



**PROGRAMLANABİLİR KAPI DİZİLERİ
ÜZERİNDE FİLTRE TASARIM YÖNTEMLERİ**

**Yüksek Lisans Tezi
Aybike ÜSTÜNDAĞ
161113104**

**Anabilim Dalı: Elektrik-Elektronik Mühendisliği
Programı: Elektronik A.B.D
Danışmanı: Prof. Dr. Mustafa TÜRK**

TEMMUZ-2019

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PROGRAMLANABİLİR KAPI DİZİLERİ ÜZERİNDE
FİLTRE TASARIM YÖNTEMLERİ

YÜKSEK LİSANS TEZİ

Aybike ÜSTÜNDAĞ

(161113104)

Tezin Enstitüye Verildiği Tarih: 11 Haziran 2019

Tezin Savunulduğu Tarih: 11 Temmuz 2019

Tez Danışmanı: Prof. Dr. Mustafa TÜRK (F.Ü)
Diğer Jüri Üyeleri: Prof. Dr. Arif GÜLTEN (F.Ü)

Dr. Öğr. Üyesi Mehmet ÜSTÜNDAĞ (B.Ü)

TEMMUZ-2019

ÖNSÖZ

Bu yüksek lisans tez çalışmasında filtre tasarım yöntemleri hakkında bilgi verilip örnek bir filtre tasarımı yapılmıştır. Tez çalışmam sırasında değerli bilgi ve tecrübeleriyle bana yol gösteren hocalarım Prof. Dr. Mustafa TRK'e, Prof. Dr. Arif GLTEN'e, desteklerini benden esirgemeyen aileme ve arkadaşım Fatma ÇULCU'ya teşekkürlerimi sunarım.

Aybike STNDAĖ
ELAZIĖ-2019

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	I
İÇİNDEKİLER	II
ÖZET	IV
SUMMARY	V
SEMBOLLER LİSTESİ	VI
KISALTMALAR LİSTESİ.....	VII
ŞEKİLLER LİSTESİ	VIII
TABLolar LİSTESİ	IX
1. GİRİŞ	1
2.FİLTRELERİN SINIFLANDIRILMASI.....	3
2.1. Filtre Parametreleri	3
2.2. Analog Filtreler.....	5
2.2.1. Alçak Geçiren Filtre	6
2.2.2. Yüksek Geçiren Filtre.....	6
2.2.3. Bant Geçiren Filtre	7
2.2.4. Bant Durduran Filtre.....	9
2.2.5. Tüm Geçiren Filtre	9
2.3. Dijital Filtreler	11
2.3.1. FIR Filtre	12
2.3.1.1. Birim Dürtü Yanıtı.....	13
2.3.1.2. Evrişim ve FIR Filtreler.....	14
2.3.1.3. FIR Filtre Uygulanmasında Kullanılan Blok Yapıları.....	14
2.3.1.4. FIR Filtrelerin Doğrusallık, Zamanla Değişmezlik ve Lineerliği	18
2.3.1.5. Evrişim Toplamının Türetilmesi.....	19
2.3.1.6. FIR Filtrelerin frekans cevabı	20
2.3.1.7. FIR Sistemlerinin Sinüzoidal Tepkisi.....	20
2.3.1.8. Kararlı Durum ve Geçici Yanıt.....	21
2.3.1.9. Z-Dönüşümü ve Doğrusal Sistemler	22
2.3.1.10. Z-Düzlemi ve Birim Çember	24
2.3.2.IIR Filtre	25

2.3.2.1. Zaman Domeni Cevabı	26
2.3.2.2. Birinci Dereceden IIR Sisteminin Dürtü Cevabı	26
2.3.2.3. IIR Filtrelerin Doğrusallığı ve Zaman Değişmezliği	27
2.3.2.4. Birinci Mertebeden Bir Özyinelemeli Sistemin Adım Cevabı	28
2.3.2.5. Bir IIR Filtresinin Sistem Fonksiyonu	29
2.3.2.6. Sistem Fonksiyonları ve Blok Diyagram Yapıları	30
2.3.2.7. Kutuplar ve Sıfırlar	33
2.3.2.8. Sistem Kararlılığı	34
3. XSG VE MATLAB SIMULINK İLE FIR FİLTRE TASARIMI	35
3.1. Xilinx System Generator	35
3.2. Tasarımı Uygulama Aşamaları	37
4. SONUÇLAR	44
5. KAYNAKLAR	45
6. EKLER	49
ÖZGEÇMİŞ	51

ÖZET

Günümüzde birçok alanda elektronik sistemlerden yararlanılmaktadır. Elektronik sistemler birçok karmaşık sinyal içerisinde kendi çalışma sistemine uygun sinyali seçerek çalışmaktadır. Karmaşık sinyaller içerisinde diğer sinyallerin oluşturduğu gürültüden arındırılarak sisteme uygun olan sinyali elde etme işlemi filtreler aracılığıyla yapılmaktadır. Pratikte sinyal net bir şekilde süzülemez. Bu nedenle sinyali en az gürültüyle süzebilmek için uygun filtre tasarımı yapılmalıdır. Çok çeşitli filtre tasarım teknikleri mevcuttur. Bu tez çalışmasında analog ve dijital filtre tasarım teknikleri ele alınıp bu konuda teorik bilgi verilmiştir. Xilinx System Generatör ve MATLAB Simulink programlarıyla farklı frekans ve genlikteki sinyallerin toplamından oluşan sinyal karmaşasından istenmeyen frekanstaki sinyali bastırmak için bant durduran filtre tasarımı yapılmış ve elde edilen sonuçlar irdelenmiştir.

Anahtar Kelimeler: Analog Filtre, Dijital Filtre, FPGA (Alanda Programlanabilir Kapı Dizileri), Matlab/Simulink, Xilinx System Generator, DSP (Sayısal Sinyal İşleme)

SUMMARY

Filter Design Methods on FPGA

Today, electronic systems are used in many areas. Electronic systems work by selecting the signal that is appropriate to their operating system from many complex signals. The process of obtaining the signal which is suitable for the system is made by filters, by eliminating the noise generated by other signals within the complex signals. In practice, the signal cannot be clearly filtered. For this reason, the filter should be designed in order to filter the signal with minimum noise. A wide variety of filter design techniques are available. In this thesis, analog and digital filter design techniques are discussed and theoretical information is given. With the Xilinx System Generator and MATLAB Simulink programs, a band stop filter design has been designed to suppress the signal at the unwanted frequency from the signal chaos composed of the signals of different frequencies and amplitudes and the results obtained are examined.



Key Words: Analog Filter, Digital Filter, FPGA (Field Programmable Gate Arrays), Matlab/Simulink, Xilinx System Generator, DSP (Digital Signal Processing)

SEMBOLLER LİSTESİ

kHz: Kilohertz

dB: Desibel

MB: Megabayt

f : Frekans

f_c : Kesim frekansı

A_{max} : Geçirme bandının dalgalanması

m : Filtrenin derecesi

A_{min} : Minimum geçirme bandı zayıflaması

f_s : Durdurma bandı frekansı

ξ : Sönümleme oranı

Q : Filtrenin kalite faktörüdür.

f_l : Alt kesim frekansı

f_h : Üst kesim frekansı

KISALTMALAR LİSTESİ

FPGA: Alan Programlanabilir Kapı Dizisi

ASIC: Uygulamaya Özel Entegre Devre

MAC: Ortam Erişim Kontrolü

FIR: Sonlu Darbe Tepki

IIR: Sonsuz Darbe Tepki

DSP: Sayısal Sinyal İşleme

LTI: Doğrusal zamanla değişmeyen

SSB-SC: Taşıyıcısı bastırılmış tek yan bant

H_{AP} : Tüm geçiren filtre transfer fonksiyonu

H_{LP} : Alçak geçiren filtre transfer fonksiyonu

H_{BP} : Bant geçiren filtre transfer fonksiyonu

H_{HP} :Yüksek geçiren filtre transfer fonksiyonu

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1. Filtre parametreleri	3
Şekil 2.2. İdeal Alçak Geçiren Filtrenin Genlik-Frekans Değişimi	6
Şekil 2.3. İdeal Yüksek Geçiren Filtrenin Genlik-Frekans Değişimi.....	7
Şekil 2.4. İdeal Bant Geçiren Filtrenin Genlik-Frekans Değişimi	8
Şekil 2.5. İdeal Bant Durduran Filtrenin Genlik-Frekans Değişimi.....	9
Şekil 2.6. FIR filtrenin dönüştürülmüş formu	12
Şekil 2.7. Birim dürtü yanıtı dizisi	14
Şekil 2.8. FIR filtre uygulamasında kullanılan blok yapıları	15
Şekil 2.9. Doğrudan Form FIR filtre blok şeması	16
Şekil 2.10. Dönüştürülmüş blok diyagram	17
Şekil 2.11. Birim çember.....	24
Şekil 2.12. Kutup ve sıfır noktaları.....	25
Şekil 2.13. Doğrudan Form I yapısı	26
Şekil 2.14. Doğrudan Form II yapısı	30
Şekil 2.15. Transpoze Doğrudan Form II	31
Şekil 2.16. Kutup ve sıfır noktaları.....	33
Şekil 3.1. FIR filtre uygulamasının tasarım akış şeması	37
Şekil 3.2. Doğrudan Form FIR filtre blok yapısının Simulink modeli.....	38
Şekil 3.3. Simulink'te Elde Edilen Giriş Sinyali - Genlik Grafiği	38
Şekil 3.4. Simulink'te Elde Edilen Çıkış Sinyali - Genlik Grafiği.....	39
Şekil 3.5. Doğrudan Form FIR filtre blok yapısının System Generator modeli.....	40
Şekil 3.6. FPGA kartı ile bilgisayar bağlantısı	40
Şekil 3.7. System Generator blok penceresi	41
Şekil 3.8. Xilinx System Generator'de Elde Edilen Giriş Sinyali - Genlik Grafiği	41
Şekil 3.9. Xilinx System Generator'de Elde Edilen Çıkış Sinyali - Genlik Grafiği.	42
Şekil 3.10. Orijinal Sinyal ve XSG'de Elde Edilen Çıkış Sinyali Fark Grafiği.....	42

TABLÖLAR LİSTESİ

Sayfa No

Tablo 2.1. Filtre tiplerinin genlik cevabı ve transfer fonksiyonu	10
---	----



1. GİRİŞ

Filtrelere özellikle, sinyal ve görüntü işleme alanlarında fazlasıyla ihtiyaç duyulur. Ayrıca sayısal sistemlerdeki hızlı ilerlemeler filtrelerin kullanımını genişletmiştir. Bu nedenle, filtreleri ve katsayılarının hesaplanmasını anlamak ve öğrenmek çok önemlidir. Bu çalışmada, filtre tasarım teknikleri hakkında teorik bilgi verilip örnek bir filtrenin Xilinx System Generatör ve MATLAB/Simulink çalışma ortamlarında tasarımı yapılmıştır.

Filtreleme, sinyal ve görüntü işleme dahil olmak üzere elektrik ve elektronik alanında temel bir süreçtir. İstenilen frekansların seçilmesi, istenmeyen seslerin bastırılması vb. için bu alanda analog ve dijital filtreler kullanılmaktadır. Sayısal elektronikte hızlı ilerlemeler, dijital filtrelerin kullanımını ve avantajlarını arttırmıştır. Dijital filtrelerin kullanımının avantajları arasında, düşük maliyetli olma, filtre parametrelerinin veya yapılarının sadece yazılım vb. kullanılarak ayarlanması veya değiştirilmesi sayılabilir. Bu nedenle, Sonlu Dürtü Tepkisi (FIR) ve Sonsuz Darbe Tepkisi (IIR) filtreleri, genel uygulama alanlarında çokça kullanılmaktadır.

Bugün, FIR veya IIR filtre katsayılarını elde etmek için birçok yöntem mevcuttur. İdeal frekans yanıtları elde etmek için yüksek dereceli filtreler tasarlamak gereklidir. Öte yandan, filtrelerin derecesini arttırmak tasarım süreçlerinde birçok zorluğu beraberinde getirir [1].

Tüm sinyal işleme ve telekomünikasyon sistemlerinde filtreler temel bir bileşendir. Filtreler iletişim sistemlerinde ve biyomedikal gibi sinyal işleme, kanal eşitleme, gürültü azaltma, ses işleme, radar, video işleme, ekonomik ve finansal verilerin analizi gibi uygulamalarında yaygın olarak kullanılmaktadır. Örneğin bir radyo alıcısı istasyon arayıcıları veya bant geçiren filtreler, radyo kanalından sinyalleri çıkarmak için kullanılır.

Dijital filtrelerin Sayısal Sinyal İşleme (DSP) uygulamalarında büyük rolü vardır. Analog filtreler ile kıyaslandığında görüntü işleme, ses işleme, veri sıkıştırma, vb. uygulamaların çoğunda tercih edilirler. Dijital filtreler Sonlu Darbe Tepkisi (FIR) ve Sonsuz Darbe Tepkisi (IIR) olarak iki kategoriye ayrılır. FIR filtreleri, doğrusal faz ve sistem kararlılığı sağlamanın bütünlüğü için görüntüleme, iletişim vb. çeşitli dijital sinyal işleme alanlarında yaygın olarak uygulanır [2].

Filtreler, sinyalleri frekansa bağımlı bir şekilde işlerler. Basit bir filtre, kapasitörlerin ve indüktörlerin empedansının frekansa bağlı yapısı incelenerek açıklanabilir. Şönt

bacağının reaktif bir empedans olduđu bir voltaj bölücüde frekans değıştikçe, reaktif empedansın değeri değışir ve voltaj bölücü oranı değışir. Bu mekanizma, frekans cevabı olarak tanımlanan giriş / çıkış transfer fonksiyonundaki frekans bağımlı değışimini verir.

Filtreler birçok pratik uygulamaya sahiptir. Basit, tek kutuplu, alçak geçiren bir filtre genellikle aşırı faz kaymasının salınımlara neden olabileceğı yüksek frekanslardaki kazancı yuvarlayarak amplifikatörleri dengelemek için kullanılır. Yüksek kazançlı amplifikatörlerde veya tek besleme devrelerinde dc bileşenini bloke etmek için basit, tek kutuplu, yüksek geçiren bir filtre kullanılabilir. Filtreler sinyalleri ayırmak, ilgilenilenleri geçmek ve istenmeyen frekansları filtrelemek için kullanılabilir.

Bunun bir örneğı, işlem yapmak istediğiniz sinyalin, sinyallerin geri kalan kısmını zayıflatırken, tipik olarak kazanç ile geçirildiğı bir radyo alıcısıdır. Veri dönüşümde A / D sistemlerinde örtüşmelerin etkilerini ortadan kaldırmak için filtreler de kullanılır. Bunlar, bir D/A çıkışındaki sinyalin yeniden oluşturulmasında, örnekleme frekansı ve harmonikleri gibi daha yüksek frekanslı komponentlerin elimine edilmesinde ve böylece dalga formunun yumuşatılmasında kullanılırlar [3].

