İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KALP SESLERİNİN GERÇEK ZAMANDA ALGILANMASI VE BİLGİSAYARDA ANALİZ EDİLMESİ

YÜKSEK LİSANS TEZİ

Müh. Özgün Onat DÜZGÜN

ANABİLİM DALI: ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
PROGRAMI: BİYOMEDİKAL MÜHENDİSLİĞİ

OCAK 2007

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

KALP SESLERİNİN GERÇEK ZAMANDA ALGILANMASI VE BİLGİSAYARDA ANALİZ EDİLMESİ

YÜKSEK LİSANS TEZİ

Müh. Özgün Onat DÜZGÜN

504031406

Tezin Enstitüye Verildiği Tarih : 25 Aralık 2006
Tezin Savunulduğu Tarih : 31 Ocak 2007

Tez Danışmanı : Doç. Dr. Zümray DOKUR ÖLMEZ

Diğer Jüri Üyeleri : Doç. Dr. Mehmet KORÜREK

Prof. Dr. Serhat ŞEKER

OCAK 2007

ÖNSÖZ

Bu çalışmanın hazırlanması sırasında yardımını ve desteğini eksik etmeyen danışmanım Doç. Dr. Zümray Dokur Ölmez'e ve Prof. Dr. Tamer Ölmez'e sonsuz teşekkürlerimi sunarım.

Değerli katkılarından dolayı Doç.Dr. Mehmet Korürek'e, arkadaşım Mehmet Kaya'ya ve desteği için ablam Özlem'e teşekkür ediyorum.

Ocak 2007 **Özgün Onat Düzgün**

İÇİNDEKİLER

ÖNSÖZ ii

İÇİNDEKİLER iii

TABLO LİSTESİ v

ŞEKİL LİSTESİ vi

ÖZET viii

SUMMARY ix

1. GİRİŞ 1

1.1. Giriş ve Çalışmanın Amacı 1

1.2. Biyolojik İşaretlerin Analiz Teknikleri 2

2. KALP SESLERİ 3

2.1. Kalbin Vücuttaki Konumu ve Yapısı 3

2.2. Kalbin Çalışması 4

2.3. Kalbin Elektriksel Aktivitesi 5

2.4. Kalbin Mekanik Aktivitesi 6

2.5. Kalbin Elektrik- Mekanik Aktivitesinin İncelendiği İşaretler 7

2.6.Kalp Sesleri 9

2.6.1 Normal Kalp Sesleri 10

2.6.2 Anormal Kalp Sesleri 11

2.6.3 Kalp Seslerinde Görülen Farklılıklar ve Nedenleri 13

2.7.Dinleme Yöntemi 14

2.7.1 Kalp Seslerini Dinleme Odakları 14

3. FOURIER ANALİZİ 16

3.1 Fourier Dönüşümü 16

3.2 Hızlı Fourier Dönüşümü 17

3.3 Kısa-Zaman Fourier Dönüşümü 20

4.KALP SESLERİ İÇİN VERİ TOPLAMA SİSTEMİ 23

4.1 Dönüştürücü Özelliklerinin Sistem Üzerine Etkileri 23

4.1.1 Kalp Seslerini Algılamak İçin Kullanılan Dönüştürücü 25

4.2 Ön Kuvvetlendirici 26

4.3 Filtre Tasarımı 29

4.4 Kuvvetlendirici Tasarımı 31

4.5 İşaretin Sayısal Ortama Aktarılması 33

5.GERÇEK ZAMANDA KALP SESLERİNİN ANALİZİ 36

5.1 İşaretin Bilgisayara Aktarılması 36

5.2 Program ve Menü Özellikleri 36

5.3 Analiz Sonuçlarının Ekranda Gösterimi 37

6. SONUÇLAR 40

KAYNAKLAR 42

EKLER 44

EK.1 44

ÖZGEÇMİŞ 65

TABLO LİSTESİ

Sayfa No

Tablo 2.1. Kardiyak sesler ……. 12
Tablo 4.1. INA118'in R_G direnciyle değişen kazanç tablosu…….. 29

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1 : Biyolojik büyüklüklerin analiz teknikleri..... 2
Şekil 2.1 : Kalbin dış görünümü 3
Şekil 2.2 : Kalbin iç yapısı 4
Şekil 2.3 : Kalbin kasılması (Sistol)..... 4
Şekil 2.4 : Kalbin gevşemesi (Diastol)..... 4
Şekil 2.5 : Kalbin elektriksel odakları ve ileti yolları..... 5
Şekil 2.6 : Kardiyak döngüdeki fazların süresi……..... 6
Şekil 2.7 : Kalpteki mekanik, akustik ve elektriksel kardiyak sinyallerin bir kardiyak döngüde zamana göre ilişkileri…..... 8
Şekil 2.8 : Sistol ve diastol…… 9
Şekil 2.9 : Kalp sesleri……..... 10
Şekil 2.10 : Kalp Seslerini Dinleme Odakları 15
Şekil 3.1 : N=8 için periyodik ve simetrik W pencere sabiti..... 18
Şekil 3.2.a : Zamanla değişen işaret s1(t)..... 19
Şekil 3.2.b : Zamanla değişen işaret s2(t)……..…..... 19
Şekil 3.3.a : s1(t)'nin Fourier Dönüşümü…….. 20
Şekil 3.3.b : s2(t)'nin Fourier Dönüşümü…….. 20
Şekil 3.4 : Kısa-zaman Fourier Dönüşümü uygulanan işarette frekans,zaman ve genliğin üç boyutlu gösterimi……... 22
Şekil 4.1 :Fizyolojik işaretlerin sayısal ortama aktarılmasını gösteren blok diyagram…… 23
Şekil 4.2 :Çalışmada kalp seslerini algılamak için kullanılan mikrofon..... 25
Şekil 4.3 : Fark kuvvetlendiricisi blok diyagramı …… 26
Şekil 4.4 : Enstrumantasyon kuvvetlendiricisi …….. 27
Şekil 4.5 : INA118 Enstrumantasyon kuvvetlendiricisi iç 28
Şekil 4.6 dizaynı…..…… 29
Şekil 4.7 : Devrede kullanıldığı haliyle INA118 blok şeması…… 30
Şekil 4.8 : Alçak geçiren filtre devresi …….. 31
Şekil 4.9 : Kuvvetlendirici devresi …….……... 32
Şekil 4.10 : Negatif gerilim çevirici…….…... 32
Şekil4.11.a : İşaretin analog olarak elde edilmesi …….....
: Analog işaretin osiloskopta görüntülenmesi(t=0.1ms ve V=0.5V 33
Şekil4.11.b için) ……...
: Analog işaretin osiloskopta görüntülenmesi(t=0.1ms ve V=0.5V 33
Şekil 4.12 için) ……... 34
Şekil 4.13 : ADUC841 pin konfigurasyonu…… 34
Şekil 4.14 : ADM3202 seri port entegresi fonksiyonel blok diyagramı……... 35
Şekil 4.15 : ADP3335 tipik uygulama devresi…….
: Kalp sesini mikrofon ile vücuttan alıp seri port ile bilgisayara 35
Şekil 5.1 aktaran sistem…… 36
Şekil 5.2 : Kalp sesi analiz menüsü…… 37

Şekil 5.3 : Gerçek zamanda alınan kalp sesi işareti…….. ……. 38
Şekil 5.4 : Gerçek zamanda S1 S2 sürelerinin gösterilmesi …….…….
: Ses işaretinin gerçek zamanda Fourier ve Kısa-Zaman Fourier 39
dönüşümünün alınması…….

KALP SESLERİNİN GERÇEK ZAMANDA ALGILANMASI VE BİLGİSAYARDA ANALİZ EDİLMESİ

ÖZET

Kalp sesleri, kalbin çalışmasının ve bu nedenle de içerisindeki kan akışının neden olduğu gürültülerdir. Kalp kapakçıklarının açılıp kapanmasının, kanın kalp loblarına dolup loblardan boşalmasının, üfürümlerin ve daha birçok kardiyak sesin dinlenmesinde klasik olarak steteskop kullanılır. Ancak çevresel gürültüler, dinleyicinin kulağının iyi duyamayabilmesi ya da sesin zor duyulur olması gibi sebeplerle steteskopla dinlenen bilgi çoğu zaman yetersiz kalmaktadır. Bu durum, tanı koyarken farklı yöntemler kullanılmasını ya da sesin daha net alınabilmesi, ayırt edilebilmesi için daha gelişmiş yöntemler kullanılmasını gerekli kılmaktadır.

Bu tez çalışmasında amaç, hastadan kalp sesinin elde edilmesi ve alınan işaretin bilgisayara gönderilerek analiz edilmesidir.

Sesin hastanın göğsünden algılanabilmesi için özel bir mikrofon kullanılmıştır. Sinyal, gürültüden ayırt edilmeyecek kadar düşük genlikte olduğundan (tüm biyolojik işaretler gibi), gürültüyü kuvvetlendirmeden, mümkün olabildiğince incelenen biyolojik işaretin kuvvetlendirilmesini sağlayan yüksek kazançlı bir enstrumantasyon kuvvetlendiricisi kullanılmıştır. Kalp sesleri düşük frekanslı olduğu için (yaklaşık 10–500 Hz aralığında), yüksek frekans bileşenlerini filtreleyen bir alçak geçiren filtre kullanılmıştır. Elde edilen işareti uygulamalarda kullanılabilir düzeye çıkarabilmek için de ayrıca bir kuvvetlendirici kullanılmıştır. Analog devrenin tasarımı, aşama aşama anlatılmıştır. ADC ile sayısal forma dönüştürülüp, seri port üzerinden bilgisayara aktarılan işaretin sayısal olarak analiz edilmesi için gerekli yazılım C programlama dilinde yazılmıştır. Yazılım ile işaretin bilgisayar ortamında gerçek zamanlı gösterimi sağlanmıştır. Analiz yöntemi olarak Fourier analizi ve Kısa-Zaman Fourier analizi seçilmiştir.

REAL TIME ACQUISITION OF HEART SOUNDS AND COMPUTER AIDED PROCESSING

SUMMARY

The heart sounds are the noises (sound) generated by the beating heart and the resultant flow of blood through it. In cardiac auscultation, an examiner uses a stethoscope to listen for these sounds, which include heart tones, or sounds, produced by sudden blood deceleration after the heart valves close, heart murmurs, and adventitious sounds, or clicks. The information that is acquired using stethoscope, generally inefficient because of the environmental noises, the probable problem at the examiner's ears or low volume of the sounds. For these reasons, it is needed to use advanced techniques for acquisition and identification of heart sounds

In this thesis study, the aim is to get the heart sounds from patient, then send it to computer and process the obtained signal.

A special microphone is used to obtain the sound from patient's chest and to convert it to electrical signal. The signal amplitude is too low to distinguish the signal (like all biological signals) from the noise. So, an instrumentation amplifier is used to amplify the signal amplitude. Heart sounds have low frequencies (~10 – ~500 Hz), so a low pass filter is used to reject high frequency components. An amplifier is used to carry the signal amplitude to a suitable level for applications. The design of the analogue circuit is explained step by step. The signal is converted to digital form using an ADC and it is transferred to the computer via serial port. A software program is written in C programming language to process the signal. Using this program, the signal can be printed and processed in real time. The analysis methods for this thesis are chosen as Fourier transform and Short-Time Fourier transform.

1.GİRİŞ

1.1 Giriş ve Çalışmanın Amacı

Kardiyolojik klinik uygulamalarda kalp seslerinin dinlenmesi, ön tanı konulmasında büyük önem taşımaktadır. Doktor, bu sesleri dinleyerek elde ettiği verileri diğer verilerle (elektrokardiyografi, ekokardiyografi) birleştirerek kesin tanı koyma yoluna gider. Ancak kalp sesini doğru algılamak hiç de kolay bir işlem değildir. Ortamın gürültüden arındırılmış olması, hastanın kas hareketlerinin minimuma indirilmesi gerekir. Bununla birlikte dinleyen kişinin de sesi iyi duyabilmesi (duyma problemi olmaması) gerekir. Duyulan seslerin ayırt edilebilmesi de ayrıca bir tecrübe gerektirir. Bu durumlar göz önünde bulundurulduğunda, sesi algılayan ve hem görsel hem de işitsel olarak işlenmeye hazır hale getiren bir sistem, ön tanı koyma işleminin daha hızlı ve doğru yapılmasında gerekli olacaktır.

Sayısal işlem yapan cihazların gelişmesiyle birlikte, görüntü ve işaret verileri de sayısal olarak işlenebilirliğini artırmıştır. Özellikle çok hızlı sayısal işaret işleyen işlemcilerin geliştirilmesi bu işlemin çekiciliğini artırmıştır. Ancak doğadaki birçok işaret (biyomedikal işaretler gibi) sürekli formda alınır ve bu işaretlerin sayısal ortamlarda işlenebilmesi için, sürekli olan işaretin sayısal forma dönüştürülmesi gerekir. Bu dönüşüm yapılırken de, bilgi kaybının en az olacağı yöntem seçilmelidir.

İşaret sayısal ortama aktarılırken, sürekli formdaki bilginin en az kayıpla sayısallaştırıldığı bir dönüşüm yöntemi seçildiği kabul edilirse, bu kez de bu işlemin ne kadar hızlı yapılabileceği sorusu karşımıza çıkar. Bu durumda işaretin tekrarlanan, yani periyodik olma özelliği kullanılarak sayısallaştırma işlemi hızlandırılabilir.

Kalp sesleri, kalbin mekanik işlevi ve kanın kardiyovasküler sistemde hareketi sırasında meydana gelen biyolojik işaretlerdir. Diğer biyolojik işaretler gibi, uygulamalarda kullanılabilir hale gelebilmesi için bir takım dönüşümlerden geçmesi gerekir. Bu nedenle bu çalışmada, kalp seslerinin insan vücudundan algılanıp analiz edilebilir hale getirilmesi amaçlanmıştır. Vücuttan bir mikrofon ile alınan biyolojik işaret, hazırlanan sistem ile elektriksel işarete dönüştürülmüş, filtreleme ile dış gürültülerden arındırılmış, incelenebilir genlik seviyesine kuvvetlendirilmiş ve bilgisayar ortamında analiz edilebilmesi için sayısal forma dönüştürülmüştür. Seri

port ile bilgisayara aktarılan işaret, Fourier dönüşümü ve kısa-zaman Fourier dönüşümü ile analiz edilmiştir.

İkinci bölümünde kalbin yapısı, elektriksel-mekanik aktiviteleri ve bu aktiviteler sırasında oluşan sesler anlatılmıştır.

Üçüncü bölümde işaret işlemede kullanılabilecek yöntemlere yer verilmiş, Fourier, Hızlı Fourier ve Kısa-Zaman Fourier Dönüşümleri anlatılmış ve bu dönüşümlerden elde edilen sonuçların farkları belirtilmiştir.

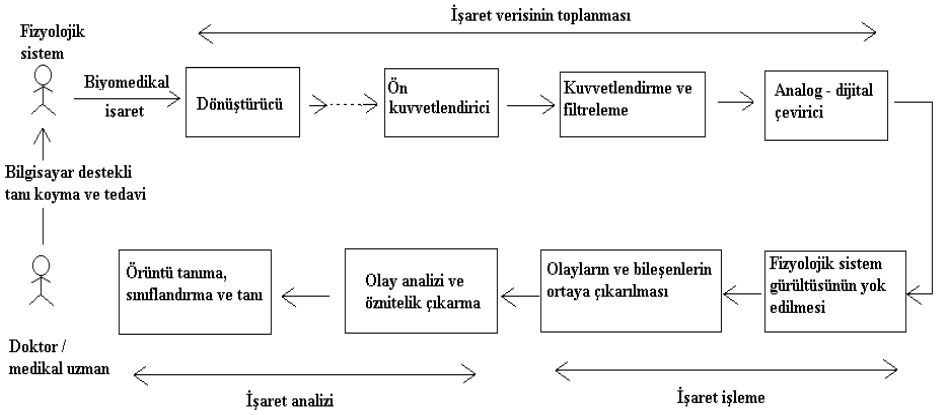
Dördüncü bölümde, kalp seslerinin sayısal ortama aktarılması için tasarlanan devre aşama aşama anlatılmış, kullanılan elemanların özellikleri belirtilmiştir.

Beşinci bölümde, sayısal ortama aktarılan işaretin, C programlama dilinde yazılan bir program ile bilgisayar ekranında görüntülenmesi ve işaretin analizi anlatılmıştır.

1.2 Biyolojik İşaretlerin Analiz Teknikleri

Canlılardan elde edilen biyolojik işaretlerin ölçülmesinde, işaretin kaynağı (canlı) da ölçüm sisteminin bir parçası olarak değerlendirilmelidir.

Canlı organizmadan alınan biyolojik büyüklükler (sıcaklık, basınç, ses, akış) dönüştürücü ile elektriksel işarete dönüştürülür. Dönüştürülen işaret, işaret işleme ünitesinde kuvvetlendirme, filtreleme ve/veya sayısal forma dönüştürme işlemlerine tabi tutulur. Gürültülerden ve diğer bozucu etkenlerden arındırılan işaret karakterize edilir ve örneklenir. Son olarak sınıflandırılır ve tanı konabilecek seviyeye ulaştırılır [1] Canlıdan alınan biyolojik işaretin sayısal forma dönüştürülmesi, işaretin işlenmesi ve analizi aşamalarını özetleyen blok diyagram Şekil 1.1'de verilmiştir.

