

T.C.
BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM
DALI



SU ALTI AKUSTİĞİNDE ADAPTİF FİLTRE UYGULAMASI
VE FPGA ÜZERİNDE GERÇEKLENMESİ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ YÜKSEK
LİSANS TEZİ

EMRE KARAKAYA

TEZ DANIŞMANI

Doç. Dr. Oktay AYTAR

İKİNCİ TEZ DANIŞMANI

Dr. Mehmet Ali ÇAVUŞLU

BOLU, HAZİRAN 2025

KABUL VE ONAY SAYFASI

Emre KARAKAYA tarafından hazırlanan “**SU ALTI AKUSTİĞİNDE ADAPTİF FİLTRE UYGULAMASI VE FPGA ÜZERİNDE GERÇEKLENMESİ**” adlı tez çalışması jürimiz tarafından Elektrik Elektronik Mühendisliği Ana Bilim Dalı Elektrik Elektronik Mühendisliği Programında Yüksek Lisans olarak oy birliğiyle kabul edilmiştir. 16/06/2025

Jüri Üyeleri

İmza

Danışman
Doç. Dr. Oktay AYTAR
Bolu Abant İzzet Baysal Üniversitesi

.....

Üye
Prof. Dr. Suhap ŞAHİN
Kocaeli Üniversitesi

.....

Üye
Doç. Dr. Nihat DALDAL
Bolu Abant İzzet Baysal Üniversitesi

.....

Lisansüstü Eğitim Enstitüsü Onayı

Prof. Dr. İbrahim KÜRTÜL
Lisansüstü Eğitim Enstitüsü Müdürü

ETİK BEYAN

Bolu Abant İzzet Baysal Üniversitesi, Lisansüstü Eğitim Enstitüsü Tez Yazım Kılavuzu ve Şablonuna uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
 - Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
 - Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
 - Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
 - Bu tezde sunduğum çalışmanın özgün olduğunu bildirir
- aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Teze ilişkin Turnitin adlı programında enstitü müdürlüğünce belirlenen filtrelemeler uygulanarak alınmış olan benzerlik raporuna göre, tezin benzerlik oranı %30'u geçmemektedir.

.....

EMRE KARAKAYA

ÖZET

**SU ALTI AKUSTİĞİNDE ADAPTİF FİLTRE UYGULAMASI VE FPGA
ÜZERİNDE GERÇEKLENMESİ
YÜKSEK LİSANS TEZİ
EMRE KARAKAYA
BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ YÜKSEK
LİSANS PROGRAMI
(TEZ DANIŞMANI: DOÇ. DR. OKTAY AYTAZ)
(İKİNCİ DANIŞMAN: DR. MEHMET ALİ ÇAVUŞLU)
BOLU, HAZİRAN - 2025
XIII + 40 sayfa**

Yapılan bu çalışmada, su altı akustik alanında kullanılan geleneksel sayısal filtreleme yöntemlerinin yerine adaptif filtre algoritmalarından olan En Küçük Ortalama Kare (LMS) ve Özyinelemeli En Küçük Kareler (RLS) algoritmalarının performansı incelenmiş olup, uygulama yöntemi gösterilmektedir. Gerçekleme ortamı olarak ise sinyal işleme uygulamalarında yaygın şekilde kullanılan Alanda Programlanabilir Kapı Dizisi (FPGA) donanımı tercih edilmiştir. Çalışmada, önce LMS ve RLS algoritmalarının matematiksel olarak donanımda gerçekleştirilmesi için analizleri yapılmış, ardından su altı ortamında kaydedilen haberleşme sinyalleri kullanılarak, her iki algoritma FPGA donanımında gerçekleştirilmiştir. Elde edilen sonuçlara göre LMS algoritması 8,85 dB olan Sinyal Gürültü Oranı (SNR) değerini 34, 71 dB seviyesine, RLS algoritması ise 28, 69 dB seviyesine çıkarmıştır. FPGA kaynak kullanımında ise LMS algoritması %2,68 Arama Tablosu (LUT), %4,7 Flip-Flop (FF) kullanımı gerçekleştirmiş, RLS algoritması ise %1,06 LUT, %1,63 FF kullanımı gerçekleştirmiştir. Diğer kaynak kullanımlarında ise her iki algoritma da %0,42 Sayısal Sinyal İşleme (DSP), %1,9 Giriş-Çıkış (IO) ve %3,13 Küresel Ara Bellek (BUFG) kullanmıştır. Elde edilen bu sonuçlara göre LMS algoritması daha fazla SNR iyileşmesi sağlarken RLS algoritması daha az kaynak tüketiminde bulunmuştur.

Yapılan bu çalışmada, LMS ve RLS algoritmalarının su altı haberleşme verilerindeki SNR değerini iyileştirdiği, su altı akustik alanındaki uygulamalarda kullanılabilirliği ve bu alandaki çalışmalara katkı sağlayabileceği gösterilmiştir.

ANAHTAR KELİMELEER: LMS, RLS, FPGA, Su altı akustik,

ABSTRACT

ADAPTIVE FILTER APPLICATION IN UNDERWATER ACOUSTICS AND IMPLEMENTATION ON FPGA

MSC THESIS

EMRE KARAKAYA

BOLU ABANT IZZET BAYSAL UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL-ELECTRONICS ENGINEERING

(SUPERVISOR: ASSOC. PROF. DR. OKTAY AYTAZ)

(CO-SUPERVISOR: DR. MEHMET ALI CAVUSLU)

BOLU, JUNE 2025

XIII + 40 pages

In this study, the performance of the Least Mean Square (LMS) and Recursive Least Square (RLS) algorithms, which are adaptive filter algorithms instead of traditional digital filtering methods used in the field of underwater acoustics, is investigated and the application method is shown. Field Programmable Gate Array (FPGA) hardware, which is widely used in signal processing applications, was chosen as the implementation environment. In the study, firstly, LMS and RLS algorithms were analyzed mathematically for their implementation in hardware, then both algorithms were implemented in FPGA hardware using communication signals recorded in the underwater environment. According to the results obtained, the LMS algorithm increased the Signal to Noise Ratio (SNR) value from 8.85 dB to 34.71 dB, and the RLS algorithm increased it to 28.69 dB. In terms of FPGA resource usage, the LMS algorithm used 2.68% Lookup Table (LUT), 4.7% Flip-Flop (FF), while the RLS algorithm used 1.06% LUT and 1.63% FF. In other resource usage, both algorithms used 0.42% Digital Signal Processing (DSP), 1.9% Input-Output (IO) and 3.13% Global Buffer Memory (BUFG). According to these results, the LMS algorithm provided more SNR improvement while RLS algorithm consumed less resources.

In this study, it has been shown that LMS and RLS algorithms improve the SNR value in underwater communication data, can be used in applications in the field of underwater acoustics and can contribute to the studies in this field.

KEYWORDS: LMS, RLS, FPGA, Underwater acoustic

İÇİNDEKİLER

Sayfa

ÖZET.....	iv
ABSTRACT	v
İÇİNDEKİLER	vi
ŞEKİL LİSTESİ.....	viii
TABLO LİSTESİ	ix
FOTOĞRAF LİSTESİ	x
KISALTMA VE SEMBOLLER LİSTESİ	xi
TEŞEKKÜR	xiii
1. GİRİŞ.....	1
2. MATERYAL ve YÖNTEM.....	4
2.1 Su Altı Akustiği	4
2.2 Su Altı Ortam Gürültüsü.....	4
2.3 Adaptif Filtre Teorisi	5
2.3.1 LMS Algoritması.....	7
2.3.2 RLS Algoritması.....	8
2.4 FPGA	9
2.5 Sayı Sistemleri	10
2.5.1 Sabit Noktalı Sayı Sistemi	10
2.5.2 Kayan Noktalı Sayı Sistemi.....	11
2.6 Donanım Tanımlama Dilleri.....	11
2.7 NEXYS 4 DDR Geliştirme Kartı	12
2.8 UART Haberleşme Protokolü	14
2.9 Akustik Veri Kaydı Deney Düzeneği	15
2.9.1 Transdüser	18
2.10 Ping Sinyalinin Üretilmesi.....	19
2.11 FPGA Üzerinde Adaptif Filtreleme İşleminin Gerçeklenmesi.....	20
2.11.1 LMS Algoritması Başlangıç Parametreleri	20
2.11.2 RLS Algoritması Başlangıç Parametreleri	21
2.12 Performans Karşılaştırma Kriterleri.....	21
2.12.1 SNR.....	22
2.12.2 FPGA Kaynak Tüketimi	22
3. BULGULAR	23
3.1 LMS Algoritması Gerçeklenme Sonucu.....	23
3.2 RLS Algoritması Gerçeklenme Sonucu.....	26

4. TARTIŞMA	33
5. SONUÇ ve ÖNERİLER.....	35
KAYNAKLAR	36
EKLER.....	39



ŞEKİL LİSTESİ

Sayfa

Şekil 2.1. Su altı ortamı (Fisheries, 3 Mayıs 2025)	5
Şekil 2.2. FIR filtre blok diyagramı (Kumar, 2013)	6
Şekil 2.3. IIR filtre blok diyagramı (Medeiros vd., 2004)	6
Şekil 2.4. Adaptif filtre blok diyagramı (Mustafa vd., 2009)	7
Şekil 2.5. Temel FPGA yapısı (Sharma, 2001)	10
Şekil 2.6. Sabit noktalı sayı formatı	10
Şekil 2.7. Tek duyarlıklı kayan sayı formatı	11
Şekil 2.8. Çift duyarlıklı kayan sayı formatı	11
Şekil 2.9. UART haberleşme protokolü bağlantı diyagramı	14
Şekil 2.10. UART haberleşme protokolü veri iletim yapısı	14
Şekil 2.11. Akustik veri kaydı deney düzeneği	15
Şekil 2.12. Yansıyan sinyal kayıt akışı	16
Şekil 2.13. Ping sinyali oluşturulması çalışma akışı	16
Şekil 2.14. Yansıyan ping sinyali	17
Şekil 2.15. Ping sinyali	19
Şekil 2.16. Uygulama süreci blok diyagramı	20
Şekil 3.1. $\mu = 0,1$ değerinde en yüksek SNR değeri değişimi	23
Şekil 3.2. $\mu = 0,01$ değerinde en yüksek SNR değeri değişimi	24
Şekil 3.3. $\mu = 0,001$ değerinde en yüksek SNR değeri değişimi	24
Şekil 3.4. $\mu = 0,0001$ değerinde en yüksek SNR değeri değişimi	25
Şekil 3.5. Gürültülü sinyal ve FPGA üzerinde filtrelenmiş sinyal	25
Şekil 3.6. $\lambda = 0,99$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi ...	27
Şekil 3.7. $\lambda = 0,99$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi	27
Şekil 3.8. $\lambda = 0,9$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi	28
Şekil 3.9. $\lambda = 0,9$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi	28
Şekil 3.10. $\lambda = 0,7$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi ...	29
Şekil 3.11. $\lambda = 0,7$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi	29
Şekil 3.12. $\lambda = 0,5$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi ...	30
Şekil 3.13. $\lambda = 0,5$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi	30
Şekil 3.14. Gürültülü sinyal ve FPGA üzerinde filtrelenmiş sinyal	31

TABLO LİSTESİ

Sayfa

Tablo 2.1. NEXYS 4 DDR geliştirme kartı giriş çıkış birimleri	13
Tablo 2.2. LMS algoritması başlangıç parametreleri	21
Tablo 2.3. RLS algoritması başlangıç parametreleri	21
Tablo 3.1. LMS algoritması FPGA donanım kaynak kullanımları	26
Tablo 3.2. RLS algoritması FPGA donanım kaynak kullanımları	31
Tablo 3.3. LMS ve RLS algoritmaları gerçekleştirme sonuçları.....	32



FOTOĞRAF LİSTESİ

Sayfa

Fotoğraf 2.1. NEXYS 4 DDR geliştirme kartı.....	13
Fotoğraf 2.2. Deney düzeneği	17
Fotoğraf 2.3. Akustik kayıt için kullanılan transdüser.....	18



KISALTMA VE SEMBOLLER LİSTESİ

AMD	: Advanced Micro Devices (İleri Mikro Aygıtlar)
A.Ş.	: Anonim Şirketi
BUFG	: Global Buffer Memory (Küresel Ara Bellek)
CLB	: Configurable Logic Block (Yapılandırılabilir Mantık Bloğu)
CW	: Continuous Wave (Sürekli Dalga)
dB	: Desibel
DDR	: Double Data Rate (Çift Veri Hızı)
DSP	: Digital Signal Processing (Sayısal Sinyal İşleme)
FF	: Flip-Flop
FIR	: Finite Impulse Response (Sonlu Dürtü Tepkisi)
FPGA	: Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizisi)
HDL	: Hardware Description Language (Donanım Tanımlama Dili)
Hz	: Hertz
IEEE	: The Institute of Electrical and Electronics Engineers (Elektrik Elektronik Mühendisleri Enstitüsü)
IIR	: Infinite Impulse Response (Sonsuz Dürtü Tepkisi)
IO	: In-Out (Giriş-Çıkış)
IOB	: In-Out Block (Giriş-Çıkış Bloğu)
JTAG	: (Ortak Test Eylem Grubu)
Kbps	: Kilo Bits Per Second (Kilobit Başına Saniye)
KHz	: Kilo Hertz
LMS	: Least Means Square (En Küçük Ortalama Kare)
LUT	: Lookup Table (Arama Tablosu)
MHz	: Mega Hertz
ms	: milisaniye
NLMS	: Normalized Least Means Square (Normalize En Küçük Ortalama Kare)
PMOD	: Peripheral Module (Çevresel Modül)
RAM	: Random Access Memory (Rastgele Erişimli Bellek)
RLS	: Recursive Least Squares (Özyinelemeli En Küçük Kareler)
RVS	: Receiving Voltage Sensivity (Alım Gerilim Hassasiyeti)
SM	: Switching Matrix (Anahtarlama Matrisi)
SNR	: Signal Noise Rate (Sinyal Gürültü Oranı)
TVR	: Transmitted Voltage Response (İletim Gerilim Yanıtı)
UART	: Universal Asynchronous Receiver- Transmitter (Evrensel Asenkron Alıcı-Verici)
USB	: Universal Serial Bus (Evrensel Seri Veriyolu)

VGA : Video Graphics Array (Video Grafik Dizisi)

VHDL : Very High Speed Integrated Circuit Hardware Description Language
(Yüksek Hızlı Donanım Tanımlama Dili)



TEŞEKKÜR

Lisans ve yüksek lisans eğitim dönemim boyunca desteğini esirgemeyen değerli danışman hocam Doç. Dr. Oktay Aytar'a teşekkür ederim.

Hem mesleki kariyerimde hem yüksek lisans eğitim dönemimde güvenini sonuna kadar hissettiren ikinci danışmanım Dr. Mehmet Ali Çavuşlu'ya teşekkür ederim.

Tez çalışması için sağladığı laboratuvar ortamından dolayı Koç Bilgi ve Savunma Teknolojileri A.Ş.'ye teşekkür ederim.

Her zaman yanımda olup desteğini hiçbir zaman esirgemeyen annem Nursan Karakaya'ya, babam Ertuğrul Karakaya'ya ve canım eşim İrem Uluçay Karakaya'ya çok teşekkür ederim.

1. GİRİŞ

Son yıllarda yaşanan teknolojik gelişmeler ve iletişim yöntemlerinin evrimi, su altı haberleşme sistemlerinde kullanılan farklı teknolojilerin gelişmesini sağlamıştır. Haberleşme sistemlerinin gürültüden arındırılarak istenen sinyallerin elde edilmesi, sistem performansını artırmak ve doğru bilgi iletimini sağlamak açısından önem taşımaktadır. Ancak su altında ortam gürültüsünün ve ortamdaki haberleşme kanalının zamansal ve mekânsal olarak farklılık göstermesi nedeniyle karasal haberleşme sistemlerinin aksine su altında veri aktarımı çözülmesi gereken zor bir problem haline gelmektedir (Çavuşlu, 2020).

