Mathematical Methods and Techniques in Signal Processing - I

Prof. Shayan Srinivasa Garani
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore

Lecture – 44
Efficient architecture for fractional decimator

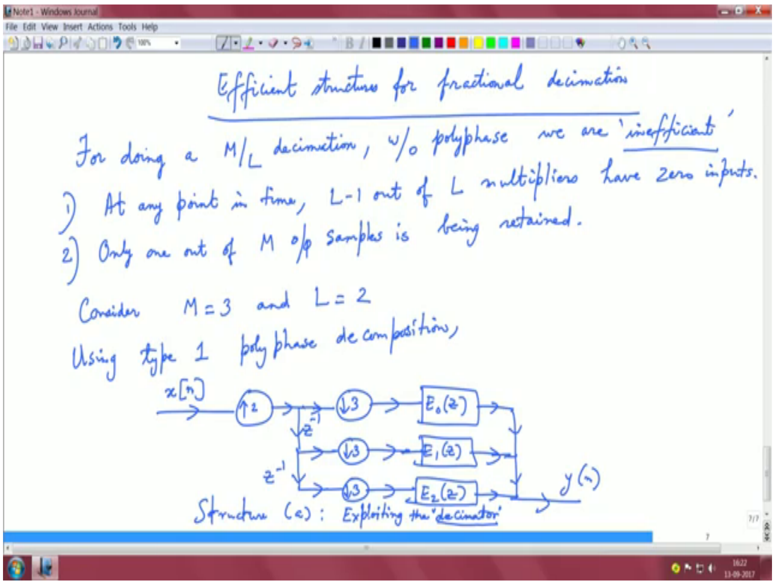
So, in the last lecture right we just studied, we studied the efficient realizations for decimation and interpolation filters. We saw what advantage we get in terms of computational complexity and you know that the multipliers are not really resting right, and we did this for an integer decimation rate and integer expansion rate considering M equals 2 L equals 2. We also discussed how certain properties on fir filters.

That mean if we assume that the fir filter has a symmetric impulse response, how that would affect the polyphase components and this we got a basic feel about some interesting properties with the polyphase filters, how those and we were, we thought how we can incorporate these aspects into efficient realization. One is using efficient structures themselves for realizing decimation interpolation filters, second is some additional properties on the base filter itself, how we can bring that factor also into efficient realization.

So, suppose we want fractional sampling rates and let us assume that we want a an M by L decimation right. So, so we want to alter the rate 1.5 reduction in sampling rate soon and so forth. How can we realize efficient structures? This is the question for rational sampling rate conversion, and if you recall at the very beginning of the multi rate module, we discussed that some of the basic problems of would be sampling rate conversion right.

Mean we would want to go from 32 kilohertz to 48 kilohertz or 30, you know 44 kilo hertz to 40 48 kilohertz to 44.1 kilohertz and so on and so forth right. So, if we need these conversion rates that are possibly rational rate conversions, how can we go about doing these? So, now we are now at a stage where we want an efficient structure for realizing these sampling rate conversions. So, let us get started.

(Refer Slide Time: 03:08)



Efficient structures for fractional decimation

For doing a M/L decimation, w/o polyphase we are 'inefficient'

1) At any point in time, L-1 out of L multipliers have zero inputs.

2) Only one out of M o/p samples is being retained.

Consider M = 3 and L = 2

Using type 1 polyphase decomposition,

Structure (a): Exploiting the 'decimator'

