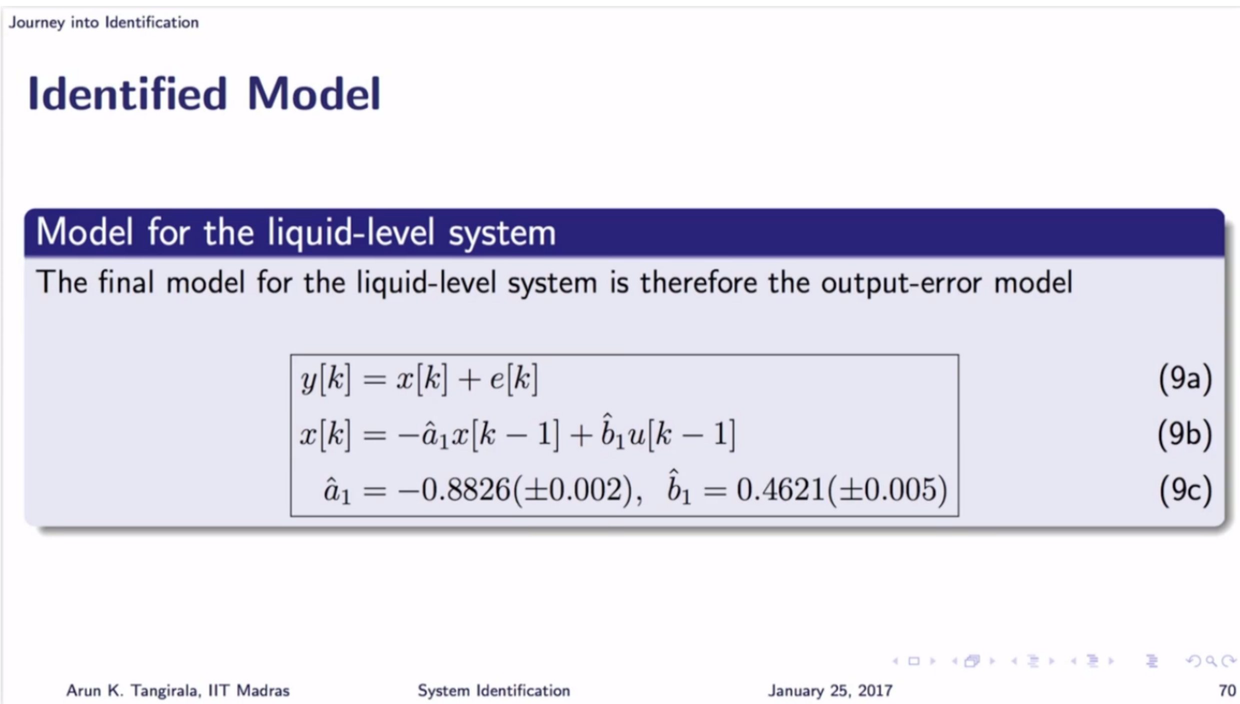
CH5230: System Identification

Journey into Identification

(Case Studies) 16

Okay. Very good morning. Sorry for the short delay. So what we'll do is spend the first few minutes in discussing summarizing what we have learned from the case study and then also briefly spend a few minutes on showing how a different kind of a model could have been built, that is a state space model. So if you recall, we have identified an output at a model for the liquid level case study and you've seen this model before, the slide before, where the deterministic part has been model as a first order. And as far as a stochastic part is concerned we have assumed, we have modeled it as white noise which means there is no predictability.

