CH5230: System Identification

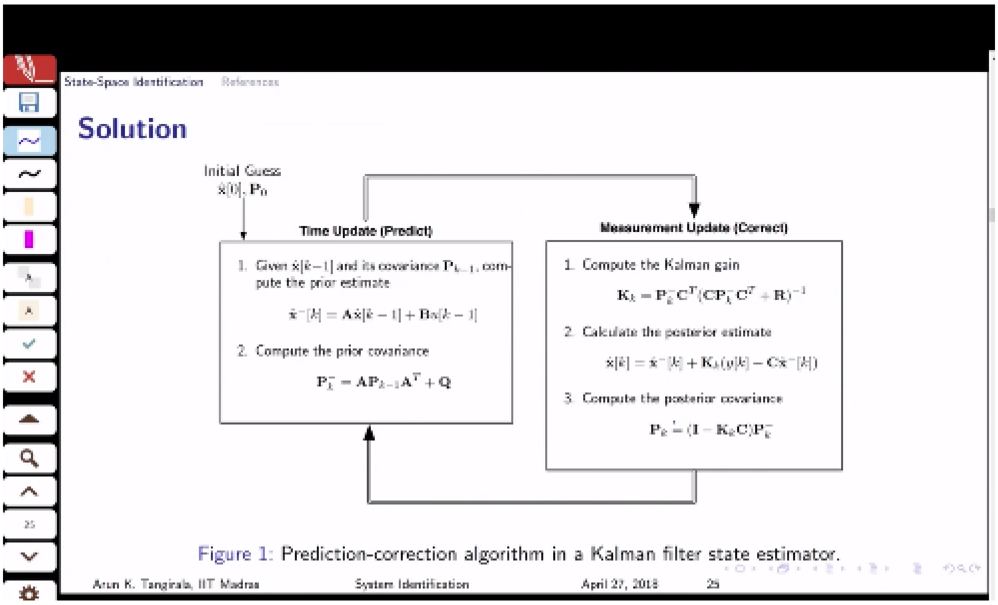
State-Space/Subspace Identification

Part 4

Okay. So welcome back to the lecture on State-Space Identification. In the previous lecture we were introduced to the concepts of observability and Kalman filtering. And we also obtained a preview of what subspace ID algorithms can do for us in this lecture what we are going to do is, we're going to go into the full details of the subspace ID algorithms. However we'll start with the simpler case where we'll assume that the impulse response data is available. And that there is no noise in this data. And then gradually build on that where the target is to reach the scenario in which the data to an arbitrary input and that is the response to an arbitrary input is available and that the data is corrupted with noise. towards the end we will talk a structure State-Space identification as well, very briefly.

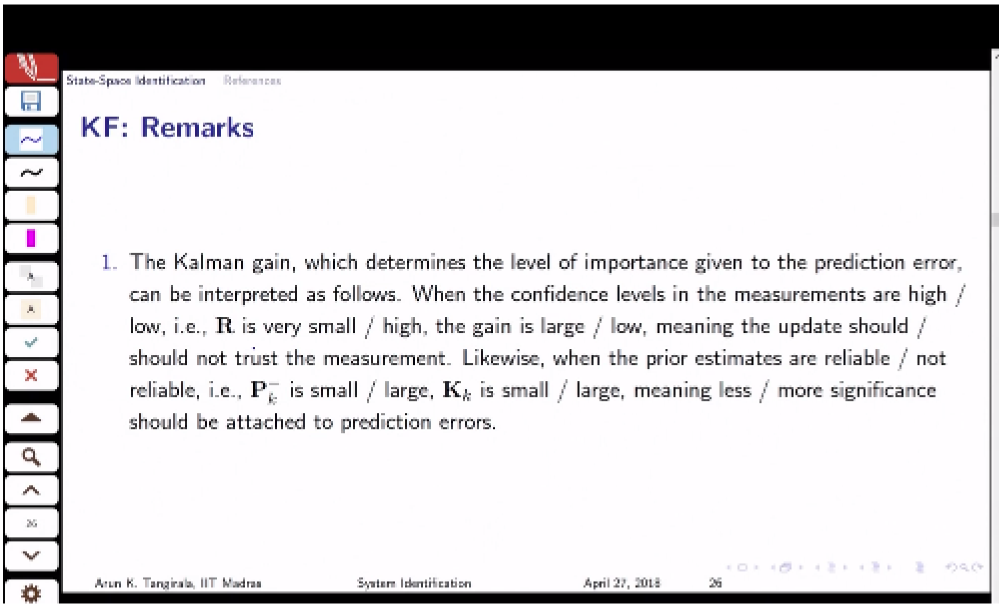
So let's start off by recapping some of the concepts of Kalman filtering as we have learnt Kalman filter is a predictor corrector algorithm. It predicts the next state based on the guests of the state at this instant and then makes a correction which we call as a filtered estimate once the new measurement is available. So this is a schematic that we discussed yesterday in the previous lecture at length.

