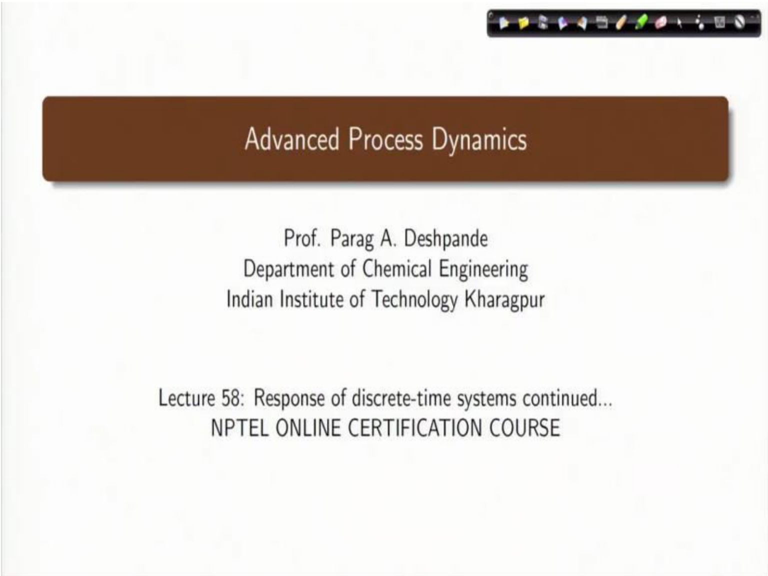
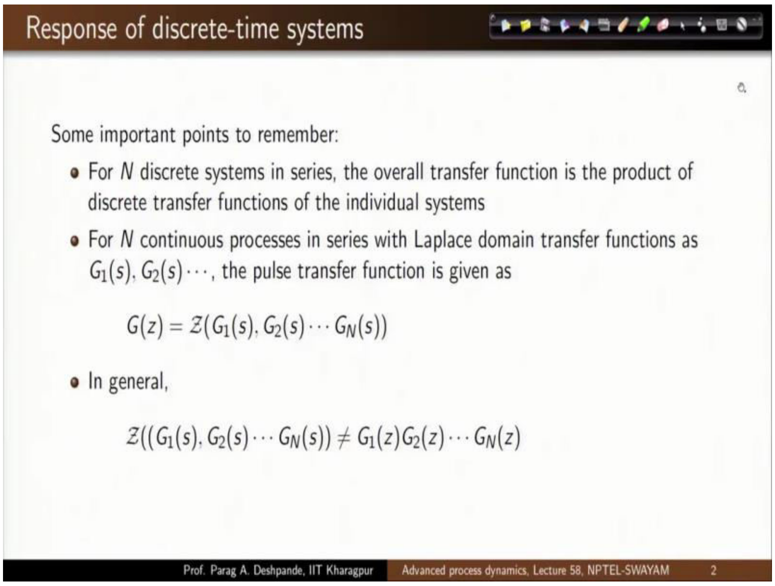
Advanced Process Dynamics

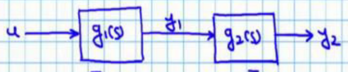
Professor Parag A Deshpande
Department of Chemical Engineering
Indian Institute of Technology Kharagpur

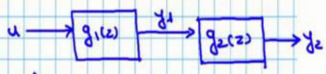
Lecture 58

Response of discrete-time system continued…









$$g_1(s) = \frac{\bar{y}_1(s)}{\bar{u}(s)} \; ; \; g_2(s) = \frac{\bar{y}_2(s)}{\bar{y}_1(s)}$$

$$g_1(s) g_2(s) = \frac{\bar{y}_1(s)}{\bar{u}(s)} \times \frac{\bar{y}_2(s)}{\bar{y}_1(s)} = \frac{\bar{y}_2(s)}{\bar{u}(s)}$$

$$g_p(s) = g_1(s) g_2(s)$$

$$g_{overall}(s) = g_1(s) g_2(s) \cdots g_N(s)$$

$$\hat{y}(z) = g(z)\,\hat{u}(z)$$

$$g_{overall}(z) = g_1(z) g_2(z) \cdots g_N(z)$$

Some important points to remember:

