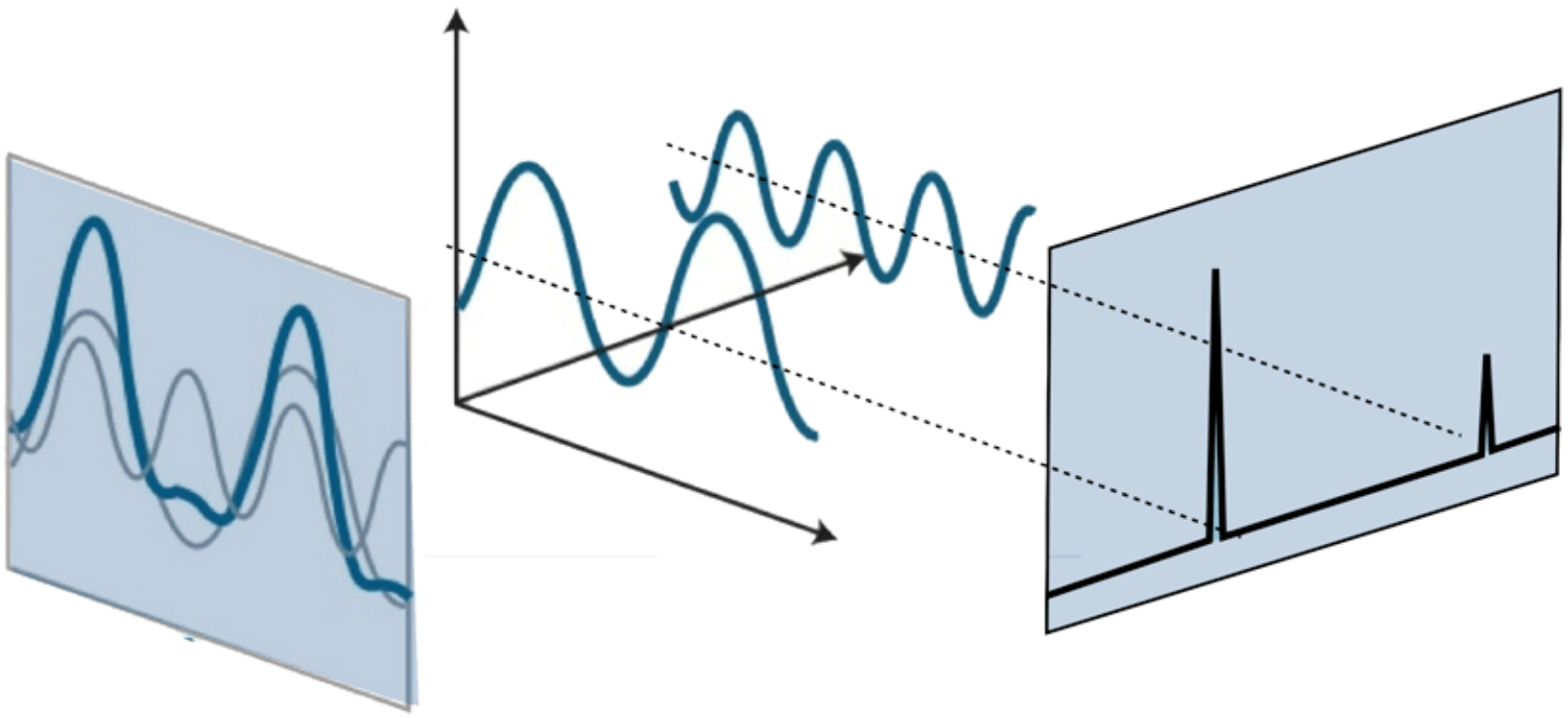
Multimedia-Lecture-Five

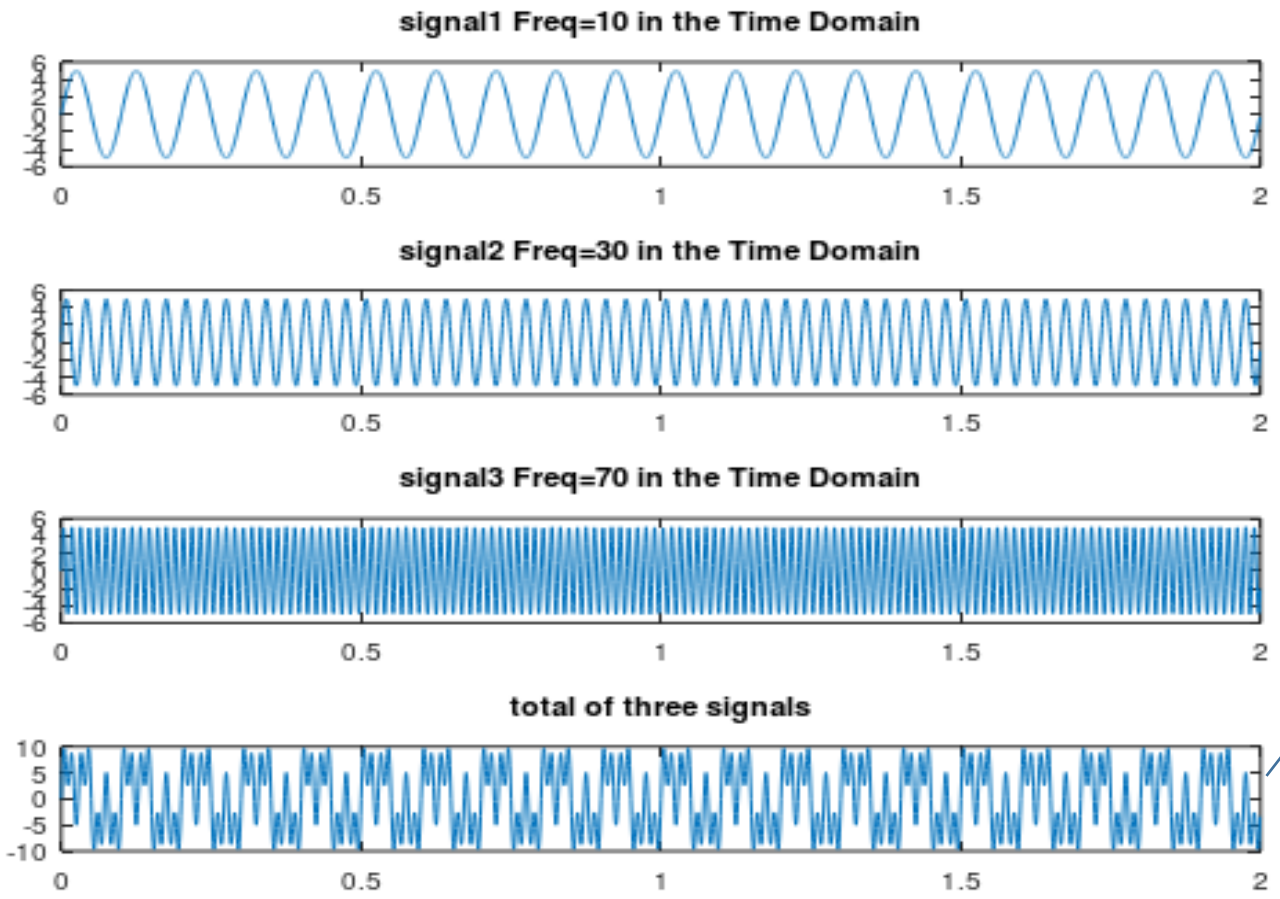
Fourier Transformations

Eng. Noor Alhakim

Fourier Transformations

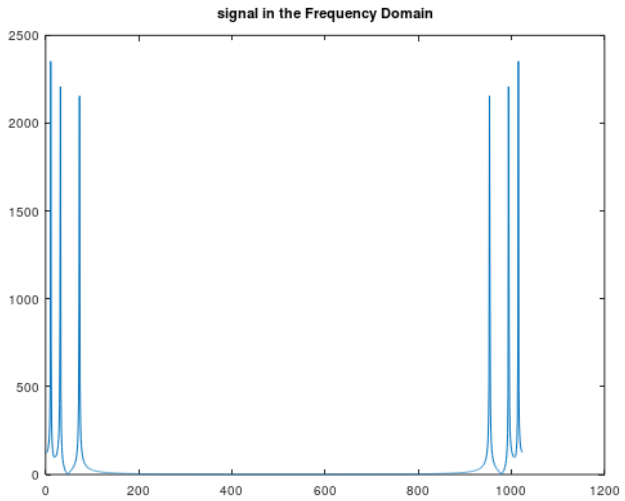


Define a Random Signal

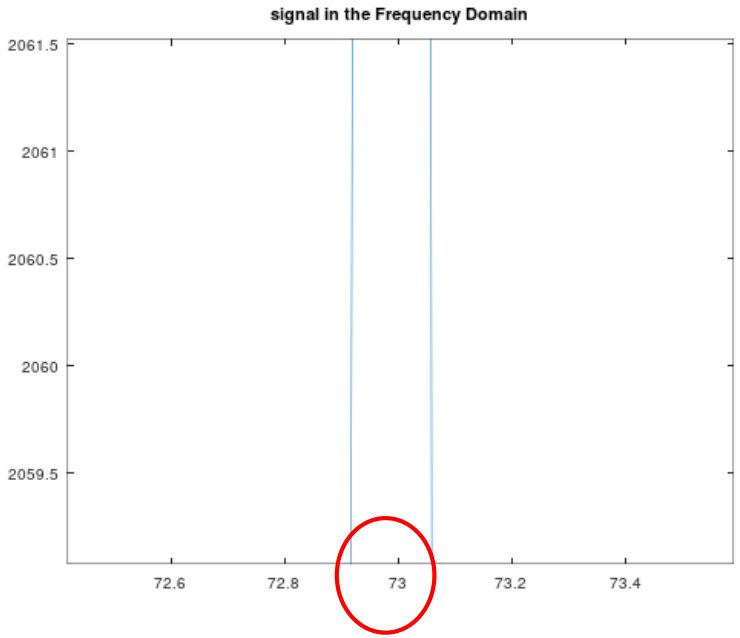


Sum of signal1, signal2, signal3

FFT on a Random Signal



Zoom in



