Estimation for Wireless Communication –MIMO/OFDM on Cellular and Sensor Networks

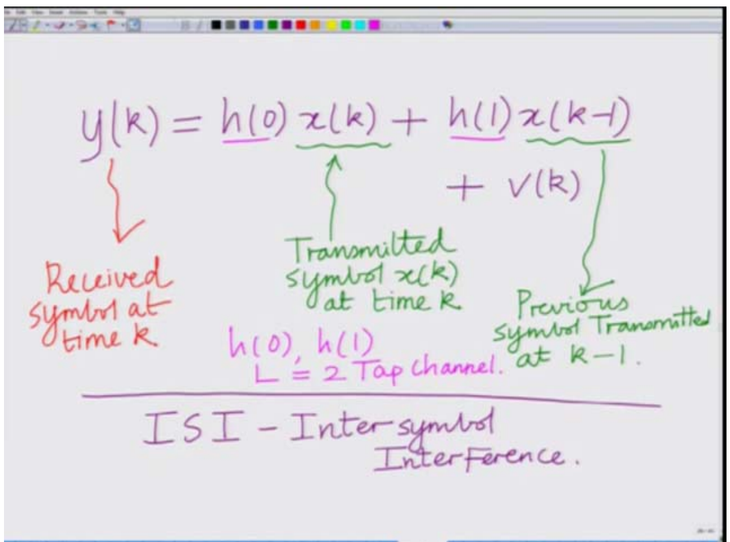
Professor A K Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture Number 22

Least-Squares based Zero Forcing Channel Equaliser

Hello, welcome to another module in this massive open online course on estimation for wireless communication. So we are looking at equalisation and we said equalisation is necessary for a wireless communication system or a wireless communication channel in which there is Inter Symbol Interference, right.

(Refer Slide Time: 00:45)



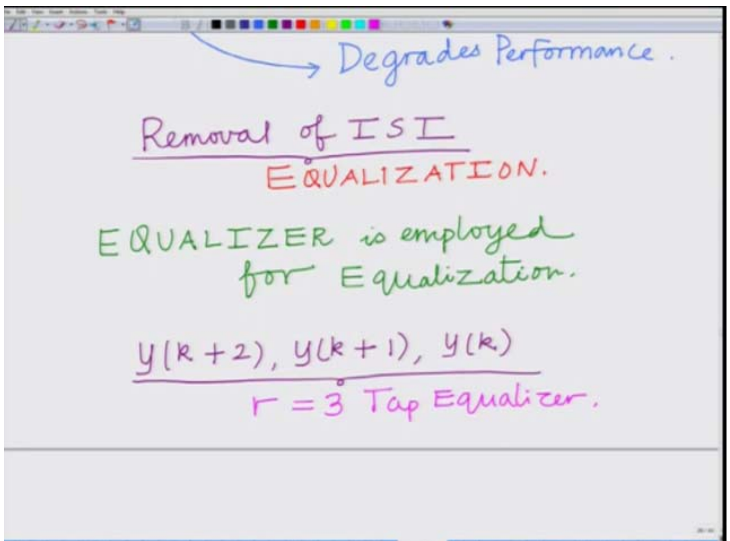
So just to refresh your memory, we looked at a system model in which the received symbol is y k equals h 0 times x k + h 1 times x k - 1 + well v k, v k is the noise. We said that this is the received symbol at time k, this is the received symbol at time k, this is the transmitted symbol x k at time k, transmitted symbol at time k and x k - 1 is the previous symbol transmitted at k - 1, previous symbol transmitted at time instant k - 1.

Therefore, y k depends not only on x k, but there is also, it depends also on x k - 1. There is interference in the detection of x k from x k - 1. That is x k - 1, which is the previous symbol is interfering with x k which is the current symbol, this is known as Inter Symbol Interference, correct. So this phenomenal is basically termed as ISI or Inter Symbol Interference.

And also here we said we are considering a 2 tap channel, this is h 0, h 1 are the taps of the channel, so h 0, h 1 basically we have said this is an L equal to 2 tap channel. That is, h 0, h 1 are known as the taps of the wireless channel and we have 2 taps that is h 0 and h 1, so the number of taps is L which is equal to 2.

For a general L, we have h 0, h 1 up to h L - 1 that is L tap that is the L Wireless channel for which the taps are denoted by the coefficients h 0, h 1 up to h L - 1. And we said we want to remove this Inter Symbol Interference. Inter Symbol Interference degrades the performance of communication therefore, we want to remove Inter Symbol Interference at the receiver, this is termed as Equalisation.

(Refer Slide Time: 03:45)



So we said, we would like to ISI this Inter Symbol Interference degrades or results in loss of performance, degrades the performance, ISI removal that is removal of ISI or Inter Symbol Interference and this is actually what is termed as your Equalisation. Removal of ISI Inter Symbol Interference is termed as Equalisation or it is also known as Channel Equalisation formally it is known as Channel Equalisation.

