INTRODUCTION

Basics of Video:

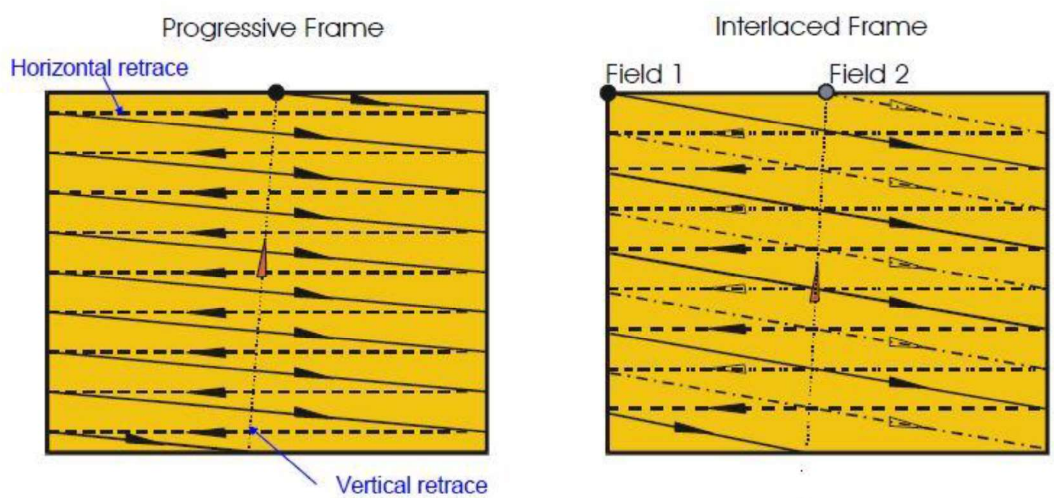
What is a Video Signal?

A video signal is a sequence of time varying images. In a video, the image sequence is in order and each image is called as a frame. So any video can be considered as a collection of frames.

Characteristics of a video signal:

Frame Rate:
Frame rate, the number of still pictures per unit of time of video.

Interlaced vs progressive:
In *progressive* scanning traces through a complete picture (a frame) row-wise for each time interval
In *interlaced* scanning odd-numbered lines are traced first, then the even-numbered lines. This results in odd and even fields—two fields make up one frame.



Aspect ratio: Aspect ratio describes the dimensions of video screens and video picture elements. All popular video formats are rectilinear, and so can be described by a ratio between width and height. The screen aspect ratio of a traditional television screen is 4:3. High definition televisions use an aspect ratio of 16:9.

Color space and bits per pixel: Color model name describes the video color representation. The number of distinct colors a pixel can represent depends on the number of *bits per pixel* (bpp).

Video quality: Video quality can be measured with formal metrics like PSNR or with subjective video quality using expert observation.

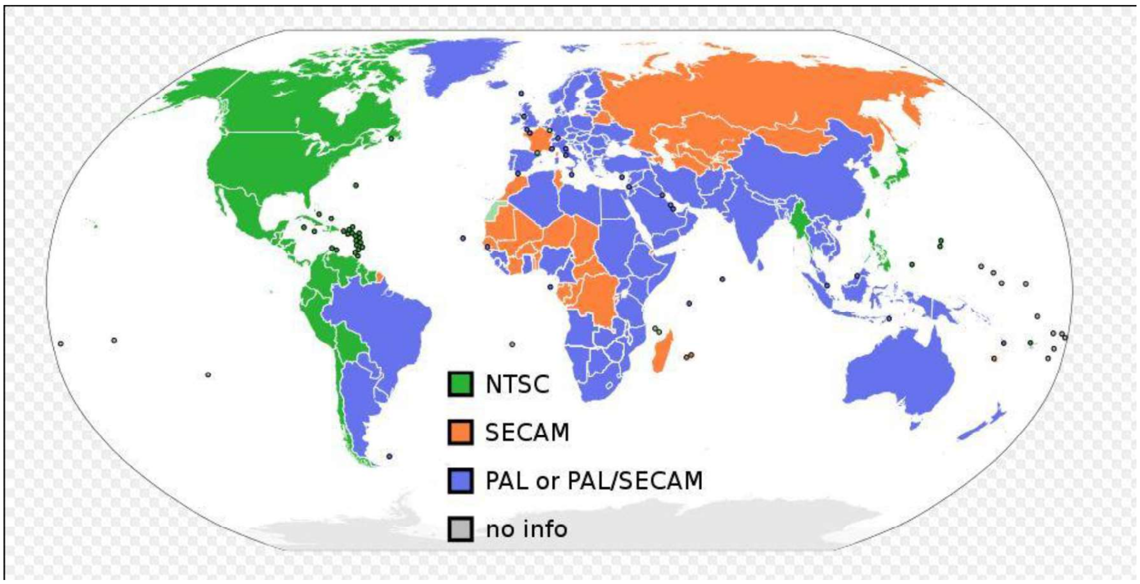
Video compression method (digital only): Video compression uses modern coding techniques to reduce redundancy in video data. Most video compression algorithms and codecs combine spatial image compression and temporal motion compensation. Video compression is a practical implementation of source coding in information theory. The most common modern standards are MPEG-2, used for DVD, Blu-ray and satellite television, and MPEG-4, used for AVCHD, Mobile phones (3GP) and Internet.

