NPTEL

NPTEL ONLINE COURSE

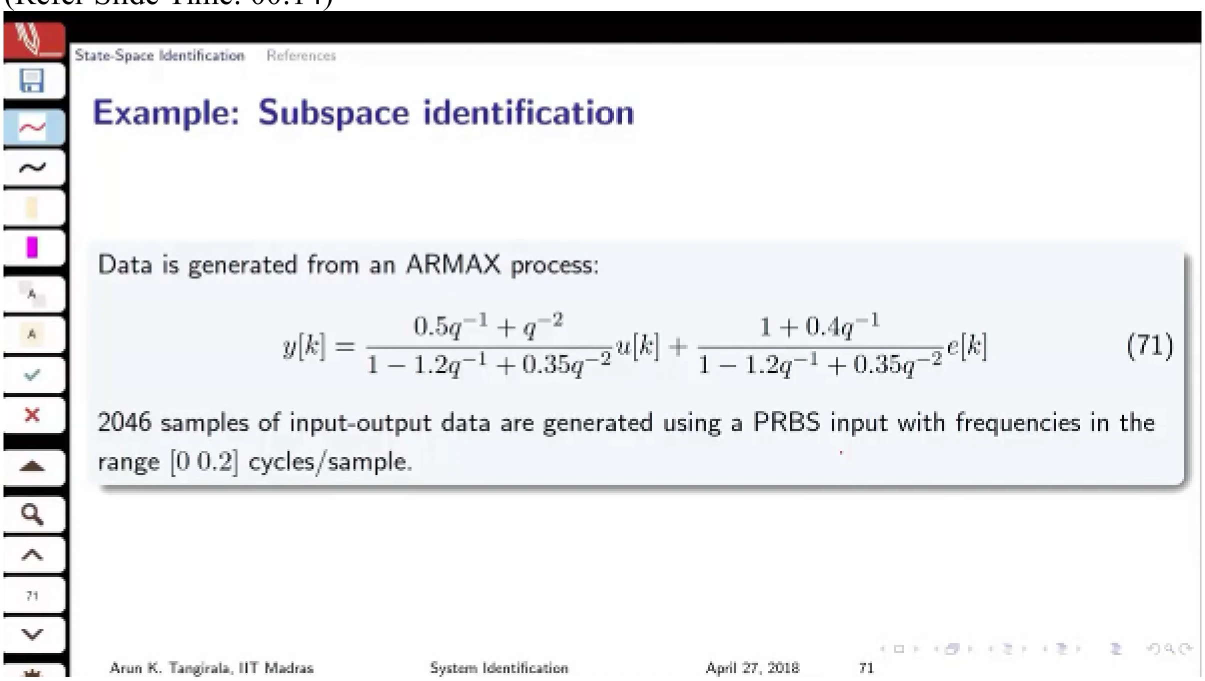
CH5230: SYSTEM IDENTIFICATION

STATE-SPACE/SUBSPACE IDENTIFICATION
PART 7

ARUN K. TANGIRALA

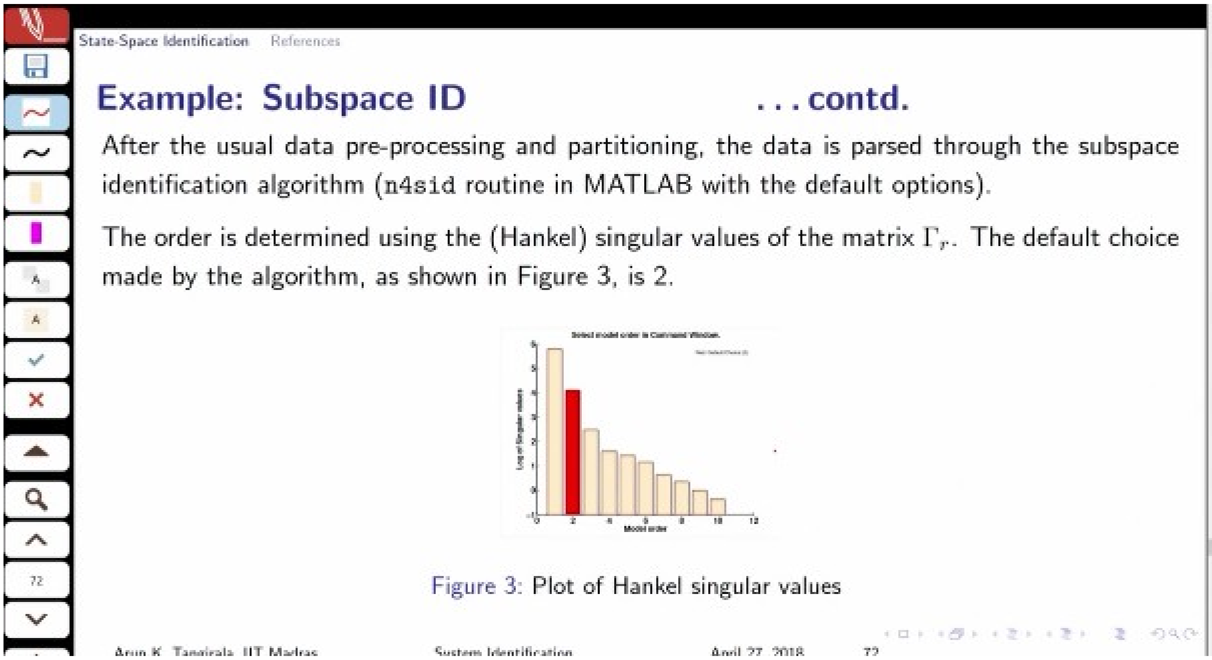
DEPARTMENT OF CHEMICAL ENGINEERING,
IIT MADRAS

(Refer Slide Time: 00:14)



So let's just work out the simple example here, our data is coming out of an ARMAX process, I've used PRBS input, sorry, and then we have done everything, all the steps,

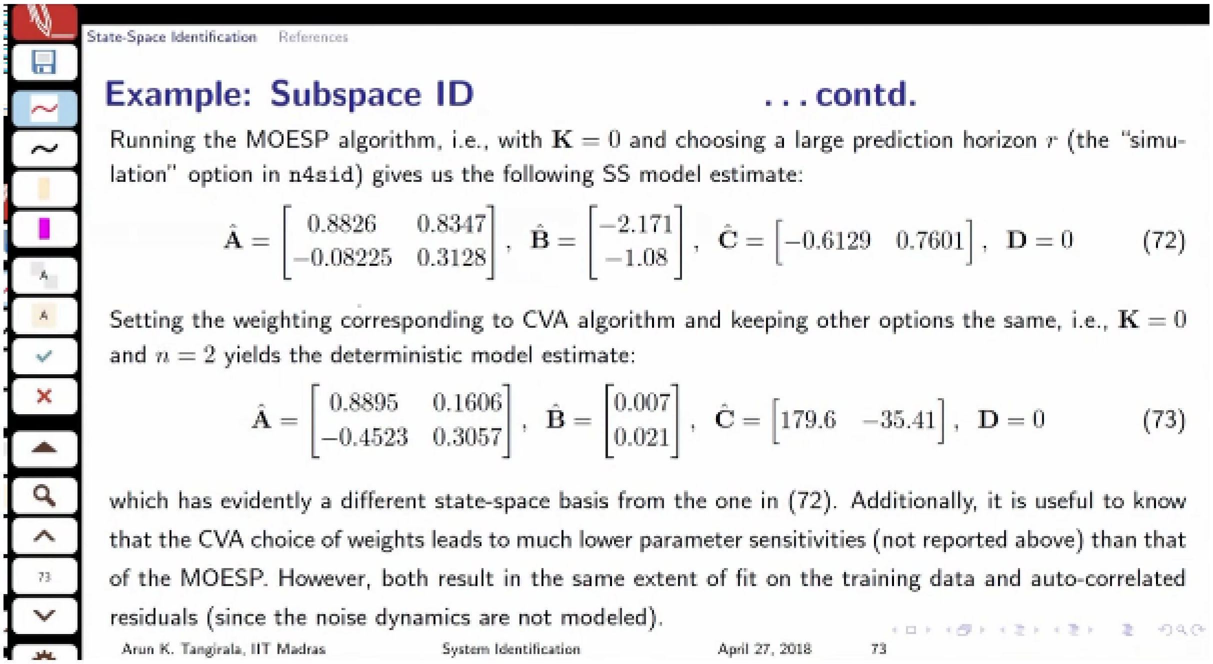
(Refer Slide Time: 00:26)



we are at the stage where we have constructed gamma R, when we are looking at the singular values of this gamma R which is you can say or the log singular values. And you see here again the same story, because you have noise, there is no clear cut picture of, sorry, what is the noise of gamma R.

