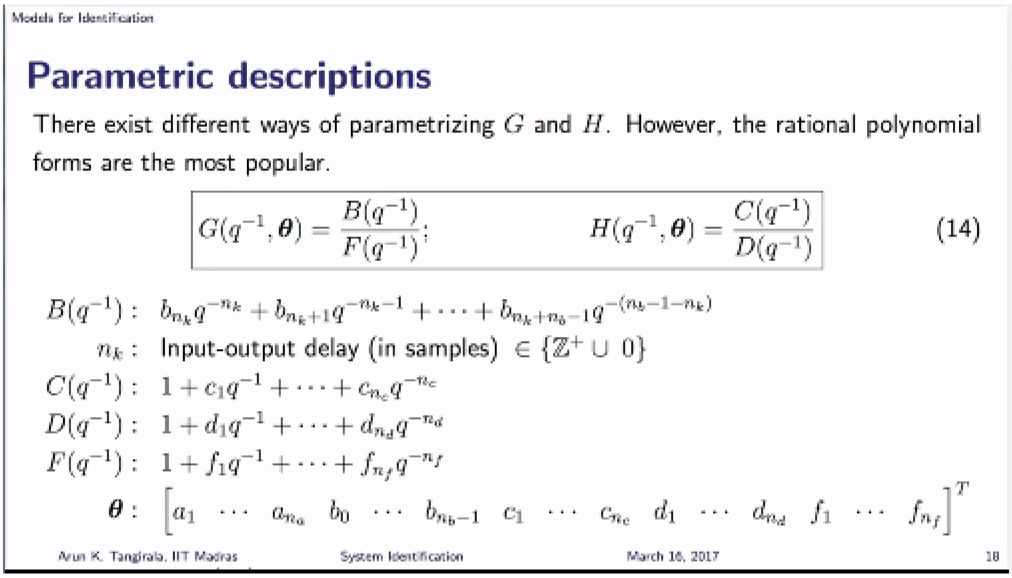
CH5230: System Identification

Models for Identification 3

So let's get going today we'll discuss the parametric models and also briefly talk about predictions. Yesterday we studied the non parametric descriptions but ultimately we developed a parametric model and therefore it's important to know what are the different model structures that are available. I'll also briefly talk about the features of each of this parametric models but then the rest of the story will be completed after we discuss estimation. That is, what is a consequence of choosing a particular model structure and so on. So if you look at the general parametric model family, we know already what parametric model means, essentially it's a consequence of parameterizing the impulse responses of the respective models and we have discussed this at length. Eventually we write the parametric model in the transfer function or the rational polynomial form. But mind you there are other ways of parameterizing the models. This is not the only way. However this is a popular way of parameterizing the model, LTI models.

(Refer Slide Time: 01:08)1



So if I assume that G is parameterized in this way that is as B over F where B has a set of parameters it's a polynomial, f is polynomial it has it's own set of parameters and likewise h is parameterized as C over D. Then the goal in identification is given input output data estimated parameters of B C D and F and in addition we should also take into account that there could be a delay. And the delay estimate is typically obtained from the non parametric modeling part. Either you look at the impulsive response or you'll use the phase of the cross spectral density. We have not talking about how the cross spectral density is estimate and so on, at a later stage we look at that. So the general form of this polynomials have given on the screen for you. As you can see there are a number of parameters that have to be estimated for the general parametric model. Now you can work with much simpler models than this. In other words, you do not have to have all the polynomials being active. Right. For example. I'll come back to that slide.

