2EC502: DIGITAL SIGNAL PROCESSING COMPREHENSIVE ASSIGNMENT - REPORT

Musical Sound Processing: Flanging & Chorus Generation

SUBMITTED BY,
VINCENT ABRAHAM – 18BEC122

DEPARTMENT OF ELECTRONICS

& COMMUNICATION ENGINEERING,

INSTITUTE OF TECHNOLOGY,

NIRMA UNIVERSITY,

AHMEDABAD, INDIA

OBJECTIVE

Audio effects are some of the very important tools in sound design and musical sound processing. Effects like echo, reverberation, flanging, vibrato, chorus are some of the basic effects used in movies, television shows, music recordings, video games etc. in order to enhance or emphasize the original audio in the desired manner. These effects can easily be achieved using digital signal processing. In this report, two such effects, namely the flanging and the chorus effect would be explored theoretically and experimentally. After discussing the necessary theory involving the physical phenomena and the underlying algorithms for both the effects, the experimental analysis would be carried out using MATLAB.

FIR COMB FILTER

Before moving towards the generation of flanging and chorus effects, it's important to understand the working of an FIR Comb filter, as both the effects would be using this filter in order to get the desired output audio signal.

An FIR Comb filter is used to delay the given input signal by a fixed amount of time. Using this filter, one could add a single delay to the input signal, and on adding the delayed and the original signal, its effect would be observed. This filter has two control parameters, one being the time duration (τ) and the second being the modified amplitude of the delayed signal (g).

$$y(n) = x(n) + g \cdot x(n - M)$$

$$where, \qquad M = \frac{\tau}{F_S}$$

Where,

$x(n)$ *is the input signal,*

$y(n)$ *is the output signal,*

F_S *is the sampling frequency of the input.*

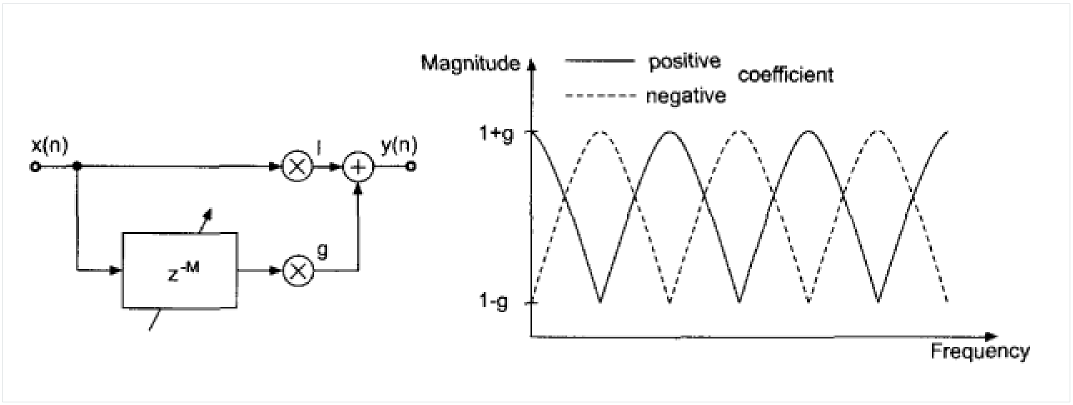


Figure 1: Block Diagram & Magnitude Response of the FIR Comb Filter [1].

As shown in the above figure, it can be observed that for positive values of (g), the magnitude response shows the amplification of the frequencies which are multiples of $1/\tau$, and attenuation of the frequencies that lie in between. The exact opposite is observed for negative values of (g). The gain of this filter ranges from $1 + g$ to $1 - g$.

Flanging and chorus, both being delay based effects, by providing different values to the parameters τ and g, these effects would be obtained.

FLANGING

Flanging is the whooshing sound effect that can be observed in some music. It isn't generally applied to speech signals. This effect can be achieved by delaying the input signal by a very small duration through the FIR comb filter. Along with this, an LFO (Low Frequency Oscillator) is used to decide the proportion by which the input signal is delayed. The LFO being a sinusoidal signal, its amplitude at any point of time decides the proportion of delay.

