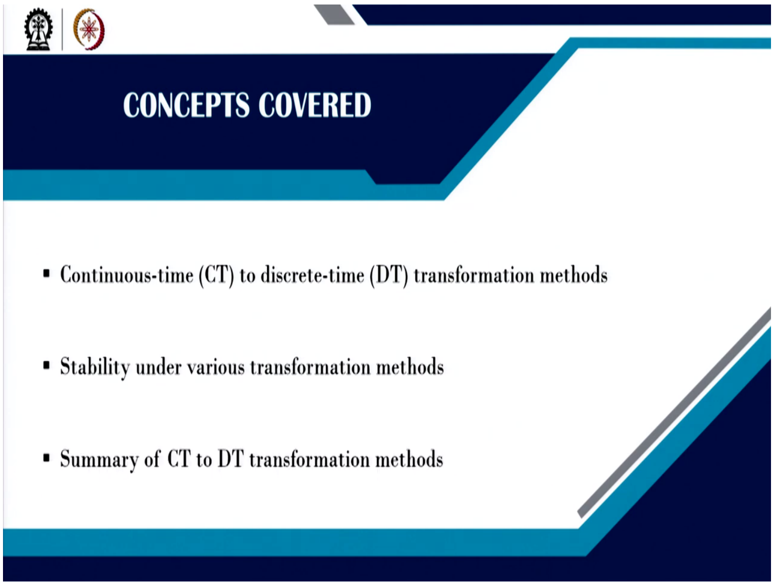
Digital Control in Switched Mode Power Converters and FPGA-based Prototyping
Prof. Santanu Kapat
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Module - 05
Frequency and Time Domain Digital Control Design Approaches
Lecture - 41
Continuous-Time to Discrete-Time Conversion Methods - A Summary

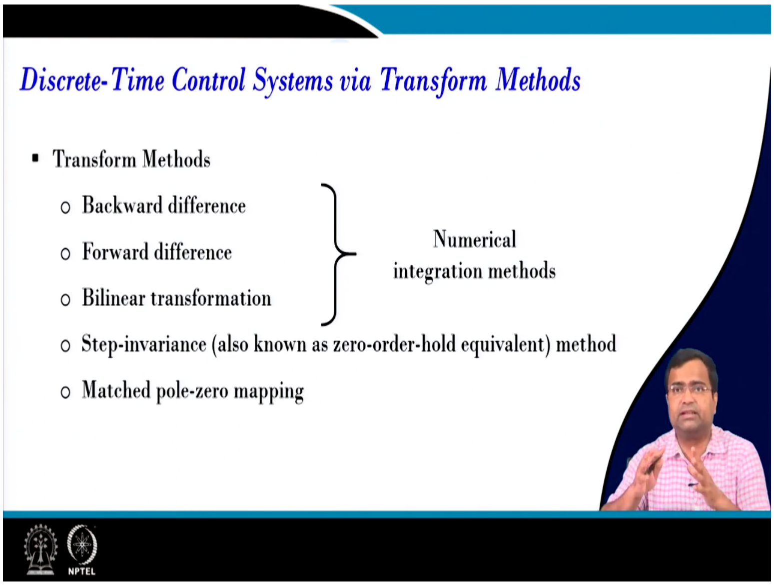
Welcome. So, this is week number 5, and this week we are going to talk about Digital Control Design. And we will start with a design based on a continuous-time model and to understand various transformations in terms of the S to Z domain. In this lecture, we are particularly focusing on various Continuous-Time to Discrete-Time Conversion Methods.

(Refer Slide Time: 00:46)



So, in this lecture we will first talk about continuous-time to discrete-time transformation methods, what are their stability under different transformations, and then, a summary of these transformation methods.

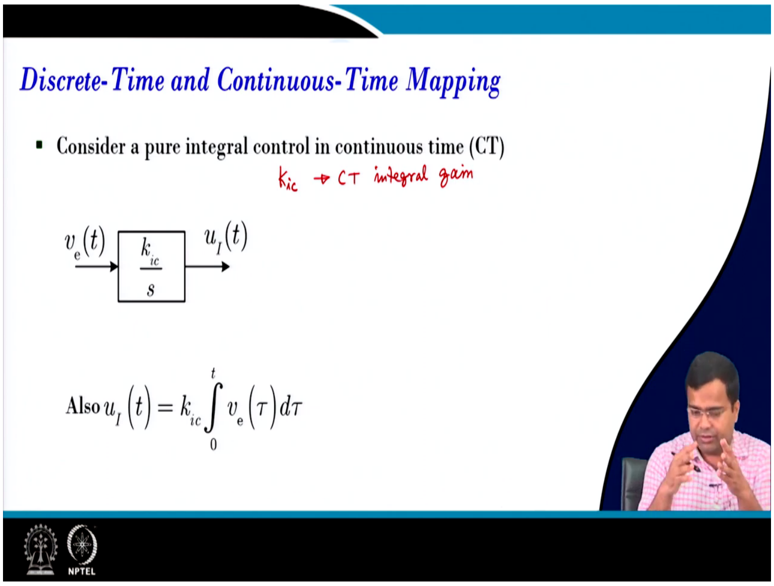
(Refer Slide Time: 00:57)



So, we will start with the discrete-time control system, if you want to design via the transform method what are the available methods? One is the backward difference, another is the forward difference, another is the bilinear transformation, then the step-invariance method, matched pole-zero and numerical.

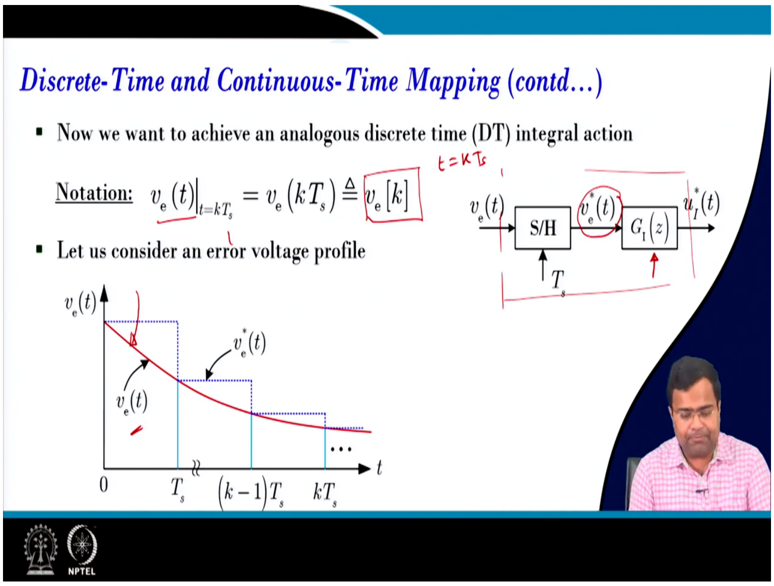
So, the first three methods are the numerical integration methods and also under bilinear transformation sometime you know people talk about frequency pre-warping. So, in this lecture, we want to first show the first three different methods how they are coming and how accurate as well as what is the implication instability.

(Refer Slide Time: 01:36)



So, we will start with a pure integral control in continuous time and we are taking an error voltage v e t and it is passed through an integrated controller with an analog integral gain; that means, k ic which our continuous-time integral gain continuous-time integral gain and consistent of the pure integrator. And we can write the time domain expression u I of t k ic into 0 to t v e T. So, we are trying to integrate over time 0 to the current time.

