工程數學 (二) project 2

鄭庭安、高逸傑、吳冠霆

2023 年 6 月 13 日

1 Synthetic signals

1.1 simple cosine signal

For a cosine signal with amplitude 1 and frequency 10Hz, its time-domain function $g(t)$ can be written as:

$$g(t) = \cos(2\pi 10t)$$

The Fourier Transform of g(t), can be derived manually as follows: Firstly, we know the definition of the Fourier Transform is

$$\hat{g}\left(t\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(t) e^{-i2\pi f t} dt$$

Substituting g(t) into it

$$\hat{g}\left(t\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \cos(2\pi 10t) e^{-i2\pi f t} dt$$

The solution to the integral is

$$\hat{g}\left(f\right) = \sqrt{\frac{\pi}{2}} (\delta\left(f-10\right) + \delta\left(f+10\right))$$

δ is Dirac delta function.

As shown in figure 1, the plot shows peaks at f=10Hz and f=-10Hz.

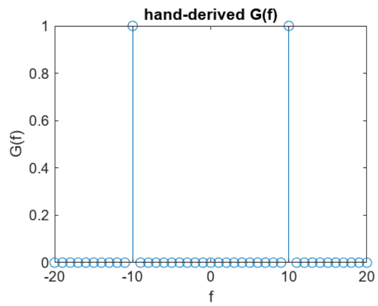
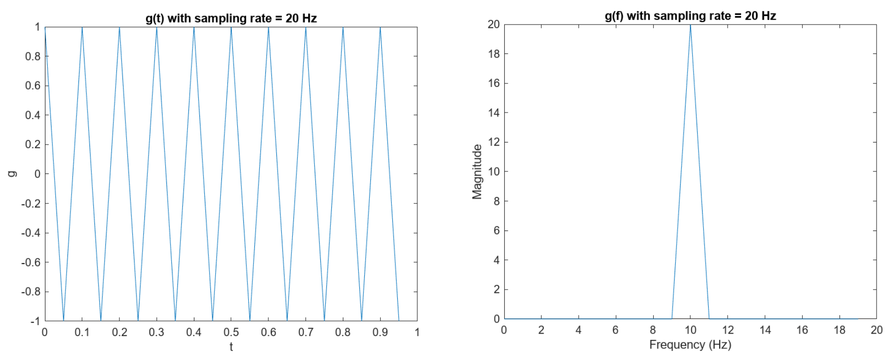


图 1: hand-derived $\hat{g}(f)$

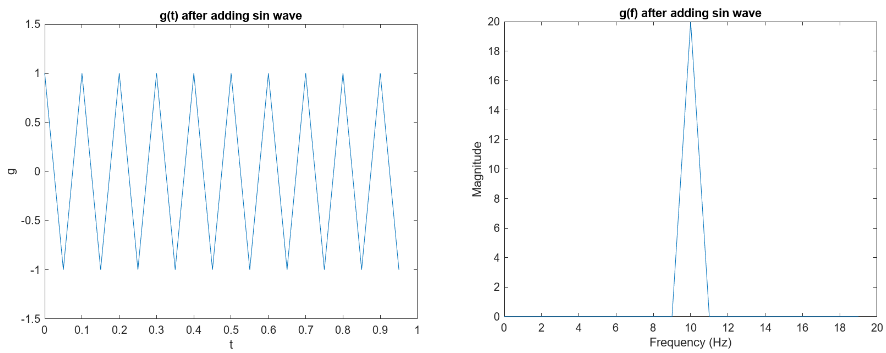
The magnitude spectrum in figure 2 should agree with the hand-derived solution, showing peaks at f=10Hz.