- For N discrete systems in series, the overall transfer function is the product of discrete transfer functions of the individual systems
- For N continuous processes in series with Laplace domain transfer functions as $G_1(s), G_2(s) \cdots$, the pulse transfer function is given as

$$G(z) = \mathcal{Z}(G_1(s), G_2(s) \cdots G_N(s))$$

- In general,

$$\mathcal{Z}((G_1(s), G_2(s) \cdots G_N(s)) \neq G_1(z) G_2(z) \cdots G_N(z)$$

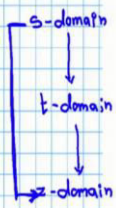
$g_1(s), g_2(s) \cdots g_N(s) \leftarrow$ Laplace domain transfer functions

$$g(s) = g_1(s) g_2(s) \cdots g_N(s) \quad -(1)$$

$$g(s) \longrightarrow g(t) \longrightarrow g(nT) \longrightarrow g(z)$$

$$g(z) = \mathcal{Z}(g(s))$$

$$\Rightarrow g(z) = \mathcal{Z}\{g_1(s) g_2(s) \cdots g_N(s)\}$$



Response of discrete-time systems

Some important points to remember:

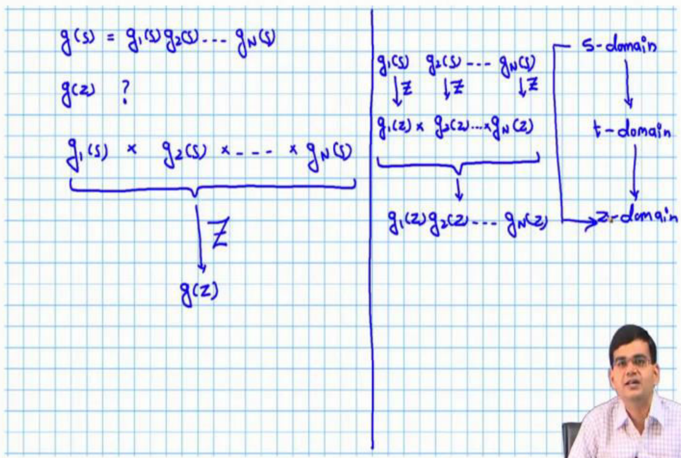
- For N discrete systems in series, the overall transfer function is the product of discrete transfer functions of the individual systems
- For N continuous processes in series with Laplace domain transfer functions as $G_1(s), G_2(s)\cdots$, the pulse transfer function is given as

$$G(z) = \mathcal{Z}(G_1(s), G_2(s)\cdots G_N(s))$$

- In general,

$$\mathcal{Z}((G_1(s), G_2(s)\cdots G_N(s)) \neq G_1(z)G_2(z)\cdots G_N(z)$$

Response of discrete-time systems



Response of discrete-time systems

Some important points to remember:

- It is possible to model multiple input-multiple output systems using difference equations
- For a multiple input-multiple output discrete system, a discrete transfer function matrix can be defined

Response of discrete-time systems

SISO

$$\frac{dx}{dt} = ax + bu \quad -(1)$$

$$y = cx + du \quad -(2)$$

$$\frac{x_{n+1} - x_n}{T} = a x_n + b u_n$$

$$\Rightarrow \frac{x_{n+1}}{T} = \left(\frac{1}{T} + a\right) x_n + b u_n$$

$$\Rightarrow x_{n+1} = T\left(\frac{1+aT}{T}\right) x_n + bT u_n$$

$$\Rightarrow x_{n+1} = (1+aT) x_n + bT u_n$$

$$1 + aT = a_1 \; ; \; bT = b_1$$

$$x_{n+1} = a_1 x_n + b_1 u_n \quad -(3)$$

$$y_n = c x_n + d u_n \quad -(4)$$

$u \rightarrow u(0), u(T), u(2T) \ldots$ } known

$\quad\;\; u_0, \;\; u_1, \;\; u_2 \ldots$

$x(0)$, x_0 } known

$x_1 = a_1 x_0 + b_1 u_0 \leftarrow x(T)$

$y_0 = c x_0 + d u_0 \leftarrow y(0)$

$x_1 \leftarrow$ known now; $u_1 \leftarrow$ known

$y_1 = c x_1 + d u_1$

$\vdots$

Response of discrete-time systems

SISO

$$y = cx + du \quad -(1)$$

$$\Rightarrow y_n = c x_n + d u_n \quad -(2)$$

$$x_{n+1} = a_1 x_n + b_1 u_n \quad -(3)$$

$$\frac{dy}{dt} = c\frac{dx}{dt} + d\left(\frac{du}{dt}\right)$$

$$y_{n+1} - y_n = c(x_{n+1} - x_n) + d(u_{n+1} - u_n)$$

$$\Rightarrow y_{n+1} = y_n + c x_{n+1} - c x_n + d u_{n+1} - d u_n$$

$$\Rightarrow y_{n+1} = y_n + c(a_1 x_n + b_1 u_n) - c x_n + d u_{n+1} - d u_n$$

$$\Rightarrow y_{n+1} = y_n + c(a_1 - 1) x_n + (c b_1 - d) u_n + d u_{n+1} \quad -(4)$$