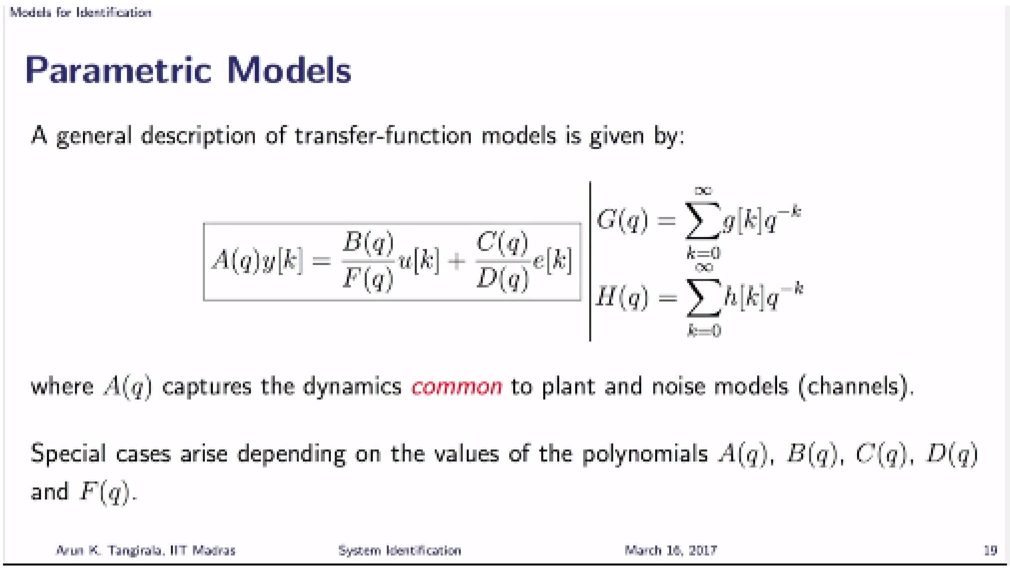
For example, if I set a AFC and D to 1 which means I have decided there is nothing to estimate there in those polynomials then I get an FIR, right. You can see that clearly because what you have is Y equals B over F u. So you have Y equals B over F. I'm going to suppress the dependency on a shift operator plus C over D e. Again BCDF, they're all polynomials I've just suppressed a dependency on the shift operator. If F is 1 and then C and D are 1 then you get so F equal C equals D equals 1. Then you have the FIR model. Right. Because B is a finite order polynomial. So depending on what you assume about each of these polynomials you'll get the specific model structure.

But this is not just some video game that is, okay. Now I'll try setting C and D to 1. Let me try if this fits. It's not exactly that. In fact, it's much, much better than that. It's not just a guess game there. Setting a certain polynomial to unity or not setting it to a unity. Has its implications on what you assume about the plant and how you assume the uncertainties are entering the process measurement. Okay. But before we discuss the different possibilities it is customary to actually-- I'm going to skip we've already written this. It is actually customary to write. I'm sorry this is what I want to discuss-- to extract the commonality between B and F. Sorry the F and D polynomials, they are the denominator polynomials. If you look at it F captures the dynamics of the plant model. D captures the dynamics of the noise model, right.

It is possible depending on how the system is wired, how the noise enters the system and the measurement and so on that the plant and or the deterministic and noise model channels can have common dynamics. We would like to highlight that common dynamics and capture that in a so that now we are just rewriting the same model. Yeah, I have used the same notation F and D but now you have to understand that F and D have nothing in common. Whereas in the previous model F and D could have some factors, some poles common in them. Now we have taken out all the common poles when they say common dynamics are essentially common poles and put them into a of q. When a of q is 1 in this kind of a structure. This is called a prediction error model structure and structure. Then you have-- you are explicitly stating in this when you set a equals 1 that look there is nothing common between plant and noise model as far as the dynamics are concerned. Typically in the numerator you would not have much in common. It's only the dynamics that can share some commonality and then you are saying look there is really nothing common that the plant and noise models share. On the other hand, if a is not equal to 1. You are saying that there is something common to the deterministic and the stochastic channel. But what does it mean, does there exist a physical system like that? Well, most of the times, no. But mathematically that can offer some convenience as we shall shortly discuss.

