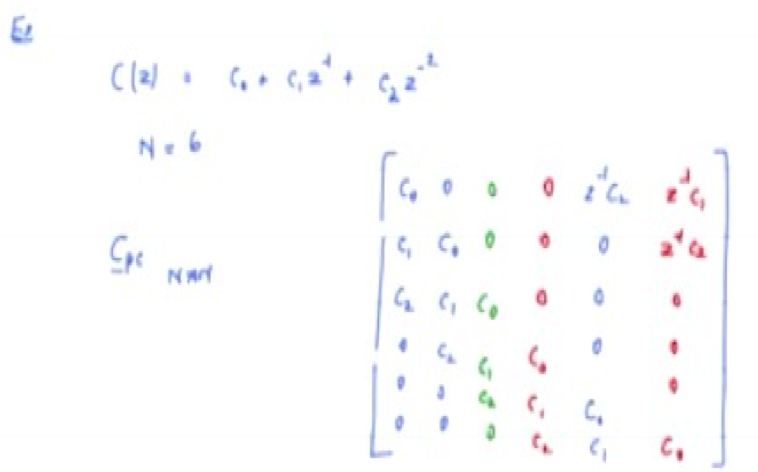
Multirate Digital Signal Processing

Prof. David Koilpillai
Department of Electrical Engineering
Indian Institute of Technology – Madras

Lecture - 35 (Part-2)
Pseudo-Circulant Structure

Okay, so the pseudo-circulant is a very important element and I thought it would be helpful for us to begin at this point, but let us do it with an example, so that it helps us solidify the results that we have been working with, okay.

(Refer Slide Time: 01:06)

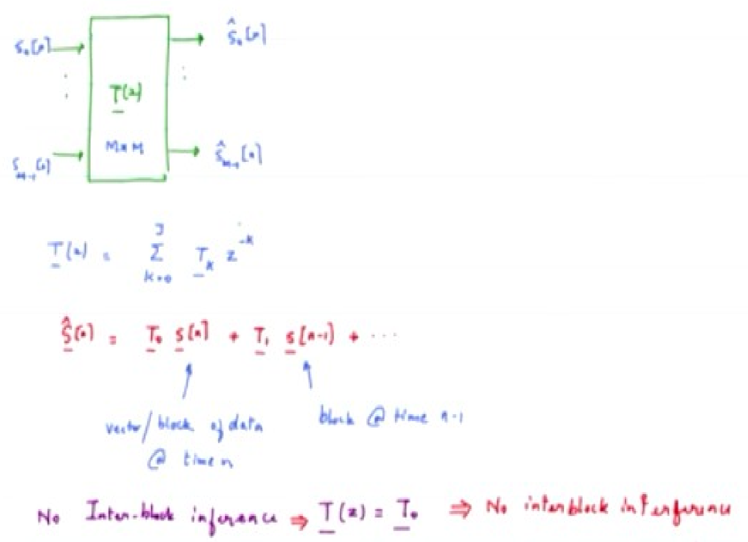


So if I have, an example, if I have a channel C (Z) = C0+C1 Z inverse +C2 (Z-2), okay and I want to have N=6 that means I am up sampling by a factor of 6. I would like us to construct what the pseudo-circulant matrix looks like and then. So if you notice that the pseudo-circulant matrix, CPC matrix has dimension NxN, so what you should get is a 6x6 matrix. So if we write down the pseudo-circulant, the first column will be C0 C1 C2 0 0 0 okay.

And the next one will be 0 C0 C1 C2 0 0, quickly write down the third one 0 0 C0 C1 C2 0. Did I miss something. If I miss something, please let me know. Next one is 0 0 0 C0 C1 C2. Now the wrap around starts to occur. Z inverse C2 okay and this is 0 0 0 C0 C1 and Z inverse-1 C1 Z inverse C2 0 0 0 and C0. That is the pseudo-circulant matrix that comes out to be when you do this, when you have up sampling by 6 followed by this channel followed by the down sample okay.

