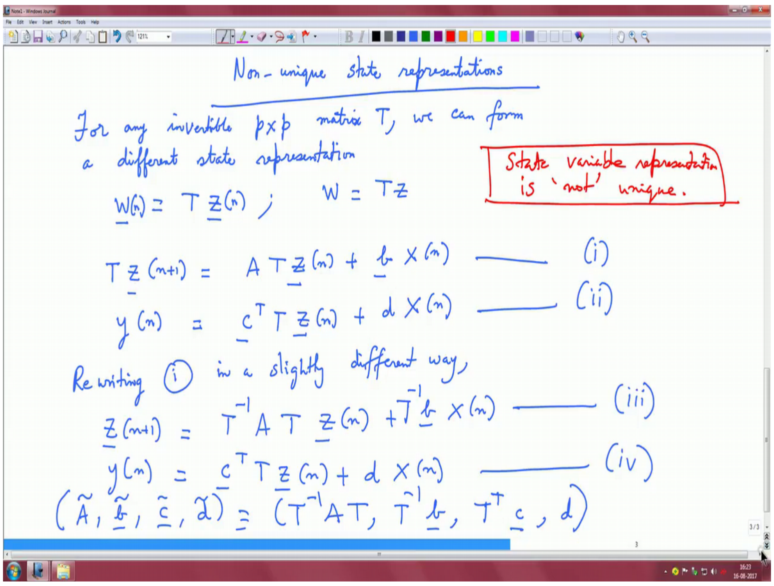
Mathematical Methods and Techniques in Signal Processing - I
Prof. Shayan Srinivasa Garani
Department of Electronic Systems Engineering
Indian Institute of Science, Bangalore

Lecture – 07
Non-uniqueness of state space representation

In the previous module we derived the state space representation, right from first principles. So, we started off with a linear time invariant model given by a transfer function for the p equals q case, we analyze the signal flow graph, we defined the state variables and we formulated a state variable model in terms of your a b c d parameters, relating the input and output right. But one of the questions we have to ask is, is the state space representation unique. Our intuition will tell us probably it is not because you can define your state variables for different points in a different way in your delay line and accordingly you can have a different representation. Intuition will tell us, but we will prove it rigorously if it is unique or not right.

(Refer Slide Time: 01:14)



So, for any invertible p by p matrix T we can form a different state representation. So, what you do is let us suppose W is some T times Z, I am omitting the vector notation here I mean with a slide abuse of notation you know we can say this is Z and I can put a n here. This is very precise, I would say with slight abuse of notations I will say W is basically T time Z, but W is a vector Z is a vector and T is a matrix which is a p by p matrix and it is invertible. So, let us define it in this form.

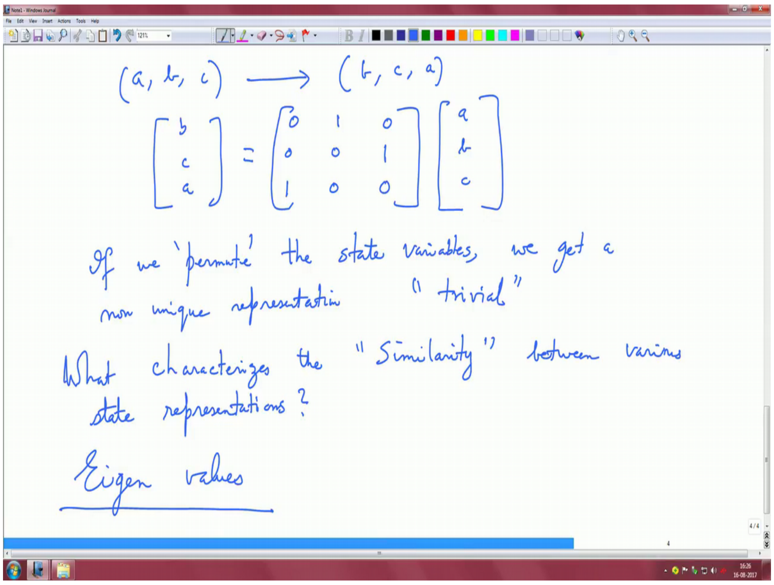
Now, going through the previous equations we have W n plus 1 is basically T times Z n plus 1 which is equal to A times W n, W n is T times Z n plus some vector b times X n right. We just or replacing W n with T times Z of n. And we are writing y n equals some C transpose times W n is basically T times Z n plus some d times X n. So, whatever is before X n that remains the same that does not change only the variable W will change because it is linked with Z through this transformation T.

