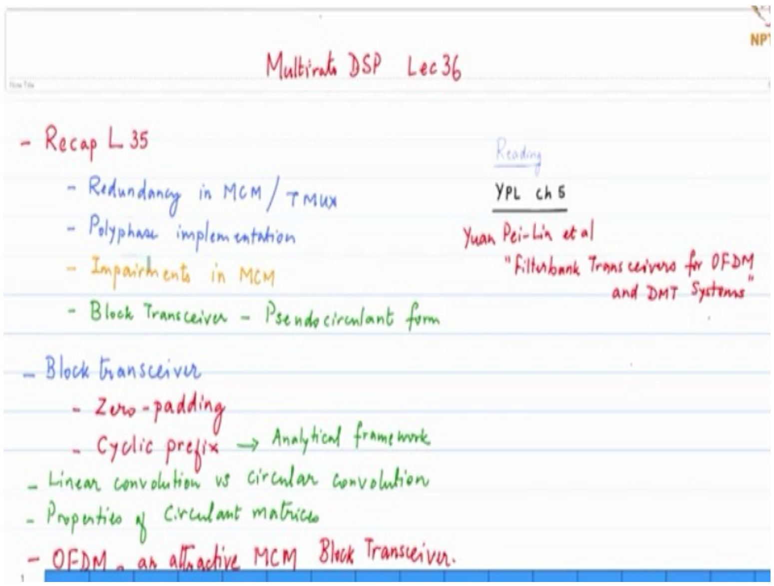
**Multirate Digital Signal Processing**
**Prof. David Koilpillai**
**Department of Electrical Engineering**
**Indian Institute of Technology – Madras**

**Lecture - 36 (Part-1)**
**MCM Impairments and CP-Part 1**

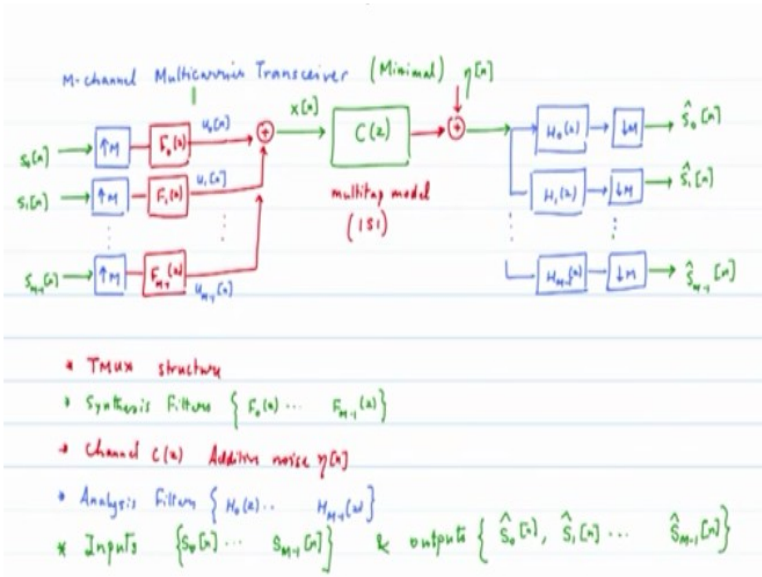
Once more, we touch upon the notion of redundancy.

**(Refer Slide Time: 00:19)**




How it comes into play in the context of the polyphase implementation and how it starts to show ways of dealing with the impairments.

