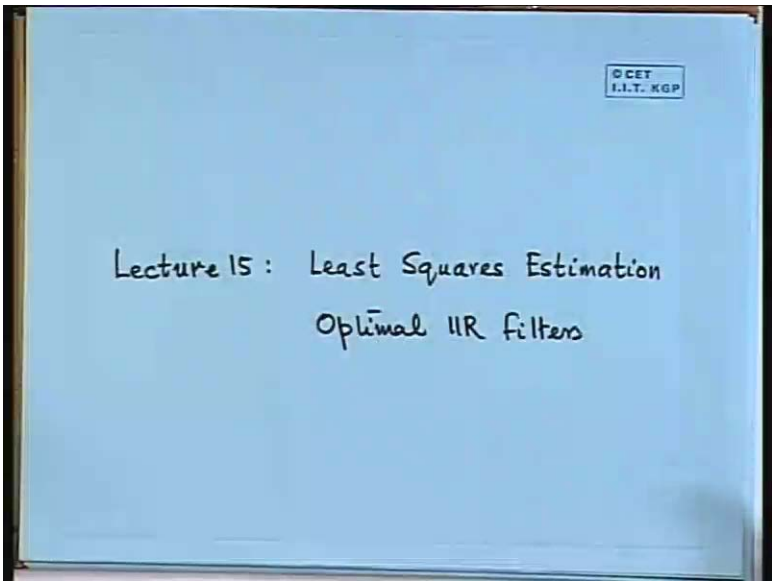
Estimation of Signals and Systems
Prof. S. Mukhopadhyay
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 15
Least Squares Estimation
Optimal IIR Filters

Good morning today we will see two concepts, which are related.

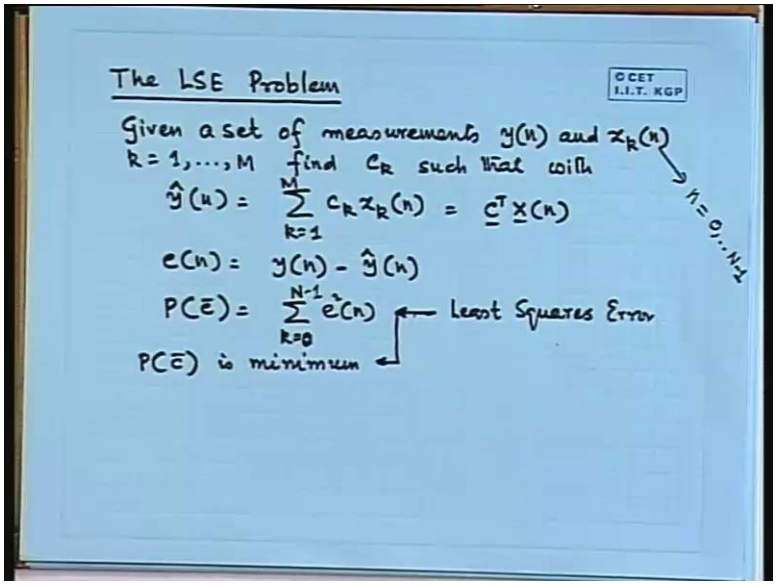
(Refer Slide Time: 00:57)



We have so far seen, how to generate an FIR filter of course we have seen the IIR case also, but and that is the wiener filter problem, but we have solved the for the filter coefficients in case of the FIR filter, using the what is known as the normal equation. So today we will see two analogous developments, first we will see how to solve for a very close problem which is called the least square problem. Previously we used called the problem of minimum mean square... error problem MMSE.

Now we are talking about the least square error problem, or the ALSE. So that is a first thing and the second thing is that, we will also look at the IIR filter design problem, not only the FIR problem. So first of all let us see, what is the LSE problem? So in the

(Refer Slide Time: 02:27)

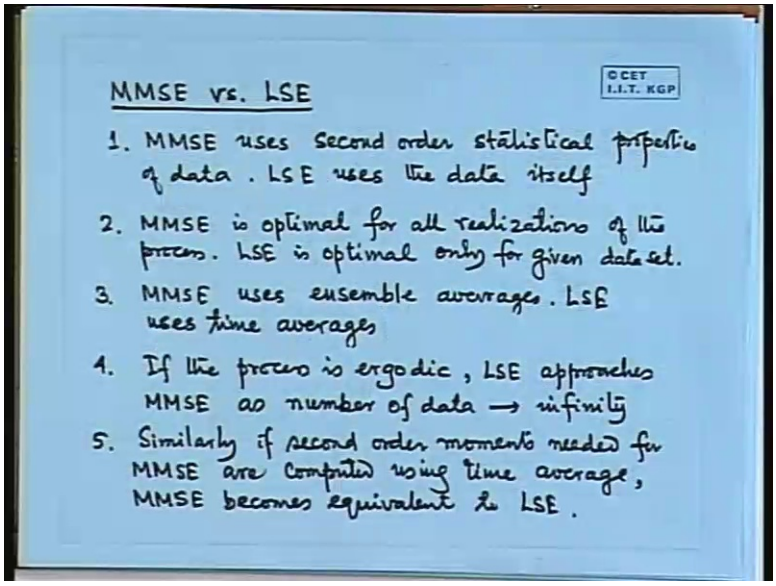


LSE problem, what do we do? LSE problem says that given a set of measurements, it is just like the MMSE problem; only thing is that it says that, given a set of measurements y n and x k n. We want to find a set of coefficients c k, such that if we estimate if we compute an estimate y hat n, this is not y n; it is y hat n using M values of x k n using and then weighting them linearly with c k. So this is an FIR filtering of x k n, if you compute a y hat n, then what are those coefficients c k, such that if you define an error as y n minus y hat n, then this some of this errors over over the n here n here n varies, between zero to; let us say actually we should say zero to n minus one, n varies between zero to n minus one. So find those c k's such that, this sum becomes minimum.

