

Uydu Haberleşme Sistemlerinde Optik Haberleşme Optimizasyonu ve Entegre Uygulamaları

Hilal ŞEN, Aytaç ALTAN
Zonguldak Bülent Ecevit Üniversitesi,
Zonguldak

Taylan Sipahi, Haluk Gözde
TUSAŞ, Ankara

ÖZET

Bu çalışmada, uydu haberleşme sistemlerinde kritik bir rol oynayan atmosferik gürültünün etkilerini ve lazer tabanlı haberleşme sistemlerinin performansını artırmak için kullanılan filtreleme teknikleri ele alınmıştır. Uydu haberleşmesi için filtreli lazer haberleşmesi yazılımının geliştirilmesi süreci, Python üzerinde detaylı bir biçimde incelenmiştir. Ayrıca MATLAB ve GNU Octave gibi program çıktıları ve bilimsel karşılaştırmaları üzerinden de örnekler verilmiştir. Radyo frekans (RF) ve optik bant genişliklerinin karşılaştırılmasıyla başlayan çalışma, hangi frekans bandında daha etkin haberleşme sağlandığının analizini içermektedir. Yazılımın temel akışını ve işlevselliğini tanımlayan bir akış diyagramı geliştirilmiş ve bu diyagrama uygun olarak Python dilinde bir yazılım geliştirilmiştir. Bulgularımız, lazer tabanlı haberleşme sistemlerinin geliştirilmesi ve atmosferik gürültü etkilerinin azaltılması konusunda yol gösterici olmuştur. Proje sonunda, edilmiş olunan bilgiler ışığında ve gelecekte yapılacak olan çalışmalar için çalışmamız bir yol gösterici olacaktır.

Anahtar Kelimeler — *Uydu haberleşme, Atmosferik gürültü, Lazer tabanlı haberleşme, Filtreleme teknikleri, Python.*

GİRİŞ

Günümüz toplumunun temel yapıtaşları, hızlı ve güvenilir bilgi alışverişine dayanmaktadır. Gelişmiş iletişim sistemleri, savunma sanayii, havacılık, taşımacılıktan küresel bilgi akışına kadar geniş bir yelpazede yer almaktadır. Özellikle, yüksek kaliteli gerçek zamanlı video iletimi, enerji kaybını azaltarak çevresel sürdürülebilirliğe katkıda bulunmaktadır. Ayrıca, gelişen yarı iletken endüstrisi ile uzay araçları ve insan taşımacılığı için iletişim ihtiyaçları artmaktadır. Kablosuz cihazların evrimi, son otuz yılda inanılmaz bir hızla ilerlemiştir ve bu gelişmeler, modern toplumun temel öğeleri haline gelmiştir [Khalighi,2014]. "Kablosuz Haberleşme" terimi, RF cihazlarının benimsenmesiyle özdeşleşmiş olup, RF spektrumunun sınırlı ve pahalı kapasitesi nedeniyle, alternatif haberleşme teknolojilerinin değerlendirilmesi gerekmektedir. Kablosuz Optik Haberleşme (KOH), yüksek bant genişliği, veri güvenliği ve uygulama kolaylığı ile ön plana çıkmaktadır. KOH, ışık spektrumunun görünür, ultraviyole ve kızılötesi segmentlerinde çalışır. Tarihsel olarak, güneş ışığı ve heliograflar gibi eski iletişim araçları, mesaj iletiminde kullanılmıştır. En son KOH versiyonları, iletişimde lazerler ve LED'ler gibi yüksek teknoloji cihazları kullanmaktadır. KOH üzerine yapılan teknolojik gelişmeler, özellikle derin uzay iletişimi gibi alanlarda ulusal güç yarışlarına neden olmaktadır.

Serbest Uzay Optiği (SUO), doğal yüksek taşıyıcı frekanslar kullanarak yüksek veri hızlarında iletişim sağlamaktadır. SUO, noktadan noktaya çözümler ve yeni nesil yüksek hızlı optik ağlar için uygun olup, çeşitli uygulamalarda etkin bir şekilde kullanılmaktadır [Nadeem,2009]. Ancak, iklim koşulları nedeniyle SUO'nun güvenilirliği ve kullanılabilirliği sınırlı olabilir. Genel olarak, SUO sistemleri, büyük veri iletimi sağlayarak kısa mesafeli iletişim için idealdir. Ancak, hava koşulları ve optik bileşenlerin sürekli izlenmesi, bu sistemlerin adaptasyonunu zorlaştırmaktadır. Faz Kaydırmalı Anahtarlama ve Karesel Genlik Modülasyonu gibi yeni modülasyon teknikleri, SUO sistemlerinin performansını artırmak için geliştirilmiştir. Bu teknikler, daha iyi spektral verimlilik sağlamak ve iletişim kalitesini iyileştirmek amacıyla kullanılmaktadır. Projede SUO'nun teorisini

Hilal ŞEN, Elektrik-Elektronik Müh. Böl., E-posta: hilalsen5555@gmail.com

Doç.Dr. Aytaç ALTAN, Elektrik-Elektronik Müh. Böl., E-posta: aytacaltan@beun.edu.tr

Dr. Taylan SİPAHI, TUSAŞ-Teknoloji Geliştirme Lider Mühendisi, AYBU-Uzay ve Havacılık Misafir Dr. Öğretim Üyesi, E-posta: taylan.sipahi@tai.com.tr, taylan.sipahi@aybu.edu.tr

Doç.Dr. Haluk GÖZDE, TUSAŞ-Yazılım Mühendisliği Direktörlüğü, E-posta: haluk.gozde@tai.com.tr

kavramak ve bu doğrultuda parametrelere sahip adaptif dijital filtreyi tercih etmek başlıca amaçlardan biridir.

