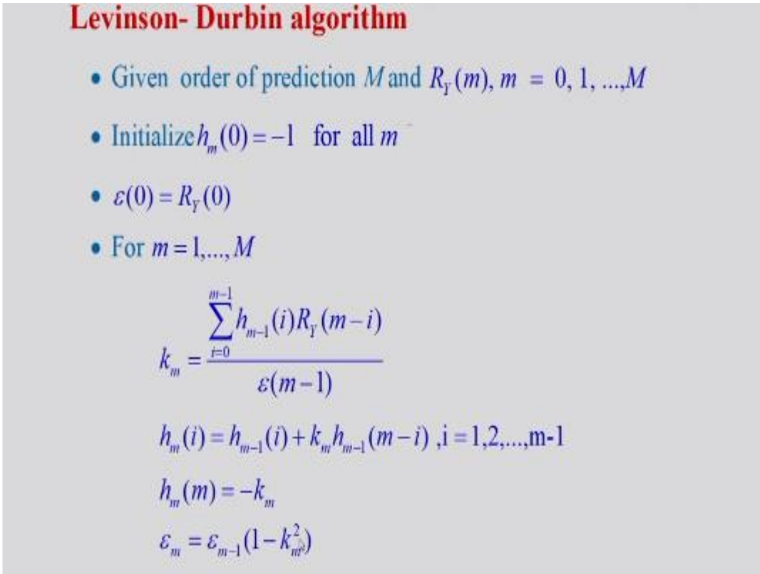
Statistical Signal Processing
Prof. Prabin Kumar Bora
Department of Electronics and Electrical Engineering
Indian Institute of Technology – Guwahati

Lecture – 23
Linear Prediction of Signals 3

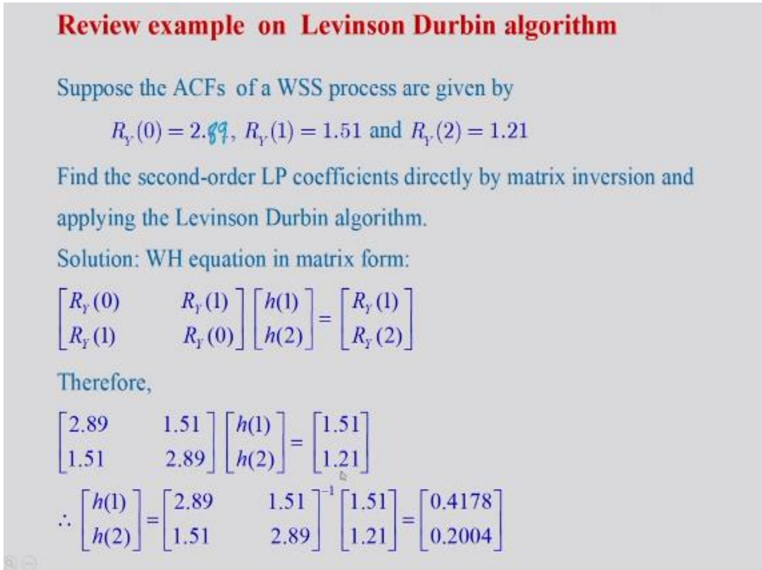
(Refer Slide Time: 00:45)



Hello students welcome to this lecture linear prediction of signals. Let us review Levinson – Durbin algorithm we are given the order of prediction M and the autocorrelation functions Rym m going from 0 to M. First, we have to initialize the predictor coefficients hm 0 =- 1 for all m and epsilon 0 that is the mean square prediction error = Ry 0 for prediction order 0. Now for M =1 up to M.

First we will determine the prediction coefficient km which is given by summation h m - 1 i into Rym – i, i going from 0 to m - 1 whole thing divided by epsilon m - 1 that is the mean square prediction error at stage m - 1 then we will update the filter coefficient hm i = h m - 1 i + km into hm - 1 at m – i i going from 1 to up to m - 1 and the last filter coefficient hm m will be updated as - km and finally we will update the mean score prediction error epsilon m =epsilon m - 1 into 1 - km square. So km we have determined here.

