Estimation for Wireless Communications – MIMO/OFDM Cellular and Sensor Networks
Professor Aditya K Jagannatham
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Lecture Number 24
Equalization and Approximation Error for Zero Forcing Channel Equalizer

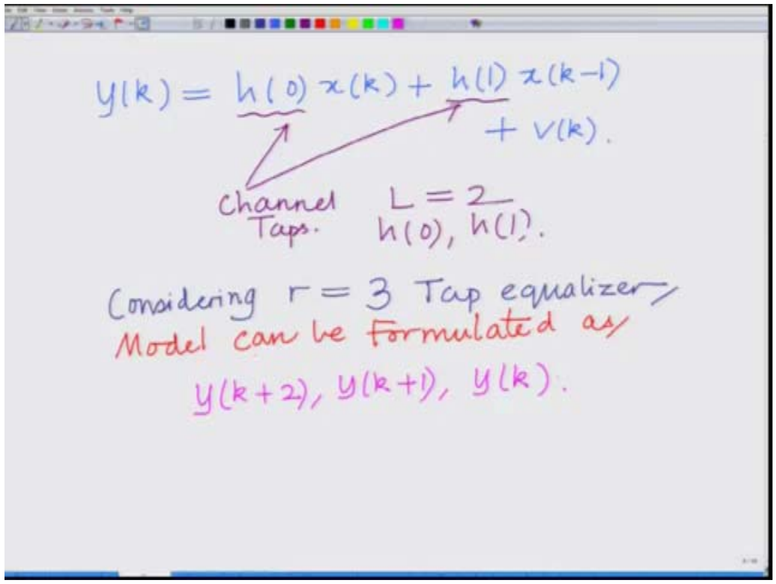
Hello! Welcome to another module, in this massive open online course on 'Estimation for Wireless Communication'. Right? So we're currently looking at equalization, and as we already said equalization is done to remove the ISI, or Inter-Symbol Interference in a wireless channel, where the previous symbols interfere with the current symbols, alright? So we have formulated the system model for equalization and we have also derived the (equali), the zero forcing equalizer based on the least squares principle.

Let us today, or in this module, look at another aspect, that is the 'approximation error of this equalizer'. So what we're going to look, do today, is to basically derive what is the error, or let's call it the 'approximation', the 'approximation error' for your, for the, the 'approximation error' for the equalizer. And as we've already seen, let us begin with the ISI channel model. Consider, consider your ISI or Inter-Symbol Interference channel, where the received symbol y(k) is basically given as h(0) times x(k) plus h(1) times x(k minus 1), plus v(k). This is the model for the channel.

We also said that this is the model which captures the Inter-Symbol Interference, y(k) is the received symbol, x(k) is the symbol transmitted at the current time instant k. x(k minus 1) is the symbol corresponding to the previous time instant k minus 1, which is interfering with the symbol x(k), that is the current symbol, alright? And that is basically Inter-Symbol Interference, alright? And v(k) is of course the noise sample. And these two co-efficients h(0) and h(1), these are known as the taps of the channel, alright?

So we have h(0), h(1), which are basically your channel taps. These are the taps of the channel and we have L equal to two channel taps. We are considering an Inter-Symbol Interference scenario with L equal to two channel taps, that is h(0), comma h(1). So now considering an r tap (eq), where the r equal to 3; considering an r equal to 3, an r equal to 3 tap equalizer, we can formulate this model as the model, corresponding system model can be formulated as, the corresponding system model can be formulated as, y(k) plus 2, that is we're considering an r equal to 3 tap equalizer, based on the symbols.