So hopefully now you understand what the a stands there for. Now depending on what you assume for each of these polynomials the models get different names. So as I said, if you assume AF C and D to 1 then you'll get an FIR model. On the other hand if I assume that F, D and C are all one which means that A and B are not 1, it's understood. Whatever has been set to 1 is 1, whatever has not been se to 1 is not 1, I have to estimate it. So when I set F, D and C to1 I have an ARX model we'll talk about each of this model structure soon. And the other model that's of interest is OE, of course, ARMAX models are also equally interesting. Where you assume that A, C and D are 1. No. As I said, you should not think of this as a game as a play where you decide, okay, this polynomial should be 1 that polynomial should not be 1and so on. It has its own implications and it'll become clear when we talk about each of these model structures. In the general form where you don't have anything in common that is you're going to estimate B C D and F then we have what is known as a Box-Jenkins model structure. Again we'll talk about that briefly. What is more important for you to remember? Apart from remembering these names you will get familiar with is that assuming a particular model structure means three things. One, that it has its implications on how you are assuming the uncertainties are affecting your measurement.

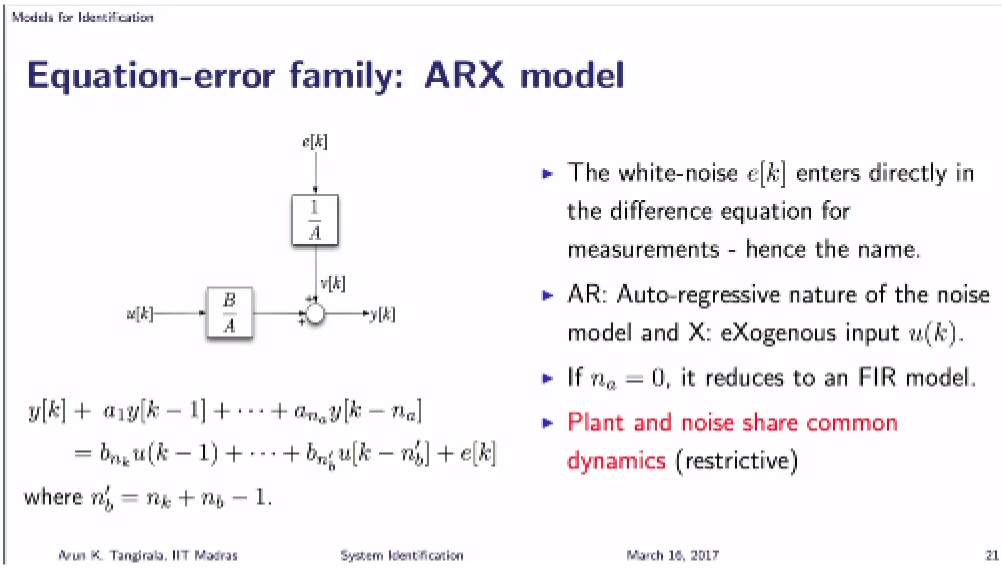
(Refer Slide Time: 8:26)



That is one implication. And secondly, whether G and H jointly or independently parameterized. And at this point you may not appreciate so much but if you recall the discussion that we had on liquid level case study. I had mentioned at that time that if you have g and h being jointly parameterized that means you're kind of tying both hands together. Then there is going to be a limitation. First of all in your ability to explain large class of processes and theoretically we shall realize later on. That it has its bearing on our ability to recover the true model that is the third consequence. In fact, a very important result in open loop identification is that if you have g and h jointly parameterized then there's no guarantee that you will actually recover the true system. Yet, people work with a lot of such models. In fact, the familiar ARX's model assumes that that is joint parameterization. Whereas the output error model and the Box-Jenkins model have independent parameterization. Of course in output error model the noise is not even parameterized. But still we say there is nothing in common.

