**İSTANBUL TEKNİK ÜNİVERSİTESİ**

**ELEKTRİK – ELEKTRONİK FAKÜLTESİ**

# GSM AĞI ÜZERİNDEN GÜVENLİ SES İLETİMİ

**BİTİRME ÖDEVİ**

**Mehmet Akif Özkan**

**040070361**

**Bölümü: Elektronik ve Haberleşme Mühendisliği Bölümü**

**Programı: Elektronik Mühendisliği**

**Danışmanı: Yrd. Doc. Dr. Sıddıka Berna ÖRS YALÇIN**

**MAYIS 2011**

## ÖNSÖZ

Tez çalışmam sırasında kıymetli zamanını benden esirgemeyen, tecrübesiyle ve bilgisiyle beni destekleyen, Yrd. Doç. Dr. Sıddıka Berna ÖRS YALÇIN'a ve İTÜ Gömülü Sistemler Laboratuarında yaptığım çalışmalarda bana yol gösteren Ar. Gör. Ramazan Yeniçeri'ye ve Ar. Gör. Tuba Ayhan'a sonsuz saygı ve teşekkürlerimi sunarım.

Ayrıca eğitim hayatım boyunca maddi ve manevi desteklerini hissettiğim aileme sonsuz saygı ve teşekkürü borç bilirim.

Mayıs 2011 M.Akif Özkan

**İÇİNDEKİLER**

**Sayfa**

**KISALTMALAR**

**GSM** : Global System for Mobile Communications
**AES** **:** Advanced Encryption Standard
**FPGA** **:** Field Programmable Gate Array
**LCD** : Liquid Crystal Display
**LSF** : Linear Spectrum Frequencies
**KBPS** **:** Kilo bit per second
**GSM FR** **:** GSM Full Rate
**LPC** **:** Linear Predictive Coding
**LBG** **:** Linde Buzo Gray Algoritması
**PSTN** **:** Public Switched Telephone Network
**CELP** **:** Code Excited Linear Prediction

## GSM AĞI ÜZERİNDEN GÜVENLİ SES İLETİMİ

## ÖZET

Mobil iletişim için küresel sistem (Global System for Mobile Communications, GSM) ile haberleşmenin yaygınlaşması hayatımızı kolaylaştırmıştır ancak çok ciddi güvenlik açıklarını da beraberinde getirmiştir. Bu çalışmada, cep telefonları ile haberleşerek GSM üzerinden yapılan görüşmelerin servis sağlayıcıları dahil üçüncü kişilere karşı korunmasını sağlayacak bir sistem geliştirilmiştir.

GSM hattı, daha verimli ve kaliteli bir iletim yapabilmesi için konuşmalara karşı duyarlıdır. Ayrıca GSM hattı üzerinden konuşmanın iletilebilmesi için, bir takım sıkıştırma algoritmaları kullanılır. Bu sebeplerden dolayı konuşma şifrelendikten sonra doğrudan hatta verilemez. Bu çalışmada, sayısal olarak şifrelenen ses, tasarlanan kodlayıcı sayesinde GSM hattı tarafından bastırılmayacak konuşmalara dönüştürüldükten sonra iletilir. Ayrıca bu tezde, yeni bir yöntem olarak eğitilebilir kodlayıcı yapıları önerilmektedir.

Çalışma boyunca sayısal verinin şifrelenmesinin sağlanması için, 128 bit gelişmiş kodlama standardı (Advanced Encryption Standard, AES) algoritması sahada programlanabilir kapı dizisi (Field Programmable Gate Array, FPGA) üzerinde gerçeklendi. Sayısal verilerden, bozulmadan iletilebilecek konuşmaların elde edilebilmesi için gerekli özelliklere karar verildi ve istenen özelliklere ait parametreler NTIMIT ses veritabanı taranarak elde edildikten sonra LBG (Linde–Buzo–Gray algorithm) algoritması kullanılarak kod kitapları tasarlandı. Sayısal verileri, kod kitaplarını kullanarak kodlayan ve doğrusal öngörü yöntemiyle (LPC) konuşmalar sentezleyebilen bir kodlayıcı tasarlandı. Matlab ortamında tasarlanan kodlayıcı ile şifreleme donanımı, seri port kullanılarak haberleştirildi. Cep telefonunu ifade eden 13 kbps (kilo bit per second) GSM FR (Full Rate) kodlayıcısı bilgisayar ortamında modellendi. FPGA de gerçeklenen şifreleme donanımına karakter LCD (Liquid Crystal Display) bağlandı. Sonu colarak sistemin çalışır olduğu gözlemlendi.

**SUMMARY**

**SECURE VOICE COMMUNICATION OVER GSM**

With the spread of Global System for Mobile communications (GSM) usage for communication facilitated our lives, but also brought many serious security vulnerabilities. In this study, a system developed which communicates through GSM mobile phones and provides protection for interviews against third parties including with service providers developed. GSM line is sensitive to human speech to be more efficient and provide more quality for transmission. In addition, a tool should be used to compress speech to transmit speech over GSM. For these reasons, speech cannot be transmitted to the GSM line directly after encrypted. In this study, the encrypted speech which is a data stream, formed speech like waveform by the designed coder to transmit through the GSM line. In addition, in this thesis, a new method is suggested as a trained coder structures.

During the study to provide encryption for data stream, 128-bit advanced encryption standard (AES) algorithm is designed using the field programmable gate array (FPGA). The features needed to achieve to transmit speech with no disruption is decided and desired speech characteristics are obtained by scanning the database of NTIMIT, then LBG (Linde-Buzo-Gray algorithm) algorithm is used to design codebooks which includes speech parameters. A coder using the linear predictive coding method (LPC) and codebooks, is designed to synthesize speech like waveforms from data stream. The coder is designed with MATLAB and communicated with encryption hardware using the serial port. Representing the mobile phone, 13 kbps (kilo bits per second) GSM FR (Full Rate) codec is modeled on computer. AES hardware is connected with character LCD (Liquid Crystal Display) in FPGA. As a result system is observed to work.

# 1. GİRİŞ

GSM teknolojisi sayesinde, herhangi kablolu bir bağlantıya ihtiyaç duyulmadan dünyanın neredeyse her noktasından haberleşme mümkün hale gelmiştir. Hayatın her alanında önemli bir yere sahip olan GSM üzerinden yapılan konuşmalar; bireylerin şahsi bilgilerinden gizli tutulması gereken devlet bilgilerine kadar, özellik arz eden içeriklere sahip olabilirler. Bu yüzden konuşmaların güvenliği çok önemlidir.

Genel aktarmalı telefon şebekesi (Public Switched Telephone Network, PSTN) dünya genelinde kullanılan devre aktarmalı telefon ağıdır. Başlangıçta sabit analog telefon şebekesi olarak kurulan bu ağ günümüzde neredeyse tamamen sayısaldır ve sabit telefonların yanı sıra mobil telefon hatlarını da içermektedir. Radyo frekansı hatlarından gelen konuşmalar arasındaki bağlantı, ağ seçim devreleri (core circuit switched networks) ile GSM-GSM veya GSM-PSTN olarak yapılır. GSM haberleşmesinde servis tarafından iletilen konuşma, radyo frekansı hat girişleri arasında güvenlidir [1]. Ancak iletilecek konuşma baz istasyonları arasındaki ağ seçim devrelerinde, darbe kodu modülasyonlu (Pulse code modulation, PCM) ve diferansiyel darbe kodu modülasyonlu halde iletileceğinden güvenli değildir [2]. Ayrıca konuşmanın şifrelenmesi servis sağlayıcısına bağlıdır ve istenildiği takdirde şifrelenmeyebilir. Bu yüzden konuşmanın güvenli hale getirilmesi için, konuşma şebekeye verilmeden şifrelenmelidir.

Bu çalışmada, GSM üzerinden yapılan görüşmelerin servis sağlayıcıları dahil üçüncü kişilere karşı korunmasını sağlayacak bir sistem geliştirilmiştir. Tasarlanan sistem, cep telefonları ile haberleşerek konuşmaları güvenli hale getirebilen gerçek zamanlı bir modül tasarımı için gerekli olan teorik altyapıyı ve tasarlanacak modülün temel yapısını içermektedir. Konuşmanın GSM üzerinde tasarlanması istenen gerçek zamanlı modül ile güvenli bir şekilde iletimi sırasında; gelen konuşma sayısallaştırılarak bit dizileri elde edilecek, elde edilen bit dizileri şifrelenecek, şifrelenen sayısal veriler kodlanarak sentetik konuşmalar üretilecek, üretilen sentetik konuşmalar cep telefonuna gönderilecek, cep telefonunun GSM kodlayıcısıyla konuşma kodlandıktan sonra hatta gönderilecek; hatta gönderilen konuşmanın iletimi

sırasında GSM servis sağlayıcısı tarafından, konuşma yeniden kodlanacak ve kodu çözülecek; Karşı taraftaki cep telefonu tarafından konuşma alındıktan sonra sistemin girişindeki halini alabilmesi için benzer işlemler tekrar yapılacaktır. Yapılması gereken işlemlerin çokluğu sebebiyle sistemin gecikmesi artar. Gecikme miktarının kabul edilebilir sınırı aşmaması için tasarlanması istenen modülün çok hızlı çalışması gerekmektedir. Tasarlanacak modülün birden fazla yonga üzerinde olması, güvenlik sebebiyle istenen bir durum değildir. Bu sebeplerden dolayı, istenen modül FPGA üzerinde gerçeklenecek bir donanım olacaktır. Tasarlanan modül, temel olarak iki blokdan oluşmaktadır. İlk blok sayısallaştırılan konuşmanın şifreleneceği donanımı içerir. Bu donanım, gelişmiş kodlama standardı (Advanced Encryption Standard, AES) şifreleme algoritmasının FPGA üzerinde gerçeklenmesidir. İkinci blok, sayısal bit dizilerinden cep telefonuna gönderilecek sentetik seslerin üretildiği kodlayıcı yapısını içerir. Kodlayıcı; 13kbps GSM Full Rate düşük bit hızlı konuşma kodlayıcısıyla (Low Bit Rate Codec) kodlanıp iletim için GSM hattına verildikten sonra karşı taraftan alındığında, doğru şekilde tanınarak sayısal karşılıklarının bulunabileceği sentetik konuşmaları üretir. Kodlayıcı; konuşmalara ait enerji, LSF (Linear Spectrum Frequencies) parametreleri, perde periyodu gibi özelliklerin saklandığı kod kitapları (codevector) kullanarak LPC (Linear Predictive Coding) yöntemiyle sayısal verilere karşı düşen konuşmaları sentezler. Sistemin kodlayıcı bloğu MATLAB ortamında tasarlanmıştır. Ayrıca kodlayıcı tasarımı için gerekli kod kitapçıkları NTIMIT veritabanı taranarak, LBG (Linde Buzo Gray) algoritmasıyla oluşturulmuştur. FPGA de gerçeklenen şifreleme donanımı seri port bağlantısıyla MATLAB ortamında tasarlanmış kodlayıcı ile haberleştirilmiştir. FPGA de şifreleme donanımına bağlanan karakter LCD bağlanmıştır. Tasarlanan sistem, bilgisayar ortamında 13kbps GSM 06.10 Full Rate kodlayıcısı ile haberleştirilmiştir. Bu sayede GSM üzerinden güvenli ses iletimi, iletim hattı etkisi ihlal edilerek modellenmiştir.

Tasarlanması istenen modül FPGA üzerinde tasarlanacak bir donanım olması özelliği ile bir ilktir. Ayrıca benzer fikirlerle önerilen kodlayıcı yapıları olmasına rağmen, oluşturulan kod kitapçıkları çalışmaya özgüdür.

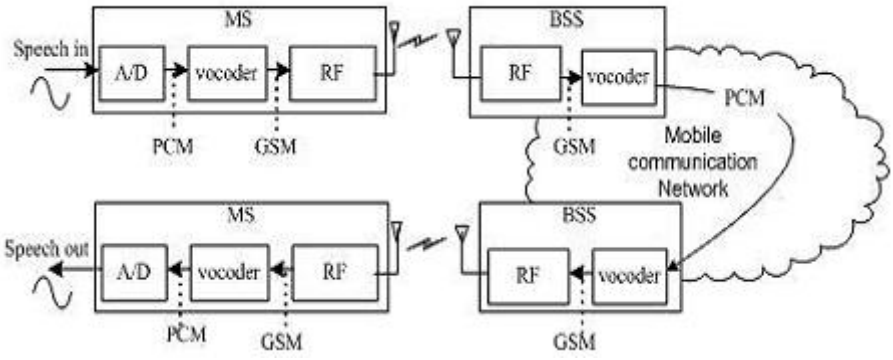
Tasarlanacak sisteme ait genel şema ve kodlayıcı yapısı ikinci bölümde, LPC ile konuşmanın analizi ve sentezi üçüncü bölümde, gerçeklenen AES şifreleme algoritması dördüncü bölümde, kod kitaplarının oluşturulması ve konuşmanın

kodlanması için kullanılması dördüncü bölümde anlatılmış, beşinci bölümde elde edilen sonuçlardan bahsedilmiştir.

## 2. KONUŞMANIN GSM ÜZERİNDEN ŞİFRELENEREK GÖNDERİLMESİ

Güvenli mobil iletişim için GSM hattının üzerindeki konuşma kanalı, veri kanalı ya da 3G iletişim kanalı kullanılarak gönderilebilir. Mesajların iletildiği GSM veri kanalı, konuşma kanalına göre kısıtlanmıştır ve konuşma kanalına göre iletilebilecek veri kapasitesi düşüktür [3]. Özellikle uluslar arası iletişimlerde uyum problemi yaşanır. Ayrıca bağlantının kurulması için geçen süre konuşma kalanlına göre fazladır [2]. 3G iletişim kanalı, verinin gönderilmesi için daha uygun olsa da henüz tam oturmuş bir sistem değildir [2] ve kullanım alanı olarak GSM kadar yaygın değildir ayrıca şifrelendikten sonra konuşma olarak iletilen bilgilerin gizliliğinin daha fazla olacağı açıktır.

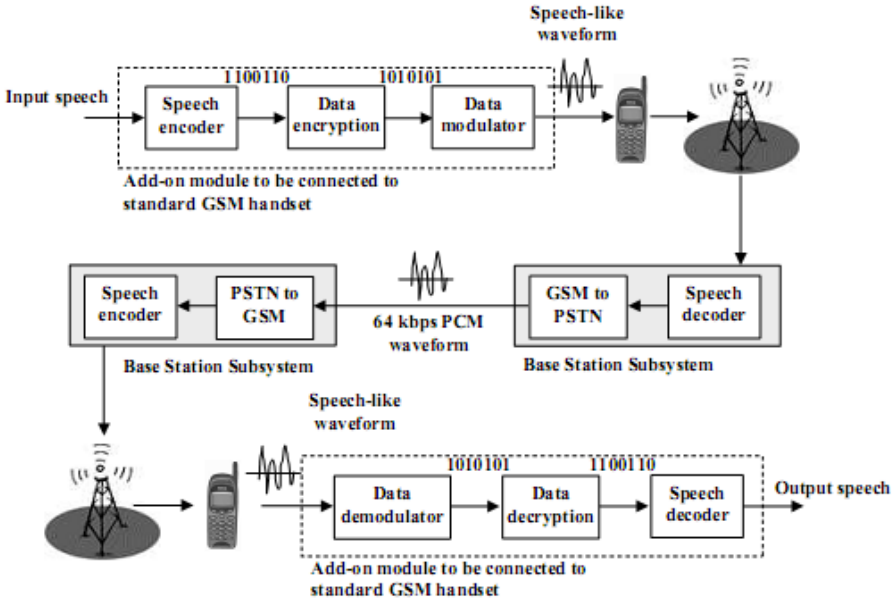
GSM iletişiminde hattın verimli kullanılabilmesi için konuşma, temel özelliklerine göre sıkıştırılarak hatta verilir ve alındıktan sonra sıkıştırma açılır. Sıkıştırma işlemi sonucunda oluşturulan sinyal, sıkıştırılmamış haline temel karakteristikleri açısından benzer ancak dalga formları aynı değildir. Konuşma ayrıca GSM iletim hattı üzerinde yeniden kodlanır, bu şekilde iki kere kodlanmasına, peş peşe kodlanma (tandeming) denir.




**Şekil 2.1:** Sesin GSM konuşma kanalı üzerinde iletimi [5].

Şifreleme işlemi, konuşmayı hatta gönderen konuşma kodlayıcısından önce ya da sonra yapılabilir. GSM hattı gelen ses üzerinde duyarlıdır ve konuşma olmayan sinyalleri bastırır. Bu yüzden sesin şifrelenerek iletilebilmesi için şifrelendikten sonra elde edilecek sayısal verilerden konuşmaya benzeyen sesler sentezleyecek bir

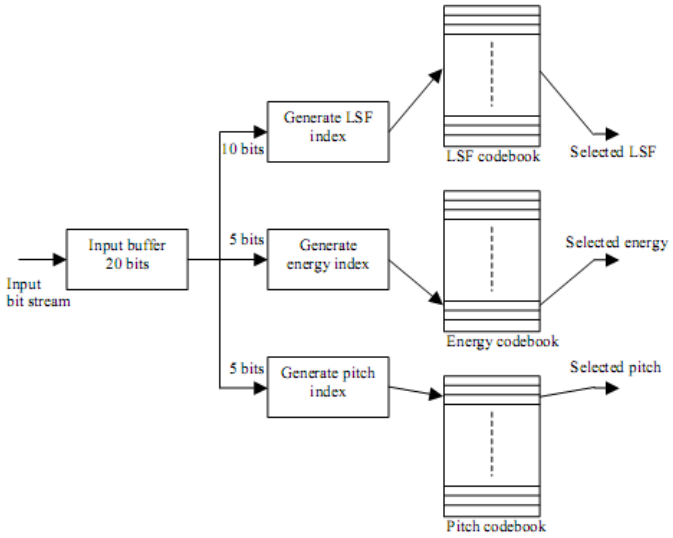
kodlayıcı gerekir. Tasarlanacak kodlayıcı, konuşma kodlayıcısında önce şifrelenen verileri konuşmalara çevirerek konuşma kodlayıcısına gönderecek, konuşma kodlayıcısı ise bu konuşmaları GSM hattına gönderecektir.




**Şekil 2.2:** Tüm sisteme ait şema [2].

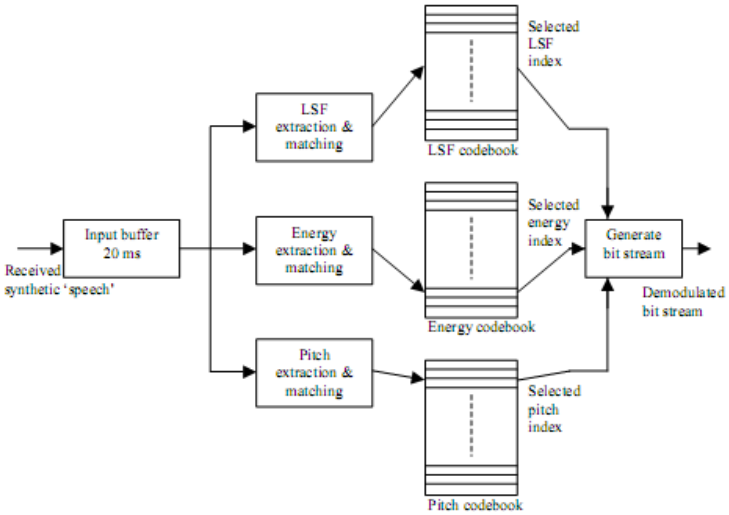
PSTN için tasarlanmış kodlayıcılar konuşmaya ait birçok parametreyi hesaba kattıkları için bu işe uygun değildirler. Konuşmanın en temel özniteliklerinin saklandığı kod kitapçıkları (Codebook) sayesinde sayısal verilerden konuşmalara ait semboller üretilerek GSM konuşma hattında daha hızlı veri iletiminin sağlanabileceğinin söylendiği çalışmalar vardır [3]. Bilgisayar ortamında; GSM üzerinden güvenli ses iletiminin sağlanması amacıyla, sayısal veriden 12 kbps GSM Enhanced Full Rate kodlayıcısına verilmek üzere sentetik bir ses üreten kodlayıcının önerildiği çalışmalar vardır [2][4]. Aynı modeli kullanarak 13kbps GSM Full Rate kodlayıcısı için simülasyon yapılan bir çalışma da bulunmaktadır [5]. Bahsi geçen araştırmalarda her ne kadar sistem modeli verilse de kod kitapları verilmemiştir ve bu modele sahip alınmış bir patent vardır.

Tasarlanacak kodlayıcı enerji, vokal filtre spektrum eğilimi, perde frekansı gibi konuşmayı ifade eden en önemli parametreleri kullanarak LPC yöntemiyle konuşmalar sentezleyecektir. Bu iş için konuşma kodlayıcısından ve GSM iletimi sırasındaki kodlayıcıdan bozulmadan geçebilecek konuşmalara ait parametrelerin saklandığı kod kitapları oluşturulacaktır.




**Şekil 2.3:** Kodlayıcının sayısal verilerden konuşma sentezleyen kısmı[4].

Sentezlenen konuşma GSM FR ses kodlayıcısı ile iletim hattına gönderilecek, karşı taraf iletilen konuşmayı aldıktan sonra kod kitaplarına bakarak sayısal veri karşılıklarını elde edecektir. Elde edilen şifre kodlarının doğru çözülebilmesi için kodlayıcı tarafından algılanan ve iletilen sayısal verilerin aynı olması gerekir.




**Şekil 2.4:** Kodlayıcının konuşmadan sayısal verileri sentezleyen kısmı[4].

Sistemden kaynaklanacak hataların giderilebilmesi için kod kitaplarının hataları düzeltecek şekilde tasarlanması gerekir. Bu çalışmada iletimin, minimum hatalı olması için kod kitaplarına yerleştirilecek konuşmalara ait özellikler belirlendi. İstenen özellikteki konuşmalara ait, temel parametrelerin öğrenilmesi için NTIMIT ses veritabanı tarandı. İletimdeki hataların düzelterek, sayısal verilerin değişmeden elde edilebilmesi için veritabanı taranarak öğrenilen parametreler LBG algoritması

kullanılarak birbirine en uzak noktalarda seçildi. Ayrıca girişteki ve çıkıştaki kod kitapları farklı seçilerek, sistemden kaynaklanan hataların düzeltilebileceği gösterilerek eğitilebilir kodlayıcı yapılarına dair yeni bir metot önerildi.

# 3. DOĞRUSAL ÖNGÖRÜ YÖNTEMİYLE KONUŞMANIN KODLANMASI

Konuşma sinyalinin, kullanılacağı uygulamaya göre daha az parametre ile ifade edilmesine konuşmanın sıkıştırılması (Speech Compression) ya da konuşmanın kodlanması (Speech Coding) denir. Konuşmanın kodlanması için geliştirilen yöntemler genellikle kayıplıdır ancak insan kulağı, konuşmanın kalitesindeki düşüşü yeterince algılayamadığı için kabul edilebilirdirler. Sıkıştırılma işlemi sırasında konuşmadaki sessiz kısımlar iletim için kodlanmaz, bu sayede konuşmayı temsil eden bit sayısında ciddi bir düşüş sağlanır.

Bant genişliğini artıracak teknolojilerin gelişmesine rağmen, konuşmanın sıkıştırılması ve bant genişliğinin daha verimli kullanılması haberleşme teknolojisinin en temel problemlerindendir. Konuşmanın sıkıştırılması için bu zamana kadar çok çeşitli teknikler geliştirilmiştir ancak bu tekniklerin büyük bir kısmı, konuşmanın 300 Hz ile 3400 Hz arasında sınırlı bir frekans aralığına sahip olması ve konuşmanın yavaş anatomik hareketlerden oluşturulması ilkeleri üzerine kurulmuştur.

