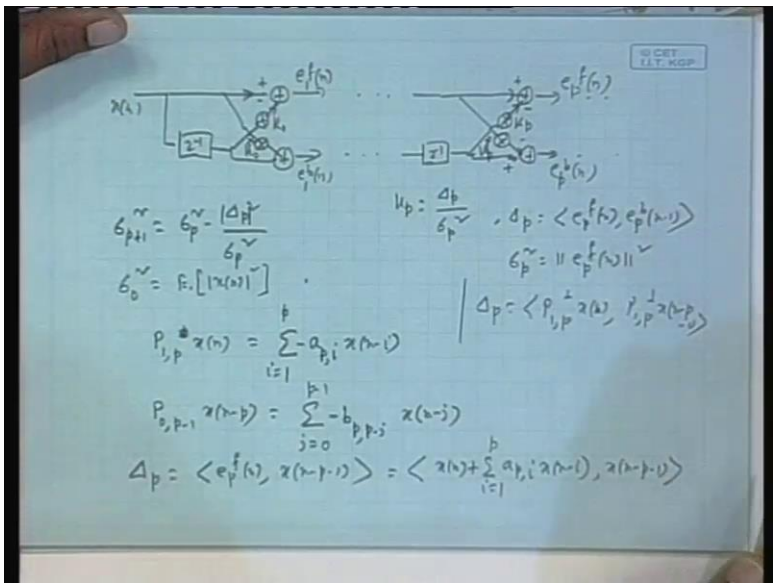
Adaptive Signal Processing
Prof. M. Chakraborty
Department of Electrical & Electronics Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 24
Lattice as Optimal Filter

(Refer Slide Time: 01:00)



Let us start with what we did last time. We had the lattice filter. It is had stage. This is just recap part actually. This has the coefficients K 0, both the coefficients are same; one is the conjugate of the other; is it not? K 0, K 0 star. This is your e 1 f n; this is your e 1 b n, dot, dot, dot; dot, dot, dot, in general. Dot, dot, dot; this is your epfn, epbn, Kp, Kp star, where Kp is delta p by sigma p square because both the forward and backward prediction error are same; variances we have seen.

So, I am not writing sigma pf square and sigma pb square; no point, I told you that I will drop that index f for a superscript f (b). But, delta p was inner product between epfn, ep these are all standard stuff; I mean, what we have discussed. Sigma p square is norm square of any of the 2. So, once the multiplied coefficients which are also called, either also called reflection coefficients.

Kp is the pf stage reflection coefficients. Once these are known you can always construct the lattice and go on filtering it; x n you really get the prediction error for varies orders, for both forward and backward. But before that you have to obtain these multipliers and all that; what will be giving is the statistics of the process; in the sense the, of course, x n

is 0 means it will give you only the correlation, auto correlation function; or atleast many values of the auto correlation functions. Or, I mean, many values means auto correlation functions for many lack values; out of using that only you have to construct these.

Then, for sigma p square you have found out; this also that was the recursion that sigma p plus square was minus, delta p square, not delta p square, divided by sigma p square, right. This is the one we did, is it not? But then, that time I told you; and, of course, how to start the recursion, sigma 0 square is nothing but e of mod x n square because e p 0 f n is x n. So, therefore, is e p 0 is a for e p 0 b n is both are x n. So, just norm square of that, means, variances of that, but how about delta p; that is the question? Because, you can go into the recursion, provided you also know delta p, the update delta p, because this sigma p plus 1 depend not only sigma p, it depends upon the delta p also. So, I have to update delta p also. From delta p to delta p plus 1, then only the recursion can be proceed. So, that is what we will be doing now.

Now, I define this. So long we have been discussing the prediction errors. Also remember, we have been doing, I mean this prediction and all that, but what is the utility of the lattice and all, that still has not got. We have got a structure. What is the purpose of structure and all, we will let you know all. I will study its property and all. I mean its utility in various forms; will leave on that with you.

Now, so long we have been discussing when we have only the prediction errors, our projection errors. For a while, I will now considering not the error, but the projection only. That is x n, as I earlier I projected x n on the space span by x n minus 1 to x n minus p to the error, now I will not take error for a while, and that is the projection I mean for a while. So, that means, I will be considering things like this; p 1 p not perpendicular, p 1 p x n; getting the error is very simple, just subtracting from x n; is it not?

This will be nothing but a linear combination of those elements, x n minus 1 to x n minus p. So, it is a linear combination will be like this, i 1 to p. Then the coefficients, for some reasons I will introduce a minus sign in these coefficients; it is I just put a minus also. And coefficients will be called, denoted by a, or subscript will be i, but also with the p th order prediction. Suppose, instead of p, I use p th order prediction I will get a new set of coefficients; is it not?

So, that is why I am putting p subscripts, p comma i. These are also; I mean, you see, these are independent of n; left hand side is function of n. You can say that how about n? n goes simply because of our logic of stationarity; remember this are nothing but optimal filter coefficients after all. When you now give x n as linear response, and x n minus 1 sequence of the input; obviously, the filter coefficients will be given by this combiner coefficients, because of stationarity.

