Національний технічний університет України

«Київський політехнічний інститут»

Факультет електроніки

Кафедра фізичної та біомедичної електроніки

Лабораторна робота №1

«Вивчення принципів побудови БЕС моніторингу

стану пацієнта на прикладі 12-канальногоелектрокардіографа

“UCARD-200” та

реанімаційно-хірургічного монітора “UM-300”**»**

З курсу: «Біомедичні комп’ютерні системи»

Виконала студентка Vкурсу

Групи ДМ-61м

Нікітюк Н.

Київ 2016

*Мета роботи:*ознайомитися з основами побудови та технічнимихарактеристиками 12-ти канального електрокардіографа “UCARD-200” тареанімаційно-хірургічного монітора “UM-300”; освоїти методики реєстрації та аналізу електрокардіосигналів та основних показників моніторингу життєво важливих функцій пацієнта.

10. Розробити стійкий до артефактів алгоритм обчислення ЧСС по каналу ЕКГ та SpO2. Реалізувати алгоритм у вигляді програми. Порівняти розраховані показники ЧСС з показниками, отриманими під час моніторингу.

*Лістинг програми:*

**script.m***(main file)***:**

% Preload

fid=fopen('record.bin');

j = fread(fid,[9 inf],'int16');

fclose(fid);

ecg = j(1,:);

Fs = 151.51;

T=(length(ecg)-1)/Fs;

t=0:1/Fs:T;

f=0:1/T:Fs;

% Змінні

k = 5; % Sets how many of FFT coefficients are to be set to 0

m = 600; % ~600 works fine | window size coefficient

threshold = 0.2; % Threshold filtering

% Видалити низькі частоти

fft\_result=fft(ecg);

fft\_result(1 : round(length(fft\_result)\*k/Fs))=0;

fft\_result(end - round(length(fft\_result)\*k/Fs) : end)=0;

corrected=real(ifft(fft\_result));

% Filter

WinSize = floor(Fs \* m / 1000);

if rem(WinSize,2)==0

WinSize = WinSize+1;

end

filtered=winmax(corrected, WinSize);

% Scale ecg

peaks=filtered/max(filtered);

% Filter by threshold filter

for j = 1:1:length(peaks)

if peaks(j) < threshold

peaks(j) = 0;

else

peaks(j)=1;

end

end

positions=find(peaks);

distance=positions(2)-positions(1);

% Returns minimum distance between two peaks

for j=1:1:length(positions)-1

if positions(j+1)-positions(j)<distance

distance=positions(j+1)-positions(j);

end

end

% Average HR

peakpos=find(peaks);

hr = 60\*Fs /(( peakpos(end) - peakpos(1))/length(peakpos));

disp('Average Heart Rate = ');

disp(hr);

% HR in dynamics

for j=1:length(peakpos)-2

instantaneousHR(j) = 2\*60\*Fs/( (peakpos(j+1) - peakpos(j) )+( peakpos(j+2) - peakpos(j+1) ) );

end

% Instantaneous Heart Rate

figure; plot( t(peakpos(1:end-2)), instantaneousHR );

title('\bf1. Instantaneous Heart Rate'); xlabel('t, sec'); ylabel('heart rate, bpm');

xaxis = [10 30]; % For better visibility

% Original input ECG j

figure; plot(t, ecg);

title('\bf1. Original ECG'); xlim(xaxis); xlabel('t, sec');

% ECG with removed low-frequency component

figure; plot(t, corrected);

title('\bf2. FFT Filtered ECG'); xlim(xaxis); xlabel('t, sec');

% Filtered ECG (1-st pass) - filter has default window size

figure; stem(t, filtered);

title('\bf3. Filtered ECG'); xlim(xaxis); xlabel('t, sec');

% Detected peaks in filtered ECG

figure; stem(t, peaks);

title('\bf4. Detected Peaks'); xlim(xaxis); xlabel('t, sec');

**winmax.m***(custom function)*:

function Filtered=winmax(Original, WinSize)

%initialising variables

WinHalfSize = floor(WinSize/2);

WinHalfSizePlus = WinHalfSize+1;

WinSizeSpec = WinSize-1;

FrontIterator = 1;

WinPos = WinHalfSize;

WinMaxPos = WinHalfSize;

WinMax = Original(1);

OutputIterator = 0;

% Finding the postion of the largest value in window

forLengthCounter = 0:1:WinHalfSize-1

if Original(FrontIterator+1) >WinMax

WinMax = Original(FrontIterator+1);

WinMaxPos = WinHalfSizePlus + LengthCounter;

end

FrontIterator=FrontIterator+1;

end

% if the first point is the highest, set ouput 1

ifWinMaxPos == WinHalfSize

Filtered(OutputIterator+1)=WinMax;

else

Filtered(OutputIterator+1)=0;

end

OutputIterator = OutputIterator+1;

