CH5230: System Identification

Estimation of parametric model

Part 3

So, I have now an OE process, the 0s on the parameter, superscript of 0 on the parameters indicate, it's a true process. And it is being excited by white-noise input is extremely important. If you change the input the results of this example can change. Now, the question that I have is suppose an ARX model is assume, is fit and I say assume is fit to the data. I've written it this way, but the coefficients are equal. In fact if you write it down in transfer function formwe'll get an OE.Yeah. OE in fact is an ARMAX, special case of ARMAX, right. But that's not how it should be viewed. So you're convinced know that it is OE, right. So first of all what can you say. Will this ARX model that I'm planning to fit. Will it, will the estimates of a1 converge, will the estimate of a1 converge tof1 0 and will b1 converge to b10.

What is your answer? Sorry?

[01:30 inaudible]

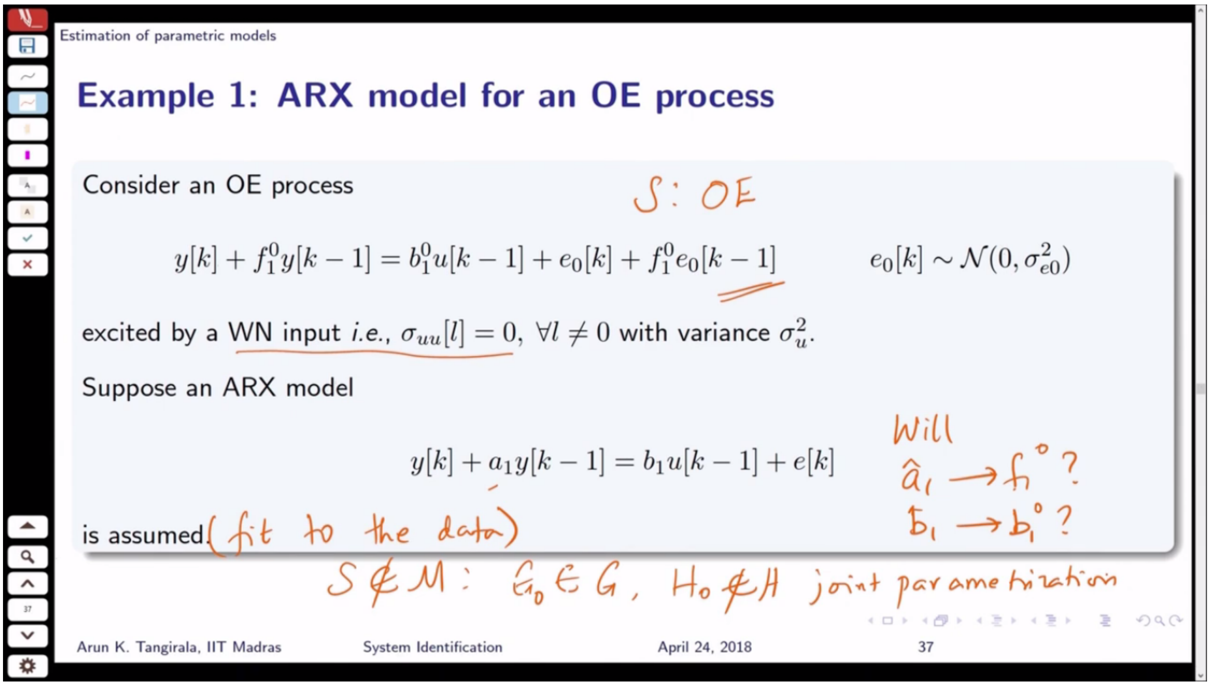
How can you say that where is the result?

[01:39 inaudible]

No, you can't do that. Based on the results that we discussed. You're not allowed to use the results from a previous birth. Right? So, based on the discussion that we had until know. What do you think? It is guaranteed?In general, suppose I don't talk of the nature of the input. Suppose I take away the fact that it is not a white-noise input, then it's a different problem from the one I gave in theassignment. Now what is your answer? We have discussed at length, no, the cases. So you just have to ask if S belongs to M, S does not belong to M and so on.To answer this you need to ask which scenario it fits into? Does S belong to M? What do you think, Suman? It doesn't. Correct. So we know that S does not belong to M. Okay. Does G belong to G0, right? G0 belongs to G, I am sorry. Right? What about H0 does it belong to H? What do you think? Does H0 belong to H? What happened, Purna? You seem to be lost. What is the confusion? Why there is so much silence with regards to H0 belongs to H? Yes or no. If it belongs to then S would have belong to M now. But why there is a cloud of confusion there? I know that we are living in a cloud era, but why there is a cloud of confusion? Okay. I leave it to you to ask. Maybe you are thinking how to ask. So H0 belongs to that does not belong to H.

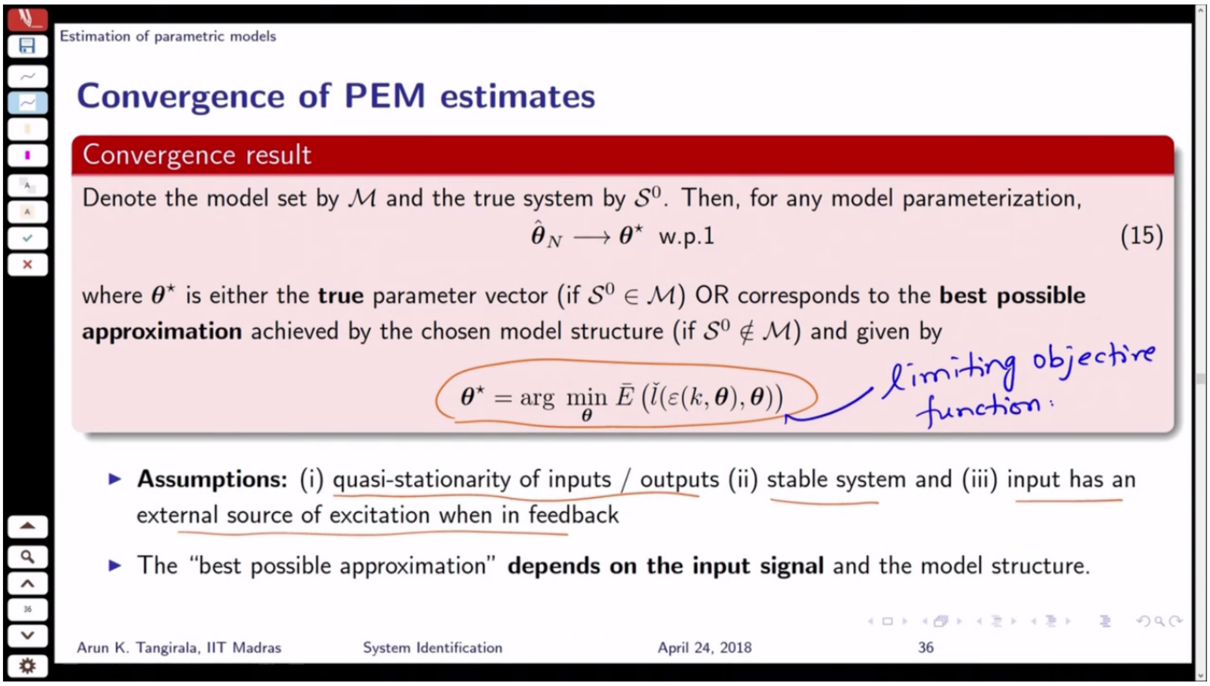
And what else do we know about this model parameterization, jointparameterization. So we know that for sure. In general I cannot guarantee. In general I cannot guarantee that a1 hat will converge to f1 0 and b1 hat will converge to b1 0.In general I cannot. That should be the answer. Then we will ask. Okay, what do they converge to? So there is no guarantee here. That is what is first the statement. Now what do this converge to when N goes to infinite. They should converge to some values, right. That is where we start doing our analysis.

