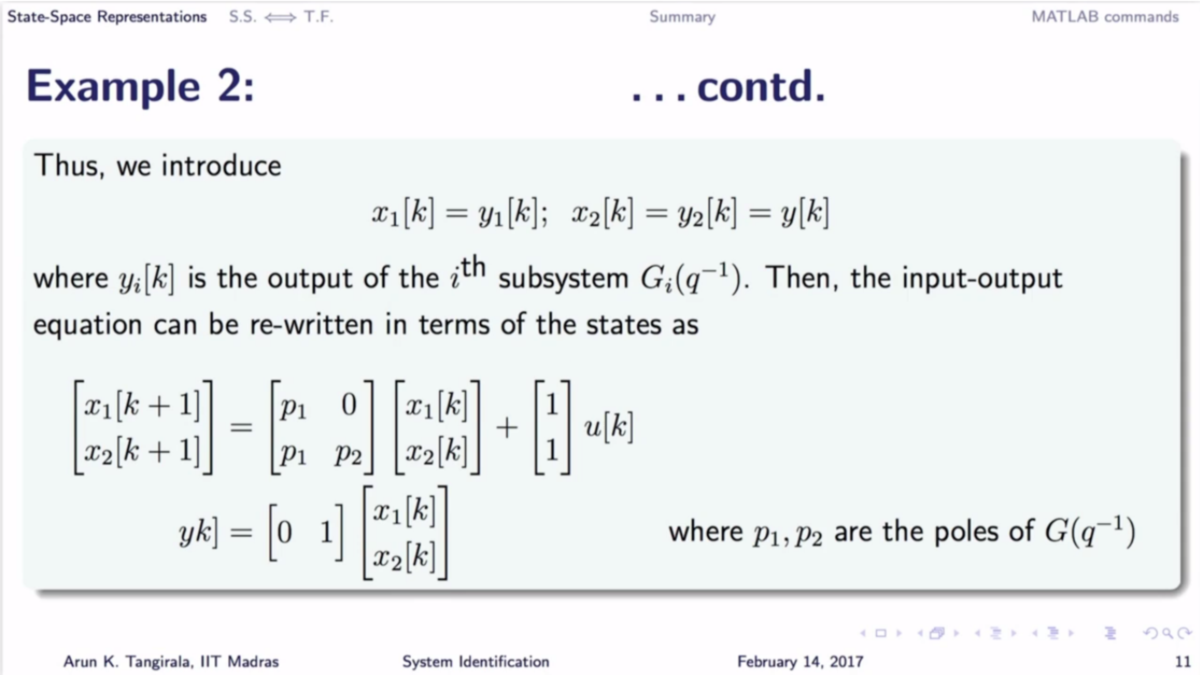
Lecture 19 Part 2

CH5230: System Identification

State-space Representation 2

So here we know already that's a second order system. And in order to complete the picture you have to introduce a second state and it's up to you, how do you assign?

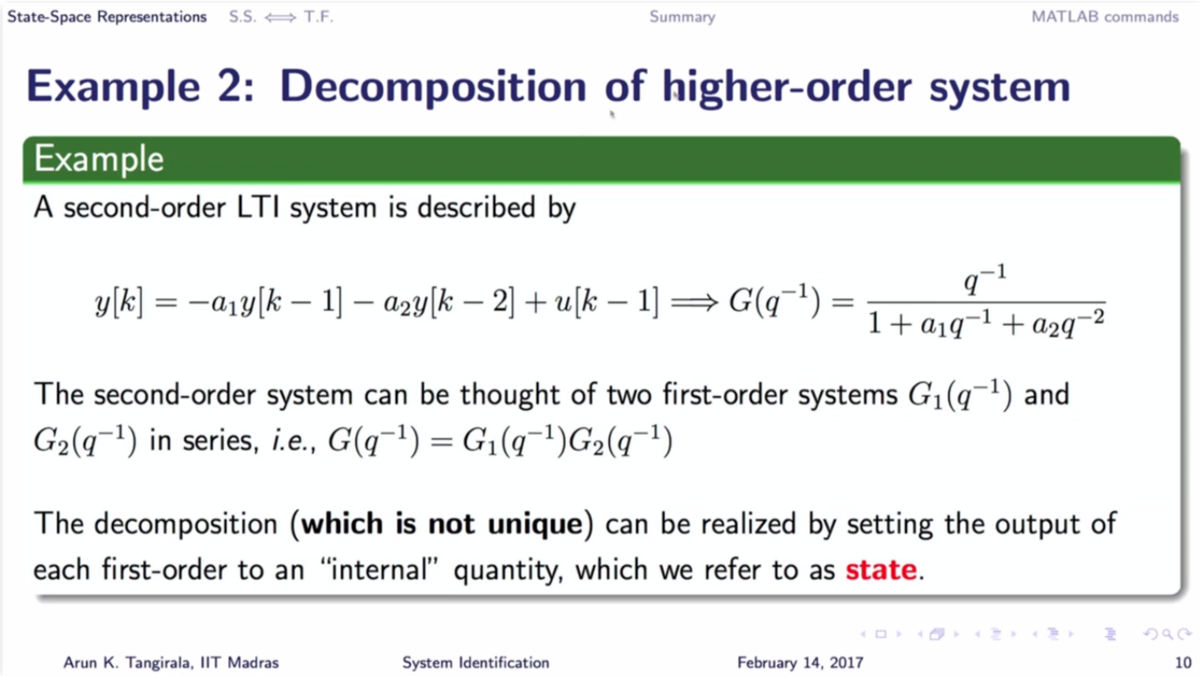
(Refer Slide Time: 00:24)



I have chosen the second state to be the output of the second system, which is G2. If you call this as y2 the output which is in fact the output of the G itself, so in this case the second state is this. And that's what you'll see on the screen. With this choice of states always remember ultimately when you want to write a state-space description you need to have a state equation in the standard form. What is meant by standard form? First order equation for each state. You have to be able to write the evolution equation for each state. And then you need a measurement equation, which is as by know you must recognize is an algebraic equation.

