

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

32614

YÜKSEK DERECEDEN AKTİF OTA - C FİLTRELERİNİN

QUAD YÖNTEMİ KULLANARAK BİLGİSAYAR YARDIMIYLA TASARIMI

YÜKSEK LİSANS TEZİ

Müh. Ramazan KÖPRÜ

Tezin Enstitüye Verildiği Tarih : 13 Haziran 1994

Tezin Savunulduğu Tarih : 30 Haziran 1994

Tez Danışmanı : Prof.Dr.Ali Nur GÖNÜLEREN

Diğer Jüri Üyeleri : Prof.Dr. Cevdet ACAR

: Doç.Dr. Sait TÜRKÖZ

**F.C. YÜKSEKÖĞRETİM KURULU
DOKÜMAN YAYINLARI**

HAZİRAN 1994

ÖNSÖZ

Tez çalışmam süresince bana her türlü imkanı temin eden , filtre tasarımı konusundaki geniş bilgi ve tecrübeleri ve olumlu eleştirileriyle çalışmamı yönlendiren değerli hocam sayın Prof. Dr. Ali Nur GÖNÜLEREN' e teşekkürü bir borç bilirim.

Ayrıca öğrenimim boyunca maddi manevi desteklerini esirgemeyen aileme ve bana yüksek lisans bursu sağlayan EREĞLİ DEMİR VE ÇELİK FABRİKALARI T.A.Ş.' ne şükranlarımı sunarım.

İstanbul, Haziran 1994

Ramazan KÖPRÜ

İÇİNDEKİLER

ŞEKİL LİSTESİ.....	v
TABLO LİSTESİ.....	vii
ÖZET	viii
SUMMARY	ix
BÖLÜM 1. GİRİŞ	1
BÖLÜM 2. QUAD YÖNTEMİ	5
2.1. Quad Alt Devre Yapısı	5
2.1.1. Quad Devresinin Gerçeklenmesi	7
2.1.2. Quad Devresine İlişkin Geribesleme Katsayısının Bulunması	9
2.2. En Genel İkinci Derece Denkleminde Alçak Geçiren - Band Geçiren Dönüşümü Yardımıyla Elde Edilen 4. Dereceden Band Geçiren Filtrelerin Tasarım	11
2.3. Yüksek Dereceden Filtrelerin Quad Yöntemi ile Tasarımı	13
2.3.1. Yüksek Dereceden Filtrelerin Tasarımında Quad Yöntemi İçin Sıfır Kutup Düzenlemesi	14
BÖLÜM 3. SİMETRİK BANDGEÇİREN FİLTRELERİN QUAD YÖNTEMİYLE TASARIM DENKLEMLERİNİN ELDE EDİLMESİ	19
3.1. Filtrelerin Sınıflandırılması	19
3.2. Yaklaşıklık Sorunu	22
3.2.1. Butterworth Yaklaşıklığı	22
3.2.2. Chebyshev Yaklaşıklığı	25
3.2.3. Eliptik Yaklaşıklık	26
3.3. Verilen Özellikleri Sağlayan Band Geçiren Filtre Transfer Fonksiyonunun Hesaplanması	29
3.3.1. Alçak Geçiren - Band Geçiren (AG-BG) Dönüşümü	30
3.4. Elde edilen BG Filtre Transfer Fonksiyonu ile Quad Tasarım....	32
3.4.1. Tasarım Adımları.....	32
BÖLÜM 4. OTA ELEMANI VE OTA-KAPASİTE ELEMANI İLE OLUŞTURULAN BİQUADLAR.....	43
4.1. Giriş	43
4.2. OTA Elemanı ve Modelleri	44
4.2.1. İdeal OTA Modeli	45

4.2.2.	Pratikte Kullanılan OTA Modelleri	45
4.2.3.	Simetrik CMOS OTA	47
4.3.	OTA ve Kapasite Elemanlarıyla Oluşturulan Biquadlar	48
4.3.1.	İncelenen OTA-C Biquadları Arasındaki Farklar	58
4.4.	Simetrik CMOS-OTA' nın İdealsizliği	59
4.4.1.	Simülasyonlar	60
BÖLÜM 5.	YÜKSEK DERECEDE SİMETRİK OTA-C BANDGEÇİREN QUAD FİLTRELERİNİN DİNAMİK DEĞİŞİMİ ARTIRACAK ŞEKİLDE GERÇEKLENMESİ VE DUYARLIK	69
5.1.	Quad Yönteminde Dinamik Değişimin Artırılması	69
5.2.	Biquadratik Altdevrelerin Kazanç Katsayılarının Bulunması ve OTA-C Elemanlarıyla Gerçeklenmesi	73
5.2.1.	Doyma Durumuna Göre Kazanç Katsayılarının Belirlenmesi.....	73
5.2.2.	Biquadratik Altdevrelerin Gerçekleştirilmesi	83
5.3.	OTA-C Quad Filtrelerinin Duyarlığı	88
BÖLÜM 6.	TASARLANAN OTA-C BANDGEÇİREN QUAD FİLTRELERİNİN MAKSİMUM GİRİŞ GERİLİMİNİN HESAPLANMASI	91
6.1.	Genel Kavramlar	91
6.2.	Butterworth, Chebyshev ve Eliptik Türden Yüksek Dereceli Quad Filtrelerinin Maksimum Giriş Geriliminin Hesaplanması.....	93
6.2.1.	Butterworth ve Chebyshev Filtreler için Maksimum Giriş Geriliminin Hesaplanması	94
6.2.2.	Eliptik Filtreler İçin Maksimum Giriş Geriliminin Hesaplanması...	96
BÖLÜM 7.	SONUÇLAR VE ÖNERİLER.....	102
KAYNAKLAR.....		105
EK :	Bilgisayar Programı "QUAD".....	108
ÖZGEÇMİŞ.....		146

ŞEKİL LİSTESİ

<u>Şekiller</u>	<u>Sayfa No</u>
2.1. Quad Devre Yapısı.....	5
2.2. Yüksek Dereceden Quad Devre Topolojisi.....	13
2.3. (2.28) Denklemine İlişkin Frekans Cevabı.....	16
2.4. Örnek 2.1'deki Filtreye İlişkin Duyarlık Karşılaştırmaları.....	18
3.1. AG Filtre Karakteristikleri.....	20
3.2. YG Filtre Karakteristikleri.....	20
3.3. BG Filtre Karakteristikleri.....	21
3.4. BS Filtre Karakteristikleri.....	21
3.5. BG Filtre Karakteristikleri.....	29
3.6. Butterworth, Chebyshev ve Eliptik Türden BG Filtreler İçin Yüksek Dereceden Quad Devresi.....	32
3.7. Örnek 3.1'deki Filtrenin Frekans Cevabı.....	37
3.8. Örnek 3.2'deki Filtrenin Frekans Cevabı.....	39
3.9. Örnek 3.3'deki Filtrenin Frekans Cevabı.....	42
4.1. OTA a) Devre Sembölü b) İdeal Eşdeğer Devresi.....	45
4.2. İdeal Olmayan OTA.....	46
4.3. Simetrik CMOS-OTA Yapısı.....	47
4.4. Temel OTA-C Yapıları a) Toplayıcı b) İntegratör.....	49
4.5. Genel OTA-C Biquad.....	49
4.6. Tablo 4.2'deki Transfer Fonksiyonlarını Gerçekleyen Genel OTA-C Biquad Devresi.....	51
4.7. Genel Amaçlı Bir OTA-C Biquad Devresi.....	53
4.8. $G(s)$ 'e Karşı Düşen İşaret Akış Diyagramı.....	54
4.9. $G(s)$ 'i Gerçekleyen OTA-C Devresi.....	55
4.10. $G(s)$ 'i Gerçekleyen Minimum OTA-C Biquad Devresi.....	56
4.11. İdeal OTA İçin SPICE Modeli a) İdeal OTA b) SPICE Modeli	60
4.12.a. 1. Deney Düzenegi.....	61
4.12.b. Simetrik CMOS-OTA'ya İlişkin İletkenlik-Frekans ($gm-f$) Eğrisi.....	62
4.12.c. Simetrik CMOS-OTA'ya İlişkin Kontrol Akımı-İletkenlik ($IB-gm$) Eğrisi.....	62
4.12.d, e. İdeal OTA ve Simetrik CMOS-OTA ile Tablo 4.4'deki 1. b Altdevresine İlişkin SPICE Simülasyonu.....	63
4.13.a,b. İdeal OTA ile Simetrik CMOS-OTA ile Tablo 4.4'deki 2 Numaralı Altdevreye İlişkin SPICE Simülasyonu.....	65
4.14. a,b. İdeal OTA ile Simetrik CMOS-OTA ile Tablo 4.4'deki 3 Numaralı Altdevreye İlişkin SPICE Simülasyonu.....	66
4.15.a,b. İdeal OTA ile Simetrik CMOS-OTA ile Tablo 4.4'deki 4 Numaralı Altdevreye İlişkin SPICE Simülasyonu.....	67
5.1. Yüksek Dereceden Quad Filtresine İlişkin Kritik Gerilimler.....	70
5.2.a. OTA-C Quad Devresindeki H_0 Bloğuna İlişkin Altdevre.....	70

5.2.b. OTA-C Quad Devresindeki To Bloğuna İlişkin Altdevre.....	71
5.2.c. Butterworth veya Chebyshev Filtre İçin OTA-C Quad Devresine İlişkin Altdevre.....	71
5.2.d. Eliptik Filtre İçin OTA-C Quad Filtresine İlişkin Altdevre.....	71
5.3. Kazanç Değerleri Atanmış Quad Devresi.....	74
5.4.a. Örnek 3.3'deki Filtrenin V_{j1} Gerilimi.....	77
5.4.b. Örnek 3.3'deki Filtrenin V_{21} Gerilimi.....	77
5.4.c. Örnek 3.3'deki Filtrenin V_{j2} Gerilimi.....	78
5.4.d. Örnek 3.3'deki Filtrenin V_{22} Gerilimi.....	78
5.4.e. Örnek 3.3'deki Filtrenin V_{j3} Gerilimi.....	79
5.4.f. Örnek 3.3'deki Filtrenin V_{23} Gerilimi.....	79
5.5.a. Kazanç Katsayıları Atandığında Örnek 3.3'teki Filtrenin V_{j1} Gerilimi.....	80
5.5.b. Kazanç Katsayıları Atandığında Örnek 3.3'teki Filtrenin V_{21} Gerilimi.....	80
5.5.c. Kazanç Katsayıları Atandığında Örnek 3.3'teki Filtrenin V_{j2} Gerilimi.....	81
5.5.d. Kazanç Katsayıları Atandığında Örnek 3.3'teki Filtrenin V_{22} Gerilimi.....	81
5.5.e. Kazanç Katsayıları Atandığında Örnek 3.3'teki Filtrenin V_{j3} Gerilimi.....	82
5.5.f. Kazanç Katsayıları Atandığında Örnek 3.3'teki Filtrenin V_{23} Gerilimi.....	82
5.6. Örnek 3.3'te Verilen Filtrenin Quad Yöntemi Kullanılarak OTA-C Elemanlarıyla Gerçeklenmesi.....	85
6.1. Örnek 3.2'deki Filtreye İlişkin Dominat Terim V_3 'ün Frekans Cevabı.....	100
a) Filtre Girişine Birim İmpuls Gerilimi Uygulandığında	
b) Filtre Girişine 318.4955 mV Değerinde İmpuls Gerilimi Uygulandığında	

TABLO LİSTESİ

<u>Tablolar</u>	<u>Sayfa No</u>
4.1. Şekil 4.5'deki Biquada İlişkin Değişik Koşullar ve Filtre Tipleri	50
4.2. Şekil 4.6.'daki Devreye İlişkin çeşitli Koşullara Göre Elde Edilen. Filtre Tipleri	52
4.3. Şekil 4.7'deki Biquad İçin Çeşitli Koşullara Göre Elde Edilen Filtre Tipleri	54
4.4. Şekil 4.8'deki İşaret Akış Grafi. Kullanılarak Oluşturulan OTA-C Filtreleri	57
4.5. CMOS-OTA'daki MOS Elemanlarının Boyutları	60
4.6. Tablo 4.4'teki 2,3,4 numaralı OTA-C Biquadların Simülasyonu İçin Gerekli Koşullar ve Sonuçlar.....	64
5.1. Örnek 3.3'teki Filtrenin Kritik Düzümlerine İlişkin Tüm Banddaki En Büyük Gerilim Değerleri.....	75
5.2. Örnek 3.3'teki Filtreye İlişkin Geribesleme ve Kazanç Katsayıları	76
5.3. Butterworth , Chebyshev ve Eliptik Türden OTA-C Quard Filtrelerine İlişkin Temel Biquad Devrelerinin Eleman Değerleri.....	84
5.4. Örnek 3.3'teki Filtre İçin Eleman Değerleri.....	85
5.5. Örnek 3.1'deki Filtreye İlişkin Tüm Banddaki En Büyük Gerilim Değerleri	86
5.6. Örnek 3.2'deki Filtreye İlişkin Biquadlara Atanacak Kazanç Katsayıları.....	86
5.7. Örnek 3.2'deki Filtreye İlişkin Eleman Değerleri.....	88
5.8. Minimum OTA-C Biquadlarına İlişkin W_o ve Q Duyarlılıkları.....	90
6.1. Örnek 3.1'deki Filtreye İlişkin H ve Y 'lerin Maksimum Değerleri.....	99

ÖZET

Literatürde OTA (Operational Transconductance Amplifier, İşlemsel geçiş iletkenlik kuvvetlendiricisi) ve kapasite elemanları ile oluşturulmuş çeşitli biquadlar bulunmaktadır. Bu biquadratik altdevreler kullanılarak yüksek dereceden filtre tasarımı önerilmekle beraber, gerçeklemelerin çoğunda opamp (İşlemsel Kuvvetlendirici, İ.K.) ve RC elemanları kullanılmıştır. Bu tezde, sadece OTA ve kapasite elemanları kullanarak, yüksek dereceden aktif filtre tasarımı amaçlanmıştır. Quad yönteminin diğer tasarım yöntemlerine göre avantajlı yönleri gözönüne alınarak tasarımın bu yöntemle yapılması kararlaştırılmıştır. Bu bakımdan tez yüksek dereceden filtrelerin OTA-C elemanlarıyla gerçekleştirilmesi anlamında yeni bir çalışma niteliğindedir.

OTA, tümleşik devre teknolojisine uygun olması, iletkenlik değerinin ayarlanabilmesi, yüksek frekans davranışının iyi olması gibi nedenlerden ötürü aktif filtre tasarımı için cazip bir eleman niteliğindedir. Son yıllarda, OTA ve kapasite elemanlarının kullanımı gittikçe yaygınlaşmaya başlamış, filtre tasarımında opamp (işlemsel kuvvetlendirici-İ.K.) ve RC elemanlarına göre daha çok tercih edilir olmuştur.

Bölüm 1' de, filtrelerin tarihsel gelişimiyle paralel olarak, kullanılan teknolojik elemanların ve filtre tasarım tekniklerinin kısa bir incelemesi yapılmaktadır.

Bölüm 2' de, literatürde Quad yöntemi olarak bilinen yüksek dereceden filtre tasarım yöntemine yer verilmiştir.

Bölüm 3' te, Butterworth, Chebyshev ve Eliptik türden simetrik bandgeçiren filtrelerin Quad yöntemiyle filtre tasarım denklemlerinin elde edilişine değinilmektedir..

Bölüm 4' te, OTA elemanı tanıtılmaktadır. OTA' ya ilişkin ideal ve pratik modeller verilmekte, pratik modellerden simetrik CMOS OTA' nın idealsizliği ile ilgili SPICE denemeleri verilmektedir. Ayrıca literatürde yer alan OTA-C biquadları incelenmektedir.

Bölüm 5' te, Quad yönteminin bilinen dinamik davranış incelemesi, simetrik bandgeçiren OTA-C filtrelere uygulanmakta ve sonuçta elde edilen tasarım denklemlerine ilişkin OTA-C gerçekleştirilmesi ve eleman değerlerinin hesaplanması verilmektedir.

Bölüm 6' da, tasarlanan filtrenin doğrusal çalışabilmesi için, maksimum filtre giriş geriliminin genliği hesaplanmakta ve dıyarlığa değinilmektedir.

EK' te, simetrik bandgeçiren OTA-C filtrelerinin Quad ve kaskad yöntemi ile tasarımını yapan bir bilgisayar programı verilmektedir.

SUMMARY

COMPUTER AIDED DESIGN OF HIGH ORDER ACTIVE OTA-C FILTERS USING QUAD CONFIGURATION

Electronics, named the technology of the today's modern age, is growing rapidly and almost everyday presents many new and useful products for the human being. Most of these products consist of circuits that select the electrical signals according to their frequencies and put these signals into another form. These kinds of circuits are known as filters, and have many applications in electronics from daily use to the high technology.

At the beginning of this century, most of voice-frequency filters were realized using discrete RLC passive circuits. For these networks, inductors were a problem since they were large components, difficult to produce and tune. In the 1950s, it was understood that the more advantageous filters could be obtained if the inductors were replaced by active networks. Therefore, designers started to commonly use the active-RC networks, which were composed of op amp's, compared to the former technique [1]. Together with the rapid growth of the integrated circuit technology, many active and passive components became realizable in very small silicon chips. Especially, VLSI technology presents many advantages to the designers.

Today, with the use of switched-capacitor filters (SC filters), the disadvantages of analog filters have been eliminated. Meanwhile, time-constants (RC) can be accurately obtained with the ratio of two MOS capacitors multiplied by a clock frequency. Moreover, resistors can be simulated with the capacitors. So, an important amount of silicon area can be saved. SC filters do not require an additional tuning and they can be tuned electronically [1], [2]. However, these types of filters can not operate properly at higher frequencies than voice frequencies because of the op amp's limited bandwidth.

Digital filters have also some limitations such as computation time, analog-digital conversion time and especially Nyquist rate.

Continuous-time analog filters do not have such frequency limitations as in SC and digital filters and theoretically can use all available frequency band. Thus, they can be used at high frequencies more reliably than the other two filter types.

Because op amps are highly developed and cheap components, they have been used in the discrete active filter design for many years. However, in order to integrate analog and other signal processing circuits in the same silicon chip, more compatible gain devices, OTAs, were required. OTAs can be easily integrated in the

monolithic form, adjusted electronically and provide simpler circuitry [1]. The transconductance gain of OTA can be used as a design parameter in the OTA based circuits. This gain can be adjusted by varying the control current of the OTA [3]. OTA uses CMOS technology; so, it overcomes difficulties of the bipolar technology (small chip area, economy, etc.).

On the other hand, grounded capacitor can also be easily realized using CMOS technology and forms a suitable component pair with OTA. Furthermore, it eliminates parasitic effects. In summary, grounded capacitor structures are preferred to floating capacitors.

Generally, a filter transfer function can be realized using only OTAs and capacitors. Therefore, a filter can be designed using less component compared to its active-RC equivalent. The high frequency performans of the OTA is noteworthy and quite advantageous according to op amp. While an OTA can operate at frequencies in MHz's, for a typical op amp this is limited in some hundred KHz's.

In summary, it seems that the use of circuits composed of OTAs and grounded capacitors in the high frequency continuous-time monolithic analog filters is extremely advantageous.

In the engineering, realization of the high order filters is quite important and there have been many studies in the literature to realize the transfer function of the form

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

These works can be classified into five groups; the direct method, cascade realization, inductance simulation, frequency dependent negative resistor (FDNR) and multiloop-feedback topologies (MF) [1], [4].

Direct method is not advisable to realize the transfer function $H(s)$ since the resulting circuits are very sensitive to component variations. Therefore, desired performance can not be obtained sufficiently using this approach.

In the cascade method, the high-order transfer function $H(s)$ is factored into second-order biquadratic functions. These functions are realized using the subcircuits called biquads to implement the $H(s)$. This method is commonly used in industry because it is easy to implement. Since it has a modular structure, obtained filter can be tuned easily. However, this kind of filters is too sensitive to the component variations in the passband.

In the inductance simulation and FDNR approach, the main idea is to get inductorless realization and the sensitivity behaviour of these circuits depend on the quality of the opamps used.

Even though the multiloop-feedback approach is similar to the cascade method and splits the $H(s)$ into second-order biquads, it uses some kind of feedback topology.

However, its most advantageous side is that it provides high-order filter realizations which have better passband sensitivity compared to those realized with the cascade method. Some studies on MF filters can be found in the literature such as FLF, PRB, SCF, LF, CB, IFLF, MSF. None of these methods can realize the general transfer function, $H(s)$. Although the method so-called Generalized FLF (GFLF) can realize a general transfer function of order n , it does not give a good sensitivity and difficult to design [4], [5].

The QUAD method uses a special topology of GMF (Generalized Multiloop-feedback Filters), [6], and some advantages compared to above methods. The major advantages of this method are the following, [4]:

1. It is general and can be applied to high order symmetrical or unsymmetrical active filters.
2. It is extremely easy according to all other methods except the cascade design
3. It gives less sensitive circuits than the other topologies.
4. It does not require summation and feedforward circuits.
5. It is modular, economic and easy to tune.

In this study, the aim is to realize high order filters using only OTAs and grounded capacitors. From the above considerations, the Quad method seems to be the most advantageous method compared to the others. Thus, combining the advantageous sides of OTA component and Quad method, it is intended to realize high order filter circuits which are completely suitable for the integrated circuit technology and easy to tune. Furthermore, resulting circuits will have less component and good high frequency behavior. From this point of view, it will be possible to realize extremely advantageous circuits in many aspects compared to the op amp-RC networks.

In the thesis, the design has been done for only symmetrical bandpass high order filters. However, it can be achieved for all types of filters with less effort. So, this study can be thought as an introduction to the general OTA-C realization using Quad method. For a given set of bandpass specifications, using well-known filter approximations (Chebyshev, Butterworth, Elliptic) and Quad method, OTA-C realization of the bandpass filter can be obtained easily with the aid of this study.

Since the cascade realization is a special case of the Quad method, OTA-C cascade realizations can be readily obtained using the same procedure.

In section 2, the main properties of the Quad method are considered. The basic Quad subcircuit contains two second order biquad circuits and a feedback path as shown in Figure 1, [4].

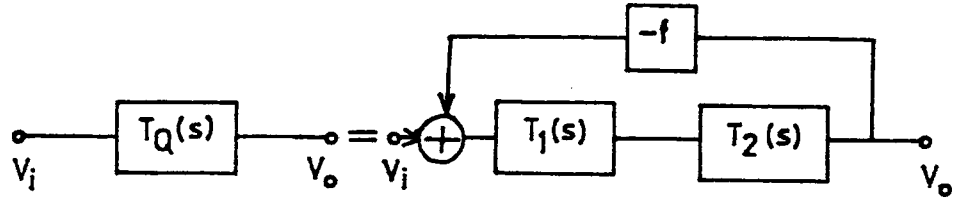


Figure 1. Basic Quad topology.

The transfer function of the circuit shown in figure 1 can be written as;

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

where,

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

To realize a high order transfer function given with the equation (1), it is split into 4th order transfer functions of the form (2). Then, these 4th order quadratic functions are factored into two biquadratic transfer functions given with the equation (3). To be able to obtain identical, economic, easy to manufacture and less sensitive circuits, the feedback coefficient is determined such that both radial frequencies and quality factors are equal to each other.

In section 3, the calculation of the Quad and cascade design equations of symmetrical bandpass filters are proposed according to the given bandpass specifications using three classical approximation; Butterworth, Chebyshev, Elliptic. Using calculated design equations, the filter can be realized with the OTA and grounded capacitors which is considered in the fifth section.

In filter design, usually passband, stopband and transition-band requirements for magnitude or phase are prescribed. The main task of the designer is to obtain a realizable network function $H(s)$ of the form given by equation (1) which meets the desired specifications. To be able to find $H(s)$ network function; first, a lowpass (LP) prototype function, $H(p)$, should be formed using one of the Butterworth, Chebyshev and Elliptic approximation. Obtained prototype function are then transformed to the $H(s)$ which meets the desired filter specifications [7].

In the thesis, from the given bandpass filter specifications, normalized lowpass filter specifications are determined. Then, using one of the Butterworth, Chebyshev Elliptic approximations, the poles/zeros of this lowpass prototype filter are calculated. Combining the complex conjugate poles/zeros, lowpass prototype filter transfer function, $H(p)$, are formed. The original bandpass filter function, $H(s)$, which meets

the given specifications are then obtained from the prototype lowpass filter using lowpass-bandpass transformation.

The resulting filter function $H(s)$ are then factored into 4th order transfer functions to be able to realize the Quad or cascade design. In addition, a design example is given for each approximation.

In section 4, OTA component will be considered. Its ideal and practical models such as bipolar and symmetrical CMOS OTA are also given. In addition, several OTA-C biquads reported in the recent literatures are considered. Most advantageous of these biquads are proposed to be used in the Quad design as the second-order building block.

OTA is basically a voltage controlled current source as seen from figure 2 and its output current is given with the expression of

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

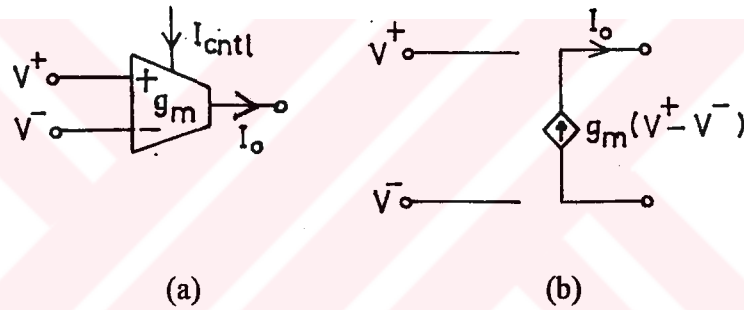


Figure 2. OTA component a) Circuit symbol b) Ideal equivalent circuit

As seen from Figure 2, its input and output impedances are infinity. The transconductance gain, g_m , can be adjusted by varying the control current, I_{ctrl} . g_m and I_{ctrl} are proportional to each other.

Many practical OTA have some problems such as finite input and output impedances, the frequency dependence of the transconductance gain and parasitic capacitors. In this case, the filter may not be operate properly.

Available bipolar OTAs such as CA3080 has limited dynamic range and it can not operate linearly for the input voltages greater than certain values. The other practical OTA model is the symmetrical CMOS-OTA. A detailed treatment of its nonideality using SPICE simulation is also given in Section 4.

Four type of OTA-C biquads are considered which are available in recent literature. These biquads are compared with each other in several ways to determine the best one which will be used in the Quad design as a second-order building block.

In section 5, the well-known dynamic range consideration of Quad filters is applied to symmetrical bandpass OTA-C Quad filters. Therefore, assigned gain factors of each biquadratic subcircuits and new feedback factors are calculated. The resulting gain assigned biquadratic subcircuits are realized with the OTA-C biquads. Some design examples are given as well.

In section 6, maximum input voltage amplitude for OTA-C Quad filters will be calculated. This level should not be exceeded so that the filters operate linearly. So, it is very important to find the value of the maximum input voltage level not causing clipping and slew rate limiting [8]. Maximum input signal levels are calculated for the OTA-C Quad filters which were given as examples in the preceeding sections. In addition, sensitivity analysis of the OTA-C Quad filters are considered briefly.

In the Appendix, a useful computer program,QUAD, which was evaluated using QUICKBASIC language, will be presented. This program calculates the Quad and cascade design equations depending on given bandpass specifications using one of the three approximations. Then, it obtains the gain factors of biquadratic subcircuits considering the maximum dynamic range. As a result, all OTA transconductance and capacitor values are calculated. Finally, the maximum input voltage amplitude is found in volts for the designed filter. At the same time, it is possible to see all node voltage transfer functions and transfer admittance functions as grafically.

BÖLÜM 1. GİRİŞ

Çağın teknolojisi olarak nitelenen elektronik, başdöndürücü bir hızla gelişmekte, hemen her gün pekçok yeni ve yararlı ürünü insanlığın hizmetine sunmaktadır. Bu ürünlerin çoğu, elektriksel işaretleri frekanslarına göre seçen ya da bu işaretleri başka bir biçime sokan devrelerden oluşmaktadır. Bu tür devreler filtre olarak bilinir ve günlük kullanımdan yüksek teknolojiye kadar pekçok uygulama alanına girmiştir.

Bu yüzyılın başlarında, çoğu ses frekans filtreleri ayrık RLC elemanlarıyla gerçekleştiriliyordu. Bu devrelerde, endüktans elemanları, geniş yer kaplamaları, üretimlerinin ve ayarlarının zor olması vb. gibi nedenlerden ötürü bir sorun oluşturmaktaydı. 1950' lerin sonunda, endüktans elemanlarının yerinin aktif elemanlar almaya başlamış, böylece İ.K.' lerden oluşmuş aktif-RC devreleri yaygın olarak kullanılmaya başlamıştır [1]. Özellikle VLSI teknolojisinin gelişimiyle beraber, pekçok aktif ve pasif eleman küçük bir silikon kırımcı içinde gerçekleştirilebilir olmuştur.

Günümüzde SC (Anahtarlama-Kapasite) filtreleri kullanılarak, analog filtrelerdeki bazı sakıncalar önlenebilmiştir. Bu anlamda; RC zaman sabitleri, iki MOS kapasite elemanı oranının bir saat frekansı ile çarpılmasından dolayı kesin olarak belirlenebilir. Ayrıca, dirençler kapasitelerle simüle (taklit) edildiği için, kullanılan silikon kırımcı alanından tasarruf edilmiş olur. SC filtreleri elektronik olarak ayarlanabilir [1], [2]. Bununla beraber, İ.K.'lerin sınırlı bandgenişliklerinden ötürü, bu tür filtrelerin ses frekansından yüksek frekanslardaki davranışı iyi değildir.

Dijital filtrelerde de; hesaplama zamanı, analog-dijital dönüştürme zamanı ve özellikle Nyquist oranı gibi sınırlamalar bulunmaktadır.

Sürekli zaman analog filtrelerinde, SC ve dijital filtrelerdeki gibi sınırlamalar yoktur ve teorik olarak tüm frekans bandında çalışabilirler.

İ.K.' ler oldukça iyi geliştirilmiş ve ucuz elemanlar oldukları için uzun zamandır aktif filtre tasarımında kullanılmaktadır. Bununla birlikte, analog ve diğer işaret işleme devrelerinin aynı silikon kırkık üzerinde tümleştirilmesi isteği, OTA gibi daha uygun kazanç elemanlarının ortaya çıkmasına neden olmuştur [1].

OTA monolitik tümleştirme tekniği ile kolayca gerçekleştirilebilir, elektronik olarak ayarlanabilir ve tasarım kolaylığı sağlar [1]. OTA elemanı ile oluşturulan filtrelerde, OTA iletkenlik kazancı tasarım parametresi olarak kullanılabilir [3]. Ayrıca OTA, CMOS teknolojisine uygun bir elemandır. Böylece OTA kullanarak, CMOS teknolojisinin bipolar teknolojisine olan üstünlüklerinden (küçük kırık alanı, ekonomi vb.) yararlanmak mümkün olmaktadır. Bir ucu topraklı kapasite elemanı da CMOS teknolojisi ile kolayca gerçekleştirilebilir ve OTA ile beraber uyumlu bir eleman çifti oluşturur. Ayrıca topraklı kapasite elemanı, pekçok paralel parazitik kapasiteyi yok eder.

Genel olarak bir filtre transfer fonksiyonu, sadece OTA ve kapasite elemanı kullanarak gerçekleştirilebilir. Böylece İ.K. , direnç ve kapasite elemanlarıyla oluşturulan aktif-RC filtrelerine göre oldukça az sayıda elemanla filtre tasarlamak mümkün olmaktadır. OTA' nın bir diğer önemli yanı, yüksek frekans davranışının İ.K.' ye oranla çok üstün olmasıdır (MHz' ler mertebesinde).

Sözü edilen nedenlerden dolayı OTA ve topraklı kapasite elemanlarından oluşmuş devrelerin yüksek frekanslı sürekli zaman analog filtrelerde kullanımının oldukça yararlı olacağı görülecektir.

Mühendislikte, yüksek dereceden filtrelerin gerçekleştirilmesi oldukça önemlidir. Öte yandan s-domenindeki ifadesi,

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

biçiminde verilen gerilim transfer fonksiyonunun aktif-RC devreleri ile gerçekleştirilmesi konusunda literatürde oldukça yoğun çalışmalar bulunmaktadır. Bu çalışmalar 5 ana grupta toplanabilir [1], [4] :

- 1) Direkt (Doğrudan) Tasarım Yöntemi
- 2) Enduktans Benzeşim Yöntemi
- 3) Kaskad Biquadratik Tasarım Yöntemi

- 4) Frekansa Bağlı Negatif Direnç Yöntemi (FDNR)
- 5) Çok Çevrimli Geribesleme (Multi-loop Feedback , MF) Yöntemi.

Öte yandan literatürde, Quad olarak bilinen tasarım yöntemi, GMF (Genelleştirilmiş MF) yönteminin özel bir hali olup aşağıda verilen avantajlara sahiptir , [4] :

- (i) Yöntem geneldir ve (1.1) denklemi ile verilen en genel filtre transfer fonksiyonuna uygulanabilir.
- (ii) Kaskad yöntemi hariç, diğer tüm yöntemlere göre oldukça kolaydır.
- (iii) Bilinen diğer yöntemlere oranla daha az ya da aynı mertebede duyarlılığa sahiptir.
- (iv) İleri besleme ve toplama devrelerine gerek duymamaktadır.
- (v) Filtrenin ayarı kolaydır.
- (vi) Ekonomiktir.

Öte yandan Kaynak [9]' la verilen tasarım yöntem, Quad yönteminin sadece özel bir hali olup temelde hiçbir farklı yanı bulunmamaktadır.

Bu tezde amaç, yüksek dereceden aktif filtre tasarımını sadece OTA-C elemanları kullanarak başarmaktır. Quad tasarım yönteminin diğer tüm yöntemlere olan üstünlükleri gözönüne alınmış ve tasarımda kullanılacak yöntem olarak seçilmiştir.

Tezde, Quad yöntemi ve OTA-C devrelerinin avantajlı yönleri birleştirilerek; tümleştirme tekniğine uygun, yüksek frekans davranışı iyi, ayarı kolay ve eleman sayısı bakımından az devreler elde edilmesi amaçlanmaktadır.

Bu çalışmada tasarım sadece simetrik bandgeçiren filtreler için yapılmıştır. Bu nedenle çalışma "Genel OTA-C Quad Tasarımı" na bir giriş niteliğindedir.

Bu anlamda, Butterworth, Chebyshev ve Eliptik türden yaklaşıklık yöntemleri kullanılarak, verilen bandgeçiren (BG) filtre özelliklerinin karşılayan filtrenin tasarım denklemleri Quad yöntemiyle elde edilmekte ve OTA-C elemanlarıyla devre gerçekleştirilmektedir.

Bölüm 2' de, Quad yöntemi Kaynak [4]' ten yararlanarak anahatlarıyla açıklanmıştır.

Bölüm 3' te , Butterworth, Chebyshev ve Eliptik simetrik BG filtrelerin Quad yöntemiyle tasarım denklemlerinin elde edilmesi incelenmektedir ve örnek tasarımlara yer verilmektedir.

Bölüm 4' te, OTA elemanı tanıtılmakta ve OTA-C elemanları ile oluşturulan ikinci derece biquadları incelenmektedir. Verilen biquadlar içinde en avantajlı olanların Quad tasarımda temel biquadratik alt devre olarak kullanılması amaçlanmaktadır.

Bölüm 5' te, Quad yönteminin bilinen dinamik davranış incelemesinden yararlanarak simetrik BG OTA-C Quad filtrelerinin dinamik davranışı incelenmektedir. Bu şekilde filtredaki biquadratik altdevrelere ilişkin kazanç katsayıları elde edilmekte ve elde edilen sonuç tasarım denklemleri ile OTA ve kapasite elemanlarının değerleri elde edilmektedir.

Bölüm 6' da, tasarlanan filtrenin girişine uygulanabilecek maksimum gerilimin genliği hesaplanmaktadır. Böylece tasarlanan filtrenin "clipping" ve "slew-rate" sorunlarının yok edilmesi amaçlanmaktadır.

Ek' te ise, Butterworth, Chebyshev ve Eliptik türden simetrik BG filtrelerin tasarımını Quad yöntemiyle yaparak OTA ve C elemanlarıyla gerçekleyen bir bilgisayar programı yer almaktadır.

BÖLÜM 2. QUAD YÖNTEMİ

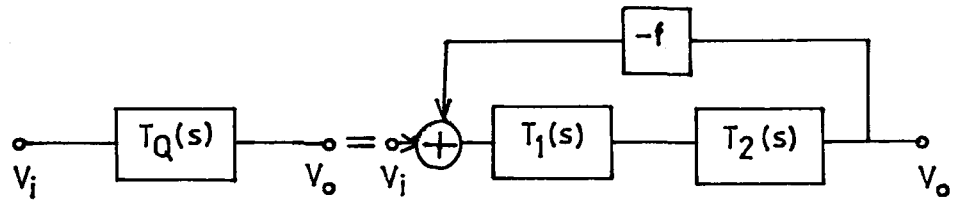
GİRİŞ

Bu bölümde, Çok Çevrimli Geribesleme (MF) Yönteminin özel bir hali olan ve literatürde Quad olarak bilinen yöntem kaynak [4] ' den yararlanarak ana hatlarıyla tanıtılacaktır. Tezin daha sonraki bölümlerinde yüksek dereceden OTA-C filtre tasarımında bu yöntemden yararlanılacaktır.

Quad yöntemi az duyarlıklı, kolayca ayarlanabilen, uygulaması, özellikle simetrik filtreler için, çok basit, ekonomik, iyi davranış gösteren ve en genel kararlı bir transfer fonksiyonunun gerçekleştirilmesine olanak veren bir yöntem olarak bilinir [4].

2.1. QUAD ALT DEVRE YAPISI

Yöntemin esasını oluşturan quad alt devresi, birbirine kaskad olarak bağlanmış iki biquadratik alt devre ve bunlara uygulanan bir negatif geribesleme ile oluşturulmaktadır [4]. Devre yapısı Şekil 2.1 ' de gösterilmiştir.



Şekil 2.1. Quad Devre Yapısı

Bu devreye ait gerilim transfer fonksiyonu,

$$T_Q(s) = \frac{T_1(s).T_2(s)}{1 + fT_1(s).T_2(s)} \quad (2.1)$$

olarak ifade edilebilir. Bu ifadede,

$$T_i(s) = \frac{(b_{i2}s^2 + b_{i1}s + b_{i0})}{(s^2 + a_{i1}s + a_{i0})} \quad (2.2.a)$$

yada

$$T_i(s) = \frac{k_i(s^2 + (w_{zi}/Q_{zi})s + w_{zi}^2)}{(s^2 + (w_{pi}/Q_{pi})s + w_{pi}^2)} \quad (2.2.b)$$

biçiminde yazılabilir [4].

