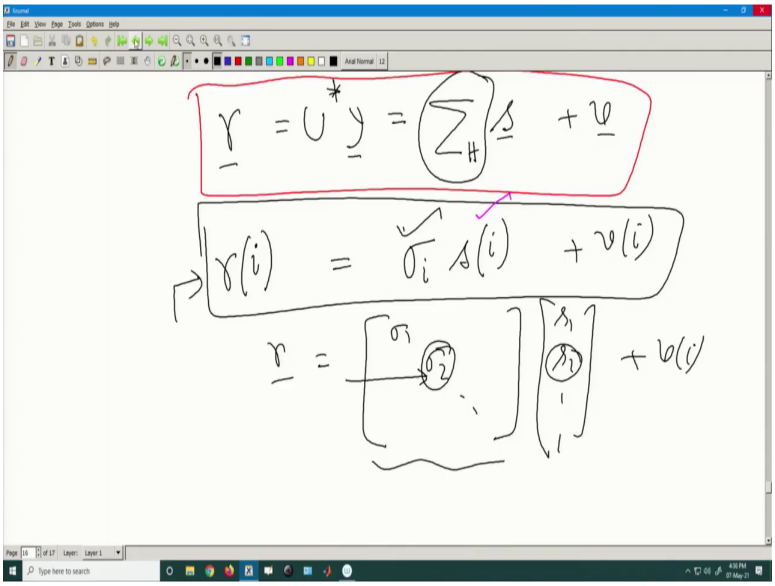
Signal Processing for mmWave Communication for 5G and Beyond
Prof. Amit Kumar Dutta
G.S. Sanyal School of Telecommunication
Indian Institute of Technology, Kharagpur

Module - 10
MIMO-OFDM beamforming
Lecture - 50
OFDM Data Model (Cont'd)

(Refer Slide Time: 00:32)

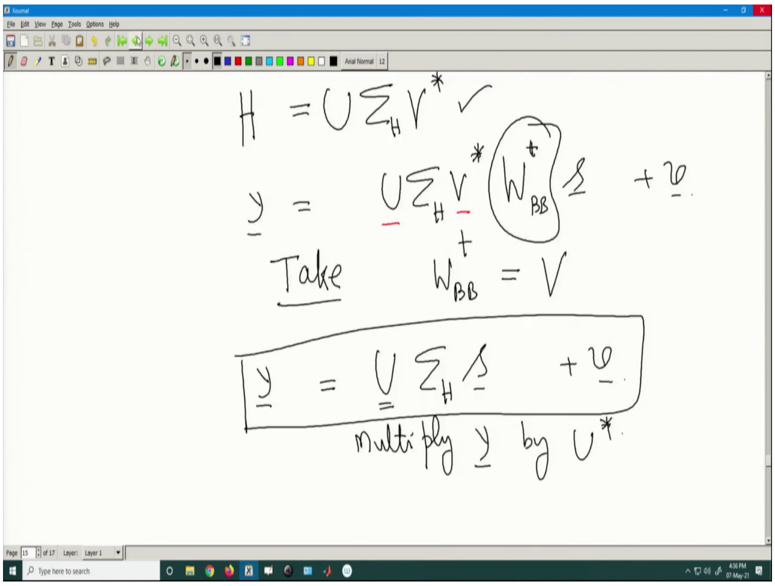
Welcome back to the millimetre Wave for 5G and Beyond. So, today we will be covering the lecture number things that will be covering are the following. So, you are starting the OFDM part. So, because of that OFDM, there is a motivation. And in the last class, we have given a simple motivation of how to remove the inversion part at the receiver side ok, and we have been successful in doing that.

(Refer Slide Time: 00:58)

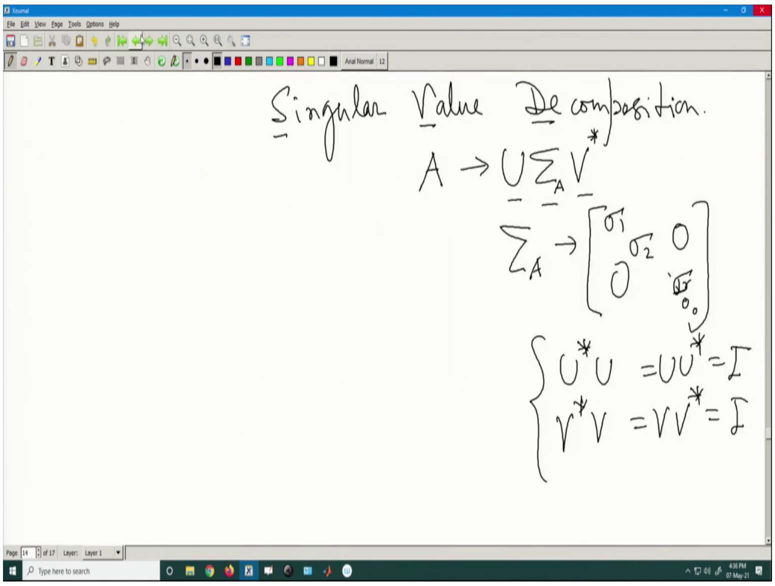


But let us go back to that equation again whatever we have drawn in the last class. And let us see is there any problem with this approach. Now, apparently does not have any problem, apparently. Why?

