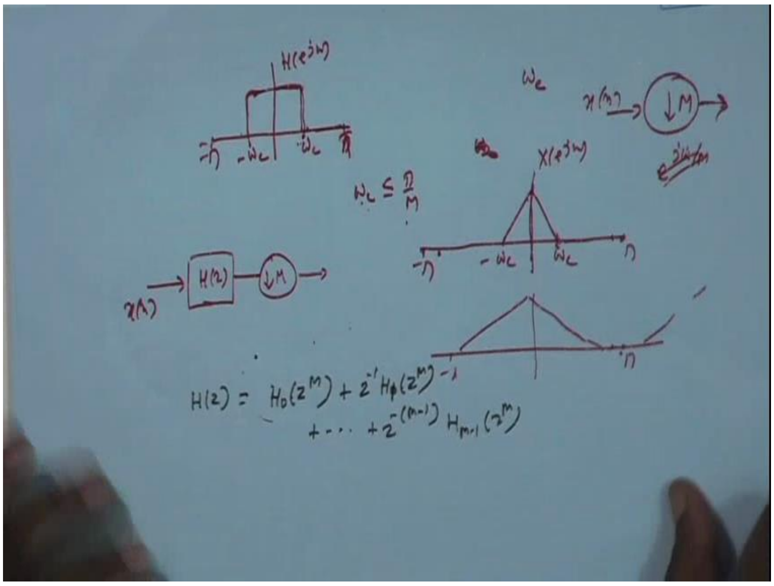
Discrete Time Signal Processing
Prof. Mrityunjoy Chakraborty
Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 26
Efficient Decimator and Interpolator Structure

(Refer slide Time: 00:25)



We have seen decimation interpolation. Often when you decimate a signal, you pass through a low pass filter. Low pass filter, if it is pi, if it is minus pi, it is omega c say minus omega c ideal low pass filter. You make sure that omega c is sufficiently less, so that after you do down conversion that decimate by m, no aliasing takes place, because signals will be perfectly band limited. If you want to decimate by M then omega c this cut off frequency you know it should be less than M types because it will be expanded M times. So, anyway you are passing through low pass filter, so that it is band limited.

Whenever you are doing say I am telling you, you are doing decimation. Before doing decimation that is your bring down the sampling rate you must make sure that the signal is properly band limited that is if it is pi, and you are decimating by a factor M then omega c should be less than equal to pi by M. After decimation this it should go this can go maximum up to pi and this can go maximum up to minus pi, still there will be no overlap from the right hand side, left hand side.

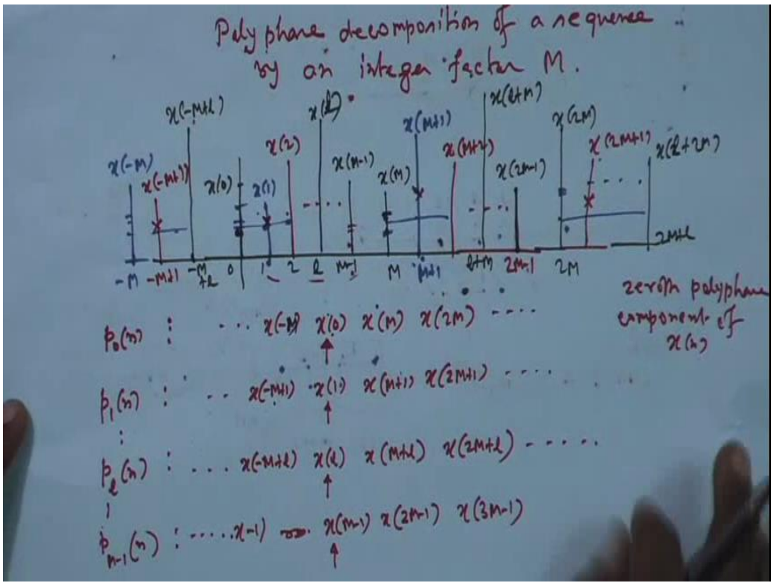
If you band limit the signal by low pass filter with cut off frequency omega c, because then suppose this original signal x e to the power J omega it is now band limited, suppose it is like this from omega c to minus omega c, and this is pi, and this is minus pi. These if you decimate you call it x n, if you decimate it output d0074ft you we will know the formula and there you see this omega c will be expanded it will go to I mean if it is exactly equal to pi by M, it will got pi otherwise it will go up to pi by M it will not touch up to pi when we go near pi minus pi. Then on the right hand side also you will have something like this, but there will be no aliasing all right, there will be no aliasing all right. Because after down conversion the new sampling rate and there were new half sampling rate will map to pi, but that will be M times small. So, omega c now it is M times less than pi. So, it will not cos pi this we have covered.