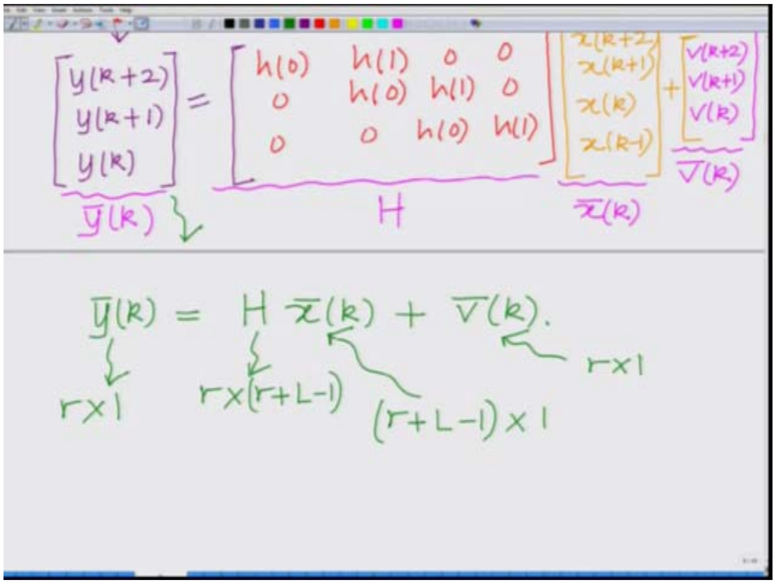
(Refer Slide Time: 04:36)



This is 3 tap equalizer, which is basically based on your, so let me write what it is based on. It is based on the 3 taps, act on the symbols y(k plus 2), y(k plus 1), and y(k), alright? And therefore, the model, the system model at the receiver can be expressed as your vector of y(k plus 2), y(k plus 1), y(k), which is equal to the co-efficients h(0), h(1), 0, 0; 0, h(0), h(1), 0; 0, 0, h(0), h(1), times the symbol vector x bar k, which is x of (k plus 2), x of (k plus 1), x of k, x of (k minus 1), plus of course, I have the noise vector and the noise vector is v of (k plus 2), v of (k plus 1), and v of k.

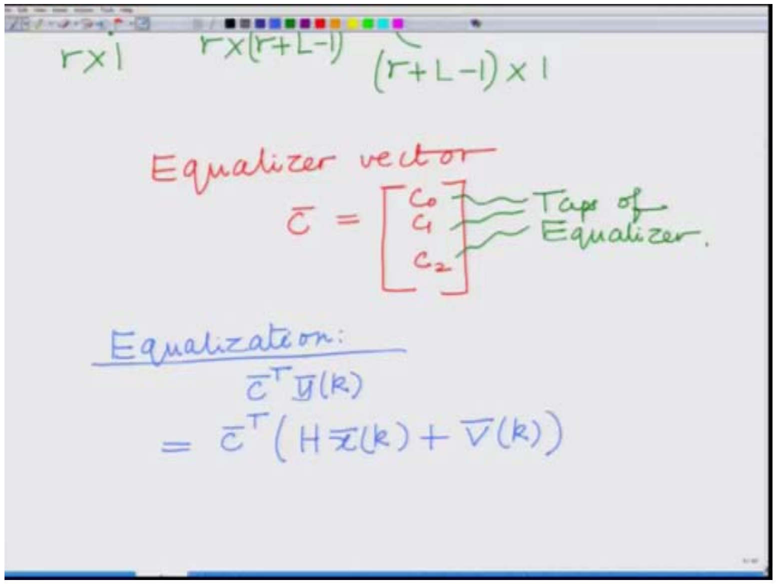
So this is my received vector y bar of k, this is my matrix h, this is my symbol vector x bar of k, and this is my noise vector v bar of k. This is something that we've already seen. We can stack the symbols, the three symbols, y(k plus 2), y(k plus 1), y(k), and write it as y bar k equals the matrix H times x bar k plus v bar k, where x bar k is the vector of transmitted symbols, x(k plus 2), x(k plus 1), x(k), x(k minus 1), and v bar k is the noise vector, alright? So I can summarize this model. This model can be summarized as y bar k equals H x bar k, plus v bar k, and we've already seen this is an r cross 1 vector, this is an r cross r plus (L minus 1) vector, this is an r plus (L minus 1) cross 1 vector, and this is an r cross 1 vector, and this is an r cross 1 vector, and now we have our 3 tap equalizer vector, that is c bar.

