

Demodülasyon Algoritmaları için En İyilenmiş Windows İşletim Sistemi Uygulamaları

Alparslan FİŞNE*

Aselsan A.Ş. Rehis Sektör Başkanlığı, 06172, Ankara

(Alınış / Received: 22.05.2016, Kabul / Accepted: 12.08.2016
Online Yayınlama / Published Online: 09.01.2017)

Anahtar Kelimeler
demodülasyon,
en iyileme,
gerçek zamanlı
ses işleme

Özet: Telsiz yayınlarından seslerin elde edilmesi için yazılım tabanlı almaçlar kullanılmaktadır. Almaçtan elde edilen sesler aynı zamanda kaydedilmiş ham verilerden çevrimdışı olarak elde edilebilmektedir. Dolayısıyla veri kayıtları üzerinde JAVA tabanlı Kullanıcı Arayüzü'nde çevrimdışı olarak demodülasyon işlemleri yapılmaktadır. Gerçek zamanlı ses işleme için demodülasyon algoritma işlem hızının ses çaldırma hızından yüksek olması gereklidir. Bu çalışmada, gerçek zamanlı ses işleme için Intel işlemci mimarisi tabanlı optimizasyonlardan bahsedilmektedir. JNI yöntemi kullanılarak yapılan optimizasyonlar sayesinde JAVA tabanlı yazılımda demodülasyon işlemleri için 10 kat hızlanma gözlemlenmiştir. Optimizasyonlar sayesinde Kullanıcı Arayüzü'nde demodülasyon işlemleri yüksek bant genişliğine sahip sinyaller için gerçek zamanlı olarak yapılmaktadır.

Optimized Windows Operating System Applications for Demodulation Algorithms

Keywords
demodulation,
optimization,
real time
audio processing.

Abstract: The software defined receivers are used for deriving of audio signals from radio broadcasting. Audio signals that are acquired from receiver are also obtained as offline from recorded raw data. Therefore, the demodulation algorithm processes can be made offline in JAVA based User Interface(UI). For real time audio processing, it is necessary that demodulation process rate is higher than audio playback rate. In this study, the Intel processor architecture based optimizations are mentioned for real time audio processing. It is observed ten times speed up due to optimizations using JNI method at JAVA based software for demodulation algorithms. The demodulation processes can be made real time due to optimizations for high bandwidth signals, in User Interface.

*Sorumlu yazar: afisne@aselsan.com.tr

1. Giriş

Yazılım tabanlı almaçlar sayesinde herhangi bir elektromanyetik bantta yayınlar tespit edilip her bir yayın için sinyal işleme işlemleri yapılabilmektedir [1]. Almaçlar sayesinde anlık olarak ses üretimi yapıp ortamındaki yayınlardan haberdar olunması sağlanmaktadır. Anlık olarak analiz edilen sinyallerin yanı sıra daha sonrası analizi kritik olacak yayınlar için dar bant veri kayıtları alınmaktadır. Kayıtlar üzerinde yayınlara ait ses üretimi gibi işlemlerin yapılması için kullanıcıya bir yazılım sunulması gereklidir. Intel bilgisayarlarda Windows İşletim Sistemi üzerinde kullanıcıya Kullanıcı Arayüz Yazılımı sunulur [2]. Kullanıcılar kayıtları incelerken sesleri kesintisiz olarak dinlemek istemektedir. Kesintisiz ses üretmek için sesin üretilme hızı sesin çaldırma hızından yüksek olma gereği ses üretme işinin gerçek zamanlı olmasını zorunlu kılmaktadır [3]. Gerçek zamanlı bir işlem yapıldığında işlem performansı ile ilgili yaşanan darboğazları aşmak için yazılım ve donanım seviyesinde optimizasyonlar gerekmektedir.