Adaptif filtreler, özellikle optik haberleşme sistemlerinde karşılaşılan teknik zorluklar nedeniyle karmaşık olabilir. NLMS adaptif filtre, sistem ayırıştırma yöntemi olarak kullanılarak hem kararlılığı hem de performansı optimize eder. Bu filtrenin uygulanması sırasında, algoritmanın ortam koşullarına duyarlılığı ve gürültülü sinyallerin varlığı gibi sorunlar ortaya çıkabilir [Lampl,2020]. Bunun yanı sıra, haberleşme ortamındaki karmaşıklıklar filtre performansını etkileyebilir. Python, MATLAB ve Octave gibi farklı yazılım ortamları karşılaştırıldığında; Python, açık kaynaklı olması, geniş kütüphane ekosistemine erişimi ve çeşitli işletim sistemlerinde çalışabilmesi nedeniyle öne çıkmıştır. Hız, doğruluk, CPU/RAM performansı gibi kriterler doğrultusunda Python tercih edilmiştir.

NLMS filtresinin parametreleri, filtre performansını doğrudan etkiler. Adım boyutu, filtre katsayılarının ne kadar hızlı güncelleneceğini belirler; çok büyük bir değer filtre kararsızlığına, çok küçük bir değer ise yavaş uyuma yol açabilir. Filtre uzunluğu, filtre hafızasını tanımlar ve hesaplama maliyetini artırırken, unutma faktörü geçmiş hataların adaptasyon sürecine etkisini düzenler [Patnaik,2024]. Ayrıca, algoritma giriş ve istenen sinyallerin eşit uzunlukta olmasını gerektirir, aksi halde hata verir. Algoritma ayrıca veri türlerini ve parametre değerlerini kontrol ederek kuvvetli bir yapı sunar, yanlış parametreler tanımlandığında hata mesajları ile yanıt verir. Bu özellikler, NLMS adaptif filtrenin etkin ve güvenilir bir şekilde çalışmasını sağlar.

YÖNTEM

Atmosferik Gürültünün Modellenmesi: Hem teorik dağılım modelleri hem de gerçek dünya verilerini kullanarak gerçekleştirilir. Bu model, rastgele gürültü kaynaklarını ve atmosferik koşulların zamanla değişimini hesaba katmakta, böylece sinyal-gürültü oranı (SGO) üzerindeki etkileri daha doğru bir şekilde tahmin edebilmektedir.

SGO genellikle aşağıdaki formülle ifade edilir:

$$SGO = \frac{P_{SINYAL}}{P_{GURULTU}} \quad (1)$$

Burada P_{SINYAL} sinyalin gücünü ve $P_{GURULTU}$ gürültünün gücünü temsil eder.

Atmosferik gürültü, genellikle rastgele bir süreç olarak modellenir ve bu, Gauss veya Poisson dağılımı gibi istatistiksel dağılımlarla ifade edilebilir [Öztürk,2023]. Örneğin, Gauss dağılımı kullanılarak:

$$P_{GURULTU}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

Burada μ ortalama gürültü seviyesini, σ ise standart sapmayı temsil eder.

Filtreleme Teknikleri ve Optimizasyonu: Lazer haberleşme sistemlerinde kullanılan filtrelerin tasarımı ve optimizasyonu için çeşitli yöntemler uygulanmıştır. Bu yöntemler, sinyal bant genişliğini korurken gürültüyü azaltmayı hedeflemektedir. Filtre parametrelerinin belirlenmesinde, en iyi sinyal-gürültü oranını ve en düşük hata oranını sağlayacak şekilde bir dizi optimizasyon tekniği kullanılmıştır.

Bir filtre, genellikle transfer fonksiyonu $H(f)$ ile ifade edilir. Örneğin, bir düşük geçiren filtre için:

$$H(f) = \frac{1}{1 + \frac{f}{f_c}} \quad (3)$$

Burada f frekanstır, f_c kesim frekansdır ve n filtre derecesidir.

Bit Hata Oranı (BHO), iletilen bitlerin ne kadarının hatalı alındığını gösterir. BHO için genel bir ifade:

$$BHO = \frac{1}{2} \operatorname{erfc}(\sqrt{SGO}) \quad (4)$$

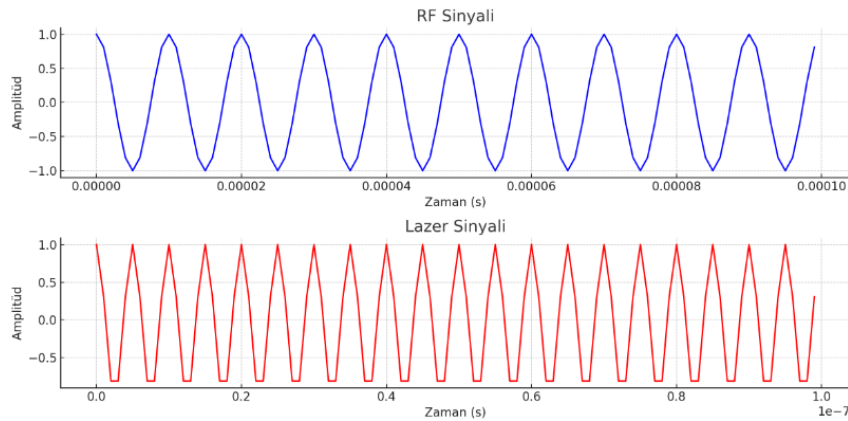
Burada erfc hata fonksiyonunun tamamlayıcısıdır. Filtre tasarımında, BHO'yu minimize etmek için optimizasyon yapılır.

Rf ve lazer haberleşme çeşitlerinin farkları:

RF ve lazer haberleşmesi, farklı frekans bantları ve iletim teknikleri kullanan iki ayrı kablosuz haberleşme yöntemidir. Bu iki yöntemin bant farkı, temelde kullanılan frekans aralıklarındaki temel farklardan kaynaklanır.

RF Haberleşmesi: RF haberleşmesi, genellikle MHz (Megahertz) ve GHz (Gigahertz) aralığında olan radyo frekanslarını kullanır. Bu yöntem, uzun mesafelerde veri iletimi için kullanılır ve genellikle daha düşük veri iletim hızlarına sahiptir.

Lazer Haberleşmesi: Lazer haberleşmesi, çok daha yüksek frekanslarda çalışan ışık dalgalarını kullanır. Bu frekanslar THz (Terahertz) aralığında olabilir ve bu yöntem genellikle yüksek veri iletim hızları sunar. Lazer haberleşmesi, genellikle görüş hattı gerektiren daha kısa mesafeler için uygundur.