(Refer Slide Time: 02:11)



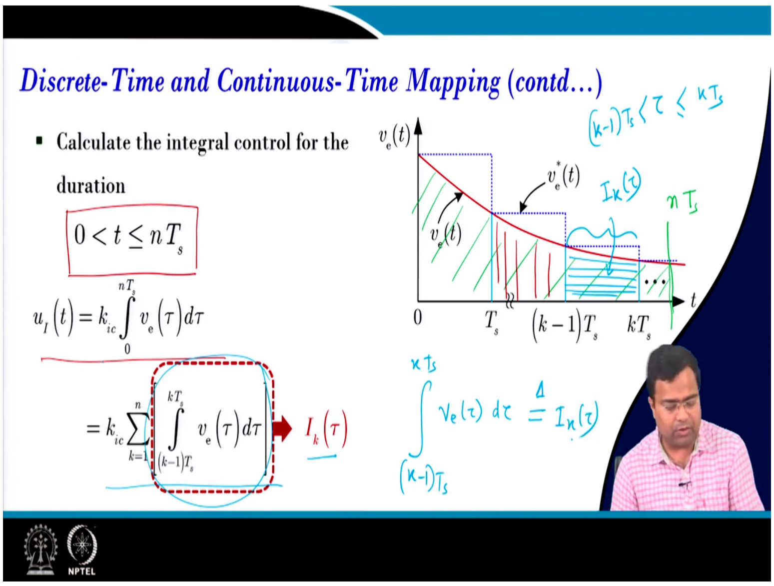
Now, to achieve an analogous discrete-time integral action first we have to draw the sample and hold. So, in digital control, this error voltage passes through a sample and hold circuit and this is the sample voltage error voltage.

And now as if it passes through an equivalent or passes through a digital integral controller or discrete-time integral controller and then we are getting u I of sample t; that means, sample u I of t. So, u star t is a sample version of this and we want to get this transfer function between u sample to this error and we want to show how they are compared with their continuous-time counterpart.

So, if we sample the error voltage which is v of t we are sampling at every edge of the sampling clock with a time period of T s. So, if you take at k_th edge there will be k T s and that is represented by v e k which is nothing, but the sample value of the error voltage at t is equal to k T s; that means, at the k_th edge of the sampling clock.

Now, let us consider an error voltage profile. So, this is an arbitrary profile that we have considered which is shown here v error voltage.

(Refer Slide Time: 03:32)



And if we want to find the integration of the error voltage; that means, we are trying to compute the integral of the error voltage during the duration 0 to n T s; that means, if we take a pure analog control and if we want to get this integration from 0 to n T s where T s is the sampling time, right now we are not sampling. So, you want to compute the total.

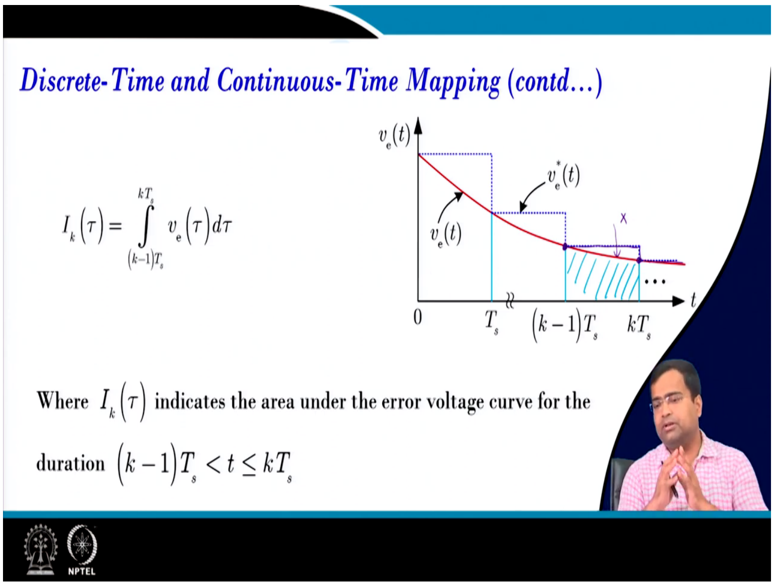
Suppose, we were what we are taking 100 of T s; that means, after 0 to T s T s to 2 T s like that we will wait till 100 T s and we want to integrate. So, then we know the expression will be 0 to n T s then if you further breakdown. So; that means if you consider this integration; that means, let us say we are talking about, maybe somewhere here we will get 100 or maybe we are talking about n T s; that means, somewhere where there will be n T s.

So, we want to find the area under this curve till this n T s time, how can you obtain it? We can obtain this by breaking this into pieces; that means, we can take this first any of this segment we can take any of this segment to let us say we are taking this segment sorry we are taking.

So, we are talking about this particular segment ok. So, this particular segment we are talking about want to get. What is this segment integration we are representing? I k of tau and where tau is varies from k minus 1 T s to k T s and this is exactly what we represent the I k.

So, we can break; that means, we can sum up if we take this window; that means, the sampling window interval and if we take the area under the curve and for each of the segments if you sum them then you will get this complete integration and this particular block we are representing integration I k tau which is nothing, but integration k minus 1 T s to k T s v e tau d tau and that we are denoting as I k tau ok.

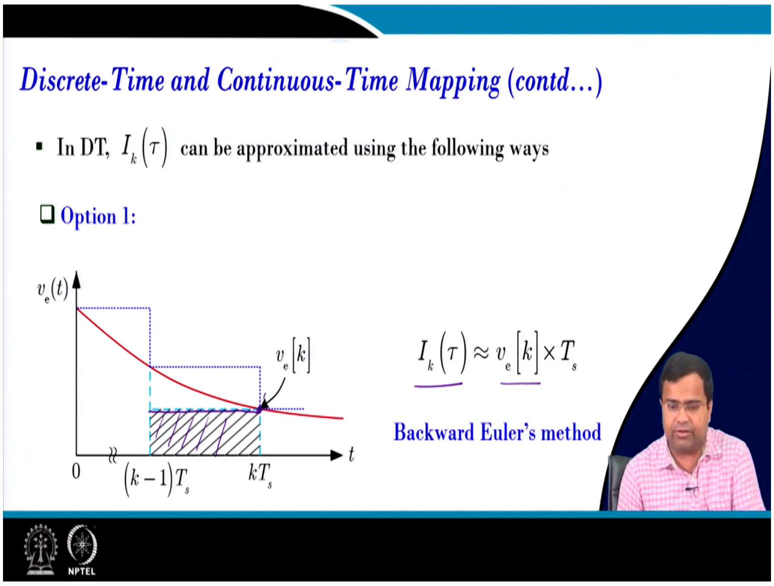
(Refer Slide Time: 06:18)



Now, we want to find this, and; that means, we want to find this exact integration, but if we sample it then we will only have the sample information; that means, for the digital control or maybe for A to D converter or sampler after sample and holds block we will only have this information; that means, the sample then hold, sample and hold, but we will not have any information inside this; that means, we will not have any information in between two samples.