There are four parameters through which, different variations in the flanging effect can be observed. First is the amount of time by which the input is delayed. This should generally be in the range of 1-10 milliseconds, as any value above that won't be able to give the proper effect. The second is the depth which is controlled by the amplitude of the sinusoidal signal (LFO). The third is the rate, which is the frequency of the LFO, this can be varied to control the speed of the flanging effect and the fourth is the modified amplitude (g) which is used to control the amount of delayed signal which is added to the original signal. A higher value of g would give a more prominent effect.

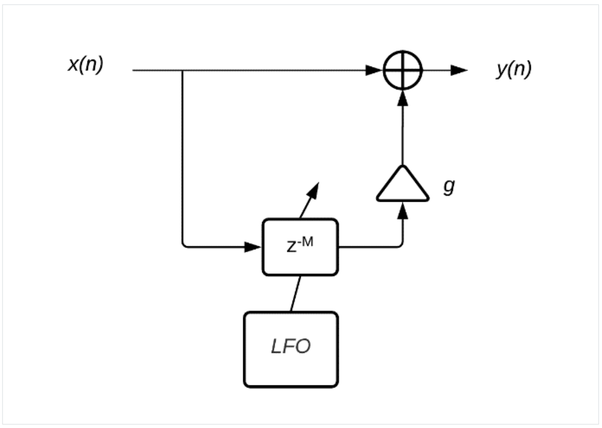


Figure 2: Block Diagram for the Flanger.

The equation for a flanger is as shown below,

$$y(n) = x(n) + g \cdot x(n - M)$$

Where,

M is the product of the amplitude of sinusoidal signal, with the amount of delay required.

CHORUS

The chorus effect is one of the most widely used audio effects and is used on vocals, as well as on instruments. It emulates the effect of multiple vocals being stacked on top of each other, or multiple instruments playing at a time, having some slight delay in between them.

The method to create the chorus effect can be considered as a slight variation in the process of creating a flanging effect where here, instead of one delay line two or more delay lines would be present and the input signal would be delayed based on the time duration and the modified amplitudes set in each delay line. Another main difference is that, the amount of time by which the input is delayed which was kept earlier between 1-3 milliseconds should here be increased to 30-50 milliseconds in order to get the adequate chorus effect. The rate here, is also kept relatively lower so that the slight flanging wouldn't be heard in the audio signal.

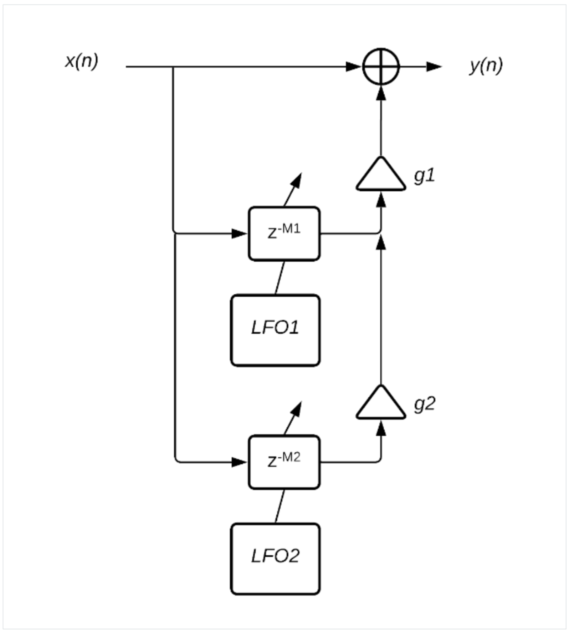


Figure 3: Block Diagram for the Chorus Effect.

The equation for the chorus effect is as shown below,

$$y(n) = x(n) + g_1 \cdot x(n - M_1) + g_2 \cdot x(n - M_2)$$

DESIGN PROBLEM

1. Write a MATLAB code in order to implement and analyze the effect of flanging on an audio signal. Plot the input and output signals in time domain. Also, plot the spectrograms of both the signals.
2. Write a MATLAB code in order to implement and analyze the chorus effect on an audio signal. Plot the input and output signals in time domain. Also, plot the spectrograms of both the signals. Compare the results with the flanging effect and draw the necessary conclusions.