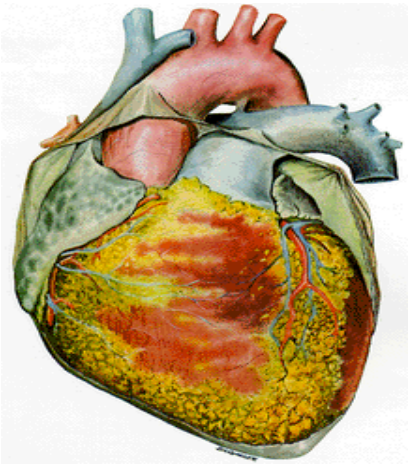


Şekil 1.1: Biyolojik büyüklüklerin analiz teknikleri, **Yazgan E. ve Korürek M. , (1996)**

2. KALP SESLERİ

2.1 Kalbin Vücuttaki Konumu ve Yapısı:

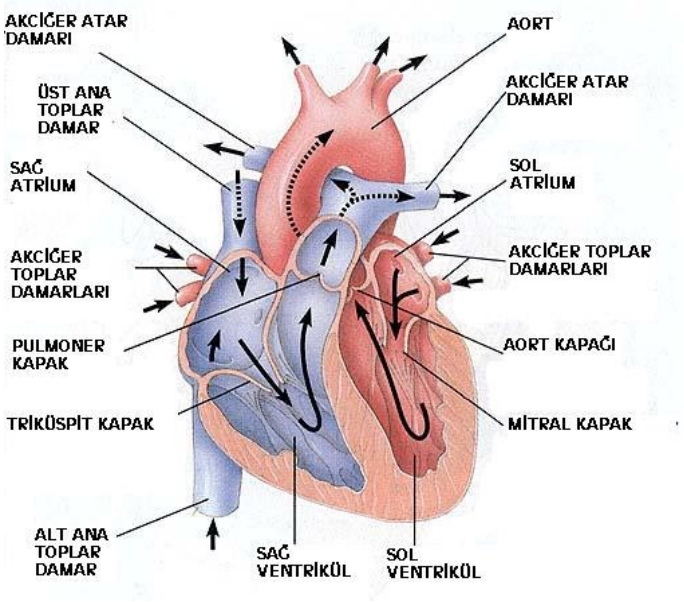
Kalp, insanda göğüs kafesi içinde, iki akciğerin arasında ve göğüs kemiğinin hemen arkasında yer alır. Kalp; tabanı üstte, tepesi altta ve ucu sola dönük olan bir koni şeklindedir. Şekil 2.1 de kalbin dış görünümü verilmiştir.



Şekil 2.1: Kalbin dış görünümü [2]

Yumruktan biraz daha büyük, kas yapısında güçlü bir pompa olan kalp, üstte iki kulakçık (atrium) ve alta iki karıncık (ventrikül) olmak üzere dört gözlüdür. Şekil 2.2 de kalbin iç yapısı gösterilmiştir. Kalp, ortadan tam bir perde ile sağ ve sol bölüm olmak üzere ikiye ayrılmıştır. Sağ bölümde, sağ kulakçık ve sağ karıncık bulunur. Bu iki odacık üçlü bir kapakçıkla (triküspit kapak) birbirine açılır. Sol bölümde, sol kulakçık ve sol karıncık yer alır. Bu iki odacık arasında ikili bir kapakçık (mitral kapak) bulunur.

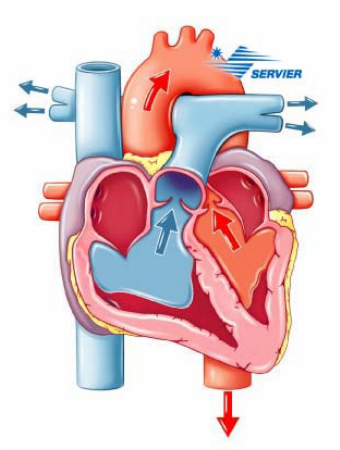
Kulakçıklar ile karıncıklar arasındaki kapakçıklar, karıncıklara doğru tek yönde açılırlar. Bu kapakçıklar, “kalp iplikleri” adı verilen sağlam yapılı özel kaslarla karıncıkların duvarına bağlanmıştır. [4]

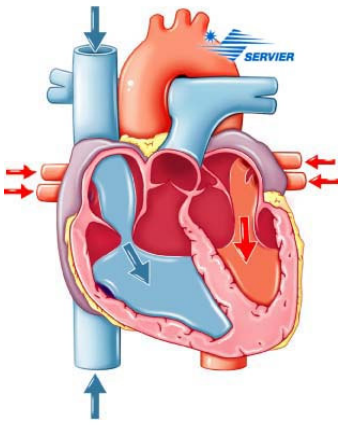


Şekil 2.2: Kalbin iç yapısı [5]

2.2 Kalbin Çalışması

Kalbin çalışması, kalp kasının kasılıp (sistol) gevşemesi (diyastol) ile olur. Kulakçık ve karıncıklar, kasılıp gevşeyerek kan dolaşımının itici gücünü oluştururlar. Bu kasılma kalbin belli yerlerinde bulunan özelleşmiş düğüm dokuları (sino-atrial düğüm ve atrio-ventriküler düğüm) ile düzenlenir. Kulakçık ve karıncıkların kasılıp gevşemesi birbirine zıttır. Biri kasılırken diğeri gevşeme durumuna geçer. Kalbin her odacığı kasılma sırasında içindeki kanı pompalar (Şekil 2.3), gevşeme anında ise kanla dolar (Şekil 2.4). [6]





Şekil 2.3: Kalbin Kasılması (Sistol) [5]

Şekil2.4: Kalbin Gevşemesi (Diyastol)[5]

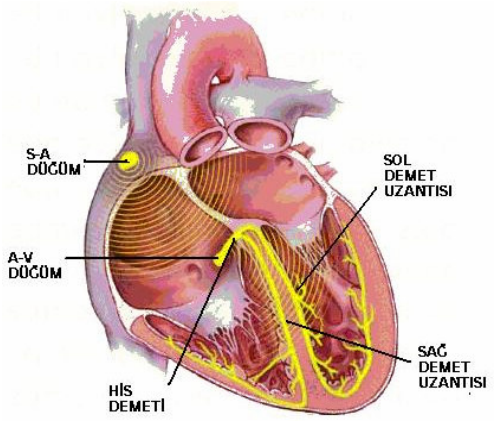
Her bir kalp atışı bir kasılma ile bir gevşemeden meydana gelir. Bir kalp atışı yaklaşık 0,85 saniye sürer. Bunun 0,15 saniyesinde kulakçıklar, 0,30 saniyesinde karıncıklar kasılır. Geri kalan 0,40 saniyelik sürede ise kalp dinlenir. [6] Sağlıklı bir insanda kalp atışının sayısı dakikada 70 – 80 kadardır.

Kalp atış hızı, sadece kalbin kendi özelliğinden değil; ayrıca sinirler, hormonlar, karbondioksit miktarı ve sıcaklıktan da etkilenir.

2.3 Kalbin Elektriksel Aktivitesi

Kalp, herhangi bir uyarı olmadan ritmik olarak kasılma yeteneğine sahiptir. Kalp atım hızı, otonom sinir sistemi tarafından kontrol edilir. Kalp atım hızını kontrol eden kardiyak merkez, beynin tabanında 'medulla oblongata'da bulunmaktadır. "Vagus siniri" adı verilen sinir kalp hızını yavaşlatırken, "accelerator" adı verilen sinir kalp hızını artırır. Her iki sinir de, sağ atriumda üst ana toplardamarın giriş yerine yakın bir noktada yer alan bir sinir kası dokusunda sonlanır. Bu doku, sino-atrial düğüm (S-A düğüm) ya da sino-atrial darbe üreteci (doğal pacemaker) olarak adlandırılır [6].

Benzer bir sinirkası dokusu da sağ ve sol atriumlar arasındaki duvarda bulunan atrio-ventriküler (A-V) düğümdür. A-V düğümden, iki ventrikülün arasında bulunan septum içinden geçen ve kardiyak kaslardan oluşan bir yapı, his demeti, uzanır. His demeti biri sağ diğeri de sol ventriküle doğru olmak üzere iki dala ayrılır. Bu iki dal, daha ince dallardan oluşan bir ağ gibi ventrikül duvarı boyunca yayılır. Bu uzantılara Purkinje fiberleri adı verilir. Purkinje fiberleri, diğer tüm kardiyak kaslara nazaran daha hızlı iletim yeteneğine sahiptir. Şekil 2.5'te kalbin elektriksel odakları ve ileti yolları belirtilmiştir.



Şekil 2.5: Kalbin elektriksel odakları ve ileti yolları [6]

Kasılma S-A düğümünden ya da pacemakerden başlatılır. İleti her iki atrium boyunca yayılır ve atrio-ventriküler düğüme ulaşır. İleti, ventriküllerin arasında bulunan His demeti ile atrio-ventriküler düğümden septuma ulaşır. His demetinin hızlı iletimi sayesinde, her iki ventrikülün kasları aynı anda kasılır [6]

2.4 Kalbin Mekanik Aktivitesi

Kalbin işi kan pompalamaktır. Kapalı bir hidrolik sistem olan dolaşım sistemi, kanın vücut içinde dolaşması fonksiyonunu üstlenir. Bunu gerçekleştirmek için de kalp kasları ritmik ve sistematik olarak kasılırlar ve her sistolde yaklaşık 70 ml kan pompalanır. Dakikada 70 atıma sahip olan bir kalp bu süre içinde yaklaşık beş litre kan pompalamaktadır (70 x 70 ml = 4900 ml). Kalp kapakçıkları ve odacıklar, her bir atım için mükemmel bir senkronizasyonla çalışırlar. Her bir atım yaklaşık olarak 0.8 saniye sürer ve aynı olaylar zinciri her bir atım için tekrarlanır [7].

Kalbin mekanik aktivitesi miyokart hücrelerinin kasılmasını, kalp kapakçıklarının açılıp kapanmasını, kalp odacıklarından kanın pompalanması ve odacıklara kan dolması olaylarını kapsar. Bu aktiviteler sırasında oluşan titreşimler, steteskop ile dinlenebilir ya da fonokardiyografi ünitesi ile dış ortama alınabilir.

Tek bir atıma karşılık gelen kardiyak döngüyü 3 ayrı fazda inceleyebiliriz. Diastol (gevşeme), atrial sistol (atriumun kasılması) ve ventriküler sistol (ventrikülün kasılması). Şekil 2.6'daki zaman çizgisinde, bu kardiyak döngüdeki her bir fazın yaklaşık süresi verilmiştir.



Şekil 2.6: Kardiyak döngüdeki fazların süresi [7]

Diastol:

Bu faz süresince, kan kalbe akar. Kan, pulmoner toplardamarlarla sol atriyuma ve üst/alt toplardamarlarla da sağ atriyuma gelir. Sağ ve sol atriumlardaki kan, mitral ve triküspit kapaklardan doğruca ventriküllere akar. Ventriküller gelen bu kanla dolar. Diastol sonunda ventriküller %80 oranında dolmuş haldedir. Bu fazda bütün

kardiyak kaslar gevşer. Aortik ve pulmoner kapaklar, bu fazda, kanın aorttan ve pulmoner arterden geri akışını engellemek için kapalıdırlar [7].

Atrial Sistol:

Diğer fazlarla kıyaslandığında daha kısa sürer. Bu fazda kan, sol ve sağ atriumdan ventriküllere pompalanır.

Ventriküler Sistol:

Bu fazda ventriküller kasılır. Kasılma başlangıcında mitral ve triküspit kapaklar, atriumlara geri akışı önlemek için kapanırlar. Aortik ve pulmoner arter kapaklar ise açılır. Bu faz sonunda aortik ve pulmoner arter kapakları kapanır ve tekrar diastol başlar.

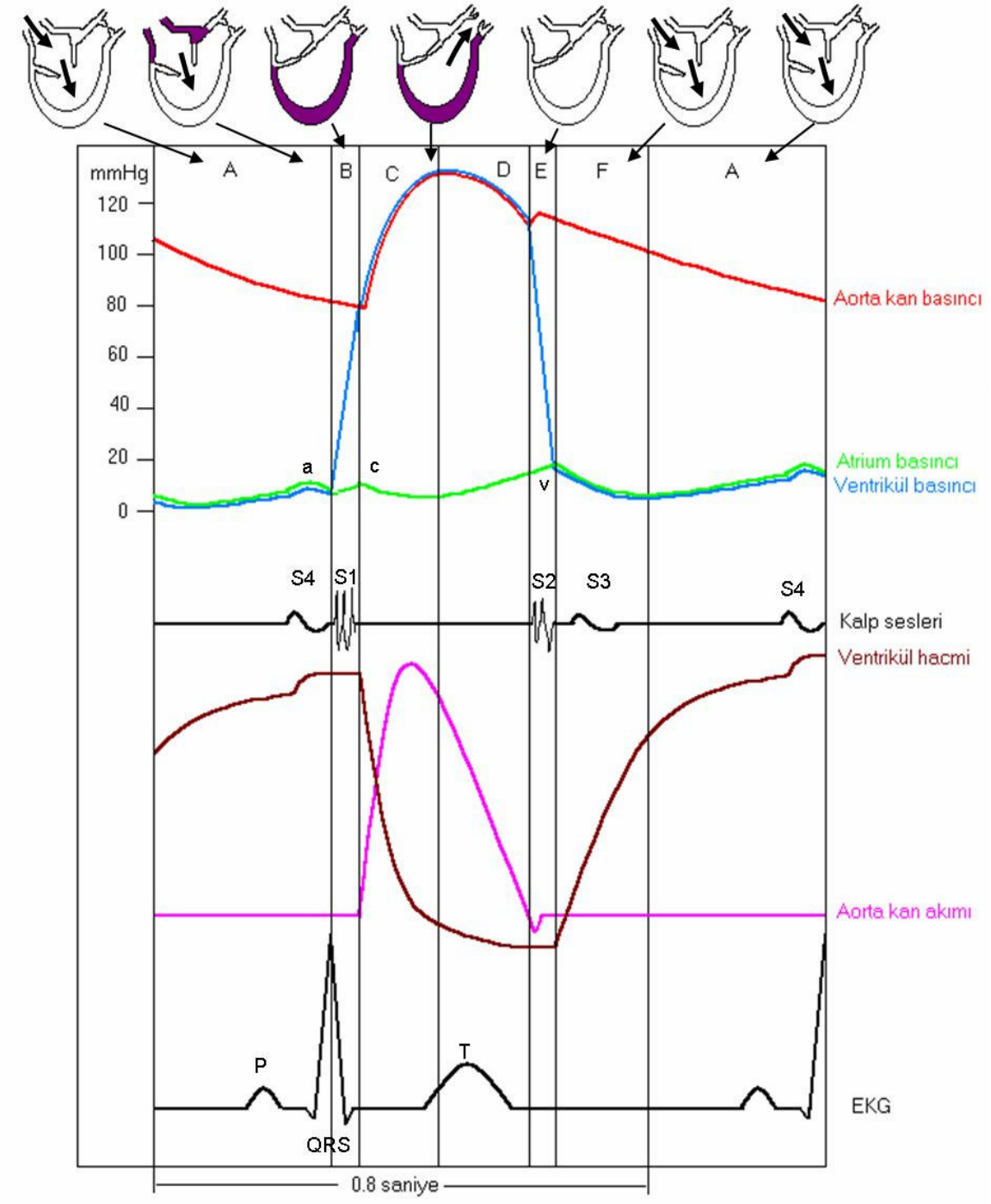
2.5 Kalbin Elektriksel-Mekanik Aktivitesinin İncelendiği İşaretler

Şekil 2.7, bir kardiyak döngü süresince gözlenen mekanik, akustik ve elektriksel işaretlerin ilişkilerini özetlemektedir (kalbin sol lobunda) .

Elektriksel R dalgası, ventriküler kasılmanın başlangıcından önce gelen ventriküler depolarizasyonu gösterir. Ventriküler kasılma sol ventrikül basıncının bir basamak yükselmesine neden olur. Ventriküler basınç atrium basıncını geçer geçmez mitral kapak kapanır ve bu anda S1 titreşimi başlar [3]

Ventriküler basınç aortik basınçtan fazla olduğunda, aortik kapak açılır ve kan ventrikülden aorta akmaya başlar. Ventriküldeki kan boşaldığında, ventrikül basıncı aortik basıncın altına düşer ve aortik kapak kapanır. Bu esnada S2 titreşimleri duyulur.

Ventrikül basıncı daha da azalıp atrial basıncın altına düştüğünde mitral kapak açılır ve atriumdan ventriküle kan dolmaya başlar (Hızlı dolum fazı = Rapid filling phase). Bu esnada S3 titreşimleri duyulabilir.



Şekil 2.7: Kalpteki mekanik, akustik ve elektriksel kardiyak sinyallerin bir kardiyak döngüde zamana göre ilişkileri [3]

2.6 Kalp Sesleri

Bir sesin oluşması ve duyulabilmesi için üç sisteme gerek vardır: Titreşimlerin oluşması, titreşimlerin iletilmesi ve onların farkına varılışı. Sesi oluşturan titreşimlerin dört karakteri vardır: Frekans, şiddet, kalite ve süre [8].

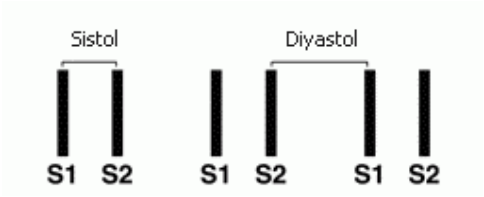
Frekans: Kulağın duyabileceği frekanslar 20 ile 20000 Hz arasındadır. Yaşlanmakla üst sınır daha aşağıya iner. Normalde 20 Hz'in altındaki frekanslı sesler duyulamazlar. Frekansı düşük fakat genliği büyük sesler duyulmamalarına rağmen elle fark edilirler. Kalp sesleri genel olarak 10–500 Hz frekanslı seslerdir, yani alçak frekanslıdırlar. Üfürümler daha yüksek frekanslıdır, 1000 Hz'e kadar çıkabilirler.

Şiddet: Sesin şiddetini belirleyen titreşimlerin genliğidir. Genlik ne kadar fazla ise şiddet de o kadar fazladır. Genlikleri aynı olsa bile yüksek frekanslı sesler alçak frekanslılardan daha iyi duyulurlar ve sanki daha şiddetliymiş gibi sanılabilirler.

Kalite: Sesin kalitesi saflık derecesine bağlıdır. Titreşim sadece tek bir frekansta kalmakta ise, duyulan ses saf ve pürüzsüzdür. Fakat titreşen cisim çeşitli frekans ve genliklerde titreşimler oluşturmakta ise (kalpte olduğu gibi) duyulan ses saf bir ses değildir, bir gürültüdür.

Süre: Titreşimlerin süresi sesin süresidir. Çevredeki dokular kalp seslerinin kısa süreli olmalarını sağlarlar.

Sesin yukarıda anlatılan özellikleri dikkate alınarak, kalp sesleri şiddet, zaman, kalite ve sistol/diastol süreleri açısından incelenir. Amaç: normal kalp seslerini ayırt edebilmek ve patolojik sesleri tanıyabilmektir. Kalp dinlenirken öncelikle kalp hızı ve ritmi incelenir. Sonra birinci ve ikinci kalp sesi ayırt edilerek sistol ve diastol belirlenir. Birinci kalp sesi ile ikinci kalp sesi arası sistol (kasılma), ikinci kalp sesi ile bir sonraki birinci kalp sesi arası diastoldür (gevşeme). Şekil 2.8 'de sistol ve diastol evreleri gösterilmektedir.