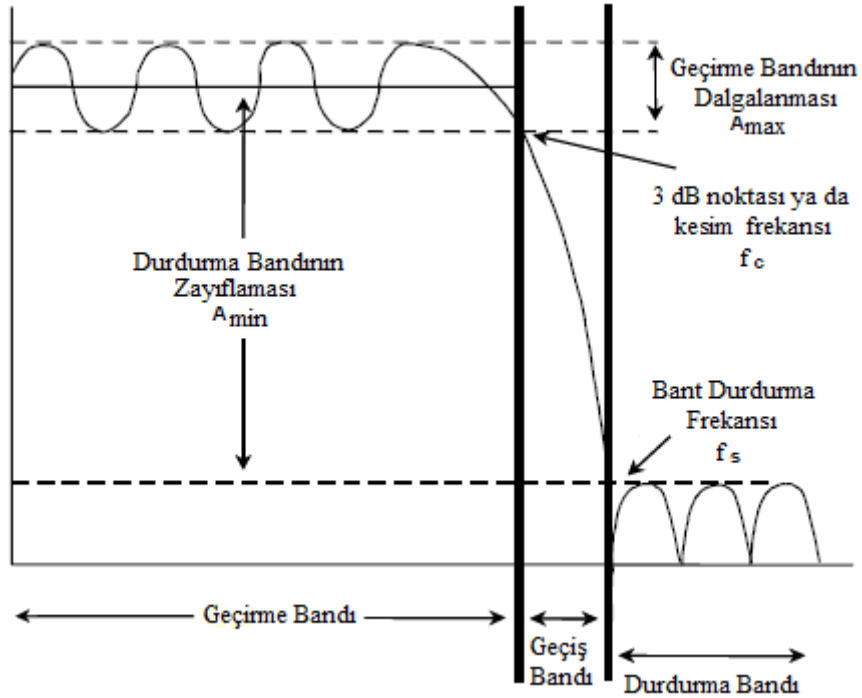
Bu tezin amacı Xilinx System Generatör ve MATLAB/Simulink programlarını kullanarak bir sinyal karmaşasından istenilen frekanstaki sinyali filtreleyip elde edilen sonuçları irdelemektir. Farklı genlik ve frekanstaki iki sinyal toplanıp istenilen sinyali süzmek için bir FIR filtre tasarımı gerçekleştirilmiş ve filtre parametrelerine göre katsayılar elde edilip bu katsayılar doğrultusunda her iki yöntem ile FIR filtre tasarımı yapılmıştır.

Tezin birinci bölümünde tezin içeriğı ve tezde kullanılacak program hakkında genel bilgiler verilmiştir. İkinci bölümünde bölümünde analog ve dijital filtre çeşitleri tanıtılarak avantaj ve dezavantajlarından bahsedilmiştir. Üçüncü bölümünde Xilinx System Generatör ve MATLAB/Simulink çalışma ortamında tasarlanan FIR filtrenin tasarım aşamaları anlatılıp dördüncü bölümünde elde edilen sonuçlar irdelenmiştir.

2. FİLTRELERİN SINIFLANDIRILMASI

2.1.Filtre Parametreleri

Pratik bir filtrenin beş parametresi Şekil 2.1’de görülmektedir. Kesim frekansı (f_c), filtre yanıtının hata bandını terk ettiği frekanstır. Durdurma bandı frekansı (f_s), durdurma bandındaki minimum zayıflamanın ulaştığı frekanstır. Geçirme bandının dalgalanması (A_{max}), geçirme bandı cevabındaki hata bandıdır. Minimum geçirme bandı zayıflaması (A_{min}), durdurma bandı içindeki minimum sinyal zayıflaması olarak tanımlanır. Filtrenin yüksekliği, filtrenin derecesi m olarak tanımlanır. Ayrıca m transfer fonksiyonundaki kutup sayısıdır. Transfer fonksiyonunun paydasının kökleri kutuplar olarak tanımlanır. Transfer fonksiyonunun payının kökleri ise sıfırlar olarak bilinir.



Şekil 2.1. Filtre parametreleri

Analog filtreler, bir kapasitörün veya bir indüktörün empedansı frekans ile değiştiği için frekansa bağlı bir cevaba sahiptir.

$$Z_L = s L \quad (2.1)$$

$$Z_C = \frac{1}{s C} \quad (2.2)$$

Bu nedenle karmaşık empedanslar Denklem (2.1) ile Denklem (2.2) sırasıyla bir endüktör ve bir kapasitörün empedansını tanımlamak için kullanılır.

Standart devre analizi teknikleri kullanılarak, filtrenin transfer fonksiyonu elde edilebilir. Bu teknikler Ohm yasasını, Kirchhoff'un gerilim, akım yasaları ile süperpozisyonu içerir. Empedansların karmaşık olduğu bilindiğine göre transfer fonksiyonunun genel formu Denklem (2.3)'te görülmektedir. Bu nedenle, $H(s)$, pay için m ve payda için n dereceli gerçek katsayılı S 'in bir rasyonel fonksiyonudur.

$$H(s) = \frac{a_m s^m + a_{m-1} s^{m-1} + \dots + a_1 s + a_0}{b_n s^n + b_{n-1} s^{n-1} + \dots + b_1 s + b_0} \quad (2.3)$$

Denklemin kökleri çözülürken, devrenin kutupları payda, sıfırları ise pay tarafından belirlenir. Her kutup bir -6 dB / octave veya -20 dB / decade yanıtı sağlayacaktır. Her sıfır ise bir $+6 \text{ dB / octave}$ veya $+20 \text{ dB / decade}$ yanıtı sağlayacaktır.

Bu kökler gerçek veya karmaşık olabilir. Karmaşık olduklarında, eşlenik çiftlerden oluşurlar. Bu kökler, yatay eksenin σ (gerçek eksen) ve dikey eksenin ω (sanal eksen) olduğu düzlem üzerine çizilir. Buna bakılarak köklerin s -düzleminde nasıl dağıldığı belirlenebilir ve devre hakkında birçok bilgi edinilebilir. Kararlılık için, kutupların s -düzleminde sol tarafında olması gerekir. Eğer orijinde bir sıfır varsa, bu paydaki sıfırdır.

F_0 filtrenin kesim frekansıdır. Bu genel olarak, yanıtın geçiş bandından 3 dB değerine düştüğü frekans olarak tanımlanır. Geçiş bandından düşeceği frekans olarak da tanımlanabilir. Filtrenin zaman alan cevabını tanımlayan faz ve gecikme eğrilerinin yanı sıra zayıflama eğrisinin şekli gerçek frekansın incelenen kesim frekansına oranıyla aynı olacaktır. Filtrenin 1 rad/s 'ye normalize edilmesi, filtrelerin tasarlanması ve karşılaştırılması

için basit bir sistem geliştirilebilir. Filtre daha sonra gerçek filtre için bileşen değerlerini belirlemek için kesim frekansı ile ölçeklendirilir.

Q , filtrenin kalite faktörüdür. Aynı zamanda α olarak da verilebilir. α ile Q arasındaki ilişki Denklem (2.4)'te görüldüğü gibidir.

$$\alpha = \frac{1}{Q} \quad (2.4)$$

Bu genellikle sönümlleme oranı ξ olarak bilinir. $\xi=2\alpha$ 'dır. Eğer $Q>0.707$ ise, filtrenin genlik cevabında biraz pik olacaktır. Eğer $Q<0.707$ ise, daha yumuşak bir eğime sahip olacaktır [3].

Filtreler analog ve dijital olmak üzere iki şekilde sınıflandırılabilir. Analog filtreler aktif ve pasif olmak üzere ikiye ayrılır. Aktif filtreler, istenen filtre özelliklerini sentezlemek için geri besleme döngülerinde dirençler ve kapasitörler ile birlikte, yükselteçlerin özelliklerinden yararlanan filtrelerdir. Pasif filtreler direnç, kapasitör, indüktörler gibi pasif devre elemanlarından oluşur. Pasif filtrelerde transistörler, işlemsel yükselteçler gibi yükseltici elemanlar kullanılmaz [10].

2.2. Analog Filtreler

Analog filtreler, hangi frekans bantlarının geçtiği ve hangi frekans bantlarının durduğuna bağlı olarak birkaç kategoride sınıflandırılabilir. Yüksek geçiren bir filtre ve alçak geçiren bir filtre birlikte bir bant geçiren filtreyi oluşturur. Bant geçiren filtre, daha düşük bir alt kesim frekansı, f_l ve bir üst kesim frekansı, f_h arasında bir frekans bandını geçirir, ayrıca f_l 'den daha düşük frekanslar durdurma bandındadır. Bant geçiren filtresinin tamamlayıcısı bant durduran filtredir. Burada, geçiş bantları f_l ve f_h frekans aralığının dışındaki frekans bantlarını içerir. Bant f_l 'den f_h 'ye durdurma bandıdır. Yukarıda tanımlanan idealize edilmiş filtreler maalesef kolayca oluşturulamaz. Geçiş bandından durdurma bandına geçiş anlık olmayıp bunun yerine bir geçiş bölgesi oluşur. Durdurma bandı zayıflaması da sonsuz olmaz [3].

Analog filtrelerde filtre derecesindeki artışla analog filtreler söz konusu olduğunda, filtre bileşenlerinin sayısı artar ve bu da karmaşıklığını artırır. Analog filtrelerin bir başka dezavantajı, kesim frekansının, direnç ve kapasitör değerleri gibi filtre bileşenlerinin

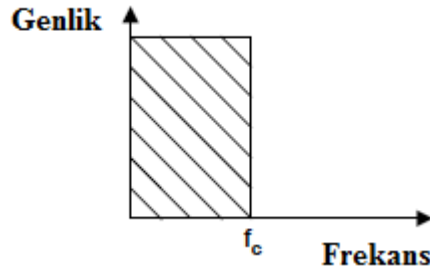
değerlerine, teknik olarak zamana bağlı olarak bileşenlerin kayması olarak adlandırılan bileşen değerlerinde meydana gelen değişikliklere bağlı olmasıdır. Sıcaklık gibi fiziksel parametreler, filtrenin kesim frekansını değiştirir [11].

2.2.1. Alçak Geçiren Filtre

Alçak geçiren filtre, düşük frekansları geçiren ve yüksek frekansları geçirmeyen bir devredir [12].

Transfer fonksiyonu $H(s)$, ω_0 ve Q cinsinden düzenlenirse Denklem (2.5)'teki eşitlik elde edilir.

$$H(s) = \frac{H_0}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \quad (2.5)$$



Şekil 2.2. İdeal Alçak Geçiren Filtrenin Genlik-Frekans Değişimi

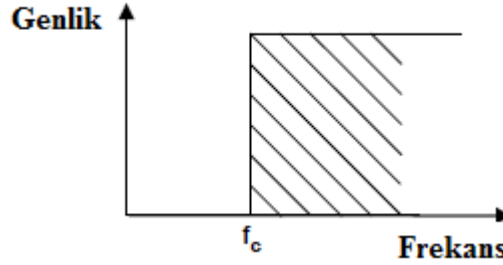
İdeal alçak geçiren filtrenin genlik-frekans değişimi Şekil 2.2'de görülmektedir

2.2.2. Yüksek Geçiren Filtre

Bu filtre f_c kesim frekansından daha yüksek olan frekansları geçirir ve düşük olanları geçirmez [12].

Alçak geçiren filtrenin transfer fonksiyonu $H(s)$ 'nin payının s^2 ile çarpılmasıyla yüksek geçiren bir filtrenin transfer fonksiyonu elde edilir. Yüksek geçiren filtrenin cevabı alçak geçiren filtrenin tersidir. Yüksek geçiren filtrenin transfer fonksiyonu Denklem (2.6)'da görülmektedir.

$$H(s) = \frac{H_0 s^2}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2} \quad (2.6)$$



Şekil 2.3. İdeal Yüksek Geçiren Filtrenin Genlik-Frekans Değişimi

İdeal yüksek geçiren filtrenin genlik-frekans değişimi Şekil 2.3’de görülmektedir [3].

2.2.3. Bant Geçiren Filtre

Bir bant geçiren filtre, ω_0 etrafında merkezlenen frekans bandını geçiren bir devredir [12]. Bu filtre tipi bir sinyal frekansı bandını geçirir ancak geçiş aralığından daha düşük veya daha yüksek olan frekansları geçirmez [13].

Alçak geçiren filtrenin transfer fonksiyonu $H(s)$ 'nin payının ω_0^2 ile çarpılmasıyla bant geçiren bir filtrenin transfer fonksiyonu elde edilir. Bant geçiren filtrenin cevabı alçak geçiren filtrenin tersidir. Bant geçiren filtrenin transfer fonksiyonu Denklem (2.7)’deki eşitlikte görülmektedir.

$$H(s) = \frac{H_0 \omega_0^2}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2} \quad (2.7)$$

Bu eşitlikte ω_0 filtre tepelerinin kazancındaki frekans olup H_0 devre kazancıdır. H_0 devre kazancı Denklem (2.8)’deki gibi tanımlanır.

$$H_0 = H/Q \quad (2.8)$$

Q, bant geirme yanıtı iin filtrenin seiciliğidir ve Denklem (2.9)'daki eřitlikte tanımlanır.

$$Q = \frac{F_0}{F_H - F_L} \quad (2.9)$$

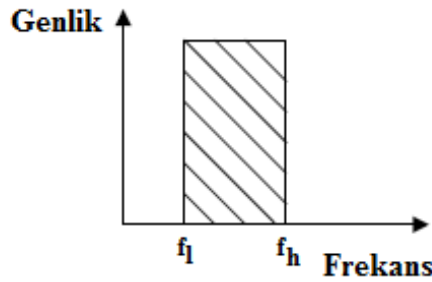
Bu eřitlikte F_L ve F_H maksimumdan 3 dB eksik olduėu cevaptaki frekanslardır. Filtrenin bant geniřliėi (BW) Denklem (2.10)'daki gibi tanımlanır.

$$BW = F_H - F_L \quad (2.10)$$

F_0 rezonans frekansı F_L ve F_H 'ın geometrik ortalamasıdır. Yani F_0 , logaritmik lekte F_L ve F_H arasında yarı dzlemde grnecektir. F_0 rezonans frekansının eřitliėi Denklem (2.11)'de grlmektedir.

$$F_0 = \sqrt{F_H F_L} \quad (2.11)$$

Ayrıca, band-pass cevabının kenarlarının her zaman logaritmik lekte F_0 etrafında simetrik olduėu sylenbilir. İdeal bant geiren filtrenin genlik-frekans deėiřimi řekil 2.4'te grlmektedir [3].



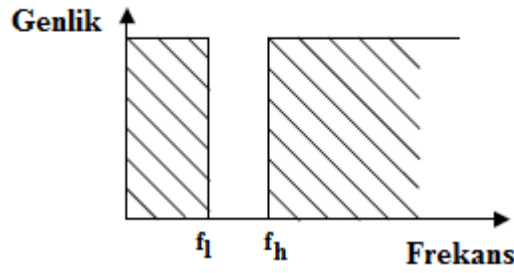
řekil 2.4. İdeal Bant Geiren Filtrenin Genlik-Frekans Deėiřimi

2.2.4. Band Durduran Filtre

Bant durduran filtre, tek bir bant dışındaki tüm frekansları geçiren bir filtredir [12]. Bu filtre tipi bir sinyal frekansı bandını geçirmez, ancak geçiş aralığından daha düşük veya daha yüksek olan frekansları geçirir [13].

Alçak geçiren filtrenin transfer fonksiyonu $H(s)$ 'nin payının $s^2 + \omega_0^2$ ile çarpılmasıyla bant durduran bir filtrenin transfer fonksiyonu elde edilir. Band durduran filtrenin transfer fonksiyonu Denklem (2.12)'deki eşitlikte görülmektedir.

$$H(s) = \frac{H_0(s^2 + \omega_0^2)}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \quad (2.12)$$



Şekil 2.5. İdeal Bant Durduran Filtrenin Genlik-Frekans Değişimi

Şekil 2.5'te ideal bant durduran filtrenin genlik-frekans değişimi görülmektedir [3].