1930'larda, darbe kod modülasyonunun (Pulse Code Modulation, PCM) geliştirilmesi ses ve konuşmanın sıkıştırılması alanındaki çalışmaların başlangıcı olarak kabul edilir. Telefon şirketlerinin bant genişliği maliyetlerinin artması üzerine 1970'lerde doğrusal öngörü ile kodlama (LPC, Linear Predicitive Coding) algoritmaları geliştirilmeye başlandı ve 1984 de United States Department of Defense, LPC algoritmalarını standartlaştırarak Federal Standard 1015 i yayınladı [6].

Konuşma, genellikle konuşma kodlayıcılar (vocoder, voice coders) kullanılarak sıkıştırılır. Konuşma kodlayıcılar, temel olarak model parametreleri tabanlı (model-base coders) ve dalga şekli tabanlı (waveform-following coders) olmak üzere ikiye ayrılırlar. Kodlama hatalarını yok sayarsak dalga şekli tabanlı konuşma kodlayıcıları ile konuşmanın dalga şekli hatasız olarak geri elde edilebilir ancak model parametreleri tabanlı kodlayıcılarda, konuşma bazı parametreler kullanılarak yeniden sentezlendiğinden kodlama hatası olmasa bile konuşmanın zaman domenindeki orijinal dalga şekli geri elde edilemez.

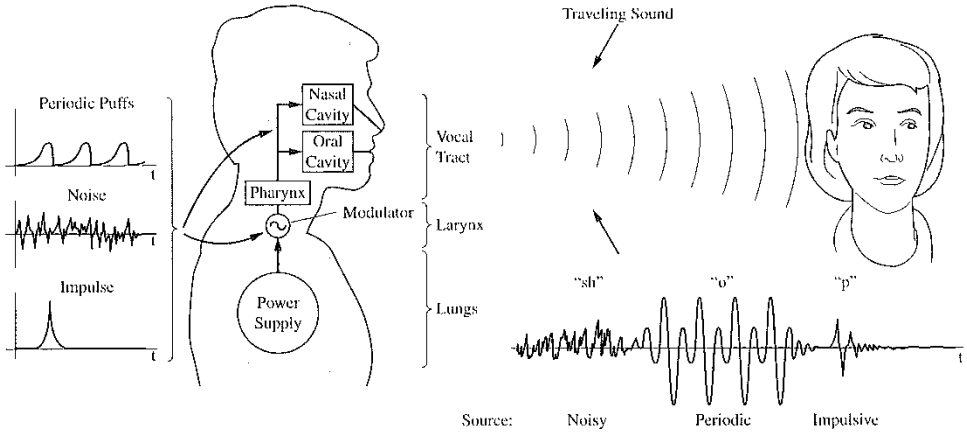
LPC, model parametreleri tabanlı bir konuşma kodlayıcısı tipidir. Konuşma kodlayıcıları; bit oranı (bit rate), gecikme (delay), karmaşıklık (complexity) ve kalite (quality) özellikleri açısından incelenir. Bit oranı, konuşma kodlayıcısının sıkıştırma miktarını ifade eder. Konuşma 8 khz ile örneklenir ve her bir örnek 8 bit ile ifade edilirse konuşma 64 kbps (kilo bit per second) bit rate ile ifade edilir ve 8 khz örnekleme frekansında 64 kbps den daha düşük bit rate ile iletilen konuşmalar sıkıştırılmış olarak kabul edilir. LPC konuşma kodlayıcıları kullanılarak 2.4 kbps bit oranında iyi bir sıkıştırma yapılabilir [7]. Konuşmanın iletimi sırasında 1 saniyelik konuşma için 300 milisaniyeye kadar gecikme kabul edilebilir. Karmaşıklık, konuşma kodlayıcılarında maliyeti ve harcanan gücü etkileyen önemli bir parametredir. Yüksek sıkıştırma oranlarının bulunduğu LPC tabanlı konuşma kodlayıcılarının gerçek zamanlı kullanımları için çift işlemci kullanılması tercih edilir. Algoritmanın donanım (hardware) ile tasarlanması sayesinde gerçek zamanlı kullanım için ciddi kazançlar sağlanabilir. Kalite, sıkıştırılma işleminden sona ortaya çıkan sesin gerçek sese yakınlığını ifade eder. Yüksek sıkıştırma oranının bulunduğu LPC konuşma kodlayıcılarda ses kalitesinden feragat edilir. Kodlayıcının çıkışındaki konuşmaya sentetik konuşma (synthetic speech) denir.

LPC algoritması, kodlama (encode) ve kod çözme (decode) olarak adlandırılan iki temel bloktan oluşur. Kodlama bloğunda konuşma çerçeveler (frame) halinde incelenir. İncelenen çerçeve analiz edilerek, yeniden sentezleme için tasarlanacak filtrenin katsayıları elde edilir. Konuşmayı ifade eden parametreler kodlama işleminden geçirilerek bit dizileri (bit stream) elde edilir. Kod çözme kısmında ise elde edilen bit dizisi kodlama aralıklarına göre çözülerek sentezleme filtresi ve bu filtreyi uyaran işaret elde edilerek sentetik ses sentezlenir.

### 3.1. İnsanda Konuşmanın Modellenmesi

İnsanda konuşmanın yapıtaşı olan ses, akciğerden çıkan havanın vokal sistemden geçip ağızdan dışarı çıkmasıyla üretilir. Vokal sistem boğaz, dil, burun ve ağızdan oluşur. Konuşurken, vokal sistem sürekli değişir ve akciğerden gelen havaya titreşimler katar. Akciğerden çıkan havanın şiddeti sesin gürültü seviyesini etkiler. Akciğerden gelen havanın ses tellerini titreştirip titreştirmemesi ve vokal sistemin kalan kısmının aldığı şekil farklı seslerin oluşturulmasını sağlar. Vokal sisteme giren hava periyodik olduğunda sesli (voiced), rastgele ya da türbülansta olduğunda sessiz

(unvoiced) sesler oluşur. Sesli seslerin enerjileri, sessiz seslere oranla daha fazladır. İnsanda sesin oluşması sistemi modellendiğinde akciğer, kaynak; vokal sistem, konuşmamızı ortaya çıkaran ses çeşitlerini oluşturan filtre olur. LPC konuşma kodlayıcısında da kullanılan bu modellemeye kaynak-filtre modeli (source-filter model) denir.

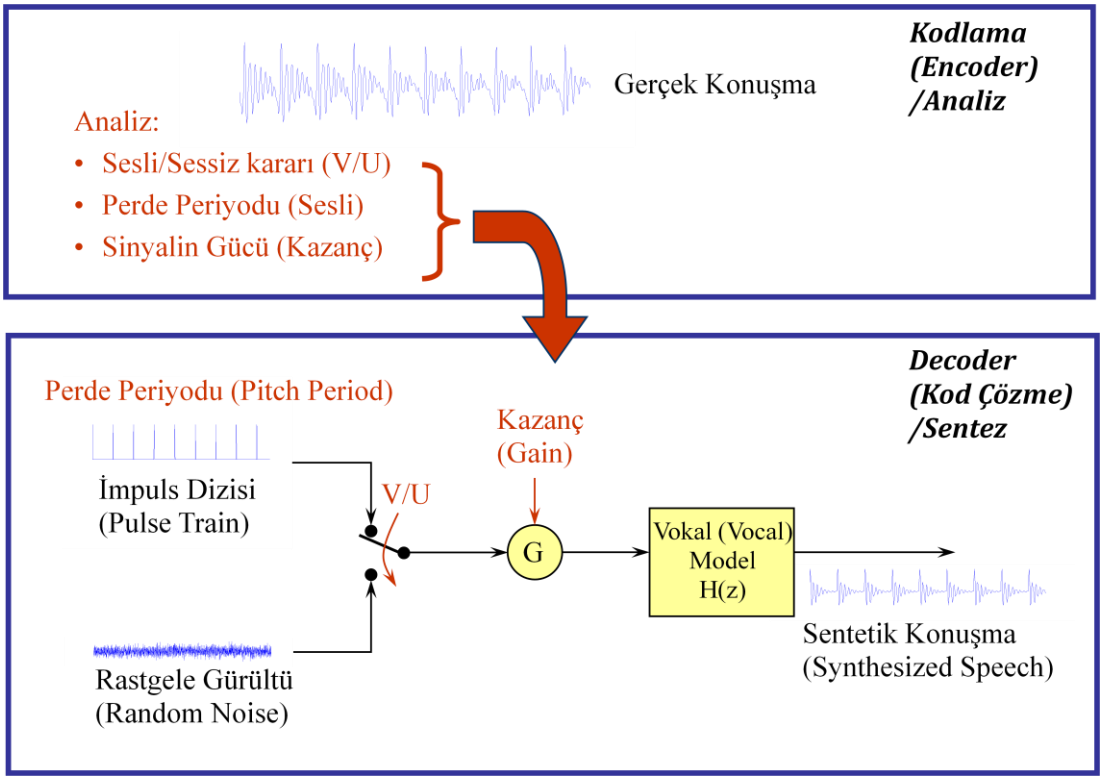



**Şekil 3.1:** İnsanda ses oluşumunun modellenmesi [8].

### 3.2. LPC Modeli

LPC, konuşma analizinde kullanılan en güçlü uygulamalardan biridir. Bu algoritma, konuşmaya ait temel parametrelerin (sesli olup olmaması, formantları, periyot bilgisi (pitch), vb. ) bulunmasında ve konuşmanın düşük bit hızında (low bit rate) iletilmesi yada saklanması için modellenmesinde kullanılan temel metotlardan biri haline gelmiştir[7]. Bu metodun en önemli özelliği konuşmayı lineer zamanla değişen bir sisteme ait parametrelerle, güvenli ve hızlı bir şekilde ifade edilmesine imkân vermesidir.

LP algoritması kullanılarak konuşmanın analiz edilmesine, LP (Linear predictive) konuşma analizi; analiz sonucu elde edilen parametreler kodlanıp sayısal bir veri elde edilmesine, LPC konuşmanın kodlanması denir. LP konuşmanın analiz edilmesi ve LPC konuşmanın kodlanması, analiz/kodlayıcı (encoder) ve sentez/kod çözücü (decoder) olmak üzere iki temel kısımdan oluşur. Konuşma, yavaş değişen bir işaret olduğundan kısa çerçevelere (20ms, 40ms gibi) bölündükten sonra elde edilen konuşma parçaları durağan sinyaller olarak kabul edilirler.



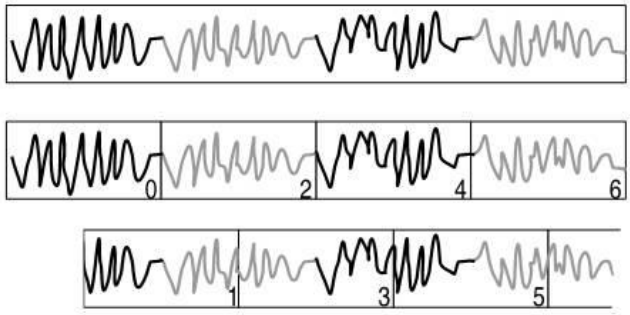

**Şekil 3.2:** LPC Konuşma Kodlama/Analizi [9].

FS-1015 standardına göre gelen konuşma 8 khz de örneklenir. Analiz kısmında, konuşma kısa çerçeveler ile parçalar haline ayrılıp her bir parça için; sinyalin gücü bulunarak kazancın, sesli (voiced) veya sessiz (unvoiced) olması, sesli parçalar için perde periyodu (pitch period) ve vokal sistemin tasarlanması için LP katsayıları (Linear prediction coefficients) bulunur. Sentez kısmında, analizden gelen parametreler incelenerek konuşma sinyali tekrar elde edilir. Sinyal; sesli ise perde periyoduna sahip bir impuls dizisi (impulse train), sessiz ise rastgele gürültü (rondom noise) elde edilen kazanç ile çarpılarak vokal model bloğuna giriş olarak verilir. LP katsayıları kullanılarak vokal model filtresi tasarlanır.

### 3.2.1. Çerçevelenen Konuşmaların Örtüştürülerek İncelenmesi

Konuşma sinyalleri analiz edilirken çerçevelere ayrıştırılmalarının gerekliliği konuşuldu. Çerçevelere ayırma işleminde konuşma sinyallerine ait bazı özellikler kaybolabilir. Aynı özelliğe sahip konuşma parçası, yarısı önceki çerçevede yarısı sonraki çerçevede olacak şekilde bölünebilir. Bu durumda iki çerçevede de bu özellik görülemez. Bu kayıptan kurtulmak için çerçeveler genelde %50 ve %25 arasında bir oranla örtüştürülerek seçilerek konuşmaya ait daha fazla özellik yakalanmaya çalışılır.

%50 oranında örtüştürülerek çerçevelenen bir konuşma aralığı şekil 3.3'de bulunmaktadır.

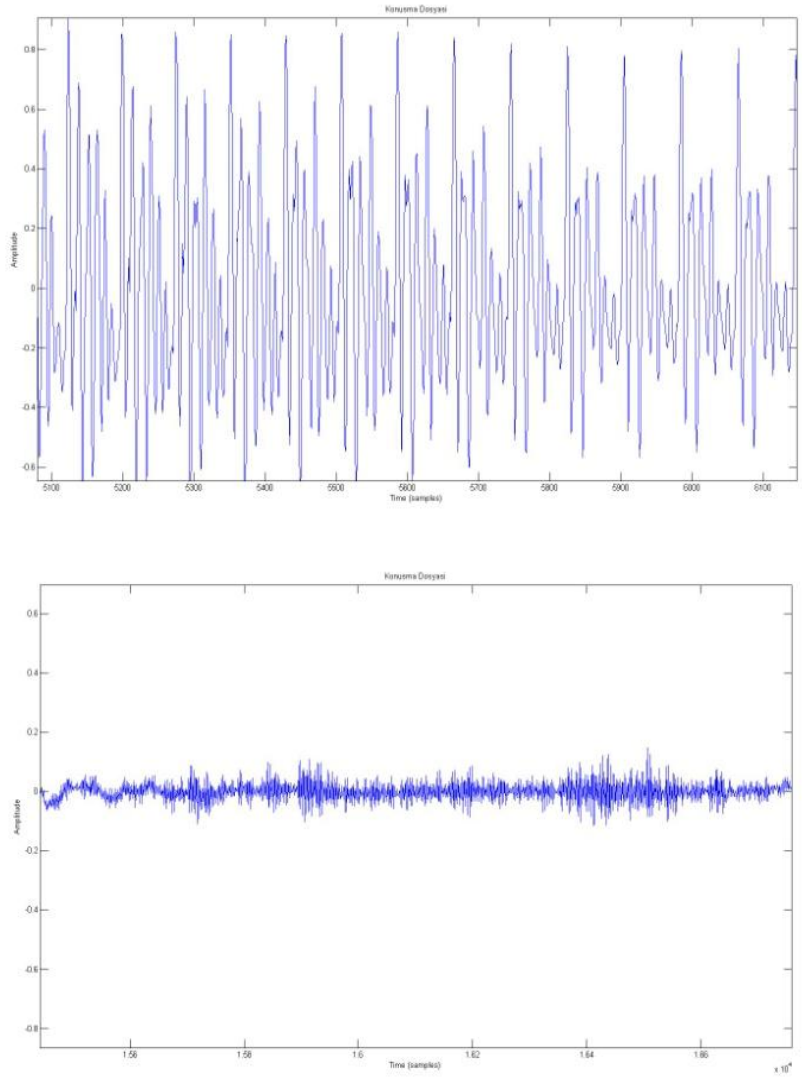



**Şekil 3.3:** %50 oranında örtüştürülerek seçilen bir konuşma aralığı [10].

Örtüştürme işlemi uygulanarak incelenen konuşmalarda, analizi yapılacak çerçeve sayısı artar; dolayısıyla her bir çerçeve, kendi uzunluğu boyunca sentezlendiğinde konuşma dosyası büyür. Analiz işleminde, örtüştürme yapıldığında sentezlenen konuşma çerçeve uzunluğu örtüşme oranına göre küçülür. Mesela %50 oranında örtüştürme yapılarak incelenen bir konuşmada, her bir periyot uzunluğundaki çerçeveler yarım periyot uzunluğunda sentezlenir.

### 3.2.2. Çerçevelenen Konuşma İşaretlerinin Seslilik Sınıflandırılması

Konuşma sinyalleri, sesli (voiced) veya sessiz (unvoiced) olarak sınıflandırılabilir. Şekil 3.4 de herhangi bir konuşmadan seçilmiş sesli ve sesiz konuşma çerçeveleri bulunmaktadır. Konuşmanın sesli aralıkları; sessiz kısımlarına nazaran daha yüksek genliklere, enerjiye sahiptirler ve periyodik bir yapıdadırlar. Sessiz aralıklar ise daha çok rastgele gürültü (random noise) yapısındadırlar ve herhangi bir periyodiklikten bahsedilemez [11]. Konuşmanın geçiş kısımları (transition regions) olarak isimlendirilen bazı aralıkları ise sesli ve sessiz seslerin birleşiminden oluşur.

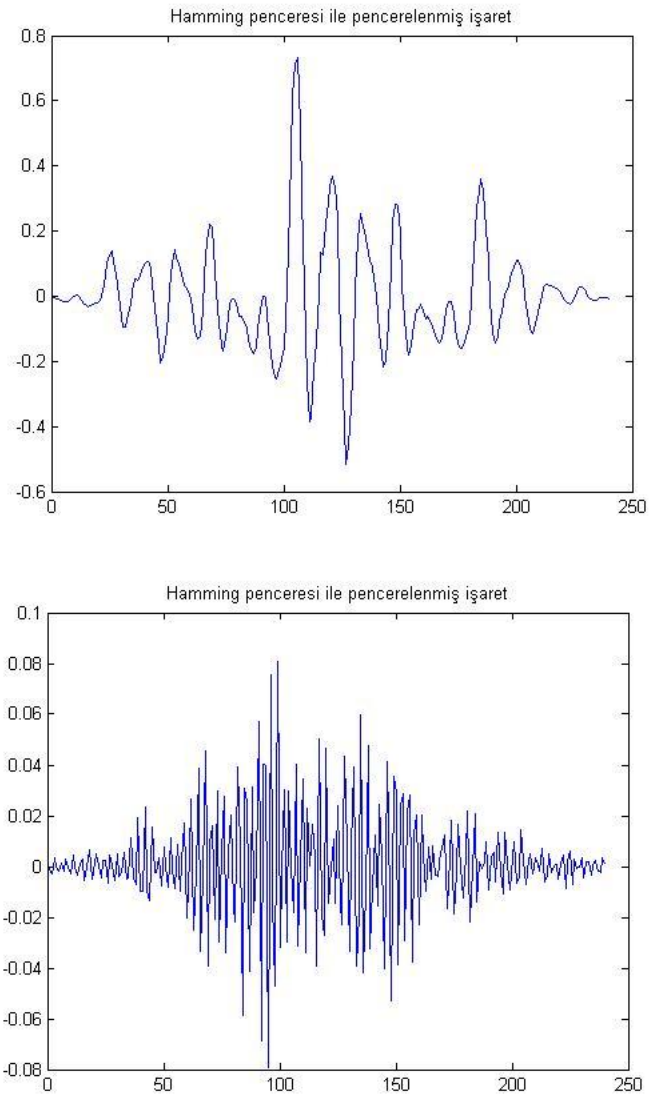



**Şekil 3.4:** Sesli ve Sessiz Konuşma Örnekleri

Konuşma işaretlerinde belirli özelliklerin incelenebilmesi için konuşma belirli zaman aralıklarına bölünür, konuşmanın bu aralıklarda incelenmesine kısa zaman analizi (short time analyse) denir. Bu işlem denklem (3.1) ve (3.2) de gösterilen dikdörtgen pencere (rectangular window), hamming penceresi (hamming window) gibi fonksiyonların, zaman domenindeki konvolüsyonu yani frekans domenindeki çapımıdır.

$$Dikdörtgen\ pencere: \quad \begin{matrix} w[m] = 1, & 0 \le \mathrm{m} \le \mathrm{N}-1 \\ 0 & diğer \end{matrix} \tag{3.1}$$

$$Hamming\ penceresi: \quad \begin{matrix} w[m] = 0.54 - 0.46cos(\frac{2\pi m}{N-1}), & 0 \le \mathrm{m} \le \mathrm{N}-1 \\ 0 & diğer \end{matrix} \tag{3.2}$$




**Şekil 3.5:** Dikdörtgen ve Hamming Pencere ile pencerelenmiş konuşma örnekleri

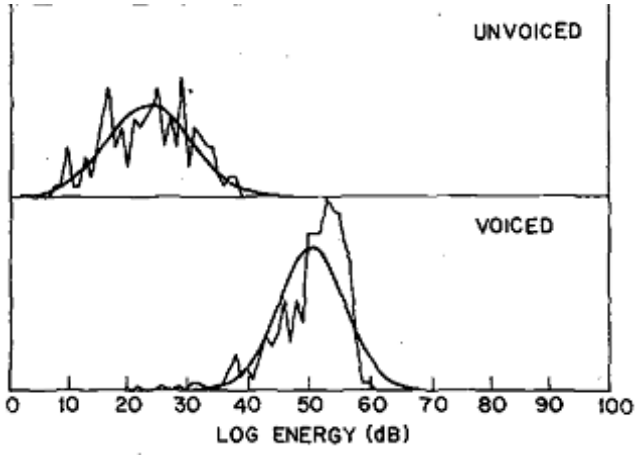
Ayrık zamanlı bir işaretin enerjisi ve kısa zaman analizi ile elde edilen işaretin enerjisi, denklem (3.3) ve denklem (3.4) de verilmiştir [11].

$$E = \sum_{m=-\infty}^{\infty} x^2(m) \tag{3.3}$$

$$E = \sum_{m=-\infty}^{\infty} x^2(m)h(n-m) = \sum_{m=n-N+1}^{n} x^2(m) \tag{3.4}$$

Konuşma; seslilik açısından sesli (voiced), sessiz (unvoiced) ve suskun (silence) olmak üzere üç temel bölümden oluşur. LPC yönteminde analiz edilecek dalga sesli ve sessiz olmak üzere iki sınıfa ayrılır. Sınıflandırma işleminin kalitesi

sentezlenen konuşmanın kalitesini doğrudan etkiler. Sınıflandırma işleminde sinyalin enerjisi ve enerji dağılımı (Low band to Full band ratio), periyodikliği, tepeli yapıya sahip olması (peakiness), ilk LP tahmin katsayısı, sıfırdan geçme oranı (zero-crossing rate) incelenen başlıca faktörlerdir.

