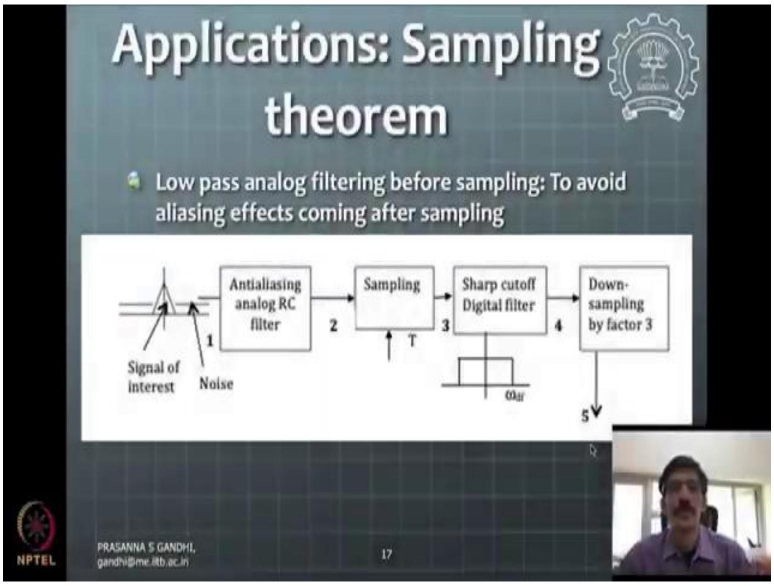
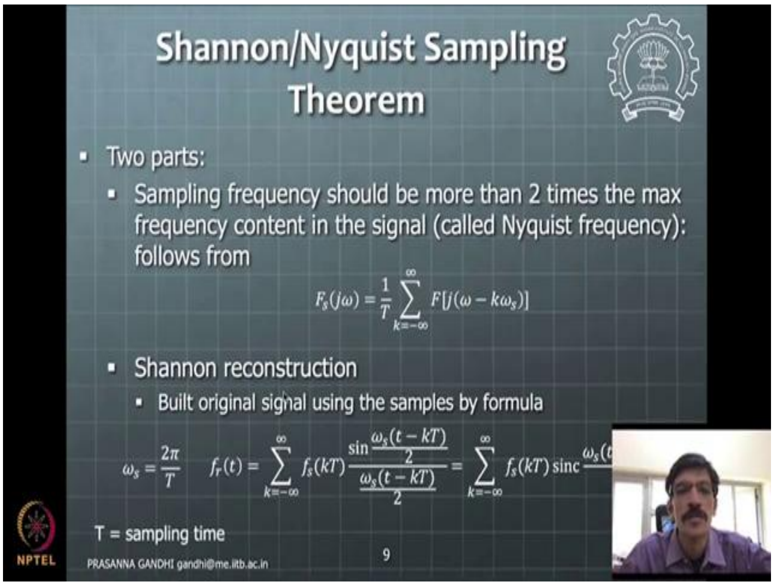
Design of Mechatronic Systems
Professor Prasanna S. Gandhi
Department of Mechanical Engineering
Indian Institute of Technology, Bombay
Lecture 45
Digital System Representation and Filters for Mechatronics

(Refer Slide Time: 00:16)



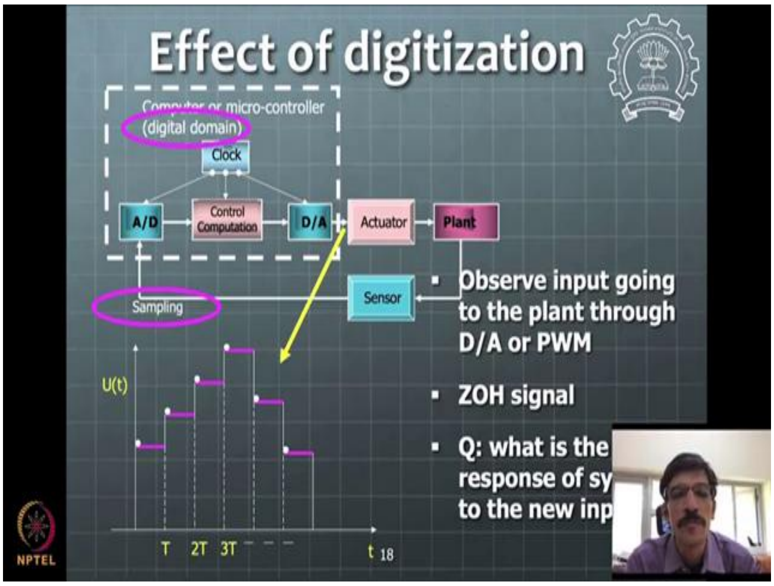
So, in the last class we are looking at sampling and this application of the sampling theorem. So, the sampling we saw two aspects like one is the sampling theorem itself.

(Refer Slide Time: 00:37)



And other one is about this reconstruction. So, ideal reconstruction will be the Shannon reconstruction and then this sampling theorem itself. That the signal needs to be sampled at least twice its frequency contents to get there, get it to the final form which digital form which is representing the original signal content. If it is not done then the signal contents are all lost ideally, we will need about 3 to 4 times the sampling frequency to really represent in the practical case scenario.

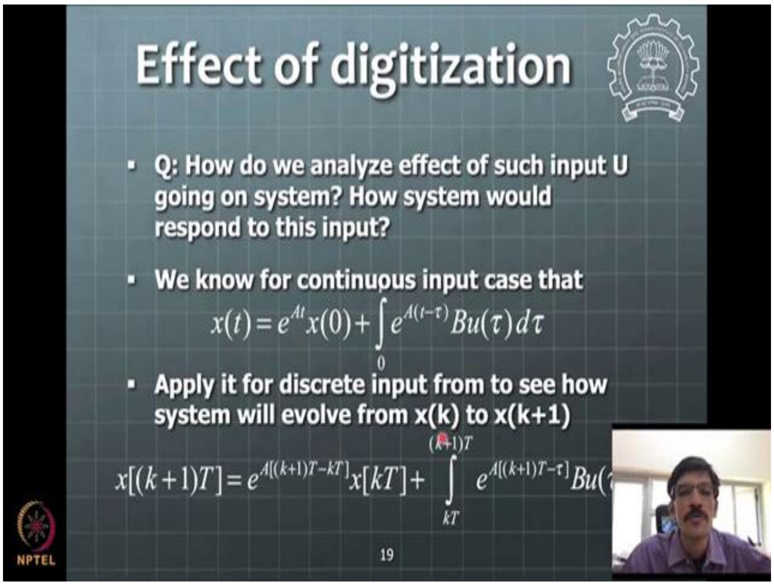
(Refer Slide Time: 01:20)



So now, we began with you know further to these we began with the digitization of the system. So, what it means is when you sample a system you represent them as these kind of dot samples that are point sample points that are represented here let me get the mouse right and these samples are held constant over the sampling period which is T.