**(Refer Slide Time: 00:30)**

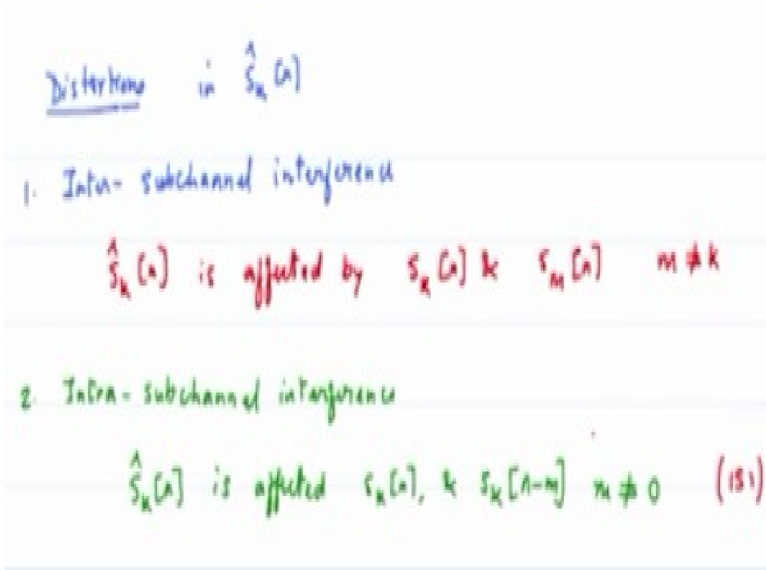



So that is the starting point for our discussion today. So the multicarrier transceiver that we are talking about is one that combines M channels, passes it through the channel, splits it back into M channels. Now this would be a minimal configuration for the multicarrier transceiver. If this M was greater, was by a number factor N which is greater than M, then you would get a transceiver with redundancy.

The notations again the, at transmitter we have the synthesis structure. At the receiver, we have the analysis structure. The input block, there are M inputs, there are M outputs. So the redundancy that comes in will introduce some additional signals, but they have to be removed when you actually present the output. So whatever is happening inside of this block is the parts that we are trying to design.

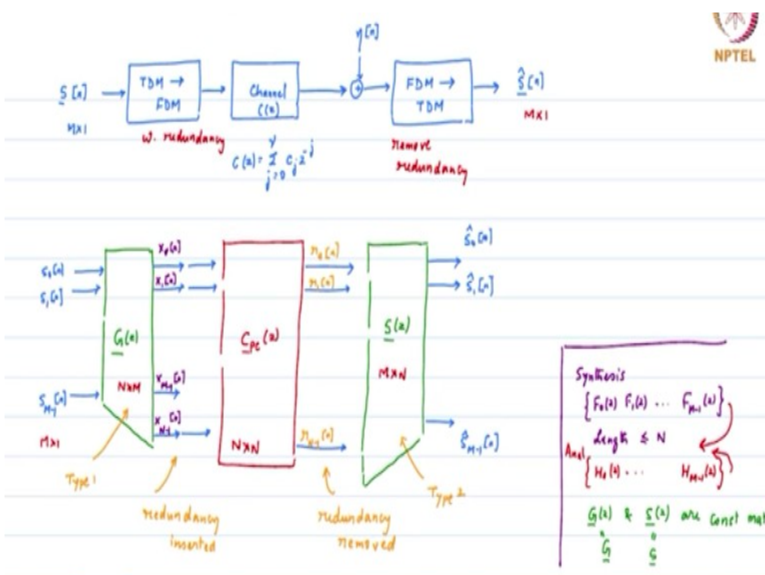
So that we can take care of the impairments that will happen in the channel and also in the block processing okay. So a couple of other observations again just by way of reminder we have. If you think of each of these as subchannels S0, S1, SM-1.

**(Refer Slide Time: 01:44)**




Then they could interfere with each other that would be inter subchannel interference and then you could have ISI between the elements or the symbols of one subchannel that would be intra subchannel interference. Both types of interference need to be addressed and need to be resolved okay. So if I were to ask you to describe what we have done so far, so this is a good way to capture the information that we have. So I have a vector that I want to transmit or a block of data okay.

**(Refer Slide Time: 02:19)**




So this is S (N), this is a M x 1 vector. I would like to pass it through a converter that does time division multiplexing to frequency division multiplexing. That is the synthesis portion and we would like to introduce redundancy. With redundancy, we have not yet exactly specified, but we know that there will be redundancy that will be brought into the picture. Then this is the signal that passes through the channel, the signal with redundancy passes through the channel/

The channel model that we gave yesterday said that there are nu+1 taps C(Z)=J=0 to nu C subscript J (Z-J). That is the channel model that we have nu+1 taps. It is an ISI channel, nu symbols of ISI. We do have noise as part of the system; however, we are not specifically doing anything to suppress it. It will be part in there and I will indicate to you where the noise actually shows up. Then at the receiver, so that is the channel is the channel C(Z) plus the noise.

At the receiver, we have to do the reverse operation of splitting the subchannels. So I call it FDM to TDM, the individual sub channels okay and at this point we also have to do the processing needed to remove the redundancy whatever redundancy we have introduced needs to be removed and leaving us with a vector S hat (N), which is also a Mx1 vector okay. So basically we have to link these 2 elements and we have to demonstrate that okay.

