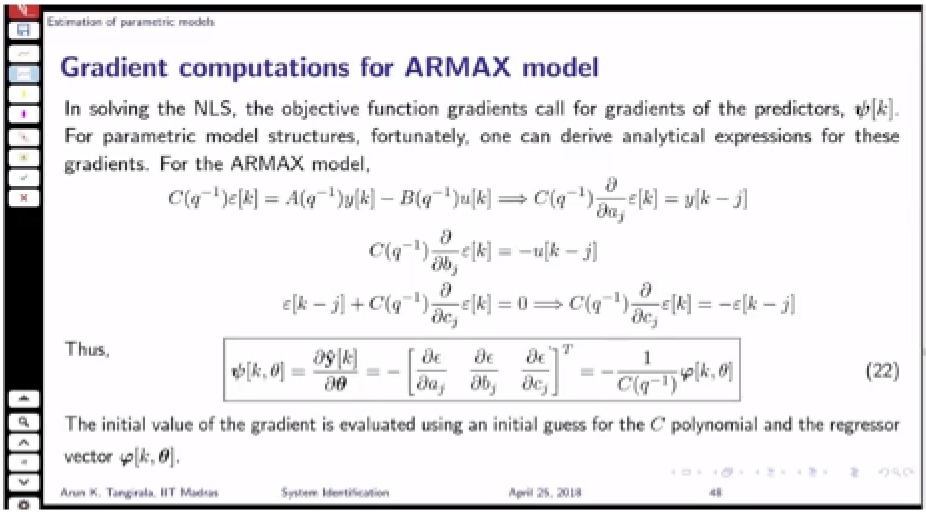
CH5230: System Identification
Estimation of parametric model
Part 4

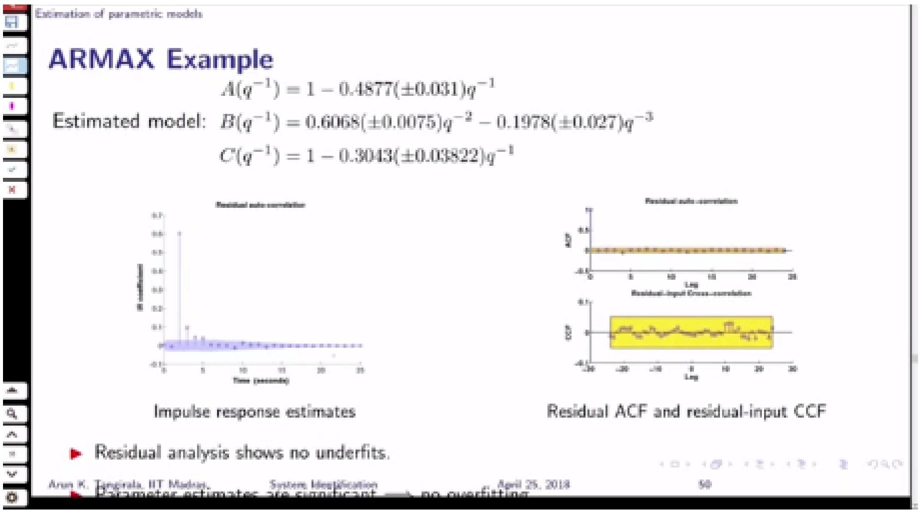
Okay, so, welcome back to the lecture on estimation of parametric models. In the previous lecture, we learn the concepts of PEM methods. And the point where we stopped was very discuss the estimation of ARMAX models, where we had the choice of using the nonlinear least squares estimators which is essentially a quadratic PEM, for which the gradient computations when necessary. And I showed you how you set up the gradients.

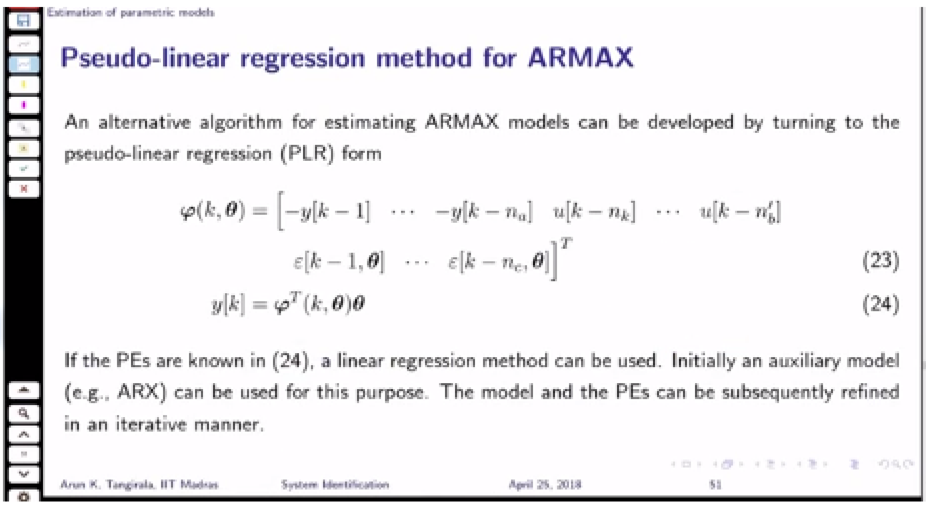
(Refer Slide Time: 00:44)



And by the way, this procedure for arriving at the analytical expression for the gradient of predictor is similar for other models structures such as OE and B-J and so on. Essentially, the fact is that you have an analytical expression for the predictor gradient, which you may not have for many other models, but the parametric models lend themselves to this benefit.

