Digital Voice and Picture Communication

Prof. S. Sengupta

Department of Electronics and Communication Engineering

Indian Institute of Technology Kharagpur

Lecture - 13

Lattice Formulation of LPC Coefficient

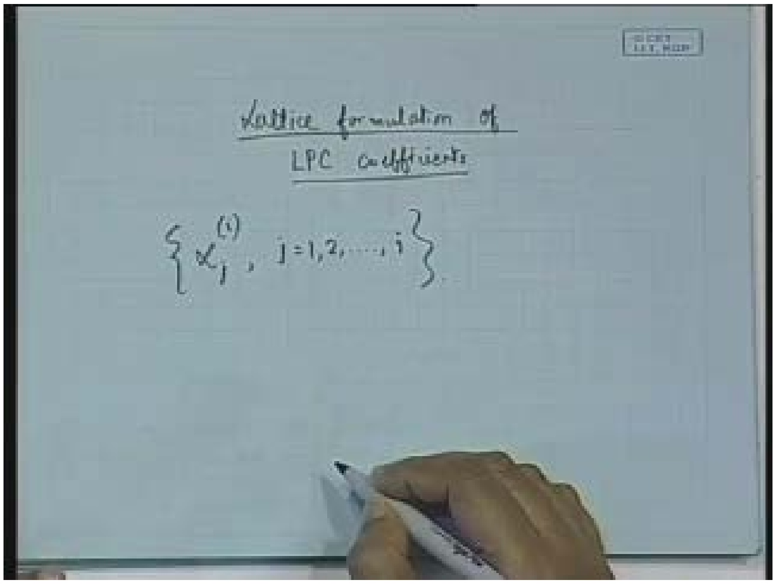
Today's lecture.............. is on the lattice formulation of LPC coefficients. Over the past few classes we have been discussing about the different approaches in order to compute the LPC coefficients. The problem we posed few classes back but then we had seen that essentially the challenge lies in trying to obtain the optimal set of coefficients which will be based on the observation based on the pitch. And we had been discussing few methods such as the covariance method and the correlation based method, auto auto autocorrelation based method and there we had seen that essentially the approach gets into an iterativpitche one so that one can compute in an efficient way all these coefficients alpha k's where k varies from 1 to p. So essentially we are obtaining the first set of estimates and then we are updating our estimates with iterations and at the end of the pth iteration we are getting the final coefficients which we have to use.

Now this involves as I said that this involves two basic operations: first is that we have to determine the correlation from the observed pitch waveform and the second stage is that we have to do the matrix operation in order to determine the LPC coefficients so these two are involved.

Now there is a methodology which we are going to call as the lattice formulation where these two approaches would be treated in an integrated manner. That means to say that we will not be having any explicit computation of the correlation but rather we will be determining the LPC coefficients in terms of the prediction errors and we will be describing the technique pertaining to that very shortly.

So, before we begin the lattice formulation let us go back to what we did in Durbin's algorithm for the autocorrelation. remember that for Durbin's algorithm what we were doing is that the set of coefficients we were calling as alpha j the set alpha j and we were writing it as alpha j with the superscript i the superscript indicating the iteration number so we were writing the set of alpha ji and this j was in the range of 1 to i this was the set that we have and the significance is that all these coefficients are the ith order optimal linear predictor

(Refer Slide Time: 4:45)



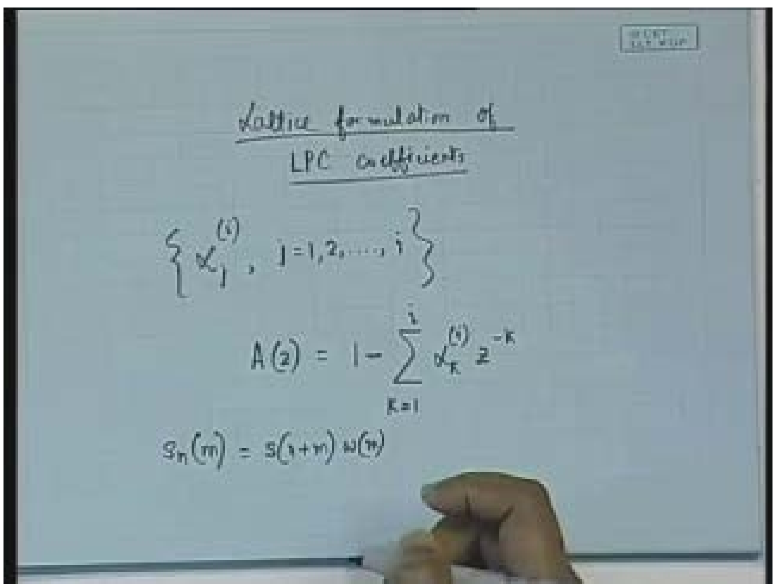
So we are having only up to i so in the ith iteration of prediction we are having only ith order of prediction i passed coefficient. So in the n we are going to at p, i is equal to p at the end and there we will be having alpha jp as the last estimate. We have already seen that. And using these coefficients means to say that using this ith order predictive coefficients it is possible for us to design a filter. And let us give the filter expression like this that we are going to call that filter in the z domain, we are going to call that as A(z) and we are going to write the filter A of z as 1 minus and then we express this as the summation of alpha k superscript (i) into z to the power minus k and this index k here changes from 1 to i (Refer Slide Time: 5:57).

So essentially what we have written is that, again with this ith order prediction we are expressing the filter. So this is just a filter. Let us let us imagine that somehow we are able to realize a filter using this alpha k's, using this prediction parameters alpha k's we are able to obtain a design a filter like this.

Now in this filter if we are giving a pitch segment as an input to this filter then what would you expect at the output of that filter?

This is a filter realized out of the prediction coefficients so what I am asking is input is the pitch segment, so we call the pitch segment; so the input to the filter is the pitch segment and we are going to write it as s suffix n of m. You know that what we mean by this expression that suffix n and within parenthesis when we are writing (m) that means to say that we are going over to the nth sample and then we are going to have (m) as our variable index and that segment we are going to write it as s(n plus m) into w(m) where w(m) is the windowing coefficients.

(Refer Slide Time: 7:41)



So essentially s n of m is the windowed version of the segment and again for simplicity instead of writing every time the segment with the subscript n we are going to write it as s of m. So, what we are writing here as s n(m) we can simplify that by writing s of m by just simply dropping this subscript m.

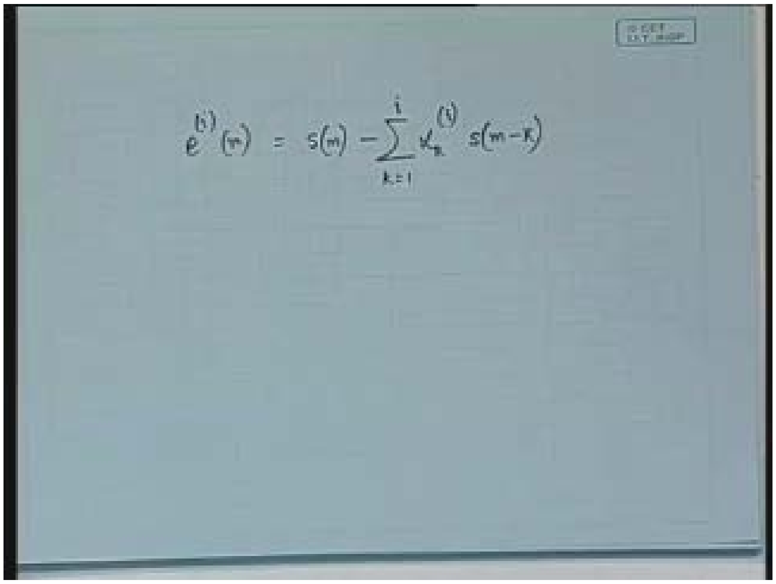
Now if s of m is the input to the filter, the filter as defined above what would you expect at the output of the filter? Any answer? See, what is this filter?

