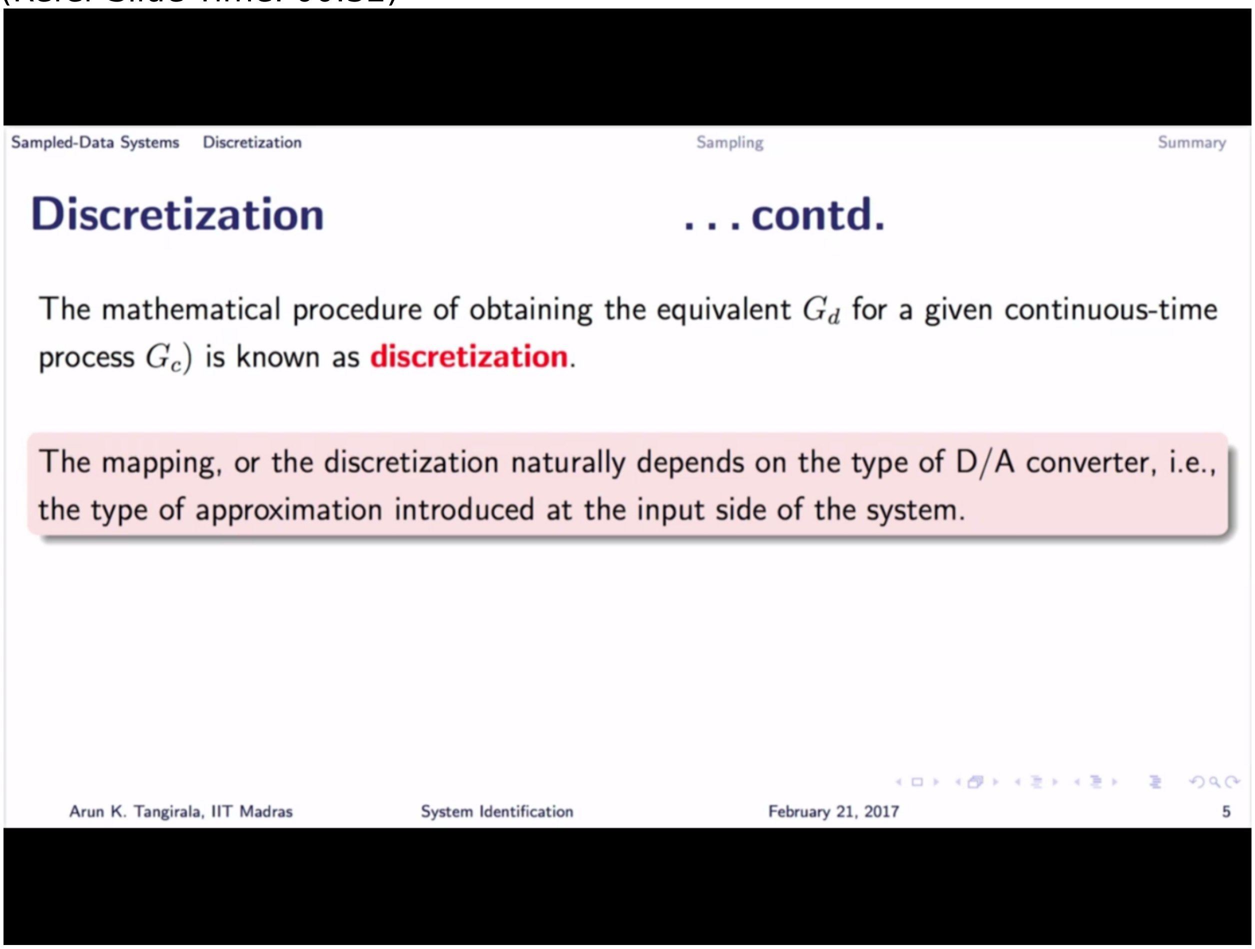
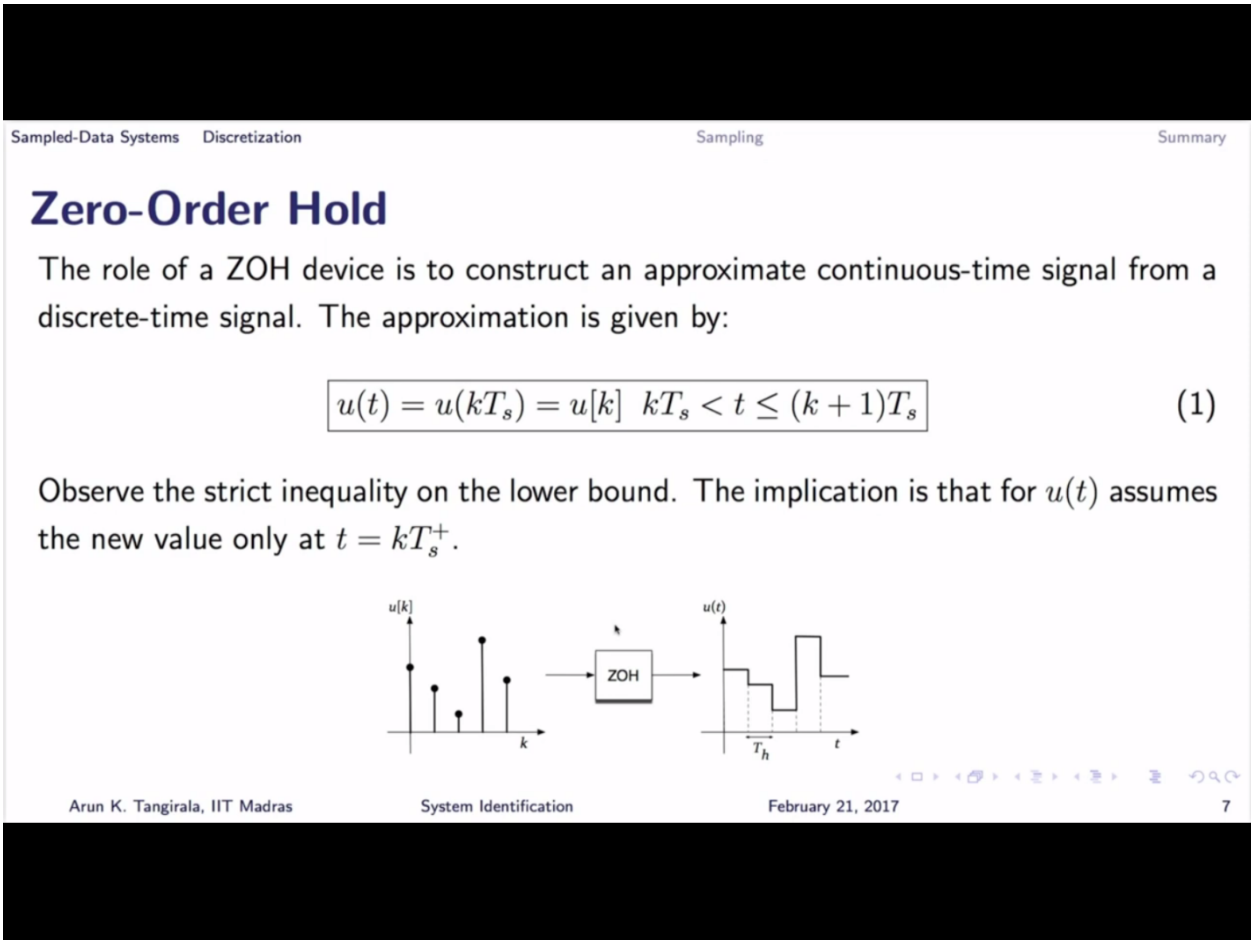
Okay so I am just formally describing what is discretization, but we have already discussed this. It's essentially the procedure of obtaining an equivalent discrete time system for a given continuous time system. So if you are using an Euler's method for discretization, that is also discretization. If you are using some other Runge Kutta method for discretization, that is also a discretization, doesn't matter.

(Refer Slide Time: 00:32)



