ECG SIMULATION USING MATLAB

Principle of Fourier Series

Presented by
R. KARTHIK
B.E. (E.C.E), 6TH Semester
College of Engineering, Guindy,
Anna University,
Chennai – 600025
e-mail: may_62003@yahoo.co.in

Introduction:

The aim of the ECG simulator is to produce the typical ECG waveforms of different leads and as many arrhythmias as possible. My ECG simulator is a matlab based simulator and is able to produce normal lead II ECG waveform.

The use of a simulator has many advantages in the simulation of ECG waveforms. First one is saving of time and another one is removing the difficulties of taking real ECG signals with invasive and noninvasive methods. The ECG simulator enables us to analyze and study normal and abnormal ECG waveforms without actually using the ECG machine. One can simulate any given ECG waveform using the ECG simulator.

Significant features of ECG waveform:

A typical scalar electrocardiographic lead is shown in Fig. 1, where the significant features of the waveform are the P, Q, R, S, and T waves, the duration of each wave, and certain time intervals such as the P-R, S-T, and Q-T intervals.

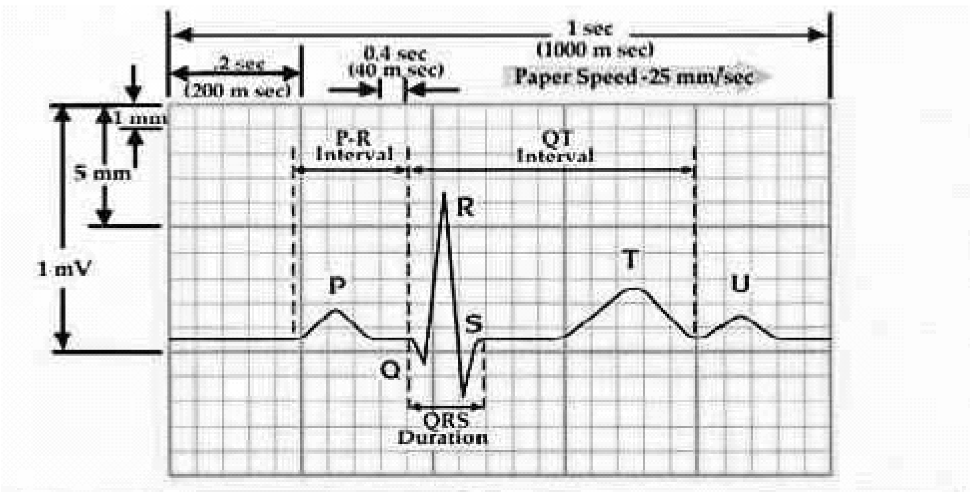


fig 1.Typical ECG signal

Main features of this simulator:

- Any value of heart beat can be set
- Any value of intervals between the peaks (ex-PR interval) can be set
- Any value of amplitude can be set for each of the peaks
- Fibrillation can be simulated
- Noise due to the electrodes can be simulated
- Heart pulse of the particular ECG wave form can be represented in a separate graph

Principle:

Fourier series

Any periodic functions which satisfy dirichlet's condition can be expressed as a series of scaled magnitudes of sin and cos terms of frequencies which occur as a multiple of fundamental frequency.

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

$$a_o = (1/l)\int_T f(x)\,dx \qquad , T = 2l \qquad \dashrightarrow (1)$$

$$a_n = (1/l)\int_T f(x)\cos(n\pi x / l)\,dx \qquad , n = 1,2,3\ldots \qquad \dashrightarrow (2)$$

$$b_n = (1/l)\int_T f(x)\sin(n\pi x / l)\,dx \qquad , n = 1,2,3\ldots \qquad \dashrightarrow (3)$$

ECG signal is periodic with fundamental frequency determined by the heart beat. It also satisfies the dirichlet's conditions:

- Single valued and finite in the given interval
- Absolutely integrable
- Finite number of maxima and minima between finite intervals
- It has finite number of discontinuities

Hence fourier series can be used for representing ECG signal.

Calculations:

If we observe figure1, we may notice that a single period of a ECG signal is a mixture of triangular and sinusoidal wave forms. Each significant feature of ECG signal can be represented by shifted and scaled versions one of these waveforms as shown below.

