NPTEL

NPTEL ONLINE COURSE

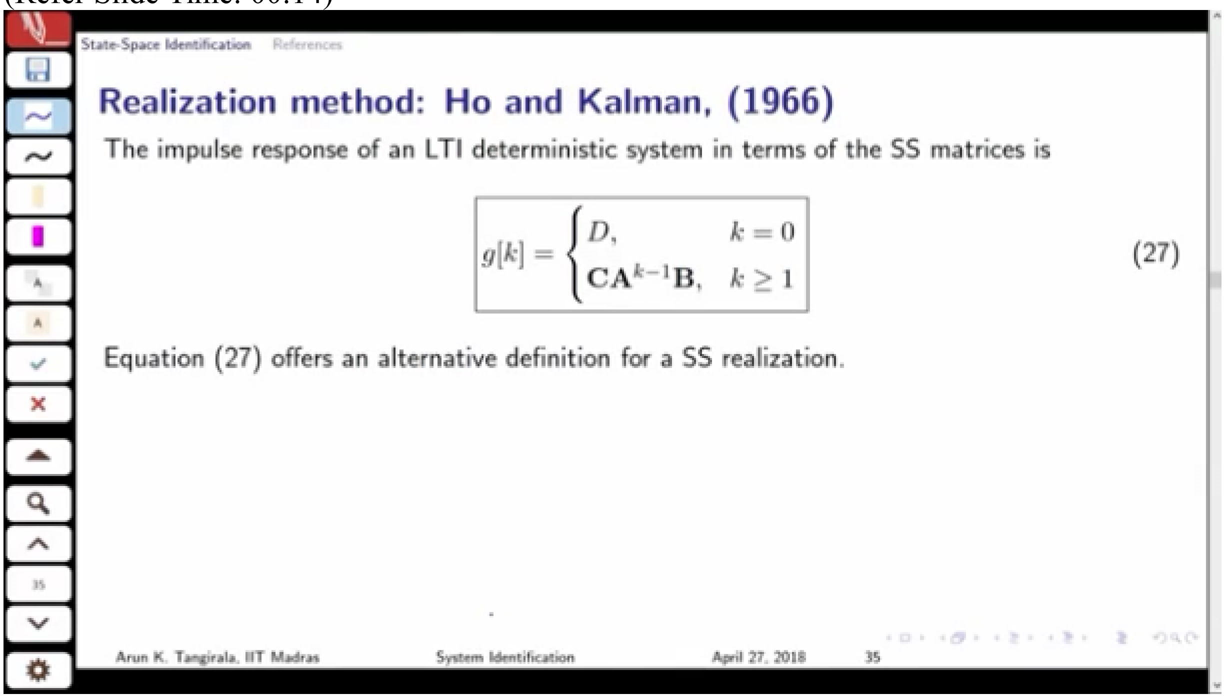
CH5230: SYSTEM IDENTIFICATION

STATE-SPACE/SUBSPACE IDENTIFICATION
PART 5

ARUN K. TANGIRALA

DEPARTMENT OF CHEMICAL ENGINEERING,v
IIT MADRAS

(Refer Slide Time: 00:14)

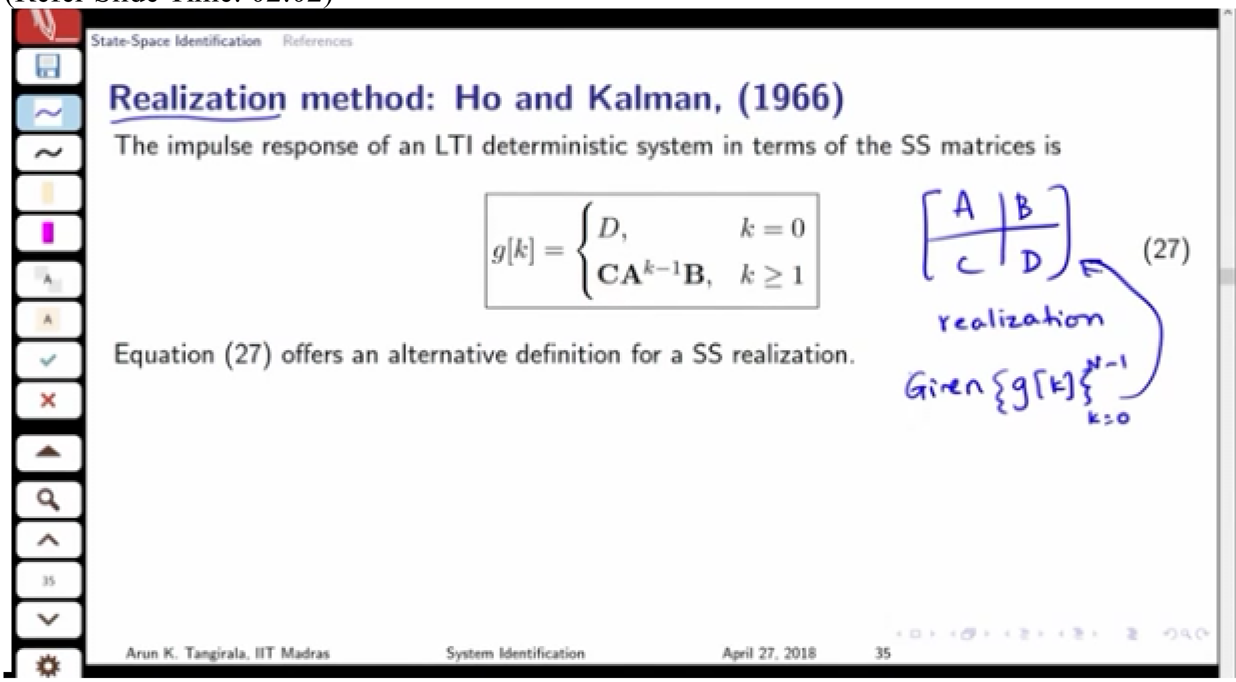


Let's start with Ho and Kalman's method which is considered to be the seminal method as far as the state space identification algorithms are concerned, and as I have said previously Ho and Kalman's method is suited for deterministic systems and for which impulse response data is available, so no noise, very nice setting, but it may be very easy today, but in 60's it was not such straightforward, because the state space models themselves where not so popular, and the mechanics of the state space model were not necessarily understood by everyone.

Whenever you construct a state space model from a data or from a model we call it as a realization, one realization, we've used this term before as well in another context in time series modeling, in state space modeling essentially A, B, C, D is set to be a realization, because I can realize the same system in many different state space ways, right, so there exist infinite number of realization, state space realizations for a given LTI system.

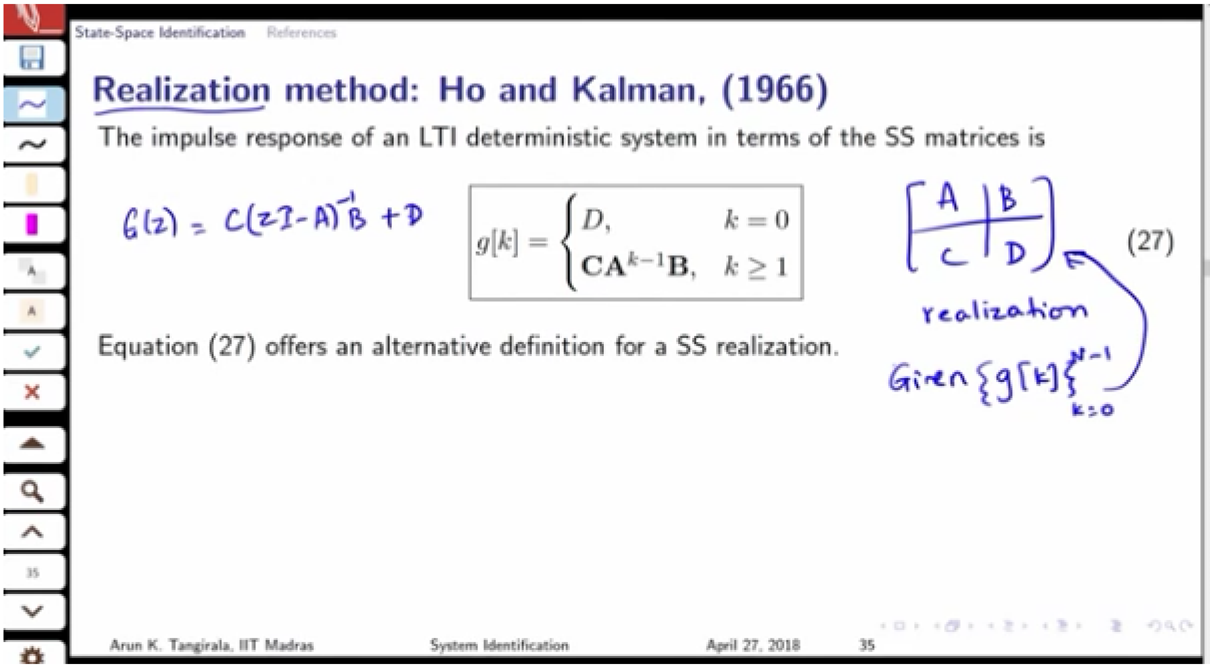
Now if I'm given A, B, C, D, because impulse response data is available, what is the objective here? I'm given the impulse response coefficients, let's not worry about how we obtained it, it could be through an experiment or through some means, but I'm given this and the goal is to identify this, recover this, this was the objective of Ho and Kalman's problem.

(Refer Slide Time: 02:02)



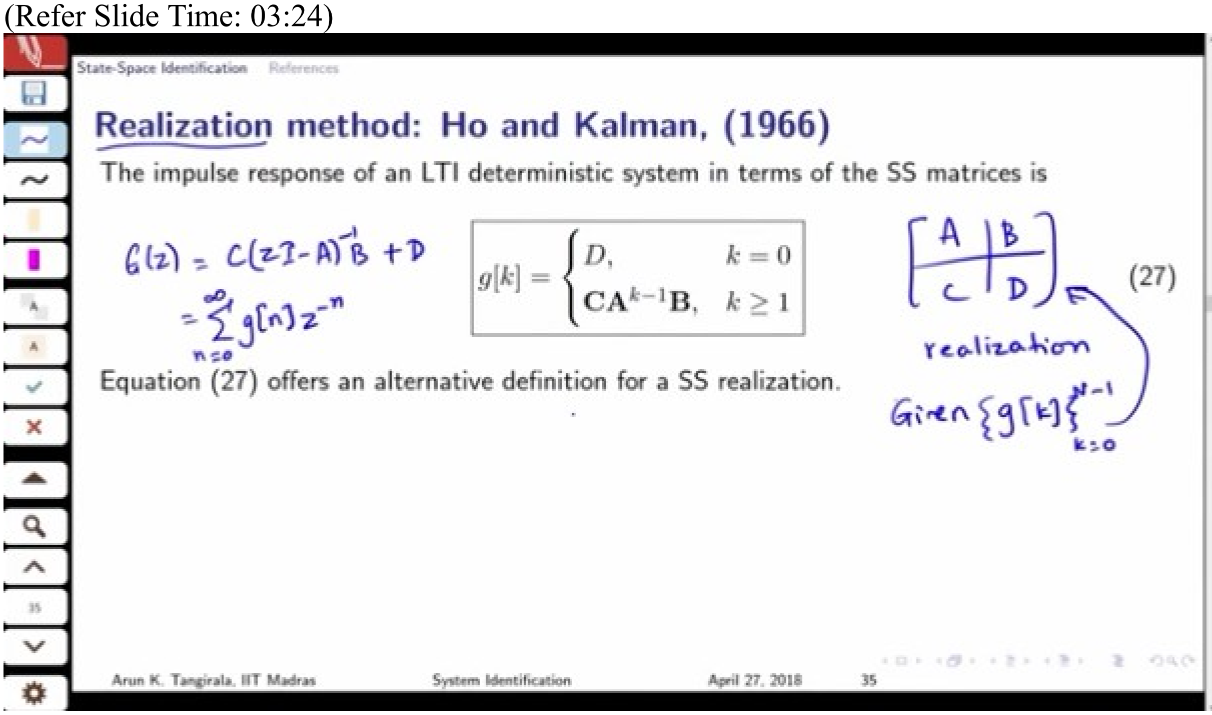
Therefore I need a mapping between the impulse response coefficients and A, B, C, D, and that's exactly what is given here in this equation in the box one, and you can very easily derive that, how do you derive this expression by the way? Correct, very good, so what you do is you write G(z) you've written this before, right, what is this? In terms of state matrices times ZI – A inverse B + D, so what do you do now? From here to get your impulses response coefficients long division, very good,