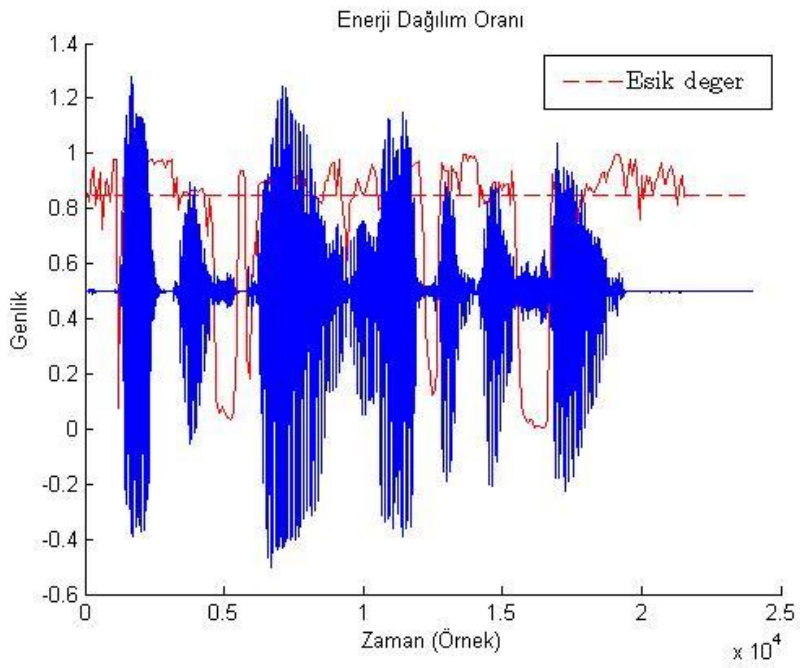



**Şekil 3.5:** Sesli ve sessiz sesler enerji dağılım grafikleri [12].

Konuşmaların sesli kısımlarının enerjilerinin logaritması sessiz kısımlarına göre genelde 15-20 dB daha fazladır [13]. Sesli, sessiz ayrışımında enerji büyüklüğüne göre sınıflandırma yapmak, özellikle kısık sesli konuşmalarla sessiz konuşmaların ayrışımında eşik değerin bulunması problemini getirir ve bunun için adaptif değişen bir eşik değeri gibi çeşitli metotlar önerilmiştir.

Konuşma sinyallerinin, frekans spektrumundaki enerji dağılımlarının incelenmesi sınıflandırma problemine hızlı bir çözüm sunar. Sesli konuşmaların düşük frekanslardaki enerji yoğunluğu yüksek frekanslardaki enerji yoğunluğundan daha yüksektir. Bu özellik sınıflandırma için kullanılabilir. Konuşma sinyalinin 1 khz deki enerjisinin toplam enerjiye oranı 1 e yakındır. Sessiz konuşmalarda ise bu oran çok daha düşüktür.

$$Enerji\ \ Dağılım\ Oranı = \frac{\sum_{i=1}^{N} s_{1khz}(i)^2}{\sum_{i=1}^{N} s(i)^2} \tag{3.5}$$



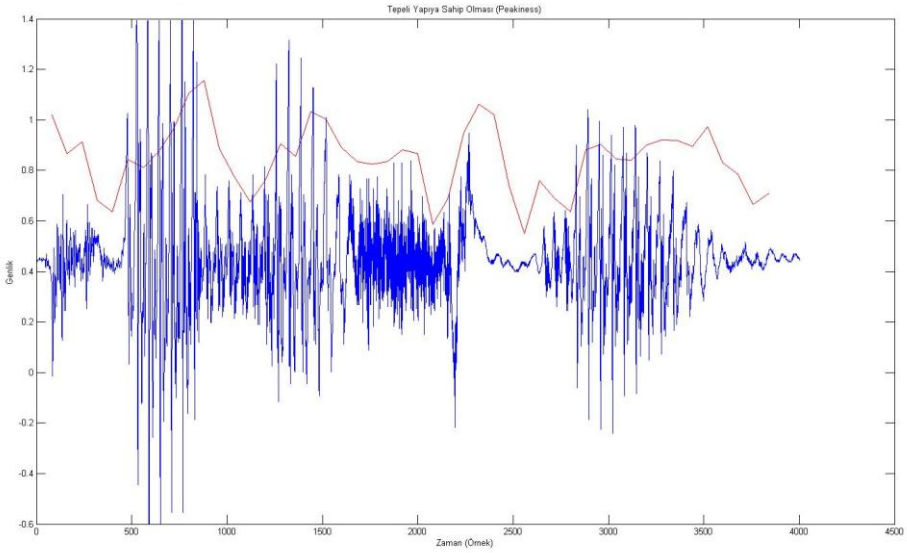

**Şekil 3.6:** Konuşmada 1khz den küçük frekanslardaki enerjinin tüm konuşmadaki enerjiye oran dağılımı, eşik değer 0.85

Şekil 3.5'de verilen sesli ve sessiz konuşma örneklerinde de görüldüğü gibi sesli konuşmalarda bir periyodiklik vardır. Dolayısıyla dalga bu periyot aralıkları ile kendine benzerliği incelenerek sesli ya da sesiz olduğu ortaya çıkartılabilir.

Sesli konuşmalar belirli periyotlarla tekrarlanan darbelerden oluşur. Tepeli (peakiness) yapıya sahip olma özelliğinden dolayı konuşma denklem (3.6) da verilen denklem ile sınıflandırılabilir.

$$PK = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} s(i)^2}}{\frac{1}{N}\sum_{i=1}^{N} |s(i)|} \tag{3.6}$$

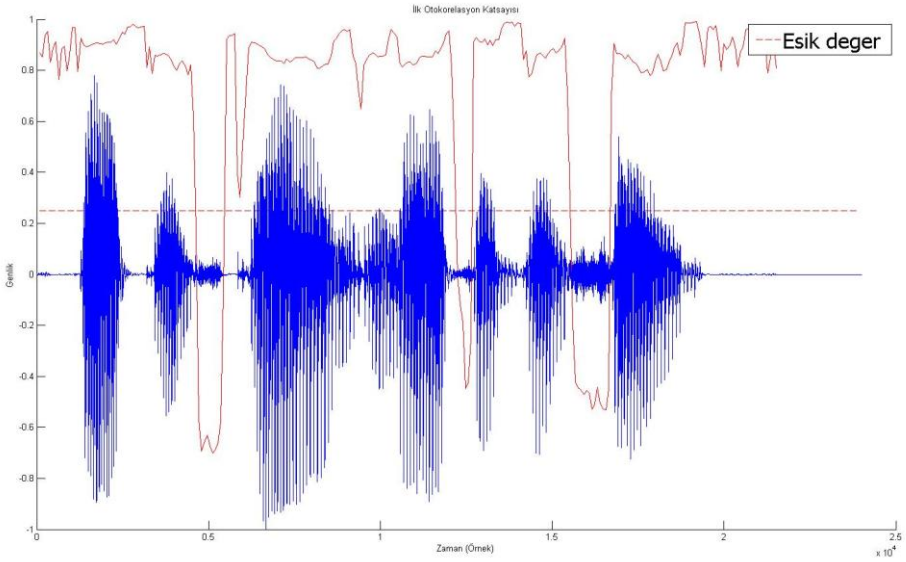
Konuşmanın tepelilik özelliğine göre seslilik açısından sınıflandırılması, şekil 3.7'de gözlemlenebileceği gibi çok ayırıcı bir metot değildir çünkü sessiz kısımlarda da belirli tepeler bulunabilir.




**Şekil 3.7:** Konuşmanın tepeli yapıya sahip olma özelliğine göre incelenmesi

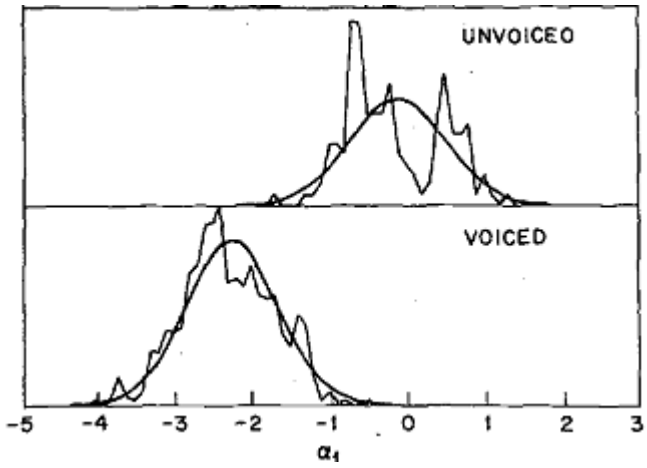
Sesli konuşma sinyallerinde örnekler bir yanındaki örneklere çok benzerler, bu sayede bir yanındaki örnekle korelasyonları çok yüksektir. Sessiz konuşmalarda ise böyle bir durum söz konusu değildir, daha çok rastgele değerlerden oluşurlar. Bu özelliklerinden dolayı hesaplanan LP tahmin katsayıları; sesli konuşmalarda -1'e yakın bir değer alırken, sessiz konuşmalarda 1 civarında bir değer alır. İlk tahmin katsayısı birinci dereceden normalize otokorelasyon ile ifade edilebilir.

$$\textit{Normalize edilmiş ilk otokorelasyon katsayısı} = \frac{\sum_{i=1}^{N} s(i)s(i-1)}{\sum_{i=1}^{N} s(i)^2} \qquad (3.7)$$



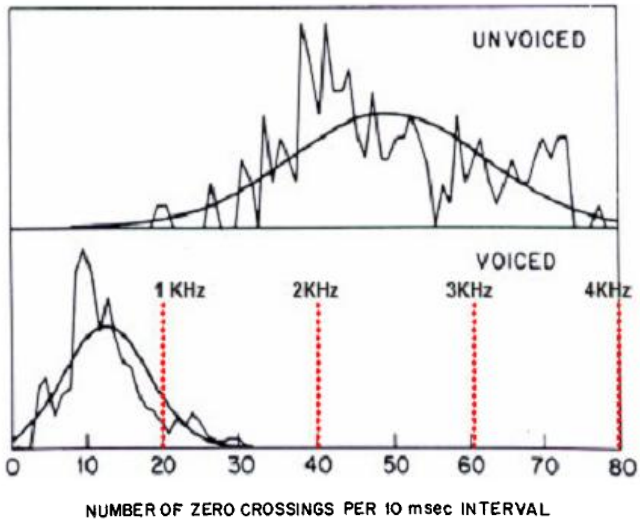

**Şekil 3.8:** Konuşmanın normalize ilk otokorelasyon katsayısına göre 0.25 eşik değeri ile sınıflandırılması

Bu fark sesli sessiz ayrımının yapımında güvenli ve etkili bir ayrışım sağlar [13]. R.Raibner bir çalışmasında dört faklı insana ait konuşmaları inceleyerek şekil 3.10'da verilen grafiği elde etmiştir.



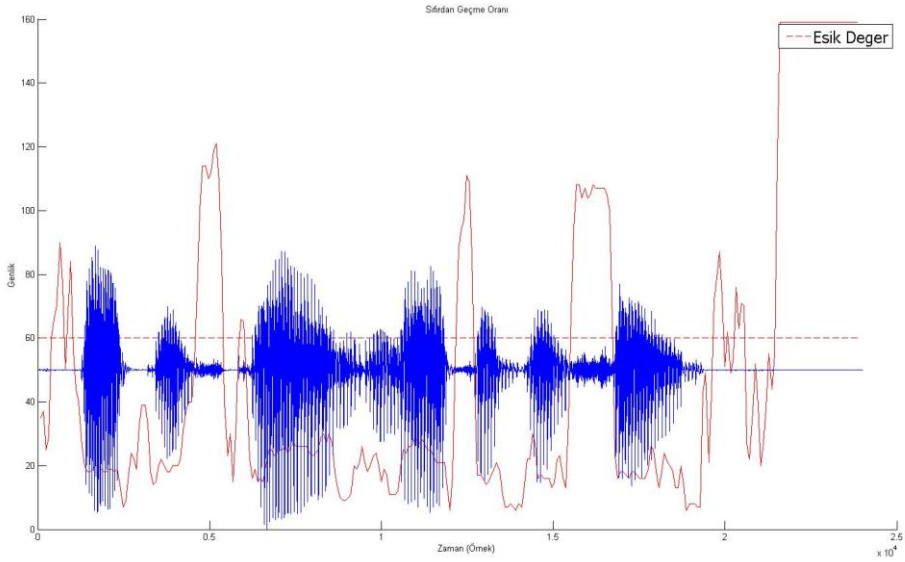

**Şekil 3.9:** İlk LPC katsayısının teorik ve ölçülen yoğunluk fonksiyonları [12].

Sıfırdan geçme oranı (zero-crossing rate), konuşma sinyalinin sıfırdan geçme miktarının hızını ifade eder. Sesli konuşmalar daha düşük frekanslarda gerçekleştiği için sessiz konuşmalara nazaran sıfırdan çok daha az geçer. Bu fark bize sesli/sessiz sınıflandırılmasında önemli bir imkân sağlar. R.Raibner bir çalışmasında dört faklı insana ait konuşmaları inceleyerek şekil 3.11'da verilen grafiği elde etmiştir.




**Şekil 3.10:** Sesli ve sessiz bölgelerin sıfırdan geçme oran yoğunluk fonksiyonlarının teorik ve pratik grafikleri [12].

Sesli konuşmalarda sıfırdan geçme hızının 3 khz in altında olması beklenir [12]. Örnek bir konuşmanın sıfırdan geçme oranına göre sınıflandırılması şekil 3.12'de gösterilmiştir.




**Şekil 3.11:** Konuşmanın sıfırdan geçme oranının 3khz eşik değeri ile sınıflandırılması

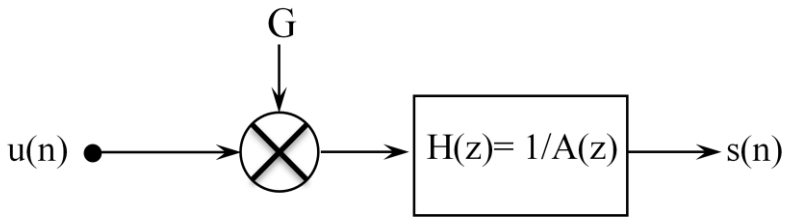
Sesli ve sessiz konuşmaların ayrımı için kullanılabilecek bazı özelliklerden yukarıda bahsedilmiştir. Sınıflandırılma probleminde incelenen özellikler ve bu özelliklerin karar vermeyi etkileme yoğunlukları hala araştırılan konular arasındadır. Karar verme amacıyla belirli yoğunluk katsayılarının lineer olarak belirlenmesinden,

örnek tanıma (pattern recognition) algoritmalarının kullanılmasına kadar farklı yöntemler önerilmiştir.

### 3.2.3. LPC Sentez Filtresinin Modellenmesi

Vokal sistem, konuşmanın sınırlı bir aralığında, tüm-kutuplu filtre modelinin (all-pole filter model; autoregressive model, AR) tasarlanması için tahmin katsayıları hesaplanarak modellenir. LP analizindeki temel fikir; konuşmanın n anındaki s[n] örneğinin, daha önceki çıkışların lineer kombinizasyonu olarak ifade edilebilmesidir.

$$s[n] = a_1 s[n-1] + a_2 s[n-2] + ... + a_p s[n-p] \tag{3.8}$$




**Şekil 3.12:** LP sentez modeli

Şekil 3.17 de ifade edilen LP sentez modeline göre, normalize edilmiş giriş işareti u(n), G kazancı ile çarpıldığında filtre uyarı işareti elde edilir. Uyarı işareti sentezleme filtresi ile süzüldüğünde çıkış işareti s[n]'i ifade eden eşitlik elde edilir.

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + Gu[n] \tag{3.9}$$

Denklem (3.9)'da ifade edilen eşitlik z domeninde incelendiğinde;

$$s[z] = \sum_{k=1}^{p} a_k z^{-k} S[z] + Gu[z] \tag{3.10}$$

$$s[z]\left(1 - \sum_{k=1}^{p} a_i z^{-k}\right) = Gu[z] \tag{3.11}$$

LPC sentez filtresinin transfer fonksiyonu;

$$H(z) = \frac{s(z)}{u(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)} \tag{3.12}$$

### 3.2.4. LPC Analiz Filtresinin Modellenmesi

Konuşmaya ait örneklerin tahmin edilen $\tilde{s}[n]$ değerleri ile gerçek $s[n]$ değerleri arasındaki fark hatayı verir.

$$e[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^{p} \alpha_k s[n-k] \tag{3.13}$$

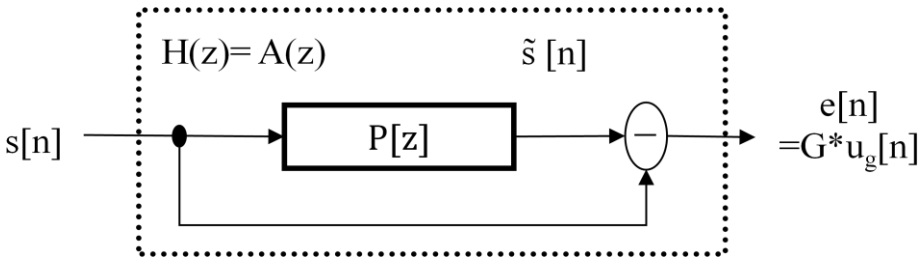
Hata işareti z domeninde incelenirse;

$$E(z) = S(z) - S(z)\sum_{k=1}^{p} \alpha_k z^{-k} = S(z)\left[1 - \sum_{k=1}^{p} \alpha_k z^{-k}\right] = S(z)A(z) \tag{3.14}$$

Denklem (3.14) indirgendiğinde A(z) hata tahmin filtresi bulunur:

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} = 1 - P(z) \tag{3.15}$$

A(z) filtresinin $\alpha_k$ katsayıları, s[n] i ifade eden $a_k$ katsayılarına eşit olursa Şekil 3.14 de gösterilen LPC analiz filtresi elde edilir.




**Şekil 3.13:** LP analiz filtre modeli

Sistemin ortalama karesel hatası;

$$E = \sum_{m=-\infty}^{\infty} e^2[m] = \sum_{m=-\infty}^{\infty} (s[m] - \tilde{s}[m])^2 = \left(s_n[m] - \sum_{k=1}^{p} \alpha_k s_n[m-k]\right)^2 \tag{3.16}$$

Ortalama karesel hatayı minimum yapan katsayılara, tahmin katsayıları (prediction coefficients) denir. Hatayı minimum yapan $\alpha_k$ katsayılarını bulmak için ortalama karesel hatanın bütün katsayılara göre kısmi türevleri sıfıra eşitlenerek katsayı adedi kadar denklem elde edilir. Bulunan denklemler çözülerek LP katsayıları bulunur.

Denklem (3.16) de verilen ortalama karesel hatanın $\alpha_k$ katsayılarına göre türevleri alınırsa;

$$\frac{\partial E_n}{\partial \alpha_i} = 0, \qquad i = 1,2,3,..,p \tag{3.17}$$

$$\frac{\partial E_n}{\partial \alpha_i} = \frac{\partial}{\partial \alpha_i} \sum_m \left( s_n[m] - \sum_{k=1}^{p} \alpha_k s_n[m-k] \right)^2 \tag{3.18}$$

$$\frac{\partial E_n}{\partial \alpha_i} = \sum_m \frac{\partial}{\partial \alpha_i} \left( s_n[m] - \sum_{k=1}^{p} \alpha_k s_n[m-k] \right)^2 \tag{3.19}$$

$$\frac{\partial E_n}{\partial \alpha_i} = \sum_m 2 \left( s_n[m] - \sum_{k=1}^{p} \alpha_k s_n[m-k] \right) \left( -\frac{\partial}{\partial \alpha_i} \sum_{k=1}^{p} \alpha_k s_n[m-k] \right) \tag{3.20}$$

$$\frac{\partial E_n}{\partial \alpha_i} = \sum_m 2 \left( s_n[m] - \sum_{k=1}^{p} \alpha_k s_n[m-k] \right) \left( -s_n[m-i] \right) \tag{3.21}$$

$$0 = \sum_m 2 \left( -s_n[m] s_n[m-i] + \sum_{k=1}^{p} \alpha_k s_n[m-k] s_n[m-i] \right) \tag{3.22}$$

LP katsayıları, kovaryans (covariance) metoduyla ya da otokorelasyon (autocorrelation) metoduyla bulunabilir. Kovaryans metodu kullanılırken incelenen aralığın içindeki değerler, dışındaki değerler kullanılarak bulunur. Otokorelasyon metodunda ise çerçevenin dışındaki değerlerin sıfır olduğu kabul edilir. Bu yüzden otokorelasyon analizi yapılacak dalga öncesinde pencerelenir. LP katsayıları sayısı p olmak üzere, otokorelasyon metoduyla elde edilen p tane denklem ve bilinmeyenden oluşan matris gösterim simetriktir ve toeplitz matris özelliğindedir ancak korelasyon metoduyla elde edilen matris gösterim simetrik olmasına rağmen toeplitz değildir. Toeplitz matrislerde köşegendeki ve bu köşegene paralel doğrultudaki elemanlar sabittir. Ayrıca otokorelasyon sonucu bulunan katsayılar ile tasarlanan filtrenin kararlı olacağı kesindir ancak korelasyon ile hesaplanan katsayılar için bu garanti yoktur. Bu çalışmadaki amaç, belirli parametrelerden gsm üzerinde bastırılmayacak seslerin tasarlanması olduğu için oluşturulacak filtrenin kararlı olması şarttır. Ayrıca toeplitz matrislerin çözümü için önerilmiş olan Levinson-Durbin algoritması hızlı ve efektif

bir yöntemdir. Donanım tasarlanması istenen bu sistemde otokorelasyon yönteminin ve levinson-durbin algoritmasının kullanılması tercih edildi.

