Illinois Institute of Technology

Digital Signal Processing II

ECE 569

Project 2

Resampling and Auto regressive signal Modelling

Karthikeyan Kumar A20376688

Due April 28^{th} 2017

Contents

Objective

- Write a function to perform resample procedure at a rate of *I/D* on the given input signal to produce an output signal.
- Observe and comment on the difference in the given audio signal before and after resampling.
- Compare the variations between the resample and the downsample function.
- Write a function that gives the model coefficients output by performing estimation of the given input sequence by autoregressive model of a specific order by solving Yule-Walker equations.
- Use the coefficients to design a filter that is capable of whitening the input sequence.
- Employ the resampling function to resample the input image, and perform the same observations as before.
- Test the working of the autoregressive model function on the given input image.

Input Data

Sample2.wav and Sample2_tone.wav are two audio signals with 8 KHz frequency, lena.tif, and lena_tone.tif are the two black and white images given as input. Let sample2.wav be speech 1 and sample2_tone.wav be speech 2

Description

Part A

Resampling

Write a code to resample the input sequence at the rate of *I/D* and produce an output sequence.

The function takes an input sequence, resamples it in time-domain by performing decimation and interpolation by a factor of D and I respectively. The function takes a series of inputs and gives an output sequence such that the output sequence is interpolated and decimated by the certain factor given as input.

Outputsequence = Resample (inputsequence, I, D)

Inputsequence- a input sequence given by the user, may be a vector or matrix

I - Interpolation factor, usually an integer value

D- Decimation factor usually an integer value

Outputsequence- the output of the resample function, same dimension as the input sequence.

Note: resample applies an anti-aliasing (lowpass) FIR filter to inputsequence during the resampling process. It designs the filter using firls with a Kaiser window.

Result

The I and D values are chosen such a way that the output sequence $T_2 = 3.5T_1$ where T_1 = 0.125msec. For the speech 1, the input signal is quite clear with many samples, therefore upon listening it sounds very clear, and audible. The amplitude values are at peaks too, so it sound is easily perceived. But upon resampling, the number of samples has been reduced and the peak amplitude is also reduced, therefore the audio sounds as if the speaker is more distant from the user, lost its power. This is because of the presence of the anti-aliasing (lowpass) filter and the difference between the amplitude is visible in Fig 1 and Fig 2.

In the case of speech 2, which is already corrupted by a steady noise of high amplitude (seen from Fig. 3) though the speech is audible, it is overpowered by that high amplitude noise. Upon resampling, the output sequence shown in Fig 4 lacks the presence of the noise which has been sampled out. Therefore there is no audible presence of the noise, but the sound feels farther than the speech 1 output.

Comparison

Upon closer inspection at the resampled output of speech 1 and speech 2 actually vary in their amplitude value for each sample, the speech 2 is lesser in amplitude than the speech 1at all the samples, this is because of the filtering effect produced by the resample function.(firls filter)

Downsampling

Downsample the input sequence into an output sequence such that $T_2 = 3T_1$. The function takes an input sequence, and downsamples it by a factor and/or shifts it by an offset of samples in order to give the final output sequence.

Z= downsample (inputseq, factor, offset)

Z= output sequence same dimension as the input sequence but downsampled by a factor.

Factor- input integer that determines the downsample rate

Offset- the number of samples that needs to be shifted before starting to sample

Inputseq- the input sequence of a particular dimension that needs to be downsampled

Result

The downsampled version of the speech 1 is shown in Fig 5. Upon listen to the output, though it is audible, it is quite evident that there is a lack of continuity between the samples and the audio is coarse to listen. The output sequence has just missed some samples and there is no mechanism in order to smoothen out the lack of enough samples, and hence there is discontinuity in the audio

In the case of speech 2 output as displayed in Fig 6. It is quite evident that no additional processing has been performed on the signal, therefore the noise is still audible. It is also quite blatantly evident from the stem output. But there is a small decrease in the noise intensity, probably because of the lesser samples.