(Refer Slide Time: 04:39)



So first step is, remember, what does the theorem say? There is theorem says that if you want to find out what it converges to then it will converge to this. That means, what you do is. You solve this optimization problem analytically. That means you find out what is the expected value of the objective function. First you determine that expression and then you minimize that the solution, the optimal solution to the limiting objective function. This E bar of l inverse cap, whatever you see that is called the limiting objective function. This is called the limiting objective function. That means in the limit as N goes to infinity. Okay.

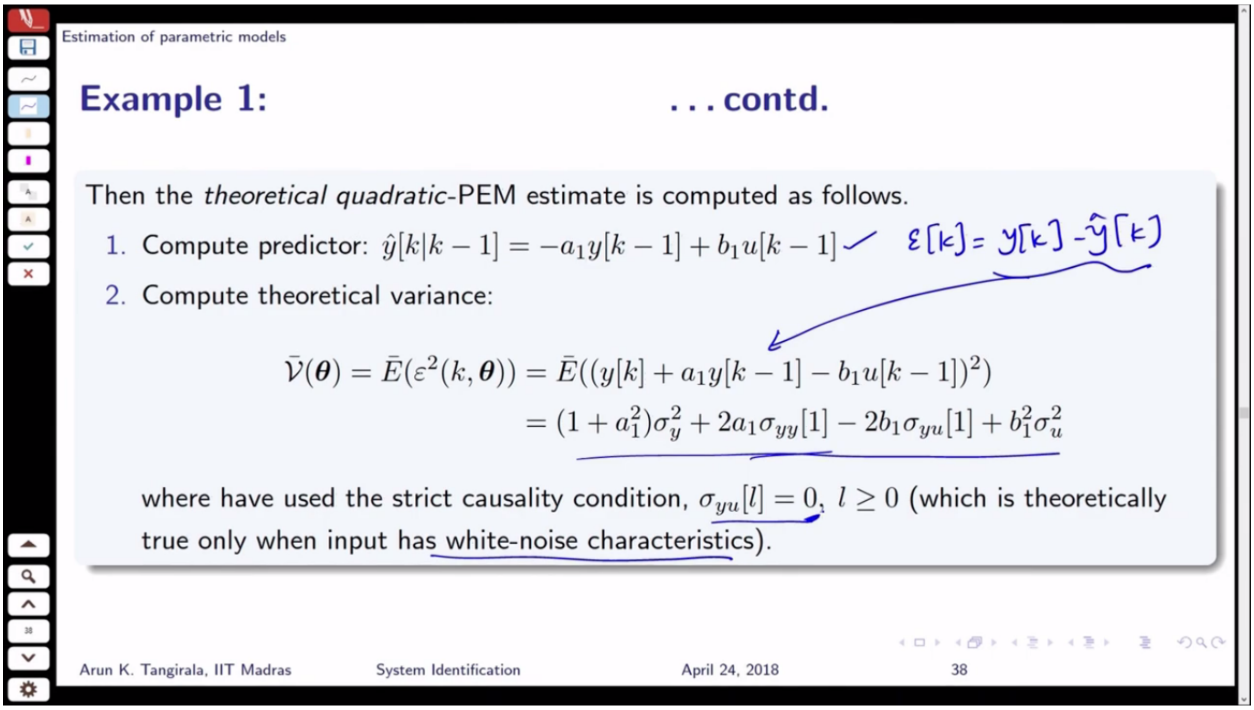
(Refer Slide Time: 05:41)



So now we need to find not the limiting objective function.To do that first we have to construct y hat, because the limiting objective function depends on the prediction error. So here is a prediction. We know that for the ARXmodel this is a predictor. Then we compute the limiting prediction error which is E bar of epsilon square. We'll assume by default if I don't state anything it is cube PEM that is quadratic-PEM. Which means l inverse cap is simply square.So E bar of epsilon square, all you have to do is square the prediction error. This is prediction. We know what the prediction error is epsilon k is simply yk minus y hat of k. And this is what essentially we have put in. All right. So I have here yk plus a1 yk minus 1 minus b1 uk minus 1 to the whole square. And I want to take an expectation of that E bar for all practical purposes here is nothing but your standard expectation.

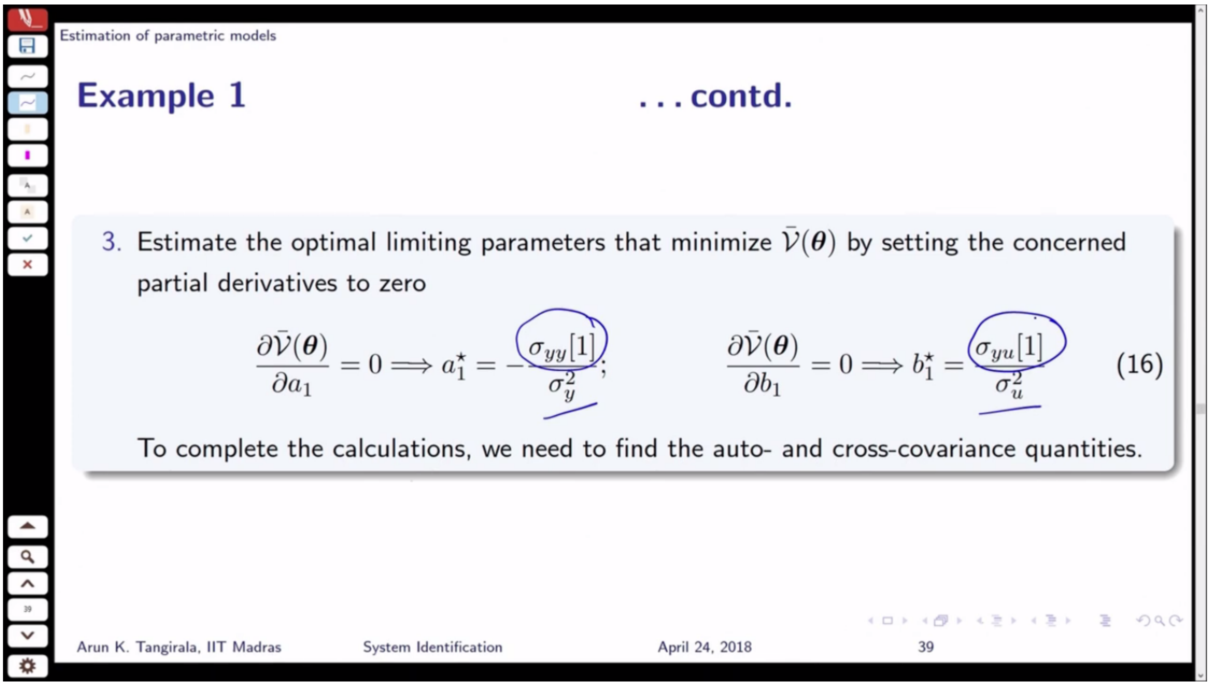
Now you evaluate the expectation and you get this explanation. This is your standard derivation that you do. And we have assumed that sigma yul of 0 is 0. In deriving this we have made use of the fact that the input as white-noise characteristics. This expression here will change if the input changes. Right. Because they will be across term that is this cross covariance between y and u. You know again, they'll be auto-covariance and so on. So in doing that in evaluating that we have used the white-noise characteristics. If the input is a coloured one, then this condition here will be at stake, remember that.That is where the noise of the input makes a big difference. Now you start seeing what is happening. What we have realized is that ARX model will not converge to the true one that we know.In general there is no guarantee. What will it converge to some optimal approximation of the process.

