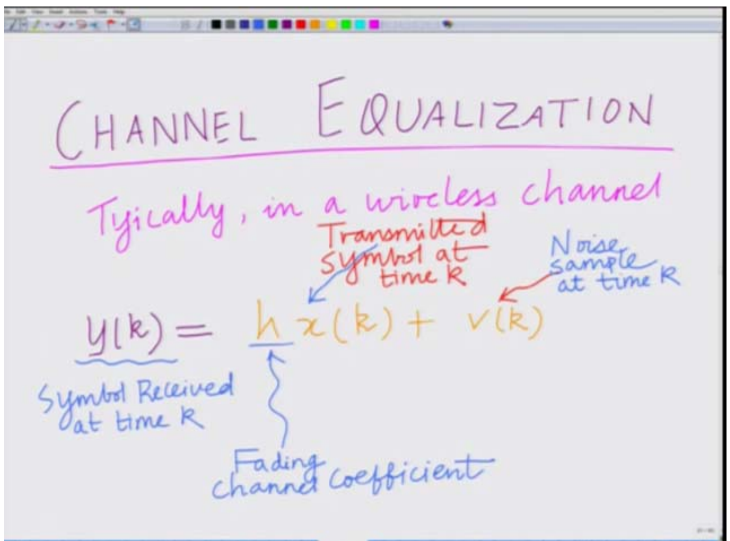
Estimation for Wireless Communication –MIMO/OFDM on Cellular and Sensor Networks
Professor A K Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Lecture Number 21
Channel Equalisation and Enter Symbol Interference (ISI) Model

Hello, welcome to another module in this massive open online course on estimation for Wireless Communications, all right. So far in the previous model, we have looked at channel estimation in particular, MIMO channel estimation. In this model, we will start looking at another aspect of wireless communication that is, Channel Equalisation that is equalisation of Wireless channels.

(Refer Slide Time: 00:37)

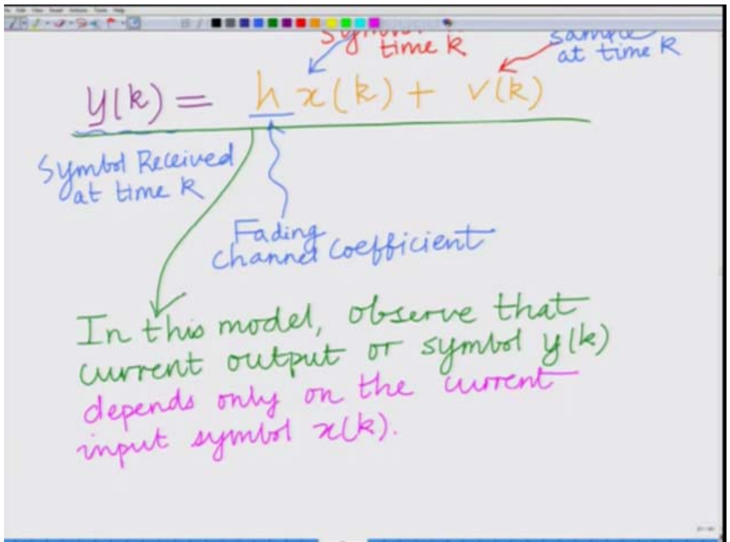


So let us start looking at Channel for what we termed as Channel equalisation. And this can be motivated as follows Channel equalisation. Typically, in a wireless channel as we have already seen before. The received output symbols at time k, y k equals h, this is the model for a typical wireless channel, h times x k + v k, which is the noise.

And what is y k, y k is symbol received at time k. Just to your memories, symbol received at time k, h is your fading Channel coefficient, h is a fading Channel coefficient in fact the flat fading Channel coefficient, x k is the transmitted symbol at time instant k. What is this? This is transmitted symbol at time k and v k is basically the noise sample, the Gaussian noise sample at time k.

