

Tezde, OTA-C elemanlarıyla alçak geçiren ve band geçiren filtrelerin tasarımı Butterworth, Chebyshev, Eliptik yaklaşıklık tipleri için hem quad hem de kaskad yöntemi kullanılarak gerçekleştirilmiştir.

Bölüm 2’de filtrelerin sınıflandırılması ve yaklaşıklık tipleri ele alınarak alçak geçiren ve band geçiren filtre transfer fonksiyonları üç yaklaşıklık tipi için ayrı ayrı elde edilmiştir.

Üçüncü bölümde quad yöntemi genel bir biçimde anlatılmıştır. Quad alt devre yapısı ve bunlara ilişkin tasarım denklemleri elde edilmiştir.

Dördüncü bölümde, transfer fonksiyonları Bölüm 2’de elde edilmiş olan alçak geçiren ve band geçiren filtreler için genel quad yöntemi yardımıyla quad ve kaskad denklemlerinin bulunması ayrıntılı bir biçimde ele alınmıştır.

Beşinci bölümde, OTA elemanı ve modelleri tanıtılıp, OTA-C elemanlarıyla oluşturulan biquadratik devreler incelenmektedir.

Altıncı bölümde, alçak geçiren ve band geçiren filtrelerin gerçekleştirilmesinde kullanılan alt devrelere ilişkin kazanç değerleri devrenin dinamik davranışını iyileştirecek biçimde belirlenmektedir. Ayrıca sonuç tasarım denklemleri yardımıyla OTA ve kapasite elemanlarının değerleri hesaplanmaktadır.

Tasarlanan filtrede clipping ve slew-rate problemleri ile karşılaşmamak için filtre girişine uygulanabilecek maksimum gerilim değerinin bilinmesi gerekmektedir. Bu amaçla, Bölüm 7’de OTA-C filtresinin girişine uygulanabilecek maksimum gerilim değerinin hesaplanması için izlenecek yol anlatılmaktadır.

Ek A’da alçak geçiren ve band geçiren filtre tasarım örnekleri ve bu örneklerle ait frekans karakteristikleri verilmiştir. Ek B’de alçak geçiren ve band geçiren filtreler için tasarım örnekleri ile frekans cevabı karakteristikleri Micro Cap Analiz programı

yardımıyla verilmiştir. Ek B’de ise alçak geçiren ve band geçiren filtrelerin tasarımını Butterworth, Chebyshev, Eliptik yaklaşıklık tipleri için quad ve kaskad yöntemini kullanarak, gerçekleyen bir bilgisayar programı verilmektedir.



BÖLÜM 2

ALÇAK GEÇİREN VE BAND GEÇİREN FİLTRELERİN TASARIMI

Filtreler, girişlerine elektriksel bir işaret uygulandığında transfer fonksiyonlarına bağlı olarak bu işaretin belirli bir frekans bandını geçiren diğer bandları istenilen değere göre zayıflatan devrelerdir. Bir filtrenin giriş - çıkış karakteristiği bu filtrenin transfer fonksiyonundan elde edilir.

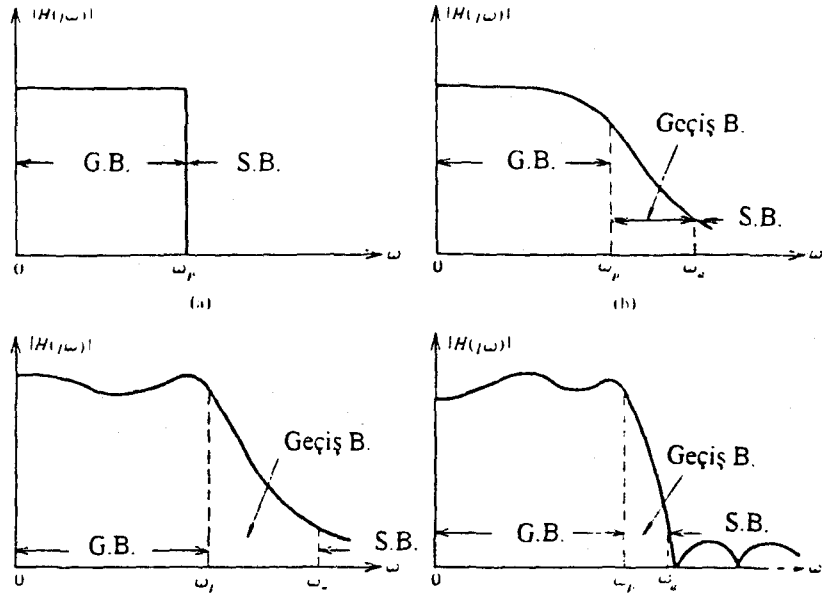
2.1 Filtrelerin Sınıflandırılması

Uygulamada dört tip filtre kullanılır. Bunlar alçak geçiren (AG), yüksek geçiren (YG), band geçiren (BG), band söndüren (BS) filtrelerdir. Şekil 2.1, Şekil 2.2, Şekil 2.3, Şekil 2.4'te bu filtrelere ait tipik frekans karakteristikleri gösterilmiştir. Her dört şekil için ilk karakteristikler ideal filtre karakteristiklerini, ikinci karakteristikler Butterworth, üçüncü karakteristikler Chebyshev, dördüncü karakteristikler Eliptik yaklaşım karakteristiklerini göstermektedir.

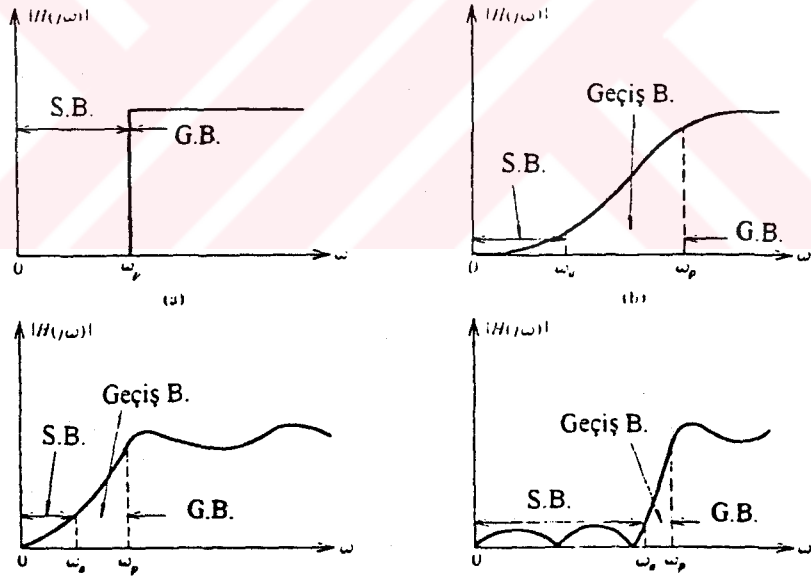
Frekans cevabının belirli bir değerin altına düştüğü frekans aralığına söndürme bandı (SB) denir. Zayıflatmadan geçen frekansların bandına da geçirme bandı (GB) denir.

2.2 Yaklaşıklık Tipleri

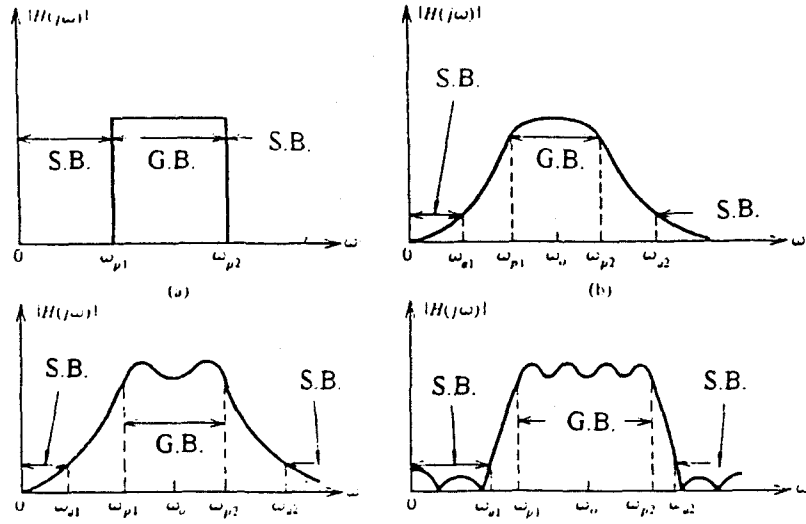
Filtre tasarımında, ilk aşamada istenilen özellikleri sağlayan transfer fonksiyonu bulunur. Verilen özelliklere göre AG, YG, BG, BS fonksiyonları eşdeğer özelliklere sahip prototip AG filtre fonksiyonuna dönüştürülür. Frekans dönüşümü kullanılarak prototip AG filtre fonksiyonu diğer tip filtre karakteristiklerine karşılık gelen fonksiyonlara dönüştürülebilir [4].



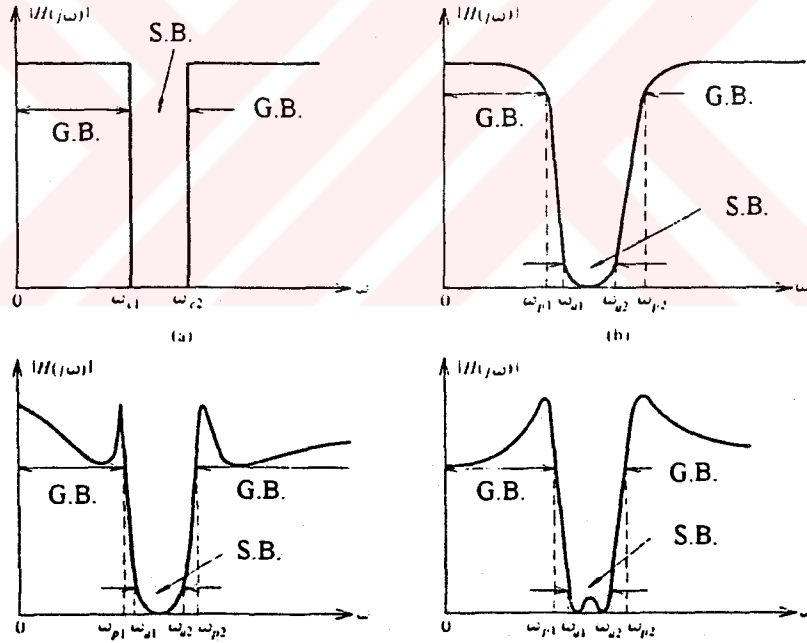
Şekil 2.1. AG filtre karakteristikleri



Şekil.2.2. YG filtre karakteristikleri



Şekil 2.3.BG filtre karakteristikleri



Şekil 2.4.BS filtre karakteristikleri

* G.B = Geçirme Bandı, S.B.= Söndürme Bandı ,Geçiş B. = Geçiş Bandı

Prototip AG filtreye ait özellikler şunlardır:

1. Maksimum geçirme bandı zayıflatması,

$$A_p, \quad 0 \leq \Omega \leq \Omega_p \text{ frekans bölgesinde}$$

2. Minimum söndürme bandı zayıflatması,

$$A_a, \quad \Omega \geq \Omega_a \text{ frekans bölgesinde}$$

Yaklaşıklık sorunu, verilen özellikleri sağlayan $H(p)$ rasyonel fonksiyonunun üç klasik yaklaşım olan Butterworth, Chebyshev, Eliptik yaklaşık ile bulunması sorunudur [4].

Butterworth yaklaşıklığı

Prototip AG karesel genlik fonksiyonu,

$$|H(j\Omega)|^2 = T(\Omega^2) = \frac{H_o^2}{1 + \varepsilon^2 \Omega^{2N}} \quad (2.1)$$

biçiminde verilebilir. $\Omega=0$ da $|H(j\Omega)|$, H_o maksimum değerini alır. ε büyüklüğü salınım (ripple) faktörüdür. N ise filtre transfer fonksiyonu $H(s)$ 'in derecesidir. ε ve N verilen özelliklerden yararlanılarak bulunur. Önce,

$$A(\Omega) = 10 \log (1 + \varepsilon^2 \Omega^{2N}) - 20 \log (H_o) \quad (2.2)$$

zayıflatması bulunur. H_o kazanç sabiti olarak bilinir. $H_o=1$ seçildiğinde,

$$A(\Omega) = 10 \log(1 + \varepsilon^2 \Omega^{2N}) \quad (2.3)$$

olur. Normalize filtre tasarımında $\Omega_p = 1$ r/s olarak alınır. (2.3) bağıntısında $\Omega_p = 1$ r/s ve $A(\Omega)=A_p$ yerine konulduğunda,

$$\varepsilon = (10^{0.1 A_p} - 1)^{1/2} \quad (2.4)$$

bağıntısı elde edilebilir. $\Omega \geq \Omega_a$ için minimum söndürme bandı zayıflatması,

$$A_a \leq 10 \log (1 + \varepsilon^2 \Omega^{2N}) \quad (2.5)$$

şeklindedir. (2.4) bağıntısı (2.5)'de yerine konulursa filtre derecesi,

$$N \geq \frac{\log\left(\left(10^{0.1A_a} - 1\right) / \left(10^{0.1A_p} - 1\right)\right)}{2 \log(\Omega_a)} \quad (2.6)$$

eşitsizliğinden bulunabilir. Burada N bir tamsayıdır. Yani N, (2.6) ifadesinin sayısal değerinin kesirli tarafı sıfırdan farklı ise tam kısmın bir fazlasına eşit, sıfır ise tam kısmına eşittir. (2.1) ifadesinde $\Omega = p/j$ yerine konulduğunda, elde edilen karakteristik denklemin kökleri $T(-p^2)$ 'nin kutupları olmak zorundadır ve,

$$p_k = R \left[\cos\left(\frac{\pi}{2} + \frac{2k-1}{N} \frac{\pi}{2}\right) + j \sin\left(\frac{\pi}{2} + \frac{2k-1}{N} \frac{\pi}{2}\right) \right] \quad k = 1, 2, \dots, N \quad (2.7)$$

yazılabilir. Burada $R = \varepsilon^{-1/N}$ dir [7] .

$$p_k, p_{N+1-k} = \text{Re}(p_k) \pm j |\text{im}(p_k)| \quad k = 1, 2, \dots, n_x \quad (2.8.a)$$

$$n_x = \frac{N - (N) \bmod 2}{2} \quad (2.8.b)$$

(2.8) ifadesinden anlaşılaçağı gibi p_k ve p_{N+1-k} kutupları eşlenik iki kutuptur. Böylece transfer fonksiyonu ,

$$H(p) = \frac{H_0}{\prod_{k=1}^N (p - p_k)} \quad (2.9)$$

biçiminde elde edilir. (2.9) ifadesi kullanılarak normalize (prototip) filtre fonksiyonu bulunabilir. Gerçekleme açısından karmaşık eşlenik kutupları birleştirmemiz gerekir. Böylece prototip AG filtre fonksiyonu,

$$H(p) = \frac{H_o}{D_o(p) \prod_{i=1}^{nx} (p^2 + A_i p + B_i)} \quad (2.10.a)$$

olarak bulunur. Burada,

$$D_o(p) = \begin{cases} p + \sigma_o & , N \text{ tek} \\ 1 & , N \text{ çift} \end{cases} \quad (2.10.b)$$

$$\sigma_o = |p_{((N+1)/2)}| \quad (2.10.c)$$

Ayrıca A_i , B_i ve H_o için,

$$A_i = -2 \operatorname{Re}(p_i) \quad i = 1, 2, \dots, nx \quad (2.10.d)$$

$$B_i = |p_i|^2 \quad i = 1, 2, \dots, nx \quad (2.10.e)$$

$$H_o = \prod_k^N |p_k| \quad (2.10.f)$$

ifadeleri yazılabilir [7] .

Chebyshev yaklaşıklığı

Bu yaklaşıklık için karesel genlik fonksiyonu,

$$|H(j\Omega)|^2 = T(\Omega^2) = \frac{H_o^2}{1 + \epsilon^2 C_N^2(\Omega)} \quad (2.11)$$

şeklindedir. $C_N(\Omega)$ N.dereceden Chebyshev polinomudur ve

$$C_N(\Omega) = \begin{cases} \cos(N \cos^{-1}(\Omega)) & , 0 \leq \Omega \leq 1 \\ \cosh(N \cosh^{-1}(\Omega)) & , \Omega \geq 1 \end{cases} \quad (2.12)$$

bağıntısıyla tanımlanır.

Prototip AG filtre için tekrar $\Omega_p = 1$ r/s kabulü ile (2.12) ifadesi (2.11)'de yerine konulursa,

$$A_p = 10 \log (1 + \epsilon^2) \quad (2.13)$$

elde edilir. Bu bağıntıda $H_0 = 1$ kabul edilmiştir. Ayrıca $\Omega \geq \Omega_a$ için

$$A_a \leq 10 \log (1 - \epsilon^2 \cosh^2 (N \cosh^{-1} \Omega_a)) \quad (2.14)$$

olur. Bu eşitsizlikten N çekilirse,

$$N \geq \frac{\cosh^{-1} (((10^{0.1A_a} - 1) / (10^{0.1A_p} - 1))^{0.5})}{\cosh^{-1} \Omega_a} \quad (2.15)$$

elde edilir. ϵ ve N bulunduktan sonra transfer fonksiyonunun kutupları belirlenebilir.

$$v = \frac{1}{N} \sinh^{-1} \left(\frac{1}{\epsilon} \right) \quad (2.16)$$

olmak üzere kutuplar,

$$p_k = -\sin\left(\frac{2k-1}{N} \frac{\pi}{2}\right) \sinh(v) + j \cos\left(\frac{2k-1}{N} \frac{\pi}{2}\right) \cosh(v) \quad k = 1, 2, \dots, N, \quad (2.17)$$

şeklinde ifade edilebilir. Butterworth yaklaşıklık tipinde olduğu gibi eşlenik kutuplar birleştirildiğinde transfer fonksiyonu (2.10) formuyla aynı olur [7] . Sadece H_o 'da aşağıdaki gibi değişiklik olur.

$$H_o = \begin{cases} \prod_k^N |p_k| & , N \text{ tek} \\ 10^{-0.05Ap} \prod_k^N |p_k| & , N \text{ çift} \end{cases} \quad (2.18)$$

(2.8) bağıntısı Chebyshev yaklaşıklık için de geçerlidir.

Eliptik yaklaşıklık

Yaklaşıklık tipleri içerisinde en verimli olanı Eliptik yaklaşıklıktır. Eliptik yaklaşıklıkta en düşük dereceli transfer fonksiyonu elde edildiğinden kullanılacak olan elemansayısı en aza indirgenmiş olur. Bununla birlikte her üç yaklaşımın kendine göre avantajlı tarafı vardır. Örneğin, geçirme bandında herhangi bir salınım istenilmediğinde Butterworth yaklaşımını kullanmak gerekir.

Eliptik yaklaşıklık için prototip AG filtre fonksiyonu,

$$H(p) = \frac{H_o}{D_o(p)} \prod_{i=1}^{nx} \frac{p^2 + C_i}{p^2 + A_i p + B_i} \quad (2.19)$$

biçimindedir. Burada nx ve $D_o(p)$ ifadeleri (2.8) ve (2.10.b) ifadeleri ile aynıdır. Ω_p ve Ω_a sırasıyla prototip AG filtrenin geçirme bandı ve söndürme bandı kenar frekanslarıdır. A_p ve A_a ise sırasıyla maksimum geçirme bandı ve minimum söndürme bandı zayıflatmasıdır.

Aşağıdaki algoritma eliptik yaklaşım için gerekli derece ve katsayıları bulmakta kullanılabilir [7] .

1. Seçicilik faktörü k hesaplanır,

$$k = \frac{\Omega_p}{\Omega_s} \quad (2.20.a)$$

2. Aşağıdaki bağıntılar hesaplanır,

$$k' = \sqrt{1 - k^2}$$

$$q_o = \frac{1}{2} \frac{1 - \sqrt{k'}}{1 + \sqrt{k'}}$$

$$q = q_o + 2q_o^5 + 15q_o^9 + 150q_o^{13}$$

$$D = \frac{10^{0.1A_a} - 1}{10^{0.1A_p} - 1} \quad (2.20.b)$$

3. Fitre derecesi tamsayı olacak şekilde hesaplanır,

$$N \geq \frac{\log 16D}{\log(1/q)} \quad (2.20.c)$$

4. Sırasıyla aşağıdaki yardımcı terimler hesaplanır,

$$D_1 = \frac{10^{0.05A_p} + 1}{10^{0.05A_p} - 1}$$

$$\lambda = \frac{\ln D_1}{2N}$$

$$\sigma_o = \frac{2q^{0.25} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sinh((2m+1)\lambda)}{1 + 2 \sum_{m=1}^{\infty} (-1)^m q^{m^2} \cosh(2m\lambda)}$$

$$W = (1 + k\sigma_o^2)(1 + \frac{\sigma_o^2}{k}) \quad (2.20.d)$$

5. A_i, B_i, C_i katsayıları aşağıdaki bağıntılar yardımıyla hesaplanır,

$$\mu = \begin{cases} i & N \text{ tek} \\ i - 0.5 & N \text{ çift} \end{cases} \quad i = 1, 2, 3, \dots, nx$$

$$\Omega_i = \frac{2q^{0.25} \sum_{m=1}^{\infty} (-1)^m q^{m(m+1)} \sin((2m+1)\mu\pi / N)}{1 + 2 \sum_{m=0}^{\infty} (-1)^m q^{m^2} \cos(2m\mu\pi / N)}$$

$$v_i = (1 - k\Omega_i^2)(1 - \frac{\Omega_i^2}{k})$$

$$C_i = \frac{1}{\Omega_i^2}$$

$$B_i = \frac{v_i \sigma_o^2 + w\Omega_i^2}{(1 + \sigma_o^2 \Omega_i^2)^2}$$

$$A_i = \frac{2\sigma_o \sqrt{v_i}}{1 + \sigma_o^2 \Omega_i^2} \quad (2.20.e)$$

6. Kazanç sabiti minimum zayıflatma 0 dB olacak şekilde bulunur,

$$H_o = \begin{cases} \sigma_o \prod_{i=1}^{nx} (B_i / C_i) & N \text{ tek} \\ 10^{0.05Ap} \prod_{i=1}^{nx} (B_i / C_i) & N \text{ çift} \end{cases} \quad (2.20.f)$$

2.3. Alçak Geçiren Filtre Transfer Fonksiyonunun Elde Edilmesi

Prototip AG fonksiyonundan yararlanarak, verilen özellikleri sağlayan AG filtre için transfer fonksiyonu aşağıda verilen adımlarla elde edilebilir.

1. $\Omega_a = w_a / w_p$ hesaplanır.
2. Verilen A_p , A_a ve hesaplanan Ω_a değeri kullanılarak ve Butterworth, Chebyshev Eliptik yaklaşıklardan biri seçilerek $H(p)$ prototip AG filtre fonksiyonu oluşturulur.
3. Prototip AG filtre fonksiyonunda,

$$p = \frac{s}{w_p} \quad (2.21)$$

yerine konularak gerçek özellikleri sağlayan $H(s)$, AG filtre fonksiyonu elde edilir. Bu dönüşümün kullanılmasının nedeni prototip AG filtrenin geçirme bandı köşe frekansı w_p ' yi gerçek frekansına kaydırmaktır.

2.3.1 Prototip alçak geçiren - alçak geçiren frekans dönüşümü

$p = s/w_p$ dönüşüm formülü kullanılarak verilen özelliklere sahip AG transfer fonksiyonunun elde edilmesi çeşitli yaklaşım tipleri için aşağıda anlatılmıştır.

1. Butterworth ve Chebyshev için :

(2.21) dönüşüm formülü (2.10)'da yerine konulduğunda,

a) N tek ise

$$H(s) = \frac{H_o}{\left(\frac{s}{w_p} + \sigma_0\right) \prod_{i=1}^{nx} \left(\left(\frac{s}{w_p}\right)^2 + A_i \frac{s}{w_p} + B_i\right)}$$

$$H(s) = \frac{H_o}{\left(\frac{s + \sigma_0 w_p}{w_p}\right) \prod_{i=1}^{nx} \frac{1}{w_p^2} (s^2 + A_i w_p s + B_i w_p^2)}$$

$$H(s) = H_o \frac{w_p}{s + \sigma_0 w_p} \prod_{i=1}^{nx} \frac{w_p^2}{(s^2 + A_i w_p s + B_i w_p^2)} \quad (2.22)$$

bulunur. Bu durumda yine iki olasılık ortaya çıkıyor. Bunlar nx'in tek veya çift olması durumlarıdır.

i) nx tek ise,

$$H(s) = H_o \frac{w_p}{(s + \sigma_0 w_p)} \frac{w_p^2}{(s^2 + A_1 w_p s + B_1 w_p^2)} \prod_{i=1}^{tx} \frac{w_p^4}{D(s, k(i))} \quad (2.23.a)$$

bulunur. Bu bağıntıda tx ve D(s,k(i)),

$$tx = \frac{nx - (nx) \bmod 2}{2} \quad (2.23.b)$$

$$k(i) = 2i \quad (2.23.c)$$

$$D(s, k(i)) = s^4 + [A_{k(i)} + A_{k(i)+1}] w_p s^3 + [A_{k(i)} A_{k(i)+1} + B_{k(i)} + B_{k(i)+1}] w_p^2 s^2 + [A_{k(i)} B_{k(i)+1} + B_{k(i)} A_{k(i)+1}] w_p^3 s + B_{k(i)} B_{k(i)+1} w_p^4 \quad (2.23.d)$$

şeklindedir.

$$T_a(s) = \frac{w_p}{s + \sigma_0 w_p} \quad (2.23.e)$$

$$T_b(s) = \frac{w_p^2}{s^2 + A_1 w_p s + B_1 w_p^2} \quad (2.23.f)$$

$$T_{Qi}(s) = \frac{w_p^4}{D(s, k(i))} \quad (2.23.g)$$

ifadeleri (2.23.a) bağıntısında yerine konulursa,

$$H(s) = H_o T_a(s) T_b(s) \prod_{i=1}^{tx} T_{Qi}(s) \quad (2.24)$$

elde edilir.

ii) nx çift ise (2.24) bağıntısı

$$H(s) = H_o T_a(s) \prod_{i=1}^{tx} T_{Qi}(s) \quad (2.25)$$

şeklini alır.

b) N çift ise,

$$H(s) = \frac{H_o}{\prod_{i=1}^{nx} \left(\left(\frac{s}{w_p} \right)^2 + A_i \frac{s}{w_p} + B_i \right)}$$

$$H(s) = \frac{H_o}{\prod_{i=1}^{nx} (1/w_p^2)(s^2 + A_i w_p s + B_i w_p^2)}$$

$$H(s) = H_o \prod_{i=1}^{nx} \frac{w_p^2}{(s^2 + A_i w_p s + B_i w_p^2)} \quad (2.26)$$

bulunur.

i) nx tek ise, H(s)

$$H(s) = H_o \frac{w_p^2}{(s^2 + A_1 w_p s + B_1 w_p^2)} \prod_{i=1}^{tx} \frac{w_p^4}{D(s, k(i))} \quad (2.27.a)$$

şeklindedir. Bu ifadede, tx'in ve D(s,k(i))'nin açık ifadesi sırasıyla (2.27.b) ve (2.27.d) bağıntılarıyla verilebilir.

$$tx = \frac{nx - (nx) \bmod 2}{2} \quad (2.27.b)$$

$$k(i) = 2i - 1 \quad (2.27.c)$$

$$D(s, k(i)) = s^4 + [A_{k(i)} + A_{k(i)+1}] w_p s^3 + [A_{k(i)} A_{k(i)+1} + B_{k(i)} + B_{k(i)+1}] w_p^2 s^2 + [A_{k(i)} B_{k(i)+1} + B_{k(i)} A_{k(i)+1}] w_p^3 s + B_{k(i)} B_{k(i)+1} w_p^4 \quad (2.27.d)$$

$H(s)$, (2.28.a) bağıntısındaki gibi daha kısa bir şekilde ifade edilebilir.

$$H(s) = H_o T_b(s) \prod_{i=1}^{tx} T_{Q_i}(s) \quad (2.28.a)$$

Bu ifadede,

$$T_b(s) = \frac{w_p^2}{s^2 + A_1 w_p s + B_1 w_p^2} \quad (2.28.b)$$

$$T_{Q_i}(s) = \frac{w_p^4}{D(s, k(i))} \quad (2.28.c)$$

ii) nx çift ise (2.28.a) bağıntısı

$$H(s) = H_o \prod_{i=1}^{tx} T_{Q_i}(s) \quad (2.29)$$

şeklini alır.

2.Eliptik için

(2.21) dönüşüm formülü (2.19) bağıntısında yerine konulduğunda,

a) N tek ise

$$H(s) = \frac{H_o}{\left(\frac{s}{w_p} + \sigma_0\right)} \prod_{i=1}^{nx} \frac{(s/w_p)^2 + C_i}{(s/w_p)^2 + A_i \frac{s}{w_p} + B_i}$$

$$H(s) = H_o \frac{w_p}{s + \sigma_0 w_p} \prod_{i=1}^{nx} \frac{(1/w_p^2)(s^2 + C_i w_p^2)}{(1/w_p^2)(s^2 + A_i w_p s + B_i w_p^2)}$$

$$H(s) = H_o \frac{w_p}{s + \sigma_0 w_p} \prod_{i=1}^{nx} \frac{s^2 + C_i w_p^2}{(s^2 + A_i w_p s + B_i w_p^2)} \quad (2.30)$$

bulunur. (2.30) bağıntısı aşağıdaki gibi iki durumda incelenebilir.

i) nx tek ise

$$H(s) = H_o \frac{w_p}{s + \sigma_0 w_p} \frac{s^2 + C_1 w_p^2}{(s^2 + A_1 w_p s + B_1 w_p^2)} \prod_{i=1}^{tx} \frac{N(s, k(i))}{D(s, k(i))} \quad (2.31)$$

bulunur. Bu bağıntıdaki tx ve D(s,k(i)) (2.23.b), (2.23.c) ve (2.23.d) bağıntılarıyla elde edilebilir.

$$T_a(s) = \frac{w_p}{s + \sigma_0 w_p} \quad (2.32.a)$$

$$T_b(s) = \frac{s^2 + C_1 w_p^2}{s^2 + A_1 w_p s + B_1 w_p^2} \quad (2.32.b)$$

$$N(s, k(i)) = s^4 + (C_{k(i)} + C_{k(i)+1}) w_p^2 s^2 + C_{k(i)} C_{k(i)+1} w_p^4 \quad (2.32.c)$$

$$T_{Qi}(s) = \frac{N(s, k(i))}{D(s, k(i))} \quad (2.32.d)$$

olmak üzere H(s) (2.32.d) bağıntısındaki gibi ifade edilebilir.

$$H(s) = H_o T_a(s) T_b(s) \prod_{i=1}^{tx} T_{Qi}(s) \quad (2.32.e)$$

ii) nx çift ise (2.32.e) ifadesi

$$H(s) = H_o T_a(s) \prod_{i=1}^{nx} T_{Qi}(s) \quad (2.33)$$

şeklini alır.

b) N çift ise

$$H(s) = H_o \prod_{i=1}^{nx} \frac{(s/w_p)^2 + C_i}{(s/w_p)^2 + A_i(s/w_p) + B_i}$$

$$H(s) = H_o \prod_{i=1}^{nx} \frac{(1/w_p^2)(s^2 + C_i w_p^2)}{(1/w_p^2)(s^2 + A_i w_p s + B_i w_p^2)}$$

$$H(s) = H_o \prod_{i=1}^{nx} \frac{s^2 + C_i w_p^2}{(s^2 + A_i w_p s + B_i w_p^2)} \quad (2.34)$$

i) nx tek ise

$$H(s) = H_o \frac{s^2 + C_1 w_p^2}{(s^2 + A_1 w_p s + B_1 w_p^2)} \prod_{i=1}^{tx} \frac{N(s, k(i))}{D(s, k(i))} \quad (2.35)$$

elde edilir. Bu bağıntıdaki, tx ve D(s,k(i)) (2.27.b), (2.27c) ve (2.27.d) ifadeleriyle bulunabilir. N(s, k(i)) ise (2.32.c) bağıntısıyla aynıdır.

$$T_b(s) = \frac{s^2 + C_1 w_p^2}{s^2 + A_1 w_p s + B_1 w_p^2} \quad (2.36.a)$$

$$T_{Qi}(s) = \frac{N(s, k(i))}{D(s, k(i))} \quad (2.36.b)$$

olmak üzere $H(s)$ için

$$H(s) = H_o T_b(s) \prod_{i=1}^{n_x} T_{Qi}(s) \quad (2.36.c)$$

ifadesi yazılabilir.

ii) n_x çift ise, (2.36.c) ifadesi

$$H(s) = H_o \prod_{i=1}^{n_x} T_{Qi}(s) \quad (2.37)$$

şeklini alır.

2.4 Band Geçiren Filtre Transfer Fonksiyonunun Elde Edilmesi

Verilen özellikleri sağlayan BG filtre transfer fonksiyonu aşağıda verilen adımlarla bulunabilir [4] :

1. $A_a = \max [A_{a1}, A_{a2}]$ ve $A_p = A_p$ olarak seçilir.
2. Band genişliği ve merkez frekans,

$$\omega_o = \sqrt{\omega_{p2} \omega_{p1}}$$

$$B = \omega_{p2} - \omega_{p1}$$

bağıntıları yardımıyla hesaplanır.

3. $\omega_a = \omega_o^2 / \omega_{a1}$ değeri hesaplanır.

4. $\omega_a \leq \omega_{a2}$ ise $\omega_{a2} = \omega_{a2}$ ve $\omega_{a1} = \omega_o^2 / \omega_{a2}$ olarak seçilir.

5. Bulunan ω_{a1} ve ω_{a2} ile

$$\Omega_a = \frac{\omega_{a2} - \omega_{a1}}{\omega_{p2} - \omega_{p1}} \quad (2.38)$$

değeri hesaplanır.

6. Butterworth, Chebyshev veya Eliptik yaklaşıklardan biri seçilir ve elde edilen A_s Ω_s ve verilen A_p kullanılarak $H(p)$, prototip AG filtre fonksiyonu oluşturulur.
7. Gerçek özellikleri sağlayan BG filtre transfer fonksiyonu $H(s)$,

$$p = \frac{s^2 + w_o^2}{Bs} \quad (2.39)$$

dönüşüm formülü $H(p)$ prototip AG fonksiyonuna uygulanarak bulunur [7].

2.4.1 Prototip alçak geçiren-band geçiren dönüşümü

Butterworth, Chebyshev ve Eliptik yaklaşım tipleri için (2.39) dönüşüm formülü kullanılarak orginal BG filtre transfer fonksiyonunun elde edilişi aşağıda anlatılmaktadır.

1. Butterworth ve Chebyshev için :

(2.39) dönüşüm formülü (2.10) bağıntısında yerine konulursa,

a) N tek ise,

$$H(s) = \frac{H_o}{\left(\frac{s^2 + w_o^2}{Bs} + \sigma_o\right) \prod_{i=1}^{nx} \left(\left(\frac{s^2 + w_o^2}{Bs}\right)^2 + A_i \left(\frac{s^2 + w_o^2}{Bs}\right) + B_i\right)}$$

$$H(s) = H_o \frac{Bs}{(s^2 + (B\sigma_o)s + w_o^2)} \prod_{i=1}^{nx} \frac{B^2 s^2}{(s^4 + (A_i B)s^3 + (2w_o^2 + B_i B^2)s^2 + (A_i B w_o^2)s + w_o^4)} \quad (2.40)$$

$$H(s) = H_o T_b(s) \prod_{i=1}^{nx} T_{Qi}(s) \quad (2.41)$$

bulunur. Bu ifadede nx , (2.8) ifadesi ile hesaplanabilir.

b) N çift ise (2.41)ifadesi

$$H(s) = H_o \prod_{i=1}^{nx} T_{Qi}(s) \quad (2.42)$$

şeklini alır.

2. Elliptik için :

(2.39) dönüşüm formülü (2.19) bağıntısında yerine konulduğunda,

a) N tek ise,

$$H(s) = H_o \frac{Bs}{(s^2 + (B\sigma_o)s + w_o^2)} \prod_{i=1}^{nx} \frac{(s^4 + (2w_o^2 + C_i B^2)s^2 + w_o^4)}{(s^4 + (A_i B)s^3 + (2w_o^2 + B_i B^2)s^2 + (A_i B w_o^2)s + w_o^4)} \quad (2.43)$$

$$H(s) = H_o T_b(s) \prod_{i=1}^{nx} T_{Qi}(s) \quad (2.44)$$

bulunur.

b) N çift ise, (2.44) bağıntısındaki $T_b(s)$ çarpanı ortadan kalkar. Yani

$$H(s) = H_o \prod_{i=1}^{nx} T_{Qi}(s) \quad (2.45)$$

ifadesi elde edilir.

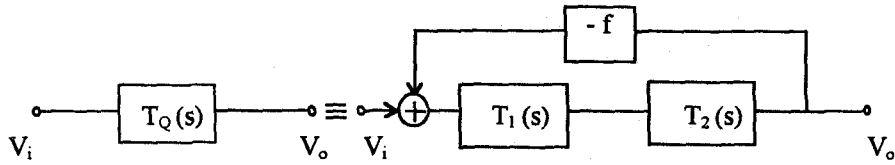
BÖLÜM 3

QUAD YÖNTEMİ

Bu bölümde az duyarlıklı kolayca ayarlanabilen, uygulaması özellikle simetrik devreler için çok basit, ekonomik, iyi dinamik davranış gösteren ve hiçbir sınırlama getirmeden gerçekleştirilmesi olası her türden filtrenin tasarımına olanak veren bir yöntem anlatılacaktır. Bu yöntem, Çok Çevrimli Geribesleme (MF) Yönteminin özel bir hali olan Quad yöntemidir [3]. Yüksek dereceden alçak geçiren ve band geçiren filtrelerin OTA elemanlarıyla tasarımında bu yöntemden yararlanılacaktır.

3.1 Quad Devre Yapısı

Yöntemin esasını oluşturan quad alt devresi birbirine kaskad olarak bağlanmış iki biquadratik alt devre ve bunlara uygulanan bir negatif geri besleme ile oluşturulmaktadır [3]. Şekil 3.1’de quad devre yapısı blok şema halinde gösterilmiştir.



Şekil 3.1 Quad devre yapısı

Bu devre yapısına ilişkin gerilim transfer fonksiyonu,

$$T_Q(s) = \frac{T_1(s)T_2(s)}{1 + fT_1(s)T_2(s)} \quad (3.1)$$

biçimindedir. Bu bağıntıda $i = 1, 2$ olmak üzere,

$$T_i(s) = \frac{(b_{i2}s^2 + b_{i1}s + b_{i0})}{(s^2 + a_{i1}(s) + a_{i0})} \quad (3.2.a)$$

ya da

$$T_i(s) = \frac{k_i(s^2 + (w_{zi}/Q_{zi})s + w_{zi}^2)}{(s^2 + (w_{pi}/Q_{pi})s + w_{pi}^2)} \quad (3.2.b)$$

biçiminde yazılabilir.

Dördüncü dereceden Quad devresine ilişkin gerilim transfer fonksiyonu (3.3) bağıntısında olduğu gibi en genel biçimiyle ifade edilebilir.

$$T_Q(s) = \frac{b_4s^4 + b_3s^3 + b_2s^2 + b_1s + b_0}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \quad (3.3)$$

Şekil 3.1'den yararlanılarak,

$$T(s) = T_1(s)T_2(s) = \frac{T_Q(s)}{1 - fT_Q(s)} \quad (3.4)$$

yazılabilir. (3.3) ve (3.4) denklemleri kullanılarak,

$$T(s) = T_1(s)T_2(s) = \frac{1}{(1 - fb_4)} \frac{b_4s^4 + b_3s^3 + b_2s^2 + b_1s + b_0}{(s^4 + a_{m3}s^3 + a_{m2}s^2 + a_{m1}s + a_{m0})} \quad (3.5)$$

bağıntısı yazılabilir. Bu ifadede geçen katsayılar,

$$a_{m3} = \frac{a_3 - fb_3}{1 - fb_4} \quad (3.6.a)$$

$$a_{m2} = \frac{a_2 - fb_2}{1 - fb_4} \quad (3.6.b)$$

$$a_{m1} = \frac{a_1 - fb_1}{1 - fb_4} \quad (3.6.c)$$

$$a_{m0} = \sqrt{\frac{a_0 - fb_0}{1 - fb_4}} \quad (3.6.d)$$

biçiminde ifade edilebilir. Öte yandan, denklem (3.6) kullanılarak aşağıdaki ifadeler elde edilebilir.

$$\frac{a_3 - fb_3}{1 - fb_4} = \frac{w_1}{Q_1} + \frac{w_2}{Q_2} \quad (3.7.a)$$

$$\frac{a_2 - fb_2}{1 - fb_4} = \frac{w_1 w_2}{Q_1 Q_2} + w_1^2 + w_2^2 \quad (3.7.b)$$

$$\frac{a_1 - fb_1}{1 - fb_4} = w_1 w_2 (w_1 / Q_2 + w_2 / Q_1) \quad (3.7.c)$$

$$\sqrt{\frac{a_0 - fb_0}{1 - fb_4}} = w_1 w_2 \quad (3.7.d)$$

Yukarıdaki denklem takımının tek çözümünün var olabilmesi için (3.7) ifadesindeki tüm katsayıların (a_i , b_i) bilinmesi ve w_1 , w_2 , Q_1 , Q_2 , f değişkenlerinden birinin bağımsız olarak seçilmesi gerekir. (3.7) ifadesinde geçen w_1 , w_2 , Q_1 , Q_2 sırasıyla w_{p1} , w_{p2} , Q_{p1} , Q_{p2} 'ye eşittir.

3.1.1 Quad devresinin gerçekleştirilmesi

Geri besleme katsayısı olan f 'in şimdilik bilindiği ve gerçel bir sayı olduğu varsayılırsa (3.7) denklemlerinden yararlanılarak biquadratik alt devrelere ilişkin w_1 , w_2 , Q_1 , Q_2 değerleri aşağıdaki biçimde hesaplanabilir.

$$w_1 = \sqrt{\frac{y \pm \sqrt{y^2 - 4a_{m0}^2}}{2}} \quad (3.8.a)$$

$$w_2 = \frac{a_{m0}}{w_1} \quad (3.8.b)$$

Burada $y > 2a_{m0}$ olmalı ve aşağıdaki denklemi sağlamalıdır [3].

$$y^3 - a_{m2}y^2 + (a_{m1}a_{m3} - 4a_{m0}^2)y + (4a_{m2}a_{m0}^2 - a_{m1}^2 - a_{m0}^2a_{m3}^2) = 0 \quad (3.8.c)$$

iyilik faktörleri ise,

$$Q_1 = \frac{w_1^4 - a_{m0}^2}{w_1(w_1^2 a_{m3} - a_{m1})} \quad (3.9.a)$$

$$Q_2 = \frac{(w_1^4 - a_{m0}^2)a_{m0}}{w_1(w_1^2 a_{m1} - a_{m3}a_{m0}^2)} \quad (3.9.b)$$

biçiminde hesaplanabilir.

Özel Durum

$a_{m1} = a_{m0} a_{m3}$ koşulu sağlanıyorsa (3.8.c)'den $y = 2a_{m0}$ elde edilir. Diğer bir deyişle ya $w_1 = w_2$ ya da $Q_1 = Q_2$ dir.

(i) $w_1 = w_2$ için radyal frekanslar ve iyilik faktörleri,

$$Q_1; Q_2 = \frac{a_{m1}}{2(a_{m2} - 2a_{m0})\sqrt{a_{m0}}} \left[1 \pm \sqrt{1 - \frac{4a_{m0}^2(a_{m2} - 2a_{m0})}{a_{m1}^2}} \right] \quad (3.10.b)$$

biçiminde elde edilebilir.

(ii) $Q_1 = Q_2$ olduğu varsayılırsa iyilik faktörleri ve w_1, w_2 değerleri aşağıdaki biçimde ifade edilebilir.

$$Q_1 = Q_2 = Q = \sqrt{\frac{(2a_{m0} + a_{m2}) + \sqrt{(a_{m2} + 2a_{m0})^2 - 4a_{m0}a_{m3}^2}}{2a_{m3}^2}} \quad (3.11.a)$$

$$w_1, w_2 = \frac{Qa_{m3} \pm \sqrt{a_{m3}^2 Q^2 - 4a_{m0}}}{2} \quad (3.11.b)$$

Diğer yandan, $a_{m1} = a_{m0} a_{m3}$ koşulu ile birlikte $a_{m3}^2 = 4(a_{m2} - a_{m0})$ koşulunun da sağlanması durumunda alt devrelere ilişkin açısal frekanslar ve iyilik faktörleri için

$$w_1 = w_2 = \sqrt{a_{m0}} \quad (3.12.a)$$

$$Q_1 = Q_2 = \frac{2\sqrt{a_{m0}}}{a_{m3}} \quad (3.12.b)$$

ifadeleri yazılabilir. Bu durumda iyilik faktörleri en küçük değerlerini alırlar. İyilik faktörleri küçük olan devrelerin genellikle daha az duyarlıklı devreler olduğu söylenebilir.

3.1.2 Quad bloğu geribesleme katsayısının belirlenmesi

Geribesleme katsayılarının belirlenmesinde, eşit radyal frekans ve eşit iyilik faktörlerinin elde edilebilmesi ilkesinden yararlanılacaktır. Bu biçimde tasarım, özellik bakımından birbirlerine yakın, imalatı kolay, kolayca ayarlanabilen ve az duyarlıklı filtreler gerçekleştirilmesine olanak sağlayacaktır [3].

Aşağıdaki denklemleri aynı anda sağlayabilen geri besleme katsayısı bulunabildiğinde, ilgilenilen devre için optimum çözüm elde edilmiş olur.

$$a_{m1} = a_{m3} a_{m0} \quad (3.13.a)$$

$$a_{m3}^2 = 4(a_{m2} - a_{m0}) \quad (3.13.b)$$

(3.13) denklemlerinin en genel durumda aynı anda sağlanabilmesi olası olmadığından bu ifadelerin her biri için ayrı ayrı geribesleme katsayıları bulunabilir. Bu koşulları ayrı ayrı sağlayan geribesleme katsayılarından isteğe göre, ya daha az duyarlıklı devreyi gerçekleştiren, ya iyilik faktörleri daha küçük olan ya da daha iyi dinamik davranış gösteren tasarıma karşı düşeni geribesleme katsayısı olarak alınır ve bu değere uygun olarak devre gerçekleştirilir.

(3.13) denklemlerini ayrı ayrı sağlayan geribesleme katsayıları aşağıda açıklanan biçimde bulunabilir.

i) $a_{m1} = a_{m3}$ koşulunu sağlayan geribesleme katsayıları :

$$\frac{a_1 - fb_1}{a_3 - fb_3} = \sqrt{\frac{a_0 - fb_0}{1 - fb_4}} \quad (3.14)$$

denklemden yararlanılarak bulunabilir. (3.13.a) deklemini, aşağıdaki denkleme indirgenerek, geribesleme değeri,

$$(b_0b_3^2 - b_1^2b_4) f^3 + (2a_1b_1b_4 + b_1^2 - a_0b_3^2 - 2a_3b_3b_0) f^2 + (2a_0a_3b_3 + a_3^2b_0 - 2a_1b_1 - a_1^2b_4) f + (a_1^2 - a_0a_3^2) = 0 \quad (3.15)$$

denklemden hesaplanabilir. $b_1 = b_3 = b_4 = b_0 = 0$ olması halinde yukarıdaki denklem kullanılamaz. (3.13.b) denklemine başvurmak gerekir.

ii) $a_{m3}^2 = 4(a_{m2} - 2a_{m0})$ koşulundan yararlanarak geribesleme katsayısının belirlenmesi:

Bu durumda geribesleme katsayısı,

$$\frac{(a_3 - fb_3)^2}{(1 - fb_4)^2} = 4 \left[\frac{a_2 - fb_2}{1 - fb_4} - 2 \sqrt{\frac{a_0 - fb_0}{1 - fb_4}} \right] \quad (3.16)$$

denklemini yardımıyla hesaplanabilir. Bu denklem aşağıdaki biçime indirgenebilir.

$$(A - 64b_0b_4^3) f^4 + (2AB + 64a_0b_4^3 + 192b_0b_4^2) f^3 + (B^2 + 2AC - 192b_4(b_0 + b_4^2)) f^2 + (2BC + 192b_4a_0 + 64b_0) f + (C^2 - 64a_0) = 0 \quad (3.17)$$

$$A = 4b_2b_4 - b_3^2$$

$$B = 2a_3b_3 - 4b_2 - 4a_2b_4$$

$$C = 4a_2 - a_3^2 \quad (3.18)$$

Geribesleme katsayıları bu denklemin çözümünden belirlenebilir.