Now, if this kind of a output or input goes to the actuator or your plant how the plant behaves is what we discussed in the in the digitization. That is what will happen when the system is digitized or like you know if you want to mathematically represent this continuous system which is like getting this kind of a input which is steppy kind of a form of input. This kind of a steppy form with this value and constant over the time interval then, what is its effect on a system that is what we started studying.

(Refer Slide Time: 02:41)

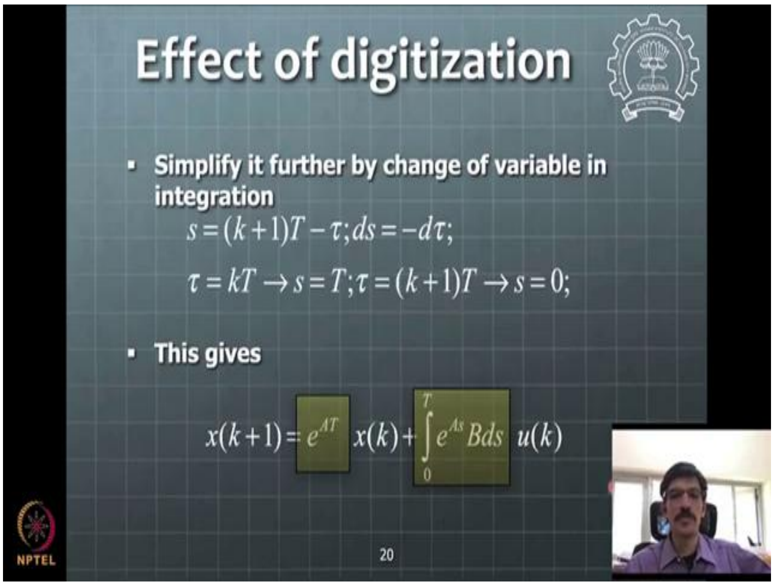


And to that effect we saw that like for any continuous domain system this is a solution of a system this is a system of the kind $\dot{x} = Ax + Bu$ and this is solution. The solution has two parts one is like a homogeneous integral homogeneous solution another is a particular integral part. Now, we apply this solution between the two time instances where we start of with like times and kT, T is the sampling time and we predict the state at (k+1)T.

So, within this time the system will progress within this time interval kT to (k+1)T which is the time interval of T and there we will get this contribution of the homogeneous part of the solution and then this is a particular integral part of the solution.

Now, notice here like we integrate not from 0 to T but now this since we are starting at kT this will be kT to (k+1)T and this 0 state here is x(kT). And similarly, you will find that this final time is (k+1)T and then τ remains as it is because it is an integral variable and then all these terms remain the same. And then we shift the time to this time variable or τ variable defined in some different way shifted way.

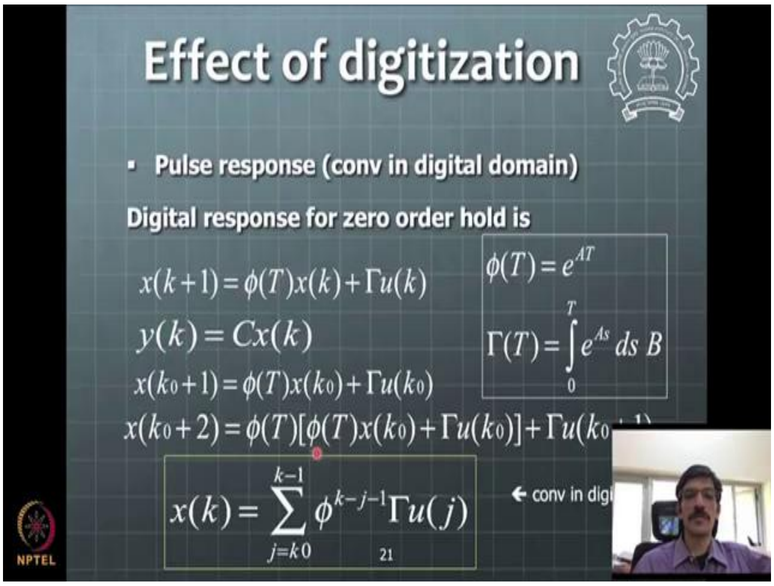
(Refer Slide Time: 04:28)



And then we simplify this to get into this final form. So, x(k+1) now is e^{AT}, A is the system matrix for the original analog system $\dot{x} = Ax + Bu$ kind of the system and capital T is your sampling time x(Tk) I mean the $e^{AT}x(k)$ *and then* $\int_0^T e^{As}Bds$.

So, this is how you get these two matrices which are independent of state there is no x coming in this at all. So, these are like system characteristics matter system this characteristics will depend upon matrix A and the sampling time that you have used something time T.

(Refer Slide Time: 05:32)



So, with this we have now the system coming in this form $x(k+1) = \phi(T)x(k) + \Gamma u(k)$

. This ϕ is this is e^{AT} matrix and Γ is integral of $\int_0^T e^{As} ds\, B$. So, is like B is not a function of S so it can come out of the integral and this is integral of this matrix definite integral 0 to T and multiplied with the vector B.

Which is a system input vector. And $y(k) = Cx(k)$ that remains as it is. And now one can iterate this further like let us say x of say we start at same state k0 then,

$x(k_0 + 1) = \phi(T)x(k_0) + \Gamma u(k_0)$

Then, $x\ (k_0 + 2)$ will be a similar way and we substitute for $x(k_0 + 1)$ here and like you get is again like this kind of a function here.