(Refer Slide Time: 02:47)



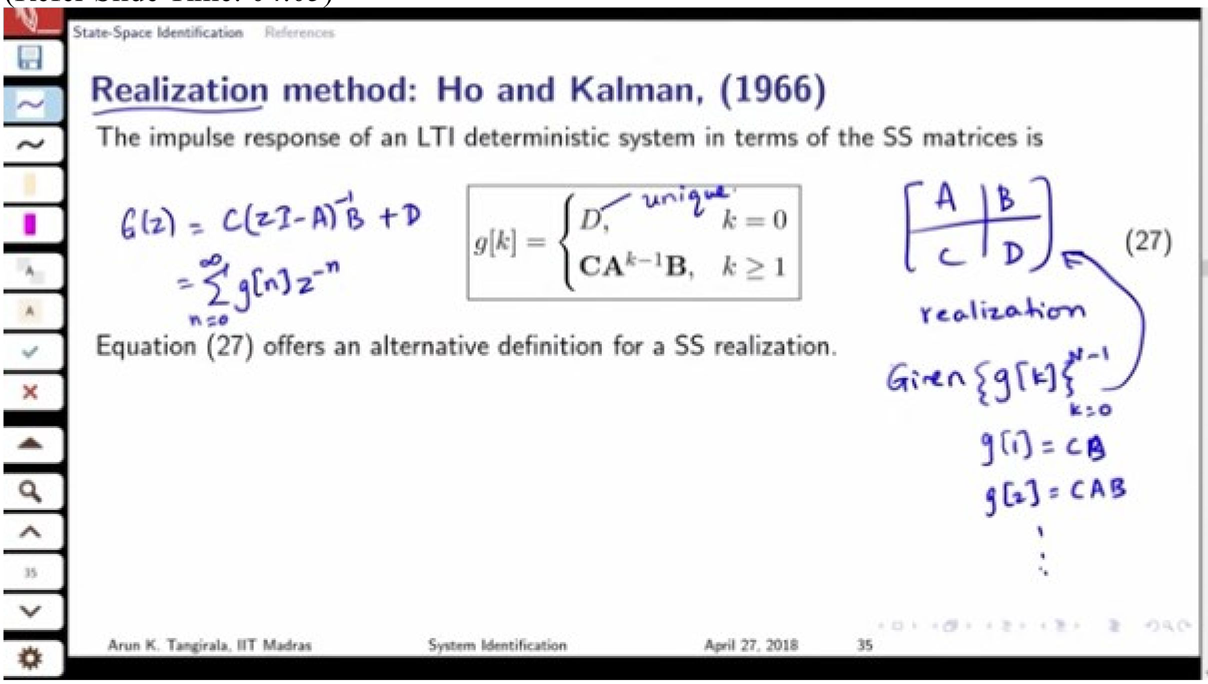
so you just do a long division and you'll notice that the first coefficient is D, so compare the long division with the definition of G(z), which is G(n) Z to the –N, when you compare this two you can read off the coefficients, G(0) is D, and G(1) is C, B and so on.

Now you can see straightway that, by here itself you can make certain observations, D can be obtained uniquely, correct,
(Refer Slide Time: 03:24)



so this is unique, I don't even need to break any sweat there, the moment I have the impulse response coefficient G(0) is D, just by reading off the first coefficient, when it comes to C, A and B what do you notice here? So if you were to write you'd get G(1) as C times B, G(2) as C, A, B and so on.

What do you notice here as far as recovering this A, B, C from impulse response coefficient is concerned,
(Refer Slide Time: 04:05)

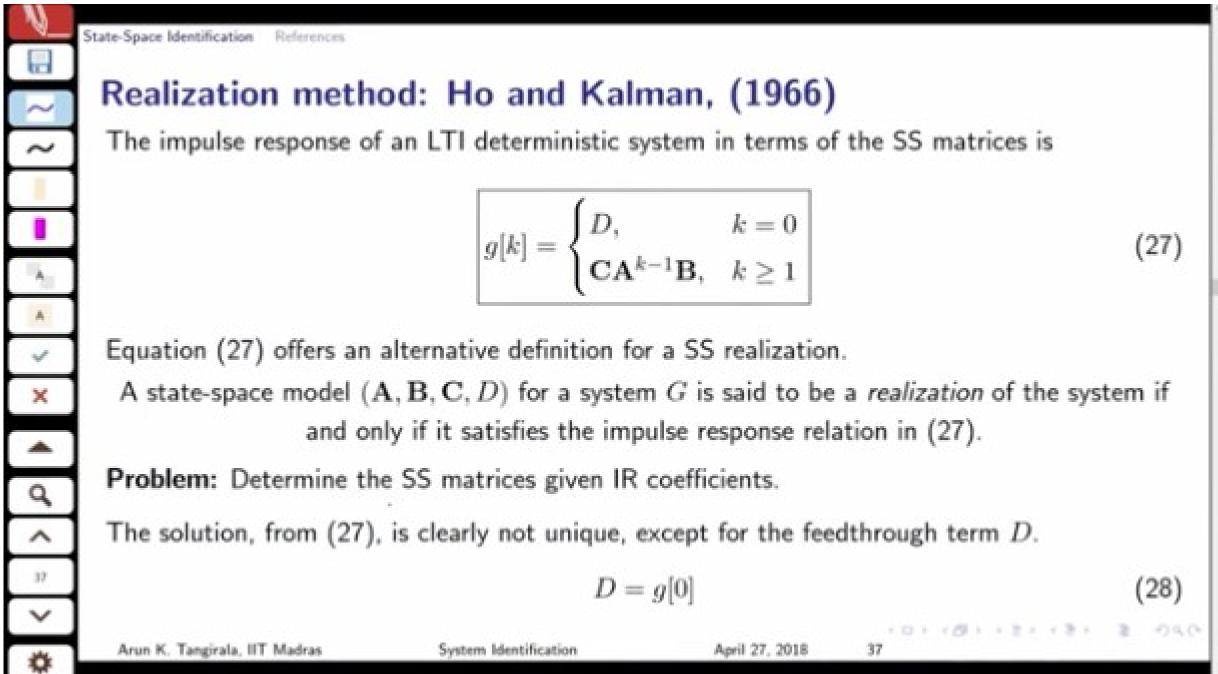


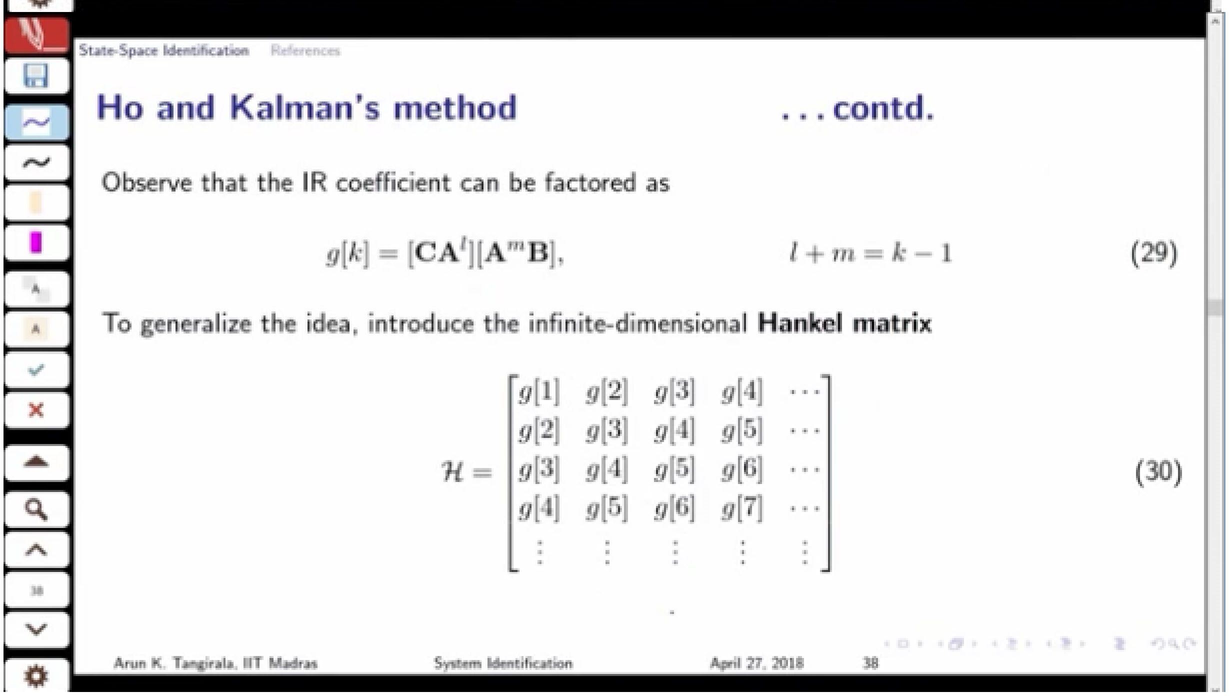
that there is a factorization involve, so what you're doing is you're actually factorizing each coefficient, correct, it's like I give you a number 6, can you find, can you express it as a product of two numbers.

Is there a unique way of doing that? There is no unique way, that's exactly the point here, there is no any unique state space model, forget about getting some data and so on, just if you look at this theoretical relations, corresponding to an LTI system there are many, many state space realizations that you can write.

Now look at it from a data view point, I'm given this coefficients and I'm supposed to recover this, and clearly the answer is, yes I can get it, but it's not going to be necessarily unique. And what we also see is that there is some kind of a factorization involved, and that is what you will find in all the subspace ideal algorithms, right from Ho and Kalman to the more modern ones, everywhere there is a factorization involved.