Alıcı birimde, algılanan sinyallerden bilgi ve su altı ortam gürültüsünün ayrıştırılması amacıyla dijital ortamda farklı filtreleme yöntemleri kullanılmaktadır. Sonlu Dürtü Yanıtı (FIR) ve Sonsuz Dürtü Yanıtı (IIR) filtrelere alternatif olarak kullanılabilecek adaptif filtreler, sinyal işleme uygulamalarının birçok alanında yaygın olarak kullanılmaktadır (Haykin, 1996).

Adaptif filtreler, uyarlamalı yapılarından dolayı klasik filtrelere göre daha esnek bir kullanım ve çevresel değişimlere daha hızlı bir tepki sağlarlar. Adaptif filtreleme algoritmalarında ise En Küçük Ortalama Kare (LMS) algoritması ve Özyinelemeli En Küçük Kareler (RLS) algoritması öne çıkmaktadır (Ganorkar vd., 2022).

Sinyal işleme uygulamalarında en çok kullanılan donanımlardan olan Alanda Programlanabilir Kapı Dizileri (FPGA) diğer donanımsal yöntemlere göre daha az donanım maliyetine sahiptir ve hızlı bir şekilde programlanabilmektedir (Safarian vd., 2015). Bu özellikleriyle sayısal filtreleme uygulamaları içinde FPGA donanımının kullanılması kaçınılmaz olmaktadır.

Adaptif filtrelerin FPGA üzerinde uygulandığı çalışmalara bakıldığında, Ganorkar vd. (2022) FIR tabanlı LMS algoritmasının FPGA ortamında gerçekleştirme işlemini göstermiştir.

Liu vd. (2023) boru hattı mimari yapısı kullanılarak modifiye edilmiş bir LMS algoritmasının FPGA üzerinde gösterdiği performansı incelemiş ve orijinal LMS algoritması ile performans karşılaştırmasını yapmıştır. Karşılaştırma sonucuna göre LMS algoritmasında işlenebilecek sinyalin maksimum frekansı artmış, buna karşın kullanılan donanımsal kaynaklarda artış olmuştur.

Mohyaldeen (2015) yapmış olduđu çalışmada RLS, LMS, Normalize En Küçük Ortalama Kare (NLMS) algoritmalarının gürültü yok etme, sinyal tahmini, sistem tanımlaması özelliklerini test etmiş ve ortalama hata değerine göre RLS algoritmasının en iyi sonucu verdiğini ifade etmiştir.

Shiva vd. (2015) kullanıcı gereksinimlerine göre FPGA üzerinde kendini LMS, NLMS ve RLS algoritması olarak yapılandırabilen bir adaptif filtre yapısı kurmuştur.

Vijayaraghavan vd. (2014) FIR tabanlı LMS algoritması kullanarak konuşma iyileştirmede gürültü filtreleme işlemini göstermiştir. Buna göre kaydedilen ses dosyasına eklenen rastgele gürültü sinyalleri başarıyla filtrelenmiştir.

Pitchaiah vd. (2014) LMS algoritmasının FPGA üzerinde daha düşük alan ve daha düşük bir gecikmeyle uygulama yöntemini anlatmıştır. Bu yöntemle göre, boru hattı mimarisiyle dağıtılmış bir aritmetik yapı içerisinde bit kaydırma ve arama tablosu kullanılarak LMS algoritması daha düşük alan kullanımı ve daha düşük bir gecikmeyle gerçekleştirilmiştir.

Tudosa ve Adochiei (2012) gerçek zamanlı bir Elektrokardiyografi (EKG) sinyali üzerinde güç hattı gürültüsünün filtrelenmesi amacıyla LMS algoritmasının FPGA donanımında gerçekleştirme performansını incelemiş ve bu uygulamanın başarılı bir şekilde gerçekleştirildiğini göstermiştir.

Elhossini vd. (2006) bir ses işleme uygulamasında LMS algoritmasının FPGA üzerinde uygulama işlemini göstermiştir. Buna göre 16 bit sabit noktalı sayı aritmetiği kullanılarak tasarlanan yapı, donanımsal gerçekleştirilerek diğer yazılımsal yöntemlere göre en hızlı yöntemi sunmuştur.

Cai vd. (2013) LMS algoritmasının FPGA tabanlı gerçekleştirme işlemini göstermiştir. Buna göre FPGA üzerinde yapılan bu uygulama diğer uygulama ortamlarına göre daha hızlı, daha az maliyetli ve daha fazla esneklik sunmaktadır.

Liang vd. (2013) RLS algoritmasının FPGA tabanlı gerçekleştirme işlemini göstermiştir.

Bellizia vd. (2017) geliştirilmiş bir RLS algoritmasının sabit nokta aritmetiği kullanarak FPGA tabanlı gerçekleştirme işlemini göstermiştir. Buna göre matris hesaplamaları yerine daha basit matematiksel hesaplamalar kullanan bu yöntem geleneksel RLS algoritmasına göre daha düşük hesaplama maliyetine ve sayısal özelliklere sahiptir.

Stanciu vd. (2015) geliştirilmiş bir RLS algoritmasını FPGA üzerinde uygulayarak akustik yankı sinyali üzerinde gürültü filtreleme işlemini gerçekleştirmiştir. Buna göre telekonferans sistemlerinde yankıyı engellemek için daha az kaynak tüketen ve daha kararlı bir RLS algoritması sunulup FPGA üzerinde etkili bir şekilde çalıştığı gösterilmiştir.

S vd. (2023) çarpansız LMS algoritmasının düşük güç tüketimi ve alan verimliliğiyle FPGA üzerinde uygulanmasını ele almıştır. Çarpansız halinin orijinal haline göre daha verimli olduğunu göstermiştir.

N vd. (2024) HDL ile geliştirilen ve donanım açısından optimize edilmiş Gecikmeli LMS algoritmali bir uyarlanabilir filtrenin FPGA üzerindeki tasarım ve uygulamasını gerçekleştirmiştir. Yapılan testlerde geleneksel LMS algoritmasına göre daha iyi yakınsama hızı, düşük güç tüketimi ve daha az kaynak kullanımı olduğunu göstermiştir.

Yapılan bu çalışmalar incelendiğinde adaptif filtrelerin kullanılabilirliği ve FPGA donanımı üzerindeki uygulanabilirliği çeşitli uygulamalar üzerinde gösterilmektedir. Bu sebeple LMS ve RLS adaptif filtrelerinin akustik sinyaller üzerindeki filtreleme etkilerinin gösterilmesi ve FPGA donanımı üzerinde gerçekleştirilebilir olduğunun gösterilerek bu alandaki çalışmalara fayda sağlaması amaçlanmıştır.

Bu çalışma LMS ve RLS adaptif filtrelerinin teorik gösterimlerinden, FPGA üzerinde gerçekleştirilmesinden ve gerçek dünyada kaydedilmiş akustik sinyaller üzerinde gerçekleştirilmesi problemlerine odaklanmıştır.

Yapılan bu çalışmanın organizasyonu ise şu şekildedir. Materyal ve metod bölümünde, su altı akustiği ile ilgili teorik bilgiler verilmiş, adaptif filtre teorisi ve uygulama esnasında kullanılan LMS ve RLS filtreler anlatılmış, FPGA donanımı ile ilgili bilgi verilmiş, uygulamada kullanılan geliştirme kartı ve haberleşme protokolü tanıtılmış, çalışmada kullanılan uygulama deney düzeneği açıklanmıştır. Bulgular bölümünde, yapılan çalışma ve çalışma çıktıları açıklanmıştır. Tartışma bölümünde, yapılan çalışmada kullanılan LMS ve RLS filtrelerin filtreleme performansları karşılaştırılmış ve çalışmanın literatürde kullanılabilirliği gösterilmiştir. Sonuç bölümünde, bu çalışmanın genel bulguları ve gelecekteki çalışmalar için öneriler sunulmuştur.

2. MATERYAL ve YÖNTEM

Bu bölümde, su altı akustiği ile adaptif filtre teorileri hakkında bilgi verilmekte, uygulama için kullanılan teorik yaklaşım ve uygulama yöntemi açıklanmaktadır.

2.1 Su Altı Akustiği

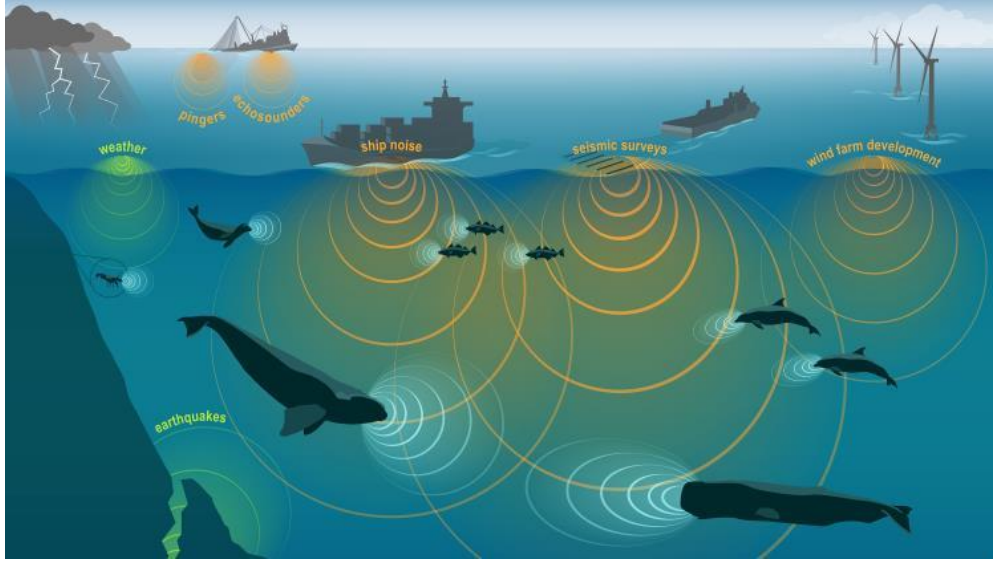
Akustik bilimi, sesin katı ve sıvı ortamlarda iletimini inceleyen, bu alandaki pratik uygulamaları araştıran bir disiplindir (Coates, 1990). Ses, iletken bir ortamda uzunlamasına dalga hareketi olarak yayılır ve özellikle su altında uzun mesafelerde enerji transferinin en verimli yolu olarak karşımıza çıkar (Coates, 1990). Bu nedenle, su altı akustiği hem askeri hem de ticari açıdan büyük bir öneme sahiptir.

Su altı akustiği uygulamaları çok eski tarihlere uzansa da, bir bilim dalı olarak gelişimi II. Dünya Savaşı'na dayanmaktadır (Urlick, 1975). Bu dönemde, su altı akustiği alanındaki literatür genişlemiş; ardından pratik uygulamaların artması ve insanların denizleri keşfetme çabalarıyla bu alandaki çalışmalar hızla ilerlemiştir (Urlick, 1975).

Su altı akustiği, su altı özelliklerini görüntülemek, su dalga kılavuzu aracılığıyla bilgi iletmek veya okyanus özelliklerini ölçmek için akustik yöntemlerin kullanılmasını içerir. Bu uygulamalar arasında ise su altı iletişimi önemli bir yer tutmaktadır. Su altı iletişiminde ortam gürültüsünün değişken olması, haberleşme ağlarının tasarımını doğrudan etkilemektedir (Çavuşlu, 2020). Bu açıdan su altı ortamında oluşturulacak haberleşme ağlarının tasarımında düşünülmesi gereken bir problemdir.

2.2 Su Altı Ortam Gürültüsü

Su altı ortam gürültüsünün yapısı incelendiğinde türbülans gürültüsü, gemi trafiği gürültüsü, yüzey gürültüsü ve termal gürültü bileşenlerinden oluştuğu görülmektedir (Çavuşlu, 2020). Bu gürültü türlerinin etkilerini analiz etmek için çalışmalar olsa da, değişken ortam koşulları ve bunların birbirleriyle etkileşimi gibi faktörler çalışma alanını zorlaştırmaktadır. Şekil 2.1'de su ortamının değişkenliğine ve gürültüye sebep olan faktörleri gösterilmektedir.



Şekil 2.1. Su altı ortamı (Fisheries, 3 Mayıs 2025)

Türbülans gürültüsü, düşük frekanslarda (1 Hz –10 Hz) etkisi yüksek olan gürültüdür (Coates, 1990).

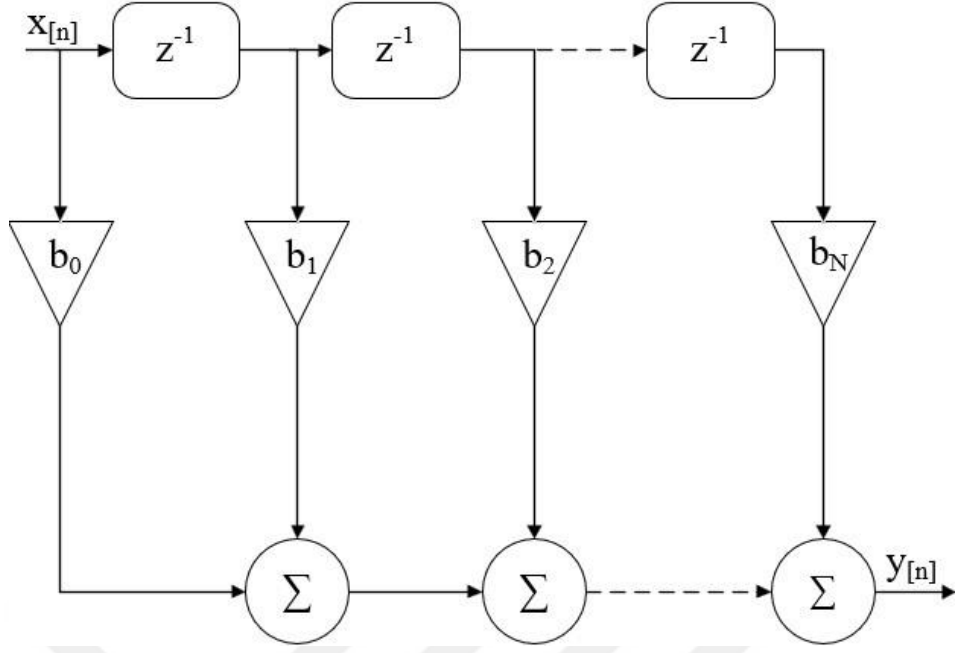
Gemi trafiği gürültüsü, su taşıtları tarafından su altı ortamına yayılan akustik gürültü bileşenleri, makine hareketleri, dişliler, akış, jeneratör ve fan motorlarından çıkan gürültülerden oluşmakta ve pervanenin yüksek hızlarda çıkardığı kavitasyon gürültüsü de eklenmektedir. Genelde 10 Hz ile 2 KHz arasında gözlemlenmektedir (Çavuşlu, 2020).

Yüzey gürültüsü, rüzgâr hızına göre değişen ve farklı seviyelerde su yüzeyinde oluşan gürültüdür (Traverso vd., 2012).

Termal gürültü, temel olarak su moleküllerinin rastgele hareket etmesi sonucu meydana gelen ve 50 KHz'den daha yüksek frekanslarda baskın olarak gözlemlenen gürültü türüdür (Coates, 1990).

