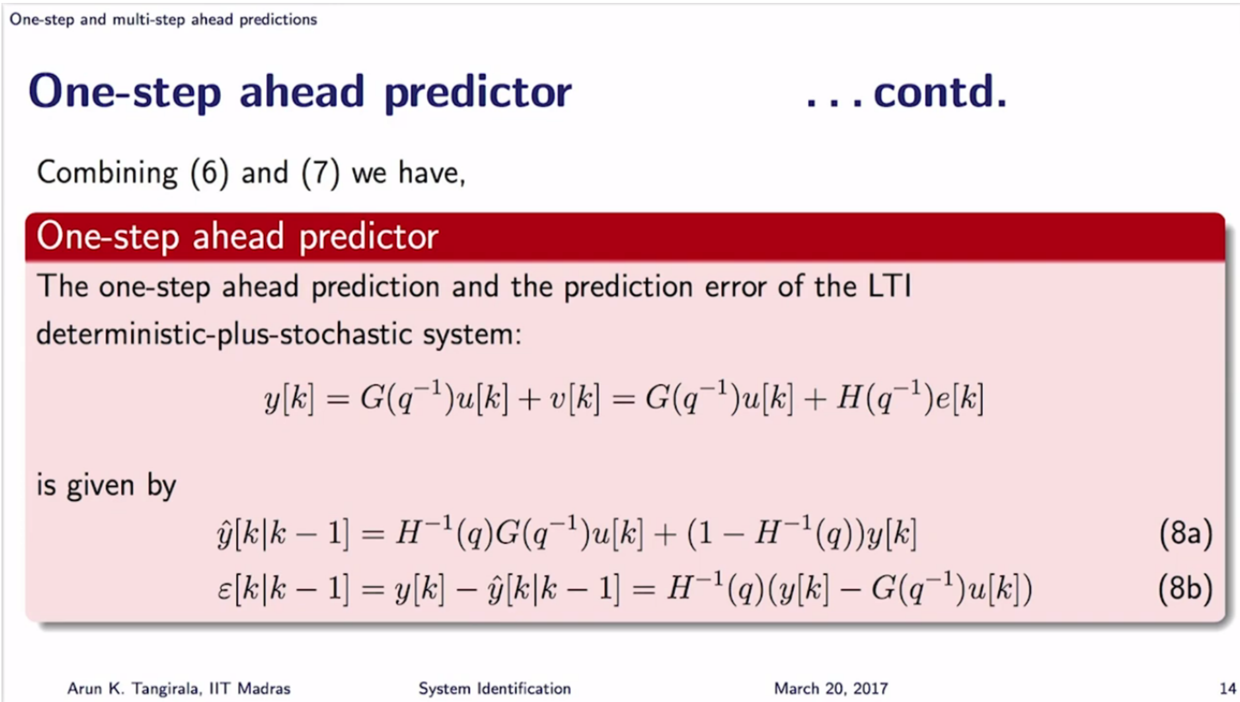
CH5230: System Identification

One step and multi-step ahead prediction

Part 3

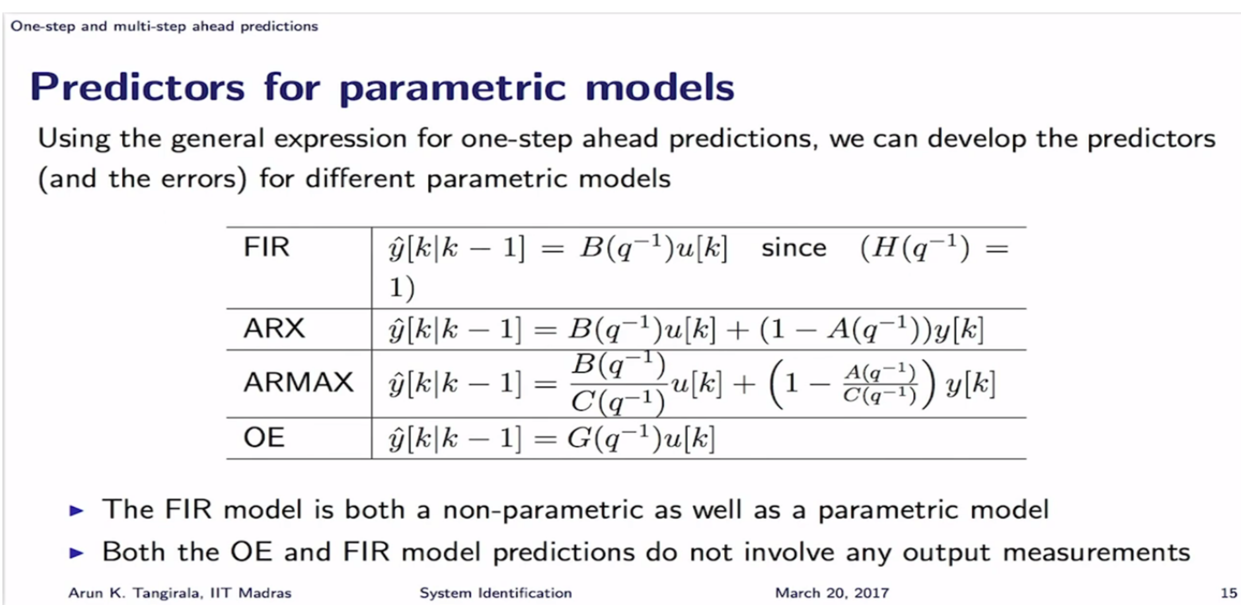
So, what we'll do is we'll continue our discussion on predictions. Spend about 10 minutes or so speaking of infinite step ahead predictions, multi-step ahead predictions.