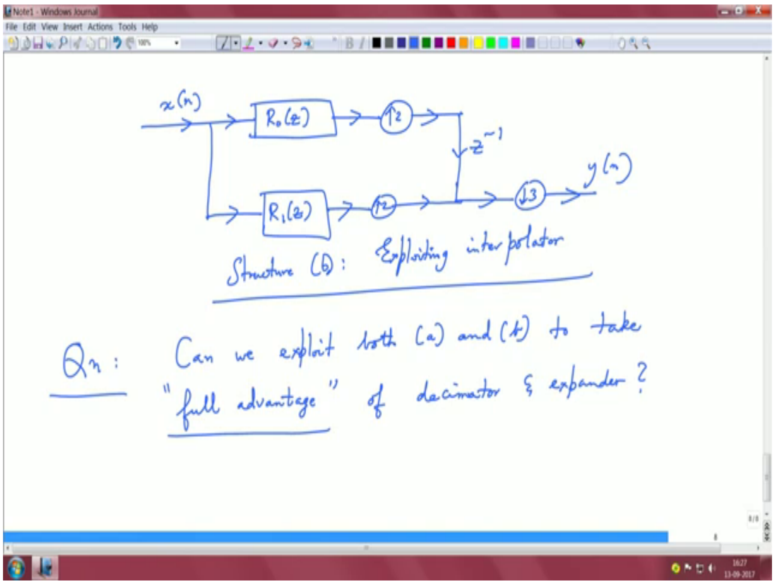
For doing an M by L decimation without polyphase, we are very inefficient, why at any point in time L minus 1, out of L multipliers have zero inputs; that means, we are just filtering these zeros second only one out of M output samples is being retained. I think this is very very important, these two points are really very important for us, what happens in up sampling, what happens in down sampling.

Now, let us consider an example here. Suppose M equals 3 and L equals 2 right. So, we have studied polyphase decomposition, we have studied noble identities. Let us see what we can do right. One may think of adopting the following architecture using type one polyphase decomposition. We use a type one polyphase decomposition, we can think of the following structure right, we have x of n discrete time signal, we have sample this by 2. Now we have three branches, because its type one polyphase decomposition, we look at it from the decimation filter decimation filter perspective.

So, we down sample by three feed this to a filter which is E naught of z. There is a delay here. A down sample this by 3 feed this to a filter which is E 1 of z and one more delay path for the last polyphase component, and then we sum all the all the outputs and we get what we want and we have accomplished an M by L conversion, this is one structure, call the structure a and this is exploiting the decimator right. If I gave this problem to one of you in the class, do an M by L decimation. Probably if you if you just studied noble identities polyphase decomposition, this is one realization one of you could provide.

Somebody else in the class can still argue. Well why should I exploit just the decimator here, because we also studied what we could do with a type two polyphase decomposition for the interpolation filter right, and we will exploit it from the interpolation filter perspective right. So, let us do that exercise.

(Refer Slide Time: 08:15)



Structure (b): Exploiting interpolator

Qn: Can we exploit both (a) and (b) to take "full advantage" of decimator & expander?

So, you could start with the signal x of n, you filter this through R naught of z, this is followed by up sampling by 2. There is a delay, then you have R 1 of z, another up sampler here followed by down sampling by 3 and then y of n, we call this structure b, this is by exploiting the interpolator

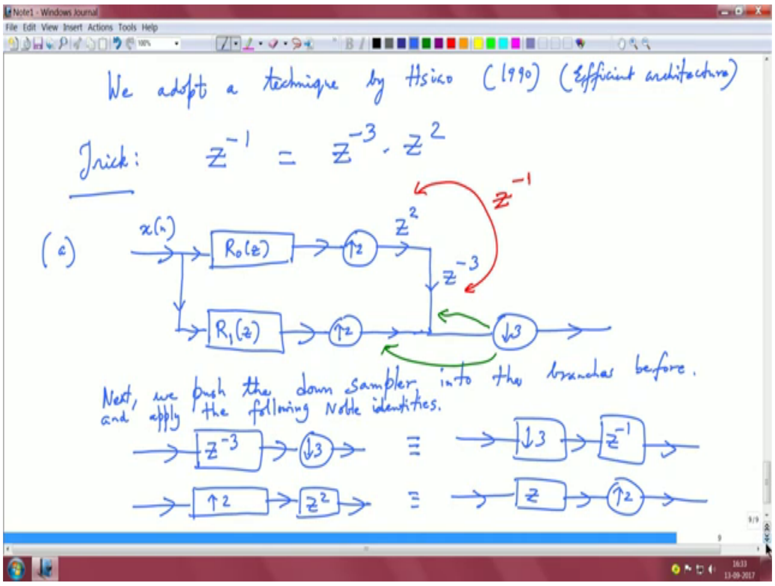
Now, one of you may question if structure b is efficient or structure a is efficient and realize the possibility that both of these are still inefficient, because in structure a you are up sampling by 2 here; that means, you are inserting an additional zero that goes through the filtering process here, which is not efficient right, because we said these two things have make an M by L decimation inefficient, because you have not exploited the polyphase at the interpolation side here, at the expansion site followed by some filtering here.