Bu bölümdeki açıklamalardan yararlanarak dördüncü dereceden en genel biçimdeki bir gerilim transfer fonksiyonunun, quad topolojisi kullanılarak tasarımı şu şekilde özetlenebilir :

1. Geribesleme katsayıları denklem (3.14) ya da denklem (3.16)'dan yararlanarak belirlenir.
2. Herbir geribesleme katsayısına karşı düşen iyilik faktörleri ve radyal frekanslar hesaplanır.
3. Elde edilen tasarım denklemleri birbirleriyle karşılaştırılır. Bunların içinden, ya en az duyarlıklı olan, ya iyilik faktörü en küçük olan ya da en iyi dinamik davranışı gösteren tasarım denklemleri seçilerek devre gerçekleştirilir.

3.2 İkinci Dereceden Transfer Fonksiyonundan, Alçak Geçiren Band Geçiren Dönüşümü Yardımıyla Elde Edilen 4.Dereceden Band Geçiren Filtrelerin Tasarımı

Dördüncü dereceden, band geçiren filtre gerilim transfer fonksiyonunun

$$T_{LP}(p) = \frac{\beta_2 p^2 + \beta_1 p + \beta_0}{p^2 + \alpha_1 p + \alpha_0} \quad (3.19)$$

ikinci derece denkleminde,

$$p = \frac{s^2 + w_0^2}{Bs} \quad (3.20)$$

dönüşümüyle elde edildiği varsayılın. Bu durumda, dördüncü dereceden filtrenin transfer fonksiyonu,

$$T(s) = \frac{\beta_2 s^4 + \beta_1 B s^3 + (2w_0^2 \beta_2 + \beta_0 B^2) s^2 + \beta_1 B w_0^2 s + \beta_2 w_0^4}{s^4 + \alpha_1 B s^3 + (2w_0^2 + \alpha_0 B^2) s^2 + \alpha_1 B w_0^2 s + w_0^4} \quad (3.21)$$

biçiminde ifade edilebilir. (3.21) denklemi $a_{m1} = a_{m3} a_{m0}$ koşulunu kendiliğinden sağlar. Denklem (3.14) yardımıyla geribesleme katsayısı $f = 0$ bulunur. Dolayısıyla bu durum kaskad tasarımdan başka bir şey değildir.

Denklem (3.21) ile verilen transfer fonksiyonu için (3.16) bağıntısı,

$$(\beta_1^2 - 4\beta_0 \beta_2) f^2 - 2(\alpha_1 \beta_1 - 2\beta_0 - 2\beta_2 \alpha_0) f + (\alpha_1^2 - 4\alpha_0) = 0 \quad (3.22.a)$$

eşitliği ile basite indirgenebilir. Bu denklemin köklerinden geribesleme katsayıları,

$$f = \frac{(\alpha_1 \beta_1 - 2\beta_0 - 2\beta_2 \alpha_0) \pm 2\sqrt{(\beta_0 - \alpha_0 \beta_2)^2 + \alpha_0 \beta_1^2 - \alpha_1 \beta_1 (\beta_0 + \beta_2 \alpha_0) + \beta_0 \beta_2 \alpha_1^2}}{(\beta_1^2 - 4\beta_0 \beta_2)} \quad (3.22.b)$$

biçiminde hesaplanabilir. Denklem (3.12) kullanılarak,

$$w_1 = w_2 = \sqrt{a_0} = w_0 \quad (3.23.a)$$

$$Q_{Q1} = Q_{Q2} = \frac{2(1 - f\beta_2)w_0}{B(\alpha_1 - \beta_1 f)} \quad (3.23.b)$$

değerleri hesaplanabilir. Aynı transfer fonksiyonunun, kaskad yöntemi ile tasarımından ise, aşağıdaki iyilik faktörü değerleri elde edilmektedir.

$$Q_{C1} = Q_{C2} = \sqrt{\frac{4w_0^2 + \alpha_0^2 B^2 + \sqrt{(4w_0^2 + \alpha_0 B^2) - 4w_0^2 \alpha_1^2 B^2}}{2B^2 \alpha_1^2}} \quad (3.24)$$

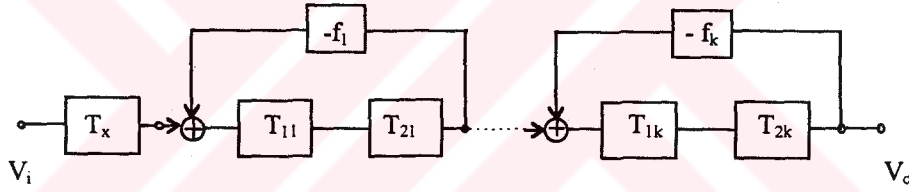
Sonuç olarak, (3.21) denklemiyle ifade edilen filtrenin tasarımında kullanılacak, ikinci dereceden alt devrelerin, transfer fonksiyonunun paydaları

$$D_1(s) = D_2(s) = s^2 + \frac{B(\alpha_1 - \beta_1 f)}{2(1 - f\beta_2)} s + w_0^2 \quad (3.25)$$

olarak yazılabilir.

3.3 Yüksek Dereceden Filtrelerin Quad Yöntemi İle Tasarımı

Bu alt bölümde yüksek dereceden filtrelerin quad yöntemi ile tasarımı konusu işlenecektir. Filtre derecesi genelliği bozmadan, çift sayı olarak alınacaktır. Şekil 3.2'de yüksek dereceden filtrenin blok diyagramı gösterilmiştir. T_x ile gösterilen alt devre biquadratik bir devreye ya da kazancı birim olan bir bağlantı devresine karşılık düşmektedir.



Şekil 3.2 Yüksek dereceden Quad devre topolojisi

Transfer fonksiyonunun sıfır ve kutupları belirlenerek, sıfır kutup düzenlemesi yapılır. Sıfır - kutup düzenlemesi sıfır ve kutupların dağılımına göre çok sayıda yapılabilir. Herhangi bir sıfır - kutup düzenlemesi sonucunda dördüncü dereceden olan transfer fonksiyonları kendi aralarında, olabildiği kadar çok sayıda, gruplandırılır. Geride transfer fonksiyonu kalmıyorsa, ya birinci ya ikinci ya da üçüncü derecedendir.

Yüksek dereceden bir aktif - RC filtresinin quad yöntemi ile gerçekleştirilmesi aşağıdaki biçimde özetlenebilir :

- 1) Verilen transfer fonksiyonunun sıfır ve kutupları belirlenir.

2) Sıfır ve kutuplar gelişigüzel düzenlenir. Bu konuda, genel bir kural yoktur. 3.3.1 alt bölümünde anlatılan yöntem, az duyarlıklı tasarım için iyi sonuçlar vererek bazı kurallara da ters düşmemektedir.

3) Dördüncü dereceden transfer fonksiyonları gruplandırıldıktan sonra geriye kalan transfer fonksiyonu, birinci dereceden ise pasif RC filtresi, ikinci dereceden ise bir biquad blok, üçüncü dereceden ise bir pasif RC filtresi ve bir biquad blok ile gerçekleştirilir. Şekil 3.2'deki T_x bloğu geriye kalan transfer fonksiyonunu gerçekleyen bloktur. Alçak geçiren filtre için bu T_x bloğu 2.3 alt bölümünde eşitlikleri elde edilmiş olan T_a ve / veya T_b bloğundan oluşmaktadır. Band geçiren filtre için ise bu T_x bloğu 2.4 alt bölümünde eşitlikleri elde edilmiş olan T_b bloğundan oluşmaktadır.

4) Geriye kalan ifadeden dördüncü dereceden $T_{q1}, T_{q2}, \dots, T_{qk}$ transfer fonksiyonları belirlenir.

5) Dördüncü dereceden quad devreleri gerçekleştirilir.

6) Olası bütün sıfır - kutup düzenlemeleri için yukarıdaki işlemler tekrarlanır.

7) Elde edilen tüm devreler, duyarlık, iyilik faktörleri ya da dinamik davranışları açısından karşılaştırılır.

8) İstenilen özellikteki, örneğin en az duyarlıklı ya da en küçük iyilik faktörü olan devre tasarımı seçilir ve gerçekleştirilir.

3.3.1 Sıfır - kutup düzenlemesi

Bu bölümde anlatılacak olan sıfır - kutup düzenlemesi yöntemi quad tasarım yönteminin yanısıra kaskad tasarım yöntemine de uygulanabilen bir yöntemdir. Yöntem aşağıdaki gibi özetlenebilir [3].

i) Quad bloklarına ilişkin payda çok terimlerinin elde edilmesi :

Bütün kutuplar, en büyük eğimli olanından başlanarak hemen hemen aynı eğime sahip olacak biçimde ikiyeşerli gruplara ayrılır. (En büyük eğime sahip olan kutup çifti birinci çift olarak adlandırılacaktır.). Elde edilen kutup çiftlerinden yararlanarak, quad transfer fonksiyonlarının payda çok terimlileri oluşturulur. Geride kalan kutuplar kullanılarak, T_x transfer fonksiyonunun paydası oluşturulur.

ii) Quad blokları pay çok terimlilerinin elde edilmesi:

Transfer fonksiyonu, şimdilik, gerçel sıfırlarının olmadığını varsayalım. Transfer fonksiyonu band geçiren [band durduran] filtreye karşılık düşüyorsa bütün sanal sıfırlar IG ile adlandırılan bir grup altında, karmaşık kutuplar da ayrı bir CG grubu altında toplanır. IG grubundan iki değer seçilerek bunların çarpımı $m_{zij} = w_{zi} w_{zj}$ ($i = 1, 2, \dots, nI$, $j = 1, 2, \dots, nI$) biçiminde oluşturulur. Burada nI , IG'de bulunan eleman sayısıdır. IG'deki olası tüm sıfır çiftleri için aynı şekilde çarpım sonuçları bulunur. Böylece $nI(nI - 1)/2$ değer elde edilir.

Payda çok terimlilerinin belirlenmesinde bulunan kutup çiftlerinden yararlanarak, $m_{pl} = w_{p1l} w_{p2l}$ ($l = 1, 2, \dots, k$ ve $k =$ kullanılan quad sayısı) çarpımları hesaplanır. En büyük eğimli kutup çiftleri (başka bir deyimle en yüksek iyilik faktörü olan kutup çiftleri) için $l = 1$ alınacaktır.

$l = 1$ değerinden başlayarak, m_{pl} ve m_{zij} değerleri karşılaştırılır. Bunların içinden farklarının mutlak değeri ($|m_{pl} - m_{zij}|$) en küçük olanlar bir araya getirilir. Tüm l değerleri için aynı işlem, geri kalan m_{pl} ve m_{zij} değerleri ile tekrarlanır. Diğer yandan, aynı l değeri için birbirine eşit, birden farklı m_{zij} çarpımı bu koşulu sağlıyorsa, m_{zij} çarpımını oluşturan w_{zi} ve w_{zj} çiftlerinden, $\min |w_{zi} - w_{zj}|$ değerini veren [Band durduran filtreler için $\max |w_{zi} - w_{zj}|$] sıfırları l 'inci quad blok kutupları ile birleştirilir. IG grubu içinde herhangi bir sıfır daha kalmışsa bu sıfır CG içerisine aktararak yukarıdaki işlemler CG grubu için de tekrarlanır. Eğer transfer fonksiyonunun gerçel sıfırları varsa bunlar geride kalan payda çok terimlileri ile uygun biçimde gruplandırılır.

BÖLÜM 4

ALÇAK GEÇİREN VE BAND GEÇİREN FİLTRELERİN QUAD VE KASKAD TASARIM DENKLEMLERİNİN ELDE EDİLMESİ

Bu bölümde, Bölüm 3’de anlatılmış olan genel quad yöntemi yardımıyla alçak geçiren ve band geçiren filtrelerin quad ve kaskad denklemlerinin elde edilmesi anlatılacaktır. Bölüm 2’de Butterworth, Chebyshev ve Eliptik yaklaşıklıklar kullanılarak verilen özellikleri sağlayan alçak geçiren ve band geçiren filtre transfer fonksiyonları elde edilmişti. Elde edilen transfer fonksiyonlarının quad yöntemi için oldukça uygun olduğu görülmektedir. Ancak quad yönteminin tamamen uygulanabilmesi için geri besleme katsayılarının ve biquadratik blokların transfer fonksiyonlarının belirlenmesi gerekmektedir. Gerekli olan denklemler bu bölümde elde edilecektir. Ayrıca kaskad tasarım için gerekli olan tasarım denklemleri de elde edilecektir.

4.1 Quad Tasarım Yöntemi Kullanılarak Alçak Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması

Butterworth ve Chebyshev yaklaşım için :

1. $w_{p1}(i)$ ve $w_{p2}(i)$ ’nin bulunması

Denklem (2.23)’de elde edilen $T_Q(s)$ fonksiyonunun katsayılarıyla denklem (3.3)’deki $T_Q(s)$ ’in katsayıları arasında birebir eşleme yapıldığında,

$$b_4(i) = 0 \quad b_3(i) = 0 \quad b_2(i) = 0 \quad b_1(i) = 0 \quad b_0(i) = w_p^4 \quad (4.1)$$

bulunur. Bu ifadede $i = 1, 2, \dots, t_x$ ’dir. t_x değeri quad bloklarının sayısı olup, daha önce denklem (2.23.b) ile verilmiştir. Bu alt bölümde bundan sonraki formüllerde

geçecek olan “i” ifadesi için de aynı değerler geçerlidir. Bulunan bu değerler denklem (3.6)’da yerine konulursa,

$$a_{m3}(i) = a_3(i) \quad (4.2.a)$$

$$a_{m2}(i) = a_2(i) \quad (4.2.b)$$

$$a_{m1}(i) = a_1(i) \quad (4.2.c)$$

$$a_{m0}(i) = \sqrt{a_0(i) - f(i)b_0(i)} \quad (4.2.d)$$

elde edilir. Tasarımımızda $a_{m1} = a_{m0} a_{m3}$ ve $a_{m3}^2 = 4(a_{m2} - 2a_{m0})$ koşullarının her ikisinin birlikte sağlandığı durum ele alınacaktır. Bu durumda (3.12) bağıntılarını kullanabiliriz. $a_{m1} = a_{m0} a_{m3}$ koşulu yardımıyla,

$$a_{m0}(i) = \frac{a_{m1}(i)}{a_{m3}(i)} = \frac{a_1(i)}{a_3(i)} \quad (4.3)$$

yazılabilir. (4.3) denklemi (3.12.a)’da yerine konulursa,

$$w_{p1}(i) = w_{p2}(i) = \sqrt{\frac{a_1(i)}{a_3(i)}} \quad (4.4)$$

elde edilir.

2. $f(i)$, geri besleme katsayılarının bulunması

Denklem (3.12.a) kullanılarak,

$$a_{m0}(i) = w_{p1}^2(i) = w_{p2}^2(i) \quad (4.5)$$

yazılabilir. Denklem (4.2.d) ve (4.5) yardımıyla geribesleme katsayıları için

$$w_{p1}^2(i) = w_{p2}^2(i) = \sqrt{a_0(i) - f(i)b_0(i)} \quad (4.6.a)$$

$$f(i) = \frac{a_0(i) - w_{p1}^4(i)}{b_0(i)} \quad (4.6.b)$$

ifadeleri elde edilebilir.

3. $Q_1(i)$, $Q_2(i)$ iyilik faktörlerinin bulunması

Denklem (3.12.a), (3.12.b)'de yerine konulursa,

$$Q_1(i) = Q_2(i) = \frac{2w_{p1}(i)}{a_{m3}(i)} = \frac{2w_{p1}(i)}{a_3(i)} \quad (4.7)$$

elde edilir.

4. Biquadratik blokların transfer fonksiyonlarının bulunması

(3.5) bağıntısında, (4.1) bağıntısındaki değerler yerine konulursa,

$$T(s) = T_{li}(s)T_{2i}(s) = \frac{w_p^4}{s^4 + a_{m3}(i)s^3 + a_{m2}(i)s^2 + a_{m1}(i)s + a_{m0}(i)} \quad (4.8)$$

ifadesi elde edilebilir. Bulunan bu denklem yardımıyla,

$$T_{li}(s) = \frac{w_p^2}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.9.a)$$

$$T_{2i}(s) = \frac{w_p^2}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.9.b)$$

yazılabilir.

Eliptik yaklaşım için:

1. w_{p1} ve w_{p2} 'nin bulunması

Denklem (2.32)'de elde edilen $T_{Qi}(s)$ fonksiyonunun katsayılarıyla denklem (3.3)'teki $T_Q(s)$ 'in katsayıları arasında birebir eşleme yapılırsa,

$$b_4(i) = 1 \quad (4.10.a)$$

$$b_3(i) = 0 \quad (4.10.b)$$

$$b_2(i) = [C_{k(i)} + C_{k(i)+1}] w_p^2 \quad (4.10.c)$$

$$b_1(i) = 0 \quad (4.10.d)$$

$$b_0(i) = C_{k(i)} C_{k(i)+1} w_p^4 \quad (4.10.e)$$

elde edilir. Bu değerler denklem (3.6)'da yerine konulursa,

$$a_{m3}(i) = \frac{a_3(i)}{1 - f(i)} \quad (4.11.a)$$

$$a_{m2}(i) = \frac{a_2(i) - f(i)b_2(i)}{1 - f(i)} \quad (4.11.b)$$

$$a_{m1}(i) = \frac{a_1(i)}{1 - f(i)} \quad (4.11.c)$$

$$a_{m0}(i) = \sqrt{\frac{a_0(i) - f(i)b_0(i)}{1 - f(i)}} \quad (4.11.d)$$

bulunur. $a_{m1} = a_{m0} a_{m3}$ koşulu ve (4.11) bağıntıları yardımıyla,

$$a_{m0}(i) = \frac{a_{m1}(i)}{a_{m3}(i)} = \frac{a_1(i) / (1 - f(i))}{a_3(i) / (1 - f(i))} = \frac{a_1(i)}{a_3(i)} \quad (4.12)$$

yazılabilir. (4.12) denklemi (3.12.a)'da yerine konulursa,

$$w_{p1}(i) = w_{p2}(i) = \sqrt{\frac{a_1(i)}{a_3(i)}} \quad (4.13)$$

elde edilir.

2. $f(i)$ geribesleme katsayılarının bulunması

Denklem (3.12.a) kullanılarak,

$$a_{m0}(i) = w_{p1}^2(i) = w_{p2}^2(i) \quad (4.14)$$

yazılabilir. Denklem(4.11.d)'deki $a_{m0}(i)$ değeri denklem (4.14)'de yerine konulursa,

$$w_{p1}^2(i) = w_{p2}^2(i) = \sqrt{\frac{a_0(i) - f(i)b_0(i)}{1 - f(i)}} \quad (4.15.a)$$

$$f(i) = \frac{a_0(i) - w_{p1}^4(i)}{b_0(i) - w_{p1}^4(i)} \quad (4.15.b)$$

ifadeleri elde edilir.

3. $Q_1(i)$, $Q_2(i)$ iyilik faktörlerinin bulunması

Denklem (3.12.a) ve (3.12.b) yardımıyla,

$$Q_1(i) = Q_2(i) = \frac{2w_{p1}(i)}{a_{m3}(i)} \quad (4.16)$$

yazılabilir. (4.11.a) bağıntısındaki $a_{m3}(i)$ değeri (4.16) denkleminde yerine konulursa,

$$Q_1(i) = Q_2(i) = \frac{2w_{p1}(i)(1-f(i))}{a_3(i)} \quad (4.17)$$

bulunur.

4. Biquadratik blokların transfer fonksiyonlarının bulunması

Denklem (4.10)'daki değerler denklem (3.5)'de yerine konulursa,

$$T(s) = \frac{1}{1-f(i)} \frac{s^4 + (C_{k(i)} + C_{k(i)+1})w_p^2 s^2 + (C_{k(i)}C_{k(i)+1}w_p^4)}{(s^4 + a_{m3}(i)s^3 + a_{m2}(i)s^2 + a_{m1}(i)s + a_{m0}(i))} \quad (4.18)$$

ifadesi elde edilir. $T(s)$ 'in payı derecesi çift olan s 'li terimlerden oluşmaktadır. Dolayısıyla denklem (3.2.b),

$$T_{1i}(s) = \frac{k_1(i)k_2(i)(s^2 + w_{z1}^2(i))}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.19.a)$$

$$T_{2i}(s) = \frac{s^2 + w_{z2}^2(i)}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.19.b)$$

$$k_1(i)k_2(i) = \frac{1}{1-f(i)} \quad (4.19.c)$$

şeklini alır. Bu ifadelerden anlaşılacağı gibi, $T_{1i}(s)$ ve $T_{2i}(s)$ fonksiyonlarının elde edilebilmesi için $w_{z1}(i)$ ve $w_{z2}(i)$ değerlerinin de bulunması gerekir.

$w_{z1}(i)$, $w_{z2}(i)$ değerlerinin bulunması

Denklem (3.5) yardımıyla

$$(s^2 + w_{z1}^2(i))(s^2 + w_{z2}^2(i)) = b_4(i)s^4 + b_3(i)s^3 + b_2(i)s^2 + b_1(i)s + b_0(i) \quad (4.20)$$

eşitliği yazılabilir. Bu eşitliğin sol tarafının açık ifadesi bulunup sağ taraftaki ifade ile birebir eşleme yapıldığında,

$$w_{z1}^2(i) + w_{z2}^2(i) = b_2(i) \quad (4.21.a)$$

$$w_{z1}^2(i)w_{z2}^2(i) = b_0(i) \quad (4.21.b)$$

ifadeleri elde edilebilir. Bu ifadeler yardımıyla,

$$(w_{z1}(i) + w_{z2}(i))^2 = w_{z1}^2(i) + w_{z2}^2(i) + 2w_{z1}(i)w_{z2}(i)$$

$$(w_{z1}(i) + w_{z2}(i))^2 = b_2(i) + 2\sqrt{b_0(i)}$$

$$(w_{z1}(i) + w_{z2}(i)) = \sqrt{b_2(i) + 2\sqrt{b_0(i)}}$$

$$w_{z1}(i) + \frac{\sqrt{b_0(i)}}{w_{z1}(i)} = \sqrt{b_2(i) + 2\sqrt{b_0(i)}}$$

$$w_{z1}^2(i) - \sqrt{b_2(i) + 2\sqrt{b_0(i)}}w_{z1}(i) + \sqrt{b_0(i)} = 0 \quad (4.22)$$

şeklinde bir ikinci dereceden denklem elde edilebilir. Bu ikinci dereceden denklem kullanılarak,

$$w_{z1}(i) = \frac{\sqrt{b_2(i) + 2\sqrt{b_0(i)}} + \sqrt{b_2(i) - 2\sqrt{b_0(i)}}}{2} \quad (4.23)$$

ifadesi bulunabilir. Denklem (4.21.b) yardımıyla,

$$w_{z2}(i) = \frac{\sqrt{b_0(i)}}{w_{z1}(i)} \quad (4.24)$$

yazılabilir.

4.2 Kaskad Tasarım Yöntemi Kullanılarak Alçak Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması

Biquadratik blokların transfer fonksiyonlarının elde edilebilmesi için ilk önce $w_{p1}(i)$, $w_{p2}(i)$, $Q_1(i)$ ve $Q_2(i)$ değerlerinin bulunması gerekmektedir. Bu değerlerin bulunma yöntemi quad tasarımının aksine, yaklaşıklık tiplerine göre değişmemektedir. $w_{p1}(i)$, $w_{p2}(i)$, $Q_1(i)$ ve $Q_2(i)$ değerleri biquadratik transfer fonksiyonlarının paydalarının elde edilmesinde kullanılmaktadırlar. Sadece biquadratik transfer fonksiyonlarının paylarının bulunmasında yaklaşıklık tiplerine göre ayırım yapmak gerekmektedir.

1. $w_{p1}(i)$, $w_{p2}(i)$ değerlerinin bulunması

Kaskad tasarım için geribesleme katsayısı $f(i) = 0$ 'dır. Dolayısıyla, (3.4) bağıntısı yardımıyla,

$$T(s) = T_1(s)T_2(s) = T_{Q1}(s) = T_{K1}(s) \quad (4.25)$$

yazılabilir. (4.25) bağıntısında (3.2.b)'deki $T_1(s)$, $T_2(s)$ ve (3.3)'deki $T_Q(s)$ değerleri yerine konulursa,

$$T_{K1}(s) = \frac{k_1 [s^2 + (w_{z1}(i)/Q_{z1}(i))s + w_{z1}^2(i)] [s^2 + (w_{z2}(i)/Q_{z2}(i))s + w_{z2}^2(i)]}{[s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)] [s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)]}$$

$$= \frac{b_4(i)s^4 + b_3(i)s^3 + b_2(i)s^2 + b_1(i)s + b_0(i)}{s^4 + a_3(i)s^3 + a_2(i)s^2 + a_1(i)s + a_0(i)} \quad (4.26)$$

ifadeleri elde edilebilir. Bu ifadelerde $i = 1, 2, \dots, tx$ 'dir. tx değeri kaskad blokların sayısı olup, denklem (2.23.b) ile verilen değerle aynıdır. Bu alt bölümde bundan sonraki formüllerde geçecek olan i ifadesi için de aynı değerler geçerlidir. Kaskad blokların, Quad bloklardan tek farkı, geribesleme bloğunun olmamasıdır. Yani bir kaskad blok sadece $T_1(s)$ ve $T_2(s)$ bloklarından oluşmaktadır. (4.26) ifadesinden yararlanarak,

$$[s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)] [s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)] = s^4 + a_3s^3 + a_2s^2 + a_1s + a_0 \quad (2.27)$$

yazılabilir. Bu bağıntının sol tarafının açık ifadesi bulunup sağ taraftaki ifadeyle birebir eşleme yapılırsa,

$$w_{p1}^2(i)w_{p2}^2(i) = a_0(i) \quad (4.28.a)$$

$$w_{p1}^2(i) + w_{p2}^2(i) + \frac{w_{p1}(i)w_{p2}(i)}{Q_1(i)Q_2(i)} = a_2(i) \quad (4.28.b)$$

$$\frac{w_{p1}(i)}{Q_1(i)} + \frac{w_{p2}(i)}{Q_2(i)} = a_3(i) \quad (4.28.c)$$

ifadeleri elde edilir. Tasarımımızda, işlemlerde kolaylık sağlamak amacıyla $Q_1(i)$ ve $Q_2(i)$ 'nin birbirine eşit olduğu varsayılacaktır. Bu durumda (4.28.a), (4.28.b) ve (4.28.c) ifadeleri,

$$w_{p1}(i)w_{p2}(i) = \sqrt{a_0(i)} \quad (4.29.a)$$

$$w_{p1}^2(i) + w_{p2}^2(i) = a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)} \quad (4.29.b)$$

$$\frac{w_{p1}(i) + w_{p2}(i)}{Q_1(i)} = a_3(i) \quad (4.29.c)$$

şeklini alır. (4.29.a) ve (4.29.b) ifadeleri yardımıyla,

$$(w_{p1}(i) + w_{p2}(i))^2 = w_{p1}^2(i) + w_{p2}^2(i) + 2w_{p1}(i)w_{p2}(i)$$

$$(w_{p1}(i) + w_{p2}(i))^2 = a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)} + 2\sqrt{a_0(i)}$$

$$\begin{aligned}
 (w_{p1}(i) + w_{p2}(i)) &= \sqrt{a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)} + 2\sqrt{a_0(i)}} \\
 w_{p1}(i) + \frac{\sqrt{a_0(i)}}{w_{p1}(i)} &= \sqrt{a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)} + 2\sqrt{a_0(i)}} \\
 w_{p1}^2(i) - \sqrt{a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)} + 2\sqrt{a_0(i)}} w_{p1}(i) + \sqrt{a_0(i)} &= 0
 \end{aligned} \tag{4.30}$$

biçiminde bir ikinci dereceden denklem elde edilebilir. Bu ikinci dereceden denklem kullanılarak,

$$w_{p1}(i) = \frac{\sqrt{a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)} + 2\sqrt{a_0(i)}} + \sqrt{a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)} - 2\sqrt{a_0(i)}}}{2} \tag{4.31}$$

ifadesi elde edilebilir. Denklem (4.29.a) yardımıyla,

$$w_{p2}(i) = \frac{\sqrt{a_0(i)}}{w_{p1}(i)} \tag{4.32}$$

yazılabilir.

2. $Q_1(i), Q_2(i)$ iyilik faktörlerinin bulunması

(4.29.c) bağıntısı kullanılarak,

$$Q_1(i) = Q_2(i) = \frac{w_{p1}(i) + w_{p2}(i)}{a_3(i)} \tag{4.33}$$

ifadesi elde edilebilir.

3. Biquadratik transfer fonksiyonlarının bulunması

Butterworth ve Chebyshev yaklaşım için :

Denklem (2.23.g)'deki $T_{Qi}(s)$ değerini denklem (4.25)'de yerine koyarsak,

$$T(s) = T_{li}(s)T_{zi}(s) = \frac{w_p^4}{D(s, k(i))} \quad (4.34)$$

ifadesi elde edilebilir. Bu ifadedeki $D(s, k(i))$ daha önce denklem (2.23.d) ile verilmişti. $D(s, k(i))$ değeri aynı zamanda, (3.3) bağıntısı yardımıyla,

$$D(s, k(i)) = s^4 + a_3(i)s^3 + a_2(i)s^2 + a_1(i)s + a_0(i) \quad (4.35)$$

biçiminde yazılabilir. (4.34), (4.35) ve (3.2.b) kullanılarak,

$$T_{li}(s) = \frac{w_p^2}{s^2 + (w_{p1}(i) / Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.36.a)$$

$$T_{zi}(s) = \frac{w_p^2}{s^2 + (w_{p2}(i) / Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.36.b)$$

bulunabilir.

Eliptik yaklaşım için :

Denklem (4.10)'daki değerler denklem (3.3)'de yerine konulursa,

$$T_{Qi}(s) = \frac{s^4 + (C_{k(i)} + C_{k(i)+1})w_p^2s^2 + (C_{k(i)}C_{k(i)+1}w_p^4)}{(s^4 + a_3(i)s^3 + a_2(i)s^2 + a_1(i)s + a_0(i))} \quad (4.37)$$

ifadesi elde edilebilir. (4.25) bağıntısı kullanılarak,

$$T(s) = T_{li}(s)T_{2i}(s) = \frac{s^4 + (C_{k(i)} + C_{k(i)+1})w_p^2 s^2 + (C_{k(i)}C_{k(i)+1}w_p^4)}{(s^4 + a_3(i)s^3 + a_2(i)s^2 + a_1(i)s + a_0(i))} \quad (4.38)$$

yazılabilir. (3.2.b) denkleminin açık ifadesi yazılırsa,

$$T_{li}(s) = \frac{k_1(i)(s^2 + (w_{z1}(i)/Q_{z1}(i))s + w_{z1}^2(i))}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.39.a)$$

$$T_{2i}(s) = \frac{k_2(i)(s^2 + (w_{z2}(i)/Q_{z2}(i))s + w_{z2}^2(i))}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.39.b)$$

elde edilir. (4.38) denkleminin pay ve paydasındaki s^4 'lü terimlerin katsayılarının değeri 1'dir. Dolayısıyla (4.39.a) ve (4.39.b) denklemlerinin çarpımından elde edilen rasyonel fonksiyonun pay ve paydasındaki s^4 'lü terimlerin katsayılarının değerinin 1 olması için,

$$k_1(i)k_2(i) = 1 \quad (4.40)$$

eşitliğinin sağlanması gerekir. Ayrıca (4.38) denklemindeki $T(s)$ 'in payı derecesi çift olan s 'li terimlerden oluşmalıdır. Bu durumda, (4.39.a) ve (4.39.b) ifadeleri,

$$T_{li}(s) = \frac{s^2 + w_{z1}^2(i)}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.41.a)$$

$$T_{2i}(s) = \frac{s^2 + w_{z2}^2(i)}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.41.b)$$

şeklini alır. Bu ifadelerden anlaşılacağı gibi, biquadratik transfer fonksiyonlarının belirlenebilmesi için $w_{z1}(i)$ ve $w_{z2}(i)$ değerlerinin bulunması gerekmektedir.

$w_{z1}(i), w_{z2}(i)$ değerlerinin bulunması :

Denklem (3.5)'de $f = 0$ değeri ve (4.41.a), (4.41.b) ifadeleri yerine konulursa,

$$T(s) = \frac{s^2 + w_{z1}^2(i)}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \frac{s^2 + w_{z2}^2(i)}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)}$$

$$= \frac{b_4(i)s^4 + b_3(i)s^3 + b_2(i)s^2 + b_1(i)s + b_0(i)}{s^4 + a_{m3}(i)s^3 + a_{m2}(i)s^2 + a_{m1}(i)s + a_{m0}(i)} \quad (4.42)$$

elde edilir. Payların eşitliği ilkesinden yararlanarak,

$$(s^2 + w_{z1}^2(i))(s^2 + w_{z2}^2(i)) = b_4(i)s^4 + b_3(i)s^3 + b_2(i)s^2 + b_1(i)s + b_0(i) \quad (4.42.a)$$

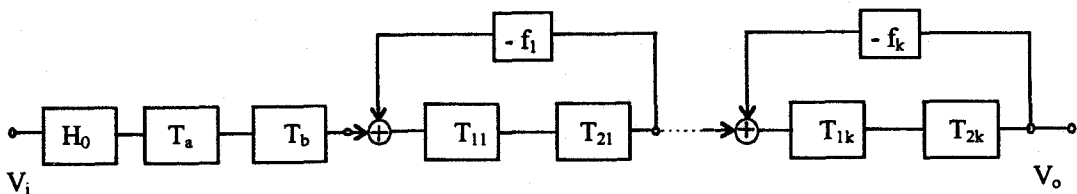
yazılabilir. Bu ifadenin, (4.20) ifadesinden hiçbir farkı yoktur. Dolayısıyla, ara işlemleri vermeden doğrudan doğruya (4.23) ve (4.24) bağıntıları yardımıyla sonuç ifadelerini yazabiliriz.

$$w_{z1}(i) = \frac{\sqrt{b_2(i) + 2\sqrt{b_0(i)}} + \sqrt{b_2(i) - 2\sqrt{b_0(i)}}}{2} \quad (4.43)$$

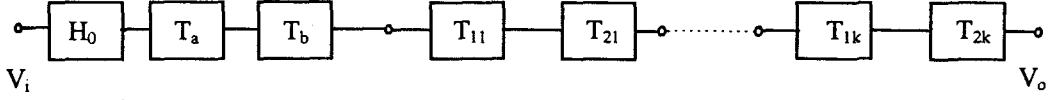
$$w_{z2}(i) = \frac{\sqrt{b_0(i)}}{w_{z1}(i)} \quad (4.44)$$

4.3 Elde Edilen Alçak Geçiren Filtre Transfer Fonksiyonu İle Quad Ve Kaskad Tasarım

Her üç yaklaşıklık metodu için kullanılacak olan yüksek dereceden alçak geçiren quad filtresinin blok diyagramı en genel biçimde Şekil 4.1'de, kaskad filtrenin blok diyagramı ise Şekil 4.2'de gösterilmiştir.



Şekil 4.1 Yüksek dereceden alçak geçiren quad filtresi blok diyagramı



Şekil 4.2 Yüksek dereceden alçak geçiren kaskad filtresi blok diyagramı

Blok diyagramlardaki T_a ve T_b blokları için doğrudan doğruya filtre derecesine bağlı olan dört olasılık bulunmaktadır.

1. Sadece T_a var olabilir,
2. Sadece T_b var olabilir,
3. T_a ve T_b bloklarından her ikisi birden var olabilir,
4. T_a ve T_b bloklarından hiçbirisi olmayabilir.

Sözü edilen olasılıkların filtre derecesine nasıl bağlı oldukları 2.3.1 alt bölümünde anlatılmıştı.

Şekil 4.1 ve Şekil 4.2'deki T_a bloğu birinci dereceden transfer fonksiyonuna sahiptir ve pasif RC devresi ile gerçekleştirilmektedir. T_b blokları ise ikinci dereceden transfer fonksiyonuna sahiptir ve Butterworth, Chebyshev türü yaklaşım için AG yapısında, Eliptik yaklaşım için ise Notch yapısındadırlar. T_{1k} , T_{2k} blokları da ikinci dereceden transfer fonksiyonuna sahiptir ve Butterworth, Chebyshev yaklaşıklık için AG yapısında, Eliptik yaklaşım için ise Notch yapısındadırlar. T_b , T_{1k} , T_{2k} blokları Bölüm 5'de anlatılacak olan OTA - C biquad devreleri kullanılarak gerçekleştirilebilirler.

Alçak geçiren OTA - C filtrelerinin quad ve kaskad yöntemleri ile tasarımı için aşağıdaki adımlar izlenir (Her iki tasarım yöntemi için algoritma aynıdır):

1. Tasarlanacak alçak geçiren filtrenin özelliklerine göre Butterworth, Chebyshev veya Eliptik yaklaşımlardan uygun olanı seçilir.
2. Filtreye ilişkin $H(s)$ transfer fonksiyonu Butterworth, Chebyshev yaklaşım için (2.24), (2.25), (2.28), (2.29) bağıntıları yardımıyla, Eliptik yaklaşım için (2.32), (2.33), (2.36), (2.37) bağıntıları yardımıyla bulunur.

3. Quad tasarım yöntemi için 4.1 alt bölümünde elde edilen bağıntılar yardımıyla biquadratik transfer fonksiyonları, geribesleme katsayıları ve iyilik faktörleri bulunur. Kaskad tasarım yöntemi için ise 4.2 alt bölümünde elde edilen bağıntılar yardımıyla biquadratik transfer fonksiyonları ve iyilik faktörleri bulunur.

4. Elde edilen quad/kaskad bloklar filtre dinamik değişimi maksimum olacak şekilde sıralanır. Sıralama yapıldıktan sonra biquadratik bloklara ilişkin kazanç sabitleri ve yeni geri besleme katsayıları Bölüm 6'da verilecek olan prosedüre göre belirlenir.

5. Devredeki OTA iletkenlikleri ve kapasite elemanlarının değerleri belirlenir.

6. "Clipping" ve "slew-rate" sorunlarını ortadan kaldıracak şekilde filtre girişine uygulanabilecek maksimum giriş geriliminin genliği hesaplanır.

4.4 Quad Tasarım Yöntemi Kullanılarak Band Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması

Butterworth ve Chebyshev yaklaşım için :

1. $w_{p1}(i)$, $w_{p2}(i)$, $Q_1(i)$, $Q_2(i)$ ve $f(i)$ 'nin bulunması

$$w_0 = \sqrt{w_{p1} w_{p2}} \quad (4.45)$$

$$w_{p1}(i) = w_{p2}(i) = w_0 \quad (4.46)$$

$$Q_1(i) = Q_2(i) = \frac{2w_0}{B} \quad (4.47)$$

$$f(i) = B_i - \frac{A_i}{4} \quad (4.48)$$

Bu ifadelerde $i = 1, 2, \dots, n_x$ 'dir. n_x değeri quad blokların sayısı olup daha önce denklem (2.8) ile verilmişti. Bu alt bölümde bundan sonraki formüllerde geçecek olan "i" ifadesi için de aynı değerler geçerlidir.

2. Biquadratik blokların transfer fonksiyonlarının bulunması

Denklem (2.41)'de elde edilen $T_{Qi}(s)$ fonksiyonunun katsayıları ile denklem (3.3)'de elde edilen $T_Q(s)$ 'in katsayıları arasında birebir eşleme yapıldığında,

$$b_4(i) = 0 \quad b_3(i) = 0 \quad b_2(i) = B^2 \quad b_1(i) = 0 \quad b_0(i) = 0 \quad (4.49)$$

bulunur. (3.5) bağıntısında (4.49) bağıntısındaki değerler yerine konulursa,

$$T(s) = T_{li}(s)T_{2i}(s) = \frac{B^2 s^2}{s^4 + a_{m3}(i)s^3 + a_{m2}(i)s^2 + a_{m1}(i)s + a_{m0}(i)} \quad (4.50)$$

ifadesi elde edilir. Bulunan bu denklem ve (3.2.b) denklemi yardımıyla,

$$T_{li}(s) = \frac{Bs}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.51.a)$$

$$T_{2i}(s) = \frac{Bs}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.51.b)$$

yazılabilir.

Eliptik yaklaşıklık için :

1. $w_{p1}(i)$, $w_{p2}(i)$, $Q_1(i)$, $Q_2(i)$ ve $f(i)$ 'nin bulunması

$$w_{p1}(i) = w_{p2}(i) = w_0 \quad (4.52)$$

$$Q_1(i) = Q_2(i) = \frac{2(1 - f(i))w_0}{BA_i} \quad (4.53)$$

$$f(i) = \frac{B_i + C_i - \sqrt{B_i - C_i^2 + C_i A_i^2}}{2C_i} \quad (4.54)$$

(4.52) ve (4.53) denklemlerinde geçen w_0 değeri (4.45) denklemi ile bulunabilir.

2. Biquadratik blokların transfer fonksiyonlarının bulunması

Denklem (2.44)'de elde edilen $T_{Qi}(s)$ fonksiyonunun katsayılarıyla denklem (3.3)'teki $T_Q(s)$ 'in katsayıları arasında birebir eşleme yapılırsa,

$$b_4(i) = 1 \quad (4.55.a)$$

$$b_3(i) = 0 \quad (4.55.b)$$

$$b_2(i) = 2 w_0^2 + C_i B^2 \quad (4.55.c)$$

$$b_1(i) = 0 \quad (4.55.d)$$

$$b_0(i) = w_0^4 \quad (4.55.e)$$

elde edilir. Bulunan bu ifadeler denklem (3.5)'de yerine konulursa,

$$T(s) = \frac{1}{1-f(i)} \frac{s^4 + (2w_0^2 + C_i B^2)s^2 + w_0^4}{(s^4 + a_{m3}(i)s^3 + a_{m2}(i)s^2 + a_{m1}(i)s + a_{m0}(i))} \quad (4.56)$$

ifadesi elde edilir. $T(s)$ 'in payı derecesi çift olan s 'li terimlerden oluşmaktadır. Dolayısıyla $T_{1i}(s)$ ve $T_{2i}(s)$ transfer fonksiyonlarının da paylarının derecesi çift olan s 'li terimlerden oluşması gerekir. Bu durumda denklem (3.2.b),

$$T_{1i}(s) = \frac{k_1(i)k_2(i)(s^2 + w_{z1}^2(i))}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.57.a)$$

$$T_{2i}(s) = \frac{s^2 + w_{z2}^2(i)}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.57.b)$$

$$k_1(i)k_2(i) = \frac{1}{1-f(i)} \quad (4.57.c)$$

şeklini alır. Bu ifadelerden anlaşılacağı gibi $T_{1i}(s)$ ve $T_{2i}(s)$ 'in tam olarak belirlenebilmesi için $w_{z1}(i)$ ve $w_{z2}(i)$ değerlerinin de bulunması gerekir. Bulunan bu denklemlerle (4.19.a), (4.19.b) ve (4.19.c) denklemlerinin hiçbir farkı yoktur. O halde, ara işlemleri vermeden doğrudan doğruya (4.23) ve (4.24) ifadeleri yardımıyla sonuç ifadelerini yazabiliriz.

$$w_{z1}(i) = \frac{\sqrt{b_2(i) + 2\sqrt{b_0(i)}} + \sqrt{b_2(i) - 2\sqrt{b_0(i)}}}{2} \quad (4.58)$$

$$w_{z2}(i) = \frac{\sqrt{b_0(i)}}{w_{z1}(i)} \quad (4.59)$$

4.5 Kaskad Tasarım Yöntemi Kullanılarak Band Geçiren Filtre Biquadratik Transfer Fonksiyonlarının Bulunması

1. $w_{p1}(i)$, $w_{p2}(i)$, $Q_1(i)$, $Q_2(i)$ 'nin bulunması

$w_{p1}(i)$, $w_{p2}(i)$, $Q_1(i)$, $Q_2(i)$ değerlerinin bulunma yöntemi yaklaşıklık tipine göre değişmemektedir. Kaskad band geçiren filtre tasarımında bu değerlerin belirlenmesi yöntemi, kaskad alçak geçiren filtre tasarımındaki aynısıdır. Sadece, $i=1, 2, \dots, tx$ yerine $i=1, 2, \dots, nx$ 'dir. Dolayısıyla bu değerlerin sonuç ifadelerini doğrudan doğruya verebiliriz.

$$w_{p1}(i) = \frac{\sqrt{a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)}} + 2\sqrt{a_0(i)}}{2} + \frac{\sqrt{a_2(i) - \frac{\sqrt{a_0(i)}}{Q_1^2(i)}} - 2\sqrt{a_0(i)}}{2} \quad (4.60)$$

$$w_{p2}(i) = \frac{\sqrt{a_0(i)}}{w_{p1}(i)} \quad (4.61)$$

$$Q_1(i) = Q_2(i) = \frac{w_{p1}(i) + w_{p2}(i)}{a_3(i)} \quad (4.62)$$

2. Biquadratik transfer fonksiyonlarının bulunması

Butterworth ve Chebyshev yaklaşıklık için:

Denklem (3.4)'de (2.41)'de elde edilen $T_{Qi}(s)$ fonksiyonunu ve $f = 0$ değerini yerine koyarsak,

$$T(s) = T_{li}(s)T_{2i}(s) = T_{Qi}(s) = \frac{B^2 s^2}{s^4 + A_i B s^3 + (2w_0^2 + B_i B^2)s^2 + (A_i B w_0^2)s + w_0^4} \quad (4.63)$$

elde edilir. (3.2.b) yardımıyla,

$$T_{li}(s) = \frac{Bs}{s^2 + (w_{p1}(i)/Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.64.a)$$

$$T_{2i}(s) = \frac{Bs}{s^2 + (w_{p2}(i)/Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.64.b)$$

yazılabilir.

Eliptik yaklaşıklık için:

Denklem (2.44)'de elde edilen $T_{Qi}(s)$ fonksiyonu ve $f = 0$ değeri denklem (3.4)'de yerine konulursa,

$$T(s) = T_{li}(s)T_{2i}(s) = T_{Qi}(s) = \frac{s^4 + (2w_0^2 + C_i B^2)s^2 + w_0^4}{s^4 + A_i B s^3 + (2w_0^2 + B_i B^2)s^2 + (A_i B w_0^2)s + w_0^4} \quad (4.65)$$

ifadesi elde edilebilir. (3.2.b) denkleminin açık ifadesi daha önce (4.39.a) ve (4.39.b) denklemleriyle verilmişti. (4.65) denkleminin pay ve paydasındaki s^4 'lü terimlerin katsayılarının değeri 1'dir. Dolayısıyla (4.39.a) ve (4.39.b) denklemlerinin çarpımından elde edilen rasyonel fonksiyonun pay ve paydasındaki s^4 'lü terimlerin katsayılarının değerinde 1 olması gerekir. Bunun sağlanması için,

$$k_1(i) k_2(i) = 1 \quad (4.66)$$

olmalıdır. Ayrıca, (4.65) denklemindeki $T(s)$ 'in payı derecesi çift olan s 'li terimlerden oluşmaktadır. Dolayısıyla, (4.39.a) ve (4.39.b) ifadelerinin paylarının, derecesi çift olan s 'li terimlerden oluşması gerekir. Bu durumda, (4.39.a) ve (4.39.b) ifadeleri,

$$T_{1i}(s) = \frac{s^2 + w_{z1}^2(i)}{s^2 + (w_{p1}(i) / Q_{p1}(i))s + w_{p1}^2(i)} \quad (4.67.a)$$

$$T_{2i}(s) = \frac{s^2 + w_{z2}^2(i)}{s^2 + (w_{p2}(i) / Q_{p2}(i))s + w_{p2}^2(i)} \quad (4.67.b)$$

şeklini alır. Bulunan $T_{1i}(s)$ ve $T_{2i}(s)$ fonksiyonlarının (4.41.a) ve (4.41.b)'den hiçbir farkı yoktur. Dolayısıyla, $w_{z1}(i)$ ve $w_{z2}(i)$ değerlerinin bulunması sırasında yine (4.42) denklemini elde ederiz. Sonuç olarak, $w_{z1}(i)$ ve $w_{z2}(i)$ değerleri (4.43) ve (4.44) ifadeleri ile aynı olacaktır. Öyleyse,

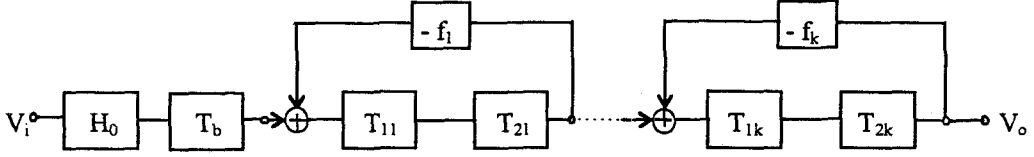
$$w_{z1}(i) = \frac{\sqrt{b_2(i) + 2\sqrt{b_0(i)}} + \sqrt{b_2(i) - 2\sqrt{b_0(i)}}}{2} \quad (4.68)$$

$$w_{z2}(i) = \frac{\sqrt{b_0(i)}}{w_{z1}(i)} \quad (4.69)$$

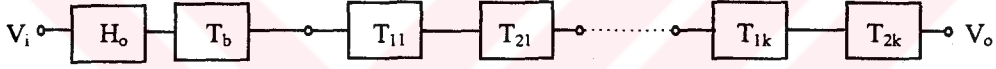
ifadeleri yazılabilir.

