CH5230: System Identification

Journey into Identification

(Case Studies) 13

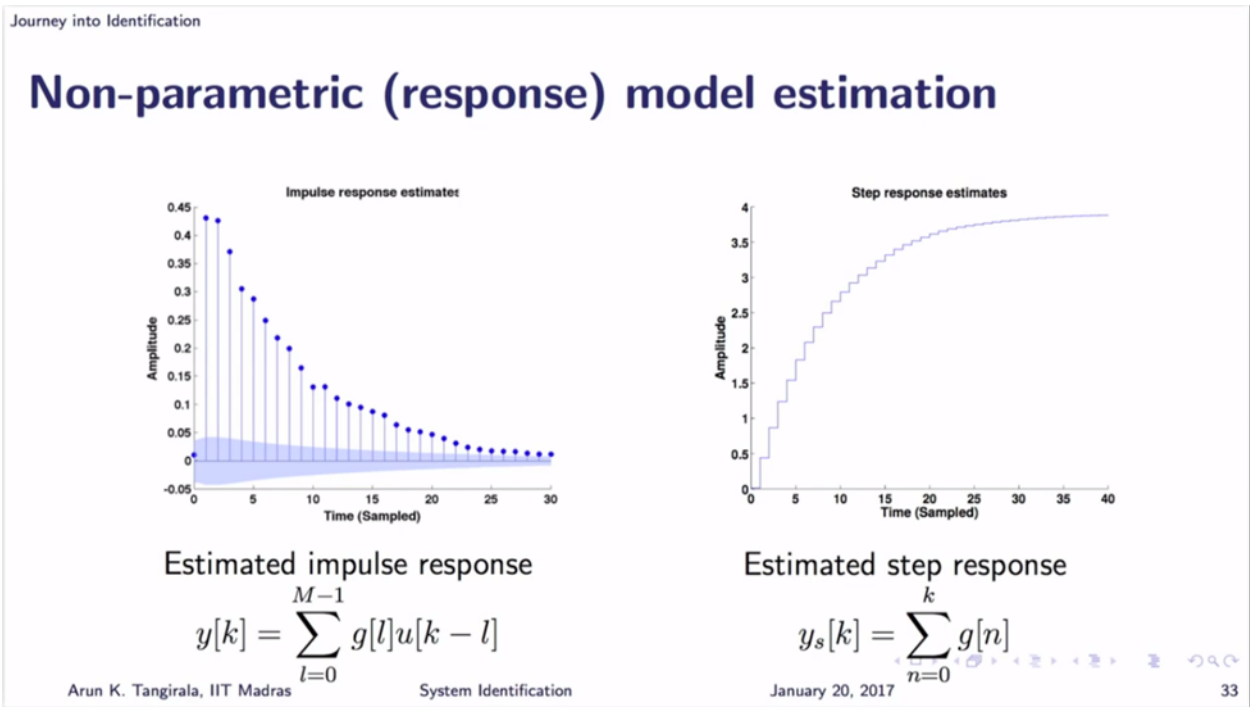
And now we turn to a model that will help us predict. It doesn't mean that these models will not allow me to predict. Even this convolution, with just, with this convolution equation I can predict. Once I have Gs, can't I use this model to make a prediction, in future? I can. But then why am I turning to a different model? What more do I need? There is so much that I have learnt through a non-parametric analysis. Why we call this non-parametric will become clear shortly. What more do I need? I have a model that'll allow me to predict, right? Why should I turn to a different model or a difference equation model? Any ideas, any arguments? Do we need to? Can we close a case study and say, look, I know a lot about the system now. And I'm ready now, I mean in the sense I have the model that'll do a good job over the prediction. At least mentioned one or two things that you think are the shortcomings of this response based descriptions that we have gone through.

Can we have predictions gone through it?

In what sense? Why? Why do you think these models will not-- let say, let's talk about the impulse response model. Okay? Let's focus on that. I know, I have estimates of G, correct? And that's it. So, do you think the prediction from this model is going to be not good enough? Why? Everything fine, everything has been taken care of. One, we have not model the noise. Remember we have only focused on the deterministic part. That is point number one, right? Of course, you can get some idea about the noise, but still we have not done a careful modeling of the noise. What is the noise contains some predictability? By definition a random signal is not accurately predictable, agreed. But there may be some element of predictability, do not be ever under the impression that just because the signal is stochastic, you cannot predict it at all.

