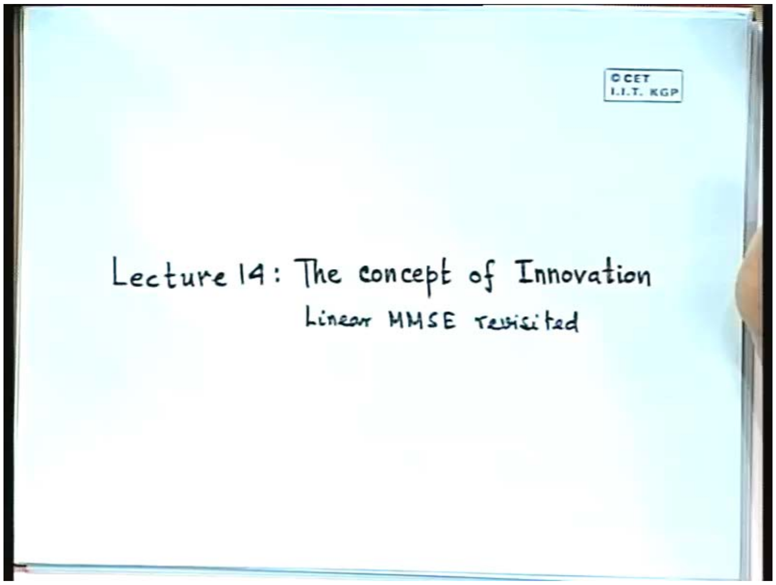
Estimation of Signals and Systems
Prof. S. Mukhopadhyay
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
The Concept of Innovation
Liner MMSE revisited

We made a digration we were we came to minimum, mean square estimation. First described, what it is, we will we will we will we will review it.

(Refer Slide Time: 0:52)



© CET
I.I.T. KGP

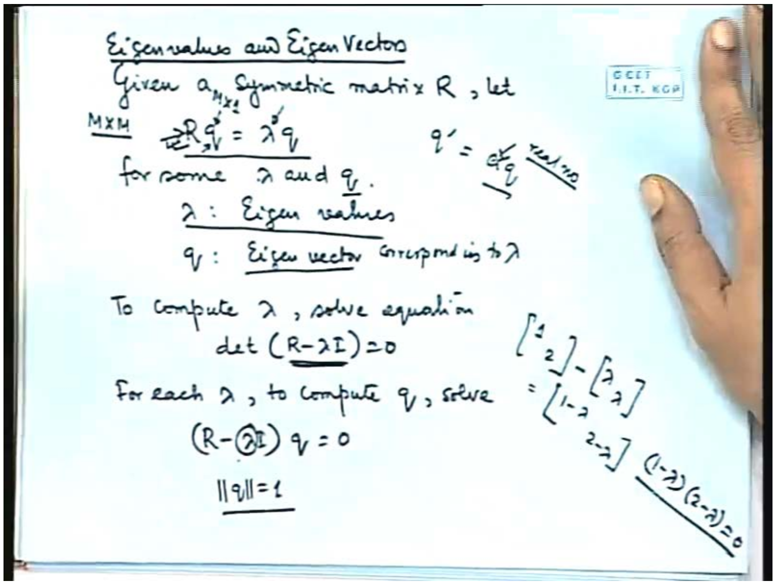
Lecture 14: The concept of Innovation
Linear MMSE revisited

And then we say the basic equations, and then suddenly realised, that to to to compute the optimal weight we need the correlation vector; I mean we need a correlation matrix. So we do not know how to get the correlation matrix, so we said that, okay let us stop here and we went back and found that, how to how to estimate the correlation matrix. And as an as a as a naturally consequence of that, we also saw how to estimate the Periodogram, right. And after that we had some confusion about Z- transforms, so we thought that, okay we will we will again review, these Z- transform convergence etcetera.

And we had also reviewed the concept of Eigen values, Eigen vectors. They were all with a purpose, the the basic purpose is that we want to go slightly deeper into linear minimum mean square estimation, because it is one of the very important and central points and and widely used for a variety of applications; from speech coding to target tracking, to I mean all sorts of applications match receivers for communication channels, in in in n number of applications, this this theory is used, right. So so we wanted to understand, we want to go a little bit inside and see you know, gain some analytical depth into it. So whatever we did was to get some you know, some of our basics again shaken up and so that now we can see.

So so we are now now we are coming back to the analysis, and this analysis in many cases will involve, this concept of innovation, okay. This I mean if we you will, we will see that, if we try to express things in terms of innovations, then the analysis becomes simpler and I mean easy to understand and and and its absolutely equivalent. I mean seeing so… so we need to.. need to understand this concept of innovations at this time, it it is the very central concept and lot of analysis is actually done in terms of it, right. So so so let’s get into it. First of all.. just you know, because we know that memory has a has a time constant, which may be short or long depends on people.

(Refer Slide Time: 03:12)



Eigen values and Eigen Vectors

Given a $_{M\times 1}$ Symmetric matrix R, let

MXM

$$R\underline{q} = \lambda \underline{q}$$

$q' = \alpha q$ real ≠ 0

for some λ and $\underline{q}$.

λ : Eigen values

$\underline{q}$: Eigen vector corresponding to λ

To compute λ, solve equation

$$\det(R - \lambda I) = 0$$

$$\begin{bmatrix} 1 & \\ & 2 \end{bmatrix} - \begin{bmatrix} \lambda & \\ & \lambda \end{bmatrix} = \begin{bmatrix} 1-\lambda & \\ & 2-\lambda \end{bmatrix}$$

$$(1-\lambda)(2-\lambda) = 0$$

For each λ, to compute $\underline{q}$, solve

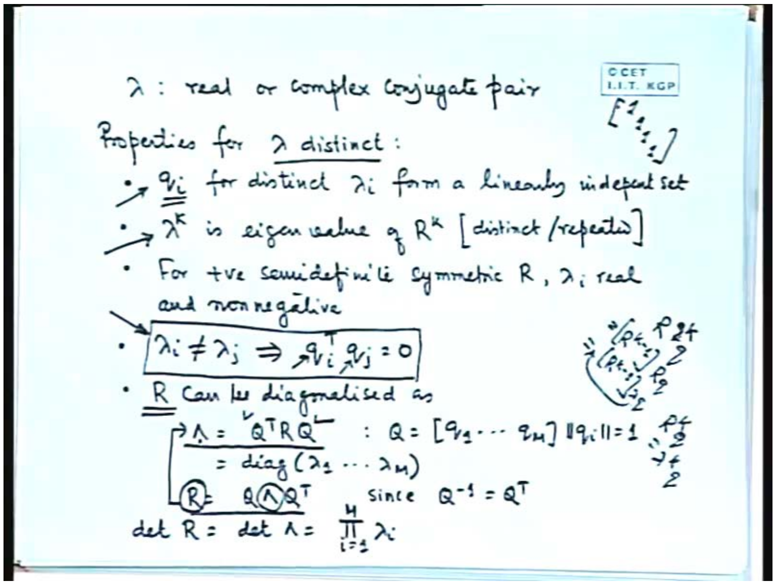
$$(R - \lambda I)\underline{q} = 0$$

$$\|\underline{q}\| = 1$$

So we are just we are just getting back to little bit recapitulation on on on Eigen values and Eigen vectors, we have done that. So we just just want to recall that, if you have a symmetric matrix R, then Rq is equal to lambda q, this is an equation; where q is a vector, R is a matrix, lambda is a number. So.. m into m into m into one becomes m into one and this is a number into an m into one, so dimension is matching. So the vector which satisfies this for some lambda, I mean this lambda is also not arbitrary, it is a it it is a particular lambda, then then that lambda is called a is actually called an Eigen value. That can be solved from this equation, determinant of R minus lambda equal to zero for example, here we had worked out an example, so if you have R as one two, just you know simple example.

Then this equation become like this, and we can solve it. It will become an nth order polynomial in lambda, so we can solve for lambda and we will get n roots. If we if it is an nth order polynomial, we will get n roots. So so corresponding to those lambda, if we solve this equation, they always solve this equation. This side now so this then we will be able to solve for this q, so this so this q, this q will be called the Eigen vector, okay. We need to understand, Eigen values and Eigen vectors and and some of their properties because this is widely used in our theory and and this properties are very interesting. This properties say that.. for for distinct Eigen values; which generally occurs, these Eigen values and Eigen vectors have very desirable properties.