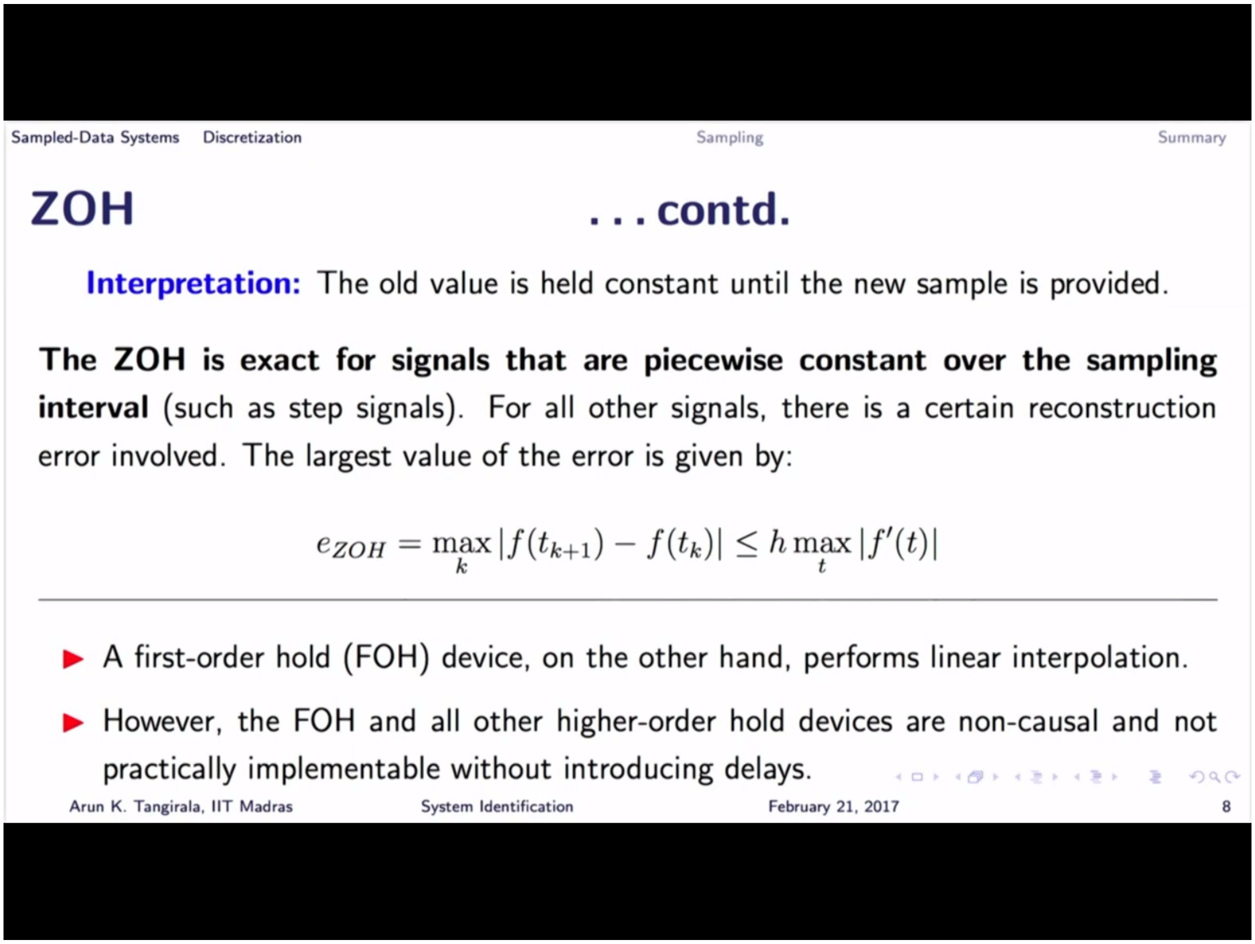
You have to be very clear in every discretization method, what is the error that you are going to incur. Okay, that is one of the things that you should be aware of in any discretization methods. And secondly of course the mapping, when you discretize a continuous time process, how do the parameters of the continuous time system map to the discrete... discretized system. Here in this setup, the mapping, that is in this setup meaning in this kind of a setup here, the mapping or the discretization essentially depends on this D to A converter, because sampling does not really cause any kind of approximation. What does sampling do, it just gets you the value at that instant in time, that's all. There is a continuous time signal, it just gets you the value. It is a D to A converter, that is bringing in some kind of an approximation. Therefore the kind of discretization or the kind of mapping that you are going to end-up with, heavily rests on this D to A converter and you will see that later on when we look at the equations, alright. So that is something to remember. Now generally when you look at D to A converters, this… you will come across what are known as hold devices, the D to A stands for digital to analogue. There are many hold devices, why do we call this hold devices, because they rest on some kind of interpolation, which hold on to some value of the signal, until the next instant arrives. So let us look at a very simple hold device, which is what we will use throughout the course, and which is what is used in industrial operations, most of the industrial operations use this hold device or the instrumentation. What is this hold device, this is called a zero order hold device. By its very name, what it is doing is it is holding on to the signal value, until the next value arrives, that is how it constructs an approximation. So you look at the schematic at the bottom and what the ZOX receives is a discrete time signal, which of course is defined only at specific instance and time, we will assume uniform sampling. We will not worry about the irregular sampling, but of course ZOH can work with both. The output of ZOH is a continuous time signal. How has this continuous time signal been constructed by holding to the value… previous value, until the new value arrives. Mathematically I have given you the definition, U of P is = to U of K times TS for all times between KTS and K+1TS. So if you look at it… this equation carefully, the left hand side for the interval of time is a strict inequality and the right hospital side is a simple inequality, right. What does this tell you that there is some Ipsilon time you can say delta… delta time involved in switching over to the new value. What this says is until K+1 TS U of T will hold on to the value at KTS and note is in notational difference here when we talk of absolute time, we will use the parenthesis, when we talk of the sampling instant we use the square brackets, right. So what this means is in the next instant that is from K+1 TS to K+2 TS, after K an Ipsilon time or a delta time after K+1 TS it will switch over to the U K+1 and remain at that value until K+2 TS. This should give you an idea as to why ZOH can cause a delay. This straight away doesn't give you the reason, but it does give you some idea as to why we experience a unit delay in all sample data systems, what are sample data systems, the one that you have seen earlier. This is called a sampled data system, consisting of a hold device and a sampler. All sample data systems that use a ZOH will result in a unit delay and we will prove that very soon. Okay so this is the mathematical distribution of a ZOH. Obviously what the ZOH is doing is constructing an approximate continuous time signal, right. If I give this discrete time signal, there are infinite number of continuous time functions that will satisfy this UK of the infinite, it is picking one of them, the simplest one. Clearly it results in a error.

(Refer Slide Time: 05:48)



That is I can start with some U of T sample it and give you UK, ZOH will give me this. Whatever, the output of ZOH is need not match with the U of T, that I begin with. So obviously there is going to be some error and you can show that this error is bounded, it's bounded above by a certain quantity, H here corresponds to the sampling interval and F is a general function that you are looking at, that is general continuous time function that you are looking at. So the error depends on the maximum value of the derivative of U of T or F of T, times of course the sampling interval. So what does it tell you for a given function, if I keep increasing the sampling interval what happens to the error, or at least the bound, upper bound, it keeps growing naturally, right. As I keep increasing the sampling interval, the error bound keeps increasing, so you should expect the error also to increase.

(Refer Slide Time: 06:59)

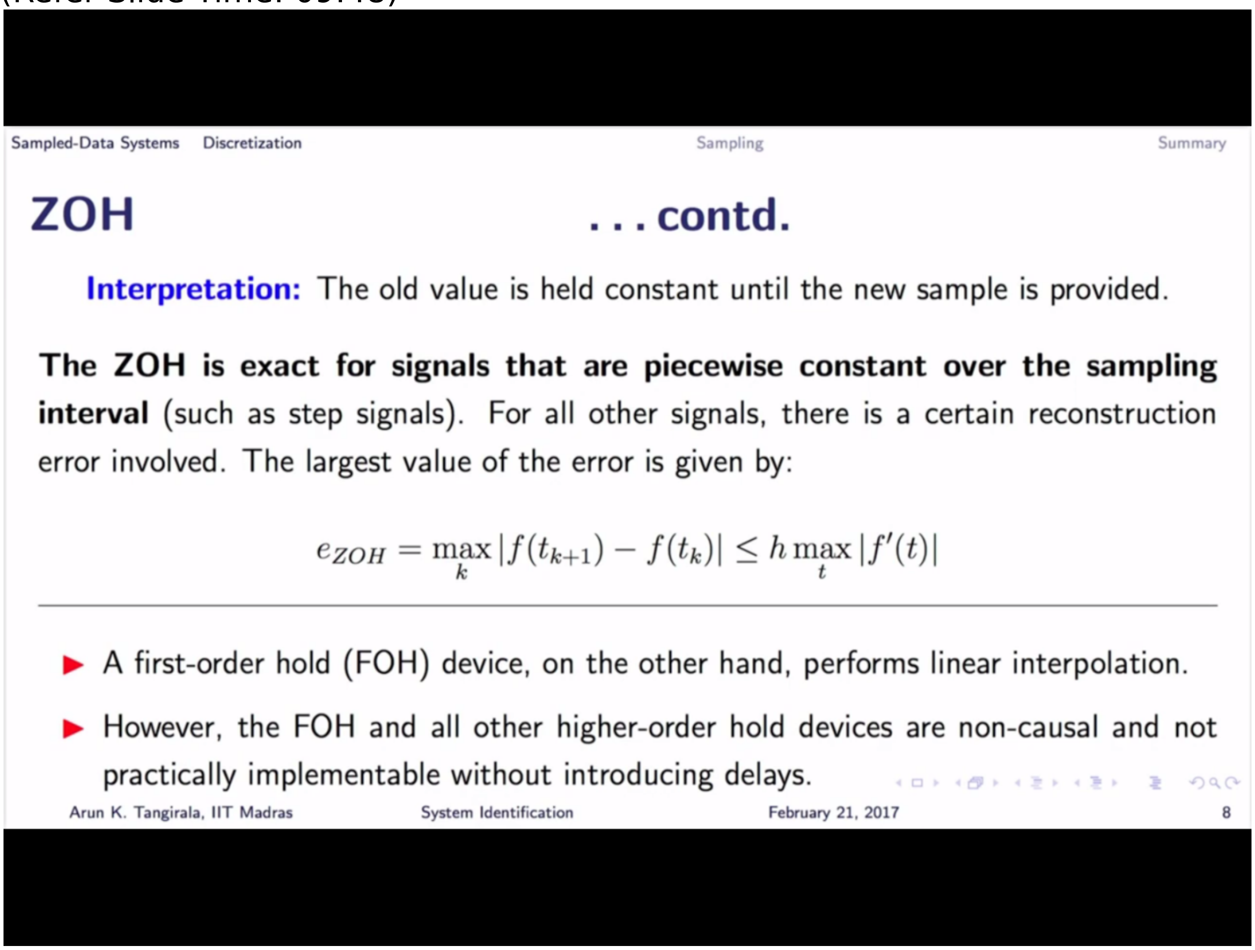


Obviously the reverse is also, converse is also true, as I keep decreasing the sampling interval, I will make lesser and lesser error. Now surprisingly, there is a class of signals for which this error in approximation is 0. Do you know what is that signal? For which class of signals, the ZOH gives you exact reconstruction? Sorry, which

(inaudible)

