Digital Voice and Picture Communication

Prof. S. Sengupta

Department of Electronics and Communication Engineering

Indian Institute of Technology, Kharagpur

Lecture - 14

Linear Predictive Synthesizer

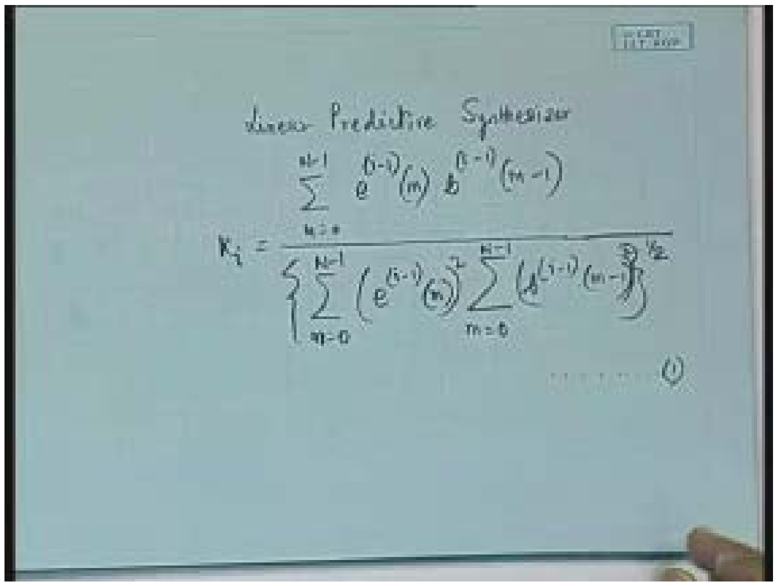
We will be discussing about the linear predictive synthesizer and also we will be discussing about certain bi-products of LPC like the LPC can be used to improve the pitch detection performance. The pitch detection approach we have already discussed; that is the time when we learnt about the autocorrelation approaches etc but then we also said that the pitch detection is always is not very robust because of the presence of the formants and the formants tend to introduce some extra harmonics and that often makes the pitch detection process difficult when we are using the normal autocorrelation method. But we can still try to continue to use the autocorrelation method. But if we are using the LPC parameter to flatten the spectrum and all that those also we will be studying. So it is not only the linear predictive synthesizer but also those aspects.

Another point that we need to discuss is that, towards the last class there were some doubts which were expressed about the k i's; in the lattice formulation what we had discussed in the last class we need to discuss we need to throw some more light on the computation of those k i's because it was very rightly asked that if we have to do the computation of the autocorrelation in order to find the k i's in that case what is it that we are gaining computationally or is there any alternative way of calculating the k i's using the lattice structure itself.

Now if you see the lattice structure you see that the lattice structure basically obtains the forward prediction error and the backward prediction error and this we are obtaining recursively. So it is that recursive structure that proceeds in our lattice filter structure so it is not directly obtaining the alpha k's; although the main objective of any LPC analysis is to obtain the alpha k's, there we are getting the error terms that is just to say the forward prediction error and the backward prediction error that is what we are getting. But what one can ask is that, what is the direct relationship between the prediction errors and the alpha k's. There is obviously a relationship. And in fact we just repeat what we said in the last class that basically the lattice formulation is nothing but the realization of Durbin's algorithm.

We have been using the Durbin's autocorrelation method in order to formulate the lattice structure so whatever equations we discussed in connection with Durbin's algorithm that holds good. So the k i's that we are talking of is still the Durbin's algorithms k i but there is a different methodology to compute the k i's. And in fact it can be shown that the k i's instead of writing in terms of the autocorrelation like writing the k i's in terms of the r's that is what we did in case of the Durbin's algorithm; instead there could be an alternative formulation to obtain the k i's from the errors, from the forward prediction error and the backward prediction error. And in terms of the prediction errors one can write the k i's like this that k i can be expressed as: in the numerator we have m equal to 0 to N minus 1 the backward prediction error that is the forward prediction error that is e (i minus 1) (m); (m) is the index of the summation and (i) as you know is the recursion step that is what we are having and this into b of i minus 1 (m minus 1) this is the backward prediction error and in the denominator term there is a normalized for the normalization we have a denominator term that goes like this: summation m is equal to 0 to capital N minus 1 e of i minus 1 (m) whole square into summation m is equal to 0 to N minus 1 into b (i minus 1) (m minus 1) and this term square sorry here there will be a square term in this case. So this is a square term and this is also another square terms so within the curly bracket what we write is summation of two squared quantities and products of that and this whole thing would be to the power half.

(Refer Slide Time: 7:07)



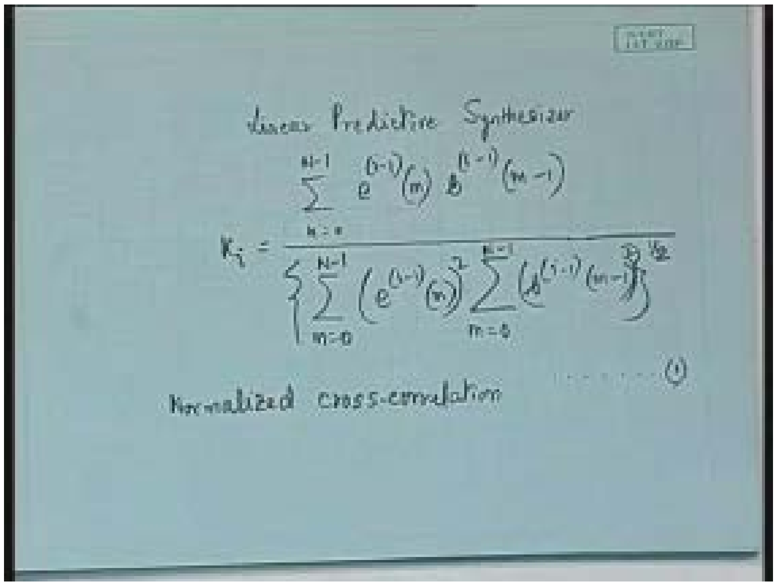
Linear Predictive Synthesizer

$$k_i = \frac{\sum_{m=0}^{N-1} e^{(i-1)}(m)\, b^{(i-1)}(m-1)}{\left\{\sum_{m=0}^{N-1} \left(e^{(i-1)}(m)\right)^2 \sum_{m=0}^{N-1} \left(b^{(i-1)}(m-1)\right)^2\right\}^{1/2}} \quad \ldots\ldots (1)$$

This let us call in terms of today’s equations let us call that as equation number 1. So just look at the expression for k i very carefully. What we have done is basically a cross correlation of the two errors: the forward prediction error and the backward prediction error they are cross correlation. In fact the cross correlation will be given by this quantity what is there in the numerator and the denominator is doing a normalization so that what we are effectively getting is a normalized cross correlation between the forward and backward prediction error. So this is nothing but the normalized cross correlation.

