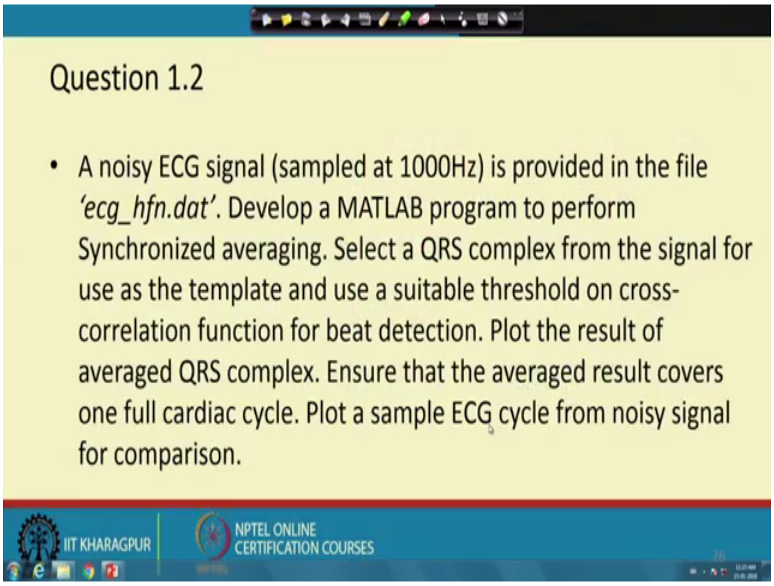
Biomedical Signal Processing
Prof. Sudipta Mukhopadhyay
Department of Electrical and Electronics Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 48
Tutorial - I (Contd.)

So, we are going further the next problem.

(Refer Slide Time: 00:21)



Question 1.2

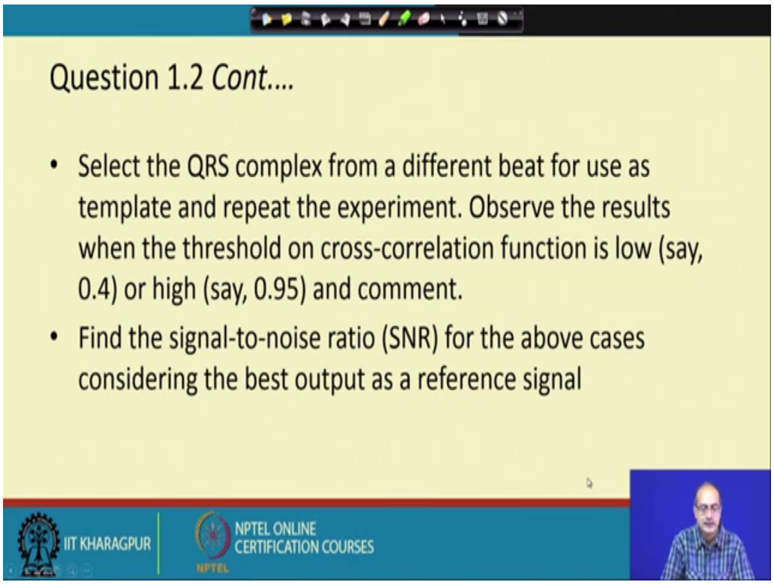
- A noisy ECG signal (sampled at 1000Hz) is provided in the file *'ecg_hfn.dat'*. Develop a MATLAB program to perform Synchronized averaging. Select a QRS complex from the signal for use as the template and use a suitable threshold on cross-correlation function for beat detection. Plot the result of averaged QRS complex. Ensure that the averaged result covers one full cardiac cycle. Plot a sample ECG cycle from noisy signal for comparison.

And here the problem is that again, we take a ECG signal sample that much higher frequency that is 1 kilohertz and this signal as the name suggest ecg underscore hfn dot dat the name suggest itself that it is corrupted with high frequency noise.

So, we have to perform synchronized averaging. So, for that we have to select the QRS complex for the signal and we need to need that that what is the span of one cycle first for the purpose of taking the that period for averaging and also the QRS complex will help us or the beat location will help us to along actually those cycles.

So, we need to perform from that exercise and to find out the width, we are asked to use correlation for that and we are also asked that we should be careful that it should not be part of the cycle, it should take a full cycle and with respect to that that noisy signal, if we put that; then we can appreciate that what is the change it brings.