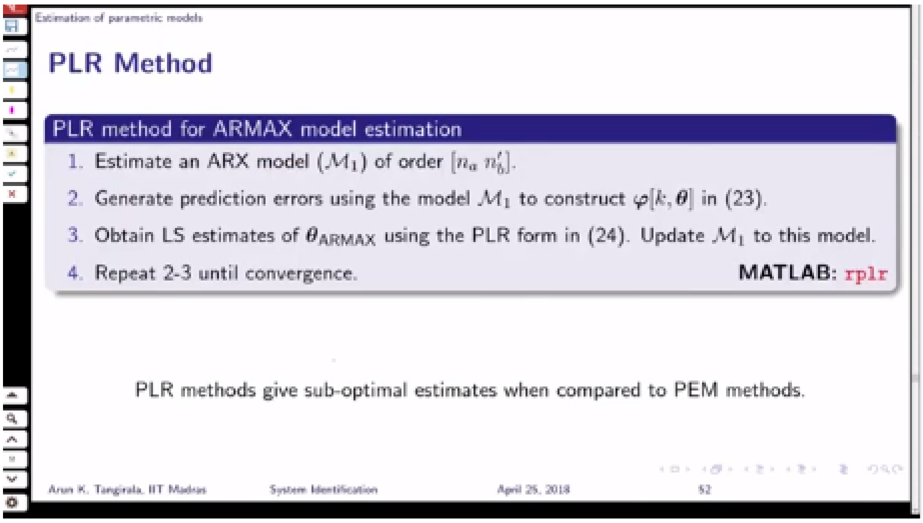
(Refer Slide Time: 01:11)

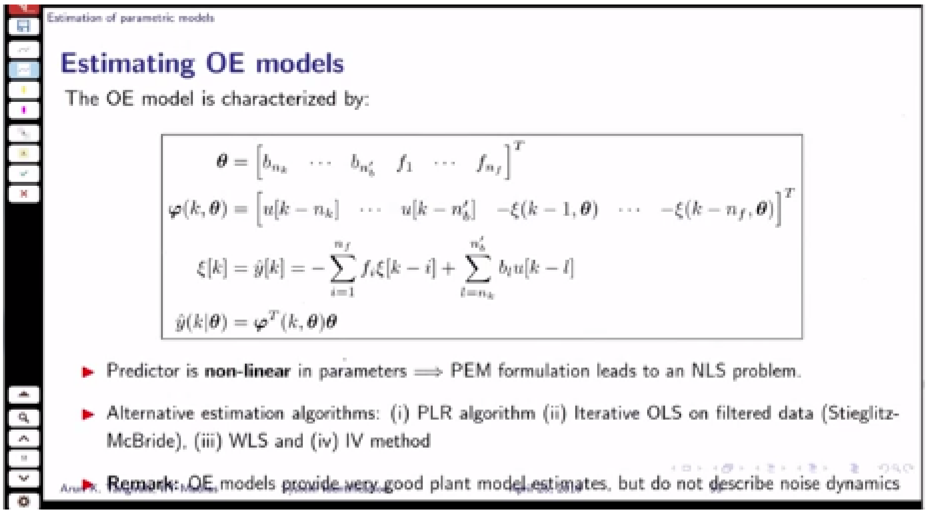




Then we also discuss the pseudo linear regression approach for the ARMAX structure. You can use the pseudo linear regression structure for other models as well. But again, I will not discuss for every other model.

(Refer Slide Time: 01:25)

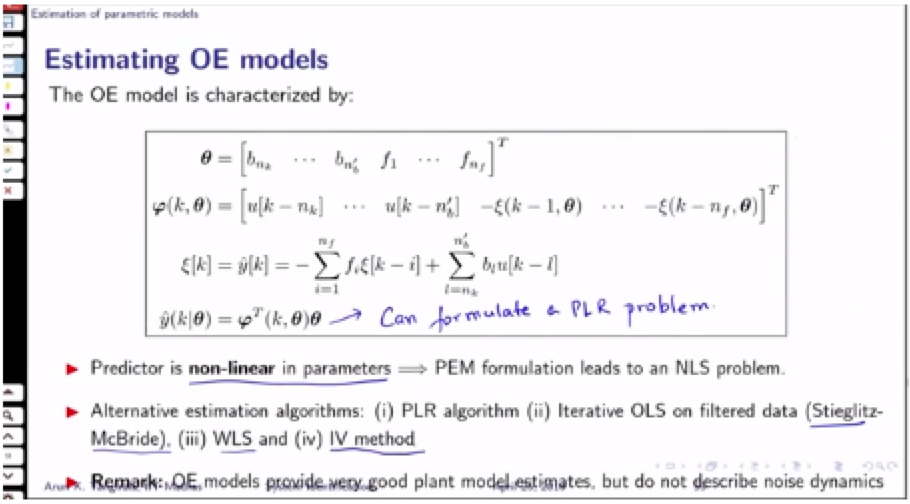




What we shall focus specifically in the beginning of today's lecture is OE models. Because they are a special class of models and they have some nice properties and as usual, they end up with the predictor being nonlinear in parameters and therefore, the quadratic PEM formulation leads to a nonlinear at least squares problem.

