CH5230: System Identification

State Space Representation 3

Very good morning. Okay. So let's get back to the discussion here. We have been discussing state-space models. Yesterday we learned at least two different ways and two different perspectives on state-space models and two different scenarios. You can think of it that way. And in that process we learned also how to convert a difference equation form to a state-space form. We are going to elaborate a bit more on that today. Now having said all of this you should also realize that in many, many situations state-space models just naturally arise, particularly when you're writing first principles models.

What we mean by first principles models? As you know is, when you are developing governing equations based on conservation laws, fundamental laws and laws of conservation of mass, conservation of energy and when you do, a force balance equation and all of that. When you do that you do get naturally a state-space model, where these states are the variables that change with time. And as always the states are those variables that are governed by far started differential equations.

So if, for example, if you take a spring mass system like when all of us are familiar with spring mass system. Very simple, spring mass system, assume no damping and if I were to write the equation for the displacement what would it be. So if I were to write the equation for the displacement relating to the force acting on the mass, you know the standard spring mass setup up. What would be the equation for the displacement? Right?
So what would what would be the equation? Can you spell it out? Assume that the mass body has a mass m and the spring as the spring constant k and that there is a force f acting on the mass. That's it. What about the force? So m x double dot equals. Well, is it going to be minus k x or plus k x? Minus k x plus. Well, assume that the force is actually aiding the displacement. Right? You're going to pull the mass so which means that this force will be responsible for the displacement of the mass. And the spring is going to push it back. Right? And that's why, you have a negative sign.

Now k x or k x dot? k x. So we have this differential equation governing the displacement of the body which has mass M. Now, if you look at this equation, here we have a second order differential equation. This is an equation that we have arrived using the force balance. Right? This is not readily available in the state-space form. You will have to introduce an artificial state and convert this to a state-space form. When you do that, how many state equations would you get? Two state equations. So that's consistent with your second order differential equation.

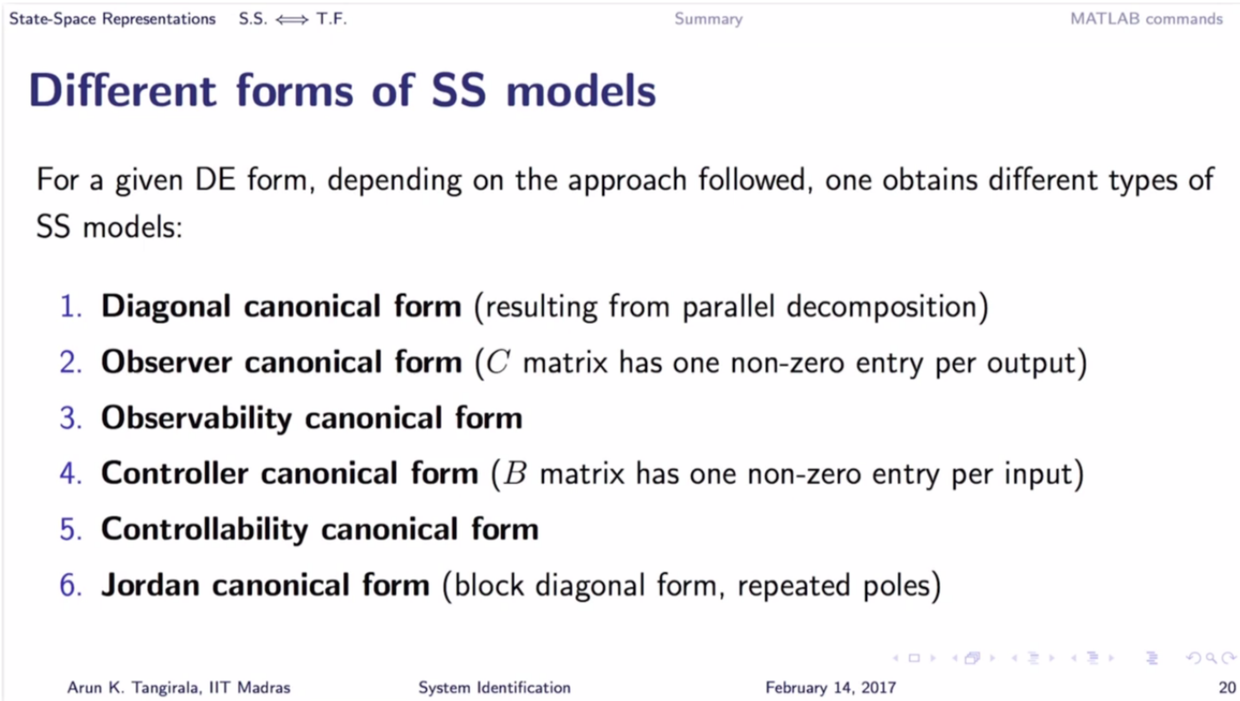
Although here, we do not have state equations in the readily available form. Still you can think of the notional state here and the final states-space model would have been arrived at from first principles

