

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MATLAB WEB FIGURE TABANLI
ELEKTROKARDİYOGRAM (EKG) BENZETİMİ**

YÜKSEK LİSANS TEZİ

Esra GÜNEY

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ

Tez Danışmanı : Yrd. Doç. Dr. Murat ÇAKIROĞLU

Haziran 2011

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**MATLAB WEB FIGURE TABANLI
ELEKTROKARDİYOGRAM (EKG) BENZETİMİ**

YÜKSEK LİSANS TEZİ

Esra GÜNEY

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ

Bu tez 29/06/2011 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



**Yrd.Doç.Dr.Murat
ÇAKIROĞLU**
Jüri Başkanı



**Prof.Dr.Etem
KÖKLÜKAYA**
Üye



**Yrd.Doç.Dr.Cüneyt
BAYILMIŞ**
Üye

TEŐEKKÜR

Tez alıőmamın gerekleőtirilmesinde yardım ve desteęini esirgemeyen, deęerli fikirleri ve yorumlarıyla beni yönlendiren danıőman hocam Sayın Yrd. Doę. Dr. Murat AKIROęLU'na sonsuz teőekkürlerimi sunarım.

Hayatım boyunca her konuda destek veren ve hiębir fedakârlıktan kaçınmayan deęerli anneme, babama ve kardeőlerime sonsuz teőekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ	vi
ÖZET.....	ix
SUMMARY.....	x
BÖLÜM 1.	
GİRİŞ.....	1
BÖLÜM 2.	
KALBİN ELEKTRİKSEL YAPISI VE EKG İŞARETLERİ	4
2.1. Kalbin Anatomik Yapısı ve Dolaşım Sistemi.....	4
2.2. Kalbin Elektriksel İletim Sistemi ve EKG İşareti.....	6
2.3. Elektrokardiyogram (EKG) ve Bileşenleri.....	8
2.4. Elektrokardiyogram Kaydetme Yöntemleri.....	10
2.4.1. Bipolar taraf derivasyonları.....	11
2.4.1.1. Derivasyon eksenleri ve einthoven üçgeni.....	12
2.4.2. Unipolar taraf derivasyonları.....	13
2.4.3. Unipolar göğüs derivasyonları.....	13
2.5. EKG Kullanım Alanları.....	14
2.6. EKG Ritim Türleri.....	15
2.6.1. Normal Sinüs Ritim.....	15
2.6.2. Kalp ritim bozuklukları.....	15
2.6.2.1. Sinüs Bradikardi.....	16
2.6.2.2. Sinüs Taşikardi.....	17

2.6.2.3. Junctional Ritim.....	17
2.6.2.4. Akselere Junctional Ritim.....	18
2.6.2.5. Birinci Derece AV Blok.....	18
2.6.2.6. İdioventriküler Ritim.....	19
2.6.2.7. Akselere İdioventriküler Ritim.....	19
2.6.2.8. Supraventriküler Taşikardi.....	19
2.6.2.9. Monomorfik Ventriküler Taşikardi.....	20
BÖLÜM 3.	
MATLAB WEB FIGURE TABANLI EKG BENZETİMİ.....	21
3.1. Sistem Mimarisi.....	21
3.2. MATLAB Web Figure Tabanlı EKG Benzetiminin Tasarım Basamakları.....	23
3.2.1. Sistem davranışlarının MATLAB ile kodlanması.....	24
3.2.1.1. EKG sinyallerinin matematiksel modeli.....	24
3.2.1.2. EKG sinyalinin benzetim yazılımı davranışlarının MATLAB ile kodlanması.....	28
3.2.2. MATLAB Builder NE ve MATLAB Web Figure yardımıyla .NET bileşenlerinin oluşturulması.....	29
3.2.3. Tasarımda ASP.NET platformunun kullanımı.....	32
BÖLÜM 4.	
TASARLANAN MATLAB WEB FIGURE TABANLI BENZETİM YAZILIMININ ÖZELLİKLERİ VE KULLANIMI.....	35
BÖLÜM 5.	
SONUÇLAR	48
KAYNAKLAR.....	49
EKLER.....	52
Ek A.....	52
Ek B.....	68
ÖZGEÇMİŞ.....	92

SİMGELER VE KISALTMALAR LİSTESİ

AV	: Atrioventriküler
CLS	: Common Language Specification
ECG	: Electrocardiogram
EKG	: Elektrokardiyogram
SA	: Sinoatrial
sn	: Saniye
SVT	: Supraventriküler Taşikardiler
VT	: Ventriküler Taşikardi

ŞEKİLLER LİSTESİ

Şekil 2.1.	Kalbin anatomik yapısı [21].....	5
Şekil 2.2.	Kalbin elektriksel iletim sistemi [21].....	7
Şekil 2.3.	Kalbin fiziksel hareketi sonucu oluşan elektriksel sinyaller [11]..	8
Şekil 2.4.	EKG işareti genlik [22].....	9
Şekil 2.5.	EKG işareti süre [22].....	9
Şekil 2.6.	EKG işareti ve dalga aralıklarının kalpteki elektriksel etkinlik karşılıkları.....	10
Şekil 2.7.	Derivasyonlar [9].....	11
Şekil 2.8.	Bipolar taraf derivasyonları [9].....	12
Şekil 2.9.	Einthoven üçgeni [9].....	13
Şekil 2.10.	Eksenlerin kaydırılması [9].....	13
Şekil 2.11.	60°'lik açılarla üç eksenli referans sistemi [9].....	13
Şekil 2.12.	Unipolar taraf derivasyonları [9].....	13
Şekil 2.13.	Unipolar göğüs derivasyonları [9].....	14
Şekil 3.1.	Sistem mimarisi.....	22
Şekil 3.2.	Önerilen sistem mimarisinin akış diyagramı.....	22
Şekil 3.3.	Tasarımda kullanılan araçlar ve tasarım basamaklarının akış diyagramı.....	23
Şekil 3.4.	QRS dalga biçiminin üretilmesi [16].....	25
Şekil 3.5.	P dalgasının üretilmesi [16].....	27
Şekil 3.6.	Sistem davranışlarının MATLAB ile kodlanması.....	29
Şekil 3.7.	Deployment Tool aracının çağırılması ve yeni proje oluşturulması işlemi.....	31
Şekil 3.8.	Matlab Builder NE ile .m dosyasının sınıfa eklenmesi işlemi.....	31
Şekil 3.9.	Matlab Builder NE ile .NET bileşeni (.dll) oluşturma işlemi.....	31

Şekil 3.10.	Web Figure Control aracının eklenmesi işlemi.....	32
Şekil 3.11.	Oluşturulan .NET bileşeni (.dll) ve Web Figure Control aracının çalışma ortamına eklenmesi işlemi.....	33
Şekil 3.12.	Oluşturulan .NET bileşeni ve Web Figure Control aracının kullanılabilmesi için gerekli kodlar.....	34
Şekil 4.1.	Tasarlanan MATLAB Web Figure tabanlı EKG benzetimi.....	35
Şekil 4.2.	EKG dalga biçiminin temel özellikleri değiştirildiğinde oluşan sinyal.....	36
Şekil 4.3.	Sinüs Ritim grafiği ve detayları.....	37
Şekil 4.4.	Sinüs Ritim grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	37
Şekil 4.5.	Sinüs Ritim grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	37
Şekil 4.6.	Sinüs Bradikardi grafiği ve detayları.....	38
Şekil 4.7.	Sinüs Bradikardi grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	38
Şekil 4.8.	Sinüs Bradikardi grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	38
Şekil 4.9.	Sinüs Taşikardi grafiği ve detayları.....	39
Şekil 4.10.	Sinüs Taşikardi grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	39
Şekil 4.11.	Sinüs Taşikardi grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	39
Şekil 4.12.	Junctional Ritim grafiği ve detayları.....	40
Şekil 4.13.	Junctional Ritim grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	40
Şekil 4.14.	Junctional Ritim grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	40
Şekil 4.15.	Monomorfik VT grafiği ve detayları.....	41
Şekil 4.16.	Monomorfik VT grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	41
Şekil 4.17.	Monomorfik VT grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	41

Şekil 4.18.	İdioventiküler Ritim grafiği ve detayları.....	42
Şekil 4.19.	İdioventiküler Ritim grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	42
Şekil 4.20.	İdioventiküler Ritim grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	42
Şekil 4.21.	Akselere İdioventriküler Ritim grafiği ve detayları.....	43
Şekil 4.22.	Akselere İdioventriküler Ritim grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	43
Şekil 4.23.	Akselere İdioventriküler Ritim grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	43
Şekil 4.24.	Supraventriküler Taşikardi grafiği ve detayları.....	44
Şekil 4.25.	Supraventriküler Taşikardi grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	44
Şekil 4.26.	Supraventriküler Taşikardi grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	44
Şekil 4.27.	Akselere Junctional Ritim grafiği ve detayları.....	45
Şekil 4.28.	Akselere Junctional Ritim grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	45
Şekil 4.29.	Akselere Junctional Ritim grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	45
Şekil 4.30.	Birinci Derece AV Blok grafiği ve detayları.....	46
Şekil 4.31.	Birinci Derece AV Blok grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği.....	46
Şekil 4.32.	Birinci Derece AV Blok grafiği üzerinde Web Figure'ün 3-D görünüm özelliği.....	46

ÖZET

Anahtar kelimeler: EKG, Web Tabanlı Benzetim, MATLAB Web Figure, MATLAB Builder NE

EKG (Elektrokardiyogram), kalbin durumu ve çalışması hakkında bilgi içeren bir sinyaldir. Günümüzde bu sinyaller yardımıyla birçok kalp rahatsızlığının teşhisi yüksek başarımlar ile gerçekleştirilebilmektedir. Ancak kalp sinyallerinin tanınması ve ayırt edilmesi rahatsızlıkların teşhisinde önemli bir gereklilik olarak karşımıza çıkmaktadır. Bu tezde, bahsedilen gereksinimler üzerine odaklanılarak tıp ve biyomedikal mühendisliği alanlarında eğitim alan veya kendisini bu alanda yetiştirmek isteyen kişilerin eğitimi kolaylaştıracak MATLAB Web Figure tabanlı yeni bir web benzetim yazılımı tasarımı gerçekleştirilmiştir. Bu benzetim yazılımı esneklik ve görsellik açısından olanaklar sunabilmektedir. Tasarlanan benzetim yazılımı yardımıyla kullanıcılar farklı atış hızlarına sahip kalp ritimlerini ve farklı ritim bozukluklarının sinyallerini çizdirebilmekte, 3-boyutlu olarak gösterebilmekte, sinyallerin istenilen kısımları büyütülmekte ve hareket ettirebilmektedir. Bu avantajları sebebiyle tasarlanan benzetim yazılımı, EKG sinyallerinin web üzerinden eğitimini kolaylaştıracak ve daha görsel hale getirecektir.

ELECTROCARDIOGRAM (ECG) SIMULATOR BASED ON MATLAB WEB FIGURE

SUMMARY

Key Words: ECG, Web Based Simulator, MATLAB Web Figure, MATLAB Builder NE

ECG (Electrocardiogram) is a signal including knowledge about condition and operation of heart. Nowadays, many heart disease diagnoses are performed at high performance using these signals. However, recognition and classification of heart signals are very important requirement for diagnosis. In this thesis, emphasizing on the above mentioned requirements, MATLAB Web Figure based a new web simulator is designed, which is aimed to facilitate the education of those students who are studying medicine and biomedical engineering or who would like to educate themselves in this field. This simulator can provide many flexibility and visualization opportunities. By using the designed simulator, the users are able to plot heart rhythm in different pulse rates and different arrhythmia, to display ECG signals in 3-dimension view, to zoom and move on graphics. Because of these advantages designed simulator, facilitates and visualizes the web based education of ECG signals.

BÖLÜM 1.GİRİŞ

Elektrokardiyogram (EKG), kalbin elektriksel potansiyelini gösteren ve kalbin durumu hakkında bilgi içeren bir sinyaldir [1, 2]. Bu sinyal genellikle düzgün bir şekilde vücuda yerleştirilen belirli sayıdaki elektrotlar aracılığıyla kaydedilmektedir [3]. Her kalp atımının karşılığı olan P,Q,R,S,T dalgalarından oluşmuş kompleks bir sinyal olan EKG yardımıyla kalp ile ilgili birçok anormal durum tespit edilebilmektedir [4]. Kalp sağlığının durumunu tespit ederken EKG dalga biçiminin şekli ve kalp atış oranına bakılır [2]. Bu nedenle EKG kalp hastalıklarına teşhis koymak için kullanılan standart bir araçtır [1]. EKG sinyallerinin doğru bir şekilde tanınması ve yorumlanabilmesi doktorların hastalıkları teşhis etmedeki başarısını yükseltecektir. Bu sebeple, özellikle tıp ve biyomedikal mühendisliği alanında eğitim gören kişilerin kendilerini bu alanda eğitmeleri son derece önemlidir.

Günümüzde, Tıp ve Biyo-Medikal Mühendisliği eğitiminde görsel ve interaktif bir eğitim gerçekleştirmek amacıyla bilgisayar tabanlı benzetimler yaygın bir şekilde kullanılmaktadır [5]. EKG sistemleri de benzetim araçlarının sıklıkla kullanıldığı alanlardan bir tanesidir [6]. Bu yazılımlar aracılığıyla herhangi bir EKG makinesi kullanmaksızın çeşitli dalga formları kolay bir şekilde oluşturulabilmektedir. Kullanım kolaylığı ve düşük maliyet gibi sebeplerden ötürü benzetim sistemleri eğitim uygulamalarında yaygın bir şekilde kullanılmaktadır. Literatürde EKG sistemleri ile ilgili çeşitli benzetim araçları sunulmuştur. Gerçekleştirilen bu çalışmalar yazılımsal ve donanımsal olmak üzere iki ayrı kategoride incelenebilmektedir. Literatürdeki donanımsal benzetim araçlarına örnek olarak [1] nolu çalışmada Digital Signal Controller (DSPIC) tabanlı EKG simülör biyomedikal mühendisliği öğrencilerinin eğitimine destek amacıyla tasarlanmıştır. [6] nolu çalışmada mikrodenetleyici denetiminde tasarlanan bir EKG simülör sistemi sayesinde kalp ritim bozuklukları simüle edilmiştir. [7]'de Çift kanallı EKG simülör ve pik detektörü tasarlanmıştır. [8]'de EKG cihazlarının kalibrasyonunu

gerçekleştirecek ve değişik EKG sinyallerini grafik lcd ekranda gösterecek EKG simülâtör tasarlanmıştır. [9] nolu çalışmada ise uzaktan kontrol edilebilen kalp cihazına LabVIEW programı aracılığıyla yazılım geliştirilmiştir. [10]'da bir sunucu üzerinden gezici hastaların kardiyak aktivitelerinin izlenebilmesine olanak sağlayan kablosuz EKG görüntüleme sistemi tasarımı yapılmıştır. [11] nolu çalışmada vücut yüzeyinden elektrotlarla alınarak elde edilen EKG sinyallerinin sayısal iletim teknikleri ile kablosuz olarak iletilebilmesi için PC tabanlı EKG biyotelemetri sistemi tasarımı yapılmıştır. [12]'de ise Zigbee modülleri kullanılarak EKG/PPG monitör sistemi tasarlanmıştır. Bir diğer çalışmada [13] ise Linux tabanlı EKG ölçüm ve izleme sistemi geliştirilmiştir. Ayrıca bunların dışında kullanılan ve ticari olarak satılan birçok donanımsal EKG simülâtörler de bulunmaktadır.

Literatürdeki yazılımsal çalışmalar ise Java, C++, C#, MATLAB gibi farklı platformlarda geliştirilmiştir. Yazılımsal olarak EKG benzetim yazılımlarına örnek olarak ise [14]'de Physionet (www.physionet.org) aracılığıyla Java ve CGI tabanlı yeni bir web tabanlı servis hizmeti sunulmuştur. Bu hizmet ile bir web tarayıcısı üzerinden Physionet'te bulunan EKG veritabanındaki verilerin grafikleri gösterilmektedir. [15]'de çok kanallı elektrokardiyografik işaretlerin bilgisayar ortamına aktararak görüntülenmesi, işlenmesi, saklanması ve sonradan izlenmek üzere çağırılmasına yönelik Windows ortamında kullanıcı arayüz yazılımı gerçekleştirilmiştir. Programlama ortamı olarak Visual Basic 6.0 programı kullanılmıştır. [16] nolu çalışmada gerçekleştirilen simülâtör MATLAB tabanlı bir EKG simülâtördür. [17]'de The Six Second ECG (Cardiac Rhythm Simulator) isimli eğitim amaçlı bir web benzetim yazılımı sunulmuştur. [18] nolu çalışmada simECG isimli bir EKG benzetimi tasarlanmış ve bu benzetim yazılımı C++ programlama dili ile yazılmıştır. [19] nolu çalışmada ise tıp öğrencileri için web tabanlı EKG yorumlama programı önerilmiştir. Literatürde yapılan bu çalışmaların örnekleri artırılabilir.

Bu tez çalışmasında MATLAB Builder NE ve MATLAB Web Figure ile web tabanlı bir EKG benzetimi tasarlanmıştır. Bu tez çalışmasının amacı tıp ve mühendislik eğitimi alan kişilere yönelik bir EKG benzetimi tasarlamak ve kullanıcılara sunmaktır. Bu benzetim yazılımı ile normal ve ritim bozukluğu olan EKG dalga

biçimleri üretilmektedir. Bu benzetim yazılımı gerçekte EKG makinesi kullanmaksızın normal ve anormal (hastalıklı) EKG dalga şekillerini kolaylıkla oluşturabilmektedir. Kullanıcı isteğe bağlı olarak temel EKG sinyallerinin genlik, süre ve kalp atış oranı gibi özelliklerini değiştirerek, EKG dalga biçimini oluşturan bileşenleri kolaylıkla analiz edebilmektedir. EKG ritim türleri olan normal ritim ve seçilen 9 tane kalp ritim bozukluğu türü tasarlanan benzetim yazılımında bulunmaktadır. Kullanıcılar herhangi bir yazılıma ihtiyaç duymaksızın internet bağlantısı olan bilgisayar, dizüstü bilgisayar, PDA veya cep telefonu vb. gibi bir aygıt üzerinden sadece web tarayıcısı kullanarak web tabanlı benzetime erişebilirler. Bununla birlikte “.NET” teknolojisi, MATLAB Builder NE ve Web Figure’ün sağlamış olduğu görsellik sayesinde çizdirilen grafikler 3 boyutlu olarak gösterilebilmekte, grafikler üzerinde yakınlaştırma/uzaklaştırma ve hareket ettirme gibi özellikler kullanılabilen ve böylece kalp ritimlerinin detayları kolaylıkla incelenebilmektedir. Bu çalışmanın diğer çalışmalardan en önemli farkı esneklik, görsellik ve kullanım kolaylığı üzerine odaklanmasının yanında MATLAB Builder NE ve Web Figure’ün biyomedikal bir uygulamada ilk kez kullanılan bir çalışma olmasıdır. Bu nedenle bu çalışma yapılacak diğer web tabanlı biyomedikal uygulamalar için örnek temsil edebilir.

Bu tez çalışmasında diğer bölümlerde; Bölüm 2’de ilk olarak kalbin anatomik yapısı ve dolaşım sistemi hakkında bilgi verildikten sonra kalbin elektriksel iletim sistemi hakkında temel bilgiler verilmektedir. Ardından Elektrokardiyogram (EKG) ve Elektrokardiyogramı oluşturan bileşenler konusunda bilgi verilmektedir. Daha sonra Elektrokardiyogram kaydetme yöntemleri ve EKG kullanım alanlarına değinilmektedir. Ardından EKG ritim türleri olan normal ritim ve seçilen 9 tane kalp ritim bozukluğu hakkında genel bilgiler verilmektedir. Bölüm 3’de geliştirilen MATLAB Web Figure tabanlı EKG benzetiminin sistem mimarisi anlatıldıktan sonra .NET teknolojisi, MATLAB Builder NE ve Web Figure kullanılarak geliştirilen web tabanlı EKG benzetiminin tasarım basamakları sunulmaktadır. Bölüm 4’de tasarlanan MATLAB Web Figure tabanlı benzetim yazılımının özellikleri ve kullanımı anlatılmaktadır. Bölüm 5’de ise sunulan tez çalışmasının sonuçlarına yer verilmektedir.

BÖLÜM 2. KALBİN ELEKTRİKSEL YAPISI VE EKG İŞARETLERİ

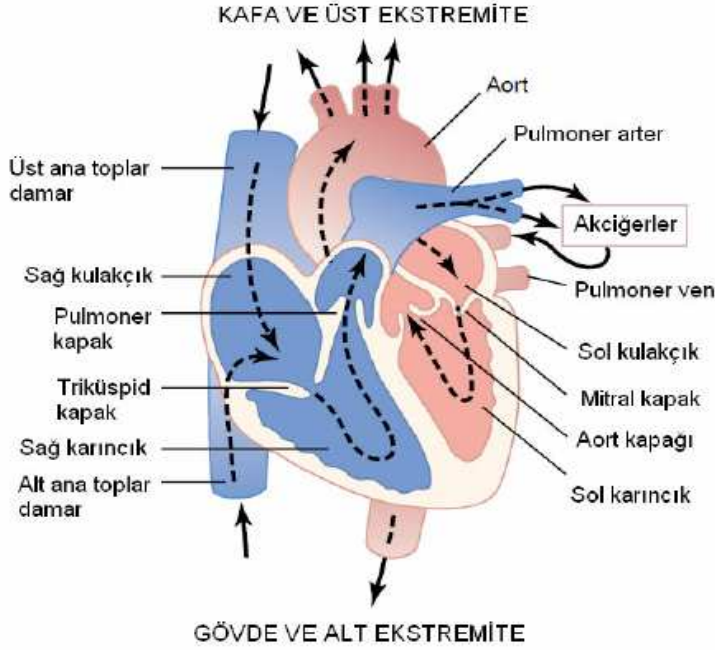
Bu bölümde ilk olarak kalbin anatomik yapısı ve dolaşım sistemi hakkında bilgi verildikten sonra, kalpte oluşan elektriksel potansiyelleri anlamak açısından kalbin elektriksel iletim sistemine değinilmektedir. Ardından kalp kasının kasılıp gevşemesi sonucu oluşan Elektrokardiyogram ve EKG işaretinin bileşenleri hakkında bilgi verilmektedir. Daha sonra EKG kaydetme yöntemleri ve EKG kullanım alanlarından bahsedilmektedir. Ardından tıp literatüründe aritmi olarak ifade edilen EKG işaretindeki şekil ve ritim bozuklukları hakkında bilgi verilmektedir.

2.1. Kalbin Anatomik Yapısı ve Dolaşım Sistemi

Kalp ve damarların meydana getirdiği sisteme dolaşım sistemi denir. Bu sistemin tam merkezinde kalp bulunmakta, damarlar kalpten çıkan ve kalbe tekrar geri dönen kapalı bir boru sistemini meydana getirmektedir. Kanın damar sistemi içinde belli bir basınç altında dolaşmasını sağlamak dolaşım sisteminin asıl fonksiyonudur. Kan belli bir basınç farkı altında, diğer bir deyişle basıncın yüksek olduğu bölgeden düşük olduğu bölgeye doğru akmaktadır. Bu özelliği ile diğer sıvıların prensiplerine benzer davranmaktadır. Kalp yüksek ve düşük basıncı oluşturacak biçimde yani bir emme basma tulumba gibi çalışmaktadır [20].