$$x_n = \frac{1}{c} y_n - \frac{d}{c} u_n \quad -(5)$$

$$y_{n+1} = y_n + c(a-1)\left[\frac{1}{c} y_n - \frac{d}{c} u_n\right] + (c b_1 - d) u_n + d u_{n+1}$$

$$y_{n+1} = \alpha y_n + \beta u_n + \gamma u_{n+1} \quad -(6)$$

$u_0, u_1, u_2, \ldots$ } known

x_0 } known

$y_0 = c x_0 + d u_0 \leftarrow$ known

$y_1 = \alpha y_0 + \beta u_0 + \gamma u_1$

Response of discrete-time systems

MIMO - 2nd order, 2-input, 2-output

$$\frac{dx_1}{dt} = a_{11} x_1 + a_{12} x_2 + b_{11} u_1 + b_{12} u_2$$

$$\frac{dx_2}{dt} = a_{21} x_1 + a_{22} x_2 + b_{21} u_1 + b_{22} u_2$$

$$y_1 = c_{11} x_1 + c_{12} x_2 + d_{11} u_1 + d_{12} u_2$$

$$y_2 = c_{21} x_1 + c_{22} x_2 + d_{21} u_1 + d_{22} u_2$$

$$x^1_{n+1} = a_{11} x^1_n + a_{12} x^2_n + b_{11} u^1_n + b_{12} u^2_n \quad -(1)$$

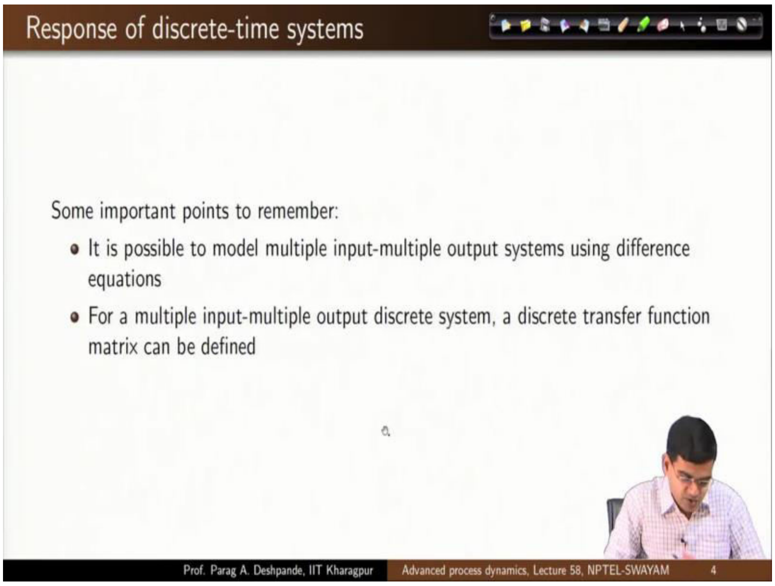
$$x^2_{n+1} = a_{21} x^1_n + a_{22} x^2_n + b_{21} u^1_n + b_{22} u^2_n \quad -(2)$$

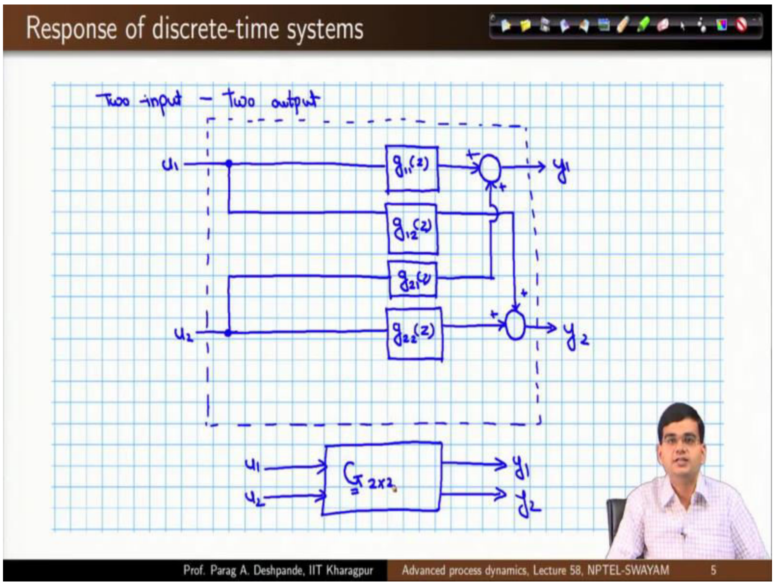
$$y^1_n = c_{11} x^1_n + c_{12} x^2_n + d_{11} u^1_n + d_{12} u^2_n \quad -(3)$$