2.2.5. Tüm Geçiren Filtre

Tüm geçiren filtre, genliği sabit bir filtredir, böylece tüm frekansları geçirir [24]. Tüm geçiren filtre sinyalin tamamının genliği geçiren ancak faz kaymasını başlatan bir filtre tipidir. Bu filtrenin amacı, devrenin yanıtına faz kayması (gecikme) eklemektir. Tüm geçiren filtrenin genliği tüm frekanslar için tektir. Bununla birlikte faz cevabı, frekans sıfırdan sonsuza doğru salındığından 0° ile 360° arasında değişir.

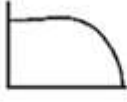
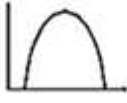
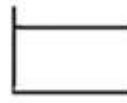
Tüm geçiren bir filtrenin amacı, tipik olarak darbe devrelerinde faz dengelemesi sağlamaktır. Ayrıca bu filtre tipi taşıyıcısı bastırılmış tek yan bant (SSB-SC) modülasyon devrelerine de uygulanır. Tüm geçiren filtrenin transfer fonksiyonu Denklem (2.13)'teki eşitlikte görülmektedir.

$$H(s) = \frac{s^2 - \frac{\omega_0}{Q}s + \omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2} \quad (2.13)$$

Bir tüm geçiren filtrenin transfer fonksiyonu Denklem (2.14)'teki gibi sentezlenebilir.

$$H_{AP} = H_{LP} - H_{BP} + H_{HP} = 1 - 2H_{BP} \quad (2.14)$$

Tablo 2.1. Filtre tiplerinin genlik cevabı ve transfer fonksiyonu

Filtre Tipi	Genlik	Transfer Fonksiyonu
Alçak Geçiren		$H(s) = \frac{H_0}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}$
Bant Geçiren		$H(s) = \frac{H_0 \omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}$
Bant Durduran		$H(s) = \frac{H_0(s^2 + \omega_0^2)}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}$
Yüksek Geçiren		$H(s) = \frac{H_0 s^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}$
Hep Geçiren		$H(s) = \frac{s^2 - \frac{\omega_0}{Q}s + \omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}$

Tablo 2.1'de filtre tipleri genlik ve transfer fonksiyonu bakımından karşılaştırılmıştır [3].

2.3. Dijital Filtreler

Dijital teknolojideki gelişmeler nedeniyle, dijital filtre ve sistem tasarımı günlük yaşamda daha önemli bir rol oynamaktadır. Akıllı teknolojiler, sensörlerden gelen sinyalleri işleyebilen ve çevre birimleri veya diğer akıllı sistemler ile iletişim kurabilen dijital sistemlerde ortaya çıkar. Dijital filtreler, dijital sinyal işleme (DSP) için çok önemlidir. Filtreler esasen, sistemlerin genlik tepkilerinde tasarlanmış durma ve geçiş bantları aracılığıyla frekans seçiciliği için kullanılır [14].

Dijital filtreler bir veya daha fazla ayrık zaman sinyali (sinyal numunesi) alır ve bir veya daha fazla çıkış üretmek için bu sinyalleri değiştirir ve filtreler gereken işlemlerine bağlı olarak frekansları geçirir veya geçirmez [13].

Dijital filtreler, genel amaçlı bir bilgisayarda veya iş istasyonunda, analog filtrelere göre kolayca tasarlanır, test edilir ve uygulanır [4]. Tasarlanan filtrenin derecesi yükseldikçe filtreleme daha keskin olur. Yüksek dereceli bir filtre, daha fazla katsayı kullanımı nedeniyle çok daha fazla hassasiyet sağlar. Dezavantajı, daha fazla bit kullanması ve bu nedenle sistemde daha fazla bellek gerektirmesidir [15].

Dijital FIR filtreleri, birçok iyi bilinen özellikleri ve popüler tasarım yöntemleri nedeniyle birçok sinyal işleme uygulamasında yaygın olarak kullanılmaktadır [16].

Dijital filtreler darbe cevabının uzunluğuna bağlı olarak, Sonlu Darbe Tepki (FIR) Filtreleri ve Sonsuz Darbe Tepki (IIR) Filtreleri olmak üzere iki türe ayrılır [17]. IIR filtreleri analog filtrelerin dijital karşılığıdır. Bu filtrenin çıktısı önceki girdilerin ve çıktıların doğrusal bir birleşimi olarak belirlenir. Başka bir deyişle, bu tür filtrelerin uygulanmasında geribildirim var. FIR filtrelerinde böyle bir geribildirim yoktur. Teorik olarak mümkün olsa da, uygulamada bilgisayar kayıtlarının sınırlı kapasitesi göz önünde bulundurulursa IIR filtrenin darbe tepkisi hiçbir zaman sıfıra düşmemektedir [18].

FIR filtresinin transfer fonksiyonu yalnızca sıfırlara sahip olduğundan dolayı daha fazla belleğe ihtiyaç duyarken, IIR filtresinin transfer fonksiyonu hem sıfırlara hem de kutuplara sahip olduğundan dolayı FIR'dan daha az bellek gerektirir [19].

IIR filtreler geri beslemeli olduğundan daha az katsayı ve değişkenler için daha az depolama alanı gerektirme, daha iyi büyüklük tepkilerine sahip olma, daha düşük gecikme süresi avantajlarına sahip olan bu filtreler analog modellere daha yakındır [20].

FIR filtre tasarımı karmaşık olmasına rağmen, kalıcı avantajları nedeniyle IIR filtrelere kıyasla filtreleme uygulamaları için yaygın olarak kullanılır. IIR filtreleri daha yüksek derecelerde kararlılık sağlayamazken, FIR emsalleri her zaman kararlıdır ve özellikle tam lineer faz tepkisi gerektiren uygulamalar için kullanışlıdır [21].

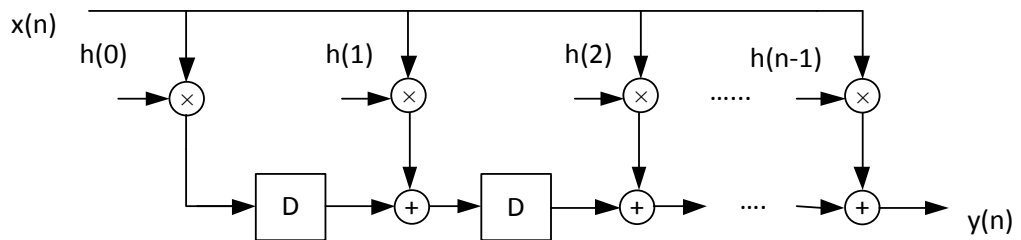
2.3.1. FIR Filtre

Dijital Sonlu-Darbe-Tepki (FIR) filtreleri temel olarak dijital sinyal işleme uygulamalarında kullanılırlar [22]. Dijital sinyal işlemede son derece popüler oldukları birçok uygun özelliğe sahiptir [23]. Bu filtreler elektronik sistemlerde gürültüyü bastırmak, seçilen frekans aralıklarındaki sinyalleri vurgulamak, bant genişliğini sınırlamak ve belirli frekanslardaki sinyalleri geçirmemek için kullanılır. FIR filtreleri çok çeşitli örnekleme oranlarında uygulanır [20].

Konuşma tanıma, konuşma sentezi, dijital ses, telekomünikasyon, sismik sinyal işleme (gürültü giderme) ve diğer çeşitli sinyal işleme alanları gibi çeşitli uygulamalarda kullanılır [24].

Bir dizi ardı ardına eklenerek FIR filtrenin yapısı elde edilir. Denklem (2.16)'da FIR filtre eşitliği görülmektedir [25]. FIR filtrenin temel yapı taşları, toplayıcılar, çarpıcılar ve sinyal gecikmesidir. Filtrenin çıkışı ideale yakın olması için çarpanlar yeterince hızlı olmalıdır.

Filtrenin en yaygın kullanılan yapıları, doğrudan form ve dönüştürülmüş form olarak iki tiptir.



Şekil 2.6. FIR filtrenin dönüştürülmüş formu

Doğrudan formda, birçok sinyal sabitlerle çarpılır ve sonradan eklenir. Ancak, toplayıcının gecikme süresini azaltmak ve yüksek verim elde etmek için toplayıcılar arasında ekstra hafıza blokları gerektirir. Aksine, dönüştürülmüş formda, toplayıcılar arasında hafıza blokları mevcuttur. Ayrıca, ilave hafıza blokları olmaksızın yüksek verimli çıkış elde edilir. Şekil 2.6'da FIR filtrenin dönüştürülmüş formu gösterilmektedir [26]. FIR filtre fark denklemi Denklem (2.15)'de görülmektedir.

$$y[n] = \sum_{k=0}^N b_k x[n-k] \quad (2.15)$$

FIR filtre kararlıdır, yinelemeli bir şekilde uygulanabilir olsa da denklemlerinde geçmiş veya şimdiki girdilerin geri beslemesi olmadığından dolayı özyinelemeli değildir [27].

2.3.1.1. Birim Dürtü Yanıtı

Bu alt bölümün birbirine bağlı üç kavramı; tek darbe dizisi, birim darbe tepkisi ve evrişim toplamıdır. Sıfır olmayan bir değere sahip olan bir dizi, yalnızca argümanı sıfıra eşit olduğunda yani, Denklem (2.16)'da görüldüğü gibi $n=0$ iken birim darbe dizisi $\delta[n]=1$ olur.

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \quad (2.16)$$

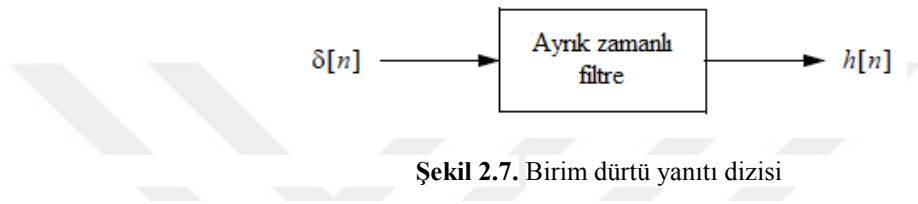
Birim örnek dizisi, Denklem(2.17)'de görüldüğü gibi n_0 tamsayısı ile sağa veya sola kaydırılabilir.

$$\delta[n-n_0] = \begin{cases} 1, & n = n_0 \\ 0, & n \neq n_0 \end{cases} \quad (2.17)$$

Darbe sırasını hem zaman kayması hem de genlik ölçeğinde ölçeklendirilebilir, bunların doğrusal bir kombinasyonu herhangi bir sırayı oluşturmak için kullanılabilir ve bu genellemelerle Denklem (2.18)'de görüldüğü gibi seri genişletilebilir.

$$\begin{aligned}
x[n] &= \sum_k x[k] \delta[n-k] \\
&= \dots + x[-1] \delta[n+1] + x[0] \delta[n] \\
&\quad + x[1] \delta[n-1] + x[2] \delta[n-2] + \dots
\end{aligned} \tag{2.18}$$

$x[n]=\delta[n]$ dizisi bir FIR filtresine giriş olarak uygulandığında filtre çıktısı (filtrenin başlangıç durumunda olduğu varsayılarak) $y[n]=h[n]$ olarak ifade edilen birim dürtü yanıtıdır. Şekil 2.7’de görüldüğü gibi bu tanım, yalnızca FIR filtreleri için değil, herhangi bir ayrık zamanlı filtre için geçerlidir.



$n_0=1$ olduğunda, $y[n]=x[n-1]$ birim gecikmesi veren bir sisteme karşılık gelir. Birim geciktirme sistemi özel bir FIR filtresi olup $b_k=\{0,1\}$ ve filtre derecesi $m=1$ ’dir.

2.3.1.2. Evrişim ve FIR Filtreler

Bir filtre çıktısının genel bir ifadesi girdi olarak ve birim dürtü yanıtı açısından Denklem (2.19)’da formülde ifade edildiği gibi gösterilebilir. Evrişim toplamı olarak adlandırılan bu formülde $y[n]$, $x[n]$ ve $h[n]$ ’nin evrişimidir.

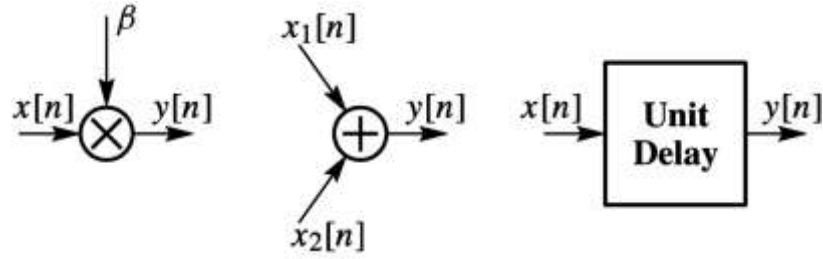
$$y[n] = \sum_{k=0}^M h_k x[n-k] \tag{2.19}$$

2.3.1.3. FIR Filtre Uygulanmasında Kullanılan Blok Yapıları

Bir FIR filtre uygulanması üç temel işlemi gerçekleştirecek blok gerektirir. Ayrıca filtre katsayılarını hafızada saklanabilmesi gerekir.

- Çarpma işlemi
- Toplama işlemi
- Sinyal gecikmesi

FIR filtre uygulanmasında kullanılan blok yapıları Şekil 2.8’de görülmektedir.



Şekil 2.8. FIR filtre uygulanmasında kullanılan blok yapıları

Bir DSP sisteminde çarpıcı hızlı olmalı ve istenen uygulamayı desteklemek için yeterli hassasiyete (bit genişliği; düşünme mantık devreleri) sahip olmalıdır.

Genelde yüksek kaliteli bir filtre, daha düşük kalitede olanlardan daha fazla çarpma gerektirir, bu nedenle, eğer çarpıcı hızlı değilse, verim düşer. Denklem (2.20)’de görüldüğü gibi giriş bir katsayıyla çarpılıp çıkışa verilir.

$$y[n] = \beta x[n] \quad (2.20)$$

Sinyal ekleme çok temel bir DSP işlevidir. Bir FIR filtresinde çarpıcılarla birlikte toplayıcılar da gereklidir, bu nedenle DSP mikroişlemciler çoğaltılmış (MAC) ünitelere sahiptir.

Toplayıcılar genellikle Denklem (2.21)’de görüldüğü gibi bir seferde yalnızca iki girişle çalışır.