% search next half of signal

forLengthCounter = 0:1:WinHalfSize-1

if Original(FrontIterator+1)>WinMax

WinMax=Original(FrontIterator+1);

WinMaxPos=WinSizeSpec;

else

WinMaxPos=WinMaxPos-1;

end

ifWinMaxPos == WinHalfSize

Filtered(OutputIterator+1)=WinMax;

else

Filtered(OutputIterator+1)=0;

end

FrontIterator = FrontIterator+1;

OutputIterator = OutputIterator+1;

end

forFrontIterator=FrontIterator:1:length(Original)-1

if Original(FrontIterator+1)>WinMax

WinMax=Original(FrontIterator+1);

WinMaxPos=WinSizeSpec;

else

WinMaxPos=WinMaxPos-1;

ifWinMaxPos< 0

WinIterator = FrontIterator-WinSizeSpec;

WinMax = Original(WinIterator+1);

WinMaxPos = 0;

WinPos=0;

forWinIterator = WinIterator:1:FrontIterator

if Original(WinIterator+1)>WinMax

WinMax = Original(WinIterator+1);

WinMaxPos = WinPos;

end

WinPos=WinPos+1;

end

end

end

ifWinMaxPos==WinHalfSize

Filtered(OutputIterator+1)=WinMax;

else

Filtered(OutputIterator+1)=0;

end

OutputIterator=OutputIterator+1;

end

WinIterator = WinIterator-1;

WinMaxPos = WinMaxPos-1;

forLengthCounter=1:1:WinHalfSizePlus-1

ifWinMaxPos<0

WinIterator=length(Original)-WinSize+LengthCounter;

WinMax=Original(WinIterator+1);

WinMaxPos=0;

WinPos=1;

forWinIterator=WinIterator+1:1:length(Original)-1

if Original(WinIterator+1)>WinMax

WinMax=Original(WinIterator+1);

WinMaxPos=WinPos;

end

WinPos=WinPos+1;

end

end

ifWinMaxPos==WinHalfSize

Filtered(OutputIterator+1)=WinMax;

else

Filtered(OutputIterator+1)=0;

end

FrontIterator=FrontIterator-1;

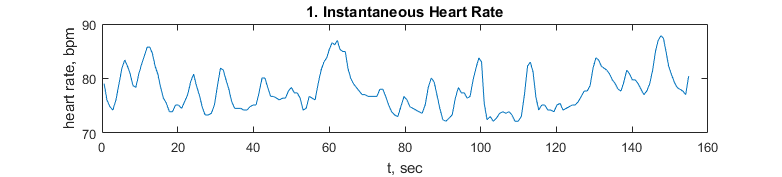
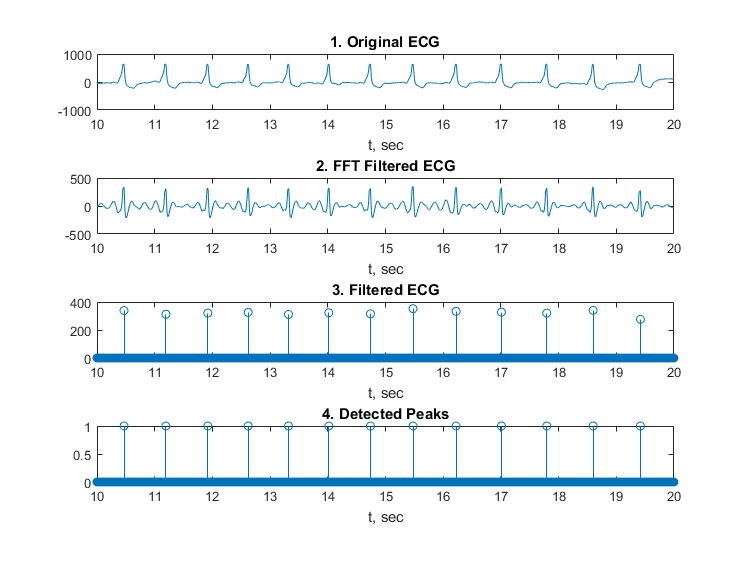
WinMaxPos=WinMaxPos-1;

OutputIterator=OutputIterator+1;

end

Результати:

Average Heart Rate = 78.1681



*Висновки*:

В результаті виконання лабораторної роботи було розроблено алгоритмобчислення ЧСС з використанням першого відведення ЕКГ. Було передбачено можливість підгонки деяких параметрів (% Variables) з метою досягнення найкращого рівня детекції R піків для кожного конкретного сигналу. Алгоритм показав себе як відносно стійкий до артефактів (за певних параметрів “чутливості” мною не було помічено жодних невідповідностей результатів виконання програми вхідному сигналу навіть на сильно зашумлених ділянках).