Kalbin yapısında dört tane boşluk vardır (Şekil 2.1). Üst taraftaki iki boşluğa atrium (kulakçık), alttakilere ise ventrikül (karıncık) adı verilmektedir. Atriumlar, sağ atrium ve sol atrium; ventriküller ise sağ ventrikül ve sol ventrikül bölümlerinden oluşmaktadır. Atriumlar ventriküllere kapaklar aracılığı ile bağlıdır. Trikuspid kapak, sağ atrium ile sağ ventrikül arasındaki kapağa verilen addır. Sol atrium ile sol ventrikül arasındaki kapağa ise bikuspid veya mitral kapak adı verilir. Kalpten çıkan iki ana damar aort ve pulmoner arterdir. Aort oksijence zengin kanı tüm vücut

dokularına dağıtan arter sisteminin ana damarı olup sol ventrikülden çıkmaktadır. Pulmoner arter ise CO₂'li kanı oksijenlenmesi için akciğerlere götüren sistemin ana damarı olup sağ ventrikülden çıkmaktadır [20].



Şekil 2.1. Kalbin anatomik yapısı [21]

Kalp, atrium ve ventriküllerden oluşan iki odacıklı ritmik uyarana gösteren birer pompa gibi düşünülebilir. Atriumlar kanı ventrikül içine gönderebilmek için zayıf bir pompa görevi üstlenir [21]. Ventriküller ise asıl pompa görevini üstlenmekte olup kanı damar sistemi içinde pompalarlar. Sağ ventrikül karbondioksit oranı yüksek olan kanı temizlenmek üzere akciğerlere, sol ventrikül ise oksijen oranı yüksek kanı tüm vücuda pompalamaktadır. Böylece organizmada her ikisi de kalpten başlayıp kalpte sonlanan iki dolaşım sistemi görülmektedir. Bunlardan biri kalbin sol ventrikülünden başlayarak organizmayı dolaştıktan sonra sağ atriumda sonlanan sistemik dolaşım veya büyük dolaşımdır. Diğeri sağ ventrikülden başlayıp akciğerleri dolaştıktan sonra sol atriumda sonlanmakta olan pulmoner veya küçük dolaşımdır. Sistemik dolaşımın yüksek basınçlı, pulmoner dolaşımın düşük basınçlı bir dolaşım sistemi olması bu iki sistemin dolaşım dinamiği açısından aralarındaki tek farktır [20].

2.2. Kalbin Elektriksel İletim Sistemi ve EKG İşareti

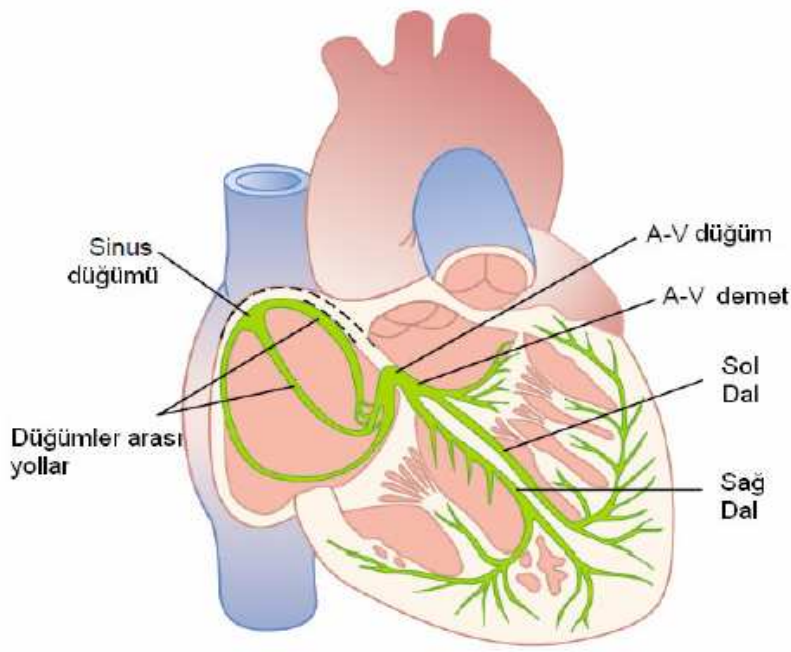
Kalp, birbirini izleyen dönemler boyunca hiç durmadan belirli bir elektrik potansiyel üreten ve bu etkinliğin ardından kontraksiyon olarak da adlandırılan mekanik işi yapan bir organdır. Bu nedenle kalp bir elektromotor kuvvet kaynağı olarak düşünülebilir [9].

Kalp kası, çok sayıda ince uzun hücreden oluşan bir dokuya sahiptir. Yan uzantılarının yardımıyla yer yer birleşen kalp kası hücreleri, bir takım diskler aracılığı ile uç uca bağlanarak dizilirler. Kalp kasının kasılması için ön koşul, hücre membranının elektriksel anlamda uyarımıdır. Kalsiyum (Ca^{++}) iyonu elektriksel anlamda uyarımı mekanik harekete dönüştüren ana etmendir. Aksiyon potansiyeli ya da elektriksel sistol, dinlenme durumundaki miyokard hücresinde belirli uyarım tarafından başlatılan elektrokimyasal içerikli etkinliğe denmektedir. Aksiyon potansiyeli, depolarizasyon (kasılma) ve repolarizasyon (gevşeme) aşamalarından oluşmaktadır. Hücrenin repolarizasyonu izleyerek başlangıç durumundaki özelliklerine dönmesi ile dinlenme potansiyeli (elektriksel diyastol) meydana gelir [22]. Kalbin elektriksel dolayısı ile mekanik etkinliğinin belirli bir düzen içinde art arda tekrar edebilmesi için, elektriksel periyodun bir döneminde hücreye giren ya da hücreden çıkan iyonun o periyod içinde yerine dönmesi gerekmektedir. Bu akışın olmasını sağlayan iyon geçitleri kalp hücre membranında yer almaktadırlar [23].

Kalbin elektriksel iletim sistemi sinoatrial düğüm (SA), his demeti, atrioventriküler düğüm, demet kolları ve purkinje fiberlerinden oluşmaktadır (Şekil 2.2) [6]. Şekil 2.2’de ritmik uyarıların üretildiği sinüs düğümü (sinoatrial düğüm veya SA düğüm); uyarıları sinüs düğümünden atrioventriküler (AV) düğüme ileten düğümler arası yollar; uyarıların atriumdan ventriküllere geçmeden önce geciktirme işlemini yapan AV düğüm; uyarıların atriumdan ventriküllere iletilme işlemini sağlayan AV demet; ve uyarıların ventriküllerin bütün yüzeyine iletilmesini sağlayan AV demetinin sağ ve sol dalı ile Purkinje fiberleri görülmektedir [21].

Kalbin normal çalışması esnasında uyarılar SA düğümden çıkmaktadır. Bu sebepten dolayı SA düğüm pacemaker olarak tanımlanmaktadır. Pacemaker, hareketi başlatan,

hareketin hızını belirleyen anlamındadır. SA düğümde kendiliğinden oluşan aksiyon potansiyeli, depolarizasyon dalgası halinde kalbin tümüne yayılmaktadır [21].

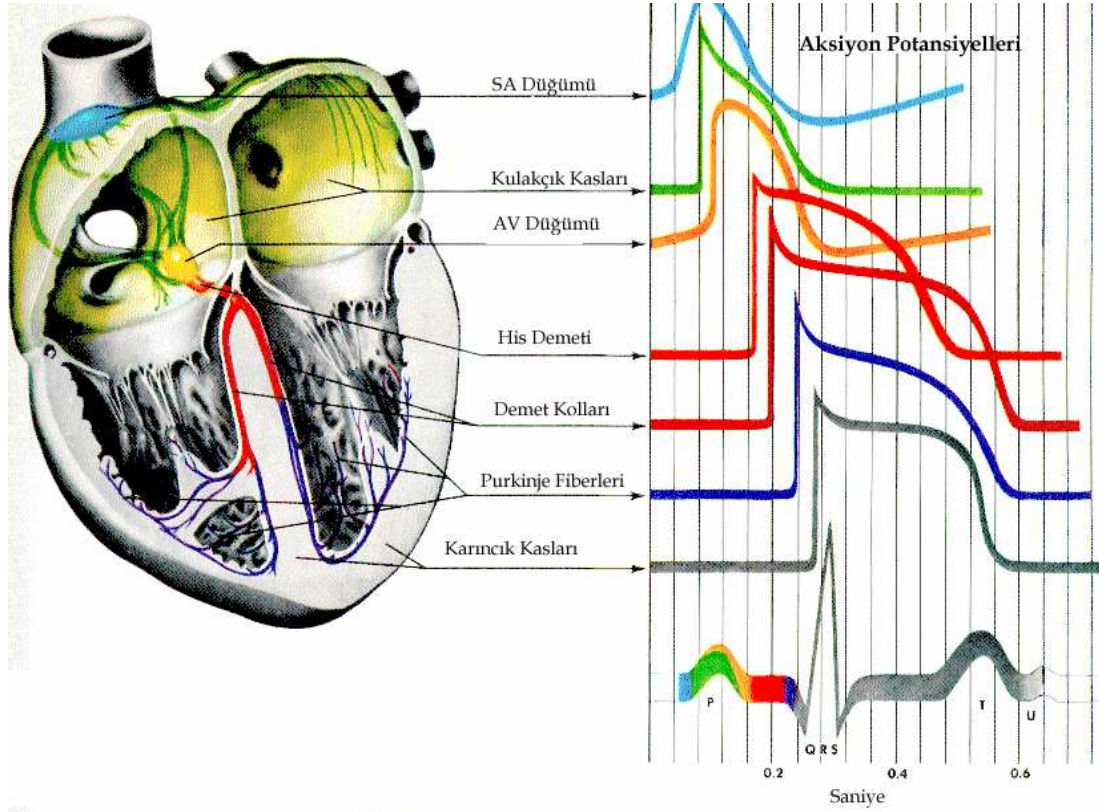


Şekil 2.2. Kalbin elektriksel iletim sistemi [21]

SA düğümünde meydana gelen aksiyon potansiyeli, atriumlar üzerindeki iletim yolları üzerinden hızlı bir şekilde yayılarak atriumların kasılmasını gerçekleştirir ve buradaki kan ventriküllere basılır. Atriumlarda aksiyon potansiyelinin hızı, 30 cm/s'dir. SA ve AV düğümleri arasındaki özel iletim hatlarında ise hız 45 cm/s 'dir. SA düğümde meydana gelen aksiyon potansiyeli 30-50 ms sonra AV düğüme ulaşır. Bu süre, atriumların içlerindeki kanı tümüyle ventriküllere doldurmaları için yeterli olmamaktadır. Bu nedenle ventriküllerin kasılmasının bir süre sonra yapılması gerekmektedir. Bu işlem, bir geciktirme elemanı gibi çalışan AV düğümde, aksiyon potansiyelinin 110 ms kadar geciktirilmesiyle elde edilir [24].

Ventriküllerin uyarılması purkinje fiberleri ile sağlanır. Burada ise aksiyon potansiyelinin hızı 2-4 m/s kadardır. Purkinje fiberleri yardımıyla uyarılan miyokardiyum kasılır ve buradaki kan arterlere pompalanır. Kalp kasının aynı anda kasılmasından sonra genliği oldukça büyük bir elektriksel işaret oluşur. Elektrokardiyogram (EKG) olarak adlandırılan bu işaret vücut üzerinden algılanabilir

[21]. Bu iletim sürelerinde kalp kasları tarafından üretilen elektriksel sinyallerin dalga şekilleri Şekil 2.3’ de görülmektedir.



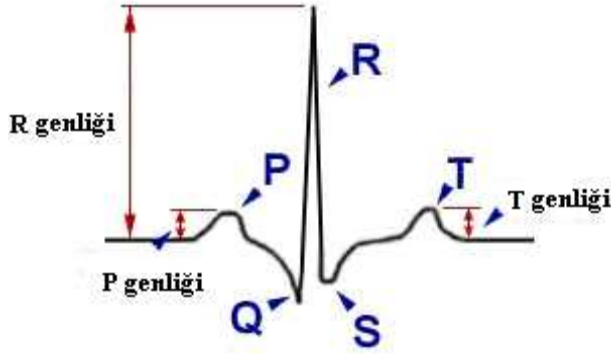
Şekil 2.3. Kalbin fiziksel hareketi sonucu oluşan elektriksel sinyaller [11]

2.3. Elektrokardiyogram (EKG) ve Bileşenleri

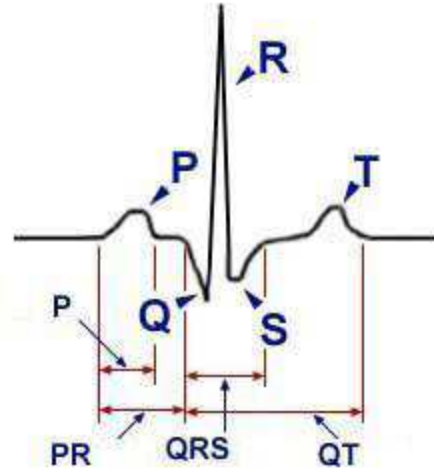
Kalbin kasılması ve gevşemesi sırasında ortaya çıkardığı elektriksel potansiyel değişiminin bedenün iletken bir ortam gibi davranmasından yararlanılarak zamana karşı çizdirildikten sonra kaydedilmesine elektrokardiyografi, bu kaydedilen (görüntülenen) değişimlere elektrokardiyogram (EKG), elektrokardiyografi sırasında kullanılan ve temelde değiştirilip geliştirilmiş bir galvanometre olan ağıta ise elektrokardiyograf adı verilir [23].

Elektrokardiyogram üzerinde, kalbin elektriksel potansiyeline ilişkin yön, genlik ve zaman içinde değişim gibi bilgiler bulunmaktadır. Normal EKG, izoelektrik hat üzerinde sırayla bulunan P, Q, R, S, T, U adları verilen dalgalardan oluşmaktadır. Q,

R ve S dalgaları EKG işaretinin en belirgin ve kolay gözlemlenebilir kısmı olan QRS kompleksini oluştururlar (Şekil 2.4, Şekil 2.5) [22].



Şekil 2.4. EKG işareti genlik [22]

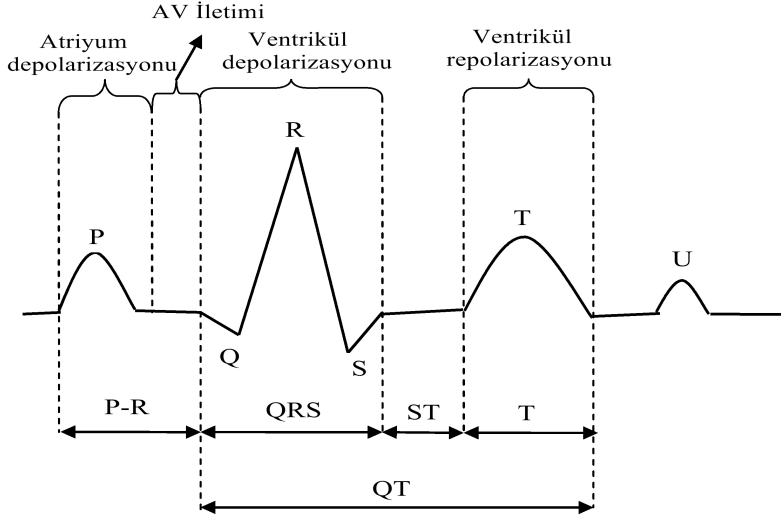


Şekil 2.5. EKG işareti süre [22]

SA düğümü tarafından üretilen doğal uyarı, ön, orta ve arka internodal yollar vasıtası ile önce sağ kulakçığa, ardından kulakçık septumuna ve son olarak sol kulakçığa yayılır. Bu duruma göre sol kulakçığın depolarizasyonu, sağ kulakçığın depolarizasyonundan daha sonra meydana gelir. Kulakçık depolarizasyonu elektrokardiyogramda P dalgasına karşılık gelir [22]. P dalgasının yüksekliği, atriyumların fonksiyonel etkinliği ile ilgili bilgi verir ve normal insanda 0.1s sürer [20]. Kulakçık repolarizasyonu ise genellikle QRS kompleksi içinde maskelenir. Kulakçık depolarizasyonunu gerçekleştirerek internodal yollar vasıtasıyla AV kavşağına ulaşan uyarı önce AV düğümüne gitmektedir. AV düğümünün üst kesiminde uyarı iletimi yavaşlayarak, düğümün orta kesiminde en düşük hıza iner ve düğümün alt kesiminde tekrardan hızlanmaya başlayarak purkinje ipliklerinde en yüksek değere çıkmaktadır. Elektrokardiyogramdaki PR, daha doğru bir ifade ile PQ aralığı, SA düğümünden kaynaklanan uyarının karıncıklara ulaşması için geçen süreyi göstermektedir [22] ve his demeti iletim zamanını ifade etmektedir [20].

Atrioventriküler düğüm ile his demetine ulaşan SA uyarı, sağ ve sol dallara girerek karıncıklara yayılır. Uyarı ya da depolarizasyon, karıncık duvarında, endokardiyal yüzden epikardiyal yüze doğru ilerler. QRS kompleksi, karıncık depolarizasyonu sonucu oluşan dalgalardır [22]. QRS kompleksi ventriküler bozuklukları göstermektedir ve normal insanda 0.08s kadardır [20]. QRS aralığı, uyarının

karıncıklara yayılma süresini gösterir. QRS kompleksini takip eden T dalgası, karıncık repolarizasyonunu gösterir [22]. Ventriküler bozukluklar T dalgasında da bozukluklara sebep olur [20].



Şekil 2.6. EKG işareti ve dalga aralıklarının kalpteki elektriksel etkinlik karşılıkları

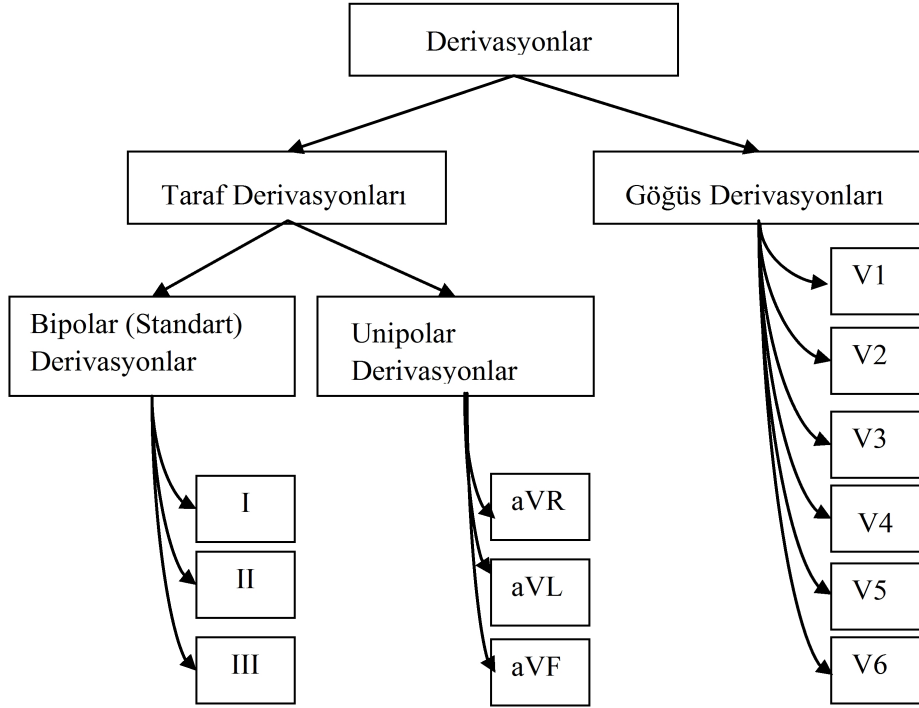
Elektrokardiyogramın depolarizasyon ve repolarizasyon dalgalarından oluştuğu Şekil 2.6'da görülmektedir. Bundan dolayı elektrokardiyografide depolarizasyon ve repolarizasyonun birbirinden ayırt edilmesi önemlidir [21].

2.4. Elektrokardiyogram Kaydetme Yöntemleri

Kalpte üretilen ve beden yüzeyine iletilen elektriksel potansiyel değişikliklerinin elektrokardiyografa aktarılabilmesi için kollara, bacaklara ve göğüs duvarı üzerindeki belli noktalara elektrot adı verilen metal parçacıklar yerleştirilir. Elektrotların belirli bir düzen içinde ve iletken tellerin yardımıyla elektrokardiyografin uçlarına bağlanması ile derivasyonlar elde edilir [9].

EKG örneklerinin açıklanıp kavranmasında derivasyonlar önemli bir yere sahiptir [22]. Günlük EKG uygulamaları sırasında kullanılan 12 adet derivasyon, taraf ve göğüs derivasyonları olmak üzere iki gruba ayrılır. Taraf derivasyonları da bipolar (standart) ve unipolar olarak iki gruba ayrılır. Göğüs derivasyonlarında ise unipolar

tipte yerleşim öngörülür [9]. Sözü edilen bu derivasyonların gruplandırılması ve adlandırılmaları Şekil 2.7’de verilmiştir.



Şekil 2.7. Derivasyonlar [9]

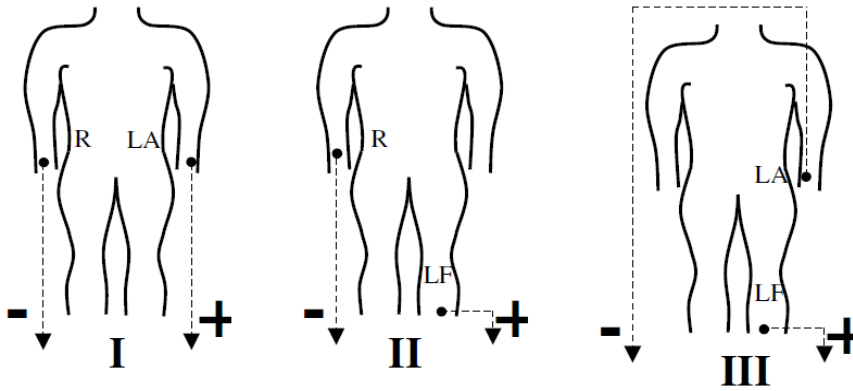
2.4.1. Bipolar taraf derivasyonları

Bipolar taraf derivasyonları, bedenin iki noktası arasındaki potansiyel farkı saptamak için kullanılırlar ve I, II, III olarak gösterilirler [22]. Ölçümler, kolay yapılabilmesi için Einthoven üçgeninin köşe noktalarından değil, bu noktalara yakın olan kol ve bacaklar üzerinde yapılır [11].

Elektrokardiyografi aygıtının negatif ucuna bağlı elektrodun sağ kola, pozitif ucuna bağlı elektrodun ise sol kola bağlanması ile elde edilen bipolar taraf derivasyonu, derivasyon I'dir. Bu derivasyon ile sağ ve sol kol arasındaki potansiyel farkı saptanmış olur (Şekil 2.8) [22].

Elektrokardiyografin negatif ucuna bağlı elektrodun sağ kola, pozitif ucuna bağlı elektrodun ise sol bacağa bağlanması ile derivasyon II elde edilir. Bu sayede sağ kol ve sol bacak arasındaki potansiyel farkı ölçülür (Şekil 2.8) [22].

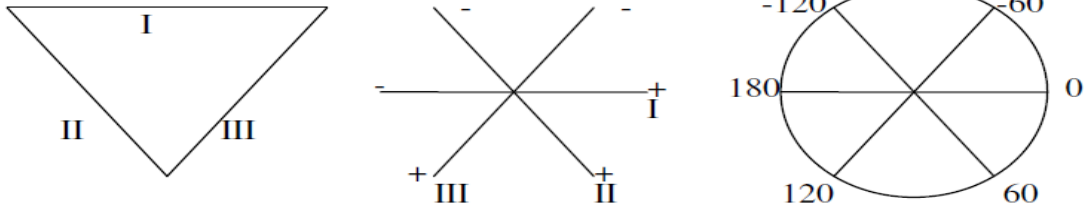
Elektrokardiyografin negatif ucuna bağlı elektrodu sol kola, pozitif ucuna bağlı elektrodu sol bacağa bağlayarak ise derivasyon III düzenlenir. Bu derivasyon sayesinde ise sol kol ile sol bacak arasındaki potansiyel fark ölçülür (Şekil 2.8) [22].