So what is the what is the difference with the minimum mean square error problem? Very similar they are also we wrote this, exactly this equation and formed this error. Only thing is that there rather than writing this, we wrote expectation of e square n, if you recall. In the minimum, it is a minimum mean square estimation problem. So rather than writing the sum of errors for this data points, there we wrote expectation of e square n. That is what we wrote in the minimum mean

square error problem. Actually they are varies, they are absolutely similar. So now let us compare the minimum mean square error problem with the least square error problem.

(Refer Slide Time: 4:44)

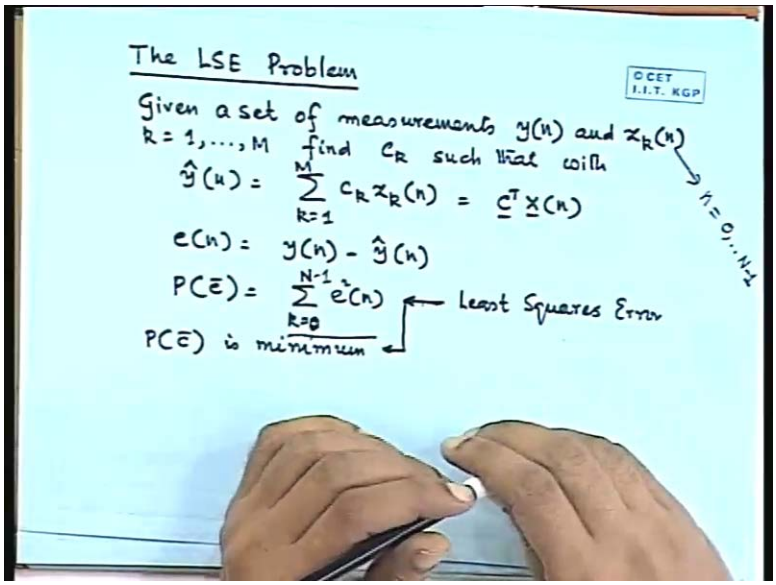


Firstly we note that, whenever we write expectation of e square n; so we are talking about the not only we are talking about this particular data, they are given some data points. But whenever we are computing expectation, we are computing expectation over the whole, over all possible realisations of y n and x n, because we are going to we are actually weighting it, using the probability density function. So we are thinking of many experiments. This particular y n and this particular x n that I am getting is just the just one outcome of the underline experiment, see y n and x n are random processes. So every time I perform an experiment, I will get a I will get one set of signals y n and and another set of signals x k n. So that is just one realisation of this stochastic process. Next time I do an experiment, I will get another set of y n another set of x k n.

So when I am constructing an when I am I mean posing the MMSE problem; it says that the e square n if we if you take an expectation over all possible realisations of this stochastic process, then that will be minimum. So we are we are essentially talking of all possible realizations of the stochastic process. That is today you do an experiment, you get one y n signal you get another one set of x k one, set of x k n signals. So if you use those c k's you will get some some e k

signal also, tomorrow you come and do another set of experiments, you get another y n signal; another x k n signals, you again use the same c k and you get another.. another e n. Like this, if you do many experiments, then all those e n's if you collect and if you take an average over them then till will be minimum. So it essentially talking of many experiments which you are performing, does not talk of any particular set of data. While in the least square problem our, look at these that are performance index is defined in terms of this particular set of data.

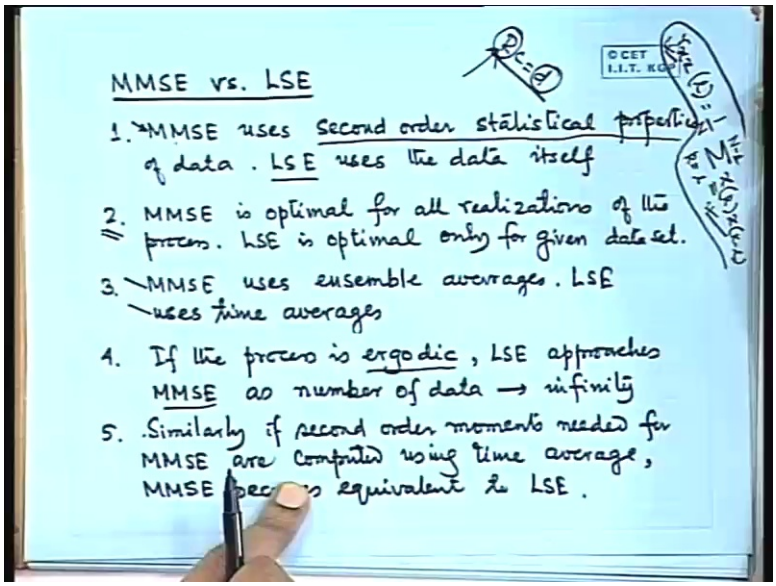
(Refer Slide Time: 07:06)