This filter is realized out of the predictor coefficient. Have you come across this filter expression (Refer Slide Time: 8:42) in one of the previous lessons; I believe you have; the lesson which we had started, I mean, when we had started this linear prediction linear predictive coding there we had come across an expression like this.

[Conversation between Student and Professor – Not audible ((00:09:01 min))] correct, that is, that is it. So whenever we are giving the s of m as an input to this filter we will be getting the error or rather the prediction error at the output. So we can write it, in the time domain expression we can write the prediction error to be like this. In fact the prediction error also we will be normally writing it as e n with the superscript (i) e n (i) (m) that is how we are going to write and again for simplicity we will be dropping this subscript n and we are simplify going to write it as e (i) of m which will be the ith prediction error. So the prediction error naturally whenever I am writing it with the superscript that means to say that the prediction error will be iterative.

Hence, the output to the filter will be the prediction error and we are going to write it like this. So let us say that........ let us number the equations. So the first one, the filter that we realize out of the predictor coefficient is the equation number 1 this z transform expression of that this we are calling as the equation number 1 and as equation number 2 we are going to write the expression for the error term. So the error expression will be e (i) (m) and that will be written as s of m minus summation k is equal to 1 to i alpha k i s of m minus k. This directly follows. You can just have a look at the expression, this is in the z transform domain which means to say that when you are back again to the time domain this z to be power minus k is there which will introduce k units of delay which gets expressed as s of m minus k when we are back again to the time domain. So this is what we are having, so this is the prediction error. Now this we are going to call as equation number 2.

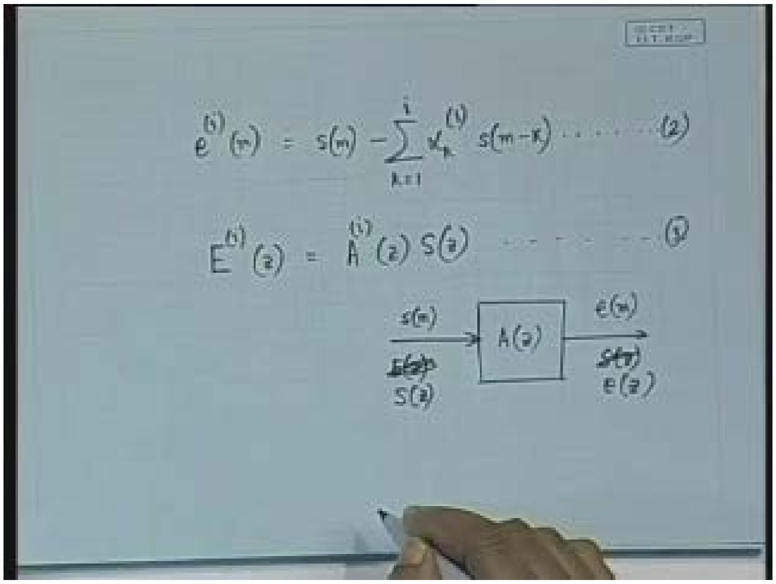
(Refer Slide Time: 11:26)



Now let us take the z transform of equation number 2. So what are we going write?

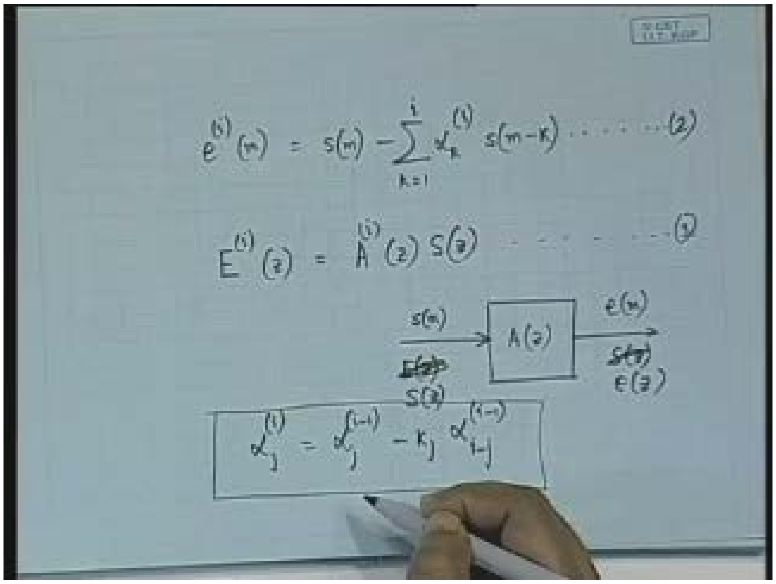
In fact taking the z transform we can write it as; now we are going to express it as the capital capital E (i) of z. In fact we did not have to use this at all because we know very well that E (i) of z is going to be A (i) of z that is to say the filter times s of z which means to say that the filter that we have is simply like this that here at the input we are going to have s of m and then we have a filter over here whose function is A of z if we say and then the output of that will be e of m or rather to say in the z transform domain we are going to write it as e of z and here it will be s of z the corresponding z transformed version. So, that in the z transform our............ sorry sorry I mean, I just wrote it in the other place wrong place; s of z and here e of z so e of z will be nothing but A of z into S of z so this becomes our expression 3.

(Refer Slide Time: 13:03)



Now again our objective will be to express the relationship in an iterative form. In order to do that let us go back to Durbin's relation what we were discussing in the last class and as per the Durbin's relation remember that we had written this kind of an expression that: alpha ji the ith estimate of the predictor we were writing as alpha j (i) minus 1 minus k j alpha i minus j of (i minus 1)th iteration this is what we had written for the alphas recursion; this is a recursive relation we got for the alphas.

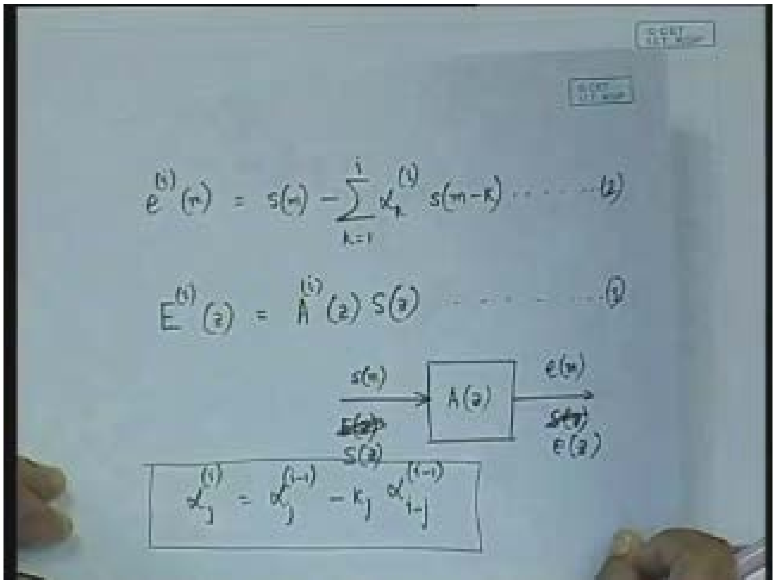
(Refer Slide Time: 13:57)



Now what we are going to do is that we are going to put this expression of alpha in............ we will be putting it in the equation number 1; equation number 1 is this; (Refer Slide Time: 14:18) of course equation number 1 is written in the z transform form, this gets the filters z transform form and in place of this alpha k i we are going write this expression which means to say that the ith prediction will be expressed in terms of i minus 1th prediction and this residual term that is what we have to do. And again let me tell you; although I will not be doing it in that details but I leave to you for working.

