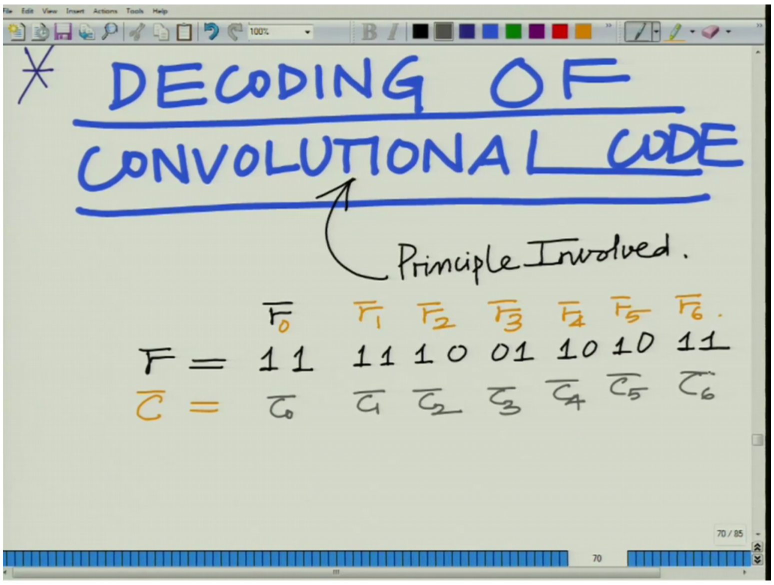
Principles of Communication Systems - Part II
Prof. Aditya K. Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture - 54
Decoding of Convolutional Code, Minimum Hamming Distance, Maximum Likelihood Codeword Estimate

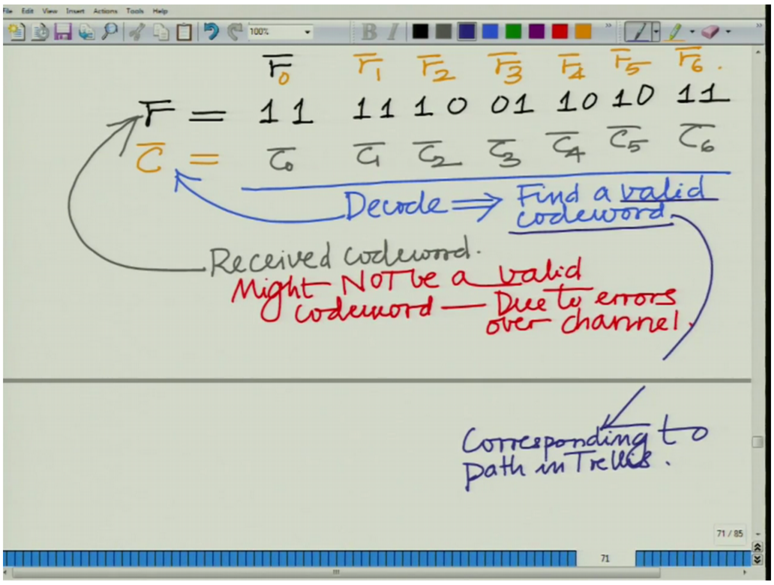
Hello. Welcome to another module in this massive open online course. So, we have looked at the trellis presentation of the convolution code. Let us now look at the principal for the decoding of the convolutional code.

(Refer Slide Time: 00:24)



So, let us look at the decoding or rather the principle involved. So, let us see we have a received code word r bar which is given as one 1 1 1 1 1 0 0 1 1 0 1 0 1 1. So, this is basically r bar at time instant 0, r bar correspondent to time instant 1, r bar correspondent to time instant 2, r bar 3, r bar 4, r bar 5 and r bar 6 and we have to decode means basically we have to find the corresponding and we would like to find the corresponding decoded code word c0 bar, c1 bar, c2 bar, c3 bar, c4 bar, c5 bar.