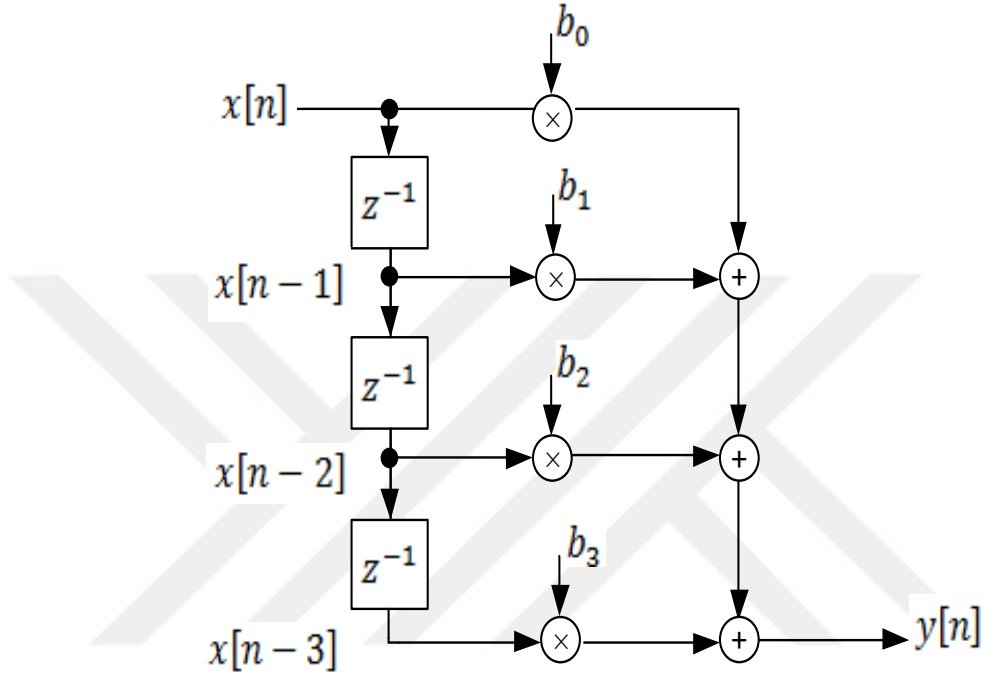
$$y[n] = x_1[n] + x_2[n] \quad (2.21)$$

Birim gecikmesi bir örnek sinyal gecikmesi sağlar. Bir örnek değeri bir örnek saat döngüsü için bellek yuvasında saklanır ve daha sonra bir sonraki işlem aşamasına bir girdi olarak kullanılabilir duruma getirilir.

Her bellek hücresinin B-bitleri olarak ifade edilen bitleri kaydetmesi gerekir. Bir M birim gecikmesi, B-bit genişliğinde kaydırma yazmacı olarak yapılandırılmış M bellek hücresi gerektirir. Birim gecikme bloğu ile Denklem (2.22)’de görüldüğü gibi girişe bir birim gecikmesi uygulanarak çıkışa verilir.

$$y[n] = x[n-1] \quad (2.22)$$

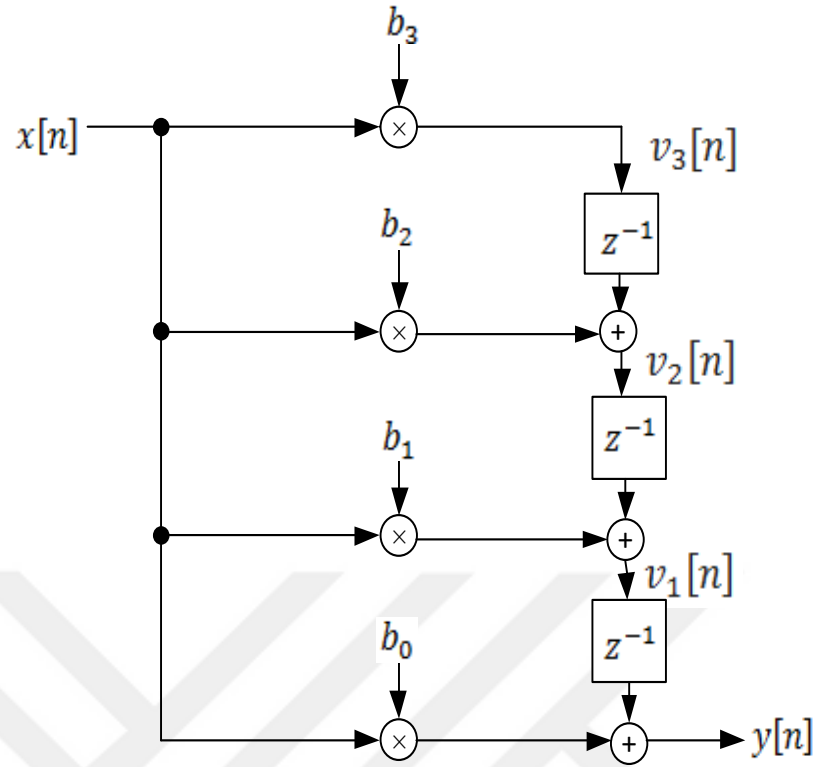
Doğrudan Form FIR filtre blok şeması Şekil 2.9’da, Doğrudan Form FIR filtre çıkış eşitliği Denklem (2.23)’te görülmektedir.



Şekil 2.9. Doğrudan Form FIR filtre blok şeması

$$y[n] = \sum_{k=0}^3 b_k x[n-k] \quad (2.23)$$

Giriş çıkış yönünde bağlayan tüm yollar ileri yönde aktığından dolayı sinyalin ileri beslemeli olduğu görülmektedir.



Şekil 2.10. Dönüştürülmüş blok diyagram

Şekil 2.10’da dönüştürülmüş blok diyagram görülmektedir.

$$\begin{aligned}
 y[n] &= b_0 x[n] + v_1[n-1] \\
 v_1[n] &= b_1 x[n] + v_2[n-1] \\
 v_2[n] &= b_2 x[n] + v_3[n-1] \\
 v_3[n] &= b_3 x[n]
 \end{aligned} \tag{2.24}$$

Denklem (2.24)’de elde edilen fark denklemi düzenlenerek Denklem(2.25)’teki eşitlik oluşturulmuştur.

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_3 x[n-3] \tag{2.25}$$

2.3.1.4. FIR Filtrelerin Doğrusallık, Zamanla Değişmezlik ve Lineerliği

Bir sistemin doğrusal olması için, $x_1[n] \Rightarrow y_1[n]$ ve $x_2[n] \Rightarrow y_2[n]$ eşitliklerini vermek üzere süperpozisyona sahip olmalıdır.

$$x[n] = \alpha x_1[n] + \beta x_2[n] \Rightarrow y[n] = \alpha y_1[n] + \beta y_2[n] \quad (2.26)$$

Bütün α ve β değerleri için Denklem (2.26)'daki eşitlik özel bir durum olarak $\alpha=\beta=1$ için Denklem (2.27)'deki gibi olur.

$$x[n] = x_1[n] + x_2[n] \Rightarrow y_1[n] + y_2[n] = y[n] \quad (2.27)$$

Ayrıca eğer $\beta=0$ ise eşitlik Denklem (2.28)'deki gibi olur.

$$x[n] = \alpha x_1[n] \Rightarrow \alpha y_1[n] = y[n] \quad (2.28)$$

Ayrık zamanlı bir sistemde, Denklem (2.29)'da görüldüğü gibi n_0 kadar geciktirilen bir girdi için çıktı aynı miktarda geciktirilir.

$$x[n - n_0] \Rightarrow y[n - n_0] \quad (2.29)$$

Başlangıçta $v[n] = x[n - n_0]$ olsun;

$$\begin{aligned} w[n] &= \sum_{k=0}^M b_k v[n - k] = \sum_{k=0}^M b_k x[(n - k) - n_0] \\ &= \sum_{k=0}^M b_k x[(n - n_0) - k] \end{aligned} \quad (2.30)$$

$$\begin{aligned} y[n] &= \sum_{k=0}^M b_k x[n - k] \\ \Rightarrow y[n - n_0] &= \sum_{k=0}^M b_k x[(n - n_0) - k] \end{aligned} \quad (2.31)$$

Denklem (2.30) ve Denklem (2.31)'de elde edilen eşitlikler zamanla değişmezlik özelliğini kanıtlanmaktadır.

Giriş $\alpha x_1[n] + \beta x_2[n]$ olmak üzere Denklem (2.32)'de elde edilen sonuç lineerlik özelliğini kanıtlamaktadır

$$\begin{aligned}
 y[n] &= \sum_{k=0}^M b_k (\alpha x_1[n-k] + \beta x_2[n-k]) \\
 &= \alpha \sum_{k=0}^M b_k x_1[n-k] + \beta \sum_{k=0}^M b_k x_2[n-k] \\
 &= \alpha y_1[n] + \beta y_2[n]
 \end{aligned} \tag{2.32}$$

2.3.1.5. Evrişim Toplamının Türetilmesi

En genel durumda, toplamın $-\infty$ 'dan ∞ 'a kadar olduğu yerlerde herhangi bir sinyal için;

$$x[n] = \sum_l x[l] \delta[n-l] \tag{2.33}$$

Denklem (2.33)'teki eşitlikten zamanla değişmezlik özelliğinden yararlanılarak Denklem (2.34)'deki eşitlik elde edilir.

$$\begin{aligned}
 \delta[n-n_0] &\Rightarrow h[n-n_0] \\
 x[l] \delta[n-l] &\Rightarrow x[l] h[n-l]
 \end{aligned} \tag{2.34}$$

Denklem (2.33)'teki eşitliğin her iki tarafına da doğrusallık özelliği uygulanarak Denklem(2.35)'deki eşitlik elde edilir.

$$\begin{aligned}
 x[n] &= \sum_l x[l] \delta[n-l] \\
 &\Rightarrow \sum_l x[l] h[n-l] = y[n]
 \end{aligned} \tag{2.35}$$

$$y[n] = \sum_{l=-\infty}^{\infty} x[l]h[n-l] \quad (2.36)$$

Tüm eksen boyunca n desteği olan bir giriş sinyali durumunda, tüm LTI sistemler için evrişim toplamı formülü Denklem (2.36)'daki gibidir.

2.3.1.6. FIR Filtrelerin frekans cevabı

Frekans cevabı filtrenin $x[n] = e^{j\hat{\omega}_0 n}$, $-\infty < n < \infty$ şeklinde bir girdisine nasıl tepki verdiğidir.

2.3.1.7. FIR Sistemlerinin Sinüzoidal Tepkisi

$x[n] = Ae^{j\phi} e^{j\hat{\omega}_0 n}$, $-\infty < n < \infty$ şeklinde girişi karmaşık bir sinüzoid olan bir FIR filtresi düşünülürse, $x[n]$ kompleks sinüzoid $x(t) = Ae^{j\phi} e^{j\hat{\omega}_0 t}$ ve $\hat{\omega}_0 = \omega_0 T_s = y$ örneklerinin alınmasıyla elde edilen $x[n]$ idi.

$$\begin{aligned} y[n] &= \sum_{k=0}^M b_k x[n-k] = \sum_{k=0}^M b_k Ae^{j\phi} e^{j\hat{\omega}_0 (n-k)} \\ y[n] &= Ae^{j\phi} e^{j\hat{\omega}_0 n} \left(\sum_{k=0}^M b_k e^{-j\hat{\omega}_0 k} \right) \\ &= Ae^{j\phi} e^{j\hat{\omega}_0 n} H(e^{j\hat{\omega}_0}), \quad -\infty < n < \infty \end{aligned} \quad (2.37)$$

Bir M+1 katsayılı FIR filtresi için Denklem (2.37)'de görülen fark denkleminde rastgele bir $\hat{\omega}$ için tanımlayacak olursak FIR filtrenin frekans cevabı Denklem (2.38)'deki gibi olacaktır.

$$H(e^{j\hat{\omega}}) = \sum_{k=0}^M b_k e^{-j\hat{\omega} k} \quad (2.38)$$

2.3.1.8. Kararlı Durum ve Geçici Yanıt

Frekans cevabı tanımı, karmaşık sinüzoid girişi için destek aralığının $[-\infty, \infty]$ olduğu gerçeğine dayanır. Bir bilgisayar simülasyonunda veya gerçek zamanlı bir sinyal işleme uygulamasında, $-\infty$ 'dan başlayan giriş sinyallerini dikkate almak pratik değildir.

Girişi açmak için birim adım işlevini kullanarak $n = 0$ ile başlayan bir giriş düşünülürse;

$$x[n] = Xe^{j\hat{\omega}}u[n] \quad (2.39)$$

$$u[n] \equiv \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (2.40)$$

Denklem (2.40)'da $X = Ae^{j\phi}$ 'dir. Denklem (2.39) ve Denklem (2.40) düzenlenerek bir LTI FIR sistemi için evrişim toplamı formülünün verimi Denklem (2.41)'de görüldüğü gibi elde edilir.

$$y[n] = \sum_{k=0}^M b_k X e^{j\hat{\omega}(n-k)} u[n-k] \quad (2.41)$$

$$y[n] = \begin{cases} 0, & n < 0 \\ \left(\sum_{k=0}^n b_k e^{-j\hat{\omega}k} \right) X e^{-j\hat{\omega}n}, & 0 \leq n < M \\ \left(\sum_{k=0}^M b_k e^{-j\hat{\omega}k} \right) X e^{-j\hat{\omega}n}, & n \geq M \end{cases} \quad (2.42)$$

Denklem (2.41)'deki toplam Denklem (2.42)'de görülen 3 durumu kapsar;
 $n < 0$ olduğunda giriş mevcut değildir, bu yüzden çıkış sıfırdır
 $0 \leq n < M$ olduğunda, giriş filtreye kısmen girmiştir ve üretilen çıkış geçici cevaptır.
 $n \geq M$ olduğunda, giriş filtreye tam olarak geçtiğinden çıkış sabit durumdadır.

Kararlı durum yanıtı, Denklem (2.43)'teki ifadeye eşdeğerdir.

$$y[n] = H(e^{j\omega})Xe^{j\omega n}, \quad n \geq M \quad (2.43)$$

Ayrıca kararlı durum çıkışında, girişin her zaman $n \geq 0$ için $Xe^{j\omega n}$ olduğu varsayılır.

2.3.1.9. Z-Dönüşümü ve Doğrusal Sistemler

Z dönüşümü, özellikle LTI sistemlerinin analizinde ve tasarımında kullanışlıdır. $x[n]$ girişi ve impuls yanıtı $h[n]$ olan herhangi bir LTI sistemi için, çıkış $y[n] = x[n] * h[n]$ olur. FIR filtre için Denklem (2.44)'deki $h[n]$ 'nin z dönüşümü yapılmak istenirse;

$$h[n] = \sum_{k=0}^M b_k \delta[n-k] \quad (2.44)$$

Bunu devam ettirmek için Denklem (2.45)'deki giriş ele alınırsa;

$$x[n] = z^n, \quad -\infty < n < \infty \quad (2.45)$$

Çıkış Denklem (2.46)'daki gibi olur.

$$\begin{aligned} y[n] &= \sum_{k=0}^M b_k x[n-k] = \sum_{k=0}^M b_k z^{n-k} \\ &= \sum_{k=0}^M b_k z^n z^{-k} = \left(\sum_{k=0}^M b_k z^{-k} \right) z^n \end{aligned} \quad (2.46)$$

Parantez içindeki terim, FIR filtresinin sistem işlevi olarak da bilinen $h[n]$ 'nin z-dönüşümüdür. $H(e^{j\omega})$ 'nın sistem işlevi Denklem (2.47)'deki şekilde tanımlanır.

$$H[z] = \sum_{k=0}^M b_k z^{-k} = \sum_{k=0}^M h[k] z^{-k} \quad (2.47)$$

Denklem (2.48)'de z-dönüşümü çifti görülmektedir.

$$\sum_{k=0}^M b_k \delta[n-k] \xleftrightarrow{z} \sum_{k=0}^M b_k z^{-k}$$

$$h[n] \xleftrightarrow{z} H(z) \quad (2.48)$$

Frekans tepkisi sonucuna benzer bir diğer sonuç ise Denklem (2.49)'da görülmektedir.

$$y[n] = h[n] * z^n = H(z) z^n \quad (2.49)$$

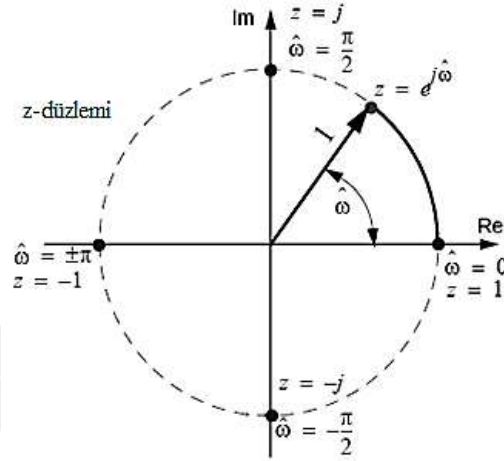
Eğer $z = e^{j\omega}$ ise, frekans tepkisi sonucu elde edilmiş olur. Sistem fonksiyonu, karmaşık değişken z'de bir M derece polinomdur. Herhangi bir polinomda olduğu gibi, M kökleri veya sıfırları olacaktır, yani z_0, M_0 değerleri vardır, öyle ki $h(z_0)=0$ 'dır. Bu M sıfırları polinomu Denklem (2.50)'de görüldüğü gibi tamamen bir kazanç sabiti içinde ölçek faktörü tanımlar.

$$\begin{aligned} H(z) &= b_0 + b_1 z^{-1} + \dots + b_M z^{-M} \\ &= (1 - z_1 z^{-1})(1 - z_2 z^{-1}) \dots (1 - z_M z^{-1}) \\ &= \frac{(1 - z_1)(1 - z_2) \dots (1 - z_M)}{z^M} \end{aligned} \quad (2.50)$$

Burada $z_k, k = 1, \dots, M$ sıfırları belirtir.

