YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

IBM PC İLE GENEL AMAÇLI SAYISAL FİLTRE GERÇEKLEMESİ

Elektronik ve Hab. Müh. Beşir TAYFUR

F.B.E. Elektronik ve Haberleşme Anabilim Dalında
hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd.Doc.Dr. Tuncay UZUN

İSTANBUL, 1997

İÇİNDEKİLER

TEŞEKKÜR ..... **III**

ÖZET ..... **IV**

ABSTRACT ..... **V**

1. GİRİŞ **1**

2. TEMEL KAVRAMLAR **3**

2.1 SONSUZ İMPULS YANITLI FİLTRE TASARIM YÖNTEMLERİ 3
2.1.1 Nümerik Yaklaşımlar 4
2.1.1.1 Geriye Doğru Fark (Backward Difference) 4
2.1.1.2 İleriye Doğru Fark (Forward Difference) 7
2.1.2 Değişmez İmpuls Yanıtı Yöntemi 10
2.1.3 Uygunlaştırılmış z-Dönüşüm Yöntemi 14
2.1.4 Bilineer Dönüşüm 15
2.2 SAYISAL FİLTRE YAPILARI 21
2.2.1 Direkt Yapılar 22
2.2.1.1 Birinci Direkt Yapı (1D) 22
2.2.1.2 İkinci Direkt Yapı (2D) 24
2.2.1.3 Üçüncü Direkt Yapı (3D) 25
2.2.1.4 Dördüncü Direkt Yapı (4D) 26
2.2.2 İkinci derece modüller 28
2.2.2.1 İkinci Derece Modüllerle Kaskad Gerçekleme 34
2.2.2.2 İkinci Derece Modüllerle Paralel Gerçekleme 38

3. IBM PC UYUMLU SİSTEM İLE SAYISAL FİLTRE TASARIMI VE GERÇEKLEŞTİRİLMESİ **44**

3.1 SAYISAL FİLTRE DONANIMI 45
3.1.1 12-bit 16 kanallı Analogdan Sayısala Dönüştürücü (ADC) 45
3.1.2 12-bit 2 kanallı Sayısaldan Analoga Dönüştürücü (DAC) 46
3.2 SAYISAL FİLTRE YAZILIMI 47
3.2.1 Sayısal Filtre Yazılımının Bölümleri 53
3.2.1.1 Örnekleme Frekansının Belirlenmesi 53
3.2.1.2 Sayısal Filtre Katsayılarının Bulunması 56
3.2.1.3 Sayısal Filtre Programının Çalıştırılması 63
3.2.1.3.1 Direkt Gerçekleme 63
3.2.1.3.2 İkinci-Derece Modüllerle Paralel Gerçekleme 68

4. SONUÇ **74**

5. KAYNAKLAR **87**

6. EKLER **88**

EK1. 80387 MATEMATİK YARDIMCI İŞLEMCİSİ VE ÖZELLİKLERİ 88
EK2. SAYISAL FİLTRELERİN DİREKT GERÇEKLENMESİ PROGRAMI 90
EK3. SAYISAL FİLTRELERİN İKİNCİ DERECE MODÜLLERLE PARALEL GERÇEKLENMESİ PROGRAMI 104

TEŞEKKÜR

Bu çalışmada, tez konusunun belirlenmesinden sonuçlandırılmasına kadar geçen sürede bana verdiği her türlü destek ve gösterdiği hoşgörü nedeniyle Sn. Yrd.Doc.Dr.Tuncay UZUN' a, karşılaştığım problemlerin çözümünde değerli fikirleriyle bana yardımcı olan başta Yavuz KILIÇ, Galib DEMİRTAŞ ve Osman N. ERTÜRK olmak üzere tüm arkadaşlarıma ve özellikle her zaman, her yerde, şartlar ne olursa olsun, bana duydukları güven ve gösterdikleri hiç eksilmeyen sevgiden dolayı aileme sonsuz teşekkürlerimi sunarım.

Beşir TAYFUR

Eylül, 1997

İstanbul

ÖZET

Son yıllarda işaret işleme alanında büyük gelişmeler olmuştur. Bu gelişmelerde sayısal filtreler önemli bir rol oynamıştır. Bugün pek çok işaret işleme uygulamasında sayısal filtrelere gereksinim duyulmaktadır. Tıbbi ve endüstriyel uygulamalar başta olmak üzere bu uygulamaların çoğunda ele alınan işaretler düşük frekanslıdır. Bu nedenle bilgisayar sistemine eklenecek genel amaçlı bir giriş/çıkış kartı kullanılarak gerçeklenecek sayısal filtreler bu tür uygulamalarda rahatlıkla kullanılabilir.

Bu çalışmada, sayısal filtre yazılım ve donanımı IBM PC uyumlu bir bilgisayar sisteminde tasarlanmış ve gerçekleştirilmiştir. Bu amaçla ilk olarak uygun sonsuz impuls yanıtlı sayısal filtre tasarım yöntemi ve sayısal filtre yapısı belirlenmiştir. Ardından filtre katsayılarının ve çarpımların kuantalanması nedeniyle meydana gelen hatalar kayan noktalı aritmetik işlem yapılarak ortadan kaldırılmıştır. Daha sonra sayısal filtrelerin direkt ve ikinci derece modüllerle paralel gerçeklenmesi için gerekli yazılımlar tasarlanmıştır. Bu amaçla örnekleme frekansını, sayısal filtre transfer fonksiyonunun katsayılarını bulan ve sayısal filtreyi çalıştıran olmak üzere üç ana program yazılmıştır.

Endüstride sayısal filtreye gerek duyulan ortamlarda kullanılan malzemeye bağlı olarak analog giriş/çıkış kartı ile sayısal işaret işlemci kartları önemli özellikleri ile karşılaştırılabilir. Her iki donanımda da performans arttıkça sayısal filtrenin maliyeti artmaktadır. Fakat bu çalışmada kullanılan çok fonksiyonlu giriş/çıkış kartı daha geniş kullanım alanına sahip olduğu ve daha ucuz olduğu için filtrenin kullanıldığı ortamda sistem maliyetinin daha ekonomik olacağı açıktır. Yapılan ölçümler sonucunda günümüzde yaygın olarak kullanılan IBM PC uyumlu bilgisayar sistemi ve bu çalışmada kullanılan MPIBM4 çok fonksiyonlu giriş/çıkış kartı ile 8.dereceden filtrelerin gerçeklenebildiği ve bu filtrelerle 5kHz'e kadar frekansa sahip işaretlerin sağlıklı bir şekilde filtrelenebildiği görülmüştür. Elde edilen bu frekans değeri, pek çok ölçme ve kontrol uygulamasında işaretlerin çoğunlukla 1kHz civarında olduğu düşünülürse iyi bir değerdir.

ABSTRACT

In recent years, there have been great improvements in digital signal processing field. Digital filters have played an important role in these improvements. Today, there is a great demand for digital filters in most Digital Signal Processing applications. The signals used in these applications (especially medical and industrial ones) are based on low frequency. Therefore, digital filters designed with IBM compatible PC and additional general purpose input/output interface can be easily used in these applications.

In this study, digital filter software and hardware were designed and realized by using IBM compatible PC. Firstly, suitable IIR (Infinite Impulse Response) filter design methodology and architecture determined. Next, the errors generated by quantization of filter coefficients and multiplications removed by using floating point arithmetic. Then, the software required for direct and parallel realization with second order modules of digital filters was formed. Three main programs below were designed.

- A program calculating sample rate,
- A program calculating filter coefficients,
- A program emulating the filter.

In some industrial applications needing digital filters, depending on used materials, analog input/output cards can be compared with digital signal processing cards concerning their important features. The cost increase depends on performance of designed digital filter in both cases. As it has a wide application range and a cost lower than a DSP card, it's obvious that in the applications based on a digital filter, the multifunctional analogue input/output card used in this study is more economical. The results of this study have showed that eighth order digital filters can be realized by using multifunctional input/output cards and IBM compatible PC's commonly used today. In addition, signals up to 5kHz can be filtered well in this way. Since most measurement and control applications have signals approximately about 1kHz, this obtained value is quite enough.

1. GİRİŞ

Günümüzde kullanılan elektronik sistemlerde değişik nedenlerle analog sistemlerden sayısal sistemlere geçiş hızlı bir şekilde sürmektedir . Bu gelişme elektronik teknolojisinde varolan bir analog elektronik sistemin sayısal düzenlemesiyle değiştirilmesi veya güncellenmesi biçiminde olabilir. Sayısal sistemler güvenirlilik problemini azaltması, elektronik gürültüye ve elektromagnetik yayılmaya karşı daha az duyarlı olmaları tercih edilmelerindeki başlıca nedenlerdir.

Geçtiğimiz on yıl içerisinde mikroelektronik alanında çok önemli gelişmeler olmuştur. Genel amaçlı sayısal bilgisayarların tasarımında kullanılan tümdevreler tasarlanarak üretilmeye başlanmıştır. Elektronik sistemlerde yaygın olarak kullanılan sayısal filtreler bu gelişmelerden doğrudan etkilenmiştir. Bundan sonra sayısal filtrelerin performansını artırmak için bu elektronik sistemler daha da geliştirilmiş ve yalnız işaret işlemeye yönelik tümdevreler üretilmeye başlanmıştır.

Sayısal filtreler son yirmi yıl içerisinde işaret işleme alanında büyük gelişmelere neden olmuştur. Günümüzde sayısal filtreler pek çok alanda kullanılmaktadır. Bunda sayısal filtrelerin transfer karakteristiğinin zaman içerisinde değişim göstermemesi, işaret kaymalarının minimal olması, emniyet izlemelerinin (safety monitoring) kolay olması, sağlamış olduğu güvenilirlilik ve esneklik önemli bir rol oynamaktadır.

Bugün 5kHz' lik bir örnekleme hızına sahip sekizinci dereceden bir sayısal filtre pek çok uygulamada kullanılabilir. Bu uygulamalara örnek olarak düşük seviyeli ses işleme (low-grade voice processing), kapalı çevrim kontrol sistemleri (closed-loop control systems), tıbbi işaret işleme (medical signal processing) ve endüstriyel amaçlı işaret işleme gösterilebilir.

Bir kapalı çevrim kontrol uygulamasında mikrobilgisayar yalnız kontrol işlemini gerçeklemez. Buna ek olarak kontrol edilen sistemdeki işaret seviyesini izleme ve hata durumunu gözleme işlemlerini yapar. Belirtilen bu işlemlerin yerine getirilmesinde sayısal filtrelerden yararlanılır. Örnek olarak bir örneklenmiş veri (sampled-data) kontrol uygulamasında kontrol edilen sisteme ait en yüksek frekanslı işaretten 2 ile 10 arasında değişen sayıda örnek alınmaktadır. Bugün pek çok kontrol uygulamasında kontrol sinyalleri 1kHz civarındadır. Buna göre 5kHz örnekleme hızına sahip bir sayısal filtre ile böyle bir işaretten 5 tane örnek almak mümkün olmaktadır.

Bir açık çevrim uygulamasında (open-loop application), bir gerçek zaman sayısal filtre belirli bir işaretin dalga formunu ve biçimini aramak için kullanılabilir. Örnek olarak kalp izleme cihazlarında tüm önemli bilgiler elektrokardiogramın özel bir dalga formunun (QRS complex) zamanla değişiminden elde edilmektedir.

2. TEMEL KAVRAMLAR

2.1 SONSUZ İMPULS YANITLI FİLTRE TASARIM YÖNTEMLERİ [1, 2, 3]

FIR (finite impulse response-sonlu impuls yanıtlı) filtreler süperbilineer faz davranışı gösterir. Buna karşın genlik frekans yanıtı iyi tutulmak isteniyorsa yüksek dereceli bir FIR filtreye gereksinim olacaktır. IIR (infinite impulse response-sonsuz impuls yanıtlı) filtreler FIR filtrelerle karşılaştırıldığında, IIR filtrelerin FIR filtrelere göre daha düşük derecelerde istenen genlik frekans yanıtını sağladığı görülmektedir. Bu da gerçekleme yapılırken yapılacak işlem (çarpma gibi) sayısını azaltmaktadır. Bu beraberinde yüksek örnekleme hızlarına erişim imkanını getirmektedir.

Bir IIR filtre fark denklemleri veya transfer fonksiyonu yardımıyla belirlenebilir. Bir IIR filtrenin transfer fonksiyonu genel olarak;

$$H(z)=\frac{N(z)}{D(z)}=\frac{\sum_{i=0}^{M}a_i z^{-i}}{1+\sum_{i=1}^{N}b_i z^{-i}}=\frac{a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_Mz^{-M}}{1+b_1z^{-1}+b_2z^{-2}+\cdots+b_Nz^{-N}} \qquad (2.1.1)$$

biçiminde yazılabilir. Burada $M \leq N$ olup, tasarım problemindeki amaç a_i ve b_i katsayılarının bulunmasıdır. Doğal olarak bu katsayılar hesaplanırken tasarımda istenen özellikler gözönünde bulundurulmalıdır.

Sayısal filtrelerin özelliklerini genlik karakteristiği, faz karakteristiği ve geçici performans oluşturmaktadır. Ancak gerçek bir tasarımda kararlılık, nedensellik ve basitlik gibi koşullardan dolayı sadece genlik karakteristiği ele alınır. O halde tasarım problemi

H(z) transfer fonksiyonunun istenen genlik yanıtını sağlayacak şekilde a_i ve b_i katsayılarının bulunmasına indirgenebilir.

IIR sayısal filtrelerin tasarlanmasında kullanılan başlıca yöntemler şunlardır;

- İstenen özellikleri sağlayan analog filtreden uygun dönüşümlerle sayısal filtrenin elde edilmesi,
- Sayısal filtrenin z-düzleminde kapalı formda doğrudan elde edilmesi,
- Bir takım optimizasyon yöntemleri kullanarak istenen frekans yanıtını sağlayacak şekilde sıfır ve kutupların yerleştirilmesi.

Bu yöntemler içerisinde istenen özellikleri sağlayacak analog filtrenin sayısallaştırılması ençok kullanılan yöntemdir. Sayısal filtrelerin analog filtrelerden elde edilmesinde kullanılan dönüşüm yöntemleri:

- Nümerik Yaklaşımlar,
- Değişmez İmpuls Yanıtı Yöntemi,
- Uygunlaştırılmış z-Dönüşüm Yöntemi,
- Bilineer Dönüşüm.

2.1.1 Nümerik Yaklaşımlar

Dönüşümlerde (mapping) kullanılan birkaç nümerik yaklaşım tekniği vardır.

2.1.1.1 Geriye Doğru Fark (Backward Difference)

Bu nümerik yöntemde türev yerine aşağıdaki ifade konulmaktadır.

$$\frac{d}{dt}y(t) = \frac{y(t) - y(t-T)}{T} \qquad (2.1.2)$$

Bu durum şekil 2.1.1' de görülmektedir. Bu ifadenin Laplace dönüşümü alınırsa;

$$sY(s) - y(0^+) = \frac{Y(s) - e^{-sT}Y(s)}{T}$$

$$sY(s) = \frac{Y(s) - e^{-sT}Y(s)}{T} + y(0^+) \tag{2.1.3}$$

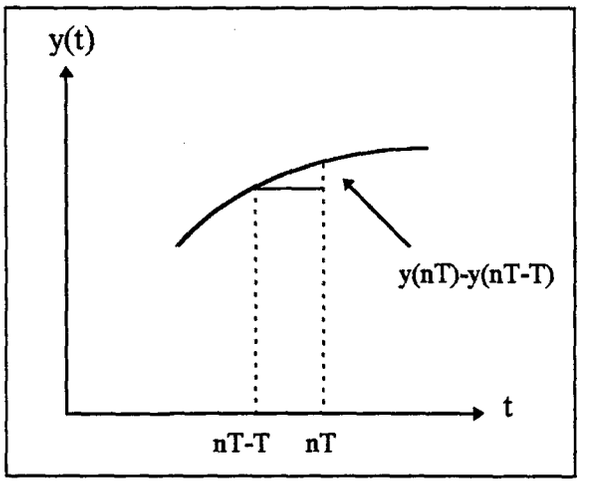
elde edilir. Eğer $y(0^+)$ değeri küçükse, her taraf Y(s)'e bölünerek;

$$s = \frac{1 - e^{-sT}}{T}$$

$$s = \frac{1 - z^{-1}}{T} \tag{2.1.4}$$

sonucu bulunur. Buna göre s-domenindeki transfer fonksiyonundan z-domenindeki transfer fonksiyonuna, s-yerine (2.1.4)' de gösterilen değeri konarak geçilebilir.

$$D(z) = G(s)\Big]_{s=\frac{1-z^{-1}}{T}} \tag{2.1.5}$$



Şekil 2.1.1 Geriye Doğru Fark Yaklaşımı

(2.1.4)' den z değeri çekilerek;

$$z = \frac{1}{1 - sT} \tag{2.1.6}$$

yazılabilir. (2.1.6)' da s=jω yazılarak;

$$z = \frac{1}{1 - j\omega T} = \frac{1}{2}\left(1 + \frac{1 + j\omega T}{1 - j\omega T}\right)$$

$$z = \frac{1}{2}\left(1 + e^{2j\tan^{-1}(\omega T)}\right) \tag{2.1.7}$$

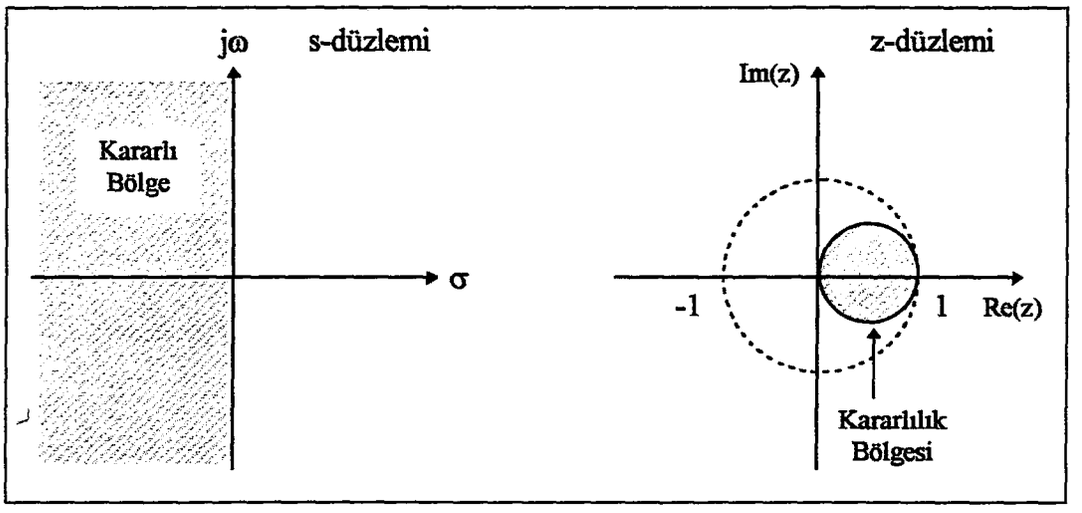
bulunur. (2.1.7)' de z' nin reel ve imajiner kısımları alınırsa;

$$\begin{aligned} \mathrm{Re}[z] &= \frac{1}{2} + \frac{\cos(2\tan^{-1}(\omega T))}{2} \\ \mathrm{Im}[z] &= \frac{\sin(2\tan^{-1}(\omega T))}{2} \end{aligned} \tag{2.1.8}$$

elde edilir. s=jω ekseninin (-∞,∞) aralığında değişmesi bu dönüşümle z-düzleminde merkezi (1/2,0) ve yarıçapı 1/2 olan bir çember oluşturur.

$$\left\{\mathrm{Re}[z] - \frac{1}{2}\right\}^2 + \mathrm{Im}[z]^2 = \left(\frac{1}{2}\right)^2 \tag{2.1.9}$$

Şekil 2.1.2' den de görüldüğü gibi sol yarı s-düzlemi birim dairenin içine karşı düşmektedir. Buna göre bu dönüşümle kararlı bir analog filtreden kararlı bir sayısal filtreye her zaman geçilebilir.



Şekil 2.1.2 Geriye doğru fark yöntemiyle dönüşüm

Yine aynı şekilden ωT' nin oldukça küçük değerleri hariç, s düzlemindeki jω ekseninin görüntüsünün z-düzleminde birim daireye karşı düşmediği görülebilir. Bu dönüşümün en büyük dezavantajıdır.

2.1.1.2 İleriye Doğru Fark (Forward Difference)

Bu nümerik yöntemde türev yerine aşağıdaki ifade konulmaktadır.

$$\frac{d}{dt}y(t) = \frac{y(t+T) - y(t)}{T} \tag{2.1.10}$$

Bu ifadenin Laplace dönüşümü alınırsa;

$$sY(s) - y(0^+) = \frac{e^{sT}Y(s) - Y(s)}{T}$$

$$sY(s) = \frac{e^{sT}Y(s) - Y(s)}{T} + y(0^+) \tag{2.1.11}$$

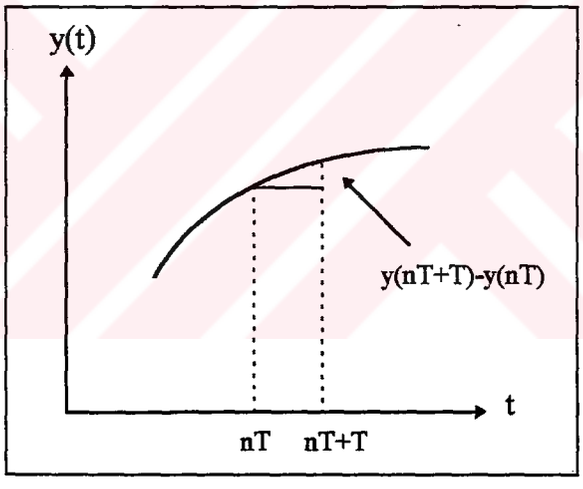
elde edilir. Eğer $y(0^+)$ değeri küçükse, her taraf Y(s)'e bölünerek;

$$s = \frac{e^{sT} - 1}{T}$$

$$s = \frac{z - 1}{T} \tag{2.1.12}$$

sonucu bulunur. Buna göre s-domenindeki transfer fonksiyonundan z-domenindeki transfer fonksiyonuna, s-yerine (2.1.12)' de gösterilen değeri konarak geçilebilir.

$$D(z) = G(s)\Big]_{s=\frac{z-1}{T}} \tag{2.1.13}$$

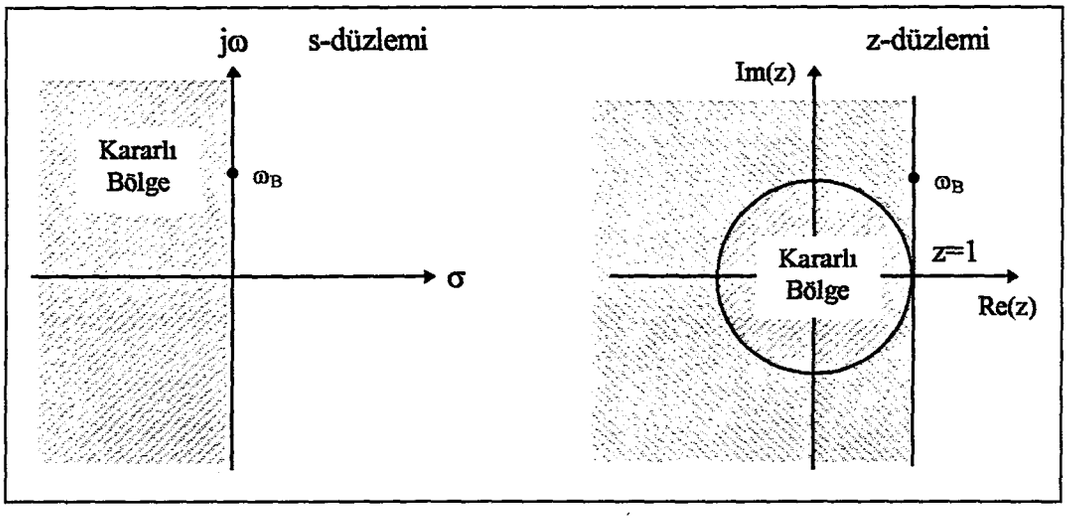


Şekil 2.1.3 İleriye Doğru Fark Yaklaşımı

(2.1.12)' den z değeri çekilerek;

$$z = 1 + sT \tag{2.1.14}$$

olarak bulunur. Şekil 2.1.4' de bu yöntem kullanılarak s-düzleminden z-düzlemine dönüşüm görülmektedir. Şekilden de görüldüğü gibi s domenindeki sol yarı düzlem, z-

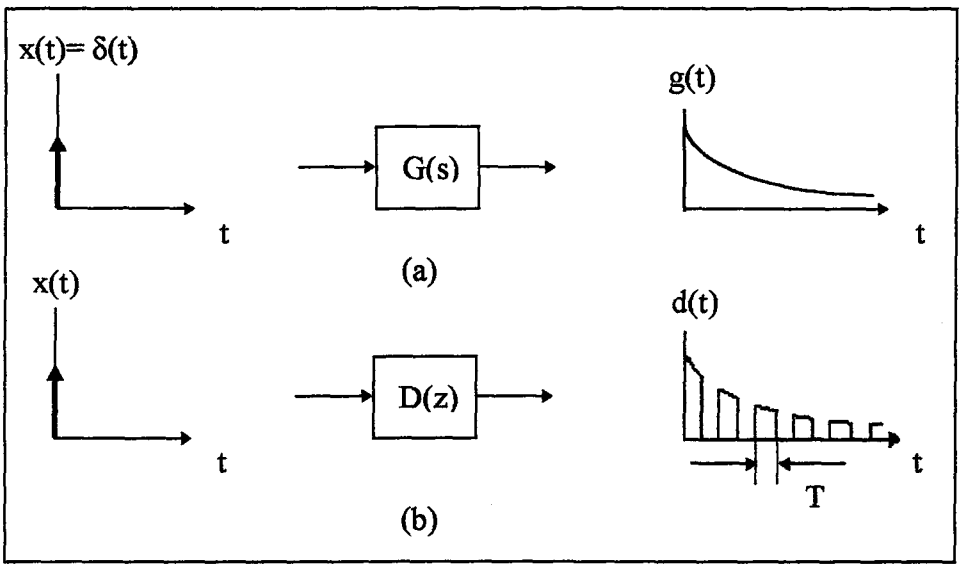


Şekil 2.1.4 İleriye doğru fark yöntemiyle dönüşüm

düzleminde z=1 doğrusunun sol tarafında kalan bölgeye karşı düşmektedir. Ancak z-düzleminde kararlı olan bölge sadece birim dairenin içidir. Bundan dolayı bazı kararlı analog filtrelerden bu dönüşümle kararsız sayısal filtreler elde edilecektir. Diğer bir dezavantaj ise s-düzlemindeki jω ekseninin z-düzleminde birim dairenin değil, z=1 doğrusunun üzerine karşı düşmesidir. Bu nedenlerden dolayı bu dönüşüm pek tercih edilmez.

Bu bahsedilen nümerik yaklaşımlar dışında dikdörtgen kuralı (rectangular rule), sol-yan kuralı (left-side rule), sağ-yan kuralı (right-side rule), yamuk kuralı (trapezoidal rule), simpson kuralı gibi başka yaklaşımlarda vardır.

2.1.2 Değişmez İmpuls Yanıtı Yöntemi



Şekil 2.1.5 Değişmez-impuls: (a) Analog filtre; (b) Sayısal filtre

Analog bir filtrenin transfer fonksiyonu $H_a(s)$ ile gösterilirse, impuls yanıtı ilk koşullar sıfır alınarak ters Laplace dönüşümü ile bulunabilir.

$$h_a(t) = L^{-1}\left[H_a(s)\right] \tag{2.1.15}$$

Sürekli-zaman impuls yanıtı $h_a(t)$' nin t=nT anlarında örneklenmesi ile elde edilen $h_a(nT)$, tasarlanmak istenen sayısal filtrenin impuls yanıtı olacaktır. O halde sayısal filtrenin transfer fonksiyonu;

$$H_D(z) = Z\left[h_a(nT)\right] \tag{2.1.16}$$

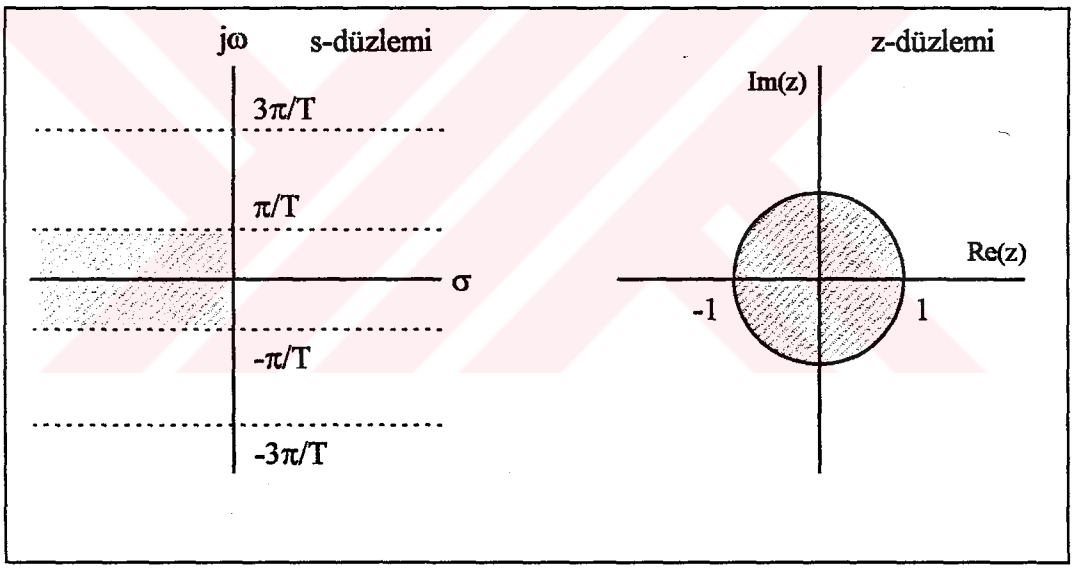
olarak bulunur. Eğer $H_a(\omega)$ sınırlı bandlı ise, elde edilen sayısal filtrenin temel band frekans spektrumu da yaklaşık olarak analog filtrenin frekans yanıtının (1/T) sabit katsayısıyla çarpımı kadar olacaktır. Örnekleme teoreminden;

$$H_D(e^{j\omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a(\omega - k\omega_s) \qquad (2.1.17)$$

Burada $\omega_s=2\pi/T$ örnekleme frekansıdır. Benzer ilişkiyi h(nT)' nin z-dönüşümü ile h(t)' nin Laplace dönüşümü arasında da görmek mümkündür.

$$H_D(z)\big|_{z=e^{sT}} = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a(s + j(2\pi / T)k) \qquad (2.1.18)$$

Şekil 2.1.6' da gösterildiği gibi $z=e^{sT}$ standart dönüşümüyle s-düzlemindeki $(2\pi/T)$ genişliğindeki taralı bölge z-düzleminde birim daire içine karşı düşmektedir.



Şekil 2.1.6 Periyodik örnekleme

$$H_a(\omega) \approx 0; \quad |\omega| \geq \frac{\omega_s}{2} \qquad (2.1.19)$$

sınırlı band koşulu sağlanırsa,

$$\sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} H_a(\omega - k\omega_s) \approx 0; \quad |\omega| \leq \frac{\omega_s}{2} \tag{2.1.20}$$

yazılabilir. Yani, (2.1.17)' de gösterilen sayısal filtrenin temel band frekans spektrumuna, ($-\omega_s/2$, $\omega_s/2$) yan bandların toplam etkisi ihmal edilecek kadar azdır. O halde,

$$H_D(e^{j\omega T}) \cong \frac{1}{T} H_a(\omega), \qquad |\omega| \leq \frac{\omega_s}{2} \tag{2.1.21}$$

olur. Buna göre, (2.1.19)' daki koşulunun sağlanması durumunda, analog impuls yanıtının örnekleri ile elde edilen sayısal filtrenin frekans yanıtı yaklaşık olarak aynı olmaktadır.

Analog filtrenin transfer fonksiyonu $H_a(s)$ sadece basit tek katlı köklere sahipse,

$$H_a(s) = \sum_{k=1}^{N} A_k / (s - s_k) \tag{2.1.22}$$

biçiminde yazılabilir. Böylece, yöntemin gerektirdiği impuls yanıtının bulunması ve örnekleme işlemleri kolaylıkla gerçekleştirilebilir. (2.1.15) denkleminden;

$$h_a(t) = L^{-1}\left[H_a(s)\right] = \sum_{k=1}^{N} a_k e^{s_k t} \tag{2.1.23}$$

ve

$$h_a(nT) = \sum_{k=1}^{N} A_k e^{s_k nT} \tag{2.1.24}$$

yazılabilir. Sonuç olarak, tasarlanan filtrenin transfer fonksiyonu,

$$H_D(z) = Z\left[h_a(nT)\right] = \sum_{k=1}^{N} \frac{A_k z}{z - e^{s_k T}} \qquad (2.1.25)$$

olarak bulunur. $H_a(s)$' nin karmaşık eşlenik kutupları nedeni ile, A_k ve $e^{s_k T}$' nin karmaşık eşlenikleri olacaktır. Sonuç olarak, 2.1.25' deki $H_D(z)$' nin katsayıları reeldir.

Kararlı analog filtrenin $s_k=\sigma_k+j\omega_k$ kutbu $H_D(z)$' nin

$$z_k = e^{s_k T} = e^{(\sigma_k + j\omega_k)T} \qquad (2.1.26)$$

kutbuna dönüşmektedir. Buradan, $\sigma_k<0$ için $|z|<1$ yani, sol yarı düzlemdeki analog kutuplar z-düzleminde birim daire içine düşmektedir. O halde, kararlı analog filtre, bu anlatılan yöntemle kararlı sayısal filtreye dönüşmektedir.

Bu dönüşüm metodunda s-düzlemi s_k kutupları z-düzlemi içindeki $z_k=e^{s_k T}$ kutuplarına karşı düşmektedir. Eğer s_k sol-yarı düzlemde ise $z_k=e^{s_k T}$ birim dairenin içerisindedir. Bununla beraber bu dönüşüm bire-bir değildir. Yani s-düzleminde birçok nokta z-düzleminde aynı noktaya karşı gelebilir.

(2.1.19)' daki koşulun pratikte sağlanamaması sayısal filtrenin frekans spektrumunda örtüşmelere neden olmaktadır. Örnekleme frekansının artırılması veya örnekleme peryodunun küçültülmesi örtüşme etkilerini azaltır. Alçak geçiren filtre için doğru sayılabilecek bu yöntem yüksek geçiren ve band söndüren filtrelerde büyük hatalara neden olur, bu nedenle bu tür filtrelerde kullanılamaz.

2.1.3 Uygunlaştırılmış z-Dönüşüm Yöntemi

Uygunlaştırılmış z-dönüşüm yöntemi (Matched z-transformation) bir analog filtreyi sayısallaştırmada kullanılan diğer yöntemdir. Bu yöntemde s-düzlemindeki kutup ve sıfırlardan z-düzlemindeki kutup ve sıfırlara doğrudan dönüşüm yapılır.

$$G(s) = \frac{\alpha_0 + \alpha_1 s^{-1} + \alpha_2 s^{-2} + \cdots + \alpha_n s^{-n}}{1 + \beta_1 s^{-1} + \beta_2 s^{-2} + \cdots + \beta_n s^{-n}} \tag{2.1.27}$$

Yukarıdaki G(s) fonksiyonunun standart z-dönüşümünün alınabilmesi için bu fonksiyonun kısmı kesirlere açılması ve

$$\frac{1}{s+u} \rightarrow \frac{1}{1 - z^{-1} e^{-uT}} \tag{2.1.28}$$

dönüşümünün yapılması gerekir. Hesaplamayı basitleştirmek için, uygunlaştırılmış z-dönüşümüyle G(s) fonksiyonunun kutup ve sıfırları z-düzlemine aşağıdaki eşleştirmeyle dönüştürülür:

$$s + a \rightarrow 1 - z^{-1} e^{-aT} \tag{2.1.29}$$

Kompleks kutuplar (sıfırlar) için de benzer şekilde;

$$\begin{aligned} &(s + a - jb)(s + a + jb) \\ &= (s+a)^2 + b^2 \rightarrow 1 - 2z^{-1} e^{-aT} \cos(bT) + e^{-2aT} \end{aligned} \tag{2.1.30}$$

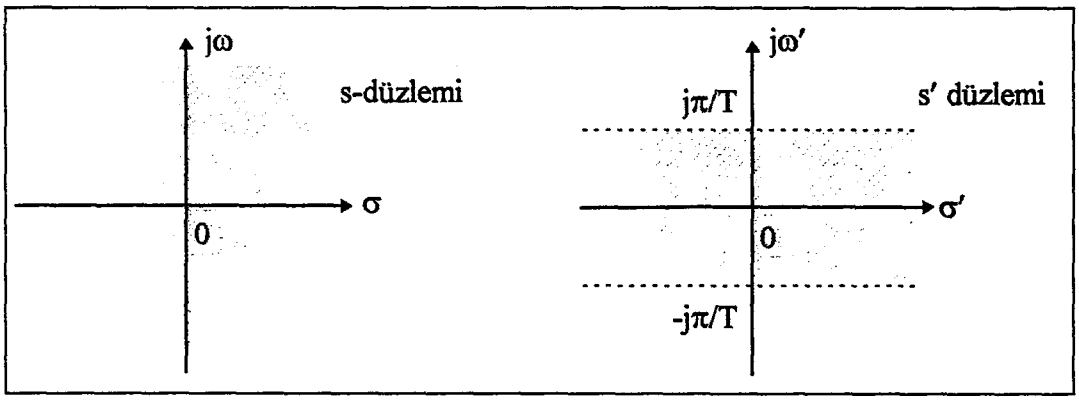
yazılabilir. Buradan uygunlaştırılmış z-dönüşüm metodu için;

$$G(z) = G(s)\Big|_{\substack{s+a_i=1-z^{-1}e^{-a_iT} \\ s+b_j=1-z^{-1}e^{-b}}} = K\frac{\prod_{i=1}^{m}(1-z^{-1}e^{-a_iT})}{\prod_{j=1}^{n}(1-z^{-1}e^{-b_jT})} \qquad (2.1.31)$$

yazılabilir.Burada K, z=1' de istenen kazancı verecek şekilde ayarlanabilir. Bu dönüşümle elde edilecek kutuplar, değişmez implus yanıtı yöntemiyle elde edilecek kutuplarla aynıdır. Ancak sıfırlar farklıdır. Bu farklılıknedeniyle bu dönüşüm band sınırlı olmayan girişler için kullanılabilir. Bu yöntemin uygulanabilmesi için G(s) kesinlikle çarpım formunda olmalıdır. Bu dönüşüm yüksek geçiren ve band durduran filtreler için iyi sonuçlar vermektedir.

