

Advanced Topics in Audio Processing Using Deep Learning

Barr Kirel, Dor Tirosh, Noam Shaked Cohen

January 3, 2026

* Code, output WAV files and all the figures can be found in our git - https://github.com/Nac0Jerk/audio_ml_tau_25_ex1

1 Analytic Part

1.1 Continuous Signals

1.1.1 Time convolution property

By definitions of convolutions and Fourier transform:

$$\mathcal{F}\{x_1(t) * x_2(t)\} = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} x_1(\tau)x_2(t-\tau)d\tau \right] e^{-j\omega t} dt \quad (1)$$

Interchanging the order of integration:

$$= \int_{-\infty}^{\infty} x_1(\tau) \left[\int_{-\infty}^{\infty} x_2(t-\tau)e^{-j\omega t} dt \right] d\tau \quad (2)$$

Let $u = t - \tau$, so $t = u + \tau$ and $dt = du$.

$$= \int_{-\infty}^{\infty} x_1(\tau) \left[\int_{-\infty}^{\infty} x_2(u)e^{-j\omega(u+\tau)} du \right] d\tau = \int_{-\infty}^{\infty} x_1(\tau)e^{-j\omega\tau} \left[\int_{-\infty}^{\infty} x_2(u)e^{-j\omega u} du \right] d\tau \quad (3)$$

This is exactly $X_2(w)$! so we got

$$= \int_{-\infty}^{\infty} x_1(\tau)e^{-j\omega\tau} X_2(\omega) d\tau = X_2(\omega) \int_{-\infty}^{\infty} x_1(\tau)e^{-j\omega\tau} d\tau = X_2(\omega)X_1(\omega) \quad (4)$$

So overall we got:

$$\mathcal{F}\{x_1(t) * x_2(t)\} = X_1(\omega)X_2(\omega) \quad (5)$$

1.1.2 Linearity property

$$\mathcal{F}(ax_1(t) + bx_2(t)) = \int_{-\infty}^{\infty} [ax_1(t) + bx_2(t)] e^{-j\omega t} dt \quad (6)$$

From linearity of integrals:

$$= \int_{-\infty}^{\infty} ax_1(t)e^{-j\omega t} dt + \int_{-\infty}^{\infty} bx_2(t)e^{-j\omega t} dt = a \int_{-\infty}^{\infty} x_1(t)e^{-j\omega t} dt + b \int_{-\infty}^{\infty} x_2(t)e^{-j\omega t} dt \quad (7)$$

Which is exactly

$$aX_1(\omega) + bX_2(\omega) \quad (8)$$

1.1.3 Scaling property

$$\mathcal{F}(x(at)) = \int_{-\infty}^{\infty} x(at)e^{-j\omega t} dt \quad (9)$$

Applying substitution of $u = at$, so $dt = \frac{1}{a}du$ and $t = \frac{1}{a}u$. the limits of the integration do not change as a is positive

$$= \int_{-\infty}^{\infty} x(u)e^{-j\omega \frac{1}{a}u} \frac{1}{a} du = \frac{1}{a} \int_{-\infty}^{\infty} x(u)e^{-j\omega \frac{1}{a}u} du = \frac{1}{a} \int_{-\infty}^{\infty} x(u)e^{-j(\frac{\omega}{a})u} du \quad (10)$$

Which is exactly the definition of the Fourier Transform with frequency $\frac{\omega}{a}$:

$$\mathcal{F}(x(at)) = \frac{1}{a} X\left(\frac{\omega}{a}\right) \quad (11)$$

1.1.4 Time shifting property

$$\mathcal{F}(x(t - t_0)) = \int_{-\infty}^{\infty} x(t - t_0)e^{-j\omega t} dt \quad (12)$$

We can substitute t to $u = t - t_0$ (note that $du = dt$) and get

$$= \int_{-\infty}^{\infty} x(u)e^{-j\omega(u+t_0)} du = \int_{-\infty}^{\infty} x(u)e^{-j\omega u} e^{-j\omega t_0} du = e^{-j\omega t_0} \int_{-\infty}^{\infty} x(u)e^{-j\omega u} du = e^{-j\omega t_0} X(\omega) \quad (13)$$

How does it affect the amplitude:

$$|e^{-j\omega t_0} X(\omega)| = |e^{-j\omega t_0}| \cdot |X(\omega)| \quad (14)$$

Since the magnitude of a complex exponential is always unity ($|e^{j\theta}| = 1$ for any real θ):

$$= |X(\omega)| \quad (15)$$

So time shifting does not change the amplitude spectrum of the signal.

As for the phase: Using the property that the angle of a product is the sum of the angles ($\angle(ab) = \angle a + \angle b$):

$$\angle [e^{-j\omega t_0} X(\omega)] = \angle(e^{-j\omega t_0}) + \angle X(\omega) \quad (16)$$

The angle of $e^{-j\omega t_0}$ is simply $(-\omega t_0)$:

$$\angle Y(\omega) = -\omega t_0 + \angle X(\omega) \quad (17)$$

So time shifting adds a linear phase term $(-\omega t_0)$ to the original phase spectrum. This means the phase is shifted by an amount proportional to the frequency ω .