considerations. In many other situations, suppose I take the liquid level system, in the liquid level system, if I were to write the governing equation for level dynamics then you would obtain. So this is for the spring mass without damping for the liquid level dynamics are simpler where you would write dh by dt. If you assume cylindrical tank and uniform cross-sectional area AC and further if you replace the outflow with the constitutive relation that we normally see in fluid mechanics.
This is the differential equation that you get. Now this isn't state equation. Here you don't have to break your head in choosing the state. The state is level itself. Of course you can say state is half the level that is where the non-uniqueness is. In these cases, in fast order ODs the non-uniqueness is only because you have chosen a state by modified by some scalar. So here I have a first principles model straight away giving me the state-space model. You may say again, what is a need for looking at this as a state-space model. We will today argue out, why state-space models are preferred to transfer function models. In what situation, there is not always that state-space models are preferred to transfer function or input-output models.

There are certain situations in which state-space models score over the input-output model. So here if you were to choose this state-space route, then you would say, the state's, there is only one state which is a liquid level and the measurement is simply the level itself. Of course, if you have noise then you would have to take that into account which will appear in the measurement noise. Okay? So the point here is state-space models can arise either from first principles models, when you write governing equations based on first principles or when you have the case of soft-sensing or inferential sensing, or when you are artificially breaking up a higher order system into several first order systems.

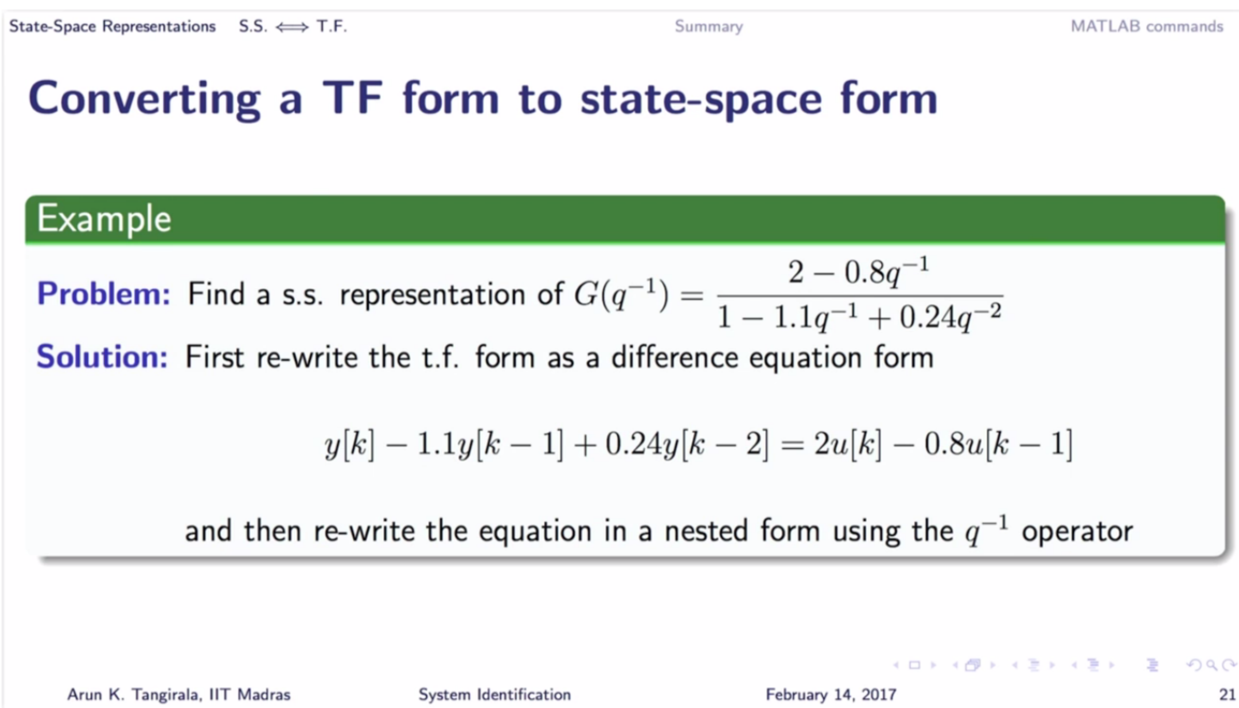
These are the three general scenarios in which you'll encounter state-space models. And when you do that, as we have learned yesterday, you can run into different canonical forms. That means where A, B, C, D matrixes have different structures and depending on the kind of structure they have. And when you can write such a structure, what that structure means, you have different names to this canonical forms. And we are just briefly discussed this canonical forms. We will come back to these canonical forms and discuss a bit more in detail when we talk of state-space identification.

