CH5230: System Identification

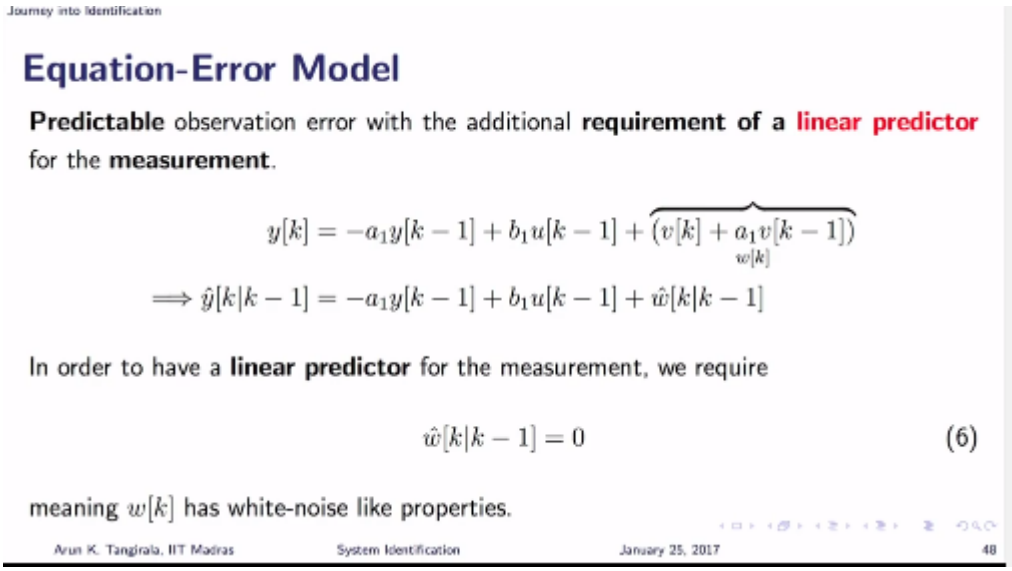
Journey into Identification

(Case Studies) 15

The other one leads us to what is known as an equation error model. The nomenclature the reasons behind this nomenclature we'll explain later on. For now just keep this in mind that we have already

looked at the output error model. Now we're looking at the equation error model. What is the second approach? In the second approach, we assume that the noise is predictable, right. But also require that additionally the predictor is a linear function of the parameters. So what do we do? The first step is we right now the expression for y.

(Refer Slide Time: 00:54)



Remember, we have expression for X, right. We have expression for x. I'm gonna replace X with Y minus V. That's all I'm going to do. And that's all I have done. I'm not changing anything here. The equation for G that we have assumed remains the same in both approaches. All I'm doing is I' m now rewriting this equation for G which describes an ice free response in terms of y and b. I'm going to replace XK with Y minus VK, YK minus VK, and XK minus 1likewise. When I do that, I get this equation on the top that you see. Okay. And from this equation I get the prediction y hat of k given K minus 1. The first term on the right hand side would be simply minus a1 yk minus 1. The second one remains b1 uk minus 1. The third term, let us say, we club this vk plus a1 vk minus 1, club this together and just introduce a new variable called wk. Whatever its prediction would be that of y hat. That is apart from these two terms.

So we have now y hat of k given k minus 1 as a sum of these three terms. Now notice that the first two terms are linear functions of A1 and B1, right. The first two terms in the prediction equation that I have for y are linear, they're linear in a1 and b1. The third one which is w hat of K given came in this one is actually a complicated function of a1 and b1, we'll not go into the details but it is, all right? It is some function of a1 and some unknowns. We would like that to be zero so that I have y hat to be a linear function of A1, B1 that's what we are requiring now. So the difference between the previous approach and this approach is in the output error approach we said, we hat of k given k minus 1 should be zero but now we are saying w hat of k given k minus 1 should 0. That's the difference. What is V? V is actually directly entering affecting X, whereas w affects the equation, enters the equation and we are saying that wk now let it be 1, unpredictable. So alphabetically of course, v and w are next