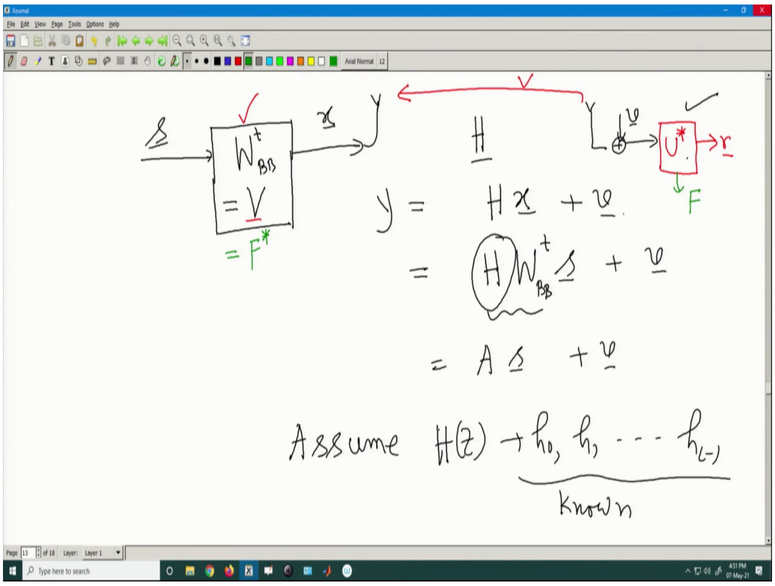
(Refer Slide Time: 01:14)



(Refer Slide Time: 01:15)



(Refer Slide Time: 01:16)



Because you see in the receiver there is no; there is no equalize, there is no inversion. So, this is just a U star. I can think of it as an equalizer. And this pre coder is a v that is also some sort of a matrix multiplication. Then what is the issue of this scheme which motivate us to go to some other concept or some other platform?

The whole problem, see I solved one problem. What is that issue? The issue was inversion at the ratio I solved that, but I invite another issue. What is the, what is the issue? You see that how do I do a singular value decomposition of this matrix channel, channel matrix because that that is also not a very easy task.

If the dimension of this channel matrix is becoming larger, getting a singular value decomposition is also probably I would say equally tough or equally comparable in terms of the computational complexity with respect to the inverse. Inversion is even tough to get it.

Singular value probably the efficient algorithm, you may get it slightly easier you may get it with slightly easier complexity, but still it is not reduced to a very simple one.

You still have this complexity, that means, the bottleneck you solve one problem but you come up with a slightly lesser issue, but still that is a significant problem called a singular value decomposition ok. And second problem, this is the first problem that singular value decomposition has to be done because then only because ultimately you have to know U matrix you have to know the V matrix right.

So, if you do not know U and V matrix, there is no point. So, know that part. There is a second issue. Second issue is that if you notice, if you look at the transmitter side, the transmitter this equalizer this is a pre coder W BB t equal to V, I am conveniently writing it but where will the transmitter know that how will it get V matrix?

Somebody has to tell it right. Either this particular channel matrix has to be you know broadcast to transmitter side and the receiver side both and then singular value decomposition or this V matrix can be you know can be developed at the r x side itself, you can do s v d. And then you feed it back to the transmitter.

So, what it essentially means? It means that this particular pre coder design mandates that you need to know the channel at the transmitter side and that is itself is a tricky task to have it. Because by the time you get your channel because what is easier to know the easier to know is that the channel should be at the receiver side. And it is very convenient to have it ok because you have a pilot you can estimate your channel at the receiver.

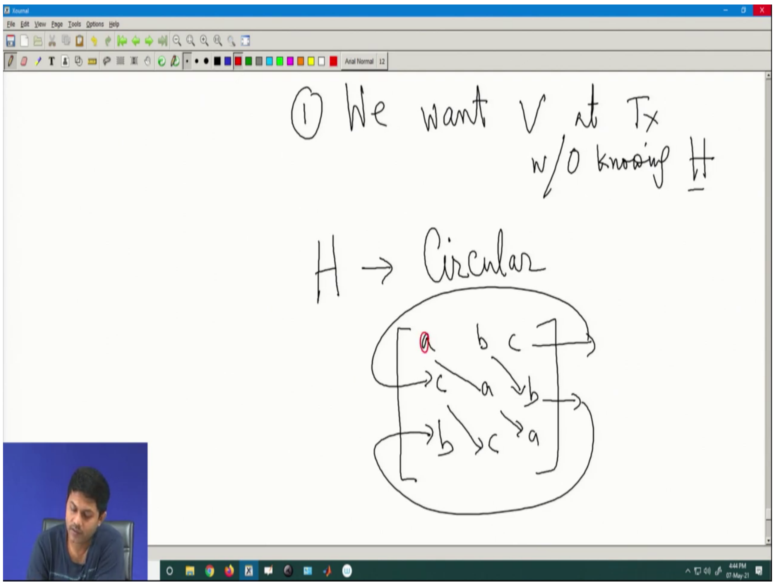
Now, to know the channel back to the transmitter either you have to transmit this whole channel back to the transmitter using you know the protocol or you have to you know or you have to re estimate your channel at the transmitter side assuming receiver is your transmitter. And if you assume some sort of a duality, probably that may or may not hold all the time because if you have slightly movement that may not work.