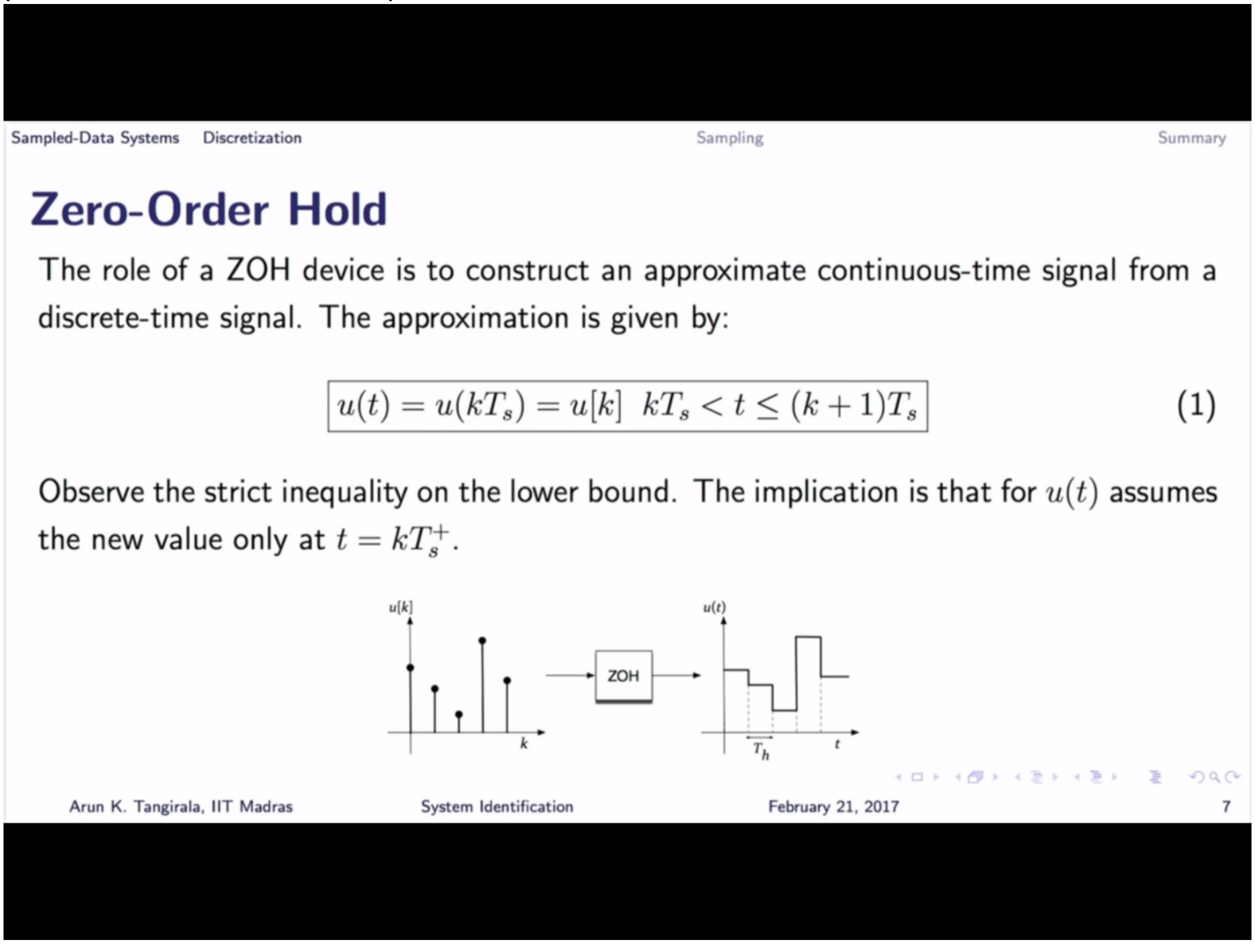
No I am referring to which continuous time signal for which the ZOH will give you correct recovery. So I start with the continuous time signal, I sample, and I give it to ZOH, it will recover exactly that, the step signal, right. Of course you can also say in the continuous time it could be piecewise constant and so on, that’s fine, but when you look at the general class of elementary signals during it, for a step signal the ZOH gives you the correct recovery. In other words, the discrete time step translates to a continuous time step, right, because of the ZOH operation. I can start with a continuous time step, sample it, give it to ZOH, it will exactly recover that fully. You can also say I can start with a piecewise constant signal, right and then give it to ZOH, it will exactly recover, but it will only exactly recover, if the original signal is piecewise constant over that sampling interval. If I change the sampling interval, the recovery is lost, right. Whereas for the step, the error is 0, regardless of the sampling interval. That is the only signal for which regardless of the sampling interval, the ZOH will exactly recover the continuous time signal and therefore the ZOH discretizations are called step discretizations, step in variant... sorry.. step invariant discretization. We will see that shortly. You understand, every hold device will give you an error and it will be exact for some situations. ZOH gives you an error in the approximation, but it is exact for step like signals. And that is why ZOH based discretizations are called step invariant discretizations.

Now you can also think of a first order hold or a second order hold and so on that 0 order, essentially refers to the order of the polynomial that you are fitting between two instants, right.
(Refer Slide Time: 09:48)



You are fitting a flat line, it's a 0^{th} order degree kind of polynomial. If you were to fit a first order polynomial, then here between two instants you would see a straight line, but then there is a problem there. I need to know the future sample to be able to construct that. You can say no I can rely on the previous one and do an extrapolation, that will introduce an artificial delay. So to avoid all this difficulties, people just work with ZOH, because it is a more simple and practically implementable ones, that is why you will see ZOH being very popular, right. The first order hold gives you better approximation, agreed.

(Refer Slide Time: 10:30)

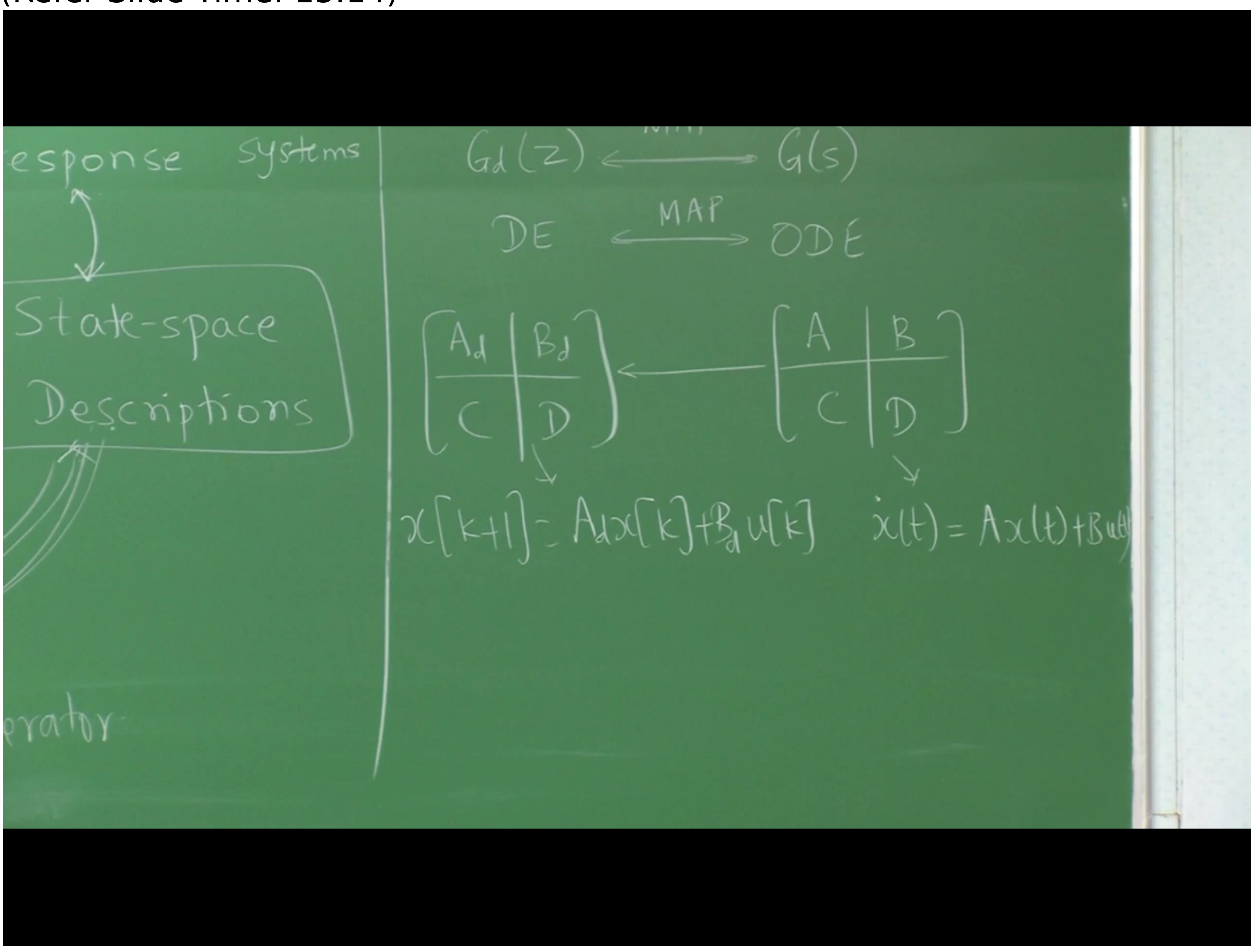


but it is non causal. I need a future sample, future observation, and if I want to work around the non causality part, then I have to introduce a delay. So we will not look at FOH. Now let's understand very quickly, the rest of it is math and it's very straight forward. This math has no complications at all. Again this is another kind of discretization where you know that you are working with a ZOH. Our goal is to arrive at G of Z starting with G of S, you can say that way or you can say that I have a contiguous time state space model and I would like to arrive at a discrete time state space model. For a specified sampling interval, given that you are going to use the ZOH. Why did I retain C and D as they are, and I only changed the subscripts on A and B.