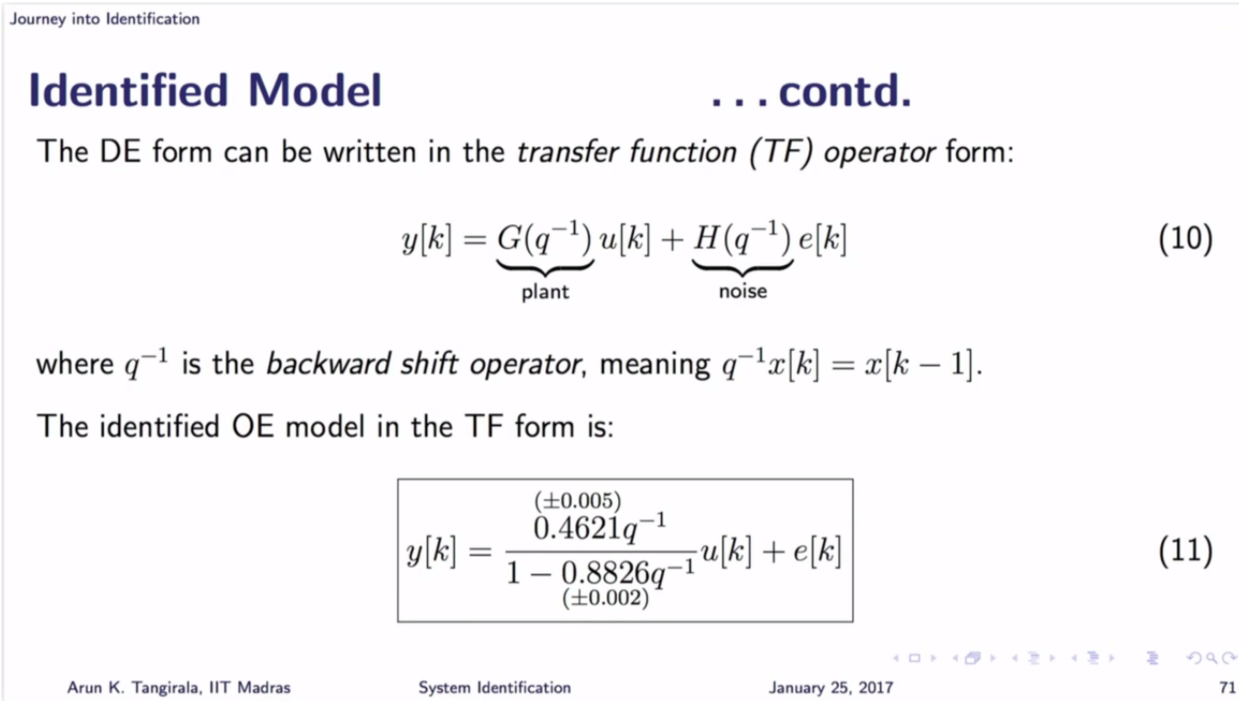
(Refer Slide Time: 01:00)



I have still not disclosed to you what kind of a process I have simulated. I've only told you that I have simulated a liquid level case study. But I have not told you what kind of stochastic component I have added into the simulation. I'll come to that very quickly.

Normally, it's conventional to represent these difference equation models by what are known as transfer function models. We will talk about the transfer function models in the lectures to come.

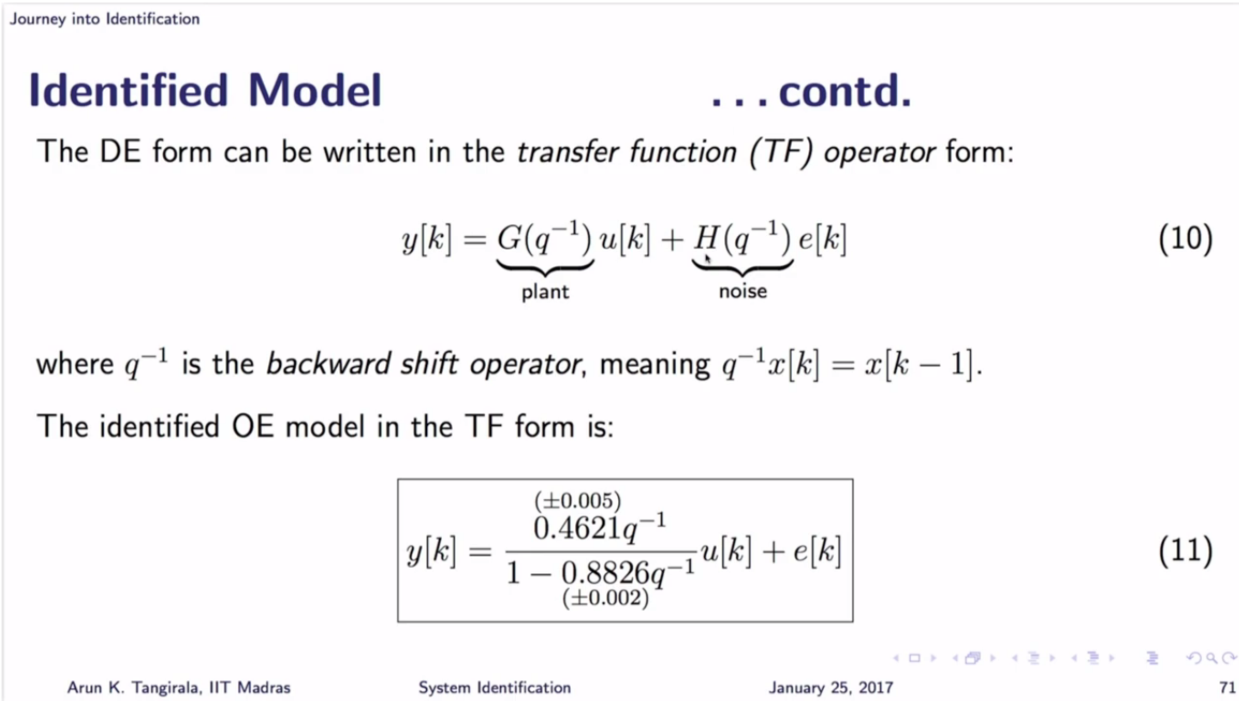
(Refer Slide Time: 01:32)



The advantage of representing a different equation form in transfer function form is only for mostly convenience but there are other implications. I mean you could do the same analysis to let me not tell you that it's only convenience, there is a lot more to it. One is the analysis of stability which we will learn later on. The moment I write a difference equation form in a transfer function from, one of the biggest advantageous is that I can do a stability analysis. We will talk about stability today. The other advantage is that by computing what are known as poles of that function. Some of you may be familiar with this terminology. If not of course, we will learn later on. By looking at the poles of the system, we can comment on what kind of response one can expect, whether you will see anoscillatory response or a damped response and so on. In a typical control course you learn this but for continuous time systems, it's the same theory applies to discrete time systems as well.