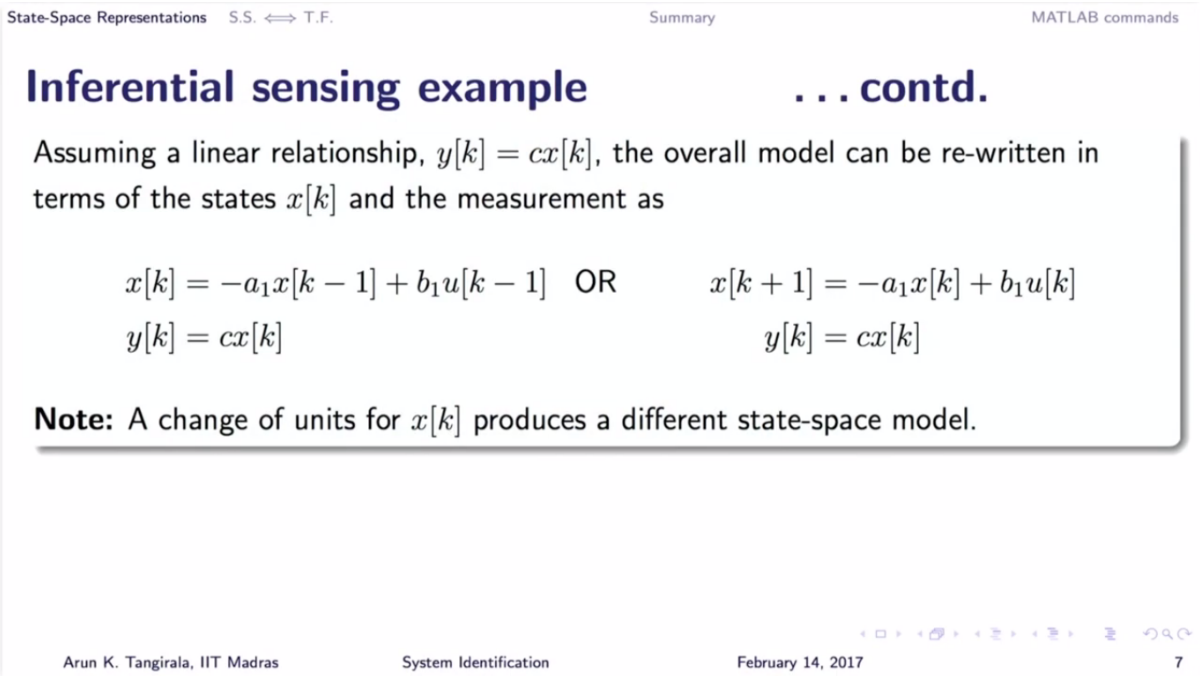
So what we are doing in a state-space model is, we are lumping or you can say consolidating all the dynamics into the states and if any other algebraic relation is present, we are capturing that in the measurement equation. So the initial part of the system goes and sits in the state. And that's what we are doing here. With this choice of states, how do I write the state equation, for example-- that means I have to get state equation for x1, state equation for x2. How do I get the state equation for x1? What are G1 and G2? I need to know that, right? So since I'm imagining G2 be made up of G1 and G2 in series and G is a second order.

(Refer Slide Time: 02:04)



I will write this G in a poll k format, k doesn't mean game. But what I'm going to do is, I'm going to factorize the denominator here and I'm going to write this as 1 minus p1q inverse times 1 minus p2q inverse. So the denominator here is going to be factorized. So that I can say-- so write here G of q inverse is q inverse over 1 minus p1q inverse and 1 minus p2q inverse. Now I have to choose my G1 and G2. Now there is another thing that you have to notice in the state equation, whether there is this example or the previous example.

(Refer Slide Time: 02:57)



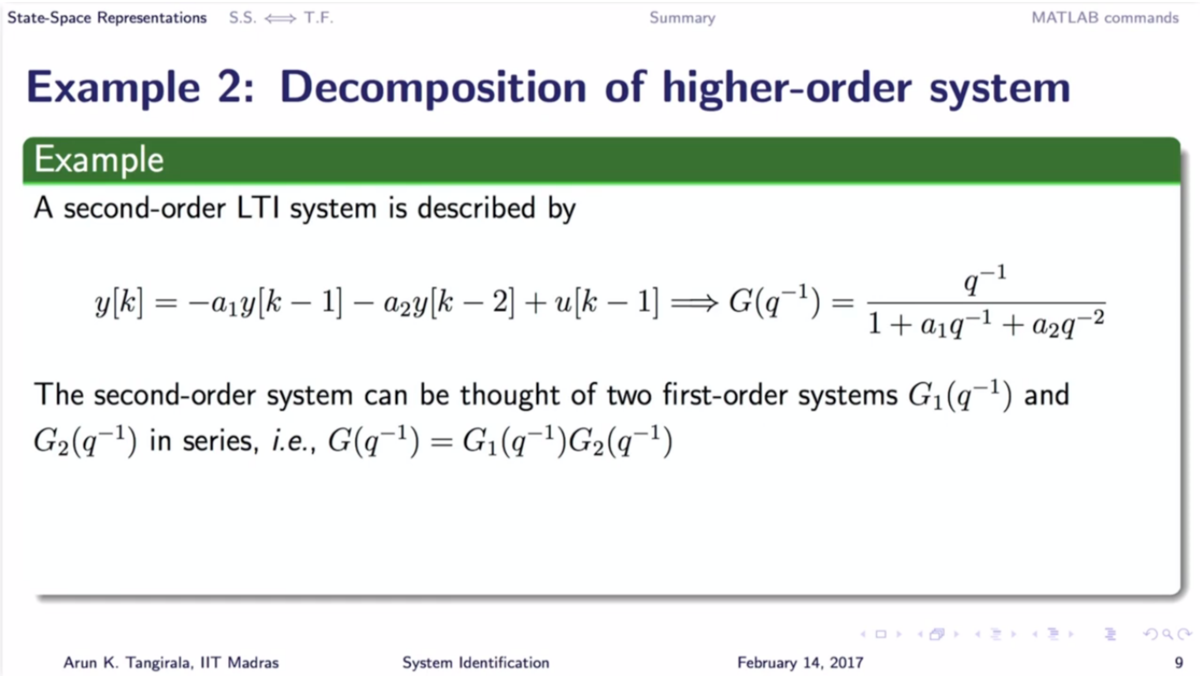
There is a built in 1 delay between the input and state. You may ask, why? Why and what happens, if the system doesn't have any delay at all? Right?