(Inaudible)

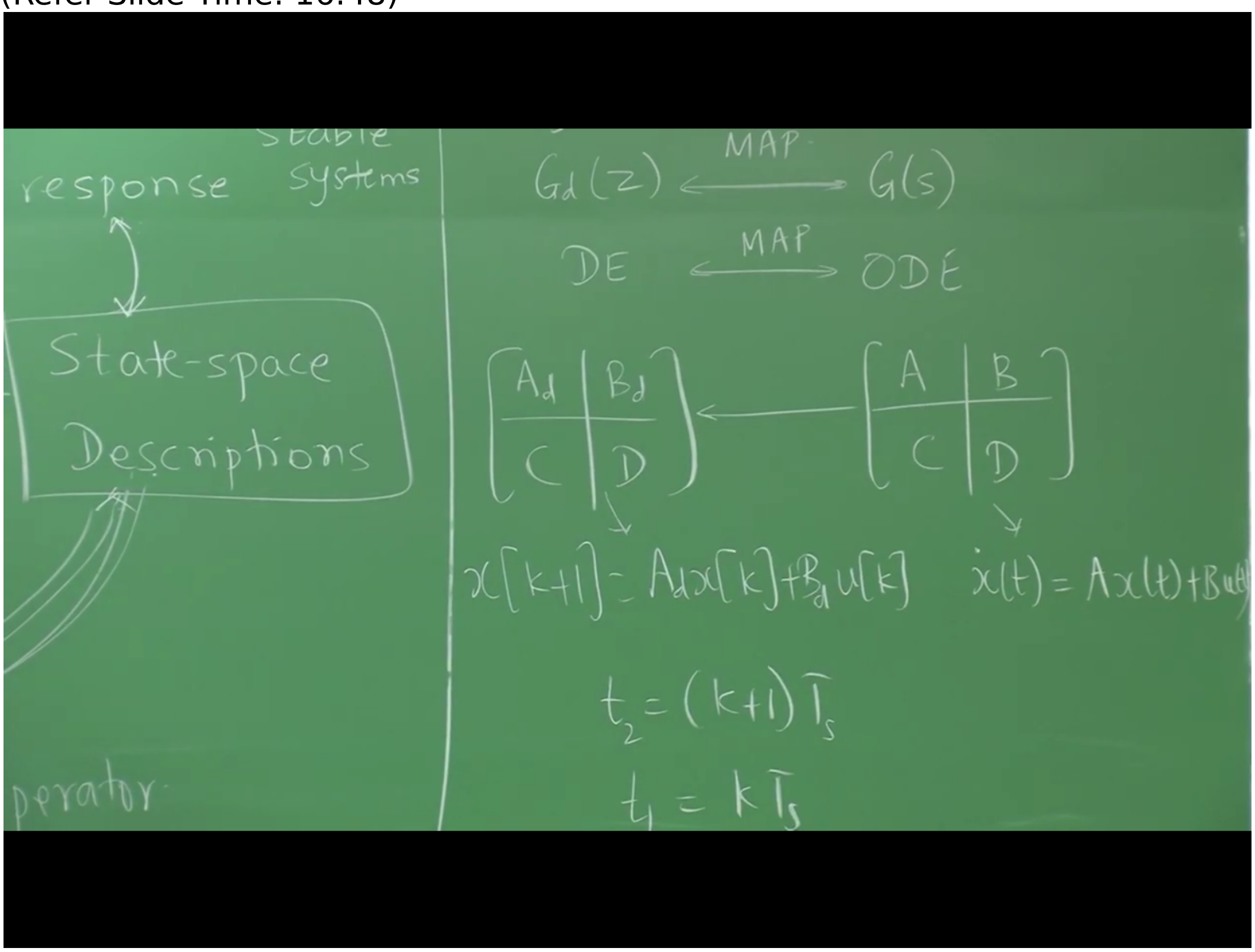
Correct, so C and D are part of the output equation, which is algebraic and an algebraic equation in time holds at all times. This is a state space model in

discrete time. This is a state space model in continuous time, right. What does this state space model do, it relates states at two successive sampling instants, correct. So if you look at the state space equation, you would have XK + 1 = AD XK + BD UK, this is for this model, whereas for this model, you would see an OD, first order OD, right, couple first order ODs. The continuous time state space model tells me, gives me the evaluation equation in the form of ODs, the discrete time state space model tells me how states at two successive instants are related. Now the question is given the continuous time state space model, how do I derive a relation between two sampling states at two successive sampling instants. It's very simple. You start with a continuous time state space model.
(Refer Slide Time: 13:14)



What do we want, we want the relation between states at two successive instants in time, which means I have to write, let's say, let's look at this here. This… this is an OD, suppose I give you a first order system, it's going to be a single state equation. And I ask you from this OD that I have given you, derive a relation between the states at two instants in time. How do you… how do you proceed? What is the first step? Suppose I ask you, what is the solution to the OD, you can do that, right at least you know how to do it. I am not going to ask you to do it right now, but you know that there exist a way in which you can write a solution. What will that solution get you? It will tell you what is the value of the state at any time D, starting from some initial condition and for an input profile. Given that solution, can you derive this relation? How will you do that? Well write your solution at two different instants in time and then connect them. It's as simple as that. There is no complication there at all, if you are thinking there is a complication, there is none. And we will go through that exercise now. So given a continuous time state space mode and straight away writing the solution to the state space model, that relates the states at two different ins… the instants. They need not be regularly spaced. At one time T1 you have state X at T1, and another time T2, you have state X ray at T2. So given this continuous time state space model, let's look at the equation, state equation alone 2A, it's very easy to derive this solution. You will find this solution everywhere and you should be able to recall this from your calculus, courses on calculus. You can either use Laplace transform method or use the standard OD solution method to write this solution. What is this equation telling me, it is telling me how X at T2 is related to is X at T1.

(Refer Slide Time: 16:48)



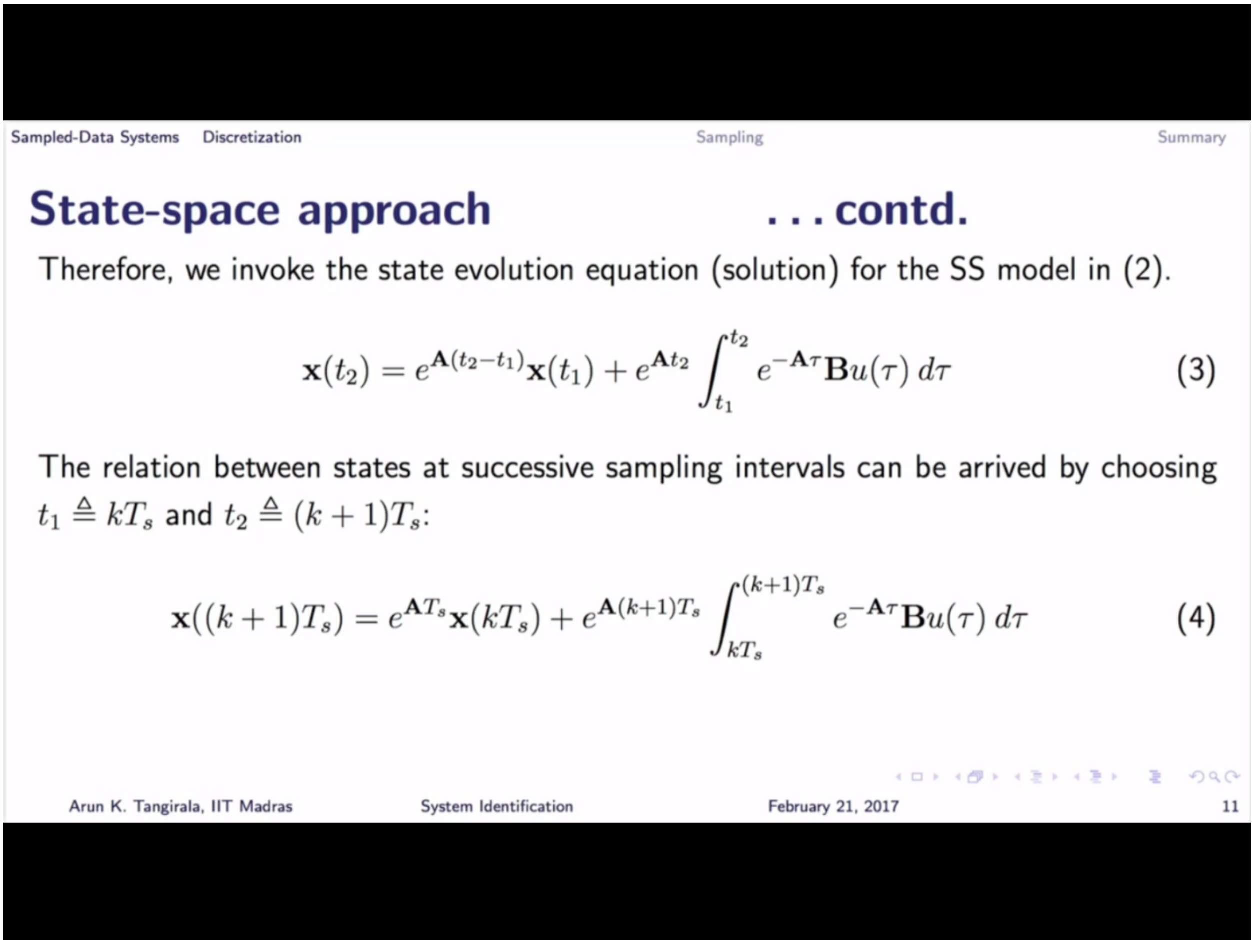
Suppose I am solving an initial condition problem, then T1 is 0, T2 is some time D, right. Suppose I am given some other condition, not initial condition, boundary condition, whatever, some other condition, this solution is useful. So it's a very generic solution. How do I use this to arrive at the discrete time state equation. This should be easy, Taylor's series, why? Why do you bring in Taylor here. We don't need any Taylor, carpenter, no one.