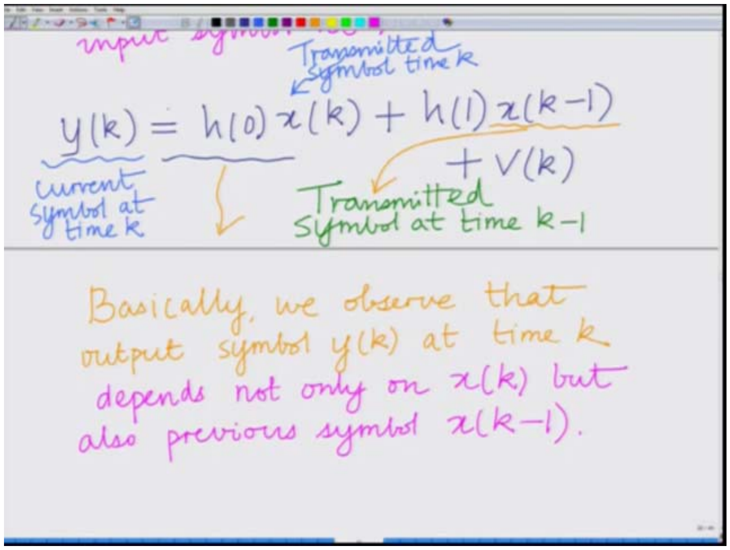
So I have the system model which can be described as y k equals h times x k + v k; y k is the received symbol, h is the fading coefficient, x k is the transmitted symbol, v k is the noise sample. And observe that in this model, the received symbol y k depends only on the current transmitted symbol x k. So this is a very subtle but an important aspect that the received symbol depends only on the current input x k.

(Refer Slide Time: 03:45)



So let us observe this, let us note this about this model. So observe that in this model, in this model observe that the current output y k on the current symbol output or symbol y k depends only on the current input symbol input symbol x k. It depends only on the current input symbol x k, right. So the current output symbol y k, it depends only on the current input symbol x k.

(Refer Slide Time: 04:45)



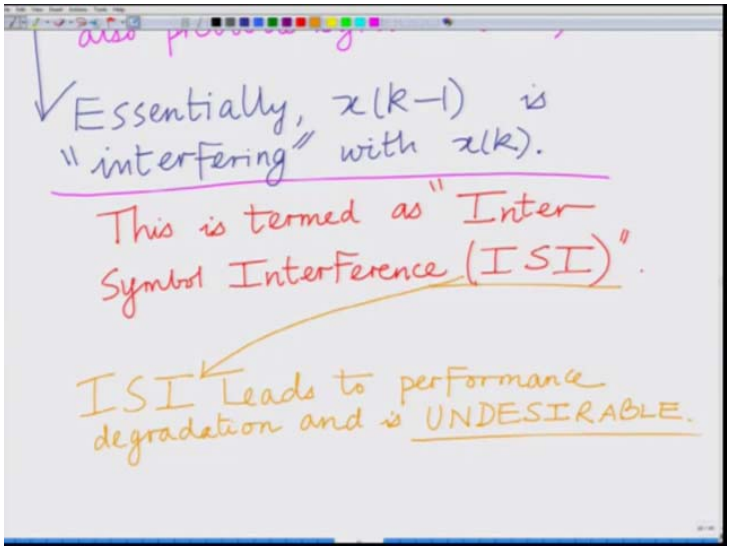
However, we might frequently have another scenario which is an undesirable scenario in which the following occurs. So frequently also it might happen that we have y k, the received symbol y k is given as h 0 times x k + h 1 times x k - 1 + v k, which is the noise. And if you look at this model, you will observe that this is y k, this is again the current symbol at time k, x k is the current input symbol that is, transmitted able at time k.

That aspect is fine however, we also have something very interesting, we have also x k - 1 that is the transmitted symbol or the input symbol at time k - 1. We have the transmitted symbol x k - 1, this is the transmitted symbol. To write it down, this is the transmitted symbol at time instant k - 1.

So what we are observing is that unlike previously, where the received output symbol y k depends only on the transmitted symbol x k at time instant k. What we have in this scenario is output symbol y k, it depends not only on x k, but it also depends on the previous symbol that is x k - 1. So let us write this down, since this is an important aspect that is what we are observing basically, we observe.

We observe that the output symbol, we observe that output symbol y k at time k depends not only on x k, but also on the previous symbol that is depends also, depends not only on x k, but also the previous symbol x k - 1. That is, y k depends not only on x k, but also the previous symbol x k - 1. Basically, which means that the previous symbol x k - 1 is interfering at the current symbol x k.