Both these problems can easily be done using the theory discussed earlier. Shown below are the various parameters which are going to be taken for both these problems,

Flanging

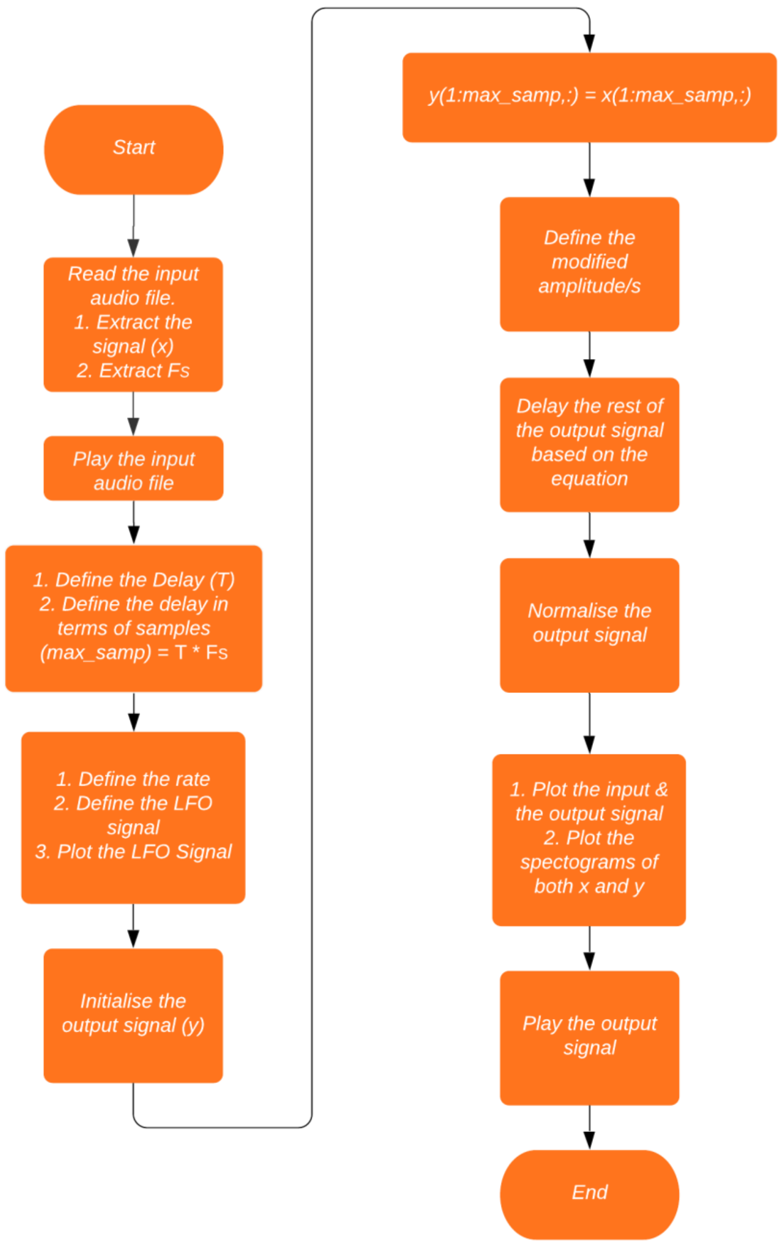
Parameters	Values
Rate	1 Hz
Delay (τ)	1 ms
Modified Amplitude (g)	0.7

Chorus

Parameters	Values
Rate	0.3 Hz
Delay (τ)	50 ms
Modified Amplitude (g_1)	0.7
Modified Amplitude (g_2)	0.5

FLOWCHART

The process of implementing both, the flanging and chorus effects would be fairly the same, the only difference would be the change in some of the parameters and the output signal, based on the equations.


