**AIM:** To study Sampling Theorem

**SOFTWARE USED:** Scilab 6.1.1

**THEORY:**

In signal processing, **sampling** is the reduction of a continuous-time signal to a discrete-time signal. A common example is the conversion of a sound wave to a sequence of samples.

A sample is a value or set of values at a point in time and/or space. When a source generates an analog signal and if that has to be digitized, having 1s and 0s i.e., High or Low, the signal has to be discretized in time.

This discretization of analog signal is called as **Sampling**.

**Sampling frequency** is the reciprocal of the sampling period. This sampling frequency, can be simply called as Sampling rate. The sampling rate denotes the number of samples taken per second, or for a finite set of values.

**Sampling Frequency** = 𝟏

𝑻𝒔 = 𝒇𝒔

where, Ts is sampling period,

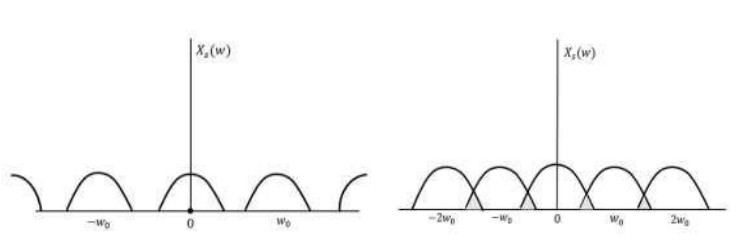
fs is sampling frequency.

For an analog signal to be reconstructed from the digitized signal, the sampling rate should be highly considered. The rate of sampling should be such that the data in the message signal should neither be lost nor it should get overlapped. Hence, a rate was fixed for this, called as **Nyquist rate.**

fs = 2W

This rate of sampling is called as **Nyquist rate.**

**Sampling Theorem**: The sampling theorem states that, “a signal can be exactly reproduced if it is sampled at the rate fs which is greater than twice the maximum frequency W.”



**CODE:**

clc;

fm=10e3;

fs=500e3;

ncyc=4;

t=0:(1/fs):(ncyc\*(1/fm));

x=sin(2\*3.14\*fm\*t);

subplot(3,1,1);

plot(t,x);

xlabel("time");

ylabel("amp");

title("message signal");

y=squarewave(2\*3.14\*fm\*t,100);

subplot(3,1,2);

plot2d3(t,y);

xlabel("time");

ylabel("amp");

title("impulse signal");

z=x.\*y;

subplot(3,1,3);

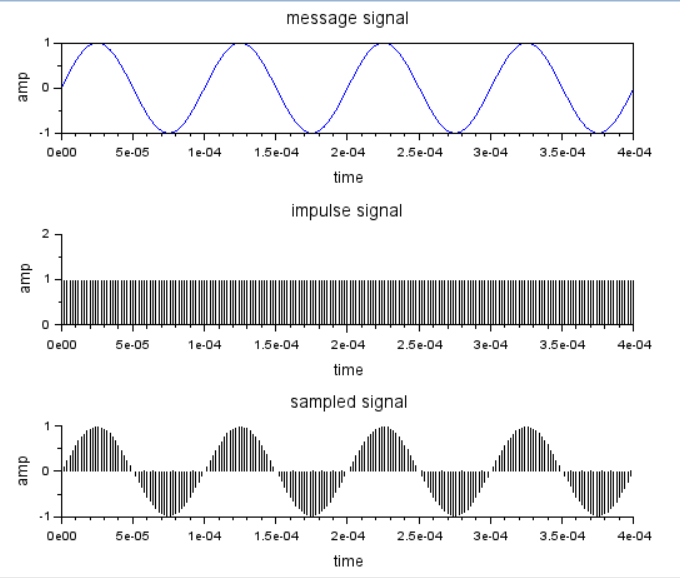
plot2d3(t,z);

xlabel("time");

ylabel("amp");

title("sampled signal");

**OUTPUT :**



**RESULT :**

Sampling has been studied and the sampled signal has been plotted successfully on the graph.

**AIM:** To Study of Pulse Code Modulation and Probability of error.

**SOFTWARE USED:** Scilab 6.1.1

**THEORY:**

A **line code** is the code used for data transmission of a digital signal over a transmission line. This process of coding is chosen so as to avoid overlap and distortion of signal such as sinter-symbol interference.

## **Properties of Line Coding**

## Following are the properties of line coding −

* As the coding is done to make more bits transmit on a single signal, the bandwidth used is much reduced.
* For a given bandwidth, the power is efficiently used.
* The probability of error is much reduced.
* Error detection is done and the bipolar too has a correction capability.
* Power density is much favorable.
* The timing content is adequate.
* Long strings of **1s** and **0s** is avoided to maintain transparency.