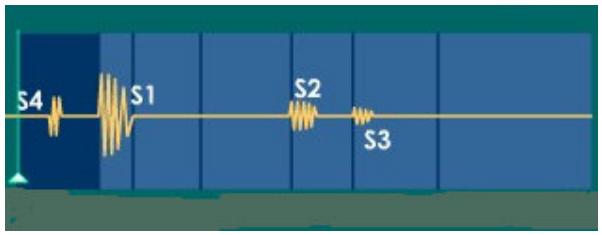


Şekil 2.8: Sistol ve diyastol

İstirahat sırasında, yani kalp hızının yüksek olmadığı zamanlarda sistol, diastolden kısadır.

Daha sonra sırası ile kalp seslerindeki değişiklikler (ikilenme vb), ek seslerin olup olmadığı (3. ve 4. kalp sesinin varlığı, açılma sesleri, sürtünme sesi) ve üfürüm olup olmadığı araştırılır.

Şekil 2.9 'da dört kalp sesi birlikte görülebilir.



Şekil 2.9: Kalp sesleri [9]

Kalbin dinlenmesi sırasında kulağın duyabildiği ses ve üfürümler düşük, orta ve yüksek titreşimli olmak üzere 3 grupta incelenir [3].

1- Düşük titreşimli sesler ve üfürümlerde titreşim sayısı saniyede 30 – 60 arasındadır. S3, S3G (3. Kalp sesi dalgalanması), S4G (4. Kalp sesi dalgalanması),

2- Orta titreşimli sesler ve üfürümlerde titreşim sayısı saniyede 60 – 150 arasındadır: S1 (1. Kalp sesi), S2 (2. Kalp sesi), sistolik üfürümler.

3- Yüksek titreşimli sesler saniyede 150'nin üzerindeki titreşimlerdir: Aort ve pulmonalis yetmezliğinin diastolik üfürümleri ile müzikal ses ve üfürümler.

2.6.1 Normal Kalp Sesleri

S1, birinci kalp sesidir. Atrio-ventriküler (solda mitral ve sağda triküspit) kapakların kapanması esnasında apeksten (mitral odaktan) duyulur. Bu kapanma, kanda ve kalp duvarlarında titreşim yaratır. Bu titreşimler de göğüs duvarına iletilir. S1 duyulan en yüksek sestir; S2 den daha kuvvetli, uzun süreli ve düşük frekanslıdır. Fonetik olarak "LAB" şeklinde duyulur.

Basitçe, birinci sesin mitral ve triküspit kapakların kapanması sırasında oluştuğu varsayılır. Ancak incelendiğinde birinci sesin sadece kapakların kapanması sonucu oluşan bir ses olmadığı, üç veya dört titreşimden oluşan genişçe bir ses olduğu görülmüştür. EKG ile eş zamanlı yapılan kayıtlarda, birinci ses başlangıcının Q dalgasından yaklaşık 0.04 saniye sonra başladığı görülür. İlk dalgalar küçük genlikli ve alçak frekanslıdır. Bu titreşimler ventriküllerin izometrik kasılma döneminde

oluşmaktadır. Bu küçük titreşimlerden sonra gelen yüksek genlikli ve yüksek frekanslı titreşimler mitral ve triküspit kapakların kapanması ile oluşmaktadır. Mitrale ait olan titreşimler öndedir. Bazen mitral ve triküspite ait olan titreşimler birbirlerinden ayrı olarak kaydedilir. Birinci sese karakterini veren kapak bileşenleridir ve bu büyük titreşimlerden sonra birinci sesin son kısmı, kanın aorta ve pulmonere doğru atılması sırasında oluşan küçük genlikli titreşimler kaydedilir [8].

S2, ikinci kalp sesidir. İkinci sesi oluşturan semilunar kapakların kapanmasıdır. Aort ve pulmonalis odaklarında en güçlü duyulur. Aort kapağına ait olan ses (A2) pulmonere ait olandan (P2) daha kuvvetlidir. Normalde aort kapağı pulmoner kapaktan önce kapanır, ancak aradaki süre çok kısa olduğundan bu fark edilmez ve S2 tek ses gibi duyulur. S1 den daha kısa ve "DAP" hecesi niteliğinde tanımlanmaktadır.

2.6.2 Anormal Kalp Sesleri

S3, üçüncü kalp sesidir. Diyastol başında ventriküllerin hızlı doluş döneminde atrio-ventriküler halka birden yükselmekte ve kordonlar ile kapaklar gerilip bu sesi oluşturmaktadır. Hasta sol yan yattığında steteskobun çan kısmı ile apeksten (mitral odaktan) dinlenebilir. S2 den (aort kapağının kapanma sesinden) yaklaşık 150 ms sonra duyulur. Sol ventriküler yetmezlikte duyulur, ventriküler gallop da denir. Atriumlardan kan ventriküle hızla atılırken içerde kalan volüme çarpar ve dolgun bir ses oluşur. Düşük frekans ve düşük genliklidir. Hamile, çocuk ve yaşlılarda patoloji olmadan da duyulur. Fonetik olarak " Dİ" şeklinde duyulur.

S4, dördüncü kalp sesidir. Normalde kulakla duyulmaz. Ventriküller dolu iken basınç atriumlara yayılır. Bu basınç atriumdaki kanı dalgalandırır, duyulan bu dalgalanma sesidir. En iyi apeksten ve S1'den hemen önce duyulur. Akut miyokart enfarktüsüne bağlı gelişir. Koroner arterlerdeki bir hastalık, atrio-ventriküler yetmezlik sonucu ortaya çıkar. Fonetik olarak "Tİ" şeklinde duyulur [3].

Gallop ritmi (S4) : S3 ve S4 ün; S1 ve S2 ile birlikte duyulmasıdır. Nabız dakikada 110 un üstünde olup, koşan at sesi gibi ses duyulur.

Murmur (üfürüm) Anormal girdap sonucu oluşan seslerdir. Frekans aralığı yaklaşık 100–500 Hz arasındadır. Kan, dar ya da düzensiz kapaktan geçerken girdap oluşur. Girdaba yol açan nedenler:

1- Kanın akışı yönünde bir lezyon varsa, düzgün akış bozularak girdaplar oluşur ve üfürümler ortaya çıkar.

2- Kanın akışı yönünde etraftaki yapılarda fiziksel değişiklikler varsa, titreşimler meydana gelir ve üfürümler duyulur.

3- Kan, dar bir aralıktan geçerken,

4- Kan, dar olmayan fakat pürüzlü bir kapak sistemi içinden geçerken,

5- Normal bir kapak sistemi içinden geçen akımın hızlanması halinde,

6- Dar bir yerden geniş bir sisteme geçtiği zaman üfürümler duyulur. Murmurlar zaman, yoğunluk, yer, frekans, kalite ve yayılmalarına göre sınıflandırılırlar [3].

Kardiyak seslerin faz, karakter, frekans, süre, meydana geldiği yer ve mekanizma bilgilerini Tablo 2.1 ile özetleyebiliriz.

Tablo 2.1. Kardiyak Sesler

	Faz	**Karakter**	**Frekans aralığı (Hz)**	**Süre (s)**	**Meydana geldiği yer**	**Mekanizma**
S1	İzovolumetrik kasılma fazı	Basık	~10-60	~0.1	Apikal bölge	Mitral ve triküspit kapakların kapanması
S2	İzovolumetrik gevşeme fazı	Keskin	~60-150	~0.08	Bazal bölge	Aortik ve Pulmoner kapakların kapanması
S3	Ventriküler hızlı dolum fazı sonu	Zayıf ve Basık	~150<	~0.04 (S2'den yaklaşık 0.12–0.18 s sonra duyulur)	Apeks (üst ve alt)	Ventriküler titreme
S4	Ventriküler diyastol fazı sonu	Zayıf	~150<	~0.1 (S1'den hemen önce duyulur)	Apeks	Atrium kasılması

2.6.3 Kalp Seslerinde Görülen Farklılıklar ve Nedenleri

Birinci Sesin Şiddetinde Değişmeler: Atrio-ventriküler kapak (mitral ve triküspit kapaklar) yapısındaki değişiklik, kalınlaşma; miyokardın kasılma gücü, AV kapakların kasılma sonunda duruş yeri birinci sesin şiddetini etkiler [8].

Diyastol ne kadar kısa, ventrikül ne kadar az dolmuşsa veya ventrikül içi basınç ne kadar yüksekse, birinci ses o kadar kuvvetlidir. Çünkü bu durumlarda atrio-ventriküler kapaklar kuvvetle kapanmaktadırlar. Taşikardilerde, ekstra sistollerde, P-R aralığının kısaldığı hallerde birinci ses şiddeti artar, çünkü atrio-ventriküler kapaklar birbirlerine yaklaşma fırsatı bulamadan birden kapanırlar.

Diyastol sonunda ventrikülün kanla fazla dolduğu hallerde atrio-ventriküler kapaklar birbirine iyice yaklaşmış olduklarından kapanmaları daha sessiz olmaktadır. P-R aralığının uzadığı hallerde, bradikardilerde, miyokart yetersizliklerinde, mitral yetersizlikte birinci ses zayıflar.

Atrial fibrilasyonda, P-R aralığının her atımda değişik olduğu durumlarda, ekstra sistolik aritmilerde birinci ses şiddeti her atımda değişiktir.

Birinci Sesin Çiftleşmesi: Normalde sol ventrikül daha önce depolarize olduğu için, sağ ventrikülden daha erken kasılır ve mitral triküspitten, aort ise pulmonerden önce kapanır. Ancak, bu iki kapağın kapanması arasındaki süre çok kısa olduğundan tek bir ses (S1) olarak duyulur. Sağ dal bloğu, Ebstein anomalisi (Triküspit kapağın anatomik olarak sağ ventrikül içine doğru yer değiştirmiş olmasıdır. Bu nedenle sağ atriyum büyür, sağ ventrikül küçülür, triküspit kapakta fonksiyon bozukluğuna bağlı yetersizlik saptanır) ve Wolff Parkinson White (WPW, sino-atrial düğümden çıkan iletilerin normal ileti yolları dışında aksesuar yollardan atrio-ventriküler düğüme gitmeden direk ventriküle ulaşması) durumlarında ise birinci ses bölünür, yani çift ses gibi duyulur.

Dördüncü sesin duyulması birinci sesin çift zannedilmesine sebep olabilir.

İkinci Sesin Şiddetinde Değişmeler: İkinci sesin şiddetini, aort ve pulmoner bileşenlerin şiddeti olarak ayrı ayrı ele almak gerekir. Aort kapağı genişlediğinde A2 sesi, pulmoner kapak genişlediğinde ise P2 sesi şiddetlenir. Aort kapağı daraldığında A2, pulmoner kapak daraldığında ise P2 sesi şiddetinde azalma görülür.

İkinci Sesin Çiftleşmesi: Aort kapağı pulmoner kapaktan önce kapanır ve normalde nefes alma sırasında pulmoner kapak geç kapandığından P2 ile A2'nin arası açılır. Bu duruma fizyolojik çiftleşme denir ve normaldir. Nefes verme sırasında ise A2 ve P2 birbirine yaklaşır ve tek ses olarak duyulur. Nefes alma ve vermeden etkilenmeyen bölünme ise sabit çiftleşmedir ve anomali olarak değerlendirilir [8].

Üfürümler: Bir kapak daraldığında ya da hasar gördüğünde, normal sistol ya da diastol sırasında, kan akışının düzgün olmamasından kaynaklanan gürültüler oluşabilir. Bu gürültüler göğüs üzerinden her yerde duyulamayabilir. Gürültünün iyi duyulduğu yerin ve hangi yöne doğru yayıldığının tespit edilmesi, üfürümün nedenini anlamak için önemli bir ipucudur. Diğer önemli bilgi ise, üfürümün sistolde mi yoksa diastolde mi olduğunun ayırt edilmesidir. Üfürüm bu fazlar süresince mi, yoksa başlangıç ve bitişlerinde mi oluyor?

Tespit edilen üfürümün yeri, şiddeti ve frekansı hastalık tespitinde önem taşımaktadır. Üfürümlerin frekansı 100–500 Hz arasında değişmektedir.

Kalp seslerindeki normal olmayan değişikliklerden faydalanarak kalp rahatsızlıklarını tespit edebilmek için, her bir sesin süresinin, sistol ve diyastol sürelerinin, bir sesin bitimiyle diğer sesin başlangıcı arasındaki sürenin tespit edilmesi gerekir. Tüm bu veriler, birbirlerine oranlanarak ve normal değerleriyle karşılaştırılarak bir sonuca ulaşılabilir.

Sağlıklı bir kalpte, kalp hızı arttıkça kasılma hızı ve kuvveti de artar. Ancak kalp yetmezliği olduğunda kasılma hızı azalır, yani frekans azalır. Kalp hızı aşırı arttığındaysa kasılma hızı artar, frekans artar. Bu azalma/artma kalpteki problemin derecesine bağlı olarak değişir.

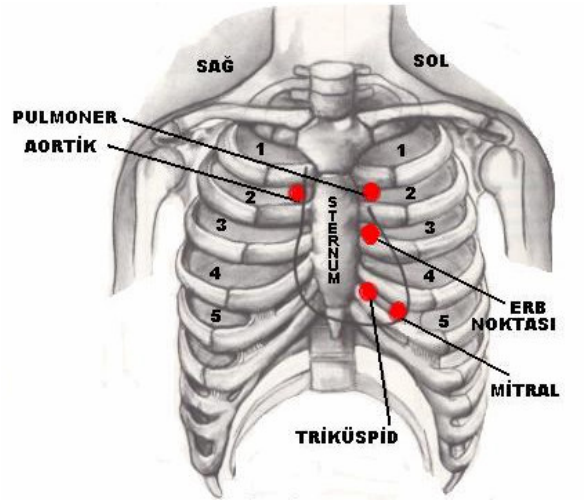
2.7 Dinleme Yöntemi

Hasta sessiz, iyi aydınlatılmış, ısı konforu uygun bir ortamda muayene edilmelidir. Hasta, sırtüstü veya yarı yatar, kolları yanlarda (uzatılmış) olmalıdır. Kalp dinlenirken hasta sakin ve sessiz nefes alıp vermeye devam etmelidir [3].

2.7.1. Kalp Seslerini Dinleme Odakları

Kapak hastalıklarında dinleme bulguları, kapaklara ait seslerin en iyi duyuldukları yerler dinlenerek anlaşılır. Bu bölgelere "dinleme odakları" denir. Dinleme odakları, kapaklardan geçen kanın yayılması nedeni ile o kapağa ait üfürümlerin göğüs duvarında en iyi duyuldukları yerdir. Bunlar; mitral odak, triküspit odak, aort odağı,

pulmoner odak ve mezokardiyak odaktır. Şekil 2.10' da dinleme odakları belirtilmiştir.



Şekil 2.10: Kalp seslerini dinleme odakları [3]

Mitral (apikal) odak: Klavikulanın orta hat çizgisinin sol 5nci interkostal aralığı kestiği bölge mitral (apikal) odaktır. Mitral kapağa ait üfürümler, hasta hafif sola dönük ve nefes alma (inspiryum) esnasında dinlenirse daha rahat duyulur [3].

Triküspit odak: 5nci interkostal aralığın, sternumun sol kenarıyla birleştiği bölgedir. Triküspit odağı nefes alma esnasında dinlenirse sesler daha rahat duyulur.

Aort odağı: Sağ ikinci interkostal aralığın, sternumun sağ kenarıyla birleştiği bölgedir. Aortla ilgili üfürümler en iyi bu bölgede duyuluyorsa da, bazen apeksten, sternumun sol kenarından hatta karotid arterlerin bulunduğu bölgeden de güçlü duyulabilir. Hasta biraz öne eğik ve ekspirasyonda(soluk verirken) en iyi dinleme sağlanır.

Pulmoner odak: Sol 2nci interkostal aralığın, sternumun sol kısmı ile birleştiği bölgedir. Üfürümler en güçlü inspiryum esnasında duyulur.

Erb noktası (mezokardiyak odak): Sternumun solunda 3ncü interkostal aralıktadır. Aort ve pulmonalis yetmezliğinin en iyi duyulduğu yerdir.

3. FOURIER ANALİZİ ALGORİTMALARI

3.1 Fourier Dönüşümü

İşaret işlemede kullanılan analiz yöntemleri karşılaştırıldığında, Fourier analizinin en çok kullanılan yöntem olduğunu söyleyebiliriz. Fourier dönüşümü ile zaman domenindeki işaret frekans domeninde ifade edilmektedir. Fourier dönüşümü yapıldığında, zaman domenindeki bir işaret hakkında bilinen her şey, frekans domeninde de aynen korunmaktadır [12].

Aşağıda 1-boyutlu sürekli Fourier dönüşümünün matematiksel tanımı verilmiştir.

$$S(w) = \int_{-\infty}^{\infty} s(t) e^{-jwt}\, dt$$

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(w) e^{jwt}\, dw \qquad , w = 2\pi f \tag{3.1}$$

Burada t, zamanı; w açısal frekansı; f frekansı belirtmektedir. s, zaman domenindeki sürekli değişen işaret, S de bu işaretin Fourier dönüşümü uygulanarak elde edilmiş frekans domenindeki karşılığıdır. (3.1) eşitliğinde, üstte s(t)'nin Fourier dönüşümü, altta ise ters Fourier dönüşümü verilmiştir.

Fourier dönüşümü sonsuz sayıda örnek gerektirdiği için, bilgisayar hesaplamasına uygun değildir. Sayısal olarak hesap yapabilmek için sonlu sayıda örnekle çalışmak gerektiğinden, yeni bir dönüşüm, Ayrık Fourier dönüşümü tanımlanmıştır. İşaretin sabit bir pencere içerisinde ayrık olarak tanımlanması durumunda aşağıdaki eşitlikler elde edilir.

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-j\frac{2\pi kn}{N}}$$

$$n, k \in Z\ ,\ \text{N = pencere genişliği} \tag{3.2}$$

$$s(n) = \frac{1}{N} \sum_{n=0}^{N-1} S(k) e^{j\frac{2\pi kn}{N}}$$

Burada s(n), ayrık zaman domeni işareti, S(k) ise ayrık frekans domeni işaretidir. S(k) katsayıları, Fourier serisini ifade eden katsayılara yaklaşık olarak eşittir [12].

Fourier dönüşümü yapılırken hesaplamanın uzun zaman alması nedeniyle, işlem süresini kısaltacak teknikler geliştirilmiştir. Bu amaçla geliştirilen tekniklerden biri, hızlı Fourier dönüşümüdür.