(a) time domain (b) frequency domain

图 2: discrete *cos* signal and it's spectrum

1.2 Phasing signal



(a) time domain (b) frequency domain

图 3: manipulated signal and it's spectrum

If we add a sine signal of identical amplitude and frequency to the original cosine signal, the new signal h(t) can be written as:

$$h\left(t\right) = \cos\left(2\pi 10t\right) + \sin(2\pi 10t)$$

This will result in changes to the raw signal in the time domain. Specifically, the new signal will have a different phase and amplitude.

The Fourier Transform of g(t), can be derived manually as follows: Firstly, we know the definition of the Fourier Transform is:

$$\hat{h}\left(t\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} h(t) e^{-i2\pi f t} dt$$

Substituting g(t) into it:

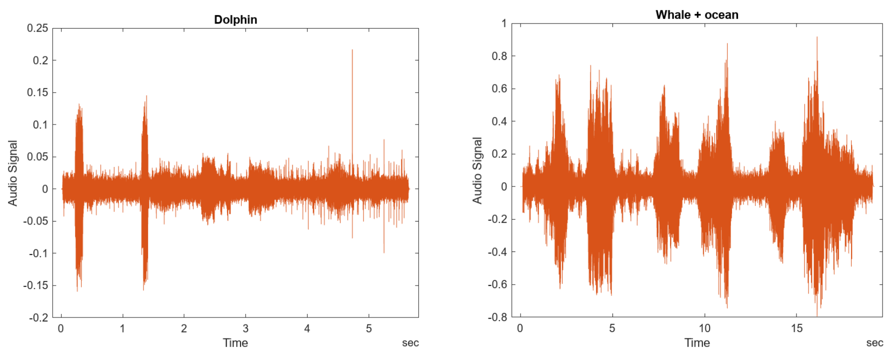
$$\hat{h}\left(f\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (\cos(2\pi 10t) + \sin(2\pi 10t)) e^{-i2\pi f t} dt$$

The solution to the integral is:

$$\hat{h}\left(f\right) = (1+i)\sqrt{\frac{\pi}{2}}\delta\left(f-10\right) + (1-i)\sqrt{\frac{\pi}{2}}\delta\left(f+10\right))$$

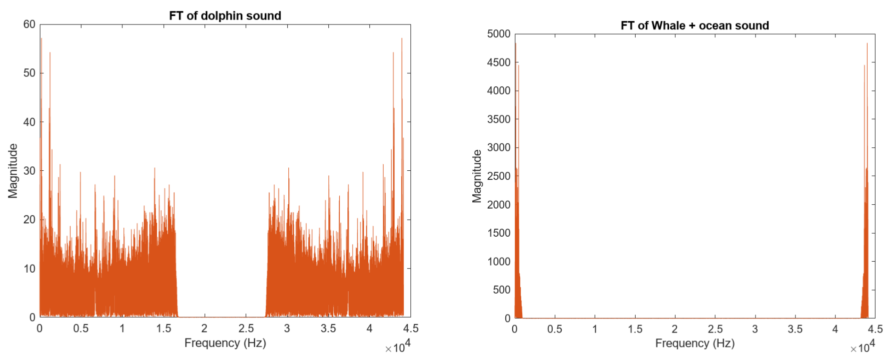
The magnitude spectrum’ s peak does not change in frequency. This is because the frequency of the signal is determined by the argument of the cosine and sine functions, which remains the same. However, the magnitude at the peak frequency changes due to the addition of the sine signal.