(Refer Slide Time: 01:55)



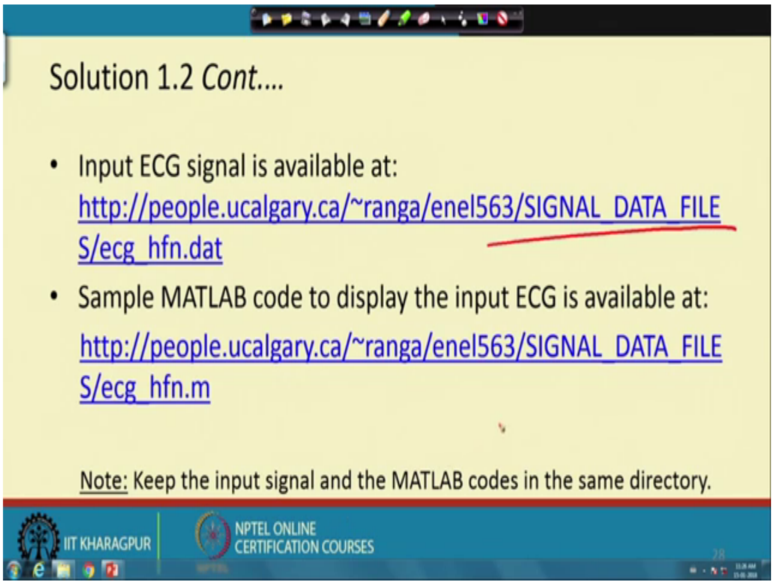
Question 1.2 *Cont....*

- Select the QRS complex from a different beat for use as template and repeat the experiment. Observe the results when the threshold on cross-correlation function is low (say, 0.4) or high (say, 0.95) and comment.
- Find the signal-to-noise ratio (SNR) for the above cases considering the best output as a reference signal

Next is that; as you continue select the QRS complex from a different beat for use as a template and repeat this experiment observe the results when the threshold on cross correlation function is low, there is 0.4 or high and comment; that means, after taking the cross correlation the beat location is actually find out by taking actually a threshold that where the correlation is high.

Now, how high is be taken as high that is the questions we are asking and if we change that threshold how that will impact the result ok. So, that is the second part next is that find that the signal to noise ratio of the above case considering the best output as the reference signal ok. So, it is like the previous one.

(Refer Slide Time: 03:08)



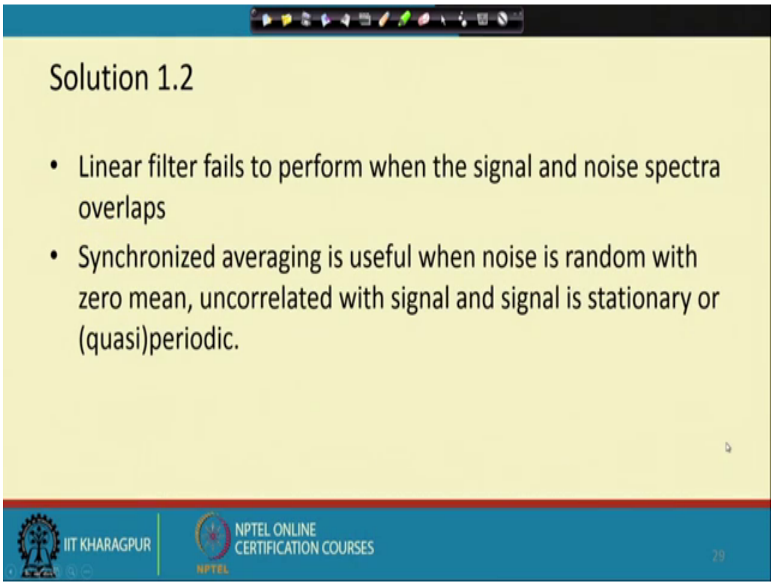
Solution 1.2 *Cont....*

- Input ECG signal is available at: http://people.ucalgary.ca/~ranga/enel563/SIGNAL_DATA_FILES/ecg_hfn.dat
- Sample MATLAB code to display the input ECG is available at: http://people.ucalgary.ca/~ranga/enel563/SIGNAL_DATA_FILES/ecg_hfn.m

Note: Keep the input signal and the MATLAB codes in the same directory.

Now, let us look at that how to get the data that here the link is given that how we can get the signal and the data file, it is given here and the corresponding that; MATLAB file is also given here dot mat and again we need to keep in mind that we should put the both the data at the code in the same file or in the working directory we can say directly. So, we know these things from the previous equipment and from there we proceed.

(Refer Slide Time: 03:55)



Solution 1.2

- Linear filter fails to perform when the signal and noise spectra overlaps
- Synchronized averaging is useful when noise is random with zero mean, uncorrelated with signal and signal is stationary or (quasi)periodic.

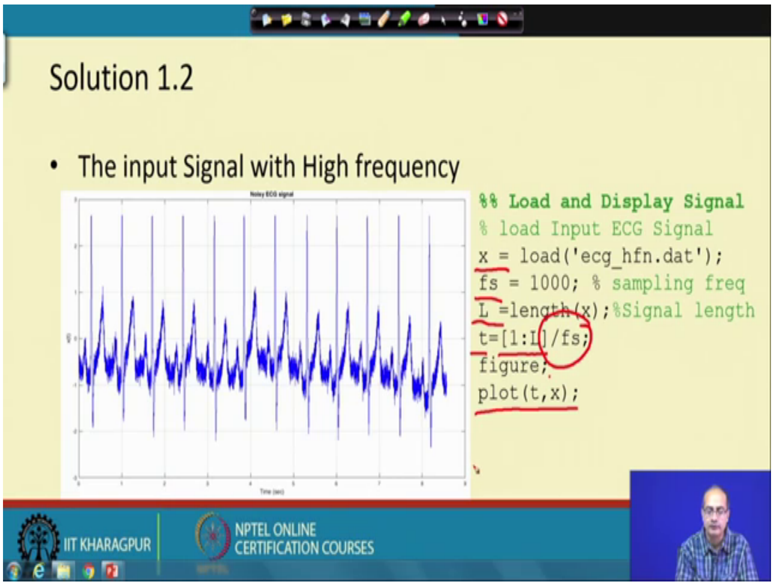
Now, what is the motivation to go for the synchronized averaging, let us look at that we know the linear filtering, it cannot; what will that when the signal and the noise spectra they have overlaps, if the signal or noise, they are separated or have a very small overlap.