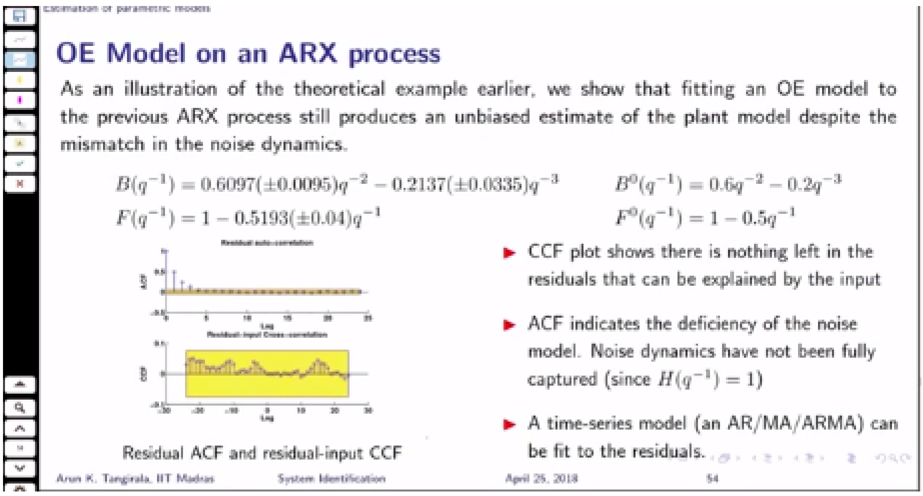
And also you can notice that you can actually formulate a PLR kind of a regression problem, pseudo linear regression problem. Again on the same lines, as we discussed for ARMAX. There are other algorithms apart from the PLR algorithm, which is the celebrated Stieglitz-McBride algorithm which preceded the PEM algorithms. The Stieglitz-McBride algorithm came about in the 60s, and we'll talk about that today. Then you have the weighted least squares options which we will skip.

(Refer Slide Time: 02:27)



And then there is IV method, which we met briefly discuss at a later time in this lecture.
So you should remember again, recall that OE models, structures offer, very good plant model estimates, but do not describe noise dynamics. So let's actually work out an example of the-- MATLAB example of estimating the OE model.

(Refer Slide Time: 02:50)



But the data generating process is ARX. This is an numerical illustration of the theoretical problem that we solved in the previous lecture, right? In a previous lecture very showed theoretically that as N goes to infinity that is asymptotically. The estimates of the OE model, when you use the quadratic PEM algorithm, converts to the truth. And you can see that here, this is the true process, this is your s. And on the left hand side, you have the estimates of the parameters, estimated from finite length data.

And you can see that the estimates-- the point estimates are pretty close to the truth. And before we jump to looking at the errors in the estimates, it's important to look at the cross correlations of the residuals with the inputs. And you can see that the cross correlations of the residuals with the inputs are insignificant because they lie in the insignificance band or significance band. On the other hand, the auto correlation of the residuals here shows that the noise model is inadequate, right?

But that's to be expected, since in the OE model, we ignore to choose the noise dynamics. So, this is something that you will run into whenever you estimate an OE model for a given data. The focus should be on the cross correlation. While keeping an eye on the auto correlation so that you know that there is a noise model that has to be fit. In any case, your non-parametric analysis would reveal whether an OE model would be suited or not.

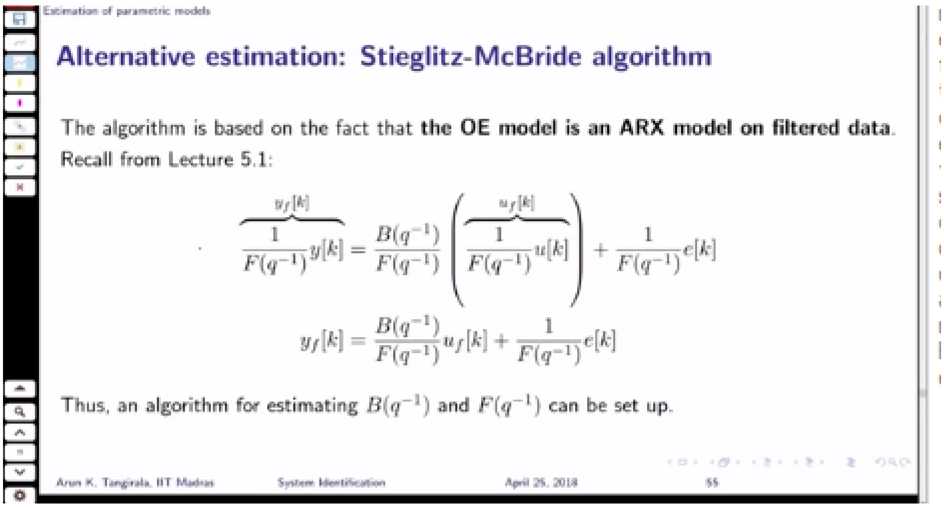
Typically in SYSID, one practice that should be developed is to make use of redundancy that is the non-parametric analysis gives you an estimate or an idea of the suitability of often an OE model structure for example. And again, here a parametric analysis also tells you whether an OE model is adequate. Then you have the non-parametric analysis in the time domain, the step response for example, telling you giving me an estimate of the gain, you can compare the gain of that step response model with the gain that you obtained here with this parametric model.

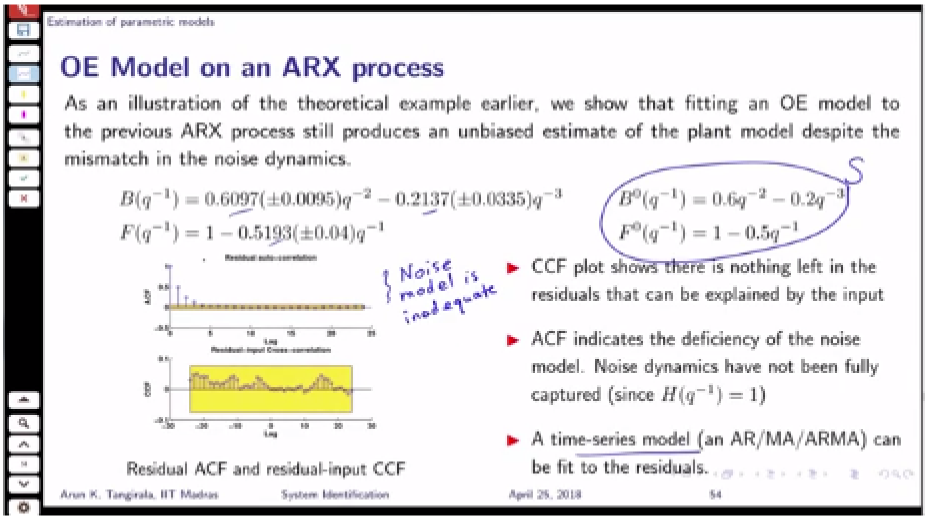
So, you should make it a habit to cross check and exploit this redundancy that prevails. Because then it will give you a sense of confidence and faith in your model. Okay. So, in this case, of course, the OE model is inadequate with respect to noise dynamics, and therefore, you will need a time series model maybe an AR or MA or ARMA which you can decide based on the ACF and the PACF. Looking at ACF, what is your proposition? Do you think you'll fit in AR model and an MA model or an ARMA? Or it's hard to say I have to look at the PACF?

