Pre-lab Questions

Question 1

An audio signal y(t) is sampled with fs=32 kHz. What sampling period does that correspond to? If you mistakenly play the signal with fs=16 kHz, will it be shorter or longer than the original? How else will it sound different?

Answer:

The sampling period is the inverse of the sampling frequency (fs).

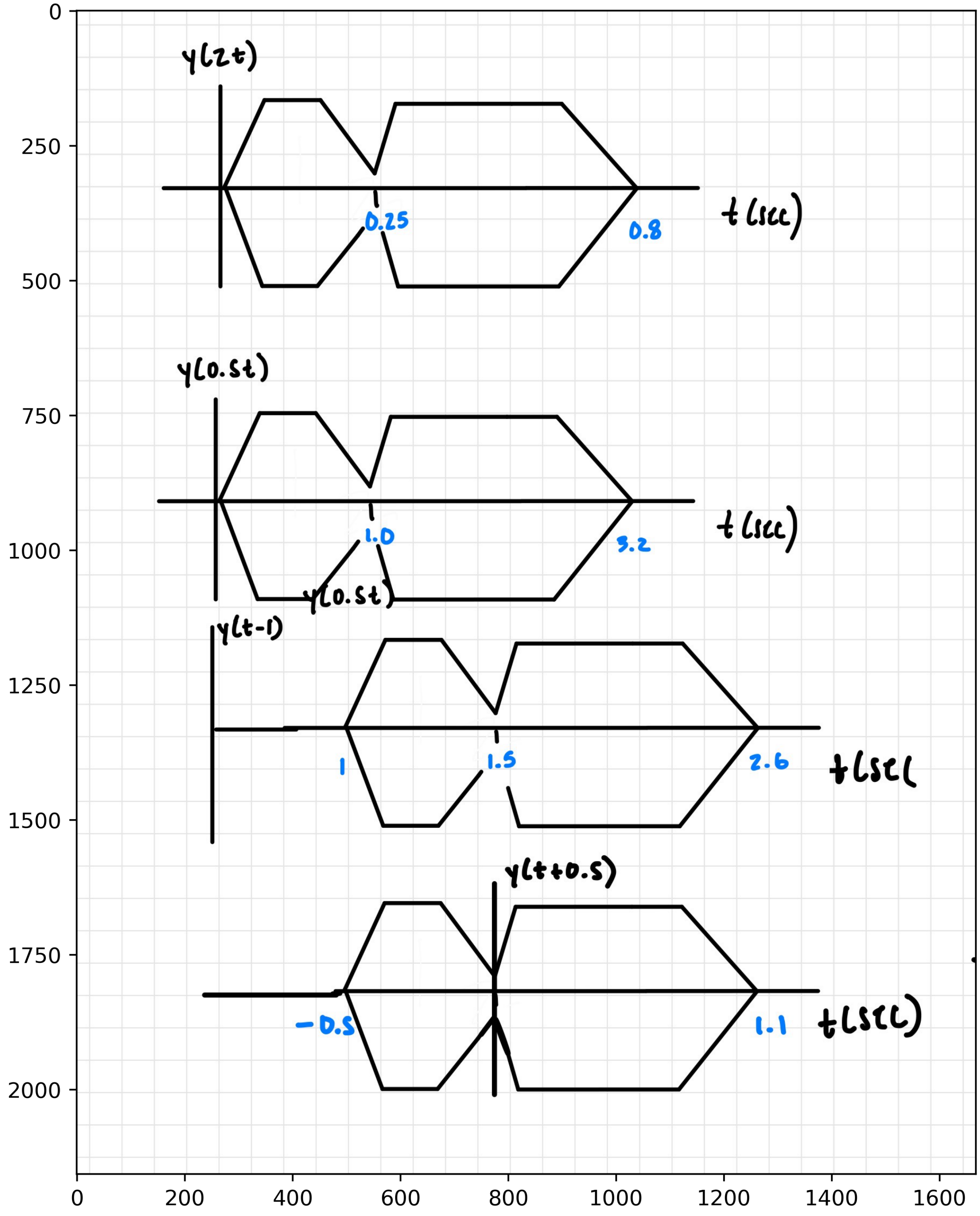
Sampling period (T) = 1/fs = 1/32000 = 31.25 µs

If you mistakenly play the signal with fs=16 kHz, it will be longer than the original. The duration will be doubled, as the sampling rate is halved. The sound will also be different, with a lower pitch, as the frequency content of the signal will be stretched out.

In [ ]:
```
# the answer is stored in PreLabP2.jpg
# Use python to display the image

import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np

plt.figure(figsize=(10,10), dpi=400)
img = mpimg.imread('PreLabP2.jpg')
plt.imshow(img)
plt.show()
```



Question 3

When the signal is digitized, you need to implement the time shift in terms of the number of samples: y[n-n1] and y[n+n2]. Find n1 and n2 (corresponding to t1=1 and t2=0.5, respectively) for the case when fs=32kHz.

Answer:

For fs = 32 kHz, the time shifts are t1 = 1 s and t2 = 0.5 s.

n1 = t1 *fs = 1* 32,000 = 32,000 samples n2 = t2 *fs = 0.5* 32,000 = 16,000 samples

Question 4

On a computer, we may have the constraint of keeping the time window fixed. Assuming the time window is constrained to be [0,3] sec, which of the time transformations in part 1 will require you to throw away some of the transformed signal? If you were to implement y(t)=x(2(t+1.5)) with a fixed time window, would it be better to scale first or shift first, or does it not matter?

Answer:

For a fixed time window of [0, 3] seconds, any time transformation that extends the signal outside this window would require us to throw away some of the transformed signal. In this case, the transformation y(t) = x(2(t+1.5)) involves both scaling and shifting.

To determine if it's better to scale first or shift first, we can analyze both options:

- Scaling first (y(t) = x(2t)): The signal is compressed along the time axis, fitting within the [0, 3] seconds window.
- Shifting after scaling (y(t) = x(2t+1.5)): The signal is shifted by 1.5 seconds. Since the scaled signal is shorter, it still fits within the [0, 3] seconds window.

- Shifting first (y(t) = x(t+1.5)): The signal is shifted by 1.5 seconds, extending beyond the [0, 3] seconds window.

- Scaling after shifting (y(t) = x(2(t+1.5))): The signal is compressed along the time axis, fitting within the [0, 3] seconds window.

Considering both options, it doesn't matter if you scale first or shift first. In both cases, the transformed signal fits within the fixed time window of [0, 3] seconds.