$$y^2_n = c_{21} x^1_n + c_{22} x^2_n + d_{21} u^1_n + d_{22} u^2_n \quad -(4)$$

$$x_{n+1} = a_1 x_n + b_1 u_n$$

$x^1_n \xrightarrow{T} x^1_{n+1}$, $\quad x^2_n \xrightarrow{T} x^2_{n+1}$





(Refer Slide Time: 0:28)

So let us have some more observations on the dynamical response of discrete time systems. So let me have some comments before we close stop it. So, what we have in front of us is some set of observations which are of special importance when you consider the pulse transfer function. So, the first thing is that a for an N discrete systems in series. Till now we considered a single system. So, in the previous lecture, we took the pulse transfer function of system, which was first order and it was equipped with a 0-order hold. We determine the first transfer function multiplied it with the z transform of the input did the inverse z transform.

(Refer Slide Time: 01:33)

Now if you have N systems in series, what would you do? Well, let us see what you would do for a continuous domain system, you have, imagine system of two sub systems in series. So, you have an input u, it goes to the first system which has the transfer function $g_1(s)$, it gives an output y_1, output y_1 access the feed to the second system which has the transfer function $g_2(s)$, you get the ultimate output y_2. So, what are the different relationships which we can write? I can write $g_1(s) = \bar{y}_1(s)/\bar{u}(s)$ and I have $g_2(s) = \bar{y}_2(s)/\bar{y}_1(s)$.

So let me write this properly, this would be $\bar{y}_2(s)/\bar{y}_1(s)$. From where I can write $g_1(s)$, $g_2(s)$, is equal to $\frac{\bar{y}_1(s)\,\bar{y}_2(s)}{\bar{y}_1(s)\,\bar{u}(s)}$, which is equal to what? $\frac{\bar{y}_2(s)}{\bar{u}(s)}$. So therefore, I can write g overall simply g(s), as $g_1(s)$, $g_2(s)$, and in fact, we did make use of this fact in the previous lecture by writing g as a product of g(p) and g(h), the process transfer function and pole transfer function, is the same concept here.

So therefore, this is a general observation that g overall of s for an N system would be $g_1(s)$ $g_2(s)$, up to $g_1(s)$. Well, now I can do this I can make an analogous case on the right hand side for a discrete time system, you have u which is discrete now is fed to the system, which has a transfer function $g_1(z)$ which acts as an output the discrete output y_1 fed to that is to second transfer function first transfer function in fact, $g_2(z)$ gives you the discrete output y_2.

So I can try exact same steps here, because if you remember the block diagram, which we wrote in the previous case, our expression was that $y_1(z)$ in fact $y(z) = g(z)\ u(z)$. So I can do this for all the individual transfer functions from where it is not very difficult to see that g overall z would be equal to what? $g_1(z)$ $g_2(z)$ up to $g_N(z)$. For N systems in series, your transfer function is simply the multiplication of individual transfer functions, this is applicable for continuous as well as discrete time domain fine.

(Refer Slide Time: 06:16)

Then what else for a N continuous processes in series with Laplace domain transfer function as $g_1(s)$, $g_2(s)$, and so on, Laplace domain transfer function. The pulse transfer function has given us this formula well did we make use of this fact beforehand. Well, we did what this practice is that g(z) is the z transform of the individual transfer functions in Laplace domain.

(Refer Slide Time: 07:02)

So let us see if $g_1(s)$, $g_2(s)$, up to $g_n(s)$ are the Laplace domain transfer functions then what would be g overall or simply g? Just now we develop this particular expression this would be

$g_1(s)$, $g_2(s)$ up to $g_N(s)$. Now you would like to determine the corresponding pulse transfer function, how would you do that? We said that you whenever you have g(s) you determine g(t). If you remember the scheme which we wrote in the previous lecture, you are in s domain, you go to t domain, from where you go to z domain. So I get g(t), I discretize this, I get g(nT) and from g(nT) I get g(z), the flow which is been later on far right.