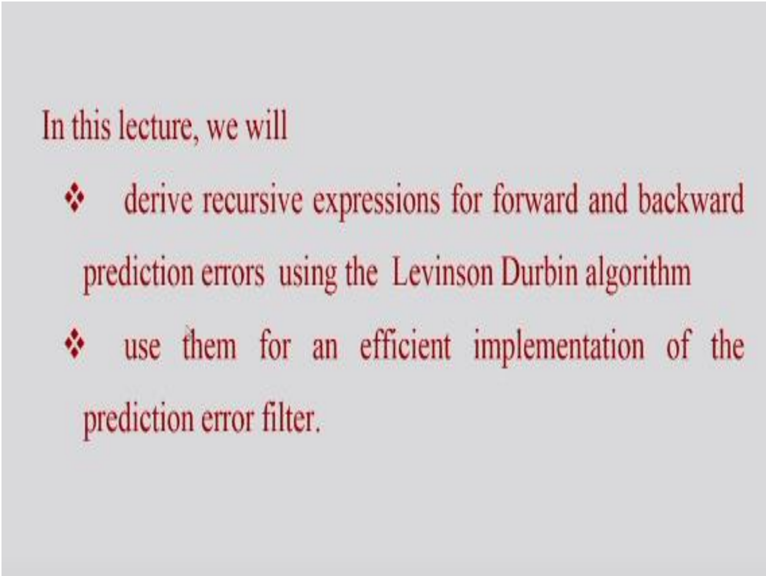
(Refer Slide Time: 02:27)



Let us consider the review example on Levinson -Durbin algorithm in the last class suppose the ACFs of a WSS process are given by this Ry of 0 is 2.89 Ry of 1 is 1.51 and Ry of 2 is 1.21 find the second-order LP coefficients directly by matrix inversion and applying the Levinson-Durbin algorithm. So solution is like this we have to first establish the WH equation.

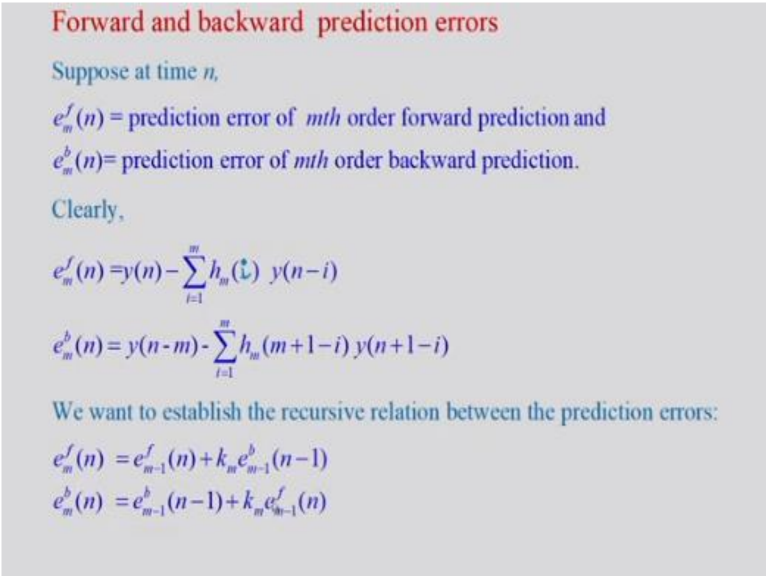
Wiener Hopf equation in matrix form that is given by this autocorrelation matrix into the coefficient vector is equal to the autocorrelation vector so that way here it is Ry0, Ry1, Ry1, Ry0 and similarly coefficients are h1, h2. Therefore, if we substitute these values from here so autocorrelation matrix is 2.89,1.51,1.51,2.89 then multiply the coefficient vector h1, h2 and this must be equal to Ry1 is 1.51, Ry2 is 1.21 and we can directly take the matrix inverse and then multiply with this 1.51, 1.21 and we will get the filter coefficients h1 will be = 0.4178, h2 =0.2004.

(Refer Slide Time: 04:11)



In this lecture, we will derive recursive expressions for forward and backward prediction errors using the Levinson-Durbin algorithm and we will use those prediction errors for an efficient implementation of the prediction error filter.

(Refer Slide Time: 04:11)