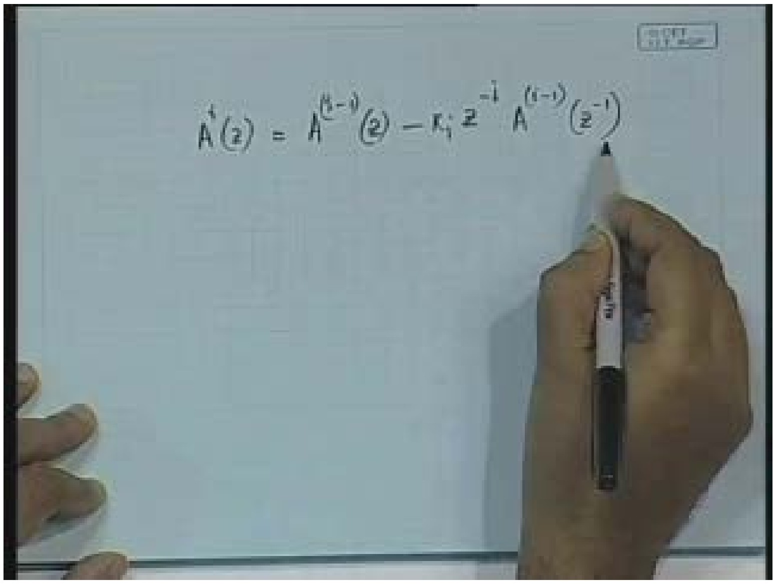
What you can play with this equation number 1 is that you can express this equation number 1 as 1 minus summation k is equal to 1 to i minus 1 and make it alpha k i minus 1 z to the power minus k and then you get another term which will be the ith the last term of the expression which will be alpha (i, i) into z to the power minus I; just replacing k by i you get the last term. And now this alpha (i, i) what you are getting, to express this alpha (i, i) you just make use of this expression (Refer Slide Time: 15:37) you get this alpha i previous and then you get alpha then you get this expression and then again in order to get this you substitute the i minus 2th prediction and so on. So in that process after doing the manipulations accordingly, in order to play with the recursive relation................. actually you have to know the art of playing with the recursive relationship, some students are quite good in that.

(Refer Slide Time: 16:18)



So if you if you can play well with those recursions ultimately what you are going to get is a recursive relation as something like this: A i (z) that can be expressed as A (i minus 1) (z) minus k i z to the power minus i. See, we can immediately note that how this z to the power minus i term will come; this is after isolating the last term in the summation process we are definitely going get a z to the power minus i term and then with a proper by properly identifying you will be finding that this becomes A (i minus 1) and the argument of this becomes (z to the power minus 1) this is very interesting.

(Refer Slide Time: 17:16)



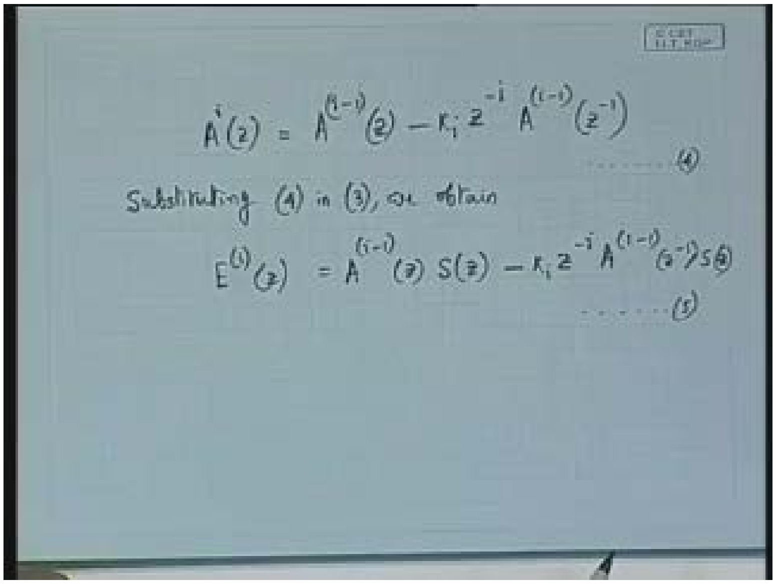
Here is a (z to the power minus 1) term as an argument of this A (z). Here it is A (z), here it is A (z) but here it is A (z to the power minus 1). By simplification, by properly working this is what you will be obtaining and this will be something which will become a very interesting..................... and more interesting will be in the time domain, very intelligent students may be able to guess what it is leading to.

Can anybody prematurely guess what is it leading to; z to the power minus 1?
Okay, then let us let us wait for a while. We will see the great thing that this z to the power minus 1 is going to do.

Now this is expression number 4and substituting 4 in the equation number 3 that will be a very simple equation. Equation number 3 is the simplest equation which just expresses the error in terms of this A (z) and S (z) and if I substitute this equation number 4 into that simply what results is; what we are going to do; E (i) (z) is equal to A (i) (z) S(z); now in place of A (i) (z) we are going to write simply this so what simply results is that it will be A (i) (z) a S(z) that means to say A (i minus 1) (z) S(z) minus this expression times S(z).

Let us write it down. So substituting 4 in 3; E (i) of (z) is equal to A (i minus 1) (z) S(z) minus k i z to the power minus i A minus 1 argument z minus 1 s of z. this is the expression and let us call this as equation number 5.

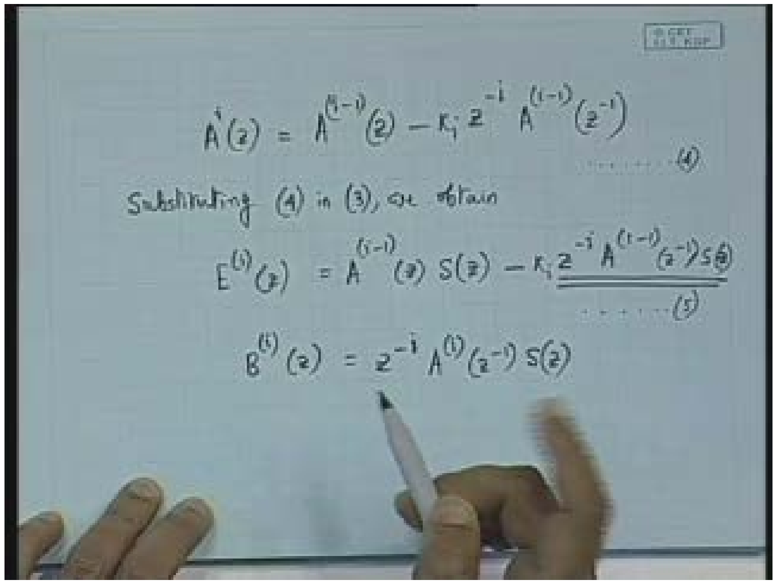
(Refer Slide Time: 19:35)



Now what is the first term of this; what is the first term of this expression?