(Refer Slide Time: 08:12)



That optimal approximation, how is it derived? That is what is it that what is it governed by the limiting objective function. The limiting objective function expression depends on the input. In other words. The approximation that your model is going to get you depends on the inputs that you have asked the questions that you have asked. So now, we differentiate the limiting objective function with respect to each of the parameters and you get this result. Right. And for fully to complete the calculations what do we need?We need expressions for these two.And how do we get that. I have to go back to the data generating process and ask, what is auto-covariance of y at lag 1 and cross covariance between y and u at lag 1?

(Refer Slide Time: 09:05)



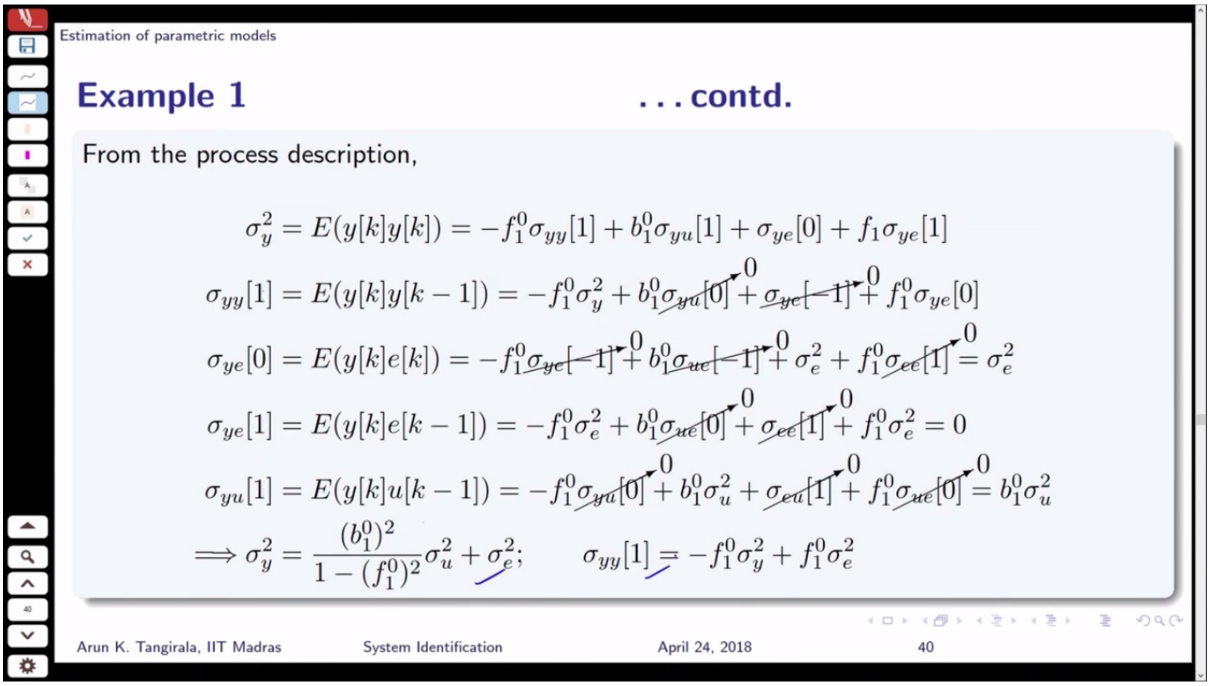
Once I do that. And standard expressions you start with your data generating process equation get your sigma yy of 1 and essentially you get these two terms. Yes.

No. If you don't know the GDP the entire question falls apart.

Then we can use sigma [09:28 inaudible]

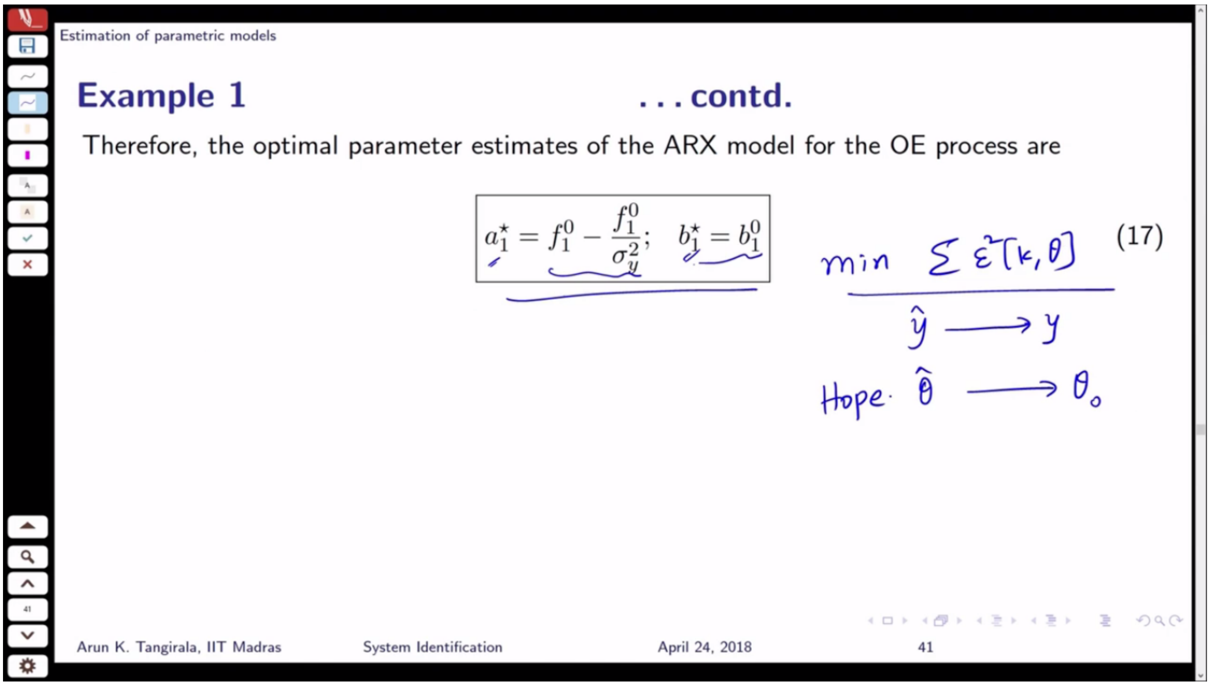
Yeah. In practice you will have to use an estimate, but here we are doing a theoretical analysis. Okay.So just want us to understand when there is a mismatch between the process and the model, where does the model go and fit in. So if you look at the consistency conditions, what it says is that if there is no mismatch in terms of the belongingthis, then convergence is guaranteed. But if there is a mismatch in the noise but not in the plant, then if they're independently parameterised the G will converge to G0. And the third situation is there is a mismatch both in the plant and there is a mismatch in the noise model, but G0 belongs to G but jointly parameterised then there is no guarantee. All right.The other scenario anyway there is no guarantee. That is if G0 does not belong to G and H0 does not belong to H, then there is no question of guaranteeing consistency. The next question that begs our attention is what approximation will I get? How far is the bias?That means essentially I'll get biased estimates. Will all the parameter estimates be biased?