Şekil 2.8. Bipolar taraf derivasyonları [9]

2.4.1.1. Derivasyon eksenleri ve einthoven üçgeni

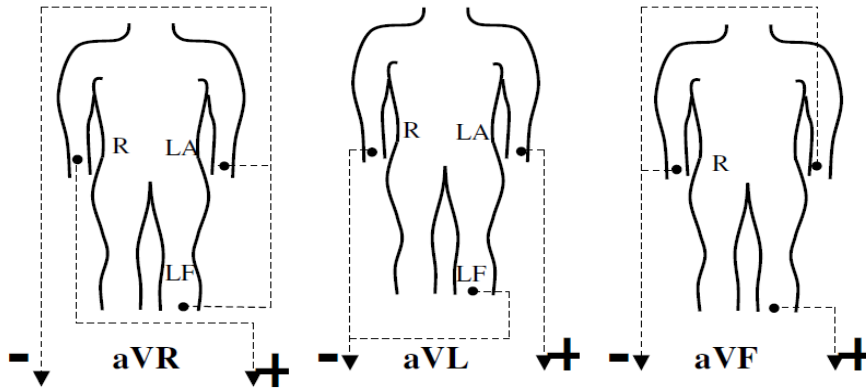
Bipolar taraf derivasyonlarına ilişkin elektrot çiftlerini birleştiren kavramsal düz çizgiler derivasyon eksenleri olarak adlandırılır. Derivasyon ekseninde, pozitif elektrotun bulunduğu uç “pozitif”, negatif elektrotun bulunduğu uç ise “negatif” dir. Eksenin uçlara eşit uzaklıktaki orta noktası “sıfır noktası”dır. I, II ve III nolu derivasyonlara ilişkin eksenler çizildiğinde “einthoven üçgeni” olarak bilinen bir şekil elde edilir. Bu üçgen ile ilgili; eşkenar bir üçgen olduğu, köşelere eş uzaklıkta sıfır noktasının bulunduğu ve üçgenin merkezinde kalbin anatomik merkezinin olduğu, kalbin anatomik merkezi ile elektriksel merkezinin çakışık olduğu şeklinde yaklaşımlar sıralanabilir. Bahsedilen üçgenin kenarlarını oluşturan eksenler ağırlık merkezi olan sıfır noktasından geçecek şekilde kaydırılarak bir referans sistemi elde edilir. Frontal düzlemde yer alan bu referans sisteminden, kalbin elektriksel eksenini belirlemek için faydalanılabilir (Şekil 2.9, Şekil 2.10, Şekil 2.11) [9].



Şekil 2.9. Einthoven üçgeni Şekil 2.10. Eksenlerin kaydırılması Şekil 2.11. 60°'lik açılarla üç eksenli referans sistemi [9]

2.4.2. Unipolar taraf derivasyonları

Elektrokardiyogram işaretlerini kaydetmek için kullanılan unipolar taraf derivasyonları yönteminde, hastanın iki uzvu dirençler aracılığıyla elektrokardiyografin negatif ucuna bağlanır. Bir diğer uzvuna da elektrokardiyografin pozitif ucu bağlanır. Pozitif uç sağ kolda olduğunda aVR derivasyonu, sol kolda olduğunda aVL derivasyonu, sol bacakta olduğunda ise aVF derivasyonu olarak adlandırılan unipolar taraf derivasyonları elde edilir [21]. Derivasyonlar isimlendirilirken önlerine konan “a” harfi augmented yani güçlendirilmiş anlamındadır. Ayrıca “V” voltaj, “R” sağ kol, “L” sol kol ve “F” harfi sol bacağı ifade etmektedir [25]. Bu yöntemle elde edilen çıkış işareti %50 oranında yükselir [9]. Şekil 2.12’de bu adlandırmalara göre yapılmış elektrokardiyografik elektrot bağlantılar görülmektedir.

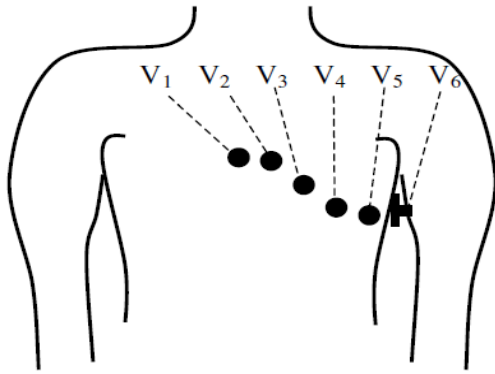


Şekil 2.12. Unipolar taraf derivasyonları [9]

2.4.3. Unipolar göğüs derivasyonları

Elektrokardiyogram göğsün ön yüzeyinde Şekil 2.13’deki gibi önceden belirlenmiş herhangi bir noktaya bir elektrotun yerleştirilmesiyle direkt kalp üzerinden

kaydedilmektedir. Bu elektrot, elektrokardiyografin pozitif ucuna bağlanır. Negatif elektrot ise eşit direnç değerleriyle sağ kol, sol kol ve sol bacağına bağlanır. Bu yöntemle genellikle altı standart göğüs derivasyonu elde edilir. Göğsün ön yüzeyinden Şekil 2.13'deki noktalara elektrotlar yerleştirilerek ölçülen bu derivasyonlar sırasıyla V1, V2, V3, V4, V5 ve V6 olarak adlandırılır. Kalp yüzeyi göğüs duvarına yakın olduğundan dolayı her bir derivasyon kalp kasının elektriksel potansiyelini elektrot aracılığıyla hemen kaydedebilmektedir [21].



Şekil 2.13. Unipolar göğüs derivasyonları [9]

2.5. EKG Kullanım Alanları

EKG, kalbin normal çalışıp çalışmadığını veya bazı anormal durumların olup olmadığını belirlerken (ritim bozukluğu, kalp atışlarının gereğinden fazla veya az olması gibi) ve aniden olan veya daha önceden gerçekleşen kalp krizlerinin zararlarını belirlemede kullanılır. İletim anormalliklerini tespit etmek (kalp tıkanıklığı gibi) ve kalbin fiziksel durumu hakkında bilgi öğrenmek için (sol ventriküler hipertropi, mitral stenosis gibi) kullanılır. Kalple ilgili olmayan bazı rahatsızlıkları da ortaya çıkarmak için (pulmonary embolism, hypothermia gibi) kullanılabilir. İstemik kalp rahatsızlığında egzersiz dayanıklılık testi esnasında bir görüntüleme aracı olarak kullanılmaktadır [24].

Ayrıca potansiyel olarak öğretmenin tehlikeli ya da zor olacağı durumlarda kliniksel açıdan öğrenme ve öğretmeyi kolaylaştırmak amacıyla EKG benzetimi uygulamaları kullanılır. EKG benzetimleri kalp sesleri ve akciğer sesleri simülasyonunda, etkileşimli fizyolojik modellemede ve sosyal olarak gerçek hasta ile kullanılmasının

zor olduđu kliniksel alıřmalarda bu ihtiyacı azaltabilmek iin eđitim amalı kullanılırlar. İnsan faktörleri mühendisliđi ve tıbbi cihazlar iin mühendislik alıřmalarında kullanılabilirliđi vardır. Biliřsel mühendislik alıřmalarında da kullanılmaktadır [7]. EKG kullanım alanları bu řekilde sıralanabilir.

2.6. EKG Ritim Türleri

2.6.1. Normal Sinüs Ritim

Kalbin normal ritmine, regüler ritim adı verilir ve bu ritim sinüs düđümünün kontrolü altındadır. Normal sinüs ritmi 60-100 atım/dakika'dır. Normal bir ritimde, EKG'de aynı dalgalar arasındaki uzaklık daima birbirine eşittir [25]. Her uyarı SA düđümden çıkar. P dalgası her vuruda aynı görünüme sahip olup tümü SA düđümden çıkar. PR aralıđı 0,12-0,20 sn'dir. QRS aralıđı ise 0,08-0,12 sn'dir [21].

2.6.2. Kalp ritim bozuklukları

Aritmi ritim bozukluđu anlamındadır [23]. Aritmi yerine disritmi terimi de sıklıkla kullanılmaktadır [25]. eřitli kalp hastalıkları sonucunda kalp atıř ritminde bozulmalar meydana gelebilir [23]. Bunun sonucunda da ařađıda sıralanan aritmiler görülebilir.

Aritminin anlaşılabilmesi iin önce normal EKG'nin bilinmesi gereklidir [25]. EKG iřaretinde, bozucu etkiler, genlikleri ok küçük olan P ve T dalgaları ile aynı iřaret seviyesinde kaldıđından dolayı bu dalgaların algılanması zorlařır. Bunun yerine, nispeten büyük genliđe sahip olan R dalgalarının varlıđı algılanabilir. Bu durumda R dalgalarının genliđi ve zaman iindeki yeri aısından incelenerek hastalık teřhisine gidilir [23].

Sözlükteki karřılıđının dıřında bir anlam yüklenen aritmi terimi ile uyarının odađı, hızı ve düzeni ile ilgili normal dıřı nitelikler ile uyarı iletimindeki bozukluklar ifade edilmek istenir. Bařka bir ifade ile aritmilere neden olan temel etmenler, uyarının oluřumundaki ve/veya iletimindeki bozukluklardır. Uyarı iletimini belirleyen temel

etmenler, depolarizasyon'un hızı, aksiyon potansiyeli'nin genliği ve uyarının taşındığı dokuların uyarılabilirlikleri, diğer bir deyiş ile uyarılabilirlik düzeyleridir [21].

Aritmiler kalbin uyarı yada ileti sisteminde meydana gelen bozukluklar sonucunda oluşacağı gibi, elektrolit dengesizlikleri, endokrin bozuklukları, ilaç intoksikasyonları, kalp kası hipertrofisi, konnektif doku hastalığı, miyokardiyal iskemi, asit-baz dengesizliği, stres, kahve-çay-tütün kullanımı, hücrel hipoksi, alkol tüketimi ile de ortaya çıkabilmektedir [25].

Tez kapsamında normal ritim, Sinüs bradikardi, Sinüs taşikardi, Junctional ritim Akselere junctional ritim, Birinci derece AV blok, İdioventriküler ritim, Akselere idioventriküler ritim, Supraventriküler taşikardi, Monomorfik ventriküler taşikardi olmak üzere on adet EKG ritim türü sunulmuştur.

2.6.2.1. Sinüs Bradikardi

Kalp atış hızının normal alt sınır değerinin altına düşmesiyle ortaya çıkan durumdur [22]. SA düğümünden çıkan uyarıların dakikada 60'ın altında olduğu görülmektedir. Bu kalp ritim bozukluğu türünün EKG özelliklerinde kalbin atım hızı 40-60 atım/dakika arasında olup ritimleri düzenlidir [6]. P dalgası mevcut ve normaldir. PR aralığı normaldir ve 0,12-0,20 sn'dir. QRS kompleksi mevcuttur, normal sürede 0,06-0,10 sn arasındadır [25]. P dalgaları ve QRS kompleksleri arasında 1:1 ilişki devam etmektedir. Her QRS kompleksinin öncesinde bir P dalgası görülür. Özellikle yaşlılarda olmak üzere sağlıklı insanlarda sık rastlanılan bir aritmi türüdür. Genellikle benign (iyi huylu) bir aritmi tipi olarak görülür. Diyastol süresini uzatıp ventrikül doluşunu arttırması nedeniyle yararlı olduğu kabul edilebilir [21]. Fizyolojik nedenlerden (sporcu, genç, yaşlı insanlarda vagal tonus artışına bağlı olarak) olduğu gibi, patolojik nedenlerden (hipotiroidi, hipotermi, kusma, intrakraniyal basınç artışı, tifo, hiperreaktif karotis, hasta sinüs sendromu veya betabloker, verapamil, digitalis, antiaritmikler gibi ilaçlar sebebiyle) dolayı da olabilir. Bu sebepten dolayı altta yatan neden saptanarak tedavi edilmelidir [26]. Kardiyak debide azalma yaparak serebral belirti ve bulgulara yol açmadıkça tedavi

yapılmaz. Semptomatik sinüs bradikardisinde ilk hazırlanacak ilaç atropindir. Yanıt alınamayan durumlarda geçiçi pace-maker uygulanarak tedavisi yapılır [25].

2.6.2.2. Sinüs Taşikardi

Kalp atış hızının normal üst sınır değerinin artmasıyla ortaya çıkan durumdur [22]. SA (sinoatrial) düğümünden çıkan uyarılar dakikada 100'ün üstüne çıkmaktadır. Bu kalp ritim bozukluğunun EKG özelliklerinde kalbin atım hızı 100-160 atım/dk arasındadır ve ritimleri düzenlidir. P dalgasının büyüklüğü ve şekli normaldir [6]. PR aralığı normaldir ve 0,12-0,20 sn'dir. QRS kompleksi mevcuttur ve normal sürede 0,06-0,10 sn arasındadır. Her QRS kompleksinin öncesinde bir P dalgası görülür [25]. Bebeklerde ve çocuklarda sık rastlanılan bir kalp ritim bozukluğudur [21]. Kalp atışındaki hızlanma dakikalar, günler veya daha uzun sürelerde olabilir. Taşikardi birçok durumda hayati tehlike taşımaz; ancak, kişinin sinirli ve gerilimli hissetmesine neden olur [22]. Sinüs taşikardisi genelde ritim bozukluğu değil başka bir hastalık nedeniyle ihtiyacın artmasından dolayı görülmektedir [27]. Fizyolojik nedenler (Küçük çocuklar, zihinsel veya bedensel stresler, emosyonel reaksiyonlar, sempatik tonusun arttığı durumlar) ve patolojik nedenlerden (ateş, hipertiroidi, anemi, hipoksi, hipotansiyon, kanama, şok, kalp yetmezliği, miyokardit, cor pulmonale, hiperkinetik kalp sendromu, hipovolemi, adrenerjik stimülasyon, anksiyete vs.) dolayı olabilir. Bu yüzden altta yatan neden bulunup tedavi edilmelidir [26,28]. Yeterli ağrı tedavisi, kalp yetmezliği için diüretik ve oksijen, hipovolemi için sıvı tedavisi, ateş veya perikardit için anti-inflamatuar ilaçlar, iskemi için beta bloker veya nitrogliserin kullanılır. Altta yatan nedenin saptanamadığı veya emin olunamadığı durumlarda, esmolol gibi çok kısa etkili beta blokerler denenerek tedavi yapılır [28].

2.6.2.3. Junctional Ritim

SA düğümünün baskılanması nedeniyle, AV düğümünde bir odak, vuru düzenleyici (pacemaker) olarak SA düğümünün yerine geçer. QRS kompleksinden hemen önce ya da sonra P dalgaları görülür [6] veya P dalgaları görülmeyebilir. D-II derivasyonunda negatif P dalgaları olabilir. PR intervali değişken olabilir [29]. Sıklıkla escape (kaçak) bir ritim olarak ortaya çıktığı görülür [30]. Junctional ritim

hızı 40-60 atım/dakika arasındadır ve bu ritim düzenlidir [6]. SA nod injürisi, SA nodda parasempatik hakimiyet, hipoksi gibi nedenlerden dolayı ortaya çıkar. Kalp hızınının belirgin derecede düşmesi sonucu Kardiak Output azalmasına bağlı semptomlar gelişebilir. Genellikle geçici ve kendini sınırlayan bir aritmi türüdür. Bradikardiye bağlı semptom olmadığı sürece müdahaleye gerek yoktur. Tedavide atropin, transkutaneöz ya da transvenöz pacing, dopamin, epinefrin kullanılabilir [29].

2.6.2.4. Akselere Junctional Ritim

Sıklıkla aşırı sayısal bilgi alımı nedeniyle AV (atrioventriküler) düğüm içinde artmış otomatisiteden kaynaklandığı görülmektedir. Hızı 60-100 atım/dakika olan akselere junctional ritimde, P dalgası kulakçığa retrograd gönderilir ve pozitif derivasyonlarda negatif, negatif derivasyonlarda ise pozitif olduğu görülür [6]. QRS kompleksi normaldir [30]. Miyokardit, akut romatizmal ateş, akut inferior miyokard infarktüsü, dijital entoksikasyonu, yavaş yol ablasyonu sonrası meydana gelebilir [31]. Asemptomatik tedavide dikkatlice gözlem yapılmalıdır. Semptomatik tedavide potasyum seviyeleri kontrol edilir ve amiodaron, beta-adrenerjik blokerler (esmolol, labetalol, metoprolol) veya kalsiyum kanal blokerleri kullanılabilir [32].

2.6.2.5. Birinci Derece AV Blok

Bu ritim türünde elektriksel uyarı SA düğümde ilk ortaya çıkar ve AV düğüme iletilir. Uyarı AV kavşağa ulaştığında karıncıklara iletilmeden önce geçici olarak gecikir. Birinci derece AV blokta ritim düzenlidir ve atım hızı normaldir [6]. PR aralığının 0.21 sn'den daha uzun olması ile bilinmektedir. Atriyumdan çıkan tüm uyarılar AV düğüm üzerinden Purkinje sistemine iletilir [21]. İlaç toksisitesi (özellikle digoksin), beta-adrenerjik blokerler veya kalsiyum kanal blokerlerinin toksisitesinden, kinidin veya prokainamid gibi antiaritmik ilaçlardan, hipo ya da hiperkalemiden ve atriyal iletim sisteminin kronik dejeneratif hastalıklarından dolayı meydana gelebilir [32]. Genellikle normal kalplerde, koroner arter hastalığında veya dijital uygulamasında görülür [33]. Sağlıklı bireylerin %1'inde görülebilir [29]. Genellikle tedavi gerektirmeyen bir ritim bozukluğudur [21].

2.6.2.6. İdiovventriküler Ritim

Ventriküler ektojik, bir odak uyarıları idiovventriküler hızda gönderdiği zaman oluşur. Sıklıkla total kalp bloğu ile beraber görülür. Hızı 30-40 atım/dakika arasındadır. Bu kalp ritim bozukluğunun ritmi düzenlidir ve bu ritimde P dalgası yoktur [6]. QRS'lerin geniş ve genellikle düzenli veya hafif düzensiz olduğu görülür. Bu bulgunun olduğu hastalarda bu kalp ritim bozukluğuna sebep olan hastalığın tedavisi vakit geçirilmeden yapılmalıdır [26]. Eğer nabız varsa; hızı bir an önce artırma girişimi (pacemaker) ve atropine, dopamine, epinephrine, isoproterenol ilaçları kullanılabilir. Nabız yoksa kardiopulmoner resusitasyon protokolü ve entübasyon uygulanmalıdır [29].

2.6.2.7. Akselere İdiovventriküler Ritim

Ventrikül kaynaklı bir ritimdir ve birkaç saniye/dakika devam edebilir [34]. Akselere (hızlandırılmış) idiovventriküler ritimin ventrikül oranı genellikle 40 ile 100-120 atım/dakika arasındadır [35]. 6-10 atımlık normal sinüs ritmi ile dönüşümlü olarak görülebilir. AIVR'nin başlangıcında veya bitişinde füzyon atımları, yavaş hız sebebiyle de capture atımları görülebilir [34]. Uyarılar ventrikülden çıktığı için bu ritimde QRS kompleksleri geniştir ve QRS'lerin önünde P dalgası bulunmamaktadır [36]. Dijital toksisitesi AIVR'ye sebep olabilir. Çoğunlukla yapısal kalp hastalığı olanlarda akut MI, tıkalı koroner arterin açılmasıyla reperfüzyon sonrası veya resusitasyon sonrasında görülmektedir. Ayrıca yapısal kalp hastalığı olmayan normal kalpte de nadiren görülebilmektedir [34]. Tehlikesizdir ve herhangi bir tedavi gerektirmez [35].

2.6.2.8. Supraventriküler Taşikardi

Supraventriküler taşikardiler (SVT), his düğümü seviyesi üzerinden kaynak alırlar. SVT'lerin daha çok dar kompleksli taşikardi şeklinde olduğu görülmektedir [25]. Sık karşılaşılan acil servise çarpıntı şikayeti ile başvuran hastaların çoğunda görülmektedir. Bu çarpıntılar (taşikardiler) genellikle atriyumdan (kulakçık) başlar ya da atriyumları içine aldığından dolayı bu şekilde adlandırılır. Aniden başlayıp

aniden sonlandıđı görölmektedir ve genellikle hızları 150-280 atım/dakika arasındadır. Hayati tehlikesi yoktur fakat oldukça rahatsız eder, hastaların yaşam kalitesini düşürür. Kalp ritim uzmanları tarafından yapılan çeşitli sınıflamaları bulunmaktadır [27]. Tedavide tercih edilen ilaç verapamil, beta-bloker, digoxin İV uygulamadır [25].

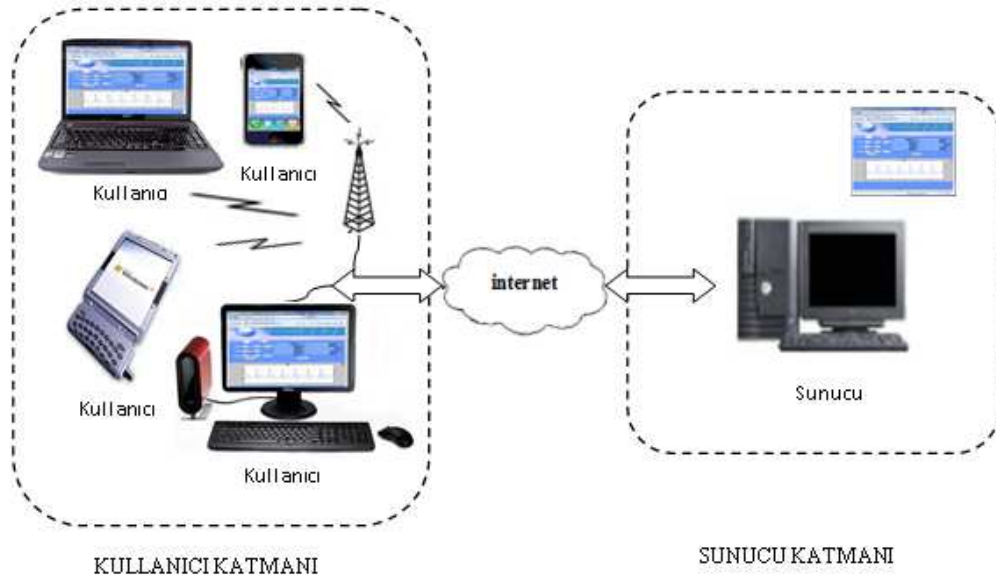
2.6.2.9. Monomorfik Ventriküler Taşikardi

Ventriküler taşikardi kendi morfolojisine göre; tüm QRS komplekslerinin benzer (monomorfik) ya da farklı (polimorfik) olup olmadığına bađlı olarak, monomorfik veya polimorfik olarak sınıflandırılabilir. Ventriküler aktivasyon sırası sabit olduğunda, elektrokardiyografik şekil aynı kalır ve aynı QRS EKG dalgasının kendini tekrarladığı görülür. Bu ritim monomorfik VT (Ventriküler taşikardi) olarak adlandırılır [37]. Kalp atış hızı sıklıkla 120'den fazladır. Ritmin çoğunlukla düzenli olduğu görülür. P dalgası, hızlı VT'de çoğunlukla görülmemektedir. AV noddan çıkan ventrikül hızından daha yavaş bir impulsla meydana gelirler. AV disosiasyon vardır. QRS geniş ve anormal morfolojidedir. Monomorfik VT'de ventriküler fonksiyon önemlidir. Stabil hastada acil kardiyoversiyon gerekmemektedir. İmkan varsa, hastanın ventrikül fonksiyonu değerlendirilir. Semptomatik ise ilk tedavi seçeneđi senkronize kardiyoversiyondur. Eğer kardiyoversiyon mümkün değilse/tercih edilmezse/başarısızsa farmakolojik tedavi yapılmalıdır. Ventrikül fonksiyonu korunmuşsa; amiodaron (sınıf IIb), IV prokainamid (sınıf IIa), IV sotalol, lidokain (Sınıf indeterminate) ilaçları kullanılabilir. Ventrikül fonksiyonu yetersizse; amiodaron, amiodaron başarısız ise kardiyoversiyon yöntemi uygulanır [28].