And in fact the k i’s like this because it is expressed in terms of this correlation these are referred to as the partial correlation coefficient. You see that if I have k 1 from here in that case I will be needing e (0) and b (0); all the e (0) and b (0) errors will be required that means to say the first order prediction errors or rather to say the zeroth order prediction errors would be needed and then when we use k is equal to 2 then it will be e (1) b (1) those error terms would be needed and we will be getting the correlation of those errors. So it is basically a partial correlation that is what we are computing that first the k 1 and then doing the k 2 computation, then doing the k 3 computation like that.

(Refer Slide Time: 9:29)



© CET I.I.T. KGP

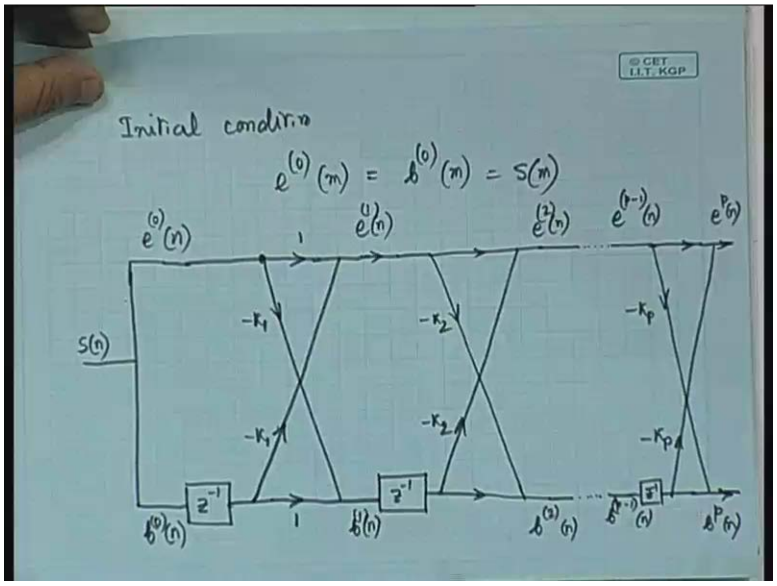
Linear Predictive Synthesizer

$$K_i = \frac{\sum\limits_{m=0}^{N-1} e^{(i-1)}(m)\, b^{(i-1)}(m-1)}{\left\{\sum\limits_{m=0}^{N-1} \left(e^{(i-1)}(m)\right)^2 \sum\limits_{m=0}^{N-1} \left(b^{(i-1)}(m-1)\right)^2\right\}^{1/2}} \quad \cdots\cdots (1)$$

Normalized cross-correlation

Therefore, again just remember our lattice structure. In the lattice structure we require first the k 1. In fact let us have a look at this from our earlier notes, we use the lattice.

(Refer Slide Time: 09:56 min)



© CET I.I.T. KGP

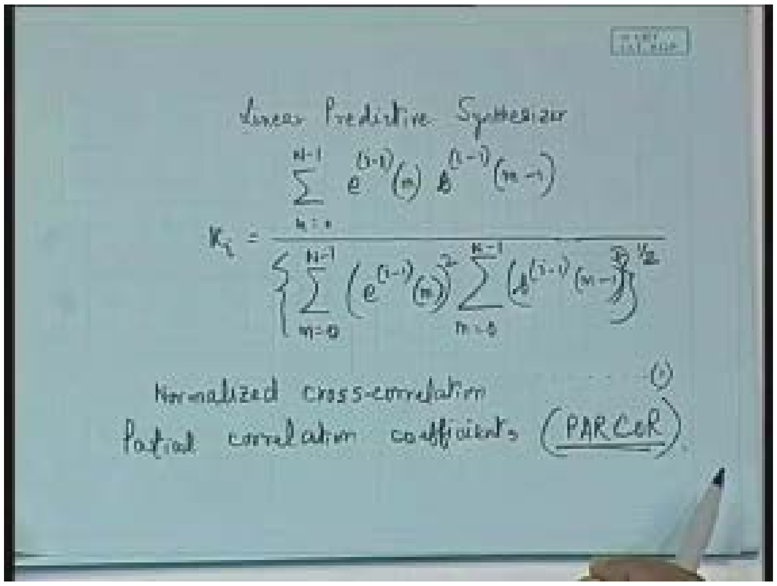
Initial condition

$$e^{(0)}(m) = b^{(0)}(m) = s(m)$$

Just this diagram; you see e (0) and b (0) we already have the e (0) and b (0) and then we can compute the k 1 that is from this expression because k 1 needs what; it requires e (0) (m) it requires b (0) (m minus 1) and we already have that. So using this e (0) and b (0) and this delay element it is possible for us to calculate the k 1 and once the k 1s are obtained in that case we will be able to compute the e (1) (n) and b (1) (n) and once the e (1) (n) b (1) (n) these delays they are given in that case it will be possible for us to compute k (2) (n).

Therefore, essentially this k (1) (n) this k 1 k 2 k three all these k i's they also can be computed using this recursive step of the lattice filter. In fact what we have to do is that ultimately because we have to relate everything with the alphas now once the k i's are determined obtaining the alphas is a simple step. IOn fact there we can follow Durbin's approach because as you know that as per Durbin we will be having the following recursions that it will be alpha (i) ith step that will be equal to k i and alpha (j) ith step would be equal to alpha j (i minus 1)th step minus k i into alpha i minus j of (i minus 1)th step. This we have to do for j varying from 1 to i minus 1. This we already know and these k i's they are called as partial correlation coefficients; these k i's they are called as partial correlation coefficients or as PARCOR.