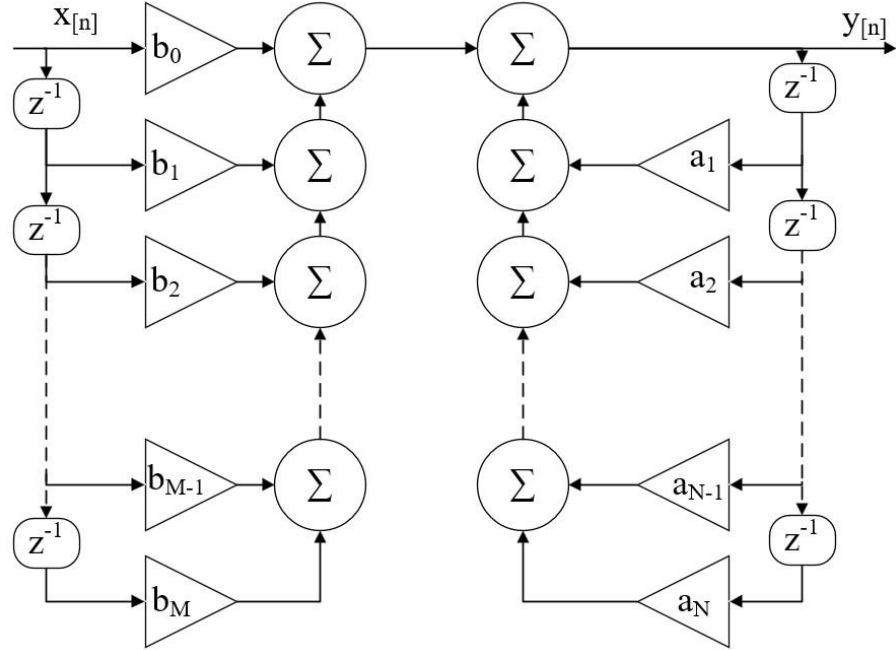
2.3 Adaptif Filtre Teorisi

Klasik sayısal filtreler IIR ve FIR filtreler olmak üzere iki çeşittir (Roberts, 2017). Aralarındaki temel fark, IIR filtrenin geri beslemeli bir yapıda olmasıdır. Bu durum FIR filtreler göre IIR filtreleri özyinelemeli hale getirir. Sonuç olarak ise aynı filtre derecesinde IIR filtreler daha keskin sonuçlar elde ederler ve FIR filtreler göre daha düşük filtre derecesinde işlem yapabilirler, bu da IIR filtreleri daha hızlı hale getirir. Buna rağmen zaman gecikmesi ve kararlılık açısından FIR filtreler daha stabil çalışmaktadır. Şekil 2.2'de FIR filtre blok diyagramı gösterilmiştir.



Şekil 2.2. FIR filtre blok diyagramı (Kumar, 2013)

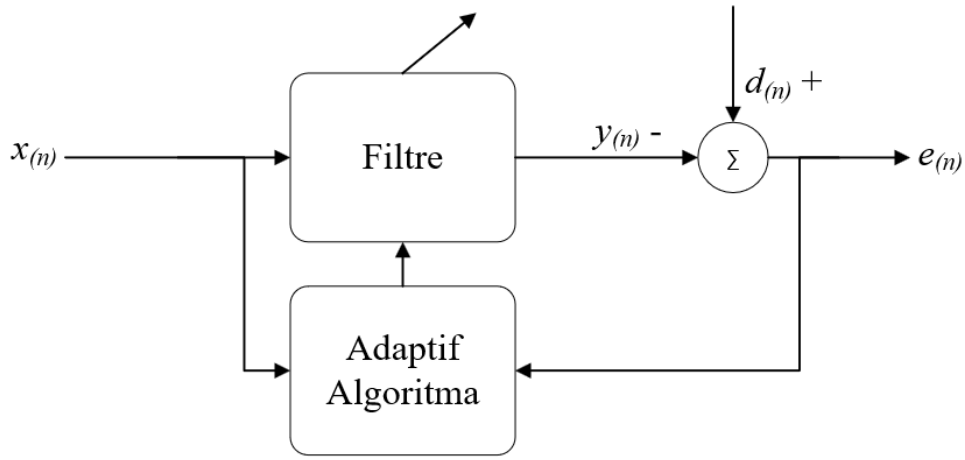
Şekil 2.3’de IIR filtre blok diyagramı gösterilmiştir.



Şekil 2.3. IIR filtre blok diyagramı (Medeiros vd., 2004)

Klasik sayısal filtreler sabit bir yapıya sahipken, adaptif filtreler çevre koşullarına ve giriş sinyalinin özelliklerine göre dinamik olarak değişir. Bu da adaptif filtrelerin zamanla değişen ortamlar veya karmaşık sinyallerle çalışırken

daha avantajlı olmasını sağlar. Adaptif filtre algoritmalarının genel blok diyagramı Şekil 2.4'te gösterilmiştir.



Şekil 2.4. Adaptif filtre blok diyagramı (Mustafa vd., 2009)

$x(n)$, algoritmaya giren gürültülü sinyali; $y(n)$, filtreleme işleminden sonra elde edilen çıkış sinyalini; $e(n)$ ise $y(n)$ çıkış sinyali ile $d(n)$ referans sinyali arasındaki farkı, yani hatayı ifade etmektedir.

Adaptif filtre algoritmaları, LMS ve RLS olmak üzere iki temel başlık altında toplanır (Lamp1, 2020). Kullanılacak algoritma sistemin donanım kaynak tüketimi, gürültülü sinyal karakteristiği, gürültü dayanımı, uygulama alanı ve ortamı gibi parametrelere bağlı olarak farklılık gösterir. Algoritmanın filtre kısmında ise FIR veya IIR filtre kullanılabilir. Bu çalışma içerisinde hesaplama kolaylığı, adaptif süreçlere uyum ve kararlılık için FIR filtre tercih edilmiştir.

2.3.1 LMS Algoritması

LMS algoritması, hata en aza indirilene kadar anlık olarak belirlenen hata değerine göre filtre katsayılarını uyarlar (Esposito vd., 2018). Bu uyarlamayı gerçekleştirmek için gradyan tabanlı bir yaklaşım kullanır ve bu yaklaşımda başlangıç filtre katsayılarının çok küçük ya da sıfır olduğu varsayılır. Gradyan pozitifse filtre ağırlıkları azaltılır, böylece hata pozitif olarak artmaz. Gradyan negatifse, filtre ağırlıkları artırılır. Adım boyutu çok küçükse, algoritma yavaş yakınsar; eğer çok büyükse, hızlı yakınsar ve sistem minimum hata değerine ulaşırken kararlılık sorunlarıyla karşılaşabilir (Ganorkar vd., 2022).

Yapılan bu çalışmada, adım boyutunun farklı değerlerine göre karşılaştırma işlemi gerçekleştirilmiştir.

LMS algoritmasında gradyan tabanlı yaklaşımla ifade edilen matematiksel eşitlik (2.1)'de gösterilmiştir (Haykin, 1996).

$$w_{(n+1)} = w_{(n)} + \frac{1}{2} \mu [-\nabla J_{(n)}] \quad (2.1)$$

Bu denklemde $J_{(n)}$ değişkeni ortalama hatanın karesini ifade etmektedir ve bu eşitlik (2.2)'de gösterilmiştir.

$$J_{(n)} = e^2_{(n)} \quad (2.2)$$

Bu ifadenin gradyanı alındığında (2.3) eşitliği elde edilir.

$$\nabla J_{(n)} = -2e_{(n)}w_{(n)} \quad (2.3)$$

(2.3) eşitliği, (2.1)'de yerine koyulduğunda (2.4) eşitliği elde edilir ve bu denklem LMS algoritmasının katsayı güncelleme denklemidir. Burada μ değişkeni adım boyutunu ifade etmektedir.

$$w_{(n+1)} = w_{(n)} + \mu x_{(n)}e_{(n)} \quad (2.4)$$

Giriş sinyali ile burada hesaplanan filtre katsayıları kullanarak filtreleme işlemi yapılırsa çıkış sinyali (2.5)'de gösterildiği gibi elde edilir.

$$y_{(n)} = w^T_{(n)}x_{(n)} \quad (2.5)$$

2.3.2 RLS Algoritması

LMS algoritmasında, adaptasyon parametresi adım boyutu sabit alındığından dolayı çalışma bölgesi sınırlanmaktadır (Meral, 2008). Bu sınırlamayı azaltmak için gerçekleştirilen çalışmalar sonucunda RLS algoritması geliştirilmiştir. RLS algoritması, referans sinyal ile filtre çıkışı arasındaki farkın kareleri toplamının en aza indirilmesini hedefler ve daha geniş bir çalışma bölgesine sahiptir. Ayrıca LMS algoritmasına göre daha hızlı yakınsama özelliğine sahiptir.

(2.6)'da RLS algoritmasının filtre katsayılarını güncellemesi için gerekli olan kalman kazancı eşitliği gösterilmiştir (Haykin, 1996). λ değişkeni unutma faktörünü, P değişkeni ise kovaryans matrisini ifade etmektedir.

$$k_{(n)} = \frac{\lambda^{-1}P_{(n-1)}u_{(n)}}{1 + \lambda^{-1}u^H_{(n)}P_{(n-1)}u_{(n)}} \quad (2.6)$$

Kalman kazancı fonksiyonu kullanılarak RLS filtrenin elde edilen katsayı güncelleme eşitliği (2.7)'de gösterilmiştir (Haykin, 1996).

$$w_{(n+1)} = w_{(n)} + k_{(n)}e_{(n)} \quad (2.7)$$

Kovaryans matrisinin eşitlik ifadesi (2.8)'de gösterilmiştir (Haykin, 1996).

$$P_{(n)} = \lambda^{-1}(P_{(n-1)} - k_{(n)}u_{(n)}^H P_{(n-1)}) \quad (2.8)$$

Unutma faktörü algoritmanın yakınsamasına ve kararlılığına etki eder. Bu da adaptif filtrenin performansını önemli ölçüde etkiler (Wang, 2009). $0 < \lambda \leq 1$ aralığında tanımlı olan unutma faktörünün 1'e yakın olması algoritmanın geçmiş değerlere daha fazla bağlı olmasını, 0'a yakın olması ise daha az bağlı olmasını ifade eder (Maraş vd., 2018). Uygulama içerisindeki sinyaller durağan ise unutma faktörü 1'e yakın, durağan değilse en yüksek başarıyı elde edecek şekilde deneysel olarak belirlenir (Meral, 2008).

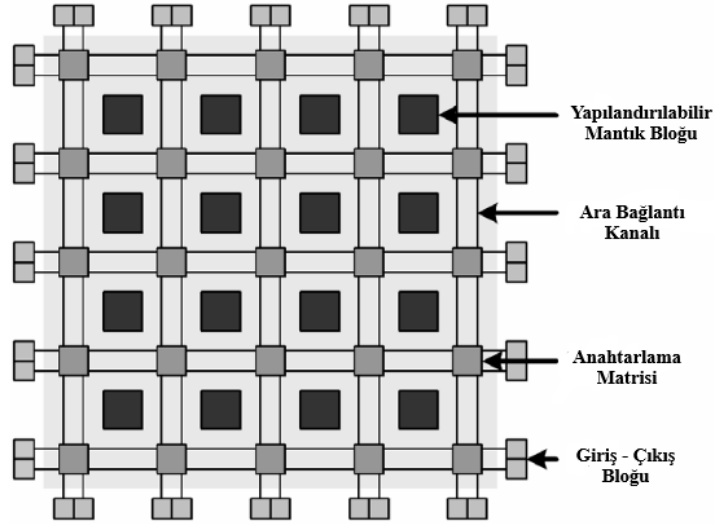
Yapılan bu çalışma içerisinde, unutma faktörünün farklı değerlerinde en yüksek başarıya göre karşılaştırma işlemi gerçekleştirilmiştir.

Kovaryans matrisi ise algoritmanın parametre güncellemeleri yapabilmesi için gerekli olan bilgiyi sağlayıp algoritma dinamiklerini izlemek ve hata minimizasyonu sağlamak amacıyla kullanılır.

2.4 FPGA

FPGA, yeniden yapılandırılabilir kapılardan oluşan bir dizi donanımdır. Kullanıcı yazılım programlama araçlarıyla FPGA üzerinde bir Donanım Tanımlama Dili (HDL) veya şema kullanarak tasarımları uygulayabilir.

FPGA'ler, Yapılandırılabilir Mantık Bloğu (CLB) dizisi olarak oluşturulur (Harris, 2022). Her CLB, kombinasyonel veya ardışık işlevleri gerçekleştirecek şekilde yapılandırılabilir. CLB'ler, harici cihazlarla arayüz oluşturmak için giriş/çıkış blokları (IOB) ile çevrilidir. IOB'ler, CLB girişlerini ve çıkışlarını çip paketindeki pinlere bağlar. CLB'ler programlanabilir yönlendirme kanalları aracılığıyla diğer CLB'lere ve IOB'lere bağlanabilir (Harris, 2022). Şekil 2.5'de temel FPGA yapısı gösterilmiştir.



Şekil 2.5. Temel FPGA yapısı (Sharma, 2001)

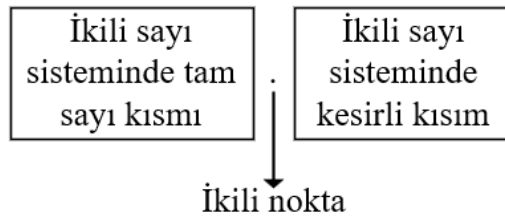
Bu tez içerisinde AMD firmasına ait XC7A100T-1CSG324C parça numaralı FPGA çipini içerisinde bulunduran, NEXYS 4 DDR geliştirme kartı kullanılmıştır.

2.5 Sayı Sistemleri

FPGA donanımları tam sayı ve kesirli sayılar üzerinde çalışabilmektedir. Tam sayıların ikili sayı sisteminde gösterimi sabit olsa da kesirli sayıların gösteriminde 2 farklı yöntem kullanılmaktadır (Harris, 2022). Kullanılan bu yöntemler FPGA üzerinde işlenecek veriye göre tercih edilebilmekte ve birbirlerine dönüşüm yapılabilmektedir.

2.5.1 Sabit Noktalı Sayı Sistemi

Sabit noktalı sayı sistemi, dinamik olmadığından dolayı verimli değildir ve donanım maliyetlerini önemli ölçüde düşürürken sayısal hassasiyette belirgin kayıplara yol açabilir. Şekil 2.6’da sabit noktalı sayı formatı gösterilmektedir.



Şekil 2.6. Sabit noktalı sayı formatı

2.5.2 Kayan Noktalı Sayı Sistemi

Bu sayı sisteminde sayıların gösterimi Elektrik ve Elektronik Mühendisleri Enstitüsü (IEEE) 754 standardı ile belirlenmiştir (Bharade vd., 2016).

Tek duyarlılıklı IEEE 754 kayan noktalı sayı standardında, işaret biti eğer sayı pozitifse 0, negatifse 1 alınır. Üst bitleri kısmında sayı normalize edildikten sonra 2'nin üzeri olan değer 127 ile toplanır ve bu işlemin sonucunun ikili sayı gösterimi bu kısma yazılır.

Kesir bitleri kısmına ise normalize edilmiş kısmın kesirli kısmı yazılır. Boş olan bitlerin yerine 0 yazılır. Şekil 2.7'de tek duyarlılıklı IEEE 754 kayan noktalı sayı formatı gösterilmektedir.



Şekil 2.7. Tek duyarlılıklı kayan sayı formatı

Çift duyarlılıklı IEEE 754 kayan noktalı sayı standardında, işaret biti eğer sayı pozitifse 0, negatifse 1 alınır. Üst bitleri kısmında sayı normalize edildikten sonra 2'nin üzeri olan değer 1023 ile toplanır ve bu işlemin sonucunun ikili sayı gösterimi bu kısma yazılır.

Kesir bitleri kısmına ise normalize edilmiş kısmın kesirli kısmı yazılır. Boş olan bitlerin yerine 0 yazılır. Şekil 2.8'de çift duyarlılıklı IEEE 754 kayan noktalı sayı formatı gösterilmektedir.