And you collect these terms in general for starting from k0 so k0 is equal to 0 to k so this is becoming some kind of a summation. Now, you see this power here which is ϕ^2 coming here or like you know $k_0 + 2$ kind of thing and that is how likely this will become now you just see this carefully for each term there will be now higher and higher powers that will be coming up for ϕ and Γ here.

So, so you can see this $\phi^2(T)x(k_0)\ and\ \phi(T)\Gamma u(k_0)$ is coming. And if you have this k0 state the initial kind of a state of a system to be 0 so we consider many times for transformation also 0 initial conditions. Then like now this will be just in terms of $u(k)$.

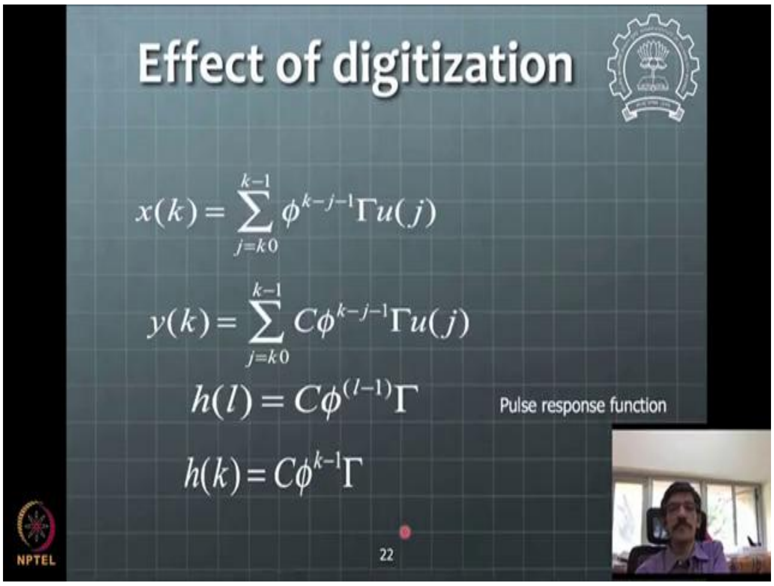
So, this expression assumes that there is a there is a 0 initial condition otherwise this term $\phi^{k-1}x(k_0)$ would be additional term that will be coming here. So now, let us what is the significance of this is that like we can get a state x(k) given u and this is matrices alone this matrix ϕ and Γ are known then based on that those matrices we can get you know the response of the system to any of any input in general any input u.

Now, all of this is now happening in the digital domain form. So, we are getting this kind of discrete values of the state vector at every time instance or like this sampling time index which is k. So, this is how the system response will come out and if you see this carefully this is similar to what we have as a convolution in the discrete domain in the continuous domain.

So, this is a convolution in the discrete domain similar to convolution in the continuous domain. So, this part here if you see as (k-1-j) kind of a form and this is a j form. So, j and (k-j) kind of forms are there so with this one can predict what is this function with which input is convolved to get your final output.

So, the same way we have conclusion in the analog domain you can see the convolution in the discrete domain and then then the function that is convolved with becomes later impulse response of that system. So, in discrete case it is called a pulse transfer function.

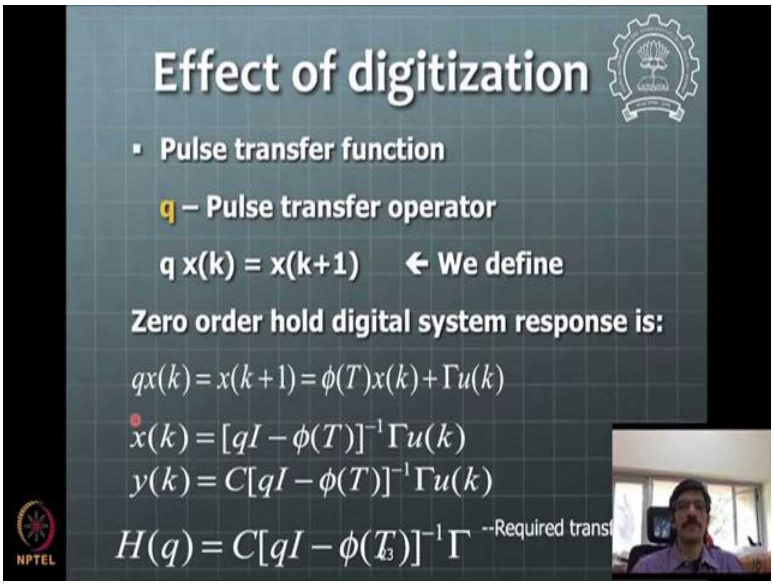
(Refer Slide Time: 10:13)



Pulse response function not transformation pulse response function. So, if you see this in the in the convolution kind of a way you can see that $C\phi^{l-1}\Gamma$ this function is pulse response function. So, so this is like input u(j) convolved with h(l) which is impulse or pulse response function then you get your output y(k).

It is one kind of a concept here. Now, further to that now we seek now a way to get some kind of a transfer function in the digital domain. Now, to get that we need to like define some new quantities. We will come to that.

(Refer Slide Time: 11:12)



So, to get our transfer function so we define this operator q which is called pulse transfer operator in such a way that when the q operates on some x(k) then it shifts the state by 1 sampling instance so $q^2x(k)$ will be $x(k+2)$ like that. So, q is like some kind of operator which we have defined now to get some kind of a forms for further for further transformation.

And we get now the way we get things in in the in the continuous domain in terms of the variable s which is a Laplace domain variable s. We now get this you know equations are like the transfer functions in the in this operator in the in the operator variable in q.

So, if you now see again our previous system, we had x(k+1) is equal to this was our derived equation previously if you see these two steps back we have this equation $x(k+1) = \phi(T)x(k) + \Gamma u(k)$. So, this equation now can be written in the form of this operator.

So, $qx(k)$ will be equal to all these things. So, if you combine terms with x(k) together then $(qI - \phi)x(k)$ will be equal to $\Gamma u(k)$ and then is inverse times like when you get these. Now, we know the output is $y(k) = Cx(k)$ so you substitute for this x(k) here and you get this as an output.