2.1.4 Bilineer Dönüşüm

Değişmez impuls yanıtı yönteminde frekans spektrumunda meydana gelen örtüşmeyi önlemek için, s-düzleminden z-düzlemine bire bir (one-to-one) dönüşüme ihtiyaç vardır. $z=e^{sT}$ dönüşümü bire bir değil, çoktan bire (many-to-one) dönüşümdür. Bu problemi çözmek amacıyla ilk olarak tüm s-düzlemini $-\pi/T \leq Im[s'] \leq \pi/T$ arasına sıkıştıracak şekilde s-düzleminden s' düzlemine bir dönüşüm tanımlanır. Ardından s' den $z=e^{s'T}$ ile örtüşme problemi olmadan z-düzlemine dönüşüm gerçekleştirilir. s-düzleminden s' düzlemine olan bu dönüşüm şekil 2.1.7' de görülmektedir.



Şekil 2.1.7 s-düzleminden s' düzlemine dönüşüm

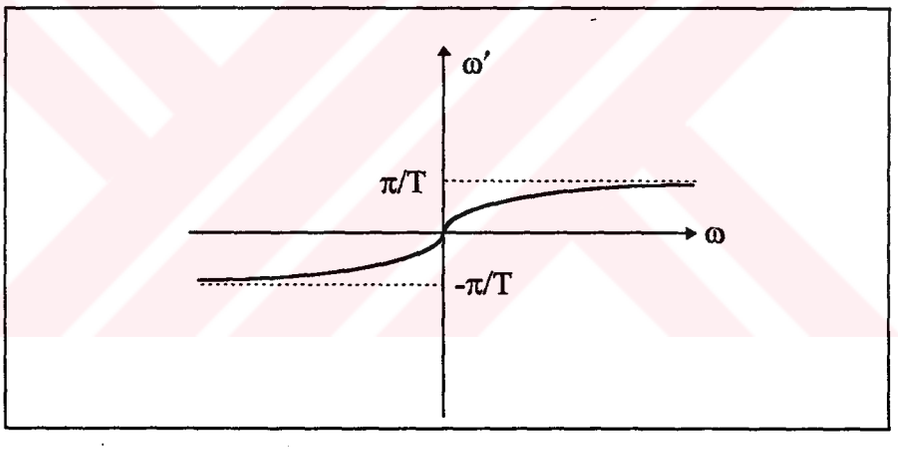
Bu amaçla bu iki düzlem arasında bire bir;

$$s' = \frac{2}{T}\tanh^{-1}\left[\frac{sT}{2}\right] \tag{2.1.32}$$

dönüşümü yapılır. s=jω ve s′=jω′ konularak;

$$\omega' = \frac{2}{T}\tan^{-1}\left(\frac{\omega T}{2}\right) \tag{2.1.33}$$

elde edilir. Şekil 2.1.8' de tüm jω ekseninin $-\pi/T \leq \omega' \leq \pi/T$ aralığına sıkıştırılması görülmektedir. İstenildiği gibi dönüşüm bire birdir.



Şekil 2.1.8 Bilineer dönüşüm ile ω' dan ω' ' e dönüşüm

(2.1.32)' nin tersinden gidilerek;

$$s = \frac{2}{T}\tanh\left(\frac{s'T}{2}\right) \tag{2.1.34}$$

ve $z=e^{sT}$, nin tersinden;

$$s' = \frac{1}{T} Inz \tag{2.1.35}$$

elde edilir. (2.1.34) ve (2.1.35)' den hareketle;

$$s = \frac{2}{T} \tanh(\frac{Inz}{2}) \tag{2.1.36}$$

bulunur. tanhx fonksiyonunun;

$$\tanh x = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{2.1.37}$$

özelliği kullanılarak (2.1.36) no' lu denklemden;

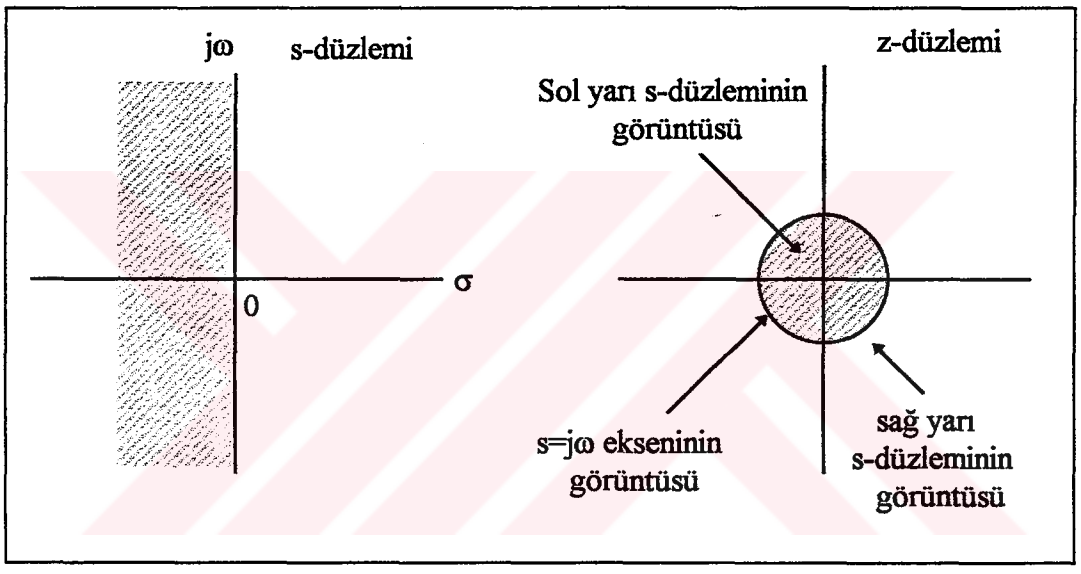
$$s = \frac{2}{T} \frac{(1 - z^{-1})}{(1 + z^{-1})} \tag{2.1.38}$$

ifadesi elde edilir. Buna göre bilineer dönüşüm formülü kullanılarak analog filtre transfer fonksiyonundan sayısal filtre tasarımı yapılabilir.

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T}\frac{(1-z^{-1})}{(1+z^{-1})}} \tag{2.1.39}$$

Bu dönüşümün doğasını anlamak için, s-düzleminden z-düzlemine geçişi anlatan şekil 2.1.9' a bakmak yeterli olmaktadır. Şekle göre bilineer dönüşümün özellikleri şunlardır:

1. Sağ yarı s-düzlem bölgesi z-düzleminde |z|=1 birim dairesinin dışında kalmaktadır.
2. s-düzleminde jω-ekseni üzerinde kalan noktalar |z|=1 birim dairesi üzerine düşer. Bilineer dönüşümün bu özelliğinden dolayı $|H_a(\omega)|$' nın maksimum ve minimumları $|H(e^{j\Omega})|$' da korunur. Böylece analog filtrenin geçirme ve söndürme bandlarına ait özellikler, sayısal filtrenin geçirme ve söndürme bandında gözükür.
3. Sol yarı s-düzlem bölgesi z-düzleminde |z|=1 birim dairesi içindeki noktalara karşı düşmektedir. Dönüşümün bu özelliğinden dolayı kararlı analog filtreden kararlı sayısal filtreye dönüşüm olacaktır.



Şekil 2.1.9 Bilineer dönüşüm ile s-düzleminden z-düzlemine geçiş

Dönüşüm sonucu elde edilen H(z)' nin pay derecesi paydanın derecesinden büyük olamaz. O halde, H(z) gerçeklenebilir bir fonksiyon olacaktır. Nedensellik özelliği korunacaktır.

Analog frekans ω ile sayısal frekans Ω arasında non-lineer bir ilişki bulunmaktadır. (2.1.38)' de s=jω ve $z=e^{j\Omega T}$ konularak;

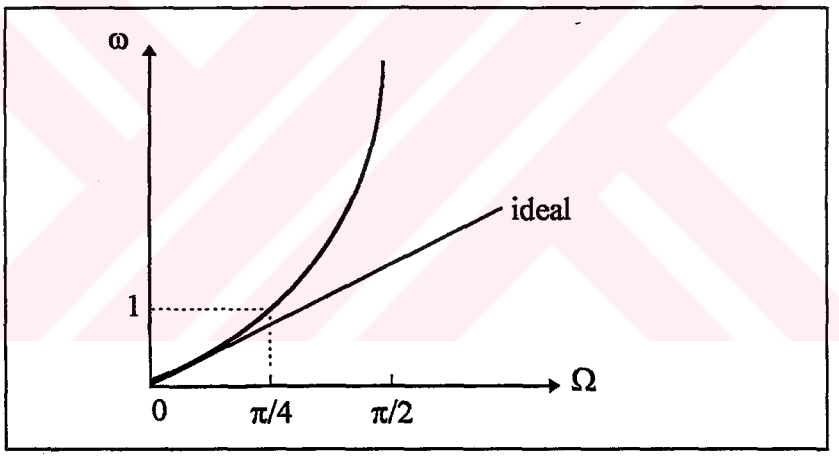
$$j\omega = \frac{2}{T}\frac{(1-e^{-j\Omega T})}{(1+e^{-j\Omega T})}$$

$$j\omega = \frac{2}{T}\frac{(e^{j(\Omega T/2)} - e^{-j(\Omega T/2)})}{(e^{j(\Omega T/2)} + e^{-j(\Omega T/2)})}$$

$$j\omega = \frac{2}{T} j \tan(\frac{\Omega T}{2})$$

$$\omega = \frac{2}{T}\tan(\frac{\Omega T}{2}) \qquad (2.1.40)$$

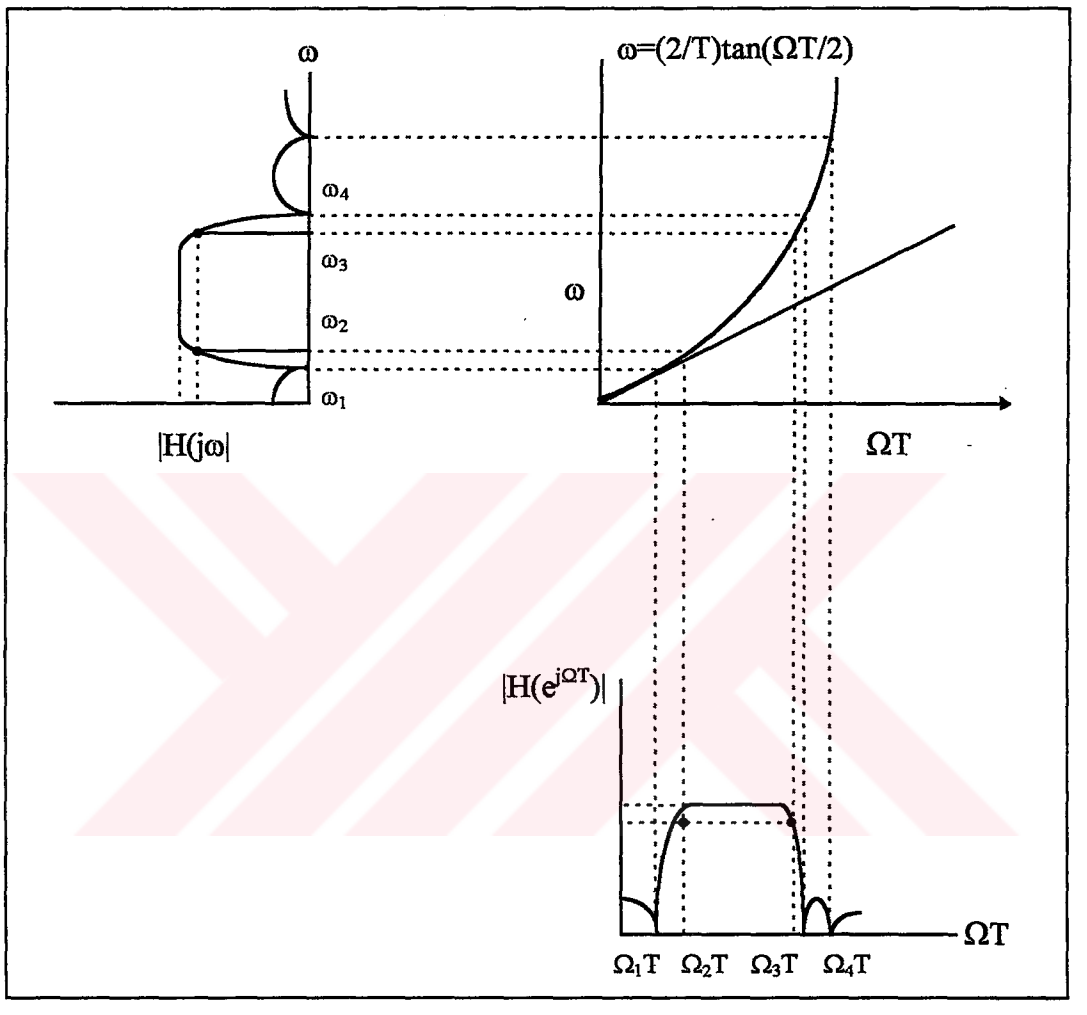
elde edilir. Bu non-lineer ilişki aşağıdaki şekilde T=2 durumu için görülmektedir. Küçük Ω değerleri için yaklaşık lineer olan bu dönüşüm, frekans arttıkça non-lineer olmaktadır.



Şekil 2.1.10 Bilineer dönüşümde analog ve sayısal frekans arasındaki ilişki

Bu, bilineer dönüşüm için önemli bir kısıtlama olup, sayısal filtrenin genlik yanıtında bozulmalara (warped) neden olmaktadır. Bu bozulmayı gidermek (kompanze etmek) için şekil 2.1.11' deki gibi bir yol izlenebilir. Burada görüldüğü gibi eğer kritik frekansları ($\Omega_1, \Omega_2, \Omega_3, \Omega_4$) olan bir sayısal filtre tasarlanacaksa öncelikli olarak bu değerler (2.1.40) bağıntısı kullanılarak, yeni analog ($\omega_1, \omega_2, \omega_3, \omega_4$) kritik frekanslarına dönüştürülür. Sonuç olarak istenen özellikleri sağlayan analog filtre elde edildikten sonra, bilineer dönüşüm kullanılarak sayısal filtreye geçiş yapılır.

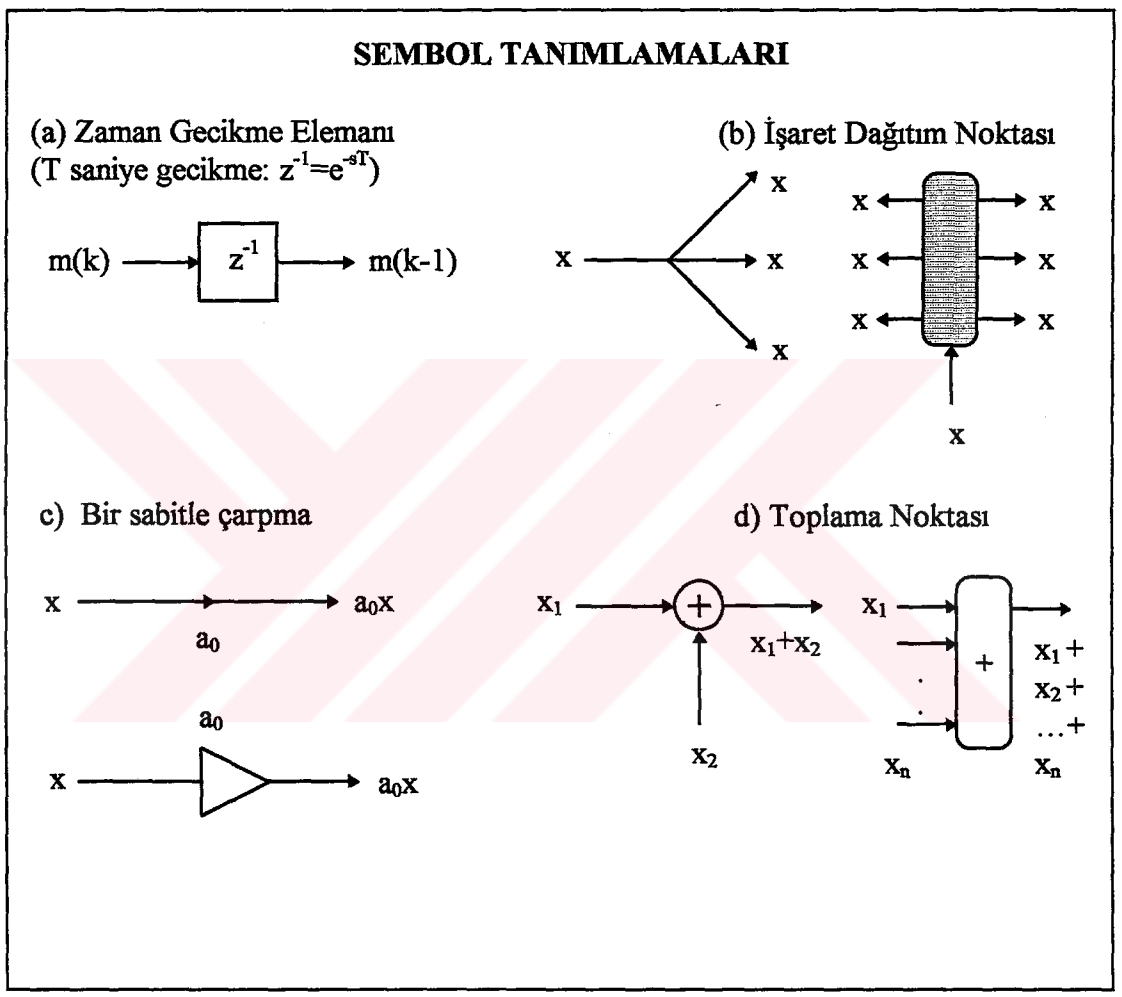
Bilineer dönüşümün dezavantajı analog filtrenin gerek impuls ve gerekse faz yanıtının korunamamasıdır.



Şekil 2.1.11 Bilineer dönüşümde meydana gelen frekans yanıtındaki bozulmanın (warped) kompanze edilmesi (prewarped) işlemi

2.2 SAYISAL FİLTRE YAPILARI [1]

Filtreleme ve control problemlerinde bazı istenen karakteristikleri sağlayan sayısal filtrenin z-domeninde transfer fonksiyonunun bulunması gerekmektedir. Bu transfer fonksiyonu aşağıdaki şekilde genel olarak ifade edilebilir:



$$D(z)=\frac{a_0+a_1z^{-1}+\cdots+a_nz^{-n}}{1+b_1z^{-1}+\cdots+b_nz^{-n}} \qquad (2.2.1)$$

Burada a_i ve b_i' ler reel katsayılar, n ise pay ve payda polinomlarının en yüksek mertebesine eşittir. Bu ifade şekli genel olup pay ve payda polinomlarının yüksek

mertebeli terimlerinin katsayıları sıfır olabilir. D(z) transfer fonksiyonunun zaman-gecikme (time-delay) elemanları, toplayıcılar (adders) ve çarpıcılar (multipliers) kullanarak blok diyagram olarak gerçeklenmesi "Filtre Yapıları" olarak adlandırılır. D(z) transfer fonksiyonunu gerçekleyen birçok yapı oluştulabilir. Bu yapılardan en önemlileri:

- Direkt-form yapıları
- İkinci-derece modüller
  - Kaskad bağlı ikinci-derece modüller
  - Paralel bağlı ikinci-derece modüller

2.2.1 Direkt Yapılar

2.2.1.1 Birinci Direkt Yapı (1D)

Sayısal filtrenin transfer fonksiyonu D(z) aşağıdaki şekilde tanımlanmış olsun.

$$D(z) = \frac{\sum_{i=0}^{n} a_i z^{-i}}{\sum_{i=0}^{n} b_i z^{-i}} \qquad (2.2.2)$$

Burada b_0=1' dir. Eğer X(z) filtrenin girişi ve Y(z) de filtrenin çıkışı olarak kabul edilirse;

$$\frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{n} a_i z^{-i}}{\sum_{i=0}^{n} b_i z^{-i}} \qquad (2.2.3)$$

yazılabilir. M(z) bir ara değişken olarak ele alınırsa;

$$\frac{Y(z)}{M(z)} \cdot \frac{M(z)}{X(z)} = \frac{\sum_{i=0}^{n} a_i z^{-i}}{\sum_{i=0}^{n} b_i z^{-i}} \tag{2.2.4}$$

yazılabilir. Bu takdirde;

$$\frac{Y(z)}{M(z)} = \sum_{i=0}^{n} a_i z^{-i} \tag{2.2.5}$$

$$\frac{X(z)}{M(z)} = \sum_{i=0}^{n} b_i z^{-i} \tag{2.2.6}$$

olur. Buradan hareketle;

$$X(z) = \sum_{i=0}^{n} b_i z^{-i} M(z) \tag{2.2.7}$$

veya

$$M(z) = X(z) - \sum_{i=1}^{n} b_i z^{-i} M(z) \tag{2.2.8}$$

ve

$$Y(z) = \left(\sum_{i=0}^{n} a_i z^{-i} \right) M(z) \tag{2.2.9}$$

yazılabilir. Zaman domeninde de;

$$m(k) = x(k) - \sum_{i=1}^{n} b_i m(k-i) \tag{2.2.10}$$

$$y(k) = \sum_{i=0}^{n} a_i m(k-i) \tag{2.2.11}$$

olur. Denklem (2.2.10) ve (2.2.11) şekil 2.2.1a' da gösterilen "birinci direkt" yapıyı tanımlamaktadır. Birinci direkt yapı (1D) yalnızca n adet zaman-gecikme elemanı (time-delay element) içerdiğinden kanonik bir yapıdır.

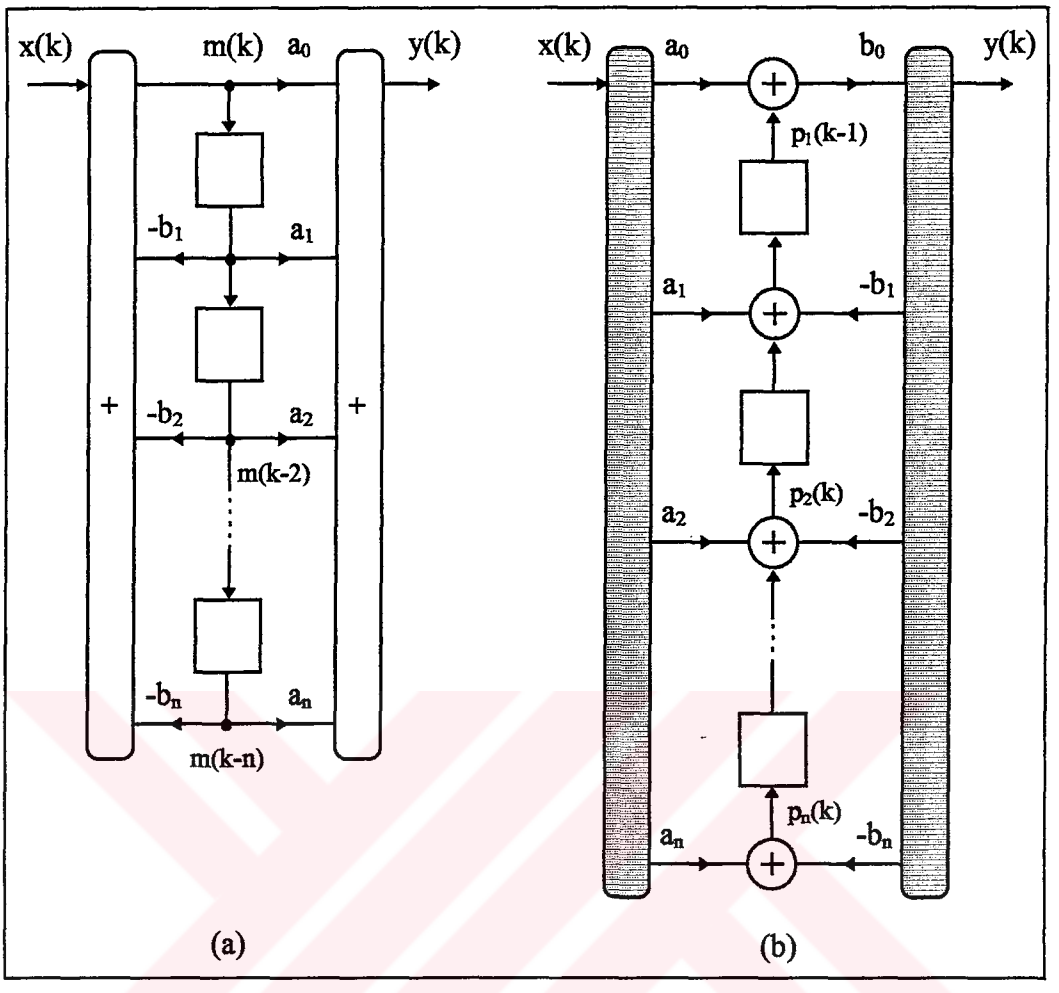
2.2.1.2 İkinci Direkt Yapı (2D)

Bir sayısal filtrenin eviriği (transpozesi) için, blok diyagramın tüm dallarındaki işaret akış yönleri ters çevrilmeli, ardından toplama düğümlerini işaret dağıtım noktaları haline, giriş de çıkış haline getirilmelidir. Bunların tersi de doğrudur. Bir sayısal filtre yapısının eviriği de aynı transfer fonksiyonuna sahiptir.

Eğer birinci direkt yapının transpozesi alınırsa şekil 2.2.1.b' de gösterilen ikinci direkt yapı elde edilir. Bu yapı ayrıca (2.2.1) no' lu denkleme uygulanabilir, ancak (n+1) tane fark denklemleri (toplam noktaları) gerektirir. Buna karşılık bu sayı 1D yapısında sadece ikidir. 2D yapısı için aşağıdaki denklemler yazılabilir:

$$\begin{aligned} p_i(k) &= p_{i+1}(k-1) + a_i x(k) - b_i y(k) \quad ; i := 1, n-1 \\ p_n(k) &= a_n x(k) - b_n y(k) \\ y(k) &= a_0 x(k) + p_1(k-1) \end{aligned} \tag{2.2.12}$$

İkinci direkt yapı da kanoniktir.



Şekil 2.2.1 (a) 1D sayısal filtre yapısı, (b) 2D sayısal filtre yapısı

2.2.1.3 Üçüncü Direkt Yapı (3D)

(2.2.1)' e geri dönerek D(z) aşağıdaki şekilde yazılabilir:

$$D(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{n} a_i z^{-i}}{\sum_{i=0}^{n} b_i z^{-i}} \qquad (2.2.13)$$

Buradan hareketle;

$$Y(z)\sum_{i=0}^{n} b_i z^{-i} = X(z)\sum_{i=0}^{n} a_i z^{-i} \tag{2.2.14}$$

$$Y(z) = \sum_{i=0}^{n} a_i z^{-i} X(z) - \sum_{i=1}^{n} b_i z^{-i} Y(z) \tag{2.2.15}$$

yazılabilir. Benzer şekilde zaman domeninde;

$$y(n) = \sum_{i=0}^{n} a_i x(n-i) - \sum_{i=1}^{n} b_i y(n-i) \tag{2.2.16}$$

olur. Bu, şekil 2.2.1.c' de blok diyagramı gösterilen üçüncü direkt yapının (3D) fark denklemidir. Dikkat edilirse bu yapıda sadece bir tane toplama noktası olmasına karşın 2n tane zaman-gecikme elemanı vardır.

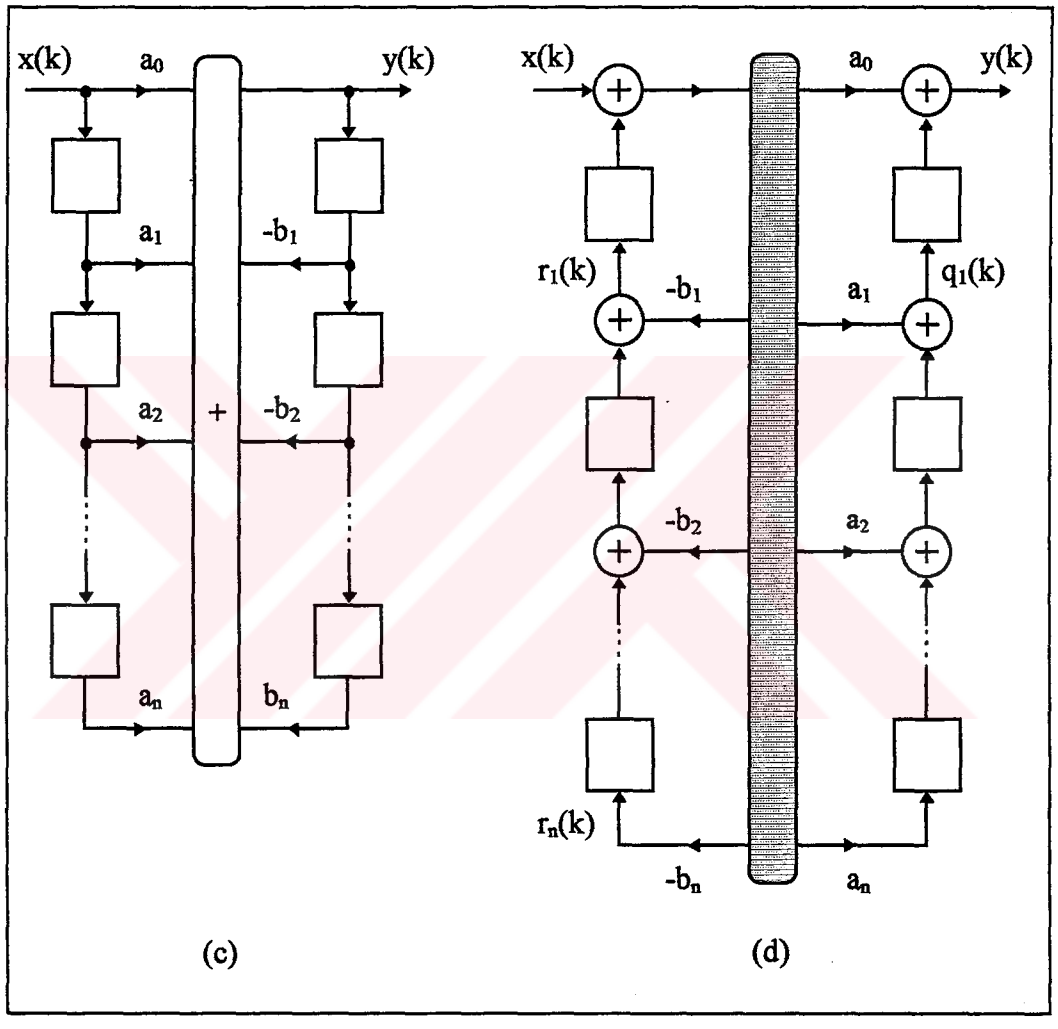
2.2.1.4 Dördüncü Direkt Yapı (4D)

Üçüncü direkt yapının eviriği (transpozesi) alınarak şekil 2.2.1.d'de gösterilen dördüncü direkt yapı (4D) elde edilir. Bu yapı sadece bir tane işaret dağıtım noktası (sinyal distribution point) içermesine karşın aşağıda belirtilen 2n tane fark denklemiyle ifade edilebilir:

$$\begin{aligned} r_0(k) &= x(k) + r_1(k-1) \\ q_n(k) &= a_n(k) r_0(k) \\ r_n(k) &= -b_n(k) r_0(k) \\ q_i(k) &= a_i r_0(k) + q_{i+1}(k-1), \quad i = 1, n-1 \\ r_i(k) &= -b_i r_0(k) + r_{i+1}(k-1) \\ y(k) &= a_0 r_0(k) + q_1(k-1) \end{aligned} \tag{2.2.17}$$

Bu dört yapının karakteristikleri tablo 2.2.1' de görülmektedir. Dikkat edilirse 1D ve 2D zaman-gecikme elemanı, 1D ve 3D' de toplama noktası bakımından avantajlıdır. Bilgisayar uygulamalarında gecikme için bellek alanı gerekir ve bellekler de bağıl olarak

pahalı değildir. Toplama noktası da dizayn açısından kolaylık sağlar. Tüm bu direkt filtre yapıları büyük n değerlerinde katsayı duyarlılığı problemiyle karşı karşıyadır. Öyle ki a_i ve b_i katsayılarında meydana gelen küçük değişiklikler D(z) transfer fonksiyonun kutup ve sıfırlarında büyük değişimlere neden olur. Meydana gelen bu değişimler de doğal olarak sayısal filtrenin genlik frekans yanıtında hatalara neden olur.



Şekil 2.2.1 (c) 3D sayısal filtre yapısı, (d) 4D sayısal filtre yapısı

Tablo 2.2.1 Direkt yapıların özellikleri (n:Filtre derecesi)

Direkt Sayısal Filtre Yapılarının Özellikleri	1D	2D	3D	4D
Zaman-Gecikme Elemanı	n	n	2n	2n
Çarpıcılar	2n+1	2n+1	2n+1	2n+1
Toplama Noktaları	2	n+1	1	2n
İşaret Dağıtım Noktaları	n+1	2	2n	1

2.2.2 İkinci derece modüller

Direkt yapılarda karşılaşılan katsayı duyarlılığı probleminden kaçınmak için D(z) transfer fonksiyonu genellikle ikinci-derece modüllerin kaskad veya paralel kombinasyonu şekline dönüştürülür.

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{2.2.18}$$

İkinci-derece modüller kullanarak 1D, 2D, 3D, 4D yapıları oluşturulabilir. Şekil 2.2.2' de ikinci-derece modüller için 1D, 2D, 3D ve 4D yapıları görülmektedir. Herbir yapı için fark denklemleri aşağıdaki biçimde tanımlanabilir:

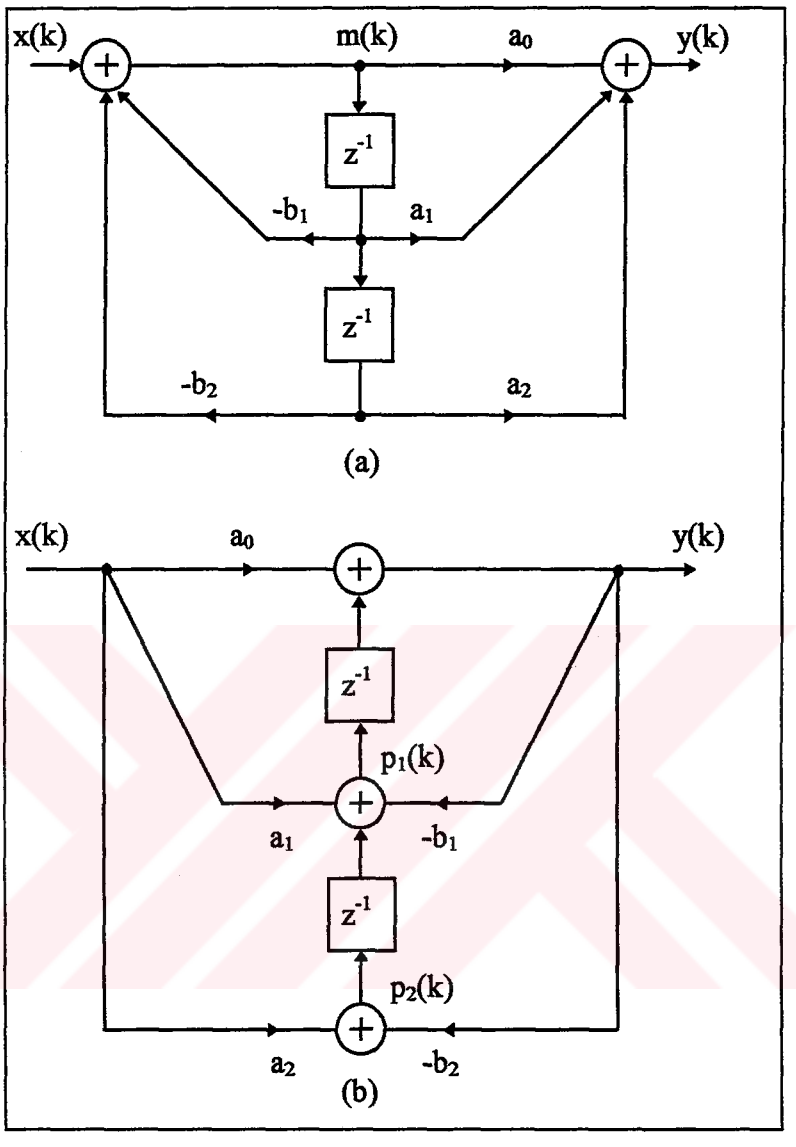
$$\begin{aligned} 1D:\ & m(k) = x(k) - b_1 m(k-1) - b_2 m(k-2) \\ & y(k) = a_0 m(k) + a_1 m(k-1) + a_2 m(k-2) \end{aligned} \tag{2.2.19}$$

$$\begin{aligned} 2D:\ & y(k) = a_0 x(k) + p_1(k-1) \\ & p_1(k) = a_1 x(k) - b_1 y(k) + p_2(k-1) \\ & p_2(k) = a_2 x(k) - b_2 y(k) \end{aligned} \tag{2.2.20}$$

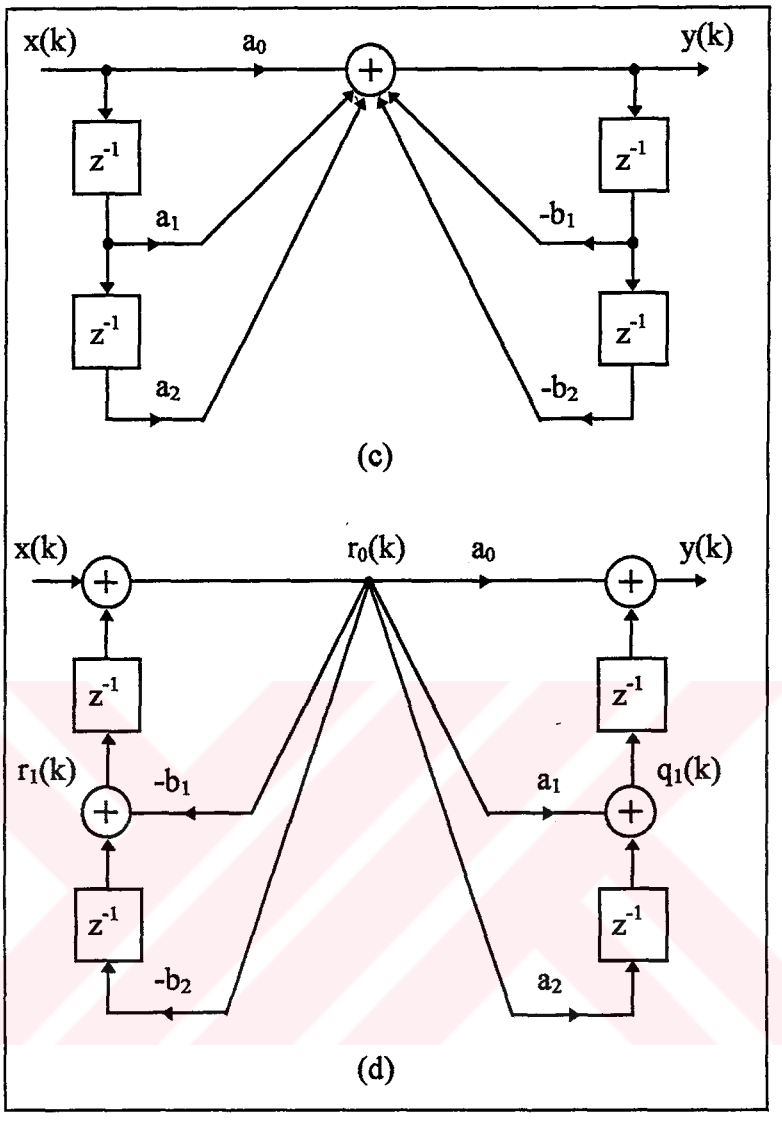
$$3D:\ y(k) = a_0 x(k) + a_1 x(k-1) + a_2 x(k-2) - b_1 y(k-1) - b_2 y(k-2) \tag{2.2.21}$$

$$\begin{aligned} 4D:\ r_0(k) &= x(k) + r_1(k-1) \\ y(k) &= a_0 r_0(k) + q_1(k-1) \\ r_1(k) &= -b_1 r_0(k) - b_2 r_0(k-1) \\ q_1(k) &= a_1 r_0(k) + a_2 r_0(k-1) \end{aligned} \tag{2.2.22}$$

Yukarıda belirtilen herbir denklem takımında işlemler verilen sırada yapılmalıdır. Örnek olarak (2.2.20) no' lu denklemde $p_1(k)$' nın hesaplanabilmesi için y(k)' nın daha önceden bilinmesi gerekmektedir. Tüm durumlarda y(k), giriş örneği x(kT) ve filtrenin çıkışı $y(kT+\tau_c)$ aralığında mümkün olan en kısa τ_c zaman gecikmesi ile hesaplanmalıdır. İdeal olarak $\tau_c=0$ olmalıdır, ancak pratikde bunu elde etmek mümkün değildir. Bunun yerine bu τ_c zamanını minimize edecek şekilde belli bir sırayla işlem yapılmalıdır. Eğer $T>>\tau_c$ ise τ_c ihmal edilebilir.