The first term is nothing but E (i minus 1) (z) because it simply results in the prediction error for i minus 1th predictor and then there is a it is not just the (i minus 1)th prediction error but also there is something, a term like this with that interesting z to the power minus 1 and let us just make a just introduce, just define a term. Let us say that we define another filter expression say B (i) of (z) we write B (i) of (z) we define as this expression: z to the power minus i A (i) with an argument (z to the power minus 1) s of z so this expression; what remains over here is this expression and this we are calling as B (i) (z). of course here it is (i minus 1) so then automatically this will become z this this will become B (i minus 1); of course that would mean that there is another term which will be there and that will be z to the power plus 1.

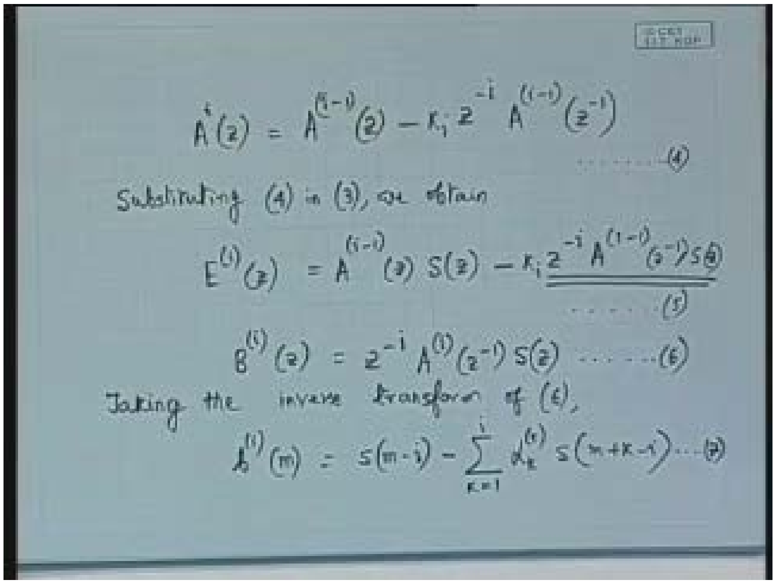
(Refer Slide Time: 21:15)



Anyway, ultimately, if you are taking the inverse of this so say this is equation 6 and taking the inverse transform of 6 taking the inverse transform of 6, in the left hand side we are definitely going to have....... I mean, call its inverse transform to be small b (i) and now instead of (z) we will be writing (m) because we are back again to the time domain and this becomes equal to s (m minus i) minus summation k is equal to 1 to i alpha k (i) s(m plus k minus i) [Conversation between Student and Professor – Not audible ((00:22:14 min))] Non-causal system; yes, good observation, any other observation? Delayed version; would you call it as delayed version really or would you call it as advanced version?

Delayed version is of course very understandable. When we were doing this alpha k's; the way we use the alpha k's and we were generating the error that error is a prediction error which we are going to call as the forward prediction error. Now there is a great deal of similarity between........ call it as equation 7............ between equation 7 what we wrote over here and between equation 2; just let us have a look at equation 2.

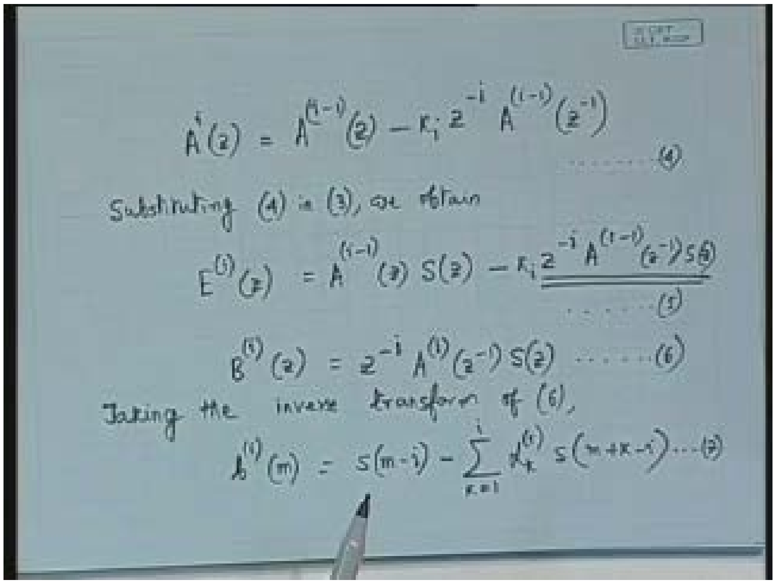
(Refer Slide Time: 23:15)



Now you see now you try to find out the similarity between equation 2 and equation 7. Both are quite similar expressions that there is a (s) term and there is a prediction term summation k is equal to 1 to i, alpha k, those things are remaining the same, the only thing is that here there is s(m) and the predicted versions are s (m minus k), the samples which are being used for prediction those are s (m minus k) and k is a positive quantity which means to say that we are going back, we are taking the past samples to predict s(m). So we are using the i past samples to predict s(m) and in that process generating a forward prediction error E (i) (m).

What we are doing over here is just something which is opposite. Here (Refer Slide Time: 24:24) it is s (m minus i) so it is as if to say that we are going to predict s(m minus i) from s (m plus k minus i) and k again is a positive quantity that means to say that we are going ahead in time. So, based on the future samples we are going to predict the present sample something like this. So in this case it is a backward prediction error and because it is a backward prediction error we will be denoting we are denoting this by b, so E we are reserving for forward prediction and b we are reserving for backward prediction.

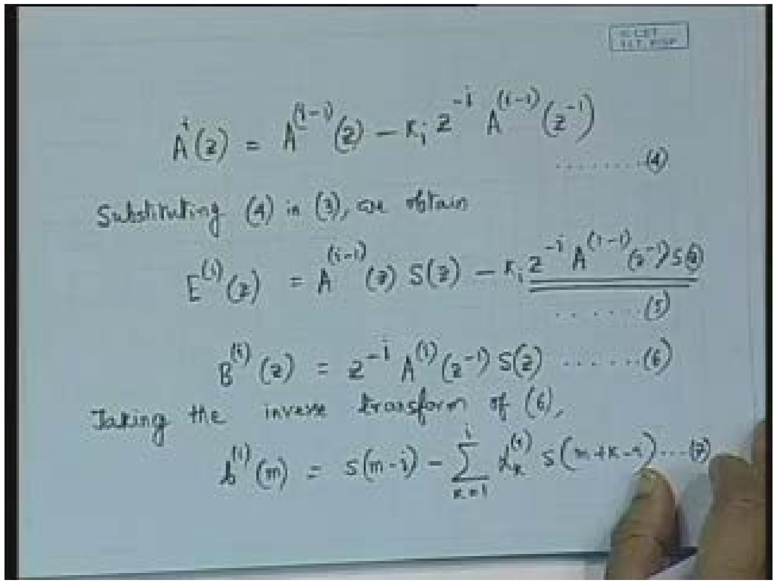
(Refer Slide Time: 25:10)



Hence, there is a backward prediction, there is a forward prediction. Now it may seem to be puzzling as one of your friends immediately expressed the application that it is getting into non-causal; yes, it is apparently becoming non-causal in the sense that we are using future samples so it is definitely in that sense a non-causal. But it does not matter. It is realizable because what you have to do is only to introduce delays. If you introduce some delays then the same set of samples could be used to predict a past one and same set of samples could be used to predict a future one.