So, this low pass filter is used actually, always the low pass filter is used before any down conversion so that it is limited within a band, as I told you cut off frequency should be omega c should be less equal to pi by M this you have seen. If omega c is less than equal to pi by M, I am repeating input band will be only within this omega c say less than equal to pi by M. After decimation the new sampling rate will be M times less than the original new half sampling rate will be M time less than the original, but new half sampling rate will map to pi. Therefore, signal this omega c all frequencies you know will be expanded. So, it will go here, but it can go maximum up to pi, because omega c is less than equal to pi by M all right.

You remember that formula decimation after decimation d t f t formula, this factors come e to the power j omega by M as a result it expands. This figure was done in class that is why I am not going deep into it am just dropping a few hints. What happens it expands by factor capital M, but if it is will it cross pi answer is no, because it was originally less than equal to pi by M. Now, new half sampling frequency is M times less that will map to pi and this will expand by M times whether it is still not cos pi because of this relation which means this will be like this within this band and here again then no aliasing. Because of this we have a low pass filter always we put down conversion.

Now, I will talk of using this noble identities which was done in previous class, I will talk of efficient decimator, efficient interpolator. There is an H Z filter is given. Now, look at this structure here the input clock rate is higher, the filter is working at higher rate, then I am bringing down the sampling rate. So, this side is operating at higher sampling rate, this side at a lower sampling rate that means, this side costly the filter is working at a faster clock anything that is higher speed anything that will faster clock that will be costly. How to make it less costly, how to make operate at lesser clock; for that I reimbursed this polyphase decomposition of H Z; since, it is capital M, I will look for a factor of M polyphase decomposition of H Z that will be H 0 Z to the power of M. I hope you remember this very quickly what are those polyphase factors.

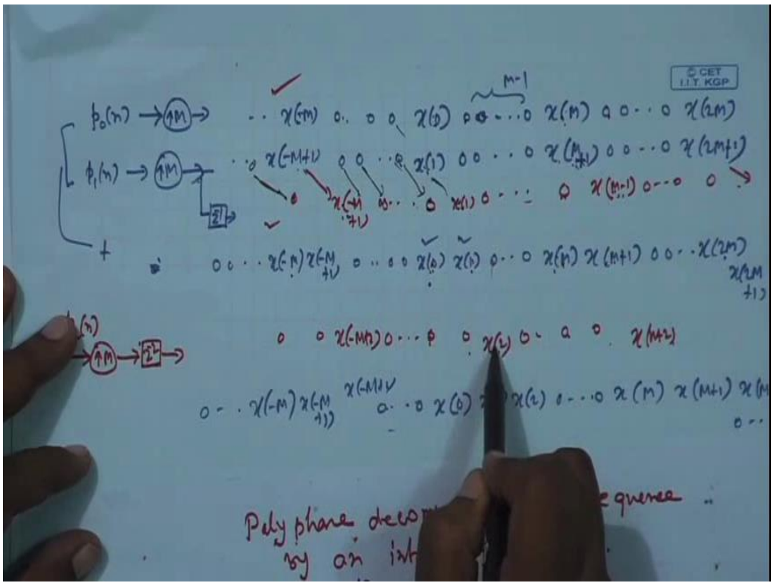
(Refer slide Time: 06:05)



Suppose instead of H or I put the sequence x. So, this is the sequence you take x 0, then Mth, 2 Mth. So, from this x 0, x m, x 2 this is the 0th sample, first sample, second sample minus 1 sample like this x minus M you call this sequence P 0 n. Then take the first sample this arrow means 0 origin. So, now, x 1 is at the origin. You take first sample M plus 1 sample 2 M plus 1 sample minus M plus 1 sample. Let them side by side. So 0th sample is x 1, first sample is x n plus 1 dot, dot, dot, you call P 1 n.