Şekil 2.8. Çift duyarlılıklı kayan sayı formatı

2.6 Donanım Tanımlama Dilleri

1990'larda, tasarımcılar daha soyut bir seviyede çalışmaya başladıklarında, sadece mantıksal işlevleri tanımlayıp bu işlevlere uygun optimize edilmiş kapıları üretmek için bilgisayar destekli tasarım araçlarını kullanmanın, verimliliklerini önemli ölçüde artırdığını fark ettiler. Bu süreç içerisinde, özellikler genellikle bir

donanım tanımlama dili kullanılarak ifade edilir ve tasarımı gerçekleştirilir (Harris, 2022).

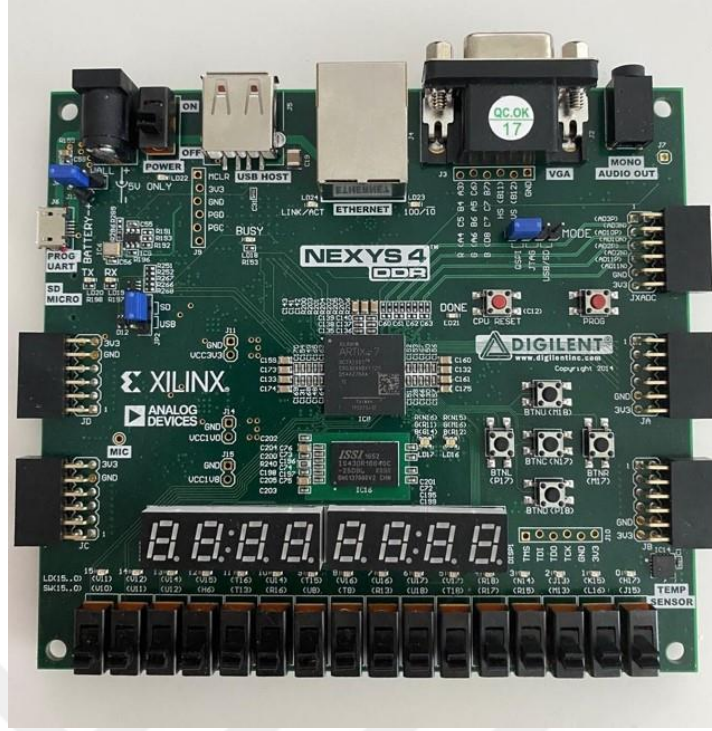
İki yaygın donanım tanımlama dili Verilog ve VHDL'dir. Verilog ve VHDL benzer ilkeler üzerine kuruludur ancak farklı sözdizimlerine sahiptir. VHDL, 1980'lerde Amerika Birleşik Devletleri Savunma Bakanlığı tarafından finanse edilen girişimin bir parçası olarak ortaya çıkmıştır (Pedroni, 2020). İlk versiyonu VHDL-87 olup daha sonra VHDL-93'e, sonra VHDL-2002'ye ve son olarak VHDL-2008'e yükseltilmiştir. Verilog, Phil Moorby ve Prabhu Goel tarafından 1984 civarında icat edilmiş ve Gateway Design Automation Inc.'e ait tescilli bir donanım modelleme dili olarak geliştirilmiştir (Digilent, 22 Ocak 2025). İlk ortaya çıktığında standartlaşmamış ve 1984 ile 1990 arasında kapsamlı bir şekilde değiştirilmiştir. 2009 itibari ile SystemVerilog ve Verilog dil standartları, günümüzde IC tasarımı ve doğrulaması için en popüler dillerden biri olan SystemVerilog 2009 ile birleştirilmiştir.

Günümüzde FPGA çipini üreten AMD, Efinix, Intel gibi çeşitli firmalar bulunmaktadır ve kullanıcıların belirli uygulamalar için kendi donanım yapılarını tasarlamalarına olanak tanırırlar. Hızlı prototiplemeye ve esnek kullanıma uygun olmasıyla telekomünikasyon, otomotiv, savunma, veri işleme gibi birçok endüstriyel alanda FPGA kullanımı giderek yaygınlaşmakta ve gelişmeye devam etmektedir.

2.7 NEXYS 4 DDR Geliştirme Kartı

Digilent firması tarafından üretilen NEXYS 4 DDR geliştirme kartı, hızlı prototipleme ve uygulamaların test edilebilmesi için üzerinde çeşitli çıkış portları ve haberleşme arayüzleri bulundurmaktadır. Bu sebeplerden dolayı bu çalışma içerisinde kullanımı tercih edilmiştir.

Geliştirme kartı üzerinde bulunan XC7A100T-1CSG324C FPGA çipinde temelde; 15850 mantık ünitesi, 4860 Kbits blok ram, 240 sayısal sinyal işleme (DSP) birimi bulunmaktadır. Fotoğraf 2.1'de NEXYS 4 DDR geliştirme kartı gösterilmiştir.



Fotoğraf 2.1. NEXYS 4 DDR geliştirme kartı

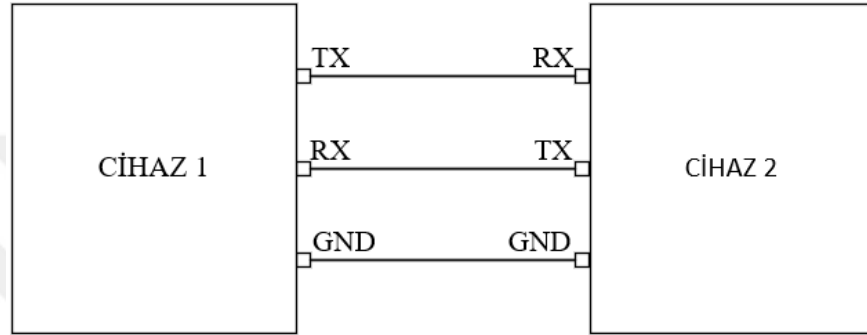
Tablo 2.1’de NEXYS 4 DDR geliştirme kartı üzerindeki giriş ve çıkış birimleri gösterilmiştir.

Tablo 2.1. NEXYS 4 DDR geliştirme kartı giriş çıkış birimleri

Donanım Adı	Adet	Donanım Adı	Adet
Anahtar	16	UART Bağlantı Birimi	1
Led	16	MicroSD Kart Yuvası	1
3 Renkli Led	2	Çevresel Modül Arayüzü (PMOD)	4
7 Segment Ekran	2	Analog Sayısal Dönüştürücü (ADC) Arayüzü	1
Video Grafik Dizini (VGA) Çıkışı	1	Sıcaklık Sensörü	1
Ethernet Portu	1	3 Eksen İvme Ölçer	1
Evrensel Seri Veri Yolu (USB) Arayüzü	1	Ses Çıkışı	1
Mikrofon Girişi	1	Rastgele Erişimli Bellek (RAM)	1
Flash	1	Ortak Test Eylem Grubu (JTAG) Bağlantısı	1

2.8 UART Haberleşme Protokolü

Bu çalışma içerisinde FPGA donanımı ile haberleşme sağlayabilmek için NEXYS 4 DDR geliştirme kartı üzerinde bulunan UART haberleşme protokolü, iki cihaz arasında seri olarak veri alışverişi gerçekleştirmek için kullanılmıştır. Bu haberleşme protokolünde veri iletimi iki yönde de gerçekleştirilebilir ve alıcı cihaz ile verici cihaz arasında 2 adet hat bağlantısı vardır. Ayrıca alıcı cihaz ile verici cihaz aynı toprak hattını kullanmalıdır (Rohde-Schwarz, 28 Ocak 2025). Şekil 2.9’da UART haberleşme protokolünün bağlantı diyagramı gösterilmiştir.



Şekil 2.9. UART haberleşme protokolü bağlantı diyagramı

UART haberleşme protokolü asenkron iletişim gerçekleştirir. Yani verici ve alıcı cihaz ortak bir saat sinyali paylaşmaz. Bu durum protokolü basitleştirse de verici ve alıcıya belirli gereklilikler yükler. Ortak bir saati paylaşmadıklarından dolayı aynı bit zamanlamasına sahip olabilmek için her iki cihaz da önceden ayarlanmış ve eş hızlarda iletim yapmalıdır. En çok kullanılan hızlar 4800 Kbps, 9600 Kbps, 19200 Kbps, 57600 Kbps ve 115200 Kbps’dır. Bunun dışında her iki cihaz da aynı veri iletim yapısı ve parametrelerini kullanmalıdır (Rohde-Schwarz, 28 Ocak 2025). Şekil 2.10’da UART haberleşme protokolü veri iletim yapısı gösterilmiştir.



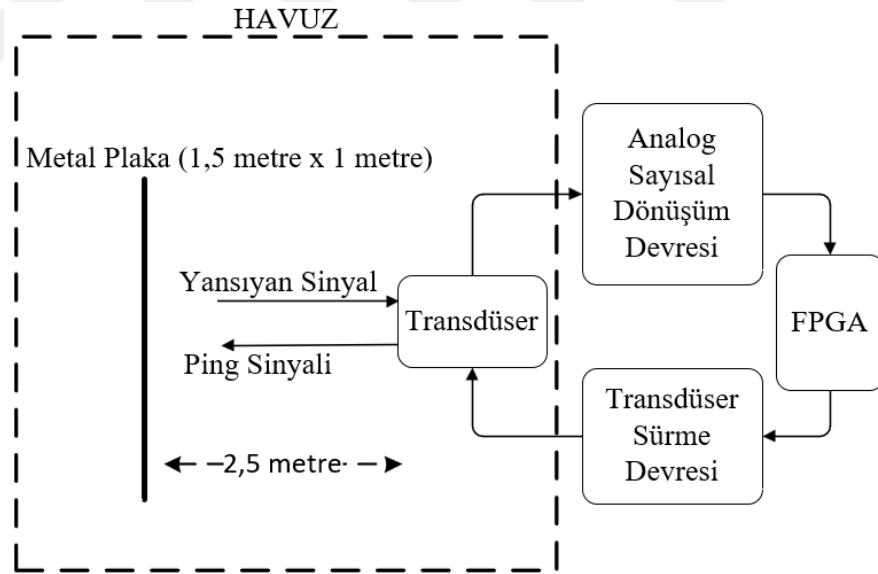
Şekil 2.10. UART haberleşme protokolü veri iletim yapısı

Başla biti, boşa yani yüksek durumdan düşük duruma geçirir ve hemen ardından kullanıcı veri bitleri gelir. Dur biti, veri bitleri bittikten sonra gönderilen verilerin sonunu belirtir ve boşa durumuna geçiş bitidir. Veri bitleri, kullanıcı verileridir. Başlangıç bitinden hemen sonra gelir. 5 ile 9 arasında kullanıcı veri biti

olabilir. Ancak 7 veya 8 bit yaygın olarak kullanılır. Ayrıca veri bitleri en az önemli bit en önce gönderilecek şekilde yapılandırılır. Parite biti, UART haberleşme protokolünde opsiyonel olarak hata algılama bitleridir. Veri bitlerinin sonu ile dur biti arasına yerleştirilir. Parite biti 2 şekilde kullanılabilir. Tek parite, verinin içindeki 1'lerin sayısı tek olacak şekilde bir parite biti eklenir. Çift Parite, verinin içindeki 1'lerin sayısı çift olacak şekilde bir parite biti eklenir.

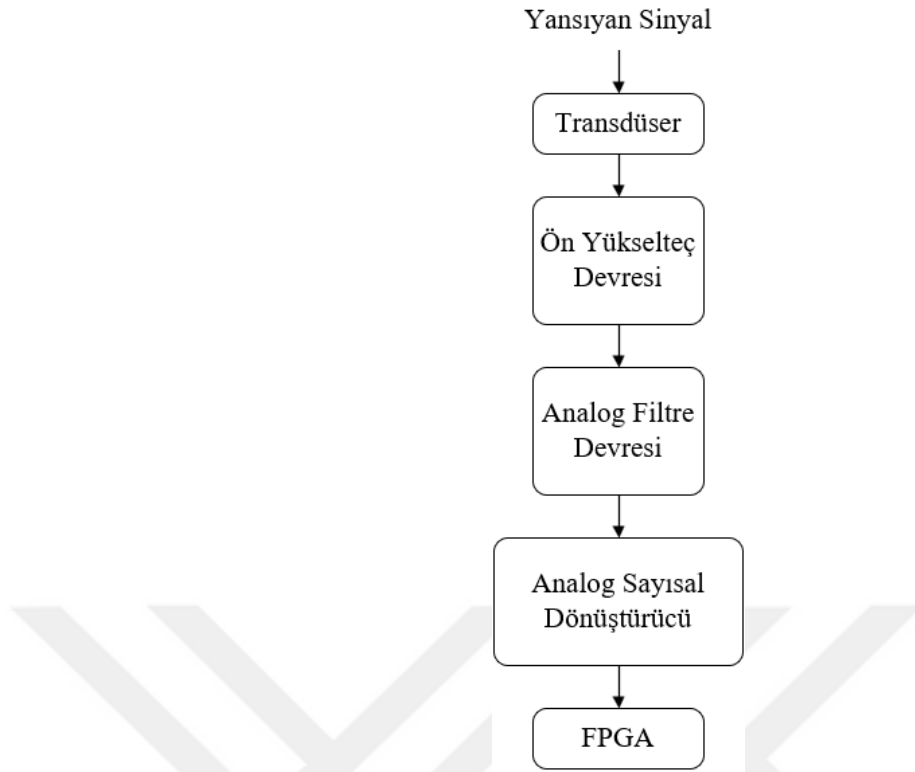
2.9 Akustik Veri Kaydı Deney Düzeneği

Bu çalışmada sunulan uygulama için öncelikle su altı ortamında akustik veri kaydı alınmıştır. Akustik veri kaydı, çalışma içerisinde LMS ve RLS adaptif filtre algoritmalarının uygulanması için gereklidir. Bu açıdan gerçek bir ortamda alınan veriler üzerinde çalışma gerçekleştirilmiştir. Bu kaydın alınmasına ilişkin kurulan deney düzeneğine ait blok diyagram Şekil 2.11'de gösterilmektedir. Burada havuzun içerisine yerleştirilen metal bir plakaya transdüser üzerinden ping sinyali gönderilmekte ve metal plakadan yansıyan sinyal transdüser üzerinden alınıp kaydedilmektedir.



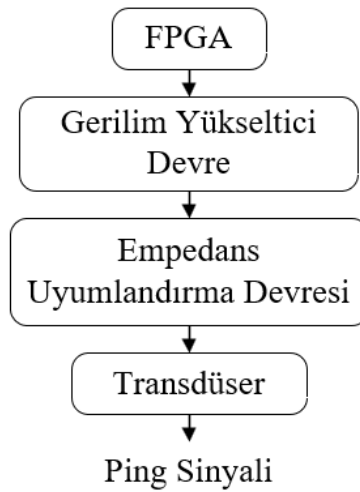
Şekil 2.11. Akustik veri kaydı deney düzeneği

Analog sayısal dönüşüm devresi, yansıyan ping sinyalinin analog sinyalden sayısal sinyale dönüştürülmesi işlemini gerçekleştirir. Transdüser gelen analog sinyaller ön yükselteç devresinde yükseltilir ve analog filtre devresine girer. Analog sayısal dönüştürücüden geçen filtrelenmiş sinyaller FPGA üzerinden kayıt edilir. Şekil 2.12'de yansıyan sinyal kayıt akışı gösterilmektedir.