(Refer Slide Time: 07:30)



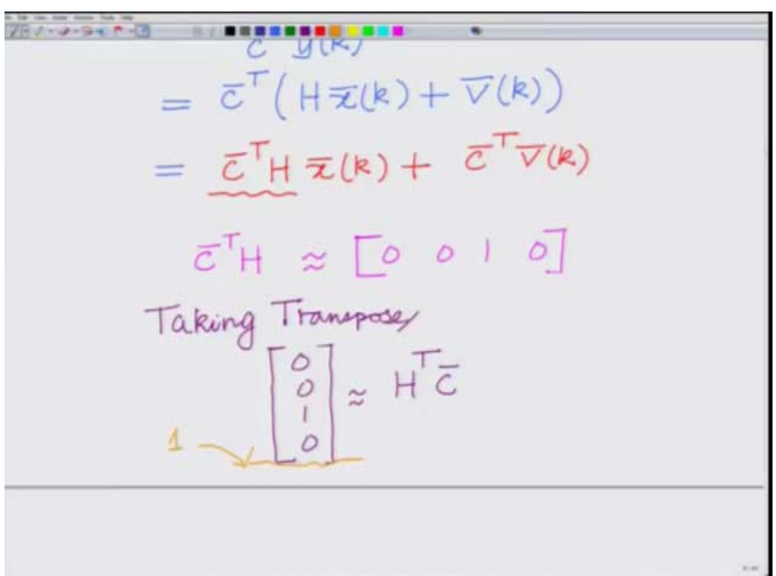
c bar, that is equal to c(0), c(1), c(2), that is the column vector c(0), c(1), c(2), which are the taps of the equalizers, or equalizer vector. So our equalizer vector, c bar equals c(0), c(1), c(2), and we said, these quantities c(0), c(1), c(2), these are the taps of; these are the taps of the equalizer. And our equalization process itself, the process of equalization, is basically c bar transpose y bar k, which is equal to c bar transpose, substituting for y bar k, I have y bar k equals H x bar k, plus v bar k, which is equal to c bar transpose H x bar k plus c bar transpose v bar k.

(Refer Slide Time: 09:00)



And we said we want, ideally we want this vector c bar transpose H to be as close as possible to this vector 0, 0, 1, 0; so our equalizer design criteria was basically to zero out the interference from x(k plus 2), x(k plus 1), and x(k minus 1) on x(k). So therefore, this vector c bar transpose H should be as close as possible to the row vector 0, 0, 1, 0; or taking the transpose of this, now we have taking the transpose, we have, basically, this vector 0, 0, 1, 0, should be as close as possible to this vector H transpose c bar, and this vector we denoted by this vector 1 bar 2,

(Refer Slide Time: 10:26)



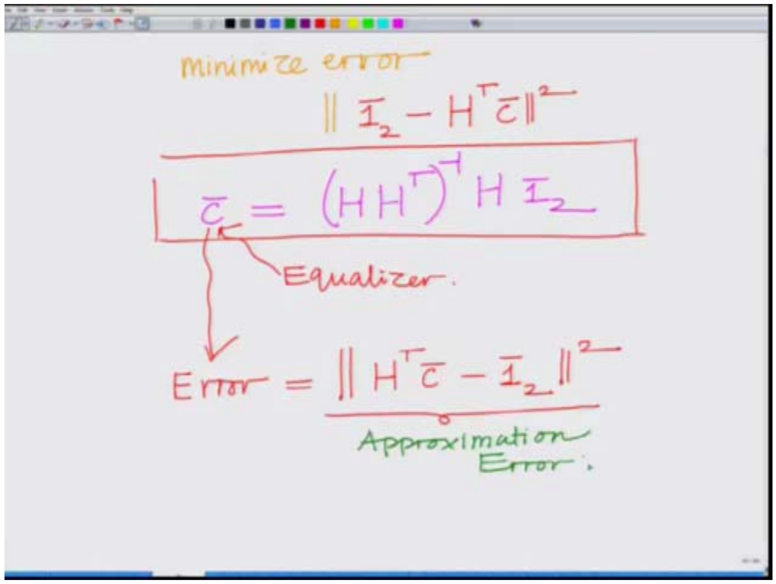
which is 1 in the zeroth position, or in other words, we want to minimize, minimize the error, where the error is basically, the error, the difference between this vector 1 bar 2, and H transpose c bar.