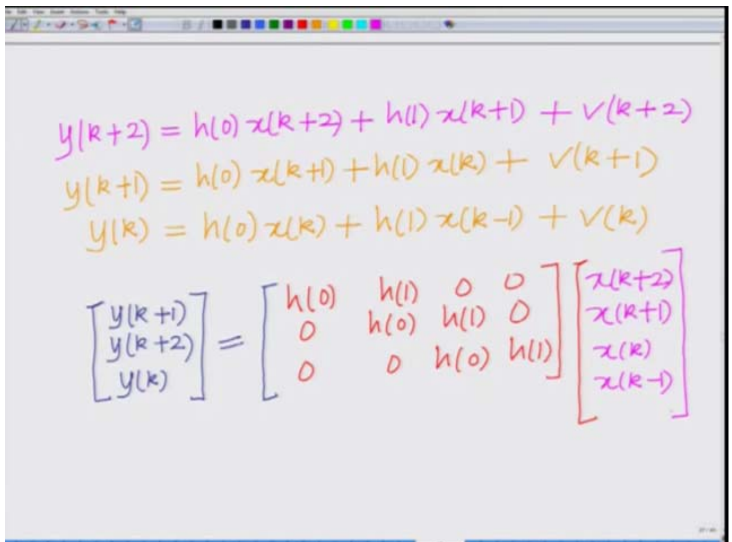
Also simply termed as Equalisation, all right. That is basically removal of the Inter Symbol Interference, right. And we said, for removal of the Inter Symbol Interference, we are going to use 3 received symbols. That is, we are going to build that is we are going to employ a 3 tap equaliser. So an equaliser is employed for equalisation, okay.

So equaliser so equaliser is basically the filter that is employed for, equaliser is the filter that is employed for equalisation. We said, we are going to employ received symbols y k + 2, y k

+ 1 and y k for equalisation that is, we are building a 3 tap that is, R equals 3 tap equaliser, yeah. So we said, we are going to build we are going to use 3 symbol that is y k + 2, y k + 1 and y k.

That is the number of symbols that is used for the purpose of equalisation is R equals 3 therefore, this is also known as 3 tap equaliser, all right. So now let us right the equations for these received symbols y k, y k + 1, y k + 2 and that also something we had written last time.

(Refer Slide Time: 06:28)

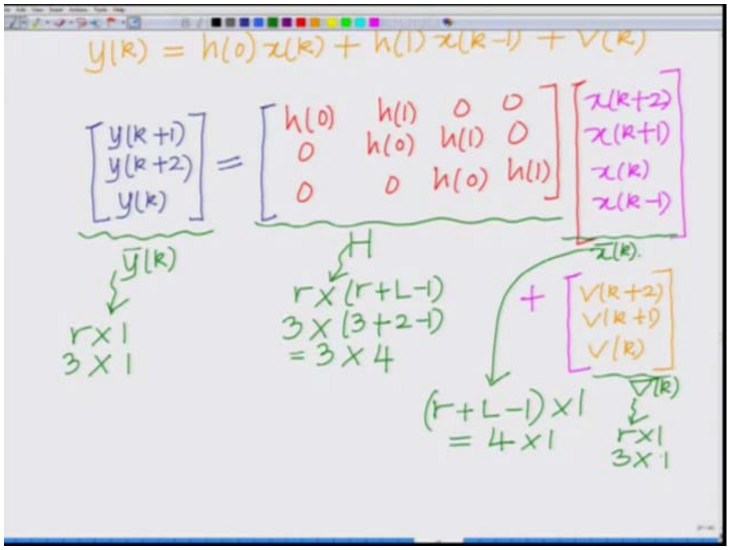


So y k equals h 0 x k + h 1 x k - 1 + v k, y k 0 1 equals h 0 x k + 1 + h 1 x k + v k + 1 and y k + 2 is h 0 x k + 2 + h 1 x k + 1 + v k + 2, yeah.

So basically now we return the equation for y k, y k + 1 and y k + 2. So we have written the equations for these 3 symbols. Now look at that y k depends on x k and x k - 1, so y k + 1 will depend basically on x k + 1 and there will be interference from x k which is the previous symbol to x k + 1. Similarly, y k + 2 will depend on the transmitted symbol x k + 2 and there will be interference from the previous symbol that is x k + 1 and so on.

So therefore, this is the model that we are employing that is the Inter Symbol Interference model. Now we are going to write these y k + 2, y k + 1, y k as a vector. So I am going to write these as a vector, so once I stack these as a vector, I have y k + 1, y k + 2, y k, which is equal to, now you can see.