(Refer Slide Time: 12:24)



Linear Predictive Synthesizer

$$k_i = \frac{\sum_{m=0}^{N-1} e^{(i-1)}(m)\, b^{(i-1)}(m-1)}{\left\{ \sum_{m=0}^{N-1} \left(e^{(i-1)}(m)\right)^2 \sum_{m=0}^{N-1} \left(b^{(i-1)}(m-1)\right)^2 \right\}^{1/2}} \quad (1)$$

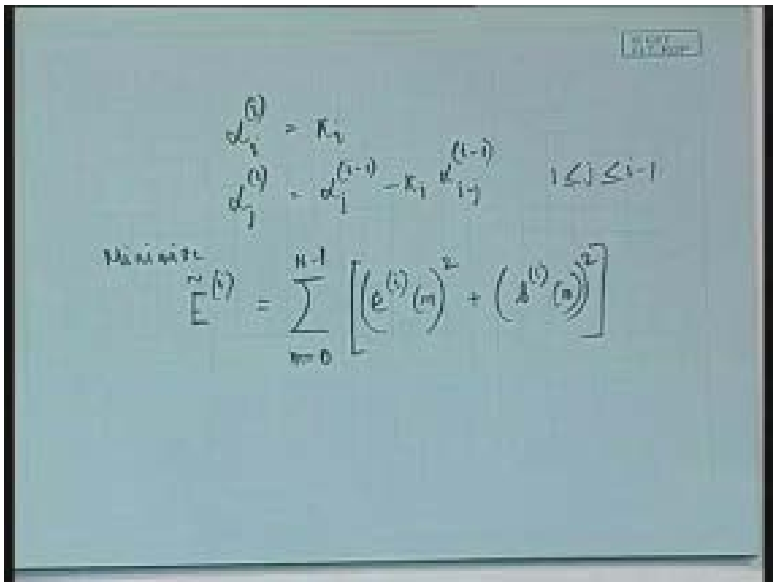
Normalized cross-correlation

Partial correlation coefficients (PARCOR)

Now there is an now this basically gives an alternative; this PARCOR is giving us an alternative way of computing the matrix inversion. That is quite obvious. And now there is a different way of specifying the mean square error. Because after all how did we obtain these coefficients, these k i’s?

These k i’s are ultimately obtained by the minimization expression of the prediction errors. So by minimizing the prediction errors we had obtained this. But the prediction errors could be defined in a different way and that is what Burg had done that Burg had defined the prediction error............ the minimization that we are required to perform that is on the mean squared forward and backward prediction errors. So, given that, one can write down E and we write it as E tilde for the (i) ith iteration and that is equal to summation m is equal to 0 to N minus 1 and then we define it as e (i) (m) this term square plus b (i) (m) this term square. So sum of the forward prediction squared and the backward prediction squared and summed up from m is equal to 0 to N minus 1; this is our new definition of the prediction error and we have to minimize this.

(Refer Slide Time: 14:38)



$$\alpha_i^{(i)} = K_i$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - K_i\,\alpha_{i-j}^{(i-1)} \qquad 1 \le j \le i-1$$

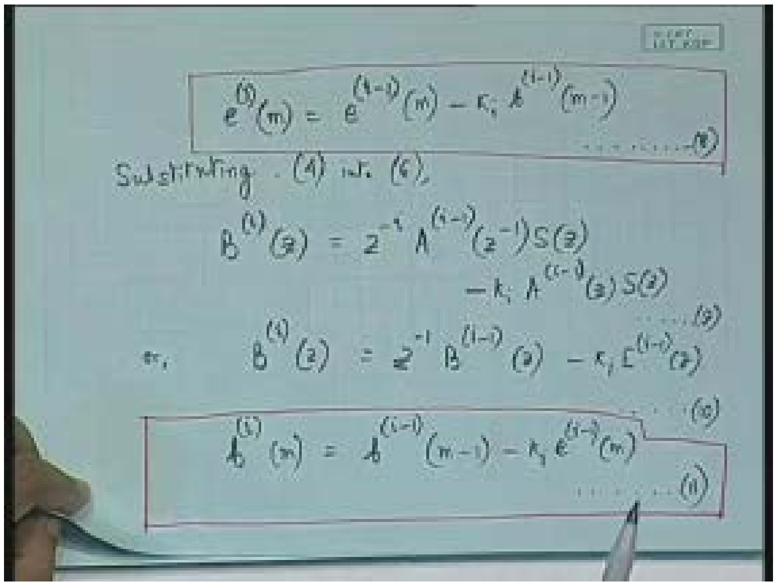
Minimize

$$\tilde{E}^{(i)} = \sum_{m=0}^{N-1}\left[\left(e^{(i)}(m)\right)^2 + \left(b^{(i)}(m)\right)^2\right]$$

Therefore, if our objective is to minimize this e tilde (i) in that case what we are required to do is to differentiate this e(i) with respect to k i and then we have to equate that to zero. So what we have to do is, to minimize this we differentiate this with respect to k i.

Now k i we do not see directly into this expression. But please let us remember that what are the expressions that we had obtained for the e (i) and b (i) in terms of the k i’s; that we had done in the last class so let us see this.

(Refer Slide Time: 15:41)



$$\boxed{e^{(i)}(m) = e^{(i-1)}(m) - k_i b^{(i-1)}(m-1)} \quad \ldots\ldots(8)$$

Substituting (4) in (6),

$$B^{(i)}(z) = z^{-1} A^{(i-1)}(z^{-1}) S(z) - k_i A^{(i-1)}(z) S(z) \quad \ldots(9)$$

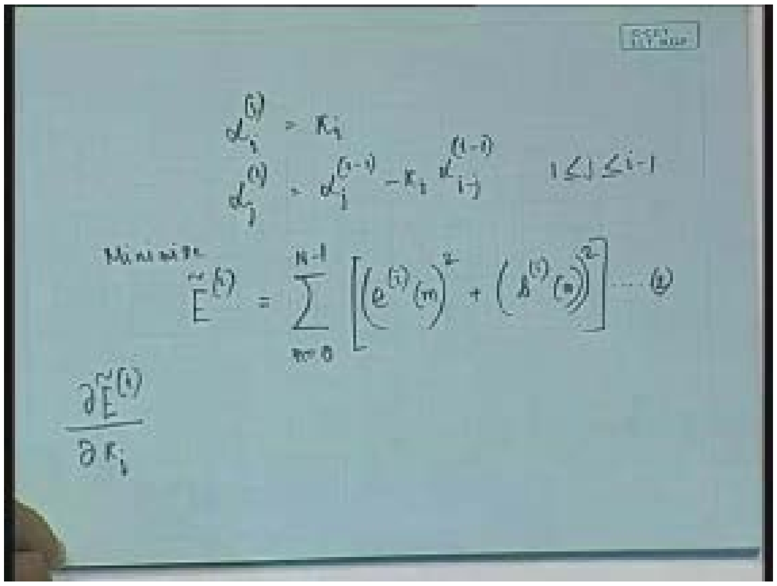
or,

$$B^{(i)}(z) = z^{-1} B^{(i-1)}(z) - k_i E^{(i-1)}(z) \quad \ldots(10)$$

$$\boxed{b^{(i)}(m) = b^{(i-1)}(m-1) - k_i e^{(i-1)}(m)} \quad \ldots\ldots(11)$$

Yes, these two very important relations that we had discussed that e (i) (m) could be expressed in terms of e (i minus 1) (m) minus k i this quantity and b (i) (m) was expressed like this the equation 8 and equation 11 of last class and now what we are going to do is to substitute this e (i) (m) and b (i) (m) into this expression; call it as equation number 2 of today.