Now you can use your AIC, BIC, CIC whatever information criteria you have, you pick the order, you figure out what has to be done and so on, MATLAB does that for you and highlights, gives you the winner of the elections here, although we're doing selections it's more like elections, and it says this is my recommendation, you pick that recommendation, but you should not stop at that, what should you do? Okay, that already it has done for you, residual analysis, you should do a residual analysis on this and then confirm that indeed this choice is good, I think we should do a residual analysis and also the elected politician and so on, then we will know whether the politician is adequate or not, or over fit and over fit no chance, only under fit all the time,

(Refer Slide Time: 01:44)

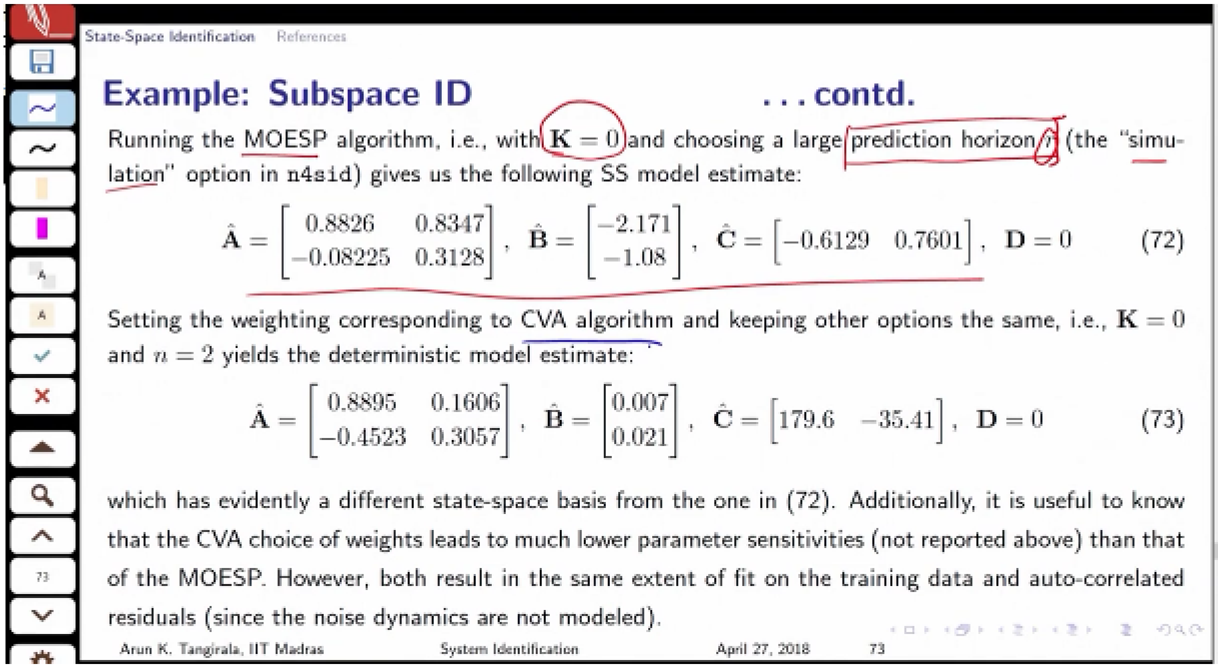


so anyway, so for this example I run the N4SID algorithm with the MOESP option, the moment you turn off the disturbances it is amounting to fitting an OE model, right, when you turn off K = 0, so remember in the input output identification, what do we do? We first fit an OE model that is one of the ways of fitting an input and output model.

Likewise here we are first fitting an multivariable or you can say state space version of the OE model by you can specify that in the MATLAB script, the MATLAB script tells you how to do that, you can turn off the disturbance by setting the disturbance to none, and then you choose a large prediction horizon R, I have not used this term before, but this R that we were choosing is called the prediction horizon, those of you who are familiar with MPC you must be familiar with this terminology, right, don't think S is a control horizon and so on.

And you can actually choose the so called simulation option N4SID, to choose a large prediction horizon, what you're essentially saying is that I want to minimize the infinite step ahead prediction errors, you can think of it that way, we have not talked of the prediction error interpretation of this algorithms yet, but those are available in the text book, you obtain these estimates.

(Refer Slide Time: 03:19)

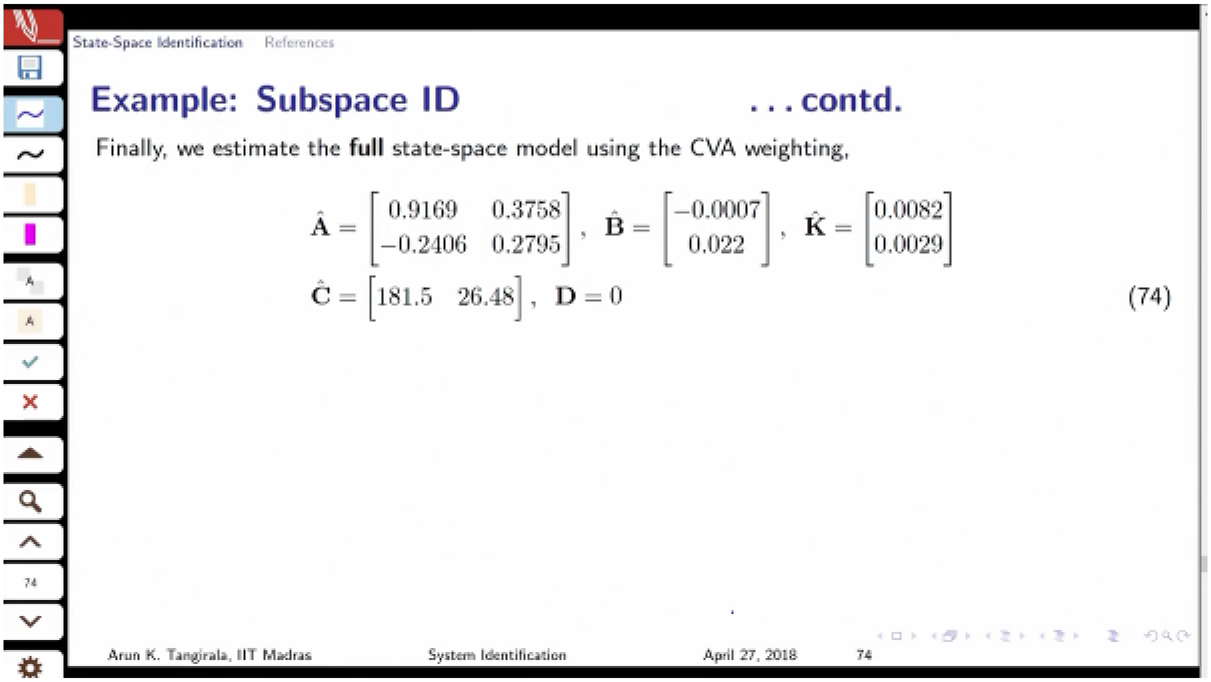


Now you turn to CVA option N4SID, the same routine you turn to CVA option and again estimate an output error model you get this A, B, C, D, are they looking identical? They're quite different from each other, the order is the same but the realization is different, okay, the truth is one, the realizations are many as always, which has evidently different state space basis right, but as I said that the CVA choice of weights leads to much lower parameter sensitivities.

Now remember here we are not reporting errors in the estimates, why is that? Why I am not reporting errors on the estimates of the state space matrices, there is not identifiable, correct, that is why you should not seek for estimation errors here and so on, if you want errors estimation then you'll have to be either fit a structured state space model or convert this to transfer function or convert this to canonical form, and identifiable canonical form then the error calculations make sense, right.