In general P l n is rooting the 1th sample x l then you take l plus M l plus 2 M like that you know x l at the 0th position origin x l plus M l plus 2 M x l minus M like that. And therefore, you can go up to P 0, P 1 up to P M minus 1; in P M minus 1 take x M minus 1 then 2 M minus 1, 3 M minus 1 like that you can have only this. If you go for P M n you will get back P 0 M, we do not need that P 0 M same as shift. Now, from this you could get back to the original sequence, how.

(Refer slide Time: 07:27)

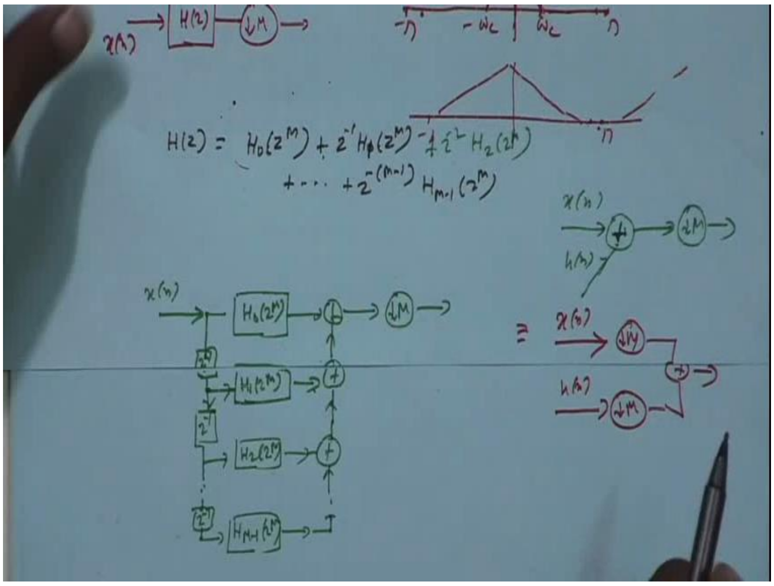


If I expand P 0 n by expand between these two M minus 1 zeros between these two M minus 1 zeros between these two M minus 1 zeros like that. Then P 1 n will also expand between these two M minus 1 0, M minus 1 0 then there is this then this sequence I shift delay by 1 at the faster clock. So, this will come below this, this will come here, this will come here x 1 like that. If you add the 2 now, x 0 and this 0, this 0 will come here, sorry this 0 will come here, this 0 will come here, and this will come here. So, x 0 plus 0 - x 0, x 1 plus 0 - x 1, this plus 0 this and like that. So, you will get something like this x minus M 0 is coming here.

So, this plus this x minus M, this is coming here this plus this x minus M plus 1 0 is coming here 0 plus x 0 x 0, x 1 is coming here x 1 plus x 1 like that dot dot. So, here x 0 x one has come with zeros then x M m plus 1 come zeros then you take the next one and shift I mean expand by 2 and shift by 2. So, in that case, x two sample will come here x M plus 2 will come here like that if you add you will get x 0, x 1, x 2 like that this way the whole sequence will build up. So, in terms of z transform, if it is P 0 n capital P 0 z it is P 0 z to the power of M because expanded it is p 1 capital P 1 z. So, capital P 1 z to the power M into z inverse so and so; P 2 capital P 2 z the capital P 2 z to the power M into z inverse 2. So, overall z transform will be summation of this that is polyphase decomposition.