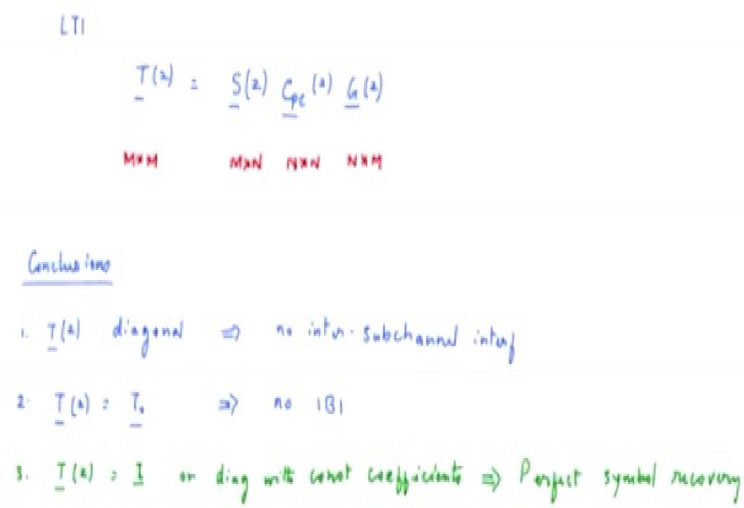
So basically this is the structure that we are talking about. So the familiarity of the pseudo-circulant and its interpretation are very important for us and what does this Z inverse C0 actually mean? You have to be very careful in implement, in understanding and interpreting that. So that is the first task for today. Okay, so here is the representation that we are trying to get.

(Refer Slide Time: 03:49)



That the input-output relationship for this structure. Where is the transmultiplexer structure, for this structure, input-output relationship is in terms of the T (Z) between S0 and this. So that is what we are trying to write down okay. So the overall transfer function that we are trying to represent the LT.

(Refer Slide Time: 04:20)



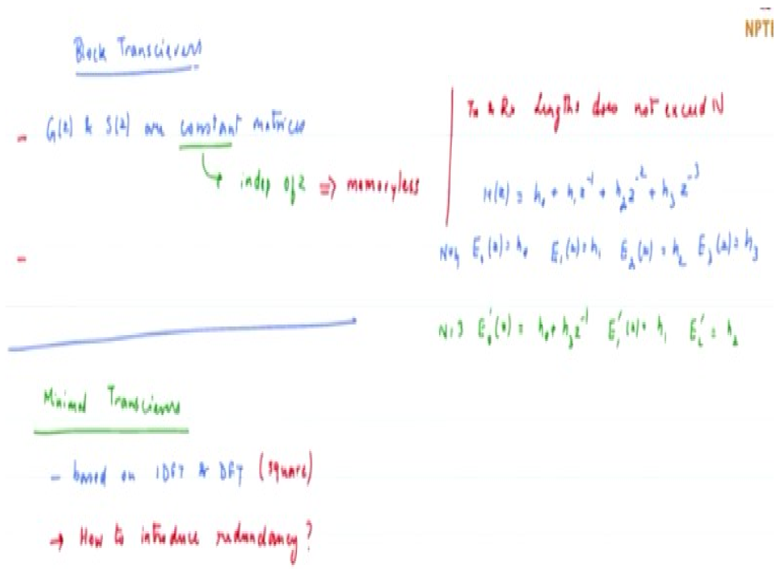
It is an LTI system T (Z) overall is the analysis filter bank S (Z) CPC (Z) * G (Z) okay. Dimensions basically T is a M x M system. So this is a M XN. CPC is N x N and this is N x M. M inputs with redundancy becomes N, where N is larger, goes through the channel treated as a N x N like a MIMO system and then is again reduced, redundancy is removed and you get a MM signals, M signals okay.

So the observations, which we want to what we want to highlight is the following. Let us write this once more, the conclusions. Conclusions say that 1 is T (Z). It is a matrix M x M. If this 1 turns out to be diagonal, I think we have written it down previously. I will just write it for completeness. Diagonal implies no inter subchannel interference okay. That is very important for us in the design of the system okay and if T (Z)= a constant matrix that means no inter block interference.

I will write it as IBI and of course if T (Z) = an identity matrix or any other diagonal matrix, diagonal matrix with constant coefficients, then we call it a perfect symbol recovery. This is perfect symbol recovery. So ideally we would like to design our system to get it to perfect symbol recovery or at least try to see how do we design our system, such that we can achieve a performance close to a perfect symbol recovery system.