### **Types of Line Coding**

There are 3 types of Line Coding

* Unipolar
* Polar
* Bi-polar

## **Unipolar Signaling**

Unipolar signalling is also called as **On-Off Keying** or simply **OOK**.

The presence of pulse represents a **1** and the absence of pulse represents a **0**.

There are two variations in Unipolar signalling −

* Non Return to Zero NRZ
* Return to Zero RZ

### **Unipolar Non-Return to Zero NRZ**

In this type of unipolar signaling, a High in data is represented by a positive pulse called as **Mark**, which has a duration **T0** equal to the symbol bit duration. A Low in data input has no pulse.

### **Unipolar Return to Zero RZ**

In this type of unipolar signaling, a High in data, though represented by a **Mark pulse**, its duration **T0** is less than the symbol bit duration. Half of the bit duration remains high but it immediately returns to zero and shows the absence of pulse during the remaining half of the bit duration.

## **Polar Signaling**

There are two methods of Polar Signaling. They are −

* Polar NRZ
* Polar RZ

### **Polar NRZ**

In this type of Polar signaling, a High in data is represented by a positive pulse, while a Low in data is represented by a negative pulse.

### **Polar RZ**

In this type of Polar signaling, a High in data, though represented by a **Mark pulse**, its duration **T0** is less than the symbol bit duration. Half of the bit duration remains high but it immediately returns to zero and shows the absence of pulse during the remaining half of the bit duration.

However, for a Low input, a negative pulse represents the data, and the zero level remains same for the other half of the bit duration.

**Bipolar Signaling**

This is an encoding technique which has three voltage levels namely **+, -** and **0**. Such a signal is called as **duo-binary signal**.

An example of this type is **Alternate Mark Inversion**AMIAMI. For a **1**, the voltage level gets a transition from + to – or from – to +, having alternate **1s** to be of equal polarity. A **0** will have a zero voltage level.

Even in this method, we have two types.

* Bipolar NRZ
* Bipolar RZ

**Manchester Line Coding**

In telecommunication and data storage, Manchester coding (also known as Phase Encoding, or PE) is a line code in which the encoding of each data bit has at least one transition and occupies the same time. It therefore has no DC component, and is self-clocking, which means that it may be inductively or capacitively coupled, and that a clock signal can be recovered from the encoded data.

**CODE :**

clc;

pi=3.14;

x=[1,0,1,1,0,1,0,1];

nx=length(x);

sign=1;

i=1;

while i<nx+1

t=i:0.001:i+1-0.001;

if x(i)==1

unipolar\_nrz=squarewave(t\*2\*pi,100);

ami\_nrz=sign\*squarewave(t\*2\*pi,100);

unipolar\_rz=(1+squarewave(t\*2\*pi,50))/2;

polar\_rz=(1+squarewave(t\*2\*pi,50))/2;

polar\_rz=(1+squarewave(t\*2\*pi,50))/2;

ami\_rz=sign\*(1+squarewave(t\*2\*pi,50))/2;

sign=sign\*-1;

manchester\_code=squarewave(t\*2\*pi,50);

else

unipolar\_nrz=0;

ami\_nrz=0;

unipolar\_rz=0;

polar\_rz=-(1+squarewave(t\*2\*pi,50))/2;

ami\_rz=0;

manchester\_code=-squarewave(t\*2\*pi,50);

end;

subplot(4,2,1);

plot(t,unipolar\_nrz);

ylabel('unipolar\_nrz');

set(gca(),"grid",[1 1]);

a=gca();

a.data\_bounds=[1 -2;9 2]

subplot(4,2,5);

plot(t,ami\_nrz,'b');

ylabel('ami\_nrz');

set(gca(),"grid",[1 1]);

a=gca();

a.data\_bounds=[1 -2;9 2]

subplot(4,2,2);

plot(t,unipolar\_rz,'r'); ylabel('unipolar\_rz');

set(gca(),"grid",[1 1]);

a=gca();

a.data\_bounds=[1 -2;9 2]

subplot(4,2,4);

plot(t,polar\_rz);

ylabel('polar\_rz');

set(gca(),"grid",[1 1]);

a=gca();

a.data\_bounds=[1 -2;9 2]

subplot(4,2,3);

plot(t,ami\_rz,'r');

ylabel('ami\_rz');

set(gca(),"grid",[1 1]);

a=gca();

a.data\_bounds=[1 -2;9 2]

subplot(4,2,7);

plot(t,manchester\_code,'r');

ylabel('manchester\_code');

set(gca(),"grid",[1 1]);