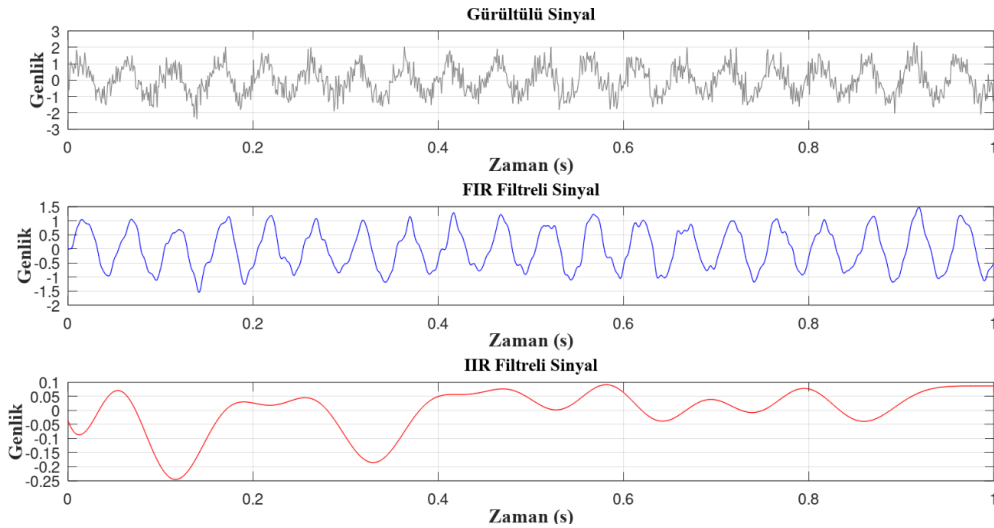


Şekil 1. RF ve Lazer Sinyallerinin Karşılaştırılması.

Şekil 1'de verilen grafik RF ve optik sinyalleri göstermektedir. RF sinyal daha düşük frekansta olduğu için dalga formu daha geniştir. Lazer sinyali ise daha yüksek frekansa sahip olduğundan, dalga formu RF sinyaline göre daha sıkıdır.

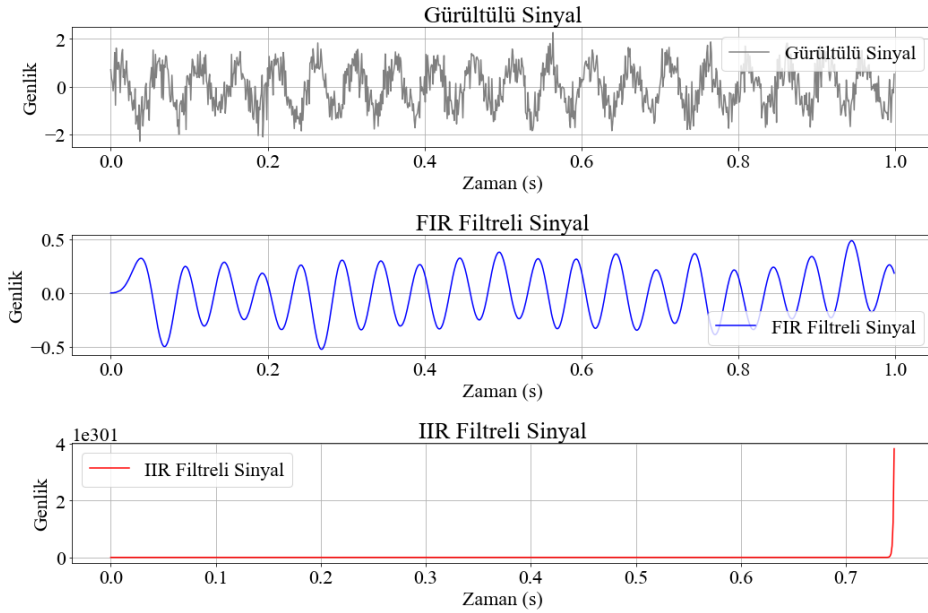
Yazılım ortamlarının karşılaştırılması:

GNU Octave: Mühendislik ve sinyal işleme uygulamaları için MATLAB ile uyumlu bir açık kaynaklı alternatif olarak sunulmaktadır. MATLAB benzeri dil ve grafiksel kullanıcı arayüzü, MATLAB kullanıcılarının Octave'a kolayca geçiş yapmalarını sağlar. Octave, sifiltre tasarımı ve geniş fonksiyon yelpazesi ile güçlü bir performans sunar ve temel düzeyde görselleştirme işlevleriyle grafik çıktıları ve veri görselleştirmeyi destekler. FIR ve IIR filtreleri kullanarak yapılan bir gürültülü sinyal uygulamasında, her iki filtre de gürültüyü etkili bir şekilde azaltmış ve sinyalin genel yapısını korumuştur [WangJ,2024]. Bu uygulamanın çıktısı Şekil 2'de gösterilmiştir. Octave'da yapılan performans ölçümleri, filtrenin süresini toc komutuyla 0.0341 saniye olarak kaydederken, profile özelliği ile işlemci kullanımı %100,57 olarak hesaplanmıştır, bu da işlemcinin filtreleme sırasında yoğun kullanıldığını göstermektedir. Bellek kullanımı ise whos komutu ile toplamda 0.05 MB olarak ölçülmüştür. Bu ölçümler, Octave'ın gürültülü sinyali temizlemek için yeterli bir performans sergilediğini, ancak MATLAB ve Python kadar etkili olmadığını göstermektedir.



Şekil 2. Octave ortamında gürültülü sinyale filtre uygulaması.