(Refer Slide Time: 00:25)



And then, get started on estimation. So in the last class if you recall, we have spoken about one-step ahead predictions. And particularly we derived this expression here for the one-step ahead prediction, which allows us to write a predictor for a general LTI model, which has G and H.

(Refer Slide Time: 00:53)



So if you take this formula and apply to different model structures, then you get these predictors as you see and the simplest of the lot is FIR, then you have ARX. And then of course comes ARMAX and OE and so on. So what you find interesting about FIR and OE, as I pointed out last time, the predictions are purely derived from the input.Whereas with ARX, ARMAX and BJ, the predictions involve past measurements as well.Right? And if you don't see that clearly here, look at the difference equation form.

(Refer Slide Time: 1:31)

Difference equation forms

For implementation purposes, the difference equation forms are highly usefu:

- **FIR:** $\hat{y}[k|k-1] = b_1 u[k-1] + \cdots + b_{n_b} u[k-n_b]$
- **ARX:**
$$\hat{y}[k|k-1] = -a_1 y[k-1] + \cdots - a_{n_a} y[k-n_a] + b_1 u[k-1] + \cdots + b_{n_b} u[k-n_b]$$
- **ARMAX(1,1,1):**
$$\hat{y}[k|k-1] = -c_1 \hat{y}[k-1] + b_1 u[k-1] + b_1 c_1 u[k-2] + (c_1 - a_1) y[k-1]$$
- **OE:**
$$\hat{y}[k|k-1] = -a_1 \hat{y}[k-1] + \cdots - a_{n_a} \hat{y}[k-n_a] + b_1 u[k-1] + \cdots + b_{n_b} u[k-n_b]$$

And you should be well-versed with the difference equation form as well. So for then FIR model, for example, you see at the top, the prediction expression here uses only past inputs. Right? Whereas ARX and ARMAX involve past measurements. And OE, on the other hand, is a bit peculiar in the sense that it doesn't use past measurements, but what does it use? What is it using? Past predictions.Right?

(Refer Slide Time: 02:20)

Predictors for parametric models

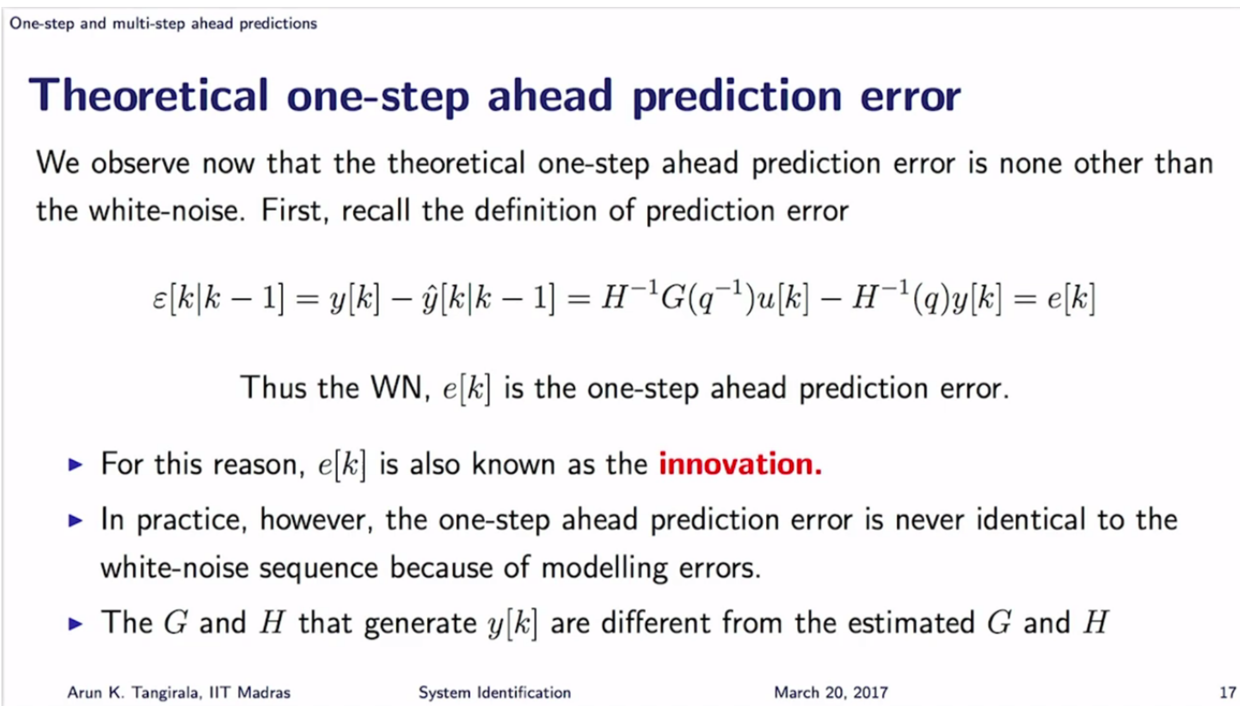
Using the general expression for one-step ahead predictions, we can develop the predictors (and the errors) for different parametric models

FIR	$\hat{y}[k|k-1] = B(q^{-1})u[k]$ since $(H(q^{-1}) = 1)$
ARX	$\hat{y}[k|k-1] = B(q^{-1})u[k] + (1 - A(q^{-1}))y[k]$
ARMAX	$\hat{y}[k|k-1] = \dfrac{B(q^{-1})}{C(q^{-1})}u[k] + \left(1 - \frac{A(q^{-1})}{C(q^{-1})}\right) y[k]$
OE	$\hat{y}[k|k-1] = G(q^{-1})u[k]$