Comparison between resample and Downsample

Though resampling involves reduction in the sampling rate, due to the filtering effect that is performed by the firls filter, the audio sounds clear, and good. But there is no such filtering effect in the case of downsample and therefore the audio sounds worse than the original signal. This goes to say there is some form of filtering mechanism necessary in order to compensate the loss of samples produced by downsampling. In case of resampling, the output is actually scaled in the amplitude values for each sample, as seen in Fig. 2. But for downsampling seen in Fig. 5, the function just drops down samples, there is no smoothing effect, which is a major difference between the two functions.

Part B

Yule-Walker

The function takes in an input sequence and the order necessary and gives out the coefficients of the autoregressive model. This coefficients are taken as the coefficients for the filter in order to process the input sequence, and give out output.

A= aryule (x, p)

A – Output coefficients corresponding to the input x. its dimension depends on the input x.

x- Input sequence, it maybe a row vector or a matrix

p – Order of the AR model. (Should be less than the rows of input sequence).

Result

First off the deafening tone is completely gone after filtering the input signal. It is noticed in Fig 7. That the output audio has been scaled drastically down in amplitude in all the samples, it is also seen that there are certain high peaks equal to the original signal near the time t=0. But this does not affect the output quality by a great deal, but there is an obvious drop in the amplitude overall and also the removal of noise. This shows how the algorithm works by factoring the correlation between the sequences and filtering the noise rather than the audio. The AR model computes the PSD of the signal, and computes the coefficients to perform the least square estimation of the signal and in this process the noise factor that is introduced is completely eliminated.

Part C

Resampling

We apply the same resampling function from the Part A to the images given. But as the image is a 2D entity, it is necessary to use the resampling function twice, once for the rows, another for the columns of the image. The conversion rate is 3.5, similar to the case of Part A

Result

The original image without tone is shown in Fig 8. And the resampled one in Fig 9. As an obvious result of resample, the image is reduced in its dimensions, but it still retains its quality pertaining to the new dimension of the image. The process of resampling coupled with the smoothening effect produced by the filter, the image retained if not slightly improved in its quality with respect to its dimension.

The original image without tone is shown in Fig 10, the result of resampling the image with tone is shown in Fig 11. It is evident that the resampling procedure has reduced the strips in the image with the tone, but it also reduces the number of samples so the dimensions of the image is reduced, nevertheless the quality with respect to its dimension is still maintained.

Downsampling

We apply the same downsampling function from the Part A to the images given. But as the image is a 2D entity, it is necessary to use the resampling function twice, once for the rows, another for the columns of the image. We desire a conversion rate of 3 here.

Result

It is quite evident that the downsampled function produces an output image that is of lesser quality than the original input signal. The pixels seem to be a bit sharper and the transition between the pixels are abrupt. This is due to the absence of the filtering effect. It is quite evident from the Fig.12 and Fig 13.

Comparison of Resample and Downsample

The resample and downsample functions produce the output image with different dimension, due to the filtering capability of the resampling filter the resampled image is of better quality as seen in Fig.9. As there is no such procedure in the downsampling, the transition from one pixel to another is very sharp, hence gives a pixelated image seen in Fig. 12. If we plot the histogram we can observe the difference.

Appendix

Note: In order to play the audio file using sound function, it is necessary to use proper sampling rate. In case of $T_2=3.5T_1$. The sampling rate is 2.278 KHz and for $T_2=3T_1$. It is 2.667 KHz

Functions used

Z= Wavread-('filename')

Used to get an audio file as input into MATLAB's work space.

'filename' refers to the path of the audio file.

z is the output variable for the input audio file

Stem(variable')

Helps in plotting data as discrete values.

Y=Filter(b,a,x)

Filters the input data according to the transfer function represented by **b and a.**

Y – the output variable, equal in dimension to the input variable.

X – input sequence, it maybe a vector or a matrix

b,a - denominator and numerator of the transfer function.

Z= Imread('filename')

Used to read the image into the MATLAB workspace, and can be accessed by the variable Z

Imshow('variable')

Used to display the MATLAB workspace variable as an image.