Now after going through the polyphase decomposition for the analysis, for the synthesis, I believe you will be comfortable if we drew the following figure okay. So I have the polyphase component matrix of the synthesis filters, which are represented by G(Z). There are M inputs going in. This is S0(N), S1(N), SM-1(N). There are M inputs. G itself is a matrix. So this would be Mx1. G itself is a NxM. M input N output, so this is where the redundancy has come in.

But we have not explicitly indicated where the redundancy is coming in. So we will just show that there are additional branches that are present. We will label them as X sub 0(N). This would be, next one would be X1(N). I want to just make sure, we get the complete picture. So we will just draw this one through X1(N). Think of this as XM-1(N) and then the last one as XN-1(N). So basically the redundancy has been inserted.

These signals pass through the channel and through the results from multirate signal processing, we have shown that this can actually be written or drawn in the following way as a pseudo-circulant matrix. A pseudo-circulant matrix of dimension NxN, so this is CPC(Z), pseudo-circulant matrix. Now I am ignoring the noise, ignoring the noise what comes out of it is the vector R and just like before, this is an Nx1 vector. So let me label them as R0(N), R1(N).

So somewhere the redundancy is still present RN-1(N) and then comes the removal of the redundancy. So here is the matrix which represents the synthesis or the analysis filters reducing it back down to M signals. So we again have the trapezoidal form and this is S(Z). This is a MxN matrix. The outputs are there are M outputs S0 hat, all the way to SM-1 hat. So this is S0 hat (N), S1 hat (N) and this is SM-1 hat of N okay.

So this is the overall framework that we have we have been able to simplify and obtain and I am sure that you are comfortable with this. So you can see that this is the point at which redundancy enters. So in redundancy is inserted and then on the other side at the receiver side redundancy is removed okay. So this now how do, we design these redundancies what is it that will give us the best performance that is what we are. This is here is the place where we remove the redundancy.

So basically input-output, we can relate by means of a, if I took into account G, CPC and S, we now have a input-output relationship between S0 through SM-1 to S0 hat to SM-1 hat this T(Z) is actually an MxM matrix which we have derived and each of these elements are obtained using this notation okay and we have interpreted what T(Z) should be, if you want to eliminate inter subcarrier, inter subchannel interference and if you want to also eliminate intra subcarrier interference as well okay.

So in this process, we did not explicitly mention it, but we have used type 2 polyphase decomposition, type 1 this is the synthesis filters. We have used type 1 polyphase component here okay. We have used type 1 in the synthesis, type 1 polyphase decomposition and we have used type 2 polyphase decomposition here again maybe just make a note, so that the overall notation everything sort of will fit in together okay.

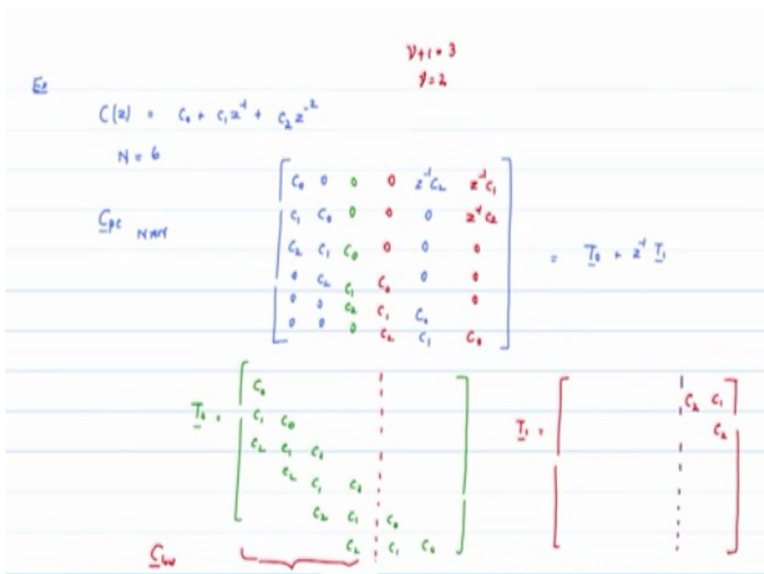
So how are these elements obtained, you can go through we have derived these results, so we would not spend more time on that okay. The matrix T(Z) you can interpret it as a polynomial of matrices and that is what tells us how to get rid of inter block and intra block, all of those elements, these are the observations that we have. Now one more comment which is useful at this point okay, this is what we had made in yesterday's class.