- QRS, Q and S portions of ECG signal can be represented by triangular waveforms
- P, T and U portions can be represented by triangular waveforms

Once we generate each of these portions, they can be added finally to get the ECG signal.

Lets take QRS waveform as the centre one and all shiftings takes place with respect to this part of the signal.

How do we generate periodic QRS portion of ECG signal

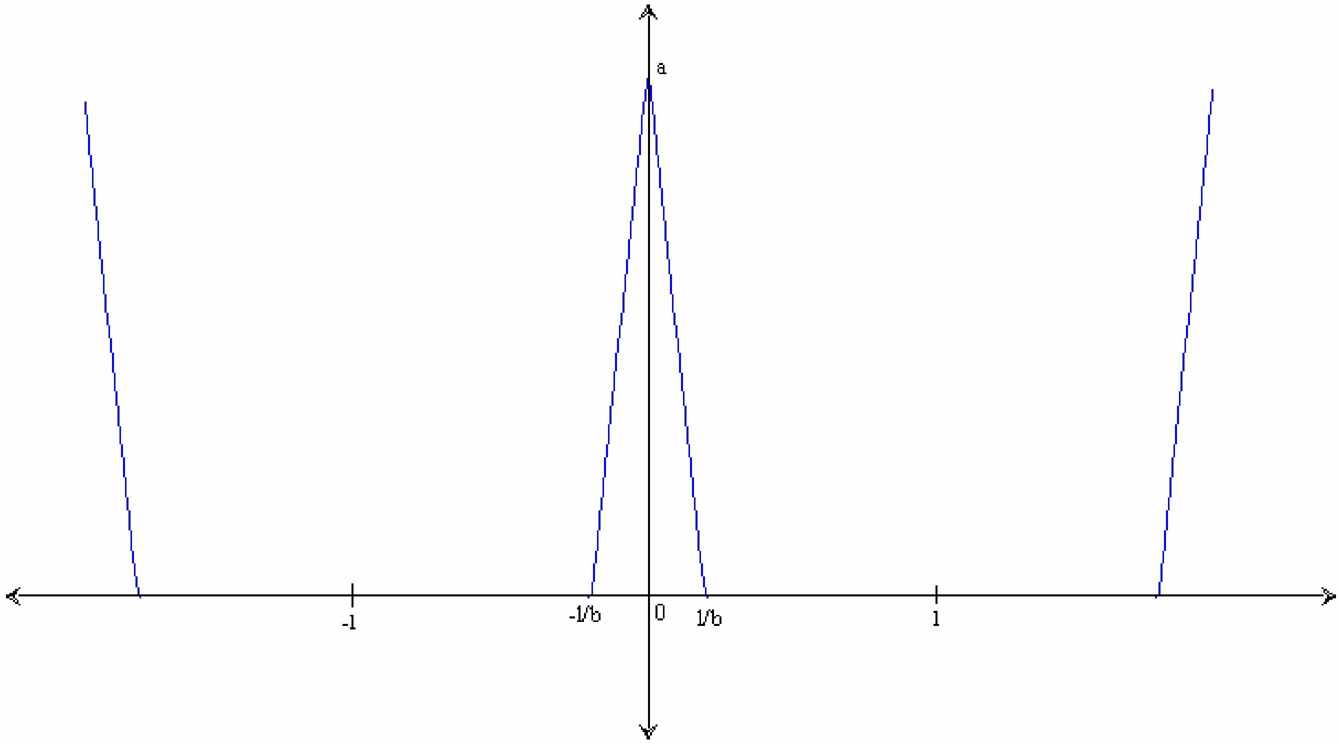


Fig 2. generating QRS waveform

From equation (1), we have

$$f(x) = (-bax/l) + a \qquad 0 < x < (l/b)$$
$$\quad = (bax/l) + a \qquad (-l/b) < x < 0$$

$$a_o = (1/l) \int_T f(x)\, dx$$
$$\quad = (a/b) * (2 - b)$$

$$a_n = (1/l) \int_T f(x) \cos(n\pi x / l)\, dx$$
$$\quad = (2ba / (n^2\pi^2)) * (1 - \cos(n\pi/b))$$

$$b_n = (1/l) \int_T f(x) \sin(n\pi x / l)\, dx$$
$$\quad = 0 \quad \text{(because the waveform is a even function)}$$

