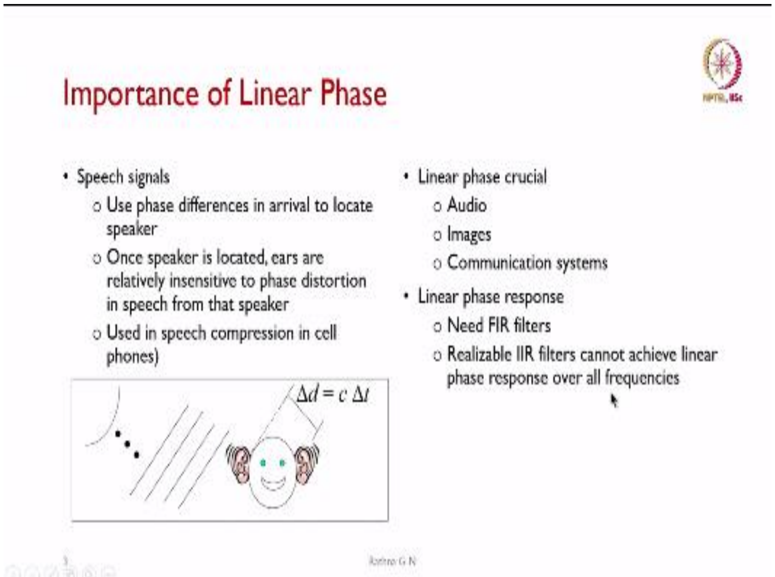
Real – Time Digital Signal Processing
Prof. Rathna G N
Department of Electrical Engineering
Indian Institute of Science - Bengaluru

Lecture – 11
IIR Filters - 1

Welcome back once again to real time digital signal processing course. So, today we will discussed about IIR filters. So, as a recap, in the last 2 classes we discussed about FIR filters, their linear phase and how we can represent them, how we can design using the window techniques?

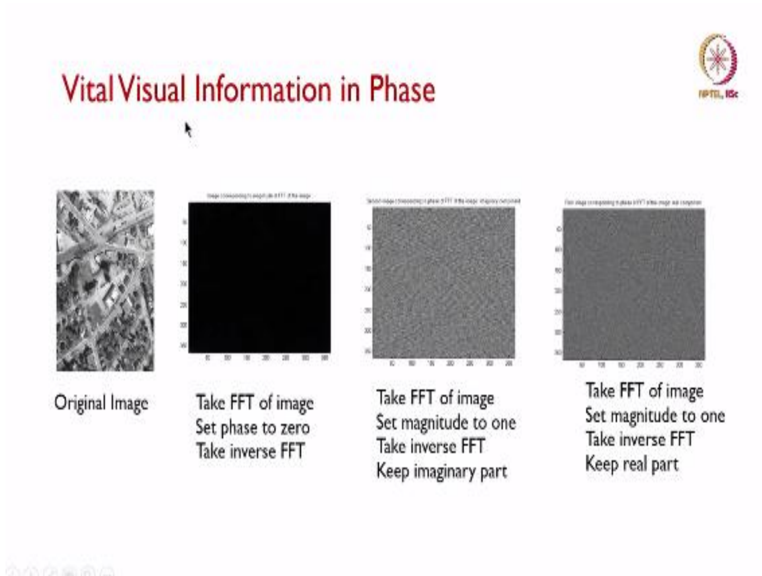
(Refer Slide Time: 00:49)



So, today we will see little bit on recap of FIR filter that is we said it is a linear phase filter. So, the importance of it as you can see with an example in the speech signal, we use phase differences in arrival to locate the speaker. So, in this case it is we may not need the phase part of it if there is any delay in the thing. So, you locate them that is we call it $\Delta d = c\,\Delta t$. So, this is how the speaker thing and then our ear automatically adjust to whatever there is a delay in the face of it.

That is what it is shown here one speaker is located ears are relatively insensitive to phase distortion in speech from that speaker. So, this is used in speech compression in cell phones, where linear phase is crucial we will see in the audio signals in images and in communication systems. So, in these cases, we need linear phase response filters, FIR filters and then a realizable IIR filters cannot achieve this linear phase response over all frequencies.

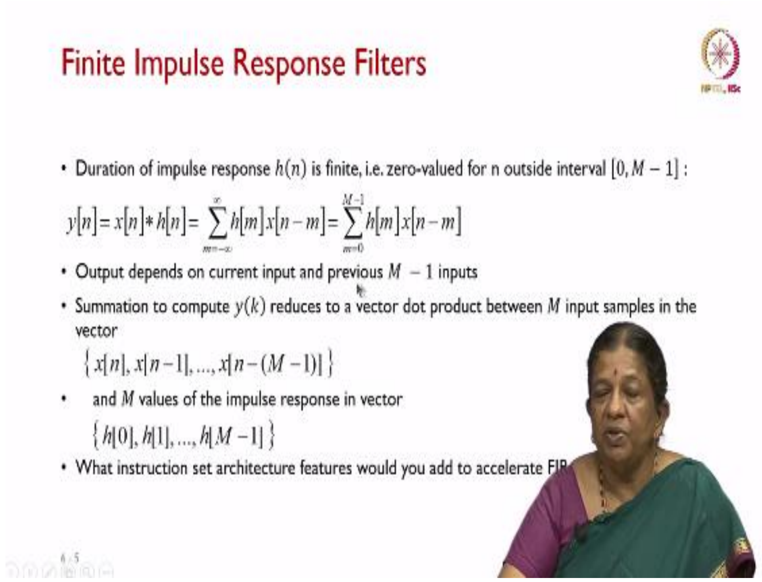
(Refer Slide Time: 02:08)



So, coming to one more example of where the phase is important that is vital visual information in phase is shown with the MATLAB original image here. So, when you take the FFT of the image and set phase to 0 and take inverse FFT as you will be seeing only you will be seeing the blank that is a black in this case, no picture whatever was in the originally seen, whereas if you take the FFT of the image and set magnitude to 1 take inverse FFT keep imaginary part.