(Refer Slide Time: 1:42)



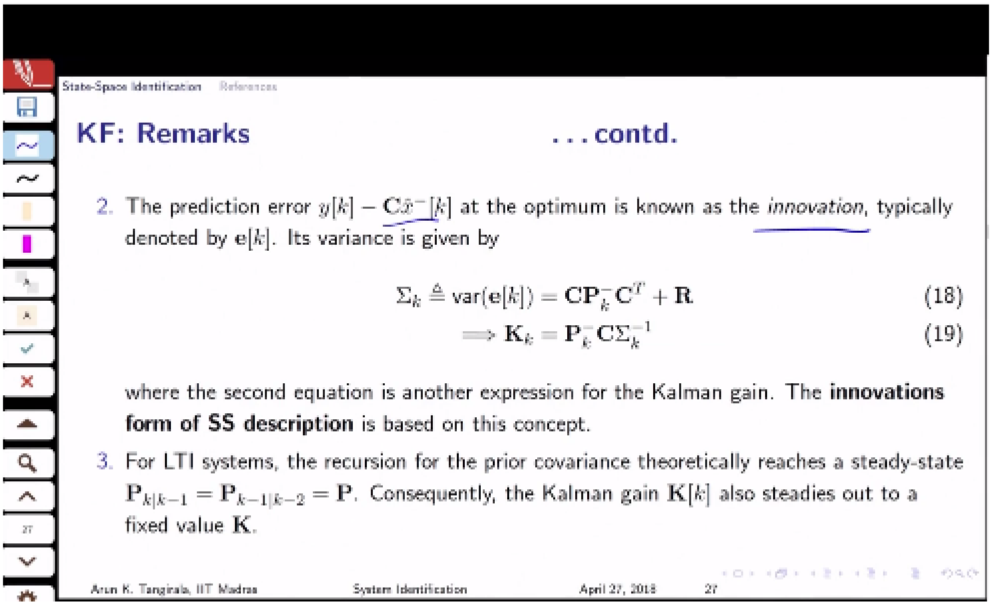
As I had said, the Kalman gain which denotes how much importance has to be given to the correction depends on two things, the error or the confidence that we have in the prior estimate and the error or the confidence that we have in the measurement. So depending on the relative ratio you would have the Kalman gain taking small or large values, right. But always when Kalman gain is small it means that I'm giving less importance to the measurement. And if it is large of course, I'm placing more importance to the measurement that I have.

(Refer Slide Time: 2:21)



And the prediction error which is yk munus C, X hat minus k at the optimum is known as the innovation. This is a term that we have seen earlier also when we were discussing times series modeling. And what this means is that the optimum prediction has been obtained through the C X hat minus K. whatever we have left out is ek. This is a property of the Kalman filter not every filter will give you necessarily this. So that is important and that should be also expected because it's a minimum means square error estimator.