So as far as the extent of fit is concerned on the training data they are the same, and we don't look at the autocorrelation of the residuals,

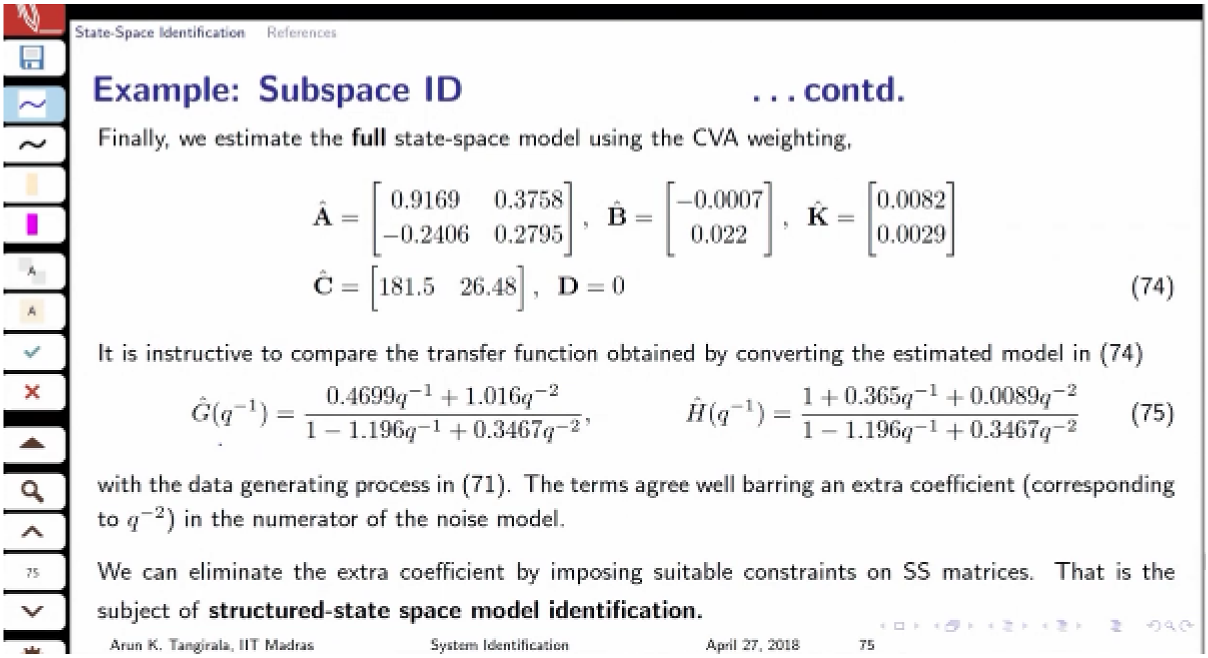
(Refer Slide Time: 04:33)



because we have not modeled it remember we are fitting in OE model, now we can estimate the full state space model using the CVA weighting because we know that CVA weighting is good and you obtain this A, B, C you also obtained the Kalman gain.

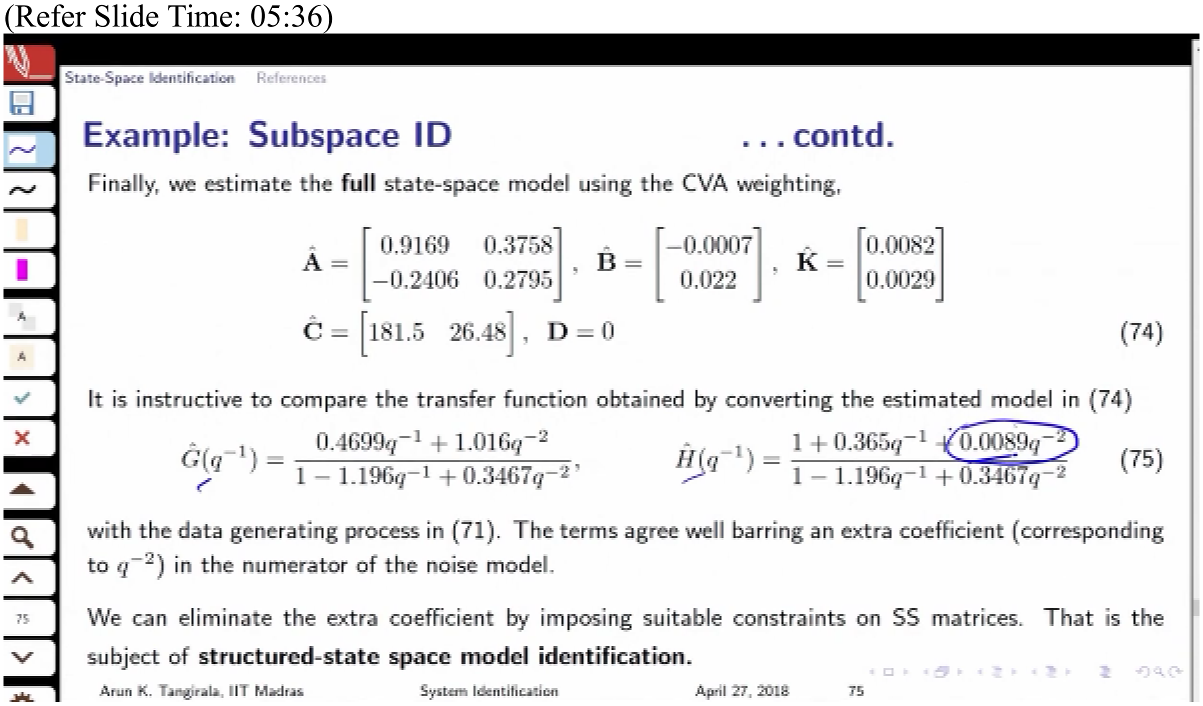
What have we simulated by the way, what is the data generating process? ARMAX, so what you should do is, I'm sorry, you should go back and write a state innovations form version of the ARMAX model, there is an worked out example in the text, going from ARMAX to innovations form of state space description, and compare these results with the true ones, obviously they'll be different except for the D part, rather than doing that you convert this to transfer function G and H, so this is G hat, this is H hat,

(Refer Slide Time: 05:30)



what was our original G and H? This is our G and H,

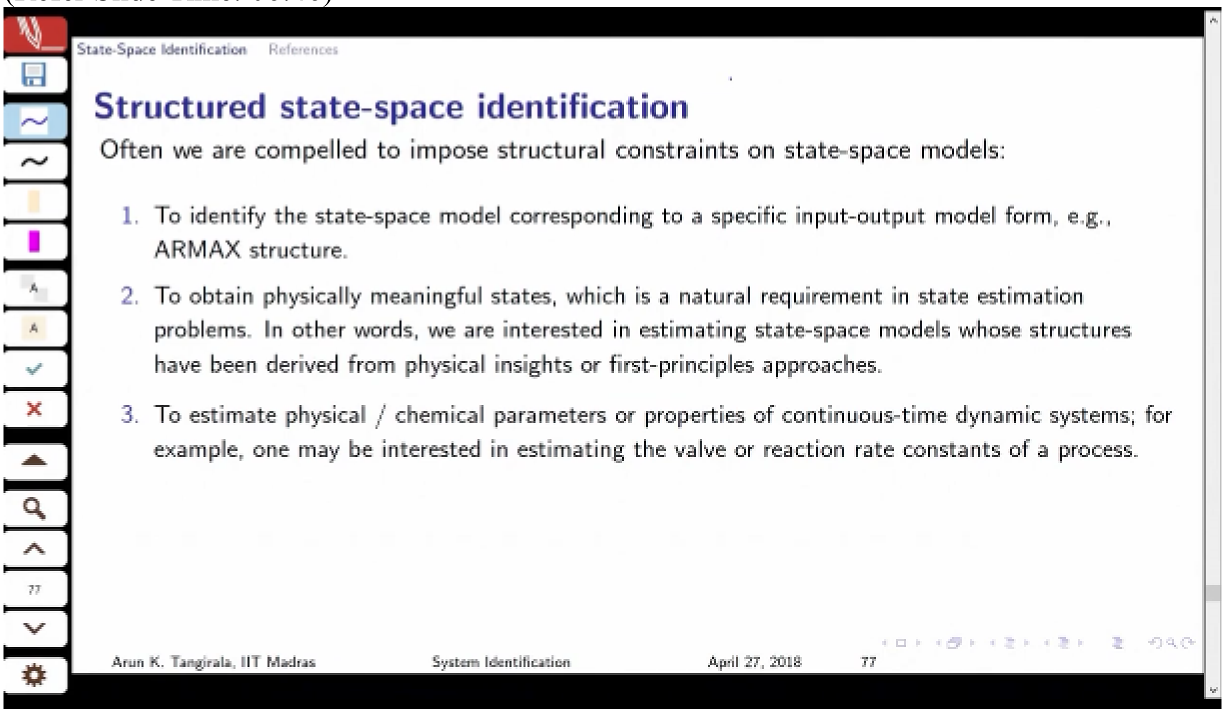
(Refer Slide Time: 05:36)