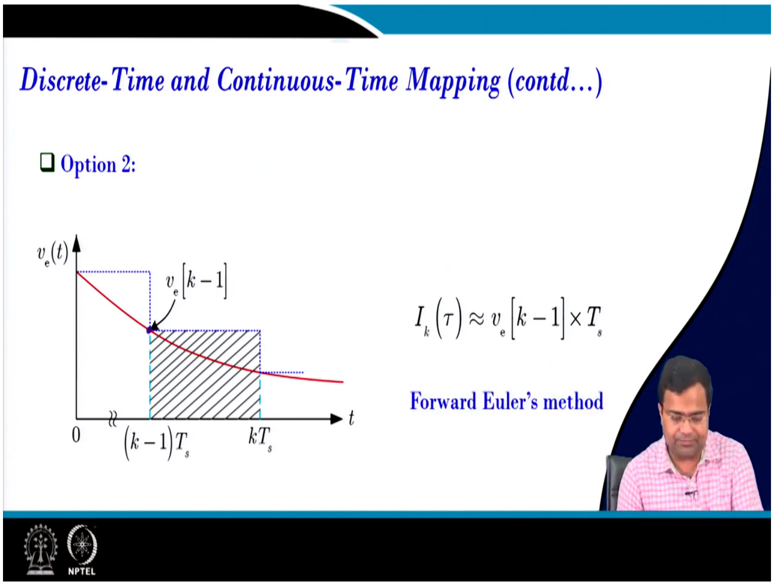
So, we need to find the area we need to approximate this area under the curve using the available sample because we are not talking about a continuous-time error signal, we are talking about a sample error signal and we need to we want to approximate the continuous-time integration using sample quantity.

(Refer Slide Time: 07:15)



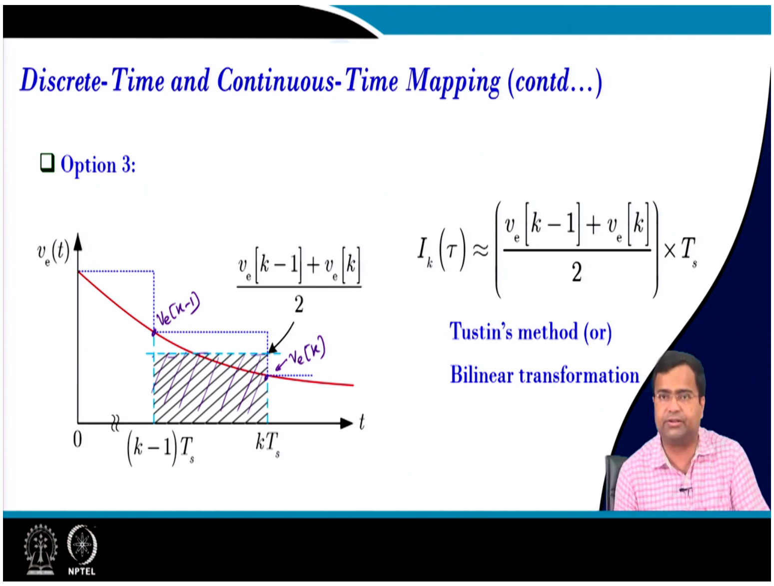
So, first method option 1, if we take this sample and then we using this sample we approximate; that means, it will be constant for this cycle for I mean as per this method we take this sample and then we take this area under the curve; that means, in this case, I k tau can be approximated to be v e k which is this sample value into T s and this method is known as Backward Euler method.

(Refer Slide Time: 07:45)



Option 2, we can also take this sample and find out this area under the curve and you will find this will be v e k minus 1 which is here into T s and that is known as the Forward Euler method.

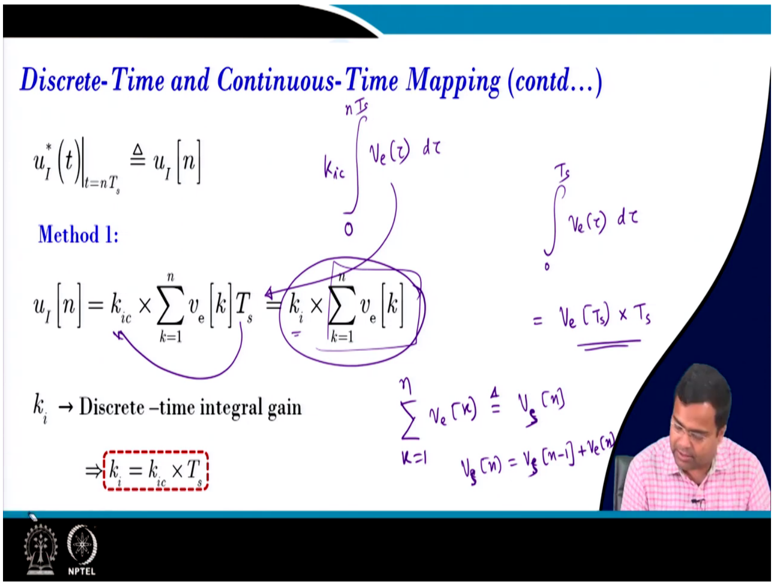
(Refer Slide Time: 08:02)



Also, we can get another way we take both the sample information which is v e k minus 1 and this sample is our v e k then we get the mid value of this and approximate using this particular area under the curve.

And then if you get this area under the curve first of all this mid-value will be the sum of these two divided by 2 and then you can approximate by this value and this is known as the Tustin method or Bilinear transformation.

(Refer Slide Time: 08:34)



Now, in the first method if you want to get an integration; that means, u I of n the n value during 1 to n what can we write; that means, we are trying to get; that means, we are trying to get what T s to n T s we are trying to get as if it is shown and as if we are writing this tau into d tau and there is a k i c term that we want to find out.

So, this can be found by this method we have not used what is v k; that means, this is the method and here k i; that means, we have taken T s out here, if we take then this k i will be k i c into T s. Now, we want to approximate this; that means, we want to find u of I. So, this is like the Euler method, because if you take any k_th interval; that means, k minus 1 T s to k T s we know that using the backward difference method we are taking the right side of the sample.

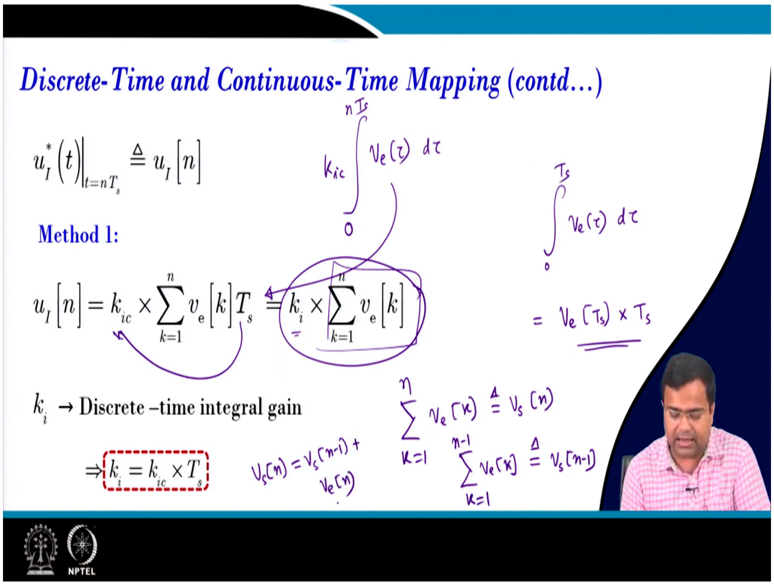
So, it is; that means, we have to simply sum it, but if you continue to sum it using method 1 we are doing, sorry. This interval should be 0 to T s it should be 0 to T s because we are using the backward method. So, even for 0 to T s this cycle if we take using this backward method we are getting v e T s into T s.