Program and Figure

Contents

Part A T2=3.5T1

```
%get the inputs
inputsignal=wavread('F:\Coll works\HOMEWORKS\Spring 17\DSP 2\Project 2\sample2.wav');
inputsignaltone=wavread('F:\Coll works\HOMEWORKS\Spring 17\DSP 2\Project 2\sample2_tone.wav');
%sound(inputsignal);
%f1 is 8khz f2 is 2.2857khz I=2 D=7
I=input('Enter I');
D=input('Enter D');
outputsignal=resample(inputsignal,I,D); %resampling of the signal
outputsignaltone=resample(inputsignaltone,I,D); %resampling of the signal
%use sampling rate of 8kHz to play input signal, 2.28kHz to play output
%signal
figure();
stem(inputsignal);
title('Input signal without Tone');
ylabel('Amplitude');
xlabel('time scale');
figure();
stem(outputsignal);
title('Resampled Output signal without Tone');
ylabel('Amplitude');
xlabel('time scale');
figure();
stem(inputsignaltone);
title('Input signal with Tone');
ylabel('Amplitude');
xlabel('time scale');
figure();
stem(outputsignaltone);
title('Resampled Output signal with Tone');
ylabel('Amplitude');
xlabel('time scale');
```

Part A T2=3T1

```
%input the signals
inputsignal=wavread('F:\Coll works\HOMEWORKS\Spring 17\DSP 2\Project 2\sample2.wav');
inputsignaltone=wavread('F:\Coll works\HOMEWORKS\Spring 17\DSP 2\Project 2\sample2_tone.wav');
incoutputsignal=downsample(inputsignal,3,2);
incoutputsignaltone=downsample(inputsignaltone,3,2);
%Use sample rate 2.667KHz to play the signal
figure();
stem(incoutputsignal);
title('Downsampled output signal');
ylabel('Amplitude');
xlabel('time scale');
figure();
stem(incoutputsignaltone);
title('Downsampled Output signal with Tone');
ylabel('Amplitude');
xlabel('time scale');
```

Contents

Part C

```
%Read the input image
inputimagetone=imread('F:\Coll works\HOMEWORKS\Spring 17\DSP 2\Project 2\lena_bw_tone.tif');
inputimagetone=im2double(inputimagetone); %convert image's data type to double to perform operations on it
inputimage=imread('F:\Coll works\HOMEWORKS\Spring 17\DSP 2\Project 2\lena_bw.tif');
inputimage=im2double(inputimage); %convert image's data type to double to perform operations on it
I=input('Enter I');
D=input('Enter D');
outputimagetone=resample(inputimagetone,I,D); %Resample in one dimension (rows)
outputimagetone=resample(outputimagetone',I,D); % Resample in next dimension (cols)
outputimage=resample(inputimage,I,D);  %Resample in one dimension (rows)
outputimage=resample(outputimage',I,D); % Resample in next dimension (cols)
figure();
imshow(inputimage);
title('Input image without Tone');
ylabel('rows');
xlabel('cols');
figure();
imshow(outputimage');
title('Resampled Output image without Tone');
ylabel('rows');
xlabel('cols');
figure();
imshow(inputimagetone);
title('Input image with Tone');
ylabel('rows');
xlabel('cols');
figure();
imshow(outputimagetone');
title('resampled output image with Tone');
ylabel('rows');
xlabel('cols');
```

