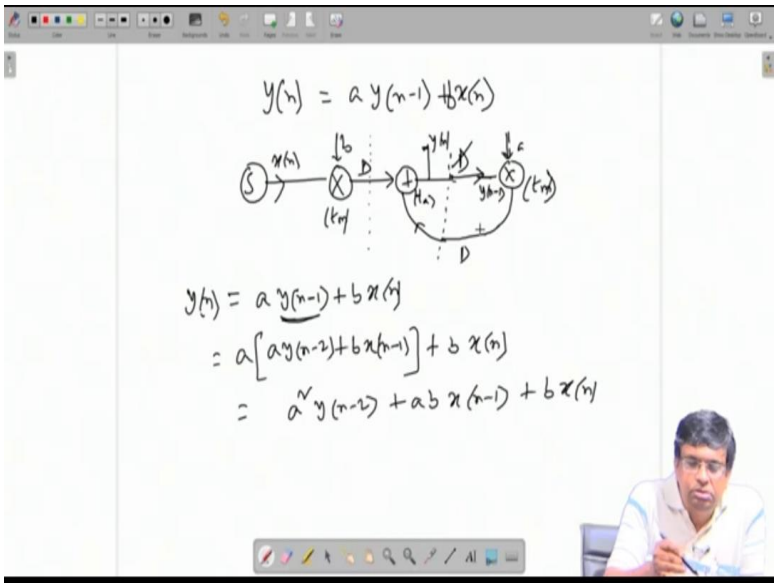
VSLI Signal Processing
Professor Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur
Lecture 09
Retiming IIR Filters

Okay, so in the last class we were doing retiming. We took the example of an IIR filter. Something like this or just simple first order IIR filter.

(Refer Slide Time: 00:34)



Okay, so if you see a signal flow graph or data flow graph, it does not matter how you draw it here now. The signal source it is generating x of n then you are it multiplying it by a factor. Then you can put a b here, b. So b into x n, then you add b into x n plus this is the plus and a into something. So there is a multiplier with a, a into whatever comes, maybe y n minus 1 comes that into a. When you add with this, this would be by this equation y n.

So if you assume that y n minus 1 is there already. It was calculated correctly in the previous cycle, so just simply it gets multiplied by a and gets added here with b x n, then by this equation that will be y n. The correct y n will be produced and it will continue cycle after cycle. If n plus 1 is cycle, correct y n will be here multiplied by a, added with the current new input and all those things okay. We have done these kind of things earlier. So there will be a delay. Here is your y n.

Okay, now what is the critical path? So this is taking time t m, okay multiply. This is t a, this is tm. Now t m plus t a or if you take this side, you cannot go any further. So if you just t m plus ta so here I can do, just I can do a cut-set retiming. This is one side of the, one sector of the DFG. This is another sector. This is only one direction, no reverse direction. So I can add any number of delay in this direction. I do not have to take out same number of delay from the opposite direction so you just put a delay here.

So now this multiplier and adder they are not in the same cycle because the delay is separating them. So t m just if I had only this much. This node and this node, nothing else, then you have to take the larger of the two, t m or t a, you have to take t a. But now you have this side also, here if you start at this, this is one output line but this is delayed. So this plus and this multiplication, we are not coming in the same cycle, we are separated by delay.

But in the lower half you have got t m multiplier and then same cycle you are following up with the adder t a. So t m plus t a. So here I have t m, here I have t m plus t a and here I have t m, here I have t a, here I have t m. Its node also the edge. Its node uses the path actually okay may be one shortest possible path. So t m plus t a. So I want to bring a delay here. So if I do cut-set retiming, this side is one sector. If you take out these 2 edges, this side is one sector, this side is on a single node another sector.

From left to right, I have got a delay. From right to left, no delay so you can take it out and you have to bring delay here. So in this case this multiplier and the adder which were earlier in the same cycle not separated by delay, they are not separated by a delay. So it will not be t m plus t a from this lower half but for the upper half, it is becoming delay free. So t a plus t m again. So critical path will turn out to be t a plus t m.