3.2 Hızlı Fourier Dönüşümü

Hızlı Fourier dönüşümü, ayrık Fourier dönüşümünü temel alan kullanışlı bir dönüşüm yöntemidir. Sayısal işaret işlemede, işaretin frekans spektrumunu elde etmede en çok kullanılan dönüşümlerden bir tanesidir.

Hızlı Fourier dönüşümü de kendi içinde çeşitlilik göstermektedir. Winograd dönüşümü, ayrık kosinüs dönüşümü (discrete cosine transform) ve ayrık Hartley dönüşümü bunlardan bazılarıdır [13].

Bu bölümde, hızlı Fourier dönüşüm yöntemlerinden en basiti olan Radix–2 algoritması incelenecektir.

(3.2) eşitliğini radix–2 algoritmasını açıklamak için yeniden yazalım.

$$S(k) = \sum_{n=0}^{N-1} s(n) W^{nk} \quad , \text{k= 0,1,2,…..N-1} \tag{3.3}$$

Burada s(n) = s(nT) ‘dir. Yani örnekleme periyodu T, ayrık zaman işareti s(n)’de bulunmaktadır. N, pencere uzunluğudur.

(3.3) eşitliğinde W sabiti,

$$W = e^{-j\frac{2\pi}{N}} \tag{3.4}$$

ifadesine karşılık gelmektedir.

(3.4) eşitliğinden W sabitinin değerinin, N pencere büyüklüğüne bağlı olduğu görülmektedir. Bu durumda, belirli bir k ayrık anındaki S(k) değerini elde etmek için, yaklaşık olarak N kompleks toplama ve N kompleks çarpma işlemi yapmak gerekecektir:

$$S(k) = s(0) + s(1)W^{k} + s(2)W^{2k} + s(3)W^{3k} + \ldots + s(N-1)W^{(N-1)k} \tag{3.5}$$

k = 0,1,…,N-1 için.

Öyleyse, S(0),S(1),S(2),…,S(N-1) değerlerini hesaplamak için, N^2 adet toplama ve N^2 adet de çarpma işlemi yapmak gerekecektir. Bu durum, özellikle büyük N değerleri için hesaplamayı zorlaştıracaktır [13].

Hızlı Fourier dönüşüm algoritması ise, W pencere sabitinin periyodiklik ve simetri özelliğini kullanarak (3.3) eşitliğinin hesaplanmasında büyük kolaylık sağlayacaktır (Çünkü işaretin seçilen her pencere içinde kendini tekrarladığı varsayılıyor) .

W sabitinin periyodik olma özelliğinden,

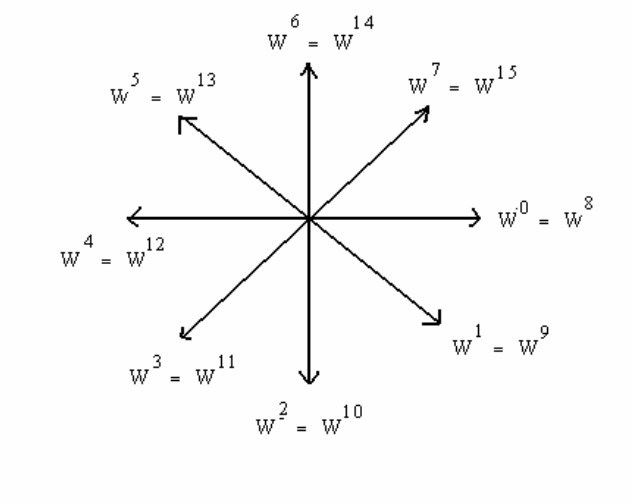
$$W^k = W^{k+N} \tag{3.6}$$

ve simetrik olma özelliğinden,

$$W^k = -W^{k+N/2} \tag{3.7}$$

yazılabilir.

Şekil 3.1, N =8 için periyodik ve simetrik W sabitlerini göstermektedir [13]



Şekil 3.1: N = 8 için periyodik ve simetrik W pencere sabiti [13]

Hızlı Fourier dönüşümü ile dönüşüm işlemi küçük dönüşümlere ayrıştırılır: N örnekli ayrık Fourier dönüşümü ile iki N/2 örnekli ayrık Fourier dönüşümü yapılır ve elde edilen bu iki ifadenin her birine yine iki N/4 örnekli ayrık Fourier dönüşümü uygulanır. Yapılacak en küçük dönüşüm değeri, hızlı Fourier dönüşümünün kökü

(radix) ile belirlenir. Örneğin radix–2 dönüşümünde N, 2'nin katı olmalıdır ve yapılacak en küçük dönüşüm N/2'ye kadardır [13].

Fourier dönüşümünün sayısız niteliklerinin yanında, yetersiz kaldığı durumlar da bulunmaktadır. (3.1) eşitliğini incelersek, zamanla sürekli değişen, durağan olmayan bir s(t) işareti, belirli bir f frekansındaki üstel bir terim ile çarpılmış ve eksi sonsuzdan artı sonsuza tüm zamanlarda integrali alınmıştır. Yani işaretin frekans bileşeninin hangi zamanda ortaya çıktığı dönüşüm sonucunu etkilemeyecektir.

Bir örnekle açıklamak istersek, zamanla değişen iki sinyalimiz olduğunu düşünelim (Şekil 3.1.a ve Şekil 3.1.b de görüldüğü gibi).

s1(t) ve s2(t) , zamanın değişik anlarında değişik frekans bileşenlerine sahip sürekli işaretler olsun.

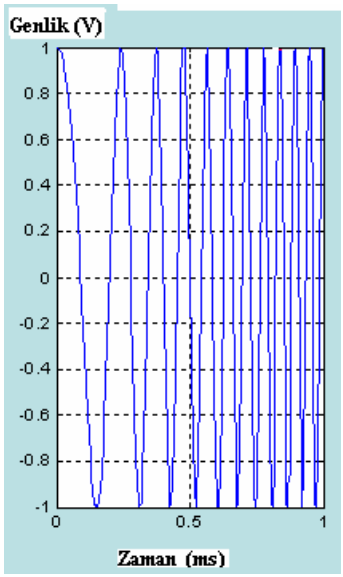
s1(t) $= Cos(2\pi * 2 * t)$, $0 \le t \le 0.5ms$ ve

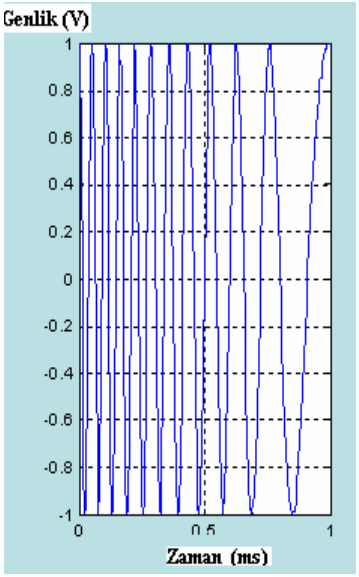
s1(t) = $Cos(2\pi * 20 * t)$, $0.5 \le t \le 1ms$ olsun.

s2(t) ise,

s2(t) $= Cos(2\pi * 20 * t)$, $0 \le t \le 0.5ms$ ve

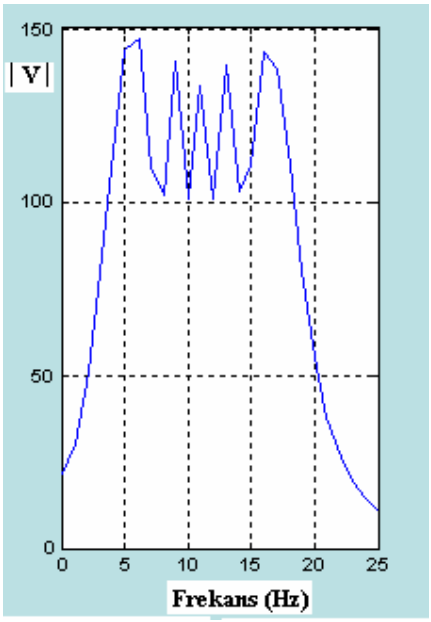
s2(t) = $Cos(2\pi * 2 * t)$, $0.5 \le t \le 1ms$ olsun

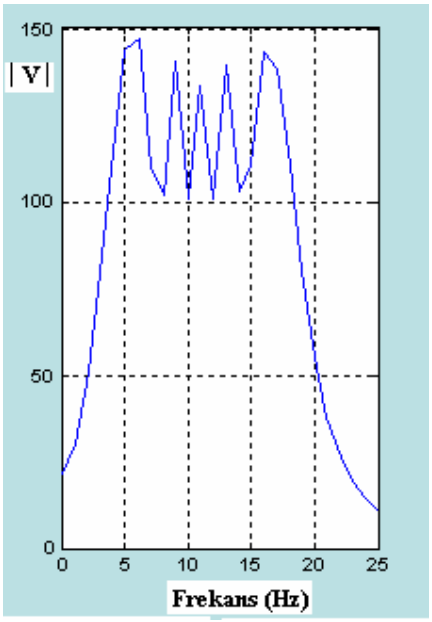




Şekil 3.2.a: Zamanla değişen işaret s1(t) **Şekil 3.2.b**: Zamanla değişen işaret s2(t)

Her iki sinyal de aynı sürelere sahip üç sinusoidal bileşenden oluşmaktadır. Bu iki sinyalin birbirinden tek farkı, bu sinusoidal bileşenlerin işarette görülme sırasıdır. Şu da aşikârdır ki, bu fark, iki işareti birbirinden ayırabilmek için önemli bir bilgidir.





Şekil 3.3.a: s1(t)'nin Fourier Dönüşümü **Şekil 3.3.b**: s2(t)'nin Fourier Dönüşümü

Her iki işaretin Fourier dönüşümü alındığında elde edilen frekans spektrumların aynı olduğu görülür (Şekil 3.2.a ve b).Bu durumda, frekans domeninde iki işareti birbirinden ayırmamızı sağlayacak karakteristik bilgi kaybolmuştur. [12]

Sonuç olarak, durağan olmayan işaretlerin Fourier dönüşümü ile analizinde zamanda lokalizasyon bilgisi kaybedilmiş olacak. Bu durumda, Fourier dönüşümüne lokalizasyon bilgisini de tutabileceğimiz yeni bir tanımlama yapılması gerekmiştir: Kısa-Zaman Fourier dönüşümü (KSFD).

3.3 Kısa-Zaman Fourier Dönüşümü

Kısa-zaman Fourier dönüşümü (KZFD), bir sinyalin içinden sırayla alınan pencerelerin Fourier dönüşümlerinin hesaplanması ile gerçekleştirilir. Kısa-zaman Fourier dönüşümü, tanım kümesi aynı anda zaman ve frekans olan işaretler hakkında bilgi vermek için kullanılabilir. Bu dönüşümde giriş işareti bölümlere ayrılır ve her bölüm frekans açısından ayrı ayrı analiz edilir [12].

$$S_{KZFD}(\tau,\omega) = \int_{-\infty}^{\infty} s(t)w^*(t-\tau)e^{-j\omega t}dt \qquad (3.8)$$

Burada $w(t-\tau)$ zaman ekseninde bir τ noktasında ötelenmiş w(t) pencere fonksiyonu (genellikle dikdörtgen, Hann ya da Gaussian) ve s(t), dönüşümü yapılacak işarettir. $S_{KZFD}(\tau,\omega)$, zaman ve frekans domenindeki işaretin faz ve genliğini ifade eden kompleks $s(t)w^*(t-\tau)$'nin Fourier dönüşümüdür [12].

(3.8) ifadesinde $w(t-\tau)$, s(t) işaretini pencereleyecek w(t) pencere fonksiyonunun zamanda ötelenmiş halidir. Diğer bir deyişle, $w(t-\tau)$, belirli bir zaman aralığında, Fourier dönüşümü yapılacak s(t) işaretinin sadece bir bölümünü alır ve işler. Bu zaman penceresi genellikle gerçek zaman fonksiyonudur, yani,

$$w^*(t-\tau) = w(t-\tau) \qquad (3.9)$$

Kısa-Zaman Fourier dönüşümünde zaman penceresi, s(t) işaretinin bir bölümünü alır ve işaretin seçilen bu kısmına Fourier dönüşümü uygulanır. Öteleme parametresi τ değiştirilerek, işaretin her bir parçasının sadece Fourier dönüşümü değil, aynı zamanda lokalizasyon bilgisi de elde edilmiş olur.

Gerçek zamanda hesaplama yapılabilmesi için sonlu sayıda örnekle çalışmak gerektiğinden sürekli zamandaki s(t) işareti ayrık zamanda incelenmek üzere nicemlenir. Her bir bölümün Fourier dönüşümü alınır. Kompleks bileşenler, zaman ve frekansla değişen her bir bileşenin faz ve genlik bilgisinin tutulduğu matrise eklenir. Bu işlemler (3.10) ifadesinde matematiksel olarak gösterilmiştir.

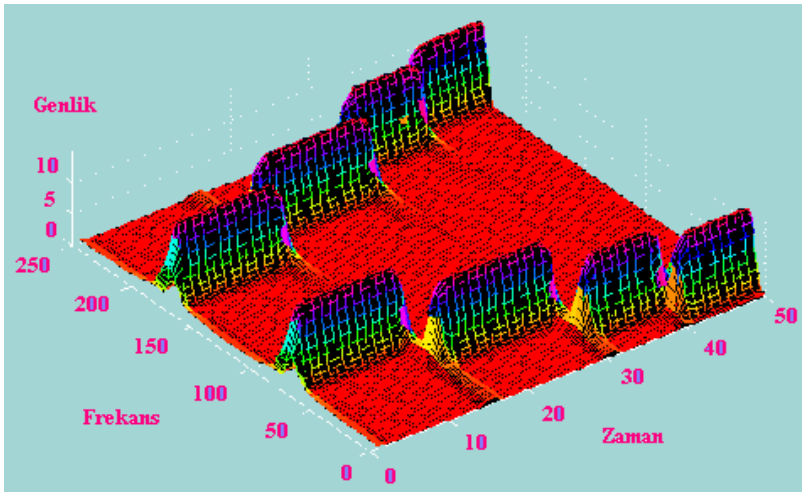
$$S(m,k) = \sum_{n=0}^{N-1} s(n)w(n-m)e^{-\frac{2\pi kn}{N}} \qquad (3.10)$$

$$n = 0,1,\ldots,N-1$$

Burada s(n) zamanla değişen s(t) işaretinin ayrık zamandaki ifadesi ve w(n) pencere fonksiyonudur. m, seçilen pencere fonksiyonunun ötelenme miktarıdır.

Kısa-Zaman Fourier dönüşümü ile ilgili analiz penceresinin seçimi sinusoidal bileşenlerin ortaya çıkarılmasını ve spektrumun düzgünlüğünü belirler.

Kısa-zaman Fourier dönüşümünü alınan bir işaretin frekans bileşenlerinin yanında lokalizasyon bilgisinin de saklandığı Şekil 3.4'te görülebilir.



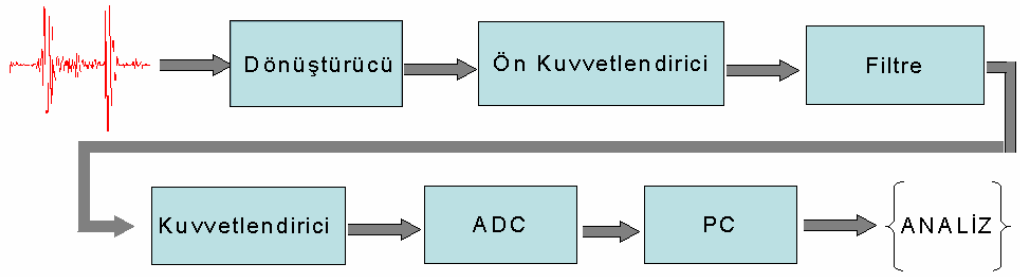
Şekil 3.4: Kısa-zaman Fourier dönüşümü uygulanan işarette frekans, zaman ve genliğin üç boyutlu gösterimi

Kısa-zaman Fourier dönüşümünde, işaretin incelendiği pencere boyutu önem taşımaktadır. Seçilen pencere boyutu arttıkça, frekansta çözünürlük yakalanacak, ancak zamanda durağanlık kaybedilecek yani çözünürlük azalacaktır. Pencere boyutu azaldığında ise zamanda çözünürlük artacak ancak frekansta çözünürlük azalacaktır. Kısa-zaman Fourier dönüşümünün bu özelliği göz önünde bulundurularak, dönüşümün uygulanacağı işaretten elde edilmek istenen bilgiye göre en uygun pencere boyutu belirlenmelidir.

4. KALP SESLERİ İÇİN VERİ TOPLAMA SİSTEMİ

Yaşayan organizmalarla ilgili büyüklüklerin ölçülmesinde, ölçüm sistemi ile üzerinde ölçüm yapılacak canlı organizma arasındaki etkileşim nedeniyle, canlı organizmanın da ölçüm sisteminin bir parçası olarak kabul edilmesi gerekir [1]. Ancak canlı organizmada doğal olarak bulunan veriler (basınç, sıcaklık, iyon, akış veya herhangi bir biyolojik bulgu) biyolojik büyüklüklerdir, ölçüm sistemi ise elektriksel işaretleri kullanmaktadır. Bu nedenle, ölçüm yapabilmek için, canlı organizmadan alınacak biyolojik büyüklüklerin, öncelikle uygun dönüştürücülerle elektriksel işarete dönüştürülmeleri gerekir. Elektriksel forma dönüştürülen işarete artık istenen işlem uygulanabilir.

Örneğin, bu çalışmada incelediğimiz ses işaretlerinin genliği çok küçük olduğundan, elektriksel forma dönüştürüldükten sonra kuvvetlendirilmesi gerekir. Bununla birlikte, çeşitli nedenlerle var olan gürültülerden arındırılması yani filtrelenmesi de gerekir. Filtrelenmiş işaretin genliğini uygulamalarda kullanılabilir seviyeye çekebilmek için de ayrıca bir kuvvetlendirici kullanılmasına gerek duyulabilir. Son olarak, elde edilen analog işareti bilgisayarda işlenir forma getirmek için de bir analog dijital çevirici kullanılır ve işaret sayısal ortamda işlenmeye hazır hale gelir. Tüm bu işlemler sırasıyla Şekil 4.1 de gösterilmiştir.