Şekil 2.2.2 Direkt ikinci-derece modüller (a) 1D, (b) 2D yapısı



Şekil 2.2.2 Direkt ikinci-derece modüller: (c) 3D, (d) 4D yapısı

İkinci-derece modüller kullanarak başka yapılar oluşturmak da mümkündür. Şekil 2.2.3 ve 2.2.4' de 1X ve 2X çapraz-bağlama (cross-coupled) yapısı görülmektedir. Bu yapı daha çok kompleks-kutup çiftleri için önerilmektedir. 1X yapısı için aşağıdaki fark denklemleri yazılabilir:

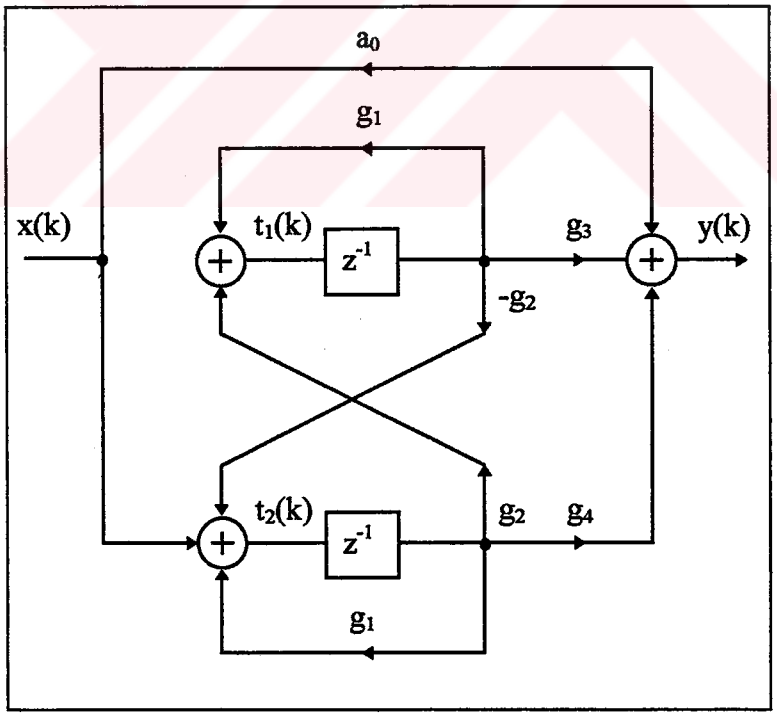
$$
\begin{aligned}
y(k) &= a_0 x(k) + s_2(k-1) \\
s_1(k) &= g_1 s_1(k-1) - g_2 s_2(k-1) + g_3 x(k) \\
s_2(k) &= g_1 s_2(k-1) + g_2 s_1(k-1) + g_4 x(k)
\end{aligned}
\tag{2.2.23}
$$

Bu denklem takımında yer alan g_i' ler aşağıdaki şekilde verilir:

$$
D(z) = a_0 + \frac{A}{z+p} + \frac{A^*}{z+p^*}
\tag{2.2.24}
$$

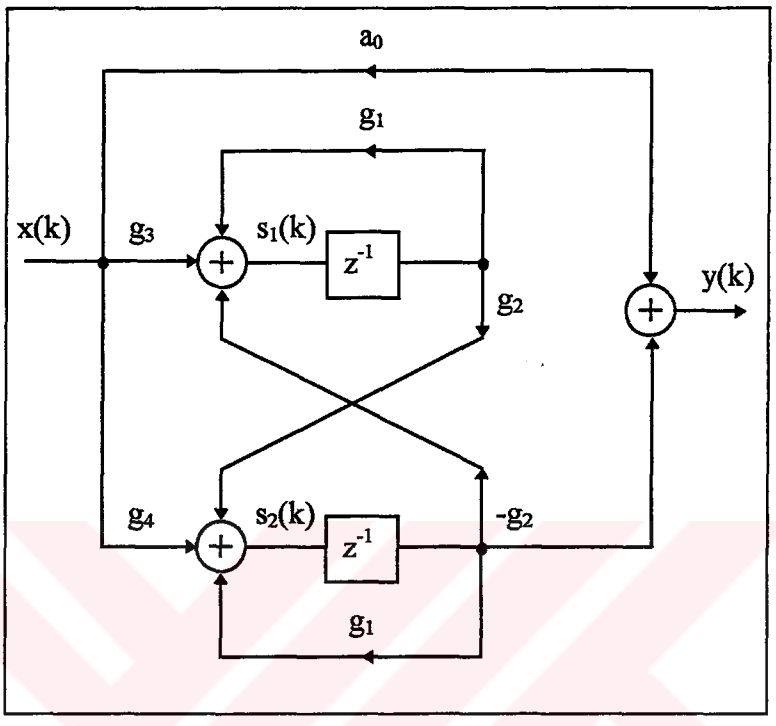
olmak üzere;

$$
\begin{aligned}
g_1 &= -\mathrm{Re}[p] \\
g_2 &= -\mathrm{Im}[p] \\
g_3 &= 2\,\mathrm{Im}[A] \\
g_4 &= 2\,\mathrm{Re}[A]
\end{aligned}
\tag{2.2.25}
$$



Şekil 2.2.3 1X sayısal filtre yapısı

Dikkat edilirse 1X yapısı da kanonik bir yapıdır. 1X yapısının eviriği alınarak şekil 2.2.4' de gösterilen 2X yapısı elde edilebilir.



Şekil 2.2.4 2X sayısal filtre yapısı

2X yapısı için de aşağıdaki fark denklemleri yazılabilir:

$$\begin{aligned} y(k) &= a_0 x(k) + g_3 t_1(k-1) + g_4 t_2(k-1) \\ t_1(k) &= g_1 t_1(k-1) + g_2 t_2(k-1) \\ t_2(k) &= x(k) + g_1 t_2(k-1) - g_2 t_1(k-1) \end{aligned} \qquad (2.2.26)$$

Bu denklem takımında kullanılan g_i' ler (2.2.25)' de tanımlanmıştır. Tablo 2.2.2' de ikinci-derece modüllerle gerçeklenmiş bu altı yapının karşılaştırılması görülmektedir. 1X ve 2X yapıları direkt yapılardan farklı olarak ilave iki çarpıcı gerektirmektedir.

Tablo 2.2.2 İkinci-derece modül yapılarının karşılaştırılması

İkinci-derece modüllerle gerçeklenen yapılar	1D	2D	3D	4D	1X	2X
Zaman-Gecikme Elemanı	2	2	4	4	2	2
Çarpıcılar	5	5	5	5	7	7
Toplama Noktaları	2	3	1	4	3	3
İşaret Dağıtım Noktaları	3	2	4	1	3	3

2.2.2.1 İkinci Derece Modüllerle Kaskad Gerçekleme

Katsayı duyarlılığı probleminden kaçınmak amacıyla (2.2.1) no' lu denklemde belirtilen D(z) transfer fonksiyonu ikinci derece çarpanlarına ayrılarak kaskad bağlı bu modüllerin çarpımı şeklinde ifade edilebilir:

$$D(z) = \frac{\prod_{i=1}^{m} (\alpha_{i0} + \alpha_{i1} z^{-1} + \alpha_{i2} z^{-2})}{\prod_{i=1}^{m} (1 + \alpha_{i3} z^{-1} + \alpha_{i4} z^{-2})} \qquad (2.2.27)$$

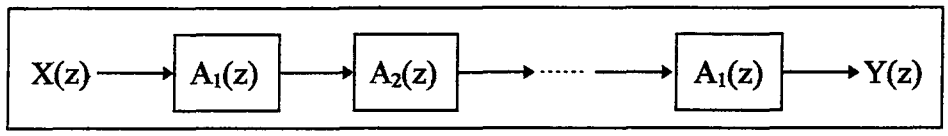
(2.2.27) no' lu denklemde m (n/2)' den büyük veya eşit en küçük tamsayıdır. D(z) transfer fonksiyonunu aşağıdaki şekilde yazmak mümkündür:

$$D(z) = \prod_{i=1}^{m} A_i(z) \qquad (2.2.28)$$

Burada;

$$A_i(z) = \frac{\alpha_{i0} + \alpha_{i1} z^{-1} + \alpha_{i2} z^{-2}}{1 + \alpha_{i3} z^{-1} + \alpha_{i4} z^{-2}} \qquad (2.2.29)$$

olarak yazılır. Şekil 2.2.5' de ikinci-derece modüllerin kaskad bağlanması görülmektedir.



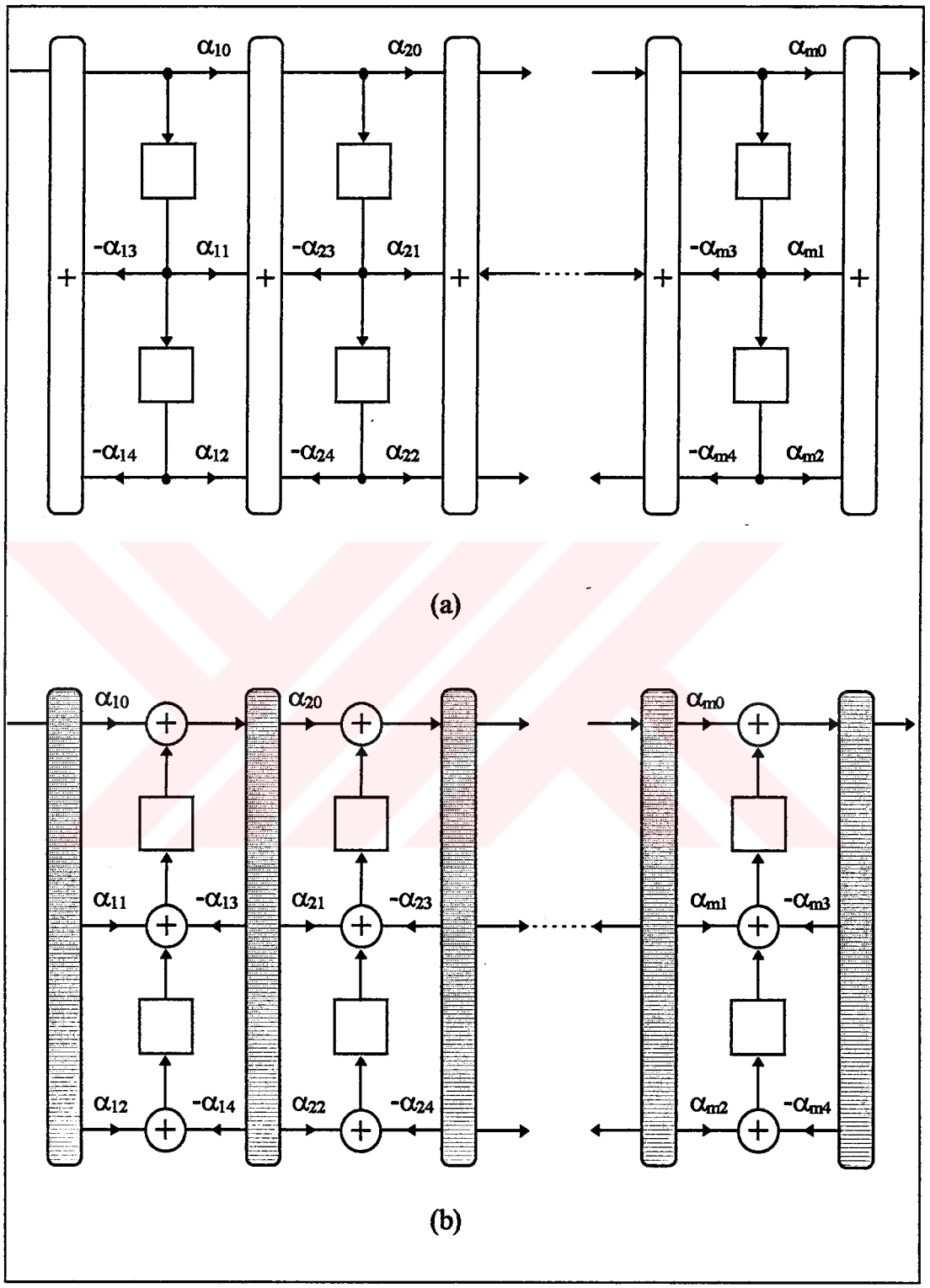
Şekil 2.2.5 Kaskad bağlı ikinci derece modüller

Şekil 2.2.6' da kaskad bağlı ikinci-derece modüller kullanarak direkt yapıların oluşturulması görülmektedir. Tablo 2.2.3' de bu yapıların karşılaştırılması yapılmaktadır. Tablo 2.2.1 ve Tablo 2.2.3 karşılaştırıldığında ikinci-derece modüllerin kaskad bağlanmasıyla elde edilen 3D ve 4D yapısında, direkt yapıya göre n-2 tane daha az zaman-gecikme elemanı kullanılmıştır. Kaskad yapıda herbir durum için ilave çarpıcılara gereksinim vardır. Kaskad direkt modüller m-1 tane ek toplama ve dağıtma noktasına gereksinim duymaktadır.

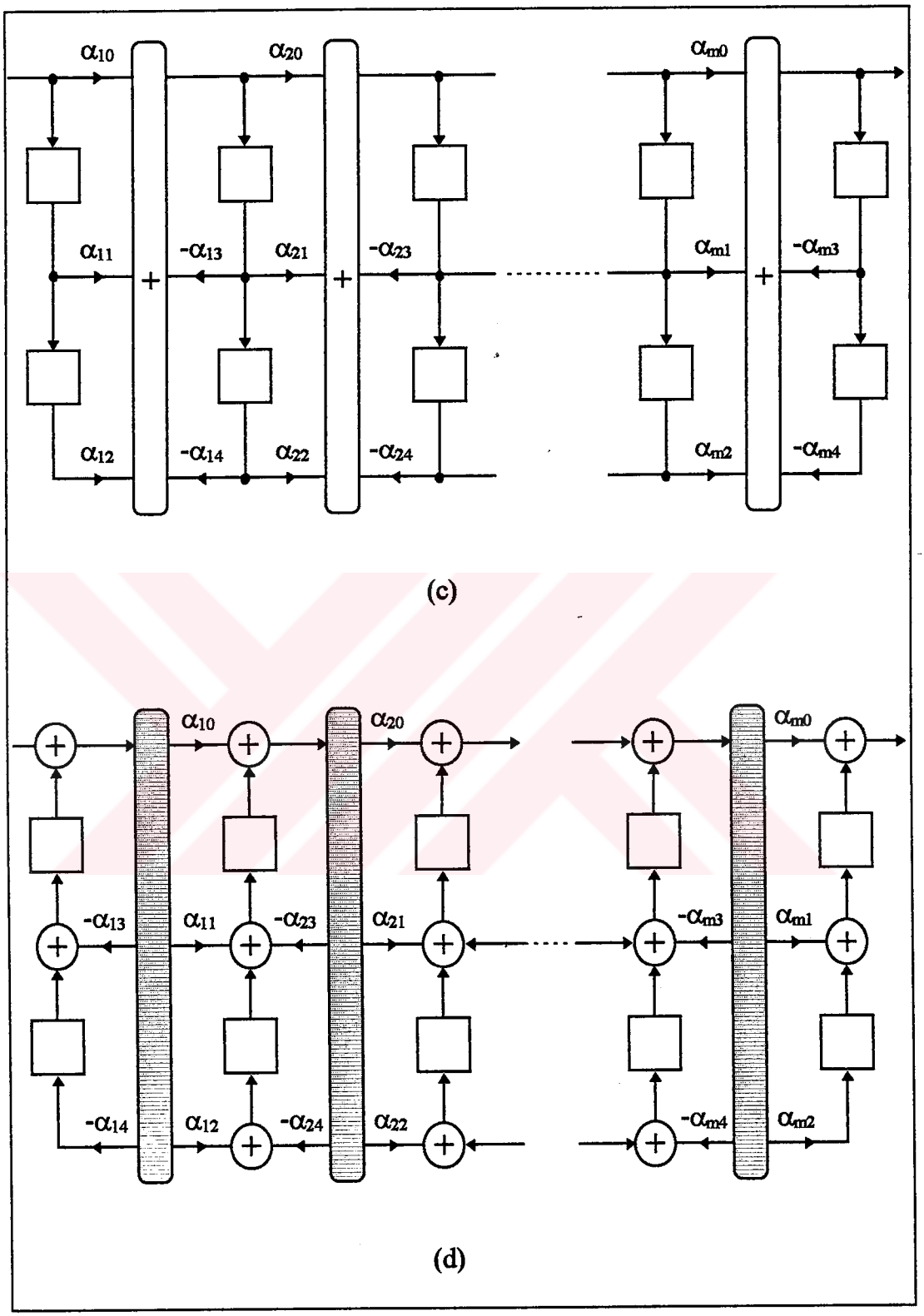
Tablo 2.2.3 Kaskad yapılar

Kaskad Sayısal Filtre Yapılarının Özellikleri	1D	2D	3D	4D
Zaman-Gecikme Elemanı	2m (n)	2m (n)	2m+2 (2n-(n-2))	2m+2 (2n-(n-2)
Çarpıcılar	5m (n+n+m)	5m (n+n+m)	5m (2n+m)	5m (2n+m)
Toplama Noktaları	m+1 (2+m-1)	3m (n+m)	m	3m+1 (2n-(m-1))
İşaret Dağıtım Noktaları	3m (n+m)	m+1 (2+m-1)	3m+1 (2n-(m-1))	m

Not: m (n/2)' ye eşit veya daha büyük en küçük tamsayıdır. Tabloda parantez içindeki sayılar m=n/2 durumu için geçerlidir.



Şekil 2.2.6 Kaskad bağlı (a) 1D, (b) 2D sayısal filtre yapıları



Şekil 2.2.6 Kaskad bağlı (c) 3D, (d) 4D sayısal filtre yapıları

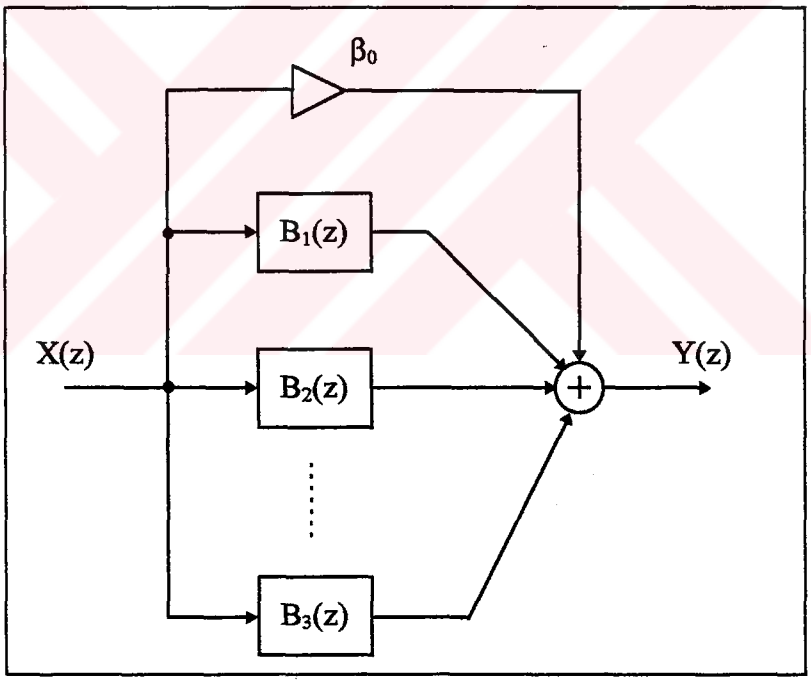
2.2.2.2 İkinci Derece Modüllerle Paralel Gerçekleme

Katsayı duyarlılığı probleminden kaçınmak amacıyla izlenecek bir diğer metod da D(z) transfer fonksiyonunun payda polinomunu çarpanlarına ayırmak ve (2.2.30) ifadesini elde etmek için kısmı kesirlere açılım yapmadır.

$$D(z) = \beta_0 + \sum_{i=1}^{m} B_i(z) \tag{2.2.30}$$

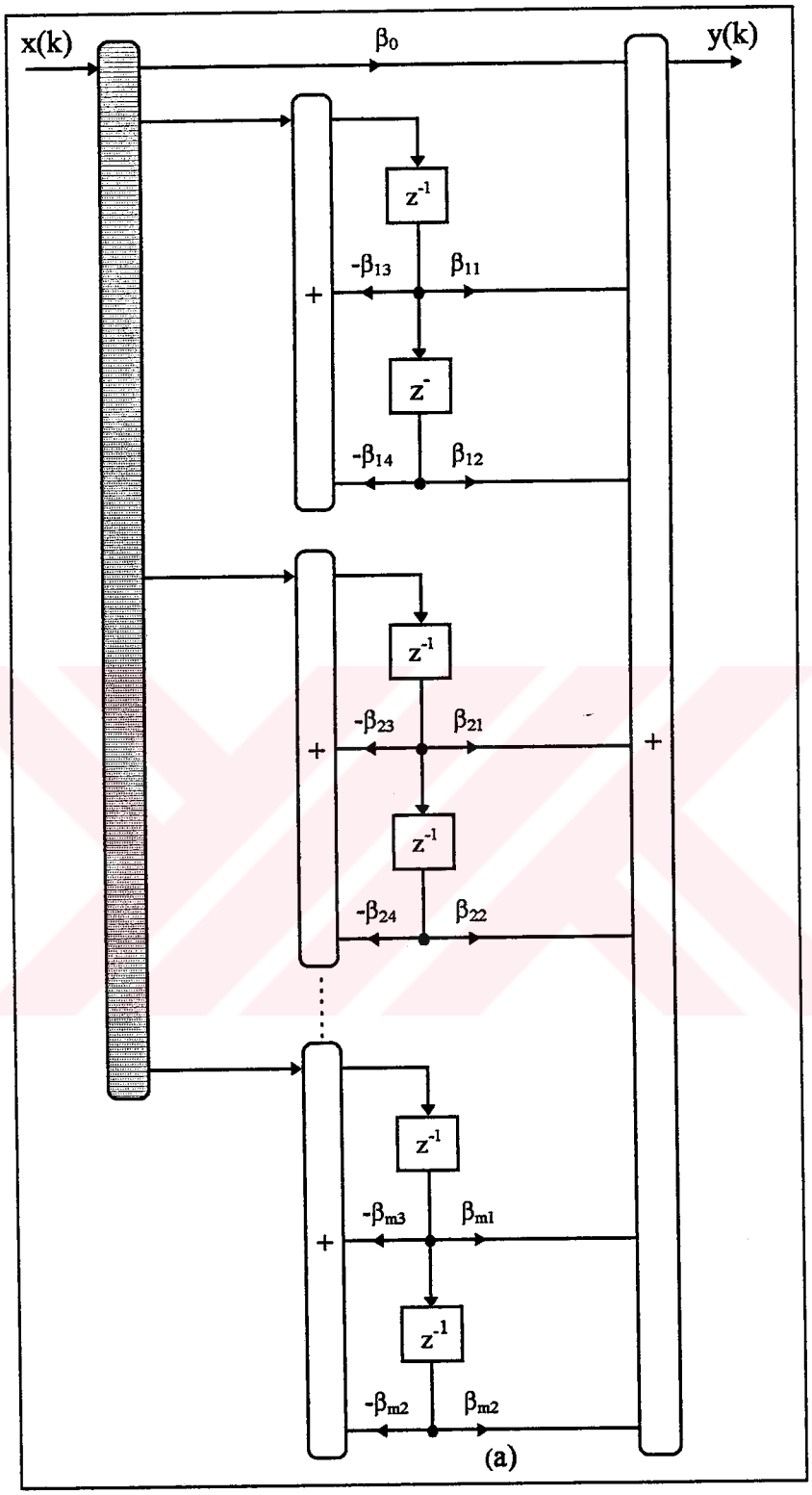
Burada;

$$B_i(z) = \frac{\beta_{i1} z^{-1} + \beta_{i2} z^{-2}}{1 + \beta_{i3} z^{-1} + \beta_{i4} z^{-2}} \tag{2.2.31}$$

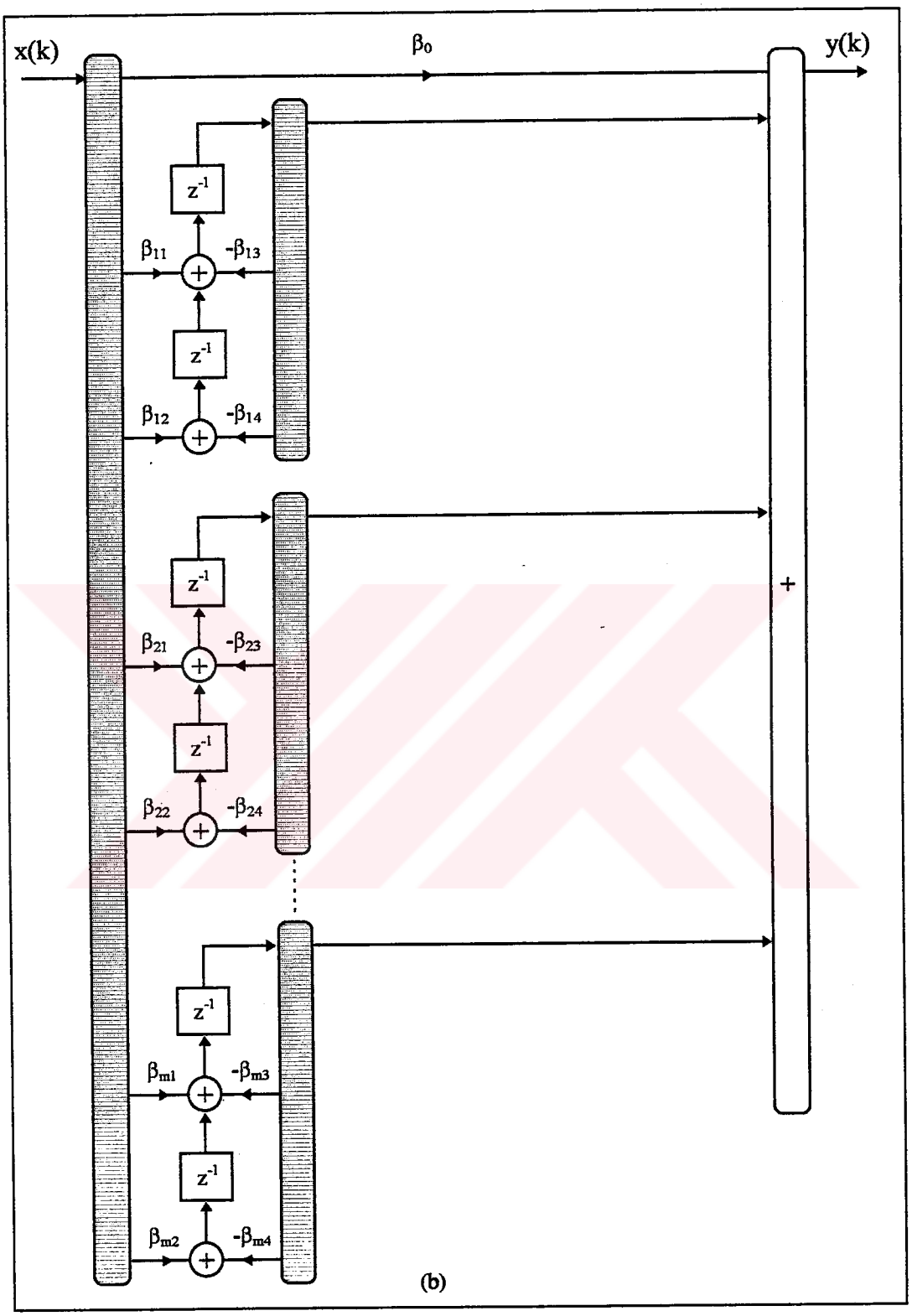


Şekil 2.2.7 Paralel yapı

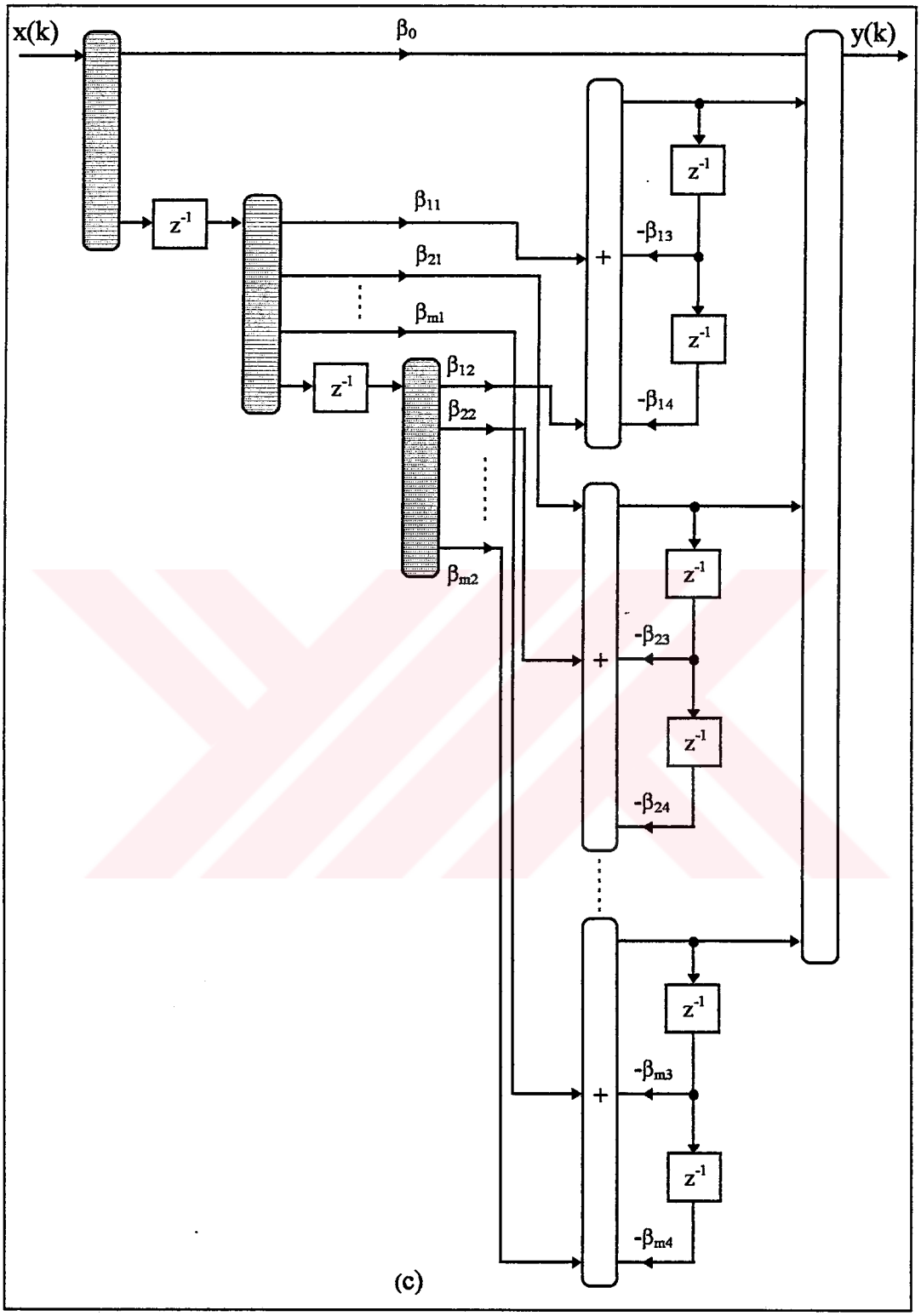
ifadesiyle verilebilir. Şekil 2.2.7' de paralel yapı görülmektedir. Tablo 2.2.2' deki altı yapıdan herhangi biri şekil 2.2.7' deki bloklar için kullanılabilir. Eğer direkt yapı kullanılırsa kaskad yapı da olduğu gibi bazı elemanlar ortak olarak kullanılabilir.



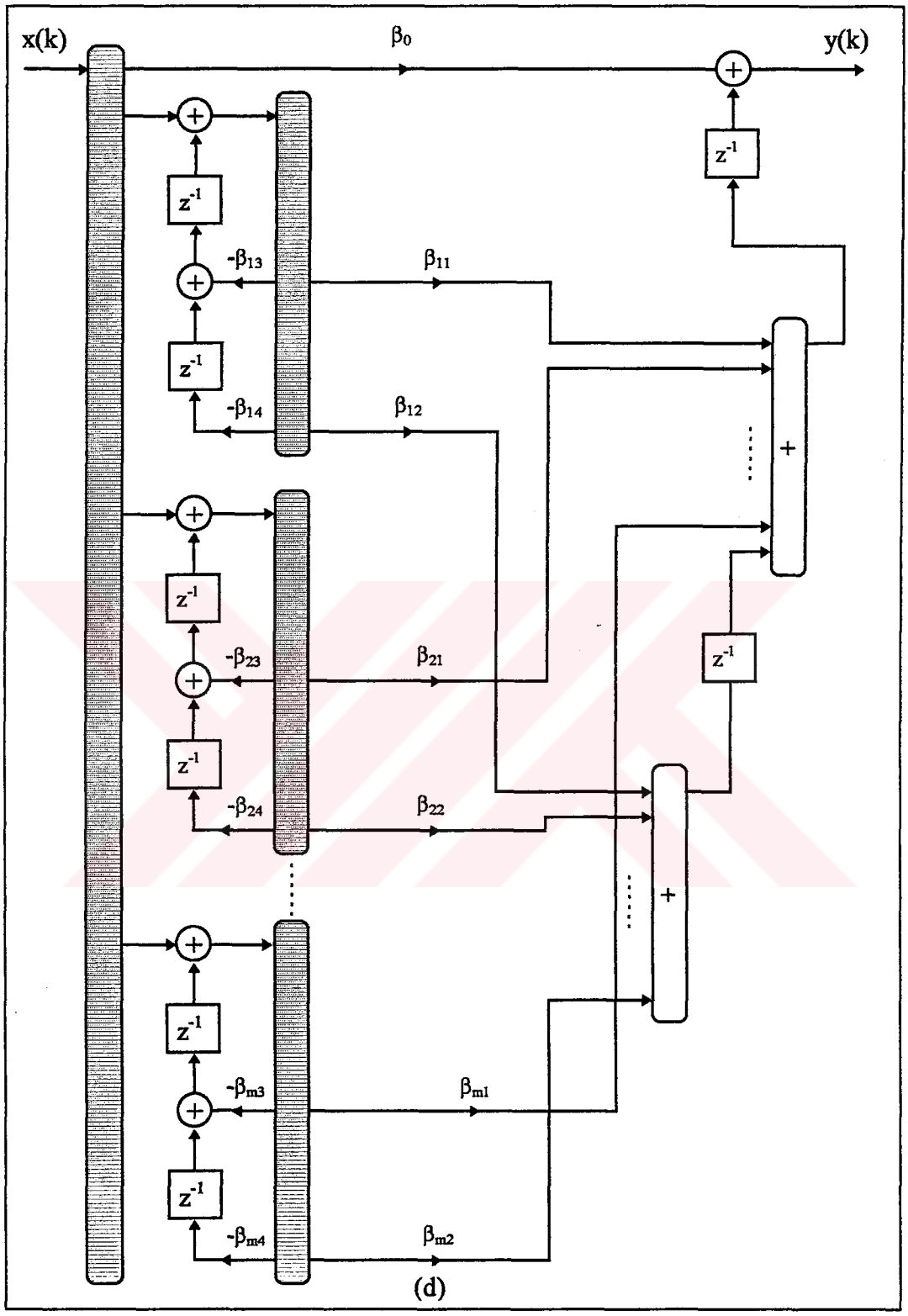
Şekil 2.2.8a Paralel bağlı ikinci derece modüllerle oluşturulan 1D yapısı



Şekil 2.2.8b Paralel bağlı ikinci derece modüllerle oluşturulan 2D yapısı



Şekil 2.2.8c Paralel bağlı ikinci derece modüllerle oluşturulan 3D yapısı



Şekil 2.2.8d Paralel bağlı ikinci derece modüllerle oluşturulan 4D yapısı

Şekil 2.2.8' de direkt paralel yapılar görülmektedir. Tablo 2.2.4' de ise bu direkt paralel yapıların karakteristikleri karşılaştırılmaktadır. Dikkat edilirse paralel direkt 3D ve 4D yapısında Tablo 2.2.1' deki direkt gerçeklemelere göre gereken zaman-gecikme elemanı sayısı n-2 tane daha azdır. Çarpıcıların sayısı Tablo 2.2.1' dekilerle aynıdır. 1D yapısı ilave m-1 toplama noktasına, 2D yapısı ilave m-1 işaret dağıtım noktasına gereksinim duymaktadır. 3D yapısı Tablo 2.2.1' e göre ek olarak m tane toplama noktasına ve n-3 tane de ilave işaret dağıtım noktasına gereksinim duymaktadır. Bu durumun tersi de 4D paralel yapısı için gereklidir. (İlave m işaret dağıtım noktası, n-3 tane ilave toplama noktası)

Tablo 2.2.4 Paralel Yapılar

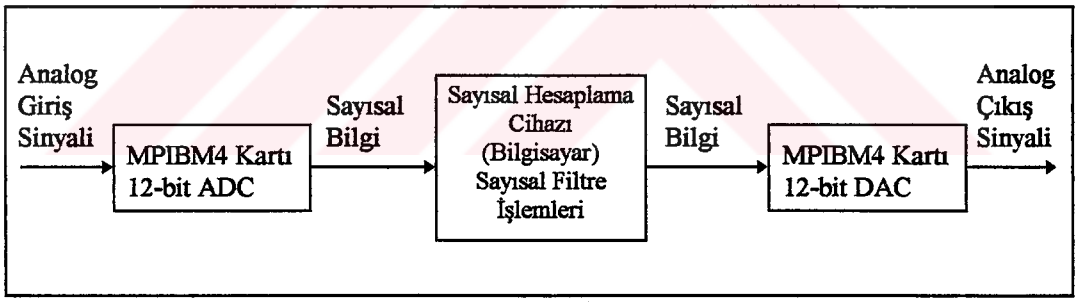
Paralel Sayısal Filtre Yapılarının Özellikleri				
	1D	2D	3D	4D
Zaman-Gecikme Elemanı	2m (n)	2m (n)	2m+2 (2n-(n-2))	2m+2 (2n-(n-2)
Çarpıcılar	4m+1 (2n+1)	4m+1 (2n+11)	4m+1 (2n+11)	4m+1 (2n+11)
Toplama Noktaları	m+1 (2+m-1)	2m+1 (n+1)	m+1 (1+m)	2m+3 (2n-(n-3))
İşaret Dağıtım Noktaları	2m+1 (n+1)	m+1 (2+m-1)	2m+3 (2n-(n-3))	m+1 (1+m)

Not: m (n/2)' ye eşit veya daha büyük en küçük tamsayıdır. Tabloda parantez içindeki sayılar m=n/2 durumu için geçerlidir.