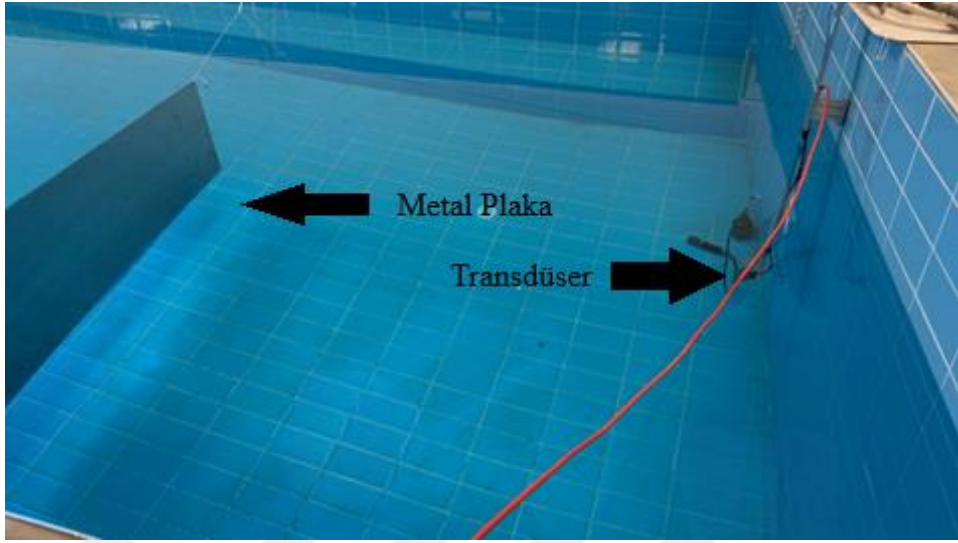
Şekil 2.12. Yansıyan sinyal kayıt akışı

Transdüser sürme devresinde, FPGA üzerinden kontrol edilen gerilim yükseltici devreden çıkan sürme voltajı, empedans uyumlandırma devresinden çıktıktan ve transdüse ulaştıktan sonra bulunduğu su ortamına akustik yayın yapmaya başlamaktadır. Şekil 2.13’de ping sinyali oluşturulmasının çalışma akışı gösterilmektedir.



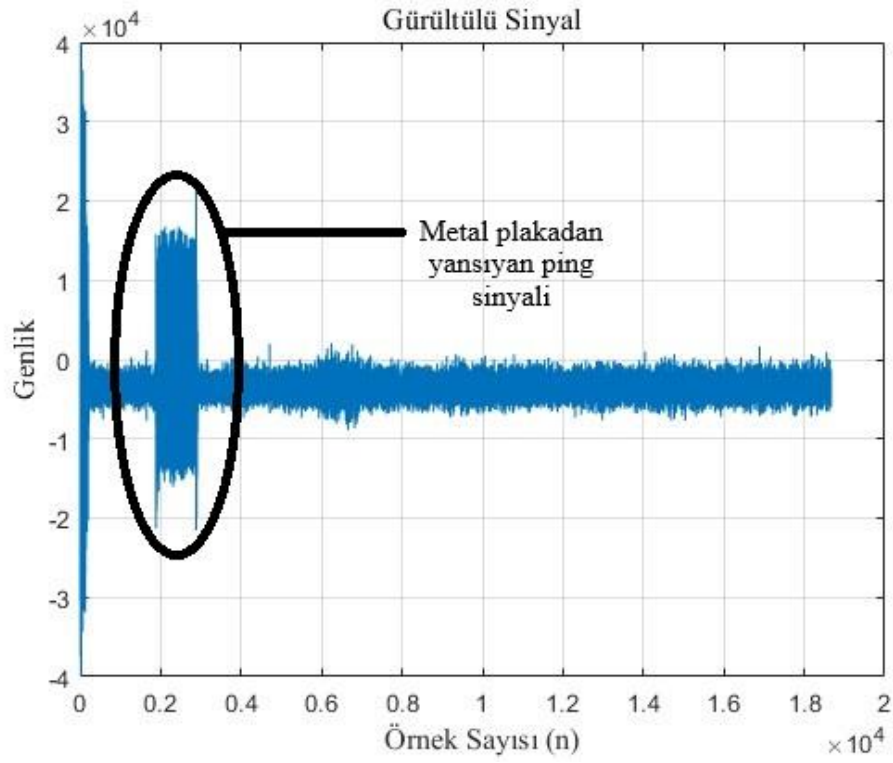
Şekil 2.13. Ping sinyali oluşturulması çalışma akışı

Havuz ortamında kurulan uygulama deney düzeneđi Fotoğraf 2.2’de gösterilmektedir.



Fotoğraf 2.2. Deney düzeneđi

Deney düzeneđi üzerinden alınan akustik veri kaydında, yansıyan ping sinyali Şekil 2.14’de gösterilmiştir.

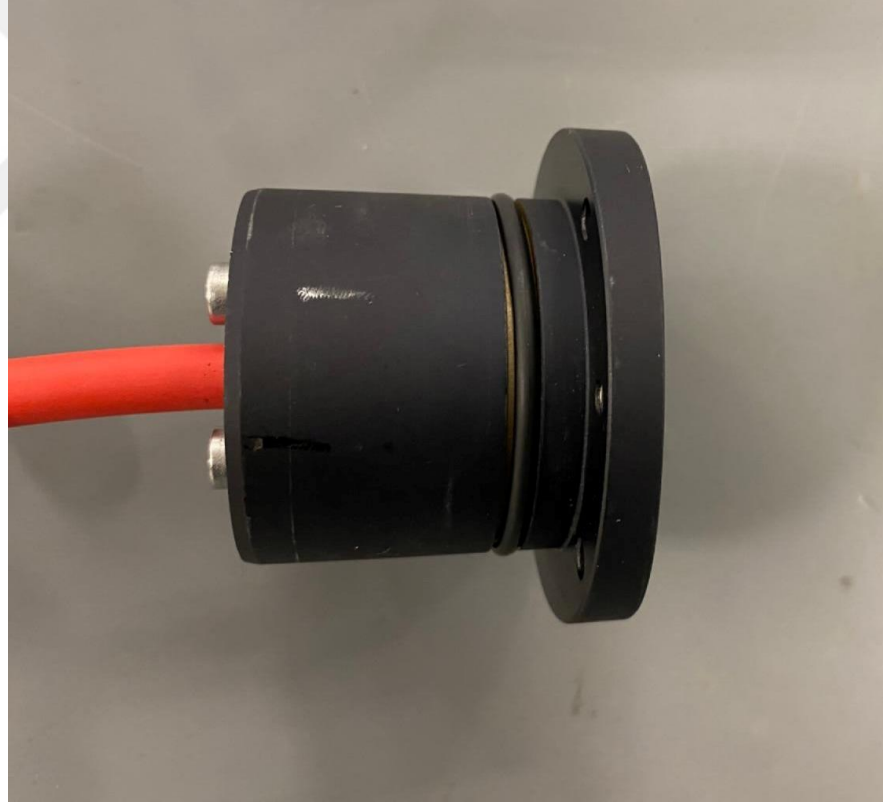


Şekil 2.14. Yansıyan ping sinyali

2.9.1 Transdüser

Transdüserler bir enerji biçimini başka bir enerji türüne dönüştüren donanımlardır. Genellikle otomasyon, ölçüm ve kontrol sistemlerinde enerji, kuvvet, tork, ışık, hareket ve konum gibi fiziksel nicelikleri elektrik sinyallerine dönüştürmek veya tam tersi için kullanılır. Dönüştürme olarak bilinen bu süreç, çeşitli fiziksel olayların ölçülmesini, izlenmesini ve kontrol edilmesini sağlar. Transdüserler; sensörler, aktüatörler ve alıcı-vericiler olmak üzere 3 temel kategoride toplanırlar (Svantek, 2 Şubat 2025).

Bu çalışmada su altı akustik kaydı için kullanılan transdüser elektrik enerjisini ses sinyallerine, ses sinyallerini de elektrik enerjisine dönüştürerek çift yönlü dönüşüm yapmaktadır. Test düzeneğinde kullanılan transdüser Koç Bilgi ve Savunma Teknolojileri A.Ş bünyesinde üretilmektedir ve uygulama test düzeneğinde kullanılan transdüser Fotoğraf 2.3’de gösterilmektedir.



Fotoğraf 2.3. Akustik kayıt için kullanılan transdüser

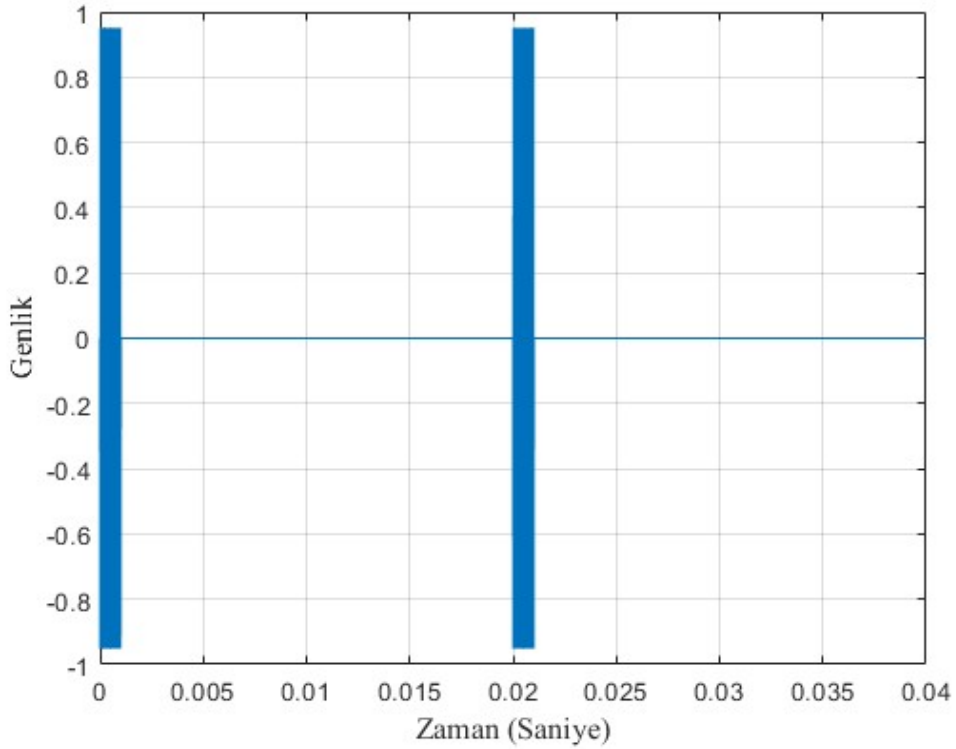
Transdüserin performansı için İletim Gerilim Yanıtı (TVR) ve Alım Gerilim Hassasiyeti (RVS) parametreleri önemlidir. Bu değerler transdüserin üretiminde belirlenen parametrelerdir. Uygulama test düzeneğinde kullanılan transdüserin 300Khz frekanslı ping sinyalinde TVR değeri 135 dB, RVS değeri -215 dB’dir.

2.10 Ping Sinyalinin Üretilmesi

Transdüser üzerinden su ortamına gönderilen ping sinyali, sabit frekans ve sabit bir süre için oluşturulmuş Sürekli Dalga (CW) bir sinüs sinyalıdır (Waite, 2002). Bu sinyal $f = 300$ khz frekansında, 1ms boyunca sürekli olarak ve $f_s = 1$ Mhz örnekleme frekansı ile oluşturulmuştur. Kullanılan transdüserin verimli olarak çalıştığı frekans 300 khz olduğu için bu frekans seçilmiştir. Oluşturulan ping sinyalinin matematiksel ifadesi (2.9)'da gösterilmektedir.

$$\text{Ping} = \begin{cases} \sin\left(\frac{2\pi f t}{f_s}\right), & \text{mod}(t, 20 \text{ ms}) < 1 \text{ ms} \\ 0, & \text{mod}(t, 20 \text{ ms}) \geq 1 \text{ ms} \end{cases} \quad (2.9)$$

Sistem çalışma süresince 1ms ping sinyali iletmekte, sonraki 19ms boyunca metal plakadan yansıyan sinyalleri kaydetmekte ve 20ms aralıklarla bu çalışma durumunu tekrar etmektedir. Oluşturulan bu yaklaşım, su altı ortamında bir yüzeye sinyal gönderip yansıyan sinyali tespit edebilmek için gerekli olan senaryonun bir gösterimidir. Oluşturulan ping sinyali 2 döngü olarak toplam 40 ms boyunca Şekil 2.15'te gösterilmektedir.



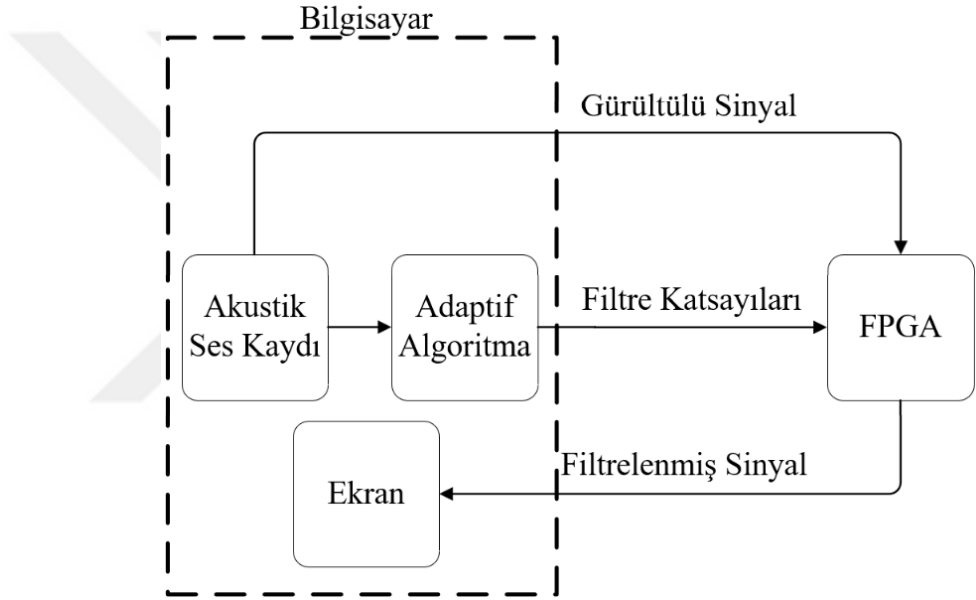
Şekil 2.15. Ping sinyali

2.11 FPGA Üzerinde Adaptif Filtreleme İşleminin Gerçeklenmesi

LMS ve RLS adaptif filtre algoritmalarının FPGA üzerinde gerçekleştirilmesi için öncelikle kullanılacak akustik ses verileri havuzda kurulan deney düzeneği üzerinden kaydedilmiştir. Sonrasında kaydedilen bu akustik ses verileri bilgisayar ortamında adaptif algoritma tarafından işlenip filtre katsayıları üretilmiştir.

Uygulama kısmında ise gürültülü akustik ses verisi UART protokolü üzerinden FPGA donanımına iletilmiş, FPGA tarafından filtre katsayıları kullanılarak işlendikten sonra filtrelenmiş akustik ses verisi UART protokolü üzerinden bilgisayar ortamına geri alınmıştır.

Uygulama süreci blok diyagramı olarak Şekil 2.16’da gösterilmektedir.



Şekil 2.16. Uygulama süreci blok diyagramı

2.11.1 LMS Algoritması Başlangıç Parametreleri

LMS algoritması başlangıcında, filtre katsayılarının bir değeri olmadığı için sıfır olarak alınmıştır. LMS algoritmasının diğer parametreleri, kaydedilen gürültülü akustik veriye uygulandığında, LMS algoritmasının parametre değerleri filtreleme işleminden sonra Sinyal-Gürültü Oranı (SNR) en iyi olacak şekilde seçilmiştir. Buradaki amaç filtreleme işleminden sonra sinyal üzerindeki gürültü etkisinin en aza indirilmek istenmesidir. Tablo 2.2’de LMS algoritması başlangıç parametre seçimleri gösterilmiştir.

Tablo 2.2. LMS algoritması başlangıç parametreleri

Parametre	Değer Seçimi
w (Filtre Katsayıları)	Başlangıç değeri olarak 0 verilmiştir.
M (Filtre Derecesi)	En iyi SNR sonucuna göre seçilmiştir.
μ (Adım Boyutu)	En iyi SNR sonucuna göre seçilmiştir.