So that is that is a very important outcome of our study and of our discussions. Okay now here is the, we use the term block transceivers.

(Refer Slide Time: 07:09)



Block transceivers now coming from a communications background, we may have certain understanding of okay what is a block transceiver. Now from let me tell you where the intersection of communications and signal processing occurs. So basically when we said blocking what did we mean? We took a signal and then we created a block, one block of data and this passed through the channel and produced another block of data. Now if there is inter block interference, when I look at the output, I cannot make a decision on the input.

Because there is, there is the effect of current input block plus some previous input blocks are also present. But if I am able to do the processing, such that my output depends only on my current input block, then I can do what is called block in-block out. I can make a decision, take the next block, next. That is what is block transceiver. So basically you have eliminated interblock interference. So the first step is the elimination of inter block.

So here is the interpretation of what a block transceiver would require. So the first thing we will take the case where the polynomials G (Z) and S (Z) are constant matrices. This is very often the case; we are; constant matrices. So here is what this means okay. Constant matrices means these are actually do not depend on Z. They are independent of Z. Now why is this important? Let me just go back to this one. Now if G (Z) was a polynomial instead of a constant, what will happen?

It will already introduce for you memory between the successive blocks. Now so what we are saying is at the transmitter, I am giving you a signal that does not have memory. Whatever is the current input data, what is, that is what is getting it to you. So that is why we are making the assumption. It is not a restrictive assumption. At the transmitter, do not create a mixing of the signals.

So independent of Z also means that we are talking about, this means that we are talking about a memory-less type of transmitted signal okay. There is no memory built in. You are not sort of combining past signals. Now if this has been satisfied, so this is condition number 1. Second condition if the transmit and receive, so basically when can we say this is the, this condition will be satisfied. When will C (Z) and G (Z) and S (Z) be.

So this will be satisfied when the transmit and receive filters or the analysis and synthesis filters, their lengths does not exceed N. N is the number of polyphase components that you are generating. So just as a quick example okay, just to validate this one. So if I have a filter, which has got 4 coefficients. These are, it could be either transmit or either the synthesis filter or the analysis filter, just a generic one H1 Z inverse + H2 Z-2 + H3 Z-3 okay.

Now if I were to ask you to get me the 4 polyphase components, you would say E0 (Z) is actually a constant H0, E1 (Z) is a constant H1, E2 (Z) is a constant H2 and E3 (Z) is a constant which is H3. This is what I mean by saying that the lengths do not exceed N. As long as they do not exceed N, your polyphase components will be constants, which means that this G and S are constant matrices and therefore there is no memory in your modulation process, okay.

Now just to highlight it, so this is the case where the number of polyphase components N=4. Now if I take the case where the number of polyphase components, I choose to be 3. Then I ask you to do the polyphase components E0 dash (Z) is equal to what H0+H3 Z inverse okay. E1 prime (Z) is H1, E2 prime = H2 okay. So notice that just a simple illustrations shows that if you had N=4, this particular channel would have given you constant matrices.

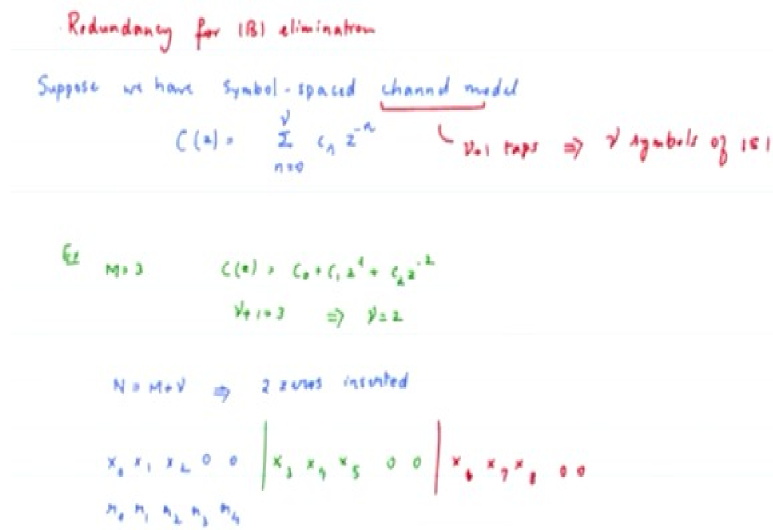
This particular polyphase decomposition, otherwise you will get this. So this is equal to H2 okay. So this is what we mean by the information okay. So keep this as one aspect. I am interested in block transceivers. I am going to look at the case where G and S are constant matrices, that means my length of my filters are chosen, such that they are the same as the size N, which is the redundancy that I am going to introduce or with which I am going to transmit my channel okay.