So, this is the backward Euler method. So, for some for 0 to n T s it will go up to k because it will take the first sample at the end of the 0 to T s it will take the end value then n minus 1 T s to n T s it will take the nth value. So, that is why this can be approximated.

So, if, you cannot implement it because this will be accumulated and it may saturate your you know storing element because in digital control we generally use a memory block. So, it will saturate. So, we need to use an incremental method or an iterative method.

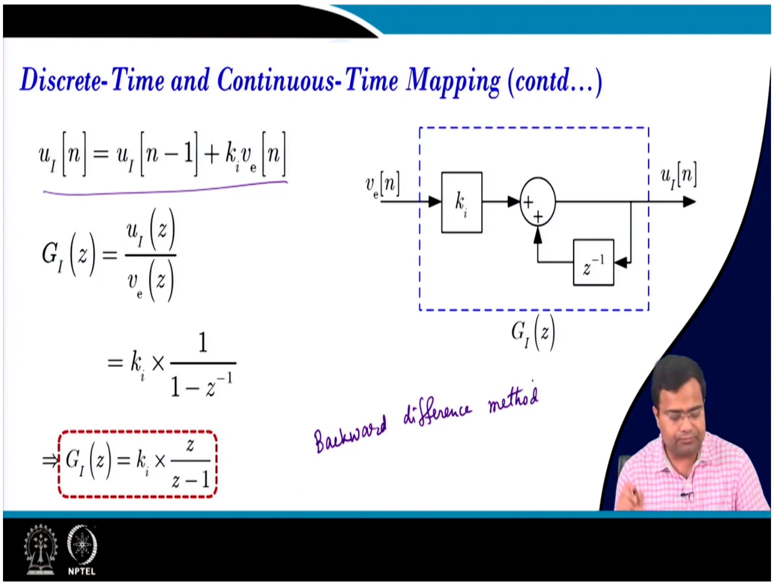
So, here if you write this expression; that means, if you write k equal to 1 to n v e k this can be written as v; that means, if this is denoted as v e n for example, then we can get v e n to be sorry v because this sum. So, v e that sum. So, then we can get v e s equal to v e n v e n minus 1 v e n minus 1 plus v e n.

(Refer Slide Time: 11:54)



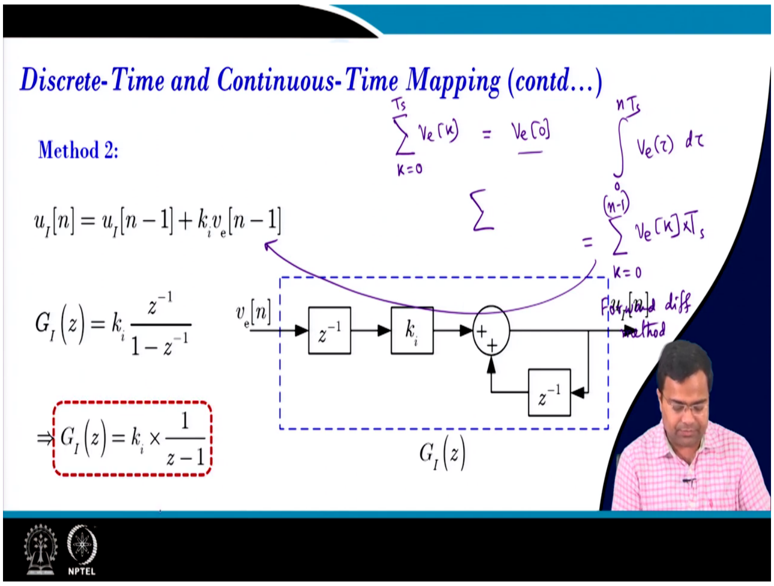
So, that is the iterative method sorry v e s sum; that means, let me write this integral once more. So, this is represented as v e s sum of this whole thing. Then, we can write k 1 to n minus 1 v e k as v s n minus 1. So, then by the iterative algorithm, we can write v e v s of n equal to v s of n minus 1 plus v e n that is it and this iterative method will be used to get this integral value ok.

(Refer Slide Time: 12:23)



And then you can realize by inserting a unit delay and if you obtain the transfer function by this method you will get k i 1 by 1 minus z inverse. So, this is the integral transfer function in the z domain and this is using the backward difference formula backward difference method.

(Refer Slide Time: 12:55)

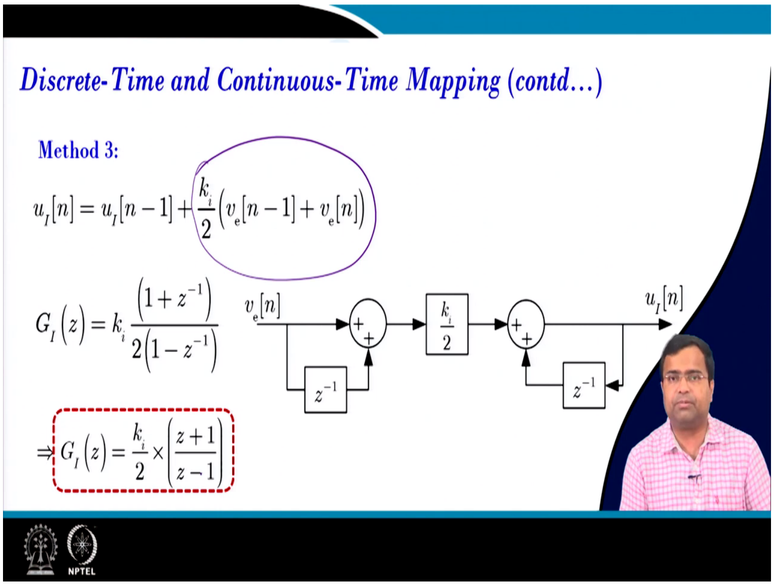


Now, by the same method if you go to the forward difference you see for the kth cycle; that means, what is the forward difference method for k equal to let us say 0 to T s, what is this in forward difference? It will be simply v e 0 we know that right, ok because it is only taking between 0 to T s this value.

And similarly, if we expand; that means if we expand the net duration; that means, we are talking about the integration from 0 to n T s v e tau d tau. So, this can be represented by k equal to 0 to n minus 1 then it will be simply v of k into t v of t. So, if we write then it will be T s. So, this is using the backward forward difference method. So, this is using the forward difference method, and this method if you again write in an iterative algorithm it will look like this ok.