So, this is how the imaginary part looks like when you take the FFT and this is with respect to real part if you take FFT, so, combine will give us the original image.

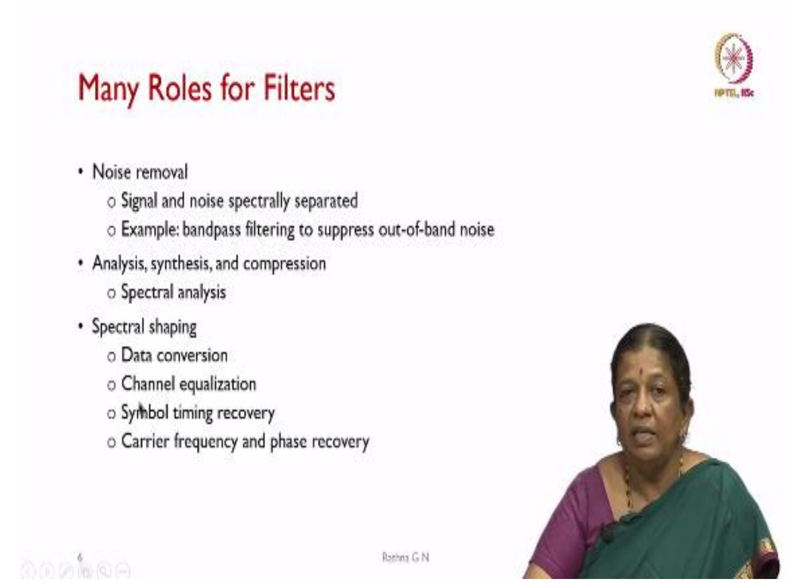
(Refer Slide Time: 03:02)



So, coming to representation of our finite impulse response filters, we know that duration of impulse response $h(n)$ is in this case is finite. So, this is 0 value for n outside interval $[0, M-1]$. So, we say that $y(n)$ is nothing but $x(n) * h(n)$, which is represented in the sigma notation in this way and then this is -∞ to ∞. So, in our case FIR filter is going to be because we will be designing order of the filter is m. So, the summation will be between $[0, M-1]$.

So, in this case output depends on current input and previous $M-1$ inputs and summation to compute our $y(k)$ reduces to a vector dot product between input M input samples in the vector domain as it is seen here $x(n)$ is represented in vector and then our impulse response is represented in vector then it is going to be a vector multiplication, what we will call it.

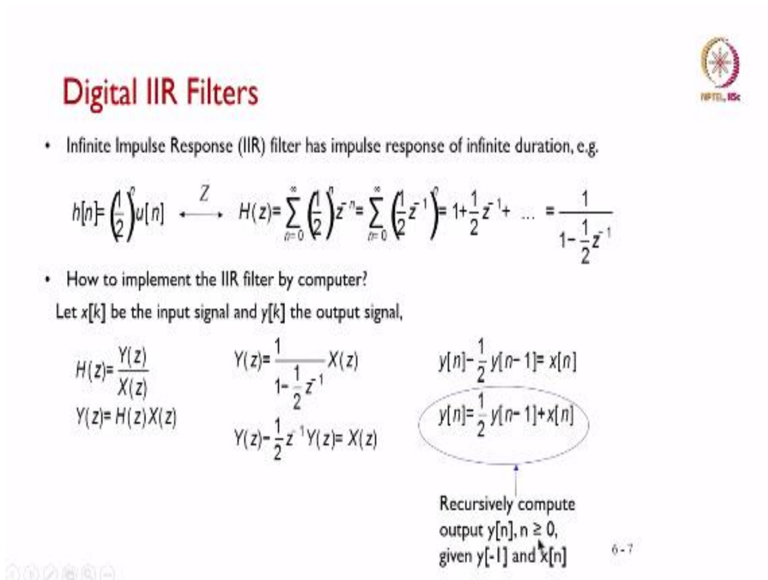
(Refer Slide Time: 04:17)



So, we know that filters play many roles we have seen the thing just to give one more brief look at the thing is noise removal, signal and noise spectrally separated basically in that case, example is we can use bandpass filtering to suppress out of band noise and in the case of analysis, synthesis and compression. So, for we use a spectral analysis basically to see how much of data what we need it and which are the ones we can do the compression in the frequency domain.

Other method is in the spectral shaping so, that is in basically for data conversion we use the filters and for the channel equalization, we know that in communication, input channel has to be reconstructed. So, we take inverse filter basically there and then do the channel equalization, whatever noise coming out of the channel. And then in the symbol timing recovery also will be using the filters and in carrier frequency and phase recovery we need filters.