Part C downsample

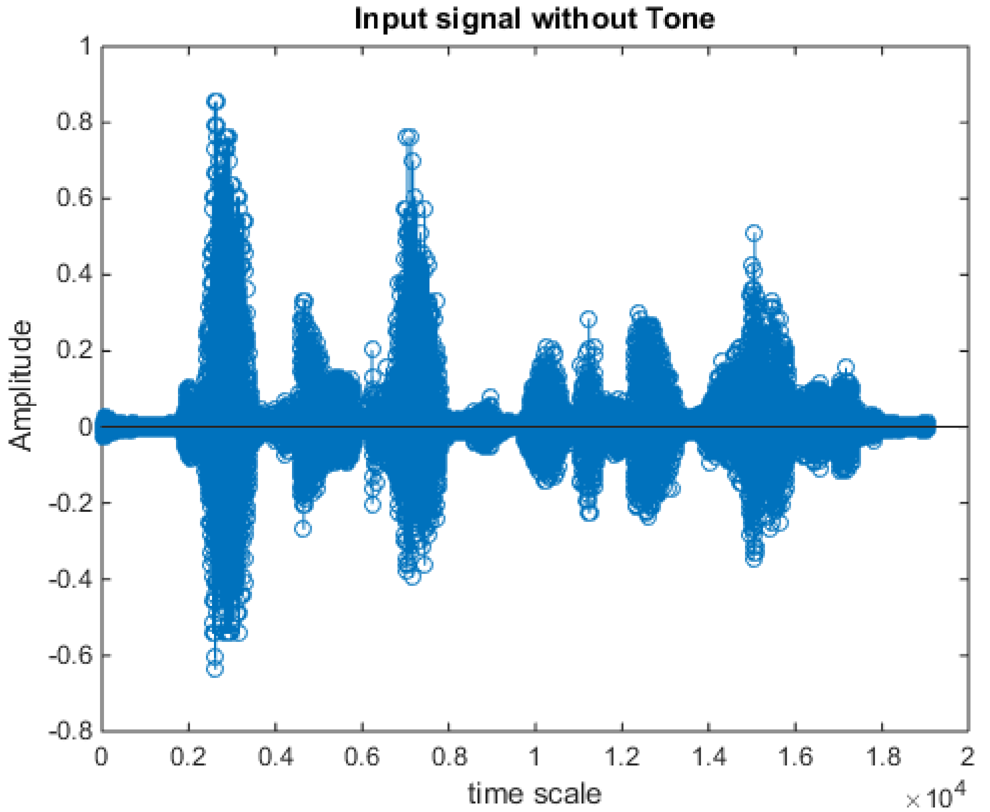
```
incoutputimagetone=downsample(inputimagetone,3,2); %Downsampling the image on both dimensions
incoutputimagetone=downsample(incoutputimagetone',3,2);
incoutputimage=downsample(inputimage,3,2);
incoutputimage=downsample(incoutputimage',3,2);
figure();
imshow(incoutputimage');
title('downsampled output image without Tone');
ylabel('rows');
xlabel('cols');
figure();
imshow(incoutputimagetone');
title('downsampled output image with Tone');
ylabel('rows');
xlabel('cols');
```

Published with MATLAB® R2014b

Contents

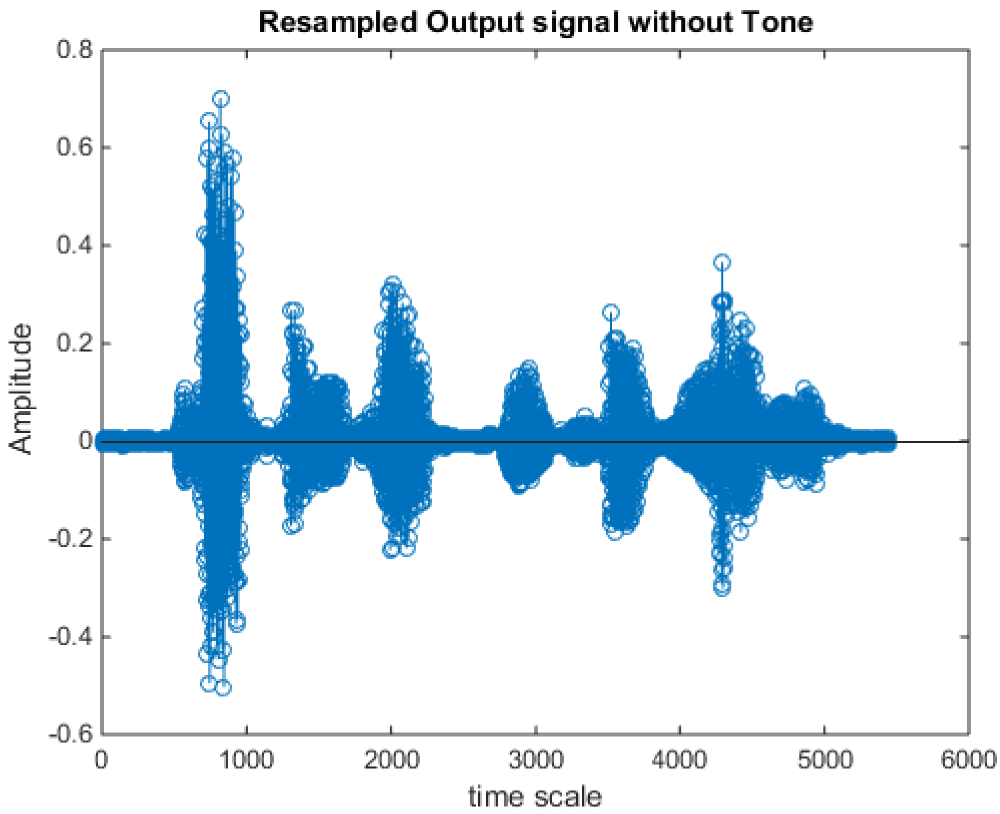
Part A T2=3.5T1

```
Warning: WAVREAD will be removed in a future release. Use AUDIOREAD instead.
Warning: WAVREAD will be removed in a future release. Use AUDIOREAD instead.
```