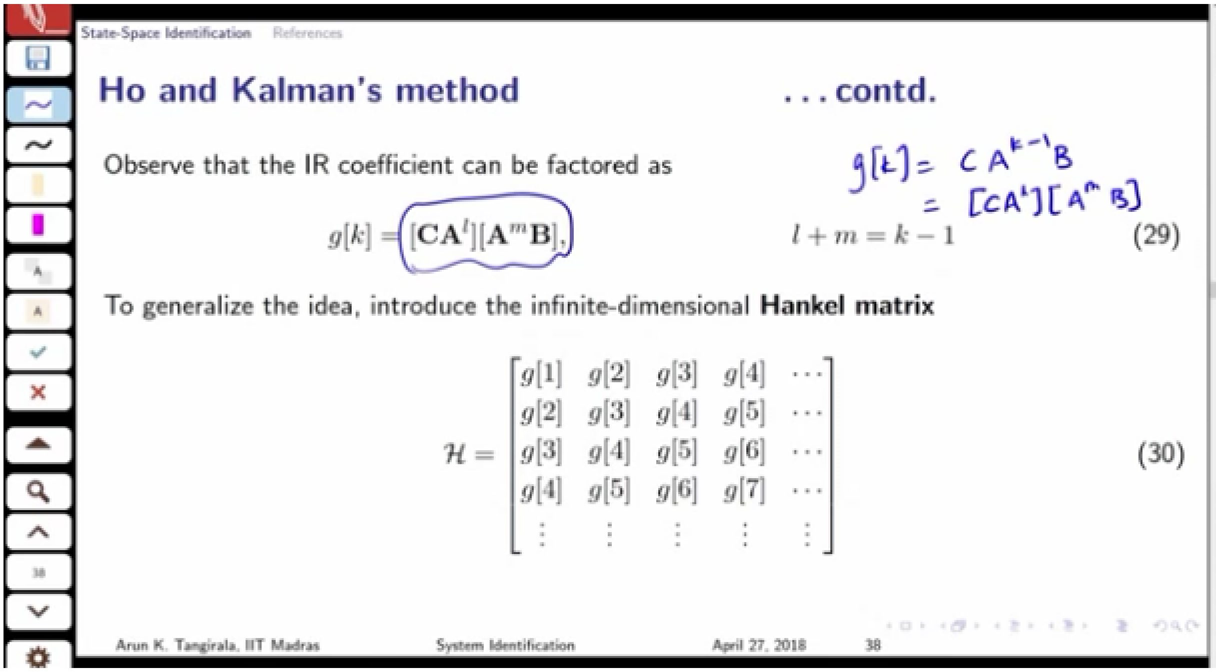
So we have already written what is a realization and this is the problem statement as I said,
(Refer Slide Time: 05:28)





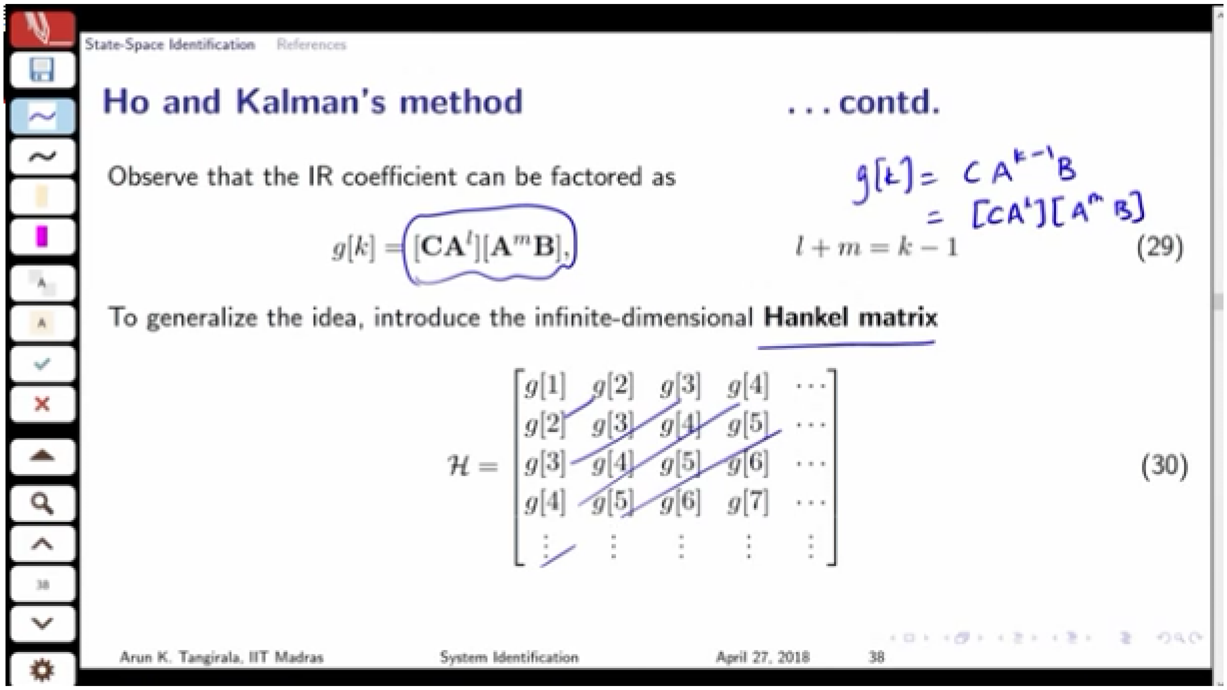
and we have already made this remark, and now you see what I have written here and what Ho and Kalman have done, you can take the impulse response coefficient and factorize it in this way, strictly speaking what is G(k) as per the previous result? CA to the K-1 times B, we have deliberately factored it as CA to the L times A to the M B, such that L + M is K-1.

(Refer Slide Time: 06:08)



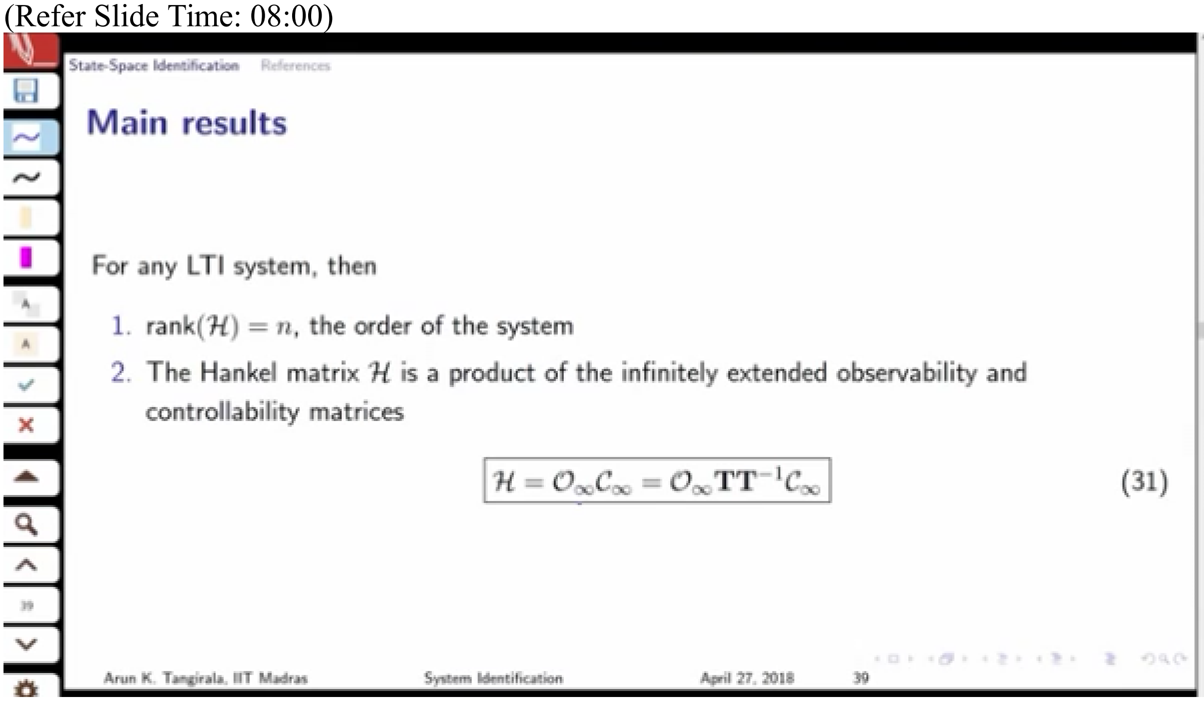
Why do you think we had, why do you think they have propose this factorization? Sorry, so you can see CA to the L is element of the observability matrix and A power M to the B is the entry of a controllability matrix, so which means it gives us some idea, the impulse response coefficients perhaps if you were to stack them in a proper in some way you can factorize that matrix of stacked impulse response coefficients as perhaps product of observability and controllability matrix, that is the idea. See look at how, there is nothing mysterious here, it's all the matter of perspective and some smart thinking, so yes, so what Ho and Kalman suggested is a construction of this Hankel matrix which consists of, this is an infinitely long Hankel matrix, let's not worry about the practical situation right now, because in practically I may not have infinite number of IR coefficients, let's assume they are available, we're still looking at theory, this Hankel matrix has a special structure if you see, these two are similar, these are similar, these are similar and so on,

(Refer Slide Time: 07:29)



that is the nature of the structure of a Hankel matrix, right, or you can also look at these diagonals here, they're also equal that doesn't matter.

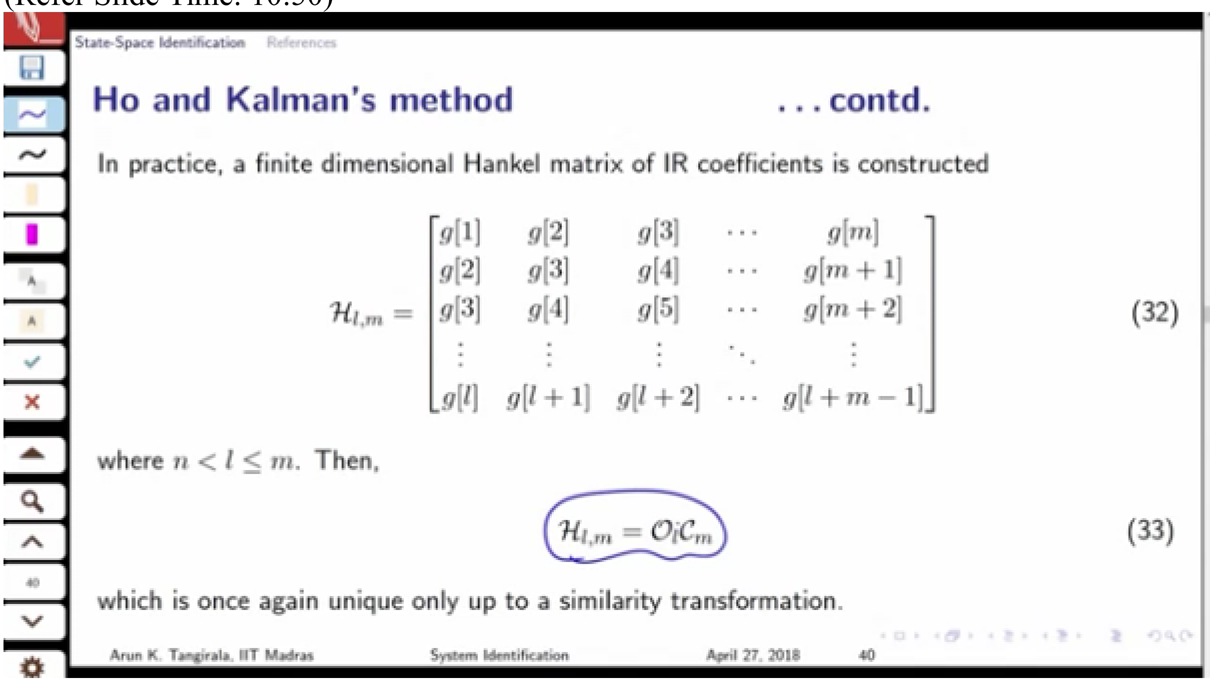
Now what do you expect the Hankel matrix to be expressed as? Given this relation here, and given our discussion that we just had, you should be able to express, correct,
(Refer Slide Time: 08:00)



so product of this extended up to infinity, because it's an infinitely long Hankel matrix and that's exactly the result here, right, and of course remember that is not unique, I can always write this product as O infinity times T, times T inverse C infinity, what does this tell me now? This tells me that once again if I try to recover the state space model from this Hankel matrix, I may not get a unique result, and the state space model will be correct up to a transformation T, that ambiguity will prevail, so what do you think is now the procedure? The procedure is to get, you first construct this Hankel matrix and then do a factorization, consider one factor as observability, other factor as controllability, and then get your order and state space matrices that is the idea.