We are not talking of any expectation, we are saying that we are given this y n and we are given these x k n's, now find out a c k; such that over this data set, this error is minimum. What will happen for for the for the other data sets, we do not know, and we are not talking about them, either. We are we will be happy, if if this particular data set error, sum of error is is minimised. This is essential difference between minimum mean square error estimation and the least square error estimation.

So one is talking of ensemble averaging, it is computing the errors at over many experiments when you is computing the mean, while the other is doing a time averaging. So the other is saying that, some e n over time, right. This is actually a practical version of MMSE.

(Refer Slide Time: 08:12)



So since now now once you understand this basic concept, we can we can realize this point, for example obviously if you are talking about the whole stochastic process; that is all realizations and if we want to obtain a solution, which is will be optimal over all possible realizations, without talking of a particular data set then, naturally I have to know the property of this stochastic process, itself. How can I minimize an error, over all possible realizations of a stochastic process without known without knowing its properties? It is not possible. So so therefore, we have found that we need to know, what was our solution on? Our solution for the for the optimal problem was, Rc equal to d.

This I mean the solution of this equation, so what is R? R is the second order moment of x. Similarly what is d? d is the cross correlation, R is the auto correlation function of x; d is the cross correlation function between x and y. So we need to know these things. So we need to know the cross correlation and auto correlation functions of the underlined stochastic processes, without this we cannot find then MMSE, right. On the other hand the LSE does not need such thing, so so so even if you do not know the properties of the underlined stochastic process, you can always compute an LSE, given at set of data. So it is a so it is a much more I mean its it can be always found out, does not require much knowledge much, prior knowledge. Everything is

calculated based on data, so it is a very practical, you know I mean technique which can be applied in all situations, that is why it is so.. it is so popular.

So so this is the first thing that, MMSE needs second order statistical property is excuse me of the data, while LSE does not need to know any second order properties, it just uses the data itself. Second thing is that second thing is what I said just now, that the MMSE is optimal for all realizations of the process, LSE is optimal only for the given data set, it does not say anything about what will happen, if you if you use the same set if you use the same weighting coefficients. Suppose by by given one data set, you have computed some optimal weights in the LSE say, address based on that set and you have got some c k's. Now now now if you use those c k's for another data set, what will happen, we do not know. We may get a very very very poor error, lot of error we may get. LSE does not talk about that at all, okay.

So so the LSE is optimal only for the given data set, the the MMSE uses ensemble averages and the LSE uses time averages. So so naturally, now you know that if the process is ergodic then the LSE, if you take long time average then the LSE we will we will approach the MMSE, because ensemble averages will be equal to long time averages. Similarly if the second now similarly you may you may ask that, okay MMSE we will use the second order statistical properties. Now will I how am I going to get second order statistical property, right? So one one way of obtaining second order statistical property is is to do is to actually do, ensemble averaging. That is perform actually perform many many many many experiments, for each for each experiment compute compute auto correlation function, auto correlation function and then sum them over then ensemble, then you will get the second order property. But remember that, that what we said that if you are given one sequence; how can you estimate the auto correlation function, so we wrote that it is r x x, can you see this on the screen?

So you saw r x x al is equal to one by n sigma, this is x k x k minus l. And if you have this will go to this will go from say k will go from l to n minus 1. So if you are given a finite sequence of data, this is how we you can you, if you are given one finite.. set of data from where x k values are are available, from zero to capital N minus 1, one N length sequence; if you are available then this gives an estimate of the auto correlation function, we had seen that. So if you you know

if you are suppose you are given one set of data, now if you want to if you say that, I want to use the MMSE, how will you get r x x l? How will you get the get get this matrix? So you say that okay I will now compute, I will now approximately compute r x x using this formula then though you are saying that; you are using second order properties, actually you are computing using time average, so therefore your solution will tend to the LSE.

So so they are actually very related I mean, that is why I thought that rather than rather than coming back to them at a later time, this is the time to discuss them because get absolutely similar equations everything similar, okay. So therefore we have a we have we have we have a very similar problem, only thing is that we are what… what we are trying to say is that, we are in the least square problem which is the more which is the more practical version; does not require much prior information you can apply always to a given set of data, you can define a least square problem and then solve it. Even when you do not know much about the underlined stochastic process, right?

