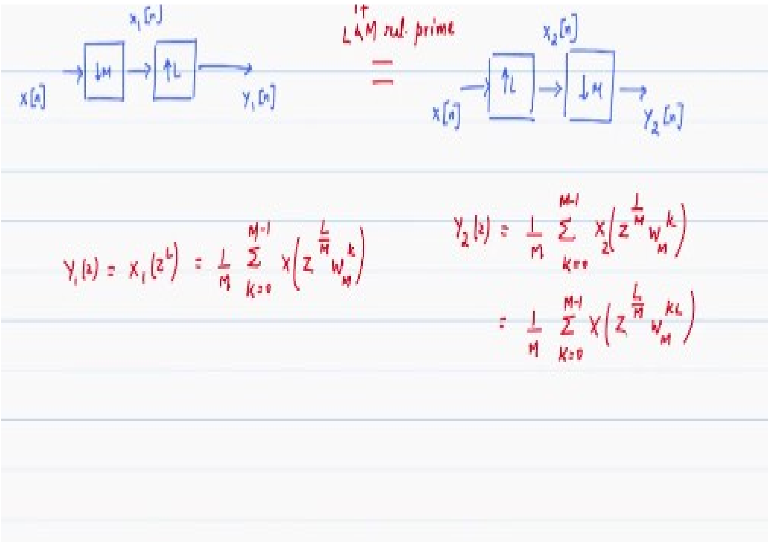
Multirate Digital Signal Processing
Prof. David Koilpillai
Department of Electrical Engineering
Indian Institute of Technology – Madras

Lecture – 13 (Part-2)
Noble Identities and Polyphase Decomposition – Part 2

(Refer Slide Time: 00:18)

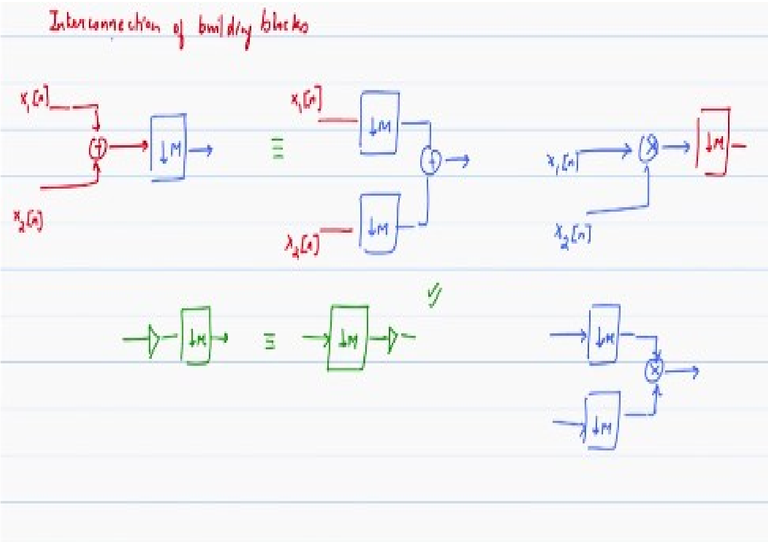


Now the second result from yesterday's lecture was that when I have a cascading of 2 blocks, down sampling and up sampling where M is not equal to L. So basically we will make it clear that L is not equal to M, but we have not said anything about common factors. In the general case the 2 results are not the same. We derived the expressions, again I wold not rewrite them just except just highlight saying that in one case the expression has WM K.

In the other case it has WM KL and based on the properties of the roots of unity, complex roots of unity you can show that if L and M are relatively prime these 2 are spanning the same set of roots which means all the distinct roots of unity will be spanned by both of these and therefore they are equal. Now if L and M are not relatively prime. So basically for example if they were 3 and 6, or there are some common factors between them.

Then the up sampling and the down sampling will give you, this operation they give, they basically give different results. So therefore this result does not hold that the equivalence only holds when L and M are relatively prime otherwise you will get different signals on both sides okay.