So now, this becomes the input output relationship and if we define the transfer function as a ratio of y(k)/u(k) then what we get on this side here is a transfer function in terms of this new operator q. So, this is a this is a transfer function of a of a digital domain system. So, this is how we start getting now the similar kind of analogies which we have in the similar to what we have in the continuous domain.

So, one can see that look how we suppose we have the system $\dot{x} = Ax + Bu$ and $y = Cx$ from there we can get to this kind of a transfer function H(q) basically using ϕ which is equal to e^{AT} where T is a sampling time and Γ is integral of $\int_0^T e^{As} ds\, B$ that is a integral of this term.

So, this A times B that means that B matrix is coming here and A matrix is coming here and C matrix is here like that we have used all these three A,B,C matrices to get to the final transfer function or in the digital domain. So, this is how we proceed to get a transfer function in the digital domain and this transfer function can then have definitions of poles and zeros and all those stability conditions which come up.

So, we will not get into those details right now we what will focus on is now how do we use this as in the in the in the application of filters. So, this is one of the ways one can see that the continuous domain system can be converted into a digital domain system. And the issue remains like how do we implement such a system or suppose you want if you have defined now say last class we saw that filter is nothing but some kind of a transfer function through which that when the input is passed then some output will be produced which is can be seen as a filtered output.

So similar way we have this digital domain transfer function we pass some input to this function and then this may act as a filter also. So, you can define this system to be a filter like you know $\dot{x} = Ax + Bu$ can be your filter which is have some certain kind of a form of matrix A and B and one can now look at this as a the filter transfer function.

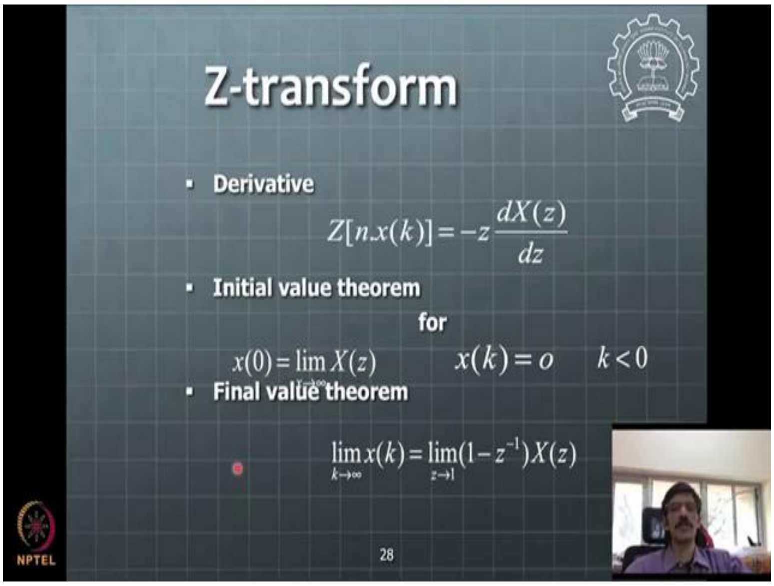
So, we will see how this can be used in in actual implementation of a filter when we look at that. So, we need to kind of have a sense of this shift operator layer here shift operator here and like this is same kind of operation that happens with the definition of Z transforms.

So, we will not get into too much of a digit transform domain things I mean there is formal definition mathematical definition of Z transform and there are some certain conditions under which the system can be transformed from you know the discrete kind of a form to Z domain and get a lot of these details of like digital domain system will appear there also.

So, this is this using pulse transfer function pulse transfer operator is one way of converting system from continuous domain into digital domain and Z transforms will be another way to do that. There are some certain differences although like you know this property is same for both operators there are certain differences between the two which we will not get into right now.

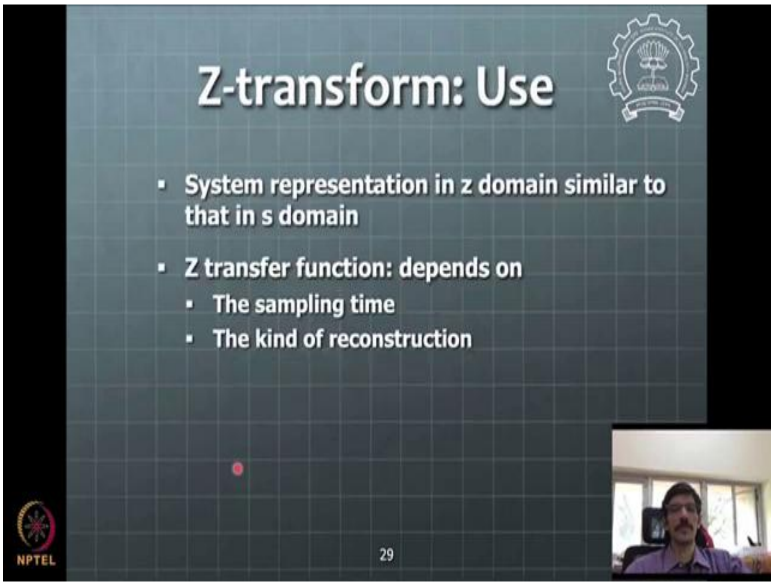
For practical purposes they are not really so important. So, as far as possible we prefer to kind of use the system with this kind of a pulse transfer operator because it has some kind of a you know good physical sense of what is happening in actual practice that is getting reflected in this in a very neat way which may not happen with the with the Z transforms. So, this is one part that is what we will we observe here.