For example, if we take the power frequency noise, it is concentrated on a particular frequency. So, what we can do if we can just eliminate all the strength at that particular frequency? We eliminate the noise completely, by that; we sacrifice some part of the signal strength also at that particular frequency, but using a filter like notch filter, we minimise that loss that; that means, one particular frequency only, we are losing that signal strength and if we can sacrifice that we can get rid of that, but when that is actually spread over a band and both the signal and noise have common actually spread of or common range of frequencies, then if we try to eliminate all the frequencies, then we have a big loss in terms of the signal also.

So, in that case, synchronized averaging could be a better way and it is specially successful when the noise is random with 0 mean and it has to be a uncorrelated with the signal and signal has to be stationary or periodic or at least quasi periodic and in this case our signal ECG is a quasi periodic signal.

So, we can go for that synchronous averaging or synchronized averaging.

(Refer Slide Time: 06:09)



Solution 1.2

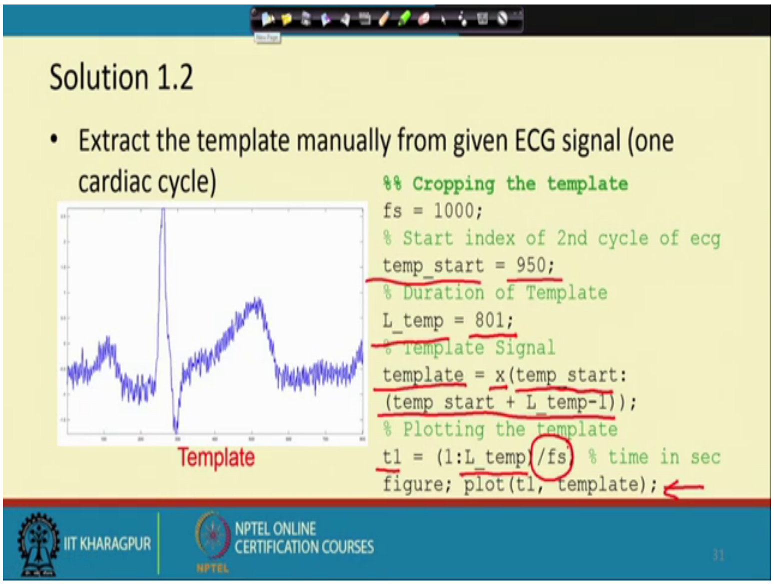
- The input Signal with High frequency

```
%% Load and Display Signal
% load Input ECG Signal
x = load('ecg_hfn.dat');
fs = 1000; % sampling freq
L =length(x);%Signal length
t=[1:L]/fs;
figure;
plot(t,x);
```

So, first we look at the signal corrupted with high frequency noise. So, here is the code for that that we load the single that here in the variable x sampling frequency is given 1000. So, we assigned the variable fs calculate the number of samples in the data that is using the comment length and then we prepare the time axis that is first, we write that ram that it is increasing in terms of the sample number 1 to L and multiply with 1 by fs that is giving us the that sampling interval and at the end of it, we issue the figure command to create the pen and plot using the command plot t comma x.

So, here we see the signal, it is having the same span between 8 to 9 second, we have the signal and the amplitude is in negative side maybe that minus 2.5 or even more than that and above also it is above 2.5. So, it is limited within that span ok.

(Refer Slide Time: 07:47)



Solution 1.2

- Extract the template manually from given ECG signal (one cardiac cycle)

```
%% Cropping the template
fs = 1000;
% Start index of 2nd cycle of ecg
temp_start = 950;
% Duration of Template
L_temp = 801;
% Template Signal
template = x(temp_start:
(temp_start + L_temp-1));
% Plotting the template
t1 = (1:L_temp)/fs % time in sec
figure; plot(t1, template);
```