(Refer Slide Time: 15:54)



$$\alpha_i^{(i)} = K_i$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - K_i \alpha_{i-j}^{(i-1)} \qquad 1 \le j \le i-1$$

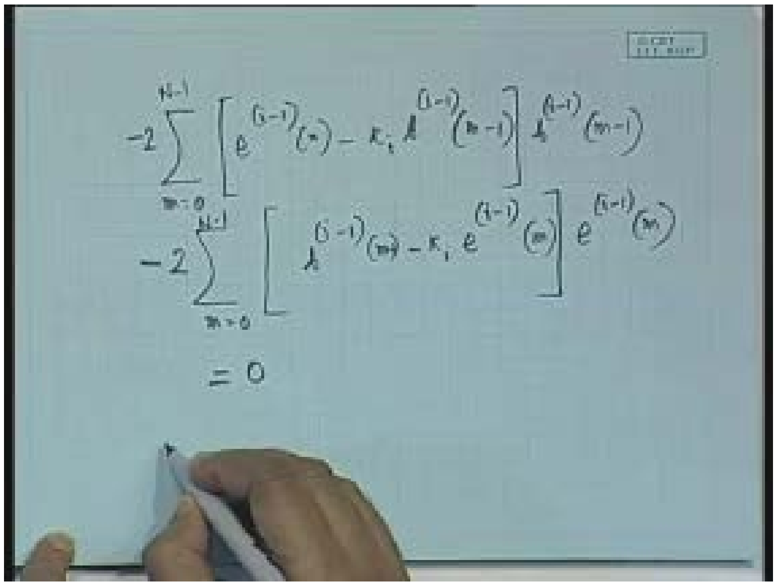
Minimize

$$\tilde{E}^{(i)} = \sum_{m=0}^{N-1} \left[\left(e^{(i)}(m)\right)^2 + \left(b^{(i)}(m)\right)^2 \right] \cdots (2)$$

$$\frac{\partial \tilde{E}^{(i)}}{\partial K_i}$$

There we substitute this e (i) (m) and b (i) (m) expression so there the k i terms are already coming in and then we are differentiating with respect to k i and equating that to zero. So if we differentiate with respect to k i and equate to zero then what results is something like this that in that case we are getting minus 2; having the derivatives it leads to this expression; you can verify this yourself by simply having the differentiation; this is e (i minus 1) (m) minus k i b (i minus 1) (m minus 1); you can definitely expect this because your e (i) expression itself is in terms of e (i minus 1) b (i minus 1) and k i and this into you will be having b (i minus 1) (m minus 1). This will be for the e (i minus 1) for e (i) (m) quantities square and differentiating that and then for the b (i) (m) quantity, after differentiation you will be finding minus 2 m is equal to 0 to N minus 1 and writing now the b (i) (m) expression which is nothing but b (i minus 1) (m minus 1) minus k i e (i minus 1) into m and this to be multiplied by e (i minus 1) (m) and this will be nothing but the derivative of this e (i) with respect to k i and this will be equated to zero and by having that one can solve for k i.

(Refer Slide Time: 18:10)



$$-2\sum_{m=0}^{N-1}\left[e^{(i-1)}(m) - k_i\, b^{(i-1)}(m-1)\right] b^{(i-1)}(m-1)$$

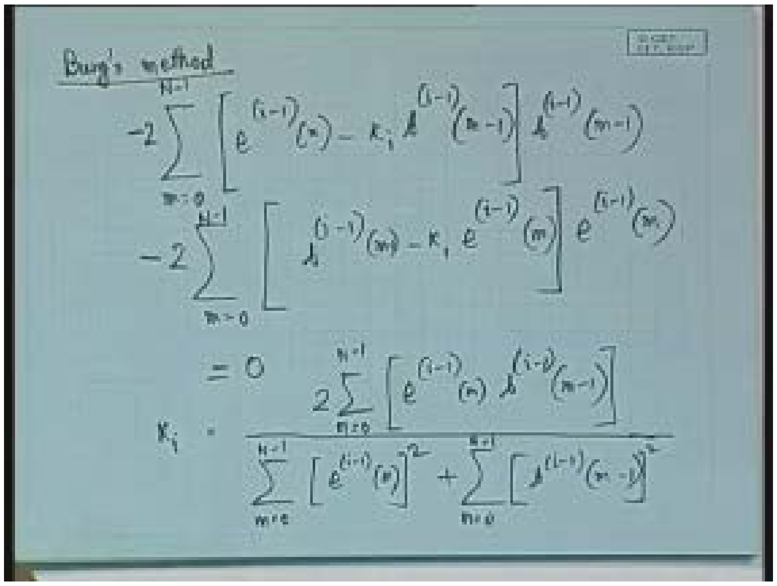
$$-2\sum_{m=0}^{N-1}\left[b^{(i-1)}(m) - k_i\, e^{(i-1)}(m)\right] e^{(i-1)}(m)$$

$$= 0$$

From here you can solve for k i and k i will be given like this that k i will be given as two times summation m is equal to 0 to N minus 1 e (i minus 1) (m) into b (i minus 1) (m minus 1) so still it is involving a correlation but only thing is that the denominator term that differs from the earlier way of writing the denominator term. This time from this expression that means to say that what we are discussing is Burg's method, so according to Burg's method of defining the error e (i) the e tilde (i) rather, we are going to have k i here as k i and the denominator term will be summation m is equal to 0 to N minus 1 e (i minus 1) (m) square plus summation m is equal to 0 to m minus 1 b (i minus 1) into (m minus 1) whole square; sum of these two squares.

You see that individually this e (i) terms e (i) terms squares summation and this squared summation they are to be added. So it is somewhat different from this expression.