We want to minimize the error, or basically the norm of the error, the norm square of the error. So basically what we've done, is we've formulated the system model, and we have reduced this equalizer design problem to a least squares problem. Basically by saying that, this vector H, H transpose c bar, should approximate or should be as close as possible to this vector 1 bar 2, which means the square of the norm of the error should be the least. Therefore we have reduced it to the least squares problem.

Now once we've reduced it to the least squares problem, the equalizer design itself is given by the least square solution. That is something that we're already familiar with, and therefore, the equalizer design, equalizer is basically your, the pseudo inverse of H transpose. Remember the pseudo inverse of H transpose, which is H H transpose inverse times H into this vector 1 bar 2, and this is the equalizer, this is the equalizer that we already designed and we've also seen it through an example.

This is the equalizer that we've already seen through the example. Now what we want to find out is we want to find out what is the error, that is how good is this equalizer, that is, how good of an approximation is this to the vector 1 bar 2, alright? So we want to now find out the

(Refer Slide Time: 12:49)

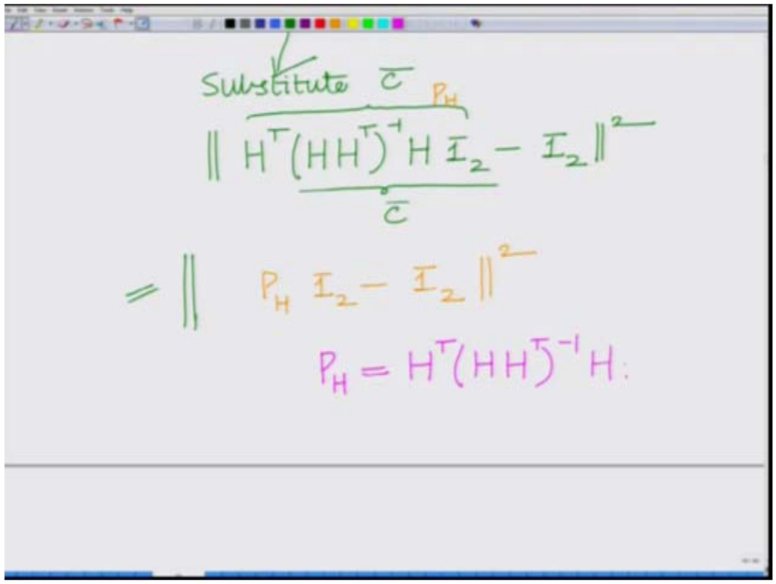


error, that is the error, which is equal to basically your, what is this ‘approximation error’? Right? This is we said, this is our ‘approximation error’. So what is the ‘approximation error’? So we said this is the error, so we want to ask the question, what is this ‘approximation error’?

In other words, if we substitute c bar, that is our least squares equalizer; substitute c bar, what do we get? So we get norm H transpose c bar, but c bar is basically, as we have derived above is H H transpose inverse, H into the vector 1 bar 2. So this is basically H H transpose inverse, H into 1 bar 2 minus this vector 1 bar 2 whole square. And this is basically the expression for c bar, that we substituted from above, and now I can write this as, now I can write this as norm of, now let us denote this whole matrix by the quantity PH, I can write this as PH times this vector 1 bar 2 minus 1 bar 2 norm square.