More appropriate way of understanding it is a random signal is not accurately predictable. You can make a prediction. It's going to be, it can be much better than by just looking at the average. That means there is a possibility of improving your prediction, then an average. Generally we say average is the best prediction. That is true for a certain class of signals, which is called the white noise kind of signals, uncorrelated. There is nothing in the past that helps you predict the future. For such signals average is the best prediction. But if there is some correlation, past has something to tell you about the present or the future. Then you can certainly make a better prediction theoretically and therefore you should exploit that. We have not done that at all, we have not looked at the noise. That is point number one.

(Refer Slide Time: 03:31)



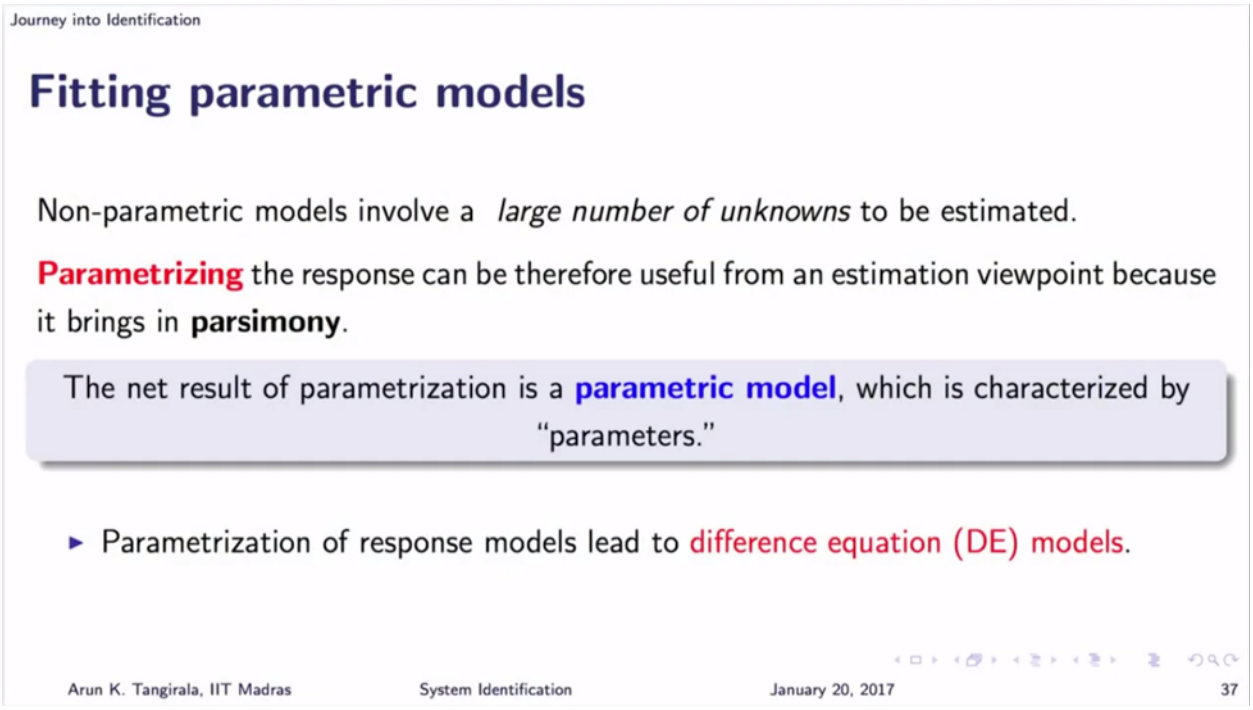
Point number two, if you turn to the impulse response estimates. How many estimates have we roughly? How many unknowns? I'm not going to use the word parameters. How many unknowns have we, roughly estimated? Thirty, right? Roughly about 30? So which means does this mean that a system like a liquid level system has to be characterized by 30 unknowns? Do I need 30 numbers to describe a simple liquid level system? It's a first order, right? When I look at it from a transfer function viewpoint or a differential equation viewpoint, how many unknowns should I roughly expect to see? Only one?Only two?I going to keep asking this until you're sure. Only three? Sure? Now, you would bethink in your mind. Take a few seconds and--