There are some theoretical advantages with working in working with those models structures. Typically you can begin with an OE model or an ARX model. If you begin with an OE model there are some advantages. While you begin with an ARX model there are some advantages but there are demerits to both. But for the output error model, the disadvantages outweigh the advantages. So a general choice of model if you don't know anything should be an output error model. And we'll talk about that. So let's talk each of about each of these models structures a bit more in detail. So the first in order, of course, we are disregarding the FIR model.

(Refer Slide Time: 10:28)

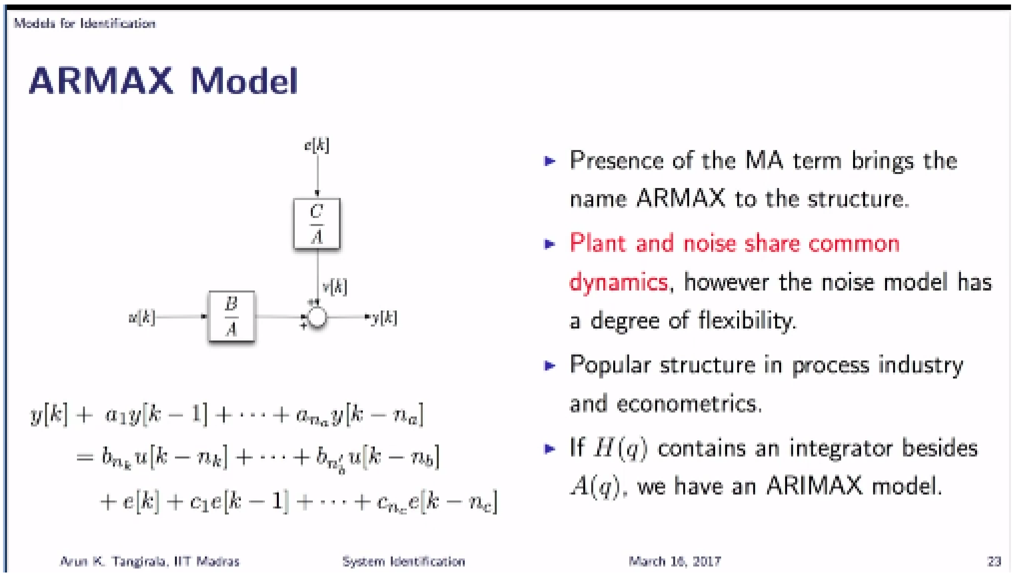


Now if you realize, I have put the FIR model in the parametric model family. But earlier, that is yesterday when we discussed non parametric models there also we discussed FIR models. So this FIR model is like a cat on the wall. It belongs to both. It's like a joker card and deck of cards. It can be viewed as a parametric model. And it can be viewed as non parametric model, both. Okay. So let's move on to the ARX model. The ARX model or the Equation-error model as it is known, the full form of it is auto regressive exogenous model is a very popular model for one reason and I'll talk about that reason in the next slide. But look at the architecture of an ARX model. It says, G is b over a and h is 1 over a, which means the plant and noise models completely share the dynamics. And it's hard to imagine a physical process that has that kind of an architecture. Because you believe that vk contains effects of output disturbance. Why will it affect the output through the same channel as the input? Right. Suppose vk is typically, vk consists of sensor noise effects of unmeasured disturbance. So why would you think that this output disturbance or sensor noise will affect output in the same way as the input? It's actually quite unrealistic and it is indeed the case, it is unrealistic. Yet, these ARX models are extremely popular.

People don't even know that they're working with ARX models. Many people don't even know. They just write an equation. You know the standard difference equation that you write directly in terms of measurement. That actually works out to be an ARX model. Now the other thing that I want to say is as much as you remember what G and H are or what is a parameterization for each of this models structures, you should also be equally comfortable with the difference equation form. Okay. The writing in the transfer function form is helpful because it gives you compact representation but ultimately when you estimate the parameters, when you implement the model it's a difference equation form that comes handy. So I have given you the difference equation form for ARX model structure. As you can see, it's a linear difference equation for the measurement. And observe something here ek which is a white noise enters the equation straightaway. In the output error model ek enters the measurement, so you see the difference. Here, ek is entering the difference equation, difference equation for the measurement. Whereas in the output error model structure as you'll see shortly, ek directly enters a measurement and that is why it's called output error model.