Similarly, one can argue that, well though you exploited the interpolator here by efficient realization, you are failing here from the decimation perspective, because anyway you are throwing away two samples when you are down sampling by 3. Now, can we combined the question that one has to ask is, can we exploit both a and b to take full advantage of this emitter and expander right. This is something interesting umm, at this point to ponder about, are we at the dead end right?

I mean I did expansion decimator, I mean expand you know polyphase realization for expansion with filtering then designation and the other way around, are we at a dead end here. So, this is an important question right when we discussed in the very beginning of the multirate central processing module. Looking at a conversion rate from 32 kilohertz to 48 or 48 to 32 3 by 2 decimation is happening right. For 48 if I want to reduce it to 32, it is 1.5 times reduction; that is exactly the same problem here a practical problem.

So, this question was addressed by gentleman, by name Xiao way back in 1990 and that led into a really truly efficient architecture for an M by L decimator ok, and this is very simple as you can see with a few elementary tricks, we can we can just get to the, get to the answer.

(Refer Slide Time: 13:05)



We adopt a technique by Hsico (1990) (Efficient architecture)

Trick: $z^{-1} = z^{-3} . z^{2}$

Next, we push the down sampler into the branches before. and apply the following Noble identities.

$$\rightarrow \boxed{z^{-3}} \rightarrow \boxed{\downarrow 3} \rightarrow \;\equiv\; \rightarrow \boxed{\downarrow 3} \rightarrow \boxed{z^{-1}} \rightarrow$$

$$\rightarrow \boxed{\uparrow 2} \rightarrow \boxed{z^{2}} \rightarrow \;\equiv\; \rightarrow \boxed{z} \rightarrow \boxed{\uparrow 2} \rightarrow$$

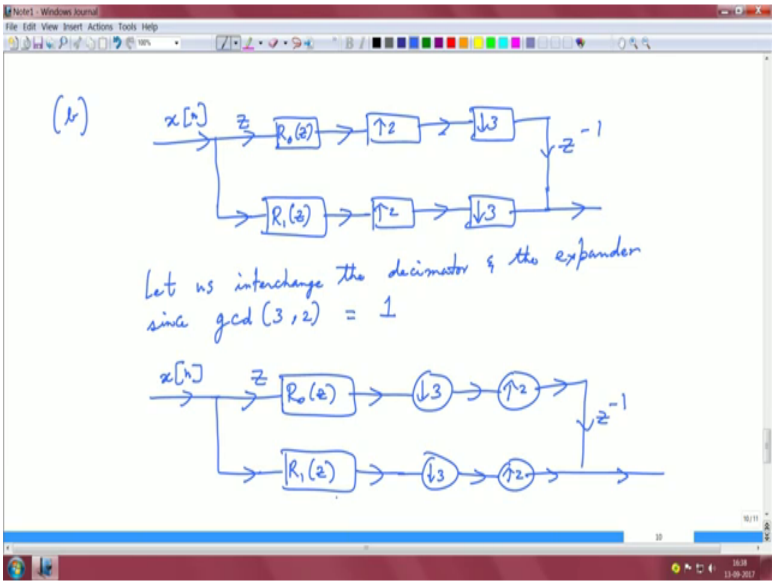
So, we adopt a technique by Hsico way back in 1990, it is published in the transactions, a trippy transactions and this uses an efficient architecture for an M by L conversion. So, the trick I think is important, what the trick is. So, if you think about a delay right, this delay is z power minus 1 can be written as z power minus 3 times z square, no big deal. If you just say what is a big deal, if I say if I write z power minus 1 as z power minus 3 and z power ma z z square, but observe the powers 3 and 2 that are appearing here, and carefully think about the M equals 3 and L equals 2 and see what you could do right