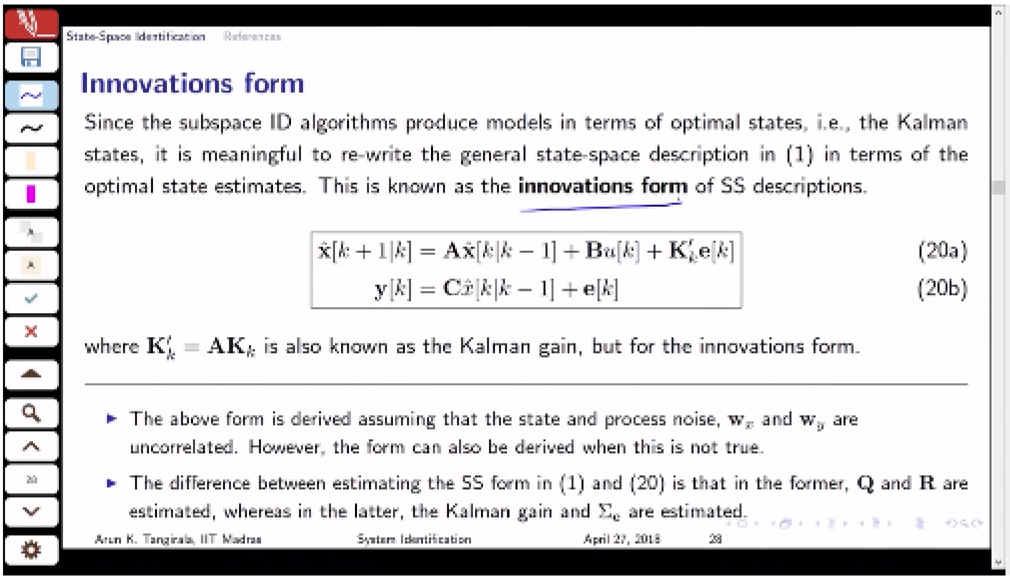
(Refer Slide Time: 3:02)



And we will based on this understanding and this term called innovation we will shortly come across this innovations form of state- space description that I briefly talked about in the previous lecture. Right. And this is the point here of Kalman gains steadying out to a fixed value. Is something that I had broached on in the previous lecture that for LTI systems if you give sufficiently long time then the Kalman gains steadies to a fixed value. So typically what is done is this fixed value itself is determine upfront and implemented right from step1, because it's anyway going to steady out. And they're versions of course where a time varying Kalman filter gain is implemented initially and then the steady one is implement.

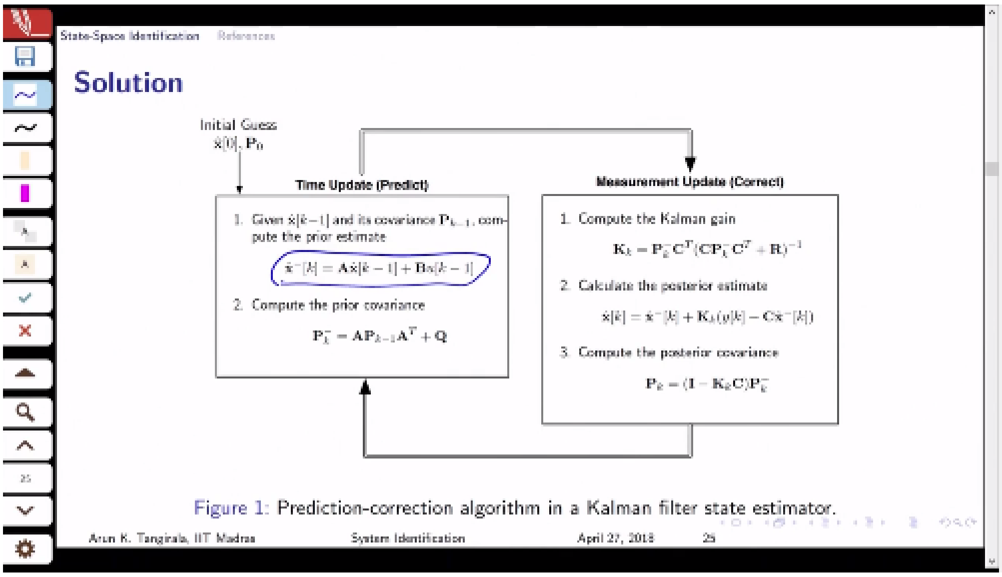
What is important is to remember that this Kalman filter has been derived under certain assumptions which is that the system is LTI. That is a deterministic part is LTI, the stochastic is stationary and that the noise follows Gaussian distribution. If any of this is violated then the Kalman filter can still be implemented but will no longer give us optimal estimates. And of course, given that the Kalman filter was proposed more than 50 years ago. There have been numerous variants and numerous applications as well off the government for that and the variants. So you have for example, if this system is non-linear there is a version called the extended Kalman filter known as the EKF and then there is a version called un-centered Kalman filter to handle non-Gaussianities and non-linearities in the deterministic process. Non-Gaussianities in this stochastic process. Details of which are provided in the textbook. So let's quickly look at what is this Innovations form. And also just wanted to add that there is a worked out example of the Kalman filter in the textbook. I would like you to refer to that example. We'll move on and we'll talk about this innovations form.