Student<Can MMSE be seen as a sum of error LSE's?>
[Conversation between Student and Professor – Not audible ((00:14:57 min))]

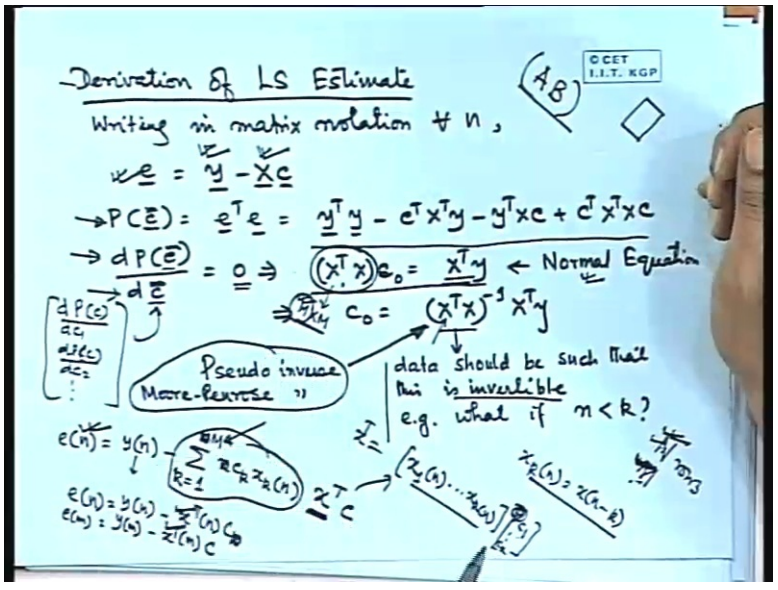
Yes in a in a sense that, if you you can later see that suppose; you have taken a taken an LSE problem and then if you, so you get an so you get an LSE estimate. Now if you try to get take take an expectation over this LSE, rather if you say that I take one N minus 1 sequence, I define an define an LSE problem and I solve it. Next time I take another sequence, on that sequence I define another LSE problem and I solve it. So I get another different set of c k's now, because an LSE problem is, LSE problem solution is always particular to that sequence. So in this way if I do it over many realizations, every time if I define an LSE problem and if I solve it, then I will get a different c k. So now my c k itself I mean the set of c k's are itself random variables.

Now if I take an expectation over expectation of this random variable, what will happen? So we will see what will happen, and we want to know I mean we want to know whether whether whether this expectation has certain good properties; because typically what will happen is that, why do we why do we construct an estimator? You see while we are constructing the estimator,

there are two phases; one in in one phase we will design the estimator, and in the other phase practically speaking, we will use it. So so if we have if we have a stochastic process, then while we are designing the estimator. We will use certain data point. So we will we will essentially design based on certain set of experiments and then we will design and we will we will arrive at some numbers c k's. Now now naturally we are going to use this c k's on other realizations, because once you have designed; we are going to use it, so then we will use it a new set of data and that will give rise to a new sets of errors. So now the question is we are we are always interested in this question is that, suppose we had designed based on those data points on which we are using, we had optimally design some c k's with respect to those data points and now I I have now I am using a c k which I have used based on, some other data points and I am using it on them. So what is the difference between these two c k's, right?

So so in other words we would like to know, several properties of the estimator that is if you take an expectation overall things, does it converge to the MMSE? We will we will answer this questions, right.

(Refer Slide Time: 17:51)



So we will derive the here the LSE absolutely similar, that is why I have not written all the equations; that you can I have directly written this you see, if you want to understand how this

equation is written? You can you can understand that e n is equal to y n minus sigma. Let us say k is equal to one to n ,x c k x k n, okay. This is my e n at any time instant n, right. So if I now stack this e's if we now stack this e one, e zero, e one, e two, e three. Stack them in the form of a vector I will get this vector. All this y's will also get stacked, so I will get this one and this sum is equal to what? Is equal to you can write x transpose c. You can write this x is a vector, this this x transpose is x one n to x k n transpose. And it is a column vector, x is a column vector. So x transpose is a row vector. So x transpose is nothing but this vector and c is a column vector, c one to c q. Can you read this? I do not know, this is probably too small. I am just trying to organise now, so so the first equation becomes e n is equal to y n minus x transpose n c k not c.

Next equation will become e m is equal to a just writing y m minus x transpose m c. So I can stack them all these x transposes then I will get the matrix X. So I have written this vector matrix equation directly. This looks absolutely like that previous thing. We are also we here wrote the same equation, because we were using the same same error equation. So again we can write a sort of performance index in terms of this this in incidentally now this X is a matrix, c is a vector. This should should be not be over bar, it should be under bar. So now we can again write, what is the what is now my performance equation is? The it is nothing but the e transpose e, sum of we have to this matrix manipulations you know, one vector row vector into column vector means, sum of squares. So if you just multiply y minus X c transpose, y minus X c; you will get this, this is very simple.

So now you differentiate it with respect to c every time, I have written over bar actually; this should be under bar. I am saying under bar just to indicate that, this is a vector because I cannot write bold or anything. So now if you differentiate it with respect to this expression, if you differentiate with respect to this vector again, every time I am repeating that this is a scalar so number this is a vector. So it means that, it is basically d p c by d c one, d p c by d d p c by d c two and so on. This vector is noted denoted as this, okay. So this if if you just do it term by term, you can always do it term by term; write a matrix as x one, x two, x three, x four. If you do not understand how it is happening first time; you can always write it in terms of scalars, actually write this expression in terms of the matrix elements and then really differentiate with respect to c one with respect to c two, you can do that. And then you will you will find that, if you if you

differentiate this and then set to zero you will get this equation. This is nothing but the same normal equation, only thing is that previously this is R; now in there you got R because you had an expectation. Now you do not have any expectation; you are only doing it with respect to the that data set and this is like d, cross cross quantity between y and x, right. So you get the same normal equation and.