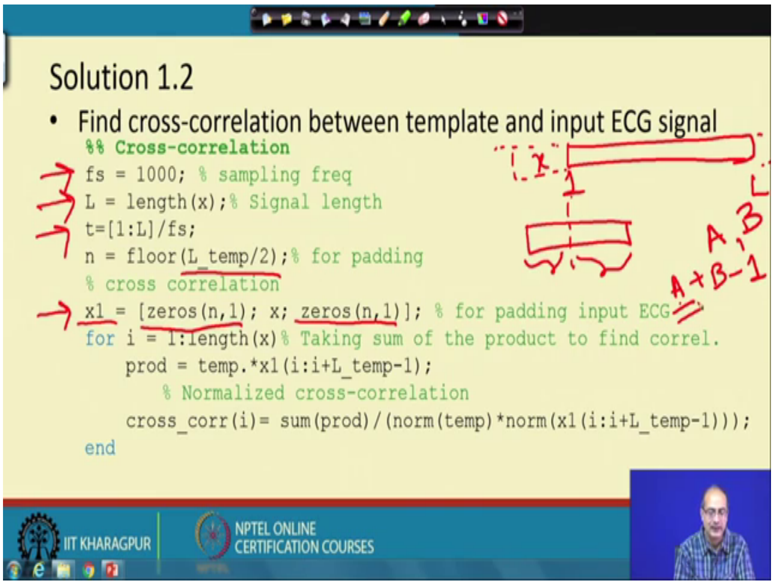
So, we have the signal now. So, next job is to find out that the template and look at it. So, here we have extracted one cycle manually one cardiac cycle and for that what we have done we have done few visual actually examination and we have taken a starting point for selecting the template that temp start, we have taken that point that starting point as 950 that is manually selected again the duration is also manually selected we have taken that as 801 that we have taken a odd number to make the window or the duration that odd and that we have assigned it to a variable that L underscore temp. So, that gives the duration of the template.

So, the template; now it is copied, there the variable or the vector template that which will keep that part of the signal x starting point from temp underscore, start there is the starting point and till the end point of the duration and then we can actually plot that signal now we create the time axis that is t 1 having the length that L underscore temp and for this RAM, we are again multiplying with 1 by fs to get the time axis and using the figure command, we have created a new pen and we have use the plot command to plot it here ok.

And let us see that what is the output here we get the output. Now we can appreciate the high frequency noise much better it has a good amount of noise, but this is good enough to see that not only that what is the amount of noise that in the ECG signal the most of the energy of the signal is concentrated near the QRS complex and rest of it is there in P and T in between the T; that means, end of t to the beginning of p in between there is supposed to be isoelectric line or no electrical activity. So, whatever is there is noise.

So, if we take a template, even if not as long as one cycle, if we just capture that P, QRS and T that is good enough for this purpose of finding out the location of the beat and that is the fact we have used here otherwise technically speaking we should have taken a little more, but increasing that we take some more samples which have more of noise. So, that does not add much value ok.

(Refer Slide Time: 11:29)



Solution 1.2

- Find cross-correlation between template and input ECG signal

```
%% Cross-correlation
fs = 1000; % sampling freq
L = length(x);% Signal length
t=[1:L]/fs;
n = floor(L_temp/2);% for padding
% cross correlation
x1 = [zeros(n,1); x; zeros(n,1)]; % for padding input ECG
for i = 1:length(x)% Taking sum of the product to find correl.
    prod = temp.*x1(i:i+L_temp-1);
        % Normalized cross-correlation
    cross_corr(i)= sum(prod)/(norm(temp)*norm(x1(i:i+L_temp-1)));
end
```

So, now let us proceed for the cross correlation what we do. So, first again we recall that fs is our sampling frequency here then length of the signal is L, we recall those things then time axis is T, we know all of them. Now, we come to a new thing that when we are computing the that correlation, actually, we need to understand that what we have we have the variable x; it has a duration ok. So, this may be the data left starting from from 1 to L.

We have a taken a template. So, template is having some length smaller than the x, but it has some length. Now the middle of it we take as the centre of it, we have taken as the index of that template and with a slide this template from the beginning to the end ok. Now in that way; what we have to do every position, we need to take the dot product of the signal and the template and sum them up to calculate the correlation.

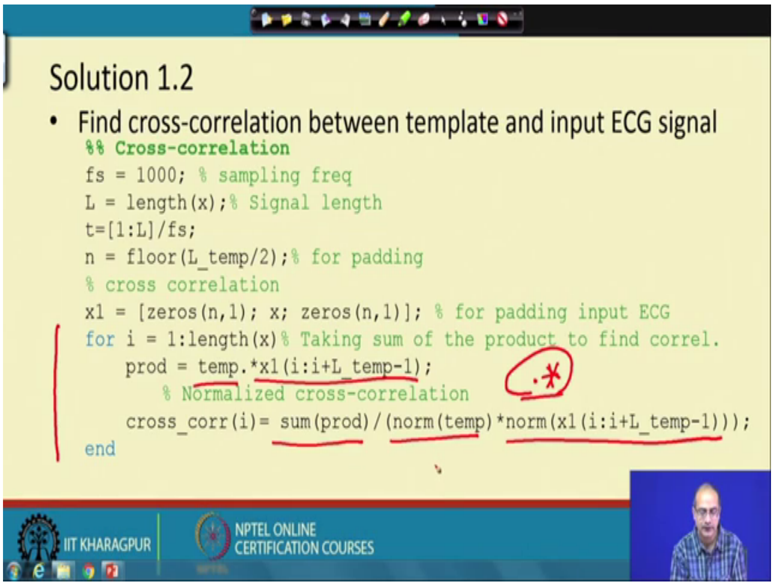
Now, for the part that when we are at the first part these portion actually that we have to take the multiplication and there is no value. In fact, nothing is assigned there. So, if we try to compute it, it will throw up error because x is not defined same way when we go to the other end beyond L, when it will look for actually the values to multiply again, we have the same problem. So, we need to pad actually the signal x from both the sides and the padding should be almost half of the length of the template; that means, actually this part that what would be outside when we start at the beginning or what would be outside when we go for at the end ok.