(Refer Slide Time: 5:10)



This innovations form of state-space description is an alternative way of describing the general state-space description that we have seen earlier. The general state-space description is something that we have seen in the previous lecture. This is the most general one and these are the covariance matrices. Now we know at least from our discussion on subspace identification algorithms that ultimately when they estimate the state-space model, the states that they have from the model are going to be optimal states, optimally estimated ones. And they are nothing but Kalman states essentially. Therefore it may be nice to write the state-space form in the form of in terms of this optimal estimates itself and that is the motivation for going towards the innovations form. So what you do is, you in order to derive this innovation form, you go to this equation and start asking-- So for example here, you look at this equation.

(Refer Slide Time: 6:21)



This equation in fact, the prediction comes from the state equation. So the starting point for innovations form is the state equation itself. Instead of calling this now at as x hat minus we revert to the more familiar notation where we write x hat of K plus 1. For example, you would write x hat of K plus 1. Given k. As in fact, sorry, here, since everything in terms of k we'll write it in k first. So x hat of k given k minus 1 is a x k minus 1 but we'll replace now with the hats. Although we say here, you know, xk minus 1, it's actually if you observed it's x hat k minus1. So this is k minus 1 given k minus 1, remember. X hat of k, k minus 1 given k minus 1. Plus B u k minus 1. So this forms the state equation for us. In fact, let me now write the equation in terms of x hat k plus 1 given K. So let's write it that way. A x hat, of k given k plus B u k that's it, right. This is what we have. Now x hat k given k has also if you go back there is a filtered estimate--that is there is an updated equation that we have here, where it says, x hat of k given k is x hat of k given k minus 1plus this Kalman gain times minus yk minus cx hat minus k. So let's use that and writ the update equation. So we can write here x hat of k given k is hat of k given k minus 1 plus, what do we have from the update equation? What does it say, here? k times yk minus, so for now I will drop the time varying index there on the Kalman gain, so it's k times yk minus C. What do we have? H hat minus K. Right. And then of course, we have plus B uk. Now with the innovations with the introductions of the innovations we know that this term here. What is this? This is ek. Right. We have already said that the optimal estimates are such that this is white noise.

So we right here a x hat of given k minus 1 plus b uk plus k times. What do you get here? Ak times ek. And I'm tempted to ask is everything okay? Too many k's here. x hat of k, so A times x hat of k given k minus 1 plus B uk plus A k times ek. So that completes the state equation for us. X hat of k plus 1 given K is this. And then what about the output equation? What do we write? How do we write the output equation? Any ideas? It should be pretty state forward, it's there on the screen. The answer is on the screen. Anybody? Y of k equals? Which is c x hat k given k minus 1 plus ek. That's all. Is there a difficulty? Why was there so much silence? So that's your state-space model now where the states are Kalman states or optimally estimated states.

So if you were to work with the state-space model such as this one let us say, A, B and K C are all given. And you are given the variance of ek. And an initial condition is given the states that you will be generating are nothing but the Kalman's states that means optimally estimated states. This is essentially your Innovations form. All right. So we will denote this AK If you want with K prime. So let's get back to the discussion here. So that's what I have done and I have brought back the dependency of Kalman gain on time here. Is a clear? So this is a much more easier form to work with because now this k has a lot of straight interpretation that it is a Kalman k. And by the way, we have derived this assuming that the state and process noise are uncorrelated. Why did they say that? Because the Kalman equations that we have used assume that process and noise are uncorrelated if they are not. Then you can also bring that into the picture. But the essential form of innovation is form-- structure of innovation form will remain more or less the same. The nice thing here is for the output error model. Now let me ask you. For the output error model, what is k prime? That is if you are looking at multivariate output error model.