The Frequencies of three signal in frequency domain same or too close from frequencies in time domain

Fast Fourier Transformations (FFT) on Signals using C#

- FFT on Audio Signal
- Plot FFt Audio Signal
- FFTshift on Audio Signal
- FFTshift on Image Signal
- Lowpass and Highpass Filter on Image.

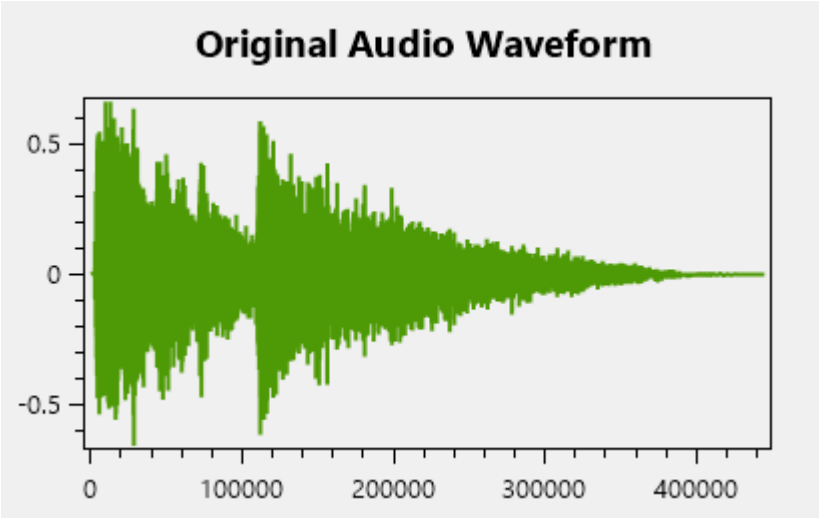
FFT on 1D signal

Audio

Plot the audio wave using OxyPlot

```
var plot = new PlotModel { Title = "Original Audio Waveform" };
var series = new LineSeries();
for (int i = 0; i < samples.Length; i++)
  {
    series.Points.Add(new DataPoint(i, samples[i]));
  }
  plot.Series.Add(series);

  Display the plot
  var plotView = new PlotView
  {
    Dock = DockStyle.Fill,
    Model = plot
  };
  var form = new Form();
  form.Controls.Add(plotView);
  Application.Run(form);
```