3. IBM PC UYUMLU SİSTEM İLE SAYISAL FİLTRE TASARIMI VE GERÇEKLEŞTİRİLMESİ

Ev, büro, endüstriyel ortamlarda yaygın olarak kullanılan IBM uyumlu kişisel bilgisayar sistemleri üzerinde çalışan genel amaçlı bir sayısal filtre tasarlanmış ve gerçekleştirilmiştir. Bilgisayarda yapılan bu gerçekleme üç ana kısımda incelenebilir. Bunlar:

- Dış ortamdan filtre edilecek analog bilginin alınması ve sayısal bilgiye dönüştürülmesi,
- Bu bilginin oluşturulan sayısal filtre yapısına göre bilgisayar ortamında değerlendirilmesi,
- Filtreleme işlemi sonucunda elde edilen sonucun analog bilgiye dönüştürülüp dış ortama gönderilmesi.



Şekil 3.1 IBM PC ile gerçekleştirilen sayısal filtrenin blok diyagramı

Şekil 3.1'de gösterilen blok diyagramda, gerçeklemenin birinci ve üçüncü aşaması donanımla, ikinci aşaması da yazılımla yapılmaktadır. Buna bağlı olarak sayısal filtre gerçeklemesi iki ana kısımda incelenebilir. Bunlar:

- Sayısal Filtre Donanımı,
- Sayısal Filtre Yazılımı.

3.1 SAYISAL FİLTRE DONANIMI

Analog giriş işareti örneklenerek sayısal işarete dönüştürülmesinde ve üzerinde filtreleme işlemi yapılan sayısal bilginin analoga işarete çevrilmesinde Bytronic firmasının IBM PC/XT/AT için geliştirdiği MPIBM4 çok fonksiyonlu giriş/çıkış kartı kullanılmıştır. Kartın özellikleri:

1. 24 adet ayarlanabilir sayısal giriş/çıkış hattı sağlayan 8255 programlanabilir paralel arabirim (PPI),
2. 3 adet 16 bitlik sayıcı/zamanlayıcı içeren 8253,
3. 16 kanallı 12 bit Analogdan Sayısala Dönüştürücü (ADC),
4. 12 bit Sayısaldan Analoga Dönüştürücü (DAC),
5. Kesme erişimi.

Kartın üzerinde bulunan anahtarlar kullanılarak kartın giriş/çıkış adresi kullanılan bilgisayar sistemine uygun şekilde seçilebilir. Yukarıda belirtilen birimlere ve bunların iç yazmaçlarına bağıl adresleme (kartın taban adresi + offset) yoluyla erişilebilir.

3.1.1 12-bit 16 kanallı Analogdan Sayısala Dönüştürücü (ADC):

MPIBM4 kartında analogdan 12-bitlik sayısala dönüşüm için, "başarım yaklaşımı" yöntemini kullanan ADS774 ve analog çoğullayıcı olarak da MPC800 tümdevresi kullanılmıştır. ADS774 dönüştürücüsü dahili örnekleme, tutma devreleri ve mikroişlemci arabirimine sahiptir. Tam çalışma sıcaklığında 12-bitlik bir dönüşüm için 8 μs zaman almaktadır. Ayrıca bir programlanabilir yükselteç PGA202 tümdevresi devresi ile çıkış 1, 10, 100, 1000 oranında kuvvetlendirilebilir. Kartın üzerinde yer alan anahtarlar kullanılarak aşağıda belirtilen giriş işareti aralıklarında örnekler alınabilir:

- 0-10V tek kutuplu,
- +/- 5V iki kutuplu,
- 0-20V tek kutuplu,
- +/- 10V iki kutuplu.

Analogdan sayısala dönüştürücü, çoğullayıcı ve programlanabilir kazanç kontrol tümdevresine aşağıdaki giriş/çıkış adresleri kullanılarak doğrudan erişim yapılabilir:

Taban Adres + 16 : 12 bit dönüşümü başlatma ve çoğullayıcı kanalı seçme,
Taban Adres + 17 : 8 bit dönüşümü başlatma ve çoğullayıcı kanalı seçme,
Taban Adres + 18 : En ağırlıklı bitleri (MSB) okuma,
Taban Adres + 19 : En az ağırlıklı bitleri (LSB) okuma,
Taban Adres + 20 : Dönüşümün tamamlanıp tamamlanmadığını kontrol etme,
Taban Adres + 24 : Çoğullayıcı kanalı seçme ve kazancını ayarlama.

3.1.2 12-bit 2 kanallı Sayısaldan Analoga Dönüştürücü (DAC):

MPIMB4 kartında sayısaldan analoga dönüşüm için yüksek performanslı 12-bitlik DAC7081 tümdevresi kullanılmıştır. DAC7801 akım çıkışlı olup, bu çıkış 2 adet TLE2022 enstrümantasyon işlemsel yükselteci ile gerilime dönüştürülür ve tamponlanır. Kartın üzerinde bulunan anahtar sayesinde sayısaldan analoga dönüştürücünün iki kutuplu veya tek kutuplu çıkış vermesi sağlanabilir. Aşağıda verilen giriş/çıkış adresleri kullanılarak kart üzerindeki DAC' lere doğrudan erişim yapılabilir:

Taban Adres + 8 : DAC A' ya en az ağırlıklı bitleri gönderme ,
Taban Adres + 9 : DAC A' ya en ağırlıklı bitleri gönderme,
Taban Adres + 10 : DAC B' ya en az ağırlıklı bitleri gönderme ,
Taban Adres + 11 : DAC B' ya en ağırlıklı bitleri gönderme,
Taban Adres + 12 : Her iki DAC yazmaçlarını güncelleme.

3.2 SAYISAL FİLTRE YAZILIMI

Sayısal filtre yazılımı oluşturulurken filtre yapılarından 1D kullanılmıştır. Tablo 2.2.1' e bakıldığında 1D yapısının zaman-gecikme elemanı bakımından 3D ve 4D yapısına göre gerçekleme yapılırken daha avantajlı olduğu görülmektedir. Ayrıca toplama noktaları dikkate alındığında da benzer avantaj 2D ve 4D yapısına göre görülmektedir. Tüm bu avantajlarından dolayı gerçekleme yapılırken 1D yapısı tercih edilmiştir. Sayısal filtre (1D yapısı) yapısına ait program, analog bilginin alınıp sayısala çevrilmesi, bu sayısal bilgi üzerinde işlem yapılıp sonucun tekrar analoga çevrilmesi işlemi gerçek zamanda yapıldığı için makina dilinde yazılmıştır.

Sayısal filtre yapısı oluşturulurken iki ayrı makina dili programı yazılmıştır. Bu programlardan birincisinde elde edilen değerler (filtre katsayıları, analog sinyalden alınan örnek değeri) mikroişlemcinin yazmaçlarına aktarılır ve bu değerler üzerindeki işlemler mikroişlemci tarafından yapılır. Bilindiği gibi istenen filtre özelliklerini sağlayan H(z) transfer fonksiyonunun katsayıları reel değerler alabilir. Bu reel değerleri mikroişlemcinin yazmaçlarına doğrudan aktarmak ve üzerinde işlem yapmak mümkün değildir. Bu değerlerin uygun bir yol izlenerek programa aktarılması gerekmektedir.

Bugün 8086 temelli işlemcilerde olduğu gibi birçok mikroişlemcide ikiye tümleyen sayı sistemi (two's complement number system (2_{cns})) kullanılmaktadır. 8086 temelli mikroişlemcilerin 16-bitlik yazmaç mimarisi olduğu düşünülürse tüm sayıları aşağıdaki şekilde göstermek mümkündür:

$$N = (SM_{14}M_{13}\cdots M_1M_0.)_{2_{cns}} \tag{3.1}$$

Buradan;

$$-2^{15} \le N \le 2^{15} - 1 \tag{3.2}$$

arasındadır. Eğer tüm sayıları;

$$N = (S.M_{14}M_{13}\cdots M_1M_0)_{2_{cns}} \tag{3.3}$$

şeklinde ölçeklendirirsek;

$$-1 \le N \le 1-2^{-15} \tag{3.4}$$

yazılabilir. Devam edilerek;

$$1 \le |N| \le 2 \tag{3.5}$$

sonucu elde edilir. Bu şekilde bir gösterim mümkün değildir. Çünkü direkt filtre yapısı veya çapraz-bağlı yapılar oluşturulurken dikkat edilmesi gereken önemli bir nokta, sayısal filtre programının doğru çalışması için katsayıların sınırlandırılması gerekliliğidir. Bu sınırlama;

$$\begin{aligned} 0 \le |a_0, a_2, b_2, g_1, g_2| < 1 \\ 0 \le |a_1, b_1, g_3, g_4| \le 2 \end{aligned} \tag{3.6}$$

biçimindedir. Bu sınırlama yüzünden tüm katsayılar kendi değerlerinin yarısı alınarak saklanır. Yani;

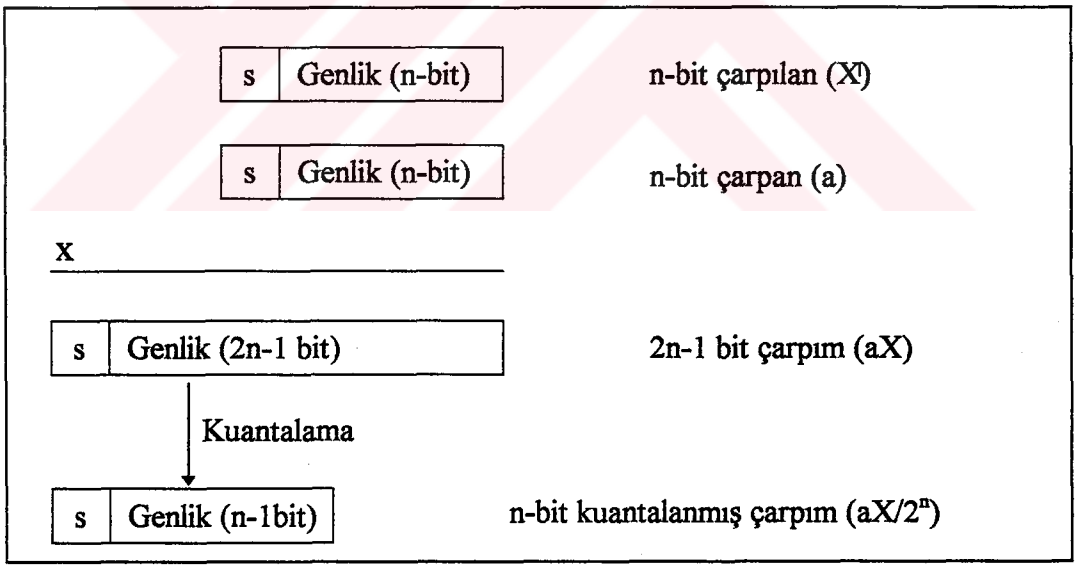
$$SaklanacakDeğer = \left[GerçekDeğer * 2^{n-2} + 0.5\right]$$

olacaktır. 16-bitlik işlem yapıldığı düşünülürse burada $2^{n-2} = 2^{14}$ olacaktır. Yazmaçlarda saklanacak bu değer, yapılan bu değişikliği kompanze etmek için daha sonra iki ile çarpılır. Buradaki [x] sembolü x' den küçük en büyük tamsayı anlamına gelmektedir.

Örnek: a_0=0.693441 gibi bir filtre katsayısının 8086 içerisine aktarıldığını ele alalım. Bu durumda aktarılacak değer;

$$\begin{aligned} Saklanacak\ Değer &= \left[a_0 * 2^{14} + 0.5\right] \\ &= \left[0.693441 * 2^{14} + 0.5\right] \\ &= \left[11361.83734\right] \\ &= \left[11361\right] \end{aligned}$$

olacaktır. İkili sayı sisteminde çarpma işlemini ele alalım: Şekil 3.2' de bu işlem görülmektedir. Burada n-bit çarpılan (bir giriş sinyali) n-bitlik bir çarpanla (bir filtre katsayısı) çarpılmakta ve işlem sonucunda 2n-bitlik bir sonuç elde edilmektedir. Elde edilen bu sonuç daha sonraki bir çarpmada çarpılan olarak kullanılabilir. Bu nedenle bu değerin n-bitle kuantalanması gerekir.



Şekil 3.2. İkili sayı sisteminde çarpma

Örnek olarak bir sinyal değeri X ve bir filtre katsayısı da a ise çarpımları;

$$a \times X = aX \tag{3.7}$$

Çarpım kuantalanırsa;

$$Q[aX] = \left[\frac{aX}{2^n}\right] \tag{3.8}$$

olur. Gerçekte (3.1) ve (3.2)' den görüldüğü gibi bilgisayar donanımı tamsayı işlem yapabildiğinden bu çarpım;

$$(X*2^{n-1}) \times (a*2^{n-1}) = aX*2^{2n-2} \tag{3.9}$$

şeklinde olacaktır. Kuantalama yapılırsa;

$$Q[aX] = \left[\frac{aX*2^{2n-2}}{2^n}\right] = \left[aX*2^{n-2}\right] \tag{3.10}$$

olur. Ardından elde edilen bu çarpım iki ile çarpılırsa (sola bir öteleme yapılırsa) sonuçta elde edilecek terim;

$$Q[aX] = \left[aX*2^{n-1}\right] \tag{3.11}$$

olacaktır. Daha önceden belirtildiği gibi sayısal filtre programının düzgün bir şekilde çalışabilmesi için filtre katsayılarının değerlerinin yarısı (half-values) alınarak saklanması gerekmektedir. Buna göre mikroişlemcinin aşağıdaki işlemleri gerçeklemesi gerekmektedir:

1. Filtre katsayıları yarı değerleri alınarak yazmaçlara yüklenmelidir.

$$AX(\text{yazmaç}) = a*2^{n-2}$$

2. Bu değer X giriş sinyali ile çarpılmalıdır.

$$\begin{aligned} DX,AX &= (a*2^{n-2}) \times (X*2^{n-1}) \\ &= (aX*2^{2n-3}) \\ &= (aX/4)*2^{2n-1} \end{aligned}$$

3. DX,AX yazmaçlarında olan bu değer 16-bitle kuantalanmalı ve en son elde edilen sonuç 4 ile çarpılmalıdır.

$$\begin{aligned} DX &= \left[\frac{(aX/4)*2^{n-1}}{2^n}\right]*4 \\ &= \left[(aX/4)*2^{n-1}\right]*4 \\ &= \left[aX*2^{n-1}\right] \end{aligned}$$

Örnek: X=0.7562867 ve a=0.4383164 olsun. Filtre katsayıları yarı değerleri alınarak saklandığı için;

$$\left[a*2^{n-2}\right] = \left[0.4383164*2^{14}\right] = 7181$$

X giriş sinyali de 1-bit işaret biti alınarak 16-bitlik gösterimle;

$$\left[X*2^{n-1}\right] = \left[.7562867*2^{15}\right] = 24782$$

olur. Çarpım yapılıp 16-bitle de kuantalama yapılırsa;

$$\left[\frac{aX*2^{29}}{2^{16}}\right] = \left[aX*2^{13}\right] = \left[2715,4471\right] = 2715$$

Sonuç olarak elde edilen bu değer 4 ile çarpılırsa;

$$[aX*2^{13}]*4 = 2715*4 = 10860$$

Elde edilen bu sonuçtan hareketle;

$$aX = \frac{10860}{2^{15}} = 0.3314209$$

bulunur. Bu çarpımın gerçek sonucu 0.3314929' dur. Görüldüğü gibi gerçek sonuçla elde edilen sonuç arasında küçük olmasına rağmen bir farklılık bulunmaktadır. Katsayıların ve çarpımların kuantalanması sonucunda meydana gelen bu hatalar filtrenin frekans yanıtında değişikliklere neden olabilir.

Sayısal filtre yapısı oluşturmak için yazılan ikinci programda elde edilen değerler (filtre katsayıları, analog sinyalden alınan örnek değeri) mikroişlemcinin içerisinde tümleşik olarak (built-in) bulunan nümerik işlemciye aktarılır ve bu değerler üzerindeki işlemler nümerik işlemci tarafından yapılır. Bilindiği gibi bugün kullanılmakta olan IBM uyumlu kişisel bilgisayarların tümünde 8087 komut temelli gelişmiş nümerik işlemcisi bulunmaktadır. Bu işlemci sahip olduğu donanım özellikleri sayesinde karmaşık matematiksel işlemlerini kolaylıkla ve hızlı yapabilmektedir. Kayan Noktalı Aritmetik (Floating Point Arithmetic) işlem yapabilme özelliğine sahip olan bu işlemci sayesinde elde edilen sayısal filtrenin katsayıları üzerinde herhangi bir işlem yapılmaksızın bu değerler oldukları gibi alınarak bu işlemciye aktarılır ve üzerinde gerekli olan aritmetik işlemler kolayca yapılabilir. Böylece katsayıların ve çarpımların kuantalanmasıyla meydana gelecek hatalar ortadan kaldırılmış olur. Bu sayede sayısal filtrenin frekans yanıtında oluşacak hatalar minimuma indirgenmiş olur. Sağladığı bu avantajlardan dolayı sayısal filtrelerin gerçeklenmesinde ikinci program tercih edilmiştir. Ek1 bölümünde 80387 işlemcinin kısa donanım özellikleri ve sayısal filtre programında kullanılan komutlarının açıklaması verilmiştir.

3.2.1 Sayısal Filtre Yazılımının Bölümleri

Sayısal filtre gerçeklemesi için oluşturulan yazılım üç alt yazılımdan meydana gelmektedir.

1. Gerçeklenecek sayısal filtre için örnekleme frekansını bulan program,
2. İstenen özellikleri sağlayacak sayısal filtrenin katsayılarını bulan program,
3. Bulunan bu katsayıları kullanarak sayısal filtreyi çalıştıran program.

3.2.1.1 Örnekleme Frekansının Belirlenmesi

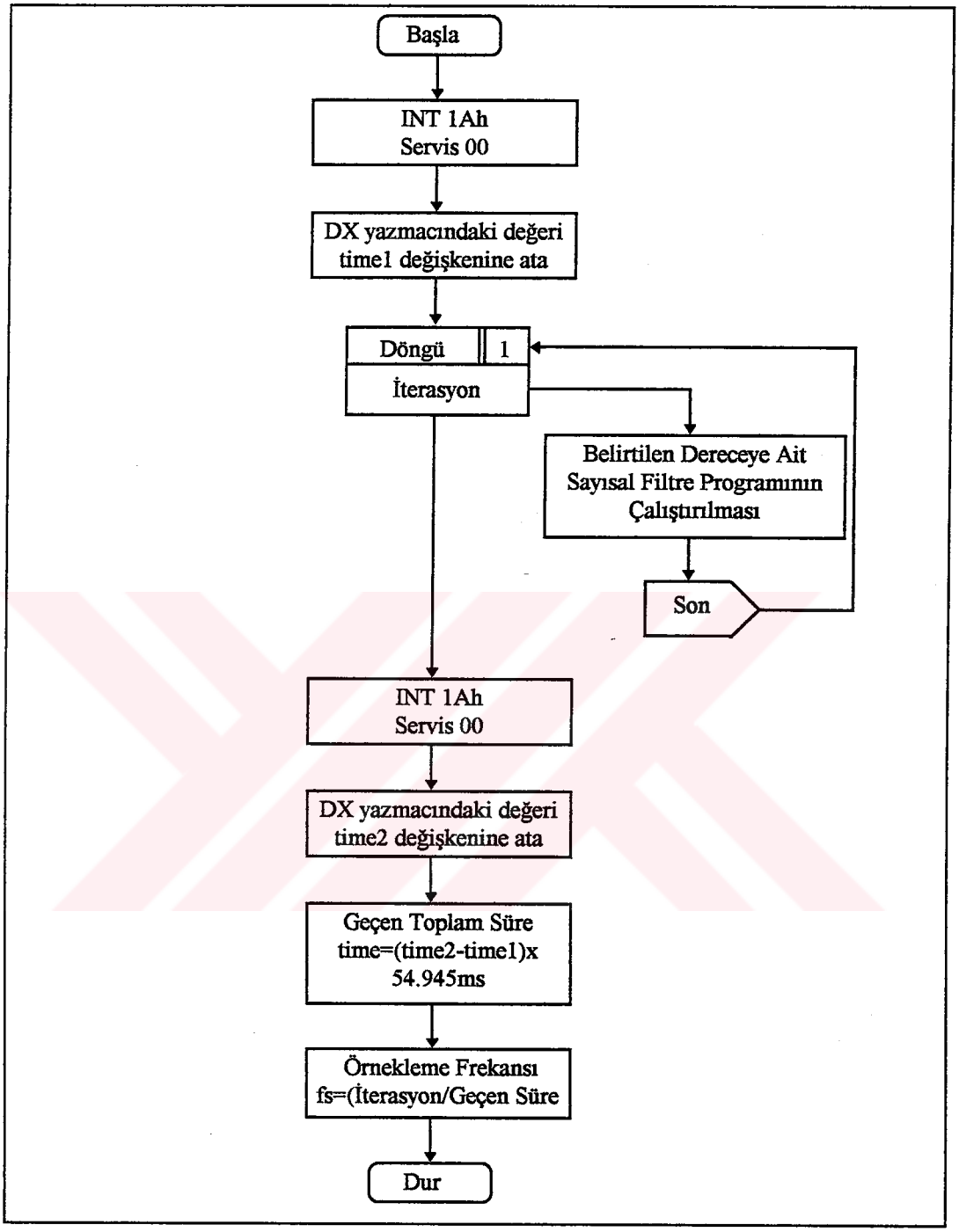
Örnekleme frekansının doğru tespit edilmesi katsayıların bulunmasında oldukça önemlidir. Donanım olarak MPIBM4 giriş/çıkış kartıyla yapılan örneklemede hız sabittir. Bununla beraber alınan bir örnek üzerinde daha sonra sayısal filtre işlemleri yapılıp, elde edilen sonuç analoga çevrilip çıkışa verildikten sonra, giriş sinyalinden ikinci bir örnek alındığı için ek bir süre daha geçmektedir. Buna göre donanım ve yazılım beraber ele alındığında analog sinyalden alınacak iki örnek arasındaki toplam süre;

Toplam Süre = ADC işlemi için gerekli süre + Sayısal filtre işlemleri + DAC işlemi için gerekli süre

olacaktır. Yani kısaca donanımla yapılan işlemler ve yazılımla yapılan işlemlerde harcanan sürelerin toplamı olacaktır. Gerçeklenecek sayısal filtrenin örnekleme frekansının tespiti için oluşturulan yazılım, filtrenin çalıştırılmasında kullanılan yazılımla aynıdır. Sadece örnekleme frekansının tespiti program belirli bir iterasyon verilerek çalıştırılır ve bu arada geçen süre değerlendirilerek örnekleme frekansı bulunur. Sayısal filtrenin derecesinin artması yapılacak işlem sayısını fazlalaştırdığı için yazılımın yürütülmesi için gereken süre artar. Bu nedenle örnekleme hızı her derece için farklı olur.

Örnekleme frekansının tespitinde bilgisayarın sistem zamanlayısından (system-timer) yararlanılmıştır. Bilgisayarda (PC' de) sistem zamanlayıcısı her 55ms (54.945ms)' de (saniyede 18.2 defa) yüksek doğrululuklu bir tik (tick) üretir. Üretilen bu tiklerin sayısı 0:46Ch adresinde 4-byte (double word) olarak saklanır. Sistem zamanlayıcısı aynı zamanda (her 55ms' de bir defa) ROM BIOS' dan INT 08'i aktif hale getirir. INT 08 0:46Ch adresinde saklı olan bu tiklerin sayısını 1 artırır. 0:46Ch adresinde saklı olan bu tik'lerin sayısı kullanılarak saat, dakika, saniye ve salise bilgisi oluşturulur. ROM BIOS' dan çağrılan INT 1Ah, servis 00h 0:46Ch adresindeki 4-byte' lık değeri okur ve buna ek olarak en son sisteme güç verilmesi, sistemin reset edilmesi veya sistem zamanlayıcısının okunması/set edilmesi işlemleri üzerinden 24 saat geçip geçmediği hakkında bilgi verir. Örnekleme frekansı bulunurken INT 1Ah kullanılılır. Bu amaçla yazılan makina dili programının girişinde bu kesme yürütülür ve DX yazmacında bulunan 0:46Ch adresindeki değerin düşük anlamlı 2 baytı bir değişkene atanır. Ardından gerçeklenmesi istenen sayısal filtreye ait program belirli bir iterasyon verilerek çalıştırılır. İterasyon bitiminde tekrar ROM BIOS' dan INT 1Ah kesmesi yürütülür ve ardından DX yazmacında bulunan 0:46Ch adresindeki değerin düşük anlamlı 2 baytı ikinci bir değişkene atanır. Bu iki değer arasındaki fark 54.945ms ile çarpılıp ardından da elde edilen sonuç iterasyon sayısına bölünürse analog sinyalden alınan iki örnek arası süre bulunmuş olur. Bu sürenin tersi de doğal olarak örnekleme frekansını verecektir. Örnekleme frekansını bulan program Turbo Pascal 7.0 ortamında makina dilinde yazılmıştır.

INT 1Ah		
Giriş:	AH	00h
Dönüş:	CX	0:46Ch adresindeki 4 - byte' lik değerin yüksek değerli 2 - byte' ı
	DX	0:46Ch adresindeki 4 - byte' lik değerin düşük değerli 2 - byte' ı
	AL	24 saat aşılmadıysa 0 aksi halde 1



Şekil 3.3 Örnekleme frekansını bulan programın akış diyagramı

Yukarıdaki şekilde örnekleme frekansının belirlenmesi için yazılan programın akış diyagramı görülmektedir.

3.2.1.2 Sayısal Filtre Katsayılarının Bulunması

Sayısal filtrenin katsayıları bulunurken MATLAB 4.0 programından yararlanılmıştır. Bu programda işaret işlemeye yönelik hazır program paketi (Signal Processing Toolbox) bulunmaktadır. Bu paket program içerisinde IIR filtre tasarımında kullanılmak üzere yazılmış Butterworth tipi filtreler için "butter" , Chebyshev I tipi filtreler için "cheby1", Chebyshev II tipi filtreler için de "cheby2" adlı Matlab programları bulunmaktadır. Eğer Matlab programı içerisinde komut satırından;

$$[B,A] = \text{butter}(N, Wn, \text{ftype})$$

yazılırsa, verilen N, Wn ve filtrenin tipini gösteren ftype değişkeninin değerine göre butterworth tipi sayısal filtrenin filtrenin katsayıları (N+1) uzunluğundaki B ve A vektörleri olarak bulunacaktır.

$$[B,A] = \text{butter}(N, Wn)$$
$$[B,A] = \text{butter}(N, Wn, \text{'high'})$$
$$[B,A] = \text{butter}(N, Wn, \text{'stop'})$$

Yukarıdaki birinci komutla butterworth alçak geçiren, ikinci komutla butterworth yüksek geçiren ve üçüncü komutla da butterworth band söndüren sayısal filtrenin katsayılarını bulmak mümkündür. Alçak geçiren filtre için kullanılan komut band geçiren filtre için de kullanılabilir. Ancak band geçiren filtre için Wn = [W1 W2] (W1<Wn<W2) şeklinde iki elemandan oluşacağı için program bu farklılığı gözönüne alarak alçak geçiren veya band geçiren filtre için katsayıları bulacaktır. N.dereceden band geçiren ve band söndüren filtreler için 2N+1 tane katsayı bulunacaktır.

Burada dikkat edilmesi gereken nokta Wn kesim frekansının örnekleme frekansının yarısına göre normalize girilmesi gerekliliğidir. Buna göre Wn;

$$0.0 < Wn < 1.0 \qquad (3.12)$$

olacaktır. Benzer şekilde Chebyshev I ve Chebyshev II tipi sayısal filtreler için de;

$$[B,A] = \text{cheby1}(N,R,Wn,\text{ftype})$$
$$[B,A] = \text{cheby2}(N,R,Wn,\text{ftype})$$

komutları yazılarak sayısal filtrenin katsayıları bulunur. Burada R sırasıyla geçirme ve söndürme bandındaki dalgalanmayı (ripple) göstermektedir. Gerek butterworth ve gerekse chebyshev tipi sayısal filtrelerin katsayılarını bulmak için yazılan bu programlarda izlenen yol (sayısal filtre tasarım protokolü) şekil 3.4' de blok diyagramla gösterilmektedir.

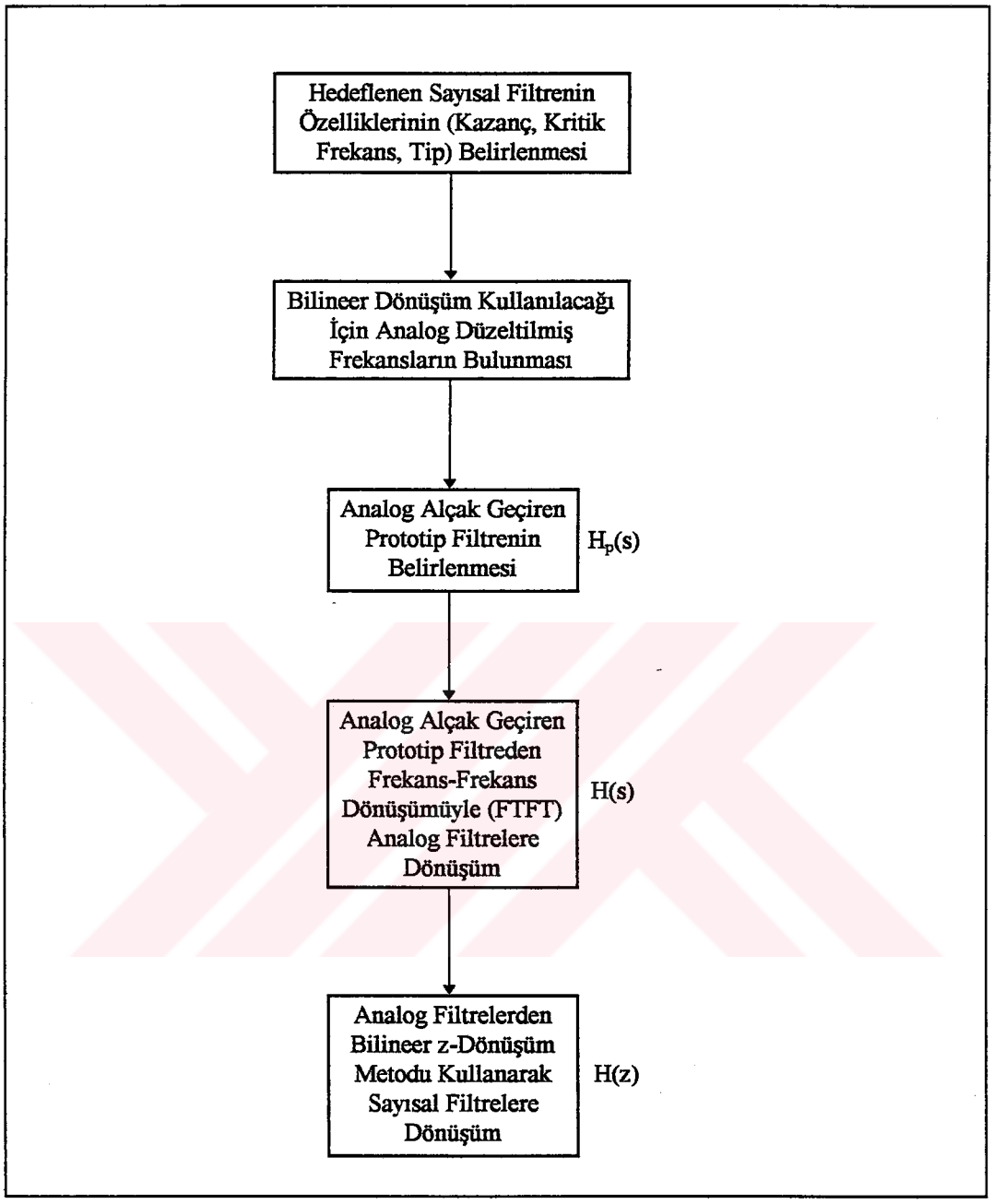
Eğer direkt gerçekleme yapılacaksa bulunan bu katsayılar doğrudan kullanılabilir. İkinci-derece modüllerle paralel gerçekleme yapılmak isteniyorsa ilk olarak elde edilen H(z) transfer fonksiyonu kısmı kesirlere açılır. (Partial-fraction expansion) Bu amaçla Matlab programının "residuez" komutu kullanılmaktadır. Komutun genel formu:

$$[r,p,k] = \text{residuez}(B,A)$$

biçiminde olup B(z)/A(z)' in kısmi kesirlere açılımında rezidüleri, kutupları ve direkt terimi bulmaktadır. Açılımın genel biçimi:

$$\frac{B(z)}{A(z)} = \frac{r(1)}{1-p(1)z^{-1}} + \frac{r(2)}{1-p(2)z^{-1}} + \cdots + \frac{r(n)}{1-p(n)z^{-1}} + k(1) + k(2)z^{-1}\cdots \qquad (3.13)$$

B: Pay polinomu katsayıları, **A:** Payda polinomu katsayıları, **r:** Rezidüler, **p:** Kutuplar **k:** Direkt terimler
biçimindedir.



Şekil 3.4 Analog Prototip'den hareketle sayısal filtre tasarımının blok diyagramı

Sayısal filtrenin transfer fonksiyonu H(z) denklem (3.13)' deki gibi kısmi kesirlere açıldığında kompleks rezidü ve kutuplara sahip terimlerle karşılaşılır. Bu kompleks rezidü ve kutuplar eşlenik çiftler halindedir. Yani açılımda eğer kompleks rezidü ve kutba sahip bir terim varsa, bu terimin sahip olduğu rezidü ve kutbun eşleniği rezidü ve kutba sahip ikinci bir terim vardır. Sayısal uygulamalarda kompleks sayılarla işlem, gerekli donanım

ihtiyacı daha fazla olduğu için tercih edilmez. Bu nedenle kompleks eşlenik rezidü ve kutuplara sahip ifadeler kendi aralarında işleme tabi tutularak ikinci derece modül yapısı elde edilebilir. Böylece hem kompleks sayılarda işlem yapma zorunluluğu ortadan kaldırılmış, hem de ikinci derece modül yapısı oluşturulmuş olur. (3.13) ifadesindeki ilk iki terimi ele alalım. Bu iki terimde yer alan r(1) r(2)' nin, p(1) de p(2)' nin eşleniği olsun. Eğer,

$$\begin{aligned} r(1) &= a(1) + ib(1) \qquad & r(2) &= r^*(1) = a(1) - ib(1) \\ p(1) &= c(1) + id(1) \qquad & p(2) &= p^*(1) = c(1) - id(1) \end{aligned} \tag{3.14}$$

şeklinde gösterilirse;

$$\begin{matrix} \dfrac{r(1)}{1 - p(1)z^{-1}} \\ (1 - p^*(1)z^{-1}) \end{matrix} + \begin{matrix} \dfrac{r^*(1)}{1 - p^*(1)z^{-1}} \\ (1 - p(1)z^{-1}) \end{matrix} \tag{3.15}$$

yazılıp r(1), $r^*(1)$, p(1) ve $p^*(1)$' in değerleri yerlerine konur ve gerekli işlemsel yapılırsa sonuç olarak;

$$\frac{2a(1) - \left[2a(1)c(1) + 2b(1)d(1)\right]z^{-1}}{1 - 2c(1)z^{-1} + \left[c^2(1) + d^2(1)\right]z^{-2}} \tag{3.16}$$

elde edilir. Görüldüğü gibi elde edilen ifade;

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}, \quad b_0 = 1 \tag{3.17}$$

biçiminde verilen ikinci derece modülün transfer fonksiyonun a_2=0 olması durumunda aynısıdır. İki ifade karşılaştırıldığında;

$$
\begin{aligned}
a_0 &= 2a(1) \\
a_1 &= -\left[2a(1)c(1) + 2b(1)d(1)\right] \\
a_2 &= 0 \\
b_0 &= 1 \\
b_1 &= -2c(1) \\
b_2 &= c^2(1) + d^2(1)
\end{aligned}
\tag{3.18}
$$

olduğu görülebilir. Aynı şekilde açılımda yer alan ve birbirinin eşleniği rezidü ve kutuplara sahip olan terimler için de benzer ifade elde edilebilir. Böylece bu terimlerden ikinci derece modüller oluşturulur. Dikkat edilirse ikinci derece modüllerin katsayıları reeldir.