- The FIR model is both a non-parametric as well as a parametric model
- Both the OE and FIR model predictions do not involve any output measurements

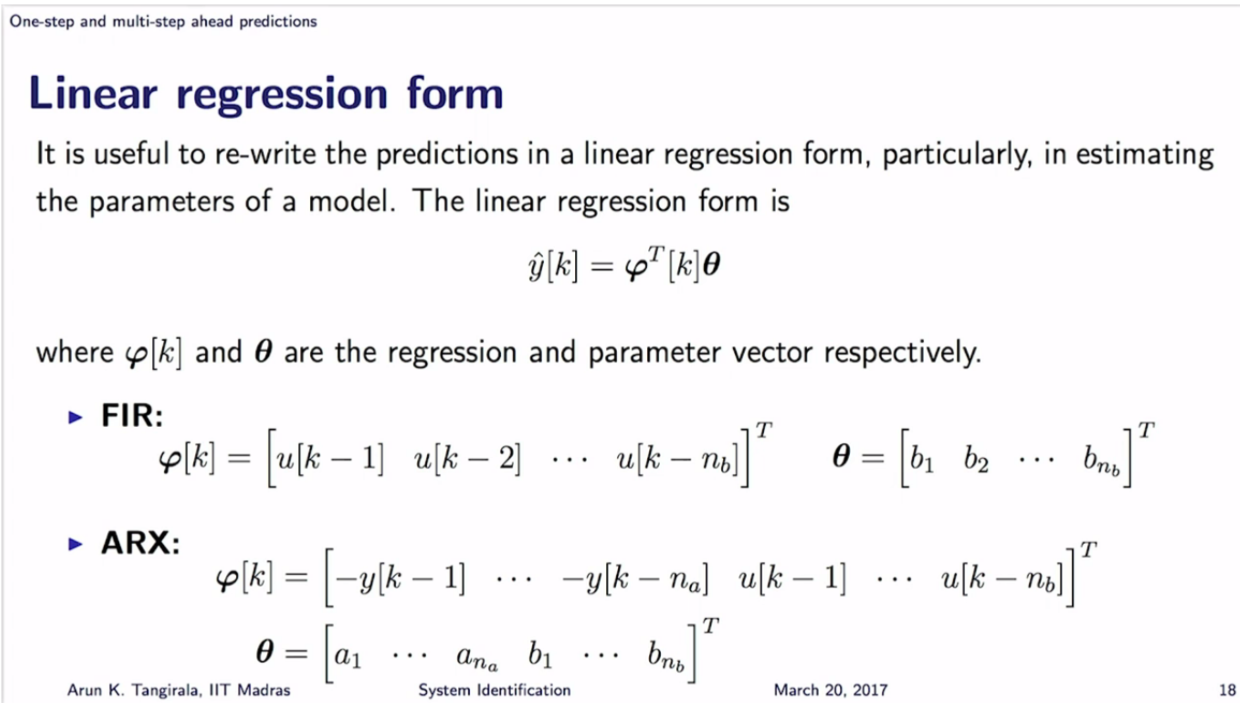
That’s because if you look at the predictor for OE, it says, the prediction is simply input pass through filter. And it’s a denominator of the filter that’s brining in the past predictions. So, the prediction is deriving out of itself. It doesn’t rely on the measurement at all. Okay. Whenever a prediction involves only inputs. So effectively if you look at OE and FIR, the predictions are being derived purely from inputs. There is no past measurement at all. Whenever that is the case, we call that as simulation. Right? We have been using the word simulation in a loose sense, but technically what simulation means is that you do not use the measurement from the plant at all. You simply use the inputs sitting disconnected from the plant and you’d say, I have this model here which will emulate the plant. And I feed the same input, that will also be fed to the plant. And then ask the model to predict what happens, but I will not use any response from the plant to make a prediction.We’ll again talk about that shortly. But what you notice is that OE model, although, the model has a denominated dynamics unlike the FIR, still doesn’t make use of the measurements at all. That’s the uniqueness or you can say peculiarity of an output error model.

(Refer Slide Time: 03:59)