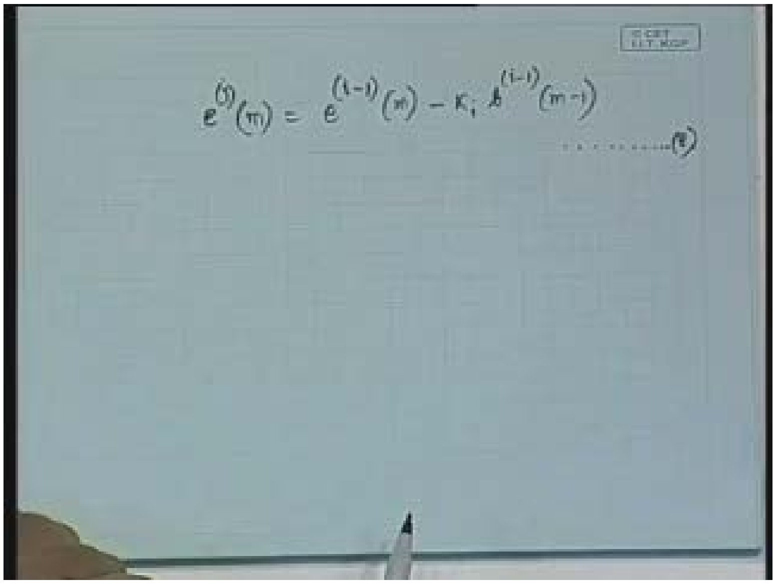
Thus, let us now live with the mathematical juggleries without going too much into the nitty-gritties of it but realizing that the physical significance that emerges from it is that we have to build simultaneously with both forward prediction error and also with the backward prediction error.

(Refer Slide Time: 26:24)



Now what we are going to do is that......... the prediction error; now we return to this equation; so this is our equation number 5 and if we return to this equation number 5 from here or rather to say; yeah you can say that if we go over to.................. in fact equation number 2 because equation number 5 is in the z transform domain so better you look at the equation number 2 which is already in the time domain. So there what we can do is that there is a summation term k is equal to 1 to i; instead if you are taking (i minus 1) you will be observing that essentially we will be able to identify the e of i minus 1 m and in fact it can be shown that the prediction error sequence e (i) of m that can be shown as: e (i) of m can be written as e (i minus 1) (m) minus k i b (i minus 1) (m minus 1) this is going to be the equation number 8.

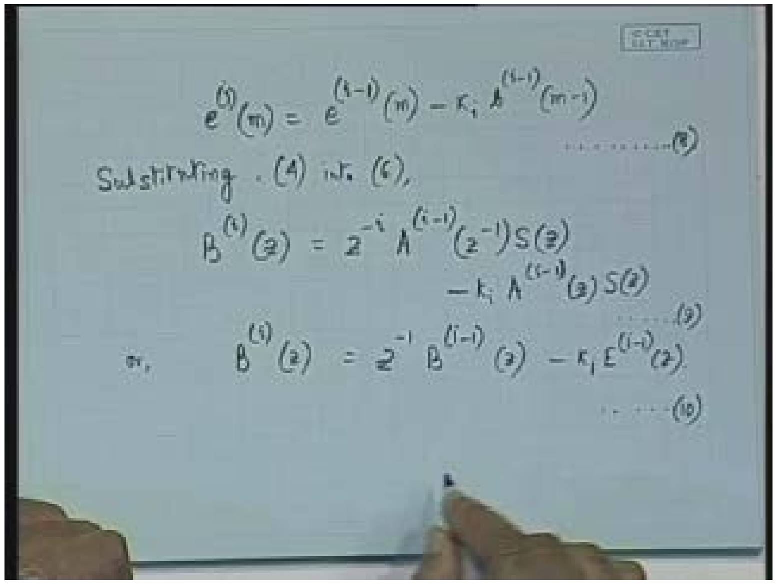
(Refer Slide Time: 28:05)



We are getting this by just simple mathematical jugglery, nothing great about it, but a very interesting fact that emerges is that the forward prediction error; the ith forward prediction error is expressed in terms of (i minus 1)th forward prediction error and (i minus 1)th backward prediction error.

Now what we can do is that......... this is the time domain expression and if we now substitute the equation 4; equation 4 is this one (Refer Slide Time: 28:51), if we substitute in 6, substituting 6 means what; in 6 we are having this A (i) (z to the power minus 1) this form so if we instead of writing A (i) (z to the power minus 1) if we derive it from this equation number 4 then what results is that....................... so substituting 4 in 6 what results is: B (i) of z that becomes equal to z to the power minus i A (i minus 1) (z to the power minus 1) s of z minus k i A (i minus 1) (z) S(z) and we are going to call this as equation number 9 or it could be written rather as in a better form: B (i) of z can be written as: z to the power minus 1 into B (i minus 1) (z); by simplification it is possible to show this B (i minus z) minus k i E (i minus 1) into z; this we are going to call as equation number 10.

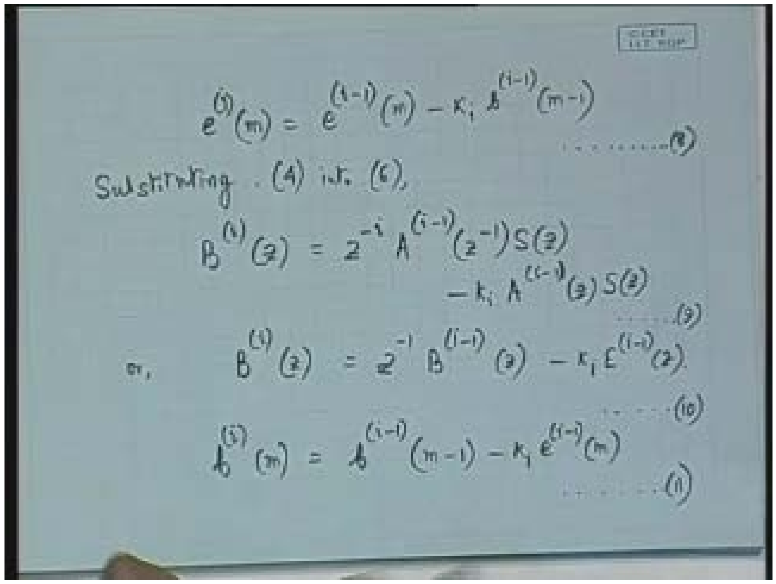
(Refer Slide Time: 30:25)



Even in the z domain also you can identify what we are trying to do. It is the ith backward prediction error that we are expressing in terms of (i minus 1)th backward prediction error and (i minus 1)th forward prediction error. This is in the z domain, it is very clear and just taking the inverse z transform what we will be getting is b (i) of m can be written as b (i minus 1) (m minus 1) yes, because of the presence of this (z to the power minus 1) this one unit of delay.