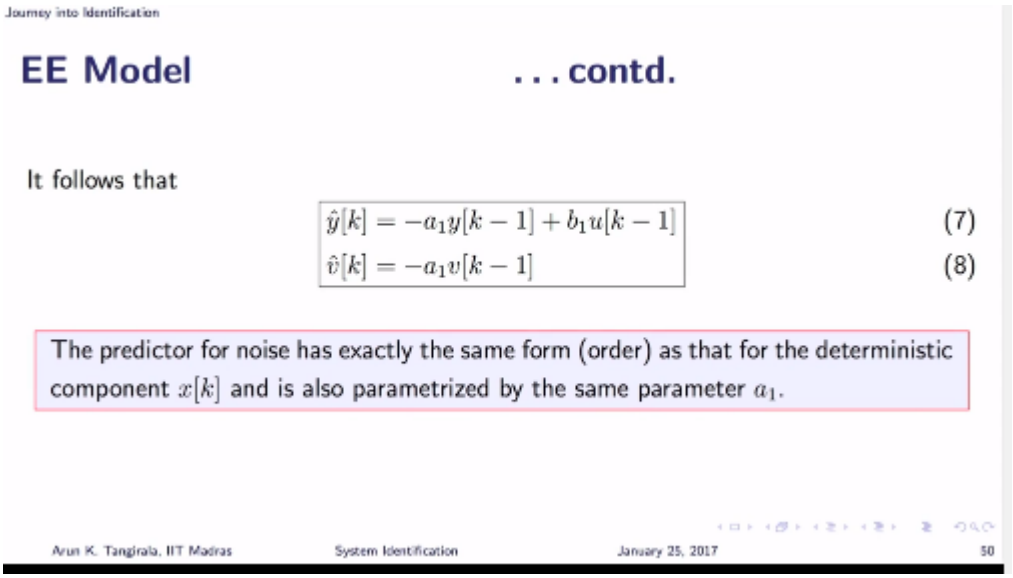
to each other but that's just a coincidence. But you should observe the prime difference in the output that model I'm assuming that the noise that corrupts the noise the noise free park is unpredictable whereas here, the noise that corrupts the equation is unpredictable and therefore the name equation error model. What is the consequence of assuming wk to be unpredictable. That means w hat of K given K minus 1 is 0, that is the consequence. Which means, we hat of K plus a1 vk minus 1 is 0. Why? Because wk is vk plus a 1, vk minus 1. Therefore w hat of K given K minus 1 is v hat of K given K minus 1 plus a 1 VK minus 1.

Why don't I use a hat on V of k, k minus 1? Why I'm I not using a hat there? It already given. K minus 1, it's given to me. There is nothing to estimate that. That's why we do not have a hat there that you should observe. What this means is, therefore we have now two models one, that is now we have a model for the noise also. Just look at a question 8, what we are seeing is now VK evolves in a particular manner such that its prediction is minus a1 vk minus1. And why the measurement evolves according to this linear equation. You should notice a prime difference between 7 and this model. Here we have assumed this model for noise free competent but 7 is a model for the measurement. And we required that. Why did we want? By the way, why did we want the linear predictor? Because when we work out the least squares optimization problem we want a unique solution that is the requirement. So that is the prime reason that I'm showing you this approach. The other thing that you should notice is the parameter that appears in the deterministic model is also the same parameter that appears in the noise model as well. There is no separate parameter for the noise evolution, right. That means, I don't have to fit a separate noise model if I fit g the noise model is also fit simultaneously. In system identification language, we call this as joint parameterization. Remember, I have a g and I have an h. g is an plant model relating u to X and h is the model that tells me how V is coming out, right. So v is assumed to be driven by some white noise. This tells me the governing equation for g. In the earlier approach, output error approach, what was h? 1. We assume straight VK is ek. But now h is not 1, you can choose a different governing equation for h but in this particular route that we have taken, I don't have to come up with a separate model for h. It is tied to g. Right. Remember, in SIS ID always that you have a left and a right hand that you can use freely to predict your process. g and h. You can leave your both hands free. You can get a lot of work done. Correct? In the output error approach we are fixing the right hand at one position. Imagine this is h and letting g do the job. Obviously, you should tell you that there is some limitation in this kind of model structures. In this approach what are we doing? Right hand is tied to the left hand. It has its own limitation, correct? Why I'm I saying, right hand is tied to the left hand? Because the parameters are the same. They share the same parameters.