(inaudible)

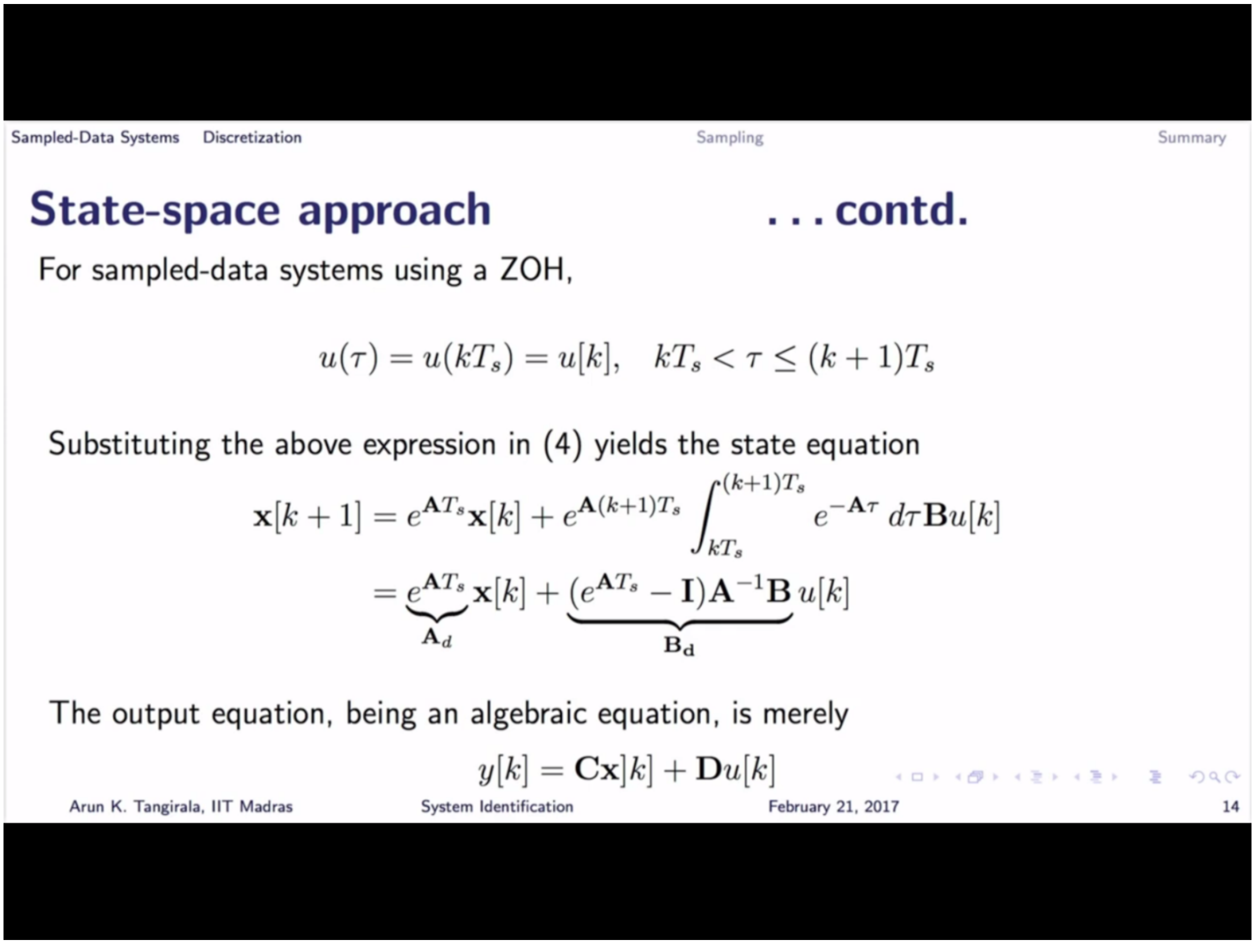
Well you want to be more correct. T1 is K times TS, because T1 is absolute time and T2 is K+1 TS, that's all. You don't have to really Taylor anything here. So T2 is, we said T2 as K+1 TS and T1 as KTS, because that is what we want, that's the relation that we want. So we do that and substitute for T2 and T1 in this equation, right. What do you get. Straight away you get AD. Can you pick AD from that equation, A to the…

(inaudible)

That's it, that's your AD. So we have straight away mapped the state, the transition matrix for the continuous time process to the transition matrix for the discrete time process. A maps to E to the ATS, what is TS, it's a sampling interval. And don't think that this E to the TS is your regular exponential, it's called matrix exponential and I will talk about that shortly, either today or tomorrow. But let us complete the picture now, we want BD, that's not so obvious from this equation, right, because the second expression here is not in this form. You are able to… you are able to pick up… point out AD, because the first term is in the standard form. Now what do we do. Now assume 0 out and bring in the 0^{th} hold, that is why I said until this point you have not made any approximation at all. this solution is valid for all processes, continuous time process, continues time process. Now depending on the hold U of Tau, which is Tau is a dummy variable, between two sampling instants can change. 0 order hold is like a "laddu" it's a sweet.
(Refer Slide Time: 18:45)

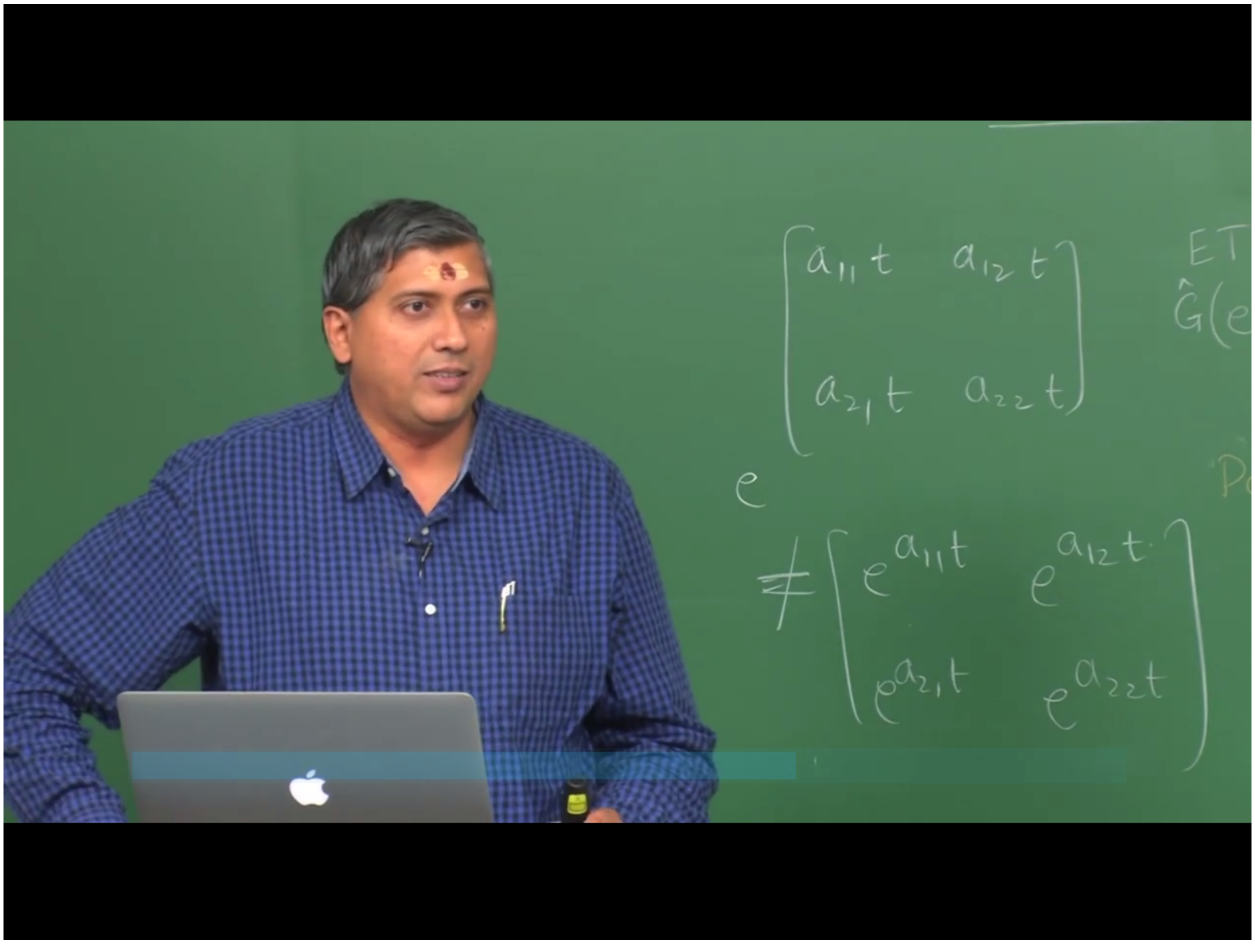


You know, It's so simple, it simply says, U of Tau is a constant between two sampling intervals… to sampling instants, which means it falls… it falls out of the integral. What is that value, that is equal to U… U at KTS or UK, right. So we just bring that out of the integral and quickly simply that integral to arrive at P, what a beauty, right. It's very simple. Straight now I have expressions for AD and BD. So first I compute AD and then I can compute BD in an easy manner. Yes there is an inverse of A involved and if A is singular, you may think there is a problem, but you can actually show that there is no problem, okay. If A is singular, A inverse may not exist is a concern, right, but that’s not an issue. We will show that, in fact you can see that it's not an issue by first understanding this matrix exponential. How is this matrix exponential defined? The same way as a regular exponential is defined. That is now we bring in Taylor, okay. Now the Taylor's series, this will come in.
(Refer Slide Time: 20:11)