Broadcast Television Systems:

Broadcast television systems are encoding or formatting standards for the transmission and reception of terrestrial television signals. There are three main analog television systems in use around the world. They are-

NTSC: National Television Standards Committee
PAL: Phase Alteration Line
SECAM: Systeme Electronic Pour Coleur Avec Memoire

Analog color television encoding systems by nation



PROJECT OVERVIEW

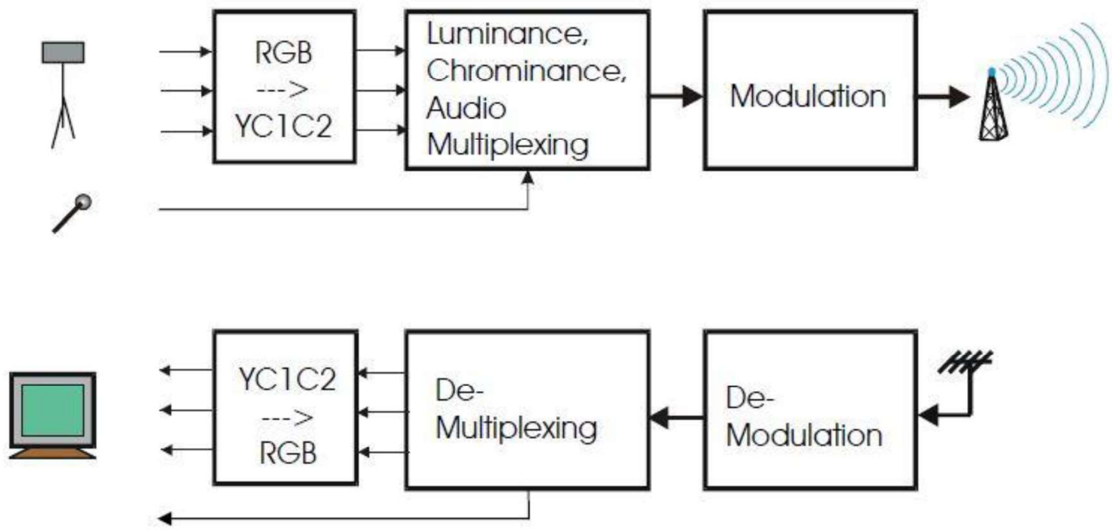
The aim of the project is to compare a digital video stream in both spatial and frequency domain. We achieve this by color space transformation of a video signal from RGB to YIQ color space in MATLAB. This will help us understand the principles of basic television system by digital implementation of analog signals, its processing and transmission.

YIQ is the color space used by the NTSC color TV system, employed mainly in North and Central America, and Japan. *I* stands for *in-phase*, while *Q* stands for *quadrature*, referring to the components used in quadrature amplitude modulation. The Y component represents the luminance information, and is the only component used by black-and-white television receivers. I and Q represent the chrominance information.