Şekil 4.1 Fizyolojik işaretin sayısal ortama aktarılmasını gösteren blok diyagram

4.1 Dönüştürücü Özelliklerinin Sistem Üzerine Etkileri

Dönüştürücü, ölçülen büyüklüğü elektriksel işarete çevirmek için kullanılır. Dönüştürülen büyüklük, sıcaklık, basınç, akış ya da herhangi bir fizyolojik büyüklük

olabilir. Dönüştürücü çıkışı daima elektriksel bir işarettir. Olaylar arasındaki ilişkilerin incelenmesi istendiğinde birden fazla dönüştürücü kullanılabilir [1].

Dönüştürücü, ölçme düzeninde hem canlı sistem hem de ölçüm sistemiyle temas halindedir. Bu nedenle dönüştürücünün hem canlı sistemi hem de ölçüm sistemini nasıl etkilediğinin bilinmesi gerekir. Bir dönüştürücünün çalışmasını belirleyen altı tasarım parametresi vardır: Bunlar; örnek yüklenmesi, çıkış empedansı, sönüm, frekans cevabı, doğrusallık ve gürültü olarak sayılabilir.[1]

Kısaca bu parametreleri incelersek:

Örnek yüklenmesi; dönüştürücünün, üzerinde ölçüm yapılan objeye yaptığı etkidir. İdeal olarak dönüştürücü, dönüştürmeye çalıştığı büyüklüğü hiçbir şekilde değiştirmemelidir.

Çıkış empedansı; dönüştürücünün çıkış empedansı, işaret işleme biriminin giriş empedansıyla uyumlu olmalıdır. İşaret işleme birimi çoğu sistemlerde bir kuvvetlendiricidir. Dolayısıyla dönüştürücünün çıkış empedansıyla kuvvetlendiricinin giriş empedansı arasındaki ilişki önemlidir.

Sönüm; dönüştürücünün fizyolojik olayı aslına sadık kalarak izleyemediği durumlarda çeşitli sönüm durumları söz konusudur. Dönüştürücü darbe ya da basamak şeklindeki bir giriş işaretine hızlı ya da yavaş cevap verirse, sönüm noktaları oluşur. İdeali bu giriş işaretlerine ne hızlı ne de yavaş cevap veren, olduğu gibi takip eden bir dönüştürücüdür.

Frekans cevabı; Eğer dönüştürücünün frekans cevabı fizyolojik büyüklüğün bant genişliğinden düşük ise işarette önemli bir bilgi kaybı olur. Eğer frekans cevabı olayın bant genişliğinden büyük ise bu kez de işaretle ilgisi olmayan bir takım işaretler (gürültüler) de alınır ve anlamsız sonular ortaya çıkar. Dönüştürücünün frekans cevabı, sezebildiği ve cevap verebildiği frekans bandı ile tanımlanır.

Doğrusallık; dönüştürücü çıkış işaretinin dönüştürücü girişindeki fizyolojik işareti izleme yeteneğini etkileyen bir özelliktir. Dönüştürücünün geçiş karakteristiği doğrusal ise dönüştürücü çıkışındaki elektriksel işaret, fizyolojik işaretin benzeri olacaktır.

Dönüştürücü gürültüsü; üç ayrı gürültü kaynağı vardır: Fizyolojik gürültü, termal gürültü ve çevre gürültüsü. Burada fizyolojik gürültü, dönüştürücünün, algılayıp ölçmeye çalıştığı fizyolojik değişkenin doğal frekansına yakın frekanslarda meydana gelen diğer fizyolojik değişmeleri sezmesinden kaynaklanır. Bu gibi gürültüler, kas

titreşimi, vücut uzuvlarının hareket etmesi ve diğer organların faaliyetleri gibi durumlarda ortaya çıkar [1].

4.1.1 Kalp Seslerini Algılamak İçin Kullanılan Dönüştürücü

Bu çalışmada, fizyolojik işaret olarak kalp sesleri incelenmektedir. Kalp seslerini vücuttan algılamak için doktorlar steteskop kullanmaktadır. Basit bir mantıkla çalışan steteskop, bir nevi mekanik yükselticidir.

Bu çalışmada ise, ölçüm sisteminin ilk aşaması olarak, sesi elektriksel işarete dönüştürmek için, mikrofon kullanılmıştır. Mikrofon ile havanın titreşimi elektriksel işarete dönüştürülerek sesler elektronik ortama aktarılır. Kullanılan mikrofonun çevresel gürültüleri algılamayacak şekilde yalıtılmış olması gerekir, çünkü ortamdaki sesler algılanmak istenen kalp seslerine eklenerek çıkış işaretini bozar.

Devrede kullandığımız mikrofon (model TC-601T), Nihon Kohden marka bir fonokardiyografi cihazının parçasıdır (Şekil 4.2).



Şekil 4.2 Çalışmada kalp seslerini algılamak için kullanılan mikrofon

TC-601T, hava yalıtımlı bir ses dönüştürücüsüdür. 10–500 Hz aralığında olan kalp sesi işaretlerini algılayabilmektedir. Mikrofonun çıkış empedansı, 0,5MΩ olarak ölçülmüştür.

Mikrofondan alınan elektriksel ses işaretine eklenebilecek çevresel ve fizyolojik gürültüleri olabildiğince azaltabilmek için bazı önlemler alınması gerekir. İşaretin alınacağı ortam olabildiğince gürültüsüz olmalıdır. Hasta sırt üstü yatar konumda ve hareketsiz olmalıdır. Mikrofon, Bölüm 2'de belirtilen kalp seslerini dinleme odaklarına yerleştirilerek, istenen sesler algılanır.

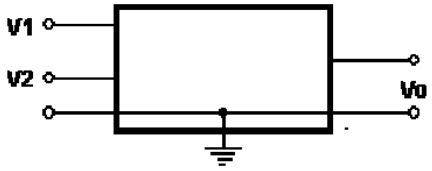
Mikrofonla gerilime dönüştürülen işaret çok zayıf ve gürültülüdür. İşaret, bundan sonra çeşitli filtre ve kuvvetlendirme devreleriyle iyileştirilmelidir.

4.2 Ön Kuvvetlendirici

Kuvvetlendirme, analog işaret işlemenin temelini oluşturur. Genel olarak, küçük genlikli biyolojik işaretlerin kuvvetlendirilmesinde kullanılan kuvvetlendiricilere 'biyopotansiyel kuvvetlendirici' denir. Biyolojik işaretler genellikle fark işaretleri şeklinde olduğundan, bunların kuvvetlendirilmesinde kullanılacak kuvvetlendiriciler de 'fark kuvvetlendiricileri' olur. Biyolojik işaretlerin, özellikle ortamda bulunan ortak mod işaretlerinden arınmış olarak kuvvetlendirilmesi gerekir [1] .

İdeal bir fark kuvvetlendirici, her iki girişinden uygulanan işaretlerin (V1 ve V2) farkını alır ve istenilen oranda yükselterek çıkışına (Vo) aktarır (Şekil 4.3). Dolayısıyla çıkış işaretinin değeri (4.1) ifadesindeki gibi formüle edilir.

$$Vo = Ad(V1 - V2) \tag{4.1}$$



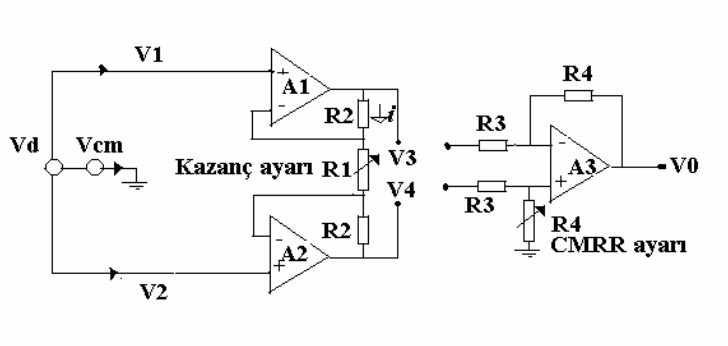
Şekil 4.3 Fark Kuvvetlendirici blok diyagramı

Ancak pratik uygulamalarda yukarıdaki Vo eşitliğini elde etmek mümkün değildir. Çünkü kuvvetlendiricinin yapımından, elemanların toleranslarından kaynaklanan kısıtlamalar vardır. Bu kısıtlamalar fark kuvvetlendiricisini ideal durumdan uzaklaştırır. Fark kuvvetlendiricisini ideal çalışmadan uzaklaştıran en önemli etken ortak işaret kazancıdır. Bu kazanç Ac olarak tanımlanır. İyi düzenlenmiş bir fark kuvvetlendiricisinde ortak işaret kazancı yok edilmeli ya da minimuma indirilmelidir. Bu durum, ortak mod bastırma oranının (CMRR) maksimuma gitmesine karşılık gelir. Aşağıda CMRR' nin matematiksel ifadesi verilmiştir.

$$\text{CMRR} = \text{Fark Kazancı} / \text{Ortak İşaret Kazancı} \tag{4.2}$$

Farksal işaretlerin işlenmesinde, her iki girişinde yüksek empedans sağladığından ve kazancı kuvvetlendirici devrede bulunan bir dirençle ayarlanabildiğinden, "enstrumantasyon kuvvetlendiricisi" yerinde bir seçimdir.

Enstrumantasyon kuvvetlendiricisi, girişine yüksek empedanslı izleyici devreleri konmuş bir fark kuvvetlendiricisidir. Şekil 4.4'te genel bir enstrumantasyon kuvvetlendirici devresi verilmiştir.



Şekil 4.4 Enstrumantasyon kuvvetlendiricisi devresi [1]

Kuvvetlendirici ortak işaret bastırma oranı R4 direnci ile kazancı ise bağımsız olarak R1 direnciyle ayarlanabilmektedir.

Aşağıda, Şekil 4.4'te verilen enstrumantasyon kuvvetlendiricisinin fark kazancı ifadeleri verilmiştir.

$$A_d = \frac{V_0}{V_2 - V_1} = \frac{V_0}{V_4 - V_3} * \frac{V_4 - V_3}{V_2 - V_1} \tag{4.3}$$

$$A_d = \frac{R_4}{R_3} * \frac{R_1 + 2R_2}{R_1} \tag{4.4}$$

Bir enstrumantasyon kuvvetlendiricisi aşağıdaki özellikleri taşımalıdır.

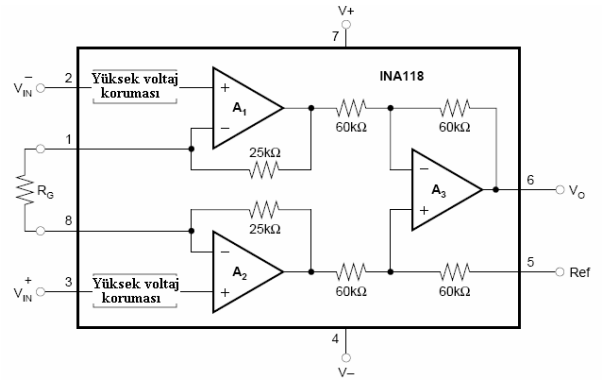
- Yüksek giriş direnci,
- Yüksek CMRR (Common Mode Rejection Ratio- Ortak İşaret Bastırma Oranı),
- Çok kararlı yüksek kazanç,
- Düşük ofset giriş gerilimi ve

- Yüksek lineerlik.

Enstrumantasyon kuvvetlendiricilerinin en önemli özelliği, ortak işaret bastırma oranlarının (100–130 DB) büyük olmasıdır [1].

Bu çalışmada, yukarıda belirtilen nedenlerden dolayı, dönüştürücüden alınan düşük genlikli elektriksel ses işaretinin ön kuvvetlendirilmesinde "Texas Instruments"e ait INA118 enstrumantasyon kuvvetlendiricisi kullanılmıştır.

INA 118 enstrumantasyon kuvvetlendiricisinin kazancı dışardan bağlanan bir direnç (R_G) ile 1 ile 10.000 arasında değiştirilebilmektedir. Düşük güçte çalışır ve (-/+) 1.35 ile (-/+)18 V' a kadar geniş bir besleme gerilimi aralığına sahiptir. Düşük ofset gerilimine (maksimum 50µV) ve yüksek CMRR oranına (kazanç 1000 iken 110 DB) sahiptir. Şekil 4.5'te INA118 enstrumantasyon kuvvetlendiricisinin iç dizaynı verilmektedir.

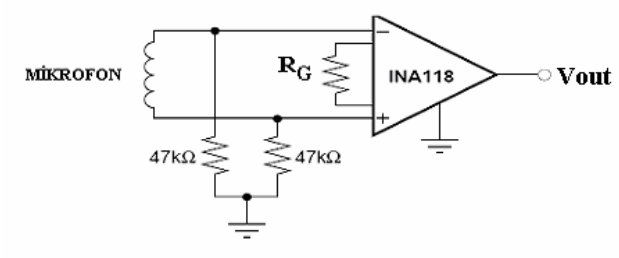


Şekil 4.5 INA118 Enstrumantasyon kuvvetlendiricisi iç dizaynı

Burada giriş, çıkış gerilimleri ve kazanç arasındaki ilişki (4.3) ifadesinde gösterildiği gibidir. G, devrenin fark kazancını göstermektedir.

$$V_0 = G(V_{IN}^{+} - V_{IN}^{-}) \tag{4.5}$$

Şekil 4.6'da, INA118 Enstrumantasyon kuvvetlendiricisinin devrede kullanıldığı haliyle şeması verilmiştir. Burada R_G direnci, kazancı ayarlamak için kullanılmaktadır.



Şekil 4.6: Devrede kullanıldığı haliyle INA118 blok şeması [15]

R_G direnci ve kazanç arasındaki ilişki (4.4) ifadesinde belirtildiği gibidir.

$$G = 1 + \frac{50k\Omega}{R_G} \tag{4.6}$$

Tablo 4.1'de INA118 enstrumantasyon kuvvetlendiricisinin R_G direnciyle dışardan ayarlanabilen kazanç değerleri gösterilmektedir.

Tablo 4.1. INA118'in R_G direnciyle değişen kazanç tablosu [15]

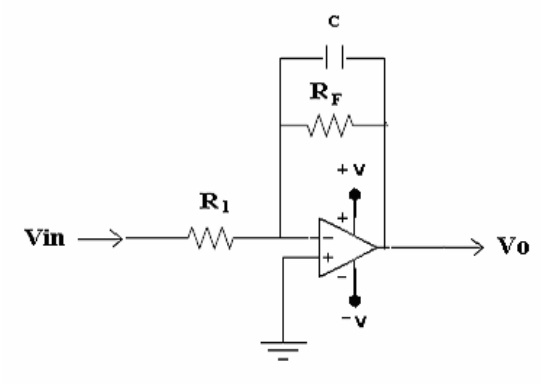
İSTENEN KAZANÇ	R_G (Ω)
1	AD
2	50.00k
5	12.50k
10	5.556k
20	2.632k
50	1.02k
100	505.1
200	251.3
500	100.2
1000	50.05
2000	25.01
5000	10.00
10000	5.001

AD (Açık Devre, bağlantı yok)

4.3 Filtre Tasarımı

Kalp sesleri düşük frekanslı işaretlerdir (yaklaşık 10 – 500 Hz arasında). Bu nedenle sesleri net algılayabilmek için, işaretin yüksek frekanslı bileşenlerden arındırılması gerekir. İşaretin yüksek frekanslı bileşenleri, alçak geçiren filtre kullanılarak yok edilebilir.

Enstrumantasyon kuvvetlendiricisinin çıkışından alınan işaret, bir alçak geçiren filtreden geçirilerek yüksek frekanslı bileşenlerden arındırılır. Şekil 4.7'de birinci dereceden bir alçak geçiren filtre devresi gösterilmektedir.



Şekil 4.7: Alçak geçiren filtre devresi

Bu devre analiz edildiğinde, $\omega = 0$ (doğru gerilim) için kazancı,

$$G = \frac{R_F}{R_1} \tag{4.7}$$

Diğer frekanslar için kazanç,

$$G_\omega = -\frac{Z_2}{Z_1} = -\frac{\frac{R_F}{1+sR_FC}}{R_1} = -\frac{R_F}{R_1} * \frac{1}{1+sR_FC} \tag{4.8}$$

ifadesi ile hesaplanabilir.

İncelenen ses işaretinin en yüksek frekans bileşeni 500 Hz'ler civarında olduğu için kullanılacak filtrenin kesim frekansının 500 Hz seçilmesi uygun bulunmuştur. Devrede kullanılacak kapasite uygun değer olarak 22 nF seçilmiştir. Bu veriler göz önünde bulundurulduğunda aşağıdaki eşitliklerden direnç değeri hesaplanabilir.

$$fo = \frac{1}{2\pi R_F C} \tag{4.9}$$

$$R_F = \frac{1}{2\pi f_0 C} \tag{4.10}$$

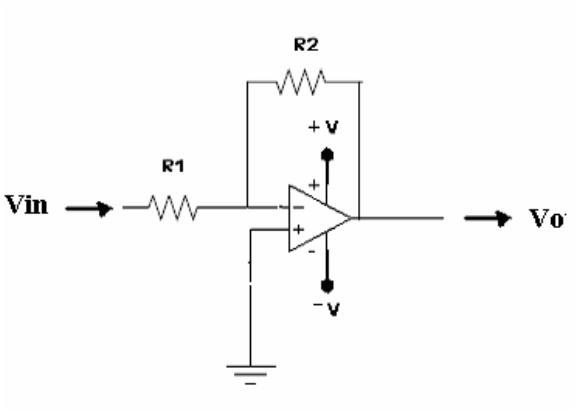
➔Buradan R_F değeri yaklaşık 12 KΩ olarak hesaplanır.

$R_1 = R_F$ seçilerek doğru gerilim kazancı G = 1 tutulmuştur.

4.4 Kuvvetlendirici Tasarımı

Filtre çıkışında elde edilen işaretin genliği küçük olduğu için tekrar kuvvetlendirilmesi gerekmiştir.