2.3.1.10. Z-Düzlemi ve Birim Çember

Bir z-düzlemi düşünülürse Şekil 2.11'deki birim çemberde $H(z)$, $H(e^{j\omega})$ 'e tekabül ettiği görülmektedir.



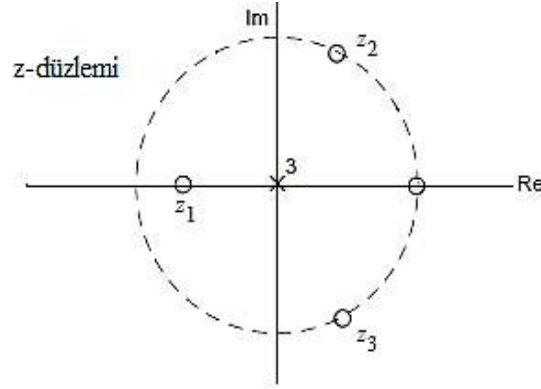
Şekil 2.11. Birim çember

$$H(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} \quad (2.51)$$

$$H(z) = (1 - z_1 z^{-1})(1 - z_2 z^{-1}) \dots (1 - z_M z^{-1}) \quad (2.52)$$

Denklem (2.51)'de $b_0 = 1$ olarak varsayılmıştır. Denklem (2.52)'deki eşitlik z^3 / z^3 ile çarpılarak, z'nin pozitif kuvvetleri cinsinden $H(z)$ Denklem (2.53)'teki şekilde yazılır.

$$\begin{aligned} H(z) &= \frac{z^3 + b_1 z^2 + b_2 z^1 + b_3 z^0}{z^3} \\ &= \frac{(1 - z_1)(1 - z_2)(1 - z_3)}{z^3} \end{aligned} \quad (2.53)$$



Şekil 2.12. Kutup ve sıfır noktaları

Sıfırlar $H(z) = 0$, kutuplar $H(z) \rightarrow \infty$ yani $z \rightarrow 0$ konumlarındadırlar. Kutup ve sıfır noktaları Şekil 2.12'deki birim çemberde görülmektedir.

2.3.2.IIR Filtre

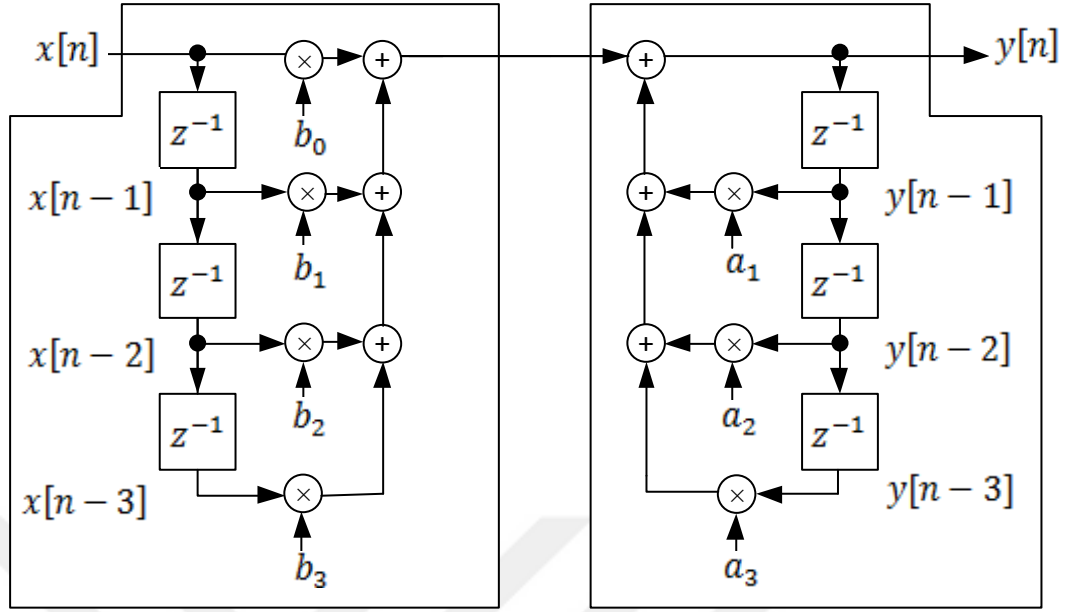
Sonsuz dürtü yanıtı (IIR) filtresi, filtrenin çıkışının mevcut ve önceki girişler ve önceki çıkışlar kullanılarak hesaplandığı yinelemeli bir filtredir. Filtre çıktının önceki değerlerini kullandığından, filtre yapısında çıktının geri bildirimi vardır. [29]. IIR fark denklemi Denklem (2.54)'de görülmektedir.

$$\sum_{l=0}^N a_l y[n-l] = \sum_{k=0}^M b_k x[n-k] \quad (2.54)$$

Denklem (2.54) yeniden düzenlenip $y[n]$ eşitliğin sol tarafına alınırsa a_l katsayısının işaret değiştirdiği Denklem (2.55)'de görülmektedir. Ayrıca $a_0 = 1$ olarak varsayılır.

$$y[n] = \sum_{l=1}^N a_l y[n-l] + \sum_{k=0}^M b_k x[n-k] \quad (2.55)$$

Toplam katsayı sayısı, $N + M + 1$ 'dir ve fark denkleminde her bir yeni çıkışı hesaplamak için birçok çarpan gerekli olduğu anlaşılmaktadır. Şekil 2.13'deki mantıksal yapı düzeni daha üst katsayılarda aynı mantıksal yapı ile uzanır.



Şekil 2.13. Doğrudan Form I yapısı

2.3.2.1. Zaman Domeni Cevabı

IIR zaman domenı analizine başlamak için $M=0$ ve $N=1$ olarak düşünülürse IIR filtre fark denklemi Denklem (2.56)'daki gibi olur.

$$y[n] = a_1 y[n-1] + b_0 x[n] \quad (2.56)$$

2.3.2.2. Birinci Dereceden IIR Sisteminin Dürtü Cevabı

Dürtü cevabı sistem başlangıcında dinlenme durumunda iken $x[n] = \delta[n]$ olarak ayarlanarak elde edilebilir.

Bir IIR filtresi için ilk dinlenme koşulları şu anlama gelir:

Giriş, başlangıç zamanından önce sıfırdır. Yani $n < n_0$ için $x[n] = 0$ olur.

Çıkış, başlangıç zamanından önce sıfırdır. Yani $n < n_0$ için $y[n] = 0$ olur.

Fark denkleminin doğrudan yineleme yoluyla Denklem (2.57)'nin dürtü cevabı bulunabilir.

$$y[0] = a_1 y[-1] + b_0 \delta[0] = b_0$$

$$y[1] = a_1 y[0] + b_0 \delta[1] = a_1 b_0$$

$$y[2] = a_1 y[1] + b_0 \delta[2] = a_1^2 b_0$$

...

$$y[n] = a_1^n b_0, n \geq 0 \quad (2.57)$$

Özetle birinci dereceden IIR filtrenin dürtü cevabı Denklem (2.58)'de görülmektedir.

$$h[n] = b_0 (a_1)^n u[n] \quad (2.58)$$

Denklem (2.51)'de birim adımı $u[n]$, $n < 0$ olduğunda çıkışın 0 olduğunu açığa çıkarmak için kullanılmıştır.

2.3.2.3. IIR Filtrelerin Doğrusallığı ve Zaman Değişmezliği

Genel IIR fark denkleminin doğrusallık ve zaman değişmezliği sergilediği gösterilebilir. Doğrusallık ve zaman değişmezliği kullanarak, birinci mertebeden sistemin çıkışını zaman kaydırmalı dürtülerin doğrusal bir kombinasyonunu Denklem (2.59)'da gösterildiği gibi bulabiliriz.

$$x[n] = \sum_{k=N_1}^{N_2} x[k] \delta[n-k] \quad (2.59)$$

Denklem (2.51)'den yola çıkarak dürtü cevabının düzenlenmiş hali Denklem (2.60)'da görülmektedir.

$$y[n] = \sum_{k=N_1}^{N_2} x[k] h[n-k]$$

$$= \sum_{k=N_1}^{N_2} x[k] \{b_0 (a_1)^{n-k} u[n-k]\} \quad (2.60)$$

2.3.2.4. Birinci Mertebeden Bir Özyinelemeli Sistemin Adım Cevabı

Adım cevabı, bir filtrenin sonsuz uzunluktaki bir girdiye nasıl tepki verdiğini görmemize olanak sağlar.

$$y[n] = a_1 y[n-1] + b_0 x[n] \quad (2.61)$$

Denklem (2.61)'deki eşitlikten yola çıkılarak fark denkleminin doğrudan özyinelemesi ile Denklem (2.62)'deki eşitlik elde edilir.

$$y[0] = a_1 y[-1] + b_0 u[0] = b_0$$

$$y[1] = a_1 y[0] + b_0 u[1] = a_1 b_0 + b_0$$

$$y[2] = a_1 y[1] + b_0 u[2] = a_1 (a_1 b_0 + b_0) + b_0$$

...

$$y[n] = b_0 (1 + a_1 + \dots + a_1^n) = b_0 \sum_{k=0}^n a_1^k \quad (2.62)$$

Özetle elde edilen sonuç çözümü olan sonlu geometrik bir diziye gösterir.

$$\sum_{k=0}^L r^k = \begin{cases} \frac{1-r^{L+1}}{1-r}, & r \neq 1 \\ L+1, & r = 1 \end{cases} \quad (2.63)$$

Denklem (2.63)'ü kullanarak ve $a_1 \neq 1$ varsayarak birinci dereceden filtrenin adım cevabı Denklem (2.64)'teki eşitlik elde edilir.

$$y[n] = b_0 \frac{1 - a_1^{n+1}}{1 - a_1} u[n] \quad (2.64)$$

$|a_1| > 1$ olduğunda, a_1^n terimi, n 'e bağlı olmadan kararsız büyür ve kararsız bir duruma neden olur. $|a_1| < 1$ olduğunda, a_1^{n+1} terimi $n \rightarrow \infty$ 'a giderken sıfıra iner ve kararlı bir durum oluşur. $a_1 = 1$ olduğunda sınırsız büyüyen $b_0(n+1)$ 'in formunda Denklem (2.63)'teki özel bir durum çıktısı olur. $a_1 = -1$ olmasıyla çıkışın işareti değişir. Bu nedenle de marjinal olarak kararlı bir durum oluşur.

2.3.2.5. Bir IIR Filtresinin Sistem Fonksiyonu

IIR filtreleri için, genel olarak hem kutuplarda hem de sıfırlarda tamamen rasyonel bir fonksiyon olacaktır. Denklem (2.65)'te görüldüğü gibi zaman (dizi) domenindeki evrişim z -domenindeki çarpmaya karşılık gelir.

$$y[n] = x[n] * h[n] \xleftrightarrow{z} X(z)H(z) = Y(z) \quad (2.65)$$

Genel IIR fark denkleminin iki tarafında da geciktirme özelliğini kullanarak z -dönüşümüne başlanır.

$$Y(z) = \sum_{l=1}^N a_l \overbrace{z^{-l} Y(z)}^{\text{ZT}\{y[n-l]\}} + \sum_{k=0}^M b_k \overbrace{z^{-k} X(z)}^{\text{ZT}\{x[n-k]\}} \quad (2.66)$$

$$Y(z) / X(z) = H(z) \quad (2.67)$$

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{l=1}^N a_l z^{-l}} = \frac{b_0 + b_1 z^{-1} + \dots + b_M z^{-M}}{1 - a_1 z^{-1} - \dots - a_N z^{-N}} \quad (2.68)$$

Denklem (2.66) ve Denklem (2.67)'deki eşitlikler kullanılarak Denklem (2.68)'deki eşitlik elde edilir. $B(z)$ ile gösterilen pay polinomunun katsayıları, fark denkleminin ileriye dönük terimlerine karşılık gelir.

$A(z)$ olarak gösterilen payda polinomunun katsayıları, z^{-1} , $l > 0$ için fark denkleminin geri besleme terimlerine karşılık gelir. Genel IIR sistemi açısından bu vektörler Denklem (2.69)'daki gibi tanımlanır.

$$a = [1, -a_1, -a_2, \dots, -a_N]$$

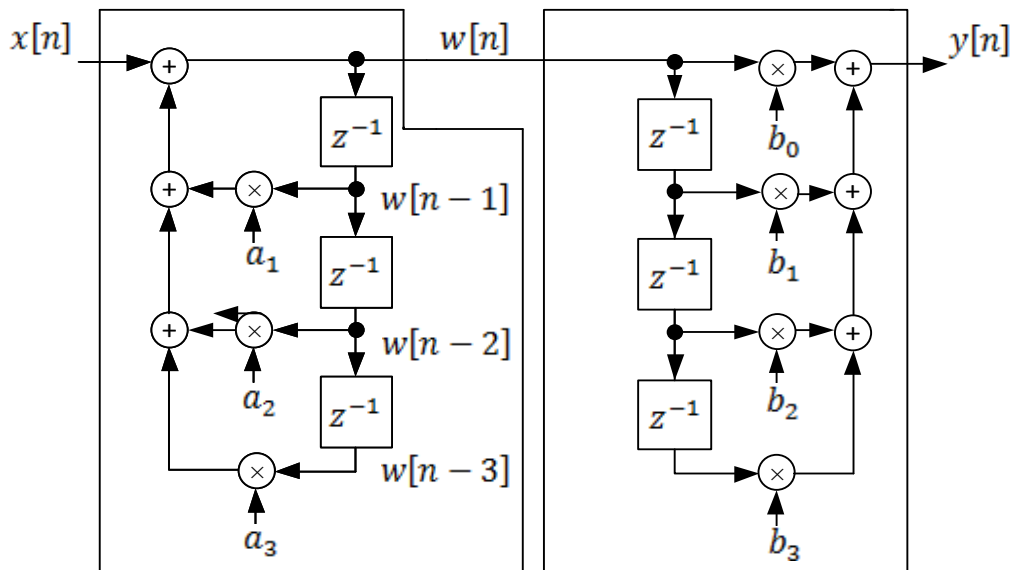
$$b = [b_0, b_1, \dots, b_M] \quad (2.69)$$

2.3.2.6. Sistem Fonksiyonları ve Blok Diyagram Yapıları

Doğrudan Form I yapısı, önce ileri besleme durumu ve ardından geri besleme durumundan oluşan iki sistemden oluşan bir dizi olarak görülebilir. Lineerlikten dolayı iki sistem kaskat bağlı gibi görülebilir ve Denklem (2.70)'deki eşitlik yazılabilir.