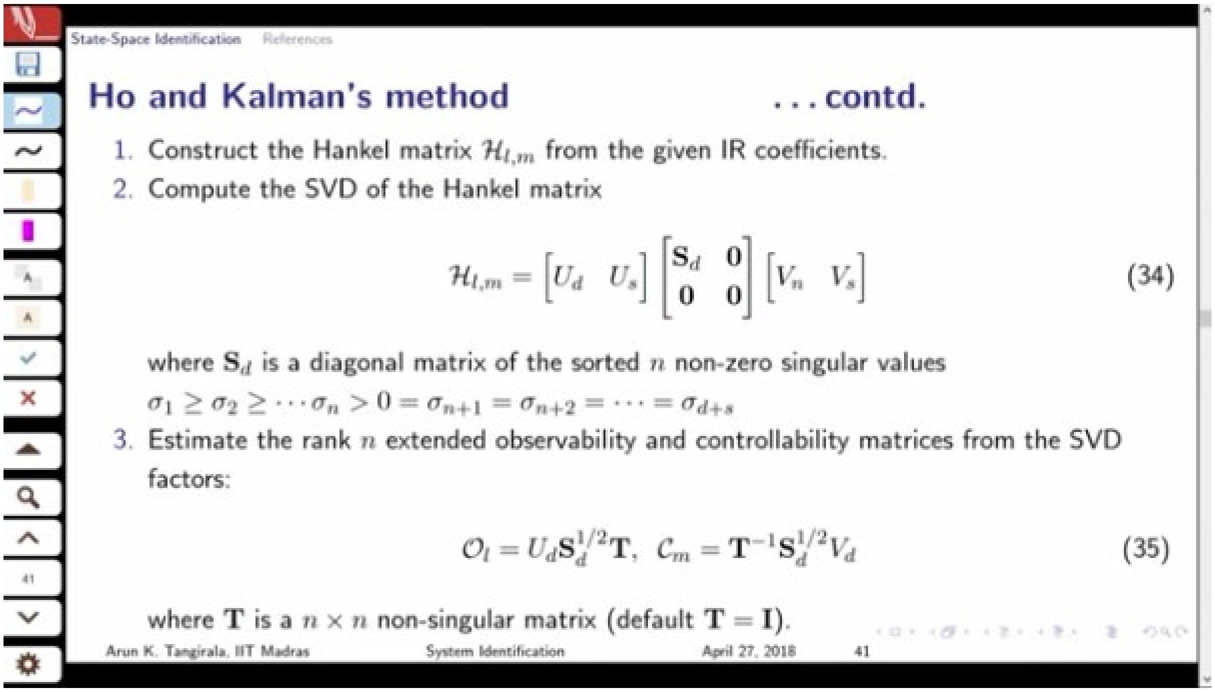
But before we do that it's necessary to be assured that this rank of H is N, that means it's full you know, that the rank of this is the same as a number of states of a minimal realization, that any way you can prove, because I'm leaving aside a proof you can look up the literature.

Practically it is not possible to construct an infinitely long Hankel matrix, now let's come to practicality, I can only construct up to a certain length, we depending on how much data is available, so this is how you construct a finite dimensional Hankel matrix where hopefully now you have included enough rows and columns, right, that means now some mild guess has to be made on the order which is also the case I think for people some of you who are learning dynamic PCA in the multivariate data analysis course you must have learnt that in dynamic PCA you stack variables when you want to build dynamic models, and since you do not know the order Apriori, and you still want to build a model you stack in excess, what is excess? You can keep on arguing forever and you'll never find an answer, but reasonably in excess, so if you think that the process is first order or second order or some low order you stack it in way in excess, so same story here, you stack the impulse response coefficients sufficiently in excess, and then you can show that when you stack H this way,
(Refer Slide Time: 10:50)



that you can write it as a product of OL times CM, you can choose L = M that is not an issue, alright.

And once again this factorization is unique only up to a similarity transformation,
(Refer Slide Time: 11:07)



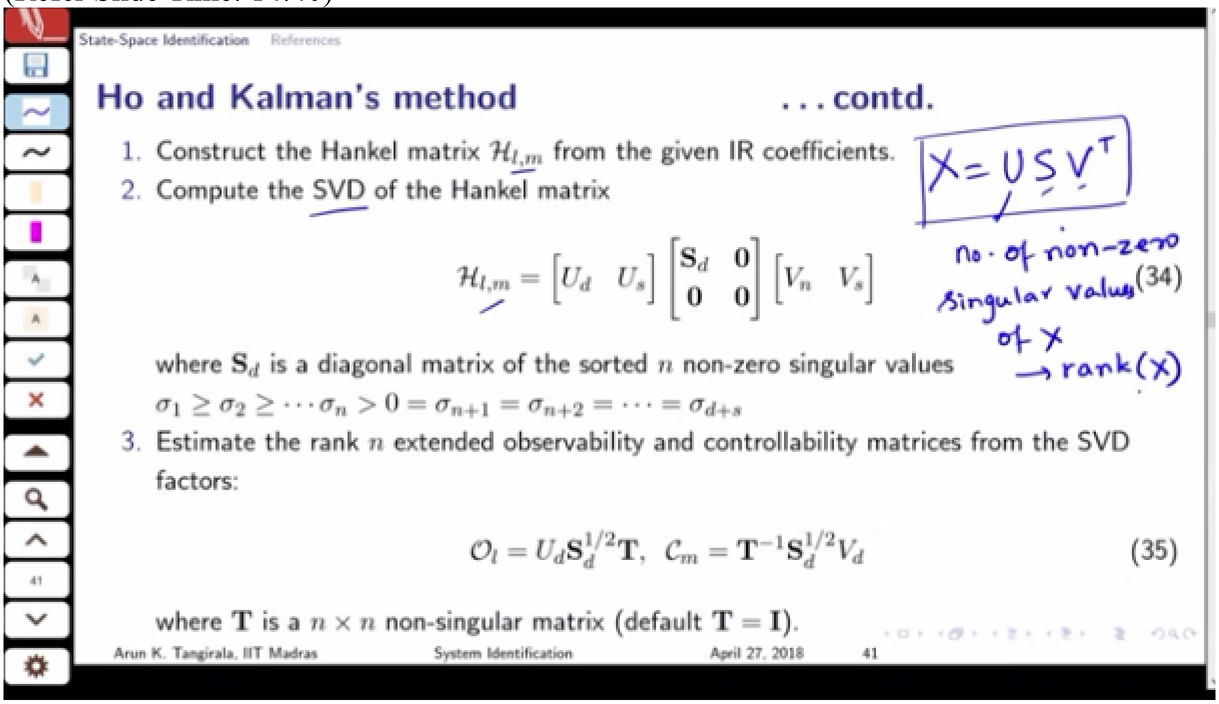
so this is the algorithm now of 1 Kalman, construct the Hankel matrix for a choice of L and M, user specify, and what is a guideline for choosing L and M? Sufficiently large, you will know it, in fact if you choose exactly then that means you have to go back and choose an excess because you might have missed out, see suppose I choose L and M as 2, just for the sake of it, for the sake of discussion, and the rank comes out to be 2, that means perhaps you made a mistake, so you want actually make sure that always ultimately the rank comes out to be lower than L, and then you compute for the first time we are talking of SVD, all of you have heard of SVD, singular value decomposition, so compute the singular value decomposition of the Hankel matrix, I can do any kind of factorization by the way, but SVD is optimal, I mean in many ways and numerically efficient and so on, and it has some very nice properties, so what you do is you do an SVD of H, so as to obtained the factors OL and CM.

And the reason for doing SVD also is because we know that ultimately after determining the rank we have to obtain the full rank versions of OL and CM, ultimately we want to get that, and we want to determine the rank, so all of that can be done in a single shot through SVD which means the order determination, the extraction of OL and CM everything can be done in a single step and that’s a beauty of working with SVD.

And in general for any matrix X, the SVD expresses X as a product of 3 factors where U is the matrix of left singular vectors and S is a kind of diagonal structure rectangular, but it has only diagonal elements so to speak of what are known as singular values, right, so that is about S and as far as V is concerned, it’s a matrix of right singular vectors, and U and V are both the columns are orthonormalized so that they have unit norm, and the columns within V are orthogonal to each other as well, right, so all of this can be given also an eigenvector decomposition flavor, typically you run into this PCA, in fact essentially what you are doing is you’re doing a PCA of H, that’s all you are doing.