So, the feeding back this information to the transmitter itself is an overhead that is what I am trying to say, that means, this particular v matrix that means feed it back, you have to somehow feed it back from receiver to transmitter and that is a overhead, it is a huge overhead ok. But I want that because I have to; I have to determine I have to reduce my complexity at the receiver. But I am increasing complexity at my transmitter.

So, now, the question come can I have a scheme where I am achieving both? Meaning I am having this kind of receiver whatever is shown here where you know yeah where I have to your receiver is nothing but some, and at the end of the day your receiver will give you this kind of data model where it is just like a diagonal matrix data plus noise because then only I can avoid the inversion.

And at the same time, without, without, I making the point clear without any channel component feedback, I will design my equalizer pre coder, that means, that v matrix the equal pre coder matrix I know it upfront at the transmitter side without knowing my channel, nobody gives me any feedback. And question is how is it possible right?

(Refer Slide Time: 06:09)



So, the problem statement is the following. The problem statement is we want V at T x without knowing channel that mean I am developing a V which will exactly do what the other system has done the system has done. But I really do not know what my channel is, so that means, my V should be a channel independent component ok. How is it possible? Because if we say V is a SVD component of my H, then how can I develop a V without knowing a channel? It is a chicken and egg problem right.

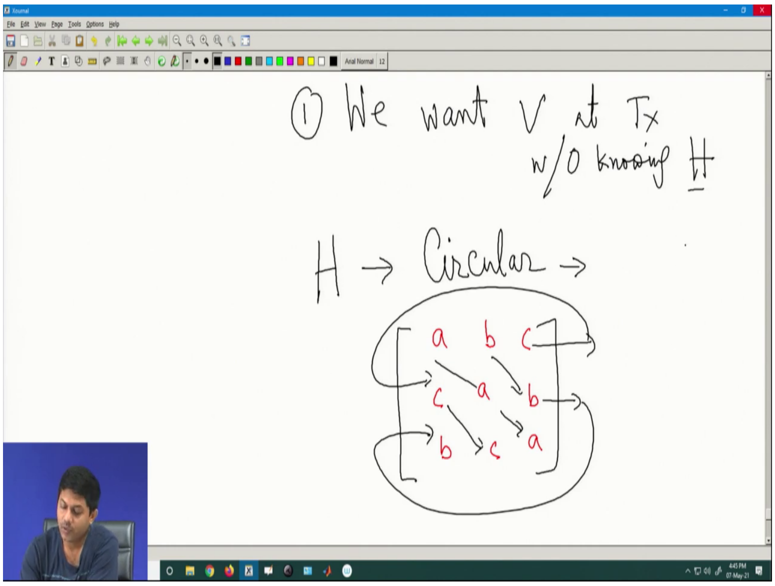
This is you have to first know H, then only you know V. Now, I am saying to no I do not want to know H, but I can develop a V. So, how can I guess even a V when my channel itself is not known to the transmitter side ok? And that is a very interesting problem statement. So, can I have that kind of system? Can I create a channel or can I have a channel where I know its V without even knowing the channel. You understand the problem ok.

And my dear friend that is a its possible, really its possible ok. How is it possible? If my channel, whatever channel I have saying that instead of a Toeplitz matrix or any other matrix it can be Toeplitz matrix if it is a SISO OFDM case, it can if it is just a MIMO case, then it can be just normal like a normal H meaning it is a matrix. Both are matrix, but it will not have any structure, because we are talking in the context of SISO. So, let us assume H only.

So, H instead of Toeplitz matrix if this matrix is a circular matrix, let us see what is mean by a circular matrix. Circular matrix is basically if I have a matrix say let us say I have a b c, there are three elements, so circular matrix meaning the row will be circularly shifted in the Toeplitz what is happening; what is happening in the Toeplitz matrix. In the Toeplitz matrix, the rows are not circularly shifted it is only right shifted or left shifted. We have seen a right shifted Toeplitz. You can also have a left shifted Toeplitz, but it is not a circular.

When it is a circular, it means that it is right shifted, but whatever is shifted out that comes back. So, which means that if a b c is the first row, the circular shift of a b c would be, a would be here, b would be here, and c which was going out it actually comes back here ok, so that is called a circular shift. And further if you do it, the a will be here, c will be here, and b that comes back here. I put a different colour, so that you can understand.