(Refer Slide Time: 02:20)

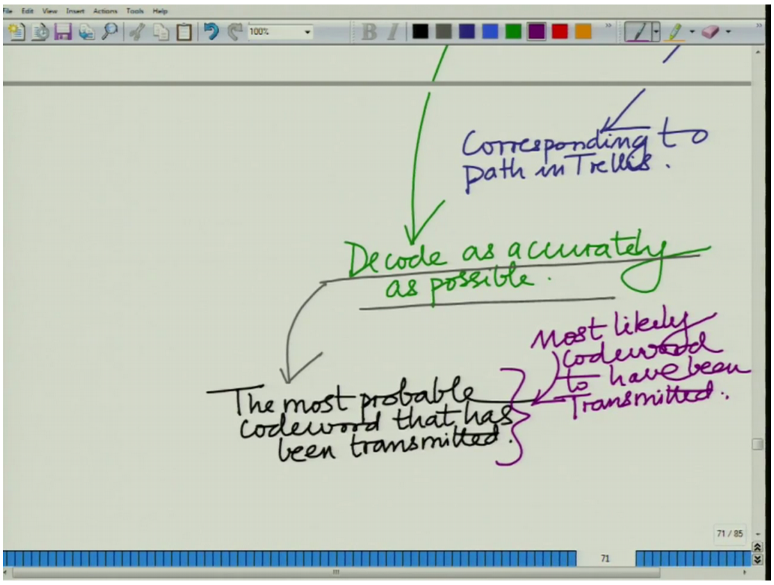


Now, note that this is the received code word, correct. Now, note that we have said that any legitimate code word corresponds to a path through the trellis, but this might not be a valid code word. It is important to note that this might not be a valid code word. This is important to realize the received code word might not be a valid code word, reason being due to errors over or due to errors introduced that is the whole point of introducing coding, right. So, we are transmitting a valid code word, but what you generate, what you receive at the receiver in this digital communication system might not be a valid code word because the communication channel can flip some bits. Once you flip some bits, it is not guaranteed to be a valid code word to anyone.

So, corresponding to the received code word, we have to find a valid code word that might have been transmitted, but this is the challenge which means we have to recover the errors that are basically, that have been basically introduced by the channel and as we have seen, it is not, it might not be possible to recover or to basically correct all the errors to recover or decode this received code word in the sense that decode this received code word to the best possible extent, ok to find the code word.

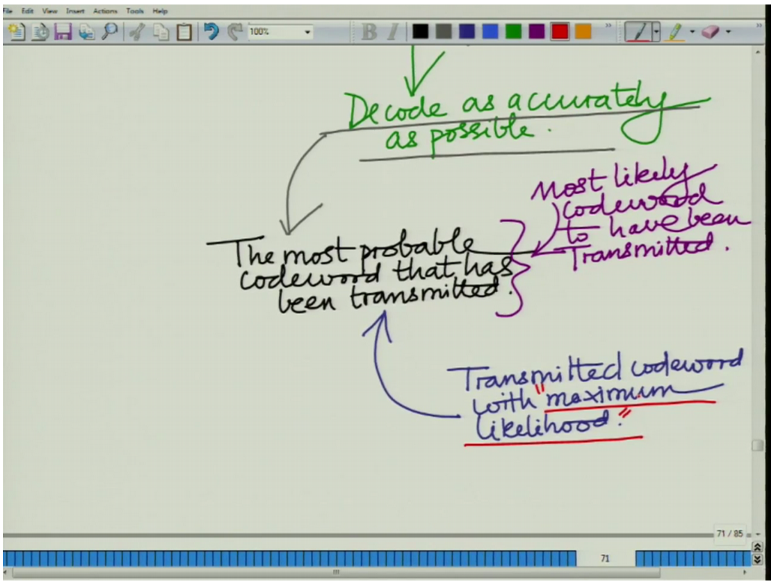
So, decode implies find a valid codeword, correct. Now, a valid codeword obviously we said valid code word means corresponding to a path in the trellis, but there might be many valid code words, right because there are many parts in the trellis. So, any of them can be valid codeword. So, how we decide which is the valid codeword? The point is we want to decode the corresponding code word to the highest, to basically the most possible extent, correct.