(Refer Slide Time: 05:31)



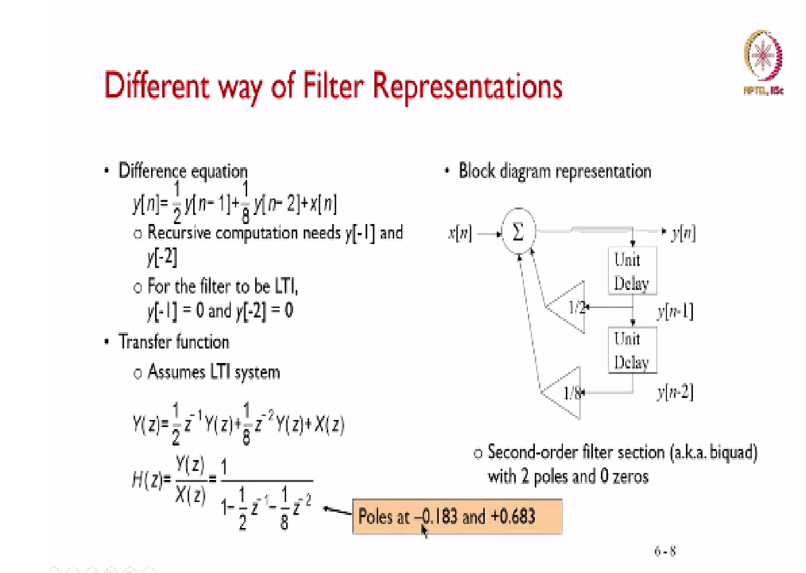
Coming to next is infinite impulse response IIR filter. So, we see that impulse response of infinite duration, that is what we call it. So, what is that mean we will be seeing in a while. So, if I give the impulse response, $h(n) = \left(\frac{1}{2}\right)^n u[n]$ that is the step function. So, the same thing in our frequency domain or in the zee domain, what we represent is $H(z) = \sum_{n=0}^{\infty} \left(\frac{1}{2}\right)^n z^{-n}$. So, which will be representing it as change it to 0 to ∞.

So, expand the summation, which becomes $1 + \frac{1}{2}z^{-1} + \cdots$. So, we will be getting $\frac{1}{1-\frac{1}{2}z^{-1}}$. So, how to implement this IIR filter by computer what we will look at it? So, we say let $x(k)$ be the output signal and then $y(k)$ be the output signal, then what happens is the zee domain representation is going to be $y(z)$ for our output and then $x(z)$, then the impulse response $H(z)$ in the zee domain is represented as $\frac{Y(z)}{X(z)}$.

So or if we want $Y(z)$, then we know that impulse response into our input $H(z)X(z)$, then what we have taken this as our impulse response, $Y(z)$ will be becoming $\frac{1}{1-\frac{1}{2}z^{-1}}X(z)$, then, if we simplify the thing, it becomes $\frac{1}{2}z^{-1}Y(z)$. That is $Y(z) - \frac{1}{2}z^{-1}Y(z) = X(z)$. So, what happens when we take the inverse z transform what happens to our $y(n)$ which is equal to $\frac{1}{2}y(n-1)$ that is a previous sample, which is equal to $x(n)$.

So, if we transform this $\frac{1}{2}y(n-1)$ to the other side, we will be seeing that $y(n)$ will be equal to $\frac{1}{2}y(n-1)+x(n)$. So, $x(n)$ is the current sample and $y(n)$ is $n-1$ is the previous output sample, which is used for calculating the current output sample. So, what we see is recursively compute output y of n for n greater than or equal to 0, given $y(-1)$ and then $x(n)$.

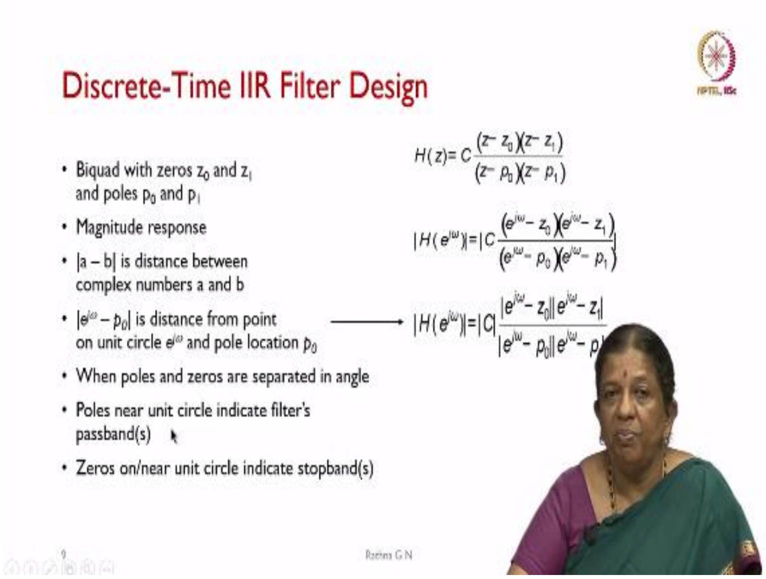
(Refer Slide Time: 08:30)



So, how we can represent this filter in a different way, so we will be seeing the difference equation first. So, $y(n)$ is given us $\frac{1}{2}y(n-1)+\frac{1}{8}y(n-2)+x(n)$. So that is we are going to do the recursive computation needs. What are the input values, what we need is $y-1$ and then $y-2$ for the filter to be linear time invariant. So, we assume that $y(-1)=0$ and $y(-2)=0$. And the block diagram to represent this difference equation is shown in this figure, $x(n)$ is the input, and $y(n)$ is the output.

So, with the delay, we will be generating $y(n-1)$ which is going to be multiplied by half. And then other unit delay will give us $y(n-2)$. So, which is multiplied by weight vector $\frac{1}{8}$. So, all the 3 are getting summed up and then we will be taking $y(n)$ as the output. So, the transfer function what we see is it assumes a linear time invariant system. So, $y(z)$ is represented as $\frac{1}{2}z^{-1}Y(z) + \frac{1}{2}z^{-2}Y(z) + X(z)$ and impulse invariance response of this is given by $\frac{Y(z)}{X(z)}$ which is nothing but $\frac{1}{1-\frac{1}{2}z^{-1}-\frac{1}{8}z^{-2}}$. So, we say that poles are at -0.183 and $+0.683$.