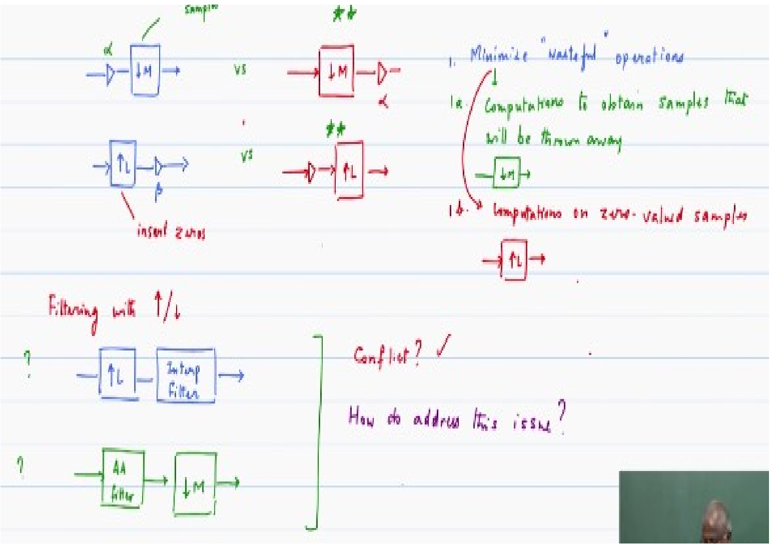
(Refer Slide Time: 01:57)



Now another element that we discussed in the last class was the interconnection of conventional building blocks from DSP, adders, scale factors, multipliers with the multirate blocks and we said that the multirate blocks are linear, so therefore you can do the addition before, after the adder block or before the adder block does not matter, similarly the scaling, similarly the multiplication. These are elements that you can that we have discussed.

I would like to build on that and make an observation and some comments on this because we are starting to now really look at the strength of the signal that we are, of the methods that we are working with. So basically what I would like you to start thinking about is the following okay, because it leads us to a very important way of looking at multirate signal processing.

(Refer Slide Time: 02:50)



So I have multiplier and then I have a down sampler, I have a down sampler. So basically one of the essential things that we keep in mind always is that the down sampler throws away samples, throws away samples. So in this case this is not a very efficient implementation because I multiply all the input samples by the scale factor alpha for example and then throw away M-1 of those.

So this versus the more efficient implementation would be use the properties of linearity and make sure that you do the down sampling followed by the multiplier alpha this would be the preferred approach. So basically I will put a star on top of this. That means that is the preferred approach of the 2.

On the other hand, similar observation regarding the other block where if I did up sampling by a factor of L followed by a scale factor beta versus the case where of course I could move the blocks around, I do the scale factor by beta, up sampling by L and this would be the sequence of operations. Now in the first case there are lots of zero valued samples, I am multiplying by beta, it is not going to make any difference.

I am just going to, the multiplication will get done, but result will be. So this is a case where this block actually inserts zeros, insert zeros, insert zero. So basically any operation after that we will be doing a lot of competitions on zero valued samples. So this would be the preferred one where you scale only the input signals and then insert the zeros. So again this would be the preferred configuration because it is more efficient okay.

So the two, in one case we say that do the decimation as early as possible because that means down conversion and do all the processing as much as it after you do the decimation. Same way up sampling is going to insert zeros, so do your processing as much as possible before you insert the zeros and then minimize the processing after we do. So you can summarize this as in using the following sort of guidelines, rules of thumb, the first one would be minimized wasteful operations okay.

That is like do not compute and throw away or do not multiply or do not do operations on zeros and this will be one of the ways in which multirate signal processing and the tools that we have are going to play an important role. One is; first of all minimize and how would we minimize? Let me call it as 1a and 1b, there is a different strategy when it comes to the down sampler, basically what we say is avoid, minimize computations that are going to be on samples that are going to be thrown away.

So this minimization we would like to apply to computations to obtain samples, computations to obtain samples that will be thrown away and keep that to a minimum because that is going to be, that will make the overall system more efficient. Samples that will be thrown away and this actually turns out to be one of the foundations on which we do polyphase decomposition and the noble identities which we will be looking at in today's lecture, that will be thrown away.