With some expertise and experience, I also said that you can write this directly. So, what will I do? I now have g(s) as this multiplication, so I will get the overall transfer function inward that to get the get g in terms of T, discretize it and then again do this transform, which basically means that g(z). Let me write this in this manner, g(z) is what the Laplace transform, the z transform of g s, you see this going from s domain to z domain, which in other words means g(z) as the z transform of $g_1(s)$, $g_2(s)$, $g_N(s)$. So this is what which has been stated.

(Refer Slide Time: 10:14)

What else? In general g(z) transform of $G_1(s)$, one small correction then it has to be a multiplication not a comma. So, $G_1(s)$, $G_2(s)$, upto $G_N(s)$ is not equal to $G_1(z)$, $G_2(z)$, $G_N(z)$ and so on. Let us try to understand what this means. So I have this overall transfer function g of s, which is equal to $g_1(s)$ and $g_2(s)$, up to $g_N(s)$. How do I obtain the corresponding z transform? Let me write this again, S domain to t domain to z domain alternatively directly my flow is very clear. So how do I go from g(s)? Is a multiplication of individual transfer function to g of z. What I do is this and how do I get g(s), I have $g_1(s)$ available with me, I have g 2 of s available with me and so on, I have $g_N(s)$ available with me.

What do I do? I multiply them multiply, multiply, multiply and then I do this, for the multiplication I do a z transform, I get g(z). Let me repeat I had these individual transfer functions available with me, I multiplied them first and then whatever product I got, which is g of s I took the transform of it z transform. I can do 1 more thing, I have $g_1(s)$, $g_2(s)$, $g_N(s)$. Now I can do this thing first, I from $g_1(s)$, I get $g_1(z)$, from $g_2(s)$ I get $g_2(z)$ and from so on from $g_N(s)$ I get $g_N(z)$. And then I do this thing I take this I multiply.

So this is taking z transform, taking z transform, taking z transform and then I multiply them, what do I get? $G_1(z)$, $g_2(z)$, up to $g_N(z)$, which is the correct method. Multiplication first followed by z transformation or z transformation first followed by multiplication are these in other words, are these 2 operations equivalent can you do 1 operation followed by other or second operation followed by the first, would you get the same answer in general you will not get the same answer, the only method to get this answer the pulse transfer from overall pulse transfer function correctly is to first multiply your transfer function in the Laplace domain itself and then do z transformation in the end.

And why does this happen? There is an elaborate proof which is not in the scope of this current discussion, we will not go into those details in this particular course. What needs to be remember this, get the overall transfer function in Laplace domain first by multiplication and do z transformation later, not the other way around.

(Refer Slide Time: 15:01)

Now we all always looked into a single input single output process, did this a transformer analysis, but for the transform domain analysis we did the transform domain analysis for multiple input multiple output systems as well, so is it possible that we do this analysis for multiple input multiple output discrete systems? Turns out that yes, it is possible. Now when you do a multiple input multiple output MIMO analysis of discrete time systems, you get a system of difference equations.

(Refer Slide Time: 15:47)

Let us quickly see how do we get it for single input single output system and then can we extend this. So, for a single input SISO system we can write the corresponding equations as a