Student 1: MA3.

Don't go with the knowledge of the process. MA3. So that is an option, right? But if I look at PACF because we are seeking parsimony always, if the PACF suggests a AR model will fit an AR model, right? I'm not showing the PACF here, I leave that as an exercise to you. You can go back and simulate this process; estimate an OE model, the MATLAB script is available in the textbook.

(Refer Slide Time: 06:40)



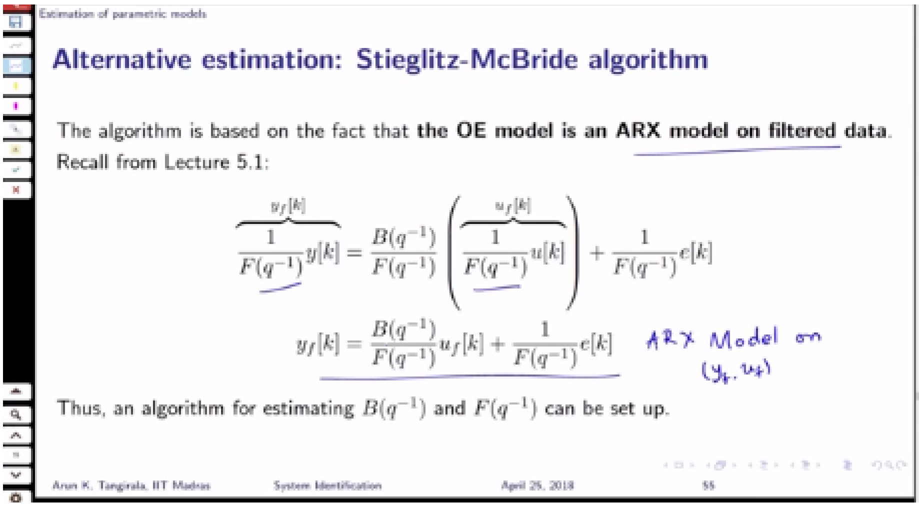


You can implement that and then look at the PACF, determine whether you should fit any AR model or not?
And once you do that, you can use the noise and the plant model together to fit a Box-Jenkins model if you wish. So that is a general procedure and the things to watch out for when you're fitting and OE model. Any questions? Okay, so let's move on and quickly study this alternative way of estimating an OE model. The estimates that we saw earlier in the numerical example were obtained through a quadratic PEM method. But you could-- we know that the quadratic PEM methods leads to a nonlinear least squares.

On the other hand, you could estimate the parameters of an OE model using linear least squares method alone, but this time using the pre-filtering idea. So pre-filtering enables certain nice things. And here the idea is that the OE model is ARX model on filter data. So, every model to an ARX structure, then straight away you can use the linear least squares method. In this case, we have already discussed this once in the class, before when we are talking of models structures were we showed that the OE model is nothing but an ARX model on pre-filter data, where the pre filter is 1 over f, that 1 over f is a part of your plant model already.

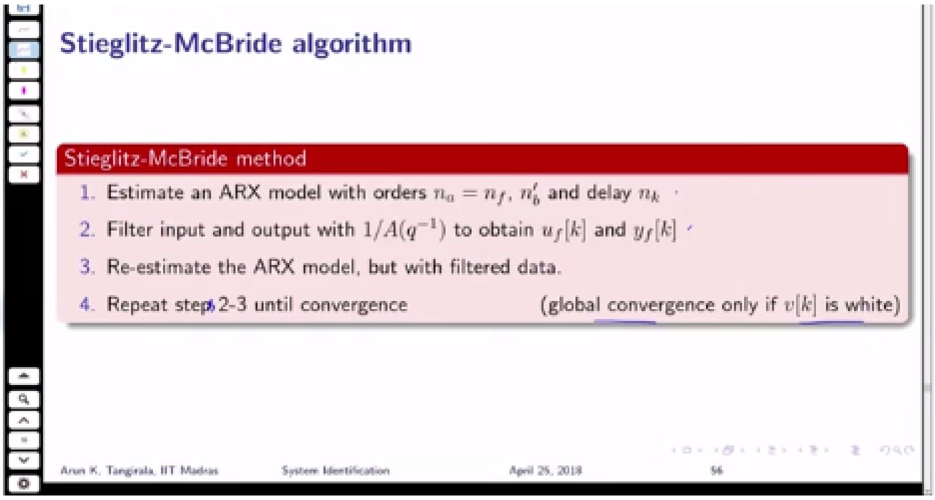
So if you know replace your yk and uk with yf and uf, then were yf are given here, then you end up with ARX model on the filter data. So here we know the pre-filter is 1 over f, but if you want to estimate now-- so this is an ARX model, but remember on yf and uf and you can use the linear least squares method to estimate b and f right? But who will get me the yf and uf is the question. And that's where an iterative algorithm was proposed by Stieglitz and McBride, who suggested that you first estimate an ARX model ignoring the filter, get some initial estimate of f, all the way right f of f, in the ARX model notation that we are follow it is A. So you estimate an ARX model, get the A for the ARX model, use that for pre-filtering and then continue until convergence. That is the idea.

