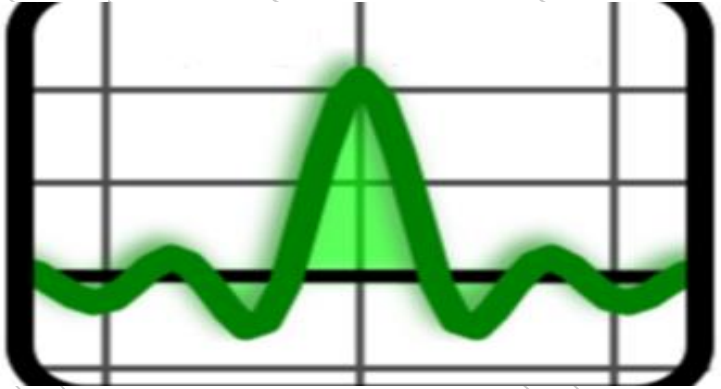


Signals & Systems For Computer Engineering

Prof.Dr. B.Berk ÜSTÜNDAĞ
Istanbul Technical University
Faculty of Computer Engineering and Informatics
bustundag@itu.edu.tr

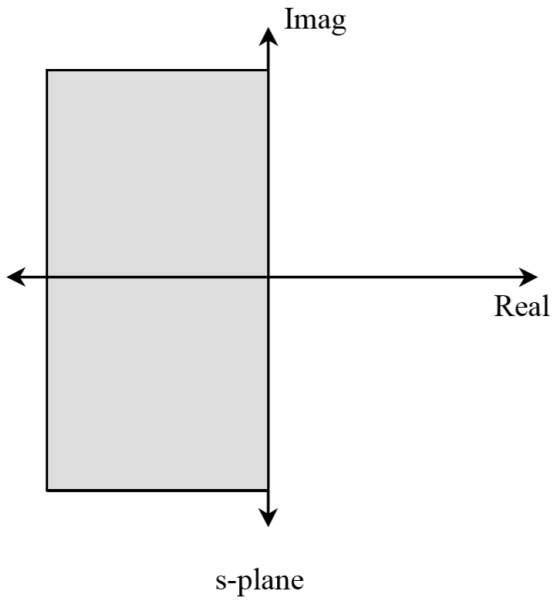
BLG354E / CRN: 21560

15th Week Lecture

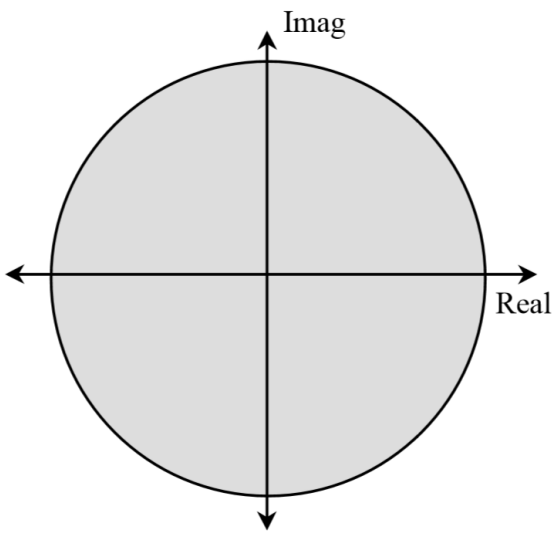
Evaluation of pole-zero configuration on impulse and the frequency response of the system

$$H(s) = H_0 \frac{(s - z_1)(s - z_2)(s - z_3) \cdots (s - z_m)}{(s - p_1)(s - p_2)(s - p_3) \cdots (s - p_n)}$$

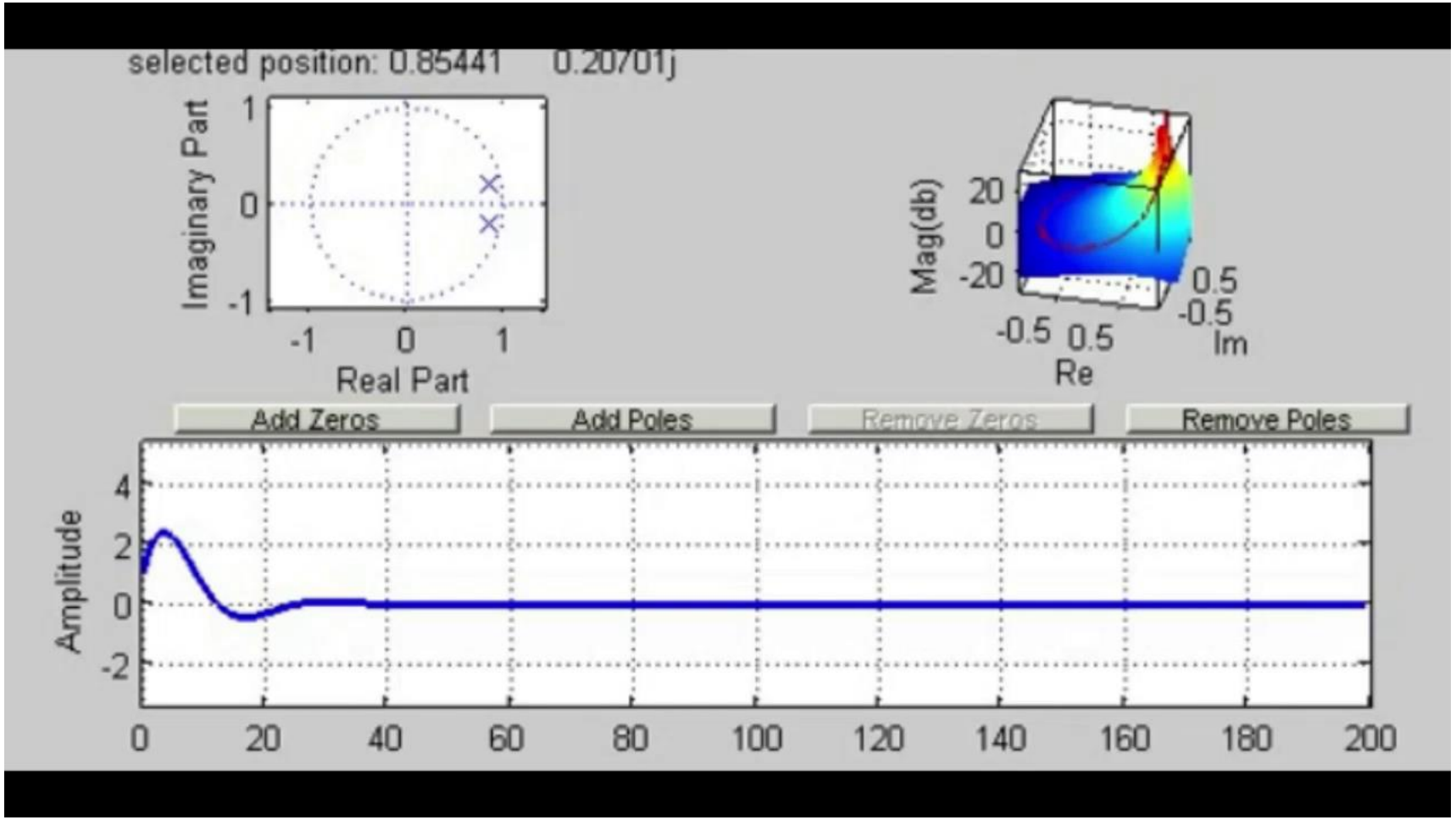
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{M-1} b_i \cdot z^{-i}}{\sum_{k=0}^{N-1} a_k \cdot z^{-k}} = \frac{b_0 + b_1 \cdot z^{-1} b_2 \cdot z^{-2} + \cdots b_{M-1} \cdot z^{-M-1}}{a_0 + a_1 \cdot z^{-1} a \cdot z^{-1} + \cdots a_{N-1} \cdot z^{-N-1}}$$



s-plane



z-plane


selected position: 0.85441 0.20701j
Imaginary Part
Real Part
Mag(db)
Re
Im
Add Zeros
Add Poles
Remove Zeros
Remove Poles
Amplitude

DT Oscillator by using unstable IIR Filter:

An infinite impulse response (IIR) digital filter can be mathematically expressed by the equation below. Zeros of the transfer function, H(z), are the roots of the numerator polynomial. Poles of the transfer function are the roots of the denominator polynomial. We will use specific placement of the poles and zeros to cause the output to ring or oscillate indefinitely at specific frequencies.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{M-1} b_i \cdot z^{-i}}{\sum_{k=0}^{N-1} a_k \cdot z^{-k}} = \frac{b_0 + b_1 \cdot z^{-1} b_2 \cdot z^{-2} + \cdots b_{M-1} \cdot z^{-M-1}}{a_0 + a_1 \cdot z^{-1} a \cdot z^{-1} + \cdots a_{N-1} \cdot z^{-N-1}}$$