The matrix transform to convert RGB to YIQ is as follows:

$$\begin{bmatrix} Y' \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595879 & -0.274133 & -0.321746 \\ 0.211205 & -0.523083 & 0.311878 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

The aim of the project can be understood by the following block diagram:



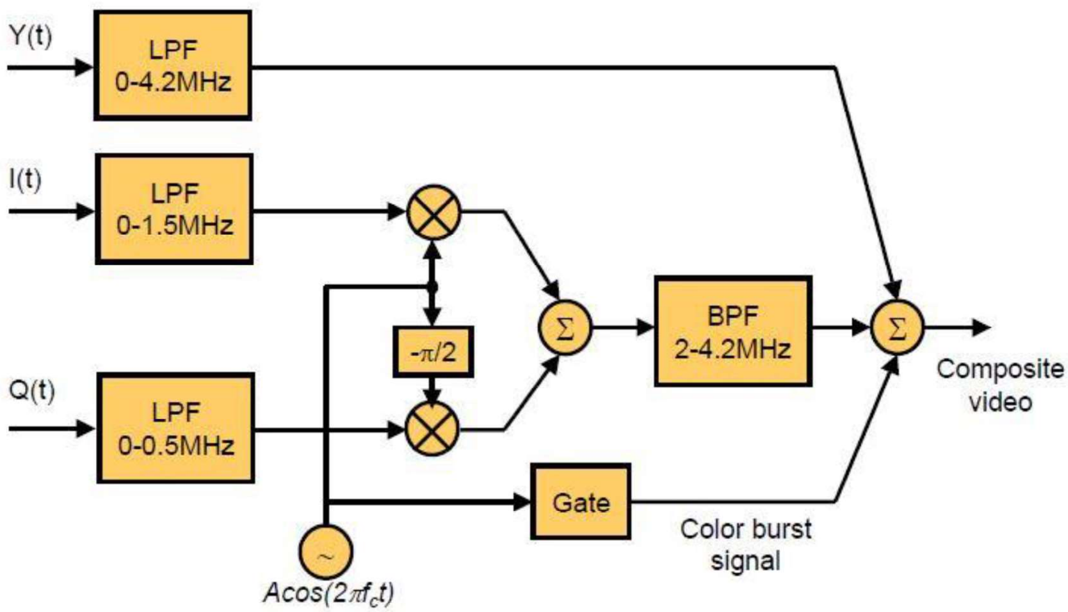
Once the original RGB image is processed into Y, I & Q components, all the three components are filtered separately and multiplexed together. This multiplexed signal is then modulated using QAM and transmitted to the receiving end.

The QAM modulated signal is then demodulated and fed to Demultiplexer where Y-component is removed from I and Q components. Then I and Q are separated with QAM Demodulator.

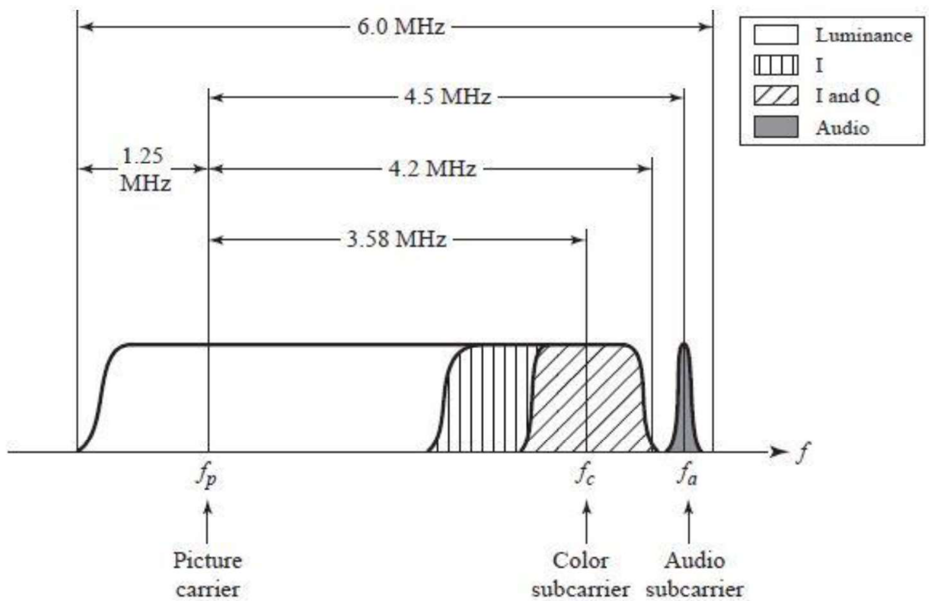
Then all the three Y, I & Q components are concatenated at the end to get a YIQ video stream. Finally, this YIQ image is then converted again to RGB color space and displayed on the screen.

The multiplexing block and the demultiplexing blocks are explained in detail below.