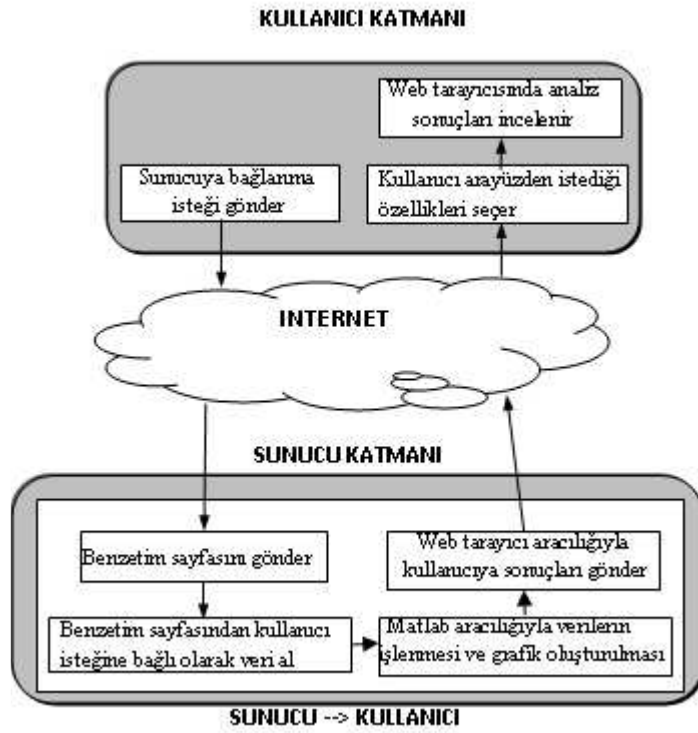
BÖLÜM 3. MATLAB WEB FIGURE TABANLI EKG BENZETİMİ

3.1. Sistem Mimarisi

Tasarlanan MATLAB Web Figure tabanlı benzetimin sistem mimarisi Şekil 3.1’de görüldüğü gibi kullanıcı ve sunucu katmanlarından oluşmaktadır. Sunucu katmanı, web tarayıcı aracılığıyla İnternet üzerinden bilgileri alır ve kullanıcının talepleri doğrultusunda işleyerek görüntüler. Sunucu katmanında tüm bu işlemleri yapmak için MATLAB Builder NE ve .NET teknolojisi kullanılmıştır. Bu sebeple sunucuda MATLAB programının yüklü olmasına gerek yoktur. Veriler sunucu tarafından işlendikten ve istenilen EKG grafikleri oluşturulduktan sonra bu veriler internet aracılığı ile kullanıcılara gönderilir. İstemci (kullanıcı) katmanı ise, sunucuya bağlanmak, benzetim sayfasından istenilen özellikleri seçmek ve sunucu tarafında oluşturulan grafiği almaktan sorumludur. Kullanıcılar, herhangi bir program kurmadan sadece web tarayıcısı aracılığı ile sunucuya erişebilmektedir. Bu nedenle, sadece internet bağlantısı olan bir bilgisayar, laptop, pda veya akıllı telefon gibi aygıtlar ile benzetim yazılımı kullanılabilir. Önerilen sistem mimarisinin akış diyagramı ise Şekil 3.2’de görülmektedir.



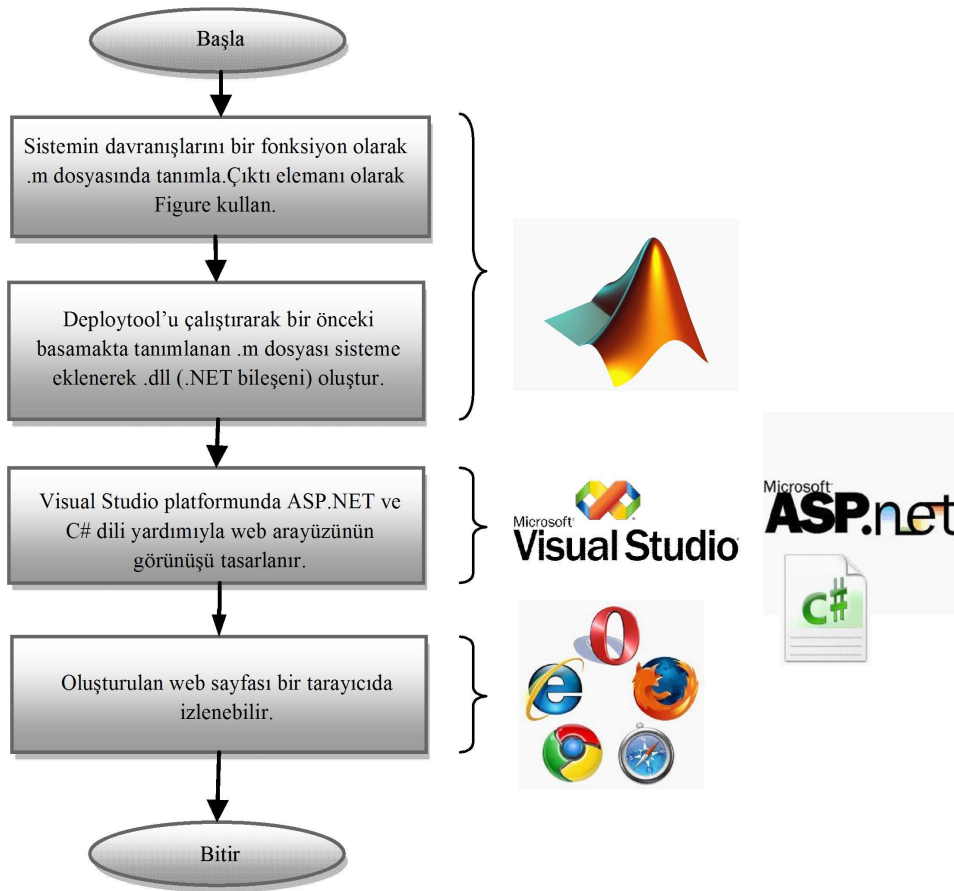
Şekil 3.1. Sistem mimarisi



Şekil 3.2. Önerilen sistem mimarisinin akış diyagramı

3.2. MATLAB Web Figure Tabanlı EKG Benzetiminin Tasarım Basamakları

Gerçekleştirilen MATLAB Web Figure tabanlı EKG benzetiminin tasarımında; MATLAB Builder NE, MATLAB Web Figure [38,39,40], Microsoft Visual Studio 2008, ASP.NET tabanlı .NET teknolojisi (.NET Framework 3.5) [41,42] kullanılmıştır. MATLAB programı ve MATLAB Builder NE ile hazırlanmış .NET bileşeni, bu bileşen ile uyumlu biçimde çalışabilen ve arayüz tasarımının gerçekleştirildiği Microsoft Visual Studio 2008 programı ve Web tasarımının alt yapısı ASP.NET ve kullanılan programlama dili ise C# olarak seçilmiştir. Bu bölümde gerçekleştirilen EKG benzetiminin tasarım basamakları anlatılmaktadır. Tasarımda kullanılan araçlar ve tasarım basamakları Şekil 3.3'de gösterilmektedir. Bu tasarımda her bir basamak aşağıda farklı bölümlerde ayrıntılı olarak sunulmaktadır.



Şekil 3.3. Tasarımda kullanılan araçlar ve tasarım basamaklarının akış diyagramı

3.2.1. Sistem davranışlarının MATLAB ile kodlanması

3.2.1.1. EKG sinyallerinin matematiksel modeli

EKG sinyallerini temsil etmek için Fourier serileri kullanılabilir [16]. Bu çalışmada EKG sinyallerini modellemek için simülasyon davranışlarının tanımlanmasında prensip olarak Fourier serileri kullanılmaktadır. Bu çalışmada EKG sinyallerinin matematiksel modellenmesinde [16] nolu çalışmadan faydalanılmaktadır.

$$f(x) = (a_0 / 2) + \sum_{n=1}^{\infty} [a_n \cos(n\omega x) + b_n \sin(n\omega x)] \quad (3.1)$$

Fourier analizi teoremine göre; periyodik bir $f(x)$ fonksiyonu periyot içinde sonlu sayıda noktalar dışında tek değerli olarak tanımlı, sonlu sayıda ekstremuma sahip, kendisi ve birinci türevi parça parça sürekli ise, Eşitlik 3.1'de tanımlanan seri, düzgün olarak $f(x)$ fonksiyonuna yakınsar. $f(x)$ fonksiyonunun bu şekildeki açılımına Fourier açılımı; a_n ve b_n katsayılarına Fourier katsayıları; seriye ise Fourier serisi denir [16,43]. Fourier serileri bir periyodik fonksiyonu basit dalgalı fonksiyonların (sinüs ve kosinüs) toplamına çevirmektedir [44]. Fourier serisi süreksizlik noktalarında fonksiyonun sağ ve sol değerlerinin ortalamasına yakınsar. Bu bahsedilen yakınsama şartlarına Dirichlet şartları adı verilir. Bu şartlar yeterlilik şartlarının sağlanması içindir [16,43]. Bu şartları içerdiğinden dolayı EKG sinyali temsil etmek için Fourier serisi kullanılabilir. EKG sinyali kalp atışına göre temel frekans ile periyodik olarak belirlenmiştir.

$$\omega = 2\pi/T, \quad T = 2l \quad (3.2)$$

Eşitlik 3.2'teki ifadeler Eşitlik 3.1'deki Fourier serisinde yerine yazılarak Eşitlik 3.3'deki denklem elde edilir.

$$f(x) = (a_0 / 2) + \sum_{n=1}^{\infty} a_n \cos(n\pi x / l) + \sum_{n=1}^{\infty} b_n \sin(n\pi x / l) \quad (3.3)$$

Yukarıda Eşitlik 3.3'de verilen Fourier serisinde a_n , b_n katsayıları ve a_0 aşağıdaki şekilde hesaplanır.

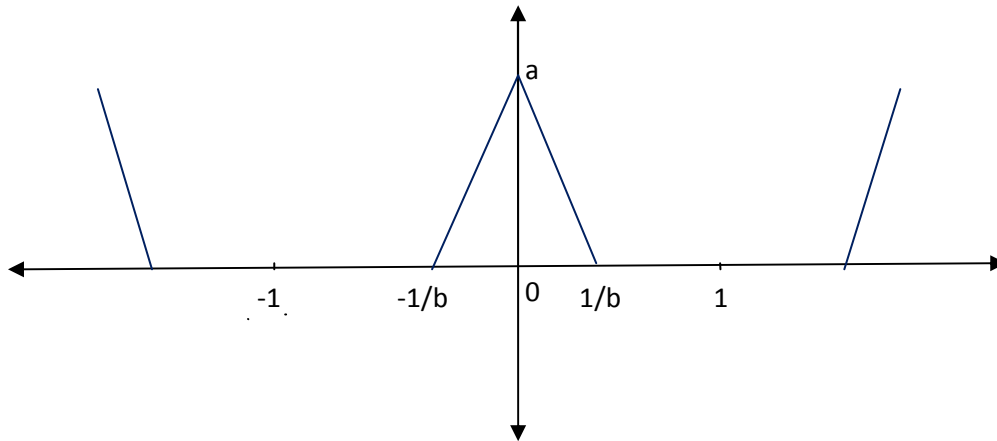
$$a_0 = (1/l) \int_T f(x) dx \quad (3.4)$$

$$a_n = (1/l) \int_T f(x) \cos(n\pi x / l) dx, \quad n=1,2,3,\dots \quad (3.5)$$

$$b_n = (1/l) \int_T f(x) \sin(n\pi x / l) dx, \quad n=1,2,3,\dots \quad (3.6)$$

Fourier serilerinin çözümü yapılırken, serinin her terimine karşı gelen çözümler yapıldıktan sonra bu çözümler toplanarak sonuç elde edilir.

EKG sinyalinde periyodik QRS dalga biçiminin üretilmesi: EKG sinyalinin bir periyodunun üçgensel ve sinüsoidal dalga biçiminin karışımından oluştuğu görülür. Buna göre EKG sinyalinin QRS dalga kısmı üçgen dalga biçimi ile temsil edilebilir. Ayrıca P, T ve U dalgaları da üçgen dalga biçimi ile temsil edilebilir. Bu dalga kısımlarının her birinin üretilip yan yana eklenmesiyle EKG sinyali oluşmaktadır. Yani EKG sinyali P, Q, R, S, T ve U dalgalarının birleşmesiyle oluşan sinyaldir [16].



Şekil 3.4. QRS dalga biçiminin üretilmesi [16]

$$\begin{aligned} f(x) &= (-bax/l) + a, \quad 0 < x < (1/b) \\ &= (bax/l) + a, \quad (-1/b) < x < 0 \end{aligned} \quad (3.7)$$

Yukarıda Eşitlik 3.7’de verilen $f(x)$ fonksiyonunun Şekil 3.4’deki grafiğe göre QRS dalga biçimini temsil etmek üzere yazılmasının ardından, bu $f(x)$ fonksiyonu Eşitlik 3.3’de bulunan Fourier serisinde yerine yazılıp hesaplamalar yapıldıktan sonra aşağıdaki a_n , b_n katsayıları ve a_0 bulunur [16].

$$\begin{aligned} a_0 &= (1/l) \int_T f(x) dx \\ &= (a/b) * (2-b) \end{aligned} \quad (3.8)$$

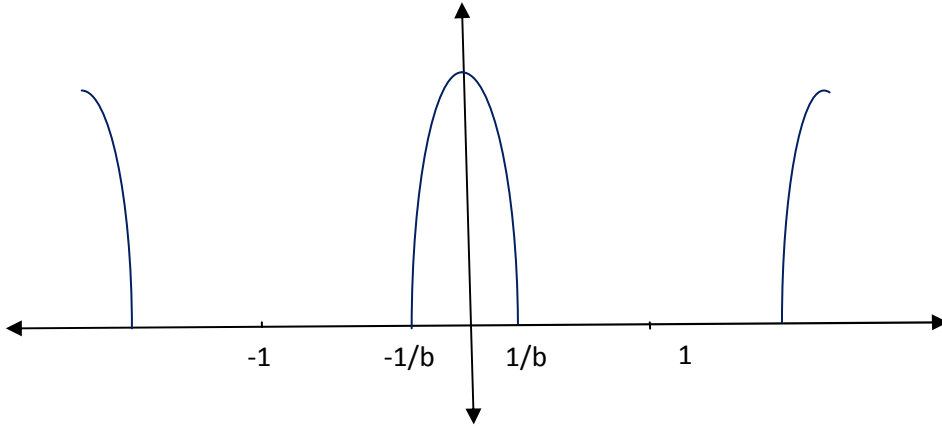
$$\begin{aligned} a_n &= (1/l) \int_T f(x) \cos(n\pi x/l) dx \\ &= (2ba / (n^2 \pi^2)) * (1 - \cos(n\pi / b)) \end{aligned} \quad (3.9)$$

$$\begin{aligned} b_n &= (1/l) \int_T f(x) \sin(n\pi x/l) dx \\ &= 0 \end{aligned} \quad (3.10)$$

Fourier açılımı yapılacak fonksiyon, periyot içinde simetrik ise b_n katsayıları (3.10) sıfır olacağından açılımda sinüslü terimler bulunmamaktadır [16,43]. Sonuçta $f(x)$ fonksiyonunun Fourier Seri açılımı aşağıdaki gibi elde edilir [16].

$$f(x) = (a_0 / 2) + \sum_{n=1}^{\infty} a_n \cos(n\pi x / l) \quad (3.11)$$

EKG sinyalinde periyodik P dalgasının üretilmesi:



Şekil 3.5. P dalgasının üretilmesi [16]

$$f(x) = \cos((\pi bx) / (2l)) , \quad (-1/b) < x < (1/b) \quad (3.11)$$

Yukarıda Eşitlik 3.11'de verilen $f(x)$ fonksiyonu Şekil 3.5'deki grafiğe göre P dalgasını temsil etmek için yazıldıktan sonra, bu $f(x)$ fonksiyonu Eşitlik 3.3'de bulunan Fourier serisinde yerine yazılıp hesaplamalar yapılarak a_n , b_n katsayıları ve a_0 bulunur [16].

$$a_0 = (1/l) \int_T \cos((\pi bx) / (2l)) dx \quad (3.12)$$

$$= (a / (2b))(2 - b) \quad (3.13)$$

$$a_n = (1/l) \int_T \cos((\pi bx) / (2l)) \cos(n\pi x / l) dx \quad (3.14)$$

$$= (((2ba) / (i^2 \pi^2))(1 - \cos(n\pi / b))) \cos((n\pi x) / l) \quad (3.15)$$

$$b_n = (1/l) \int_T \cos((\pi bx) / (2l)) \sin(n\pi x / l) dx \quad (3.16)$$

$$= 0 \quad (3.17)$$

Fourier açılımı yapılacak fonksiyon, periyot içinde simetrik olduğunda b_n katsayıları (3.17) sıfır olacağından açılımda sinüslü terimler bulunmamaktadır [16,43]. Böylece P dalgasının da Fourier açılımı Eşitlik 3.18'deki gibi elde edilmiştir [16].

$$f(x) = (a_0 / 2) + \sum_{n=1}^{\infty} a_n \cos(n\pi x / l) \quad (3.18)$$

Bu aşamada EKG sinyalinin bileşenlerini oluştururken kullanılan matematiksel model verilmiştir. Bu matematiksel model bir diğer aşamada benzetim yazılımı davranışlarını MATLAB programı yardımıyla kodlarken kullanılmaktadır.

3.2.1.2. EKG sinyalinin benzetim yazılımı davranışlarının MATLAB ile kodlanması

İlk aşamada EKG sinyalinin matematiksel modeli verildikten sonra tasarım basamakları MATLAB Web Figure tabanlı EKG benzetim yazılımı davranışlarının MATLAB programı yardımıyla tanımlanması aşamasıyla devam etmektedir. Sistem davranışları MATLAB komutları ile kodlandıktan sonra MATLAB fonksiyonu/fonksiyonları haline getirilerek “.m” dosyası/dosyaları olarak kaydedilmektedir. Uygulamada MATLAB'ın Web Figure özelliği kullanılacağı için fonksiyon içerisinde bir veya daha fazla figure tanımlaması yapılmalıdır. Şekil 3.6'da örnek bir tanımlama görülmektedir.

MATLAB Web Figure tabanlı EKG benzetimi tasarımında bulunan EKG dalga biçimini oluşturan bileşenler ve her bir EKG ritim türü ayrı “.m” dosyalarında fonksiyon olarak tanımlanmaktadır. Burada tanımlanan fonksiyonlarla ilgili kodlar Ek A'da verilmiştir.


```

1 function webgraf=ecg_cis(heart_rate,a_pwave,d_pwave,t_pwave,a_qrswave,d_qrswave,a_twave,d_twave,signal1,signal2)
2
3     x=0.01:0.01:6;
4     ...
5     ...
6     %pwav output
7     pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
8
9     %qwav output
10    qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
11
12    %qrswav output
13    qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
14    ...
15    ...
16
17    %ecg output
18    ecg=pwav+qrswav+twav+swav+qwav+uwav;
19
20    f=figure;
21    set(f,'Visible','off');
22    plot(x,ecg);
23    grid on;
24    webgraf=webfigure(f);
25    close(f);
26    end

```

Şekil 3.6. Sistem davranışlarının MATLAB ile kodlanması

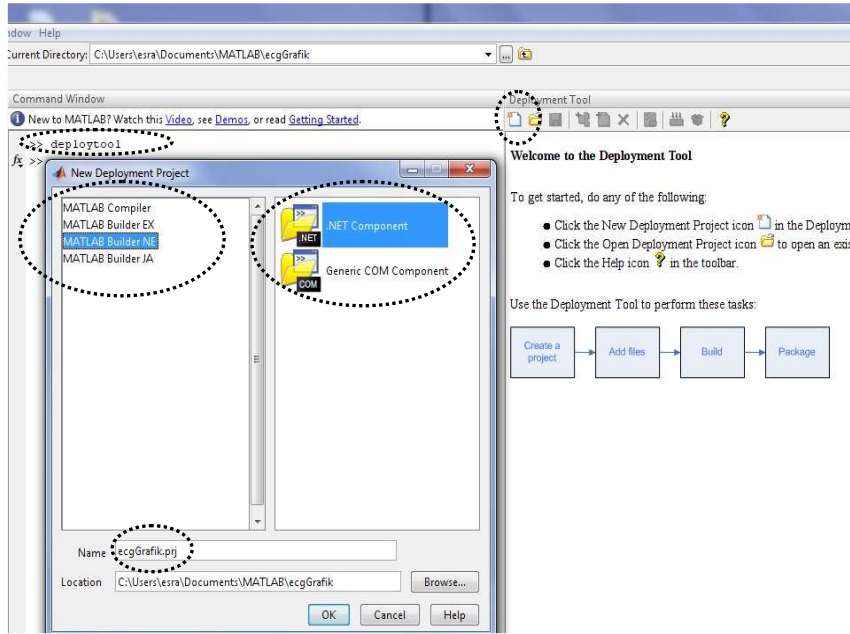
3.2.2. MATLAB Builder NE ve MATLAB Web Figure yardımıyla .NET bileşenlerinin oluşturulması

MATLAB Builder NE, MATLAB Compiler ürününün bir ek ürünüdür [38]. MATLAB Builder NE, MATLAB fonksiyonlarına .NET programcılarının CLS (Common Language Specification)-compliant dilleri olan C#, VB.Net, ASP.NET veya C++ programlama dilleriyle ulaşmalarına olanak sağlayan bir derleyicidir [38,39]. MATLAB Builder NE, MATLAB kodlarını derleyerek MATLAB fonksiyonlarını .NET metodlarına dönüştürür. MATLAB Builder NE ile oluşturulan her .NET bileşeni bir veya daha fazla sınıf (class) içerebilir. Her sınıf ise derleme işleminden önce eklenen MATLAB fonksiyonları için arayüz oluşturur [38].

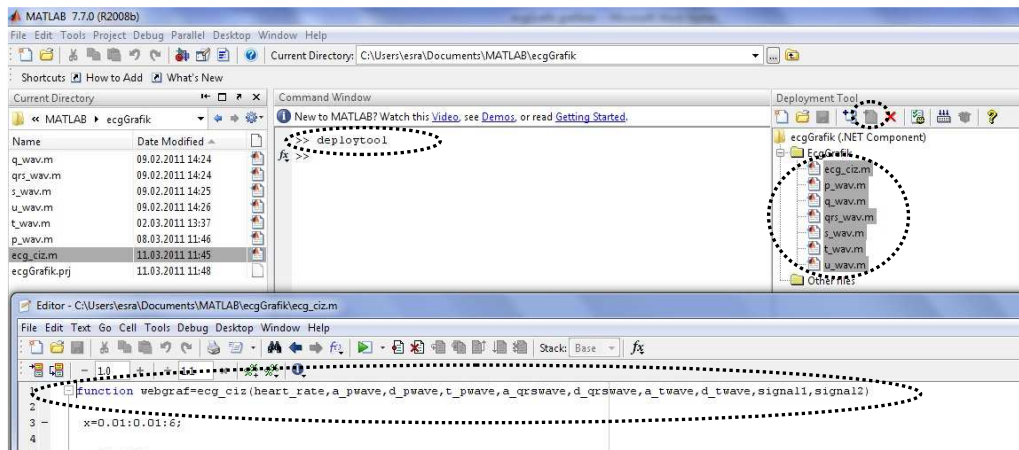
MATLAB Builder NE veri dönüşümü, veri sıralama ve dizi biçimlendirme olanakları sunarak, derlenmiş kod çağrıldığında MATLAB fonksiyonları ve özelliklerinin esnek biçimde kullanılmasını sağlamaktadır. Ayrıca MATLAB Builder NE MATLAB veri tiplerinin desteklenmesi için MWArray.dll dosyasında tanımlı olan MWArray veri dönüşüm sınıflarını bulundurmaktadır. Kullanılan veri ve dizileri MATLAB'a uygun biçime dönüştürmek ve MATLAB'tan gelen verileri hazırlanan programda kullanabilmek için MWArray.dll dosyasını uygulamaya referans olarak eklemek gerekmektedir [38].