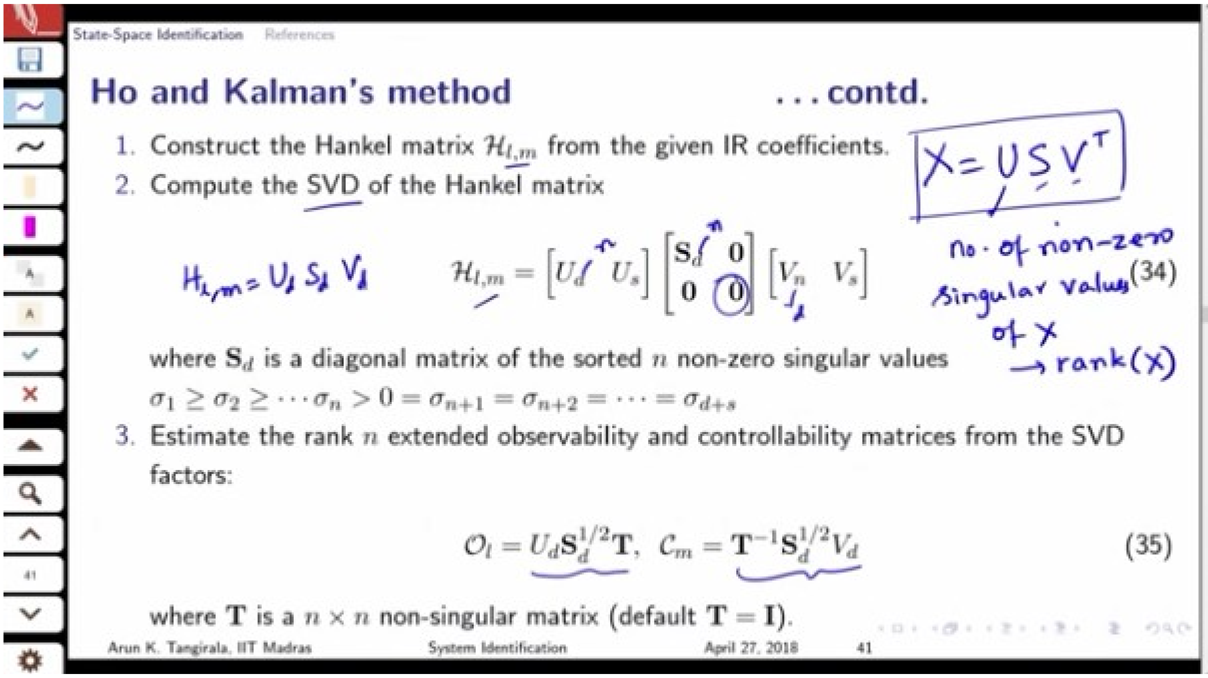
Since you have stacked L and M in excess of the actual order you should expect certain singular values to 0, what happens is when you perform SVD of any X, when you do this kind of a factorization the number of zero singular values or let me put it in nonzero singular values, nonzero singular values of X is, what is it? Is the rank of X,

(Refer Slide Time: 14:40)



okay, so assume that you have stacked an excess and therefore H will have, H will be of lower rank than L and therefore certain singular values will be zero and that is why I have put a zero here, so that now you can write HLM as UD times SD, in fact this D is nothing but N itself, so what you are going to write is HLM is UD times SD times D if you follow the notation that I have written here, so estimate the rank and extended observability matrix if you want, you can write it as UD SD 1/2 transpose, and T inverse SD square root, so what we have done is if you write it out here H of HLM is nothing UD SD VD,

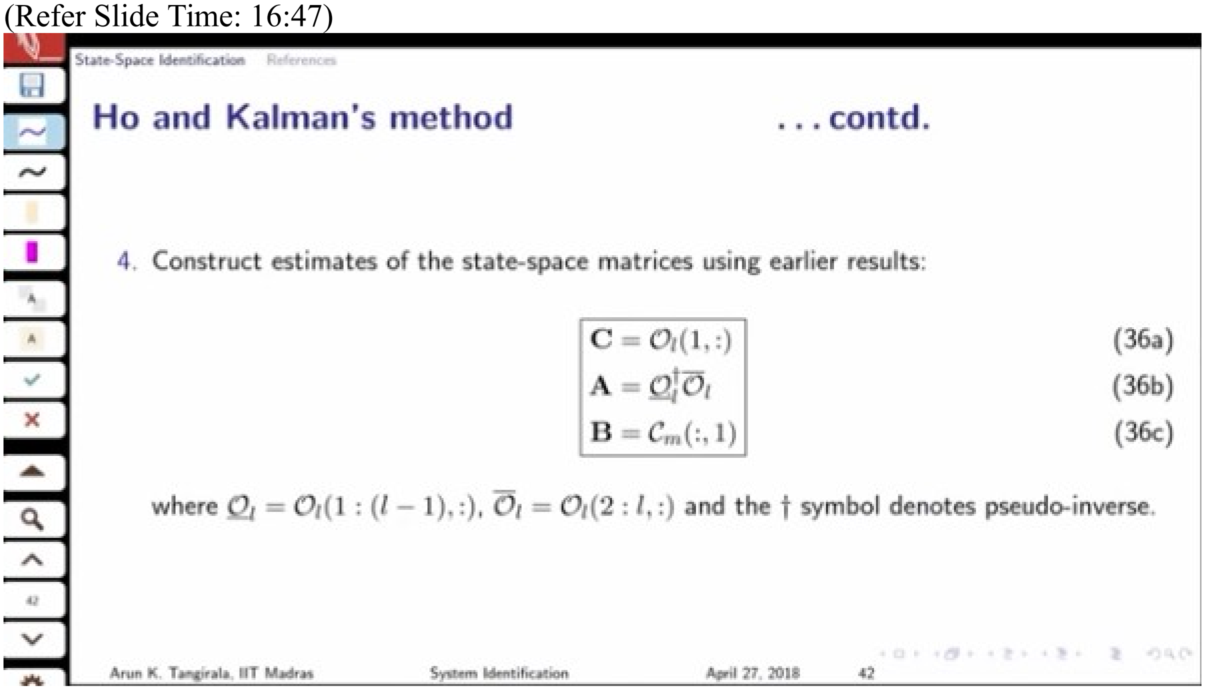
(Refer Slide Time: 15:43)



because this matrix is 0, so the contribution of US and VS is gone to H.

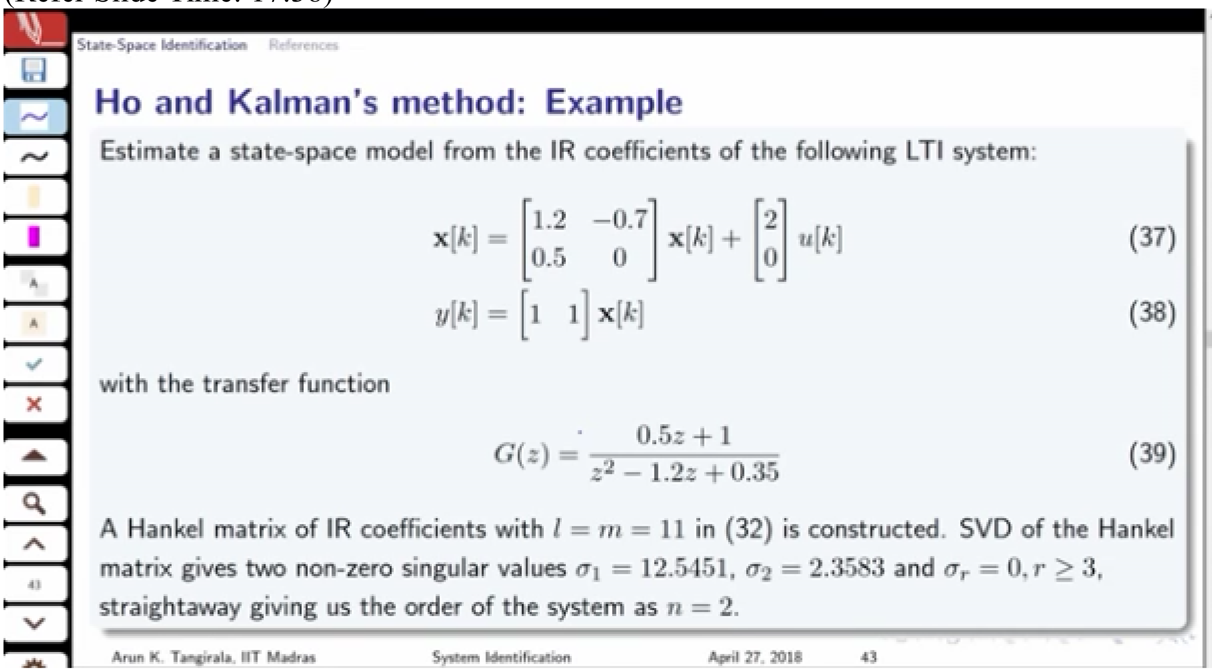
Originally it is USV, but since this L-D singular values are 0 we can construct a lower dimensional version of H, and that is what we had done, now it is all about factorizing this into O times C, and we know there is an ambiguity T so we put in that T here, normally we said T equals identity, if I have no particular choice of bases for the states, this is just to show that there is an ambiguity factor, so all you do is you assign, you can do this assignment in many ways of 2 and C, but this is generally followed that's it, so once you construct your OL and CM you can construct the estimates, right, you have C is O, 1

(Refer Slide Time: 16:47)



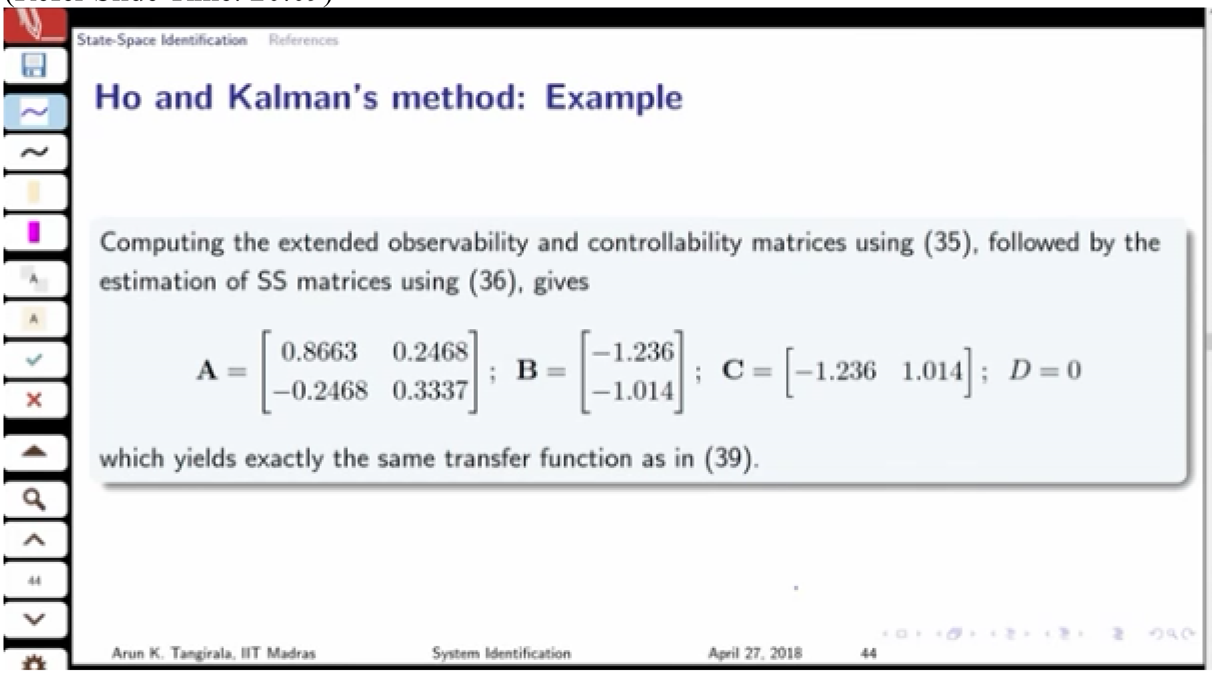
the first row or all the things that we have seen, although I write here, a pseudo inverse and strictly speaking this inverse is enough, and again this correction has to be made, we know that this is 1 to L and this is 2 to L+1, and likewise you can get your B, that's a D already you obtained, so very simple this algorithm became very popular because I have impulse response coefficients, I can construct a state space model from data, this was the first algorithm to come into limelight, because all along until then the state space models were built from first principals.