Now here is another element, which is very important for us. Minimal block transceivers, okay minimal transceivers, one of the simplest and the one that we have already looked at is if you choose this G and S to be the IDFT matrix and the DFT matrix, because you know you can see the polyphase structure. There is a notion of the IDFT and DFT being useful in these types of environments. So based on the IDFT and the DFT okay, very important.

These are square right. These are square matrices. Now when I want to introduce redundancy what are the structure of G and S. They were rectangular matrices, remember the trapezoidal structure that we had. So the key question is, wait, this is going to be a problem for me, because IDFT and DFT are always square right, I cannot. So how do you introduce redundancy? If you are interested in the IDFT/DFT family, how do you introduce redundancy.

Again this is a very, very important question, which is what we will address because that is important and the key question is how are we going to introduce redundancy and how are you going to address it with this particular setup okay and it turns out that you already know the answer to it.

(Refer Slide Time: 14:15)



So let me just refresh your memory as to why you know the answer to this already. So the first task for us is to design block transceivers, where there is no inter block interference. So basically I want to use the redundancy first to eliminate interblock interference. For inter block interference elimination okay, so take the general case that we have a channel model. Suppose we have, this is again from digital communications.

We have a symbol spaced equivalent channel model okay. I am assuming you are familiar with what that means. If I have a discrete-time sample signal, I can represent the relationship between the input and output through the symbol spaced channel model, okay. Symbol spaced channel model given by C (Z) which is given by N=0 through eta or nu. This is CN Z power -N okay. So this is a channel model with nu+1 taps which implies there are nu symbols of ISI.

This is an ISI channel, symbols of inter symbol interference. So this is a standard notation. This is the channel model that we have and so now the important question that that we will do is just a simple example. So take the case where M=3 that means I have 3 parallel channels that are going in okay and I have C (Z) = C0 + C1 Z inverse + C2 Z-2 okay. This is a scalar example, just to illustrate the point. So basically this means that nu+1=3, okay.

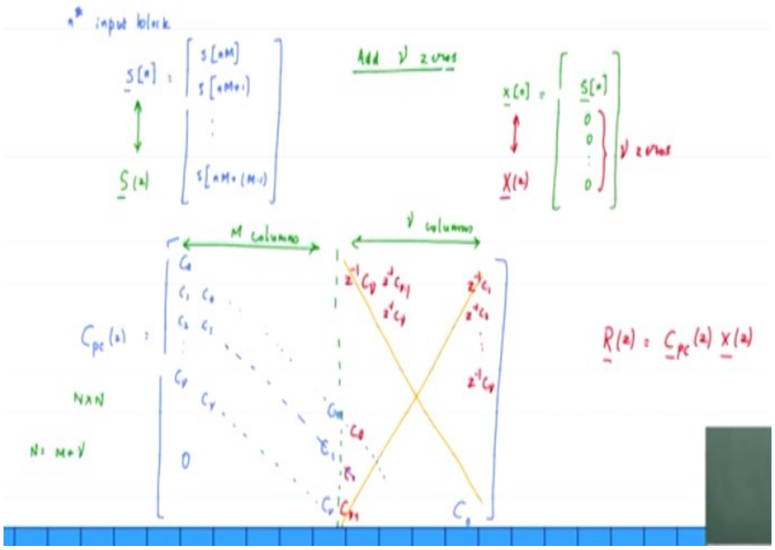
So which means that nu=2, 2 symbols of ISI, the previous symbol and the one before that will cause the interference. So now if I tell you that let N, what is N? N is the with the redundancy built in. Take N to be = M + nu okay. So it is 3+2=5 and wherever you do not have any data, we just say insert zeros. So which means there are 2 zeros inserted, okay. So if I have the situation where R0, R1, R2, R3, R4 basically need to be transmitted.