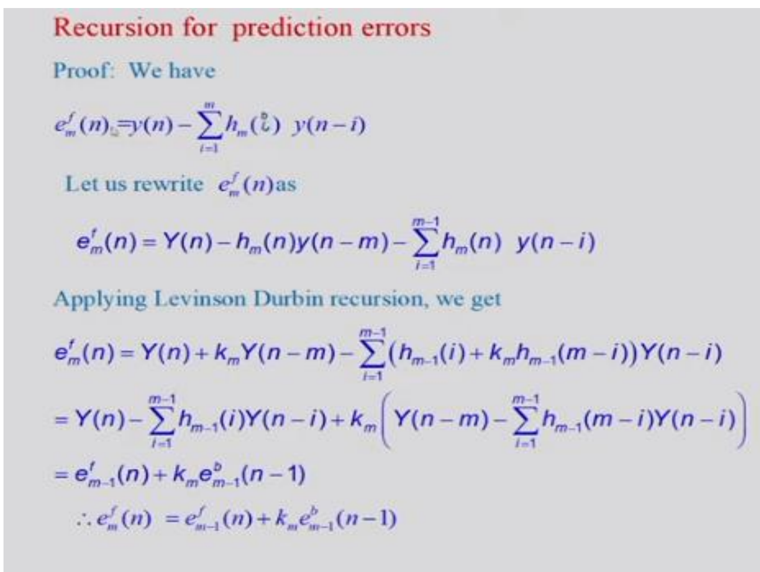


I will derive the expression for forward and backward prediction errors recursively. Let us denote emfn that is the prediction error of a mth order forward prediction. Similarly corresponding backward prediction is given by embn that is the prediction error of a mth order backward prediction clearly forward prediction error emfn = yn - summation hm i into y n – i i going from 1 to m.

So this is the mth order prediction, so the difference is the prediction error similarly embn that is the mth order backward prediction error now we are predicting y n - m by this relationship. So, that way the backward prediction error is given by yn - m - backward prediction, which is given by summation hm m + 1 - i y n + 1 - i i going from 1 to m. We want to establish the recursive relation between the prediction errors emfn = em - 1 fn + k m times m – 1bn - 1 similarly backward prediction error embn = em – 1bn - 1 + km times em - 1 fn.

So that way for our prediction error emfn is given by the forward prediction error for order m – 1, em - 1 fn + m times the backward prediction error of order m - 1 which is delayed by 1. So this is a delayed version of the backward prediction error. Similarly backward prediction error also delayed version of the backward prediction error of order m - 1 + km times forward prediction error of order m - 1 at instant n.

(Refer Slide Time: 07:08)



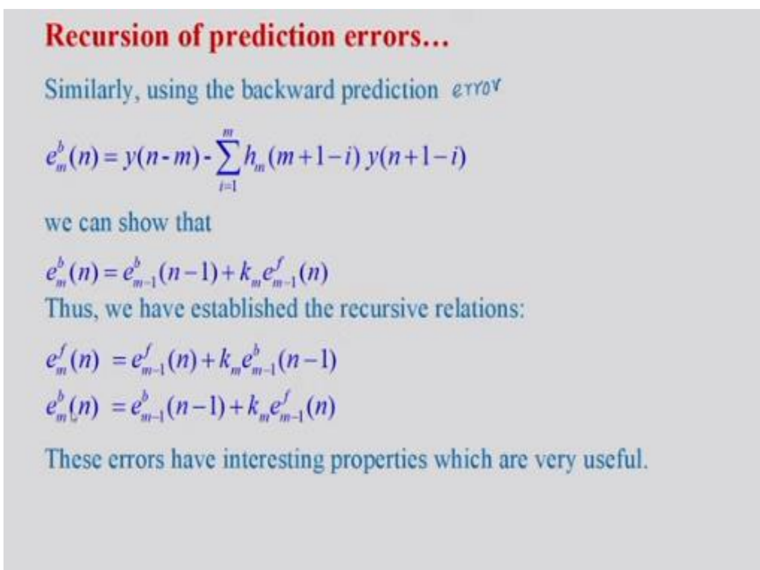
And now we will prove the recursion of forward error emfn = yn - summation hmi yn – i i go in from 1 to m and now we can write emf and as yn minus that last coefficient here I am taking out hm and into yn - m so when we put m here hmm into yn- m that I am taking it out and then summation is same i into y m- i i going from 1 to m – 1. Now we know that this - hmn = km by definition and here we can apply the Levinson - Durbin recursion.

Applying Levinson- Durbin recursion we get emf of n = yn + km into y n- m - summation i going from 1 to m - 1 of now we are substituting hmi = sm - y i + k m times hm - 1 ,m - i so this is the recursion we are using for hmi which is true for i = 1 to m - 1 and then whole thing into yn – i. So this is the relationship we get from here now we can bring these terms containing km together so that way yn minus.

This will bring here, yn - summation hm – 1 i into y n- i i going from 1 to m - 1 + km times yn – m minus this term summation h of m - 1 m - i y n - i i going from 1 to m – 1. Now this part is the forward prediction error of order m – 1. So this expression will be em - 1 fn + km times now this term is yn - m that is the true value, and this is the backward predicted value. So that where this is the backward prediction error at n – 1.