### 3.2.4.1. Korelasyon Yöntemiyle Analiz Filtre Denklemlerinin Elde Edilmesi

Denklem (3.22) de verilen eşitlik düzenlenirse denklem (3.23) de verilen eşitlik elde edilir.

$$\sum_m s_n[m-i]s_n[m] = \sum_{k=1}^{p} \alpha_k \sum_m s_n[m-i]s_n[m-k] \qquad 1 \le i \le p \tag{3.23}$$

Denklem (3.24)'deki gibi bir $\Phi$ fonksiyonu tanımlanıp denklem (3.23)'e yerleştirildiğinde, daha sade bir gösterim olan denklem (3.25) elde edilir.

$$\Phi[i,k] = \sum_m s_n[m-i]s_n[m-k] \tag{3.24}$$

$$\sum_{k=1}^{p} \alpha_k \Phi[i,k] = \Phi[i,0], \qquad i = 1,2,3,...,p \tag{3.25}$$

Ortalama karesel hatayı ifade eden denklem (3.16) açılırsa, denklem (3.26) elde edilir.

$$\begin{aligned} E_n &= \sum_m \left( s_n[m] - \sum_{k=1}^{p} \alpha_k s_n[m-k] \right)^2 \\ E_n &= \sum_m s_n^2[m] - 2\sum_{m=-\infty}^{\infty} s_n[m] \sum_{k=1}^{p} \alpha_k s_n[m-k] + \sum_m \sum_{k=1}^{p} \alpha_k s_n[m-k] \sum_{l=1}^{p} \alpha_l s_n[m-l] \end{aligned} \tag{3.26}$$

Denklem (3.26) deki son terim, denklem (3.27) deki gibi düzenlenir ve denklem (3.24)'den faydalanılırsa denklem (3.28) elde edilir.

$$\begin{aligned} \sum_m \sum_{k=1}^{p} \alpha_k s_n[m-k] \sum_{l=1}^{p} \alpha_l s_n[m-l] &= \sum_{l=1}^{p} \alpha_l \left\{ \sum_{k=1}^{p} \alpha_k \sum_m s_n[m-k]s_n[m-l] \right\} \\ &= \sum_{l=1}^{p} \alpha_l \sum_m s_n[m-l]s_n[m] \end{aligned} \tag{3.27}$$

$$\begin{aligned} E_n &= \sum_{m=-\infty}^{\infty} s_n^2[m] - 2\sum_{k=1}^{p} \alpha_k \sum_{m=-\infty}^{\infty} s_n[m-k]s_n[m] + \sum_{l=1}^{p} \alpha_l \sum_{m=-\infty}^{\infty} s_n[m-l]s_n[m] \\ &= \sum_{m=-\infty}^{\infty} s_n^2[m] - \sum_{k=1}^{p} \alpha_k \sum_{m=-\infty}^{\infty} s_n[m-k]s_n[m] \end{aligned} \tag{3.28}$$

Denklem (3.25)'deki gösterim denklem (3.28) üzerinde uygulandığında ortalama karesel hata değeri korelasyon cinsinden bulunur.

$$E_n = \Phi_n[0,0] - \sum_{k=1}^{p} \alpha_k \Phi_n[0,k] \tag{3.29}$$

### 3.2.4.2. Otokorelasyon Yöntemiyle Analiz Filtre Denklemlerinin Elde Edilmesi

Otokorelasyon yönteminde seçilen N genişliğindeki zaman aralığının dışındaki örneklerin sıfır olduğu düşünülür ve ortalama karesel hata ±∞ aralığında hesaplanır. Bu işlem, 0 ≤ m ≤ N-1 olmak üzere s[n] konuşma dizisinin denklem (3.30)'da verilen N genişliğindeki w[m] dikdörtgen (rectangular) pencere ile pencerelenerek, pencerelenmiş işaretin sıfıra eşit olmayan ilk elemanının m=0 olacak şekilde kaydırılmasıdır.

$$s_n\,[m] = \begin{cases} s[m+n]w[m], & 0 \le \mathrm{m} \le \mathrm{N}-1 \\ 0 & diğer \end{cases} \tag{3.30}$$

Denklem (3.24)'de gösterilen korelasyon gösterimi, düzenlenerek otokorelasyon denklemleri elde edilmiştir.

$$\Phi_n[i,k] = \sum_{m=0}^{N_w-1-(i-k)} s_n[m] s_n[m+(i-k)], \quad 1 \le i \le p,\ 1 \le k \le p \tag{3.31}$$

Denklem (3.31)'deki $\Phi_n$ fonksiyonu sadece değişkenlerin farkına bağlı olduğu için bu eşitlik tek değişken ile gösterilebilir. Ayrıca otokorelasyon çift bir fonksiyondur.

$$r_n[\tau] = r_n[|\,i-k\,|] = \Phi_n[i,k] \tag{3.31}$$

P tahmin derecesi olmak üzere, denklem (3.31) ve denklem (3.24) birleştirildiğinde otokorelasyon özilişki katsayıları bulunur.

$$r_n[i] = \sum_{m=1}^{N-1-i} s_n[m] s_n[m+i] \tag{3.32}$$

Z domeninde yapılan otokorelasyon işlemi, zaman domeninde konvolüsyon işlemine denk gelir ve sinyalin kendine benzerliğinin bir ölçüsüdür. Otokorelasyon sonucunun i=0 anındaki çıkışı dalganın enerjisini verir ve otokorelasyonun alabileceği en büyük değerdir.

$$s_n[m] \text{ dalgasının enerjisi} = r_n[0] = \sum_{m=-\infty}^{\infty} \left|s_n[m]\right|^2 e^{i\theta} \tag{3.33}$$

P tahmin derecesi, $\alpha_k$ tahmin katsayıları olduğuna göre denklem (3.25) ve denklem (3.31) kullanılarak denklem (3.34) de verilen eşitlikler elde edilir.

$$\sum_{k=1}^{p} \alpha_k r_n[i-k] = r_n[i], \qquad 1 \le k \le p \tag{3.34}$$

Denklem (3.34) matris biçiminde gösterildiğinde denklem (3.35)'de verilen toeplitz otokorelasyon matrisi elde edilir. Levinson-Durbin algoritmasıyla bu matrisin çözümü yapıldığında LP tahmin katsayıları bulunur.

$$\begin{bmatrix} r_n[0] & r_n[1] & r_n[2] & \cdots & r_n[p-1] \\ r_n[1] & r_n[0] & r_n[1] & \cdots & r_n[p-2] \\ r_n[2] & r_n[1] & r_n[0] & \cdots & r_n[p-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_n[p-1] & r_n[p-2] & r_n[p-3] & \cdots & r_n[0] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} r_n[1] \\ r_n[2] \\ r_n[3] \\ \vdots \\ r_n[p] \end{bmatrix} \tag{3.35}$$

Denklem (3.29) ve denklem (3.31) kullanılarak ortalama karesel hatanın otokorelasyon sonuçları ile gösterimi bulunur.

$$E_n = r_n[0] - \sum_{k=1}^{p} \alpha_k r_n[k] \tag{3.36}$$

Otokorelasyon yönteminde, hata sadece pencerelenen aralıkta sıfır değildir. Pencerelenen aralığın dışındaki değerler sıfır olduğu için pencerenin en sağında ve en solunda, hata en büyük değerini alır ve buna kenar etkisi (edge effect) denir. Ortalama karesel hatanın, kenar hataları tarafından belirlemesinden kaçınmak için sinyal, otokorelasyon yöntemiyle incelenmeden önce denklem (3.2)'de verilen hamming penceresi ile pencerelenir.

### 3.2.5. Levinson-Durbin Algoritması

Levinson-Durbin algoritması, toeplitz matrislerin çözümünde kullanılan bilindik en popüler ve en etkili yöntemdir [11]. Çözüm, otokorelasyon matrisinin tersi alınarak $p^3$ tane çarpma ve toplama içeren işlemlerle yapılırken, levinson algoritması sayesinde sonuç $p^2$ işlemde bulunabilir. Bu algoritma kullanılarak denklem (3.35) de verilen

otokorelasyon matrisi çözülerek $a_1,\ a_2, \ldots, a_{10}$ vokal sistem filtre katsayıları bulunabilir. Levinson-Durbin algoritması aşağıda verilen dört adımın yinelenmesinden ve beşinci adımımdan oluşmaktadır:

***Adım 1(Başlangıç durumu):***

$$E(0) = R(0) \tag{3.37}$$

***Adım 2:***

$$k_i = \left[\mathrm{R(i)} - \sum_{\mathrm{j}=1}^{\mathrm{i}-1} a_{\mathrm{j}}^{(\mathrm{i}-1)} \mathrm{R(i-j)}\right] \Big/ \mathrm{E}^{(\mathrm{i}-1)}, \qquad 1 \leq \mathrm{i} \leq \mathrm{p} \tag{3.38}$$

***Adım 3:***

$$a_i = k_i \tag{3.39}$$

$$a_j^i = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, \qquad 1 \leq j \leq i-1 \tag{3.40}$$

***Adım 4:***

$$E^i = (1 - k_i^2) E^{(i-1)} \tag{3.41}$$

***Adım 5:***

(2,3,4). adımlar *i=1,2,…,p* için çözüldükten sonra denklem (3.42) de verilen işlem yapılarak sonuç bulunur.

$$a_i = a_j^p \tag{3.42}$$

Levinson-Durbin algoritmasında $E^i$ parametreleri, i. yineleme için öngörü hatasını vermektedir. Otokorelasyon matrisindeki *R(i)* değişkenleri yerine denklem (3.43) de verilen normalize otokorelasyon katsayıları kullanılırsa algoritma sonucu elde edilen $a_i$ katsayıları değişmez ve denklem (3.44) de verilen normalize öngörü hatası elde edilebilir [11].

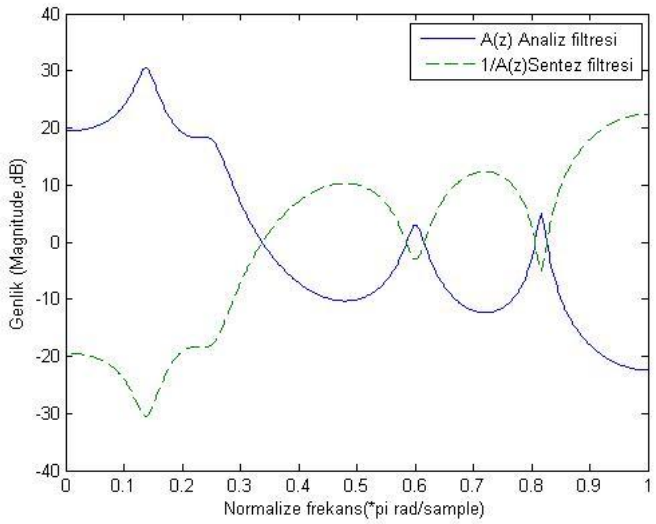
$$r(k) = R(k)/R(0) \tag{3.43}$$

$$V^{(i)} = \frac{E^{(i)}}{R(0)} = 1 - \sum_{i=1}^{i} a_k r(k), \qquad 0 < V^{(i)} \leq 1 \; ve \;\; 0 \leq i \tag{3.44}$$

Denklem 3.44, denklem 3.38 de elde edilen $k_i$ katsayıları ile gösterilebilir ve $k_i$ katsayıları denklem (3.45)'de verilen aralıkta bulunduğunda hesaplanan LP katsayıları ile tasarlanan filtre kararlı olur [11].

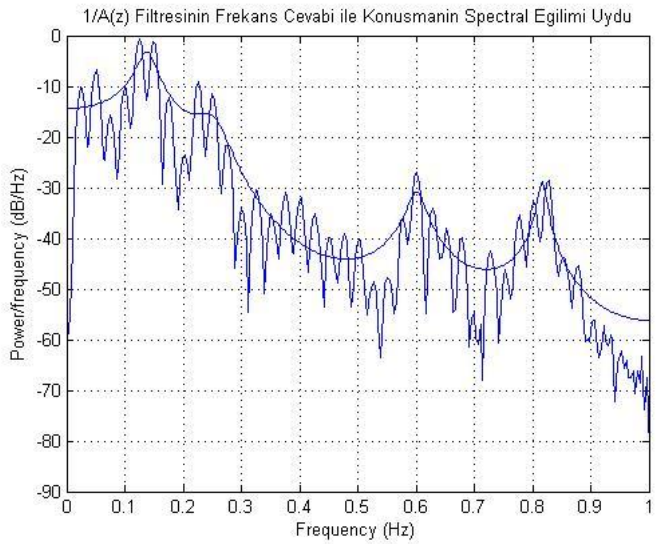
$$V^{(p)} = \prod_{i=1}^{p} (1 - k_i^2), \qquad -1 \leq k_i \leq 1 \tag{3.45}$$

Şekil 3.5 de verilen sesli konuşma dosyasının onuncu dereceden (p=10) analizi ile bulunan sentez ve analiz filtrelerinin frekans cevabı şekil 3.19 da bulunmaktadır.



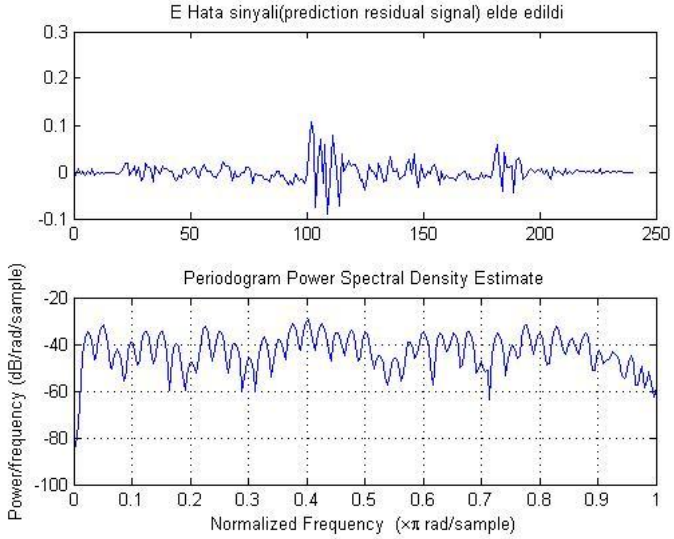

**Şekil 3.14:** Konuşma aralığına ait LPC sentez ve analiz filtreleri frekans yanıtları

Konuşmanın enerji dağılım grafiği ve sentez filtresinin frekans cevabı fonksiyonların spektral eğilimlerinin (envelope) uyuştuğu, şekil 3.16 da görülmektedir.




**Şekil 3.15:** Konuşma aralığına ait LPC sentez ve analiz filtreleri frekans yanıtları

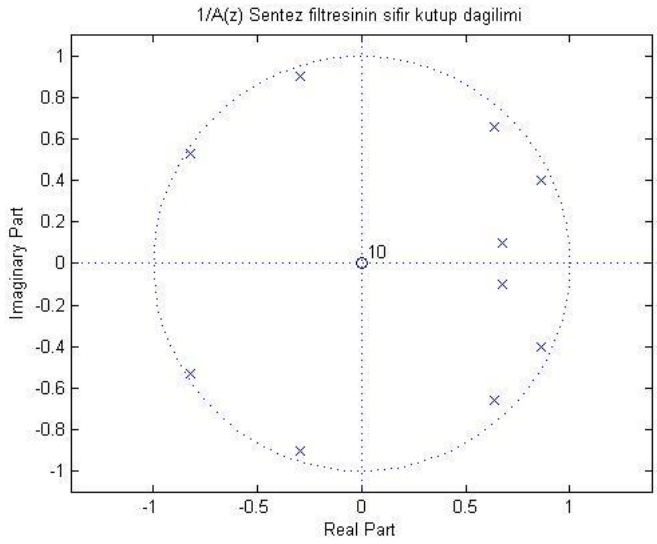
Analiz filtresi alçak geçiren bir filtre tipinde olması sebebiyle, analiz filtresinden geçirilen konuşma işaretinin sadece yüksek frekanslı bileşenleri kalır. Elde edilen hata sinyali şekil 3.17 de görülmektedir.




**Şekil 3.16:** Vokal filtre çıkışında görülen hata sinyali

Sentez filtresine hata işareti giriş olarak verilirse, konuşma sinyalinin kendisi geri elde edilir.

Sentez filtresinin z domeninde sıfır-kutup dağılımları incelendiğinde filtrenin kararlı bir yapıya sahip olduğu görülür.




**Şekil 3.17:** LPC sentez filtresinin sıfır-kutup dağılımı

### 3.2.6. Kazanç Hesabı

Sentezlenen ses ile analiz edilen sesin aynı enerjiye sahip olması için filtrenin girişine verilen sinyallerin belirli bir G kazancıyla çarpılması gerekir. (3.9) ve (3.13) denklemleri incelendiğinde kazancın, hata sinyaline göre belirlenebileceği görülmektedir. Bu eşitlik denklem (3.46)'da verilmiştir.

$$Gu[n] = s[n] - \sum_{k=1}^{p} a_k s[n-k] = e(n) \tag{3.46}$$

*G* kazanç sabitinin hata sinyali üzerinden belirlenmesi, kazancın bulunmasında çok güvenilir bir yol değildir [5]. Hata sinyalinin ve giriş sinyalinin enerjilerinin birbirine eşit olması gerektiği açıkça görülmektedir. Bu sayede kazanç, iki sinyalin enerjilerini eşitleyecek şekilde seçilir. Sonuç olarak kazanç, hata sinyalinin enerjisine bağlı olarak seçilir. Denklem (3.36)'dan yararlanıldığında kazancın otokorelasyon katsayıları cinsinden gösterimi de bulunabilir.

$$G^2 = E_n = r_n[0] - \sum_{k=1}^{p} \alpha_k r_n[k] \tag{3.47}$$

Şekil 3.20 de görülen -konuşma işareti, sentezleme filtre cevabı- grafiğinde filtre yanıtının, y ekseninde işaret ile aynı seviyeye gelmesi (dc ofset değerinin belirlenmesi) için filtre frekans yanıtının logaritması normalize hata enerjisinin kökü alınarak hesaplanan kazanç ile çarpılmıştır.

### 3.2.7. Enerjinin Normalize Değer Gösterimi

Normalize edilmiş ortalama karesel hatanın otokorelasyon yöntemine göre gösterimi denklem (3.48) deki gibi korelasyon gösterimine çevirilebilir.

$$V_n = \frac{\sum_{m=0}^{N+p-1} e_n(m)^2}{\sum_{m=o}^{N+p-1} s_n(m)^2} = \frac{\sum_{m=0}^{N-1} e_n(m)^2}{\sum_{m=o}^{N-1} s_n(m)^2} \tag{3.48}$$

Hata işareti denklem (3.49)'da tanımlanmıştır.

$$\alpha_1 = -1 \quad ise \quad e_n(m) = \sum_{k=0}^{p} \alpha_k s_n(m-k) \tag{3.49}$$

Denklem (3.49) ve denklem (3.48) birleştirilir ve denklem (3.24) deki gösterime göre denklemler düzenlenirse denklem (3.50) elde edilir.

$$V_n = \sum_{i=0}^{p}\sum_{j=0}^{p} \alpha_i \frac{\phi_n(i,j)}{\phi_n(0,0)} \alpha_j \tag{3.50}$$

Denklem (3.25) ile denklem (3.50) birleştirildiğinde ve denklem (3.31) 'e göre düzenlendiğinde denklem (3.51)'de verilen, hata enerjisinin normalize değerinin otokorelasyon ile ifadesi elde edilir.

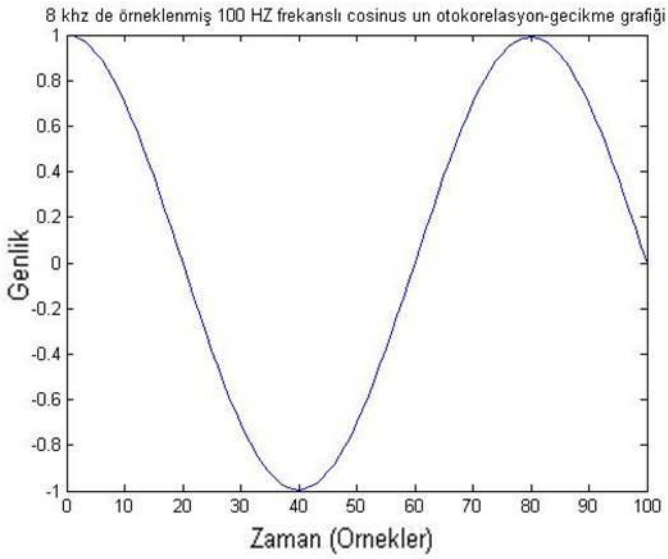
$$V_n = -\sum_{i=0}^{p} \alpha_i \frac{\phi_n(0,i)}{\phi_n(0,0)} = -\sum_{i=0}^{p} \alpha_i \frac{r_n(i)}{r_n(0)} \tag{3.51}$$

Hata sinyali için tanımlanan normalize enerji için başka bir gösterim de denklem (3.45) de verilmiştir.

### 3.2.8. Perdenin Tahmin Edilmesi

Sesli konuşmalardaki perde periyodunun belirlenmesi ortaya çıkan sesin kalitesinde çok önemli bir etkidir. Konuşmalardaki periyodikliğin mükemmel olmaması, konuşmadaki sesli kısmın başlangıcının tam olarak belirlenememesi ve gerçek hayattaki konuşmalara gürültünün ve yankının eklenmesi başlıca problemlerdir. Genellikle periyot tahmin hassaslığı arttıkça hesaplama yükü de artar.

Otokorelasyon, sinyalin kendine benzerliğinin bir ölçüsü olduğundan periyot tahmini için bu yöntemden yararlanılabilir. Periyodik bir sinyalde gecikme artırılarak otokorelasyon katsayıları hesaplanır. Otokorelasyon değerinin en yüksek olduğu gecikme, periyot bilgisini verir. Şekil 3.19'da periyot değeri seksen gecikmeye (100 hz lik frekansa sahip,8 khzde örnekleme hızında) eşit olan bir cosinüs işaretinin gecikme miktarına göre otokorelasyon grafiği verilmiştir.




**Şekil 3.18:** İşaretin otokorelasyon değerinin gecikme miktarına göre değişimi

Her bir gecikme değeri için otokorelasyon işlemi yapmak, ciddi bir işlem yükü doğuracağından perde tespiti için denklem (3.52)'de verilen genlik fark fonksiyonu (AMDF, Average Magnitude Difference Function) tercih edilir. Genlik fark fonksiyonu çarpma işlemi gerektirmediğinden özellikle donanım tasarımına çok daha uygundur ancak performans açısından iyi değildir.

$$AMDF[l] = \sum_{n=0}^{N-1} |S[n] - S[n-l]| \qquad (3.52)$$

İnsan sesi 50 Hz ile 200 Hz arasındadır, dolayısıyla 8 khz ile örneklenmiş bir konuşma işareti perde periyodu tahmini için incelenirken otokorelasyon ve genlik fark fonksiyonları gecikme miktarının (lag) 20 ve 150 arasındaki değerleri için kontrol edilerek işlem yükü azaltılabilir. Eğer periyodiklik bu değerlerin dışında ise incelenen sesin sessiz olduğuna karar verilebilir.

Şekil 3.17 da görüldüğü gibi periyodik sinyallerin hata sinyalleri de periyodiktir. Hata sinyalleri incelenerek periyot bilgisi öğrenilebilir.

### 3.3. LPC Modelinin Matlab Ortamında Tasarlanması

Bu çalışmada LPC analiz ve sentez modeli matlab ortamında tasarlanmıştır ve örnek bir konuşmanın analizi ve sentezi yapılmıştır. Yapılan işler maddeler halinde aşağıda verilmiştir:

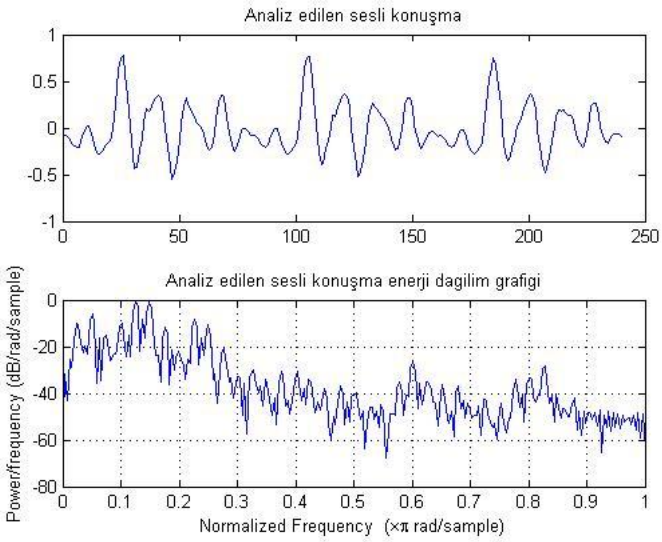
***Analiz Kısmında:***

- Konuşma 8 khz de darbe kodlayıcısı (PCM) ile örneklendi.
- 10ms üst üste örtüşecek (overlap) şekilde 20 ms lik çerçevelere ayrılarak sırasıyla aşağıdaki işlemlerden geçirildi.
- Hamming penceresi ile pencerelendi.
- Seslilik detektörü (VAD, voice activity dedector) sayesinde incelenen çerçeve sesli veya sessiz olarak ayrıştırıldı.
- Otokorelasyon katsayıları oluşturuldu.
- Levinson-Durbin algoritmasıyla LPC tahmin katsayıları hesaplandı.
- Kazanç hesaplandı.
- Sesli çerçeveler için periyot bilgisi bulundu.

**Sentez Kısmında:**

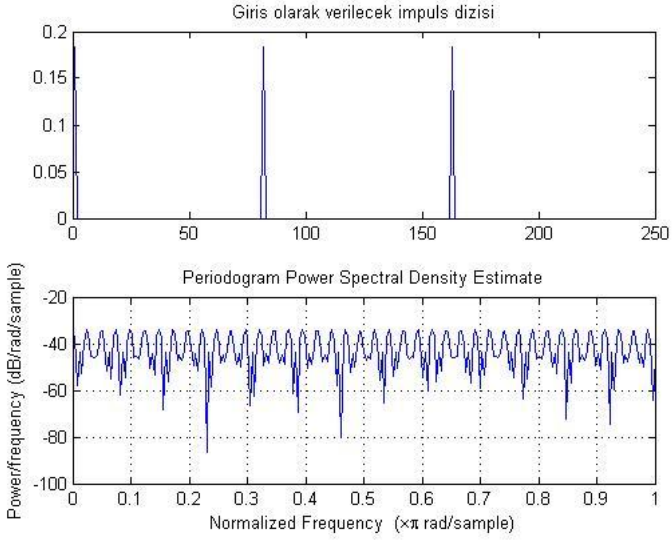
- Konuşma seslilik bilgisi incelenerek 10 ms uzunluğunda uyarı (excitation) sinyali oluşturuldu:
  - Sessiz ise beyaz gürültü,
  - Sesli ise konuşmanın periyot bilgisine bakılarak darbe dizisi oluşturuldu.
- Oluşturulan çerçeve kazanç ile çarpıldı.
- Elde edilen sinyal, LPC tahmin katsayıları ile tasarlanan 1/A(z) sentezleme filtresine giriş olarak verilerek sentetik ses oluşturuldu.