4.6 Elde Edilen Band Geçiren Filtre Transfer Fonksiyonu İle Quad Ve Kaskad Tasarım

Butterworth, Chebyshev ve Eliptik yaklaşıklık tipleri için yüksek dereceden band geçiren quad filtresinin blok diyagramı Şekil 4.3'de, kaskad filtrenin blok diyagramı ise Şekil 4.4'de gösterilmiştir.



Şekil 4.3 Yüksek dereceden band geçiren quad filtresi blok diyagramı



Şekil 4.4 Yüksek dereceden band geçiren kaskad filtresi blok diyagramı

Blok diyagramlarda görülen $T_b(s)$ bloğu, filtre derecesi N 'in tek olması durumunda var olmaktadır. N 'in çift olması durumunda $T_b(s)$ var olmayacaktır.

Blok diyagramlardaki $T_b(s)$ bloğu ikinci dereceden transfer fonksiyonuna sahiptir ve tüm yaklaşıklık tipleri için BG yapısındadır. T_{1k} , T_{2k} blokları da ikinci dereceden transfer fonksiyonuna sahiptir ve Butterworth, Chebyshev yaklaşıklık tipi için BG yapısında, Eliptik yaklaşım için ise Notch yapısındadır.

Band geçiren OTA-C filtrelerinin quad ve kaskad yöntemleri ile tasarımı için aşağıdaki adımlar izlenir (Her iki tasarım yöntemi için algoritma aynıdır) :

1. Tasarlanacak band geçiren filtrenin özelliklerine göre Butterworth, Chebyshev veya Eliptik yaklaşıklık tiplerinden uygun olanı seçilir.

2. Verilen band geçiren filtre özelliklerini sağlayan $H(s)$ transfer fonksiyonu 2.4 alt bölümünde anlatılan algoritma yardımıyla bulunur.
3. Quad tasarım yöntemi için 4.4 alt bölümünde elde edilen bağıntılar yardımıyla biquadratik transfer fonksiyonları, geribesleme katsayıları ve iyilik faktörleri bulunur. Kaskad tasarım yöntemi için ise 4.5 alt bölümünde elde edilen bağıntılar yardımıyla biquadratik transfer fonksiyonları ve iyilik faktörleri bulunur.
4. Elde edilen quad /kaskad bloklar filtre dinamik değişimi maksimum olacak şekilde sıralanır. Sıralama yapıldıktan sonra biquadratik bloklara ilişkin kazanç sabitleri ve yeni geri besleme katsayıları Bölüm 6'da verilecek olan prosedüre göre belirlenir.
5. Devredeki OTA iletkenlikleri ve kapasite elemanlarının değerleri belirlenir.
6. "Clipping" ve "slew-rate" sorunlarını ortadan kaldıracak şekilde filtre girişine uygulanabilecek maksimum giriş geriliminin genliği hesaplanır.

BÖLÜM 5

OTA MODELLERİ VE AKTİF OTA-C BİQUADRATİK DEVRELERİ

5.1 Giriş

Aktif filtrelerin çoğu işlemsel kuvvetlendiricilerle gerçekleştirilmesine rağmen işlemsel kuvvetlendiricilerden oluşmuş filtrelerin uygulamalarının sınırlı kaldığı görülmektedir. İşlemsel kuvvetlendirici kazancının frekansa bağlı olmasından dolayı, yüksek frekanslara çıkıldıkça filtre karakteristiğinde önemli bozulmalar oluşmaktadır. Bu nedenle işlemsel kuvvetlendiricilerle gerçekleştirilen aktif filtrelerin uygulamalarının alçak frekanslarla sınırlı kaldığı görülmektedir [1].

OTA elemanı kullanılarak tasarlanan filtrelerle daha yüksek frekanslarda çalışmak mümkün olmaktadır. Sözü edilen sakıncadan dolayı yüksek frekanslarda çalışan aktif filtrelerin tasarımında işlemsel kuvvetlendiricilerin yerine OTA'ların kullanılması kaçınılmaz hale gelmiştir. OTA'lar genellikle işlemsel kuvvetlendiricilerden oldukça büyük band genişliğine sahiptirler. Ayrıca, OTA elemanları daha basit yapıya sahiptirler ve bir kontrol akımı aracılığıyla elektronik olarak kolayca ayar edilebilirler. OTA'ların ayarlanabilme özelliğinden yararlanılarak OTA ve kapasite elemanlarıyla programlanabilir aktif filtreler gerçekleştirilebilir [5], [6]. OTA'larla oluşturulan analog filtreler genellikle işlemsel kuvvetlendiricili eşdeğerlerinden daha az sayıda eleman içermektedir. Bunun dışında, OTA ve kapasiteler tümleşik devre teknolojisiyle kolayca gerçekleştirilebilirler.

Bu bölümde, OTA elemanı ve modelleri kısaca tanıtılıp, OTA - C elemanlarıyla oluşturulan biquadlar incelenecektir.

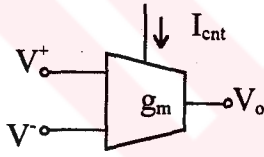
5.2 OTA Elemanı ve Modelleri

5.2.1 İdeal model

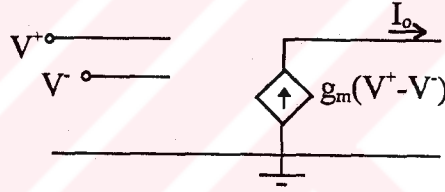
İdeal bir OTA, gerilim kontrollü akım kaynağından oluşmaktadır ve çıkış akımının ifadesi,

$$I_o = g_m (V^+ - V^-) \quad (5.1)$$

biçiminde tanımlanabilir. Şekil 5.1'de ideal bir OTA'nın devre sembolü ve Şekil 5.2'de eşdeğer devresi gösterilmiştir. g_m kontrol akımı, I_{cntl} akımı değiştirilerek ayarlanabilir ve $g_m = k I_{cntl}$ şeklindedir. I_{cntl} ve g_m birbiriyle doğru orantılıdır [1].



Şekil 5.1 OTA devre sembolü

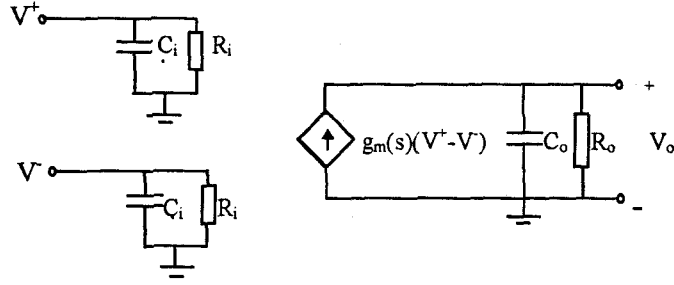


Şekil 5.2 İdeal eşdeğer devre

OTA kazancı doğrudan doğruya I_{cntl} akımının kontrolü ile ayarlanabileceği gibi aynı kontrol ucundan uygulanan bir gerilim ile de ayarlanabilir [7].

5.2.2 Gerçek OTA modeli

Pratik uygulamalarda, OTA'ların giriş ve çıkış empedanslarının ve g_m 'in frekansla değiştiği gözönünde bulundurulmalıdır. Gerçekte OTA'nın giriş ve çıkış empedansları sonlu değerler almaktadır. Şekil 5.3'de ideal olmayan OTA modeli gösterilmektedir. Burada, eviren ve evirmeyen uçlardan toprağa olan empedansların eşit olduğu varsayılmaktadır. Ayrıca, giriş uçları arasındaki empedans önemsenmektedir[7]. Giriş, çıkış kapasitelerinin varlığı ve g_m 'in frekansa bağlı olması, eleman davranışının bozularak filtrenin band genişliğinin, alçak frekans performansının azalmasına ve kararsızlığa sebep olmaktadır.

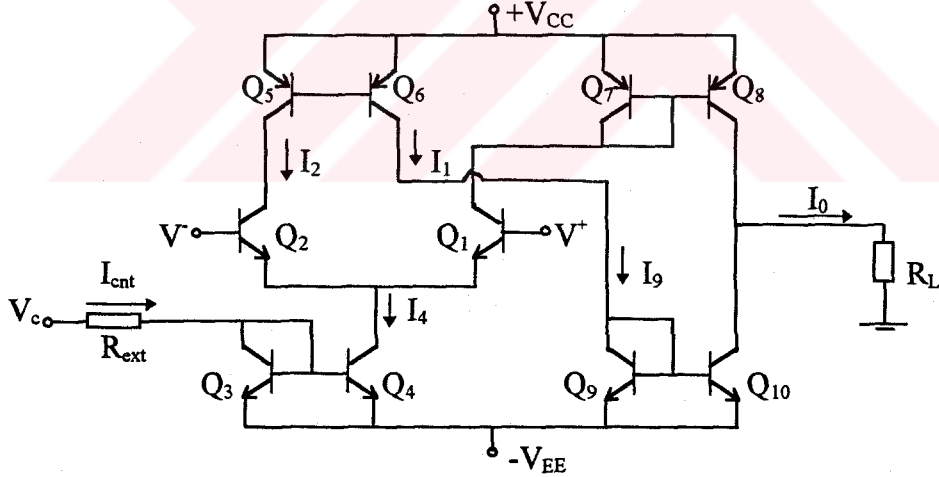


Şekil 5.3 İdeal olmayan OTA eşdeğer devresi

Ticari amaçla sunulmuş OTA'lar bipolar tekniğiyle üretilmişlerdir ve bazı pratik sınırlamaları vardır. Bazı OTA'ların girişlerine uygulanabilecek gerilim sınırlı olduğundan, bu OTA'lar filtre tasarımında kullanıldıklarında filtre dinamik davranışında sorunlar çıkabilir. Bu nedenle filtre dinamiğinin öneminin büyük olduğu uygulamalarda daha iyi tasarlanmış OTA'lar kullanılmalıdır.

5.2.3 Bipolar OTA

Bipolar tekniğiyle yapılmış olan OTA'nın iç yapısı Şekil 5.4'de gösterilmiştir.



Şekil 5.4 Bipolar OTA'nın iç yapısı

Q_1 ve Q_2 transistörleri bir fark kuvvetlendirici, Q_3 ve Q_4 ise bir akım aynası olarak çalışmaktadır. I_{cnt} akımı bir dış direnç (R_{ext}) ile kontrol edilir. Q_3 - Q_4 bir akım aynası oluşturduğundan $I_4 = I_{cnt}$ 'dir. I_4 akımı, Q_1 ve Q_2 'nin emetör akımlarının toplamına eşittir ($I_4 = I_1 + I_2$)

Q_5 - Q_6 akım aynasından $I_9=I_2$ ve Q_9 - Q_{10} akım aynasından $I_{10}=I_9=I_2$ yazılabilir. Benzer şekilde Q_7 - Q_8 akım aynasından $I_1=I_8$ ve düğüm denkleminde $I_o=I_8-I_{10}=I_1-I_2$ elde edilebilir. Buradan gerilim kazancı hesaplanırsa,

$$A_v = V_o / V_{in} = I_o R_L / V_{in} = g_m R_L \quad (5.2)$$

olarak bulunur. I_s , Q_1 ve Q_2 'nin ters doyma akımı olmak üzere,

$$I_{cnt} = I_1 + I_2 = I_s [\exp(V_1/V_T) + \exp(V_2/V_T)] \quad (5.3)$$

ifadesi yazılabilir. Sonuç olarak çıkış akımı,

$$I_o = I_1 - I_2 = I_{cnt} \frac{\exp((V_1 - V_2)/2V_T) - \exp(-(V_1 - V_2)/2V_T)}{\exp((V_1 - V_2)/2V_T) + \exp(-(V_1 - V_2)/2V_T)} \quad (5.4)$$

bulunabilir [9].

OTA'nın lineer bölgede çalışabilmesi için giriş geriliminin değişim aralığının çok küçük seçilmesi gerekir. Eğim ,

$$g_m = |dI_o/dV_{in}| = I_{cnt} / 2V_T \quad (5.5)$$

biçiminde bulunabilir. OTA'nın gerilim kazancı, kontrol akımı I_{cnt} 'ile değiştirilebilir. Oda sıcaklığında $V_T=26$ mV ve $g_m=I_{cnt}/(2V_T)=I_{cnt}/52\text{mV}$ olur. (5.1) bağıntısı kullanılarak,

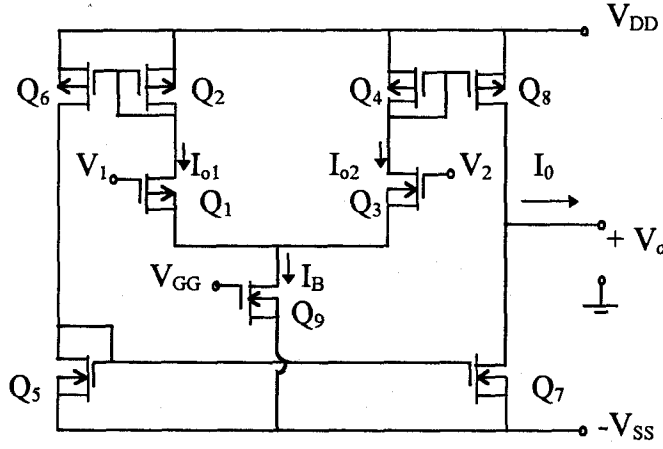
$$I_o = (I_{cnt} / 52\text{mV}) V_{in} \quad (5.6)$$

elde edilir.

5.2.4 CMOS - OTA

Simetrik CMOS - OTA yapısı Şekil 5.5'de gösterilmektedir. Eğiminin I_B kutuplama akımı ile kontrol edilebilmesi ve geniş bandlı olması, tümleştirmeye uygun ve basit

olmasından dolayı bu yapı yaygın bir kullanım alanına sahiptir. Bu devre yapısından, OTA-C filtre yapılarının yanısıra, analog çarpma devreleri ve yüksek frekans osilatörlerinin gerçekleştirilmesinde yararlanılmaktadır [13].



Şekil 5.5 Simetrik CMOS OTA yapısı

Q_9 doyma bölgesinde çalışarak, I_B kontrol akımını üretir. Q_1 ve Q_3 fark kuvvetlendiricisi görevini görmektedirler. (Q_2-Q_6) ve (Q_4-Q_8) akım aynaları, Q_1 ve Q_3 için dinamik yük olarak kullanılmıştır. OTA'nın eğimi [10]

$$G_m = B \sqrt{K_n \cdot I_B \cdot (W/L)_1} \quad (5.7)$$

çıkış akımının maksimum değeri,

$$I_{\text{omax}} = -I_{\text{omin}} = -B I_B \quad (5.8)$$

gerilim kazancı,

$$K_v = G_m R_o \quad (5.9)$$

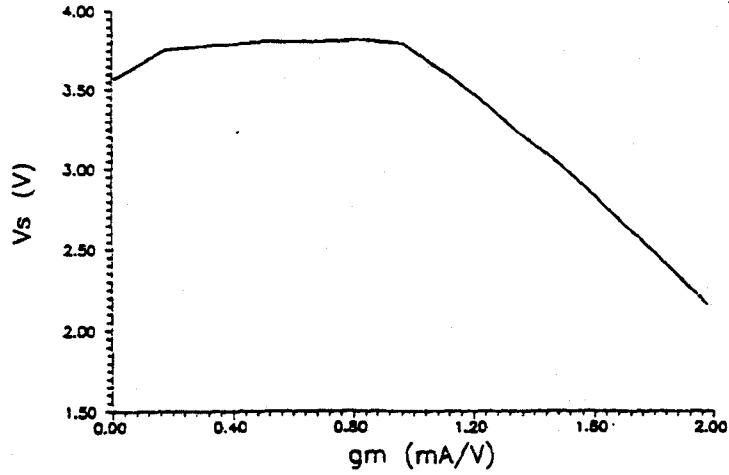
Tipik değerler Şekil 5.5'de kullanılan tranzistorlar için Tablo 5.1'de verilmiştir. Gerilim kaynağı ± 5 V ve kontrol gerilimi $V_{GG} = -3.24$ V seçilirse, OTA'nın eğimi (g_m) $I_B = 336 \mu\text{A}$ için 1.33 mA/V 'a eşit olur [11].

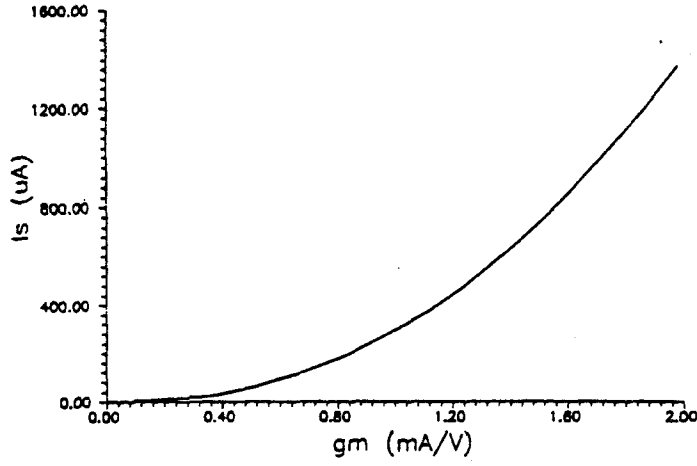
Tablo 5.1 Şekil 5.5’de kullanılan tranzistorlar için tipik değerler

Tranzistor	W(μm)	L(μm)
Q ₁	30	3
Q ₂	12	3
Q ₃	30	3
Q ₄	12	3
Q ₅	12	3
Q ₆	36	3
Q ₇	12	3
Q ₈	36	3
Q ₉	45	3

OTA’nın çıkış işareti belirli bir değeri aştığında lineer çalışma bölgesi dışına çıkar. Devredeki herhangi bir OTA’nın çıkış gerilimi doyma değerini aşarsa, devre çıkışındaki işaret kırılır. Çıkış akımının doyma akım değerini aşması halinde “Slew - rate” sorunu ortaya çıkacaktır [12].

OTA’nın eğimine bağlı olarak V_s ve I_s değerleri değişecektir. Tablo 5.1’deki CMOS’lar için V_s ve I_s ’in g_m ile değişim eğrileri Şekil 5.6 ve Şekil 5.7’de gösterilmektedir [11].

Şekil 5.6 V_s ’nin g_m ile değişimi



Şekil 5.7 I_s 'nin g_m ile değişimi

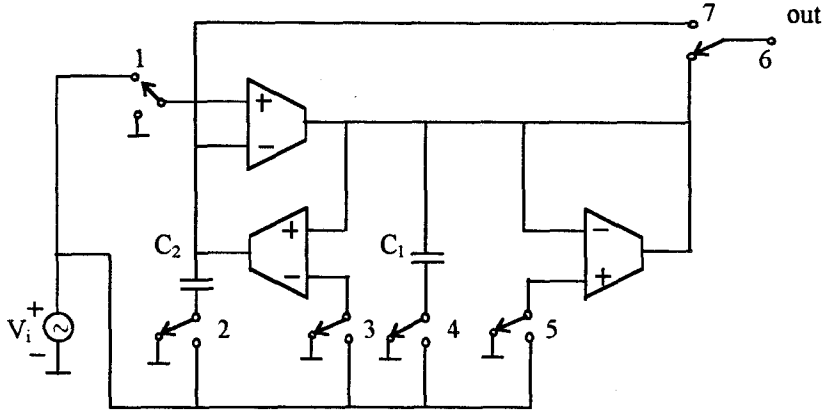
5.3 Literatürde Verilmiş Olan Aktif OTA-C Biquadratik Devreleri

Bu alt bölümde literatürde karşılaşılan bazı OTA - C biquadları incelenecektir [13], [14]. İçlerinden en avantajlı olanlar (az sayıda eleman, kapasite elemanlarının topraklı olması, duyarlık v.b.) quad/kaskad bloklarına ilişkin biquadratik temel altdevrelerde kullanılacaktır.

1.Biquad

İkili - integratör çevrimi yöntemine göre tasarlanmış az duyarlıklı genel OTA - C biquad devresi Şekil 5.8'de görülmektedir [15]. Bu devrede C_1 ve C_2 kapasiteleri topraklı değildir ve bu bir dezavantajdır. Çünkü topraklı kapasite tümleşik devre teknolojisi açısından doğal bir elemandır ve MOS tranzistorun giriş kapasitesi ile gerçekleştirilebilir. Ayrıca topraklı kapasite, paralel parazitik kapasitif etkileri yok edebilir ve tümleşik devrelerdeki en ucuz lineer kapasitedir [2].

Şekil 5.8'deki devrede S_1 , S_2 , S_3 , S_4 , S_5 anahtarları kullanılarak 1, 2, 3, 4, 5 düğümleri toprağa veya giriş gerilim kaynağına bağlanır. Filtre çıkışı 6 veya 7 düğümlerinden alınır.



Şekil 5.8 Tablo 5.2'deki transfer fonksiyonlarını gerçekleyen genel OTA - C biquad devresi

1, 2, 3, 4, 5 düğümlerindeki gerilimler [15],

$$V_j = k_j V_i \quad , \quad j=1, 2, \dots, 5 \quad (5.10)$$

şeklinde ifade edilebilir. Bu bağıntıda V_i giriş gerilimidir. k_j ise 0 veya 1 değerlerini alır [15].

OTA'ların giriş empedanslarının ihmal edilmesi durumunda, 6 ve 7 nolu düğümlerin gerilimlerinin frekans domenindeki ifadeleri [15],

$$\frac{V_6(s)}{V_i(s)} = \frac{N_6(s)}{D(s)} \quad \frac{V_7(s)}{V_i(s)} = \frac{N_7(s)}{D(s)} \quad (5.11.a)$$

biçiminde yazılabilir. Bu bağıntıda geçen ifadeler,

$$D(s) = s^2 + s \frac{g_{m3}}{C_1} + \frac{g_{m1}g_{m2}}{C_1C_2} \quad (5.11.b)$$

$$N_6(s) = s^2k_4 + \frac{s}{C_1} [g_{m1}(k_1 - k_2) + g_{m3}k_5] + \frac{g_{m1}g_{m2}}{C_1C_2}k_3 \quad (5.11.c)$$

$$N_7(s) = s^2 k_2 + \frac{s}{C_1} [g_{m3} k_2 + g_{m2} (k_3 + k_4)] + \frac{g_{m2}}{C_1 C_2} [g_{m1} k_1 + g_{m3} (k_3 + k_5)] \quad (5.11.d)$$

$$w_0 = \sqrt{\frac{g_{m1} g_{m2}}{C_1 C_2}} \quad B = \frac{g_{m3}}{C_1} \quad (5.11.e)$$

şeklinde. Burada w_0 kutup genliğini, B ise band genişliğini göstermektedir [15]. Çeşitli koşullara göre elde edilen biquad tipleri ve bunlara ait pay polinomlarının ifadeleri Tablo 5.2’de gösterilmiştir.

Tablo 5.2 Şekil 5.8’deki devreye ilişkin çeşitli koşullara göre elde edilen filtre tipleri

Filtre Tipi	Pay Polinomu	Giriş Düzğümleri, $k=1$	Çıkış Düzgümü	Topraklanan Düzğümler $k=0$
AG (1.Tip)	$(g_{m1}g_{m2})/(C_1C_2)$	3 veya 1	6 7	1, 2, 3, 4, 5 2, 3, 4, 5
AG (2.Tip)	$(g_{m2}g_{m3})/(C_1C_2)$	5	7	1, 2, 3, 4
BG (1.Tip)	$(sg_{m1})/C_1$	1 veya 4	6 7	2, 3, 4, 5 1, 2, 3, 5
BG (2.Tip)	$(sg_{m3})/C_1$	5	6	1, 2, 3, 4
YG	s^2	4	6	1, 2, 3, 5
Notch	$s^2 + (g_{m1}g_{m2})/(C_1C_2)$	3, 4 veya 1, 2, 3, 4	6 6	1, 2, 5 5

2. Biquad

Literatürde ikili - integratör çevrimi (two -integrator - loop), [16], diye bilinen az duyarlıklı temel yapıya dayanarak tasarlanmış genel OTA - C biquad devresi Şekil 5.10'da gösterilmiştir [1]. Şekil 5.9'da ise biquad tasarımında kullanılan temel OTA yapıları verilmektedir. Şekil 5.10'daki devrede V_a , V_b , V_c , giriş olarak düşünüldüğünde devrenin çıkış gerilimleri,

$$V_1 = \frac{sC_2g_{m1}(g_{m5}V_a - g_{m4}V_c) + g_{m1}g_{m4}V_b}{D(s)} \quad (5.12.a)$$

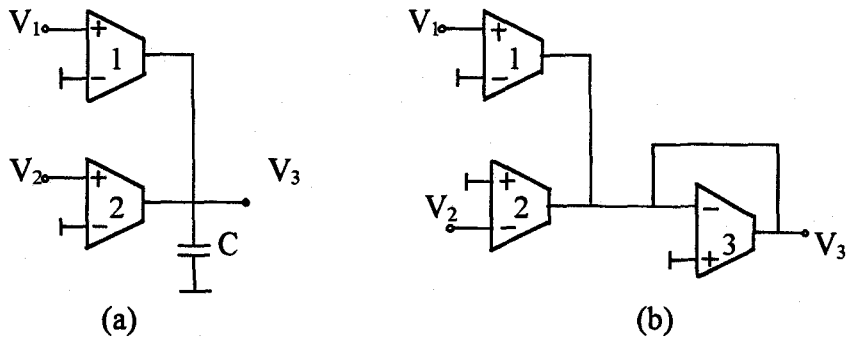
$$V_2 = \frac{(sC_1g_{m2}g_{m5} - g_{m1}g_{m2}g_{m3})V_b + g_{m1}g_{m2}(g_{m4}V_c - g_{m5}V_a)}{D(s)} \quad (5.12.b)$$

$$V_3 = \frac{s^2C_1C_2g_{m4}V_c + s(C_2g_{m1}g_{m3}V_a - C_1g_{m2}g_{m4}V_b) + g_{m1}g_{m2}g_{m4}V_a}{D(s)} \quad (5.12.c)$$

şeklinde bulunur. Bu ifadelerde geçen payda polinomu,

$$D(s) = C_1C_2g_{m5} \left[s^2 + s \frac{g_{m1}g_{m3}}{C_1g_{m5}} + \frac{g_{m1}g_{m2}g_{m4}}{C_1C_2g_{m5}} \right] \quad (5.12.d)$$

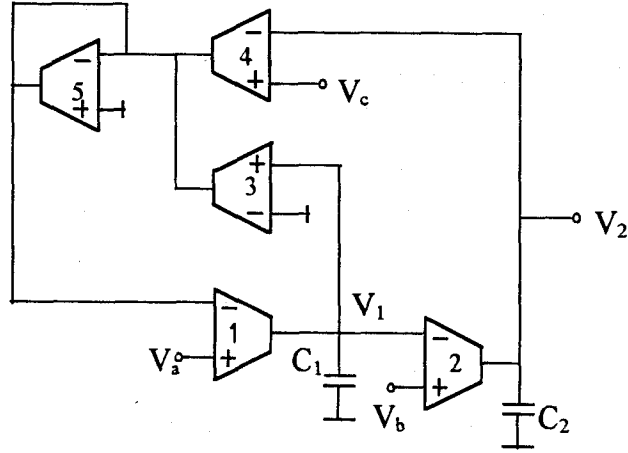
biçimindedir.



Şekil 5.9 Temel OTA - C yapıları

a) İntegral alıcı devre, $V_3 = (1/sC) (g_{m1}V_1 - g_{m2}V_2)$

b) Toplayıcı, $V_3 = (g_{m1}/g_{m3})V_1 - (g_{m2}/g_{m3})V_2$



Şekil 5.10 Genel OTA - C biquad

Tablo 5.3’de çeşitli koşullar için elde edilen filtre tipleri verilmiştir. Bu yapı kullanılarak, Tablo 5.3’deki koşullara uygun olarak AG,YG, BG, Notch türünden biquadlar gerçekleştirilebilir.

Tablo 5.3 Şekil 5.10’daki biquada ilişkin değişik koşullar ve filtre tipleri

Filtre Tipi	Pay Polinomu	Giriş Uçları	Çıkış Ucu	
AG	$g_{m1}g_{m2}g_{m4}$	$V_c = V_i$	V_2	$V_a = V_b = 0$
YG	$s^2C_1C_2g_{m4}$	$V_c = V_i$	V_3	$V_a = V_b = 0$
BG	$-s C_2g_{m1}g_{m4}$	$V_c = V_i$	V_1	$V_a = V_b = 0$
Notch	$s^2C_1C_2g_{m4} + g_{m1}g_{m2}g_{m4}$	$V_a = V_b = V_c = V_i$	V_3	$C_2g_{m1}g_{m3} = C_1g_{m2}g_{m4}$

3.Biquad

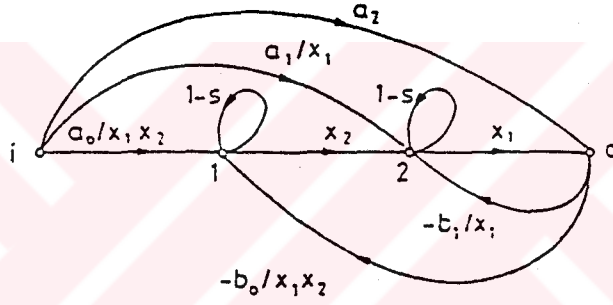
Bir başka biquad devre yapısı olarak, Kaynak [11]’de yer alan, minimum OTA elemanı gerektiren bir OTA-C biquad tasarımı ele alınacaktır. Aktif filtre tasarımında

çok güçlü bir yöntem olan işaret akış diyagramına dayanan bu prosedür yardımıyla minimum OTA ve topraklı kapasiteye sahip biquadratik devre yapılarını tasarlamak mümkün olmaktadır [11].

Genel biquadratik transfer fonksiyonu,

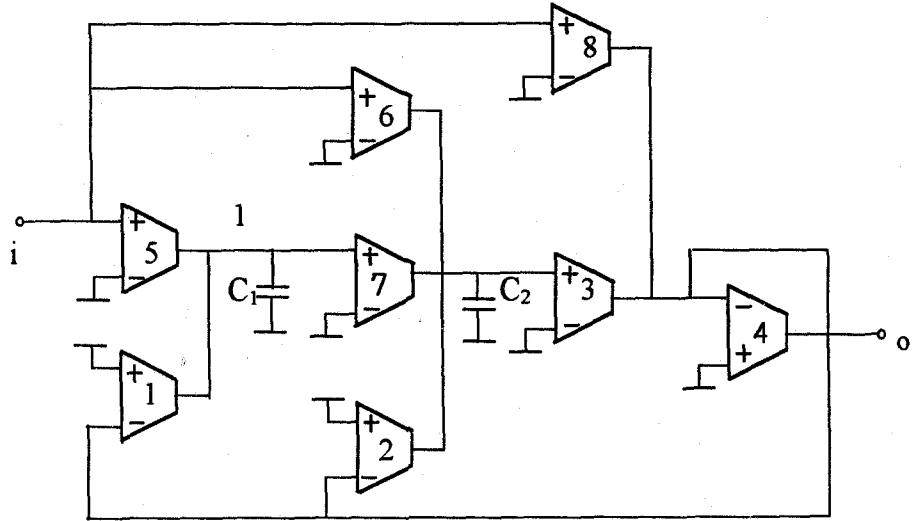
$$G(s) = \frac{V_o}{V_i} = \frac{a_2 s^2 + a_1 s + a_0}{s^2 + b_1 s + b_0} \quad (5.13)$$

biçiminde verilsin. Şekil 5.11'de bu transfer fonksiyonuna ait işaret akış diyagramı gösterilmektedir.



Şekil 5.11 $G(s)$ 'e karşı düşen işaret akış diyagramı

Şekil 5.12'de bu akış diyagramına ait OTA-Topraklı Kapasite devresi gösterilmektedir.



Şekil 5.12 $G(s)$ 'i gerçekleyen OTA - C devresi

Bu devreye ilişkin tasarım denklemleri,

$$g_{m1} / C_1 = b_0 / x_1 x_2 \quad (5.14.a)$$

$$g_{m2} / C_2 = b_1 / x_1 \quad (5.14.b)$$

$$g_{m3} / g_{m4} = x_1 \quad (5.14.c)$$

$$g_{m5} / C_1 = a_0 / x_1 x_2 \quad (5.14.d)$$

$$g_{m6} / C_2 = a_1 / x_1 \quad (5.14.e)$$

$$g_{m7} / C_2 = x_2 \quad (5.14.f)$$

$$g_{m8} / g_{m4} = a_2 \quad (5.14.g)$$

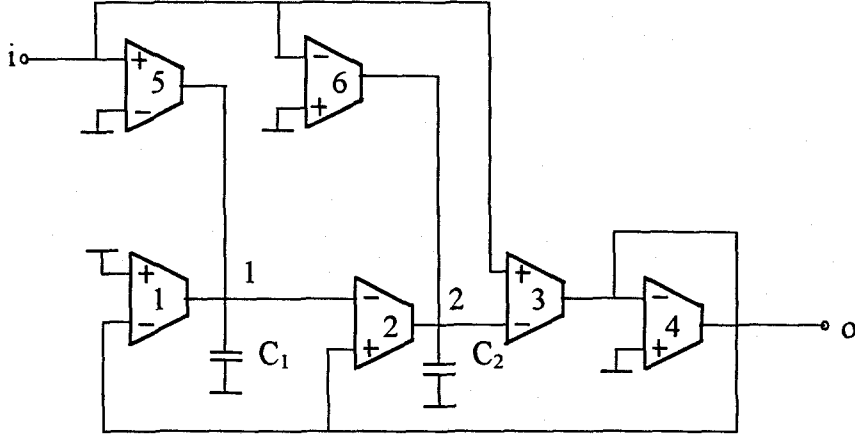
biçiminde verilebilir. x_1 ve x_2 bağımsız parametreler olup tasarıma esneklik kazandırır. $x_1 = -a_2$ ve $x_2 = -b_1 / a_2$ seçildiğinde Şekil 5.11'deki bazı dal çiftlerinin transmittansları eşit genlik ve ters işaretle olmaktadır [11]. Bu durum,

$$t_{02} = t_{12} = b_1 / a_2 \quad (5.15.a)$$

$$t_{i0} = -t_{20} = a_2 \quad (5.15.b)$$

şeklinde ifade edilebilir. Bu bağıntılarda t_{pq} , (p, q) dalına ait transmittansı gösterir. Böylece bu çiftlerin herbiri, iki OTA yerine bir OTA ile gerçekleştirilebilir. Bu sayede, toplam olarak iki OTA daha az kullanılmış olur [11].

Şekil 5.13'de bu duruma karşı gelen devre gösterilmektedir. Bu devrede OTA sayısı minimumdur. Genel bir biquadratik fonksiyonda 5 farklı katsayı olduğundan, bu keyfi katsayıları gerçekleştirmek için en az 5 OTA'ya gereksinim vardır. 6. OTA ise çıkıştaki toplayıcı devreyi gerçekleştirmek içindir. Kullanılacak OTA sayısı, bazı katsayıların sıfır olması durumunda daha da azalacaktır [11].



Şekil 5.13 $G(s)$ 'i gerçekleyen minimum OTA - C biquad devresi

Bu filtre için tasarım eşitlikleri,

$$g_{m1} / C_1 = b_0 / b_1 \quad (5.16.a)$$

$$g_{m2} / C_2 = b_1 / a_2 \quad (5.16.b)$$

$$g_{m3} / g_{m4} = a_2 \quad (5.16.c)$$

$$g_{m5} / C_1 = a_0 / b_1 \quad (5.16.d)$$

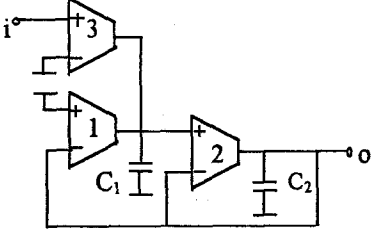
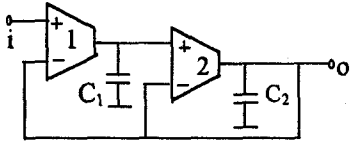
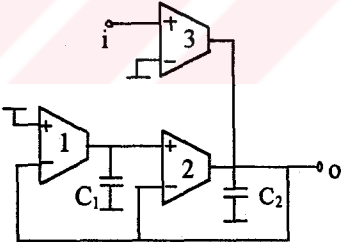
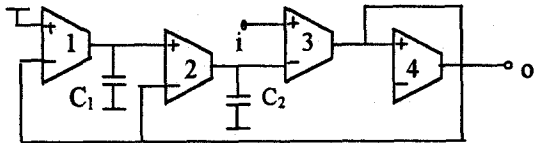
$$g_{m6} / C_2 = a_1 / a_2 \quad (5.16.e)$$

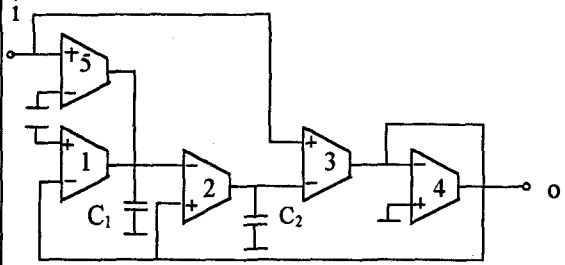
biçiminde verilebilir. Payda polinomu ise,

$$D(s) = s^2 + \frac{g_{m2}g_{m3}}{g_{m4}C_2}s + \frac{g_{m1}g_{m2}g_{m3}}{g_{m4}C_1C_2} \quad (5.17)$$

biçimindedir. Sözü edilen prosedüre göre elde edilmiş çeşitli tipten minimum OTA-C filtreleri Tablo 5.4'de verilmiştir [11].

Tablo 5.4 Şekil 5.11'deki işaret akış grafi kullanılarak oluşturulan OTA-C filtreleri

Filtre Tipi ve Transfer Fonksiyonu	OTA-C Gerçeklemesi
1. AG $a_0/(s^2 + b_1s + b_0)$	(a)  $\frac{g_{m1}}{C_1} = \frac{b_0}{b_1}, \quad \frac{g_{m2}}{C_2} = b_1$ $\frac{g_{m3}}{C_1} = \frac{a_0}{b_1}$ (b)  $\frac{g_{m1}}{C_1} = \frac{b_0}{b_1}, \quad \frac{g_{m2}}{C_2} = b_1$
2. BG $a_1s/(s^2 + b_1s + b_0)$	 $\frac{g_{m1}}{C_1} = \frac{b_0}{b_1}, \quad \frac{g_{m2}}{C_2} = b_1, \quad \frac{g_{m3}}{C_2} = a_1$
3. YG $a_2s^2/(s^2 + b_1s + b_0)$	 $\frac{g_{m1}}{C_1} = \frac{b_0}{b_1}, \quad \frac{g_{m2}}{C_2} = \frac{b_1}{a_2}, \quad \frac{g_{m3}}{g_{m4}} = a_2$

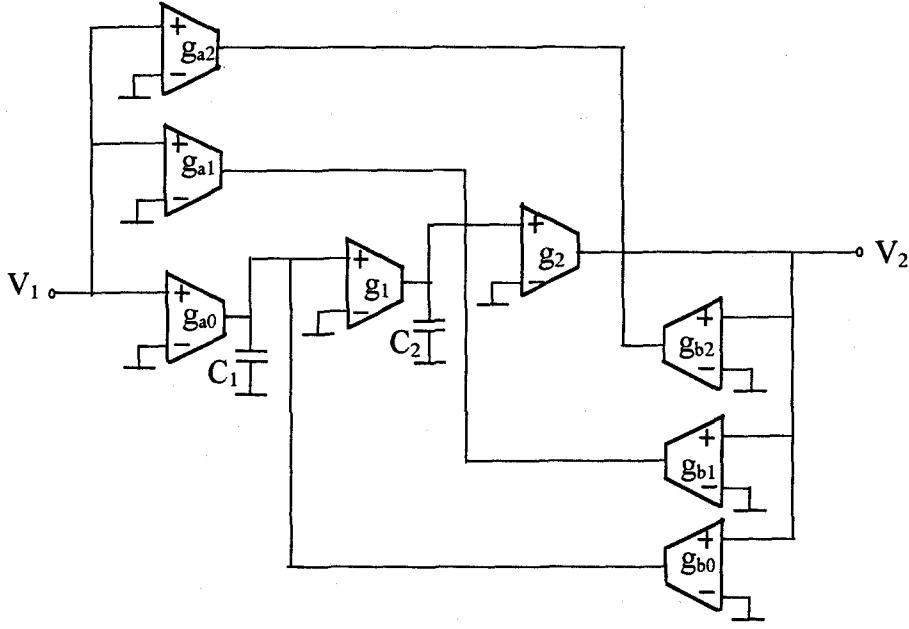
Filtre Tipi ve Transfer Fonksiyonu	OTA-C Gerçeklemesi
4. Notch ($a_2s^2 + a_0$)/($s^2 + b_1s + b_0$)	 $\frac{g_{m1}}{C_1} = \frac{b_0}{b_1}, \quad \frac{g_{m2}}{C_2} = \frac{b_1}{a_2}, \quad \frac{g_{m5}}{C_1} = \frac{a_0}{b_1}, \quad \frac{g_{m3}}{g_{m4}} = a_2$

4. Biquad

Genel amaçlı bir OTA - C biquad devresi, [17], Şekil 5.14'de gösterilmektedir. Devredeki kapasite elemanları topraklıdır. C_1 , g_1 ve C_2 , g_2 bölümleri birer integral alıcı devreyi temsil etmektedir. C_1 ve C_2 'den akan OTA akımlarının toplamının integrasyonu ile g_1 ve g_2 'nin giriş gerilimleri elde edilmektedir. Biquadratik transfer fonksiyonu, doğrudan bu akımların toplanması ve integre edilmesinden bulunmaktadır [17].

$$H(s) = \frac{V_2}{V_1} = \frac{a_2s^2 + a_1s + a_0}{b_2s^2 + b_1s + b_0} \quad (5.18)$$

Bu bağıntıda, $a_0 = g_{a0} g_1 g_2$, $a_1 = g_{a1} g_2 C_1$, $a_2 = g_{a2} C_1 C_2$, $b_0 = g_{b0} g_1 g_2$, $b_1 = g_{b1} g_2 C_1$ ve $b_2 = g_{b2} C_1 C_2$ 'dir. g_{ai} ve g_{bi} transkondüktansları yardımıyla, a_i ve b_i katsayıları ayrı ayrı ayarlanabilir [17]. Filtrenin frekans cevabı, lineer bir şekilde transkondüktanslar değiştirilerek ayarlanabilir. Tablo 5.5'de çeşitli koşullara göre elde edilen biquad tipleri ve bunlara ait pay polinomlarının ifadeleri yer almaktadır.



Şekil 5.14 Genel amaçlı bir OTA - C biquad devresi

Tablo 5.5 Şekil 5.14'deki biquad için çeşitli koşullara göre elde edilen filtre tipleri

Filtre Tipi	Pay polinomu	Koşul
AG	a_0	$a_1 = a_2 = 0$, $g_{a1} = g_{a2} = 0$
YG	$a_2 s^2$	$a_0 = a_1 = 0$, $g_{a0} = g_{a1} = 0$
BG	$a_1 s$	$a_0 = a_2 = 0$, $g_{a0} = g_{a2} = 0$
Notch	$a_2 s^2 + a_0$	$a_1 = 0$, $a_0 = a_2$, $g_{a1} = 0$
TG (Tüm Geçiren)	$a_2 s^2 - a_1 s + a_0$	$a_0 = b_0$, $a_1 = -b_1$, $a_2 = b_2$

İncelenen dört biquadın avantajlı ve dezavantajlı yönleri aşağıdaki biçimde sıralanabilir :

Şekil 5.8'deki devrede kapasiteler topraklı değildir. Bu daha önce anlatıldığı gibi devrenin olumsuz bir yönüdür.

Şekil 5.10'daki devrede kapasiteler topraklı ve eleman sayısı azdır. Farklı filtre türleri için işaret giriş ve çıkış uçları değiştiğinden devre ek anahtarlama devrelerine ihtiyaç duymaktadır. Bu ise tümleşik devrenin alanını artıracak gibi, ek bir maliyet de getirecektir.

Şekil 5.14'deki devre ile her türden filtre gerçekleştirilebilir. Anahtarlama sorunu yoktur ve kapasiteler topraklıdır. Bu avantajlarına rağmen devredeki eleman sayısının fazla olması bir dezavantajdır. Bundan dolayı, yüksek dereceden bir tümleşik devrenin alanı büyük olur ve maliyet artar.

3.biquad yapısı olarak verilen biquadratik devre yapısı, Tablo 5.4'den anlaşılacağı gibi filtre türüne göre değişmektedir. Eleman sayısı, özellikle AG ve BG filtre için, diğer biquadlara göre daha azdır. Anahtarlama sorunu yoktur, kapasiteler topraklıdır. Eleman sayısının azlığı nedeniyle yüksek dereceden alçak ve band geçiren filtrelerin tasarımında Tablo 5.4'deki devrelerden yararlanılacaktır.

BÖLÜM 6

YÜKSEK DERECEDEKİ QUAD VE KASKAD FİLTRELERİNİN DİNAMİK DEĞİŞİMİ

Tasarım denklemlerinin daha önceki bölümlerde belirlenmesine rağmen, devrenin gerçekleştirilmesi hala tek değildir. Alt devrelere ilişkin kazanç değerlerine gelişigüzel değerler atanabilir. Bu değerler, devrenin dinamik davranışını iyileştirecek biçimde tek olarak seçilmelidir. Yani, biquadratik blokların kazanç katsayılarının belirlenmesi, aktif filtrelerin dinamik değişimi üzerinde büyük rol oynamaktadır [18]. Bu bölümde Kaynak [3]'te verilmiş olan dinamik değişimi artırmaya yönelik prosedür, daha önceki bölümlerde elde edilmiş olan tasarım denklemlerine uygulanacaktır.

6.1 Quad ve Kaskad Yönteminde Dinamik Değişimin İyileştirilmesi

Şekil 6.1, Şekil 6.2, Şekil 6.3 ve Şekil 6.4'te sırasıyla quad alçak geçiren, kaskad alçak geçiren, quad band geçiren, kaskad band geçiren filtrelerin dinamik değişimini inceleyebilmek için blok diyagramlar verilmiş ve bu devrelere ilişkin kritik düğüm gerilimleri şekillerde gösterilmiştir. Quad/kaskad tasarım yöntemleri için aşağıda verilen tanımlardan yararlanılacaktır.