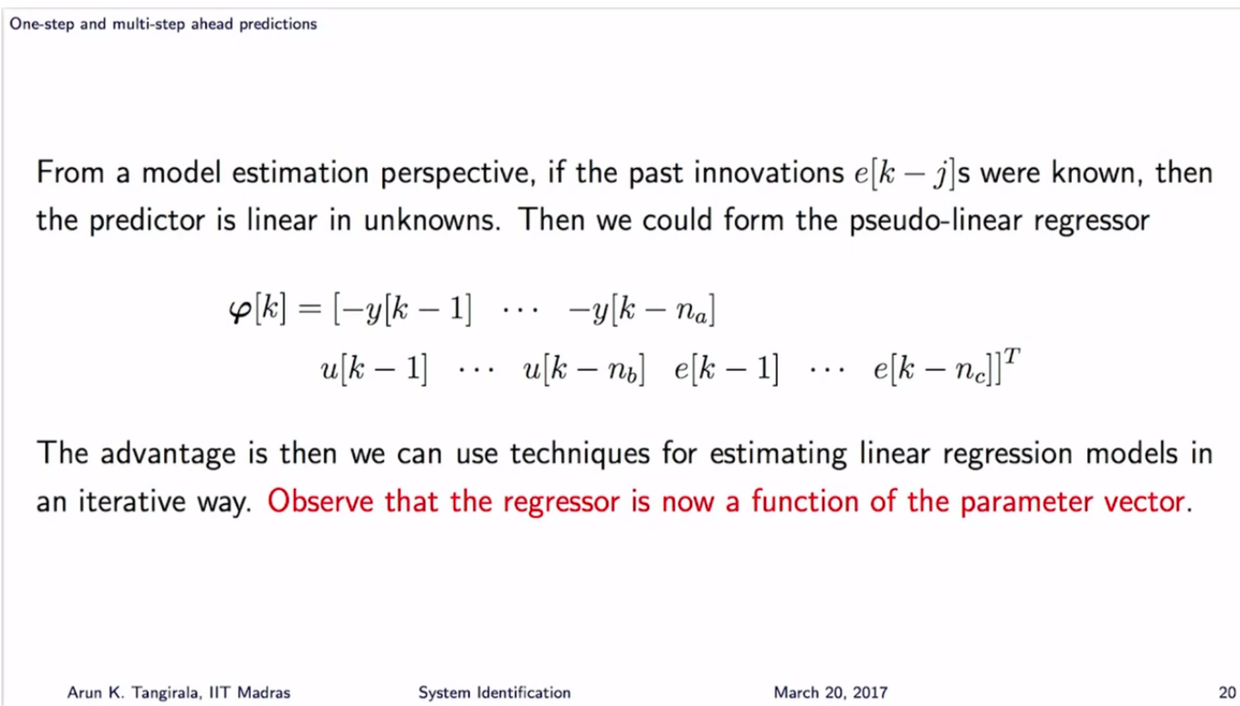
Okay, now let’s move on and again recall one important aspect, which is that the theoretical one-step ahead prediction error.We have discussed this before, is nothing but the white-noise. So this is just a reaffirmation of a fact that we have learnt earlier. Until now we have spoken about predictions, now we are talking about what is left behind, which is a prediction error. And that theoretically, one-step ahead prediction error is white-noise. That’s fairly obvious to see. All you have to do is bring the expression for y, I had from here, that we have derived. And subtract it from y. And you’ll be left with e[k]. So, in other words, the white-noise is nothing but whatever you could not predict in a one-step ahead prediction, which we already know. It is indeed the unpredictable portion. For this reason, this white-noise is also known as innovation. In the sense that you imagine there is a creator, or there is someone who is generating this data, that someone is being very innovative. Every time you predict, the person beats you with this e[k]. Right? That is some new thing that is happening at every k and that's why it is called an innovation. So the person is being very innovative. Every time you say, okay, this is going to be a prediction, the process beats you by saying, no, no, I'm innovating further, you will not be able to predict that. So innovations is a very common term used for white-noise, or one-step ahead prediction errors.

(Refer Slide Time: 05:46)



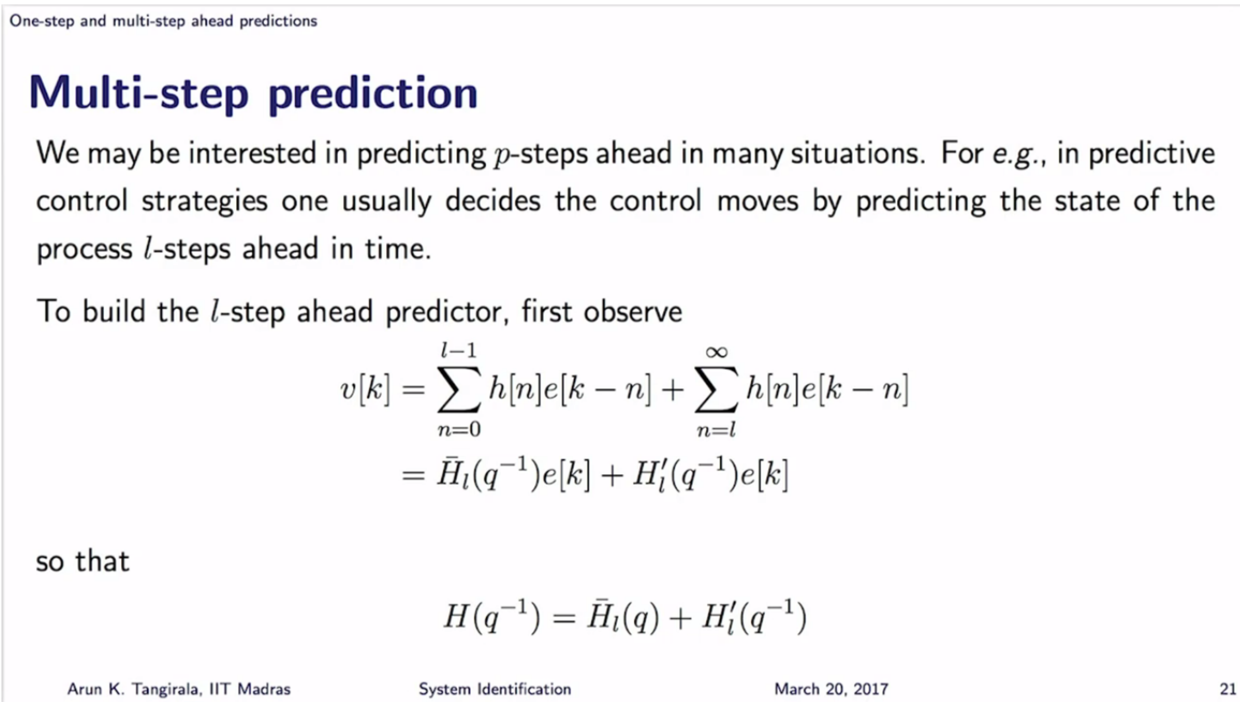
And what we have looked at is how to express a prediction in a transfer function form, in a difference equation form, now we are looking at how to express a prediction in a linear regression form. Right? Transfer function form is for compactness. Difference equation form is for implementation and understanding. Linear regression form is for estimating the parameters. We have talked about this already. We have talked about pseudo linear regression form and so on.So if you take FIR and ARX model structures, as we have already discussed, the predictions can be expressed in a standard linear regression form, right? What is a linear regression form? Where you are able to express the approximation or prediction as a linear combination of some regresses, that is why, this is called a linear regression form. So by the very statement that we have made on the board, the regressor this Psi is a regression, in general thetaa vector of size P by 1,therefore Psi k is also P by 1, column vector. So although I don't indicate it, it is also a vector. It consists of some regressors. A better word for that is explanatory variable. Although we call them as regressors, Psi is essentially being used to explain y, right? It's being used in explaining y. And that's why the variables that sit in Psi are called explanatory variables. So, in FIR the regressoras we know is made up of past inputs allow. So you are explaining y, solely using past inputs. Whereas in an ARX that is your prediction, let me not say explaining, you're predicting y solely using past inputs and whereas in ARX you're using both past inputs and outputs. When I say output here, it'sa measurement not the truth. Whereas with an ARMAX we have already discussed this, structures like ARMAX and OE do not lend themselves to a noiselinear regression form. Because if you go back to the difference equation form, you can see clearly, right? In the difference equation form, how do you write the linear regression from? You go back to the difference equation form, identify the regressors, identify the parameters. So for an ARMAX here, if you had toright, of course,we have now. Earlier that is last time, when we wrote a pseudo linear regression form for ARMAX, what we're the regressors that we wrote, recall for an ARMAX one comma one comma one, you can even look up your notes. Well, there is also delay parameter, but I am ignoring that right now. What were the regressors that we wrote? To be additionally a delay also. Consisted of-- very good, minus y[k] minus 1, u[k] minus 1 and E or you can say one-step aheadprediction error, correct? Whereas in the slide that I show here, so, sorry, if you look at the difference equation form I have-- I do not have epsilon, but I have y hat. Right? What are the parameters here by the way associated to theseregressors. Theta is minus a, so a1 b1 c1. Whereas the difference equation form that I've here is slightly different, but essentially the same. All I have done is replaced epsilon with y minus y hat, that's all. Okay? And so instead of having c1 epsilon k minus 1, I have c1 times y[k] minus 1 minus y hat k minus one, that's all.