$$\frac{dx}{dt} = ax + bu \qquad ...(1)$$

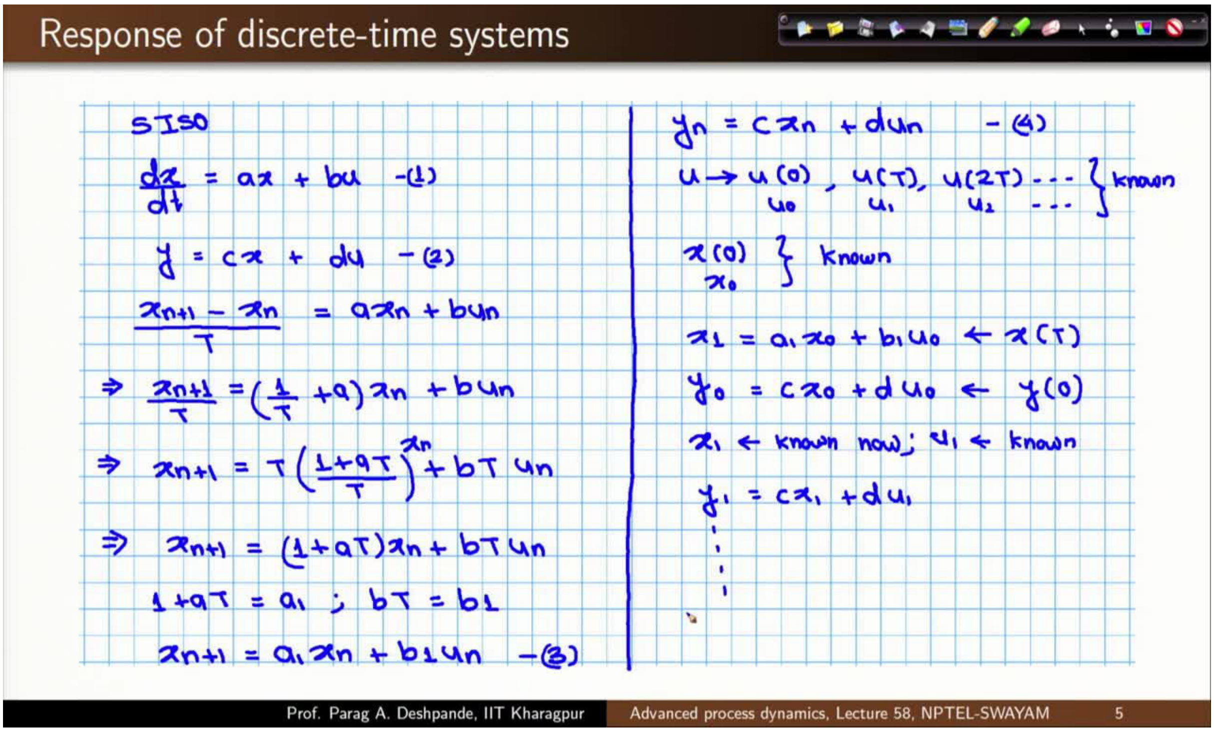
this is the dynamical equation, let us call this as equation 1

$$y = cx + d \qquad ...(2)$$

and, this is the equation for the output variable. So to determine the corresponding discrete time equations for this SISO system, let us first discretize equation number 1 we can write

$$\frac{x_{n+1} - x_n}{T} = ax_n + bu_n$$

Let us do some simplifications this



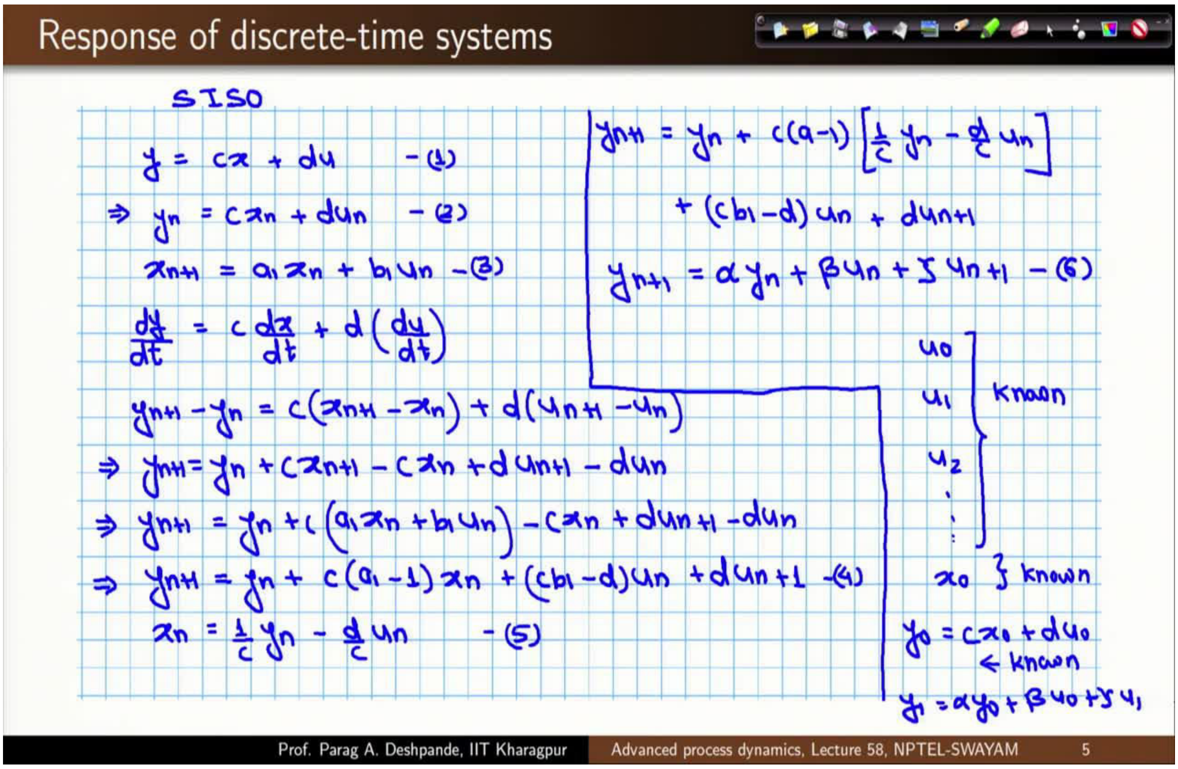
So, the value of the output variable at any given instant of time can be obtained by multiplying the value of the dynamical variable at that instant of time by c and followed by addition of the multiplication of d with the value of the forcing function at that given instant of time. So let us see how this scheme works. You have u and u would be known at, u at 0, u at T, u at 2 T, and so on, and all of these would be known, let us call these as u(0), u(1), u(2) and so on.