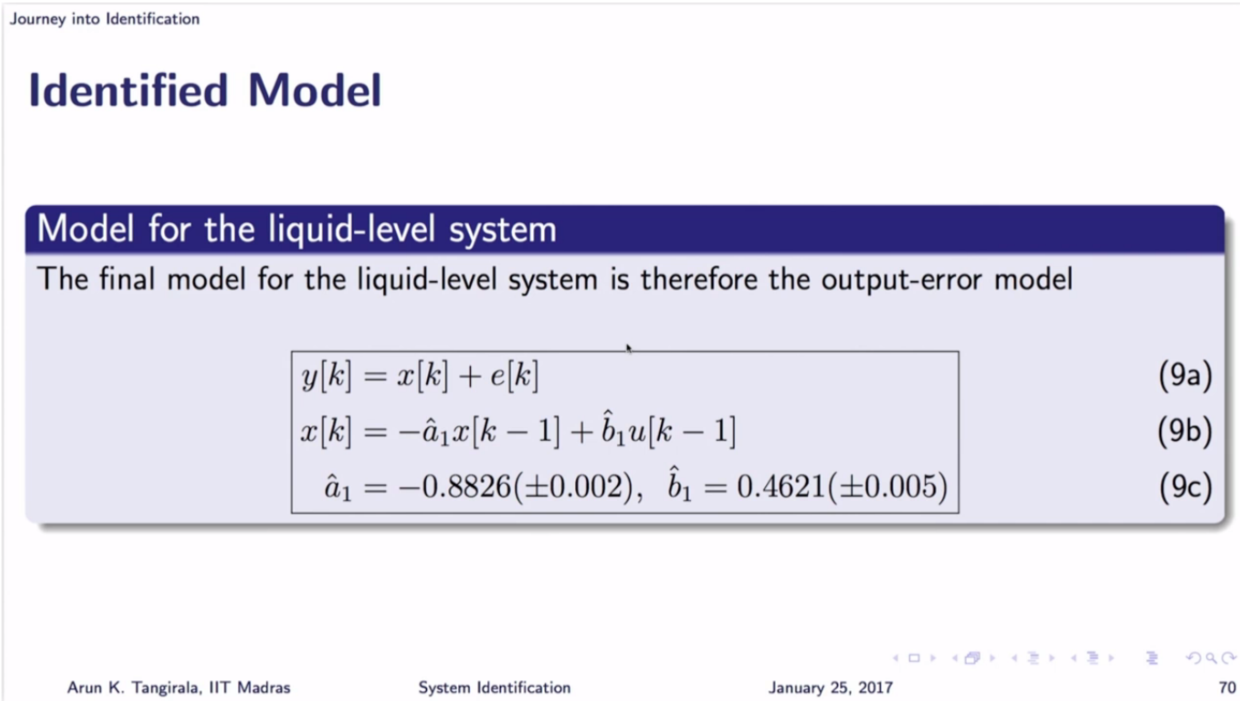
So what we have done here is we have taken the difference equation form that we had written for the liquid level process and we have called that and written it into the form of a transfer function and we denoted this by G.

(Refer Time Slide: 02:55)



We have already introduced that notation earlier. And the stochastic part has been modeled as a white noise being passed through H. Later on, we will come across a very important terminology. We have already used that term, known as filter. So you can also now imagine that the stochastic component has been modeled as white noise passing through a filter. At a later stage we'll understand, what is the consequence of assuming that under what conditions we can assume stochastic component being represented as white noise passing through a filter and so on. For now just remember that these are transfer functions. The filtering perspective will become clearer when we talk of frequency response functions and so on. So from the identified model it's clear that this G of q inverse which we often call as plant model is given here in the equation 11, as you can see. What all we have done is that, in this difference equation form we have introduced what is known as a shift operator. Again, I'll talk about that a bit later but it's a very straight forward operator.