(Refer Slide Time: 07:25)



For now it suffices to know that there are these different canonical forms and each has its own name for a reason. All right.

(Refer Slide Time: 07:38)



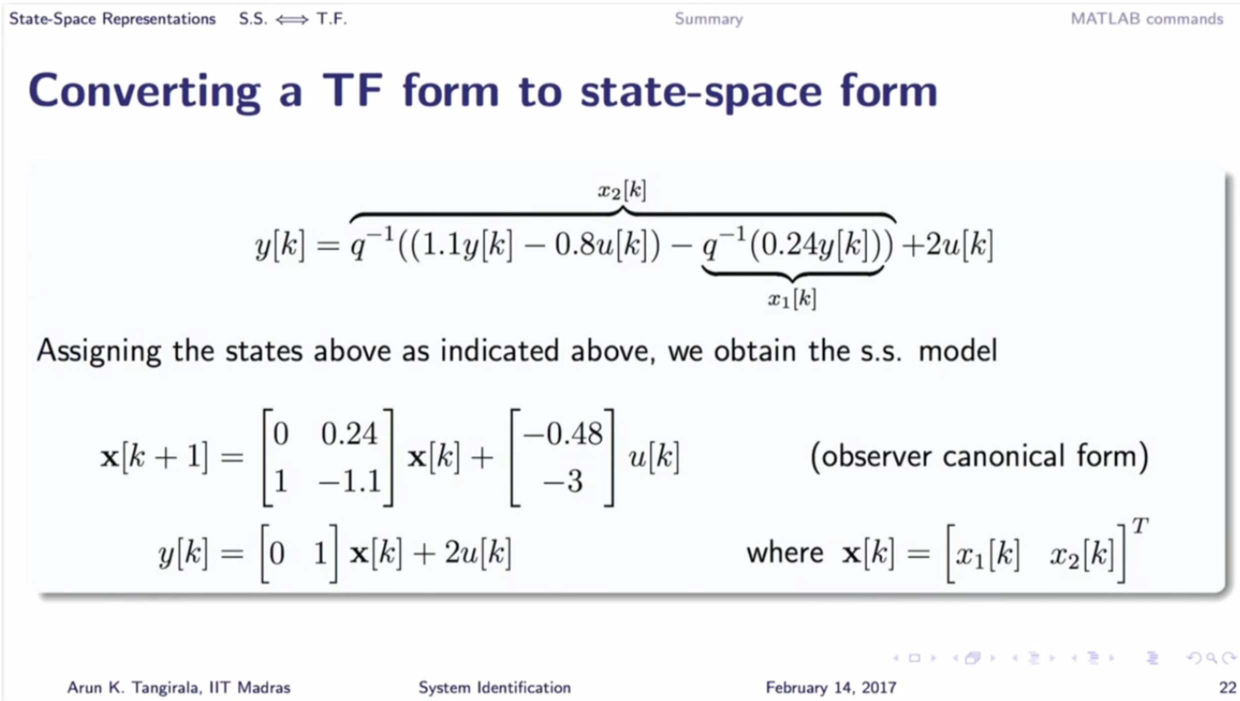
Now, let's move on and look at this example, where we are interested in converting a transfer function form to state-space form. Now this is not new, as I said yesterday. We went to an example, where we learned how to rewrite a difference equation form into a state-space form by introducing artificial variables. The story is the same here. But just so that you understand when you have a transfer function where the numerator dynamics are also present. In yesterday's example, if I can take you back. Here there were no numerator dynamics. There's just a delay.