(Refer Slide Time: 10:37)



But only some will be biased, right. Which ones are those? How does the input play a central role in that? So this example really answers many questions, because this will tell us then if I use certain class of inputs. Then maybe I'll be able to achieve at least estimate certain parameters without bias. So when we do all this analysis and plug in then here is what you get. So what do we observe?When I use of a white-noise input b1 hat converges to b1 0, the star essentially means it's a limiting estimate, right. So I have here a1 star, which is the limiting estimate.As N goes to infinity a1 hat will go and sit at this value and b1 hat will go and sit at the true value. Isn't it very interesting? And you can verify this through simulations also. All I have to do is generate data. Maybe with N said to some hundred thousand observations. Right. And fit an ARX model and see that there is a systematic bias in fact, you can take an average across all realisations and see that the average estimate as N goes to infinity N becomes very large for a1 will be biased. Whereas for b1 will not be provided to use a white-noise input. If you're not used white-noise input or some coloured input, then both estimates are going to be biased.

(Refer Slide Time: 12:15)

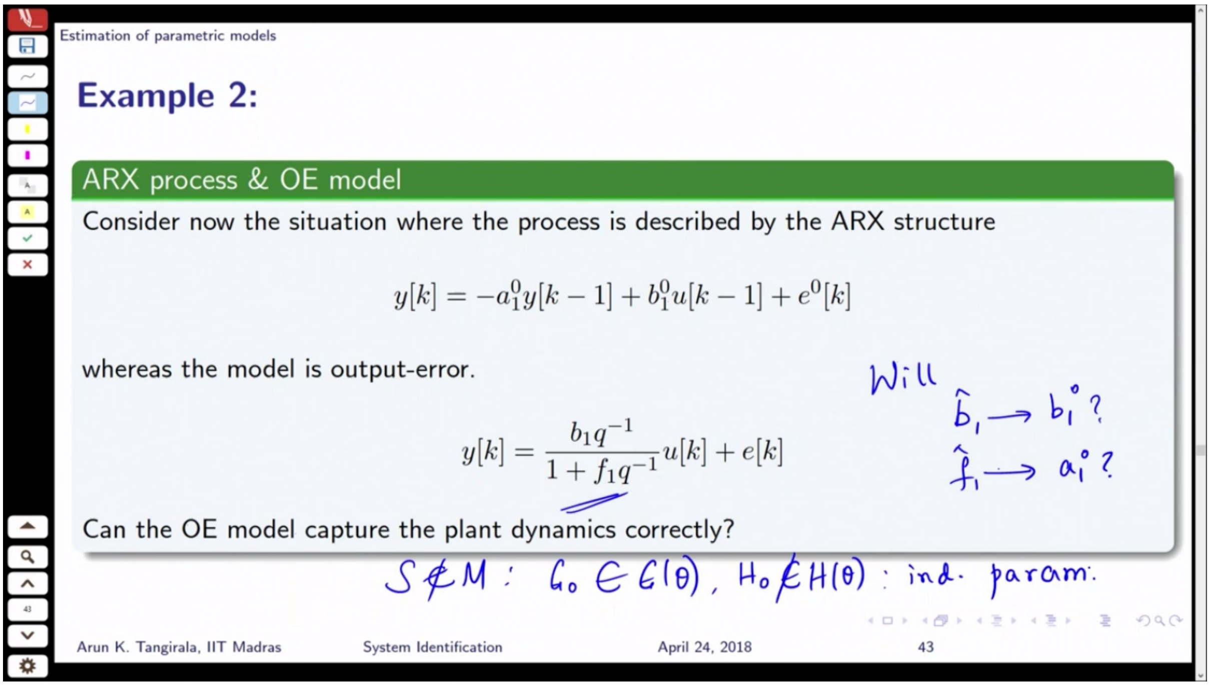


Now at this point I want to tell you something. Insystem identification the central discussion for us always has been convergence of models. But there exists also other objectives in system identification.I don't care about the model. I just want my model that I have chosen for whatever reason to generate optimal predictions or to minimize prediction errors. This model does that because you have asked for minimizing prediction errors. So what has happened? The data is being generated by some process. You have assumed a different model and then you are asking. The optimiser to fit the model such that the prediction errors are minimized, you are not asking it to find the correct model. See the problem statement says minimize sigma epsilon square k. So what is the optimiser going to do?It says, I don't know about your model and the truth and so on. I don't know about any of that business. You have given me a model and there are two parameters in my hand which I can freely vary. I'm going to find my best ways of figuring out how to choose that such that this is achieved.

So truly do you think in the objective function statement of PEM methods, there is a statement telling the optimiser, please find the true model. No. There is nothing like that, right. It is we are hoping that as y hat is being driven to y as close as possible. We are hoping that theta hat will also go and sit at theta 0.That is a hope, right. But is that happening? It happens only if you choose your models correctly. Right. So one has to be careful in remembering all of this, if you are hell bent on getting, I mean, if the objective is to get the right model, then definitely you will have to choose the model structure and so on, such that it meets the consistency requirements. But if the objective is to get some, is to fit a model that minimizes the prediction error among all the ARX models that you can fit, this one gets you the least prediction error. That it has achieved. Okay. Remember that.

So we know that the least squares estimates, I mean you can look at it this way. Why did I get a biased estimate? Other perspective after all quadratic-PEM, what does it simplify to for fitting ARX models? It will simplified to linear least squares. And what do we know from linear least squares. Whenever the errors that you have left out the residuals that you are left out are correlated with what you have included. You will get biased estimates. So in this case the true process has this form. This is what you have left out. Structurally this is what you left out, right. And what are you regressors? Yk minus 1 and uk minus 1. Those are the ones that have included structurally in your model. Obviously, whatever you have left out is correlated with what you have included. And therefore you should expect biased estimates. So this doesn't come as a surprise. It is a corroboration of what we know already, but we have used a different route to arrive at this result. Okay.Likewise now you can flip around and ask if the data generating process is ARX and if I fit an OE model. What do you think is the answer? I cannot ask if the noise model will be correct, because anyway I have force a noise model to be 1. So my focus is only on this. So in the questions now are again the same.Will b1 hat converge to b1 0, f1 hat converge to a1 0. What is your answer? Yes. Because first of all we know that this is Snot belonging M scenario. And we know that G0 belongs to G of theta. And of course, needless to say H0 does not belong to H of theta, but independent parameterisation. So by invoking the conditions of consistency we know straightaway that this will be the case, but let us use that limiting function theorem, limiting objective function theorem and see if indeed it happens.

(Refer Slide Time: 17:12)



So the solution is more or less on the same lines, butwhen you drive the one-step ahead predictorit is useful, if you would write it this way. Okay. So this is your, I'll just read it in the model this way, so that it becomes to write easy to write the predictor and I can use a projection theorem. So the least squares estimate. Are obtained by simply, because PEM now is essentially non-linear least squares I can use a projection theorem. Yes.

[17:51 inaudible]

Correct. Of course, yeah, yeah, yeah.

[18:02 inaudible]

Since if you are fitting an ARX model for [18:10 inaudible] process, if he had scaled with error [18:13 inaudible]

Correct. Correct. You're right, absolutely.

So then when we are going [18.19 inaudible]

Correct. Exactly. So then the model structure will change. The moment you see scaled one.So you will have to now compare the scale data. See when you're talking of models system belonging to model there is an input that is an output right. When you scale situation will change. So what is the scaling that you were thinking of?

That alone will not help. That alone will not help. I know that you are trying to invoke some ideas from multivariate data analysis. But the situation here is simple scaling alone won't help.You will have to actually also do some filtering, pre-filtering.Here the issue is not that you seem between a principal component analysis and IPC homo scholastic errors versus hydro scholastic. No that is not the issue the issue is that the filters are different. So if you pre-filter the data.So that your ARX becomes an OE, then things will change.But then the conditions will also apply differently the scenario changes. Okay. Anyway, so coming back to this problem you write this one-step ahead prediction and then you solve the,you set up these two equations. I'm just solving the limiting objective function thing in a slightly different way, because the results are going to be the same whether you minimize in the least square sense or whether you apply the projection theorem the results are the same, because projection theorem is offers an alternative way of solving the least squares. So we get two equations here. And once you evaluate from the process you should verify that this is what you get. And ultimately we get the golden result..