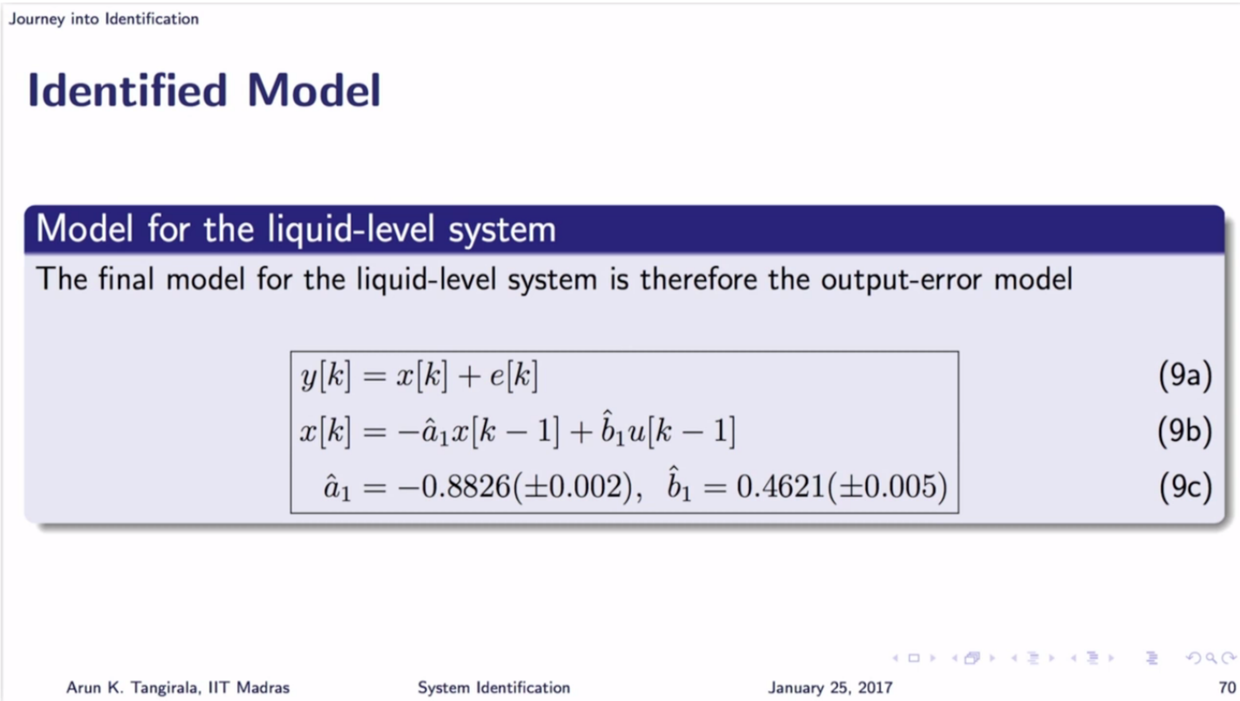
(Refer Time Slide: 04:16)



The role of a shift operator is to simply shift an observation to one observation in the past. Right. So it's q inverse is an operator. It's not a variable. You have to make it very clear in your minds. Whenever q inverse is used, it should be treated as an operator and it operates on observations or signals and so on.

So if you look at the difference equation here, in (9b), I can replace x of k minus 1 with q inverse x k and u at k minus 1 with q inverse u k. Right. And of course, there is also this forward shift operator, q inverse is called as a backwards shift operator.

(Refer Time Slide: 05:00)



You have the forward shift operator analogously define which shifts the observation to one observation in future. Why do we introduce a shift operator? Again, because then we can write, we can represent this difference equation in a compact manner to begin with. The q inverse is different from what you must have encountered in some of your courses as z inverse.You know, in some of the courses you must have learnt, some of you, where you deal with z inverse, and z inverse is a variable,whereas q inverse is an operator, z inverse is a complex variable. We will also talk about that at a later stage. Nevertheless, now coming back here, the difference equation that you see in (9b), can be now rewritten as what we call as a transfer function operator form to be more precise. How do you do that? It's very simple.

(Refer Slide Time: 06:01)

Identified Model ... contd.

The DE form can be written in the *transfer function (TF) operator* form:

$$y[k] = \underbrace{G(q^{-1})}_{\text{plant}} u[k] + \underbrace{H(q^{-1})}_{\text{noise}} e[k] \tag{10}$$

where q^{-1} is the *backward shift operator*, meaning $q^{-1}x[k] = x[k-1]$.

The identified OE model in the TF form is:

$$\boxed{y[k] = \frac{\underset{(\pm 0.005)}{0.4621}q^{-1}}{1 - \underset{(\pm 0.002)}{0.8826}q^{-1}} u[k] + e[k]} \tag{11}$$