So, how we get that length we can simply divide the that the template length by 2. So, that is what we have done here and we have taken the floor function because it is a odd number; how many samples should be extra, it cannot be fraction. So, it would be just the lower integer. So, we have taken the floor function to get that and next, we have created the padded variable x 1 where at the beginning we have the zeros and at the end also we have 0 and here one small thing just to note that we have taken the x is a column vector and so, we have taken this things also are that column vectors. So, they are adding up in this way and the resultant is also a column vector.

Now, the length of it is actually the length of the signal x plus length of the template minus one how we are getting minus one because we take the floor operation here ok. So, that is how we get and if you recall your books then you would recall that if I take the correlation of two signal one is of length a another is of length b that the number of times we get a plus b minus 1 ok. So, that much here we have to create the length of the padded vector that is what we have done, but we are interested only the correlation value of the a point and we will just keep track of that ok.

So, now in this loop, we are just doing that for the length of the signal x, we are computing that. So, we are taking the further product that is dot product of the template and the signal and for the dot product please look at that symbol brought into star ok.

(Refer Slide Time: 16:35)



Solution 1.2

- Find cross-correlation between template and input ECG signal

```
%% Cross-correlation
fs = 1000; % sampling freq
L = length(x);% Signal length
t=[1:L]/fs;
n = floor(L_temp/2);% for padding
% cross correlation
x1 = [zeros(n,1); x; zeros(n,1)]; % for padding input ECG
for i = 1:length(x)% Taking sum of the product to find correl.
    prod = temp.*x1(i:i+L_temp-1);
        % Normalized cross-correlation
    cross_corr(i)= sum(prod)/(norm(temp)*norm(x1(i:i+L_temp-1)));
end
```

.*

So, if we have given star, it would have been multiplication of two vectors. So, you need to check the dimensionality that whether that we can take the product in a meaningful way.

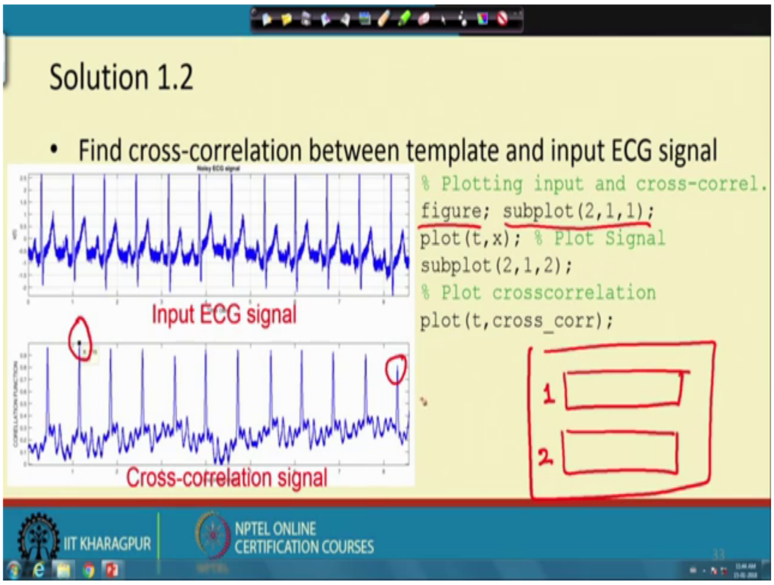
But as long as the length of the two matches, we can take the point wise product and we need not have to bother about that whether the dimensionality; that means, that number of rows and number of columns they are in agreement or not ok, only the length should be same that is good enough ok. So, there is the symbol for the dot product.

Next, what we do we normalise that thing that we need to take the sum of the these all the product terms and normalise it with the energy of the template and energy of the signal of that portion ok. Now what the norm is doing, it is actually calculating the L 2 node. So, what it is doing? It is first taking the we can say again the dot product of the signal; that means, temp dot star temp, then we take the sum of it and then we take the square root of it and we get the norm temp ok.

The same thing we are doing for that segment of the signal of the length that we are the after same length as a that the template ok. So, that is how we get the energy and we are dividing or normalising with that the energy or the I would say that rather standard division or square root of energy of the template and the signal at that portion the reason is that here that the sum of product otherwise its it would be affected by the scaling of the signal or the template.

So, to make it invariant, we should take the that normalisation and instead of correlation we are getting correlation coefficient in that way that will help us to judge that whether the correlation is high or low. Otherwise any scaling of the that template would actually change the correlation values. So, we simply want to actually eliminate that problem and make it scaling variant.