(Refer Slide Time: 04:52)



λ : real or complex conjugate pair

© CET I.I.T. KGP

$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$

Properties for $\underline{\lambda \text{ distinct}}$:

- $\underline{\underline{q_i}}$ for distinct λ_i form a linearly indepent set
- λ^k is eigen value of R^k [distinct /repeated]
- For +ve semidefinite symmetric R, λ_i real and nonnegative
- $\boxed{\lambda_i \neq \lambda_j \Rightarrow q_i^T q_j = 0}$
- $\underline{\underline{R}}$ Can be diagonalised as

$\underline{\Lambda = Q^T R Q}$: $Q = [q_1 \cdots q_M]$ $\|q_i\| = 1$

$= \text{diag}(\lambda_1 \cdots \lambda_M)$

$\underline{R = Q \Lambda Q^T}$ since $Q^{-1} = Q^T$

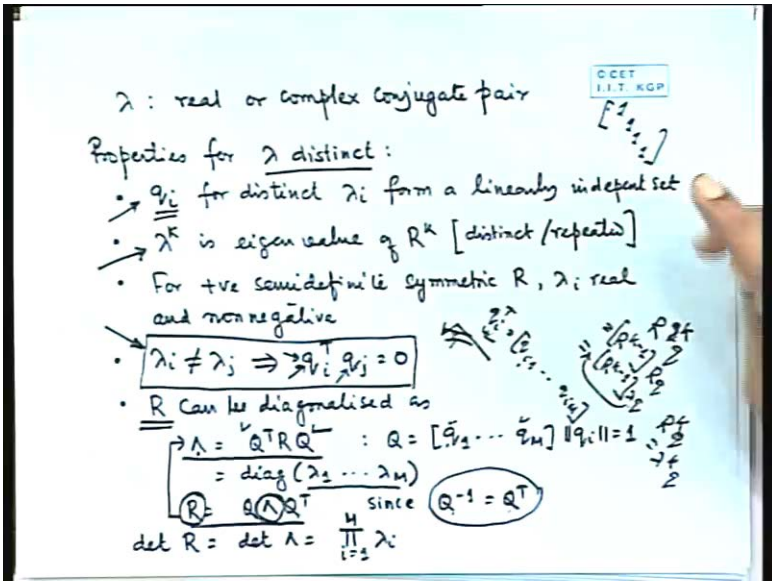
$\det R = \det \Lambda = \prod_{i=1}^{M} \lambda_i$

So one of the property is which is very key property is that, these each Eigen vector is actually, all the Eigen vectors are actually orthogonal to each other. They are actually perpendicular to each other. If you have a vector r and if you have another vector q in an end dimensional space, then how do you express the fact that r and q are perpendicular by execute an dot product? So this is nothing but a dot product of the vectors. So it says that the dot product is zero, right. So which says that, these Eigen vectors are actually perpendicular vectors; which immediately tells us that, we can define you know, this is we are actually talking about, we are actually talking about m dimensional vectors.

So we have an m dimensional space, now if you want to, it can be defined in terms of some arbitrary x y z coordinate axis, but what we are saying is that, the the moment we have found that, we have got this a, we have got this vectors in this space q's which are perpendicular to each other. That immediately tells us that we can probably define, we can probably define a coordinate system in terms of q's. Normally when we when we define a coordinate system, then we need a for example; to define a three dimensional coordinate system, you need three vectors which are, ideally speaking which are mutually independent, know linearly independent vector.

That will do but then just for having you know convenient, so that one does not have a component on the other, we want in to be perpendicular, so here we have got a set of perpendicular vectors in that space, so so that immediately tells us that, okay now we can choose;

(Refer Slide Time: 6:49)



λ : real or complex conjugate pair

Properties for λ distinct :

- q_i for distinct λ_i form a linearly independent set
- λ^k is eigen value of R^k [distinct / repeated]
- For +ve semidefinite symmetric R, λ_i real and non negative
- $\boxed{\lambda_i \neq \lambda_j \Rightarrow \vec{q}_i^T \vec{q}_j = 0}$
- R can be diagonalised as

$$\Lambda = Q^T R Q : Q = [\vec{q}_1 \cdots \vec{q}_M] \; \|q_i\| = 1$$

$$= \mathrm{diag}(\lambda_1 \cdots \lambda_M)$$

$$R = Q \Lambda Q^T \quad \text{since} \quad Q^{-1} = Q^T$$

$$\det R = \det \Lambda = \prod_{i=1}^{M} \lambda_i$$

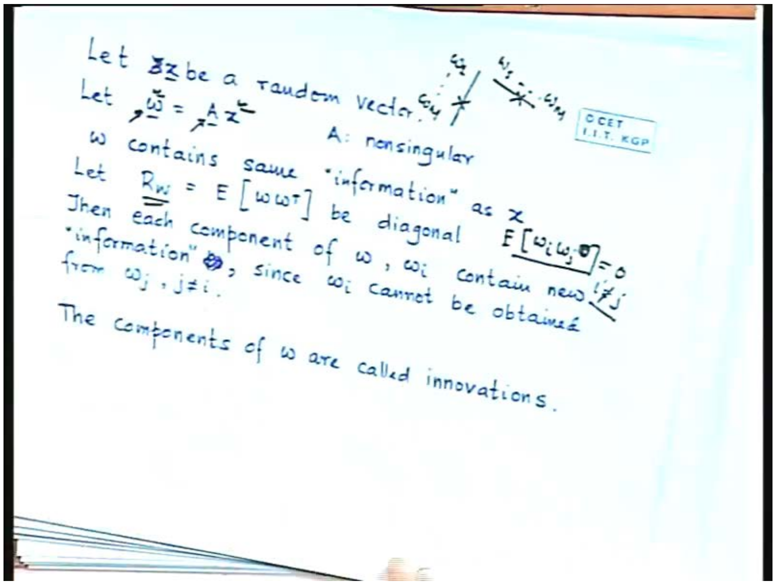
Suppose we choose these as our coordinate axis, do we get some advantage? We get a lot of advantage, as we will see, right. So so remember that we will be very interested in using these as our coordinate axis and given any random vector describe them in terms of these vectors, right. We we will do that very often, now. And if you I mean just just as the consequence of these that, these are orthogonal. You see you you have these two results, which says that the matrix R can be broken up like this, where this what are this q's? This q's are basically a matrix whose columns are are these vectors. So this is q one to q m, you can imagine q is a q is an m into m matrix q is a square matrix. So it has m columns, so each column is a vector and this vectors of the Eigen vectors.

So if you take the Eigen vectors and put them one by one and then make a matrix that is q. And so now R can be always expressed as, Q gamma, Q transpose. But this gamma is going to be a diagonal matrix that is very advantage; that is that is going to turn out to be a very advantageous thing for us, because gamma is going to be diagonal, that we will use in n number of I mean we will use it to a lot of advantage, but but just let us remember that at this point, there are every interesting properties of this matrix. That first of all any matrix R can be now broken up into this form, where gamma will be a diagonal matrix and not only any any diagonal matrix, the the elements of gamma will be the Eigen values.

So gamma will be these, it will be lambda one, lambda two, lambda three, lambda four, as a diagonal matrix. And not only that this matrices are such that, Q inverse equal to Q transpose. It it it turns out again, because of these, just multiply Q by Q inverse, so Q Q inverse should be equal to the should be the should be the identity matrix. Now you multiply Q, Q transpose. So so if you multiply Q, Q transpose you will get terms like this. There is a ijth element of Q, Q transpose will be terms like these. So if you so then immediately, you can see that that that all the off all the off diagonal terms for which i and j are different, if they are going to be zero and and if you choose choose the length of i as one, that is norm of that is that is Q is a vector, Qi is the vector it has n components.

So you just choose them such that, root over q one square, q one one square, qi one square, qi is a vector which has elements qi transpose q is a qi is a column vector, so qi one up to qi m, these are these are the elements of this vector, these are these are numbers. So if you choose this this such that root over qi ones, sorry you cannot see perhaps. That is just see that, the what is a length of this, root over x, one square plus x, two square plus x, three square that is the length of this vector. See if you choose the length to be one, then then then all the qi, qi transpose terms, will give you one. So it will be an identity matrix that, therefore Q inverse will be equal to Q transpose, all are properties of this thing, this is the key property, right. So these are some of the properties of these these Eigen vectors, which are very useful for us, which are going to be. So so so we will now, use this properties to our advantage.

(Refer Slide Time: 10:47)



Let $\underline{x}$ be a random vector

Let $\underline{\omega} = \underline{A}\,\underline{x}$ A: nonsingular

ω contains same "information" as x

Let $\underline{\underline{R_w}} = E[\omega\omega^T]$ be diagonal $\underline{E[\omega_i\omega_j]} = 0$ $i \neq j$

Then each component of ω, ω_i contain new "information", since ω_i cannot be obtained from ω_j, $j \neq i$.

The components of ω are called innovations.

So first of all let us try to understand, the the what is innovations? See in English innovations means what it means that, I mean you you actually generate something new, something which was not existing, right. I mean I remember I I read the, what is the difference between discovery and and and oh no that was invention, that was different. So in any case, I mean innovation means, which is something new which has been generated, which was not there, right. So here, so in this sense the word is used, so why the word innovation has been used? That is because, so let’s try to understand this. Suppose x is a random vector of any length, it has some components x one, x two, x is a vector, so it has components.

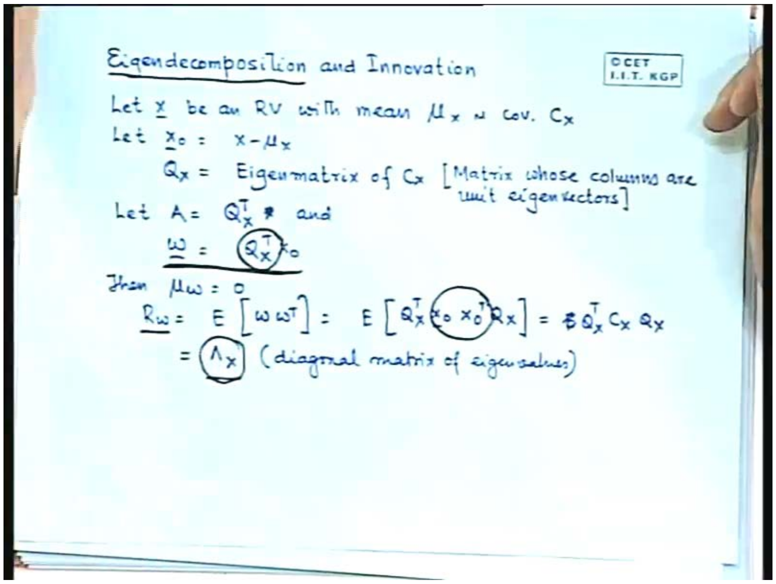
Now suppose from x multiplying by a matrix A, I can generate a vector w. one vector multiplied by another vector will give you one matrix, will give me another vector, right. Now in a sense these w's have been generated from x, so they contain they so there is there is no new whatever information is there in x, you have used that to generate w. So so w contains the same information as x, you can say. Now if you can generate this w in such a manner that, this Rw is diagonal, expectation of w w transpose is diagonal. What does that mean? It means that, wi it means that, what are the what are the elements of this? The elements of these are, that is ijth element of this matrix will be expectation of wi wj transpose, is it not? W what is, what is the, this is the vector, this is the vector actually, this transpose actually, wi wj suppose w has elements w one up to wn, here also w one up to wn. When you multiplying a vector, column vector by a row vector. So what will be the i? So this will give you a matrix? What is ijth element of this matrix? It is wi multiplied by wj, so expectation of the ijth element of Rw is… is this.