So that way this will be km times em - 1 b and - 1 because this is m - 1 at order prediction and prediction starts at y n - 1 so that way this is backward prediction at time n – 1.So that way we have arrived in relationship that emfn = em-1 fn + km times em-1b at n-1.

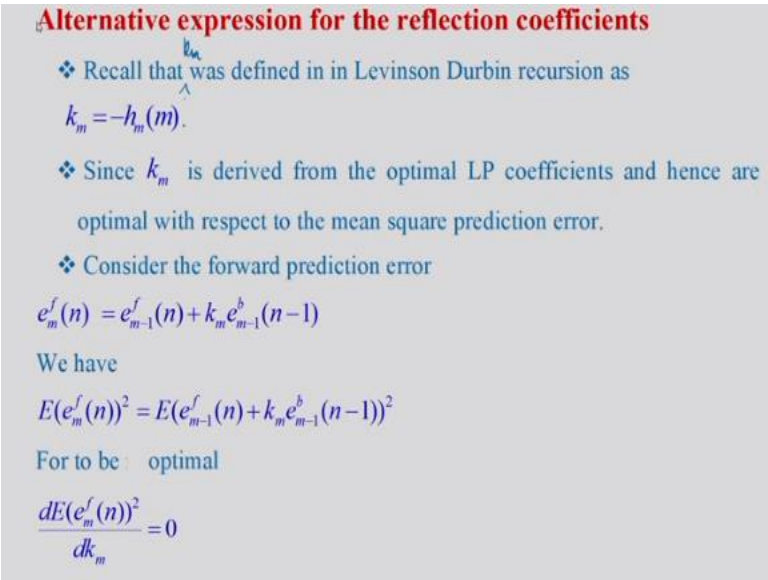
(Refer Slide Time: 10:38)



Similarly, using the backward prediction error which is given by embn =y n - m - summation is same m + 1 - i into y n + 1 – i i going from 1 to m. This is the backward prediction error and now here also we will be using the Levinson-Durbin recursion then we can show that embn = em – 1 bn - 1 + km times em – 1 fn. Thus we have established a recursive relations emfn = em - 1fn +

km times em – 1 bn – 1. Similarly embn = em – 1 bn - 1 + km times em - 1fn and these errors have interesting properties which are very useful for practical purposes.

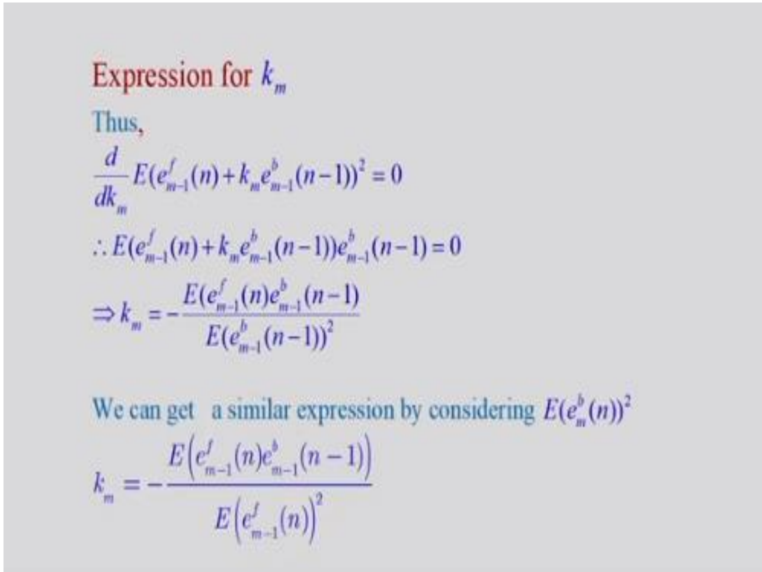
(Refer Slide Time: 11:50)



Now we will derive an alternative expression for the reflection coefficients in terms of the forward and backward prediction errors. Recall that km was defined in the Levinson - Durbin recursion as km = -hm of m. So if it is a mth order prediction that last coefficient is with negative sign is km. Since km is derived from the optimal LP coefficients it is optimal with respect to the mean square prediction error.

Now let us consider the forward prediction error that is emfn = em-1 fn + km times em - 1 b n - 1 we have a of emfn whole square that is the mean square prediction error = e of em - 1fn + km times em - 1 bn - 1 whole square. So this is the forward prediction error and for km to be optimal the derivative of the mean squared with respect to km must be = 0.