Okay, maybe two or three or four whatever. But we certainly know, it doesn't require this many unknowns, correct? Right? If you were to write in fact it's actually three unknowns including the delay, if you take into account the delay. If it isn't then it's only two, the gain and time constants. Because it's a first order, right? But here we are describing the system with 30 unknowns. And if the system had a larger time constant, the impulse response would have taken a longer time to settle down and as a result I would have estimated even more. From an estimation viewpoint that just too many unknowns that you are estimating from data, you may have a lot of food, lot of data points, but that doesn't still mean I go ahead with this bag of 30 numbers wherever I go for this liquid level system.

I would like to follow the principle of parsimony. Here is where you have to be business minded, right? Where you have to be as miserly as possible, when it comes to characterizing, describing a system. You want to minimize the number of unknowns not only from an estimation viewpoint but also from an implementation viewpoint. The simpler the model is the easier it is to implement the lighter it is. People are looking at lighter things, right? Definitely you're looking at, you know, being able to do everything with a watch or with a small chip in your button and so on, which will have everything including your

house in it. Okay? That is what we have heading to. Therefore from these two viewpoints we are in search of better model descriptions. But those better model descriptions are they require some inputs which now we have our some insights. Which now we have through this analysis, right?

(Refer Slide Time: 06:25)



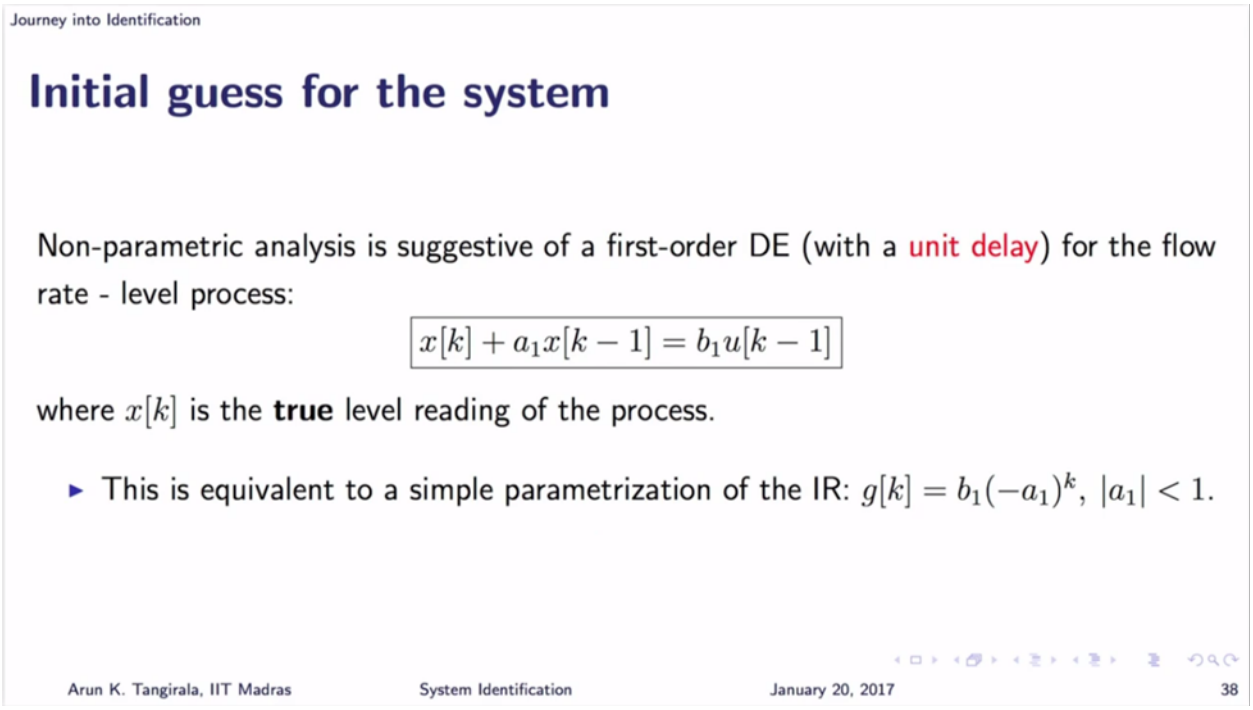
So now let us turn to a more parsimonious model, which is known as a parametric model. So what we're going to do, why are these called parametric models? Well, the reason why these are called parametric models is just now we said this impulse response estimate has 30 unknowns in there. But if you look at the response carefully, it seems to follow some equation of curve. Correct? Maybe I can now try to just fit a curve to this and that is essentially what we known as parameterization. We have learned this in high school parameterization of curve. All parabolas are parameterized in a particular way. All hyperbolas are parameterized in a particular way. There are some curves, but they have an equation and that parameter will tell me the nature of that curve, correct? So here these 30 unknowns that I have individually estimate-- estimated perhaps can be tied together with a thread. And I just need to fit the question of thread. Correct.