Start
Read the input audio file.
1. Extract the signal (x)
2. Extract Fs
Play the input audio file
1. Define the Delay (T)
2. Define the delay in terms of samples (max_samp) = T * Fs
1. Define the rate
2. Define the LFO signal
3. Plot the LFO Signal
Initialise the output signal (y)
y(1:max_samp,:) = x(1:max_samp,:)
Define the modified amplitude/s
Delay the rest of the output signal based on the equation
Normalise the output signal
1. Plot the input & the output signal
2. Plot the spectograms of both x and y
Play the output signal
End

MATLAB CODES

MATLAB CODE FOR FLANGING EFFECT

```
% Code for Flanging Generation
% With the help of LFO, input audio signal is delayed by either 0-3, 0-10,
0-15 ms

clc;
clear all;
close all;

% Defining the input and output files for reading and writing the audio
input_file = 'drums.wav';
output_file = 'flanger_output.wav';

% Reading the input audio file
[x, FS] = audioread(input_file);
sound(x, FS);
pause(6)

% Defining the maximum delay in time and samples for the flanger
max_time = 0.001;                   % 1 ms
max_samp = round(max_time * FS);    % For example, 1ms * 11KHz = 11 samples

rate = 1;                           % Frequency of the LFO, 1 Hz

t = 1:length(x);                    % Discrete time for the sine signal
sig = (sin(2*pi*(rate/FS)*t))';     % Defining the LFO signal

subplot(212)
plot(t, sig)
axis([0 2.73e5 -1 1])
xlabel('Discrete Time')
ylabel('Amplitude')
title('LFO Sinusoidal Signal')

% Defining the output signal
y = zeros(size(x));

y(1:max_samp,:)=x(1:max_samp,:);
mod_amp = 0.7;                     % Modified Amplitude of the signal

for i = max_samp + 1 : length(x)
    delay = ceil(abs(sig(i)) * max_samp);
    y(i,:) = x(i,:) + mod_amp * (x(i - delay,:));    % Delayed Output
end

% Normalising the Output Signal
y = y./max(abs(y));

% Plotting the Input and Output Signals in Time Domain
subplot(211)
plot(t, x(1:length(x)))
hold on;
```

```
plot(t, y(1:length(y)))
hold off;
axis([0 2.73e5 -1 1])
xlabel('Discrete Time')
ylabel('Amplitude')
title('Original & the Output Signal')
legend('Original', 'Output')

% Plotting the Spectograms of the Input and Output Signals
figure(2);
subplot(211)
spectrogram(x(1:length(x)), 1024, 512, 1024, FS, 'yaxis')
title('Spectogram of the Original Signal')
subplot(212)
spectrogram(y(1:length(y)), 1024, 512, 1024, FS, 'yaxis')
title('Spectogram of the Output Signal')

sound(y, FS);
pause(6);
audiowrite(output_file, y, FS);
```

MATLAB CODE FOR THE CHORUS EFFECT

```
% Code for Generating a Chorus Effect
% With the help of LFO, input audio signal is delayed by 30-50 ms

clc;
clear all;
close all;

% Defining the input and output files for reading and writing the audio
input_file = 'drums.wav';
output_file = 'chorus_output.wav';

% Reading the input audio file
[x, FS] = audioread(input_file);
sound(x, FS);
pause(6)

% Defining the maximum delay in time and samples for the chorus effect
max_time = 0.05;                  % 50 ms
max_samp = round(max_time * FS);  % For example, 50ms * 11KHz = 550 samples

rate = 0.3;                       % Frequency of the LFO, 0.3 Hz

t = 1:length(x);                  % Discrete time for the sine signal
sig = (sin(2*pi*(rate/FS)*t))';   % Defining the LFO signal

subplot(212)
plot(t, sig)
axis([0 2.73e5 -1 1])
xlabel('Discrete Time')
ylabel('Amplitude')
title('LFO Sinusoidal Signal')

% Defining the output signal
y = zeros(size(x));
```

```
y(1:max_samp,:) = x(1:max_samp,:);
mod_amp1 = 0.7;                  % Defining the modified amplitudes
mod_amp2 = 0.5;



for i = max_samp + 1 : length(x)
    delay = ceil(abs(sig(i)) * max_samp);
    y(i,:) = x(i,:) + mod_amp1*(x(i - delay, :)) + mod_amp2*(x(i - delay,
:));
end

% Normalising the Output Signal
y = y./max(abs(y));

% Plotting the Input and Output Signals in Time Domain
subplot(211)
plot(t, x(1:length(x)))
hold on;
plot(t, y(1:length(y)))
hold off;
axis([0 2.73e5 -1 1])
xlabel('Discrete Time')
ylabel('Amplitude')
title('Original & the Output Signal')
legend('Original', 'Output')

% Plotting the Spectograms of the Input and Output Signals
figure(2);
subplot(211)
spectrogram(x(1:length(x)), 1024, 512, 1024, FS, 'yaxis')
title('Spectogram of the Original Signal')
subplot(212)
spectrogram(y(1:length(y)), 1024, 512, 1024, FS, 'yaxis')
title('Spectogram of the Output Signal')

sound(y, FS);
pause(6);
audiowrite(output_file, y, FS);
```

