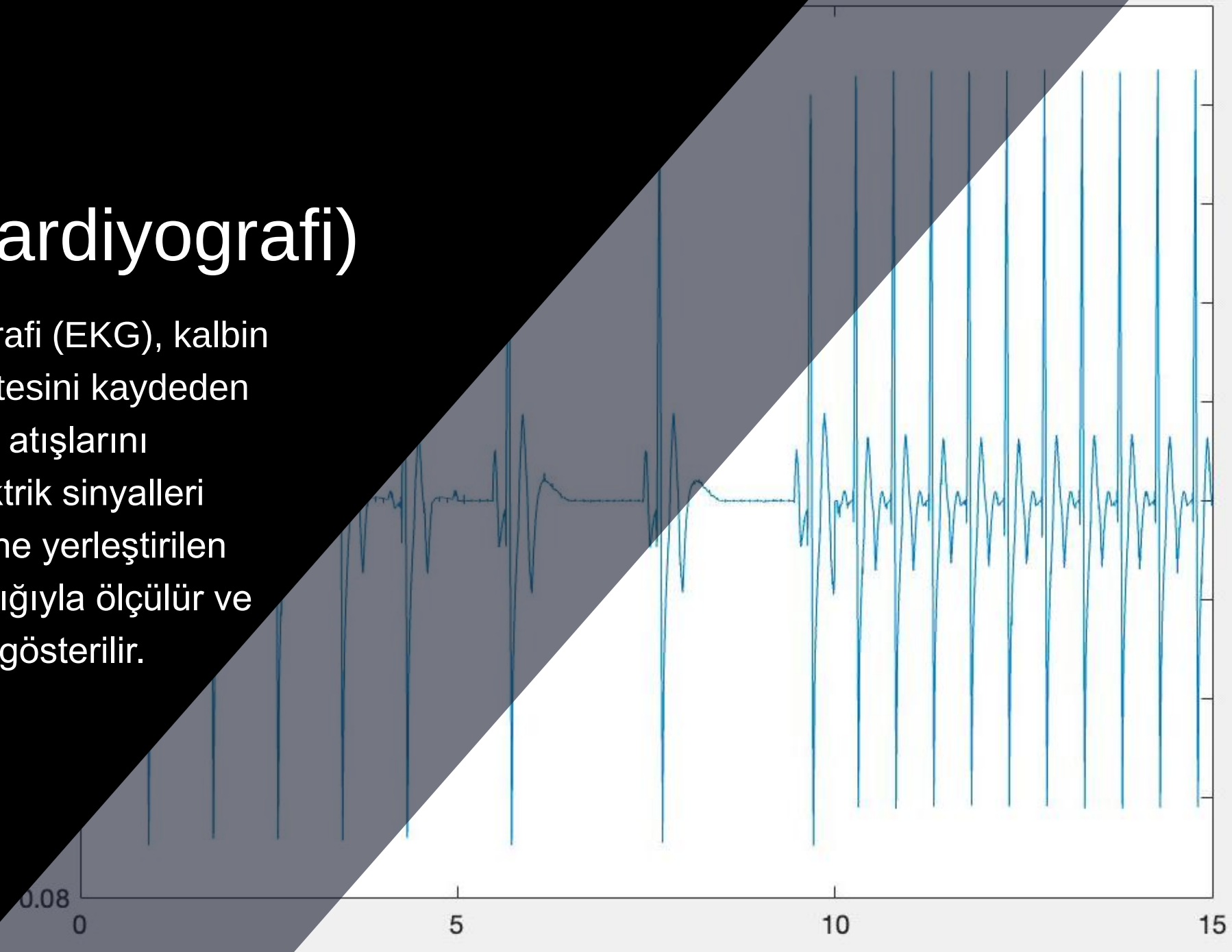
BİL265-BME207 PROJE

Ali Emir Septioğlu
Özlem Kayhan
Çağan Kırmızı
Zeynep Bahar Kaya
Selin Koç
Berna Akpınar

EKG (Elektrokardiyografi)

- Elektrokardiyografi (EKG), kalbin elektriksel aktivitesini kaydeden bir testtir. Kalbin atışlarını düzenleyen elektrik sinyalleri vücudun yüzeyine yerleştirilen elektrotlar aracılığıyla ölçülür ve bir grafik olarak gösterilir.