(Refer Slide Time: 18:36)



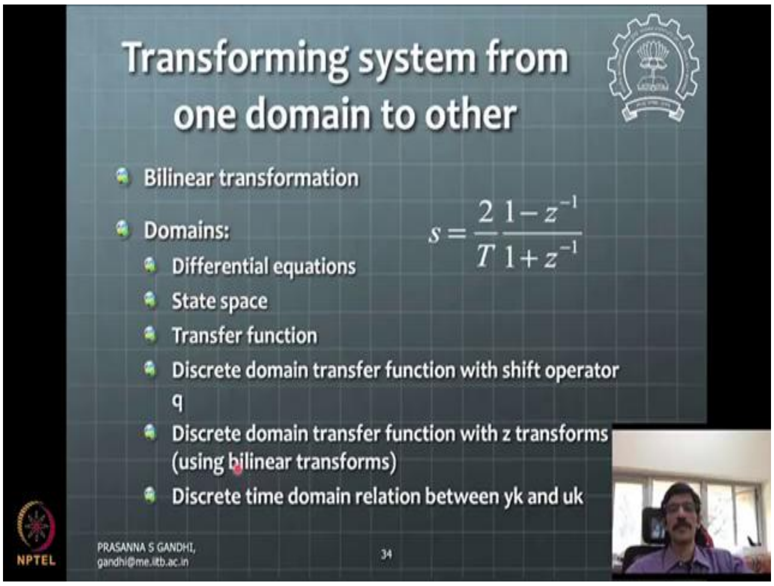
And we from here we will just move on to some filters and its implementation so this is our transfer function.

(Refer Slide Time: 18:58)



And then we will skip this part of the Z transforms and we will go to the filters eventually. So, you can these are like you know mainly mathematical kind of definitions and it gives some kind of properties which are similar to q operator properties.

(Refer Slide Time: 18:59)



But, not exactly the same. So, that is what we will see. Now, we get to the some kind of a fundamentals of a filter. So, the system to transform from one domain to other domain. So, now we are using this Z as a similar to like you know q operator we use this something called Bilinear transformation.

Where it pops up from is basically if you see the Eigen values of so you know this is like coming from how the poles get mapped. So, the bilinear transformation, so we will see what is this bilinear transformation will come to that in a while but one can see now that there are these different domains that you started off with this differential equation then you convert it into state space form or you can convert it into transfer function or you can have a discrete domain transfer function with the shift operator q or you can have discrete domain transformation with Z transforms or like you know using these Bilinear transforms.

So, see you can either use this discrete domain transition with the shift operator q or like you know using this bilinear transform these are the two ways that you know we can convert the S domain system to Z domain or to discrete domain. And so discrete domain discrete time relationship between yk and uk would be that is what we derived for with the shift operator.

So, there are different, different kinds of domains of operation that we can see here. Let us talk a little bit about a simplified way of doing the thing with this you know Bilinear transform way of doing things where this Bilinear transform comes from.

So, let us go back to our basic derivation of this system so the system in the discrete domain is $x(k+1) = \phi(T)x(k) + \Gamma u(k)$. Now, look this ϕ and Γ are some kind of a constant matrices. Now, if you see in very similar to what we have in the continuous domain this is like $\dot{x} = Ax + Bu$.

And we know that A matrix has is a very special matrix for the system because, the eigen values of A would be the poles of a system. So similarly, in this case eigen values of this ϕ matrix are going to the poles of a discrete domain system and this ϕ matrix is nothing but e^{AT}.

So, eigen values of ϕ or the poles of the system are going to get mapped to e to the power pole times T see if I say eigen values of ϕ will be equal to eigen value of e^{AT} and if I know eigen values of A then like you know it will be simply e to the power eigen value of A times T.

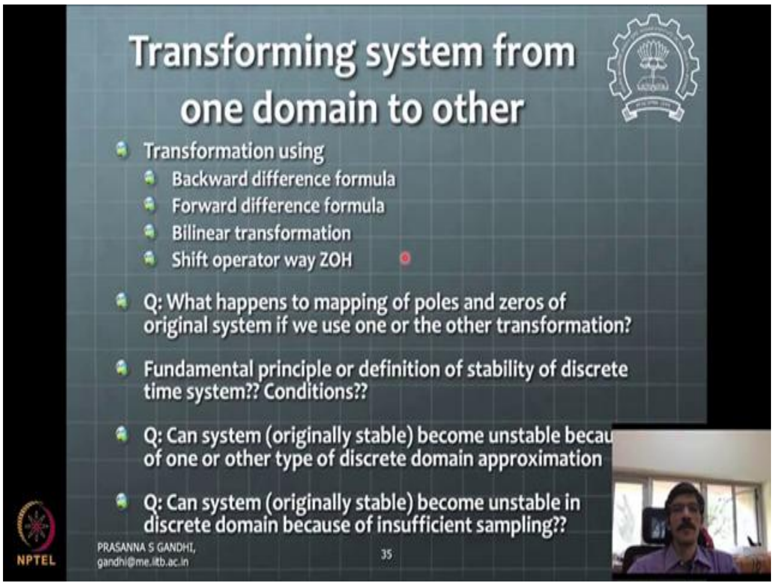
So, the poles get mapped so this is nothing but eigen value of A is nothing but a pole so e to the power pole times t is going to be the new pole for discrete domain system. The pole for a discrete domain system is nothing but e to the power pole for a continuous domain system times like sampling time T.

So, that is a basic nature here so with this you can see that e to the power so if I use s as a variable for the pole then e^{sT} is my pole for discrete domain system. So, this is like termed as C here or q here. So, $Z = e^{sT}$ and now if we simplify this in terms of its this exponential in terms of polynomial in the series expansion you get you know some first and second order terms based on that after some approximation of the first order terms one can now get this relationship which is termed as bilinear transforms.

$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$ this comes from that so I will leave it to you to derive that from that. So, this is genesis and relationship $Z = e^{sT}$. So, the approximation to this relationship is get brought into for polynomial first order terms collecting all the first order terms for the polynomial expansion we get this relationship.

So, I would leave it to you to derive this relation. So now, we will see how do you use this relation for then the implementation of filter or device design of filters that is what is our whole point of discussion. So, we are doing this for finally be able to implement filter in the in the discrete domain or in our microcontroller.