RESULTS & OBSERVATIONS

RESULTS OF THE CODE FOR FLANGING EFFECT

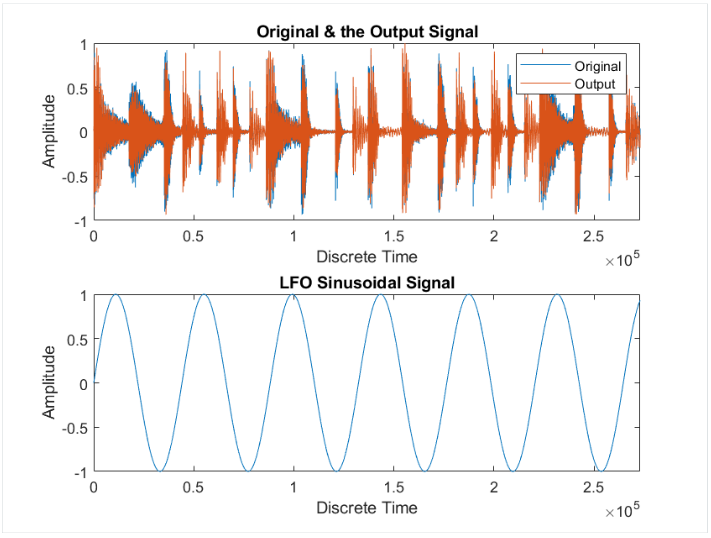


Figure 4: Plotting the Original & the Output Signal, as well as the LFO Signal.

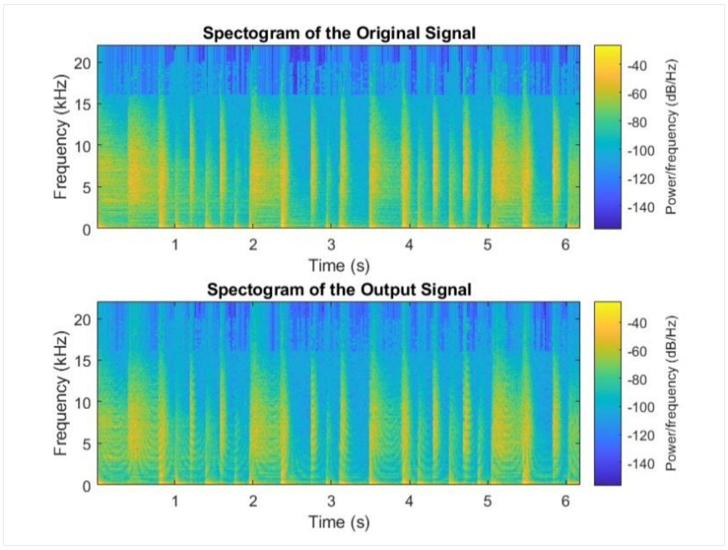


Figure 5: Plotting the Spectograms of the Original & the Output Signal.