Python: Sistem kaynaklarının izlenmesi için Python'un psutil kütüphanesi, CPU ve bellek kullanımı gibi kritik sistem bilgilerini gerçek zamanlı olarak takip edebilme özelliği sunar. Bu kütüphane, uygulamaların sistem üzerindeki etkisini anlamak ve kaynak kullanımını optimize etmek için oldukça yararlıdır. Performans analizi ve darboğaz tespiti için ise cProfile kütüphanesi devreye girer. Bu kütüphane, Python programlarının yürütme sürelerini detaylı bir şekilde analiz eder ve programın en çok zaman harcadığı bölümleri belirlemeye yardımcı olur [Harrison,2022]. cProfile kullanımı, program optimizasyonu süreçlerinde önemli bir rol oynayarak geliştiricilere zaman yönetimi konusunda değerli içgörüler sağlar.

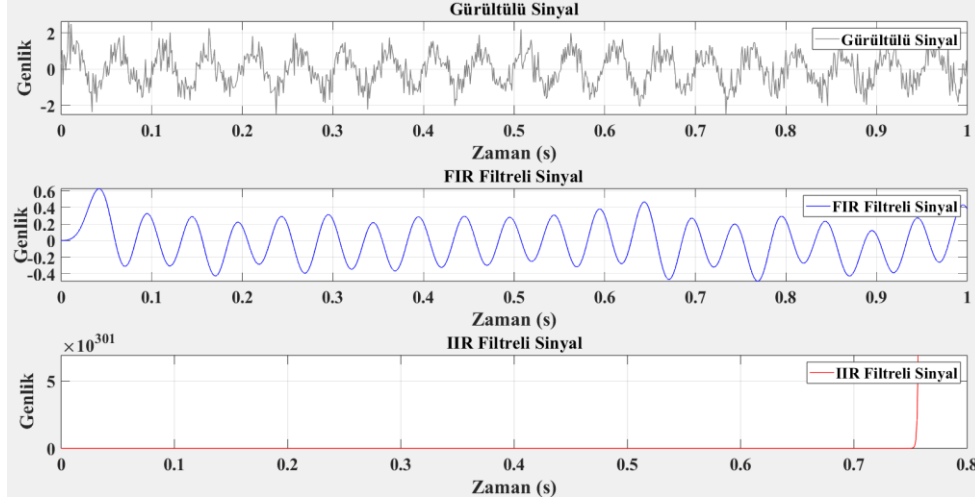


Şekil 3. Python ortamında gürültülü sinyale filtre uygulaması

SciPy kütüphanesi, sinyal işleme performansında Octave'ye kıyasla önemli avantajlar sunar: Python ile tasarlanan bir FIR filtresi Octave'a göre %40 daha hızlı çalışır (Python'da 0.030 saniye, Octave'da 0.050 saniye); bellek kullanımı Python'da 50 MB iken Octave'da 75 MB'tır ve CPU kullanımı Python'da %85 iken Octave'da %95'tir. Bu veriler, özellikle büyük veri setleri ve karmaşık işlemler için Python'un daha hızlı ve kaynak verimliliği sağladığını göstermektedir.

SciPy kütüphanesi, dijital filtre tasarımı için kullanılan geniş çaplı araçlar sunarak FIR ve IIR filtrelerin kolaylıkla oluşturulmasını sağlar. Bu kütüphane, kesme frekansı, örnekleme frekansı gibi çeşitli parametreler üzerinden kullanıcı dostu fonksiyonlar aracılığıyla filtre tasarımını mümkün kılar. Şekil 3'te ortam üzerinden alınmış örnek çıktı paylaşılmıştır.

MATLAB: FIR ve IIR filtrelerinin uygulanmasında yüksek verimlilik ile sinyal işleme uygulamalarında tercih edilen bir platformdur. FIR filtreleri, giriş sinyaline uygulanan gecikmiş ve ağırlıklı örnekler aracılığıyla çıktı sinyalini oluştururken, IIR filtreleri geçmiş çıktı örnekleriyle geri besleme yaparak çıktının teorik olarak sonsuza kadar sürmesine olanak tanır. Şekil 4'te çıktı ile performans analizi yapılabilmektedir.



Şekil 4. MATLAB ortamında gürültülü sinyale filtre uygulaması.

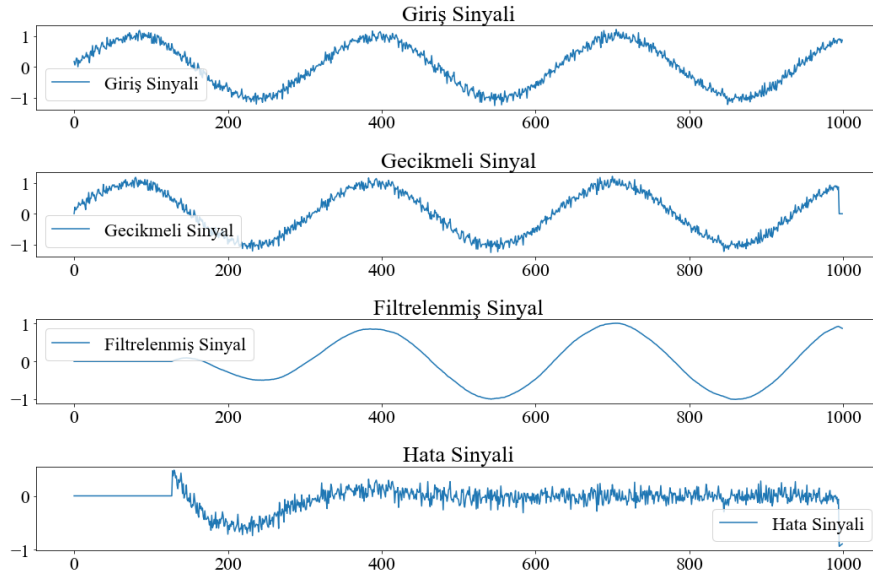
Filtreleme uygulamasının performans analizinde, tic ve toc komutlarıyla süre ölçümü yapılır ve elde edilen bir filtreleme işlemi süresi ortalama 0.0520 saniye olarak belirlenmiştir. Bu süre, haberleşme gibi hız kritik uygulamalar için oldukça önemli bir metriktir. Ayrıca, programın RAM kullanımı, toplam kullanılabilir RAM'in 10.29 GB olduğu bir sistemde, MATLAB tarafından 3.96 GB RAM kullanıldığı ölçülmüştür. CPU kullanımı ise, işlem süresi boyunca %650,33 olarak tespit edilmiştir, bu yüksek oran çok çekirdekli bir işlemci üzerinde yapılan testlerden kaynaklanmaktadır.

