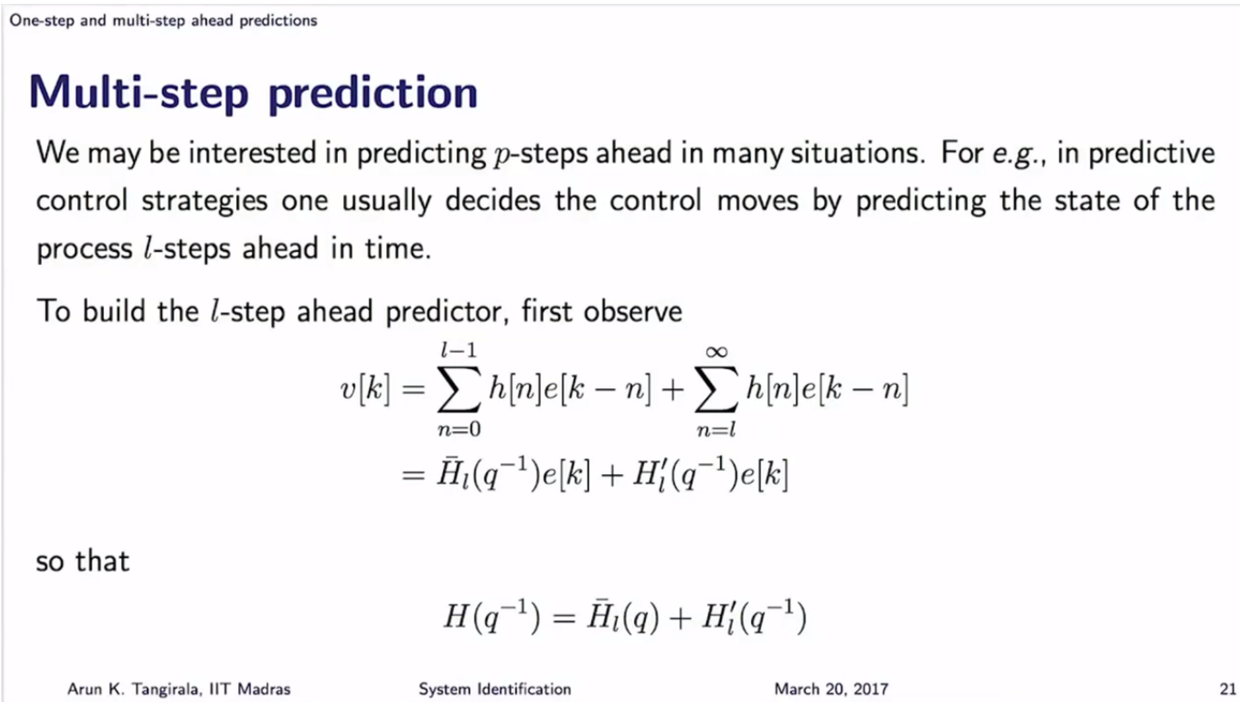
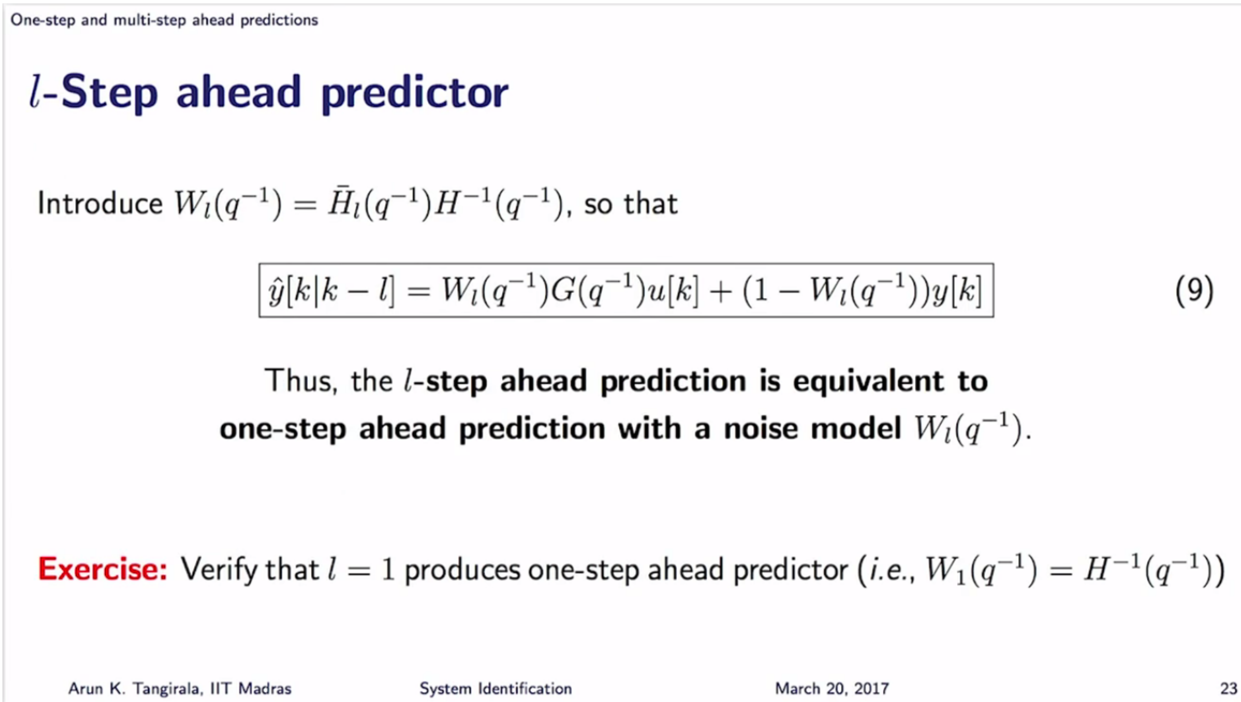
So let's look at infinite step ahead prediction. So without showing this slide let me ask you when l goes to infinity we have spoken about l going to 1. When l goes to infinity what happens to Wl? What is the expression for Wl? Why are we interested in infinite step ahead prediction?