So that is for a down sampler, will be thrown away and that is primarily linked to the down sampling operation okay. Then 1b, will be avoid computations, minimize computations on 0 valued samples, computations on 0 valued samples and this basically is another way of saying what we had already mentioned; for a down sampler, try to do the down sampling as early as possible and therefore and then the processing for the up sampler do the processing first and then the up sampling process, so that way you will satisfy 1a and 1b and that will give us an overall efficient implementation okay. Now having understood this and very happy that yes multirate gives us some significant tools to work with, we go back and say what do I do.

When do we have computations that go along with the up sampling and the down sampling most of it is when we do sampling rate conversion, it is the filtering that goes along with it. So let us look at filtering with up sampling and down sampling okay, with up or down sampling okay. So the first case that we have looked at is; when you did up sampling, you said, we made the observation that we would have to follow it up with an interpolation filter.

Now why do I do interpolation filter to remove images, when does the images come? Because I insert zeros. Now does it make sense to say do the filtering before the zero insertion? No. So you kind of say well you know I am stuck, though I realize that I should not be doing a lot of operations on zero valued samples, I may not have an option because this is the case where okay.

So this is one that is sort of is going to, I will put a question mark okay, I am not going to be able to achieve or utilize some of the observations in the previous one. The second case that we looked at was when I want to do down sampling in that case we said okay, precede it with an anti-aliasing filter okay and then followed by the downsampling, downsampling okay. So the most common or the simplest of operations that you would combine with up sampler and down sampler are the corresponding filters.

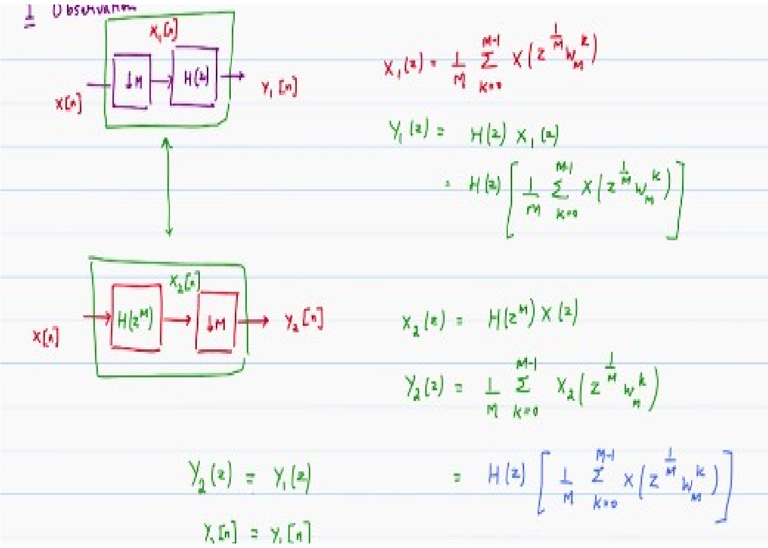
Of course there is lot more that we will do and we will study but at the foundational level, these are the 2 operations. Now this one also is conflicting with the requirement because what does it say if you are going to do down sampling do not do all the operations and throw away samples, but the problem is if you do the down sampling without doing the filtering then what will happen? Aliasing will come and then I will be in bigger trouble.

So again this says well there is a big question mark saying okay what do I do. There is probably nothing, because in both cases the filtering needs to happen where it is happening. In one case it is happening after the insertion of zeros, in the other cases before you throw away samples. So really how do we take advantage of that. So we see that there is a possibly a conflict with the observations that are above.

So is there a conflict? The answer is yes okay, yes there is a conflict, there is nothing I can do we said. So then how do we address this, how do we address it, and this is where the sort of the beauty of multi-rate signal processing comes in we will look at. How do you address this fundamental issue? Okay that is the question that we answer in today's lecture okay. So this is the part.

So how do you do the operation so that those are happening in the most efficient and efficient possible manner. Okay so let us quickly dive in and build the blocks that we need and develop that.