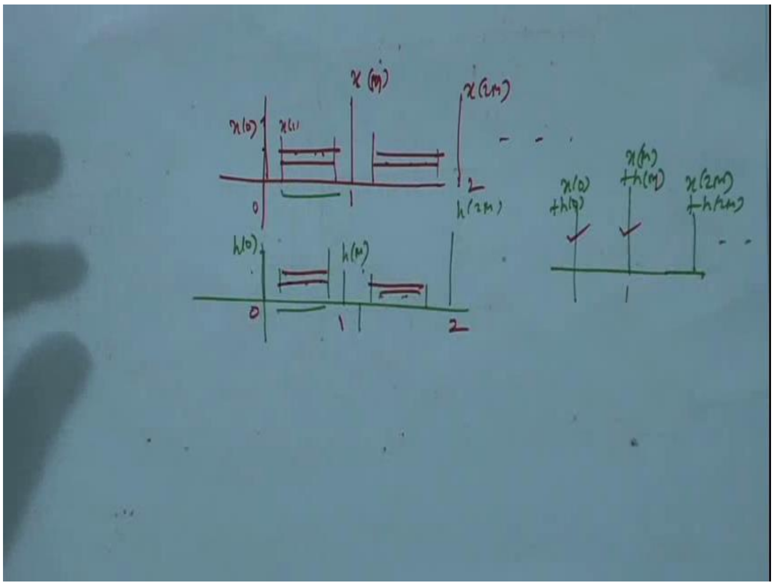
H 0 expanded so H 0 z to the power M then h one expanded sorry this is h one h one expanded and then z inverse is the one shift. Next one will be h 2 z to the power M because expanded then to z to the power minus two dot, dot, dot. This was your polyphase decomposition. So, H Z, I go for the polyphase decomposition, so x z into H Z means H Z into these plus x z into this plus like that.

(Refer slide Time: 09:48)



So, what can you do, this filter can be realised like this x n that is in terms of z transform x z, x z into this component; x z into this then x z into H 1 z to the power M with z inverse. So, put z inverse that into h one z to the power M. Then another z inverse because next will be z inverse 2 H 2, so another z inverse. So, total 2 z inverse then again H 2 dot, dot, dot, dot. Lastly and you have to add all of them because of this plus sign. So, x z into this coming here x into z this coming here H Z into next coming here all of them to be added. So, go on adding (Refer Time: 10:59) something which takes two input one output that is why I have to do so many additions, this is the output and then you are decimating right. Now, you see if suppose two signals x n, suppose two signal h n they are first added and decimated by M, you will get the same thing this is equivalent to first decimate these and then add this very simple.

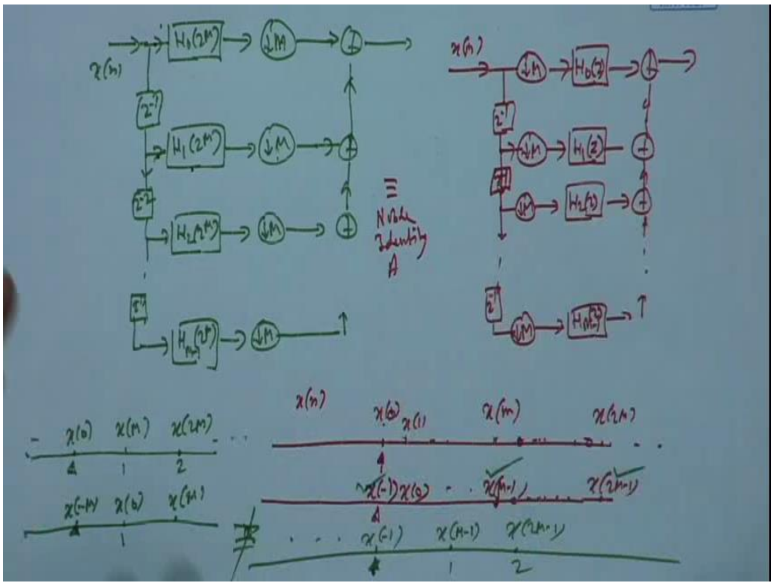
(Refer slide Time: 11:51)



Suppose, you have got x 0, then x 1, dot, dot, dot x M here then again, dot, dot, dot, dot x 2 again then, dot, dot, dot, again I have got h, dot, dot, dot there is h, again, dot, dot, dot h 2 M. Suppose, you first add and then decimate. You will have x 0 plus these then next M minus 1 samples you throw away then take x M plus h m. So, you will have x 0 plus h 0 then first sample will be x M plus h M because you are throwing at the intermediate samples then will be x 2 M plus h 2 M, dot, dot, dot. See, if add and then decimate by M this is what you will get.