So, if you we will see how this trick is useful right. Let us start with this step, we will start with type two decomposition. At the interpolation stage we have R naught of z followed by an up sampler, then we have R 1 of z followed by an up sampler and here we had a delay element right, we had a delay element and this delay element we will write it as z power minus 3 here, at this branch we will say z square and this is this unit delay z power minus 1 followed by down sampling by 3

So, this is the first step, the next step is by pushing this down sampler into each of the branches, as what we have usually done, and then now think about applying noble identities to this pass. So, let us now the second. This is the first step right, next what we do is, we push the down sampler into the branches before, therefore, and this would go in here as well as here ok

And what we would do is, we will use the following identities z power minus 3 followed by down sampling by 3. This is just like noble identities right, we can we can write this as down sampling by 3 followed by a delay and then we have an up sampler by 2 followed by z square to be equivalent to an advancement, which is z followed by an up sampler by 2. So, this will simplify things a little bit ok. So, what we do is, we following so we have now x of n.

(Refer Slide Time: 18:50)



Let us interchange the decimator & the expander since gcd (3, 2) = 1

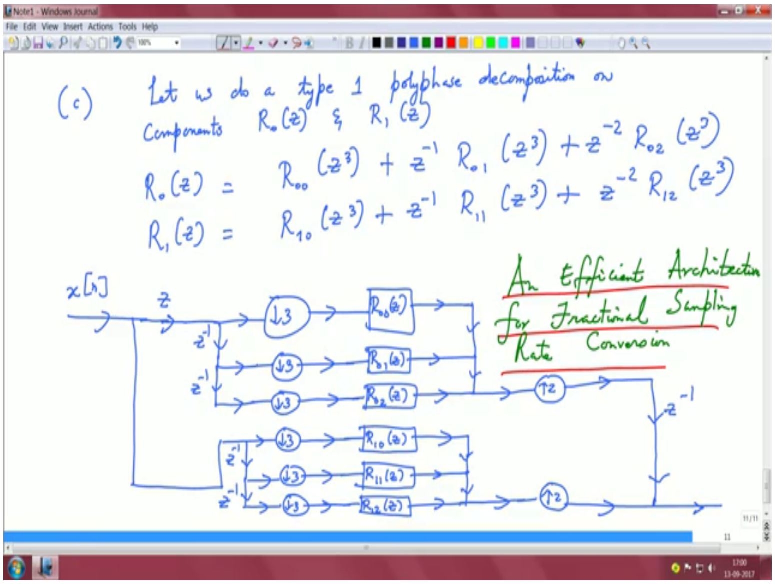
So, I just write this z here R naught z up sample by 2 down sample by 3, a delay followed by R 1 z up sample by 2 and then down sample by 3. So, you might be wondering ha, where this big z appear right, if you apply this noble identity right. So, you have at this point you have up sampling by 2 followed by z square. We replace this by z with up sampling by 2. Now this is like a cascade of R naught of z and z you can just exchange the cascade right, you can say is R naught of z times z is basically z times R naught of z and that is what is happening here

Now, this seems to be, this seems to be good, we have done some simplification, but we are not let there, if you can sort of realize. So, because we have R naught of z and R 1 of z and then we have these, these up samplers followed by down samplers. Now there is another trick, we could invoke which is from the interconnecting systems. Notice that you have L equals 2 and M equals 3 the gcd of L and M is 1. So, you can basically exchange them

So, the next step is to realize that we can interchange the decimator and the expander. Since gcd of 3 and 2 is 1 ok. We do that and what we get is, x of n, the rest is all the same, we are down sampling by 3, sampling by 2. We have a delay element parable of c sample by 2; since its down sample by this, is down sample by 3 followed by up sampling by 2 and we just exchange the two, we just swap these two

Now, what we have to do is. Now these filters are not in R 1, these are polyphase filters obtained from interpolation. We can do a type 1 polyphase decomposition on these filters further, because this is basically filtering followed by decimation. We will do a type one polyphase decomposition over R naught and R 1 further

(Refer Slide Time: 23:52)