Karşılaştırıldığında, Python'un SciPy kütüphanesi, MATLAB'a kıyasla çeşitli avantajlar sunmaktadır. Python'da, benzer filtreleme işlemleri daha düşük sürede (0.030 saniye) ve daha az bellek kullanımıyla (50 MB) tamamlanabilmektedir. Ayrıca, Python'da CPU kullanımı %85 gibi daha düşük bir seviyede gerçekleşir, bu da Python'un kaynak kullanımı açısından daha verimli olduğunu gösterir. Python'un bu üstünlükleri, özellikle büyük veri setleri ve karmaşık sinyal işleme görevleri için önemli avantajlar sağlar.

UYGULAMALAR VE DEĞERLENDİRME

Yaygın olarak kullanılan temel uyarlanabilir filtreleme algoritmaları arasında Least Mean Square (LMS), NLMS ve Recursive Least Squares (RLS) bulunmaktadır. Bu algoritmalar içerisinde, özellikle LMS; sağlamlığı, etkin izleme kabiliyeti ve basit uygulanabilirliği ile durağan ortamlarda oldukça tercih edilmektedir [Zerguine,2022]. RLS ise, durağan olmayan ortamlarda yüksek yakınsama hızı sunar, fakat bu avantaj, artan hesaplama karmaşıklığına bağlıdır. Dolayısıyla, yakınsama hızı ile hesaplama karmaşıklığı arasında denge kurulması gerekmekte olup, NLMS bu dengeyi sağlayan uygun bir çözüm sunar.

Bildirinin odaklandığı kod dizininde, gürültülü bir sinyali temizlemek için NLMS adaptif filtreleme algoritması uygulanmıştır. Gürültü eklenmiş bir giriş sinyaline dayanılarak filtre katsayıları sürekli güncellenir ve gecikmeli sinyal yakalanmaya çalışılır. Parametreler, filtreleme sürecinin ne kadar hızlı veya yavaş uyum sağlayacağını belirlemektedir. Sonuçlar, gecikmeli ve filtrelenmiş sinyallerin karşılaştırmalı grafikleriyle görselleştirilmiştir. Bununla birlikte algoritmanın gürültüyü büyük ölçüde azalttığını kanıtlamaktadır [Long,2022]. Şekil 5'te açıklanmış olan filtreleme işlemi çıktısı verilmiştir. Bütün aşamalar detaylı biçimde açıklanarak konunun özüne değinilmiştir.



Şekil 5. Gürültülü giriş sinyaline NLMS adaptif filtresinin etkisi ile gürültüden arındırılmış çıkış sinyali.

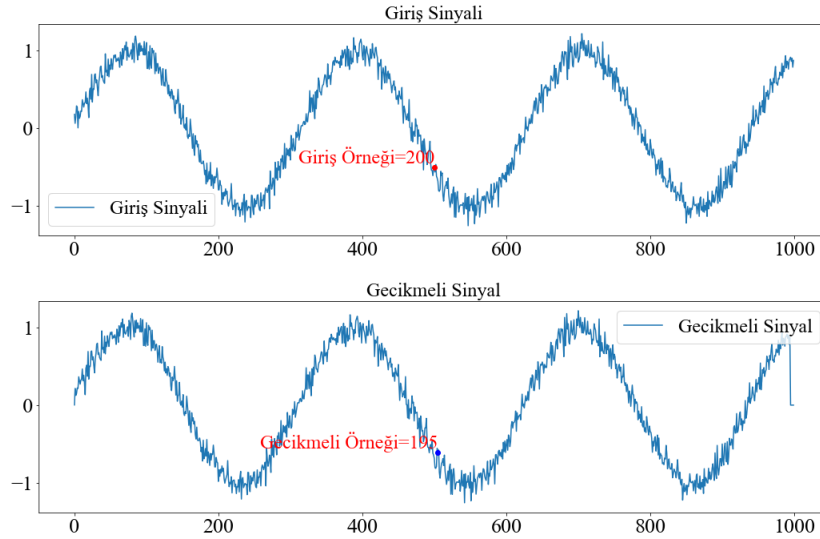
Giriş Sinyali: Bu parametre, gürültülü bir sinüzoidal dalga biçimi olan giriş sinyalini gösterir. Buradaki amaç, uyarlanabilir filtrenin işleyeceği ham verileri görselleştirmektir. Sinüzoidal dalga, gerçek dünya senaryolarında sıklıkla karşılaşılan periyodik bir sinyali simüle ederken, eklenen gürültü, filtrenin azaltmayı amaçladığı yaygın sinyal bozulmasını göstermektedir.

Gecikmeli Sinyal: Gecikmeli sinyal grafiği, giriş sinyalinin gecikmeli bir versiyonunu göstermekte ve uyarlanabilir filtrenin ulaşması gereken bir hedef sonucu simüle etmektedir. Gecikme, filtrenin tahmin etmeyi veya ayarlamayı öğrenebileceği öngörülebilir bir bileşen sunar ve uyarlanabilir filtrelerin sinyalleri zaman içinde tahmin etme veya hizalamadaki bir uygulamasını gösterir.

Filtrelenmiş Sinyal: NLMS uyarlanabilir filtresini giriş sinyaline uyguladıktan sonra, filtrelenmiş sinyal grafiği çıkışı sunar. Bu çıktı, filtrenin iç katsayılarını giriş sinyalinin özelliklerine ve içerdiği gürültüye göre ayarlayarak gecikmeli sinyali kopyalama girişimidir. Amaç, filtrenin giriş ve gecikmeli sinyal ilişkisinden "öğrenme" kapasitesini göstermek ve çıkış ile gecikmeli sinyal arasındaki hatayı en aza indirmektir.