And this PH will have an important role to play, which we have just defined this PH, let us simply at this moment, PH is simply H transpose H H transpose inverse times H. So I have, so I

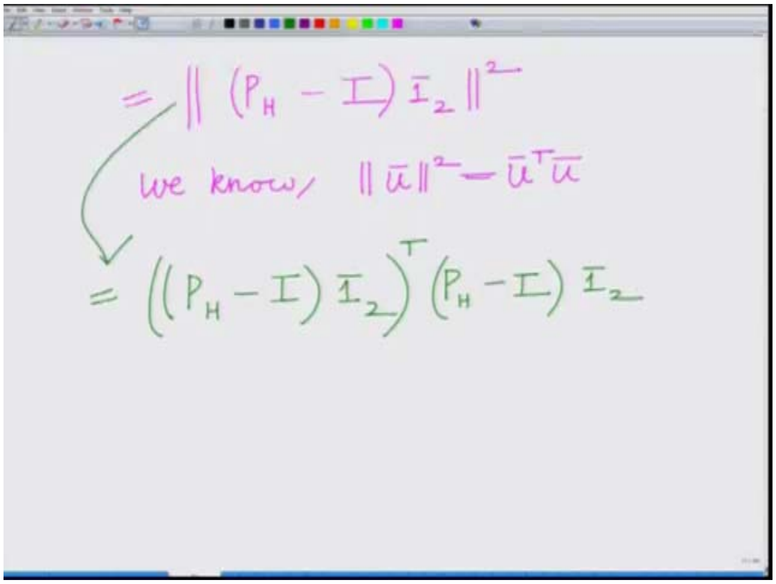
(Refer Slide Time: 14:24)



want to find the 'approximation error', that is how good is the equalizer (())(14:28), that is how good is, how good is the, how well is the equalizer able to approximate this vector 1 bar 2, which is basically H transpose c bar minus 1 bar 2 the norm square. And I've substituted the expression for c bar, alright? And I've also, denoted by PH, this matrix which is H transpose times H H transpose inverse times H, alright?

This matrix, we have denote, we are denoting by PH. Therefore now, this is equal to norm of, now I can write this as PH minus the identity matrix times the vector 1 bar 2 whole square, okay? And now, we know that for any vector v bar, are any vector u bar, norm u bar square equals u bar transpose u bar, correct? We know that for any vector u bar, norm u bar square, that is norm of a vector square is the vector transpose times itself. And now I'm going to use this property over here. Just simplify it little bit, that is, the norm of this vector square is basically this vector transpose, times itself, which is basically equal to 1 bar 2.

(Refer Slide Time: 16:10)