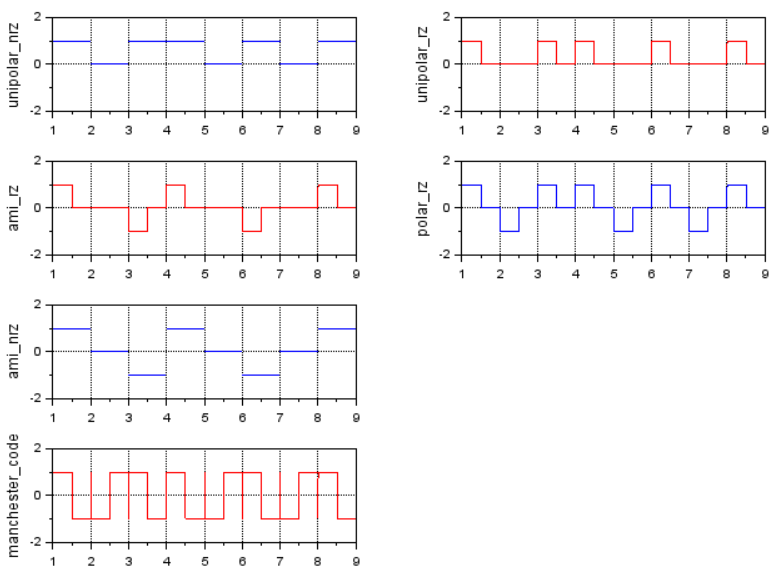
a=gca();

a.data\_bounds=[1 -2;9 2]

i=i+1;

end;

**OUTPUT :**



**RESULT :**

Line coding has been studied successfully using scilab 6.1.1

**AIM:** Write a program to study Delta Modulation and Demodulation.

**SOFTWARE USED:** Scilab 6.1.1

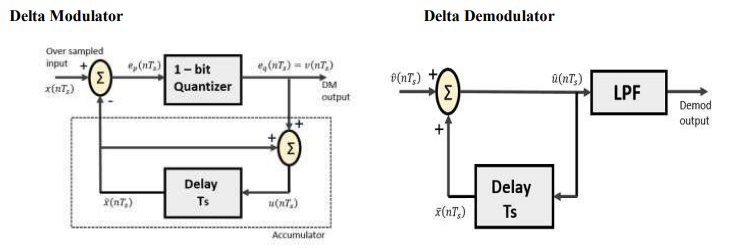
**THEORY:**

The type of modulation, where the sampling rate is much higher and in which the step size after quantization is of a smaller value **Δ**, such a modulation is termed as **delta modulation**.

Delta Modulation is a simplified form of DPCM technique, also viewed as **1-bit DPCM scheme**. As the sampling interval is reduced, the signal correlation will be higher.

**Features of Delta modulation:**

* An over-sampled input is taken to make full use of the signal correlation.
* The quantization design is simple.
* The input sequence is much higher than the Nyquist rate.
* The quality is moderate.
* The design of the modulator and the demodulator is simple.
* The stair-case approximation of output waveform



**CODE :**

pi = 3.14;

t = 0 : pi / 50 : pi;

x=sin(t);

plot (x, "r");

l = length(x);

delta = 0.2;

xn = 0;

for i = 1 : l,

if x(i) > xn(i)

d(i) = 1;

xn(i+1) = xn(i) + delta;

else

d(i) = 0;

xn(i+1) = xn(i) - delta;

end;

end;

plot2d2(xn);

for i=1:d,

if d(i ) > xn(i)

d(i) = 0;

xn(i+1) = xn(i) - delta;

else

d(i) = 1;

xn(i+1) = xn (i) + delta;

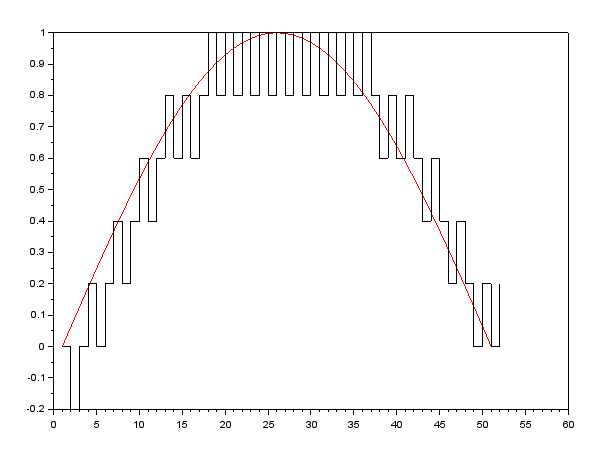
end;

end;

plot2d2(xn) ;

title("DM");

**OUTPUT :**

****

**RESULT :**

Delta Modulation has been studied successfully using scilab 6.1.1