(Refer Slide Time: 08:26)



That is, it is causing interference in the detection of the current symbol x k, this is known as Inter Symbol Interference, all right. So essentially, x k - 1 is to put it, this is interfering, correct. This is interfering with x k, x k - 1 is interfering with x k, this is termed as Inter Symbol Interference or ISI, so this is this phenomenon is termed as Inter Symbol Interference or ISI.

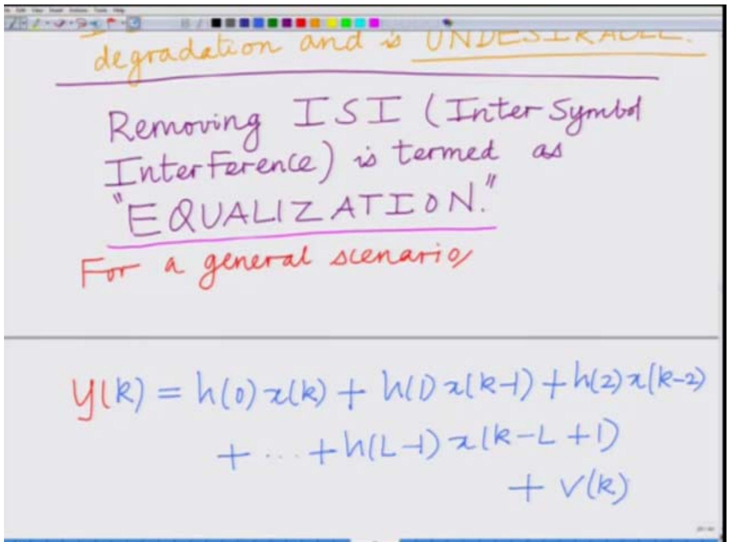
Where the previous symbol x k - 1 is also interfering with the current symbol x k that is, y k has the effect of both x k and x k - 1, this is termed as Inter Symbol Interference. And naturally, this ISI or Inter Symbol Interference leads to performance degradation because there is interference because x k - 1 is interfering with x k, this leads to performance degradation, correct at the receiver.

So this is an undesirable affect, so this ISI leads to performance degradation naturally and is undesirable and Inter Symbol Interference is basically undesirable. Hence therefore, we would like to remove Inter Symbol Interference that is what we would like to do at the receiver for the detection of x k.

That is you want to infer what x k is, we would like to remove the interference from x k - 1. And this process of Inter Symbol Interference that is this removing this Inter Symbol Interference is termed as equalisation or Channel Equalisation. So Channel Equalisation is nothing but removing the effect of Inter Symbol Interference to improve the performance at the receiver.

So removing this Inter Symbol Interference essentially removing this ISI Inter Symbol Interference is basically what is termed as equalisation. So let me write that down, this process of removing Inter Symbol Interference is basically what is termed as Equalisation, Correct.

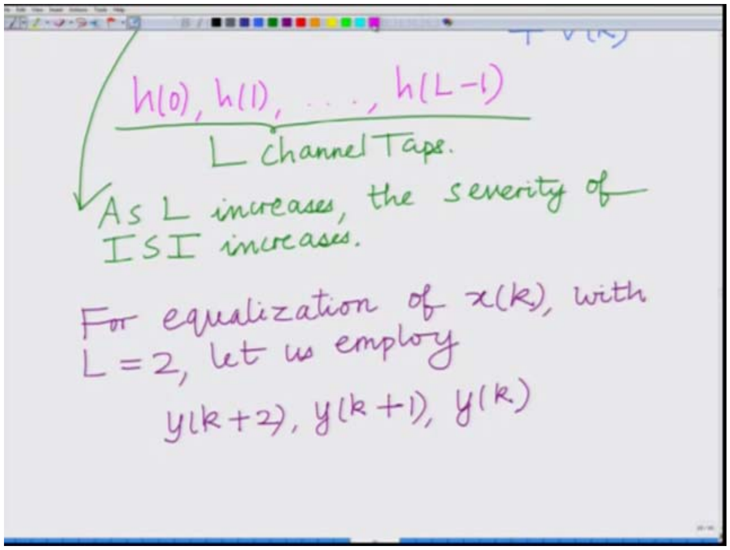
(Refer Slide Time: 11:45)



So this process of removing the effect of x k - 1 on x k is termed as, or basically removing this interference from x k - 1 on x k or the Inter Symbol Interference is termed as Equalisation. And further I would also like to point out that the Inter Symbol Interference need not be restricted to x k - 1, it can be on a larger number of previous symbols. So here in this system model, if we will look at your system model over here.