This error that we're talking about is white noise. Here ek is entering the equation and you can say that is one of the reason it is called equation error model. There is another reason but we'll not worry about that. These are very popular because when I write the prediction form of this, what would be the predictor? Suppose, I had-- I have an ARX 1, 1, 1. Now, this notation you should get used to.

(Refer Slide Time: 13:52)



This corresponds to the order of A or in other words, na, this is nb and this is delay nk. Because this is how you will specify in the MATLAB routine as well. You have to specify, how many parameters you have in the a polynomial? How many in the B? And what is the delay? The delay information comes from the non-parametric model.

And you have to supply at least in the [SIS ID] tool box you have to supply the number of parameters or specify in the alphabetical order of the polynomial names. So for example, in ARX it's okay, you have a and b. In ARMAX, you'll have ABC. In OE, you will have B and F. Okay. So you should not get confused because there you specify NB, NF and so on. So it is the alphabetical order unfortunately that you have to keep in mind. Anyway, so if I have an ARX 1, 1, 1 then the difference equation is y k plus a 1 yk minus 1 equals B 1 uk minus 1. Alright. So that is difference equation form. Is there something missing? Plus e. It's important. Now I just know said that his ARX models are popular because the predictors are the predictions that I'm going to make, predictions of what of the measurement. Why are linear in parameters? And let's see, why that is true? Typically we're interested in one step ahead prediction. So the construct the one step ahead prediction, we'll do, we'll learn the formal way of constructing the one step ahead prediction shortly. But if you were to look at it, how will you write the one step ahead prediction? You take all the terms that you know to the right. By the way, when we write here, typically we say, model is also given. Maybe in the previous slide, I should have talked about this. The general. If you look at a general polynomials here, the pragmatic family, there is a theta that we talk about. This theta consists of all the parameters that your are estimating, right.

That is now understood somewhat obviously. So here, what would be theta? A1 and B1. Very good. So theta would be A1 and B1, so if I were to be given A1 and B, If I were to be given all the measurements up to k minus 1. What would be my prediction of y? That is what this notation means. And meanwhile conditional expectation is the best prediction in a minimum means square error sense. So when you apply that all you have to do is, rewrite this as minus a1 plus b1, so now what would be the predictor? Or what would be the prediction expression here? What would I get? If I look at the right hand side terms, I'm giving measurements up to k minus 1. I'm also given the model. Then what would be my prediction? Sorry. Simply RHS, that it? Without error. Why is that? Because ek.. White Noise by definition as there is no information in the past that you can actually use to improve the prediction of ek beyond its average. Right. And we assume ek to be 0 mean.

So the first term is known because I'm given both I'm also given the input and the model, the expectation of ek given k minus 1 and theta is going to be 0 because there is nothing in the past that can actually-- strictly speaking it should be the mean. I keep saying that so that you remember it. So now you look at the predictor then if I were to ask you whether it is linear or non-linear and parameters, what would be your answer? Linear or non-linear in parameters. Linear. Right.? Because I can ask several questions, is it linear in u k. Is it linear in y. Here we're interested in linearity in theta. Why are we interested in this because, ultimately when we estimate the model parameters. So he had we wrote it assuming model is given. Suppose model is not given. Then I'll set up an optimization problem such that the optimization will drive the prediction errors or you can say will drive the predictions very close to y. As close as possible to y. In that case typically I will drive it in the least square sense. And we know very well, at least if you don't know you will know shortly and we discuss estimation that whenever the predictor is linear in unknowns, you can think of y hat has a prediction as an approximation whichever way you like.