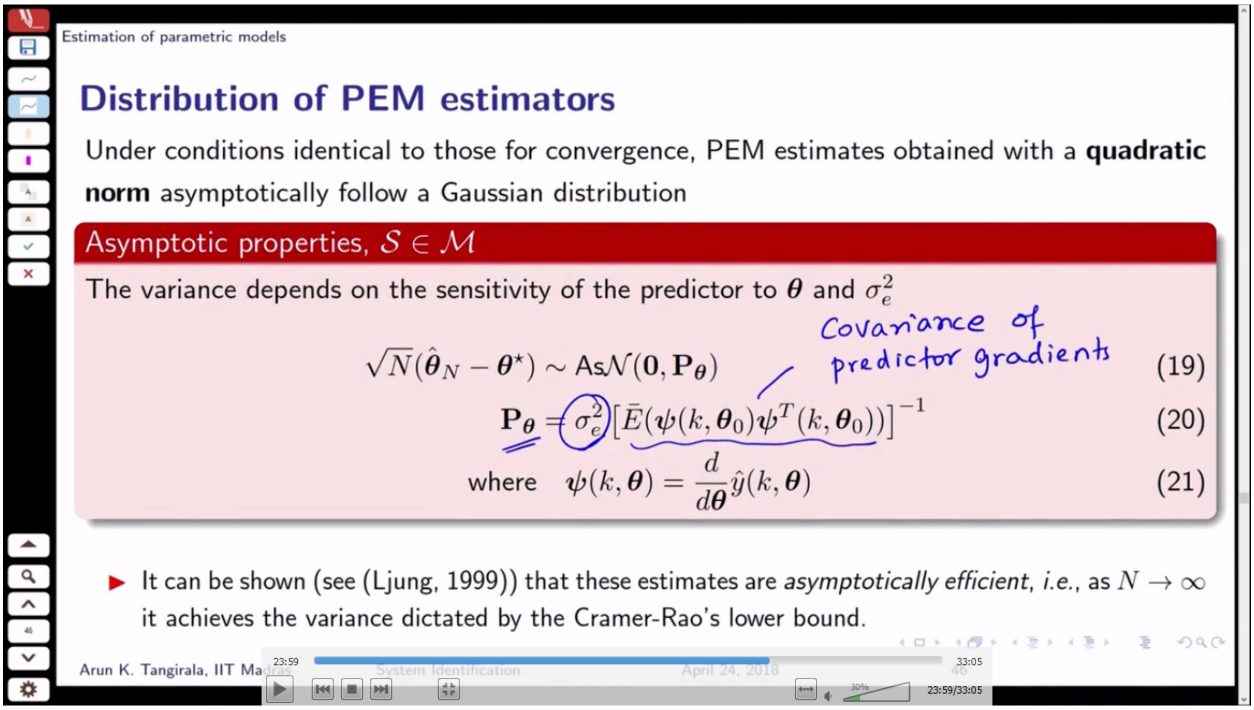
Okay. So that is why the OE models are always preferred a starting points. The only difficulty with the OE model it is the non-linear estimation problem, but that's okay, that's not a big deal.At least it gets you a good. It allows you to focus on G alone. And then once you get your G right, you can get some decent estimate of the time series model. Use both of those to estimate the full model, that is Z. Yes.

[21:00 inaudible]

Yeah, but I'm not so sure if. Unfortunately you have to estimate them, to get optimal estimates you have to get estimate them jointly. In this case G0, the fourth scenario we'll have to see what happens when do you actually get that. What are the results will, I have to go back and check whether. If G0 does not belong to G, then will it be now the same, you know H that I estimate will it go and say. It may converge to the truth as N goes to infinity you may be allowed to decouple, but for finite N maybe that's not the optimal solution. Okay.So it's a good question. My guess is that as N goes to infinity you're allowed to decouple. But for finite N that may not be the optimal way, you may have to do it jointly. You may be pretty close. So you can use this as a starting points for your BJ model estimation. Okay.So let me just spend about 10 more minutes and then I know it's a heavy class lecture will conclude quickly today. Then we'll take up the remainder tomorrow.

So obviously now that we have studied consistency it's time to talk of distribution so that I can derive confidence regions, without going into the proof and so on you can expect things.By the way all this results are for quadratic-PEM that we are talking about. These results can change if you change the norm by the way. So it say, the results said that asymptotically theta hat, sorry as a Gaussian distribution. With this error covariance and remember that this error covariance matrix requires a knowledge of sigma square E. And this is asymptotic so you're using some limiting values here.In practice the way you recognizes this is nothing but the covariance of what?Predictor gradients. In the least squares you would see Phi transpose Phi inverse, right. I mean,Phi transpose Phi essentially. So this is, in practice what you do is you replace sigma square E with its estimate. And you replace this inner matrix with the covariance that you estimate of the predictor gradients at the optimum that you have estimated.Where ever you stopped your optimization algorithm at that point you evaluate the predictor gradients and estimate, standard estimate your covariance, sample covariance matrix, computersample covariancematrix use that. And get keep going. And interestingly these estimates are asymptotically efficient that you should expect becauseit's Emily it's also non-linear least squares and so on. So it should achieve the Cramer-Rao's bound.

(Refer Slide Time: 24:21)