Identified Model

Model for the liquid-level system

The final model for the liquid-level system is therefore the output-error model

$$y[k] = x[k] + e[k] \tag{9a}$$

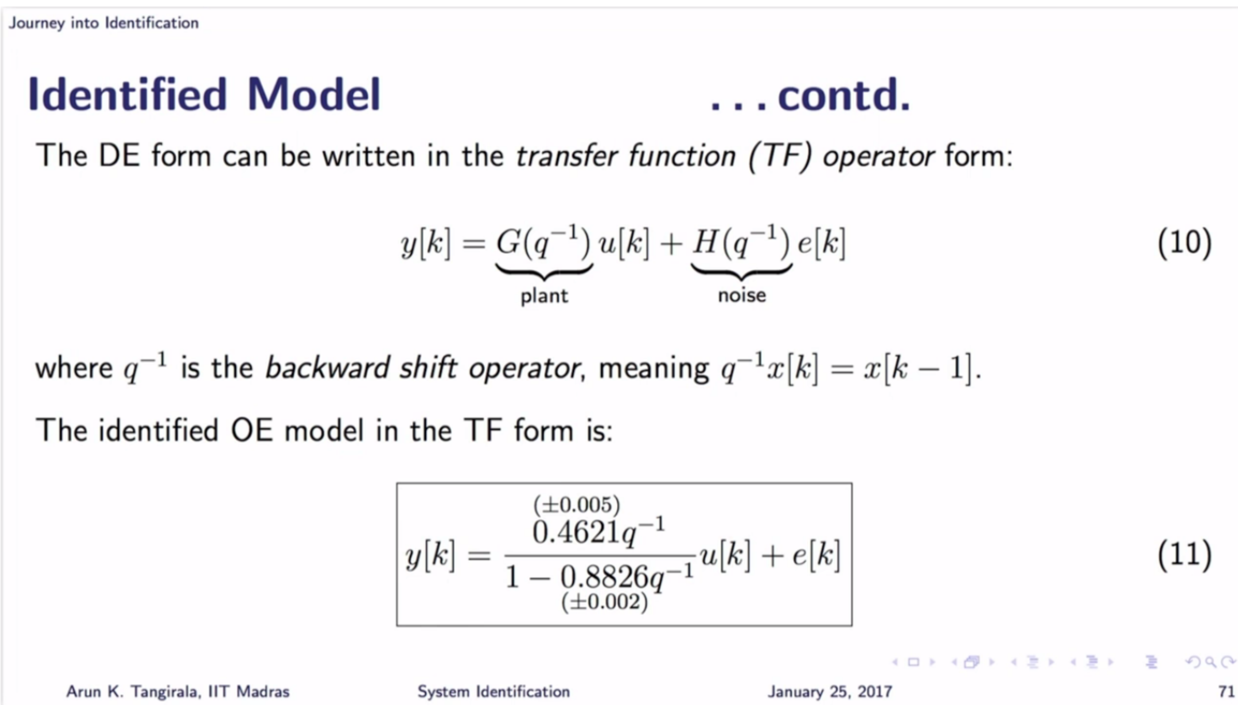
$$x[k] = -\hat{a}_1 x[k-1] + \hat{b}_1 u[k-1] \tag{9b}$$

$$\hat{a}_1 = -0.8826(\pm 0.002), \quad \hat{b}_1 = 0.4621(\pm 0.005) \tag{9c}$$

Go back and hear as I said, replace x of k minus 1 with q inverse x k. And u k minus 1 with q inverse u k. And then collect all the terms, like terms together so that you can write the difference equation form as 1 plus a1 q inverse x k equals b1 q inverse u k, from where the transfer function operator is arrived at. So now the transfer function operator here is a rational polynomial but polynomial in operators. Not polynomial in variables. Of course, there is a calculus associated with this operators, with this transfer function operators, and rational polynomials and so on. We don't have to go deep into it. For us, the basic understanding suffices. One of the things that you should remember is not to cancel out if you encounter at any stage in future, at any time in future, common factors between the numerator and denominator. You're not supposed to cancel them out because they are not variables. They are not polynomials and variables, there are polynomials in operators. They have a meaning. Okay. So now it must be clear as to how I have arrived at equation11 on the, that you see on the slide.

And of course, I've also indicated the errors in the parameter estimates there. Instead of a1 we have a1 hat and likewise for b1 as well. And this is a notation, this is a representation that one has to get used to as much as possible because you will see this almost in every other class.

(Refer Time Slide: 08:05)