Let me just sort of put it in a box to the right. If the filters, so synthesis filters there are M of them F0(Z), F1(Z), FM-1(Z), if all of these have length less than or equal to N okay. This is M. M is the number of subchannels, N is the redundancy that you are going to introduce. If the filters are of length less than N, similarly the same thing applies for the analysis filters, which are H0(Z) HM-1(Z). If both of these satisfy this condition that the length than or equal to N, then this important result G(Z) the matrix and S(Z) are constant matrices.

There is no function of it, because when you do the polyphase decomposition, each of them has got only one entry. So therefore you get a constant matrices okay, so which means that you can represent it as G. This becomes G, this becomes S. Those are constant matrices, which can again it will be helpful once we look at it in today's class, but I thought it is a good point to mention to keep in mind okay.

So the process of simplification involved that we did the polyphase decomposition and also polyphase decomposition of the channel, which is what gave us the pseudo-circulant matrix.

**(Refer Slide Time: 12:28)**



And yesterday we did an example of the pseudo-circulant matrix and we also did some calculations on how to show that if I do zero padding that the inter block interference is removed, but what I would like to do is revisit that because there were a few questions after class maybe a little bit of clarification would be helpful okay. Now for this matrix, can you can you write for me what is T0.

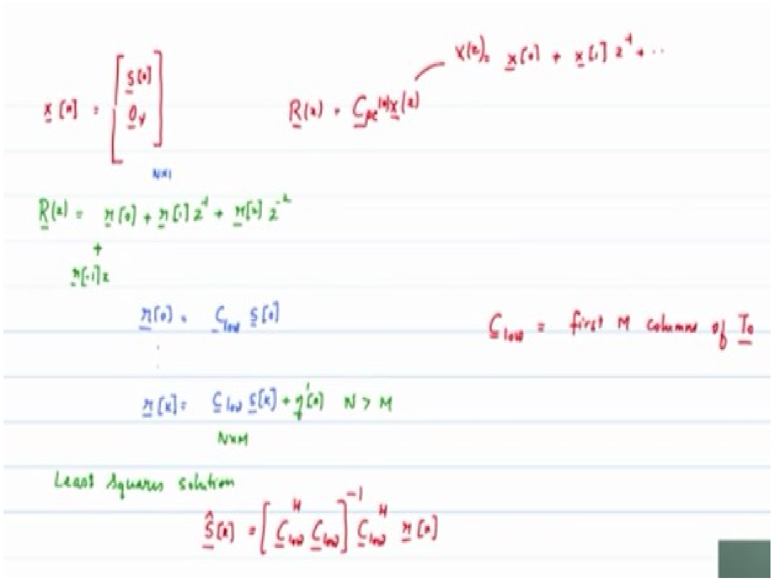
Write it in the form basically I would like to write it in the form T0+Z inverse * T1 okay. What are T0 and T1 okay. So T0 in this case would be a 5, 6x6 matrix C0, C1, C2, C0, C1, C2, C0, C1, C2, then C0, C1, and then C0 that would be the T0 matrix okay, plus then you would have, I do not need to put a plus here, Z inverse times T1, T1 of course would be whatever is left out. So T1 matrix would have mostly zeros. I will only write the non-zero entries, so you can fill in the others.

There would be a C2 here, a C1 here and a C2 here, that is only, the rest of it are all zeros okay. So if I padded with zeros, how many zeros would I have to pad basically this is a case where nu+1=3, so nu would be equal to 2. I would have to pad with 2 zeros if I padded my input vector with 2 zeros and then processed it through these matrices. Notice that I would have to I can draw a line, which separates the last 2 columns because they are going to be multiplied by zero valued entries and similarly the T0 and T1 will have that line.

Notice that whatever was there in T1 completely got killed because they were mostly zeros and there were some nonzero elements, but they were killed by the 0 padding and in this case you did have non-zero values, but those also got killed off, last 2 columns. So this portion of T0, the first M rows of T0 was what we refer to as C low. So what did we get as the dimensions you can verify that it is an NxM matrix, which is what we refer to as C low okay.

So some observations, if the earlier argument was maybe a little bit confusing, maybe this is one way of visualizing it or maybe even one more way of visualizing it.