(Refer Slide Time: 10:44)



So first of the observations that we will want to come up with okay, this is an observation, observation number 1, I want to look at down sampling followed by filtering okay. Now let us assume that this is the particular scenario where I am not so much worried about the anti-aliasing part let us say there is some processing that is happening on the down sample signal. So this is a very important step and we will see in a moment, why it is so.

So if my input is X of N and my output is Y1 of N and intermediate signal is X1 of N. Now can you please help me write the expression for Y1 in terms of X of N. It is straightforward, let us first write down X1 of Z that is the down sample signal 1/M summation K = 0 through M – 1, X of Z power 1/M WM K that we are comfortable. Now Y is the X1 of N filtered through H of Z.

And so we can write down the output expression Y1 of Z = H of Z times X1 of Z this can be written as H of Z and within the bracket; 1/M summation K = 0 to M-1 X of Z power 1/M WM K, okay, I do not have a direct input output relationship because this is not an LTI system right. I have X of Z and I have shifted versions and of course there is frequency scaling and all of that is present in the picture.

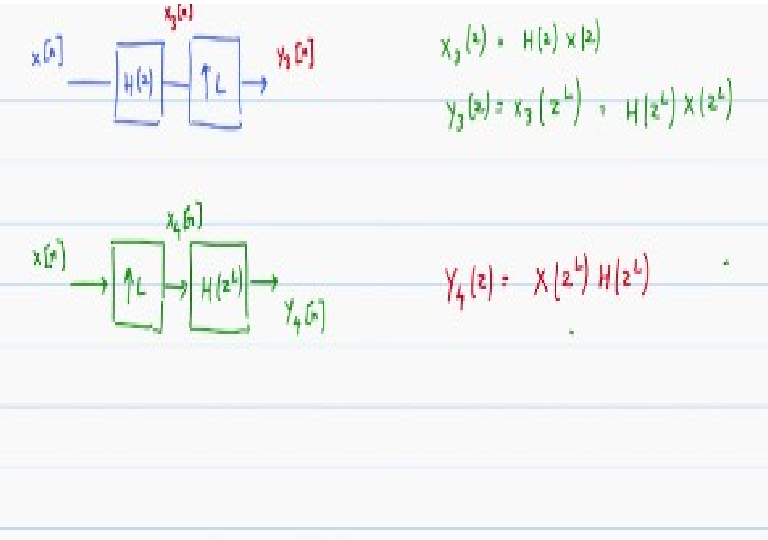
But nevertheless this equation is correct between Y1 of Z = H of Z times this expression. There is nothing wrong in this equation, this is correct. Now I would like you to apply your focus on to the following structure, another structure where I have the input X of N same input, but this time I am going to modify my filter H of Z power M. If I give you a H of Z which is a low-pass filter what will H of Z power M be?

It will be a multiple pass bands, bandpass filter, it passes low frequencies, but it also passes several bands in-between so and does it also pass high frequencies? It does, right, well we do not know depends on where these images will come, certain values of M you will get the high frequencies also, but interestingly you think about what H of Z power M looks like okay. So H of Z is, H of Z power M is what we have followed by a down sampler and this output I will label as Y2 of N with the intermediate signal being X2 of N, okay.

Please write down the output for this. The first step says I have X2 of Z that is H of Z power M times X of Z, Y2 of Z is the down sampler 1/M summation K = 0 to M -1, X2 of Z power 1/M WM K right. Now please substitute for X2 of Z from the earlier equation and please do verify that the result that we get is actually H of Z times 1/M summation K = 0 to M-1 X of Z power 1/M WM K.

Which means that we have Y2 of Z is actually equal to Y1 of Z which means Y2 of N is the same as Y1 of N, which basically means that these 2 blocks are one and the same. So this block is the same as this block. These 2 are the same, the green boxes are the same. So I can equate, I can actually draw downsampling by a factor of H. So let me just maybe do that.

(Refer Slide Time: 15:45)



Down sampling by a factor of M followed by H of Z is identically equal to filtering by H of Z power M, followed by down sampling by a factor of M. These 2 are identical okay. Now. very easy to see, the principles that we have said in the last lecture earlier slide which says that it is always advantageous to do the down sampling first followed by any operation. So which says that this of these 2, this will be the preferred structure okay.