(Refer Slide Time: 17:38)



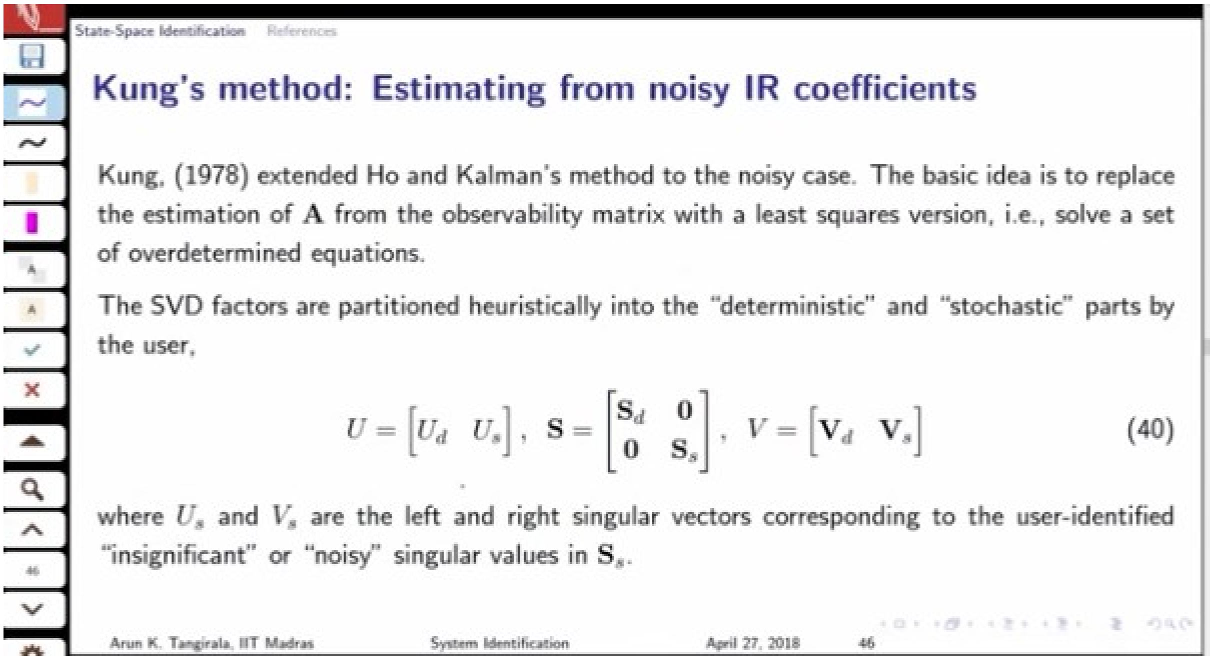
Let's implement this algorithm on a simple example, so this is our data generating process, and this is the transfer function just for our reference later on, so this is also the transfer function, we simulate this process, the MATLAB script for this example is available in the text and on the website, so we generate IR coefficients so that I can stack well L = M = 11, so I generate sufficient number of impulse response coefficients, construct the Hankel matrix than the moment I do an SVD as expected, what is the rank that I should expect? 2, because it's the second order, of course before you do that you should indeed check that I have this state space model that I have used is a minimal realization one, right, it is I'm just asking you to be careful, so as expected we get 2 nonzero singular values of sigma 1 and sigma 2, and the remaining ones are, how many are left? Sorry, how many this, I say sigma R = 0, R greater than 3 but what is upper bound and R in this example? The way I have constructed my H, what is the dimension of H? Right, so H is 11/11, which means 9 singular values are zero value, only two are nonzero.

Straightway giving us the order of to say it's beautiful, just from the impulse response coefficients I'm able to figure out what is the order, compare this with what we were trying to do in input output identification, of course we never had impulse response coefficients we had input output data, but that's where we are heading, we had to really guess the order, look at the step response, do a nonparametric analysis and all of that, and then still that may not be correct I have to do a residual answers refine and so on, here also you will have to do the residue answers eventually, but still it gives you a major leap in terms of start your, kick start order, that

is your initial guess of the orders, anyway this is still noise free data so all is well in the noise free world.
(Refer Slide Time: 20:09)


And then you construct the extended observability matrix and controllability matrix as per the algorithm that I have given, followed by the estimation of state space matrices using our results earlier, and these are the state space matrices that you get.

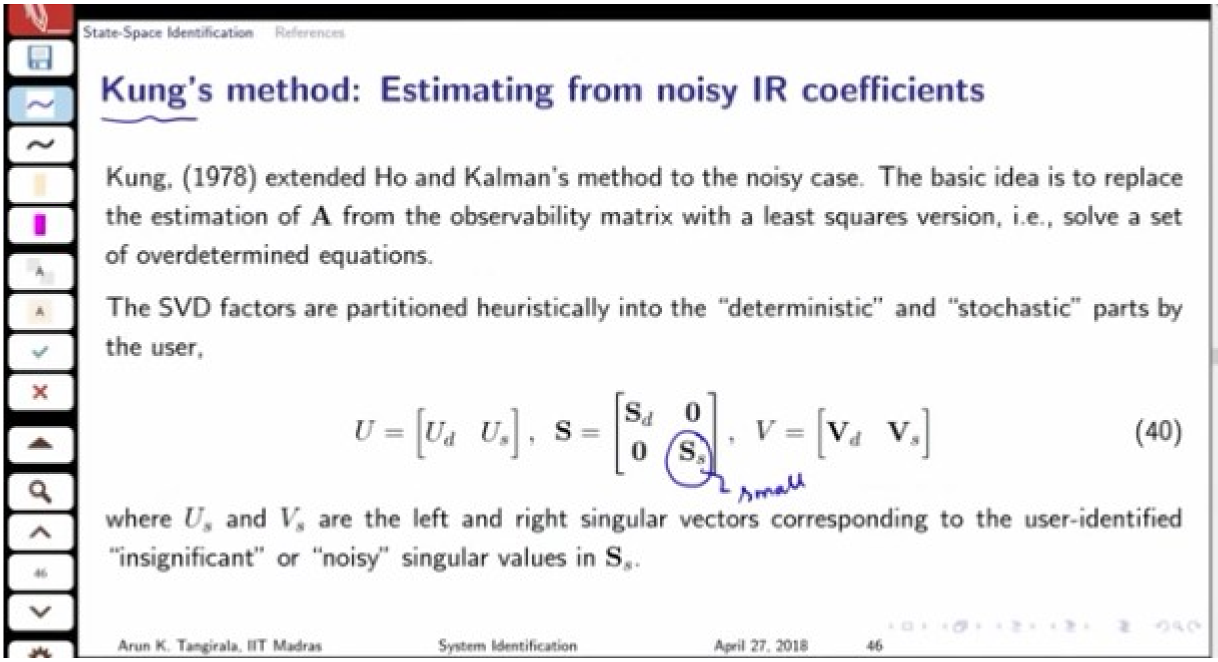
And it gives a same transfer function as you obtained, but this doesn't give me the, it gives me same transfer function, but does it give me the same state space model? It doesn't, that is the problem with your state space identification, you start with one realization, and generate your data but you could end up with the completely different realization. Could I have asked for this particular structure? Maybe, but it's going to be very tough, there is no constraint way of imposing constraints here, then you have to do a constraint SVD, for which you know algorithms are still developing, unconstrained SVD is the easiest one to implement. Any questions on Ho and Kalman's method? Alright, so this is the script.
(Refer Slide Time: 21:19)



Now let's quickly talk of the noisy case and about 10 years later Kung came up with a method which is not very dramatic I should say, although all credit to this method, all this method says is work with an approximation, it is like going from the noise free world PCA to noisy data PCA, so it's exactly that, we are using essentially PCA or SVD, we know PCA and SVD are synonymous, right, so what do you think whom would have suggested? Use the elbow plot, knee plot, hand plot, face plot or you know basically use an approximation, look at you make a call because this elbow plot, knee plots are very common terms in the noisy data PCA world.

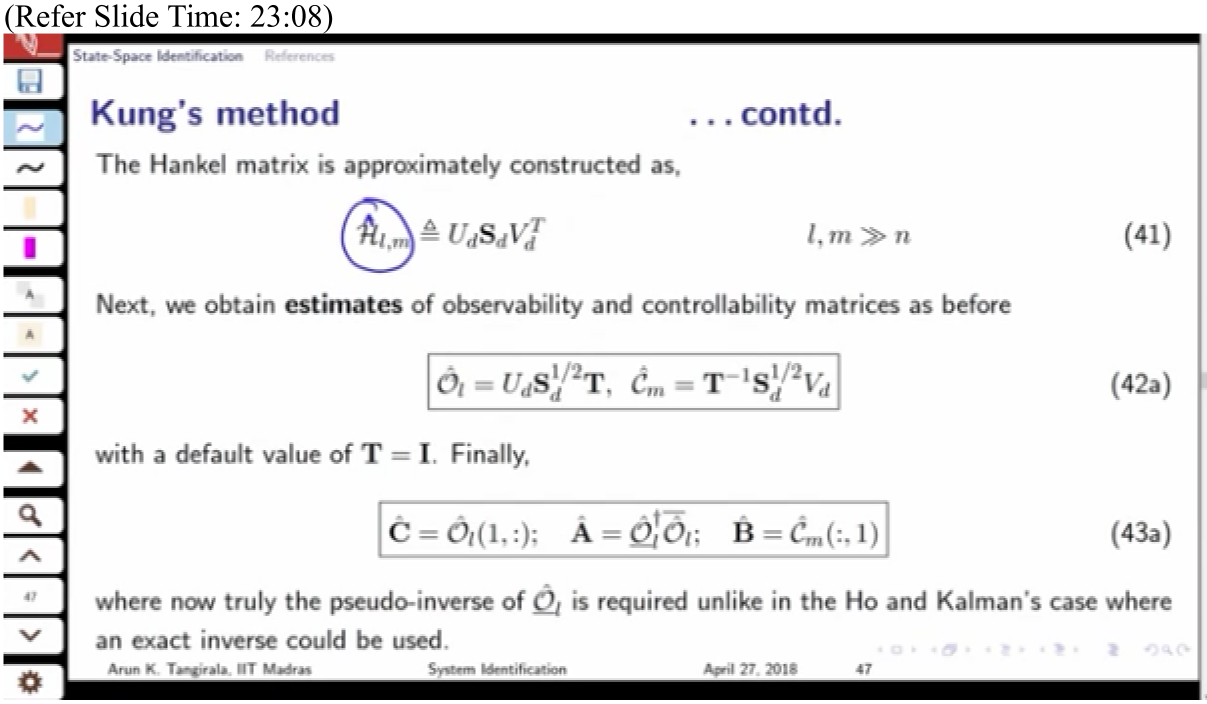
So what this method essentially suggest is, simply take your call decide what is small, okay, and then you construct a low rank approximation of the Hankel matrix and from there you extract the observability and controllability matrix, right, so that is what exactly now Kung's idea is, and notice now because of the noise, the zero singular values are missing, you will actually find all the singular values being nonzero, and this S stands for small you can think of it that way, this partitioning that I have done is now user dependent.