(Refer Slide Time: 10:33)



So let us go back to the regression-- pseudo linear regression from, you will get the samething, here. So in general when you have an ARMAX model of nanbnc, these are your regressors. Clearly as we have argued earlier, these prediction errors themselves had a function of the model. Which means a regressor is also a function of theta. Therefore strictly speaking ARMAX, for ARMAX models I cannot write a pure linear regression form. But, if somehow some other model or this model at some it may not be the optimum model, but some model of this form is given to me, I will be able to generate the epsilons. Because given a model I can compute y hat and then I can compute epsilon. In other words if I know theta, at some local value not necessarily the optimal value, then I can construct regressor. So at a value of theta Psi I can write a linear regression from. Okay? So for ARMAX and OE andBJ, I can only write this way. Of course this theta can be the same as this or slightly different from this. When you are able to write in this way, then we call this as a pseudo linear regression from. It's not strictly linear, but linear in some sense that is if theta is given at some local point, then I can construct my regressor and I can write the linear regression equation. Advantage of this is that as we have discussed earlier, I can set up an iterative algorithm for estimating the parameters. Okay. So, let's move on quickly to multi-step ahead prediction.

(Refer Slide Time: 12:40)



Very often we may want to predict not just one-step ahead, but l-steps ahead. This is quite common. If you take a game of chess, when I'm playing, whether it's me or the opponent, I would predict a few moves ahead. Not just one move ahead, correct? What is the information that I use for my prediction? Whatever moves have been made until now, that is information I have until now and a model that means my understanding of my opponent. That's a model. I can't write any equation, but I have a model. I'm going to use these both and make a one-step ahead, two-step ahead, l-step ahead, prediction, correct? Computers are supposed to be capable of this deep blue and so on, they will make many steps ahead prediction. That is what now, we are going to look at the only difference then-- that is we are going to derive a formal expression and then we will observe a very important-- beautiful and important perspective of noise models. Earlier we have said one perspective of changing the noise model amounts to pre fill drain the data, right? If I change the noise model, if I move from ARX to OE, it amounts to pre fill drain, if I vice versa. Once we finish deriving the l-stepahead predictionexpression, we will learn another beautiful perspective about noise models. So, let's quickly derive this, it's very straightforward. Again the starting point is-- see always a challenge is writing the prediction expression forv[k]. Once I'm done with that the rest is fairly easy. So what we do is, we go back to the convolution equation, earlier broke it up into e[k]and the rest. Because I was interested in one-step ahead prediction, but now I am interested in l-step ahead prediction. So we are going to break up that convolution equation into two parts. One summation that runs from 0 to l minus 1, okay? Why are we doing that? Because what is l-step ahead prediction? L-step ahead prediction is, we v hat of k given k minis l or hat of k plus l given k, doesn't matter, same. For stationary processes it doesn't matter. So we will look at this. We are interested in, v hat of k given k minis l. Which means I have information only up to k minus l. And that is a reason,I'm breaking this up into two parts. One that runs up to l minus 1, and the rest that runs from n equals l to infinity. So I we're to ask you, in the summation, what do I know whether directly or indirectly? Which summation, I mean which-- I'm given H already, when I'm deriving predictions I assume H is known, right? In which of the summations I have the information. Do I have the information on terms in both summations, or only in one, or partly in one, or none. Remember I'll be given only v[k]'s, but given v[k]'s, we know theoretically, we can recover e[k]. So for now assume e[k] is kind of given. But in an l-step ahead prediction, I'm given up to what point? K minus l, correct? That means, I'm given up tov[k] minus l. For a moment, let us assume and given up to e[k] minus l instead of v[k] minus l, because I can theoretically recover. Therefore which summation do I know? Second, very good. The first summation, I do not know. Can I predict those e[k]'s in the first summationusing the other e[k]'s? Yes or no? For example here, in the first summation I have e[k], e[k] minus 1, e[k] minus 2 up to e[k]minus l plus 1. Can I predict any of those using the e[k]'s that I know? In the second term, do I know-- can I predict? Yes or no? What do you think? Is there information in e[k] minus l, e[k] minus l minus 1, e[k] minus l minus 2 and so on? About e[k]for example, is there any information? What do you think? Very simple question. Yes or no? Right? Correct. White noise is given how much [ever passed18:02] it's not going to help me improve the prediction. So there is nothing in e[k]minus l, e[k]minus lminus 1, there's nothing much to think here. You're just applying the definition of white noise again and again and again. That's all we're doing. So which means there is nothing in the second term that will help me predicte[k]. What about e[k] minus 1? What do you think? Is there anything in e[k] minus-- let me ask you, is there anything-- so what I am given? I am given up to v[k] minus l. So here I have-- and so on. So this is and given up to this point. And we want to know, if you can predict e[k], so it was easy. Why is it taking so long to answer this question? As to whether there is anything here that will help me predict this. No, right? Because all these are the result of past shock waves. There's nothing in the past which will help me predict this, correct? So, what is a big deal now? Okay, I have realized that there is nothing in this information that can help me predict this, so what? How do I use that to construct my prediction? What is the next step? Why did they break this up into two parts? Tell me, what you think? Why is it of importance, that question that we asked?