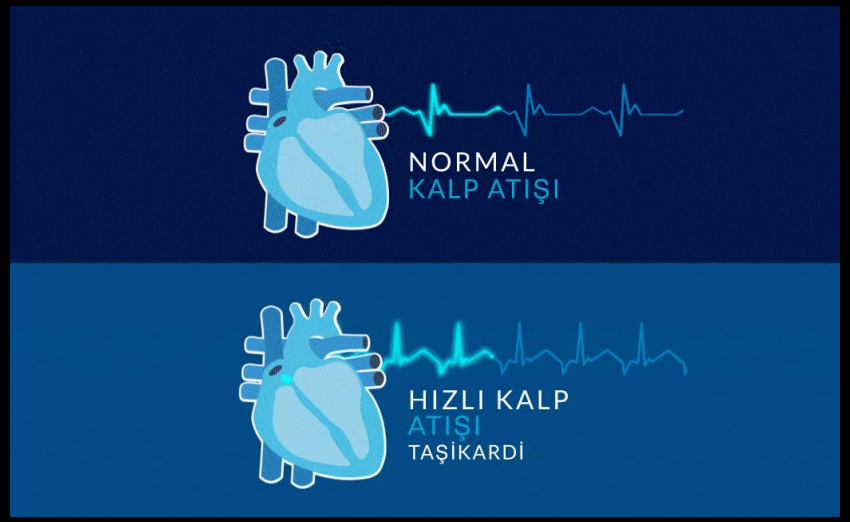


Kullanım Alanları:

- Kalp ritim bozukluklarını (aritmi) teşhis etmek
- Kalp krizi (miyokard enfarktüsü) belirtilerini değerlendirmek
- Kalp kası ve kalp kapakçığı hastalıklarını incelemek
- Elektriksel iletim sorunlarını saptamakEKG, hızlı ve non-invaziv bir test olması nedeniyle kardiyoloji pratiğinde sıkça kullanılır.