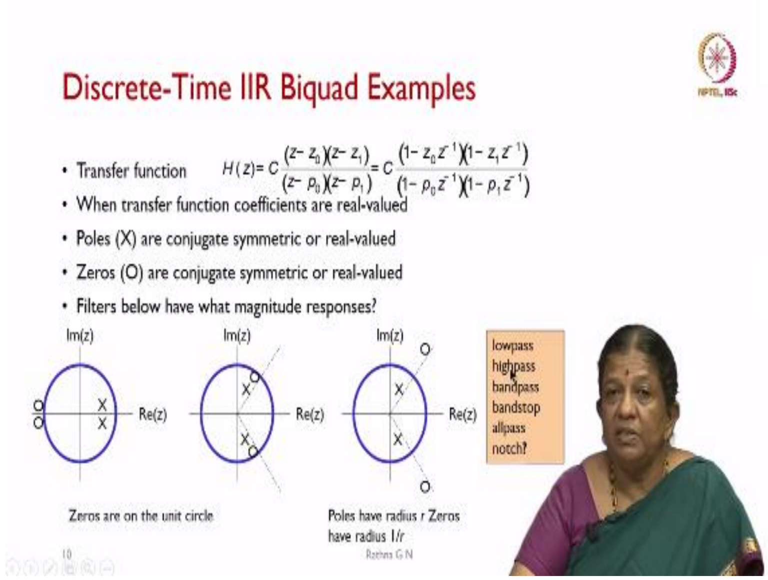
(Refer Slide Time: 10:19)



So, how we are going to design this, we say it is designed with respect to Biquad zeros z_0 and z_1 and poles will be represented as p_0 and p_1. So, always we use the Biquad section. So, in that case how it is going to represent it $H(z)$ is given by C is the weight factor what we have it, $\frac{(z-z_0)(z-z_1)}{(z-p_0)(z-p_1)}$. So, how we represent the magnitude response, we will be taking in the frequency domain we substitute $z = e^{j\omega}$ then take the magnitude response. So, our $z = e^{j\omega}$ has to be put in here also.

And this represents our magnitude response and then our, we say that magnitude of $a - b$ is distance between our complex numbers a and b and $e^{jw} - p_0$ magnitude is a distance from our point on unit circle e^{jw} and pole location at p_0. So, that is how we represent the thing when poles and 0s are separated in angle and poles near unit circle indicate filters passbands and zeros on near unit circle indicate the stopband. So, we can see that how the poles and zeros have to be placed to achieve our passband response and then this stopband attenuation how we can arrive at.

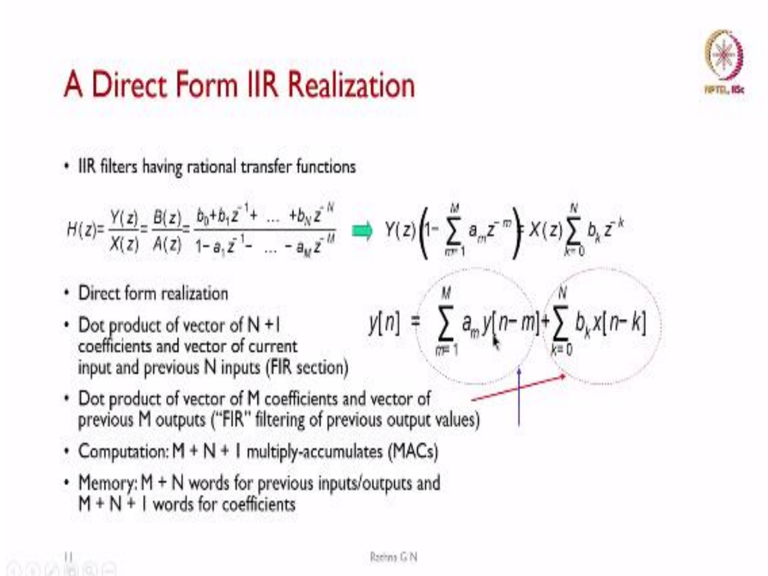
(Refer Slide Time: 12:09)



So, coming to Biquad example continuing with the thing that transfer function is given by this equation and when transfer function coefficient are real valued, then poles we say x are conjugate symmetric or real valued and then zeros we represent with 0 or conjugate symmetric or real valued. So, filters below have what magnitude response one has to answer this. So, what you see is these are the 2 complex conjugate poles here and corresponding 0s are outside the unit circle, this is our real axis and this is our imaginary axis.

So, as we say that zeros are on the unit circle basically, and here you will be seeing that poles and zeros are inside the unit circle and the other one you will be seeing it poles are inside whereas zeros are outside the unit circle. So, poles have radius r zeros have radius 1 by r that is what what we represented. So, we have what kind of filters we can design using IIR method or FIR method, lowpass filter, highpass, bandpass, bandstop or allpass or even notch filter what we can design the thing.

(Refer Slide Time: 13:40)