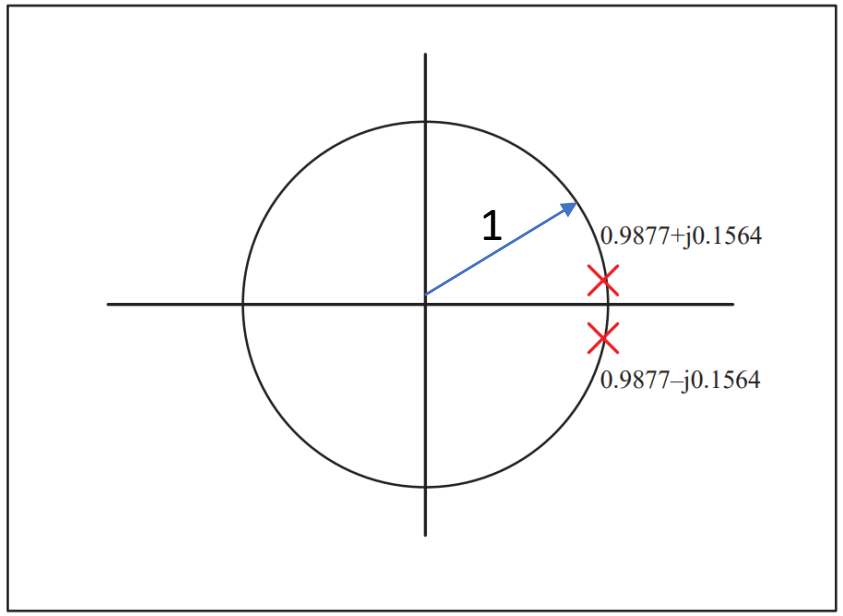
The z transform of a sine wave is a second order IIR filter that is expressed by the Equation below. It has two complex poles at $e^{\pm j2\pi Fo/Fs}$ and two zeros at the origin.

$$H(z) = \frac{\sin(2\pi f_0/F_s) \cdot z^{-1}}{1 - 2 \cdot \cos\left(\frac{2\pi f_0}{F_s}\right) \cdot z^{-1} + z^{-2}} = \frac{Z_1 \cdot z^{-1}}{(1 - p_0 \cdot z^{-1}) \cdot (1 - p_1 \cdot z^{-1})}$$

$$H(Z)=\frac{Y[Z]}{X[Z]}=\frac{1}{1-1.9754Z^{-1}+Z^{-2}}$$

a+jb

a-jb



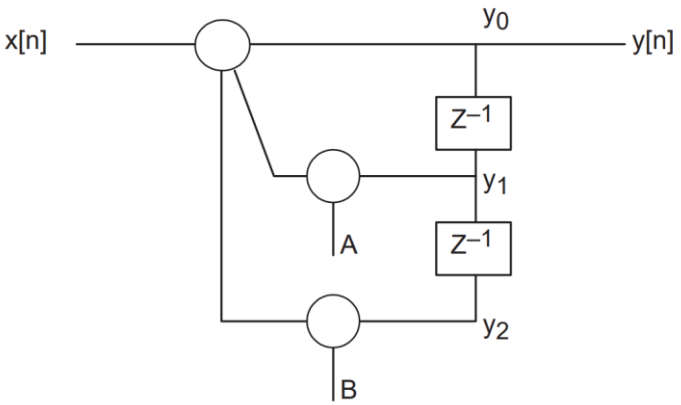
$$y[n]=A\bullet y[n-1]+B\bullet y[n-2]$$