The initial condition, because equation number 1 is an initial value problem. So all of these are known. Since equation number 1 is an initial value problem, x at 0 would have be known, so this is x(0), and this is also known. So if I know u(0) and I know x(0), then from equation number 3, x(1) would be what? $a_1x(0) + b_1u(0)$. And since x(0) and y(0) are known, you now know x(1) at time t, this is known. Now what is the meaning of equation 4? y at 0 would be equal to cx(0) + du(0), which means y at 0 would be known because x(0) and u(0) are known.

Now since x(0) and u(0) are known, you can determine you could determine x (1). So therefore, from x 1, which is known now and u(1) which is always known, you can determine y(1) as cx(1) + du(1) and this way you can continue this whole procedure. So the basic method is that you start with the dynamical variable, the dynamical equation, discretize it and we what we learned is that from equation number 3, your value of the dynamical variable at next instant of time can be known from the value of the dynamical variable and the forcing function at the current instant of time. So $x_{n+1} = a_1x_n + b_1u_n$, a_1 and b_1 are constants. Further y_n, the value of the output variable at a given instant of time can be calculated from the value of the dynamical variable and the value of the input function at that given instant of time, so now this helps me propagate my system.

(Refer Slide Time: 23:37)

There is only one thing which is left that you have $y = cx + du$, from where you wrote $y_n = cx_n + du_n$. Which means to determine the value of the output variable at a given instant of time, you need the value of the forcing function as well as the dynamical variable at that instant of time. Again we since we are having a SISO system single input single output system, can we devise a method such that I know only I need to feed only the input variable and I get the output variable directly. Well, that is possible, I have let me refer this to as equation 1, refer this to as equation 2, and $x_{n+1} = a_1x_n + b_1u_n$, this is what we obtain by discretization of the dynamical equation.