(Refer Slide Time: 19:23)



Solution 1.2

- Find cross-correlation between template and input ECG signal

```
% Plotting input and cross-correl.
figure; subplot(2,1,1);
plot(t,x); % Plot Signal
subplot(2,1,2);
% Plot crosscorrelation
plot(t,cross_corr);
```

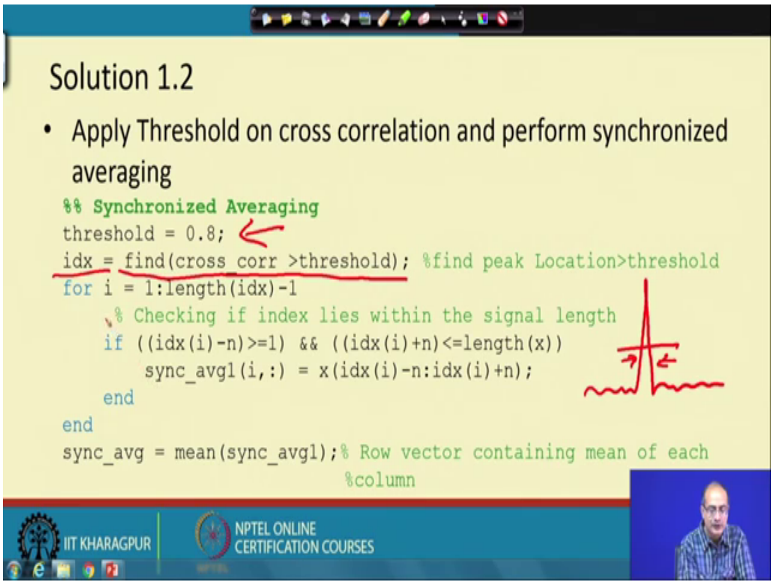
Now, here we find that the correlation between the template and the ECG signal ok. So, for that; we are showing that the input and the cross correlation. So, first we have taken the figure, then again use the subplot command the intention is to divide the pen into 2 parts that is 2 strips of rows ok, like this the full image is divided into two parts the top part we are plotting the signal that is plot that tx in the first one and the second one is plotting the cross correlation or correlation coefficient ok.

So, here we get that it looks like very similar to the ECG signal itself; the cross correlation signal or the cross correlation coefficient is very near to the ECG signal only thing because we have normalised the values are between 0 to 1 and apart from the QRS complex, there are some undulations. So, what actually we can infer from that that maximum match is there when it is matching with the QRS complex because maximum energy is present there and the moment it gets decided, there is a sudden change and that gives rise to high value of the that that correlation at some point which gives rise to some feeling that we have got the are we back ok.

However when we are telling that it is high, if we look at that span from 0 to 8.5 second, the maximum is here where we have taken the from where we have taken the template for the other cycle the maximum is not that high, but most of the cases they are above 0.9, but let us look at the last one; it has come below much below the 0.9 it is for averages little above the 0.8 here ok.

So, the threshold also can have some effect ok. So, that is the thing we understand and let us proceed for that how we can do the synchronized averaging.

(Refer Slide Time: 22:15)



Solution 1.2

- Apply Threshold on cross correlation and perform synchronized averaging

```
%% Synchronized Averaging
threshold = 0.8;
idx = find(cross_corr >threshold); %find peak Location>threshold
for i = 1:length(idx)-1
    % Checking if index lies within the signal length
    if ((idx(i)-n)>=1) && ((idx(i)+n)<=length(x))
        sync_avg1(i,:) = x(idx(i)-n:idx(i)+n);
    end
end
sync_avg = mean(sync_avg1);% Row vector containing mean of each
                                    %column
```

Now, we pass from the threshold and first point is the we have taken the threshold starting from 0.8 ok, we found that it is to be a good value and from there that what we have taken we have taken those points we are interested in those points that where the cross correlation is more than the threshold ok.

Now, I would request you to visualise the cross correlation graph. So, it looks like undulation and then high and then again undulations are there. So, once we have taken taking a threshold. So, above that threshold, actually, we would not get just one point, but all the points within this pad within this span all the points should come. So, all those indexes will be stored here ok.

Now, what we are doing that we are taking that passing through that indexes and to find out the length that we have to do the that synchronous averaging and we need to make sure that while doing the averaging values are defined ok. So, we are starting from one to length of the index minus 1. So, for each of these occurrence you take one width is there and we have to take that average of that length and for that as we are taking the average or the sum; first, we would like to take those parts as a separate vectors and then we will do the averaging the first we pick up those part.