Demodülasyon işlemlerinde sinyal işleme adımları uzun süreden Intel işlemci mimarisi tabanlı Tek İşlem Çok Veri (SIMD) mantığına dayanan optimizasyon yapılmaktadır [4]. Optimizasyon içeren yazılımlar C/C++ programlama dilleri ile yazıldığı için JAVA programlama dili ile yazılan Kullanıcı Arayüz Yazılımı'nda optimizasyon içeren işlemlere ait yazılım kütüphaneleri kullanılır. Başka bir programlama dili ile yazılmış yazılım kütüphanelerini JAVA programlama dilinde kullanmak için JNI (Java Native Interface) adlı arayüz kullanılır [5]. Bu makalede, JNI arayüzü ile optimize edilmiş bileşenleri içeren Kullanıcı Arayüz Yazılımı'ndaki demodülasyon işlemlerinden bahsedilmektedir. Bu çalışma 5 ana bölümden oluşmaktadır.

Çalışmanın ikinci bölümünde analog demodülasyon algoritmalarından, üçüncü bölümde yapılan optimizasyonlar anlatılmaktadır. Dördüncü bölümde JNI arayüz alt yapısından bahsedilecek olup optimizasyon sonucunda elde edilen işlem süreleri karşılaştırılmaktadır. Son bölüm olan beşinci bölümde ise yapılan çalışmanın sonuçları ile ilgili bilgiler aktarılmaktadır.

2. Demodülasyon Algoritmaları

Demodülasyon işlemleri analog ve sayısal modülasyonlu yayınlarındaki seslerin üretilmesi için kullanılmaktadır. Analog demodülasyon işlemlerin karmaşıklık düzeyinin az olması sebebiyle çalışmamızda analog demodülasyon algoritmaları incelenmiştir. Çevrimdışı bir zamanda dar bant veri kayıtları incelenerek analog demodülasyon işlemleri yapılabilmektedir.

2.1. AM Demodülasyon Algoritması

Verici birimlerde mesajlar bir taşıyıcı sayesinde gönderilmektedir. AM demodülasyon işleminde dar bant sinyalin genliği alınarak çözülmüş mesaj sinyali elde edilir [6].

2.2. FM Demodülasyon Algoritması

Frekans modülasyonu taşıyıcı dalga frekansının, bilgi sinyalinin frekans ve genliğine bağlı olarak değiştirilmesidir. Günümüzde, ticari amaçla yayın yapan FM vericiler 87,5 - 108 MHz arasında yayın yaparlar [6].

2.3. CW Demodülasyon Algoritması

Genellikle askeri amaçlı haberleşmede kullanılan CW (sürekli dalga) telsizlerden yapılan yayınları analiz etme ve yayınlardan ses üretmek için CW Demodülasyon Algoritması kullanılır [6].

Genel olarak analog demodülasyon işlemleri dar bant veri üzerinde yapılır. İşlemler sonucunda sayısal ses verisi

elde edilir. Elde edilen sayısal ses verisi uygun formata dönüştürüldükten sonra ses kartına yollanarak çalınması sağlanır. Ses kartına yollanmadan önce seslerin çalınması için sayısal ses verisinin örnekleme hızı ayarlanır. Ses kartına gönderilen sayısal ses verilerinin üretilme işi için yapılan işlemler almaç ve çevrimdışı kayıt analiz birimlerinde benzerlik göstermektedir. Şekil 1'de demodülasyon işlemi ve ses üretimine ait bir işlem akış şeması bulunmaktadır. Anlık olarak üretilen ses verisi için almaç mimarilerinde optimizasyon yapılır. Optimizasyonların yapılması sebebi gerçek zamanlı bir işlemin bulunmasıdır. Kullanıcı, yayılara ait sesleri kesintisiz olarak dinlemek istemektedir. Sesin çaldırılma hızı üretilme hızından büyük olduğunda ses kesintileri olmaktadır. Almaçlarda yapılan optimizasyonların çevrimdışı demodülasyon işlemlerinde de yapılması gerekmektedir.