[03:17 inaudible]

Yeah, so what happens is there is an instantaneous effect. Then the input goes and sits into the measurement equation. Okay? But if there is a mix, there are many situations in which there is an instantaneous effect of the input as well as a delayed effect, okay. In which case you will see the input appearing both in the state equation and the output equation or the measurement equation. When you have the input appearing in a measurement equation, we say there is a feed through term, there is a direct feed through term. That means that a part of the input is bypassing the inertial system. I have talked about this earlier as well. Suppose a part of the flow, so if you here, you look at it a part-- suppose I take a part of the flow and mix it here and I write the state-space model for temperature.

Then a change in input will go through this inertial system, where there are dynamics. And then the-- a part of the change will directly affect the temperature, instantly. In which gives the measurement equation or the output equation will also have the input. Okay, so you have to understand now. First, point number one, each state equation has one unit delay and it's a first order difference equation. And the measurement equation typically does not have the input appearing un-- in it, unless there is a feed through term. We deal with sample data systems, where we have a zero order hold and as we will learn tomorrow the zero order hold introduces a unity late. And therefore we don't have to worry unless there is a direct feed through term. Alright, so that most of times a state-space models that we encounter may not have that feed through term, but you have to watch out, depending on the situation. So let us go back to this example here.

(Refer Slide Time: 05:15)



Now I have to factorize. Here, how do I write the state equation? If I look at the decomposition, I had G1 and G2. Suppose I choose this to be the G1, here. I call this as G1. Then what do I get as a state equation? G1 is mapping input u to y1, right? So what do I get there? Let us look at this. So here with-- this is what we want to call as G1. Which means that x1k is q inverse over 1 minus p1q inverse, p1 is one of the poll, it's a first poll. Operating on the input u. Remember the input of G1 is u, right? Now can you write the first state equation? So what would be the state equation? You can just because if you're a beginner then the state equation for this would be x1k equals here p1x1k minus 1 plus uk minus 1. Am I right? Correct? So which means the state equation for x1 is p1x1k, which is what you see, plus uk. Because you'll write in term-- in the form of a forward difference, so the first [aid 7:14] equation is done.

What about the second one? What do you do for the second one? Same, it's the same idea. Instead of input you would have x1 as input, right? So for the, for the second thing, so let us say 1 over 1 minus p1q inverse is your G2. In this case, I have x2k as 1over 1 minus p2q inverse operating on x1k, good. Now what do I get here? There probably a small mistake in the example, there. So here I have x2k, what do you get here? P2, no, there is no mistake, its fine. X2k minus 1 plus x1k, correct? Well, the answer is correct. And whatever you see in the screen is correct. Then how do you proceed further? You are [derivate 8:27] in the forward from. Correct, so let's first write, some baby step here. There is a baby step in between x2k plus 1 equals p2x2k excellent, plus x1 at k plus 1, is it correct? But is it in the desired form? Because we want express a right-hand side purely in terms of states at kth instant and input at kth instant. So what is a record for us?

Substitution, substitute for x1k plus 1 and that's how you get the second equation. Right? So in this form of a decomposition, where with this choice of states the input is actually affecting both the states and you have A that is the matrix, this now, the instead of a scalar in the earlier example you have a matrix A, which is of lower triangular form. So there is a specific structure, B has one [9:44 inaudible] and then y by choice of states has this C there is no D. Why there is no-- I mean there is no D meaning, there is no direct effect of input, because the system itself has one delay. If we go back to the transfer function clearly says there is a unit delay between the input and output. As a simple homework exercise you have to ask yourself, suppose there is no q inverse here. Then what would you do? How would the state-space model change? Just go and work it out. Suppose instead of q inverse I had 1 in the numerator, which means there is no delay. Then how would the state equations change? How would the output equation change?

Think about it. So is it clear now? Now what should-- again to summarize state questions are first order difference equations. Each state equation has a unit delay with respect to the input. The output equation is an algebraic equation. Typically the input doesn't appear in the output equation unless there is a direct effect of the input on the output. And thirdly that the choice of states is not unique, in this example it's clear. Even in the previous example we argued the choice of states is not unique. How can A-- now guarantee uniqueness, suppose I'm looking at a state-space identification problem now, what is a state-space identification problem? I'll be given again the same story, y and u. Right? And by the way there's a square bracket missing in y, I'll fix it. So there-- I'm given y and u and I'm supposed to estimate this matrix here 2 by 2 matrix, this 2 by 1 matrix and this 2 by 1 matrix, so how many unknowns? In terms of model of 8, correct? Compare this with the input-output approach. Suppose I am directly estimating the transfer function not the state-space model.