Ideally you want two parameters g and h independently. Let g have its own governing equation, let h have its own governing equation. Then I will be able to model a larger class of processes. Correct? But we'll come to that later on. They are called the boxing strain constructors. Right now I just want to show two popular classes of models that people assume to begin within their system identification exercise. Gradually, as you are. Once you have understood how your left hand works by fixing the right hand. See, if I fix right hand to 1 and let the left hand go around. At least I have understood how my left hand works. Right, but in the second approach I'm tying it. So even my left hand can't go too far, if my hand right hand was here, my left hand can go really far but if it's like this it's going to be tough, right. So you should expect the second approach to have more limitations than the first one. And that is the case indeed. It's not a new thing that we discussed. It has been well articulated and well documented in this SIS ID literature.

Nevertheless, why are we assuming, why are we following this approach. Still although this is tied is there a benefit to this giant privatization? Easy to-- correct. The parameter estimates can be obtained easily although we don't prove it here, we'll realized later on that the solution to the optimization problem is very easy to solve. Computationally very friendly. I can solve by hand, I can do a lot of theoretical analysis once I have a close from expression a lot of benefits are there but this benefit comes at a price. Remember that. And by the way, these models are also known as ox models, auto regressive exogenous models and just giving you normal you'll know later on.

(Refer Slide Time: 10:22)