Dördüncü dereceden quad devresine ilişkin gerilim transfer fonksiyonunun en genel ifadesi,

$$T_Q(s) = \frac{b_4s^4 + b_3s^3 + b_2s^2 + b_1s + b_0}{s^4 + a_3s^3 + a_2s^2 + a_1s + a_0} \quad (2.3)$$

şeklindedir. Ayrıca Şekil 2.1' deki devreye ait denklem aşağıdaki gibi yazılabilir.

$$T(s) = T_1(s).T_2(s) = \frac{T_Q(s)}{1 - fT_Q(s)} \quad (2.4)$$

Denklem (2.3) den de yararlanarak aşağıdaki eşitlikler bulunabilir.

$$T(s) = T_1(s).T_2(s) = \frac{1}{1 - fb_4} \frac{b_4s^4 + b_3s^3 + b_2s^2 + b_1s + b_0}{(s^4 + a_{m3}s^3 + a_{m2}s^2 + a_{m1}s + a_{m0})} \quad (2.5)$$

$$a_{m3} = \frac{a_3 - fb_3}{1 - fb_4} \quad (2.6.a)$$

$$a_{m2} = \frac{a_2 - fb_2}{1 - fb_4} \quad (2.6.b)$$

$$a_{m1} = \frac{a_1 - fb_1}{1 - fb_4} \quad (2.6.c)$$

$$a_{m0} = \left(\frac{a_0 - fb_0}{1 - fb_4} \right)^{1/2} \quad (2.6.d)$$

Diğer taraftan denklem (2.6) yardımı ile aşağıdaki ifadeler elde edilebilir.

$$\frac{a_3 - fb_3}{1 - fb_4} = \frac{w_1}{Q_1} + \frac{w_2}{Q_2} \quad (2.7.a)$$

$$\frac{a_2 - fb_2}{1 - fb_4} = \frac{w_1 \cdot w_2}{Q_1 \cdot Q_2} + w_1^2 + w_2^2 \quad (2.7.b)$$

$$\frac{a_1 - fb_1}{1 - fb_4} = w_1 \cdot w_2 \left(\frac{w_1}{Q_2} + \frac{w_2}{Q_1} \right) \quad (2.7.c)$$

$$\left(\frac{a_0 - fb_0}{1 - fb_4} \right)^{1/2} = w_1 \cdot w_2 \quad (2.7.d)$$

Denklem (2.7) de geçen w_1 , w_2 , Q_1 , Q_2 sırasıyla w_{p1} , w_{p2} , Q_{p1} , Q_{p2} ye eşittir. Aynı denklemde yer alan a_i ve b_i lerin bilindiği düşünülürse, bu denklem takımının çözümünün tek olmadığı görülür. Bu yüzden değişkenlerden biri bağımsız olarak seçilmelidir.

2.1.1. Quad Devresinin Gerçeklenmesi

Geribesleme katsayısı f 'nin bilindiği ve gerçel bir sayı olduğu varsayımı ile denklem (2.7) den yararlanarak, biquadratik alt devrelere ait w_1 , w_2 , Q_1 ve Q_2 değerleri aşağıdaki gibi hesaplanabilir.

$$w_1 = \sqrt{\frac{y \pm \sqrt{y^2 - 4a_{m0}^2}}{2}} \quad (2.8.a)$$

$$w_2 = \frac{a_{m0}}{w_1} \quad (2.8.b)$$

Burada $y > 2a_{m0}$ olmalı ve aşağıdaki denklemi sağlamalıdır [4].

$$y^3 - a_{m2}y^2 + (a_{m1}a_{m3} - 4a_{m0}^2)y + (4a_{m2}a_{m0}^2 - a_{m1}^2 - a_{m0}^2a_{m3}^2) = 0 \quad (2.8.c)$$

İyilik faktörleri ise,

$$Q_1 = \frac{w_1^4 - a_{m0}^2}{w_1(w_1^2 a_{m3} - a_{m1})} \quad (2.9.a)$$

$$Q_2 = \frac{(w_1^4 - a_{m0}^2)a_{m0}}{w_1(w_1^2 a_{m1} - a_{m3}a_{m0}^2)} \quad (2.9.b)$$

şeklinde hesaplanabilir. Özel bir durum olarak $a_{m1} = a_{m0}a_{m3}$ koşulu sağlanıyorsa, denklem (2.8.c) den $y = 2a_{m0}$ olarak elde edilir. Diğer bir deyişle, ya $w_1 = w_2$ ya da $Q_1 = Q_2$ dir. Her iki durum ayrı ayrı gözönüne alınarak incelenirse [4],

(i) $w_1 = w_2$: Bu durumda radyal frekanslar,

$$w_1 = w_2 = \sqrt{a_{m0}} \quad (2.10.a)$$

iyilik faktörleri ise,

$$Q_1; Q_2 = \frac{a_{m1}}{2(a_{m2} - 2a_{m0})\sqrt{a_{m0}}} (1 \pm \sqrt{1 - \frac{4a_{m0}^2(a_{m2} - 2a_{m0})}{a_{m1}^2}}) \quad (2.10.b)$$

denklemleri ile ifade edilir.

(ii) $Q_1 = Q_2$: Bu durumda iyilik faktörleri

$$Q_1 = Q_2 = Q = \sqrt{\frac{(2a_{m0} + a_{m2}) + \sqrt{(a_{m2} + 2a_{m0})^2 - 4a_{m0}a_{m3}^2}}{2a_{m3}^2}} \quad (2.11.a)$$

şeklindedir. Radyal frekanslar ise,

$$w_1, w_2 = \frac{Q \cdot a_{m3} \pm \sqrt{a_{m3}^2 Q^2 - 4a_{m0}}}{2} \quad (2.11.b)$$

olarak hesaplanır.

Öte yandan yukarıda incelenen $a_{m1}=a_{m0} \cdot a_{m3}$ koşulu ile birlikte $a_{m3}^2=4(a_{m2}-2a_{m0})$ koşulu da sağlanıyorsa, w_i ve Q_i ifadeleri oldukça basitleşerek aşağıdaki biçimde yazılabilir [4].

$$w_1 = w_2 = \sqrt{a_{m0}} \quad (2.12.a)$$

$$Q_1 = Q_2 = \frac{2\sqrt{a_{m0}}}{a_{m3}} \quad (2.12.b)$$

Her iki koşulun birlikte sağlanması durumunda, alt devrelere ait radyal frekansların ve iyilik faktörlerinin birbirlerine eşit olmasının yanı sıra (2.10.b) ve (2.11.a) denklemleri ile belirlenen iyilik faktörleri hesaplanarak gösterilebilir ki; iyilik faktörleri de en küçük değerlerini alırlar. İyilik faktörleri küçük olan devrelerin, genellikle, az duyarlıklı devreler olduğu göz önüne alınırsa, bu durum bir avantaj sağlar [4].

2.1.2. Quad Devresine İlişkin Geribesleme Katsayısının Bulunması

Geribesleme katsayısının belirlenmesinde, eşit radyal frekans ve/veya eşit iyilik faktörlerinin elde edilmesi durumu gözönüne alınacaktır. Bu biçimde tasarım, özellik bakımından birbirlerine yakın, üretimi kolay, kolayca ayarlanabilen ve az duyarlıklı filtreler gerçekleştirilmesine olanak sağlayacaktır. Hem radyal frekansların hem de iyilik faktörlerinin eşit olduğu duruma karşı düşen, yani, aşağıdaki denklemleri aynı anda sağlayabilen geribesleme katsayısı bulunabildiğinde, ilgilenilen devre için optimum çözüm elde edilmiş olur [4].

$$a_{m1} = a_{m3} \cdot a_{m0} \quad (2.13.a)$$

$$a_{m3}^2 = 4(a_{m2} - 2a_{m0}) \quad (2.13.b)$$

Fakat (2.13) denklemlerinin aynı anda sağlanabilmesi en genel durumda olası değildir. Buna karşılık, bu ifadelerin herbiri için ayrı ayrı geribesleme katsayıları bulunabilir. En genel durumda, her iki koşul birlikte sağlanamıyor ise, bu koşulları ayrı ayrı sağlayan geribesleme katsayılarından, isteğe göre ya daha az duyarlıklı devreyi gerçekleştiren, ya da daha iyi dinamik davranış gösteren, ya da iyilik faktörleri daha küçük olan tasarıma karşı düşeni, geribesleme katsayısı olarak alınır ve devre bu değere uygun olarak gerçekleştirilir. Diğer yandan, alçak geçiren filtrelerden türetilmiş, band geçiren filtrelerde, sözkonusu koşulların birlikte sağlanabileceği geribesleme katsayısı kolaylıkla bulunabilir [4].

Bu koşullar geribesleme katsayısının bulunması açısından incelenecek olursa,

(i) $a_{m1}=a_{m3} \cdot a_{m0}$ koşulundan yararlanarak geribesleme katsayısının belirlenmesi:

Bu koşulu sağlayan geribesleme katsayısı(ları) aşağıdaki denklemin çözümünden bulunabilir.

$$\frac{a_1 - fb_1}{a_3 - fb_3} = \sqrt{\frac{a_0 - fb_0}{1 - fb_4}} \quad (2.14)$$

Ancak bu denklem, $b_1=b_3=b_4=b_0=0$ olduğu durumlarda kullanılmaz, bu durumda geribesleme katsayısı (ii) durumundan yararlanarak bulunur. Diğer taraftan, geribesleme katsayısı, denklem (2.13.a) yı indirgeyerek elde edilen,

$$(b_0b_3^2 - b_1^2b_4)f^3 + (2a_1b_1b_4 + b_1^2 - a_0b_3 - 2a_3b_3b_0)f^2 + (2a_0a_3b_3 + a_3^2b_0 - 2a_1b_1 - a_1^2b_4)f + (a_1^2 - a_0a_3^2) = 0 \quad (2.15)$$

denklemini yardımı ile hesaplanabilir.

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

Bu durumda geribesleme katsayısı,

$$\frac{(a_3 - fb_3)^2}{(1 - fb_4)^2} = 4 \left(\frac{a_2 - fb_2}{1 - fb_4} - 2 \sqrt{\frac{a_0 - fb_0}{1 - fb_4}} \right) \quad (2.16)$$

denklemini yardımı ile belirlenebilir. Geribesleme katsayıları ise, bu denklemin indirgenmiş hali olan,

$$\begin{aligned} &(A - 64b_0b_4^3)f^4 + (2AB + 64a_0b_4^3 + 192b_0b_4^2)f^3 \\ &(B^2 + 2AC - 192b_4(b_0 + b_4^2))f^2 \\ &(2BC + 192b_4a_0 + 64b_0)f + (C^2 - 64a_0) = 0 \end{aligned} \quad (2.17)$$

$$\begin{aligned} A &= 4b_2b_4 - b_3^2 \\ B &= 2a_3b_3 - 4b_2 - 4a_2b_4 \\ C &= 4a_2 - a_3^2 \end{aligned} \quad (2.18)$$

denkleminin çözümünden belirlenebilir [4].

Özetle, 4. dereceden en genel biçimdeki gerilim transfer fonksiyonu, Quad topolojisi kullanılarak şu adımlarla tasarlanabilir [4].

Denklem (2.14) veya denklem (2.16) dan yararlanarak devrenin geribesleme katsayıları belirlenir.

Belirlenen her bir geribesleme değerine karşı düşen iyilik faktörleri ve radyal frekansları hesaplanır.

Elde edilen tasarım denklemleri karşılaştırılarak, içlerinden ya en az duyarlıklı olan, ya en iyi dinamik davranışı gösteren, ya da iyilik faktörü küçük olan tasarım denklemleri seçilerek devre gerçekleştirilir.

2.2. EN GENEL İKİNCİ DERECE DENKLEMİNDEN, ALÇAK GEÇİREN BAND GEÇİREN DÖNÜŞÜMÜ YARDIMIYLA ELDE EDİLEN, 4. DERECEDEN BAND GEÇİREN FİLTRELERİN TASARIMI

4. dereceden band geçiren filtre gerilim transfer fonksiyonunun,

$$T_{LP}(p) = \frac{b_2p^2 + b_1p + b_0}{p^2 + a_1p + a_0} \quad (2.19)$$

ikinci derece denkleminde,

$$p = \frac{s^2 + w_0^2}{Bs} \quad (2.20)$$

dönüşümü ile elde edildiği varsayılın. Bu anlamda, 4. dereceden filtrenin transfer fonksiyon ifadesi,

$$T(s) = \frac{\beta_2 s^4 + \beta_1 B s^3 + (2w_0^2 \beta_2 + \beta_0 B^2) s^2 + \beta_1 B w_0^2 + \beta_2 w_0^4}{s^4 + \alpha_1 B s^3 + (2w_0^2 + \alpha_0 B^2) s^2 + \alpha_1 B w_0^2 s + w_0^4} \quad (2.21)$$

biçiminde yazılabilir. Transfer fonksiyonunun paydasının katsayılarının aşağıdaki eşitliği sağladığı görülmektedir [4].

$$\alpha_1 = \alpha_3 \sqrt{\alpha_0} \quad (2.22)$$

Böylelikle, denklem (2.14) yardımıyla, geribesleme katsayısı $f=0$ olarak bulunur. Geribesleme olmadığından, bu durum doğrudan kaskad tasarıma karşı düşer [4].

Quad bloğunun diğer geribesleme katsayıları ise, denklem (2.16) dan yararlanarak belirlenir. (2.21) ile verilen transfer fonksiyonu için, denklem (2.16),

$$(\beta_1^2 - 4\beta_0\beta_2)f^2 - 2(\alpha_1\beta_1 - 2\beta_0 - 2\beta_2\alpha_0)f + (\alpha_1^2 - 4\alpha_0) = 0 \quad (2.23.a)$$

eşitliği ile basite indirgenebilir. Bu denklemin köklerinden geribesleme katsayıları,

$$f = \frac{(\alpha_1\beta_1 - 2\beta_0 - 2\beta_2\alpha_0) \pm 2\sqrt{(\beta_0 - \alpha_0\beta_2)^2 + \alpha_0\beta_1^2 - \alpha_1\beta_1(\beta_0 + \beta_2\alpha_0) + \beta_0\beta_2\alpha_1^2}}{(\beta_1^2 - 4\beta_0\beta_2)} \quad (2.23.b)$$

biçiminde hesaplanabilir. Diğer yandan, denklem (2.16) kullanıldığında $w_1=w_2$, $Q_1=Q_2$ olduğu ve değerlerinin denklem (2.12) yardımıyla belirlendiği gözönüne alınarak,

$$w_1 = w_2 = \sqrt{\alpha_{m0}} = w_0 \quad (2.24.a)$$

$$Q_{Q1} = Q_{Q2} = \frac{2(1 - fb_2)w_0}{B(a_1 - fb_1)} \quad (2.24.b)$$

değerleri kolaylıkla hesaplanabilir. Bu durumda verilen filtrenin tasarımında kullanılacak, ikinci dereceden alt devrelerin transfer fonksiyonu paydaları,

$$D_1(s) = D_2(s) = s^2 + \frac{B(a_1 - fb_1)}{2(1 - fb_2)}s + w_0^2 \quad (2.25)$$

olarak yazılabilir [4].

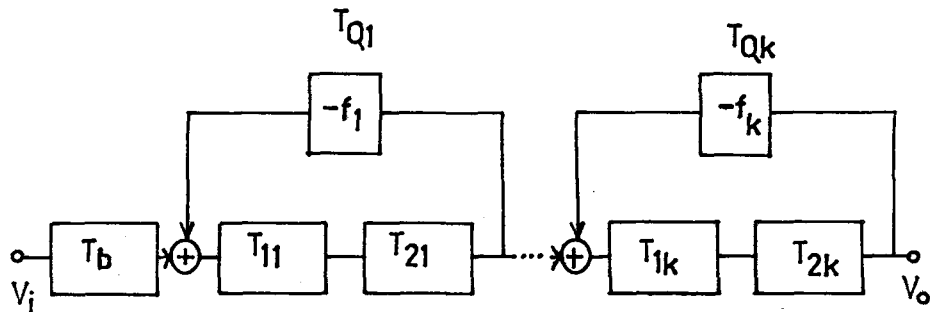
Aynı transfer fonksiyonu alınarak kaskad tasarım yöntemi için iyilik faktörleri ise,

$$Q_{c1} = Q_{c2} = \sqrt{\frac{4w_0^2 + a_0B^2 + \sqrt{(4w_0^2 + a_0B^2) - 4w_0^2a_1^2B^2}}{2B^2a_1^2}} \quad (2.26)$$

eşitliği ile belirtilebilir [4].

2.3. YÜKSEK DERECEDE FİLTRELERİN QUAD YÖNTEMİ İLE TASARIMI

Bu alt bölümde incelenecek olan filtrelerin derecesi, genelliği bozmadan çift sayı olarak alınacaktır. Filtrenin blok diyagramı, şekil 2.2 ' de verilmiştir. Şekilde, T_b olarak gösterilen alt devre, 2. dereceden bir aktif-RC devresi ya da kazancı birim olan bir bağlantı devresine karşılık düşmektedir [4].



Şekil 2.2. Yüksek dereceden Quad devre topolojisi

Transfer fonksiyonu,

$$T(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (2.27)$$

ile verilen, yüksek dereceden bir aktif filtrenin Quad yöntemi ile gerçekleştirilmesi aşağıdaki gibi özetlenebilir [4]:

- Transfer fonksiyonunun sıfır ve kutupları belirlenir.
- Sıfır ve kutuplar rastgele düzenlenir (bu konuda bir kural olmamakla beraber, Bölüm 2.3.1 de anlatılacak yöntem uygulandığında örneklerde az duyarlılık tasarımı için iyi sonuç vermiştir. Ayrıca daha önce verilen bazı yöntemlere de ters düşmemektedir [4]).
- Filtrenin derecesine göre, T_b 2.derece transfer fonksiyonu belirlenir ve gerçekleştirilir.
- Geride kalan ifade kullanılarak, 4. dereceden $T_{q1}, T_{q2}, \dots, T_{qk}$ transfer fonksiyonları belirlenir.
- 4.dereceden quad devreleri gerçekleştirilir.
- Olası bütün sıfır-kutup düzenlemesi için yukarıdaki işlemler tekrarlanır.
- Elde edilen tüm devreler, iyilik faktörleri, duyarlılık ya da dinamik davranış açısından karşılaştırılır.
- En küçük iyilik faktörü, en az duyarlılık olması ya da en iyi dinamik davranış göstermesi yönüyle, aranan özelliği sağlayan tasarım seçilir ve gerçekleştirilir.

2.3.1 Yüksek Dereceden Filtrelerin Tasarımında Quad Yöntemi İçin Sıfır Kutup Düzenlemesi

Burada Quad tasarım yöntemi için önerilen yöntem, kaskad tasarım yöntemine de uygulanabilmektedir. Yöntem aşağıdaki gibi özetlenebilir [4].

(i). Quad bloklarına ilişkin payda çok-terimlilerinin elde edilmesi

Bütün kutuplar, en büyük eğimli olanından başlanarak, hemen hemen aynı eğime sahip olacak biçimde, ikiye gruplara ayrılır (en büyük eğime sahip olan kutup çifti, birinci çift olarak adlandırılacaktır). Böylece bir araya getirilmiş kutup çiftlerinden yararlanarak, quad transfer fonksiyonlarının payda çok-terimlileri

oluşturulur. Geride kalan kutup ya da kutuplar yardımıyla da, T_b transfer fonksiyonunun paydası oluşturulur [4].

(ii). Quad blokları pay çok-terimlilerinin elde edilmesi

Bu aşamada, transfer fonksiyonunun, gerçel sıfırları olmadığı varsayılın. Diğer yandan, gerçel sıfırlar varsa, bunlar aşağıdaki işlemlerin bitirilmesinden sonra, geride kalan payda çok terimlileriyle uygun biçimde gruplandırılabilir. Transfer fonksiyonu band geçiren (band durduran) filtreye karşılık düşüyorsa, bütün sanal sıfırlar IG ile adlandırılan bir grup altında toplanır. Diğer yandan, kompleks sıfırlar da ayrı bir CG grubu altında gruplanır. IG grubu içinden rastgele iki değer seçilip bunların çarpımı $m_{zij}=w_{zi}w_{zj}$ ($i=1, 2, \dots, nI$ $j=1, 2, \dots, nI$) biçiminde oluşturulur. Buradaki nI değeri, IG de bulunan eleman sayısını göstermektedir. Benzer biçimde, tüm olası sıfır çiftleri için bu çarpım sonuçları bulunur. Böylece, $nI(nI-1)/2$ değer hesaplanabilir [4].

Payda çok terimlilerinin belirlenmesinde bulunan kutup çiftlerinden yararlanarak, $m_{pl}=w_{p1l}w_{p2l}$ (burada, $l=1, 2, \dots, k$ ve k kullanılan Quad sayısını göstermektedir) çarpımları hesaplanır. En büyük eğimli kutup çiftleri (başka bir deyimle en yüksek iyilik faktörü olan kutup çiftleri) için $l=1$ alınacaktır [4].

$l=1$ değerinden başlayarak, m_{pl} ve m_{zij} değerleri karşılaştırılır. Bunların içinden farklarının mutlak değeri ($|m_{pl}-m_{zij}|$) en küçük olanları bir arada toplanır. Benzer biçimde, tüm l değerleri için aynı işlem, geri kalan m_{pl} ve m_{zij} değerleri için tekrarlanır. Diğer yandan, aynı l değeri için birbirlerine 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, 1. Quad blok kutuplarıyla birleştirilir. IG grubu içinde herhangi bir sıfır daha kalmışsa, bu sıfırı CG içine aktararak, yukarıdaki işlemler, CG grubu için de tekrarlanır [4].

Örnek 2.1

Aşağıdaki 6. dereceden gerilim transfer fonksiyonunun verildiğini varsayalım [4].

$$T(s) = \frac{s(s^2 + 2.25)(s^2 + 2.25)}{(s^2 + 2s + 1.01)(s^2 + 0.09s + 0.83)(s^2 + 1s + 1.18)} \quad (2.28)$$

Denklem (2.28) ' e ilişkin frekans cevabı, Şekil 2.3 ' te gösterilmiştir. Aslında, Quad devresinde yer alan biquadratik sıfır-kutup sıralarının değiştirilmesi, geribesleme katsayılarının belirlenmesine etki etmez [4]. Bu örnek yardımıyla, bir önceki bölümde sunulan, sıfır-kutup yöntemi tartışılacak ve tasarımın devrenin duyarlılığına etkisi gözden geçirilecektir.

Önceki bölümde sunulan yöntemden yararlanarak, sıfır, kutup düzenlemesi yapılırsa, en az duyarlıklı Quad tasarım denklemleri aşağıdaki gibi ifade edilebilir [4].

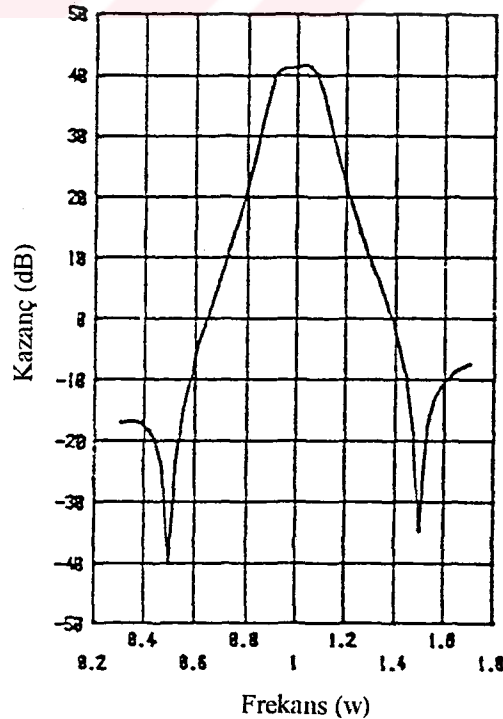
$$T_1(s) = \frac{s}{s^2 + 2s + 1.01} \quad (2.29.a)$$

$$T_2(s) = \frac{1.033302(s^2 + 2.25)}{s^2 + 1.008287s + 1.059091} \quad (2.29.b)$$

$$T_3(s) = \frac{s^2 + 2.25}{s^2 + 0.094041825s + 0.9851202} \quad (2.29.c)$$

$$f_1 = 0.0322286$$

Bu tasarım denklemlerine karşı düşen filtre Kaynak [4] ' te verilen Tow-Thomas (TT) biquadratik alt devresiyle gerçekleştirilmiştir. Bu anlamda filtrenin duyarlılığının hesaplanmasında bu alt devreden yararlanılmaktadır.



Şekil 2.3. (2.28) Denkleminin frekans cevabı

Gerçekleştirilen filtrenin $w_1=8$; $w_2=1.2$ frekans bandı için hesaplanan transfer fonksiyonu duyarlılığı, $S(\text{Quad}_1)=.01326541$ dir. Öte yandan, denklem (2.29.b) ve denklem (2.29.c) deki sıfırların yer değiştirmeleri ile elde edilecek ikinci bir Quad tasarım devresi, aynı frekans bandında $S(\text{quad}_2)=.01338386$ değerinde duyarlılık vermektedir [4].

Diğer yandan, en az duyarlıklı olarak tasarlanan Quad devresinin iyilik faktörü, 10.977123 değerindedir. Bu değer kaskad tasarımda elde edilen en büyük iyilik değeri ile karşılaştırıldığında, kaskad tasarımından elde edilen 10.86278 değerindeki iyilik faktöründen biraz büyüktür. Başka bir deyimle, en az duyarlıklı quad tasarımı, en küçük değerde iyilik faktörü değerine sahip değildir. Buna karşın aradaki fark önemsenmeyebilir. İstenirse, en küçük iyilik faktörü değerine sahip quad tasarımı da yapılabilir. Bu anlamda en küçük Q değerini veren quad tasarım denklemleri [4],

$$T_1(s) = \frac{s}{s^2 + 2s + 1.01} \quad (2.30.a)$$

$$T_2(s) = \frac{1.029256(s^2 + 25)}{s^3 + 100797s + 1.059211} \quad (2.30.b)$$

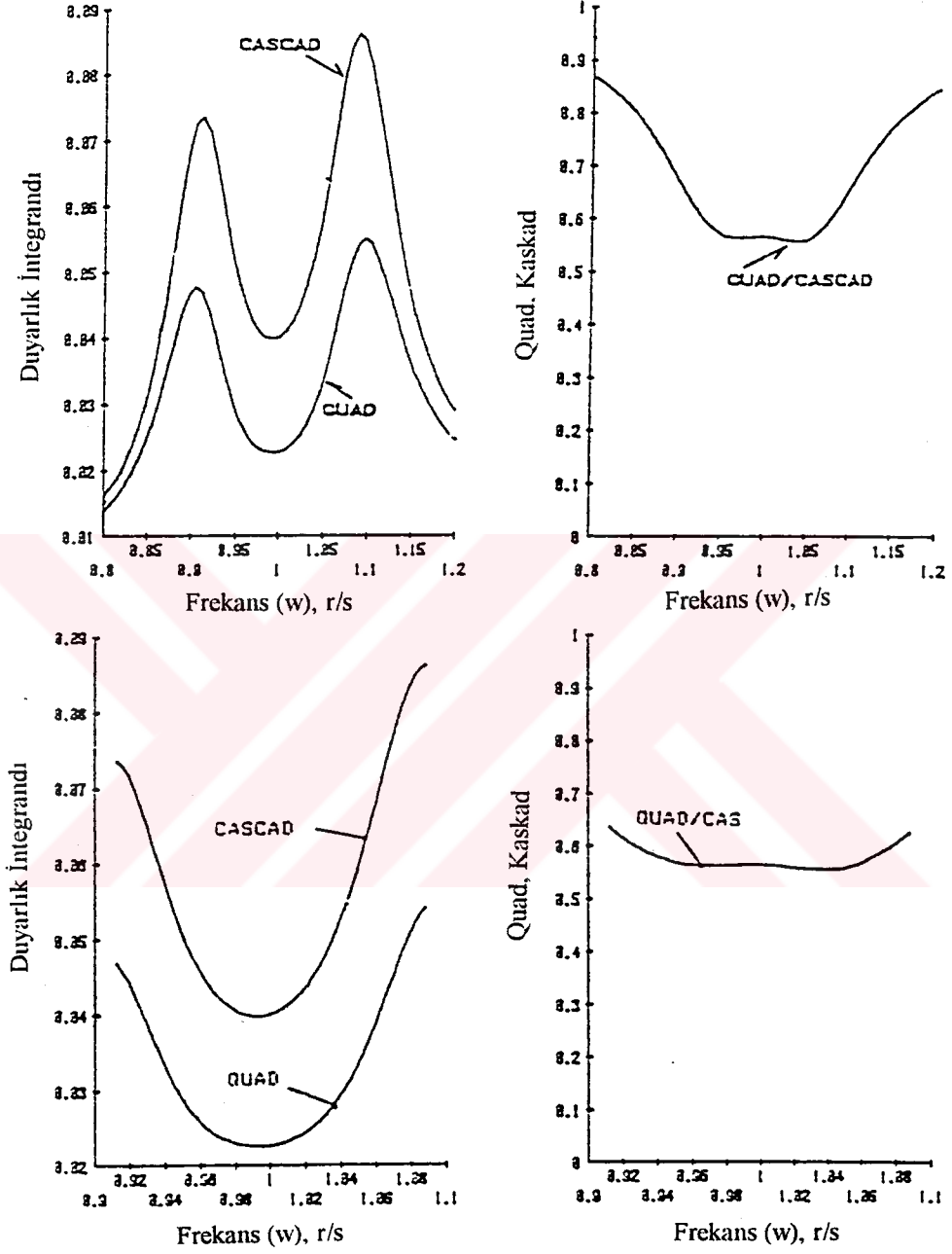
$$T_3(s) = \frac{s^2 + 2.25}{s^2 + 094761638s + 9361656} \quad (2.30.c)$$

$$f_2=.02842411$$

biçiminde ifade edilebilir.

Tasarım denklemleri yukarıda verilen Quad devresi, 10.210427 değerinde iyilik faktörüne sahiptir. Yeni tasarlanan Quad devresinin duyarlık analizi aynı band için ($w_1=8$; $w_2=1.2$) yapıldığında, .01399732 değeri elde edilir ki; bu değer en az duyarlıklı tasarım için elde edilen .0136541 değerine çok yakındır. Sonuç olarak, bu filtre tasarımı için Quad yöntemi kullanıldığında, en az duyarlıklı ve en küçük iyilik faktörü olan tasarım bir arada elde edilemez. Buna karşılık, herhangi bir kritere göre seçilen tasarım denklemleri, optimum değere çok yakın duyarlık (ya da iyilik faktörü) vermektedir [4].

Her iki Quad tasarımı ve kaskad tasarıma ilişkin duyarlık integrandlarının frekansa göre değişimleri Şekil 2.4. ' te görülmektedir.



Şekil 2.4. Örnek 2.1' deki filtreye ilişkin duyarlık karşılaştırmaları

BÖLÜM 3. SİMETRİK BANDGEÇİREN FİLTRELERİN QUAD YÖNTEMİYLE TASARIM DENKLEMLERİNİN ELDE EDİLMESİ

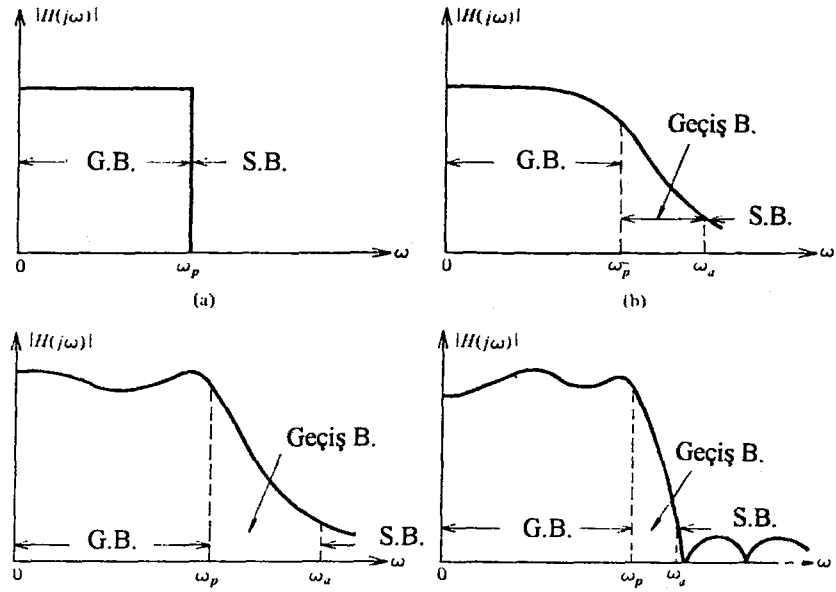
GİRİŞ

Mühendislik uygulamalarında kullanılan devrelerin çoğunda amaç elektriksel işaretleri işlemektir. Böyle işaret işleyen devreler filtre olarak isimlendirilir. Böylece, mühendislikte her devre bir filtre olarak düşünülebilir. Bir filtrenin girişine bir işaret uygulandığında belirli bir frekans bandı geçirilirken diğer bandlar istenilen değere göre zayıflatılır. Bir filtrenin giriş-çıkış frekans karakteristiği bu filtrenin transfer fonksiyonundan elde edilir ve önemi büyüktür.

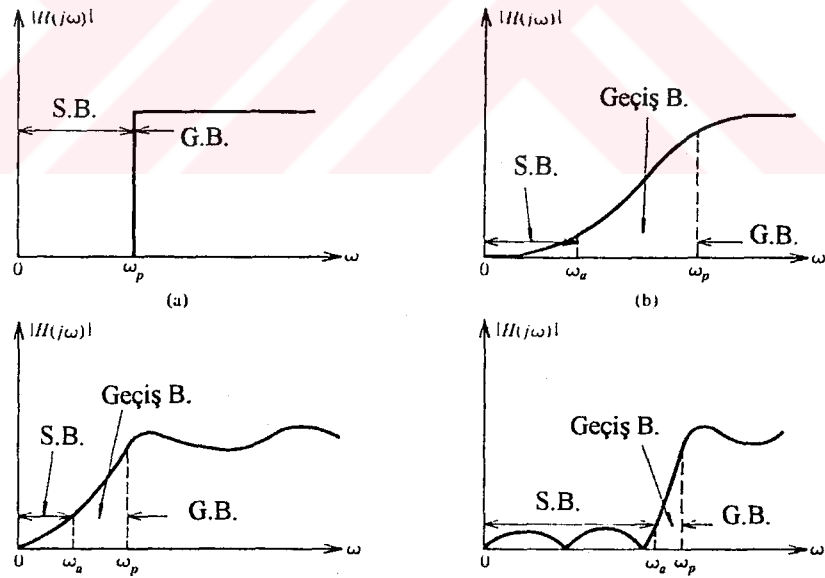
3.1. FİLTRELERİN SINIFLANDIRILMASI

Uygulamaların çoğunda dört temel filtre kullanılır : Alçak geçiren (AG), yüksek geçiren (YG), band geçiren (BG), band söndüren (BS) Bu filtreler için tipik frekans karakteristikleri Şekil 3.1, Şekil 3.2, Şekil 3.3 ve Şekil 3.4 ' te verilmiştir. Bu şekillerdeki ilk karakteristikler ideal filtre karakteristiğidir ve pratikte gerçekleştirilmeleri imkansızdır. Bununla birlikte bu karakteristiklerin çeşitli yaklaşıklık metodlarından yararlanarak pratikte gerçekleştirilmeleri mümkündür.

Şekillerden de görüleceği gibi frekans cevabının belirtilen bir değerin altına düştüğü frekans bandına zayıflatma ya da söndürme bandı denir. Aynı şekilde zayıflatmadan geçen frekansların bandına da geçirme bandı denir. her dört şekil için ikinci, üçüncü ve dördüncü karakteristikler sırasıyla Butterworth, Chebyshev ve Eliptik karakteristikler olarak adlandırılır [7].

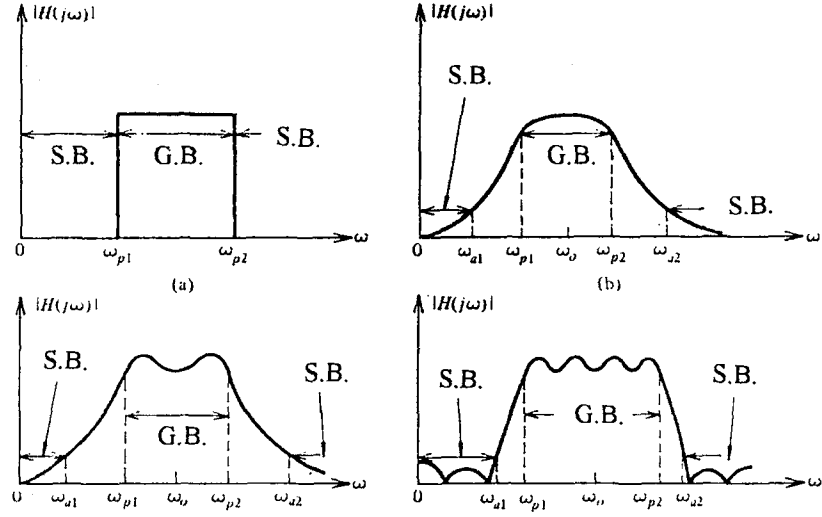


Şekil 3.1. AG filtre karakteristikleri

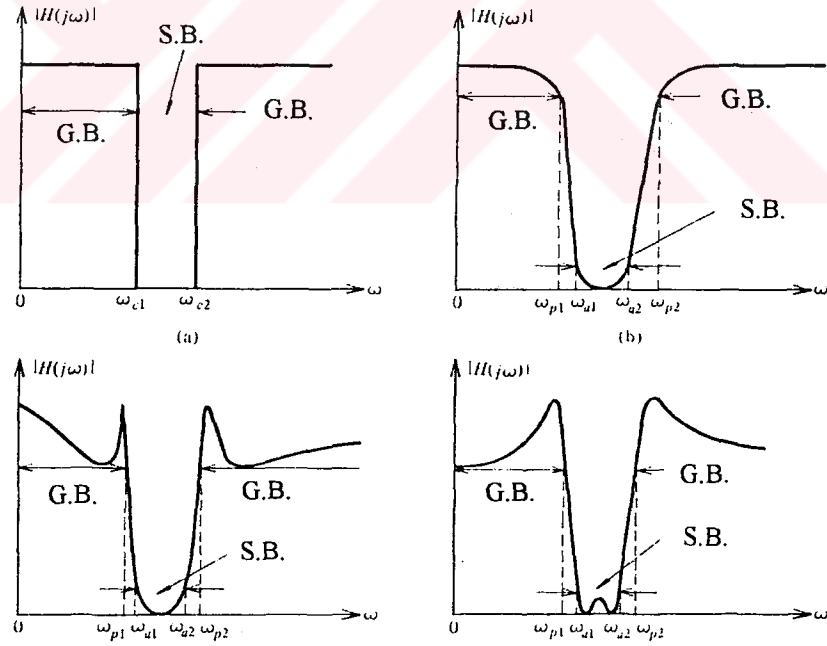


Şekil 3.2. YG filtre karakteristikleri

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



Şekil 3.3. BG filtre karakteristikleri



Şekil 3.4. BS filtre karakteristikleri

3.2. YAKLAŞIKLIK SORUNU

Bir filtre tasarımında ilk aşamada, verilen özellikleri (specifications) sağlayan transfer fonksiyonu bulunur. Çoğu filtre tasarımında bu özellikler genellikle desibel cinsinden verilir. Verilen özelliklerle AG, YG, BG, BS fonksiyonları öncelikle eşdeğer özelliklere sahip prototip AG filtre fonksiyonuna dönüştürülür. Prototip AG fonksiyon elde edildikten sonra frekans dönüşümü kullanılarak bu fonksiyon, diğer tip filtre karakteristik ve özelliklerini karşılayan fonksiyonlara dönüştürülebilir [7].

Prototip AG filtre için özellikler şunlardır :

1. Maksimum geçirme bandı zayıflatması ,

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

2. Minimum söndürme bandı zayıflatması ,

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

Yaklaşıklık sorunu, verilen özellikleri sağlayan $H(p)$ rasyonel fonksiyonunu üç klasik yaklaşımla (Butterworth, Chebyshev, Eliptik) bulmak olarak bilinir [7].

İzleyen bölümlerde her üç yaklaşıklık yöntemine kısaca değinilecektir. Konu hakkında ayrıntılı bilgi literatürde, [1], [7], [10], [11], bulunabilir.

3.2.1. Butterworth Yaklaşıklığı

Prototip AG karesel genlik fonksiyonu ,

$$|H(j\Omega)|^2 = T(\Omega^2) = \frac{H_0^2}{1 + \varepsilon^2 \Omega^{2N}} \quad (3.1)$$

şeklindedir. $|H(j\Omega)|$, $\Omega=0$ da H_0 maksimum değerini alır. ε büyüklüğü salınım (ripple) faktörü olarak bilinir. N ise filtre transfer fonksiyonunun , $H(s)$, derecesidir. ε ve N , $H(s)$ ' in kutup ve sıfırları bulunmadan önce elde edilmesi gereken büyüklüklerdir. ε ve N ' yi bulmak için verilen özelliklerden yararlanılır.