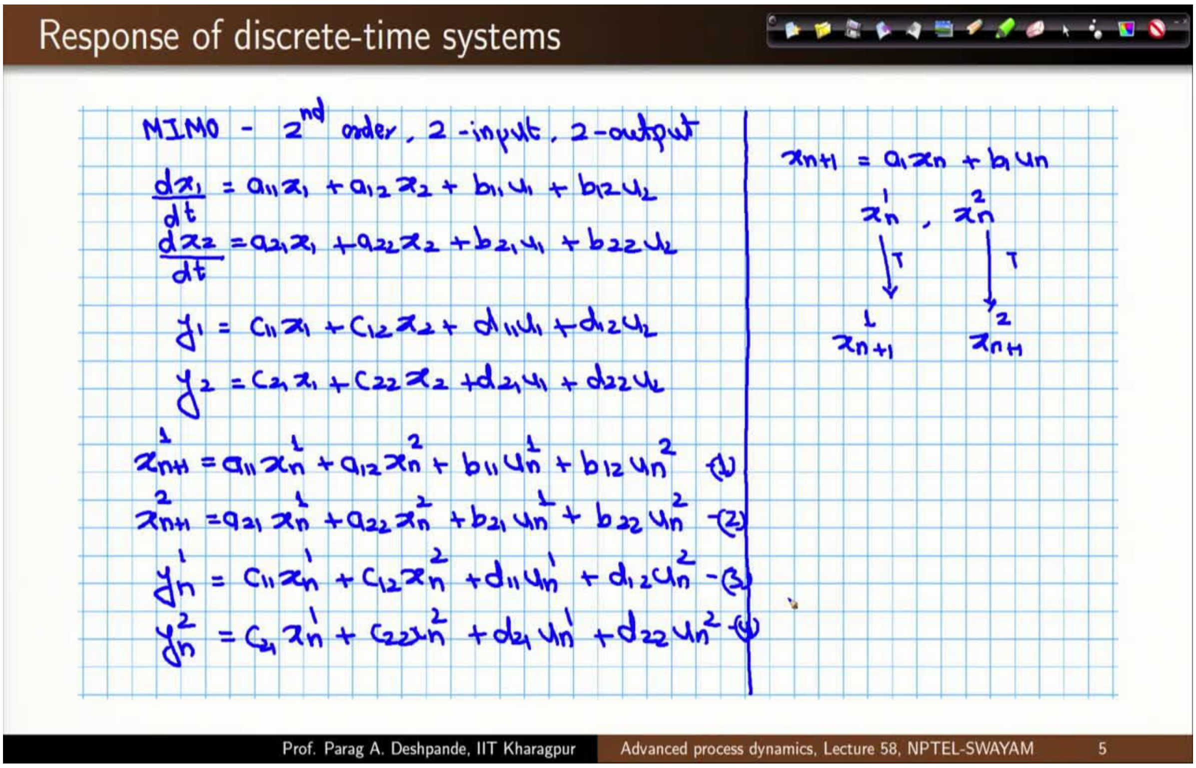


If I know the value of the output function at the current instant of time and the value of the forcing function at current instant of time and the next instant of time, which will always be known, you can know the value of the output variable directly without having to calculate the value of the dynamical variable.

So, u_0 is known, u_1 is known, u_2 is known and so on, all of these are known, because you have the function forcing function. Because you have an initial value problem x_0 also would be known. And since x_0 is known, $y_0 = cx_0 + du_0$ this would be known. And since u_0 is known, u_1 is known and y_0 is known, $y_1 = \alpha y_0 + \beta u_0 + \gamma u_1$ is known and therefore, in this manner, we can propagate our system. So this was the way to convert single input single output system to a difference equation form.

(Refer Slide Time: 30:54)

So now let us extrapolate this to obtain the relationship for MIMO system. So let us consider a second order 2 input, 2 output system, and then we can generalize it to n-eth order m input p output system. So how will you deal with second order input-output, 2 input, 2 output system?



(Refer Slide Time: 36:49)

You can model multiple input multiple output system using the difference equations which we wrote here. And then for multiple input multiple output discrete system, you can determine a discrete transfer function matrix. Let us see the genesis of this discrete transfer function matrix.

(Refer Slide Time: 37:11)

For that let us make 2 input 2 output system, let us make a block diagram. So you have input 1u_1 you have an input 2 u_2, go to the system and you have the ultimate output y_1 and you have an ultimate output y 2 from the system. Now u_1 is acting on the system to give you output as y 1. So I will have a block here and this gives you y_1, the effect of input u_1 on the output y_1. How would you do that? It would be via a transfer function g(z). Similarly, u_2 would act on a transfer function to give you an output y_2 g(z). But now u_1 may act on y_2 on may affect y_2 as well, so therefore, what I need to do is this I will have the way y_1u_1 at via some transfer function g such that the combination of the effect from the block down and the block up will give you the overall effect. So, I have an add-up here, so plus plus.

So I have u_1 to understand this, I have u_1 which is acting on a block g(z) which gives you the effect on output y_1. Similarly, it gives the effect of effect on y_2 via another block. Well I can draw an analogous situation here g(z) and let me make another adder here, plus, plus. So this is the overall flow of different effects. I can do one thing; I can make a large block like this and what I can see is that there are a lot of operations which are going inside this dashed block but all I can see from outside is that I have 2 inputs and have 2 outputs. So, what I can do is, I can assign a block which has 2 inputs and which has 2 outputs, u1 u_2, y_1 y_2 and inside that is a set of operations.

Now what does the top post block give me? g(z), it gives you the effect of input 1 on output 1. What does the next block gives me? It gives me the effect of input 1 on output let us follow the arrow it goes to 2 output 2. Similarly, the third block from top gives me the effect of u_2 on y_1 and finally, the 4th block gives me the effect of u_2 on y_2. So therefore, I will have a transfer function matrix g 2 cross 2, whose elements would be the individual transfer functions $g_{11}(z)$, $g_{12}(z)$, $g_{21}(z)$ and $g_{22}(z)$. So in this manner, it is possible to determine the transfer function matrix of a multiple input, multiple output discrete time systems.

So, what we understood today is that it is possible that you have a multiple input multiple output system and how to deal with such a system? Well, you deal with such a system in an analogous manner as you did with the continuous systems. In continuous systems, you had the transfer function matrix, in this case also you have a pulse transfer function matrix where the elements individual elements of the transfer function matrix would give you the individual effects of the inputs on the individual outputs on the system. So, we will stop here today and we will take an interesting case of analysis of stability of systems in the next lecture. Till then goodbye.