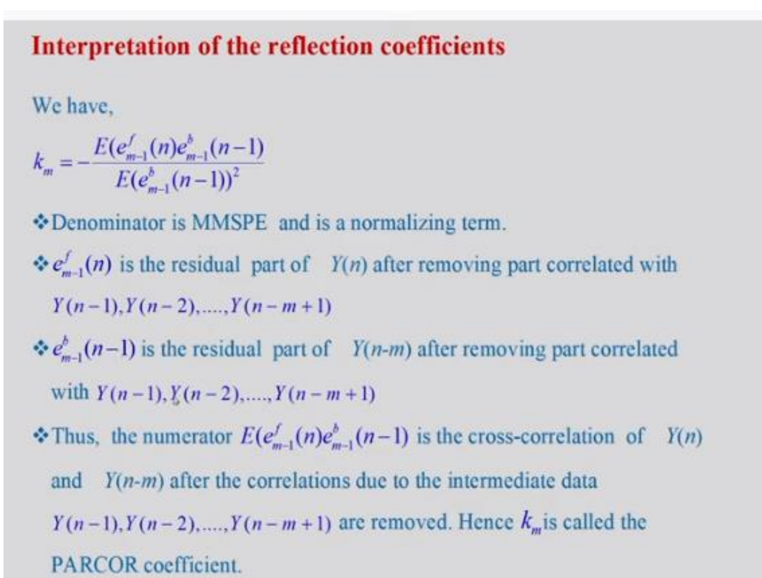
(Refer Slide Time: 13:11)



Thus we get the dkm of E of em-1fn + km times em – 1 bn - 1 whole square =0 so we take the derivative so it is a derivative of this and then again inside it is a function so that way we will get E of m - 1fn + km times e of m - 1 bn- 1 now this expression when we take the derivative with respect to km we will get em - 1 bn - 1 that must be = 0 implies that k m is =- E of em-1 fn into e m – 1bn - 1 divided by em - 1 bn - 1 whole square.

We can get a similar expression by considering the mean square backward prediction error and in that case, km can be shown to be like this. So same value of km we can either get through this expression or this expression.

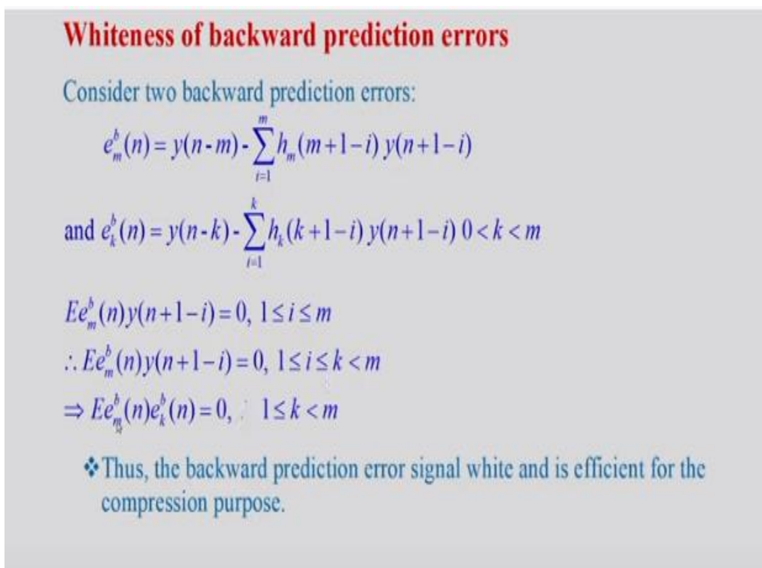
(Refer Slide Time: 14:34)



Now let us try to interpret the reflection coefficient km so km is given by we are considering the this expression km is given by this the denominator is MMSPE and is a normalizing term em-1fn is the residual part of Yn after removing the correlation with Yn – 1 Yn - 2 etc up to Yn - m + 1. So after removing the correlation of the intermediate data whatever residual is there that is em - 1 fn similarly em – 1, bn - 1 is the residual part of Yn - m after removing the part correlated with Y n – 1 Yn -2 up to Yn - m + 1 same set of data.

Thus the numerator this expression e of en - 1 fn into em - 1 bn - 1 is the cross correlation of Yn and Yn - m after the correlation due to the intermediate data already moved. Hence km is called the partial correlation coefficient. So you have remove the correlation because the intermediate data and then you were determining the correlation between em - 1 fn and em - 1 b n - 1 so these are the residuals after removing the correlation due to the intermediate data. So that is right the name is PARCOR coefficient.