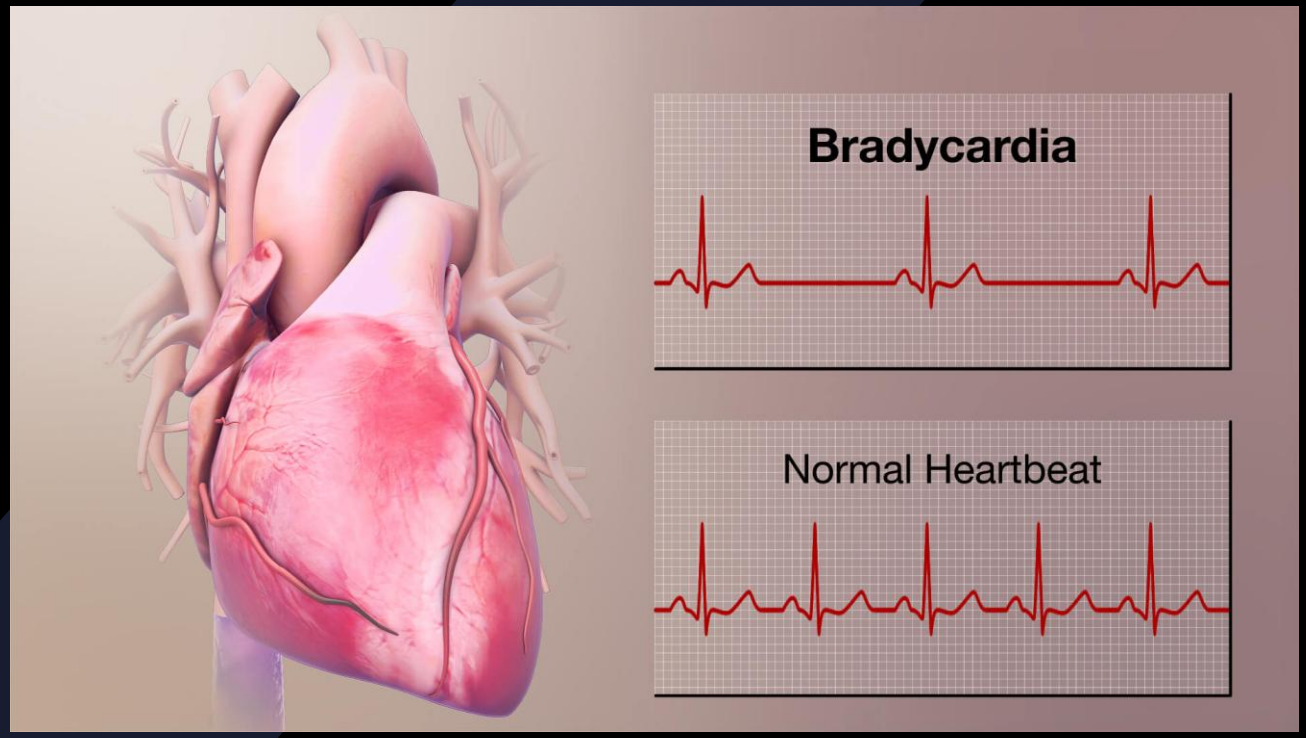
Taşikardi

- Taşikardi, kalbin dakikada 100'den fazla atması durumudur. Bu durum stres, egzersiz, ateş, kansızlık, ya da kalp ritim bozuklukları gibi nedenlerle ortaya çıkabilir.



Bradikardi

- Bradikardi, kalbin dakikada 60'tan az atması durumudur. Atletik bireylerde bu durum normal kabul edilebilirken, yaşlılarda veya kalp iletim sistemi hastalıklarında ciddi sorunlara işaret edebilir.



```
clear all
close all

load('data07 (1).dat');
bradycardia = data07__1_;


load('data14 (1).dat');
normal = data14__1_;


load('data25 (1).dat');
tachycardia = data25__1_;

t = 1/8000:1/8000:15;


alldata = vertcat(bradycardia, normal, tachycardia);
plot(t,alldata);


veriTable = table(t',alldata,'VariableNames',{'Time','Voltage'});


disp(veriTable);

writetable(veriTable, 'alldata.txt','Delimiter','\t');

disp(veriTable);

```

Gerçekleştirilen Adımlar

1- Veri Gönderimi:

Simülatör ile oluşturulan EKG sinyalleri hazır halde bize iletildi.

Sinyaller, PQRST dalga dizilerini içeriyordu.

2- Veri Hazırlığı:

EKG verileri 5 saniyelik kayıtlarla alınmış, farklı ritimler birleştirilerek 15 saniyelik setler oluşturulmuştu

3- Verilerin Analizi:

Gönderilen verilerden P, R ve T dalga noktaları belirlendi.

BPM hesaplanarak “Normal”, “Taşikardi” ve “Bradikardi” olarak sınıflandırıldı.

4- Sonuçların Birleştirilmesi:

Sınıflandırılan veriler, toplu dosyalarda birleştirildi ve analiz raporları oluşturuldu.

Voltaj ve Zaman Değerlerini Tutan Sınıf

Berna Akpınar

```cpp
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
using namespace std;

class dd{
    public:
    dd(double a=0 , double b=0): times(a), value(b){};

    void display(){
        cout<<times<<"          "<<value<<endl;
    };

    double times, value;
};
```

Main Fonksiyonu

Özlem Kayhan

```
136  int main()
137  {
138      vector<dd> rt1, rt2, rt3;
139
140      ifstream data1("data08.txt",ios::in);
141      ifstream data2("data16.txt",ios::in);
142      ifstream data3("data23.txt",ios::in);
143
144      if (!data1 || !data2 || !data3) {
145          cerr << "Error opening DATA file!" << endl;
146          return EXIT_FAILURE;
147      }
148
149      rHesaplama(data1,rt1);
150      rHesaplama(data2,rt2);
151      rHesaplama(data3,rt3);
152
153      BPMhesapla(rt1,"Patient");
154      BPMhesapla(rt2,"Patient");
155      BPMhesapla(rt3,"Patient");
156
157      merge("Patient");
158
159      cout<<"Operation Complated Successfuly..."<<endl;
160
161      return 0;
162  }
```