Simulation. We want to know the simulation properties of a model. What happens when l goes to infinity? What happens to Wl? What is the answer? Why? I am sorry.

Student: [00:01:06]



Arun K. Tangirala: H-bar. Sorry. Here. Right. If you go back. When l is infinity is there anything in the second term? There is nothing left. That means I have not given anything. So H-bar becomes H itself.



As a result W becomes 1. correct. As l goes to infinity W goes to 1. So what happens to my l step ahead prediction? What form does it take?

The second term manages, right. So I am left only with the first term Gu. Correct? But what is Gu? Gu is your OE model one step ahead prediction. You see that. earlier we wrote one step ahead prediction for OE model. What was that?

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

If you go back we said for the OE model it is simply Gu. That's one step ahead prediction.

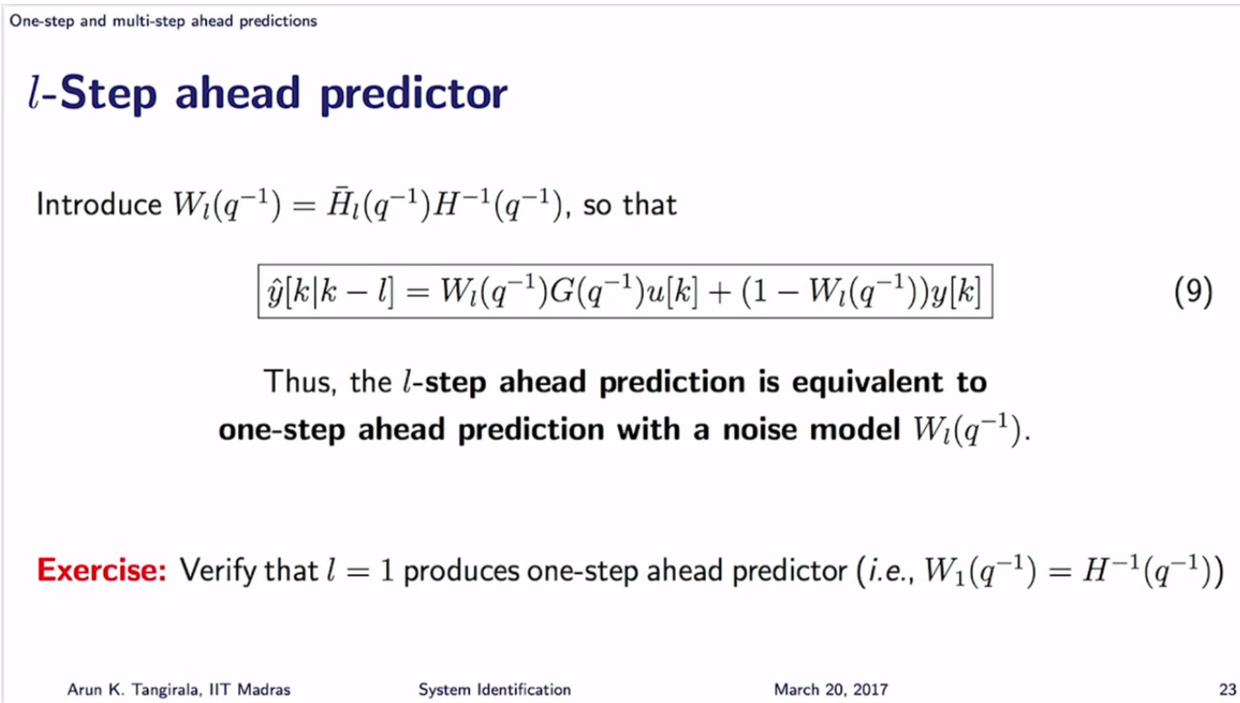
Predictors for parametric models

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

So one step ahead prediction of an OE model is in general the infinite step ahead prediction of any other model. Why is that? Because of the equivalence that we talked about just now that l-step ahead prediction is equivalent to one step ahead prediction with a noise model W inverse. Not Wl.



So which means that the infinite step ahead prediction of any model is the one step ahead prediction with a noise model of Wl inverse. But Wl is one when l goes to infinity. Correct?

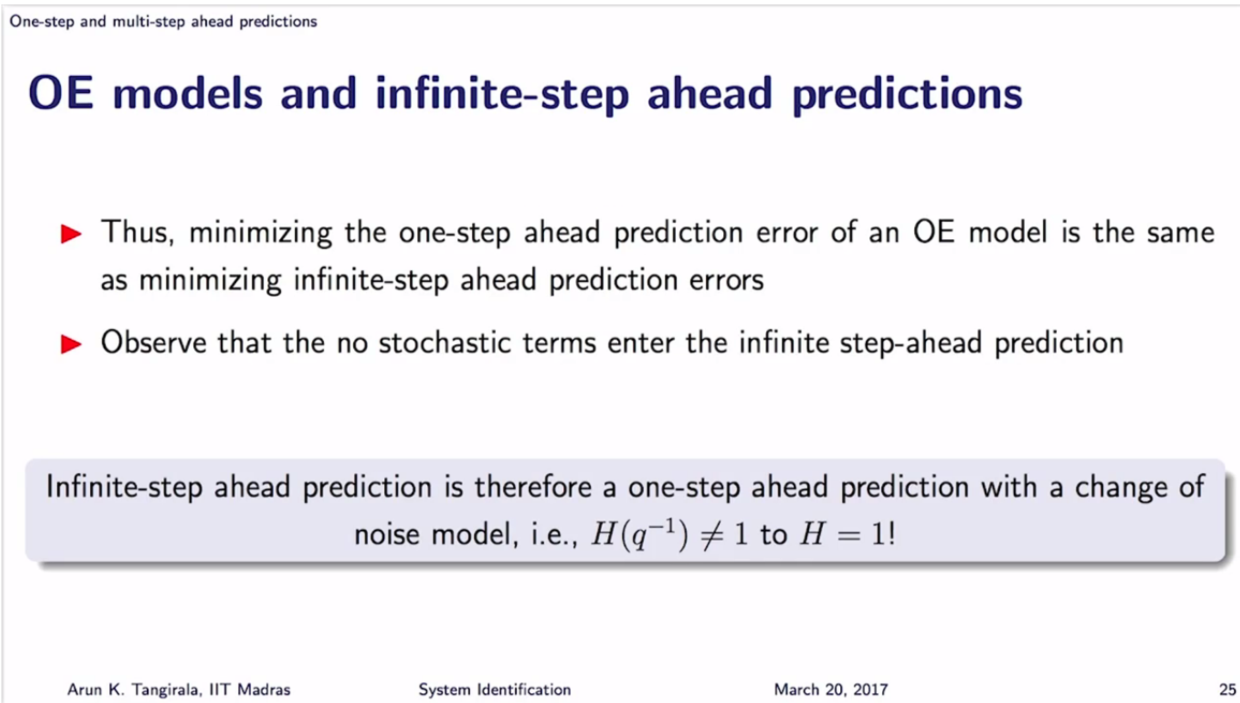
So what will be the noise model for that 1? and what is the structure with the noise model 1 OE model? So with the OE model whenever you make one step ahead prediction it is the same as making an infinite step ahead prediction witi any other noise model.