2.11.2 RLS Algoritması Başlangıç Parametreleri

RLS algoritması başlangıcında, filtre katsayılarının bir değeri olmadığı için sıfır olarak alınmıştır. Kovaryans matrisinin başlangıç olarak belirsizdir. Model parametrelerinin bilinmezliğini ifade etmek için (2.10) eşitliğinde gösterilen başlangıç değeri kullanılır (Lampl, 2020). Burada δ değişkeni sabit bir sayıyı göstermektedir. Sistemin davranışı test edilerek uygun değer seçimi yapılmaktadır.

$$P(0) = \delta^{-1}I \quad (2.10)$$

RLS algoritmasının diğer parametreleri, LMS algoritmasında yapıldığı gibi kaydedilen gürültülü akustik veriye uygulandığında, RLS algoritmasının parametre değerleri filtreleme işleminden sonra SNR en iyi olacak şekilde seçilmiştir. Tablo 2.3’de RLS algoritması başlangıç parametreleri gösterilmiştir.

Tablo 2.3. RLS algoritması başlangıç parametreleri

Parametre	Değer Seçimi
w (Filtre Katsayıları)	Başlangıç değeri olarak 0 verilmiştir.
M (Filtre Derecesi)	En iyi SNR sonucuna göre seçilmiştir.
λ (Unutma Faktörü)	En iyi SNR sonucuna göre seçilmiştir.
P (Kovaryans Matrisi)	En iyi SNR sonucuna göre seçilmiştir.

2.12 Performans Karşılaştırma Kriterleri

Bu tezde sunulan çalışmanın uygulanabilirliği dışında, kullanılan LMS ve RLS algoritmalarının birbirleriyle performans karşılaştırmaları da yapılmıştır. Bu

karşılaştırma yapılırken SNR değeri ve FPGA kaynak tüketimi üzerinden değerlendirme yapılmıştır.

2.12.1 SNR

SNR, sinyal ve sinyal üzerindeki gürültünün birbirine oranı olarak tanımlanır (Bari vd., 2016). SNR değerinin matematiksel olarak nasıl hesaplandığı (2.11)'de gösterilmiştir. SNR değerinin yüksekliği sinyalin kalitesini belirler. Çünkü SNR değeri ne kadar yüksekse, gürültü sinyale o kadar az etki etmiş olur.

$$SNR(dB) = 10 \log \left(\frac{P_{signal}}{P_{noise}} \right) \quad (2.11)$$

Akustik veri kaydı düzeneği üzerinden kaydedilen sinyallerde, 1ms sürekli olan ping sinyali tespit edilmiş olup, 20ms içerisinde tespit edilen bu sinyalin yansımaları görülebilmektedir. Bu durumda kaydedilen sinyallerde, 1ms olan sürekli ping sinyalinden örnekleme yapılarak sinyalin gücü bulunmuş, gürültünün gücü ise yansıyan sinyal tespit edilmeden önceki zaman aralığından örnekleme alınarak hesaplanmıştır. Aynı zamanda ölçülen gürültü gücünün boş havuzun gürültü gücüne denk olduğu doğrulanmıştır. Gürültülü sinyalin SNR değeri ise 8,85 dB olarak hesaplanmıştır.

2.12.2 FPGA Kaynak Tüketimi

FPGA, üzerindeki donanım kaynakları olan Arama Tablosu (LUT), Flip-Flop (FF), Sayısal Sinyal İşleme (DSP), Giriş-Çıkış (IO), Küresel Ara Bellek (BUFG) kullanımına göre performans değerlendirmesi yapılabilmektedir.

LUT, FPGA üzerinde giriş değerlerine karşılık gelen çıkış değerlerini saklayan bir tabloyu ifade eder. FPGA'ler programlanabilir mantık devrelerini içeren entegre devreler oldukları için çok sayıda LUT içerir (fpgainsights, 16 Şubat 2025). FF, FPGA üzerinde saat döngüleri arasındaki mantığı senkronize etmek ve mantıksal durumları kaydetmek için kullanılan bellek elemanlarıdır (ni, 16 Şubat 2025). DSP, FPGA üzerinde sinyal işleme görevlerini hızlı ve verimli bir şekilde gerçekleştirmek için özel olarak tasarlanmış bileşenlerdir (ni, 16 Şubat 2025). IO, FPGA donanımının dış dünya ile iletişimini sağlayan giriş ve çıkış bağlantılarıdır. BUFG, FPGA üzerinde saat sinyallerinin doğru ve zamanında iletilmesini sağlayan donanım bileşenidir.

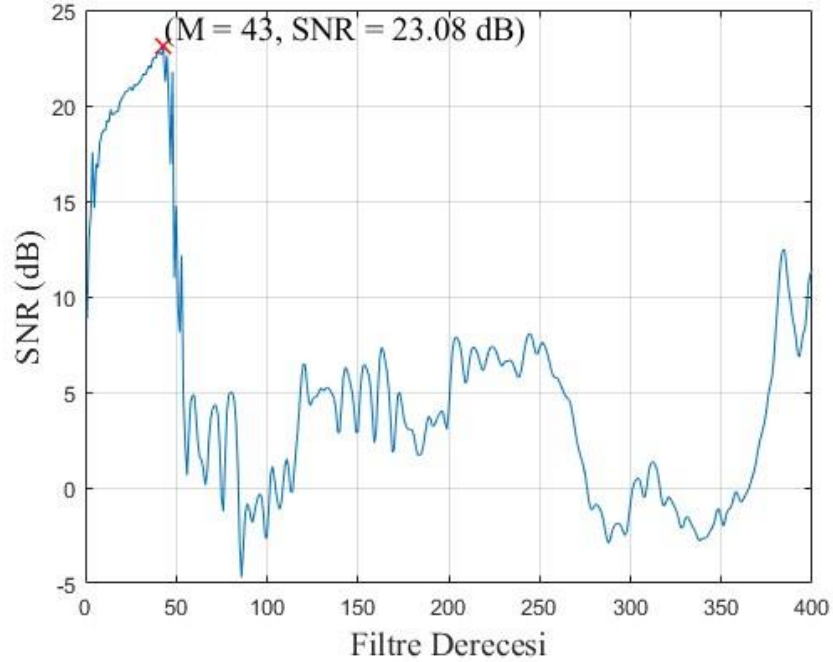
Bu çalışma içerisinde, sentezleme işleminden sonra FPGA donanımının bu kaynakları kullanma durumuna göre performans değerlendirmesi yapılmıştır.

3. BULGULAR

Bu bölümde LMS ve RLS algoritmalarının gerçekleştirme işlemi sonucunda elde edilen sonuçlar gösterilmekte, FPGA kaynak tüketimi ve SNR değeri üzerinden elde edilen veriler karşılaştırılmaktadır.

3.1 LMS Algoritması Gerçeklenme Sonucu

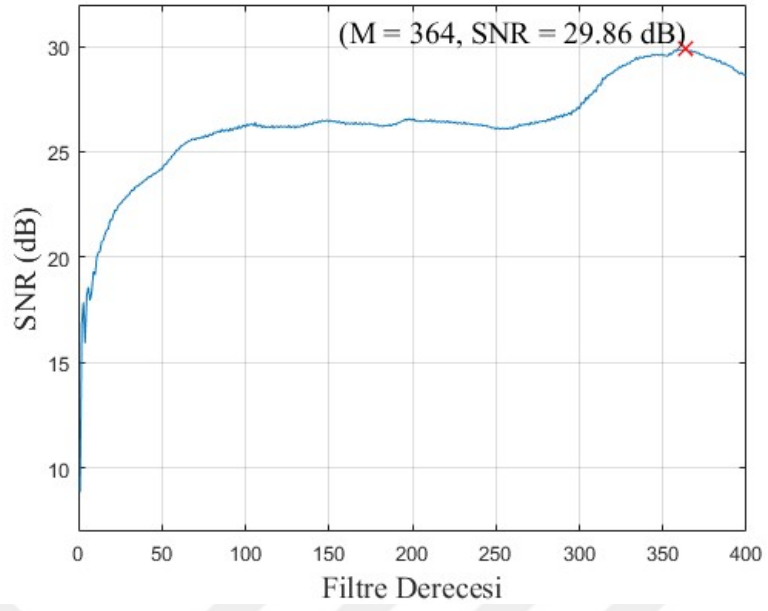
LMS algoritmasının gürültülü sinyale uygulanması esnasında, farklı adım boyutu ve farklı filtre derecelerinde sistem davranışı test edilmiştir. En yüksek SNR değerini hangi filtre derecesinde ve adım boyutunda verdiği bulunmuştur. $\mu = 0,1$ olarak alındığında SNR değişimi Şekil 3.1’de gösterilmiştir.



Şekil 3.1. $\mu = 0,1$ değerinde en yüksek SNR değeri değişimi

$M = 43$ ve $\mu = 0,1$ alındığında sistem en yüksek SNR değeri olan 23,08 dB değerine ulaşmıştır.

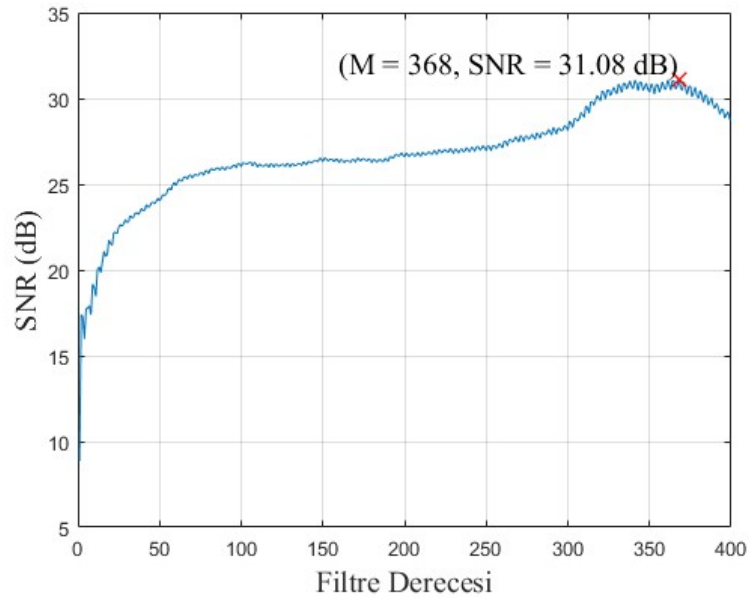
$\mu = 0,01$ olarak alındığında SNR değişimi Şekil 3.2’de gösterilmiştir.



Şekil 3.2. $\mu = 0,01$ değerinde en yüksek SNR değeri değişimi

M = 364 ve $\mu = 0,01$ alındığında sistem en yüksek SNR değeri olan 29,86 dB değerine ulaşmıştır.

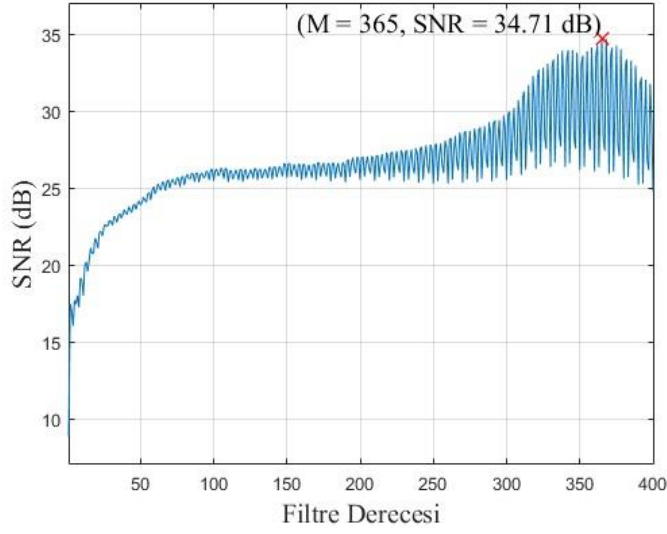
$\mu = 0,001$ olarak alındığında SNR değişimi Şekil 3.3’de gösterilmiştir.



Şekil 3.3. $\mu = 0,001$ değerinde en yüksek SNR değeri değişimi

M = 368 ve $\mu = 0,001$ alındığında sistem en yüksek SNR değeri olan 31,08 dB değerine ulaşmıştır.

$\mu = 0,0001$ olarak alındığında SNR değişimi Şekil 3.4’de gösterilmiştir.

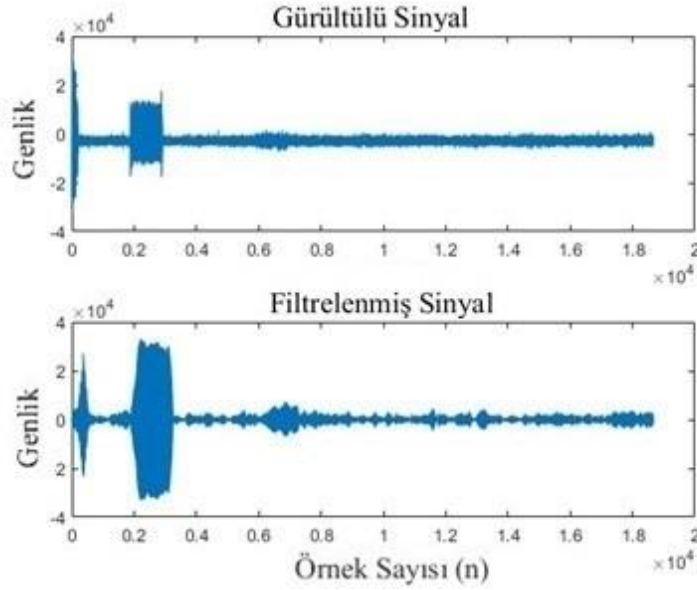


Şekil 3.4. $\mu = 0,0001$ değerinde en yüksek SNR değeri değişimi

$M = 365$ ve $\mu = 0,0001$ alındığında sistem en yüksek SNR değeri olan 34,71 dB değerine ulaşmıştır.

LMS algoritmasının farklı μ ve M uygulanması sonucunda, en yüksek SNR değerine $\mu = 0,0001$ ve $M = 365$ olarak alındığında ulaşıldığı görülmüştür. Buna göre gürültülü sinyalin SNR değeri 8,85 dB'den 34,71 dB değerine ulaşmıştır.

$\mu = 0,0001$ ve $M = 365$ alınarak FPGA üzerinde gerçekleştirme yapıldığında filtrelenen sinyal Şekil 3.5'de gösterilmiştir.



Şekil 3.5. Gürültülü sinyal ve FPGA üzerinde filtrelenen sinyal

FPGA üzerinde gereklenme iřleminden sonra donanım kaynak kullanımları Tablo 3.1’de gsterilmektedir.

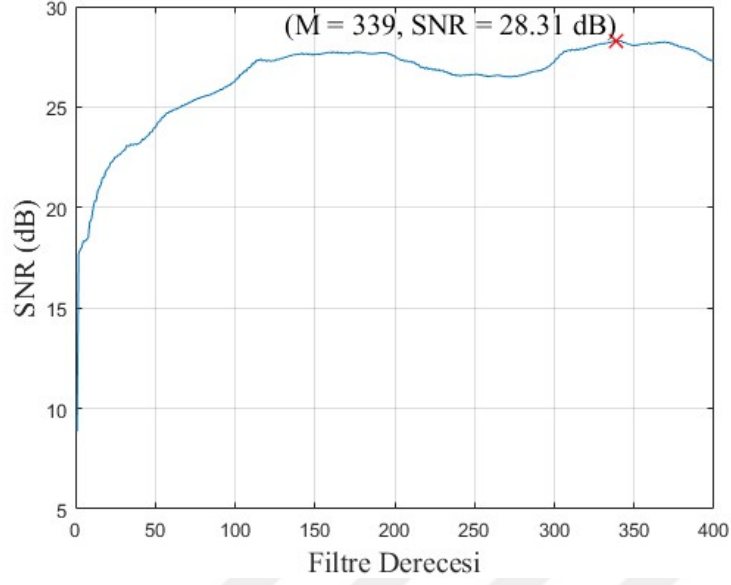
Tablo 3.1. LMS algoritması FPGA donanım kaynak kullanımları

Donanım Adı	Kullanılan	Kullanılabilir	Kullanım Yüzdesi
LUT	1790	63400	%2,82
FF	6103	126800	%4,81
DSP	1	240	%0,42
IO	4	210	%1,9
BUFG	1	32	%3,13

3.2 RLS Algoritması Gereklenme Sonucu

RLS algoritmasının grltl sinyale uygulanması esnasında, farklı unutma faktrleri, farklı filtre dereceleri ve kovaryans matrisinin farklı bařlangı deęerlerinde sistem davranıřı test edilmiřtir. En yksek SNR deęerini hangi filtre derecesinde, unutma faktrnde ve kovaryans matrisi bařlangı deęerinde verdięi bulunmuřtur.