Şekil 3.20 ve şekil 3.23 de analiz edilen ve sentezlenen sesli konuşma çerçevelerinin ve enerji yoğunluk fonksiyon grafikleri bulunmaktadır.



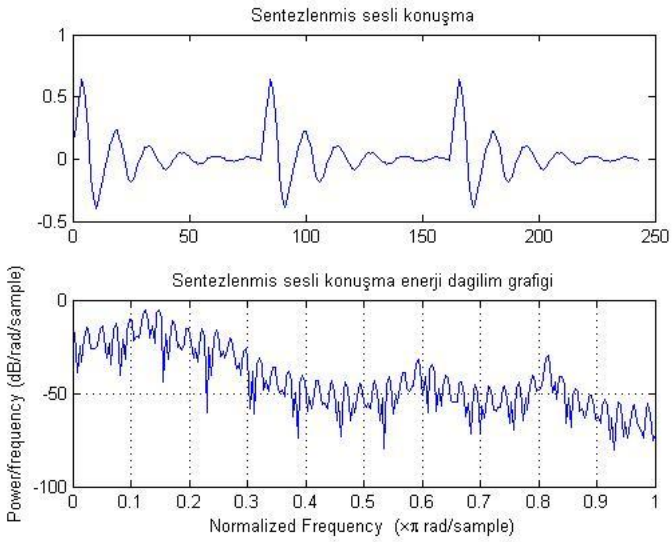

**Şekil 3.19:** Analiz edilen sesli konuşma

Şekil 3.21 de sesli konuşmanın sentezlenmesi için filtrenin girişine verilecek uyarı sinyali ve enerji yoğunluk fonksiyonu grafiği bulunmaktadır. Uyarı sinyalinin hata sinyaline benzemediği açıkça görülmektedir ancak bu iki sinyalin enerji spektrumlarının sınırlarında benzerlik vardır.



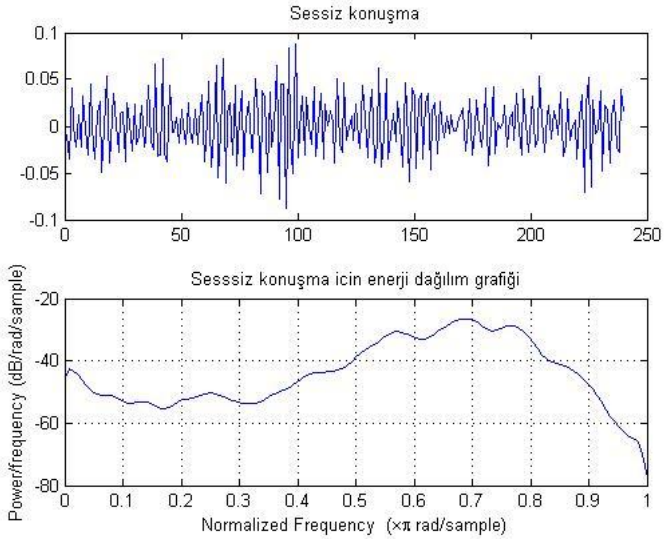

**Şekil 3.20:** Analiz edilen ses ile aynı periyodikliğe sahip kazanç ile çarpılmış darbe dizisi

Sentezlenen ve analiz edilen sinyallerin, dalga yapılarında ciddi bir farklılık vardır. Bunun en büyük sebebi LP analizinin faz spektrumunu hesaplamamasıdır. Enerji spektrumları incelendiğinde iki işaretin birbirine çok yakın olduğu görülür. Sentezlenen konuşma, incelenen konuşmaya göre fazla harmonik bir yapıya sahiptir.



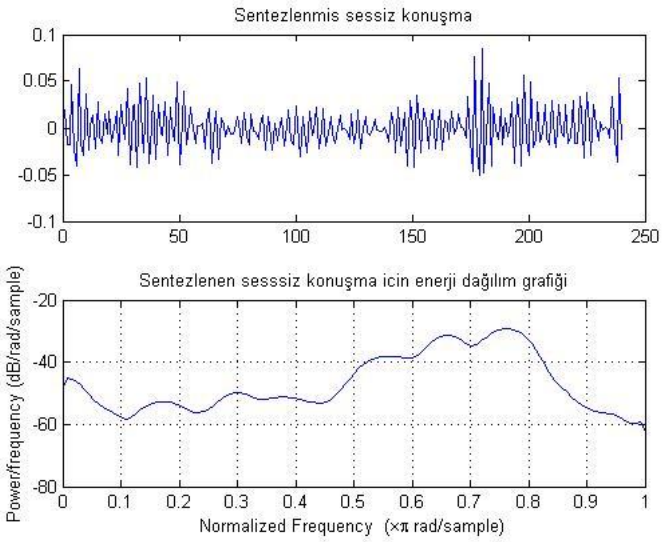

**Şekil 3.21:** Sentezlenen sesli konuşma

Şekil 3.27 ve şekil 3.28 de analizi yapılan ve sentezlenen sessiz konuşmalar ve ortalama enerji yoğunluk fonksiyon grafikleri bulunmaktadır.



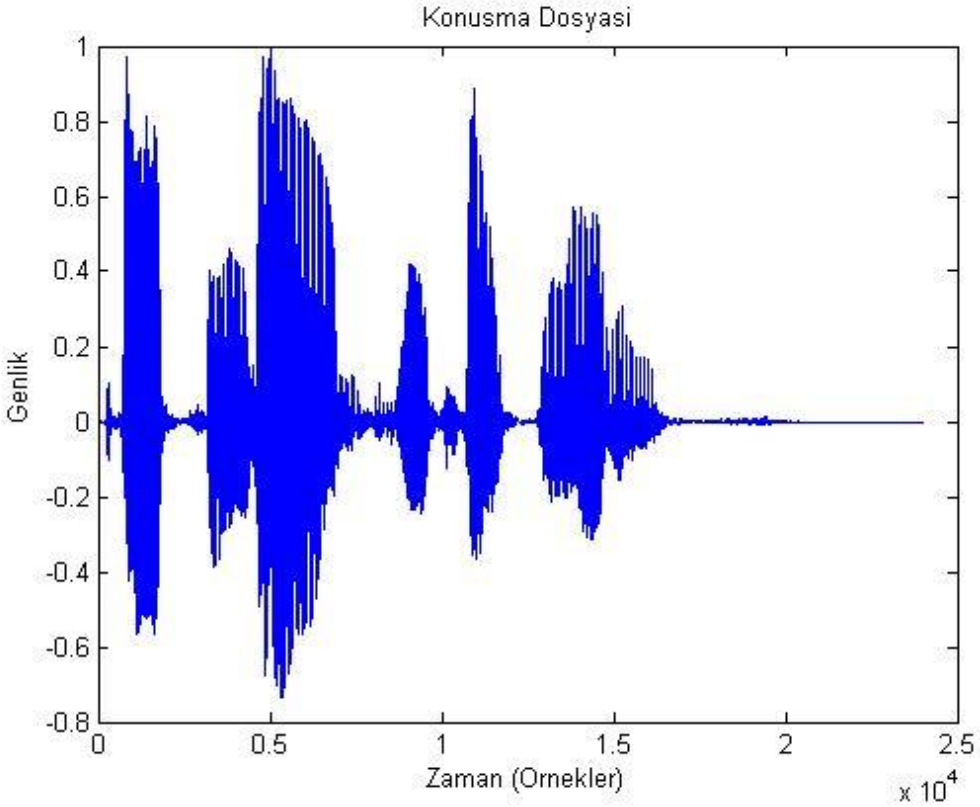

**Şekil 3.22:** Analizi yapılan sessiz konuşma ve ortalama enerji yoğunluk fonksiyon grafiği

Analizi yapılan ve sentezlenen sessiz konuşmalar karşılaştırıldığında hiçbir ortak örneklerinin olmadığı görülmektedir. Bunun yanında dalgaların spektral eğilimleri birbirine çok yakındır.

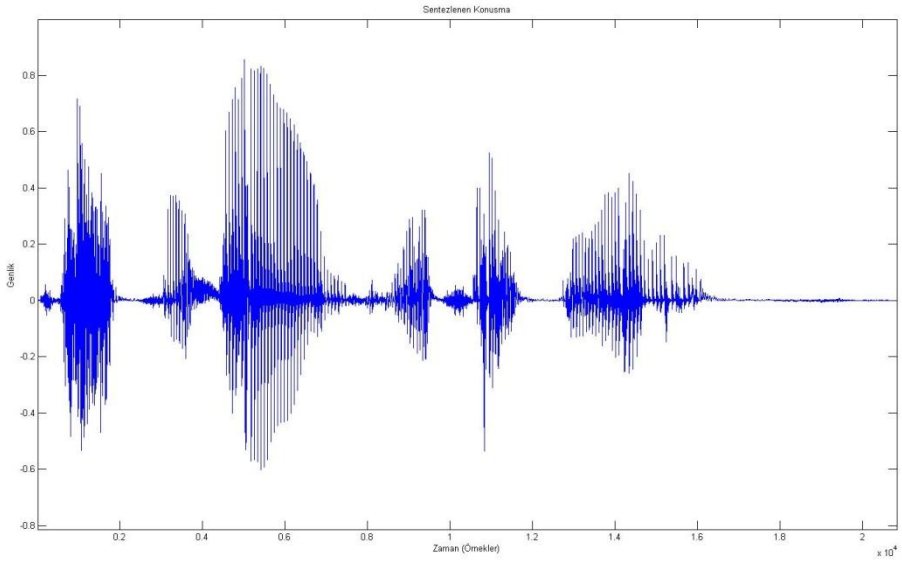



**Şekil 3.23:** Analizi yapılan sessiz konuşma ve ortalama enerji yoğunluk fonksiyon grafiği

Şekil 3.29 de ve şekil 3.30 da LP yöntemiyle incelenen ve sentezlenen konuşmalar verilmiştir.

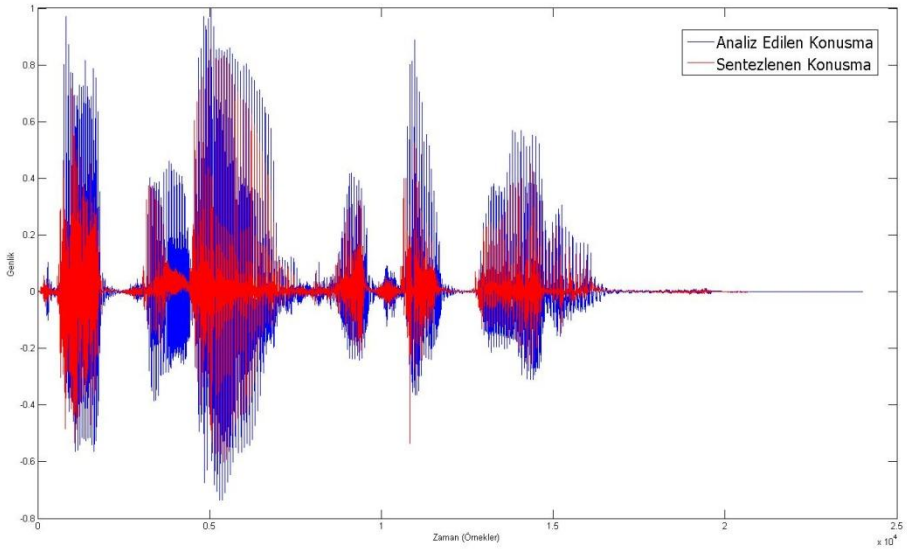



**Şekil 3.24:** Analizi yapılan Konuşma




**Şekil 3.25:** Sentezlenen Konuşma

Sıkıştırma işlemi sonucunda elde edilen konuşmanın orijinal konuşmaya göre daha seyrek, ortalama olarak daha düşük genlikli olduğu ancak genliğinin yüksek inişlere ve çıkışlara sahip olduğu bölgelerde daha yüksek genlikler elde edildiği gözlemlendi.




**Şekil 3.26:** Analizi yapılan konuşma ve Sentezlenen Konuşma

Bu bölümde LPC analiz ve sentez yöntemleri üzerinde duruldu, sesli ve sessiz konuşmaların arasındaki temel farklardan bahsedildi. Konuşmanın kodlanmasından ve kodlama hatalarından bahsedilmedi. Bölüm 5 de konuşmanın kodlanmasından işimize yarayacağı kadarıyla bahsedilecektir.

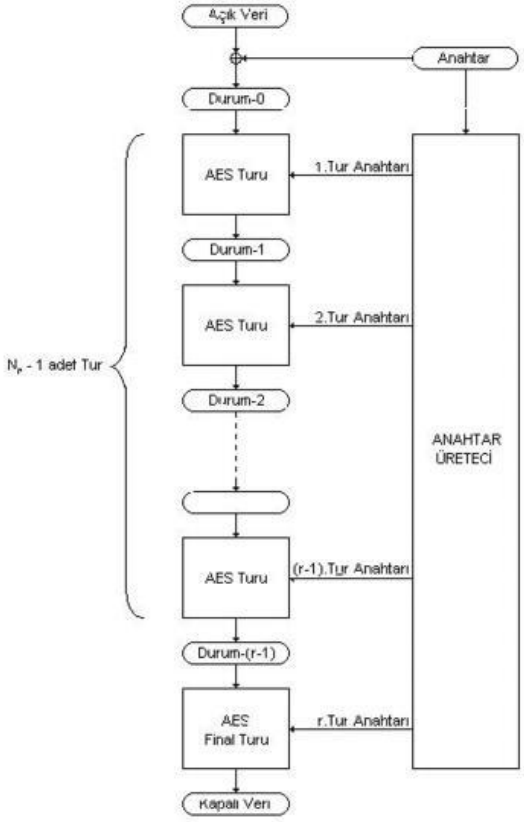
# 4. GELİŞMİŞ ŞİFRELEME STANDARDI

Gelişen teknoloji ve artan işlemci hızları karşısında güvenilebilirliğini yitirmeye başlayan veri Kodlama Standardının (DES: Data Encryption Standart [7]) yerini alması için, ocak 1997'de ABD Ulusal Standartlar ve Teknolojiler Enstitüsü (NIST) yeni bir şifreleme standardının geliştirilmesi amacıyla bir yarışma düzenledi. Yeni standardın 128 bitlik blok boyunu ve 128, 192 veya 256 bitlik anahtar uzaylarını destekleyen bir simetrik blok şifreleyici olması talep ediliyordu. NIST, Joan Daemen ve Vincent Rijmen tarafından tasarlanan, Rijndael algoritmasının hiçbir değişiklik talep edilmeden Gelişmiş Kodlama Standardı (AES: Advanced Encryption Standard) olarak kullanılacağını ilan etti. Rijndael algoritmasında blok ve anahtar uzunlukları 128 bitten 256 bite kadar 32'lik araklıklarla birbirinden bağımsız olarak değişebildiği halde, AES yukarıda belirtilen sabitlenmiş blok ve anahtar uzunluklarıyla kullanılmaktadır [14].

GSM üzerinden güvenli ses iletimi için, sayısallaştırılan konuşmalar AES-128 şifreleme standardı ile şifrelenecektir. Şifreleme standardı, FPGA üzerinde Levent Ordu'nun [14] nolu çalışmasından faydalanılarak gerçeklenmiştir.

Bir algoritmada tekrarlanarak yürütülen işlemlerin oluşturduğu yapıya tur denir. AES-128 bit şifreleme işlemi 10 turdan oluşur. Algoritma şifreleme ve şifre çözme olmak üzere iki temel bloktan oluşur. Her turda şifre anahtar üreteci tarafından değiştirilir. Şifreleme ve şifre çözme blokları; bayt değiştirme (Sub Bytes), satırları kaydırma (Shift Rows), sütun karıştırma (Mix Columns), tur anahtarı ekleme (Add Round Key) işlemlerini ve ters işlemlerini yapan dört alt bloktan oluşmaktadır [15].

İlk anahtar toplamasında, algoritmaya girilen veri ve ana anahtarın karşılıklı bitleri arasında özel veya (XOR) işlemi uygulanır ve 16 bayttan oluşan ilk durum elde edilir ve bu durum ilk turda işlenir. İlk tur çıkışı, ikinci turun girişini oluşturur. Bu şekilde N-1 kez tur'da işlenen veri son olarak final turunda işlenir ve algoritma çıkışı elde edilir [14]. Şekil 4.1 de bu işlem gösterilmiştir.



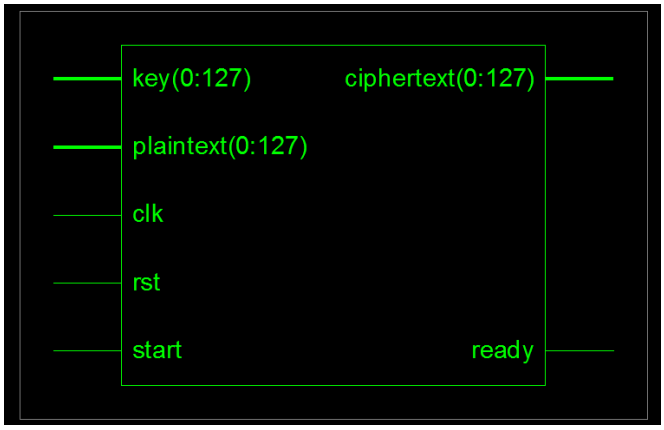

**Şekil 3.3 :** AES şifreleyici blok diyagramı [14].

Şifreleme işlemi için fips-197 standardında verilen sözde kod (Pseudo Code) şekil 4.2 de verilmiştir [15].

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[0, Nb-1])                  // See Sec. 5.1.4

   for round = 1 step 1 to Nr-1
      SubBytes(state)                              // See Sec. 5.1.1
      ShiftRows(state)                             // See Sec. 5.1.2
      MixColumns(state)                            // See Sec. 5.1.3
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   end for

   SubBytes(state)
   ShiftRows(state)
   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   out = state
end
```

**Şekil 3.3 :** AES şifreleyici sözde kodu [15].

Şifreleyici donanım şematik gösterimi şekil 4.3 de verilmiştir.



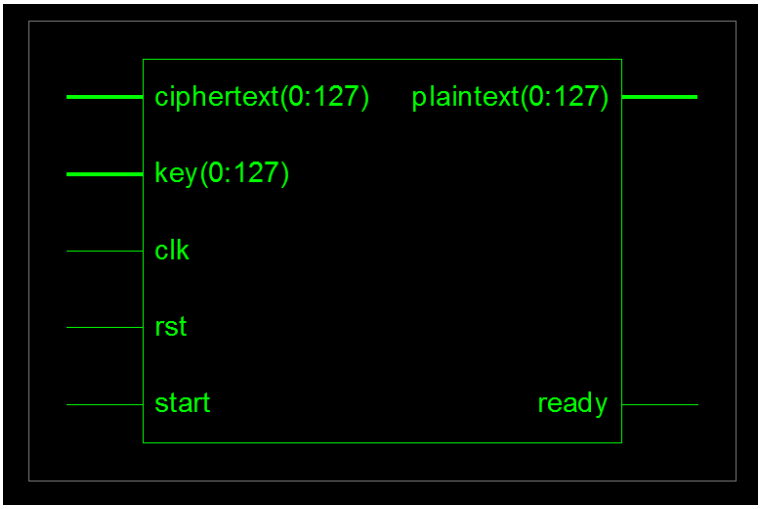

**Şekil 3.3 :** AES şifreleyici donanım şematik gösterimi.

Şifre çözme işlemi için fips-197 standardında verilen sözde kod (Pseudo Code) şekil 4.4 de verilmiştir [15].

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1]) // See Sec. 5.1.4

   for round = Nr-1 step -1 downto 1
      InvShiftRows(state)                    // See Sec. 5.3.1
      InvSubBytes(state)                     // See Sec. 5.3.2
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
      InvMixColumns(state)                   // See Sec. 5.3.3
   end for

   InvShiftRows(state)
   InvSubBytes(state)
   AddRoundKey(state, w[0, Nb-1])

   out = state
end
```

**Şekil 3.3 :** AES şifre çözücü sözde kodu [15].

Şifre çözücü donanım şematik gösterimi şekil 4.5'de verilmiştir.

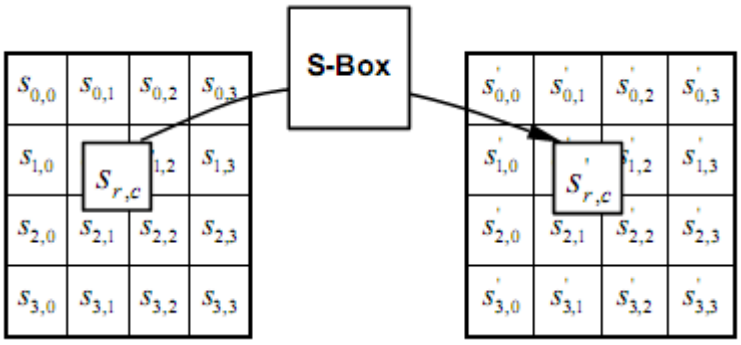



**Şekil 3.3 :** AES şifre çözücü donanım şematik gösterimi.

Şifreleme ve şifre çözme işlemleri için tasarlanması gereken alt bloklar aşağıda açıklanmıştır.

### 4.1. Bayt Değiştirme (Sub Bytes)

Bayt değiştirme işlemi sırasında 128 bitlik veri 8 bitlik 16 parçaya bölünür ve şekil 4.6'da gösterilen 4x4 boyutundaki durum matrisi oluşturulur. Girişteki durumun her bir baytı S-kutusu (S-Box) adı verilen değiştirme tablosu kullanılarak farklı baytlara dönüştürülür. Şekil 4.4'de gösterilen bayt değiştirme işlemi, matematiksel olarak giriş baytının indirgeme polinomu kullanılarak çarpmaya göre tersinin alınması işlemi ve elde edilen sonucun geçiş matrisi olarak adlandırılan bir matrisle çarpılmasıyla bayt değiştirme işleminin yapılmasıdır.

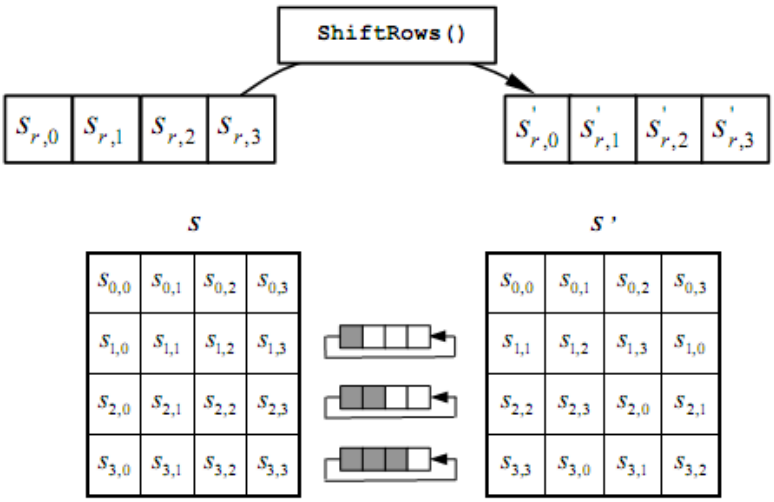



**Şekil 4.1:** Durum Matrisi [15].

Ters bayt değiştirme işlemi için aynı model ters S-kutusu kullanılarak yapılır.