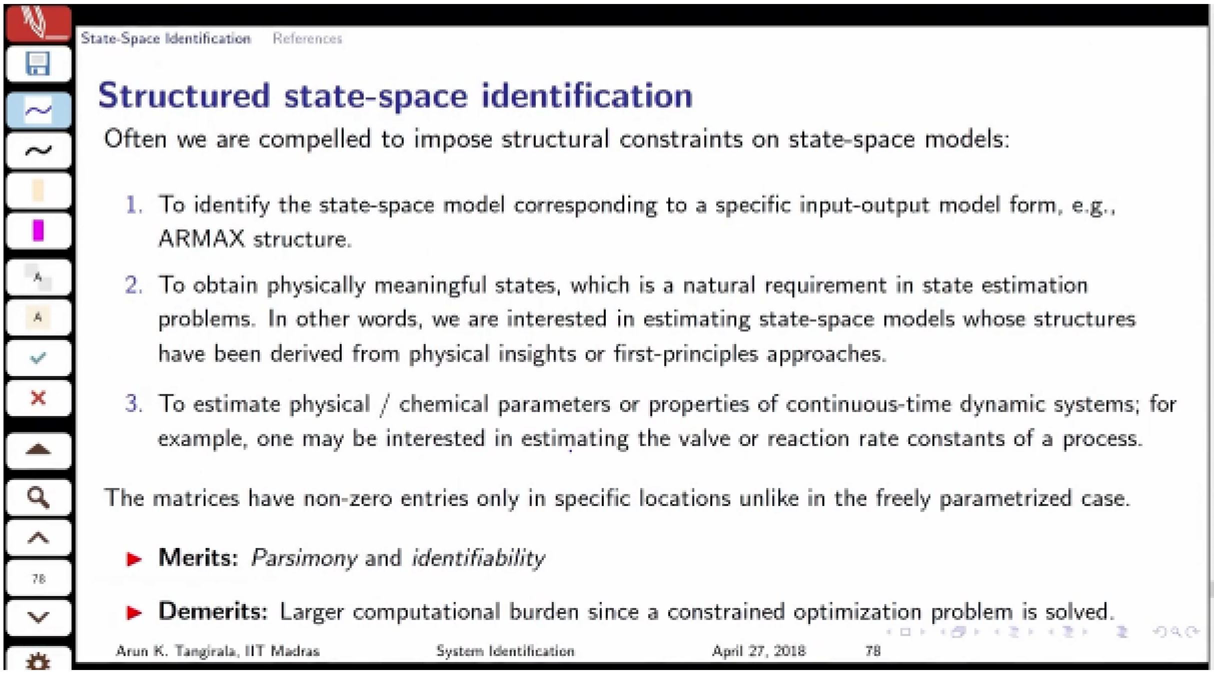
right, sorry, so what do you notice as the difference? Sorry, this additional coefficient here, right, this is the one that we have an issue with, so how do we figure out now? That this should be thrown out or not, you will have the compute the errors on this, but I don't know how to compute the errors on that, if I have come by a state space route, either I have to do bootstrapping or I have to do something to get, to figure out if this is significant or not, let's say I did that, when I want to force fit, this kind of a structure, but I still want to use the straight space route.

Did you see I can always use the input output route, but suppose I want to estimate a state space model such that a model of a certain structure is obtained, then I have to turn to structured state space identification, so this is the MATLAB code I'm not going to spend time on that you can go over that.

So we'll just spend 5 minutes on structured state space identification,
(Refer Slide Time: 06:46)



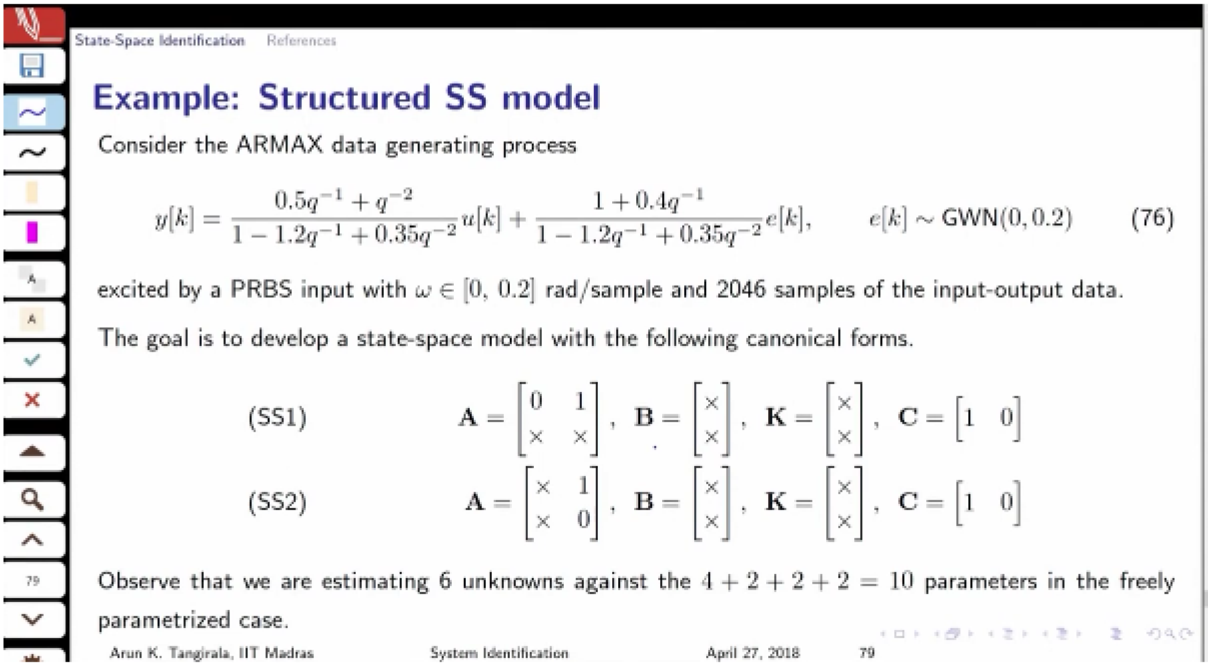
so here is your structured state space identification, details, essentially what you are going to do is you want to identify a state space model corresponding to the specific input output model form or so the objectives can be one of these in structured state space identification, or to obtain physically meaningful states, remember the states that we obtained in subspace through subspace ideal algorithms they may not have any meaning at all, physical meaning at all, they are just states, but suppose I want to obtain physically meaningful states then I have to impose a structure on the state space model that is commensurate with the first principal's model or sometimes you may want to estimate physical or chemical parameters or some properties of continuous time dynamic systems, again here you will have to impose the certain structure, the entries of which will be the parameters of interest and so on,
(Refer Slide Time: 07:47)



so that the basic motivation for structured state space identification is one that you can get some meaningful states, if your structure is motivated from first principals or your structure can be motivated from parsimony, that means if you are working with canonical forms, so that could be another motivation, the main point is that you may be guaranteed identifiability, but what do you lose? You pay a price, you can’t use a subspace ideal algorithms anymore, because nowhere in the algorithm that we discussed we could impose constraints unless you do a constrained SVD, that is the point.

So therefore there is going to be larger computational burden now, because one would be solving a constraint optimization problem that is the issue here, so let me just show you by means of an example,

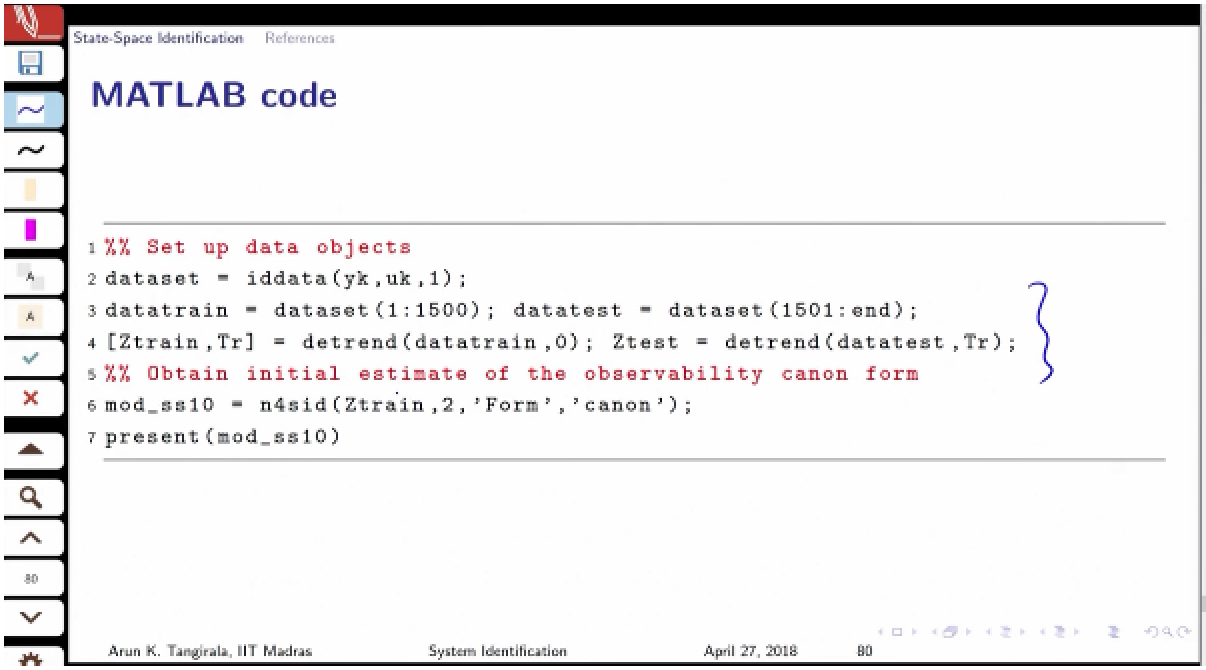
(Refer Slide Time: 08:40)