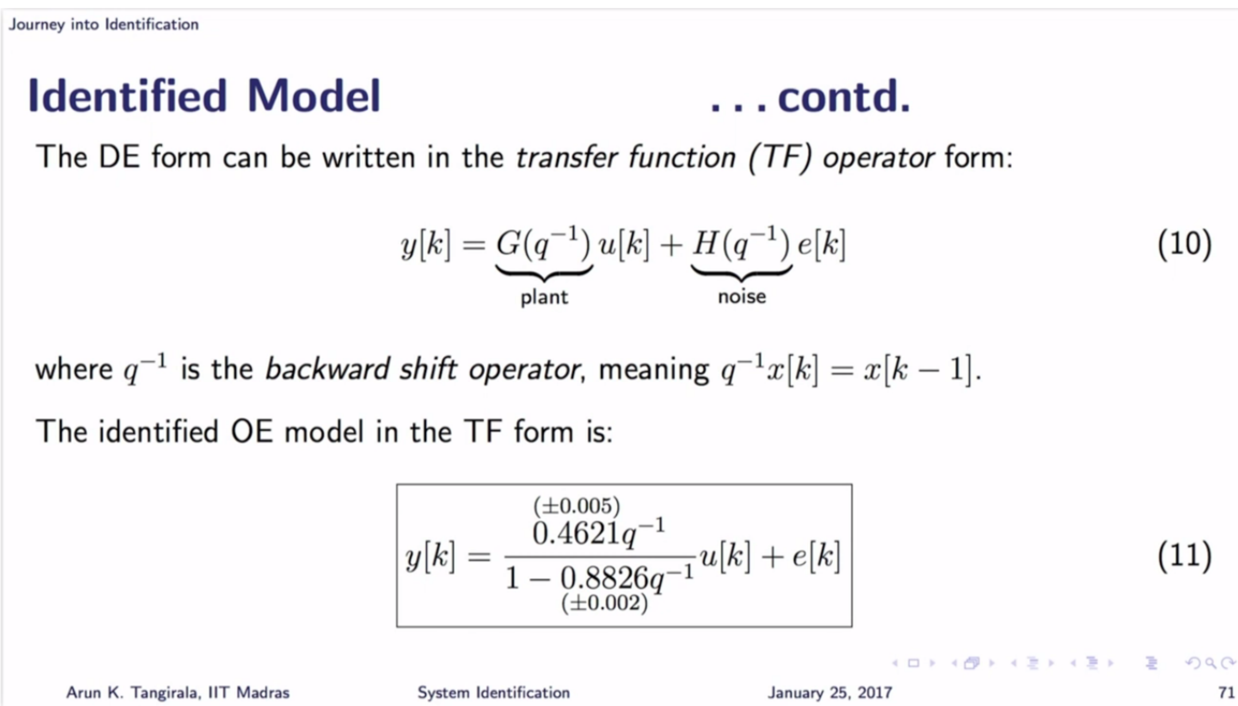


And the models that you would develop would be reported preferably in this fashion.

Now coming to the noise models, since we have assumed v k to be white, in other words, h to be 1, the h of q inverse simply is 1, that's it. For this liquid level case study, h of q inverse happens to be 1. Although, I don't say that here but it's pretty obvious from what we have modeled. We say, now by looking at the transfer function operator, this is not transfer function. You would call something as a transfer function, if you would write it as a function of variables. Since you are writing this as a function of operators, clearly we would call this as transfer function operators. Later on when we learn what a transfer function is, you will notice that the transfer function exactly, I mean, it looks identical, except that in place of q inverse you would see z inverse. But there are subtle differences that I'll point out later on, right. Okay.

So by looking at the transfer function also one can say that this is a first order system. And how does one do that by looking at the roots of the denominator and so on. If you are familiar with differential equations, you must have heard or you can even recall the terminology called characteristic equation. Right. And in characteristic-- in deriving a characteristic equation you replace the, essentially differential operator with some variable, with some, you know operator d.

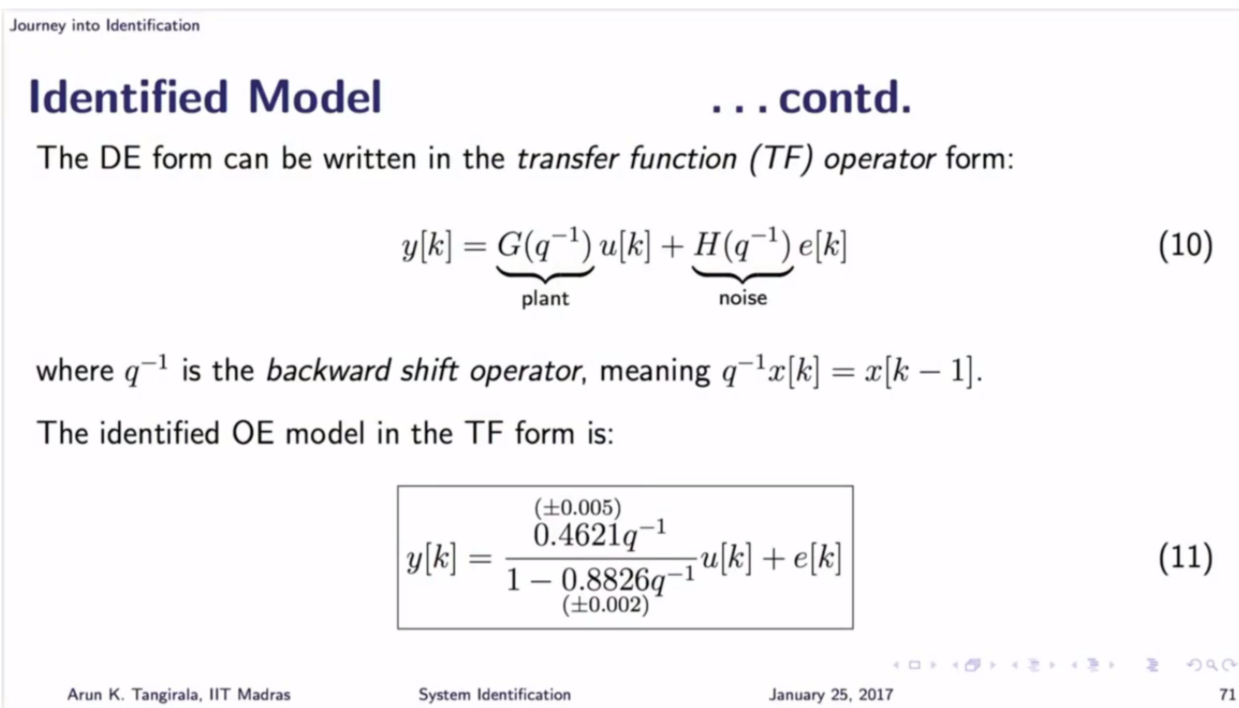
(Refer Time Slide: 09:46)