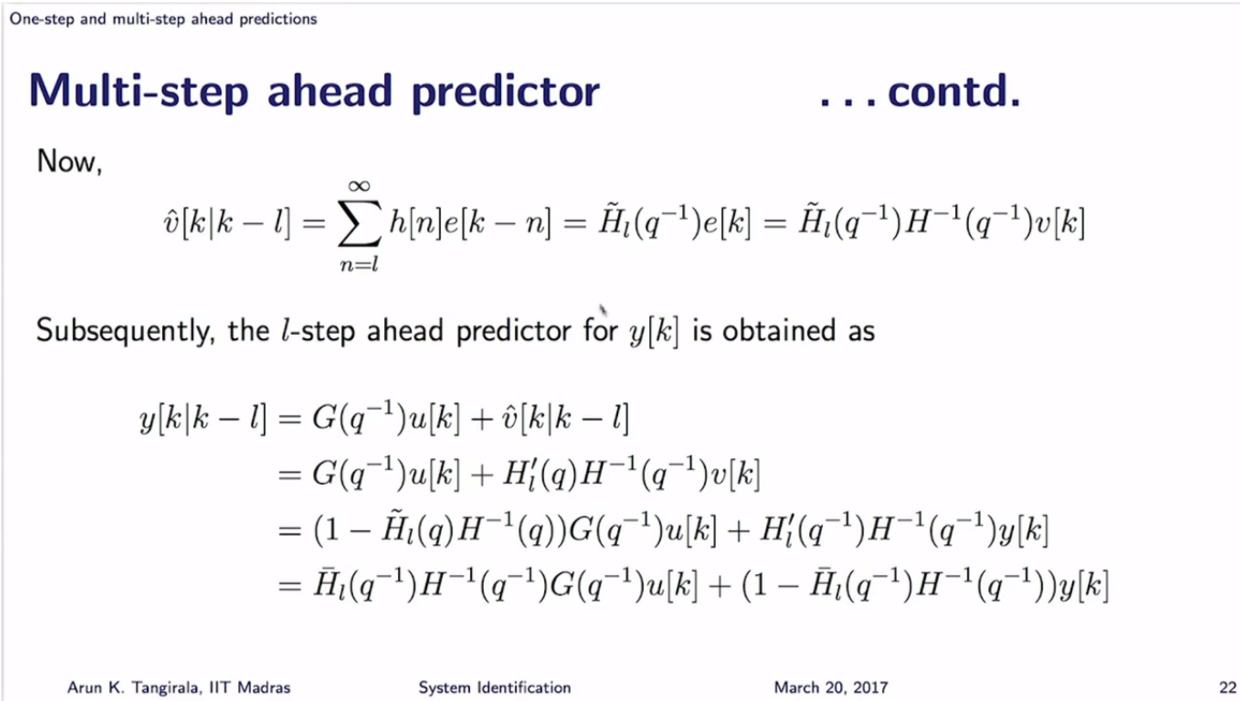
Student 1: [20:35].

Correct, given k minus-- given information up there. Do you know, now can we do anything about the first term? Can we make any prediction at all?

Student 1: No, [20:50].

Yeah, that's what it is. On the right hand side I have white noise, right? So which means, in an l-step ahead prediction, now, the first term will be absent. What is the difference between this and one-step ahead? Conceptually there is no difference. Except that in one-step ahead there was only one term, in the first term you had only e[k]. And the rest was all sitting in the second term, because we had information up to v[k] minus 1. Given information up to v[k] minus 1, theoretically I know everything about e[k] up to e[k] minus 1. But not about e[k], so I pulled that out. So that in my prediction that will not participate. Now, we have pulled out e[k], e[k] minus 1, up to e[k] minus l plus 1. Why again the same story? Because they will not participate in the prediction, okay? So you do understand this point carefully and clearly. Now introduce two transfer functions. H of q inverse we already know. Introduce two transfer functions here. One is your H bar and another is H prime. What is H bar?