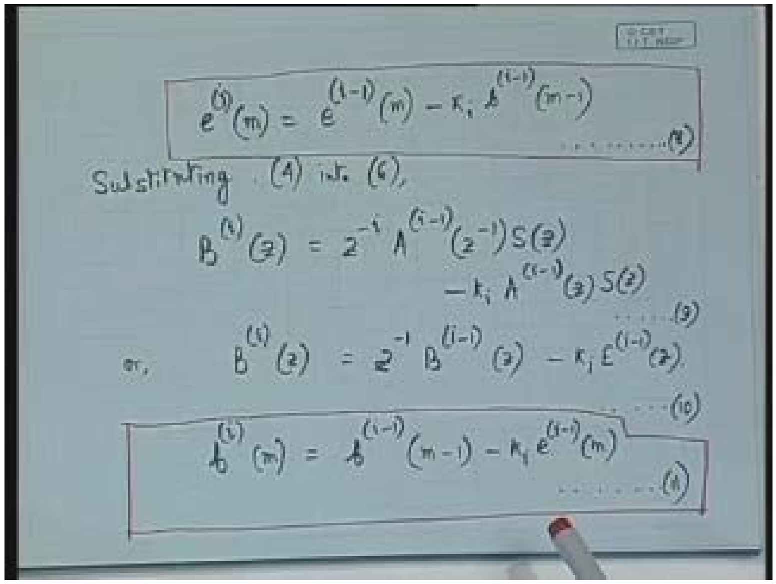
So, in the realization of the filter we had to give that one unit of delay and then this is minus k i into e (i minus 1) as it is and here it is e (i minus 1) (m) simply. So this is equation number 11.

(Refer Slide Time: 31:35)



Now we have got two nice equations at our hand. One is equation number 8; the other is this equation number 11. So these two are intermingled; the ith forward prediction being expressed like (i minus 1)th forward prediction, (b minus 1)th backward prediction and ith backward prediction is expressed as (i minus 1)th backward prediction and (i minus 1)th forward prediction. That means to say again this is a very nice iterative expression. That means to say that if we know a starting point, we can iteratively go up to pth and if we can go iteratively up to pth then we solve the problem because then this is what we are realizing ultimately that in terms of this e (i) (m) and using this b (i) (m) we will be able to express everything.

(Refer Slide Time: 32:58)



So what is going to be our starting point; what is i?

i is the order of the filter essentially and it is the intermediate order of the filter in an iterative filtering process. It basically leads to an iterative filtering process. In the iterative filtering process (i) is the intermediate order we are defining so our starting point could be the zeroth order prediction. And what is zeroth order prediction?

Zeroth order prediction means we are not having any past value. So in absence of any past value what will be the prediction error for s of m? A zeroth order predictor will give you a prediction error equal to s of m that is the zeroth order forward prediction. And zeroth order backward prediction also, s of m. So let that be the starting point.

Therefore, with initial condition that we know e (0) of m and we know b (0) of m we should be in a position to now determine e (1) of m. Can we determine e (1) of m?

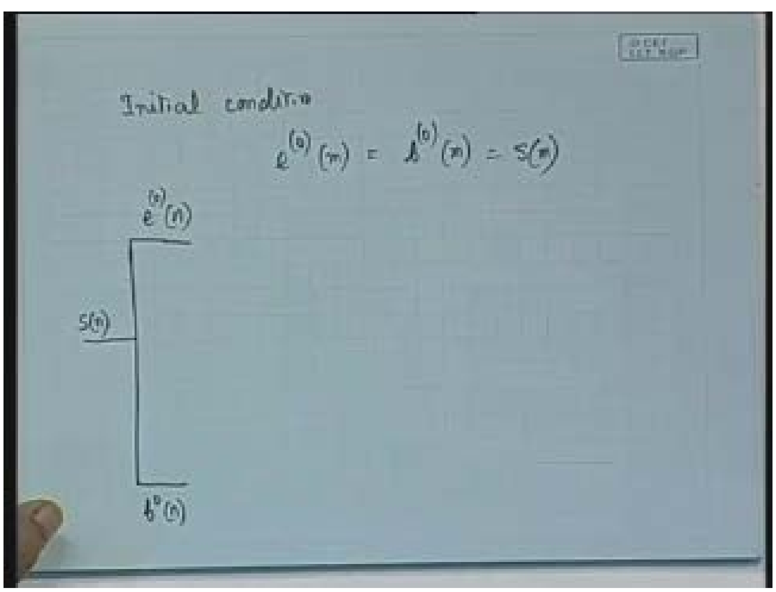
Yes. We now go to equation number 8, we want to determine e (1) of m; we know e (0) of m; e (0) of m is nothing but s of m the sample value itself minus k i b (0) (m minus 1); we know b (0) (m minus 1) because b (0) is nothing but the sample itself that is going to be s of m minus. So essentially what we have is simply s of m e (1) of m becomes equal to s (m minus k i) or in in this case it is going to be k 0 so k 0 into s (m minus 1).

How to get s (m minus 1)?

Give a unit delay and you get s of m minus 1 so you can get e (i) (m) or e (1) (m) is obtained. Again you can obtain b (1) (m) no problem because you know b (0) (m); you know e (0); you know b (0) you know e (0) so you get b (1) and if you know b (1) in that case you will able to determine e (2) and you will be able to determine b (2) and the computations are going on hand in hand.

So, what is the filter structure? Now anybody getting a filter structure in their dreams it becomes a lattice. Hence, in the filter realization what we can simply do is that, given the initial condition that e (0); when for the zeroth order prediction given the initial condition that e(0) of m is equal to b (0) of m is equal to s of m, we can depict these two equations what we have marked with the red (Refer Slide Time: 36:38) this 8 and 11 these together can be depicted in a flow graph like this. You get s(n) as the input sample and s(n) is same as that of or rather just telling it in the opposite way e (0) of n is same as that of s(n) and similarly b (0) of n is going to be same as that of s(n).

(Refer Slide Time: 37:17)



Now what we do is that, b (0) of n to that we put a delay. So we put a delay z to the power minus 1. So what we have over here is b (0) n minus 1.

Now what we want to do?

Let us say that we want to get e (1) of n. To get e (1) of n; what we need? We need e (0) of n and we get and we want to take b (0) of m minus 1. Now b (0) of n minus 1 is already obtained from this. But not simply this: b (0) of n minus 1; in this case instead of writing n we are writing m that is not a problem. But b (0) m minus 1 is multiplied by a coefficient k i. So in this case it is becoming k 1. So we have to multiply this b (0) n minus 1 by a factor k 1; not exactly k 1 rather minus k 1. So what we simply do is that just multiply this by minus by a coefficient minus k 1 and then we are going to add up.

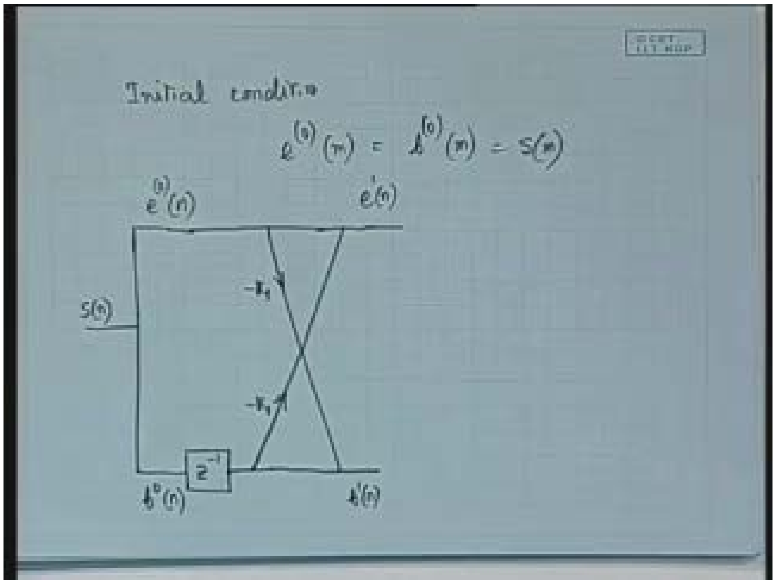
So, if we add up what results over here?

e (1) (n) so here we get e (1) (n). And now let us try to get the other part that is to say b (1) of n.