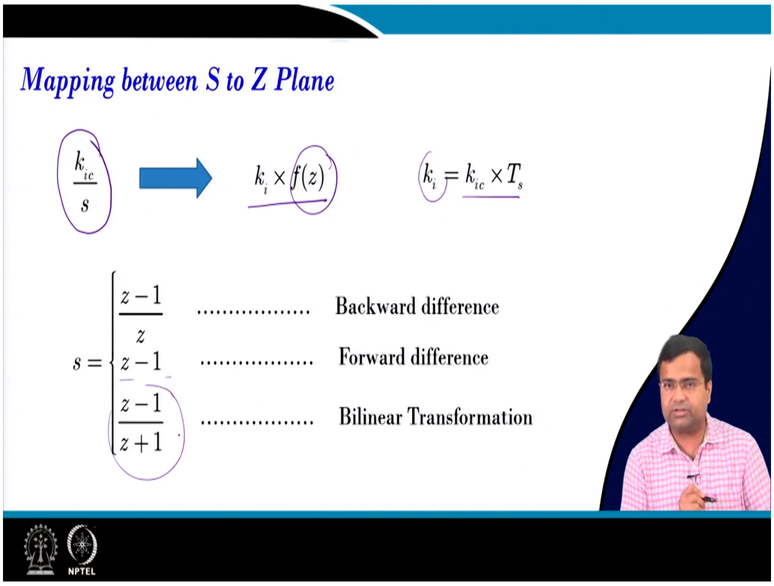
So, you can implement this method by this transfer function, and if you obtain this is the transfer function.

(Refer Slide Time: 14:32)



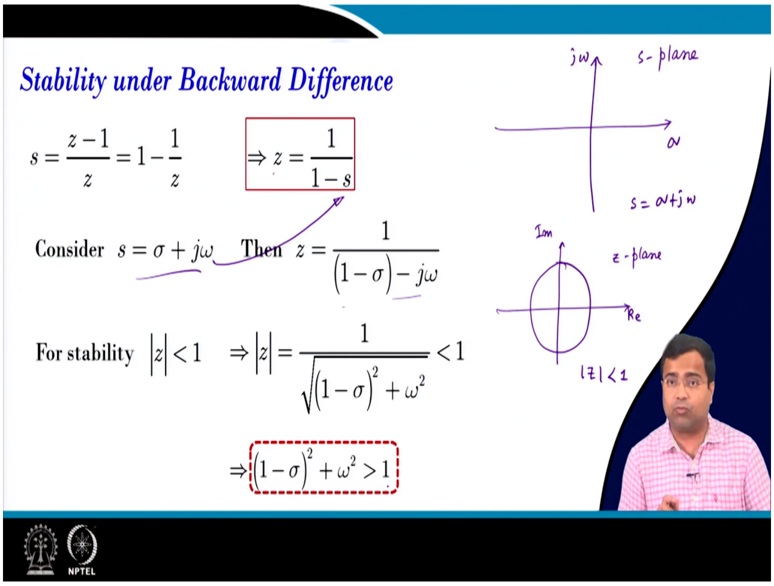
Next by the bilinear transformation, the only difference is the last term and here it will be replaced by the average of these two errors and which is written here if we obtain the implementation of this block and if you find the transfer function it will be k i by 2 z plus 1 by z minus 1.

(Refer Slide Time: 14:52)



Now, if you want to map; that means, a continuous-time integration we started with and we got a discrete-time integration, and k i the discrete-time integral gain is this. What is f z; that means, if we map the S domain to the Z domain for the backward difference it will be z minus 1 by z for forward difference z minus 1, and for bilinear transformation is z minus 1 by z plus 1.

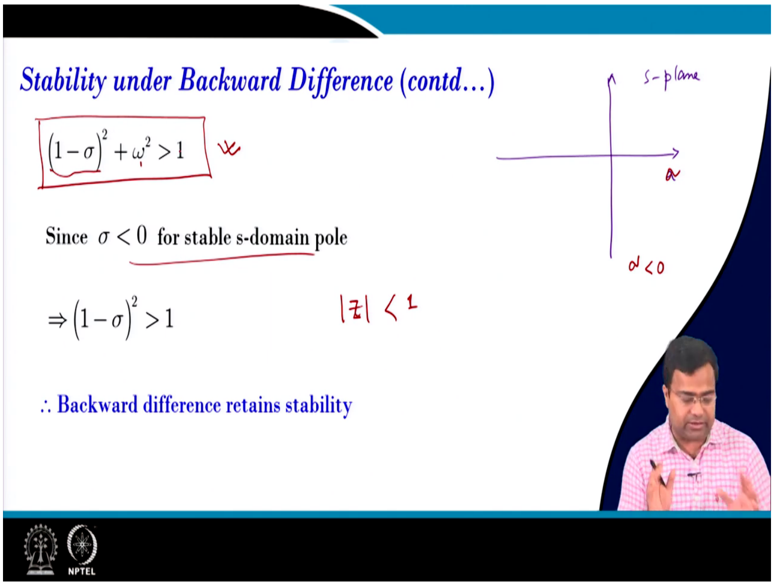
(Refer Slide Time: 15:18)



So, now we want to see stability. If you use forward-backward difference and if you get z in terms of s. Now, we are talking about an s domain this is my s plane, this is my sigma and this is my j omega we know. So, we can write s equal to sigma plus j omega this is the standard term that we have written and we replace this s expression here. Then what we will get z equal to this? And what do we need for stability in the z domain? So, this is in the z plane, this is the real part and the imaginary part. So, it should be within the unit circle.

That means we need mod z should be smaller than unity and; that means, if you take the mode of this function it will come like this and if you simplify the requirement is this 1 minus sigma whole square plus omega square should be greater than 1.

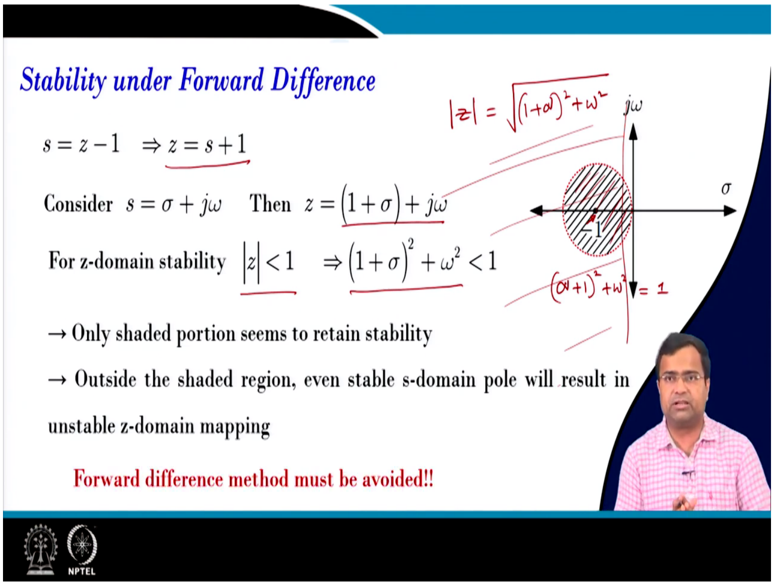
(Refer Slide Time: 16:24)



Now, for the stable system in the s plane ok and this is our real part sigma, we generally for stable case the sigma; that means, the sigma must be negative for the stable case then 1 minus sigma whole square should be greater than 0 all the time because sigma is negative.

So; that means, we can establish that this will be always whether it is 0 or something else, this quantity is always greater than 0. So, we can maintain this condition. So, that will ensure that z will be always in the unit circle; that means, by this transformation method a stable s plane flow will always be stable, but it is a conservative choice because you know this will you know it will make sure it is stable, but it is a conservative choice.