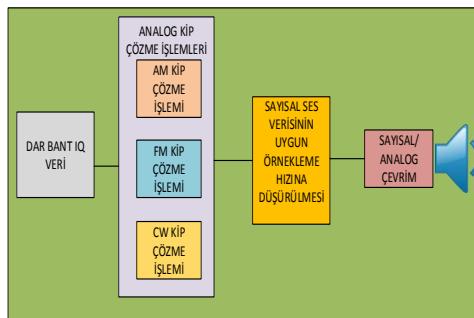
kullanılarak kodlanmaktadır. Analiz sonuçlarının gönderileceği ve veri kayıtlarının yönetildiği bir Kullanıcı Arayüz Yazılımı'na ihtiyaç olmaktadır. Sonuçlar Kullanıcı Arayüz Yazılımı'nda gösterilecek olup seslerin çaldırılması Kullanıcı Arayüz Yazılımı aracılığıyla yapılmaktadır. Optimizasyonlar vektör işlemlerinde üzerinde yapılmaktadır. İşlemcinin özelleşmiş mimarilerini kullanmak optimizasyon başarısına etki etmektedir.

3.1. Intel Vektör İşleme Mimarileri

Intel 1999 yılında Pentium III serisi ile vektör işlemlerinde performans sağlamak için geliştirdiğini mimari SSE (Streaming SIMD Extensions) mimarisini duyurdu [7]. Tek işlem çöktü veri (SIMD) mantığına dayanan mimari sayesinde 128 bit yazmaçlar geliştirilmiştir [7]. İlerleyen yıllarda bu mimarilerin gelişimi sağlanmış olup SSE2, SSE3 ve SSE4 mimarili Intel işlemciler tanıtılmıştır. 2011 yılında AVX (Advanced Vector Extension) mimarisi ile 128 bit yazmaç yerine 256 bit yazmaç kullanımı mümkün olmuştur. Intel Haswell işlemcisi ile AVX2 mimarisi tanıtılmış olup aynı anda 32 adet kayar nokta işleminin yapılmasına olanak sağlanmıştır.

3.2. Intel IPP Kütüphanesi

Intel işlemcili bilgisayarlarda ve gömülü birimlerde geliştirilen C/C++ programlama dilleri ile kodlanmış yazılımlarda sinyal işleme, matris işlemleri, görüntü işleme ve kriptoloji alanında ihtiyaç duyulan fonksiyonları barındıran IPP (Integrated Performance Primitives) yazılım kütüphanesi Intel firması tarafından hazırlanmıştır [8]. IPP Kütüphanesi aritmetik, trigonometrik ve mantıksal işlevlere ait fonksiyonları da içermektedir. Kullanıcıya bir Uygulama Programlama Arayüzü (API) sunulmuş olup üzerinde çalışan Intel işlemciye göre en uygun ve optimize şekilde işlemlerin yapılmasını



Şekil 1. Analog Demodülasyon Senaryosu

3. Optimizasyonlar

Çevrimdışı analiz ve demodülasyon işlemleri, yazılım tabanlı almaçlarda yayın varlığında kaydedilen dar bant veri kayıtları kullanılarak Windows İşletim Sistemi üzerinde geliştirilen bir yazılımla yapılmaktadır. Yaygınlaşan Intel işlemcili Windows İşletim Sistemi yükü bilgisayarlarında Çevrimdışı Analiz Yazılımı çalışmaktadır. Çevrimdışı Analiz Yazılımı optimizasyonlar barındırdığı için düşük seviyeli diller olan C/C++ programlama dilleri

sağlamaktadır. Bu çalışmada Analog Demodülasyon Algoritmaları Intel IPP kütüphanesini sağladığı fonksiyonlar kullanılarak kodlanmıştır.

3.3. Intel Derleyici Optimizasyonları

Intel firması aynı zamanda C++ programlama dili için bir derleyici tasarlamıştır. Bu derleyici Microsoft Visual Studio platformu ile entegre olabilecek şekilde çalışmakta olup optimize kod üretmesine olanak sağlamaktadır. Otomatik vektör işleme ve otomatik paralel işleme özellikleri ile işlemlerin daha optimize çalışmasını sağlamaktadır.