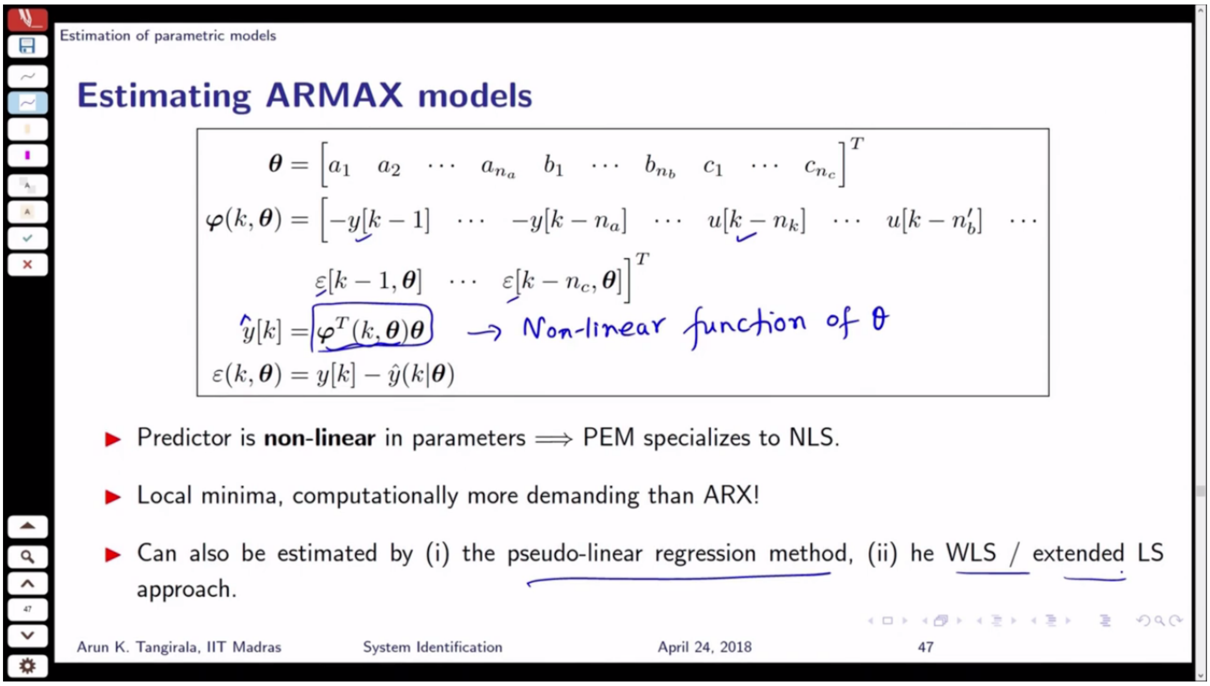


So what we have learned until now is the properties of PEM estimatorsthe philosophy behind it and so on. Now when it comes to specific model structures they may exist some specialized algorithms. So remember in general your PEM approach can lead to a non-linear least square problem except for ARX and FIR. For output error for ARMAX, BJ or any other model structure, the predictor is a non-linear function of the parameters and therefore you would be solving a non-linear least squares problem. In order to solve that non-linear least squares problem you may need some good initialization. PEM can be quite sensitive to initial guesses. To generate this initial guesses or to generate some decent suboptimal estimates of these model structures such as ARMAX, OEand BJ, there are alternative ways and now we just want to discuss some of those and we'll continue this discussion in the next lecture as well.So ARX models there is no need to discuss it’s a linear least squares problem. We already know how to solve linear least squares problem for ARMAX problems we know straightaway that the predictor it should not be yk, but y hat of k is a non-linear function of theta. What is also interesting is that for the ARMAX structure, if I give you a value of theta let us say now anyway we know that we will have to iterate. If I choose one theta at some specific iteration and I want to improve the theta at the next iteration. I can plug in the theta from the previous iteration into this matrix are into this vector. Then it becomes linear for the next iteration. When that happens we say that the predictor is pseudo-linear in theta. All right. That means for something to be called pseudo-linear, it should be of this form. So what I do is I use the theta from the previous iteration. Or I somehow construct my psi initially the regressor vector from some source, because what does the regressor vector require? It requiresy's u's and the prediction errors. y and u are given to me, prediction errors I will generate through an external method. They are not available on any of the

Amazon, Flipkart website, but I will have to generate them, I can't buy them, right. I cannotorder them online. But I will generate for example by fitting a high order ARX model to kick start the algorithm.

Essentially what this epsilons? These are nothing but one-step ahead prediction errors.So I will generate one-step ahead prediction errorsthrough some other model, high order ARX model and I will improve upon theta. Once I get the theta, I can use linear regression at that point in time. Use theta and then improve the estimates of epsilons. So that is called a pseudo-linear regression method, right. So that is the idea behindpseudo-linear. Of course you can use weighted least squaresand so on.I won't talk about it. It is just in principle you can use a weighted least squares. We have talked about it. Or you can use the IV method also, if your focus is only on obtaining the coefficients of the G that will talk about it later.

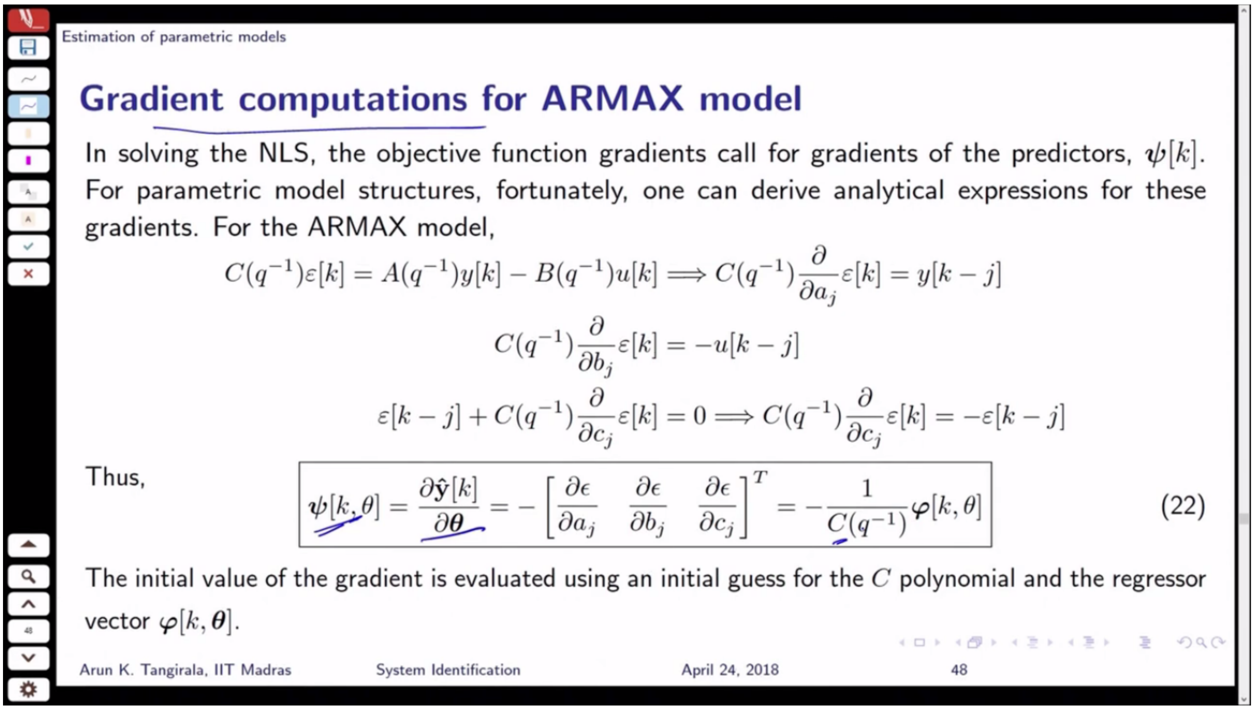
(Refer Slide Time: 28:21)



So the pseudo-linear regression method always exists. It's going to give you a suboptimal estimate, but you can use that to initialize your non-linear least squares estimator and the non-linear least squares estimator requires gradient computation, remember,because it needs ∂y hat by ∂ theta. And this slide I'm showing how to obtain that y hat by theta and it's essentially nothing but minus 1 over C psi.