(Refer Slide Time: 19:52)



Burg's method

$$-2\sum_{m=0}^{N-1}\left[e^{(i-1)}(m) - k_i b^{(i-1)}(m-1)\right] b^{(i-1)}(m-1)$$

$$-2\sum_{m=0}^{N-1}\left[b^{(i-1)}(m) - k_i e^{(i-1)}(m)\right] e^{(i-1)}(m)$$

$$= 0$$

$$k_i = \frac{2\sum_{m=0}^{N-1}\left[e^{(i-1)}(m)\, b^{(i-1)}(m-1)\right]}{\sum_{m=0}^{N-1}\left[e^{(i-1)}(m)\right]^2 + \sum_{m=0}^{N-1}\left[b^{(i-1)}(m-1)\right]^2}$$

There we had......... like the earlier case, the earlier case k i where like this (Refer Slide Time: 20:00) that the summation's products were there, product of the summation; in this case it is sum of the summations so somewhat different but anyway using this the K i's can be computed. So if one applies this kind of a lattice formulation what one has to do becomes very clear that one has to feed the s of n the pitch segment and then one has to obtain the e (0) n and b (0) n which is nothing but given here already that e (0) m and b (0) m is nothing but s of m, the first estimate because it is as good as the zeroth order predictor and then obtain k 1 out of this in terms of this e (0) and b (0) and then compute these k 1's and using these k 1's you do the filtering; obtain e (1) n and b (1) (n) and continue until e (p) of n and b (p) of n and then what you are basically getting out of this k 1 to K P using that you can compute the alphas and those alphas are going to be your final alphas that is what you are using for the........ those are the alphas that you are using in the LPC as the LPC parameters.

Now this is quite an effective technique no doubt. And let us see what are the other benefits one gets out of this LPC parameter. Now we have seen already that out of the that using the LPC parameters in this kind of a lattice formulation what has evolved is that we are getting all this e (0) e (1) etc up to e p b (0) b (1) up to bp's so all these are the prediction error signals.

and in fact In general the prediction error signals would be given by this expression which we have already known that e of n we are going to write as s of n that is the signal minus whatever prediction we are making. So, if in general we make a pth order prediction then we sum it up from K is equal to 1 to P into what alpha k alpha k because k is the running index s(n minus k). So, using the past samples we are predicting this so this is already known to us and this will be written as G of u(n). So what we can use is that that these form of n if we now take e(n) to be like our signal then we will be finding that e(n) is often a very good approximation to the excitation.

What is after all this G of u(n)?
We will come this very shortly. In fact in the pitch synthesis model which we are going discuss very shortly, you will be finding that we need an excitation to be given and that excitation is the excitation which is given to the vocal tract model. And as this excitation we can feed this error signal e of n and the benefit that we are getting from this e of n is that, you can notice that what will be the characteristics of this e(n).

Now what happens is that when the signal suddenly attends a peak in that case what we are going to do; we are going to predict based on the past samples but the past samples may be still smaller in magnitude and then we are predicting the present sample where the present sample has already attained a larger value but predicting it from the smaller value will lead to more prediction error. So whenever the signal changes suddenly in that case even the prediction error also will be large. So a sudden large signal would also result in a large prediction error. Now that will happen in the voice pitch when especially at the pitch periods. whenever we are having the corresponding to the pitch periods we are going to have large peaks available at the s of n; s of n is our original pitch signal input as we are saying. But it is not only s of n but even in e of n we will be having existence of such peaks. then can we not use this e(n) signals; instead of s(n) signals can we not use this e(n) signals in order to estimate the pitch; you may ask that what difference does it make because s of n is already available to us so why should we predict and then make a prediction error signal and use that.

Well, if you are doing the spectral analysis of this s of n in that case just remember that s of n is not only going to contain the signal corresponding to the pitch period but also it will contain some harmonics which results out of the formant frequencies. The formant resonances would also like to give some extra peaks in addition to the genuine pitch period peaks and we already discussed that it is those extra pitch which often tend to give confusing results whenever you are estimating the pitch period using the autocorrelation on the s of n.

So what is the culprit there?
The formant frequencies.

Now what happens in the case of e(n) is that e(n) is after all an error signal. So error signal; what happens is that the error signal becomes uncorrelated, relatively uncorrelated from sample to sample. So error signal is more having a noise like characteristic so it will be uncorrelated. So although the error signal will show some peaks corresponding to the pitch period duration but at other times other than the pitch period it will have some kind of a random behavior and in the unvoiced part of the pitch always the signal will always have a kind of random nature. So, as a result of that if we now take an autocorrelation of e of n instead of s of n in that case what we can expect.

In that case what happens is that the autocorrelation peaks will be obtained corresponding to the genuine pitch periods because of the absence of the formants in e of n, because e of n is having a relatively flattened spectrum is it not; as compared to s of n. s of n spectrum will contain the formant frequencies. But by having e of n instead of s of n all those formant frequencies will be smoothened out and using the e(n) and a correlation over e(n) it should be possible for us to get the extra get the autocorrelation peaks and those autocorrelation peaks will correspond to the pitch periods. So it will be effectively used, this e of n can then be very effectively used for pitch period estimation as compared to s of n. that is one advantage that we are getting by using e of n rather than s of n.

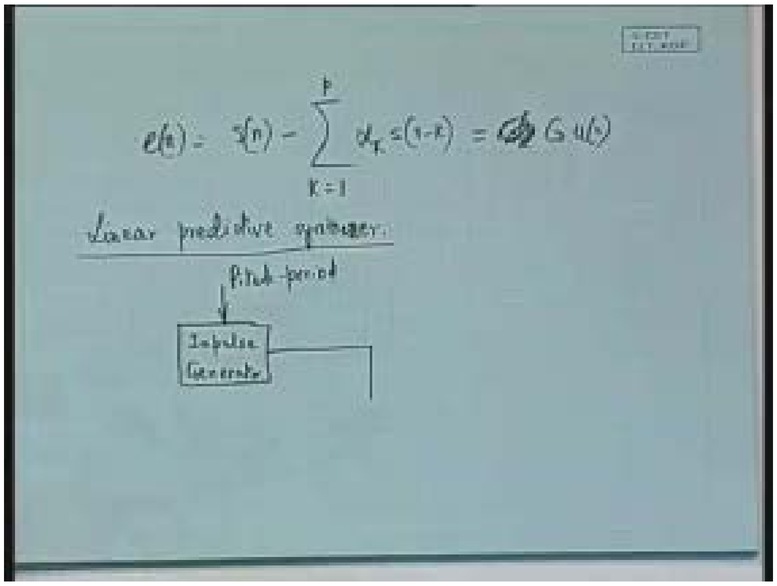
Now we are going over to what we had originally discussed as the title of today’s lecture and that is about the linear predictive synthesizer.