Now we are saying that that Rw is diagonal, what does it mean? That wi wj is equal to zero, for i, not equal to j, that is what we are saying. That means that the element of w are uncorrelated, which means that, now each element has no bearing with the other, each component, so as if each component carries some new information, which cannot be obtained from the other. So whatever information wi holds, it cannot be generated from wj, all the others because, because wi has has no relationship with any other wj. So it contains some unique information, which is new, which is not there in other components.

That is why the that is why these vector w will be called called an innovations vector; where each component of it will will now see see it, the elements of x, were kind of mixed up. That is x one may have some correlation with x two may have some correlation with x three, it may happen. But which means that, the the information that that x one contains at this part of it can be generated from x two, because it is correlated with x two but that does not happened to in the case of w. So by transforming x, I have generated a vector w, where the information is kind of compartmentalized some information in, some information in w one and some information in w two, some in w three. Each of them are new new information, one cannot be generated from the other. So in that sense w is the the elements of w are the innovations, okay.

We will presently see what is the advantage of what is the advantage of dealing with such w? So there are two facts that we must remember that, as far as the information concern as far as the information is concerned; this vector w is perfectly equivalent to the total information of x, because it has been generated from x. So all that in the information that is contained, I am using the term information in a vague sense, but I mean, I have not defined the term information, right. I mean mathematically, but what I am saying is that in a in a statically average sense, components of x can be generated from other components of x in that sense that, there is there is overlap of information, okay. So all I am trying to say is that, you have been able to generate a vector which contains overall contains, the same information as that of x, but now the components of the vector are each one contains new new information, by transforming it by a matrix. This is what what I have done. Now let us see what is the advantage. Now, first of let us see how to generate this a, I mean I have not defined this a, I mean I mean I have not told, how what a will actually achieve this. So now we will see that, this a will be again in the terms of the Eigen vector matrices.

(Refer Slide Time: 16:37)



Eigendecomposition and Innovation

O CET
I.I.T. KGP

Let $\underline{x}$ be an RV with mean μ_x & cov. C_x

Let $\underline{x}_0 = x - \mu_x$

$Q_x =$ Eigenmatrix of C_x [Matrix whose columns are unit eigenvectors]

Let $A = Q_x^T$ and

$\underline{\omega} = \boxed{Q_x^T} x_0$

Then $\mu_\omega = 0$

$\underline{R_\omega} = E\left[\omega \omega^T\right] = E\left[Q_x^T \boxed{x_0 x_0^T} Q_x\right] = Q_x^T C_x Q_x$

$= \boxed{\Lambda_x}$ (diagonal matrix of eigenvalues)

So now, so this is sometimes called an Eigen decomposition that is, if you decompose a vector in terms of the Eigen vectors. So let let x be a let x be a random variable. See you can assume that, it is with some mean; in if you assume it, it has some mean. Then you have to define a new variable which is x minus m u x, transform it to a zero mean variable or you can start with a zero mean variable, either way it is okay. And c x is the covariance matrix of x. Now so c x is like my R x, I just used the terms c, because x I assumed with a non-zero mean. So I should say covariance rather than correlation, but c x is nothing but correlation of x naught, same thing.

Now so that is my Rx, that I that I consider. Now I can find out the Eigen values and the Eigen vectors of Rx, and I can stack those Eigen vectors to make this matrix Qx. This is the Q matrix that, I was talking about, which consist of the Eigen vectors of R, I have stacked them up.

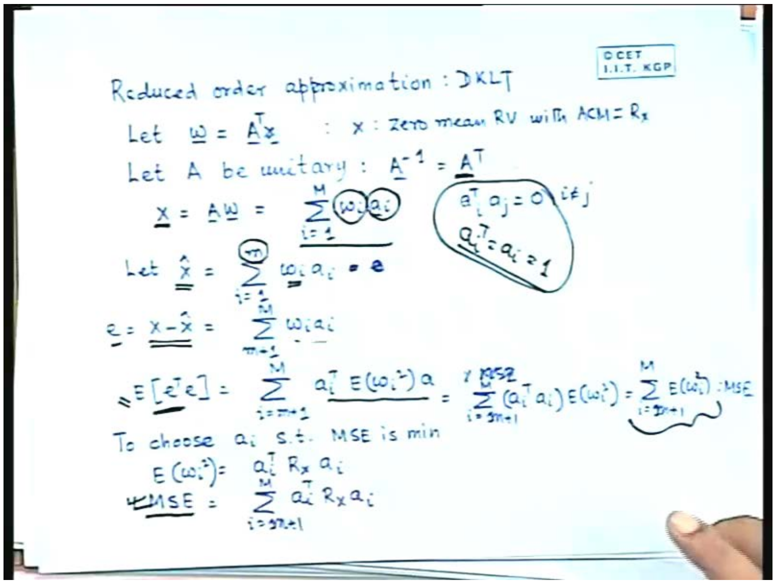
And then now now if I, I will show you that if I make a transformation like this, that is if I choose this as the A matrix, then Rw will become diagonal. That is simple because if I define this then, what is expectation of w w transpose? it is simply this. This is my R R x, zero x, zero transpose or c x whatever. So now I have Q x transpose, R x, Q x, which is nothing but gamma x. That is my previous relationship. So so you see that if I change, if I now transform the by this Eigen vector matrix, I get the innovations, right. That is why we needed to know, what are what are Eigen vectors?

Now, okay, so I have been able to generate innovations, so what? Why should I generate innovations? What is the advantage? There are various kinds of advantages. Some, one big advantage is analytical simplicity, but there is some there are there is some practical advantage, rather than you know conceptual advantage. That is, so I will show you one of the advantages which is very much used in what is known as coding, okay. What do you do in coding? Suppose you have a signal, right. Suppose you are saying that, a a I mean we know that any signal can be represented as, what is known as a in terms of a set of standard signal. For example, we try to represent a signal in terms of sine waves, so sine waves are our standard signals. And we try to express these signal in terms of a sum of sine waves, right. Sine waves are not the only ones; there are many other kinds of waves, many other kinds of what are known as basis functions, in terms of which you can break up a signal, okay.

Now suppose I am saying, suppose I have I have decided suppose, I have a think of a transmitter and a receiver. So I have to transmit a a signal x from here to here, okay. Now I have some bandwidth constraints; that is I want to reduce the amount of data that, I want to transfer. This is the very important requirement. Suppose you want to have a video conference, so you have to send real time image data over the network, each image data is going to be a say, one thousand twenty-four by one thousand twenty-four pixels. So huge amount if you if you want to send the raw data, huge amount of data will have to be send and its completely I mean it is not necessary at all. Because, especially because in because in video conferencing, you are not really you know trying to measure, what is the distance from the left eye ball to the tip of nose, these kind of measurements, you are not doing. The who who is who you ever will see it? He should be he or she should be comfortable, should be able to understand the speech and the motion that is all reasonable accuracy.

So firstly remember that, there is no need to to to transfer each pixel for this application. Now the so so now come back to this question that I want to transfer a signal from here to here. I do not want to transfer everything. I want to transfer some approximation of it, and I want to using those approximations, I want get the best accuracy once I, once I billed up the signal there. So I want to send something some data from here, I want to code the signal. Sent the codes over the network and then using the codes, I want to decode it at the receiving end and get back my signal. This is a this is a very important requirement in communication. So now suppose, that I am trying to trying to send, suppose I have a signal x, which I want to send, okay.

(Refer Slide Time: 21:52)



D CET
I.I.T. KGP

Reduced order approximation : DKLT

Let $\underline{w} = \underline{A}^T\underline{x}$: x : zero mean RV with ACM = R_x

Let A be unitary : $\underline{A}^{-1} = \underline{A}^T$

$\underline{X} = \underline{A}\underline{w} = \sum_{i=1}^{M} w_i a_i$

$a_i^T a_j = 0$ if $i \neq j$

$a_i^T a_i = 1$

Let $\hat{x} = \sum_{i=1}^{m} w_i a_i$

$e = x - \hat{x} = \sum_{m+1}^{M} w_i a_i$

$E[e^T e] = \sum_{i=m+1}^{M} a_i^T E(w_i^2) a = \sum_{i=m+1}^{M} (a_i^T a_i) E(w_i^2) = \sum_{i=m+1}^{M} E(w_i^2)$: MSE

To choose a_i s.t. MSE is min

$E(w_i^2) = a_i^T R_x a_i$

MSE $= \sum_{i=m+1}^{M} a_i^T R_x a_i$

Now, signal can be anything. So it is very difficult to, so we can assume that, it is going to be a random variable, for example speech signal. Have you seen speech signals? they are very much look like random variable. So I have a so the now the question is that, suppose x is the signal and it can be represented on a set of basis vectors. So suppose x is a random vector, it can be broken up into some coordinate vectors. So these are the my coordinates and these are my coordinate vectors. I can always break it up. Suppose x is such that, you have i is equal to one to m, that is x can be broken up into a set of such m components, okay.