Önce ,

$$A(\Omega) = 10 \log(1 + \varepsilon^2 \Omega^{2N}) - 20 \log(H_0) \quad (3.2)$$

zayıflatması bulunur. H_0 değeri kazanç sabiti olarak bilinir. $H_0=1$ seçilirse,

$$A(\Omega) = 10 \log(1 + \varepsilon^2 \Omega^{2N}) \quad (3.3)$$

olur. Normalize filtre tasarımında $\Omega_p=1$ r/s alıp,

$$\varepsilon = \sqrt{(10^{0.1A_p} - 1)} \quad (3.4)$$

elde edilir. Ω_a ve üzerindeki frekanslar için minimum söndürme bandı zayıflatması,

$$A_a \leq 10 \log(1 + \varepsilon^2 \Omega_a^{2N}) \quad (3.5)$$

şeklindedir. (3.4) ifadesi (3.5) de yerine konursa filtre derecesi ,

$$N \geq \frac{\log((10^{0.1A_a} - 1)/(10^{0.1A_p} - 1))}{2 \log(\Omega_a)} \quad (3.6)$$

eşitsizliği ile elde edilir. Burada N bir tamsayıdır. (3.1) ifadesinde $\Omega=p/j$ konursa, elde edilen karakteristik denklemin kökleri $T(-p^2)$ nin kutupları olmak zorundadır ve,

$$p_k = R.(\cos(\frac{\pi}{2} + \frac{2k-1}{N} \frac{\pi}{2}) + j \sin(\frac{\pi}{2} + \frac{2k-1}{N} \frac{\pi}{2})) , \quad k=1,2,...,N \quad (3.7)$$

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

Böylece transfer fonksiyonu ,

$$H(p) = \frac{H_0}{\prod_{k=1}^N (p - p_k)} \quad (3.8)$$

şeklinde bulunur. Şimdi (3.8) kullanılarak normalize (prototip) filtre fonksiyonu bulunabilir. Gerçekleme açısından, karmaşık eşlenik kutupları birleştirmemiz gerekir. Böylece prototip AG filtre fonksiyonu,

$$H(p) = \frac{H_0}{D_0(p) \cdot \prod_{i=1}^{N_1} (p^2 + A_i p + B_i)} \quad (3.9.a)$$

olarak bulunur. Burada,

$$D_0(p) = \begin{cases} p + \sigma_0 & , \quad N \text{ tek} \\ 1 & , \quad N \text{ çift} \end{cases} \quad (3.9.b)$$

ve

$$N_1 = \begin{cases} (N-1)/2 & , \quad N \text{ tek} \\ N/2 & , \quad N \text{ çift} \end{cases} \quad (3.9.c)$$

tir. Ayrıca A_i , B_i ve H_0 için,

$$A_i = -2 \operatorname{Re}(p_i) \quad , \quad B_i = |p_i|^2$$

$$H_0 = \begin{cases} \sigma_0 \prod_{i=1}^{N_1} B_i & , \quad N \text{ tek} \\ \prod_{i=1}^{N_1} B_i & , \quad N \text{ çift} \end{cases} \quad (3.9.d)$$

ifadeleri verilebilir [7].

3.2.2. Chebyshev Yaklaşıklığı

Bu yaklaşıklıkta karesel genlik fonksiyonu ,

$$|H(j\Omega)|^2 = T(\Omega^2) = \frac{H_0^2}{1 + \varepsilon^2 C_N^2(\Omega)} \quad (3.10)$$

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

$$C_N(\Omega) = \begin{cases} \cos(N \cos^{-1}(\Omega)) & , \quad 0 \leq \Omega \leq 1 \\ \cosh(N \cosh^{-1}(\Omega)) & , \quad \Omega \geq 1 \end{cases} \quad (3.11)$$

ifadesi ile tanımlanır.

Prototip AG filtre için tekrar $\Omega = 1$ r/s kabul ediyoruz. (3.11) ifadesi (3.10) 'da yerine konursa, $\Omega = 1$ r/s için ,

$$A_p = 10 \log(1 + \varepsilon^2) \quad (3.12)$$

bulunur. Burada $H_0 = 1$ kabul edilmiştir. Ayrıca Ω_a ve üzeri frekanslar için ,

$$A_a \leq 10 \log(1 + \varepsilon^2 \cosh^2(N \cosh^{-1} \Omega_a)) \quad (3.13)$$

olur. N için bu eşitsizlik çözülürse ,

$$N \geq \frac{\cosh^{-1}(((10^{0.1A_a} - 1)/(10^{0.1A_p} - 1))^{0.5})}{\cosh^{-1} \Omega_a} \quad (3.14)$$

elde edilir. ε ve N bulunduktan sonra artık transfer fonksiyonunun kutupları bulunabilir. Bunun için önce ,

$$v = \frac{1}{N} \sinh^{-1}\left(\frac{1}{\varepsilon}\right) \quad (3.15)$$

bulunur. O halde kutuplar ,

$$p_k = -\sin\left(\frac{2k-1}{N} \cdot \frac{\pi}{2}\right) \cdot \sinh(v) + j \cos\left(\frac{2k-1}{N} \cdot \frac{\pi}{2}\right) \cdot \cosh(v), \quad k=1,2,\dots,N \quad (3.16)$$

şeklinde ifade edilir.

Yine aynı şekilde eşlenik kutuplar birleştirilirse transfer fonksiyonu (3.9) formuyla aynı olur [7]. Sadece H_0 için ifade,

$$H_0 = \begin{cases} \sigma_0 \prod_{i=1}^{N_1} B_i, & N \text{ tek} \\ 10^{-0.05A_p} \prod_{i=1}^{N_1} B_i, & N \text{ çift} \end{cases}$$

şeklinde olacaktır.

3.2.3. Eliptik Yaklaşıklık

Her üç yaklaşıklık içinde Eliptik yaklaşıklık en verimli olanıdır. Verilen özellikleri karşılayan bir filtre tasarımı için Eliptik yaklaşıklığı seçmek en akılcı olanıdır. Zira, en düşük dereceli transfer fonksiyonu elde edilmektedir. Böylece gerçekleştirilen devredeki eleman sayısı da en aza indirgenmiş olur. Bununla birlikte her üç yaklaşımın kendine göre avantajlı tarafı vardır. Örneğin geçirme bandında herhangi bir salınıma sahip olmasını istemediğimiz bir filtreyi, Butterworth yaklaşımı kullanarak tasarlamak gerekir.

Eliptik yaklaşım için prototip AG filtreye ait transfer fonksiyonu ,

$$H(p) = \frac{H_0}{D_0(p)} \cdot \prod_{i=1}^{N_1} \frac{p^2 + C_i}{p^2 + A_i + B_i} \quad (3.18)$$

şeklinde dir. Burada N_1 ve $D_0(p)$ ifadeleri (3.9.b) ve (3.9.c) denklemleri ile aynıdır. Ω_p ve Ω_a nın sırasıyla prototip AG filtrenin geçirme bandı ve söndürme bandı kenar frekansları olduğunu kabul edelim. A_p ve A_a ise sırasıyla

maksimum geirme bandı ve minimum sndürme bandı zayıflatmasıdır. Böylece aşağıda verilen algoritma derece ve katsayıları bulmakta kullanılabilir [7].

1. Seçicilik faktörü k hesaplanır ,

$$k = \frac{\Omega_p}{\Omega_a} \quad (3.19.a)$$

2. Aşağıdaki bağıntılar hesaplanır ,

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

$$q_0 = \frac{1}{2} \cdot \frac{1 - \sqrt{k'}}{1 + \sqrt{k'}}$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13}$$

$$D = \frac{10^{0.1A_a} - 1}{10^{0.1A_p} - 1}$$

(3.19.b)

3. Derece tamsayı olacak şekilde hesaplanır ,

$$N \geq \frac{\log 16D}{\log(1/q)}$$

(3.19.c)

4. Sırasıyla aşağıdaki yardımcı terimler hesaplanır ,

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

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

$$\sigma_0 = \frac{2q^{0.25} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sinh((2m+1)\lambda)}{1 + 2 \sum_{m=1}^{\infty} (-1)^m q^{m^2} \cosh(2m\lambda)}$$

$$W = (1 + k\sigma_0^2) \left(1 + \frac{\sigma_0^2}{k}\right)$$

(3.19.d)

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

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

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

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

$$B_i = \frac{v_i \sigma_0^2 + W \Omega_i^2}{(1 + \sigma_0^2 \Omega_i^2)^2}$$

$$A_i = \frac{2\sigma_0 \sqrt{v_i}}{(1 + \sigma_0^2 \Omega_i^2)}$$

(3.19.e)

Burada ,

$$\mu = \begin{cases} i \\ i - 0.5 \end{cases} \quad , N \text{ çift} \quad i=1,2,3,\dots,N_1$$

şeklindedir.

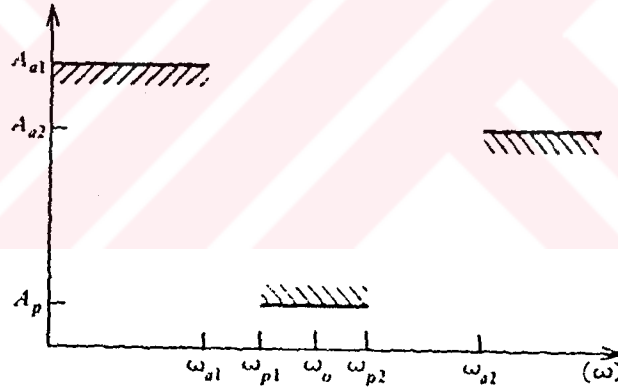
6. Minimum zayıflatma 0 dB olacak şekilde kazanç sabiti hesaplanır,

$$H_0 = \begin{cases} \sigma_0 \prod_{i=1}^{N1} (B_i/C_i) & , N \text{ tek} \\ 10^{-0.05A_p} \prod_{i=1}^{N1} (B_i/C_i) & , N \text{ çift} \end{cases} \quad (3.19.f)$$

Yukarıdaki algoritma bilgisayarla programlamaya oldukça elverişlidir. σ_0 ve Ω_i ifadeleri için 3 veya 4 terim çoğu uygulama için yeterli olmaktadır [7], [11].

3.3. VERİLEN ÖZELLİKLERİ SAĞLAYAN BAND GEÇİREN FİLTRE TRANSFER FONKSİYONUNUN HESAPLANMASI

Bu alt bölümde , verilen filtre özelliklerinden yola çıkarak simetrik BG filtre tasarımı ele alınacaktır. Sözü edilen özellikler Şekil 3.5 ' te gösterilen değerlerdir.



Şekil 3.5. BG filtre için özellikler.

Verilen özelliklere göre BG filtreye ait transfer fonksiyonu aşağıda verilen adımlarla bulunabilir [7] :

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

$$w_0 = \sqrt{w_{p2} \cdot w_{p1}}$$

$$B = w_{p2} - w_{p1}$$

bağıntıları yardımıyla band genişliği ve merkez frekans hesaplanır.

3.

$w_a = w_0^2/w_{a1}$ ifadesi hesaplanır.

4.

$w_a \leq w_{a2}$ ise $w_{a1} = w_{a1}$ ve $w_{a2} = w_a$ veya,

$w_a > w_{a2}$ ise $w_{a2} = w_{a2}$ ve $w_{a1} = w_0^2/w_{a2}$

olarak seçilir.

5. Seçilen w_{a1} ve w_{a2} ile ,

$$\Omega_a = \frac{w_{a2} - w_{a1}}{w_{p2} - w_{p1}} \quad (3.20)$$

değeri hesaplanır.

6. Verilen A_p ve önceki adımlardan bulunan A_a (adım1), Ω_a (adım 5) değerleri biliniyor. Butterworth, Chebyshev veya Eliptik yaklaşıklıklardan biri seçilir ve prototip AG filtre fonksiyonu, $H(p)$, oluşturulur.

7. Adım 2 ' de bulunan değerler kullanılarak ve

$$p = \frac{s^2 + w_0^2}{Bs} \quad (3.21)$$

dönüşüm formülünü $H(p)$ prototip AG fonksiyonuna uygulayarak gerçek özellikleri karşılayan BG filtre transfer fonksiyonu , $H(s)$, bulunur [7].

3.3.1. Alçak Geçiren - Band Geçiren (AG-BG) Dönüşümü

Aşağıda (3.9) ve (3.18) eşitliklerindeki prototip AG fonksiyonundan (3.21) dönüşüm formülü kullanılarak orijinal BG transfer fonksiyonunun elde edilişi gösterilmiştir.

1. Butterworth ve Chebyshev için :

(3.21) formülü (3.9)' da yerine konulduğunda, N tek ise,

$$\begin{aligned}
 H(s) &= \frac{H_0}{\left(\frac{s^2 + w_0^2}{Bs} + \sigma_0\right) \cdot \prod_{i=1}^{N1} \left(\left(\frac{s^2 + w_0^2}{Bs}\right)^2 + A_i \left(\frac{s^2 + w_0^2}{Bs}\right) + B_i\right)} \\
 &= H_0 \cdot \frac{Bs}{(s^2 + (B\sigma_0)s + w_0^2)} \cdot \frac{\prod_{i=1}^{N1} B^2 s^2}{\prod_{i=1}^{N1} (s^4 + (A_i B)s^3 + (2w_0^2 + B_i B^2)s^2 + (A_i B w_0^2)s + w_0^4)} \\
 &= H_0 \cdot T_b(s) \cdot \prod_{i=1}^{N1} T_{Qi}(s)
 \end{aligned} \tag{3.22.a}$$

bulunur. N çift ise bu ifade ,

$$= H_0 \cdot \prod_{i=1}^{N1} T_{Qi}(s) \tag{3.22.b}$$

şeklini alır. Buradaki $T_b(s)$ ve $T_{Qi}(s)$ ifadeleri Bölüm 2 'deki Şekil 2.2 ' de görülen yüksek dereceden Quad filtresindeki bloklara ilişkin transfer fonksiyonlarını göstermektedir. H_0 ise filtre girişindeki kazanç sabitidir.

2. Eliptik için :

(3.21) formülü (3.18)' de yerine konulduğunda, N tek ise ,

$$\begin{aligned}
 H(s) &= H_0 \cdot \frac{Bs}{(s^2 + (B\sigma_0)s + w_0^2)} \cdot \prod_{i=1}^{N1} \frac{(s^2 + (2w_0^2 + C_i B^2)s^2 + w_0^4)}{(s^4 + (A_i B)s^3 + (2w_0^2 + B_i B^2)s^2 + (A_i B w_0^2)s + w_0^4)} \\
 &= H_0 \cdot T_b(s) \cdot \prod_{i=1}^{N1} T_{Qi}(s)
 \end{aligned} \tag{3.23}$$

bulunur. N çift olması halinde bu ifade ,

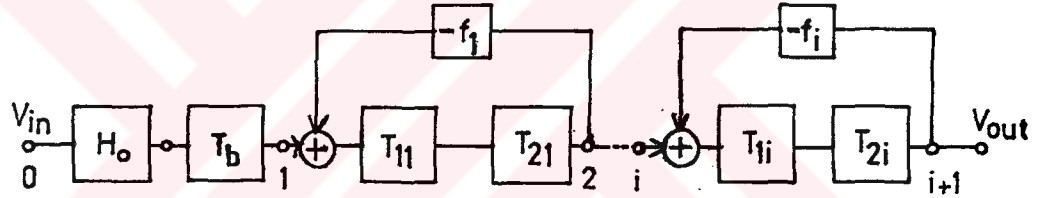
$$= H_0 \cdot \prod_{i=1}^{N/2} T_{Q_i}(s)$$

(3.24)

şeklini alır.

3.4. ELDE EDİLEN BG FİLTRE TRANSFER FONKSİYONU İLE QUAD TASARIM

3.3.1 alt bölümünde elde edilen (3.22) - (3.24) ifadelerinin Quad tasarım için uygun şekle getirildiği görülmektedir. Bu ifadelerden anlaşılacağı üzere filtre derecesi tek ise $T_b(s)$ bloğu filtre girişinde yer alacak, çift ise böyle bir blok olmayacaktır. Böylece her üç yaklaşıklık metodu için kullanılacak yüksek dereceden Quad filtresinin blok diyagramı Şekil 3.6 'daki gibi olacaktır.



Şekil 3.6. Butterworth, Chebyshev ve Eliptik türden BG filtreler için yüksek dereceden Quad devresi

Şunu önemle vurgulamak gerekir ki, Şekil 3.6 'daki diyagramdaki bloklar AG, YG, BG ya da Notch türü biquadlardan oluşabilir. Ancak simetrik BG filtrelerde durum daha basittir ve biquadların topolojileri aynıdır. Butterworth ve Chebyshev için tüm biquadlar BG iken Eliptikte (eğer varsa T_b bloğu hariç) tüm biquadlar notch türünden devrelerden oluşacaktır. Bu da tasarım kolaylığı sağlamaktadır. Bununla birlikte farklı topolojideki biquadlar için Bölüm 4 'te verilecek olan genel OTA-C biquad devresi kullanılarak istenen AG, YG, BG, BS filtre tiplerinden herhangi biri gerçekleştirilebilir.

3.4.1. Tasarım adımları

Bu tezde, simetrik BG türden OTA-C filtrelerin Bölüm 2 'de verilen Quad yöntemi ile tasarımı için aşağıdaki adımlar izlenecektir.

1. Öncelikle tasarlanacak BG filtrenin özelliklerine göre Butterworth, Chebyshev veya Eliptik yaklaşımlardan uygun olanı seçilir.

2. Verilen BG filtre özelliklerinden yararlanarak 3.3 alt bölümündeki algoritma yardımıyla, filtreye ilişkin $H(s)$ transfer fonksiyonu oluşturulur. $H(s)$ filtre transfer fonksiyonu, (3.23) veya (3.24) formlarından birine benzetilir.

3. Böylece dördüncü dereceden quadratik fonksiyonlardan yararlanarak Bölüm 2 ' de verilen klasik Quad tasarım yöntemine göre biquadratik transfer fonksiyonları ve geribesleme katsayıları elde edilir.

4. Elde edilen Quad blokları, filtre dinamik değişimi maksimum olacak şekilde sıralanır. Sıralama yapıldıktan sonra Bölüm 5 ' te verilecek prosedüre göre biquadratik bloklara ilişkin kazanç sabitleri ve yeni geribesleme katsayıları 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.

Öte yandan simetrik BG filtreler için geribesleme katsayıları (2.23.a) ya da (2.23.b) formüllerinden biri yardımıyla bulunabilir ,

1) Butterworth ya da Chebyshev için ,

$$f_i = B_i - \frac{A_i^2}{4} \quad (3.25)$$

2. Eliptik için ,

$$f_i = \frac{B_i + C_i - \sqrt{B_i^2 - C_i^2 + C_i A_i^2}}{2C_i} \quad (3.26)$$

şeklindedir.

Quad ve biquadratik blokların sıralanması filtre dinamik değişimini artırmak üzere şu şekilde yapılacaktır :

- Filtre girişindeki yüksek frekanslı gürültülerin etkisini yok etmek amacıyla band geçiren alt devreler filtrenin ilk devreleri olarak gerçekleştirilecektir [1], [4]. Derecenin tek olduğu durumda T_b alt devesi her üç yaklaşıklık için de girişe konmalıdır. Çünkü T_b band geçiren bir alt devredir.
- Kalan Quad ve biquadratik bloklar ise optimuma yakın şekilde sıralanacaktır. Optimuma en yakın ilk tahmin, biquadları artan değer katsayılarına göre sıralamaktır. Yani, $Q_1 < Q_2 < Q_3 \dots \dots \dots < Q_N$ olacak şekilde, en düzgün transfer fonksiyonu genliğine sahip biquad önce gelmek üzere biquadların sıralaması yapılacaktır. Böylece en az düzgün transfer fonksiyonu genliğine sahip biquad en sona konulacaktır [1].

Filtrenin dinamik değişimini artıracak şekilde tasarımın ele alınması Bölüm 5 'te ayrıca verilecektir.

Aşağıda her üç yaklaşıklıktan yararlanarak tasarlanan bazı filtre örnekleri verilmiştir. Örneklerdeki değerlerin tümü tezin sonundaki Ek 'te verilen "QUAD" adlı bilgisayar programı ile hesaplanmıştır.

Örnek 3.1.

$f_{a1}=1$ MHz, $f_{p1}=1.2$ MHz, $f_{p2}=1.6$ MHz, $f_{a2}=2$ MHz, $A_p < 0.5$ dB ve $A_a > 30$ dB özelliklerine sahip BG Butterworth filtre tasarlanacaktır. Bu filtre özellikleri ile 3.3 alt bölümündeki tasarım adımlarını kullanarak Quad ve kaskad tasarım denklemlerini elde edelim.

- $A_a=30$ dB ve $A_p=0.5$ dB dir.
- Verilen frekanslardan yararlanarak filtreye ilişkin merkez frekans ve band genişliği, sırasıyla $\omega_0=8.706237$ Mr/s ve $B=2.5132741$ Mr/s olarak bulunur.
- $\omega_{a2}=12.06372$ Mr/s ve $\omega_{a1}=6.283186$ Mr/s değerleri bulunur.
- (3.20) bağıntısı yardımıyla, $\Omega_a = 2.3$ değeri elde edilir. Ω_p ise daima 1 'e eşittir.
- Böylece prototip AG filtreye ilişkin Ω_p , Ω_a , A_p ve A_a değerleri elde edilmiş olur. (3.6) bağıntısı yardımıyla $N=6$ olarak bulunur. Bu değer ve (3.7) denklemi kullanılarak prototip AG filtreye ilişkin kutuplar,

$$p_1 = p_6^* = -0.3084094 \pm j1.150999$$

$$p_2 = p_5^* = -0.8425899 \pm j0.8425898$$

$$p_3 = p_4^* = -1.150999 \pm j0.3084092$$

prototip AG filtrenin katsayıları ,

$$A_1 = 2.301998 \quad , \quad B_1 = 1.419915$$

$$A_2 = 1.68518 \quad , \quad B_2 = 1.419915$$

$$A_3 = 0.6168187, \quad B_3 = 1.419915$$

ve kazanç sabiti ,

$$H_o = 2.862776$$

olarak bulunur. O halde prototip AG filtre (3.9) bağıntısına göre ,

$$H(p) = H_o \cdot \frac{1}{(p^2 + A_1 p + B_1)} \cdot \frac{1}{(p^2 + A_2 p + B_2)} \cdot \frac{1}{(p^2 + A_3 p + B_3)}$$

(3.27)

şeklinde olacaktır. Bu prototip AG filtreden AG-BG dönüşümü ile elde edilen transfer fonksiyonunun biçimi (3.22.b) denklemine benzeyecektir. Elde edilen dördüncü dereceden kuadratik transfer fonksiyonlarından , $T_{Qi}(s)$, Bölüm 2 ' de sözü edilen Quad tasarım yönteminin bağıntıları kullanılarak biquadratik ikinci dereceden transfer fonksiyonları elde edilir. Aşağıda filtreye ilişkin Quad ve kaskad tasarım denklemleri verilmiştir.

Quad Tasarım Denklemleri :

$$H_o = 2.862776$$

$$T_{11}(s) = \frac{2513275s}{s^2 + 2892777s + 7.579856 \times 10^{13}} , \quad T_{21}(s) = \frac{2513275s}{s^2 + 2892777s + 7.579856 \times 10^{13}}$$

$$Q_1 = 3.009647 \quad f_1 = 0.09511618$$

(3.28.a)

$$T_{12}(s) = \frac{2513275s}{s^2 + 2117660s + 7.579856 \times 10^{13}}, \quad T_{22}(s) = \frac{2513275s}{s^2 + 2117660s + 7.579856 \times 10^{13}}$$

$$Q_2 = 4.111254, \quad f_2 = 0.7099575$$

(3.28.b)

$$T_{13}(s) = \frac{2513275s}{s^2 + 775117.4s + 7.579856 \times 10^{13}}, \quad T_{23}(s) = \frac{2513275s}{s^2 + 775117.4s + 7.579856 \times 10^{13}}$$

$$Q_3 = 11.23215, \quad f_3 = 1.324799$$

(3.28.c)

Kaskad Tasarım Denklemleri :

$$T_{11}(s) = \frac{2513275s}{s^2 + 3023228s + 8.295774 \times 10^{13}}, \quad T_{21}(s) = \frac{2513275s}{s^2 + 2762326s + 6.925722 \times 10^{13}}$$

$$Q_3 = 3.0127117$$

(3.29.a)

$$T_{12}(s) = \frac{2513275s}{s^2 + 2375176s + 9.678552 \times 10^{13}}, \quad T_{22}(s) = \frac{2513275s}{s^2 + 1860143s + 5.936242 \times 10^{13}}$$

$$Q_2 = 4.141993$$

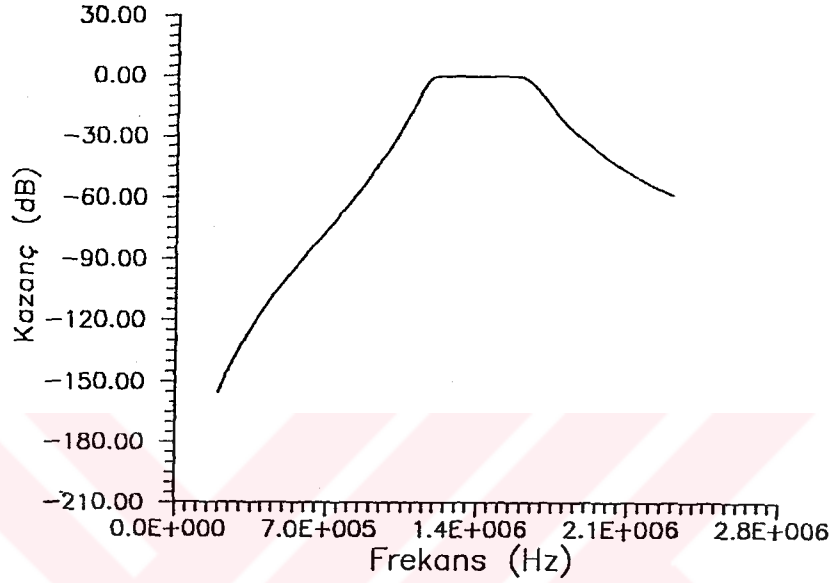
(3.29.b)

$$T_{13}(s) = \frac{2513275s}{s^2 + 902267.6s + 1.055464 \times 10^{14}}, \quad T_{23}(s) = \frac{2513275s}{s^2 + 647967.2s + 5.443504 \times 10^{13}}$$

$$Q_3 = 11.3864$$

(3.29.c)

Filtrenin frekans cevabı Şekil 3.7 ' de görülmektedir.



Şekil 3.7. Örnek 3.1 ' deki filtrenin frekans cevabı

Örnek 3.2.

$f_{a1}=100$ KHz , $f_{p1}=500$ KHz , $f_{p2}=550$ KHz , $f_{a2}=1.2$ MHz , $A_p < 0.5$ dB ve $A_a > 30$ dB özelliklerine sahip BG filtre için , Chebyshev yaklaşımı kullanılarak Quad ve Kaskad tasarım denklemleri elde edilmek isteniyor. "QUAD" programı kullanılarak aşağıdaki değerler elde edilmiştir.

Prototip AG filtrenin kutupları ve katsayıları ,

$$p_1 = p_2^* = -0.7128123 \pm j1.004042$$

$$A_i = 1.425625 , \quad B_i = 1.516202 , \quad i = 1$$

Quad Tasarım Denklemleri ,

$$H_o = 1.431388$$

$$T_{11}(s) = \frac{314159.3s}{s^2 + 223936.6s + 1.085657 \times 10^{13}} \quad , \quad T_{21}(s) = \frac{314159.3s}{s^2 + 223936.6s + 1.085657 \times 10^{13}}$$

$$Q_1 = 14.71368 \quad f_1 = 1.008101$$

(3.30)

(3.30) 'daki bağıntılardan anlaşılacağı üzere her iki biquadratik transfer fonksiyonu birbirinin aynıdır. Böylece filtre, özellik bakımından birbirinin aynı iki biquada sahip olmasından dolayı kolay ayar edilebilen, ekonomik ve az tasarım zamanı gerektiren bir devre olarak görünmektedir. Bu da Quad tasarımının temel amaçlarından biridir. Bununla birlikte Bölüm 5 'te de değinileceği gibi, dinamik değişimi iyileştirmek üzere yapılan tasarım sonucu biquadlara kazanç değerleri atandığında, (3.30) 'daki her iki fonksiyonun payları farklı olacaktır.

Kaskad Tasarım Denklemleri :

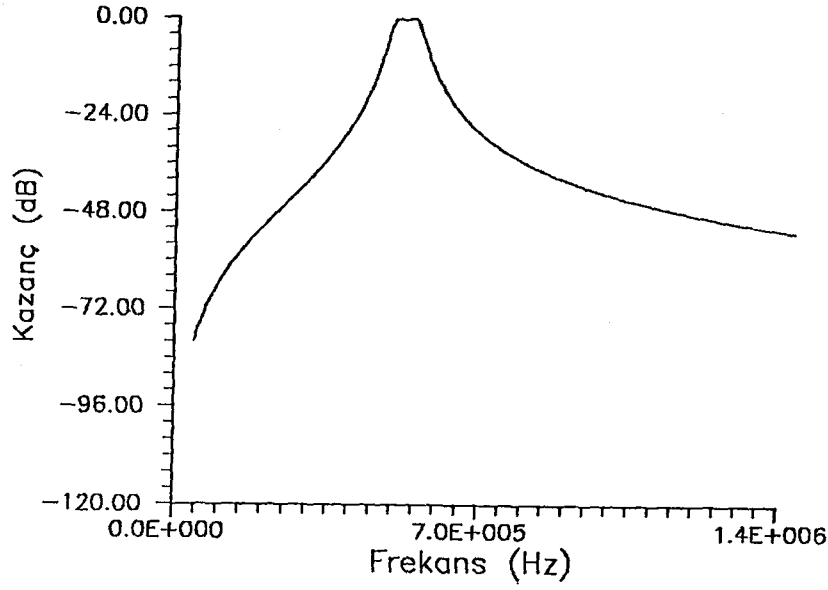
$$H_o = 1.431388$$

$$T_{11}(s) = \frac{314159.3s}{s^2 + 234649.4s + 1.194748 \times 10^{13}} \quad , \quad T_{21}(s) = \frac{314159.3s}{s^2 + 213223.8s + 9.865264 \times 10^{12}}$$

$$Q_1 = 14.73054$$

(3.31)

Tekrar ifade etmek gerekir ki, tasarımda dinamik davranış gözönüne alınmamış ve biquadlara kazanç değerleri atanmamıştır. Bu durum Bölüm 5 'te ele alınacaktır. Filtrenin frekans cevabı Şekil 3.8. 'de gösterilmektedir.



Şekil 3.8. Örnek 3.2 'deki filtrenin frekans cevabı

Örnek 3.3.

$f_{a1}=1$ MHz , $f_{p1}=1.2$ Mhz , $f_{p2}=2.4$ MHz , $f_{a2}=2.6$ MHz , $A_p < 0.5$ dB ve $A_a > 50$ dB özelliklerine sahip BG Eliptik filtrenin tasarım denklemleri elde edilmek isteniyor. "QUAD" programıyla aşağıdaki değerler elde edilmiştir.

Prototip AG filtrenin sıfır ve kutupları ,

$$z_1 = z_6^* = \pm j3.837859$$

$$z_2 = z_5^* = \pm j1.570306$$

$$z_3 = z_4^* = \pm j1.269106$$

$$p_1 = p_6^* = -0.3548209 \pm j0.3612536$$

$$p_2 = p_5^* = -0.1739308 \pm j0.8396773$$

$$p_3 = p_4^* = -0.04401241 \pm j1.007497$$

Prototip AG filtrenin katsayıları ,

$$A_1 = 0.7096418, \quad B_1 = 0.256402, \quad C_1 = 14.72916$$

$$A_2 = 0.3478616, \quad B_2 = 0.73531, \quad C_2 = 2.465861$$

$$A_3 = 0.08802482, \quad B_3 = 1.016986, \quad C_3 = 1.61063$$

$$H_o = 0.003094317$$

O halde Prototip AG filtre ,

$$H(p) = H_o \cdot \frac{(p^2 + C_1)}{(p^2 + A_1 p + B_1)} \cdot \frac{(p^2 + C_2)}{(p^2 + A_2 p + B_2)} \cdot \frac{(p^2 + C_3)}{(p^2 + A_3 p + B_3)}$$

(3.32.a)

şeklinde olacaktır. Quad Tasarım Denklemleri ,

$$H_o = 0.003094317$$

$$T_{11}(s) = \frac{1.008862s^2 + 1.061777 \times 10^{14}}{s^2 + 2698996s + 1.136978 \times 10^{14}}, \quad T_{21}(s) = \frac{s^2 + 1.228296 \times 10^{13}}{s^2 + 2698996s + 1.136978 \times 10^{14}}$$

$$Q_1 = 3.950699 \quad f_1 = 0.008784507$$

(3.32.b)

$$T_{12}(s) = \frac{1.391072s^2 + 4.565324 \times 10^{14}}{s^2 + 1824262s + 1.136978 \times 10^{14}}, \quad T_{22}(s) = \frac{s^2 + 3.938968 \times 10^{13}}{s^2 + 1824262s + 1.136978 \times 10^{14}}$$

$$Q_1 = 5.845058 \quad f_2 = 0.28113$$

(3.32.c)

$$T_{13}(s) = \frac{2.689523s^2 + 7.296977 \times 10^{14}}{s^2 + 892506.9s + 1.136978 \times 10^{14}}, \quad T_{23}(s) = \frac{s^2 + 4.764713 \times 10^{13}}{s^2 + 892506.9s + 1.136978 \times 10^{14}}$$

$$Q_3 = 11.94716 \quad f_3 = 0.6281869$$

(3.32.d)

Kaskad Tasarım Denklemleri ise ,

$$T_{11}(s) = \frac{s^2 + 1.05245 \times 10^{15}}{s^2 + 3016839s + 1.469782 \times 10^{14}} , \quad T_{21}(s) = \frac{s^2 + 1.228296 \times 10^{13}}{s^2 + 2333734s + 8.795317 \times 10^{13}}$$

$$Q_1 = 4.018596$$

(3.33.a)

$$T_{12}(s) = \frac{s^2 + 3.281875 \times 10^{14}}{s^2 + 1685224s + 2.043604 \times 10^{14}} , \quad T_{22}(s) = \frac{s^2 + 3.938968 \times 10^{13}}{s^2 + 937590.6s + 6.325687 \times 10^{13}}$$

$$Q_2 = 8.482827$$

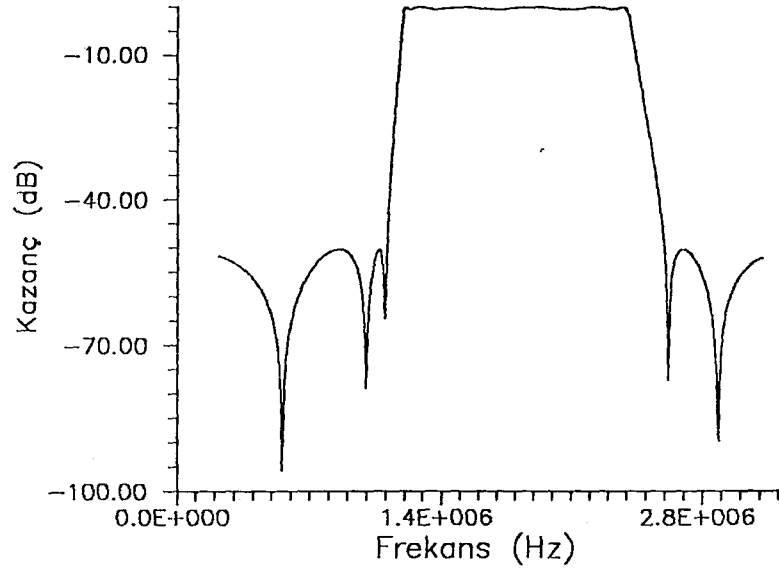
(3.33.b)

$$T_{13}(s) = \frac{s^2 + 2.713112 \times 10^{14}}{s^2 + 443207.8s + 2.28551 \times 10^{14}} , \quad T_{23}(s) = \frac{s^2 + 4.764713 \times 10^{13}}{s^2 + 220483.8s + 5.656156 \times 10^{13}}$$

$$Q_3 = 34.11019$$

(3.33.c)

Filtrenin frekans cevabı Şekil 3.9 ' da verilmiştir.



Şekil 3.9. Örnek 3.3 ' teki filtrenin frekans cevabı

BÖLÜM 4. OTA ELEMANI VE OTA - KAPASİTE ELEMANI İLE OLUŞTURULAN BİQUADLAR

Bu bölümde, önceki bölümde değinilen Quad tasarımda kullanılmaya elverişli OTA-C (OTA - Topraklı Kapasite) biquadları ele alınacaktır. Bu anlamda, literatürde yer alan bazı OTA-C biquadları tanıtılacak ve içlerinden en avantajlı olanlar (az sayıda eleman, kapasite elemanlarının topraklı olması, duyarlık vb.) incelenecek ve Quad tasarımdaki Quad bloklarına ilişkin biquadratik temel altdevre/ler olarak kullanılacaktır.

4.1. GİRİŞ

Tamamen tümleşik sürekli zaman filtreleri (Fully Integrated Continuous-Time Filters), gün geçtikçe devre tasarımcıları açısından önem kazanmakta ve bu alanda yoğun çalışmalar yapılmaktadır.

Sayısal filtrelerde çalışılan frekans aralığı, analog-sayısal dönüşüm ve Nyquist oranı ile belirli olan hesaplama zamanı nedeniyle sınırlıdır. Bununla birlikte, sürekli zaman filtreleri, mevcut teknolojilere nazaran, hemen tüm frekans aralığında kullanılabilir. Kapasite-Anahtarlamalı (Switched -Capacitor) Filtreler ses frekansı gibi düşük frekanslı uygulamalarda bir endüstri standardı haline gelmiştir. Bununla birlikte yüksek frekans uygulamalarında artan örnekleme frekansı ile birlikte anahtarlama, örnekle-tut vb. gibi işlemlerde sorunlar çıkmaktadır [12].

Ancak sürekli zaman domeninde kalarak bu türden sorunların önüne geçilebilir.

Sürekli zaman filtrelerinin monolitik tümleştirilmesi basit, tümleştirme aşamaları sistematik olduğu için, sözü edilen diğer filtrelere nazaran daha çok tercih edilmektedir [13].

Aktif filtrelerin pekçoğu İ.K.' lerle gerçekleştirilmelerine karşın, İ.K.'lerden oluşmuş filtrelerin uygulamalarının sınırlı kaldığı görülmektedir. Daha da önemlisi İ.K. kazancının frekansa bağlı olmasından dolayı, filtre davranışında ciddi bozulmalar oluşmakta ve bu da aktif filtre uygulamalarının alçak frekansla sınırlı kalmasını getirmektedir. Dahası, birkaç özel uygulama dışında, İ.K.'lerle gerçekleştirilen aktif-RC analog filtrelerinin monolitik tümleştirilmesinde pek başarılı olunamamıştır. Diğer anlatımla, İ.K.-RC filtrelerinin, diğer analog ve/veya sayısal devrelerle aynı silikon kırık üzerinde tümleştirmek için uygun bir yöntem bulunamamıştır [1].

Sözü edilen bu sakıncayı gidermek için çalışmalar daha doğal kazanç elemanlarına, OTA, yönlendirilmiştir. OTA ' lar genellikle İ.K.'lerden oldukça büyük band genişliklerine sahiptirler. Ayrıca, daha basit olmaları yanı sıra, bir kontrol akımının vasıtasıyla elektronik olarak ayar kolaylığı da sağlamaktadır [1]. Bu şekilde, OTA' nın ayarlanabilme özelliği kullanılarak OTA ve kapasite elemanlarıyla programlanabilir aktif filtreler, [14], [15], gerçekleştirilebilir.