MULTIPLEXING BLOCK:

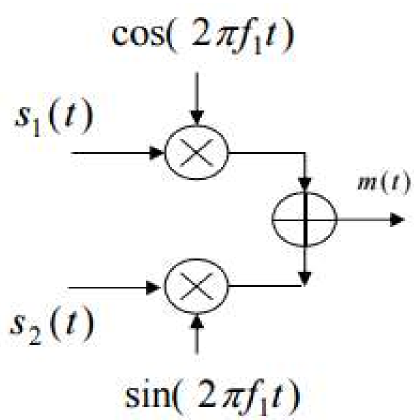


The overall spectrum of a color TV signal (including Y, I, Q and audio) when modulated over a carrier frequency is as follows.

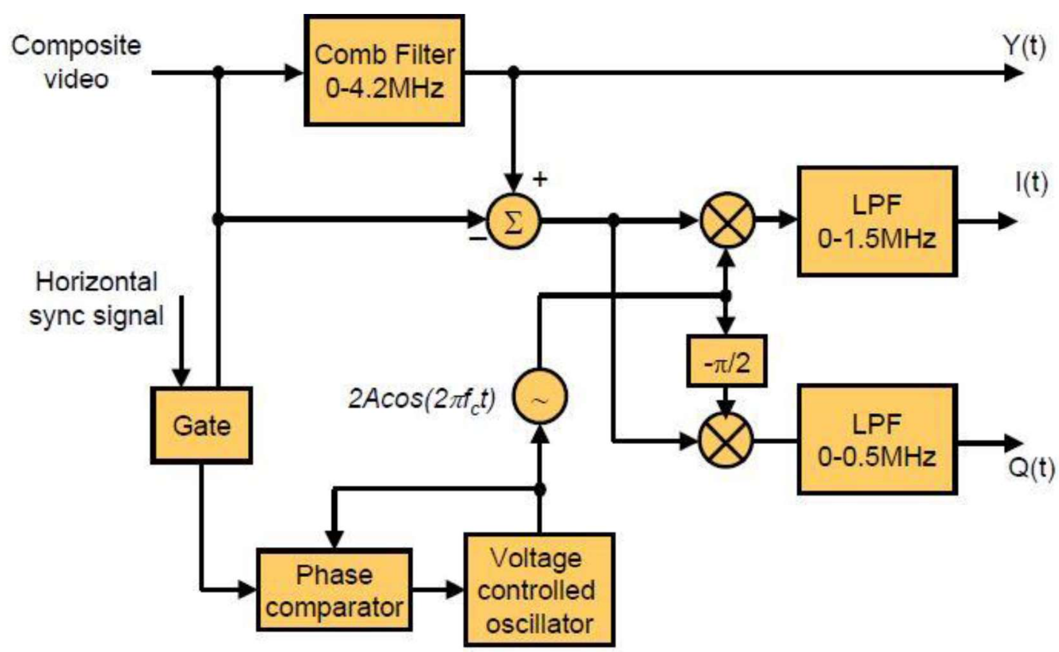


The Y-component has the maximum amount of data. The Y-component is passed through the Low Pass Filter with cut-off frequency of 4.2MHz. For multiplexing the I-component and Q-component of the signal, QAM is used. QAM is a method to modulate two signals onto the same carrier frequency, but with 90 degree phase shift. I-component and Q-component contain very less data. The I-component is passed through the Low Pass Filter with cut-off frequency of 1.5MHz and the Q-component is passed through the Low Pass Filter with cut-off frequency of 0.5MHz.

Following shows the block of QAM:



DEMULTIPLEXING BLOCK:



At the receiver end, we pass the received composite video through the Low Pass Filter instead of Comb Filter, with the upper band-limit of 4.2MHz. Using the adder-subtracter, we get signal with both I and Q components. By using Phase Lock Loop and a Low Pass Filter, with upper band-limit frequency of 1.5MHz and 0.5MHz, we get corresponding I and Q components.