2 Audio manipulation



(a) Dolphin (b) Whale + Ocean

图 4: Noise in time domain



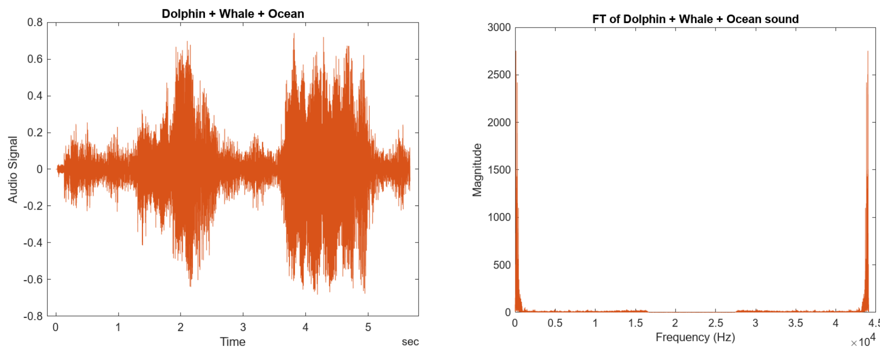
(a) Dolphin (b) Whale + Ocean

图 5: Spectrum of the noise

For the dolphin sound plot, it contains a range of frequency peaks, from 0 to 1.5×10^4Hz and 2.75×10^4 to 4.4×10^4Hz. With amplitude at around 30 and highest at around 60.

For the Whale + ocean sound plot, the frequency domain shows two prominent peaks. The frequency range is smaller than that of the dolphin plot. One at around 0.1×10^4 Hz, the other at around 4.3×10^4 Hz. With amplitude at around 2700 and highest at around 4800.

The range of frequency peaks in the dolphin plot is broader than that of the Whale + ocean noise plot, but the amplitudes are smaller compared to that of the Whale + ocean plot. There are some higher frequencies in the dolphin sound, compared to the Whale + ocean sound.



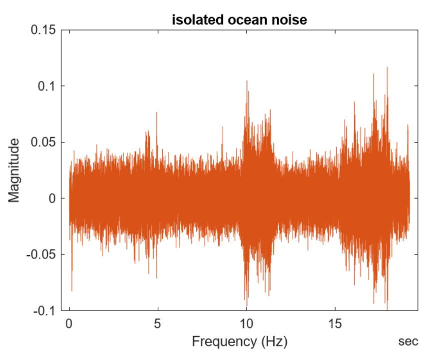
(a) time domain (b) frequenct domain

图 6: combined noise of dolphin, whale and ocean

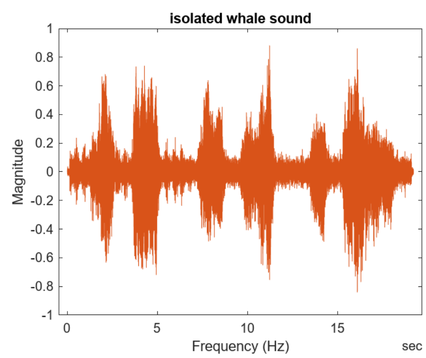
The dolphin+whale+ocean plot combines the individual dolphin and whale+ocean plots. This is logical since we simulated the simultaneous recording of dolphin and whale sounds by adding them together. As a result, the frequencies in the simulation should encompass all the frequencies present in both plots.

The broad frequency range observed in the dolphin plot within the dolphin+whale+ocean plot appears to have smaller magnitudes. However, it is important to note that the magnitudes are not inherently smaller, but rather appear smaller due to their ratio when compared to the magnitude of the whale+ocean sound.

Therefore, the two prominent magnitude peaks around 2500 are attributed to the whales+ocean plot, while the broad range of frequencies with low magnitude (around 30) is contributed by the dolphin plot.



(a) Ocean noise



(b) Whale sound

图 7: separated signal

Assume that the background noise of ocean repeats in a relatively constant pattern and are therefore always under certain amplitude. We can separate the whale sound from Whale+Ocean by muting all the frequency whose amplitude is under a threshold. With trial-and-error method, we decide the threshold to be 200, since the ocean noise under this constraint (in figure 7) shows fewer abrupt regions and still has acceptable performance in filtering.