1.1.5 Fourier transform of unit gate (rect) function

Applying the Fourier Transform definition on $\text{rect}(\frac{t}{\tau})$:

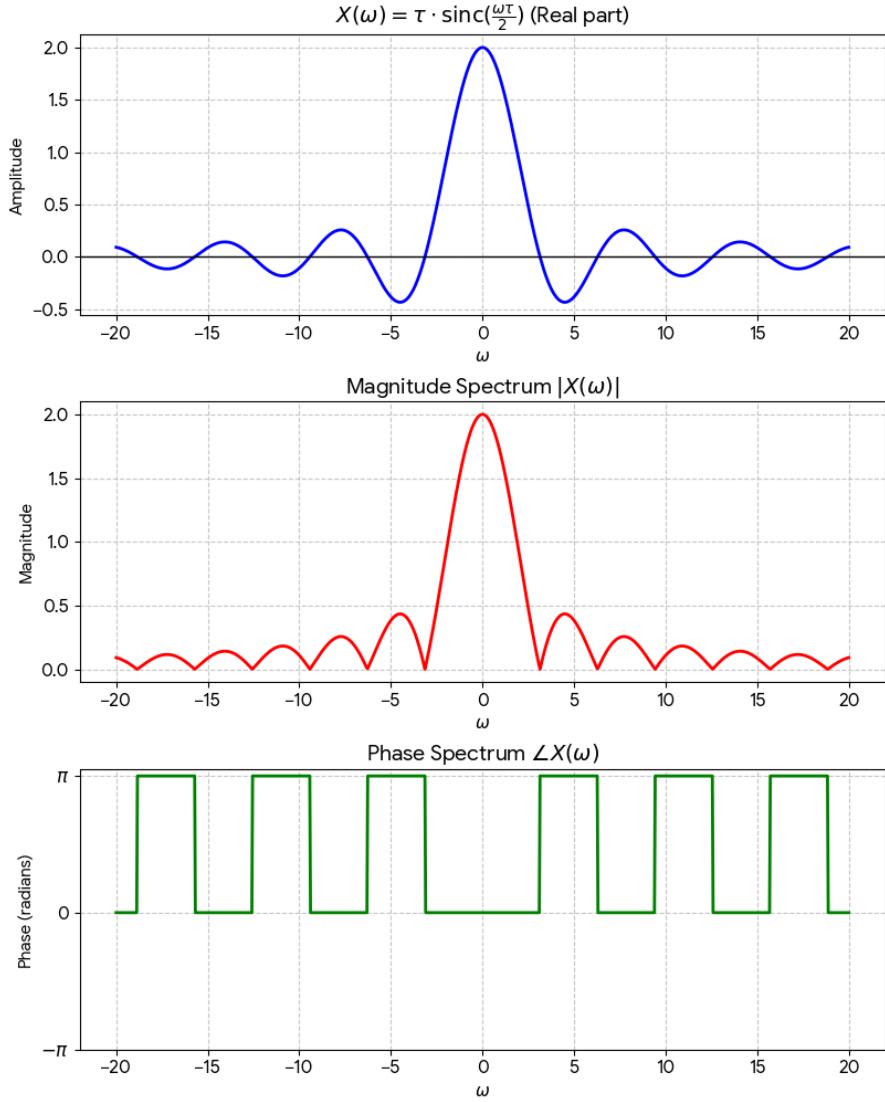
$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt = \int_{-\tau/2}^{\tau/2} 1 \cdot e^{-j\omega t} dt \quad (18)$$

Solving the integral:

$$X(\omega) = \left[\frac{e^{-j\omega t}}{-j\omega} \right]_{-\tau/2}^{\tau/2} \quad (19)$$

$$= \frac{1}{-j\omega} \left(e^{-j\omega\tau/2} - e^{-j\omega(-\tau/2)} \right) \quad (20)$$

$$= \frac{1}{j\omega} \left(e^{j\omega\tau/2} - e^{-j\omega\tau/2} \right) \quad (21)$$



Using $\sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j}$:

$$= \frac{2}{\omega} \left(\frac{e^{j\omega\tau/2} - e^{-j\omega\tau/2}}{2j} \right) \quad (22)$$

$$= \frac{2}{\omega} \sin \left(\frac{\omega\tau}{2} \right) \quad (23)$$

We can multiply and divide by $\tau/2$:

$$= \tau \cdot \frac{\sin(\frac{\omega\tau}{2})}{\frac{\omega\tau}{2}} \quad (24)$$

$$X(\omega) = \tau \text{sinc} \left(\frac{\omega\tau}{2} \right) \quad (25)$$

In order to draw it, note that: Power of $X(\omega)$ is just $|X(\omega)|$ And that because $X(\omega)$ is totally real, $\angle X(\omega)$ is either 0 or π , depends on whether its positive or negative.

1.2 Fourier Series

1.2.1 Fourier series of delta function

Given a unit impulse train $x(t) = \delta_{T_0}(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT_0)$.