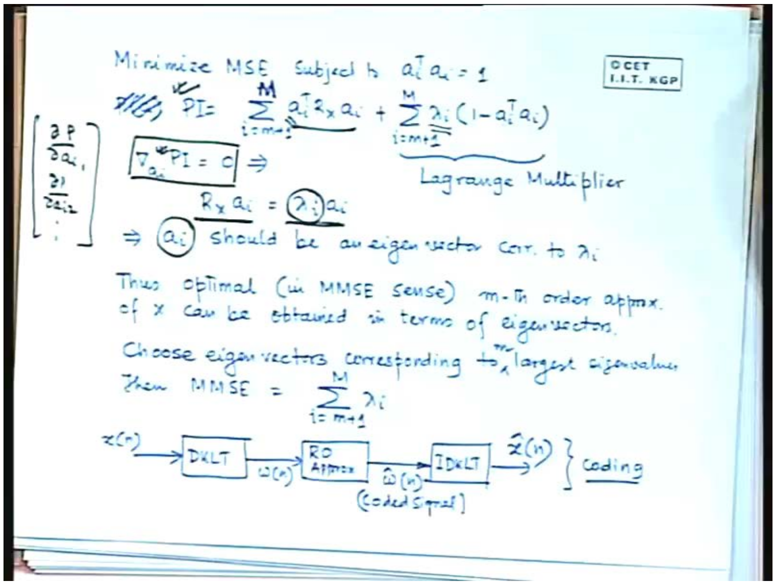
Now the point is that, rather so so what you can do, what you can do is, if you know the suppose the suppose the transmitter and the receiver at at at the both ends, if this vectors ai's are known; if they have decided, that that that I am going to use the same coordinate vectors. Then I do not need to transfer x, but rather I can transfer this wi's. I can send this w'is then, at the receiver end they will calculate this sum and they will make x. Now the point is that, what is that what is the advantage of sending the wi's over this x? No advantage because x contains m coefficient, this also contain m coefficient. So I am not getting anything, so I want to send less number of wi's, right.

So suppose, I want to send small m number of wi's, so then what will be the receiver do? Receiver will have to do with this small m number of wi's it has receive. So it may have to reconstruct it, using this. Now less number of terms, and it will in general, construct an approximation of x. And this will be the error. Now I want to choose this ai's such that this error is the least, this is a very very valid problem. So I want to send suppose, suppose the vector can be actually represented in terms of three hundred components; I want to send only thirty and yet I must choose my basis vector, in such a manner, that yet when I reconstruct, I will get the least amount of error, what I can do? So so what is the error? Error is nothing but i equal to one to m is there. So i equal to m plus one to m, that is rest of the things that is there I did not send. Now I find the power of the error, that that that turns out to be this a i transpose this.

This A matrix, we will definitely choose as what is known as an unitary matrix. That is A inverse equal to A transpose… Note that in our case, Q inverse was equal to Q transpose, but so so the Eigen vector matrix is a unitary matrix; but there can be other unitary matrixes, which are not Eigen vector matrices. So so... this I am doing because, I want to I mean I can get this wi's very easily, just just by inverse. That is if w equal to A transpose x, x will be equal to Aw, because A transpose is I, so just by transposing, I can I can do it. So now it turns out that, its expectation of e transpose e becomes this one. That is so the error is actually, the power of those components, which I did not send, right.

Now so now but but then but then this.., what is my what is my original problem? My original problem is that, I want to minimize this MSE. I am given x, I want to minimize this MSE and I want to choose this ai's, that is my, that is my problem, okay. Subject to this this this requirement, that ai transpose ai is equal to one and ai transpose aj is zero. Subject to these constraint, I want to minimize this function. So it is a constraint minimization, problem. Do you remember what we do, when we have a performance function and we want to minimize it and we have some constraints? We use, what is known as Lagrange multipliers, basically it is a it is a it is a way of see that is we write the we we converge the constraint optimisation problem into an unconstrained optimisation problem, by adding these additional terms.

(Refer Slide Time: 26:43)



Minimize MSE subject to $a_i^T a_i = 1$

$$PI = \sum_{i=m+1}^{M} \underline{a_i^T R_x a_i} + \underbrace{\sum_{i=m+1}^{M} \lambda_i (1 - a_i^T a_i)}_{\text{Lagrange Multiplier}}$$

$$\begin{bmatrix} \frac{\partial P}{\partial a_{i_1}} \\ \frac{\partial I}{\partial a_{i_2}} \\ \vdots \end{bmatrix}$$

$\boxed{\nabla_{a_i} PI = 0} \Rightarrow$

$$\underline{R_x a_i} = \underline{\boxed{\lambda_i} a_i}$$

$\Rightarrow \boxed{a_i}$ should be an eigenvector corr. to λ_i

Thus optimal (in MMSE sense) m-th order approx. of x can be obtained in terms of eigenvectors.

Choose eigenvectors corresponding to m largest eigenvalues

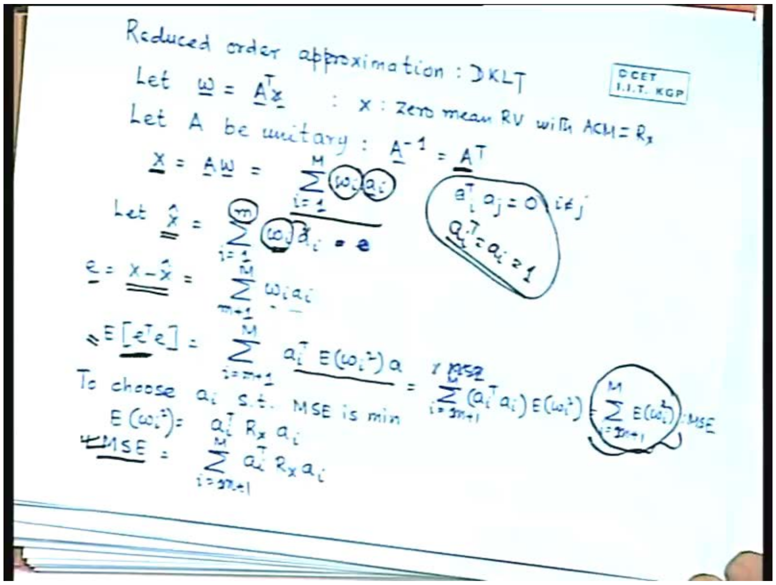
Then MMSE $= \sum_{i=m+1}^{M} \lambda_i$

These are these are these will give us and then, I will differentiate it with respect to these which I want to optimise. I want to find out a i and I will synthetically, that is artificially so I actually I should not use this you know this here, lambda has been used, because lambda turns out to be an Eigen value, otherwise normally can be anything. So we should, if we differentiate it with respect to these variables, we will we will get back the constraint equations. So which means that, the that the this PI equal to zero which is my unconstraint. If you if I want to optimize something with respect to a i, what I have to do? I have to differentiate with respect to i. This is nothing but a vector, this means that this is nothing but you know delta p, by delta a i one, delta p by delta a i two, this is a vector. p is a scalar, I am differentiating a scalar with respect to a vector, so what does it mean? It means that, if the components of the vector a i are a i one, a i two, a i m then this term, this is this is nothing but a notation. This notation implies this vector, delta p by delta a i one, delta p by delta a i two and so on. This vector is is denoted by this symbol.

So we have to find that, this equal to zero solution. And not only that, we have to so we have to solve for this and these, because these are my constraint equations. If you just recalling Lagrange multiplier. So if you differentiate it, you can you can see it is you can, if you want you can take a three by three vector and actually, write this term and then differentiate. If you can if you want to use scalar theory. But it turns out that, if you differentiate these, this term gives you R x a i and this term will you lambda a i. This is like x square, okay.

So so now you see that, if if that is so, what does what is this equation? This is that this is that old Eigen value Eigen vector equation if you see so, which means that for minimum mean square error, these should be Eigen vectors and these should be Eigen values, right. So so now see the importance, that if you want to send less number of components and still commit the commit the least amount of error, then you should decompose. When you are sending those those those wi's you should do it, on the basis of Eigen vectors.

(Refer slide Time: 29:44)



Reduced order approximation : DKLT

DCET I.I.T. KGP

Let $\underline{w} = \underline{A}^T\underline{x}$: x : zero mean RV with ACM = R_x

Let A be unitary : $\underline{A}^{-1} = \underline{A}^T$

$\underline{x} = \underline{A}\underline{w} = \sum_{i=1}^{M} w_i a_i$

$a_i^T a_j = 0$ $i \neq j$

$a_i^T a_i = 1$

Let $\hat{x} = \sum_{i=1}^{m} w_i a_i$

$\underline{e} = x - \hat{x} = \sum_{m+1}^{M} w_i a_i$

$E[e^T e] = \sum_{i=m+1}^{M} a_i^T E(w_i^2) a = \sum_{i=m+1}^{M} (a_i^T a_i) E(w_i^2) = \sum_{i=m+1}^{M} E(w_i^2)$: MSE

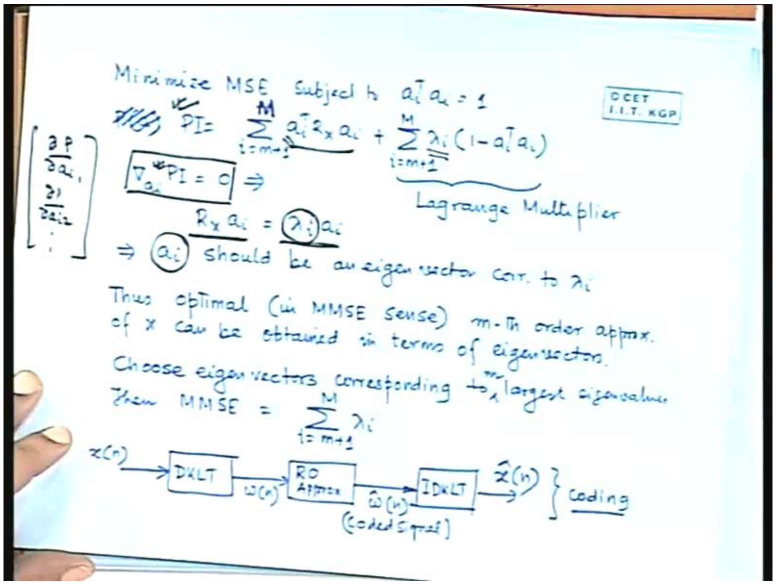
To choose a_i s.t. MSE is min

$E(w_i^2) = a_i^T R_x a_i$

$\text{MSE} = \sum_{i=m+1}^{M} a_i^T R_x a_i$

Then if you, now which which Eigen vectors, obviously you should send it such that such that, this sum is the least, is it not? So obviously you should send those those components; see each component w i is corresponding to a particular Eigen vector, so it will turn out that you should send those components which correspond to the largest Eigen vectors.