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

How do we generate periodic p-wave portion of ECG signal

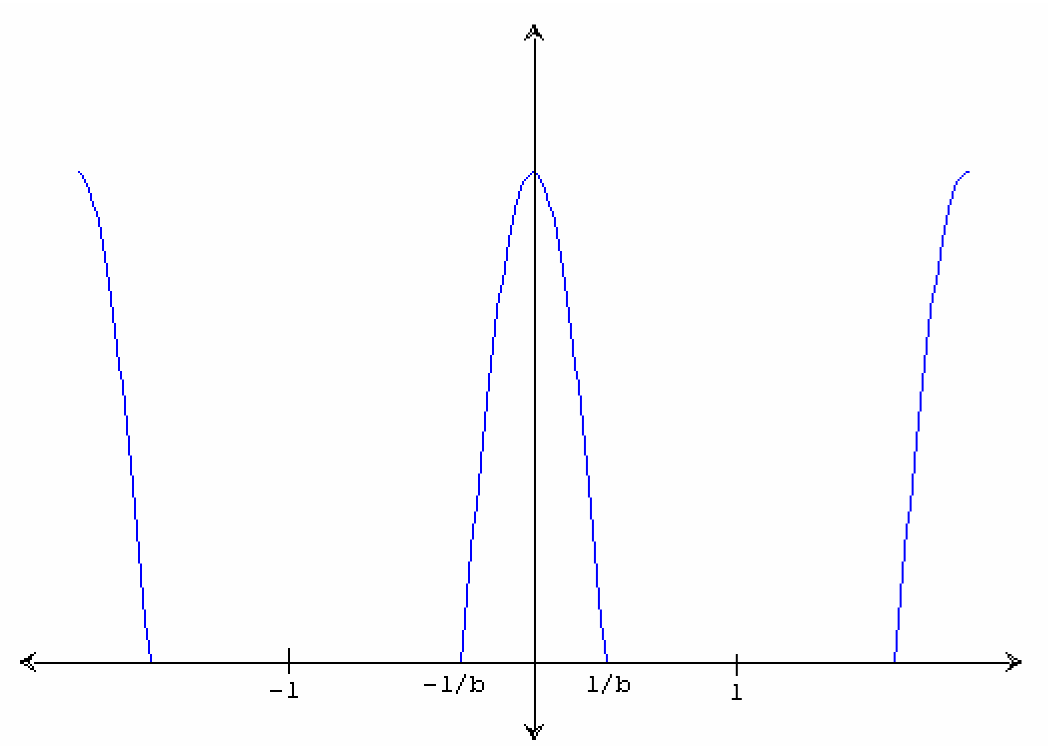


Fig 3. generation of p-wave

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

$$a_o = (1/l)\int_T \cos((\pi bx)/(2l))\,dx$$
$$= (a/(2b))(2-b)$$

$$a_n = (1/l)\int_T \cos((\pi bx)/(2l))\cos(n\pi x/l)\,dx$$
$$= (((2ba)/(i^2\pi^2))(1-\cos((n\pi)/b)))\cos((n\pi x)/l)$$

$$b_n = (1/l)\int_T \cos((\pi bx)/(2l))\sin(n\pi x/l)\,dx$$
$= 0$ (because the waveform is a even function)

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

Implementation in MATLAB:

Code:

Save the below file as complete.m

```
x=0.01:0.01:2;
default=input('Press 1 if u want default ecg signal else press 2:\n');
if(default==1)
    li=30/72;

    a_pwav=0.25;
    d_pwav=0.09;
    t_pwav=0.16;

    a_qwav=0.025;
    d_qwav=0.066;
    t_qwav=0.166;

    a_qrswav=1.6;
    d_qrswav=0.11;

    a_swav=0.25;
    d_swav=0.066;
```

```
      t_swav=0.09;

      a_twav=0.35;
      d_twav=0.142;
      t_twav=0.2;

      a_uwav=0.035;
      d_uwav=0.0476;
      t_uwav=0.433;
else
   rate=input('\n\nenter the heart beat rate :');
   li=30/rate;

   %p wave specifications
   fprintf('\n\np wave specifications\n');
   d=input('Enter 1 for default specification else press 2: \n');
   if(d==1)
      a_pwav=0.25;
      d_pwav=0.09;
      t_pwav=0.16;
   else
     a_pwav=input('amplitude = ');
     d_pwav=input('duration = ');
     t_pwav=input('p-r interval = ');
     d=0;
   end


   %q wave specifications
   fprintf('\n\nq wave specifications\n');
   d=input('Enter 1 for default specification else press 2: \n');
   if(d==1)
      a_qwav=0.025;
      d_qwav=0.066;
      t_qwav=0.166;
   else
     a_qwav=input('amplitude = ');
     d_qwav=input('duration = ');
     t_qwav=0.1;
     d=0;
   end

   %qrs wave specifications
```

```
fprintf('\n\nqrs wave specifications\n');
d=input('Enter 1 for default specification else press 2: \n');
if(d==1)
    a_qrswav=1.6;
    d_qrswav=0.11;
else
   a_qrswav=input('amplitude = ');
   d_qrswav=input('duration = ');
   d=0;
end


%s wave specifications
fprintf('\n\ns wave specifications\n');
d=input('Enter 1 for default specification else press 2: \n');
if(d==1)
    a_swav=0.25;
    d_swav=0.066;
    t_swav=0.125;
else
   a_swav=input('amplitude = ');
   d_swav=input('duration = ');
   t_swav=0.125;
   d=0;
end


%t wave specifications
fprintf('\n\nt wave specifications\n');
d=input('Enter 1 for default specification else press 2: \n');
if(d==1)
    a_twav=0.35;
    d_twav=0.142;
    t_twav=0.18;
else
   a_twav=input('amplitude = ');
   d_twav=input('duration = ');
   t_twav=input('s-t interval = ');
   d=0;
end
%u wave specifications
fprintf('\n\nu wave specifications\n');
d=input('Enter 1 for default specification else press 2: \n');
```

```
    if(d==1)
        a_uwav=0.035;
        d_uwav=0.0476;
        t_uwav=0.433;
    else
       a_uwav=input('amplitude = ');
       d_uwav=input('duration = ');
       t_uwav=0.433;
       d=0;
    end
end

 pwav=p_wav(x,a_pwav,d_pwav,t_pwav,li);


 %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);


 %uwav output
 uwav=u_wav(x,a_uwav,d_uwav,t_uwav,li);

 %ecg output
 ecg=pwav+qrswav+twav+swav+qwav+uwav;
 figure(1)
 plot(x,ecg);
```

Save the below file as p_wav.m

```
function [pwav]=p_wav(x)
l=1;
a=0.25
x=x+(1/1.8);
b=3;
n=100;
p1=1/l
p2=0
for i = 1:n
   harm1=(((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);
   p2=p2+harm1
end
pwav1=p1+p2;
pwav=a*pwav1;
```

Save the below file as q_wav.m

```
function [qwav]=q_wav(x)
l=1;
x=x+l/6
a=0.025;
b=15;
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);
```

Save the below file as qrs_wav.m

```
function [qrswav]=qrs_wav(x)
l=1;
a=1;
b=5;
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;
```

Save the below file as s_wav.m

```
function [swav]=s_wav(x)
l=1;
x=x-l/6
a=0.25;
b=15;
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);
```

Save the below file as t_wav.m

```
function [twav]=t_wav(x)
l=1;
a=0.35
x=x-(1/1.8);
b=7;
n=20;
t1=1/l
t2=0
for i = 1:n
   harm2=(((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);
   t2=t2+harm2
end
twav1=t1+t2;
twav=a*twav1;
```

Save the below file as u_wav.m

```
function [uwav]=u_wav(x)
l=1;
a=0.03;
x=x-(1/1.1);
b=21;
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;
```

precautions:

- All the files have to be saved in the same folder
- Save the files in the names mentioned above the code
- While entering the specification, give the amplitude in mV and duration in seconds

Output waveform:

Default Specification

- Heart beat :72
- Amplitude:

  P wave 25mV

  R wave 1.60mV

  Q wave 0.025mV

  T wave 0.35mV

- Duration:

  P-R interval 0.16s

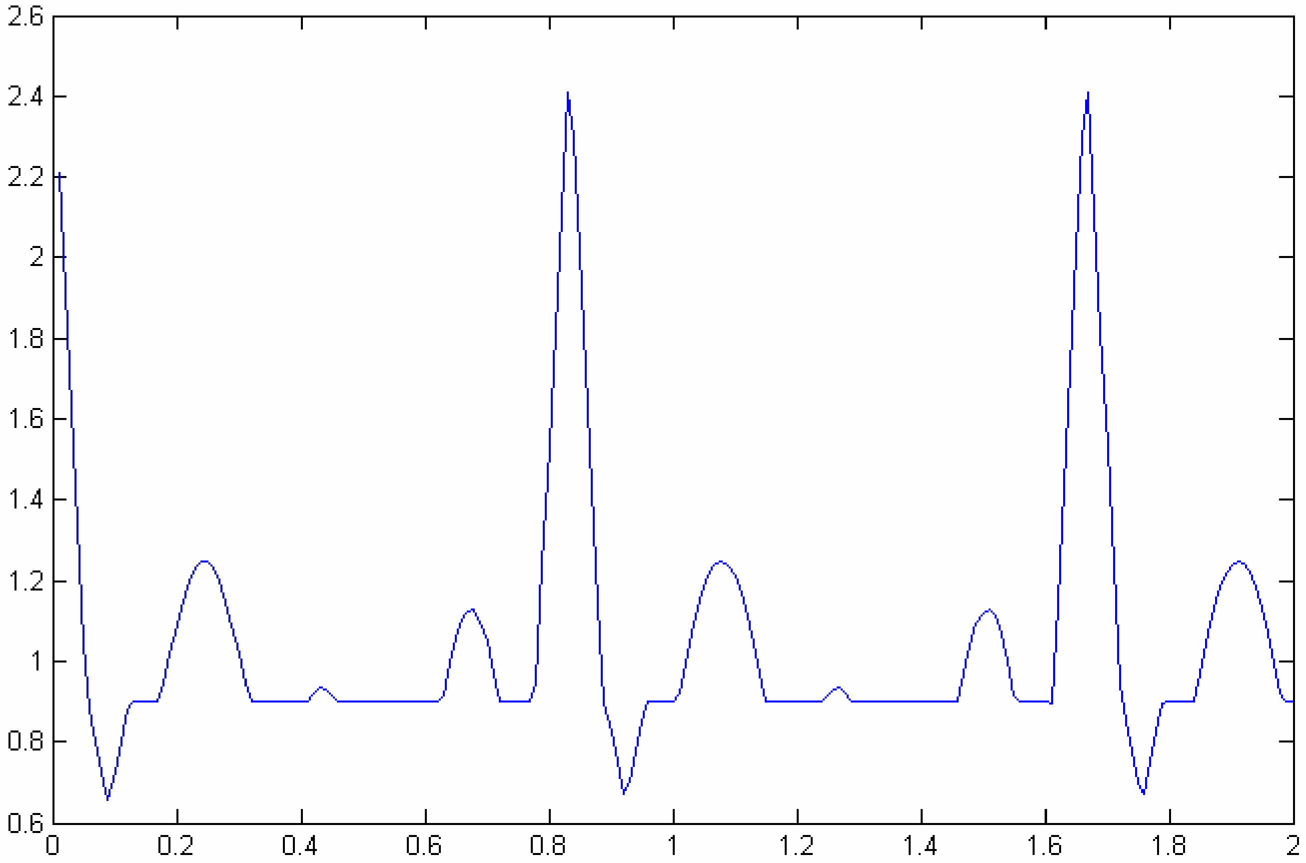
  S-T interval 0.18s

P interval 0.09s

QRS interval 0.11s

Not all the default values are specified here. They can be obtained from the code of the simulator from the file complete.m. The user can enter their desired values of specifications too. Other concepts of the code are simple and are self explanatory.

A typical output for the above specification will be like this:



References:

- R.S. Khandpur, Handbook of Biomedical Instrumentation
- Leslie Cromwell, Biomedical Instrumentation and Measurements, Prentice Hall of India.
- MATLAB The Language of Technical Computing, The Mathworks.