MATLAB Builder NE’de bulunan Web Figure özelliği sayesinde ise MATLAB figürler bir web sitesinde gösterilebilir ve bu figürler üzerinde görsel olarak işlem yapabilme olanağı sağlanır. Bu görsel işlemler sayesinde MATLAB figürlerinde yakınlaştırma/uzaklaştırma (zoom), kaydırma ve döndürme gibi işlemler yapılır. Bu özellik ile son kullanıcılar, MATLAB programına ve diğer araçlara gerek kalmadan grafiksel uygulamaları herhangi bir yerden sadece web tarayıcı yardımıyla internet üzerinden gerçekleştirebilirler [38,39].

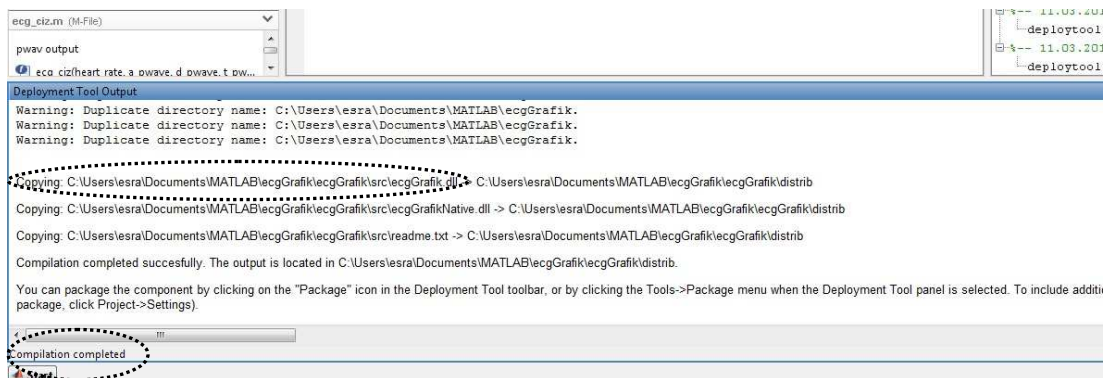
MATLAB Builder NE derleyicisini kullanarak “.dll” uzantılı .NET bileşeni oluşturulurken ilk tasarım basamağında tanımlanan MATLAB fonksiyonu (.m dosyası) hazır olarak bu basamakta kullanılır. Şekil 3.7’de görüldüğü gibi bu işlem için öncelikle MATLAB komut satırına “deploytool” yazılarak derleme aracı (Deployment Tool) çalıştırılır. Ekranı gelen pencereden yeni bir proje açılır. Oluşturulan bileşenin platformunu ve adını belirledikten sonra yeni bir proje oluşturulur. Bu belirleme işleminde “.NET” ile ilgili seçenekler seçilmesine dikkat edilmelidir. Proje oluştururken oluşturulan bileşenin ve içindeki sınıfın isimlerinin belirlenmesi de bir diğer dikkat edilmesi gereken noktadır. Bu belirlenen isimler daha sonra web arayüz tasarımında aynen kullanılacaktır. Şekil 3.8’de görüldüğü gibi daha sonra oluşturulan “.m” dosyaları dosya ekle (add file) ikonuna tıklanarak sınıf (class)’a eklenir. Böylece oluşturulan bileşen içerisindeki fonksiyon Visual Studio programında aynen kullanılabilir. Derlenecek olan “.m” dosyaları seçildikten sonra MATLAB Builder NE derleyicisi kullanılarak derleme ikonu yardımıyla derleme (build) işlemi gerçekleştirilir. Şekil 3.9’da görüldüğü üzere derleme işlemi sonlandıktan sonra, tasarım basamağının son aşaması olan Visual Studio programında kullanılacak “.dll” dosyası bir diğer deyişle “.NET” bileşeni (component) elde edilmektedir.



Şekil 3.7. Deployment Tool aracının çağırılması ve yeni proje oluşturulması işlemi



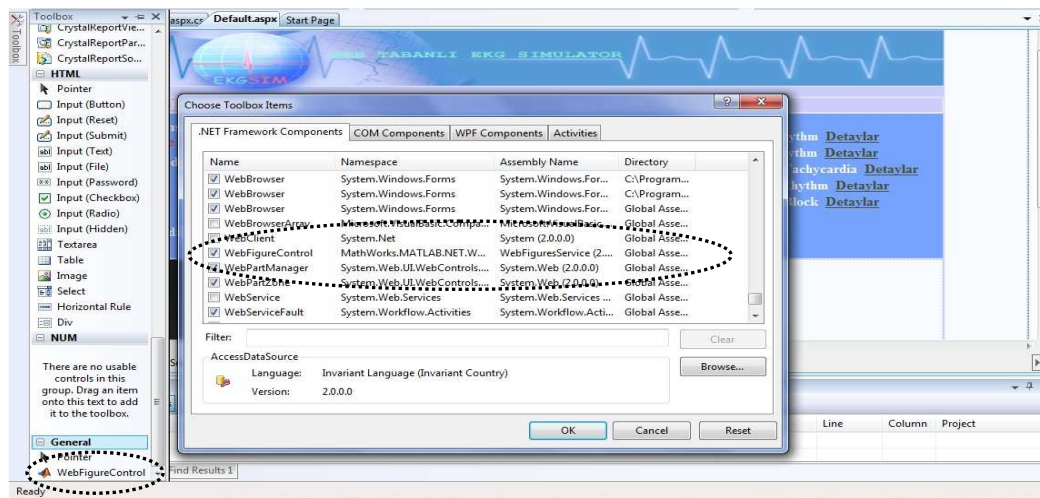
Şekil 3.8. MATLAB Builder NE ile .m dosyasının sınıfa eklenmesi işlemi



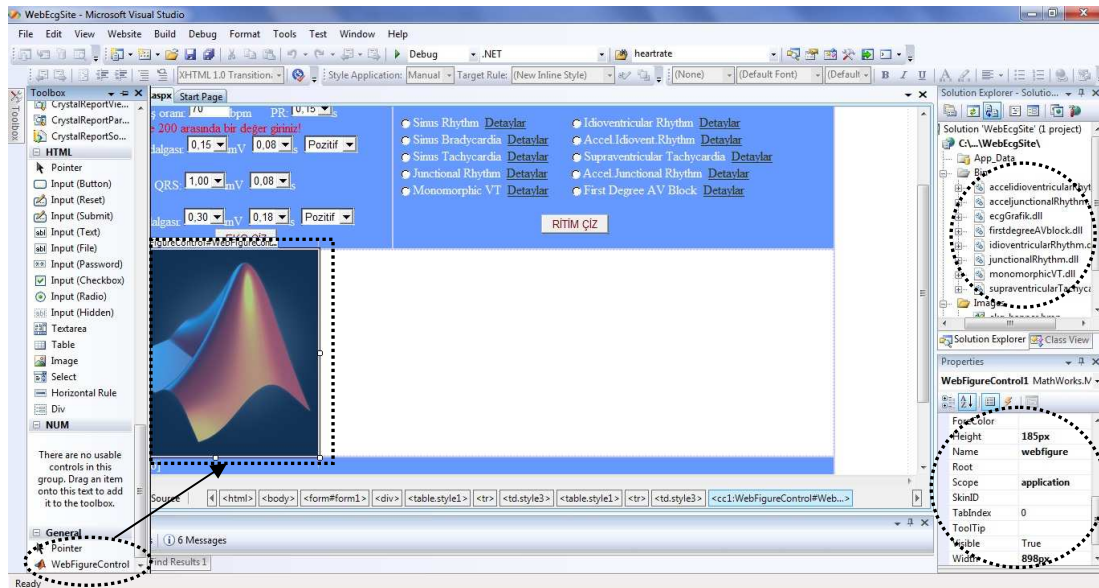
Şekil 3.9. MATLAB Builder NE ile .NET bileşeni (.dll) oluşturma işlemi

3.2.3. Tasarımda ASP.NET platformunun kullanımı

Tasarım basamağının son aşaması benzetim yazılımının web sayfası tasarımı bölümüdür. Web sayfasını tasarlamak amacıyla Microsoft Visual Studio 2008 platformu kullanılmıştır [41,42] ve programlama dili olarak C# seçilmiştir. Öncelikle yeni bir ASP.NET web sitesi oluşturulmalıdır. Web sitesinde olması gereken buton, check box, text box, dropdownlist gibi komponentlerin eklendikten sonra özelliklerinin ayarlanması ile web sitesinin görsel tasarımı yapılmış olur. Daha sonra .NET için oluşturulmuş MATLAB Web Figure arac (tool)'ını kullanmak için araç kutusu (toolbox) sekmesindeki "General" kısmına sağ tıklanarak "Choose Items" seçeneğinden "Browse" butonuna tıklanarak dosya yolundan v2.0 klasörüne ulaşılır. v2.0 klasöründen WebFigureService.dll dosyası seçildikten sonra WebFigureControl bulunarak işaretlenir (Şekil 3.10) ve işlem tamamlanır. Bu işlemler sonucunda işaretli ikonun araç kutusuna eklendiği görülmelidir (Şekil 3.10). İkon fare yardımıyla çalışma alanına sürüklenerek bırakıldıktan sonra Web Figure Control aracı üzerinde yerleşim ve boyutlandırma gibi işlemler yapılabilmektedir. Çalışma alanına eklenen bu aracın özellikler (properties) sekmesinde isim (name) kısmına uygun bir isim verildikten sonra "scope" kısmında "application" seçeneği seçilmelidir. Bu işlemlerden sonra daha önceki basamakta MATLAB programında MATLAB Builder NE ile oluşturulan ".NET" bileşenleri (.dll) referans olarak gösterilmelidir. Şekil 3.11'de oluşturulan .NET bileşeninin referans olarak eklenmesi ve Web Figure Control aracının çalışma ortamına eklenmesi işlemi görülmektedir.

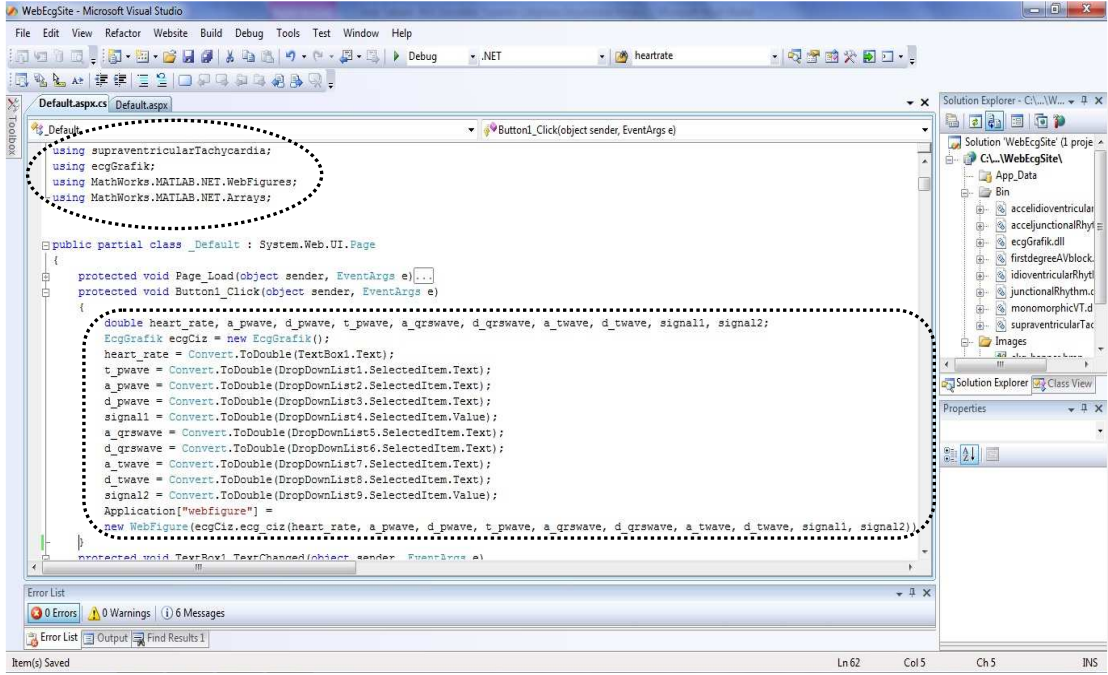


Şekil 3.10. Web Figure Control aracının eklenmesi işlemi



Şekil 3.11. Oluşturulan .NET bileşeni (.dll) ve Web Figure Control aracının çalışma ortamına eklenmesi işlemi

Görsel tasarım bittikten sonra son adım olarak çalışma alanında sağ tıklanarak “View Code” seçeneğinden kod yazılacak yer olan “.cs” uzantılı dosya açılır. Bu dosyada ilk olarak referansa eklenen .NET bileşenleri “using” ifadesiyle çağrılır. Daha sonra web sayfasına eklenen buton, radio button gibi görsel elemanlar için gerekli olan C# kodları yazılmalıdır. Kod ekranında tanımlanan fonksiyon ve değişkenler “.NET” bileşenlerinde tanımlanan fonksiyon ve değişkenlerdir. Bu .NET bileşeninde tanımlanan “ecg_ciz, junctional_rhythm, monomorphic_VT, idioventricular_rhythm, acceleratedidioventrivular_rhythm, supraventricular_tachycardia, acceleratedjunctional_rhytm, firstdegree_AV_block” gibi fonksiyonlar kod ekranında çağrılarak kullanılır. Örneğin Şekil 3.12’de “ecgCiz” değişkenine ait olarak çağrılan “ecg_ciz” fonksiyonu oluşturulan .NET bileşeninde tanımlanan fonksiyonlardandır. “EcgGrafik” sınıfı MATLAB’ta oluşturulan bileşen içindeki sınıf olup “ecgCiz” değişkeni bu sınıf için oluşturulan bir değişkendir. Burada dikkat edilmesi gereken nokta tasarım ekranında Web Figure Control aracına verilen isim ile kodlarda tırnak içinde (webfigure) yazılan isim aynı olmalıdır. Bu fonksiyonlar, oluşturulan .NET bileşenleri ve Web Figure Control aracının kullanılabilmesi için gerekli kodların bir kısmı Şekil 3.12’de görülmektedir. Bu kodların tamamı Ek B’de verilmiştir.



Şekil 3.12. Oluşturulan .NET bileşeni ve Web Figure Control aracının kullanılabilmesi için gerekli kodlar

BÖLÜM 4. TASARLANAN MATLAB WEB FIGURE TABANLI BENZETİM YAZILIMININ ÖZELLİKLERİ VE KULLANIMI

Bu bölümde MATLAB Builder NE ve MATLAB Web Figure kullanımı ile tasarlanan web tabanlı EKG benzetim yazılımının özellikleri ve kullanım örnekleri sunulmaktadır. Tasarlanan MATLAB Web Figure tabanlı EKG benzetim yazılımı Şekil 4.1’de görüldüğü gibi 2 kısımdan oluşmaktadır. Şekil 4.1’de Matlab Builder NE ve Web Figure kullanılarak tasarlanan web tabanlı benzetim yazılımının ilk ekrana nasıl geldiği görülmektedir. Arayüzde “EKG Temel Özellikleri” alanı olarak adlandırılan ilk kısımdan kalp atış oranı, P, Q, R, S ve T gibi bir EKG sinyalini oluşturan bileşenlerin genlik, yön ve periyot değerleri gibi temel değerleri ayarlanarak “EKG ÇİZ” butonu ile istenilen EKG sinyali çizdirilebilmektedir. Bu şekilde kullanıcı EKG dalga biçiminin özelliklerini analiz edebilmektedir. Şekil 4.2’de EKG temel bileşenlerinin özelliklerinin değiştirilmesi ile oluşan bir EKG sinyal örneği görülmektedir.



Şekil 4.1. Tasarlanan MATLAB Web Figure tabanlı EKG benzetimi



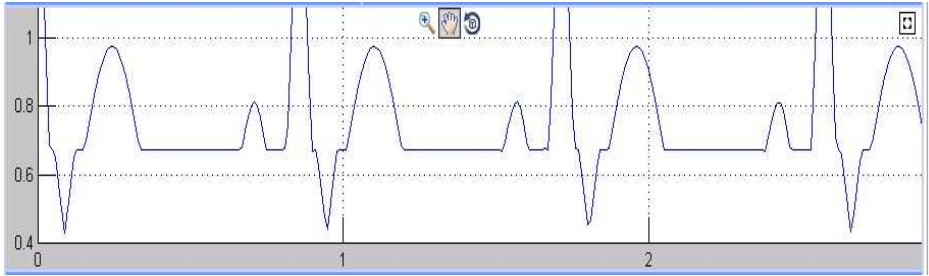
Şekil 4.2. EKG dalga biçiminin temel özellikleri değiştirildiğinde oluşan sinyal

Web tabanlı benzetim yazılımının ikinci kısmı ise “EKG Ritim Türleri” olarak adlandırılan bölümdür. Bu bölümde biri normal ritim diğerleri 9 farklı kalp ritim bozukluğu türü olmak üzere toplam 10 adet EKG ritim türü bulunmaktadır. Kullanıcı istediği herhangi bir ritim türünü seçerek “RİTİM ÇİZ” butonuna tıkladığında ilgili ritim türünün grafiği ekranda gösterilmektedir. Ayrıca detaylar linkine tıkladığında seçilen ritim türünün özellikleri hakkında bilgi verilmektedir. Şekil 4.3’de Sinüs Ritim grafiği ve detayları görülmektedir.

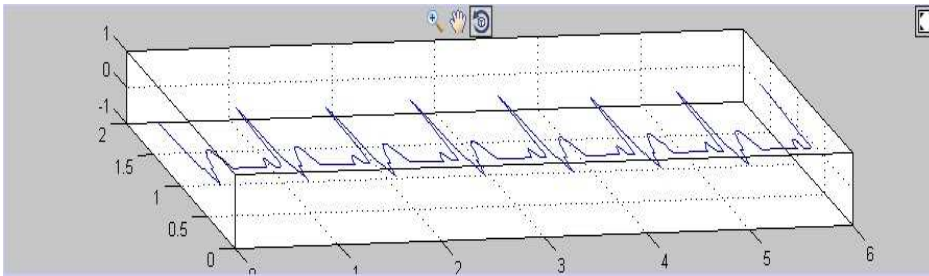
Web tabanlı benzetim yazılımı ile çizdirilen herhangi bir EKG grafiğinin üzerine fareyle gelindiğinde 3 boyutlu (3-D) görüntü, yaklaştırma/uzaklaştırma ve hareket ettirme gibi MATLAB Web Figure’ün özellikleri grafik üzerinde görülür. Büyüteç ikonu şekli yakınlaştırmak, el ikonu şeklin yerini değiştirmek ve döndürme ikonu ise şeklin bakış açısını değiştirmek için kullanılabilir. Şekil 4.4’de Sinüs Ritim grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği gösterilmektedir. Kullanıcılar bu özelliklerle yakınlaştırma ve uzaklaştırma derecesini ayarlayarak grafikleri kolaylıkla analiz edebilirler. Şekil 4.5’de Sinüs Ritim grafiği üzerinde Web Figure’ün 3-boyutlu görünüm özelliği görülmektedir.



Şekil 4.3. Sinüs Ritim grafiği ve detayları



Şekil 4.4. Sinüs Ritim grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği

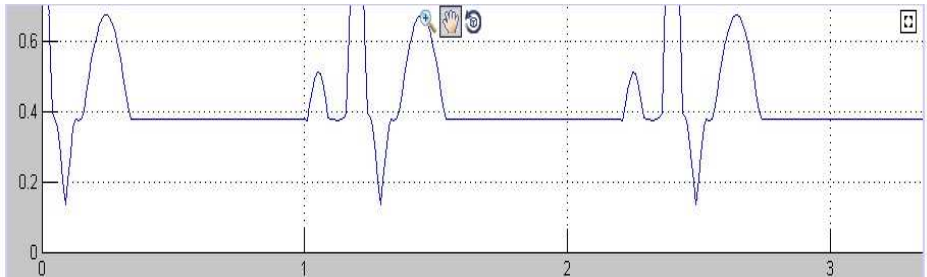


Şekil 4.5. Sinüs Ritim grafiği üzerinde Web Figure'ün 3-D görüntüm özelliği

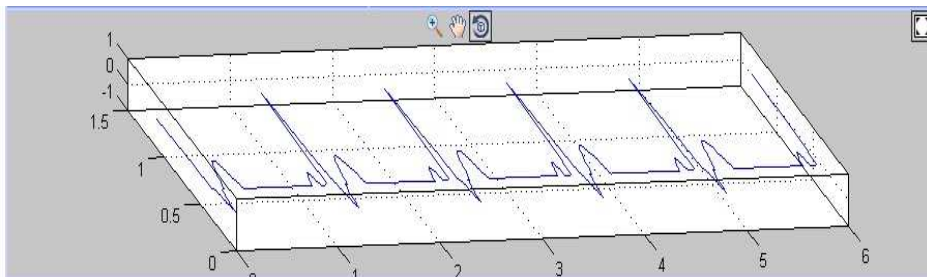
Şekil 4.6’da Sinüs Bradikardi grafiği ve detayları görülmektedir. Şekil 4.7’de Sinüs Bradikardi grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği, Şekil 4.8’de ise Sinüs Bradikardi grafiği üzerinde Web Figure’ün 3-D görünüm özelliği gösterilmektedir.



Şekil 4.6. Sinüs Bradikardi grafiği ve detayları

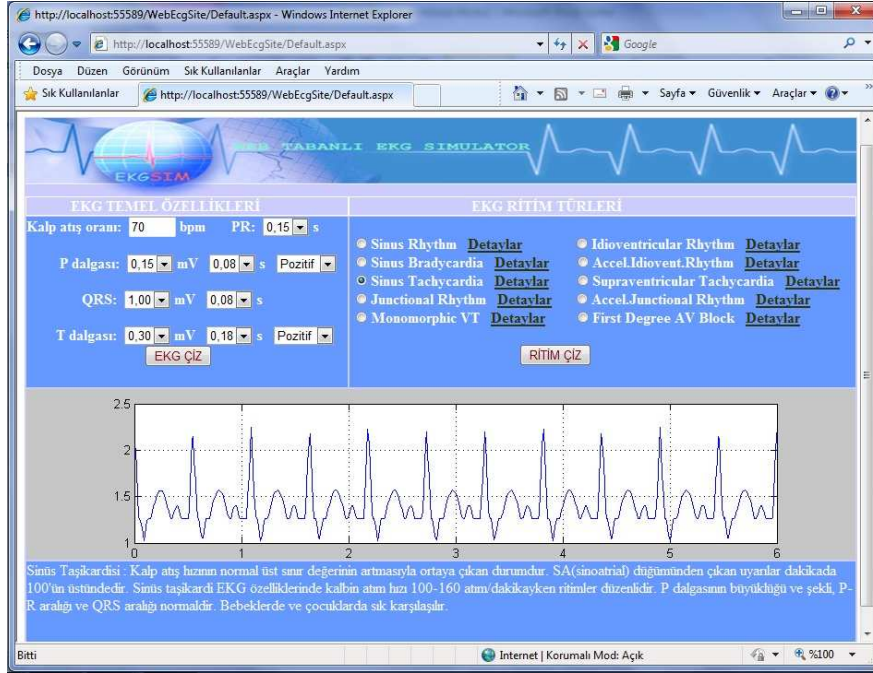


Şekil 4.7. Sinüs Bradikardi grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği

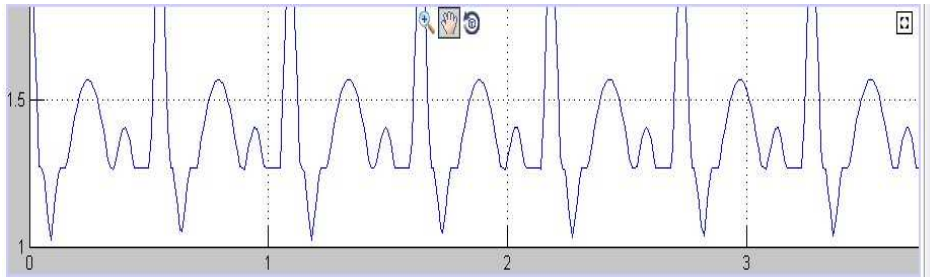


Şekil 4.8. Sinüs Bradikardi grafiği üzerinde Web Figure’ün 3-D görünüm özelliği

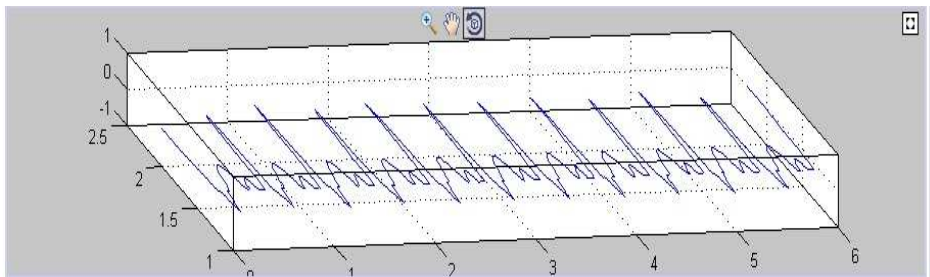
Aşağıda Sinüs Taşikardi grafiği ve detayları verilmektedir (Şekil 4.9). Şekil 4.10'da Sinüs Taşikardi grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği ve Şekil 4.11'de ise Sinüs Taşikardi grafiği üzerinde Web Figure'ün 3-D görünüm özelliği görülmektedir.