Whenever this prediction or approximation is a linear function of the unknowns that you are optimizing and you set up a least squares problem to estimate those parameters we call that as a linear least squares problem. And linear least squares problem have unique solutions. That's the biggest advantage of working with an ARX model structure. So through this simple exercise you should realize that choosing a model structure has its bearing on the estimation effort. If I choose an ARX model what kind of a predictor will I get in terms of parameters? Linear predictor. So what is a big deal about linear predictor? When I estimate the parameters using least squares with linear predictors, when I say linear it's in linear and parameters. I have a unique solution. I have an analytical solution that I can implement.

I do not have to seek a numerical optimum. And that has obviously a lot of advantages in many ways. One I got a unique solution, two it becomes easy for me to derive the expressions on the errors of the parameter estimates. You will see, you learn later on estimation that deriving the expressions for the errors and parameter estimates is easy if the estimatorof the predicted is linear or of the solution is unique and there is a closed form expression and so on. So there are several advantages of working with an ARX model. But what is a flip side to choosing this ARX model?

I'm sorry. So what does it mean?

First of all my hands are tied. Secondly quite unrealistic. And as I said theoretically there is a result which says if the plant and noise models are jointly parameterized. Either is no guarantee that you will recover the true model. Unless the truth itself is ARX. Unless the truth itself is ARX, is the truth is not ARX there is no guarantee that you'll recover the G correctly for example. Whereas with output error model that is not the case. That's a big advantage with output error model. So you can see choosing a model structure means a lot of things. It is not just playing around because you have lot of time, you do not know which one to work with.

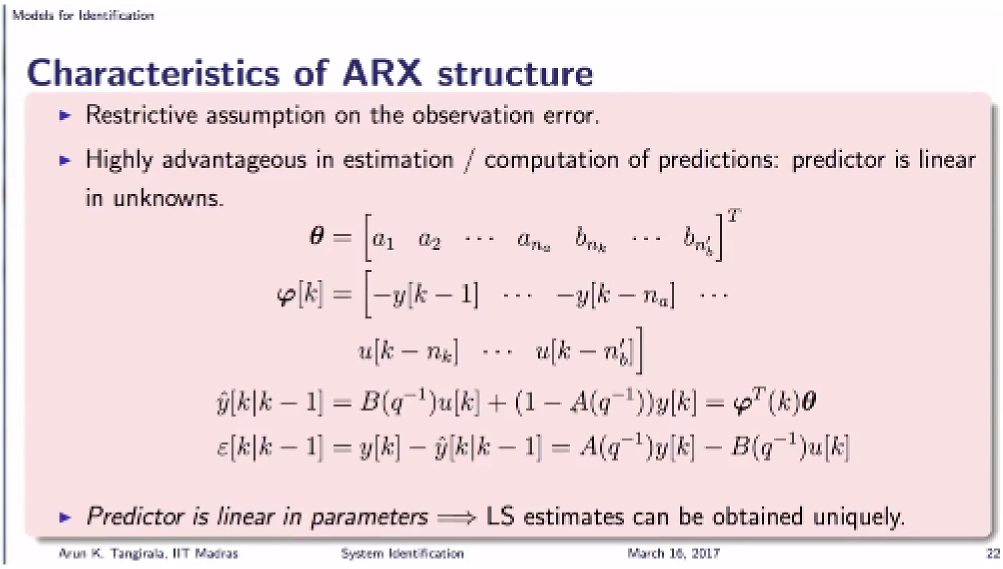
That is what distinguishes you between you and a blind user or a beginner. Who does not know, what is a consequence of choosing a model structure? There are several implications. What you're assuming about the physics of the problem, at least the architecture of the process. What bearing it has on the estimation effort, what bearing it has on your ability to recover the true model? You have to always keep these three factors in mind and have a checklist or at least answer these three before you to the model structure. Until, you know, get used to it. There is another advantage to with choosing with working with ARX models which is that because you have a unique solution there exists an algorithm which estimates ARX models of several orders simultaneously.