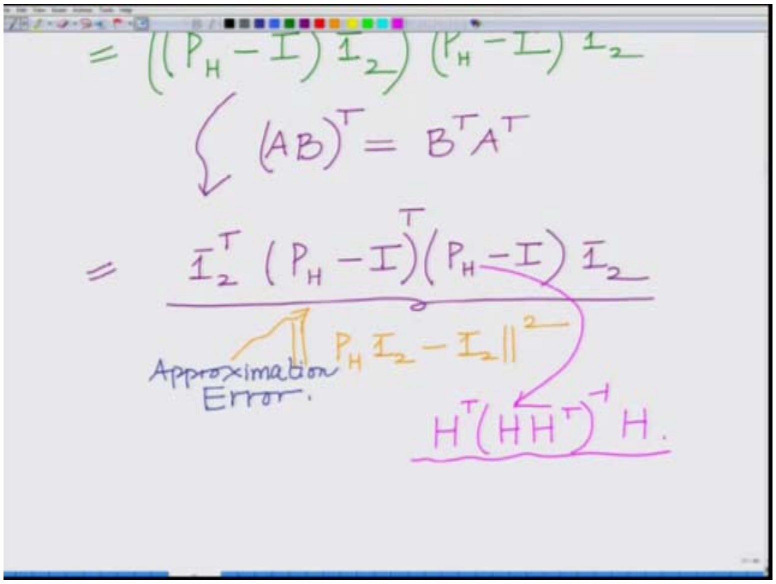


Now we can use another property here, that is, if I have two matrices A, B, the transpose of the product AB transpose, is basically B transpose A transpose. So I can use this property here, to simplify this as follows. This will basically be 1 bar 2 transpose times, that is this vector 1 bar 2 transpose, transpose PH minus I times PH minus I transpose times PH minus I times 1 bar 2, and therefore, we have simplified this. This is nothing but your norm of PH times 1 bar 2 minus 1 bar 2 whole square, which is basically nothing but your 'approximation error'.

This is what we are calling the 'approximation'; this is what we're calling the 'approximation error', or so we're able to derive this expression for the, we're able to derive this expression for the 'approximation error', which is basically 1 bar 2 transpose PH minus I transpose times PH minus I times 1 bar 2, where this matrix PH itself, remember just to clarify again, we have defined this matrix as your H transpose H H transpose inverse times H. That is basically the definition of this matrix PH, okay?

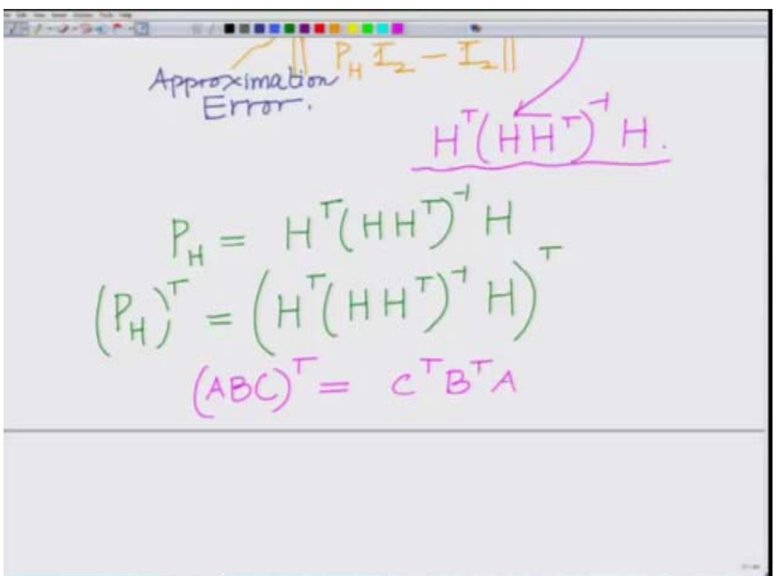
So, now what we want to do, is we want to simplify this expression, that we've derived for the 'approximation error', but before that, let us explore the properties of this matrix PH, alright? So before we do that, let us explore the properties of this matrix PH. And the matrix PH, you can see as several interesting properties, one, if you look at PH, PH equals H transpose H H transpose inverse H. Now if you look at PH transpose, the transpose of this matrix is H transpose H H transpose inverse H transpose. Now similar to previously, we have the result, if I have

(Refer Slide Time: 18:15)



three matrices A, B, C, and the product transpose, that is equal to C transpose, B transpose times A transpose.