$$y[2] = A\bullet y[1] + B\bullet y[0],$$

$$y[3] = A\bullet y[2] + B\bullet y[1].$$

$$y_2 = x[0] = \sin\left(\frac{2\pi}{40}\cdot 0\right) = 0$$

$$y_1 = x[1] = \sin\left(\frac{2\pi}{40}\cdot 1\right) \cong 0.1564$$



$$y[0] = \sin\left(\frac{2\pi}{40}\cdot 0\right) = 0$$

$$y[1] = \sin\left(\frac{2\pi}{40}\cdot 1\right) \cong 0.1564$$

$$y[2] = \sin\left(\frac{2\pi}{40}\cdot 2\right) \cong 0.3090$$

$$y[3] = \sin\left(\frac{2\pi}{40}\cdot 3\right) \cong 0.4540$$

$$0.3090 = A \times 0.1564 + B \times 0,$$

$$0.4540 = A \times 0.3090 + B \times 0.1564$$

$$y[n] = 1.9754\bullet y[n-1] - y[n-2] + x[n]$$

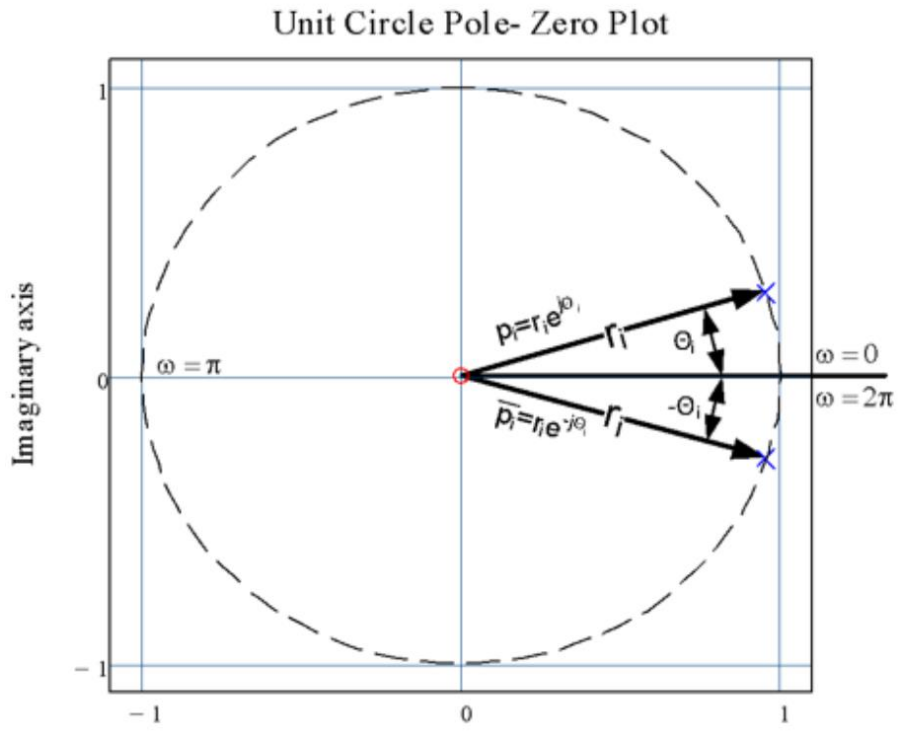
A

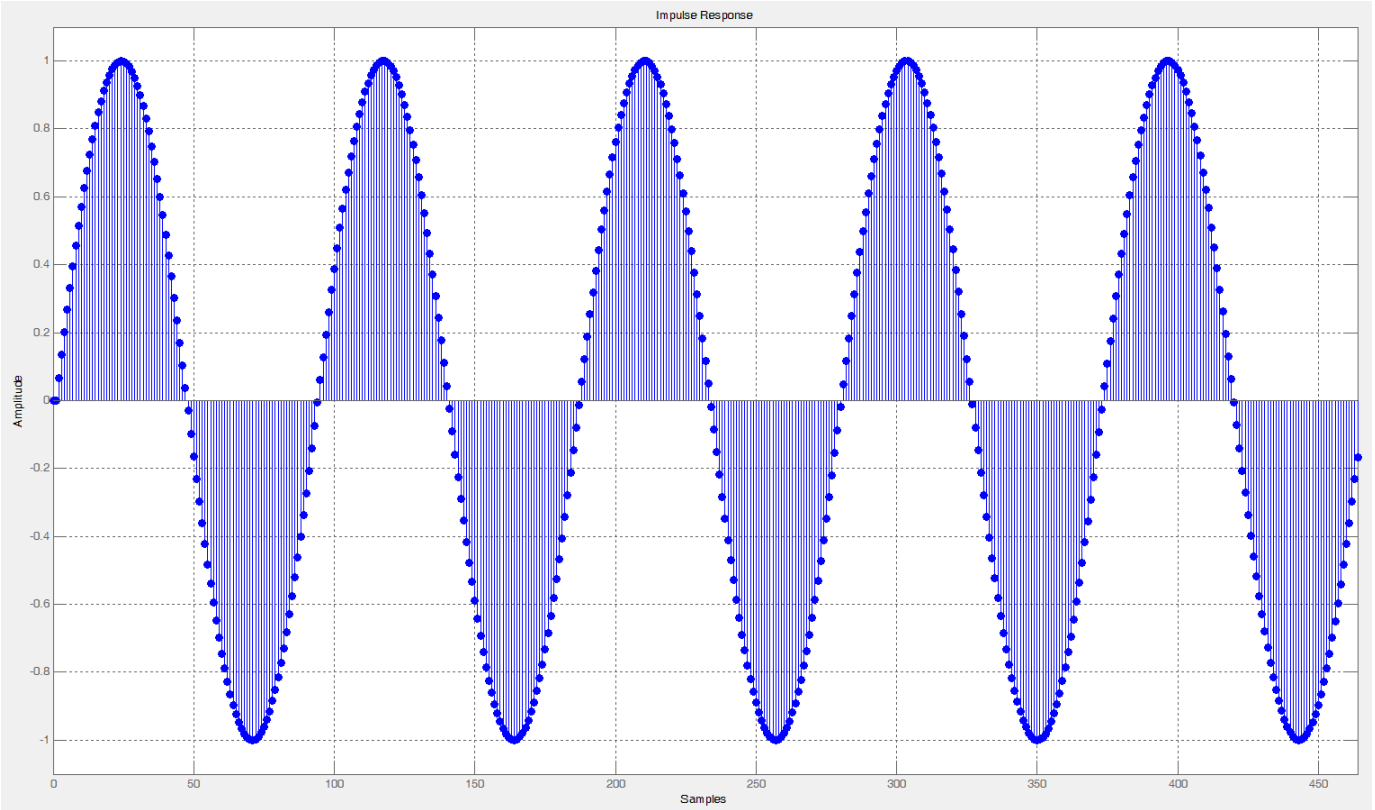
B=-1

Poles and zeros are the roots of the denominator and numerator polynomials, respectively expressed in above Eqs. These roots can be represented in the frequency domain as a vector, as shown in Fig. and expressed by below Eq. The variable r represents the magnitude of the vector and the angle Θ represents the frequency normalized by the sampling frequency, expressed in radians.

$$p_{0,1} = r_i \cdot e^{\pm j0_i}$$

r*exp(j*Θ) = r*[cos(Θ) + j*sin(Θ)]





$$y[n] = -a_1 \cdot y[n-1] - a_2 \cdot y[n-2]$$

$$a_1 = -\cos\left(\frac{2\pi F_0}{Fs}\right) \qquad a_2 = 1 \qquad y[1] = \sin\left(\frac{2\pi F_0}{Fs}\right) \qquad y[0] = 0 \qquad n > 1$$

Fs=1000Hz, F0=100Hz a1=-0.809 y[1]=0.588, a2=1 y[0]=0
y[k]=0.809*y[k-1]-y[k-2],
y[0]=0
y[1]=0.588
y[2]=0.809*0.588+0
y[3]=0.809*(0.809*0.588)-0.588
.....

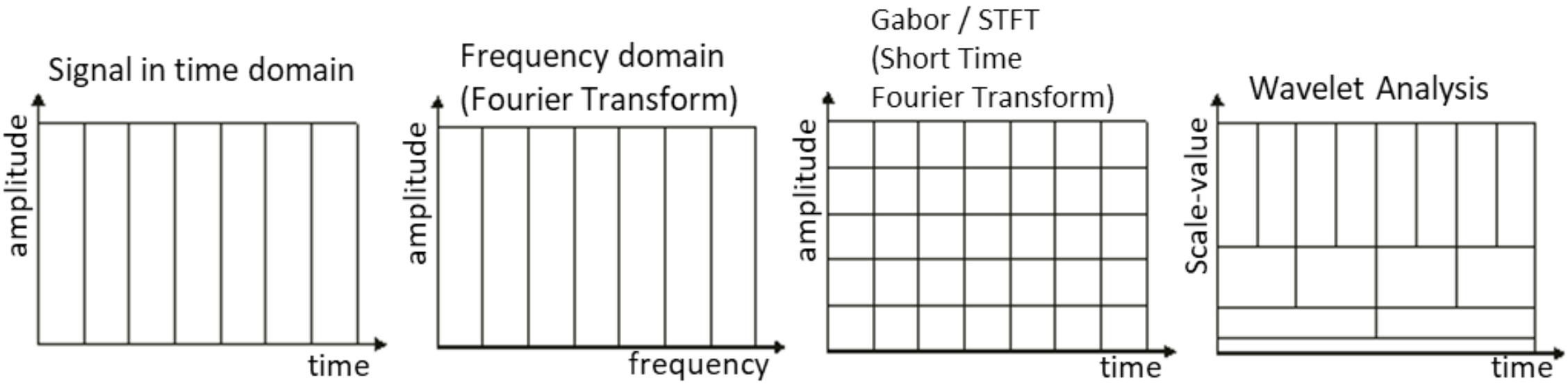
The digital oscillator can be implemented as an infinite impulse response (IIR) digital filter with two multipliers, two delays, and an adder. Since we have a digital filter, it necessarily has an input and at least one output. The input, x[n], needs to be a value of 1 for x[0] (n=0), and x[n] must be 0 for all n after n=0. The filter has two poles and the poles of the filter are located at z=(1-exp(j*Θ)) and z=(1-exp(j*Θ)).

Here's a 16-point example with a table for the values that shows the terms needed for computing the output values and the total y[n] output for the filter. The sine output is the last column. The numbers with an exponent of E-15 or E-16 are essentially zero. All angles are in radians. Here are the constants for my example:

sin(Θ) = 0.3823864
r = 0.999, r^2 = 0.998001
2*r*cos(Θ) = 1.845911306

Below the table is a plot of the table points showing a sine wave.

Point#	x[n]	sin(Θ)*x[n]	y[n-1]	y[n-2]	y[n]=Output
1	1	0.382683432	0	0	0.382683432
2	0	0	0.382683432	0	0.706399674
3	0	0	0.706399674	0.382683432	0.922032697
4	0	0	0.922032697	0.706399674	0.997002999
5	0	0	0.997002999	0.922032697	0.920189554
6	0	0	0.920189554	0.997002999	0.703578311



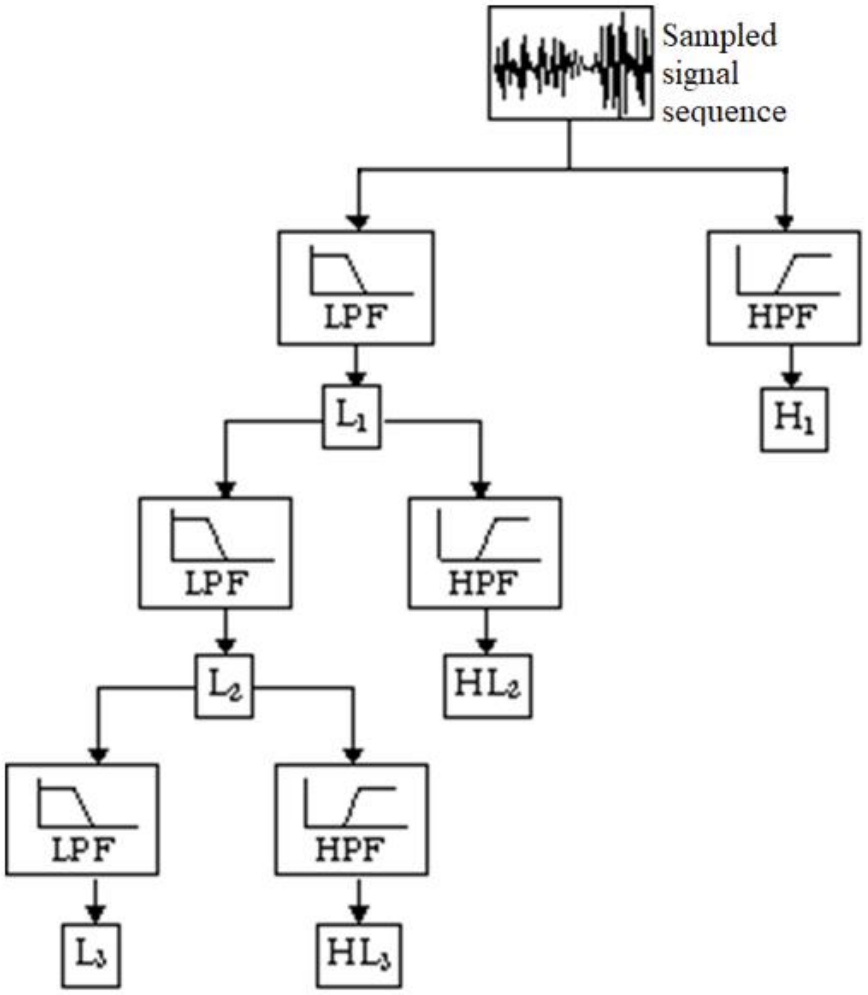
Representation of a signal pattern a) as amplitude variation in time a) Frequency spectrum as Fourier transform c) Short Time Fourier Transform d) Wavelet transform (Scale-value)

Continuous wavelet transform (CWT) is a convolution of the input data sequence with a set of functions generated by the mother wavelet. Discrete Wavelet Transformation (DWT) depends on a similar convolution in discrete time. DWT coefficients $W(j,k)$ can be given as,

$$W(j,k) = \sum_{n=0}^{N-1} f(n) * \psi_{j,k}^{*}(n)$$

where $f(n)$ is a sequence with length N, and $\psi_{j,k}^{*}(n)$ is the discretized mother wavelet function.