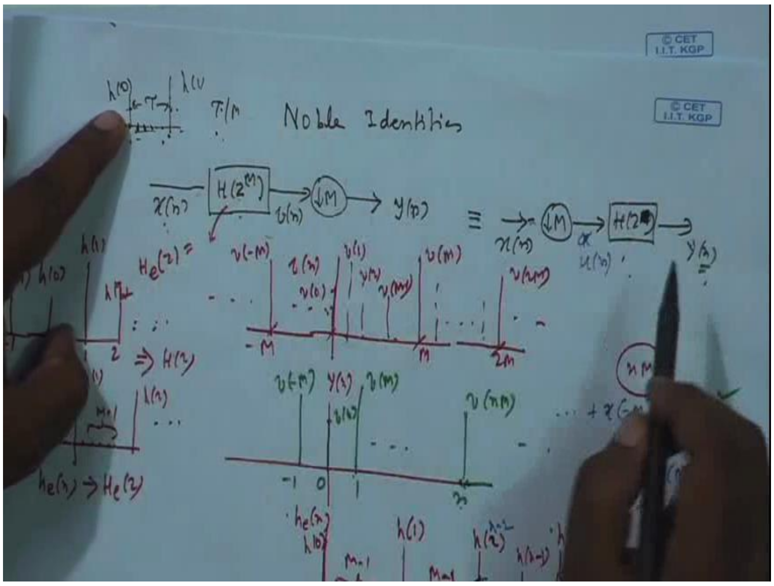
On the other hand, if you decimate first and then add you will get the same thing because if you decimate first you are throwing like these. So, this will become sample number one, this will become sample number two, this will become sample number 1, 2, 0, 0. Now, if you add x 0 plus h 0 present here. Then sample number one will be x M plus h M present here. The sample number two will be x 2 M plus h 2 M present here. So, either you add or throw away intermediate samples, you get the same thing or you first throw the intermediate samples and add you will get the same. If that be then these two are getting added then decimated I can bring the decimator here and here; again this two getting added decimated. So, this decimated I can bring here and the here. Again these two are getting added then decimate. So, this decimator I can bring here and then here like that. So, I can bring eventually decimator will be here, here, here, here, so that means, the new structure will be.

(Refer slide Time: 14:13)



New structure will be, this is the new structure decimator moved here, here, here by (Refer Time: 15:14). Now you remember noble identity one. What was noble identity one.

(Refer slide Time: 15:26)



These are noble identity one. If you have a x n processed filter of the from H Z to power M that is already expanded by M 0s M minus 1 0s are there between every pair of samples originally H Z then between h 0 h 1 put n minus 1 samples, again n minus 1 again M sample then the real term transfer function is called H Z to the power M it will be you have seen. And then if you decimate by M you will get the same thing if you bring the decimator first and then the filter where H Z to the power M is replaced by h z. So, between h 0 and h one no zeros this is 0th this is fast between h 1 and h 2 no 0s this first second like that.

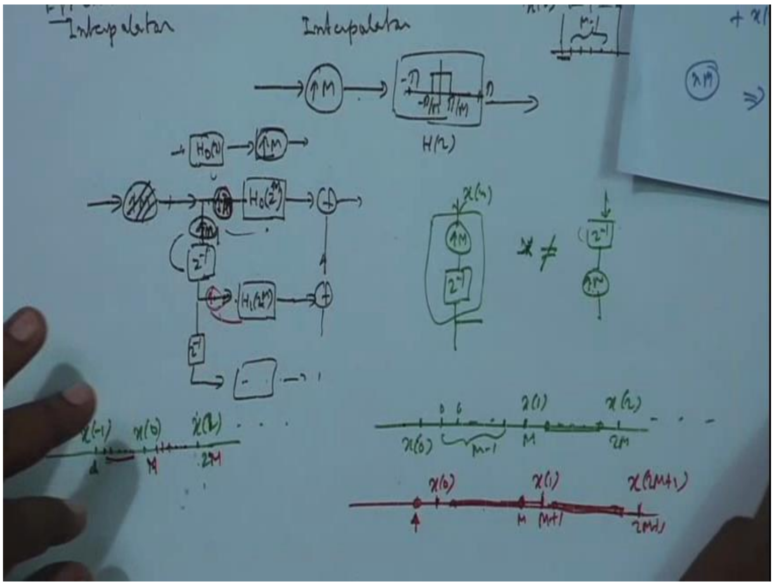
So, this I can apply between these two this decimator can come in the front it will be h 0 z then between this two decimator come in the front it will become h 0, h 1 z, dot, dot, dot, dot. So, this will be equivalent to this. Now, j because of the noble identity noble by noble identity A, it will be this structure what is the advantage of this structure this filters are working at a lower clock rate sampling rate because sampling rate has first been brought down m. So, now, the clock is lower n times lower filter is working at a slower late. So, it is less costly whereas previously first you had a filter which was working at higher sampling rate and therefore, more costly then you down sample it.