Any questions at this stage; on this part what we discussed because you had some doubts in the last class and I hope that that doubt has now been solved. So definitely when we discussed about the lattice formulation the doubt was that whether K i's would link some extra computational burden and then that will nullify the advantage of lattice structure it is not so. So K i's can be computed in the lattice structure itself and in fact what advantage we get in the process of lattice formulation is that here no such explicit computation of the correlations is required.

One can just keep on recursively compute the error signals and that itself would lead K i's and K i's would ultimately lead to the predictions.

Now, in the linear predictive synthesizer it would be like this that we will be having, in the form of block diagram we will be having one train of pulse so there will be an impulse generator there will be an impulse generator, just train of impulse that correspond to the................. what will be the parameter of that impulse, train of impulse so the pitch period should be a parameter. So, if we feed the pitch period or our estimated pitch period to the impulse generator then it will accordingly generate a train of impulse at that frequency. And if next time the pitch period changes then the train of impulses would also readjust it in between impulse periods.

(Refer Slide Time: 31:56)



$$e(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) = G\,u(n)$$

Linear predictive synthesizer.

Pitch period
Impulse Generator

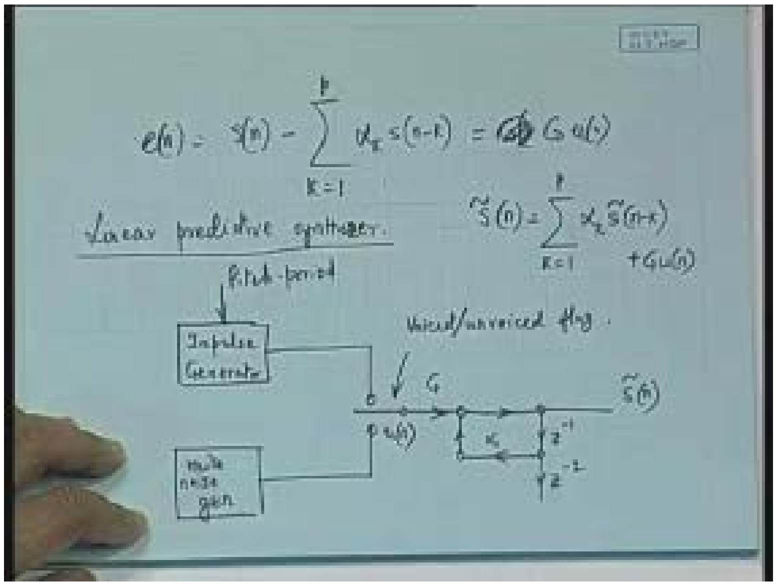
Now this train of impulses will be generated........... for what; for the purpose of the voiced pitch and for the unvoiced pitch there will be a white noise generator. So we will be having a white noise generator white noise generator and this white noise generator also will have its output over here and then there will be a selector switch and this selector switch we will be referring to as the voiced/unvoiced flag. And remember that from our pitch analysis system linear predictive analysis system we are also giving the voiced unvoiced flag. First is the pitch period estimation and then the voiced/unvoiced flag that will be provided and then there will be a gain parameter G which will basically dictate what is going to be the amplitude of the pitch signal so that in the synthesis process also the same amplitude can be used. And after that we are going to have what............? Then in the normal pitch synthesizer what are we supposed to have after this? We are supposed to have the filter and that filter is nothing but the formants filter.

Now who is going to provide that filter?
Yes, using our linear predictive coding; because after all what is it that we are going to obtain? We are going obtain s tilde of n which will be nothing but the synthesized pitch and the synthesized pitch can be written as the summation alpha k s tilde n minus k; s tilde n minus k means all the synthesized past samples and k would go from k is equal to 1 to P and this plus G u(n) u(n) is going to be the unit step function so G u(n) so u(n) would be available over here so G into u(n) the signal available at this is G as u(n) so G multiplied by un and that should be added to this, so this would essentially form our synthesis filter. So this added to the synthesis filter would then generate s tilde of n.

So, to obtain s tilde of n what we have to do; we have to make a realization of this filter. Now realization of this filter we can have a direct realization using this equation. So a direct realization of this equation would give us a structure; a filter structure like this (Refer Slide Time: 35:20) that we will be generating........ if we have s tilde n over here then we have to generate the set of samples s tilde n minus 1, s tilde n minus 2 all the past samples have to be made available to us for the linear prediction and to make that available we have to add the delay elements that means to say z to the power minus 1.

Then s tilde n is here, so after z to the power minus 1 what we are getting here is s tilde n minus 1 at this point. And s tilde n minus 1 we have to multiply by alpha 1 so that this would be multiplied by alpha 1 and then it is to be added with this G u of n; G u of n is already there so this should be added to G u of n.

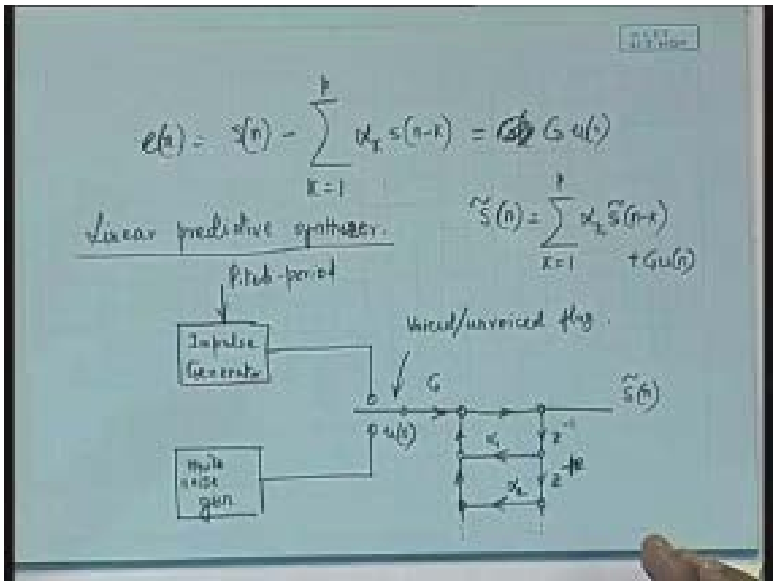
(Refer Slide Time: 36:30)



$e(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) = G u(n)$
Linear predictive synthesizer.
$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k \tilde{s}(n-k) + G u(n)$
Pitch-period
Impulse Generator
Voiced/unvoiced flag.
G
$\tilde{s}(n)$
$u(n)$
α_1
z^{-1}
z^{-2}
White noise gen

This is only up to the first sample, so this is only a first-order predictor but then for a second-predictor what we need to have; we need to have yet another delay element sorry z to the power minus 1 only so that totally it is having two elements delay, this is z to the power minus 1 so that this is s tilde n minus 2 and this has to be to multiplied by alpha 2 and this also will be added to this and then this goes on.... I do not have the space to write down, I hope you can follow this.