But suppose I have numerator dynamics then how can I the state-space form. Often students get stuck at some point in converting these kinds of transfer function forms to state-space form because they're not sure how to deal with this u k minus 1 or u k and u k minus 1 and so on. In this transfer function form there are two features that you notice with respect to the input. One, that what is I delay?

Why does it take so long? Zero. So someone said 1, which is not correct. You look at the equation and you should be able to say there is no delay. Right? Which means what you would expect in the states based model? Correct. You should expect to see it feed through term. Right. A direct term, that means a D element is not going to be zero, if you have done correctly, that is how it should come out to be.

Secondly you have some input dynamics, what we mean by input dynamics is there is not just the instantaneous input that's affecting, you have the past input also affecting the output. So there is some input memory. So let's understand how to convert this transfer function form to a state-space form. The first step is to write the difference equation. Okay. So we have written the difference equation form. That's a no brainer.

(Refer Slide Time: 09:54)



Now, what we do is, we collect. We just retain in this difference equation form. We just read y k to the left, on the left, and then take everything to the right. So that's all I have done, as a first step. And the second step is to rewrite the right hand side in this fashion. Where I have collected like terms of y k and U k. When I say like terms at those instance. Right? When I take, for example here, all this terms to the rate I would have y k minus 1, u k minus I, that's a pair.

And then I have y k minus 2 that is by itself and then I have u k, that is by itself. So the idea is to, now write this as a nested form. I'm showing you in terms of equations, many textbooks that spend time on explaining how to arrive at state-space descriptions from transfer function or even, you know, whether it's discrete or continuous. Spend time in showing you block diagrams. Where each block is a fast order system and output of that is a state and so on. And you can draw nice state diagrams which I don't do this in this course, in a dynamic scores such as an introductory course on control. I spend time on showing how you could even draw a blog diagram.

Here we just rely on the equation. The idea is to write this in a nested form and see that here. I have kept the U k outside, because u k is anyway a special one. It's an instantaneous effect on y, straightaway I see the D is 2. So what we are doing is we are writing the output equation first. Right? Even yesterday we wrote the output put equation first and state the way we are able to say what is c what is d. Here I am able to say D is 2. Now, what remains to be found, determine that A, B and C. And that I can do in a number of ways. This is not the only.

What I have done here is, as I said, I have-- Remember that there is a unit delay in each state equation and I have exploited that fact and therefore written this in a nested way. So here I have written this entire thing in terms of the shift operator. So that I can say that the output of this outer one the ratio is one state and the output of the inner one here is another state. Okay. So that straightaway I get my state equations, for example, if I say x 2 k is q inverse 0.24 y k. Then x 2 k plus 1 would be simply 0.24 y k. Of course I have to rewrite that y k again in terms of states. That we will do.

But writing the state equations is easy, because I've already assigned state as the output of a delayed system. That is the idea behind this approach. As I said this is not the only way but this is a nice way of arriving at the states- space model. This leads to one canonical form. That's it. So here, the first state is going to be q inverse 0.24 y k and therefore x 1 k plus 1 is going to be simply 0.24 y k. x 1 k is q inverse 0.24 y k that means, it's 0.24 y k minus 1. Now, since x 1 k plus 1 is 0.24 y k and I need a state equation purely in terms of states and inputs. I have to again come back to this y k. What does this y k telling me? What is a question for y k? What is the output equation?

x 2 k plus 2 U k. That's all. So, all I have to do is, here substitute. Did I get my stated question for x 1. What about x 2 k, in a similar fashion, you can write x 2 k as well. The state equation for x 2. x 2 k plus 1 would be 1.1 y k minus 0.8 U k minus x 1 k. Again I substitute for y k. And then as a result I get this state question on. Clear? So the idea behind this approach is straight away, I am saying that the output. So if I were to look at the output equation here, you see that the output is nothing but a second state plus 2 u k.

If you ignore the 2 u k part that the state is second state is simply the output itself. Right? For the time being keeping aside the input. And then if you look at the state equation it reminds us of the observer canonical form. Okay. So it has a certain canonical form. You can actually write this. Choose your states in such a way that you can get controllable canonical, controller canonical form or controllability canonical form. There are different ways of doing it. But I'm just showing you one way.