That would be the case for differential equations, for difference equations, we work with the shift operators. That's the prime difference. But otherwise, the rest of the concepts pretty much apply. There we talk of roots of characteristic equation, here also we can talk of roots of characteristicquestion. The order of the characteristic equation there would tell you what is the order of the ODE, here also the order of the difference equation is determined by looking at the characteristic equation. So pretty much all concepts apply. Of course, we will talk about this a bit more in detail later on. I just wanted you to get a preview of this transfer function and transfer function operator representation. Now the other thing that we talked about in the last class is that the model that we have identified is in very close agreement with the model that I would have derived theoretically beginning with the continuous time non-linear ODE, I would take the continuous time non-linear ODE that I can write for the liquid level system linearize it around the operating point. When I conducted the-- when I excited the system with the PRBS, I took the system to a steady state, If you remember, I had said that. So that becomes the operating point for your linearization. The linearized model would still be in continuous time. You'll have to further discretize it, so as to compare the model that you have identified because the model that we have identified is a discrete time model. You cannot directly compare with the continuous time model.

How to discretizeor linear continuous time model is something that we will again learn a bit later. So you can see that this example if you just to follow each thread coming out of this example this case study.

(Refer Slide Time: 11:36)



You would be led to different concepts and system identification and that's exactly what I have also talked about towards the end of this case study in my book. So if you have taken a look at the book, go to the end of chapter 2, and you would see that I have raised a number of questions. I am not going to talk of all of that but you should now be clear that if you follow one thread in this case study, you would be led to one concept and another thread would lead to another concept and so on. That's why this case study is very useful in many ways. Now there are two more questions that I want to ask you. And then discuss that before I move on to the state space model, then we'll get onto the theory. Why do you think the output error model worked well as compared to the equation error model? We know by now for this case study that the equation error model structure did not fetches a satisfactory deterministic model. Whereas the output error model assumption or the structure led us to a satisfactory deterministic model. Why do you think it happened that way? Any ideas? Sorry.

She is going to the root cause. Unfortunately that is need not be the root cause. I mean, I could have actually added some other kind of noise, not white noise, correlated noise and still the output error model structure as we shall learn again, later on theoretical you see, everything I'm saying, we can learn later but it's true for under open loop conditions regardless of the nature of the noise that corrupts your data, whether it is white or colored an output error model structure is guaranteed to give you to fetch to the best deterministic model provided you have, guess the order right and so on.

Right.Whereas that is not necessarily true of equation error models. We'll learn the theoretical reasons, why, on why this is true later on. But as of now, what do you think has caused this issue? I mean, I mean, has caused this kind of situation where the equation error model structure has failed for this case study.

No. But output-- so, no output error model also has the same. They're both in the linear framework, which non-linearity are you talking about?

That's different. So you have to be clear, when you say, non-linearity. You're talking of non-linearity of predictors. Okay. That's a slightly advanced thing that I would say and it has a lot more subtle details, we'll not get into that. The important thing to observe first is a similarity between these two. That is when I work with the output error and equation error models. I have assumed the same deterministic model. Right, in the sense, we used the same first order. The prime difference was in the noise, right. The assumptions that we made about noise. That is one aspect of it but the more important aspect of it is that-- you can, it should not give you the impression that you should always assume the noise to be white regardless of the situation and you will get good model. Yes, you will get a good deterministic model but it turn out that your noise dynamics are not modeled properly. The more important reason is in fact that we have tied together the parameterization of the plant and noise models. As I said, last class, in the last class symbolically I showed you that your g and h are like your left and right hand, and whichever suits you.