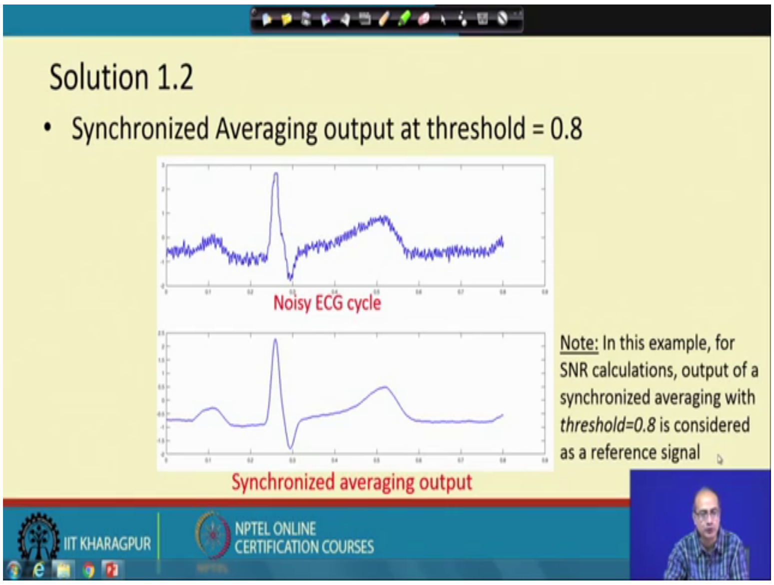
So, how we pick up that part that we are taking from one index that indexes minus n and index that xi plus n please keep in mind this n is the value of the half of the template ok. So, we need to keep that in mind otherwise it will look like a misery piece of code that this is the half of the length of the template. So, we are actually copying same as the length of the template that part of the signal keeping that index where it is crossing the signal value; that means, the big location in mind centring the big location we are taking that value ok.

And while doing that another thing we need to keep in mind as we are taking both the side some part is going towards the left some part is going towards the right of that location ind xi. So, the left part should not be less than 1 because if it is less than 1, then that would be an error the program will crash and same way it will program will crash if we try to pick up some value beyond the length of the vector. So, that also these two things we need to check and in that way we pick up all the points or all those vectors that how many copies of the signal, we get and in this process, we may actually means the last or the first bit if there is not sufficient number of points ok. So, that could be the penalty, but at least we would not have any crash in the program.

And once we have all those that vectors we now can take the mean; now these mean it is not a single value, it is the mean of the vectors of two dimensional vectors and so, this itself would be a row vector and that will give us the synchronized average value that row ok. So, that is the way, we have been successful in computing the that synchronized average that is the code.

Now, let us proceed to see that how we get these values.

(Refer Slide Time: 27:37)



Solution 1.2

- Synchronized Averaging output at threshold = 0.8

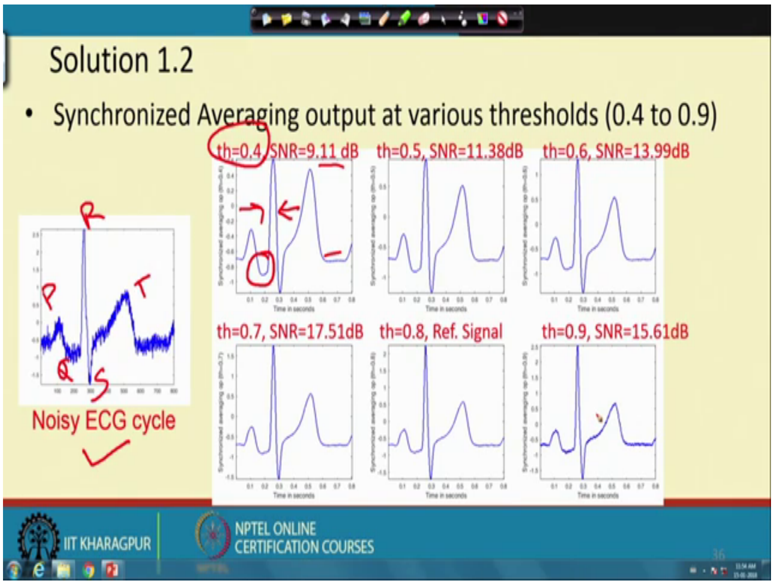
Noisy ECG cycle

Synchronized averaging output

Note: In this example, for SNR calculations, output of a synchronized averaging with *threshold=0.8* is considered as a reference signal

So, here for 0.8, we are plotting that the noisy signal first at the top, here is the noisy signal one cycle, typically, we have taken the template and below we are showing the that synchronized average output it looks so clean and as it looks very near to the signal, we have taken these as a reference see that all the high frequency noise of the undulations due to that has gone ok. So, we take that as the reference signal for threshold equal to 0.8 and with respect to now we judge that how the other thresholds will do.

(Refer Slide Time: 28:44)



Solution 1.2

- Synchronized Averaging output at various thresholds (0.4 to 0.9)

th=0.4, SNR=9.11 dB

th=0.5, SNR=11.38dB

th=0.6, SNR=13.99dB

th=0.7, SNR=17.51dB

th=0.8, Ref. Signal

th=0.9, SNR=15.61dB

Noisy ECG cycle

So, now what we do here in the left hand side; we are showing again the noisy ECG we have shown it in a small way. So, that we can see the other waveforms for different value of the threshold and we can see them together in the same place; the first we start with threshold 0.4 and with respect to the reference, we get it is 9.11 dB. So, the SNR is not that good.

Now, what is the reason for that? If you look at the signal the signals in all these cases may be here we can get some noise, but up to 0.8, all the cases they are very smooth. So, all of them they are successful in suppressing the noise, but what has gone here it seems the shape of the that the signal has got distorted ok.