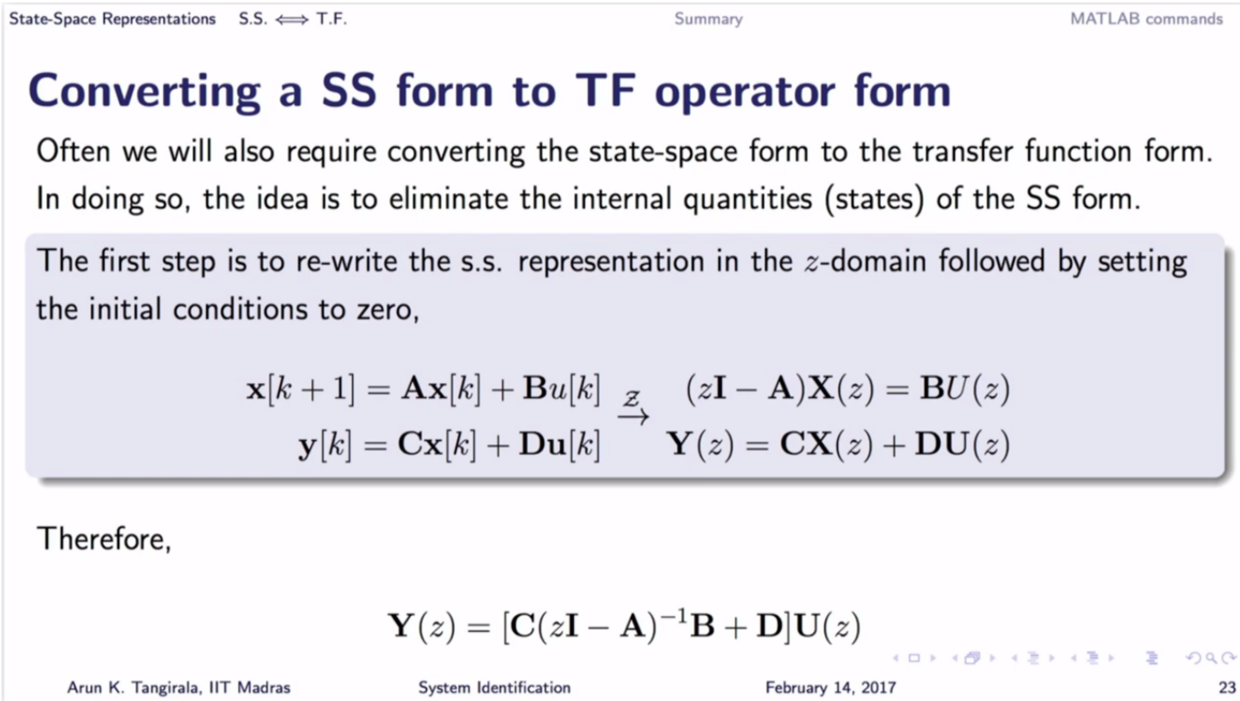
And since this is not an exclusive course on dynamics, we are not going to explore those other option but we are free to choose. It doesn't matter. Essentially, you should be equipped with a method to arrive at a state-space form. Unless a specific structure is asked you should not be worried. All right. At this stage we we're not ready on a specific structure at all. Our task is to learn how to go from one form to the other form. That we have been doing that, when we learn convolution, we learn how to write the impulse response, some impulse response. We learned how to go to step. From impulse response we've learned how to go to frequency vice-versa. Then difference equation forms. So every form that we learn, we are learning how to actually go from one form to another, because we know very well that all of these represent the LTI system. Okay. So any questions on this example?

All right. Now, of course you have comments in MATLAB to do this for you. I will discuss a MATLAB comments. In fact I was also supposed to discuss the MATLAB comments for the input-output domain. I'll do that together. Okay. Now we have learned how to at least one or two different ways of writing a state-space form for a given transfer function form. One way is to factories the G. Where we can imagine a serious decomposition or a parallel decomposition and the other way is to just follow the approach that we just discussed.

We should also be conversant with going from state-space farm to transfer function from. Many items that maybe required. Why are we learning all of this, so that we understand how the parameters of one model mapped to parameters of another model and which will become later on useful in identification. Okay. That is as far as society is concerned. In a pure dynamite course, you're supposed to be well versed just by of these conversions.

Now in order to go from state-space form to transfer function operate, in fact, write operator form here. It should be a transfer function form. I will correct that. If it is an operator form the method doesn't change much, in place of z you use q. It becomes easier. So here what we're doing is, we are taking the states based model writing them in z domain, assuming initial conditions to be zero.