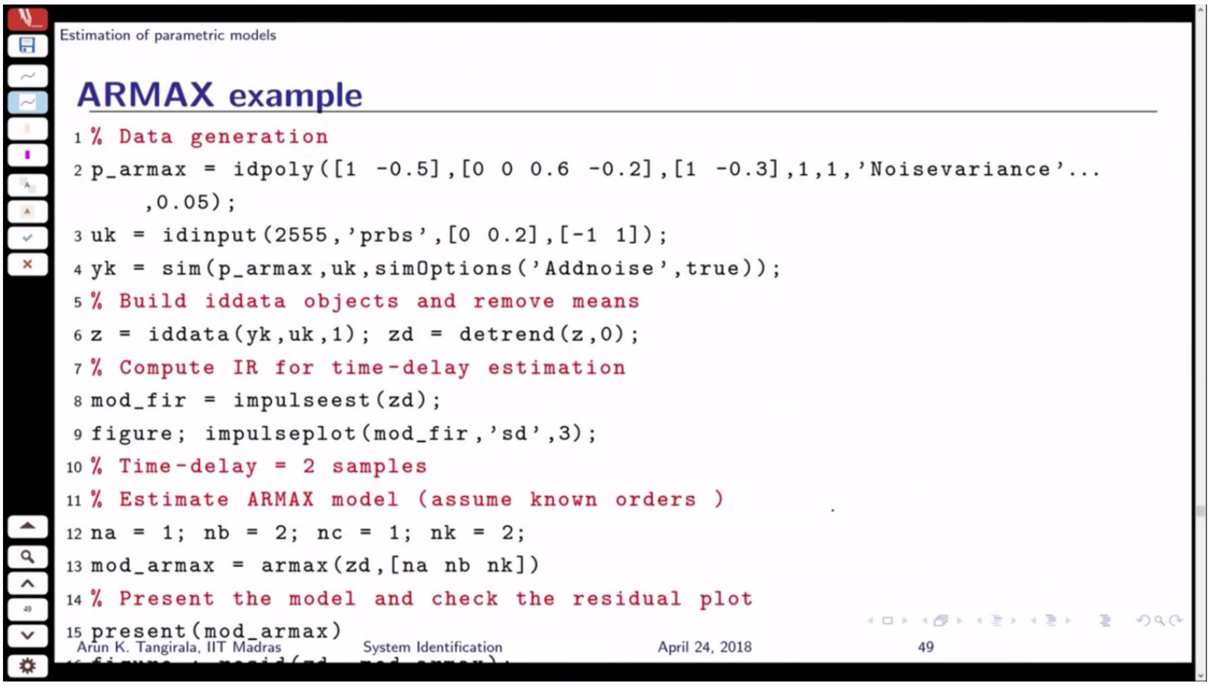
So what you can do is for non-linear least squares you get a decent estimate. Of all it says is that to kickstart the non-linear least squares optimizer, I have to somehow beg or borrow no stealing business. So beg or borrow C or estimate guess C from somewhere. That is what the user does. In the SysID tool box it is generated internally. This is generated internally. The initial guess. Use that to construct your regressor or the predictor gradient. And then using the estimates of theta you get you'll continue to refine the values ofC. So that is, you can use a PLR to get any decent estimate of C if you want.

(Refer Slide Time: 29:41)



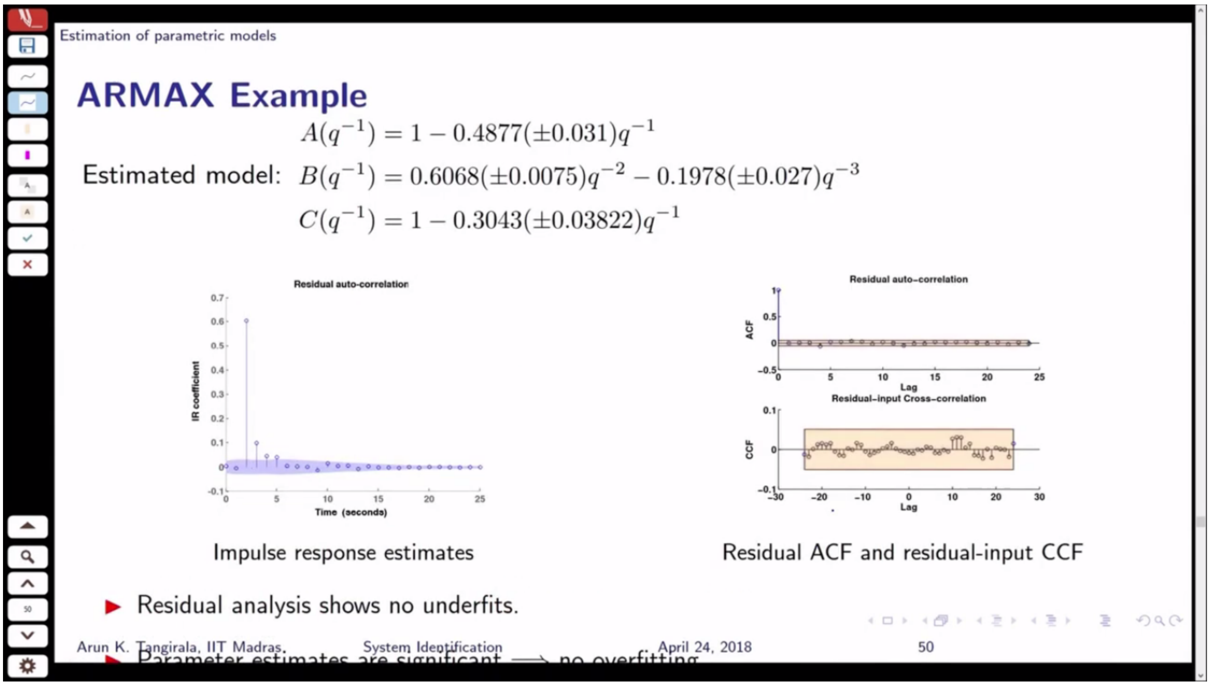
So there are many different ways, right. So this is just an example MATLAB script for you to estimate ARMAX by now your experts at that.

(Refer Slide Time: 29:49)



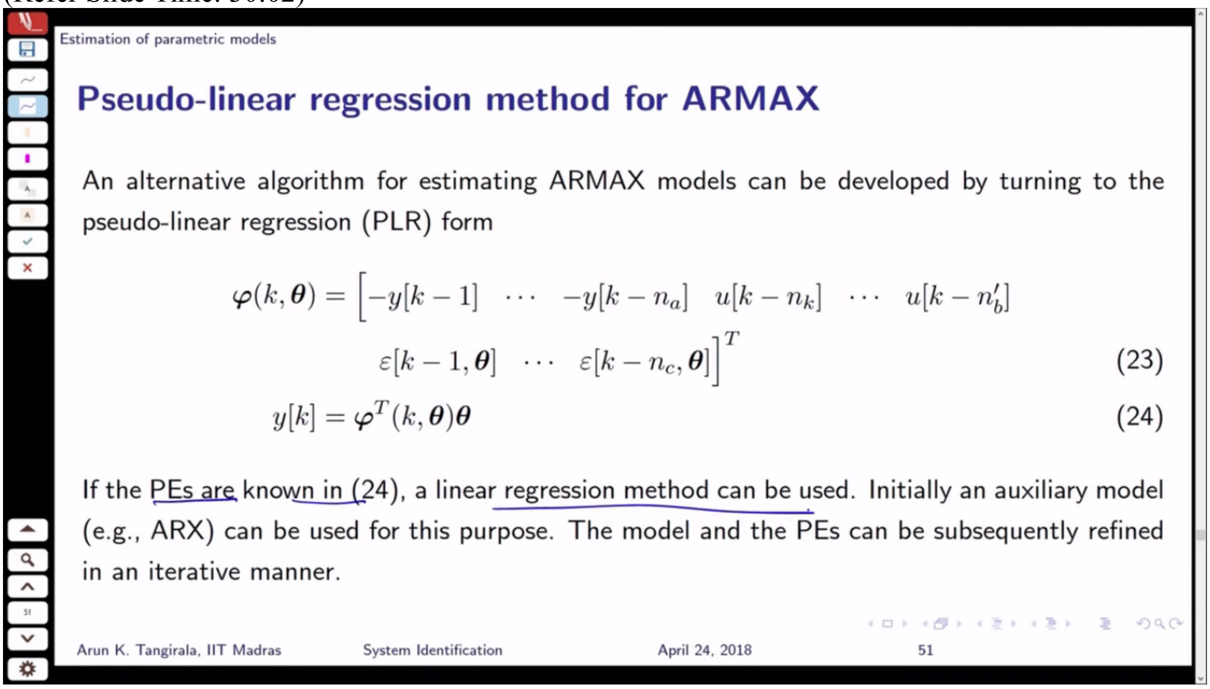
I will not go over these examples is just for your own digression.