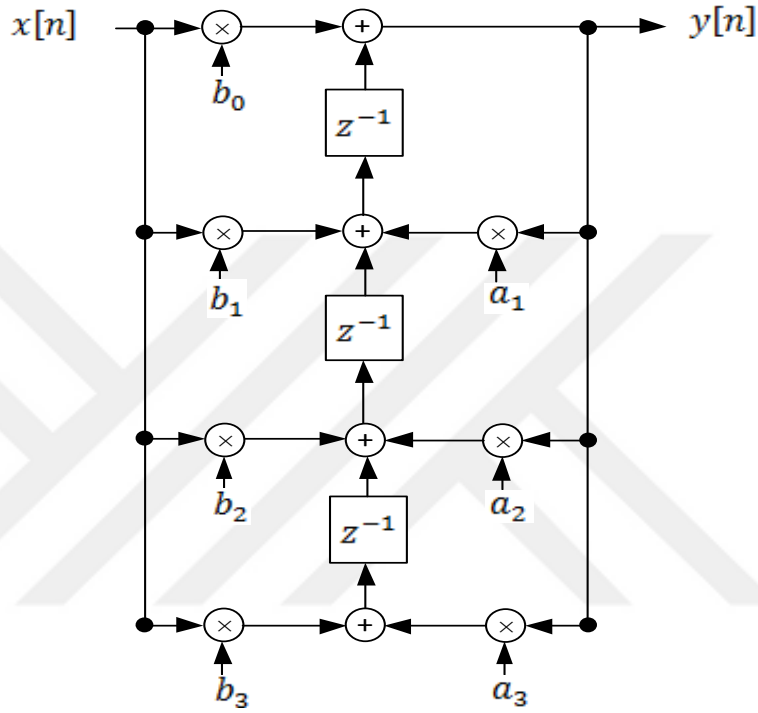
$$H(z) = B(z) \cdot \frac{1}{A(z)} = \frac{1}{A(z)} B(z) \quad (2.70)$$

Temsili IIR filtre blok diyagramı Şekil 2.14'te görülmektedir.



Şekil 2.14. Doğrudan Form II yapısı

Doğrudan Form II yapısı, Doğrudan Form I' den daha az gecikme bloğu kullanır. Filtre blok diyagramlarının bir özelliği tüm oklar tersine döndüğünde tüm dallanma noktaları toplama düğümleri olur; tüm toplama düğümleri de dallanma noktası olur. Giriş ve çıkış değişmesine rağmen sistem fonksiyonu değişmez. Doğrudan Form II yapısının transpoze edilmiş hali Şekil 2.15'te görülmektedir.



Şekil 2.15. Transpoze Doğrudan Form II

Dürtü cevabı ve sistem fonksiyonu z-dönüşümü ile ilişkilidir. IIR sistemleri için z-dönüşümü elde etmek için daha fazla çalışma gereklidir.

$$y[n] = ay[n-1] + x[n] \quad (2.71)$$

Denklem (2.71) için düşünüldüğünde dürtü cevabı Denklem (2.72)'deki gibi olmaktadır.

$$h[n] = a^n u[n] \quad (2.72)$$

Z-dönüşümü tanımından ;

$$H(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n \quad (2.73)$$

Denklem (2.73)'deki toplam, sonsuz bir geometrik serisidir. Bu serinin genel terimi Denklem (2.74)'de görülmektedir.

$$S = \sum_{n=0}^{\infty} r^n = \frac{1}{1-r}, |r| < 1 \quad (2.74)$$

Toplam formülü Denklem (2.74)'e uygulanırsa sonuç Denklem (2.75)'deki gibi olur.

$$H(z) = \sum_{n=0}^{\infty} (az^{-1})^n = \frac{1}{1-az^{-1}}, |z| > |a| \quad (2.75)$$

$|z| > |a|$ şartı z-dönüşümünün yalnızca z'nin bu değerleri için mevcut olduğunu belirtir. Z-düzlem bölgesi yakınsama bölgesi olarak bilinir. Denklem (2.76)'da z-dönüşüm ilişkisi gösterilmiştir.

$$a^n u[n] \xleftrightarrow{z} \frac{1}{1-az^{-1}} \quad (2.76)$$

Denklem (2.77)'deki formül z-dönüşümünü bulmak için kullanılabilir.

$$h[n] = b_0 (a_1)^n u[n] + b_1 (a_1)^{n-1} u[n-1] \quad (2.77)$$

Doğrudan doğrusallık ve gecikme özelliği kullanılarak Denklem (2.78)'deki eşitlik elde edilir.

$$H(z) = b_0 \frac{1}{1 - a_1 z^{-1}} + b_1 z^{-1} \frac{1}{1 - a_1 z^{-1}}$$

$$= \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \quad (2.78)$$

2.3.2.7. Kutuplar ve Sıfırlar

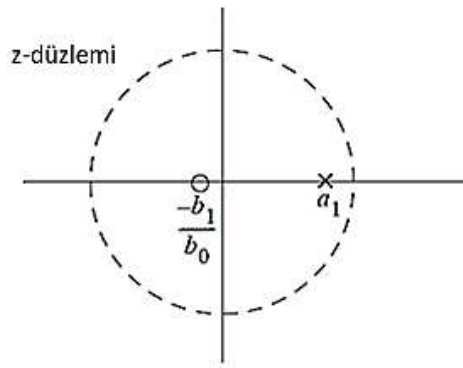
Payda payda polinomlarını çarpanlara ayırmak $H(z)$ 'nin kutuplarını ve sıfırlarını bulmayı sağlar. Birinci dereceden bir sistem için sadece cebirsel işleme ihtiyaç vardır. Bu cebirsel işlem Denklem (2.79)'da gösterilmiştir.

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \cdot \frac{z}{z} = \frac{b_0 z + b_1}{z - a_1} = b_0 \frac{z + b_1 / b_0}{z - a_1} \quad (2.79)$$

Denklem (2.80)'deki eşitlikte görülen tek kutup ve sıfır noktaları Şekil 2.16'daki birim çemberde görülmektedir.

$$p_1 = a_1$$

$$z_1 = -b_1 / b_0 \quad (2.80)$$



Şekil 2.16. Kutup ve sıfır noktaları

2.3.2.8. Sistem Kararlılığı

Nedensel LTI IIR sistemleri, sistemdeki tüm kutuplar birim çemberin içinde yer almaktadır. Başlangıçtaki dinlenme durumunda eğer sistem fonksiyonunun kutuplarının hepsi birim çemberin içinde yer alıyorsa sistem kararlıdır [28].



3. XSG VE MATLAB SIMULINK İLE FIR FİLTRE TASARIMI

3.1. Xilinx System Generator

MATLAB/Simulink ile arayüz oluşturur ve dijital olarak gerçekleştirilebilen Simulink bloklarının ve yaygın olarak kullanılan dijital programlama bloklarının verimli uygulamalarını sağlar [4].

System Generator bloğu, sistem ve simülasyon parametrelerinin kontrolünü sağlar ve kod üreticisine yardımcı olur. Xilinx blok kümesinden herhangi bir elemanı olan her Simulink modelinde en az bir System Generator bloğu olmalıdır. Bir modele bir System Generator bloğu eklendiğinde, kod oluşturma ve simülasyon yapılabilir [2].

XSG, öncelikle karmaşık DSP uygulamaları ile ilgilenmek için oluşturulmuştur, ancak aynı zamanda birçok görüntü işleme uygulamalarında da kullanılıyor. XSG'deki bloklar, donanım uygulamasında daha iyi bir yaklaşım için, sabit nokta tipinde Boole değerleri veya rasgele değerler ile çalışır. Aksine Simulink, çift duyarlılıklı kayan nokta sayısı ile çalışır. XSG blokları ve Simulink blokları arasındaki bağlantı ağ geçidi bloklarıdır [5].

Xilinx System Generator'ün avantajı, döngüde Alanda Programlanabilir Kapı Dizileri (FPGA) donanımını Simulink simülasyonları ile birlikte kullanabilmesidir. Bu tasarım için, Xilinx panosuna programlanan tasarımı çalıştırmak için donanım eş simülasyonu kullanılır. Donanım eş simülasyonu, Sistem Generator araçları tarafından sağlanır. Bu araçlar, bilgisayardaki tasarımla bağlantılı olan Simulink simülasyonu sırasında tasarım donanımını kontrol eder. Tasarım ve FPGA kartı, JTAG kablosu kullanılarak bağlanır. Giriş bilgisayardan beslenir ve çıkış bilgisayara geri çevrilir [6]. Xilinx System Generator'de tasarım simüle edildikten sonra, FPGA uygulaması için bir donanım eş simülasyon bloğu oluşturulur. Oluşturulan bu blok daha sonra FPGA'a yüklemek için kullanılır [7]. Bloğun bağlantı noktaları, orijinal tasarım bağlantı noktalarına benzer adlara ve türlere sahiptir [6].

Simulink tasarımının Xilinx bölümündeki girişler Xilinx Gateway In bloklarıdır. Bu bloklar Simulink double, sabit nokta ve tamsayı veri türlerini System Generator sabit nokta türüne dönüştürür. System Generator tarafından oluşturulan HDL tasarımı, üst düzey giriş portunun her bloğu tarafından tanımlanır. Xilinx bölümündeki çıkışlar, Xilinx Gateway Out bloklarıdır [2].

FPGA, temel mantık hücrelerinin azaltılmış bir şekilde uygulanmasına izin vermesi nedeniyle dijital filtrelerin uygulanması için uygun özelliklere sahip bir cihazdır. Ayrıca, derleyiciler ve sentezleyiciler, çarpımların uygulanmasını hem boyut hem de hız sabitleri ile optimize eder [8].

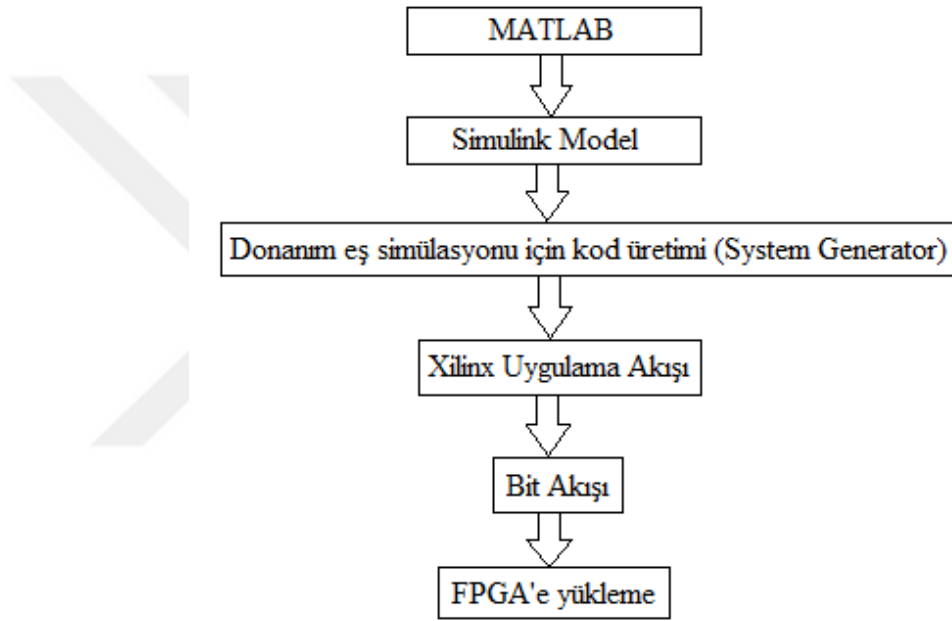
Xilinx'ten FPGA tasarımı için Math Works model tabanlı Simulink tasarım ortamının kullanılmasını sağlayan bir DSP tasarım aracı olan System Generator, ISE® Design Suite'in bir parçasıdır ve uygulamaya özel tasarım için toplayıcılar, çarpanlar, kayıtlar, filtreler ve hafızalar gibi Xilinx DSP Blokseti sağlar. Bu bloklar, seçilen cihaz için optimize edilmiş sonuçlar sunmak üzere Xilinx IP çekirdek jeneratörlerini kullanır. System Generatoru kullanmak için Xilinx FPGA veya RTL tasarım yöntemleri ile ilgili deneyim gerekli değildir. Tasarımlar, bir Xilinx'e özgü Blockset kullanılarak Simulink modelleme ortamında gerçekleştirilir. Bir FPGA programlama dosyası oluşturmak için, sentez ve yer dahil olmak FPGA uygulama adımlarının tümü otomatik olarak gerçekleştirilir. Donanım uygulaması için Xilinx sistem jeneratörünün kullanılmasının avantajı, Xilinx Blockset'in, MATLAB Simulink blokları tarafından sağlanan piksel vektörü ile FPGA modülünün eş simüle edilmesinde yardımcı olan MATLAB Simulink ile yakın entegrasyon sağlamasıdır [9].

Büyük boyutlu bir filtrenin gerçekleştirilmesinde, ortak dijital sinyal işlemcilerinin hız, maliyet, esneklik vb. bakımından performansını etkileyen birçok karmaşık hesaplama ihtiyacı duyulmasına rağmen, FPGA, genel sistem performansını arttırmak için yoğun sayısal sinyal işleme algoritmalarını kullanarak üretken hale gelmiştir.

FPGA kitlerinde DSP uygulamalarını desteklemek için son yazılım aracı geliştirmesinde dikkate değer bir ilerleme kaydedilmiştir. System Generator, Simulink modelinden bir FPGA devresinin üretimini içerecek şekilde genişleten Xilinx FPGA'lar için bir tasarım aracıdır. System Generator, DSP işlevleri, hafızalar, aritmetik ve mantıksal işlevler için daha güçlü Simulink blokları sağlar. Ayrıca, yüksek düzeyli modelleme ve otomatik kod oluşturma özelliğini destekleyerek, System Generator , donanım merkezli unsurlar ile matematiksel soyutlama arasındaki etkileşimi belirlemek için yeni fırsatlar oluşturur [2].

3.2. Tasarımı Uygulama Aşamaları

Bu tez çalışmasında, MATLAB kodu ile örnekleme oranı 64000 olan 256 örneklik genliği 15 olan 1 kHz frekanslı ve genliği 2 olan 15 kHz frekanslı iki sinüsoidal sinyal karıştırılmıştır. Karıştırılan bu sinyaller Simulink ve Xilinx System Generator'e giriş sinyali olarak aktarılmak üzere .mat formatında kaydedilmiştir. MATLAB kodunda.mat formatında kaydedilen bu sinyaller From File bloğu ile istenilen sinyalin filtrelenmesi için Simulink'e ve Xilinx System Generator'e aktarılmıştır.