(Refer Slide Time: 25:43)



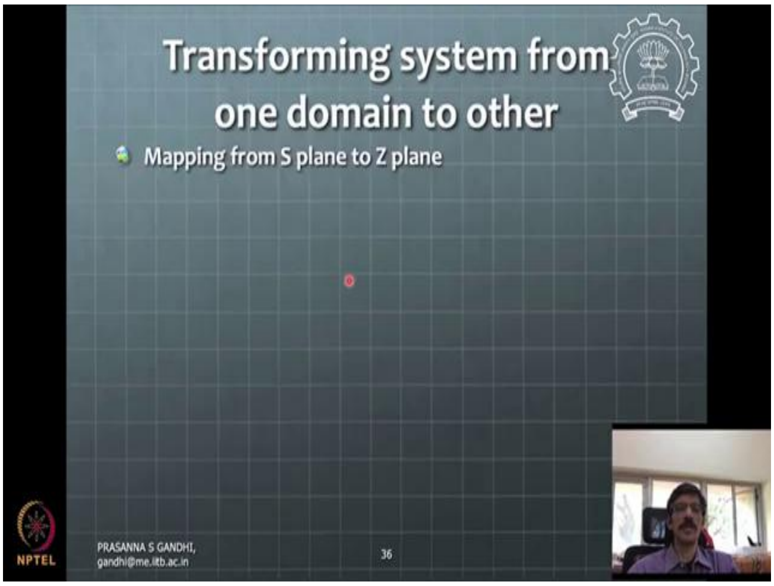
Now, so these are like you know different ways to transform the system into one form to another form so you can look at this say for example you want to transform system to this discrete domain one can use backward difference formula or forward difference formula or like you can take a transfer function of that system and so this backward and forward distance from the formula it will be applicable to the differential equation.

So, the differential equation operator d/dx will be approximated with this d/dt will be approximated with backward or forward difference formula and you can get the discrete version of the system. Or you can use the state space representation and do it by shift operator way.

Or you can use transfer function representation and s you replace that s by value here in terms of z and you get like know that z domain or this can be also q domain. So, both are very similar right now we will start using the terminology in a Z domain. So, for now we will not make much of a distinction between q and Z with our similar kind of operators that we will be interested.

Then, these are the questions about mapping of the poles and zeros how will this so these are kind of questions one can think about ponder over and answer we are not going to kind of get to them right now so we will see right now like the main focus is for the filter. So, this is so once we have original system in s domain any system in s domain we are able to transfer it into a discrete domain by using these different, different kinds of transformations. That is what is the main crux here to consider. Once we have this transformation available then how then it is easy to implement in the microcontroller.

(Refer Slide Time: 28:05)

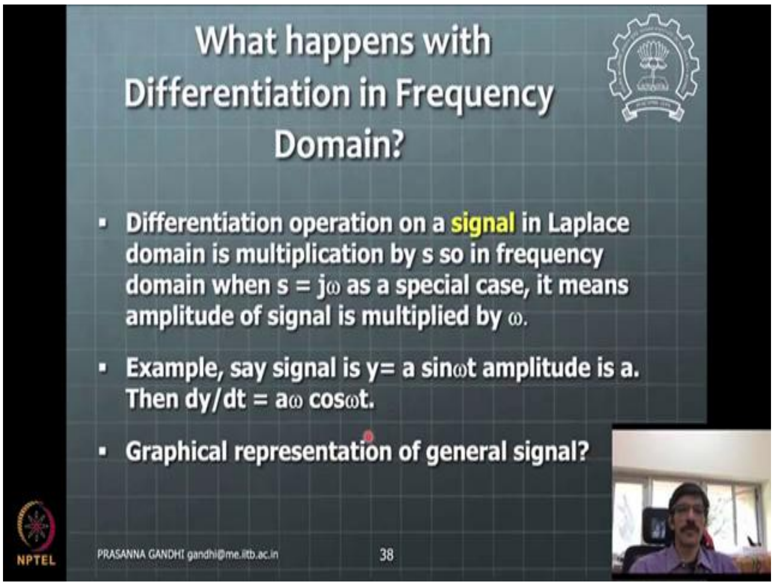


How do you see that in a minute. So, this is what we are talking about mapping of mapping from S plane to Z plane. So, we saw that in terms of poles it should it is like known $Z = e^{sT}$ that is what is a mapping from S to Z plane.

(Refer Slide Time: 28:28)

Now, filter we need in mechatronic system for frequency the noise in the system and to avoid the the performance degradation when we when this noise goes into the feedback. Then, so typically filters are used to remove the high frequency noise and the sources of noise you can have electromagnetic radiation coming from the fan, tube lights and other kind of elements which are around the mechatronic system or your motor itself will be generating a lot of electromagnetic signals which will act as a noise for your sensors.

(Refer Slide Time: 29:15)



Then, we have so we have some operations that are going on in the time domain. So, what will happen with those operations in the frequency domain is another thing that we need to look at. Why? Because, the filters have a role to play there also.

What role this filter have to play? where we do this kind of operation differentiation is where like say for example for your motor if you want to implement PD control what you do you take a value of your encoder and you use some kind of a difference formula or you are differentiating that numerically the encoder values and getting your speed.

So, this numerical differentiation has some bearing on the on the noise. So, it is very important to see what is happening with this differentiation operation in the frequency domain. So, if you see a differentiation operation like d/dt of a signal f (t) if it is done then Laplace transform of this d/dt of f will be simply like S into Laplace transform of the f.

So, Laplace domain gives you this kind of multiplication of the original Laplace transform by s when you differentiate a particular function. Now, if you replace this s by $j\omega$ so to kind of typically if you know if you remember you have s equal to $j\omega$ we use for getting a board plot of the system or getting a frequency domain response of a system.

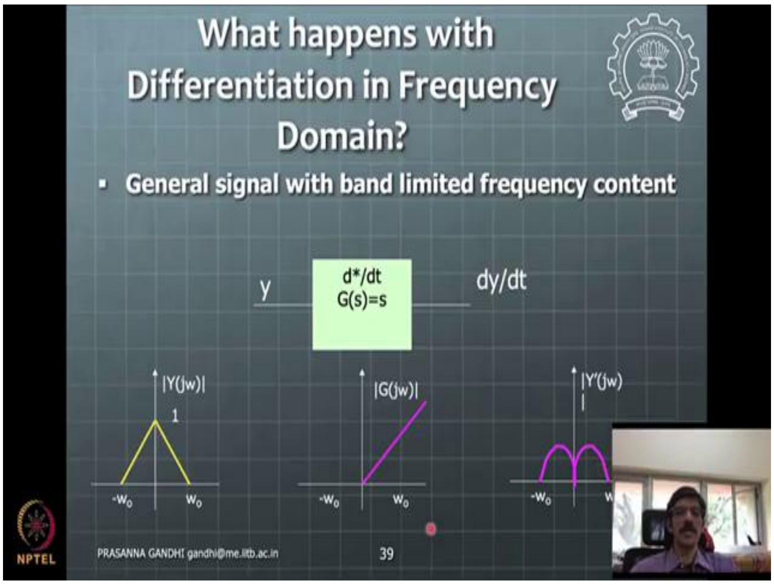
So, when you use s is equal to $j\omega$ then at that time like you know what happens is differentiation if you see in terms of when s is equal to $j\omega$ which is simply multiplication by $j\omega$ to the original $f(j\omega)$ so $f(j\omega)$ or capital $F(j\omega)$ is our nothing but our signal content.