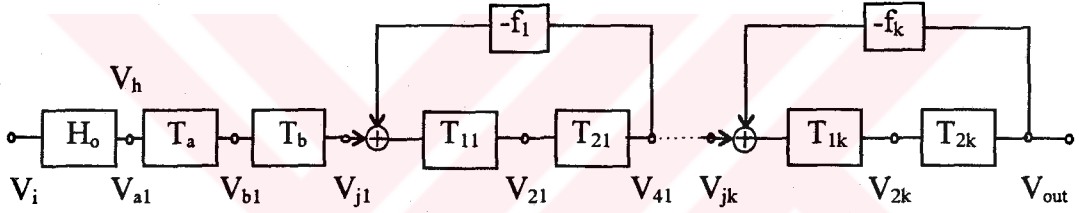
$$C_{ji\min} = \min_{w_L < w < w_H} |V_{ji}| = \min_{w_L < w < w_H} \left| \frac{TV_{in}}{\prod_{i=1}^k T_{QK_i}} \right| \quad (6.1.a)$$

$$C_{ji\max} = \max_{0 < w < \infty} |V_{ji}| = \max_{0 < w < \infty} \left| \frac{TV_{in}}{\prod_{i=1}^k T_{QK_i}} \right| \quad (6.1.b)$$

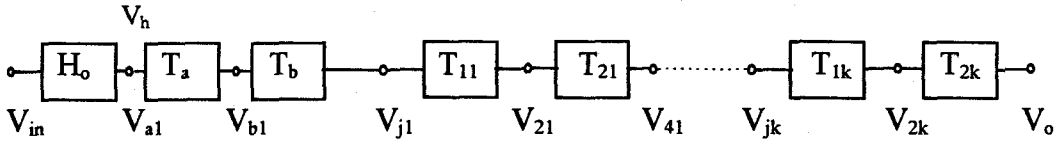
$$C_{2i\min} = \min_{w_L < w < w_H} |V_{2i}| = \min_{w_L < w < w_H} \left| \frac{TV_{in}}{T_{2i} \prod_{j=i+1}^k T_{QK_i}} \right| \quad (6.1.c)$$

$$C_{2i\max} = \max_{0 < w < \infty} |V_{2i}| = \max_{0 < w < \infty} \left| \frac{TV_{in}}{T_{2i} \prod_{j=i+1}^k T_{QK_i}} \right| \quad (6.1.d)$$

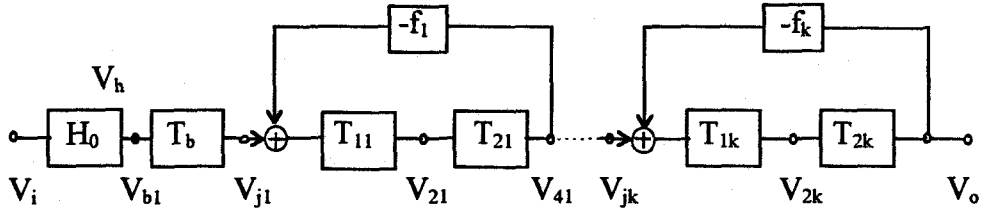
Bu bağıntılarda geçen k , quad/kaskad bloğu sayısıdır. $i = 1, 2, 3, \dots, k$ dir ve i , quad /kaskad numarasını göstermektedir. C_{ji} ve C_{2i} sırasıyla, i numaralı quad/kaskad bloğunun j ve 2 ile gösterilen düğümlerindeki gerilimleri ifade etmektedir [3].



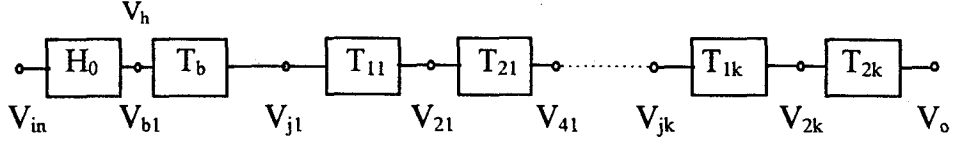
Şekil 6.1 Yüksek dereceden alçak geçiren quad filtresine ilişkin kritik gerilimler



Şekil 6.2 Yüksek dereceden alçak geçiren kaskad filtresine ilişkin kritik gerilimler



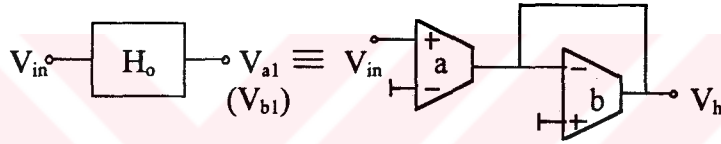
Şekil 6.3 Yüksek dereceden band geçiren quad filtresine ilişkin kritik gerilimler



Şekil 6.4 Yüksek dereceden band geçiren kaskad filtresine ilişkin kritik gerilimler

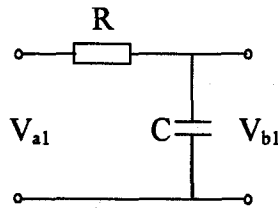
T ilgilenilen gerilim transfer fonksiyonudur. T_{QKi} quad tasarım için T_{Qi} , kaskad tasarım için T_{ki} 'dir. w_L , w_H ise ilgilenilen frekans bandının sırasıyla alt ve üst sınırlarını göstermektedir.

Blok diyagramlardaki H_o kazançlı giriş bloğu Şekil 6.5'deki devre ile gerçekleştirilebilir.



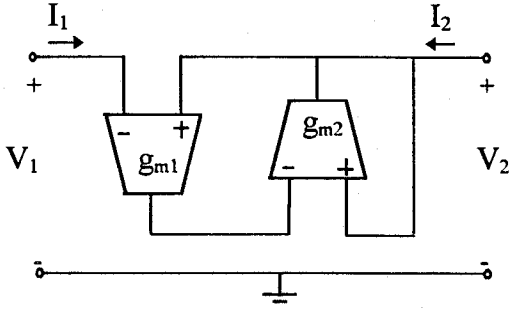
Şekil 6.5 H_o kazançlı giriş bloğuna ilişkin alt devre

Şekil 6.1 ve Şekil 6.2'deki blok diyagramlarda T_a bloğu birinci derecedendir ve Şekil 6.6.a'da gösterilen devre ile gerçekleştirilebilir.

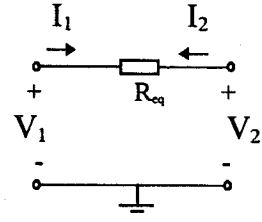


Şekil 6.6.a T_a bloğuna ilişkin RC devresi

T_a bloğu aynı zamanda R ve C elemanlarından oluşan Şekil 6.6.a'daki devre yerine bir OTA - C devresi ile de gerçekleştirilebilir. Bunun için Şekil 6.6.a'daki R direnci yerine Şekil 6.6.b'deki OTA'lardan oluşan devre kullanılabilir [8].



Şekil 6.6.b OTA'larla direnç gerçekleştirilmesi



Şekil 6.6.c Eşdeğer devre

Şekil 6.6.b'deki devrenin eşdeğeri Şekil 6.6.c'de olduğu gibi gösterilebilir ve bu devreler için aşağıdaki bağıntılar yazılabilir:

$$I_1 = -g_{m1}(V_2 - V_1) \quad , I_2 = -g_{m2}(V_1 - V_2) \quad (6.2.a)$$

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} g_{m1} & -g_{m1} \\ -g_{m2} & g_{m2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \quad (6.2.b)$$

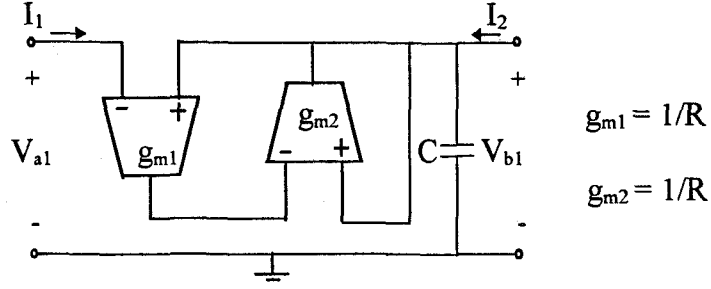
$g_{m1} = g_{m2} = g_m$ seçilirse,

$$R_{eq} = 1/g_m \quad (6.2.c)$$

bulunur. Şekil 6.6.b'deki devreyi R direncini elde edebilmek için kullandığımıza göre, $R_{eq} = R$ olacaktır ve böylece,

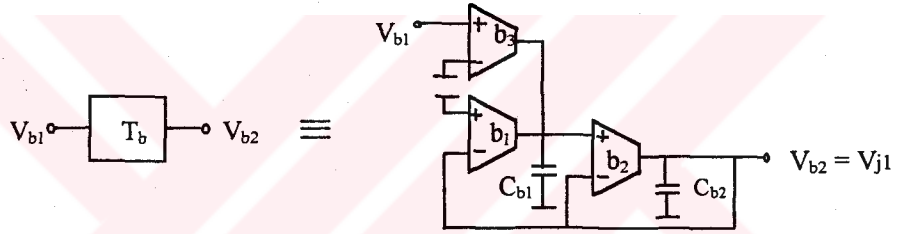
$$R = R_{eq} = 1/g_m \quad (6.2.d)$$

bağıntısı elde edilir. En sonunda Şekil 6.6.a'daki R, C elemanlarından oluşan devre Şekil 6.6.d'deki gibi bir OTA-C devresi haline gelmiş olur. Yani, T_s bloğu Şekil 6.6.a ve Şekil 6.6.d'deki devrelerden herhangi biriyle gerçekleştirilebilir.

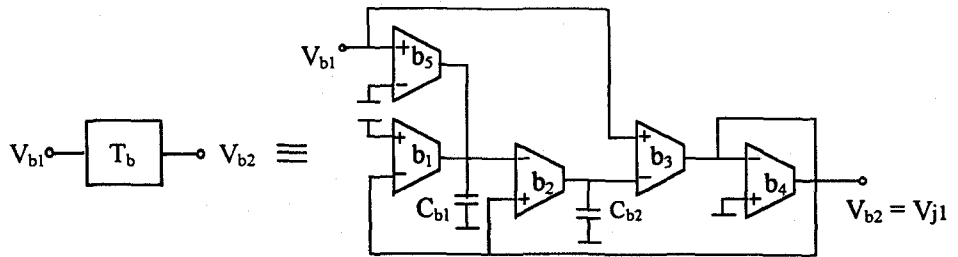


Şekil 6.6.d T_a bloğuna ilişkin OTA-C devresi

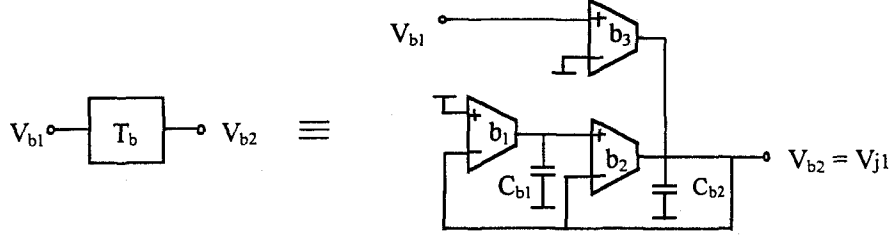
Şekil 6.1 ve Şekil 6.2'deki blok diyagramlarda görülen T_b blokları Butterworth, Chebyshev yaklaşıklık için Şekil 6.7.a, Eliptik yaklaşıklık için Şekil 6.7.b'de gösterilen OTA-C devresi ile, Şekil 6.3 ve Şekil 6.4'deki T_b blokları ise yaklaşıklık tipi ne olursa olsun Şekil 6.8'de gösterilen OTA-C devresi ile gerçekleştirilebilir.



Şekil 6.7.a Butterworth ve Chebyshev yaklaşıklık için alçak geçiren OTA-C quad/kaskad devresindeki T_b bloğuna ilişkin alt devre

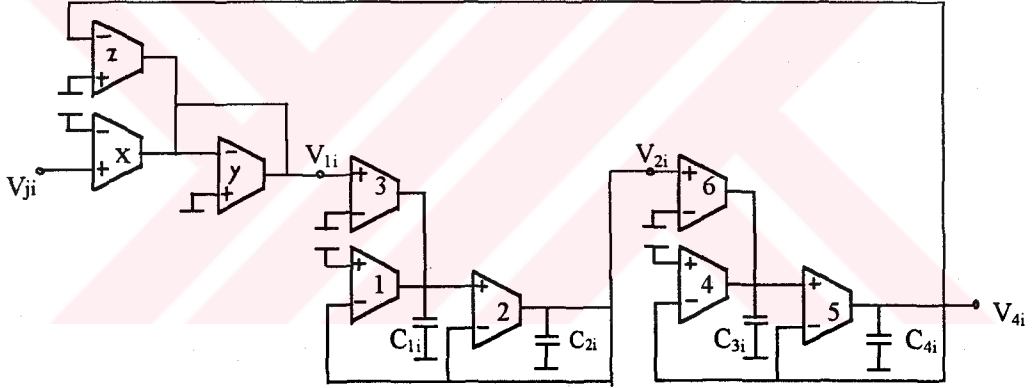


Şekil 6.7.b Eliptik yaklaşıklık için alçak geçiren OTA-C quad/kaskad devresindeki T_b bloğuna ilişkin alt devre

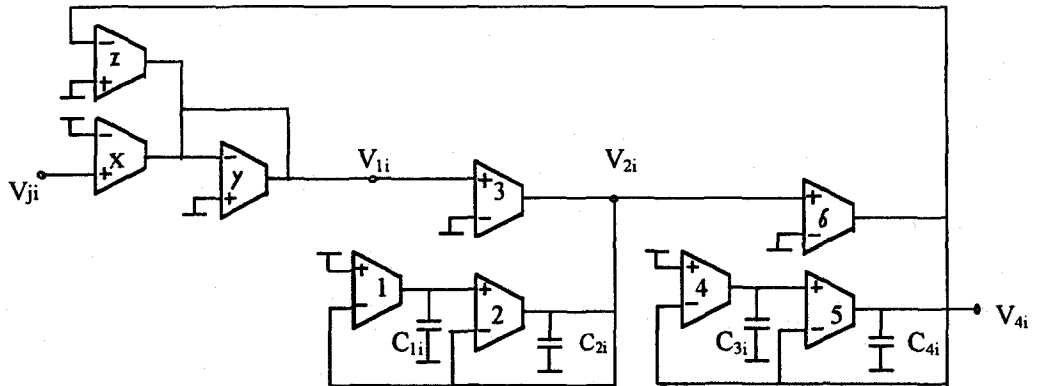


Şekil 6.8 Band geçiren OTA-C quad/kaskad devresindeki T_b bloğuna ilişkin alt devre

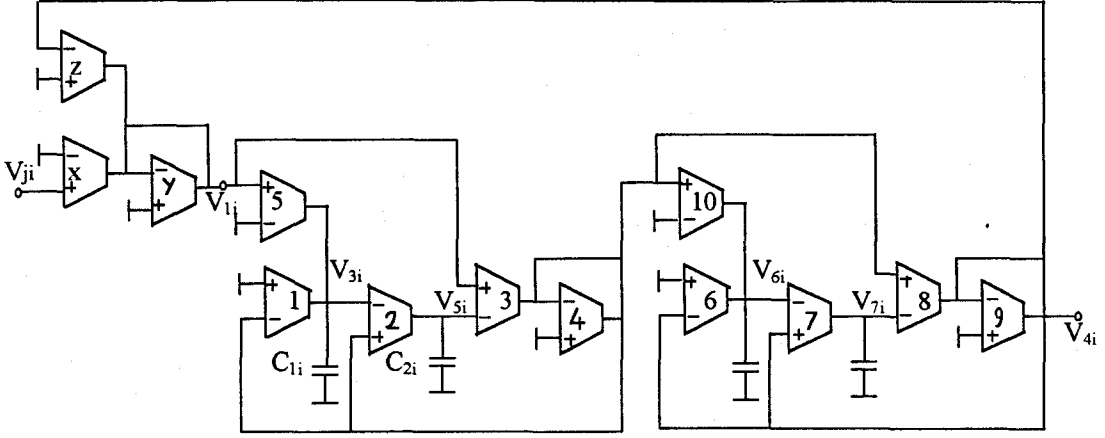
Blok diyagramlarda görülen quad bloklarına (T_{1i} , T_{2i} ve geribesleme bloklarından oluşan ana bloklar) ilişkin OTA-C devreleri çeşitli yaklaşıklık ve filtre tipleri için Şekil 6.9'da verilmektedir. Kaskad bloklarına (T_{1i} , T_{2i} bloklarından oluşan ana bloklar) ilişkin OTA-C devreleri, quad blokları için verilen bu devrelerde z OTA'sının kaldırılmasıyla elde edilebilirler.



Şekil 6.9.a Butterworth ve Chebyshev filtre için alçak geçiren OTA-C quad alt devresi



Şekil 6.9.b Butterworth ve Chebyshev filtre için band geçiren OTA-C quad alt devresi



Şekil 6.9.c Eliptik filtre için alçak geçiren/band geçiren
quad alt devresi

H_o sabit kazanç bloğu ve T_a bloğu hariç verilen tüm OTA-C devre şemaları Tablo 5.4'deki alt devrelerin birleştirilmesiyle elde edilmiştir.

Herbir kritik düğümün dinamik değişimine bir ölçü olarak aşağıdaki tanımlar yapılabilir:

$$r_{ji} = \frac{C_{jimax}}{C_{jimin}} \quad (6.3.a)$$

$$r_{2i} = \frac{C_{2imax}}{C_{2imin}} \quad (6.3.b)$$

Sıfır-kutup düzenlemesi ve alt devrelerin farklı biçimlerde sıralanmasıyla, (6.3) denklemlerindeki r_{ji} ve r_{2i} değerleri için farklı gruplar oluşturulabilir. İyi bir dinamik değişim için, sıfır-kutup düzenlemesi ve alt devrelerin sıralanması işlemi, $\max[r_{ji}]$ ve $\max[r_{2i}]$ minimum olacak şekilde yapılmalıdır [18]. Bu durumda, herbir grup için,

$$\min(r_{\max}) = \min(\max(r_{ji}), \max(r_{2i})) \quad (6.4)$$

değeri hesaplanmalıdır. Bulunabilecek tüm $\min[r_{\max}]$ değerleri için, sıfır-kutup düzenlemesi ve biquadratik blokların sıralaması yapılırsa, her bir kritik düğüm için eşit dinamik davranış değişimi elde edilmiş olur. Böylece, girişe uygulanan en büyük (en küçük) gerilim seviyelerine karşılık, herhangi bir kritik düğüm geriliminin diğerlerinden önce doymaya (kesime) gitmemesi garantilenmiş olur. Sıfır-kutup düzenlemesi devrenin duyarlılığına etki ettiği halde quad bloklarının ve bu bloklara ilişkin biquad alt devrelerinin sıralanmaları etki etmez. Quad/kaskad blokları ve bu bloklara ilişkin biquadratik blokların dinamik değişimi artıracak şekilde sıralanması aşağıdaki kurala göre yapılabilir:

(i) Olabildiğince çok sayıda alçak geçiren türden quad/kaskad alt devreleri, filtrenin ilk alt devreleri olarak seçilmelidir. Alçak geçiren türden quad/kaskad alt devresi yoksa, band geçiren alt devreler kullanılarak filtrenin ilk alt devreleri gerçekleştirilmelidir. Bu sıralama, filtre girişindeki yüksek frekanslı gürültülerin etkisini yok edecektir. Alçak geçiren Butterworth ve Chebyshev türünden filtreler için tüm alt devreler AG yapısındadır. Dolayısıyla bu koşul sağlanır. Alçak geçiren Eliptik türden filtreler için ise T_a bloğu hariç tüm alt devreler Notch yapısında olduğundan bu koşul eğer T_a bloğu varsa sağlanır, T_a bloğu yok ise sağlanmaz.

Band geçiren Butterworth ve Chebyshev türünden filtreler için tüm alt devreler BG türündendir. Bu koşul yine sağlanır. Band geçiren Eliptik türden filtreler için T_b bloğu hariç tüm alt bloklar Notch türündendir. T_b bloğu ise BG türündendir. Bu durumda T_b bloğu var ise bu koşul sağlanır.

(ii) Filtrenin son blokları olarak olabildiği kadar çok sayıda yüksek geçiren quad/kaskad alt devreleri (yoksa, band geçiren devreler) kullanılmalıdır. Bu sıralama sayesinde önceki devrelerden aktarılan D.A. değerleri yok edilir ve kullanılan OTA elemanları içinde üretilen alçak frekanslı gürültüler ortadan kaldırılmış olur. Bu koşul sadece band geçiren Butterworth ve Chebyshev türden filtreler için sağlanmaktadır.

6.2 Kazanç Katsayılarının Belirlenmesi ve Biquadratik Alt Devrelerin Gerçeklenmesi

Kazanç katsayıları, devrelerin tasarımında kullanılan aktif elemanların (OTA'ların) teknolojik sınır değerlerine göre, şöyle seçilebilir:

(i) Filtre girişine uygulanan gerilimin genliği büyükse, herhangi bir biquadratik blok çıkışının doyma bölgesine girmesi söz konusu olabilir. Kazanç katsayıları hiçbir biquadratik blok diğerinden önce doyma bölgesine girmeyecek şekilde seçilmelidir.

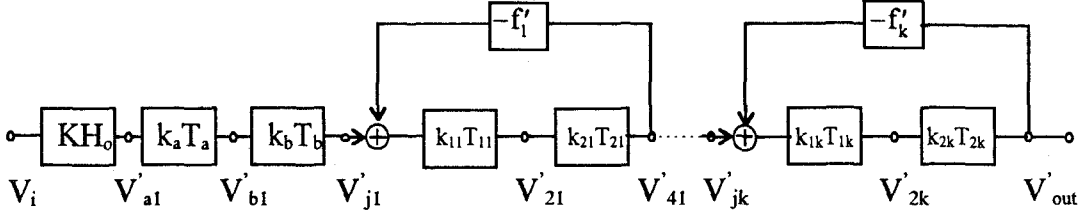
(ii) Filtrenin girişindeki işaret genliği çok küçük değerde ise, örneğin tıbbi ve sismik cihazlar v.b., işaret değişimi, sistemin gürültüsü tarafından bastırılabilir. Bunu önlemek için kazanç katsayılarının tüm biquadratik blokların çıkışlarının, ilgilenilen frekans bandı içinde belli bir gerilim değerinin altına düşmemesini sağlayacak şekilde belirlenmesi gerekir.

(iii) Yukarıda sözü edilen her iki durum birlikte ele alınarak kazanç katsayıları belirlenebilir.

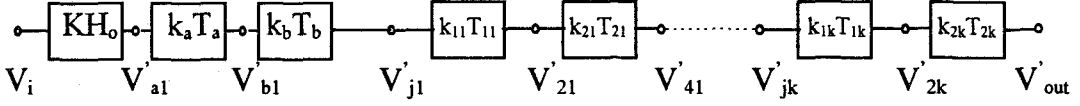
Tezde, tasarım sadece (i) şıkında belirtilen durum yani doyma durumu gözönüne alınarak yapılacaktır.

6.2.1 Doyma durumuna göre kazanç katsayılarının bulunması

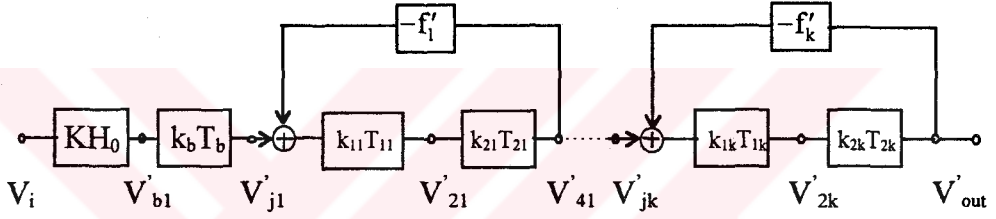
Filtrenin girişine uygulanan işaretin genliği büyük ise, bazı biquadların çıkışlarının doymaya gitmesi filtre cevabını etkileyecektir. O halde, maksimum izin verilen filtre giriş gerilimi, kullanılan OTA'ların maksimum çıkış işaretiyle sınırlandırılmaktadır. Sonuçta, biquadların kazanç katsayıları, herhangi bir biquad çıkışının diğerlerinden önce doymaya gitmesini önleyecek şekilde seçilmelidir [3]. Şekil 6.1, Şekil 6.2, Şekil 6.3 ve Şekil 6.4'teki yüksek dereceli filtrelerle ilişkin kazanç değerleri atanmış eşdeğer devreler sırasıyla Şekil 6.10, Şekil 6.11, Şekil 6.12 ve Şekil 6.13'de görülmektedir.



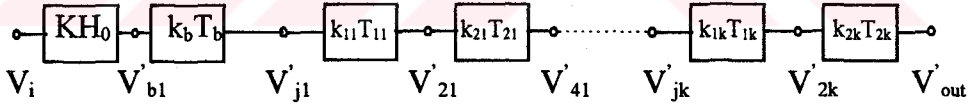
Şekil 6.10 Kazanç değerleri atanmış alçak geçiren quad filtresi



Şekil 6.11 Kazanç değerleri atanmış alçak geçiren kaskad filtresi



Şekil 6.12 Kazanç değerleri atanmış band geçiren quad filtresi



Şekil 6.13 Kazanç değerleri atanmış band geçiren kaskad filtresi

Filtrelerin girişine darbe (impuls) biçiminde gerilim uygulandığı varsayılarak her aynı türden devre çifti için (1.çift:Şekil 6.1 ve Şekil 6.10, 2.çift: Şekil 6.2 ve Şekil 6.11, 3.çift:Şekil 6.3 ve Şekil 6.12, 4.çift:Şekil 6.4 ve Şekil 6.13) kritik düğümlere ilişkin gerilim değerleri hesaplanabilir. Eşdeğer devreler için,

$$\max_{0 < \omega < \infty} |V'_{b1}| = \max_{0 < \omega < \infty} |V'_{ji}| = \max_{0 < \omega < \infty} |V'_{2i}| = \max_{0 < \omega < \infty} |V'_{4k}| = \max_{0 < \omega < \infty} |V'_{out}| \quad (6.5)$$

bağıntısının sağlanması gerekir. Devre çiftleri ve (6.5) denkleminde yararlanarak biquadratik alt devrelere ilişkin kazanç katsayıları,

$$k_{li} k_{2i} = \frac{\max_{0 < w < \infty} |V_{ji}|}{\max_{0 < w < \infty} |V_{4i}|} \quad (6.6.a)$$

$$k_{li} = \frac{\max_{0 < w < \infty} |V_{ji}|}{\max_{0 < w < \infty} |V_{2i}|} \quad (6.6.b)$$

$$k_{2i} = \frac{\max_{0 < w < \infty} |V_{2i}|}{\max_{0 < w < \infty} |V_{4i}|} \quad (6.6.c)$$

$$k_a = \frac{\max_{0 < w < \infty} |V_{a1}|}{\max_{0 < w < \infty} |V_{b1}|} \quad (6.6.d)$$

$$k_b = \frac{\max_{0 < w < \infty} |V_{b1}|}{\max_{0 < w < \infty} |V_{j1}|}, \quad i = 1, 2, \dots, k \quad (6.6.e)$$

biçiminde hesaplanabilir. Filtrenin frekans cevabının aynı kalmasını sağlamak amacıyla, filtre girişine uygulanması gereken giriş kazanç katsayısı ve yeni geribesleme katsayıları (6.6) denklemleri kullanılarak,

$$K = \frac{1}{k_a k_b} \prod_{i=1}^k (k_{li} k_{2i})^{-1} \quad (6.7.a)$$

$$f'_i = \frac{f_i}{k_{li} k_{2i}}, \quad i = 1, 2, \dots, k \quad (6.7.b)$$

biçiminde belirlenebilir.

6.2.2 Biquadratik alt devrelerin OTA-C elemanlarıyla gerçekleştirilmesi

Biquadratik alt devrelerin gerçekleştirilmesinde Şekil 6.6'daki RC devresi ve Şekil 6.7, Şekil 6.8, Şekil 6.9'daki OTA-C devreleri kullanılacaktır. Bu devrelerdeki eleman değerleri ve gerçekleştirilecek filtreye ait biquadratik transfer fonksiyonlarının katsayıları karşılaştırılırsa, devrelere ilişkin OTA iletkenlikleri ve kapasite elemanlarının değerleri kolayca bulunabilir. Daha önce Tablo 5.4'te çeşitli türden biquadratik alt devreler ve bu devrelerde kullanılacak olan elemanların değerleri için bağıntılar verilmişti. Dolayısıyla Tablo 5.4'ten yararlanarak Butterworth, Chebyshev ve Eliptik filtreler için tipik biquadratik OTA-C alt devreleri ve bunlara ait eleman değerleri bulunabilir.

Tablo 5.4'deki 1., 2. ve 4. devrelerde, devredeki parametre sayısının tasarım eşitliklerinden fazla olduğu görülmektedir. O halde, bu bağımsız değişkenlere uygun değerler verilerek, ya en az duyarlıklı ya da iyi dinamik davranışa sahip devreler elde edilebilir.

Birinci dereceden olan T_a bloğunu gerçekleştirmekte kullanılacak olan RC devresinin eleman değerleri (2.23.e.) bağıntısıyla verilmiş olan $T_a(s)$ fonksiyonundan yararlanarak bulunabilir. Tablo 6.1'de Butterworth, Chebyshev ve Eliptik türden filtreler için T_a bloğu eleman değerleri ve biquadratik alt devreler için OTA iletkenlikleri ve kapasite elemanlarının değerleri dinamik değişimi artıracak şekilde verilmektedir.

Tablo 6.1 Butterworth, Chebyshev ve Eliptik türden quad/kaskad filtrelerine ilişkin T_a bloğunun ve temel biquad devrelerinin eleman değerleri

Devre	Devre Parametreleri	
	Bağımsız değişkenler	Bağımlı değişkenler
Şekil 6.6.a'daki RC devresi (T_a bloğu)	—	$R = 1/\sigma_0$ $C = 1/w_p$
Tablo 5.4'te 1.devre	$g_{m2} = 100 \mu s$ $g_{m3} = 100 \mu s$	$C_1 = (g_{m3} b_1)/a_0$ $g_{m1} = (C_1 b_0)/b_1$ $C_2 = g_{m2}/b_1$
Tablo 5.4'te 2.devre	$g_{m1} = 100 \mu s$ $g_{m3} = 100 \mu s$	$C_2 = g_{m3}/a_1$ $g_{m2} = C_2 b_1$ $C_1 = (g_{m1} b_1)/b_0$
Tablo 5.4'te 4.devre	$g_{m2} = 100 \mu s$ $g_{m4} = 100 \mu s$ $g_{m5} = 100 \mu s$	$C_1 = (g_{m5} b_1)/a_0$ $g_{m1} = (C_1 b_0)/b_1$ $C_2 = (g_{m2} a_2)/b_1$ $g_{m3} = a_2 g_{m4}$

BÖLÜM 7

OTA-C QUAD VE KASKAD FİLTRELERİNİN MAKSİMUM GİRİŞ GERİLİMİNİN HESAPLANMASI

Tasarlanan filtrede clipping ve slew-rate problemleri ile karşılaşmamak için filtre girişine uygulanabilecek maksimum gerilim değerinin belirlenmesi gerekmektedir. Bu bölümde, tasarlanan OTA-C filtresinin girişine uygulanabilecek maksimum gerilim değerinin hesaplanması için izlenecek yol anlatılacaktır.

7.1 Genel Kavramlar

Pratikte, OTA'lar çıkış işaretlerinin belirli bir değeri aşması halinde doğrusal davranış gösteremezler. Çıkış akımı doyma değerini aşarsa, slew-rate problemi ortaya çıkar. Devredeki herhangi bir OTA'nın çıkış gerilimi doyma değerini geçtiği zaman, filtre çıkışında kırılmış bir işaret elde edilir. Maksimum filtre giriş gerilimi, devredeki herhangi bir OTA'nın çıkış gerilimi ve akımı doyma değerlerini aşmayacak şekilde belirlenmelidir [11],

$$|V_k| \leq V_{ks} \quad , k = 1, 2, \dots, n \quad (7.1.a)$$

$$|I_k| \leq I_{ks} \quad , k = 1, 2, \dots, n \quad (7.1.b)$$

Belirli bir frekans aralığında, $w \in [w_1, w_2]$, (7.1.a) ve (7.1.b) ifadelerinin her ikisinin birlikte sağlanması gerekir. Bu ifadelerde n, kullanılan OTA sayısı olmak üzere $V_k = V_k(jw)$ ve $I_k = I_k(jw)$ sırasıyla k numaralı OTA'ya ait fazör gerilim ve akımlarıdır. Devrede kullanılan OTA'ların benzer olması halinde tüm doyma gerilimi ve tüm akım değerleri birbirine eşit kabul edilebilir [11]. Bu durumda,

$$V_{ls} = \dots = V_{ns} = V_s \quad (7.2.a)$$

$$I_{ls} = \dots = I_{ns} = I_s \quad (7.2.b)$$

ifadeleri yazılabilir. (7.1.a) ve (7.1.b) bağıntıları giriş geriliminin genliği cinsinden

$$|V_i| |H_k| \leq V_{ks} = V_s, \quad k = 1, 2, \dots, n \quad (7.3.a)$$

$$|V_i| |Y_k| \leq I_{ks} = I_s, \quad k = 1, 2, \dots, n \quad (7.3.b)$$

biçiminde ifade edilebilir. Bu bağıntılarda $|V_i|$, filtre giriş geriliminin genliği olmak üzere, $H_k = H_k(j\omega)$ ve $Y_k = Y_k(j\omega)$ k numaralı OTA'nın gerilim transfer ve transfer admitans fonksiyonlarıdır [11]. H_k ve Y_k için

$$H_k = V_k / V_i, \quad k = 1, 2, \dots, n \quad (7.4.a)$$

$$Y_k = I_k / V_i, \quad k = 1, 2, \dots, n \quad (7.4.b)$$

bağıntıları yazılabilir. (7.3) ifadeleri yardımıyla,

$$|V_i| \leq V_s / |H_k|, \quad k = 1, 2, \dots, n \quad (7.5.a)$$

$$|V_i| \leq I_s / |Y_k|, \quad k = 1, 2, \dots, n \quad (7.5.b)$$

eşitsizlikleri elde edilir. (7.5) eşitsizliklerinin ortak çözümü ile filtre girişine uygulanabilecek maksimum giriş gerilimi genliği,

$$|V_i|_{\max} = \min \left[V_s / |H_k(jw)|_{\max} , I_s / |Y_k(jw)|_{\max} , k = 1, 2, \dots, n \right] \quad (7.6)$$

biçiminde ifade edilebilir. Burada $|H_k|_{\max}$ ve $|Y_k|_{\max}$ ilgilenilen $w \in [w_1, w_2]$ frekans aralığında $|H_k|$ ve $|Y_k|$ 'nin maksimum değerleridir [11].

7.2 Yüksek Dereceden Butterworth, Chebyshev ve Eliptik Quad/Kaskad Filtrelerinin Maksimum Giriş Geriliminin Hesaplanması

(7.6) denkleminden anlaşılaçağı gibi maksimum giriş gerilimini hesaplayabilmek için daha önce Şekil 6.9'da verilen Butterworth, Chebyshev ve Eliptik türden quad/kaskad OTA-C devrelerinde bulunan bütün OTA'lara ait gerilim transfer fonksiyonu (H) ve transfer admitans (Y) fonksiyonlarının hesaplanması gerekmektedir.

Tanım:

$|V_i|_{\max}$ değerinin bulunabilmesi için filtredeki tüm OTA'lara ait $V_s/|H_k|_{\max}$ ve $I_s/|Y_k|_{\max}$ değerlerinin hesaplanıp Dominant OTA ve Dominant Terimin tespit edilmesi gerekir.

$$V_s / |H_a|_{\max} < \left[V_s / |H_k|_{\max} , I_s / |Y_k|_{\max} \right] , k \neq a$$

$$I_s / |Y_a|_{\max} < \left[V_s / |H_k|_{\max} , I_s / |Y_k|_{\max} \right] , k \neq a$$

(7.7)

ise $k = a$ numaralı OTA'ya "Dominant OTA" denir. $|H_a|_{\max}$ ya da $|Y_a|_{\max}$ 'a da "Dominant Terim" denir. Filtrenin maksimum giriş geriliminin genliğini dominant OTA'ya ait dominant terim belirlemektedir. O halde maksimum giriş geriliminin genliği,

$$|V_i|_{\max} = \begin{cases} V_s / |H_a|_{\max} & , \text{ dominant terim H ise} \\ I_s / |Y_a|_{\max} & , \text{ dominant terim Y ise} \end{cases} \quad (7.8)$$

biçiminde de ifade edilebilir.

7.2.1 Alçak geçiren ve band geçiren Butterworth ve Chebyshev filtrelerin maksimum giriş geriliminin bulunması

Şekil 6.9.a’da alçak geçiren Butterworth ve Chebyshev filtreler için verilmiş olan OTA-C quad devresinde bulunan tüm OTA’lara ait gerilim transfer ve transfer admitans fonksiyonları aşağıda verilen algoritmayla hesaplanabilir [19].

1) i numaralı quadın giriş gerilimi V_j , $i-1$ numaralı quadın çıkış gerilimi V_4 ’e eşit olduğu için, “ $H_j(i)$ ” değeri bilinmektedir.

2) $w \in [w_1, w_2]$ frekans bandı için,

$$H_1(i) = \frac{1}{(1 + f_i H_{41}(i))} H_j(i) \quad (7.9.a)$$

$$H_2(i) = H_{21}(i) H_2(i) \quad (7.9.b)$$

$$H_3(i) = -\frac{b_{10}(i)}{b_{11}(i)s} H_2(i) \quad (7.9.c)$$

$$H_4(i) = H_{42}(i) H_2(i) \quad (7.9.d)$$

$$H_5(i) = -\frac{b_{20}(i)}{b_{21}(i)s} H_4(i) \quad (7.9.e)$$

gerilim transfer fonksiyonları hesaplanır.

3) Yine belirlenen frekans bandı için,

$$Y_1(i) = -g_1(i) H_2(i) \quad (7.10.a)$$

$$Y_2(i) = g_2(i) (H_3(i) - H_2(i)) \quad (7.10.b)$$

$$Y_3(i) = g_3(i) H_1(i) \quad (7.10.c)$$

$$Y_4(i) = -g_4(i) H_4(i) \quad (7.10.d)$$

$$Y_5(i) = g_5(i) (H_5(i) - H_4(i)) \quad (7.10.e)$$

$$Y_6(i) = g_6(i) H_2(i) \quad (7.10.f)$$

$$Y_x(i) = g_x(i) H_j(i) \quad (7.10.g)$$

$$Y_y(i) = -g_y(i) H_1(i) \quad (7.10.h)$$

$$Y_z(i) = -g_z(i) H_4(i) \quad (7.10.i)$$

transfer admitans fonksiyonları hesaplanır. Burada,

$H_p(i)$: i numaralı quad devresinin p numaralı düğümü ile tüm filtre girişi arasındaki gerilim transfer fonksiyonudur ve $V_p(i)/V_{in}$ olarak tanımlanır.

$V_p(i)$: i numaralı quad devresinin p numaralı düğüm gerilimidir.

V_{in} : Tüm filtre giriş gerilimini ifade eder.

$H_{21}(i)$: i numaralı quadın birinci biquadratik transfer fonksiyonudur ve,

$$H_{21}(i) = \frac{a_{21}(i)s^2 + a_{11}(i)s + a_{10}(i)}{s^2 + b_{11}(i)s + b_{10}(i)}$$

şeklinde tanımlanır.

$H_{42}(i)$: i numaralı quadın ikinci biquadratik transfer fonksiyonudur ve,

$$H_{42}(i) = \frac{a_{22}(i)s^2 + a_{21}(i)s + a_{20}(i)}{s^2 + b_{21}(i)s + b_{20}(i)}$$

biçiminde ifade edilir.

$H_{41}(i)$: i numaralı quad devresinin 4 numaralı düğümünün 1 numaralı düğümüne göre gerilim transfer fonksiyonudur.

$Y_m(i)$: i numaralı quad devresine ait m numaralı OTA elemanının filtre girişine göre transfer admitans fonksiyonudur ve $I_m(i)/V_{in}$ olarak tanımlanır.

$I_m(i)$: i numaralı quad devresine ait m numaralı OTA'nın çıkış akımıdır.

$g_m(i)$: i numaralı quad devresine ait m numaralı OTA'nın çıkış iletkenliği

$i = 1$ 'den başlanmak üzere sonuncu quada kadar verilen adımlar tekrarlanır. Tüm OTA'lara ilişkin H ve Y 'lerin tasarım bandı için maksimum değerleri, $|H_k|_{max}$ ve $|Y_k|_{max}$, hesaplanır. Dominant OTA ve dominant terim (7.7) eşitsizliğinden yararlanarak tespit edilir ve $|V_i|_{max}$ değeri (7.8) ifadesi kullanılarak hesaplanır.

Butterworth ve Chebyshev yaklaşıklık tipi kullanılarak gerçekleştirilen alçak geçiren OTA-C kaskad filtrelerine uygulanabilecek maksimum giriş gerilimini hesaplamak için izlenecek yol aynıdır. Sadece z OTA'sı kaskad devrelerde olmadığı için (7.10.i) formülüne gerek yoktur ve (7.9.a) ifadesi, kaskad devrelerde $f_i = 0$ olduğu için,

$$H_1(i) = H_j(i) \quad (7.11)$$

şeklini alınır. Geriye kalan ifadeler tamamıyla aynıdır. Anlatılan farklı durumlar, Butterworth ve Chebyshev yaklaşıklık tipi kullanılarak gerçekleştirilen band geçiren OTA-C kaskad alt devresi için de söz konusudur.

Şekil 6.9.b’de verilmiş olan Butterworth ve Chebyshev türünden band geçiren quad alt devresi, Şekil 6.9.a’da verilmiş olan alçak geçiren quad alt devresinden farklı olmasına rağmen verilen algoritma tamamıyla aynı olmaktadır.

7.2.2 Alçak geçiren ve band geçiren Eliptik filtrelerin maksimum giriş geriliminin hesaplanması

Bir önceki alt bölümde anlatılan algoritma Şekil 6.9.e’deki eliptik filtre için aşağıdaki gibi olacaktır.

1) i numaralı quad için, “ $H_j(i)$ ” değeri bilinmektedir.

2) Aşağıdaki gerilim transfer fonksiyonları belirlenen frekans bandı için bulunur.

$$H_1(i) = \frac{1}{(1 + f_i H_{41}(i))} H_j(i) \quad (7.12.a)$$

$$H_2(i) = H_{21}(i) H_1(i) \quad (7.12.b)$$

$$H_3(i) = \frac{1}{b_{11}(i)s} (a_{10}(i) H_1(i) - b_{10}(i) H_2(i)) \quad (7.12.c)$$

$$H_4(i) = H_{42}(i) H_2(i) \quad (7.12.d)$$

$$H_5(i) = \frac{b_{11}(i)}{a_{12}(i)s} (H_2(i) - H_3(i)) \quad (7.12.e)$$

$$H_6(i) = \frac{1}{b_{21}(i)s} (a_{20}(i) H_2(i) - b_{20}(i) H_4(i)) \quad (7.12.f)$$

$$H_7(i) = \frac{b_{21}(i)}{a_{22}(i)s} (H_4(i) - H_6(i)) \quad (7.12.g)$$

3) Aşağıda verilen transfer admitans fonksiyonları belirlenen frekans bandı için bulunur.

$$Y_1(i) = -g_1(i) H_2(i) \quad (7.13.a)$$

$$Y_2(i) = g_2(i) (H_2(i) - H_3(i)) \quad (7.13.b)$$

$$Y_3(i) = g_3(i) (H_1(i) - H_5(i)) \quad (7.13.c)$$

$$Y_4(i) = -g_4(i) H_2(i) \quad (7.13.d)$$

$$Y_5(i) = g_5(i) H_1(i) \quad (7.13.e)$$

$$Y_6(i) = -g_6(i) H_4(i) \quad (7.13.f)$$

$$Y_7(i) = g_7(i) (H_4(i) - H_6(i)) \quad (7.13.g)$$

$$Y_8(i) = g_8(i) (H_2(i) - H_7(i)) \quad (7.13.h)$$

$$Y_9(i) = -g_9(i) H_4(i) \quad (7.13.i)$$

$$Y_{10}(i) = g_{10}(i) H_2(i) \quad (7.13.j)$$

$$Y_x(i) = g_x(i) H_j(i) \quad (7.13.k)$$

$$Y_y(i) = -g_y(i) H_1(i) \quad (7.13.l)$$

$$Y_z(i) = -g_z(i) H_4(i) \quad (7.13.m)$$

Alçak geçiren ve band geçiren eliptik türden quad alt devreleri için aynı devre (Şekil 6.9.c) kullanıldığı için verilen algoritma alçak geçiren ve band geçiren filtreler için ortak algoritma olmaktadır.

Alçak geçiren/band geçiren eliptik türden kaskad alt devresi için maksimum giriş gerilimini hesaplamada kullanılacak olan algoritma da hemen hemen aynıdır. Sadece, (7.13.m) ifadesine gerek yoktur ve (7.12.a) ifadesi, kaskad filtreler için $f_i = 0$ olduğundan dolayı,

$$H_1(i) = H_j(i) \quad (7.14)$$

şeklini alır.

SONUÇLAR VE ÖNERİLER

Bu tezde, alçak geçiren ve band geçiren filtrelerin tasarımı ekonomik, iyi dinamik davranış gösteren ve az duyarlıklı bir yöntem olan quad yöntemi kullanılarak OTA elemanlarıyla gerçekleştirilmiştir. Ayrıca kaskad yöntemi için gerekli olan tasarım eşitlikleri de elde edilmiştir.

Tasarım denklemleri elde edilirken Kaynak [3]'de verilmiş olan genel quad tasarımından yola çıkılmıştır. Alçak geçiren filtreler için quad tasarım eşitliklerinin elde edilmesinde yapılan işlemler band geçiren filtreler için yapılan işlemlerden oldukça karmaşıktır. Böyle olmasına rağmen alçak geçiren filtrelerin gerçekleştirilmesinde kullanılan OTA-C devreleri band geçiren filtrelerde kullanılanlar kadar basit bir yapıda olmaktadır.

Tasarımda işlemsel kuvvetlendirici yerine OTA kullanılarak birçok avantaj sağlanmıştır. Bunlardan en önemlisi OTA'ların yüksek frekans davranışlarının işlemsel kuvvetlendiricilerden çok daha iyi olmasından dolayı daha yüksek frekanslarda çalışabilen filtre devreleri elde edebilmek olmuştur. Ayrıca OTA-C devreleriyle elde edilen devreler işlemsel kuvvetlendiricilerle elde edilenlerden daha az sayıda elemana sahiptir.

Tezde, alçak geçiren ve band geçiren OTA-C filtrelerinin tasarımında Butterworth, Chebyshev ve Eliptik yaklaşıklıklar kullanılmıştır. Alçak geçiren filtreler için f_a , f_p , A_p ve A_a olmak üzere sadece dört özelliğin, band geçiren filtreler için ise f_{a1} , f_{p1} , f_{p2} , f_{a2} , A_p ve A_a olmak üzere sadece altı özelliğin bilinmesiyle yüksek dereceden filtreler tasarlanabilmektedir.

OTA ve kapasite elemanlarıyla gerçekleştirilen filtreler, dışarıdan elektriksel olarak kontrol edilebilir OTA iletkenlik parametresi nedeniyle işlemsel kuvvetlendiricili filtrelerle oranla daha kullanışlı devrelerdir ve kolayca ayarlanabilmektedirler.

Tasarlanan filtreler sadece OTA ve topraklı kapasite elemanları içerdiğinden yani, direnç elemanı içermediğinden tümleştirme sonucunda elde edilen silikon kırılgan alanı da küçük olacaktır. Bu nedenle elde edilen filtreler tümleşik devre tekniğine uygundur.

Tezde, OTA-C filtreleri optimuma yakın dinamik davranış elde edilecek şekilde tasarlanmıştır. Ayrıca filtrelerin doğrusal çalışabilmesi için maksimum filtre giriş geriliminin genliği hesaplanmıştır.

Ek A'da alçak geçiren ve band geçiren filtre tasarım örnekleri ve bu örneklere ait frekans karakteristikleri verilmiştir. Ek B'de verilen bilgisayar programı yardımıyla Butterworth, Chebyshev, Eliptik yaklaşıklık tiplerinden biri seçilerek, alçak geçiren filtreler için girilen dört filtre özelliği, band geçiren filtreler için ise girilen altı filtre özelliği ile quad ve kaskad yöntemi kullanılarak alçak geçiren ve band geçiren OTA-C filtreleri tasarlanabilmektedir.