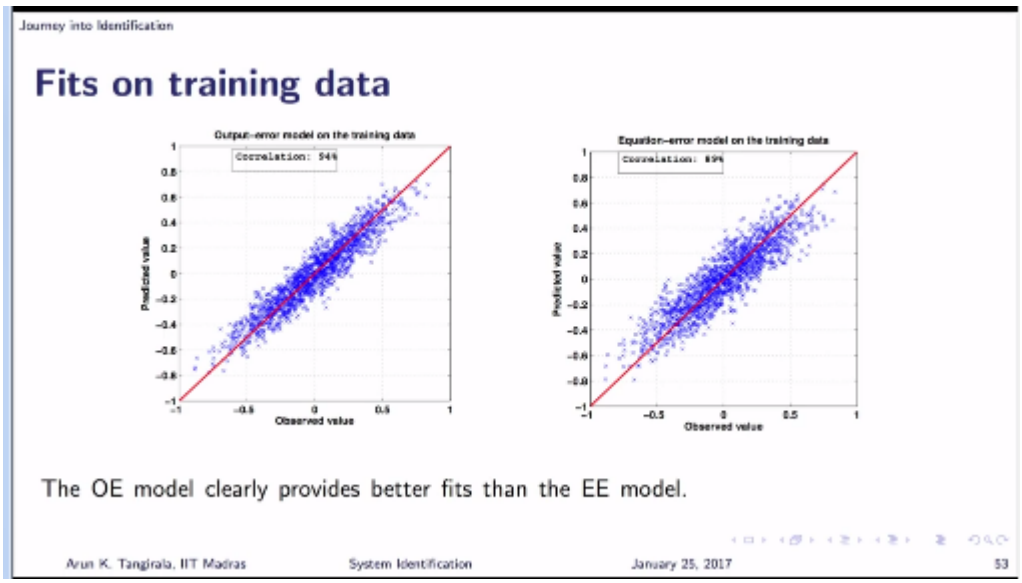


All right, so now that we have this expression for y hat, I straight away use the least squares analytical solution and now I have this estimates of a1 and b1. Earlier with the output error module we had another set of estimates. Again, the same question. Do these estimates make sense.? We had some checklist. Stability. Does it satisfy the stability criterion? Yes or no? Yes. Because a1 has to be less than 1 in magnitude that satisfied. What about the gain? What is a gain of this model? I mean as given by this model? So what do you expect now , this model to do well or not compare to the previous one? Which is closer to the gain that we estimated through non parametric analysis? The output error model gave me a better model that sense. It doesn't mean that that model is good in all respects. Remember that we are just having some checklists, right? When you go for a visa application there is a certain checklist. One after the other you have to satisfy all of them. But these are the main things. Stability. It's an important checklist. Gain, yes, it is important to check. Just by the simple analysis we can expect, we can develop an intuition that this equation error model may not have done a good job of fitting learning the data. Now how do we check? Still we should not look at the errors reported in the estimate brackets in the parenthesis. First thing to do is has it understood the homework problems that have been given. In all the courses that I teach whether it's process control or SIS ID or time series when students come back and say, I don't follow some of the concepts I say, and they pressed

the panic button before the exam I say, you go and first learn how to solve homework problems that means, you should learn those concepts at least then at least you're in a good position. So here also the same story I ask I present the model with the same data that I used for learning for training them. Which one has learned better?

The first one, the output error model. You would understand is plots on the y axis I have prediction, on the x axis I have the absurd value. We are only comparing with measurements remember, and the red line is a 45 degree line. Obviously there should be some scattered around it because of noise, correct. And also maybe because of non- linearities because there is some mild nonlinearity in the process.

(Refer Slide Time: 13:22)

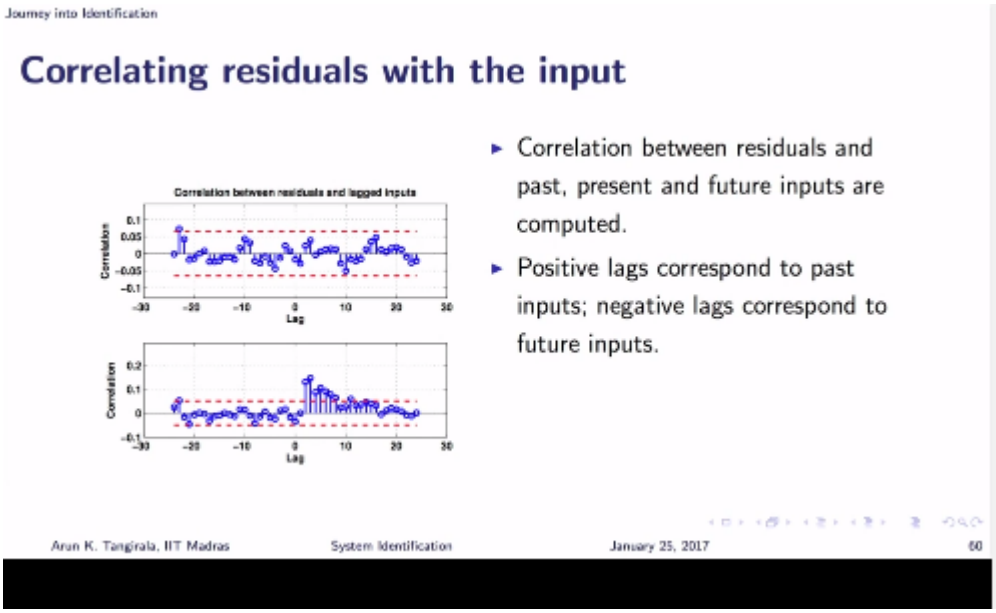


The arx or the equation error model gives rise to more scattered which means again intuition is confirmed that's the equation error model may not be doing a good job. Now what is up good check? This are all visualization of course, I can compute some metrics and so on. But what is a clean way of determining whether a model has fallen short of learning? This is only telling me relatively one model performs better than the other. But how do I say now that output error model has learned whatever is there to be learnt. That is the next question I'm going to ask. Right? How do we do that by means of what is known as residual analysis? And this is the key in Model Assessment and validation. You should not bypass this step under any conditions before you report the model. So what we do is we generate this residual. Essentially this is a leftover of what has been learnt. And then what do we demand? We demand that first that there should be nothing in the residuals that you can explain as a function of the input. Right.