Eğer H(z) sayısal filtre transfer fonksiyonunu kısmı kesirlere açılımda k.terim reel rezidü ve kutba sahipse bu terim;

$$
\frac{r(k)}{1 - p(k)z^{-1}} \tag{3.19}
$$

biçiminde olacaktır. Görüldüğü gibi bu ifade de $a_1=a_2=b_2= 0$ alınmak süretiyle ikinci derece modül yapısına getirilebilir. Burada;

$$
\begin{aligned}
a_0 &= r(k) \\
b_1 &= -p(k)
\end{aligned}
\tag{3.20}
$$

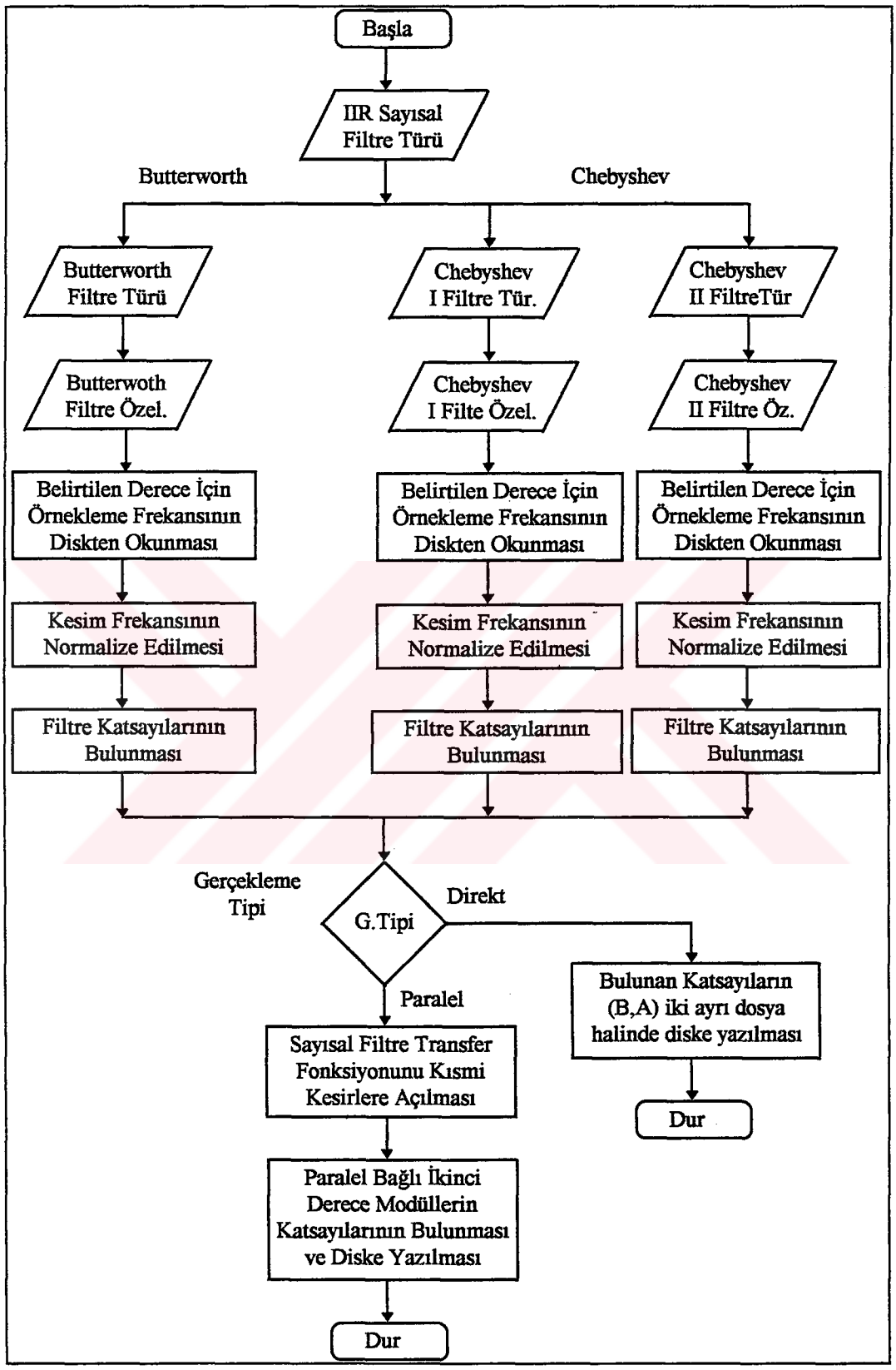
olacaktır. Böylece H(z) transfer fonksiyonu bir direkt terim ve ikinci-derece modüllerin toplamı biçiminde ifade edilebilir ki; bu da transfer fonksiyonun şekil 2.2.7' de görüldüğü gibi ikinci-derece modüllerle paralel gerçeklemesidir.

Şekil 3.5' de sayısal filtrelerin direkt veya paralel gerçeklenmesi için yazılan Matlab programı için akış diyagramı görülmektedir. Görüldüğü gibi ilk olarak IIR filtre türü belirtilir. Ardından bu filtrenin alçak geçiren, yüksek geçiren, band geçiren veya band söndüren filtre tiplerinden hangisi olduğu belirlenir. Bu aşamadan sonra filtrenin

istenen özellikleri girilir. Eğer filtre türü Butterworth ise bu özellikler derece ve kesim frekansı şeklindedir. Chebyshev türü filtrelerde ise bu özelliklere ek olarak Chebyshev I türü filtrelerde geçirme bandında, Chebyshev II türü filtrelerde ise söndürme bandındaki dalgalanma verilir. Daha sonra verilen derece için örnekleme frekansı diskten okunur. İstenen kesim frekansı örnekleme frekansının yarısına bölünerek normalize hale getirilir. Bu aşamadan sonra daha önceden belirtilen Matlab programı komutları yardımıyla katsayılar bulunur.

Eğer direkt gerçekleme yapılacaksa elde edilen katsayılar transfer fonksiyonunun pay ve payda (B ve A) katsayıları olmak üzere iki ayrı dosyaya yazılır.

İkinci derece modüllerle paralel gerçekleme yapılacaksa, bu takdirde de direkt terim, eğer reel rezidü ve reel kutba sahip terim varsa (3.20) ifadesinde yer alan a_0 ve b_1 değerleri ve kompleks eşlenik rezidü ve kutba sahip terimlerim kendi aralarında (3.15)' de gösterilen şekilde işleme tabi tutulmasıyla elde edilen ikinci-derece modülün reel katsayıları (a_0, a_1, b_1 ve b_2) üç ayrı dosyaya yazılır.



Şekil 3.5. Katsayıları bulan Matlab programı için akış diyagramı

3.2.1.3 Sayısal Filtre Programının Çalıştırılması

Sayısal filtrelerin direkt ve ikinci derece modüllerle paralel gerçeklemesi yapıldığı için iki ayrı program yazılmıştır. Gerek direkt gerçeklemede ve gerekse ikinci derece modüllerle paralel gerçeklemede daha önce belirtilen avantajlarından dolayı 1D yapısı kullanılmıştır. Katsayıları kuantalamadan işlem yapabilmek için her iki programda da nümerik işlemci komutları kullanılmıştır. Böylece katsayıların kuantalanmasından ve çarpımlardan doğacak hatalar giderilmiştir. Yapısal farklılıktan dolayı direkt ve ikinci derece modüllerle paralel gerçekleme için yazılan programlar ayrı ayrı ele alınmıştır.

3.2.1.3.1 Direkt Gerçekleme

Şekil 2.2.1'e bakıldığında 1D yapısı için, n filtre derecesini göstermek üzere;

$$\begin{aligned} T_1 &= -(b_1 m(k-1) + b2m(k-2) + \cdots + b_n m(k-n)) \\ T_2 &= a_1 m(k-1) + a_2 m(k-2) + \cdots + a_n m(k-n) \end{aligned} \tag{3.20}$$

olarak gösterilirse;

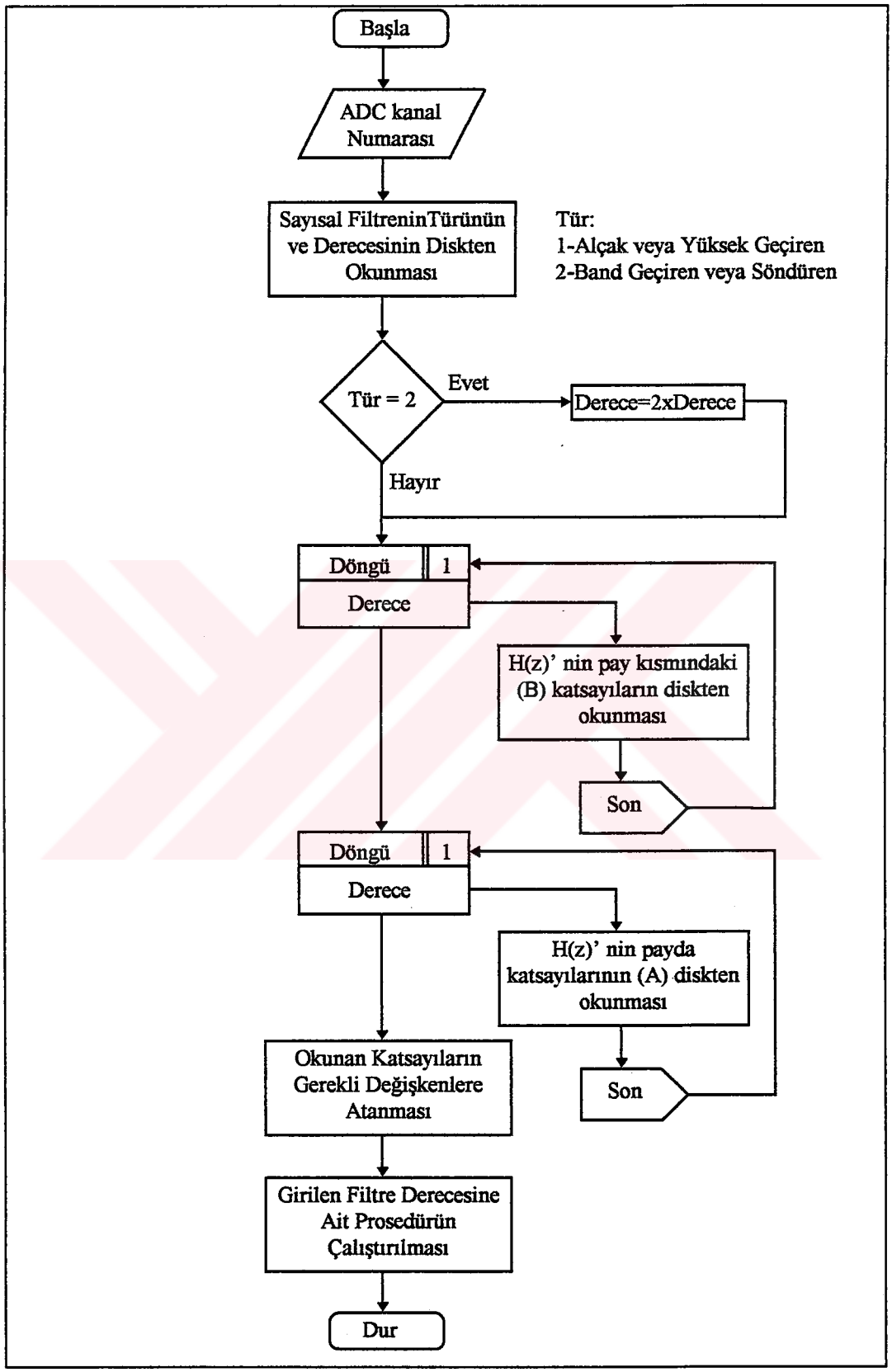
$$\begin{aligned} m(k) &= x(k) + T_1 \\ y(k) &= a_0 m(k) + T_2 \end{aligned} \tag{3.21}$$

yazılabilir. (3.20)' den görüldüğü gibi filtre derecesinin değişmesi T1 ve T2' yi hesaplamak için gerekli işlem sayısını değiştirmektedir.

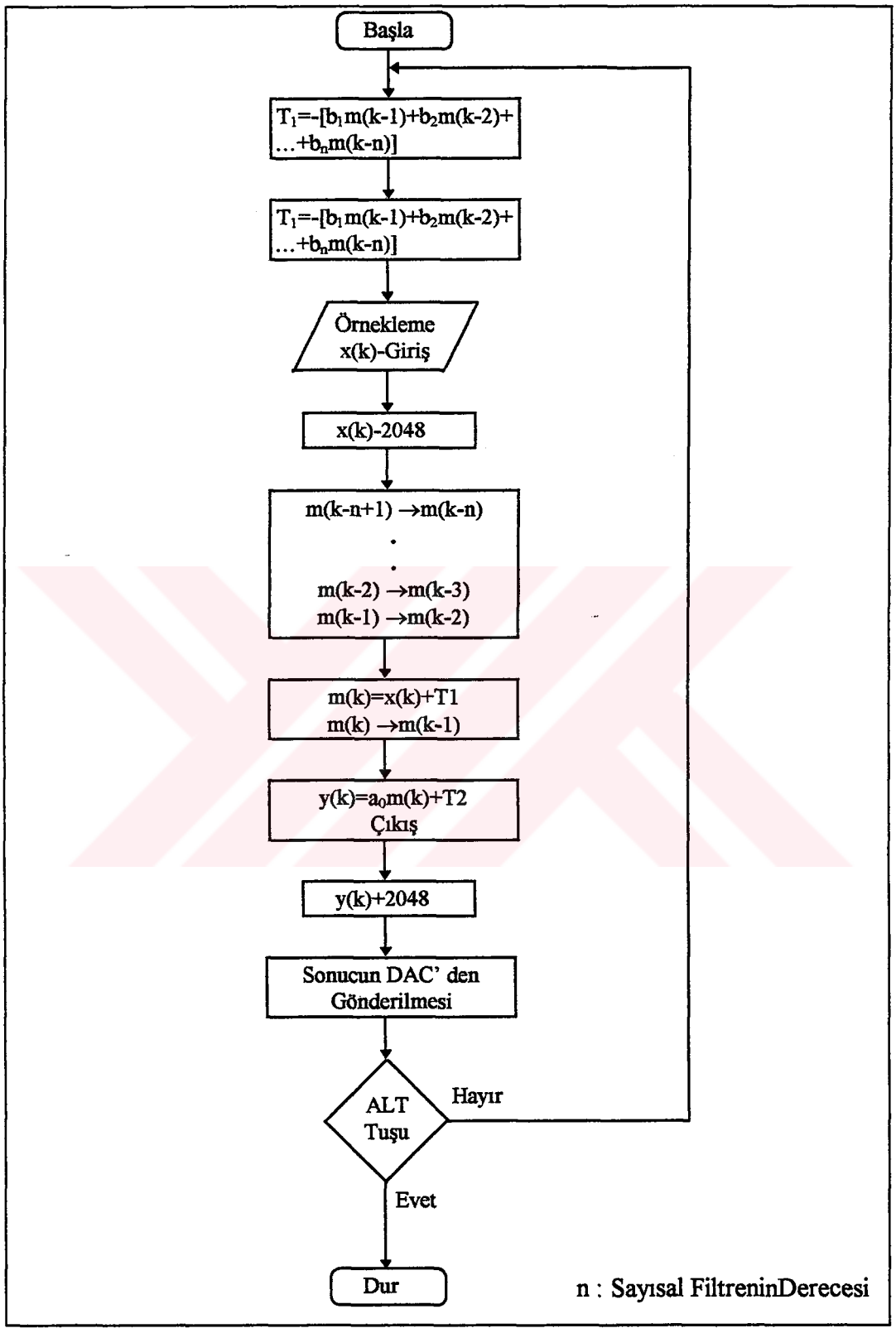
Şekil 3.6' da direkt gerçekleme için sayısal filtre programının akış diyagramı görülmektedir. Diyagramdan da görüldüğü gibi programın başlangıcında ADC' nin hangi kanalından örnek alınacağı belirtilir. Ardından bu kanal numarası program tarafından donanıma iletilir. Daha sonra gerçeklenecek sayısal filtrenin türünün, derecesinin ve katsayılarının kayıtlı olduğu dosyalar okunmak üzere açılır. İlk olarak filtrenin türüne

bakılır. Eğer gerçeklenmek istenen sayısal filtre band geçiren veya band söndüren ise bu takdirde derece iki katına çıkarılır. Çünkü N.dereceden band geçiren veya band söndüren filtre için 2N+1 tane katsayı oluşur. Bu sayı 2N.derece alçak geçiren veya yüksek geçiren filtrenin gerçeklenmesinde oluşacak katsayıların sayısına eşittir. Bulunan derece bilgisi de gözönüne alınarak bir döngü içerisinde katsayılar diskten okunur. Okunan bu katsayılar makina dilinde kullanılacak değişkenlere atanır.

Programda her dereceye ait prosedür yapısı oluşturulduğu için derece bilgisine göre bunlardan biri çağrılır. Bu prosedürler içinde bazı modüller (analogdan sayısala dönüşüm, sayısaldan analoga dönüşüm) ortak olmasına rağmen her prosedürde tekrarlanır. Burada amaç herhangi bir alt programı çağırma için geçecek süreyi yoketmedir. Her derece için oluşturulan bu prosedürlerde izlenen yol aynıdır. Sadece denklem (3.20)'de görüldüğü gibi T1 ve T2 değerleri farklıdır. Şekil 3.7' de her derece için çağrılan bu prosedürler için akış diyagramı görülmektedir.



Şekil 3.6. Direkt gerçekleme için yazılan programın akış diyagramı



Şekil 3.7 Direkt gerçekleme için oluşturulan prosedürlerde program akışı

Şekil 3.7' deki akış diyagramında görüldüğü gibi 1D yapısında ilk olarak T_1 ve T_2 de değerleri bulunur. Ardından giriş sinyalinden 12-bitlik örnek alınır. Bu sıra ters şekilde de olabilir. Yani ilk olarak giriş sinyalinden örnek alınıp, ardından T_1 ve T_2 hesaplanabilir. MPIBM4 kartında kullanılan ADS774 ile bu gerçeklemede -10/+10V arasındaki işaretler örneklenmektedir. Bu aralıktaki işaretler 0-4095 (0-FFFh) aralığındaki sayılara karşılık gelmektedir. Bu gösterimle;

$$-10V \rightarrow 0$$
$$0V \rightarrow 2048$$
$$+10V \rightarrow 4095$$

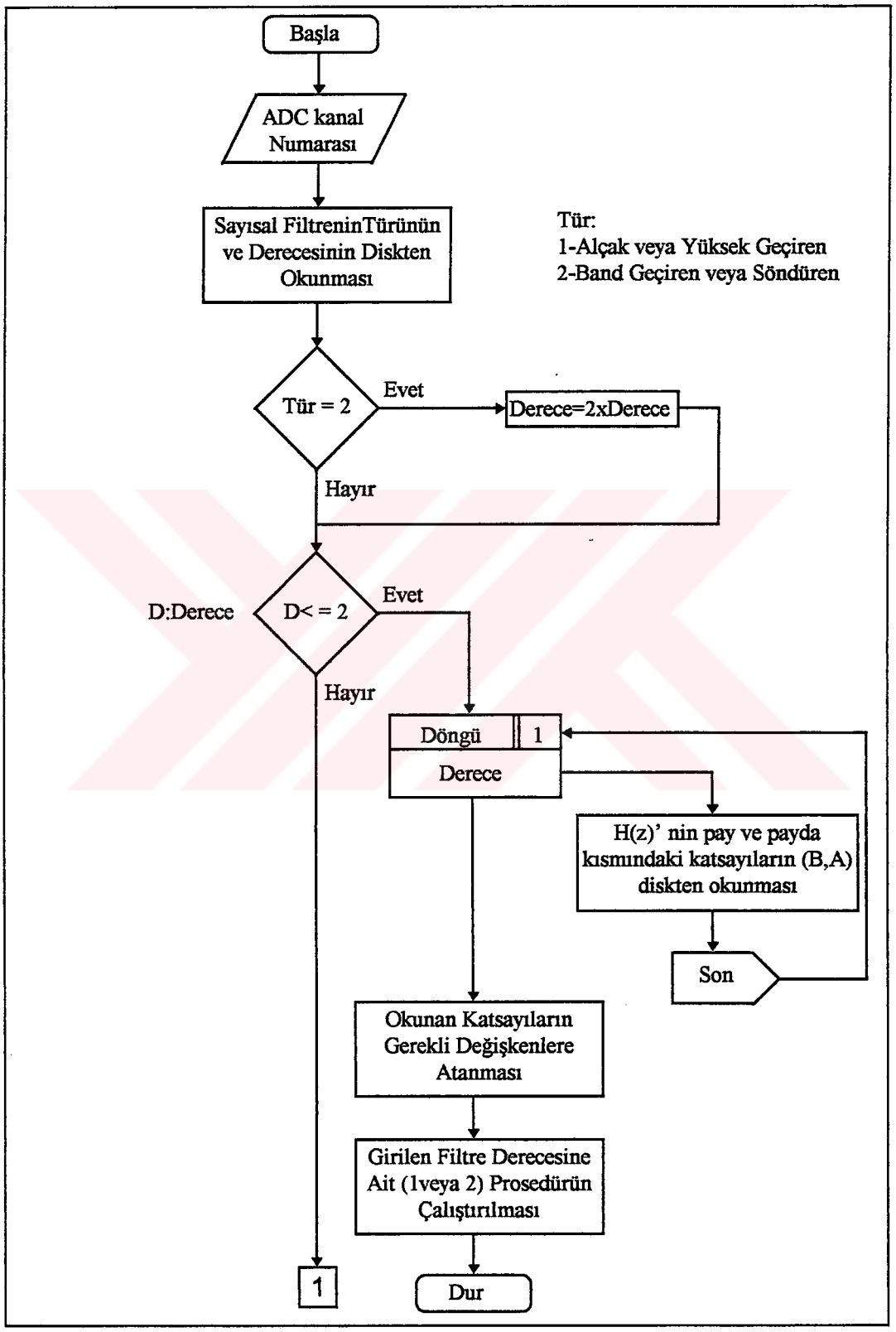
olmaktadır.Bunun yerine giriş sinyalinden alınan örnekleri işaretli hale getirmek için eldeki değerden 2048 sayısı çıkarılır. Böylece;

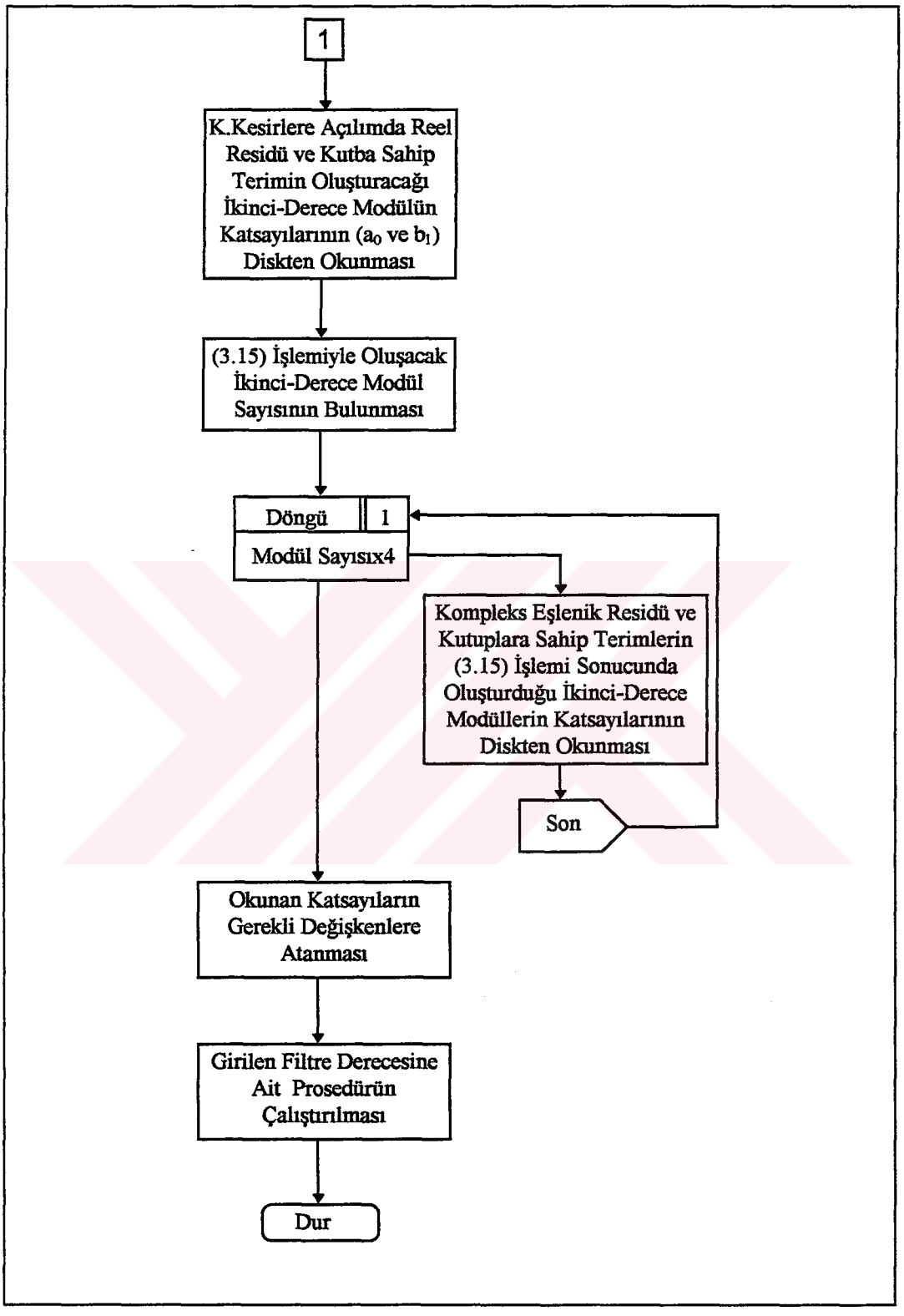
$$-10V \rightarrow -2048$$
$$0V \rightarrow 0$$
$$+10V \rightarrow 2047$$

karşı gelmiş olur. Daha sonra zaman-gecikmelerini (time-delay) sağlamak için bu elemanlar arasında atamalar yapılır. Bunun ardından m(k) terimi oluşturulur ve bu değer bir sonra giriş işaretinden alınacak örnek düşünüldüğünde m(k-1) terimi olacağı için, bu terime atanır. Oluşturulan m(k) terimi a_0 katsayısıyla çarpılır ve T_2 değeriyle toplanırsa çıkış ifadesi olan y(k) terimi elde edilmiş olur. Bunu takiben doğru çıkış alınabilmesi için daha önceden giriş sinyalinden çıkarılan 2048 sayısı çıkış değerine ilave edilir. Elde edilen sonuç 12-bit olarak DAC' den gönderilir.

Program ALT tuşuna basılmadığı sürece çalışmasına devam eder. ALT tuşuna basılıp basılmadığını anlamak için ROM BIOS' dan INT 16h servis 02h kesmesi yürütülür. Bu kesme klavye üzerindeki Insert, Num Lock, Caps Lock, Scrool Lock, Alt, Ctrl ve Shift tuşlarının basılı olup olmadıkları hakkında bilgi verir. Programda sadece ALT tuşunun etkili olması için diğer tuşlar maskelenmiştir.

3.2.1.3.2 İkinci-Derece Modüllerle Paralel Gerçekleme





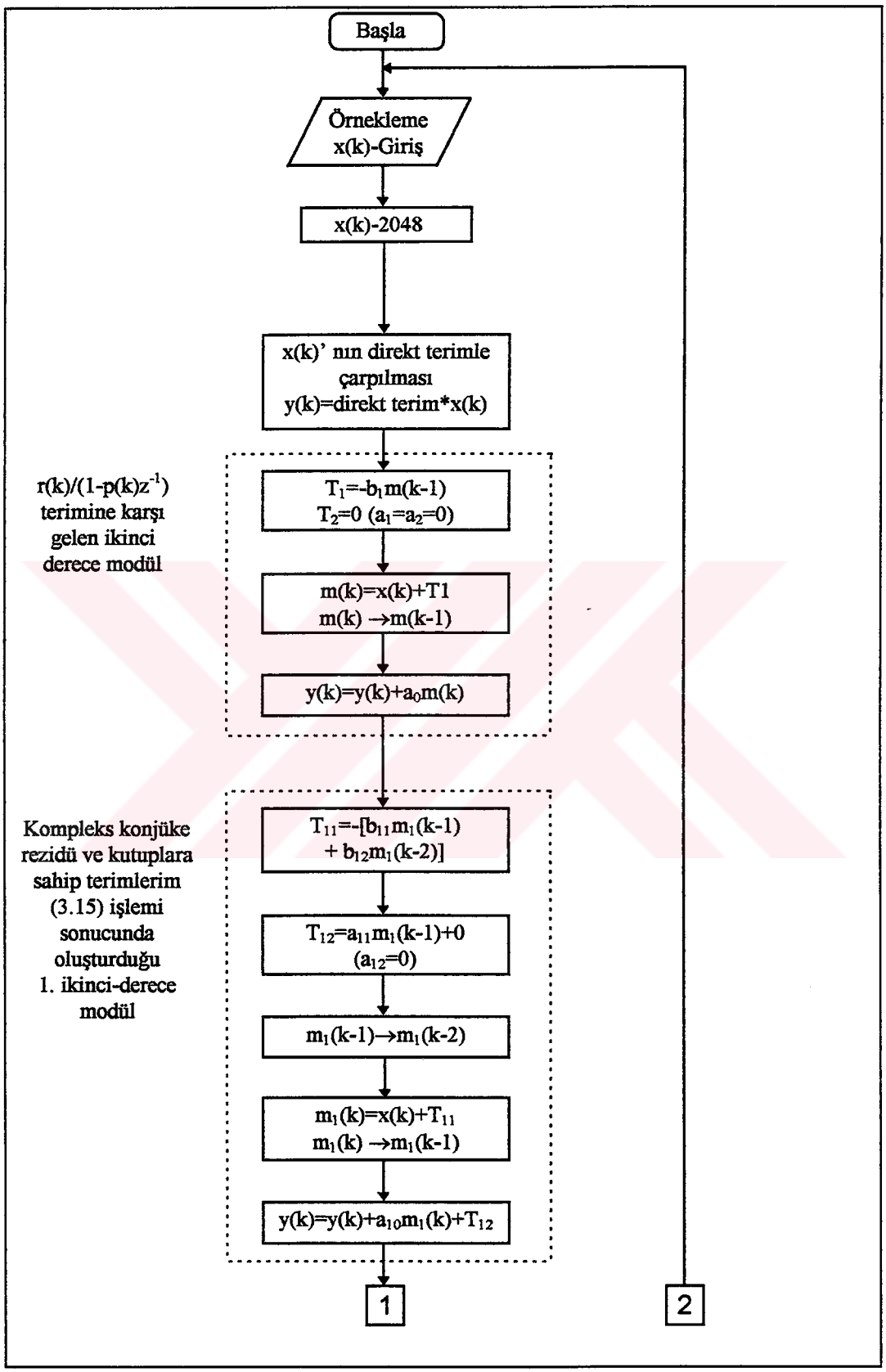
Şekil 3.8 Sayısal filtrenin ikinci-derece modüllerle paralel gerçeklenmesi için yazılan programın akış diyagramı

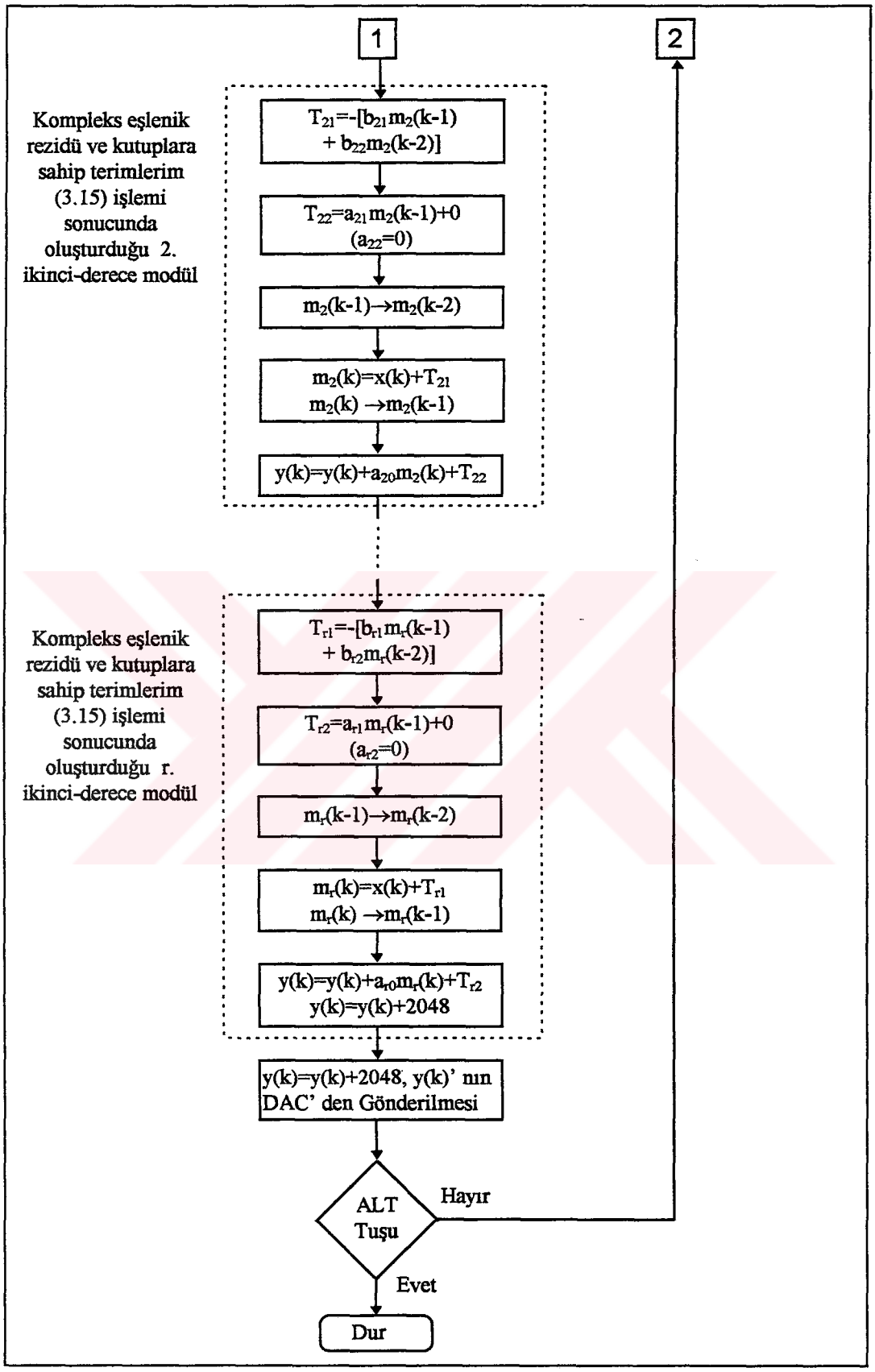
Yukarıdaki akış diyagramından da görüldüğü gibi ilk olarak direkt gerçeklemede olduğu gibi ADC kanal numarası ve ardından sayısal filtrenin türü ve derecesi belirlenmektedir. Yine direkt gerçekleme de olduğu gibi eğer band geçiren veya band söndüren bir filtre gerçekleniyorsa filtre derecesi iki katına çıkarılmaktadır. Eğer gerçeklenecek filtre birinci veya ikinci dereceden ise bu takdirde herhangi bir işlem yapılmaz ve direkt gerçeklemede olduğu gibi katsayılar okunur. Okunan bu katsayılar makina dilinde kullanılacak değişkenlere atanır ve ardından belirtilen dereceye (birinci veya ikinci derece) ait altprogram çalıştırlır.

Eğer derece ikiden büyükse bu sayısal filtrenin transfer fonksiyonu, ikinci derece modüllere ayrılabilir. Bu ayırma işleminin ne şekilde yapılacağı daha önceden belirtilmiştir. Bu amaçla ilk olarak direkt terim okunur. Ardından transfer fonksiyonunun kısmi kesirlere açılımında denklem (3.19)' da belirtilen biçimde bir terim varsa, bu terimin oluşturaçağı ikinci derece modülün katsayıları (a_0 ve b_1, $a_1=a_2=b_2=0$) diskten okunur.

Daha sonra kompleks eşlenik rezidü ve kutuplara sahip terimlerin oluşturacağı ikinci derece modüllerin sayısı belirlenir. Her modül için dört terim (a_0, a_1, b_1, b_2) okunduğu için, toplam sayı modül sayısının dört katı olacaktır. Okunan bu katsayılar gerekli değişkenlere atanır ve belirtilen dereceye ait prosedür çalıştırılır.

Birinci ve ikinci dereceye ait prosedürlerde oluşturulan yapı direkt gerçeklemedeki yapıyla aynıdır. Sayısal filtrenin derecesi ikiden büyükse izlenecek yol farklıdır. Eğer derece tek sayı (3,5 gibi) daha önceden belirtildiği gibi kısmı kesirlere açılımda (3.19) terimi oluşur. Bu terim için de ikinci derece modül yapısı oluşturulur. Derece çift sayı ise böyle bir terim oluşmaz. Aşağıda ikiden büyük derecelere ait altprogramlar için akış şeması görülmektedir.