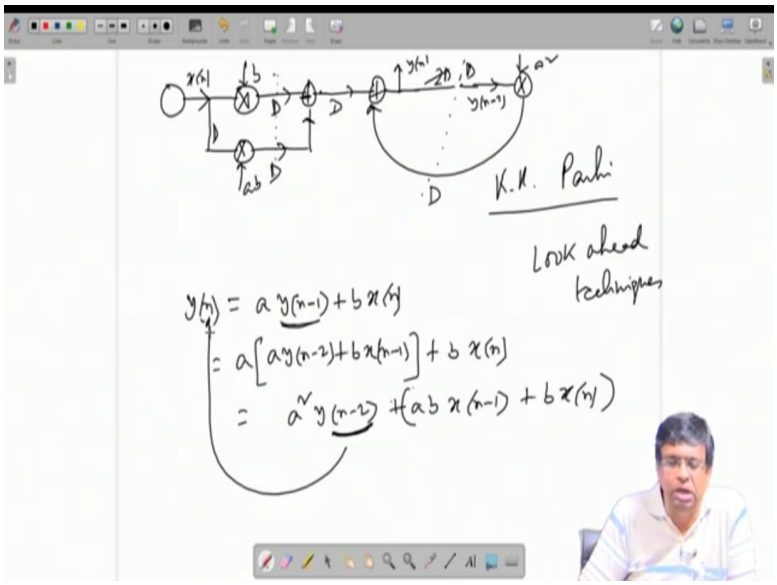
This happening because as you know in retiming a loop. There is a loop here total loop has to remain constant. So earlier, I had one d in the upper half, lower have no delay. So total loop delay was 1. So that has to be preserved. That is why now there is 1 delay in the lower half and no delay on the upper half. So loop delay remains 1 same, okay.

Alright, but can I bring down the critical path by some trick? The trick I can play here you see. Look at this equation y n is equal to a times y n minus 1 plus b x n, y n minus 1 I replace the left

hand side by n minus 1. So this becomes n minus 2, this becomes n minus 1. When this y n minus 1 from the same equation can be written as a y n minus 2.

When it is n, it is n minus 1, when it is n minus 1, it is n minus 2, a y n minus 2 plus b, x n minus 1. This is this spot from this equation. The left-hand side if it becomes y n minus 1 right hand side is this path, plus as before b x n. So what you have is, a square y n minus 2 plus a b x n minus 1 plus b x n. Now if you see the, if you draw the DFG or signal flow graph, you will see the loop delay will be increased to 2 not 1, 2 which will come to our advantage. Okay, how to have this equation? If you have this equation now, you see I will erase this figure.

(Refer Slide Time: 06:51)



And I will just draw the corresponding DFG for this. So x n, signal source b x n, then x n minus 1 a b, a b they are all known beforehand so b or a b these factors are known beforehand. There is a delay. They get added. Okay, a b to b x n and a b x n minus 1 get added, so I got this part. Now with this I have a plus so this is the plus. This much with the plus so plus, plus with what? Some a squared times y n minus 2. Suppose y n minus 2 is here. Suppose by some way I have got y n minus 2 here. So simply I have to multiply by a constant a square and bring it back here. If we add, this will become your y n and to complete the loop you bring in 2 delay here, okay.

Now as we watch, I have a cut-set retiming, the signal path in one direction. Okay this is 1sector, this is another sector. I happily bring one delay here (sorry). Okay so on this side and here also I

can do a cut-set retiming. Okay so I can bring then this is also forward direction, forward direction. So this is one side, the left hand side is another side. I have only 1 direction, 2 edges so I can happily add any delay here. So I bring D here, D here okay. So the upper half or lower half if you go up to these, the multiplier and adder, they are not coming in the same cycle separated by delay. Here also separated by delay. So t m and t a or tm and t a I have to take larger of t m up to this, larger of t m and t a which is t m

Then again I cannot go further, there is a delay. They are not coming in the same time. Now comes here. Here I have got t m plus t a okay t m plus t a. Here nothing because after adder there is a delay 2D. Earlier I had 1 D but because of this trick, it has become 2 D. So if I now do cut-set retiming I just reduce 2D to 1 D that is take out 1delay from this side, bring another delay on this side okay. So now you see both this path is delayed, this path is delayed. So adder and multiplier here or multiplier and adder there but coming in the same cycle.

So we are not delay free path. Delay, delay, delay, delay, delay, delay which means on that note which takes maximum time there is a multiplier that I have to take for the critical path. So it is just t m. Earlier it was t m plus t a. So this is how I can bring down the critical path but here I am paying a price of course that is on the input side earlier I had only 1 line that is x n into b. It was going directly here. Now I have got 2 lines x n b, x n into xn minus 1 into a b.