To get b (1) of n what we need?

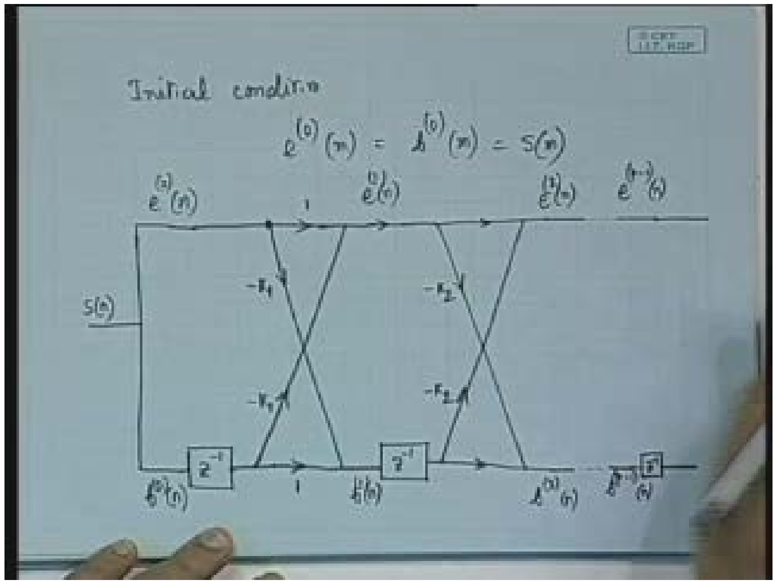
b (1) of n we need b (0) n minus 1, we already have it. At the lower arm we already have b (0) (n minus 1) and we also need to get minus k i e (0) (m) e (0) (n) rather. Now e (0) (n) we already have so what we have to do simply is to multiply that by minus k 1. Hence, this also needs to be multiplied by minus k 1. So we multiply this by minus k 1 and now we get at this end what are we going to call b (1) of n (Refer Slide Time: 40:06). Thus, we have got e (1) of n; we have got b (1) of n.

(Refer Slide Time: 40:12)



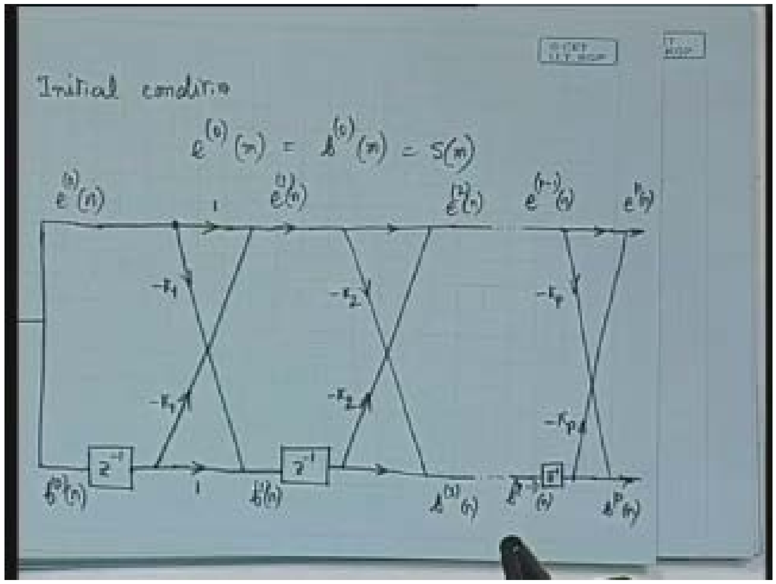
Now the structure is going to repeat. So just for a completeness sake if we are drawing the flow graph in a very formal way we should write here as 1 1 and all these things so you know that we do like that in a flow graph. Now this e (1) n is available and b (1) n is available and now this structure we have to continue which means to say that now there will be another delay z to the power minus 1 and in the next one again there will be a coefficient multiplication but this time it will be not minus k 1 instead we will be having minus k 2 so this will be a multiplication by minus k 2 and in this case also very similarly there will be a multiplication by minus k 2. So in this process you will be getting here e (2) of n and here you will be getting b (2) of n let us write it with a bracket, better; b (2) of n.

(Refer Slide Time: 41:36)



Now we are showing dotted lines just to indicate that a very similar structure is in progress and the last realization will be; in the lattice the last realization will be e of p minus 1 (n) and here we will be having b (p minus 1) (n) and then there will be a z to the power minus 1 and after the z to the power minus 1 there will be................. the diagram should not appear clumsy to you, let me just try to draw it in a clear manner. So this will be minus k p this will be minus k p over here so that at the end here you obtain e p of n and then you obtain here b p of n. So here the filtering ends ultimately using this e p of n and b p of n.

(Refer Slide Time: 42:44)



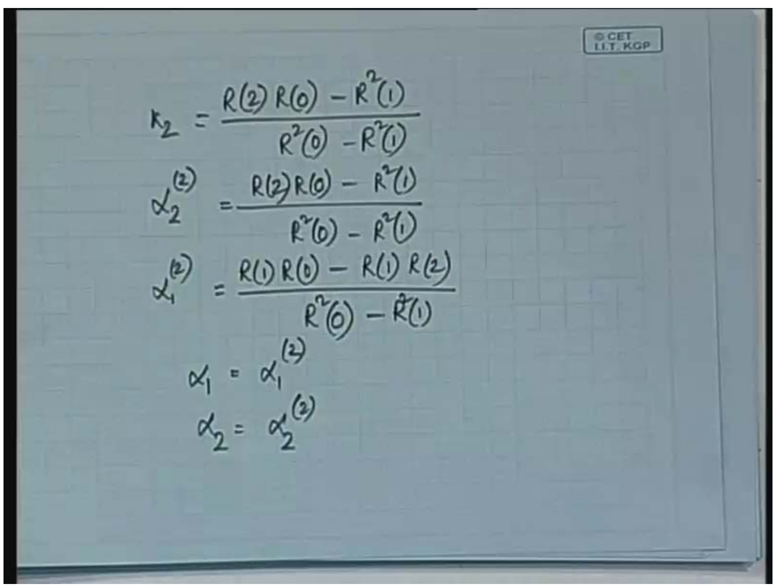
Now, in other words this is also giving you the coefficients. So essentially what we did? As if to say it is Durbin's algorithm only. It is an implementation of Durbin's algorithm, then what is the new thing in it. We have utilized a lattice filter structure for that but essentially it is Durbin's algorithm.

But what advantage is the lattice structure bringing before us? You see, in case of Durbin's algorithm or rather that iterative algorithm which we have got the recursive algorithm for the autocorrelation method there we had a requirement that the autocorrelation values are to be precomputed and then only you will be going in for Durbin's approach of matrix solution.