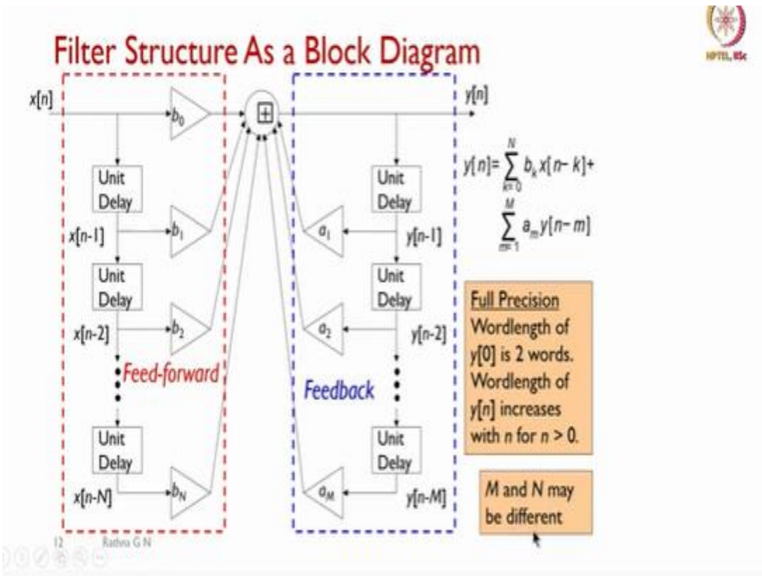


So, first we will see the direct form IIR realization how it is going to be represented. So, rational transfer function what we call it, so, our impulse response $H(z)$ is given by $\frac{Y(z)}{X(z)}$. So, we say that $B(z)$ represents our zeros and $A(z)$ represents our poles. So, which is given by $b_0 + b_1z^{-1}$, b_Nz^{-N}so, we have N zeros and we say M poles. So, 1-$a_1z^{-1} \pm \cdots a_Mz^{-M}$.

So, which implies that my $Y(z)$ is represented as $(1 - \sum_{m=1}^{M} a_m z^{-m})$ so, this is my represents my poles basically, which is equal to $X(z)$ into this we will be varying $\sum_{k=0}^{N} b_k z^{-k}$ we say direct form realization that is dot product of a vector of $n + 1$ what we are assuming it coefficients and vector of current input and previous inputs, what we call it as a FIR section for our zeros and dot product of a vector of M coefficients and vector of previous M outputs.

So that is FIR filtering of previous output values what we will be taking it here as it is represented, and then computation of $M + N + 1$, multiply accumulate what we need it as we can see the thing. And memory is also going to be $M + N$ words for previous inputs and outputs. And then $M + N + 1$ words for the coefficients as it is represented here. This is our previous output. And this is our input represented with the coefficients b_k.

(Refer Slide Time: 16:05)

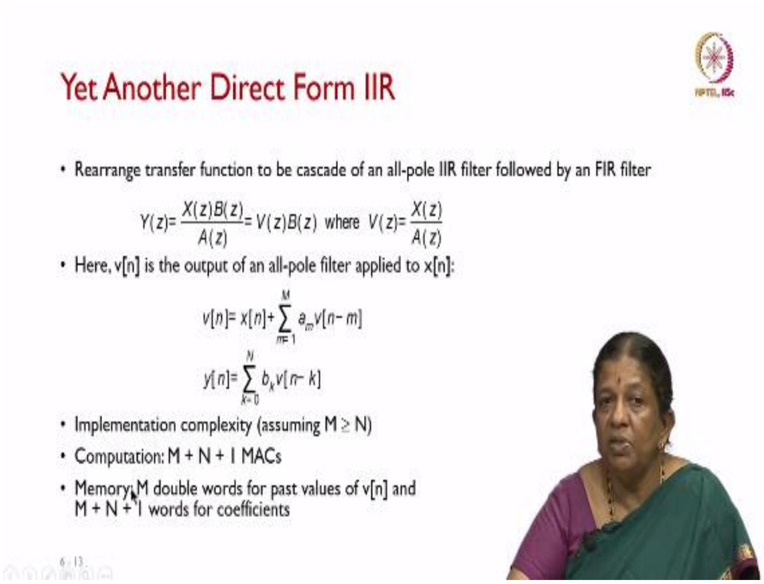


So, when we represent this equation in this form, what happens to our structure, we will see, $x(n)$ is the input. And we see that b_0, b_1, b_N coefficients are in the feed forward, so we have the unit delay, and this is $x(n-1)$ to $x(n-N)$, which are going to be summed up in this unit, and then output is going to be our $y(n)$. So, what is the feedback here, so, we will be delaying with one unit delay and coefficient is going to be multiplied with a_1. So, we will be $a_1 y(n-1)$ so, which goes up to $y(n-M)a_M$.

So, all this is we call it as the feed forward path and this is the feed backward path. So, completely added here, if there is as we have seen in the earlier case it is $-a_1 y(n-1)$. So, some of the coefficients, as we will see, they may add and then get subtracted and then they may nullify, and we may not have the overflow, may or may not we can tell that y of n maybe overflowing or not overflowing in these cases depends on the coefficients what we are using it see we call it is when we want the full precision. So, what we need is word length of $y(0)$ is going to be 2 words. So, as we have seen the thing multiplication and then addition, so, it will be going up to 2 words. So, what happens to word length of $y[n]$ if we start putting y of with 2 words, we will be seeing the output next time is going to be 4 words or it goes on increasing to avoid this, what we have to do is we have to come back our $y[n]$ representation with whatever n bit representation so, that we are not going to have overflow in our representation of $y[n]$.

So, and then M and N may be different, we do not know what kind of order of the filter we are going to design. So, they may be the same or they can have different values for them.

(Refer Slide Time: 18:34)

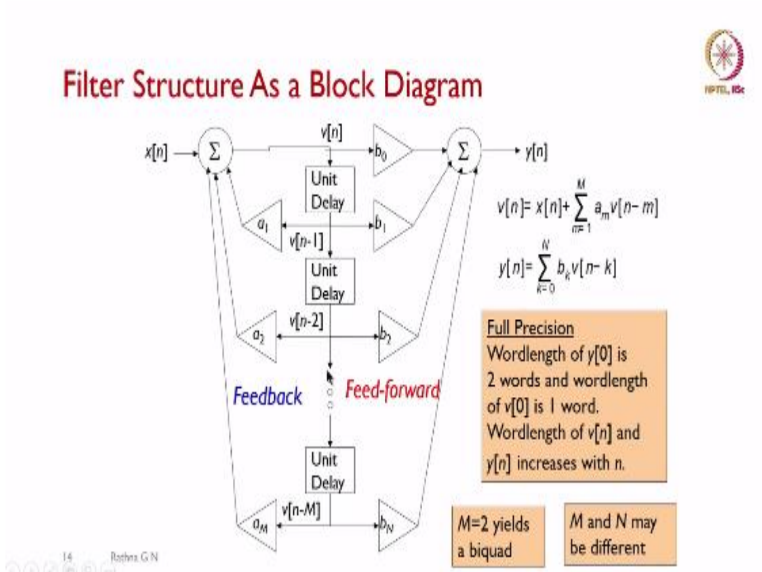


Then, how we are going to represent again the direct form of IIR filter as it is shown here. So, we will be rearranged transfer function to be cascade of an all pole IIR filter followed by an FIR filter. So, that is $Y(z) = \frac{X(z)B(z)}{A(z)}$. So, we represent this as $V(z) = \frac{X(z)}{A(z)}$