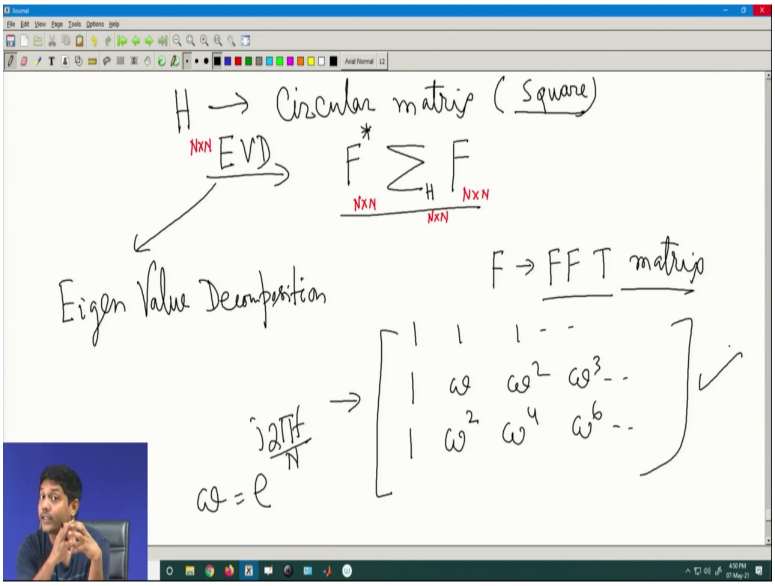
(Refer Slide Time: 09:36)



So, let us say I put a different colour. So, it is a b c, this should be a, this will be b, this will be c, this will be a, this will be c this will be b, so that is the matrix. This is circular matrix ok. Now, what so great about this circular matrix, why I am talking about all this? Because the, that is a very interesting property of a circular matrix.

The property of the circular matrix is that you can always do Eigen Value Decomposition of a circular matrix in this way. If H or H is H, you can think of any matrix with a circular matrix like this kind of structure.

(Refer Slide Time: 10:20)



So, it will be the property is that if H is a circular matrix, then I can break the H break meaning I can do some decomposition, but it is not a singular value decomposition rather it can be an Eigen Value Decomposition, but that is ok. We do not care whether its a singular value or an Eigen Value Decomposition as long as I create some decomposition I find you can have a QR decomposition also.

So, in this case, it is a Eigen Value Decomposition. So, I can say it is a Eigen Value Decomposition, Eigen Value Decomposition. If it is a Eigen Value Decomposition that is a; that is a matrix property it says that if any matrix is a circular matrix, of course it is a it has to a square matrix then only you can have the Eigen Value Decomposition here.

So, obviously, I am talking of this is also another point. If you look at the other problem, there channel matrix was not a circle was not a square matrix right because it was m cross m plus l.

But now I am putting a restriction what if H is a circular matrix and a square matrix? Now, we will see in a single antenna case how do you get an H which is circular and square, is it really possible or I am artificially creating that kind of thing, so that is the point of OFDM ok.

So, if these two properties are satisfied circular and square, then I can have an Eigen Value Decomposition an of H has let us call it still a sigma H, then it is not a Eigen it is not a singular value, but other Eigen Values. I can always break it like that. Now, what is F? F is FFT matrix, that means, if H is a square and circular matrix instead of a Toeplitz matrix, I can always have a Eigen Value Decomposition of this H as that this, three matrix I can break it F star, sigma H, F where F is a FFT matrix.

Now, you see FFT matrix. Is it dependent on H at all? Suppose, this H is a N cross N matrix ok, then this will also be N cross N, this will also be N cross N, and naturally this will also be N cross N, or an N cross N. So, it is a FFT matrix. And what is the structure of in FFT matrix?

Structure of FFT matrix is the following 1 1 1 1 omega omega square omega to the power cube, and all then 1 omega square omega to the power 4 omega to the power 6 and so on like that. What is omega? It is e to the power j 2 pi. And in this case, it will not be the case. So, it will be yeah. So, f will not be there 2 pi f yeah omega. So, it is a that is the point here ok.

Now, the question is what am I bringing it here ok? I am bringing a standard matrix structure. Now, this F is a FFT matrix. And as F is a standard matrix, it really does not depend on H. No matter what your H is as long as it is circular and square, this F is constant and the dimension is fixed. F is independent of your H.

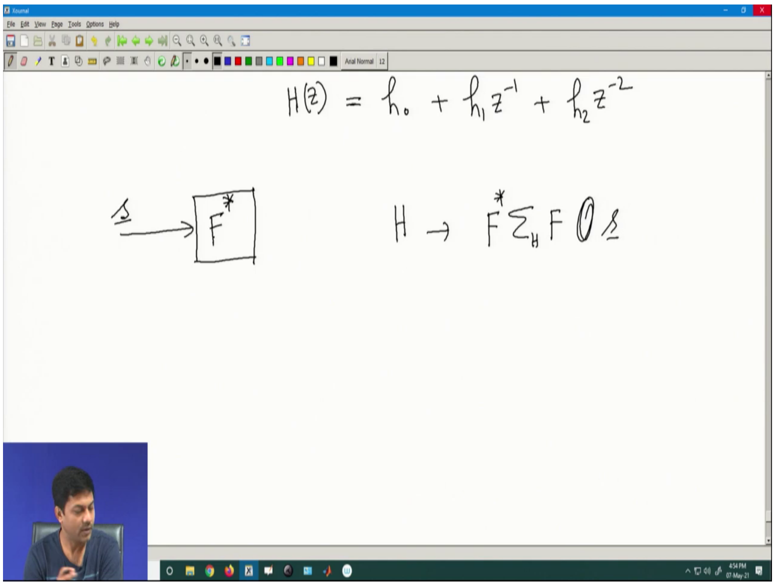
Now, assume, go back here, go back is this is a clear. Assume your H assume H is square and circular, then what will happen? What will be this W BB? It will be what should be the case? Then it will be F star inverse FFT. What is this U? This U will be F only. Now, you see in case this H is made square and circular at the transmitter side, I really do not have to know what is my channel. I just blindly put F star because F star would be the other side of the matrix.