(Refer Slide Time: 09:22)



So these are the four steps. Of course, it's assumed that you already estimated the delay through non-parametric analysis. So is the first step. Then filter the input and output. Re-estimate the ARX model but now with filter data, and repeat steps two to three until convergence, right? And studies have been done, sorry on the convergence properties of this algorithm, and it was shown that global convergence is possible only if v k is white. For all other disturbances, maybe, I mean, it may converts globally around. But in general the Stieglitz-McBride method will give you decent estimates. You can use this to even initialize your OE model. You had a question? Okay.

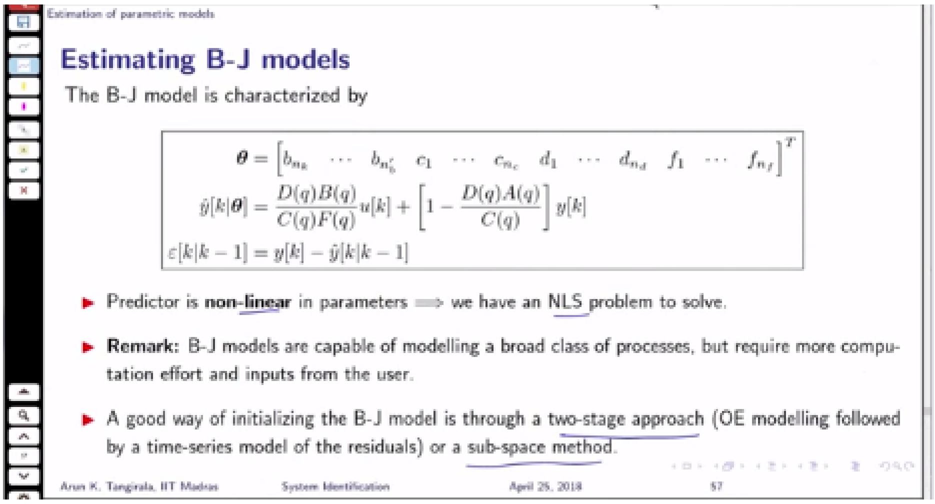
(Refer Slide Time: 10:21)



So very fairly easy algorithm to code. I don't think the system identification tool box carries any routine that is devoted to Stieglitz McBride, you can write your own. Okay, so we now move on to the estimation of B-J models, which is the most general parametric model that you have. And as usual as expected, the predictor is nonlinear in parameters and we have a nonlinear least squares problem to solve. The important aspect of this Box-Jenkins model structure is as we know the noise and plant models are independently parameterized. And therefore they can model a broader class of processes, but then the price that you pay is the computational part and estimation burden, because the user is required to guess orders or four different polynomials.

In order to minimize that burden, you can actually use a two stage approach, which is what I've advocated earlier. Begin with an OE model; get a good estimate of the orders and the parameters of the plant model. And then do a time series modeling of the residuals. Get a good estimate of the orders of the noise model and the estimates, and then use that as an initial guess for Box-Jenkins model. So, that makes the problem a lot more easy, or you can use a subspace identification method which we shall discuss shortly. You can of course, formulate a pseudo linear regression approach here and so on but normally that is not used.

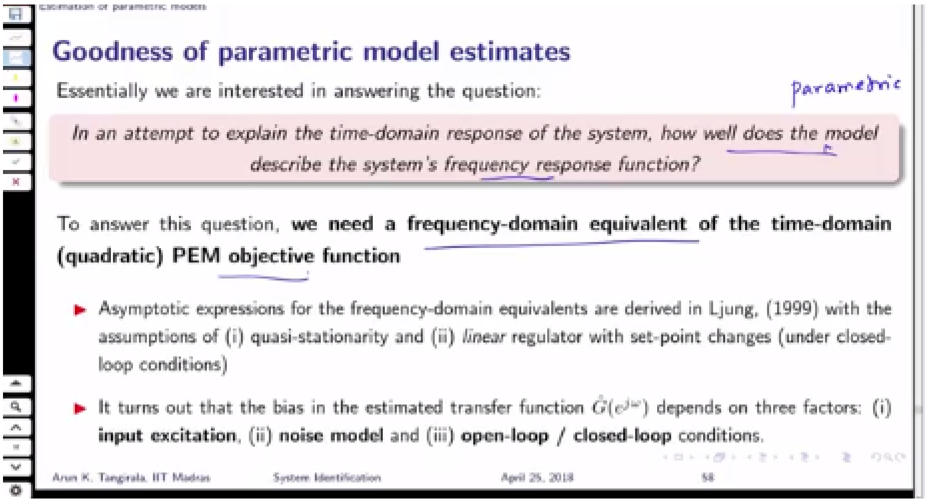
(Refer Slide Time: 11:57)



Right? So that completes the discussion on the estimation of different models structures, but it doesn't really bring an end to the discussion on this PEM methods. We would like to answer one more question which is that, how well does the model-- this parametric model described the system frequency response function? Earlier in the non-parametric method class we have learned how to estimate the frequency response functions non-parametrically, but we always have the option of estimating these impulse response, step response and frequency response in a parametric way. Pretty much like estimating power spectral density, if you recall power spectral density can be estimated in non-parametric as well as parametric means. So here we are interested in knowing if I take the parametric route, where the model structure becomes an important part of the exercise, how well does the resulting frequency response function estimate turn out to be? How good does it turn out to be? And as expected, the answer will depend on the model structure. Right?