as $V(z)B(z)$ where that is what we have z is given by this equation. And then we see here $v(n)$ is the output of an all pole filter applied to $x(n)$ that is what we call it.

So, $v(n) = x(n) + \sum_{m=1}^{M} a_m v(n-m)$ basically m will be varying 1 to M. And $y(n)$ you will be seeing that it is represented with our 0s multiplied by $v(n-k)$. So, the implementation complexity it we can assume $M \geq N$, the number of computation what we need is $M + N + 1$ MACs in this case, and memory is going to be M double words for past values $v(n)$ and $M + N + 1$ words for the coefficients

(Refer Slide Time: 20:02)

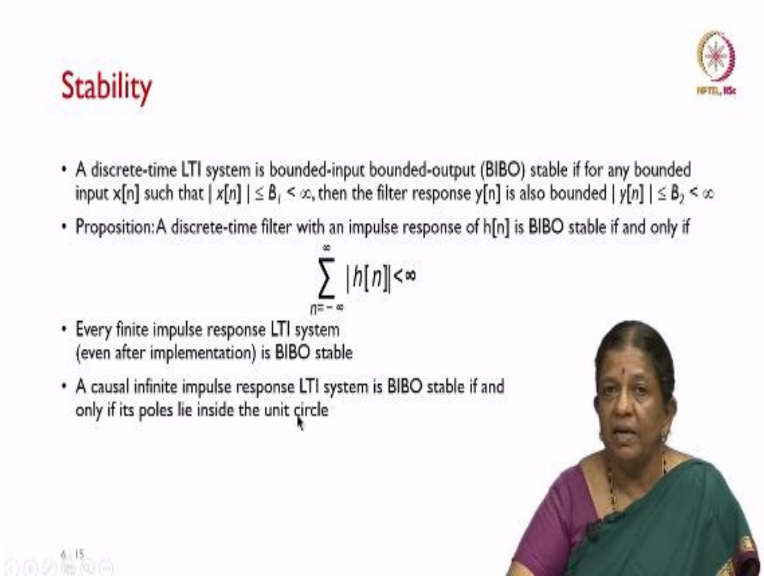


How this is going to represented as you can see here, this is our $v(n)$ point and we have summation here and summation here. So, x of n is going to be multiplied with all IIR filter feedback basically, that is, this is the feedback structure. So, $v(n-1)$, $v(n-M)$, which is going to be summed up. So, which is going to be given as, as you will be seeing that that is our $v(n)$, which is getting multiplied, we will be seeing that $b_k v(n-k)$ with that and then summed up here with all feed forward 0s and $y(n)$ will be output.

So, as you notice from single summation, what we have come down to double summation. So, the critical path, what we call it is our over flow or under flow, because here all of them are negative we may underflow in this case, and all of them are positive, we may over flow also. So, these are the 2 critical notes what we call it, one has to take care that summation is not going to over flow or under flow these 2 points from one single critical node, we have Biquad into 2 critical notes.

But as you can see that the delay in itself come down by 2 compared to the previous direct form structure representation. So, that is what the advantage.

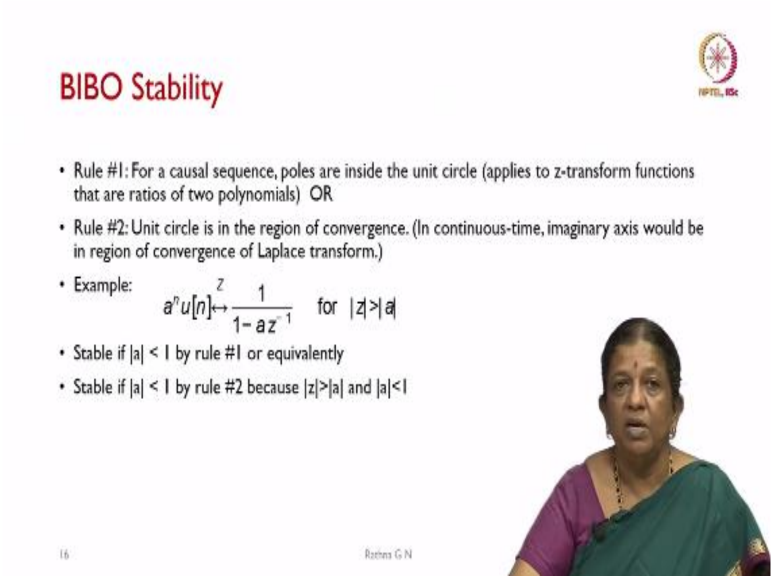
(Refer Slide Time: 21:44)



Now comes with the stability because we are more worried with respect to stability. So, we will be seeing that our linear time invariant system is what we say is bounded input bounded output BIBO stable what we call it. So, if our any bounded input $x(n)$, such that $x(n)$ is within the bounds what we call it between $x(n)$is less than or equal to our some bound value, which is less than ∞, then the filter response $y(n)$ is also bounded what we claim it that is whatever B_2 it should be less than ∞.