So what is the big deal about it? Well that gives a very nice property to the OE model structure which is that suppose I am estimating G assuming OE model structure, and minimizing one step ahead prediction errors I am actually minimizing infinite step ahead prediction error. Okay. I repeat. If I am estimating G with an OE model structure, and I am minimizing one step ahead prediction error that is equivalent to minimizing infinite step ahead prediction error with any other model. Is it good or bad? It is good because the test of any model is infinite step ahead prediction. That means its ability to predict on its own without taking any measurements on a plan. It should be able to emulate the plan. That is why when you are testing a model one step predictions will look always good. You try to making infinite step ahead predictions that is the true test of a model.

OE model structure when you have chosen OE model structure and when you are optimizing G you are actually doing the best thing. Whereas with any other model structure when you are minimizing one step ahead prediction error you are actually minimizing one step ahead prediction error only. Whereas to the OE model although you are minimizing one step ahead prediction error you are actually minimizing infinite step ahead prediction.

That gives the beautiful nature to choosing an OE model. We have been saying start off with an OE model, get the best estimate of G then look at the residuals and so on. Right?

So this explains a lot about the different model structures. Is that clear? So the output error model has a certain nice appeal to it. So when l goes to infinity you are left with this and this is nothing but the one step ahead prediction of the OE model.



And I have just written here what I have just stated. Okay. minimizing the one step ahead prediction error of an OE model is the same as minimizing infinite step ahead prediction errors. Whereas with any other model structure you take ARX for example. Leave aside FIR. You take ARX you will get one estimate if you minimize one step ahead prediction error. You will get another estimate if you minimize two step ahead prediction error and so on because expression keeps changing. Whereas with OE it doesn't matter whether you minimize one step ahead, two step ahead, infinite step ahead, the prediction error expression doesn't change. It is infinite step ahead prediction error. So that is why output error model or as it is known in literature the measurement error models are very good. Okay.

One-step and multi-step ahead predictions

Predictor filter representations

The one-step ahead prediction in (6) can be re-written as

$$\boxed{\hat{y}[k|k-1] = W_u(q^{-1})u[k] + W_y(q^{-1})y[k] = \mathbf{W}(q^{-1})\mathbf{z}[k]} \quad (11)$$

where $W_u(.)$ and $W_y(.)$ are *input* and *output* predictor-filters respectively,

$$\mathbf{W}(q^{-1}) = \begin{bmatrix} W_u(q^{-1}) & W_y(q^{-1}) \end{bmatrix}^T; \quad \mathbf{z}[k] = \begin{bmatrix} u[k] & y[k] \end{bmatrix}^T \quad (12)$$

$$W_u(q^{-1}) = H^{-1}(q^{-1})G(q^{-1}), \;\; W_y(q^{-1}) = (1 - H^{-1}(q^{-1})) \quad (13)$$