And here also this for this U, I really do not have to know my channel I have to just, I do not have to do any SVD that is what I am trying to say without doing any SVD, I can get my U matrix. So, this F and F star or rather this equalizer and the pre coder are freely available now. Now, without any effort, whatever, why it is called a without any effort?

Because I really do not have to know my channel for my pre coder and my equalizer design. Now, to why do not, why I have to know channel? Yes, that is for final decoding you have to know the channel, because equalizer you because your eigen value still you have to know it right, but that is simpler to do it. I will also show you that is also very simple to know ok for a matrix. You really do not have to do any Eigen Value Decompositions big steps to find it out. It is very simple if it is a circular matrix.

Now, the question would be how do I really make this whole system a square and a circular channel ok? So, that is what the OFDM really comes into picture ok. So, which means if I really if I really do not know the channel, I can reconstruct my pre coder and equalizer. Now, let us see how can I make my channel circular matrix ok. So, let us take a very simple case, so that it will be easy for you to understand.

(Refer Slide Time: 17:21)

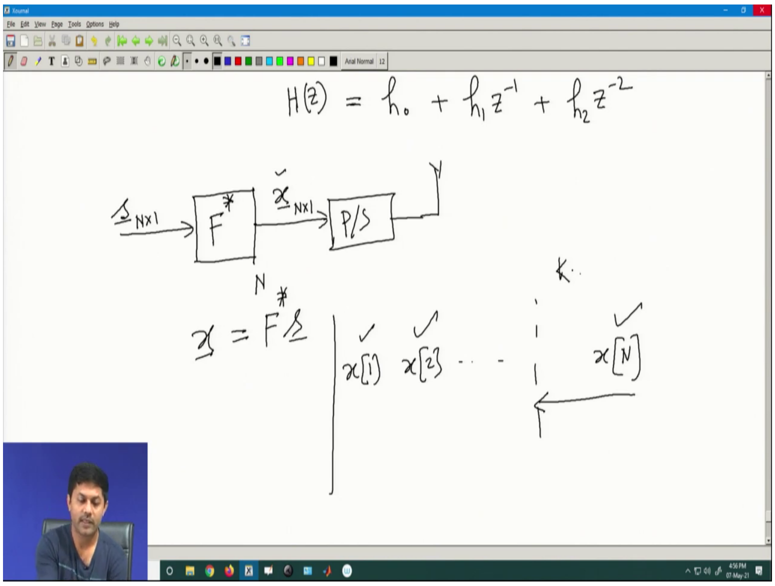


Let us take a simple case. Let us say my channel is just channels transfer function, I am talking, it is a let us take an example. Let us take I am taking let us say I am taking only three type channel, simple three type channel ok, wave channel ok. Then what I do is the following. I had my s vector ok.

Then what I said I multiply my pre coder with my F star matrix because I have decided somehow I am confident that I will make my H circular and square. So, I blindly put my equalizer blindly put my pre coder as F star that is what I have said ok. Why, because even if there is a channel I break it like a F sigma H and F star, sorry opposite F star.

I hope I have made the other part yeah, F star signal because this is the decomposition. And then because and here the s, s has to be coming with the equalizer part with a pre coder part, so obviously, the counterpart will be F star, so that is the reason I have given and obviously that ok. So, now, it is clear.

(Refer Slide Time: 19:05)



So, now let us say this is what I am transmitting. Instead of transmitting this vector like that what I do, let us assume that I am taking its length as N, it is an N length equalizer N length pre coder ok. So, it means that I am serially sending N number of data which will go through this particular matrix multiplication. So, this will be my N cross 1 length vector this will also be a N cross 1 length vector.

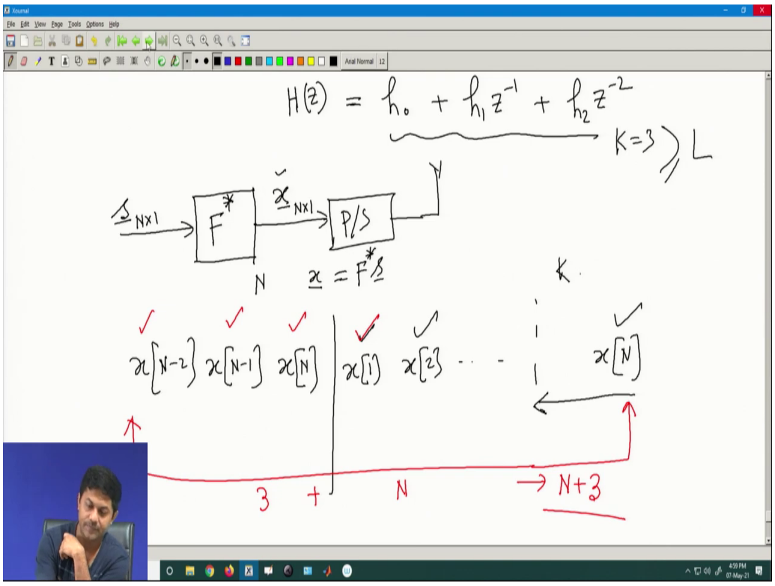
So, tentatively my x cap x bar is equal to s this is what my data model now. And this is what I am going to transmit, but hold on I am not transferring x directly. What I do is the following. So, which means that this is my data x of 1, x of 2, x of N, this is my data, this is the my data, I am sending it right. This is that N length data I am going to send it.