These are the two hands with which you want to get your work done. That means you want to explain the data. If you fix one of the hands to one position and not move. And let the other hand move freely. That is your output error model. Whereas, with the ARX's model that we fit. We are not only fixing the one of the models but we are tying it so that we restrict the movement of the other hand. So clearly, the kind of processes that you can model is going to be limited, particularly if you think the left hand has the g and right hand has h, then this the moment h comes and ties together with g, the kind of processes are the deterministic processes that you can model is going to be restricted. At least, in output error model, you've fixed your h and let g roam freely. So that I can actually figure out the correct g. I'm just giving you a very symbolic and a qualitative explanation here. The theoretical explanation will come at a later stage. So it is this joint parameterization that actually is the main culprit and we will show theoretically later on, we'll understand at least that indeed that it is a joint parameterization that is the main culprit. Okay. In fact this is a special case of joint parameterization where all the parameters in h are tied to the parameters in g. There is another case of joint parameterization, where h is partly tied, not the entire hand, a few fingers are tied, Okay. Something like that. Right. So it's not, the hand is not holding, the right hand is not tied to the left hand completely maybe, two fingers. So that there are additional degrees of freedom.That additional degrees of freedom can actually get you a better model. I mean probably take you closer to the truth. But it is, in this ARX's models specifically where it is not just jointly parameters, it's fully jointly parameterized. Partly jointly parameterized models can actually get you closer to the truth. Where we willencounter models such as R max models. So you have a range of possibilities, where you have output error model, where I do not bother to model the noise dynamics at all. And under open loop conditions, you are guaranteed that you will be able to recover the g, correctly. Then you have on the other extreme where h is parameterized there is a model that you're building but it shares all its parameters with g. Okay. Which means that h is completely tied to G.

Student: One thing, what if noise’s parameter is going to change?

Correct. Yeah. That is the thing that we don't discuss at this stage. I'm just showing you two extremes that have prevailed and that continue to prevail in the literature. There are intermediate ones and then there is one where g and h are independently parameterized. And I want to instill these concepts upfront. This notion of parameterization, so that when you're building a model you know clearly what are the degrees of freedom that you have that it can play an important role and so on. That is one point that is not sufficiently emphasized in many system identification courses. I mean, even if it is emphasize, its emphasize at a much later stage. I would like to bring that aspect upfront itself. So and it's a very good question that you asked. If I had an additional parameter in the noise model, which did not share the bench with g, then I would have had an additional degree of freedom and possibly that would have countered the joint parameterization that I have. Right. For example, in the ARX's model g and h, share the same parameters.

(Refer Slide Time: 19:44)

Journey into Identification

Identified Model ...contd.

The DE form can be written in the *transfer function (TF) operator* form:

$$y[k] = \underbrace{G(q^{-1})}_{\text{plant}} u[k] + \underbrace{H(q^{-1})}_{\text{noise}} e[k] \quad (10)$$

where q^{-1} is the *backward shift operator*, meaning $q^{-1}x[k] = x[k-1]$.

The identified OE model in the TF form is:

$$\boxed{y[k] = \frac{\underset{(\pm 0.005)}{0.4621}q^{-1}}{1 - \underset{(\pm 0.002)}{0.8826}q^{-1}} u[k] + e[k]} \quad (11)$$

Arun K. Tangirala, IIT Madras System Identification January 25, 2017 71

Of course, not all parameters of g going to h but all parameters of h are present in g. Right. If I have an additional parameter, let's say, maybe c1or something like that then that could have counted, it could have canceled the effect of the joint parameterizationand I would have probably ended up. For example, here we set for the output error model, we have assumed h of q inverse to be 1 but for the ARX model of first order that we fit h of q inverse, what would it be? If you recall the discussion that we had? Wow. 1 over 1 plus a1 q inverse, because we assumed the predictor to be this right. V hat of k to be minus a1, v k minus 1 or we said, v k is minus a1 v k minus 1 plus e k. Right, this is what we ended up assuming. Why, because we wanted to predictor to be linear in parameters. As a result h of q inverse is 1 over 1 plus a1 q inverse. So to answer your question, suppose, I had assumed, right. Suppose, I had assume h of q inverse to be 1 plus c1 q inverse over 1 plus a1 q inverse. Then the algorithm would have driven c1, such that it almost cancels out the effect of a1. Because it's estimates I say, almost. So that would have given you straightaway a good estimate of g. We didn't we didn't pursue this option. But you can think of this. Clear on, I mean, fit an R max model, specify the orders and see, if actually does well. This case study offers plenty of room for exploration. Okay.

So to summarize it is the way you parameterize the plant and noise models that can make a big difference. And why because your parameters are your knobs that you turn around and the way-- if you have fixed one knob you said, this knob is going, not going to be opened at all. I mean, not going to be touched. There's only one knob that you play around with. Fine. That is your output error model. But if you tie a thread for one of the knobs and say that whenever this rotates this way, the other knob is also going to rotate this way, so you don't have truly two degrees of freedom or that many degrees of freedom. The full degrees of freedom is available in what is known as a Box–Jenkins structure. Where g and h do not…