You know there exact values also, but I will not go for that calculation that R inverse p and that here; I will recursion updating that. So, they are independent of n. Similarly, this is for forward prediction; for the backward prediction we have p 0 p minus 1, and x n minus p was projected; x n minus p was projected on the space span by p future terms; p future terms are standing at n minus 0 that is x n; x n, x n minus 1, upto x n minus p plus 1.

So, that projection, instead of the projection error again I am taking the projection; that again will be a linear combination of those elements which has erring the space. So, introduce j; the range I will write later. Suppose, you write n minus j, so obviously, j has to go from 0 to p minus 1; a minus sign again I put deliberately.

Coefficient side denoted by b, but I will just, you know, to confirm to notation of the thing that is used, what I will do, first co index will be of course p because it is p th order prediction; instead of having second index as j I will call it p minus j; that is I am just naming this, this way. When j equal to 0 that is x n, this should be called b p p; while j equal to 1, x n minus 1, this should be called b p p minus 1.

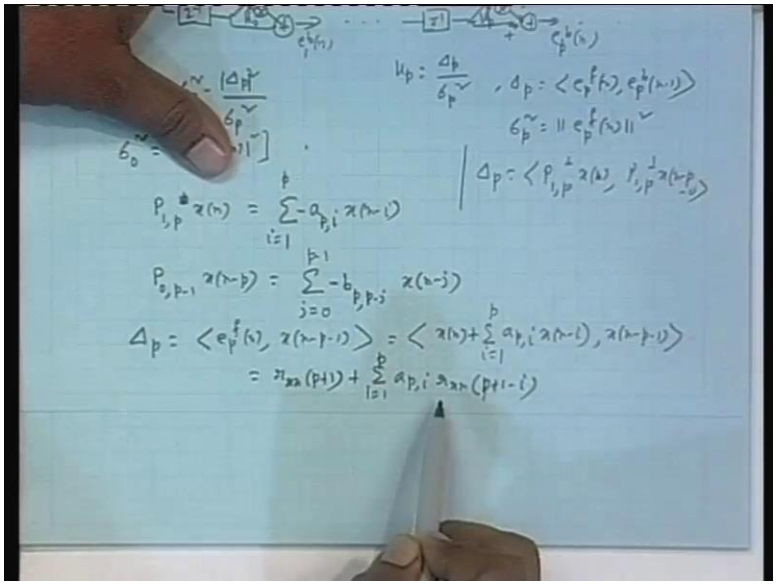
Unlike here, a p 1 for x n minus 1, a p 2 x n minus 2, a p 3 x n minus 3, here, my, the person I am projecting is n minus p. But I am not taking the coefficient with n minus p plus 1; there is the immediate neighbor as b p, 1. I am taking that to be e p comma 1, infact. You understood that clearly; x n projected on space span by x n minus 1, n minus 2, upto n minus p, the nearest 1 n minus 1, put the index 1 for that, next one put the index 2 for that, similarly n minus p.

So, when it is n minus p plus 1, that is when j equal to p minus 1 here, that time I want 1 to be here; project this p minus 2 that time I want 2 to be here. It is just a notation that is a p minus j. Suppose, these are known, the p f stage we are known, then obviously, delta p, what is delta p? Delta p we have seen already; but delta p can also be written as; can I? I am being doing this again and again.

What is e p? You understand how it comes? Those who are forgotten, for them, I am saying, I am writing here. Delta p is nothing but, if you, you can write this e p f n as p 1, p perpendicular x n; and this is as p 1, p perpendicular x, is it not? I am being doing it again, again, again. And then, I will just drop this. Logic, you all know; several times I have explained. And then, again you call it epfn, that is what I have done here. And, epfn is what? X n minus the projection.

What is epfn? X n minus this quantity. This I would write here. X n and then it becomes plus, that is why I put a minus here; x n minus this; and, minus and minus plus comma; and what is inner product? Inner product is correlation. So, what you can do? You take out this term; take out correlation of this term; forgot to, take the correlation; and what will you get in the first term? X n into x n minus p plus 1; that gives what?

(Refer Slide Time: 11:22)



In terms of correlation, what is it? Using inner product, inner product means what, e (x n) into x star, n minus p plus 1; that is the difference of the inner product; that is correlation. What is the lag? P plus 1. And then, other ones, a p remain as it is; this and this, this will be R x x, what, n minus i x minus i into x star n minus p minus 1. So, you know the lag that will be p plus 1 minus i.

So, if you know this coefficients a p i's, and if you know all the correlation values of the process x n, then you can find out delta p. So, that is how delta p is to be computed, mind it. There is no past algorithm, recursive way of updating delta p directly; recursive way means efficient, computationally efficient.

Here you see, so much of computational is equal; as the value of p increases because you go higher stage number of operations in this because more and more, but this is something have to be I mean you cannot help. There are ways to paralyses this computational that I am not going to, but surely I will not go into recursive algorithm; but you understand, this computational I mean depends on p, and this you have to carry out. Sigma p square could be updated recursively, very simply. You do not want to really carry out sigma p square computation elaborately and explicitly all terms; just recursively.