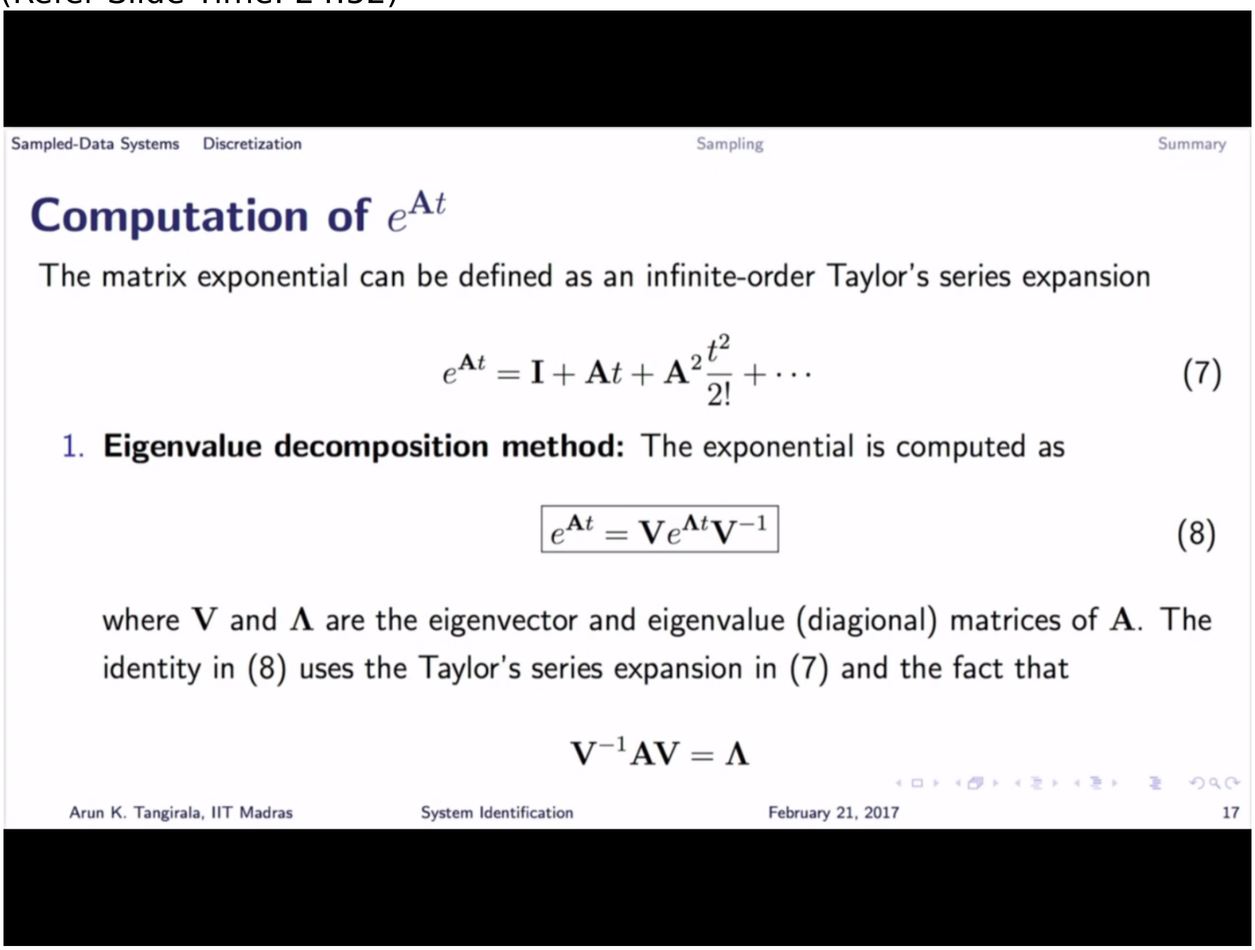
So let's define, this is generally now giving you the mapping under ZOH, you have to remember. If the hold device is changed, you have to go back to this equation, AD remains the same. The B changes, right. So what is this matrix exponential, it is this matrix exponential, it is this infinitely long Taylor's series expansion. Now we invoke Taylor and write it as I+AT+A square T square/2 factorial and so on. Generally the misconception is, matrix exponential is simply the exponential of the elements of that matrix, that is not, right. The other... the other possibility for us to think of is, I write A times T, A is a matrix and I write... So if A was a square matrix, then I would have AT as A11T, A12T, let's say a second order system, A21 times T and A22T. And we may think this exponential E to the this could be simply as A to the A11T, E to the... sorry... E to the A11T, E to the A12T, and so on. Unfortunately this is wrong.

(Refer Slide Time: 21:35)



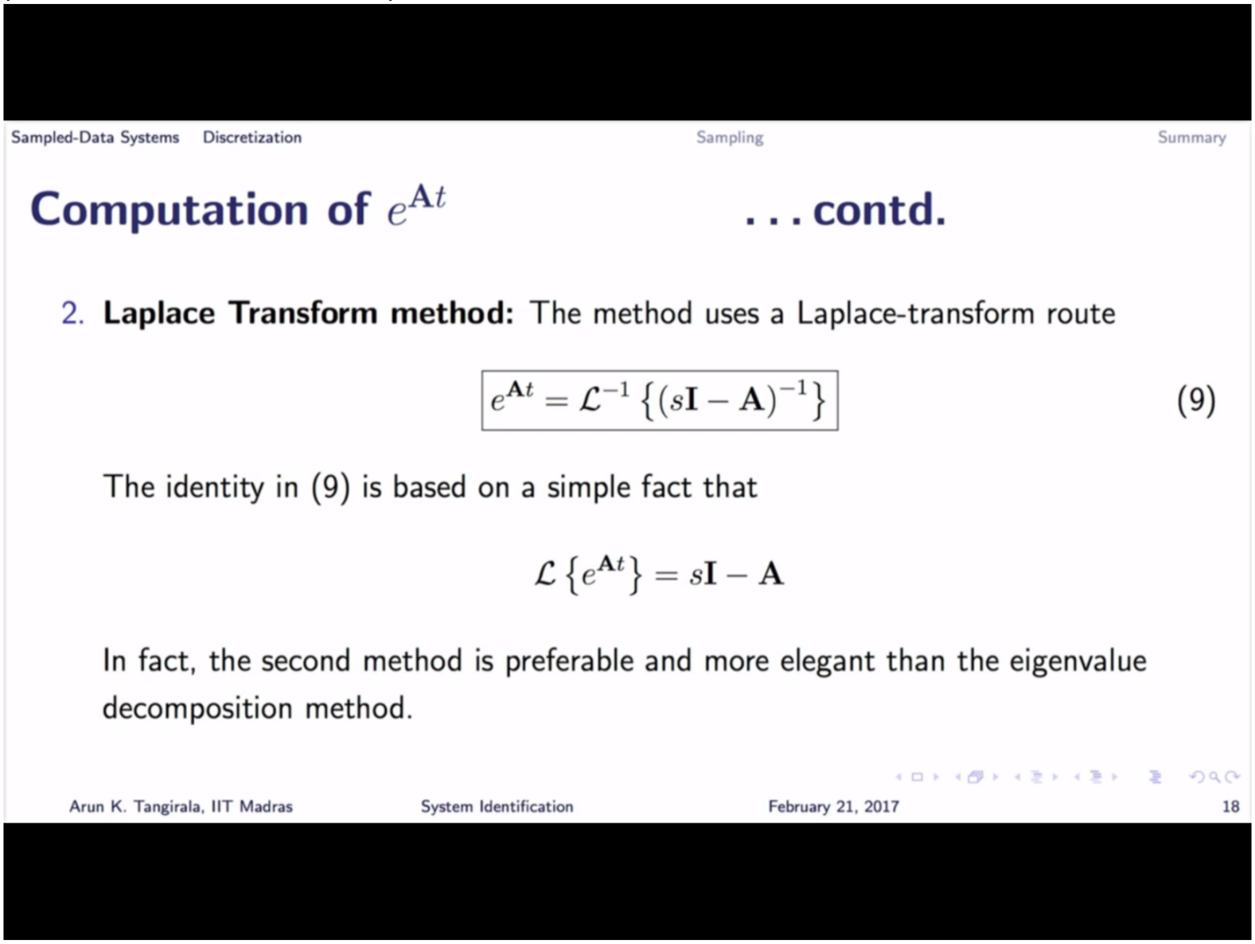
When is it true, only for diagonal matrixes, okay, so matrix… in fact now computing this matrix exponential, I am always reminded of this fantastic article by Cleve Moler, I don’t know how many of you have heard of Cleve Moler, have you any, has any one of you heard of Cleve Moler? Well he is a founder of Mathworks, he is the maker of MATLAB, and he is the architect of MATLAB. And he is a fantastic mathematician as well. There was an article by him long ago on 19 dubious way of computing matrix expansion. It’s a very nice article, you should read that. We will not learn those 19 dubious ways, we will only learn two correct ways of computing the matrix exponential. Because for them, it all… it's all about computing this in as précised, numerically précised manner as possible, and that depends on the approach that you do. So how do you… as at least by hand compute matrix exponential, right. The rest is done by MATLAB. You may say, well when MATLAB is there, why should I worry, but MATLAB is not going to be there all the time with you. You should also be comfortable calculating by hand. So one of the ways of computing this matrix exponential is through the Eigenvalue decomposition method, which says that E to the AT can be written as a product of V times E to the Lambda T times V inverse. What is this big V, this big V is a matrix of Eigen vectors of A, remember A is a square matrix. So you can calculate Eigen values. This Lambda, big Lambda is a diagonal matrix of Eigen values of A and basically you can arrive at this result by starting with this Taylor series expansion and writing the fact that V inverse AV is Lambda, okay. So you can either multiply... you can multiply, pre-multiply, and post multiply with V inverse and V on both sides here and get to this identify here or you can write A itself as V times Lambda times V inverse, that is your standard Eigen value decomposition. So in place of A on the right hand side of equation 7, wherever you find A, plug in V Lambda V inverse. When you do that you will see the V... V falls out to the... as a left factor and V inverse falls out as a right factor and you will be ending... ending up with I+Lambda T+Lambda square T square/2 factorial and so on, which is E to the Lambda T. The difference between E to the AT and E to the Lambda T is, what is it? E to the Lambda T is very easy to compute, it's just a elements... exponential of the diagonal elements that is the advantage. That’s it.