so all you do is now you impose the structure, rewrite the input output relation, now in terms of the entries of the state space model and run at pem, that is what essentially is done, so here is an ARMAX process, and I'm just showing you an example where we are interested either in developing this kind of a structured state space model or this kind of a structured state space model.

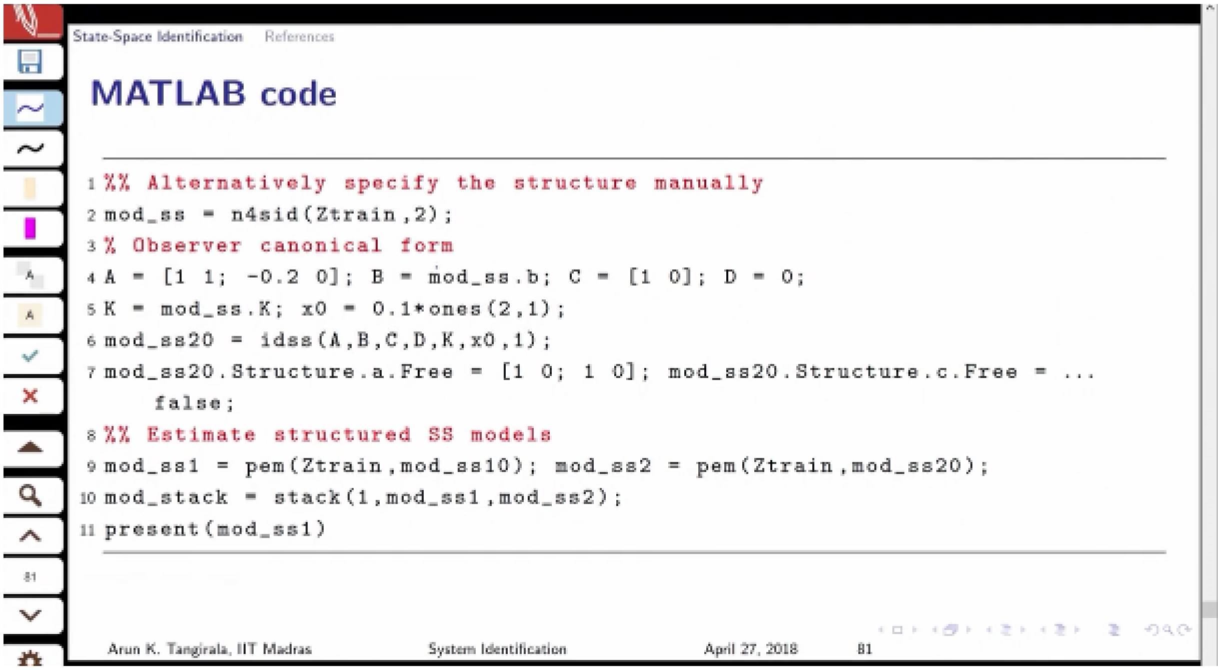
All in all how many unknowns are we estimating here? Here we have 2, 4, 6, again 2, 4, 6, so total 6 as against 10 parameters if you did not impose any structure, so if you were estimating a black box state space model there will be 4 in A, 2 in B, 2 in K, and 2 in C, so you will have 4 + 6, 10 parameters, of course here the benefit that you obtain is not so much, but in the last dimensional system, higher order system working with structured state space model gives you a lot more benefit with respect to parsimony.

So what we do is we first generate the data here,
(Refer Slide Time: 09:53)



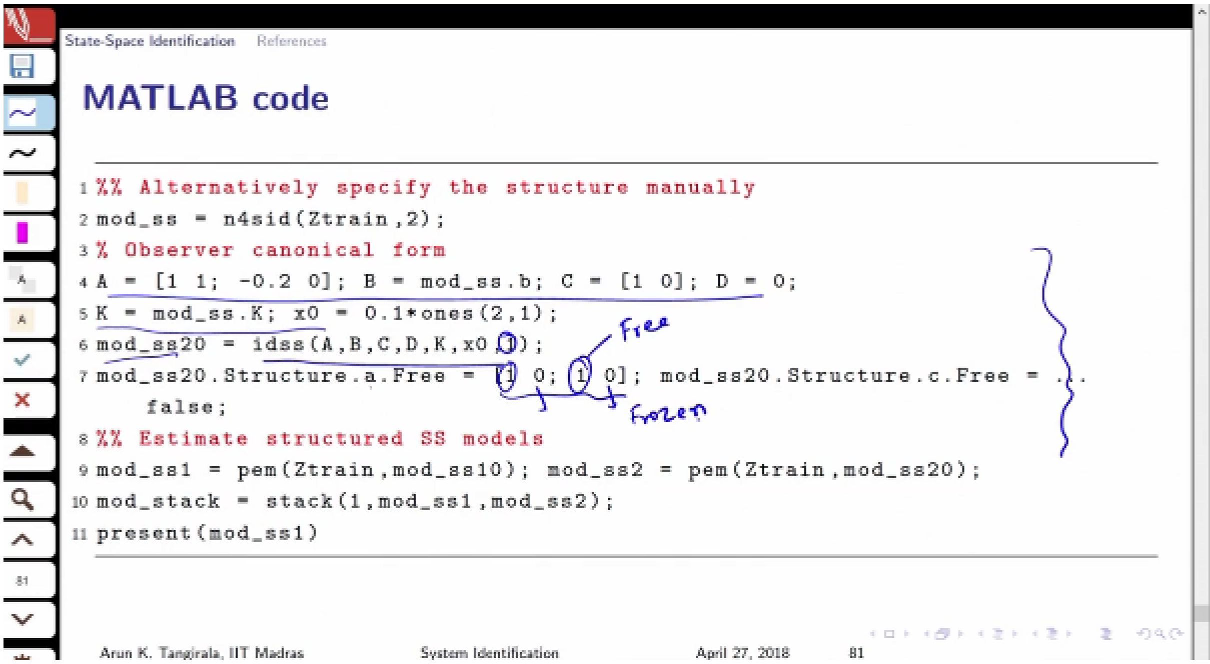
obtain initial estimate of the observability canonical form, so we do that and N4SID routine also has that ability to get you certain structured state space models, not any structured state space models, some standards structured state space models N4SID can do it for you, one of the standard thing is that canonical form, there is also a diagonal canonical form that it can get which is called model in mechanical engineering, when you are working with diagonal canonical forms you have poles on the diagonal, so it’s called a model form, but it is recommended that you work with this observability canonical form and N4SID gets that for you.

On the other hand if you want to impose some other structure, then N4SID is not the correct way, there is an ID grey to do that for you.
(Refer Slide Time: 10:41)



And what we do here is now the other route is as I said you specify manually what the structure is, this route is much better because you have much more freedom in estimating any structured state space form, okay, but here we are still in the discrete time, we are still identifying structured state space form where the structure is dictated by the discrete time state space model. In the next example that I show you the structure would be dictated by continuous time state space model form, so here already we have estimated the observability canonical form, the other form that we want to estimate is remember we were trying to estimate 2 forms here, N4SID gives you only one of them, the other one you had to specify, so the procedure is you first create some A, B, C, D, right, and then also K and X naught these are in fact some of the guesses for these are being derived from the ones that you've obtained already, and then you put together this A, B, C, D and K and so on, this 1 is the noise covariance into a state space model and then you tell the algorithm which of this entries are free to vary.