3 Matlab code

```
%Part 1.1 c
x = -20:1:20;
y = dirac(x-10) + dirac(x+10);
idx = y == Inf; % find Inf
y(idx) = 1;      % set Inf to finite value
stem(x,y)

hold on
xlabel('f')
ylabel('g(f)')
title('hand-derived g(f)')
hold off

%Part 1.1 d
%% create cos function
```

```
% sampling period & frequence
Ts = 1/20; % for unknown reason, the peak in g(w) will start to leave 10Hz
    as the sampling period varies from (1/(2* 10Hz))
fs = 1/Ts;

% create the actual function
t = 0:Ts:1-Ts;
g = cos(2*pi*10*t);

plot(t,g)
xlabel('t')
ylabel('g')
title('g(t) with sampling rate = 20 Hz')

%%% same function with different smapling rate
Ts2 = 1/60;
fs2 = 1/Ts2;
t2 = 0:Ts2:1-Ts2;
g2 = cos(2*pi*10*t2);

plot(t2,g2)
xlabel('t')
ylabel('g')
title('g(t) with sampling rate = 60 Hz')

%% create fourier transform of g(t)
G= fft(g);
w = (0:length(G)-1)*fs/length(G);

G2= fft(g2);
w2 = (0:length(G2)-1)*fs2 /length(G2);


%% plot
plot(w,abs(G))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('g(f) with sampling rate = 20 Hz')


plot(w2,abs(G2))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('g(f) with sampling rate = 60 Hz')

%Part 1.2
```

```
%% create sin function
Ts = 1/20;
fs = 1/Ts;
t = 0:Ts:1-Ts;
g = g + sin(2*pi*10*t);

plot(t,g)
xlabel('t')
ylabel('g')
title('g(t) after adding sin wave')

%% do the FT
G= fft(g);
w = (0:length(G)-1)*fs/length(G);

plot(w,abs(G))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('g(f) after adding sin wave')

%Part 2
Dolpin = "C:\Users\WIN\OneDrive\Documents\MATLAB\EM 2 project\
    Audio_bottlenose-dolphin (online-audio-converter.com).wav";
WO = "C:\Users\WIN\OneDrive\Documents\MATLAB\EM 2 project\Audio_humpback-
    whale and ocean-noise (online-audio-converter.com).wav";

%% import audio data of dolphin
infoDolphin = audioinfo(Dolpin)

[yDolphin,FsDolphin] = audioread(Dolpin);

tDolphin = 0:seconds(1 / FsDolphin):seconds(infoDolphin.Duration);
tDolphin = tDolphin(1:end-1);


plot(tDolphin,yDolphin)
xlabel('Time')
ylabel('Audio Signal')
title('Dolphin')

%% import audio data of Whale + ocean
infoWO = audioinfo(WO)
[yWO,FsWO] = audioread(WO);

tWO = 0:seconds(1/FsWO):seconds(infoWO.Duration);
tWO = tWO(1:end-1);
```

```
plot(tWO,yWO)
xlabel('Time')
ylabel('Audio Signal')
title('Whale + ocean')

%% FT
G= fft(yDolphin);
w = (0:length(yDolphin)-1)*FsDolphin/length(yDolphin);
plot(w,abs(G))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('FT of dolphin sound')

G= fft(yWO);
w = (0:length(yWO)-1)*FsWO/length(yWO);
plot(w,abs(G))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('FT of Whale + ocean sound')

%% summation
ySum = yDolphin + yWO(1:length(yDolphin),:);
plot(tDolphin,ySum)
xlabel('Time')
ylabel('Audio Signal')
title('Dolphin + Whale + Ocean')

G= fft(ySum);
w = (0:length(ySum)-1)*FsDolphin/length(ySum);
plot(w,abs(G))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('FT of Dolphin + Whale + Ocean sound')

sound(ySum,FsDolphin)

%Bonus

GW = GWO;
for i = 1:length(GWO)
    for j = [1,2]
        if abs(GWO(i,j)) < 200
            GW(i,j) = complex(0,0);
        end
    end
```

```
end

yW = ifft(GW);

sound(yW, FsWO)

plot(tWO,yW)
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('manipulated W')
```