I can in general have that is y k that is for a general L, for a general scenario, I can have y y k equals h 0 k k + h 1 x - 1 + h 2 x k - 2 + 1 h L - 1 x k - L + 1 + v k, correct. Therefore, you have L channel taps, so you have h 0, h 1, h 2, h L - 1. These are the coefficients, these are known as channel taps. So h 0, h 1, h L - 1, these are known as the channel taps.

(Refer Slide Time: 13:45)



These are basically your L channel taps, okay. So we are saying for a general system with Inter Symbol Interference y k equals h 0 x k + h 1 x k - 1 so on so forth until h L - 1 times xk - L + 1 + the noise sample v k. And therefore, we have L channel taps; h 0, h 1 up to h L - 1.

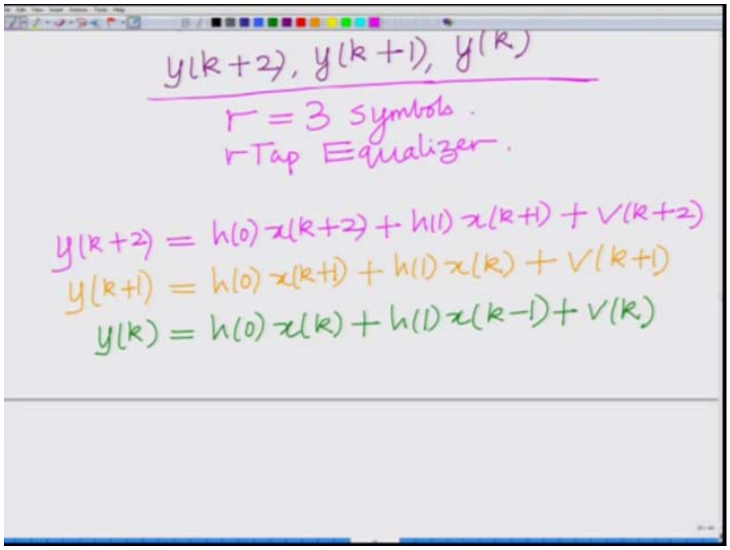
And therefore, as L increases, the severity of ISI or Inter Symbol Interference increases, which means more and more symbols previous symbols x k - 1. X k - 2 interfere with the current symbol x k so as the severity as L increases, the severity of the ISI increases, all right so that is basically the idea.

And then that what you said is basically, we want to remove the effect of this Inter Symbol Interference at the receiver, that is termed as Equalisation. And that can be done as follows, what we will do is to basically detect x k, we will also employ x k, y k + 1 and y k + 2. So previously, for a flat fading scenario to dictate x k, we simply employ the received symbol y k.

However, now because there is Inter Symbol Interference, we will not only employ y k, but we will also employ y k + 1 and y k + 2, which is what I am going to illustrate now. So Equalisation for equalisation k of course let us go back to our simple scenario with L equal to 2, let us employ y k + 2, y k + 1, y k. That is, we are employing 3 symbols or basically R equal to 3 symbols, so this is known as a 3 tap equaliser.