Original signal

Figure 1



resampled to the time 3.5*T1

Figure 2

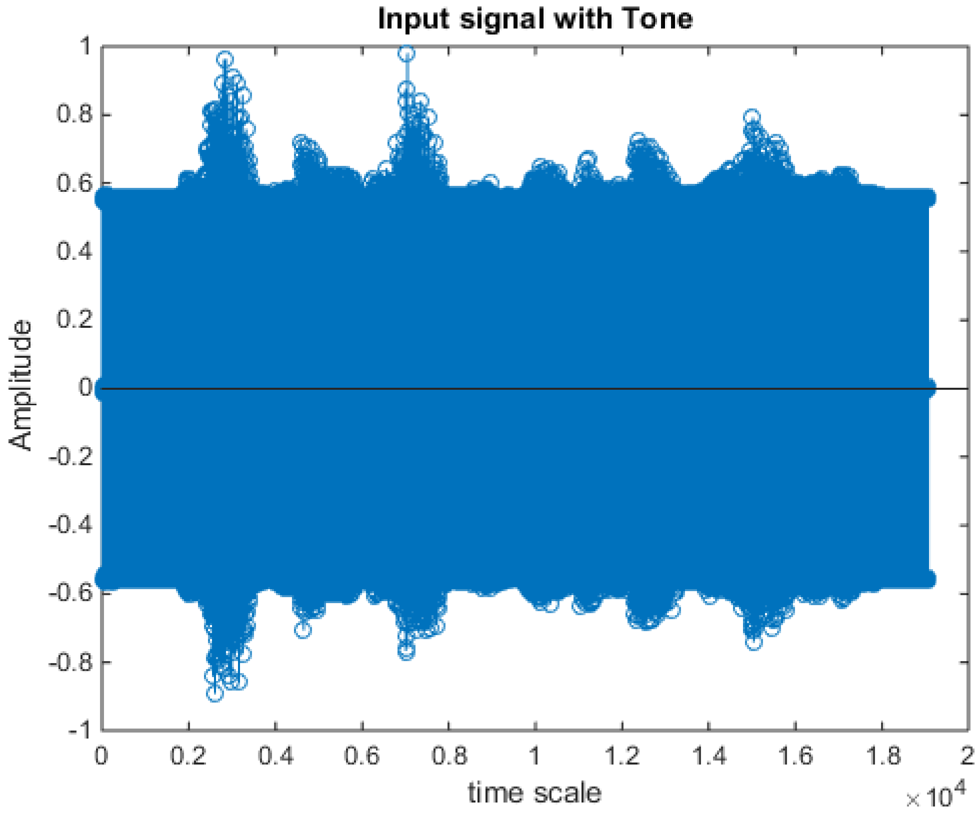


Figure 3

Input signal with noise

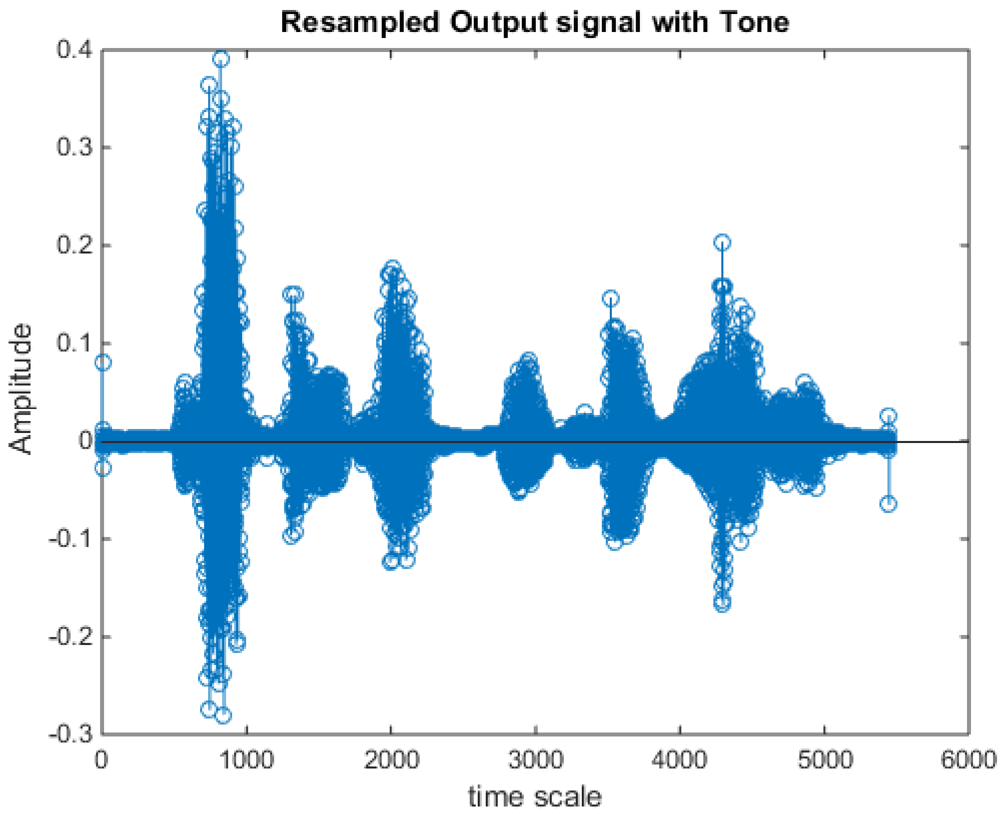


Figure 4

Complete absence of noise

Part A T2=3T1

```
Warning: WAVREAD will be removed in a future release. Use AUDIOREAD instead.
Warning: WAVREAD will be removed in a future release. Use AUDIOREAD instead.
```