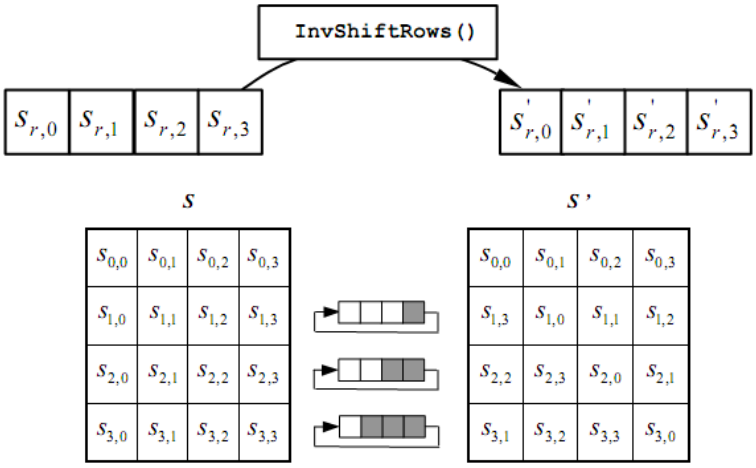
## 4.2. Satırları Kaydırma (Shift Rows)

Satır Kaydırma işleminde, durum matrisinin ilk satırı sabit kalır, ikinci satırı bir, üçüncü satırı iki, dördüncü satırı ise üç kere sola kaydırılır. Şekil 4.7'de verilen satır kaydırma işlemi sonucunda elde edilen durum matrisindeki elemanlar birleştirilerek 128 bitlik veri elde edilir.




**Şekil 4.1:** Satırları Kaydırma İşlemi [15].

Satırları kaydırma işleminin tersi şekil 4.8'de görüldüğü gibi, yukarıda yapılan işlemlerin tersinin yapılmasıdır.



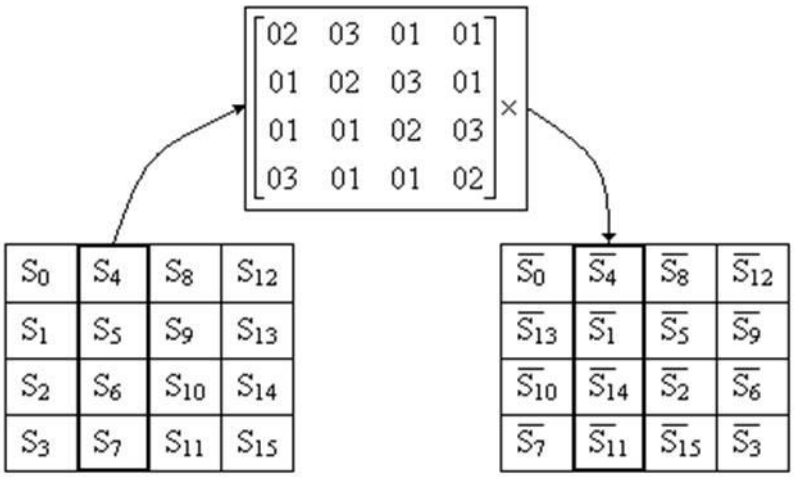

**Şekil 4.1:** Satırları Kaydırma İşleminin Tersi [15].

## 4.3. Sütunları Karıştırma (Mix Columns)

Sütunları karıştırma işlemiyle giriş durum matrisinin her sütunu şekil 4.9'da gösterilen işlemden geçirilerek, çıkış durum matrisi elde edilir. Her bir sütun

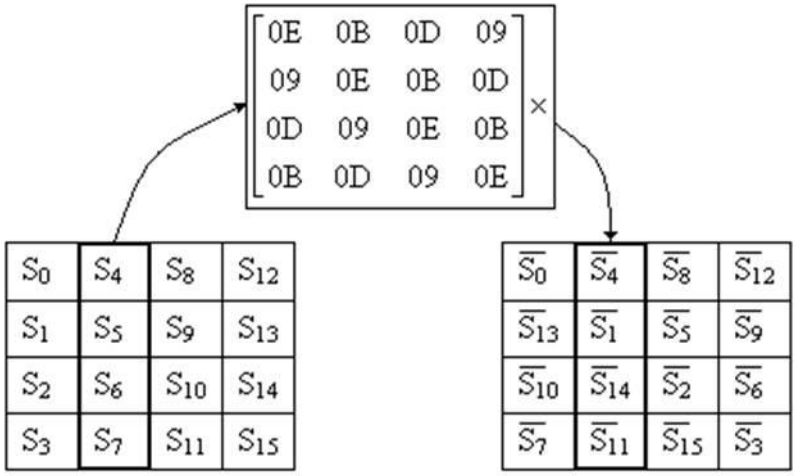
denklem (4.1) de verilen A(x) polinomu ile modülo ($x^4$ + 1)'de çarpma işlemi gerçekleştirilir.

$$A(x) = (03)x^3 + (01)x^2 + (02) \quad (4.1)$$




**Şekil 4.1:** Sütunları Kaydırma İşlemi [14].

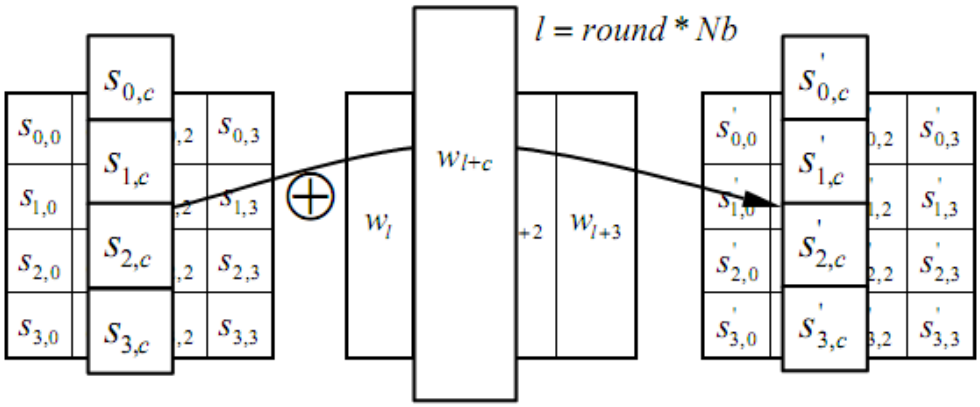
Şifrenin çözülmesi sırasında aynı işlem ters denklem için uygulanır.




**Şekil 4.1:** Sütunları Kaydırma İşleminin Tersi [14].

### 4.4. Tur Anahtarı Ekleme

Durum matrisinin baytları ile tur anahtarında karşılık düşen baytlar arasında özel veya (XOR) işlemi yapılır. Şifreleme ve şifre çözme işlemleri için bu aşama aynıdır.




**Şekil 4.1:** Tur Anahtarı Ekleme İşlemi [14].

### 4.5. Anahtar Üreteci

Anahtar üreteci, her turun son adımında üretilen ara değer ile toplanacak olan tur anahtarlarını üretmektedir. Şifreleme veya şifre çözme esnasında tur sayısı kadar tur anahtarı üretilir. 128 bit uzunluğundaki tur anahtarlarının şifreleme işlemi için üretilmesi şekil 4.12'de verilen algoritma ile sağlanır.

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
   word  temp

   i = 0

   while (i < Nk)
      w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
   end while

   i = Nk

   while (i < Nb * (Nr+1)]
      temp = w[i-1]
      if (i mod Nk = 0)
         temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
      else if (Nk > 6 and i mod Nk = 4)
         temp = SubWord(temp)
      end if
      w[i] = w[i-Nk] xor temp
      i = i + 1
   end while
end
```

**Şekil 4.1:** Şifreleme işlemi için anahtar üreteci [14].

Şifre çözme işlemi için, şekil 4.13'de verilen algoritma ile anahtar üreteci tasarlanır.

```
EqInvCipher(byte in[4*Nb], byte out[4*Nb], word dw[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, dw[Nr*Nb, (Nr+1)*Nb-1])

   for round = Nr-1 step -1 downto 1
      InvSubBytes(state)
      InvShiftRows(state)
      InvMixColumns(state)
      AddRoundKey(state, dw[round*Nb, (round+1)*Nb-1])
   end for

   InvSubBytes(state)
   InvShiftRows(state)
   AddRoundKey(state, dw[0, Nb-1])

   out = state
end


For the Equivalent Inverse Cipher, the following pseudo code is added at
the end of the Key Expansion routine (Sec. 5.2):
   for i = 0 step 1 to (Nr+1)*Nb-1
      dw[i] = w[i]
   end for