(Refer Slide Time: 22:13)

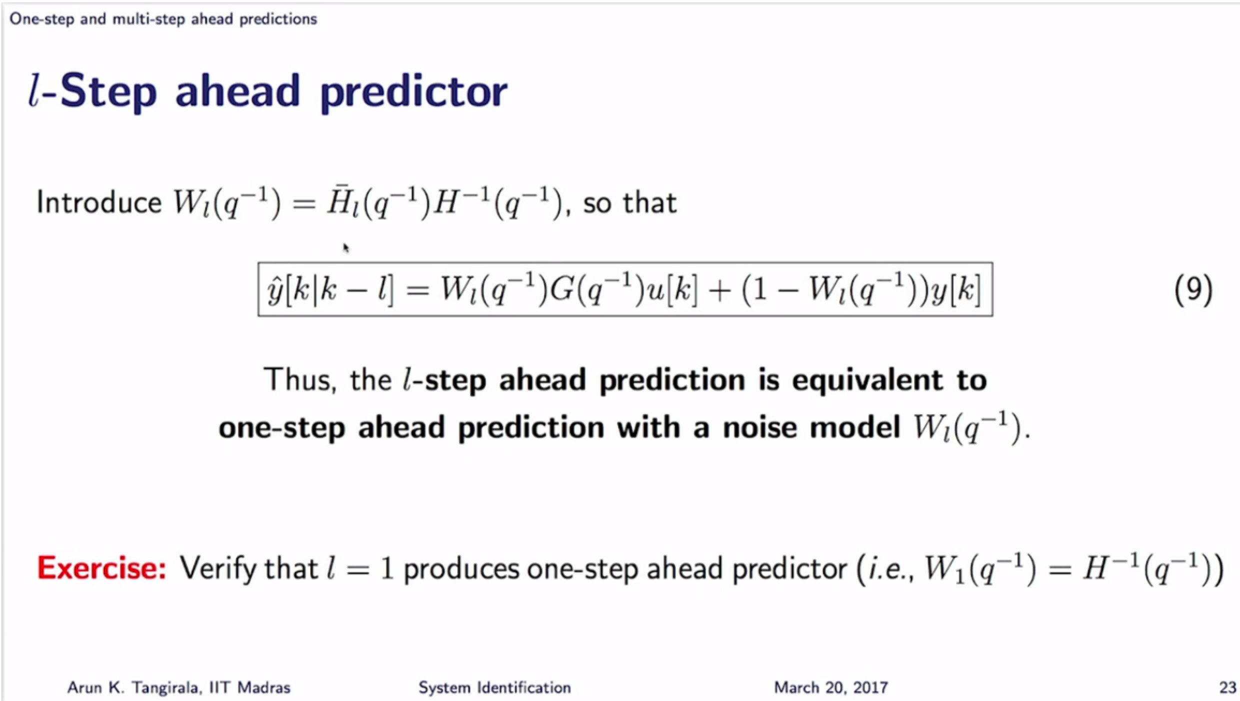


H bar actually is, this summation here. Sigma h[n] q to the minus n, that runs from zero to l minus 1. Okay? And H prime is the rest, whatever is left out. So that you have h as H bar, so all you doing is just splitting H of q inverse into two parts. One part will run from 0 to l minus 1, the other part runs from l to infinite. Accordingly you have H bar and H prime. Now based on our earlier discussion v hat of k given k minus l is only the second term, is that clear to everyone? Because the first term, I cannot predictusing information given to me.Therefore that vanishes from the picture and I'm left with only the second term and what is a second term as per our notation? Here, I have H bar, sorry, I think the notations are reversed here. Because I've used H bar for the second part. Please I'll-- maybe correct that. But for now please follow here all we are saying is now the summation runs from l to infinity. What was the case for one-step ahead prediction? How did the summation-- what did the summation run from? One, that's all l equals to 1. Right? You should keep verifying. Cross checking. Now, I can write this H bar, I'm sorry, here it's read-- it reads H tilde, but it should be H prime. That's only correction that has to be made. This is a prime. Now, I'm going to-- here please ignore the tildes and so on. Just follow the notation that I am saying.The H prime is this one corresponding to the second part and e[k] is H inverse v[k]. So what I have is essentially, v hat of k, given k minus l is H prime the slide reads H tildeor if I'm not I think I'm using H prime, later on. Here, correct. So at this point this is correct. So the only problem is in the topic equation. So H prime of q inverse e[k], but e[k] is itself H inverse q inverse v[k].