OTA ve Kapasiteler, bütün tümleştirme teknolojilerinde kolayca gerçekleştirilebilmektedir. Ayrıca bu tür filtrelerin çevresindeki sayısal ayarlama devreleriyle uyumluluğu açısından, CMOS teknolojisi, [16], tümleştirme için doğal bir tercih olarak görünmektedir.

Son olarak, OTA ' larla oluşturulan analog filtreler genellikle İ.K.'li eşdeğerlerinden daha az sayıda eleman içermektedirler. Böylelikle silikon kırığının alanı da küçük kalmaktadır.

Bu özelliklerinden dolayı OTA ' lar gün geçtikçe aktif filtre tasarımında söz sahibi olmaya ve İ.K.'lerin yerini almaya başlamaktadırlar.

4.2. OTA ELEMANI VE MODELLERİ

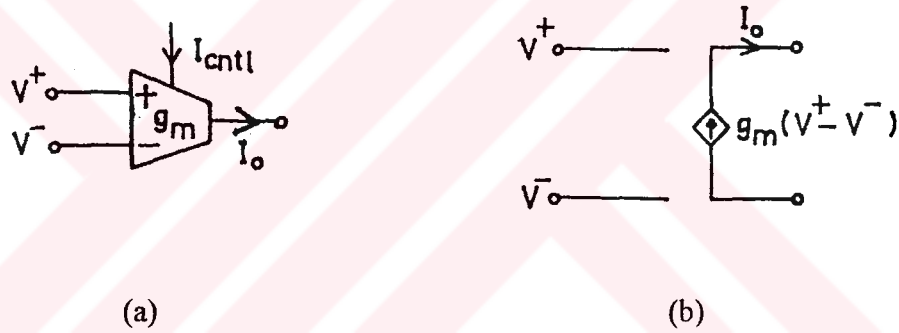
OTA elemanı, temel olarak gerilim kontrollü bağımlı akım kaynağıdır. Gerçek çalışma koşullarında eleman, idealden oldukça farklı davranışlar göstermektedir.

4.2.1. İdeal OTA Modeli

İdeal bir OTA' nın devre sembolü ve eşdeğer devresi Şekil 3.1' de gösterilmiştir. İdeal bir OTA, gerilim kontrollü akım kaynağıdır ve çıkış akımının ifadesi,

$$I_o = g_m(V^+ - V^-) \quad (4.1)$$

olarak tanımlanır. Şekil 4.1.b ' de gösterildiği gibi giriş ve çıkış empedansları sonsuzdur. Çoğu tasarımda g_m kontrol akımı, I_{cntl} akımının değiştirilmesi suretiyle ayarlanabilir bir değişkendir. Burada $g_m = k \cdot I_{cntl}$ şeklinde olup g_m ve I_{cntl} birbiriyle doğru orantılıdır [1].

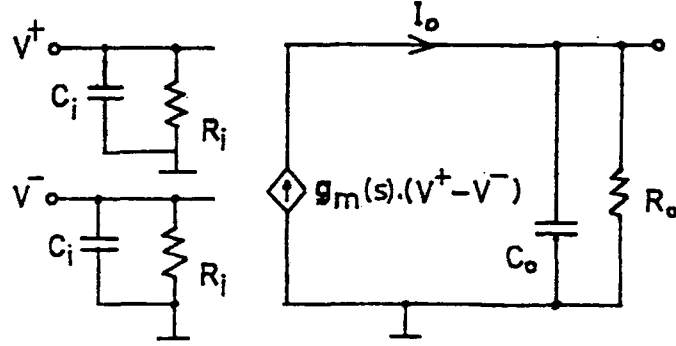


Şekil 4.1. OTA a) Devre sembolü b) İdeal eşdeğer devresi

I_{cntl} akımının kontrolü ile OTA kazancı doğrudan ayarlanabildiği gibi, aynı kontrol ucundan uygulanan bir gerilim ile de kontrol edilebilir [17].

4.2.2. Pratikte Kullanılan OTA Modelleri

Pratikte tasarımcı OTA elemanı ile çalışırken dikkatli olmak zorundadır. OTA tabanlı filtreler tasarlanırken, gerçek OTA 'nın giriş ve çıkış empedanslarının frekansa bağlı ve sonlu oldukları gözönünde bulundurulmalıdır. Aynı zamanda g_m de frekansla değişmektedir [1], [3]. Gerçek OTA modeli Şekil 4.2. ' de görülmektedir. Burada, eviren ve evirmeyen uçlardan toprağa olan empedansların eşit olduğu varsayılmaktadır ve giriş uçları arasındaki empedans önemsenmemektedir [1]



Şekil 4.2. İdeal olmayan OTA

Pek çok uygulamada OTA kazancının frekans bağımlılığı ve giriş ile çıkış kapasitelerinin varlığı nedeniyle, eleman davranışı bozulmaktadır. Bu da filtrenin bandgenişliğinin ve alçak frekans performansının azalmasının yanısıra kararsızlığa da neden olabilmektedir [8]. Dahası filtre davranışının arzu edilenden uzaklaşmasına yani filtrenin frekans cevabının şeklen önemli ölçüde bozulmasına yol açmaktadır.

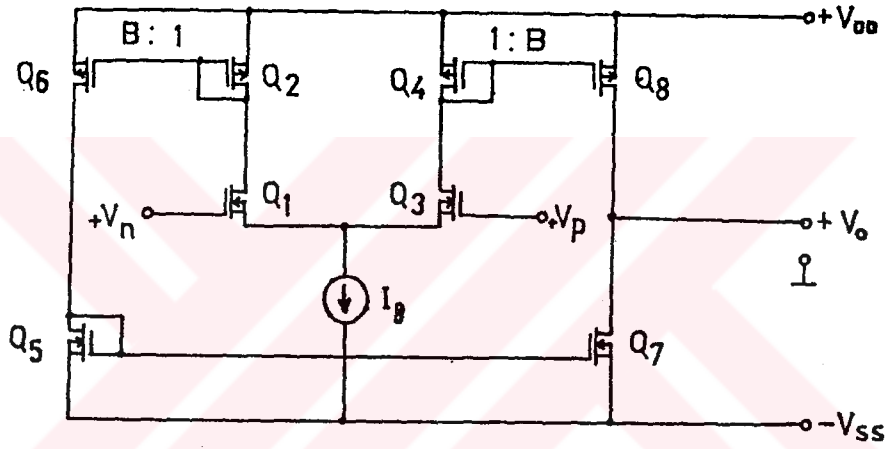
OTA 'nın giriş empedansı da filtre cevabı üzerinde önemli etkilere neden olabilmektedir. Bu etkiyi kesin olarak tahmin etmek mümkün olmamakla birlikte, OTA çıkışlarının Darlington yapılarla desteklenmesi pratik açıdan yeterlidir. Bu tür OTA kırmıklar ticari olarak mevcuttur. OTA kazancı (transkonduktansı veya eğimi) üzerinde ısının da büyük ölçüde etkisi vardır ve gözardı edilemez [18]. Mutlak sıcaklıkla doğrusal değişen akım kaynakları kullanılarak bu etkinin önüne geçilebilir [19].

Ticari amaçla sunulmuş OTA 'lar bipolar tekniğiyle üretilmişlerdir ve bazı pratik sınırlamalara sahiptirler. CA3080 elemanı bunlara bir örnek olarak verilebilir. CA3080 gibi OTA 'lar filtre tasarımında kullanıldıklarında filtre dinamik davranışında sorunlar çıkması muhtemeldir. Çünkü bu gibi OTA 'ların girişine uygulanabilecek gerilim sınırlı ($< 20 \text{ mV}$) olmaktadır. Bu nedenden ötürü CA3080 elemanı düşük maliyetli ancak dinamiğin öneminin az olduğu filtrelerde kullanılabilir [19]. Daha gelişmiş uygulamalarda ise daha iyi tasarlanmış OTA 'lar, [20], [21], kullanılmalıdır.

4.2.3. Simetrik CMOS OTA

Aktif filtre yapılarında kullanılmaya elverişli olan OTA yapılarından biri olan simetrik CMOS - OTA , geniş bantlı olması, eğiminin I_B kutuplama akımı ile kontrol edilebilmesi, yapısının tümleştirmeye uygun ve basit olması gibi nedenlerden dolayı yaygın bir kullanım alanı bulmakta, OTA-C filtre yapılarının yanısıra, analog çarpma devreleri ve yüksek frekans osilatörlerinin gerçekleştirilmesi amacıyla da bu devre yapısından yararlanılmaktadır [22], [23].

Bahsedilen simetrik CMOS-OTA yapısı Şekil 4.3 ' te görülmektedir.



Şekil 4.3. Simetrik CMOS-OTA yapısı.

Aşağıda simetrik CMOS-OTA ' ya ait bağıntılar verilmektedir [22],

OTA ' nın eğimi,

$$G_m = B \cdot \sqrt{K_n \cdot I_B \cdot (W/L)_1} \quad (4.2)$$

çıkış akımının maksimum değeri,

$$I_{o \max} = -I_{o \min} = -B \cdot I_B \quad (4.3)$$

gerilim kazancı,

$$K_v = G_m \cdot R_o \quad (4.4)$$

baskın kutbu,

$$f_d = \frac{1}{2 \cdot \pi \cdot R_o \cdot (C_{n5} + C_L)} \quad (4.5)$$

baskın olmayan kutupları,

$$f_{nd1} = \frac{gm_4}{2 \cdot \pi \cdot C_{n6}} \quad (4.6)$$

$$f_{nd2} = \frac{2 \cdot gm_5}{2 \cdot \pi \cdot C_{n8}} \quad (4.7)$$

kazanç-band genişliği çarpımı,

$$GBW = K_v \cdot f_d \quad (4.8)$$

yükselme eğimi,

$$YE = \frac{B \cdot I_B}{C_L + C_{n5}} \quad (4.9)$$

Bu bağıntılarında B büyüklüğü, $Q_4 - Q_8$ ve $Q_2 - Q_6$ tranzistorlarının W/L oranlarının birbirine oranını, I_B kutuplama akımını; g_{mi} , i numaralı tranzistorun eğimini, C_{nk} k numaralı düğüme gelen toplam kapasiteyi (W/L)_i i numaralı tranzistorun (W/L) oranını göstermektedir [22].

Bu yapı ile ilgili bazı deneysel (SPICE) incelemeleri 4.4 başlığı altında verilmektedir.

4.3. OTA VE KAPASİTE ELEMANLARIYLA OLUŞTURULAN BİQUADLAR

OTA ve kapasite elemanları kullanarak ikinci dereceden biquadratik altdevre oluşturmaya yönelik çok sayıda çalışma, [23], [24], bulunmaktadır. Bu alt bölümde literatürde karşılaşılan bazı OTA-C biquadlarına değinilecek ve ayrıldıkları noktalar vurgulanacaktır. İçlerinden en uygun olanları simetrik bandgeçiren

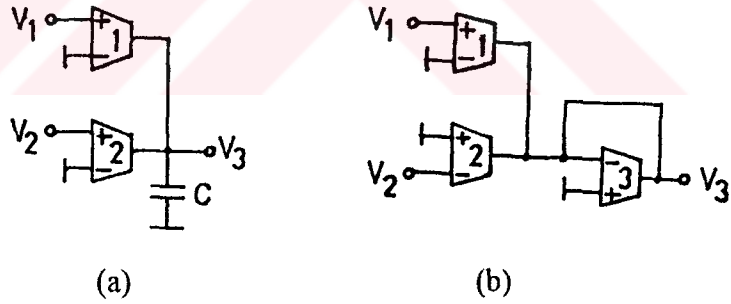
filtreleri gerçekleştirmek üzere Quad yapısındaki biquadratik bloklarda temel altdevreler olarak seçilecektir.

OTA ve Topraklı kapasitelerin, bütün tümleşik devre teknolojilerinde gerçekleştirilebilmeleri kolaydır. Dahası, topraklı kapasite tümleşik devre teknolojisi açısından "doğal" bir elemandır ve MOS tranzistörün giriş kapasitesi ile gerçekleştirilebilir. Topraklı kapasitenin "floating" kapasiteye tercih edilmesinin nedeni, paralel parazitik kapasitif etkileri yokedebilmesindendir. Ayrıca topraklı kapasite, tümleşik devrelerdeki en ucuz lineer kapasitedir [13].

Aşağıda literatürde yer alan bazı OTA-C biquad devreleri verilmektedir.

1. Biquad

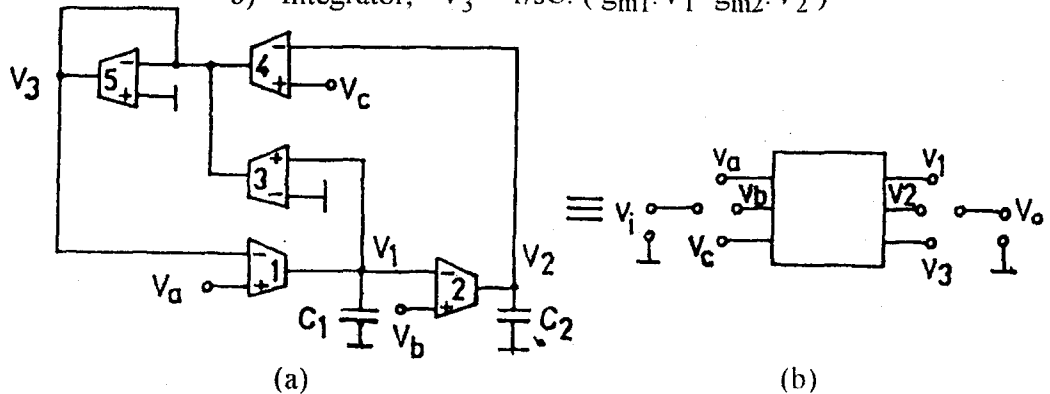
Şekil 4.5 ' te görülen OTA - Topraklı Kapasite devresi, literatürde ikili - integratör çevrimi (two - integratör - loop), [25], diye bilinen az duyarlıklı temel yapıya dayanarak tasarlanmış genel bir biquaddır [1]. Şekil 4.4. ' te ise biquad tasarımında kullanılan temel OTA yapıları verilmektedir.



Şekil 4.4. Temel OTA - C yapıları

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

b) İntegratör, $V_3 = 1/sC. (g_{m1}.V_1 - g_{m2}.V_2)$



Şekil 4.5. Genel OTA-C biquad

V_a , V_b , V_c giriş olarak düşünüldüğünde devrenin çıkış gerilimleri,

$$V_1 = \frac{sC_2g_{m1}(g_{m5}V_a - g_{m4}V_c) + g_{m1}g_{m2}g_{m4}V_b}{D(s)} \quad (4.10.a)$$

$$V_2 = \frac{(sC_1g_{m2}g_{m5} - g_{m1}g_{m2}g_{m3})V_b + g_{m1}g_{m2}(g_{m4}V_c - g_{m5}V_a)}{D(s)} \quad (4.10.b)$$

$$V_3 = \frac{s^2C_1C_2g_{m4}V_c + s(C_2g_{m1}g_{m3}V_a - C_1g_{m2}g_{m4}V_b) + g_{m1}g_{m2}g_{m4}V_a}{D(s)} \quad (4.10.c)$$

olarak bulunur. Burada,

$$D(s) = C_1C_2g_{m5}(s^2 + s \cdot \frac{1}{C_1} \cdot \frac{g_{m1}g_{m3}}{g_{m5}} + \frac{g_{m1}g_{m2}g_{m4}}{C_1C_2g_{m5}}) \quad (4.10.d)$$

dir [1]. Tablo 4.1 'de çeşitli koşullar için elde edilen filtre tipleri verilmiştir.

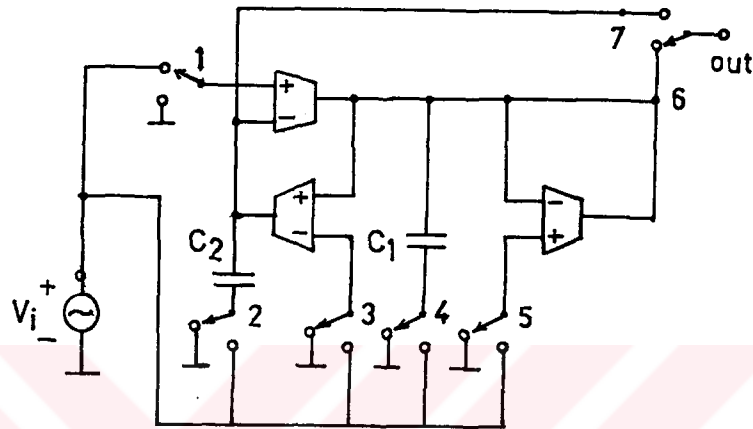
Tablo 4.1. Şekil 4.5 'teki biquada ilişkin değişik koşullar ve filtre tipleri

Filtre Tipi	Pay Polinomu	Giriş Uçları	Çıkış Ucu	Koşul
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	$-sC_2g_{m1}g_{m4}$	$V_c = V_i$	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	$C_2g_{m1}g_{m3} = C_1g_{m2}g_{m4}$

Bu yapıyla, Tablo 4.1. 'deki koşullara göre AG, YG, BG, Notch türünden biquadlar gerçekleştirilebilir.

2. Biquad

Şekil 4.6 ' da görülen OTA-C devresi de ikili - integratör çevrimi yöntemine göre tasarlanmış, az duyarlıklı genel bir biquaddır [19]. Buradaki önemli nokta C_1 ve C_2 kapasitelerinin topraklı olmamasıdır. Daha önce bahsedilen nedenden ötürü bu bir dezavantajdır.



Şekil 4.6. Tablo 4.2.'deki transfer fonksiyonlarını gerçekleyen genel OTA-C biquad devresi

1, 2, 3, 4, 5 düğümlerinin toprağa veya giriş gerilim kaynağına (S_1 , S_2 , S_3 , S_4 , S_5 anahtarları) bağlandığını ve filtre çıkışının 6 veya 7 düğümlerinden alındığını kabul edelim. 1, 2, 3, 4, 5 düğümlerindeki gerilimler [19],

$$V_j = k_j \cdot V_i \quad , \quad j=1,2,\dots,5 \quad (4.11)$$

olarak yazılabilir. Burada V_i giriş gerilimidir, k_j ise 0 veya 1 değerlerini alır [19].

OTA'ların giriş empedanslarını ihmal edersek, 6 ve 7 nolu düğümlerdeki gerilimlerin frekans domenindeki ifadesi [19],

$$\frac{V_6(s)}{V_i(s)} = \frac{N_6(s)}{D(s)}, \quad \frac{V_7(s)}{V_i(s)} = \frac{N_7(s)}{D(s)} \quad (4.12.a)$$

şeklinde ifade edilir. Burada,

$$D(s) = s^2 + s \frac{g_{m3}}{C_1} + \frac{g_{m1}g_{m2}}{C_1C_2} \quad (4.12.b)$$

$$N_6(s) = s^2 k_4 + \frac{s}{C_1} [g_{m1}(k_1 - k_2) + g_{m3}k_5] + \frac{g_{m1}g_{m2}}{C_1C_2} k_3 \quad (4.12.c)$$

$$N_7(s) = s^2 k_2 + \frac{s}{C_1} [g_{m3}k_2 + g_{m2}(k_3 + k_4)] + \frac{g_{m2}}{C_1C_2} [g_{m1}k_1 + g_{m3}(k_3 + k_5)] \quad (4.12.d)$$

$$w_0 = \sqrt{\frac{g_{m1}g_{m2}}{C_1C_2}}, \quad B = \frac{g_{m3}}{C_1} \quad (4.12.e)$$

Burada w_0 kutup genliğini, B ise band genliğini göstermektedir [19].

Tablo 4.2 ' de çeşitli koşullara göre elde edilen biquad tipleri ve bunlara ait pay polinomlarının ifadeleri yer almaktadır [19].

Tablo 4.2. Şekil 4.6 ' daki devreye ilişkin çeşitli koşullara göre elde edilen filtre tipleri

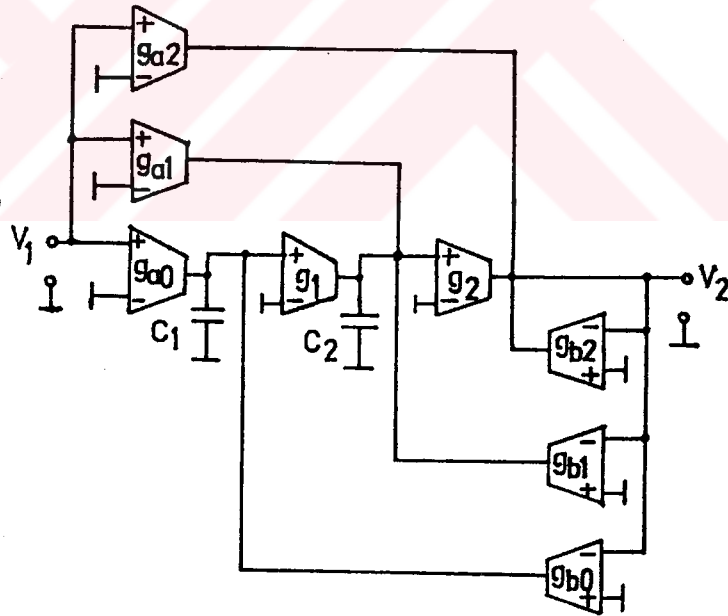
Filtre Tipi	Pay Polinomu	Giriş Düzgümleri, $k=1$	Çıkış Düzgümü	Topraklanan Düzgümler, $k=0$
AG (1.Tip)	$\frac{g_{m1}g_{m2}}{C_1C_2}$	3 veya 1	6 7	1,2,3,4,5 2,3,4,5
AG (2.Tip)	$\frac{g_{m2}g_{m3}}{C_1C_2}$	5	7	1,2,3,4
BG (1.Tip)	$\frac{sg_{m1}}{C_1}$	1 veya 4	6 7	2,3,4,5 1,2,3,5
BG (2.Tip)	$\frac{sg_{m3}}{C_1}$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

3. Biquad

Şekil 4.7 ' de genel amaçlı bir OTA-C biquad devresi, [26], görülmektedir. Devredeki kapasite elemanlarının topraklı olduğu görülmektedir. C_1 , g_1 ve C_2 , g_2 bölümleri sırasıyla birer integratörü temsil etmektedir. g_1 ve g_2 ' nin giriş gerilimleri, C_1 ve C_2 ' den akan OTA akımlarının toplamının integrasyonundan elde edilmektedir. Biquadratik transfer fonksiyonu, doğrudan bu akımların toplanması ve integre edilmesinden bulunmaktadır [26],

$$H(s) = \frac{V_2}{V_1} = \frac{a_2 s^2 + a_1 s + a_0}{b_2 s^2 + b_1 s + b_0} \quad (4.13)$$

Burada, $a_0 = g_{a0}g_1g_2$, $a_1 = g_{a1}g_2C_1$, $a_2 = g_{a2}C_1C_2$, $b_0 = g_{b0}g_1g_2$, $b_1 = g_{b1}g_2C_1$ ve $b_2 = g_{b2}C_1C_2$ dir. a_i ve b_i katsayıları ise, g_{ai} ve g_{bi} transkonduktansları ile ayrı ayrı ayarlanabilir [26]. Transkonduktanslar değiştirilerek, filtrenin frekans cevabı lineer bir şekilde ayarlanabilir. Tablo 4.3 ' te çeşitli koşullar için elde edilen filtre tipleri verilmektedir.



Şekil 4.7. Genel amaçlı bir OTA-C biquad devresi

Tablo 4.3. Şekil 4.7 ' 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)	$a_2 s^2 - a_1 s + a_0$	$a_0 = b_0$, $a_1 = -b_1$, $a_2 = b_2$

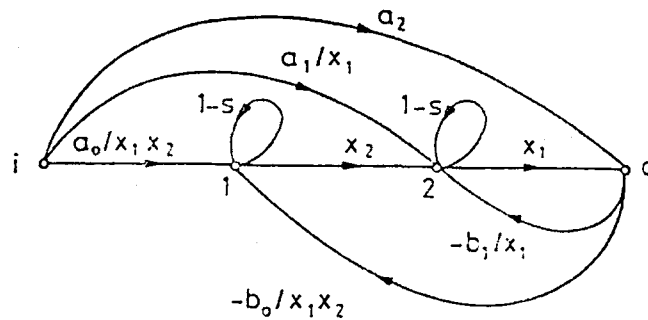
4. Biquad

Burada, Kaynak [8]' te yer alan, minimum OTA elemanı gerektiren bir OTA-C biquad tasarımı ele alınacaktır. Adı geçen kaynakta verilen prosedür ile biquadratik filtrelerin minimum OTA ve topraklı kapasite kullanarak oluşturulması mümkündür. Bu prosedür, aktif filtre tasarımında çok güçlü bir yöntem olan işaret akış diyagramına dayanmaktadır [8].

Genel biquadratik transfer fonksiyonu,

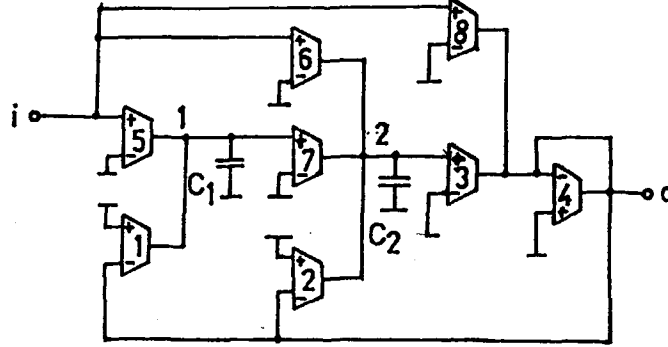
$$G(s) = \frac{V_o}{V_i} = \frac{a_2 s^2 + a_1 s + a_0}{s^2 + b_1 s + b_0} \quad (4.14)$$

şeklinde verilsin. Bu transfer fonksiyonu Şekil 4.8 ' deki işaret akış diyagramıyla gösterilebilir [8].



Şekil 4.8. $G(s)$ ' e karşı düşen işaret akış diyagramı .

Bu diyagrama ait OTA-Topraklı Kapasite devresi Şekil 4.4. ' teki temel OTA-C alt devreleriyle gerçekleştirilebilir. Devre Şekil 4.9. ' da verilmektedir [8].



Şekil 4.9. $G(s)$ ' i gerçekleyen OTA-C devresi

Şekil 4.9 ' daki devreye ilişkin tasarım eşitlikleri,

$$g_{m1} / C_1 = b_0 / x_1 x_2 \quad (4.15.a)$$

$$g_{m2} / C_2 = b_1 / x_1 \quad (4.15.b)$$

$$g_{m3} / g_{m4} = x_1 \quad (4.15.c)$$

$$g_{m5} / C_1 = a_0 / x_1 x_2 \quad (4.15.d)$$

$$g_{m6} / C_2 = a_1 / x_1 \quad (4.15.e)$$

$$g_{m7} / C_2 = x_2 \quad (4.15.f)$$

$$g_{m8} / g_{m4} = a_2 \quad (4.15.g)$$

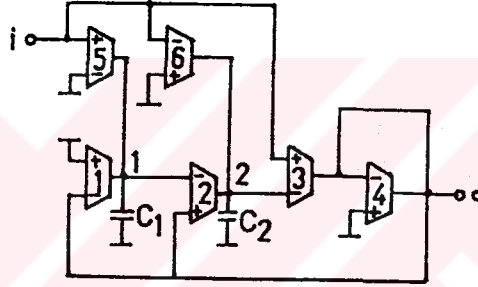
şeklindedir. x_1 ve x_2 bağımsız parametreler olup tasarıma esneklik kazandırır. $x_1 = -a_2$ ve $x_2 = -b_1 / a_2$ seçilirse Şekil 4.8 ' deki bazı dal çiftlerinin transmittansları eşit genlik ve ters işaretle olmaktadır [8]. Yani,

$$t_{o2} = t_{i2} = b_1 / a_2 \quad (4.16.a)$$

$$t_{io} = -t_{2o} = a_2 \quad (4.16.b)$$

Burada $t_{pq}, (p,q)$ dalına ait transmittansı gösterir. Böylece bu çiftlerin herbiri, iki yerine bir OTA ile gerçekleştirilebilir. Bu şekilde, toplam olarak iki OTA daha az kullanılmış olunur [8].

Bu duruma karşı gelen devre ise Şekil 4.10 'da verilmektedir. Burada dikkat edilecek nokta, elde edilen devredeki OTA sayısının minimum olmasıdır. Genel bir biquadratik fonksiyonda 5 farklı katsayı olduğu için, bu keyfi katsayıları gerçekleştirmek üzere en az 5 OTA 'ya gereksinim vardır. 6. OTA ise çıkıştaki toplayıcıyı gerçekleştirmek içindir. Şunu da vurgulamak gerekir ki, bazı katsayıların sıfır olması durumunda kullanılacak OTA sayısı daha da azalacaktır [8].



Şekil 4.10. $G(s)$ 'i gerçekleyen minimum OTA-C biquad devresi

Bu filtre için tasarım eşitlikleri,

$$g_{m1} / C_1 = b_0 / b_1 \quad (4.17.a)$$

$$g_{m2} / C_2 = b_1 / a_2 \quad (4.17.b)$$

$$g_{m3} / g_{m4} = a_2 \quad (4.17.c)$$

$$g_{m5} / C_1 = a_0 / b_1 \quad (4.17.d)$$

$$g_{m6} / C_2 = a_1 / a_2 \quad (4.17.e)$$

olarak verilir. Payda polinomu ise,

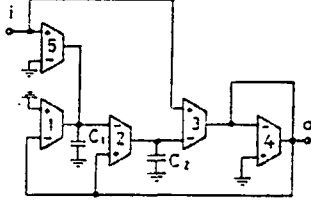
$$D(s) = s^2 + \frac{g_{m2}g_{m3}}{g_{m4}C_2}s + \frac{g_{m1}g_{m2}g_{m3}}{g_{m4}C_1C_2} \quad (4.18)$$

şeklindedir [8].

Tablo 4.4 ' te yukarıda sözü edilen prosedüre göre üretilmiş çeşitli tipten minimum OTA-C filtreleri verilmiştir [8].

Tablo 4.4. Şekil 4.8. ' deki işaret akış grafi kullanılarak oluşturulan OTA-C filtreleri

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

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

4.3.1. İncelenen OTA-C Biquadları Arasındaki Farklar

Her dört biquadın avantajlı ve dezavantajlı yönleri şöyle sıralanabilir :

Şekil 4.5 ' teki devre eleman sayısı bakımından az ve devredeki kapasiteler topraklıdır. Ayrıca çeşitli koşullar yerine getirildiğinde her türden filtreyi gerçekleyebilmektedir. Bu olumlu yanlarının yanı sıra, farklı filtre türleri için işaret giriş ve çıkış uçları değişmektedir. Bu nedenle devre ek anahtarlama devrelerine gereksinim duymaktadır. Bu ise tümleşik devrenin alanını artıracak gibi, ek bir maliyet de getirecektir.

Şekil 4.6. ' daki devre de aynı avantaj ve dezavantajlara sahip olmakla birlikte Şekil 4.5. ' teki devreden ayrıldığı nokta kapasitelerin topraklı olmamasıdır. Bu ise devrenin olumsuz bir yönüdür.

Şekil 4.7 ' deki devre her türden filtreyi gerçekleyebilmektedir. Kapasitelerin topraklı olması da bir avantajdır. Öte yandan 1 ve 2 numaralı biquadlardaki gibi anahtarlama sorunları yoktur. Çünkü devre her türden filtre için aynı giriş-çıkış uçlarını kullanmaktadır. Bununla birlikte eleman sayısı her iki devreye nazaran daha fazladır ve bu bir dezavantaj olarak görünmektedir. Böylece devre diğer biquadlara nazaran daha az ekonomiktir. Ayrıca bu devreyi alt devre olarak kullanan yüksek dereceden bir tümleşik devre kırılganlığının alanı da büyük olacaktır.

Tablo 4.4 ' te verilen devrelerin topolojileri ise filtre türüne göre değişmektedir. Bu nedenle uygulamanın türüne göre farklı topolojiye sahip alt devre seçilmelidir. Yani devre genel amaçlı değildir. Bununla birlikte eleman sayısı - özellikle AG ve BG filtreler için - diğer biquadlara göre daha azdır. Anahtarlama sorunu yoktur, kapasiteler de topraklıdır.

Bu tezde Quad tasarımıdaki biquadratik blokların gerçekleştirilmesinde eleman sayısının azlığı nedeniyle Tablo 4.4 ' teki devrelerden yararlanılacaktır.

Bölüm 3 ' te de değinildiği gibi Butterworth ve Chebyshev yöntemleri için tüm biquadratik altdevreler BG türden olmaktadır ve bu devreler Tablo 4.4 ' teki 2.devre ile gerçekleştirilecektir. Eliptik filtrelerde ise tüm biquadratik fonksiyonlar Notch (eğer varsa T_b hariç) türündendir ve biquadratik alt devreler Tablo 4.4 ' deki 4.devre ile gerçekleştirilecektir.

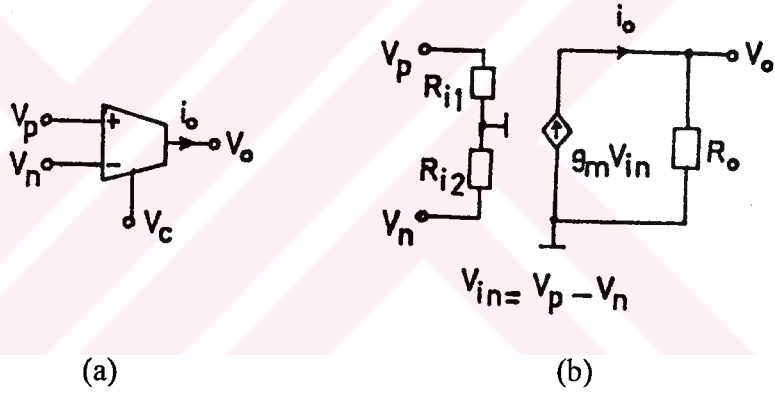
4.4. SİMETRİK CMOS - OTA ' NİN IDEALSİZLİĞİ

Bu alt bölümde ikinci dereceden bazı OTA-C biquadları SPICE programı ile simüle edilmiştir. Simülasyonda Tablo 4.4 ' teki 1b, 2, 3 ve 4 numaralı OTA-C biquadları kullanılmıştır. Aktif eleman olarak ideal OTA ve simetrik CMOS-OTA kullanılmıştır. Her iki simülasyon grubundan elde edilen filtre karakteristikleri birbirleriyle karşılaştırılacaktır. Böylece, simetrik CMOS-OTA ' nın, OTA-C filtre tasarımında güvenle kullanılabilecek bir aktif eleman olup olmadığı konusunda bir karara varılacaktır.

İdeal SPICE simülasyonu için Şekil 4.11 ' deki alt devre kullanılacaktır. Simetrik CMOS-OTA için ise Şekil 4.3 ' teki devre kullanılacaktır. Simetrik CMOS-OTA ' nın SPICE simülasyonunda kullanılacak MOS elemanlarına ait boyutlar Tablo 4.5 ' te verilmiştir.

Tablo 4.5. CMOS -OTA 'daki MOS elemanlarının boyutları

Tranzistor	W (μm)	L (μm)
Q1	100	5
Q2	25	5
Q3	100	5
Q4	25	5
Q5	20	5
Q6	75	5
Q7	20	5
Q8	75	5

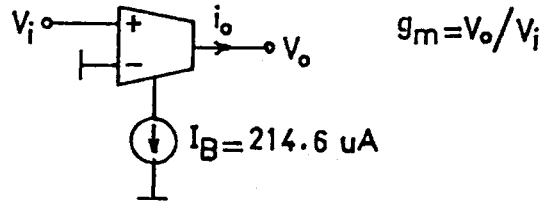


Şekil 4.11. İdeal OTA için SPICE modeli a) İdeal OTA b) SPICE modeli

4.4.1. Simülasyonlar

1. Deney

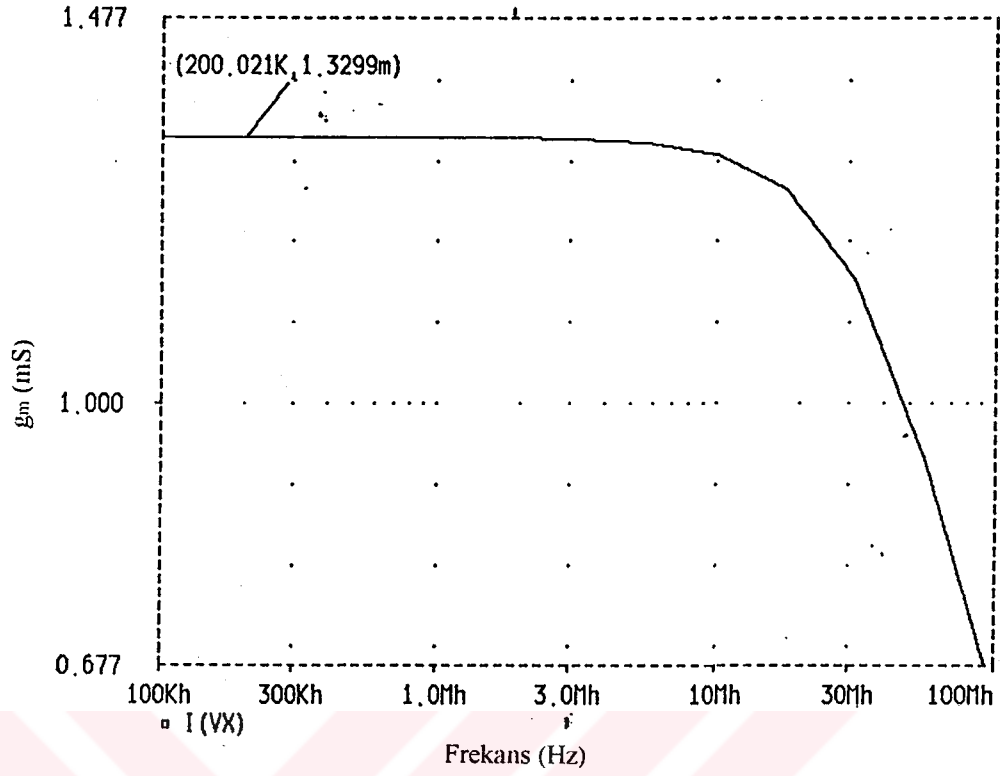
Bu deneyde, belirli bir I_B kontrol akımı için simetrik CMOS OTA 'nın transkonduktans - frekans ($g_m - f$) grafiği çizdirilmektedir. Şekil 4.12.a 'da I_B akımı 214.6 μA değerine ayarlanmış ve devre SPICE programı ile simüle edilmiştir. Elde edilen $g_m - f$ grafiği Şekil 4.12.b' de görülmektedir. Grafikten anlaşılacağı üzere CMOS OTA 'nın g_m değeri yüksek frekanslarda düşmektedir. Bunun anlamı, bu g_m değeri kullanılarak tasarlanan bir filtre, yüksek frekanslarda arzu edildiği gibi çalışmayacaktır.