RESULTS:

Properties of our video are as follows:

Command Window

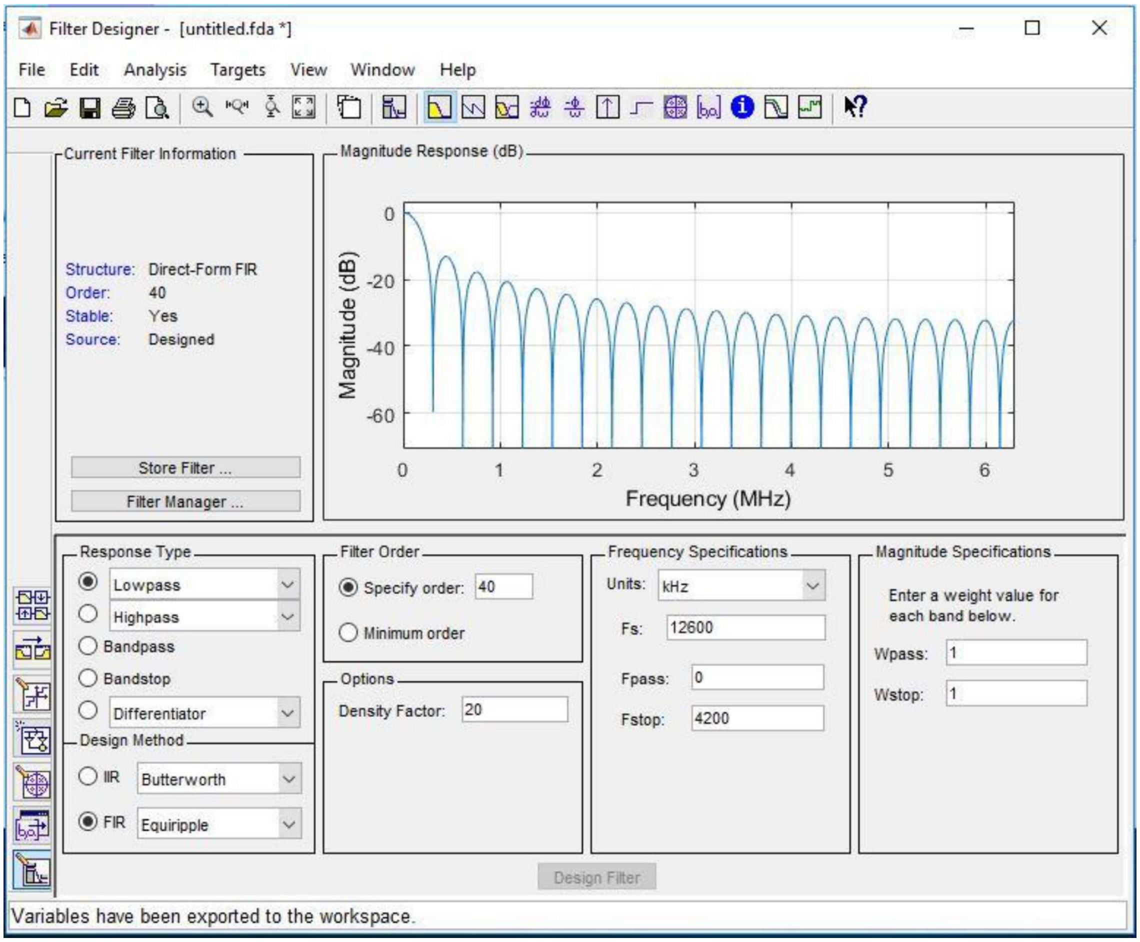
```
V =

  VideoReader with properties:

   General Properties:
            Name: 'vidi.mp4'
            Path: 'C:\Users\bhuvan\Desktop\Matlab'
        Duration: 2.6670
     CurrentTime: 2.6670
             Tag: ''
        UserData: []

   Video Properties:
           Width: 720
          Height: 480
       FrameRate: 29.5078
    BitsPerPixel: 24
     VideoFormat: 'RGB24'
```

The filter used in our project is as mentioned below and has the following specifications.



The MATLAB outputs are as follows:

1. Output of the frame after rgb2ntsc transformation.

YIQ Image

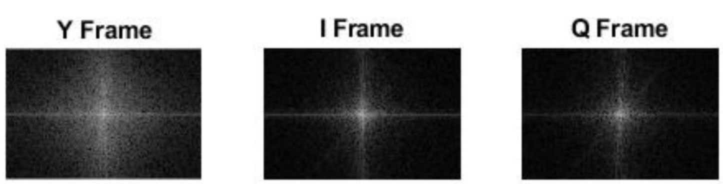
2. Original Y image

3. Original I image.

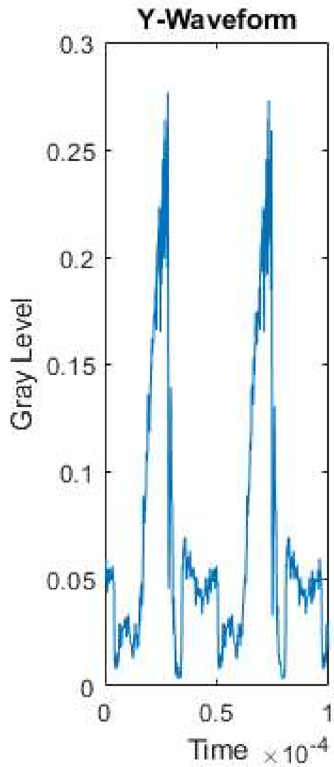
4. Original Q Image.