(Refer Slide Time: 37:03)

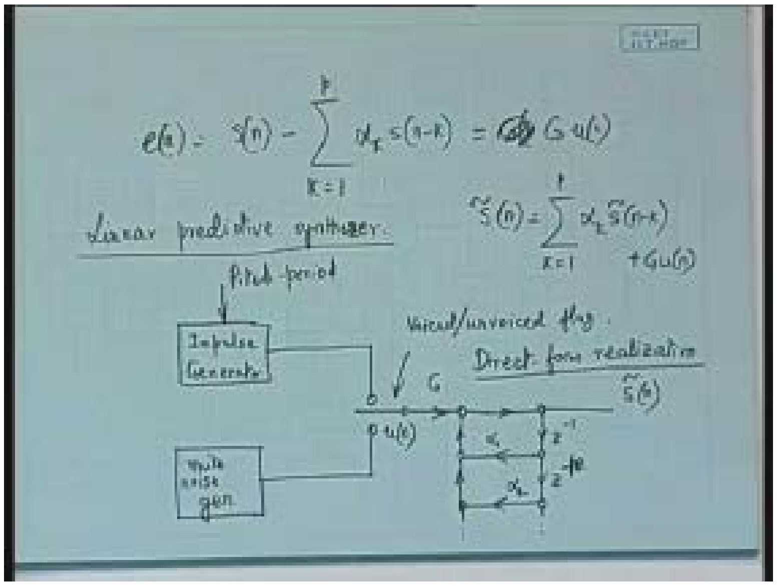

$e(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) = G u(n)$
Linear predictive synthesizer.
$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k \tilde{s}(n-k) + G u(n)$
Pitch-period
Impulse Generator
Voiced/unvoiced flag
G
$\tilde{s}(n)$
$u(n)$
White noise gen
α_1
α_2
z^{-1}
z^{-1}

So all this z to the power minus 1 elements will follow one below the other and then that will give us what is called as the direct form realization. This is nothing but a direct form direct form realization of the filter. Now, direct form is having advantage definitely because the direct form realization is a very simple realization, it is easy to implement. But coming to the disadvantage side that it is having; it is having sensitivity with respect to the variations in this alpha 1 alpha 2 so variations in this alpha case make the realizations sensitive and there are alternative forms of filter realization which will be which are possible.

We are not going into all those details because there are some excellent references which are available, because now it is time that we have to draw the curtain on our discussions regarding the LPC analysis because we have been spending the past few lectures at least on this. I think that the direct form realizations alternatives one can study oneself and then get a feel of that.

Now, in this case in the linear predictive synthesizer there are two more aspects that one needs to consider. First is that these parameters; these parameters means the pitch period and this gain how often are you required to provide this information. Now, as far as the unvoiced pitch is concerned it is sufficient that if this information is given or updated per frame; because it is even for a frame only that we will be having the pitch period estimation or the gain estimation so for every frame we are estimating that, also the voiced/unvoiced flag status, so at the beginning of every frame we will be obtaining these parameters. But for these unvoiced pitch, once per frame should be sufficient. But regarding the voiced pitch, if we have a pitch period that is considered to be fixed with reference to that frame then we have some problem in the synthesis of the pitch.

(Refer Slide Time: 39:31)



$$e(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) = G u(n)$$

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k \tilde{s}(n-k) + G u(n)$$

In fact ultimately when we hear that segment of the pitch we will be finding that it is not a very natural pitch why because that even within one frame whatever number of pitch periods are accommodated within that the pitch periods do vary to some extent. that is why what we have to do is that, based on the past pitch period estimations and the pitch period that is estimated from this frame we do some kind of an interpolation so that at every pitch period we are estimating a new parameter. This is called as pitch synchronous synthesis.

So a pitch synchronous synthesis is considered to be much more efficient as compared to what is called as the asynchronous synthesis. This is a pitch synchronous synthesis and the alternative to this is asynchronous synthesis. What we have to do in a pitch synchronous synthesis is that, for every pitch period we have to interpolate and obtain a fresh estimate of the pitch parameter or rather the pitch period estimation and then realize this; then realize this linear predictive synthesizer.

(Refer Slide Time: 41:54)

pitch-synchronous synthesis

asynchronous synthesis

Now, we also show that how the LPC can be used for a very good pitch period estimation because again the success of the pitch synthesis is really dependent upon how good or how accurately your pitch period estimation is. Now we just qualitatively discuss that error signal could serve as a better pitch period estimation. But is there any mechanism whereby using our original signal that s of n itself, if we can filter out the formant frequencies; meaning that if we can flatten the spectrum of this s of n using any other technique then that itself will suffice, we do not have to really go for the error signals because the error signal; I mean, again the uncorrelated part of it is a noise like thing and there it is not free from its inherent disadvantages so we will be now discussing about the LPC method of pitch period estimation; LPC method of pitch period estimation from the signal itself.

Therefore, in that what we do is that we have what is called as the simple inverse filter tracking algorithm simple inverse filter tracking algorithm and because this becomes as a very long sentence we write it in the short form S I F and T. so this is referred to as the SIFT algorithm;