Başla
Örnekleme
x(k)-Giriş
x(k)-2048
x(k)' nın direkt terimle çarpılması
y(k)=direkt terim*x(k)
r(k)/(1-p(k)z^-1) terimine karşı gelen ikinci derece modül
T1=-b1m(k-1)
T2=0 (a1=a2=0)
m(k)=x(k)+T1
m(k) →m(k-1)
y(k)=y(k)+a0m(k)
Kompleks konjüke rezidü ve kutuplara sahip terimlerim (3.15) işlemi sonucunda oluşturduğu 1. ikinci-derece modül
T11=-[b11m1(k-1) + b12m1(k-2)]
T12=a11m1(k-1)+0 (a12=0)
m1(k-1)→m1(k-2)
m1(k)=x(k)+T11
m1(k) →m1(k-1)
y(k)=y(k)+a10m1(k)+T12
1
2



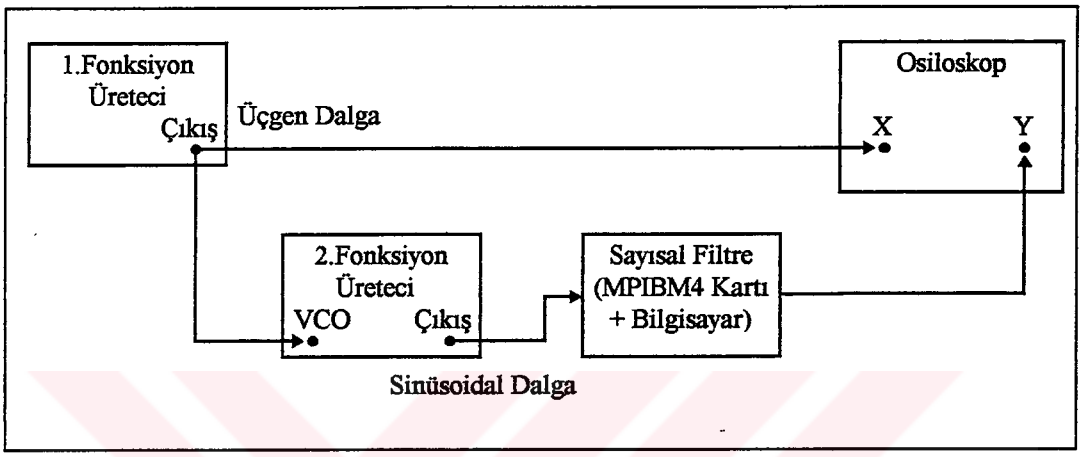
Şekil 3.9 İkiden büyük derecelere ait prosedürler için akış diyagramı

Yukarıdaki akış şemasından da görüldüğü gibi ilk olarak giriş sinyali örneklenmektedir. Bunu takiben okunan direkt terimle alınan örneğin değeri çarpılır ve bir değişkende saklanır. Eğer gerçeklenecek filtrenin derecesi tek sayı ise akış diyagramında görüldüğü gibi açılımda (3.19) terimine karşı gelen bir ikinci- derece modül oluşturulur. Bu modül, derecesi çift sayı olan prosedürlerle yoktur. Ardından kompleks eşlenik rezidü ve kutuplara sahip terimlerin (3.15)' de gösterilen işlem sonucunda oluşturduğu ikinci derece modüllere geçilir. Akış diyagramındaki r, filtre derecesinin ikiye bölünmesiyle elde bölümün tam kısmıdır. Örnek olarak beşinci dereceden bir filtre için (3.15)' den iki tane ikinci dereceden modül oluşur. Derece tek sayı olduğu için (3.19) terimi açılımda gözükür ve bu terimden de bir tane ikinci derece modül gelir. Böylece toplam ikinci-derece modül sayısı üç olur. Sonuçta girişin direkt terimle çarpılmasından oluşan değerle, bütün ikinci derece modüllerden gelen değerler toplanır ve sayısal analog dönüştürücüye gönderilir.

Direkt gerçeklemede olduğu gibi burada da program ALT tuşuna basılmadığı sürece çalışmaya devam eder.

4. SONUÇ

Gerçeklenen sayısal filtenin ilk olarak frekans spektrumu gözlenmiştir. Filtrenin frekans spektrumunu elde etmek için aşağıdaki düzenek kurulmuştur.

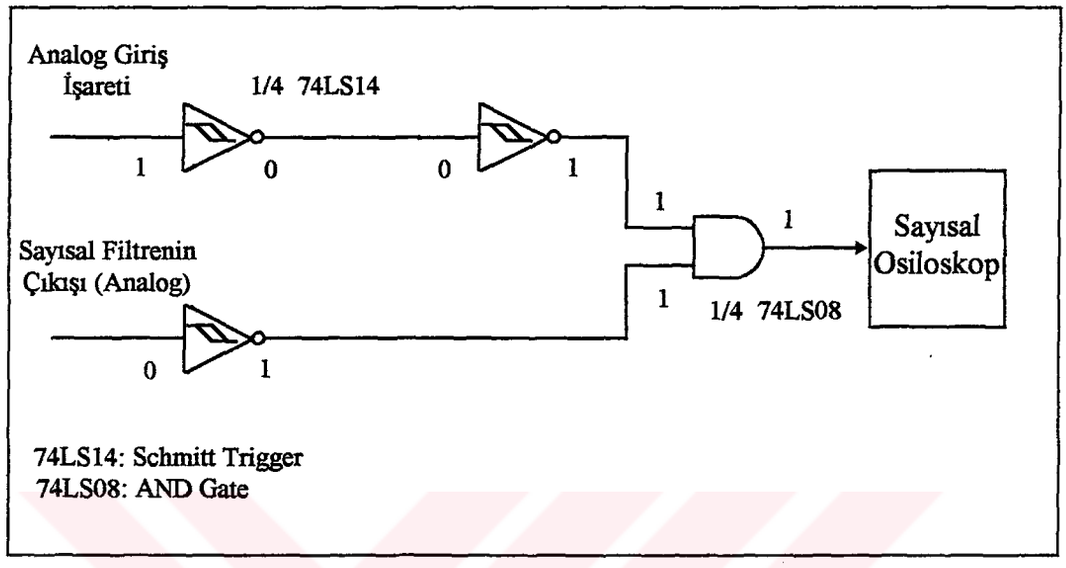


Şekil 4.1 Sayısal filtrenin genlik frekans spektrumunun gözlenmesi için oluşturulan deney düzeni

Bu deney düzeninde birinci fonksiyon üretecinin üçgen dalga çıkışı uygun gerilim değerleri veren yokuş fonksiyonu oluşturacak biçimde ayarlanır ve sinüs fonksiyonu oluşturacak biçimde ayarlanan ikinci fonksiyon üretecinin gerilim kontrollü osilatör (VCO- Voltage Controlled Osilator) girişine verilir. Böylece birinci işaret üretinden gelen değişik gerilim seviyelerine göre ikinci fonksiyon üretecinden belirli bir aralıkta küçükten büyüğe artan frekanslara sahip sinüsoidal işaretler üretilir ve bu işaretler sayısal filtreye uygulanır. Birinci fonksiyon üretecinin üçgen dalga çıkışı osiloskobun X kanalına (Ch1) ve sayısal filtrenin çıkışı da osikoskobun Y kanalına (Ch2) uygulanırsa sayısal filtrenin genlik frekans spektrumu osiloskop ekranında görülebilir.

Sayısal filtrenin genlik frekans spektrumu elde edildikten sonta Butterworth ve Chebyshev I türü tüm filtreler için kazanç ve faz farkı ölçümleri yapılmıştır. Aşağıdaki

şekilde sayısal filtrenin girişi ve çıkışı arasındaki faz farkını bulmak için oluşturulan düzenek görülmektedir.



Şekil 4.2 Faz farkı ölçümü için oluşturulan düzenek

Şekil 4.3' de görüldüğü gibi giriş işaretinin sıfırdan büyük değer alması ve sayısal filtrenin çıkışının ise sıfırdan küçük olması halinde AND kapısının çıkışı lojik 1 seviyesinde olur ve çıkışta iki işaret arasındaki zaman farkı (sayısal filtrenin çıkışının sıfırdan büyük olması anına kadar geçen süre) kadar genişliği olan bir darbe elde edilir. Bu darbenin genişliği kullanılan sayısal osiloskopla direkt olarak okunabilmektedir.

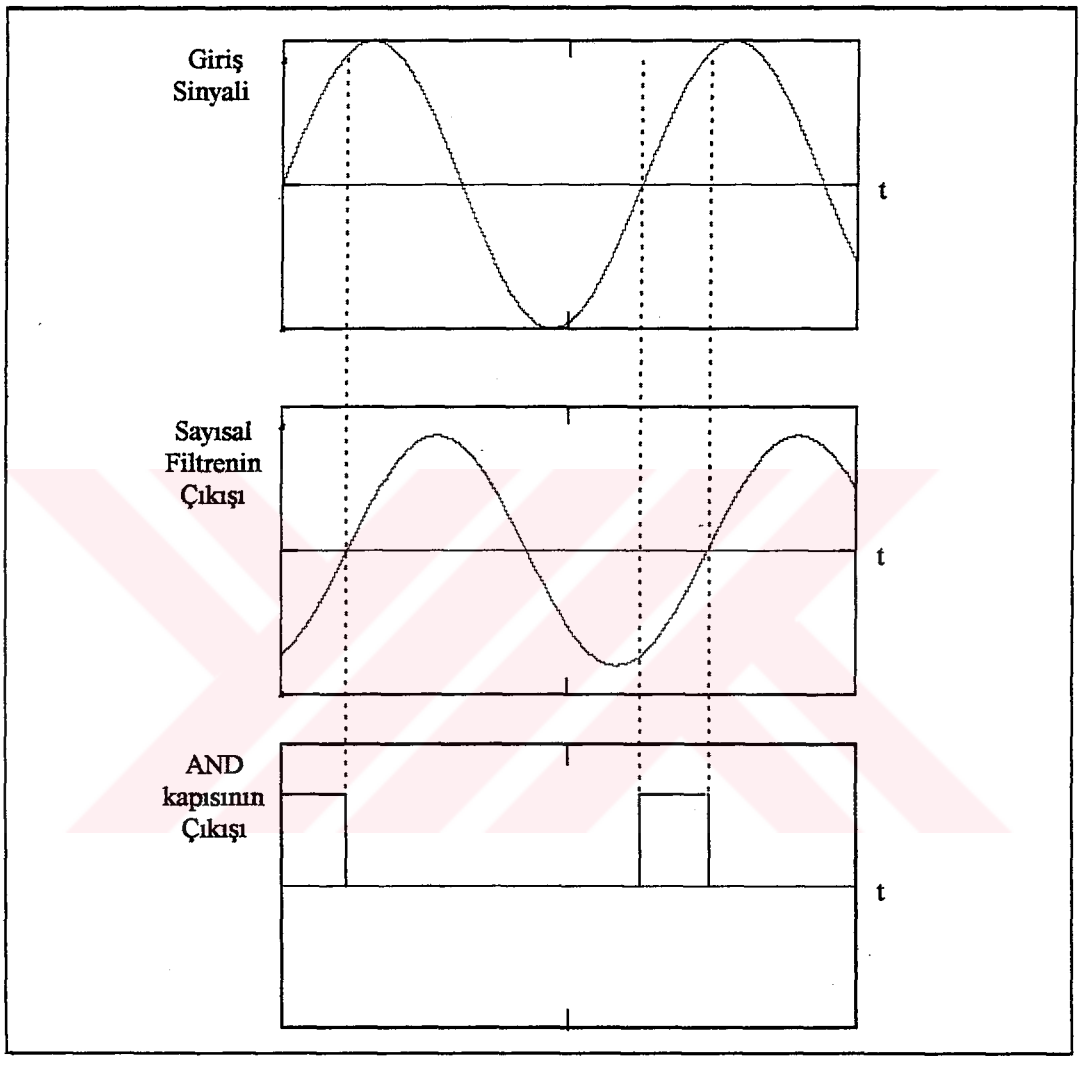
Elde edilen bu zaman farkı (Δt), analog frekansla (ω) çarpılırsa radyan olarak faz farkı bulunabilir:

$$Faz\ Farkı(Radyan) = \Delta t \times \omega$$

Eğer faz farkının derece olarak bulunması isteniyorsa, bu durumda;

$$Faz\ Farkı(Derece) = \frac{180}{\pi}\Delta t \times \omega = 360 \times f \times \Delta t$$

olacaktır.



Şekil 4.3

Aşağıda Butterworth ve Cbehyshev I türü filtreler için yapılan ölçümler görülmektedir. Alçak ve yüksek geçiren filtreler ikinci dereceden, band geçiren ve band söndüren filtreler birinci derecedendir. Kazanç ölçümlerinde sağlıklı ölçüm yapılabilmesi için sınır 5kHz olarak belirlenmiştir. Faz farkı ölçümlerinde bu sınır 3kHz' dir. Ölçüm yapılan alçak ve yüksek geçiren filtrelerde kesim frekansı 1kHz, band geçiren ve band söndüren filtrelerde

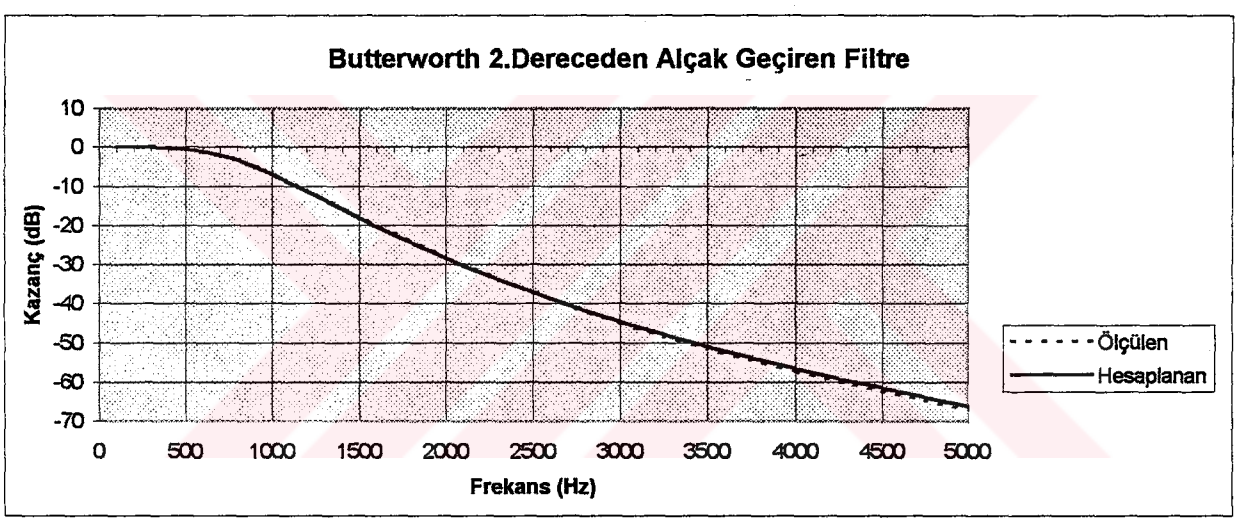
ise kesim frekansları 500Hz ve 1.5kHz' dir. (Wn = [W1 W2]) Chebyshev I türü filtreler için geçirme bandındaki zayıflatma 3dB' dir.

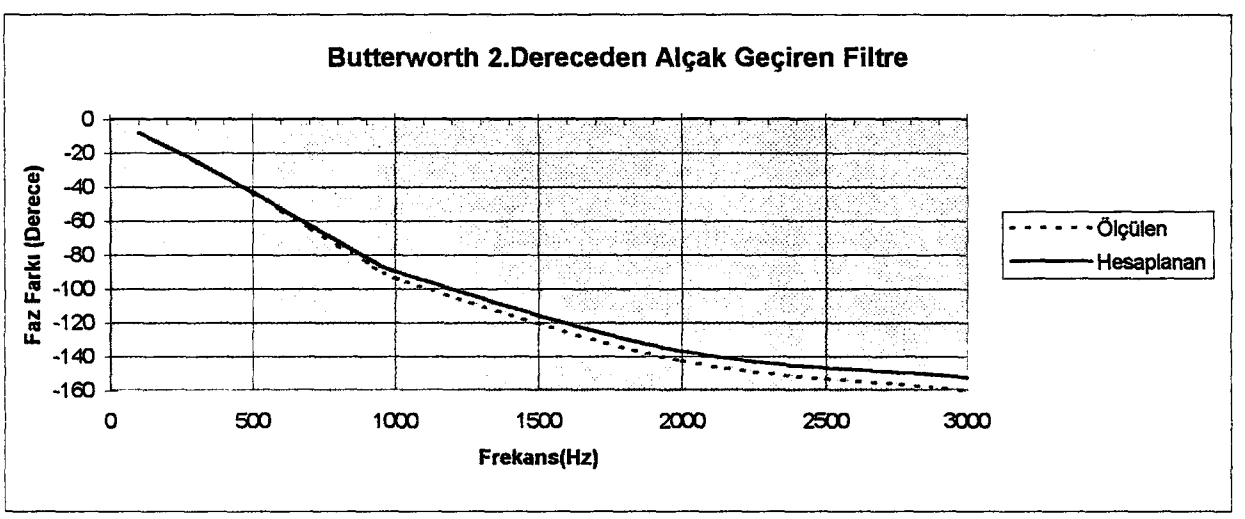
Ölçümler sonucunda gerçeklenen sayısal filtreyle frekansı 5kHz' e kadar olan işaretlerin sağlıklı bir şekilde örneklenip filtrelenebildiği görülmüştür. Bu değer kullanılan sayısal filtrenin donanımına bağlı olarak değişebilir. Eğer bu gerçeklemede kullanılanlardan daha hızlı ADC ve DAC kullanılırsa, sağlıklı örnekleme yapılabilen bu frekans değeri daha yukarı seviyelere çıkarılabilir. Yine ek olarak bu gerçeklemede kullanılan 12-bit ADC ve DAC yerine 16-bit ADC ve DAC kullanılırsa giriş sinyalinin kuantalanmasından meydana gelecek hatalar daha alt seviyelere indirilebilir.

Buna karşın bu gerçeklemede erişilen 5kHz' lik değer bile bugün birçok endüstriyel uygulama için yeterli bir seviye oluşturmaktadır. Endüstride sayısal filtreye gerek duyulan ortamlarda kullanılan malzemeye bağlı olarak analog giriş/çıkış kartı ile sayısal işaret işlemci kartları önemli özellikleri ile karşılaştırılılabilir. Sayısal filtre tasarımı, IBM uyumlu kişisel bilgisayar donanımına ek olarak kullanılan analog giriş/çıkış kartı, sayısal işaret işlemci kartından ucuz olduğundan daha ekonomik olarak gerçekleştirilmiştir. Bununla beraber her iki donanımın genel performansını artıran temel neden bu donanımlarda kullanılan tümdevrelerin ve yazılan filtre yazılımının performansına doğrudan bağlıdır.

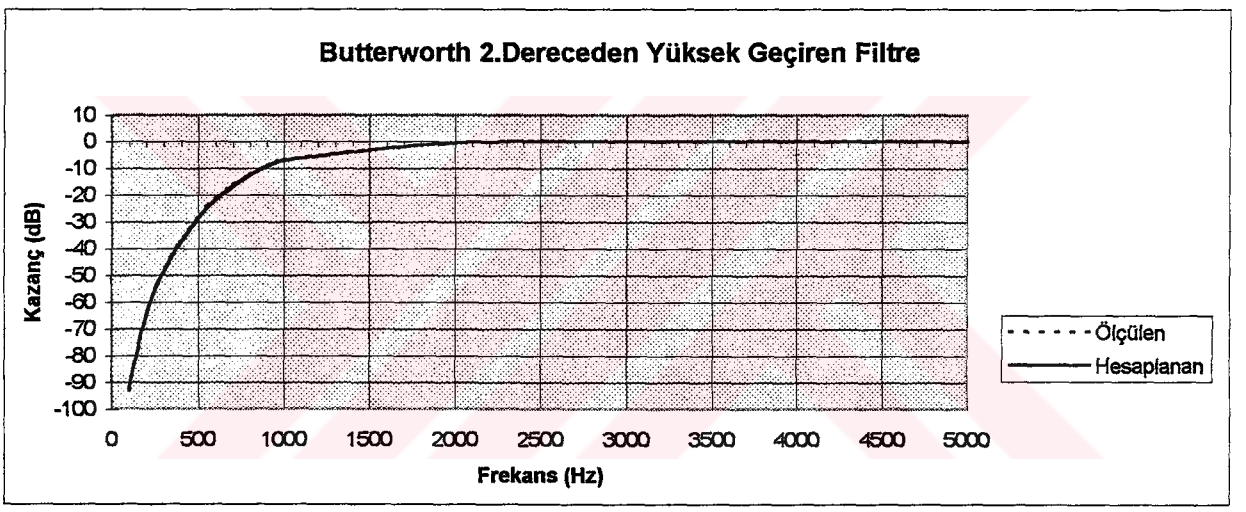
Sonuç olarak her iki veya diğer donanımlarda performans arttıkça sayısal filtrenin maliyeti artmaktadır. Fakat bu çalışmada kullanılan çok fonksiyonlu giriş/çıkış kartı daha geniş kullanım alanına sahip olduğu için filtrenin kullanıldığı ortamda sistem maliyetinin daha ekonomik olacağı açıktır.

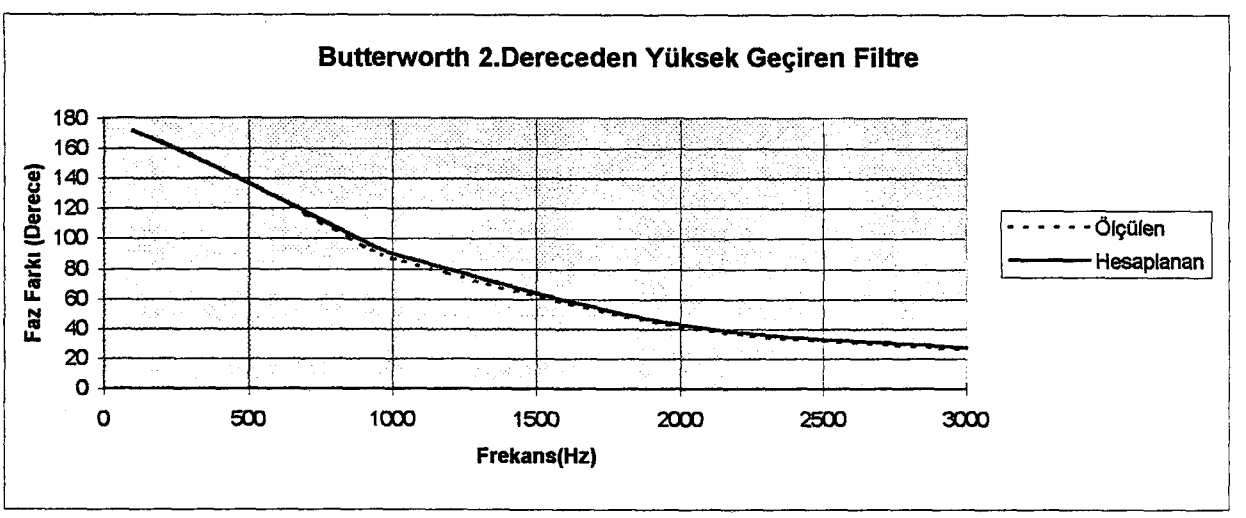
BUTTERWORTH 2.DERECE ALÇAK GEÇİREN SAYISAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	5,005	5,000	0,020	-0,001	0,005	0,100	-8,060	-8,107	0,047	0,584
200	4,990	4,996	-0,040	-0,016	0,006	0,120	-16,490	-16,369	0,121	0,739
300	4,990	4,980	-0,040	-0,080	0,010	0,200	-25,121	-24,924	0,197	0,791
400	4,950	4,938	-0,200	-0,250	0,012	0,250	-34,128	-33,862	0,266	0,786
500	4,870	4,852	-0,530	-0,601	0,018	0,370	-43,560	-43,195	0,365	0,844
600	4,730	4,707	-1,110	-1,208	0,023	0,490	-53,568	-52,844	0,724	1,370
700	4,470	4,493	-2,240	-2,137	0,023	0,520	-64,008	-62,607	1,401	2,238
800	4,240	4,216	-3,300	-3,413	0,025	0,580	-74,304	-72,221	2,083	2,884
900	3,860	3,890	-5,180	-5,023	0,030	0,760	-84,240	-81,400	2,840	3,489
1000	3,570	3,539	-6,740	-6,911	0,031	0,870	-93,600	-89,920	3,680	4,093
2000	1,220	1,202	-28,210	-28,511	0,018	1,510	-142,560	-136,914	5,646	4,124
3000	0,525	0,537	-45,080	-44,639	0,012	2,160	-159,840	-152,487	7,353	4,822
4000	0,285	0,295	-57,290	-56,618	0,010	3,320	Sağlıklı Ölçüm Yapılamıyor.			
5000	0,175	0,182	-67,050	-66,253	0,007	3,900				



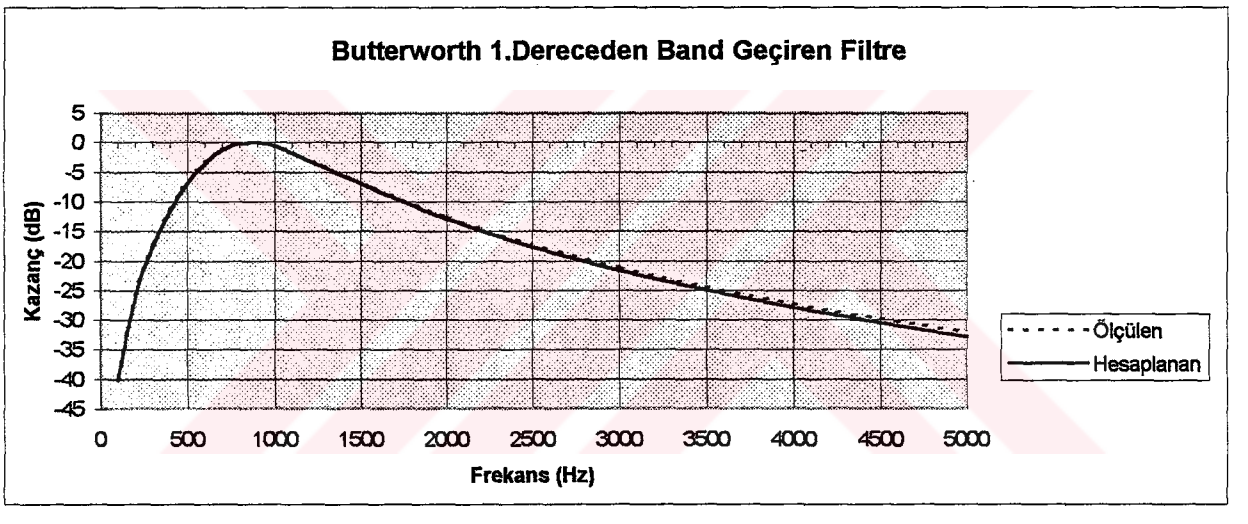


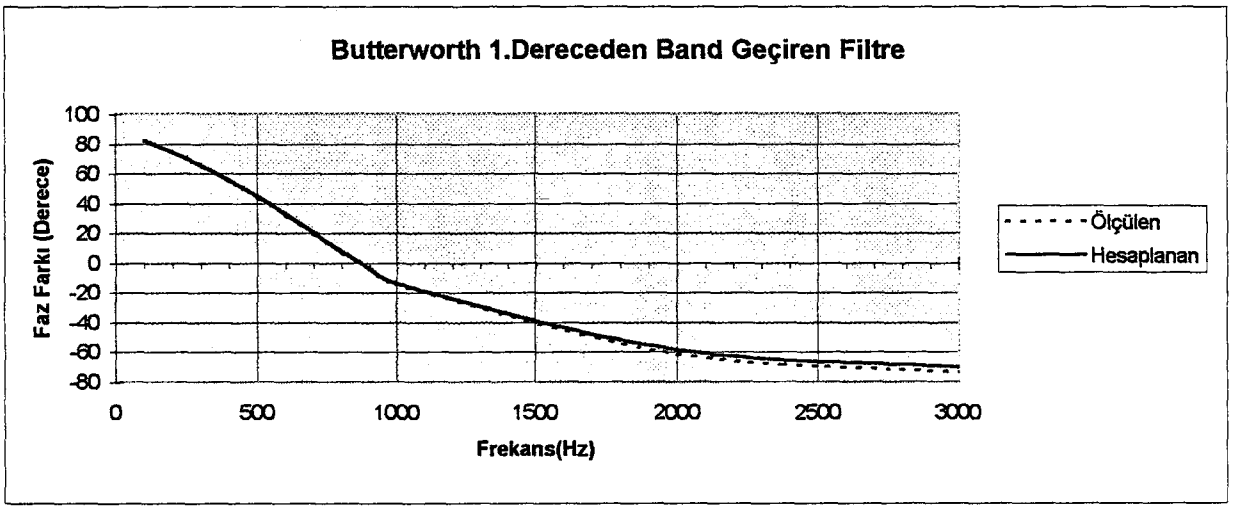
BUTTERWORTH 2.DERECE YÜKSEK GEÇİREN SAYISAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	0,048	0,050	-93,003	-92,224	0,002	3,823	171,360	171,893	0,533	0,310
200	0,193	0,199	-65,090	-64,508	0,006	2,869	164,232	163,631	0,601	0,367
300	0,458	0,446	-47,806	-48,351	0,012	2,760	155,736	155,077	0,659	0,425
400	0,805	0,786	-36,527	-37,007	0,019	2,430	146,880	146,139	0,741	0,507
500	1,235	1,207	-27,967	-28,424	0,028	2,311	137,520	136,805	0,715	0,523
600	1,725	1,687	-21,284	-21,731	0,038	2,259	127,872	127,157	0,715	0,563
700	2,240	2,193	-16,059	-16,483	0,047	2,143	115,920	117,393	1,473	1,255
800	2,740	2,689	-12,030	-12,408	0,051	1,908	105,984	107,779	1,795	1,666
900	3,180	3,142	-9,051	-9,292	0,038	1,213	95,256	98,600	3,344	3,392
1000	3,580	3,532	-6,682	-6,952	0,048	1,362	86,976	90,080	3,104	3,446
2000	4,900	4,853	-0,404	-0,595	0,047	0,960	41,400	43,086	1,686	3,914
3000	5,010	4,971	0,040	-0,116	0,039	0,783	26,201	27,513	1,313	4,771
4000	5,030	4,991	0,120	-0,035	0,039	0,775	Sağlıklı Ölçüm Yapılamıyor.			
5000	5,030	4,997	0,120	-0,013	0,033	0,666				



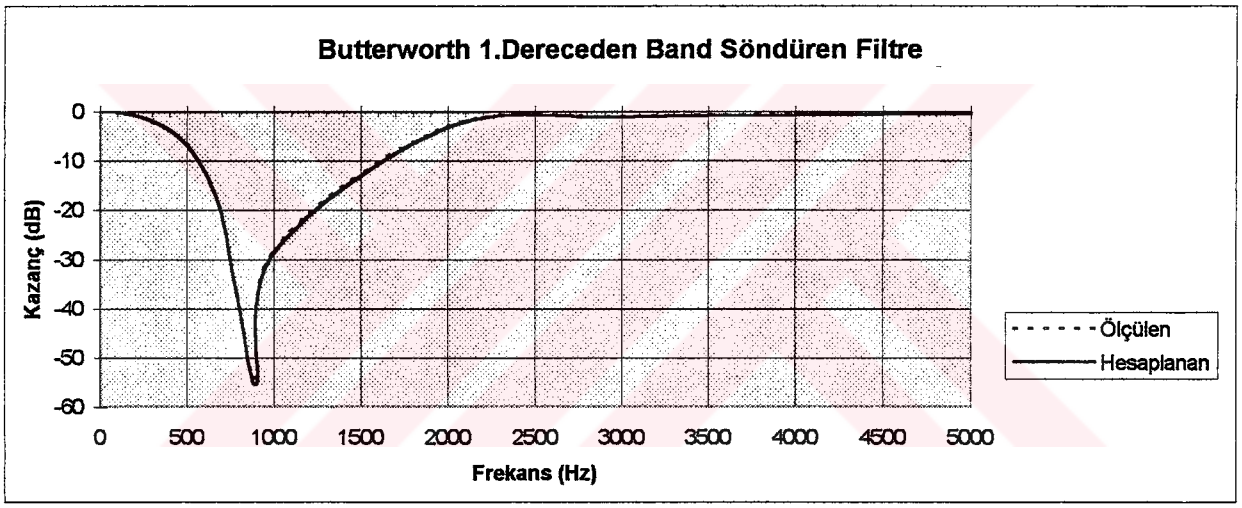


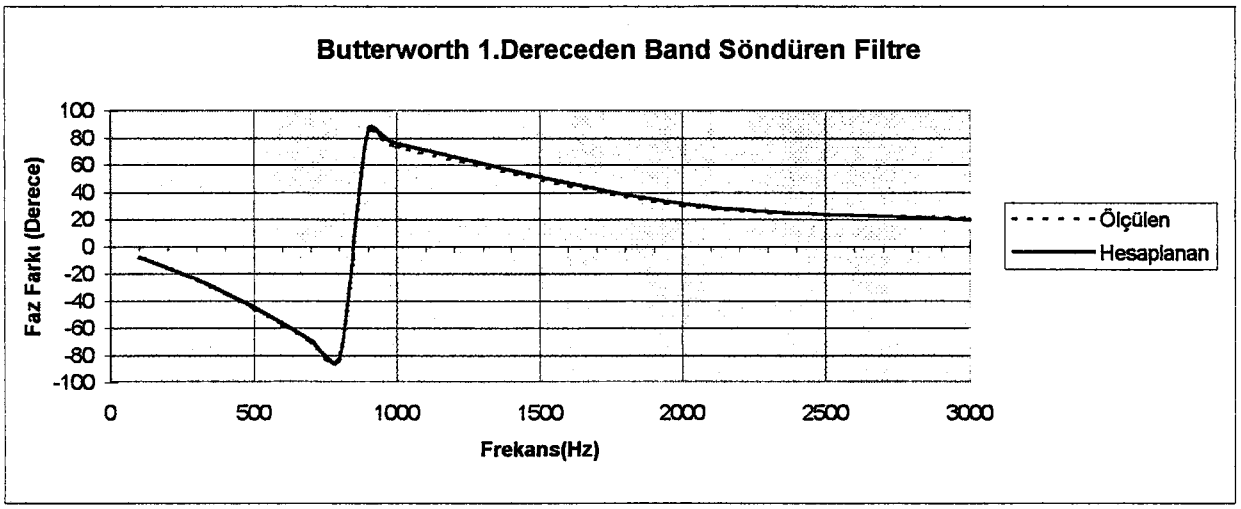
BUTTERWORTH 1.DERECE BAND GEÇİREN SAYISAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	0,690	0,670	-39,610	-40,200	0,020	3,016	82,080	82,300	0,220	0,267
200	1,390	1,356	-25,600	-26,100	0,034	2,522	74,592	74,261	0,331	0,446
300	2,110	2,069	-17,250	-17,650	0,042	2,006	65,880	65,546	0,334	0,509
400	2,850	2,804	-11,240	-11,570	0,046	1,641	55,296	55,869	0,573	1,026
500	3,580	3,532	-6,680	-6,950	0,048	1,359	44,460	45,029	0,569	1,263
600	4,230	4,187	-3,340	-3,550	0,043	1,017	32,616	33,083	0,467	1,410
700	4,710	4,683	-1,200	-1,310	0,027	0,585	20,160	20,472	0,312	1,523
800	4,960	4,951	-0,160	-0,200	0,009	0,178	7,776	7,953	0,177	2,221
900	4,985	4,990	-0,060	-0,040	0,005	0,100	-3,788	-3,678	0,110	2,980
1000	4,850	4,855	-0,610	-0,590	0,005	0,099	-14,400	-13,906	0,494	3,555
2000	2,660	2,616	-12,620	-12,960	0,044	1,686	-61,200	-58,465	2,735	4,679
3000	1,740	1,690	-21,110	-21,700	0,050	2,977	-73,721	-70,233	3,488	4,966
4000	1,280	1,237	-27,250	-27,930	0,043	3,451	Sağlıklı Ölçüm Yapılamıyor.			
5000	1,010	0,967	-31,990	-32,860	0,043	4,436				



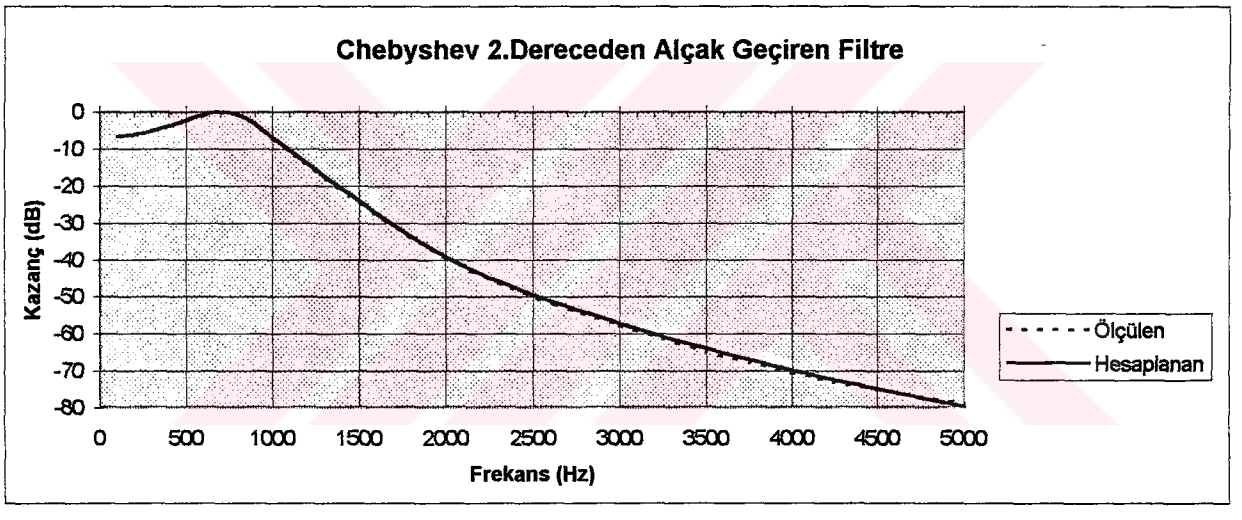


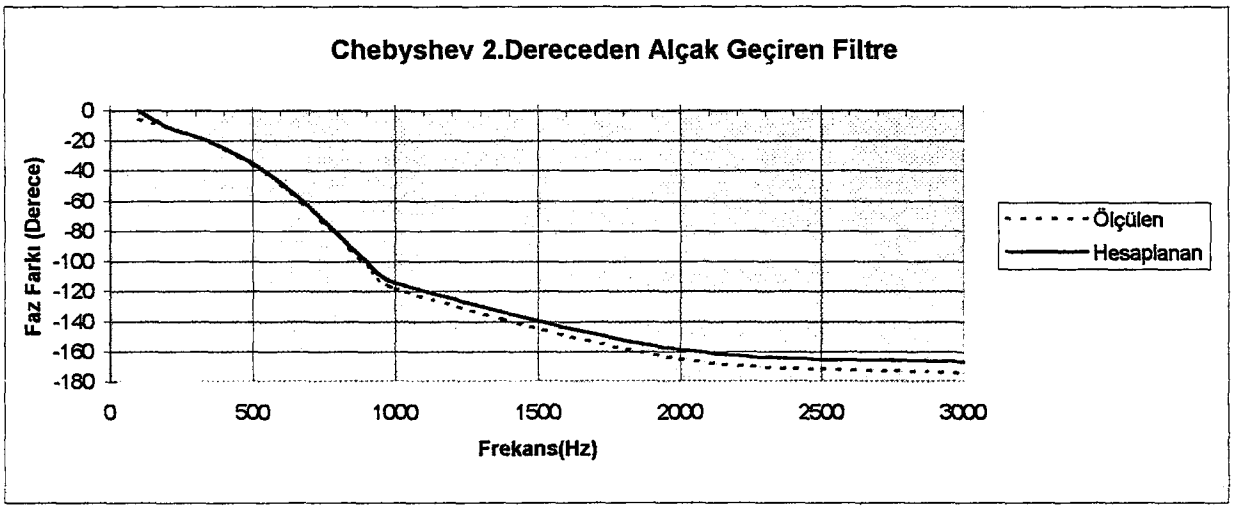
BUTTERWORTH 1.DERECE BAND SÖNDÜREN SAYISAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	4,980	4,955	-0,080	-0,180	0,025	0,507	-7,704	-7,701	0,003	0,045
200	4,840	4,813	-0,650	-0,760	0,027	0,567	-15,696	-15,733	0,037	0,238
300	4,520	4,552	-2,020	-1,880	0,032	0,705	-24,300	-24,437	0,137	0,559
400	4,110	4,140	-3,920	-3,780	0,030	0,720	-34,416	-34,108	0,308	0,902
500	3,570	3,539	-6,740	-6,910	0,031	0,873	-45,540	-44,943	0,597	1,329
600	2,700	2,732	-12,320	-12,090	0,032	1,186	-57,672	-56,872	0,800	1,407
700	1,730	1,753	-21,230	-20,960	0,023	1,318	-70,560	-69,477	1,083	1,559
800	0,680	0,697	-39,900	-39,410	0,017	2,439	-83,520	-81,985	1,535	1,873
900	0,330	0,315	-54,360	-55,270	0,015	4,662	84,240	86,385	2,145	2,483
1000	1,230	1,196	-28,050	-28,600	0,034	2,817	73,800	76,158	2,358	3,096
2000	4,310	4,261	-2,970	-3,200	0,049	1,148	30,240	31,547	1,307	4,143
3000	4,750	4,706	-1,030	-1,210	0,044	0,939	20,736	19,750	0,986	4,993
4000	4,880	4,845	-0,490	-0,630	0,035	0,733	Sağlıklı Ölçüm Yapılamıyor.			
5000	4,930	4,906	-0,280	-0,380	0,024	0,497				