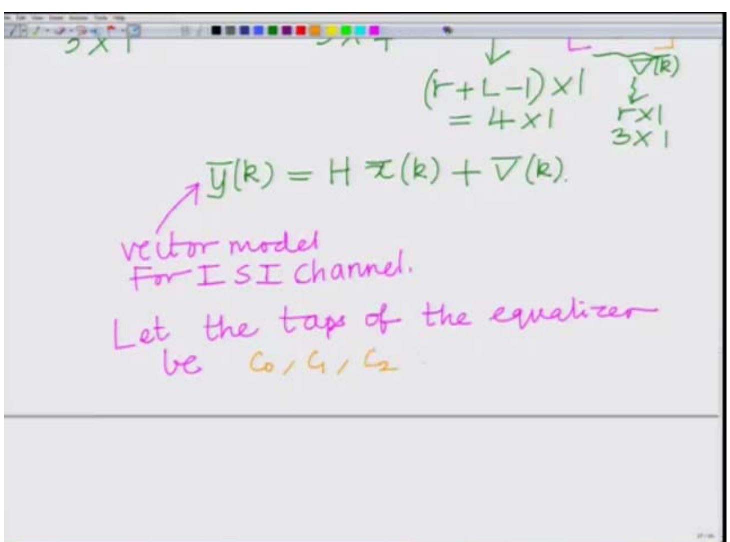
(Refer Slide Time: 09:25)



This can be written as the matrix h 0, h 1, 0, 0, 0 h 0, h 1 0, 0, 0, h 0, h 1, is the matrix h times the vector x k + 2, x k + 1, x k, x k - 1, yeah + the vector + the noise vector, which is v k 0 2, v k +1 and v k. And now we call this as the vector y bar of k, which has y k + 1, y k + 2, y k. This we call the matrix H, so y bar of k, this is your r cross 1 vector that is in this case r equals 3, so this is 3 cross 1.

H is basically r cross r + L - 1, r equals 3, so this is 3 cross 3 + L equals 2, 2 - 1, which is equal to 3 cross 4 matrix. This is x bar k, which is basically r + L - 1 cross 1, so this is r + L - 1 cross 1, which is equal to 4 cross 1 and this is your vector v bar of k which is r cross 1, that is basically a 3 cross 1 vector.

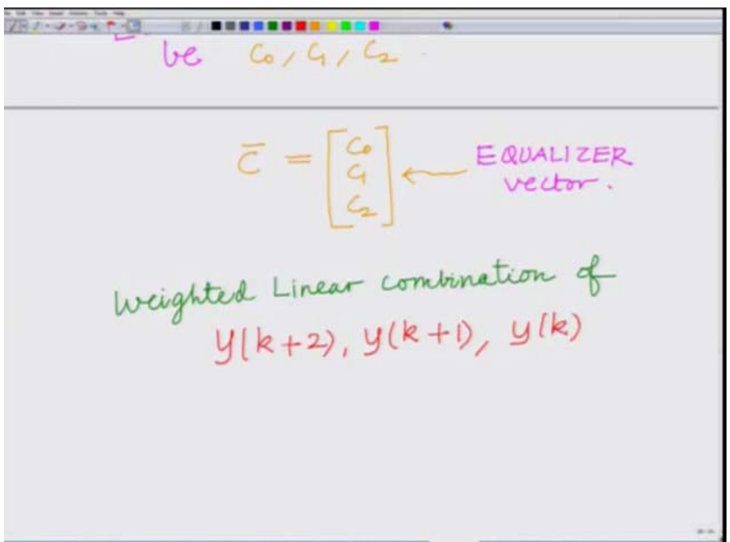
(Refer Slide Time: 11:02)



So I have this system model, now I have converted this into the vector form in which I have the system model y bar k equals the matrix H times x bar k + v bar k, okay. So let me write that, I have y bar k equals H times x bar k + v bar k, this is the vector model for the ISI channel. That is, communication channel with for ISI channel or ISI means basically, Inter Symbol Interference.

That is the communication channel with Inter Symbol Interference, this is the vector model for the communication channel with Inter Symbol Interference. Now we are we have to design the equaliser. Remember, we are designing a 3 tap equaliser that is R equal to 3, so let the taps of the equaliser be denoted by C 0, C 1, C 2. So let, let the taps of the equaliser be C 0, C 1, C 2, this can be denoted by the vector C bar equals C 0, C 1, C 2.