So we can, therefore think of a curve that onto which this impulse response estimates are hanging like clothes. And I just need to get that curve. Once I get the equation of the curve, I know the impulse response estimate at any point in time. Correct? So imagine that you're drawing a curve like this. Question is now, what is the equation of that curve? Okay. One possibility, I'm just writing it straight away is to say that perhaps I can write this as b times minus a raise to k minus, on 0 at k less than or equal to 1. You may wonder oh my god, how did this guy write even this model, all of a sudden? Did I know the answer a priori. No, and partly yes. Because I have some background in linear systems theory, without worrying

about the expression what it says is there is an exponential decay. Of course, why there is decay, because I assume that a is less than 1 in magnitude, but that's okay.

We'll worry about that later on. Essentially, this is called an exponential curve in discrete time. Don't think of E to do something always as exponential. Something raise to the power is also exponential. I've use minus a, you could use plus a. That doesn't matter, that system matter of choice. I am postulating that this curve, why did I choose k greater than or equal to 1? Why didn't it begin from k of a 0. Because there is a delay, right? So that information didn't have a priori, I got this from impulse response. I'm presupposing now that this is a model for this curve, I may be wrong. If I'm wrong, the residual analysis will tell me whether I'm wrong or right. That's a beauty. I may make a mistake, this we call-- by the way, this is the impulse response of a first order system. In fact rather than presupposing an equation of curve for the impulse response, I might as well straightaway write a difference equation, right?

(Refer Slide Time: 09:57)