(Refer Slide Time: 24:52)

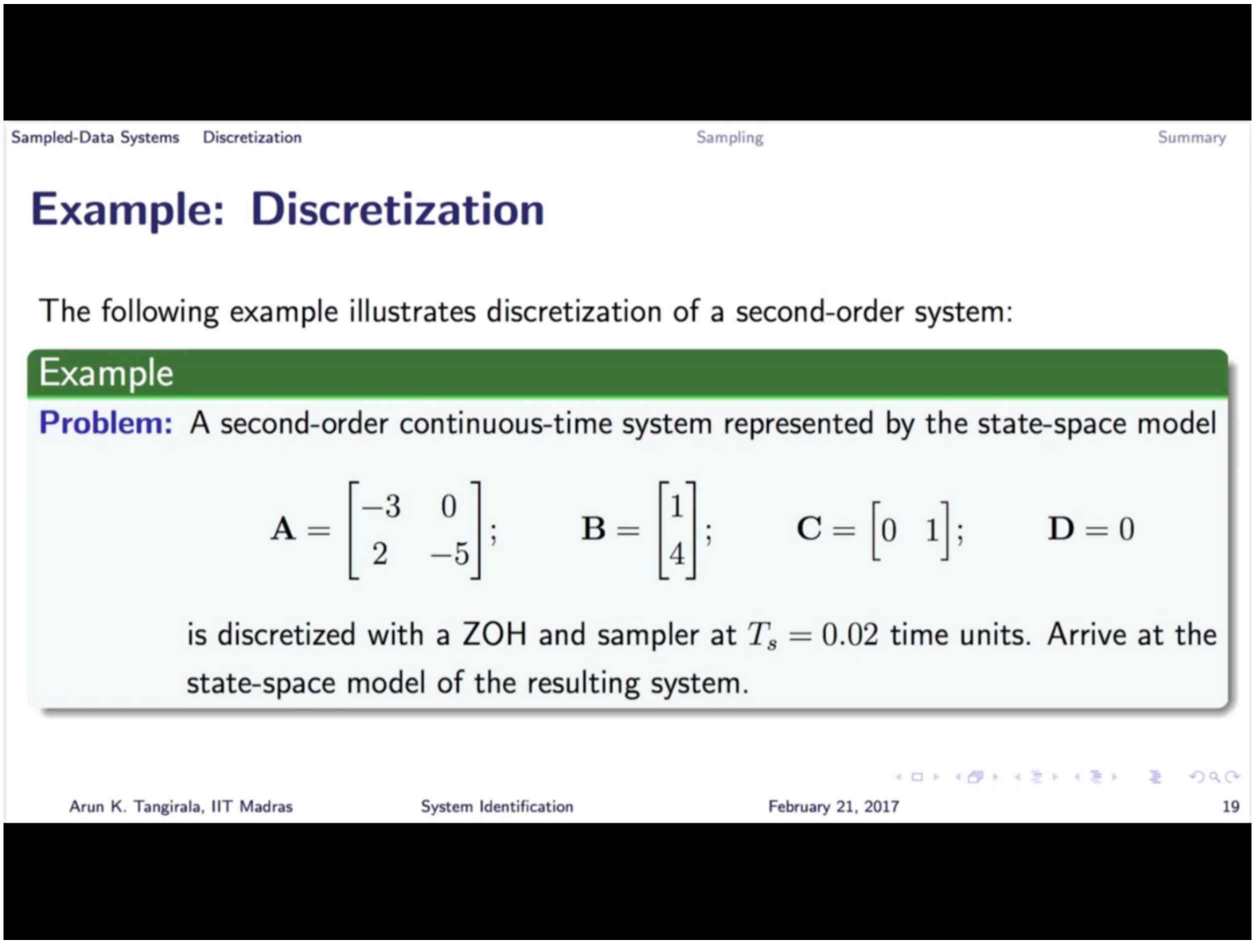


So that is one method, the other method is the so called Laplace transform method, which is also very eelegant way of computing the matrix exponential. Here you make use of this identity from Laplace transforms, which says that the matrix exponential is the inverse Laplace of SI – A inverse. You can easily verify this for a scalar. If A is a scalar, we know that

Laplace transform of E to the AT is 1 over SI minus A... S minus A. So this result is an extension to the matrix case.
(Refer Slide Time: 25:21).

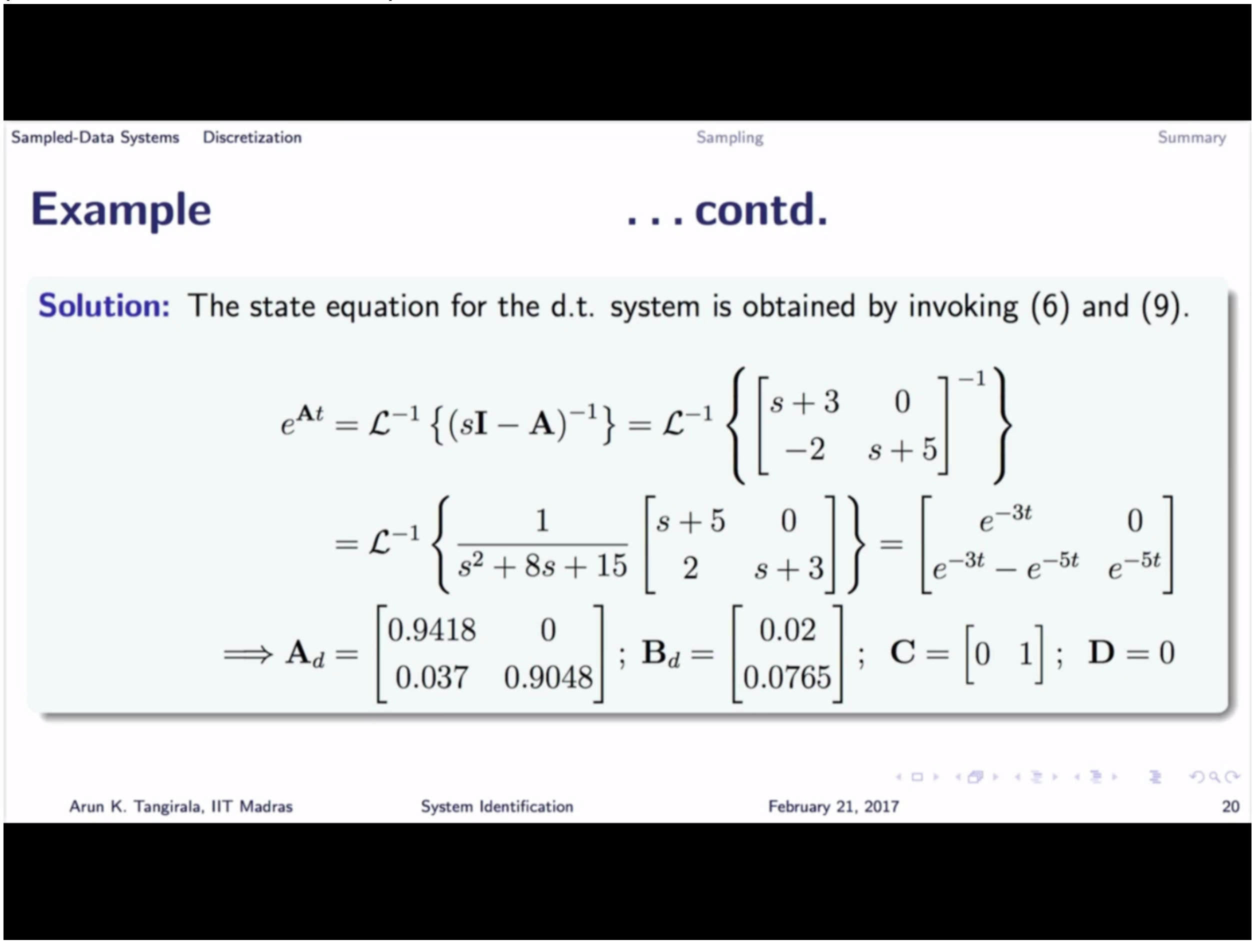


How do I make use of this result? The idea is that first I go into the... I will show you in this example and then we will conclude today's class and then we will complete discretization tomorrow. So let's take this second order state space model and I will show you how the Laplace transform method works. Discretized with a ZOH, the sampling interval has to be specified.
(Refer Slide Time: 25:58)