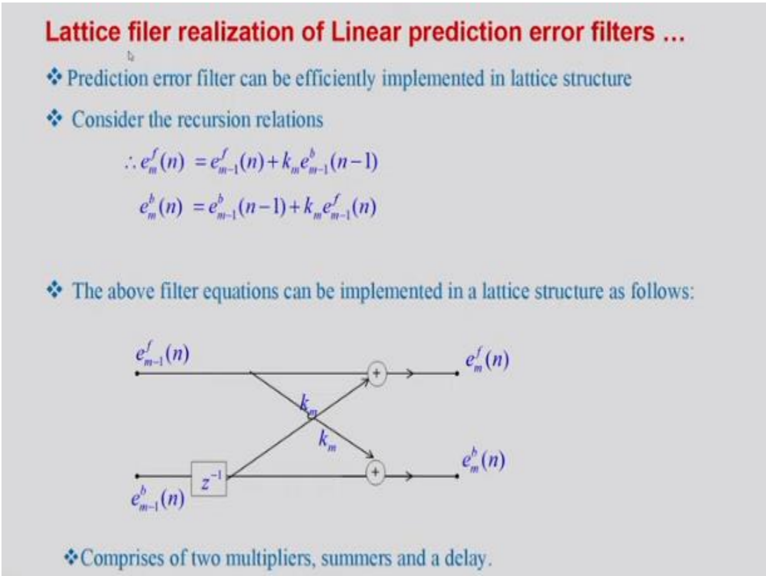
(Refer Slide Time: 16:40)



Now we will prove one important result witness update backward prediction errors consider 2 backward prediction errors suppose embn = yn - m - summation hm, m + 1 - i into Yn + 1 – i i going from 1 to m and suppose at kth order backward prediction error ekbn = yn - k in summation hk k + 1 - i into y n+ 1 – i, i going from 1 to k. So this is for suppose k is some number between 0 and m clearly from this E of embn is orthogonal to yn + 1 – i, i lying between 1 and m because of the linear minimum mean square error estimation.

And therefore since our k is less than m so E of embn into y and + 1 - i will be = 0 for all i less than = k where k is less than m. Now this backward prediction error is the combination of data between 1 and k. So therefore from this expression we will get that this backward prediction error Embn will be orthogonal to ekbn for all l lying within 1 and m because this ekbn is a linear combination of y n + 1 – i, i lying between 1 to k. Therefore, expected value of Embn into ekbn will be = 0 for k lying between 1 and m thus the backward prediction error signal is white and is efficient for the compression purpose.

(Refer Slide Time: 19:00)

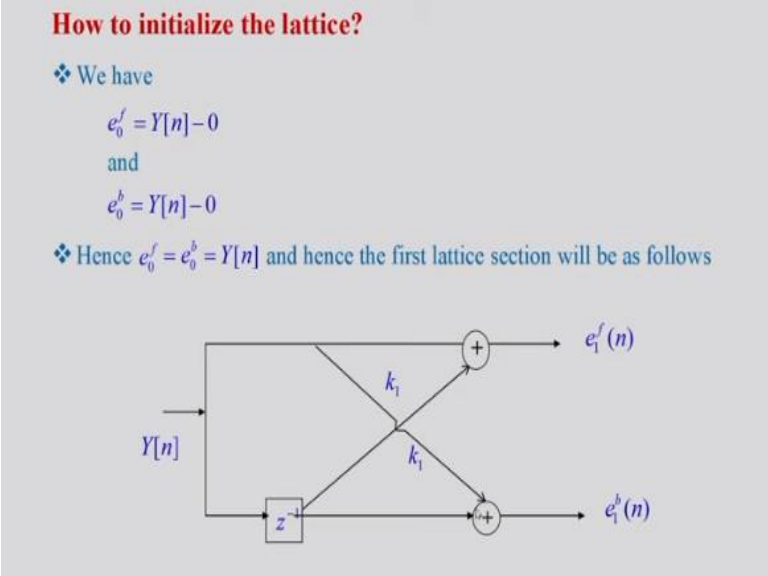


Now let us see Lattice filter realization of linear prediction error filters, prediction error filter can be efficiently implemented in lattice structure consider the recursion relations emfn = em – 1 fn + km times em - 1 bn - 1 and similarly embn = em - 1 bn - 1 + km times em - 1fn. The above filter equations can be implemented in a lattice structure as follows. Because this is the output of the filter section that is emfn and input is em - 1 fn + em - 1 bn – 1.