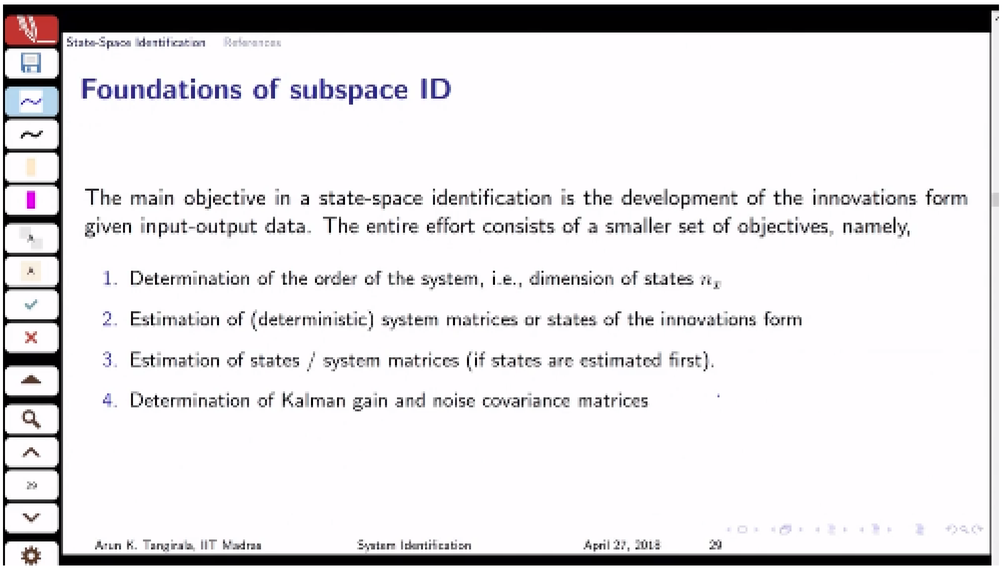
And you want to write an innovations form for that structure model. What would k prime? What does output error model assume? What is assumption in that structure that there is about the process and measurement noise. What does output error-- working with output and model tantamount to? Okay, then which noise is unpredictable?

So you're still in the transfer function? If you recall the previous lecture we talked, we interpreted the output error model in the state-space domain, right? What did we say about the assumptions on process and measurement noise? You must recall the previous lecture. Correct. So you have said wx is 0, so there is no process noise, right? I know the institute calendar said, 26th was the last working day in the previous calendar, apparently I have updated it. So we are still expected to keep your mind alert and switched on. What is the confusion Purna? So in the output error model the process noise is absent which means for an output error model k prime is 0. Right. So for an OE model k prime is 0. Of course, I can put a subscript here it doesn't matter. So that you the process noise is missing and the measurement noise is only present in. The white noise is present only in the measurement equation.

And likewise you can derive the structure of K for other structure-- the innovations form for other model structures such as ARMAX, BJ and so on and even ARX. Now as far as the difference between estimating the states-space form in the original one that we have seen and the innovation form here is that the objects of interest are different but they're modeling the same process. In the original state equation we were interested in A, B, C, D, Q and R. Right. Noise covariance process and measurement noises. So here of course, D is missing you can always had a D, that we have anyway we have omitted assuming one delay.

In the innovations form the objects of interest are A, B, C, D and what else? K. So I'm going to estimate the Kalman gain also from data now that is where we are heading. Until now, what we have learnt is that the Kalman gain is computed given the model and the noise covariance matrices. But now we have only data. We don't have any model. So we are supposed to estimate A, B, C, D the Kalman gain and the noise variance of ek, covariance of ek. ek remember it is a vector of white noise equation. So we have not escaped the burden of identification but we have made it simpler and more interpretable. That's all. All right. And remember that the Kalman gain itself implicitly the function of a, b, c, d and the original noise covariances q and r. But the dependency is so complicated that we don't even want to highlight it. We want to consider k itself as a separate. entity to be our or separate quantity to be estimated. Right. So you should remember that Kalman gain is a function of A, B, C, Q, and R. And then there is also a prior covariance matrix. Bu this is essentially the dependency of Kalman again. It's too complicated. So we just don't worry about it. We just say that it is yet another thing to be identified.