Now, instead of sending this directly that mean x 1 is the first data to be send, x 2 is the second data to be send and so and so on, it is a vector. But finally, I have to transmit through one single antenna. So, how do I, how do I transmit a vector? Vector I cannot transmit in a single time. So, what I do is that I am transmitting x 1 first, then x 2 first, and then x N first like that.

So, serially I will be transmitting. So, you can think of that there is there must be some sort of a parallel to serial converter here. Parallel to serial converter here, and that I will feed it to the antenna, but here I will have to do some more things. What I do? Now, what I am doing is that instead of you know sending this whole vector serially I mean serially I have to send it, but instead of sending like x 1, x 2 and so on so forth I piggyback the last K length vector and put it back into the front ok.

So, what does it mean? Let me just put this part slightly top, so that it will be easy to write it here.

(Refer Slide Time: 21:08)



So, x bar is equal to F star s bar ok. Now, what I do here is the following. The last key length part I piggyback ok. So, in this case let us assume as my channel length is 3, I take K is equal to 3. I will show that you should take K equal to 3, 4, 5 that is more than my channel length. So, K should be always greater than my channel length greater or equal to my channel length.

So, in this case as channel length is 3, so I am taking the last three and piggyback it here. So, what I am saying is this is my x N. What is my N? N can be whatever you think of. I am not getting into that direction how spectrum of the OFDM and those kind of thing will come into picture because that is more of a wireless communication topic, but this is more of a millimetre wave topics.

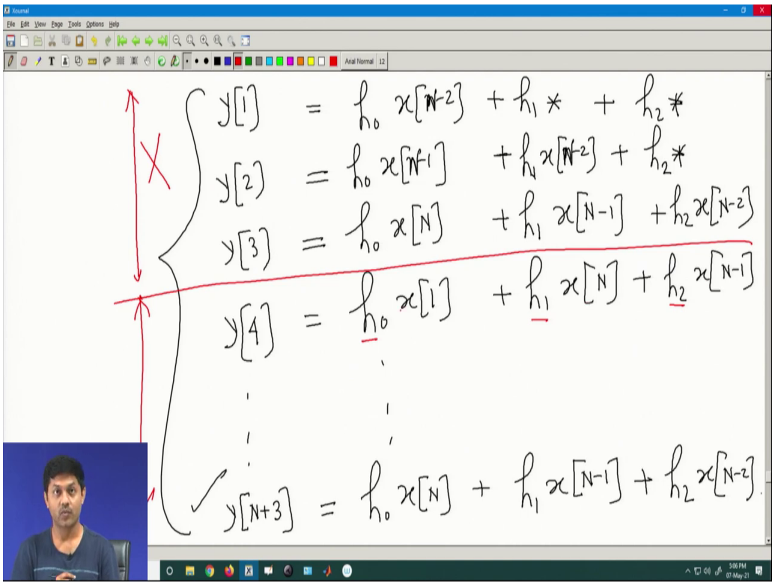
So, I assume that part I am not that part probably may be knowing it. And you can if not you can always you know do aspect do a simulation and see what are the spectrums and all so that is totally a different part. Now, what I am saying is this part this x N part, I am just piggy backing the stuff here. So, that means, this is the last three I am putting it here. So, it should be x N, x N minus 1, and x N minus 2. Now, this is my new data length ok. So, this is l N plus I add extra 3.

So, let us assume this is my N plus 3 length ok that is precisely what I am doing it here ok, so that is my length. So, instead of x what I am saying is that I piggyback this and now this is my first data, this is my second data, this is my third data, this is my fourth data and so and so forth, and last one sorry here there is a small mistake. It will be going up to here N plus 3 total data. Total data is like that ok.

Now, this is that this is the data I am sending first x N minus 2, x N minus 1 x N, then x 1, x 2, x 3 and blah blah blah all things are coming. So, that how many data I am sending? N plus 3 instead of N. And what is the N? N can be you take up your numbers. Now, if it is an FFT you know, it should be some 2 to the power m format like 16, 64, 128, 256. And how you determine that? That is again it is the separate topics, I am not getting into that ok.

So, let us assume N is fixed for you ok. Then what will happen? What are the interesting stuff that will come into picture here? The interesting stuff that will come into picture here is that, so your first data is you know x N minus 2, x N minus 1 and so and so forth right.