What I am going to do is after a block which is R0, R1, and R2, I am going to insert 2 zeros okay. So basically and then after that will come R3, R4, R5 followed by 2 zeros, then R6, R7, R8. I do not know if I have used R or some X, but basically zero insertion is the key here okay. R8 followed by 2 zeros. So what this basically says is that there is and I have a channel. Now pass this data through the channel, what will you get?

You will get a, let me just make sure I have got the correct notation. What is the, it is actually X and R. I am sorry, so let us go back to X and R okay. Just change this to X okay. Give me a minute. So this is X0, X1, X2, X3, X4, X5 and X6, X7, X8 okay. Now passing through this channel what we will get is R0, R1, R2, R3, R4 right. When you convolve, you will get the longer sequence okay and here it will be again you will get the corresponding vector.

So then the numbering you have to be careful with, but basically you can see that there will not be any interference between the block number 0 and block number 1 and this is a very important structure for us to work with okay. So now let us quickly see how to build on this and develop it okay. So if I were to write the Nth block in this fashion okay, the Nth input block, that is at the extreme input.

(Refer Slide Time: 19:51)



The Nth input block S (N) the structure of this will be S (NM), S (NM+1),… S (NM+M-1) okay. This is the blocking operation, that is happening right at the input okay. Now to this, we have to add some number of zeros, add nu zeros, that number of, that depends on the number of taps in the channel. That will produce for me the vector that will get processed through the channel. So that will be nothing but the vector S (N) which is the blocked version of the input.

And then a bunch of zeros and this will be this vector and the number of zeros that we have introduced is actually nu zeros. This is zero padding and I am just writing it in the form of a matrix, matrix form okay. But the most important step is can you know, if I am going to feed this to my channel okay, can you now tell me what should be the structure, what will be the structure of the pseudo-circulant matrix that is caused by the channel okay.

So here is the key part of this discussion pseudo-circulant of Z okay. So it is a new tap channel. So the first column is going to be C0, C1, C2, … C nu okay. By the way, the first thing is the dimensions of the pseudo-circulant matrix is actually NxN where N=M+nu okay. Now I am basically writing down the matrix that I need for the convolution of the vector X with the pseudo circulant matrix okay. So the next column is going to be C0, C1 all the way to the 1 lower row C nu.

So basically what we will get is a diagonal matrix okay. So the last row is going to be C0, the previous one will get basically, you will get a Toeplitz structure. So you will get all C1 along this diagonal and then all C nu's along this diagonal okay. So basically look at the column which has got C nu at the bottom that will have C1 here and C0 here. The one immediately after that is the column of interest.

This will be C nu-1. This will be C1, this will be C0 and this is where the pseudo-circulant will show up. This will be Z inverse * C nu. Then after that and of course because there are several, these are all 0 valued entries. There are lots of zeros here okay. Then the next column, will have Z inverse C nu -1, Z inverse C nu and then the last column is going to have, I just write it a little further away.

The last column is going to have C0 over here, no this will be, yeah this will be C0 all the way to the end. The rest of the entries are Z inverse C1, Z inverse C2, … Z inverse C nu okay. Now if I were to draw a line here, basically until the C0 to C nu form a Toeplitz portion. Can you tell me what will be the dimension, how many columns have I captured here? Look at the last row, it goes from C0 to C nu-1. So it is actually nu rows, right nu columns. This is new columns okay.

If that is nu, then this one has to be M, M columns. So it is a NxN matrix where there is a banded Toeplitz structure for the first M columns and then after that the pseudo-circulant effect starts to kick in okay. Now if I look at this sequence of blocks that are coming in, then I can talk about a block Z transform, which will be the block Z transform of this will be S (Z), that means S (0) + S (1) Z inverse, basically this is block Z transform.