So 2 multipliers. We are parallel but I am paying more in hardware. This is the price I pay more in hardware and power but I gain in speed because I could bring down the critical path okay. This kind of technique you can see in detail in Parhi's book, Parhi as I told you KK Parhi. They are called Look Ahead technique. That is if you are standing at this index and looking ahead at a future index n. I have got a delay between them, then the delay, the 2D that could be used for doing the retiming okay.

Parhi has actually developed one full you know I mean direction. One full way I mean topic of developing pipelined, designing pipeline IIR filters okay by various look ahead techniques. This is one. I pay in hardware but I gain in speed. We want good example. Okay I will take another example but then that is very involved. That is in the context of adaptive filter and since I do not

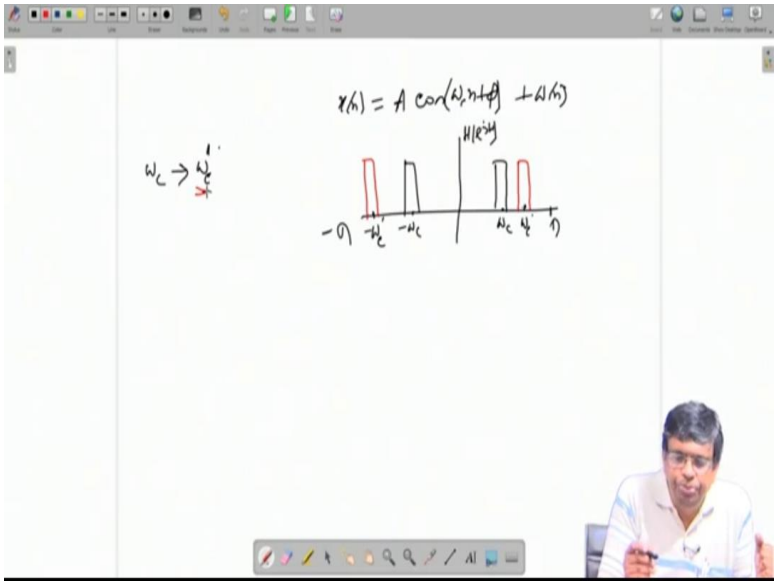
presume any background in adaptive filter, I will also develop, I will also tell you about adaptive filter equation all those things.

I will not derive anything here because it is not a course on adaptive filter but then unless I really explain the background of adaptive filter, basic equations and all then you will not be able to appreciate it. So we will do that side by side. Okay, so far in digital filters what you do? You design a filter to a particular pass band or I mean stop band and all that.

You design filter either by FIR method or IIR method. Once you design them, they are fixed, the coefficients are fixed. Then you construct the filter, develop a circuit okay and it will remain as it is all along for all future, okay it will continue to do filtering of the input producer desired output. But suppose your purpose of filtering the surrounding or all that you know it changes, the requirement changes. Then you cannot throw as the filter and redesign another filter put it in the circuit again.

Can you, I mean, when the filter adjusts itself to the new reality, new surroundings okay so that it remains the best filter, desired filter under the current situation. To give an example. Suppose you have got 1 input sinusoid okay particular frequency and if there it is added with lot of white noise, you know, high power up I mean, high level of noise okay, there is a high noise power. You want to filter out the noise as much as possible and pass that sinusoid.

(Refer Slide Time: 13:49)



Okay that is digital domain when we have got a single x n which has got some a Cosine some constant, you know, digital frequency plus some phase phi and some noise w n. W n is noise, white noise with I mean appreciable noise power. So this signal is buried in noise. You want to recover the signal. What will you do? You will develop a band pass filter, you will design a band pass filter, narrow band around omega c and minus omega c. Very narrow band.

If this is a filter transformation like this it will be like this. So the signal will pass through. Most of the noise will not pass through because noise is white band, white noise. So since the bands are very small, very narrow, most of the noise will remain outside the band during filter out.

Very good. You design the filter once for all be happy with this, as long as you get this kind of signals it will be a beautiful band pass filter, but now after some time suppose the frequency of the input changes. Omega c changes to a new frequency omega c phi but you have not designed a band pass filter for omega c Phi as the pass band frequency. Okay so your band pass filter which you designed and constructed that will remove the signal also because omega c phi is outside this band.