Now so so this is, what is the dimension of this matrix? What is the what is the dimension? I have written X transpose, X inverse, is it invertible? Is its square? N by N, capital N by capital N, N for north. It is capital M by capital M, where M is the number of terms here. Sorry this should be M this one, that is the order of your FIR filter. N is the number of points, M is the order of FIR filter, in a in the estimate how many terms you are taking. See when you are doing an FI, why I say FIR, because this this is the very general description. You can also you can always write that x k n is equal to x n minus k. You always define x k m in that way, then it will become a true FIR filtering. Now it is not I am saying FIR filter FIR filter, but it is not it is actually as if data is coming from M capital M different sensors and we are making them sum, into this equation looks like that.

So it is an M into M matrix, it is square but we do not know whether it is invertible. So how are we writing these? So so we can only write this provided it is invertible. So when is an when is an M into M matrix invertible, when its rows or columns are linearly independent, then only it is invertible. Now first of all, why it should be linearly independent? First of all remember that; this is a composite matrix which is remember that, if a if you if you multiply a matrix A by a matrix B, then the rank of AB can can never exceed the rank of A or B. So this has to be of rank M, this only then X transpose, X inverse can have full value; otherwise this this this M into M matrix will be singular and you cannot solve this problem, like this.

So when will the matrix X will have a rank M? Matrix X has how many columns, how many rows? Matrix X has Capital N, capital N rows, because we are writing stacking it in terms of data and it has capital M columns. So so first of all for any non-square matrix, the maximum rank is the minimum of number of rows and number of columns; rank can never go beyond that If you have three by two matrix, maximum rank can be two. So so so the rank is going to be minimum

of N and M. So firstly; if N is less than M then you cannot solve this problem, because then the rank is going to be less than M and. So if you have number of data points; which are lower than the number of the if FIR filter, then then obviously there is an solution I mean, right?

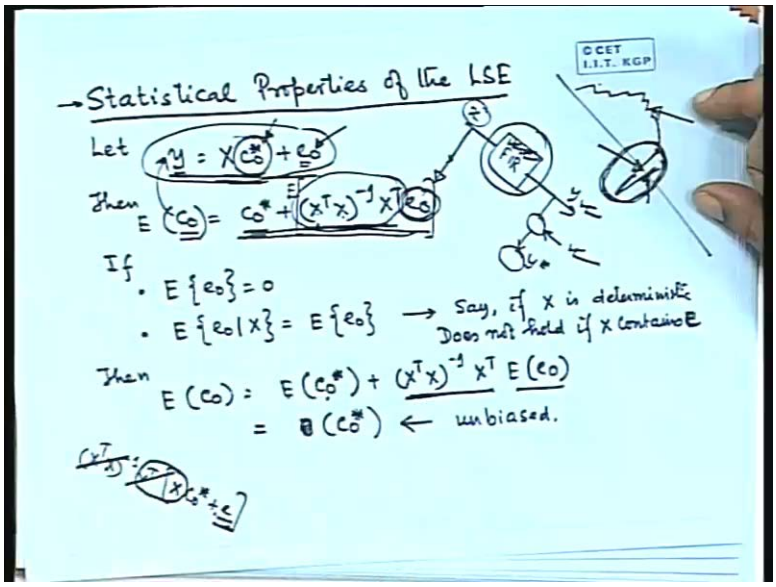
But it is not guaranteed that; if you use the just just if you use more number of data points than M, this matrix is going to be invertible. If N is less than M, no question of being invertible; if even if N is greater than M, now it is may now it may or may not be invertible. So you should choose such data points such that, it is invertible; otherwise there is no solution, right. So we have to remember that, this thing is incidentally called as pseudo inverse; because if this if in this equation x was invertible, x itself was square. Suppose it was; then you would have calculated c as x inverse y, but it is not square. So if it is so you are, it is actually a rectangular still you are in a in some sense you are trying to invert this term. So that is why it is called a pseudo inverse or sometimes it is called a Moore- Penrose inverse; named after the scientist who has discovered this.

So actually what you are doing? What what are doing? You are you have you see when when N is greater than M, how many unknowns here? Capital M number of unknowns, because so many components. I mean so many coefficients of the filter, you have to solve for. And how many equations, Capital N number of equations. So if you have five equations and if you have three unknowns, what solution do you get, linear equation solving. No no either you do not get any solution, I mean you in general do not get any exact solution. If you have five equations and you have three unknowns, if you have three equations and three unknowns, and if the equations are independent you always get a unique solution. If you have more equations than unknowns, then in general you cannot solve it; that is using there will be no value of those unknowns which will simultaneously satisfy all the equations, that is the general situation. But in some cases of those equations, it might if you have two equations and three unknowns, then you have an infinite number of values of the unknowns, which will satisfy those two equations, okay.