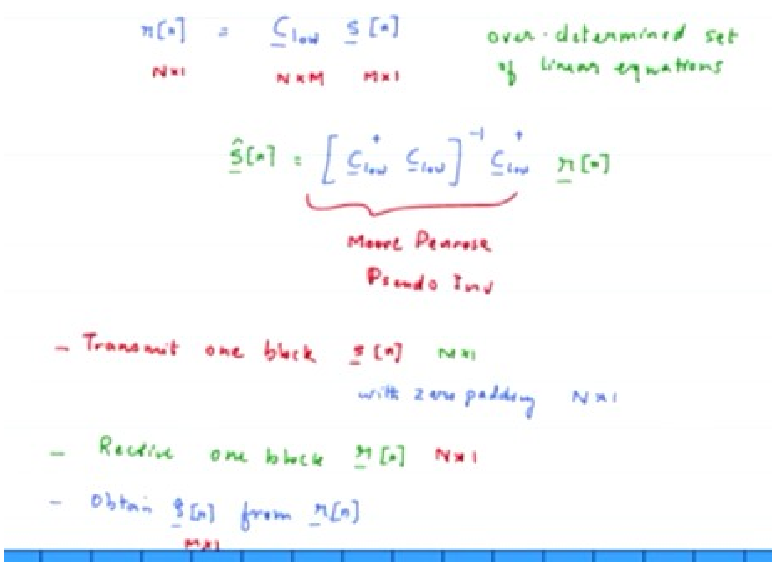
In the same way, I talk about a block Z transform for X that will be X (Z) okay. So basically these vectors that are written in the form of a Z transform. So given this notation, the received vector okay in the block Z transform, there is a R that comes out of the pseudo-circulant matrix. The block Z transform of the received vector is nothing but CPC (Z)*X (Z), basically X (Z) okay. Now I wanted you to pay close attention.

Basically what I have written down is if I visualize it in the time domain, it is a convolution, but if I think of it as a matrix, then I can write it in terms of the Z transform of the input times the transfer function times this, which is what I have done okay. Now please pay close attention to this one result that I am going to point out to you. What is the structure of X? It has got the input vector S followed by a bunch of zeroes right. So what does S, what does X actually do?

That means basically you would have to multiply this CPC with X (Z) right. If you have a bunch of zeros a part of the vector, those columns do not have any effect on my output, am I correct? What did the, where did the zeros kill, what did the zeros kill? They basically kill the last nu columns well that is what got. So if I killed these what did I remove out of my structure? I removed any memory in the system. I removed the block.

So what I need to worry about is this matrix, which is NxM, which is given by this form. So let me just quickly write it down. So we will call this portion of the matrix as C times low, low meaning it is basically looking at the diagonal matrix C0 and then looking at. So this is a NxM matrix. The input-output relationship between the, so there is no memory in the system right, no memory in the system.

(Refer Slide Time: 28:00)



So what we can write down as the input-output relationship is that R (N) the output vector. This is a Nx1 vector, can be written as equal to C low, which is a constant matrix times the vector S

(N) that is the zero padding has been taken out okay. So this is a N x M matrix, where we have kind of took out the last nu columns and this is a Mx1 system without the zero padding this or the redundancy. So this is the input-output relationship.

And the reason you are able to write this equation is because and only because you had those zeros coming into play. They kind of took out the effect of the previous block and therefore you got this equation. Now this is nothing but a over determined set of equations, because N is greater than M, over determined set of linear equations, okay and these can be solved using the pseudo-inverse technique okay.

So basically S hat (N) once you get R (N), you can obtain an estimate of the transmitted symbols using the pseudo-inverse. The pseudo-inverse structure is given by C low transpose conjugate times C low inverse C low transpose conjugate. Again I am sure this is the standard what is called the Moore-Penrose pseudo-inverse. I just mention that. This is the Moore-Penrose pseudo-inverse, which we use all the time for solving these type of system.

Moore-Penrose pseudo-inverse times R (N) okay and this basically gives us the estimate. So the way the whole thing will work is I transmit one block. I transmit one block of data, one block of data is given by S (N) and the dimensions of this is Mx1. With zero padding, it becomes N x 1 with zero padding, it becomes N x 1, then it passes through the channel and from the output of the channel, I receive one block of data.