Obviously in-- suppose the process was governed by a second ordered difference equation. And you fit of a fist order like we did just now. Obviously the residual will contain some dynamics left

unexplained dynamics. Those unexplained dynamics again are due to input, you don't have to keep asking whether output effects are remaining, it is sufficient to check for input effects. Therefore, the first test is always a correlation or a cross correlation between the residuals and the input. Once that is satisfied I can say, yeah, the model for g looks good. The next check is the model for h. What is an acid test for the goodness of h that the residuals there should be nothing left that you can predict as a function of its past which means whatever you have is not only now predictable by the input but also by its past that means, your residual now should have white noise like characteristics. Only when these two are satisfied you should turn to the errors in the parameter estimates. Because then you are guaranteed that under-fit has not occurred. Now the question is over-fit. Have I over parameterized my model? Have I fit the third order model to a first order process or the second order process. Remember, we talked about this one other symptoms of over fitting is large errors in parameter estimates. Right. Because you don't know what the truth is. Therefore the only way for us is to check for errors in parameter estimates. So these are the three things that you have to do before you jump to cross validation. That is presenting the student with an exam paper. Unfortunately our semester system which is like jam-packed it does not allow us to do all these three. If given a chance to me I would not let the student write the exam until I am convinced that the student is well trained. Am I right? That's what you also feel. Oh, my God, I don't have time to learn all of this, why one after the other like shatabdi exams are coming. And by the time you realize you graduated also. Okay. Okay. But whether it's by your will or God's will or some other person's will, there was a will written for you, you should graduate, you graduate. Anyway, so generally people tend to think that cross validation is a first step for model check. No. cross validation should come towards the end. Never think of taking the exam until you're sure you have taken enough steps to guarantee that you learnt well. Okay, so let's quickly look at the correlation plots. These are the two correlation plots that are corresponding to those two models.

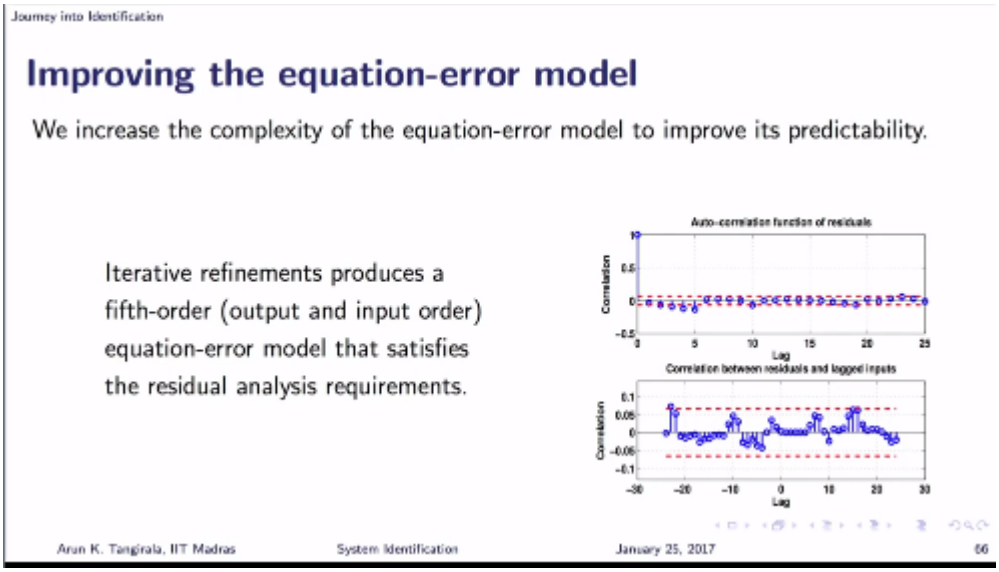
(Refer Slide Time: 17:24)