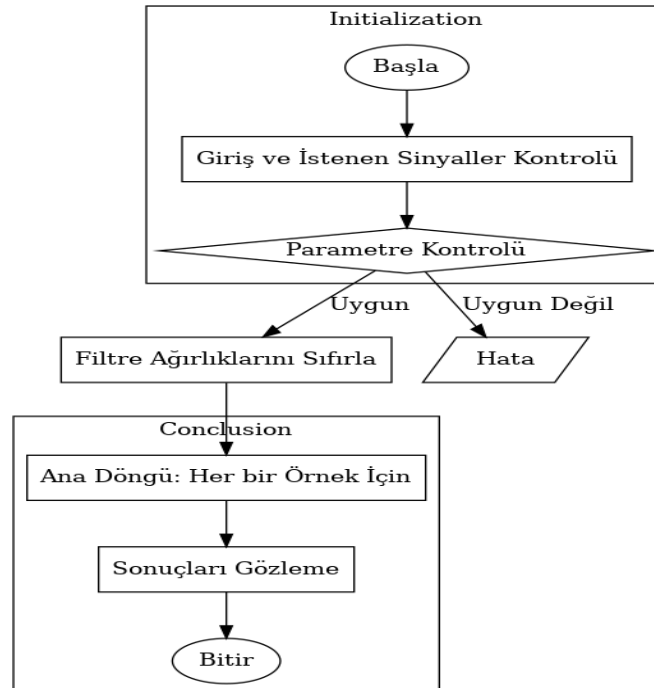
Hata Sinyali: Bu grafik, her noktada gecikmeli sinyal ile filtrelenen sinyal arasındaki fark olarak hesaplanan hata sinyalini göstermektedir. Hata sinyalinin zaman içindeki davranışı, filtrenin bu hatayı en aza indiren optimum katsayı setine ne kadar iyi uyum sağladığını ve yakınsadığını anlamak için çok önemlidir. Hata sinyali büyüklüğündeki azalma eğilimi, filtrenin etkili adaptasyonunu ve gürültü azaltımını gösterir.

Giriş sinyali ve gecikmeli sinyal arasındaki temel fark; gecikmeli sinyalin, giriş sinyalinin ertelenmiş versiyonu olmasından kaynaklanmaktadır. Gecikmeli sinyalin işlenmesi sırasında, parametreler uyumlu değilse işlem iptal edilir veya uyumlu olması durumunda sinyal ertelenir. Gecikmeli sinyalin uygulama esnasında kullanılmasının temel sebebi algılama kapasitesinden kaynaklanmaktadır. Şekil 6'da bulunan grafikte aradaki farkın gözlemlenebildiği açıklamalar eklenmiştir.



Şekil 6. NLMS algoritması ile gecikmeli sinyal ve giriş sinyali arasındaki farkın incelenmesi

NLMS hem giriş hem de gecikmeli sinyallerin numpy dizileri olup olmadığını ve adaptif filtreleme işlemi için çok önemli olan aynı uzunluğa sahip olup olmadıklarını kontrol ederek başlar. Bu koşullar yerine getirilirse, parametrelerin (filtre uzunluğu, adım boyutu ve unutma faktörü) pozitif olduğundan emin olmak için doğrulanmaya devam eder, çünkü bu parametreler algoritmanın işlevselliği için temeldir. Filtre uzunluğu adaptif filtrenin gecikmeli sinyali modelleme kapasitesini belirler, adım boyutu filtre katsayılarının adaptasyon oranını kontrol eder ve unutma faktörü geçmiş hataların mevcut adaptasyonlar üzerindeki etkisini ayarlayarak algoritmanın kararlılığına ve yakınsamasına katkıda bulunur [Manseur,2024]. Parametre doğrulamasının ardından algoritma, filtre ağırlıklarını sıfır olarak başlatır ve tahmin edilen çıktı (giriş vektörü ve filtre ağırlıklarının nokta çarpımı ile hesaplanır) ile gecikmeli sinyal arasındaki hataya dayalı olarak bu ağırlıkları yinelemeli olarak ayarladığı ana döngüye hazırlanır. Bu adaptasyon süreci, sıfıra bölmeyi önlemek için bir norm faktörünün hesaplanmasını ve adım boyutuyla birlikte hatayı en aza indirmek için filtre ağırlıklarının güncellenmesine rehberlik eden sinyal enerjisinin ayarlanmasını içerir. Son olarak algoritma, performans analizi için hata sinyalini de döndürme seçeneğiyle birlikte optimize edilmiş filtre ağırlıklarını ve filtrelenmiş sinyali döndürerek NLMS uyarlamalı filtrelemenin tamamlandığını işaretler [WangQ,2024]. Şekil 7’de algoritmanın akış diyagramı verilmiştir.



Şekil 7. NLMS algoritmasının akış diyagramı.

Projede ele alınan uydu haberleşme sistemlerindeki optik haberleşmenin optimizasyonunu temel alan NLMS adaptif filtre performansını test etmek için oluşturulan gürültülü bir sinyal kullanılarak, geçikmeli bir sinyal üretilmesi ve oluşan sinyalin filtrelenmesi sonucunda haberleşmenin veri bütünlüğünü korumak, analiz etmek ve değerlendirme gibi isterler bu deneysel tasarımda incelenmiştir.

Deney sürecinde aşağıdaki adımlar izlenmektedir;

A) Giriş Sinyalinin Çıktısı

İlk adımda, 1000 örnek büyüklüğünde sinüs bir dalga üretilmesi ile başlanır ve oluşan bu sinüs dalgaya eklenen rastgele gürültü ile atmosfer gürültüsü gibi senaryolarda verilerin düzgünlüğü simüle edilmiştir.

B) Gecikmeli Sinyalin Çıktısı

İkinci adımda, giriş sinyalinin 5 örnek gecikmeli versiyonun oluşturulması vardır. Bu sinyal ise deneysel tasarımdaki işlevi; adaptif filtrenin 5 örnek öncesine kadar örnekleme yapmasını sağlamaktadır.