The exponential Fourier series coefficients are defined as:

$$D_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-jn\omega_0 t} dt \quad (26)$$

Considering the interval $[-T_0/2, T_0/2]$, the function $x(t)$ contains only the single impulse at $t = 0$, so $x(t) = \delta(t)$.

$$D_n = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} \delta(t) e^{-jn\omega_0 t} dt \quad (27)$$

Using the sifting property of the delta function $\int \delta(t) f(t) dt = f(0)$:

$$D_n = \frac{1}{T_0} \cdot e^{-jn\omega_0(0)} = \frac{1}{T_0} \cdot 1 \quad (28)$$

So:

$$D_n = \frac{1}{T_0} \quad (29)$$

Since $x(t)$ is periodic, we can represent it as a Complex Exponential Fourier Series:

$$x(t) = \sum_{n=-\infty}^{\infty} D_n e^{jn\omega_0 t} \quad (30)$$

We previously calculated the Fourier coefficients D_n :

$$D_n = \frac{1}{T_0} \quad (31)$$

To find $X(\omega)$, we will be substituting this right back into the series:

$$x(t) = \sum_{n=-\infty}^{\infty} \frac{1}{T_0} e^{jn\omega_0 t} \quad (32)$$

We apply the Fourier Transform to both sides:

$$X(\omega) = \mathcal{F} \left\{ \frac{1}{T_0} \sum_{n=-\infty}^{\infty} e^{jn\omega_0 t} \right\} \quad (33)$$

$$X(\omega) = \frac{1}{T_0} \sum_{n=-\infty}^{\infty} \mathcal{F} \{ e^{jn\omega_0 t} \} \quad (34)$$

We use the standard Fourier Transform for a complex exponential:

$$\mathcal{F} \{ e^{j\omega_c t} \} = 2\pi\delta(\omega - \omega_c) \quad (35)$$

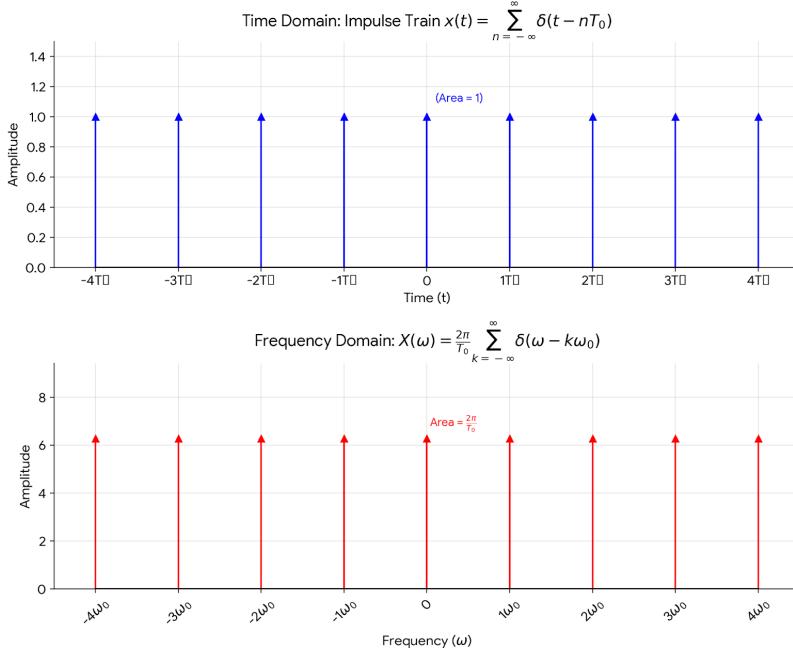
Therefore:

$$\mathcal{F} \{ e^{jn\omega_0 t} \} = 2\pi\delta(\omega - n\omega_0) \quad (36)$$

So:

$$X(\omega) = \frac{1}{T_0} \sum_{k=-\infty}^{\infty} 2\pi\delta(\omega - k\omega_0) = X(\omega) = \frac{2\pi}{T_0} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_0) \quad (37)$$

T_0 is the time duration between each two consecutive deltas in the time domain, and the interval between each D_i and D_{i+1} in the frequency domain is $\omega_0 = \frac{2\pi}{T_0}$



1.3 spectrum of a specific continuous, periodic function

The function is a sum of time shifted rects:

$$x(t) = \sum_{k=-\infty}^{\infty} rect\left(\frac{t+2\pi k}{\pi}\right) \quad (38)$$

In other words:

$$x(t) = rect\left(\frac{t}{\pi}\right) * \sum_{k=-\infty}^{\infty} \delta(t-2\pi k) \quad (39)$$

It's spectrum is:

$$X(\omega) = \mathcal{F} \left[rect\left(\frac{t+2\pi k}{\pi}\right) * \sum_{k=-\infty}^{\infty} \delta(t-2\pi k) \right] \stackrel{\text{from (5)}}{=} \mathcal{F} \left[rect\left(\frac{t+2\pi k}{\pi}\right) \right] \mathcal{F} \left[\sum_{k=-\infty}^{\infty} \delta(t-2\pi k) \right] \quad (40)$$