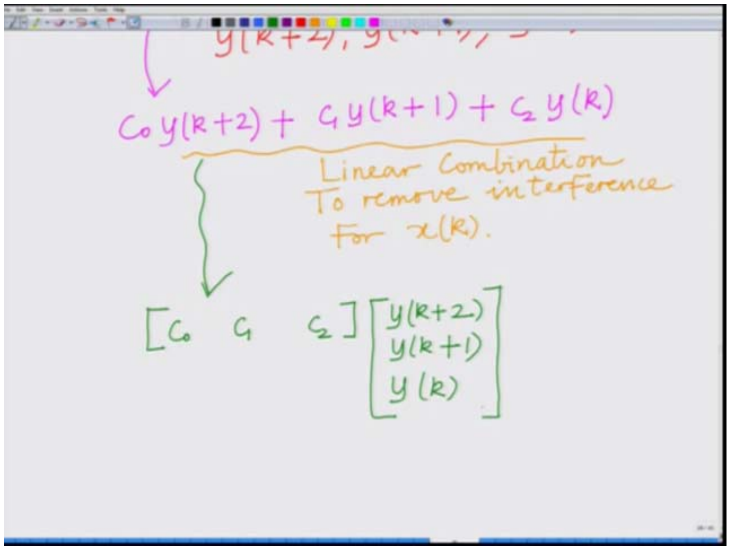
(Refer Slide Time: 12:28)



So this is also your equaliser vector, you can think of this as the equaliser vector. What is what the equaliser vector is? That is an r dimensional vector which has the taps of the equaliser that is C 0, C 1 up to C r. Since we are considering r equal to 3 tap equaliser, we have C 0, C 1 up to C 2 these are the 3 taps, okay. Now as we said last time what is the equaliser to?

Essentially, it takes a waited linear combination of the output symbols y k, y k + 1, y k + 2, all right. And basically with that weighted linear combination, we would like to eliminate the Inter Symbol Interference on x k. And how do we do that? So we take a weighted linear combination of y k + 2, y k + 1 and y k. And this we do as basically the weighted linear combination is, this is given as…

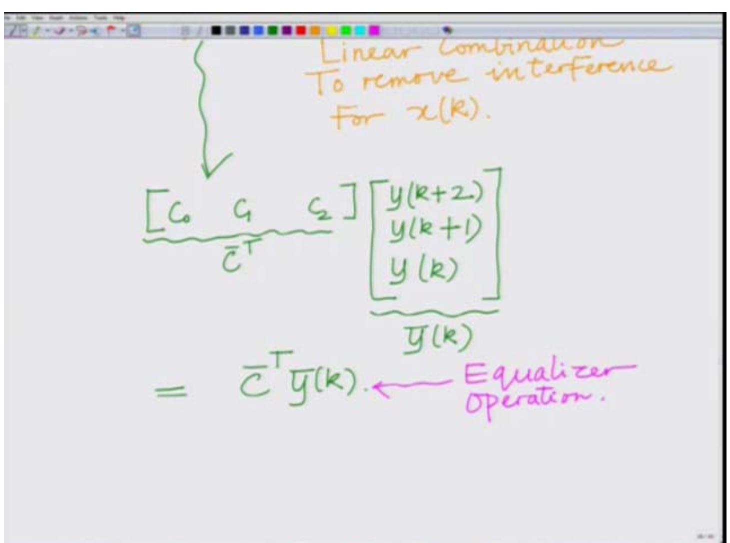
(Refer Slide Time: 13:55)



C 0 times y k + 2, C 1 times y k + 1, C 2 times y k, so this is the weighted linear combination that we are performing to remove the Inter Symbol Interference on x k, okay. So we are performing this linear combination, let us put it that way, to remove the interference, we are performing this weighted linear combination to remove the interference for x k.

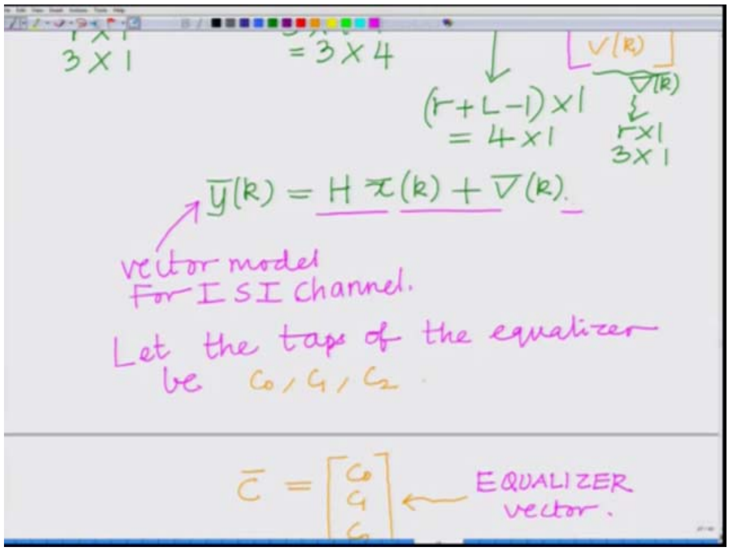
And therefore, now I can write this as look at this, I can write this linear combination as a row vector C 0, C 1, C 2 times the column vector y k + 2, y k + 1 times y k that is the row vector C 0, C 1, C 2 times the column vector y k + 2, y k + 1, y k. And now look at this, this row vector is nothing but C bar transpose that is, transpose of the equaliser vector times y bar of k. So this is basically c bar transpose y bar k.