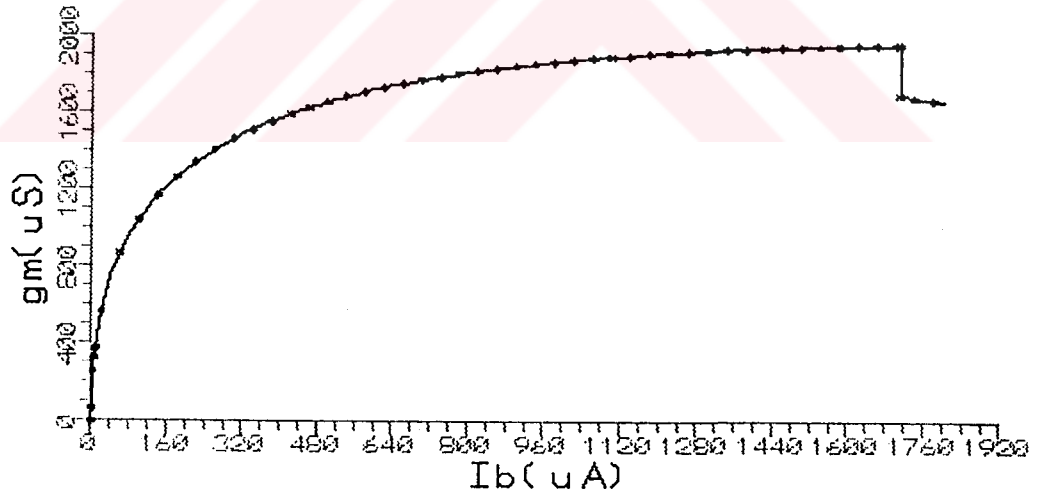
Şekil 4.12.a. 1. Deney düzeneği

Şekil 4.12.c' de ise CMOS - OTA 'ya ait kontrol akımı - iletkenlik, $I_B - g_m$, grafiği çizdirilmiştir. Bu grafik çizdirilirken, I_B 'ye 0.025 uA 'den 1800 uA ' e kadar 20 uA ' lik artımlar verilmiş ve yaklaşık 90 adet simülasyon yapılmıştır. Bu grafik simetrik CMOS OTA 'nın g_m değerini ayarlamak için kullanılabilir.

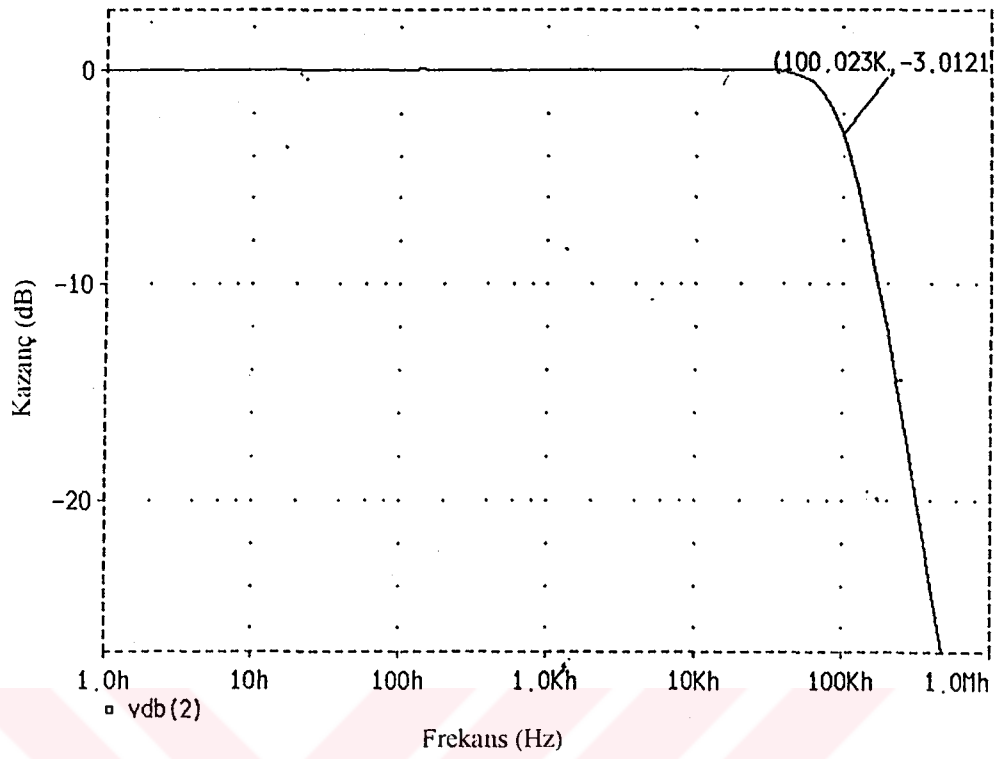
Tablo 4.4 ' teki 1.b devresinin her iki OTA için (ideal ve simetrik CMOS-OTA) frekans cevabı, sırasıyla Şekil 4.12.d ve Şekil 4.12.e ' de verilmiştir. Bu devre için $g_{m1}=g_{m2}=1\text{mS}$ ve $C_1=2.251 \text{ nF}$, $C_2=1.125 \text{ nF}$ olarak seçilmiştir. 1mS 'lik transkonduktansı elde edebilmek için I_B akımı 89 uA ' e ayarlanmıştır.



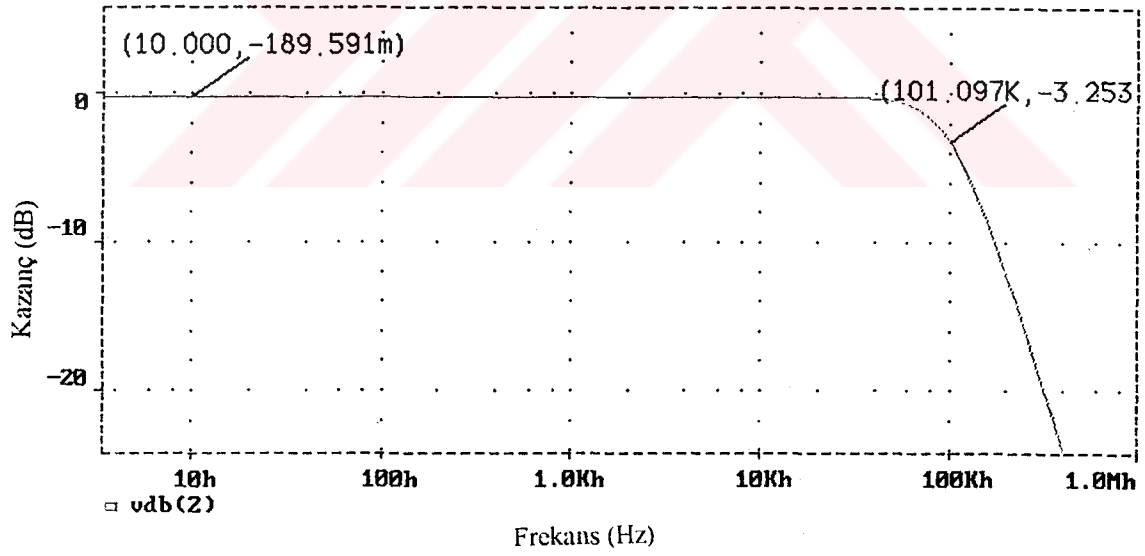
Şekil 4.12.b. Simetrik CMOS-OTA' ya ilişkin iletkenlik-frekans (g_m - f) eğrisi.



Şekil 4.12.c. Simetrik CMOS-OTA' ya ilişkin kontrol akımı-iletkenlik (I_B - g_m) eğrisi.



(d)



(e)

Şekil 4.12. Tablo 4.4' teki 1.b altdevresine ilişkin SPICE simülasyonu

(d) İdeal OTA ile

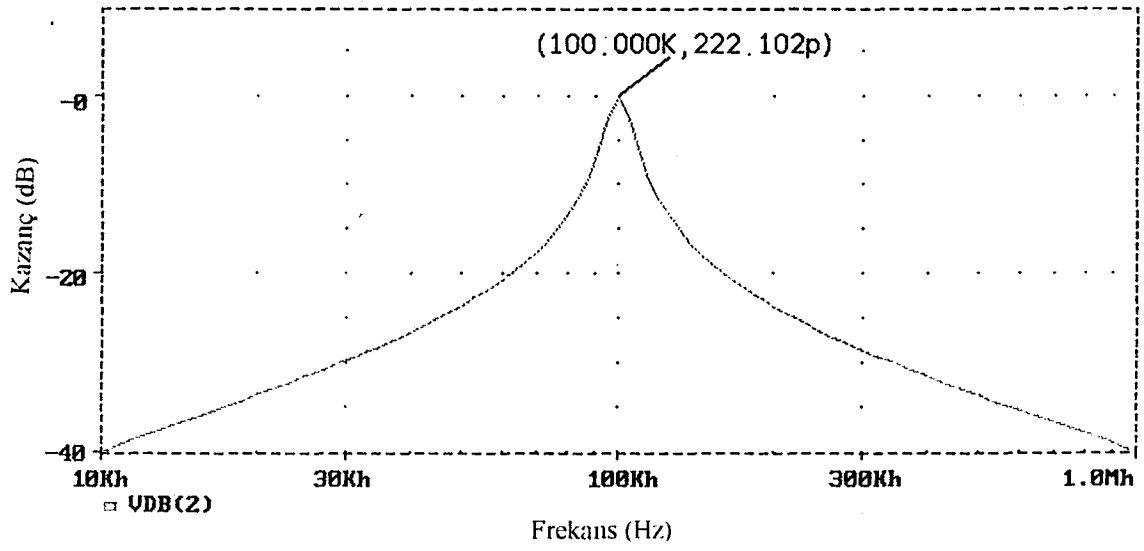
(e) Simetrik CMOS-OTA ile

2. Deney

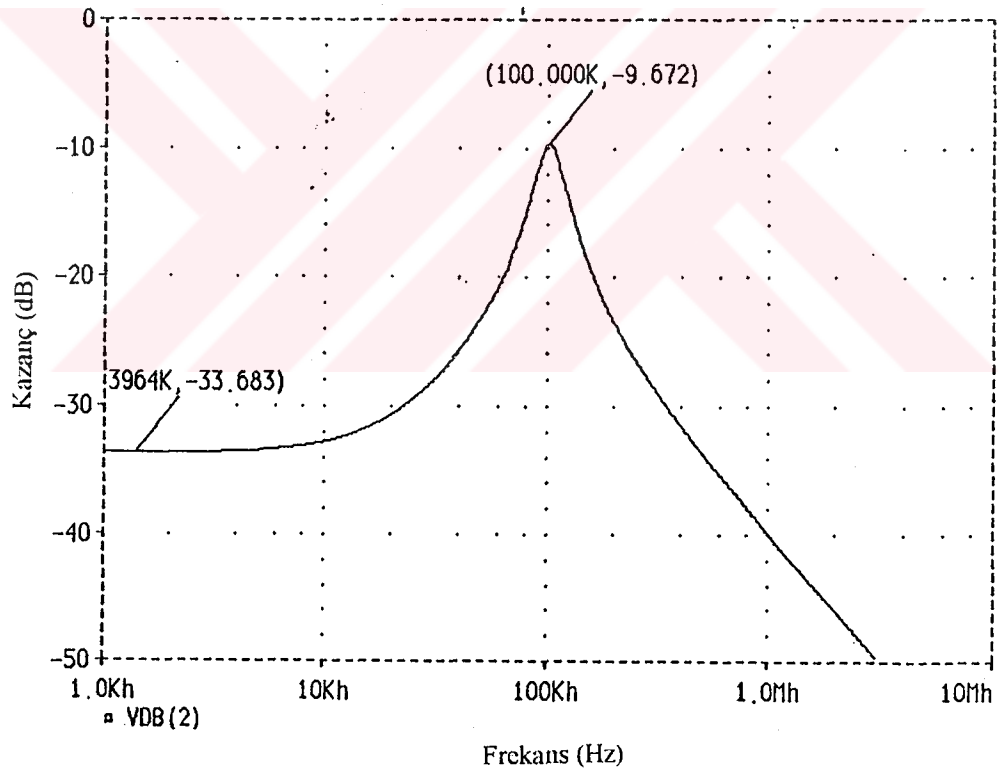
Bu deneyde, Tablo 4.4 'teki 2, 3 ve 4 numaralı OTA-C biquadların, hem ideal hem de simetrik CMOS-OTA için ayrı ayrı simülasyonu yapılmıştır. Arzu edilen filtre özellikleri, filtreye karşılık düşen eleman değerleri ve filtre karakteristikleri Tablo 4.6 'da verilmiştir.

Tablo 4.6. Tablo 4.4 'teki 2,3,4 numaralı OTA-C biquadların simülasyonu için gerekli koşullar ve sonuçlar

Biquad	İstenen Filtre Özellikleri f_0 (KHz)	Hesaplanan Eleman Değerleri	I_B (uA)	Filtre Cevabı	
				İdeal	CMOS-OTA
2	$f_0=100$ $Q=10$	$g_{mi}=1$ mS, $i=1,2,3$ $C_1=159.155$ pF $C_2=15.916$ pF	89	Şekil 4.13.a	Şekil 4.13.b
3	$f_0=100$ $Q=1/\sqrt{2}$	$g_{mi}=1$ mS $i=1,2,3,4$ $C_1=2.251$ nF $C_2=1.125$ nF	89	Şekil 4.14.a	Şekil 4.14.b
4	$f_0=54.9$ $Q=14.75$	$g_{m1}=243.76$ uS $g_{mi}=1934.8$ uS $i=2,3,4,5$ $C_1=47.82$ pF $C_2=82.53$ nF	2.6565 1400	Şekil 4.15.a	Şekil 4.15.b



(a)

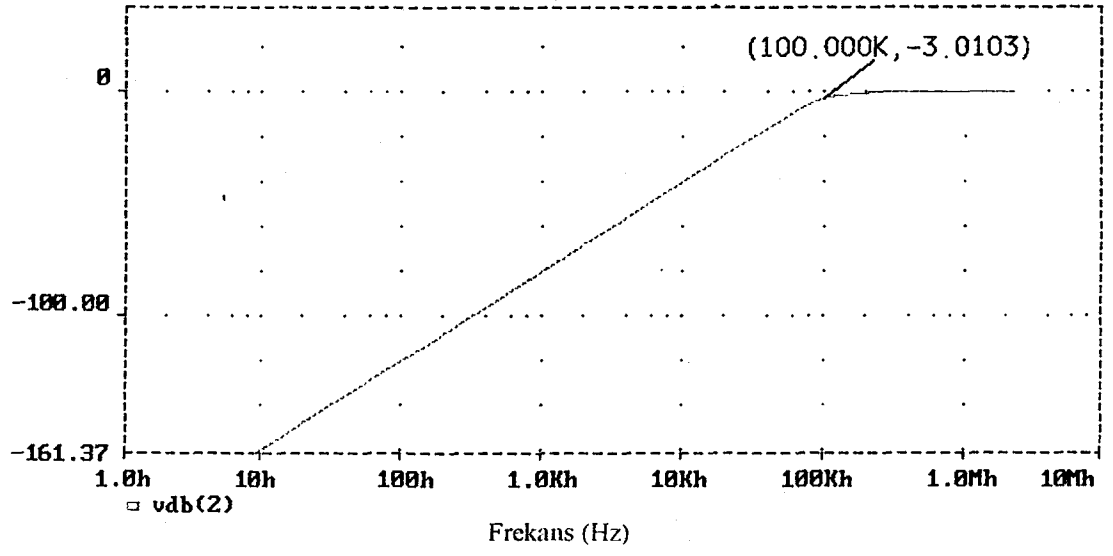


(b)

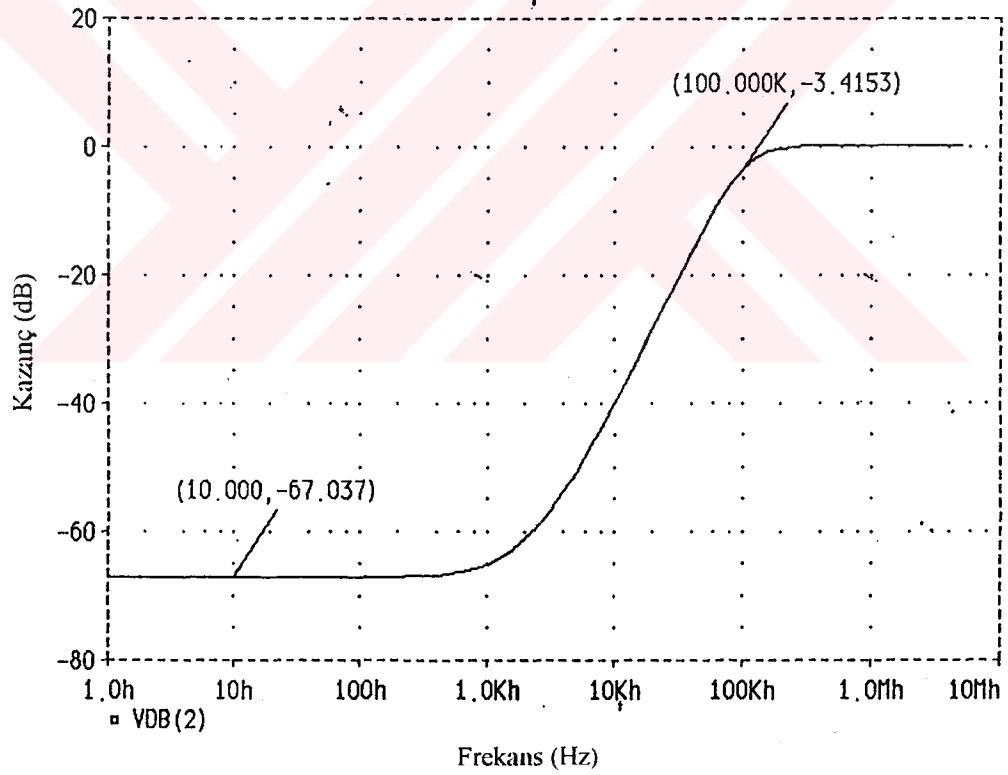
Şekil 4.13. Tablo 4.4' teki 2 numaralı altdevreye ilişkin SPICE simülasyonu

(a) İdeal OTA ile

(b) Simetrik CMOS-OTA ile



(a)

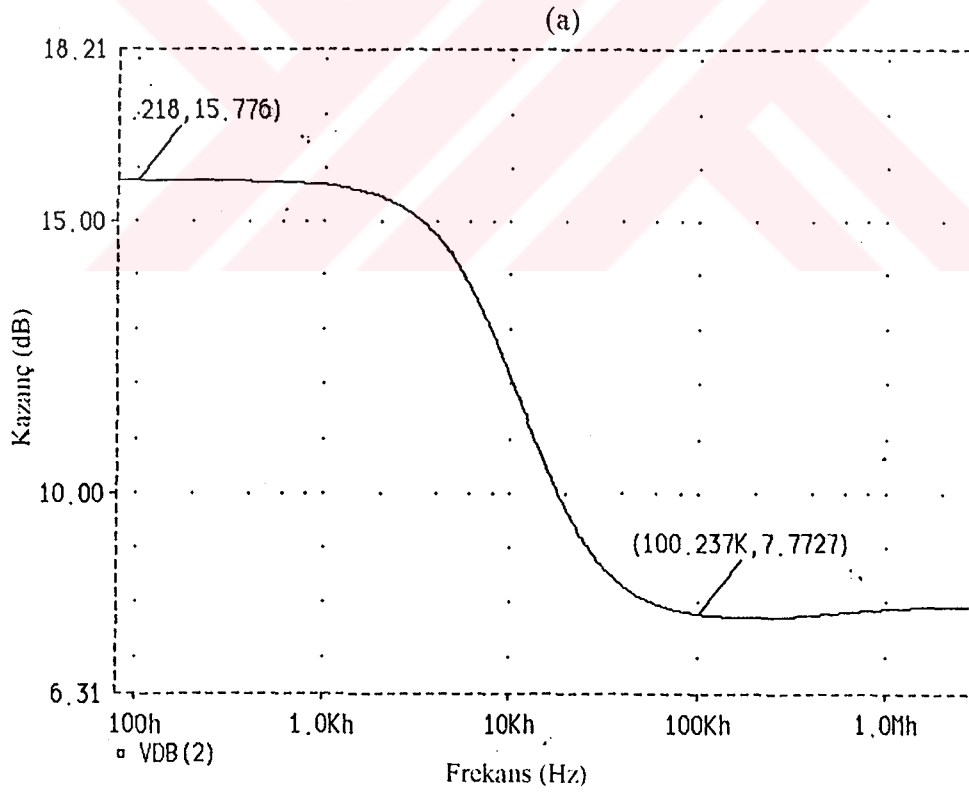
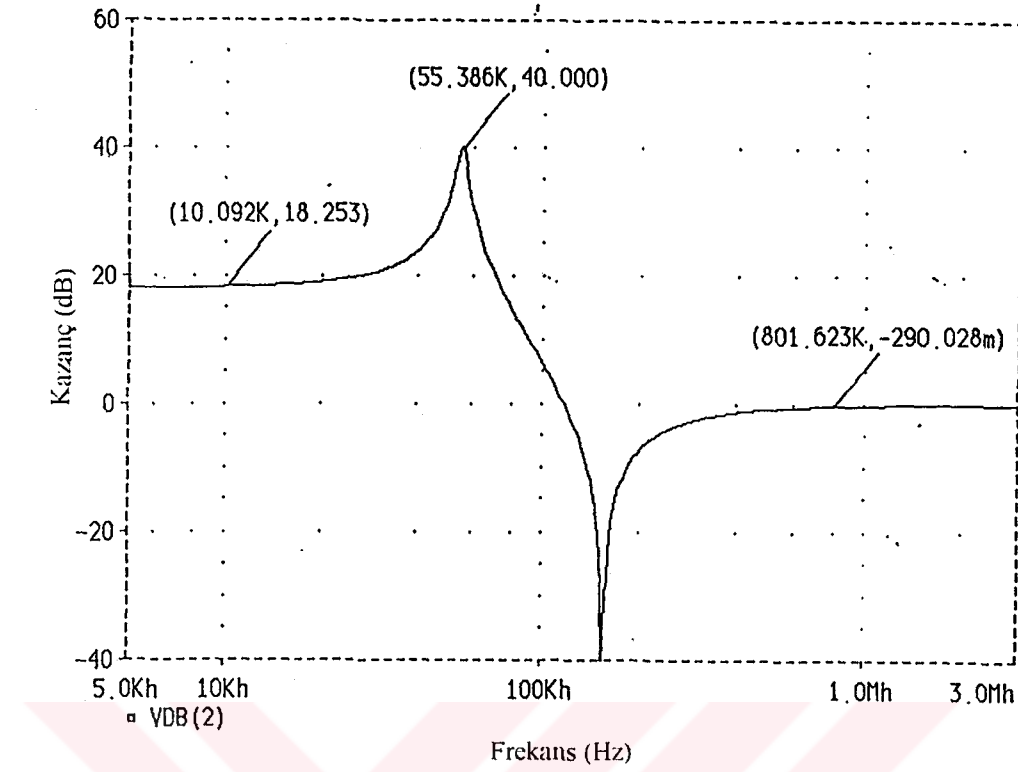


(b)

Şekil 4.14. Tablo 4.4' teki 3 numaralı altdevreye ilişkin SPICE simülasyonu

(a) İdeal OTA ile

(b) Simetrik CMOS-OTA ile



Şekil 4.15. Tablo 4.4' teki 4 numaralı altdevreye ilişkin SPICE simülasyonu

(a) İdeal OTA ile

(b) Simetrik CMOS-OTA ile

Simülasyon Sonuçları

Şekil 4.9.a - Şekil 4.15.b ' den de görüleceği üzere, simetrik CMOS - OTA ile tasarlanan filtrelerin frekans cevabı, ideal OTA ' ya ilişkin frekans cevabından oldukça farklıdır. Tablo 4.4 ' teki 1.b numaralı AG filtre devresi hariç, simetrik CMOS - OTA kullanılarak tasarlanan OTA - C filtreleri, arzu edilen filtre özelliklerini asla karşılayamamaktadır. Bunun temel nedeni 4.2.2 alt bölümünde de sözü edilen OTA idealsizlikleridir ve simetrik CMOS - OTA ' da da bu tür idealsizlikler oldukça etkindir.

Pratikte kullanılan OTA' lar, ideal OTA özelliğinden oldukça uzaktırlar ve çıkış admitansları sıfır değildir. Özellikle yüksek frekanslarda, bu çıkış admitansları, OTA elemanının çıkışında etkili olmakta; OTA elemanının bu nedenle davranışı bozulmaktadır. Bu ise filtrenin davranışının istenenden oldukça farklılaşmasına neden olmaktadır. Bu tür etkilerin, simetrik CMOS OTA' da oldukça baskın olduğu yapılan simülasyonlardan anlaşılmaktadır.

Öte yandan, simetrik CMOS-OTA yapılarını içeren aktif filtre devrelerinin SPICE simülasyonunda kullanılmaya yönelik önerilen bir makromodel , [22], ile sözü edilen simülasyonlar tekrar edilmiş ve benzer sonuçlar bulunmuştur.

Simülasyonlar göstermiştir ki, simetrik CMOS - OTA ' daki bu tür idealsizlikleri önlemeye yönelik tasarım iyileştirmeleri yapılmazsa, OTA - C filtre tasarımında güvenilir bir eleman olarak kullanılamaz.

BÖLÜM 5. YÜKSEK DERECEDEDEN SİMETRİK OTA - C QUAD FİLTRELERİNİN DİNAMİK DEĞİŞİMİ ARTIRACAK ŞEKİLDE GERÇEKLENMESİ VE DUYARLIK

Bölüm 2 ' de , yüksek dereceden aktif filtrelerin geribesleme katsayıları , biquadların kutup ve sıfırları gibi tasarım denklemlerinin Quad yöntemiyle nasıl elde edildiği Kaynak [4] ' den yararlanarak verilmişti. Ayrıca Bölüm 3 ' te de simetrik BG filtreler için durum incelenmiş , Butterworth , Chebyshev ve Eliptik türden filtrelerle ilgili Quad tasarım örnekleri verilmişti. Ancak , hala biquadratık bloklara ilişkin kazanç katsayılarının belirlenmesi tek değildir [4]. Biquadratık blokların kazanç katsayılarının belirlenmesi , aktif filtrelerin dinamik değişimi üzerinde önemli rol oynamaktadır [27].

Bu bölümde , önce Kaynak [4] ' te her türden (simetrik olan ya da olmayan) filtre için verilmiş olan dinamik değişimi artırmaya yönelik prosedür, Bölüm 3 ' te sözü edilen simetrik filtrelere uygulanacaktır. Daha sonra bu kritere göre filtrelerin eleman değerleri hesaplanarak devre tasarlanacaktır.

5.1. QUAD YÖNTEMİNDE DİNAMİK DEĞİŞİMİN ARTIRILMASI

Quad filtrelerin dinamik değişimini artırmaya yönelik prosedür , burada simetrik BG filtrelere uygulanacaktır. Şekil 5.1 ' de Butterworth , Chebyshev ve Eliptik türden simetrik filtrelerin dinamik değişimini inceleyebilmek için bir blok diyagram verilmiş ve devreye ilişkin kritik düğüm gerilimleri de şekilde gösterilmiştir. Devredeki Quad bloğu sayısı k ' dır. $i=1,2,3,\dots,k$ olmak üzere

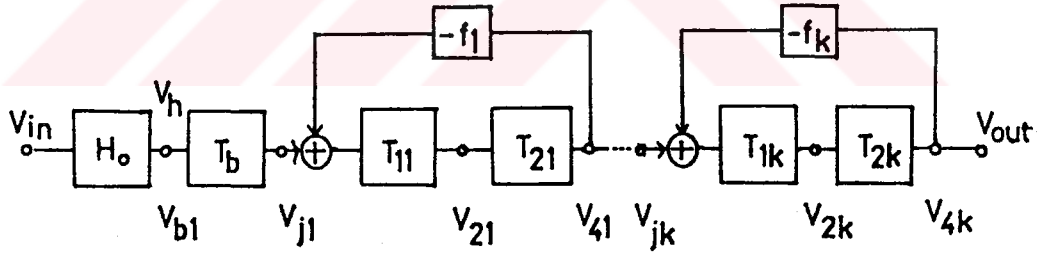
$$C_{ji \min} = \min_{\omega L < \omega < \omega H} |V_{ji}| = \min_{\omega L < \omega < \omega H} \left| \frac{T \cdot V_{in}}{\prod_{i=1}^k T_{Qi}} \right| \quad (5.1.a)$$

$$C_{ji \max} = \max_{0 < w < \infty} |V_{ji}| = \max_{0 < w < \infty} \left| \frac{T \cdot V_{in}}{\prod_{i=1}^k T_{Q_i}} \right| \quad (5.1.b)$$

$$C_{2i \min} = \min_{wL < w < wH} |V_{2i}| = \min_{wL < w < wH} \left| \frac{T \cdot V_{in}}{T_{2i} \prod_{j=i+1}^k T_{Q_j}} \right| \quad (5.11.c)$$

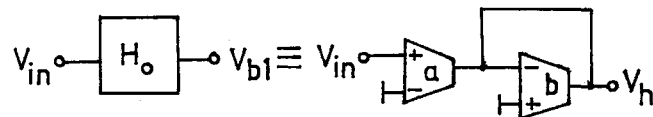
$$C_{2i \max} = \max_{0 < w < \infty} |V_{2i}| = \max_{0 < w < \infty} \left| \frac{T \cdot V_{in}}{T_{2i} \prod_{j=i+1}^k T_{Q_j}} \right| \quad (5.1.d)$$

Burada , i Quad numarasını göstermektedir. C_{ji} ve C_{2i} ise sırasıyla , i numaralı Quad bloğunun j ve 2 ile gösterilen düğümlerindeki gerilimleri ifade etmektedir [4].

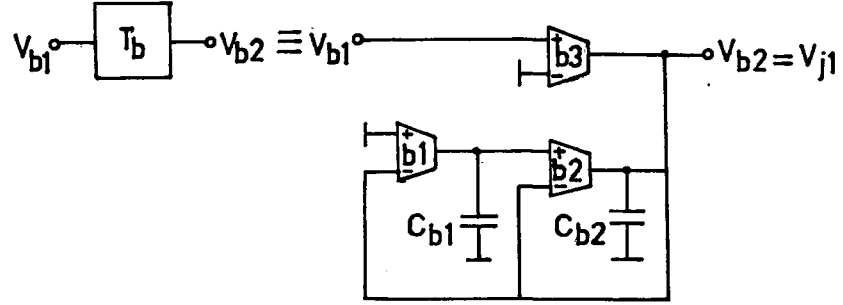


Şekil 5.1. Yüksek dereceden Quad filtresine ilişkin kritik gerilimler

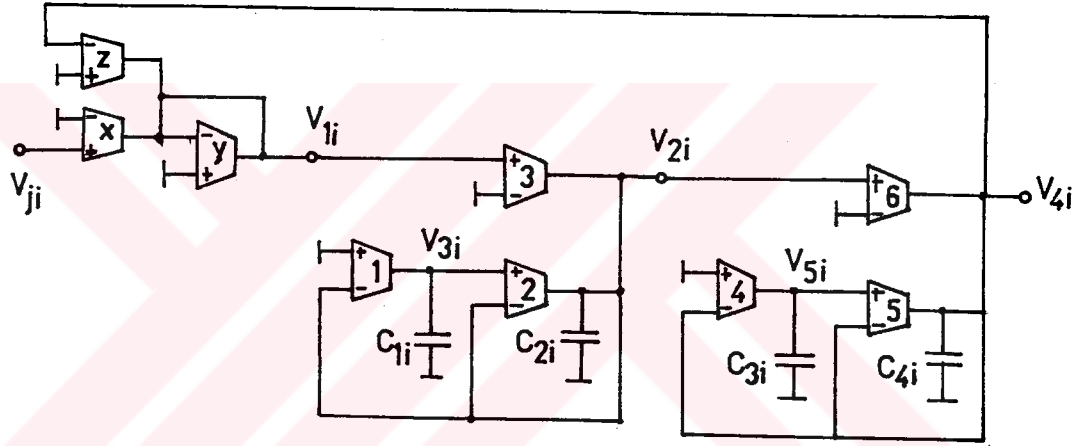
Şekil 5.1' deki Quad bloklarına ilişkin ayrıntılı OTA-C devre diyagramı Şekil 5.2' de verilmektedir.



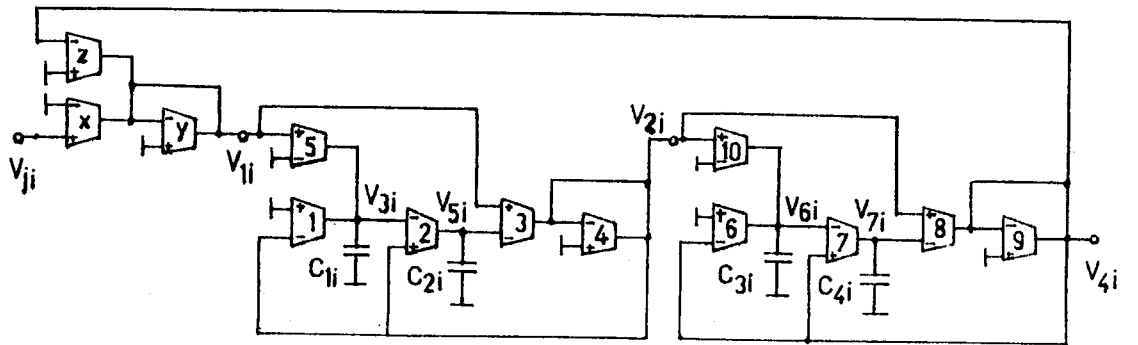
Şekil 5.2.a. OTA-C Quad devresindeki Ho bloğuna ilişkin altdevre



Şekil 5.2.b. OTA-C Quad devresindeki T_b bloğuna ilişkin altdevre



Şekil 5.2.c. Butterworth veya Chebyshev filtre için OTA-C Quad devresine ilişkin altdevre



Şekil 5.2.d. Eliptik filtre için OTA-C Quad filtresine ilişkin altdevre

Şekil 5.1. deki Quad filtresinin ayrıntılı OTA - C devre diyagramı Şekil 5.2 ' de verilmektedir. Şekil 5.2 ' deki Quad devreleri , Tablo 4.4 ' teki altdevrelerin birleştirilmesiyle oluşturulmuştur. H_0 sabit kazanç bloğu ise Şekil 4.4 ' teki devre ile oluşturulmuştur.

(5.1.a) - (5.1.a) denklemlerinde w_L ve w_H , sırasıyla ilgilenilen frekans bandının alt ve üst kenar frekanslarıdır [27].

$$r_{ji} = \frac{C_{ji \max}}{C_{ji \min}} \quad (5.2.a)$$

$$r_{2i} = \frac{C_{2i \max}}{C_{2i \min}} \quad (5.2.b)$$

Sıfır - kutup eşleme ve Quadların sıralaması gözönüne alınarak farklı r_{ji} ve r_{2i} grupları bulunabilir. Böylelikle , iyi bir dinamik değişim için , sıfır kutup eşleme ve sıralama , $\max [r_{ji}]$ ve $\max [r_{2i}]$ minimum olacak şekilde seçilmelidir [27] ,

$$\min (r_{\max}) = \min (\max (r_{ji}) , \max (r_{2i})) \quad (5.3)$$

Bunun anlamı , olası bütün durumlar gözönüne alınarak bulunacak $\min (r_{\max})$ değeri için , sıfır-kutup eşlemesi ve biquadratik blokların sıralaması yapıldığında , her kritik düğüm için eşit dinamik sağlanır [4]. Dikkat edilecek diğer bir husus , sıfır- kutup eşlemenin yüksek dereceli bir filtrenin duyarlılık davranışı üzerindeki önemli etkisidir. Bununla birlikte , Quad bloklarının ve bu bloklara ait biquadratik alt devrelerin sıralanmalarının devrenin duyarlılığına etki etmediğini belirtmek gerekir [4].

Quad blokları ve bu bloklara ilişkin biquadratik blokların dinamik değişimi artıracak şekilde sıralanması aşağıdaki kurala göre yapılabilir [4] :

(i) . Eğer varsa , bir AG (yoksa BG) biquad , filtrenin ilk altdevresi olarak girişe konmalıdır [4]. Butterworth ve Chebyshev türünden filtreler için tüm biquadlar BG türden (T_b dahil) olduğu daha önce söylenmişti. Eliptik türünden filtreler için ise (Eğer varsa T_b hariç) tüm biquadlar Notch türündendir.

(i i) . Filtrenin son blokları - eğer varsa - YG veya BG altdevrelerle sonlandırılmalıdır. Bu sıralama, önceki devrelerden aktarılabacak D.A. değerlerini yok edeceği gibi, her bir biquadratik blokta kullanılan aktif eleman (OTA ya da İ.K.) içinde üretilen alçak frekanslı gürültülerin de ortadan kalkmasını sağlayacaktır [4]. Bu koşul Butterworth ve Chebyshev türden filtreler için kolaylıkla sağlanabilir. Ne var ki Eliptik filtre için bu mümkün olmamaktadır.

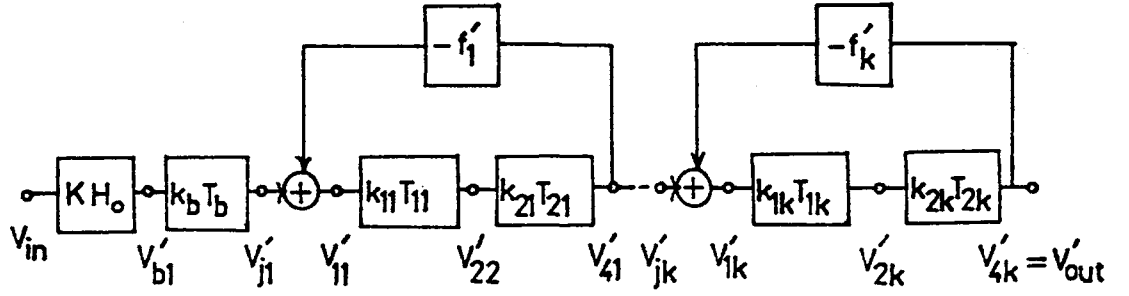
5.2. BİQUADRATİK ALTDEVRELERİN KAZANÇ KATSAYILARININ BULUNMASI VE OTA-C ELEMANLARIYLA GERÇEKLENMESİ

Biquadların kazanç sabitleri doyma , gürültü , veya her iki durum gözönüne alınarak belirlenebilir. Tezde , tasarım sadece doyma durumu gözönüne alınarak yapılacaktır. Bununla birlikte , uygulamanın türüne göre diğer iki duruma göre de kazanç sabitleri bulunabilir. Konu hakkında ayrıntılı bilgi Kaynak [4] ' de bulunabilir.

5.2.1. Doyma Durumuna Göre Kazanç Katsayılarının Belirlenmesi

Filtrenin girişine uygulanan işaretin genliği büyükse, bazı biquadların çıkışlarının doymaya gitmesi filtre cevabını etkileyecektir. O halde , maksimum izin verilen filtre giriş gerilimi , kullanılan OTA ' ların maksimum çıkış işaretiyle sınırlandırılmalıdır. Sonuçta , biquadların kazanç katsayıları , herhangi bir biquad çıkışının diğerlerinden önce doymaya gitmesini önleyecek şekilde seçilmelidir [4].

Şekil 5.1 ' deki yüksek dereceli filtre ve bu filtreye ilişkin " kazanç değerleri atanmış " eşdeğer devre Şekil 5.3 ' te görülmektedir.



Şekil 5.3. Kazanç değerleri atanmış Quad devresi

Şekil 5.1 ve Şekil 5.3 'deki her iki devre için de kritik düğümlere ilişkin gerilim değerleri , filtrenin girişine birim (impuls) işareti uygulanarak hesaplanmıştır. Öte yandan Şekil 5.3.'teki eşdeğer devre için ,

$$\max_{0 < w < \infty} |V_{b1}'| = \max_{0 < w < \infty} |V_{ji}'| = \max_{0 < w < \infty} |V_{2i}'| = \max_{0 < w < \infty} |V_{4k}| = \max_{0 < w < \infty} |V_{out}'| \quad (5.4)$$

ifadesi geçerlidir. Şekil 5.1 , Şekil 5.3 ve (5.4) ifadesinden yararlanarak biquadratik altdevrelerin kazanç katsayıları ,

$$k_{1i} k_{2i} = \frac{\max_{0 < w < \infty} |V_{ji}|}{\max_{0 < w < \infty} |V_{4i}|} \quad (5.5.a)$$

$$k_{2i} = \frac{\max_{0 < w < \infty} |V_{2i}|}{\max_{0 < w < \infty} |V_{4i}|} \quad (5.5.b)$$

$$k_{1i} = \frac{\max_{0 < w < \infty} |V_{ji}|}{\max_{0 < w < \infty} |V_{2i}|} \quad (5.5.c)$$

$$k_b = \frac{\max_{0 < w < \infty} |V_{b1}|}{\max_{0 < w < \infty} |V_{j1}|} \quad (5.5.d)$$

şeklinde bulunur. Frekans cevabının aynı kalmasını sağlamak için girişe konulması gereken " giriş kazanç katsayısı " ,

$$K = \frac{1}{k_b} \cdot \prod_{i=1}^k (k_{1i} \cdot k_{2i})^{-1} = \max_{0 < w < \infty} |V_{out}| \quad (5.5.e)$$

ve yeni geribesleme katsayıları da ,

$$f_i' = \frac{f_i}{k_{1i} \cdot k_{2i}} \quad (5.5.f)$$

olarak bulunur.