(Refer Slide Time: 17:18)



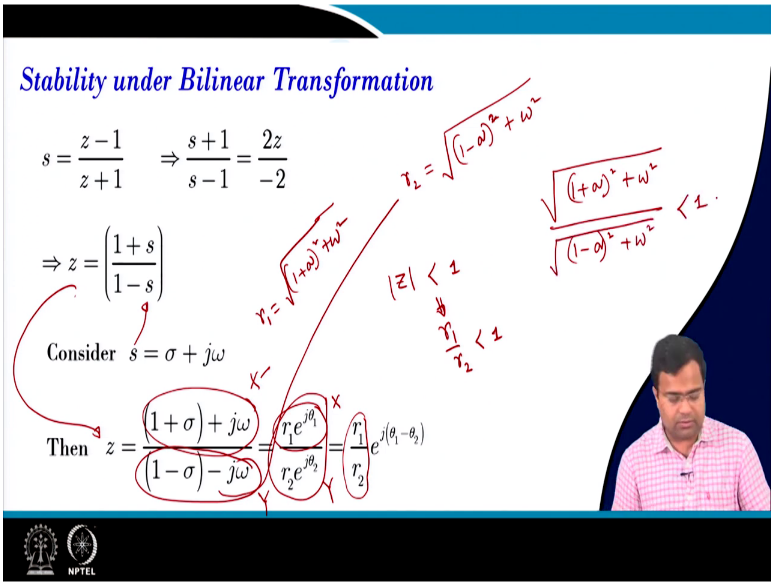
The forward difference we know is that z is equal to s plus 1, now if we replace s equal to sigma plus j omega then we will get z equal to 1 plus sigma plus j omega and we need to ensure that the z mod of z should be smaller than 1. And if you take the mod of this function what we will get? That means the mod of this z means will be the square root of 1 plus sigma whole square plus omega square, and the mod less than 1 means if you take the square it should be smaller than 1.

And what does it indicate sigma and j omega; that means, minus 1 is the radius; that means, it is something like that plus omega square if we take equal to 1. So, this represents the equation of a circle with the center minus 1 0 and that is exactly shown here and with the radius of unity.

So, it can only be stable in this transformation, if we take any pole in this s plane inside this circle; that means, that means because we know that for stability in the s plane, the real part should have a negative real part; that means, for the actual stable plane it takes the entire left-hand side, but other than this shaded region outside it will become unstable in the z domain.

So, this transformation may lead to a stable pole only the shaded region can retain stability, and outside the stable region in a stable domain pole may become unstable in the z domain. So, that is why the forward difference method must be avoided.

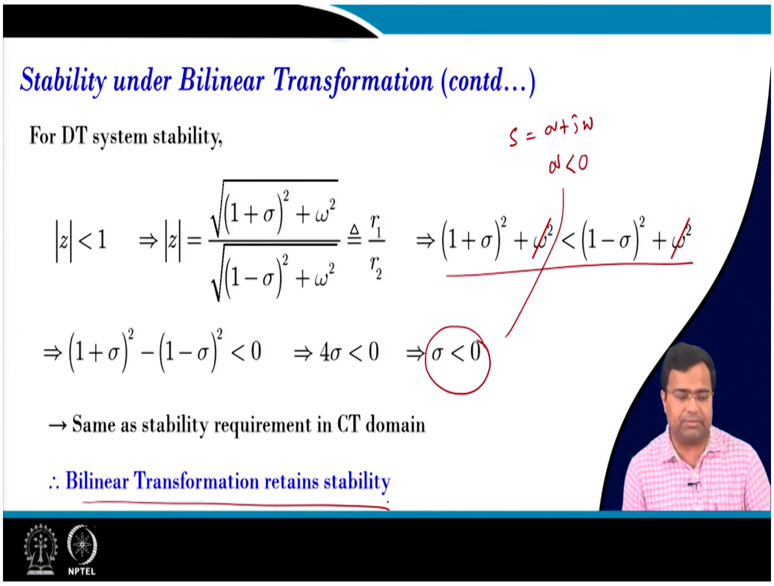
(Refer Slide Time: 19:01)



The next is under bilinear transformation if we write s equal to z minus 1 by z plus 1 and again if you simplify to write z in terms of s, now you write s equal to sigma plus j omega and you replace and you rearrange; that means, you obtain the z from here by substituting s in this expression then you will get 1 plus sigma plus j omega 1 minus sigma minus of j omega.

And if you write in a complex number because this is one complex number let us say X and this is another complex number let us say Y and this polar representation of X and polar representation of Y. So, we can take real part r 1 by r 2, then j. So, we want to make sure that the z mod should be mod of z should be smaller than 1. So, we are interested in the magnitude, we are not interested in the phase. So, we want to make sure that r 1 by r 2 must be smaller than 1. So, this will require this result.

(Refer Slide Time: 20:05)

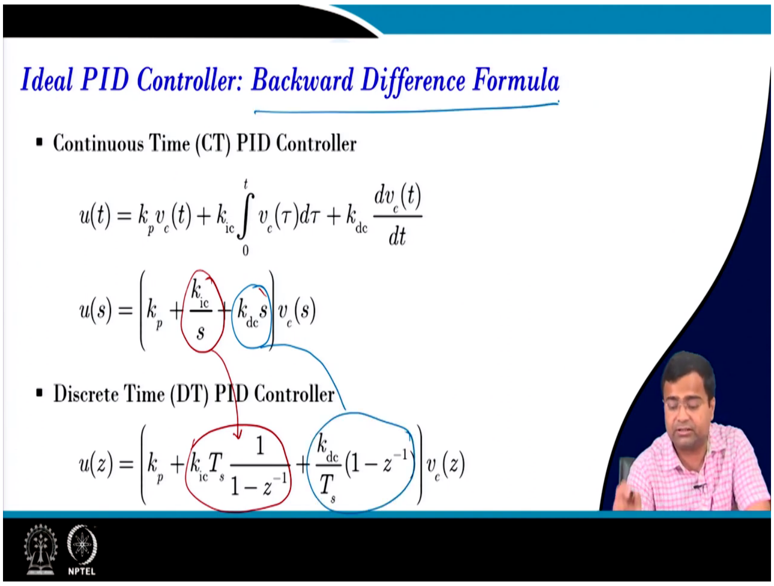


So; that means if we again write what is my r 1. So, if you take the first term, what is the r 1 for this? So, here r 1 will be 1 plus sigma whole square plus omega square the square root. And what is my r 2? It will be it is coming from here right? So, it will be 1 minus sigma whole square plus omega square.

So, if you take r 1 by r 2 it will be 1 plus sigma whole square plus omega square divided by 1 minus sigma whole square. So, this must be smaller than 1 and if we write it here; that means, if you simply square it you need to satisfy this and if you. So, it will end up with sigma being less than unity.

So, its requirement for if you want to ensure a stable z domain pole then it requires the sigma should be negative and that is exactly the requirement of any; that means if you take any s equal to sigma plus j omega for a stable s domain requirement the sigma must be negative and which is constant; that means, in this method, the stable pole in s domain will remain stable in the z domain. So, it will retain the stability of the continuous-time and the binary transformation it retains in the stability.