RESULTS FOR THE CODE FOR CHORUS EFFECT

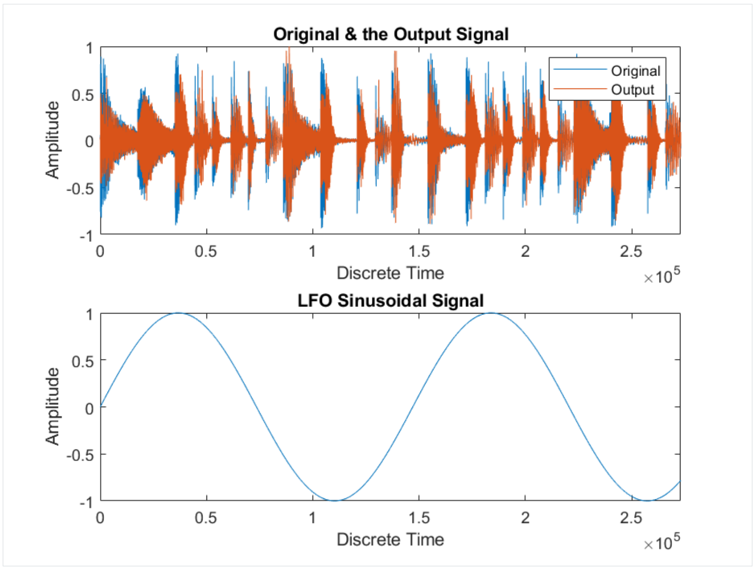


Figure 6: Plotting the Original & the Output Signal, as well as the LFO Signal.

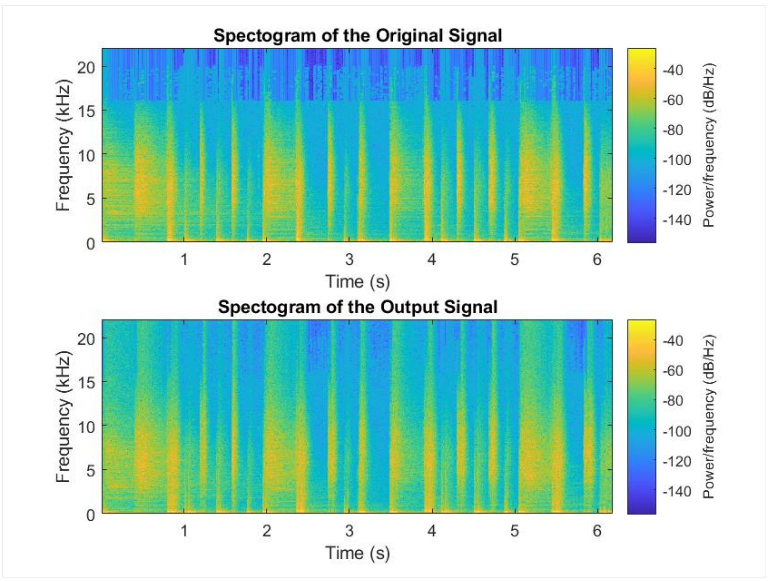


Figure 5: Plotting the Spectograms of the Original & the Output Signal.

The Google Drive links mentioned below consist of the input audio file and the output files which were generated for both the written codes (can only be accessed using the Nirma University mail id),

- Input File:
  https://drive.google.com/file/d/1AV4lnOMfs2_W05mcChYowl3jcvBSHxLt/view?usp=sharing
- Flanger Output File:
  https://drive.google.com/file/d/1KKX50b2uIBbyPVl3jpueJA-YVet0X_y4/view?usp=sharing
- Chorus Output File:
  https://drive.google.com/file/d/1T27lkIm31jJijzaFinRFWS6ENk7zj3lQ/view?usp=sharing

By executing the code for flanging for various parameters, it was observed that, the lesser the duration of delay, the more audible is the effect. Also, by reducing the rate of the LFO, a better and a more prominent effect can be observed.

By changing the parameters for the chorus effect, it was observed that, the higher the duration of delay, more audible is the effect. Also, the rate of LFO should be kept lower than 0.5 Hz in order for the slight flanging, due to the LFO to not be heard.

Effect of changing the modified amplitudes for both the effects was observed to be the same as in, more the amplitude, the more prominent is the effect.

For the flanging effect, there’s not much change observed in the input and output spectrograms but for the chorus effect, higher intensity in power is observed in the spectrogram of the output signal. This adds weight to the fact that chorus emulates the same instrument or audio playing twice.

CONCLUSION

Two very popular audio effects, namely the flanging and chorus effect were discussed in this report. By understanding the working of an FIR Comb filter, the procedure of implementing both these effects using an FIR Comb filter was explained. After this, experimental analysis was carried out for both the effects using MATLAB. The effect of changing parameters like the duration of delay, rate of the LFO and modified amplitude on the output signal were also observed and discussed in this report. The same effects can also be realized using an IIR Comb Filter, or a Universal Comb Filter. Using these two filters, one would get a more realistic output.

REFERENCES

[1] Udo Zolzer, *DAFX: Digital Audio Effects*, John Wiley & Sons Ltd., April 2002.