How the shape has changed if we look at this portion that point Q, say here actually this is the Q, this is R, this is S, this is P, this is T. So, Q point; if you look at; it has become very white whatever happened that these peak QRS peak has become also white ok, in this case and T wave look at that that compared to the T wave in the original signal, it has become much higher ok.

So, it has distorted to a good extent and that is why SNR is solo now threshold when we take 0.5 the situation has improved, but still all those problems are there, 0.6 again somewhat better 0.7, even better 0.8 is the best and 0.9 when we are taking, then we see that again the noise is reappearing. In fact, that if we can magnify the signal, you can get the feeling that 0.7 onward to every noise is where we trying to reappear kind of thing and therefore, sure visible in 0.9.

Now, why this thing is happening to understand that we need to go back a little we need to go back to the that place where we have that the cross correlation signal. So, in this place; now if I take a threshold say at 0.4, what will happen that here and number of points actually will appear above the threshold, it is not just one sample for synchronized averaging will be successful, if we have one sample; that means, one interval we are taking, but in this case will have actually multiple samples above the threshold. So, we are taking actually copies which are little desired ok. So, number of copies we are taking. So, the effect of the summing up of them would be actually smearing that signal.

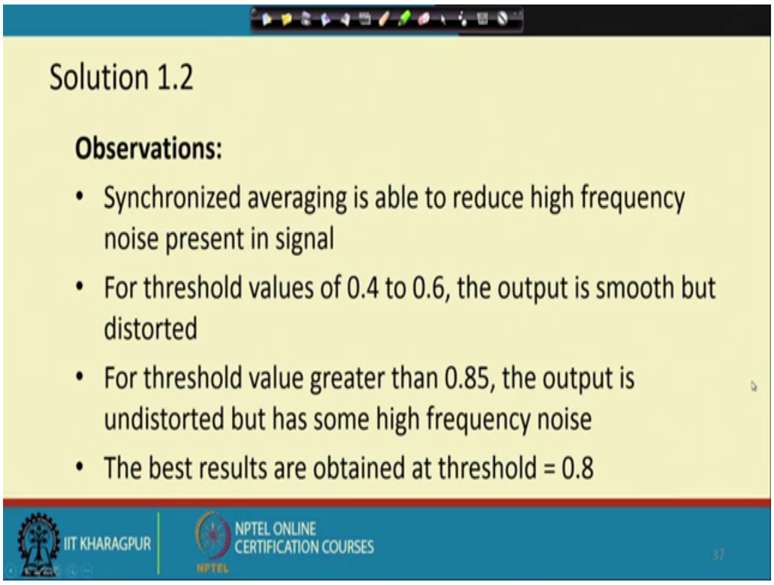
Now, as we have more and more copies suppression of the noise or averaging would be better leaving to better suppression of the noise, but at the same time because of the smearing of the signal the width of the that peaks would increase, it will become even more smooth. So, that is the challenge and some of the peaks here the noise peaks which are just above the threshold that which are no alignment is there that they will also come. So, some spurious peaks will come. So, that will distort the signal further.

Now, as we move it up as we move it up this threshold when we go above say 0.9, what is happening say when we are going there then some of the peaks this peak this peak this peak, probably, this peak also some of the peak would be missing and as the peaks are missing first of all that the averaging will suffer. So, the noise actually reappearing there though the signal morphology is not changed in that case. So, there is a contradiction if we take the threshold very high.

Then; what will happen we have insufficient averaging because the number of beats we could catch is less the noise suppression would not be complete and when we push it very low, then the spurious peaks are coming and they are disturbing the signal morphology changing that. So, we need to take something in between.

The ideal point would be that if we can get one copy of each cycle and 0.8 could actually be close to that. So, that is the way we have got actually with 0.8, the giving the best signal and we have taken that reference and we have use the PSNR or the in this case SNR, actually to get that thing to quantify that changes that for 0.4, it is starting from 9. So, it has increased to 17.5 for 0.7 as the threshold and when it going for threshold 0.9, it has very slight increase ok; that means, the morphology has become better, but then noise is coming which is counter balancing the gain ok.

(Refer Slide Time: 36:09)



Solution 1.2

Observations:

- Synchronized averaging is able to reduce high frequency noise present in signal
- For threshold values of 0.4 to 0.6, the output is smooth but distorted
- For threshold value greater than 0.85, the output is undistorted but has some high frequency noise
- The best results are obtained at threshold = 0.8

So, that is the thing we learn here which is summarised at the end that synchronous averaging, it is a very good technique to reduce the high frequency noise present in the signal and as we change the threshold value from 0.4 to 0.6; the output becomes more and more smooth, but they are distorted ok; that means, low threshold it gives good suppression, but distorted signal.

Threshold value again above 0.85; what we see that it gives undistorted waveform, but some high frequency noise are seems to be present because the averaging is not complete or there are less number of templates to take average. So, the best result is obtained at threshold 0.8 and with that we conclude this problem number 2 in the tutorial number 1.

Thank you.