Receive one block after the channel that is R (N) and that has got dimensions Nx1 okay and through the pseudo-inverse process, we obtain S hat (N) from R (N) okay and by the way this is Mx1. This is the structure of a block transceiver basically this is a channel that has ISI, but we have been able to do the zero padding and bring the whole thing together, so that we can then obtain it in this form and why did we go through this matrix.

And you may say well you know I already have this intuition, you do zero padding, I will get. See it is not trivial to sort of go from scalar to matrix. What we have shown is that actually you have gone into a matrix notation. We have actually introduced the inter block interference in the structure and then showed that if you insert the zeros, you can get the advantage of the, of an ISI free system okay.

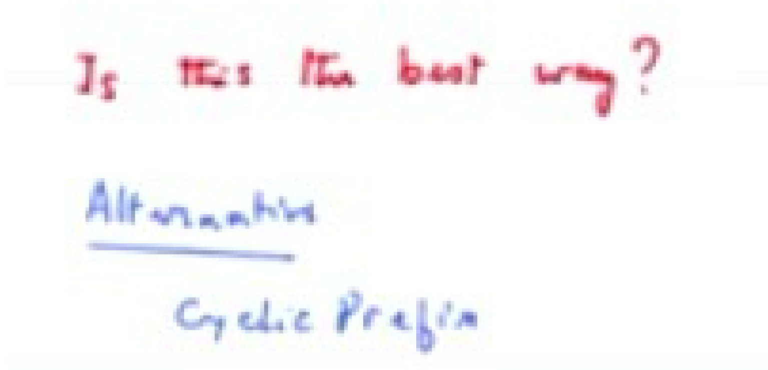
Now the one key element that I just want to mention, I would like close with that okay. It is a padding X (N) that is a vector. If I write down the Z transform of this, it will be X okay. Let us write it again, good question okay. What is X (Z)? It is X (0) * X (1) * Z inverse, … okay. So basically what we are saying is at every, the input-output relationship is given by this. I can look at R (0) and say that it depends only on X (0).

It does not depend on the others. You can write down the right, you can write. That is what I am saying. So R (0) depends on S (0). R(1) depends on S (1). So the powers of Z are getting matched. I am not saying anything about X (Z) having zeros, no, no. X (Z) has a certain structure, but at each for each power Z, if I compare right hand side and left hand side I can get the right. That is a good that you asked to clarify that one okay.

So here is the key observation. So basically this is the structure of a block receiver. Now one of the issues that we are going to have to deal with this, where are the zeroes inserted in X (N). That means there are certain points where you have said that the transmitted signal is 0. So which means that there is nothing going on the channel okay. Point number one is, is that the right thing to do? Basically you are wasting things on the time.

But you are saying, well that is the way I get to get to eliminate the inter block interference okay. But from a power amplifier viewpoint, this is not necessarily a good thing because what you are saying is please transmit at a certain power level and suddenly the signal will go to 0, again it will come up. So it is much better to maintain an uniform transmission. So again the question is, yes we have been able to obtain a structure that has eliminated the inter block interference.

(Refer Slide Time: 35:00)



The question that we ask is, is this the best way? Is this the best way to achieve the inter block? elimination of inter block interference. The answer turns out to be there are other ways of doing it and there is one more way that we would like to introduce that is called, the is this the best way. We do not know yet, but there is another way alternative, which is introducing something known as a cyclic prefix.

The reason a cyclic prefix would be somewhat different would be is that you are transmitting something all the time. There are no zero valued symbols, but it is very important to know how to fill, what to transmit in those point, at those points, so that you can it does not cause you a interfere, undesirable effect okay. So we will find, we will basically define what a cyclic prefix is, take this existing structure that we have developed.

And then redo it for a system where the redundancy is introduced not by zero padding, but by cyclic prefixes and then show that that is actually the better way to do it and that is how OFDM was born okay. So basically we are one step away from actually establishing OFDM as one of the most powerful multi-carrier techniques that we have and this comes because we are able to approach it both from communications and from the signal processing. We will pick it up from here tomorrow. Thank you.