So so this is about the solution of this equation and this matrix invertibility is the key issue, in being able to compute the estimator, okay. So now we are coming to your question that, what are the statistical properties of the LSE? Where it is important, because we are going to design it based on some datasets and we are going to use it on a other data sets, right. This is called a

generalization property okay; very important for example, very important for neural network training and other things. I mean whenever you you construct an estimator like an like a neural net is an estimator it is a it is a non-linear estimator which is very popular, you have you have to ensure, you have to train it to ensure that you have a generalization property. That is it will not only fit this data set well, but it will fit other data sets well as well this is the key property, okay. Sometimes it may happen especially with things like, neural network it will happen that; they have they have they are they are they are so powerful in the sense that, they can match almost any data set. I mean once it happened that; I was I mean I am just giving you this example, that I was trying to model the load variation in the city of Calcutta, using a neural.

(Refer Slide Time: 30:36)



So the load variation in the city of the I mean electric power load variation if you if you plot, you will have you you I mean it is typically a periodic series. Firstly of all there will be a period over an year, during summer load will be more; during winter load will be less, this will repeat summer winter summer winter. If you go little little inside, you will find the there is a smaller period over weeks, right. Typically depending on the industrial activities, nature of the load, you will find that may be every Mondays load will be slightly higher than every Wednesdays. So in the week there is a cycle which repeats over weeks. If you go in a day, you will find still smaller cycles, right. In the morning it will be low, around nine ten you will get what is known as the

morning peak; it will come down again you will get an evening peak. So it is a so it it also cycles over a day. So you get generally its random but you get a periodic pattern.

So I was trying to model it, now what happen is that you know load is going, suppose one day one particular day I was plotting for a year; there was some you know, political meeting or something, something happened, so that the load just fell and then it rose again. So that particular data set that I were using had this key, because of something peculiar there, probably there was some some accident happened or may be some bandh(foreign language) or something, so the electric load fell. Now when I was using a neural network, my neural net actually this is a peculiar problem with neural nets; does not generally happen with other kind of estimators, but I am mentioning it just to emphasise the point on generalization. My neural net.., see if I train it, what I am trying to do, I am tweaking the weights of the neural nets such that; this data set is is I mean nicely approximated, so i am actually solving an LSE problem but on an but on but my but my model is not these, my model is a much more complicated non-linear model. So my neural net actually model disk, it was so powerful that it model disk kinks.

Now the point is if it if it model this kinks, then on this data set the estimator has performed very nicely but; in generalization capability is seriously impaired, because if you use any other now if you give it. I mean why I was do it? Because I was doing it, on behalf of an electric power company; to to show them that is how to how given the last seven days load, they can estimate next day's load because they need to know, when to send a boiler for maintenance. So what is going to be the load? So they need to need to anticipate it, given the weather conditions vegera vagera [foreign language word], okay. So because my net learnt this behaviour, every time during that part of the year; even if you take another year if you give another load on that day, it will give this kink, because it has learnt to give that kink.

So it so it is very accurate on the set of data on which it was designed, but it is too much accurate and it has lost its generalization capability. So if you compute on another set of data, you will actually get too much error on these points. So so so actually during neural net training, you should prevent the neural net from learning those features of the data; which are not generalized, you should actually prevent the net there should not train there, okay. So that is very important.

So so when okay well we are we are we are we are considering lower order, I mean lower order estimator. So we are interested to know that, if we have the MMSE which is you know which is supposed to be grand; because it is designed based on it is supposed to be optimal over the whole stochastic process realization, all possible realizations in an in an in an average sense.

So now what we are saying? No sorry sorry this is not what I am saying what I am saying is that, if you want to know the statistical properties, one one one obvious way of looking at it is that let if y is actually this is not y hat, this is y, that is the data which you want to match using your estimate. Suppose y is actually generated with with the same model, that we are trying to assume but only thing is that, there is some additional there is some additional term it is like, it is like as if as if the the data y and x are actually generated using a FIR filter; some x which you are measuring and some unknown FIR filter has generated y, the observation. We are just assuming and then while sensing y, you have got some noise. Every sensor will give you noise, so here I have a noisy measurement of y and I have this x.

This is the very common practical situation for example, if you are having a control system then why you will not consider x to be noisy. Because of the fact that, x is this is this just I am giving an example that; suppose you are having a computer control system, so x is coming from your computer itself, so therefore you already have x. So there is no question of measurement while why you are measuring through sensors. So therefore they may be measurement, this is just one situation. So we are assuming that that the the data has been generated, actually using an FIR filter which is unknown. Now the point is that, if I if I if I use the same FIR filter structure and then solve an LSE problem, will i get back this these parameters which were used here? They are this so called; you know so called true parameters, which were used to generate the data. If I use an least square estimate, will I get back those parameters?

This is obviously; a if I get back it is good. So we are saying that, let y generate the data exactly in the same manner, that is we are using the we are in assuming that, the data generation mechanism is the same as the estimator structure, only with an additive noise. It may or may not be same, in general it is not same. You can data may be generated; let us say by a by a tenth order filter and you can still, it is still valid to try to estimate a second order filter, nothing wrong

in that. But in this case we are just synthetically assuming that, if a tenth order filter has been used to generate data and if we are estimating a tenth order filter, then can I get back those parameters which were actually used in generating y?