Using (25) and (8) we get:

$$X(\omega) = \pi sinc\left(\frac{\pi\omega}{2}\right) \sum_{k=-\infty}^{\infty} \mathcal{F}[\delta(t-2\pi k)] \stackrel{(37)}{=} \pi sinc\left(\frac{\pi\omega}{2}\right) \cdot \frac{2\pi}{2\pi} \sum_{k=-\infty}^{\infty} \delta(\omega-k) \quad (41)$$

1.4 Unit step function analysis

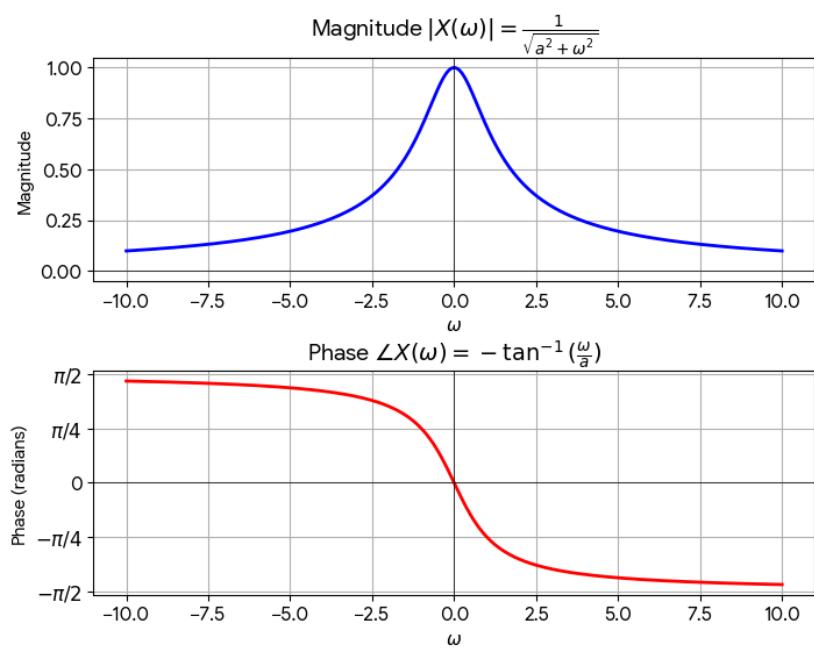
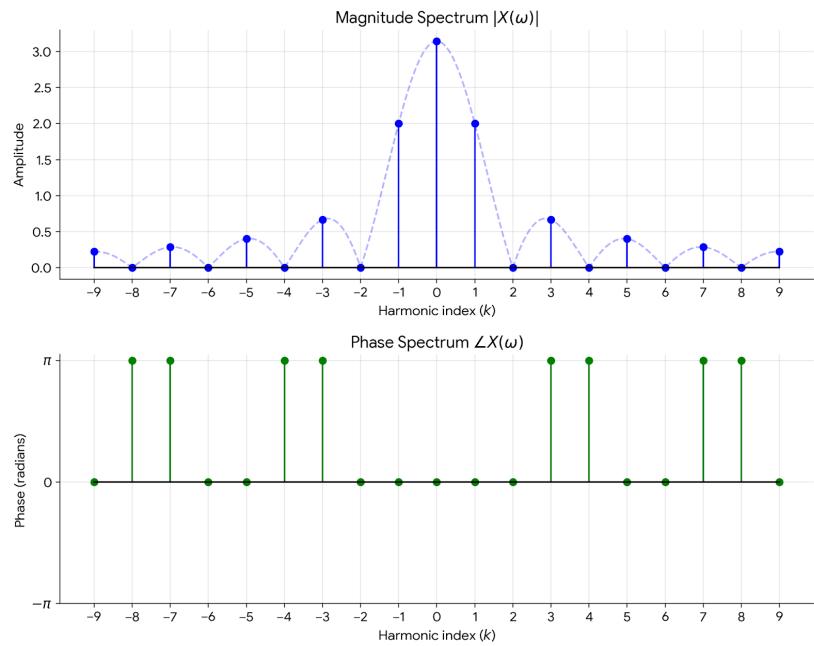
$$X(\omega) = \mathcal{F}[e^{-at} u(t)] = \int_{-\infty}^{\infty} e^{-at} u(t) e^{-j\omega t} dt = \int_0^{\infty} e^{-(a+j\omega)t} dt \stackrel{\text{integration}}{=} \left[\frac{e^{-(a+j\omega)t}}{-(a+j\omega)} \right]_0^{\infty} \quad (42)$$

Given that $a > 0$, the upper limit vanishes:

$$X(\omega) = 0 - \frac{1}{-(a+j\omega)} = \frac{1}{a+j\omega} \quad (43)$$

Its magnitude and phase are:

$$|X(\omega)| = \sqrt{\frac{1}{a-j\omega} \cdot \frac{1}{a+j\omega}} = \frac{1}{\sqrt{a^2 + \omega^2}} \quad (44)$$



$$\angle X(\omega) \stackrel{\text{angle identity}}{=} -\angle \frac{1}{X(\omega)} = -\angle(a + j\omega) = -\tan^{-1}\left(\frac{\omega}{a}\right) \quad (45)$$

Analyzing the filter type by checking limits:

$$\lim_{\omega \rightarrow 0} |X(\omega)| = \frac{1}{a}, \quad \lim_{\omega \rightarrow \infty} |X(\omega)| = 0 \quad (46)$$

Since low frequencies pass and high frequencies are attenuated, this is a Low Pass Filter.