How many unknowns would be involved? I know, you'll tell me

two. There will be a numerator also, right? Yet you know the numerator constant is 1. So you may have to estimate the numerator constant also D1. So there are three unknowns. And here I have in terms of just modelling itself, I am estimating more unknowns. So you have to ask yourself, why am I turning to state-space identification? When the number of unknowns in terms of parameters itself is growing. In addition to that I have to estimate states. So is it really worth, taking the state-space route when in an input-output model modelling approach? I only have to estimate three. I mean, if I fix the order. Assuming order is fixed and delays don't. On the other hand, if I pre fix the structure, we'll answer that question, why we want to look at stage space models and so on. Suppose I say that, no I want a state-space model with this structure only. Structure, I'm not saying that I know exactly that there is p1 here, p1p2, I don't know that, that is an additional layer of information. Right.

Suppose I just tell you A has this structure, this matrix A now, has this structure. And let us say I know that B has equal elements. And C has this kind of a structure. This is one kind of information that I feel. Now, how many unknowns do I have. Six. So I have [13:9 inaudible]some information. I have six unknowns, still larger than what they have in the state-space model. This is one layer of fixing the state-space model uniqueness. There is no guarantee always that if I fix the structure for any state-space model, I've guaranteed uniqueness because within this structure there may be many solutions. But I can go one more step further. And I say I can say that or I know that these three, among these three actually there are only two unknowns, that is additional layer of information. So slowly I'm moving from a black-box model to a structured state-space model and then I'm introducing some greyness, all right. And maybe further I can say, oh, you know these are equal, so I can say this is b, this is b. Now, how many unknowns do I have?

Four.

Some gradually sinking it. Further I say, I know that this is one. I have three. I have brought it to that input output thing. So you see the input output form that you see actually is equivalent to identifying the structured state-space model, we call this as structured state-space identification and where do you know the structure from, is it's all on Amazon? Can you peep into your friends homework and figure out. Typically you will have to derive it from some prior knowledge. Is the prior knowledge is driven from the physics of the process, we say that we are identifying grey-box models. Otherwise a generic term for identifying state-space models with a prior structure is called a structured state-space identification. So we learn that now, input output modelling as we have been learning, that is parametric input output modelling, is nothing but structure state-space identification.

We still have not answered the question, why and when we would actually turn to an unstructured state-space identification instead of input output identification, very well knowing that unstructured state-space identification has more number of unknowns to be estimated. We have not answered that question as to why we would want to look at that but we know that there are these different classes of state-space identification problems. So hopefully through these two examples I have given you a feel of where you would encounter state-space identification. One scenario is where the states are hidden physical variables that they are not sensing and you also want to identify the model, both estimate the variables and identify the model, that's where you would run into state-space identification.

The other one is where you are just estimating a state-space model for a given system, where simply the states are the outputs of first order systems. We are deliberately chosen that way, so that I can now say that the straightaway assigned states because I know states are all satisfying first order difference equations. So one hint you have to always remember is, the states are always outputs of some first order subsystem in the process. And there is no unique way of breaking up a higher order system into first order subsystems. I could have written a parallel decomposition. What would have changed very quickly if I did a parallel decomposition?

Yeah. Which of the matrices would change? Do all of them change? Will A,B,C change? If I were to imagine G to be made up of G1 plus G2.

A.

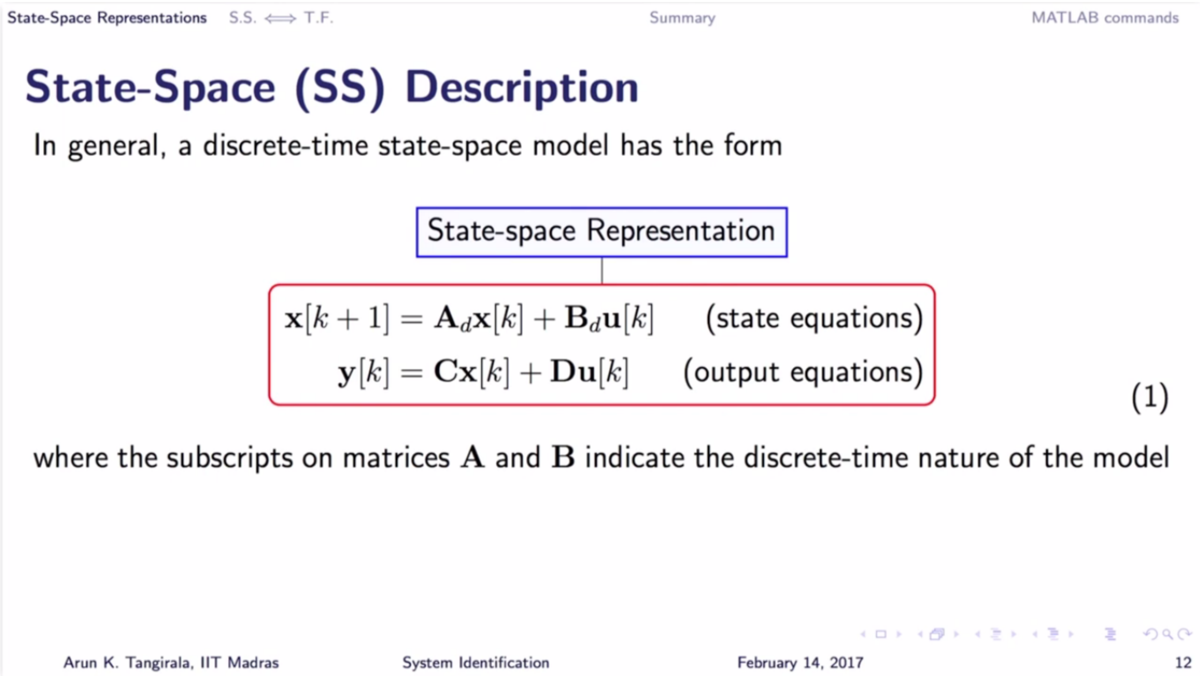
A. A would change. In what way?