So, if the signal content is $f(j\omega)$ then the signal content for the differentiated version of the signal would be $j\omega f(j\omega)$ magnitude of entire operating state. So, one can see very easily if your signal is $a\ sin\omega t$ if you take a differentiation of this with respect to time t then it becomes $a\omega\ cos\omega t$.

So, you can see that when you differentiate the signal the frequency content the frequency content would change and it would change in a way that you know it is whatever origins frequency content it is getting multiplied by ω. So, this is of course is in the time domain this is time domain signal you can see here we are just observing this time domain signal to see what is happening in the frequency domain.

It is not we actually if you want to kind of see this ideally we need to take a Laplace transform or Fourier transform of this signal to say that oh look the Fourier transform of the signal is integrating multiplied by ω to get a Fourier transform of this signal. So, when this is differentiated the Fourier transforms get multiplied by ω.

(Refer Slide Time: 33:43)



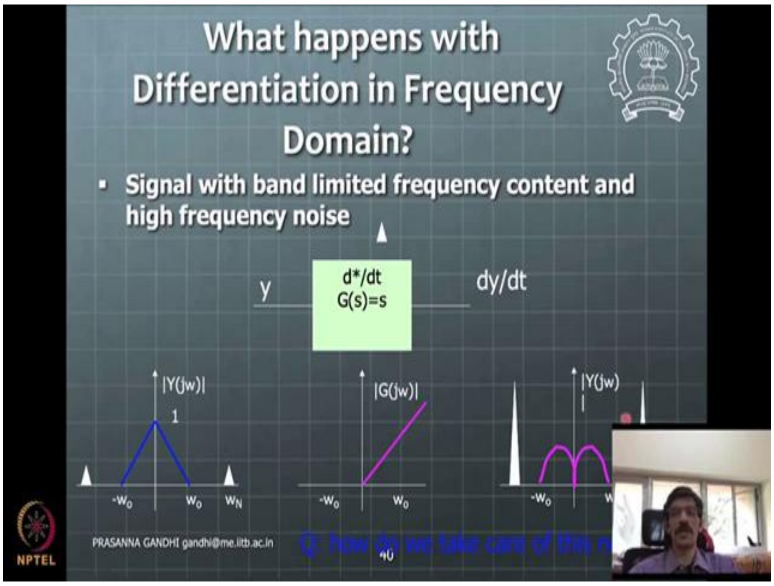
So, what it means is graphically one can say that if you have a signal here and you are using this differential operation here so you differentiate and get a signal dy/dt here and let us say this operation is some kind of a continuous domain ideal differentiation that is happening. We are not really doing this approximation by using forward or backward difference formula or anything like that.

I am just saying this is like a full you know differentiation then this differentiation operation would be this G(s) = s here so this is like a transform that is happening y(s) coming here getting multiplied by s and then we are getting s times y(s) as that Laplace domain function whose Laplace inverse will be dy/dt.

So, in the Laplace domain or in the frequency domain this is simply getting multiplied by ω. This ω plotted on the ω axis will be like a 45 degree line here. So, this is what will happen to this signal now is when it passes through this filter is this can be seen here that this signal will become 0 here it is return value 1 it was there if that value when multiplied by this signal will become 0 and then at ω_0 there will be the signal this signal has 0 value and this has some final value but that will become 0 and then this is some kind of a response that we will get.

So, this is a differentiated version of a signal. Now, so this is what the differentiation operation does to the signal.

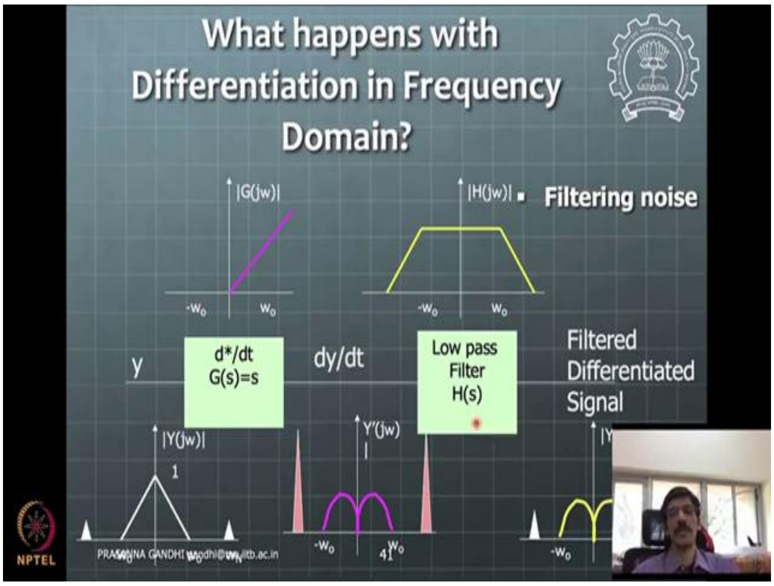
(Refer Slide Time: 35:35)



So, one can see that if the audio signal has some noise content now it has some typically the signals will have a noise at a high frequency and when you pass it now the differentiation kind of operation these are even if the ideal differentiation you will have you can find that now this noise is say ω_n frequency which is further away from ω_0 typically higher frequency noise.

The higher the frequency that much is a the amplification for the noise that you have to get. So, the noise in the signal will get amplified there that is what is a is a main effect of operation of differentiation of the signal. And because of this we need now filter to get rid of these extra peaks otherwise your feedback differentiated feedback is going to be noisy. In the PD control especially this is important because then the derivative part will will get noisy and you will not be able to use very high gains in the derivative expression.