**(Refer Slide Time: 15:44)**



The R vector we said is the S vector with the zeros appended okay. How many zeros, it is a nux1 column. So basically there are nu zeros. So this is a dimension Nx1 matrix okay. Now basically we have inserted these many zeros and the equation that we wrote down a few people were confused. Now if I take the Z transform of this, so then it becomes R(Z). Now what exactly am I doing? I am taking R0. This is R0, the vector transmitted at time N.

The vector time transmitted at time 1 with the Z inverse R 2. One second, I will be with you and if there is a R-1, that will come R-1 will come with a power of Z. Did I use X, X for what, here instead of R, okay, good? I will then change it. Thank you okay. So that is correct that is correct okay. R is actually at the output after you get the okay, so CPC*X=R right. That is basically X is the vector going into the pseudo-circulant matrix.

So the only way I can write this correctly it would be, this would have to be X (Z). This would have to be R(Z) okay. Let me just explain okay. So R(Z) is basically the vectors that are received at each time instant and if there was a R(-1) that would have power of Z okay. So you can think visualize that X(Z) this is nothing but X sub 0 which is obtained by a zero padding X sub 1 Z inverse, … and of course if you had X(-1) there would be a power of Z.

Now you can multiply the CPC(Z) by the way this is CPC(Z) with the Z transform, this is X(Z) and notice that because of the zero padding, the effect of the Z inverse terms in CPC actually get cancelled. So effectively what we will get into the expression when you compare the like powers of Z will be, you will get the following equations you will get R sub 0, if you were to, that is a constant term on the left hand side will be equal to C low*X0, no S0.

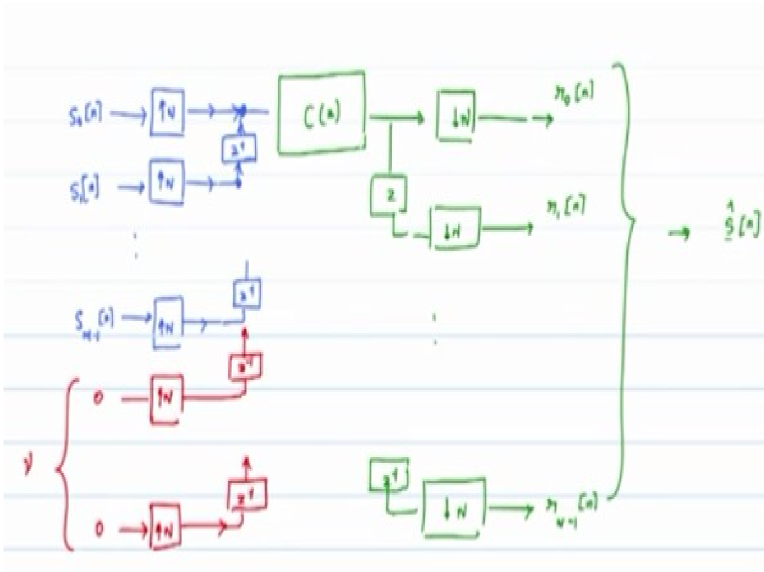
Because after you have removed the 0 padding, yes. So this would be the equation and at all time instances this is valid and if you have R sub K, the K-th time instant it is C low *SK. Dimension of C low as we mentioned is an NxM and we know that N>M because of redundancy. So this is actually an over determined system okay. So now maybe this is a good point to tell you where the noise comes in. Actually when you process it through this, this equation will actually have some noise term present okay.

So basically the received signal is S(K) passing through the channel plus some corrupted by noise. Now this is an over determined set of system in the presence with noisy observations. So the best thing that we can do is the least square solution. Least square solution, the best estimate of SK will be the least square solution, which is what we wrote down yesterday. The least square solution says that the best estimate for SK in the presence of uncorrelated Gaussian noise would be SK equal to the pseudo inverse, Moore-Penrose pseudo-inverse.

Yesterday we had written a dagger symbol, that is nothing but the Hermitian. So we will write it down with the Hermitian. C low Hermitian, C low is a constant matrix, so this is very, this is well defined inverse, inversion of that times C low Hermitian*R(N) okay. So this is a block by block decision process. There is no inter block interference and the way we have gotten rid of the inter block interference is primarily because of the zero insertion.