It's very easy. Imagine that you're doing a partial fraction expansion, right. That's what is parallel decomposition. The first term would be a first order subsystem, second term would be a first order, another first order subsystem. Okay, C change. C would change to 1, 1. I mean it's your choice. Structurally how would A change?

Diagonal.

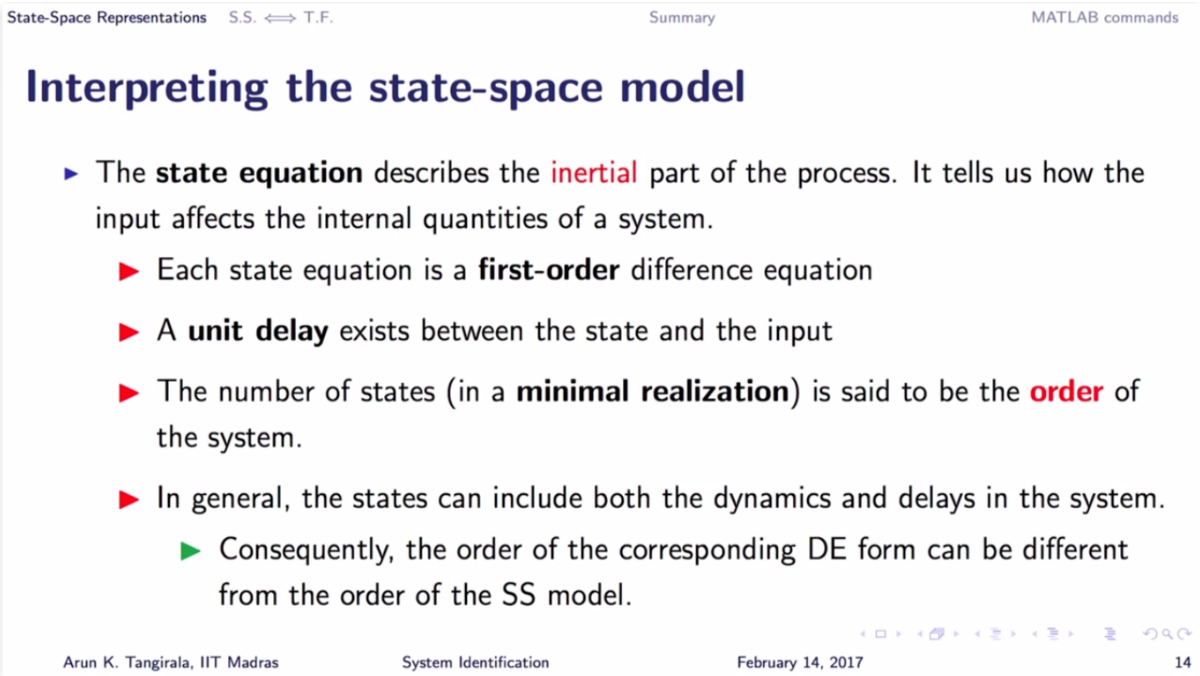
Correct. Diagonal. Right. And the diagonal elements would be the poles. That's called a diagonal canonical form. So now there are different canonical forms to states-space models. Canonical meaning having some structure. So let's quickly look at the state-space model here.

(Refer Slide Time: 18:29)



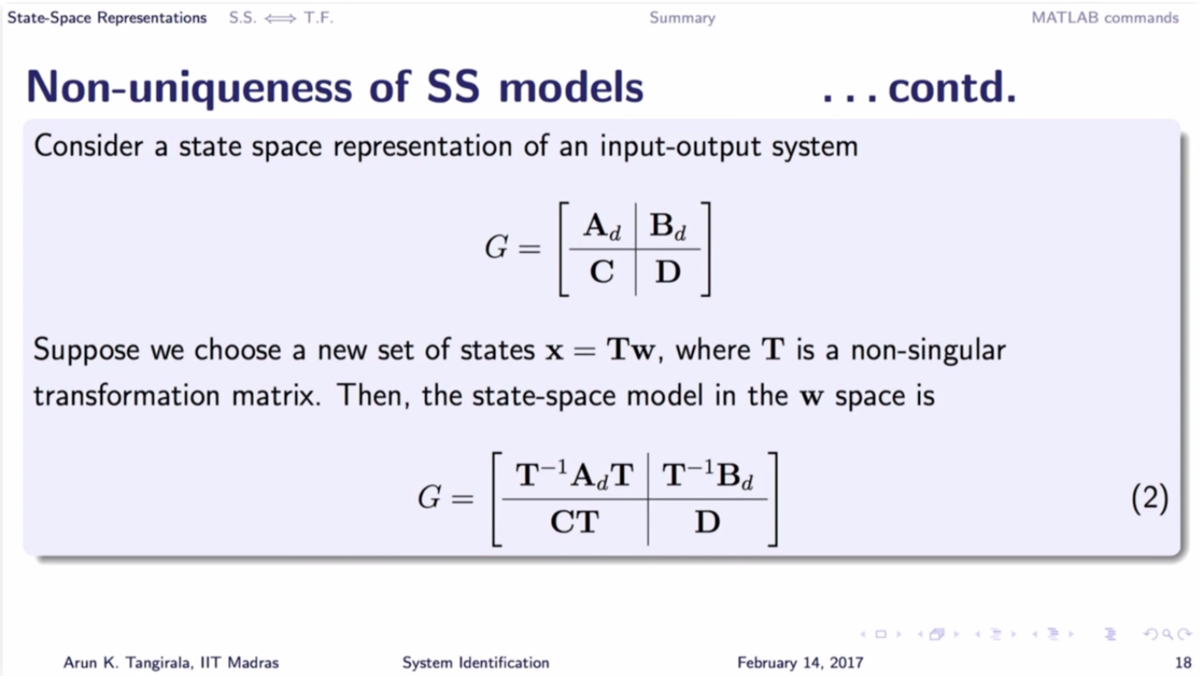
This is a general discrete time state-space model that you will run into the subscript D, there is only denoting that this is discrete time. Very soon we will drop that. This form is the same even for continuous time systems. For continuous time processes also you have A B C D, except that the state equations there would be differential equations. Each state equation would be a first order [18:54 inaudible]. Clear. But otherwise the output still remains an algebraic equation. And as I said in the state-space identification the goal is given input and output data. We have to estimate A B C D and X. You should slowly understand that generally in systems, by the end of system integration you would have learnt the A to Z of system identification literally that means you would have used all the alphabets from A to Z, lowercase and uppercase. Plus almost all the Greek symbols. So any more developments in system unification will call for other kinds of scripts. And my goal is at least in my lifetime I should be able to use Devanagari script symbols. Why not? Right. Okay. Anyway so, I've already talked about this interpreting the state-space models. I'm going to skip.