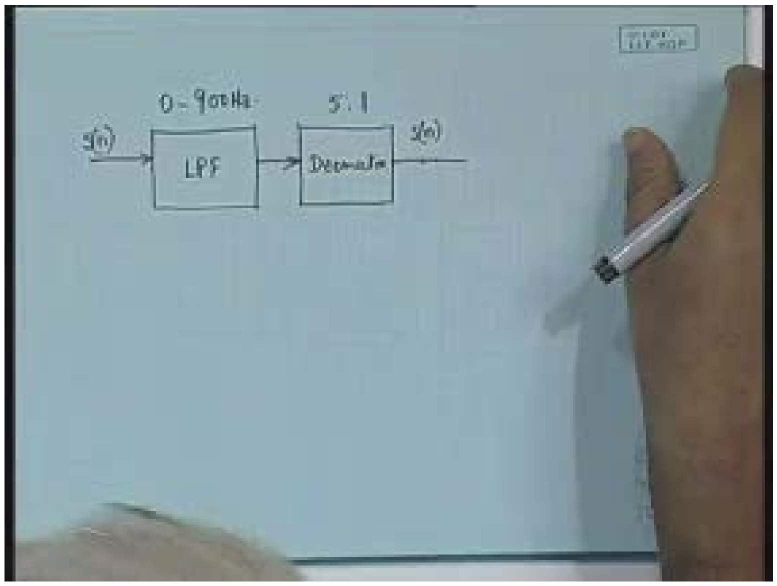
simple inverse filter tracking and in the SIFT algorithm the block diagram goes like this that here we will be having s of n as the input and then we will be having the low pass filter and this low pass filter is going to have a cutoff in the range of lower cutoff in the range of, upper cutoff in the range of 900 Hz so the filter would accept only the signals in the range of 0 to 900 Hz and then followed by the LPF, we are going to have what is called as the decimator, and this decimator is going to decimate down the signal in the ratio 5 is to 1.

Now, what for a decimator is used?
This will be very clear. The s of n is sampled at what rate; let us say 10 kHz rate and then now this is low pass filtered at 900 Hz. So, at the output of the LPF you are having 900 Hz as the maximum frequency content so what you can do is that a sampling frequency of 2 kHz should be good enough for you. so you can so you need not have to keep all the five samples; every one out of five samples you can discard so this is called as a 5 is to 1 decimator. basically it is just to down sample or to reduce the sampling frequency, so you will be getting so you will be using the decimator. Hence, here (Refer Slide Time: 46:25) you will be obtaining s of n but a decimated version of s of n after it is low pass filtered.

And why this 900 Hz; what is the beauty of this 900 Hz, because you must be wondering that we always say that this pitch signal goes up to 3.3 or 3.4 kHz so why are we band limiting it to 900 Hz? Well, the whole purpose of this block diagram is pitch detection. So for pitch detection we have to avoid the formants. So what we do is that by having a filtering up to 900 Hz we avoid the formants, still the first or the second formant at the most would still be present.

In fact, if we have a LPF below that then you can ask me that what if we have our LPF designed such that it is instead of 900 Hz it comes down to 450 Hz or 500 Hz like that. well, the difficulty there is that the pitch frequency's variation also is often like this because pitch is very often found to be the maximum limit, the upper limit of the pitch frequency is often found to be in the range in a close range of the format frequency and that is why 900 Hz or less than 1 kHz is a reasonably good choice so that only the first or the second formant frequency is still kept but the higher formants are all discarded through this LPF process and also, the signal is still keeping the pitch period information.

(Refer Slide Time: 48:19)



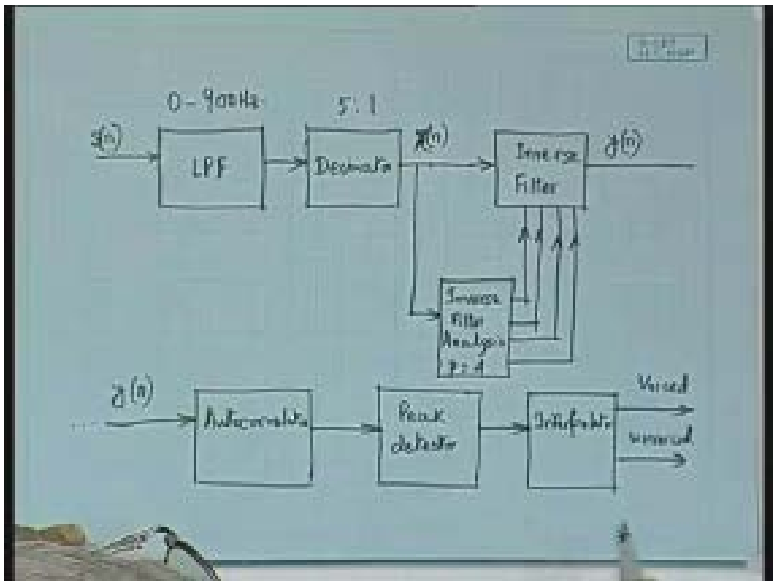
So this s of n; now what happens is that this s of n what we have to do is to flatten its spectrum. So, in order to flatten the spectrum we need to have a filter and that filter is referred to as an inverse filter where we will be having a flattened spectrum version of this s of n and at the output of the filter we will be referring to the output as y of n.

In fact, the decimated; let us call it as x of n as the decimated signal and y of n as the output of the inverse filter. Now what is the realization of this inverse filter? This inverse filter would be based on the LPC parameters. How do we do that?
Using this x of n we add a block which we are calling as the inverse filter analysis block. So this is an inverse filter analysis block and we are going to have that for P equal to f4. Therefore, P is equal to 4 means we will be having four four parameters alpha 1 alpha 2 alpha 3 and alpha 4 so these four parameters alpha 1 alpha 2 alpha 3 and alpha 4 they all will be fed to this inverse filter so now this y of n is nothing but the flattened spectrum version of this s of n.

Therefore, now, since it is a flattened spectrum what we are going to do after this in order to do the pitch period estimation; what would you expect as the block following this?

Autocorrelation; so we will now be carrying out the autocorrelation, we will be performing the autocorrelation over this y of n. so what we do is that use this y of n; this is a continuation of the block diagram so please note that point. So y of n would be fed to an autocorrelator or a block that performs autocorrelation and then the autocorrelation output will contain the peaks corresponding to the pitch period. There will be a peak detector or peak picker whatever you can say, peak detector and the peak detector output again.......... what we have to do is to use an interpolator because using the interpolator we are going to decide about the voiced and the unvoiced classification and the pitch period the estimated pitch period is of course obtained from this thing.

(Refer Slide Time: 51:49)



Now this is really something very useful. So here we find a very good application of the LPC that using the LPC parameters we are doing a kind of inverse filter and then have a better pitch period estimator as compared to the conventional approach what we had studied. And in fact, for the ease and simplicity, the LPC has become a very popular approach, a very popular and widely used methodology for pitch analysis and synthesis because of its wide application.

We have in fact studied the theory of theory behind the LPC to a sufficient extent. So we will now be in a position to just integrate everything and realize what is called as the LPC vocoder. We have all the concepts available with us. We know that what all have to be done. But just to assemble those blocks and realizing an LPC vocoder which we are going to take up at the beginning of the next class, we will really make our discussion related to the LPC conflict. So thank you for now.