   for round = 1 step 1 to Nr-1
      InvMixColumns(dw[round*Nb, (round+1)*Nb-1])     //  note  change  of
type
   end for
```

**Şekil 4.1:** Şifre çözme işlemi için anahtar üreteci [14].

### 4.6. Gelişmiş Şifreleme Standardının FPGA ile Gerçeklenmesi

Gerçeklenen şifreleme ve şifre çözme donanımlarının sentez raporu şekil 4.14'de ve şekil 4.15'de verilmiştir. Ayrıca donanımların zamanlama rapor sonuçları tablo 4.1 de verilmiştir.

| | Minimum Periyot | Maksimum Saat Frekansı |
|---|---|---|
| Şifreleme Donanımı | 10.515ns | 95.098MHz |
| Şifre Çözme Donanımı | 10.389ns | 96.256MHz |

**Tablo 4.1:** Donanımların Zamanlama Rapor Sonuçları

| AES Project Status | | | |
|---|---|---|---|
| **Project File:** | aes.ise | **Current State:** | Synthesized |
| **Module Name:** | cipher_table_serial | • **Errors:** | No Errors |
| **Target Device:** | xc3s1600e-4fg320 | • **Warnings:** | 8 Warnings |
| **Product Version:** | ISE 9.2i | • **Updated:** | Mon Jun 6 00:19:47 2011 |

| AES Partition Summary |
|---|
| No partition information was found. |

| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** |
| Number of Slices | 2722 | 14752 | 18% |
| Number of Slice Flip Flops | 475 | 29504 | 1% |
| Number of 4 input LUTs | 5031 | 29504 | 17% |
| Number of bonded IOBs | 388 | 250 | 155% |
| Number of GCLKs | 1 | 24 | 4% |

**Şekil 4.1:** AES Şifreleme Donanımının Sentez Raporu.

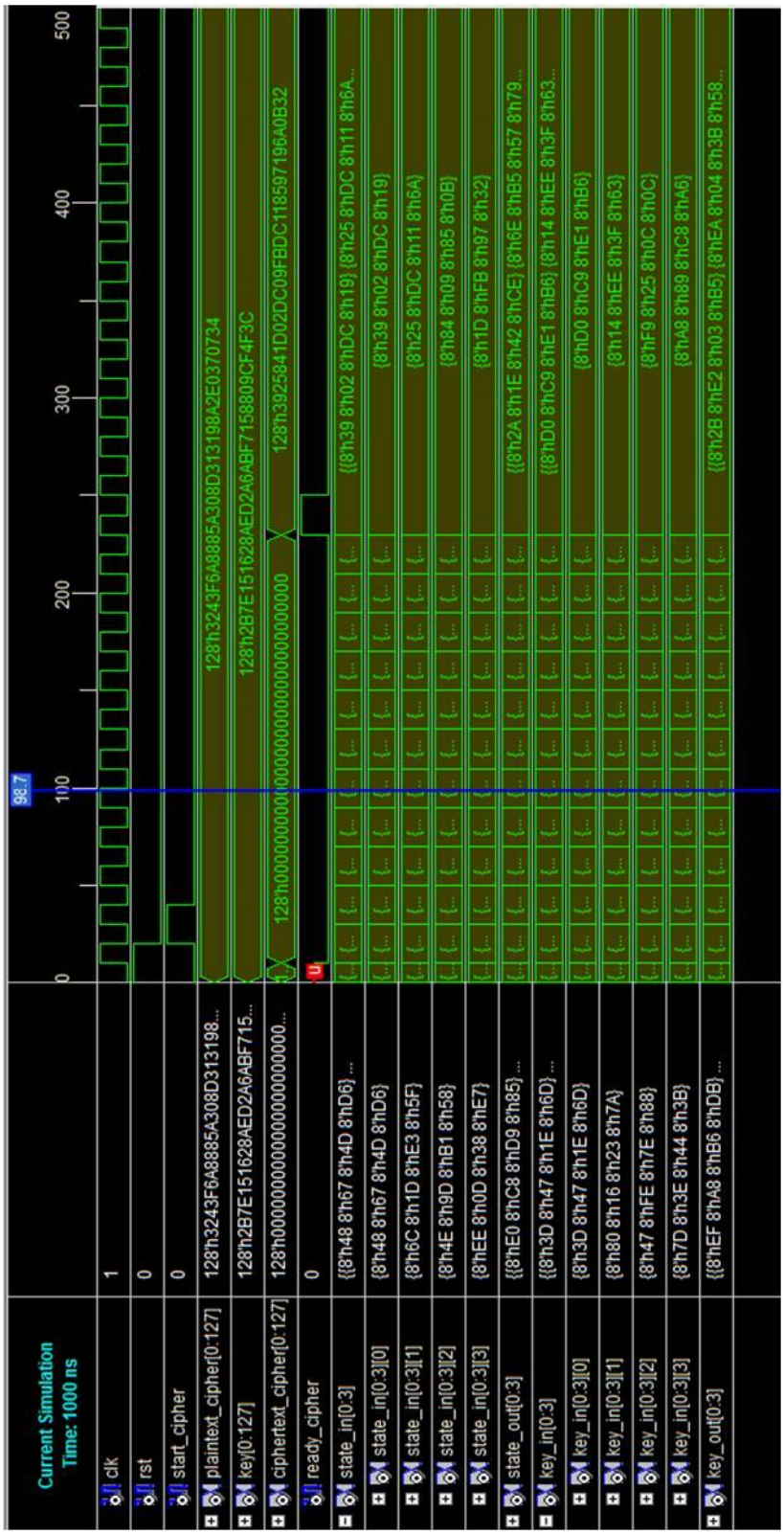
| AES Project Status | | | |
|---|---|---|---|
| **Project File:** | aes.ise | **Current State:** | Synthesized |
| **Module Name:** | invcipher_table_serial | • **Errors:** | No Errors |
| **Target Device:** | xc3s1600e-4fg320 | • **Warnings:** | 14 Warnings |
| **Product Version:** | ISE 9.2i | • **Updated:** | Mon Jun 6 01:09:03 2011 |

| AES Partition Summary |
|---|
| No partition information was found. |

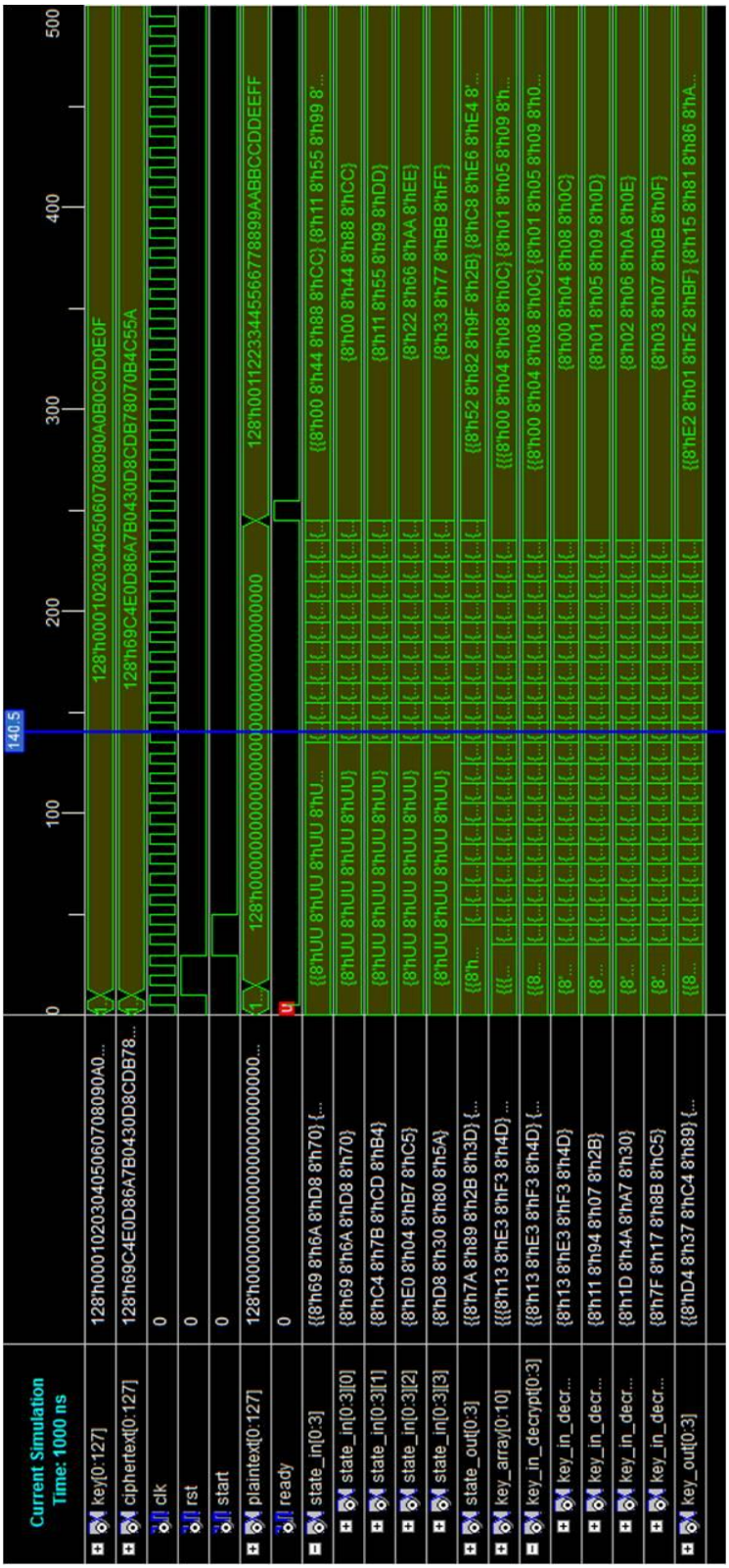
| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** |
| Number of Slices | 4162 | 14752 | 28% |
| Number of Slice Flip Flops | 2083 | 29504 | 7% |
| Number of 4 input LUTs | 7931 | 29504 | 26% |
| Number of bonded IOBs | 388 | 250 | 155% |
| Number of BRAMs | 20 | 36 | 55% |
| Number of GCLKs | 1 | 24 | 4% |

**Şekil 4.1:** AES Şifre Çözme Donanımının Sentez Raporu.

Gerçeklenen şifreleme ve şifre çözme donanımlarının benzetim sonuçları şekil 4.16'da ve şekil 4.17'de verilmiştir.



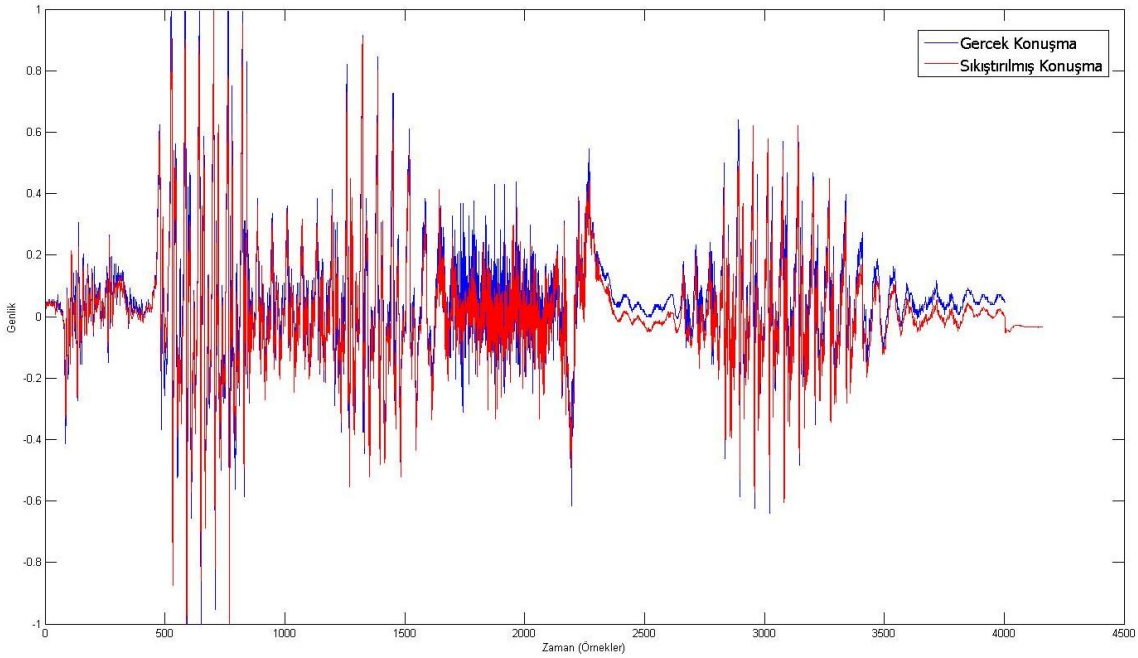

**Şekil 4.1:** AES Şifreleme Donanımının Benzetim Sonucu [14].




**Şekil 4.1:** AES Şifre Çözme Donanımının Benzetim Sonucu [14].

# 5. KODLAYICI TASARIMI İÇİN KONUŞMANIN KODLANMASI

GSM kodlayıcıda, konuşma modellenerek iletildiği için iletilen konuşma gerçek konuşmadan farklı olur. Şekil 5.1 de görüldüğü gibi GSM kodlayıcı gerçek konuşmaya karakteristik olarak benzeyen ancak örnek değerleri açısından farklı bir konuşma sentezlemiştir.




**Şekil 5.27:** GSM kodlayıcıya gönderilen ve alınan konuşmalar

Sayısal bir veriden konuşma sentezlenip, GSM kodlayıcıdan geçtikten sonra minimum hata ile sayısal verinin elde edilebilmesi için konuşma, dalga formu olarak değil model parametreleri tabanlı bir yöntem ile sentezlenmelidir. Konuşmaya ait en temel özellikleri ifade eden parametreler LPC filtre katsayıları, enerji ve perde periyodudur.

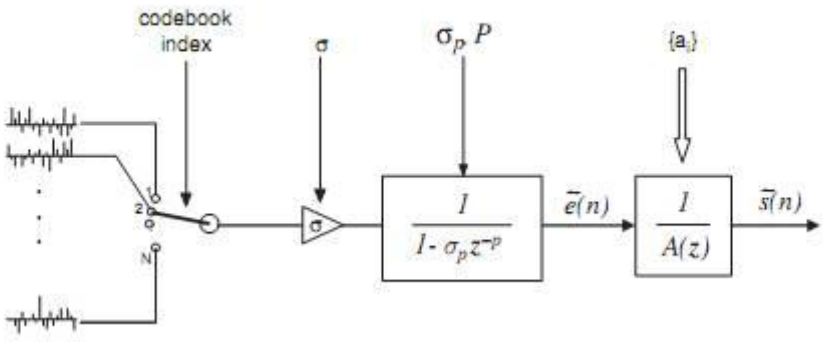
Güvenli ses iletiminin başarılabilmesi için sayısal verilere karşılık düşen konuşmaların kodlayıcı tarafından sıkıştırıldıktan sonra yeniden tanınabilmesi için kabul edilebilir bir hata oranının üzerinde değişmemesi gerekir. Bu tip konuşmaların elde edilebilmesi için GSM kodlayıcının yapısı araştırıldı ve uygun konuşmalara ait özellikler belirlendi. NTIMIT veritabanı içerisindeki konuşmalar taranıp istenilen

koşulları sağlayan çerçeveler seçildi ve bu konuşmalara ait LPC parametreleri elde edildi. Sistemin hatasız çalışabilmesi için bulunan parametreler LBG algoritması ile sınıflara ayrılıp, sayısal verilerden konuşmaların kodlanmasına imkan verecek kod kitapları oluşturuldu. Bu kod kitapları sayesinde kodlayıcının girişindeki ve çıkışındaki konuşmalar farklı olsalar bile sistem tarafından doğru algılanılacaktır.

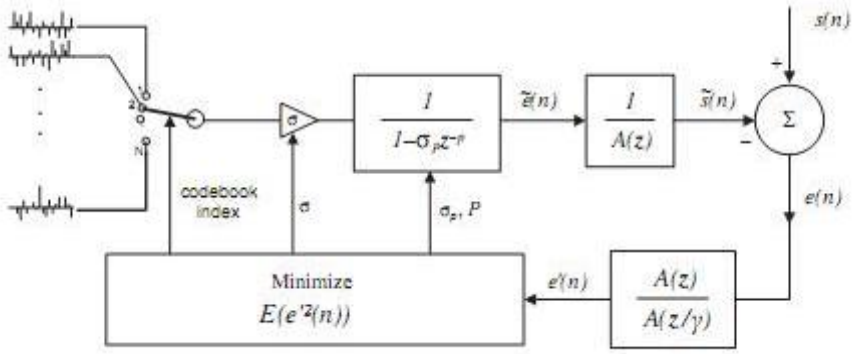
### 5.1. GSM FR Kodlayıcısı

GSM-FR kodlayıcısı Avrupa Telekomünikasyon Standartları Enstitüsü (European Telecommunications Standards Institute, ETSI) tarafından mobil cep telefonu uygulamaları için standartlaştırılan 8 khz örnekleme frekanslı ve 13 kpbs bit oranına sahip bir konuşma sıkıştırma algoritmasıdır. Bu kodlayıcı CELP (Code Excited Linear Prediction) algoritmalarından biri olan RPE-LTP (Regular Pulse Excitation-Long Term Prediction-Linear Predictive coding) tabanlı bir sıkıştırma tekniğine sahiptir.

CELP algoritması, analiz edilerek sentezleme (analysis by synthesis) fikri kullanılarak hata işaretine vektörel kuantalama (vector quantization) yoluyla ifade eden bir sıkıştırma şeklidir. Bu algoritma ile hata işareti daha iyi modellenir ve bu sayede daha doğru bir konuşma sentezlenir. Bu yaklaşımda uyarma sinyali, içerisinde olası uyarma sinyallerine ait bilgilerin bulunduğu önceden tanımlanmış rasgele kod kitabından (stochastic codebook) bir dizi seçilerek oluşturulur. Ayrıca LTP (Long Term Prediction) analizi ile hata sinyalinin önceki değerleri geciktirilerek konuşmaya ait bazı bilgiler seçilir ve adaptif olarak değişen bir kod kitabı oluşturulur. Sistem bu iki kod kitaplarındaki bilgilere göre konuşmalar sentezler ve gerçek sinyal ile karşılaştırarak en uygununu bulur ve indeks numarasını kodlar. Sisteme ait kodlama ve kod çözme blok şeması şekil 5.2'de bulunmaktadır.

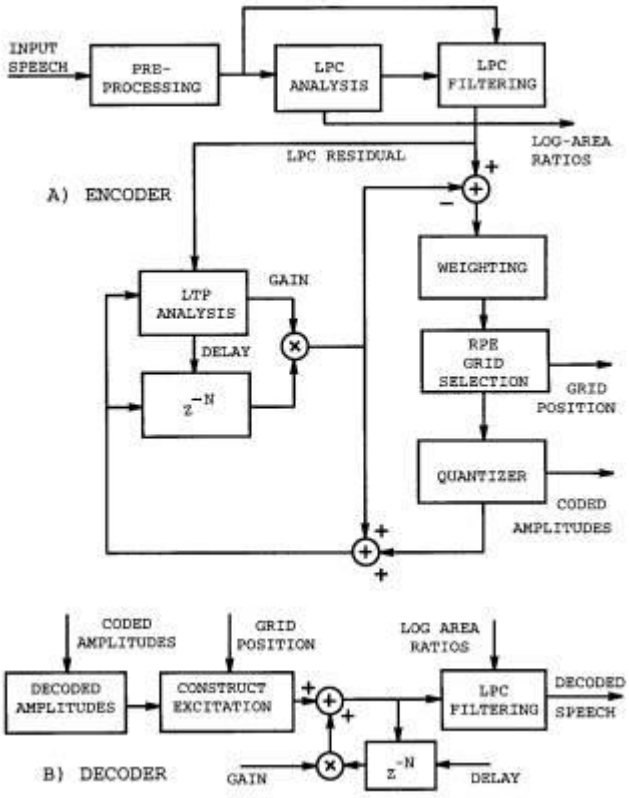


**Şekil 5.28:** Celp algoritmasının kodlayıcı kısmı [16].




**Şekil 5.29:** Celp algoritmasının kodlayıcı kısmı [16].

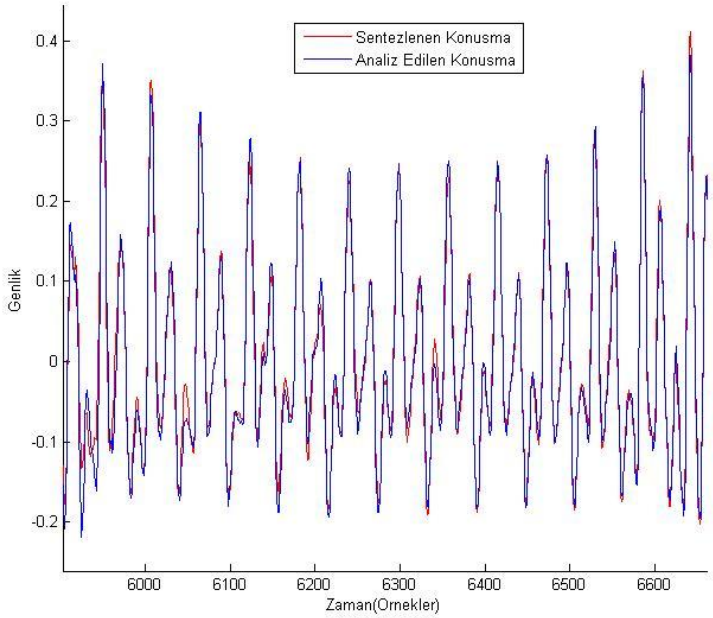
RPE-LTP kodlama yönteminde, CELP kodlama algoritmasındaki rastgele değerlerin bulunduğu kod kitabı düzenli aralıklara sahip darbe dizilerini barındırır. RPE-LTP kodlama yöntemi GSM kodlayıcıda kullanıldığı haliyle şekil 5.4 de verilmiştir.



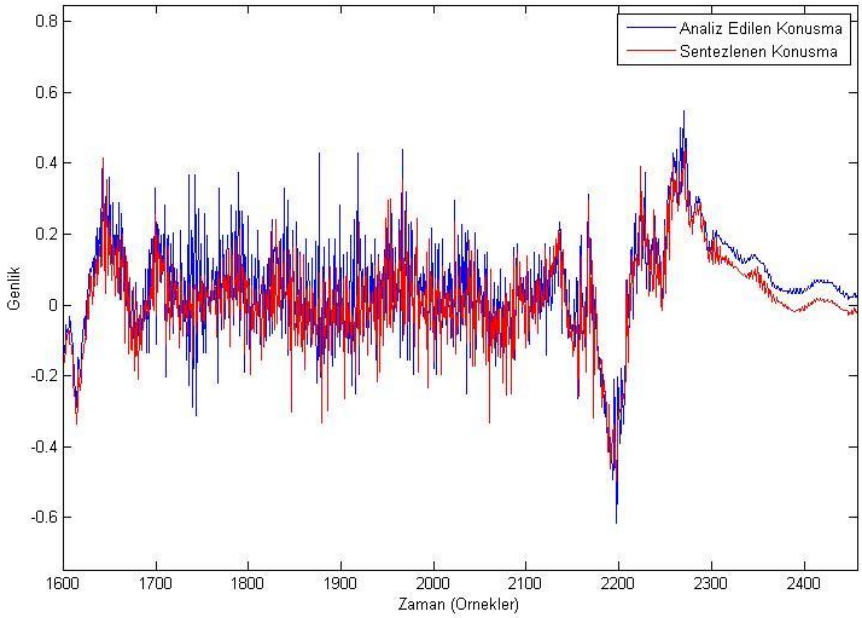

**Şekil 5.30:** RPE-LTP, GSM konuşma kodlayıcısı [17].

## 5.2. Kod Kitaplarında Saklanacak Konuşmaların Belirlenmesi

GSM kodlayıcı incelenirse konuşmanın sessiz kısımları gürültü ile modellenirken, sesli kısımları analiz edilip elde edilen parametrelere göre sentezlenir.




**Şekil 5.31:** GSM kodlayıcı ile kodlanmış konuşmanın sesli bir kısmı.




**Şekil 5.32:** GSM kodlayıcı ile kodlanmış konuşmanın sessiz bir kısmı.

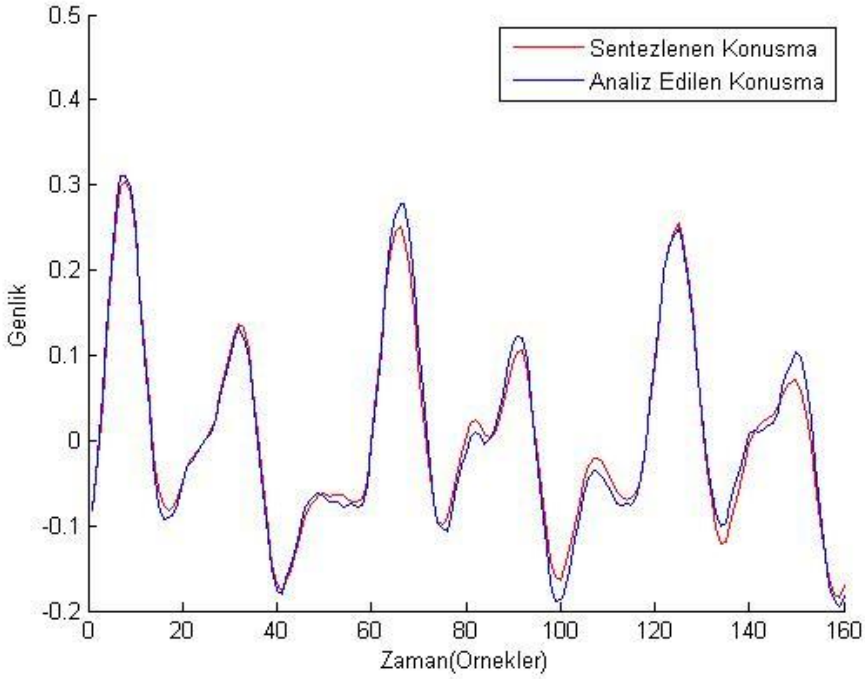
GSM kodlayıcının yapısı dikkate alındığında ve örnek konuşmalar üzerine yapılan deneysel incelemeler sonucunda edilen veriler değerlendirildiğinde, belirlenen model ile GSM üzerinden konuşma iletiminin sağlanması için sayısal verilerin sadece sesli seslerden oluşan konuşmalara çevrilmesi gerektiğine karar verildi.

20 ms uzunluğundaki sesli bir konuşmanın GSM kodlayıcı tarafından kodlanmasıyla elde edilen işaretin ve ilk halinin hamming penceresi ile pencerelendikten sonraki perde periyodu, enerjisi ve LP tahmin katsayıları tablo 5.1 de verilmiştir.

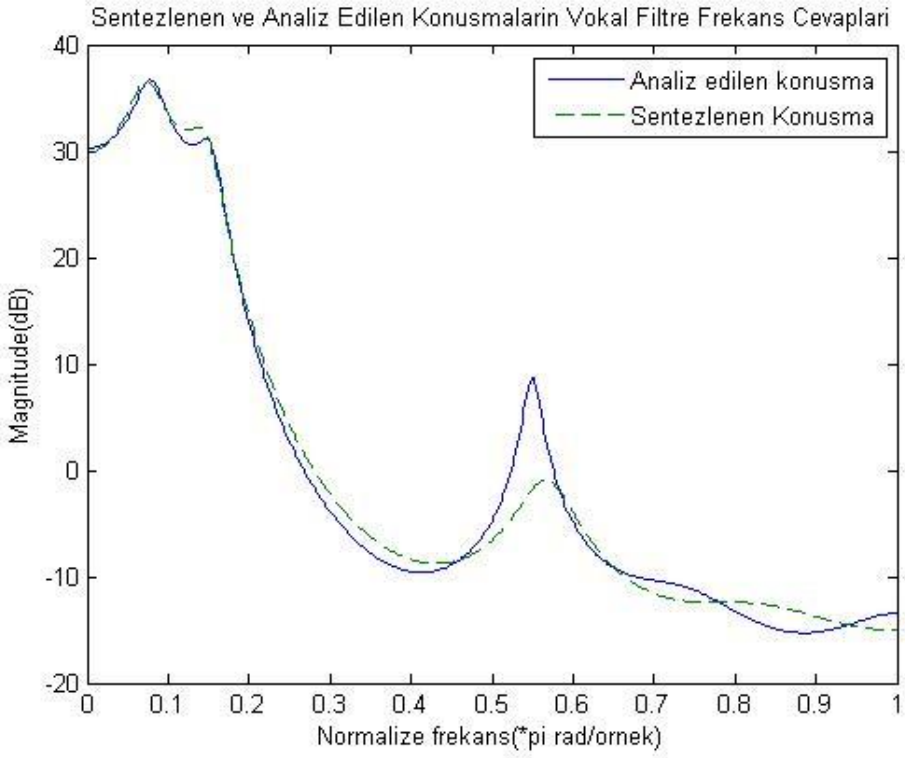
| | Perde Periyodu | Enerji | LP Tahmin Katsayıları |
|---|---|---|---|
| Analiz Edilen Konuşma | 59 örnek | 0.8475 | -2.1645 1.3734 -0.5692 0.7886 0.1465<br>-0.6921 -0.2131 0.1596 0.4616 -0.2602 |
| GSM ile Kodlanan Konuşma | 59 örnek | 0.6957 | -2.2109 1.3781 -0.3141 0.5654 -0.0512<br>-0.4917 -0.2163 0.4587 -0.0218 -0.0639 |

**Tablo 5.1:** GSM kodlayıcı ile analiz edilen ve sentezlenen konuşmaların özellikleri

GSM kodlayıcı ile kodlanan ve analiz edilen konuşmalar şekil 5.7'de, vokal filtre frekans cevapları şekil 5.8'de verilmektedir.




**Şekil 5.33:** GSM kodlayıcı ile analiz edilen ve sentezlenen konuşmalar




**Şekil 5.34:** GSM kodlayıcı ile analiz edilen ve sentezlenen konuşmaların filtre frekans yanıtları

LPC katsayıları kodlamaya daha uygun olan LSF (Line Spectrum Frequencies) katsayılarına çevirilecek ve vektörel kuantalama ile oluşturulacak kod kitaplarında saklanacaktır.

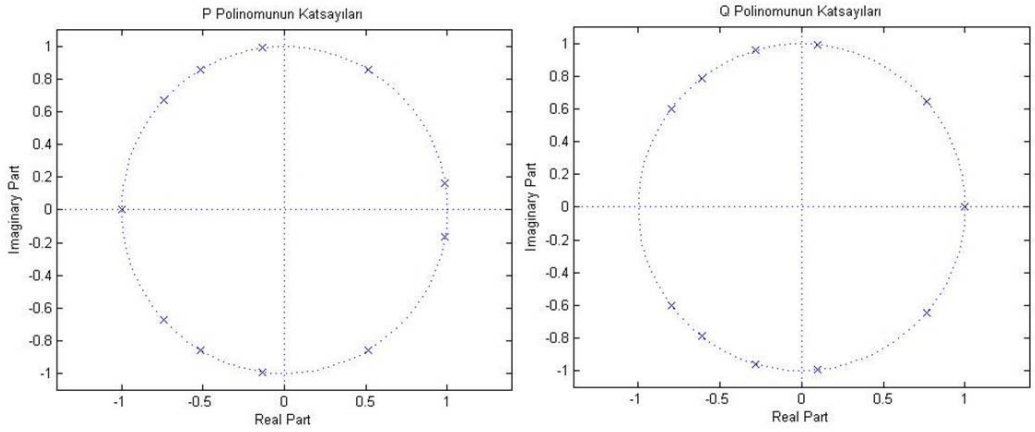
## 5.3. LPC Katsayılarının LSF Parametreleri ile Gösterimi

LP filtre katsayıları, denklem (5.1)'de ve denklem (5.2)'de verilen polinomlarla ifade edilebilir.

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1}) \tag{5.1}$$

$$Q(z) = A(z) - z^{-(p+1)}A(z^{-1}) \tag{5.2}$$

P(z) ve Q(z) polinomlarının kökleri birim çember üzerindedir ve birbiri içine girmiş (interlaced) durumdadır. Şekil 3.22 de verilen LPC katsayılarının oluşturduğu polinomların kökleri şekil 5.9 de verilmiştir.




**Şekil 5.35:** P(z) ve Q(z) polinomlarının kökleri

P(z) ve Q(z) polinomları p/2 tane kompleks konjuge kök setlerinden oluşmaktadır dolayısıyla her polinom beş tane kök ile temsil edilebilir. Polinomların uygun kökleri alındığında LSF parametreleri elde edilir. LSF parametreleri sıralıdır ve formantlar ile doğrudan bir ilişkiye sahiptirler.

LSF parametrelerinden LPC filtre katsayıları denklem (5.2) de verilen ifade sayesinde geri elde edilebilir.

$$A(z) = \frac{P(z) + Q(z)}{2} \tag{5.3}$$

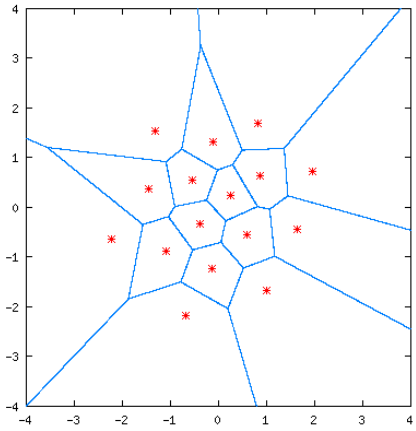
Tablo 5.1 de verilen LPC katsayılarının, LSF parametreleri cinsinden ifadesi tablo 5.2 de verilmiştir.

| | LSF Parametreleri (rad) | | | | | |
|---|---|---|---|---|---|---|
| Analiz Edilen Konuşma | 0.1888 | 0.2785 | 0.4215 | 0.5062 | 0.8091 | 1.6516 |
| | | 1.7357 | 1.9526 | 2.3579 | 2.7669 | |
| GSM ile Kodlanan Konuşma | 0.1883 | 0.2816 | 0.4209 | 0.5049 | 0.8124 | 1.5144 |
| | | 1.7734 | 1.9756 | 2.4102 | 2.7515 | |

**Tablo 5.2:** GSM kodlayıcı ile analiz edilen ve sentezlenen konuşmaların LSF parametreleri cinsinden gösterilimi

### 5.4. LBG Algoritması

Vektörel kuantalama kayıplı bir sıkıştırma yöntemidir. 1981 de LBG algoritmasının geliştirilmesiyle kullanım alanı genişlemiştir. Özellikle görüntü ve ses sıkıştırmada kullanılır.

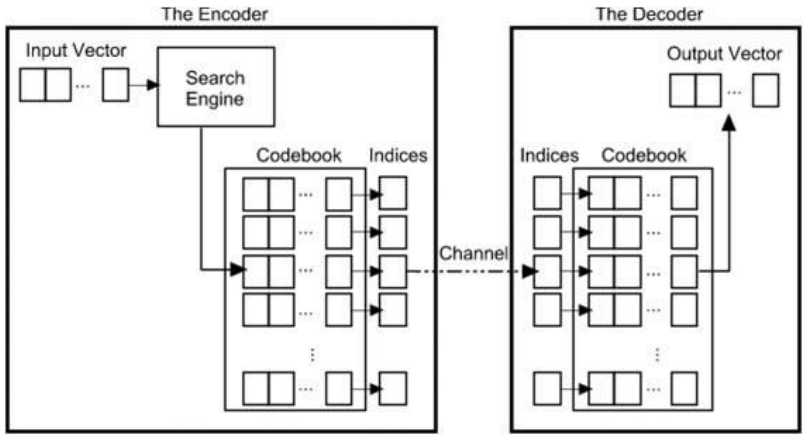



**Şekil 5.36:** 2 boyutlu düzlem için vektörel kuantalama [18].

İki boyutlu bir düzlem için vektörel kuantalama örneği şekil 5.9'da verilmiştir. Toplam alan on altı bölgeye ayrılmıştır. Kırmızı noktalar kod kelimelerini (code words) ifade eder. On altı kod kelimelerinden oluşan, kod kitabı dört bit ile kodlanabilir.

Sıkıştırma işleminde, tüm ses veya resim içinde birbirine benzer parçalar bulunur ve bunlar kod kelimelerini oluşturur. Sıkıştırılma işleminden sonra dosyadaki

parçalar kod kitabındaki indeksler ile ifade edilir. Bu durum şekil 5.11 da gösterilmiştir.




**Şekil 5.37:** Vektörel kuantalama ile dosyanın sıkıştırılması [19].

LBG algoritması sayesinde sisteme ait vektörlerin en verimli şekilde nasıl sınıflandırılacağı belirlenir. Birbirine yakın vektörler aynı kod kelimesinin etrafında toplanırken, birbirinden uzak vektörler farklı kod kelimeleri ile sınıflandırılır.

LBG algoritmasında başlangıç olarak, sınıflandırılacak vektörlerden sınıf sayısı kadarı kod kelimesi olarak seçilir. Ardından her bir kod kitabı kendisine en yakın LBG algoritmasında başlangıç değerleri rastgele seçildiği zaman, sonuç ilk seçilen vektörlere göre değişir. Böyle bir hataya maruz kalmamak için bölünen (splitting) LBG algoritması geliştirilmiştir. Bu algoritmanın gerçeklenmesi için uzaklığı ifade eden ölçüt seçildikten sonra (Burada Öklid mesafesi (Euclidean Distance) seçilmiştir) uygulanması gereken adımlar aşağıda sıralanmıştır [20]:

***Algoritmanın işleyişi sırasında kullanılacak temel kavramlar:***

Sınıflandırılacak vektörler;

$$B=\{B_1, B_2,\ldots, B_m\} \tag{5.4}$$

Elde edilecek kod kelimeleri;

$$C=\{C_1, C_2,\ldots, C_n\} \tag{5.5}$$

Uzaklık kullanılacağı uygulamaya göre farklı şekillerde belirtilebilir. Bu çalışmada kullanılan öklid mesafesi, vektör boyutu l olmak üzere denklem (5.6)'da verilmiştir.

$$D(B,C) = \sum_{i=1}^{l} (B_i - C_i)^2 \tag{5.6}$$