Therefore emfn is the sum of e m - 1fn and km times em – 1 b n-1. So this we can write using filter elements like emfn = em - 1fn plus this is the delayed version of em-1bn. So here output will be em – 1 bn-1 and that will be multiplied by km and it will be added so that way emfn is the sum of this signal plus this signal delayed by 1 unit and then multiplied by km. So that way we see that this expression can be implemented by this part.

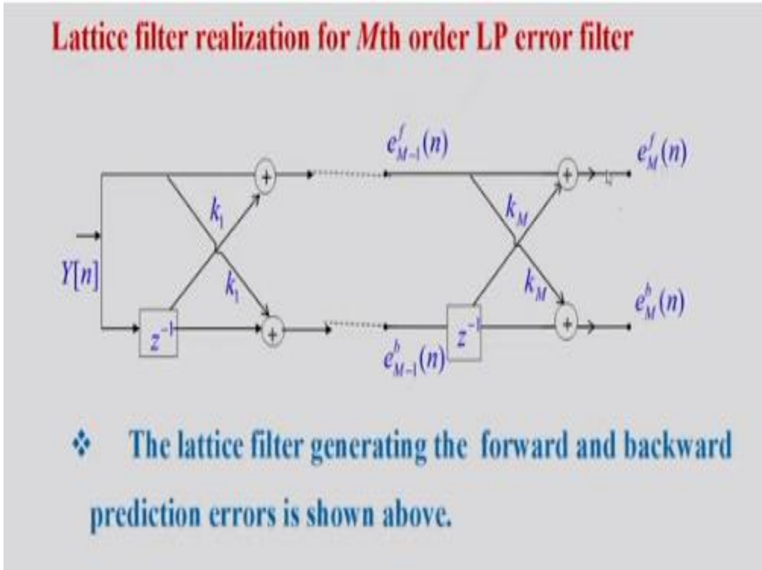
Similarly this backward prediction can be implemented embn is equal to this delayed backward prediction error this plus the forward prediction error multiplied by km. So that way embn will be equal to this part e m-1 b n-1 because it is delayed here + em - 1fn multiplied by km and that will give us embn and so this is a simple implementation because same multiplier is used here and there is there somewhere here somewhere here and then this delay unit. Thus we have seen that this recursive relation can be implemented by this filter section.

(Refer Slide Time: 21:58)



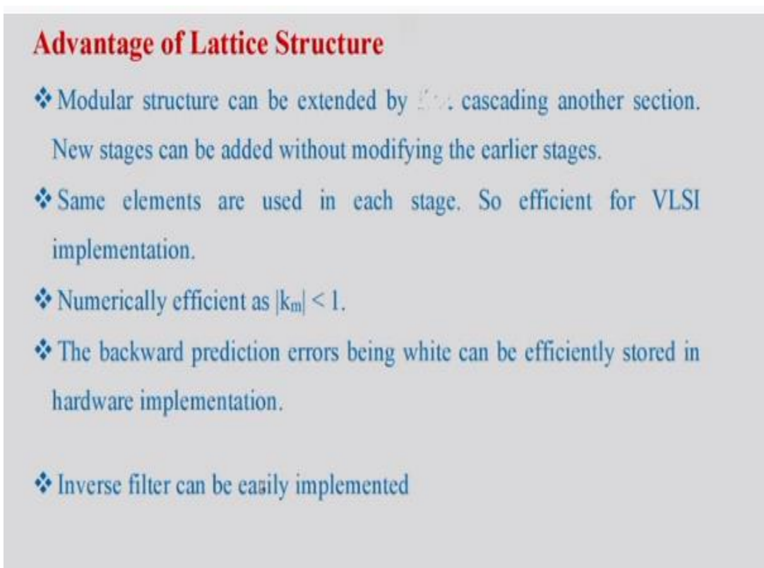
How to initialize this lattice we know that suppose derivation order prediction in that case forward prediction error will be Yn - 0 that is =Yn similarly backward prediction error also will be = Yn - 0 that = Yn hence ef 0 = eb 0 is = Yn and this can be the initial input and the first lattice section will be as follows. So Yn it will be passing through this path it will be passing through this path and then when delayed version multiplied by k1 is added here you will get e1fn and similarly when delayed version and this part Yn multiplied by k1 is added here this sum will be you e1bn. So that way initialization also we have seen.

(Refer Slide Time: 23:02)



Therefore complete filter can be realized by cascading all these sections like this. So input will be Yn here and then first checks on output will be suppose e1fn and here e1bn and like that suppose last section because we are interested to find out the mth order prediction errors, So both forward and backward prediction errors will be obtained by the last section. So this last section input will be em - 1fn + em - 1bn thus we can get the forward and backward prediction errors using this lattice filter structure with yn at the input.