(Refer Slide Time: 15:50)



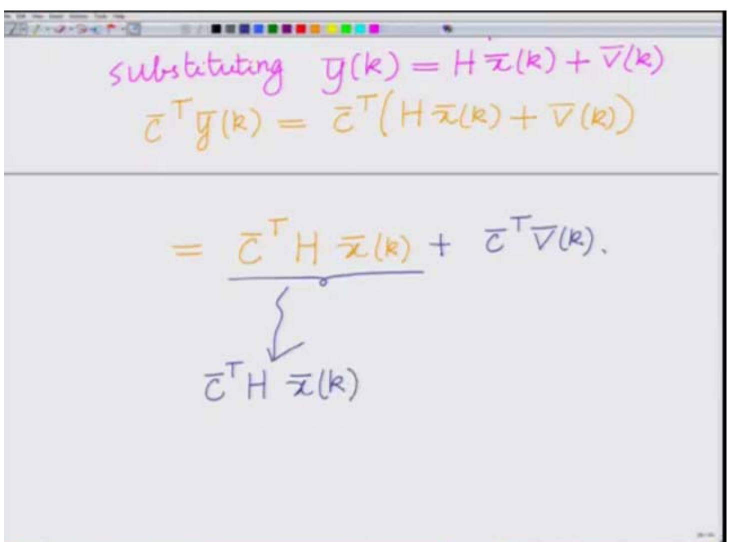
This is the equaliser operation. So the operation of the equaliser basically, what does it equaliser doing, it is performing a weighted linear combination of the symbols y k + 2, y k + 1, y k, all right. And that is C 0 times y k + 2 + C 1 times y k + 1 + C 2 times y k, which can be expressed as the equaliser vector C bar transpose that is the row vector C 0, C 1, C 2 times the column vector y bar k, which is y k + 2, y k + 1, y k.

(Refer Slide Time: 16:47)



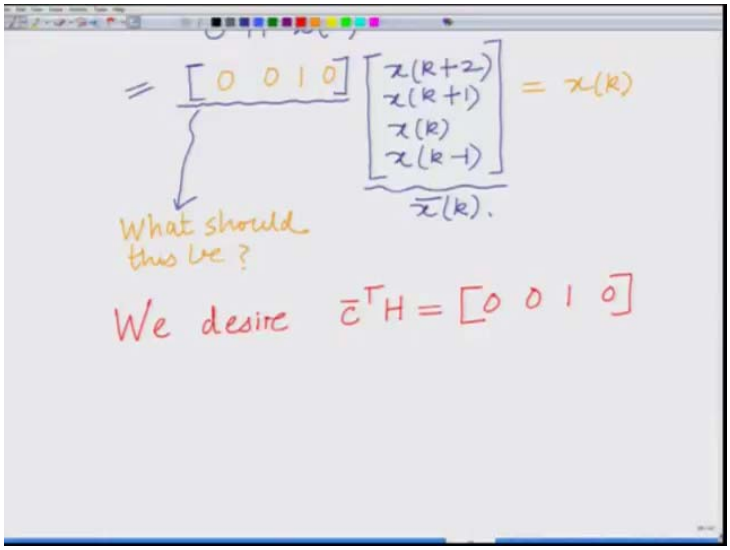
Now if I substitute the expression for y bar k from above. Look at this, I have already derived the expression for y bar of k, y bar of k equals H x bar k + v bar of k.

(Refer Slide Time: 16:52)



So now substituting y bar of k, substituting y bar of k equals h x bar k + v bar k, I have well I have C bar transpose y bar of k equals C bar transpose H x bar of k + v bar of k. Now removing the brackets I have C bar transpose H x bar of k + C bar transpose v bar of k.

(Refer Slide Time: 18:03)



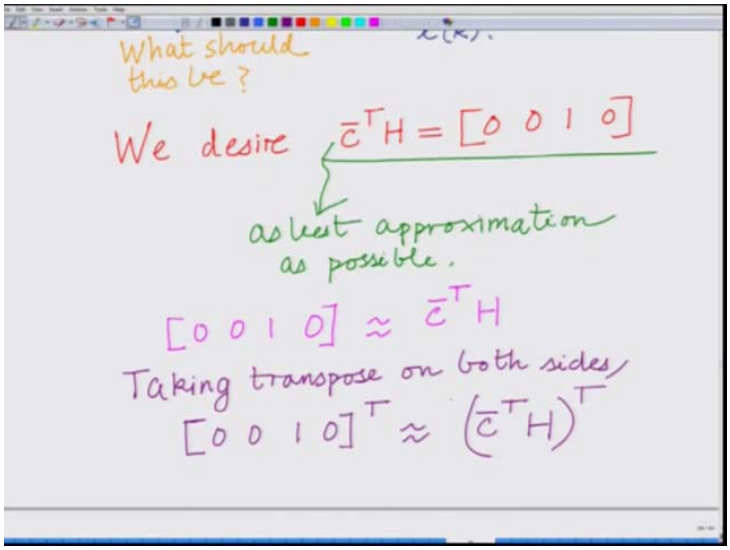
Now let us look at this quantity; C bar transpose H x bar of k, which is basically equal to some row vector times x bar of k, which x of k + 2, x of k +1, x of k, x of k - 1, this is basically or x bar of k. So I can write C bar transpose H x bar of k as some row vector times the column vector x bar k. Now the question is, what should this row vector be, what should this row vector C bar transpose H be?