Algoritmanın uygulanmasında birbirinden farklı yaklaşımlar vardır. Klasik metotta sınıflandırılacak vektörlerden sınıf sayısı kadarı kod kelimeleri başlangıç değeri olarak atanır. Ancak farklı başlangıç değerleri için algoritma farklı sınıflandırma sonuçları ile sonlanır. Bu hatadan kurtulmak için bölünen (splitting) LBG algoritması geliştirilmiştir. Bu yaklaşıma göre başlangıçta bir sınıf vardır ve bütün vektörler bu sınıfın elemanlarıdır. Sınıf sayısı ikiye katlanır ve yeni sınıflara göre iterasyon ile yeni kod kitapları bulunur. Bölünen (splitting) LBG algoritması aşağıdaki adımlardan oluşmaktadır [20]:

***Adım 1:***

Başlangıç değerleri atanır.

- Sınıf sayısı, N =1.
- Ortalama Hata, $D^{(0)}$=0.
- e > 0 olmak üzere küçük bir e sayısı belirlenir.
- Bir sınıftan oluşan sistem için kod kelimesi;

$$C_1 = \frac{1}{M} \sum_{k=1}^{M} B_m \tag{5.7}$$

***Adım 2:***

Bölünme (Splitting) işleminin yapıldığı kısımdır, sınıf sayısı iki katına çıkar.

$$N=2*N; \tag{5.8}$$

i = 1,2,3,…,N için

$$C_i^{(k)} = (1+ e * C_i^{(k)}) \tag{5.9}$$

$$C_i^{(k)} = (1 – e * C_i^{(k)}) \tag{5.10}$$

***Adım 3:***

Her sınıf için en uygun kod kelimesinin arandığı iterasyon işleminin başlandığı kısımdır. Her bir vektör için kendisine en yakın kod kelimesi aranır ve vektörler bulunan kod kelimesine ait sınıfa yerleştirilir. P sınıf vektörleri olmak üzere;

$$P_i^{(k)} = B_n : d(B_n, C_i^{(k)}) < d(B_n, C_j^{(k)}) \; ; \quad j \neq i \quad ; \qquad i=1,2,\ldots,M \tag{5.11}$$

***Adım 4:***

Bütün $P_i^{(k)}$ sınıflarındaki $B_i$ vektörleri için denklem (5.7)' de verilen formül ile ağırlık merkezi hesaplanır ve bulunan değer $C_i^{(k)}$ kod kelimesi olur.

***Adım 5:***

Yeni belirlenen sınıflar için hata hesaplanır;

$$D^{(k)} = \frac{1}{M}\sum_{k=1}^{N} D_k, \quad her\ sınıf\ için \quad D_k = \sum_{j=1}^{uzunluk(P^{(k)})} (C_k - B_j)^2 \tag{5.8}$$

***Adım 6:***

- Hata büyüdüyse ya da E kabul edilebilir hata değeri için denklem (5.9)'da verilen eşitlik sağlanıyorsa adım 7'ye geçilir.

$$(D^{(k-1)} - D^{(k)}) / D^{(k)} \leq E \tag{5.9}$$

- Hata küçüldüyse ya da denklem (5.9)'da verilen eşitlik sağlanmıyorsa, denklem (5.10)'de verilen işlem yapılarak adım 3'e geçilir.

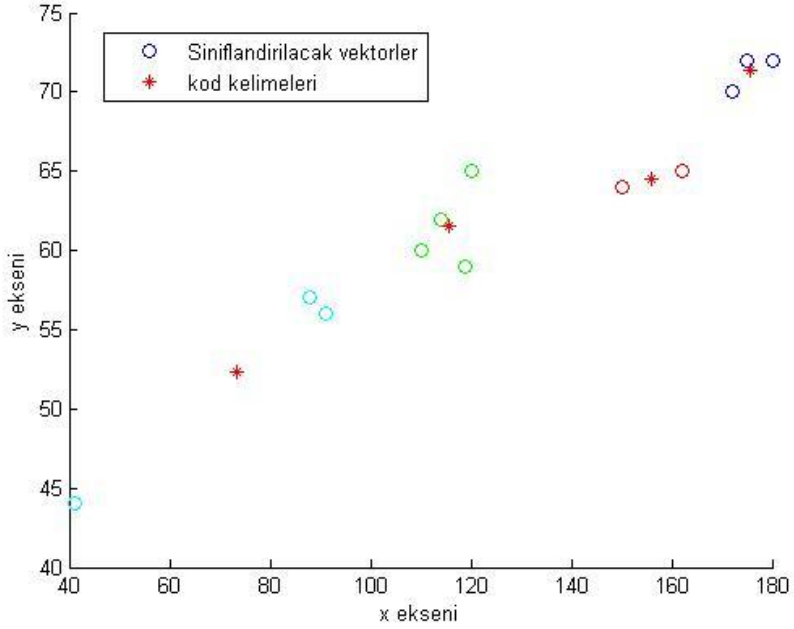
$$k=k+1 \tag{5.10}$$

***Adım 7:***

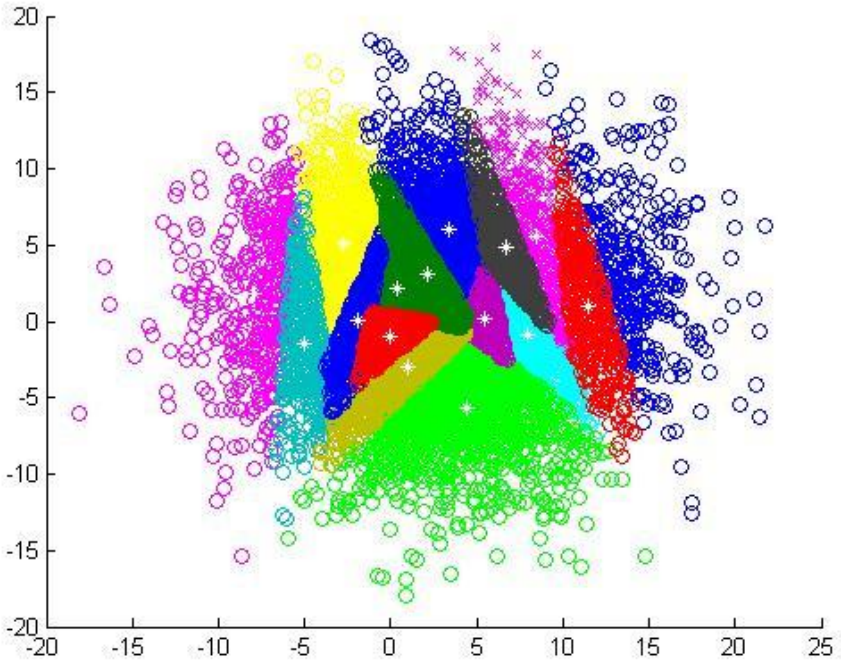
Sınıf sayısı, istenen sınıf sayısına eşit değil ise adım 2'ye geçilir. İstenen sınıf sayısına ulaşılmış ise denklem (5.11)'de verilen eşitlik sayesinde kod kelimeleri bulunarak algoritma sonlandırılır.

$$C_i = C_i^{(k)} \tag{5.11}$$

Dağınık LBG algoritması kullanılarak yapılan sınıflandırma işlemine ait örnekler şekil 5.12 ve şekil 5.13'de verilmiştir.




**Şekil 5.38:** Dağınık LBG algoritmasıyla noktaların dört sınıfa ayrılması




**Şekil 5.39:** Rastgele oluşturulan 1000 vektörün dağınık LBG algoritmasıyla on altı sınıfa ayrılması

### 5.5. Kod Kitaplarının Oluşturulması

Sayısal verilerden GSM hattında bastırılmadan geçebilecek konuşmaların sentezlenebilmesi için konuşmaların temel özelliklerini içeren perde periyodu, enerji ve LSF parametrelerinden oluşan kod kitaplarının oluşturulması gerektiği ve oluşturulacak kod kitaplarının konuşmaların sadece sesli kısımlarını içermesi gerektiğinde bahsedildi.

3.bölümde anlatılan seslilik kriterleri doğrultusunda NTIMIT veritabanı taranarak, sesli konuşmaların LSF parametreleri elde edilmiştir. Bulunan LSF parametreleri, LBG algoritması sayesinde dağınık (splitting) vektörel kuantalama yöntemi için sınıflandırılarak istenilen kod kitapları üretilmiştir. Dağınık vektörel kuantalama yöntemi, bir kod kitabı yerine aynı veri için birden fazla kod kitabının kullanılmasıdır. Bu sayede toplamda daha az kod kelimesi oluşturularak daha iyi ayrıştırılma sağlanır. LSF parametrelerinin ilk 4 parametresinin bir kod kitabı ve son 6 parametresinin ayrı bir kod kitabı olması tavsiye edilir [13]. LBG algoritması, optimizasyon sırasında hata büyüklüğü-iterasyon grafiğinin lokal minimum noktalarına denk geldiğinde en iyi sonucu veremeyebilir. Bu hatanın biraz daha iyileştirilmesi adına algoritmanın 6.adımındaki koşullar sağlansa bile iterasyona belirli bir sayı kadar devam edilerek daha küçük bir hatanın varlığı araştırılmıştır.

Sonuç olarak tasarlanan sistemde kullanılmak üzere (4,6) LSF parametreleri için (5,5) bit sayısal veriler ile ifade edilmek üzere LSF kod kitapları tasarlanmıştır.

| | Enerji | Enerjinin Logaritması | Nominal Enerji | Nominal Enerjinin Logaritması |
|---|---|---|---|---|
| Analiz Edilen Konuşma | 0.8475 | -1.6544 | 1.6164e-005 | -110.3271 |
| GSM ile Kodlanan Konuşma | 0.6957 | -3.6290 | 1.2562e-005 | -112.8483 |

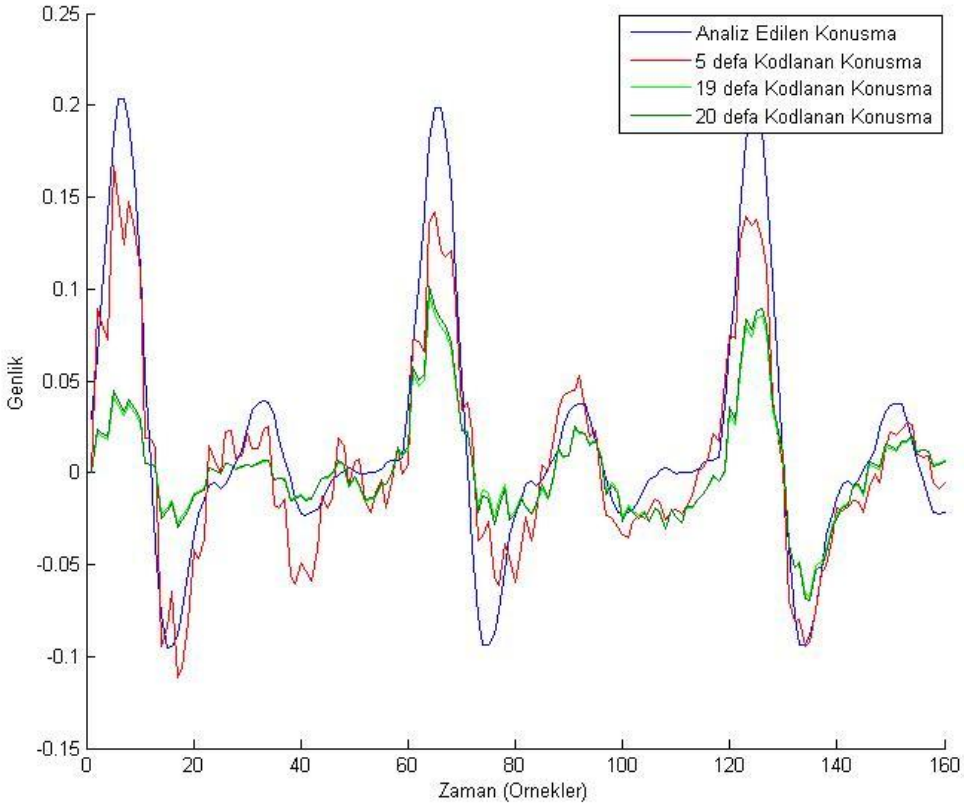
**Tablo 5.3:** Enerji, nominal enerji ve logaritmalarının incelenmesi

Tablo 5.3 incelenirse GSM kodlayıcı kullanılarak sıkıştırılan konuşma ile analiz edilen konuşmanın enerjinin logaritması kodlama için daha uygundur. Ayrıca

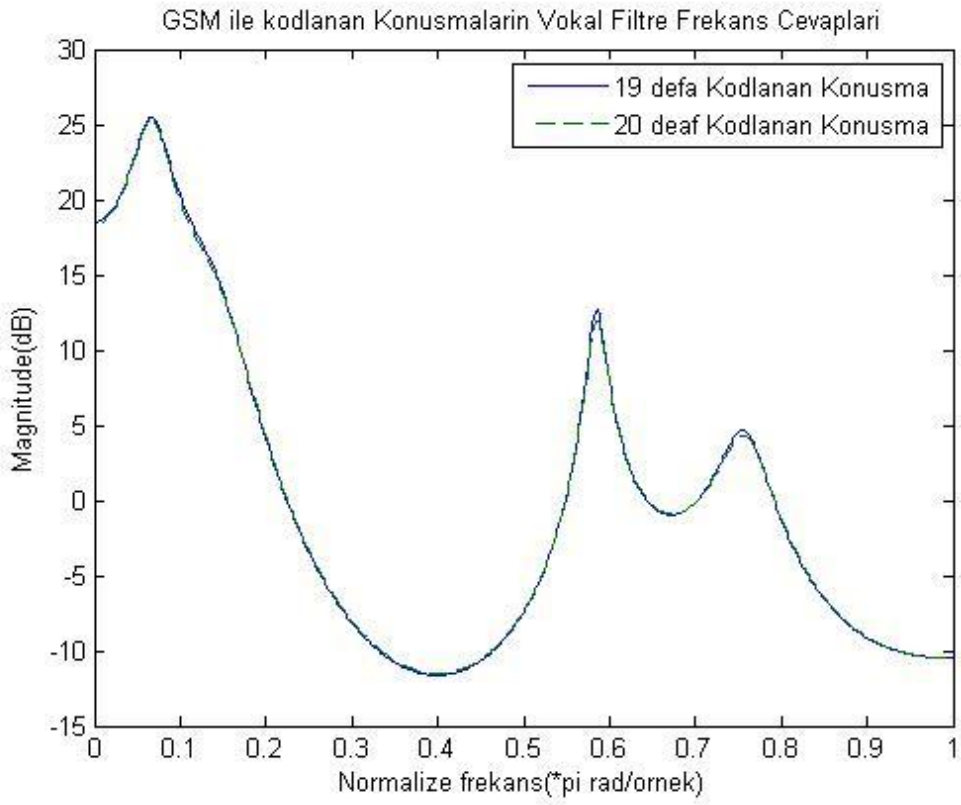
işaretlerin enerjisi incelenirken, denklem 3.51'de verilen nominal enerjinin logaritmasının kullanılması kodlama hatalarının giderilmesi açısından faydalıdır. Nominal enerjinin logaritması 2 bit, perde periyodunun logaritması 4 bit ile kodlanmış, gerekli kod kitapları lineer kuantalama yöntemi ile oluşturulmuştur.

Enerji ve perde periyoduna ilişkin özelliklerin GSM kodlayıcıda kabul edilebilir miktarda bozulmasına karşın, LSF parametrelerinin daha iyi kodlanması ya da bu parametrelerin tanınması için bir takım hata giderici yapıların kullanılması sistemin gürültü duyarlılığını azaltacaktır.

GSM kodlayıcı konuşmayı analiz edip sentezlediğinde kendi yapısındaki kodlama çözünürlüğünde sonuçlar verecektir. Bu yapı düşünülerek GSM kodlayıcıya üretilen kod kitaplarındaki konuşmalar sentezlenerek konuşmalar bozulmayana kadar analiz edildi ve yeniden sentezlendi. Elde edilen yeni konuşmalar LPC yöntemi ile analiz edilerek kod kitaplarındaki LSF parametreleri güncellendi. Şekil 5.7 de verilen konuşma örneği için anlatılan geliştirme yapıldı ve şekil 5.14-5.15, tablo 5.4 de verildiği gibi sisteminin çok daha iyi çalıştığı gözlemlendi.




**Şekil 5.40:** Kodlayıcıya Tekrar ederek Verilen Konuşmadaki Değişim




**Şekil 5.41:** 5 defa kodlanan konuşmanın filtre cevabı

| | LSF Parametreleri | | | | | |
|---|---|---|---|---|---|---|
| GSM ile yirmi defa kodlanan konuşma | 0.1888 | 0.2859 | 0.4506 | 0.6994 | 1.4886 | 1.8211 |
| | | 1.8826 | 2.2529 | 2.4056 | 2.6704 | |
| GSM ile on dokuz defa kodlanan konuşma | 0.1886 | 0.2859 | 0.4561 | 0.7060 | 1.4823 | 1.8205 |
| | | 1.8866 | 2.2493 | 2.4087 | 2.6715 | |

**Tablo 5.4:** GSM ile 19 defa ve 20 defa kodlanan konuşmaların LSF parametreleri

GSM kodlayıcıda kullanılan hataların ve hattan kaynaklanan hataların giderilmesi adına kod kitapları en iyi şekilde tasarlanmaya çalışıldı. GSM kodlayıcıda değişmeyerek kodlanabilen konuşmalar elde edilmiş olsa da LPC yöntemi ile analiz ile değişmeyen konuşmalara ait parametreler bulunamadı. Farklı bir yaklaşım olarak, aynı kod kitapları ile konuşmanın sentezlenip analiz edilmesi yerine sentezlenirken kullanılan kod kitapları ile analiz edilirken kullanılan kod kitapları birbirinden farklı tasarlanabilir. Bu sayede kodlayıcıdan kaynaklanan hata oranı sıfıra indirgenir ve

hattan kaynaklanan gürültü duyarlılığı azaltılır. Bir takım yapay sinir ağları veya konuşmacı tanıma algoritmaların sisteme eklenmesi ile eğitilebilir bir kodlayıcı tasarlanabilir ve bu sayede iletişim kalitesi artar.

# 6. SONUÇLAR VE TARTIŞMA

Bu bitirme çalışmasında GSM konuşma hattı üzerinden konuşmanın güvenli gönderilebileceği bir sistem gösterildi. Bu sistemde konuşma, sayısallaştırılır ve şifrelenir. Şifrelenen bit dizileri, 13 kpbs GSM FR konuşma kodlayıcısında ve GSM iletim hattında hiç bozulmadan ya da düzeltilebilir bir hata miktarı ile iletilebilecek konuşmalara dönüştürülerek hatta verilir. Alıcı taraf; konuşmayı alır, analiz eder ve ilk konuşma yeniden üretilir.

Yapılan çalışma sırasında, sayısal veri şifrelenmesi için 128 bit AES algoritması FPGA üzerinde gerçeklendi. Sayısal verilerden, bozulmadan iletilebilecek konuşmaların elde edilmesi için gerekli özelliklere karar verildi ve istenen özelliklere ait parametreler NTIMIT ses veritabanı taranarak elde edildikten sonra LBG algoritmasıyla tasarlanan kodlayıcı tarafından kullanılabilir hale getirildi. Matlab ortamında tasarlanan kodlayıcı ile şifreleme donanımı seri port ile haberleştirildi. Bilgisayar ortamında cep telefonunu ifade eden 13 kbps GSM FR kodlayıcısı modellendi. FPGA de şifreleme donanımına karakter LCD bağlandı ve GSM iletim hattı hataları ihlal edildiğinde sistemin çalışır olduğu gözlemlendi. Ayrıca kodlayıcı yapısının eğitilebilir olması için giriş ve çıkış kod kitaplarının farklı olmasına dair yeni bir metot önerildi.

Bu çalışma gelecekte, gerçek GSM hatlarında gerçek sistemler ile denenip, uygun kod kitapları oluşturulduktan sonra FPGA üzerinde yapılacak bir donanınım tasarımının ilk halidir ve teorik bir modellenmesidir.

## KAYNAKLAR


**[1] C. Lo, Y. Chen,** "Secure communication mechanisms for GSM networks," IEEE Transactions on Consumer Electronics, Vol. 45, No. 4, pp. 1074-1079, November 1999.

**[2] N.N. Katugampala, K.T.Al-Naimi, S. Villette, A. Kondoz;** "Real time data transmission over GSM voice channel for secure voice and data applications", Secure Mobile Communications Forum: Exploring the Technical Challenges in Secure GSM and WLAN, The 2nd IEE (Ref. No. 2004/10660) ,London,23 Sept. 2004.

**[3] C.K. LaDue, V.V. Sapozhnykov, K.S. Fienberg,** "A Data Modem for GSM Voice Channel," *IEEE Transactions on Vehicular Technology,* , vol.57, no.4, pp.2205-2218, July 2008.

**[4] N.N. Katugampala, S. Villette, A. Kondoz,** "Secure voice over GSM and other low bit rate systems", Secure GSM and Beyond: End to End Security for Mobile Communications, IEE Seminar on (Digest No. 2003/10059), London,11 Feb. 2003.

**[5] Y. Yang, S. Feng, W. Ye, X. Ji,** "A Transmission Scheme for Encrypted Speech over GSM Network", *Computer Science and Computational Technology, 2008. ISCSCT '08. International Symposium on* , Vol.2, pp.805-808, 20-22 Dec. 2008.

**[6] FS 1016,** "The Proposed Federal Standard 1016 4800 bps Voice Coder: CELP", Joseph P. Campbell Jr., Thomas E. Tremain and Vanoy C. Welch. **Speech Technology Magazine**, April/May 1990, pp. 58-64.

**[7] Bradbury,** Linear predictive coding. University of Biritish Colombia, J. 2000.

**[8] A. V. Oppenheim and R. W. Schafer,** "Digital Signal Processing", Prentice Hall, 1975.

**[9] S. H. Bae,** "LPC10 2.4 kbps federal Standard in speech coding",Georgia Institue of Technology, 17.03.2004 accessed from *http://users.ece.gatech.edu/~juang/8873/Bae-LPC10.ppt* in 20.05.2011.

**[10] I. McLoughlin,** "Applied Speech and Audio Processing", Cambridge University Press,New York, 2009.

**[11] L. R. Rabiner, R. W. Schafer, "**Digital Processing of Speech Signals" Prentice-Hall (Signal Processing Series), 1978.

**[12] B. Atal, , L. Rabiner,** "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition", *Acoustics, Speech and Signal Processing, IEEE Transactions on* , Vol.24, No.3, pp. 201- 212

**[13] A. M. Kondoz,** "Digital Speech:Coding for Low Bit Rate Communication Systems", John Wiley&Sons, Second edition 2004.

**[14] L. Ordu,** 2006. AES Algoritmasının FPGA Üzerinde Gerçeklenmesi ve Yan Kanal Analizi Saldırılarına Karşı Güzlendirilmesi, *Yüksek Lisans Tezi*, İTÜ Fen Bilimleri Enstitüsü, İstanbul.

**[15**] **FIPS 197,** 2001. Advanced Encryption Standard, *National Institute of Standarts and Technology*.

**[16**] **T. Dutoit, F. Marques,** 2009."Applied Signal Processing", Springer.

**[17**] **R. Goldberg, L. Riek,** *A Practical Handbook of Speech Coders*, 2000.

**[18**] **http://www.data-compression.com/vq.shtml,** "Vector Quantization" accessed at 25.05.2010.

**[19**] **D. Salomon,** "Data compression:The Complete Referance", Springer 4th Edition, 2006.

**[20**] **Y. Linde, A. Buzo, R. Gray,** "An Algorithm for Vector Quantizer Design", *IEEE Transactions on Communications,* , Vol.28, No.1, pp. 84- 95, Jan 1980.

## ÖZGEÇMİŞ

**Adı Soyadı:** Mehmet Akif Özkan

**Doğum Yeri ve Tarihi:** Ankara, 1989

**Lise:** Ankara Atatürk Anadolu Lisesi;  2003-2007

**Lisans:** İstanbul Teknik Üniversitesi, Elektronik Mühendisliği;  2007-2011