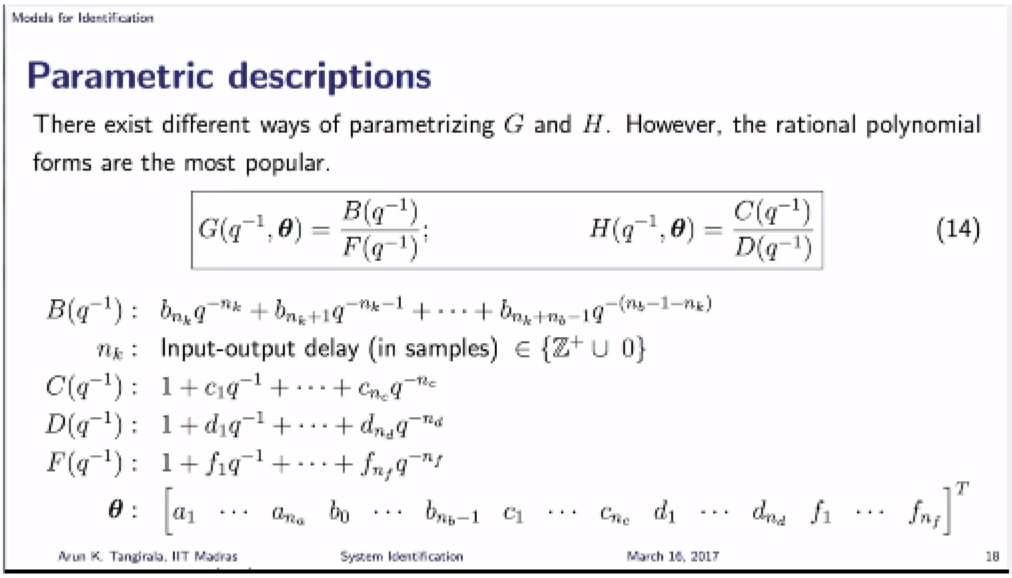
That should not come so much as a surprise because you have the Durbin-Levinson algorithm also, right? For AR models, that allows you to recursively estimate. So there is a recursion that you can easily build, a build into the algorithm so that you can estimate ARX models of various order simultaneously also. Not only recursively but simultaneously. So there are several advantages to the ARX model computationally from an estimation viewpoint but the flip side is this. So you see there is always, there is a tradeoff in nature. And I've talked about this and I will. I've given you the expression, for each model structure what I have given is, the parameters that have to be estimated, their regressor. Now, quite often, we write this predictor in a regression form, if possible.

So for example here, I can write this in a linear regression form. What do we mean by linear regression? This shy that I have written is a set of regressors. And theta is a vector of unknowns. When you have a prediction or an approximation of this form, we already know, this is a linear equation in unknowns. It's called linear regression. And then you can borrow all the ideas from statistics on how to estimate parameters of a linear regression here. Right. So it's customary to write the predictor in a linear regression form and you should get used to these forms but not all model structures are friendly. Here, I am able to write this in a linear regression form because it's linear in the parameters but very soon you'll see for ARMAX for example, that's not the case. So what is it aggressive here? I've written it is in this form. But what is shy exactly?

t is a vector. And what would be the victor? Sorry. Is it yk minus 1 or minus yk minus 1? Minus 1 because theta by default is a1 and b1. So minus yk minus 1 and uk minus 1. Good. So all you have to do is, set up your regressors and then use the linear least squares and go ahead and solve the parameters.

(Refer Slide Time: 25:02)





So that is what I've written here. Of course, I've given this expression here b of q inverse yk plus 1 minus a of q inverse uk. That prediction expression will derive, how to write? What I'll show you later on is given a model structure straight away, how do you write a prediction expression. Her we went through this difference equation and then we talked about conditional expectation and so on. But you can bypass all of that and straightaway write the prediction expression given G and H. Which is very useful. And I also give you the expression for prediction error. Because ultimately you will be minimizing be some form function of prediction errors.