So if I look at this row vector, what should this and I ask the question, what should, what should this row vector be. Ideally, this row vector should be 0, 0, 1, 0, so that when I multiply this row vector by x bar of k, I get 0 times x k + 2 + 0 times x k + 1 + 1 time x k + 0 times x k - 1, which is x of k. So we actually desire, let me write this clearly.

And this is an important note, so it is important to understand this point, see we have C bar transpose H times x bar k. Now how should we design the equaliser C bar. Ideally, we would like C bar transpose H to be the row vector 0, 0, 1, 0, so that when it is multiplied by x bar of k, the interference from x k + 2, x k +1 and x k - 1 is removed and what we are left with is x k.

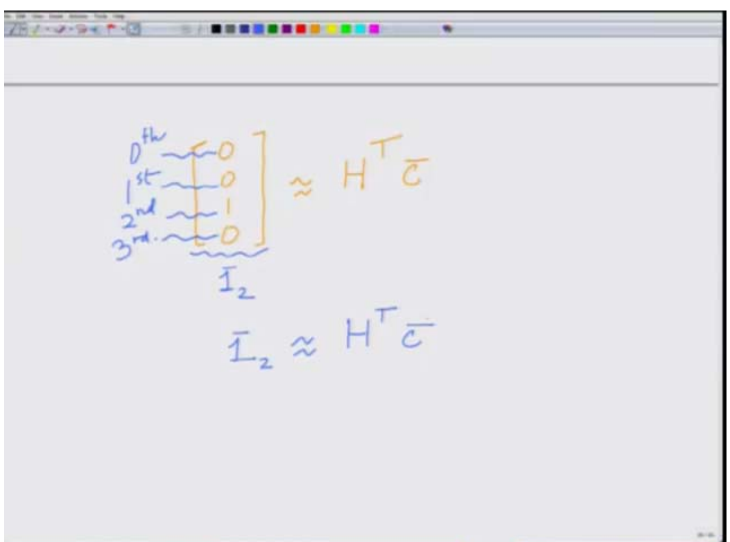
So ideally we desire C bar transpose H equals 0, 0, 1, 0. Now we might not get that exactly, that is C bar transpose H might not exactly be 0, 0, 1, 0, but if not exactly, then we would like to at least approximately as closely as possible, that is the idea. So C bar transpose H if possible should exactly be 0, 0, 1, 0, if not possible exactly then at least to the best possible approximation it should be close to the vector 0, 0, 1, 0.

(Refer Slide Time: 21:05)



And that is the point between equaliser that is we want C bar transpose H equals this thing or as best approximation that is, C bar transpose H should approximate 0, 0, 1, 0 as best as possible. Therefore, what we desire is 0, 0, 1, 0 should be approximately equal to C bar transpose H. Now taking the transpose on both sides that is, taking transpose on both sides, we have 0, 0, 1, 0 transpose equal to C bar transpose H transpose.

(Refer Slide Time: 22:2)

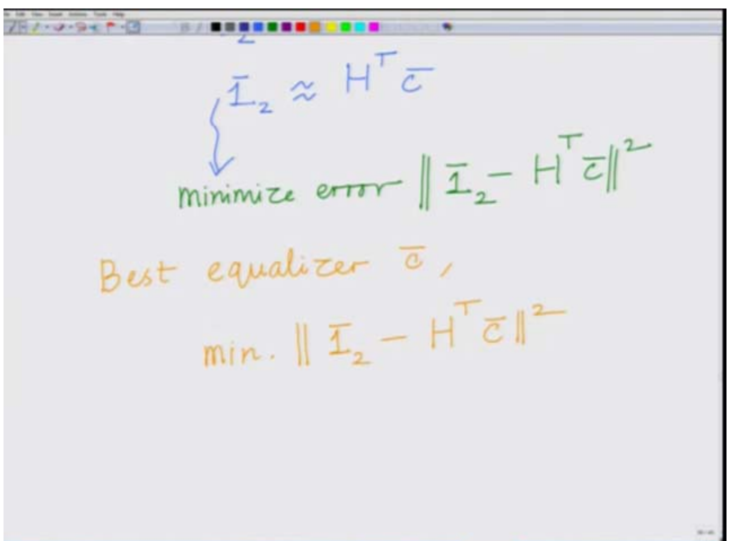


Or basically approximately should be approximately equal to this quantity. Now therefore, if you take the transpose of course, the row vector will become the column vector. Therefore, I can have 0, 0, 1, 0 should be equal to well or should be approximately equal to C transpose H transpose H transpose times C bar, yeah. So what we should have is this column vector 0, 0, 1, 0 should be very close to H transpose times C bar.