(Refer Slide Time: 36:46)



So, this is how like then you will start off if you do this differentiation operation and then you use a low pass filter to filter out this noise and you get like the filter differentiated here. So, this is how like one can see what is happening in the frequency domain and we use the filters appropriately to do the job that like we want to reduce the noise in the differentiation operation.

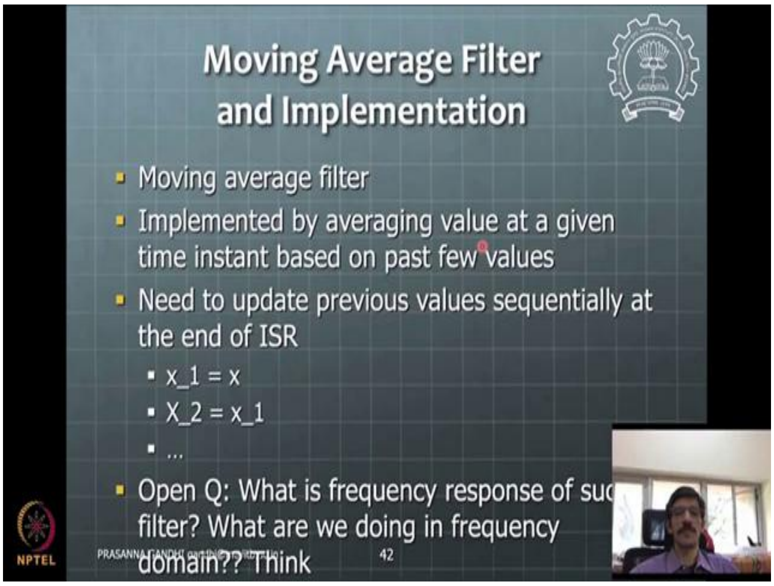
So, sources of noise are not only like you know whatever ambient sources but your mathematical operations also can induce like you know enhance the noise that is already there in the in the signal. So, we have not yet talked about like see this is the ideal differentiation but if you have a forward difference a backward difference there is some kind of a different kind of a form that will happen here which you are not getting bothered about for now.

So, we just leave with this like know idea that we are doing like you know continuous domain differentiation and we work with that kind of continuous domain system and then say that this is more or less similar to what will happen with the sampling time is fairly appropriate.

So, the filter if you see here is some kind of a transfer function which has some kind of a frequency response or bode plot and then the signal passes through that and it produces the output that is a very simple concept of filter. So, you see this is like a first order filter here so it has some kind of a cutoff frequency beyond which like you know it has this tapering of some slope will happen here.

You can have second order filter or like larger order filters will have like know this is a sharper and sharper cutoff will happen here instead of this going in a in a gradual slope kind of fashion it will go down fast that will be a high order of record.

(Refer Slide Time: 39:14)

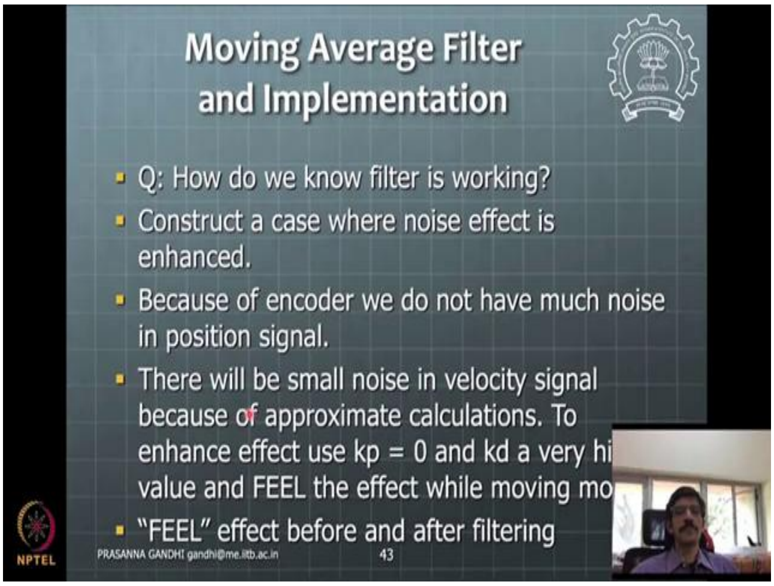


So, when we have to implement this filter in the so we will see how this filter can be implemented so this is one of the ways of implementing printer is like this moving average kind of a filter. So, moving average filter will have a formula so what you do is like at a given point you will consider a previous two or three or more samples and take an average of all those samples and keep it at that value.

And you need to critically update the previous values of the sample and we have seen that that update can be done by using this kind of a way. So, the one previous value will be current value then secondary value will be previous current value like that we can go ahead and do the updates.

These are the updates that we have seen when we implement derivative control we need to do this kind of updation of the previous value so that we are ready for the next sampling instance variation derivative to be completed in the next couple of instances.

(Refer Slide Time: 40:22)



So, we will not worry about these sequences of such a filter but like you know one of the ways of implementing filter is like moving average kind of filter. And if given a transfer function then one can use this one can use this suppose this transfer function is given $\frac{1}{\tau s+1}$ kind of a transfer function then one can use s is equal to this bilinear transform formula and get that as a discrete domain transfer function.

And using a discrete domain transfer function you can transfer it into the discrete domain time function time equations. So, how do we do that that will be done by using this shift operator. So, we will stop for here for now and I think this some of these concepts although you may find that there are kind of little bit of a tough to grasp immediately you will need to ponder where you may need to go back and forth over the slides and listen carefully then you will be able to grasp this concept.

I understand that these are like too much of a too many kind of different concepts are packed here together but they are like done in the in a manner to kind of see that you are able to kind of develop yourself or implement filters yourself. But, these are the ideas that you will typically need for the defined implementation of a given mechatronic system.

Because, many times you will be bothered by the noise in your sensors and when you further do this differentiation operation your noise is futher increase and then you usually get further what to do in such kind of a scenario how to kind of handle that time this concepts are going to be very very useful.