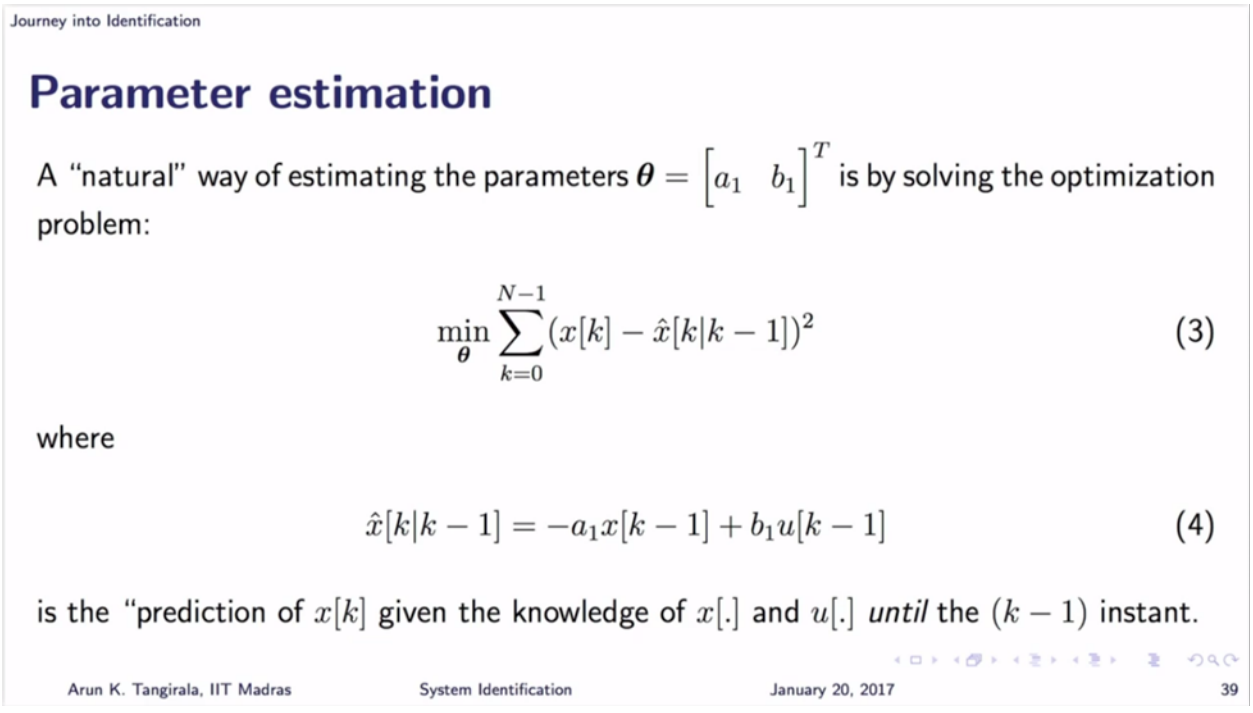


I can write a difference equation straightaway of this form. By the way x is y star for me. This g is still the-- now we'll look at only the discrete time system. So u k, this is y star, which I'm denoting in the slide as x. So y and -- y star and x are the same. There is a reason why I'm using x, will become obvious later on. So here is y. So what I am writing here is a model for g for the discrete time, deterministic LTI system. Either you can straightaway write this first order difference equation. How did we write this first order difference equation? Because I already know it from the step response, the system seems to have first order dynamics and I already know from the impulse response is one. So I am just presupposing that the deterministic LTI system can be describe by this difference equation.

Either I write this way or I start from here. You can sure that both are equivalent. Both are one on one mapping, if you're written a different equation on that front you can in fact you should take the difference equation of that form, you can in fact-- you should take the difference equation form and show that the impulse response of this system is the same that I've written on the board. Except that I have b1and a1 there, that's all. Typically users would start with this. But not many know that writing a difference equation form amounts to parameterization of the responses. That is exactly-- that is extremely important. When we estimated impulse response we didn't assume anything. We just assumed LTI, but the moment I write this difference equation form whether I stated explicitly or not. I am behind the scenes assuming that the response is actually following a certain equation of curve.