(Refer Slide Time: 29:52)



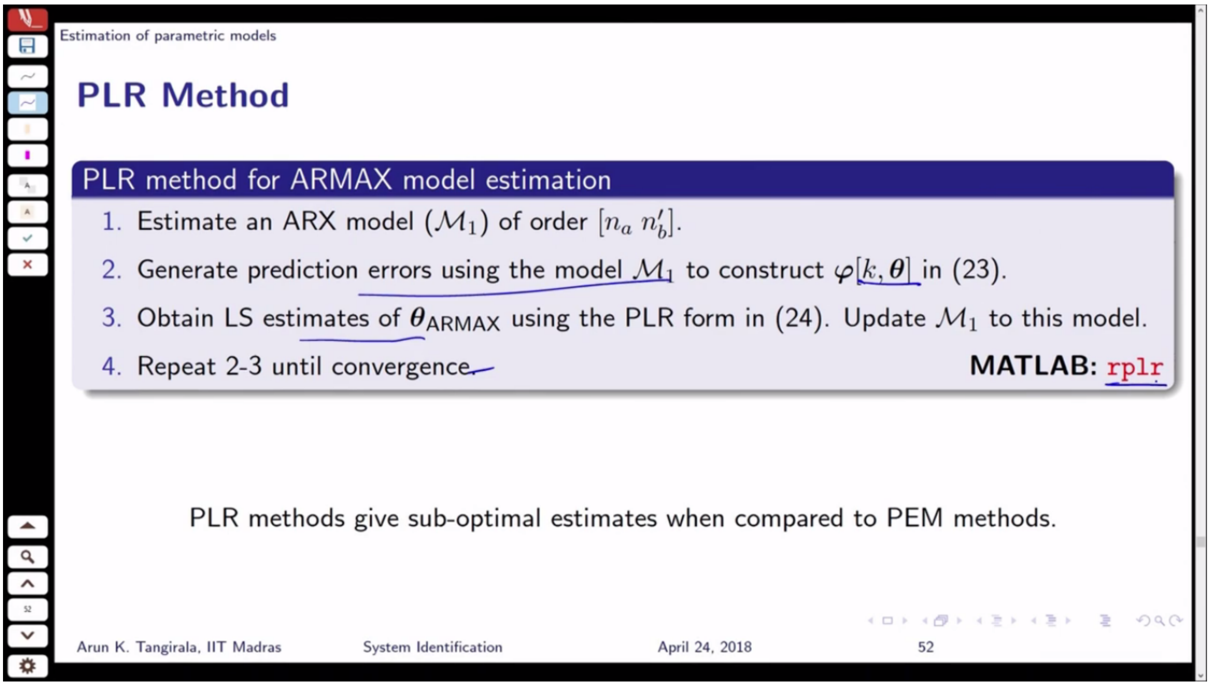
So this is a pseudo-linear regression method for ARMAX, I've already talked about it. If the prediction errors are known then a linear regression method can be used.That's idea in pseudo linear.

(Refer Slide Time: 30:02)



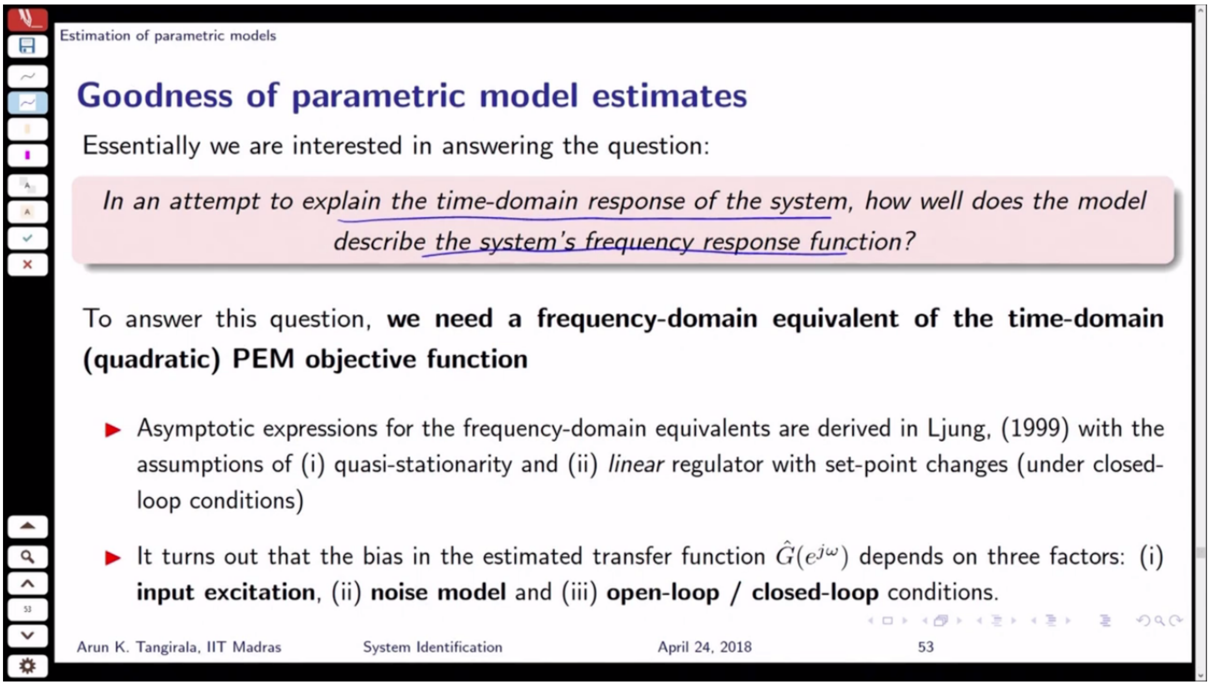
So there's a procedure for PLR estimate an ARX model of sufficiently high order generate prediction errors using the Model M to construct your regressor vector. Obtain least square estimates, repeat 2 to 3 until convergence. There is something called an rplr in MATLAB, which it does actually does recursively. But here you can say. So what is the difference between PLR and rplr? rplr does it, one observation at a time or a few observations at a time. So this are recursive versions of the algorithm. So it says I'll implement my PLR on 10 and then add one more observation 11, 12 and so on. And it basically it's a nice thing it'll show you after how many estimates the PLR. After how many observations the PLR method converges. So you can look up the rplr routine. And you should expect, in general the PLR method to give you suboptimal methods but they're decent enough.

(Refer Slide Time: 31:02)



Okay.So in the next lecture when we meet, we want to ask this question. So you've learned how to handle ARMAX,we'll also talk about OE model estimation. And then we will talk of the goodness of parametric model estimates finally.What do you mean my goodness already we're talking about consistency why do you want to bombard may be this question again and again?You can ask but the question this time is different. You are saying, we are fitting models in time domain, but how well does it describe the systems frequency response when it converges agreed. But we know that in some cases there can be bias. What is that bias being shaped by?We already now input plays a role.

(Refer Slide Time: 31:47)



Can we now obtain quantified expressions of those biases? We have already seen in the example, if I had used acoloured input, I would have obtained bias estimates fully. But even with the white-noise input some other parameter estimates are biased. So we want to ask what is that bias being shaped by can I control that bias and so on.And such questions can be answered by first answering this question. So when we meet in the next lecture, we will talk about this and the IV methods and we'll also talk about the Steiglitz-McBride method briefly which I've already told you in one of the lectures.That will conclude the estimation of parametric models and then we'll have only two wonderful topics left,which is essentially the state-space modelling and input design, very briefly input design. Okay. Thank you very much.