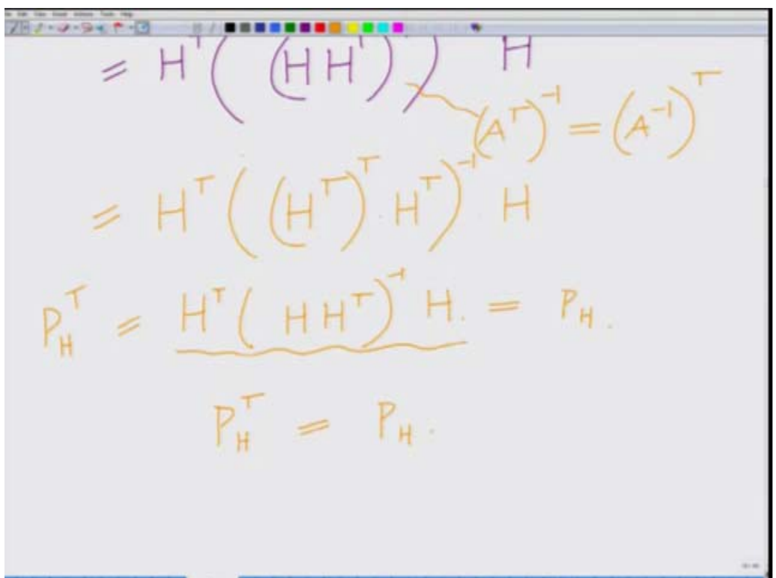
(Refer Slide Time: 19:13)



Therefore I can write this matrix as H transpose times H H transpose inverse, the transpose of this matrix times H transpose, transpose. Basically what we're saying is this is my matrix A, this is my matrix B, and this is my matrix P, so I'm basically writing C transpose, B transpose, A transpose. Now this is equal to H transpose. Now the transpose of an inverse is basically the inverse of the transpose. So basically I'm going to write this as H H transpose, I'm going to bring the transpose inside, and the inverse outside, and H transpose transpose is H. This is basically follows from the property A transpose inverse equals A inverse transpose, yeah?

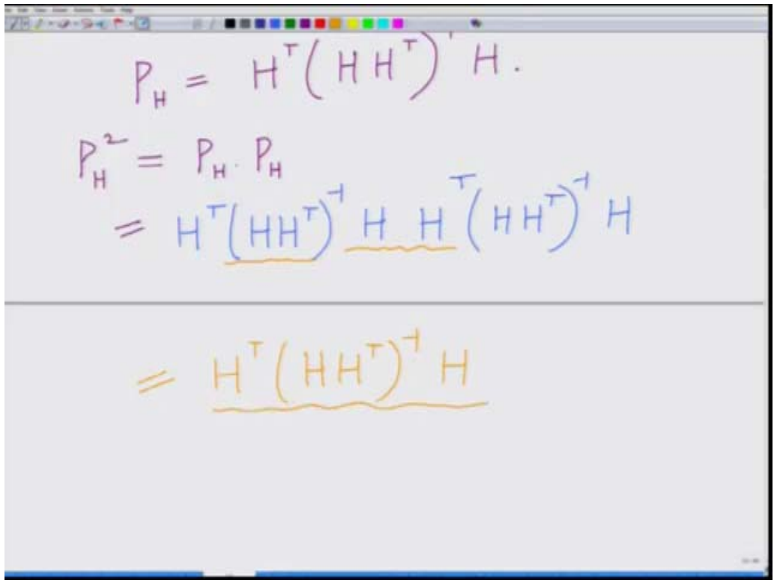
And now, I'm going to write this as H transpose, again look here I have AB transpose, which is B transpose A transpose, that is H transpose transpose times H transpose, into, your matrix H. Of course there is an inverse, and now I can write this as H transpose times H H transpose inverse times the matrix H, okay? So basically now, what I've shown is if I take the transpose of this matrix H, I again get back the same matrix which is in fact PH, that is, PH transpose equals PH. So we have this interesting property, that if I look at the transpose of the matrix, that is PH transpose, that is equal to PH itself, yeah?