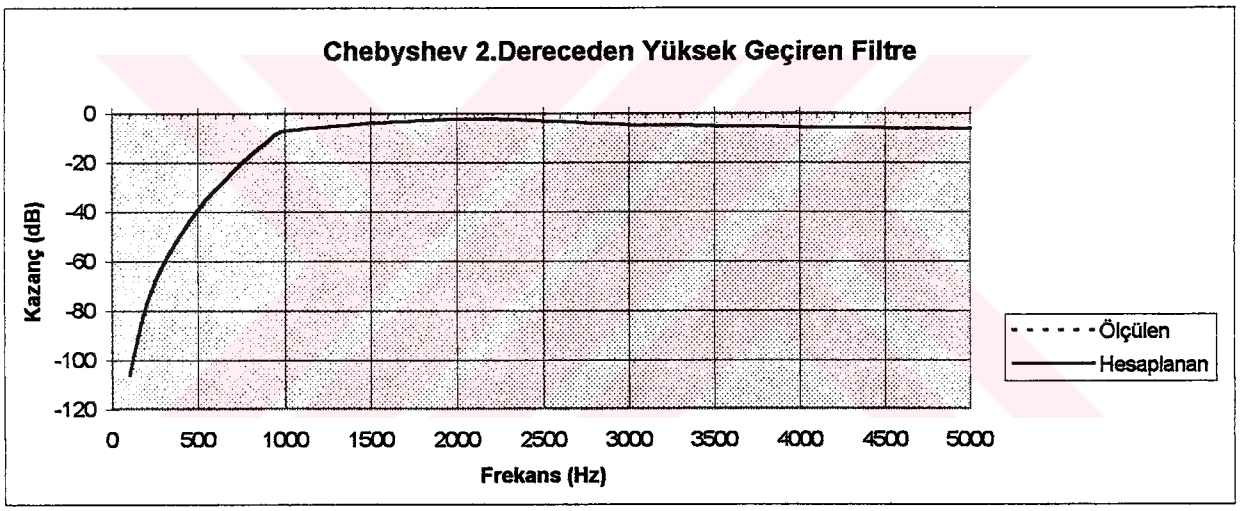


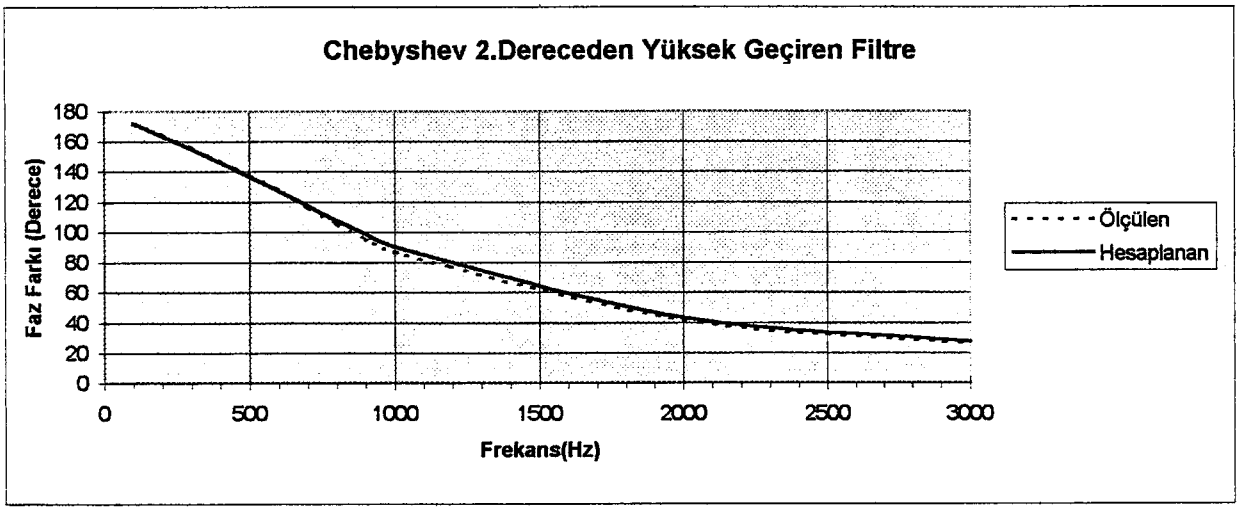
CHEBYSHEV 2.DERECE ALÇAK GEÇİREN SAYISAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	3,572	3,575	-6,726	-6,709	0,003	0,080	-5,300	-5,2654	0,035	0,700
200	3,678	3,683	-6,141	-6,115	0,005	0,130	-10,800	-10,898	0,098	0,900
300	3,862	3,868	-5,165	-5,133	0,006	0,160	-17,500	-17,326	0,174	1,000
400	4,126	4,135	-3,843	-3,801	0,008	0,210	-25,500	-25,118	0,382	1,500
500	4,460	4,470	-2,286	-2,241	0,010	0,220	-35,600	-35,013	0,587	1,700
600	4,800	4,812	-0,816	-0,768	0,012	0,240	-48,800	-47,840	0,960	2,000
700	4,980	4,999	-0,080	-0,006	0,019	0,370	-65,500	-63,982	1,518	2,400
800	4,800	4,821	-0,816	-0,729	0,021	0,440	-84,400	-82,248	2,152	2,600
900	4,240	4,261	-3,297	-3,200	0,021	0,490	-102,700	-99,769	2,931	2,900
1000	3,520	3,547	-7,020	-6,867	0,027	0,760	-118,080	-114,236	3,844	3,400
2000	0,690	0,701	-39,610	-39,285	0,011	1,610	-164,900	-158,738	6,162	3,900
3000	0,278	0,286	-57,791	-57,259	0,007	2,630	-173,900	-167,080	6,820	4,100
4000	0,147	0,152	-70,535	-69,814	0,005	3,540	Sağlıklı Ölçüm Yapılamıyor.			
5000	0,097	0,093	-78,850	-79,692	0,004	4,300				



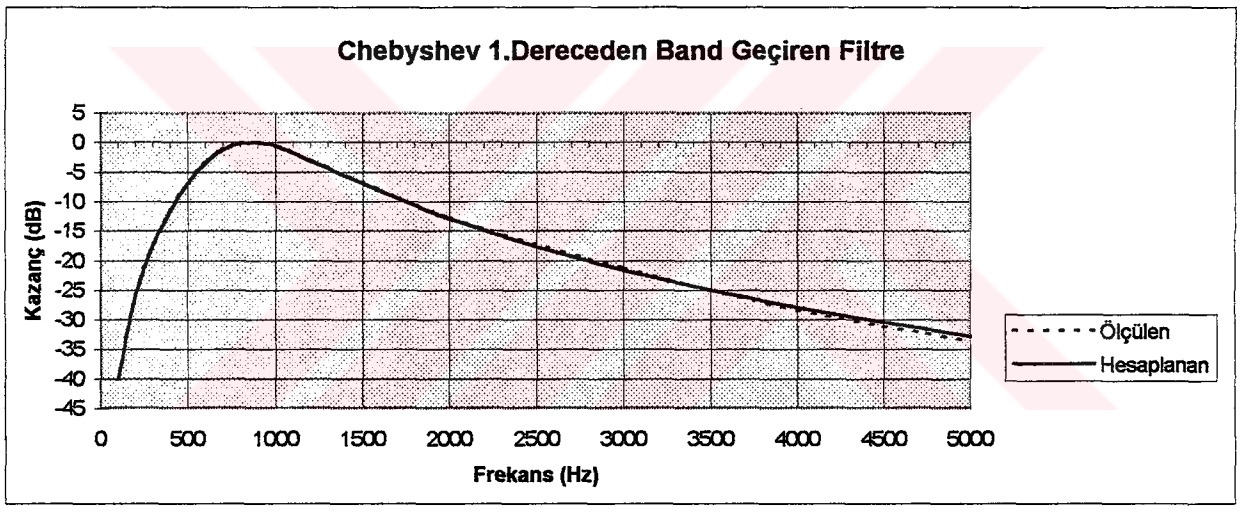


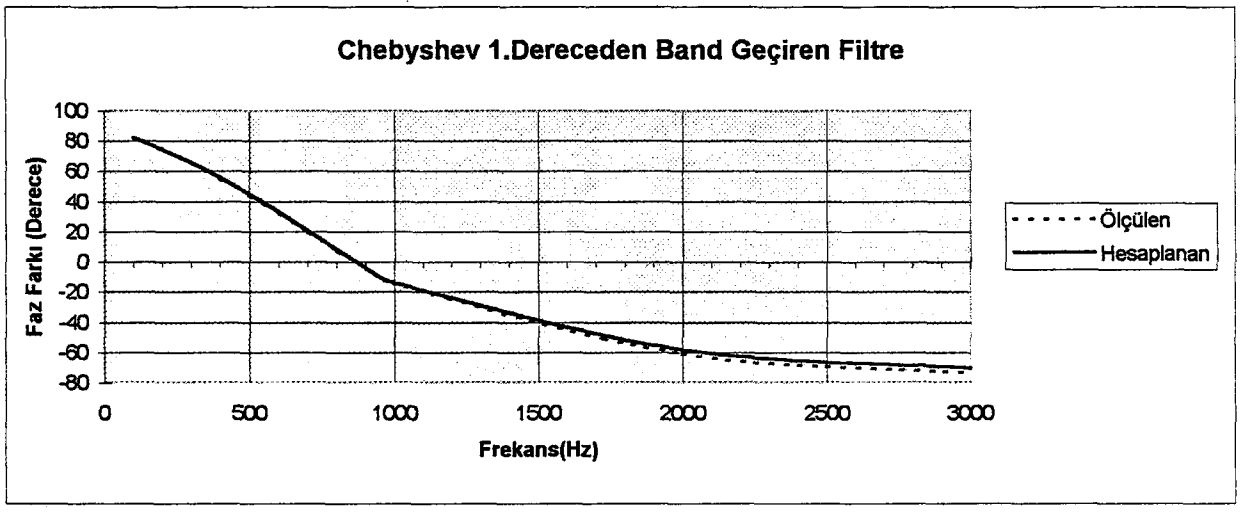
CHEBYSHEV 2.DERECE YÜKSEK GEÇİREN SAYISAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	0,026	0,025	-105,182	-105,966	0,001	4,000	171,360	171,893	0,533	0,310
200	0,105	0,102	-77,265	-77,903	0,003	3,245	164,232	163,631	0,601	0,367
300	0,24	0,235	-60,731	-61,195	0,006	2,345	155,736	155,077	0,659	0,425
400	0,425	0,432	-49,302	-48,989	0,007	1,552	146,880	146,139	0,741	0,507
500	0,715	0,705	-38,898	-39,183	0,010	1,433	137,520	136,805	0,715	0,523
600	1,08	1,069	-30,650	-30,856	0,011	1,038	127,872	127,157	0,715	0,563
700	1,555	1,539	-23,359	-23,564	0,016	1,027	115,920	117,393	1,473	1,255
800	2,14	2,124	-16,973	-17,119	0,016	0,734	105,984	107,779	1,795	1,666
900	2,825	2,809	-11,419	-11,531	0,016	0,566	95,256	98,600	3,344	3,392
1000	3,55	3,533	-6,850	-6,949	0,017	0,495	86,976	90,080	3,104	3,446
2000	4,482	4,466	-2,187	-2,260	0,016	0,363	41,400	43,086	1,686	3,914
3000	3,925	3,938	-4,841	-4,774	0,013	0,335	26,201	27,513	1,313	4,771
4000	3,735	3,754	-5,834	-5,732	0,019	0,506	Sağlıklı Ölçüm Yapılamıyor.			
5000	3,645	3,671	-6,322	-6,181	0,026	0,700				



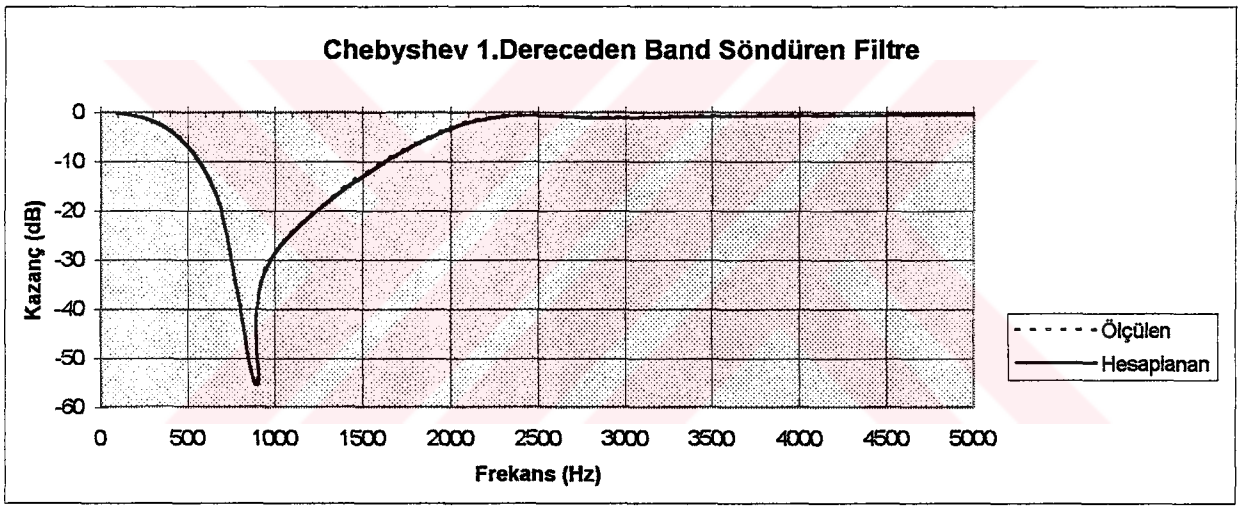


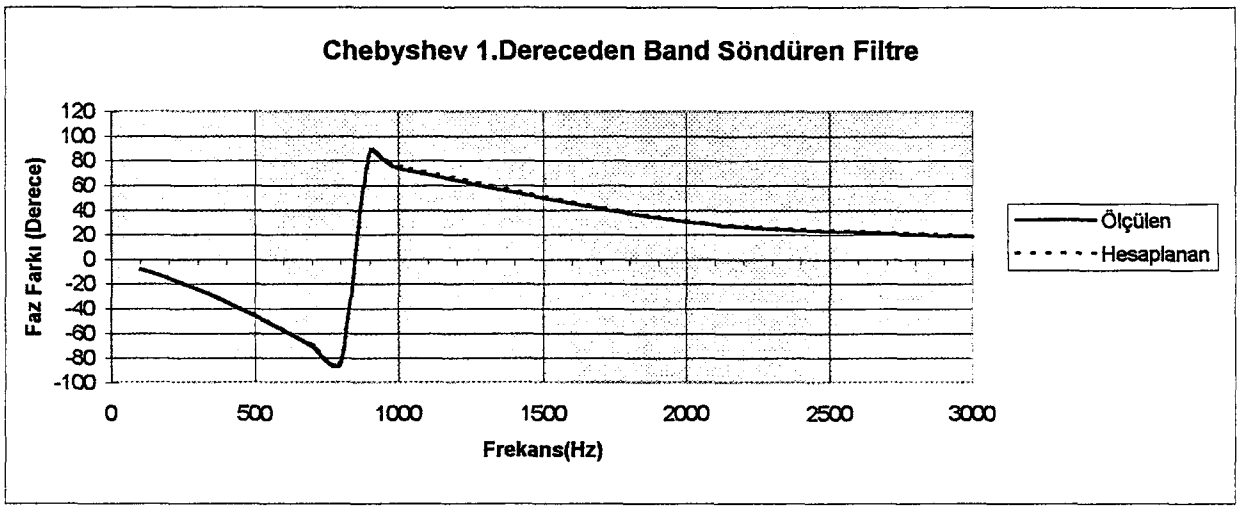
CHEBYSHEV 1.DERECE BAND GEÇİREN SAYİSAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	0,688	0,671	-39,668	-40,157	0,017	2,470	82,080	82,282	0,202	0,250
200	1,380	1,359	-25,747	-26,057	0,021	1,560	74,520	74,232	0,288	0,390
300	2,097	2,073	-17,379	-17,613	0,024	1,180	65,880	65,512	0,368	0,560
400	2,830	2,809	-11,383	-11,536	0,022	0,770	55,382	55,829	0,447	0,800
500	3,560	3,536	-6,794	-6,928	0,024	0,670	44,568	44,989	0,421	0,940
600	4,214	4,190	-3,421	-3,533	0,024	0,570	32,616	33,065	0,449	1,360
700	4,709	4,684	-1,199	-1,306	0,025	0,530	20,160	20,478	0,318	1,550
800	4,970	4,951	-0,120	-0,195	0,019	0,380	7,805	7,993	0,188	2,350
900	5,006	4,990	0,024	-0,040	0,016	0,320	-3,726	-3,610	0,116	3,210
1000	4,865	4,855	-0,547	-0,587	0,010	0,200	-14,328	-13,814	0,514	3,720
2000	2,650	2,621	-12,698	-12,921	0,030	1,130	-60,840	-58,390	2,450	4,200
3000	1,725	1,693	-21,284	-21,656	0,032	1,880	-73,440	-70,205	3,235	4,610
4000	1,210	1,240	-28,376	-27,887	0,030	2,420	Sağlıklı Ölçüm Yapılamıyor.			
5000	0,933	0,969	-33,576	-32,812	0,036	3,740				



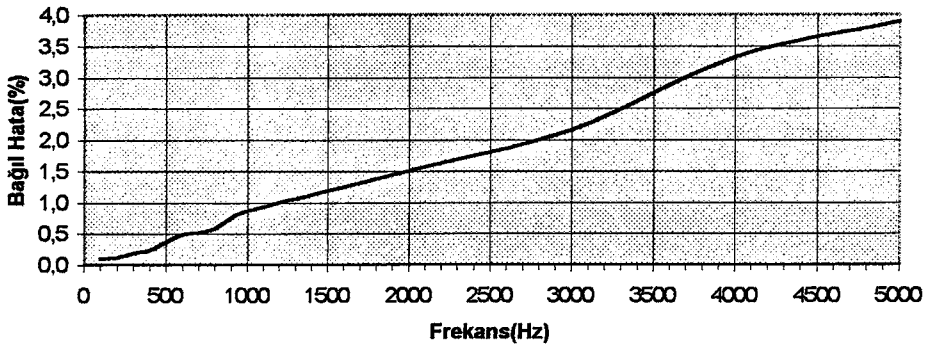


CHEBYSHEV 1.DERECE BAND SÖNDÜREN SAYISAL FİLTRE										
	Çıkış(V)		Kazanç(dB)		Çıkıştaki Hata		Faz Farkı(Derece)		Hata	
Frekans(Hz)	Ölçüm	Hesap	Ölçüm	Hesap	Mutlak	Bağıl(%)	Ölçüm	Hesap	Mutlak	Bağıl(%)
100	4,975	4,955	-0,100	-0,180	0,020	0,400	-7,704	-7,683	0,021	0,270
200	4,840	4,814	-0,650	-0,760	0,026	0,550	-15,797	-15,699	0,098	0,620
300	4,525	4,554	-1,996	-1,869	0,029	0,630	-24,602	-24,385	0,217	0,890
400	4,175	4,143	-3,606	-3,761	0,032	0,770	-34,416	-34,045	0,371	1,090
500	3,510	3,543	-7,076	-6,888	0,033	0,940	-45,504	-44,874	0,630	1,400
600	2,700	2,737	-12,324	-12,052	0,037	1,350	-57,672	-56,814	0,858	1,510
700	1,730	1,757	-21,226	-20,920	0,027	1,520	-70,560	-69,431	1,129	1,630
800	0,680	0,699	-39,902	-39,362	0,019	2,660	-83,520	-81,967	1,553	1,890
900	0,329	0,316	-54,423	-55,223	0,013	4,080	87,480	86,373	1,107	1,280
1000	1,225	1,199	-28,130	-28,559	0,026	2,170	74,160	76,123	1,963	2,580
2000	4,299	4,264	-3,021	-3,185	0,035	0,820	30,240	31,484	1,244	3,950
3000	4,740	4,707	-1,068	-1,207	0,033	0,700	18,792	19,710	0,918	4,660
4000	4,875	4,845	-0,506	-0,629	0,030	0,620	Sağlıklı Ölçüm Yapılamıyor.			
5000	4,880	4,906	-0,486	-0,380	0,026	0,530				

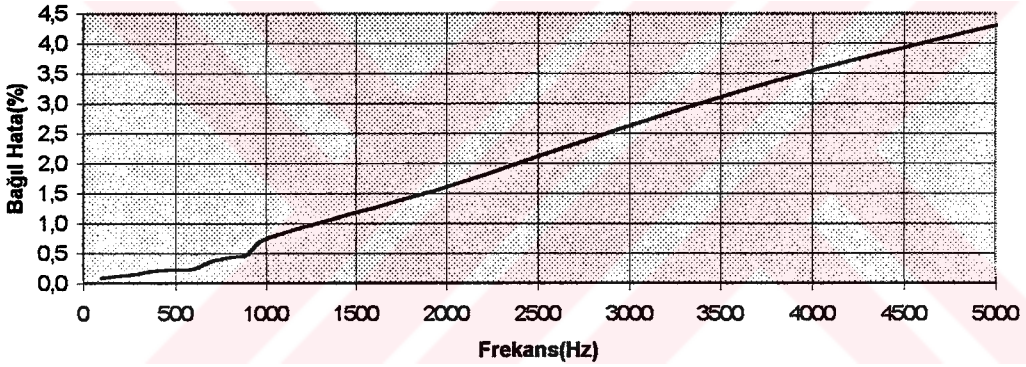




Butterworth 2.Dereceden Alçak Geçiren Filtrenin Çıkış Gerilimindeki Bağıl Hata



Chebyshev 2.Dereceden Alçak Geçiren Filtrenin Çıkış Gerilimindeki Bağıl Hata



5. KAYNAKLAR

[1] Phillips , Charles L.; Nagle H.Troy, "Digital Control System Analysis and Design", Prentice-Hall Internationals Editions, 1990

[2] Lawrence, R.Rabiner; Gold, Bernard, "Theory and Application of Digital Signal Processing", Prentice-Hall, Inc 1975.

[3] Kayran, Ahmet H., "Sayısal İşaret İşleme", İstanbul Teknik Üniversitesi Matbaası, Gümüşsuyu 1990.

[4] Nagle, H.T; Nelson, V.P, "Digital Filter Implementation on 16-bit Microcomputers", IEEE Micro, February 1991,pp.23-41

[5] Antoniou, Andreas, "Digital Filters Analysis, Design and Applications", Second Edition, McGraw-Hill, Inc.

[6] Ziemer, Rodger E.; Tranter, William H.; Fannin, D.Ronald, "Signals and Systems: Continuous and Discrete", Second Edition, Macmillan Publishing Company, New York

[7] Oppenheim, Alan V.; Schafer, Ronald W., "Digital Signal Processing", Prentice-Hall, Inc 1975

[8] Şen, A.Eşref, " Analog/Dijital Devreler ve Sistemler Teori ve Pratik", Kipaş Dağıtımcılık 1986.

[9] Detmer, Richard C., "Fundamentals of Assembly Language Programming Using The IBM PC and Compatibles", D.C.Heath and Company 1990.

[10] Etter, D.M, "Engineering Problem Solving with MATLAB", Prentice-Hall, Inc. 1993.

[11] Matlab 4.0 User's Guide.

[12] Turbo Pascal 5.0 User's Guide.

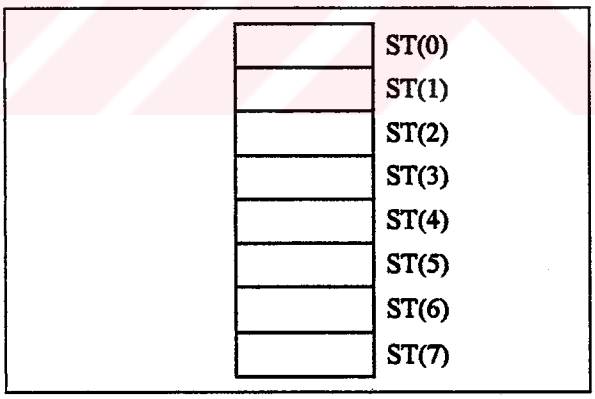
[13] Turbo Assembler Version 2.0 User's Guide.

[14] Embedded Microcomputers, Intel 1996.

6. EKLER

Ek1. 80387 Matematik Yardımcı İşlemcisi ve Özellikleri

Bir matematik yardımcı işlemcisi (Math Coprocessor) bilgisayar sistemlerinde karmaşık matematik işlemlerinin gerçeklenmesinde merkezi işlem birimine yardımcı olur. Bugün kullanılmakta olan tüm bilgisayarlarda bu işlem birimi mikroişlemcinin içerisinde tümleşik (built-in) olarak bulunmaktadır.Bu sayede bilgisayarlarda kayan noktalı aritmetik işlemleri, toplama ve çarpma gibi sıradan işlemler ve daha karmaşık matematiksel işlemler yapılabilir. Bir matematik işlemci merkezi işlem biriminden (CPU) tümüyle bağımsız kendi içsel yazmaçlarına sahiptir. Bellekten sadece kayan noktalı okuma veya yazma yapmaz. Bunlara ek olarak bellekten tamsayı ve BCD değerleri de okuyabilir. Ancak bu değerler kayan noktalı hale dönüştürülerek yazmaçlara alınır. Ayrıca içsel kayan nokta formatında olan bir sayıyı da tamsayı veya BCD hale getirerek belleğe de yazabilir.



Şekil E.1.1 80387' nin veri yazmaçları

80387 her biri 80 bit uzunluğunda olan sekiz tane veri yazmacına sahiptir. Bu yazmaçlarda değer saklanırken ANSI/IEEE-754 standartı kullanılılır. Yazmaçlar basit bir yığın (stack) şeklinde organize edilmiştir. Örnek olarak bellekten bir değer yüklendiğinde, bu değer yığının en üstündeki yazmaça yerleştirilir ve diğer yazmaçlar bir aşağıya çekilir.

Bununla beraber bazı komutlarla bu sekiz yazmaçtan herhangi birine erişilebilir. Yani organizasyon tümüyle yığın şeklinde değildir.

Sayısal filtre programında kullanılan 80387 Komutları:

finit: 80387' yi başlatma (yazmaçlarını sıfırlama)
fld memory(real)**:** Bellekteki reel bir değeri 80387' nin yazmacına (ST(0)'a) yazma.
fild memory(integer)**:** Bellekteki bir tamsayıyı kayan noktalı (floating point) hale getitip yığına (80387' nin yazmacına) koyma.
fst st(num)**:** ST(num)' ın içeriğini ST' nin içeriği ile değiştirme. Bu komuttan sadece ST(num) etkilenir.
fst memory(real)**:** ST' nin içeriğini bir reel değer olarak bellekte saklama.
fist memory(integer)**:** ST' nin içeriğini tamsayıya çevirip, bellekte tamsayı bir değer olarak saklama.
fadd memory(real)**:** ST' nin içeriğini bellekteki bir reel değer ile toplama ve toplama işlemi sonucunda elde edilen değeri ST' de saklama.
fmul: ST(ST(0)) ve ST(1)' in değerlerini yığından çekme ve bunların çarpımları sonucunda elde edilen değeri ST' ye yazma.
fchs: ST' deki sayının işaretini değiştirme.

Ek2. Sayısal Filtrelerin Direkt Gerçeklenmesi Programı

```
Program SAYISAL_FİLTRELERİN_DİREKT_GERÇEKLENMESİ;

{Beşir TAYFUR-Y.T.Ü 1997}

{Kullanılan sayısal filtre yapısı = 1D, Maksimum derece = 8}
{MPIBM4 kartının taban adresi 0300H = 768}

{*************************************************************************}

Uses Crt;
Var

Dos1:text;
Dos2:text;
Dos3:text;
Dos4:text;

derece:integer;
kanal:byte;
tur:integer; {1-Alçak veya Yüksek Geçiren,
              2-Band Geçiren veya Söndüren}

n:integer;
temp:word;
adc:word;

{Sayısal filtre katsayıları}
a0:double;
a1:double;
a2:double;
a3:double;
a4:double;
a5:double;
a6:double;
a7:double;
a8:double;

b0:double;
b1:double;
b2:double;
b3:double;
b4:double;
b5:double;
b6:double;
b7:double;
b8:double;

T1:double;
T2:double;

{Zaman gecikme elemanları}
tempm1:double;  {m(k-1)}
tempm2:double;  {m(k-2)}
tempm3:double;  {m(k-3)}
tempm4:double;  {m(k-4)}
tempm5:double;  {m(k-5)}
tempm6:double;  {m(k-6)}
tempm7:double;  {m(k-7)}
tempm8:double;  {m(k-8)}

coeffa:array[0..8] Of real;
coeffb:array[0..8] Of real;

{*************************************************************************}

  Procedure derece1;

  begin
  asm

@l4: mov bx,1
@l3: mov cx,50000

{-----------------------------------------}
{T1 değerinin hesaplanması}

@l2: finit                    {T1=-(b1.m(k-1))}
    fld tempm1
    fst st(1)
    fld b1
    fmul
    fchs
    fst T1

{-----------------------------------------}
{T2 değerinin hesaplanması}

    fld tempm1                {T2=a1.m(k-1)}
    fst st(1)
    fld a1
    fmul
    fst T2

{-----------------------------------------}
 {Giriş sinyalinden 12-bit örnek alınması}
```

```
  mov ah,00
  mov dx,784
  mov al,kanal          {ADC kanalı seçimi}
  out dx,al

  mov dx,788
@11:in al,dx
  test al,01      {Dönüşüm tamamlandı mı ?)
  jz @11

  dec dx          {Dönüştürülen değeri okuma}
  in al,dx
  and al,240  {En düşün ağırlıklı 4-biti okuma
ve istenmeyen bitleri maskeleme}
  shr al,4
  mov temp,ax

  dec dx    {Yüksek anlamlı baytı okuma ve en
düşük anlanlı 4-bitle toplayıp 12-bit değer elde
etme}
  in ax,dx
  shl ax,4
  add ax,temp
  sub ax,2048     {Girişi işaretli hale getirme}
  mov adc,ax

{-------------------------------------------}
  finit                        {m(k)=x(k)+T1}
  fadd T1
  fild adc
  fst tempm1
  fst st(1)

{-------------------------------------------}
  fld a0                       {y(k)=a0.m(k)+T2}
  fmul
  fadd T2
  fist adc

  mov ax,adc
  add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776  {DAC A' ya en az ağırlıklı
bitleri gönderme }
  out dx,al
  inc dx        {DAC A' ya en ağırlıklı bitleri
gönderme }
  mov al,ah
  out dx,al
  mov dx,780            {DAC'yi güncelleme}
  mov al,00
  out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @12

  dec bx
  cmp bx,00
  jne @13

  mov ah,02
  int 16h
  and al,08       {ALT tuşu dışındaki tuşları
maskeleme}
  cmp al,08
  je @15
  jmp @14

@15: end;
   end;
```

{**}

Procedure derece2;

```
  begin
  asm

@14: mov bx,1
@13: mov cx,50000

{-------------------------------------------}
{T1 değerinin hesaplanması}

@12: finit
   fld tempm1
   fst st(1)
   fld b1
   fmul
   fst T1

   fld tempm2
   fst st(1)
   fld b2
   fmul
   fadd T1
   fchs
   fst T1

{-------------------------------------------}
{T2 değerinin hesaplanması}
```

```
  fld tempm1
  fst st(1)
  fld a1
  fmul
  fst T2

  fld tempm2
  fst st(1)
  fld a2
  fmul
  fadd T2
  fst T2

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

  mov ah,00
  mov dx,784
  mov al,kanal
  out dx,al

  mov dx,788
@l1:in al,dx
  test al,01
  jz @l1

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec dx
  in ax,dx
  shl ax,4
  add ax,temp
  sub ax,2048
  mov adc,ax

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm1          {m(k-2)→m(k-1)}
  fst tempm2

{-------------------------------------------}
```

```
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-------------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc
  mov ax,adc
  add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @l2

  dec bx
  cmp bx,00
  jne @l3

  mov ah,02
  int 16h
  and al,08
  cmp al,08
  je @l5
  jmp @l4

@l5: end;
   end;
```

{**}

```
Procedure derece3;
  begin
  asm

@l4: mov bx,1

@l3: mov cx,50000

{-------------------------------------------}
{T1 değerinin hesaplanması}
```

```
@l2: finit
    fld tempm1
    fst st(1)
    fld b1
    fmul
    fst T1

    fld tempm2
    fst st(1)
    fld b2
    fmul
    fadd T1
    fst T1

    fld tempm3
    fst st(1)
    fld b3
    fmul
    fadd T1.
    fchs
    fst T1

{-------------------------------------------}
{T2 değerinin hesaplanması}

    finit
    fld tempm1
    fst st(1)
    fld a1
    fmul
    fst T2

    fld tempm2
    fst st(1)
    fld a2
    fmul
    fadd T2
    fst T2

    fld tempm3
    fst st(1)
    fld a3
    fmul
    fadd T2
    fst T2

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

  mov ah,00
  mov dx,784
  mov al,kanal
  out dx,al

  mov dx,788
@l1:in al,dx
  test al,01
  jz @l1

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec dx
  in ax,dx
  shl ax,4
  add ax,temp
  sub ax,2048
  mov adc,ax

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm2
  fst tempm3
  fld tempm1
  fst tempm2

{-------------------------------------------}
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-------------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc
  mov ax,adc
  add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
```

```
  jne @l2

  dec bx
  cmp bx,00
  jne @l3

  mov ah,02
  int 16h
  and al,08
```

```
                cmp al,08
                je @l5
                jmp @l4
@l5: end;
                end;
```

{**}

Procedure derece4;

```
  begin
  asm

@l4: mov bx,1
@l3: mov cx,50000

{-----------------------------------------}
{T1 değerinin hesaplanması}

@l2: finit
    fld tempm1
    fst st(1)
    fld b1
    fmul
    fst T1

    fld tempm2
    fst st(1)
    fld b2
    fmul
    fadd T1
    fst T1

    fld tempm3
    fst st(1)
    fld b3
    fmul
    fadd T1
    fst T1

    fld tempm4
    fst st(1)
    fld b4
    fmul
    fadd T1
    fchs
    fst T1

{-----------------------------------------}
{T2 değerinin hesaplanması}

    finit

    fld tempm1
    fst st(1)
    fld a1
    fmul
    fst T2

    fld tempm2
    fst st(1)
    fld a2
    fmul
    fadd T2
    fst T2

    fld tempm3
    fst st(1)
    fld a3
    fmul
    fadd T2
    fst T2

    fld tempm4
    fst st(1)
    fld a4
    fmul
    fadd T2
    fst T2

{-----------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

    mov ah,00
    mov dx,784
    mov al,kanal
    out dx,al

    mov dx,788
@l1:in al,dx
    test al,01
    jz @l1

    dec dx
    in al,dx
    and al,240
    shr al,4
    mov temp,ax
```

```
  dec dx
  in ax,dx
  shl ax,4
  add ax,temp
  sub ax,2048
  mov adc,ax

{-----------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm3
  fst tempm4
  fld tempm2
  fst tempm3
  fld tempm1
  fst tempm2

{-----------------------------------------}
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-----------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc
  mov ax,adc
  add ax,2048

{-----------------------------------------}
```

```
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al

{-----------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @l2

  dec bx
  cmp bx,00
  jne @l3

  mov ah,02
  int 16h
  and al,08
  cmp al,08
  je @l5
  jmp @l4

@l5: end;
   end;
```

{**}

Procedure derece5;

```
  begin
  asm

@l4: mov bx,1
@l3: mov cx,50000

{-----------------------------------------}
{T1 değerinin hesaplanması}

@l2: finit
    fld tempm1
    fst st(1)
    fld b1
    fmul
    fst T1

    fld tempm2
    fst st(1)
    fld b2
    fmul
    fadd T1
    fst T1

    fld tempm3
    fst st(1)
    fld b3
    fmul
    fadd T1
    fst T1

    fld tempm4
    fst st(1)
    fld b4
    fmul
    fadd T1
    fst T1

    fld tempm5
    fst st(1)
```

```
    fld b5
    fmul
    fadd T1
    fchs
    fst T1

{-------------------------------------------}
{T2 değerinin hesaplanması}

    finit
    fld tempm1
    fst st(1)
    fld a1
    fmul
    fst T2

    fld tempm2
    fst st(1)
    fld a2
    fmul
    fadd T2
    fst T2

    fld tempm3
    fst st(1)
    fld a3
    fmul
    fadd T2
    fst T2

    fld tempm4
    fst st(1)
    fld a4
    fmul
    fadd T2
    fst T2

    fld tempm5
    fst st(1)
    fld a5
    fmul
    fadd T2
    fst T2

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

  mov ah,00
  mov dx,784
  mov al,15
  out dx,al

  mov dx,788
@11:in al,dx
  test al,01
  jz @11

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec dx
  in ax,dx
  shl ax,4
  add ax,temp

  sub ax,2048
  mov adc,ax

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm4
  fst tempm5
  fld tempm3
  fst tempm4
  fld tempm2
  fst tempm3
  fld tempm1
  fst tempm2

{-------------------------------------------}
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-------------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc

  mov ax,adc
  add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al

{-------------------------------------------}
```

```
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @l2

  dec bx
  cmp bx,00
  jne @l3
  mov ah,02
  int 16h
  and al,08
  cmp al,08
  je @l5
  jmp @l4

@l5: end;
  end;
```