How do I read this plots? The top 1 is both are cross correlations between the residuals and inputs. Don't worry about the theory and so on. But learn to interpret this plots. In both plots x axis is what is

known as a lag and y axis is the value of the cross correlation. By definition cross correlations are bounded between 0 and 1 magnitude based or you can say, between minus 1 and 1. And what is required for us is that there should be no significant correlation between the residuals and the inputs. Again the same story statistical significance, those dash lines can be thought of as Lakshman rekhas. Okay? If you find any the cross correlation at any lag, what do we mean by lag here? What we are looking for is, remember, we have an epsilon K which is the residual. We are searching for correlation between Epsilon K and uk minus l. And that means any past, present, or future input if there is an effect left in the residual, that l is a lag l. What the top law tells me is that the residual from the output error model has no significant effects of any past, present. or future inputs. Of course, you should not expect future inputs effects to be there because we are only looking at Causal Models but sometimes if you have you know, inadvertently ignored the feedback. Whenever you have feedback, you can see future inputs appearing because inputs are calculated based on measurement. That's why we show all the negative. The negative lags on the plot correspond to future inputs. The positive lags correspond to past inputs and of course, the zero lag is a present one. You can see straightaway that there is no significant effect of input in the residual generated from the output error model whereas that's not the case for the equation error model that you see at the bottom. There is significant correlation. What we therefore conclude is that the model identified through the output error assumption is adequate. Which model is adequate? The g, we have not yet done a test for h, correct? The deterministic model has come out all right. But with the assumption that I made in the equation error structure which is that the noise is predictable and forcing the predictor to be linear did not work out well. Still at this moment I pretend I do not know the physics of the process but if we slightly deviate from that pretends and say, liquid level system dynamics, right? We know its first order. So did we make a difference in terms of assumptions on g in both this model structures or g has the same model? g has the same model but it is because of the difference in the noise structure. We have a result which states that if you assume that the noise is unpredictability you got right g. Right g in the sense, we know it's a first order dynamics whereas if you're know assume noise is predictable and you forced the predictor to be lenient. You don't have the right. What do you mean by right g is that the model is falling short of learning. We may have to increase the model order, in order to explain those effects but we know physically that's not right but suppose I do that. Let me show you. Come back to this plot. Suppose I do that, it turns out that I may have to fit the fifth order model. That's too much, right? Imagine fitting a fifth order difference equation model for a poor, little liquid level system, which follows first order dynamics.

(Refer Slide Time: 21:26)