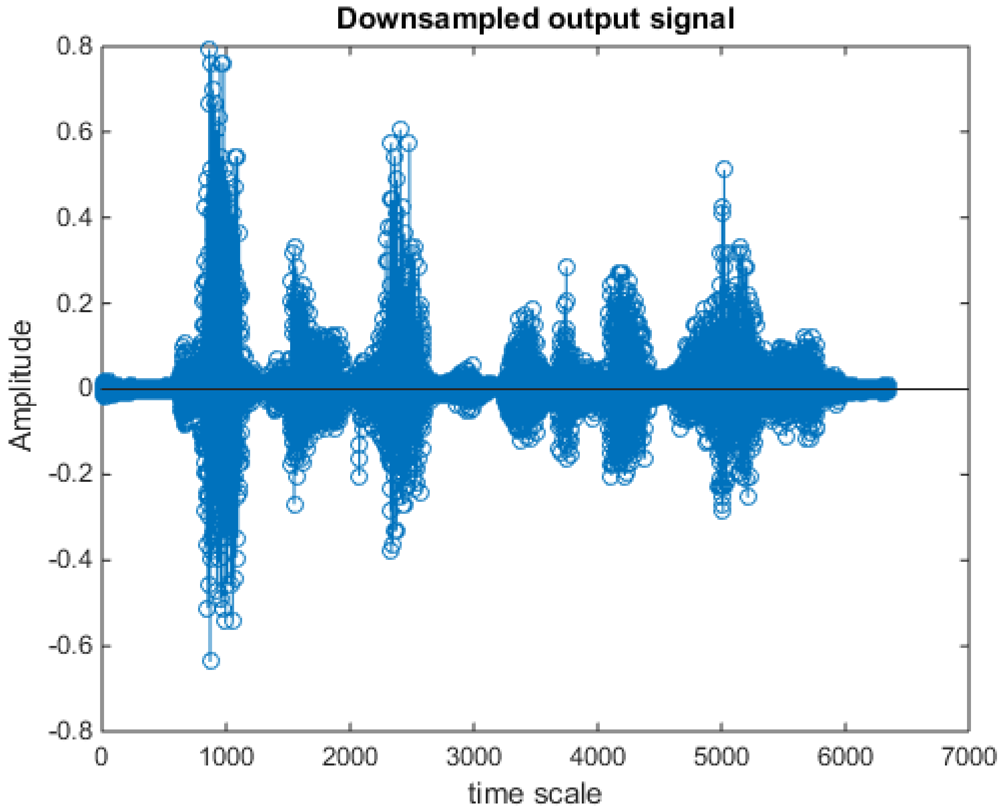


Figure 5

Transitions are too sudden and sharp.