Şekil 4.9. Sinüs Taşikardi grafiği ve detayları



Şekil 4.10. Sinüs Taşikardi grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği

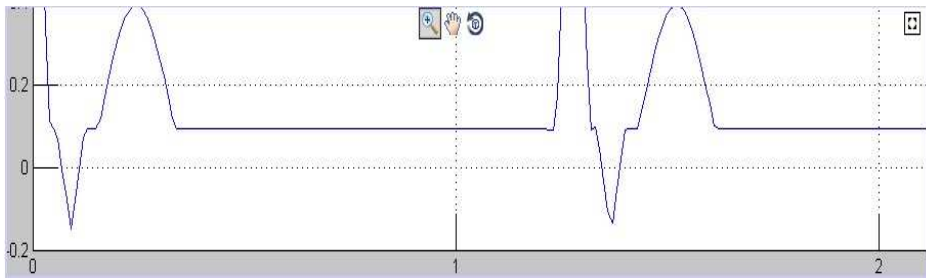


Şekil 4.11. Sinüs Taşikardi grafiği üzerinde Web Figure'ün 3-D görünüm özelliği

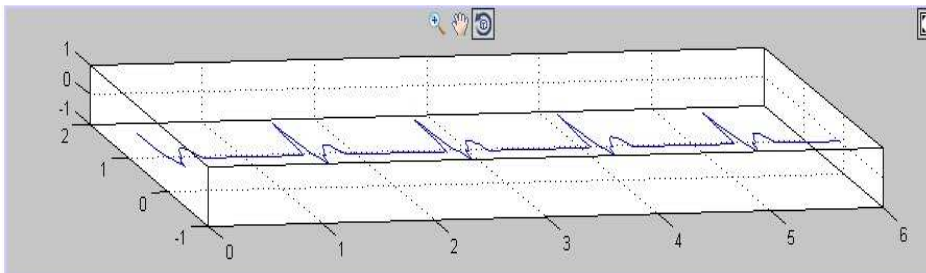
Şekil 4.12’de Junctional Ritim grafiği ve bu grafik hakkında detaylar verilmekte, Şekil 4.13’de Junctional Ritim grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği ve daha sonra Şekil 4.14’de Junctional Ritim grafiği üzerinde Web Figure’ün 3-D görünüm özelliği sunulmaktadır.



Şekil 4.12. Junctional Ritim grafiği ve detayları

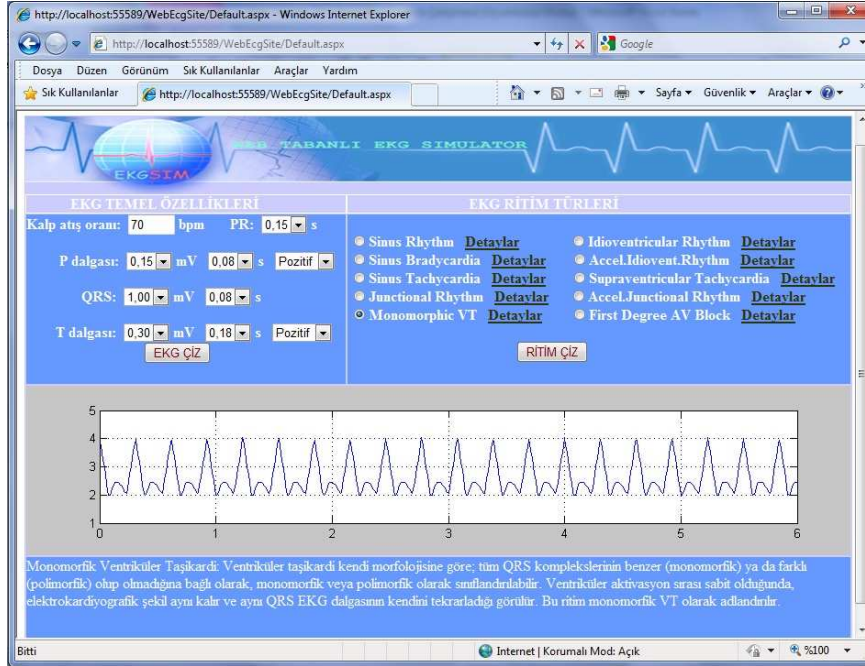


Şekil 4.13. Junctional Ritim grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği

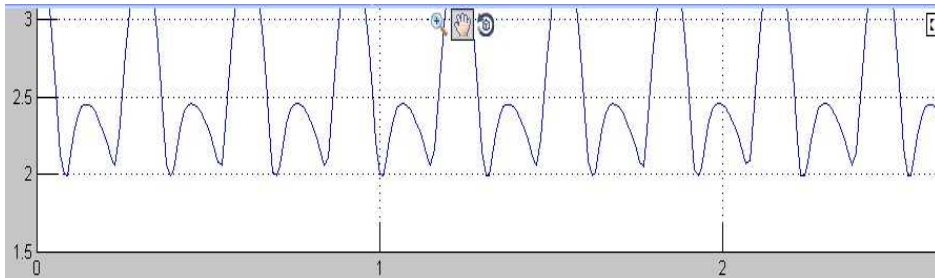


Şekil 4.14. Junctional Ritim grafiği üzerinde Web Figure’ün 3-D görünüm özelliği

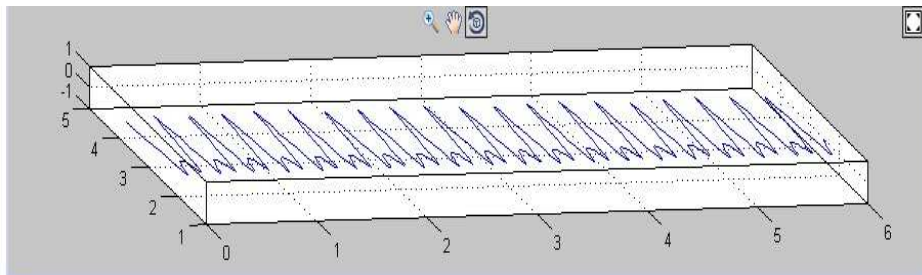
Monomorfik VT grafiđi ve detayları verilmektedir (Şekil 4.15). Şekil 4.16'da Monomorfik VT grafiđi üzerinde Web Figure'ün zoom ve hareket ettirme özelliđi ve Şekil 4.17'de ise Monomorfik VT grafiđi üzerinde Web Figure'ün 3-D görünüm özelliđi gösterilmektedir.



Şekil 4.15. Monomorfik VT grafiđi ve detayları



Şekil 4.16. Monomorfik VT grafiđi üzerinde Web Figure'ün zoom ve hareket ettirme özelliđi



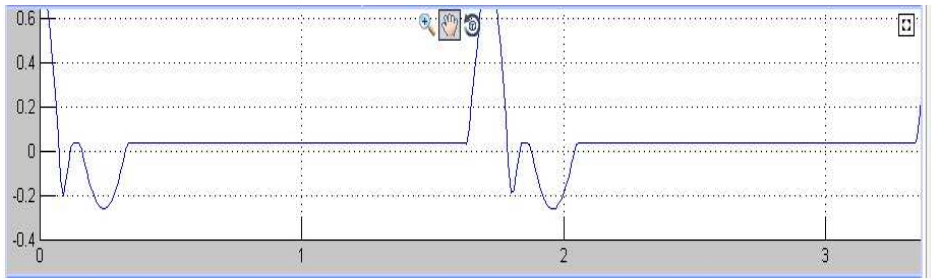
Şekil 4.17. Monomorfik VT grafiđi üzerinde Web Figure'ün 3-D görünüm özelliđi

Aşağıdaki Şekil 4.18’de İdioventiküler Ritim grafiği ve detayları verilmektedir.

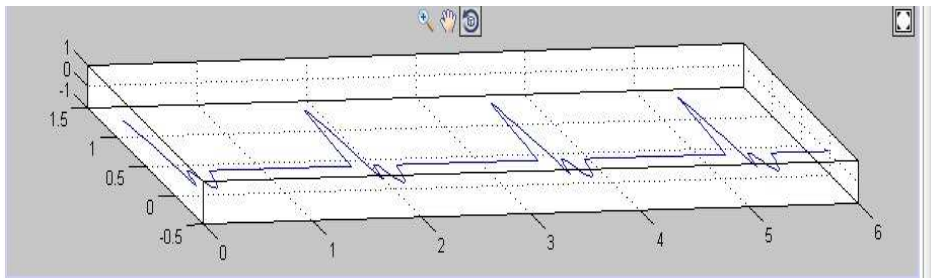


Şekil 4.18. İdioventiküler Ritim grafiği ve detayları

Şekil 4.19’da İdioventiküler Ritim grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği ve Şekil 4.20’de ise İdioventiküler Ritim grafiği üzerinde Web Figure’ün 3-D görünüm özelliği sunulmaktadır.



Şekil 4.19. İdioventiküler Ritim grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği

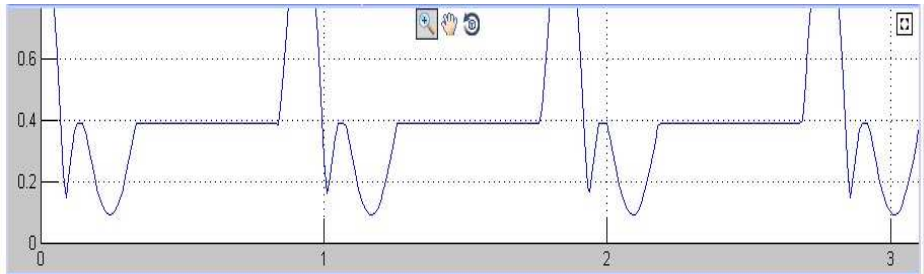


Şekil 4.20. İdioventiküler Ritim grafiği üzerinde Web Figure’ün 3-D görünüm özelliği

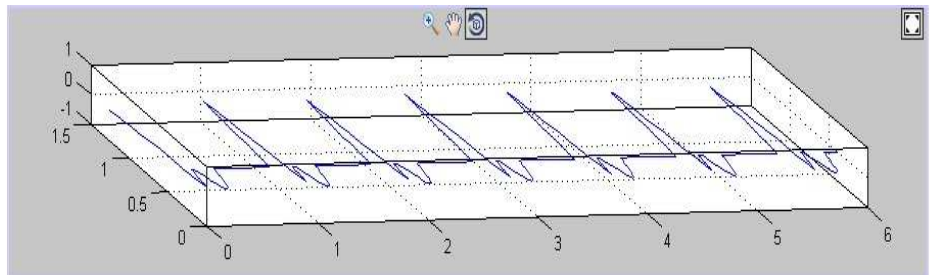
Şekil 4.21’de Akselere İdioventriküler Ritim grafiği ve detayları verilmekte, aynı grafik üzerinde Şekil 4.22’de Web Figure’ün zoom ve hareket ettirme özelliği ve ayrıca Şekil 4.23’de Web Figure’ün 3-D görünüm özelliği gösterilmektedir.



Şekil 4.21. Akselere İdioventriküler Ritim grafiği ve detayları

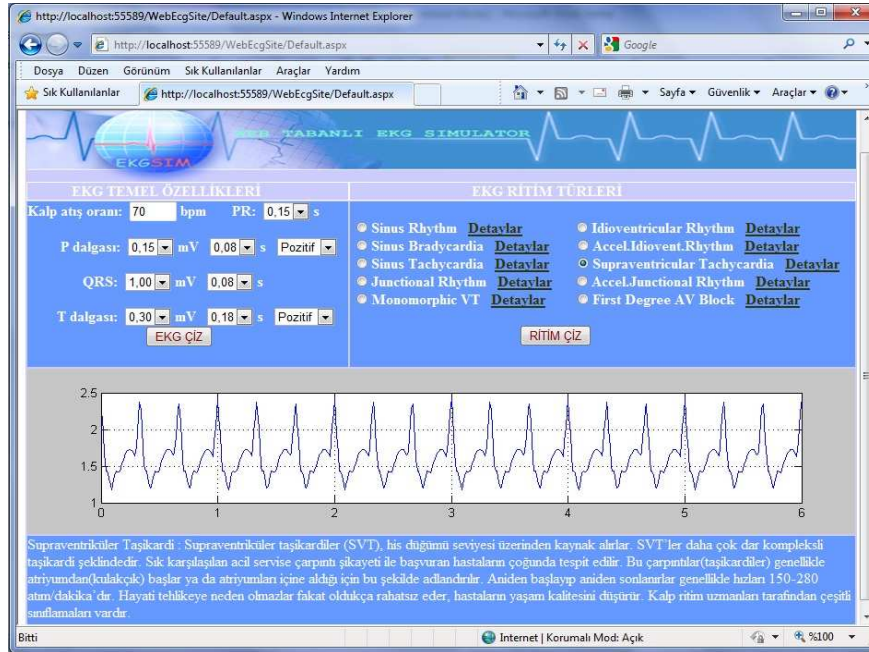


Şekil 4.22. Akselere İdioventriküler Ritim grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği



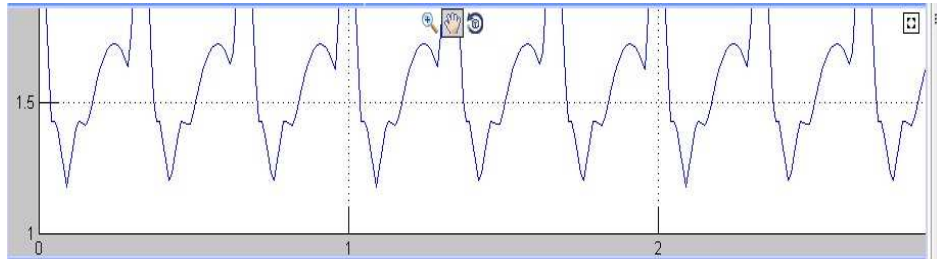
Şekil 4.23. Akselere İdioventriküler Ritim grafiği üzerinde Web Figure’ün 3-D görünüm özelliği

Şekil 4.24’de Supraventriküler Taşikardi grafiği ve bu grafik hakkında bilgiler verilmektedir.

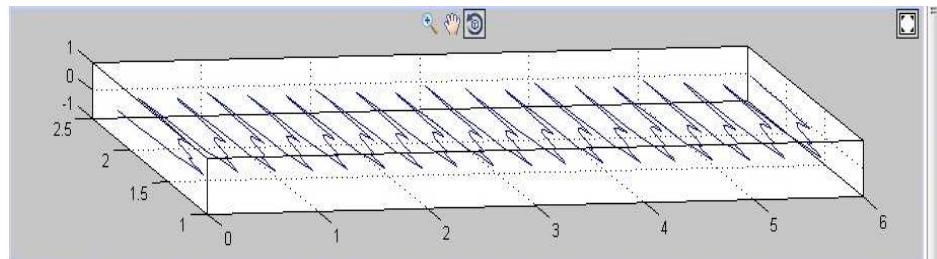


Şekil 4.24. Supraventriküler Taşikardi grafiği ve detayları

Şekil 4.25’de Supraventriküler Taşikardi grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği ve Şekil 4.26’da Supraventriküler Taşikardi grafiği üzerinde Web Figure’ün 3-D görünüm özelliği görülmektedir.



Şekil 4.25. Supraventriküler Taşikardi grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği

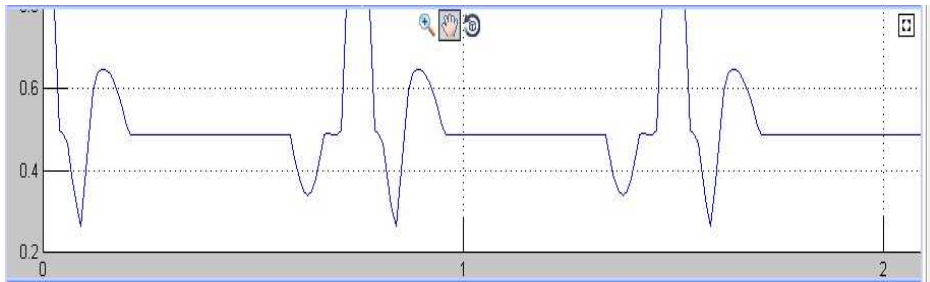


Şekil 4.26. Supraventriküler Taşikardi grafiği üzerinde Web Figure’ün 3-D görünüm özelliği

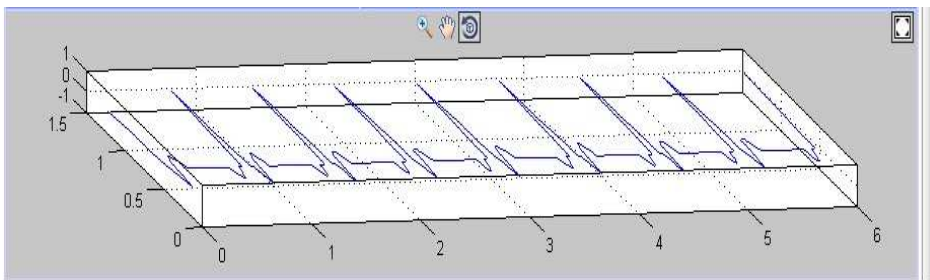
Şekil 4.27’de Akselere Junctional Ritim grafiği ve bu grafik hakkında bilgi verilmekte, aynı grafik üzerinde Şekil 4.28’de Web Figure’ün zoom ve hareket ettirme özelliği ve Şekil 4.29’da ise Web Figure’ün 3-D görünüm özelliği görülmektedir.



Şekil 4.27. Akselere Junctional Ritim grafiği ve detayları



Şekil 4.28. Akselere Junctional Ritim grafiği üzerinde Web Figure’ün zoom ve hareket ettirme özelliği

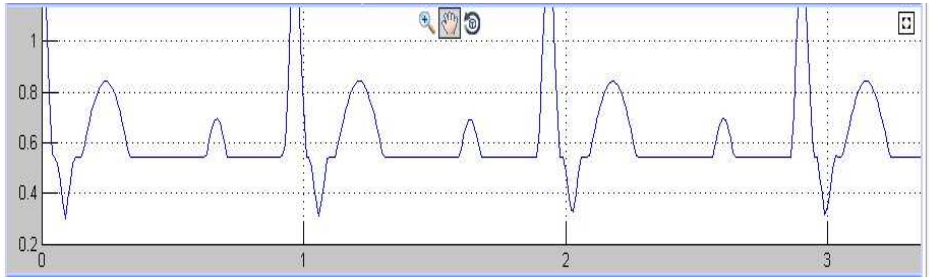


Şekil 4.29. Akselere junctional Ritim grafiği üzerinde Web Figure’ün 3-D görünüm özelliği

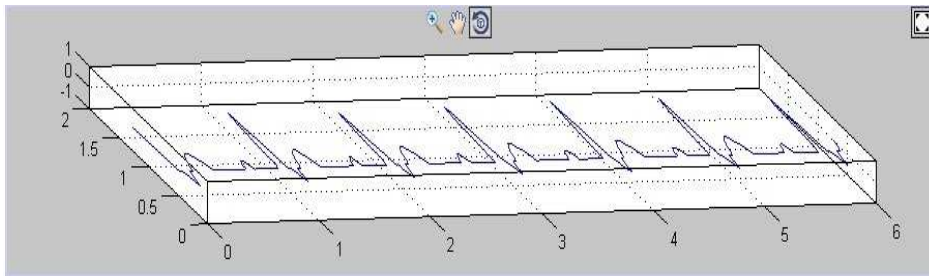
Şekil 4.30'da Birinci Derece AV Blok grafiği ve detayları verilmekte, aynı grafik üzerinde Şekil 4.31'de Web Figure'ün zoom ve hareket ettirme özelliği ve ayrıca Şekil 4.32'de Web Figure'ün 3-D görünüm özelliği sunulmaktadır.



Şekil 4.30. Birinci Derece AV Blok grafiği ve detayları



Şekil 4.31. Birinci Derece AV Blok grafiği üzerinde Web Figure'ün zoom ve hareket ettirme özelliği



Şekil 4.32. Birinci Derece AV Blok grafiği üzerinde Web Figure'ün 3-D görünüm özelliği

Geliştirilen web tabanlı EKG benzetim yazılımında biri normal ritim diğerleri 9 farklı kalp ritim bozukluğu türü olmak üzere toplam 10 adet EKG ritim türü ile ilgili grafik örnekleri yukarıda sunulmuştur. Çizdirilen tüm grafikler üzerinde hareket ettirme, yaklaştırma ve 3 boyutlu görünüm özelliklerinin kullanılabildiği örneklerde görülmektedir.

BÖLÜM 5. SONUÇLAR

Bu tez çalışmasında, temel EKG dalga biçimleri ve kalp ritim bozukluklarını üretebilen eğitim amaçlı bir web tabanlı benzetim tasarımı sunulmuştur. Bu çalışmada önerilen web tabanlı benzetim yazılımının literatürdeki çalışmalardan en büyük farkı MATLAB Builder NE ve MATLAB Web Figure araçlarını kullanılarak bir web uygulaması gerçekleştirilmiş olmasıdır. Tasarlanan bu benzetim yazılımının 3 boyutlu görünüm, yaklaştırma ve hareket ettirme gibi görsel özelliklere izin vermesi EKG sinyallerinin analizini oldukça kolaylaştırmakta ve bu alanda bilgi birikimi az olan kişilerin eğitimine katkı sağlamaktadır. Bu benzetim yazılımının bir diğer önemli avantajı ise kullanıcıların herhangi bir programa gerek kalmadan sadece bir web tarayıcısı kullanarak tasarlanan benzetim yazılımına erişebilmeleridir. Bu sayede tasarlanan benzetim yazılımında MATLAB programının analiz özellikleri kullanılırken MATLAB programının kurulu olmasına gerek yoktur. Çalışmanın bir diğer önemli özelliği ise MATLAB Builder NE ve MATLAB Web Figure araçlarının web tabanlı biyomedikal uygulamalarında kullanılabilirliğini gösteren ve bunun için bir metodoloji sunan ilk çalışma olmasıdır. Bu sebeple önerilen bu çalışma web tabanlı birçok biyomedikal uygulama için emsal kabul edilebilir.