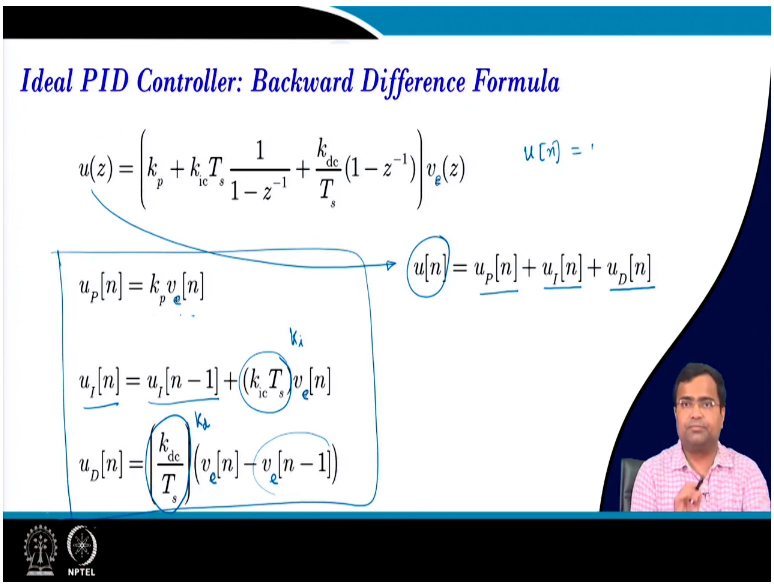
(Refer Slide Time: 21:37)



Now, we want to convert a PID controller from a continuous-time PID controller to a discrete-time. So, this is the continuous-time ideal PID controller parallel form and if we write in the Laplace domain it will be the function with 0 initial conditions. Then discrete domain if we use a backward difference formula what will be our integral gain? I mean we know that this term can be replaced by this term.

Similarly, this term if you take this term it will be replaced by this term and this is we are using the backward difference formula.

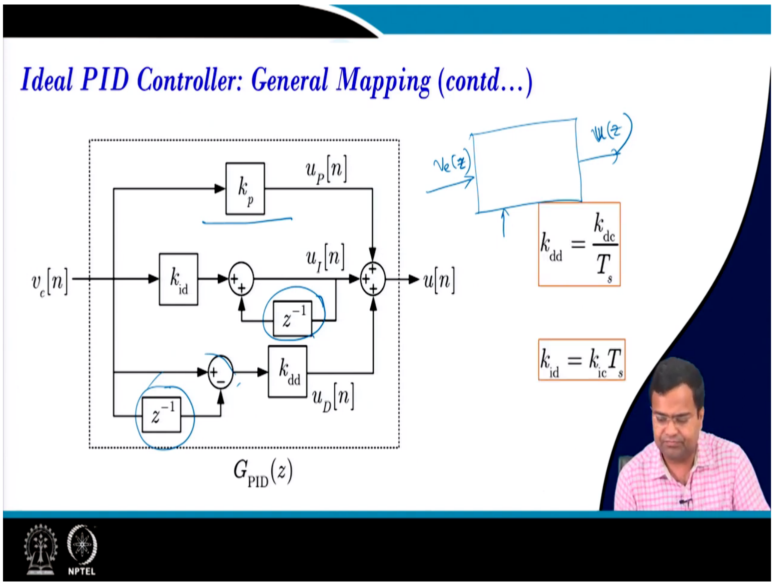
(Refer Slide Time: 22:22)



Then if you write this in the full form we can separate two things; that means if you take the inverse Laplace u of n, I want to write yeah it is written here which is the inverse Laplace of this it is coming from here. So, as if it is a PID controller it has three components, the component due to the proportional action, the component due to the integral action, and the component due to the derivative action.

What is up with n? It will be simply k p into v c n; that means, there is an error which is common this is my common error signal. What am I? We already know that under backward transformation that means difference formula, u I of n is nothing, but u I of n minus 1 plus this term is nothing, but discrete-time integral gain and this is of course, an error. And then the derivative is nothing, but the discrete-time derivative gain and k dc is the continuous-time derivative gain by T s and it will be v error n minus v error n minus 1.

(Refer Slide Time: 23:43)



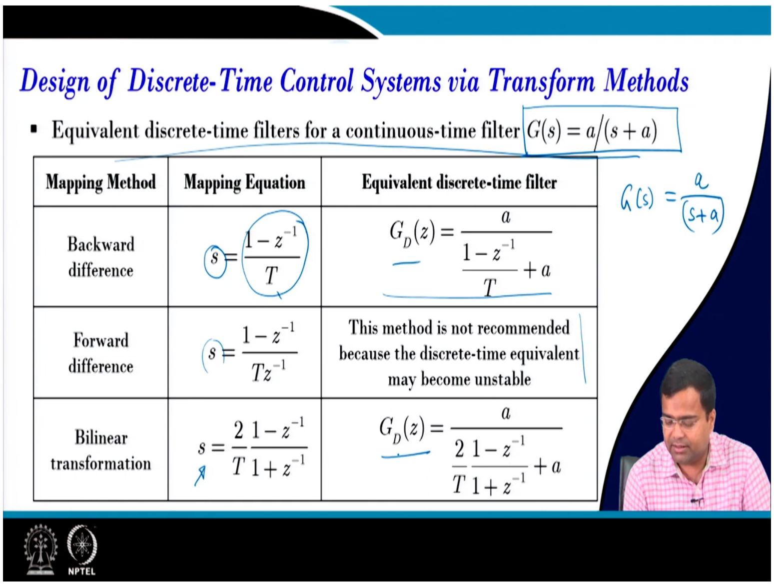
So, this is the formulation. So, if you want to realize this ideal PID controller so; that means, if your first term is straightforward it is simply going. In the second term, you have this one z in the delay term which is here in the feedback path and in the third term you need a delay and then subtraction. So, you need a delay this is delayed input and then a subtractor, multiplied by this digital derivative gain and this is the digital integral gain.

So, this can be realized by a register and we want to realize using MATLAB, you know I think we have already realized in the previous week, how to realize this you know block level MATLAB implementation. So, one can write the transfer function the whole transfer function as the v e z in the z domain and v u of z, but you know we have also learned if we want to delay this signal because we want to sample the signal at the edge of the clock that we want, we want to customize then the strata if you write this transfer function in the MATLAB you know it will be difficult to manage the edge.

That means it will compute at every 0 times T s time 2 T s time like that, but suppose you want to compute somewhere in between 0 to T s time and we want to only update the controller, to do that we have also learned how to use how to implement MATLAB block using difference equation.

So, this is a z inverse block and if you use this difference you can also realize this using the difference equation simply by plugging these three equations you can difference equation that we have also learned. And this delay will be adjusted based on the clock edge which will be used to update it because this controller can also be implemented for the varying time period for variable frequency modulation, which also we have learned ok.