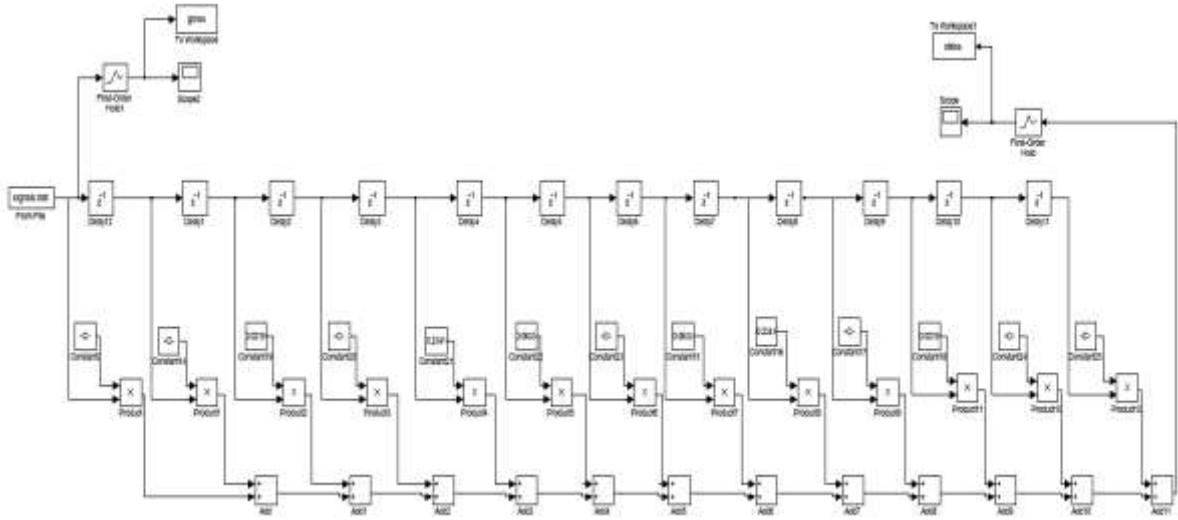


Şekil 3.1. FIR filtre uygulamasının tasarım akış şeması

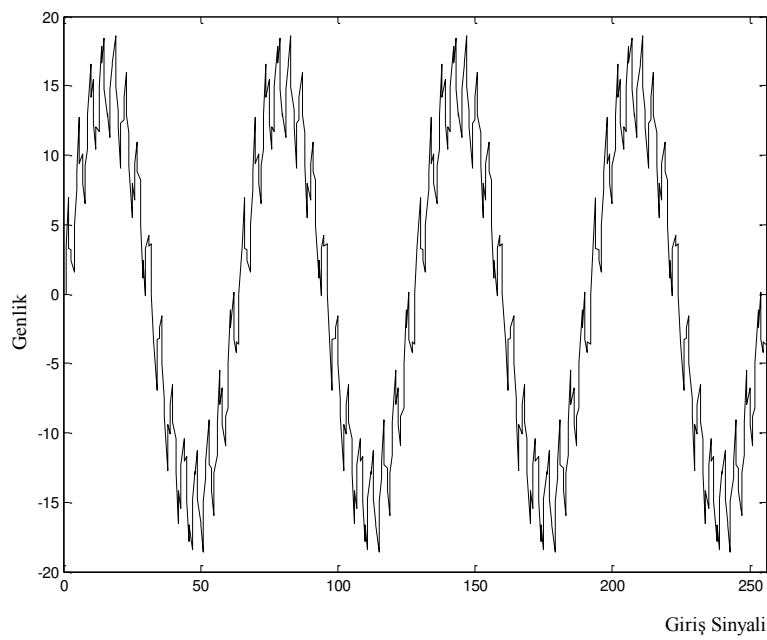
Şekil 3.1.'de FIR filtre uygulamasının tasarım akış şeması görülmektedir [29]. MATLAB'da 6 kHz ile 20 kHz arasındaki frekansları bastırmak amacıyla 14. dereceden bant durduran FIR filtre kodu ile 13 adet simetrik filtre katsayıları elde edilmiştir. Elde edilen bu katsayı değerleri Simulink'te modeli oluşturulan Doğrudan Form FIR filtre blok yapısındaki Coefficient blokları içerisine girilmiştir.

Çarpma, toplama ve birim gecikme blokları ile Şekil 3.2' de görülen doğrudan form FIR filtre blok yapısı oluşturulmuştur. Bu blok yapısı oluşturulurken From File bloğu kullanılarak alınan karmaşık giriş sinyali ilk katsayı ile çarpılmak üzere Product bloğuna aktarılmıştır.

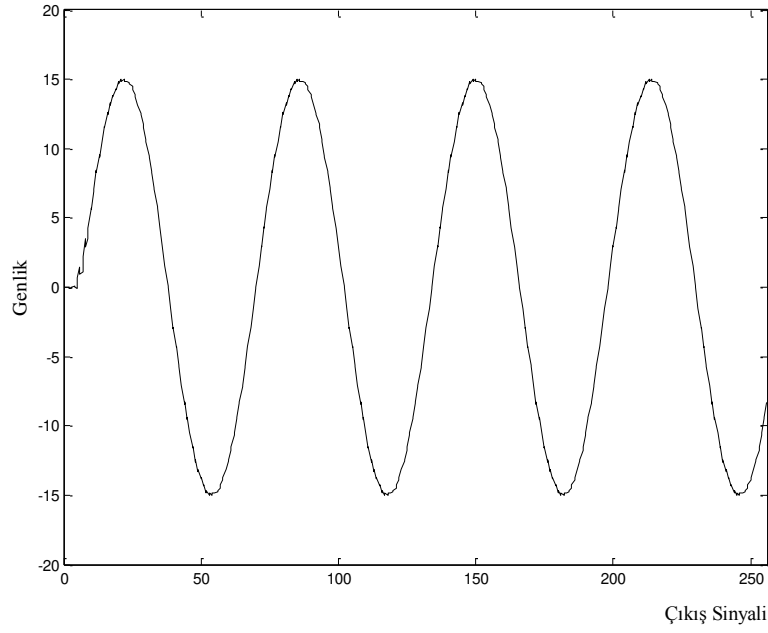
Giriş sinyaline Delay bloğu eklenerek bir birim gecikme uygulanmış giriş sinyali ile de ikinci katsayı çarpılmıştır. Bu iki çarpım sonucu toplanmak üzere Add bloğuna aktarılmıştır. Birim gecikme bloğunun ardına 11 adet birim gecikme bloğu daha kullanılarak aynı şekilde diğer katsayılarla sırayla çarpma işlemleri yapıp her çarpım sonucu bir önceki Add bloğu çıkışıyla toplanarak çıkış elde edilmiştir. Simulink'te elde edilen giriş sinyali-genlik grafiği Şekil 3.3'te görülmektedir.



Şekil 3.2. Doğrudan Form FIR filtre blok yapısının Simulink modeli



Şekil 3.3. Simulink'te Elde Edilen Giriş Sinyali - Genlik Grafiği

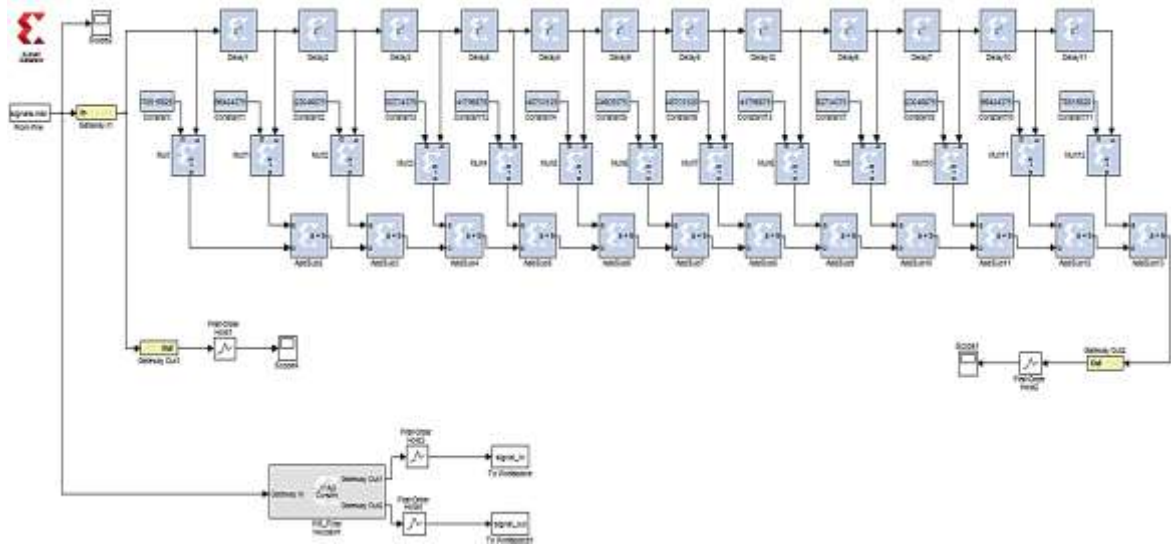


Şekil 3.4. Simulink'te Elde Edilen Çıkış Sinyali - Genlik Grafiği

Simulink'te tasarlanan FIR filtre ile 15 kHz frekanslı sinyal bastırılıp, 1 kHz frekanslı sinyal geçirilmiştir. Daha sonra From file ile alınan karmaşık sinyal FIR filtre blok yapısına giriş sinyali olarak verilmiş ve çıkışta genliği 15 olan 1 kHz frekanslı sinyal elde edilmiştir. Elde edilen bu dijital sinyal First Order Hold bloğu kullanılarak analog sinyale çevirmiştir. Simulink'te elde edilen çıkış sinyali-genlik grafiği Şekil 3.4'te görülmektedir.

Simulink'te simülasyonu elde edilen filtreyi FPGA ile gerçeklemek için System Generator araçları kullanılarak Şekil 3.5'te görülen Doğrudan Form FIR filtre blok yapısı oluşturulmuştur.

FPGA kartı olarak Virtex 5 modeli kullanılmıştır. System Generator kullanılarak MATLAB 2012a başlatılıp modeli oluşturulduktan sonra Hardware Co-simulation bloğunu çalıştırabilmek için Şekil 3.6'da görüldüğü gibi Virtex 5 kartı JTAG kablosu ile bilgisayara takılmıştır.



Şekil 3.5. Doğrudan Form FIR filtre blok yapısının System Generator modeli



Şekil 3.6. FPGA kartı ile bilgisayar bağlantısı

FPGA kartını tanıtmak için modelde yer alan System Generator bloğuna tıklayıp açılan Şekil 3.7’deki pencereden sıra ile aşağıdaki işlemler yapılmıştır.

Compilation → Hardware Co-Simulation → New Compilation Target seçeneğine tıklayıp yeni açılan Target Board Information penceresinde;

Board Name : Virtex 5

Frequency : 100

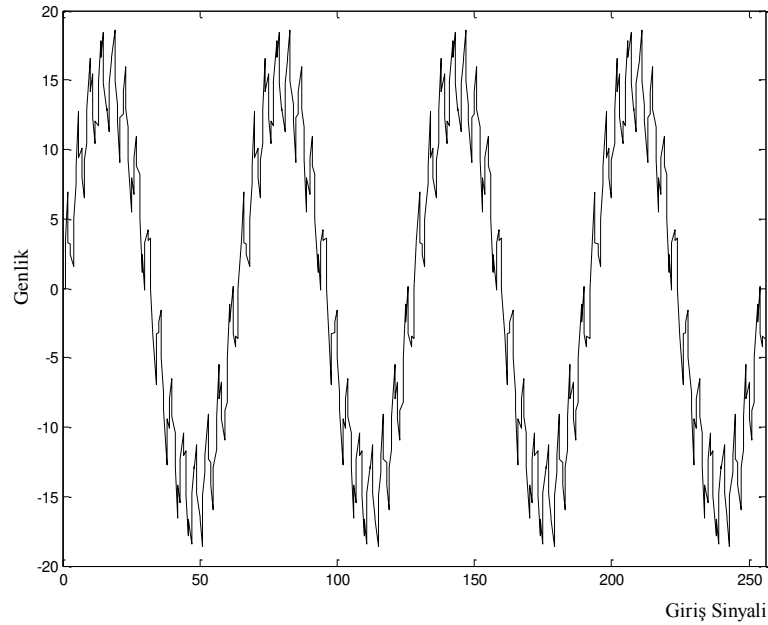
Pin Location : AG18 olarak girilmiştir.

Detect’e basılıp IR Lengths çekildikten sonra , Boundary Scan Posistion değeri de 1 olarak girilmiştir.

Target Board Information penceresinde, Targetable Devices başlığı altında Add yazan kısma tıklanır, buradan sırasıyla, Virtex 5 → xc5vlx110t → -1 (ff1136) seçilmiştir. Virtex 5.zip dosyası kaydedilmiştir.

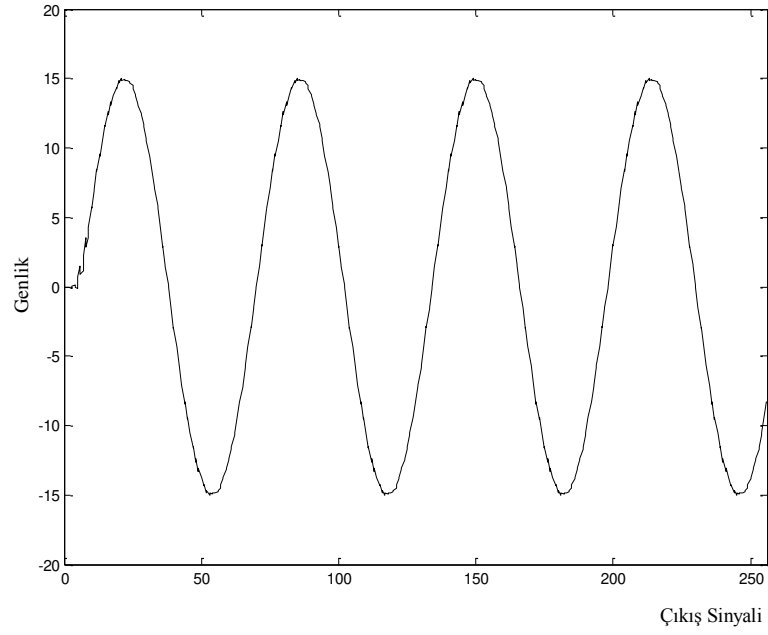


Şekil 3.7. System Generator blok penceresi

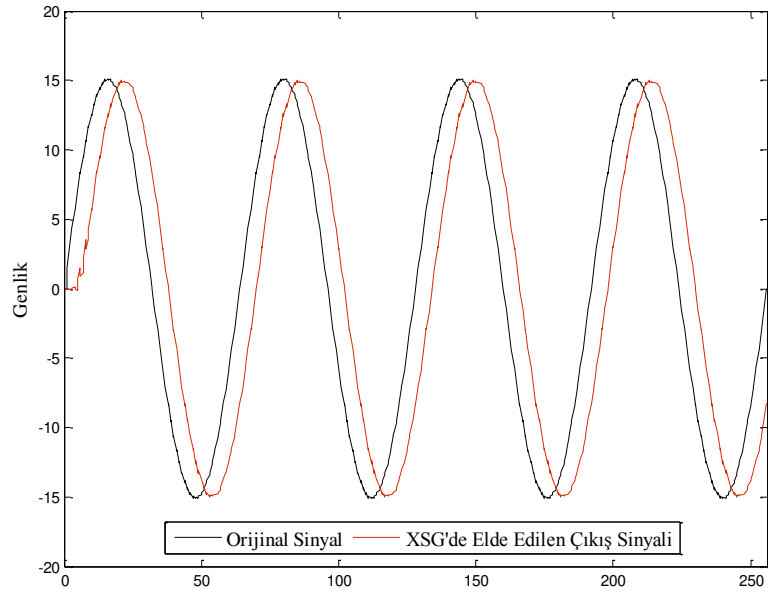


Şekil 3.8. Xilinx System Generator’de Elde Edilen Giriş Sinyali - Genlik Grafiği

Xilinx System Generator’de elde edilen giriş sinyali-genlik grafiği Şekil 3.8’de görülmektedir.