KAYNAKLAR

- [1] **GHAUSI, M.S., R.SCHAUMAN, K.R.LAKER**, "Design of Analog Filters: Passive, Active-RC, Switched Capacitor", Prentice Hall, Inc., 1990
- [2] **TAN, M.A. and SCHAUMANN, R.**, "Generation of Transconductance-Grounded Capacitor Filters by Signal-Flow-Graph Simulation of LC-Ladders", IEEE Trans. Circuits Syst., vol.CAS-36, pp.299-307, Feb.1989
- [3] **GÖNÜLEREN,A.N.**, "Simetrik Olmayan Aktif RC Filtrelerinin Çok Çevrimli Geribesleme Yöntemiyle Tasarımında Az Duyarlıklı Uygulaması Basit Yeni Bir Yöntem:QUAD (Tetra Quad) ve Çok Çevrimli Geribesleme Yöntemlerinin En Genel Biçimde Analiz ve Tasarımı", Doçentlik Tezi, İstanbul Teknik Üniversitesi, 1981
- [4] **NATARAJAN, S.**, "Theory And Design of Linear Active Networks", McGraw-Hill, Singapore, 1989
- [5] **R.G.SPARKES and A.S.SEDRA**, "Programmable Active Filters", IEE J. Solid-State Circuits, vol. SC-8, pp.93-95, Feb.1973
- [6] **S.FRANCO**, "Use Transconductance Amplifier to Make Programmable Active Filters", Electronic Design, Vol.24.,pp.98-101, Sept 1976
- [7] **GEIGER.R.L. and SANCHEZ-SINENCIO.E.**, "Active Filter Design Using Transconductance Amplifiers :A Tutorial", IEEE Circuits and Devices Magazine, March 1985
- [8] **HUELSMAN LAWRENCE P.**, "Active and Passive Analog Filter Design", 1993
- [9] **CHOUDHURY.ROY**, "Linear Integrated Circuits" John Wiley & Sons , 1991

- [10] **PARKS, T.W., C.S. BURRUS**, "Digital Filter Design", John Wiley c Sons, Inc., 1987.
- [11] **ACAR.C, ANDAY.F, KUNTMAN.H**, " On the realization of OTA-C Filters" International Journal Of Circuit Theory And Application Vol 21 Iss 4 pp 331-341 1993
- [12] **NEDUUNGADI. A. and VISWANATHAN.T.R.**, "Design Of Linear CMOS Transconductance Elements" IEEE Transactions on Circuits and Systems" Vol.CAS-31, NO.10 ,October 1984
- [13] **KUNTMAN.H.**, "İleri Lineer Tümdevre Tasarımı, Analog Devreler", Şubat 1984
- [14] **R.L.GEIGER and J.FERRELL**, "Voltage Controlled Filter Design Using Operational Transconductance Amplifiers", Proc. IEEE/ISCAS, pp.594-597, May 1983
- [15] **H.S.MALVAR**, "Electronically Controlled Active-C Filters and Equalizers with Operational Transconductance Amplifiers", IEEE Transactions on Circuits and Systems, vol.CAS-31, No.7,July 1984
- [16] **E.SANCHEZ-SINENCIO, R.L.GEIGER and H.NEVAREZ-LOZANO**, "Generation of Continuous-Time Two Integrator Loop OTA Filter Structures", IEEE Trans. Circuits and Systems, CAS-35, 936-946(1988)
- [17] **R.NAWROCKI, U.KLEIN**, "New OTA-Capacitor Realization A Universal Biquad", Electronics Letters, 2nd January 1986, vol.22, No.1
- [18] **A.N.GÖNÜLEREN**, "Sensitivity And Dynamic Range of Quad Filters And Some Comparative Results", Circuit Theory And Applications, vol.10,175-200 (1982)

- [19] **KÖPRÜ. R.**, “Yüksek Dereceden Aktif OTA-C Filtrelerinin Quad Yöntemi Kullanarak Bilgisayar Yardımıyla Tasarımı” Y.Lisans Tezi İTÜ Fen Bilimleri Enstitüsü 1994



T.C. İTÜ FEN BİLİMLERİ ENSTİTÜSÜ
KÜTÜPHANE BİRLİĞİ
2024

EK A

TASARIM ÖRNEKLERİ

Örnek 1

$f_p = 1.1$ Mhz, $f_a = 1.5$ Mhz, $A_p < 0.5$ dB, $A_s > 25$ dB özelliklerine sahip Butterworth yaklaşıklı alçak geçiren filtre tasarlanmak isteniyor. Bu özellikleri sağlayan tasarım denklemlerini ve eleman değerlerini elde edelim.

Verilen özellikleri sağlayan prototip alçak geçiren fonksiyonuna ilişkin kutuplar ve kazanç sabiti,

$$p_1, p_{13} = -0.1306 \pm j 1.0763$$

$$p_2, p_{12} = -0.3844 \pm j 1.0138$$

$$p_3, p_{11} = -0.6159 \pm j 0.8923$$

$$p_4, p_{10} = -0.8115 \pm j 0.7190$$

$$p_5, p_9 = -0.9600 \pm j 0.5038$$

$$p_6, p_8 = -1.0527 \pm j 0.2594$$

$$h_0 = 2.8627$$

olarak elde edilir. Filtre 13.dereceden bir fonksiyona sahip olup, bir T_a bloğu ve üç quad bloğundan oluşmaktadır. Alt devrelere ilişkin kazanç katsayıları Tablo A1'de verilmektedir.

Tablo A1. Örnek 1'e ilişkin kazanç katsayıları

Katsayı	T _a Bloğu	1.Quad Blok	2.Quad Blok	3.Quad Blok
k _a	1.0842	—	—	—
k _{1i}	—	1.1799	1.1950	1.2087
k _{2i}	—	1.1713	1.1565	1.0279
K	1.1123			

Kazanç katsayıları atanmış alt devrelere ilişkin transfer fonksiyonları, geribesleme katsayıları ve iyilik faktörleri ise şöyledir;

$$T_a(s) = 7493947/(s + 7493938) \quad (A1.a)$$

$$T_{11} = 5.6365 \times 10^{13}/(s^2 + 1.3911 \times 10^7 s + 5.5953 \times 10^{13})$$

$$T_{21} = 5.5954 \times 10^{13}/(s^2 + 1.391173 \times 10^7 s + 5.59539 \times 10^{13})$$

$$Q_1 = 0.5376925 \quad f_1' = 7.294517 \times 10^{-3} \quad (A1.b)$$

$$T_{12} = 5.708849 \times 10^{13}/(s^2 + 9866336s + 5.524481 \times 10^{13})$$

$$T_{22} = 5.708849 \times 10^{13}/(s^2 + 9866336s + 5.524481 \times 10^{13})$$

$$Q_2 = 0.753338 \quad f_2' = 3.229566 \cdot 10^{-2} \quad (A1.c)$$

$$T_{13} = 5.774073 \times 10^{13}/(s^2 + 3560682s + 5.462068 \times 10^{13})$$

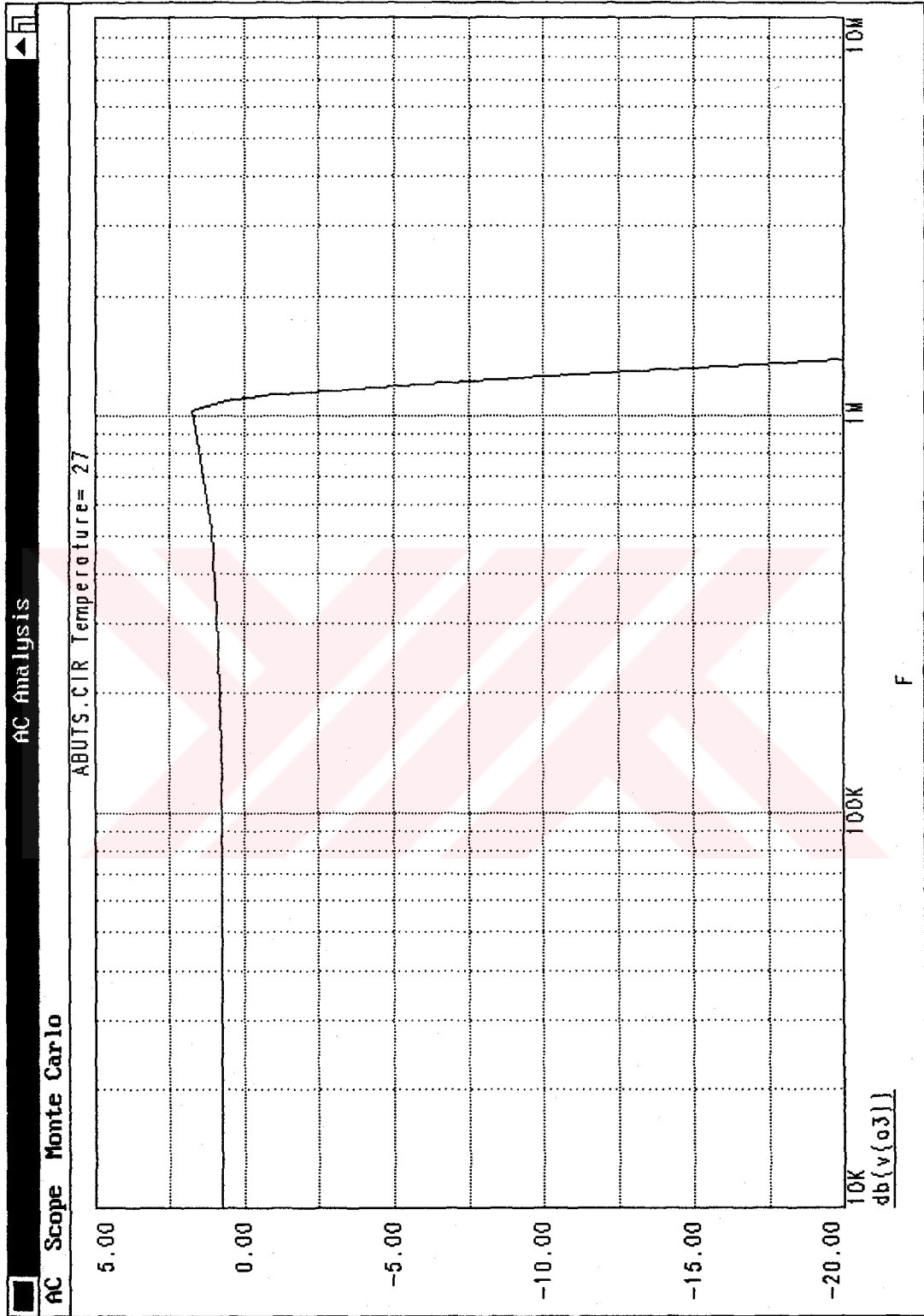
$$T_{23} = 4.910283 \times 10^{13}/(s^2 + 3560682s + 5.462068 \times 10^{13})$$

$$Q_3 = 2.075608 \quad f_3' = 0.0601101 \quad (A1.d)$$

Alçak geçiren Butterworth yaklaşıklı filtreye ilişkin eleman değerleri Tablo A2’de verilmektedir.

Tablo A2. Örnek 1’e ilişkin eleman değerleri

Elemanlar $g_i(\mu S), C_i(pF), R(\Omega)$	1.Quad	2.Quad	3.Quad
g_1	99.2703	96.7704	94.5964
g_2	100	100	100
g_3	100	100	100
g_4	99.2703	96.7704	94.5964
g_5	100	100	100
g_6	100	100	100
g_x	100	100	100
g_y	100	100	100
g_z	0.7294	3.2295	6.0110
C_1	24.6814	17.2825	6.1666
C_2	7.1881	10.1354	28.0845
C_3	24.8628	17.8593	7.2514
C_4	7.1881	10.1354	28.0845
g_a	111.2372		
g_b	100		
R	0.9222		
C	$1.446 \cdot 10^5$		



Şekil A1. Örnek 1'deki Alçak geçiren Butterworth yaklaşık
filtreye ilişkin Frekans Karakteristiği

Örnek 2

$f_p = 1.3$ Mhz, $f_a = 1.6$ Mhz, $A_p < 0.5$ dB, $A_s > 65$ dB özelliklerine sahip Chebyshev yaklaşıklı alçak geçiren filtre için tasarım denklemlerini ve eleman değerlerini elde edelim.

Bu özellikleri sağlayan prototip alçak geçiren fonksiyonuna ilişkin kutuplar ve kazanç sabiti,

$$p_1, p_{14} = -1.4226 \times 10^{-2} \pm j 1.0017$$

$$p_2, p_{13} = -4.1966 \times 10^{-2} \pm j 0.9514$$

$$p_3, p_{12} = -6.7601 \times 10^{-2} \pm j 0.8535$$

$$p_4, p_{11} = -8.9847 \times 10^{-2} \pm j 0.7127$$

$$p_5, p_{10} = -0.1075 \pm j 0.5363$$

$$h_0 = 3.4945 \times 10^{-4}$$

olarak elde edilir. Filtre 14.dereceden bir fonksiyona sahiptir. Tablo A3'de alt devrelere ilişkin kazanç katsayıları verilmektedir.

Tablo A3. Örnek 2'deki filtreye ilişkin kazanç katsayıları

Katsayı	T_b Bloğu	1.Quad Blok	2.Quad Blok	3.Quad Blok
k_b	0.0286	—	—	—
k_{1i}	—	0.1764	0.5998	0.9432
k_{2i}	—	0.2094	0.6163	0.6918
K	1.3663			

Kazanç değerleri atanmış alt devrelere ilişkin transfer fonksiyonları, geribesleme katsayıları ve iyilik faktörleri ise şöyledir;

$$T_b(s) = 1.9135 \times 10^{12} / (s^2 + 2062691s + 1.9135 \times 10^{12})$$

$$Q_b = 0.6706 \quad k_b = 2.8681 \times 10^{-2}$$

(A2.a)

$$T_{11}(s) = 1.1774 \times 10^{13} / (s^2 + 1858421s + 1.4153 \times 10^{13})$$

$$T_{21}(s) = 1.3971 \times 10^{13} / (s^2 + 1858421s + 1.4153 \times 10^{13})$$

$$Q_1 = 2.0243 \quad f_1' = -0.2038625$$

(A2.b)

$$T_{12}(s) = 4.0021 \times 10^{13} / (s^2 + 1286067s + 4.1656 \times 10^{13})$$

$$T_{22}(s) = 4.1124 \times 10^{13} / (s^2 + 1286067s + 4.1656 \times 10^{13})$$

$$Q_2 = 5.0185 \quad f_2' = -0.0309$$

(A2.c)

$$T_{13}(s) = 6.2929 \times 10^{13} / (s^2 + 458992s + 6.3712 \times 10^{13})$$

$$T_{23}(s) = 4.6161 \times 10^{13} / (s^2 + 458992s + 6.3712 \times 10^{13})$$

$$Q_3 = 17.3903 \quad f_3' = -2.4445 \times 10^{-3}$$

(A2.d)

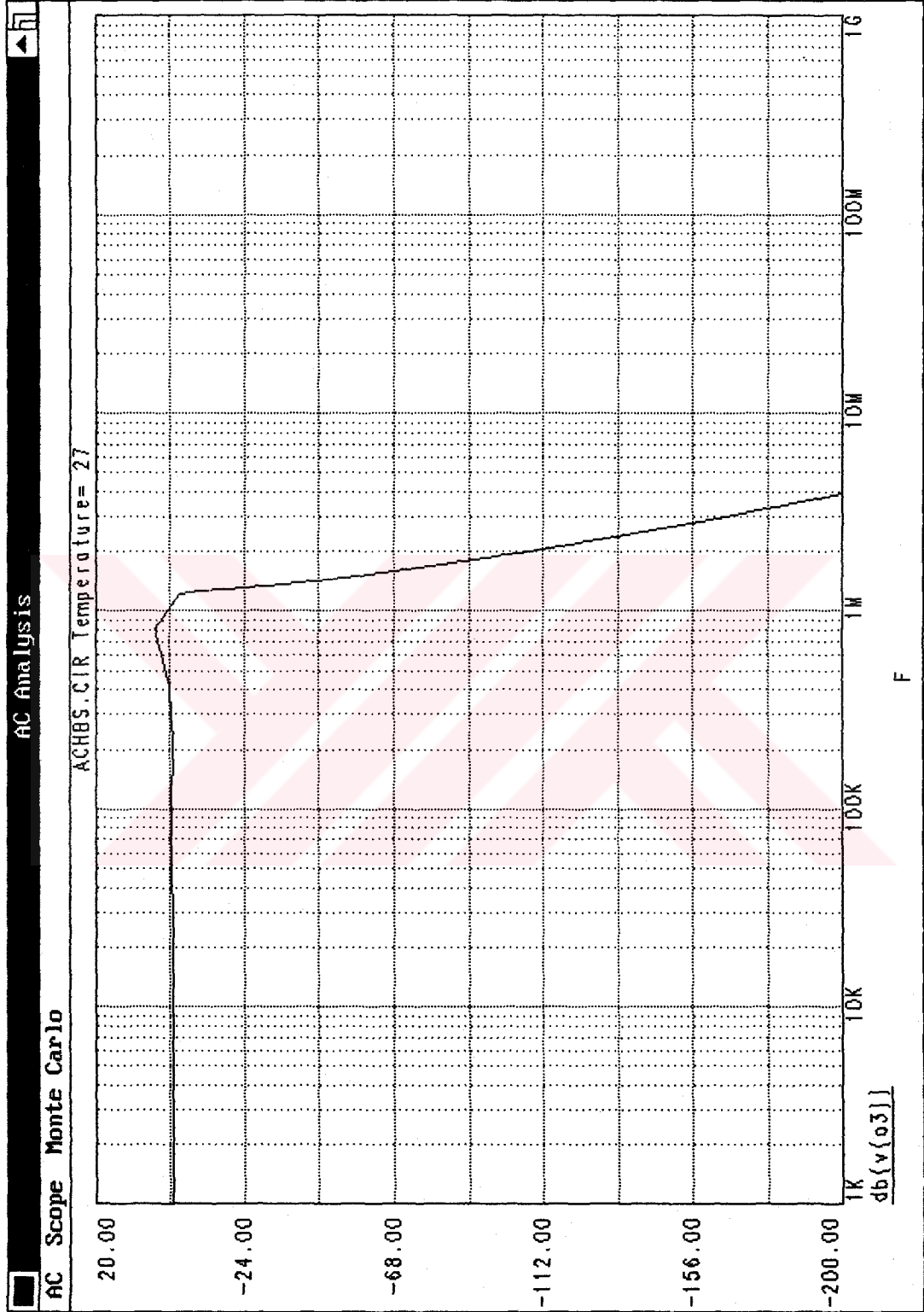
Alçak geçiren Chebyshev yaklaşıklı filtreye ilişkin eleman değerleri Tablo A4'de verilmektedir

Tablo A4.a. Örnek 2'ye ilişkin alt blok eleman değerleri

Elemanlar $g(\mu S), C(pF)$	Giriş Bloğu	T_b Bloğu
g_a	136.6302	—
g_b	100	—
g_{b1}	—	99.9988
g_{b2}	—	100
g_{b3}	—	100
C_{b1}	—	107.792
C_{b2}	—	48.4803

Tablo A4.b. Örnek 2'ye ilişkin quad blokları eleman değerleri

Elemanlar $g_i(\mu S), C_i(pF)$	1.Quad	2.Quad	3.Quad
g_1	120.2043	104.0872	101.2443
g_2	100	100	100
g_3	100	100	100
g_4	120.2043	104.0872	101.2443
g_5	100	100	100
g_6	100	100	100
g_x	100	100	100
g_y	100	100	100
g_z	20.3862	3.0986	0.2444
C_1	15.7831	3.2134	0.7293
C_2	53.8091	77.7564	217.8687
C_3	13.3011	3.1272	0.9943
C_4	53.8091	77.7564	217.8687



Şekil A2. Örnek 2'deki Alçak geçiren Chebyshev yaklaşık
filtreye ilişkin Frekans Karakteristiği

Örnek 3

$f_p = 1.15$ Mhz, $f_a = 1.25$ Mhz, $A_p < 0.5$ dB, $A_a > 50$ dB özelliklerine sahip Eliptik yaklaşık alçak geçiren filtre tasarlanmak isteniyor. Bu özellikleri sağlayan tasarım denklemlerini ve eleman değerlerini elde edelim.

Verilen özellikleri sağlayan prototip alçak geçiren fonksiyonuna ilişkin sıfırlar, kutuplar ve kazanç sabiti,

$$z_1, z_8 = \pm j 3.7555$$

$$z_2, z_7 = \pm j 1.5040$$

$$z_3, z_6 = \pm j 1.1746$$

$$z_4, z_5 = \pm j 1.0945$$

$$p_1, p_8 = -0.3182 \pm j 0.3190$$

$$p_2, p_7 = -0.1704 \pm j 0.7626$$

$$p_3, p_6 = -6.3635 \times 10^{-2} \pm j 0.9457$$

$$p_4, p_5 = -1.5118 \times 10^{-2} \pm j 1.0026$$

$$h_0 = 2.8627$$

olarak elde edilir. Filtre 8.dereceden bir fonksiyona sahip olup, iki quad bloğundan oluşmaktadır. Alt devrelere ilişkin kazanç katsayıları Tablo A5’de verilmektedir.

Tablo A5.Örnek 3’e ilişkin kazanç katsayıları

Katsayı	1.Quad Blok	2.Quad Blok
k_{1i}	0.0207	0.6519
k_{2i}	0.1523	0.7098
K	1.3715	

Kazanç katsayıları atanmış alt devrelere ilişkin transfer fonksiyonları, geribesleme katsayıları, iyilik faktörleri ise şöyledir;

$$T_{11}(s) = (2.0708 \times 10^{-2} s^2 + 1.5248 \times 10^{13}) / (s^2 + 3527717s + 2.0671 \times 10^{13})$$

$$T_{21}(s) = (0.1523 s^2 + 1.7996 \times 10^{13}) / (s^2 + 3527717s + 2.0671 \times 10^{13})$$

$$Q_1 = 1.2888 \quad f_1' = -0.3267$$

(A3.a)

$$T_{12}(s) = (0.6513 s^2 + 4.6927 \times 10^{13}) / (s^2 + 568575.4s + 4.9643 \times 10^{13})$$

$$T_{22}(s) = (0.7098 s^2 + 4.4404 \times 10^{13}) / (s^2 + 568575.4s + 4.9643 \times 10^{13})$$

$$Q_2 = 12.3920 \quad f_2' = -1.8203 \times 10^{-3}$$

(A3.b)

Alçak geçiren Eliptik yaklaşık filtreye ilişkin eleman değerleri Tablo A6'da verilmektedir.

Tablo A6.Örnek 3'e ilişkin eleman değerleri

Elemanlar $g_i(\mu S), C_i(pF)$	1.Quad	2.Quad
g_1	135.5635	105.7865
g_2	100	100
g_3	2.070	65.13758
g_4	100	100
g_5	100	100
g_6	114.8634	111.7977
g_7	100	100
g_8	15.2383	70.9892
g_9	100	100
g_{10}	100	100
g_x	100	100
g_y	100	100
g_z	32.6770	0.1820
C_1	23.1342	1.2115
C_2	0.5870	114.5628
C_3	19.6017	1.2804
C_4	4.3196	124.8545
g_a	137.15	
g_b	100	

Örnek 4

$f_{a1} = 1.2 \text{ Mhz}$, $f_{p1} = 1.5 \text{ Mhz}$, $f_{p2} = 1.7 \text{ Mhz}$, $f_{a2} = 1.9 \text{ Mhz}$ $A_p < 0.5 \text{ dB}$, $A_s > 40 \text{ dB}$ özelliklerine sahip band geçiren Butterworth filtre tasarlanmak isteniyor.

Verilen özellikleri sağlayan prototip alçak geçiren fonksiyonuna ilişkin kutuplar ve kazanç sabiti,

$$p_1, p_6 = -0.3084 \pm j 1.1509$$

$$p_2, p_5 = -0.8425 \pm j 0.8425$$

$$p_3, p_4 = -1.1509 \pm j 0.3084$$

$$h_o = 2.862776$$

bulunur. Filtre üç quad bloğundan oluşmaktadır. Tablo A7’de alt devrelere ilişkin kazanç katsayıları verilmektedir.

Tablo A7.Örnek 4’e ilişkin kazanç katsayıları

Katsayı	1.Quad	2.Quad	3.Quad
k_{1i}	1.2336	1.6787	1.4372
k_{2i}	1.1509	0.8458	0.9879
K	1		

Kazanç katsayıları atanmış alt devrelere ilişkin transfer fonksiyonları ise şöyledir;

$$T_{11}(s) = (15502355s)/(s^2 + 1446389s + 1.0067 \times 10^{14})$$

$$T_{21}(s) = (1446389s)/(s^2 + 1446389s + 1.0067 \times 10^{14})$$

$$Q_1 = 6.9368 \quad f_1' = 6.698710^{-2}$$

(A4.a)

$$T_{12}(s) = (2109614s)/(s^2 + 1058831s + 1.0067 \times 10^{14})$$

$$T_{22}(s) = (1062869s)/(s^2 + 1058831s + 1.0067 \times 10^{14})$$

$$Q_2 = 9.4759 \quad f_2' = 0.5$$

(A4.b)

$$T_{13}(s) = (1806104s)/(s^2 + 387558.9s + 1.0067 \times 10^{14})$$

$$T_{23}(s) = (1241481s)/(s^2 + 387558.9s + 1.0067 \times 10^{14})$$

$$Q_3 = 25.8888$$

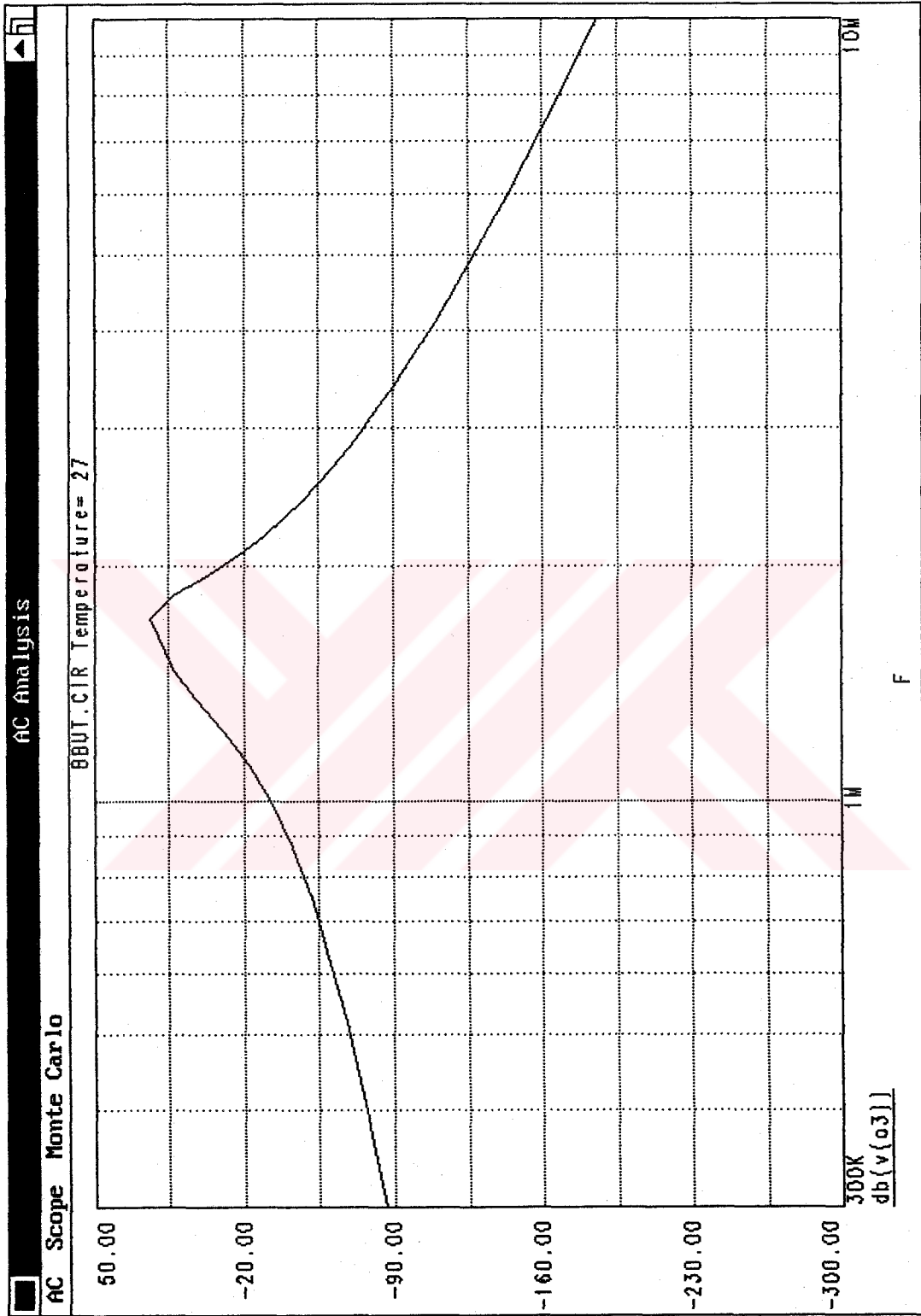
$$f_3' = 0.9330$$

(A4.c)

Band geçiren Butterworth yaklaşıklı filtreye ilişkin eleman değerleri Tablo A8'de verilmektedir.

Tablo A8. Örnek 4'e ilişkin eleman değerleri

Elemanlar $g_i(\mu S), C_i(pF)$	1. Quad	2. Quad	3. Quad
g_1	100	100	100
g_2	93.3012	50.1907	21.4582
g_3	100	100	100
g_4	100	100	100
g_5	99.9999	99.6200	31.2174
g_6	100	100	100
g_x	100	100	100
g_y	100	100	100
g_z	6.6987	50	93.3013
C_1	1.4367	1.0517	0.3849
C_2	64.5063	47.4020	55.3678
C_3	1.4367	1.0517	0.3849
C_4	69.1376	94.0849	8.0548
g_a	100		
g_b	100		



Şekil A3. Örnek 4'deki Band geçiren Butterworth filtresine ilişkin Frekans Karakteristiği

Örnek 5

$f_{a1} = 1.1$ Mhz, $f_{p1} = 1.2$ Mhz, $f_{p2} = 1.6$ Mhz, $f_{a2} = 1.7$ Mhz, $A_p < 0.5$ dB, $A_a > 30$ dB özelliklerine sahip band geçiren Chebyshev filtre tasarlanmak isteniyor.

Verilen özellikleri sağlayan prototip alçak geçiren fonksiyonuna ilişkin kutuplar ve kazanç sabiti,

$$p_1, p_6 = -7.7650 \pm j 1.008$$

$$p_2, p_5 = -0.212 \pm j 0.7382$$

$$p_3, p_4 = -0.2897 \pm j 0.2702$$

$$h_o = 0.0894$$

olarak elde edilir. Filtre üç quad bloğundan oluşmaktadır. Tablo A7’de alt devrelere ilişkin kazanç katsayıları verilmektedir.

Tablo A9.Örnek 5’e ilişkin kazanç katsayıları

Katsayı	1.Quad	2.Quad	3.Quad
k_{1i}	0.4467	1.3800	1.0031
k_{2i}	0.3514	0.4219	0.9755
K	0.9999		

Kazanç katsayıları atanmış alt devrelere ilişkin transfer fonksiyonları ise şöyledir;

$$T_{11}(s) = (1122755s)/(s^2 + 728331.8s + 7.5798 \times 10^{13})$$

$$T_{21}(s) = (883256.7s)/(s^2 + 728331.8s + 7.5798 \times 10^{13})$$

$$Q_1 = 11.95367 \quad f_1' = 0.4650$$

(A5.a)

$$T_{12}(s) = (3468509s)/(s^2 + 533175.9s + 7.5798 \times 10^{13})$$

$$T_{22}(s) = (1060460s) / (s^2 + 533175.9s + 7.5798 \times 10^{13})$$

$$Q_2 = 16.3290 \quad f_2' = 0.9359$$

(A5.b)

$$T_{13}(s) = (2521069s) / (s^2 + 195155.9s + 7.5798 \times 10^{13})$$

$$T_{23}(s) = (2451793s) / (s^2 + 195155.9s + 7.5798 \times 10^{13})$$

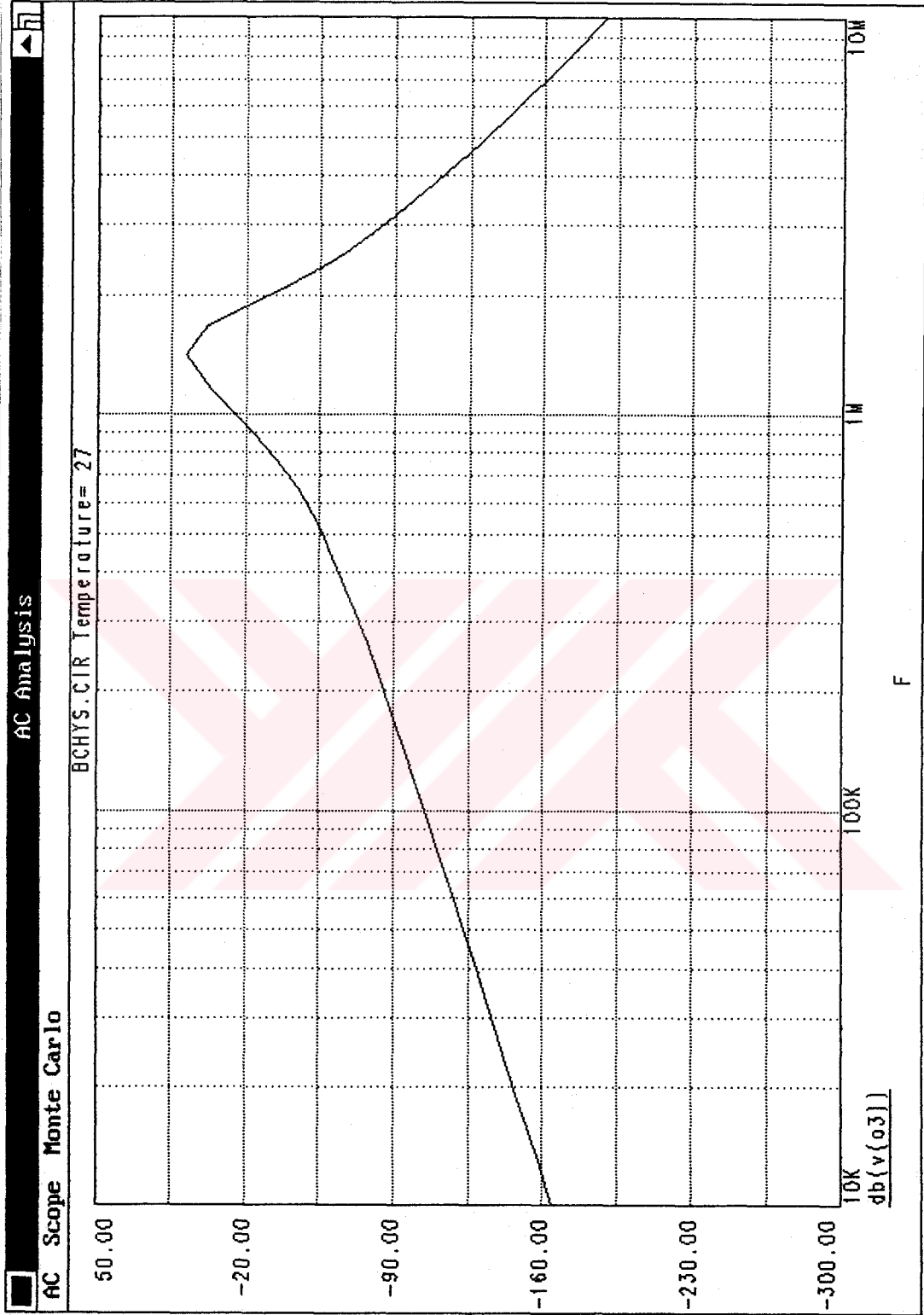
$$Q_3 = 44.6117 \quad f_3' = 1.0392$$

(A5.c)

Band geçiren Chebyshev yaklaşıkli filtreye ilişkin eleman değerleri Tablo A10'da verilmektedir.

Tablo A10.Örnek 5'e ilişkin eleman değerleri

Elemanlar $g_i(\mu S), C_i(pF)$	1.Quad	2.Quad	3.Quad
g_1	100	100	100
g_2	64.8700	15.3719	7.741
g_3	100	100	100
g_4	100	100	100
g_5	82.4598	50.2778	7.9597
g_6	100	100	100
g_x	100	100	100
g_y	100	100	100
g_z	46.5083	93.5929	103.9272
C_1	0.9608	0.7034	0.2574
C_2	89.0665	28.8308	39.6657
C_3	0.9608	0.7034	0.2574
C_4	113.2174	94.2987	40.7864
g_a	99.9997		
g_b	100		



Şekil A4. Örnek 5'deki Band geçiren Chebyshev filtresine ilişkin Frekans Karakteristiği

Örnek 6

$f_{a1} = 400 \text{ KHz}$, $f_{p1} = 500 \text{ KHz}$, $f_{p2} = 550 \text{ KHz}$, $f_{a2} = 1.5 \text{ Mhz}$, $A_p < 0.5 \text{ dB}$, $A_s > 55 \text{ dB}$ özelliklerine sahip band geçiren Eliptik filtre tasarlanmak isteniyor.

Bu özellikleri sağlayan prototip alçak geçiren fonksiyonuna ilişkin sıfırlar, kutuplar ve kazanç sabiti,

$$z_1, z_2 = \pm j 6.6268$$

$$p_1, p_3 = -0.3083 \pm j 1.0238$$

$$p_2 = -0.6331$$

$$h_0 = 1.6485 \times 10^{-2}$$

olarak elde edilir. Filtre bir T_b bloğu ve bir quad bloğundan oluşmaktadır. Tablo A11'de alt devrelere ilişkin kazanç katsayıları verilmektedir.

Tablo A11. Örnek 6'ya ilişkin kazanç katsayıları

Katsayı	Giriş Bloğu	T_b Bloğu	Quad Blok
k_{1i}	—	—	0.1076
k_{2i}	—	—	0.2418
K	0.9999	—	—
k_b	—	0.6331	—

Kazanç değerleri atanmış alt devrelere ilişkin transfer fonksiyonları, iyilik faktörleri ve geribesleme katsayıları aşağıdaki gibi olur.

$$T_b(s) = (198921.9s)/(s^2 + 198921.8s + 1.0856 \times 10^{13})$$

$$Q_b = 16.5639$$

$$k_b = 0.6331$$

(A6.a)

$$T_1(s) = (0.1102s^2 + 2.2293s \times 10^{12}) / (s^2 + 99234.91s + 1.0856 \times 10^{13})$$

$$T_2(s) = (0.2418s^2 + 1.4101 \times 10^{12}) / (s^2 + 99234.91s + 1.0856 \times 10^{13})$$

$$Q_1 = 33.2033$$

$$f_1' = 0.9148$$

(A6.b)

Aşağıda, band geçiren Eliptik yaklaşık filtreye ilişkin eleman değerleri verilmektedir

Giriş Bloğu

$$g_a = 99.9999 \mu S$$

$$g_b = 100 \mu S$$

T_b Bloğu

$$g_{b1} = g_{b3} = 100 \mu S$$

$$g_{b2} = 99.9999 \mu S$$

$$C_{b1} = 1.8322 pF$$

$$C_{b2} = 0.5027 nF$$

Quad Blok

$$g_2 = g_4 = g_5 = g_7 = g_9 = g_{10} = g_x = g_y = 100 \mu S$$

$$g_1 = 486.9929 \mu S$$

$$g_3 = 11.0265 \mu S$$

$$g_6 = 769.9106 \mu S$$

$$g_8 = 24.1877 \mu S$$

$$C_1 = 4.4513 pF$$

$$C_2 = 0.1111 nF$$

$$C_3 = 7.0374 pF$$

$$C_4 = 0.2437 nF$$

EK B

Verilen bilgisayar programı QUICKBASIC programlama dili kullanılarak yazılmıştır. Program yardımıyla Butterworth, Chebyshev, Eliptik yaklaşıklık tiplerinden biri seçilerek alçak geçiren filtreler için dört filtre özelliği (f_a , f_p , A_p , A_a), band geçiren filtreler için ise girilen altı filtre özelliği (f_{a1} , f_{p1} , f_{p2} , f_{a2} , A_p , A_a) ile quad ve kaskad yöntemi kullanılarak alçak geçiren ve band geçiren OTA-C filtreleri tasarlanabilmektedir. Ayrıca filtrenin frekans cevabı grafik olarak elde edilebilmektedir.



İSTANBUL TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ALÇAK GEÇİREN VE BAND GEÇİREN OTA-C
FİLTRELERİNİN QUAD VE KASKAD YÖNTEMİ İLE TASARIMI

Yüksek lisans tezi

Ömer ORAK

Yöneten: Prof. Dr. Ali Nur GÖNÜLEREN

-----ANA PROGRAM-----

```

DECLARE SUB sekil (sol%, sag%, ust%, alt%)
DECLARE SUB lplprans ()
DECLARE SUB spe2 ()
DECLARE SUB dinamik ()
DECLARE SUB transyaz ()
DECLARE SUB dinamikbc ()
DECLARE SUB dinamikell ()
DECLARE SUB eleman ()
DECLARE SUB grafik ()
DECLARE SUB frame ()
DECLARE FUNCTION addre! (a1!, a2!, b1!, b2!)
DECLARE FUNCTION addim! (a1!, a2!, b1!, b2!)
DECLARE FUNCTION mulre! (a1!, a2!, b1!, b2!)
DECLARE FUNCTION mulim! (a1!, a2!, b1!, b2!)
DECLARE FUNCTION divre! (a1!, a2!, b1!, b2!)
DECLARE FUNCTION divim! (a1!, a2!, b1!, b2!)
DECLARE SUB ekran ()
DECLARE SUB delay2 ()
DECLARE SUB kaskad ()
DECLARE FUNCTION geribes! (kk!, ll!, mm!)
DECLARE SUB polezero ()
DECLARE SUB quads ()
DECLARE SUB lpbprans ()
DECLARE SUB prototip ()
DECLARE SUB katsay ()
DECLARE SUB delay1 ()
DECLARE SUB delay ()
DECLARE SUB anykey ()
DECLARE SUB ell (oma, Ap, Aa)
DECLARE SUB chy (oma, Ap, Aa)
DECLARE SUB bw (oma, Ap, Aa)
DECLARE SUB basla ()
DECLARE SUB specifications ()
DECLARE FUNCTION modul (pre, pim)
COMMON SHARED ft$, ck, ta$, nx$, tx, ee, C, ka, h, yt$, N$, tb$, wp, wa, oma
COMMON SHARED Ap, Aa, ho, N, m, w0, BAND, pi, w, wa1, wa2, wp1, wp2, po, so
COMMON SHARED mx, nx, qx, hy, S, maxgenlik, kq$, test, Sw, unit$
COMMON SHARED fa1, fp1, fp2, fa2, maxfr, minfr
DIM SHARED pre(40), pim(40), zre(40), zim(40)
'Tipik biquadratik terim : (ds2+es+c)/(s2+as+b)
DIM SHARED a(40), B(40), C(40), d(40), E(40)
'Tb(s)=(bb2s2+bb1s+bb0)/(ab2s2+ab1s+ab0)
COMMON SHARED aq, aq0, aq1, ab2, ab1, ab0, bb2, bb1, bb0
'TQ(s)=(b4s4+b3s3+b2s2+b1s+b0)/(a4s4+a3s3+a2s2+a1s+a0)
DIM SHARED a4(40), a3(40), a2(40), a1(40), a0(40), b4(40), b3(40),
DIM SHARED b2(40), b1(40), b0(40)

```

```

'pole-zero pairing için
DIM SHARED pole(40), zero(40)
'kaskad ve quadratik blokların katsayıları:
'Tq,c(s)=(b12s2+b11s+b10)/(a12s2+a11s+a10)
DIM SHARED b12(40), b11(40), b10(40), a12(40), a11(40), a10(40), b22(40)
DIM SHARED b21(40), b20(40), a22(40), a21(40), a20(40), wz1(40), wz2(40)
DIM SHARED wp1(40), wp2(40), q1(40), q2(40)
'quad için geribesleme katsayısı: f,fd=f/(k1*k2)
DIM SHARED F(20), fd(20), K(20)
'eleman değerleri ( gm , C )
DIM SHARED gx(20), gy(20), gz(20), g1(20), g2(20), g3(20), g4(20), g5(20)
DIM SHARED g6(20), g7(20), g8(20), g9(20), g10(20), c1(20), c2(20), c3(20)
DIM SHARED c4(20)
COMMON SHARED gb1, gb2, gb3, gb4, gb5, cb1, cb2, ga, gb
'bağlantı katsayıları:
DIM SHARED h(20)
'Dynamic range katsayıları:kd --->K=(k1i*k2i)^-1 , i=1,2,3...nx
COMMON SHARED kd, kb, kaz, r
DIM SHARED k1(20), k2(20), k12(20)
'grafik için
DIM SHARED genlik(580)
S = 580: test = 0: pi = 3.141592654#
CALL basla
start:
CLEAR , 40000
CLS : SCREEN 9
'filtre türü -----
CALL sekil(15, 65, 5, 20)
LOCATE 8, 34: PRINT "FİLTRE TÜRÜ"
COLOR 15, 1: LOCATE 10, 28: PRINT "-----"
LOCATE 11, 30: PRINT "[A]...Alçak geçiren"
LOCATE 14, 30: PRINT "[B]...Band geçiren"
LOCATE 16, 28: PRINT "-----"
basal:
LOCATE 17, 32: INPUT "Seçiminiz...>", fT$
IF fT$ = "B" OR fT$ = "b" THEN
fT$ = "BG"
GOTO basl1
END IF
IF fT$ = "a" OR fT$ = "A" THEN
fT$ = "AG"
GOTO basl1
END IF
BEEP: GOTO basal
basl1:
'yaklaşıklık türü -----
CLS
CALL sekil(15, 65, 5, 20)
LOCATE 8, 31: PRINT "YAKLAŞIKLIK TÜRÜ"
COLOR 15, 1: LOCATE 9, 25: PRINT "-----"
LOCATE 10, 30: PRINT "[B]...Butterworth"
LOCATE 12, 30: PRINT "[C]...Chebyshev"
LOCATE 14, 30: PRINT "[E]...Elliptik"
LOCATE 15, 25: PRINT "-----"
yine:
LOCATE 17, 31: INPUT "Seçiminiz...>", yt$
IF yt$ = "b" OR yt$ = "B" THEN
yt$ = "bw"
GOTO basl2
END IF

```

```

IF yt$ = "c" OR yt$ = "C" THEN
yt$ = "chy"
GOTO basl2
END IF
IF yt$ = "e" OR yt$ = "E" THEN
yt$ = "ell"
GOTO basl2
END IF
BEEP: GOTO yine
'-----

```

```

basl2:
CLS
CALL ekran
IF fts$ = "AG" THEN
CALL spe2
ELSE
CALL specifications
END IF
CLS
'-----

```

```

CALL ekran
IF yt$ = "bw" THEN
CALL bw(oma, Ap, Aa)
ELSEIF yt$ = "chy" THEN
CALL chy(oma, Ap, Aa)
ELSEIF yt$ = "ell" THEN
CALL ell(oma, Ap, Aa)
END IF
CALL anykey: CALL ekran: CALL polezero: CALL anykey: CALL ekran
CALL katsay: CALL anykey
IF fts$ = "AG" THEN
CALL lplptrans
ELSE
CALL lpbptrans
END IF
CLS

```

```

CALL sekil(15, 65, 5, 20)
LOCATE 8, 36: PRINT "YÖNTEM": LOCATE 9, 29: PRINT "-----"
COLOR 15, 1: LOCATE 11, 34: PRINT "[K]...Kaskad"
LOCATE 13, 34: PRINT "[Q]...Quad": COLOR 14, 1
LOCATE 15, 29: PRINT "-----": COLOR 15, 1

```

tekr:

```

LOCATE 17, 32: INPUT "Seçiminiz...>", kq$
IF kq$ = "k" OR kq$ = "K" THEN
CALL kaskad
ELSEIF kq$ = "q" OR kq$ = "Q" THEN
kq$ = "quad": CALL quads
ELSE
BEEP: GOTO tekr
END IF

```

```

CALL ekran: CALL dinamik: CALL eleman: CALL transyaz
yeniden:

```

```

IF fts$ = "AG" AND yt$ = "ell" THEN
CLS : CALL sekil(15, 65, 5, 20): COLOR 15, 1
LOCATE 9, 24: PRINT "-----"
LOCATE 10, 26: PRINT "[1]...Yeni tasarım için"
LOCATE 12, 26: PRINT "[2]...Programdan çıkmak için"
LOCATE 13, 24: PRINT "-----"

```

sec:

```

LOCATE 16, 31: INPUT "Seçiminiz...>", sec

```

```

IF sec = 1 THEN
GOTO start
ELSEIF sec = 2 THEN
END
ELSE
BEEP: GOTO sec
END IF
END IF
IF ft$ = "BG" AND yt$ = "ell" THEN
CALL dinamikell
END IF
IF yt$ = "bw" OR yt$ = "chy" THEN
CALL dinamikbc
END IF
CALL grafik: CALL anykey: CLS : CALL sekil(15, 65, 5, 20): COLOR 15, 1
LOCATE 8, 24: PRINT "-----"
LOCATE 9, 26: PRINT "[1]...Grafikleri görmek için"
LOCATE 11, 26: PRINT "[2]...Yeni tasarım için"
LOCATE 13, 26: PRINT "[3]...Programdan çıkmak için"
LOCATE 14, 24: PRINT "-----"
sz:
LOCATE 16, 31: INPUT "Seçiminiz...:>", sec
IF sec = 1 THEN
GOTO devam
ELSEIF sec = 2 THEN
GOTO start
ELSEIF sec = 3 THEN
END
ELSE
BEEP: GOTO sz
END IF
devam: test = test + 1
VIEW (0, 0)-(639, 349)
WINDOW (0, 0)-(639, 349)
GOTO yeniden: CLS
END

```

-----FUNCTION addim-----

```

-
FUNCTION addim (a1, a2, b1, b2)
'iki karmaşık sayının toplamının sanal kısmı
addim = a2 + b2
END FUNCTION
FUNCTION addre (a1, a2, b1, b2)
'iki karmaşık sayının toplamının reel kısmı
addre = a1 + b1
END FUNCTION