(Refer Slide Time: 17:23)



Okay, so with that innovations form in mind we will march ahead toward subspace identification. Already we have learned the concepts of observerability, controllability and now the Kalman filter and innovations form. The main objective of course, is subspace identification I should say, not necessarily state-space. Is the development of the innovations form given input, output data. And the entire effort consists of, you can say, you can break down the objectives into smaller ones. The first thing is to determine the order. How many states? What is the dimension A? If I'm saying, if I'm interested in estimating A, B, C, D first I should know what the dimensions are. And those dimensions are dictated by the order. I already know the number of outputs and inputs, right. And then the estimation of A, B, C, D depending on the method that you choose to work with the method may either first estimate the system matrices and then the states or vice versa.

There is one class of methods that estimates the states and then estimates A, B, C, D. That is also done. And then finally the determination of Kalman gain and noise covariance matrices. So what I intend to do in the rest of this lecture is to talk about these two mainly and I will not be spending time on how to determine the Kalman gain and noise covariance matrices because they are less complicated and relatively more straightforward ones these two items have resolved. So I would like you to refer to the textbook for the determination of Kalman gain and noise covariance matrices. They are much more simpler. Because once you get for example, A, B, C, D then it's all a matter of generating the residuals and then finding the Kalman gain a noise covariance matrices. Okay. O

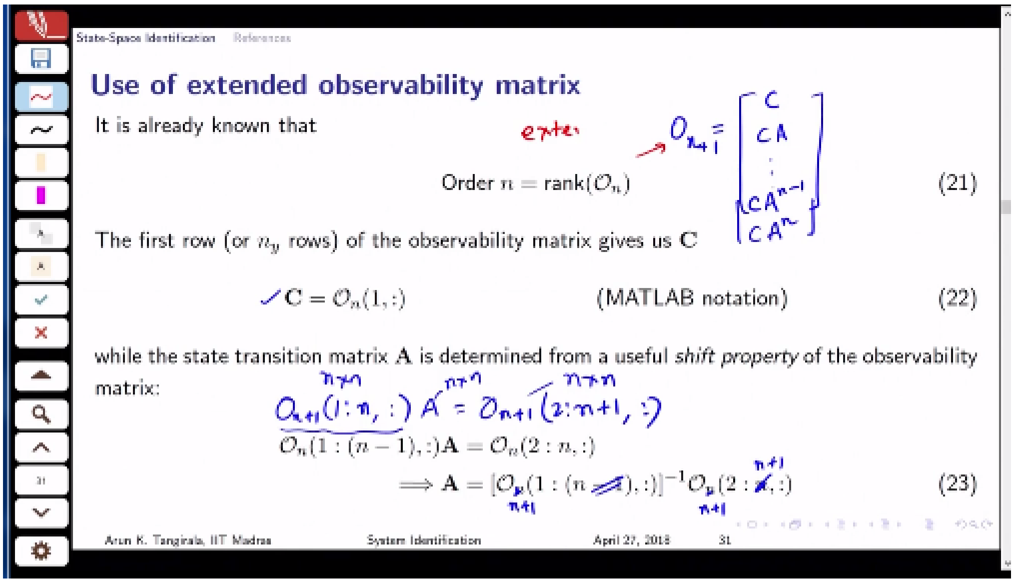
The core of any method whether-- broadly speaking there are three classes of methods that you encounter in subspace ID. One is called the N4SID. And the second one is called the MOESP and then the third one is the canonical variant analysis called the CVA. Although the names are different the core idea is the same which is first you construct the so-called extended observerability matrix and get the order. After that their tracks are different. One method estimates the state-space matrices focuses on getting your A, B, C, D and K. First A, B, C, D and then the states and so on. The other method says, no I estimate states first and then estimate A, B, C, D and so on depending on the situation. And then there is also another variation that we learn later on which will lead to the CVA. But as I said, the core is all the same.