Şekil 3.9. Xilinx System Generator’de Elde Edilen Çıkış Sinyali - Genlik Grafiği



Şekil 3.10. Orijinal Sinyal ve XSG’de Elde Edilen Çıkış Sinyali Fark Grafiği

Filtreleme işlemi sonucunda elde edilen sinyal dijital olduğu için bu sinyali analoğa çevirmek için , bilgisayardan yeni bir numara çıkana kadar girişi belirleyen bir tutma devresiyle birleştirilmiş bir DA dönüştürücüsü olan First Order Hold bloğu kullanılmıştır.

First Order Hold bloğu ile uygun tempoda birinci dereceden bir polinom kullanılarak girdiler bilgisayardan gelen o andaki ve önceki çıktılarından hesaplanarak yumuşak sinyal elde edilmiştir [30]. Xilinx System Generator’de elde edilen çıkış sinyali-genlik grafiği Şekil 3.9’da, Orijinal sinyal ve XSG’de elde edilen çıkış sinyali fark grafiği Şekil 3.10’da görülmektedir.

Xilinx System Generator’de tasarlanan modeli FPGA’e yüklemek için Hardware Co-simulation bloğu oluşturulmuştur. Bu bloğu oluşturmak için yine model üzerindeki System Generator bloğuna tıklayıp Hardware Co-simulation listesine kaydolun Virtex 5 modeli seçilerek Generate butonuna basılmıştır. Yeni bir pencerede oluşan Hardware Co-simulation bloğu tasarlanan modele alınarak giriş kısmı bağlanıp çıkışına First Order Hold bloğu yerleştirilip Run tuşuna basılıp analog giriş sinyali ve filtrelenmiş çıkış sinyali elde edilmiştir. Elde edilen bu sinyallerin modelde Hardware Co-simulation bloğu oluşturulmadan önce bağlı olan Scope bloklarında görülen giriş ve çıkış sinyalleriyle aynı olduğu teyit edilmiştir.

Alanda Programlanabilir Kapı Dizisi (FPGA) tasarım teknolojisi, sinyal işleme algoritmalarını değerlendirmek ve uygulamak için tercih edilen platform haline gelmiştir. FPGA yaklaşımının dijital filtre uygulamasına getirdiği avantajlar, geleneksel DSP yongalarında elde edilenden daha yüksek örnekleme oranları, orta hacimli uygulamalar için uygulamaya özel entegre bir devreden (ASIC) daha düşük maliyetler ve alternatif yaklaşımlardan daha fazla esneklik içerir. Mevcut birçok FPGA mimarisi sistem içinde olduğundan programlanabilir, gerektiğinde farklı işlevler uygulamak için cihazın yapılandırması değiştirilebilir.

FPGA, düşük güçte, alan açısından verimli yeniden yapılandırılabilir dijital sinyal işleme mimarisidir. FPGA çözümü tasarımda tam bir esneklik sunar. Çip sayısını azaltarak, sistemin genel güvenilirliğini artırır, düşük alan, düşük güç ve FIR filtrelerinin yüksek hızda uygulanmasını sağlar. Aynı zamanda filtre gecikmesini azaltır [31].

4. SONUÇLAR

Bu yüksek lisans tez çalışmasında filtre tasarım yöntemleri hakkında bilgi verilip, Xilinx System Generator ve Simulink'te örnek bir filtre modeli tasarlanmıştır. Öncelikle farklı genlik ve frekanstaki iki sinüs sinyal toplanarak bir sinyal karmaşası elde edilmiştir ve bu sinyal karmaşasından istenilen frekanstaki sinyalin filtrelenmesi için oluşturulan filtre modelinin girişine verilmiştir. Simulink'te simülasyonu ve FPGA kartına yüklenerek gerçekleştirilmesi yapılan bu örnek filtreleme sonucunda istenilen frekanstaki sinyal elde edilmiştir. Elde edilen çıkış sinyalleri karşılaştırıldığında her iki sonucun da aynı olduğu gözlemlenmiştir. Filtre çeşidi olarak bant durduran FIR filtre yapısı kullanılmıştır. Bu yapının oluşturulmasında; güç tüketimi, çalışma hızı ve gerçek zamanlı olarak maliyet açısından genel filtre yapılarından daha iyi bir performans sağlayan Doğrudan Form FIR filtre yapısı dikkate alınmıştır. Tasarımın gerçekleştirilmesi için maksimum bellek kullanımı 640 MB olup kullanılan Flip Flop sayısı 809'dur.

Bu yapı tasarlanırken sabit nokta veri türü kullanılmıştır. Sabit nokta uygulamaları daha az kaynak ve daha az güç tüketimi bakımından kayan noktadaki eşdeğerlerinden daha verimlidir. Tasarlanan bu filtrede kullanılan yapı ve veri türünün maliyeti düşürmesi, gecikme süresini ve güç tüketimini azaltması gibi sağladığı avantajlarla dijital sinyal işleme ve birçok alanda rahatlıkla tasarlanıp kullanılabilir.

5. KAYNAKLAR

- [1] **Z. B. GARIP, and A. F. BOZ**, 2018, “The FIR Filter Design based on Genetic Algorithm”, Balkan Journal of Electrical & Computer Engineering , vol.6, no.custom number, pp. 33–36.
- [2] **Mr. Mayur B. Kachare, Prof. D. U. Adokar**, 2016, “A Distributed Arithmetic (DA) Based Digital FIR Filter Realization”, International Conference on Global Trends in Engineering, Technology and Management (ICGTETM), pp. 24-28.
- [3] **Kendall Su**, 2002, “Analog Filters”, Kluwer Academic Publishers.
- [4] **Suraj R. Gaikwad ve Gopal S. Gawande**, 2014, “Design and Implementation of Efficient FIR Filter Structures using Xilinx System Generator”, International Journal of scientific research and management (IJSRM), vol. 2, issue 3, pp. 599-604.
- [5] **Mrs. S. Allin Christe, Mr.M.Vignesh, Dr.A.Kandaswamy**, 2011, “An Efficient FPGA Implementation of MRI Image Filtering and Tumour Characterization Using Xilinx System Generator ”, International Journal of VLSI design & Communication Systems (VLSICS) , vol.2,no.4, pp. 95–109.
- [6] **N.M.Zawawi, M.F.Ain, S.I.S.Hassan, M.A.Zakariya, C.Y.Hui ve R.Hussin**, 2008, “Implementing WCDMA Digital Up Converter In FPGA”,IEEE International RF and Microwave Conferance Proceedings, Kuala Lumpur , Malaysia.
- [7] **Taoufik Saidani, Mohamed Atri, Yahia Said ve Rached Tourki**, 2012, “FPGA Real Time Acceleration For Discrete Wavelet Transform Of The 5/3 Filter For Jpeg 2000 Standard”, 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), vol.2, no.1, pp. 393-399.
- [8] **Edilberto Carlos Vivas González, Diego Mauricio Rivera Pinzón ve Edwar Jacinto Gómez**, 2014, “Implementation and Simulation of IIR Digital Filters in FPGA Using

MatLab System Generator”, IEEE 5th Colombian Workshop on Circuits and Systems (CWCAS).

[9] **Neha. P. Raut, Prof.A.V.Gokhale**, 2013, “**FPGA Implementation for Image Processing Algorithms Using Xilinx System Generator**”, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) , vol.2, issue.4, pp. 26-36.

[10] **Douglas Self**, 2018, “The Design of Active Crossovers”, Focal Press.

[11] **Suvadip Roy, L. Srivani, D. Thirugnana Murthy**, 2015, “Digital Filter Design Using FPGA”, International Journal of Engineering and Innovative Technology (IJEIT), vol.5, issue.4, pp. 78-82.

[12] **W. Alan Davis**, 2010, “Radio Frequency Circuit Design”, Wiley-IEEE Press.

[13] **Ian Grout**, 2011, “Digital Systems Design with FPGA and CPLDs”, Newnes.

[14] **Abdullah Ateş, Gürkan Kavuran, Barış Baykant Alagöz, Celalettin Yeroglu**, 2016, “Improvement of IIR filter discretization for fractional order filter by discrete stochastic optimization”, 39th International Conference on Telecommunications and Signal Processing (TSP), Vienna, Austria.

[15] **Maloth Santhoshi, Mrs. E. P. Vanetha ve P. Srikanth**, 2014, “An Efficient Implementation of Digit FIR Filters using Memory based Realization”, Double Blind Peer Reviewed International Research Journal, vol.14, issue.9, pp. 7-12.

[16] **Raija Lehto, Tarja Taurén ve Olli Vainio**, 2011, “Recursive FIR Filter Structures on FPGA”, Microprocessors and Microsystems , vol. 35, issue 7, pp. 595–602.

[17] **Harish V. Dixit, Dr. Vikas Gupta**, 2012, “IIR filters using Xilinx System Generator for FPGA Implementation”, International Journal of Engineering Research and Applications (IJERA) , vol. 2, issue 5, pp. 303-307.

[18] **Naila Allakhverdiyeva**, 2016, “Application of Neural Network for Digital Recursive Filter Design”, IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan.

[19] **Pankaj R. Ambilduke ve Manish Kumar Gurjar**, 2016, “Performance Comparison for Efficient Digital Decimator Filter Structures”, IJRET: International Journal of Research in Engineering and Technology, vol. 5, issue 10, pp. 29-36.

[20] **Disha Yadav ve Ashutosh Nandi**, 2016, “Comparative analysis of digital IIR filter using add and shift method on Xilinx platform”, International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) , Kumaracoil, India.

[21] **Mrs.Bhagyalakshmi N, Dr.Rekha K R ve Dr.Nataraj K R**, 2014, “System Level Tools for Designing FIR Filter on FPGA”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 3, issue 2, pp. 7149- 7154.

[22] **Gurpadam Singh ve Neelam R. Prakash**, 2017, “FPGA Implementation of Higher Order FIR Filter”, International Journal of Electrical and Computer Engineering (IJECE), vol. 7, no. 4, pp. 1874-1881.

[23] **Rajib Das, Amrita Guha ve Ananya Bhattacharya**, 2016, “FPGA Based Higher Order FIR Filter Using XILINX System Generator”, International conference on Signal Processing, Communication, Power and Embedded System (SCOPEs) , Paralakhemundi, India.

[24] **Kumudini Sahu ve Rahul Sinha**, 2015, “FIR Filter Designing using MATLAB Simulink and Xilinx system Generator”, International Research Journal of Engineering and Technology (IRJET), vol. 1, issue 8, pp. 1608-1611.

[25] **Raaed Hassan**, 2012, “Field Programmable Gate Array Based FIR filter design”, Baghdad.

[26] **Manish B. Trimale, Purushottam G. Chilveri**, 2017, “FIR Filter Implementation on FPGA Using MCM Design Technique”, International Conference on Circuits, Controls, and Communications (CCUBE), India.

[27] **Abdulbasit M. Sabaawi**, 2016, “FIR Filter for Audio Signals Based on FPGA: Design and Implementation”, American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS), vol. 24, no. 1, pp. 63-75.

[28] **Steve Winder**, 2002, “Analog and digital filter design”, ELSEVIER SCIENCE & TECHNOLOGY.

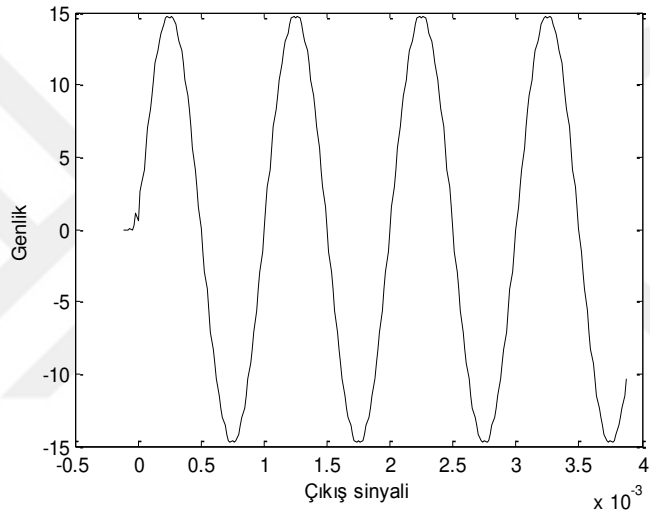
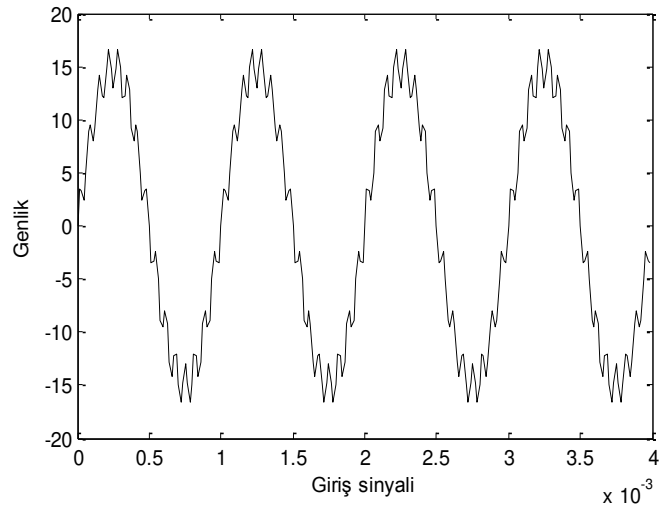
[29] **Mr.P.C.Bhaskar ve Dr.M.D.Uplane**, 2016, “High Frequency Electromyogram Noise Removal from Electrocardiogram Using FIR Low Pass Filter Based On FPGA”, Global Colloquium in Recent Advancement and Effectual Researches in Engineering, Science and Technology (RAEREST)

[30] **Wittenmark, Björn; Årzén, Karl-Erik; Åström, Karl Johan** , 2002, “Computer Control: An Overview”, IFAC PROFESSIONAL BRIEF.

[31] **Rakhi Thakur ve Kavita Khare**, 2013, “High Speed FPGA Implementation of FIR Filter for DSP Applications”, International Journal of Modeling and Optimization, vol. 3, no. 1, pp. 92-94.

6. EKLER

```
sample_rate = 64000;
nsamples = 256;
F = [1 15] * 1000;
A = [15 2];
t = (0:nsamples-1) / sample_rate;
signal = A * sin(2*pi*F*t);
ts=timeseries(signal)
save('signals','ts','-v7.3')
cutoff_hz = [6000 20000];
nyq_freq = sample_rate / 2;
cutoff_norm = cutoff_hz / nyq_freq;
order = 14;
fir_coeff = fir1(order, cutoff_norm, 'stop');
filtered_signal = filter(fir_coeff, 1, signal);
group_delay = median(grpdelay(fir_coeff));
group_delay_s = group_delay / sample_rate;
figure(1)
plot(t, signal)
figure(2)
plot(t-group_delay_s, filtered_signal, 'g-')
```



ÖZGEÇMİŞ

1994 yılında Elazığ'da doğan Aybike ÜSTÜNDAĞ, lise öğrenimini Elazığ Anadolu Lisesi'nde tamamlamıştır. 2012 yılında kazandığı Fırat Üniversitesi Elektrik Elektronik Mühendisliği Bölümü'nden 2016 yılında mezun olmuştur. 2016 yılında, Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda yüksek lisans öğrenimine başlamıştır. 2018 yılından beri Munzur Üniversitesi Elektrik Elektronik Mühendisliği Bölümü'nde Araştırma Görevlisi olarak çalışmaktadır.