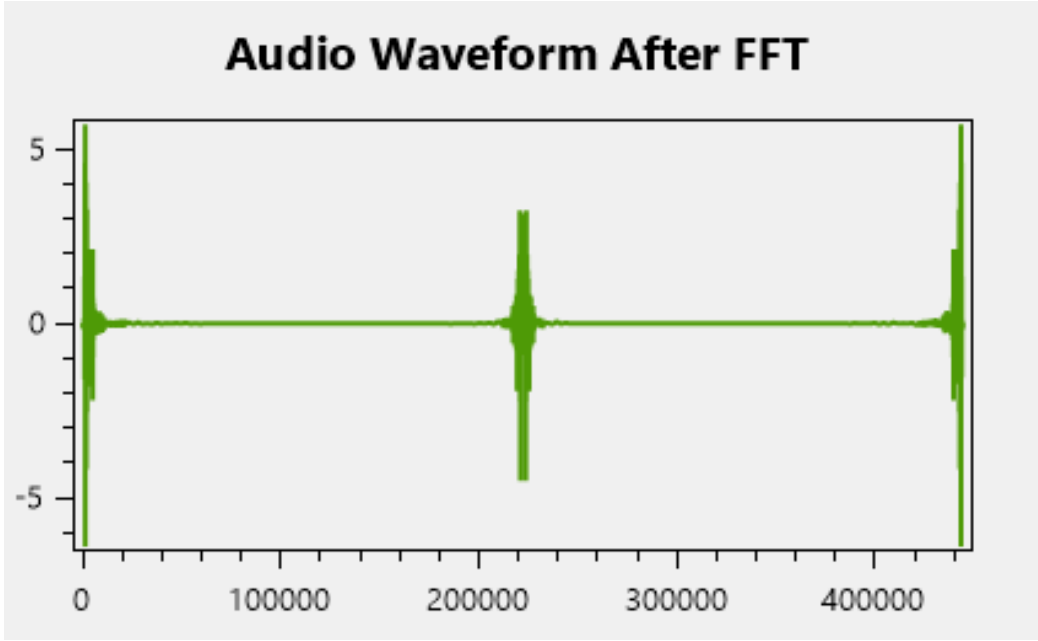


Read an audio file, then apply FFT function on audio signal

```
using MathNet.Numerics.IntegralTransforms;
using System.Numerics;
WaveFileReader reader = new WaveFileReader(audioFile2);

// Read audio data into buffer
byte[] buffer = new byte[reader.Length];
reader.Read(buffer, 0, (int)reader.Length);
// Convert bytes to float samples
float[] samples = new float[buffer.Length / 2];
for (int i = 0; i < buffer.Length / 2; i++)
{
  short sample = BitConverter.ToInt16(buffer, i * 2);
  samples[i] = sample / 32768f;
}
// Perform FFT
Complex[] fft = new Complex[samples.Length];
for (int i = 0; i < samples.Length; i++)
{
  fft[i] = new Complex(samples[i], 0);
}
Fourier.Forward(fft);
```

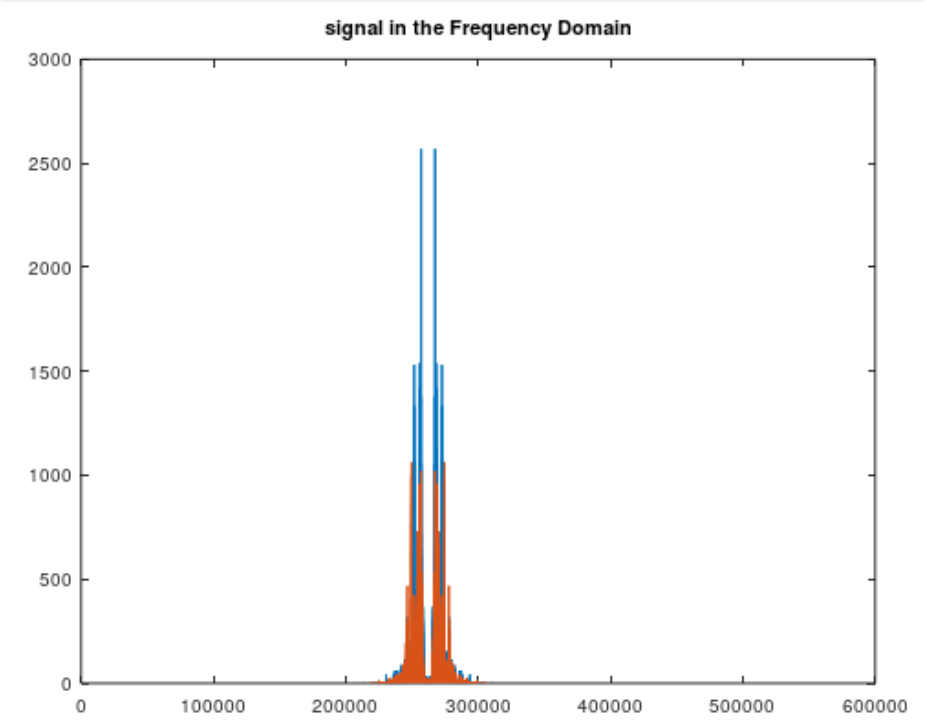
Try to plot the audio signal after FFT using OxyPlot