Şekil 4.8 de gösterilen faz çeviren kuvvetlendirici devresinde kazanç, R1 ve R2 dirençlerine bağlıdır. Devre analiz edildiğinde kazanç (G),



Şekil 4.8: Kuvvetlendirici devresi

$$G = -R2/R1 \tag{4.8}$$

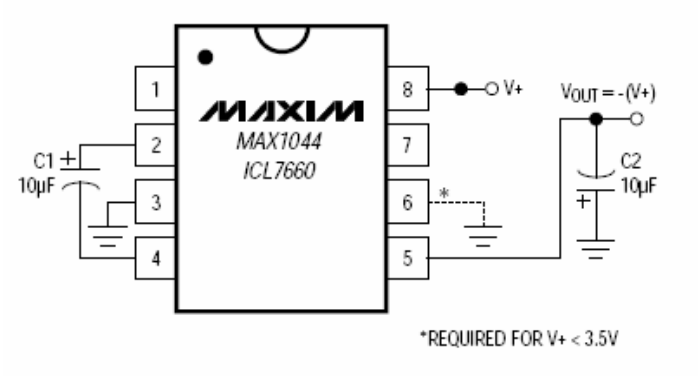
olarak elde edilir. R2 = 56KΩ ve R1= 5.6KΩ seçilmiş, dolayısıyla kuvvetlendirici kazancı G=10 olarak elde edilmiştir. Böylece, filtre çıkışında elde edilen işaret istenen gerilim seviyesine çekilmiştir.

Filtre ve kuvvetlendirici devrelerini gerçeklemek için Analog Devices'ın OP490 işlemsel kuvvetlendiricisi kullanılmıştır. OP490, ± 0.8 V , ± 18V besleme gerilimi aralığına sahiptir.

Devrede kullanılan elemanların negatif gerilim ihtiyacını karşılamak için negatif gerilim çeviriciye ihtiyaç duyulmuştur.

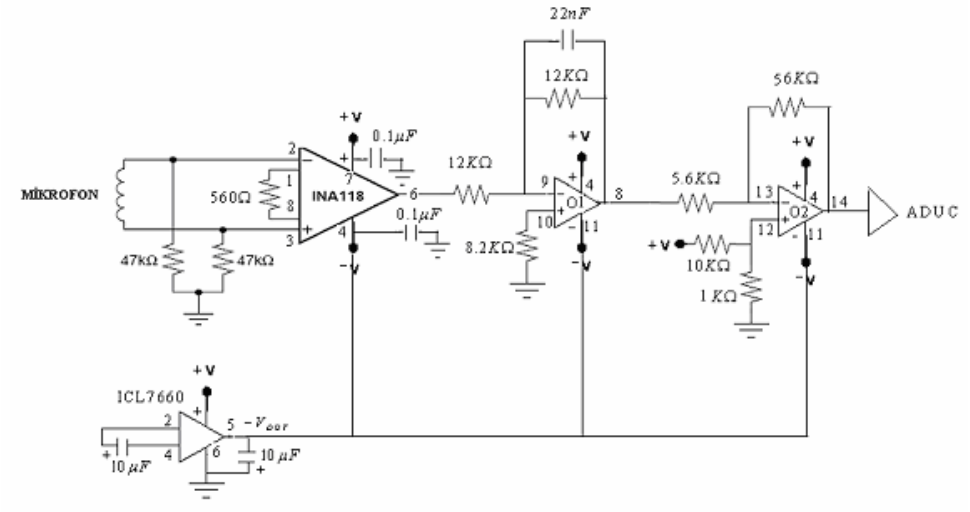
Negatif gerilim çevirici olarak Maxim'in ICL7660 gerilim çevirici elemanı kullanılmıştır. 1.5 V - 10 V besleme gerilimi aralığına sahiptir. Bu eleman ile girişe uygulanan besleme gerilimi negatife çevrilebilir, gerilim bölünebilir, katı alınabilir ya da aynı değerde ikinci bir kaynak gerilimi elde etmek için kullanılabilir.

Şekil 4.9 de negatif gerilim çevirici devresi verilmiştir.



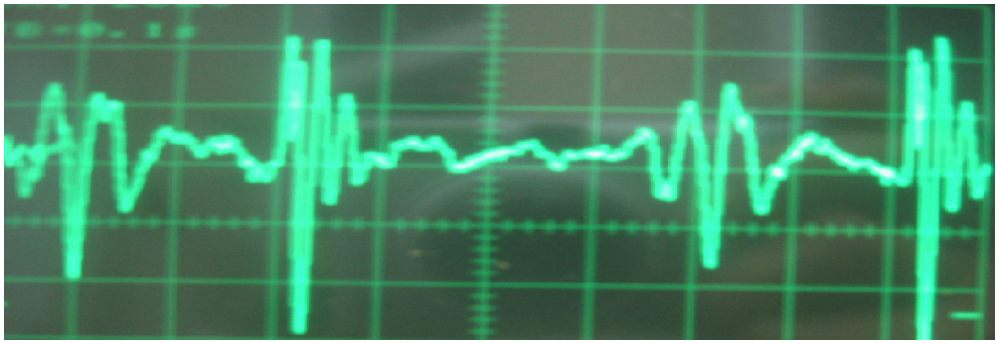
Şekil 4.9 Negatif gerilim çevirici [16]

Buraya kadar yapılan işlemlerle, analog işaret elde edilmiş durumdadır. Analog işaretin elde edildiği sistem bir bütün olarak Şekil 4.10 da gösterilmiştir. O1 ve O2 OP490 işlemsel kuvvetlendiricisidir.



Şekil 4.10: İşaretin analog olarak elde edilmesi

Şekil 4.10 da görülen devrenin ürettiği analog çıkış geriliminin zamana göre değişimi Şekil 4.11.a ve 4.11.b 'de görülmektedir.



Şekil 4.11.a: Analog işaretin osiloskopta görüntülenmesi,(t=0.1 s/kare ve V = 0.5V/kare için)

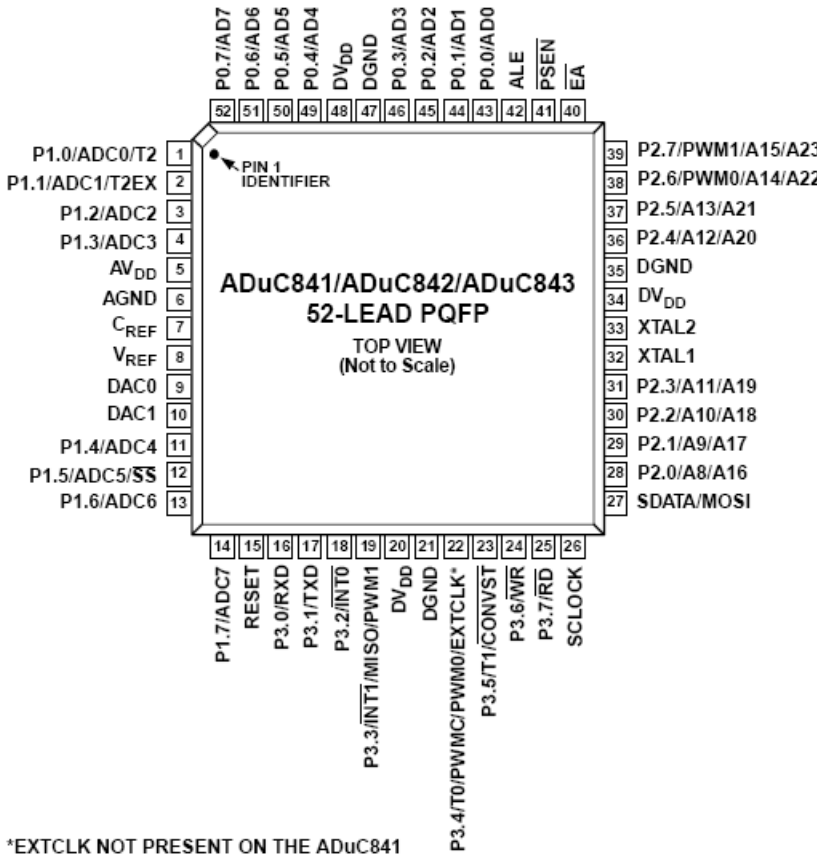


Şekil 4.11.b: Analog işaretin osiloskopta görüntülenmesi,(t=0.2s/kare ve V = 0.5V/kare için)

4.5 İşaretin Sayısal Ortama Aktarılması

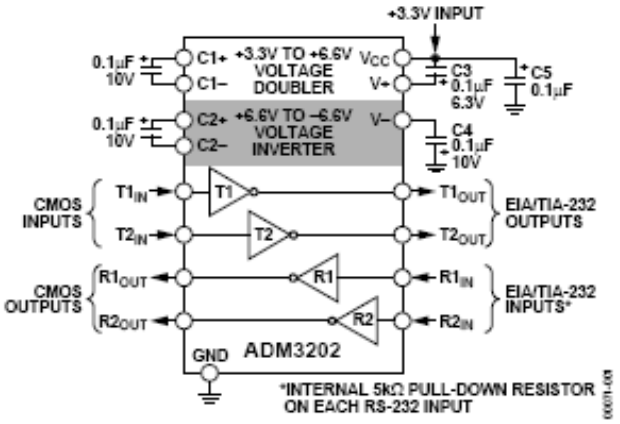
Kuvvetlendirici çıkışında elde edilen işaret 12 bitlik ADC ile sayısal işarete dönüştürülür.

ADC olarak "Analog Devices"ın ADUC841 mikro denetleyici elemanı seçilmiştir (Şekil 4.12). 11.0592 MHz'lik kristal kullanılmıştır. Kalp sesi işaretinin en yüksek frekans bileşeni f_{max} = 500 Hz'ler civarında olduğu için, sayısal ortama dönüştürülürken Nyquist kriteri uyarınca örnekleme frekansı $f_0 = 2f_{max}$ = 1000 Hz olarak seçilmiştir.



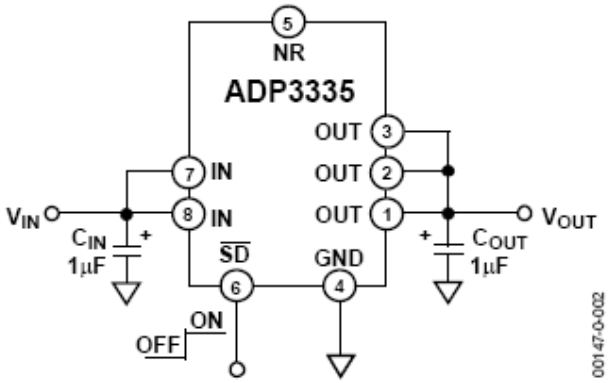
Şekil 4.12: ADUC841 pin konfigürasyonu [17]

ADC çıkışında alınan işaret seri port ile bilgisayara aktarılır. Seri haberleşmeyi sağlamak için "Analog Devices"ın ADM3202 elemanı kullanılmıştır (Şekil 4.13).



Şekil 4.13: ADM 3202 seri port elemanı fonksiyonel blok diyagramı [18]

Devrede kullanılan tüm elemanlar 3.3 V gerilim ile beslenecek şekilde seçilmiştir. Dışardan bağlanan 9 V'luk pil ile çalışan devre için, bu gerilimi 3.3 V 'a çeviren bir regülatör kullanılması gerekmiştir. Bu dönüşüm için, "Analog Devices"ın ADP3335 elemanı kullanılmıştır. ADP3335, 2.6 -12 V beslene gerilimi aralığına sahiptir. Şekil 4.14'te ADP3335'in tipik uygulama devresi verilmiştir.



Şekil 4.14: ADP3335 tipik uygulama devresi [19]

Şekil 4.15'te, kalp sesini mikrofon ile vücuttan alıp, seri port ile bilgisayara aktaran sistem bir bütün olarak gösterilmiştir.



Şekil 4.15 Kalp sesini mikrofon ile vücuttan alıp seri port ile bilgisayara aktaran sistem

Bu aşamadan sonra, işaret sayısal olarak bilgisayarda analiz edilebilir.

5. GERÇEK ZAMANDA KALP SESLERİNİN ANALİZİ

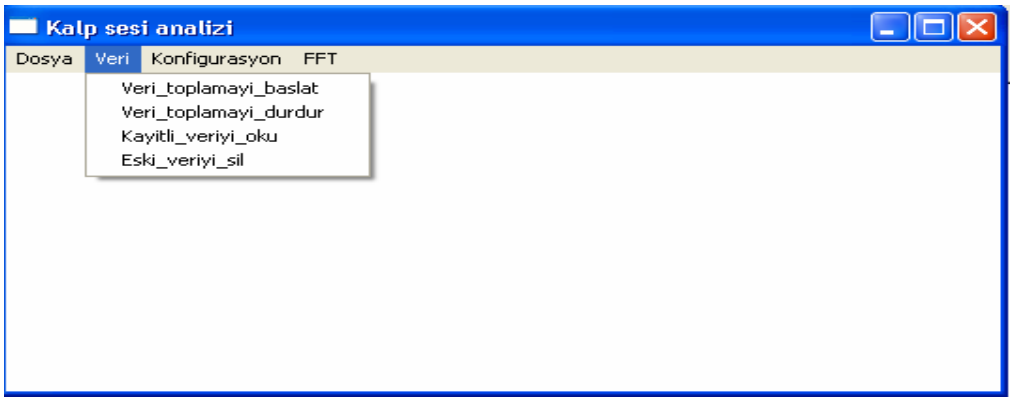
Bilgisayara aktarılan işaret, MATLAB, C gibi çeşitli uygulama ortamlarında incelenebilir. Bu çalışmada işaret, C dilinde yazılan bir programla ekranda bastırılmış ve Kısa-Zaman Fourier analizi yapılmıştır.

5.1 İşaretin Bilgisayara Aktarılması

İşaretin alınması, çalışır durumdaki devrenin seri bağlantı ile bilgisayara bağlı olduğu durumda, programdan verilen istekle portların açılması ile başlar. Porttan gelen her bir veri 8 bit değerindedir. Bu nedenle 2 byte'lık veri alınır ve bunun işaret bilgisini taşıyan 12 biti kullanılır (ADC 12 bitlik olduğu için işaret verisini 12 bit taşımaktadır).

Alınan veriler iki farklı geçici bellekte saklanarak ekrana iletilmektedir. Belleklerden birine veri girişi yapılırken diğerine yüklenmiş olan veri ekrana bastırılmaktadır.

Devreden veri akışını kontrol eden ve alınan verilerin ekrana bastırılması ve analiz işlemlerinin yapılabildiği menü Şekil 5.1'de gösterilmiştir.



Şekil 5.1 Kalp sesi analiz menüsü

5.2 Program ve Menü Özellikleri

Seri porttan gerçek zamanda veri alınmak istendiğinde, öncelikle "**Konfigürasyon**" seçeneğinden ilkleştirme yapılır, yani hangi portun kullanılacağı girilir.

Portun doğru olarak belirtilmesinin ardından,

"**Veri**" ikonundan,

"**Veri toplamayı başlat**" seçeneği porttan veri alma işlemini başlatır ve gerçek zamanda alınan işaret ekrana bastırılır.

"**Veri toplamayı bitir**" seçeneği ile port kapatılır, veri toplama işlemi biter.

Porttan veri alma işlemi sırasında, alınan veriler aynı zamanda, daha sonra incelenmek üzere bir dosyaya aktarılmaktadır.

"**Kayıtlı Veriyi Oku**" seçeneği ile kaydedilmiş bu veri ekrana bastırılabilir. Kayıttan veri okunurken, ekrana bastırmak için klavyede space tuşuna basılarak verinin izlenmesi sağlanır.

"**Eski veriyi sil** " seçeneği ile daha önce kaydedilmiş dosyaların işlem fazlalığı yaratmaması istendiğinde silinebilmesini sağlar.

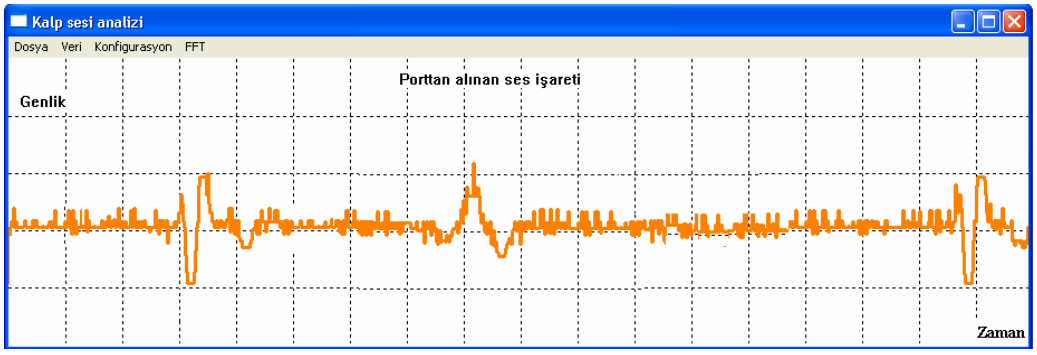
"**FFT**" ikonundan

"**FFT başlat** " ile FFT analizi yapılacaktır. Dosyadaki veri okunduğunda, ya da porttan veri alındığında, ekranda aynı zamanda işaretin FFT'si görülecektir.

"**FFT bitir** " seçeneği ile FFT işlemi sonlandırılır.

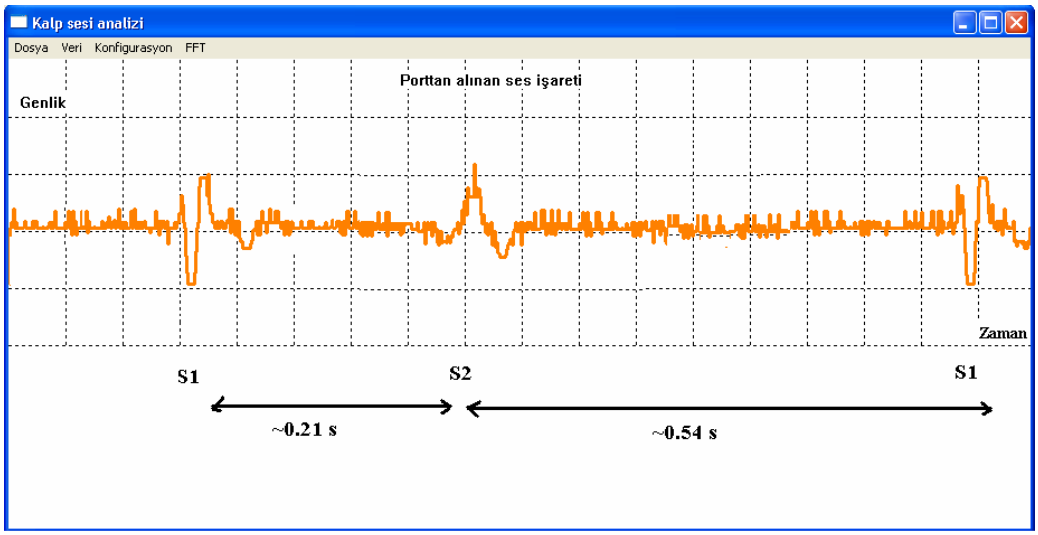
5.3 Analiz Sonuçlarının Ekranda Gösterimi

Bilgisayara aktarılan kalp sesleri, ekranda açılan "Kalp sesi analizi" penceresinde görüntülenmektedir (Şekil 5.2).