Only one thing you have to consider sorry there will be a z inverse here. Only one thing you cannot push this behind because of the simple digit that if you have one sequence say the delay and this cannot be interchanged. Suppose, you got a sequence x n it is like this, these are origin x 0 then x 1, dot, dot, dot x M. And again you know all these points x 2 M. If you decimate if you delay it, first then z inverse if you delay it first and then decimate; if you delay it x 0 will go here x minus 1 will come here, x M minus 1 will come here, this is the Mth point. So, x and you have got all other data I am not naming them.

If you now decimate what you will get is this guy, then this guy this guy like that. So, you will have x minus 1 then x this number one point number two point will be this x 2 M minus 1 like that, dot, dot, dot. There is if you delay fast and then decimate you get this, but on other hand if you first decimate if you first decimate it you will get x 0 the number one point will be x M, and the number two point will be x 2 M, dot, dot, dot, if we fast decimate and then delay. Now, you delay here x minus M will come from the left side at one position x 0 will come, two position x M will come, and you see this not same as this is not same as this. Here at 0 x minus 1 here at 0 minus n here at number one M minus 1 here is x 0.

So, they are not same actually this is not a linear this not a shift in variable this is linear, but not a shift invariant structure that is why the two cannot be interchanged. The two can be interchanged two system both are linear shift in variant then overall transfer function is H Z which is h 1 z into h 2 z which you can also write as h 2 z h 1 z. So, two can be interchanged. But not otherwise, if you have got a system h 1 z transfer function the system is linear shift in variant. You have got another system in cascade transfer function h 2 z this also linearized shift in variant overall transfer function is h 1 z into h 2 z which is same as h 2 z into h 1 z. So, you can bring the block h 2 z in the front h one z in to the right still thing will be same. So, they can be interchanged, but decimated is not shift in various structure you can verify and that is why your decimator and delay cannot be interchanged, but anyway this is an efficient decimate structure we have seen filters are working at a lower rate.

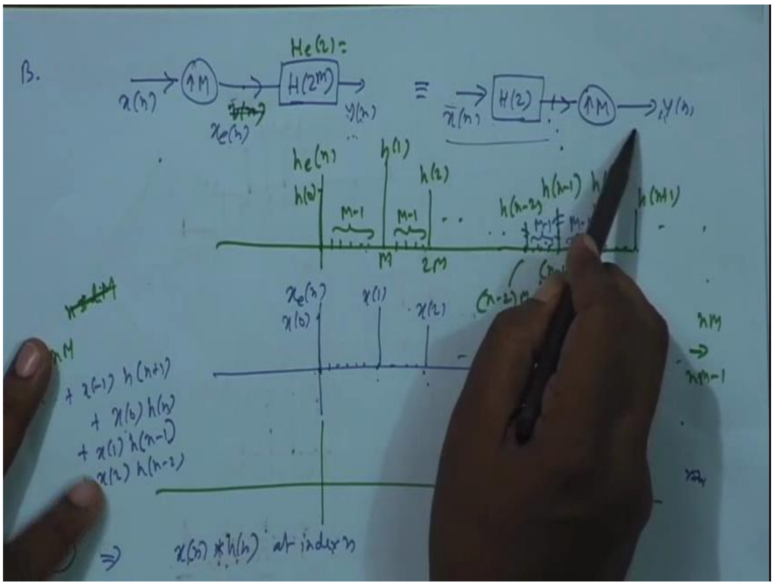
(Refer slide Time: 21:14)