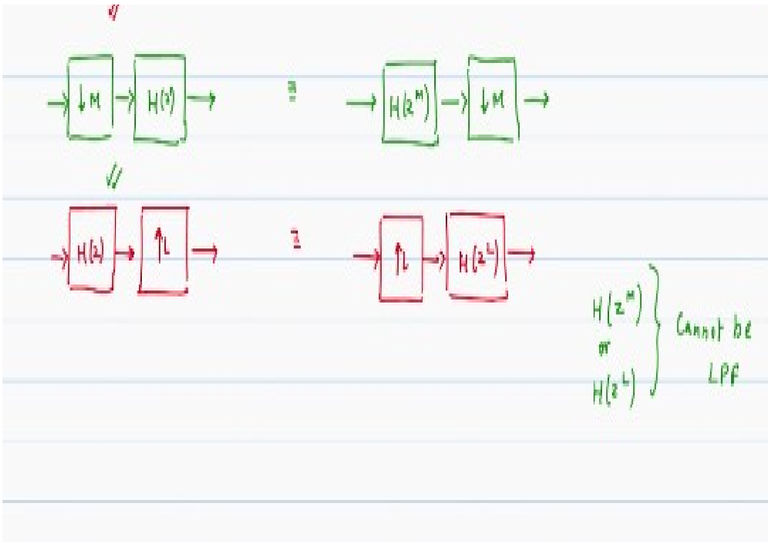
Now go back and look at our, whenever you want to do anti-aliasing filtering it comes out into this form right, it comes out into this form, you have to filter first and then down sample. So if my processing before downsampling can be written as a filter which is H of Z power M something which has the same exponent or power as the down sampling factor then I have a very efficient implementation.

Now go back and look at the counter part example. So I have input X of N, input X of N, I want to pass it through a filter H of Z followed by a up sampler by a factor of L. Let me call the intermediate signal as X3 of N and the output as Y3 of N. Please give me the expression for Y3 of Z. This is fairly straightforward X3 of Z will be H of Z times X of Z, that is the filtering operation.

Y3 of Z is X3 of Z raised to the power L wherever there is Z, I have to substitute, replace it with Z power L. If I use it in the previous equation this becomes H of Z power L X of Z power L okay, maybe it is already obvious from the expression, but we will just write it for completeness. It says that this structure if I do the following up sample by L and then multiply or do the processing with H of Z power L.

What will it be, this is X of N, X4 of N, Y4 of N, X4 of Z will be X of Z power L, output Y4 of Z will be, so let us write down the final answer Y4 of Z is nothing but X of Z power L H of Z power L. So here again we have shown 2 structures that are equivalent.

(Refer Slide Time: 19:04)



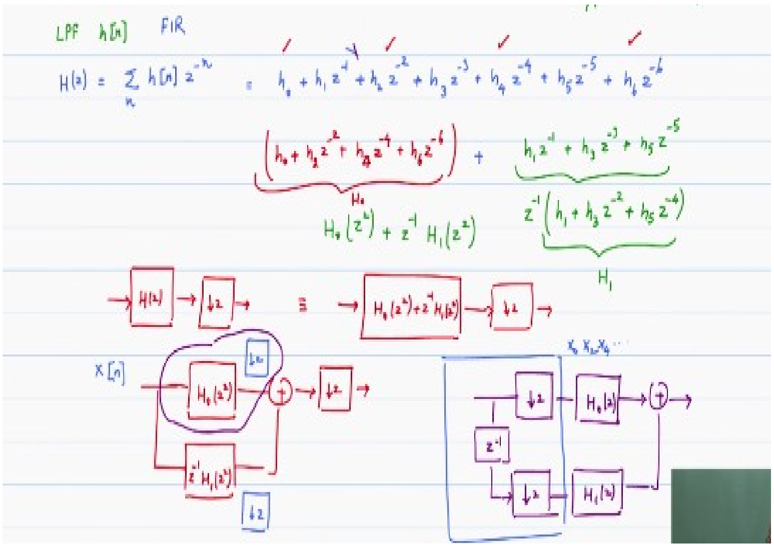
Let us capture that information in our result which says that if I have a filter H of Z followed by a up sampler, this is identically equal to an operation where I have up sampling by a factor of L followed by H of Z power L, okay. These are exactly equivalent and of course going back and looking at our basic premise, up sampling introduces a lot of zero valued samples better to do the processing before you do the up sampling.