Arun K. Tangirala, IIT Madras   System Identification   March 20, 2017   27

Alright. So we have spoken about simulation. I will close this discussion with what is known as a predictor filter representation. Until now we have spoken about models, that means G and H. Let us take a different perspective. If I were to look it from prediction view point, what am I actually doing if I look at the prediction expression that we have derived. Let us even just take the one step ahead prediction. What is it doing? It says there is this input U and then there is this output Y they are being processed by these filters H inverse G and 1-H inverse and then both are adding up right here to produce Y hat of K given K-1. So it's as if -so what am I given sys ID I am given input, output data, right. This is what I am given. What did you want ultimately prediction. Once I give you a model you are going to use it for prediction. Why not directly compute a prediction itself?

So here we write in terms of H inverse G and 1-H inverse but I might as well erase this, and call them Wu some filter, and Wy some other filter. And instead of optimizing G and H I can optimize Wu and Wy. I can do that. until now we have been saying choose G, choose H. And once you obtain G and H construct a one step ahead prediction. So that's a two step procedure for constructing the prediction.

But now I can go back and say look do I have to really choose G and H? Why can't I just choose these two filters and optimize them? You can do that. There is nothing wrong in doing that. It is as good as choosing G and H. This kind of a representation is called prediction filter representation where instead of choosing plant and noise models you will be choosing instead predictor filters. That's all. The predictor filter representation has other uses which we will talk about a bit later when we talk of identifyability. We will revisit identifyability at that time we may have to talk of equality of models that's where this predictor filter representation will come handy.

Predictor filter representations (cont.)

The representation in (11) is known as the **predictor filter representation** of an LTI system.

Further,

$$\mathbf{T} = \begin{bmatrix} G & H \end{bmatrix}^T \longleftrightarrow \mathbf{W} = \begin{bmatrix} W_u(q^{-1}) & W_y(q^{-1}) \end{bmatrix}^T \tag{14}$$

i.e., the mapping between the models and predictor filters is unique.

In other words specifying G and H is as good as specifying W_u and W_y.

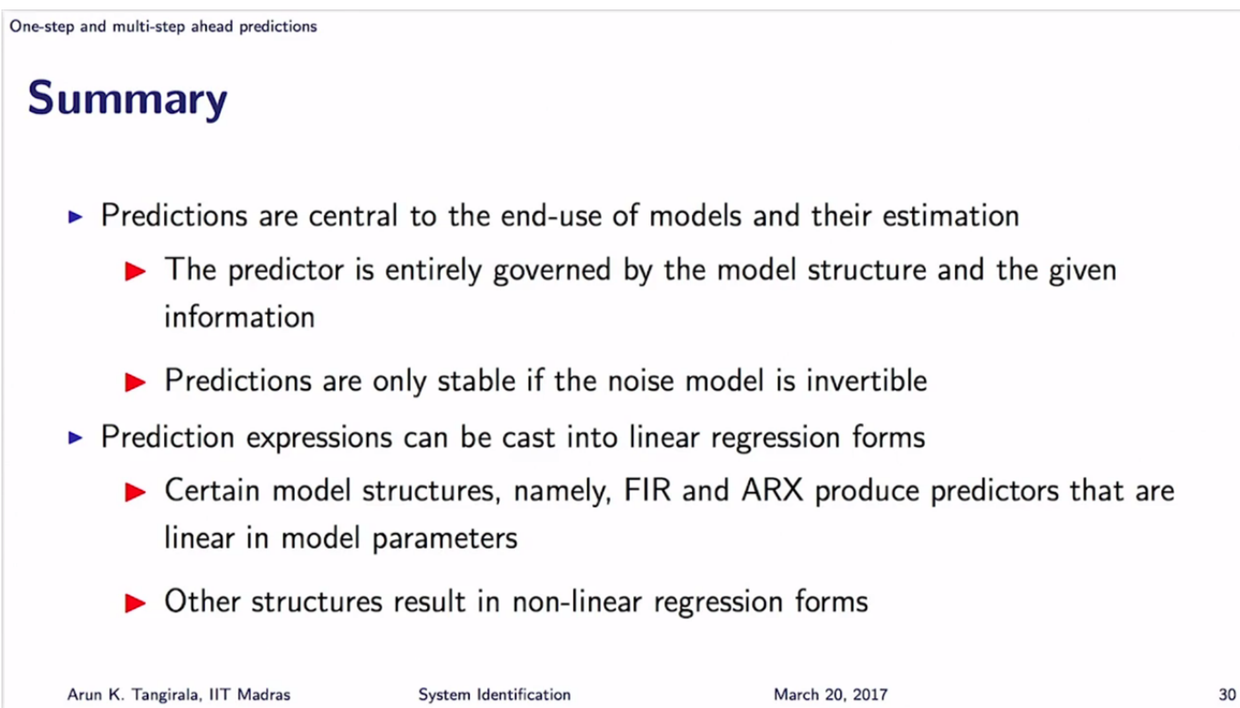
But one thing to remember is there is one-on-one mapping. If you choose G and H then there exists only one Wy and Wu. If you choose Wy and Wu there exists only corresponding one G and H. So the mapping is unique which is reassuring for us that yes instead of choosing G and H I can optimize this way also. Okay.