H transpose times C bar should be as close as possible to 0, 0, 1, 0 and naturally we have already seen a framework to this to do this. That is to minimise the squared error that is, when we want the approximation to be very good, what we would like to do is to basically decrease the error as much as possible, which means basically we have to consider that vector C bar which gives the least norm of the error and that leads us to the least squares solution.

So what we would like to do in fact, is basically we would like to, if I denote this vector by 1 bar of 2 or if we would like to denote this vector appropriately by let us say this notation 1 bar of 2 basically, what this is, look at this, this is has a 1 in the second position. If you call this as the 0th position, this as the 1st position, this is the 2nd position and this as the 3rd position, this has a 1 in the 2nd position and 0s elsewhere.

(Refer Slide Time: 24:16)

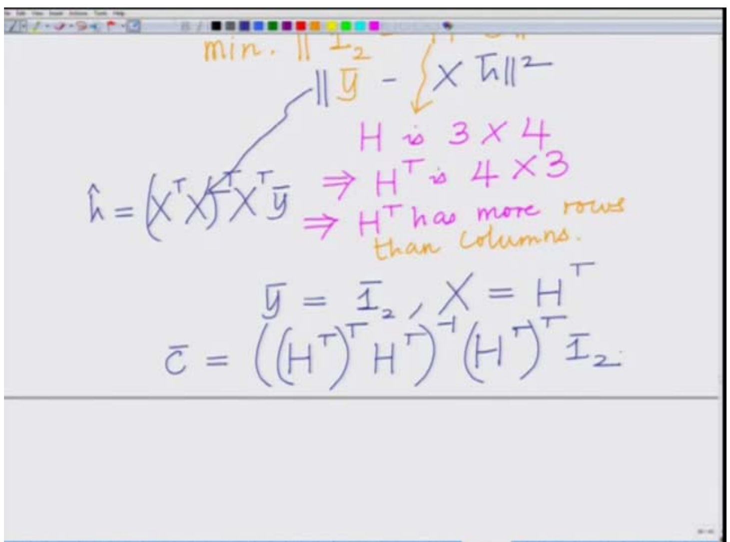


So this is, then denoting with the vector 1 bar 2, so 1 bar 2 must be approximately equal to H transpose C bar, which means I have to minimise the error 1 bar 2 - H transpose C bar. Of course, 1 bar 2 is a vector H transpose C bar is a vector, so when I say minimise the error, since this is a vector I have to look at minimising the norm of the error or the squared norm of the error.

It is the same thing as minimising the norm of the error, so basically that now gives rise to the least squares cost function and that is it. So equaliser C bar, so best equaliser C bar what does it do? It minimises that is, it is the minimum of 1 bar 2 - H transpose C bar whole square. That is the best equaliser C bar is such that H transpose C bar is as close to this vector 0, 0, 1, 0 as possible.

So we are basically considering that C bar which has the least error um that is which minimise the error 1 bar 2 - H transpose C bar that is basically the least squares solution. And now you can see, this is indeed the least squares solution because look at this H, size of H, H is 3 cross 4 implies H transpose is 4 cross 3 matrix implies H transpose has more rows than columns.

(Refer Slide Time: 26:18)

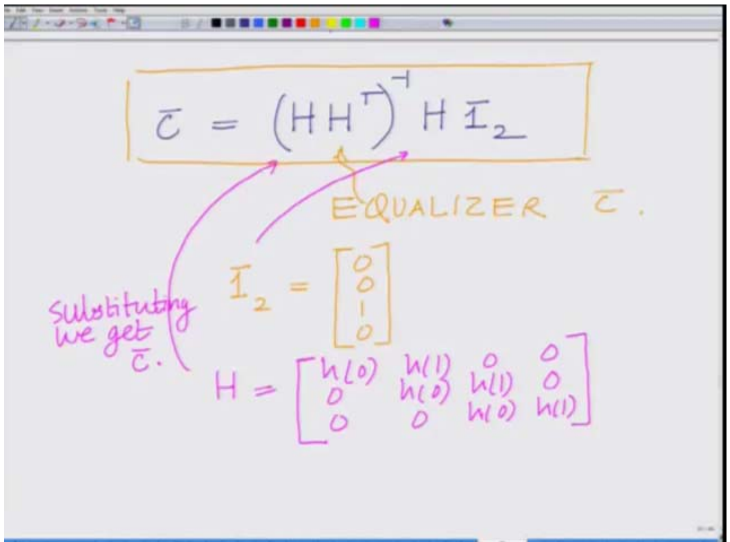


H transpose, the matrix H transpose has more rows than columns. Therefore, H transpose is a tall matrix and this is exactly the kind of problem for which we have the least squares solution, right. So we have remember previously we had y equals x times h bar in which the pilot matrix h had more rows than columns that is more observations than the antennas and year we have something similar except in a slightly different context.