(Refer Slide Time: 22:54)



So now you construct a hat you see now H wears a hat, because it's a lower rank approximation of the H,

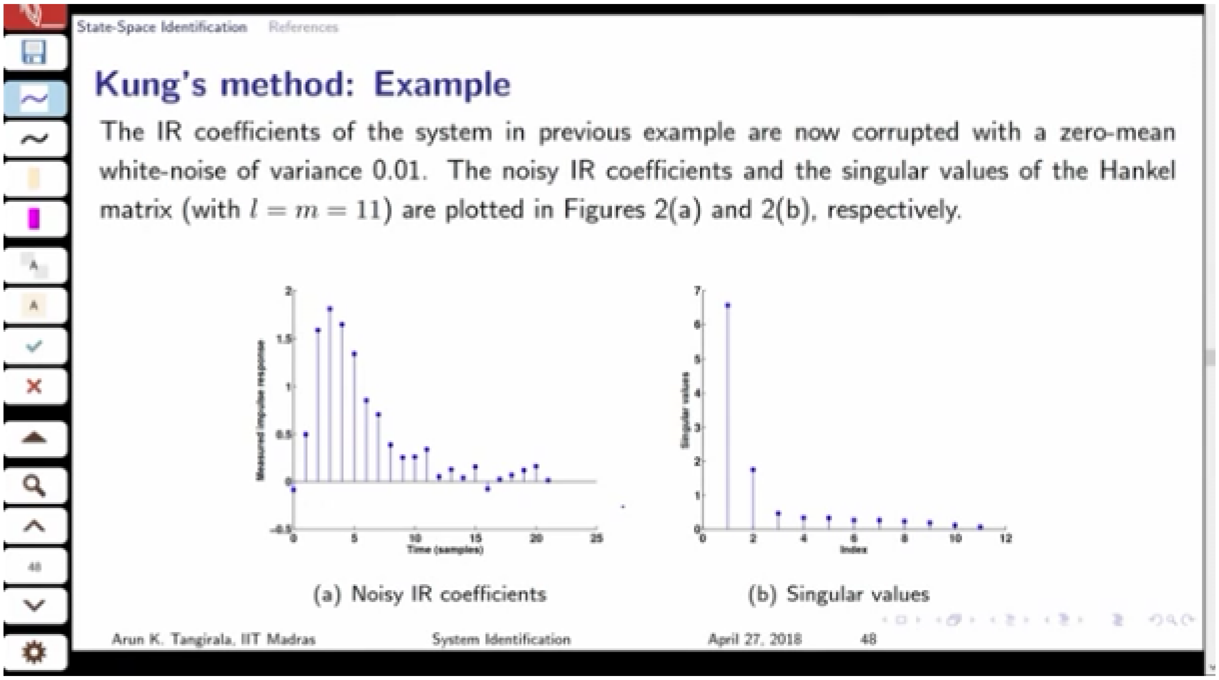
(Refer Slide Time: 23:08)



and from this lower rank approximation you construct estimates of observability and controllability matrices as before, and then extract your C hat, and this time you will truly use a pseudo inverse, so this time what you do is you don't exactly consider N rows, you'll take the full observability matrix that you have, you want, and construct use all the rows to get an estimate of A, so this time you won't get exact A, exact C and so on.

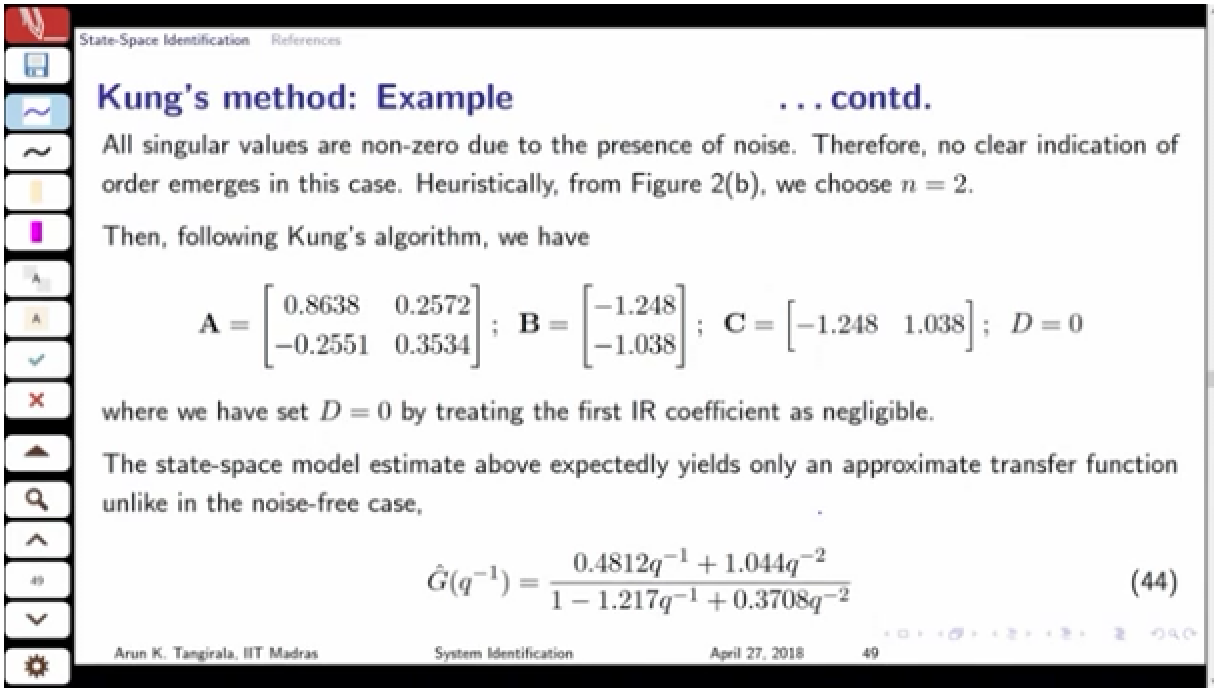
So let's look at the example here, the same system, now we have added some noise,

(Refer Slide Time: 23:42)



white noise of variance 0.01, it's not clear how this algorithm performs or colored noise and so on, we've just added white noise here, and the impulse response coefficients are shown here and the singular values are shown here. If there was no noise the singular values starting from 3 to 11 would be all zero value, but now they have woken up, the sleeping lions have been woken up, or sleeping kutty cubs have been woken up, and all of them are active, so the user can look at this plot and say may be the order is true, but if I'm a first C is low then I would say no order is 3, if I'm first C here then 4 and so on, so it depends, so it's all heuristic now, it's anyone's game, and that is the ambiguity that remains here, and at least there has been no attempt to determine the order exactly despite the presence of noise, I should be able to see that, but unfortunately I'm not able to see that, there is no method as of now, but we are presently working on a method to see if we can see through the noise and exactly determine the order, in a more convincing manner, right now it's heuristic, so we pick two, now let us see what happens, again the same procedure, we still pick two singular values,

(Refer Slide Time: 25:12)

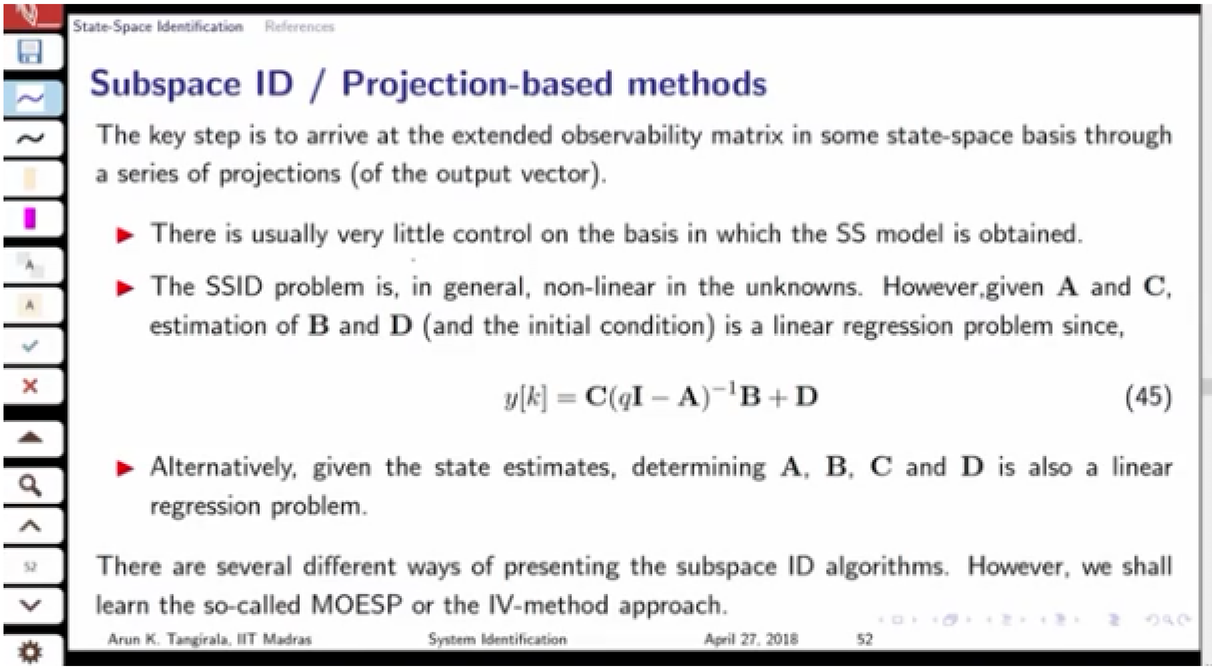


not because we know the answer because the plot seems to be suggestion two.

And remember if I change the variance of the noise, then what would happen? How would this plot change here, the singular value plot change? Yeah, you will start seeing even more significant values here, even more larger values, so the magnitude of the singular values here from 3 to 11 depends on the noise, and as the SNR drops the singular values will shoot up, making it even more difficult for you to guess the order, okay, so right now life is good because we have chosen white noise variance point zero and everything is in my control and now we obtain these estimates of A, B, C, and this is the transfer function that we obtain, a bit different from what we started off with, right, if you call recall what did we have? We had 0.5, 1.2 and 0.35, and 1 as a coefficients, whereas here we have 0.48, 1.044, 1.217, and 0.370 obviously as a noise variance increases you will find them even more different from the truth, okay, so any questions on Ho and Kalman and Kung's method, so to summarize what we have learnt is how to estimate state space models from impulse response data, both noise free and noisy versions, but in that process we have learnt the basics of sub space ideal algorithm which is that some where we'll stack data in some form, in this case we have stacked impulse response coefficients because that is what has been given to me, and then once I stack what do I do? I do an SVD, because that will then tell me the order, and once the order is determined I do a factorization.