So which says that the this is the better or the preferred of these 2 structures this would be the preferred one when you want to do interpolation I have to follow it by the interpolation filter. Unfortunately this is not an interpolation filter, this is H of Z power L which is not a low-pass filter. So maybe we can just make a mention of it which says H of Z power M or H of Z power L these cannot be low-pass filters, cannot be low-pass filter.

So you cannot interpret either of them as an anti-aliasing filter or as an interpolation filter, nevertheless if you can whatever processing that comes in a after up sampling if it can be clubbed in the form of H of Z power L, you have a very efficient implementation which is in this form. So the question was can a low-pass filter be written in this form? And I am glad you asked because maybe we will answer that question that was supposed to be after the next section, but that does not matter we will answer it, it is a very important question.

Let us answer that question and you will see that the beauty of multi-rate signal processing actually emerging from this very simple slide okay.

(Refer Slide Time: 21:02)



So if I were to look at a low-pass filter with impulse response H of N okay, it is low-pass filter. Just for argument sake let us assume that it is an FIR low-pass filter, finite impulse response low-pass filter. H of Z will be summation over N, H of N, Z power –N. So if you were to write this down what will it come out to be let us assume that it is a causal filter it will be H0 + H1 Z inverse, H2 Z-2.

I will write up to H6 just to illustrate, H3 z power -3 + H4 Z power -4, H5, Z power -5, H6 Z power -6 okay. Now if I were to group these even terms of the impulse response, I would get H0 + H2 Z power -2 + H4 Z power -4 + H6 Z power -6 + Z inverse. Let me just write it like this; H1 Z inverse, H3 Z-3 + H5 Z-5 which can be rewritten as Z inverse times H1 + H3 Z-2 + H5 Z-4 right.

So I can write this as H0 of Z squared, some polynomial with argument being Z squared, not Z anymore, Z squared. Because all of them are even powers + Z inverse H1 Z squared correct. So this is I am calling this as my polynomial H0 and this, without the Z inverse as my polynomial H1. Now here is the beauty of what we have been talking about so far. I want to do down conversion by a down sampling by a factor of 2 which means I must precede it with an anti-aliasing filter.

So I have H of Z, which is going to be followed by a downsampling by a factor of 2. This structure is not efficient because I am doing the processing and then throwing away samples; however, if I do this sort of splitting then I can say that this is identically equal to the same filter, but now written as H0 of Z squared + Z inverse H1 of Z squared, followed by a down sampling by a factor of 2 okay.

Please draw for me a structure that implements this, I separate out the blocks H0 of Z squared then another block which says Z inverse, H1 of Z squared and then you have to add the outputs together. So the input X of N gets filtered by H0 of Z squared that gives you one output and then the same input is filtered by Z inverse H1 of Z squared these are linear blocks.

So therefore I can write it as a single block or I can separate it out basically the way you would create filter structures, these are identical. So basically this is what we are doing and now I have to follow it up with a down sampling by a factor of 2, am I right, that is the process. So I do the filtering followed by the down sampling. Everyone with me so far? H of Z is the anti-aliasing filter.

I want to do down sampling by a factor of 2, as the question was raised, can I group terms to make them not polynomials of Z, but polynomials of higher power of Z? Which is exactly what we have done. Now invoke the property of linearity which says that I can take the down sampler inside okay. So which means that I can take the down sampler inside. So I can insert a down sampler on the upper branch.

I can insert a down sampler on the lower branch. What does, in fact these results are actually what we refer to as a noble identities, I forgot to mention that. These are the noble identities. The noble identities basically tell us which is the more efficient structure from a multi-rate viewpoint. So now if I were to apply the noble identities, identity for this combination it says I can interchange the order in which they are done.