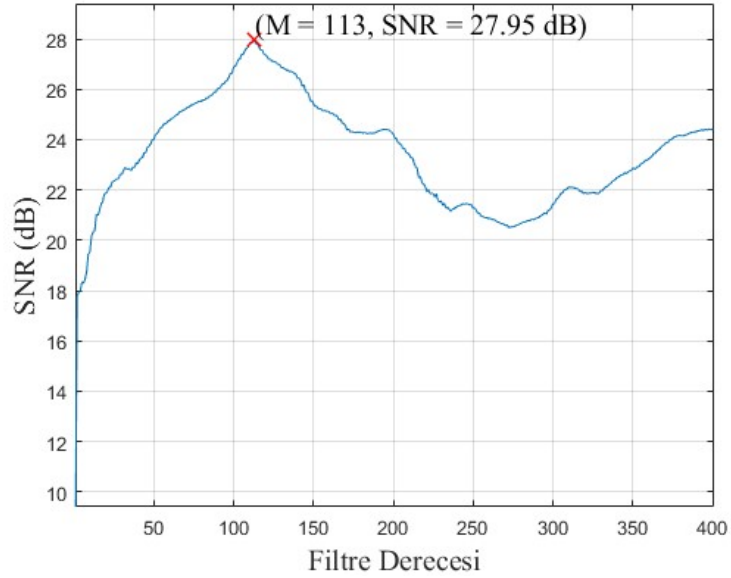
$\lambda = 0,99$ ve kovaryans matrisi bařlangı deęeri iin $\delta = 1000$ alındıęında filtre derecesi ve SNR deęiřimi řekil 3.6’da gsterilmektedir.



Şekil 3.6. $\lambda = 0,99$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi

$M = 339$, $\lambda = 0,99$ ve $\delta = 1000$ alındığında sistem en yüksek SNR değeri olan 28.31 dB değerine ulaşmıştır.

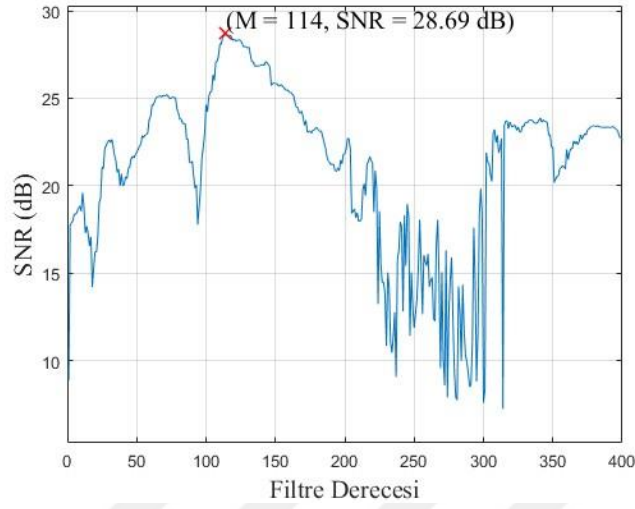
$\lambda = 0,99$ ve kovaryans matrisi başlangıç değeri için $\delta = 1$ alındığında filtre derecesi ve SNR değişimi Şekil 3.7’de gösterilmektedir.



Şekil 3.7. $\lambda = 0,99$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi

$M = 113$, $\lambda = 0,99$ ve $\delta = 1$ alındığında sistem en yüksek SNR değeri olan 27.95 dB değerine ulaşmıştır.

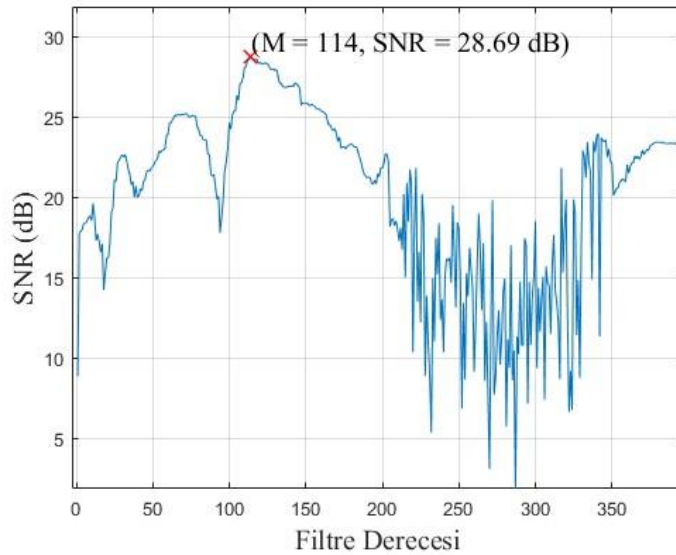
$\lambda = 0,9$ ve kovaryans matrisi başlangıç değeri için $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi Şekil 3.8’de gösterilmektedir.



Şekil 3.8. $\lambda = 0,9$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi

$M = 114$, $\lambda = 0,9$ ve $\delta = 1000$ alındığında sistem en yüksek SNR değeri olan 28.69 dB değerine ulaşmıştır.

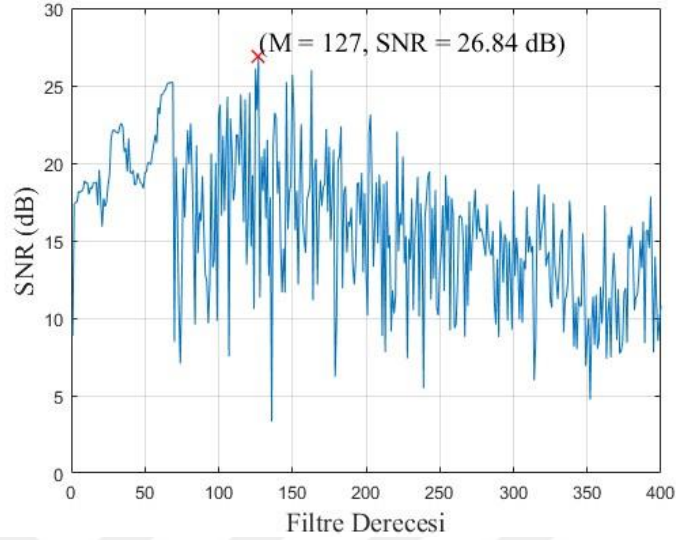
$\lambda = 0,9$ ve kovaryans matrisi başlangıç değeri için $\delta = 1$ alındığında filtre derecesi ve SNR değişimi Şekil 3.9’da gösterilmektedir.



Şekil 3.9. $\lambda = 0,9$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi

$\lambda = 0,9$ alındığında, δ değişimi en yüksek SNR ve filtre derecesi değerinde bir farklılık oluşturmamıştır.

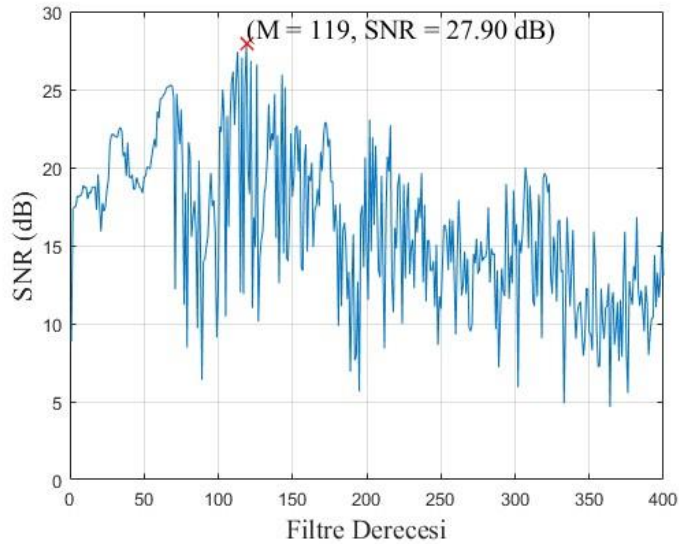
$\lambda = 0,7$ ve kovaryans matrisi başlangıç değeri için $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi Şekil 3.10’da gösterilmektedir.



Şekil 3.10. $\lambda = 0,7$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi

$M = 127$, $\lambda = 0,7$ ve $\delta = 1000$ alındığında sistem en yüksek SNR değeri olan 26.84 dB değerine ulaşmıştır.

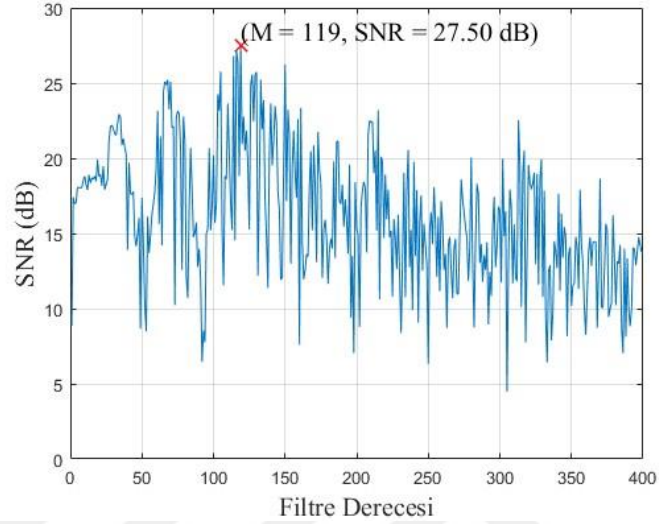
$\lambda = 0,7$ ve kovaryans matrisi başlangıç değeri için $\delta = 1$ alındığında filtre derecesi ve SNR değişimi Şekil 3.11’de gösterilmektedir.



Şekil 3.11. $\lambda = 0,7$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi

$M=119$, $\lambda = 0,7$ ve $\delta = 1$ alındığında sistem en yüksek SNR değeri olan 27.90 dB değerine ulaşmıştır.

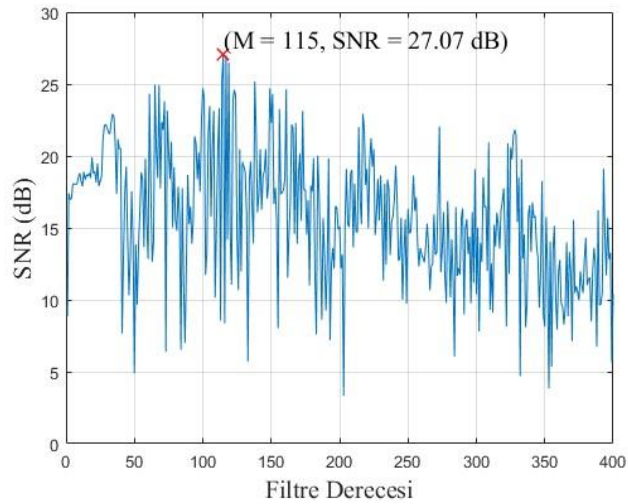
$\lambda = 0,5$ ve kovaryans matrisi başlangıç değeri için $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi Şekil 3.12’de gösterilmektedir.



Şekil 3.12. $\lambda = 0,5$ ve $\delta = 1000$ alındığında filtre derecesi ve SNR değişimi

$M=119$, $\lambda = 0,5$ ve $\delta = 1000$ alındığında sistem en yüksek SNR değeri olan 27.50 dB değerine ulaşmıştır.

$\lambda = 0,5$ ve kovaryans matrisi başlangıç değeri için $\delta = 1$ alındığında filtre derecesi ve SNR değişimi Şekil 3.13’de gösterilmektedir.

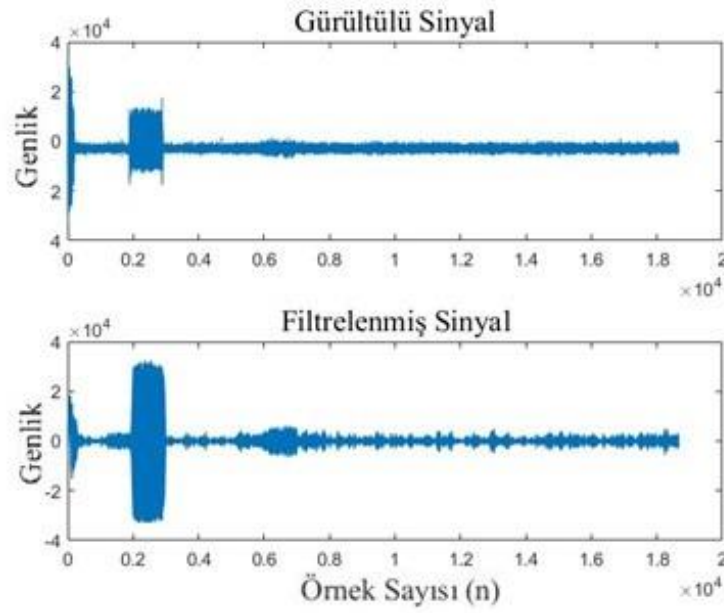


Şekil 3.13. $\lambda = 0,5$ ve $\delta = 1$ alındığında filtre derecesi ve SNR değişimi

$M=115$, $\lambda = 0,5$ ve $\delta = 1$ alındığında sistem en yüksek SNR değeri olan 27,07 dB değerine ulaşmıştır.

RLS algoritmasının farklı λ ve δ değerlerinde uygulanması sonucunda, en yüksek SNR değerine $\lambda = 0,9$ ve $\delta = 1000$ olarak alındığında filtre derecesi 114 olduğunda ulaştığı görülmüştür. Buna göre gürültülü sinyalin SNR değeri 8,85 dB'den 28,69 dB değerine ulaşmıştır.

$\lambda = 0,9$ ve $\delta = 1000$ alınarak FPGA üzerinde gerçekleştirme yapıldığında filtrelenen sinyal Şekil 3.14'de gösterilmektedir.



Şekil 3.14. Gürültülü sinyal ve FPGA üzerinde filtrelenmiş sinyal

FPGA üzerinde gerçekleştirme işleminden sonra donanım kaynak kullanımları Tablo 3.2'de gösterilmektedir.

Tablo 3.2. RLS algoritması FPGA donanım kaynak kullanımları

Donanım Adı	Kullanılan	Kullanılabilir	Kullanım Yüzdesi
LUT	675	63400	%1,06
FF	2069	126800	%1,63
DSP	1	240	%0,42
IO	4	210	%1,9
BUFG	1	32	%3,13

LMS ve RLS algoritmalarının gereklenme iřlemi sonucunda SNR deęerleri ve kaynak kullanımları karřılařtırması Tablo 3.3’de gsterilmektedir.

Tablo 3.3. LMS ve RLS algoritmaları gereklenme sonuları

Algoritma	SNR	LUT	FF	DSP	IO	BUFG
LMS Algoritması	34,71 dB	%2,82	%4,81	%0,42	%1,9	%3,13
RLS Algoritması	28,69 dB	%1,06	%1,63	%0,42	%1,9	%3,13

4. TARTIŞMA

Sunulan bu tez çalışması, su altı akustiğinde kullanılan haberleşme sinyallerindeki gürültü etkisini, LMS ve RLS adaptif filtre algoritmaları kullanarak en aza indirmek ve bu algoritmaların FPGA donanımı üzerinde gerçekleştirilebilirliğini göstermek amacıyla uygulamalı olarak gerçekleştirilmiştir.