(Refer Slide Time: 21:09)



So PH, so PH is the matrix, which has this property that (())(21:20) as transpose symmetry. PH transpose is equal to PH. In fact, let us look at another interesting property. Another interesting property that you can see is the following thing. Recall that PH is equal to H transpose H H transpose inverse times H. Now let us look at this quantity PH square. PH square equals PH times PH, which is equal to, which is equal to H transpose H H transpose inverse H, times H transpose, times H transpose H H transpose inverse H. Now if you look at this quantity H H transpose and H H transpose inverse, H H transpose inverse into H H transpose equals identity, so basically now, you have, this is equal to H transpose H H transpose inverse times H.

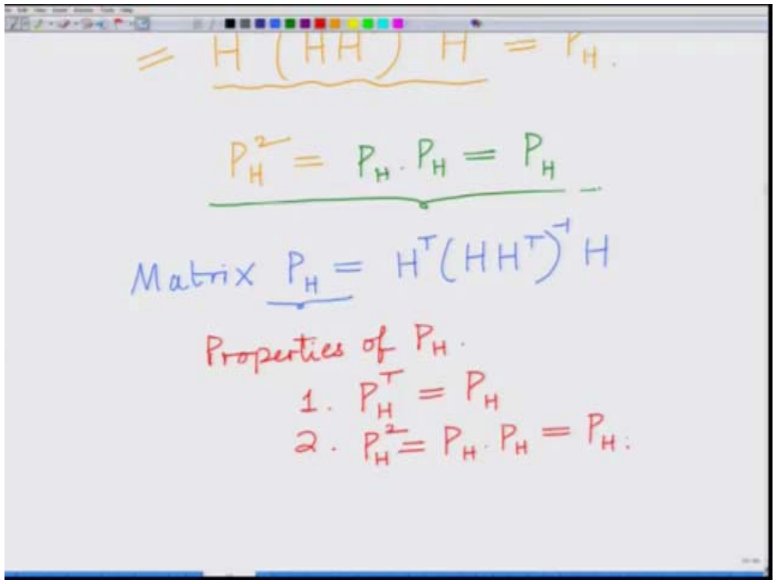
(Refer Slide Time: 22:34)



And now, if you look at this quantity once again, this quantity is nothing but PH. So we have PH square, that is basically PH times itself, PH into this matrix PH into PH is equal to PH, and this is an interesting property of this matrix PH, right? So PH has two interesting properties, one, is it's transpose symmetric, and, that is PH transpose is equal to PH, and the other interesting property is that, PH square, or PH times PH is equal to again, once again the same matrix PH. And this matrix PH is also known as the projection matrix, this is a projection matrix of H transpose.

So PH satisfies two interesting property, that is matrix PH, H transpose H H transpose inverse H, this satisfies the properties it satisfies, that is basically the properties are, the properties of this matrix. Properties of PH are one, it is transpose symmetric, PH transpose equals PH, and two, PH square, that is PH times PH equals PH, and this matrix PH is known as the projection matrix of H. This is known as the projection matrix of H transpose, this is the projection matrix of H transpose. So this matrix PH satisfies some interesting properties, this is known as the projection matrix of H transpose, alright?

(Refer Slide Time: 24:21)



So basically what we're doing in this module is we're using this equalizer, that is the least squares equalizer, that is c bar, which we have estimated as, umm, which for which we have derived the expression previously that is c bar equals H H, umm, transpose inverse times H into 1 bar 2. Now what we're doing is we're trying to find the 'approximation error' of this equalizer, that is, what is the error, that is how close is this equalizer H transpose c bar to the vector 1 bar 2? We're finding the norm square of this error, and we're simplifying, umm, the expression for the norm squared of this 'approximation error'.

And what we have, where we have come, so far that we've developed an expression of this involving this interesting matrix PH, PH, which is equal to H transpose times H H transpose inverse into H, we have said this is the projection matrix corresponding to H transpose. We've shown two interesting properties of this, one, that is PH transpose is equal to PH, two, PH square, that is PH into PH is equal to PH. We will use these two properties in the subsequent modules and we will simplify the expression for the 'approximation error', alright? So we will close this module here, we'll continue with other aspects in the subsequent modules. Thank you.