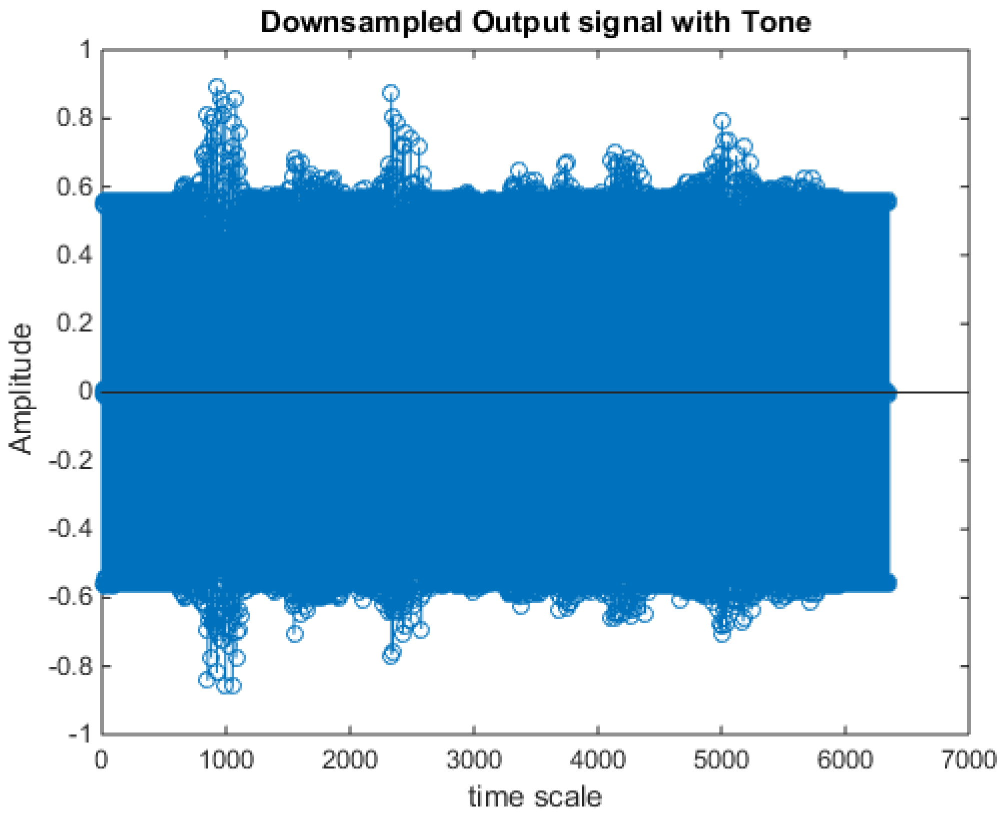


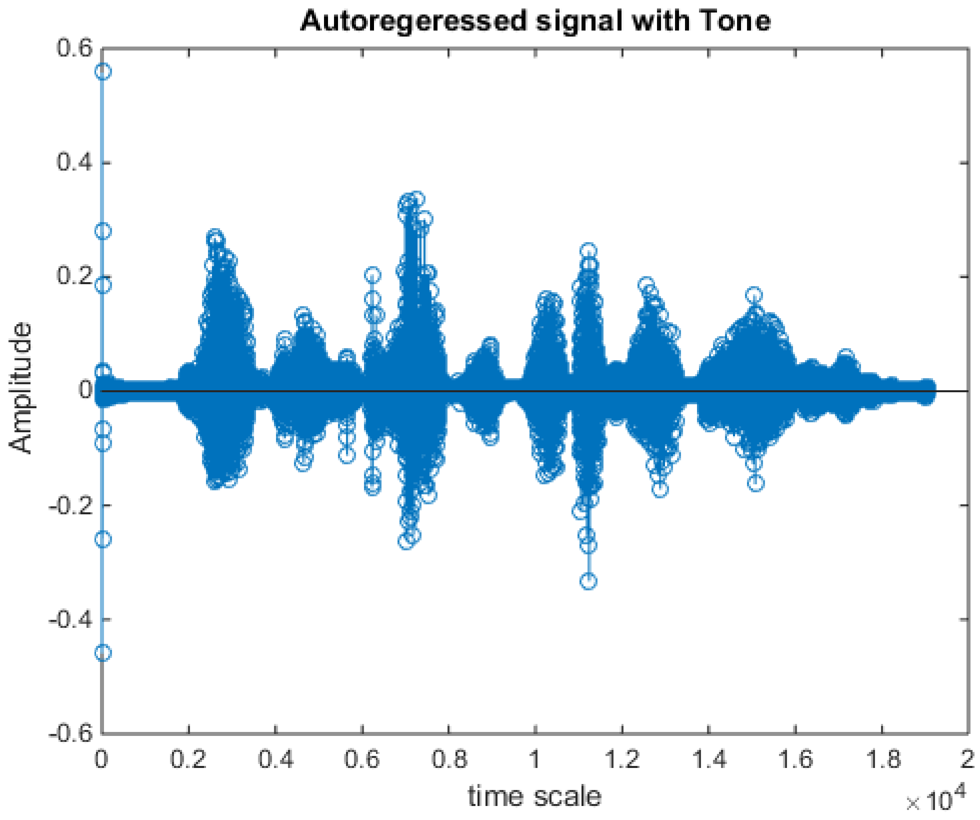
Figure 6

No reduction in the noise presence

Part B

```
inputsignal=wavread('F:\Coll works\HOMEWORKS\Spring 17\DSP 2\Project 2\sample2_tone.wav');
%p=input('Enter the Order p'); %say 10
p=10;
a=aryule(inputsignal,p); %% Coefficients of the filter found using Yule-walker method
zz=filter(a,1,inputsignal); %%whitening filter using coefficients
figure();
stem(zz);
title('Autoregeressed signal with Tone');
ylabel('Amplitude');
xlabel('time scale');
```

```
Warning: WAVREAD will be removed in a future release. Use AUDIOREAD instead.
```



After the auto regression model and filtering(whitening)

Figure 7

Published with MATLAB® R2014b

Contents

Part C

Input image without Tone

Figure 8. Original input image

Resampled Output image without Tone

Figure 9

Resampled image, dimension is reduced due to reduction in sample

Input image with Tone

Figure 10. Original image with tone

resampled output image with Tone

Figure 11

The tones are missing, owing to resampling and aliasing filter

Part C downsample

downsampled output image without Tone

Downsampled image, with lesser samples, looks slightly pixelated

Figure 12

downsampled output image with Tone

Figure 13