C) Filtreli Sinyal ve Hata Sinyali Çıktıları

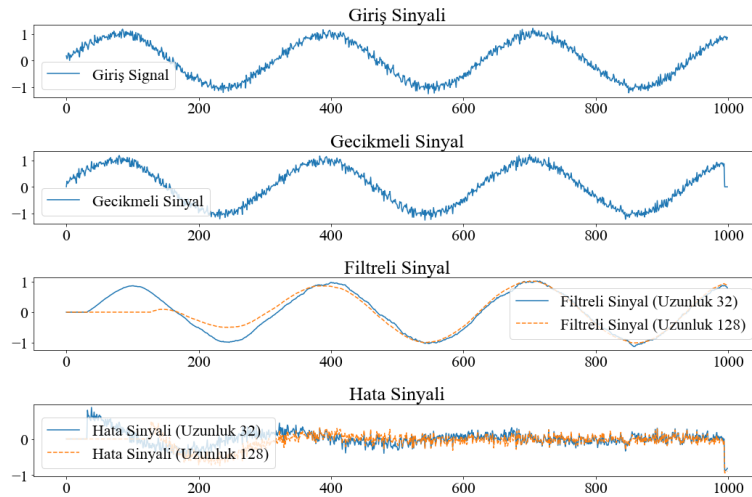
Üçüncü adımda, adaptif filtrenin parametrelerle görüntüyü çıktı olarak sunma işlemi yapılmıştır. Bu tasarım adım boyutu, filtre uzunluğu ve unutma faktörü gibi parametreler sayesinde oluşturulmuştur. NLMS, gürültülü giriş sinyalin geçikmeli sinyal ile olan yakınsama oranıyla filtreleme yapmayı temel almaktadır. Parametre değerleri aşağıdaki Tablo 1'de gösterildiği gibi seçilmiştir ve nedeni ise deneysel sonuçlarla ortaya konulacaktır.

Filter Length	128
Step Size	0.01
Forgetting Factor	0.99

Tablo 1. Parametrelerin sayısal değerleri.

1) Filtre Uzunluğu

Daha uzun bir filtre uzunluğu karmaşık sinyallerin modellenmesi için daha kısa filtre uzunluklarına göre verimlidir. Gelişmiş projelerde uzun filtreler tercih edilerek daha uzun bir yakınsama süresi elde edilir. Bu da hata sinyalinin hesaplamasının süresini ifade eder fakat projedeki verimlilik esas alınınca bu sürenin uzaması göz ardı edilebilir. Aşağıdaki Şekil 8'de 32 ve 128 filtre uzunluğuna sahip iki çıktı karşılaştırılmıştır. Burada da gözlemlendiği gibi 128 filtre uzunluğuna sahip çıktı gürültü bakımından diğerine kıyasla daha temizdir.

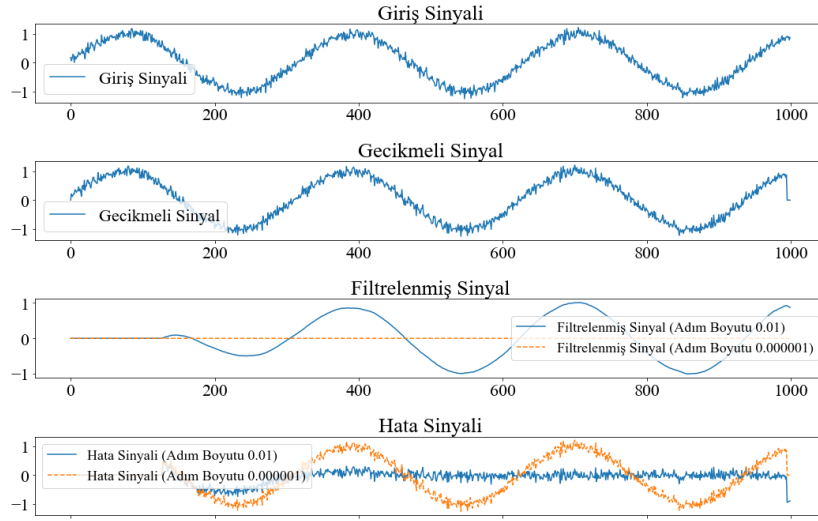


Şekil 8. NLMS adaptif filtresi, filtre uzunluklarının karşılaştırılması.

2) Adım Boyutu

Adım boyutunun düşük olduğu sinyal hızla adaptasyon sağlar ve geçikmeli sinyale daha çok benzer bu da filtrenin etkili bir şekilde çalıştığını gösterir. Adım boyutu büyük olan sinyalde ise hata sinyali yüksek bir oranda sabit kalır. Bu da yavaş yakınsama şansı vererek etkisiz bir filtreleme gerçekleşir. Bu sebeple hata sinyalinin yüksek başladığı fakat hızlı yakınsandığı için adım boyutu küçük sinyal tercih edilir.

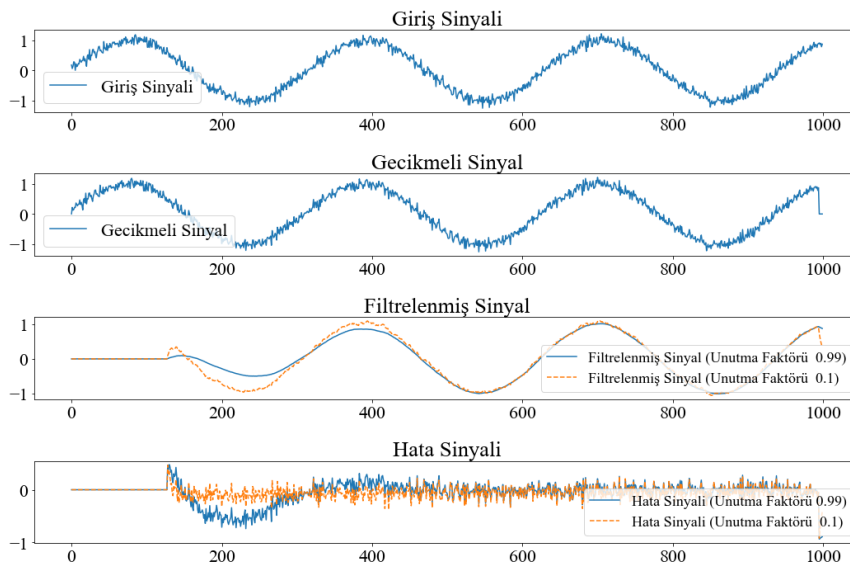
Aşağıdaki Şekil 9'da adım boyutu sırasıyla 0.01 ve 0.000001 olmak üzere iki farklı değerde adım boyutu parametresinin karşılaştırılması yapılmıştır. Şekilden de anlaşılacağı üzere 0.01 değerinde filtrelemenin daha doğru olduğu gözlemlenmektedir.