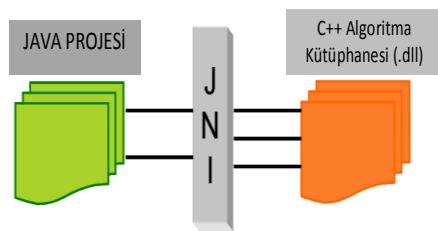
4. JNI Arayüzü ile Gerçeklenmiş Demodülasyon İşlemleri

Demodülasyon işlemleri için iki tane yazılım kullanılmaktadır. Fakat bu iki yazılım aynı birim üzerinde de çalışabilir. Bu durumda iki yazılım arasındaki arayüz kullanılarak yazılım karmaşıklığı artırmak yerine daha verimli çözümler tasarılanabilir. Çevrimdışı Analiz Yazılımı'na bağımlılık algoritmaların C++ tabanlı IPP kütüphanesi kullanımı ile ortaya çıkmaktadır. Fakat IPP kütüphanesinin JAVA ile geliştirilen Kullanıcı Arayüz Yazılımı'nda kullanılması mümkün olduğu takdirde sadece Kullanıcı Arayüz Yazılımı ile dar bant kayıtlar analiz edilip demodülasyon işlemleri yapılabılır. IPP kütüphanesinin JNI yöntemi ile JAVA platformlarda çalışıldığı Intel yazılım geliştiricileri tarafından duyurulmuştur [9].

4.1. JNI

C/C++ gibi dillerde yazılmış kodların JAVA'dan çağrılmasını sağlayan Uygulama Programlama Arayüzü'dür. JNI bazı dezavantajları barındırır. En önemli dezavantaj vektör işlemlerinde dizilerin başka bir platforma kopyalanması sebebiyle oluşan ek yük süresidir. Dolayısıyla JNI kullanılarak yapılan algoritma işlemleri için veri

transferinin az olmasına dikkat edilmesi gerekmektedir. İşlenecek bütün verinin transfer edilmesi gerekmektedir. Uygulamaya özel olarak en uygun bir veri boyu seçilerek platformlar arası veri transfer süresinin önemsiz yapılması hedeflenmelidir. JNI yönteminin kullanımı Şekil 2'deki şema ile gösterilmektedir.



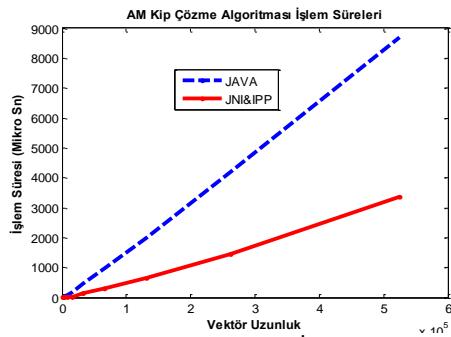
Şekil 2. JNI Arayüzü

4.2. JNI & IPP Kütüphanesi

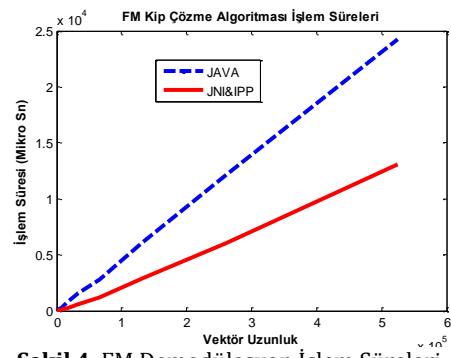
Intel IPP Kütüphanesi ile Analog Kip Çözme Algoritmaları Intel C++ derleyici kullanılarak dinamik bağlı kütüphane yaratılmaktadır. Bu kütüphane projesi JAVA Derleyici ile derlenerek JAVA Projesi'nde kullanılacak başlık dosyası oluşturulur. Böylece JAVA Projesi'ne başka bir programlama dili ile yazılmış bir fonksiyon tanıtılmış olmaktadır. Kullanıcı Arayüz Yazılımı Intel IPP kütüphanesi fonksiyonları JNI yöntemi ile kullanmakta olup algoritma senaryolarının tek başına yapabilmektedir.