(Refer Slide Time: 30:07)



Minimize MSE subject to $a_i^T a_i = 1$

DCET I.I.T. KGP

$PI = \sum_{i=m+1}^{M} a_i^T R_x a_i + \underbrace{\sum_{i=m+1}^{M} \lambda_i (1 - a_i^T a_i)}_{\text{Lagrange Multiplier}}$

$\begin{bmatrix} \frac{\partial P}{\partial a_{i1}} \\ \frac{\partial I}{\partial a_{i2}} \\ \vdots \end{bmatrix}$

$\boxed{\nabla_{a_i} PI = 0} \Rightarrow$

$R_x a_i = \lambda_i a_i$

$\Rightarrow$ a_i should be an eigen vector corr. to λ_i

Thus optimal (in MMSE sense) m-th order approx. of x can be obtained in terms of eigenvectors.

Choose eigen vectors corresponding to m largest eigenvalues

Then $\text{MMSE} = \sum_{i=m+1}^{M} \lambda_i$

$x(n) \rightarrow$ DKLT $\xrightarrow{w(n)}$ RO Approx $\xrightarrow{\hat{w}(n)}$ IDKLT $\rightarrow \hat{x}(n)$ } Coding

(Coded Signal)

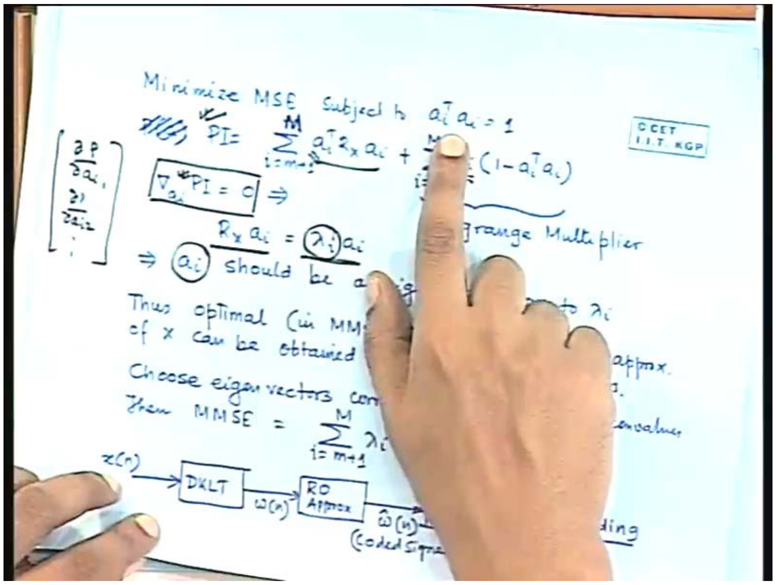
So send those components, so first send the largest Eigen vector component, second next send the large largest Eigen vector means that; Eigen vector which corresponds to the largest Eigen value. See the Eigen values are all real, because R is a symmetric matrix symmetric positive definite matrix. So therefore, you can order order the Eigen values. So, one will be the largest, then the next, then the next, then the next and so on. So first send that component, which corresponds to the largest Eigen value, then send the next one, then send the next one. Then whatever finally, what whatever will remain will be the will be the once, which are which are whose power, that is the that the power of the error will be proportional to the power of the remaining Eigen values, that is what it says.

So you see that, this first thing is that this Eigen value Eigen vector decomposition is good for making a best reduced order approximation of a signal, if it really needs a large number of coefficients, and if you.. if you want to approximate it, using a small number of coefficients then you should always be done based on an Eigen decomposition, that is the first important thing.

Student<sir A transpose is equal to ai transpose equal to>
[Conversation between Student and Professor – Not audible ((00:31:33 min))]

a i transpose a i equal to one, length of the vectors should be equal to one. a i is a column vector, so a i transpose a i means what, a i one square plus, a i two square plus, a i three square plus, a i four square so is like that.

(Refer Slide Time: 31:49)



Minimize MSE subject to $a_i^T a_i = 1$

$PI = \sum_{i=m+1}^{M} a_i^T R_x a_i + \sum_{i=m+1}^{M} \lambda_i (1 - a_i^T a_i)$

Lagrange Multiplier

$\begin{bmatrix} \frac{\partial P}{\partial a_{i1}} \\ \frac{\partial I}{\partial a_{i2}} \\ \vdots \end{bmatrix}$

$\boxed{\nabla_{a_i} PI = 0} \Rightarrow$

$R_x a_i = \lambda_i a_i$

$\Rightarrow a_i$ should be a eig… to λ_i

Thus optimal (in MM… of x can be obtained … approx.

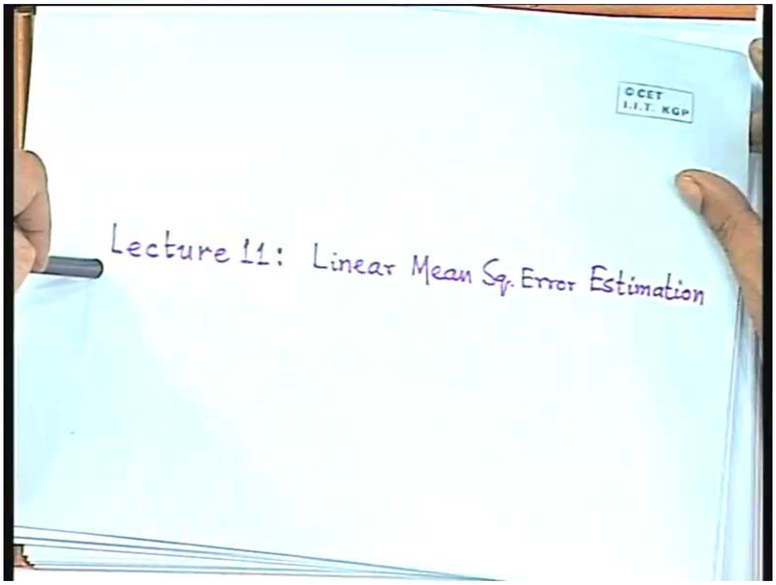
Choose eigen vectors corr… …envalues

Then MMSE $= \sum_{i=m+1}^{M} \lambda_i$

$x(n) \rightarrow$ DKLT $\xrightarrow{w(n)}$ RO Approx $\rightarrow \hat{w}(n)$ (Coded Signa… …ding

a i transpose a i equal to one, see this one, okay. So now let us come back this is this our lecture eleven, if you see where; we studied minimum mean square estimation.

(Refer Slide Time: 31:57)

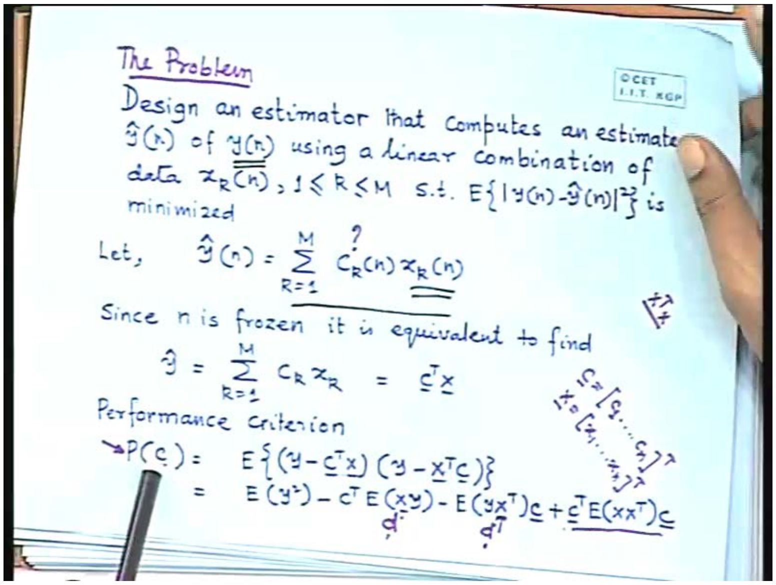


Lecture 11: Linear Mean Sq. Error Estimation

CCET
I.I.T. KGP

Now we can have some more inside into that, so let’s recapitulate that little bit, so what did we say? We said that, we have samples of some things x k and we have another thing y, which we may not able to measure, but we want to estimate y based on x, using linear estimators such functions we are not going to use logarithm, something. See we are not using those functions, okay.