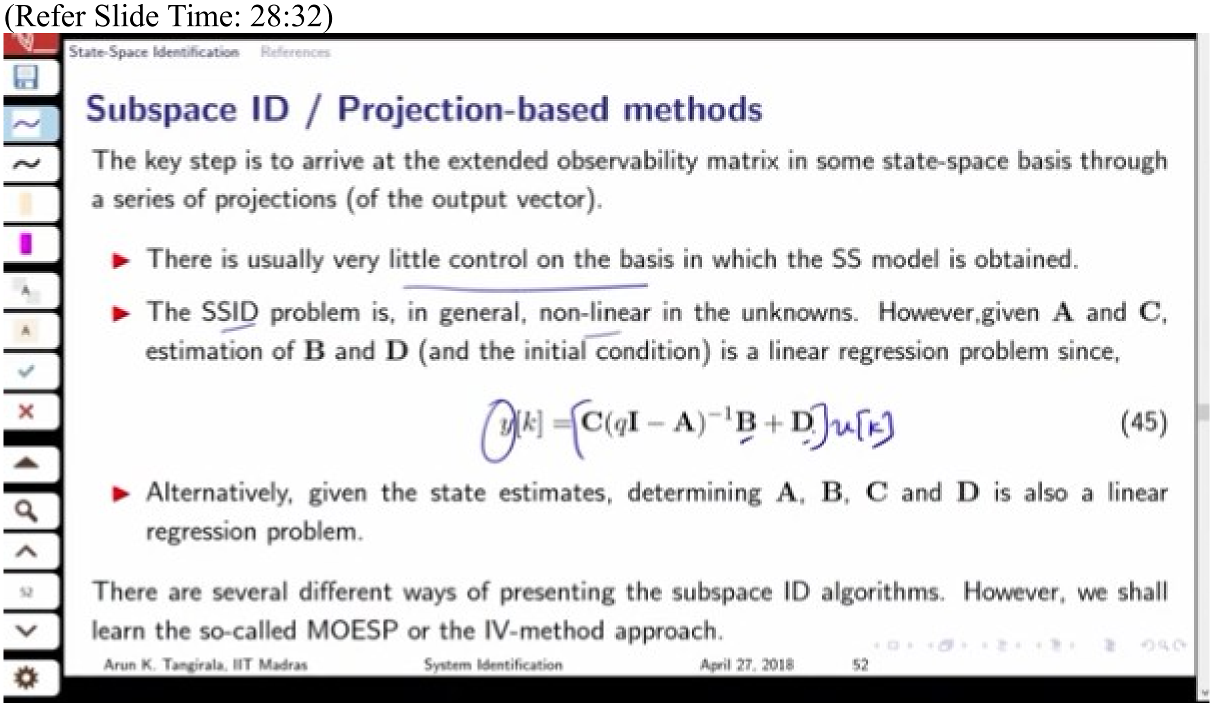
Anyway factorization order determination is all done in one go, I extract the factors namely the observability matrix and the controllability matrix get my A, B, C, that idea is the one that carries forward to the sub space identifications algorithms as well, the same idea.

So before we jump again let us reminisce that with sub space ideal algorithms as well as Ho and Kalman's algorithm there is going to be very little control on the basis,
(Refer Slide Time: 27:45)



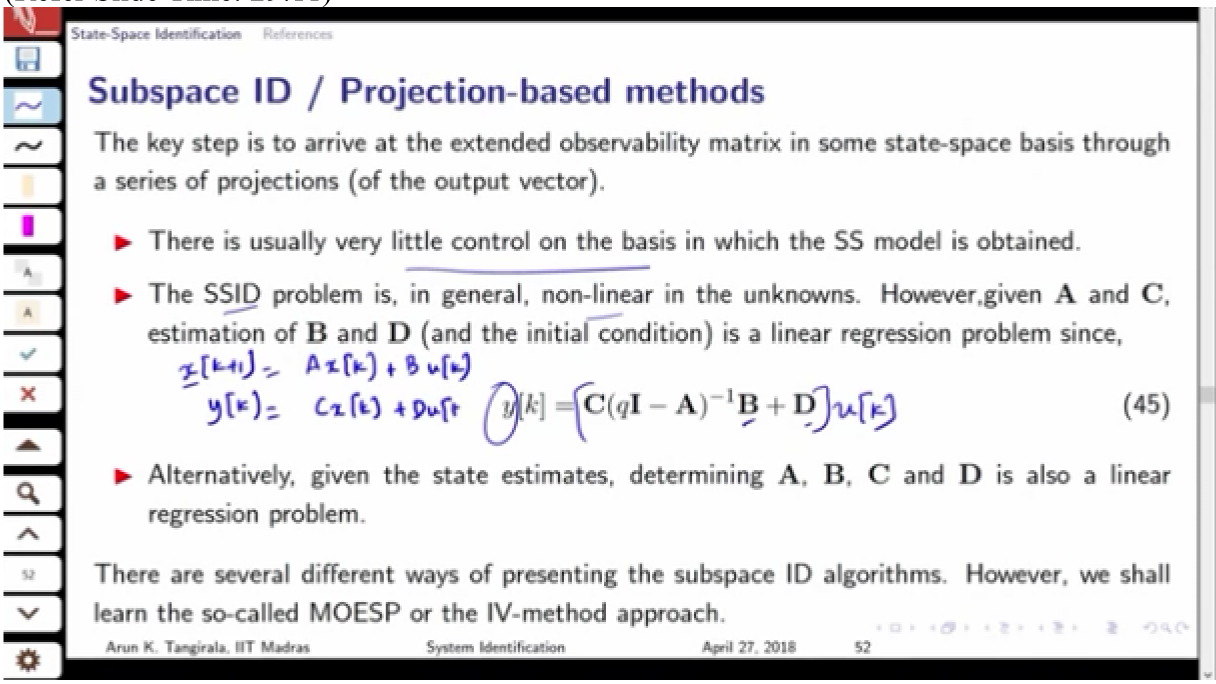
and in general that the sub space identification problem is nonlinear as I had mentioned earlier, because you are given now Y, this is missing here U(K), you are given Y and then the C, A, B, D all of them have to be determined, so obviously that factorization, I mean it's a nonlinear problem and it also has identifiability issues, alright,

(Refer Slide Time: 28:32)



and if I'm given the state equations, now suppose I'm given for example B and D in this, if you look at this equation, if I'm given B and D then I can estimate A and C, because what I can do is I can write this as a difference equation with A and C as the unknowns and you will see that is identifiable.

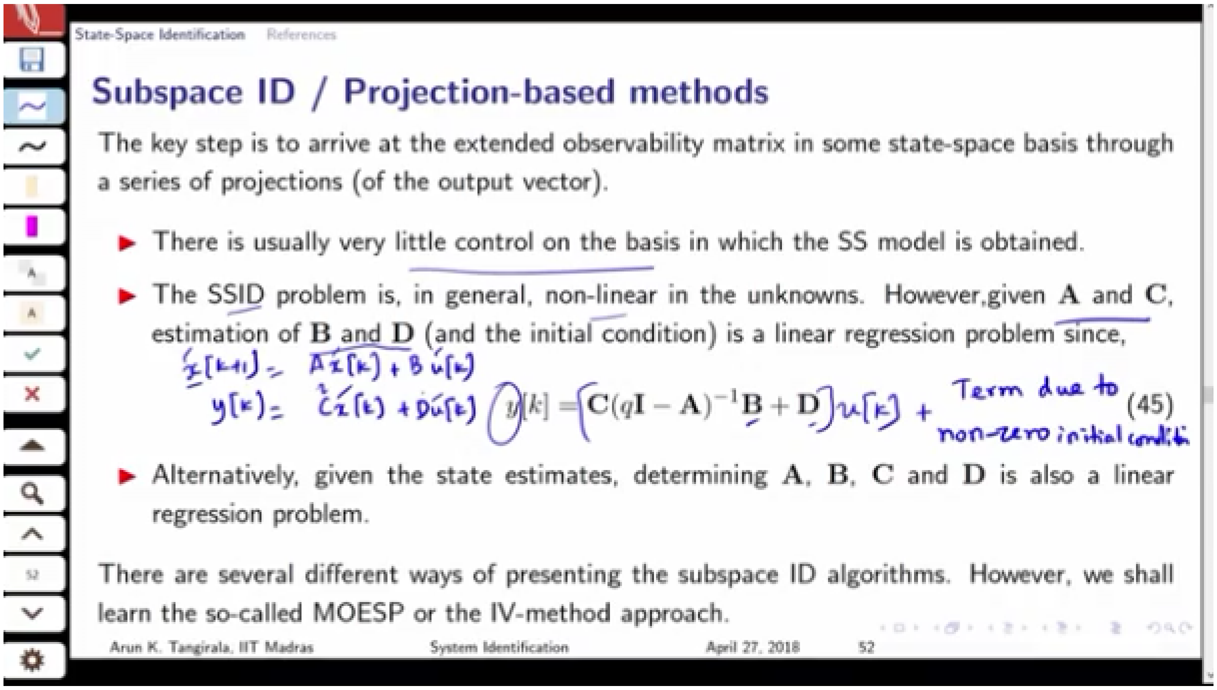
Alternatively if you go back to the state space equations CX + D I'm just writing the noise free version here, if I'm given the state estimates,
(Refer Slide Time: 29:11)



if I'm given X then it's very easy right, then it's a linear regression problem, I can estimate A and B because I'm anyway given U, and I'm given X and U so I can also get now C, so I'm given X and U, I can therefore estimate C and D uniquely, so what this tells us is that, sorry earlier I said given B and U you can estimate A and C it's not, it's other way around, if you're given A and C then B and D can be estimated easily through linear regression, so please correct that statement that I made.

Given A and C estimation of B and D is a linear regression problem as per this equation, in fact there should be also another term in general due to initial nonzero initial conditions, if that is the case you can include and you can even estimate the initial condition, okay.

So let me summarize in general the state space identification problem is a nonlinear identification problem,
(Refer Slide Time: 30:19)



but I'm given A and C, then I can estimate B and D through linear regression, that you can see from this equation, if I'm given C and A then it's simply linear regression problem because once you're given this you can write a difference equation form and get the coefficients B and D, and also the initial condition Y(0).

On the other hand if I'm given states, then I can estimate A, B and C and D of course through linear regression, this is the difference between the approach as in sub space ideal algorithms, one class of sub space ideal algorithms try to first estimate A and C, okay, once A and C are obtained B and D can be obtained, they bypass the state estimation problem.

The other class of algorithm says let me get the estimates of states first, once I get the estimates of states then all I have to do is a multivariate linear regression where I stack these and then I can write this entire thing as a linear regression problem, where A, B, C and D are just the coefficients of the linear regression, so that is the difference between the two prominent algorithms, the algorithms that estimates the state first and then estimate A, B, C, D are called N4SID algorithms, and the algorithms that estimate A, C and then B and D are called, they're essentially your MOESP algorithms and so on, and then there is a canonical variate analysis which is a slightly different version of this to algorithms.

Now these have their full expanded forms, for example N4SID is numerical nonlinear, no sorry numerical state space subspace identification algorithm, okay, that 4 is not on N, but that 4 is on S, okay, N stands for numerical, and 4 stands for state space subspace, because it would be a mouthful they call it as N4SID, and MOESP stands for multivariable output error state space algorithms or subspace algorithms, then CVA stands for canonical variate analysis, although there are these different names as far as a MATLAB routine is concerned there is only one routine and that is N4SID, it doesn't mean that it doesn't catered to MOESP and CVA, you just have to turn on the options and you can use MOESP or CVA and so on, okay, you should remember that, so the N4SID name in the literature is for N4SID algorithms, whereas N4SID routine in MATLAB is for all the algorithms, okay.

So what we will do is the easier once to understand are the MOESP ones, the more involved ones are the N4SID ones, and each of them has its own advantage and disadvantage and so on, but the core idea remains the same.

We'll spend now the next 15 minutes or so in understanding the MOESP algorithm, and the MOESP has an instrumentable variable method flavor to it, there is a very nice book on filtering and identification by Michael Verhaegen and others, that is a reference at the end of this slides, you should read that, it's I think in my opinion it's much more comprehensible then the deadly subspace identification book written by Overschee Van and Overschee De Moor, Van Overschee and Moor where the proponents of the N4SID. In fact the book is also very comprehensive, but you will find the book by Verhaegen a lot more accessible.

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

Copyright Reserved