Örnek 5.1.

Örnek 3.3 'te verilen Eliptik filtre ele alınsın . Filtreye ait kritik düğüm gerilimleri "QUAD" programıyla hesaplanmış ve Şekil 5.4 'te bu gerilimlerin frekansla değişimi verilmiştir. Aşağıdaki tabloda filtrenin kritik düğümlerine ilişkin tüm banddaki en büyük gerilim değerleri verilmektedir. Her quadın giriş gerilimi, bir önceki quadın çıkış gerilimine eşittir. Bu durum tabloda görülmektedir.

Tablo 5.1. Örnek 3.3 'teki filtrenin kritik düğümlerine ilişkin tüm banddaki en büyük gerilim değerleri.

Gerilim	1. Quad	2. Quad	3. Quad
V_{ii}	0.003094317	0.1777548	0.6117478
V_{2i}	0.077666736	0.7822863	5.89929
V_{4i}	0.1777548	0.6117478	0.9999974

Tablo 5.2 'de ise filtrenin doyma durumuna göre hesaplanmış kazanç ve geribesleme katsayıları verilmiştir.

Tablo 5.2. Örnek 3.3'deki filtreye ilişkin geribesleme ve kazanç katsayıları

katsayı	1. Quad	2. Quad	3. Quad
k_{1i}	0.03984063	0.2272248	0.1036986
k_{2i}	0.4369353	1.278773	5.899305
f_i'	0.5046311	0.9675159	1.02687
K	323.1720095		

Aşağıda örnek filtreye ilişkin kazanç değerleri atanmış Quad tasarım denklemleri verilmektedir. Biquadratik altdevrelere kazanç değerlerinin atanması halinde, filtreye ilişkin kritik düğüm gerilimleri (5.4) eşitliği gereğince birbirine eşit olacaktır. Bu duruma ilişkin kritik düğüm gerilimlerinin frekans cevabı ise Şekil 5.5' te verilmektedir.

$$T_{11}(s) = \frac{0.04019372 s^2 + 4.230186 \times 10^{13}}{s^2 + 2698996 s + 1.136978 \times 10^{14}} \quad (5.6.a)$$

$$T_{21}(s) = \frac{0.4369353 s^2 + 5.36686 \times 10^{12}}{s^2 + 2698996 s + 1.136978 \times 10^{14}} \quad (5.6.b)$$

$$Q_1 = 3.950699$$

$$T_{12}(s) = \frac{0.3160861 s^2 + 1.037355 \times 10^{14}}{s^2 + 1824262 s + 1.136978 \times 10^{14}} \quad (5.6.c)$$

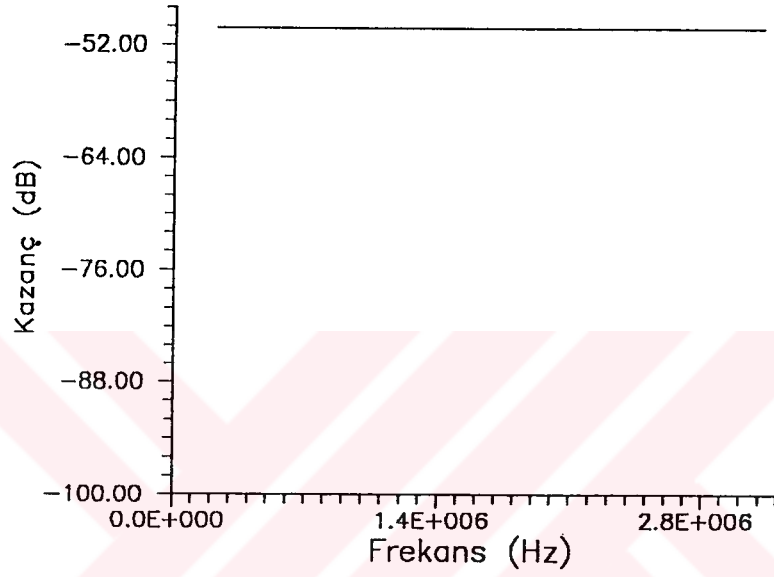
$$T_{22}(s) = \frac{1.278773 s^2 + 5.037044 \times 10^{13}}{s^2 + 1824262 s + 1.136978 \times 10^{14}} \quad (5.6.d)$$

$$Q_2 = 5.845058$$

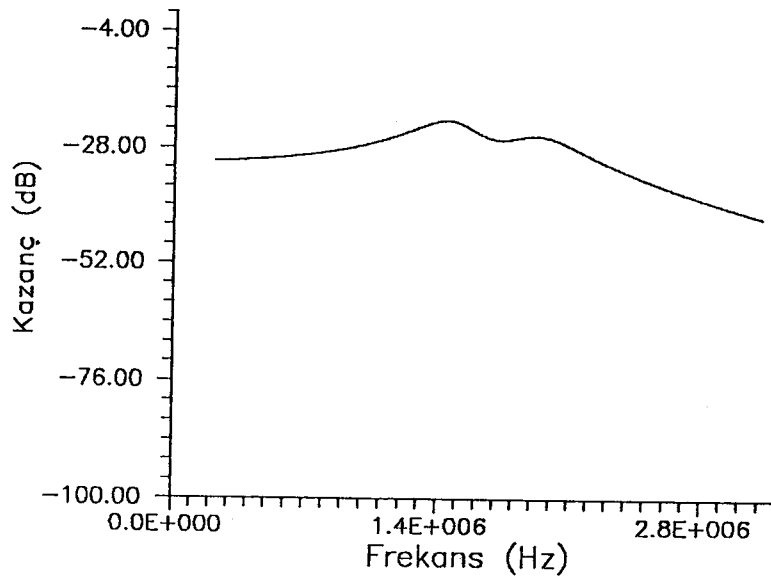
$$T_{13}(s) = \frac{0.2788997 s^2 + 7.566859 \times 10^{13}}{s^2 + 892506.9 s + 1.136978 \times 10^{14}} \quad (5.6.e)$$

$$T_{23}(s) = \frac{5.899305s^2 + 2.810849 \times 10^{14}}{s^2 + 892506.9s + 1.136978 \times 10^{14}} \quad (5.6.f)$$

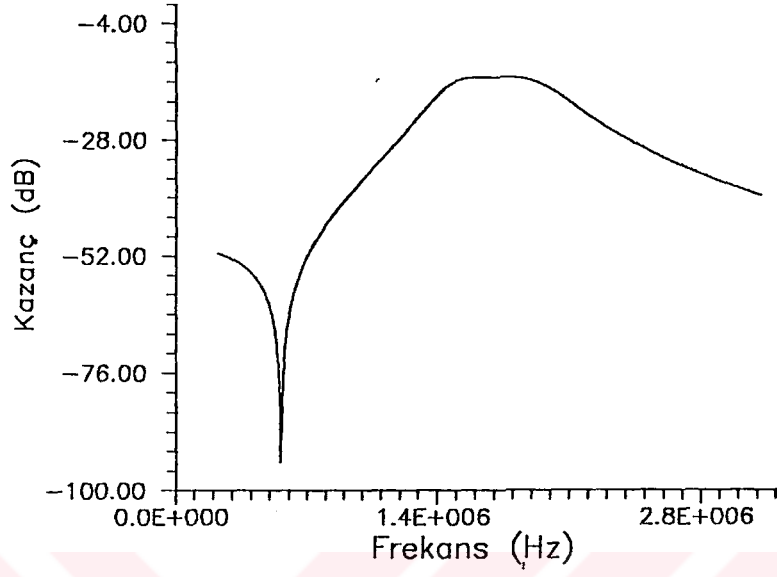
$$Q_3 = 11.94716$$



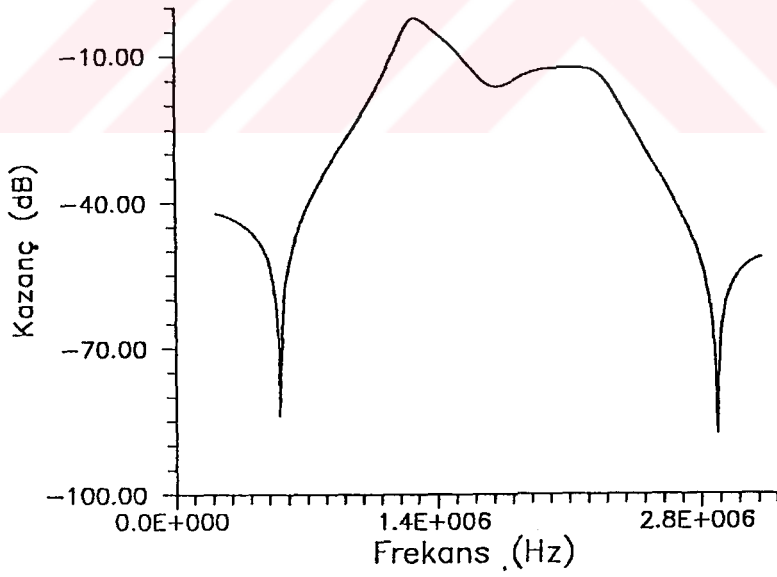
Şekil 5.4.a. Örnek 3.3 'teki filtrenin V_{j1} gerilimi



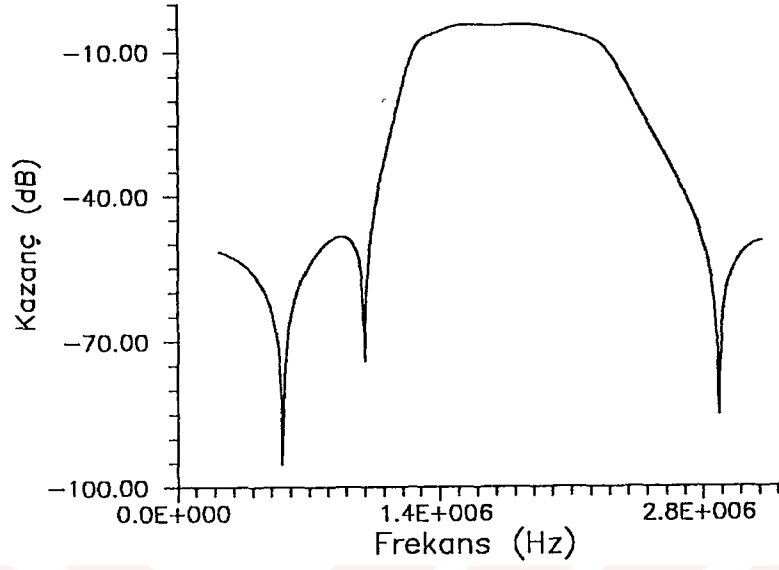
Şekil 5.4.b. Örnek 3.3 'teki filtrenin V_{21} gerilimi



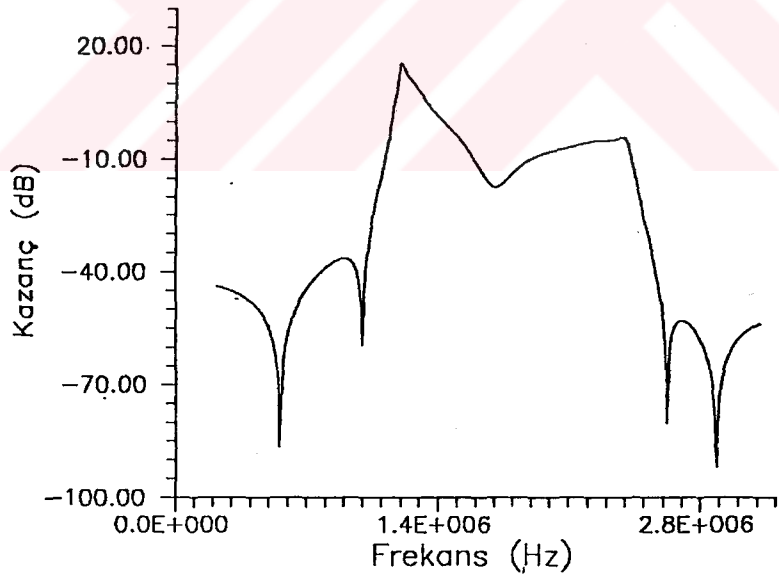
Şekil 5.4.c Örnek 3.3 'teki filtrenin V_{j2} gerilimi



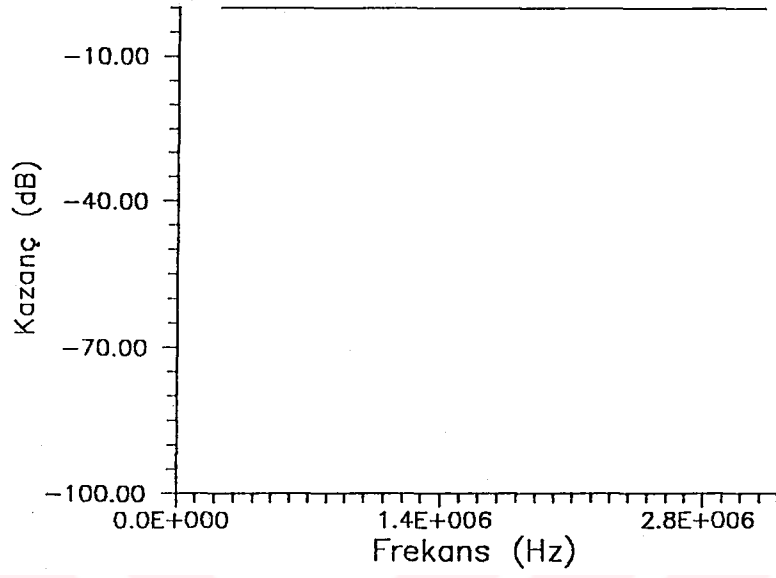
Şekil 5.4.d Örnek 3.3 'teki filtrenin V_{22} gerilimi



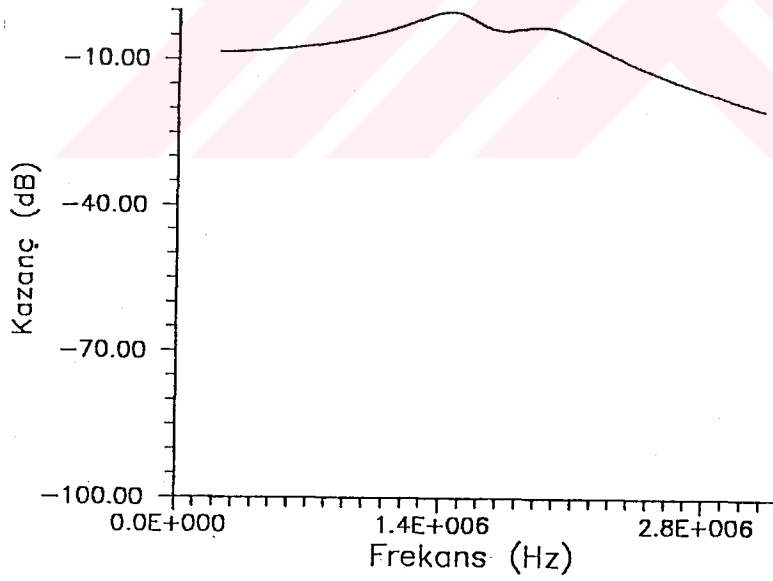
Şekil 5.4.e Örnek 3.3 'teki filtrenin V_{j3} gerilimi



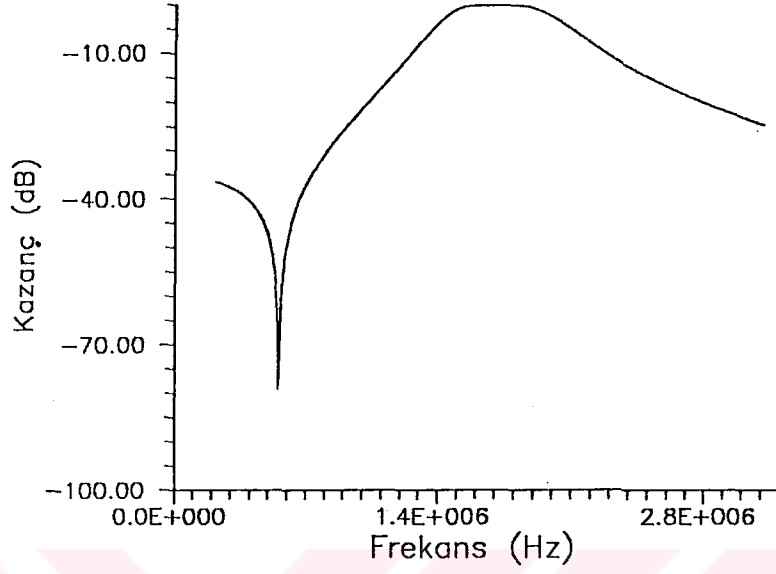
Şekil 5.4.f Örnek 3.3 'teki filtrenin V_{23} gerilimi



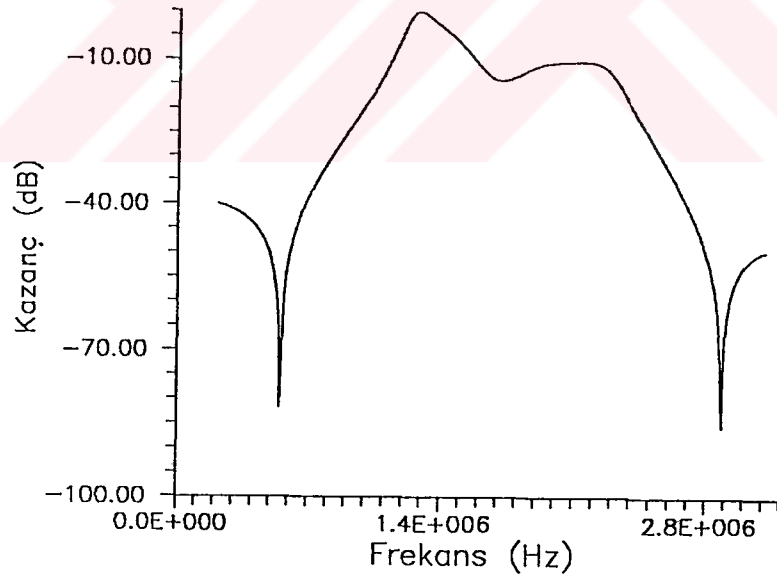
Şekil 5.5.a Kazanç katsayıları atandığında Örnek 3.3'teki filtrenin V_{j1} ' gerilimi



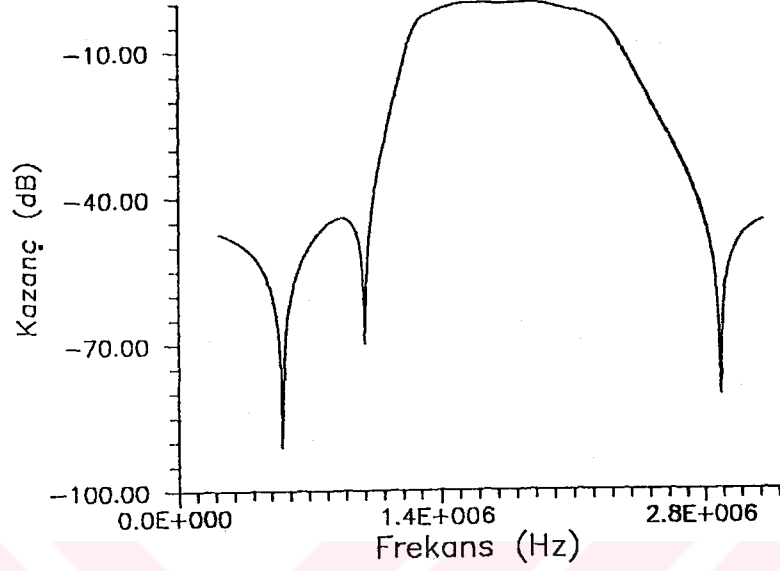
Şekil 5.5.b Kazanç katsayıları atandığında Örnek 3.3'teki filtrenin V_{21} ' gerilimi



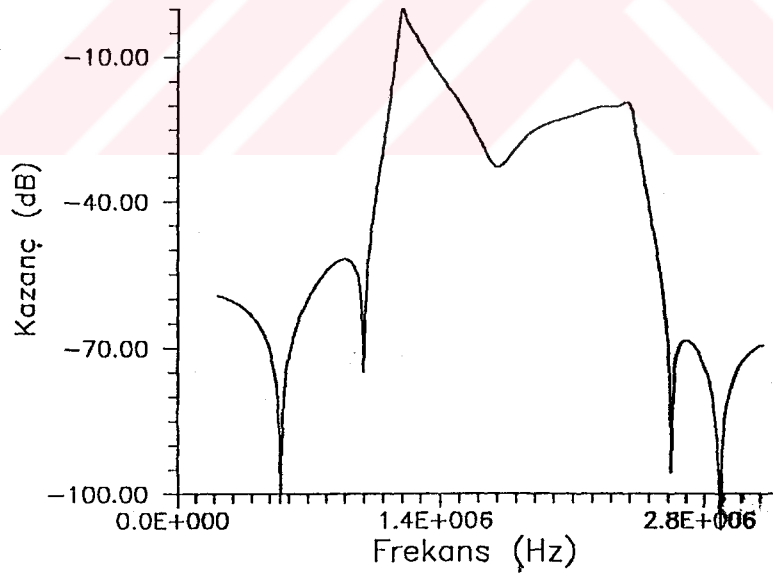
Şekil 5.5.c Kazanç katsayıları atandığında Örnek 3.3'teki filtrenin V_{j2}' gerilimi



Şekil 5.5.d Kazanç katsayıları atandığında Örnek 3.3'teki filtrenin V_{22}' gerilimi



Şekil 5.5.e Kazanç katsayıları atandığında Örnek 3.3'teki filtrenin V_{j3} gerilimi



Şekil 5.5.f Kazanç katsayıları atandığında Örnek 3.3'teki filtrenin V_{23} gerilimi

Filtreye ait Quad blokları, (5.3) ifadesiyle verilen kritere göre sıralandığında dinamik değişim için optimum çözüm elde edilir. Olası bütün durumlar gözönüne alındığında, hesaplama zamanı oldukça artacaktır. Bu nedenle, quad blokları artan iyilik katsayılarına göre sıralanmış ve optimuma yakın çözüm elde edilmesi amaçlanmıştır [1].

5.2.2. Biquadratik Alt Devrelerin Gerçekleştirilmesi

Bu alt bölümde , dinamik değişim göz önüne alınarak tasarım denklemleri elde edilmiş olan Quad filtrenin eleman değerleri, (OTA iletkenlikleri ve kapasite eleman değerleri) hesaplanacaktır.

Gerçeklemede , şekil 5.2 ' deki OTA-C Quad devreleri kullanılacaktır. Bu devrelerdeki eleman değerleri ve gerçekleştirilecek filtreye ait biquadratik transfer fonksiyonlarının katsayıları karşılaştırılırsa , devrelere ilişkin OTA iletkenlik ve kapasite elemanlarının değerleri kolayca bulunabilir.

Diğer taraftan , tasarımda temel biquadlar olarak kullanılacak olan Tablo 4.4 ' teki 2. ve 4. devrelerde , devredeki parametre sayısının tasarım eşitliklerinden fazla olduğu görülmektedir. Böylece , bu bağımsız değişkenlere uygun değerler verilerek , ya en az duyarlılık ya da iyi dinamik davranışa sahip devreler elde edilebilir. Butterworth ya da Chebyshev ve Eliptik filtreler için tipik biquadratik OTA-C altdevreleri ve bunlara ait eleman değerleri (OTA iletkenlikleri ve kapasite elemanlarının değerleri) Tablo 4.4 ' ten yararlanarak rahatça bulunabilir. Aşağıdaki tabloda Butterworth ya da Chebyshev ve Eliptik türden filtrelere ait biquadratik altdevreler için (dinamik değişimi artıracak yönde) OTA iletkenlikleri ve kapasite elemanlarının değerleri verilmektedir.

Tablo 5.3. Butterworth, Chebyshev ve Eliptik türden OTA-C Quad filtrelerine ilişkin temel biquad devrelerinin eleman değerleri

Filtre Tipi	OTA-C Devresi	Devre Parametreleri (g_m , C)	
		Bağımlı	Bağımsız
Butterworth veya Chebyshev	Tablo 4.4 ' te 2.Devre	$C_2 = g_{m3} / a_1$	$g_{m3} (100 \text{ uS})$
		$g_{m2} = C_2 b_1$	
		$C_1 = g_{m1} \cdot b_1 / b_0$	$g_{m1} (100 \text{ uS})$
Eliptik	Tablo 4.4 ' te 4.Devre	$g_{m3} = a_2 g_{m4}$	
		$C_2 = g_{m2} \cdot a_2 / b_1$	$g_{m4} (100 \text{ uS})$
		$C_1 = g_{m5} \cdot b_1 / a_0$	$g_{m2} (100 \text{ uS})$
		$g_{m1} = C_1 \cdot b_0 / b_1$	$g_{m5} (100 \text{ uS})$

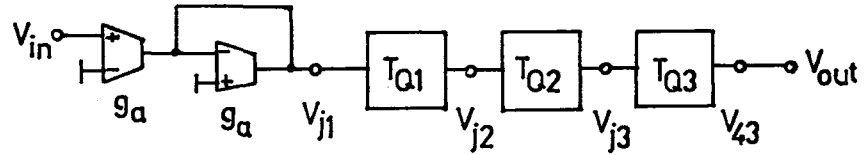
Tablo 5.3 ' teki bağımsız devre parametreleri (OTA iletkenlikleri) birbirine eşit ve 100 uS seçilmiştir. Bu değer çoğu örnek için uygun netice vermiştir. (Bu değer 1 mS olması durumunda , denenen bazı örneklerde transkonduktans değerlerinin pratik açıdan kabul edilemez değerlere ulaştığı gözlenmiştir.)

Örnek 5.2.

Örnek 3.3 ' te verilen Eliptik filtre için OTA transkonduktansları ve kapasite elemanlarının değerleri Şekil 5.2 ve Tablo 5.3 ' ten yararlanarak hesaplanmış ve Tablo 5.4 ' te verilmiştir. Bütün transkonduktanslar uS , kapasite elemanları ise pF cinsinden verilmiştir. Ayrıca filtreye ait tam devre diyagramı Şekil 5.6 ' da görülmektedir. Diyagramda görülen Quad blokları Şekil 5.2. ' deki OTA-C alt devreleri ile gerçekleştirilmiştir.

Tablo 5.4. Örnek 3.3 ' teki filtre için eleman değerleri

Eleman Değerleri g_i (μS) C_i (pF)	1. Quad	2. Quad	3. Quad
g_1	268.7774	109.6036	150.2576
g_2	100	100	100
g_3	4.019372	31.6086	27.88997
g_4	100	100	100
g_5	100	100	100
g_6	2118.517	225.7233	40.44963
g_7	100	100	100
g_8	43.69353	127.8773	589.9305
g_9	100	100	100
g_{10}	100	100	100
g_x	100	100	100
g_v	100	100	100
g_z	50.46311	96.75159	102.687
C_1	6.380	1.759	1.179
C_2	1.489	17.327	31.249
C_3	50.290	3.622	0.318
C_4	16.189	70.098	660.982
g_a	99.99976		
g_b	100		



Şekil 5.6. Örnek 3.3'de verilen filtrenin Quad yöntemi kullanılarak OTA-C elemanlarıyla gerçekleştirilmesi

Örnek 5.3.

Örnek 3.1 ' de verilen BG Butterworth filtreye ilişkin dinamik değişim gözönüne alınarak, Quad tasarım denklemleri elde edilmek istensin. Filtrenin kritik düğümlerine ilişkin tüm banddaki en büyük gerilim değerleri EK ' teki bilgisayar programı ile hesaplanmış ve sonuçlar Tablo 5.5 ' te verilmiştir. Bu değerlerden ve (5.5) ifadelerinden hareket ederek filtredeki biquadratik altdevrelere atanacak kazanç katsayıları yine program yardımıyla hesaplanmış ve sonuçlar Tablo 5.6 ' da verilmiştir. Filtrenin eleman değerleri ise Tablo 5.7 ' de verilmiştir.

Tablo 5.5. Örnek 3.1'deki filtreye ilişkin tüm banddaki en büyük gerilim değerleri

	1. Quad	2. Quad	3. Quad
V_{ii}	2.862776	2.016159	1.419915
V_{2i}	2.320597	1.200969	0.9879387
V_{4i}	2.016159	1.419915	1

Tablo 5.6. Örnek 3.2'deki filtreye ilişkin biquadlara atanacak kazanç katsayıları

	1. Quad	2. Quad	3. Quad
k_{1i}	1.233637	1.678777	1.43725
k_{2i}	1.150999	0.8458037	0.9879383
f_i'	0.06698719	0.4999999	0.9330135
K	0.3493115		

Filtreye ait kazanç değerleri atanmış Quad tasarım denklemleri ,

$$T_{11}(s) = \frac{31004695s}{s^2 + 2892777s + 7.579856 \times 10^{13}} \quad (5.7.a)$$

$$T_{21}(s) = \frac{2892778s}{s^2 + 2892777s + 7.579856 \times 10^{13}} \quad (5.7.b)$$

$$Q_1 = 3.009647$$

$$f_1' = 0.06698719$$

$$T_{12}(s) = \frac{4219227s}{s^2 + 2117660s + 7.579856 \times 10^{13}} \quad (5.7.c)$$

$$T_{22}(s) = \frac{2125737s}{s^2 + 2117660s + 7.579856 \times 10^{13}} \quad (5.7.d)$$

$$Q_2 = 4.111254$$

$$f_2' = 0.49999999$$

$$T_{13}(s) = \frac{3612203s}{s^2 + 775117.4s + 7.579856 \times 10^{13}} \quad (5.7.e)$$

$$T_{23}(s) = \frac{2482960s}{s^2 + 775117.4s + 7.579856 \times 10^{13}} \quad (5.7.f)$$

$$Q_3 = 11.23215$$

$$f_3' = 0.9879383$$

Tablo 5.7. Örnek 3.2. ' deki filtreye ilişkin eleman değerleri

Eleman Değerleri g_i (uS) , C_i (pF)	1. Quad	2. Quad	3. Quad
g_1	100	100	100
g_2	93.30126	50.19071	21.4583
g_3	100	100	100
g_4	100	100	100
g_5	99.99998	99.62003	31.21747
g_6	100	100	100
g_v	100	100	100
g_v	100	100	100
g_z	6.698719	49.99999	93.30135
C_1	3.816401	2.793799	1.022602
C_2	32.25318	23.70103	27.68394
C_3	3.816401	2.793799	1.022602
C_4	34.56885	47.04252	40.27451
g_a	100		
g_b	100		

5.3. OTA - C QUAD FİLTRELERİN DUYARLIĞI

Mühendislikte duyarlık oldukça önemli bir kavramdır ve devre parametrelerinde meydana gelen küçük değişikliklerden, devreyi belirleyen büyüklüğün ne şekilde etkileneceği incelemeye değer bir noktadır. Parametrelerinin nominal değerlerinden çeşitli nedenlerle sapma göstermesi sonucu devrenin davranışı bozulabilir . Sıcaklık , nem, yaşlanma, üretim hataları vb. nedenlerden devre parametrelerinde ortaya çıkan değişimler, devre fonksiyonlarının genliğinde, fazında, sıfır ve kutuplarında, pay ve payda polinomlarının katsayılarında değişimlere yol açarlar.

Duyarlık konusunda yapılan çalışmalar arasında, çeşitli yöntemler mevcuttur. Bunlar arasında en kullanışlı olanı " İstatistiksel Çok Parametrelili Duyarlık Ölçüsü " , [28] , olarak bilinen yöntemdir. Bu yöntem, " Monte Carlo Yöntemi " , oranla çok daha az bilgisayar hesaplama zamanı gerektirmektedir. Hala süregelen duyarlık

çalışmaları ; devreleri duyarlık açısından karşılaştırabilmek ve düşük duyarlıklı devreleri bulmaya yöneliktir.

Quad tasarımı , ilgilenilen transfer fonksiyonunun $j\omega$ ekseninde sıfırlarının olup olmaması durumuna göre, duyarlılık analizi ayrı ayrı ele alınmaktadır. Öte yandan daha az bilgi işlem zamanı gerektirdiği için duyarlık analizinde İstatistiksel Çok Parametrelili Duyarlık Ölçüsü kullanılmaktadır. Bunun yanısıra gösterilmiştir ki , Quad yöntemi FLF, F, MF gibi yöntemlerle aynı seviyede duyarlılığa sahiptir. Ayrıca simetrik Butterworth ve Chebyshev BG filtreleri için filtre derecesi yükseldikçe Quad tasarım yöntemi , FLF tasarım yöntemine oranla daha az duyarlı filtre devreleri vermektedir [4].

Bu tezde, OTA-C Quad filtrelerinin tasarımı, dinamik değişimi artıracak şekilde gerçekleştirilmeye çalışılmıştır. Bunun yanısıra, iyi dinamik değişime sahip bir devre, aynı zamanda duyarlık yönünden de iyi davranış gösterebilir. Öte yandan, tasarlanan filtrenin , ayrıntılı duyarlık hesaplamalarına girilmemiştir. Bununla birlikte, Kaynak [4]' te İ.K., direnç,kapasite (Active-RC) elemanlarından oluşan biquadlar için Quad filtrelerine ilişkin yapılan duyarlık incelemelerinde büyük değişimler olmayacaktır.

Kaynak [4] ' de ; Quad filtrelerinden biquadratik altdevreler, İ.K. , direnç ve kapasite elemanlarından oluşmakta ve transfer, genlik ya da faz fonksiyonlarının duyarlığı bu parametrelere göre hesaplanıyordu. OTA- Topraklı Kapasite filtre devrelerinde, devre parametreleri OTA iletkenliği (transkonduktans) ve kapasite değerlerinden oluşmaktadır. Bu parametrelerde (g_m ve C) meydana gelen küçük değişimlerin filtre transfer fonksiyonu, ω_0 ve Q üzerindeki etkisinin belirlenebilmesi, duyarlık analizi ile mümkündür.

OTA-C elemanlarıyla gerçekleştirilen tümleşik devre içindeki, OTA elemanlarının iletkenliklerini belirleyen kontrol akımlarının kesin ve kararlı olması gerekir. Aksi halde OTA iletkenlik değerleri değişecek ve filtre istenen davranıştan uzaklaşacaktır. Bununla birlikte bu sakınca, OTA iletkenliğinin kontrol akımıyla ayarlanabilme özelliği sayesinde önemli ölçüde ortadan kaldırılır. OTA elemanının bu ayarlanabilme özelliği kullanılarak, devre fonksiyonundaki değişimleri en aza indirgeyerek ayarlama teknikleri geliştirilmiştir. Bunlar arasında en önemli olanı, "Otomatik Ayarlama (Automatic Tuning) " , [1] , olarak bilinen yöntemdir.

Tezde, simetrik filtrelerin tasarımı için biquadratik OTA-C altdevreleri olarak önerilen devrelerin (Tablo 4.4) duyarlıkları düşük devreler oldukları gösterilmiştir [8]. Böylece minimum duyarlıklı bu altdevrelerle oluşturulacak yüksek dereceden filtrenin (Quad veya Kaskad) duyarlılığı da düşük olacaktır. Üstelik OTA-C biquadlardaki eleman sayısının azlığı nedeniyle; tasarlanan filtre, İ.K.-RC elemanlarıyla gerçekleştirilen eşdeğerinden daha az duyarlık verebilir.

Tablo 5.8 ' de , tezde kullanılan Tablo 4.4. ' deki temel OTA-C altdevrelerinin w_0 ve Q duyarlıkları verilmektedir. Tablodan, da anlaşılacağı üzere sözü edilen biquadların duyarlılığı 1 ' den küçük ve makul düzeydedir. Bu duyarlıklar bulunurken altdevrelere ilişkin (4.18) ile verilen $D(s)$ payda polinomundan yararlanılır. (4.18) ' de,

$$w_p = \sqrt{\frac{g_{m1}g_{m2}g_{m3}}{g_{m4}C_1C_2}} \quad (5.8.a)$$

$$Q_p = \sqrt{\frac{g_{m1}g_{m4}C_2}{g_{m2}g_{m3}C_1}} \quad (5.8.b)$$

olarak verilir.

Tablo 5.8. Minimum OTA-C biquadlarına ilişkin w_0 ve Q duyarlıkları

Devre parametresi, $x \rightarrow$	g_{m1}	g_{m2}	g_{m3}	g_{m4}	C_1	C_2
$S_x^{w_0}$	1/2	1/2	1/2	-1/2	-1/2	-1/2
S_x^Q	1/2	-1/2	-1/2	1/2	-1/2	1/2

BÖLÜM 6. TASARLANAN OTA - C QUAD FİLTRESİNİN MAKSİMUM GİRİŞ GERİLİMİNİN HESAPLANMASI

Bu bölümde, tasarlanan OTA - C filtresinin girişine uygulanacak, gerilim genliğinin izin verilen maksimum değeri bulunacaktır. Böylece filtrenin "clipping" ve "slew-rate" problemlerinin ortadan kaldırılması amaçlanmaktadır.

6.1. GENEL KAVRAMLAR

OTA - C filtreleri gerçekleştirirken, dikkat edilecek önemli bir nokta giriş işaretinin seviyesidir. Pratikte, OTA'ların çıkış işaretleri belirli değerleri aştığında bu elemanlar doğrusal davranış gösteremezler. Devredeki herhangi bir OTA'nın çıkış gerilimi doyma değerini geçtiğinde, filtre çıkışında kırılmış bir işaret elde edilir. Eğer çıkış akımı doyma değerini geçerse, "slew - rate" sorunu ortaya çıkar [8].

Bir OTA - C filtresinin doğrusal çalışabilmesi için, devredeki hiçbir OTA'nın çıkış işaretleri (çıkış gerilim ve akımı) doyma değerlerini aşmamalı ve maksimum filtre giriş gerilimi buna göre belirlenmelidir [8],

$$|V_k| \leq V_{ks} \quad , \quad k=1,2,\dots,n \quad (6.1.a)$$

$$|I_k| \leq I_{ks} \quad , \quad k=1,2,\dots,n \quad (6.1.b)$$

Burada n, devrede kullanılan OTA sayısını göstermektedir. $V_k = V_k(j\omega)$ ve $I_k = I_k(j\omega)$ ise sırasıyla k nolu OTA'ya ait fazör gerilim ve akımlardır. (6.1.a) ve (6.1.b) ifadelerinin her ikisi de tasarımcının belirlediği bir frekans aralığında, $\omega \in [\omega_1, \omega_2]$, aynı anda sağlanmalıdır [8].