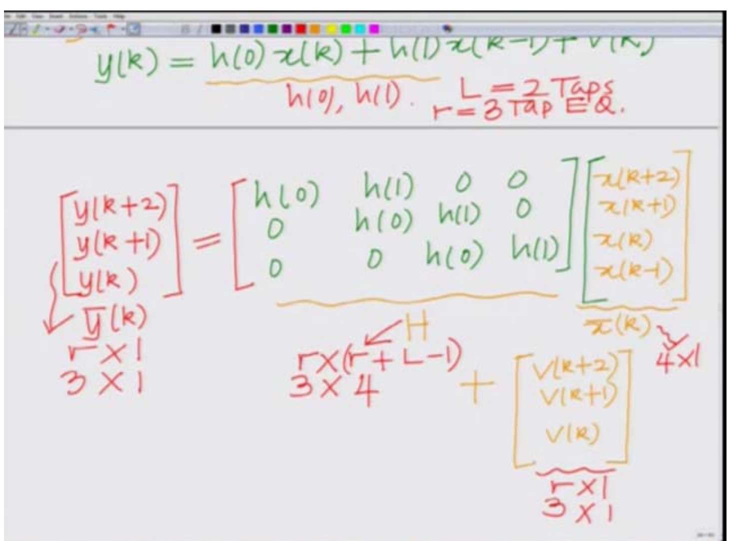
So we are employing R equal to 3 symbols that is, y k + 2, y k + 1, y k, this is known as a R tap or basically 3 tap, this is known as a R tap or 3 tap equaliser. And what we are going to do is basically we are going to linearly combine these symbols, all right. So we are going to linearly combine these symbols to eliminate the effect of Inter Symbol Interference.

(Refer Slide Time: 16:48)



So first let us write the system model for these 3 symbols y k + 2, y k + 1, y k, so you can see that well, y k equals well h 0 x k + h 1 x k - 1 + v k. Now y k + 1 will be similarly, h 0 x k + 1 + h 1 times the previous symbol that is x k + v k + 1 and similarly, y k + 2 equals h 0 times x k + 2 + h 1 times x k + 1 + v k + 2. Now what I am going to do is I am going to tag these symbols y k + 2, y k + 1, y k as a vector.

(Refer Slide Time: 18:13)



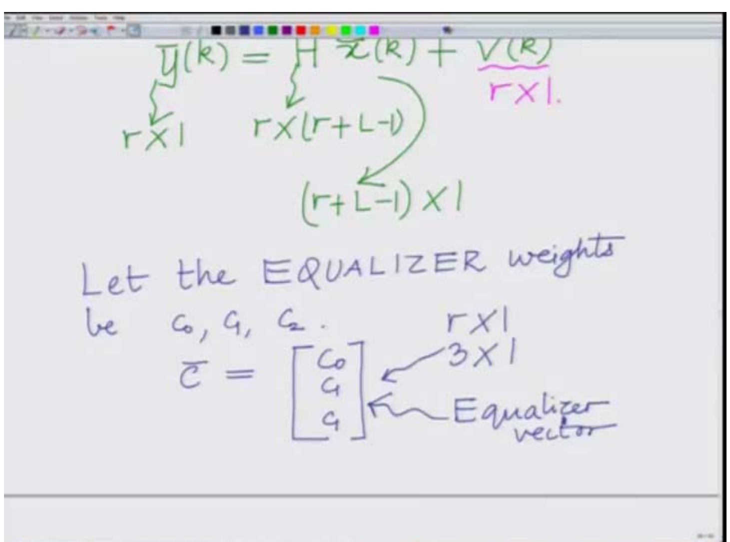
And you can see; now I will get the system model. So when I stack these as a vector, what I will get is y k + 2, y k + 1, this is the vector remember we can call this the vector y bar k, which is an R cross 1 or basically a 3 cross 1. Remember, we are looking at a 3 tap equaliser, so I have an R cross 1 vector. This is equal to the following matrix, h 0, h 1, 0, h 0, h 1, 0, 0 h 0, h 1 times, now my symbols x k + 2, x k 0 1, x k and x k - 1.

Let us call this your matrix h, let us call this your matrix your vector x bar k + your noise vector, the noise vector is basically v k + 2, v k + 1, v k. So basically, what you can observe is here we are considering an L equal to 2 tap channel, because we have h 0, h 1 as the channel taps, so we have L equal to 2 taps, R equal to 3 tap equaliser, so we have here R equal to 3 tap equaliser, so this y bar is basically a 3 cross 1.

Now this matrix H is you can see is R cross R + L - 1 that is, 3 cross 3 + L equal to 2, 2 - 1, so 3 cross 4, x bar is R + L - 1 cross 1, so this is 4 cross 1 and v bar k is simply R cross 1 that is 3 cross 1. So now we have written the vector model for this input for this basically Inter Symbol Interference channel, which is basically y bar k which contains the symbol y k + 2, y k + 1, y k, equals the matrix h.