```

'-----ANYKEY Alt Programı-----

```

SUB anykey
yaz: LOCATE 23, 74: PRINT "anykey"
IF INKEY$ = "" THEN
GOTO yaz
ELSE
END IF
CLS
END SUB

```

'-----BASLA Alt Programı-----

```

SUB basla
CLS : SCREEN 9: COLOR 14, 1
LOCATE 10, 10: PRINT "ALÇAK GEÇİREN VE BAND GEÇİREN FİLTRE TASARIMI"

```

END SUB

-----BW Alt Programı-----

```
SUB bw (oma, Ap, Aa)
pi = 3.141592654#: N = LOG((10 ^ (.1 * Aa) - 1) / (10 ^ (.1 * Ap) - 1)) / (2 * LOG(oma))
N = FIX(N) + 1
'n tek mi çift mi.....
IF N MOD 2 = 0 THEN N$ = "even" ELSE N$ = "odd"
E = SQR(10 ^ (.1 * Ap) - 1): r = E ^ (-1 / N): ho = 1
FOR K = 1 TO N
pre(K) = r * COS((1 + (2 * K - 1) / N) * (pi / 2)): pim(K) = r * SIN((1 + (2 * K - 1) / N) * (pi / 2))
IF N$ = "odd" AND K = (N + 1) / 2 THEN
pim(K) = 0: po = pre(K)
ELSE:END IF
ho = ho * modul(-pre(K), -pim(K)): NEXT: CALL prototip
END SUB
```

-----CHY Alt Programı-----

```
SUB chy (oma, Ap, Aa)
pi = 3.141592654#: X = SQR((10 ^ (.1 * Aa) - 1) / (10 ^ (.1 * Ap) - 1))
pay = LOG(X + SQR(X ^ 2 - 1)): payda = LOG(oma + SQR(oma ^ 2 - 1))
N = pay / payda: N = FIX(N) + 1
'n tek mi çift mi.....
IF N MOD 2 = 0 THEN N$ = "even" ELSE N$ = "odd"
E = SQR(10 ^ (.1 * Ap) - 1)
V = (1 / N) * (LOG((1 / E) + SQR((1 / E) ^ 2 + 1)))
'sinh(v) ve cosh(v) lerin hesabı.....: R1=sinh(v)=(exp(v)-exp(-v))/2
R2=cosh(v)=(exp(v)+exp(-v))/2
R1 = (EXP(V) - EXP(-V)) / 2: R2 = (EXP(V) + EXP(-V)) / 2: ho = 1
FOR K = 1 TO N
pre(K) = -R1 * SIN((2 * K - 1) * pi / (2 * N))
pim(K) = -R2 * COS((2 * K - 1) * pi / (2 * N))
IF N$ = "odd" AND K = (N + 1) / 2 THEN
pim(K) = 0: po = pre(K)
ELSE
END IF
ho = ho * modul(-pre(K), -pim(K)): NEXT
IF N$ = "even" THEN ho = 10 ^ (-.05 * Ap) * ho
CALL prototip
END SUB
```

-----DELAY Alt Programı-----

```
SUB delay
FOR i = 1 TO 10000
i = i + 1
NEXT
END SUB
SUB delay1
FOR i = 1 TO 300
i = i + 1
NEXT
END SUB
SUB delay2
FOR i = 1 TO 500
i = i + 1
NEXT
END SUB
```

-----DİNAMİK Alt Programı-----

```
SUB dinamik
'k1,k2 ve K yı bulabilmek için tüm kritik nokta gerilimleri hesaplanıyor.
S = 580
```

```

'V(i,s) in anlamı ( s=1,2,..600 i=1,2,..nx ) : i.ci quadın variable adlı geriliminin/akımının
's hertz frekansındaki değeri.
'tüm filtrenin giriş gerilimi : Vi = a + j b -----
vire = 1:viim = 0
'Hb giriş bloğu -----
'Hb= ( nbre + j nbim )/( dbre + j dbim ) = hbre + j hbim
'Quad blokları-----
'quadın giriş gerilimi:vj
'1.ci biquadın pay ve paydası:H1=( n1re + j n1im )/( d1re + j d1im )
'1.biquadın giriş ve çıkış gerilimleri:v1,v2
'2.ci biquadın pay ve paydası:H2=( n2re + j n2im )/( d2re + j d2im )
'2.biquadın giriş ve çıkış gerilimleri:v2,v4
'1.ci ve 2.ci biquadların çarpımı: H41=H42*H1
'1/(1+fk*H41) terimini aşağıdaki sırayla hesaplar:
'1. UO=fk*H41 bulur
'2. U1=1+UO
'3. UU=hk/U1
'Tüm gerilimlerin maximum ve minimumunu bulmak için ilk max.ve min.lar sıfırlanıyor..
IF fTS = "AG" THEN ck = tx: ELSE ck = nx
DIM vjmax(ck), v2max(ck), v4max(ck)
DIM vjmin(ck), v2min(ck), v4min(ck)
vb1max = 0: vb1min = 1000: vq1max = 0: vq1min = 100: vq2max = 0: vq2min = 100
vb2min = 1000: vb2max = 0
FOR i = 1 TO ck
vjmax(i) = 0: v2max(i) = 0: v4max(i) = 0: vjmin(i) = 1000: v2min(i) = 1000
v4min(i) = 1000: NEXT
FOR i = 1 TO ck
h(i) = 1: NEXT
IF fTS = "AG" THEN
minfr = 0
ELSE
minfr = wp1
END IF
maxfr = wp2
'filtrenin tüm iç düğümlerindeki gerilimler bir frekans bandında
'hesaplanıyor.(Tb giriş bloğu filtrenin girişinden ayrılıp Vj(1) den impuls
'uygulanarak tüm kritik gerilimler hesaplanabilir..
a = (maxfr - minfr) / S
IF fTS = "AG" THEN
w = minfr
ELSE
w = minfr - a
END IF
CALL ekran: COLOR 14, 1: LOCATE 2, 3
PRINT "Düğüm Gerilimlerinin Hesaplanarak Biquadratik Kazanç Katsayılarının Bulunması"
COLOR 15, 1: LOCATE 5, 10: PRINT "Her kritik voltaj için , 580 nokta inceleniyor!"
IF fTS = "AG" THEN P$ = "0" ELSE P$ = "fp1"
LOCATE 6, 10: PRINT " frekans aralığı [", P$, ", fp2 ] "
LOCATE 8, 10: PRINT "Başlangıç frekansı.....": fp1: unit$: "Hz"
LOCATE 9, 10: PRINT "Artış frekansı.....": a / 2 / pi: "Hz"
LOCATE 10, 10: PRINT "Duruş frekansı.....": fp2: unit$: "Hz"
LOCATE 11, 10: PRINT "Veri noktası sayısı.....: 580"
LOCATE 14, 10: PRINT "Hesaplanıyor....."
IF unit$ = "K" THEN
bol = 1000
ELSEIF unit$ = "M" THEN
bol = 1000000
ELSE
bol = 1
END IF

```

```

FOR K = 1 TO S
w = w + a
LOCATE 14, 34:
IF unit$ = "K" THEN
bol = 1000
PRINT w / 2 / pi / bol: LOCATE 14, 45: PRINT unit$, "Hz"
ELSEIF unit$ = "M" THEN
bol = 1000000
PRINT w / 2 / pi / bol: LOCATE 14, 45: PRINT unit$, "Hz"
ELSE
PRINT w / 2 / pi: LOCATE 14, 45: PRINT "Hz"
END IF
LOCATE 14, 51: PRINT K; ".data noktası için"
IF ta$ = "var" THEN
vq1re = vire * ho: vq1im = viim * ho: nqre = aq: nqim = 0: dqre = aq0
dqim = w: hqre = divre(nqre, nqim, dqre, dqim)
hqim = divim(nqre, nqim, dqre, dqim): vq2re = vq1re * hqre
vq2im = vq1re * hqim: vare = vq2re: vaim = vq2im: vq1 = modul(vq1re, vqim)
IF vq1 > vq1max THEN
vq1max = vq1
ELSEIF vq1 < vq1min THEN
vq1min = vq1
ELSE
END IF
vq2 = modul(vq2re, vq2im)
IF vq2 > vq2max THEN
vq2max = vq2
ELSEIF vq2 < vq2min THEN
vq2min = vq2
ELSE
END IF
ELSE
vare = vire * ho: vaim = viim * ho
END IF
IF tb$ = "var" THEN
vblre = vare: vblim = vaim
'Hb=Nb/Db hesaplanıyor
nbre = bb0 - bb2 * w ^ 2: nbim = bb1 * w: dbre = ab0 - ab2 * w ^ 2
dbim = ab1 * w: hbre = divre(nbre, nbim, dbre, dbim)
hbim = divim(nbre, nbim, dbre, dbim)
'Vb2=Vi*Hb hesabı
vb2re = mulre(vblre, vblim, hbre, hbim)
vb2im = mulim(vblre, vblim, hbre, hbim)
'l.ci quadın giriş gerilimi Hb nin çıkış gerilimi
'ile aynıdır: Vj(1, k) = Vb2(k)
vcre = vb2re: vcim = vb2im
'Tb giriş bloğu için gerilim ve akım maximumları....
vb1 = modul(vblre, vblim)
IF vb1 > vb1max THEN
vb1max = vb1
ELSEIF vb1 < vb1min THEN
vb1min = vb1
ELSE
END IF
vb2 = modul(vb2re, vb2im)
IF vb2 > vb2max THEN
vb2max = vb2
ELSEIF vb2 < vb2min THEN
vb2min = vb2
ELSE

```

```

END IF
'Tb için maximumlar hesaplandı....
ELSE
vcre = vare: vcim = vaim
END IF
FOR i = 1 TO ck
vjre = vcre: vjim = vcim
'H1'in hesabı-----
n1re = b10(i) - b12(i) * w ^ 2: n1im = b11(i) * w
d1re = a10(i) - a12(i) * w ^ 2: d1im = a11(i) * w
hlre = divre(n1re, n1im, d1re, d1im): hlim = divim(n1re, n1im, d1re, d1im)
'H2'in hesabı-----
n2re = b20(i) - b22(i) * w ^ 2: n2im = b21(i) * w
d2re = a20(i) - a22(i) * w ^ 2: d2im = a21(i) * w
h2re = divre(n2re, n2im, d2re, d2im)
h2im = divim(n2re, n2im, d2re, d2im)
'H41=H42*H21 nin hesabı-----
'T=H41 olarak alınıyor.
'tre=H41re tim=H41im dir..
tre = mulre(h1re, h1im, h2re, h2im): tim = mulim(h1re, h1im, h2re, h2im)
'Quad için 1/(1+f*H1*H2) nin hesabı:----
uore = F(i) * tre: uoim = F(i) * tim: ulre = 1 + uore: ulim = uoim
uure = divre(1, 0, ulre, ulim): uuim = divim(1, 0, ulre, ulim)
'v1=(1/(1+f*H41))*vj=uu*vj-----
v1re = mulre(uure, uuim, vjre, vjim): v1im = mulim(uure, uuim, vjre, vjim)
v2re = mulre(v1re, v1im, h1re, h1im): v2im = mulim(v1re, v1im, h1re, h1im)
v4re = mulre(v2re, v2im, h2re, h2im): v4im = mulim(v2re, v2im, h2re, h2im)
vcre = v4re: vcim = v4im
'Kritik gerilimler için maximum ve minimumlar.....
vj = modul(vjre, vjim)
IF vj > vjmax(i) THEN
vjmax(i) = vj
ELSEIF vj < vjmin(i) THEN
vjmin(i) = vj
ELSE
END IF
v2 = modul(v2re, v2im)
IF v2 > v2max(i) THEN
v2max(i) = v2
ELSEIF v2 < v2min(i) THEN
v2min(i) = v2
ELSE
END IF
v4 = modul(v4re, v4im)
IF v4 > v4max(i) THEN
v4max(i) = v4
ELSEIF v4 < v4min(i) THEN
v4min(i) = v4
ELSE
END IF
'TQ/TC için max ve min.lar hesaplandı.....
NEXT: NEXT: kd = 1
IF fTS = "AG" THEN
GOTO ifl
END IF
IF N$ = "odd" THEN
kb = vb1max / vb2max: bb2 = bb2 * kb: bb1 = bb1 * kb: bb0 = bb0 * kb
ELSE
kb = 1:END IF: GOTO fors
ifl:

```



```

IF N$ = "odd" THEN
ka = vq1max / vq2max.aq = aq * ka
ELSE
ka = 1
END IF
IF nx$ = "odd" THEN
kb = vb1max / vb2max: bb2 = bb2 * kb: bb1 = bb1 * kb: bb0 = bb0 * kb
ELSE
kb = 1
END IF
fors: FOR i = 1 TO ck
k1(i) = vjmax(i) / v2max(i): k2(i) = v2max(i) / v4max(i)
k12(i) = k1(i) * k2(i): kd = kd / k12(i): F(i) = F(i) / k12(i)
b12(i) = b12(i) * k1(i): b11(i) = b11(i) * k1(i): b10(i) = b10(i) * k1(i)
b22(i) = b22(i) * k2(i): b21(i) = b21(i) * k2(i): b20(i) = b20(i) * k2(i)
NEXT: kd = kd / kb: IF fts$ = "AG" THEN
kd = kd / ka
END IF
kaz = kd * hoLOCATE 23, 6
PRINT "Bütün kritik voltaj ve kazanç çarpanları hesaplandı..."
CALL anykey
END SUB

```

```

'-----DİNAMİKBC Alt Programı-----
SUB dinamikbc
'AG bw, chy, ell ve BG bw,chy,ell için dinamiklik hesabı yapar.
IF test > 0 THEN GOTO goster ELSE GOTO hesapla
hesapla:
S = 580
'OTA için saturasyon gerilimi ve akımı:
vsat = 5: isat = .0015: vimax = 10
'V/I(i,s) in anlamı (s=1,2,..600 i=1,2,..nx) : i.ci quadın variable adlı
'geriliminin/akımının s hertzz frekansındaki degeri.
'tüm filtrenin giriş gerilimi : Vi = a + j b -----
vire = 1: viim = 0
'Hb giriş bloğu -----
'Hb= ( nbre + j nbim )/( dbre + j dbim ) = hbre + j hbim
'Hb nin tüm gerilim ve akımları:
'Quad blokları-----
'quadın giriş gerilimi:vj
'1.ci biquadın pay ve paydası:H1=( n1re + j n1im )/( d1re + j d1im )
'1.biquadın tüm gerilimi/akımları:v1,v2,v3,i1,i2,i3
'2.ci biquadın pay ve paydası:H2=( n2re + j n2im )/( d2re + j d2im )
'2.biquadın tüm gerilim ve akımları:v4,v5,v6,i4,i5,i6
'1.ci ve 2.ci biquadların çarpımı: H41=H42*H1
'bağlantı elemanlarının akımları:ix,iy,iz
'hk/(1+fk*H41) terimini aşağıdaki sırayla hesaplar:
'1. UO=fk*H41 bulur
'2. U1=1+UO
'3. UU=hk/U1
'Tüm gerilim ve akımların maximumunu bulmak için ilk max.lar sıfırlanıyor..
DIM vjmax(ck), v1max(ck), v2max(ck), v3max(ck), v4max(ck), v5max(ck)
DIM i1max(ck), i2max(ck), i3max(ck), i4max(ck), i5max(ck), i6max(ck)
DIM ixmax(ck), iymax(ck), izmax(ck)
vq1max = 0: vq2max = 0: vb1max = 0: vb2max = 0: vb3max = 0: ib1max = 0
ib2max = 0: ib3max = 0
FOR i = 1 TO ck
vjmax(i) = 0: v1max(i) = 0: v2max(i) = 0: v3max(i) = 0: v4max(i) = 0
v5max(i) = 0: i1max(i) = 0: i2max(i) = 0: i3max(i) = 0: i4max(i) = 0
i5max(i) = 0: i6max(i) = 0: ixmax(i) = 0: iymax(i) = 0: izmax(i) = 0
NEXT

```

```

'Kaz bloğu için..
OPEN "kaz.dat" FOR OUTPUT AS #21
'Tb için...
OPEN "tb.dat" FOR OUTPUT AS #20
'Tq/Tc için...
'Dosya numaraları: 1,2,3....9. olarak sıralanıyor.Max 9 adet dosya açılıyor.
'Dosya isimleri : Q1.dat,Q2.dat,Q3.dat,...Q9.dat şeklindedir.
'Her bir dosya bir Quad/Kaskad bloğuna ait tüm gerilim/akım büyüklüklerini
'(580 tane frekans için) tutmaktadır.
FOR i = 1 TO ck
d$ = "q" + CHR$(48 + i) + ".dat"
OPEN d$ FOR OUTPUT AS #i
NEXT
IF fTS = "AG" THEN
minfr = 0: maxfr = 1.5 * wa2
ELSE
minfr = .2 * wa1: maxfr = 1.2 * wa2
END IF
a = (maxfr - minfr) / S
IF fTS = "AG" THEN
w = minfr
ELSE
w = minfr - a
END IF
CALL ekran: COLOR 14, 1: LOCATE 2, 4
PRINT "      Tüm gerilim ve akımların hesaplanması.....": COLOR 15, 1
LOCATE 5, 3: PRINT "Her bir OTA'nın çıkış gerilimi ve akımının 580 nokta";
PRINT "incelenerek hesaplanması"
IF fTS = "AG" THEN
j$ = "[ 0 , 1.5*fa2 ]": j = 1.5
ELSE
j$ = "[0.2 * fa1, 1.2 * fa2]": j = 1.2
END IF
LOCATE 6, 4: PRINT "      Frekans aralığı      "; j$
LOCATE 8, 10: PRINT "Başlangıç frekansı.....": .2 * fa1; unit$, "Hz"
LOCATE 9, 10: PRINT "Artış frekansı.....": a / 2 / pi; "Hz"
LOCATE 10, 10: PRINT "Duruş frekansı.....": j * fa2; unit$, "Hz"
LOCATE 11, 10: PRINT "Veri noktası sayısı.....: 580"
LOCATE 14, 10: PRINT "Hesaplanıyor....."
IF unit$ = "K" THEN
bol = 1000
ELSEIF unit$ = "M" THEN
bol = 1000000
ELSE
bol = 1
END IF
'filtrenin tüm iç düğümlerindeki gerilim ve akımlar bir frekans bandında
'hesaplanıyor.
FOR K = 1 TO S
w = w + a: LOCATE 14, 34: IF unit$ = "K" THEN
bol = 1000
PRINT w / 2 / pi / bol: LOCATE 14, 45: PRINT unit$, "Hz"
ELSEIF unit$ = "M" THEN
bol = 1000000
PRINT w / 2 / pi / bol: LOCATE 14, 45: PRINT unit$, "Hz"
ELSE
PRINT w / 2 / pi: LOCATE 14, 45: PRINT "Hz"
END IF
LOCATE 14, 51: PRINT ""; K; ".data noktası için.."
IF ta$ = "var" THEN

```

```

vqlre = vire * kaz: vqlim = viim * kaz: nqre = aq: nqim = 0: dqre = aq0
dqim = w: hqre = divre(nqre, nqim, dqre, dqim)
hqim = divim(nqre, nqim, dqre, dqim): vq2re = mulre(vqlre, vqlim, hqre, hqim)
vq2im = mulim(vqlre, vqlim, hqre, hqim): vare = vq2re: vaim = vq2im
vql = modul(vqlre, vqlim): vqlmax = vql: vq2 = modul(vq2re, vq2im)
IF vq2 > vq2max THEN
vq2max = vq2
ELSE
END IF
ELSE
vare = vire * kaz: vaim = viim * kaz
END IF
IF tb$ = "var" THEN
'Hb=Nb/Db hesaplanıyor
nbre = bb0 - bb2 * w ^ 2: nbim = bb1 * w: dbre = ab0 - ab2 * w ^ 2
dbim = ab1 * w: hbre = divre(nbre, nbim, dbre, dbim)
hbim = divim(nbre, nbim, dbre, dbim): vblre = vare: vblim = vaim
'Vb2=Vb1*Hb hesabı
vb2re = mulre(vblre, vblim, hbre, hbim)
vb2im = mulim(vblre, vblim, hbre, hbim)
'vb3=(-ab0/ab1*s)*vb2 den hesaplanacak-----
'yardımcı terim yb=(-bb0/bb1*s) olsun
nybre = -ab0: nybim = 0: dybre = 0: dybim = ab1 * w
ybre = divre(nybre, nybim, dybre, dybim)
ybim = divim(nybre, nybim, dybre, dybim)
vb3re = mulre(ybre, ybim, vb2re, vb2im)
vb3im = mulim(ybre, ybim, vb2re, vb2im)
'1.ci quadın giriş gerilimi Hb nin çıkış gerilimi
'ile aynıdır: Vj(1, k) = Vb2(k)
vcre = vb2re: vcim = vb2im
'Tb giriş bloğu için gerilim ve akım maximumları....
vb1 = modul(vblre, vblim)
IF vb1 > vb1max THEN
vb1max = vb1
ELSE
END IF
vb2 = modul(vb2re, vb2im)
IF vb2 > vb2max THEN
vb2max = vb2
ELSE
END IF
vb3 = modul(vb3re, vb3im)
IF vb3 > vb3max THEN
vb3max = vb3
ELSE
END IF
vb2re = -gb1 * vb2re: vb2im = -gb1 * vb2im: ib1 = modul(vb2re, vb2im)
IF ib1 > ib1max THEN
ib1max = ib1
ELSE
END IF
ib2re = addre(vb3re, vb3im, -vb2re, -vb2im)
ib2im = addim(vb3re, vb3im, -vb2re, -vb2im): ib2re = gb2 * ib2re
ib2im = gb2 * ib2im: ib2 = modul(ib2re, ib2im)
IF ib2 > ib2max THEN
ib2max = ib2
ELSE
END IF
ib3 = gb3: ib3max = gb3
'Tb için maximumlar hesaplandı....

```

```

PRINT #20, vb1; vb2; vb3; ib1; ib2; ib3
ELSE
vcre = vare: vcim = vaim
END IF
ia = ga: iamax = ia
IF ta$ = "var" THEN
ib = gb * vq1: ibmax = ib
ELSE
END IF
IF ta$ = "yok" THEN
IF tb$ = "var" THEN
ib = gb * vb1: ibmax = ib
ELSEIF tb$ = "yok" THEN
ib = gb * vire * kaz: ibmax = ib
END IF
ELSE
END IF
PRINT #21, ia; ib
FOR i = 1 TO ck
vjre = vcre: vjim = vcim
nlre = b10(i) - b12(i) * w ^ 2: nlim = b11(i) * w
dlre = a10(i) - a12(i) * w ^ 2: dlim = a11(i) * w
hlre = divre(nlre, nlim, dlre, dlim): hlim = divim(nlre, nlim, dlre, dlim)
n2re = b20(i) - b22(i) * w ^ 2: n2im = b21(i) * w
d2re = a20(i) - a22(i) * w ^ 2: d2im = a21(i) * w
h2re = divre(n2re, n2im, d2re, d2im): h2im = divim(n2re, n2im, d2re, d2im)
'H41=H42*H21 nin hesabı-----
'T=H41 olarak alınıyor.
'tre=H41re tim=H41lim dir..
tre = mulre(hlre, hlim, h2re, h2im): tim = mulim(hlre, hlim, h2re, h2im)
'Quad için 1/(1+f*H1*H2) nin hesabı:----
uore = F(i) * tre: uoim = F(i) * tim: ulre = 1 + uore: ulim = uoim
uure = divre(h(i), 0, ulre, ulim): uuim = divim(h(i), 0, ulre, ulim)
'v1=(h/(1+f*H41))*vj=uu*vj-----
vlre = mulre(uure, uuim, vjre, vjim): vlim = mulim(uure, uuim, vjre, vjim)
v2re = mulre(vlre, vlim, hlre, hlim): v2im = mulim(vlre, vlim, hlre, hlim)
'v3=(-a10/a11*s)*v2 den hesaplanacak-----
'yardımcı terim y1=(-a10/a11*s) olsun
nylre = -a10(i): nylim = 0: dylre = 0: dylim = a11(i) * w
ylre = divre(nylre, nylim, dylre, dylim)
ylim = divim(nylre, nylim, dylre, dylim)
v3re = mulre(ylre, ylim, v2re, v2im): v3im = mulim(ylre, ylim, v2re, v2im)
v4re = mulre(v2re, v2im, h2re, h2im): v4im = mulim(v2re, v2im, h2re, h2im)
'v5=(-a20/a21*s)*v4 den hesaplanacak-----
'yardımcı terim y2=(-a20/a21*s) olsun
ny2re = -a20(i): ny2im = 0: dy2re = 0: dy2im = a21(i) * w
y2re = divre(ny2re, ny2im, dy2re, dy2im)
y2im = divim(ny2re, ny2im, dy2re, dy2im)
v5re = mulre(y2re, y2im, v4re, v4im): v5im = mulim(y2re, y2im, v4re, v4im)
vare = v4re: vaim = v4im
'Tüm gerilim ve akımlar için maximumlar....
vj = modul(vjre, vjim)
IF vj > vjmax(i) THEN
vjmax(i) = vj
ELSE
END IF
v1 = modul(vlre, vlim)
IF v1 > v1max(i) THEN
v1max(i) = v1
ELSE

```

```

END IF
v2 = modul(v2re, v2im)
IF v2 > v2max(i) THEN
v2max(i) = v2
ELSE
END IF
v3 = modul(v3re, v3im)
IF v3 > v3max(i) THEN
v3max(i) = v3
ELSE
END IF
v4 = modul(v4re, v4im)
IF v4 > v4max(i) THEN
v4max(i) = v4
ELSE
END IF
v5 = modul(v5re, v5im)
IF v5 > v5max(i) THEN
v5max(i) = v5
ELSE
END IF
ilre = -g1(i) * v2re: ilim = -g1(i) * v2im: i1 = modul(ilre, ilim)
IF i1 > i1max(i) THEN
i1max(i) = i1
ELSE
END IF
i2re = addre(v3re, v3im, -v2re, -v2im): i2im = addim(v3re, v3im, -v2re, -v2im)
i2re = g2(i) * i2re: i2im = g2(i) * i2im: i2 = modul(i2re, i2im)
IF i2 > i2max(i) THEN
i2max(i) = i2
ELSE
END IF
i3re = g3(i) * v1re: i3im = g3(i) * v1im: i3 = modul(i3re, i3im)
IF i3 > i3max(i) THEN
i3max(i) = i3
ELSE
END IF
i4re = -g4(i) * v4re: i4im = -g4(i) * v4im: i4 = modul(i4re, i4im)
IF i4 > i4max(i) THEN
i4max(i) = i4
ELSE
END IF
i5re = addre(v5re, v5im, -v4re, -v4im): i5im = addim(v5re, v5im, -v4re, -v4im)
i5re = g5(i) * i5re: i5im = g5(i) * i5im: i5 = modul(i5re, i5im)
IF i5 > i5max(i) THEN
i5max(i) = i5
ELSE
END IF
i6re = g6(i) * v2re: i6im = g6(i) * v2im: i6 = modul(i6re, i6im)
IF i6 > i6max(i) THEN
i6max(i) = i6
ELSE
END IF
ixre = gx(i) * vjre: ixim = gx(i) * vjim: ix = modul(ixre, ixim)
IF ix > ixmax(i) THEN
ixmax(i) = ix
ELSE
END IF
iyre = -gy(i) * v1re: iym = -gy(i) * v1im: iy = modul(iyre, iym)
IF iy > iymax(i) THEN

```

```

iymax(i) = iy
ELSE
END IF
izre = -gz(i) * v4re: izim = -gz(i) * v4im: iz = modul(izre, izim)
IF iz > izmax(i) THEN
izmax(i) = iz
ELSE
END IF
'TQ/TC için maximumlar hesaplandı.....
'-----Elde edilen tüm gerilim/akımlar dosyaya atılacak-----"
PRINT #i, vj, v1; v2; v3; v4; v5; i1; i2; i3; i4; i5; i6; ix; iy; iz
NEXT
NEXT
CLOSE #20
CLOSE #21
FOR i = 1 TO ck
CLOSE #i
NEXT
'Maxiumlar yazılsın.....
CALL ekran
LOCATE 2, 6: COLOR 14, 1
PRINT " Maksimum gerilim transfer ve transfer admitans fonksiyonu...."
COLOR 15, 1: CALL anykey: CALL ekran: LOCATE 2, 34: COLOR 14, 1
PRINT " Giriş bloğu kazancı...": COLOR 15, 1: LOCATE 5, 2
PRINT "Yamax=", iamax * 1000000!; "[uA/V]": iamax = isat / iamax
IF iamax < vimax THEN
vimax = iamax: domb = 0: dom$ = "Ya"
ELSE
END IF
LOCATE 6, 2: PRINT "Ybmax=", ibmax * 1000000!; "[uA/V]": ibmax = isat / ibmax
IF ibmax < vimax THEN
vimax = ibmax: domb = 0: dom$ = "Yb"
ELSE
END IF
CALL anykey
IF tb$ = "var" THEN
CALL ekran: LOCATE 2, 34: COLOR 14, 1: PRINT "Tb Bloğu...": COLOR 15, 1
LOCATE 5, 2: PRINT "Hb1max=", vb1max: vb1max = vsat / vb1max: vimax = vb1max
domb = 0: dom$ = "Hb1": LOCATE 6, 2: PRINT "Hb2max=", vb2max
vb2max = vsat / vb2max: vimax = vb2max: domb = 0: dom$ = "Hb2"
LOCATE 7, 2: PRINT "Hb3max=", vb3max: vb3max = vsat / vb3max
IF vb3max < vimax THEN
vimax = vb3max: domb = 0: dom$ = "Hb3"
ELSE
END IF
LOCATE 5, 30: PRINT "Yb1max=", ib1max * 1000000!; "[uA/V]"
ib1max = isat / ib1max
IF ib1max < vimax THEN
vimax = ib1max: domb = 0: dom$ = "Yb1"
ELSE
END IF
LOCATE 6, 30: PRINT "Yb2max=", ib2max * 1000000!; "[uA/V]"
ib2max = isat / ib2max
IF ib2max < vimax THEN
vimax = ib2max: domb = 0: dom$ = "Yb2"
ELSE
END IF
LOCATE 7, 30: PRINT "Yb3max=", ib3max * 1000000!; "[uA/V]"
ib3max = isat / ib3max
IF ib3max < vimax THEN

```

```

vimax = ib3max: domb = 0: dom$ = "Yb3"
ELSE
END IF
CALL anykey
FOR i = 1 TO ck
CALL ekran: LOCATE 2, 34: COLOR 14, 1:PRINT i; ".Blok":COLOR 15, 1:LOCATE 5, 2
PRINT "Hjmax="; vjmax(i): vjmax(i) = vsat / vjmax(i)
IF vjmax(i) < vimax THEN
vimax = vjmax(i): domb = i: dom$ = "Hj"
ELSE
END IF
LOCATE 6, 2: PRINT "H1max="; vlmax(i): vlmax(i) = vsat / vlmax(i)
IF vlmax(i) < vimax THEN
vimax = vlmax(i): domb = i: dom$ = "H1"
ELSE
END IF
LOCATE 7, 2: PRINT "H2max="; v2max(i)
v2max(i) = vsat / v2max(i)
IF v2max(i) < vimax THEN
vimax = v2max(i): domb = i: dom$ = "H2"
ELSE
END IF
LOCATE 8, 2: PRINT "H3max="; v3max(i): v3max(i) = vsat / v3max(i)
IF v3max(i) < vimax THEN
vimax = v3max(i): domb = i: dom$ = "H3"
ELSE
END IF
LOCATE 9, 2
PRINT "H4max="; v4max(i): v4max(i) = vsat / v4max(i)
IF v4max(i) < vimax THEN
vimax = v4max(i): domb = i: dom$ = "H4"
ELSE
END IF
LOCATE 10, 2: PRINT "H5max="; v5max(i): v5max(i) = vsat / v5max(i)
IF v5max(i) < vimax THEN
vimax = v5max(i): domb = i: dom$ = "H5"
ELSE
END IF
LOCATE 5, 30: PRINT "Y1max="; ilmax(i) * 1000000!; "[uA/V]"
ilmax(i) = isat / ilmax(i)
IF ilmax(i) < vimax THEN
vimax = ilmax(i): domb = i: dom$ = "Y1"
ELSE
END IF
LOCATE 6, 30: PRINT "Y2max="; i2max(i) * 1000000!; "[uA/V]"
i2max(i) = isat / i2max(i)
IF i2max(i) < vimax THEN
vimax = i2max(i): domb = i: dom$ = "Y2"
ELSE
END IF
LOCATE 7, 30: PRINT "Y3max="; i3max(i) * 1000000!; "[uA/V]"
i3max(i) = isat / i3max(i)
IF i3max(i) < vimax THEN
vimax = i3max(i): domb = i: dom$ = "Y3"
ELSE
END IF
LOCATE 8, 30: PRINT "Y4max="; i4max(i) * 1000000!; "[uA/V]"
i4max(i) = isat / i4max(i)
IF i4max(i) < vimax THEN
vimax = i4max(i)

```

```

domb = i: dom$ = "Y4"
ELSE
END IF
LOCATE 9, 30: PRINT "Y5max="; i5max(i) * 1000000!; "[uA/V]"
i5max(i) = isat / i5max(i)
IF i5max(i) < vimax THEN
vimax = i5max(i): domb = i: dom$ = "Y5"
ELSE
END IF
LOCATE 10, 30: PRINT "Y6max="; i6max(i) * 1000000!; "[uA/V]"
i6max(i) = isat / i6max(i)
IF i6max(i) < vimax THEN
vimax = i6max(i): domb = i: dom$ = "Y6"
ELSE
END IF
LOCATE 11, 30: PRINT "Yxmax="; ixmax(i) * 1000000!; "[uA/V]"
ixmax(i) = isat / ixmax(i)
IF ixmax(i) < vimax THEN
vimax = ixmax(i): domb = i: dom$ = "Yx"
ELSE
END IF
LOCATE 12, 30: PRINT "Yymax="; iymax(i) * 1000000!; "[uA/V]"
iymax(i) = isat / iymax(i)
IF iymax(i) < vimax THEN
vimax = iymax(i): domb = i: dom$ = "Yy"
ELSE
END IF
IF kq$ = "quad" THEN
LOCATE 13, 30: PRINT "Yzmax="; izmax(i) * 1000000!; "[uA/V]"
izmax(i) = isat / izmax(i)
IF izmax(i) < vimax THEN
vimax = izmax(i): domb = i: dom$ = "Yz"
ELSE
END IF
ELSE
END IF
CALL anykey
NEXT
ELSE
FOR i = 1 TO ck
CALL ekran: LOCATE 2, 35: COLOR 14, 1: PRINT i; "Blok": COLOR 15, 1
LOCATE 5, 2: PRINT "Hjmax="; vjmax(i); vjmax(i) = vsat / vjmax(i)
IF vjmax(i) < vimax THEN
vimax = vjmax(i): domb = i: dom$ = "Hj"
ELSE
END IF
LOCATE 6, 2: PRINT "H1max="; v1max(i); v1max(i) = vsat / v1max(i)
IF v1max(i) < vimax THEN
vimax = v1max(i): domb = i: dom$ = "H1"
ELSE
END IF
LOCATE 7, 2: PRINT "H2max="; v2max(i); v2max(i) = vsat / v2max(i)
IF v2max(i) < vimax THEN
vimax = v2max(i): domb = i: dom$ = "H2"
ELSE
END IF
LOCATE 8, 2: PRINT "H3max="; v3max(i); v3max(i) = vsat / v3max(i)
IF v3max(i) < vimax THEN
vimax = v3max(i): domb = i: dom$ = "H3"
ELSE

```



```

END IF
LOCATE 9, 2: PRINT "H4max="; v4max(i)
v4max(i) = vsat / v4max(i)
IF v4max(i) < vimax THEN
vimax = v4max(i): domb = i: dom$ = "H4"
ELSE
END IF
LOCATE 10, 2: PRINT "H5max="; v5max(i): v5max(i) = vsat / v5max(i)
IF v5max(i) < vimax THEN
vimax = v5max(i): domb = i: dom$ = "H5"
ELSE
END IF
LOCATE 5, 30: PRINT "Y1max="; i1max(i) * 1000000!; "[uA/V]"
i1max(i) = isat / i1max(i)
IF i1max(i) < vimax THEN
vimax = i1max(i): domb = i: dom$ = "Y1"
ELSE
END IF
LOCATE 6, 30: PRINT "Y2max="; i2max(i) * 1000000!; "[uA/V]"
i2max(i) = isat / i2max(i)
IF i2max(i) < vimax THEN
vimax = i2max(i): domb = i: dom$ = "Y2"
ELSE
END IF
LOCATE 7, 30: PRINT "Y3max="; i3max(i) * 1000000!; "[uA/V]"
i3max(i) = isat / i3max(i)
IF i3max(i) < vimax THEN
vimax = i3max(i): domb = i: dom$ = "Y3"
ELSE
END IF
LOCATE 8, 30: PRINT "Y4max="; i4max(i) * 1000000!; "[uA/V]"
i4max(i) = isat / i4max(i)
IF i4max(i) < vimax THEN
vimax = i4max(i): domb = i: dom$ = "Y4"
ELSE
END IF
LOCATE 9, 30: PRINT "Y5max="; i5max(i) * 1000000!; "[uA/V]"
i5max(i) = isat / i5max(i)
IF i5max(i) < vimax THEN
vimax = i5max(i): domb = i: dom$ = "Y5"
ELSE
END IF
LOCATE 10, 30: PRINT "Y6max="; i6max(i) * 1000000!; "[uA/V]"
i6max(i) = isat / i6max(i)
IF i6max(i) < vimax THEN
vimax = i6max(i): domb = i: dom$ = "Y6"
ELSE
END IF
LOCATE 11, 30: PRINT "Yxmax="; ixmax(i) * 1000000!; "[uA/V]"
ixmax(i) = isat / ixmax(i)
IF ixmax(i) < vimax THEN
vimax = ixmax(i): domb = i: dom$ = "Yx"
ELSE
END IF
LOCATE 12, 30
PRINT "Yymax="; iymax(i) * 1000000!; "[uA/V]": iymax(i) = isat / iymax(i)
IF iymax(i) < vimax THEN
vimax = iymax(i): domb = i: dom$ = "Yy"
ELSE
END IF

```

```

IF kq$ = "quad" THEN
LOCATE 13, 30: PRINT "Yzmax="; izmax(i) * 1000000!; "[uA/V]"
izmax(i) = isat / izmax(i)
IF izmax(i) < vimax THEN
vimax = izmax(i): domb = i: dom$ = "Yz"
ELSE
END IF
ELSE
END IF
CALL anykey
NEXT
END IF
'Maximumlar yazıldı-----
'Vimax hesaplandı-----
CALL ekran: LOCATE 2, 6: COLOR 14, 1
PRINT "    Filtre girişine uygulanabilecek maksimum giriş gerilimi..."
COLOR 15, 1: LOCATE 4, 6: PRINT "": LOCATE 5, 6: PRINT ""
LOCATE 6, 6: PRINT "": LOCATE 8, 6: PRINT "Vimax="; vimax * 1000; "mV"
IF domb = 0 THEN
LOCATE 10, 6: PRINT "Dominant Blok ..."; "Tb": LOCATE 11, 6
PRINT "Dominant H/Y ..."; " "; dom$
ELSE
LOCATE 10, 6: PRINT "Dominant Blok ..."; domb:LOCATE; 11, 6
PRINT "Dominant H/Y ..."; " "; dom$
END IF
CALL anykey
'grafik için v/i (gerilim/akım) secimi
goster:
'Elde edilen gerilim ve akımlardan hangisini istiyoruz
CALL ekran: COLOR 14, 1: LOCATE 2, 2
PRINT "    Blok seçimini yapınız"
COLOR 15, 1: LOCATE 4, 2
IF tb$ = "var" THEN
LOCATE 5, 4: PRINT "[ 20..Giriş kazanç bloğu için ]": LOCATE 6, 4
PRINT "[ 0 .. Tb Bloğu için ]"
FOR i = 1 TO ck
LOCATE i + 6, 4: PRINT "[ "; i; "..."; i; ".Blok için ]"
NEXT
ELSE
LOCATE 5, 4: PRINT "[ 20..Giriş kazanç bloğu için ]"
FOR i = 1 TO ck
LOCATE i + 5, 4: PRINT "[ "; i; "..."; i; ".Blok için ]"
NEXT
END IF
LOCATE 16, 4: INPUT "[ Seçiminiz... ] :> ", qx: CALL ekran: COLOR 14, 1
LOCATE 2, 20: PRINT "    H/Y seçimini yapınız": COLOR 15, 1: LOCATE 4, 4
IF qx = 20 THEN
LOCATE 5, 4: PRINT "[ 1 ... Ya ]": LOCATE 6, 4: PRINT "[ 2 ... Yb ] "
ELSEIF qx = 0 THEN
LOCATE 5, 4: PRINT "[ 1 ... Hb1 ]": LOCATE 6, 4: PRINT "[ 2 ... Hb2 ]"
LOCATE 7, 4: PRINT "[ 3 ... Hb3 ]": LOCATE 5, 30: PRINT "[ 4 ... Yb1 ]"
LOCATE 6, 30: PRINT "[ 5 ... Yb2 ]": LOCATE 7, 30: PRINT "[ 6 ... Yb3 ]"
ELSEIF qx <> 0 AND qx <> 20 THEN
LOCATE 5, 4: PRINT " [ 1 ... Hj ]": LOCATE 6, 4: PRINT " [ 2 ... H1 ]"
LOCATE 7, 4: PRINT " [ 3 ... H2 ]": LOCATE 8, 4: PRINT " [ 4 ... H3 ]"
IF qx = nx THEN
LOCATE 9, 4: PRINT " [ 5 ... H4 ]": LOCATE 9, 17: PRINT "*"
LOCATE 19, 4: PRINT "* Toplam filtrenin frekans cevabı"
ELSE
LOCATE 9, 4: PRINT " [ 5 ... H4 ]"

```

```

END IF
LOCATE 10, 4: PRINT " [ 6 ... H5 ]": LOCATE 5, 30: PRINT " [ 7 ... Y1 ]"
LOCATE 6, 30: PRINT " [ 8 ... Y2 ]": LOCATE 7, 30: PRINT " [ 9 ... Y3 ]"
LOCATE 8, 30: PRINT " [ 10... Y4 ]": LOCATE 9, 30: PRINT " [ 11... Y5 ]"
LOCATE 10, 30: PRINT " [ 12... Y6 ]": LOCATE 11, 30: PRINT " [ 13... Yx ]"
LOCATE 12, 30: PRINT " [ 14... Yy ]"
IF kq$ = "quad" THEN
LOCATE 13, 30: PRINT " [ 15... Yz ]"
ELSE
END IF
ELSE
END IF
LOCATE 22, 4
INPUT " [ Seçiminiz... ] :> ", hy
OPEN "tb.dat" FOR INPUT AS #20
OPEN "kaz.dat" FOR INPUT AS #21
FOR i = 1 TO ck
d$ = "q" + CHR$(48 + i) + ".dat"
OPEN d$ FOR INPUT AS #i
NEXT
maxgenlik = 0
FOR K = 1 TO S
IF qx = 20 THEN
IF hy = 1 THEN
INPUT #21, ia, ib: genlik(K) = ia
ELSEIF hy = 2 THEN
INPUT #21, ia, ib: genlik(K) = ib
ELSE
END IF
'Tb için
ELSEIF qx = 0 THEN
IF hy = 1 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = vb1
ELSEIF hy = 2 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = vb2
ELSEIF hy = 3 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = vb3
'admitanslar.....
ELSEIF hy = 4 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = ib1
ELSEIF hy = 5 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = ib2
ELSEIF hy = 6 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = ib3
ELSE
END IF
ELSEIF qx <> 0 AND qx <> 20 THEN
'quad için
IF hy = 1 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = vj
ELSEIF hy = 2 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = v1
ELSEIF hy = 3 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = v2
ELSEIF hy = 4 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = v3

```

```

ELSEIF hy = 5 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = v4
ELSEIF hy = 6 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = v5
'Admitanslar.....
ELSEIF hy = 7 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = i1
ELSEIF hy = 8 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = i2
ELSEIF hy = 9 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = i3
ELSEIF hy = 10 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = i4
ELSEIF hy = 11 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = i5
ELSEIF hy = 12 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = i6
ELSEIF hy = 13 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = ix
ELSEIF hy = 14 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = iy
ELSEIF kq$ = "quad" AND hy = 15 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, i1, i2, i3, i4, i5, i6, ix, iy, iz
genlik(K) = iz
ELSE
END IF
ELSE
END IF
IF genlik(K) > maxgenlik THEN
maxgenlik = genlik(K): mfr = K
ELSE
END IF
NEXT
CLOSE #20
CLOSE #21
FOR i = 1 TO ck
CLOSE #i
NEXT
END SUB

```

'-----DİNAMİKELL Alt Programı-----'

SUB dinamikell

'ell için dinamiklik hesabı yapar.