5. YIQ frames in spatial domain

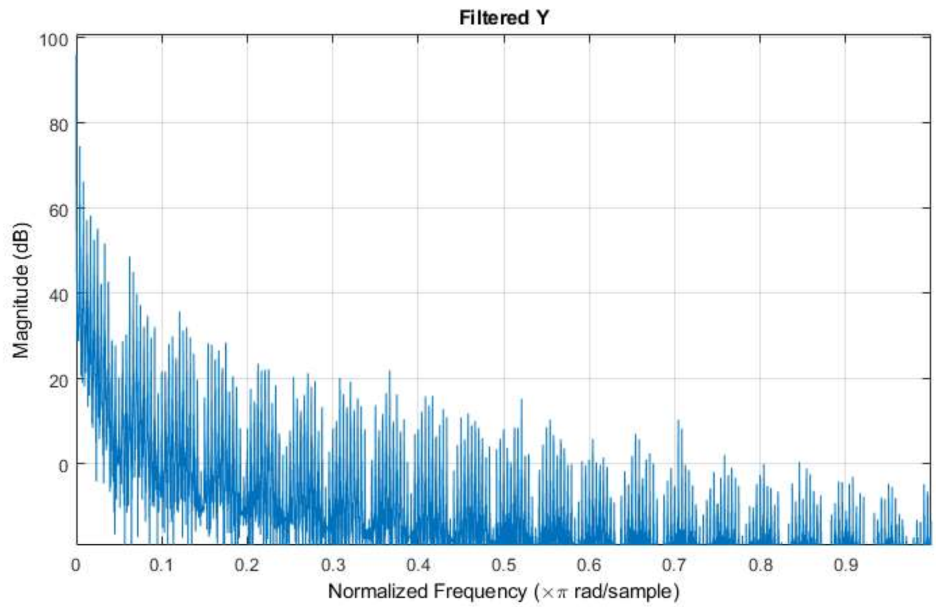
6. YIQ frames in frequency domain.



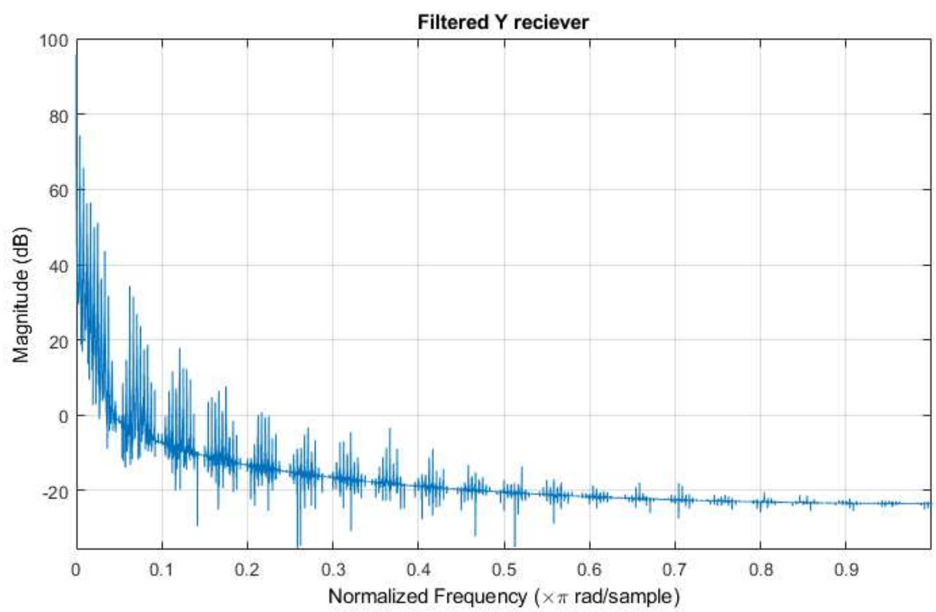
7. Y- Waveform representation as Raster vector