(Refer Slide Time: 18:10)



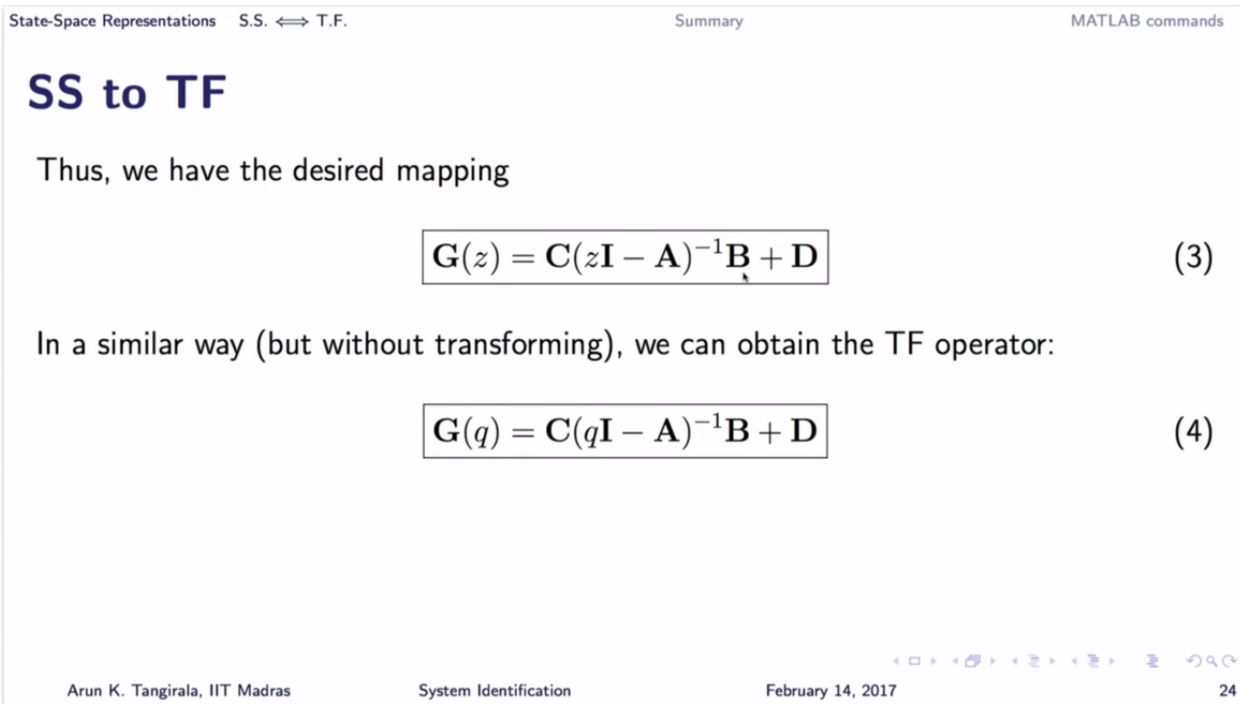
If you were to use a shift operator you don't have to make any such assumptions. You can directly write in terms of q. Right? So in place of z you would have q That's all, and in place of x of z you would have simply x k. Clear? Because when we write down so function operator forms, g of q inverse relates y k to U k. It doesn't relate y z to u z. That's it. So these are first order differential equations and you have to use the property of the z-transform. Which property would you use here for converting the state equation?

Signal advanced by one unit. And that's a property that you use, but we had given that the initial condition is zero. Therefore you have z I minus A times x of z equals B times U of z. The output equation is fairly straightforward. It is simply an algebraic equation. There's not much to worry about. Therefore y z is simply C times x z plus D U z. So in writing these equations I'm assuming a very generic system. I have multiple inputs, multiple outputs many states and so on.
That is why you see boldfaced notation for all the variables involved there.

So Y of z for example is a vector of z-transforms and likewise U z as well. This u is still in scalar mode. It's yet to wake up but we will correct that. Since I want to write a transfer function form. The goal is to eliminate the state. And I do that by simply writing x of z as z I minus A inverse B times U of z. And substitute that in the output equation to get this relation here. So this entire thing here C times z I minus A inverse B plus B. The order of multiplication is very important. Is your transfer function. That's, that is what your transfer function is. C times z I minus A inverse B plus D.