IF test > 0 THEN GOTO goster1 ELSE GOTO hesapla1

hesapla1:

S = 580

'OTA için saturasyon gerilimi ve akımı:

vsat = 5: isat = .0015: vimax = 10

'-----'

'V/I(i,s) in anlamı (s=1,2,..600 i=1,2,..nx) : i.ci quadın variable adlı

```

'geriliminin/akımının s hertz frekansındaki değeri.
'tüm filtrenin giriş gerilimi : Vi = a + j b
vire = 1: viim = 0
'Hb giriş bloğu -----
'Hb= ( nbre + j nbim )/( dbre + j dbim ) = hbre + j hbim
'Hb nin tüm gerilim ve akımları:
'Quad blokları-----
'quadın giriş gerilimi:vj
'1.ci biquadın pay ve paydası:H1=( n1re + j n1im )/( d1re + j d1im )
'1.biquadın tüm gerilimi/akımları:v1,v2,v3,v5,i1,i2,i3,i4,i5
'2.ci biquadın pay ve paydası:H2=( n2re + j n2im )/( d2re + j d2im )
'2.biquadın tüm gerilim ve akımları:v4,v6,v7,i6,i7,i8,i9,i10
'1.ci ve 2.ci biquadların çarpımı: H41=H42*H21
'bağlantı elemanlarının akımları:ix,iy,iz
'hk/(1+fk*H41) terimini aşağıdaki sırayla hesaplar:
'1. UO=fk*H41 bulur
'2. U1=1+UO
'3. UU=hk/U1
'-----
'Tüm gerilim ve akımların maximumunu bulmak için ilk max.lar sıfırlanıyor..
DIM vjmax(nx), v1max(nx), v2max(nx), v3max(nx), v4max(nx), v5max(nx)
DIM v6max(nx),v7max(nx),i1max(nx), i2max(nx), i3max(nx), i4max(nx), i5max(nx)
DIM i6max(nx), i7max(nx), i8max(nx), i9max(nx), i10max(nx),ixmax(nx), iymax(nx)
DIM izmax(nx)
vb1max = :vb2max = 0:vb3max = 0:ib1max = 0:ib2max = 0:ib3max = 0
FOR i = 1 TO nx
vjmax(i) = 0: v1max(i) = 0: v2max(i) = 0: v3max(i) = 0: v4max(i) = 0
v5max(i) = 0: v6max(i) = 0: v7max(i) = 0: i1max(i) = 0: i2max(i) = 0
i3max(i) = 0: i4max(i) = 0: i5max(i) = 0: i6max(i) = 0: i7max(nx) = 0
i8max(nx) = 0: i9max(nx) = 0: i10max(nx) = 0: ixmax(i) = 0: iymax(i) = 0
izmax(i) = 0
NEXT
'-----Gerilim/akımların atılacağı dosyalar açılıyor-----"
'Kaz bloğu için..
OPEN "kaz.dat" FOR OUTPUT AS #21
'Tb için...
OPEN "tb.dat" FOR OUTPUT AS #20
'Tq/Tc için...
'Dosya numaraları: 1,2,3....9. olarak sıralanıyor.Max 9 adet dosya açılıyor.
'Dosya isimleri : 1.dat,2.dat,3.dat,....9.dat şeklindedir.
'Her bir dosya bir Quad/Kaskad bloğuna ait tüm gerilim/akım büyüklüklerini
'(580 tane frekans için) tutmaktadır.
FOR i = 1 TO nx
d$ = "q" + CHR$(48 + i) + ".dat"
OPEN d$ FOR OUTPUT AS #i
NEXT
'-----Dosyalar açıldı-----
minfr = .2 * wa1: maxfr = 1.2 * wa2: a = (maxfr - minfr) / S: w = minfr - a
'hesap başlıyor..
CALL ekran: COLOR 14, 1: LOCATE 2, 4
PRINT "      Tüm gerilim ve akımların hesaplanması.....": COLOR 15, 1
LOCATE 5, 3: PRINT "Her bir OTA'nın çıkış gerilimi ve akımının 580 nokta ";
PRINT "incelenerek hesaplanması"
LOCATE 6, 4: PRINT "Frekans aralığı [ 0.2*fa1 , 1.2*fa2 ]"
LOCATE 8, 10: PRINT "Başlangıç frekansı.....": 2 * fa1; unit$: "Hz"
LOCATE 9, 10: PRINT "Artış frekansı.....": a / 2 / pi; "Hz"
LOCATE 10, 10: PRINT "Duruş frekansı.....": 1.2 * fa2; unit$: "Hz"
LOCATE 11, 10: PRINT "Veri noktası sayısı.....: 580"
LOCATE 14, 10: PRINT "Hesaplanıyor....."
IF unit$ = "K" THEN

```

```

bol = 1000
ELSEIF unit$ = "M" THEN
bol = 1000000
ELSE
bol = 1
END IF
'filtrenin tüm iç düğümlerindeki gerilim ve akımlar bir frekans bandında
'hesaplanıyor.
FOR K = 1 TO S
w = w + a: LOCATE 14, 34: IF unit$ = "K" THEN
bol = 1000
PRINT w / 2 / pi / bol: LOCATE 14, 45: PRINT unit$, "Hz"
ELSEIF unit$ = "M" THEN
bol = 1000000: PRINT w / 2 / pi / bol: LOCATE 14, 45: PRINT unit$, "Hz"
ELSE
PRINT w / 2 / pi: LOCATE 14, 45: PRINT "Hz"
END IF
LOCATE 14, 51: PRINT " "; K; ".data noktası için.."
IF tb$ = "var" THEN
'Hb=Nb/Db hesaplanıyor
nbre = bb0 - bb2 * w ^ 2: nbim = bb1 * w: dbre = ab0 - ab2 * w ^ 2
dbim = ab1 * w: hbre = divre(nbre, nbim, dbre, dbim)
hbim = divim(nbre, nbim, dbre, dbim): vb1re = vire * kaz: vb1im = viim * kaz
'Vb2=Vb1*Hb hesabı
vb2re = mulre(vb1re, vb1im, hbre, hbim)
vb2im = mulim(vb1re, vb1im, hbre, hbim)
'vb3=(-ab0/ab1*s)*vb2 den hesaplanacak-----
'yardımcı terim yb=(-bb0/bb1*s) olsun
nybre = -ab0: nybim = 0: dybre = 0: dybim = ab1 * w
ybre = divre(nybre, nybim, dybre, dybim)
ybim = divim(nybre, nybim, dybre, dybim)
vb3re = mulre(ybre, ybim, vb2re, vb2im)
vb3im = mulim(ybre, ybim, vb2re, vb2im)
'1.ci quadın giriş gerilimi Hb nin çıkış gerilimi
'ile aynıdır: Vj(1, k) = Vb2(k)
vare = vb2re: vaim = vb2im
'Tb giriş bloğu için gerilim ve akım maximumları....
vb1 = vb1re: vb1max = vb1
vb2 = modul(vb2re, vb2im)
IF vb2 > vb2max THEN
vb2max = vb2
ELSE
END IF
vb3 = modul(vb3re, vb3im)
IF vb3 > vb3max THEN
vb3max = vb3
ELSE
END IF
vb2re = -gb1 * vb2re
vb2im = -gb1 * vb2im
ib1 = modul(vb2re, vb2im)
IF ib1 > ib1max THEN
ib1max = ib1
ELSE
END IF
ia = ga
ib = gb * vb1
iamax = ia
ibmax = ib
ib2re = addre(vb3re, vb3im, -vb2re, -vb2im)

```

```

ib2im = addim(vb3re, vb3im, -vb2re, -vb2im)
ib2re = gb2 * ib2im: ib2im = gb2 * ib2im: ib2 = modul(ib2re, ib2im)
IF ib2 > ib2max THEN
  ib2max = ib2
ELSE
  END IF
ib3 = gb3: ib3max = gb3
'Tb için maximumlar hesaplandı....
'-----Elde edilen tüm gerilim/akımlar dosyaya atılacak-----"
PRINT #20, vb1; vb2; vb3; ib1; ib2; ib3
ELSE
  vare = vire * kaz: vaim = viim * kaz: ia = ga: ib = gb * vare: iamax = ia
  ibmax = ib
  END IF
  PRINT #21, ia; ib
  FOR i = 1 TO nx
    vjre = vare: vjim = vaim
    'H1'in hesabı-----
    nlre = b10(i) - b12(i) * w ^ 2: nlim = b11(i) * w
    dlre = a10(i) - a12(i) * w ^ 2: dlim = a11(i) * w
    hlre = divre(nlre, nlim, dlre, dlim): hlim = divim(nlre, nlim, dlre, dlim)
    'H2'in hesabı-----
    n2re = b20(i) - b22(i) * w ^ 2: n2im = b21(i) * w
    d2re = a20(i) - a22(i) * w ^ 2: d2im = a21(i) * w
    h2re = divre(n2re, n2im, d2re, d2im): h2im = divim(n2re, n2im, d2re, d2im)
    'H41=H42*H21 nin hesabı-----
    'T=H41 olarak alınıyor.
    'tre=H41re tim=H41im dir..
    tre = mulre(hlre, hlim, h2re, h2im)
    tim = mulim(hlre, hlim, h2re, h2im)
    'Quad için h/(1+f*H1*H2) nin hesabı:----
    uore = F(i) * tre: uoim = F(i) * tim: ulre = 1 + uore: ulim = uoim
    uure = divre(h(i), 0, ulre, ulim): uuim = divim(h(i), 0, ulre, ulim)
    'v1=(h/(1+f*H41))*vj=uu*vj-----
    vlre = mulre(uure, uuim, vjre, vjim): vlim = mulim(uure, uuim, vjre, vjim)
    v2re = mulre(vlre, vlim, hlre, hlim): v2im = mulim(vlre, vlim, hlre, hlim)
    'v3=(1/a11*s)*(b10*v1-a10*v2) den hesaplanacak-----
    'yardımcı terimler y1=(b10/a11*s) ve y2=(-a10/a11*s) olsun
    nylre = b10(i): nylim = 0: dylre = 0: dylim = a11(i) * w
    ylre = divre(nylre, nylim, dylre, dylim)
    ylim = divim(nylre, nylim, dylre, dylim): ny2re = -a10(i): ny2im = 0
    dy2re = 0: dy2im = a11(i) * w: y2re = divre(ny2re, ny2im, dy2re, dy2im)
    y2im = divim(ny2re, ny2im, dy2re, dy2im)
    'y3=y1*v1 ve y4=y2*v2 dir. Ayrıca v3=y3+y4 tür.
    y3re = mulre(ylre, ylim, vlre, vlim): y3im = mulim(ylre, ylim, vlre, vlim)
    y4re = mulre(y2re, y2im, v2re, v2im): y4im = mulim(y2re, y2im, v2re, v2im)
    v3re = addre(y3re, y3im, y4re, y4im): v3im = addim(y3re, y3im, y4re, y4im)
    v4re = mulre(v2re, v2im, h2re, h2im): v4im = mulim(v2re, v2im, h2re, h2im)
    'v5=(a11/b12*s)*(v2-v3) den hesaplanacak-----
    'yardımcı terim y1=(a11/b12*s) olsun
    nylre = a11(i): nylim = 0: dylre = 0: dylim = b12(i) * w
    ylre = divre(nylre, nylim, dylre, dylim)
    ylim = divim(nylre, nylim, dylre, dylim)
    'y2=y1*v2 ve y3=y1*v3 ve v5=y2-y3 tür.
    y2re = mulre(ylre, ylim, v2re, v2im)
    y2im = mulim(ylre, ylim, v2re, v2im)
    y3re = mulre(ylre, ylim, v3re, v3im): y3im = mulim(ylre, ylim, v3re, v3im)
    v5re = addre(y2re, y2im, -y3re, -y3im): v5im = addim(y2re, y2im, -y3re, -y3im)
    'v6=(1/a21*s)*(b20*v2-a20*v4) den hesaplanacak-----
    'yardımcı terimler y1=(b20/a21*s) ve y2=(-a20/a21*s) olsun

```

```

nylre = b20(i): nylim = 0: dylre = 0: dylim = a21(i) * w
ylre = divre(nylre, nylim, dylre, dylim)
ylim = divim(nylre, nylim, dylre, dylim)
ny2re = -a20(i): ny2im = 0: dy2re = 0: dy2im = a21(i) * w
y2re = divre(ny2re, ny2im, dy2re, dy2im)
y2im = divim(ny2re, ny2im, dy2re, dy2im)
'y3=y1*v2 ve y4=y2*v4 dir. Ayrıca v6=y3+y4 tür.
y3re = mulre(y1re, ylim, v2re, v2im): y3im = mulim(y1re, ylim, v2re, v2im)
y4re = mulre(y2re, y2im, v4re, v4im): y4im = mulim(y2re, y2im, v4re, v4im)
v6re = addre(y3re, y3im, y4re, y4im): v6im = addim(y3re, y3im, y4re, y4im)
'v7=(a21/b22*s)*(v4-v6) den hesaplanacak-----
'yardımcı terim y1=(a21/b22*s) olsun
nylre = a21(i): nylim = 0: dylre = 0: dylim = b22(i) * w
ylre = divre(nylre, nylim, dylre, dylim)
ylim = divim(nylre, nylim, dylre, dylim)
'y2=y1*v4 ve y3=y1*v6 ve v7=y2-y3 tür.
y2re = mulre(y1re, ylim, v4re, v4im): y2im = mulim(y1re, ylim, v4re, v4im)
y3re = mulre(y1re, ylim, v6re, v6im): y3im = mulim(y1re, ylim, v6re, v6im)
v7re = addre(y2re, y2im, -y3re, -y3im): v7im = addim(y2re, y2im, -y3re, -y3im)
vare = v4re: vaim = v4im
'Tüm gerilim ve akımlar için maximumlar.....
vj = modul(vjre, vjim)
IF vj > vjmax(i) THEN
vjmax(i) = vj
ELSE
END IF
v1 = modul(v1re, v1im)
IF v1 > v1max(i) THEN
v1max(i) = v1
ELSE
END IF
v2 = modul(v2re, v2im)
IF v2 > v2max(i) THEN
v2max(i) = v2
ELSE
END IF
v3 = modul(v3re, v3im)
IF v3 > v3max(i) THEN
v3max(i) = v3
ELSE
END IF
v4 = modul(v4re, v4im)
IF v4 > v4max(i) THEN
v4max(i) = v4
ELSE
END IF
v5 = modul(v5re, v5im)
IF v5 > v5max(i) THEN
v5max(i) = v5
ELSE
END IF
v6 = modul(v6re, v6im)
IF v6 > v6max(i) THEN
v6max(i) = v6
ELSE
END IF
v7 = modul(v7re, v7im)
IF v7 > v7max(i) THEN
v7max(i) = v7
ELSE

```



```

END IF
i1re = -g1(i) * v2re: i1im = -g1(i) * v2im: i1 = modul(i1re, i1im)
IF i1 > i1max(i) THEN
i1max(i) = i1
ELSE
END IF
i2re = addre(v2re, v2im, -v3re, -v3im): i2im = addim(v2re, v2im, -v3re, -v3im)
i2re = g2(i) * i2re: i2im = g2(i) * i2im: i2 = modul(i2re, i2im)
IF i2 > i2max(i) THEN
i2max(i) = i2
ELSE
END IF
i3re = addre(v1re, v1im, -v5re, -v5im): i3im = addim(v1re, v1im, -v5re, -v5im)
i3re = g3(i) * i3re: i3im = g3(i) * i3im: i3 = modul(i3re, i3im)
IF i3 > i3max(i) THEN
i3max(i) = i3
ELSE
END IF
i4re = -g4(i) * v2re: i4im = -g4(i) * v2im: i4 = modul(i4re, i4im)
IF i4 > i4max(i) THEN
i4max(i) = i4
ELSE
END IF
i5re = g5(i) * v1re: i5im = g5(i) * v1im
i5 = modul(i5re, i5im)
IF i5 > i5max(i) THEN
i5max(i) = i5
ELSE
END IF
i6re = -g6(i) * v4re: i6im = -g6(i) * v4im: i6 = modul(i6re, i6im)
IF i6 > i6max(i) THEN
i6max(i) = i6
ELSE
END IF
i7re = addre(v4re, v4im, -v6re, -v6im): i7im = addim(v4re, v4im, -v6re, -v6im)
i7re = g7(i) * i7re: i7im = g7(i) * i7im: i7 = modul(i7re, i7im)
IF i7 > i7max(i) THEN
i7max(i) = i7
ELSE
END IF
i8re = addre(v2re, v2im, -v7re, -v7im): i8im = addim(v2re, v2im, -v7re, -v7im)
i8re = g8(i) * i8re: i8im = g8(i) * i8im: i8 = modul(i8re, i8im)
IF i8 > i8max(i) THEN
i8max(i) = i8
ELSE
END IF
i9re = -g9(i) * v4re: i9im = -g9(i) * v4im: i9 = modul(i9re, i9im)
IF i9 > i9max(i) THEN
i9max(i) = i9
ELSE
END IF
i10re = g10(i) * v2re: i10im = g10(i) * v2im: i10 = modul(i10re, i10im)
IF i10 > i10max(i) THEN
i10max(i) = i10
ELSE
END IF
ixre = gx(i) * vjre: ixim = gx(i) * vjim: ix = modul(ixre, ixim)
IF ix > ixmax(i) THEN
ixmax(i) = ix
ELSE

```

```

END IF
iyre = -gy(i) * vlre: iyim = -gy(i) * vlim: iy = modul(iyre, iyim)
IF iy > iymax(i) THEN
iymax(i) = iy
ELSE
END IF
izre = -gz(i) * v4re: izim = -gz(i) * v4im: iz = modul(izre, izim)
IF iz > izmax(i) THEN
izmax(i) = iz
ELSE
END IF
'TQ/TC için maximumlar hesaplandı.....
'-----Elde edilen tüm gerilim/akımlar dosyaya atılacak-----"
PRINT #i, vj; v1; v2; v3; v4; v5; v6; v7; i1; i2; i3; i4; i5; i6; i7; i8; i9; i10; ix; iy; iz
NEXT
NEXT
CLOSE #20
CLOSE #21
FOR i = 1 TO nx
CLOSE #i
NEXT
'Maxiumlar yazılsın.....
CALL ekran: LOCATE 2, 6: COLOR 14, 1
PRINT "Maksimum gerilim transfer ve transfer admitans fonksiyonu...."
COLOR 15, 1: CALL anykey: CALL ekran: LOCATE 2, 30: COLOR 14, 1
PRINT " Giriş bloğu kazancı...": COLOR 15, 1: LOCATE 5, 2
PRINT "Yamax="; iamax * 1000000!; "[uA/V]": iamax = isat / iamax
IF iamax < vimax THEN
vimax = iamax: domb = 0: dom$ = "Ya"
ELSE
END IF
LOCATE 6, 2: PRINT "Ybmax="; ibmax * 1000000!; "[uA/V]": ibmax = isat / ibmax
IF ibmax < vimax THEN
vimax = ibmax: domb = 0: dom$ = "Yb"
ELSE
END IF
CALL anykey
IF tb$ = "var" THEN
CALL ekran: LOCATE 2, 34: COLOR 14, 1: PRINT "Tb Bloğu...": COLOR 15, 1
LOCATE 5, 2: PRINT "Hb1max="; vb1max: vb1max = vsat / vb1max: vimax = vb1max
domb = 0: dom$ = "Hb1": LOCATE 6, 2: PRINT "Hb2max="; vb2max
vb2max = vsat / vb2max: vimax = vb2max: domb = 0: dom$ = "Hb2": LOCATE 7, 2
PRINT "Hb3max="; vb3max: vb3max = vsat / vb3max
IF vb3max < vimax THEN
vimax = vb3max: domb = 0: dom$ = "Hb3"
ELSE
END IF
LOCATE 5, 30: PRINT "Yb1max="; ib1max * 1000000!; "[uA/V]"
ib1max = isat / ib1max
IF ib1max < vimax THEN
vimax = ib1max: domb = 0: dom$ = "Yb1"
ELSE
END IF
LOCATE 6, 30: PRINT "Yb2max="; ib2max * 1000000!; "[uA/V]"
ib2max = isat / ib2max
IF ib2max < vimax THEN
vimax = ib2max: domb = 0: dom$ = "Yb2"
ELSE
END IF
LOCATE 7, 30: PRINT "Yb3max="; ib3max * 1000000!; "[uA/V]"

```

```

ib3max = isat / ib3max
IF ib3max < vimax THEN
vimax = ib3max: domb = 0: dom$ = "Yb3"
ELSE
END IF
CALL anykey
FOR i = 1 TO nx
CALL ekran: LOCATE 2, 34: COLOR 14, 1: PRINT i; ".Blok": COLOR 15, 1
LOCATE 5, 2: PRINT "Hjmax="; vjmax(i): vjmax(i) = vsat / vjmax(i)
IF vjmax(i) < vimax THEN
vimax = vjmax(i): domb = i: dom$ = "Hj"
ELSE
END IF
LOCATE 6, 2: PRINT "H1max="; vlmax(i): vlmax(i) = vsat / vlmax(i)
IF vlmax(i) < vimax THEN
vimax = vlmax(i): domb = i: dom$ = "H1"
ELSE
END IF
LOCATE 7, 2: PRINT "H2max="; v2max(i): v2max(i) = vsat / v2max(i)
IF v2max(i) < vimax THEN
vimax = v2max(i): domb = i: dom$ = "H2"
ELSE
END IF
LOCATE 8, 2: PRINT "H3max="; v3max(i): v3max(i) = vsat / v3max(i)
IF v3max(i) < vimax THEN
vimax = v3max(i): domb = i: dom$ = "H3"
ELSE
END IF
LOCATE 9, 2: PRINT "H4max="; v4max(i): v4max(i) = vsat / v4max(i)
IF v4max(i) < vimax THEN
vimax = v4max(i): domb = i: dom$ = "H4"
ELSE
END IF
LOCATE 10, 2: PRINT "H5max="; v5max(i): v5max(i) = vsat / v5max(i)
IF v5max(i) < vimax THEN
vimax = v5max(i): domb = i: dom$ = "H5"
ELSE
END IF
LOCATE 11, 2: PRINT "H6max="; v6max(i): v6max(i) = vsat / v6max(i)
IF v6max(i) < vimax THEN
vimax = v6max(i): domb = i: dom$ = "H6"
ELSE
END IF
LOCATE 12, 2: PRINT "H7max="; v7max(i): v7max(i) = vsat / v7max(i)
IF v7max(i) < vimax THEN
vimax = v7max(i): domb = i: dom$ = "H7"
ELSE
END IF
LOCATE 5, 30: PRINT "Y1max="; ilmax(i) * 1000000!; "[uA/V]"
ilmax(i) = isat / ilmax(i)
IF ilmax(i) < vimax THEN
vimax = ilmax(i): domb = i: dom$ = "Y1"
ELSE
END IF
LOCATE 6, 30: PRINT "Y2max="; i2max(i) * 1000000!; "[uA/V]"
i2max(i) = isat / i2max(i)
IF i2max(i) < vimax THEN
vimax = i2max(i): domb = i: dom$ = "Y2"
ELSE
END IF

```

```

LOCATE 7, 30: PRINT "Y3max="; i3max(i) * 1000000!; "[uA/V]"
i3max(i) = isat / i3max(i)
IF i3max(i) < vimax THEN
vimax = i3max(i): domb = i: dom$ = "Y3"
ELSE
END IF
LOCATE 8, 30: PRINT "Y4max="; i4max(i) * 1000000!; "[uA/V]"
i4max(i) = isat / i4max(i)
IF i4max(i) < vimax THEN
vimax = i4max(i): domb = i: dom$ = "Y4"
ELSE
END IF
LOCATE 9, 30: PRINT "Y5max="; i5max(i) * 1000000!; "[uA/V]"
i5max(i) = isat / i5max(i)
IF i5max(i) < vimax THEN
vimax = i5max(i): domb = i: dom$ = "Y5"
ELSE
END IF
LOCATE 10, 30: PRINT "Y6max="; i6max(i) * 1000000!; "[uA/V]"
i6max(i) = isat / i6max(i)
IF i6max(i) < vimax THEN
vimax = i6max(i): domb = i: dom$ = "Y6"
ELSE
END IF
LOCATE 11, 30: PRINT "Y7max="; i7max(i) * 1000000!; "[uA/V]"
i7max(i) = isat / i7max(i)
IF i7max(i) < vimax THEN
vimax = i7max(i): domb = i: dom$ = "Y7"
ELSE
END IF
LOCATE 12, 30: PRINT "Y8max="; i8max(i) * 1000000!; "[uA/V]"
i8max(i) = isat / i8max(i)
IF i8max(i) < vimax THEN
vimax = i8max(i): domb = i: dom$ = "Y8"
ELSE
END IF
LOCATE 13, 30: PRINT "Y9max="; i9max(i) * 1000000!; "[uA/V]"
i9max(i) = isat / i9max(i)
IF i9max(i) < vimax THEN
vimax = i9max(i): domb = i: dom$ = "Y9"
ELSE
END IF
LOCATE 14, 30: PRINT "Y10max="; i10max(i) * 1000000!; "[uA/V]"
i10max(i) = isat / i10max(i)
IF i10max(i) < vimax THEN
vimax = i10max(i): domb = i: dom$ = "Y10"
ELSE
END IF
LOCATE 15, 30: PRINT "Yxmax="; ixmax(i) * 1000000!; "[uA/V]"
ixmax(i) = isat / ixmax(i)
IF ixmax(i) < vimax THEN
vimax = ixmax(i): domb = i: dom$ = "Yx"
ELSE
END IF
LOCATE 16, 30: PRINT "Yymax="; iymax(i) * 1000000!; "[uA/V]"
iymax(i) = isat / iymax(i)
IF iymax(i) < vimax THEN
vimax = iymax(i): domb = i: dom$ = "Yy"
ELSE
END IF

```

```

IF kq$ = "quad" THEN
LOCATE 17, 30: PRINT "Yzmax="; izmax(i) * 1000000!; "[uA/V]"
izmax(i) = isat / izmax(i)
IF izmax(i) < vimax THEN
vimax = izmax(i): domb = i: dom$ = "Yz"
ELSE
END IF
ELSE
END IF
CALL anykey
NEXT
ELSE
FOR i = 1 TO nx
CALL ekran: LOCATE 2, 36: COLOR 14, 1: PRINT i; ".Blok": COLOR 15, 1
LOCATE 5, 2: PRINT "Hjmax="; vjmax(i): vjmax(i) = vsat / vjmax(i)
IF vjmax(i) < vimax THEN
vimax = vjmax(i): domb = i: dom$ = "Hj"
ELSE
END IF
LOCATE 6, 2
PRINT "H1max="; v1max(i)
v1max(i) = vsat / v1max(i)
IF v1max(i) < vimax THEN
vimax = v1max(i): domb = i: dom$ = "H1"
ELSE
END IF
LOCATE 7, 2: PRINT "H2max="; v2max(i): v2max(i) = vsat / v2max(i)
IF v2max(i) < vimax THEN
vimax = v2max(i): domb = i: dom$ = "H2"
ELSE
END IF
LOCATE 8, 2: PRINT "H3max="; v3max(i): v3max(i) = vsat / v3max(i)
IF v3max(i) < vimax THEN
vimax = v3max(i): domb = i: dom$ = "H3"
ELSE
END IF
LOCATE 9, 2: PRINT "H4max="; v4max(i): v4max(i) = vsat / v4max(i)
IF v4max(i) < vimax THEN
vimax = v4max(i): domb = i: dom$ = "H4"
ELSE
END IF
LOCATE 10, 2: PRINT "H5max="; v5max(i): v5max(i) = vsat / v5max(i)
IF v5max(i) < vimax THEN
vimax = v5max(i): domb = i: dom$ = "H5"
ELSE
END IF
LOCATE 11, 2: PRINT "H6max="; v6max(i): v6max(i) = vsat / v6max(i)
IF v6max(i) < vimax THEN
vimax = v6max(i): domb = i: dom$ = "H6"
ELSE
END IF
LOCATE 12, 2: PRINT "H7max="; v7max(i): v7max(i) = vsat / v7max(i)
IF v7max(i) < vimax THEN
vimax = v7max(i): domb = i: dom$ = "H7"
ELSE
END IF
LOCATE 5, 30: PRINT "Y1max="; ilmax(i) * 1000000!; "[uA/V]"
ilmax(i) = isat / ilmax(i)
IF ilmax(i) < vimax THEN
vimax = ilmax(i): domb = i: dom$ = "Y1"

```

```

ELSE
END IF
LOCATE 6, 30: PRINT "Y2max=", i2max(i) * 1000000!; "[uA/V]"
i2max(i) = isat / i2max(i)
IF i2max(i) < vimax THEN
vimax = i2max(i): domb = i: dom$ = "Y2"
ELSE
END IF
LOCATE 7, 30: PRINT "Y3max=", i3max(i) * 1000000!; "[uA/V]"
i3max(i) = isat / i3max(i)
IF i3max(i) < vimax THEN
vimax = i3max(i): domb = i: dom$ = "Y3"
ELSE
END IF
LOCATE 8, 30: PRINT "Y4max=", i4max(i) * 1000000!; "[uA/V]"
i4max(i) = isat / i4max(i)
IF i4max(i) < vimax THEN
vimax = i4max(i): domb = i: dom$ = "Y4"
ELSE
END IF
LOCATE 9, 30: PRINT "Y5max=", i5max(i) * 1000000!; "[uA/V]"
i5max(i) = isat / i5max(i)
IF i5max(i) < vimax THEN
vimax = i5max(i): domb = i: dom$ = "Y5"
ELSE
END IF
LOCATE 10, 30: PRINT "Y6max=", i6max(i) * 1000000!; "[uA/V]"
i6max(i) = isat / i6max(i)
IF i6max(i) < vimax THEN
vimax = i6max(i): domb = i: dom$ = "Y6"
ELSE
END IF
LOCATE 11, 30: PRINT "Y7max=", i7max(i) * 1000000!; "[uA/V]"
i7max(i) = isat / i7max(i)
IF i7max(i) < vimax THEN
vimax = i7max(i): domb = i: dom$ = "Y7"
ELSE
END IF
LOCATE 12, 30: PRINT "Y8max=", i8max(i) * 1000000!; "[uA/V]"
i8max(i) = isat / i8max(i)
IF i8max(i) < vimax THEN
vimax = i8max(i): domb = i: dom$ = "Y8"
ELSE
END IF
LOCATE 13, 30
PRINT "Y9max=", i9max(i) * 1000000!; "[uA/V]": i9max(i) = isat / i9max(i)
IF i9max(i) < vimax THEN
vimax = i9max(i): domb = i: dom$ = "Y9"
ELSE
END IF
LOCATE 14, 30: PRINT "Y10max=", i10max(i) * 1000000!; "[uA/V]"
i10max(i) = isat / i10max(i)
IF i10max(i) < vimax THEN
vimax = i10max(i): domb = i: dom$ = "Y10"
ELSE
END IF
LOCATE 15, 30: PRINT "Yxmax=", ixmax(i) * 1000000!; "[uA/V]"
ixmax(i) = isat / ixmax(i)
IF ixmax(i) < vimax THEN
vimax = ixmax(i): domb = i: dom$ = "Yx"

```

```

ELSE
END IF
LOCATE 16, 30: PRINT "Yymax="; iymax(i) * 1000000!; "[uA/V]"
iymax(i) = isat / iymax(i)
IF iymax(i) < vimax THEN
vimax = iymax(i): domb = i: dom$ = "Yy"
ELSE
END IF
IF kq$ = "quad" THEN
LOCATE 17, 30: PRINT "Yzmax="; izmax(i) * 1000000!; "[uA/V]"
izmax(i) = isat / izmax(i)
IF izmax(i) < vimax THEN
vimax = izmax(i): domb = i: dom$ = "Yz"
ELSE
END IF
ELSE
END IF
CALL anykey
NEXT
END IF
'Maximumlar yazıldı-----
'Vimax hesaplandı-----
CALL ekran: LOCATE 2, 6: COLOR 14, 1
PRINT "    Filtre girişine uygulanabilecek maksimum giriş gerilimi..."
COLOR 15, 1: LOCATE 4, 6: PRINT "": LOCATE 5, 6: PRINT "": LOCATE 6, 6
PRINT "": LOCATE 8, 6: PRINT "Vimax="; vimax * 1000; "mV"
IF domb = 0 THEN
LOCATE 10, 6: PRINT "Dominant Blok ..."; "Tb": LOCATE 11, 6
PRINT "Dominant H/Y ..."; " "; dom$
ELSE
LOCATE 10, 6: PRINT "Dominant Blok ..."; domb: LOCATE 11, 6
PRINT "Dominant H/Y ..."; " "; dom$
END IF
CALL anykey
'-----
'grafik için v/i (gerilim/akım) secimi
goster1:
'Elde edilen gerilim ve akımlardan hangisini istiyoruz
CALL ekran: COLOR 14, 1: LOCATE 2, 2
PRINT "          Blok seçimini yapınız": COLOR 15, 1
LOCATE 4, 2
IF tb$ = "var" THEN
LOCATE 5, 2: PRINT "[ 20..Giriş kazanç bloğu için ]": LOCATE 6, 2
PRINT "[ 0 .. Tb Bloğu için ]"
FOR i = 1 TO nx
LOCATE i + 6, 2
PRINT "[ "; i; "..."; i; ".Blok için ]"
NEXT
ELSE
LOCATE 5, 2: PRINT "[ 20..Giriş kazanç bloğu için ]"
FOR i = 1 TO nx
LOCATE i + 5, 2: PRINT "[ "; i; "..."; i; ".Blok için ]"
NEXT
END IF
LOCATE 16, 2: INPUT "[ Seçiminiz... ] :> ", qx: CALL ekran: COLOR 14, 1
LOCATE 2, 20: PRINT "    H/Y seçimini yapınız": COLOR 15, 1: LOCATE 4, 2
IF qx = 20 THEN
LOCATE 5, 2: PRINT "[ 1 ... Ya ]": LOCATE 6, 2: PRINT "[ 2 ... Yb ]"
ELSEIF qx = 0 THEN
LOCATE 5, 2: PRINT "[ 1 ... Hb1 ]": LOCATE 6, 2: PRINT "[ 2 ... Hb2 ]"

```

```

LOCATE 7, 2: PRINT "[ 3 ... Hb3 ]": LOCATE 5, 30: PRINT "[ 4 ... Yb1 ]"
LOCATE 6, 30: PRINT "[ 5 ... Yb2 ]": LOCATE 7, 30: PRINT "[ 6 ... Yb3 ]"
ELSEIF qx <> 0 AND qx <> 20 THEN
LOCATE 5, 2: PRINT "[ 1 ... Hj ]": LOCATE 6, 2: PRINT "[ 2 ... H1 ]"
LOCATE 7, 2: PRINT "[ 3 ... H2 ]": LOCATE 8, 2: PRINT "[ 4 ... H3 ]"
IF qx = nx THEN
LOCATE 9, 2: PRINT "[ 5 ... H4 ]": LOCATE 9, 17: PRINT "*": LOCATE 19, 2
PRINT "* Toplam filtrenin frekans cevabı"
ELSE
LOCATE 9, 2: PRINT "[ 5 ... H4 ]"
END IF
LOCATE 10, 2: PRINT "[ 6 ... H5 ]": LOCATE 11, 2: PRINT "[ 7 ... H6 ]"
LOCATE 12, 2: PRINT "[ 8 ... H7 ]"
LOCATE 5, 30: PRINT "[ 9 ... Y1 ]": LOCATE 6, 30: PRINT "[ 10... Y2 ]"
LOCATE 7, 30: PRINT "[ 11... Y3 ]": LOCATE 8, 30: PRINT "[ 12... Y4 ]"
LOCATE 9, 30: PRINT "[ 13... Y5 ]": LOCATE 10, 30: PRINT "[ 14... Y6 ]"
LOCATE 11, 30: PRINT "[ 15... Y7 ]": LOCATE 12, 30: PRINT "[ 16... Y8 ]"
LOCATE 13, 30: PRINT "[ 17... Y9 ]": LOCATE 14, 30: PRINT "[ 18... Y10 ]"
LOCATE 15, 30: PRINT "[ 19... Yx ]": LOCATE 16, 30: PRINT "[ 20... Yy ]"
IF kq$ = "quad" THEN
LOCATE 17, 30: PRINT "[ 21... Yz ]"
ELSE
END IF
ELSE
END IF
LOCATE 22, 2: INPUT " [ Seçiminiz...] :> ", hy
OPEN "tb.dat" FOR INPUT AS #20
OPEN "kaz.dat" FOR INPUT AS #21
FOR i = 1 TO nx
d$ = "q" + CHR$(48 + i) + ".dat"
OPEN d$ FOR INPUT AS #i
NEXT
maxgenlik = 0
FOR K = 1 TO S
IF qx = 20 THEN
IF hy = 1 THEN
INPUT #21, ia, ib: genlik(K) = ia
ELSEIF hy = 2 THEN
INPUT #21, ia, ib: genlik(K) = ib
ELSE
END IF
'Tb için
ELSEIF qx = 0 THEN
IF hy = 1 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = vb1
ELSEIF hy = 2 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = vb2
ELSEIF hy = 3 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = vb3
'admitanslar.....
ELSEIF hy = 4 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = ib1
ELSEIF hy = 5 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = ib2
ELSEIF hy = 6 THEN
INPUT #20, vb1, vb2, vb3, ib1, ib2, ib3: genlik(K) = ib3
ELSE
END IF
ELSEIF qx <> 0 AND qx <> 20 THEN
IF hy = 1 THEN

```



```

INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = vj
ELSEIF hy = 2 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = v1
ELSEIF hy = 3 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = v2
ELSEIF hy = 4 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = v3
ELSEIF hy = 5 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = v4
ELSEIF hy = 6 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = v5
ELSEIF hy = 7 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = v6
ELSEIF hy = 8 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = v7
'Admitanslar.....
ELSEIF hy = 9 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i1
ELSEIF hy = 10 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i2
ELSEIF hy = 11 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i3
ELSEIF hy = 12 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i4
ELSEIF hy = 13 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i5
ELSEIF hy = 14 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i6
ELSEIF hy = 15 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i7
ELSEIF hy = 16 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i8
ELSEIF hy = 17 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i9
ELSEIF hy = 18 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = i10
ELSEIF hy = 19 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = ix
ELSEIF hy = 20 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = iy

```

```

ELSEIF kq$ = "quad" AND hy = 21 THEN
INPUT #qx, vj, v1, v2, v3, v4, v5, v6, v7, i1, i2, i3, i4, i5, i6, i7, i8, i9, i10, ix, iy, iz
genlik(K) = iz
ELSE
END IF
ELSE
END IF
IF genlik(K) > maxgenlik THEN
maxgenlik = genlik(K):mfr = K
ELSE
END IF
NEXT
CLOSE #20
CLOSE #21
FOR i = 1 TO nx
CLOSE #i
NEXT
END SUB

```

-----FUNCTION Divim-----

```

FUNCTION divim (a1, a2, b1, b2)
'iki karmaşık sayının bölümünün sanal kısmı
divim = mulim(a1, a2, b1, -b2) / mulre(b1, b2, b1, -b2)
END FUNCTION

```

-----FUNCTION Divre-----

```

FUNCTION divre (a1, a2, b1, b2)
'iki karmaşık sayının bölümünün reel kısmı
divre = mulre(a1, a2, b1, -b2) / mulre(b1, b2, b1, -b2)
END FUNCTION

```

-----EKTRAN Alt Programı-----

```

SUB ekran
CLS: VIEW (0, 0)-(639, 349): WINDOW (0, 0)-(639, 349)
LINE (1, 1)-(639, 349), 15, B: LINE (5, 346)-(635, 316), 14, B: COLOR 15, 1
END SUB

```

-----ELEMEN Alt Programı-----

```

SUB eleman
'Eleman hesabı-----
G = .0001: gb = G: ga = gb * kaz
'AG filtre için Tq bloğu eleman değerleri
IF ta$ = "var" THEN
h = 1 / so: ee = 1 / wp2
IF tb$ = "var" THEN
IF yt$ = "bw" OR yt$ = "chy" THEN
'bw ve chy AG filtre için Tb bloğu eleman değerleri:-----
gb3 = G: gb2 = G: cb1 = (gb3 * ab1) / bb0: gb1 = (cb1 * ab0) / ab1
cb2 = gb2 / ab1
ELSE
'elliptik AG filtre için Tb bloğu eleman değerleri
gb4 = G: gb3 = bb2 * gb4: gb2 = G: gb5 = G: cb2 = gb2 * bb2 / ab1
cb1 = gb5 * ab1 / bb0: gb1 = cb1 * ab0 / ab1
END IF
'quad/kaskad blok için eleman değerleri-----
FOR i = 1 TO ck
gy(i) = G: gx(i) = gy(i) * h(i): gz(i) = F(i) * gy(i): gz(i) = ABS(gz(i))
IF yt$ = "ell" THEN
'ell AG quad/kaskad blokları eleman değerleri
g4(i) = G: g3(i) = b12(i) * g4(i): g2(i) = G
c2(i) = (g2(i) * b12(i)) / a11(i): g5(i) = G
c1(i) = (g5(i) * a11(i)) / b10(i): g1(i) = (c1(i) * a10(i)) / a11(i)
g9(i) = G: g8(i) = b22(i) * g9(i): g7(i) = G
c4(i) = (g7(i) * b22(i)) / a21(i): g10(i) = G

```

```

c3(i) = (g10(i) * a21(i)) / b20(i); g6(i) = (c3(i) * a20(i)) / a21(i)
ELSE
'bw ve chy AG filtre için quad blokları eleman değerleri
g3(i) = G: c2(i) = g3(i) / a11(i); g2(i) = G
c1(i) = (g3(i) * a11(i)) / b10(i); g1(i) = (c1(i) * a10(i)) / a11(i)
g6(i) = G: g5(i) = G: c4(i) = g5(i) / a21(i)
c3(i) = (g6(i) * a21(i)) / b20(i); g4(i) = (c1(i) * a20(i)) / a21(i)
END IF
NEXT
ELSE
'quad/kaskad blok için eleman değerleri-----
FOR i = 1 TO ck
gy(i) = G: gx(i) = gy(i) * h(i): gz(i) = F(i) * gy(i): gz(i) = ABS(gz(i))
IF yt$ = "ell" THEN
'ell AG için quad/kaskad blokları eleman değerleri
g4(i) = G: g3(i) = b12(i) * g4(i): g2(i) = G
c2(i) = (g2(i) * b12(i)) / a11(i); g5(i) = G
c1(i) = (g5(i) * a11(i)) / b10(i); g1(i) = (c1(i) * a10(i)) / a11(i)
g9(i) = G: g8(i) = b22(i) * g9(i): g7(i) = G
c4(i) = (g7(i) * b22(i)) / a21(i); g10(i) = G
c3(i) = (g10(i) * a21(i)) / b20(i); g6(i) = (c3(i) * a20(i)) / a21(i)
ELSE
'bw ve chy AG filtre için quad/kaskad blokları eleman değerleri
g3(i) = G: c2(i) = g3(i) / a11(i); g2(i) = G
c1(i) = (g3(i) * a11(i)) / b10(i); g1(i) = (c1(i) * a10(i)) / a11(i)
g6(i) = G: g5(i) = G: c4(i) = g5(i) / a21(i)
c3(i) = (g6(i) * a21(i)) / b20(i); g4(i) = (c1(i) * a20(i)) / a21(i)
END IF
NEXT
END IF
ELSE
IF tb$ = "var" THEN
IF ft$ = "AG" THEN
IF yt$ = "bw" OR yt$ = "chy" THEN
'bw ve chy AG filtre için Tb bloğu eleman değerleri
gb3 = G: gb2 = G: cb1 = (gb3 * ab1) / bb0: gb1 = (cb1 * ab0) / ab1
cb2 = gb2 / ab1
ELSE
'elliptik AG filtre için Tb bloğu eleman değerleri
gb4 = G: gb3 = bb2 * gb4: gb2 = G: gb5 = G: cb2 = gb2 * bb2 / ab1
cb1 = gb5 * ab1 / bb0: gb1 = cb1 * ab0 / ab1
END IF
ELSEIF ft$ = "BG" THEN
'BG filtre için Tb bloğu eleman değerleri
gb3 = G: cb2 = gb3 / bb1: gb2 = cb2 * ab1: gb1 = G: cb1 = (gb1 * ab1) / ab0
END IF
FOR i = 1 TO ck
gy(i) = G: gx(i) = gy(i) * h(i): gz(i) = F(i) * gy(i): gz(i) = ABS(gz(i))
IF yt$ = "ell" THEN
'ell AG ve ell BG için quad/kaskad blokları eleman değerleri
g4(i) = G: g3(i) = b12(i) * g4(i): g2(i) = G
c2(i) = (g2(i) * b12(i)) / a11(i); g5(i) = G
c1(i) = (g5(i) * a11(i)) / b10(i); g1(i) = (c1(i) * a10(i)) / a11(i)
g9(i) = G: g8(i) = b22(i) * g9(i)
g7(i) = G: c4(i) = (g7(i) * b22(i)) / a21(i); g10(i) = G
c3(i) = (g10(i) * a21(i)) / b20(i); g6(i) = (c3(i) * a20(i)) / a21(i)
ELSE
IF ft$ = "BG" THEN
'bw ve chy BG filtre için quad/kaskad blokları eleman değerleri
g3(i) = G: c2(i) = g3(i) / b11(i); g2(i) = c2(i) * a11(i); g1(i) = G

```

```

c1(i) = (g1(i) * a1(i)) / a10(i): g6(i) = G: c4(i) = g6(i) / b21(i)
g5(i) = c4(i) * a21(i): g4(i) = G: c3(i) = (g4(i) * a21(i)) / a20(i)
ELSE
'bw ve chy AG filtre için quad/kaskad blokları eleman değerleri
g3(i) = G: c2(i) = g3(i) / a11(i): g2(i) = G
c1(i) = (g3(i) * a11(i)) / b10(i): g1(i) = (c1(i) * a10(i)) / a11(i)
g6(i) = G: g5(i) = G: c4(i) = g5(i) / a21(i)
c3(i) = (g6(i) * a21(i)) / b20(i): g4(i) = (c1(i) * a20(i)) / a21(i)
END IF
END IF
NEXT
ELSEIF tb$ = "yok" THEN
FOR i = 1 TO ck
gy(i) = G: gx(i) = gy(i) * h(i): gz(i) = F(i) * gy(i): gz(i) = ABS(gz(i))
IF yt$ = "ell" THEN
'ell AG ve ell BG için quad/kaskad blokları eleman değerleri
g4(i) = G: g3(i) = b12(i) * g4(i): g2(i) = G
c2(i) = (g2(i) * b12(i)) / a11(i): g5(i) = G
c1(i) = (g5(i) * a11(i)) / b10(i): g1(i) = (c1(i) * a10(i)) / a11(i)
g9(i) = G: g8(i) = b22(i) * g9(i): g7(i) = G
c4(i) = (g7(i) * b22(i)) / a21(i)
g10(i) = G: c3(i) = (g10(i) * a21(i)) / b20(i)
g6(i) = (c3(i) * a20(i)) / a21(i)
ELSE
IF ft$ = "BG" THEN
'bw ve chy BG filtre için quad/kaskad blokları eleman değerleri
g3(i) = G: c2(i) = g3(i) / b11(i): g2(i) = c2(i) * a11(i): g1(i) = G
c1(i) = (g1(i) * a11(i)) / a10(i): g6(i) = G: c4(i) = g6(i) / b21(i)
g5(i) = c4(i) * a21(i): g4(i) = G: c3(i) = (g4(i) * a21(i)) / a20(i)
ELSE
'bw ve chy AG filtre için quad/kaskad blokları eleman değerleri
g3(i) = G: c2(i) = g3(i) / a11(i): g2(i) = G
c1(i) = (g3(i) * a11(i)) / b10(i): g1(i) = (c1(i) * a10(i)) / a11(i)
g6(i) = G: g5(i) = G: c4(i) = g5(i) / a21(i)
c3(i) = (g6(i) * a21(i)) / b20(i)
g4(i) = (c1(i) * a20(i)) / a21(i)
END IF
END IF
NEXT
END IF
END IF
END SUB

```

'-----ELL Alt Programı-----'

```

--
SUB ell (oma, Ap, Aa)
DIM ccc(40), aaa(40), bbb(40), X(40), Y(40)
K = 1 / oma: u = (1 - SQR(SQR(1 - K ^ 2))) / (2 * (1 + SQR(SQR(1 - K ^ 2))))
q = u + 2 * u ^ 5 + 15 * u ^ 9 + 150 * u ^ 13
d = (10 ^ (Aa / 10) - 1) / (10 ^ (Ap / 10) - 1)
N = (LOG(16 * d)) / (LOG(1 / q)): N = FIX(N) + 1
'n tek mi çift mi.....
IF N MOD 2 = 0 THEN NS = "even" ELSE NS = "odd"
'actual Aa.....
Aa = (10 * LOG(1 + ((10 ^ (Ap / 10) - 1) / (16 * q ^ N))) / LOG(10))
V = (1 / (2 * N)) * LOG((10 ^ (.05 * Ap) + 1) / (10 ^ (.05 * Ap) - 1))
'po 'in hesabı.....
pay = 0: payda = 0
FOR m1 = 0 TO 4
G = ((-1) ^ m1) * (q ^ (m1 * (m1 + 1))) * .5
TR = G * (EXP((2 * m1 + 1) * V) - EXP(-(2 * m1 + 1) * V))

```

```

pay = pay + TR
NEXT
FOR m1 = 1 TO 4
sr = ((-1) ^ m1) * (q ^ (m1 ^ 2)) * .5 * (EXP(2 * m1 * V) + EXP(-2 * m1 * V))
payda = payda + sr
NEXT
po = ABS(((2 * (q ^ (.25))) * pay) / (1 + 2 * payda))
ww = SQR((1 + (po ^ 2) / K) * (1 + K * po ^ 2))
m = N - N MOD 2: r = m / 2
pi = 3.141592654#
FOR i = 1 TO r
IF N$ = "odd" THEN mu = i ELSE mu = i - .5
pay = 0: payda = 0
FOR m1 = 0 TO 4
zf = ((-1) ^ m1) * (q ^ (m1 * (m1 + 1))) * SIN((2 * m1 + 1) * mu * pi / N)
pay = pay + zf: NEXT
FOR m1 = 1 TO 4
payda = payda + ((-1) ^ m1) * (q ^ (m1 ^ 2)) * COS(2 * m1 * mu * pi / N)
NEXT
X(i) = (2 * q ^ .25) * pay / (1 + 2 * payda)
Y(i) = SQR((1 - X(i) ^ 2 / K) * (1 - K * X(i) ^ 2))
aaa(i) = 1 / X(i) ^ 2
bbb(i) = 2 * po * Y(i) / (1 + po ^ 2 * X(i) ^ 2)
ccc(i) = ((po * Y(i)) ^ 2 + (X(i) * ww) ^ 2) / (1 + po ^ 2 * X(i) ^ 2) ^ 2
NEXT
'rescale
term = 1
FOR i = 1 TO r
term = term * ccc(i) / aaa(i): NEXT
IF N$ = "odd" THEN
term = term * po
ELSE
term = term * 10 ^ (-Ap / 20)
END IF
ho = term: alfa = SQR(oma): alfasqrd = alfa ^ 2
IF N$ = "odd" THEN
ho = ho * alfa: po = po * alfa
ELSE
END IF
FOR i = 1 TO r
aaa(i) = aaa(i) * alfasqrd: ccc(i) = ccc(i) * alfasqrd
bbb(i) = bbb(i) * alfa: NEXT
'kutup ve sıfırların bulunması
FOR i = 1 TO r
pre(i) = -.5 * bbb(i): pre(N + 1 - i) = pre(i)
pim(i) = .5 * SQR(4 * ccc(i) - bbb(i) ^ 2): pim(N + 1 - i) = -pim(i)
zim(i) = SQR(aaa(i)): zim(2 * r + 1 - i) = -zim(i)
NEXT
IF N$ = "odd" THEN
pre(r + 1) = -po: pim(r + 1) = 0
ELSE
END IF
po = -po: CALL prototip
END SUB
'-----FRAME Alt Programı-----
SUB frame
CLS: LINE (1, 1)-(639, 349), 15, B
END SUB
'-----FUNCTION geribes-----
FUNCTION geribes (kk, ll, mm)-