Dyadic filter tree algorithm representing a wavelet basis as high-pass (HPF) and low-pass filter (LPF) bank



A very simple example is using the average and difference of even and odd values in the sampled signal sequence. As an example, suppose we are given a 1D image consisting of 4 pixels, A= [8 6 2 4]. If we apply the DWT process shown in figure 11.20 then we can go down to the second layer that yields L_2 and HL_2 because there are 4 samples in the sequence. Initial resolution is 4 and respecting average column is the data sequence itself as shown in table 11.4. If we simply use the average for the low pass filter and the average of the difference as the high pass filter then we will get the data in the row of resolution 2. Here, 7 is the average of 8 and 6; 3 is the average of 2 and 4. First set of details H_1 contains 1 as the average of the difference between 8 and 6, and it contains -1 as the average of the difference between 2 and 4 (i.e., (2-4)/2=-1). If we continue to this filtering iteration, L_2 will be the last average as 5. It is also the average of all data in the sequence. It is also referred to as the wavelet coefficient a_0. The last detail is HL_2=(7-3)/2=2.

Resolution	Average	Detail Coefficients
4	[8 6 2 4]	[]
2	L_1=[7 3]	H_1=[1 -1]
1	L_2=[5]	HL_2=[2]

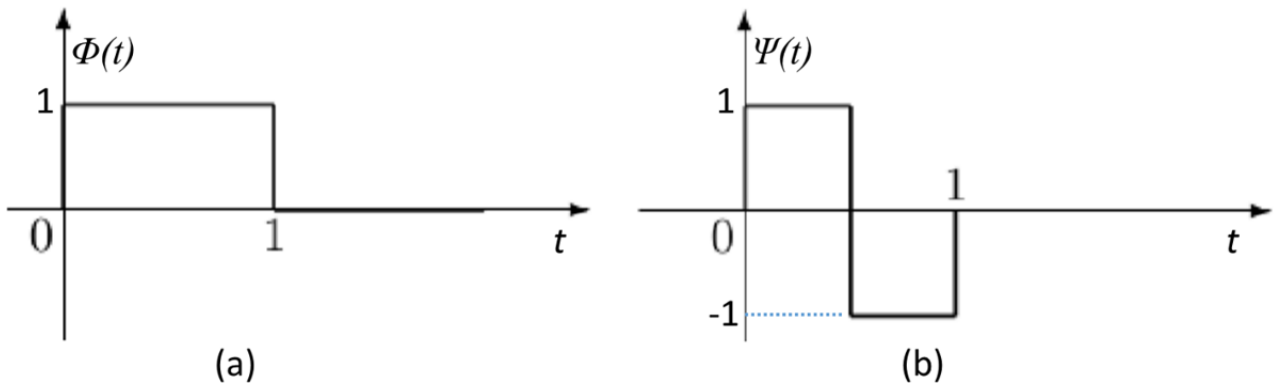
Hence a Haar wavelet representation of pixel sequence A corresponds to [L_2 HL_2 H_1]=[5 2 1 -1]

The superscript * denotes a complex conjugate (Smith et al., 1998). Haar scaling function Φ(t) and the wavelet function Ψ(t) are defined as,

$$\phi(t) = \begin{cases} 1, & if\ 0 \le t < 1 \\ 0, & otherwise \end{cases}$$

$$\psi(t) = \phi(2t) - \phi(2t-1)$$

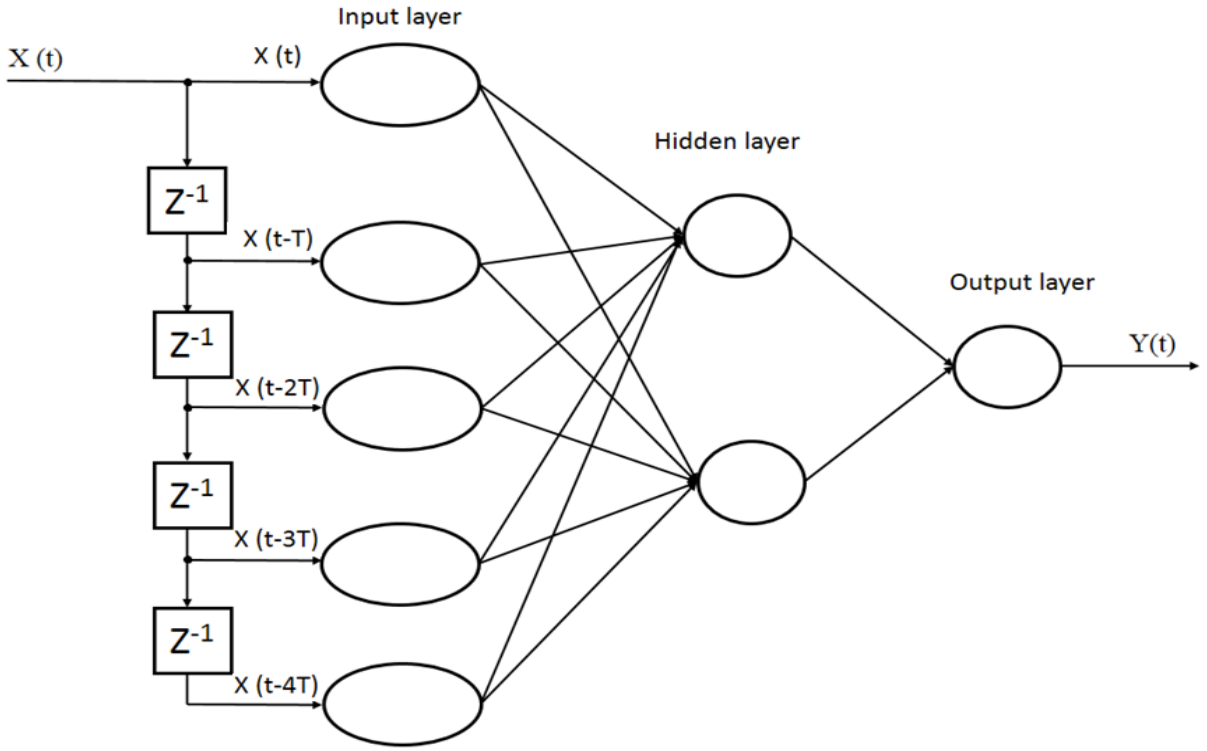
$$\psi(t) = \begin{cases} 1, for\ t\epsilon \left[0, \frac{1}{2}\right), \\ -1, for\ t\epsilon \left[\frac{1}{2}, 1\right), \\ 0, otherwise. \end{cases}$$



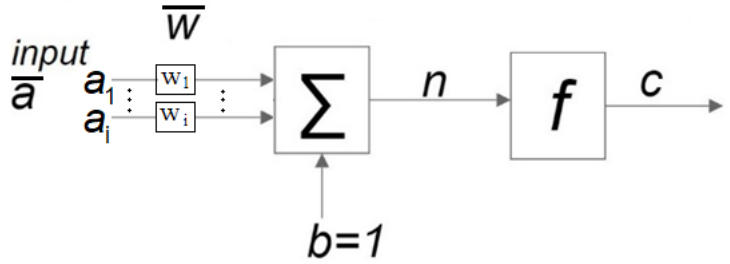
a) Haar scaling function Φ(t) and b) Haar wavelet function Ψ(t)

Additional Applications and Transformations:
(Recommended for Future Study)

Time-delay network example for processing of temporal data:



A simplified neuron model used in fully connected layers:

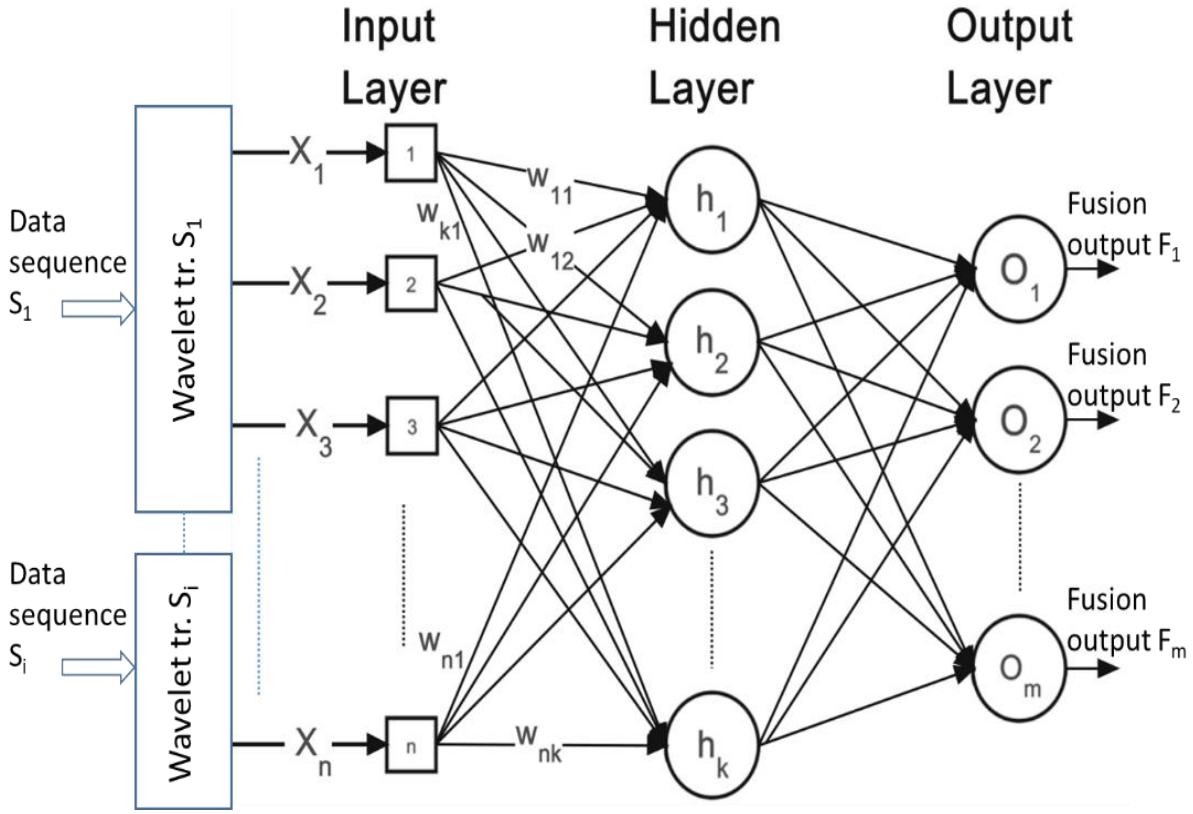


$$c = f\left(\sum w_i * a_i + b\right)$$

Log-sigmoid activation function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Example application of transformations for pre-processing of signal/data

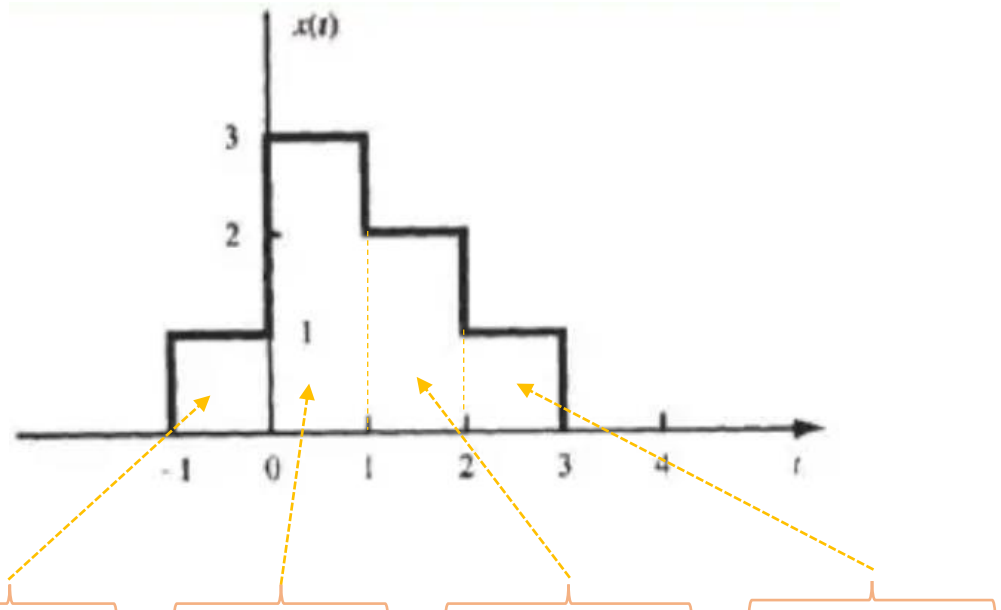


A data fusion model based on wavelet neural network

Quiz Questions for Past Lectures

Q1.

Express the signals shown in the figure in terms of unit step functions:

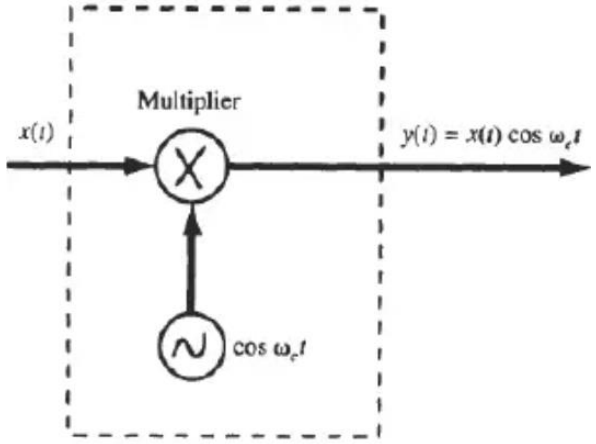


$x(t)=(u(t+1)-u(t))+3(u(t)-u(t-1))+2(u(t-1)-u(t-2))+(u(t-2)-u(t-3))$

$x(t)=u(t+1)+2u(t)-u(t-1)-u(t-2)-u(t-3)$

Q2. Consider the system shown in the figure and determine,

a) whether it is memoryless,
b) whether it is causal,
c) whether it is linear,
d) whether it is time-invariant,
e) whether it is stable



a) $y(t) = T\{x(t)\} = x(t)\cos\omega_c(t)$ → It is memoryless because the output depends only the current value of x(t).

b) $y(1)=x(1)\cos\omega_c(1)$, $y(2)=x(2)\cos\omega_c(2)$ → It is casual because the output is not dependent on future values of x(t).

c) If we check the linearity criteria then $x(t) = \alpha_1 x(t) + \alpha_2 x(t)$ must be satisfied.

$y(t) = [\alpha_1 x1(t) + \alpha_2 x2(t)]\cos\omega_c t = \alpha_1 x_1(t)\cos\omega ct + \alpha_2 x_2(t)\cos\omega_c t = \alpha_1 y_1(t) + \alpha_2 y_2(t)$ → the system is linear.

d) If y1(t) is the output of the shifted input, then $x_1(t) = x(t - t_0)$ → $y_1(t) = T\{x(t-t_0)\} = x(t-t_0)\cos\omega_c(t)$

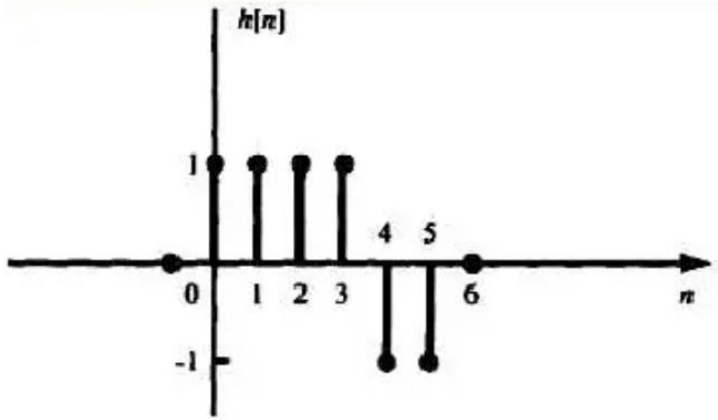
$y(t-t_0) = x(t-t_0)\cos\omega_c(t-t_0 \neq y_1(t)$ hence the system is not time invariant

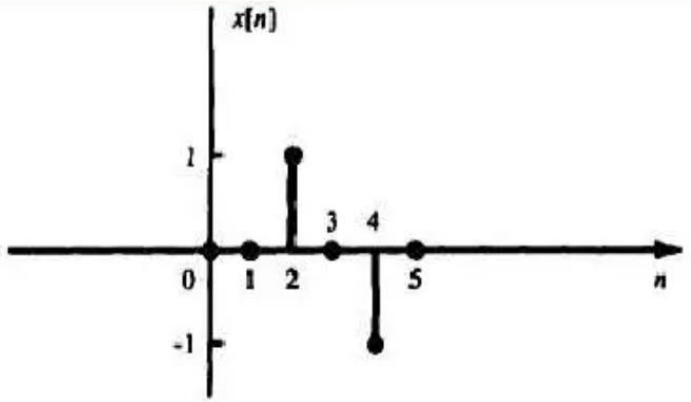
e) Output range depends only the range of input because $|\cos\omega_c t| \leq 1$ and $|y(t)| = |x(t)\cos\omega_c t| \leq |x(t)|$