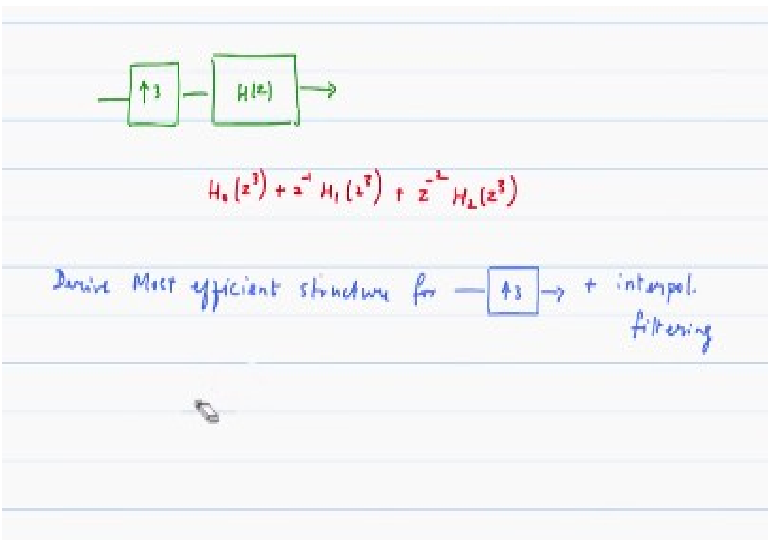
The better order will be downsample by a factor of 2 followed by, H0 of Z square, it will be H0 of Z. Similarly, I cannot operate on Z inverse H 1 of Z square. Because that is not a polynomial with power 2. So I am going to separate it out as a delay element followed by H1 of Z squared by down sampling again apply the noble identity. Noble identity says you can do down sampling first followed by H1 of Z and then of course you have to add the 2 signals up together.

Now tell me what is this? What is this? It is a demultiplexer which ways the commutator switching? Counterclockwise, but in a 2-channel case it does not matter whether goes counter clockwise or counter clockwise, but if the important thing is the, so what is this data going in the upper branch? X0, X2, X4 …. Lower branch X1, X3, X5 … okay. Each of these is getting processed by a filter of half the length and then added to produce the output.

Nothing is wasted, no computation is wasted because you made it as efficient as you can because in the original case what were you doing. You took X of N, you multiplied it with H0, you filtered it with H1, then you threw away half the samples and then finally obtained the output. Now this you can see is the power of multirate signal processing. So whether it is upsampling or downsampling, with the appropriate filtering, this is what we refer to as polyphase decomposition.

Polyphase decomposition probably was the discovery that made multirate signal processing, sort of opened up the world of multirate signal processing. So what you will now find is that any filtering operation, you can now apply the noble identities to make it very efficient because just as you did the down sampling by a factor of 2, what I would encourage you to think about is the up sampling by let us say by a factor of 3.

(Refer Slide Time: 28:22)



Up sampling by a factor of 3, which you would then have to follow up with an interpolation filter H of Z which I would like you to now think of it in terms of H0 of Z cubed + Z inverse H1 of Z cubed + Z-2 of H2 of Z cubed okay. Instead of 1 polynomial, I wrote it as 3 polynomials each of which are powers of 3, but with the Z-1 and the Z-2 you get all the powers of Z.

So using this equivalent representation, can you draw for me the most efficient interpolation operation? Derive the most efficient, derive the most efficient interpolation, and you will find that you are not doing any operation on any nonzero samples, it is the beauty of this is most efficient structure for interpolation, for upsampling by 3, + interpolation filtering okay. So what you will find is that we have introduced 2 very powerful concepts.

One is the noble identities, initially, first glance of the noble identity says well you know what use is this because it is not fitting in with my framework. The next comes polyphase decomposition which says you can always split your filter into sub filters and then all of a sudden the opportunities to use the decomposition and you will find that in this case just like you got the demultiplexer in the case of the down and the anti-aliasing filtering.

What you will find in this case will be the multiplexing operation. Please try it out we will start from here and build on it in the next lecture. We will see you tomorrow. Thank you.