So after all what is my c zero? c zero is c zero is the is my LSE; that is generated from y using that equation, x transpose, x inverse, x transpose y. So now if you put y is equal to this, then you will find that the equation comes to this x transpose, x inverse, x transpose, now I will put this equation in terms of y, so x c zero star plus e. So this x transpose x, so this and this will get cancel. Now you get c zero star here and then this multiplied by this, so you get this equation. So now you see that, now now we are ask this question that what is expectation of c zero star? That is now this expectation is over what?

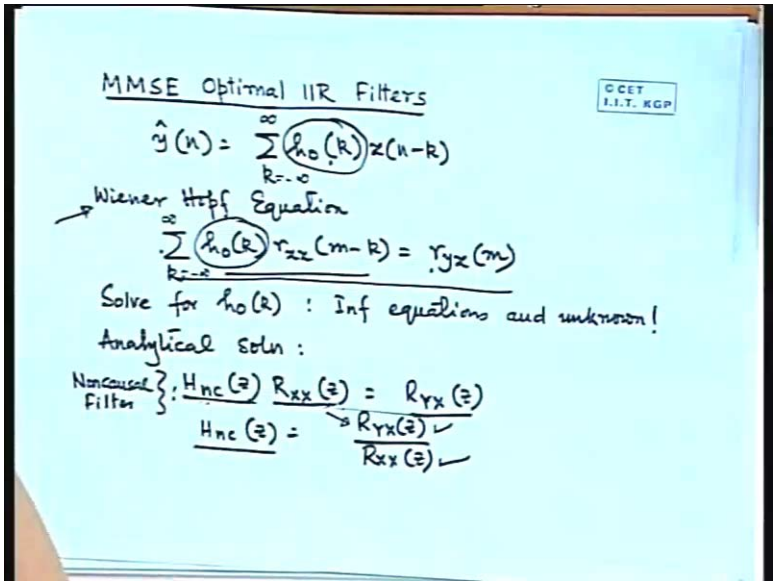
Expectation is always over a random process, which is the random process here? I have not said that x is a random process; x can be a very deterministic process. You can send a square wave to a system and see its response, triangular wave, whatever. So it is not necessary so for for x x may or may not be random. We are assuming that, this e's random because this noise, okay. So we are now taking the expectation, wherever you take expectation you have to know, over what. So we are taking this expectation over all possible e's, okay. So again if we assume that, now now when we do we say that, expect obviously expectation of c zero is equal to expectation of c zero square star.

Now expectation of c zero star is c zero star, because we have used some values of coefficients to generate the data; it may be unknown but it is constant, it is not varying. So therefore this is like c zero star and ideally speaking, you should put the expectation here, outside expectation of the whole quantity. Now when can you write expectation of the whole quantity is equal to this into expectation of this? That is I am now taking the expectation inside, only on these this I am cutting out, when when is it possible? If it is a deterministic sequence it is possible, otherwise also if these are if these two if this random process and this random process are independent then it is possible. But if they are not independent that is for example, if if x itself contains e then, it is not possible you have to remember this. So if I mean a very simple result is that, if it is deterministic or if you can take the expectation then you can take.

So you suppose, if it is deterministic then this is a there is no expectation. It will come out and you take this and then if you have this as zero, then you get c zero star. So remember that, the LSE will give you will give you back the true parameters only under certain conditions, it will not give you the true parameters otherwise. In general it is biased; in the sense that if you… if if data was generated in this fashion and if you compute an LSE, and even if you take expectation it may not come back to this. That depends on what is x and what is e and what you what are their mutual properties. Especially what will happen is that, when we here we are using this kind of a model; sometimes we use an auto regressive model, that is x contains past samples of y, now y contains e, y contains e, so the past samples of y also contains the past samples of e.

Now if the past samples of e and this sample of e are correlated; that is they are not white then then this thing will not you cannot write this. This situation we are which the in when we will study identification we will see that various kinds of model that is why the model structure is very important. What model structure we choose, based on that we may get, we may get a bias estimate, we will get an unbiased estimate. Some of you who are doing process monitoring and fault detection might have come across this, okay. So we are till we have no time to discuss this second point, today that is MMSE optimal IIR filter but let me a start the discussion.

(Refer Slide Time: 43:23)



We are going to repeat this, that is so far is a you know, we are we always talking of FIR filters that is I am taking a finite number of samples; but I can take a IIR filter, I can take any infinite number of samples. So how do you calculate that? How do you calculate such a filter when there any finite number of terms? So basically the the thing is that, finite or infinite you always get the super normal equations or wiener Hopf equation. If there… if they have an infinite, you get the wiener Hopf equation; minus infinity to plus infinity; that is you you have defined this filtering problem k equal to minus infinity to plus infinity. We are not assuming whether this filter is causal or non-causal, you are first defining a general problem and we want to we want to; what is the our objective, our objective is to solve for these. These are the impulse response coefficients of the filter causal or non-causal. We want to find out these such that the expectation of e square n is minimized, this this is the problem. So it was given by the wiener Hopf equation that, if if you define that problem then you can get this coefficients by solving this equations.