V_{ks} ve I_{ks} ise k numaralı OTA 'ya ait sırasıyla doyma gerilim ve akımıdır. Bütün OTA 'ların benzer olması durumunda bu değerler birbirine eşit olacaktır [8].

$$V_{1s} = \dots = V_{ns} = V_s \quad (6.2.a)$$

$$I_{1s} = \dots = I_{ns} = I_s \quad (6.2.b)$$

Giriş geriliminin genliği cinsinden bu koşullar,

$$|V_i| |H_k| \leq V_{ks} = V_s, \quad k=1,2,\dots,n \quad (6.3.a)$$

$$|V_i| |Y_k| \leq I_{ks} = I_s, \quad k=1,2,\dots,n \quad (6.3.b)$$

şeklinde olacaktır. Burada $|V_i|$, filtre giriş geriliminin genliğidir. $H_k = H_k(j\omega)$ ve $Y_k = Y_k(j\omega)$ ise k numaralı OTA 'ya ait sırasıyla gerilim transfer ve transfer admitans fonksiyonlarını ifade etmektedir [8].

$$H_k = V_k / V_i, \quad k=1,2,\dots,n \quad (6.4.a)$$

$$Y_k = I_k / V_i, \quad k=1,2,\dots,n \quad (6.4.b)$$

(6.4) eşitsizliklerinden,

$$|V_i| \leq V_s / |H_k|, \quad k=1,2,\dots,n \quad (6.5.a)$$

$$|V_i| \leq I_s / |Y_k|, \quad k=1,2,\dots,n \quad (6.5.b)$$

ifadeleri elde edilir. (6.5) 'deki eşitsizliklerin ortak çözümü sonucu, "clipping" ve "slew - rate" sorunlarını önleyecek maksimum filtre giriş geriliminin genliği,

$$|V_i|_{max} = \min \left[V_s / |H_k(j\omega)|_{max}, I_s / |Y_k(j\omega)|_{max}, k=1,2,\dots,n \right] \quad (6.6)$$

şeklinde ifade edilebilir. Burada $|H_k|_{max}$ ve $|Y_k|_{max}$ ise sırasıyla $|H_k|$ ve $|Y_k|$ 'nın ilgilenilen $\omega \in [\omega_1, \omega_2]$ frekans bandındaki maksimum değerleridir [8].

6.2. BUTTERWORTH, CHEBYSHEV VE ELİPTİK TÜRDE YÜKSEK DERECELİ QUAD FİLTRELERİNİN MAKSİMUM GİRİŞ GERİLİMİNİN HESAPLANMASI

Her üç yaklaşıklık için de filtre girişine uygulanan gerilimin alabileceği maksimum değer bulunurken 6.1 başlığı altında verilen prosedürden yararlanılacaktır. Butterworth, Chebyshev ve Eliptik türden filtrelerle ilişkin OTA - C Quad devresi Şekil 5.2.c ve Şekil 5.2.d 'de verilmişti. Bu devrelerdeki bütün OTA 'lara ait gerilim transfer (H) ve transfer admitans (Y) fonksiyonları kolaylıkla bulunabilir. Bulunan bu fonksiyonlardan yararlanarak (6.6) denklemi yardımıyla maksimum filtre giriş geriliminin genliği, $|V_i|_{\max}$, hesaplanabilir.

Tanım.

Filtredeki tüm OTA 'lara ait $V_s / |H_k|_{\max}$ ve $I_s / |Y_k|_{\max}$ ifadeleri hesaplanmış olsun.

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

ya da

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

(6.7)

ise $k = a$ numaralı OTA 'ya " Dominant OTA " denir. $|H_a|_{\max}$ ya da $|Y_a|_{\max}$ 'a da " Dominant Terim " denir.

Anlaşılabacağı üzere maksimum filtre giriş geriliminin genliğini dominant OTA 'ya ait dominant terim belirlemektedir. O halde maksimum giriş geriliminin genliği,

$$\begin{aligned} |V_i|_{\max} &= V_s / |H_a|_{\max}, \quad \text{dominant terim H ise} \\ &I_s / |Y_a|_{\max}, \quad \text{dominant terim Y ise} \end{aligned} \quad (6.8)$$

ifadesinden de bulunabilir.

6.2.1. Butterworth ve Chebyshev Filtreler için Maksimum Giriş Geriliminin Bulunması

Şekil 5.2.c' de Butterworth ve Chebyshev filtre tasarımında kullanılan OTA - C Quad devresi görülmektedir. Bu devredeki tüm OTA 'lara ilişkin gerilim transfer ve transfer admitans fonksiyonları aşağıda verilen adımlarla bulunabilir. Algoritma i numaralı Quad için verilmiştir.

1. i numaralı Quad için ,

$$H_j^i$$

değeri bilinmektedir.

2.

$$H_1^i = \frac{1}{(1 + f_i \cdot H_{41}^i)} \cdot H_j^i \quad (6.8.a)$$

$$H_2^i = H_{21}^i \cdot H_1^i \quad (6.8.b)$$

$$H_3^i = -\frac{b_{10}^i}{b_{11}^i \cdot s} \cdot H_2^i \quad (6.8.c)$$

$$H_4^i = H_{42}^i \cdot H_2^i \quad (6.8.d)$$

$$H_5^i = -\frac{b_{20}^i}{b_{21}^i \cdot s} \cdot H_4^i \quad (6.8.e)$$

gerilim transfer fonksiyonları $w \in [w_1, w_2]$ frekans bandı için hesaplanır.

3.

$$Y_1^i = -g_1^i \cdot H_2^i \quad (6.8.f)$$

$$Y_2^i = g_2^i \cdot (H_3^i - H_2^i) \quad (6.8.g)$$

$$Y_3^i = g_3^i \cdot H_1^i \quad (6.8.h)$$

$$Y_4^i = -g_4^i \cdot H_4^i \quad (6.8.i)$$

$$Y_5^i = g_5^i \cdot (H_5^i - H_4^i) \quad (6.8.i)$$

$$Y_6^i = g_6^i \cdot H_2^i \quad (6.8.j)$$

$$Y_x^i = g_x^i \cdot H_j^i \quad (6.8.k)$$

$$Y_y^i = -g_y^i \cdot H_1^i \quad (6.8.l)$$

$$Y_z^i = -g_z^i \cdot H_4^i \quad (6.8.m)$$

transfer admitans fonksiyonları yine belirlenen tasarım bandı için hesaplanır. Burada,

H_p^i : i numaralı Quad devresinin p numaralı tüm filtre girişine göre gerilim transfer fonksiyonudur ve V_p^i / V_{in} olarak tanımlanır.

V_p : i numaralı Quad devresinin p numaralı düğüm gerilimini gösterir.

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

H_{21}^i : i numaralı Quadın birinci biquadratik transfer fonksiyonudur ve,

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

olarak tanımlanır.

H_{42}^i : i numaralı Quadın ikinci biquadratik transfer fonksiyonudur ve,

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

olarak tanımlanır.

H_{41}^i : i numaralı Quadın 4 nolu düğümünün 1 numaralı düğümüne göre gerilim transfer fonksiyonudur.

Y_m^i : i numaralı Quad 'a 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 'a ait m numaralı OTA 'nın çıkış akımıdır.

g_m^i : i numaralı Quad 'a ait m numaralı OTA elemanına ilişkin iletkenliği gösterir.

Verilen algoritma $i=1$ 'den başlamak üzere sonuncu Quada kadar tekrar edilip, tüm filtreye ilişkin H ve Y ' ler hesaplanır. i numaralı Quadın giriş gerilimi V_j , i-1 numaralı Quadın çıkış gerilimi V_4 'e eşit olduğu için algoritmanın ilk adımında H_j^i bilinmektedir. Diğer büyüklükler bu değerden yararlanarak hesaplanmaktadır.

Filtredeki tüm OTA 'lara ilişkin H ve Y ' lerin tasarım bandı için maksimum değerleri, $|H_k|_{\max}$ ve $|Y_k|_{\max}$, hesaplanır. (6.7) eşitsizliğinden yararlanarak dominant OTA ve dominant terim belirlenir ve (6.8) formülü kullanılarak $|V_i|_{\max}$ hesaplanır.

6.2.2. Eliptik Filtreler İçin Maksimum Giriş Geriliminin Hesaplanması

Şekil 5.2.d 'de Eliptik filtre tasarımında kullanılan OTA - C Quad devresi görülmektedir. Önceki alt bölümde sözü edilen algoritma, Eliptik filtre için aşağıdaki gibi olacaktır.

1. i numaralı Quad için ,

$$H_j^i$$

değeri bilinmektedir.

2.

$$H_1^i = \frac{1}{(1 + f_i \cdot H_{41}^i)} \cdot H_j^i \quad (6.9.a)$$

$$H_2^i = H_{21}^i \cdot H_1^i \quad (6.9.b)$$

$$H_3^i = \frac{1}{b_{11}^i \cdot s} \cdot (a_{10}^i \cdot H_1^i - b_{10}^i \cdot H_2^i) \quad (6.9.c)$$

$$H_4^i = H_{42}^i \cdot H_2^i \quad (6.9.d)$$

$$H_5^i = \frac{b_{11}^i}{a_{12}^i \cdot s} \cdot (H_2^i - H_3^i) \quad (6.9.e)$$

$$H_6^i = \frac{1}{b_{21}^i \cdot s} \cdot (a_{20}^i \cdot H_2^i - b_{20}^i \cdot H_4^i) \quad (6.9.f)$$

$$H_7^i = \frac{b_{21}^i}{a_{22}^i \cdot s} \cdot (H_4^i - H_6^i) \quad (6.9.g)$$

gerilim transfer fonksiyonları belirlenen tasarım bandı için hesaplanır.

3.

$$Y_1^i = -g_1^i \cdot H_2^i \quad (6.9.h)$$

$$Y_2^i = g_2^i \cdot (H_2^i - H_3^i) \quad (6.9.i)$$

$$Y_3^i = g_3^i \cdot (H_1^i - H_5^i) \quad (6.9.i)$$

$$Y_4^i = -g_4^i \cdot H_2^i \quad (6.9.j)$$

$$Y_5^i = g_5^i \cdot H_1^i \quad (6.9.k)$$

$$Y_6^i = -g_6^i \cdot H_4^i \quad (6.9.l)$$

$$Y_7^i = g_7^i \cdot (H_4^i - H_6^i) \quad (6.9.m)$$

$$Y_8^i = g_8^i \cdot (H_2^i - H_7^i) \quad (6.9.n)$$

$$Y_9^i = -g_9^i \cdot H_4^i \quad (6.9.o)$$

$$Y_{10}^i = g_{10}^i \cdot H_2^i \quad (6.9.p)$$

$$Y_x^i = g_x^i \cdot H_j^i \quad (6.9.r)$$

$$Y_y^i = -g_y^i \cdot H_1^i \quad (6.9.s)$$

$$Y_z^i = -g_z^i \cdot H_4^i \quad (6.9.t)$$

transfer admitans fonksiyonları belirlenen tasarım bandı için hesaplanır.

Örnek 6.1

Örnek 3.2 ' deki BG Chebyshev filtre gözönüne alınsın . Bu filtreye ilişkin kazanç değerleri atanmış Quad tasarım denklemleri,

$$T_{11}(s) = \frac{387875 s}{s^2 + 223936.6 s + 1.085657 \times 10^{13}} \quad (6.10.a)$$

$$T_{21}(s) = \frac{364221.6 s}{s^2 + 223936.6 s + 1.085657 \times 10^{13}} \quad (6.10.b)$$

olarak bulunur. Ayrıca,

$$Q_1 = 14.71368$$

$$k_1 = 1.234645$$

$$k_2 = 1.159353$$

$$f' = 0.7042816$$

şeklinde hesaplanmıştır. Bu değerlerin tümü Ek ' te yer alan " QUAD " programıyla hesaplanmıştır. Filtre Şekil 5.2.c 'deki devreyle gerçekleştirilebilir. Filtreye ilişkin OTA iletkenliklerinin ve kapasite elemanlarının değerleri,

$$g_1 = 100 \text{ uS}$$

$$g_2 = 57.73421 \text{ uS}$$

$$g_3 = 100 \text{ uS}$$

$$g_4 = 100 \text{ uS}$$

$$g_5 = 61.48361 \text{ uS}$$

$$g_6 = 100 \text{ uS}$$

$$g_x = 100 \text{ uS}$$

$$g_y = 100 \text{ uS}$$

$$g_z = 70.42815 \text{ uS}$$

$$C_1 = 2.062683 \text{ pF}$$

$$C_2 = 0.257815 \text{ nF}$$

$$C_3 = 2.062683 \text{ pF}$$

$$C_4 = 0.2745581 \text{ nF}$$

$$K = 0.9999987$$

$$g_a = 99.99987 \text{ uS}$$

$$g_b = 100 \text{ uS}$$

olarak bulunmuştur.

6.2.1 başlığı altında verilen algoritma ile tüm OTA'lara ait H ve Y'lerin maksimum değerleri Tablo 6.1'de verilmiştir.

Tablo 6.1. Örnek 3.1'deki filtreye ilişkin H ve Y'lerin maksimum değerleri

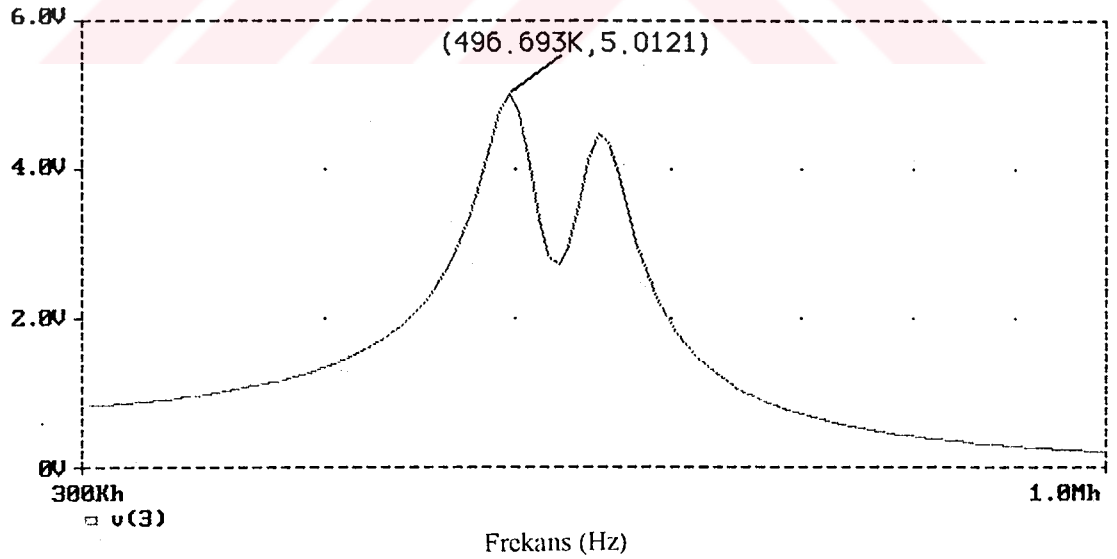
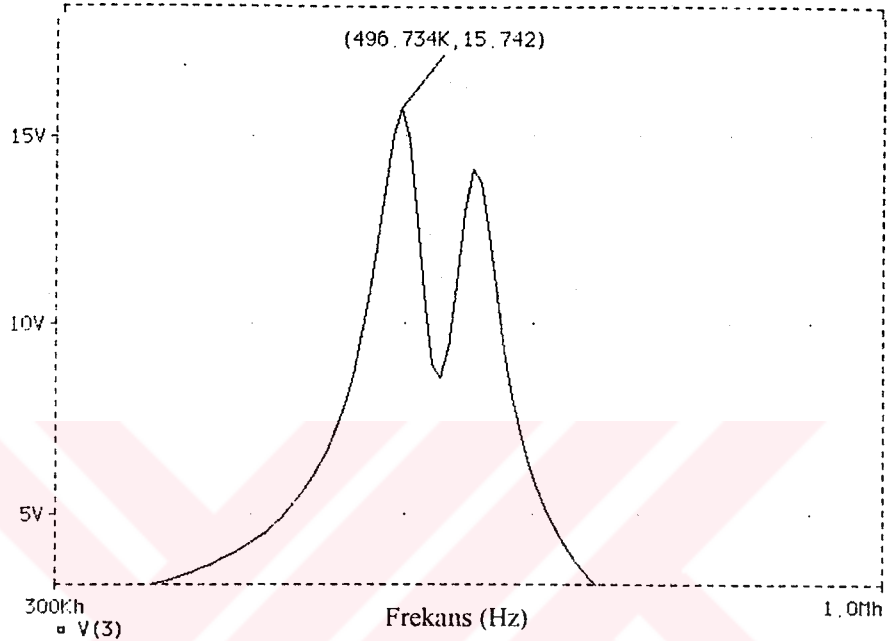
H_1	0.9999987	Y_1	101.3056 uS
H_1	1.223012	Y_2	908.2421 uS
H_2	1.013056	Y_3	122.3012 uS
H_3	15.69881	Y_4	99.99136 uS
H_4	0.9999136	Y_5	937.6974 uS
H_5	15.21862	Y_6	101.3056 uS

Ayrıca $Y_a = Y_b = 100 \text{ uS}$ olarak bulunur.

Filtreye ilişkin dominant terim H_3 ve dominant OTA 1 numaralı OTA'dır. "Clipping" ve "slew - rate" sorunlarına yol açmayacak, izin verilen en büyük giriş geriliminin genliği, $|V_i|_{\max}$, 318.4955 mV olarak hesaplanmıştır. $|V_i|_{\max}$ hesaplanırken tüm OTA'lara ait çıkış doyma gerilim ve akımları (V_s ve I_s) sırasıyla 5 V ve 1500 uA olarak seçilmiştir.

Şekil 6.1.a'da dominant terim V_3 'e ilişkin frekans cevabı görülmektedir. Bu şekil, filtre girişine birim impuls gerilimi uygulandığı varsayılarak elde

edilmiştir. Filtre girişine genliği maksimum 318.4955 mV olan impuls gerilimi uygulandığında H_3 'ün maksimum değeri 5 V olmaktadır. Bu durum Şekil 6.1.b' de görülmektedir. Diğer tüm OTA 'ların çıkış işaretlerinin doyma sınırının altında kaldığı gözlenmiştir. Bu anlamda filtredeki bütün OTA' ların çıkış gerilim ve akımları sırasıyla 5 V ve 1500 uA değerlerini aşmamaktadır.



Şekil 6.1. Örnek 3.2' deki filtreye ilişkin dominant terim H_3 'ün frekans cevabı

(a) Filtre girişine birim impuls gerilimi uygulandığında

(b) Filtre girişine 318.4955 mV değerinde impuls gerilimi uygulandığında

OTA 'nın çıkış doyma akım ve gerilim değerleri ne kadar büyükse, filtre girişine uygulanabilecek gerilim genliğinin değeri o kadar yüksek olacaktır. Bu şekilde daha iyi dinamik davranışa sahip devreler elde etmek mümkün olacaktır. Eğer tasarımda, çıkış doyma akım ve/veya gerilim değerleri küçük olan OTA kullanılırsa, filtre girişine uygulanabilecek gerilimin genliği belli bir değeri geçemeyecektir. Bunun bir dezavantaj olduğu görülmektedir. Örnek olarak tasarımda kullanılan tüm OTA 'ların çıkış doyma gerilim ve akımları sırasıyla 3V ve 1mA olsun. Bu durumda, bu filtrenin "clipping" ve "slew-rate" problemlerine maruz kalmaması , diğer anlatımla doğrusal çalışabilmesi için, girişine uygulanabilecek gerilim genliğinin en büyük değeri 191.0973 mV olarak , "QUAD" programı yardımıyla , hesaplanmıştır.

Öte yandan OTA iletkenliklerine uygun değer vermek suretiyle daha iyi dinamik davranış elde etmek mümkündür.

Örnek 6.2

Örnek 3.3 ' te verilen filtre için Örnek 5.1 ' de kazanç değerleri atanmış , Örnek 5.2 ' de ise eleman değerleri hesaplanmıştır. Bu filtreye ilişkin $|V_{i\max}|$ değeri bulunmak istensin. Filtreye ilişkin maksimum H ve Y değerleri "QUAD " programıyla kolayca hesaplanabilir. Ayrıca filtreye ilişkin dominant OTA, 3 numaralı Quada ait 6 numaralı OTA olarak bulunmuştur. $|V_{i\max}|$ ise 86.48604 mV olarak hesaplanmıştır.

Örnek 6.3

Örnek 3.2 ' de verilen özelliklere sahip filtrenin , Örnek 5.3 ' te dinamik değişimi artırıcı yönde tasarımı yapılmış ve eleman değerleri hesaplanmıştır. Bu filtrenin doğrusal çalışabilmesi için (clipping ve slew-rate sorunlarının önüne geçmek için) filtre girişine uygulanacak gerilim genliğinin en büyük değeri , EK' teki program ile , 516.8877 mV olarak bulunmuştur. Ayrıca 1 numaralı Quad bloğuna ait 3 numaralı OTA dominant OTA olup, bu elemana ilişkin çıkış gerilimi $|V_{i\max}|$ ' ı belirleyen büyüklüktür, yani H_3 dominant terimdir.

BÖLÜM 7. SONUÇLAR VE ÖNERİLER

Bu çalışma ile elde edilen sonuçlar ve bunlara ilişkin öneriler aşağıdaki gibi özetlenebilir:

1) Tezde, yüksek dereceden filtrelerin OTA ve kapasite elemanlarıyla tasarımı yapılmıştır. Quad yönteminin sunduğu avantajlar ile daha etkin bir tasarım ve gerçekleştirme sağlanmaktadır.

(i) Bu anlamda tezde tasarlanan filtreler tamamen tümleşik devre tekniğine uygundur. Öte yandan gerçekleştirilen devreler sadece OTA ve topraklı kapasite elemanlarından oluşmakta, diğer deyişle direnç elemanlarını içermemektedir. Böylece tümleştirme sonucunda elde edilen silikon kırılgan alanı da küçük kalacaktır.

(ii) Bu devreler OTA gibi ayarlanabilir alt devre elemanlarına sahip olduğu için kolayca ayarlanabilir. Otomatik ayarlama, [4], devreleriyle birleştirildiğinde; ısı, nem, üretim hataları gibi nedenlerden dolayı filtre karakteristiğinde oluşacak kaymalar giderilebilir.

(iii) Tezde elde edilen OTA-C devreleri İ.K.-RC elemanlarıyla gerçekleştirilen eşdeğerlerine oranla daha az sayıda eleman içermektedirler. Bu ise daha ekonomik ve kolay bir tasarımı getirmektedir. Tezde kullanılan OTA-C biquadratik altdevrelerin tasarım eşitlikleri, Kaynak [4] ile verilen İ.K.-RC elemanlarıyla oluşturulmuş altdevrelere (özellikle Thow-Thomas, TT devresi) göre oldukça kolaydır.

(iv) Tezde, Aktif-RC tasarımıdaki İ.K. gibi yüksek frekans davranışı iyi olmayan aktif elemanların yerine OTA elemanı kullanıldığı için, yüksek frekanslarda da çalışabilen devreler elde edilmiştir.

2) Öte yandan tezde sadece simetrik bandgeçiren yüksek dereceden filtrelerin tasarımı yapılmıştır. Bununla birlikte AG, YG, BS gibi diğer tüm filtre tipleri de kolayca gerçekleştirilebilir. Bu açıdan bakıldığında çalışma, "Genel OTA-C Quad Tasarım" a bir giriş niteliğindedir. Esasen sözü edilen filtrelerin Butterworth, Chebyshev ve Eliptik yaklaşıklıklar kullanılarak elde edilmesi incelenmiş ve bunlara

ilişkin bir bilgisayar programı da yapılmıştır. Tezde amaçlanan kapsamın dışında kaldığı için bu durum incelenmemiştir.

3) Tezde simetrik bandgeçiren filtreler ; Butterworth, Chebyshev yaklaşıklıklar yanı sıra Eliptik yaklaşıklığı da kullanarak tasarlanmıştır. Bu anlamda elde edilen biquadratik altdevrelere ilişkin ikinci derece transfer fonksiyonlarının pay ifadeleri Butterworth ve Chebyshev için $k.s$ şeklinde, Eliptik için ise $k.(s^2 + w_z^2)$ şeklinde terimler içerecektir. Kaynak [9] ile verilen CBR yöntemi de Quad tasarım yöntemini esas alarak simetrik bandgeçiren filtre tasarımını amaçlamakla birlikte sadece Butterworth ve Chebyshev türden filtreler için kullanılabilir. Bu bakımdan tezdeki çalışma daha etkin gözükmetedir.

4) Tezde, verilen BG filtre özelliklerine göre (f_{a1} , f_{p1} , f_{p2} , f_{a2} , A_p , A_a) Butterworth , Chebyshev ya da Eliptik yaklaşıklıklardan biri kullanılarak , simetrik BG filtreler Quad ve kaskad tasarım ile kolayca yapılabilir. Bu anlamda sadece 6 özelliğin bilinmesiyle yüksek dereceden simetrik BG bir filtre OTA-C elemanlarıyla tasarlanabilmektedir. Ayrıca elde edilen filtreler pratik gerçeklemeye uygun olmaktadır. "QUAD" programı ile elde edilen eleman değerleri gerçekleştirilebilir değerlerdir.

5) Pratikte kullanılan OTA elemanı; teknolojik nedenlerle üretimi sırasında ortaya çıkan parazitik elemanlar, yapılan bazı varsayım ve ihmaller nedeniyle idealden oldukça uzak davranış göstermektedir. Tezde, simetrik CMOS OTA'nın bu ideal olmayan özellikleri sonucunda iyi bir filtre tasarım elemanı olarak kullanılamayacağı SPICE simülasyonları ile ortaya konmuştur. Bu anlamda tasarlanan filtrelere ilişkin OTA-C filtrelerinde simetrik CMOS-OTA'nın kullanılamayacağı sonucuna varılmış ve filtre karakteristikleri ideal OTA' larla elde edilmiştir.

Bununla birlikte, literatürde daha verimli OTA elemanları geliştirmeye yönelik çalışmalar mevcuttur [20], [21].

6) Tezdeki filtre tasarımında iyi dinamik davranışa sahip filtre elde edilmesi amaçlanmıştır. Bununla birlikte Quad blokları, optimuma yakın dinamik davranış elde edilmesi durumuna göre sıralanmıştır. Optimum dinamik davranış elde etmek için yeterli bilgi Kaynak [1]' de bulunabilir.

7) Tezde gereklenen filtrelele ilgili ayrıntılı duyarlık hesapları yapılmamıştır. Bununla birlikte Kaynak [4] ve [28] yardımıyla bu filtrelerin duyarlığını hesaplamak mümkündür.

8) Tezde, tasarlanan OTA-C filtrelerinin doğrusal çalışabilmesi için maksimum filtre giriş geriliminin genliği hesaplanmıştır. Bu anlamda gereklenen filtrelerin "clipping" ve "slew-rate" problemleri ortadan kaldırılmaktadır.

9) Çalışma yeterli örneklerle desteklenmiş, örneklere ilişkin tüm tasarım hesaplamaları ve grafikler EK' te verilen bilgisayar programı yardımıyla elde edilmiştir.

10) EK' de verilen bilgisayar programına girilen, sadece 6 filtre özelliği ile simetrik BG OTA-C filtreler, hem Quad hem de kaskad yöntemleriyle etkin bir şekilde tasarlanabilmektedir. Program modüler halde olduğu için, AG, YG, BS gibi diğer filtre türleri de programa ilave edilebilir. İstenirse gereklenen filtreye ilişkin duyarlık hesaplamalarını yapmak üzere bir altprogram da ilave edilebilir.

KAYNAKLAR

- [1] GHAUSI, M.S., R.SCHAUMAN, K.R.LAKER, "Design of Analog Filters: Passive, Active-RC, Switched Capacitor", Prentice Hall, Inc., 1990
- [2] GHAUSI, M.S. and K.R.LAKER, "Modern Filter Design", Englewood Cliffs, N.J.:Prentice-Hall, 1981
- [3] B.LINARES-BARRANCO, A.RODRIGUEZ-VAZQUEZ, E.SANCHEZ-SINENCIO and J.L.HUERTAS, "CMOS OTA-C High-Frequency Sinusoidal Oscillators", IEEE J. Solid-State Circuits, vol.26, N0.2, February 1991
- [4] 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
- [5] GÖNÜLEREN, A.N., "Multiloop Feedback Unsymmetrical Active Filters Using Quads", Int. J. of Circuit Theory and Applications, vol.10,1-18 (1982)
- [6] GÖNÜLEREN, A.N., "Analysis and Design of Generalized Multiloop Feedback Active Filters", 1980 Midwest Symp. on Circuits and Systems, (Aug.1980), Toledo, pp.206-210
- [7] NATARAJAN, S., "Theory And Design of Linear Active Networks", McGraw-Hill, Singapore, 1989
- [8] C.ACAR, F.ANDAY, H.KUNTMAN, "On The Realization of OTA-C Filters", Int. J. of Circuit Theory And Applications, vol.21, 331-341 (1993)
- [9] S.POTOPULOS and T.DELIYANNIS, "Active RC Realization of High-Order Bandpass Filter Functions By Cascading Biquadratic sections", Int. J. of Circuit Theory And Applications, vol 12., 223-238 (1984)
- [10] PARKS, T.W., C.S.BURRUS, "Digital Filter Design", John Wiley & Sons, Inc.,1987
- [11] RORABAUGH,C.BRITTON, "Digital Filter Designer's Handbook", USA, McGraw-Hill, 1993
- [12] GREGORIAN, G., MARTIN, K.W. and TEMES, G.C., "Switched Capacitor Circuit Design", Proc. IEEE, vol.71,1989

- [13] 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
- [14] R.G.SPARKES and A.S.SEDRA, "Programmable Active Filters", IEEE J. Solid-State Circuits, vol. SC-8, pp.93-95, Feb.1973
- [15] S.FRANCO, "Use Transconductance Amplifier to Make Programmable Active Filters", Electronic Design, Vol.24., pp.98-101, Sept 1976
- [16] H.H.KUNTMAN, "Uygulamaya Özgü Tümdevre (ASIC) Teknolojisine Giriş", İstanbul Teknik Üniversitesi, Aralık 1993
- [17] R.L.GEIGER and E.SANCHEZ-SINENCIO, "Active Filter Design Using Operational Transconductance Amplifiers: A Tutorial", IEEE Circuits and Devices Magazine, March 1985
- [18] H.S.MALVAR, "Electronically Controlled Active Filters with Operational Transconductance Amplifiers", IEEE Trans. Circuits Syst., vol. CAS-29, pp.333-336, May 1982
- [19] 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
- [20] A.NEDUNGADI and T.R.VISWANATHAN, "Design of Linear CMOS Transconductance Elements", IEEE Transactions on Circuits and Systems, vol.CAS-31, No.10, October 1984
- [21] K.D.PETERSON and R.L.GEIGER, "CMOS OTA Structures with Improved Linearity ", Proc.27th Midwest Symp. on Ciircuits and Syst., June 1984
- [22] H.H.KUNTMAN, "CMOS OTA-C Süzgeçlerinde Kullanılmaya Yönelik Bir OTA Makromodeli, Elektrik Mühendisliği 5.Ulusal Kongresi
- [23] H.H.KUNTMAN, "İleri Lineer Tümdevre Tasarımı, Analog Devreler", Şubat 1984
- [24] R.L.GEIGER and J.FERRELL, "Voltage Controlled Filter Design Using Operational Transconductance Amplifiers", Proc. IEEE / ISCAS, pp.594-597, May 1983
- [25] 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)
- [26] R.NAWROCKI, U.KLEIN, " New OTA-Capacitor Realization A Universal Biquad", Electronics Letters, 2nd January 1986, vol.22, No.1

- [27] A.N.GÖNÜLEREN, "Sensitivity And Dynamic Range of Quad Filters And Some Comparative Results", Circuit Theory And Applications, vol.10, 175-200 (1982)
- [28] ACAR.C., "Duyarlık ve Tolerans Analizi", İstanbul Teknik Üniversitesi 1979



EK. BİLGİSAYAR PROGRAMI : " QUAD "

Burada verilen program, QUICKBASIC programlama dili kullanılarak yazılmıştır. Program , girilen BG filtre özelliklerine göre , Butterworth , Chebyshev ve Eliptik yaklaşıklık yöntemlerinden istenen birini kullanarak ; dinamik değişim maksimum olacak şekilde Quad veya Kaskad tasarım denklemlerini hesaplamaktadır. (Dinamik değişimin maksimum olabilmesi için Quad bloklarını artan değer katsayılarına göre sıralamaktadır.) Tasarım denklemlerinden hareketle OTA iletkenlikleri ve kapasite elemanlarının değerleri hesaplanmaktadır. Son olarak da , " clipping" ve " slew - rate " problemlerini ortadan kaldıracak şekilde , tasarlanan filtrenin girişine uygulanabilecek maksimum giriş geriliminin genliği hesaplanmaktadır. Ayrıca devredeki tüm düğümlere ilişkin H ve Y değerlerinin frekansa bağlı grafikleri yanısıra , filtrenin frekans cevabı da çizdirilmektedir.

Aşağıda programın akış mantığı verilmekte , izleyen sayfalarda da QuickBasic programlama dilinde yazılmış dökümü yer almaktadır.

1. BG Filtre özelliklerini girin: f_{a1} , f_{p1} , f_{p2} , f_{a2} , A_p , A_a
2. Yaklaşımın tipini girin : bw (Butterworth), chy (Chebyshev), ell (Eliptik)
3. Prototip AG filtre özelliklerini hesapla : Ω_p , Ω_a , A_p , A_a
4. Prototip AG filtrenin kutup , sıfır ve transfer fonksiyonu katsayılarını hesapla .
5. Prototip AG transfer fonksiyonundan AG-BG dönüşümü ile BG transfer transfer fonksiyonunu oluştur.
6. Yüksek dereceli filtrenin gerçekleştirileceği topolojiyi seçin :
Seçim QUAD ise 7.adıma git , KASKAD ise 9.adıma git.
7. QUAD TASARIM : T_b , T_{Q1i} , T_{Q2i} transfer fonksiyonlarının ifadelerini bul ve f_1 geribesleme katsayılarını hesapla.
8. 10. Adıma git.
9. KASKAD TASARIM : T_b , T_{C1i} , T_{C2i} transfer fonksiyonlarının ifadelerini bul
10. Devreye ilişkin kritik gerilimleri hesapla ve bu değerleri kullanarak dinamik değişim maksimum olacak şekilde biquadratik kazanç katsayılarını hesapla: k_{1i} , k_{2i} , K .
11. Filtreye ilişkin OTA iletkenliklerinin ve kapasite elemanlarının değerlerini hesapla .
12. Tasarlanan filtreye ilişkin dominant Quad bloğu ve dominant terimi bul.
13. Filtre girişine uygulanacak gerilim genliğinin maksimum değerini bul.
14. Filtrenin frekans cevabını çiz.

TECHNICAL UNIVERSITY OF ISTANBUL
INSTITUTE OF SCIENCE AND TECHNOLOGY

SYMMETRICAL BANDPASS OTA-C FILTER
REALIZATION USING QUAD METHOD

M.Sc. Thesis
Ramazan KÖPRÜ

Supervisor: Prof. Dr. Ali Nur GÖNÜLEREN

"-----MAIN PROGRAM-----"

```

DECLARE SUB dinamik ()
DECLARE SUB transyaz ()
DECLARE SUB dinamikbc ()
DECLARE SUB dinamikell ()
DECLARE SUB eleman ()
DECLARE SUB grafik ()
DECLARE SUB frame ()
DECLARE SUB dinamic ()
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 lpbptrans ()
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 a$, n$, tb$, wp, wa, oma, Ap, Aa, ho, n, m, w0, band, pi
COMMON SHARED w, wa1, wa2, wp1, wp2, po, so, mx, nx, qx, hy, s
COMMON SHARED maxgenlik, kq$, test, sw
DIM SHARED pre(40), pim(40), zre(40), zim(40)
"Typical biquadratic term : (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 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)

```

```

DIM SHARED b4(40), b3(40), b2(40), b1(40), b0(40)
"pole-zero pairing
DIM SHARED pole(40), zero(40)
"coefficients of cascade and quad blocks:
"Tq,c(s)=(b12s2+b11s+b10)/(a12s2+a11s+a10)
DIM SHARED b12(40), b11(40), b10(40), a12(40), a11(40), a10(40)
DIM SHARED b22(40), b21(40), b20(40), a22(40), a21(40), a20(40)
DIM SHARED wz1(40), wz2(40), wp1(40), wp2(40), q1(40), q2(40)
"feedback factors :1) f 2) fd=f/(k1*k2) (according to the dynamic range)
DIM SHARED f(20), fd(20), k(20)
"component values ( gm , C )
DIM SHARED gx(20), gy(20), gz(20), g1(20), g2(20), g3(20), g4(20)
DIM SHARED g5(20), g6(20), g7(20), g8(20), g9(20), g10(20)
DIM SHARED c1(20), c2(20), c3(20), c4(20)
COMMON SHARED gb1, gb2, gb3, cb1, cb2
COMMON SHARED ga, gb
'DIM SHARED h(20)
"Dynamic range coefficients:kd ---->K=(k1i*k2i)^-1 , i=1,2,3...nx
COMMON SHARED kd, kb, kaz
DIM SHARED k1(20), k2(20), k12(20)
"for grafics
DIM SHARED genlik(580)
s = 580 : test = 0 : pi = 3.141592654#
CLEAR , 40000
CALL basla : CALL delay : LOCATE 1, 1
start: CLS
"Filter specifications
CALL ekran : CALL specifications
"enter the approximation type-----
CLS
CALL ekran : COLOR 14, 1 : LOCATE 2, 20: PRINT "Enter the approximation type"
COLOR 15, 1 : LOCATE 4, 2 : PRINT "[bw.... Butterworth]"
LOCATE 5, 2 : PRINT "[chy]....Chebyshev..]"
LOCATE 6, 2 : PRINT "[ell]....Elliptic...]" : PRINT
LOCATE 8, 2 : yine: INPUT "your choice.....>", a$ : CLS
CALL ekran
IIF a$ = "bw" THEN
CALL bw(oma, Ap, Aa)
ELSEIF a$ = "chy" THEN
CALL chy(oma, Ap, Aa)
ELSEIF a$ = "ell" THEN
CALL ell(oma, Ap, Aa)
END IF
CALL anykey : CALL ekran : CALL polezero : CALL anykey : CALL ekran
CALL katsay : CALL anykey
CALL lpbptrans
CLS : CALL ekran : LOCATE 2, 20 : COLOR 14, 1 : PRINT "Kaskad/Quad gerçekteleme "
COLOR 15, 1 : LOCATE 4, 4 : INPUT "[Seçiminiz.....kas/quad]:", kq$
IF kq$ = "kas" THEN
CALL kaskad
ELSEIF kq$ = "quad" THEN
CALL quads
ELSE
END IF
CALL ekran : CALL dinamik
CALL eleman
CALL transyaz

```

```

'yeniden:      IF a$ = "ell" THEN
CALL dinamikell
ELSE
CALL dinamikbc
END IF
CALL grafik : CALL anykey : CLS : CALL ekran
LOCATE 2, 10 : COLOR 14, 1 : PRINT "Choose one of the following..."
COLOR 15, 1 : LOCATE 5, 2 : PRINT "[ 1 ...Continue seeing graphics]"
LOCATE 6, 2 : PRINT "[ 2 ...New Design          ]" : LOCATE 7, 2
PRINT "[ 3 ...Exit from the program  ]" : LOCATE 10, 2 : INPUT "[1,2,3]"; sec
IIF sec = 1 THEN
GOTO devam
ELSEIF sec = 2 THEN
GOTO start
ELSEIF sec = 3 THEN
END
ELSE
END IF
devam: test = test + 1
VIEW (0, 0)-(639, 349) : WINDOW (0, 0)-(639, 349)
GOTO yeniden
CLS : END
-----END OF MAIN PROGRAM-----
-----subroutine "BASLA"-----
SUB basla
"Opening menu of the program
CLS : SCREEN 9 : COLOR 14, 1
LOCATE 10,10 : PRINT "WELCOME TO THE ACTIVE FILTER DESIGN"
PRINT "Developed by Ramazan KÖPRÜ, İ.T.Ü. , 1994 "
-----subroutine "SPECIFICATIONS"-----
SUB specifications
'Specifications of the Bandpass filter
pi = 8 * ATN(1) : LOCATE 2, 20 : COLOR 14, 1
PRINT "Design of Bandpass filters using Quads"
COLOR 15, 1 : CALL delay : CALL delay : CALL delay : LOCATE 2, 20 : COLOR 14, 1
PRINT " Enter the Bandpass specifications " : COLOR 15, 1 : PRINT
"enter bp specifications
LOCATE 4, 2 : INPUT "fa1=", fa1: wa1 = pi * fa1
LOCATE 5, 2 : INPUT "fp1=", fp1: wp1 = pi * fp1
LOCATE 6, 2 : INPUT "fp2=", fp2: wp2 = pi * fp2
LOCATE 7, 2 : INPUT "fa2=", fa2: wa2 = pi * fa2
LOCATE 8, 2 : INPUT "Ap=", Ap
LOCATE 9, 2 : INPUT "Aa=", Aa
'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
-----subroutine "BW" (Butterworth)-----
SUB bw (oma, Ap, Aa)
'Butterworth approximation
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