(Refer Slide Time: 05:26)



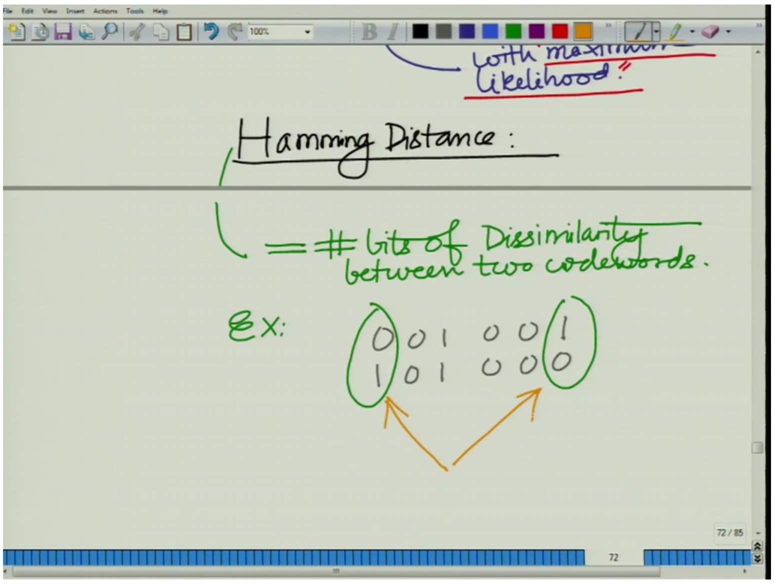
I can say as accurately as possible. So, decode as accurately as possible. Now, how do you decide which codeword is the most probable code word? So, what we want to do is, somehow we would like to come up with the metric when use as accurately as possible, would like to come up with the metric to deceive the most probable codeword that could Have been transmitted. Now, the most probable codeword that has been transmitted or basically the most likely or another way of say it is most likely codeword to have been transmitted or the codeword with the maximum likelihood, the transmitted codeword with the maximum likelihood.

(Refer Slide Time: 07:11)



To find that we want to define what is known as Hamming distance. We want to define the Hamming distance.

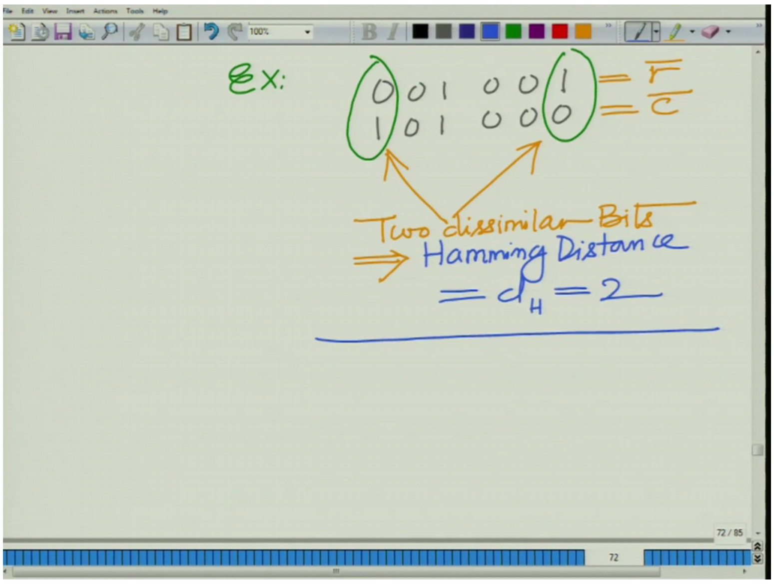
(Refer Slide Time: 07:47)



Hamming distance is nothing, but the number of bits of dissimilarity. This is equal to number of bits of dissimilarity. The hamming distance between two code words is a number of bits of dissimilarity between two code words.

For example, we have code word 0 0 1 0 0 1 1 0 1 0 0 0. Look at this. There are two points of dissimilarity 0 1 and 1 0. So, we can say 0 has been flip to 1, 1 has been shift to 0. So, 2 bits are dissimilar.