That is in the context of equalisation. For instance, if I have to draw a parallel, not an exact parallel but just simply in the perspective from the point of view of least squares, I had y - X h bar norm square and the solution was basically, your h hat equals X transpose X inverse X transpose y bar, that is the least squares solution. Now all I am doing is replacing my y bar by 1 bar 2.

X is equal to, so my y bar, so in this example for equalisation y bar is basically your vector 1 bar 2, your vector x is nothing but the matrix H transpose and I am trying to estimate C bar which plays the role of H bar here. So my so my equaliser C bar is basically H transpose transpose times H transpose inverse times H transpose transpose 1 bar 2 that is the least squares solution.

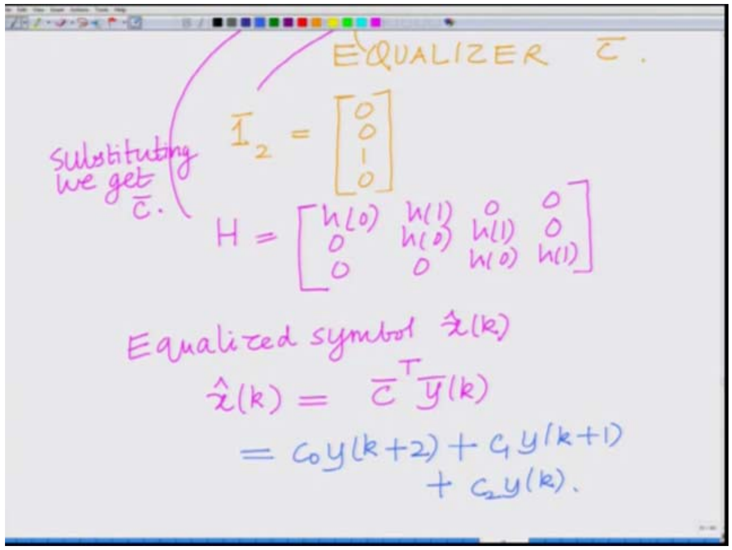
(Refer Slide Time: 28:20)



And that C bar is equal to H H transpose, C bar is equal to H H transpose H H transpose inverse H times 1 bar of 2. This is the solution of the equaliser. Now you can see, this is the solution of the, this is basically your equaliser that is C equaliser C bar. This is basically the equaliser C bar, which is H H transpose inverse H transpose H times the vector 1 bar 2, this is basically the equaliser and we have already defined the matrix H earlier, right.

And 1 bar 2 is also the vector that we have defined before, but anyway just to again confirm that just to again remind you 1 bar 2 is the vector 0, 0, 1, 0 and if you remember, H is this matrix which is formed from the channel coefficients h 0, h 1, 0, 0, 0, h 0, h 1, 0, 0, 0 and this and substituting we get the equaliser which is C bar.

(Refer Slide Time: 30:25)



And once you have the equaliser, the resultant equalised the symbol x hat of k remember, x hat of k that is equalised symbol x hat of k equals C bar transpose y bar of k, which is basically now you can see that is equal to C 0 times y k + 2 + C 1 times y k + 1 + C 2 times y k, this is the equaliser. So this is, how you equalise, that is you design C 0 the taps C 0, C 1, C 2 as H H transpose inverse times H into this vector 1 bar of 2.

Now once you get the taps C 0, C 1, C 2, you use them as waste to linearly combine y k + 2, y k + 1, y k as C 0 times C 0 times y k + 2 + C 1 times y k + 1 + C 2 times y k to get an estimate of the symbol x hat of k and basically that explains equaliser. So what we have done is interestingly basically, we have looked at this model for Inter Symbol Interference of a simple L equal 2 tap Wireless Channel, right.

We have modelled it, we have extracted the vector model for this Inter Symbol Interference channel and then we have motivated this equaliser design to remove Inter Symbol Interference at the receiver and we have also demonstrated that this equaliser design can be demonstrated, can be reduced to a least squares problem.

That is basically, we have demonstrated that it is the best equaliser C bar minimises, minimises the error 1 bar 2 - H transpose C bar that is minimises the squared norm of the error and this equaliser C bar is found as the least squares solution. And of course now we have considered a simple channel with L equal to 2 taps.

Naturally, you can extend it to more general channel with L taps and of course a more general equaliser with r taps. Here we have considered r equal to 3 but of course you can again extend it um using this based on this example, you can in a straightforward way extend it to a scenario with any arbitrary number of taps, all right.

So this basically comprehensively explains equaliser design, this is also known as the design of a 0 forcing equaliser. That is probably worth mentioning, this explains the design of a 0 forcing equaliser and we will look at an example um in the next lecture to understand this better, thank you.