MATLAB commands

Command	Functionality
`predict`	Compute one-step and multi-step ahead predictions
`forecast`	Predict the model outside the time range of data
`pe`	Compute prediction errors
`compare`	Comparing measurements with p-step ahead predictions
`resid`	Compute residuals (prediction errors) for a given model and data

So let me summarize with a list of the MATLAB commands. In MATLAB there is a command a sys ID toolbox there is command called predict which computes one step and multi step ahead predictions. Check out the syntax and defaults. Then you have forecast. Now the difference between predict and forecast is that predict takes in your model and your test data or whatever data that you give and predicts over the time period of the data. Okay. It pretends as if you are predicting over that period of time only. Forecast takes the model and the data and predicts beyond the time of the data. So suppose I am giving data corresponding to first one hour I give that data and the model. Predict will compute whatever predictions you would have computed online during that one hour. Forecast is different. The forecast routine takes the model, takes the data that has been given and predicts beyond one hour that you can specify how long.

You understand. Forecast looks at beyond. Predict only focuses within the time period that you have given in the data. And then you have Pe, that computes prediction error which is quite useful. That means your epsilons. And then compare allows you to compare the predictions of the model with a measurement. We use all these commands after we have gone through one round of model estimation. Correct? For example we use Pe to generate residuals. And once I have residuals, I do a whiteness test. Of course resid command does that for you in – it combines both these steps. It computes the prediction error and it also plots two things. I have shown you earlier even in the liquid level case study. It generates two plots. One is a cross-co-relation between residuals and inputs which will tell me how good my G is and other is, what is other one? Auto co-relation of the residuals which tells me how good the H is. That's all right. Ultimately, in sys ID you have to remember if you remember anything or not given data add G, H, sigma square E. Okay.



G, H; government hospital. Okay. Anyway. So that is what it is. And then this is just a summary. So that brings us to a close on predictions. The prediction theory is a lot more than what we have learned. I have just tailored the lectures for sys ID. If you read a corresponding chapter in my textbook, I also talk about best linear predictors and so on. So a lot more is there but it suffices for this course for you to know whatever is there in this lecture notes. Correspondingly, there is material in the textbook. You don't have to be aware of best linear predictors and so on. But if you are interested you can read on them in the text. Okay. So given a model you should be able to theoratically write the one step ahead prediction, infinite step ahead prediction and so on. The questions in the assignment will help you do that.

Okay. So I will just spend about 10-15 minutes in introducing estimation, then we will continue our discussion tomorrow.