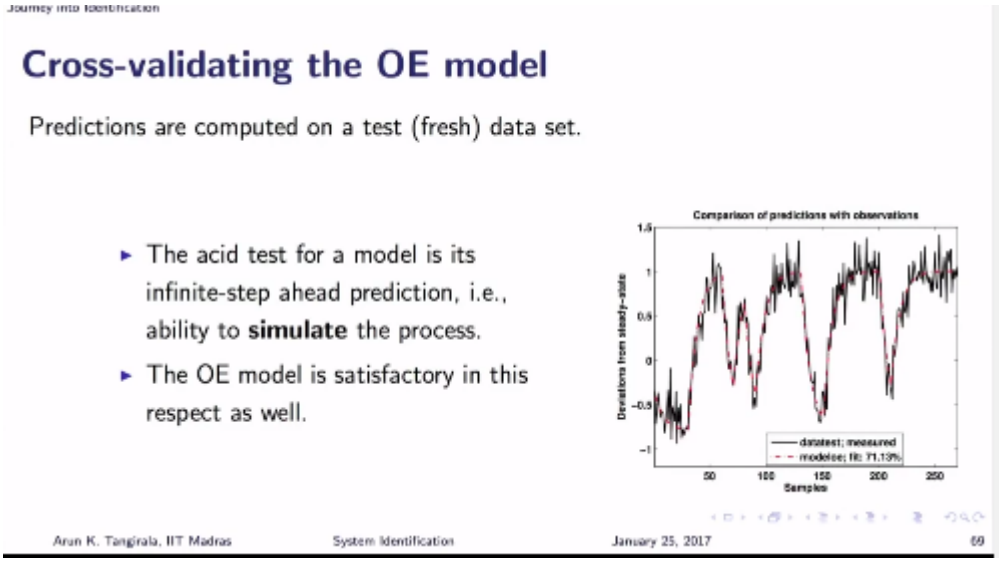


Suppose I didn't know the physics, right. This argument is only based on physics. How do I convince that the fifth order is good. Well, I'm showing you the correlation plots that the bottom plot is a cross correlation between the residual and input. Yes, the residual from this fifth order equation error model is insignificant with respect to the input, doesn't have any significant correlation. That top plot is the what is known as auto correlation plot. That is how much the residual is dependent on itself. In a sense the predictability of the residual that is also missing. Okay. Which means both g and h are satisfactory for this model but it's a fifth order model that you're fitting. Okay, maybe the processes fifth order. However, when I look at the parameter estimates of the fifth order model, you can straightaway see that there are a few parameter estimates that have large errors in them. Right? Can you spot at least one? a1, a2, right. So at least, one parameter you can spot that has been estimated with large error. Which means this model has run into this over fitting domain and it's not acceptable. I see on many websites unfortunately, which claim to offer very good tutorials on time series model and they fit fantastic models and you look at the errors in the parameter estimates they will be as large as any other country in the world. So it has big because that is a continent. So they just conveniently ignore say, look we have such a fantastic model. Don't buy into such models. Okay, so what this leads us to the conclusion is that I cannot fit an equation error model that makes sense. That makes sense both from a system identification viewpoint as well as from a physical viewpoint. What about the output error model? We tested the goodness of g. What about h? And that is what you're seeing here.

This is the auto correlation test. Remember there are two tests that we said, one is for g which it past. The other is for h, that is predictability of the residuals within itself, which is what we call as auto correlation. Once again, the interpretation is the same on the y axis you have auto correlation, x axis you have-- so auto correlation is simply correlation of epsilon with itself. And you will learn later on that auto correlation is a symmetric function. Therefore it is sufficient to show for positive legs, whereas cross correlation isn't. And once again the correlation values at all lags are statistically insignificant. Therefore we conclude that there is nothing left in the residuals to be predicted. That

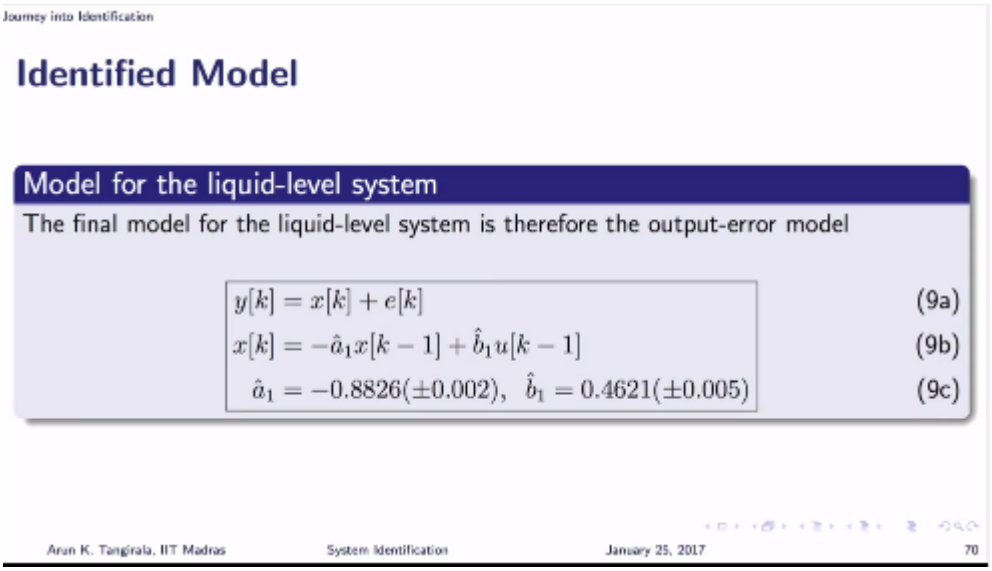
means our assumption that h equals 1 is correct. Correct in the sense, not truth but adequate. It does work. So the conclusion is that this output error model is the winner.