{**********}

Procedure derece6;

```
  begin
  asm

@l4: mov bx,1
@l3: mov cx,50000

{-----------}
{T1 değerinin hesaplanması}

@l2: finit
    fld tempm1
    fst st(1)
    fld b1
    fmul
    fst T1

    fld tempm2
    fst st(1)
    fld b2
    fmul
    fadd T1
    fst T1

    fld tempm3
    fst st(1)
    fld b3
    fmul
    fadd T1
    fst T1

    fld tempm4
    fst st(1)
    fld b4
    fmul
    fadd T1
    fst T1


    fld tempm5
    fst st(1)
    fld b5
    fmul
    fadd T1
    fst T1

    fld tempm6
    fst st(1)
    fld b6
    fmul
    fadd T1
    fchs
    fst T1

{-----------}
{T2 değerinin hesaplanması}

    finit
    fld tempm1
    fst st(1)
    fld a1
    fmul
    fst T2

    fld tempm2
    fst st(1)
    fld a2
    fmul
    fadd T2
    fst T2

    fld tempm3
    fst st(1)
    fld a3
    fmul
    fadd T2
    fst T2

    fld tempm4
    fst st(1)
    fld a4
    fmul
    fadd T2
    fst T2

    fld tempm5
    fst st(1)
```

```
  fld a5
  fmul
  fadd T2
  fst T2

  fld tempm6
   fst st(1)
  fld a6
  fmul
  fadd T2
  fst T2

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

  mov ah,00
  mov dx,784
  mov al,15
  out dx,al

  mov dx,788
@11:in al,dx
  test al,01
  jz @11

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec dx
  in ax,dx
  shl ax,4
  add ax,temp

  sub ax,2048
  mov adc,ax

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm5
  fst tempm6
  fld tempm4
  fst tempm5
  fld tempm3
  fst tempm4
  fld tempm2
  fst tempm3
  fld tempm1
  fst tempm2

{-------------------------------------------}
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-------------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc
  mov ax,adc
  add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @12

  dec bx
  cmp bx,00
  jne @13

  mov ah,02
  int 16h
  and al,08
  cmp al,08
  je @15
  jmp @14

@15: end;
   end;
```

```
{*****************************************************************************}
```

Procedure derece7;

```
  begin
  asm

@l4: mov bx,1
@l3: mov cx,50000

{-----------------------------------------}
```

{T1 değerinin hesaplanması}

```
@l2: finit
    fld tempm1
    fst st(1)
    fld b1
    fmul
    fst T1

    fld tempm2
    fst st(1)
    fld b2
    fmul
    fadd T1
    fst T1

    fld tempm3
    fst st(1)
    fld b3
    fmul
    fadd T1
    fst T1

    fld tempm4
    fst st(1)
    fld b4
    fmul
    fadd T1
    fst T1

    finit
    fld tempm5
    fst st(1)
    fld b5
    fmul
    fadd T1
    fst T1

    fld tempm6
    fst st(1)
    fld b6
    fmul
    fadd T1
    fst T1

    fld tempm7
    fst st(1)
    fld b7
    fmul
    fadd T1
    fchs
    fst T1

{-----------------------------------------}
```

{T2 değerinin hesaplanması}

```
    finit
    fld tempm1
    fst st(1)
    fld a1
    fmul
    fst T2

    fld tempm2
    fst st(1)
    fld a2
    fmul
    fadd T2
    fst T2

    fld tempm3
    fst st(1)
    fld a3
    fmul
    fadd T2
    fst T2

    fld tempm4
    fst st(1)
    fld a4
    fmul
    fadd T2
    fst T2

    finit
    fld tempm5
    fst st(1)
    fld a5
    fmul
    fadd T2
    fst T2

    fld tempm6
    fst st(1)
    fld a6
    fmul
    fadd T2
    fst T2

    fld tempm7
    fst st(1)
    fld a7
```

```
    fmul
    fadd T2
    fst T2

{-------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

 mov ah,00
  mov dx,784
  mov al,15
  out dx,al

  mov dx,788
@11:in al,dx
  test al,01
  jz @11

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec
  in ax,dx
  shl ax,4
  add ax,temp

  sub ax,2048
  mov adc,ax

{-------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm6
  fst tempm7
  fld tempm5
  fst tempm6
  fld tempm4
  fst tempm5
  fld tempm3
  fst tempm4
  fld tempm2
  fst tempm3
  fld tempm1
  fst tempm2

{-------------------------------------}
```

```
  finit
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc

  mov ax,adc
  add ax,2048

{-------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al

{-------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @12

  dec bx
  cmp bx,00
  jne @13

  mov ah,02
  int 16h
  and al,08
  cmp al,08
  je @15
  jmp @14

@15: end;
   end;
```

```
{*************************************************************************}
```

Procedure derece8;

```
  begin
  asm

@14: mov bx,1
@13: mov cx,50000
```

```
{-------------------------------------}
{T1 değerinin hesaplanması}

@l2: finit
   fld tempm1
   fst st(1)
   fld b1
   fmul
   fst T1

   fld tempm2
   fst st(1)
   fld b2
   fmul
   fadd T1
   fst T1

   fld tempm3
   fst st(1)
   fld b3
   fmul
   fadd T1
   fst T1

   fld tempm4
   fst st(1)
   fld b4
   fmul
   fadd T1
   fst T1

   finit
   fld tempm5
   fst st(1)
   fld b5
   fmul
   fadd T1
   fst T1

   fld tempm6
   fst st(1)
   fld b6
   fmul
   fadd T1
   fst T1

   fld tempm7
   fst st(1)
   fld b7
   fmul
   fadd T1
   fst T1

   fld tempm8
   fst st(1)
   fld b8
   fmul
   fadd T1
   fchs
   fst T1

{-------------------------------------}
{T2 değerinin hesaplanması}

   finit
   fld tempm1
   fst st(1)
   fld a1
   fmul
   fst T2

   fld tempm2
   fst st(1)
   fld a2
   fmul
   fadd T2
   fst T2

   fld tempm3
   fst st(1)
   fld a3
   fmul
   fadd T2
   fst T2

   fld tempm4
   fst st(1)
   fld a4
   fmul
   fadd T2
   fst T2

   finit
   fld tempm5
   fst st(1)
   fld a5
   fmul
   fadd T2
   fst T2

   fld tempm6
   fst st(1)
   fld a6
   fmul
   fadd T2
   fst T2

   fld tempm7
   fst st(1)
   fld a7
   fmul
   fadd T2
   fst T2

   fld tempm8
   fst st(1)
```

```
   fld a8
   fmul
   fadd T2
   fst T2

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

  mov ah,00
  mov dx,784
  mov al,15
  out dx,al

  mov dx,788
@11:in al,dx
  test al,01
  jz @11

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec dx
  in ax,dx
  shl ax,4
  add ax,temp

  sub ax,2048
  mov adc,ax

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm7
  fst tempm8
  fld tempm6
  fst tempm7
  fld tempm5
  fst tempm6
  fld tempm4
  fst tempm5
  fld tempm3
  fst tempm4
  fld tempm2
  fst tempm3
  fld tempm1
{-------------------------------------------}
  fst tempm2
  finit
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-------------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc

  mov ax,adc
  add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @12

  dec bx
  cmp bx,00
  jne @13

  mov ah,02
  int 16h
  and al,08
  cmp al,08
  je @15
  jmp @14

@15: end;
   end;
```

```
{*****************************************************************************}
```

{ Ana Program }

```
begin
ClrScr;
Gotoxy(28,12);
Write('ADC kanal numarasi(0-15)= ');
Read(kanal);

asm
mov dx,792    {ADC kanal seçimi}
mov al,kanal
out dx,al
end;

Assign(Dos1,'C:\matlab\bin\numlp.txt');
Assign(Dos2,'C:\matlab\bin\denlp.txt');
Assign(Dos3,'C:\matlab\bin\derece.txt');
Assign(Dos4,'C:\matlab\bin\tur.txt');

Reset(Dos1);
Reset(Dos2);
Reset(Dos3);
Reset(Dos4);

Read(Dos3,derece);
Read(Dos4,tur);

If tur=2 then derece:=derece*2; {Filtre
N.dereceden band geciren veya band söndüren
filtre ise 2N+1 tane katsayi gelecektir.}

{Filtre katsayılarını diskten okuma}

for n:=0 to derece do
begin
Read(Dos1,coeffa[n]);
end;

for n:=0 to derece do
begin
Read(Dos2,coeffb[n]);
end;

Close(Dos1);
Close(Dos2);
Close(Dos3);
Close(Dos4);

{Katsayıları gerekli değişkenlere atama}

a0:=coeffa[0];
a1:=coeffa[1];
a2:=coeffa[2];
a3:=coeffa[3];
a4:=coeffa[4];
a5:=coeffa[5];
a6:=coeffa[6];
a7:=coeffa[7];
a8:=coeffa[8];

b0:=coeffb[0];
b1:=coeffb[1];
b2:=coeffb[2];
b3:=coeffb[3];
b4:=coeffb[4];
b5:=coeffb[5];
b6:=coeffb[6];
b7:=coeffb[7];
b8:=coeffb[8];

ClrScr;
Gotoxy(16,12);
 Writeln('Programi Durdurmak icin "ALT"
Tusuna Basiniz....');

Case derece of
1:derece1;
2:derece2;
3:derece3;
4:derece4;
5:derece5;
6:derece6;
7:derece7;
8:derece8;
end;
end.
```

Ek3. Sayısal Filtrelerin İkinci Derece Modüllerle Paralel Gerçeklenmesi Programı

Program SAYISAL_FİLTRELERİN_PARALEL_GERÇEKLENMESİ;

{Beşir TAYFUR-Y.T.Ü 1997}

{Kullanılan sayısal filtre yapısı = 1D, Maksimum derece = 8}
{MPIBM4 kartının taban adresi 0300H = 768}

```
{*************************************************************************}

Uses Crt;
Var
Dos1:text;
Dos2:text;
Dos3:text;
Dos4:text;
Dos5:text;
Dos6:text;
Dos7:text;

derece:integer;
n:integer;
temp:word;
adc:word;
kanal:byte;
tur:integer;    {1-Alçak veya Yüksek Geçiren,
                 2-Band Geçiren veya Söndüren}

{Birinci ve ikinci derece için filtre katsayıları}
a0:double;
a1:double;
a2:double;

b0:double;
b1:double;
b2:double;

T1:double;
T2:double;

{Birinci ve ikinci derece için zaman gecikme
elemanları}
tempm1:double;  {m(k-1)}
tempm2:double;  {m(k-2)}

direkt:double;    {Direkt terim}

{Kısmi kesirlere açılımda reel residü ve kutba
sahip terimin katsayıları}
ga0:double;
gb1:double;

{Kısmi kesirlere açılımda kompleks eşlenik
residü ve kutuplara sahip terimlerin
oluşturacağı ikinci derece modüllerin
katsayıları}
ka10:double;
ka11:double;
kb11:double;
kb12:double;
ka20:double;
ka21:double;
kb21:double;
kb22:double;
ka30:double;
ka31:double;
kb31:double;
kb32:double;
ka40:double;
ka41:double;
kb41:double;
kb42:double;

T11:double;
T12:double;
T21:double;
T22:double;
T31:double;
T32:double;
T41:double;
T42:double;

output:double;

{Parale ikinci derece modüller için zaman
gecikme elemanları}
tempm11:double;  {m(k-1)}
tempm12:double;  {m(k-2)}
tempm21:double;  {m(k-1)}
tempm22:double;  {m(k-2)}
tempm31:double;  {m(k-1)}
tempm32:double;  {m(k-2)}
tempm41:double;  {m(k-1)}
tempm42:double;  {m(k-2)}

tam:integer;
coeffa:array[0..8] Of real;
coeffb:array[0..8] Of real;
```

```
gercel:array[1..2] Of real;                    komplex:array[1..16] Of real;

{*************************************************************************}
  Procedure derece1;

  begin
  asm

@14: mov bx,1
@13: mov cx,50000

{-------------------------------------------}
{T1 değerinin hesaplanması}

@12: finit                          {T1=-(b1.m(k-1)}
   fld tempm1
   fst st(1)
   fld b1
   fmul
   fchs
   fst T1

{-------------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm1                       {T2=a1.m(k-1)
   fst st(1)
   fld a1
   fmul
   fst T2

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

  mov ah,00
  mov dx,784
  mov al,kanal              {ADC kanalı seçimi}
  out dx,al

  mov dx,788
@11:in al,dx
  test al,01        { Dönüşüm tamamlandı mı ? }
  jz @11

  dec dx            {Dönüştürülen değeri okuma}
  in al,dx
  and al,240  {En düşün ağırlıklı 4-biti okuma
ve istenmeyen bitleri maskeleme}
  shr al,4
  mov temp,ax

  dec dx    {Yüksek anlamlı baytı okuma ve en
düşük anlanlı 4-bitle toplayıp 12-bit değer elde
etme}
  in ax,dx
  shl ax,4
  add ax,temp
  sub ax,2048       {Girişi işaretli hale getirme}
  mov adc,ax

{-------------------------------------------}
  finit                              {m(k)=x(k)+T1}
  fadd T1
  fild adc

  fst tempm1
  fst st(1)

{-------------------------------------------}
  fld a0                            {y(k)=a0.m(k)+T2}
  fmul
  fadd T2
  fist adc

  mov ax,adc
  add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776  {DAC A' ya en az ağırlıklı
bitleri gönderme }
  out dx,al
  inc dx        {DAC A' ya en ağırlıklı bitleri
gönderme }
  mov al,ah
  out dx,al
  mov dx,780              {DAC'yi güncelleme}
  mov al,00
  out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @12

  dec bx
  cmp bx,00
  jne @13

  mov ah,02
  int 16h
  and al,08        {ALT tuşu dışındaki tuşları
maskeleme}
  cmp al,08
  je @15
  jmp @14
```

```
@15: end;                                   end;

{*****************************************************************************}
```

Procedure derece2;

```
  begin
  asm

@14: mov bx,1
@13: mov cx,50000

{-----------------------------------------}
{T1 değerinin hesaplanması}

@12: finit
    fld tempm1
    fst st(1)
    fld b1
    fmul
    fst T1

    fld tempm2
    fst st(1)
    fld b2
    fmul
    fadd T1
    fchs
    fst T1

{-----------------------------------------}
{T2 değerinin hesaplanması}

    fld tempm1
    fst st(1)
    fld a1
    fmul
    fst T2

    fld tempm2
    fst st(1)
    fld a2
    fmul
    fadd T2
    fst T2

{-----------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

  mov ah,00
  mov dx,784
  mov al,kanal
  out dx,al

  mov dx,788
@11:in al,dx
  test al,01
  jz @11

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec dx
  in ax,dx
  shl ax,4
  add ax,temp
  sub ax,2048
  mov adc,ax

{-----------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm1              {m(k-2)→m(k-1)}
  fst tempm2

{-----------------------------------------}
  fild adc
  fadd T1
  fst tempm1
  fst st(1)

{-----------------------------------------}
  fld a0
  fmul
  fadd T2
  fist adc

  mov ax,adc
  add ax,2048

{-----------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

  mov dx,776
  out dx,al
  inc dx
  mov al,ah
  out dx,al
  mov dx,780
  mov al,00
  out dx,al
```

```
{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
  jne @l2

  dec bx
  cmp bx,00
  jne @l3

  mov ah,02
  int 16h
  and al,08
  cmp al,08
  je @l5
  jmp @l4

@l5: end;
   end;
```

{**********}

Procedure derece3;

```
  begin
  asm

@l4: mov bx,1
@l3: mov cx,50000

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

@l2: mov ah,00
   mov dx,784
   mov al,kanal
   out dx,al

   mov dx,788
@l1: in al,dx
   test al,01
   jz @l1

   dec dx
   in al,dx
   and al,240
   shr al,4
   mov temp,ax

   dec dx
   in ax,dx
   shl ax,4
   add ax,temp
   sub ax,2048
   mov adc,ax

{-------------------------------------------}
{Direkt terimle giriş çarpıldı}

   finit
   fild adc
   fst st(1)
   fld direkt
   fmul
   fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda reel rezidü ve kutba sahip terimden gelen ikinci derece modülün oluşturulması}

   fld tempm11   {T1=-b1.m(k-1) terimi oluşturuldu.T2=0}
   fst st(1)
   fld gb1
   fmul
   fchs
   fst T11

{-------------------------------------------}
   fild adc      {m(k)=x(k)+T1 oluşturuldu ve m(k) m(k-1)'e atandı.}
   fadd T11
   fst tempm11
   fst st(1)

{-------------------------------------------}
   fld ga0       {y(k)=a0.m(k) oluşturuldu.}
   fmul
   fadd output
   fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve kutuplara sahip terimlerden gelen ikinci derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   finit         {T1=-(b1.m(k-1)+b2.m(k-2))'i oluşturuldu.}
   fld tempm21
   fst st(1)
   fld kb11
   fmul
   fst T21
```

```
    fld tempm22
    fst st(1)
    fld kb12
    fmul
    fadd T21
    fchs
    fst T21

{-------------------------------------------}
{T2 değerinin hesaplanması}

    fld tempm21          {T2=a1.m(k-1) terimi
    oluşturuldu.(a2=0)}
    fst st(1)
    fld ka11
    fmul
    fst T22

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

    finit
    fld tempm21     {m(k-1) m(k-2)' e atandı.}
    fst tempm22

{-------------------------------------------}
    fild adc        {m(k)=x(k)+T1 oluşturuldu.}
    fadd T21
    fst tempm21          {m(k) değeri m(k-1)'e
atandı.}
    fst st(1)

{-------------------------------------------}
    fld ka10        {y(k)=a0.m(k)+T2 değeri
oluşturuldu.}
    fmul
    fadd T22
    fadd output          {Sonuc çıkış değeri
oluşturuldu.}
    fist adc
```

```
    mov ax,adc
    add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

    mov dx,776
    out dx,al
    inc dx
    mov al,ah
    out dx,al
    mov dx,780
    mov al,00
    out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

    dec cx
    cmp cx,00
    jne @12

    dec bx
    cmp bx,00
    jne @13

    mov ah,02
    int 16h
    and al,08
    cmp al,08
    je @15
    jmp @14

@15: end;
    end;
```

```
{*****************************************************************************}
```

Procedure derece4;

```
begin
  asm

@14: mov bx,1
@13:mov cx,50000

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

@12:mov ah,00
   mov dx,784
```

```
   mov al,kanal
   out dx,al

   mov dx,788
@11:in al,dx
   test al,01
   jz @11

   dec dx
   in al,dx
   and al,240
```

```
   shr al,4
   mov temp,ax

   dec dx
   in ax,dx
   shl ax,4
   add ax,temp
   sub ax,2048
   mov adc,ax

{-----------------------------------------
{Direkt terimle giriş çarpıldı}

   finit
   fild adc
   fst st(1)
   fld direkt
   fmul
   fst output

{-----------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 1. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   fld tempm11
   fst st(1)
   fld kb11
   fmul
   fst T11

   fld tempm12
   fst st(1)
   fld kb12
   fmul
   fadd T11
   fchs
   fst T11

{-----------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm11
   fst st(1)
   fld ka11
   fmul
   fst T12

{-----------------------------------------}

   finit
   fld tempm11
   fst tempm12

{-----------------------------------------}
   fild adc
   fadd T11
   fst tempm11
   fst st(1)

{-----------------------------------------}
   fld ka10
   fmul
   fadd T12
   fadd output
   fst output

{-----------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 2. ikinci
derece modülün oluşturulması}

   {T1 değerinin hesaplanması}

   finit
   fld tempm21
   fst st(1)
   fld kb21
   fmul
   fst T21

   fld tempm22
   fst st(1)
   fld kb22
   fmul
   fadd T21
   fchs
   fst T21

{-----------------------------------------}
   {T2 değerinin hesaplanması}

  fld tempm21
   fst st(1)
   fld ka21
   fmul
   fst T22

{-----------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

   finit
   fld tempm21
   fst tempm22

{-----------------------------------------}
   fild adc
   fadd T21
   fst tempm21
   fst st(1)

{-----------------------------------------}
```

```
    fld ka20
    fmul
    fadd T22
    fadd output
    fist adc

    mov ax,adc
    add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

   mov dx,776
   out dx,al
   inc dx
   mov al,ah
   out dx,al
   mov dx,780
   mov al,00
   out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

   dec cx
   cmp cx,00
   jne @12

   dec bx
   cmp bx,00
   jne @13

   mov ah,02
   int 16h
   and al,08
   cmp al,08
   je @15
   jmp @14

@15: end;
    end;
```

{**********}

Procedure derece5;

```
    begin

    asm

@14: mov bx,1
@13:mov cx,50000

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

@12: mov ah,00
    mov dx,784
    mov al,kanal
    out dx,al

    mov dx,788
@11: in al,dx
    test al,01
    jz @11

    dec dx
    in al,dx
    and al,240
    shr al,4
    mov temp,ax

    dec dx
    in ax,dx
    shl ax,4
    add ax,temp
    sub ax,2048
    mov adc,ax

{-------------------------------------------}
{Direkt terimle giriş çarpıldı}

    finit
    fild adc
    fst st(1)
    fld direkt
    fmul
    fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda reel rezidü ve kutba
sahip terimden gelen ikinci derece modülün
oluşturulması}

    fld tempm11
    fst st(1)
    fld gb1
    fmul
    fchs
    fst T11

{-------------------------------------------}
    fild adc
    fadd T11
    fst tempm11
    fst st(1)

{-------------------------------------------}
```

```
    fld ga0
    fmul
    fadd output
    fst output

{-------------------------------------------}
```

{Kısmi kesirlere açılımda kompleks rezidü ve kutuplara sahip terimlerden gelen 1. ikinci derece modülün oluşturulması}

{T1 değerinin hesaplanması}

```
    finit
    fld tempm21
    fst st(1)
    fld kb11
    fmul
    fst T21

    fld tempm22
    fst st(1)
    fld kb12
    fmul
    fadd T21
    fchs
    fst T21

{-------------------------------------------}
```

{T2 değerinin hesaplanması}

```
    fld tempm21
    fst st(1)
    fld ka11
    fmul
    fst T22

{-------------------------------------------}
```

{Zaman gecikme elemanları arasındaki atamalar}

```
    finit
    fld tempm21
    fst tempm22

{-------------------------------------------}
    fild adc
    fadd T21
    fst tempm21
    fst st(1)

{-------------------------------------------}
    fld ka10
    fmul
    fadd T22
    fadd output
    fst output

{-------------------------------------------}
```

{Kısmi kesirlere açılımda kompleks rezidü ve kutuplara sahip terimlerden gelen 2. ikinci derece modülün oluşturulması}

{T1 değerinin hesaplanması}

```
    finit
    fld tempm31
    fst st(1)
    fld kb21
    fmul
    fst T31

    fld tempm32
    fst st(1)
    fld kb22
    fmul
    fadd T31
    fchs
    fst T31

{-------------------------------------------}
```

{T2 değerinin hesaplanması}

```
    fld tempm31
    fst st(1)
    fld ka21
    fmul
    fst T32

{-------------------------------------------}
```

{Zaman gecikme elemanları arasındaki atamalar}

```
    finit
    fld tempm31
    fst tempm32

{-------------------------------------------}
    fild adc
    fadd T31
    fst tempm31
    fst st(1)

{-------------------------------------------}
    fld ka20
    fmul
    fadd T32
    fadd output
    fist adc

   mov ax,adc
   add ax,2048

{-------------------------------------------}
```

```
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

   mov dx,776
   out dx,al
   inc dx
   mov al,ah
   out dx,al
   mov dx,780
   mov al,00
   out dx,al

{-----------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

  dec cx
  cmp cx,00
   jne @l2

   dec bx
   cmp bx,00
   jne @l3

   mov ah,02
   int 16h
   and al,08
   cmp al,08
   je @l5
   jmp @l4

@l5: end;
     end;
```

{**********}

Procedure derece6;

```
   begin
   asm

@l4: mov bx,1
@l3:mov cx,50000

{-----------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

@l2:mov ah,00
   mov dx,784
   mov al,kanal
   out dx,al

   mov dx,788
@l1:in al,dx
   test al,01
   jz @l1

   dec dx
   in al,dx
   and al,240
   shr al,4
   mov temp,ax

   dec dx
   in ax,dx
   shl ax,4
   add ax,temp
   sub ax,2048
   mov adc,ax

{-----------------------------------------}
{Direkt terimle giriş çarpıldı}

   finit
   fild adc
   fst st(1)
   fld direkt
   fmul
   fst output

{-----------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 1. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   fld tempm11
   fst st(1)
   fld kb11
   fmul
   fst T11

   fld tempm12
   fst st(1)
   fld kb12
   fmul
   fadd T11
   fchs
   fst T11

{-----------------------------------------}
{T2 değerinin hesaplanması}
```

```
   fld tempm11
   fst st(1)
   fld ka11
   fmul
   fst T12

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

   finit
   fld tempm11
   fst tempm12

{-------------------------------------------}
   fild adc
   fadd T11
   fst tempm11
   fst st(1)

{-------------------------------------------}
   fld ka10
   fmul
   fadd T12
   fadd output
   fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 2. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   finit
   fld tempm21
   fst st(1)
   fld kb21
   fmul
   fst T21

   fld tempm22
   fst st(1)
   fld kb22
   fmul
   fadd T21
   fchs
   fst T21

{-------------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm21
   fst st(1)
   fld ka21
   fmul
   fst T22

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

   finit
   fld tempm21
   fst tempm22

{-------------------------------------------}
   fild adc
   fadd T21
   fst tempm21
   fst st(1)

{-------------------------------------------}
   fld ka20
   fmul
   fadd T22
   fadd output
   fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 1. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   finit
   fld tempm31
   fst st(1)
   fld kb31
   fmul
   fst T31

   fld tempm32
   fst st(1)
   fld kb32
   fmul
   fadd T31
   fchs
   fst T31

{-------------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm31
   fst st(1)
   fld ka31
   fmul
   fst T32

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}
```

```
    finit
    fld tempm31
    fst tempm32

{-------------------------------------------}
    fild adc
    fadd T31
    fst tempm31
    fst st(1)

{-------------------------------------------}
    fld ka30
    fmul
    fadd T32
    fadd output
    fist adc

    mov ax,adc
    add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

   mov dx,776
   out dx,al
   inc dx
   mov al,ah
   out dx,al

{*******************************************************************************}
```

Procedure derece7;

```
    begin
    asm

@14: mov bx,1
@13: mov cx,50000

{-------------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

@12: mov ah,00
    mov dx,784
    mov al,kanal
    out dx,al

    mov dx,788
@11: in al,dx
    test al,01
    jz @11

    dec dx
    in al,dx
    and al,240
    shr al,4

    mov dx,780
    mov al,00
    out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

    dec cx
    cmp cx,00
    jne @12

    dec bx
    cmp bx,00
    jne @13

    mov ah,02
    int 16h
    and al,08
    cmp al,08
    je @15
    jmp @14

@15: end;
    end;
```

```
    mov temp,ax

    dec dx
    in ax,dx
    shl ax,4
    add ax,temp
    sub ax,2048
    mov adc,ax

{-------------------------------------------}
{Direkt terimle giriş çarpıldı}

    finit
    fild adc
    fst st(1)
    fld direkt
    fmul
    fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda reel rezidü ve kutba
sahip terimden gelen ikinci derece modülün
oluşturulması}
```

```
    fld tempm11
    fst st(1)
    fld gb1
    fmul
    fchs
    fst T11

{-------------------------------------------}
    fild adc
    fadd T11
    fst tempm11
    fst st(1)

{-------------------------------------------}
    fld ga0
    fmul
    fadd output
    fst output

{-------------------------------------------}
```

{Kısmi kesirlere açılımda kompleks rezidü ve kutuplara sahip terimlerden gelen 1. ikinci derece modülün oluşturulması}

{T1 değerinin hesaplanması}

```
    finit
    fld tempm21
    fst st(1)
    fld kb11
    fmul
    fst T21

    fld tempm22
    fst st(1)
    fld kb12
    fmul
    fadd T21
    fchs
    fst T21

{-------------------------------------------}
```

{T2 değerinin hesaplanması}

```
    fld tempm21
    fst st(1)
    fld ka11
    fmul
    fst T22

{-------------------------------------------}
```

{Zaman gecikme elemanları arasındaki atamalar}

```
    finit
    fld tempm21
    fst tempm22

{-------------------------------------------}
    fild adc
    fadd T21
    fst tempm21
    fst st(1)

{-------------------------------------------}
    fld ka10
    fmul
    fadd T22
    fadd output
    fst output

{-------------------------------------------}
```

{Kısmi kesirlere açılımda kompleks rezidü ve kutuplara sahip terimlerden gelen 2. ikinci derece modülün oluşturulması}

{T1 değerinin hesaplanması}

```
    finit
    fld tempm31
    fst st(1)
    fld kb21
    fmul
    fst T31

    fld tempm32
    fst st(1)
    fld kb22
    fmul
    fadd T31
    fchs
    fst T31

{-------------------------------------------}
```

{T2 değerinin hesaplanması}

```
    fld tempm31
    fst st(1)
    fld ka21
    fmul
    fst T32

{-------------------------------------------}
```

{Zaman gecikme elemanları arasındaki atamalar}

```
    finit
    fld tempm31
    fst tempm32

{-------------------------------------------}
    fild adc
    fadd T31
    fst tempm31
```

```
   fst st(1)

{-------------------------------------}
   fld ka20
   fmul
   fadd T32
   fadd output
   fst output

{-------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 3. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   finit
   fld tempm41
   fst st(1)
   fld kb31
   fmul
   fst T41

   fld tempm42
   fst st(1)
   fld kb32
   fmul
   fadd T41
   fchs
   fst T41

{-------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm41
   fst st(1)
   fld ka31
   fmul
   fst T42

{-------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

   finit
   fld tempm41
   fst tempm42

{-------------------------------------}
   fild adc
   fadd T41
   fst tempm41
   fst st(1)

{-------------------------------------}
   fld ka30
   fmul
   fadd T42
   fadd output
   fist adc

   mov ax,adc
   add ax,2048

{-------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

   mov dx,776
   out dx,al
   inc dx
   mov al,ah
   out dx,al
   mov dx,780
   mov al,00
   out dx,al

{-------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

   dec cx
   cmp cx,00
   jne @12

   dec bx
   cmp bx,00
   jne @13

   mov ah,02
   int 16h
   and al,08
   cmp al,08
   je @15
   jmp @14

@15: end;
   end;

{*************************************************************************}
```

Procedure derece8;

```
  begin
  asm

@l4: mov bx,1
@l3:mov cx,50000

{-----------------------------------------}
{Giriş sinyalinden 12-bit örnek alınması}

@l2:mov ah,00
  mov dx,784
  mov al,kanal
  out dx,al

  mov dx,788
@l1:in al,dx
  test al,01
  jz @l1

  dec dx
  in al,dx
  and al,240
  shr al,4
  mov temp,ax

  dec dx
  in ax,dx
  shl ax,4
  add ax,temp
  sub ax,2048
  mov adc,ax

{-----------------------------------------}
{Direkt terimle giriş çarpıldı}

  finit
  fild adc
  fst st(1)
  fld direkt
  fmul
  fst output

{-----------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 1. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

  fld tempm11
  fst st(1)
  fld kb11
  fmul
  fst T11

  fld tempm12
  fst st(1)
  fld kb12
  fmul
  fadd T11
  fchs
  fst T11

{-----------------------------------------}
{T2 değerinin hesaplanması}

  fld tempm11
  fst st(1)
  fld ka11
  fmul
  fst T12

{-----------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

  finit
  fld tempm11
  fst tempm12

{-----------------------------------------}
  fild adc
  fadd T11
  fst tempm11
  fst st(1)

{-----------------------------------------}
  fld ka10
  fmul
  fadd T12
  fadd output
  fst output

{-----------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 2. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

  finit
  fld tempm21
  fst st(1)
  fld kb21
  fmul
  fst T21

  fld tempm22
  fst st(1)
  fld kb22
```

```
   fmul
   fadd T21
   fchs
   fst T21

{-------------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm21
   fst st(1)
   fld ka21
   fmul
   fst T22

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

   finit
   fld tempm21
   fst tempm22

{-------------------------------------------}
   fild adc
   fadd T21
   fst tempm21
   fst st(1)

{-------------------------------------------}
   fld ka20
   fmul
   fadd T22
   fadd output
   fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 3. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   finit
   fld tempm31
   fst st(1)
   fld kb31
   fmul
   fst T31

   fld tempm32
   fst st(1)
   fld kb32
   fmul
   fadd T31
   fchs
   fst T31

{-------------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm31
   fst st(1)
   fld ka31
   fmul
   fst T32

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

   finit
   fld tempm31
   fst tempm32

{-------------------------------------------}
   fild adc
   fadd T31
   fst tempm31
   fst st(1)

{-------------------------------------------}
   fld ka30
   fmul
   fadd T32
   fadd output
   fst output

{-------------------------------------------}
{Kısmi kesirlere açılımda kompleks rezidü ve
kutuplara sahip terimlerden gelen 4. ikinci
derece modülün oluşturulması}

{T1 değerinin hesaplanması}

   finit
   fld tempm41
   fst st(1)
   fld kb41
   fmul
   fst T41

   fld tempm42
   fst st(1)
   fld kb42
   fmul
   fadd T41
   fchs
   fst T41

{-------------------------------------------}
{T2 değerinin hesaplanması}

   fld tempm41
   fst st(1)
```

```
    fld ka41
    fmul
    fst T42

{-------------------------------------------}
{Zaman gecikme elemanları arasındaki
atamalar}

    finit
    fld tempm41
    fst tempm42
{-------------------------------------------}

    fild adc
    fadd T41
    fst tempm41
    fst st(1)

{-------------------------------------------}
    fld ka40
    fmul
    fadd T42
    fadd output
    fist adc

    mov ax,adc
    add ax,2048

{-------------------------------------------}
{Elde edilen sonucun DAC ile dış ortama
gönderilmesi}

    mov dx,776
    out dx,al
    inc dx
    mov al,ah
    out dx,al
    mov dx,780
    mov al,00
    out dx,al

{-------------------------------------------}
{Belirli aralıklarla klavyenin ALT tuşuna
basılıp basılmadığını test etme}

    dec cx
    cmp cx,00
    jne @l2

    dec bx
    cmp bx,00
    jne @l3


    mov ah,02
    int 16h
    and al,08
    cmp al,08
    je @l5
    jmp @l4

@l5: end;
    end;
```

{**}

{Ana Program }

```
begin
  ClrScr;
  Gotoxy(28,12);
  Write('ADC kanal numarasi(0-15)= ');
  Read(kanal);

  asm
  mov dx,792    {ADC kanal seçimi}
  mov al,kanal
  out dx,al
  end;

  Assign(Dos3,'C:\matlab\bin\derece.txt');
  Assign(Dos7,'C:\matlab\bin\tur.txt');
  Reset(Dos3);
  Reset(Dos7);
  Read(Dos3,derece);
  Read(Dos7,tur);
  Close(Dos3);
  Close(Dos7);


  if tur=2 then derece:=2*derece;

  if derece<=2 then

{Birinci ve ikinci derece için katsayıları
okuma}

  begin
  Assign(Dos1,'C:\matlab\bin\numlp.txt');
  Assign(Dos2,'C:\matlab\bin\denlp.txt');
  Reset(Dos1);
  Reset(Dos2);
```

```
for n:=0 to derece do
begin
Read(Dos1,coeffa[n]);
end;

for n:=0 to derece do
begin
Read(Dos2,coeffb[n]);
end;

Close(Dos1);
Close(Dos2);

a0:=coeffa[0];
a1:=coeffa[1];
a2:=coeffa[2];
b0:=coeffb[0];
b1:=coeffb[1];
b2:=coeffb[2];

ClrScr;
Gotoxy(16,12);
Writeln('Programi Durdurmak icin "ALT"
Tusuna Basiniz....');

Case derece of
1:derece1;
2:derece2;
end;
end

else
```

{Paralel gerçekleme için direkt terimi ve ikinci derece modüllerin katsayılarını okuma (ikiden büyük dereceler için)}

```
begin
Assign(Dos4,'C:\matlab\bin\direkt.txt');
Assign(Dos5,'C:\matlab\bin\realkok.txt');
Assign(Dos6,'C:\matlab\bin\komplex.txt');
Reset(Dos4);
Reset(Dos5);
Reset(Dos6);

Read(Dos4,direkt);

for n:=1 to 2 do
begin
Read(Dos5,gercel[n]);
end;
```

{Reel residü ve kutba sahip terimin oluşturduğu ikinci derece modülün katsayilari}

```
ga0:=gercel[1];
gb1:=gercel[2];

tam:=derece div 2; {Kompleks eşlenik residü
ve kutuplara sahip terimlerin oluşturduğu
ikinci derece modül sayısı}

for n:=1 to tam*4 do
begin
Read(Dos6,komplex[n]);
end;
```

{Kompleks eşlenik residü ve kutuplara sahip terimlerin oluşturduğu 1.ikinci derece modülün katsayıları}

```
ka10:=komplex[1];
ka11:=komplex[2];
kb11:=komplex[3];
kb12:=komplex[4];
```

{Kompleks eşlenik residü ve kutuplara sahip terimlerin oluşturduğu 2.ikinci derece modülün katsayıları}

```
ka20:=komplex[5];
ka21:=komplex[6];
kb21:=komplex[7];
kb22:=komplex[8];
```

{Kompleks eşlenik residü ve kutuplara sahip terimlerin oluşturduğu 3.ikinci derece modülün katsayıları}

```
ka30:=komplex[9];
ka31:=komplex[10];
kb31:=komplex[11];
kb32:=komplex[12];
```

{Kompleks eşlenik residü ve kutuplara sahip terimlerin oluşturduğu 4.ikinci derece modülün katsayıları}

```
ka40:=komplex[13];
ka41:=komplex[14];
kb41:=komplex[15];
kb42:=komplex[16];

ClrScr;
Gotoxy(16,12);
Writeln('Programi Durdurmak icin "ALT"
Tuşuna Basınız....');

Case derece of
3:derece3;
4:derece4;
5:derece5;
6:derece6;
```

```
7:derece7;
8:derece8;
end;
```

```
end;
end.
```

ÖZGEÇMİŞ

Doğum Tarihi	: 17 Nisan 1972	
Doğum Yeri	: Bayburt	
Eğitim	: 1987-1990	Bağcılar Lisesi, İstanbul
	1990-1994	Y.T.Ü Elektrik-Elektronik Fakültesi Elektronik ve Haberleşme Müh. Böl.
Mesleki Geçmiş	: 1995-	Araştırma Görevlisi Y.T.Ü Elektrik-Elektronik Fakültesi Elektronik ve Haberleşme Müh. Böl Devreler ve Sistemler Anabilim Dalı
İlgi Alanları	: Bilgisayar donanımı ve yazılımı, programlama dillleri, mikroişlemciler ve mikroişlemcili sistemler	