(Refer Slide Time: 24:28)



So, now, what I do at the receiver? Receiver, I am just I can just take the first data let us draw few, so that you can easily understand what I am saying is then there is a. This is what it is right. Now, what is y 1? y 1 is what was the channel? Channel was h 0, h 1, h 2 right. So, if I sense x N minus 2 that is my first data right, so that is my first data, then next come x N minus 1 and so and so forth ok.

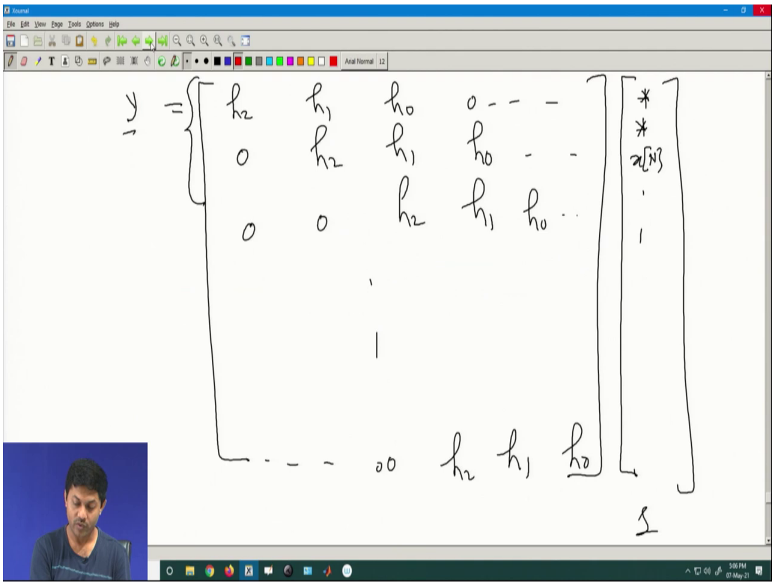
So, what is my data it will be h 0 x N minus 2 plus h 1 will be here. It is a garbage. Why? Because it is some data which is which was happening previously right. So, I call it some garbage. So, let us call it I do not know what is the best notation, but let us call it some cross it just notation, it is a previous data that mean before x N minus 1 sorry before x N minus 2 what was the previous data plus h 2 again that is some garbage. Clear?

Then I go to next one. What will happen? Here what will happen is that here plus h 2, this will be a garbage. What about the y 3? This will be x 1. Yeah, I should write it capital N ok, this will be 1, this is what it is. You can have similarly y 4 also. What will be the y 4? Here interesting point this will be x 1 because that is the first right. y 4 meaning at fourth position I am saying, so that is the first data, and subsequently the other three right.

So, this would be h 1 and so and so forth right. I spent some time to complete this part. And you can have an y N plus 3, so the last one, last one let us complete that part ok. So, this should be h 0, it should be x 1 plus h 1 x N minus 1 ok, plus h 2 x N minus 2 ok that is what it is, correct, ok, very good. This is what my last data.

Now, what is my next job? Stack it together and find out, ok fine. Let us do that ok. So, let us do that. So, let us say I stack the whole thing, stack meaning this whole data I stack the whole thing ok.

(Refer Slide Time: 28:49)



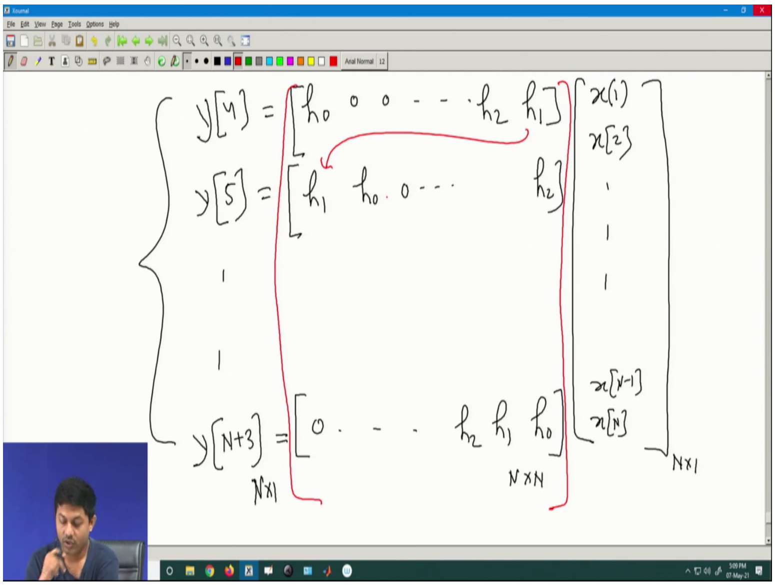
So, what will I get? I get h 2 h 1 h 0, then it will be all 0, then this will be we have seen it right, h 1 h 0 blah blah blah, finally, at some point of time h 0 h 1 h 2 0 0 like this. This is what I will Toeplitz matrix. There is nothing so great about it. So, what will happen? First two terms will be for s vector, first two term will be garbage. Garbage meaning from the previous frame some something like that.

