**USMAN MARUF**

**3041120**

**COMPUTER ENGINEERING 4**

4.a.

The maximum frequency humans can hear is 20 kHz. In musical applications however, it may be lower. For practical purposes, we assume the maximum frequency to be 20 kHz. Then by Nyquist rate will be 40 kHz.

The sampling period is therefore 1/40 kHz ≈ 2.5 ×10−5 seconds.

4.b.

load bach\_fugue.mat

note\_durations = theVoices.durations;

note\_numbers = theVoices.noteNumbers;

note\_frequencies = 440 \* 2.^((note\_numbers - 69)/12);

total\_duration = sum(note\_durations);

4.c.

load bach\_fugue.mat

note\_durations = theVoices.durations;

note\_numbers = theVoices.noteNumbers;

note\_frequencies = 440 \* 2.^((note\_numbers - 69)/12);

total\_duration = sum(note\_durations);

fs = 40000;

total\_samples = round(total\_duration \* fs);

waveform = zeros(1, total\_samples);

current\_sample = 1;

for i = 1:length(note\_numbers)

duration = note\_durations(i);

num\_samples = round(duration \* fs);

t = (0:num\_samples-1) / fs; % Time vector for the current note

f = note\_frequencies(i); % Frequency of the current note

sinusoid = sin(2 \* pi \* f \* t);

waveform(current\_sample:current\_sample+num\_samples-1) = sinusoid;

current\_sample = current\_sample + num\_samples;

end

4.d.

selected\_notes = [1, 2, 3];

figure;

hold on;

for i = selected\_notes

duration = note\_durations(i);

num\_samples = round(2 \* (1/f) \* fs);

t = (0:num\_samples-1) / fs; % Time vector for the current note

f = note\_frequencies(i); % Frequency of the current note

sinusoid = sin(2 \* pi \* f \* t);

plot(t, sinusoid, 'DisplayName', sprintf('Note %d: %.2f Hz', note\_numbers(i), f));

end

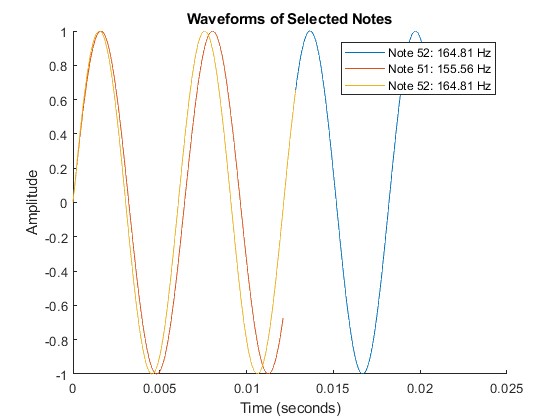
xlabel('Time (seconds)');

ylabel('Amplitude');

title('Waveforms of Selected Notes');

legend;

hold off;



4.e.

load bach\_fugue.mat

note\_durations = theVoices.durations;

note\_numbers = theVoices.noteNumbers;

note\_frequencies = 440 \* 2.^((note\_numbers - 69)/12);

total\_duration = sum(note\_durations);

fs = 40000;

total\_samples = round(total\_duration \* fs);

waveform = zeros(1, total\_samples);

current\_sample = 1;

figure;

hold on;

for i = 1:length(note\_numbers)

duration = note\_durations(i);

num\_samples = round(duration \* fs);

t = (0:num\_samples-1) / fs; % Time vector for the current note

f = note\_frequencies(i); % Frequency of the current note

sinusoid = sin(2 \* pi \* f \* t);

waveform(current\_sample:current\_sample+num\_samples-1) = sinusoid;

current\_sample = current\_sample + num\_samples; % Update the current sample index

end

portion\_duration = 2; % seconds

portion\_samples = round(portion\_duration \* fs);

portion\_waveform = waveform(1:portion\_samples);

window\_length = 512;

overlap = window\_length / 2;

nfft = 1024;

figure;

spectrogram(portion\_waveform, window\_length, overlap, nfft, fs, 'yaxis');

title('Spectrogram of a Portion of the Synthesized Music');

ylim([0 2]);

colorbar;