KAYNAKLAR

- [1] CANER, C., ENGİN, M., ENGİN, E., The programmable ECG simulator, J Med Syst, 32, pp. 355-359, 2008.
- [2] ACHARYA, U. R., BHAT, P.S., IYENGAR, S.S., ROO, A., DUA S., Classification of heart rate data using artificial neural network and fuzzy equivalence relation, Pattern Recognition, 36, pp. 61-68, 2003.
- [3] EL-KHANSA, L., NAÏT-ALÏ, A., Parametrical modeling of a premature ventricular contraction ECG beat: Comparison with the normal case, Computers in Biology and Medicine, 37, pp. 1-7, 2007.
- [4] ÜBEYLİ, E. D., Biyomedikal işaretlerin sınıflandırılmasında otomatik teşhis sistemlerinin başarımı, Gazi Üniv. Müh. Mim. Fak. Der., 22, 3, sf. 461-469, 2007.
- [5] AYDIN, S., Biyomedikal mühendisliği eğitiminde bilgisayar tabanlı görsel eğitim uygulamaları, Elektrik mühendisliği, 431, sf. 113-114, 2007.
- [6] DEMİR, B. E., YORULMAZ, F., GÜLER, İ., Mikrodenetleyici denetimli EKG simülatörü, IEEE, sf. 1-47, 2010.
- [7] KAUR, G., Design and development of dual channel ECG simulator and peak detector, Master of Engineering, Thapar Institute of Engineering & Technology Deemed university, 2006.
- [8] KARAYALÇIN, S., YÜKSEKKAYA, M., YAZGI, S., EKG simülatörü, Başkent Üniversitesi, Biyomedikal Mühendisliği Bölümü, IEEE, 2010.
- [9] ERBİL, N., Uzaktan kontrol edilebilen kalp cihazına ait yazılım geliştirilmesi, Yüksek Lisans Tezi, Selçuk Üniversitesi, Elektronik ve Bilgisayar Sistemleri Ana Bilim Dalı, 2007.
- [10] KOCATÜRK, M., Design of a wi-fi based electrocardiography monitoring system, Master of Science, Boğaziçi University, Biomedical Engineering, 2007.
- [11] KABALCI, E., PC tabanlı kablosuz EKG biyotelemetri sistemi tasarımı ve yapımı, Yüksek Lisans Tezi, Gazi Üniversitesi, Elektronik ve Bilgisayar Eğitimi Bölümü, 2006.

- [12] SHİN, W., CHA, Y. D., YOON, G., ECG/PPG integer signal processing for a ubiquitous health monitoring system, *J Med Syst*, 34, 891-898, 2010.
- [13] TAN, T. H., CHANG, C. S., HUANG, Y. F., CHEN, Y.F., LEE,C., Development of a portable linux-based ECG measurement and monitoring system, *J Med Syst*, DOI 10.1007/s10916-009-9392-4.
- [14] OEFİNGER, M. B., MARK, R. G., A web-based tool for visualization and collaborative annotation of physiological databases, *Computers in Cardiology*, 32, pp. 163–165, 2005.
- [15] NİZAM, A., EKG işaretlerini gerçek zamanda bilgisayara aktaran ve işleyen arayüz programı tasarımı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2000.
- [16] KARTHİK, R., ECG simulation using Matlab, Anna University, Guindy, Chennai
<http://www.mathworks.com/matlabcentral/fileexchange/10858>, 10.12.2010.
- [17] The Six Second ECG(Cardiac Rhythm Simulator),
http://www.skillstat.com/ecg_sim_demo.html, 12.01.2011.
- [18] ECG Simulator,
<http://simecg.sourceforge.net/>, 15.01.2011.
- [19] NILSSON, M., BOLINDER, G., HELD, C., JOHANSSON B. L., FORS,U., ÖSTERGREN, J., Evaluation of a web-based ECG-interpretation programme for undergraduate medical students, *BMC Medical Education*, 8, 25, 2008, doi:10.1186/1472-6920-8-25.
- [20] KAYA, M., Elektrokardiyogram işaretlerinin sıkıştırılması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2006.
- [21] DOĞAN, B., Parçacık sürü optimizasyonuna dayalı yeni bir aritmi sınıflama yöntemi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2009.
- [22] NİZAM, A., Karınca koloni optimizasyonuna dayalı yeni bir aritmi sınıflama tekniği, Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2008.
- [23] KARADAĞ, A., Dalgacık ağlarıyla elektrokardiyografik aritmilerin sınıflandırılması, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2009.
- [24] AKTÜRK, T. B., PDA tabanlı gerçek zamanlı EKG görüntüleme sistemi, Yüksek Lisans Tezi, Afyon Kocatepe Üniversitesi, Bilgisayar Ana Bilim Dalı, 2009.
- [25] DOĞAN, H. D., Hemşirelerin acil kalp hastalıklarında görülen EKG

bulgularını tanıyabilme ve uygun tedavi yaklaşımlarını değerlendirebilme düzeylerinin tespiti, Yüksek Lisans Tezi, Afyonkarahisar Kocatepe Üniversitesi, Sağlık Bilimleri Enstitüsü, 2006.

- [26] <http://www.kardiyo.net/kitap/aritmiler2.shtml>, 01.04.2011.
- [27] http://turkaritmi.com/sinus_tasikardisi-51-tamam.html, 26.03.2011.
- [28] http://168.144.121.167/_atuder.org/5atks/eken.pdf, 15.04.2011.
- [29] <http://www.tkd.org.tr/SunuMerkezi/view.asp?slide=LI25N6TT5>, 05.04.2011.
- [30] <http://www.sunummerkezi.org/?s=5F21292A2D5D22395A>, 18.04.2011.
- [31] http://www.itf-kardiyoloji.org/dersler_slayt/fehmi_mercanoglu/%C4%B1v/23.4.2008-ders-EKG%20IV-fm.pdf, 20.04.2011.
- [32] <http://www.scribd.com/doc/49579000/49/Accelerated-Junctional-Rhythm-Junctional-Tachycardia>, 20.04.2011.
- [33] <http://lokman.cu.edu.tr/anestezi/anestezinot/ekg&.htm>, 20.03.2011.
- [34] ÇAKMAK, A., ATAR, İ., ÖZİN, B., Çarpıntı yakınması ile başvuran, yavaş hızlı geniş QRS ritmi bulunan genç bir hastaya yaklaşım, Türk Aritmi, Pacemaker ve Elektrofizyoloji Dergisi, 7, 1, 2009.
- [35] http://www.wikidoc.org/index.php/Accelerated_idioventricular_rhythm, 10.04.2011.
- [36] <http://www.metealpaslan.com/ekg/vair.htm>, 10.04.2011.
- [37] <http://emedicine.medscape.com/article/159075-overview>, 22.02.2011.
- [38] Matlab Builder NE 3 User's Guide, The Mathworks Inc., Ekim 2008.
- [39] Matlab builder NE 4.0 for Microsoft .NET Framework, 2011.
<http://www.mathworks.com/products/netbuilder/>, 20.02.2011.
- [40] ÇETİN A. E., Herkes için MATLAB 7, ALFA yayınları.
- [41] GÖBEKÇİN, T., Herkes için C# ile ASP.NET 2.0, ALFA yayınları.
- [42] DEMİRKOL, Z., C# ile ASP.NET 3.5, KODLAB yayınları.
- [43] www.akademi.itu.edu.tr/ayhanb/DosyaGetir/.../Fourier%20Serileri.pdf, 05.04.2011.
- [44] http://tr.wikipedia.org/wiki/Fourier_serileri, 05.04.2011.

EKLER

Ek A

MATLAB’da “.m” dosyalarında tanımlanan fonksiyonların kodları

ecg_ciz.m dosyası:

```
function webgraf=ecg_ciz(heart_rate,a_pwave,d_pwave,t_pwave,  
a_qrswave, d_qrswave,a_twave,d_twave,signal1,signal2)  
x=0.01:0.01:6;  
li=30/heart_rate;  
a_pwav=a_pwave;  
d_pwav=d_pwave;  
t_pwav=t_pwave;  
a_qwav=0.025;  
d_qwav=0.066;  
t_qwav=0.166;  
a_qrswav=a_qrswave;  
d_qrswav=d_qrswave;  
a_swav=0.25;  
d_swav=0.066;  
t_swav=0.09;  
a_twav=a_twave;  
d_twav=d_twave;  
t_twav=0.2;
```

```

a_uwav=0.035;
d_uwav=0.0476;
t_uwav=0.433;
%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure;
set(f,'Visible','off');
plot(x,ecg);
grid on;
webgraf=webfigure(f);
close(f);
end

```

p_wav.m dosyası:

```

function [pwav]=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1)
l=li;
a=a_pwav;
x=x+t_pwav;
b=(2*l)/d_pwav;

```

```

n=100;
p1=1/l;
p2=0;
for i = 1:n
    if (signal1==1)
        harm1=(((sin((pi/(2*b))*(b-(2*i))))/(b-
(2*i))+sin((pi/(2*b))*(b+(2*i))))/(b+(2*i)))*(2/pi))*signal1*cos((i*pi*x)/l);
        p2=p2+harm1;
    elseif(signal1==-1)
        harm1=(((sin((pi/(2*b))*(b-(2*i))))/(b-
(2*i))+sin((pi/(2*b))*(b+(2*i))))/(b+(2*i)))*(2/pi))*signal1*cos((i*pi*x)/l);
        p2=p2+harm1;
    end
end
pwav1=p1+p2;
pwav=a*pwav1;

```

q_wav.m dosyası:

```

function [qwav]=q_wav(x,a_qwav,d_qwav,t_qwav,li)
l=li;
x=x+t_qwav;
a=a_qwav;
b=(2*l)/d_qwav;
n=100;
q1=(a/(2*b))*(2-b);
q2=0;
for i = 1:n
    harm5=(((2*b*a)/(i*i*pi*pi))*(1-cos((i*pi)/b)))*cos((i*pi*x)/l);
    q2=q2+harm5;
end
qwav=-1*(q1+q2);

```

qrs_wav.m dosyası:

```

function [qrswav]=qrs_wav(x,a_qrswav,d_qrswav,li)
l=li;
a=a_qrswav;
b=(2*l)/d_qrswav;
n=100;
qrs1=(a/(2*b))*(2-b);
qrs2=0;
for i = 1:n
    harm=((2*b*a)/(i*i*pi*pi))*(1-cos((i*pi)/b))*cos((i*pi*x)/l);
    qrs2=qrs2+harm;
end
qrswav=qrs1+qrs2;

```

s_wav.m dosyası:

```

function [swav]=s_wav(x,a_swav,d_swav,t_swav,li)
l=li;
x=x-t_swav;
a=a_swav;
b=(2*l)/d_swav;
n=100;
s1=(a/(2*b))*(2-b);
s2=0;
for i = 1:n
    harm3=((2*b*a)/(i*i*pi*pi))*(1-cos((i*pi)/b))*cos((i*pi*x)/l);
    s2=s2+harm3;
end
swav=-1*(s1+s2);

```

t_wav.m dosyası:

```

function [twav]=t_wav(x,a_twav,d_twav,t_twav,li,signal2)
l=li;
a=a_twav;
x=x-t_twav-0.045;
b=(2*1)/d_twav;
n=100;
t1=1/l;
t2=0;
for i = 1:n
    if (signal2==1)
        harm2=(((sin((pi/(2*b))*(b-(2*i)))))/(b-
(2*i))+sin((pi/(2*b))*(b+(2*i))))/(b+(2*i)))*(2/pi)*signal2*cos((i*pi*x)/l);
        t2=t2+harm2;
    elseif(signal2==-1)
        harm2=(((sin((pi/(2*b))*(b-(2*i)))))/(b-
(2*i))+sin((pi/(2*b))*(b+(2*i))))/(b+(2*i)))*(2/pi)*signal2*cos((i*pi*x)/l);
        t2=t2+harm2;
    end
end
twav1=t1+t2;
twav=a*twav1;

```

u_wav.m dosyası:

```

function [uwav]=u_wav(x,a_uwav,d_uwav,t_uwav,li)
l=li;
a=a_uwav
x=x-t_uwav;
b=(2*1)/d_uwav;
n=100;
u1=1/l

```

```

u2=0
for i = 1:n
    harm4=(((sin((pi/(2*b))*(b-(2*i)))))/(b-
(2*i))+sin((pi/(2*b))*(b+(2*i))))/(b+(2*i)))*(2/pi))*cos((i*pi*x)/l);
    u2=u2+harm4;
end
uwav1=u1+u2;
uwav=a*uwav1;

```

junctional_rhythm.m dosyası:

```

function webfig=junctional_rhythm(heart_rate,signal1,signal2)
x=0.01:0.01:6;
%heartrate=47;
li=30/heart_rate;
a_pwav=0.15;
d_pwav=0.08;
t_pwav=0.15;
a_qwav=0.025;
d_qwav=0.066;
t_qwav=0.166;
a_qrswav=1.0;
d_qrswav=0.08;
a_swav=0.25;
d_swav=0.066;
t_swav=0.09;
a_twav=0.30;
d_twav=0.18;
t_twav=0.2;
a_uwav=0.035;
d_uwav=0.0476;
t_uwav=0.433;

```

```

%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
pwav=0;
%signal1=1;
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
qwav=0;
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
%signal2=1;
%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure(1);
set(f,'Visible','off');
plot(x,ecg);
grid on;
webfig=webfigure(f);
close(f);
end

```

acceleratedjunctional_rhythm.m dosyası:

```

function webfig=acceleratedjunctional_rhythm(heart_rate,signal1,signal2)
x=0.01:0.01:6;
%heart_rate=80;

```

```
li=30/heart_rate;
a_pwav=0.15;
d_pwav=0.08;
t_pwav=0.12;
a_qwav=0.025;
d_qwav=0.066;
t_qwav=0.166;
a_qrswav=1.0;
d_qrswav=0.08;
a_swav=0.25;
d_swav=0.066;
t_swav=0.09;
a_twav=0.16;
d_twav=0.12;
t_twav=0.10;
a_uwav=0.035;
d_uwav=0.0476;
t_uwav=0.433;
%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
%signal1=-1;
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
qwav=0;
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
%signal2=1;
%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
```



```
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure(1);
set(f,'Visible','off');
plot(x,ecg);
grid on;
webfig=webfigure(f);
close(f);
end
```

idioventricular_rhythm.m dosyası:

```
function webfig=idioventricular_rhythm(heart_rate,signal1,signal2)
x=0.01:0.01:6;
%heart_rate=35;
li=30/heart_rate;
a_pwav=0.15;
d_pwav=0.08;
t_pwav=0.15;
a_qwav=0.025;
d_qwav=0.066;
t_qwav=0.166;
a_qrswav=1.0;
d_qrswav=0.16;
a_swav=0.25;
d_swav=0.066;
t_swav=0.09;
a_twav=0.30;
d_twav=0.18;
t_twav=0.2;
a_uwav=0.035;
d_uwav=0.0476;
```

```

t_uwav=0.433;
%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
pwav=0;
%signal1=1;
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
qwav=0;
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
%signal2=-1;
%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure(1);
set(f,'Visible','off');
plot(x,ecg);
grid on;
webfig=webfigure(f);
close(f);
end

```

acceleratedidioventricular_rhythm.m dosyası:

```

function webfig=acceleratedidioventricular_rhythm(heart_rate,signal1,signal2)
x=0.01:0.01:6;
%heart_rate=65;

```

```
li=30/heart_rate;
    a_pwav=0.15;
    d_pwav=0.08;
    t_pwav=0.15;
    a_qwav=0.025;
    d_qwav=0.066;
    t_qwav=0.166;
    a_qrswav=1.0;
    d_qrswav=0.16;
    a_swav=0.25;
    d_swav=0.066;
    t_swav=0.09;
    a_twav=0.30;
    d_twav=0.18;
    t_twav=0.2;
    a_uwav=0.035;
    d_uwav=0.0476;
    t_uwav=0.433;
%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
pwav=0;
%signal1=1;
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
qwav=0;
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
%signal2=-1;
```

```

%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure(1);
set(f,'Visible','off');
plot(x,ecg);
grid on;
webfig=webfigure(f);
close(f);
end

```

firstdegree_AV_block.m dosyası:

```

function webfig=firstdegree_AV_block(heart_rate,signal1,signal2)
x=0.01:0.01:6;
%heart_rate=62;
li=30/heart_rate;
    a_pwav=0.15;
    d_pwav=0.08;
    t_pwav=0.3;
    a_qwav=0.025;
    d_qwav=0.066;
    t_qwav=0.166;
    a_qrswav=1.0;
    d_qrswav=0.08;
    a_swav=0.25;
    d_swav=0.066;
    t_swav=0.09;
    a_twav=0.30;
    d_twav=0.18;
    t_twav=0.2;

```

```

a_uwav=0.035;
d_uwav=0.0476;
t_uwav=0.433;
%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
%signal1=1;
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
qwav=0;
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
%signal2=1;
%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure(1);
set(f,'Visible','off');
plot(x,ecg);
grid on;
webfig=webfigure(f);
close(f);
end

```

monomorphic_VT.m dosyası:

```
function webfig=monomorphic_VT(heart_rate,signal1,signal2)
```

```
x=0.01:0.01:6;
%heart_rate=195;
  li=30/heart_rate;
  a_pwav=0;
  d_pwav=0.01;
  t_pwav=0.08;
  a_qwav=0.025;
  d_qwav=0.066;
  t_qwav=0.166;
  a_qrswav=2.00;
  d_qrswav=0.15;
  a_swav=0.25;
  d_swav=0.066;
  t_swav=0.09;
  a_twav=0.40;
  d_twav=0.16;
  t_twav=0.10;
  a_uwav=0.035;
  d_uwav=0.0476;
  t_uwav=0.433;
%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
pwav=0;
%signal1=1;
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
qwav=0;
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
```

```

%signal2=-1;
%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure(1);
set(f,'Visible','off');
plot(x,ecg);
grid on;
webfig=webfigure(f);
close(f);
end

```

supraventricular_tachycardia.m dosyası:

```

function webfig=supraventricular_tachycardia(heart_rate,signal1,signal2)
x=0.01:0.01:6;
%heart_rate=180;
li=30/heart_rate;
a_pwav=0.15;
d_pwav=0.08;
t_pwav=0.15;
a_qwav=0.025;
d_qwav=0.066;
t_qwav=0.166;
a_qrswav=1.0;
d_qrswav=0.08;
a_swav=0.25;
d_swav=0.066;
t_swav=0.09;
a_twav=0.30;
d_twav=0.18;

```

```
t_twav=0.2;
a_uwav=0.035;
d_uwav=0.0476;
t_uwav=0.433;
%pwav output
pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li,signal1);
pwav=0;
%signal1=1;
%qwav output
qwav=q_wav(x,a_qwav,d_qwav,t_qwav,li);
%qrswav output
qrswav=qrs_wav(x,a_qrswav,d_qrswav,li);
%swav output
swav=s_wav(x,a_swav,d_swav,t_swav,li);
%twav output
twav=t_wav(x,a_twav,d_twav,t_twav,li,signal2);
%signal2=1;
%uwav output
uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);
uwav=0;
%ecg output
ecg=pwav+qrswav+twav+swav+qwav+uwav;
f=figure(1);
set(f,'Visible','off');
plot(x,ecg);
grid on;
webfig=webfigure(f);
close(f);
end
```


Ek B

Web arayüzündeki ASP.NET (C#) kodları

Default.aspx.cs dosyası:

```
using System;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using accelidioventricularRhythm;
using acceljunctionalRhythm;
using firstdegreeAVblock;
using idioventricularRhythm;
using junctionalRhythm;
using monomorphicVT;
using supraventricularTachycardia;
using ecgGrafik;
using MathWorks.MATLAB.NET.WebFigures;
using MathWorks.MATLAB.NET.Arrays;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        { double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
```

```

EcgGrafik temelegc = new EcgGrafik();
heart_rate = Convert.ToDouble(TextBox1.Text);
heart_rate = 70;
a_pwave = 0.15;
d_pwave = 0.08;
t_pwave = 0.15;
a_qrswave = 1.00;
d_qrswave = 0.08;
a_twave = 0.30;
d_twave = 0.18;
signal1 = 1;
signal2 = 1;
Application["webfigure"] =
    new WebFigure(temelegc.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));
RadioButton1.Checked = true; }

```

```

protected void Button1_Click(object sender, EventArgs e)
{ double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
    EcgGrafik ecgCiz = new EcgGrafik();
    heart_rate = Convert.ToDouble(TextBox1.Text);
    t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
    a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
    d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
    signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
    a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
    d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
    a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
    d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
    signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
    Application["webfigure"] =

```

```
new WebFigure(ecgCiz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));
```

Label16.Text = "Elektrokardiyogram (EKG) üzerinde, kalbin elektriksel potansiyeline ilişkin yön, genlik ve zaman içinde değişim gibi bilgiler bulunmaktadır. Normal EKG, izoelektrik hat üzerinde sıralanan ve P, Q, R, S, T adları verilen dalgalardan oluşur. Q, R ve S dalgaları EKG işaretinin en belirgin ve kolay gözlemlenebilir kısmı olan QRS kompleksini oluştururlar."; }

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
```

```
{ double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
```

```
EcgGrafik ecggraf = new EcgGrafik();
```

```
heart_rate = Convert.ToDouble(TextBox1.Text);
```

```
t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
```

```
a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
```

```
d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
```

```
signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
```

```
a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
```

```
d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
```

```
a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
```

```
d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
```

```
signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
```

```
Application["webfigure"] =
```

```
new WebFigure(ecggraf.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }
```

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
```

```
{ double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
```

```
EcgGrafik tPwawciz = new EcgGrafik();
```

```
heart_rate = Convert.ToDouble(TextBox1.Text);
```

```
t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
```

```
a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
```

```

d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
Application["webfigure"] =
    new WebFigure(tPwaviciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
{ double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
    EcgGrafik aPwaviciz = new EcgGrafik();
    heart_rate = Convert.ToDouble(TextBox1.Text);
    t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
    a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
    d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
    signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
    a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
    d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
    a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
    d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
    signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
    Application["webfigure"] =
        new WebFigure(aPwaviciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

protected void DropDownList3_SelectedIndexChanged(object sender, EventArgs e)
{
    double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;

```

```

EcgGrafik dPwaviciz = new EcgGrafik();
heart_rate = Convert.ToDouble(TextBox1.Text);
t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
Application["webfigure"] =
    new WebFigure(dPwaviciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

protected void DropDownList4_SelectedIndexChanged(object sender, EventArgs e)
{ double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
EcgGrafik signal1ciz = new EcgGrafik();
heart_rate = Convert.ToDouble(TextBox1.Text);
t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
Application["webfigure"] =
    new WebFigure(signal1ciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

```

```

protected void DropDownList5_SelectedIndexChanged(object sender, EventArgs e)
{
    double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
    EcgGrafik aQrswavciz = new EcgGrafik();
    heart_rate = Convert.ToDouble(TextBox1.Text);
    t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
    a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
    d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
    signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
    a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
    d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
    a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
    d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
    signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
    Application["webfigure"] =
        new WebFigure(aQrswavciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

```

```

protected void DropDownList6_SelectedIndexChanged(object sender, EventArgs e)
{
    double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
    EcgGrafik dQrswavciz = new EcgGrafik();
    heart_rate = Convert.ToDouble(TextBox1.Text);
    t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
    a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
    d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
    signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
    a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
    d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
    a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
    d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
    signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
    Application["webfigure"] =

```

```

        new WebFigure(dQrswavciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

```

```

protected void DropDownList7_SelectedIndexChanged(object sender, EventArgs e)

```

```

    { double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;

```

```

        EcgGrafik aTwavciz = new EcgGrafik();

```

```

        heart_rate = Convert.ToDouble(TextBox1.Text);

```

```

        t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);

```

```

        a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);

```

```

        d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);

```

```

        signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);

```

```

        a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);

```

```

        d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);

```

```

        a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);

```

```

        d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);

```

```

        signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);

```

```

Application["webfigure"] =

```

```

        new WebFigure(aTwavciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

```

```

protected void DropDownList8_SelectedIndexChanged(object sender, EventArgs e)

```

```

    { double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;