Çalışma sürecinde, LMS ve RLS algoritmalarının literatürdeki kullanım alanları incelenmiş, yapılan çalışmalarda adaptif filtrelerin gürültü etkisini azaltma uygulamalarının başarıyla sonuçlandığı görülmüştür. LMS ve RLS algoritmalarının su altı akustik sinyallerde uygulanması ve FPGA üzerinde gerçekleştirilmesi için yeterli kadar çalışma olmamasından dolayı bu tez çalışmasının literatüre katkı sağlaması amaçlanmıştır.

Çalışma için, hem ping sinyali üretip hem de sinyalleri kaydedebilen çift yönlü bir transdüser kullanılarak su altı ortam koşullarını sağlayabilmek için havuz ortamında deney düzeneği kurulmuştur.

Deney düzeneği ile kaydedilen su altı akustik ses verisi incelendiğinde metal plakadan yansıyan ping sinyali görülebilmektedir.

Kaydedilen bu sinyale bilgisayar ortamında LMS ve RLS algoritması uygulanmıştır.

LMS algoritmasında, gerçekleştirme işlemi için en yüksek SNR değerini veren parametreler dikkate alınarak filtre katsayıları ve filtre derecesi belirlenmiş, adım boyutu parametresinin SNR değerinde ve filtre derecesinde etkili olduğu görülmüştür. LMS algoritmasıyla belirlenen filtre katsayıları FPGA donanımına iletilerek filtreleme işlemi gerçekleştirilmiş ve filtrelenen sinyalde gürültü etkisinin azaldığı gözlemlenmiştir. LMS algoritması, 8,85 dB olan SNR değerini 34,71 dB değerine çıkartmış, FPGA üzerinde %2,68 LUT, %4,7 FF, %0,42 DSP, %1,9 IO ve %3,13 BUFG kaynak kullanımı gerçekleştirmiştir.

RLS algoritmasında, gerçekleştirme işlemi için en yüksek SNR değerini veren parametreler dikkate alınarak filtre katsayıları ve filtre derecesi belirlenmiş, unutma faktörü parametresi ve kovaryans matrisinin başlangıç parametresinin SNR değeriyle filtre derecesinde etkili olduğu görülmüştür. RLS algoritmasıyla belirlenen filtre katsayıları FPGA donanımına iletilerek filtreleme işlemi gerçekleştirilmiş ve filtrelenen sinyalde gürültü etkisinin azaldığı gözlemlenmiştir. RLS algoritması, 8,85 dB olan SNR değerini 28,69 dB değerine çıkartmış, FPGA

üzerinde %1,06 LUT, %1,63 FF, %0,42 DSP, %1,9 IO ve %3,13 BUFG kaynak kullanımı gerçekleştirmiştir.

Gerçeklenme sonuçlarına göre daha yüksek SNR tercihinde LMS algoritması, daha az kaynak kullanımı tercihinde RLS algoritması etkili görülmektedir.

LMS ve RLS algoritmalarının uygulanmasında, başlangıç parametrelerinin doğru belirlenmesinin algoritma performansı üzerinde önemli bir etkisi olduğu gözlemlenmiştir. Bu parametreler, kullanılan sinyalin özelliklerine bağlı olarak değişiklik gösterebilir ve sistem davranışı test edilerek optimize edildiklerinde, uygulamanın performansını daha verimli hale getirdiği görülmüştür.



5. SONUÇ ve ÖNERİLER

Bu çalışma, su altı akustiğinde adaptif filtrelerin sinyal işleme süreçlerinde etkili bir çözüm sunduğunu ve FPGA donanımı üzerinde başarılı bir şekilde uygulanabileceğini göstermektedir. Çalışmadan elde edilen sonuçlar ise, su altı akustiği uygulamalarında adaptif filtrelerin denizaltı iletişimi, sonar sistemleri ve diğer akustik algılama teknolojilerinde önemli bir rol oynayabileceğini ve bu tür sistemlerde gürültü engelleme, sinyal iyileştirme gibi kritik görevleri yerine getirebileceğini ortaya koymaktadır.

Gelecekte yapılacak çalışmalarda, farklı adaptif filtre algoritmalarının su altı akustiği alanında karşılaşılan zorluklara ne kadar uyum sağlayabileceği ve ne tür iyileştirmeler yapılabileceği üzerine ayrıntılı çalışmaların olabileceği düşünülmektedir. Özellikle filtrelerin adaptif parametrelerinin çevresel koşullara ve sistem gereksinimlerine bağlı olarak otomatik şekilde optimize edilebilmesi, sistem performansı üzerinde önemli ölçüde artışa neden olabileceği öngörülmektedir. Bu noktada, yapay zekâ ve makine öğrenimi tekniklerinin entegrasyonu, sistemin daha dinamik ve çevresel değişimlere daha hızlı adapte olmasını sağlayabilir.

Su altı akustiği ve FPGA tabanlı adaptif filtrelerin entegrasyonu üzerine yapılacak bu çalışmalar deniz bilimleri, savunma sanayi ve denizaltı keşif teknolojileri gibi birçok alanda yenilikçi çözümler üretilmesine olanak tanıyacaktır.

KAYNAKLAR

- Almallahmed, O.M. (2015). *Noise Cancelling Using Adaptive Filter Algorithms* [Yayımlanmış Yüksek Lisans Tezi], Çankaya Üniversitesi.
- Bari, M., Dewasthale, M.M. (2016, 9-11 Haziran). *Performance Comparison of LMS Algorithms for Acoustic Noise Cancellation*. 2016 Conference on Advances in Signal Processing (CASP), Pune, India.
- Bellizia, D., Monsurrò, P., Trifiletti, A. (2017, 04-06 Eylül). *VHDL Implementation of FWL RLS Algorithm*. 2017 European Conference on Circuit Theory and Design (ECCTD), Catania, Italy.
- Bharade, P., Joshi, Y., Manthalkar, R. (2016, 09-11 Haziran). *Design and Implementation of IIR lattice filter using floating point arithmetic in FPGA*. 2016 Conference on Advances in Signal Processing (CASP), Pune, India.
- Cai, G., Yang, J., Liang, C., Li, H. (2013, 23-24 Aralık). *Design and Implementation of LMS Adaptive Filter Algorithm Based on FPGA*. 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), Toronto, ON, Canada.
- Coates, R. F. (1990). *Underwater Acoustic Systems* (1. Baskı), Wiley.
- Çavuşlu, M. A. (2020). *Sualtı Akustiği için Veri Temelli Kanal Kestirimi ve Modellenmesi* [Yayımlanmış Doktora Tezi], Kocaeli Üniversitesi.
- Digilent. (2025). Verilog HDL Background and History. https://digilent.com/reference/learn/fundamentals/digital-logic/verilog-hdl-background-and-history/start?srsId=AfmBOopwq6CFq4DicTH2_3JBWSt9ih9swGYU01XO38vuRURikNkDW1nW adresinden 22 Ocak 2025 tarihinde alınmıştır.
- Elhossini, A., Areibi, S., Dony, R. (2006, 20-22 Eylül). *An FPGA Implementation of The LMS Adaptive Filter for Audio Processing*. 2006 IEEE International Conference on Reconfigurable Computing and FPGA's (ReConFig 2006), San Luis Potosi, Mexico.
- Esposito, D., Meo, G. D., Caro, D. D., Strollo, A. G. M., Napoli, E. (2018, 09-12 Aralık). *Quality-Scalable Approximate LMS Filter*. 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Bordeaux, France.
- Fisheries. (2025). Ocean Noise. <https://www.fisheries.noaa.gov/national/science-data/ocean-noise> adresinden 3 Mayıs 2025 tarihinde alınmıştır.
- FPGA Insights. (2025). Overview of Lookup Tables Lut in FPGA Design. <https://fpgainsights.com/fpga/overview-of-lookup-tables-lut-in-fpga-design/> adresinden 16 Şubat 2025 tarihinde alınmıştır.
- Ganorkar, S., Sharma, S., Jain, A. (2022, 23-24 Nisan). *Implementation of FIR Adaptive Filter with FPGA*. 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT), Indore, India.
- Harris, S., Harris, D. (2022). *Digital Design and Computer Architecture, RISC-V Edition* (1. Baskı), Morgan Kaufmann.
- Haykin, S. (1996). *Adaptive Filter Theory* (3. Baskı), Prentice Hall.
- Kumar, N. (2013). *Optimal Design of FIR and IIR Filters using some Evolutionary Algorithms* [Yayımlanmış Yüksek Lisans Tezi], National Institute of Technology Durgapur.

- Lamp, T. (2020). *Implementation of Adaptive Filtering Algorithms for Noise Cancellation* [Yayımlanmış Yüksek Lisans Tezi], Gävle Üniversitesi.
- Liang, P., Guocang, S., Haihua, D., Ming, C. (2013, 23-24 Aralık). *FPGA Implementation for A Recursive Least Square Algorithm*. 2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA), Toronto, ON, Canada.
- Liu, Y., Sun, H., Liu, L., Ma, Y. (2023, 09-11 Ağustos). *Implementation of LMS Algorithm Based Adaptive Filter Using Verilog HDL*. 2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI), Harbin, China.
- Maraş, M., Ayvaz, E.N., Özen, A. (2018, 14-16 Ekim). *A Novel Adaptive Variable Forgetting Factor RLS Algorithm*. 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey.
- Medeiros, C.A.G., Belchior, F.N., de Oliveira, J.C. (2004, 12-15 Eylül). *A Proposal for UIE/IEC Flicker Measurement Using Digital Filters: Equipment Prototype and Validation*. 2004 11th International Conference on Harmonics and Quality of Power (IEEE Cat. No.04EX951), Lake Placid, NY, USA.
- Meral, O. (2008). *Doğrusal Öngörülü Kodlama ve Adaptif Algoritma Tabanlı Konuşmacı Tanıma* [Yayımlanmış Yüksek Lisans Tezi], Kocaeli Üniversitesi.
- Mustafa, R., Mohd Ali, M.A., Umat, C., Al-Asady, D.A. (2009, 04-06 Ekim). *Design and Implementation of Least Mean Square Adaptive Filter on Altera Cyclone II Field Programmable Gate Array for Active Noise Control*. 2009 IEEE Symposium on Industrial Electronics & Applications, Kuala Lumpur, Malaysia.
- N, L., Das, M. J., Chunkath, Job. (2024, 6-7 Aralık). *Efficient FPGA Implementation of the Delayed LMS Algorithm*. 2024 International Conference on Smart Electronics and Communication Systems (ISENSE). Kottayam, India.
- National Instruments. (2025). *FPGA Fundamentals*. <https://www.ni.com/en/shop/electronic-test-instrumentation/add-ons-for-electronic-test-and-instrumentation/what-is-labview-fpga-module/fpga-fundamentals.html?srsId=AfmBOorPtafT8aGPXYvk2iUpZeo-GJ6j-s-10pYAv7ZKhwrCTGuoNaH5> adresinden 16 Şubat 2025 tarihinde alınmıştır.
- Pedroni, V. A. (2020). *Circuit Design with VHDL* (3. Baskı), The MIT Press.
- Pitchaiah, T., Lakshmi M, D., Devi, P.V.S. (2014, 01-02 Ağustos). *FPGA Implementation of Low Area and Delay Efficient Adaptive Filter Using Distributed Arithmetic*. 2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014), Unnao, India.
- Roberts, M. J. (2017). *Signals and Systems Analysis Using Transform Methods and MATLAB®* (3. Baskı), McGraw-Hill Education.
- Rohde-Schwarz. (2025). *Understanding UART*. https://www.rohde-schwarz.com/cz/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html adresinden 28 Ocak 2025 tarihinde alınmıştır.
- S, K., G, S., M, S., K, Y. (2023, 23-25 Şubat). *An Efficient FPGA Implementation of the Multiplier-less LMS Adaptive Filter*. 2023 7th International Conference on Computing Methodologies and Communication (ICCMC). Erode, India.
- Safarian, C., Ogunfunmi, T., Kozacky, W. J., Mohanty, B.K. (2015, 21-24 Temmuz). *FPGA Implementation of LMS-Based FIR Adaptive Filter for Real Time Digital Signal Processing Applications*. 2015 IEEE International Conference on Digital Signal Processing (DSP), Singapore, Singapore.
- Sharma, A. (2001). *Development of a Place and Route Tool for the RaPiD Architecture* [Yayımlanmış Yüksek Lisans Tezi], University of Washington.

- Shiva, A., Senthilkumar, E., Manikandan, J., Agrawal, V.K. (2017, 22-24 Mart). *FPGA Implementation of Reconfigurable Adaptive Filters*. 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India.
- Stanciu, C., Anghel, C., Stanciu, L. (2015, 09-10 Temmuz). *Efficient FPGA Implementation of The DCD-RLS Algorithm for Stereo Acoustic Echo Cancellation*. 2015 International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania.
- Svantek. (2025). Transducer. <https://svantek.com/academy/transducer/> adresinden 2 Şubat 2025 tarihinde alınmıştır.
- Traverso, F., Vernazza, G., Trucco, A. (2012, 21-24 Mayıs). *Simulation of non-White and non-Gaussian Underwater Ambient Noise*. 2012 Oceans – Yeosu, Yeosu, Korea (South).
- Tudosa, I., Adochiei, N. (2012, 25-27 Ekim). *FPGA Approach of An Adaptive Filter for ECG Signal Processing*. 2012 International Conference and Exposition on Electrical and Power Engineering, Iasi, Romania.
- Urick, R. J. (1975). *Principles of Underwater Sounds* (1. Baskı), McGraw- Hill.
- Vijayaraghavan, R., Raajan, N. R., Sowmiya, P., Ragavi, A., Manoorubini, R., Selvi, T.A. (2014, 09-11 Ocak). *Inhibition of Acoustic Noise Using an Adaptive LMS Filter*. 2014 International Conference on Advances in Electrical Engineering (ICAEE), Vellore, India.
- Waite, A. D. (2002). *Sonar for Practising Engineers* (3. Baskı), Wiley.
- Wang, J. (2009, 27-29 Ekim). *A Variable Forgetting Factor RLS Adaptive Filtering Algorithm*. 2009 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, Beijing, China.

EKLER

EK 1 LMS Filtre Matlab Uygulama Kodu

```
load matlab.mat;
fs=1e6;
f1=3e5;
t=0:1/fs:1e-3;

D=sin(2*pi*f1*t);

n=numel(D);
X=raw_data;
dizi_min = min(X);
dizi_max = max(X);
dizi_norm_0_1 = (X - dizi_min) / (dizi_max - dizi_min);
dizi_norm_neg1_1 = 2 * dizi_norm_0_1 - 1;
X=dizi_norm_neg1_1;

M=365;
w=zeros(1,M);
E = zeros(1, n);
Y = zeros(1, n);
mu=0.0001;

for i=M:n
    x1 = X(i:-1:i-M+1);
    Y(i)=w * x1';
    E(i) = D(i) - Y(i);
    w=w+2*mu*E(i).*x1;
end
```

EK 2 RLS Filtre Matlab Uygulama Kodu

```
load matlab.mat;
fs=1e6;
f1=3e5;
t=0:1/fs:1e-3;

D=sin(2*pi*f1*t);

n=numel(D);
X=raw_data;
nt = (1:length(X));

dizi_min = min(X);
dizi_max = max(X);
dizi_norm_0_1 = (X - dizi_min) / (dizi_max - dizi_min);
dizi_norm_neg1_1 = 2 * dizi_norm_0_1 - 1;
X=dizi_norm_neg1_1;

M=114;
lambda=0.9;
w = zeros(M, 1);
P = eye(M)/1000;

for i=M:n
    u = X(i:-1:i-M+1)';
    y(i) = w' * u;
    error(i) = D(i) - y(i);
    k = (P * u) / (lambda + u' * P * u);
    w = w + k * error(i);
    P = (P - k * u' * P) / lambda;
end
```