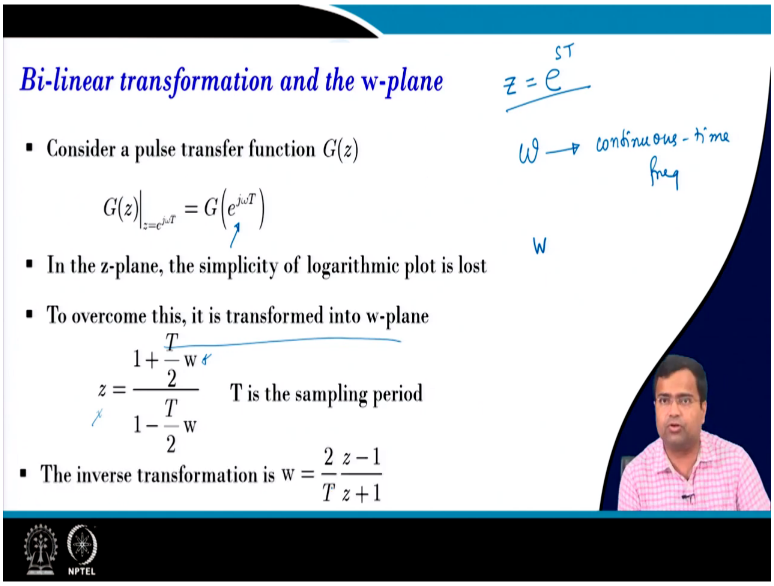
(Refer Slide Time: 25:52)



Now, design of the discrete-time. So, if you want to design then equivalent discrete-time for a continuous-time filter; that means, G s 1 by s plus a. Suppose if you this is the continuous-time filter that you want to design which is a kind of low pass filter s plus a, then how can we get this digital version?

If you use the backward difference we simply replace s equal to this term that we know and there is a sampling time so, it will take this form. If you use forward difference then we replace s equal to this, but this method is not recommended because it may lead to an unstable system which is why you are not writing the z domain expression. Or you can use bilinear transformation where you can get G D of z in terms of you to know you can simplify this expression.

(Refer Slide Time: 26:46)

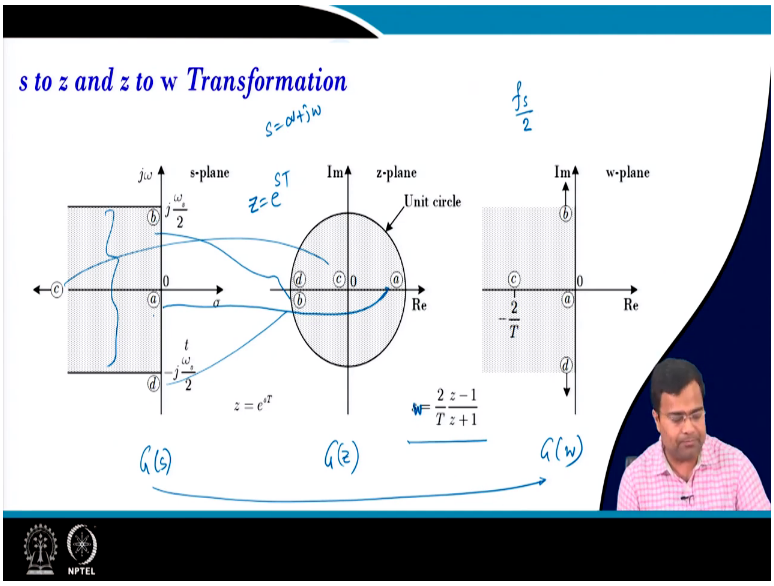


Now, this bi-linear transformation is when you get a z domain transformation; that means if you know that which is called pole-zero mapping it is z equal to e to the power S T. That is the exact mapping and that comes from the fundamental Laplace transform theorem; that means, if you go for you know how to obtain the transfer function using the Laplace transform and if you go to z transform by two basic theorem you can relate z equal to e to the power S T.

But this is an irrational function; that means if you want to retain the logarithmic property and this is the irrational transfer function. So, you cannot plot it. So, to get a rational transfer function you need to approximate and that is done by the w domain. And this is simply similar to a bi-linear transfer function, a bi-linear transformation where z which is the irrational function of omega in the continuous domain can be converted into a rational transfer function of the w domain where the w domain and omega are somewhat analogous.

That means you know when you say this omega it is a continuous-time frequency continuous-time frequency when you say w it is a transform domain from the z transfer function ok.

(Refer Slide Time: 28:20)



So, we want to and this is the inverse transformation also fine. We want to make sure how is the mapping. So, if you take s plane j omega b a this is the point that you want to map.

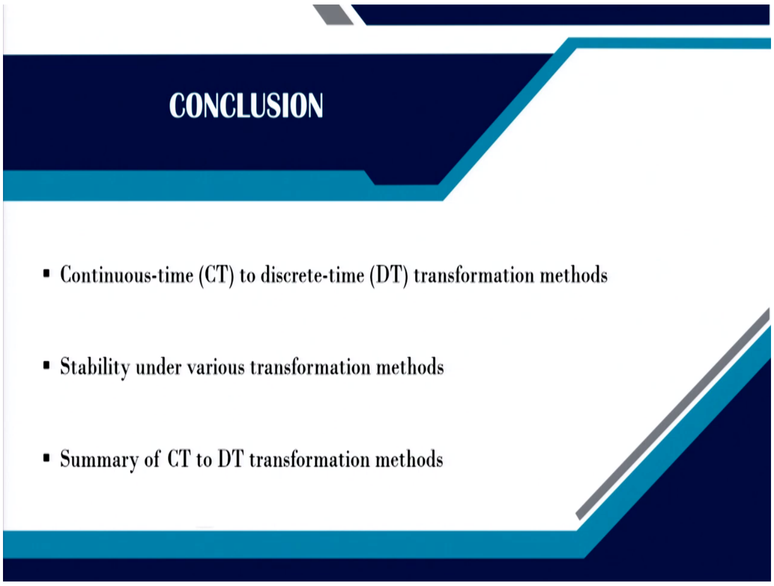
So, if you apply z equal to e to the power S T, where S is equal to sigma plus j omega, then you can get this a coming to be here, then you can get b will be somewhere here, d will be here, then c will come here like that. So, it will be one-to-one mapping for the primary strip and this strip is known as the primary strip. Because it is valid from omega j omega s by 2 by minus j omega s by 2, beyond that you know there will be a repetition. So, within the primary strip, there will be a one-to-one mapping.

Now, when you go to the w domain by using I would say it is rather this is the z equal to w domain it is in the w domain. So, in the w domain you will get almost similar; that means, it can be shown more or less the w domain and the omega is the continuous-time they are more or less close except for the problem of pre warping problem frequency warping problem and that can be taken care by the frequency PRP.

So, in summary, the omega domain and w domain are somewhat analogous where the w domain is obtained from the z domain using similar to a bi-linear transformation, but it will introduce when you convert from z to w domain, and if you; that means if you take a transfer function in G s and if you get G z and then if you get G w then between G s to G w you may get one extra 0 due to this sampling that extra 0 will come at sampling frequency by 2.

So, that will introduce an additional 0 using a transformation, otherwise, they will match more or less retain the property.

(Refer Slide Time: 30:22)



So, in summary, we have discussed the continuous-time to discrete-time transformation method, then we have discussed stability under various transformation methods and we have summarized the continuous-time to discrete-time transformation method and we have learned that the back-forward difference method should be avoided, otherwise the stable plane can become unstable in z domain.

And we have also learned that you know some implementation aspects and we learned the forward difference is a very easy sorry backward difference and is simplified, but it is somewhat conservative, bi-linear transformation is a more accurate method, but it will increase the hardware complexity.

So, in the subsequent lecture, we will learn how to design a controller using this transformation method, and then finally, we will also want to see using an accurate discrete-time model how to design, I want to compare this design using the transform domain and directly designing the z domain. So, that is it for today.

Thank you very much.