(Refer Slide Time: 24:30)



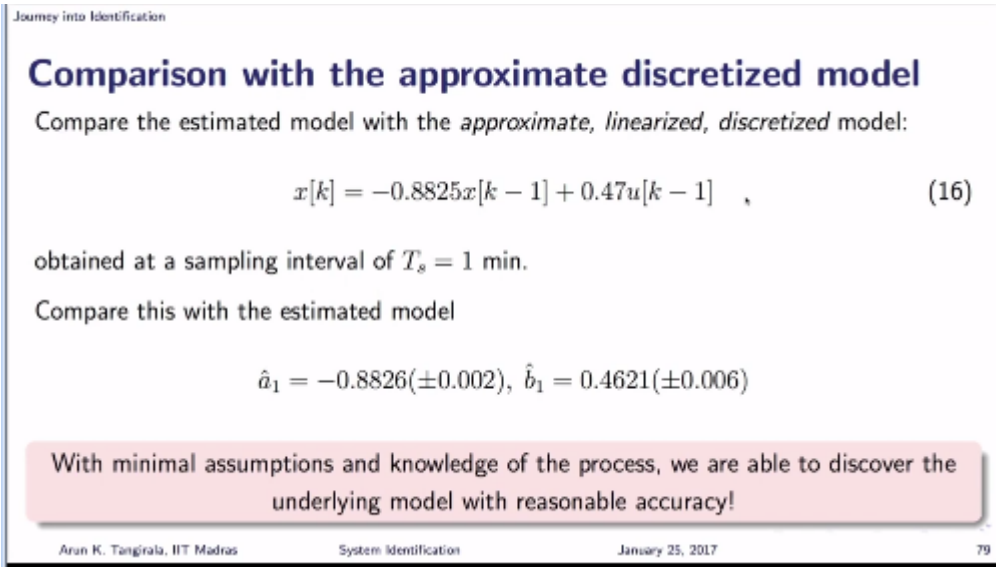
If I had a trumpet like sound I would have blown that. But I don't have it. But the output error model is the winner. And what do we do? The last step is cross validation. And I'm doing the cross validation here of the output error model on a fresh dataset. Remember we set aside some data set for testing which is what I'm showing. And it does a reasonable job don't expect it to follow those fluctuations there. There is nothing in the model that can explain those noisy fluctuations because h is 1. So we conclude that a reasonable model for the liquid level system working model is this. The all the equations that are given along with estimates and now we look at the error also.

(Refer Slide Time: 25:15)



We should have done that before cross validating, sorry. But anyway, we already know that extremely small and therefore I'm fine. I'll just conclude with one last thing. Suppose, I had linearized. I will talk about this transfer function later on. Suppose I had linearized. Remember, our ode is a non-linear one continuous time non-linear ode. What I am fighting is a discrete time linearized difference equation. For this example, I can compare with the truth at least reasonably. What do I do? I take the nonlinear ode, the linearize it, using Taylor series expansion, I linearize it. And then what I do is none as discretization which you will also learn. Discretization is writing a difference equation equivalent. When I do that I get this equation on the top. Okay.

(Refer Slide Time: 26:06)



This is what I get after linearizing and discretizing the non-linear continuous time ode. And this is the model that I've identified. Do you find them very close? So how did we come up with this model? We didn't use any physics. By and large, we just use system identification principles. So this is again to convince you that yes, there is merit a lot of merit to following a systematic procedure. And you can get very good models through empirical approaches provided of course, your data permits you to do so. But more importantly, equally importantly a systematic approach as follows. We'll close this case study in the first 15 minutes tomorrow and then we'll start off with the theory. Okay?