In this case the two have been integrated into one. You just do the filtering in this manner you get the coefficients. So now what we can do is that, which one is more efficient than the...............yes please; [Conversation between Student and Professor – Not audible ((00:44:18 min))] yes, what are those k 1 to k p values; how to get them? What are the k 1 to k p values? You see, this k 1 to k p values are ultimately this is what; just to go back to Durbin's algorithm; what are those ki's?

ki's etc are the intermediate values. So ultimately this kp's these are going to give you the coefficients. Just refer to your earlier things. Ultimately see you are not interested in e p and b p, you are interested in these things only. So it is these coefficients that one gets. The first iteration is giving us the k 1, the first estimate and then the last one is going to give us the k p. So this k 1 to k p......... let me go back to the old notes then perhaps [Conversation between Student and Professor – Not audible ((00:45:37 min))] this is from there only, I am just trying to take out the old slide. Anyway this should be clear to you when we were discussing about the autocorrelation approach that k p...............

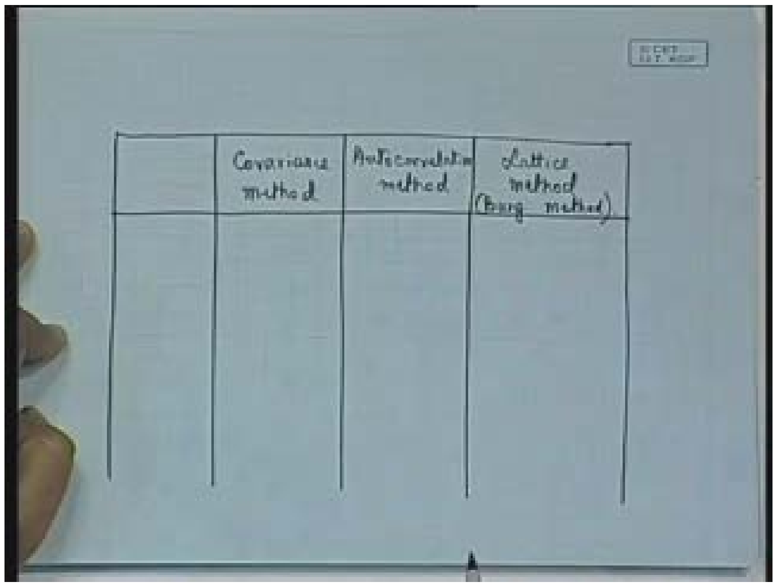
(Refer Slide Time: 00:46:01 min)



Remember, this is what we were doing; k 1 k 2 and all these things. So ultimately alpha 2 alpha (1, 2), alpha (2, 2) these are those values only so it is based on those things. So now we come to some amount of comparison between these methods.

Now going in by the LPC....... now the different approaches that we have dealt with so far there are three approaches so let us make a tabular comparison. Let us write down the covariance method here. This is the autocorrelation method (Refer Slide Time: 47:16) and this is the lattice method. This is covariance means that Cholesky's method and then autocorrelation is Durbin's and the lattice method this is also called as the Burg method.

(Refer Slide Time: 47:49)



Now essentially this question must have come to your mind that then what is the use of the lattice structure because ultimately we are getting this e p's and b p's. Now the thing is that it is these errors; if we encode the errors then it is effectively getting a realization of our LPC parameters in effect.

In one of the coming lectures I will be also telling you about the synthesis aspect that using the LPC coefficients how the synthesis can be done and then this point will be more clear to you. We do the comparison of these three methods based on two considerations: one is the storage and the other is the computational consideration. We will evaluate these three methodologies from these two aspects. As far as the storage is concerned let us first take the data storage requirements.

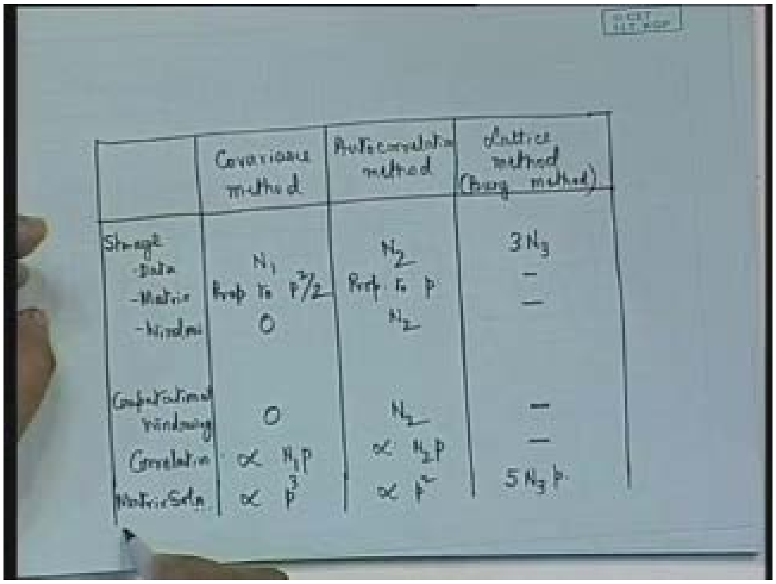
Now we normally go in for an n point analysis. Remember, in covariance method also we were truncating the series to n n number of samples. in autocorrelation method also we were having n number of samples and lattice also ultimately whenever we are inputting the samples there is a certain finite number of samples that we are making into the system bringing into the system so let us call those things as n. but because there are methodologies, the number of points that we will be requiring may not be exactly the same in these three approaches so datawise let us call that let us say covariance method has got N 1 number of points; autocorrelation method has got N 2 number of points and lattice method that has got N 3 number of points but in fact it will be three times N 3.

Now do you know that why it is three times of N 3?
It is because we also need to have the errors to be stored. Here it is three times of N 3 and then the matrix. It is three times of N 3 because you have to store the data, you have to store the forward errors, and you have to store the backward errors. Matrix: here it is proportional to P square by 2 and in the case of autocorrelation it is proportional to p and there is no matrix storage that will be required in lattice method, so here the question does not arise.

Now regarding the window windowing, covariance method uses 0, autocorrelation uses N 2 and this does not require any windowing. And for the computation of windowing; windowing computation here there is no windowing that is involved in the covariance method whereas in the autocorrelation indeed the windowing is there with N 2 number of points (Refer Slide Time: 51:25) and in the lattice method there is no such windowing concept.

And then regarding the correlation; for correlation the covariance method has the correlation computation proportional to N 1 into P and here for the autocorrelation method it is N 2 into p and there is no correlation computation for the lattice method. But for the matrix solution that means to say that for the matrix inversion here this is proportional to covariance method is proportional to p q, very heavy computational burden whereas autocorrelation method is proportional to p square and lattice method has got five N 3 times p.

(Refer Slide Time: 52:31)



In fact this follows from the algorithm; we are not going in to the exact derivation but it is just to give you an idea about what sort of storage and computational comparisons one gets by comparing these three methods: covariance, autocorrelation and lattice method.

Yes please?

Sir, in lattice method it is set to correlation.......... yeah........... there should be the........ there is a p i and these values of p i depend on the other correlations so there must be computational complicity involved with respect to correlation calculation; also every P proportional to N P; proportional to N P; no, there is no there is no correlation computation that is done. Then how will we get the value of k i because to calculate ki's we require correlation coefficients.

I suppose that this aspect I will throw some more light on it; I will be explaining this. In fact correlation is not required but how one can get an idea of this ki's this aspect I will talk of in the coming class because we are already coming to the end of the time for the present class; so thank you for now.