And so maybe what worth noting here is that C low corresponds to the first M columns of T0. All the other terms got killed because of the zero padding okay. So this is the framework that we that we derived yesterday. So I hope you are completely comfortable with this representation. So maybe it is good for us to do a block diagram that actually implements this. It may seem a little trivial, but I think this is a potential area where we could avoid any possible confusion.

**(Refer Slide Time: 21:57)**



So I take S0(N). I up sample by a factor of N and basically this is the synthesis part where we are combining the signals, where redundancy was going to be introduced. The second channel is S1(N) and likewise we have M-1 channels SM-1(N) up sampled by N. All of these will be combined through a delay chain okay. So this is the summing node. All of them are moving in the upward direction. Each of these are summing nodes and all arrows to the right okay.

This is also going through a delay chain okay. Now the zero padding part very important because we have only M-1 signals, but you are trying to multiplex N. So this is a zero valued input, which you can visualize as being up sampled by N, nothing is happening, but and it passes through the delay chain and then gets added to the other, basically the whole process. So like this there are many, many branches which are adding only zeros.

How many of them are there that is exactly the amount of zeros that you want to add, nu branches. Each of them up sampled by N passing through a delay and getting multiplexed with the rest of the signals okay. So this is the representation of the signals. Now this combined signal multiplex signal which now has the redundancy in the form of zero padding passing through the channel C(Z). Basically this will become the pseudo-circulant form.

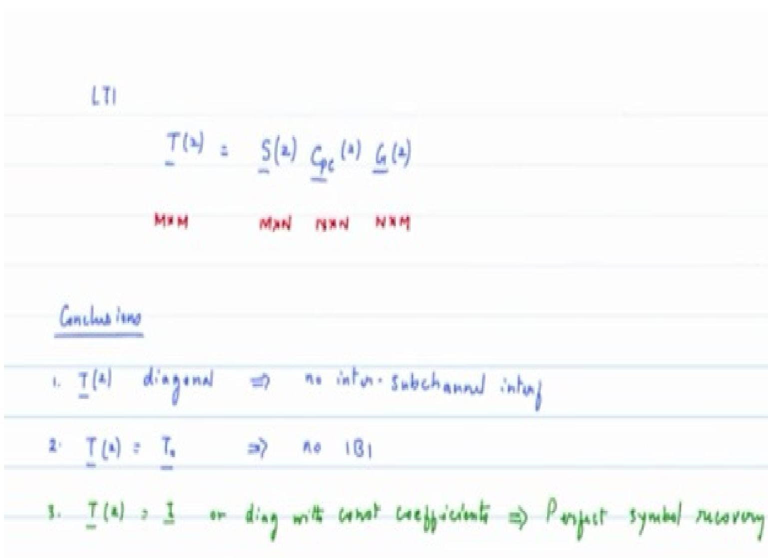
Once you take the up sampling by N and that you will get a pseudo circulant and then at the other end basically the down sampling by N and what comes out of this, we will call this as

R0(N) passing through an advanced operator down sampling by a factor of N. This is R1(N) all the way down to the last Z inverse down sampling by N. This is RN-1(N) okay. So this is the insertion of the redundancy and what came out of it so I throw away the last nu entries of R.

Is that what I should do, how do I get rid of the redundancy? How do I get rid of the redundancy? no not at this stage. I can only remove the redundancy; the reason the redundancy is to help me solve the output. So basically from here, from this set of outputs, I have to get S hat (N) and that is through the pseudo inverse and that pseudo inverse actually requires you to have this entire vector. So do not throw any of these samples away, not yet.

The removal or the exploiting of the redundancy and removing it is actually implicit in the pseudo inverse computations that we will show okay. So this is a good point for us to sort of summarize and say okay. Now we have a good handle on what it is that takes the ability to work with zero padded systems and how it all works together, why we are able to get a very clean solution and we are able to get.

**(Refer Slide Time: 26:07)**




Now the question that we asked yesterday's class was. By the way we can go back and expand on these in terms of the C(Z) okay. Now this is the channel model C(Z)=N=nu C(N)Z power –N. This is a symbol space channel model and the structure of the pseudo-circulant matrix again one more time for the full representation, we have we will not expand on this anymore. The pseudo inverse also has been described.

This was in yesterday's class and the last question that we asked is, is there any other way by which we could exploit redundancy and also take advantage and for that we said, yes, there is the processors, which will have the cyclic prefix.