Şekil 5.2 Gerçek zamanda alınan kalp sesi işareti

İşaretin kesim frekansı 500 Hz civarında olduğu için, örnekleme frekansı f_0=1000 Hz olarak seçilmiştir. Ekranda bastırma işlemi için de bu değer göz önünde tutulmuştur. Penceredeki işaret 0–1 s aralığındadır. Karelerin her birinin süresi 0.05 s'dir. Aşağıda örnek bir ses işaretinin S1 ve S2 sesleri arsındaki süreler gösterilmiştir (Şekil 5.3).



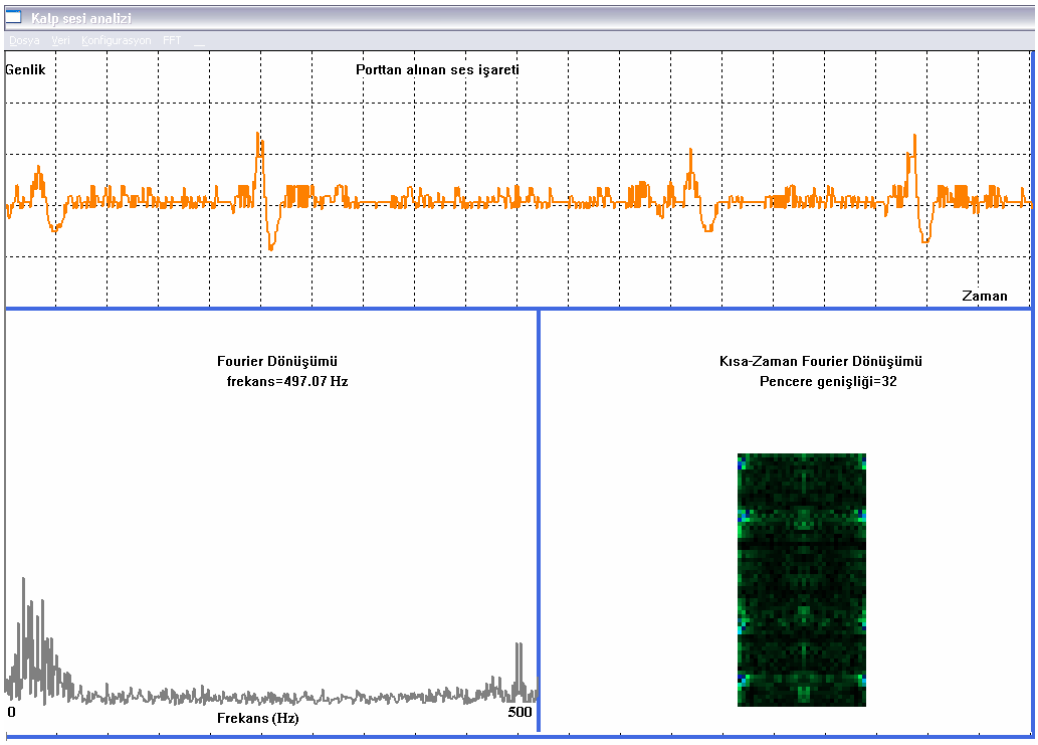
Şekil 5.3 S1-S2 sesleri arasındaki sürenin gösterilmesi

Buradan kareleri sayarak iki S1 vurusu arasındaki sürenin yaklaşık 0.75 s olduğunu söyleyebiliriz.

Bu sonuç, Bölüm 2'de belirtilen sistol-diastol süreleriyle uyumludur. Dolayısıyla işaretin herhangi bir gecikme olmadan gerçek zamanda aktarımı başarılmış olmaktadır.

İşaretin hızlı Fourier analizi için seçilen pencere genişliği örnekleme frekansıyla f_0=1000 olduğu için N= 1024 alınmıştır.

Porttan ya da dosyadan alınan her N veri için hızlı Fourier dönüşümü uygulanmakta ve ekrana bastırılmaktadır.



Şekil 5.4 Ses işaretinin gerçek zamanda Fourier ve Kısa- Zaman Fourier dönüşümünün alınması

Frekans spektrumu incelendiğinde, işaretin 0, 500 Hz civarında frekans bileşenleri olduğu görülmektedir. Program çalışırken işaretçi frekans ekseni üzerinde dolaştırıldığında istenen noktadaki frekans görülebilmektedir. İşaretin iki boyutlu kısa-zaman Fourier analizi de Şekil 5.4'te gösterilmiştir. Dönüşüm yapılırken seçilen pencere boyutu da ekrana yazdırılmaktadır. Şekil 5.4'te, pencere boyutu 32 (N=1024 olmak üzere) seçildiğinde elde edilen sonuç görülmektedir.

Kısa-zaman Fourier dönüşümünde işaretin incelendiği pencere boyutu önem taşımaktadır. Pencere boyutu küçüldükçe işaretin durağanlık özelliği artarken, frekans çözünürlüğü azalır, pencere boyutu büyüdüğünde ise, işaretin frekans çözünürlüğü artarken zamanda çözünürlük azalır. Pencere boyutu, bu durum göz önünde bulundurularak seçilmelidir.

Ses işaretinin seri port üzerinden alınarak bilgisayarda analiz edilmesi için C programlama dilinde yazılan program EK.1 de verilmiştir.

6. SONUÇLAR

Kalbin dinlenmesi (Oskültasyon) kalbin fonksiyonel doğruluğunun en hassas testlerinden biridir. Çok kez kalp sesleri içerisindeki değişiklikler veya üfürümler, organik kalp hastalığının belli işaretleridir. Bunlar kardiyovasküler sistem içerisindeki önemli basınç değişikliklerine bağlı bulgu ve semptomlar ortaya çıkmadan saptanabilirler.

Bu çalışma ile kalp seslerinin insan vücudundan algılanıp analiz edilebilir forma getirilmesi ve doktorun tanı koymasına yardımcı olacak, anormallikleri tespit edecek, sınıflandırabilecek bir sisteme hazırlık amaçlanmıştır. Alınmak istenen biyolojik işaretin genliği oldukça küçüktür ve çevresel gürültüler nedeniyle tespit edilmesi zordur. Ölçülen büyüklüklerin gerçek büyüklükleri gösterebilmesi için, canlı organizmanın iç yapısı ve özellikleri, ölçüm sisteminin tasarımı ve uygulanması sırasında dikkate alınmalıdır. Düşük frekanslı kalp seslerini algılamak için kullanılan mikrofonun hassas olması, sesin alınacağı ortamın gürültüden arındırılmış olmasını gerekli kılmaktadır.

Tasarlanan devre 9 V'luk pil ile çalıştırılmaktadır. Dışardan alınan 9 V besleme gerilimi, devre içinde 3.3'V a dönüştürülmüş ve diğer elemanlar 3.3 V ile beslenmiştir. Uygulanan besleme geriliminin küçük olması, devrenin gürültüye olan hassasiyetini artırmıştır. Bu durum, devrede kullanılacak elemanların seçiminde göz önünde bulundurulmuştur. Şebekeden beslenmediği için ayrıca yalıtım amaçlı bir devre yapılmasına gerek duyulmamıştır.

Sayısal ortama aktarılan ses işaretinin gerçek zamanda analizi C programlama dili ile yapılmıştır. Seri porttan alınan işaret, ekranda bastırılmış ve istendiğinde Fourier ve kısa-zaman Fourier analizi yapılmasını sağlayacak bir menü hazırlanmıştır.

Fourier dönüşümünde frekansta çözünürlük yakalanırken uzamsal bilgi kaybedilmektedir. Kısa-zaman Fourier dönüşümünde ise işaretin incelendiği pencere boyutu önem taşımaktadır. Yapılacak yeni bir çalışmada analiz için, wavelet dönüşümü denenebilir.

Ekranda bastırılan işaretin zaman aralığı incelendiğinde elde edilen S1 S2 süreleri, bilinen sürelerle uyumludur. Dolayısıyla işaretin gerçek zamanlı gösterimi başarılmıştır.

Bu aşamadan sonra, yapılan çalışmanın klinik uygulamalarda kullanılabilir hale gelmesi için, sağlıklı ve problemli kalp seslerinden bir veri tabanı oluşturulması gerekmektedir. Sağlıklı kalp seslerini problemli olanlarından ayırabilecek bir sistem yazılım ile gerçeklenebilir ve sınıflandırma yapılabilir.

KAYNAKLAR

[1] **Yazgan E. ve Korürek M., 1996**. Tıp Elektroniği, İstanbul.

[2] **Philips, R.E. ,** The Heart and the Circulatory System
http://www.accessexcellence.org/AE/AEC/CC/heart_background.html

[3] www.acilveilkyardim.com/acilbakim/akcigerkalpsesdin.htm

[4] www.kfshrc.edu.sa/perfusion/education/, Cardiak Cycle,

[5] www.servier.com , Medical Images,

[6] **G.Amit,** May 2004, Heart Sound Analysis: Theory, Techniques and
Applications, Advanced Research Seminar

[7] http://library.thinkquest.org/C003758/Function/CardiacCycleM.htm, Cardiac
Cycle

[8] **Abaoğlu,C., Aleksanyan, V.**, 1980. Semptomdan Teşhise, İstanbul

[9] http://library.med.utah.edu/kw/pharm/hyper_heart1.html

[10] **Kula, S.** Kardiyovasküler Sistem Muayenesi

[11] **Dalling, Z**. Cardiovascular Examination

[12] **Najarian, K. , Splinter, R.** ,2006, Biomedical Signal and Image Processing

[13] **Chassaing, R**. ,1992, Digital Signal Processing with C and theTMS320C30

[14] http://pharyngula.org/~pzmyers/MyersLab/teaching/Bi104/l06/img/heartanat.jpg

[15] Analog Devices,INA118, Datasheet

[16] Maxim, Switched-Capacitor Voltage Converters, Datasheet,

[17] Analog Devices ADUC841, Datasheet

[18] Analog Devices ADM3202, Datasheet

[19] Analog Devices ADP3335, Datasheet

[20] **Bonagura**, **J.D.,** DVM, MS, Diplomate ACVIM (Cardiology, Internal Medicine) The Ohio State University

[21] **Rangayyan R.M**., 2002,.Biomedical Signal Analysis

[22] **Semmlow, J.** , 2005,Circuits,Signals and Systems for Bioengineers : A Matlab Based-Introduction, Elseiver Academic Press