So let's understand the core problem how it is solved. As I said, the key step in subspace ID estimation of extended observability matrix. So we'll spend some time on understanding this extended observability matrix and then again also go through what is known as a realization based method put together everything and then we'll understand fully that how this subspace ID algorithms work. Now in that process at some point in time we learn how A and C are estimated from the extended observability matrix. Remember the extended observability matrix will give us two things. One, its ranks will give me the order. Two, I'll be able to estimate the state matrices A and C. So what is this extended observability matrix? The extended observability matrix is nothing but a bloated regular observability matrix. For example, what if you recall from the previous lecture, we have learnt that the observability matrix subscript n that is its dimension is M by N. Its rank is nothing but the order of the system. That is the number of states and the minimal realization. And if you recall. Let me write the expression here for the observability matrix. What is the observability matrix that we had Expression C, CA up to CA to the n minus 1. So we can straightaway see that the first row, assume that right now we are dealing with the single input, single output systems. If you're dealing with multi output then the first n y rows. For simplicity assume that we have single output. So the first row of ON is your C, so straightaway I know, how to get, if I'm given observability matrix. Let us first talk about that and then we'll talk about the extended observability matrix. And then how do I determine A? What you do is, you use this shift property. What is this shift property telling me? ON in fact, this is not how it should be written. But because. So fine, this is fine. So what do we do here? We take the first n minus one rows and all the columns. How many columns does On has? O N has n columns. So you should see the dimensionally there is a consistency here. So here, you have an n minus 1 by n. All right. And then this is an n by n. And what about On 2, 2 n, ; what is the dimension of that? It is n minus 1 by n. So dimensionally things are compatible and you simply use this shift property.

Is there any issue? Sorry. There is a problem. That is why I said, there is an issue. So what you do is in fact-- right. No, no. So what we can do is, we don't need to use this thing. We can construct now an extended observability matrix now. So for example, we can go to O n plus1. So if I stack this up, If I stack this up then the next element would be C A to n. Right. That is why earlier there was an issue that I had pointed out in this equation, we'll correct that. So let's now work out things again. So you say, O n plus 1 times 1 off 1 to n. The number of columns in O n plus 1 is still n only. That doesn't change, remember. In O n plus 1, what has changed as compared to O n, the number of rows has increased by 1. If you're dealing with a multi output system, the number of rows increases by n y. So we will write the correct version of this equation here. 2 2 n plus 1, ;. Now this O n plus 1, I have written, I have used MATLAB notation here. It's not the standard notation. So here 1:n, 1, 2 n , : this dimension of this matrix is n by n. And the dimension of this matrix is n by n. So matrix compatibility is obey and now I can write. So instead of n here, I should have n plus 1 and there's no n minus 1. And likewise, this goes to n plus 1. That's the correction that has to be made. So this O n plus 1 that we have constructed for the purpose of estimating a is an example of the extended observability matrix. Why do we call it extended? Because if the system is of order n then O n is full rank.

O n plus 1 is of Frank rank, right. Anyway the maximum rank possible for O n plus 1 is n only, because you have only n columns. But you have more rows than what you can actually-- what you are supposed to fill for full row rank. So the column rank is always n, that's a max that you can get. And because for any matrix column rank and row rank are identical. If you wanted full row rank you should have only constructed O n, but we have constructed O n plus 1. So this O n plus 1 is nothing but the extended an example of the extended observability matrix.