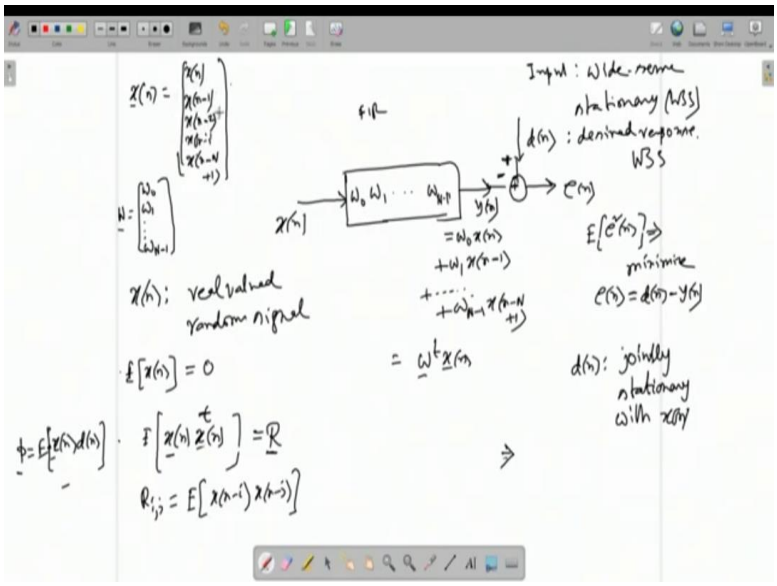
But can I make it adaptive? Can I make it self-adjusting? So that by looking at the input signal and doing some calculation, it can identify what it should do, what should be the desired response and change the coefficients, change the filter coefficients okay. Maybe in the iterative

manner so that eventually you get, eventually this filter becomes a new band pass filter because we are changing the coefficients internally.

The self-adjustment process, the filter coefficients are changing, clock after clock after clock in the iterative manner. Eventually, it should become another band pass filter but its characteristic should be like this. So now it will filter out noise outside the band. It will capture the signal, omega c phi with frequency omega C phi. So that means you need to adapt, the filter needs to adapt itself. So this adaptation mechanism when you impart to a filter then this together is called adaptive filter.

So adaptive filter, filter coefficients are not fixed whether they are flexible by an internal algorithm of adaptation or adjustment. They are continuously adjusted from clock to clock to clock so that depending on the circumstances or surroundings and all that you know, it always goes to the base filter. Okay this is called adaptive filter, alright.

(Refer Slide Time: 16:59)



I will show you the typical framework of an adaptive filter. What happens is this x n is a random, I am taking them to a real valued but they are all valued for or valid for complex case also, for all purposes is enough if it a real valued. Random signal, one is given is e of x n that is expected value at any index n, it is 0 okay and another is if I define a vector, first let me come to this figure, x n is here and I am filtering it through a filter, FIR filter.

Filter coefficients are like this w 0 to w n minus 1 capital n coefficients 0 1 up to n minus 1, output is y n. Output you can write as w 0 times x n. Convolution sum w 1times x n minus 1 plus dot-dot-dot last one is w n minus 1 x n minus n plus 1 okay. Alternatively you can write in a matrix vector form. Suppose I define a vector x n, starting coefficient x of n x, x of n, next is x of n minus 1, next is x of n minus 2 dot-dot-do last is x okay this is your x n vector and filter coefficient vector w, I put w 0, then w 1, then this entire y n can be written as w transpose x n, in a very compact manner.

Why W transpose? Because if you transpose w, it becomes a rho vector and then first guy w 0 times x n, w 1 times x n minus 1 dot-dot-dot so that is what it is. So y n is w transpose x n. What is given? X of n is a real-valued random signals. All samples are random, every sample has 0 mean and when it comes to correlation, there is expected value of this column vector into its transpose, this is transpose. So column vector into rho vector.

So x n into x n that gets square n, x n into x n minus 1, xn into xn minus 2 dot-do- dot. That is called correlation, autocorrelation matrix. That is given to be r, so they are function of n but r is independent of n that is because we assumed stationarity that is input is stationary process. In fact, it is called wide stain stationary. Let me tell you though we are going little outside relation signal processing but then this knowledge might be helpful, WSS.