```

```

        EcgGrafik dTwavciz = new EcgGrafik();

```

```

        heart_rate = Convert.ToDouble(TextBox1.Text);

```

```

        t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);

```

```

        a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);

```

```

        d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);

```

```

        signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);

```

```

        a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);

```

```

        d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);

```

```

        a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);

```

```

    d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
    signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
Application["webfigure"] =
    new WebFigure(dTwavciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }
protected void DropDownList9_SelectedIndexChanged(object sender, EventArgs e)
{ double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
Ecgrafik signal2ciz = new Ecgrafik();
heart_rate = Convert.ToDouble(TextBox1.Text);
t_pwave = Convert.ToDouble(DropDownList1.SelectedItem.Text);
a_pwave = Convert.ToDouble(DropDownList2.SelectedItem.Text);
d_pwave = Convert.ToDouble(DropDownList3.SelectedItem.Text);
signal1 = Convert.ToDouble(DropDownList4.SelectedItem.Value);
a_qrswave = Convert.ToDouble(DropDownList5.SelectedItem.Text);
d_qrswave = Convert.ToDouble(DropDownList6.SelectedItem.Text);
a_twave = Convert.ToDouble(DropDownList7.SelectedItem.Text);
d_twave = Convert.ToDouble(DropDownList8.SelectedItem.Text);
signal2 = Convert.ToDouble(DropDownList9.SelectedItem.Value);
Application["webfigure"] =
    new WebFigure(signal2ciz.ecg_ciz(heart_rate, a_pwave, d_pwave, t_pwave,
a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

protected void Button2_Click(object sender, EventArgs e)
{
    if (RadioButton1.Checked == true)
    { double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
Ecgrafik rithmciz1 = new Ecgrafik();
heart_rate = 70;
a_pwave = 0.15;
d_pwave = 0.08;
t_pwave = 0.15;

```



```

a_qrswave = 1.00;
d_qrswave = 0.08;
a_twave = 0.30;
d_twave = 0.18;
signal1 = 1;
signal2 = 1;

Application["webfigure"] = new WebFigure(rithmciz1.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton2.Checked == true)
    { double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
    EcgGrafik rithmciz2 = new EcgGrafik();
    heart_rate = 50;
    a_pwave = 0.15;
    d_pwave = 0.08;
    t_pwave = 0.15;
    a_qrswave = 1.00;
    d_qrswave = 0.08;
    a_twave = 0.30;
    d_twave = 0.18;
    signal1 = 1;
    signal2 = 1;

    Application["webfigure"] = new WebFigure(rithmciz2.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton3.Checked == true)
    { double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
    EcgGrafik rithmciz3 = new EcgGrafik();
    heart_rate = 110;
    a_pwave = 0.15;
    d_pwave = 0.08;

```

```

t_pwave = 0.15;
a_qrswave = 1.00;
d_qrswave = 0.08;
a_twave = 0.30;
d_twave = 0.18;
signal1 = 1;
signal2 = 1;
Application["webfigure"] = new WebFigure(rithmciz3.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton4.Checked == true)
    { double heart_rate, signal1, signal2;
      JunctionalRhythm rithmciz4 = new JunctionalRhythm();
      heart_rate = 47;
      signal1 = 1;
      signal2 = 1;
Application["webfigure"] = new WebFigure(rithmciz4.junctional_rhythm(heart_rate,
signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton5.Checked == true)
    { double heart_rate, signal1, signal2;
      MonomorphicVT rithmciz5 = new MonomorphicVT();
      heart_rate = 195;
      signal1 = 1;
      signal2 = 1;
Application["webfigure"] = new WebFigure(rithmciz5.monomorphic_VT(heart_rate,
signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton6.Checked == true)
    { double heart_rate, signal1, signal2;
      IdioventricularRhythm rithmciz6 = new IdioventricularRhythm();
      heart_rate = 35;
      signal1 = 1;

```

```

        signal2 = -1;
Application["webfigure"] = new
WebFigure(rithmciz6.idioventricular_rhythm(heart_rate, signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton7.Checked == true)
{ double heart_rate, signal1, signal2;
    AccelidioventricularRhythm rithmciz7 = new AccelidioventricularRhythm();
    heart_rate = 65;
    signal1 = 1;
    signal2 = -1;
Application["webfigure"] = new
WebFigure(rithmciz7.acceleratedidioventricular_rhythm(heart_rate, signal1,
signal2));
    Label16.Text = " "; }
else if (RadioButton8.Checked == true)
{ double heart_rate, signal1, signal2;
    SupraventricularTachycardia rithmciz8 = new SupraventricularTachycardia();
    heart_rate = 180;
    signal1 = 1;
    signal2 = 1;
Application["webfigure"] = new
WebFigure(rithmciz8.supraventricular_tachycardia(heart_rate, signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton9.Checked == true)
{ double heart_rate, signal1, signal2;
    AcceljunctionalRhythm rithmciz9 = new AcceljunctionalRhythm();
    heart_rate = 80;
    signal1 = -1;
    signal2 = 1;
Application["webfigure"] = new
WebFigure(rithmciz9.acceleratedjunctional_rhythm(heart_rate, signal1, signal2));
    Label16.Text = " "; }
else if (RadioButton10.Checked == true)

```

```

    { double heart_rate, signal1, signal2;
      FirstdegreeAVblock rithmciz10 = new FirstdegreeAVblock();
      heart_rate = 62;
      signal1 = 1;
      signal2 = 1;
Application["webfigure"] = new
WebFigure(rithmciz10.firstdegree_AV_block(heart_rate, signal1, signal2));
      Label16.Text = " "; }
  }
protected void RadioButton1_CheckedChanged(object sender, EventArgs e)
  { double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
  EcgGrafik rithmciz1 = new EcgGrafik();
  heart_rate = 70;
  a_pwave = 0.15;
  d_pwave = 0.08;
  t_pwave = 0.15;
  a_qrswave = 1.00;
  d_qrswave = 0.08;
  a_twave = 0.30;
  d_twave = 0.18;
  signal1 = 1;
  signal2 = 1;
Application["webfigure"] = new WebFigure(rithmciz1.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));}

protected void RadioButton2_CheckedChanged(object sender, EventArgs e)
  { RadioButton1.Checked = false;
  double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;
  EcgGrafik rithmciz2 = new EcgGrafik();
  heart_rate = 50;
  a_pwave = 0.15;

```

```

d_pwave = 0.08;
t_pwave = 0.15;
a_qrswave = 1.00;
d_qrswave = 0.08;
a_twave = 0.30;
d_twave = 0.18;
signal1 = 1;
signal2 = 1;

```

```

Application["webfigure"] = new WebFigure(rithmciz2.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); }

```

```

protected void RadioButton3_CheckedChanged(object sender, EventArgs e)

```

```

{ RadioButton1.Checked = false;
  double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave,
a_twave, d_twave, signal1, signal2;

```

```

  EcgGrafik rithmciz3 = new EcgGrafik();

```

```

  heart_rate = 110;
  a_pwave = 0.15;
  d_pwave = 0.08;
  t_pwave = 0.15;
  a_qrswave = 1.00;
  d_qrswave = 0.08;
  a_twave = 0.30;
  d_twave = 0.18;
  signal1 = 1;
  signal2 = 1;

```

```

Application["webfigure"] = new WebFigure(rithmciz3.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2)); },

```

```

protected void RadioButton4_CheckedChanged(object sender, EventArgs e)

```

```

{
  RadioButton1.Checked = false;
  double heart_rate, signal1, signal2;

```

```

JunctionalRhythm rithmciz4 = new JunctionalRhythm();
heart_rate = 47;
signal1 = 1;
signal2 = 1;
Application["webfigure"] = new WebFigure(rithmciz4.junctional_rhythm(heart_rate,
signal1, signal2)); }

```

```

protected void RadioButton5_CheckedChanged(object sender, EventArgs e)
{ RadioButton1.Checked = false;
  double heart_rate, signal1, signal2;
  MonomorphicVT rithmciz5 = new MonomorphicVT();
  heart_rate = 195;
  signal1 = 1;
  signal2 = 1;
Application["webfigure"] = new WebFigure(rithmciz5.monomorphic_VT(heart_rate,
signal1, signal2)); }

```

```

protected void RadioButton6_CheckedChanged(object sender, EventArgs e)
{ RadioButton1.Checked = false;
  double heart_rate, signal1, signal2;
  IdioventricularRhythm rithmciz6 = new IdioventricularRhythm();
  heart_rate = 35;
  signal1 = 1;
  signal2 = -1;
Application["webfigure"] = new
WebFigure(rithmciz6.idioventricular_rhythm(heart_rate, signal1, signal2)); }

```

```

protected void RadioButton7_CheckedChanged(object sender, EventArgs e)
{ RadioButton1.Checked = false;
  double heart_rate, signal1, signal2;
  AccelidioventricularRhythm rithmciz7 = new AccelidioventricularRhythm();
  heart_rate = 65;
  signal1 = 1;

```

```

    signal2 = -1;
Application["webfigure"] = new
WebFigure(rithmciz7.acceleratedidioventricular_rhythm(heart_rate, signal1,
signal2)); }

protected void RadioButton8_CheckedChanged(object sender, EventArgs e)
{ RadioButton1.Checked = false;
  double heart_rate, signal1, signal2;
  SupraventricularTachycardia rithmciz8 = new SupraventricularTachycardia();
  heart_rate = 180;
  signal1 = 1;
  signal2 = 1;
Application["webfigure"] = new
WebFigure(rithmciz8.supraventricular_tachycardia(heart_rate, signal1, signal2)); }

protected void RadioButton9_CheckedChanged(object sender, EventArgs e)
{ RadioButton1.Checked = false;
  double heart_rate, signal1, signal2;
  AcceljunctionalRhythm rithmciz9 = new AcceljunctionalRhythm();
  heart_rate = 80;
  signal1 = -1;
  signal2 = 1;
Application["webfigure"] = new
WebFigure(rithmciz9.acceleratedjunctional_rhythm(heart_rate, signal1, signal2)); }

protected void RadioButton10_CheckedChanged(object sender, EventArgs e)
{ RadioButton1.Checked = false;
  double heart_rate, signal1, signal2;
  FirstdegreeAVblock rithmciz10 = new FirstdegreeAVblock();
  heart_rate = 62;
  signal1 = 1;
  signal2 = 1;

```

```

Application["webfigure"] = new
WebFigure(rithmciz10.firstdegree_AV_block(heart_rate, signal1, signal2)); }

protected void LinkButton1_Click(object sender, EventArgs e)
{
    RadioButton1.Checked = true;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
    RadioButton4.Checked = false;
    RadioButton5.Checked = false;
    RadioButton6.Checked = false;
    RadioButton7.Checked = false;
    RadioButton8.Checked = false;
    RadioButton9.Checked = false;
    RadioButton10.Checked = false;
}

double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave,
d_twave, signal1, signal2;

    EcgGrafik rithmciz1 = new EcgGrafik();
    heart_rate = 70;
    a_pwave = 0.15;
    d_pwave = 0.08;
    t_pwave = 0.15;
    a_qrswave = 1.00;
    d_qrswave = 0.08;
    a_twave = 0.30;
    d_twave = 0.18;
    signal1 = 1;
    signal2 = 1;

Application["webfigure"] = new WebFigure(rithmciz1.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));

    Label16.Text = "Sinüs Ritim : Kalbin normal ritmine, regüler ritim denir ve
ritim sinüs düğümünün kontrolü altındadır. Normal sinüs ritmi 60-100
atım/dakika'dır. Normal bir ritimde, EKG'de aynı dalgalar arasındaki uzaklık daima
birbirine eşittir. P dalgası her vuruda aynı görünümündedir, tümü SA(sinoatrial)

```


düğümünden çıkar.Genelde, PR aralığı 0,12-0,20 sn ve QRS aralığı 0,08-0,12 sn'dir.";

```
protected void LinkButton2_Click(object sender, EventArgs e)
```

```
{ RadioButton1.Checked = false;
  RadioButton2.Checked = true;
  RadioButton3.Checked = false;
  RadioButton4.Checked = false;
  RadioButton5.Checked = false;
  RadioButton6.Checked = false;
  RadioButton7.Checked = false;
  RadioButton8.Checked = false;
  RadioButton9.Checked = false;
  RadioButton10.Checked = false;
```

```
double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave,
d_twave, signal1, signal2;
```

```
Ecgrafik rithmciz2 = new Ecgrafik();
```

```
heart_rate = 50;
a_pwave = 0.15;
d_pwave = 0.08;
t_pwave = 0.15;
a_qrswave = 1.00;
d_qrswave = 0.08;
a_twave = 0.30;
d_twave = 0.18;
signal1 = 1;
signal2 = 1;
```

```
Application["webfigure"] = new WebFigure(rithmciz2.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));
```

Label16.Text = "Sinüs Bradikardi: Kalp atış hızının normal alt sınır değerinin altına düşmesiyle ortaya çıkan durumdur. SA(sinoatrial) düğümünden çıkan uyarılar dakikada 60'ın altındadır. Sinüs bradikardisi EKG özelliklerinde kalbin atım hızı 40-60 atım/dakika arasındadır ve ritimleri düzenlidir. P dalgaları ve QRS kompleksleri arasında 1:1 ilişki sürmektedir. Her QRS kompleksinin öncesinde bir P dalgası

görülür. Özellikle yaşlılarda olmak üzere sağlıklı insanlarda sık görülen bir aritmi tipidir." ; }

```
protected void LinkButton3_Click(object sender, EventArgs e)
{
    RadioButton1.Checked = false;
    RadioButton2.Checked = false;
    RadioButton3.Checked = true;
    RadioButton4.Checked = false;
    RadioButton5.Checked = false;
    RadioButton6.Checked = false;
    RadioButton7.Checked = false;
    RadioButton8.Checked = false;
    RadioButton9.Checked = false;
    RadioButton10.Checked = false;
}

double heart_rate, a_pwave, d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave,
d_twave, signal1, signal2;

Ecgrafik rithmciz3 = new Ecgrafik();
heart_rate = 110;
a_pwave = 0.15;
d_pwave = 0.08;
t_pwave = 0.15;
a_qrswave = 1.00;
d_qrswave = 0.08;
a_twave = 0.30;
d_twave = 0.18;
signal1 = 1;
signal2 = 1;

Application["webfigure"] = new WebFigure(rithmciz3.ecg_ciz(heart_rate, a_pwave,
d_pwave, t_pwave, a_qrswave, d_qrswave, a_twave, d_twave, signal1, signal2));

Label16.Text = "Sinüs Taşikardisi : Kalp atış hızının normal üst sınır değerinin
artmasıyla ortaya çıkan durumdur. SA(sinoatrial) düğümünden çıkan uyarılar
dakikada 100'ün üstündedir. Sinüs taşikardi EKG özelliklerinde kalbin atım hızı 100-
```

160 atım/dakikayken ritimler düzenlidir. P dalgasının büyüklüğü ve şekli, P-R aralığı ve QRS aralığı normaldir. Bebeklerde ve çocuklarda sık karşılaşılır."; }

```
protected void LinkButton4_Click(object sender, EventArgs e)
{
    RadioButton1.Checked = false;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
    RadioButton4.Checked = true;
    RadioButton5.Checked = false;
    RadioButton6.Checked = false;
    RadioButton7.Checked = false;
    RadioButton8.Checked = false;
    RadioButton9.Checked = false;
    RadioButton10.Checked = false;
    double heart_rate, signal1, signal2;
    JunctionalRhythm rithmciz4 = new JunctionalRhythm();
    heart_rate = 47;
    signal1 = 1;
    signal2 = 1;
    Application["webfigure"] = new WebFigure(rithmciz4.junctional_rhythm(heart_rate,
    signal1, signal2));
    Label16.Text = "Junctional Ritim: SA(sinoatrial) düğümün baskılanması
    nedeniyle, AV(atrioventriküler) düğümünde bir odak, vuru düzenleyici (pacemaker)
    olarak SA düğümünün yerini alır. P dalgaları QRS kompleksinden hemen önce ya da
    ondan sonra gelir.Sıklıkla kaçak (escape) bir ritim olarak ortaya çıkar.Junctional
    ritim hızı 40-60 atım/dakikadır ve ritmi düzenlidir."; }
```

```
protected void LinkButton5_Click(object sender, EventArgs e)
{
    RadioButton1.Checked = false;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
    RadioButton4.Checked = false;
```

```

RadioButton5.Checked = true;
RadioButton6.Checked = false;
RadioButton7.Checked = false;
RadioButton8.Checked = false;
RadioButton9.Checked = false;
RadioButton10.Checked = false;
double heart_rate, signal1, signal2;
MonomorphicVT rithmciz5 = new MonomorphicVT();
heart_rate = 195;
signal1 = 1;
signal2 = 1;

```

```

Application["webfigure"] = new WebFigure(rithmciz5.monomorphic_VT(heart_rate,
signal1, signal2));

```

Label16.Text = "Monomorfik Ventriküler Taşikardi: Ventriküler taşikardi kendi morfolojisine göre; tüm QRS komplekslerinin benzer (monomorfik) ya da farklı (polimorfik) olup olmadığına bağlı olarak, monomorfik veya polimorfik olarak sınıflandırılabilir. Ventriküler aktivasyon sırası sabit olduğunda, elektrokardiyografik şekil aynı kalır ve aynı QRS EKG dalgasının kendini tekrarladığı görülür. Bu ritim monomorfik VT olarak adlandırılır."; }

```

protected void LinkButton6_Click(object sender, EventArgs e)
{
RadioButton1.Checked = false;
RadioButton2.Checked = false;
RadioButton3.Checked = false;
RadioButton4.Checked = false;
RadioButton5.Checked = false;
RadioButton6.Checked = true;
RadioButton7.Checked = false;
RadioButton8.Checked = false;
RadioButton9.Checked = false;
RadioButton10.Checked = false;
double heart_rate, signal1, signal2;
IdioventricularRhythm rithmciz6 = new IdioventricularRhythm();

```

```

heart_rate = 35;
signal1 = 1;
signal2 = -1;
Application["webfigure"] = new
WebFigure(rithmciz6.idioventricular_rhythm(heart_rate, signal1, signal2));
Label16.Text = "İdioventriküler Ritim: Ventriküler ektopik, bir odak uyarıları
idioventriküler hızda yolladığı zaman oluşur.Sıklıkla total kalp bloğu ile birlikte
görülür. Hızı 30-40 atım/dakika ve ritmi düzenli olan idioventriküler ritimde P
dalgası yoktur."; }

```

```

protected void LinkButton7_Click(object sender, EventArgs e)
{
RadioButton1.Checked = false;
RadioButton2.Checked = false;
RadioButton3.Checked = false;
RadioButton4.Checked = false;
RadioButton5.Checked = false;
RadioButton6.Checked = false;
RadioButton7.Checked = true;
RadioButton8.Checked = false;
RadioButton9.Checked = false;
RadioButton10.Checked = false;
double heart_rate, signal1, signal2;
AccelidioventricularRhythm rithmciz7 = new AccelidioventricularRhythm();
heart_rate = 65;
signal1 = 1;
signal2 = -1;
Application["webfigure"] = new
WebFigure(rithmciz7.acceleratedidioventricular_rhythm(heart_rate, signal1,
signal2));
Label16.Text = "Akselere İdioventriküler Ritim : Akselere (hızlandırılmış)
idioventriküler ritimin ventrikül oranı genellikle 40 ile 100-120 atım/dakika
arasındadır. Bu ritimde P dalgası yoktur. Genellikle geçici bir ritimdir ve nadiren
tedavi gerektirmektedir."; }

```

```
protected void LinkButton8_Click(object sender, EventArgs e)
{
    RadioButton1.Checked = false;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
    RadioButton4.Checked = false;
    RadioButton5.Checked = false;
    RadioButton6.Checked = false;
    RadioButton7.Checked = false;
    RadioButton8.Checked = true;
    RadioButton9.Checked = false;
    RadioButton10.Checked = false;
    double heart_rate, signal1, signal2;
    SupraventricularTachycardia rithmciz8 = new SupraventricularTachycardia();
    heart_rate = 180;
    signal1 = 1;
    signal2 = 1;
    Application["webfigure"] = new
    WebFigure(rithmciz8.supraventricular_tachycardia(heart_rate, signal1, signal2));
    Label16.Text = "Supraventriküler Taşikardi : Supraventriküler taşikardiler
(SVT), his düğümü seviyesi üzerinden kaynak alırlar. SVT'ler daha çok dar
kompleksli taşikardi şeklindedir. Sık karşılaşılan acil servise çarpıntı şikayeti ile
başvuran hastaların çoğunda tespit edilir. Bu çarpıntılar(taşikardiler) genellikle
atriyumdan(kulakçık) başlar ya da atriyumları içine aldığı için bu şekilde adlandırılır.
Aniden başlayıp aniden sonlanırlar genellikle hızları 150-280 atım/dakika'dır. Hayati
tehlikeye neden olmazlar fakat oldukça rahatsız eder, hastaların yaşam kalitesini
düşürür. Kalp ritim uzmanları tarafından çeşitli sınıflamaları vardır."; }
```

```
protected void LinkButton9_Click(object sender, EventArgs e)
{
    RadioButton1.Checked = false;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
    RadioButton4.Checked = false;
    RadioButton5.Checked = false;
}
```

```

RadioButton6.Checked = false;
RadioButton7.Checked = false;
RadioButton8.Checked = false;
RadioButton9.Checked = true;
RadioButton10.Checked = false;
double heart_rate, signal1, signal2;
AcceljunctionalRhythm rithmciz9 = new AcceljunctionalRhythm();
heart_rate = 80;
signal1 = -1;
signal2 = 1;
Application["webfigure"] = new
WebFigure(rithmciz9.acceleratedjunctional_rhythm(heart_rate, signal1, signal2));
    Label16.Text = "Akselere Junctional Ritim: Sıklıkla aşırı sayısal bilgi alımı
nedeniyle AV(atrioventriküler) düğüm içinde artmış otomatisiteden kaynaklanır.
Hızı 60-100 atım/dakika olan akselere junctional ritimde, P dalgası kulakçığa
retrograd gönderilir ve pozitif derivasyonlarda negatif, negatif derivasyonlarda
pozitiftir."; }

protected void LinkButton10_Click(object sender, EventArgs e)
{RadioButton1.Checked = false;
    RadioButton2.Checked = false;
    RadioButton3.Checked = false;
    RadioButton4.Checked = false;
    RadioButton5.Checked = false;
    RadioButton6.Checked = false;
    RadioButton7.Checked = false;
    RadioButton8.Checked = false;
    RadioButton9.Checked = false;
    RadioButton10.Checked = true;
    double heart_rate, signal1, signal2;
    FirstdegreeAVblock rithmciz10 = new FirstdegreeAVblock();
    heart_rate = 62;
    signal1 = 1;

```

```
signal2 = 1;
Application["webfigure"] = new
WebFigure(rithmciz10.firstdegree_AV_block(heart_rate, signal1, signal2));
    Label16.Text = "Birinci Derece AV Blok : Burada elektriksel uyarı
SA(sinoatrial) düğümde doğar ve AV(atrioventriküler) düğüme iletilir. Uyarı AV
kavşağa ulaştığında karıncıklara iletilmeden önce geçici olarak gecikir. Düzenli bir
ritme sahip olan birinci derece AV blokta atım hızı normaldir. PR aralığının 0.21
sn'den daha uzun olması ile karakterizedir. Genellikle tedavi gerektirmeyen bir ritim
bozukluğudur."; }
}
```


ÖZGEÇMİŞ

Esra GÜNEY, 05.01.1987 tarihinde Erzincan'da doğdu. 2004 yılında Merzifon Lisesi'nden mezun oldu. 2005 yılında başladığı Kocaeli Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği Bölümünü 2009 yılında bitirdi. Aynı yıl Sakarya Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi ABD'de yüksek lisans eğitimine başladı. 2009-2010 yılları arasında Adapazarı ATSO Ticaret Meslek Lisesi'nde Bilgisayar öğretmenliği yaptı.