Now, we will go for the other (Refer Time: 21:09) that is expand r. What do you do in the case of expand r interpolation efficient what we do in interpolation we first pass it through an expander and then you have a filter H z, where transfer function is like this. Ideally like this pi by M minus pi by M and then zeros up to pi and minus pi there are transformer frequent response. You first expand by M and then pass it through this filter if you have forgot go through my notes on interpolation there are interpolation interpolator. We discuss this at length interpolator first we expand between every two samples put in M minus 1 zeros say expand it then pass it through a filter of this kind ideal low pass filter if possible from minus pi by M to pi by M that is pass band otherwise 0. If pass it through it output will be the interpolated thing. So, suppose the transfer function is H Z.

Now here you see first sequence was loaded sequence, some sampling clockwise for that now this is after expansion it is becoming higher faster sequence because between every pair of samples I am bringing M minus 1 zeros that is am creating between say x 0 and x 1. Earlier so much was the sampling period and sampling rate was small. Now I am bringing in intermediate sampling points total M minus 1 zeros. So, now, sampling period is tau, tau is total is t tau is t by M. So, sampling period is going down by sampling clock is M times faster. So, filter is working at faster clock rate M times faster clock means it is costly. How to bring down the cost how to make it (Refer Time: 23:05) lower rate well first you should see I can still bring this replacing by that from, that polyphase decomposition as we did last time when will see it will not work now here. After drawing the remaining part because I will not hold on to this, this will not work will see, we know the noble identity of that kind two.

(Refer slide Time: 24:52)



Give me a minute. (Refer Time: 24:51) we have type B. Here if we first expand x n, so from low rate it become high rate and there you pass it through the filter of the kind H Z to the power M. So, here again in the transfer function between every pair of sample there is M minus 1 zeros expanded version of impulse response. So, that is equivalent to first passing x n through H z, but there is no such zeros between every pair of samples.

So, filter is working at same rate as the same sampling clock as the original input. So, it is less costly and then you expand by M you will get same thing.

So, if you apply that here this decimator interpolator, this expander I can bring here I can bring here then instead of here there is first to expand between every pair of sample you plug in see zeros and give here and here instead you can do here also you can do here. Same sequence I am repeating the thing here and here, but again this two cannot be interchanged I cannot bring it here. Here I see I can apply the noble identity b that is h 0 z to the power M and M. So, this will become this particular part will become H 0 z first, it will come here followed by expander this part; this part, this part will become like that. So, this will be working at a slower clock.

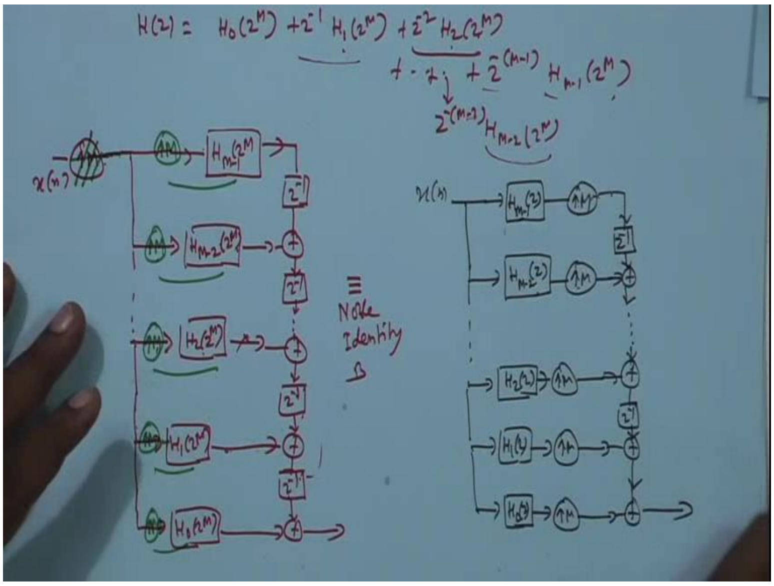
But these interpolator expander cannot be pushed here, that is if you have expander this is actually that part I am redrawing because it is little unclear here so. So, this part, this part is not equivalent to first delay and then expansion. They cannot be interchanged very simply why because suppose I have got a sequence is coming here say x z. If I expand it, I will have x zero then zero zero, dot, dot, dot zeros M minus 1 then x 1 which will be Mth sample then 0 0 0 then 2 M sample will be x 2, dot, dot, dot. Then if I delay it by 1, I will have a 0 coming from the left 0; x 0 will move here then these zeros at M th point also I will have zeros. Then at M plus one th point I will have x one then again 0 zero 0 at twoth point also 0 then at 2 M plus 1, I will have x two M plus 1, dot, dot, dot.