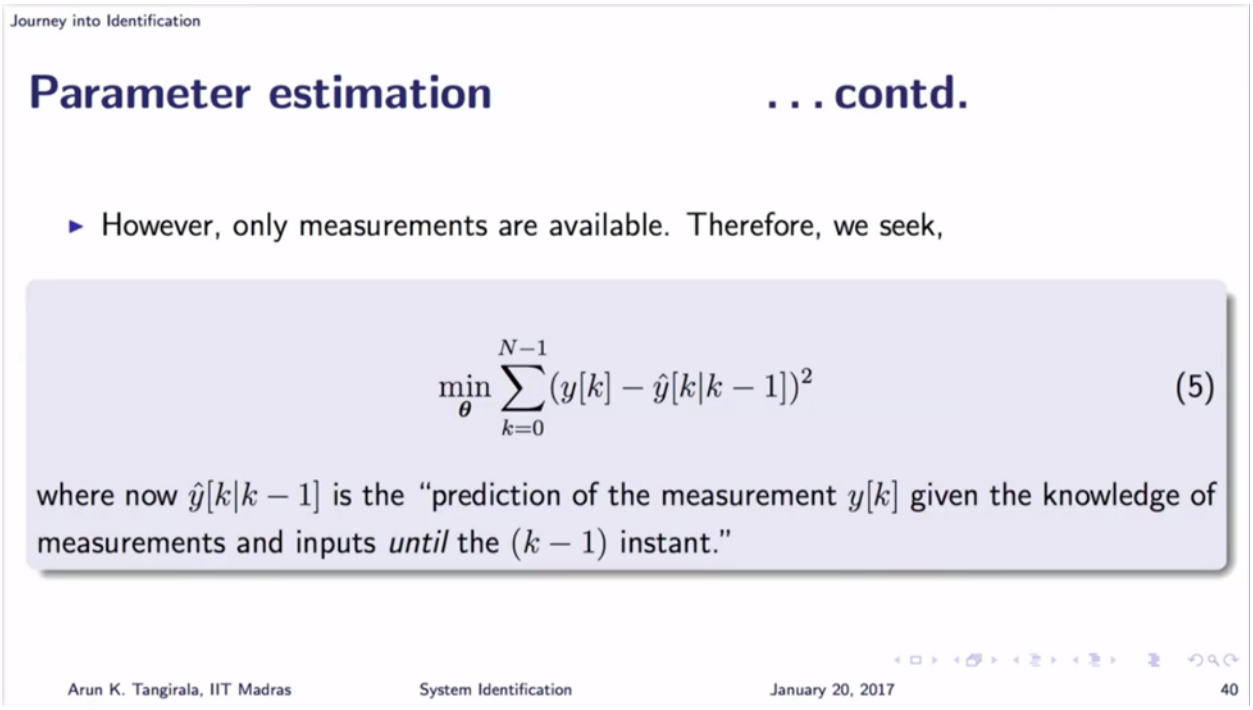
And you-- Although I have written this for impulse response you can show this hold for, the step response and frequency response as well. Okay? So now the goal, we are going to now replace the estimation problem of impulsive responses or step responses or whatever, with a new estimation problem. The question that should come to our mind is, I don't have x. I have only measurement y, right? Unless I make an assumption on v, right, on v I cannot estimate n. I don't have access to x, if I give you x, how many points will you require to estimate a and b? Just two points, it's a noise free equation. Just take two points and estimate. And if the system was pakka linear then fine, it will work. But we have noise and we have nonlinear it is also. So we want to fit an approximate model and we have only measurements with us. So as a result I need to actually turn to the measurements.

(Refer Slide Time: 13:04)



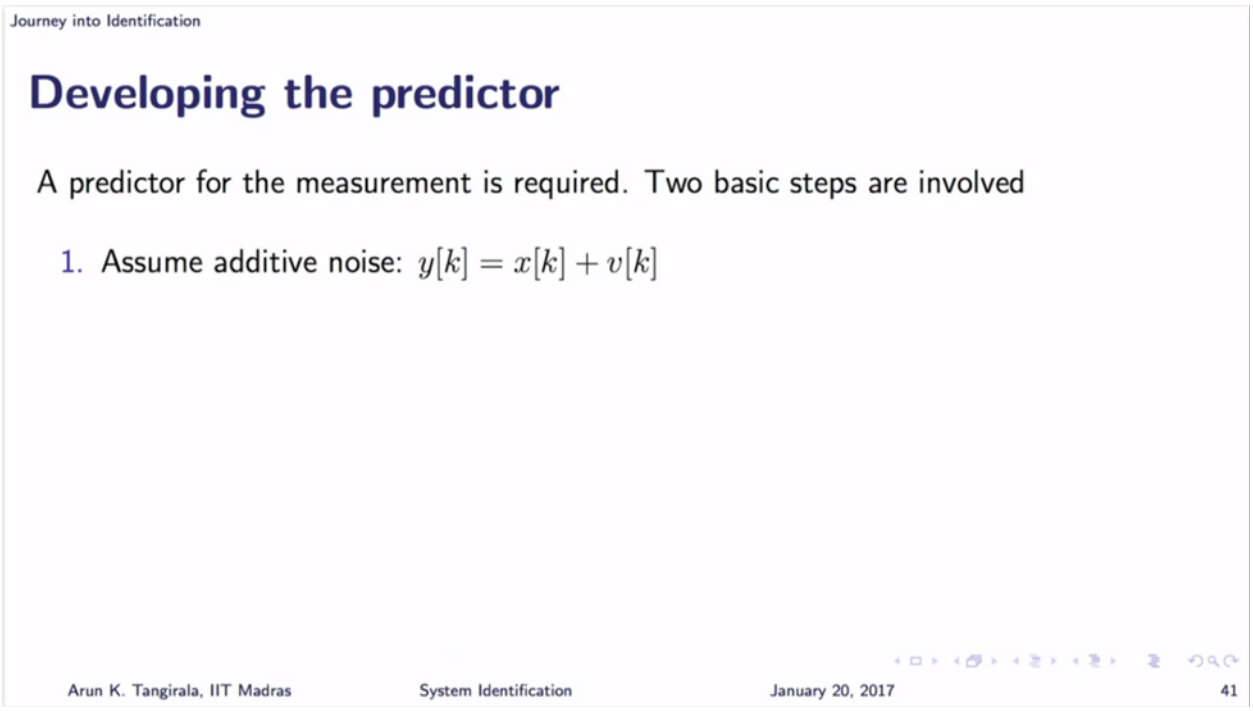
And if ever given x, and the system was non-linear then I would fit a model this form. But unfortunately I did not have x.