Typically, what is r? What is i, jth element here? I comma j th element will be x n minus i. May be you can take x n minus i from here and then I mean you make a row, take x n minus j, so minus i into x n minus j that will come in the i comma j th position. So r i j will be nothing but x n minus i, x n minus j but though n is present, this is function of I and j only. This is the assumption of stationarity that is the 2 samples, I mean, their 2 correlations depends only on the gap between them. That is how close they are separated or how far they are separated.

If they are far correlation normally is less, if they are very close correlation is high. So gap between them will be, this index is n minus i, another is n minus j. So if you subtract one from the other, say, n minus j if you subtract from n minus i. N cancels with n that is why n disappears. So it becomes a function of j minus i. So basically it functions of i and j that is why r i j.

Okay, so under this assumption r is not a correlation matrix independent of n and mean also 0 for all sample. So mean also does not depend on n because of whatever be the index n mean is 0. This is what is given you have the process it through some filter coefficients, FIR filter so that output you can write this way. How to derive the coefficients? This is our task. This should be a very good estimate of some another signal d n which is sometimes given to you.

During that time we call it training time, adaptation time, adjustment time. During that using the given d n you adapt it, the coefficients to become a better filter or you know whatever be the best filter required currently under the current circumstances, I mean adjust them, adapt them by some formula clock by clock by clock iteratively during that window. That time d n will be given and then d n is not given afterwards.

That time whatever filter you design, you came with, be happy with that and use it for a while. Again a training period will come and like that. So d n is called the desired response it is also a random signal. This also WSS (sorry), during training period what I want? I want to have the filter coefficients, the best that is y n should be a very good estimate of d n. That means if I take the error between them, difference, this is plus, this is minus and this is the error e n.

So e n, the power, e n is a random signal because if x n is random, y n is random, d n is random. So e n is random, so just square off e n power is not enough because I have to see statistically. So it is a random thing, you know, maybe at some index at a particular observation time power will be less for that observation. Next time when you observe power will be high. So that is why instead of taking the e square n we measure its average power, expected power and this we minimize.

Now this is what? D n minus y n. What is e n and y n is this okay y n is this. If you substitute y n as this d n minus w 0 x n minus w 1 x n minus 1 dot-dot-dot and square up and then apply e, expected value. This entire thing will be a function of what? W 0 square, w 1 square, w 2 square, w a minus square and w 0 w 1, w 1, w 2, w 0 w 2. So it will be a second order function of all the coefficients and I have to minimize this second order coefficient with respect to this coefficient.

So that for whatever coefficient, I mean, this is the minima that is the base. So I have to take partial derivative of this entire thing with respect to w 0 equate to 0, with respect to w 1 equate to

0, with respect to w 2 equate to 0 and second order when derived will become first order. So you will get, as you see, will get a set of equations, first order equations which will give an unique solution. That solution will give rise to the best w 0 w 1 all that. That will be called the optimum filter.

Before that there is one more thing on d n. D n also is assumed to be jointly stationary with x n means if you take correlation between d n and any sample of x n that will not depend on n. That will only depend on gap between d n, there is n index n and the index whatever is the index you have okay. Therefore another vector that we consider here is p which is the e of this x n d n okay x n d n.

There is d n if you multiply this x n vector. See all vectors are denoted with underscore. I should have mentioned this. All numbers, all variables with the underscore that indicates vector when if it is a lower case with underscore it is a vector, column vector. If you take a rho vector that means it is a transpose of a column vector. So we follow a notation, every column vector is denoted by a lowercase letter with the underscore. Rho vector means first draw the column vector then put a transpose on it so that it becomes rho.

For a matrix, again we use the uppercase letter but with underscore. So we use the underscore. If it is lower case, it is a vector only. If it is a uppercase letter it is the matrix only but if there is no underscore, it does not matter whether it is lowercase or uppercase, it will be a scalar. So p is e x n underscores, so this x n vector into d n. D n is a scalar no underscore. So x n vector into d n means this into d n, this into d n, this into d n, this into d n. All the samples of x n into d n. So another vector and e on them.

As I told you if it is jointly stationary, say x n into d n, that will not depend on n. Index n here, n here so gap is 0. So it will be just some constant then it means d n and x n minus 1okay then gap is 1 and then if it is d n and x n minus 2 gap is 2 so on and so forth. Okay that is why it is independent of n and that is what is given to you. Okay so we will start, we will cover this in the next half. Thank you.