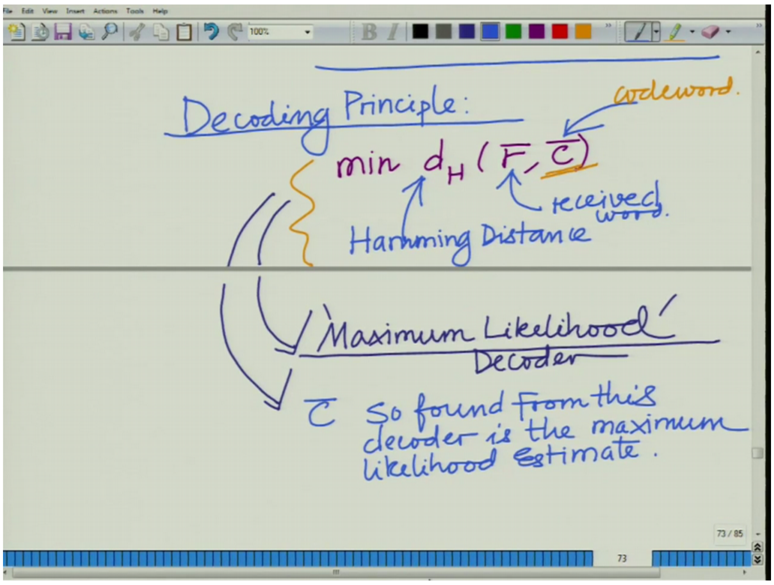
(Refer Slide Time: 09:07)



So, let us say this is c1, this is equal to c2 or wait or still better to a loot. Why we need this? Let us note this as the received code word r, this is the corresponding codeword c bar. So, there are two dissimilar bits. This implies hamming distance which will denote by d H. The hamming distance d H is equal to 2, ok.

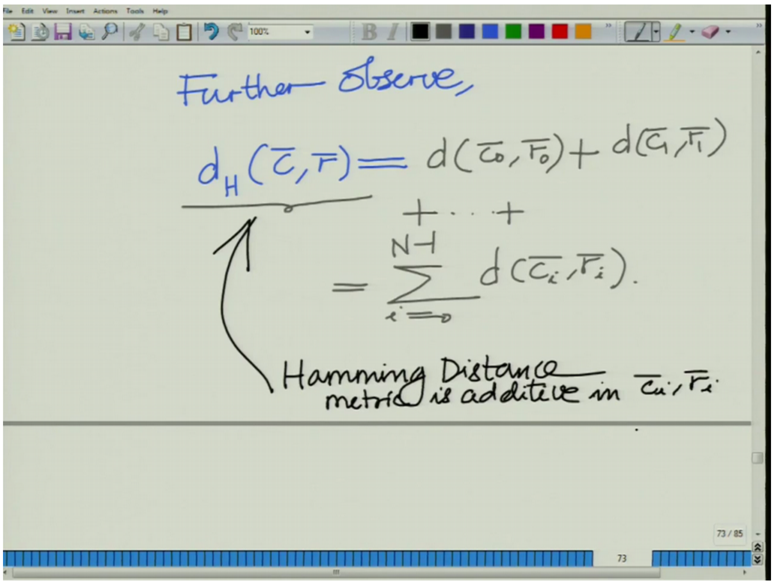
Now, the principle is clear. The most likely code word, there is a maximum likelihood estimate of the transmitted codeword is 1 which has the minimum hamming distance from the received codeword. We are not going to rigorously justify this, but it is intuitively clear we are basically trying to find codeword, but not any codeword because several possible code word can be generated by the trellis. We are interested in finding that code word which is as close as possible to the received code word.

(Refer Slide Time: 10:36)



So, the decoding principle can be further stated more precisely as that is, we want to minimize the hamming distance between r bar received code word and c bar. So, remember d H is the hamming distance, r bar is the receive, r bar is the received word and c bar is, so we want to find that particular c bar which is the closest, which has the minimum hamming distance with respect of the received word r at this principle that is system has the maximum likelihood decoder. This principal is termed as the maximum likelihood decoder and this estimate c bar, the code word estimate c bar is termed as a maximum likelihood estimate or is the maximum likelihood decoder and the c bar. So, found is termed as, so found from this decoder is the c bar. So, found is the maximum likelihood estimate, ok.