(Refer Slide Time: 32:13)



The Problem

Design an estimator that computes an estimate $\hat{y}(n)$ of $\underline{\underline{y(n)}}$ using a linear combination of data $x_R\overline{(n)}$, $1 \leqslant R \leqslant M$ s.t. $E\{|y(n)-\hat{y}(n)|^2\}$ is minimized

Let, $\hat{y}(n) = \underline{\sum_{R=1}^{M} \overset{?}{C_R}(n)\, \underline{\underline{x_R}}(n)}$

Since n is frozen it is equivalent to find

$$\hat{y} = \sum_{R=1}^{M} c_R x_R = \underline{c}^T \underline{x}$$

$\underline{c} = [c_1 \ldots c_M]^T$

$\underline{x} = [x_1 \ldots x_M]^T$

Performance criterion

$$\rightarrow P(\underline{c}) = E\{(y-\underline{c}^T\underline{x})(y-\underline{x}^T\underline{c})\}$$

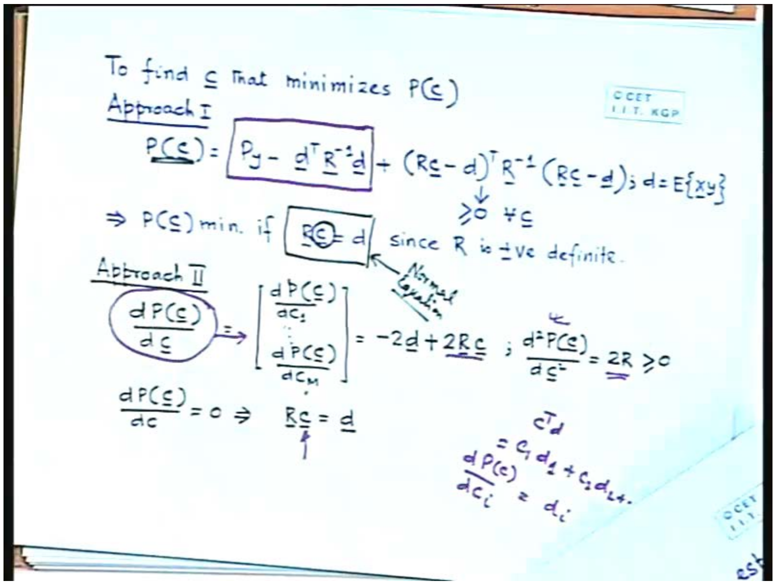
$$= E(y^2) - c^T \underset{\underline{d}}{E(\underline{x}y)} - \underset{\underline{d}^T}{E(y\underline{x}^T)}\underline{c} + \underline{\underline{c}^T E(xx^T)\underline{c}}$$

So what is the best, so ideally speaking if I say that, okay okay I can use maximum m number of components then, this is one way of saying that. And here I have assumed that, this is a you know here I have assumed that there are k number of that is x is itself a vector, so it is slightly more complicated but but the but the basic problem is that, you are given a set of, you are given some signals x, based on its samples you want to estimate some other signal y, using linear functions, such that the estimation that the minimum mean square error in estimation occurs. This is the linear minimum mean square estimation problem. You are given one random process, you want to estimate another random process based on the samples of this random process, using linear estimators; such that the mean square error is minimum.

So there are in any estimation setting, there are three quantities, three quantities. One are the sets of random processes, that is you have some observations, you want to estimate something else, so what you have and what you want to estimate? Then you have some some some structures of estimators, you do not say, among all possible a among all possible estimators; that is not useful because, the the the the I mean set of all possible estimators is so vast that you can never explore it. So we so we so we will restrict ourselves… to some kind of estimators, in this case we are considering linear estimators, all right. And we is define a performance criterion, that is how do we us decide whether estimator A is is is better than estimator B, are or not?

So we say that, the our performance criterion is mean square error, should be minimum. So you have a model set that is an estimator structure you have a performance criterion and you have some signals, for any estimation settings these things are required, okay. So what did we see? So we load the performance criterion like this; see this is the error, so error error transpose. So I am so this is my expectation of error, transpose should be minimum and then we said that, then we before so we we found that that, this will be minimum if this equation is satisfied, okay.

(Refer Slide Time: 34:59)



To find $\underline{c}$ that minimizes $P(\underline{c})$

Approach I

$$P(\underline{c}) = \boxed{P_y - \underline{d}^T \underline{R}^{-1} \underline{d}} + (R\underline{c} - d)^T \underline{R}^{-1} (\underline{R}\underline{c} - \underline{d}); \; d = E\{\underline{x}y\}$$

$$\geq 0 \;\; \forall \underline{c}$$

$$\Rightarrow P(\underline{c}) \text{ min. if } \boxed{\underline{R}\underline{c} = d} \text{ since } R \text{ is } +\text{ve definite.}$$

Normal Equation

Approach II

$$\frac{dP(\underline{c})}{d\underline{c}} \Rightarrow \begin{bmatrix} \frac{dP(\underline{c})}{dc_1} \\ \vdots \\ \frac{dP(\underline{c})}{dc_M} \end{bmatrix} = -2\underline{d} + 2R\underline{c} \; ; \; \frac{d^2P(\underline{c})}{d\underline{c}^2} = 2R \geq 0$$

$$\frac{dP(\underline{c})}{dc} = 0 \Rightarrow \underline{R}\underline{c} = \underline{d}$$

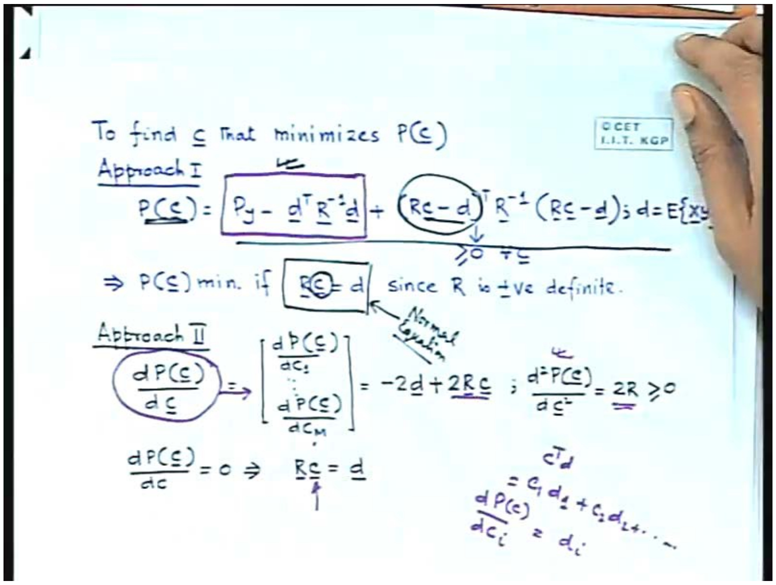
$$c^T d = c_1 d_1 + c_2 d_2 + .$$

$$\frac{dP(c)}{dc_i} = d_i$$

This is sometimes called the normal equation, it is sometimes called the normal equation, which you which you which you have to solve, if you have to get the optimal weight c. Then you will get the minimum value of these performance criteria. Very simple, you can obtain it by just casting it like this or you can obtain by differentiating, either way it is okay. Now, we also found out that if you use the optimal estimators, then also you will get some error. It is not that, if you use the optimal estimator, you will get zero error.

You do not, never get zero error; you get some error, but the but the properties of the optimal estimator is such that, the properties of the optimal estimator is such that, you have what is known as the principle of orthogonality, that is the that is the error components are going to be perpendicular to the estimates, that we had also seen, okay. We will we will come back to that, that has also very interesting property, so let us see that, okay. Now now let us see that, so what did we find that for a given, suppose you

(Refer Slide Time: 36:19)



To find $\underline{c}$ that minimizes $P(\underline{c})$

© CET
I.I.T. KGP

Approach I

$$P(\underline{c}) = \boxed{P_y - \underline{d}^T \underline{R}^{-1} \underline{d}} + (\underline{R}\underline{c} - \underline{d})^T \underline{R}^{-1} (\underline{R}\underline{c} - \underline{d}); \; d = E\{\underline{x}y$$

$\geq 0 \;\forall\; \underline{c}$

$\Rightarrow P(\underline{c})$ min. if $\boxed{\underline{R}\underline{c} = d}$ since R is +ve definite.

Normal Equation

Approach II

$$\frac{dP(\underline{c})}{d\underline{c}} \Rightarrow \begin{bmatrix} \frac{dP(\underline{c})}{dc_1} \\ \vdots \\ \frac{dP(\underline{c})}{dc_M} \end{bmatrix} = -2\underline{d} + 2\underline{R}\underline{c} \; ; \; \frac{d^2P(\underline{c})}{d\underline{c}^2} = 2R \geq 0$$

$$\frac{dP(\underline{c})}{dc} = 0 \Rightarrow \underline{R}\underline{c} = \underline{d}$$

$c^T d$

$= c_1 d_1 + c_2 d_2 + \ldots$

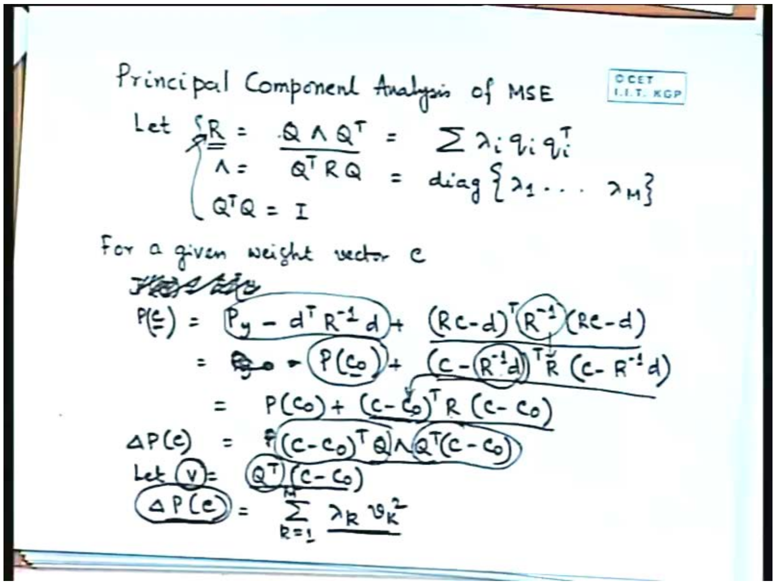
$$\frac{dP(c)}{dc_i} = d_i$$

choose an arbitrary weight c, then what is going to be the error? This is going to be the error, total. If weight c was equal to c zero, that is an optimal weight then this term will be zero, this last term will be zero. See last term is always positive, because R is a positive definite matrix. So therefore; if you multiply with any vector, x transpose R x, R positive definite means, R inverse also positive definite. So if you multiply it by any vector x transpose R x, R positive definite means, R inverse also positive. So if you multiply by any vector x transpose R x, it will this it will generate a positive number. So this number cannot, be can never be less than zero. Which means that this sum must be always more than, this because this is positive which means that, this is the minimum error power that is possible. And that will occur only when Rc equal to d, So that is the optimal, minimum mean square error when this will be zero, correct.