Now, to answer this question, we need a frequency domain equivalent of the time domain objective function. So what is happening is you're operating in the time domain, you're estimating things in the parameters and the time domains and so on, but behind the scenes remember, time and frequency domains are connected to the thread of Fourier transform.

(Refer Slide Time: 13:34)

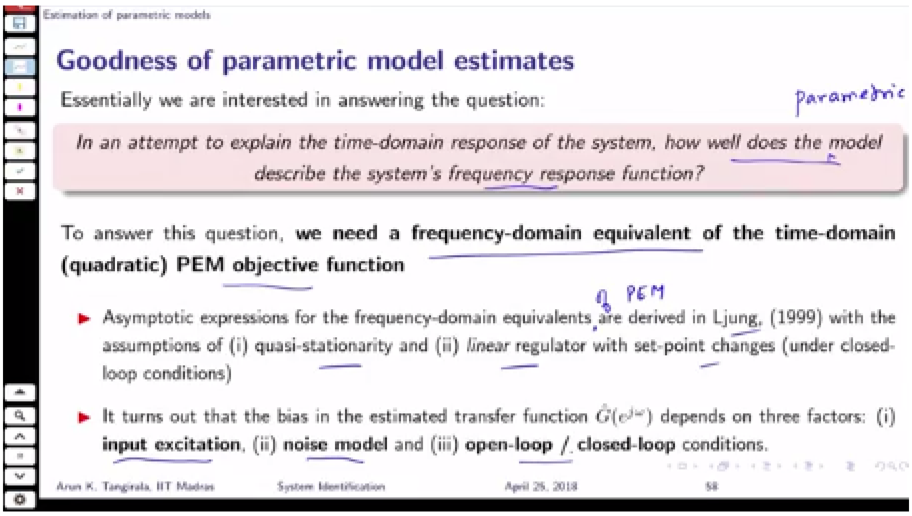


As you're operating here, there is some operation being performed in the frequency domain whether you like it or not.

Since we want to know, what is a goodness of the parametrically estimated FRF, it's natural to turn to the frequency domain equivalent of the PEM objective function, right? And the asymptotic expressions for the frequency domain equivalence of the PEM that the objective function that we have seen in the previous lecture are derived in [14:11 inaudible] book under these assumptions, okay? When I say linear regulator that is for closed loop conditions under open loop conditions, you don't have to worry about that. And I will briefly discuss, I'll review the expression and will interpret those expressions. But the saramshor the summary of the findings is that, remember we said there is a bias likely in the estimates are the parameters depending on the data generating process and the model that you choose, right?

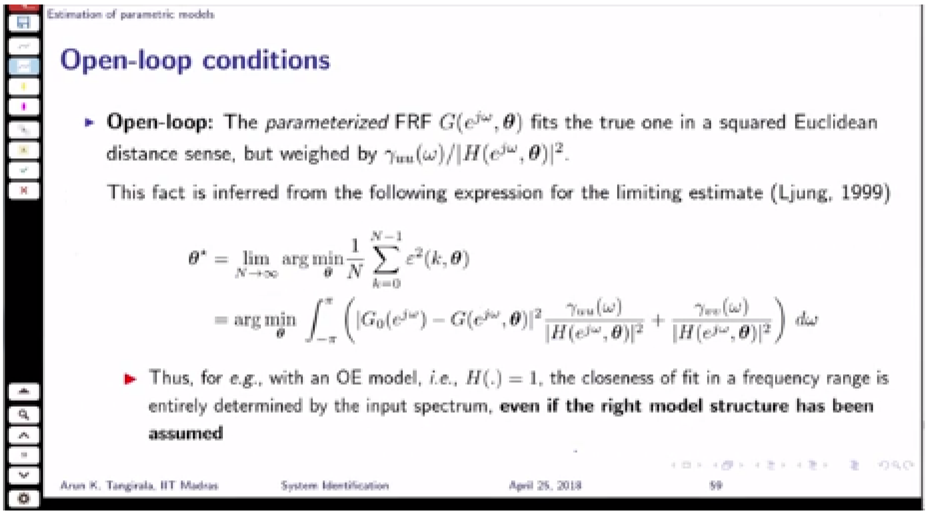
And we learned through previous lecture that this bias depends on the input, you recall example that we had where the data generating process was OE and we were fitting in ARX model, we realized that when we assume the noise, sorry, input to be white, then one of the parameter estimates is unbiased but the other one is not. Overall, you should expect the FRF to be biased, naturally. And that bias now we know will depend on the input excitation, the noise model, and whether you're looking at open loop or closed loop conditions.

(Refer Slide Time: 15:21)



So these are the three things that will determine the goodness of the parametrically estimated PEM, sorry, the FRF through PEM.

(Refer Slide Time: 15:30)



So it turns out that the frequency domain equivalent of the quadratic PEM, this is quadratic PEM objective function called the quadratic criterion PEM, QC PRM, not quadratic. When you write down the limiting objective function to this in the frequency domain, this is what you see. So now it's time to interpret this. So, look at it there are two terms here, right? What is the first time tell you? What is the first time quantify? The square of the bias, right? G naught is the true FRF, that is a systems FRF and G of e to the j omega comma theta is the estimated one or we need to find that. But anyway, so this will be the parametrically estimated FRF.