So, coming with the FIR filter, that is $h(n)$ what we have it so, we said $x(n)$ is bounded already, then when the FIR filter is going to be stable, that is BIBO stable, if and only if what we claim is magnitude of our impulse response $h(n)$ is less than infinity for n is varying from -∞ to ∞. So, we say every finite impulse response of LTI system, even after implementation is going to be bounded input bounded output stable. So, we say causal infinite impulse response LTI system is BIBO stable if and only if its poles lie inside the unit circle that is what the meaning of it.

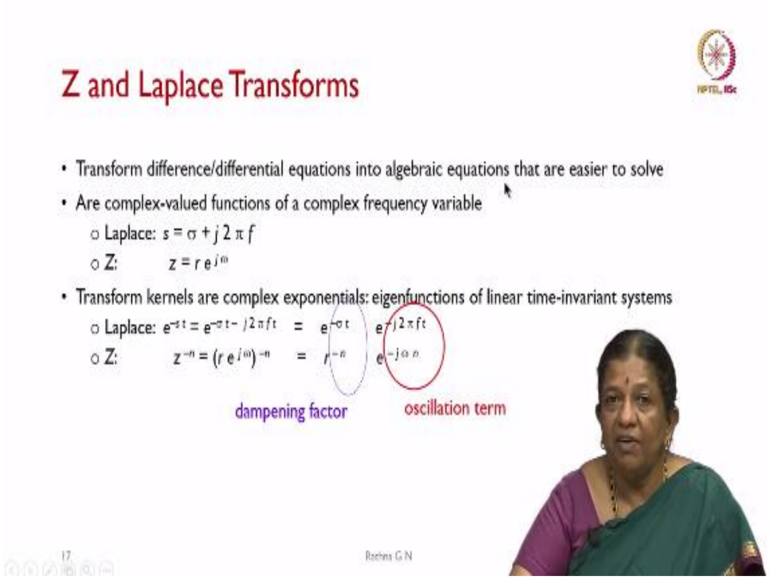
(Refer Slide Time: 23:22)



So, continuing with the thing, so, what it says is rule one is for a causal sequence poles are inside the unit circle applies to z transform functions that are ratios of 2 polynomials or the rule two is unit circle is in the region of convergence in continuous time imaginary axis would be in region of convergence of Laplace transform what we claim. As an example, a^n into units step function when we take the z transform of it, which is nothing but $\frac{1}{1-az^{-1}}$ for all |z| > |a|.

So, stable if $|a| < 1$ by a rule 1 applying it or equivalently stable if magnitude of $|a| < 1$ by rule 2 also because magnitude of z is greater than $|a|$ and $|a| < 1$, so, that we will be achieving the stability.

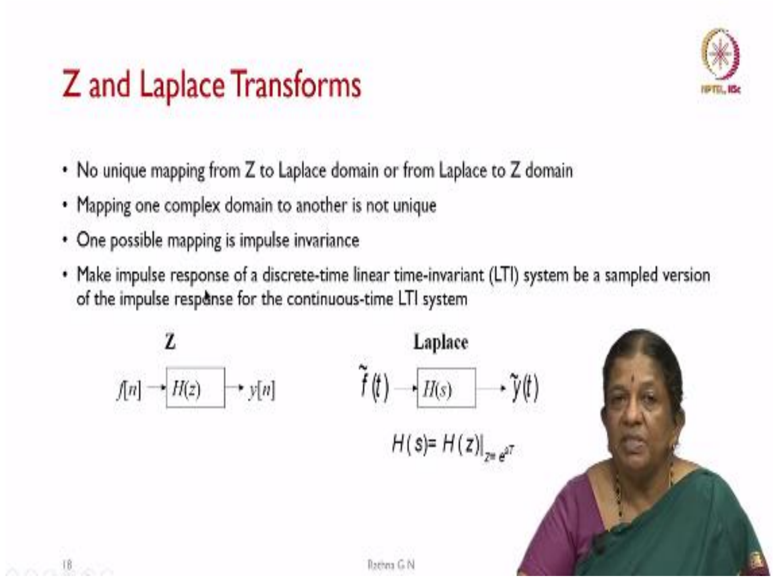
(Refer Slide Time: 24:34)



So, we will see how we can represent Z and then Laplace transform what is the relationship so, that is transformed difference or differential equations into all algebraic equations that are easier to solve. So, our complex valued functions have a complex frequency variable, what we call it as Laplace in transforms, what we represent $s = \sigma + j\,2\,\pi\,f$, whereas in the Z transform, we will have $z = e^{jw}$. So, the transform kernels are complex exponentials eigenfunctions of linear time invariant systems what we call it.

So, we have Laplace $= e^{-st} = e^{-\sigma t - j2\pi f t\ t}$. So, which is nothing but $e^{-\sigma t}$ and then these are the exponential function to exponential function what we are representing. So, in the Zee domain what we have $z^{-n} = \left(re^{j\omega}\right)^{-n}$ that is z we are replacing with $re^{j\omega}$. So, which is nothing but r^{-n} and then $e^{-j\omega n}$. So, we constitute to that e power $-\sigma t$ and then $r - n$ as dampening factor whereas, the other exponential factors $e^{-j2\pi f t}$ and $e^{-j\omega n}$ are the oscillation term in our representation.