1.5 Aliasing

1.5.1 Part (a) - Aliasing

Given:

- Sampling Rate: $F_s = 8000$ Hz
- Input Frequency: $f_{in} = 10$ kHz

Analysis: The Nyquist frequency is half the sampling rate:

$$F_{Nyquist} = \frac{F_s}{2} = \frac{8000}{2} = 4000 \text{ Hz}$$

Since the input frequency $f_{in} = 10$ kHz exceeds the Nyquist limit, aliasing occurs. The apparent (aliased) frequency f_{alias} is calculated by folding the frequency back into the valid range $[0, F_s/2]$:

$$f_{alias} = |f_{in} - k \cdot F_s|$$

Using $k = 1$ to minimize the result:

$$f_{alias} = |10000 - 8000| = 2000 \text{ Hz}$$

Answer: The 10 kHz signal will be aliased to **2 kHz**.

1.5.2 Part (b): Prevention

To prevent aliasing, the analog signal must be filtered before it enters the sampler. We use an **Analog Low Pass Filter** (An Anti-Aliasing Filter) to filter out all frequencies above the Nyquist frequency (4000 Hz).

Implementation option: A basic physical implementation is a passive RC circuit. The cutoff frequency f_c is determined by:

$$f_c = \frac{1}{2\pi RC}$$

For a design requirement of $f_c = 4000$ Hz, we can select a capacitor value of $C = 10$ nF, and a resistor with resistance of:

$$R = \frac{1}{2\pi f_c C} = \frac{1}{2\pi(4000)(10^{-8})} = \frac{10^5}{8\pi} \Omega \approx 3.98 \text{ k}\Omega$$

Note: RC circuits are classic but not a perfect solution for anti aliasing. A modern and better solution is using an Active Filter (using an Op-Amp) to prevent signal loss (because of loading effects) and to create a sharper cutoff slope.

1.6 Stereo Hearing Experiment

Observations

We performed the experiment and got the following results:

- When the left channel (`audio_l.wav`) is shifted by 2 ms (delayed relative to the right), the sound source appears to originate from the **right side**.
- When the right channel (`audio_r.wav`) is shifted by 2 ms (delayed relative to the left), the sound source appears to originate from the **left side**.

Explanation

The phenomenon observed is due to the fact we use two sensors in different location to preserve sound. Specifically using the delay between ears is called **Interaural Time Difference**.

1. **Mechanism:** If a sound originates from the right, it has a shorter path to the right ear and a longer path to the left ear. This results in the signal reaching the right ear first. The brain processes this time lead as a directional cue. The speed of sound is about 300 m/s , and the distance between our ears is about $\tilde{20} \text{ cm}$, therefore sound that come (for example) directly from the left side arrives to the right in a delay of about 0.6 ms .
2. **Simulation:** By artificially introducing a 2 ms delay to one channel, we simulate a path length difference.
 - Delaying the left channel implies the signal arrived at the right ear first, corresponding to a source on the **right**.
 - Delaying the right channel implies the signal arrived at the left ear first, corresponding to a source on the **left**.

1.7 Z-Transform Properties

1.7.1 Part (a): Convolution Property

Goal: Prove that $\mathcal{Z}(x_1[n] * x_2[n]) = X_1(z)X_2(z)$.

Proof: Let $y[n] = x_1[n] * x_2[n]$. By the definition of the Z-transform:

$$Y(z) = \sum_{n=-\infty}^{\infty} (x_1[n] * x_2[n])z^{-n}$$

Substitute the definition of discrete convolution:

$$Y(z) = \sum_{n=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x_1[k]x_2[n-k] \right] z^{-n}$$

Swap the order of summation:

$$Y(z) = \sum_{k=-\infty}^{\infty} x_1[k] \left[\sum_{n=-\infty}^{\infty} x_2[n-k]z^{-n} \right]$$

Perform a change of variable for the inner sum. Let $m = n - k$, which implies $n = m + k$. As $n \rightarrow \pm\infty$, $m \rightarrow \pm\infty$:

$$Y(z) = \sum_{k=-\infty}^{\infty} x_1[k] \left[\sum_{m=-\infty}^{\infty} x_2[m]z^{-(m+k)} \right]$$

Separate the exponent $z^{-(m+k)} = z^{-m}z^{-k}$ and factor out terms independent of m :

$$Y(z) = \sum_{k=-\infty}^{\infty} x_1[k]z^{-k} \left[\sum_{m=-\infty}^{\infty} x_2[m]z^{-m} \right]$$

Identify the Z-transform definitions $X_1(z)$ and $X_2(z)$:

$$Y(z) = X_1(z) \cdot X_2(z)$$

1.7.2 Scaling Property

Goal: Prove that $\mathcal{Z}(a^n x[n]) = X\left(\frac{z}{a}\right)$.