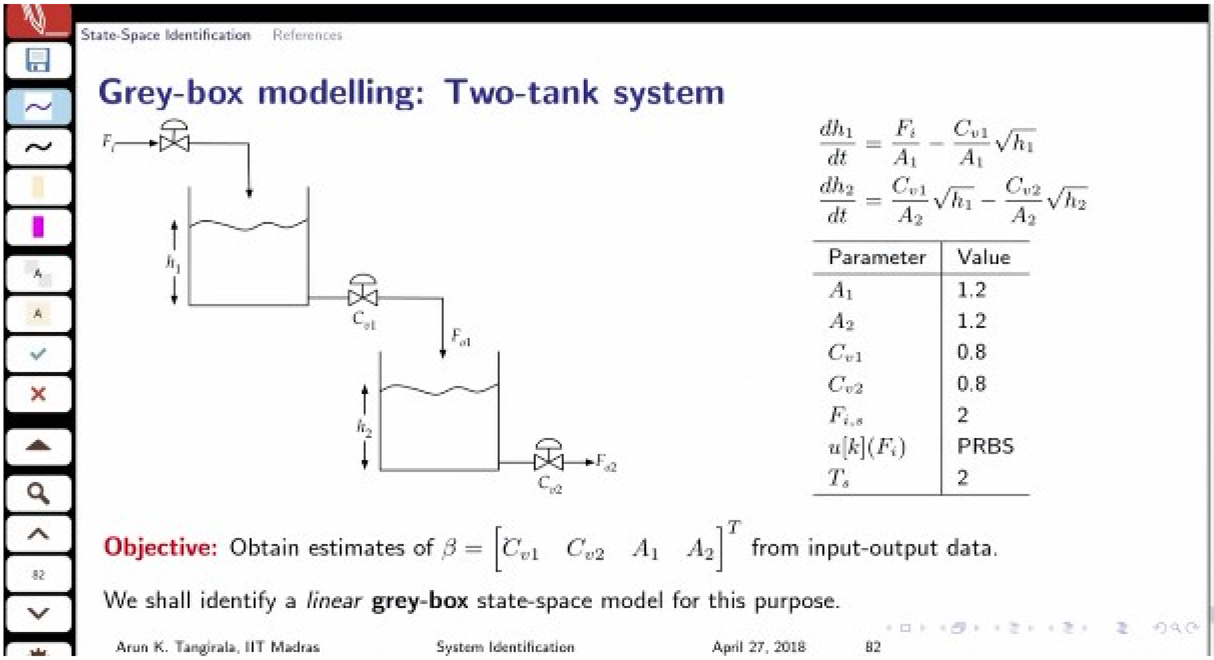
So by saying that structure.a.free for this mod, by writing it this way what does it mean? It means that these are allowed to vary, these are free, and by indicating 0 here these are frozen,
(Refer Slide Time: 12:27)



so the algorithm is not supposed to play around with those two, then you can say that C is 1 0, nothing should be changed in it, so you can, if you think that the entire matrix is frozen you can do it with one shot saying false or you can go ahead with this index as well, whichever they, so you are saying C is not available for tuning, you are not supposed to play around with C, and then you pass it on to pem, what do you pass on to pem? The training data and two things that is mod SS10 is your original, that is whatever you have given here, you obtained as a canonical form, you're just simply giving it to pem so that you get a better estimate, remember N4SID will give you some canonical form, how does it obtained this canonical form? It estimates a black box state space form and then converts that to a canonical form for you, it is not really directly estimating the canonical form, we want to estimate two canonical forms, right, so this one we have generated an initial guess through N4SID, then you have to further refine that through pem, which then optimally directly estimates that canonical form.

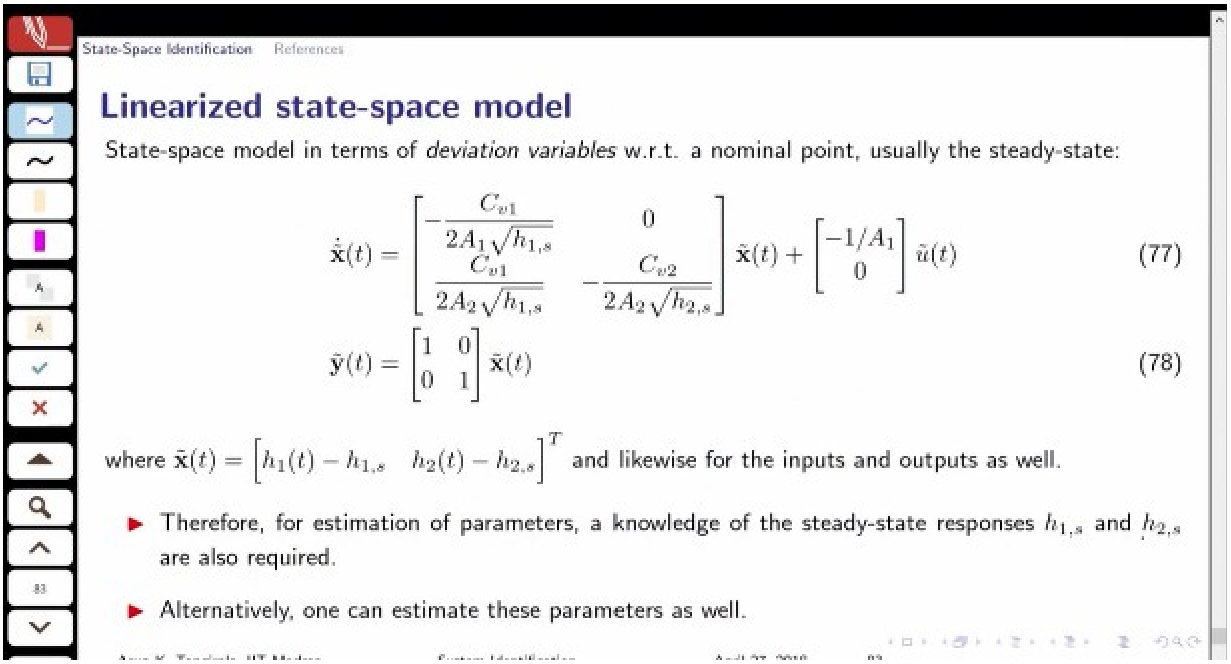
And the other case is this one, remember we are solving constrained optimization problems which are generally nonlinear optimization problems therefore we will have to provide initial guesses, and that is what we have done until now.

And now I have estimates of this two state space models you can stack them and present both of them simultaneously if you want or you can only present one of them, so I'm not reporting that the results are given, but I've just went over the MATLAB code, you will be able to see that you will get descent estimates of this canonical forms.
(Refer Slide Time: 14:19)



So we'll close the discussion on the structured state space modeling with grey box modeling, so under structured state space modeling there are two branches, one is the structure, where the structure is imposed on the discrete time state space model directly, and is not driven necessarily by physical considerations, the other option is that the structure is driven by physical considerations and the constraint is mostly on the continuous time state space model, so here is an example of two tank system after all this course begin with the single tank it should at least enclosure and with two tanks so that you feel that you've learned something, okay, so we consider two tank process here, these are the two differential equations that we have, and these are the parameters that I've used for simulation, and we want to obtain estimates of the two valve coefficients as well as the cross sectional areas of the two tanks, you may say why do I want to estimate cross sectional areas I should know them, but sometimes I may not know, right, so let us say I want to estimate this four, from the input output data, how do I do that? For that we shall pursue the grey box approach, so what you do is step 1, you write the state space model linearized state space model,

(Refer Slide Time: 15:45)