And you should check if dimensionally it is consistent. Right? Remember will--
What is the dimension of G for a multi input-multi output system? If I have N Y outputs, N U inputs, then what is the dimension of G, N Y by N U. And you should check indeed you get that. C is going to be N Y by N X and B is going to be N X by N U. So this part is going to N Y by N U. D by definition is N Y by N U.

(Refer Slide Time: 21:05)

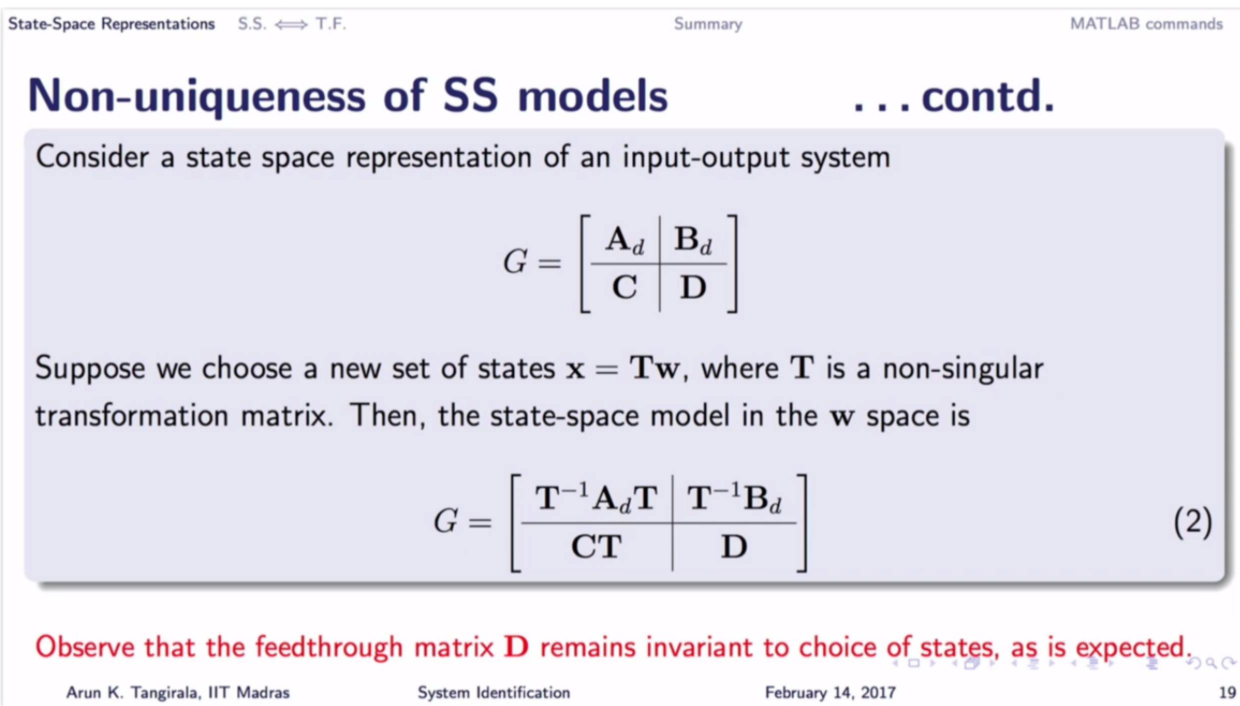


So here G of z is in fact N Y by N U. And this is an expression that you're just supposed to remember as well as you remember your name. It's. It has, in fact if you forget you can always derive it on the flight. It just takes a few seconds. As I said the transfer function and the transfer function operator don't differ much. But you should remember the subtle difference that we talked about between these two. That's it, so this is the expression that is used in all these packages that convert state-space models to a transfer function forms. Whereas there is no formula for going from transfer function to state-space form, you have if you choose to work with a canonical form.

If you recall, yesterday I showed you a few canonical forms, in those canonical forms you had some coefficients As and Bs and so on. Where are those As and Bs coming from. They are coming from your transfer function. So that is like a formula. All you're going to do is take the coefficients of the numerator and denominator. And then place them at appropriate places depending on the canonical form. So that is a formula that is that. That's it. It's a direct placement there's no formula. Just take those coefficients and place it. Therefore, you should be aware in any software package that you use that converts a transfer function to state space form as to what canonical form it returns. So as a simple exercise, go and figure out in MATLAB when you use TF to SS. TF number 2 SS. Ask or search in the documentation what canonical structure it returns. Typically it gives you two or three different options. One of them is called a model form, this model form is nothing but the diagonal form, diagonal canonical form then there is one or two other canonical forms. Go and read so that you are aware as to what canonical fancied returns.