On the other hand, here if I first delay and then expand what will happen if I first delay this? So, originally x n is first delaying it. So, you will have x minus 1 coming at origin than x 0 sample number one then x 2 sorry x 1 sample number two, dot, dot, dot there is delayed. Now, if you expand, you will have 0 0 0, this will become Mth; then 0 0 0 this will become 2 Mth. So, you see zeroth here is x minus 1, here zeroth is 0, Mth here is x 0. Mth here is 0, so that two are not same that is because this is not shift in variant structure linear, but not shift in variant structure that is two cannot be interchanged. Because (Refer Time: 28:42) from here.

If you first delay the input sequence and then decimate delay means x 0 will move to position 1, x 2 will move to position 2, x minus to 1, 0 and then you are plugging in M minus 1 0. So, this becomes Mth point this become 2th point. On the other hand, if you first expand and decimate the origin you have got 0 x 0 came to here; Mth location till at point I am got 0 band then M plus 1th, I have got x 1, then again zero band two M plus 1th, I have x 2 M plus 1 like that. So, these are same you can see that is why I cannot move it here to take advantage of that noble identity.

So, this way I will not work, I could only do that here this is M and this I could interchange. If h zero z to the power M h zero z comes first and this expander that go second this can be interchanged here because there is no delay. So, these two I can apply this noble identity b here, but I cannot bring that interpolator here and then apply noble identity b on these two and same downwards also. So, this will not work, but I can get around by this (Refer Time: 29:52) by clear manipulation of this kind.

(Refer slide Time: 29:59)



That H z, I know is you start with this first that is top filter instead of h 0 z inverse this then we write this, then let there be one delay. Then next filter one delay, dot, dot, dot, this is the situation. You see x n there is x z into this, x z into this that signal passes through one delay z inverse. If you forget about these contributors, one z inverse, then again z inverse again z inverse, total there will z to the power minus M minus 1. Then next signal, if you want you can write the signal here that is z to the power minus M minus 2. So, x z into these means x z into this, this part first, x into this part first then that will go through M minus two delays, you forget about this top delay, it is going through one delay, another delay, another delay, another delay total M minus 2 delays.

Consider these x z into h 2 z to the power x z into z to the power M here it should pass through two delays, so one delay, another delay. Then consider this x z into h one z to the power M one z to the power h 1 z to the power M then that will pass through only one delay and x z into h 0 z to the power M no delay; and all of them get added in this process this is called transpose structure.

Here if I have these I can happily copy it in every place, because expanded thing going here, going here, going here same thing as I mean you can write like this. Instead of having it here that was originally I can have the expander here, I can now have the expander here, I can now have the expander here, that is expanded and then expanded thing will go into all. So, I am making the expansion right at the beginning of filter that is at the input of filter same x n expanded going here expanded same as before, earlier it was expanded and going in. So, x n expanded going in, x n expanded going in, x n expanded going in, so I am not making any change. But between these two between these two between these two between these two I can apply the noble identity B and this will become this nice structure.

This is the structure here. Here you see this filters all working at the same sampling clock rate clock at the input. So, and then only they are expanded extra N minus 1, sampling points created and there zeros are put that are expanded meaning of expansion in all of them and they are getting added. But before that the filters, this polyphase filters they are working at the lower clock because that is the sampling clock involved in the x n present with x n that is the original clock rate, and that is lower and after expansion that is higher. So, previously if they are working after expansion, so this filter were working at a faster clock. Now, they are working at slower clock, so that is why this is efficient. And by the same talking, you cannot push this interpolator here end of the delay because as I told you this expander ahead of the delay because delay an expander cannot be interchanged, because this is linear, but not shift in variant, we have seen it already. So, that is all for this class, we move to the next one.

Thank you.