Proof: Let $y[n] = a^n x[n]$. By the definition of the Z-transform:

$$Y(z) = \sum_{n=-\infty}^{\infty} (a^n x[n]) z^{-n}$$

Combine the terms raised to the power of n :

$$Y(z) = \sum_{n=-\infty}^{\infty} x[n] (a^n z^{-n}) = \sum_{n=-\infty}^{\infty} x[n] (a^{-1} z)^{-n}$$

Rewrite the term inside the parenthesis as a fraction:

$$Y(z) = \sum_{n=-\infty}^{\infty} x[n] \left(\frac{z}{a}\right)^{-n}$$

Comparing this to the definition $X(z) = \sum x[n] z^{-n}$, we see that the argument z is replaced by z/a :

$$Y(z) = X\left(\frac{z}{a}\right)$$

1.8 DTFS

1.8.1 Fundamental Period

Given: $x[n] = \cos(0.1\pi n)$.

To find the fundamental period N_0 , we use the periodicity condition:

$$\frac{\omega_0}{2\pi} = \frac{1}{N_0}$$

where N_0 is integer.

Given $\omega_0 = 0.1\pi$:

$$\frac{0.1\pi}{2\pi} = \frac{0.1}{2} = \frac{1}{20}$$

Thus, the fundamental period is:

$$N_0 = 20 \text{ samples}$$

1.8.2 DTFS Coefficients

The Discrete Time Fourier Series representation of a periodic signal $x[n]$ with period N_0 is:

$$x[n] = \sum_{k=\langle N_0 \rangle} a_k e^{jk \frac{2\pi}{N_0} n}$$

Using Euler's formula, we expand $x[n]$:

$$x[n] = \cos\left(\frac{2\pi}{20} n\right) = \frac{1}{2} e^{j \frac{2\pi}{20} n} + \frac{1}{2} e^{-j \frac{2\pi}{20} n}$$

Comparing this to the DTFS definition, we identify the coefficients a_k (where the fundamental frequency is $\frac{2\pi}{20}$):

- For $k = 1$: coefficient is $\frac{1}{2}$.
- For $k = -1$ (equivalent to $k = 19$): coefficient is $\frac{1}{2}$.
- For all other k in the range $0 \leq k < 20$: $a_k = 0$.

Final Answer:

$$a_k = \begin{cases} 0.5 & k = 1, 19 \text{ (modulo 20)} \\ 0 & \text{otherwise} \end{cases}$$

2 Technical Part

In this section we will shortly answer the answer related to the technical part that was implemented using `main.py`. The outputs of this section would be provided along side the code in the `outputs` directory, this directory will include both the resulting audio files and higher resolution diagrams. Also most of the arbitraly chosen parameters in this exercise are configurable via the arguments to the script (use `--help` to see the possible options).

2.1 Initial Analysis

This section focuses on actually loading the the audio and drawing some plots for it. TBH we didn't see any clear difference between taking every even sample or using the `scipy.signal.resample` function, neither in sound quality or in the plots. By reading the documentation of `scipy.signal.resample` it looks like it should provide better anti-aliasing but on the other hand assumes the signal is periodic.

2.1.1 Resampled audio plots

The following diagrams (1,2) were generated using the command

```
python ./main.py --audio-file ./recordings/combined.wav --question a
```

We can visibly see that the Pitch Countour is missing on areas without speech meaning we couldn't find a matching pitch for that section.

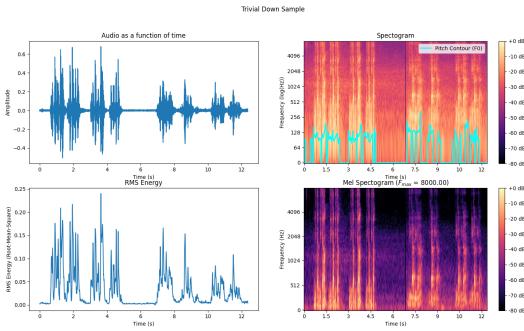


Figure 1: Plot of `1.c.trivial_down_sample.wav`

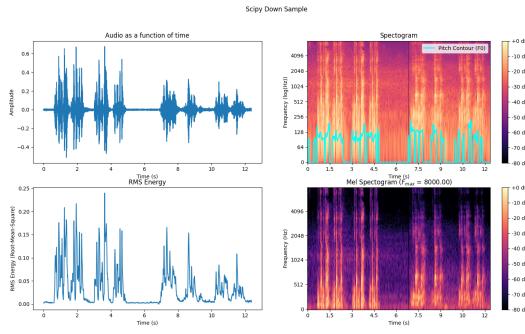


Figure 2: Plot of `1.c.scipy_down_sample.wav`

2.2 Adding noise

We were now requested to add noise to our signal. The following diagrams (3, 4, 5) show the original audio vs the noised audio and then the spectrogram of the noise and the spectrogram of the noised audio. The diagrams were generated using the command:

```
python ./main.py --audio-file ./recordings/combined.wav --question b:
```