FFTshift on Audio Signal

FFTshift is a method to rearranges the outputs of FFT by moving the zero-frequency component to the center of the array.

Try to apply fftshift on the previous example audio signal and see the result, it should be similar to the side plot.



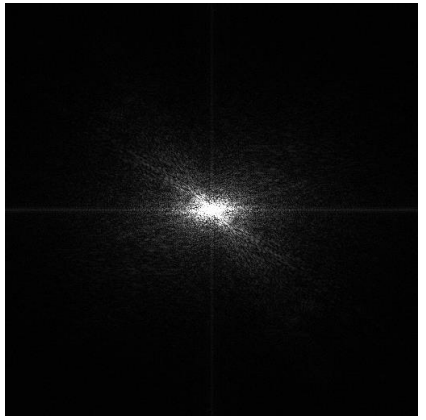
FFT on 2D signal

Image in gray scale

Read an image in gray scale, then apply FFT on image signal

```
using AForge.Imaging;
using AForge.Imaging.Filters;
// Load the image
Bitmap image = new Bitmap(imagePath);
// Convert the image to grayscale
Grayscale filter = new Grayscale(0.2125, 0.7154, 0.0721);
Bitmap grayImage = filter.Apply(image);
// Apply FFT to the grayscale image
ComplexImage complexImage = ComplexImage.FromBitmap(grayImage);
complexImage.ForwardFourierTransform();
// Compute the magnitude spectrum for visualization
Bitmap magnitudeImage = complexImage.ToBitmap();
// Display or save the magnitude spectrum
magnitudeImage.Save("magnitude_spectrum.jpg");
```

Orginal image



You should install these Libraries from NuGet:

AForge.Net

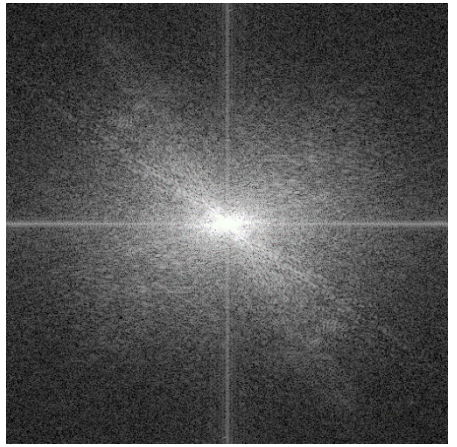
AForge.Imaging

AForge.Math

Read an image in gray scale, then apply FFT on image signal

```
using AForge.Imaging;
using AForge.Imaging.Filters;
// Load the image
Bitmap image = new Bitmap(imagePath);
Grayscale filter = new Grayscale(0.2125, 0.7154, 0.0721);
Bitmap grayImage = filter.Apply(image);
// Apply FFT to the grayscale image
ComplexImage complexImage = ComplexImage.FromBitmap(grayImage);
complexImage.ForwardFourierTransform();
// Compute the magnitude spectrum for visualization
Bitmap logMagnitudeImage = LogTransform(complexImage.ToBitmap());
// Display or save the magnitude spectrum
logMagnitudeImage.Save("log_magnitude_spectrum.jpg");
```

Orginal image



Try to write the LogTransform function to show the output of FFTshift image

Low/High Pass Filter on Image Using FFT

Low pass Filter (LPF):

removes high-frequency noise from a digital image and preserves low-frequency components.

Note: Low frequencies are represent smooth parts of the image

High pass Filter (HPF):

enhances the fine details and highlight the edges in a digital image.

Note: High frequencies are represent the rough parts (such as contours, lines and so on)

Exercise:

Write a C#-code to:

a. Apply Low pass filter using FFT and IFFT.
b. Show the images after applying the filters.

Tip:
Use AForge Library
IFFT: Inverse Fast Fourier Transformation

That’s All