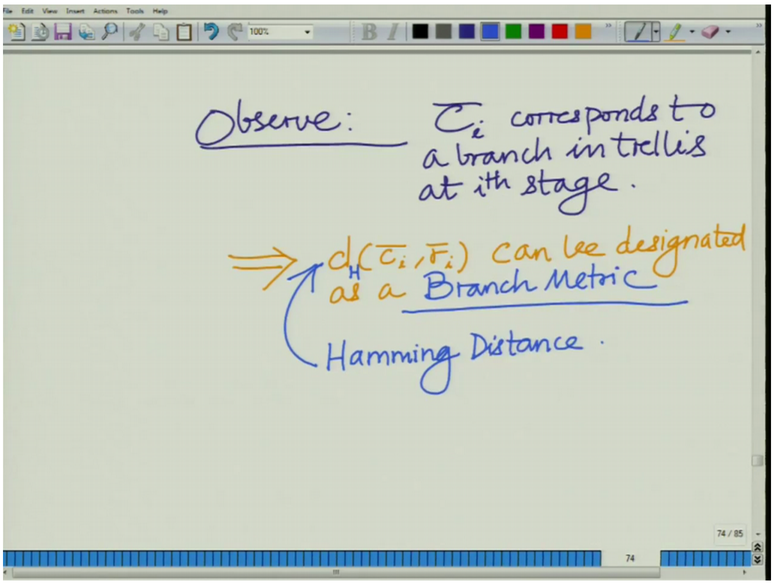
(Refer Slide Time: 13:08)



Now, further observe that this hamming distance metric which is basically the number of points of dissimilarity, number of bits of dissimilarity, this is additive. This is the hamming distance between c naught bar, r naught bar plus the hamming distance between c1 bar number of dissimilar bits between c1 bar r1 bar so on and so forth. That is basically submission i equal to 0, 2 in minus 1, where n is the block length, the hamming distance between c i bar.

So, this hamming distance metric is additive that is the total hamming distance between the code words can be found at some of the constituent component hamming distances. So, this hamming distance metric is additive. This is very additive in c bar i, r bar i. So, you add the hamming distances of the varies components between, that is if u will look at each time instant i c bar i and received code word r bar i, take the hamming distance correct, you add all these hamming distance, you will get the overall hamming distance between c bar and r bar. That is an important point. It is a very obvious point, all right. It is very obvious and intuitive also. It is very important, all right to realize this and because this is going to lay the foundation of the decoding procedure that we are going to illustrate subsequently, ok.

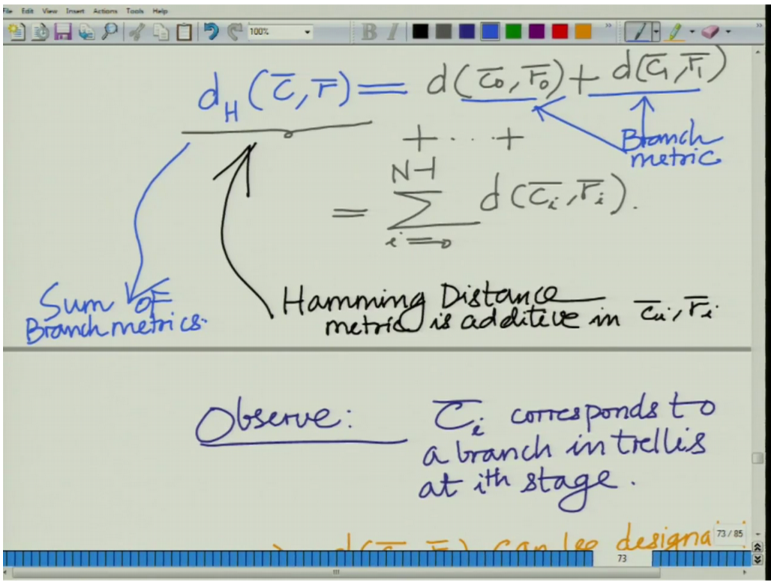
(Refer Slide Time: 15:08)