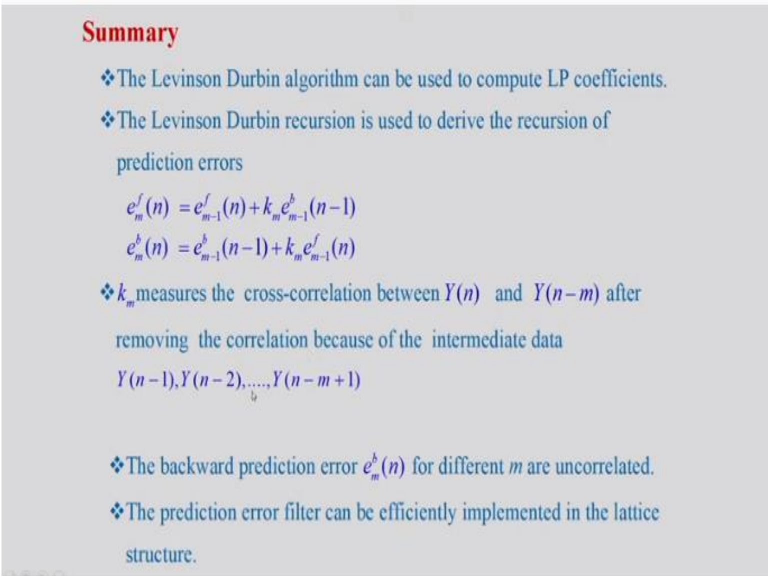
(Refer Slide Time: 24:01)



So this diagram shows the latest filter implementation the Lattice structure has several advantages it is a modular structure it can be extended by cascading another section. New stages can be added without modifying the earlier stages. Same elements are used in each stage that is multiplier and some are used so efficient for VLSI implementation. It is numerically efficient because this came coefficient reflection coefficient magnitude is always less than 1 therefore quantization error will be less here.

The backward prediction errors being white can be efficiently stored in a in Hardware implementation and inverse filter can also easily be implemented these are some advantages of this lattice realization.

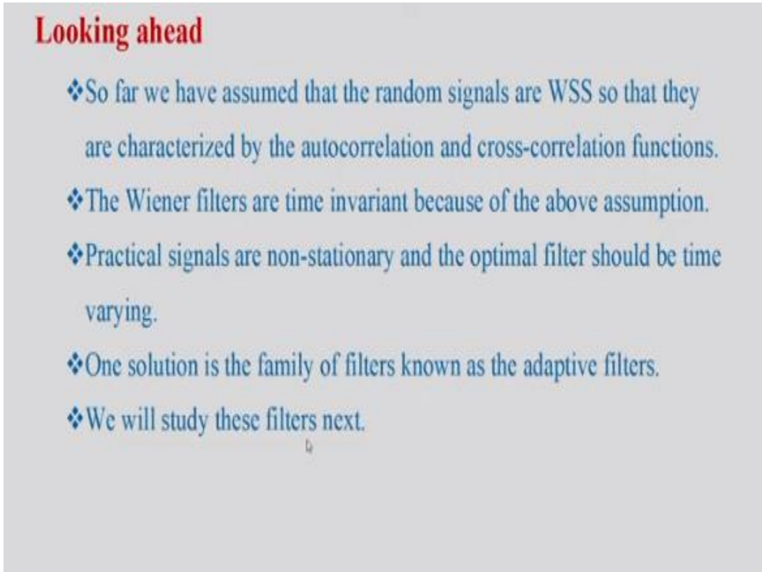
(Refer Slide Time: 25:00)



Let us summarize this lecture the Levinson-Durbin algorithm can be used to compute LP coefficients. The Levinson Darwin recursion is used to directly recursion of the prediction error these are the expression we obtained. The reflection coefficient km measures the cross correlation between Yn and Yn - m after removing the correlation because of the intermediate data Yn – 1 Yn - 2 up to Yn - m + 1.

The backward prediction error embn for different m are uncorrelated that also we established. The prediction error filter can be efficiently implemented in the lattice structure.

(Refer Slide Time: 25:51)



Looking ahead so far, we have assumed that the random signals are WSS wide-sense stationary so that they are characterized by the autocorrelation and cross correlation functions. The wiener filters are time invariant because of the above assumption. Practical signals are non-stationary, and the optimal filter should be time varying. One solution is the familiar of filters known as the adaptive filter we will study these filters next. Thank you.