Now we want to see that, how this error are distributed. That is how first of all, how this how this error surfaces look? We want to… we want to look a little bit on this error, you know, that if I so I want to study that, I know what is my what is my optimal weight error. This is the this is the minimum possible you cannot obtain, using linear estimators less than this. Now I want to see that, if I if I am away from the minimum, if my if any if my weight estimate is away from the optimal estimate, then I get a I get some penalty, because my performance indicate performance index increases.

Now I want to characterise; that okay if I if my deviation is on this side, whether the performance penalty will be more or if it is on that side, performance penalty will be more? I want to understand that; how this performance penalty varies, if I deviate from the optimal estimate by an given amount say on that direction not in this direction, that I want to understand, okay. So, for doing that I use Eigen decomposition.

(Refer Slide Time: 38:27)



Principal Component Analysis of MSE

Let $\underline{\underline{R}} = Q \Lambda Q^T = \sum \lambda_i q_i q_i^T$

$\Lambda = Q^T R Q = \text{diag}\{\lambda_1 \ldots \lambda_M\}$

$Q^T Q = I$

For a given weight vector c

$P(\underline{c}) = \boxed{P_y - d^T R^{-1} d} + (Rc-d)^T \boxed{R^{-1}} (Rc-d)$

$= \boxed{P(\underline{c_0})} + (c - \boxed{R^{-1} d})^T R \, (c - R^{-1} d)$

$= P(c_0) + (c - c_0)^T R \, (c - c_0)$

$\Delta P(c) = \boxed{(c-c_0)^T Q} \Lambda \boxed{Q^T (c - c_0)}$

Let $\boxed{v} = \boxed{Q^T} (c - c_0)$

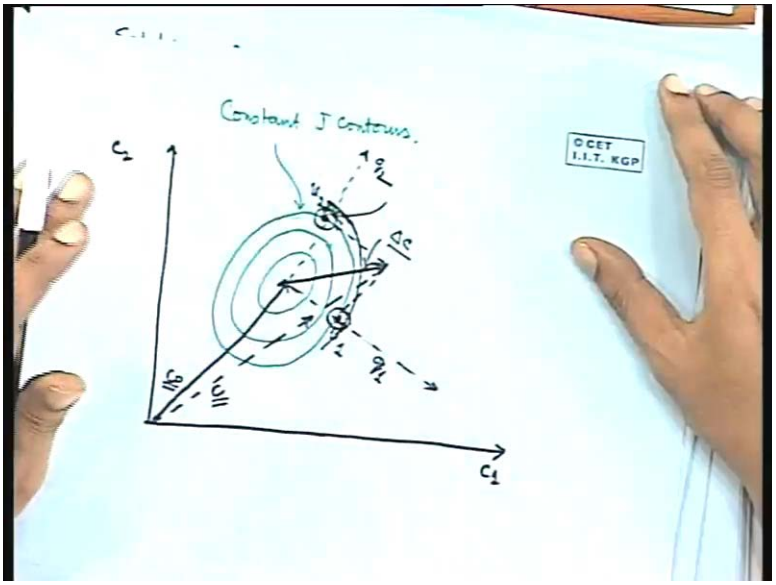
$\boxed{\Delta P(c)} = \sum_{k=1}^{M} \underline{\lambda_k v_k^2}$

So I divide R into again like this; these are the Eigen vector matrices of R, so these results so as we have already seen. This will become diagonalised, this Q transpose Q is I, these result we have already seen. This is just properties of the Eigen vector matrices. Now I see that suppose, this was my penalty. So my delta pc, and now what is this? This is my pc zero, this term; that is if c was equal to c zero, this would have been the penalty that is the minimum. And this one I can actually cast like this, this simple just multiply Rc minus d by by by R inverse. You will, get see here it is R inverse has become R and the here R inverse will come, that is easy to see.

So now what is R inverse d? R inverse d is c zero, because R c zero equal to d; that is the equation c zero should satisfy. So this is c minus c zero, this is my deviation vector, this is my optimal weight vector, this is my… given weight vector. So this is my deviation weight vector. So my penalty, now I have expressed in terms of deviation from c zero, okay. Now what is R? Now I may can now I am making an Eigen decomposition, like this. So I find that if I could if I defined a new vector v, which will transform this c minus c zero by this matrix, what does it mean? That I am trying to express taking taking this components of c and then express it in terms of the perpendicular component, that is the Eigen vector coordinates. See if I express v in terms of Eigen vector coordinates, then my then my error is actually expressed like this; because because now this will be my v and this will be my v transpose, rather rather opposite… This this will be v transpose and this will be v.

So it is v transpose gamma v, gamma is diagonal. So it is nothing but lambda k, v k square, lambda k are the Eigen values of R. So they are the elements of gamma; elements of gamma are lambda one, lambda two, lambda three, lambda four, that we have seen. So this now the penalty can be written as a, what what is the advantage? The advantage is that, now they are become decoupled, that is now it has become a scalar sum. This these v k's are, this v k's are now scalars. It is actually v k transpose, v k; v k transpose v k means, length of v k. So if I these so now now see see the diagram; so suppose this is c zero and this is my vector c.

(Refer Slide Time: 41:31)



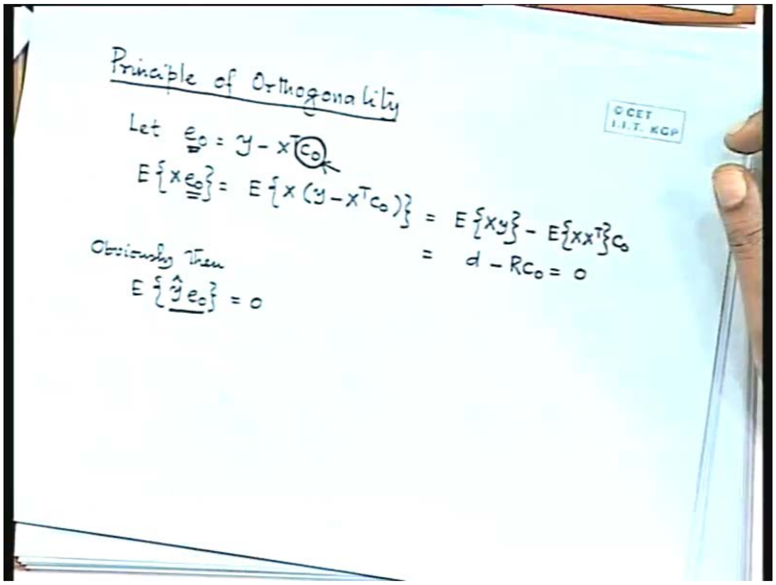
I have suppose; I have only two using, two vectors I want to estimate. So my weight space is now two dimensional, so I have a c one, c two weight space in which my optimal weight is c zero. Now if I choose some other weight c, which is this vector. This is my delta c, now suppose; this is my one Eigen vector rather q1, this is the other q Eigen vector q2. So if I now project this vector in terms of this, this will be my v1 and this will be my v2. That is this vector, I am trying to express in terms of these coordinate systems; they are also orthogonal coordinate systems, I can always express it.

So now the error becomes a function of v1 and v2 that is the constant error surfaces are are ellipses. What is the what is the equation of the error? Now the that that is the penalty in error, it is lambda 1, v1 square plus lambda 2, v2 square. So that is a equation of an ellipse on the on the v one, v two coordinate system. So now these are my constant penalty contours, and from this I can I can see that, the if I if I make a if I make a small error here, then my j will go up more. That is if I here that is if that is on the v2 axis, I have to go more to for a given amount of error, but here I have to go less which means that; this axis is actually more sensitive to error, which means that I should be more careful about weights on this axis, rather than on this axis. This axis I can do little bit of error, performance index will not increase.

So much so you see that just by decomposing it on the on the Eigen vector, I can now get this inside; that on that on which side, I can make errors and on which side I cannot. It will be more… it will give me more penalty. Later on we will see that, when we will discuss adaptive filters; we will see that, if I want to adapt the weights, change the weights, how do I prove whether the whether this adaptation mechanism is is going to be stable? So how do I know that these weights will not explode? So I have to prove certain things and then breaking on to these coordinates will be will will simply matters to a great degree; in the sense that, we will get scalar differential equations other than vector differential equations.

So so so you see that by by by breaking up my error, I can characterise penalty in a nice manner. There are there is one last point, which I want to discuss, that comes from the…what is known as principle of orthogonality.