(Refer Slide Time: 19:51)



Again remember minimal realisation concept. In minimal realisation you would just have the minimum number of states required to describe what the input output relationship. That is the benchmark for you always. Given an input output realization how many states are required. If you choose a minimum set of states then you're working with a minimal realisation. In state, in identification, there is this additional problem of identifiability. That means the true system may have some minimal realisation, fine. Maybe the true system is third order or fourth order, But because of lack of excitation you may not be able to identify the full states minimal realization model, depending on the level of excitation, like we discussed, one of the early example and identifiability. If I don't excite well enough, I may end up identifying a lower order system because there is not enough information in the data. So that has got to do with absorbability, which eventually translates to identifiability. I'm going to skip the output equation part. We've also spoken about non uniqueness.

(Refer Slide Time: 21:00)



This is just to show that if you change the states from X to W. Where X and W are related to a non-singular transformation, then the state-space model in terms of this W is this, T inverse AdT, T inverse Bd, C times T and D. This is important to remember but even if you don't remember. no big deal. You can just drive it on the fly. All you have to do is plug in for X as T times W. That's all. In place of x you write Tw, since T is given to non-singular, T inverse exists. What does this tell me when I do a change of states, ABC change but D won't change. Why is that?

[21:44 inaudible]

Sorry.

[21:49 inaudible]

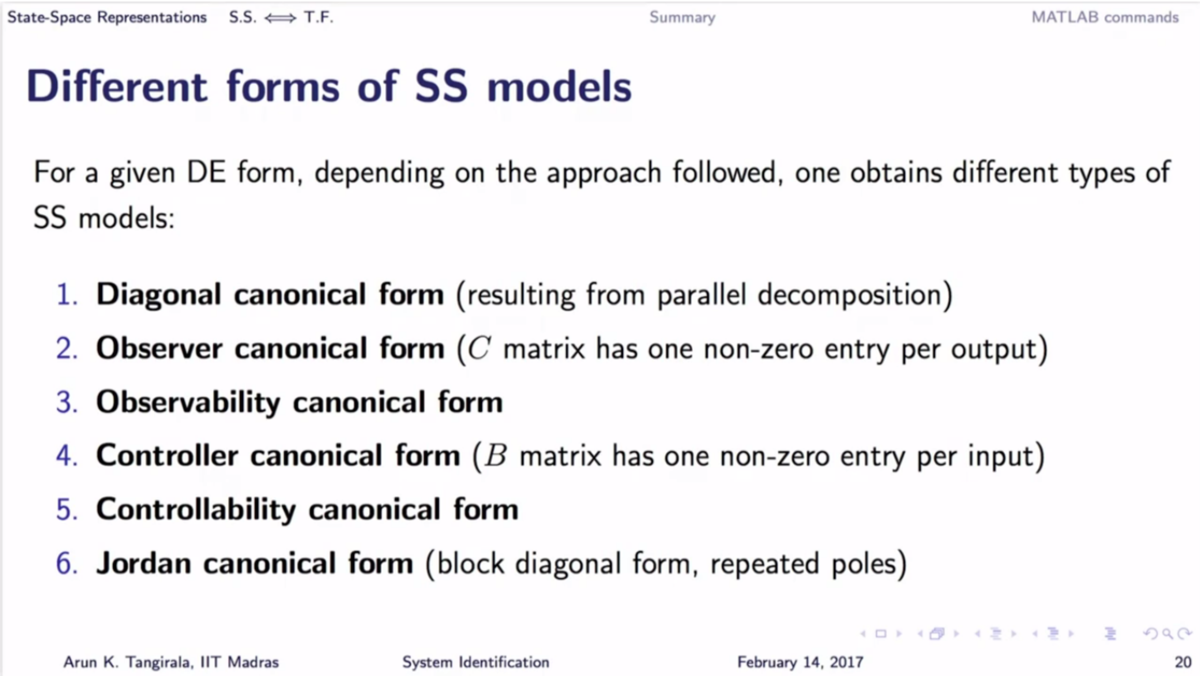
It is.

[21:51 inaudible]

So it bypasses the initial system. Therefore that is unaffected by how you describe this system. Suppose there was a [22:00 inaudible] cycle here and mixing up, you can choose to represent this in terms of 50 states if you like U.S. or may of you if you like India maybe 29 states and so on whatever. That relation for the feed through terms remains un-effective. It says I don't care. And that is why D remains unchanged. Okay. That is something to remember.