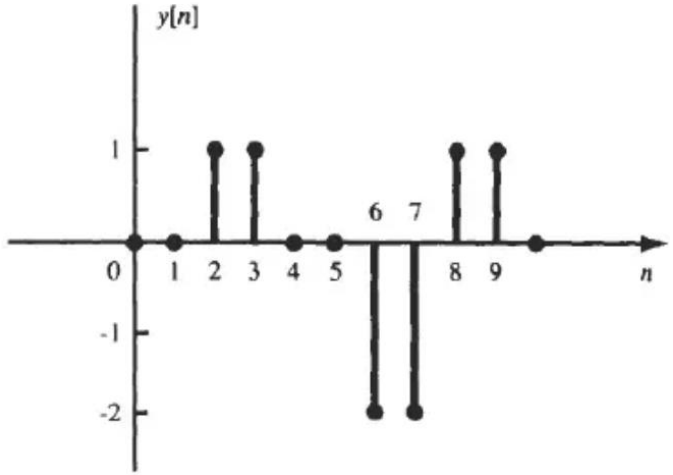
In this case, if the input x(t) is bounded, then the output y(t) is also bounded and the system is BIBO stable

Q3.

The impulse response h[n] of a discrete-time LTI system is given below in the figure. Determine and sketch the output y[n] of this system to the input x[n]







$$h[n] = \delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3] - \delta[n-4] - \delta[n-5]$$

$$x[n] = \delta[n-2] - \delta[n-4]$$

$$x[n] * h[n] = x[n] * \{\delta[n] + \delta[n-1] + \delta[n-2] + \delta[n-3] - \delta[n-4] - \delta[n-5]\}$$

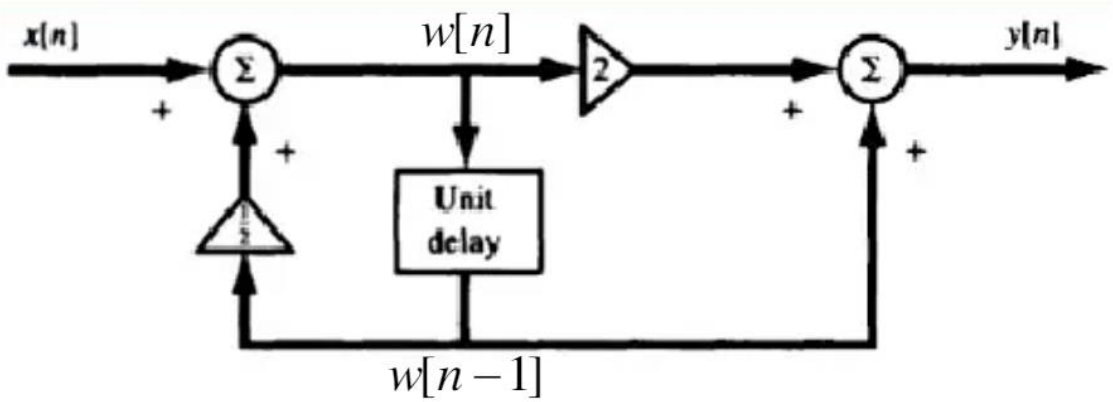
$$= x[n] + x[n-1] + x[n-2] + x[n-3] - x[n-4] - x[n-5]$$

$$y[n] = \delta[n-2] - \delta[n-4] + \delta[n-3] - \delta[n-5] + \delta[n-4] - \delta[n-6] + \delta[n-5] - \delta[n-7] - \delta[n-6] + \delta[n-8] - \delta[n-7] + \delta[n-9]$$

$$= \delta[n-2] + \delta[n-3] - 2\delta[n-6] - 2\delta[n-7] + \delta[n-8] + \delta[n-9]$$

$$y[n] = \{0,0,1,1,0,0,-2,-2,1,1\}$$

Q4. Find the difference equation and the respecting transfer function for the given discrete time system



$$w[n] = x[n] + \frac{1}{2} w[n-1] \qquad y[n] = 2w[n] + w[n-1]$$

$$w[n-1] = \frac{1}{2} y[n] - x[n] \qquad w[n] = \frac{1}{4} y[n] + \frac{1}{2} x[n]$$

If we change n by n-1 $$w[n-1] = \frac{1}{4} y[n-1] - \frac{1}{2} x[n-1]$$

$$\frac{1}{2} y[n] - x[n] = \frac{1}{4} y[n-1] - \frac{1}{2} x[n-1]$$

$$2y[n] - 4x[n] = y[n-1] - 2x[n-1]$$

$$\mathrm{H(z)} = \frac{Y(z)}{X(z)} = \frac{2 - 1z^{-1}}{1 - 0.5z^{-1}}$$

$2y[n]-y[n-1]=4x[n]-2x[n-1]$

$2Y(z)-z^{-1}Y(z)=4X(z)-2z^{-1}X(z)$

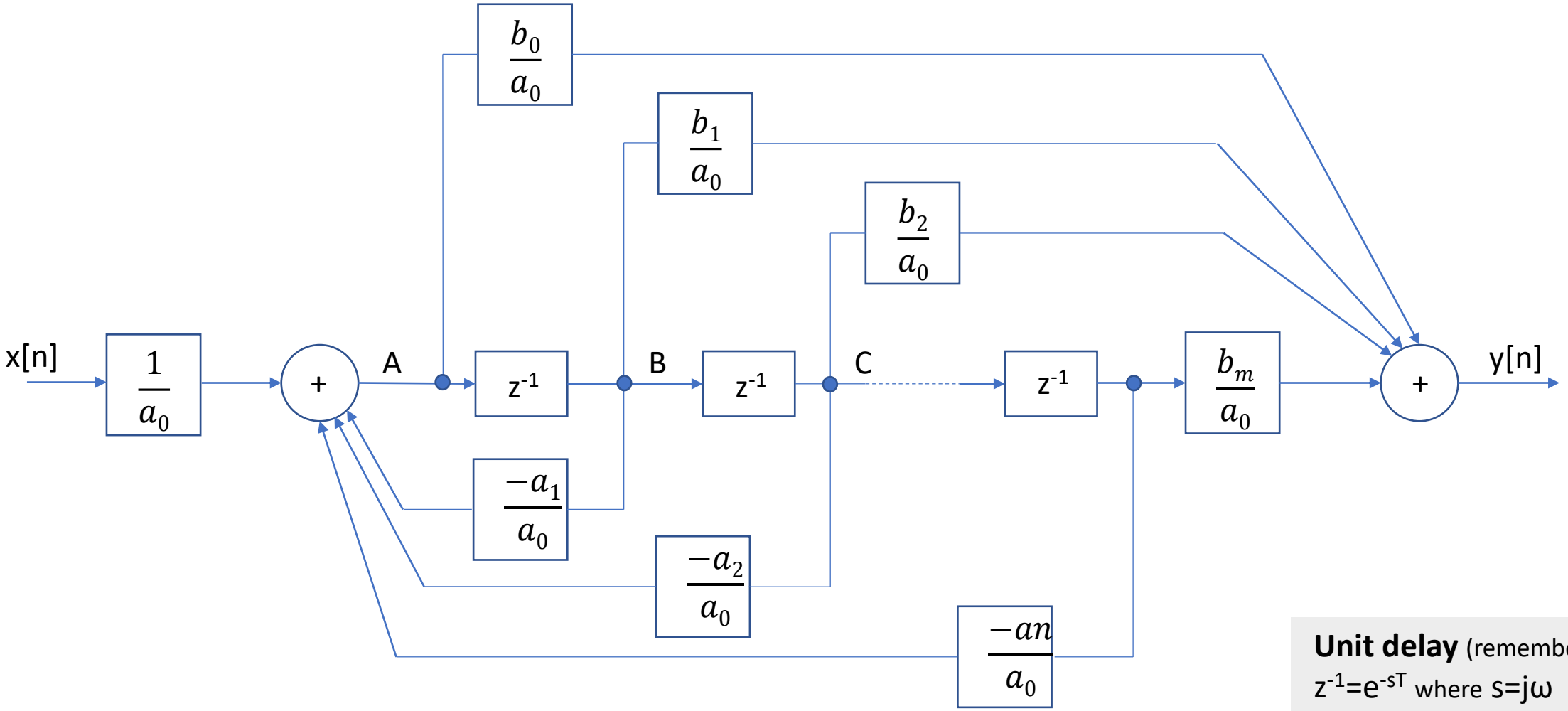
$2Y(z)-z^{-1}Y(z)=4x(z)-2z^{-1}x(z)$

$(2-z^{-1})Y(z)=(4-2z^{-1})X(z)$

$T(z)=Y(z)/X(z) =(4-2z^{-1})/ (2-z^{-1})$

$T(z)=(2-z^{-1})/ (1-0.5z^{-1})$

Reminder: Direct Programming (canonical form) of $H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \ldots\ldots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \ldots\ldots + a_n z^{-n}}$