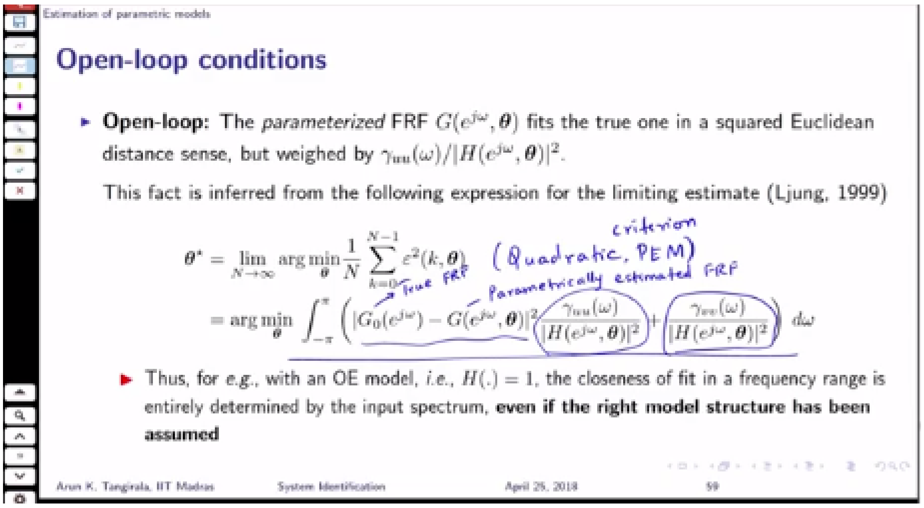
What is expression tells us is, when I am working with a quadratic criterion PEM to estimate the parameters of a parametric model, the bias that is error between the true FRF and the estimated FRF is being weighted by this factor here, right?

This is the factor that shapes the bias. What do we want? We want an unbiased estimator. Correct? That means, I don't want really necessarily, unless the user says that I would like the bias to be zero at all frequencies that is fine, but also the bias to be uniform at all frequencies. I don't necessarily require good estimates over a certain range of frequencies only. I want the bias to be minimum at all frequencies. This is n going to infinite in practice, it's not possible to drive the bias to zero for finite n, right?

A and what this expression tells me is that quadratic criterion PEM essentially places this kind of a weighting function, in other words, it is solving a weighted least squares problem in the frequency domain. There is also this term; it is trying to minimize this also. So, there are two terms that it is minimizing. But among the two one contains G the other one contains only H. One contains both G and H, other contains H rather than saying one contains both G and H, we can say that the bias in the FRF estimate of the plant model is being weighted by this factor, and this factor here gamma u u of omega by square of H of e to the j omega. We will determine, how good the estimate is going to be. For example, with an OE model, we know His 1.

When H is 1 the weighting completely depends on the input. If the input is wide, then that means you have uniform importance to all frequencies. So, there is no particular bias. Your model will not be necessarily good only in a certain range of frequencies. Whatever goodness exists, it will be the same at all frequencies.

(Refer Slide Time: 19:16)



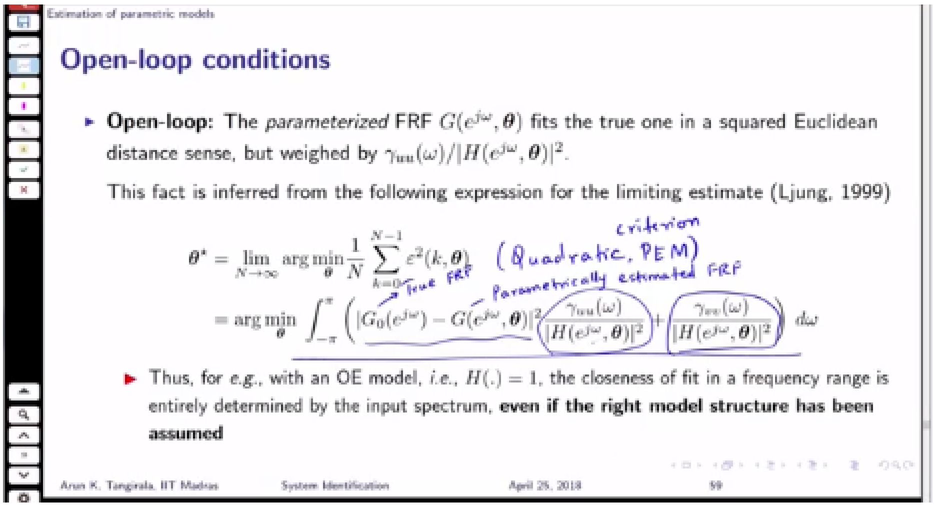
On the other hand, if you use a colored input and trying and then estimate an OE model, then clearly because the input itself contains only select frequencies, your model will be good only in those frequencies.

So, if your input is colored which means that gamma u u of omega is not flat. it has more power, let us say in the low frequencies as against high frequencies that means more important will be given to the model at low frequencies. If low frequency fits will turn out to be better than the high frequency ones, right? So, your input is now this expression was critical to understanding what are the design parameters that are available for the user? What is it that the user can play around with, so that the user can shape the bias, or user can control to what extent, the bias can be minimized and so on? In practice, generally we use only colored inputs; we generally don't use white noise inputs, although they are preferred. Simply because we know that the most of the systems that we deal with are not all-pass filters. They are either low pass, or high pass or band pass and we will focus our identification effort only in that range, and that will be the demand of every application. So it is okay to do that. But what is nice is when you choose an OE model structure, the bias is completely determined by the input alone, which means that not much you can do post experiment.

Now with non OE models, what can you do? Right? Suppose I use a non OE model and suppose I use a white input? I've used let say white noise like input, but I'm using-- I'm fitting a non OE kind of model, then H is not 1. Then what kind of models we like it. So there is an example to that effect in the textbook you should go back and check. For example you have fitting ARX, what is H for ARX?1 over A, right? So 1 over A is a low pass filter. Right?