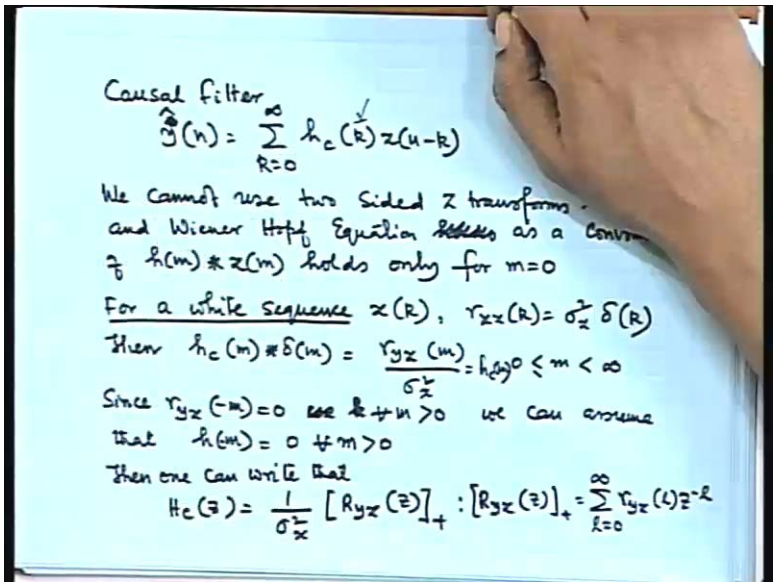
Now the main problem is that, you have an infinite number of equations and you have an infinite number of unknowns. So you so you so you cannot even write them, to solve it. So then how do you solve it? You can always define a problem nicely but practically speaking, how do you solve it? So so some people said that okay, can I can I even solve it, I mean sort of you know non-parametrically, that is can I get some closed form solution in terms of something? Here I am not getting any solutions, I I only say that if you want to get get this you have to solve these equations, other than that I cannot I cannot write the solution. So first thing is that people said that, okay it is it is simple, I mean at least you you use Z-transforms. So if you Z –transforms, then this is nothing but a convolution equation. So you write this as the product H n c z, n c for non-causal because this goes some minus infinity to plus infinity. So you can easily a Z-transforms and then you and then use the Z- transform property; this will be R x x z, this will be R y x z. We know that the convolution in a in the time domain becomes multiplication in the different. So so so in the Z trans if you take the Z- transform of these and the Z-transform of these, then this will be related like this.

Then you simply define that H n c z, is this by. So what is it give you? It gives you some some some h and c z, it is some plot. You just compute this R y x z as some plot and you get; I mean if you just happened to define these and these, this is you can actually compute based on data. Then

if you divided it, you will get this. This is at least you got some solution. You do not know, how to do this so far, but is you could write the solution. Then people said that okay, this is fine I mean at least you can you can do it, analytically, symbolically but this is for non-causal. Now in some application domains, non-causal filters are useful. They can be physically computed like an image processing, but there are some domains were they are not useful because because it cannot be computed. So there are some application domains where we have to restrict our self for. If we want to do anything practical, we have we have to restrict our self to causal effects, cannot use non-causal effects. So what about that solution, what about the causal solution?

So then people said that okay, so the so so now the causal solution I mean the and the main problem of not the, we will discuss this in more detail, in the later class today. We do not have time to go through it, but the main problem is that you cannot define. You see this this sum now is a is from zero to infinity. It is not from minus infinity to plus infinity, because you are looking for a causal filter.

(Refer Slide Time: 47:58)



You are trying to solve for a filter. So that filter must satisfy this equation and this equation is defined, that is your filter. These are it will be a causal filter. So so these are going to be zero for k less than zero, right. So which means that now obviously if you have this equation; you cannot

write a Z -transform equation for k is equal to minus infinity to plus infinity you cannot take, because you you do not have values okay. You cannot define auto correlation functions,, okay. So therefore people are people are saying that well. So so we will we will see in a later class; people will take a slightly roundabout way to compute the solution, I mean not through a direct Z transform taking, because a direct sided Z- transform cannot be taken.

So what people will nicely do is that, they will now they will use the innovation process that is they will first from x they will generate a white process. Using slightly different filter I mean using a filter which is causal. See we always have to we are always looking for a causal filter, so whatever we generate we must use causal filter. So first of all what we will do is that, we will see that how from x, I can generate a white sequence using a causal filter.

As we have seen that given random vectors; they can be made into uncorrelated random vectors, which are called innovations., And we have so even a random sequence x, you can construct a corresponding random sequence w, which is white that is called the innovation sequence for x using linear filters. So since I have I have I mean the the argument goes like this, that since you are generated w from x using a linear filter; therefore w is equivalent to x. So the problem of estimating y from x is the same as the problem of estimating y from w because you can generate w from x. So now you do it two steps; first x, from x we generate a white w, and from that white w you generate y. So that is called a whitening process and that I mean that actually you know clarifies the maths. So we will do that in the in the next class, today we have do not time and okay, so.