8. LPF Filtered Y waveform at the transmitter side.



9. LPF filtered Y waveform at the receiver side.



10. Recovered Y frame

APPENDIX:

MATLAB Code:

```
% Reading Video file
clc
clear all
V=VideoReader('vidi.mp4');
while hasFrame(V)
    video = readFrame(V);
end
whos video

display(V); %To Show properties of a Video file
implay('vidi')
YIQ=rgb2ntsc(video); %RGB TO YIQ
Y = YIQ(:,:,1); %luminance (intensity) Y
I = YIQ(:,:,2); %first color component I
Q = YIQ(:,:,3); %second color component Q
figure, imshow(Y);
figure, imshow(I);
figure, imshow(Q);
figure, imshow(YIQ);
title('YIQ Image');


% Fast Fourier transform of Y,I&Q
Y_freq = fft2(Y);
Y_freq = fftshift(Y_freq);
Y_freq = abs(Y_freq);
Y_freq = log(Y_freq+1);
Y_freq = mat2gray(Y_freq);

I_freq = fft2(I);
I_freq = fftshift(I_freq);
I_freq = abs(I_freq);
I_freq = log(I_freq+1);
I_freq = mat2gray(I_freq);

Q_freq = fft2(Q);
Q_freq = fftshift(Q_freq);
Q_freq = abs(Q_freq);
Q_freq = log(Q_freq+1);
Q_freq = mat2gray(Q_freq);

% Y,I,Q in spatial domain
figure;
subplot(1,3,1),imshow(Y);
title('Y Frame')
subplot(1,3,2),imshow(I);
title('I Frame')
subplot(1,3,3),imshow(Q);
title('Q Frame')
% Y,I,Q in in frequency domain
figure;
subplot(1,3,1),imshow(Y_freq,[]);
title('Y Frame');
subplot(1,3,2),imshow(I_freq,[]);
```

```
title('I Frame');
subplot(1,3,3),imshow(Q_freq,[]);
title('Q Frame');

%Conversion of frame data to raster
Y_raster=im2col(Y,[480 720],'distinct');
I_raster=im2col(I,[480 720],'distinct');
Q_raster=im2col(Q,[480 720],'distinct');

figure;
Fs=30*720*480; % Fs = sampling rate
subplot(1,3,1),plot([0:1/Fs:720*480/Fs-1/Fs],Y_raster(1:720*480));
ylabel('Gray Level');
xlabel('Time');
title('Y-Waveform') %Plotting Y Waveform
axis ([0,1E-4,0,0.3]);
% Filtering Y raster through Low pass filter
Filtered_Y=filter(Hdy,Y_raster);
fvtool(Filtered_Y);
title('Filtered Y')



%Reciever side

Filtered_Y_reciver=filter(Hdy,Filtered_Y);
fvtool(Filtered_Y_reciver);
title('Filtered Y reciever')
% YIQ recovered
YIQ_recovered=Filtered_Y_reciver+I_raster+Q_raster;
subplot(1,1,1),plot([0:1/Fs:720*1/Fs-1/Fs],YIQ_recovered(1:720*1));
ylabel('Gray Level');
xlabel('Time');
title('YIQ_recovered') %Plotting YIQ_recovered Waveform

% Raster to Image frames
YIQ_Frames=col2im(YIQ_recovered(1:480*720),[1 1],[480 720]);
imshow(YIQ_Frames);
y=YIQ_Frames(:,:,1);
i=YIQ_Frames(:,:,2);
q=YIQ_Frames(:,:,3);

%Converting Recovered YIQ to RGB
r_frame=y+0.956*i+0.620*q;
g_frame=y-0.272*i-0.647*q;
b_frame=y-1.108*i+1.7*q;

rgb(:,:,1)=r_frame;
rgb(:,:,2)=g_frame;
rgb(:,:,3)=b_frame;
%Recovered Final Video
implay(rgb);
```

BIBILOGRAPHY:

1. Feature Detectors and Motion Detection in Video Processing (Nilanjan Dey (Techno India College of Technology, Kolkata, India), Amira Ashour (Tanta University, Egypt) and Prasenjit Kr. Patra (Bengal College of Engineering and Technology, India))
2. en.wikipedia.org
3. Fundamentals of Multimedia (Ze-Nian Li, Mark S. Drew, Jiangchuan Liu)
4. Video Basics Slides by Yao Wang
5. MATLAB Mathworks Forum