Which is found out of the channel coefficient, which is an R cross R + L - 1 matrix, where R is the number of taps in the equaliser, L is the number of taps of the channel coefficients times x bar k, which is R + L - 1 cross 1 + v bar k which is an, this is annoying vector which is R cross 1. And now we would like to design our equaliser, so let me just write this down again.

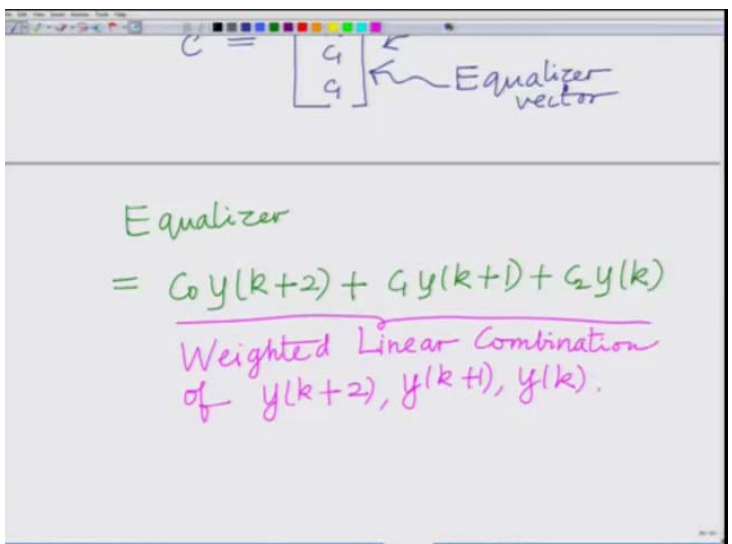
(Refer Slide Time: 21:28)



To summarise this thing, if we consider an R Equaliser, so I have y bar k equals h times x bar k + v bar k, where this is R cross 1, this is R cross R + L - 1, this is x is basically R + L - 1 cross 1 and this v bar k, this is simply an R cross 1 vector and we would now like to design an equaliser. Remember, we said we would like to design the equaliser based on the 3 symbols; y k + 2, y k + 1, y k.

And we would like to linearly combine these symbols y k + 2, y k + 1, y k, so let the combining weights be C 0, C 1, C 2. So let the equaliser weights or the combining weights let the equaliser weights be C 0, C 1, C 2. Therefore, now you can denote this equaliser, so I can denote this equaliser by C bar, which is C 0, C 1, C 2. This is observe this, this is an R cross 1 vector, this is R cross 1 or basically 3 cross 1 vector.

(Refer Slide Time: 23:25)



This is the equaliser, also we can call this as the equaliser vector and our equaliser and our equaliser therefore is now given as C 0 times C 0 times y k + 2 + C 1 times + y k + 1 + C 2 times y k, which is basically the linear this is basically the linear combination of y k + 2, y k + 1 and y k. This is basically your or weighted linear combination, let me put it that way.

This is your weighted linear combination of the outputs y k + 2, y k + 1, y k and y k. So what we are saying is basically, we have a 3 tap equaliser which comprises of the weights C 0, C 1, C 2, these are the 3 the equaliser vector C bar. And now what we're doing is we are taking the outputs y k + 2, y k + 1, y k, we are linearly combining them using these weights as C 0 times yh k + 2 + C 1 times y k + 1 + C 2 times y k, this is the operation of the equaliser.

So we would like design the weights basically, C 0, C 1, C 2, so as to eliminate the Inter Symbol Interference for the detection of x k, that is the problem of equaliser. So what we have done in this module so far is to basically motivate this problem of Inter Symbol Interference when we have the current symbol y k depending not only on the input symbol, x k but also the past symbols x k - 1 and so on, all right.

So this is known as Inter Symbol Interference and Inter Symbol Interference leads to performance degradation which has to be removed and the receiver and this is known as equalisation and for equalisation, we have to design this equaliser. Equaliser is basically nothing but, these are the coefficients or the weights C 0, C 1, C 2, which linearly combine output symbols.

That is, we consider a 3 tap equaliser linearly combine the output symbols y k + 2, y k + 1, y k to basically equalise or remove the Inter Symbol Interference towards the detection of x k. So how exactly do we design these weights C 0, C 1, C 2 or how exactly we design this equaliser that we are going to look at the next module, all right? So let us stop this module here, thank you very much.