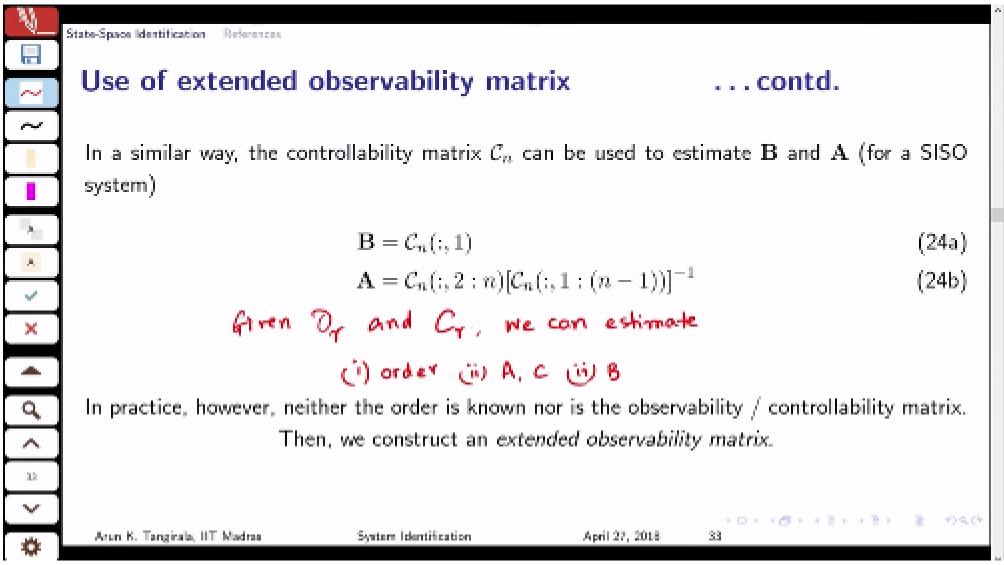
(Refer Slide Time: 26:53)



It's an example, there are when you talk of extended observability matrix it can be extended up to any point. Now in a similar way the controllability matrix and you may want to make the same corrections that we made here. So you want to make here C n plus 1 this should be n plus 1 and this should be n and this is n plus 1. So in a similar manner you can think of an extended controllability matrix as well. Although normally in all this subspace identification algorithms, the extended observability matrix is the one that appears predominantly. You very rarely see, the appearance of the controllability matrix except in the early algorithms by Ho and Kalman which we will discuss briefly. Right.

So the reality is that I do not know. I do not have the observability matrix with me or the controllability matrix with me. What I have is data. If I'm given the model, I can construct this or if I'm given observability matrix somehow then I can extract three things. What we have learnt is, given the observability and controllability matrices, I can extract so given O n. Let us now replace O n with O r, r is some dimension we don't know. And C r which is the controllability matrix of some dimension. I can estimate, we can estimate. What are the things that we can infer? Order, that's the most important thing, right. I can estimate the order. Tow A and C and Three, D from controllability matrix. D anyway you can estimate. That is something that you should understand in the states-space identification. The major challenge in the state-space identification is identification of A, B, N, C. Identifying D is very easy. Why? Because it is the first impulsive response coefficient. Remember, that it is a feat through term, right.

(Refer Slide Time: 29:21)



It represents a feat through them. If G of 0 is 0 that means there is a unit delay and D0. If G0 is not 0 then that means that it's D itself.

And D can always be estimated uniquely by and large the major challenge in subspace state-space identification is that A,B,C are unknowns and X's are also unknowns. That is what makes the problem challenging. So coming back to the discussion that we are in. We need to somehow get these extended observability matrix or the extended controllability matrix that this area, once I get that then things become very easy. So here is our extended observability matrix. And the rank of O r is nx. Although I have written here C, CA, CA to the r minus 1. In practice I wouldn't know the-- So here when I write it, it appears that there are n x columns for O r, correct? Which is equal to n which is true. Always the number of columns in O r will be the number of columns in C. But the number of states is not necessarily equal to that of a minimal realization is not necessarily equal to the number of columns of O r when you from data. When you construct this from model, yes, then it will tally. Provided C, A, are all drawn from the minimal realization. In practice, we will only attempt to get this and from here we first determine the rank and then from the rank we know that then the number of columns that are actually present in C of a minimal realization. So that is procedure. The first is you obtain O r and then you compute the rank. Once you obtain the rank then you construct a lower order approximation. The rank r version of O r. So that the number of columns of the rank are approximation of O r or version of O r. We'll exactly equal to the number of columns of C of the minimal realization. Anyway we learn all of that. Likewise, one constructs an extended observability matrix and so on. Fine. So the goal here is to get this extended observability matrix.