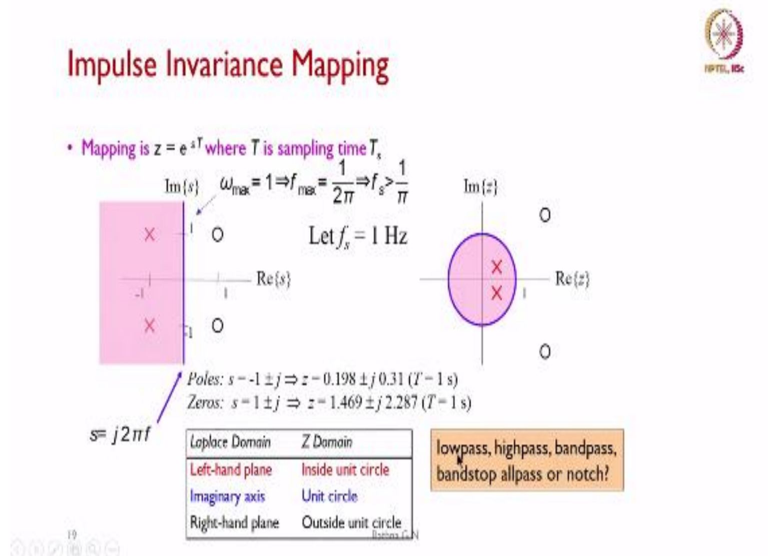
(Refer Slide Time: 26:20)



So, no unique mapping from Z to Laplace domain or from Laplace to Z domain. So, we can have mapping one complex domain to another is not going to be unique and one possible mapping is impulse invariance. So, make impulse response of a discrete time linear time invariant system be a sample version of the impulse response of the continuous time LTI system. So, how what is that I have a function $f(n)$. So, in the Zee domain I represent, pass it through with $H(z)$ that is my impulse response, and what I will get output is $y(n)$.

Whereas, in the case of a Laplace transform. So, will be a continuous time signal is represented with $\widetilde{f}\ (t)$, which is transformed using $H(s)$. So, I will be getting the $y(t)$. So, how do we represent in that case, $H(s)$ is given as $H(z)|_{z=e^{sT}}$ that is, we call it as impulse invariance mapping.

(Refer Slide Time: 27:33)

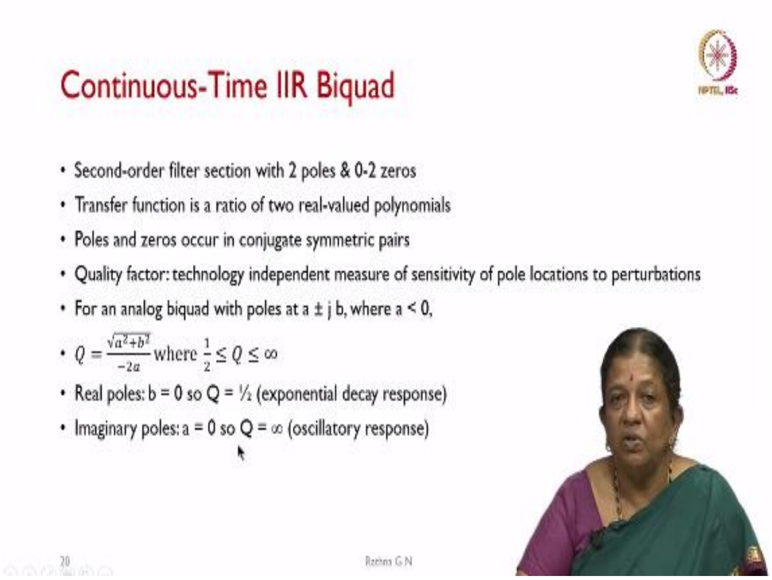


So, coming to the mapping how it is going to happen in the impulse invariance, what it is shown here, mapping is $z = e^{sT}$ where T sampling time what we call it as T_s basically. So, we will be seeing the axis our real axis and then imaginary axis, the poles on the left hand side of our what we call it as Laplace transform, which is going to be mapped inside the unit circle and our imaginary axis which is shown in blue actually $s = j2\pi ft$ gets mapped on the unit circle which becomes 1.

And then the 0s on the right hand side of it can be mapped to outside our unit circle. So, this is how from Laplace transform to Zee transform, the transformation is going to happen. So, what we have is omega max is nothing but 1 in this case, which is f_{max} which is given by $1/2\pi f$ which is implied that f_s should be greater than $1/\pi$. So, we assume let $f_s = 1$ hertz what we are assuming it then what happens our poles $s = -1 \pm j$ in the, is getting mapped in the z domain as 0.198 + or - j 0.31 into T - 1 second.

So, 0s will be getting mapped as $s = -1 \pm j$ is getting mapped as in the z domain 1.469 + or - j 2.287 into T - 1 second. So, you will be seeing the Laplace domain, left hand plane inside the unit circle which we discussed and imaginary axis is the unit circle and right hand plane is going to be our outside unit circle. So, what we say is with this we will be having lowpass, highpass, bandpass, bandstop allpass or notch filter what it is going to be designed.

(Refer Slide Time: 29:55)



So, how we are going to represent our continuous time IIR Biquad section. So, second order filter section with 2 poles and 0 to 2 poles what we can have 0 to 2 zeros. That means to say I need not have to have any 0s in our second order filter. If m and n are different if they are equal, then we will be having 2 poles and then 2 zeros. And that is what we will be representing and in the transfer function is the ratio of 2 real valued polynomials what we consider and then poles and 0s occur in conjugate symmetric pairs.

And what we define the quality factor, it is a technology independent measure of sensitivity of our pole locations to perturbations. For an analogue biquad with a poles at $a \pm j\, b$, where $a < 0$, then the quality factor what $Q = \frac{\sqrt{a^2+b^2}}{-2a}$, where it is going to be half less than or equal to Q which is less than or equal to ∞. So, if we have the real polls, then b will be 0.

So, the quality factor will be half that is we call it as exponential decay response. And if we have imaginary poles, a will be 0. So, Q becomes ∞. So, we say we are going to have the oscillatory response. Thank you.