(Refer Slide Time: 44:43)



Principle of Orthogonality

© CET I.I.T. KGP

Let $\underline{\underline{e_0}} = y - x^T \boxed{c_0}$

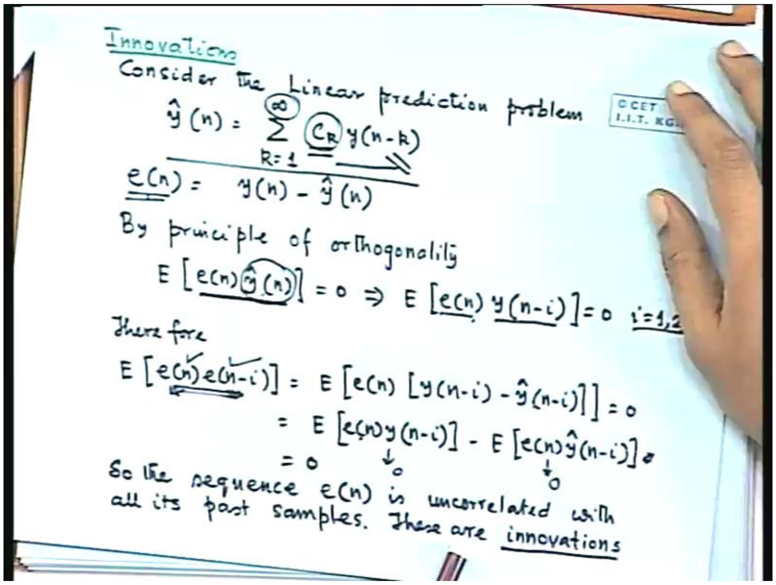
$$E\{x\underline{\underline{e_0}}\} = E\{x(y - x^T c_0)\} = E\{xy\} - E\{xx^T\}c_0$$
$$= d - Rc_0 = 0$$

Obviously Then

$E\{\underline{\hat{y} e_0}\} = 0$

What does it say, that this optimal estimate; that is if you choose the weights as c zero, it has a very nice property. What is the property? That property is that the error; see this is the error, if you use the weight, now the error becomes uncorrelated with the the estimate. That is the estimation, error is uncorrelated with the estimate, now what does it mean? Let us it in a in a in a in a particular case, okay so we had. For example, suppose consider this problem of prediction, okay. What what do you want to do here?

(Refer Slide Time: 45:31)



Innovations

Consider the Linear prediction problem

$$\hat{y}(n) = \sum_{k=1}^{\infty} c_k\, y(n-k)$$

$$e(n) = y(n) - \hat{y}(n)$$

By principle of orthogonality

$$E[e(n)\hat{y}(n)] = 0 \Rightarrow E[e(n)\, y(n-i)] = 0 \quad i = 1, 2$$

There fore

$$E[e(n)e(n-i)] = E[e(n)[y(n-i) - \hat{y}(n-i)]] = 0$$

$$= E[e(n)y(n-i)] - E[e(n)\hat{y}(n-i)]$$

$$= 0 \qquad \downarrow 0 \qquad \downarrow 0$$

So the sequence e(n) is uncorrelated with all its past samples. These are innovations

We want to estimate, the this are very standard problem; that you are given some data, infinite past from zero to n minus one, and you want to estimate y n before it occurs. The very very common problem for example, adaptive noise cancellation; so I do not have the noise sample, but but before then I must estimate it and generate another sort of anti-noise, which will kill the noise, that is what I want to do.

So, linear prediction is a very common problem. So so just just consider this problem, only thing is that this is I have considered this to be infinity, which is the which is done which, when you have a wiener filtering problem; that is if you want, if you want to estimate it using a IIR filter rather than FIR filter, then you will get an infinite number of terms. That is always true, because the IIR filter has an infinite impulse response, rather than a finite impulse response. So now so now but it but still it is a its a linear estimator. So you have a number of weights and so you will have an error, and if you use optimal weights which are given by the wiener filter equations, then what will happen? What will the property, that that the principle of orthogonality will say, that e n will be orthogonal to y hat n.

Now you see y hat n, you have generated, so what? You have generated from y zero to y n minus one? So as if you have extracted, whatever information you could from the measurements of y zero to y n minus1 to estimate y n. So all the information from y zero to y n minus1 has been extracted into y hat n, but then y n is will have some new information. So this error represents that, new information which is in y n but cannot be deduced from the past values of y n. So it is a new… new information, so it is an innovation, right. So it turns out now now, obviously since since innovations are going to be uncorrelated, now so that is what I am going to prove.

So the so the principle of orthogonality says that, this must be orthogonal to y hat n, okay. And y hat n is nothing, but functions of past samples of y. So e n must be orthogonal to all the past samples of y, for all i, right.

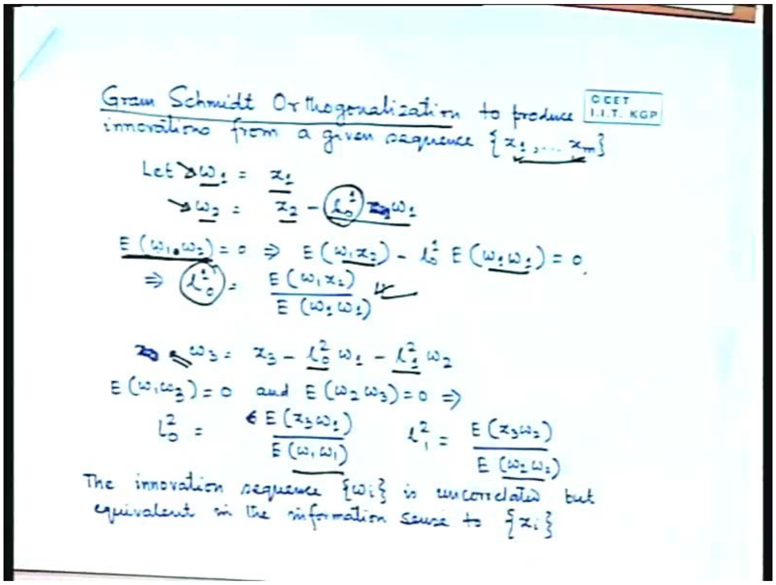
If you have this then you can also say that, all the errors that is you are getting at each instant; you are getting an error y 1 minus y hat 1 is e 1, y 2 minus y hat 2 is e 2 and so on. So you are getting e sequences, now this e sequences are mutually uncorrelated, because what is e n into e n minus y expectation? That is e n into y n minus i minus y hat n minus i, that is the definition and y hat n minus i is again at again has been generated from the past samples. But then e n is e n is uncorrelated with all past samples. So therefore it so therefore this term is also zero and this term is also zero.

So therefore the they the these are innovations and they are like, we we are learnt that innovations must be uncorrelated, so they are indeed uncorrelated, okay. So now what did we see we we said that, we if you have a data sequence y one, y two, y three, y four it has some information. We had said that we can transform this vector into a set of innovations, which will contain the same information as this sequence; but between them they will have nothing, no information overlap, that is e one will will have some new information, e two will have some new information, e three will have another new information. Together e one, e two, e three, e four all together they will contain the total information, which is there in y one, y two, y up to y n. Only thing is the information is now is now compartmentalized, which is not there in y's, right. So the sequence e n is uncorrelated with all its past samples and these are the innovations, and they are white.

This gives you a fundamental property of linear MMSE estimation, that innovations must be white. So you know that you have really reached optimality, for example suppose; you are you are you know this happens in in adaptive filtering very often that, you are trying to adapt these weights and then trying to come to the optimal weight. Now how do you test whether, you have come to the optimal weight, by testing the innovations. So at so at one point, if you find that the innovations are becoming white, then you know that you have reached the optimal weight.

This is the standard test of determining whether the set of estimates have converged to the optimal weights, right.

(Refer Slide Time: 50:51)



Gram Schmidt Orthogonalization to produce innovations from a given sequence $\{x_1, \dots x_m\}$

OCET I.I.T. KGP

Let $\omega_1 = x_1$

$\omega_2 = x_2 - l_0^1 \omega_1$

$E(\omega_1 \omega_2) = 0 \Rightarrow E(\omega_1 x_2) - l_0^1 E(\omega_1 \omega_1) = 0$

$\Rightarrow l_0^1 = \dfrac{E(\omega_1 x_2)}{E(\omega_1 \omega_1)}$

$\omega_3 = x_3 - l_0^2 \omega_1 - l_1^2 \omega_2$

$E(\omega_1 \omega_3) = 0$ and $E(\omega_2 \omega_3) = 0 \Rightarrow$

$l_0^2 = \dfrac{E(x_3 \omega_1)}{E(\omega_1 \omega_1)} \quad l_1^2 = \dfrac{E(x_3 \omega_2)}{E(\omega_2 \omega_2)}$

The innovation sequence $\{\omega_i\}$ is uncorrelated but equivalent in the information sense to $\{x_i\}$

So we will have; and this is just just one minute, this is just you know this is a standard mathematical procedure called, gram Schmidt orthogonalization. Do not get, I mean it is it has a very high sounding name, okay. It mainly tries to say that, the that the innovation sequence you can generate, you can easily generate an innovation sequence just by simple substitution. For example, suppose you are given a sequence x one to x m. How do you how do you how do you transform it into a set of innovations, which will contain the same information as x, but will be uncorrelated? How do you construct it?

So so it is very simple, first take take omega one equal to x one, the first sum take it equal to the first sum. Then take suppose w two is x two minus this; so you are subtracting the w one, first component by a by an multiplying by a weight and then subtracting from the second sample, that you are assuming as w two. Now what will be the weight such that, these two will be uncorrelated, that is the question you are asking. You are asking that, how do I choose this weights, such that these two are uncorrelated? So if you solve this equation, this expectation of w one, w two equal to expectation on w one, x two minus this. That is equal to zero. So then if you choose the weight is equal to this, this will be zero. So now you have got the second innovation element which is uncorrelated with w one.

Now you choose the third, so the third should be uncorrelated both if w two and w one. So you now get two equations and you have two parameters, to solve for. So you solve for the parameters and you get this. So in this way you can go on doing, then you will get a set of sequences w one which had been generated from x one, x two, x n. So they have the same information but they are uncorrelated.

So so these are the innovation sequences, corresponding to this. All this is being done, because as we shall see that for many filtering problems, what we will do is; for suppose I want to generate x from y, necessarily what I will do is, first of all from x I will generate an innovation sequence. I will makes x white, that is called a whitening process. So from x using a filter, I will first generate some white white innovations, then from the innovations I will go to y. This gives me a clear picture of what is happening, rather than going from directly from x to y, we will especially when we have infinite number of terms, etcetera. We shall we shall often go in two steps first, we will whiten the innovations, then from the white innovations we will go to y. So we will come to that step in the next class. Thank you.