```

-----subroutine "CHY" (Cheyshev) -----

```

SUB chy (oma, Ap, Aa)
"Chebyshev approximation
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 even or odd ?.....
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)))
"calculation of sinh(v) and cosh(v) .....: 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

```

-----subroutine "ELL" (Elliptic)-----

```

SUB ell (oma, Ap, Aa)
"Elliptic approximation
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 even or odd ? .....
IF n MOD 2 = 0 THEN n$ = "even" ELSE n$ = "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))
"calculation of po .....
pay = 0 : payda = 0
FOR m1 = 0 TO 4
pay = pay + ((-1) ^ m1) * (q ^ (m1 * (m1 + 1))) * .5 * (EXP((2 * m1 + 1) * V))
NEXT
FOR m1 = 1 TO 4
payda = payda + ((-1) ^ m1) * (q ^ (m1 ^ 2)) * .5 * (EXP(2 * m1 * V) + EXP(-2

```

```

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
'FOR i = 1 TO r
IF n$ = "odd" THEN mu = i ELSE mu = i - .5
pay = 0: payda = 0
FOR m1 = 0 TO 4
pay = pay + ((-1) ^ m1) * (q ^ (m1 * (m1 + 1))) * SIN((2 * m1 + 1) * mu * pi /
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
"calculation of poles and zeros
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

-----subroutine "PROTOTIP-----
SUB prototip
"poles /zeros of the prototype LP filter
LOCATE 2, 20 : COLOR 14, 1 : PRINT "poles/zeros of the prototype LP filter"
COLOR 15, 1 : PRINT : mx = (m - m MOD 2) / 2 : st = 3
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 : PRINT
st = st + 1 : nx = (n - n MOD 2) / 2
FOR i = 1 TO nx

```

```

st = st + 1 : LOCATE st, 4 : LOCATE
PRINT "p"; i; ", "; "p"; n + 1 - i; "="; pre(i); CHR$(241); "j"; ABS(pim(i))
NEXT : PRINT : LOCATE st + 2, 4
PRINT "ho="; ho : PRINT
LOCATE st + 2, 4
IF n$ = "odd" THEN
PRINT "po="; "p"; nx + 1; " = "; po
ELSE
END IF
PRINT
END SUB

```

```

-----subroutine "POLEZERO "-----
SUB polezero
"This subroutine pairs the poles and zeros of the LP prototype func.
"The aim is ordering the poles and zeros according to their Qs from the min.Q
"to max.Q ,respectively.
"1) For BW and CHY this order is : p1 , p2 , p3 ,.....pn
" Because the Qi's order is : (Q1>Q2>Q3>.....>Qn)
"2) For ELL this order is : zn-pn , z(n-1)-p(n-1) ,.....z2-p2 , z1-p
" Because the Qi's order is : (Qn>Q(n-1)>.....>Q1)
FOR i = 1 TO nx
pole(i) = i : zero(i) = i
NEXT
IF a$ = "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, 20 : COLOR 14, 1 : PRINT " Pole-zero pairing " : COLOR 15, 1
IF a$ = "bw" OR a$ = "chy" THEN
st = 3
FOR i = 1 TO nx
st = st + 1 : LOCATE st, 4 : PRINT "(1/p"; pole(i); ")"
NEXT
ELSE
st = 3
FOR i = 1 TO nx
st = st + 1 : LOCATE st, 4 : PRINT "(z"; zero(i); "/p"; pole(i); ")"
NEXT
END IF
END SUB

```

```

-----subroutine "KATSAY" -----
SUB katsay
"calculation of the coefficients of the prototip LP function,a,b,c,
"using obtained poles and zeros
FOR i = 1 TO nx
l = pole(i) : a(i) = -2 * pre(l) : b(i) = modul(pre(l), pim(l)) ^ 2
NEXT
IF a$ = "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, 20 : COLOR 14, 1: PRINT "LP prototip fonk. biquadratik terimleri"
COLOR 15, 1
st = 3
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

```

-----subroutine "LPBPTRANS"-----

```

SUB lpbptrans
'forming of Tbp(s) from Tlp(s) using lp-bp transformation
'where ; 1) Tlp(s)=1/(s2+as+b) , for bw or chy
'          2) Tlp(s)=(s2+c)/(s2+as+b) , for ell
"Tbp(s)=(b4s4+b3s3+b2s2+b1s+b0)/(a4s4+a3s3+a2s2+a1s+a0)
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 a$ = "bw" OR a$ = "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 a$ = "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

```

-----subroutine "KASKAD"-----

```

SUB kaskad
'this subroutine calculates the coefficients of the cascade blocks
"Tc1(s)=(b12s2+b11s+b10)/(a12s2+a11s+a10) ,
'Tc2(s)=(b22s2+b21s+b20)/(a22s2+a21s+a20)
DIM xx(40), yy(40), zz(40)
FOR i = 1 TO nx
xx(i) = 4 * w0 ^ 2 + b(i) * band ^ 2 : yy(i) = (a(i) * band * w0) ^ 2
zz(i) = SQR(xx(i) / 2 + SQR((xx(i) / 2) ^ 2 - yy(i)))

```

```

wp1(i) = zz(i) / 2 + SQR((zz(i) / 2) ^ 2 - w0 ^ 2) : wp2(i) = w0 ^ 2 / wp1(i)
q1(i) = (wp1(i) + wp2(i)) / (a(i) * band) : 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 a$ = "el" THEN
FOR i = 1 TO nx
xx(i) = 4 * w0 ^ 2 + c(i) * band ^ 2 : yy(i) = 0 : zz(i) = SQR(xx(i))
wz1(i) = zz(i) / 2 + SQR((zz(i) / 2) ^ 2 - w0 ^ 2) : wz2(i) = w0 ^ 2 / 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
ELSE
FOR i = 1 TO nx
b12(i) = 0 : b11(i) = band : b10(i) = 0 : b22(i) = 0 : b21(i) = band : b20(i) = 0
NEXT
END IF
END SUB

```

```

-----subroutine "QUADS"-----
SUB quads
"this subroutine calculates the coefficients of the quad blocks
"Tq1(s)=(b12s2+b11s+b10)/(a12s2+a11s+a10)
"Tq2(s)=(b22s2+b21s+b20)/(a22s2+a21s+a20)
"wp1=wp2=wp=w0 ve q1=q2=q=2*w0/am3 tür
DIM xx(20), yy(20), zz(20)
FOR i = 1 TO nx
f(i) = geribes(a(i), b(i), c(i)) : k(i) = 1 / (1 - f(i) * b4(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)
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 a$ = "el" THEN
FOR i = 1 TO nx
xx(i) = 4 * w0 ^ 2 + c(i) * band ^ 2 : yy(i) = 0
zz(i) = SQR(xx(i)) : wz1(i) = zz(i) / 2 + SQR((zz(i) / 2) ^ 2 - w0 ^ 2)
wz2(i) = w0 ^ 2 / wz1(i)
b12(i) = 1 * k(i) : b11(i) = 0 : b10(i) = wz1(i) ^ 2 * k(i) : b22(i) = 1 : b21(i) = 0
b20(i) = wz2(i) ^ 2
NEXT
ELSE
FOR i = 1 TO nx
b12(i) = 0 : b11(i) = band * k(i) : b10(i) = 0 : b22(i) = 0 : b21(i) = band : b20(i) = 0
NEXT
END IF
END SUB

```

```

-----subroutine "DINAMIK"-----
SUB dinamik
"calculates the critical voltages of biquadratic subcircuits
"to find the biquadratic gain factors,k1i,k2i,K,which maximize the dynamic ran
"voltages will be calculated for 580 points in the frequency axis
s = 580
"V(i,s),( s=1,2,..600 i=1,2,..nx ) : i=ith quad block
"s=complex frequency variable
"Overall filter input voltage : Vi = a + j b -----
"where, a=reel part of the input voltage expression
"      b=imaginary part of the input voltage expression
vire = 1

```

```

viim = 0
"Hb (or Tb) input biquadratic block
"Hb= ( nbre + j nbim )/( dbre + j dbim ) = hbre + j hbim
"Quad blocks--
"input voltage of any quad block : vj
"1.biquad :H1=( n1re + j n1im )/( d1re + j d1im )
"input and output voltages of 1.biquad :v1,v2
"2.biquad :H2=( n2re + j n2im )/( d2re + j d2im )
"input and output voltages of 2.biquad :v2,v4
"transfer function of a quad : H41=H42*H1
"the term 1/(1+fk*H41) is calculated as the following :
"1. UO=fk*H41
"2. U1=1+UO
"3. UU=hk/U1
"Critical biquadratic voltages of the quad block :
DIM vjmax(nx), v2max(nx), v4max(nx), vjmin(nx), v2min(nx), v4min(nx)
vb1max = 0 : vb1min = 100
FOR i = 1 TO nx
vjmax(i) = 0 : v2max(i) = 0 : v4max(i) = 0 : vjmin(i) = 100 : v2min(i) = 100 : v4min(i) = 100
NEXT
CALL ekran: COLOR 14, 1 : LOCATE 2, 30 : PRINT "Calculating the critical voltages..."
LOCATE 15, 1
FOR i = 1 TO nx
h(i) = 1
NEXT
minfr = wp1
'(lower edge frequency of the interested band)
maxfr = wp2
'(upper " " " " " ")
"frequency step
a = (maxfr - minfr) / s
w = minfr - a
FOR k = 1 TO s
w = w + a
IF tb$ = "var" THEN
vb1re = vire * ho : vb1im = viim * ho
"Hb=Nb/Db hesaplaniyor
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 hesabi
'vb2re = mulre(vb1re, vb1im, hbre, hbim) : 'vb2im = mulim(vb1re, vb1im, hbre, hbim)
vb2re = vb1re * hbre : vb2im = vb1re * hbim
"Vj(1, k) and Vb2(k) are same
vare = vb2re : vaim = vb2im
vb1 = modul(vb1re, vb1im)
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
"maximum voltages calculated for Tb....
ELSE
vare = vire * ho : vaim = viim * ho
END IF
FOR i = 1 TO nx
vjre = vare : vjim = vaim
"calculation of H1.....
n1re = b10(i) - b12(i) * w ^ 2 : n1im = b11(i) * w
d1re = a10(i) - a12(i) * w ^ 2 : d1im = a11(i) * w
h1re = divre(n1re, n1im, d1re, d1im) : h1im = divim(n1re, n1im, d1re, d1im)
"calculation of H2.....
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)
"calculation of H41=H42*H21.....
"tre=H41re tim=H41im dir..
tre = mulre(h1re, h1im, h2re, h2im) : tim = mulim(h1re, h1im, h2re, h2im)
"calculation of the 1/(1+f*H1*H2) term...
uore = f(i) * tre : uoim = f(i) * tim : u1re = 1 + uore : u1im = uoim
uure = divre(1, 0, u1re, u1im) : uuim = divim(1, 0, u1re, u1im)
"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)
v are = v4re : vaim = v4im
"maximum and minimum values of the critical voltages....
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
NEXT : NEXT
"biquadratic gain factors, k1i,k2i,K, will be calculated.....
kd = 1
IF n$ = "odd" THEN
kb = vb1max / vb2max : bb2 = bb2 * kb : bb1 = bb1 * kb : bb0 = bb0 * kb
ELSE
kb = 1
END IF
FOR i = 1 TO nx

```

```
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)
```

```
"biquadratic gain factors are being assigned...."
```

```
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 :kaz = kd * ho
```

```
END SUB
```

```
-----subroutine "ELEMEN"-----
```

```
SUB eleman
```

```
"This subroutine calculates the OTA transconductans and capacitor values
```

```
g = .0001
```

```
"bw ve chy için eleman de°erlerini (gmi ve Ci) bulur.
```

```
gb = g : ga = gb * kaz
```

```
IF tb$ = "var" THEN
```

```
"component values for Tb block....."
```

```
gb3 = g : cb2 = gb3 / bb1 : gb2 = cb2 * ab1 : gb1 = g : cb1 = (gb1 * ab1) / ab0
```

```
"component values for quad/cascade blocks...."
```

```
FOR i = 1 TO nx
```

```
gy(i) = g : gx(i) = gy(i) * h(i) : gz(i) = F(i) * gy(i)
```

```
IF a$ = "ell" THEN
```

```
"ell için eleman degerleri
```

```
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
```

```
"component values for bw and chy type filters..."
```

```
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)
```

```
END IF
```

```
NEXT
```

```
ELSE
```

```
FOR i = 1 TO nx
```

```
gy(i) = g : gx(i) = gy(i) * h(i) : gz(i) = F(i) * gy(i)
```

```
IF a$ = "ell" THEN
```

```
"component values for ell type filters
```

```
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
```

```
"for bw and chy....."
```

```
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)
```

```
END IF
```

```
NEXT
```

```
END IF
```

```
"component values calculated....."
```

```
END SUB
```

```
-----subroutine "TRANSYAZ"-----
```

```
SUB transyaz
```

```
"prints the obtained Quad/Cascade transfer functions on the screen
```

```
CALL ekran : LOCATE 2, 20 : COLOR 14, 1 : PRINT " input Block " : COLOR 15, 1
```

```
st = 4 : c = 2 : st = st + 2: LOCATE st, c : PRINT "Input gain factor="; 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 n$ = "odd" THEN
CALL ekran: LOCATE 2, 20: COLOR 14, 1: PRINT "  Tb Block      "
COLOR 15, 1: c = 57: st = 4: 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$(253); "+"; ab1; "s+"; ab0
qb = SQR(ab0) / ab1: st = st + 2: LOCATE st, 2: PRINT "Qb="; qb
st = st + 1: LOCATE st, 2: PRINT "kb="; kb: st = 4
LINE (400, 1)-(400, 315), 14
st = st + 1: LOCATE st, c: PRINT "gb1="; gb1 * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "gb2="; gb2 * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "gb3="; gb3 * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "cb1="; cb1 / 1E-12; "pF"
st = st + 1: LOCATE st, c: PRINT "cb2="; cb2 / 1E-09; "nF"
ELSE
END IF
IF n$ = "odd" THEN
CALL anykey: CALL ekran
ELSE
END IF
st = 4
IF a$ = "bw" OR a$ = "chy" THEN
FOR i = 1 TO nx
LOCATE 2, 20: COLOR 14, 1
IF kq$ = "quad" THEN
PRINT "          "; i; ". Quad Block      "
ELSE
PRINT "          "; i; ". Cascade Block      "
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 "Q="; q1(i)
st = st + 1: LOCATE st, 2: PRINT "f="; F(i)
st = st + 1: LOCATE st, 2: PRINT "k="; k(i)
ELSE
st = st + 2: LOCATE st, 2: PRINT "Q="; q1(i)
END IF
st = st + 2: LOCATE st, 2: PRINT "k1="; k1(i): st = st + 1: LOCATE st, 2
PRINT "k2="; k2(i)
c = 57: st = 4
LINE (400, 1)-(400, 315), 14
st = st + 1: LOCATE st, c: PRINT "g1="; g1(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "g2="; g2(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "g3="; g3(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "g4="; g4(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "g5="; g5(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "g6="; g6(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "gx="; gx(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c: PRINT "gy="; gy(i) * 1000000!; "uS"
IF kq$ = "quad" THEN

```

```

st = st + 1: LOCATE st, c : PRINT "gz="; gz(i) * 1000000!; "uS"
ELSE
END IF
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 = 4 : CALL anykey : CALL ekran
ELSE
END IF
NEXT : PRINT
ELSE
"for ell
FOR i = 1 TO nx
LOCATE 2, 20 : COLOR 14, 1
IF kq$ = "quad" THEN
PRINT "      "; i; ". Quad Block      "
ELSE
PRINT "      "; i; ". Cascade Block      "
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 "Q="; q1(i)
st = st + 1: LOCATE st, 2 : PRINT "f="; F(i)
st = st + 1: LOCATE st, 2 : PRINT "k="; k(i)
ELSE
st = st + 2: LOCATE st, 2 : PRINT "Q="; q1(i)
END IF
st = st + 2: LOCATE st, 2 : PRINT "k1="; k1(i) : st = st + 1
LOCATE st, 2 : PRINT "k2="; k2(i)
st = 4 : c = 57
LINE (400, 1)-(400, 315), 14
st = st + 1: LOCATE st, c : PRINT "g1="; g1(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g2="; g2(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g3="; g3(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g4="; g4(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g5="; g5(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g6="; g6(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g7="; g7(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g8="; g8(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g9="; g9(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "g10="; g10(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "gx="; gx(i) * 1000000!; "uS"
st = st + 1: LOCATE st, c : PRINT "gy="; gy(i) * 1000000!; "uS"
IF kq$ = "quad" THEN
st = st + 1: LOCATE st, c : PRINT "gz="; gz(i) * 1000000!; "uS"
ELSE
END IF
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 = 4 : CALL anykey: CALL ekran
ELSE
END IF
CALL anykey
NEXT
END IF
CALL anykey
END SUB

```

```

-----subroutine "DINAMIKBC"-----
SUB dinamikbc
"calculates all OTA voltages and currents in the filter for bw and chy
IF test > 0 THEN GOTO goster ELSE GOTO hesapla
'hesapla:
s = 580
"saturation voltage and current for OTA :
vsat = 5 : isat = .0015 : vimax = 10
'Tüm gerilim ve akımların maximumunu bulmak için ilk max.lar sıfırlanıyor..
DIM vjmax(nx), v1max(nx), v2max(nx), v3max(nx), v4max(nx), v5max(nx), i1max(nx)
DIM i2max(nx), i3max(nx), i4max(nx), i5max(nx), i6max(nx), ixmax(nx), iymax(nx)
DIM izmax(nx)
vb1max = 0 : 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: 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
"open the files where the calculated voltages and currents will be saved
"for Kaz block...
OPEN "kaz.dat" FOR OUTPUT AS #21
"for Tb...
OPEN "tb.dat" FOR OUTPUT AS #20
"for TQ/TC...
"file numbers: 1,2,3....9
"file names : Q1.dat,Q2.dat,Q3.dat,....Q9.dat
"each file saves the all voltages and currents for a Quad/Cascade block
FOR i = 1 TO nx
d$ = "q" + CHR$(48 + i) + ".dat"
OPEN d$ FOR OUTPUT AS #i
NEXT
.....files opened.....
CALL ekran : COLOR 14, 1 : LOCATE 2, 30 : PRINT "Calculating all voltages and
currents..."
LOCATE 15, 1
minfr = .2 * wa1 : maxfr = 1.2 * wa2
a = (maxfr - minfr) / s
w = minfr - a
FOR k = 1 TO s
w = w + a
IF tb$ = "var" THEN
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
vb2re = mulre(vb1re, vb1im, hbre, hbim) : vb2im = mulim(vb1re, vb1im, hbre, hbim)

```

```

".... vb3=(-ab0/ab1*s)*vb2 .....auxiliary term yb=(-bb0/bb1*s)
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)
vare = vb2re : vaim = vb2im : 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 * ib2re
ib2im = gb2 * ib2im : ib2 = modul(ib2re, ib2im)
IF ib2 > ib2max THEN
ib2max = ib2
ELSE
END IF
ib3 = gb3 : ib3max = gb3 : 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
n1re = b10(i) - b12(i) * w ^ 2 : n1im = b11(i) * w
d1re = a10(i) - a12(i) * w ^ 2 : d1im = a11(i) * w
h1re = divre(n1re, n1im, d1re, d1im) : h1im = divim(n1re, n1im, d1re, d1im)
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)
tre = mulre(h1re, h1im, h2re, h2im) : tim = mulim(h1re, h1im, h2re, h2im)
uore = F(i) * tre : uoim = F(i) * tim : u1re = 1 + uore : u1im = uoim
uure = divre(h(i), 0, u1re, u1im) : uuim = divim(h(i), 0, u1re, u1im)
v1re = mulre(uure, uuim, vjre, vjim) : v1im = mulim(uure, uuim, vjre, vjim)
v2re = mulre(v1re, v1im, h1re, h1im) : v2im = mulim(v1re, v1im, h1re, h1im)
"-----v3=(-a10/a11*s)*v2 den hesaplanacak-----yardımcı terim y1=(-a10/a11*s).....
ny1re = -a10(i) : ny1im = 0 : dy1re = 0 : dy1im = a11(i) * w
y1re = divre(ny1re, ny1im, dy1re, dy1im) : y1im = divim(ny1re, ny1im, dy1re, dy1im)
v3re = mulre(y1re, y1im, v2re, v2im) : v3im = mulim(y1re, y1im, 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)-----
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
".....Maximum values of all voltages and currents.....
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
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(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
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
vyre = -gy(i) * v1re : iyim = -gy(i) * v1im : 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
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 nx
CLOSE #i
NEXT
CALL ekran : LOCATE 2, 2 : COLOR 14, 1
PRINT "Maximum values of Transfer voltage/admitans functions are written..."
COLOR 15, 1 : CALL anykey : CALL ekran : LOCATE 2, 10 : COLOR 14, 1
PRINT "Gain Block..." :
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, 15 : COLOR 14, 1 : PRINT "Tb Block..." : 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, 15 : COLOR 14, 1 : PRINT i; ".Block" : 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
ELSE
FOR i = 1 TO nx
CALL ekran : LOCATE 2, 15 : COLOR 14, 1 : PRINT i; "Block": 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) : 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
CALL ekran
LOCATE 2, 15 : COLOR 14, 1 : PRINT "Maximum input voltage for the overall filter "
COLOR 15, 1 : LOCATE 6, 6 : PRINT "Vimax="; vimax * 1000; "mV"
LOCATE 7, 6 : PRINT "-----"
IF domb = 0 THEN
LOCATE 8, 6 : PRINT "Dominant Block is ..."; "Tb"
LOCATE 9, 6 : PRINT "Dominant H/Y is ..."; " "; dom$
ELSE
LOCATE 8, 6 : PRINT "Dominant Block is ..."; domb
LOCATE 9, 6 : PRINT "Dominant H/Y is ..."; " "; dom$
END IF
CALL anykey
"grafik için v/i (gerilim/akım) secimi

```

```

goster:
CALL ekran : COLOR 14, 1
LOCATE 2, 2 : PRINT " Choose the quad/cascade block for which you want to see H/Y"
COLOR 15, 1 : LOCATE 4, 2
IF tb$ = "var" THEN
LOCATE 5, 2 : PRINT "[ 20]..Giris için" : LOCATE 6, 2 : PRINT "[ 0 ]..Tb  için"
FOR i = 1 TO nx
LOCATE i + 6, 2 : PRINT "["; i; "]."; i; ".Blok için"
NEXT
ELSE
LOCATE 5, 2 : PRINT "[ 20]..Giris için" :
FOR i = 1 TO nx
LOCATE i + 5, 2 : PRINT "["; i; "]."; i; ".Blok için"
NEXT
END IF
LOCATE 5, 30 : INPUT "[ seçiminiz... ] >", qx
CALL ekran : COLOR 14, 1
LOCATE 2, 20 : PRINT "choose the H/Y you want to see" : COLOR 15, 1 : LOCATE 4, 2
IF qx = 20 THEN
LOCATE 5, 2 : PRINT "[ 1 ]..Ya .için" : LOCATE 6, 2 : PRINT "[ 2 ]..Yb .için"
ELSEIF qx = 0 THEN
LOCATE 5, 2 : PRINT "[ 1 ]..Hb1 .için" : LOCATE 6, 2 : PRINT "[ 2 ]..Hb2 .için"
LOCATE 7, 2 : PRINT "[ 3 ]..Hb3 .için" : LOCATE 5, 30 : PRINT "[ 4 ]..Yb1 .için"
LOCATE 6, 30 : PRINT "[ 5 ]..Yb2 .için" : LOCATE 7, 30 : PRINT "[ 5 ]..Yb3 .için"
ELSEIF qx <> 0 AND qx <> 20 THEN
LOCATE 5, 2 : PRINT "[ 1 ]..Hj...için" : LOCATE 6, 2 : PRINT "[ 2 ]..H1...için"
LOCATE 7, 2 : PRINT "[ 3 ]..H2...için" : LOCATE 8, 2 : PRINT "[ 4 ]..H3...için"
LOCATE 9, 2 : PRINT "[ 5 ]..H4...için" : LOCATE 10, 2 : PRINT "[ 6 ]..H5...için"
LOCATE 5, 30 : PRINT "[ 7 ]..Y1...için" : LOCATE 6, 30 : PRINT "[ 8 ]..Y2...için"
LOCATE 7, 30 : PRINT "[ 9 ]..Y3...için" : LOCATE 8, 30 : PRINT "[ 10 ]..Y4...için"
LOCATE 9, 30 : PRINT "[ 11 ]..Y5...için" : LOCATE 10, 30 : PRINT "[ 12 ]..Y6...için"
LOCATE 11, 30 : PRINT "[ 13 ]..Yx...için" : LOCATE 12, 30 : PRINT "[ 14 ]..Yy...için"
IF kq$ = "quad" THEN
LOCATE 13, 30 : PRINT "[ 15 ]..Yz...için"
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
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)
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)
ELSE
END IF
NEXT
CLOSE #20 : CLOSE #21
FOR i = 1 TO nx
CLOSE #i
NEXT

```

END SUB

-----subroutine DINAMIKELL-----

SUB dinamikell

'calculates the all voltages and currents for Elliptic Filter

IF test > 0 THEN GOTO goster1 ELSE GOTO hesapla1

hesapla1:

s = 580: vsat = 5: isat = .0015: v1max = 10: vire = 1: viim = 0

DIM vjmax(nx), v1max(nx), v2max(nx), v3max(nx), v4max(nx), v5max(nx), v6max(nx)

DIM v7max(nx)

DIM i1max(nx), i2max(nx), i3max(nx), i4max(nx), i5max(nx), i6max(nx), i7max(nx)

DIM i8max(nx), i9max(nx), i10max(nx), ixmax(nx), iymax(nx), izmax(nx)

vb1max = 0: 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

OPEN "kaz.dat" FOR OUTPUT AS #21: OPEN "tb.dat" FOR OUTPUT AS #20

FOR i = 1 TO nx

d\$ = "q" + CHR\$(48 + i) + ".dat" : OPEN d\$ FOR OUTPUT AS #i

NEXT

CALL ekran: COLOR 14, 1: LOCATE 2, 30 : PRINT "calculating all voltages and currents..."

LOCATE 15, 1

minfr = .2 * wa1: maxfr = 1.2 * wa2

a = (maxfr - minfr) / s: w = minfr - a

FOR k = 1 TO s

w = w + a

IF tb\$ = "var" THEN

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 : vb2re = mulre(vb1re, vb1im, hbre, hbim)

vb2im = mulim(vb1re, vb1im, hbre, hbim)

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)

vare = vb2re: vaim = vb2im: 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 * ib2re : ib2im = gb2 * ib2im: ib2 = modul(ib2re, ib2im)

IF ib2 > ib2max THEN

ib2max = ib2

ELSE

```

END IF
ib3 = gb3: ib3max = gb3 : 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
n1re = b10(i) - b12(i) * w ^ 2: n1im = b11(i) * w
d1re = a10(i) - a12(i) * w ^ 2: d1im = a11(i) * w
h1re = divre(n1re, n1im, d1re, d1im): h1im = divim(n1re, n1im, d1re, d1im)
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)
tre = mulre(h1re, h1im, h2re, h2im): tim = mulim(h1re, h1im, h2re, h2im)
uore = F(i) * tre: uoim = F(i) * tim: u1re = 1 + uore: u1im = uoim
uure = divre(h(i), 0, u1re, u1im): uuim = divim(h(i), 0, u1re, u1im)
v1re = mulre(uure, uuim, vjre, vjim): v1im = mulim(uure, uuim, vjre, vjim)
v2re = mulre(v1re, v1im, h1re, h1im): v2im = mulim(v1re, v1im, h1re, h1im)
ny1re = b10(i): ny1im = 0: dy1re = 0: dy1im = a11(i) * w
y1re = divre(ny1re, ny1im, dy1re, dy1im)
y1im = divim(ny1re, ny1im, dy1re, dy1im)
ny2re = -a10(i): ny2im = 0: dy2re = 0: dy2im = a11(i) * w
y2re = divre(ny2re, ny2im, dy2re, dy2im)
y2im = divim(ny2re, ny2im, dy2re, dy2im)
y3re = mulre(y1re, y1im, v1re, v1im): y3im = mulim(y1re, y1im, v1re, v1im)
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)
ny1re = a11(i): ny1im = 0: dy1re = 0: dy1im = b12(i) * w
y1re = divre(ny1re, ny1im, dy1re, dy1im) : y1im = divim(ny1re, ny1im, dy1re, dy1im)
y2re = mulre(y1re, y1im, v2re, v2im): y2im = mulim(y1re, y1im, v2re, v2im)
y3re = mulre(y1re, y1im, v3re, v3im): y3im = mulim(y1re, y1im, v3re, v3im)
v5re = addre(y2re, y2im, -y3re, -y3im)
v5im = addim(y2re, y2im, -y3re, -y3im)
ny1re = b20(i): ny1im = 0: dy1re = 0: dy1im = a21(i) * w
y1re = divre(ny1re, ny1im, dy1re, dy1im)
y1im = divim(ny1re, ny1im, dy1re, dy1im)
ny2re = -a20(i): ny2im = 0: dy2re = 0: dy2im = a21(i) * w
y2re = divre(ny2re, ny2im, dy2re, dy2im)
y2im = divim(ny2re, ny2im, dy2re, dy2im)
y3re = mulre(y1re, y1im, v2re, v2im): y3im = mulim(y1re, y1im, 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)
ny1re = a21(i): ny1im = 0: dy1re = 0: dy1im = b22(i) * w
y1re = divre(ny1re, ny1im, dy1re, dy1im)
y1im = divim(ny1re, ny1im, dy1re, dy1im)
y2re = mulre(y1re, y1im, v4re, v4im)
y2im = mulim(y1re, y1im, v4re, v4im)
y3re = mulre(y1re, y1im, v6re, v6im)
y3im = mulim(y1re, y1im, v6re, v6im)
v7re = addre(y2re, y2im, -y3re, -y3im)
v7im = addim(y2re, y2im, -y3re, -y3im)
vare = v4re: vaim = v4im : 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) * 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
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
CALL ekran: LOCATE 2, 2: COLOR 14, 1
PRINT "Maximum values of Transfer voltage/admitans functions are written..
COLOR 15, 1: CALL anykey: CALL ekran: LOCATE 2, 10: COLOR 14, 1
PRINT "Gain Block...": COLOR 15, 1: LOCATE 5, 2
PRINT "Yamax="; iamax * 1000000!; "[uA/V]"

```

```

imax = isat / imax
IF imax < vimax THEN
vimax = imax: 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, 15: COLOR 14, 1: PRINT "Tb Blo°u..."
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, 15: 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="; 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 "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, 15: 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="; 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 "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
CALL ekran: LOCATE 2, 15: COLOR 14, 1
PRINT "Maximum input voltage for the overall filter "
COLOR 15, 1: LOCATE 6, 6 : PRINT "Vimax="; vimax * 1000; "mV"
LOCATE 7, 6 : IF domb = 0 THEN
LOCATE 8, 6: PRINT "Dominant Block is ..."; "Tb"
LOCATE 9, 6: PRINT "Dominant H/Y is ..."; " "; dom$
ELSE
LOCATE 8, 6: PRINT "Dominant Block is ..."; domb
LOCATE 9, 6: PRINT "Dominant H/Y is ..."; " "; dom$
END IF
CALL anykey
goster1:
CALL ekran: COLOR 14, 1: LOCATE 2, 2
PRINT " Choose the quad/cascade block for which you want to see H/Y"
COLOR 15, 1: LOCATE 4, 2
IF tb$ = "var" THEN
LOCATE 5, 2: PRINT "[ 20]..girif için"
LOCATE 6, 2: PRINT "[ 0 ]..Tb için"
FOR i = 1 TO nx
LOCATE i + 6, 2:PRINT "[ "; i; "].."; i; ".Blok için"
NEXT
ELSE
LOCATE 5, 2: PRINT "[ 20]..Girif için"
FOR i = 1 TO nx
LOCATE i + 5, 2: PRINT "[ "; i; "].."; i; ".Blok için"
NEXT
END IF
LOCATE 5, 30: INPUT "[ seçiminiz... ] >", qx
CALL ekran: COLOR 14, 1: LOCATE 2, 20
PRINT "choose the H/Y you want to see"
COLOR 15, 1: LOCATE 4, 2
IF qx = 20 THEN
LOCATE 5, 2: PRINT "[ 1 ].Ya .için" : LOCATE 6, 2: PRINT "[ 2 ].Yb .için"
ELSEIF qx = 0 THEN
LOCATE 5, 2: PRINT "[ 1 ].Hb1 .için" : LOCATE 6, 2: PRINT "[ 2 ].Hb2 .için"
LOCATE 7, 2: PRINT "[ 3 ].Hb3 .için" : LOCATE 5, 30: PRINT "[ 4 ].Yb1 .için"
LOCATE 6, 30: PRINT "[ 5 ].Yb2 .için" : LOCATE 7, 30: PRINT "[ 5 ].Yb3 .için"
ELSEIF qx <> 0 AND qx <> 20 THEN

```

```

LOCATE 5, 2: PRINT " [ 1 ]..Hj...için" : LOCATE 6, 2: PRINT " [ 2 ]..H1...için"
LOCATE 7, 2: PRINT " [ 3 ]..H2...için" : LOCATE 8, 2: PRINT " [ 4 ]..H3...için"
LOCATE 9, 2: PRINT " [ 5 ]..H4...için" : LOCATE 10, 2: PRINT " [ 6 ]..H5...için"
LOCATE 11, 2: PRINT " [ 7 ]..H6...için" : LOCATE 12, 2: PRINT " [ 8 ]..H7...için"
LOCATE 5, 30: PRINT " [ 9 ]..Y1...için" : LOCATE 6, 30: PRINT " [ 10 ]..Y2...için"
LOCATE 7, 30: PRINT " [ 11 ]..Y3...için" : LOCATE 8, 30: PRINT " [ 12 ]..Y4...için"
LOCATE 9, 30: PRINT " [ 13 ]..Y5...için" : LOCATE 10, 30: PRINT " [ 14 ]..Y6...için"
LOCATE 11, 30: PRINT " [ 15 ]..Y7...için" : LOCATE 12, 30: PRINT " [ 16 ]..Y8...için"
LOCATE 13, 30: PRINT " [ 17 ]..Y9...için" : LOCATE 14, 30: PRINT " [ 18 ]..Y10...için"
LOCATE 15, 30: PRINT " [ 19 ]..Yx...için" : LOCATE 16, 30: PRINT " [ 20 ]..Yy...için"
IF kq$ = "quad" THEN
LOCATE 17, 30: PRINT " [ 21 ]..Yz...için"
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
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
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) = 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
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) = 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)
ELSE
END IF
NEXT
CLOSE #20: CLOSE #21
FOR i = 1 TO nx
CLOSE #i
NEXT
END SUB

```

'-----subroutine GRAFIK-----'

SUB grafik

"draws obtained filter response and all nodal transfer functions

"and transfer admitans functions

CLS : SCREEN 9: COLOR 14, 1: CALL frame

maxx = 640: maxy = 350

LINE (45, 330)-(50, 340), 14: LINE (50, 340)-(55, 330), 14

LINE (620, 75)-(630, 70), 14: LINE (630, 70)-(620, 65), 14

VIEW (50, 10)-(630, 280): WINDOW (0, 0)-(580, 270)

LINE (580, 0)-(0, 0), 14: LINE (0, 0)-(0, 270), 14

LOCATE 23, 76: PRINT "wait"

genlikayar = 230 / maxgenlik

LINE (1, 0)-(1, genlikayar * genlik(1)), 1

FOR k = 2 TO s - 8

'LINE -(k, genlikayar * ABS(genlik(k)) + 1), 15

LINE -(k, 4.86 * (20 * LOG(genlikayar * ABS(genlik(k)) + 1) / LOG(10)) + 1), 15

NEXT

LOCATE 4, 2: PRINT maxgenlik: LOCATE 20, 2: PRINT genlik(1)

LOCATE 11, 2: PRINT " G": LOCATE 12, 2: PRINT " a"

LOCATE 13, 2: PRINT " i": LOCATE 14, 2: PRINT " n"

LOCATE 15, 2: PRINT " ,": LOCATE 16, 2: PRINT " dB"

LOCATE 23, 40: PRINT "Frekans,hz."

END SUB

-----function GERIBES-----

FUNCTION geribes (kk, ll, mm)

"This subroutine calculates the feedback factor of the ith Quad block

'kk=a,ll=b,mm=c

IF a\$ = "bw" OR a\$ = "chy" THEN

geribes = ll - (kk ^ 2) / 4

ELSE

geribes = (ll + mm - SQR((ll - mm) ^ 2 + mm * kk ^ 2)) / (2 * mm)

END IF

END FUNCTION

-----function MODUL-----

FUNCTION modul (a, b)

"finds the magnitude of a pole/zero

modul = SQR(a ^ 2 + b ^ 2)

END FUNCTION

-----function ADDRE-----

FUNCTION addre (a1, a2, b1, b2)

"real part of an addition of two complex number

addre = a1 + b1

END FUNCTION

-----function ADDIM-----

FUNCTION addim (a1, a2, b1, b2)

'imaginary part of an addition of two complex numbers

addim = a2 + b2

END FUNCTION

-----function MULRE-----

FUNCTION mulre (a1, a2, b1, b2)

"real part of a multiplication of two complex numbers

mulre = a1 * b1 - a2 * b2

END FUNCTION

-----function MULIM-----

FUNCTION mulim (a1, a2, b1, b2)

"imaginary part of multiplication of two complex numbers

mulim = a1 * b2 + a2 * b1

END FUNCTION

-----function DIVRE-----

```

FUNCTION divre (a1, a2, b1, b2)
"real part of division of two complex numbers
divre = mulre(a1, a2, b1, -b2) / mulre(b1, b2, b1, -b2)
END FUNCTION

```

```

-----function DIVIM-----
FUNCTION divim (a1, a2, b1, b2)
"imaginary part of division of two complex numbers.
divim = mulim(a1, a2, b1, -b2) / mulre(b1, b2, b1, -b2)
END FUNCTION

```

```

-----subroutine EKRAN-----
SUB ekran
CLS
VIEW (0, 0)-(639, 349)
WINDOW (0, 0)-(639, 349)
LINE (1, 1)-(639, 349), 15, B
LINE (3, 347)-(637, 317), 14, B
COLOR 15, 1
END SUB

```

```

-----subroutine FRAME-----
SUB frame
CLS
LINE (1, 1)-(639, 349), 15, B
END SUB

```

```

-----subroutine ANYKEY-----
SUB anykey
yaz:
LOCATE 23, 74
PRINT "anykey"
IF INKEY$ = "" THEN
GOTO yaz
ELSE
END IF
CLS

```

```

-----subroutine DELAY-----
SUB delay
FOR i = 1 TO 4500
i = i + 1
NEXT
END SUB

```

```

-----END OF THE PROGRAM-----

```

ÖZGEÇMİŞ

Ramazan KÖPRÜ 1968 yılında Çankırı' da doğdu. 1987' de Haydarpaşa Teknik Lisesi Elektronik Bölümü' nden mezun oldu. 1991 yılında, Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü' nü bitirdi. Aynı yıl, İstanbul Teknik Üniversitesi' nin açmış olduğu sınavlarda başarı göstererek Elektronik ve Haberleşme Mühendisliği Bölümü' nde yüksek lisans öğrenimine başladı. Yine aynı yıl Ereğli Demir ve Çelik Fabrikaları T..A.Ş.' nin yurtiçi yüksek lisans bursunu almaya hak kazandı. Halen İstanbul Teknik Üniversitesi' nde yüksek lisans öğrenimini sürdürmektedir.