Why am I doing this, because I have write a prediction in terms of known quantities, v[k] is known, e[k] is not known directly. Like we did last time we are replacing e[k] with H inverse. Now, let's put together everything. So I have a prediction-- I have now an l-step aheadprediction. You can check if this is indeed what we had for one-step ahead. By the way let me ask you for one-step ahead, what is here H bar and H prime. There is a subscript l here, because it depends on l. If l equal to 1, what is H bar and H prime? Correct, H bar is 1 and the rest Hl-- H prime starts from summation from N equal to 1 to infinity. So we should remember that, we'll use that a bit later. So we put together now, the l-step ahead prediction for y G of q inverse u[k] plus v hat of k given k minus l and I have Gu plus H prime H inverse all right. Now, again we use the same story. We go through the same story. I don't know v [k], but v[k] itself in sys ID, v[k] is not known to me but I know it is y[k] minus Gu, this I know. So I'm going to make that substitution in this expression here, in place of v[k], I'm going to write y[k] minus Gu. So that I get this expression here, everything in terms of u and y. Okay? Now, you see, yes it looks a bit more complicated than the one-step ahead. But let's quickly verify. If this expression here the penultimate expression here, simplifies to the 1 that we know for one-step ahead. For one-step ahead H bar is 1, correct? So which means here, again I brought in H tilde here orH tilde or whatever. So H bar, what I write here is 1 minus H prime there's a mistake here. Quite a few, I'm sorry. So it should be 1minus H prime times H inverse G u plus H prime H inverse y. So let me write that for you, y hat of k given k minus l is G u plus this explosion here. But v[k] is given by that and therefore I have y hat of k given k minus l as, here, 1 minus H prime of q inverse, right? Times G u, because it's a seesaw system and just changing the order.So what happens is in v[k], you substitute y[k] minus G u.

When you do that and then after that you collect together the terms corresponding to u. So you have G u minus H prime H inverse times G. That's why we have written this. What is the next term? Plus I have H prime of q inverse, well, there is subscript l here. There is no l on this one. So H inverse y. So you should quickly verify, when l equal to 1 we do get indeed the one-step ahead prediction. How do you do that? When l equals 1, what is H prime? What is H bar? One, therefore H prime is H minus one, right? That is what we have here at the bottom, correct? So H prime is always H minus H bar. This is true. When l equal to 1 H prime of q inverse is going to be H of q inverse minus 1. So just substitute that here, what do you get? H prime is going to be H minus 1. Multiply, here H minus 1 times H inverse. What really we left with here, a new plug in? That's a simple algebra. In place of H prime, you're going to plug in this. So, you'll be left only with G H inverse, right? You'll only be left with H inverse. That's correct, so you have now for one-step ahead prediction, if you recall. The filter on the u was H inverse times G, correct? Okay, so the first part is verified. Second part, what happens to this product here? Simply I have to plug in here. What do I get? 1 minus H inverse, which is what I had for one-step ahead. So which means our expression at least is correct. Atl equals 1, It specializes to what we have seen. Yes, it looks a bit more complicated than the one-step ahead, but you just have to remember. Now we'll simplify things. So that it becomes easy to remember.

(Refer Slide Time: 31:59)



By introducing this filter called H, which is Hl prime times H inverse, sorry, H bar. So what we'll do is we'll introduce a new filter call Wl, which is H bar times H. So that the idea is remember H prime is H minus H bar, right? So if I plug that in here, what do I get? So here I have H prime, which is H minus H bar. When I plug this into here, I'll be left with H bar times H inverse. Right? All I'm doing is,I'm rewriting in terms of H bar instead of H prime. Because it becomes easy, I know, l-step-- one-step aheadH bar is 1. So it becomes easy to verify. So with the introduction of this, I can write y hat of k given k minus l as this which is Wl. And what about this here, when I plug in here, I would get, here I have H minus H bar, so I'd get 1 minus Wl, correct? So I have here, y hat of k given k minus l as Wl of q inverse times G plus 1minus Wl inverse-- Wl times y. This looks quite similar to the one-step ahead. What is Wl when equal to 1. What will-- what is Wl when equal to 1?That's all. So it's easy to remember. In general it is some filter times G, which is a function of l, that filter is derived from H plus 1 minus that. So that's a complimentary. So you can see clearly that there is some complementary thing here. You have Wl here, you have 1 minus Wl, right? Now, where l equal to 1, Wl is H inverse. Very nice. Therefore now the l-step ahead prediction is equivalent to one-step ahead prediction. Now, hereit's very interesting. So if you were to write the expression for k minus one I have H inverse q I am going to suppress the dependence on q and the interest of time, so this is what I had for one-step ahead. For l-step ahead I have this. There is no inverse, WGu but W is a function of l, so what do you notice as a similarity between these two? The similarity is as follows. Suppose I had a noise model. Suppose I had a noise model that was Wl inverse. For some structure, so there are two structures now, one which has a noise model G and H. So consider two model structures M1 model one and M2. Model one has G and H. And model two has G and Wl inverse. What can you say about these two models? Is there a relation between these two, in terms of predictions? What is the relation? The l-step aheadprediction of this model is this, correct? What would be the one-step ahead prediction of this model? The same, this one.Right? If I were to give you this and ask you to make an l-step ahead prediction, you would come up with this expression. If I were to give you this model and ask you to make a one-step ahead prediction, you will use the same expression. Am I right? Is that clear? Or if you have-- if you're not understood, you can raise your hand. With M1, I make an l-step ahead prediction. This would be the expression. With M2 I'll make a one-step aheadprediction. Again it's the same expression. Why? Because you think of this as some noise model, right? Call this as some whatever Htilde or whatever you want to call. This is another noise model. What would you do, if I were to ask you to do a one-step ahead prediction, you'll use this expression. That's all. So, the l-step ahead prediction of a model is equivalent to one-step ahead prediction of a noise model, which is Wl inverse. With a noise model of Wl inverse. Actually this should not be Wl, it should be Wl inverse. I'll correct that. So you may ask, what is a big deal about this observation? The big deal about this observation is if I change the noise model, now reverse it the other way around. What we have learnt is if I choose a noisemodel in a specific manner then the one-step ahead and l-step ahead predictions coincide. But if I just change the noise model arbitrarily that means what I'm doing and I'm-- what I'm doing is effectively I'm changing the prediction horizon and that becomes more clear now. When we go to infinite-step ahead prediction you'll understand.