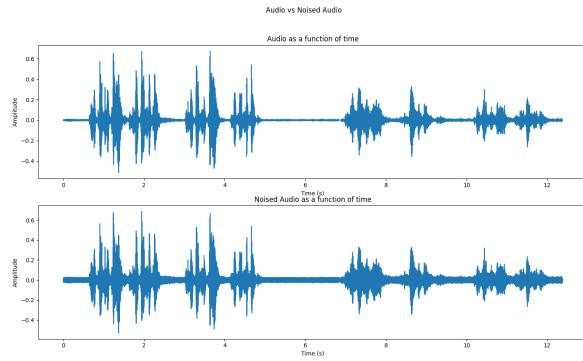


Figure 3: Plot of `1.c.scipy_down_sample.wav` vs `2.noisy_sample.wav`

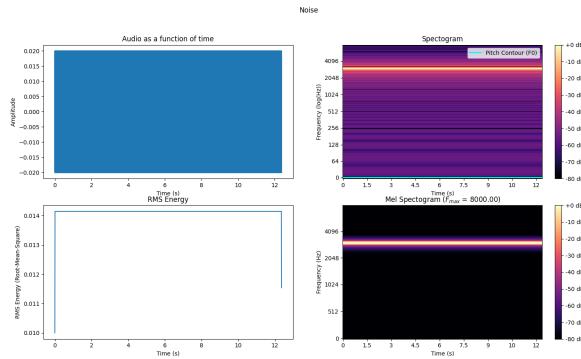


Figure 4: Plot of `stationary_noise.wav`

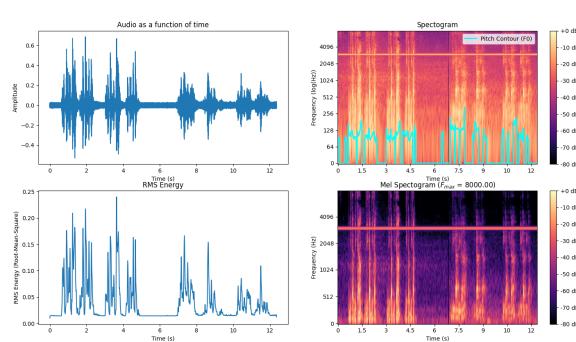


Figure 5: Plot of `2.noisy_sample.wav`

2.3 Spectral Subtraction

In this section we were requested to perform spectral subtraction in order to clear the noise we just add, we choose -17.95 dB as our noise threshold and used two different methods to clear the noise:

1. Classic averaging of the noise and then reduction
2. Interpolation of the noise between unnoised section

The rest of the exercise used the first method but we added both the code and the resulting diagrams for the second method as well. The following diagrams (6, 7) were generated using the following command:

```
python ./main.py --audio-file ./recordings/combined.wav --question c:
```

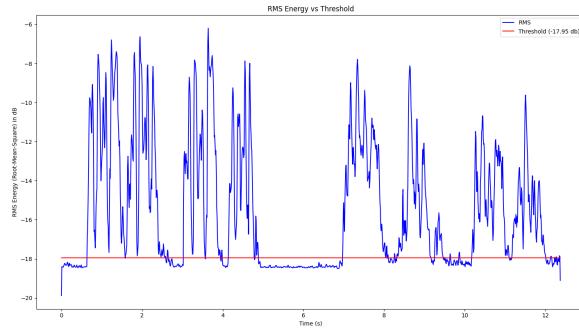


Figure 6: Plot of rms of 1.noisy_sample.wav vs the chosen threshold

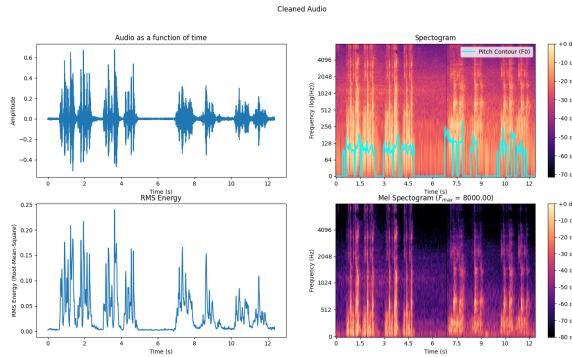


Figure 7: Plot of 3.cleaned_audio.wav

The following diagram (8) were generated using the following command:

```
python ./main.py --audio-file ./recordings/combined.wav --question c
--use-interpolate-for-spectral:
```