EKLER

EK.1

```
// KALP SESLERININ SERI PORTTAN ALINMASI VE FFT ANALIZI ////ARALIK 2006//
#include <windowsx.h>
#include "windows.h"
#include <stdlib.h>
#include <stdio.h>
#include "dibit.h"
#define          DEST_PORT                    80
#define          ASIZE 1024
#define          FSIZE 1024
#define          WSIZEX 1024
#define          WSIZEY 256
#define          XBOL 20
#define          YBOL 5
#define N 1024   // ornek sayisi
double fo = N;
unsigned char            alan0[ASIZE],alan1[ASIZE];
struct cmpx              ALANF[N];
int                      ALANPWR[ASIZE],alanF[ASIZE];
int                      sayx=0,sayz=4999,diger=0,bekle = 0,l1=0,l2=0,p,o,xx,yy;
int                      ll,jj=1,ala1=0,ala0=1,ciz=0,goster=0,EKGs=0; //EKGss=0,NN=4;
int                      kalinlik, renk;
float                    pi=3.14159265;
char                     str[255], KayitliDosyaIsmi[30] ;
int                      devam=0,sayy=0,okuma=0,gol=1,goll=0,bekle1=0,bt1=0,OKAY=0;
char                     PortNum[4]="COM10";
FILE                     *dosya1,*dosya2,*dosya3,*dosyaF;
DWORD          dwThreadId, dwThrdParam = 0;
HANDLE                   hThread1=0,hThread2=0,hThread=0,hReadThread=0;
```

```
COLORREF                RGB;
HDC                     hdcScreen,hdcScreen1;
WORD                    wOperation = IDM_SETDIB;
HANDLE                  hDIBInfo = NULL;
HWND                    hWnd;
DWORD WINAPI PortReadThread();
HANDLE                  hInst;
BOOL                    FFTbit;
SOCKET          Sock = INVALID_SOCKET;
SOCKADDR_IN             source_sin, dest_sin;
WSADATA                 WSAData;
HDC                     hdc1;
HANDLE                  hPort=0;
void  PASCAL    Proc1();
void  PASCAL    Proc2();
void  PASCAL    Proc3();
void  PASCAL    Dosya_oku();
int PASCAL WinMain(hInstance, hPrevInstance, lpCmdLine, nCmdShow)
HANDLE hInstance;
HANDLE hPrevInstance;
LPSTR lpCmdLine;
int nCmdShow; {
        MSG msg;
          if (!hPrevInstance)
        if (!InitApplication(hInstance))
        return (FALSE);
         if (!InitInstance(hInstance, nCmdShow))
         return (FALSE);
         while (GetMessage(&msg,  NULL,NULL,NULL))
                {
                TranslateMessage(&msg);
                DispatchMessage(&msg);
                }
return (msg.wParam);
}
```

```
BOOL InitApplication(hInstance)
HANDLE hInstance;{
   WNDCLASS  wc;
   wc.style = NULL;
   wc.lpfnWndProc = MainWndProc;
   wc.cbClsExtra = 0;
   wc.cbWndExtra = 0;
   wc.hInstance = hInstance;
   wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
   wc.hCursor = LoadCursor(NULL, IDC_ARROW);
   wc.hbrBackground = COLOR_WINDOW+1;
   wc.lpszMenuName =  "dibitMenu";
   wc.lpszClassName = "dibitWClass";
   return (RegisterClass(&wc));
}
BOOL FAR PASCAL __export Sorgulama(hDlg, message, wParam, lParam)
HWND hDlg;
unsigned message;
WORD wParam;
LONG lParam; {
   switch (message) {
                        case WM_INITDIALOG:
                                return (TRUE);
                        case WM_COMMAND:
                                devam=0;
                                if (wParam == IDOK){
                                        devam=1;
                                        EndDialog(hDlg, LOWORD(wParam));
                                        return TRUE;
                                        }
                                if (wParam == IDCANCEL) {
                                        devam=0;
                                        EndDialog(hDlg, LOWORD(wParam));
                                        return TRUE;
```

```
                                        }
                        break;
                        }
   return (FALSE);
}
BOOL InitInstance(hInstance, nCmdShow)
 HANDLE hInstance;
 int            nCmdShow;{
   hInst = hInstance;
   hWnd = CreateWindow("dibitWClass",
                                    "Kalp sesi analizi",
                                    WS_OVERLAPPEDWINDOW,
                                    CW_USEDEFAULT,
                                    CW_USEDEFAULT,
                                    CW_USEDEFAULT,
                                    CW_USEDEFAULT,
                                    NULL,
                                    NULL,
                                    hInstance,
                                    NULL );
if (!hWnd)
      return (FALSE);
 ShowWindow(hWnd, nCmdShow);
UpdateWindow(hWnd);
hdcScreen=GetDC(hWnd);
return (TRUE);
}
int                     hh=0,pp=0;
long FAR PASCAL __export MainWndProc(hWnd, message, wParam, lParam)
HWND hWnd;
UINT message;
WPARAM wParam;
LPARAM lParam; {
   FARPROC lpProcAbout;
```

```
        char szBuffer2[100];
        TCHAR  tusabas ;
        int xPos,yPos;
switch (message) {
        case WM_CREATE:
           hDIBInfo = GlobalAlloc(GMEM_MOVEABLE,
                (DWORD)(sizeof(BITMAPINFOHEADER) + 256 * sizeof(RGBQUAD)));
break;
        case WM_MOUSEMOVE:
                hdc1=GetDC(hWnd);
                xPos=LOWORD(lParam);
                yPos=HIWORD(lParam);
                if(FFTbit==1 && yPos>300){
                sprintf(szBuffer2,"frekans=%0.2f",1000.0*(1.0*xPos/FSIZE));
                TextOut(hdc1,0,600,szBuffer2,strlen(szBuffer2)); }
        break;
        case WM_INITMENU:
           CheckMenuItem(wParam, wOperation, MF_CHECKED);
break;
        case WM_COMMAND:
                switch(wParam) {
                        case IDM_ABOUT:
                                lpProcAbout = MakeProcInstance(About, hInst);
                                DialogBox(hInst,
                                "Konfigurasyon",
                                hWnd,
                                lpProcAbout);
                                FreeProcInstance(lpProcAbout);
                        break;
                        case IDM_EXIT:
                                FFTbit = 0;
                                DestroyWindow(hWnd);
                        break;
                        case IDM_ECGB:
```

```
                    if (bt1==0){
                    goster=0;
                    hh=0;
                    pp=1;
                    PortInitialize();
                    bt1=1;     }
                break;
                case IDM_ECGS:
                    if (bt1==1){
                        pp=0;
                        hh=1;
sus:
                        if (gol==0)                    goto sus;
                        gol=0;
                        if (hPort!=NULL)   CloseHandle(hPort);
                        bt1=0;
                    }
                break;
                case IDM_EKGOku:
                    EKGs=0;
                    pp=0;
                    hh=1;
                    goster=1;
                    diger=0;
sus1:
                    if (gol==0)                    goto sus1;
                    gol=1;//gol=0
                    if (hPort!=NULL)   CloseHandle(hPort);
                    if (dosya1==NULL)              dosya1=fopen("EKG_verisi.dat","rb");
                    okuma=1;
                    bt1=0;
                    break;
                case IDM_SEKG_Oku:
                break;
```

```
                case IDM_Eski_Sil:
                                if(remove("EKG_Verisi.dat")!= 0)
                                MessageBox (hWnd, TEXT("Dosya silinemedi"),
                                TEXT(""), MB_OK);
                        else
                                MessageBox (hWnd, TEXT("Dosya silindi"),
                                TEXT(""), MB_OK);
                        break;
                        case IDM_FFT:
                                FFTbit = 1;
                        break;
                        case IDM_FFT_SON :
                                FFTbit = 0;
                        break;
                        default:
                return (DefWindowProc(hWnd, message, wParam, lParam));
   }
   break;
case WM_USER:
        goster=0;
        Proc3();
        Beep(1000,1000);
        bekle1=0;
break;
case WM_USER+1:
        if      (goster==0)
        Proc2();
break;
case WM_CHAR:
        if(okuma==1){
                tusabas =(TCHAR) wParam;
                if(tusabas ==0x20)      {Dosya_oku(); }
                }
break;
```

```
            case WM_DESTROY:
bekleme:
                        if (bekle==1)          goto bekleme;
                        CloseHandle(hReadThread);
                        CloseHandle (hPort);
                        ReleaseDC(0,hdcScreen );
                        PostQuitMessage(0);
            break;
            default:
                        return (DefWindowProc(hWnd, message, wParam, lParam));
   }
   return (NULL);
}
void PASCAL NEAR Proc3(){
            DWORD dwCommModemStatus;
            DWORD dwModemStatus;
            BOOL  fCTS, fDSR, fRING, fRLSD;
            int           x;
            COMMTIMEOUTS CommTimeouts;
            COMMTIMEOUTS timeouts;
            COMSTAT comStat;
            DWORD   dwErrors;
            BOOL    fOOP, fOVERRUN, fPTO, fRXOVER, fRXPARITY, fTXFULL;
            BOOL    fBREAK, fDNS, fFRAME, fIOE, fMODE;
            DWORD dwError, dwThreadID;
            TCHAR   szError[100];
            DCB                   PortDCB;
            if ( hPort != INVALID_HANDLE_VALUE ) CloseHandle (hPort);
deneme1x:
            hPort = CreateFile (PortNum,
                  GENERIC_WRITE | GENERIC_READ,
                  FILE_SHARE_READ | FILE_SHARE_WRITE,
                   NULL,
                   OPEN_EXISTING,
```

```
                FILE_FLAG_RANDOM_ACCESS,
                NULL);
            if ( hPort == INVALID_HANDLE_VALUE ) goto deneme1x;
            PortDCB.DCBlength = sizeof (DCB);
            GetCommState (hPort, &PortDCB);
             PortDCB.BaudRate = 57600;//38400;
             PortDCB.fBinary = TRUE;
              PortDCB.fParity = FALSE;
              PortDCB.fOutxCtsFlow = FALSE;
              PortDCB.fOutxDsrFlow = FALSE;
              PortDCB.fDtrControl = DTR_CONTROL_ENABLE;
              PortDCB.fDsrSensitivity = FALSE;
              PortDCB.fTXContinueOnXoff = TRUE;
              PortDCB.fOutX = FALSE;
              PortDCB.fInX = FALSE;
              PortDCB.fErrorChar = FALSE;
              PortDCB.fNull = FALSE;
              PortDCB.fRtsControl = RTS_CONTROL_ENABLE;
              PortDCB.fAbortOnError = FALSE;
              PortDCB.ByteSize = 8;
              PortDCB.Parity = NOPARITY;
              PortDCB.StopBits = ONESTOPBIT;
 if (!SetCommState (hPort, &PortDCB))  {
   MessageBox (hWnd, TEXT("Unable to configure the serial port"),
   TEXT("Error"), MB_OK);
   dwError = GetLastError ();
   return FALSE;
}
GetCommTimeouts (hPort, &CommTimeouts);
CommTimeouts.ReadIntervalTimeout = 1000;
CommTimeouts.ReadTotalTimeoutMultiplier = 1000;
CommTimeouts.ReadTotalTimeoutConstant = 1000;
CommTimeouts.WriteTotalTimeoutMultiplier = 1000;
CommTimeouts.WriteTotalTimeoutConstant = 1000;
```

```
 if (!SetCommTimeouts (hPort, &CommTimeouts))  {
            MessageBox (hWnd, TEXT("Unable to set the time-out parameters"),
           TEXT("Error"), MB_OK);
            dwError = GetLastError ();
            return FALSE;
 }
EscapeCommFunction (hPort, SETDTR);
EscapeCommFunction (hPort, SETRTS);
return 0;
}
///////// DOSYADAN KAYITLI VERİYİ OKUMA ////////////////////////
void  PASCAL Dosya_oku(){
int                     x,i,numread;
char                    szBuffer2[100];
unsigned char           veri1,veri2,Byte,ch;
ala1=0;
            for(i=1;i<=FSIZE;i++) {
                        numread = fread( &veri1, sizeof( char ), 1, dosya1 );
                        if(numread==0) goto sonn;
                        numread = fread( &veri2, sizeof( char ), 1, dosya1 );
                        if(numread==0) goto sonn;
                        x = 0;
                        x = veri2;
                        x = x << 8;
                        x = x & (0x0F00);
                        x = x | veri1;
                        x = x /16;
                        veri1 = x;
                        alan0[i]  = veri1;
                        }
                        Proc2();
            goto atla2;
sonn:
            okuma=0;
```

```
		goster=0;
		fclose(dosya1);
		dosya1=NULL;
atla2:
		hh=0;
}
////////////////////// EKRANA BASTIRMA/////////////////////////////////////////////////
void  PASCAL Proc2(){
		int			xq,yq,yqF;
		char			szBuffer2[100];
		xq=0;
		hdc1=GetDC(hWnd);
		kalinlik=1;
		renk=RGB(0,0,0);
		BitBlt(hdc1,0, 0, 1024, 1024, NULL, 0,0, WHITENESS);
		SelectObject(hdc1,CreatePen(PS_DOT,kalinlik,renk));
		for(xq=WSIZEX/XBOL; xq < WSIZEX; xq += WSIZEX/XBOL){
				MoveToEx(hdc1,xq,0,NULL);
				LineTo(hdc1,xq,WSIZEY);
			}
		for(yq=WSIZEY/YBOL; yq < WSIZEY; yq += WSIZEY/YBOL){
				MoveToEx(hdc1,0,yq,NULL);
				LineTo(hdc1,WSIZEX, yq);
			}
			sprintf(szBuffer2,"Porttan alınan ses işareti ");
			TextOut(hdcScreen, 350,10,szBuffer2,strlen(szBuffer2) );
			sprintf(szBuffer2,"Genlik ");
			TextOut(hdcScreen, 0,10,szBuffer2,strlen(szBuffer2) );
			sprintf(szBuffer2,"Zaman");
			TextOut(hdcScreen, FSIZE-50,235,szBuffer2,strlen(szBuffer2) );
 kalinlik=2;
renk=RGB(255,128,0);
SelectObject(hdc1,CreatePen(PS_SOLID,kalinlik,renk));
if (ala1==1){
```

```
if(FFTbit == 1){
                        for (xq=0;xq<FSIZE;xq=xq+1) {
                                ALANF[xq].real=alan0[xq];
                                ALANF[xq].imag=0;
                                }
                                FFT(ALANF,N,0);
                                for (xq=0;xq<FSIZE;xq=xq+1){
                                ALANPWR[xq]=    (int)sqrt(    (ALANF[xq].real*ALANF[xq].real)    +
(ALANF[xq].imag *ALANF[xq].imag ) );
                                }
                        }
kalinlik=2;
renk=RGB(255,128,0);
SelectObject(hdc1,CreatePen(PS_SOLID,kalinlik,renk));
MoveToEx(hdc1,0,200-alan0[0],NULL);
yq= 200-alan1[0];
yqF = 700-50*ALANPWR[0]/FSIZE;
for (xq=0;xq<FSIZE;xq=xq++)  {
                        yq = 200-alan1[xq];
                        LineTo(hdc1,xq,yq);
                        MoveToEx(hdc1,xq,yq,NULL);
            }
if (FFTbit== 1){
                        kalinlik=2;
                        renk=RGB(128,128,128);
                        SelectObject(hdc1,CreatePen(PS_SOLID,kalinlik,renk));
                        MoveToEx(hdc1,xq,700-50*ALANPWR[0]/FSIZE,NULL);
                        for(xq=0;xq<FSIZE;xq++){
                                sprintf(szBuffer2,"Frekans");
                                TextOut(hdcScreen, FSIZE-50,500,szBuffer2,strlen(szBuffer2) );
                                yqF= 700-50*ALANPWR[xq]/FSIZE;
                                LineTo(hdc1,xq,yqF);
                                MoveToEx(hdc1,xq,yqF,NULL);          }
            }
}
```

```
if (ala1==0){
        if(FFTbit == 1)     {
            for (xq=0;xq<FSIZE;xq=xq+1) {
                ALANF[xq].real=alan0[xq];
                ALANF[xq].imag=0;
                }
                FFT(ALANF,N,0);
                for (xq=0;xq<FSIZE;xq=xq+1){
                ALANPWR[xq]=  (int)sqrt(  (ALANF[xq].real*ALANF[xq].real)  +
            (ALANF[xq].imag *ALANF[xq].imag ) );
                }
        }
        kalinlik=2;
        renk=RGB(255,128,0);
        SelectObject(hdc1,CreatePen(PS_SOLID,kalinlik,renk));
        MoveToEx(hdc1,0,200-alan0[0],NULL);
        yq= 200-alan0[0];
        yqF = 700-50*ALANPWR[0]/FSIZE;
        for (xq=0;xq<FSIZE;xq=xq++) {
            yq = 200-alan0[xq];
            LineTo(hdc1,xq,yq);
            MoveToEx(hdc1,xq,yq,NULL);          }
        if (FFTbit== 1)     {
            kalinlik=2;
            renk=RGB(128,128,128);
            SelectObject(hdc1,CreatePen(PS_SOLID,kalinlik,renk));
            MoveToEx(hdc1,xq,700-50*ALANPWR[0]/FSIZE,NULL);
            for(xq=0;xq<FSIZE;xq++){
                sprintf(szBuffer2,"Frekans");
                TextOut(hdcScreen, FSIZE-50,500,szBuffer2,strlen(szBuffer2) );
                yqF= 700-50*ALANPWR[xq]/FSIZE;
                LineTo(hdc1,xq,yqF);
                MoveToEx(hdc1,xq,yqF,NULL); }
        }
}
```

```
}
LRESULT PASCAL ErrorCallbackProc(HWND hWnd, int nErrID,LPSTR lpErrorText) {
   return (LRESULT) TRUE ;
}
BOOL FAR PASCAL __export About(hDlg, message, wParam, lParam)
HWND hDlg;
unsigned message;
WORD wParam;
LONG lParam;{
char        bak4[30]="COM1";
switch (message)    {
                        case WM_INITDIALOG:
                        SetDlgItemText(hDlg,
                        IDC_EDIT2,
                        bak4        );
                        return (TRUE);
                        case WM_COMMAND:
                                if (wParam == IDOK){
                                        GetDlgItemText(
                                        hDlg,
                                        IDC_EDIT2,
                                        PortNum,
                                        30  );
                                        EndDialog(hDlg, LOWORD(wParam));
                        return TRUE;        }
                        break;
                        }
   return (FALSE);
}
BOOL PortInitialize (){
  DWORD dwError, dwThreadID;
  DCB PortDCB;
  COMMTIMEOUTS CommTimeouts;
deneme1:
```

```
            hPort = CreateFile (PortNum,
                    GENERIC_WRITE | GENERIC_READ,
                   FILE_SHARE_READ | FILE_SHARE_WRITE,
                    NULL,
                    OPEN_EXISTING,
                    FILE_FLAG_RANDOM_ACCESS,
                    NULL);
if ( hPort == INVALID_HANDLE_VALUE ) goto deneme1;
PortDCB.DCBlength = sizeof (DCB);
GetCommState (hPort, &PortDCB);
PortDCB.BaudRate = 57600;
PortDCB.fBinary = TRUE;
PortDCB.fParity = FALSE;
PortDCB.fOutxCtsFlow = FALSE;
PortDCB.fOutxDsrFlow = FALSE;
PortDCB.fDtrControl = DTR_CONTROL_DISABLE;
PortDCB.fDsrSensitivity = FALSE;
PortDCB.fTXContinueOnXoff = TRUE;
PortDCB.fOutX = FALSE;
PortDCB.fInX = FALSE;
PortDCB.fErrorChar = FALSE;
PortDCB.fNull = FALSE;
PortDCB.fRtsControl = RTS_CONTROL_DISABLE;
PortDCB.fAbortOnError = FALSE;
PortDCB.ByteSize = 8;
PortDCB.Parity = NOPARITY;
PortDCB.StopBits = ONESTOPBIT;
if (!SetCommState (hPort, &PortDCB))  {
MessageBox (hWnd, TEXT("Unable to configure the serial port"),
            TEXT("Error"), MB_OK);
dwError = GetLastError ();
   return FALSE;
  }
 GetCommTimeouts (hPort, &CommTimeouts);
```

```
 CommTimeouts.ReadIntervalTimeout = 1000;                    //MAXDWORD;
 CommTimeouts.ReadTotalTimeoutMultiplier = 1000;     //0;
 CommTimeouts.ReadTotalTimeoutConstant = 1000;               //0;
 CommTimeouts.WriteTotalTimeoutMultiplier = 1000;    //10;
 CommTimeouts.WriteTotalTimeoutConstant = 1000;      //1000;
 if (!SetCommTimeouts (hPort, &CommTimeouts))  {
            MessageBox (hWnd, TEXT("Unable to set the time-out parameters"),
            TEXT("Error"), MB_OK);
             dwError = GetLastError ();
             return FALSE;
          }
 EscapeCommFunction (hPort, SETDTR);
 EscapeCommFunction (hPort, SETRTS);
 hReadThread = CreateThread(NULL,
                                0,
                                PortReadThread,
                                &dwThrdParam,
                                0,
                                &dwThreadId );
          if (hReadThread == NULL) {
                                sprintf(str,"Thread Oluşturulamadı");
                                MessageBox(hWnd, str, "Pencere", MB_ICONSTOP | MB_OK);
                                }
OKAY=1;
 return TRUE;
}
DWORD WINAPI PortReadThread (){
unsigned char       veri1,veri2,Byte;
int                 x,sayac,jj=1;
BOOL    bak;
DWORD   dwBytesTransferred;
 dosya1=fopen("EKG_verisi.dat","ab");
 sayac=254;
 gol=0;
 sayx=0;
```

```
  while (hh==0)   {
            sayac=sayac+1;
            if (sayac==255) {
            sayac=0;
basadon:
            bak=ReadFile (hPort, &Byte, 1, &dwBytesTransferred, 0);
            if (dwBytesTransferred==0) { PostMessage(hWnd,WM_USER,0,0);bekle1=1;goto atla;}
            if (Byte!=0xFF) goto basadon;
            bak=ReadFile (hPort, &Byte, 1, &dwBytesTransferred, 0);
            if (dwBytesTransferred==0)
            { PostMessage(hWnd,WM_USER,0,0);bekle1=1;goto atla;}
            if (Byte!=0xFF) goto basadon;
            }
            bak=ReadFile (hPort, &veri1, 1, &dwBytesTransferred, 0);
            if (dwBytesTransferred==0) { PostMessage(hWnd,WM_USER,0,0);bekle1=1;goto atla;}
            fputc(veri1,dosya1);
    bak=ReadFile (hPort, &veri2, 1, &dwBytesTransferred, 0);
            if (dwBytesTransferred==0) { PostMessage(hWnd,WM_USER,0,0);bekle1=1;goto atla;}
            fputc(veri2,dosya1);
            x = 0;
            x = veri2;
            x = x << 8;
            x = x & (0x0F00);
            x = x | veri1;
            x = x /16;
            veri1 = x;
            if ((ala1==0) && (goster==0)) {
                              alan1[jj] =veri1;
                              jj=jj+1;
if (jj>FSIZE-1) {
jj=1;ala1=1;ciz=0;PostMessage(hWnd,WM_USER+1,0,0);goto atla;}
            }
            if ((ala1==1) && (goster==0)){
                  alan0[jj] =veri1;
```

```
                    jj=jj+1;
                    if (jj>FSIZE-1) {jj=1;ala1=0;ciz=0;PostMessage(hWnd,WM_USER+1,0,0);goto atla;}
                        }
atla:
              if (bekle1==1)                    goto atla;
  }
  fclose(dosya1);
  CloseHandle(hReadThread);
  gol=1;
  return 0;
}
//////////////////////////// FFT.C ////////////////////////////////////
void FFT(COMPLEX *Y, int N, int t)  {
 COMPLEX temp1,temp2;
  int i,j,k;
  int upper_leg, lower_leg;
  int leg_diff;
  int num_stages=0;
  int index, step;
  if (t==1)    {
    for (i=0;i<=255;i++)       {
       w[i].real=w[i].real;
       w[i].imag=-w[i].imag;
       }
    }
 i=1;
  do   { num_stages+=1;
          i=i*2;
         } while (i!=N);
           leg_diff=N/2;
           step=1024/N;
*//  N-nokta FFT                                  */
  for (i=0;i<num_stages;i++){
    index=0;
```

```
  for (j=0;j<leg_diff;j++) {
    for (upper_leg=j;upper_leg<N;upper_leg+=(2*leg_diff)) {
      lower_leg=upper_leg+leg_diff;
      temp1.real=(Y[upper_leg]).real + (Y[lower_leg]).real;
      temp1.imag=(Y[upper_leg]).imag + (Y[lower_leg]).imag;
      temp2.real=(Y[upper_leg]).real - (Y[lower_leg]).real;
      temp2.imag=(Y[upper_leg]).imag - (Y[lower_leg]).imag;
      (Y[lower_leg]).real=temp2.real*(w[index]).real-temp2.imag*(w[index]).imag;
      (Y[lower_leg]).imag=temp2.real*(w[index]).imag+temp2.imag*(w[index]).real;
      (Y[upper_leg]).real=temp1.real;
      (Y[upper_leg]).imag=temp1.imag;
      }
    index+=step;
    }
  leg_diff=leg_diff/2;
  step*=2;
  }
j=0;
for (i=1;i<(N-1);i++)    {
  k=N/2;
  while (k<=j) {  j=j-k;
                  k=k/2;  }
  j=j+k;
  if (i<j)  {
    temp1.real=(Y[j]).real;
    temp1.imag=(Y[j]).imag;
    (Y[j]).real=(Y[i]).real;
    (Y[j]).imag=(Y[i]).imag;
    (Y[i]).real=temp1.real;
    (Y[i]).imag=temp1.imag;
    }
  }
return;
}
```

ÖZGEÇMİŞ

1982 yılında Elazığ'da doğdu. 1998 yılında Yıldız Teknik Üniversitesi Elektrik Mühendisliği Bölümü'nü kazandı. 2000 yılında iç transferle Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü'ne geçiş yaptı ve 2003 yılında Elektronik ve Haberleşme Mühendisi unvanını aldı. Aynı yıl, İstanbul Teknik Üniversitesi Biyomedikal Mühendisliği Programı'na Yüksek Lisans öğrencisi olarak kabul edildi. Kalp pili, defibrilatörler ve elektrofizyoloji üzerine çalışmalarına devam etmektedir.