So, observe that the important observation that c bar i corresponds to a branch each codeword c bar i. So, for each branch we have a generated code word c bar i. So, c bar i corresponds to a branch in the trellis at the ith stage. This basically implies that d of c bar i, r bar, I can be designated as a branch metric. That is the important point. This can be designated thought of as a branch metric. So, I have r i and for any branch in the trellis that is at the ith stage, I have several branches. So, I have states at i minus 1 i plus 1 states i minus 1 states at i several branches going from i minus 1 to i. So, each branch corresponds to a certain code word c bar i, all right. I can look at the distance between c bar i and r bar i and I can designate the branch with this hamming distance. This is denoted by the branch metric, ok.

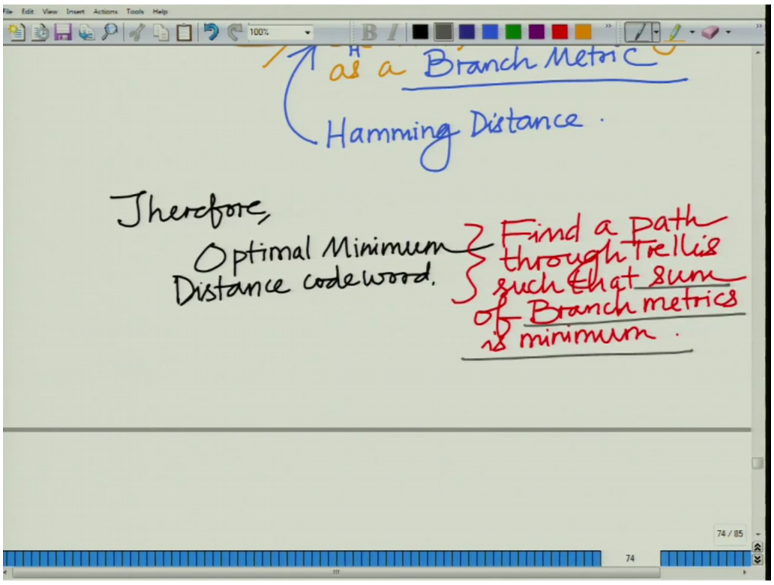
So, I am just writing the distance here, but it should be obvious that we are considering the hamming distance. I am not going to explicitly keep writing d H, but all such distances are the hamming distance. So, we are going to consider only the hamming distance because our optimality criteria and everything is in terms of the hamming distance and therefore, for the optimal minimum distance.

(Refer Slide Time: 17:32)



Now, we have seen the total distance. So, now you can see the total distance that is d H bar, this total distance is some of the branch metrics because each of these is the branch metric. That is what we have seen each of these is a branch metric, ok. So, this over all hamming distance is basically nothing, but the sum of branch metrics.

(Refer Slide Time: 18:08)



So, basically decoding implies for optimal minimum distance codeword for the optimal minimum distance code word. We have to find path through the trellis, such that some of the branch metrics is minimum. That is the point. Find a path through the trellis, such that the sum of the constituent branch metrics is minimum that is the point. So, we have to find a path through the trellis, such that this path as the minimum.

So, what we are doing is very simple or what we are going to do is basically denote each branch in the trellis with the branch metric. This branch metric is the hamming distance between the codeword that is the correspond generated by that branch and the received word at that time instant. Now, we have to find path through the trellis, such that the sum of its branch metrics is minimum that corresponds to the minimum hamming distance codeword that corresponds to the code word which has the minimum hamming distance with respect to the received word.

Therefore, intern this code word is the maximum likelihood estimate of corresponded, the maximum likelihood estimate of the transmitted codeword corresponding to the received codeword r and therefore, now our life, our task is basically simple. We have to find because we said that any valid code word corresponds to a path through the trellis. We have to find a path through the trellis. Now, there are many paths through the trellis. We want to find the path through the trellis which has the minimum metric. What is the minimum metric hamming distance? Minimum hamming distance with respect to the received codeword and that minimum hamming distance can now be split in terms of the hamming distances of the constituent branches of this path and therefore, we want to find that path which has the minimum over all metric and that is what we are going to do in the subsequent models.

Thank u very much.