(c) Let us do a type 1 polyphase decomposition on Components $R_0(z)$ & $R_1(z)$

$$R_0(z) = R_{00}(z^3) + z^{-1} R_{01}(z^3) + z^{-2} R_{02}(z^3)$$

$$R_1(z) = R_{10}(z^3) + z^{-1} R_{11}(z^3) + z^{-2} R_{12}(z^3)$$

An Efficient Architecture for Fractional Sampling Rate Conversion

So, let us to a type one polyphase decomposition on components R naught of z and R 1 of z. So, if we did this, say R naught of z can be written as some R naught naught z cube plus z power minus 1 R naught 1 of z cube plus 2 delay elements R naught 2 z cube, and we have R 1 of z is R 1 0 z cube delay with R 1 1 of z cube plus 2 delays R 1 2 of z cube and then we will sketch this.

So, basically we have x of n, we have a z here. So, we have down sample by 3. So, basically then you apply noble identities right, you plug the polyphase representation and apply noble identities, I get R naught of R naught naught of z delay down sample by 3 R naught 1 of z and one more delay down sample by 3 followed by filtering with this filter R naught 2 of z, and then I combine all of these components together, and then I up sample by 2 followed by a delay right. And then similarly we have one more branch and from which we have to realize something similar to what we did.

So, we have down sample by 3 followed by this filter R 10 of z plus branch, then we have a delay. So, I think I would just erase some of these blocks here just to get this diagram a little symmetric followed by down sampling by 3 with R 1 1 of z and then 1 more delay element here down sample by 3, filter this through R 1 2 of z. Now we combine all of these up sample this by 2 and then sum this up at this point right.

And this is an efficient realization because we started off with a type two decomposition from the interpolation side and then we did a type one decomposition, exploiting the decimation side and then we have really, basically we are very efficient we are not throwing away samples as a part of the decimation process right. We are not wasting our efficiency on that and then. Similarly we are not just filtering these zeros, because we started off with an efficient realization for interpolation in the very first step right and this is an efficient architecture for fractional sampling rate conversion

I mean you could decimate, you could reduce rate by 1.5, you could increase rate by 1.5, you could do any rational sampling rate conversion, I mean here I mentioned this is a fractional sampling rate, because any we are doing an M by L, we are doing a reduction by 1.5 times ok. So, this is a sort of the goal and we have applied all the tricks that we could think of we did polyphase decomposition, we then applied noble identities. We looked at inter connecting systems, we did both type one and type two. So, all the tricks that we learnt as identities, we have basically taken care of in this in this architecture

Now, we can imagine why this gcd has to be, why this gcd is important right and if we want a nearest M and L parameter; such as such that the gcd of M comma L equals 1. So, if you think about 48 kilohertz and 44.1 kilohertz, look at the ratio 48 by 441 and reduce this to the nearest rate that you nearest M by L rates that you want and then you can go about realization of efficient architectures for this.

So, I think this is a very important first step and then if you. Now we did not think about how we could exploit the base filter itself, I mean I leave this as a homework exercise. Suppose I give you a base filter which is an fir filter and that has a symmetric impulse response. Now take that property into account and then see how these polyphase filters could be more efficiently constructed right. I mean how you could do better filtering operations with these filters.

At this point you might just wonder about one little detail, which is you have this element, which is basically z, which is an anticipation element, it is not a delay right, it is an anticipation element and one way wonder how can you realize this in practice, and it is not too difficult to think about it and what is what is this delay this anticipation element telling us; that means, we need to look at one sample in the future right.

I mean this architecture would seem futile, if you are thinking about serial communication where you are sending one sample at a time for processing and there you really cannot anticipate anything right, because everything is the present or whatever you have in the past, but if you buffer these samples by some number of samples, initially then you can point to a particular location in your buffer. I mean you can always basically get over this anticipation, because you, since you buffered you already have some samples at one shot.

So, therefore, you can really realize this advancement by figuring out, what that sample has to be and I think post that everything would be basically pipeline and the whole architecture can be very efficiently realized in hardware.

So, this is a little bit of detail, I think a little bit of thought from your side can make you realize and understand that this is indeed a very practical architecture by just buffering, you can you can realize everything to work as is ok. I think with this we are done with this lecture.