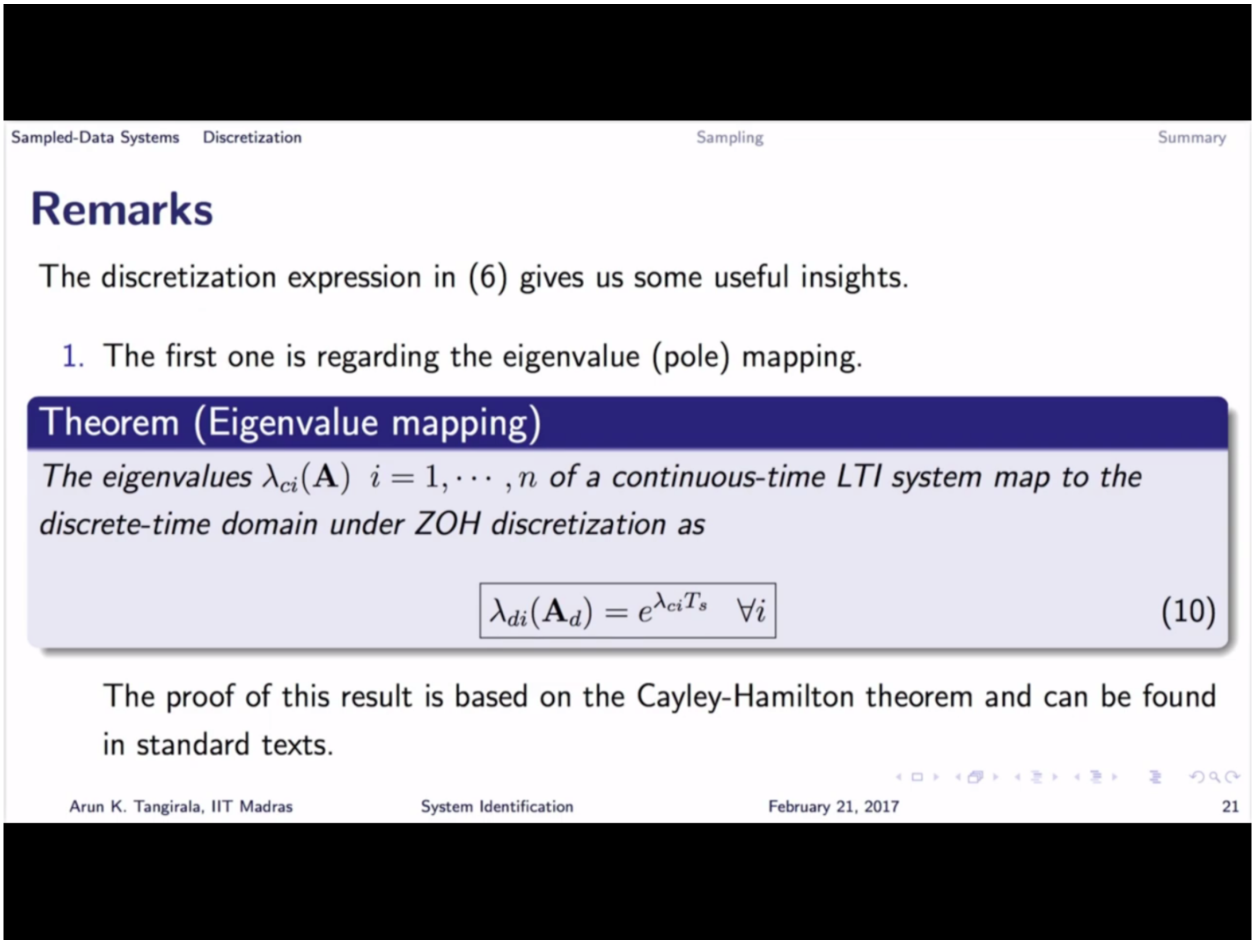
Let us look at the state space model. I am not going to go through the Eigen value method, because that's very easy. You can actually take these values and do your calculations in MATLAB. Let's look at the Laplace transform method, which is elegant. All you do is, given A construct SI – A inverse. For 2/2 its fairly easy. So it's all symbolic, here SI minus A inverse, the only thing that you have to remember is how to compute the inverse of a 2 by 2 matrix, which I suppose you all should know, else we have to recheck your admission criteria. Alright, so that is your inverse, now the nice thing about taking the Laplace inverse of a matrix is you can take the Laplace inverse of individual elements of a matrix and that is why it's advantageous. Earlier in the Eigen value method, the advantage was that you are going to deal with E to the Lambda T, where Lambda is a diagonal and you could take the exponential of the elements. Here the advantage is, you can take the Laplace inverse of the individual elements and that you can do, you can look up any standard table and come up with the inverse Laplace of each of this elements. That's it, straight away you have E to the AT. And since you are given TS, plug in for TS here and get your AD and BD, alright. In this case, it so happens that the structure of A is preserved, but you have to think, whether the structure, if A has... that is the continuous time state space model has a certain structure, will AD also retain the same structure. Think about it and you can answer me tomorrow.

(Refer Slide Time: 27:55)



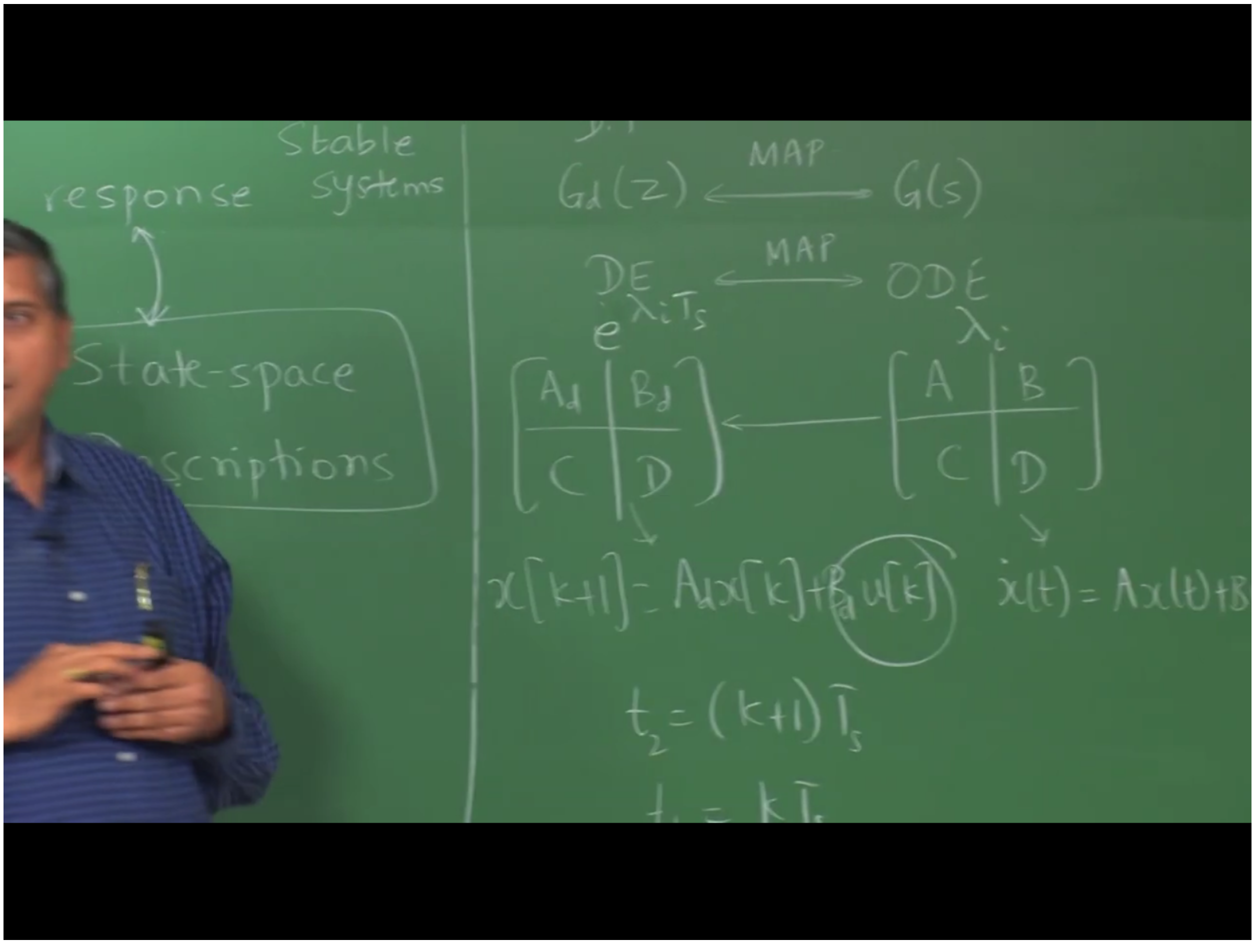
One of the things that we can straight away infer is the Eigen values of AD, because now I am very interested in stability preservation. My sampling should... should not spoil stability. If the continuous time process is stable, the discrete time... discretized process should also be stable. For this I need to know Eigen value mapping, and Eigenvalue mapping theorem is that if the... if Lambda's are the Eigen values of the continuous time LTI system, then the Lambda's of AD are simply E to the Lambda TS. This is... the proof can be obtained by using, what is known as the Kelly Hamilton Theorem. What is Kelly Hamilton Theorem? Satisfies its own characteristic equation. So using that you can prove this result that the Eigen values of the continuous time system map to... we set out to map models.

(Refer Slide Time: 28:58)



But now we realize if Lambda I is an Eigen value here, E to the Lambda I TS is the pole here.

(Refer Slide Time: 29:05)



So do you think stability is preserved? Because continuous time systems are stable, if the Eigen values are strictly negative, at least the real part of the Eigen value should be in the left off plane, that… they should be negative. And for discrete time systems, they should be within the unit circle, right. So that is satisfied, therefore stability is preserved in this discretization. If the continuous time process is stable, your discretized process cannot go unstable, alright, but you should realize that the pole now in the discrete time depends on two things, the pole in the continuous time and the sampling interval. If the sampling interval is very small, what does it mean and sampling very fast, what happens to the pole in the discrete time, close to the unit circle, is it good? Because we are moving towards the border, line of control, and that can cause a lot of issues in identification. When there is a pole that is very close to the unit circle, small errors in your estimation can actually end-up with an estimate outside the unit circle, that's why it is not good to sample very fast. We are talking of relative only, when you say fast, it is a product that we are looking at, Lambda I times TS. So if TS is extremely small, relative to Lambda I, you are putting yourself on the border and you know border. And you know border, usually it is always busy at firing, so you don’t want to be really a part of that firing. So see how beautifully this mapping has allowed us to quickly comment on what sampling can do to identification to your system and so on. Tomorrow when we meet, we will learn another way of discretizing the system, but there we will be given a transfer function, can I then arrive at the transfer function without going to the state space route. I can always, for a given transfer function, I can convert the state space and then use this result, and then go to the transfer function, that's a very circuitous road. Directly given the transfer function of the continuous time system, can I arrive at the discrete time transfer function and then discuss some sampling aspects and then move on to very quickly the sampling theorem, that way we would have completed our discussion on discretization.