Now, we will rearrange let us call this equation 1, let us call this equation 2 we can rewrite equation 1 in a slightly different way. So, when we do that what really happens is as follows right. You see just Z n plus 1 is basically we pre multiply equation 1 by T inverse and since it is invertible matrix T inverse should exist right. So, therefore, it is T inverse times A times T Z of n plus b x of n and y n remains the same. That is a T inverse here which is basically T inverse times b X n and I will call this say equation three and y n is basically C transpose T Z n plus d times X n.

Now, if I ask you for the state variable representation of this system where W and Z which are 2 different state variable representations are linked via the transformation T the parameters are as follows. So, your A tilde, b tilde, c tilde and d tilde will now be T inverse at is going to be, T inverse b right and then we said C transpose is what we had C transpose is basically now T transpose times c and then d. So, our original parameters that we had as A, b and then c transpose and d are now in this form link to via transformation T right.

Now, if you ask the question how many such possibilities of invertible matrices can you find, infinite right. So, therefore, what can we conclude? State variable representation is it unique, it is not unique, so that is the take home message. State variable representation is not unique and a trivial invertible matrix is the permutation matrix. So, if you just think about a permutation matrix it is an invertible matrix.

(Refer Slide Time: 08:19)

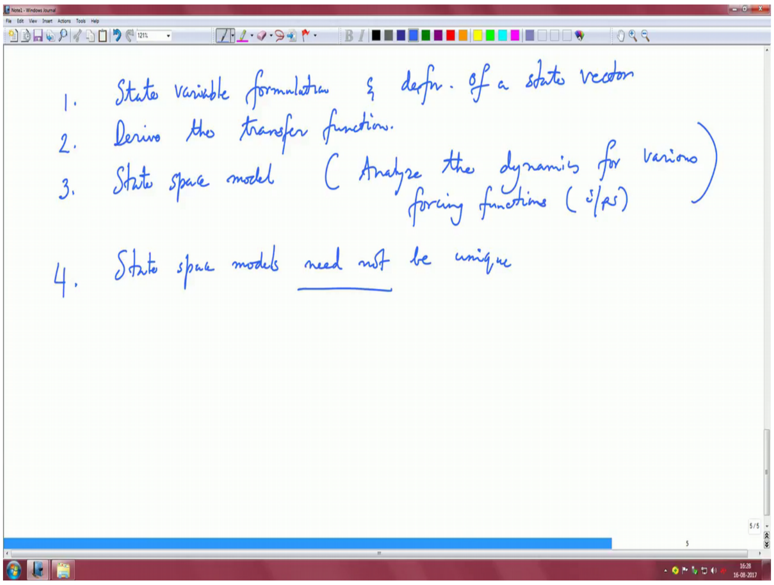


And a good example is suppose I have an object a b c and I permute this object by b c and a right. What is the 3 by 3 transformation? That can take you from this subject to the other object right. So, here is what I would do right if I want b c a on this side, but I want to link this via a matrix to a b c. So, basically grab b out first then I want c then I have a this is a linear transformation that can take a b c to b c a which is a permutation right and this is a invertible matrix. So, which clearly tells you that if we permute the state variables we get a non unique representation and this is really a trivial transformation, one of the trivial transformations. So, that is an important parameter.

Now, if I ask you another question here which is what characterizes the similarity between various state representations. Well we studied just recently that state variable representation is there the state space model is not unique right, but what characterizes the similarity between various state representations and that is precisely the eigen values and if you invoke the similarity transformation that is what it does, that is what we did.

Your modes should be the same no matter what. So, this is a very important step and I think the take home from this state variable representation or a few salient points which I want to bring to your attention.

(Refer Slide Time: 11:46)



The first point is state variable formulation and definition of a state vector. This need not be unique because we saw in various ways how we can make this non unique using similarity transformations right, so definition of the state vector. So, once you formulate the state variables and we defined the state vector then we can derive the transfer function and once we derive the transfer function you know using state space model we can analyze the dynamics for various forcing functions or various inputs. This is very important otherwise if you were to go the algebraic route in trying to formulate all your equations and derive things you will be struggling around. So, therefore, it is very important to formulate the state space model and then once it is ready you can plug it into your favourite simulation setup and you can exercise to figure out the dynamics of the model.

And the last take home message is the state space models are not unique and the similarity of various state space representations is through eigen values and therefore, it is coupled via the similarity transformation. So, this completes the module on state space representation.