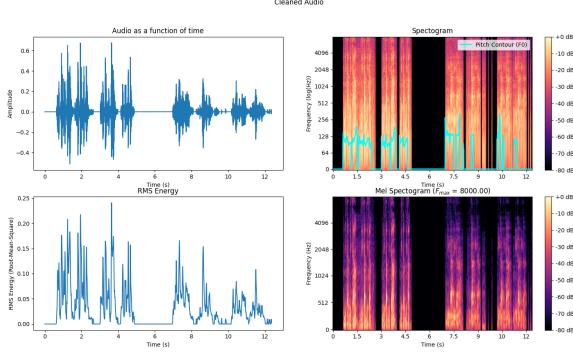


Figure 8: Plot of 3.cleaned_audio_interpolation.wav cleaned using interpolation of the noise

2.4 Auto Gain Control

In this section we were requested to apply Auto Gain Control on the previous cleaned audio. We chose a target rms of (-6 dB) and chose not to amplify areas detected as noise. The following diagrams (9, 10) were generated using the following command:

```
python ./main.py --audio-file ./recordings/combined.wav --question d
```

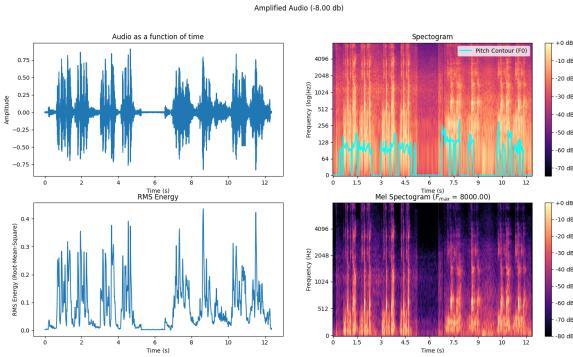


Figure 9: Plot of 4_cleaned_and_amplified_audio.wav

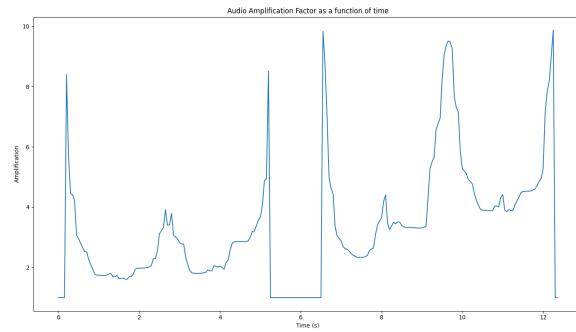


Figure 10: Plot of the amplification 3_cleaned_audio.wav experienced as a function of time

2.5 Time Stretching

This section was implemented by:

1. Performing `stft` on the input

2. Creating an interpolation function using `scipy.interpolate.interp1d`
3. Creating samples using the formula $1.5x$
4. Performing `istft` to get the final result

For fun we also generated different speeds. The following diagram (11, 12, 13) were generated using the following command:

```
python ./main.py --audio-file ./recordings/combined.wav --question e
--speed-up-factor X
```

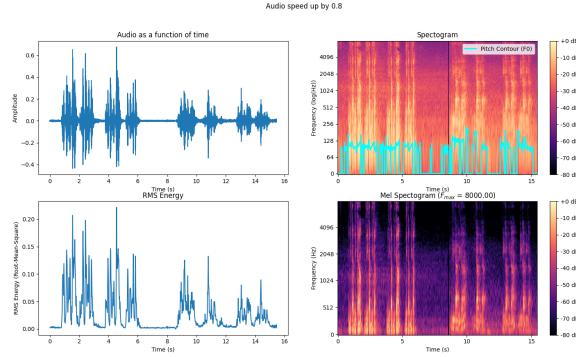


Figure 11: Plot of 5_speed_up_by_0.8_audio.wav

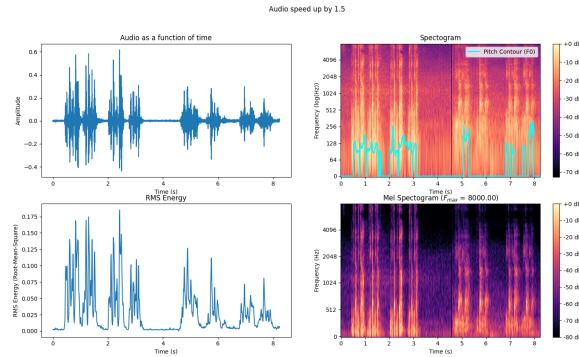


Figure 12: Plot of 5_speed_up_by_1.5_audio.wav

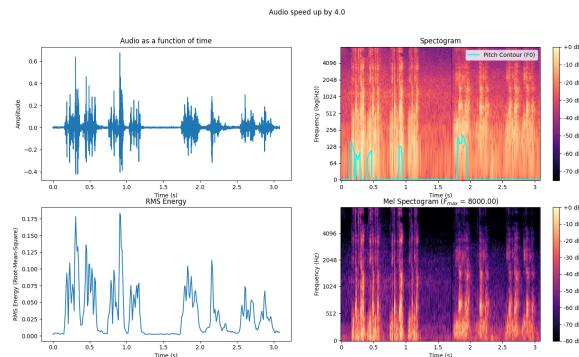


Figure 13: Plot of 5_speed_up_by_4_audio.wav