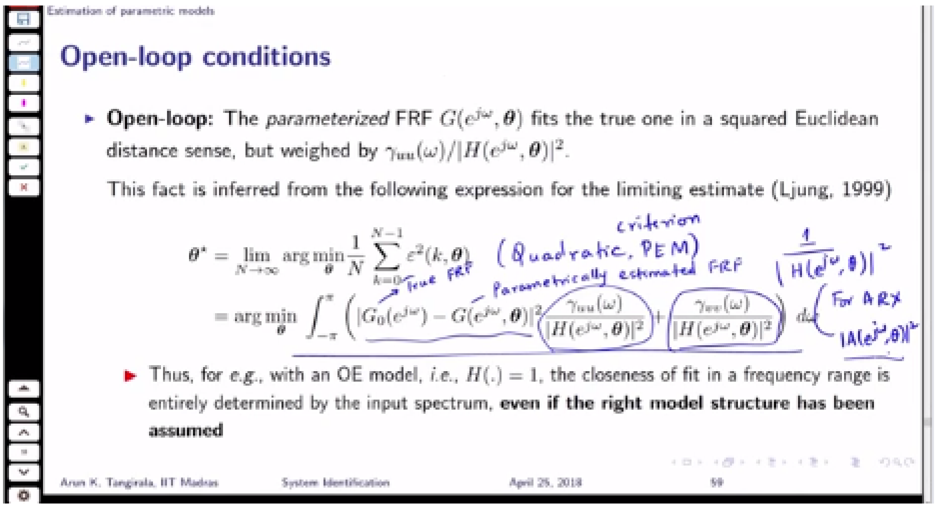
Typically 1 over A will have, if assume that 1 over A has low pass filter and characteristics, that means this weighting factors will focus more on the high frequencies, because what is happening is you have 1 over H here, if you leave aside gamma u u, you have 1 over H of e to the j omega comma theta whole square.

(Refer Slide Time: 21:20)



If you're fitting an ARX model, right, for ARX what do you get? This quantity turns out to be simply A of e to the j omega comma theta whole square. If 1 over A is low pass, A high pass, which means that this quantity will have larger values at high frequencies and smaller value at low frequencies. In other words, when I am fitting an ARX model and I'm using white noise input, the ARX model tends to place more importance on the high frequencies. Of course, I can counter that by designing the input accordingly, but I wouldn't know it's a cache 22 problem, how do I know what the inputs should I estimate the model, right? The other alternative is to pre-filter. You pre-filter the input with the inverse of the noise model.

(Refer Slide Time: 22:44)



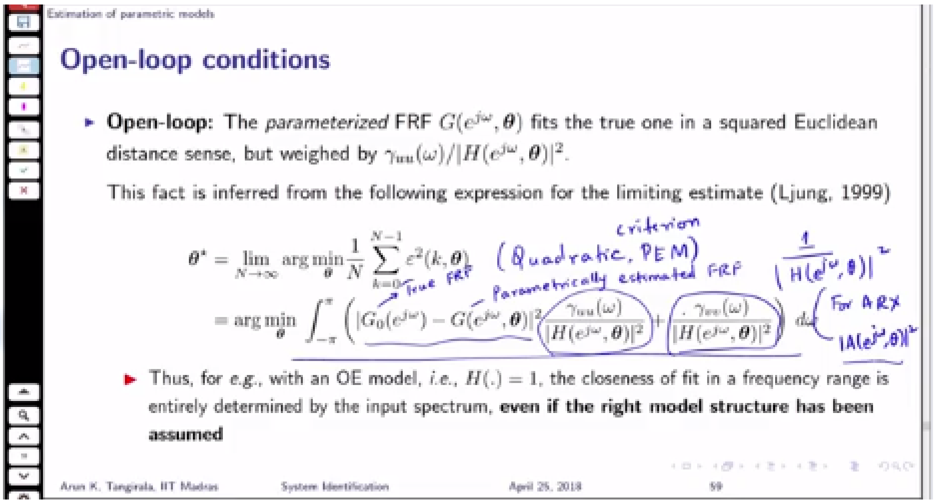
Then that amounts to actually, if you're doing that for ARX then it becomes an OE. So the same interpretations are appearing, but what this expression tells me is that the bias is shaped by two things. One, the input itself, input spectrum and other is the noise inverse of the noise model. If the noise model is fixed, then the soul factor is the input itself. If it is not, then for a given input, the inverse of the noise model will determine what the bias is, right?

And if you look at the second term, you're minimizing 1 over H gamma vv of omega, anyway gamma vv of omega is not in our hands. So we are essentially minimizing H. If you choose H, then there is nothing to minimize there. If you choose H equals 1 for OE models, if you've already chosen H to be

1 then there is nothing to minimize, otherwise it will try to minimize H in such a way that this second term is also minimized.

Their G has no role to play. But what comes out beautifully is the statement that we made long ago at the beginning of the course, which we said that the noise model plays a critical role in the goodness of your plant models. If you make-- depending on the choice that you make for the noise model, you will get a different plant model, right?

(Refer Slide Time: 24:15)



And now we see the quantified expression of that. Any questions?

Student 2: If this objective assumption is not the same as the limited objective function involving in generalized expectation of a [24:33 inaudible]

More or less the same thing? In this model it's limiting right? So, it limit N tends to infinity, they don't expectation here that's all we think. It's just an asymptotic. There is no averaging across realization. Okay? Fine, so now that kind of brings us to the close on PEM methods. There are other issues with PEM methods. PEM methods are good, but there are other issues such as sensitivity to initialization, being able to handle multi variable methods, multi variable systems and so on. But by enlarge-- when you have large samples prediction error minimization methods are very good. But when you have small number of observations, PEM can be really bad, because it's after all MLE, you know, so small sample properties are pretty poor. Although there exist no expressions, for those you can expect that at least what you're guaranteed is large sample property circle. That is what you're guaranteed.