```

```
IF yt$ = "bw" OR yt$ = "chy" THEN
```

```
geribes = ll - (kk ^ 2) / 4
```

```
ELSE
```

```
geribes = (ll + mm - SQR((ll - mm) ^ 2 + mm * kk ^ 2)) / (2 * mm)
```

```
END IF
```

```
END FUNCTION
```

```
'-----GRAFIK Alt Programı-----
```

```
SUB grafik
```

```
CLS: SCREEN 9: COLOR 14, 1: CALL frame
```

```
maxx = 640: maxy = 350: x1 = 100: y1 = 20: x2 = 600: y2 = 250: mx = x2 - x1: my = y2 - y1
```

```
VIEW (x1, y1)-(x2, y2): WINDOW (0, 0)-(mx + 20, my + 20): LINE (580, 0)-(0, 0), 14
```

```
LINE (0, 0)-(0, 230), 14
```

```
FOR Y = 20 TO 220 STEP 20
```

```
FOR X = 1 TO 500 STEP 6
```

```
PSET (X, Y), 15
```

```
NEXT
```

```
NEXT
```

```
FOR X = 50 TO 500 STEP 50
```

```
FOR Y = 1 TO 220 STEP 6
```

```
PSET (X, Y), 15
```

```
NEXT:NEXT: LOCATE 2, 9: PRINT " Gain,dB": LOCATE 23, 40: PRINT "Frekans,"; unit$; "Hz"
```

```
LOCATE 23, 76: PRINT "wait"
```

```
genlikayar = 180 / maxgenlik: LINE (1, 0)-(1, genlikayar * genlik(1)), 1
```

```
FOR K = 80 TO S
```

```
LINE -(K - 80, 4.86 * (20 * LOG(genlikayar * ABS(genlik(K)) + 1) / LOG(10)) + 1), 15
```

```
NEXT
```

```
PSET (0, 4.86 * (20 * LOG(genlikayar * ABS(genlik(80)) + 1) / LOG(10)) + 1), 12
```

```
mindb = 20 * LOG(ABS(genlik(80))) / LOG(10): LOCATE 16, 2: PRINT mindb
```

```
maxdb = 20 * LOG(ABS(maxgenlik)) / LOG(10): LOCATE 4, 2: PRINT maxdb
```

```
k80 = (.2 * wa1 + ((maxfr - minfr) / S) * 79) / 2 / pi
```

```
IF unit$ = "K" THEN
```

```
bol = 1000
```

```
ELSEIF unit$ = "M" THEN
```

```
bol = 1000000
```

```
ELSE
```

```
bol = 1
```

```
END IF
```

```
VIEW: WINDOW: LINE (100, 250)-(120, 275), 12
```

```
LINE (580, 250)-(550, 275), 12: LOCATE 21, 12: PRINT k80 / bol; unit$; "Hz"
```

```
LOCATE 21, 65: PRINT 1.2 * fa2; unit$; "Hz"
```

```
VIEW: WINDOW
```

```
END SUB
```

```
'-----KASKAD Alt Programı-----
```

```
SUB kaskad
```

```
'Tc1(s)=(b12s2+b11s+b10)/(a12s2+a11s+a10)
```

```
'Tc2(s)=(b22s2+b21s+b20)/(a22s2+a21s+a20)
```

```
DIM xx(40), yy(40), er(40)
```

```
FOR i = 1 TO ck
```

```
xx(i) = 2 * SQR(a0(i)) + a2(i)
```

```
yy(i) = SQR(xx(i) ^ 2 - 4 * SQR(a0(i)) * a3(i) ^ 2)
```

```
er(i) = (SQR((xx(i) + yy(i)) / 2)
```

```
wp1(i) = er(i) + SQR((xx(i) + yy(i)) / 2 - 4 * SQR(a0(i)))) / 2
```

```
wp2(i) = SQR(a0(i)) / wp1(i)
```

```
q1(i) = (wp1(i) + wp2(i)) / a3(i): q2(i) = q1(i)
```

```
a12(i) = 1: a11(i) = wp1(i) / q1(i): a10(i) = wp1(i) ^ 2
```

```
a22(i) = 1: a21(i) = wp2(i) / q2(i): a20(i) = wp2(i) ^ 2
```

```
NEXT
```

```
IF yt$ = "bw" OR yt$ = "chy" THEN
```

```
SELECT CASE ck
```

```
CASE nx
```

```

FOR i = 1 TO nx
b12(i) = 0: b11(i) = BAND: b10(i) = 0: b22(i) = 0: b21(i) = BAND: b20(i) = 0
NEXT
CASE tx
FOR i = 1 TO tx
b12(i) = 0: b11(i) = 0: b10(i) = (wp2) ^ 2
b22(i) = 0: b21(i) = 0: b20(i) = (wp2) ^ 2: NEXT: END SELECT
ELSEIF yt$ = "ell" THEN
FOR i = 1 TO ck
xx(i) = b2(i) - 2 * SQR(b0(i)): yy(i) = b2(i) + 2 * SQR(b0(i))
wz1(i) = (SQR(yy(i)) + SQR(xx(i))) / 2: wz2(i) = SQR(b0(i)) / wz1(i)
b12(i) = 1: b11(i) = 0: b10(i) = wz1(i) ^ 2
b22(i) = 1: b21(i) = 0: b20(i) = wz2(i) ^ 2: NEXT
END IF
END SUB

```

-----KATSAY Alt Programı-----

```

SUB katsay
'lp prototip fonksiyonuna ait kutup ve sıfırlardan yararlanarak
'a,b,c katsayılarının bulunması
FOR i = 1 TO nx
L = pole(i): a(i) = -2 * pre(L): B(i) = modul(pre(L), pim(L)) ^ 2: NEXT
IF yt$ = "ell" THEN
FOR i = 1 TO nx
L = zero(i): C(i) = modul(0, zim(L)) ^ 2: d(i) = 1: E(i) = 0: NEXT
ELSE
FOR i = 1 TO nx
d(i) = 0: E(i) = 0: C(i) = 1: NEXT
END IF
IF N$ = "odd" THEN so = -po
LOCATE 2, 12: COLOR 14, 1
PRINT "Prototip AG filtre ait transfer fonksiyonunun katsayıları"
COLOR 15, 1: st = 4: FOR i = 1 TO mx
st = st + 1: LOCATE st, 4: PRINT "c(", i, ")="; C(i): NEXT: PRINT
st = st + 1
FOR i = 1 TO nx
st = st + 1: LOCATE st, 4
PRINT "a(", i, ")="; a(i), "b(", i, ")="; B(i)
NEXT
st = st + 2
LOCATE st, 4: PRINT "ho="; ho: st = st + 2
IF N$ = "odd" THEN
LOCATE st, 4: PRINT "so="; so
ELSE
END IF
END SUB

```

-----LPBPTRANS Alt Programı-----

```

SUB lpbptrans
'Tlp(s) fonksiyonundan lp-bp dönüşümü ile bp fonksiyonunun oluşturulması
'Burada lp prototip fonksiyonu ; 1) Tlp(s)=1/(s2+as+b) , bw veya chy için
' 2) Tlp(s)=(s2+c)/(s2+as+b) , ell için
'Tbp(s)=(b4s4+b3s3+b2s2+b1s+b0)/(a4s4+a3s3+a2s2+a1s+a0) şeklindedir.
ta$ = "yok"
FOR i = 1 TO nx
a4(i) = 1: a3(i) = a(i) * BAND: a2(i) = 2 * w0 ^ 2 + B(i) * BAND ^ 2
a1(i) = a(i) * BAND * w0 ^ 2: a0(i) = w0 ^ 4: NEXT
IF N$ = "odd" THEN
tb$ = "var"
bb2 = 0: bb1 = BAND: bb0 = 0: ab2 = 1: ab1 = BAND * so: ab0 = w0 ^ 2
ELSE
tb$ = "yok"

```

```
bb2 = 0: bb1 = 0: bb0 = 1: ab2 = 0: ab1 = 0: ab0 = 1
```

```
END IF
```

```
IF yt$ = "bw" OR yt$ = "chy" THEN
```

```
FOR i = 1 TO nx
```

```
b4(i) = 0: b3(i) = 0: b2(i) = BAND ^ 2: b1(i) = 0: b0(i) = 0: NEXT
```

```
ELSEIF yt$ = "ell" THEN
```

```
FOR i = 1 TO nx
```

```
b4(i) = 1: b3(i) = 0: b2(i) = 2 * w0 ^ 2 + C(i) * BAND ^ 2: b1(i) = 0
```

```
b0(i) = w0 ^ 4: NEXT
```

```
ELSE
```

```
END IF
```

```
END SUB
```

```
'-----LPLPTRANS Alt Programı-----
```

```
SUB lplptrans
```

```
'Tlp(s) fonksiyonundan AG-AG dönüşümü ile lp fonksiyonunun oluşturulması
```

```
FOR i = 1 TO tx
```

```
IF nx$ = "odd" THEN
```

```
K = 2 * i
```

```
ELSE
```

```
K = (2 * i - 1)
```

```
END IF
```

```
a4(i) = 1: a3(i) = (a(K) + a(K + 1)) * wp2
```

```
a2(i) = (a(K) * a(K + 1) + B(K) + B(K + 1)) * wp2 ^ 2
```

```
a1(i) = (a(K) * B(K + 1) + B(K) * a(K + 1)) * wp2 ^ 3
```

```
a0(i) = (wp2 ^ 4) * B(K) * B(K + 1): NEXT
```

```
IF NS = "odd" THEN
```

```
ta$ = "var": aq = wp2: aq0 = so * wp2: aq1 = 1
```

```
ELSE
```

```
ta$ = "yok": aq = 1: aq0 = 1: aq1 = 0
```

```
END IF
```

```
IF nx$ = "odd" THEN
```

```
tb$ = "var": ab2 = 1: ab1 = a(1) * wp2: ab0 = B(1) * wp2 ^ 2: SELECT CASE yt$
```

```
CASE "ell"
```

```
bb2 = 1: bb1 = 0: bb0 = C(1) * wp2 ^ 2
```

```
CASE ELSE
```

```
'for bw and chy
```

```
bb2 = 0: bb1 = 0: bb0 = wp2 ^ 2
```

```
END SELECT
```

```
ELSEIF nx$ = "even" THEN
```

```
tb$ = "yok": bb2 = 0: bb1 = 0: bb0 = 1: ab2 = 0: ab1 = 0: ab0 = 1
```

```
END IF
```

```
IF yt$ = "bw" OR yt$ = "chy" THEN
```

```
FOR i = 1 TO tx
```

```
b4(i) = 0: b3(i) = 0: b2(i) = 0: b1(i) = 0: b0(i) = wp2 ^ 4: NEXT
```

```
ELSE
```

```
END IF
```

```
IF yt$ = "ell" THEN
```

```
FOR i = 1 TO tx
```

```
SELECT CASE nx$
```

```
CASE "odd"
```

```
K = 2 * i
```

```
CASE "even"
```

```
K = 2 * i - 1
```

```
END SELECT
```

```
b4(i) = 1: b3(i) = 0: b2(i) = (C(K) + C(K + 1)) * wp2 ^ 2
```

```
b1(i) = 0: b0(i) = C(K) * C(K + 1) * wp2 ^ 4: NEXT
```

```
ELSE
```

```
END IF
```

```
END SUB
```

```
'-----FUNCTION modul-----
```



```

FUNCTION modul (a, B)
'girilen kutbun modulünü bulur
modul = SQR(a ^ 2 + B ^ 2)

```

```

END FUNCTION

```

```

'-----FUNCTION mulim-----

```

```

FUNCTION mulim (a1, a2, b1, b2)
'iki karmaşık sayının çarpımının sanal kısmı
mulim = a1 * b2 + a2 * b1
END FUNCTION

```

```

'-----FUNCTION mulre-----

```

```

FUNCTION mulre (a1, a2, b1, b2)
'iki karmaşık sayının çarpımının reel kısmı
mulre = a1 * b1 - a2 * b2
END FUNCTION

```

```

'-----POLEZERO Alt Programı-----

```

```

SUB polezero
'1) BW ve CHY için : p1 , p2 , p3 , .....pn
'                               : (Q1>Q2>Q3>.....>Qn)
'2) ELL için : zn-pn , z(n-1)-p(n-1) , .....z2-p2 , z1-p1
'                               : (Qn>Q(n-1)>.....>Q1)
FOR i = 1 TO nx
pole(i) = i: zero(i) = i: NEXT
IF yt$ = "ell" THEN
FOR i = 1 TO nx
pole(i) = i: zero(i) = i: NEXT
ELSE
FOR i = 1 TO nx
pole(i) = nx + 1 - i: NEXT
END IF
LOCATE 2, 6: COLOR 14, 1
PRINT "      Maksimum dinamiklik için sıfır-kutup eşleştirme      "
COLOR 15, 1
IF yt$ = "bw" OR yt$ = "chy" THEN
st = 3
FOR i = 1 TO nx
st = st + 1: LOCATE st, 5
PRINT i; ".sıfır-kutup eşleştirme: ": LOCATE st, 31
PRINT " ( z "; CHR$(236); " ve p"; pole(i); ")"
NEXT
ELSE
st = 4
FOR i = 1 TO nx
st = st + 1: LOCATE st, 5
PRINT i; ".sıfır-kutup eşleştirme: ": LOCATE st, 31
PRINT " ( z"; zero(i); " ve p"; pole(i); ")"
NEXT
END IF
END SUB

```

```

'-----PROTOTİP Alt Programı-----

```

```

SUB prototip
'lp prototip fonksiyonuna ait kutup ve sıfırlar
LOCATE 2, 12: COLOR 14, 1
IF yt$ = "bw" OR yt$ = "chy" THEN
PRINT "      Prototip alacak geçiren filrenin kutupları      "
ELSE
PRINT "      Prototip alacak geçiren filrenin sıfırları ve kutupları      "
END IF
COLOR 15, 1: PRINT : mx = (m - m MOD 2) / 2: st = 3
IF yt$ = "ell" THEN

```

```

st = st + 1: COLOR 14, 1
LOCATE st, 4: PRINT "Sifirlar...": COLOR 15, 1: ELSE: END IF: st = st + 1
FOR i = 1 TO mx
st = st + 1: LOCATE st, 4: PRINT "z"; i, "; ", "z"; m + 1 - i, "="; CHR$(241); "j"; ABS(zim(i)); ""
NEXT: st = st + 2: COLOR 14, 1: LOCATE st, 4: PRINT "Kutuplar..."
COLOR 15, 1: nx = (N - N MOD 2) / 2: tx = (nx - nx MOD 2) / 2
IF ft$ = "AG" THEN
ck = tx
ELSEIF ft$ = "BG" THEN
ck = nx
END IF
IF nx MOD 2 = 0 THEN
nx$ = "even"
ELSE
nx$ = "odd"
END IF
st = st + 1
FOR i = 1 TO nx
st = st + 1: LOCATE st, 4: m = modul(pre(i), pim(i))
PRINT "p"; i, "; ", "p"; N + 1 - i, "="; pre(i); CHR$(241); "j"; ABS(pim(i))
NEXT: st = st + 1: LOCATE st, 4
IF ns = "odd" THEN
PRINT "po="; "p"; nx + 1, " = "; po: ELSE: END IF: PRINT: st = st + 2
LOCATE st, 4: PRINT "ho="; ho
END SUB

```

-----QUADS Alt Programı-----

```

SUB quads
Tq1(s)=(b12s2+b11s+b10)/(a12s2+a11s+a10)
Tq2(s)=(b22s2+b21s+b20)/(a22s2+a21s+a20)
DIM ll(40), nn(40)
SELECT CASE ck: CASE nx
FOR i = 1 TO nx
F(i) = geribes(a(i), B(i), C(i)): wp1(i) = w0: wp2(i) = wp1(i)
q1(i) = 2 * (1 - F(i) * d(i)) * w0 / (BAND * (a(i) - E(i) * F(i)))
q2(i) = q1(i)
IF yt$ = "bw" OR yt$ = "chy" THEN
b12(i) = 0: b11(i) = BAND: b10(i) = 0: b22(i) = 0: b21(i) = BAND: b20(i) = 0
END IF: NEXT: CASE tx
FOR i = 1 TO tx
IF nx$ = "odd" THEN
K = 2 * i
ELSE
K = (2 * i - 1)
END IF
grs(i) = (4 * (a(K) * a(K + 1) + B(K) + B(K + 1)) - (a(K) + a(K + 1)) ^ 2)
wp1(i) = SQR(((wp2 ^ 2) * grs(i)) / 8)
wp2(i) = wp1(i)
IF yt$ = "bw" OR yt$ = "chy" OR yt$ = "BW" OR yt$ = "CHY" THEN
q1(i) = 2 * wp1(i) / (wp2 * (a(K) + a(K + 1))): q2(i) = q1(i)
F(i) = (a0(i) - wp1(i) ^ 4) / b0(i)
b12(i) = 0: b11(i) = 0: b10(i) = wp2 ^ 2
b22(i) = 0: b21(i) = 0: b20(i) = wp2 ^ 2
ELSEIF yt$ = "ell" THEN
F(i) = (a0(i) - wp1(i) ^ 4) / (b0(i) - wp1(i) ^ 4)
q1(i) = (2 * wp1(i) * (1 - F(i))) / a3(i): q2(i) = q1(i)
END IF
NEXT: END SELECT
FOR i = 1 TO ck
a12(i) = 1: a11(i) = wp1(i) / q1(i): a10(i) = wp1(i) ^ 2
a22(i) = 1: a21(i) = wp2(i) / q2(i): a20(i) = wp2(i) ^ 2: NEXT

```

```
IF yt$ = "ell" THEN
```

```
FOR i = 1 TO ck
```

```
K(i) = 1 / (1 - F(i) * b4(i)): ll(i) = b2(i) - 2 * SQR(b0(i))
```

```
nn(i) = b2(i) + 2 * SQR(b0(i)): wz1(i) = (SQR(nn(i)) + SQR(ll(i))) / 2
```

```
wz2(i) = SQR(b0(i)) / wz1(i): b12(i) = K(i): b11(i) = 0
```

```
b10(i) = wz1(i) ^ 2 * K(i): b22(i) = 1: b21(i) = 0: b20(i) = wz2(i) ^ 2
```

```
NEXT
```

```
END IF
```

```
END SUB
```

```
'-----SEKİL Alt Programı-----
```

```
SUB sekil (sol%, sag%, ust%, alt%)
```

```
SCREEN 9: COLOR 14, 1: LOCATE ust%, sol%: PRINT CHR$(201)
```

```
LOCATE ust%, sag%: PRINT CHR$(187): LOCATE alt%, sol%: PRINT CHR$(200)
```

```
LOCATE alt%, sag%: PRINT CHR$(188)
```

```
FOR dikey% = ust% + 1 TO alt% - 1
```

```
LOCATE dikey%, sol%: PRINT CHR$(186): LOCATE dikey%, sag%: PRINT CHR$(186);
```

```
NEXT
```

```
yatay% = sag% - sol% - 1: ycizgi$ = STRING$(yatay%, 205)
```

```
LOCATE ust%, sol% + 1: PRINT ycizgi$: LOCATE alt%, sol% + 1: PRINT ycizgi$
```

```
END SUB
```

```
'-----SPE2 Alt Programı-----
```

```
SUB spe2
```

```
'giriş değerleri
```

```
pi = 8 * ATN(1): LOCATE 2, 20: COLOR 14, 1
```

```
PRINT " Alcak Geçiren Filtre Değerlerini Giriniz "
```

```
COLOR 15, 1: PRINT : CALL delay
```

```
S = 580: IF yt$ = "ell" OR yt$ = "ELL" THEN
```

```
OPEN "baslat1.dat" FOR INPUT AS #1: ELSEIF yt$ = "chy" OR yt$ = "CHY" THEN
```

```
OPEN "baslat2.dat" FOR INPUT AS #2
```

```
ELSEIF yt$ = "bw" OR yt$ = "BW" THEN
```

```
OPEN "baslat3.dat" FOR INPUT AS #3
```

```
ELSE: END IF
```

```
FOR K = 2 TO S - 8
```

```
IF yt$ = "ell" THEN
```

```
INPUT #1, v4
```

```
ELSEIF yt$ = "chy" THEN
```

```
INPUT #2, v4
```

```
ELSEIF yt$ = "bw" THEN
```

```
INPUT #3, v4
```

```
ELSE
```

```
END IF
```

```
genlik(K) = v4: NEXT
```

```
maxgenlik = 1
```

```
VIEW (50, 50)-(630, 280): WINDOW (0, 0)-(590, 230)
```

```
genlikayar = 200 / maxgenlik: LINE (1, 20)-(750, 20), 1
```

```
FOR K = 572 TO S - 380 STEP -1
```

```
IF K >= 235 THEN
```

```
LINE -(K, 4.86 * (20 * LOG(genlikayar * ABS(genlik(K)) + 1) / LOG(10))), 14
```

```
ELSE
```

```
LINE (1, 20)-(100, 10), 1
```

```
END IF: NEXT
```

```
'x eksen: LINE (600, 0)-(233, 0), 15
```

```
'y eksen
```

```
LINE (233, 0)-(233, 250), 15
```

```
IF yt$ = "ell" THEN
```

```
'Ap için
```

```
CALL delay
```

```
'fp2 için
```

```
FOR X = 1 TO 221 STEP 4
```

```
PSET (436, X), 12
```

```

NEXT: CALL delay
'Aa için
FOR X = 15 TO 100 STEP 4
PSET (500, X), 12: NEXT
FOR X = 110 TO 222 STEP 4
PSET (500, X), 12: NEXT: CALL delay: CALL delay
ELSEIF yt$ = "chy" THEN
'Ap için
CALL delay
'fp2 için
FOR X = 1 TO 214 STEP 4
PSET (390, X), 12: NEXT: CALL delay
'Aa için
FOR X = 30 TO 100 STEP 4
PSET (500, X), 15: NEXT
FOR X = 110 TO 222 STEP 4
PSET (500, X), 15: NEXT: CALL delay
'fa2 için
FOR X = 1 TO 29 STEP 4
PSET (500, X), 12: NEXT: CALL delay: CALL delay
ELSEIF yt$ = "bw" THEN
'Ap için
CALL delay
'fp2 için
FOR X = 1 TO 214 STEP 4
PSET (350, X), 12: NEXT: CALL delay
'Aa için
FOR X = 40 TO 100 STEP 4
PSET (500, X), 15: NEXT
FOR X = 110 TO 222 STEP 4
PSET (500, X), 15: NEXT: CALL delay
'fa2 için
FOR X = 1 TO 39 STEP 4
PSET (500, X), 12: NEXT: CALL delay
END IF
'sil
LOCATE 2, 20: COLOR 14, 1
PRINT "
LOCATE 23, 45: PRINT "Frekans,( )": LOCATE 12, 23: PRINT "Kazanç,(dB)"
VIEW (50, 50)-(630, 330): WINDOW (0, 0)-(630, 280)
'unit2:
LOCATE 2, 10
INPUT " Frekans birimini giriniz , (H=Hz, K=KHz, M=MHz)...: ", unit$
IF unit$ = "k" THEN
unit$ = "K": GOTO yaz2:END IF
IF unit$ = "m" THEN
unit$ = "M": GOTO yaz2:END IF
IF unit$ = "h" THEN
unit$ = "H": GOTO yaz2:END IF
IF unit$ = "H" OR unit$ = "K" OR unit$ = "M" THEN
GOTO yaz2:ELSE:END IF
BEEP: GOTO unit2
'yaz2:
LOCATE 23, 55: PRINT unit$, "Hz": LOCATE 2, 10
PRINT "
LOCATE 2, 10: PRINT " fp, geçirme bandı frekansını giriniz"
IF yt$ = "ell" THEN
LINE (465, 49)-(425, 40), 12:ELSEIF yt$ = "chy" THEN
LINE (415, 49)-(425, 40), 12:ELSEIF yt$ = "bw" THEN
LINE (370, 49)-(425, 40), 12:ELSE:END IF

```

```

LOCATE 22, 56: INPUT "", fp2: wp2 = pi * fp2
LOCATE 2, 10: PRINT "      fa, söndürme bandı frekansını giriniz"
IF yt$ = "ell" OR yt$ = "ELL" THEN LINE (518, 49)-(538, 40), 12
ELSEIF yt$ = "chy" THEN LINE (535, 49)-(538, 40), 12
ELSEIF yt$ = "bw" THEN LINE (535, 49)-(538, 40), 12
ELSE:END IF:LOCATE 22, 70: INPUT "", fa2: wa2 = pi * fa2
wp2 = pi * fp2: wa2 = pi * fa2:IF unit$ = "K" OR unit$ = "k" THEN
wp2 = wp2 * 1000: wa2 = wa2 * 1000
ELSEIF unit$ = "M" OR unit$ = "m" THEN
wp2 = wp2 * 1000000: wa2 = wa2 * 1000000:ELSE:END IF:LOCATE 2, 10
PRINT "      Ap(dB), maksimum geçirme bandı zayıflatmasını giriniz"
LOCATE 5, 34: PRINT "{": LOCATE 5, 31: INPUT "", Ap: LOCATE 2, 10
PRINT "      Aa(dB), minimum söndürme bandı zayıflatmasını giriniz"
LOCATE 13, 68: INPUT "", Aa
CLOSE #1
CLOSE #2
CLOSE #3
CALL anykey
CLS : omp = 1: oma = wa2 / wp2
END SUB

```

-----SPECIFICATIONS Alt Programı-----

```

SUB specifications
'giriş değerleri
pi = 8 * ATN(1): LOCATE 2, 20: COLOR 14, 1
PRINT "      Band Geçiren Filtre Değerlerini Giriniz      ": COLOR 15, 1
PRINT
S = 580
IF yt$ = "ell" THEN
OPEN "baslat1.dat" FOR INPUT AS #1
ELSEIF yt$ = "chy" THEN
OPEN "baslat2.dat" FOR INPUT AS #2
ELSEIF yt$ = "bw" THEN
OPEN "baslat3.dat" FOR INPUT AS #3
ELSE:END IF
FOR K = 2 TO S - 8
IF yt$ = "ell" THEN
INPUT #1, v4
ELSEIF yt$ = "chy" THEN
INPUT #2, v4
ELSEIF yt$ = "bw" THEN
INPUT #3, v4
ELSE
END IF
genlik(K) = v4
NEXT
maxgenlik = 1
VIEW (50, 50)-(630, 280): WINDOW (0, 0)-(590, 230)
genlikayar = 200 / maxgenlik: LINE (1, 20)-(1, 20), 1
FOR K = 2 TO S - 8
IF K >= 100 THEN
LINE -(K, 4.86 * (20 * LOG(genlikayar * ABS(genlik(K)) + 1) / LOG(10))), 14
ELSE
LINE (1, 20)-(100, 10), 1
END IF: NEXT
'x ekseni
LINE (600, 0)-(100, 0), 15
'y ekseni
LINE (100, 0)-(100, 250), 15
IF yt$ = "ell" THEN

```

```

'Ap için
FOR X = 101 TO 194 STEP 4
PSET (X, 223), 12: PSET (X, 214), 12: NEXT: CALL delay
'fp1 için
FOR X = 1 TO 221 STEP 4
PSET (200, X), 12: NEXT: CALL delay
'fp2 için
FOR X = 1 TO 221 STEP 4
PSET (436, X), 12: NEXT: CALL delay
'Aa için
FOR X = 15 TO 100 STEP 4
PSET (500, X), 12: NEXT
FOR X = 110 TO 222 STEP 4
PSET (500, X), 12: NEXT: CALL delay: CALL delay
ELSEIF yt$ = "chy" THEN
'Ap için
FOR X = 101 TO 218 STEP 4
PSET (X, 223), 12: PSET (X, 214), 12: NEXT: CALL delay
'fp1 için
FOR X = 1 TO 214 STEP 4
PSET (218, X), 12: NEXT: CALL delay
'fp2 için
FOR X = 1 TO 214 STEP 4
PSET (390, X), 12: NEXT: CALL delay
'Aa için
FOR X = 30 TO 100 STEP 4
PSET (500, X), 15: NEXT
FOR X = 110 TO 222 STEP 4
PSET (500, X), 15: NEXT: CALL delay
'fa2 için
FOR X = 1 TO 29 STEP 4
PSET (500, X), 12: NEXT: CALL delay
'fa1 için
FOR X = 1 TO 29 STEP 4
PSET (170, X), 12: NEXT: CALL delay: CALL delay
ELSEIF yt$ = "bw" THEN
'Ap için
FOR X = 101 TO 225 STEP 4
PSET (X, 223), 12: NEXT
FOR X = 101 TO 218 STEP 4
PSET (X, 214), 12: NEXT: CALL delay
'fp1 için
FOR X = 1 TO 214 STEP 4
PSET (218, X), 12: NEXT: CALL delay
'fp2 için
FOR X = 1 TO 214 STEP 4
PSET (350, X), 12: NEXT: CALL delay
'Aa için
FOR X = 40 TO 100 STEP 4
PSET (500, X), 15: NEXT
FOR X = 110 TO 222 STEP 4
PSET (500, X), 15: NEXT: CALL delay
'fa2 için
FOR X = 1 TO 39 STEP 4
PSET (500, X), 12: NEXT: CALL delay
'fa1 için
FOR X = 1 TO 39 STEP 4
PSET (150, X), 12: NEXT: CALL delay: CALL delay: END IF
'sil
LOCATE 2, 20: COLOR 14, 1

```

```

PRINT "
LOCATE 23, 40: PRINT "Frekans,( )": LOCATE 12, 7: PRINT "Kazaç,(dB)"
VIEW (50, 50)-(630, 330): WINDOW (0, 0)-(630, 280)
unit1:
LOCATE 2, 10
INPUT "      Frekans birimini giriniz ,(H=Hz, K=KHz, M=MHz)...: ", unit$
IF unit$ = "k" THEN
unit$ = "K": GOTO yaz1:END IF
IF unit$ = "m" THEN
unit$ = "M": GOTO yaz1:END IF
IF unit$ = "h" THEN
unit$ = "H": GOTO yaz1:END IF
IF unit$ = "H" OR unit$ = "K" OR unit$ = "M" THEN
GOTO yaz1:ELSE:END IF
BEEP: GOTO unit1
yaz1:
LOCATE 23, 51: PRINT unit$, "Hz"
LOCATE 2, 10: PRINT "
LOCATE 2, 10: PRINT "      fa1, Alt söndürme bandı frekansı"
IF yt$ = "ell" THEN
LINE (115, 40)-(190, 49), 12
ELSEIF yt$ = "chy" THEN
LINE (160, 40)-(180, 49), 12
ELSEIF yt$ = "bw" THEN
LINE (135, 35)-(160, 50), 12
ELSE:END IF
LOCATE 22, 20: INPUT "", fa1: wa1 = pi * fa1
LOCATE 2, 10: PRINT "      fp1, Alt geçirme bandı frekansı"
IF yt$ = "ell" THEN
LINE (215, 49)-(250, 40), 12
ELSEIF yt$ = "chy" THEN
LINE (230, 49)-(250, 40), 12
ELSEIF yt$ = "bw" THEN
LINE (230, 49)-(250, 40), 12
ELSE:END IF
LOCATE 22, 35: INPUT "", fp1: wp1 = pi * fp1
LOCATE 2, 10: PRINT "      fp2, Üst geçirme bandı frekansı"
IF yt$ = "ell" THEN
LINE (465, 49)-(425, 40), 12
ELSEIF yt$ = "chy" THEN
LINE (415, 49)-(425, 40), 12
ELSEIF yt$ = "bw" THEN
LINE (370, 49)-(425, 40), 12
ELSE:END IF
LOCATE 22, 56: INPUT "", fp2: wp2 = pi * fp2
LOCATE 2, 10: PRINT "      fa2, Üst söndürme bandı frekansı"
IF yt$ = "ell" THEN
LINE (518, 49)-(538, 40), 12
ELSEIF yt$ = "chy" THEN
LINE (535, 49)-(538, 40), 12
ELSEIF yt$ = "bw" THEN
LINE (535, 49)-(538, 40), 12
ELSE:END IF
LOCATE 22, 70: INPUT "", fa2: wa2 = pi * fa2
wa1 = pi * fa1: wp1 = pi * fp1: wp2 = pi * fp2: wa2 = pi * fa2
IF unit$ = "K" THEN
wa1 = wa1 * 1000: wp1 = wp1 * 1000: wp2 = wp2 * 1000: wa2 = wa2 * 1000
ELSEIF unit$ = "M" THEN
wa1 = wa1 * 1000000: wp1 = wp1 * 1000000: wp2 = wp2 * 1000000
wa2 = wa2 * 1000000:ELSE:END IF

```

```

LOCATE 2, 10: PRINT " Ap(dB), maksimum gecirme bandı zayıflatmasını giriniz"
LOCATE 5, 18: PRINT "{" : LOCATE 5, 13: INPUT "", Ap
LOCATE 2, 10: PRINT " Aa(dB), minimum söndürme bandı zayıflatmasını giriniz"
LOCATE 13, 68: INPUT "", Aa
CLOSE #1
CLOSE #2
CLOSE #3
CALL anykey
'normalized bandpass filter specifications
CLS : omp = 1: w0 = SQR(wp2 * wp1): BAND = wp2 - wp1: wa = w0 ^ 2 / wa1
IF wa <= wa2 THEN wa2 = wa ELSE wa1 = w0 ^ 2 / wa2
oma = (wa2 - wa1) / BAND
END SUB

```

```

'-----TRANSYAZ Alt Programı-----
SUB transyaz
CALL ekran: LOCATE 2, 20: COLOR 14, 1
PRINT " Giriş Blogu ": COLOR 15, 1: st = 4: C = 2
st = st + 2: LOCATE st, C: PRINT "Giriş kazanç kat..="; kaz: st = st + 2
LOCATE st, C: PRINT "ga="; ga * 1000000!; "uS": st = st + 1: LOCATE st, C
PRINT "gb="; gb * 1000000!; "uS": CALL anykey: CALL ekran
st = 4: LOCATE st, 2
IF ta$ = "var" THEN
CALL ekran: LOCATE 2, 20: COLOR 14, 1
PRINT " Ta Blok ": COLOR 15, 1: C = 57: st = 3: st = st + 1
LOCATE st, 2: PRINT " "; aq: st = st + 1: LOCATE st, 2
PRINT "Ta(s) = -----"
st = st + 1: LOCATE st, 2: PRINT " s+"; aq0
st = st + 1: LOCATE st, 2: PRINT "Kazanç katsayısı..., ka="; ka: st = 3
st = st + 1: COLOR 14, 1
LOCATE st, C: PRINT " Ta Blok...": COLOR 15, 1: LINE (400, 1)-(400, 315), 14
st = st + 1: LOCATE st, C: PRINT "R="; h; "ohm": st = st + 1: LOCATE st, C
PRINT "C="; ee; "F"
ELSE
END IF
IF ta$ = "var" THEN
CALL anykey: CALL ekran
ELSE
END IF
IF tb$ = "var" THEN
IF ft$ = "BG" THEN
CALL ekran: LOCATE 2, 20: COLOR 14, 1
PRINT " Tb Blok ": COLOR 15, 1: C = 57
st = 3: st = st + 1: LOCATE st, 2: PRINT " "; bb1; "s"
st = st + 1: LOCATE st, 2: PRINT "Tb(s) = -----"
st = st + 1: LOCATE st, 2: PRINT " s"; CHR$(253); "+"; ab1; "s+"; ab0
qb = SQR(ab0) / ab1: st = st + 2: LOCATE st, 2
PRINT "İyilik faktörü.....,qb="; qb
st = st + 1: LOCATE st, 2: PRINT "Kazanç katsayısı...,kb="; kb
st = 3: st = st + 1: COLOR 14, 1: LOCATE st, C
PRINT "İletkenlikler...": COLOR 15, 1: LINE (400, 1)-(400, 315), 14
st = st + 1: LOCATE st, C: PRINT "gb1="; gb1 * 1000000!; "µs"
st = st + 1: LOCATE st, C: PRINT "gb2="; gb2 * 1000000!; "µs"
st = st + 1: LOCATE st, C: PRINT "gb3="; gb3 * 1000000!; "µs"
COLOR 14, 1: st = st + 1: LOCATE st, C: PRINT "Kapasiteler...": COLOR 15, 1
st = st + 1: LOCATE st, C: PRINT "Cb1="; cb1 / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "Cb2="; cb2 / 1E-09; "nF"
ELSEIF ft$ = "AG" THEN
IF yt$ = "bw" OR yt$ = "chy" THEN
CALL ekran: LOCATE 2, 20: COLOR 14, 1
PRINT " Tb Block ": COLOR 15, 1

```



```

C = 57: st = 3: st = st + 1: LOCATE st, 2
PRINT "          "; bb0;
st = st + 1: LOCATE st, 2: PRINT "Tb(s) = -----"
st = st + 1: LOCATE st, 2: PRINT "          s"; CHR$(253); "+"; ab1; "s+"; ab0
qb = SQR(ab0) / ab1: st = st + 2: LOCATE st, 2
PRINT "İyilik faktörü...., Qb="; qb: st = st + 1: LOCATE st, 2
PRINT "Kazanç katsayısı...., kb="; kb
st = 3: st = st + 1: COLOR 14, 1: LOCATE st, C
PRINT "İletkenlikler....": COLOR 15, 1: LINE (400, 1)-(400, 315), 14
st = st + 1: LOCATE st, C: PRINT "gb1="; gb1 * 1000000!; "μs"
st = st + 1: LOCATE st, C: PRINT "gb2="; gb2 * 1000000!; "μs"
st = st + 1: LOCATE st, C: PRINT "gb3="; gb3 * 1000000!; "μs"
COLOR 14, 1: st = st + 1: LOCATE st, C: PRINT "Kapasiteler...."
COLOR 15, 1: st = st + 1: LOCATE st, C: PRINT "Cb1="; cb1 / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "Cb2="; cb2 / 1E-09; "nF"
ELSEIF yt$ = "ell" THEN
CALL ekran: LOCATE 2, 20: COLOR 14, 1
PRINT "          Tb Blok   ": COLOR 15, 1
C = 57: st = 3: st = st + 1: LOCATE st, 2
PRINT "          "; bb2; "s"; CHR$(253); "+"; bb0
st = st + 1: LOCATE st, 2: PRINT "Tb(s) = -----"
st = st + 1: LOCATE st, 2
PRINT "          s"; CHR$(253); "+"; ab1; "s+"; ab0
qb = SQR(ab0) / ab1: st = st + 2: LOCATE st, 2
PRINT "İyilik faktörü, Qb="; qb: st = st + 1: LOCATE st, 2
PRINT "Kazanç katsayısı...., kb="; kb
st = 3: st = st + 1: COLOR 14, 1: LOCATE st, C
PRINT "İletkenlikler....": COLOR 15, 1: LINE (400, 1)-(400, 315), 14
st = st + 1: LOCATE st, C: PRINT "gb1="; gb1 * 1000000!; "μs"
st = st + 1: LOCATE st, C: PRINT "gb2="; gb2 * 1000000!; "μs"
st = st + 1: LOCATE st, C: PRINT "gb3="; gb3 * 1000000!; "μs"
st = st + 1: LOCATE st, C: PRINT "gb4="; gb4 * 1000000!; "μs"
st = st + 1: LOCATE st, C: PRINT "gb5="; gb5 * 1000000!; "μs"
COLOR 14, 1: st = st + 1: LOCATE st, C: PRINT "Kapasiteler...."
COLOR 15, 1: st = st + 1: LOCATE st, C: PRINT "Cb1="; cb1 / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "Cb2="; cb2 / 1E-09; "nF"
END IF
END IF
ELSE
END IF
IF tb$ = "var" THEN
CALL anykey: CALL ekran
ELSE
END IF
st = 4
IF yt$ = "bw" OR yt$ = "chy" THEN
IF ft$ = "BG" THEN
FOR i = 1 TO ck
LOCATE 2, 20: COLOR 14, 1
IF kq$ = "quad" THEN
PRINT "          "; i; ". Quad Blok      "
ELSE
PRINT "          "; i; ". Kaskad Blok      "
END IF
COLOR 15, 1: st = st + 1: LOCATE st, 2
PRINT "          "; b11(i); "s": st = st + 1: LOCATE st, 2
PRINT "T1(s) = -----"
st = st + 1: LOCATE st, 2
PRINT "          s"; CHR$(253); "+"; a11(i); "s+"; a10(i)
st = st + 2: LOCATE st, 2

```

```

PRINT "          ", b21(i); "s": st = st + 1: LOCATE st, 2
PRINT "T2(s) = -----"
st = st + 1: LOCATE st, 2
PRINT "          s"; CHR$(253); "+"; a21(i); "s+"; a20(i)
IF kq$ = "quad" THEN
st = st + 2: LOCATE st, 2: PRINT "İyilik faktörü.....,Q =", q1(i)
st = st + 1: LOCATE st, 2: PRINT "Yeni geri besleme kats.,f=", F(i)
ELSE
st = st + 2: LOCATE st, 2: PRINT "İyilik faktörü.....,Q =", q1(i)
END IF
st = st + 2: LOCATE st, 2: PRINT "Biküadratik kazanç katsayıları..."
st = st + 2: LOCATE st, 2: PRINT "k1="; k1(i)
st = st + 1: LOCATE st, 2: PRINT "k2="; k2(i): COLOR 15, 1: C = 57: st = 3
st = st + 1: COLOR 14, 1: LOCATE st, C: PRINT "İletkenlikler....": COLOR 15, 1
LINE (400, 1)-(400, 315), 14: st = st + 1: LOCATE st, C
PRINT "g1="; g1(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g2="; g2(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g3="; g3(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g4="; g4(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g5="; g5(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g6="; g6(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "gx="; gx(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "gy="; gy(i) * 1000000!; "μS"
IF kq$ = "quad" THEN
st = st + 1: LOCATE st, C: PRINT "gz="; gz(i) * 1000000!; "μS"
ELSE
END IF
COLOR 14, 1
st = st + 1: LOCATE st, C: PRINT "Kapasiteler....": COLOR 15, 1
st = st + 1: LOCATE st, C: PRINT "C1="; c1(i) / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "C2="; c2(i) / 1E-09; "nF"
st = st + 1: LOCATE st, C: PRINT "C3="; c3(i) / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "C4="; c4(i) / 1E-09; "nF"
st = st + 1: LOCATE st, C
IF i < nx THEN
st = 3
CALL anykey
CALL ekran
ELSE
END IF
NEXT
PRINT
ELSEIF fts = "AG" THEN
FOR i = 1 TO tx
LOCATE 2, 20: COLOR 14, 1
IF kq$ = "quad" THEN
PRINT "          ", i; ".Quad Blok      "
ELSE
PRINT "          ", i; ".Kaskad Blok      "
END IF
COLOR 15, 1: st = st + 1: LOCATE st, 2: PRINT "          "; b10(i);
st = st + 1: LOCATE st, 2: PRINT "T1(s) = -----"
st = st + 1: LOCATE st, 2
PRINT "          s"; CHR$(253); "+"; a11(i); "s+"; a10(i)
st = st + 2: LOCATE st, 2: PRINT "          "; b20(i);
st = st + 1: LOCATE st, 2: PRINT "T2(s) = -----"
st = st + 1: LOCATE st, 2
PRINT "          s"; CHR$(253); "+"; a21(i); "s+"; a20(i)
IF kq$ = "quad" THEN
st = st + 2: LOCATE st, 2: PRINT "İyilik faktörü.....,Q=", q1(i)

```

```

st = st + 1: LOCATE st, 2: PRINT "Yeni geri besleme kats...,f="; F(i)
ELSE
END IF
st = st + 2: LOCATE st, 2: PRINT "Biquadratik kazanç katsayıları..."
st = st + 2: LOCATE st, 2: PRINT "k1="; k1(i)
st = st + 2: LOCATE st, 2: PRINT "k2="; k2(i)
COLOR 15, 1: C = 57: st = 3: st = st + 1
COLOR 14, 1: LOCATE st, C: PRINT "İletkenlikler...."
COLOR 15, 1: LINE (400, 1)-(400, 315), 14: st = st + 1: LOCATE st, C
PRINT "g1="; g1(i) * 1000000!; "μs": st = st + 1: LOCATE st, C
PRINT "g2="; g2(i) * 1000000!; "μs": st = st + 1: LOCATE st, C
PRINT "g3="; g3(i) * 1000000!; "μs": st = st + 1: LOCATE st, C
PRINT "g4="; g4(i) * 1000000!; "μs": st = st + 1: LOCATE st, C
PRINT "g5="; g5(i) * 1000000!; "μs": st = st + 1: LOCATE st, C
PRINT "g6="; g6(i) * 1000000!; "μs": st = st + 1: LOCATE st, C
PRINT "gx="; gx(i) * 1000000!; "μs": st = st + 1: LOCATE st, C
PRINT "gy="; gy(i) * 1000000!; "μs"
IF kq$ = "quad" THEN
st = st + 1: LOCATE st, C: PRINT "gz="; gz(i) * 1000000!; "μs"
ELSE
END IF
COLOR 14, 1: st = st + 1: LOCATE st, C: PRINT "Kapasiteler...."
COLOR 15, 1: st = st + 1: LOCATE st, C
PRINT "c1="; c1(i) / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "C2="; c2(i) / 1E-09; "nF"
st = st + 1: LOCATE st, C: PRINT "C3="; c3(i) / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "C4="; c4(i) / 1E-09; "nF"
st = st + 1: LOCATE st, C
IF i < tx THEN
st = 4: CALL anykey: CALL ekran
ELSE
END IF
NEXT: PRINT
END IF
ELSEIF yt$ = "ell" THEN
FOR i = 1 TO ck
LOCATE 2, 20: COLOR 14, 1
IF kq$ = "quad" THEN
PRINT "      "; i; ". Quad Blok      "
ELSE
PRINT "      "; i; ". Kaskad Blok      "
END IF
COLOR 15, 1
st = st + 1: LOCATE st, 2
PRINT "      "; b12(i); "s"; CHR$(253); "+"; b10(i)
st = st + 1: LOCATE st, 2: PRINT "T1(s) = -----"
st = st + 1: LOCATE st, 2
PRINT "      s"; CHR$(253); "+"; a11(i); "s+"; a10(i)
st = st + 2: LOCATE st, 2
PRINT "      "; b22(i); "s"; CHR$(253); "+"; b20(i)
st = st + 1: LOCATE st, 2: PRINT "T2(s) = -----"
st = st + 1: LOCATE st, 2
PRINT "      s"; CHR$(253); "+"; a21(i); "s+"; a20(i)
IF kq$ = "quad" THEN
st = st + 2: LOCATE st, 2: PRINT "İyilik faktörü.....,Q ="; q1(i)
st = st + 1: LOCATE st, 2: PRINT "Yeni geri besleme kats...,f="; F(i)
ELSE
st = st + 2: LOCATE st, 2: PRINT "İyilik faktörü.....,Q ="; q1(i)
END IF
st = st + 2: LOCATE st, 2: PRINT "Biquadratik kazanç katsayıları..."

```

```

st = st + 2: LOCATE st, 2: PRINT "k1="; k1(i)
st = st + 1: LOCATE st, 2: PRINT "k2="; k2(i): COLOR 15, 1
C = 57: st = 3: COLOR 14, 1: st = st + 1
LOCATE st, C: PRINT "İletkenlikler...": COLOR 15, 1
LINE (400, 1)-(400, 315), 14: st = st + 1: LOCATE st, C
PRINT "g1="; g1(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g2="; g2(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g3="; g3(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g4="; g4(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g5="; g5(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g6="; g6(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g7="; g7(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g8="; g8(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g9="; g9(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "g10="; g10(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "gx="; gx(i) * 1000000!; "μS": st = st + 1: LOCATE st, C
PRINT "gy="; gy(i) * 1000000!; "μS"
IF kq$ = "quad" THEN
st = st + 1: LOCATE st, C: PRINT "gz="; gz(i) * 1000000!; "μS"
ELSE
END IF
COLOR 14, 1
st = st + 1: LOCATE st, C: PRINT "Kapasiteler...": COLOR 15, 1
st = st + 1: LOCATE st, C: PRINT "C1="; c1(i) / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "C2="; c2(i) / 1E-09; "nF"
st = st + 1: LOCATE st, C: PRINT "C3="; c3(i) / 1E-12; "pF"
st = st + 1: LOCATE st, C: PRINT "C4="; c4(i) / 1E-09; "nF"
st = st + 1: LOCATE st, C
IF i < nx THEN
st = 3
CALL anykey: CALL ekran
ELSE
END IF
NEXT
END IF
CALL anykey
END SUB

```

ÖZGEÇMİŞ

Ömer ORAK, 1969 yılında İzmit'te doğdu. 1986' da İzmit Endüstri Meslek Lisesi Elektronik Bölümünden mezun oldu. 1993 yılında, İstanbul Teknik Üniversitesi Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü'nü bitirdi. 1994 yılında İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Anabilim Dalı, Elektronik ve Haberleşme programında yüksek lisans eğitimine başladı.