Unit delay (remember):

$z^{-1}=e^{-sT}$ where $s=j\omega$

$B=A\cdot z^{-1} \Rightarrow B(k)=A(k-1)$

Q4. Impulse response of a system is given by $h[n] = u[n] - u[n-N]$

Find the output y[n] for the input x [n] given as $x[n] = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$

$$h[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise.} \end{cases} \qquad x[n] = a^n u[n]$$

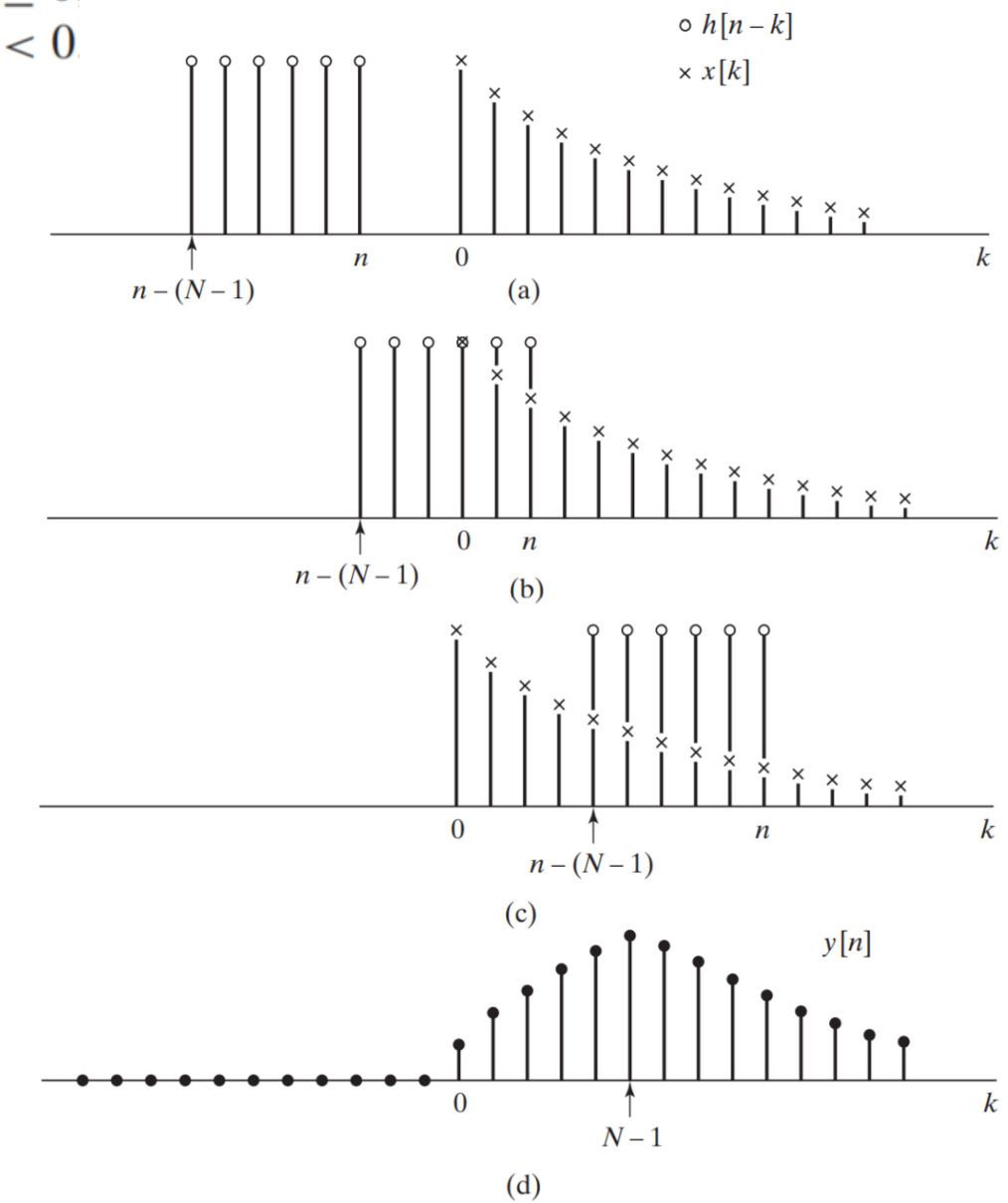
$$y[n] = 0, \qquad n < 0$$

$$x[k]h[n-k] = a^k, \qquad \text{for } 0 \leq k \leq n \quad \text{when } 0 \leq n \leq N-1$$

$$y[n] = \sum_{k=0}^{n} a^k, \qquad \text{for } 0 \leq n \leq N-1$$

$$\sum_{k=N_1}^{N_2} \alpha^k = \frac{\alpha^{N_1} - \alpha^{N_2+1}}{1-\alpha}, \qquad N_2 \geq N_1$$

$$y[n] = \frac{1-a^{n+1}}{1-a}, \qquad 0 \leq n \leq N-1$$



when $0 < n - N + 1$ or $N - 1 < n$ $\rightarrow$ $x[k]h[n-k] = a^k, \qquad n - N + 1 \leq k \leq n$

$$y[n] = \sum_{k=n-N+1}^{n} a^k, \qquad \text{for } N - 1 < n$$

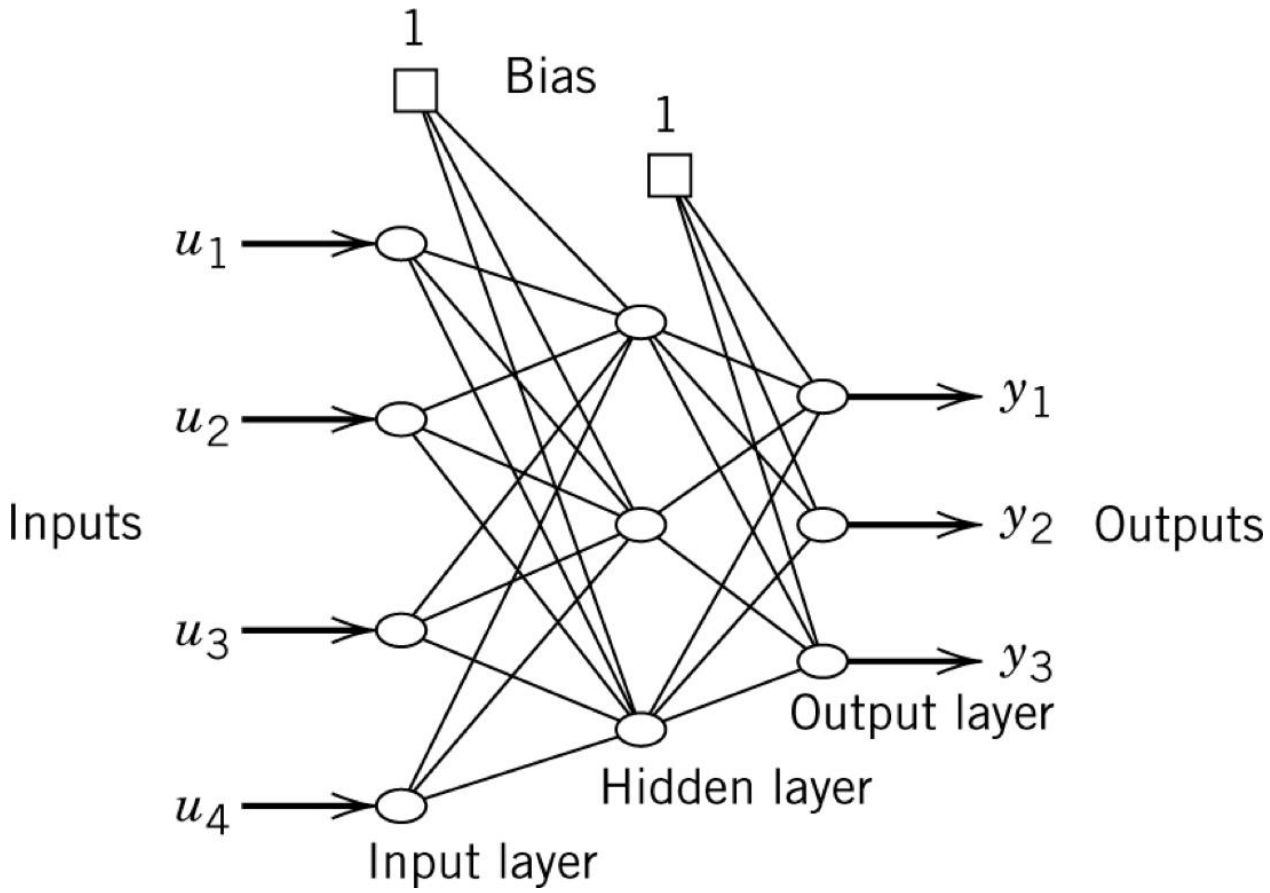
$$y[n] = \frac{a^{n-N+1} - a^{n+1}}{1-a} = a^{n-N+1}\left(\frac{1-a^N}{1-a}\right)$$