Then what will I get? You can just construct it from here what is will be the x N, x N minus 1, x N minus 2 and so and so forth. So, it will be x N, and blah blah blah all will be coming into picture here. So, there is nothing great about it ok. Then I just do one small change here. What is the small change? Small change is that I discard the first three elements that mean when I stack it all of them together, I may find everything is like a Toeplitz matrix.

So, instead of taking this whole thing together like that I start taking from here onward ok, that mean, the first three I discard it, got it. The first three what I do I just discard it ok. Now, you think from y 4 to y N plus 3. So, now, instead of this you just take if I just take them together, it will be Toeplitz matrix.

Now, I am saying that hey I am actually not taking all of them together rather I just discard this part, this part, and I discard this part that mean when I receive it, but I do not consider it. I start considering only from y 4 that is my point ok. Now, you think this is 0, h 1, h 2, but who is that x N, x 1 minus 1 x 1 right.

(Refer Slide Time: 31:12)



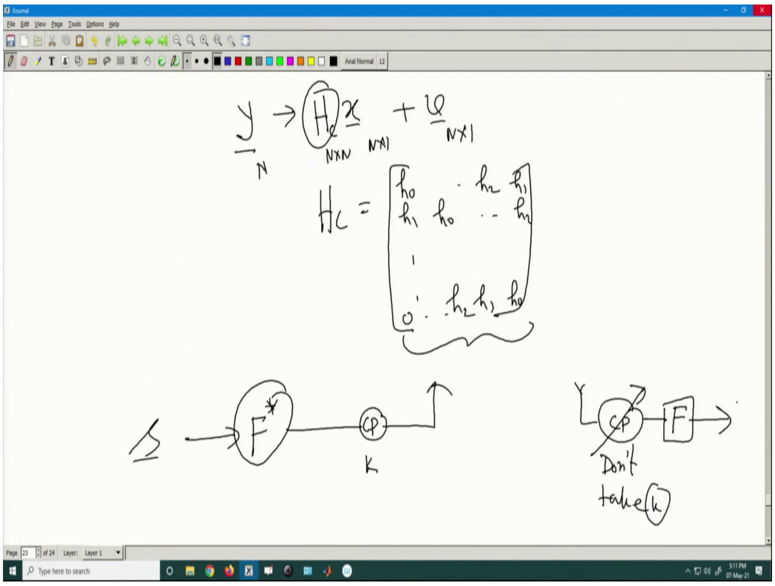
So, let us go back to your model instead of this now I am saying that you find out y 4. What will be the case? Let us say I stack it together like this x 1 x 2 x N x N minus 1. What will be the case? It is a h 0 x 1, but h 1 will be x 1 and this one alright. So, what does it mean? It means that with this, this will be h 0 right. But with x N who is there h 1, h 1 and this will be my h 2, rest of them will be all 0 ok.

Similarly, when I put it y 5, I am just doing some you know surgery. I am not doing any surgery, I am just doing rearranging. This is what it is coming right. What is y 5? You see this there. What is y 5? Obviously, this will be shifted back ok. So, you can check it, it will be h 1 h 0 h 2 altogether y N plus 3. You will see this will be h 0 h 1 h 2, rest will be all 0.

Now, see what is the nature of this channel? Now, what is the length here? Now, I put stack it together. This is what I have stacked. I am not done anything. I have just stacked them from y 4 to N plus 1. So, this is N cross 1. This will be N cross N because this itself is N cross 1 x 1 to x N right ok.

Now, look at the nature of this channel. Now, you can think of this is a matrix right. This is a matrix. Now, look at the nature of this matrix, actually this is circular matrix because h 0 h 2 h 1 look at the next row is like you shift it to the right side and put it back. So, this h 1 comes back here, it is a circular shifting.

(Refer Slide Time: 33:43)



So, now again so which means I got back my data, I got a data model which I call it y bar let us call it, and I got a channel I got x bar and v. And all their dimension is N, this is N cross N, this is N cross 1, this is N cross 1. But now this particular channel let us call it H c to ensure that the circular channel, this channel is basically a circular channel. In our earlier case, it was h 2 h 1, then it will be h 1 h 0 h 2 and finally all will be 0 h 0 h 1 h 2. This is how my channel is. You can generalize it of course ok.

So, now I am done. So, it is, what is my equivalent model? So, which means my this is my s bar and I put F star as a pre coder matrix, then I do some addition of my data, and then I send it back that extra that extra K data, and then that I call it CP. So, this particular one is called cyclic prefix because it is a prefix. So, it is called CP cyclic prefix.

And then at the receiver, what I do this cyclic prefix I just remove it. How do I remove it? Just whatever data you observe it, do not take that first initial data if the length is K here, do not take first K data that is it. Then what I should do, obviously, my F is there, this is what I will get something. We will see what next. So, but what I am trying to say is that I have made my channel a simple circular channel ok.

Then we will see what is the consequence of this case in the next classes. So, with this, I end the session here for the basic OFDM system. And we will see how the decoding has to be happening ok.

(Refer Slide Time: 35:40)

(Refer Slide Time: 35:45)

So, in conclusion, we have kind of completed the yeah and the reference will be same as what we have ok.

Thank you.