R Dalgası Hesaplama Fonksiyonu

Çağan Kırmızı

```
51
52  void rHesaplama(ifstream& file1,vector<dd>& rt){
53
54      vector<dd> f1;
55      double tim, val;
56
57      ifstream timef("time.txt",ios::in);
58
59      while(timef>>tim){
60          file1>>val;
61          dd nmb(tim, val);
62
63          f1.push_back(nmb);
64      }
65
```

```
66
67      for (vector<dd>::iterator it1 = f1.begin() ; it1!=f1.end()-2 ; it1++) {
68
69          while( (*it1).value < 0.04 && it1!=f1.end()-1)
70              it1++;
71
72          if(it1==f1.end())
73              break;
74
75          vector<dd>::iterator it2 = it1+1;
76
77          double max = 0;
78          while((*it2).value > 0.04 && it2!=f1.end()-1){
79              if( (*it2).value > max){
80                  max = (*it2).value;
81                  it1 = it2;
82              }
83              it2++;
84          }
85
86          if(it2==f1.end())
87              break;
88
89          if((*it1).value > 0.05)
90              rt.push_back( (*it1) );
91          it1 = it2;
92
93      }
94      if (rt.empty()) {
95          cout << "R bulunamadi";
96          exit(1);
97      }
98
99      file1.close();
00      timef.close();
01  }
```

BPM Hesaplama Fonksiyonu

Zeynep Bahar Kaya

```
25
26  void BPMhesapla(vector<dd>& rt, string patient){
27
28      double bpm;
29
30      bpm = (60.0 / (rt[1].times - rt[0].times) );
31
32      if (bpm < 60.0){
33          fstream bradi(patient+"_Bradikardi.txt",ios::out);
34          bradi<<"Time:       "<<"    Bradikardi Value:"<<endl;
35          write(rt,bradi);
36      }
37      else if (bpm > 100.0){
38          fstream tasi(patient+"_TaÅŸikardi.txt",ios::out);
39          tasi<<"Time:        "<<"    TaÅŸikardi Value:"<<endl;
40
41          write(rt,tasi);
42      }
43      else{
44          fstream normal(patient+"_Normal.txt",ios::out);
45          normal<<"Time:      "<<"    Normal Value:"<<endl;
46          write(rt,normal);
47      }
48
49
50  }
```

Dosyaya Yazdırma Fonksiyonu

Berna Akpınar

```
17
18  void write(vector<dd>& vec, fstream& file){
19
20      for(size_t i=0 ; i<vec.size() ; i++)
21          file<<vec[i].times<<"       "<<vec[i].value<<endl;
22
23      file.close();
24  }
25
```

Verilerin Birleştirilmesi Fonksiyonu

Selin Koç

```
void merge(string name){

    ifstream bradi(name+"_Bradikardi.txt");
    ifstream tasi(name+"_TaÅŸikardi.txt");
    ifstream normal(name+"_Normal.txt");
    fstream merge(name+".txt",ios::out);

    if (!bradi || !tasi|| !normal || !merge) {
        cerr << "Error opening PATIENT file!" << endl;
        exit(3);
    }

```

```
    string t,v;

    merge<<"BRADÄ°KARDÄ°\n----------\nTime:     Bradikardi Value:"<<endl;
    bradi>>t>>v>>v;
    while(bradi>>t>>v)
        merge<<t<<"          "<<v<<endl;
    merge<<"\n\n\n\n";

    merge<<"NORMAL\n----------\nTime:       Normal Value:"<<endl;
    normal>>t>>v>>v;
    while(normal>>t>>v)
        merge<<t<<"          "<<v<<endl;
    merge<<"\n\n\n\n";

    merge<<"TAÅ Ä°KARDÄ°\n----------\nTime:     TaÅŸikardi Value:"<<endl;
    tasi>>t>>v>>v;
    while(tasi>>t>>v)
        merge<<t<<"          "<<v<<endl;
    merge<<"\n\n\n\n";
}
```