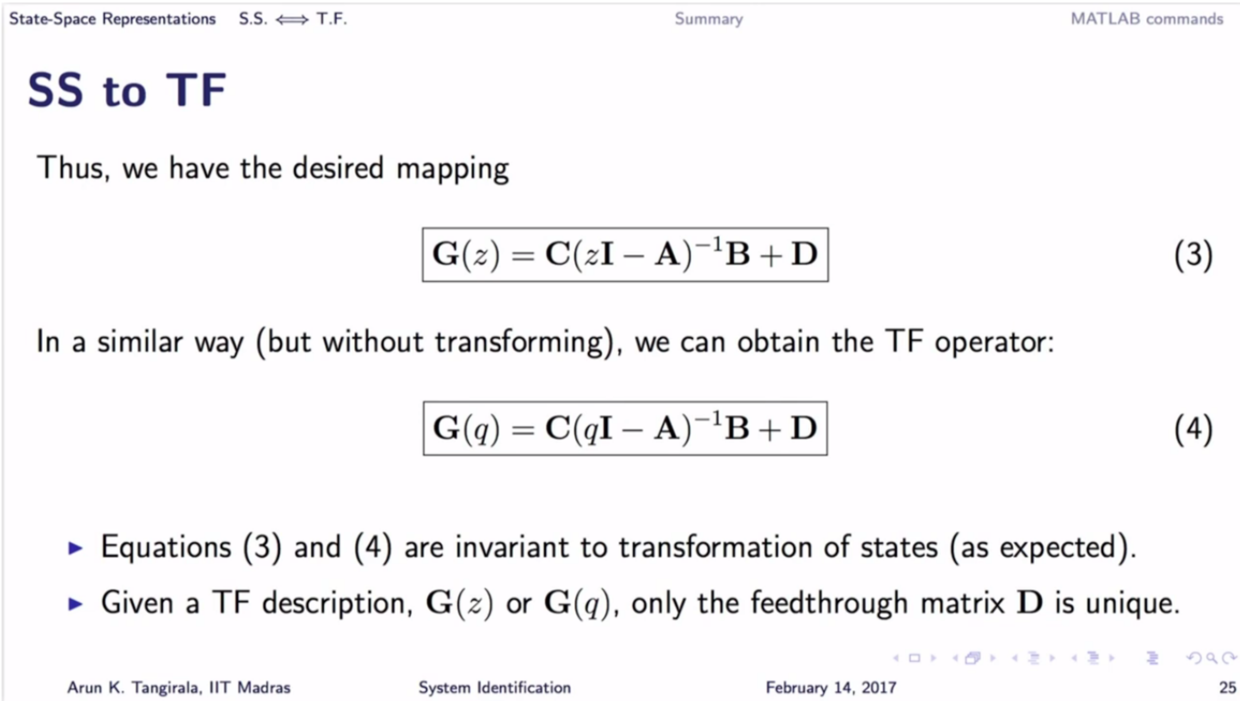
SS to TF, it's going to be one relation. And quickly what you should check here is a change of states is not going to affect the transfer function.

(Refer Slide Time: 23:21)



So if you recall, we said, "When I changed states ABC change accordingly." Right? A becomes T inverse AT, B becomes T inverse B, C maps to C T inverse, D remains unaltered. Plugging that into the expression here. And you will see that nothing changes. I mean for all state-space models should map to the same input-output relation.

(Refer Slide Time: 23:55)



Okay. So of course, the feedthrough term is unique. If I give you a transfer function for example G you can view the state-space form as a factorization of the transfer function. In some sense that is what you're doing. Right? What are you doing when you go from transfer function to state-space form. You are expressing actually G as a multiplication of C times z I minus A inverse times B.

That straight away tells you that this is not unique. I can always pre and post-multiply these matrixes to get the same result, whereas D remains unaltered. You can't change D. So for a given transfer function. The free throw time is unique. Again I keep reiterating this because you have to make really register in your minds that state-space models are not unique but all these non-unique state-space models, different state-space models that do have maps to the same input-output relation.