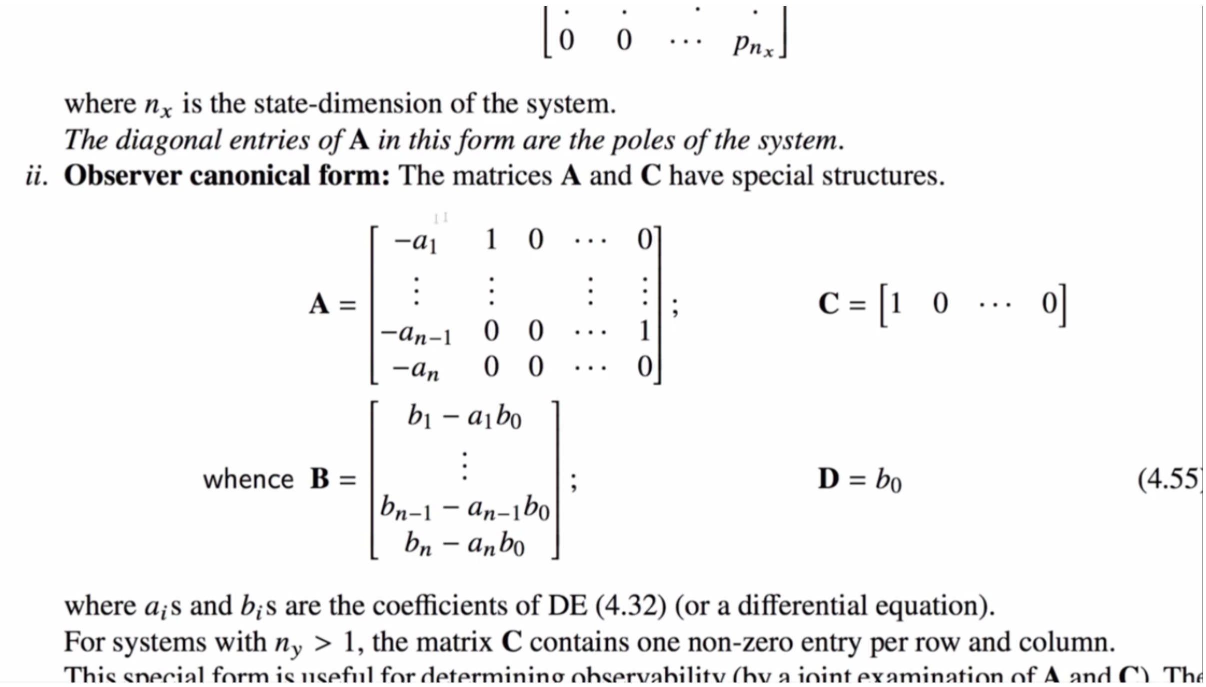
And as I just said there are different forms of state-space models depending on these structures of ABC and D. There is diagnosed canonical form, that we just discussed coming out of parallel decomposition. Then there is observable canonical form. When you have observability canonical form and so on. How these names have come about because the structures that these canonical forms have for example, the observer canonical form has a special structure. In fact if you look at my textbook you will see that, I'll just show you this and then we'll end the discussion.

(Refer Slide Time: 23:01)



So here, I have the observer canonical form. Look at the structure of ABC, D forget it. You can keep aside. A has all the, what are this a1's to an's, they are the coefficients of the characteristic equation. Okay.

(Refer Slide Time: 23:20)



$$\begin{bmatrix} 0 & 0 & \cdots & p_{n_x} \end{bmatrix}$$

where n_x is the state-dimension of the system.
The diagonal entries of **A** *in this form are the poles of the system.*

ii. **Observer canonical form:** The matrices **A** and **C** have special structures.

$$\mathbf{A} = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ -a_{n-1} & 0 & 0 & \cdots & 1 \\ -a_n & 0 & 0 & \cdots & 0 \end{bmatrix}; \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$\text{whence } \mathbf{B} = \begin{bmatrix} b_1 - a_1 b_0 \\ \vdots \\ b_{n-1} - a_{n-1} b_0 \\ b_n - a_n b_0 \end{bmatrix}; \qquad \mathbf{D} = b_0 \qquad (4.55$$

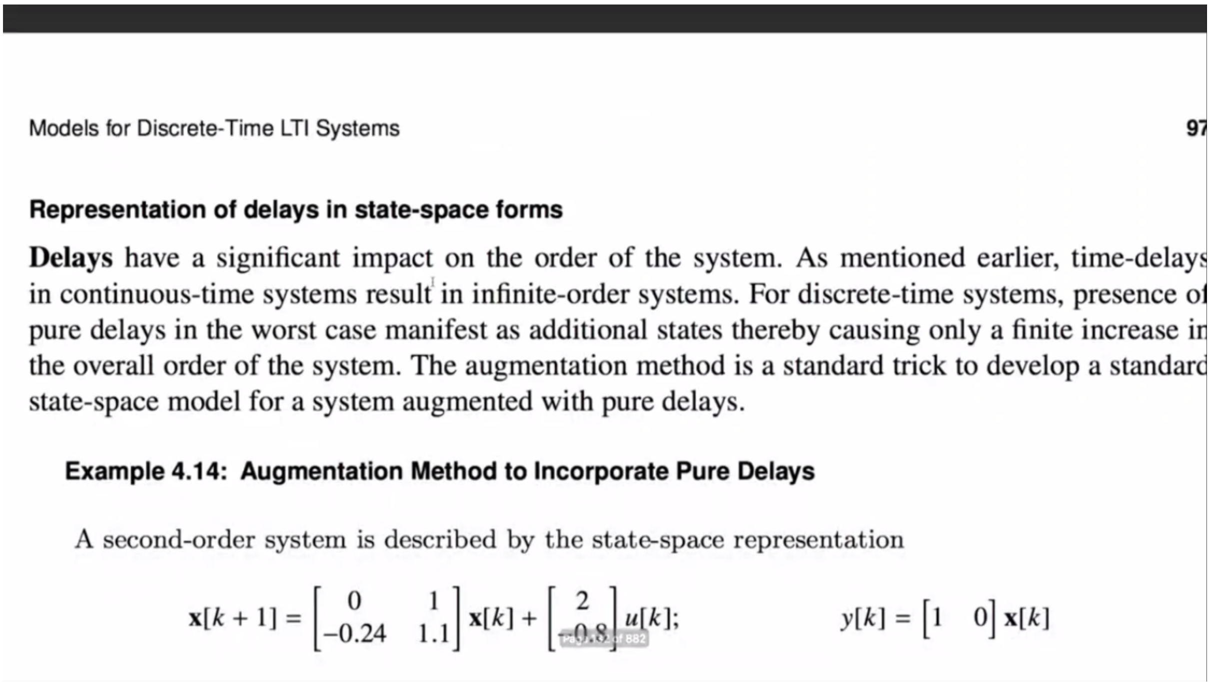
where a_is and b_is are the coefficients of DE (4.32) (or a differential equation).
For systems with $n_y > 1$, the matrix **C** contains one non-zero entry per row and column.
This special form is useful for determining observability (by a joint examination of **A** and **C**). The