4.3. Analog Demodülasyon İşlem Süreleri

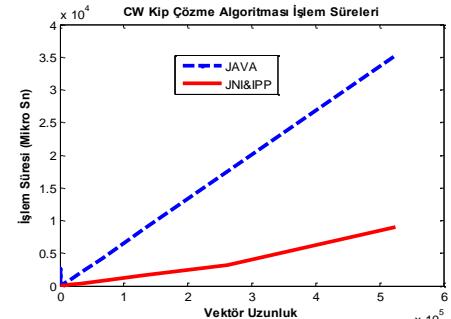
Analog Kip Çözme Algoritmaları sadece JAVA dili kullanılarak da yazılmaktadır. Fakat ses kaldırma ile ilgili gereken dolayı işlem süreleri kritik olmaktadır. Dolayısıyla optimizasyon gerekmektedir. Yapılan çalışmalarla sadece JAVA dili kullanılarak yazılmış Analog Kip Çözme Algoritmaları ile JNI ara yüzü sayesinde IPP kütüphane kullanımı ile yazılmış algoritmaların işlem süreleri karşılaştırılmaktadır.



Şekil 3. AM Demodülasyon İşlem Süreleri



Şekil 4. FM Demodülasyon İşlem Süreleri



Şekil 5. CW Demodülasyon İşlem Süreleri

5. Tartışma ve Sonuç

Yapılan çalışmada en önemli kazanç iki yazılım yerine tek yazılım kullanarak senaryoların basitleştirilmesidir. Şekil 3, Şekil 4 ve Şekil 5'e bakıldığında yapılan optimizasyonlar ile JAVA tabanlı analiz yazılımindan işlem performansı için 10 kata yakın hızlanma sağlandığı görülmektedir. JNI yöntemi ile oluşan ek yük süresi vektör boyu arttıkça önemsiz olmaktadır. IPP kütüphanesinin içерdiği alanlar düşünürse JNI yöntemi ile

Kullanıcı Arayüz Yazılımları'na esneklik ve işlem kabiliyetleri kazandırılmıştır.

Kaynakça

- [1] Cummings, M.; Haruyama, S. 1999. FPGA in the software radio, in *Communications Magazine, IEEE*, Cilt.37, No.2, s. 108-112.
- [2] Mitola, J.; Maguire, G.Q., Jr. 1999. Cognitive radio: making software radios more personal, in *Personal Communications, IEEE*, Cilt.6, No.4, s. 13-18.
- [3] Tzanetakis, G.; Cook, P. 2002. Musical genre classification of audio signals, in *Speech and Audio Processing, IEEE Transactions on*, Cilt.10, No.5, s. 293-302.
- [4] Frigo, M.; Johnson, S.G. 2005. The Design and Implementation of FFTW3, in *Proceedings of the IEEE*, Cilt.93, No.2, s. 216-231.
- [5] Sangchul Lee; Jae Wook Jeon, 2010. Evaluating performance of Android platform using native C for embedded systems, in *Control Automation and Systems (ICCAS), 2010 International Conference on*, s. 1160-1163.
- [6] Mitola, J. 1995. The software radio architecture, in *Communications Magazine, IEEE*, Cilt. 33, No. 5, s. 26-38.
- [7] Peleg, A.; Weiser, U. 1996. MMX technology extension to the Intel architecture, in *Micro, IEEE*, Cilt.16, No.4, s.42-50.
- [8] Tianyi Hong; Huabiao Qin; Qianshu Sun, 2007. An Improved Real Time Eye State Identification System in Driver Drowsiness Detection, in *Control and Automation, IEEE International Conference on*, s. 1449-1453.
- [9] Blake, A.M.; Witten, I.H.; Cree, M.J. 2013. The Fastest Fourier Transform in the South, in *Signal Processing, IEEE Transactions on*, Cilt. 61, No.19, s. 4707-4716.