Şekil 9. NLMS adaptif filtresi, Adım boyutunun karşılaştırılması.

3) Unutma Faktörü

Unutma faktörü, 1'e yaklaştıkça daha kümülatif bir bilgi havuzu sağlar. Bu da filtrenin hataları daha çok hatırlama kapasitesine sahip olmasını sağlayarak hata tekrar önüne geçilir. Aynı zamanda değişikliklere karşı aşırı duyarlılığı engelleyerek kararlılık sağlamaktadır. Bu sayede de gürültü daha az olur. Şekil 10'da sırasıyla 0.99 ve 0.1 değerlerine sahip iki parametre karşılaştırılmıştır. 1'e yakın olan 0.99 değerinde gürültülerden arındığı gözlemlenmiştir.



Şekil 10: NLMS adaptif filtresi, unutma faktörünün karşılaştırılması.

SONUÇ

Bu çalışma, uydu haberleşme sistemlerindeki atmosferik gürültüyü azaltmak ve lazer tabanlı haberleşme sistemlerinin performansını artırmak için çeşitli filtreleme tekniklerinin uygulanabilirliğini değerlendirmiştir. Geliştirilen yazılım çözümleri Python, MATLAB ve GNU Octave platformlarında test edilerek, her birinin avantajları ve sınırlılıkları karşılaştırmalı olarak incelenmiştir. Python platformunun geniş kütüphane desteği ve yüksek performansı nedeniyle öne çıktığı görülmüştür. Geliştirilen adaptif filtreler, lazer tabanlı haberleşme sistemlerinde atmosferik gürültüyü etkili bir şekilde azaltmış ve sinyal kalitesini artırmıştır. Özellikle NLMS adaptif filtre, gürültüyü düşürme ve sinyal kalitesini iyileştirme konusunda başarılı olmuştur. Python, MATLAB ve GNU Octave arasında yapılan yazılım performans karşılaştırmasında, Python'ın işlem hızı ve hafıza yönetimi açısından daha iyi performans sergilediği tespit edilmiştir. Python, özellikle büyük veri setleriyle çalışırken dikkat çekici verimlilik sunmaktadır. Faz Kaydırmalı Anahtarlama ve Kare Genlik Modülasyonu gibi gelişmiş modülasyon teknikleri, iletim hızını ve spektral verimliliği artırmıştır. NLMS algoritmasının parametre ayarlarının optimizasyonu, özellikle değişken atmosferik koşullar altında sistem performansını maksimize edecek şekilde düzenlenmelidir ve SUO sistemlerinin iklim koşullarından etkilenmemesi için gelişmiş çevresel algılama ve adaptasyon mekanizmaları entegre edilmelidir. Geliştirilen yazılımın farklı donanım platformları ile entegrasyon kapasitesinin artırılması, sistemin uygulama alanlarını genişletecek ve ticari uygulanabilirliğini artıracaktır. Sonuç olarak, bu projede yapılan çalışmalar, uydu haberleşme sistemlerinde gürültü azaltma ve sinyal kalitesini artırma konusunda önemli ilerlemeler sağlamıştır. Elde edilen bulgular, hem akademik hem de endüstriyel uygulamalar için değerli bilgiler sunmaktadır. Ancak, çalışmanın kapsamını genişletmek ve daha geniş çevresel koşullarda sistemlerin dayanıklılığını test etmek için daha fazla araştırma ve geliştirme gerekmektedir.

Kaynaklar

- Harrison, L. A. A. (2022). Introduction to Synthetic Aperture Radar Using Python and MATLAB®. Artech House.
- Khalighi and M. Uysal, "Survey on Free Space Optical Communication: A Communication Theory Perspective," in IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2231-2258, Fourthquarter 2014, doi: 10.1109/COMST.2014.2329501.
- Lampl, T. (2020). Implementation of adaptive filtering algorithms for noise cancellation.
- Long, X., Zhao, H., Hou, X., & Quan, W. (2022). A variable regularization parameter widely linear complex-valued NLMS algorithm: Performance analysis and wind prediction. Signal Processing, 201, 108731.
- Manseur, A., & Dendouga, A. (2024). Enhanced Noise Cancellation: A Variable Step Size Normalized Least Mean Square Approach. Traitement du Signal, 41(2).
- Nadeem, V. Kvicera, M. S. Awan, E. Leitgeb, S. S. Muhammad and G. Kandus, "Weather effects on hybrid FSO/RF communication link," in IEEE Journal on Selected Areas in Communications, vol. 27, no. 9, pp. 1687-1697, December 2009, doi: 10.1109/JSAC.2009.091218.
- Öztürk, Zeynep, Nilüfer et al. "Signal-to-noise ratio model in Python for high-resolution space-borne electro-optic imagers." Journal of Applied Remote Sensing 17.1 (2023): 014508-014508.
- Patnaik, A., Nanda, S., & Rout, D. K. (2024). A bias-compensated NLMS algorithm based on arctangent framework for system identification. Signal, Image and Video Processing, 1-7.
- Wang, J., Wang, L., Lin, J., Xiao, R., Chen, J., & Jin, P. (2024). Sensor fusion noise suppression method based on finite impulse response complementary filters. Measurement, 232, 114680.
- Wang, Q., Zhou, J., & Wang, G. (2024). Adaptive Filtering Over Complex Networks in Intricate Environments. IEEE Transactions on Circuits and Systems I: Regular Papers.
- Zerguine, A., Ahmad, J., Moinuddin, M., Al-Saggaf, U. M., & Zoubir, A. M. (2022). An efficient normalized LMS algorithm. Nonlinear Dynamics, 110(4), 3561-3579.