(Refer Slide Time: 13:12)



Therefore I have to fit a and b such that the errors between the measurement and the prediction of the measurement is minimized. Specifically the sum square, this-- so-called least squares objective function is solved, optimization problem is solved. What is right here, the measurement, y hat is, y hat of k given k minus 1, so-called one step ahead prediction of the measurement. Given k minus 1 would mean, I have measurements up to k minus 1. Inputs are always given. Which-- what they says is if I were to give you the model, if I were to give you the model and the measurements how would you predict your measurement? Okay, fine. That is y hat. Now, you're telling the optimizer find a and b, tune a and be such that this prediction error is minimized. The question is how do I construct the prediction? This clearly tells you, why we study prediction theory even before we sit to estimate parameters, because we are going to estimate parameters by minimizing prediction errors.

(Refer Slide Time: 14:24)



So in order to construct y hat I need to assume a model between y and x. Because I need to construct y hat, I don't have x. And the assumption that we are making is that y is x plus v. I just take a few more seconds and we'll conclude. This assumption that we have of additive noise is already been stated. I've already said that y is x plus v. Now what is remaining is making a suitable assumption on v. We have a model for x, don't we? We have already come up with a model for x. Now I need to make some-- get some-- assume some model for v. We will show next week, what happens when I make two different assumptions on v?

(Refer Slide Time: 15:07)

Developing the predictor

A predictor for the measurement is required. Two basic steps are involved

1. Assume additive noise: $y[k] = x[k] + v[k]$
2. Postulate a (noise) model for predicting $v[k]$. Two possible options:
   - 2.1 Observation noise is **unpredictable**, *i.e.*, $\hat{v}[k] = 0$.
     This assumption leads to what are known as **output-error models**.

One is that the noise is unpredictable. Okay? That means I'm not going to fit any noise model at all. I'm going to assume that the noise is white. I don't know, I'm just going to assume because I'm given the freedom.

(Refer Slide Time: 15:21)

Developing the predictor

A predictor for the measurement is required. Two basic steps are involved

1. Assume additive noise: $y[k] = x[k] + v[k]$
2. Postulate a (noise) model for predicting $v[k]$. Two possible options:
   - 2.1 Observation noise is **unpredictable**, *i.e.*, $\hat{v}[k] = 0$.
     This assumption leads to what are known as **output-error models**.
   - 2.2 Observation noise is **predictable**, with the requirement that a linear predictor be obtained
     This route takes us to the class of **equation-error models**. Also known as the auto-regressive eXogenous (ARX) models.

And the other route that we will explore is noise is predictable. That means there is a model that I'll build for noise as well. But on top of it, I will demand that the predictor y hat, what is a predictor going to be a function of-- of course a past measurements, but also your model. I will demand that the predictor y hat be a linear function of the unknowns a and b. Why do we want to do that, we'll discuss next week. But these are the two routes that are typically followed, there are widely prevalent in system identification everywhere, the first route is called the output error approach, output error model and the second approach is called equation error approach. And we will see, which we'll see--