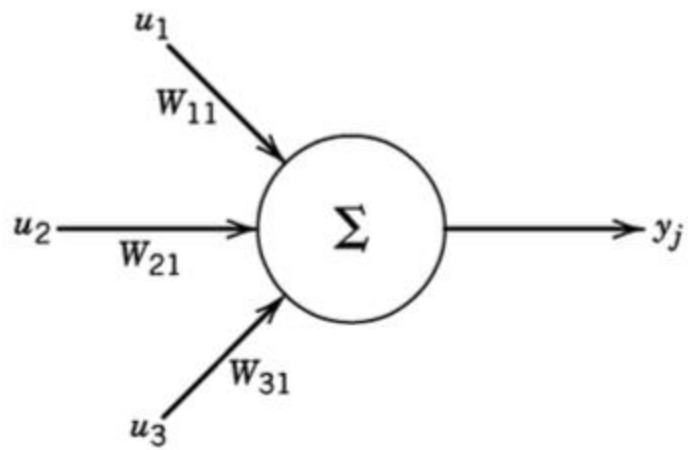
$$y[n] = \begin{cases} 0, & n < 0, \\ \dfrac{1-a^{n+1}}{1-a}, & 0 \leq n \leq N-1 \\ a^{n-N+1}\left(\dfrac{1-a^N}{1-a}\right), & N-1 < n. \end{cases}$$

x[n]={3, -1, 3, -5} Find the 4 points DFT of this signal if it is sampled at 4Hz

X[k]=? Draw the frequency response (DFT)

Regression of the Signals:

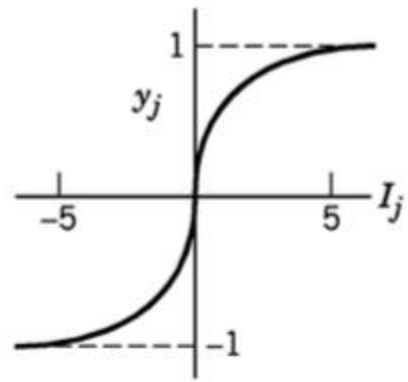




u_i = Input signals

W_{ij} = Weights

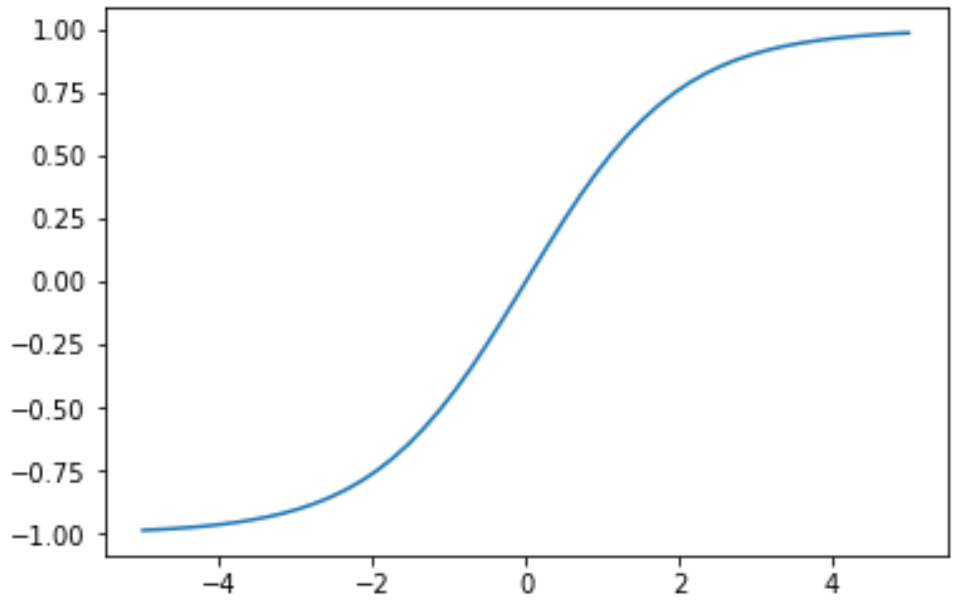
y_j = Output signal



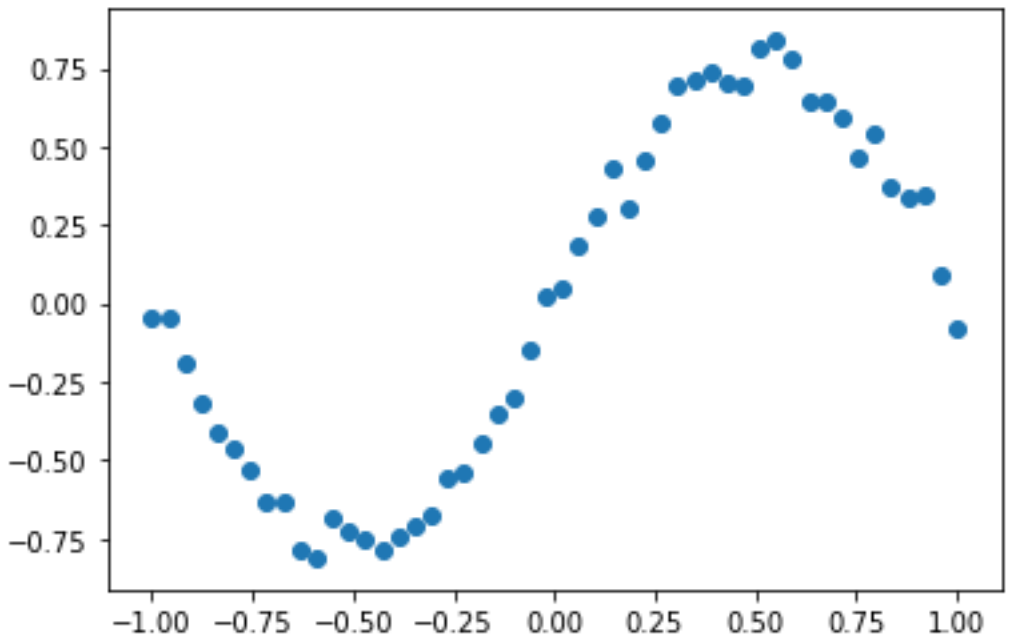
$$I_j = \sum_{j=1}^{n} W_{ij} u_i$$

$$y_j = \frac{1 - e^{-I_j}}{1 + e^{-I_j}}$$

```
import numpy
import matplotlib.pyplot as plt

Nhidden = 3

w_in_hidden = numpy.random.rand(Nhidden)
w_hidden_out = numpy.random.rand(Nhidden)

bias_hidden = numpy.random.rand()
bias_output = numpy.random.rand()

def sigmoid(i):
    expi = numpy.exp(-i)
    return ((1 - expi)/(1 + expi))

x = numpy.linspace(-5, 5)
plt.plot(x, sigmoid(x))
```



```
def network_output(u, w_in_hidden, w_hidden_out,
bias_hidden, bias_output):
    h = sigmoid(w_in_hidden*u + bias_hidden)
    y = sigmoid((w_hidden_out*h + bias_output).sum())

    return y

network_output(0.1, w_in_hidden, w_hidden_out,
bias_hidden, bias_output)

known_u = numpy.linspace(-1, 1)
known_y = numpy.sin(known_u*numpy.pi)*0.8 +
numpy.random.randn(len(known_u))*0.05

plt.scatter(known_u, known_y)
```



```
import scipy.optimize

def pack(w_in_hidden, w_hidden_out, bias_hidden, bias_output):
    return numpy.concatenate([w_in_hidden,
                              w_hidden_out,
                              numpy.array([bias_hidden]),
                              numpy.array([bias_output])])

def unpack(parameters):
    parts = numpy.split(parameters, [Nhidden, 2*Nhidden, 2*Nhidden + 1])
    return parts

p0 = pack(w_in_hidden, w_hidden_out, bias_hidden, bias_output)

def predict(parameters, us):
    w_in_hidden, w_hidden_out, bias_hidden, bias_output = unpack(parameters)
    return numpy.array([network_output(u, w_in_hidden, w_hidden_out, bias_hidden, bias_output) for u in us])

def plotfit(predictions):
    plt.scatter(known_u, known_y, alpha=0.4)
    plt.plot(known_u, predictions)

plotfit(predict(p0, known_u))
```