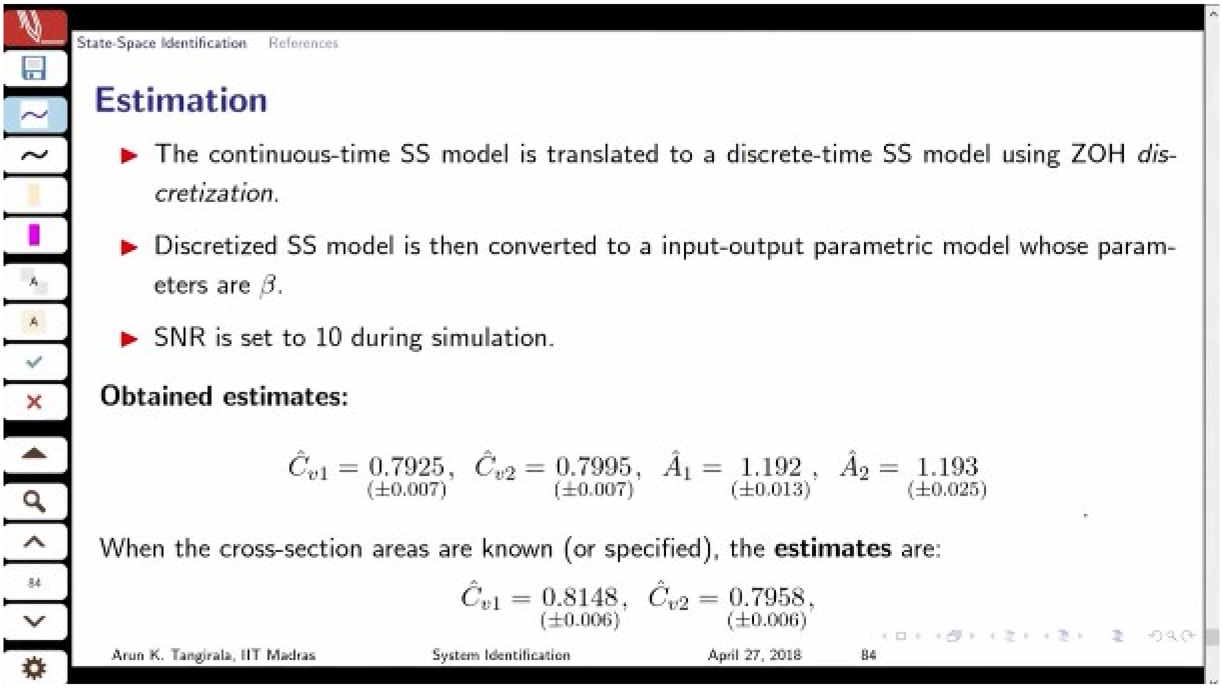
I'm only showing you, how to do the linear grey box identification, you can also the nonlinear grey box identification as well, where the continuous time state space model is a nonlinear one, again example of that isn't there in the text, but what we have done is we have linearized the equations that are given here, these are nonlinear state equations, this is H1 is X1, H2 is X2.

Now my states have a physical meaning and I want that physical meaning to be preserved in the state space identification that is a goal here, only when I do that I can estimate CV1 and so on, so what is happening here is I've linearized and this matrix A is continuous time state space matrix, and it is a function of the parameters that I am estimating.

So now the focus is not on estimating the entries of A per say, the focus is on estimating beta which in turn influence the entries of A, and of course here also you have B being influenced by one of the parameters, C is fixed, and as usual because we are linearizing, we are working with deviation variables.

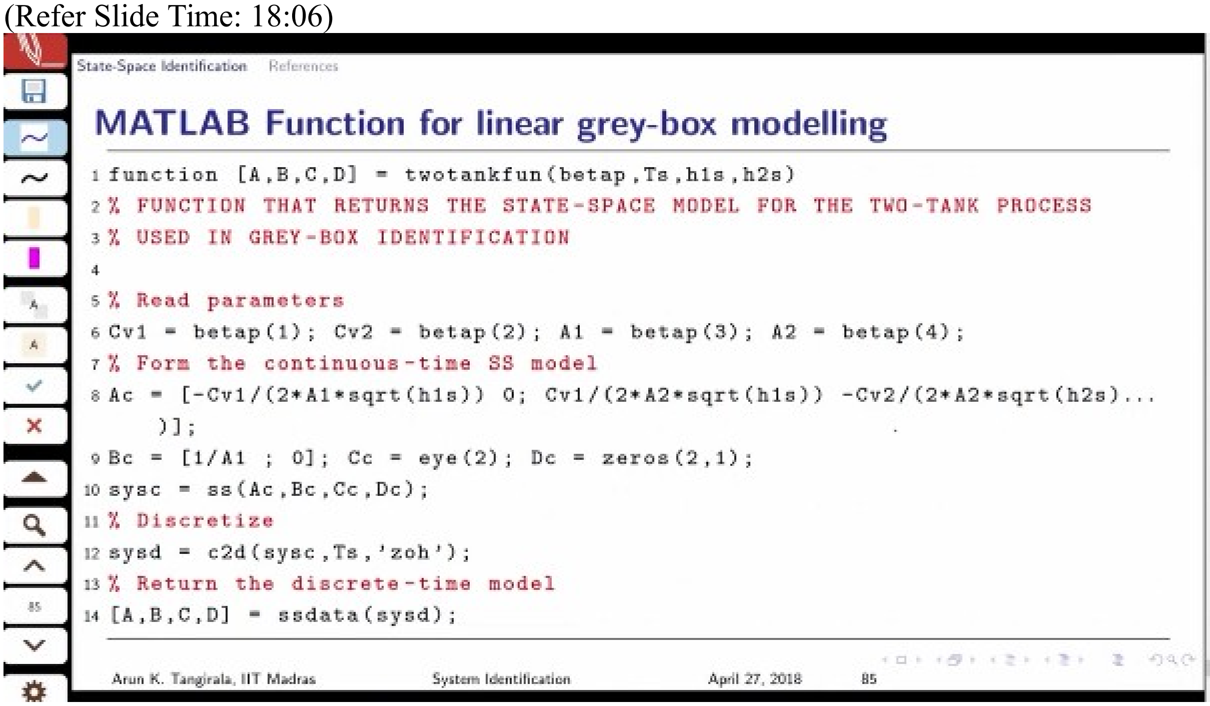
So now we can also by the way estimate these study state values as well which is very interesting, but we will assume that they are available for now, so what we do is step 1 linearize,

(Refer Slide Time: 17:13)



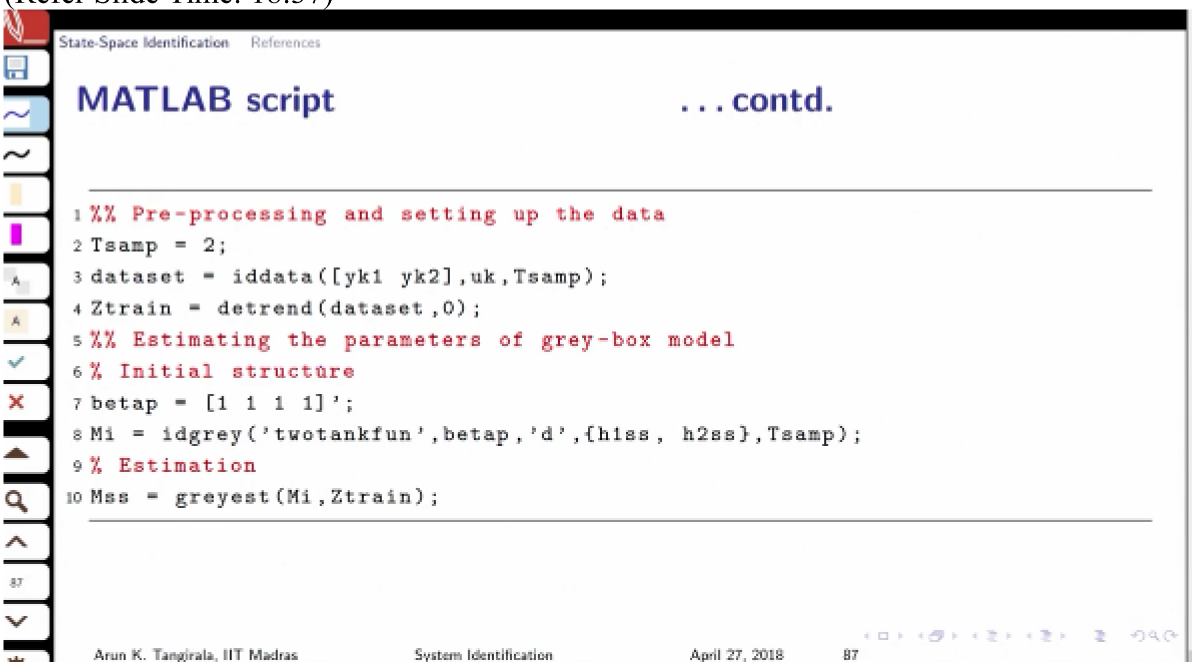
step 2 discretize, you have learnt how to discretize, the discretize version may have either the same or a different structure from the linearized one, we know that already, so the discretize state space model is then converted to an input output parametric model which is what time does for you, you don't have to do that, and the entire model is now a function of this beta, so the parameters here unlike in the previous example are not the entries of A, B, C, D, rather than that the parameters are beta of which the entries are a function, so that has to be remembered, so what we do is we with the MATLAB script that I have here, we run the estimation algorithm so all you have to do is you have to set up first a function that returns a discretized one,