And they all fall in the first column of A, well that is oaky. You can always [23:25 inaudible] the columns of A. That's not a big issue. And the remaining is simply made up of 0 and 1, in this particular way. And then C, is simply 1 and then the remaining 0. And D is made up of this way. So in what is so special about this by looking at the structure. In fact I can say, whether an observer can be built. In fact you can write an observer canonical form for a given state-space model only if it is fully observable. That means this a's that you have in the first column they're all non-zero values. And C here will, is telling you straight away that the first estimate this, the first state is output itself. So there itself I have the estimate of the first state.

How do I get the estimate of the second state, by exploiting this relations here in your state equation. So by looking at this A and C together, I can straightaway comment on whether all states can be observed upon.

Likewise you have observability canonical form and then you have controller canonical form.

(Refer Slide Time: 24:36)



Models for Discrete-Time LTI Systems 97

Representation of delays in state-space forms

Delays have a significant impact on the order of the system. As mentioned earlier, time-delays in continuous-time systems result in infinite-order systems. For discrete-time systems, presence of pure delays in the worst case manifest as additional states thereby causing only a finite increase in the overall order of the system. The augmentation method is a standard trick to develop a standard state-space model for a system augmented with pure delays.

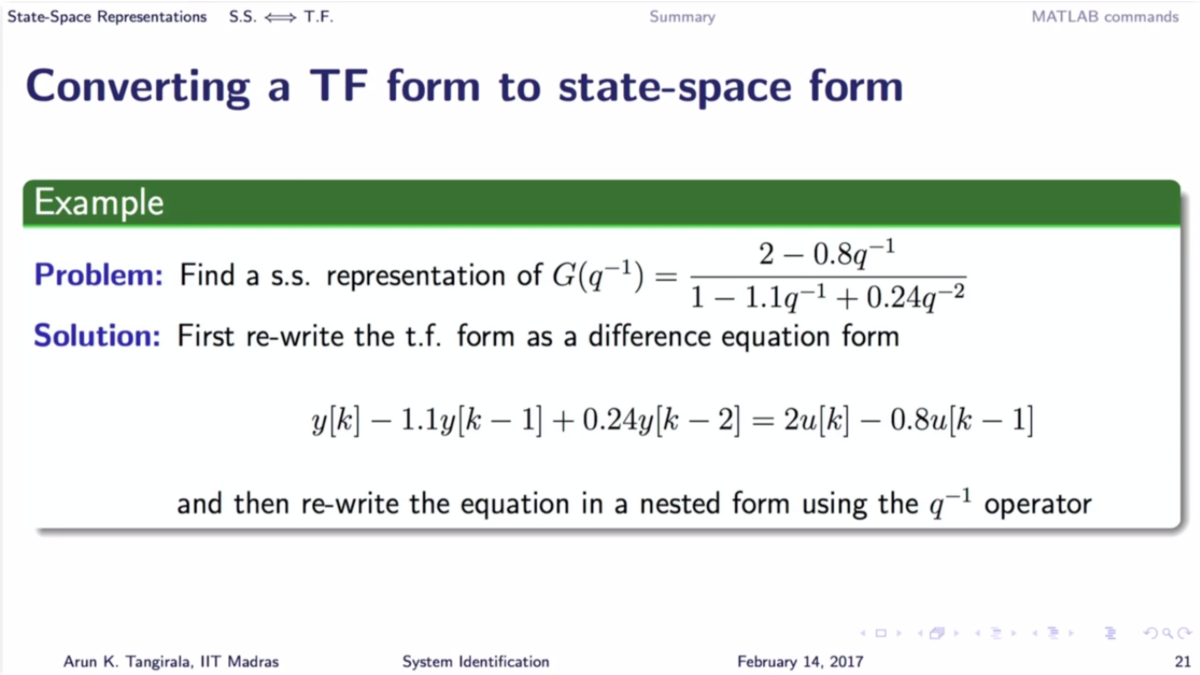
Example 4.14: Augmentation Method to Incorporate Pure Delays

A second-order system is described by the state-space representation

$$\mathbf{x}[k+1] = \begin{bmatrix} 0 & 1 \\ -0.24 & 1.1 \end{bmatrix} \mathbf{x}[k] + \begin{bmatrix} 2 \\ -0.8 \end{bmatrix} u[k]; \qquad y[k] = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}[k]$$

Again there the controller canonical form has got to do with whether I mean, controllability canonical forms and controller canonical forms has got to do with whether your system is controllable or not. Observability is about being able to identify state from measurements, controllability is about being able to drive the states from point A to point B using your inputs. Obviously that will depend on the relation between states and inputs. That means when it comes to controllability A and B matrices matter. And when it comes to observability A and C matter. And then there exist some beautiful relations which we'll talk about it later on. But what we'll do tomorrow is wind up this discussion on state-space models, where we have the relations between how to go from transfer function from the state-space form. We already discussed one such example.

(Refer Slide Time: 25:36)



But more importantly how to come back to the transfer function form and then quickly talk about how delays can increase the order of state-space modelling. Then we will briefly talk about discretization, okay.