(Refer Slide Time: 18:06)



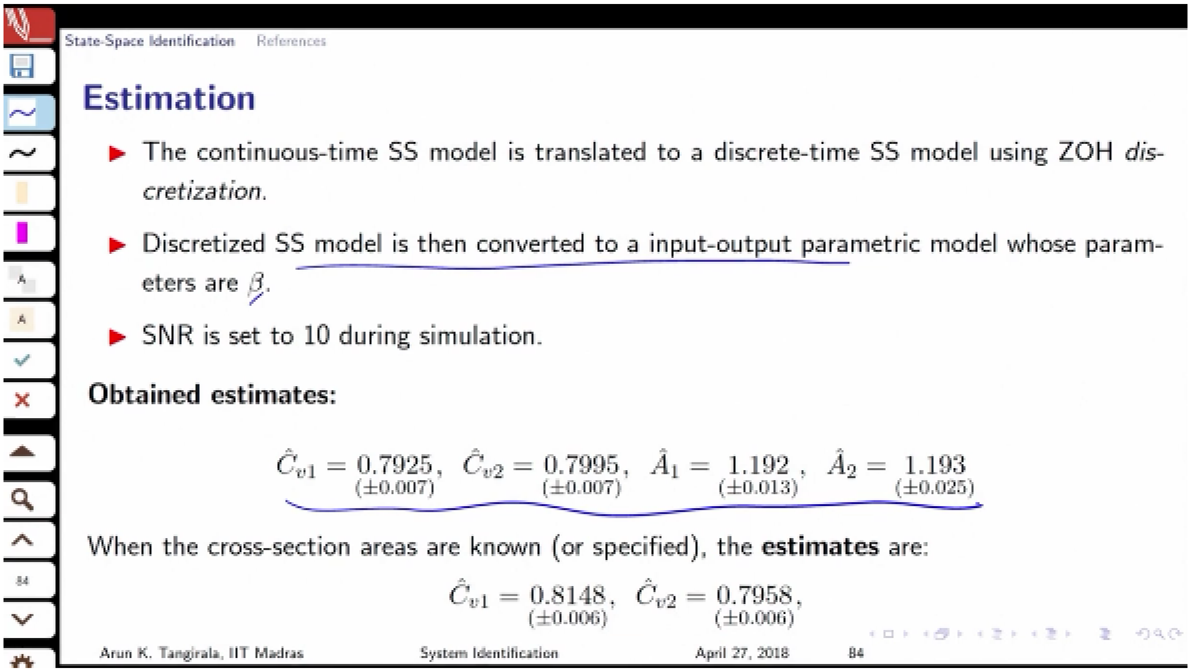
what does it take in? It takes in beta, the sampling interval and the steady state values and it returns the discretized state space model, this is the optimizer needs for a chosen value of beta, it will figure out what the state space model is, then see if the state space model is optimal, if it doesn't then it will keep it rating.

(Refer Slide Time: 18:37)



And then you pass that on to ID grey, you can actually do that this is data generation and once you generate the data you use ID grey. Two tank fun is the function that I have return, that returns the structured state space model and beta P is the parameter I'm suggesting, D is essentially the vector that I'm passing along, and T samp is the sampling interval, and then ultimately once this state space model is set up then you pass it on to greyest which will get you the estimates, so once you do all of that these are the estimates that you obtained.

(Refer Slide Time: 19:20)



Now I'm reporting the errors, because this model is identifiable, so I have CV1 hat, CV2 hat, A1 hat, A2 hat we can compare with the true values, I've used 0.8, and I have used 1.2 as a cross sectional areas, pretty close.

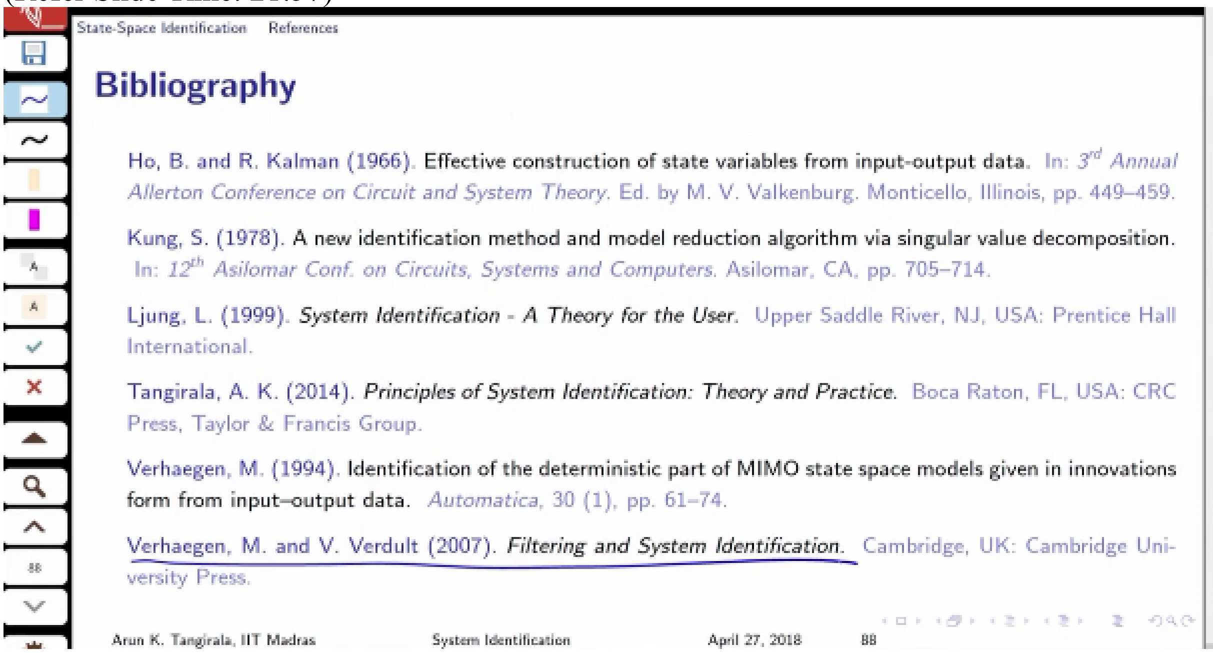
Now suppose I say that I know the cross sectional areas, it was only to trouble you that I asked you to estimate A1 and A2, then you drop that from your beta only C1 and C2, CV2 are the parameters of interest, then I obtain these estimates, keeping aside the point estimates you should notice that there is a drop in the error, not significant but definitely there is a minor drop, that is because the number of unknowns has come down, so this is the grey box modeling story for you.

So with this we close the discussion on state space identification, so you've learnt in the state space identification the basic concepts, there were a lot of concepts that we have learnt in fact, starting from observability, controllability, then the philosophy of Kalman filtering, innovation state space form, the main challenges in state space identification, then the key ideas in Ho and Kalman's method and subspace ideal algorithms, all of that pertaining to black box or freely parameterize or unstructured state space identification, then very briefly we spent some time on structured state space identification, which allows you to identify state space models with constraints, but the difference between unstructured state space identification and the structured one is that, you are actually solving an optimization problem in the structured case, whereas in unstructured one you are not explicitly solving an optimization problem, you are using projection methods which is the beauty of the subspace ideal algorithms, and that's why they have become very popular.

And although you are not explicitly solving an optimization problem, it has been shown that there is a prediction error equivalent to that, that means optimality is not being sacrificed in anyway, and that you'd be expected because in least square method we know that the solution can be obtained through projections, so that optimality should be expected. Any questions?

Okay, so that closes the state space identification thing, so I would, the book that I was referring to earlier was this book, it's a nice book to read, nice book to have,

(Refer Slide Time: 21:57)



and of course there are references to Ho and Kalman's method and Kung's method, the one reference that is missing here is the Van Overschee and Moor.

Online Editing and Post Production

Karthik
Ravichandran
Mohanarangan
Wilhelm Benjamin Anand
Sribalaji
Komathi
Vignesh
Mahesh Kumar

&

Web-Studio Team

NPTEL Co-ordinators

Prof. Andrew Thangaraj
Prof. Prathap Haridoss

IIT Madras Production

Funded by
Department of Higher Education
Ministry of Human Resource Development
Government of India

www.nptel.ac.in