University of Washington ECE Department
EE 235 Lab 2 – Time Transformations of Signals

In this lab, you will work through a series of exercises to perform transformations on continuous-time signals, and get comfortable with relevant Python functions. Important concepts will include transformations in time (time shift and time scale), as well as multiple transformations using the order of operations.

Lab 2 Turn-in Checklist

- 4 pre-lab exercises
- 2 Assignment check-offs with TA
- Lab report, completed and submitted as a team
  - Submit as a Jupyter notebook following the format of the template provided
  - Submit the audio file recovered in assignment 3

Note: All assignments except the prelab should be completed in groups of 2-3 people. The pre-lab exercises are to be completed individually and submitted via Canvas before lab section.

Pre-lab

Read the Lab 2 Background document, then complete the following 4 exercises.

1. The envelope of an audio signal **y(t)** is approximated below:
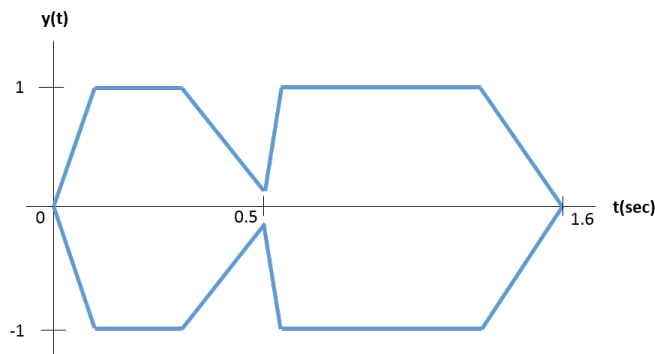
   a. Now consider the signal **y(2t)**.

      i. Sketch the signal **y(2t).** Label the height and the time points corresponding to **t = 0**, **t = 0.5**, **t = 1.6** after the time scaling.

      ii. If you played **y(2t)**, would it sound like **y(t)** has higher frequencies or lower frequencies?

   b. Repeat **(a)** for the signal **y(0.5t)**

   c. Sketch the signal **y(t-1)** and **y(t+0.5),** again labeling critical time points.

2. On a computer, we may have the constraint of keeping the time window fixed. If the time window is constrained to be [0,3] sec, then 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?

3. In lab, we will create a function that performs time scaling, and computes **y(t) = x(at)** when sampled at rate **fs**. We will use the header:

   def timescale(x, fs, a):

   a. What are this function's input parameters? Assume that it will return two variables **y** and **t**, how will both be returned simultaneously? (Hint: it involves one return line and commas)

   b. How would you call this function if you wanted to time scale a signal vector **y** with sampling rate **fs**, by a factor **2**, and store the variables it returns to **y1** and **t_y1**?

4. A signal y(t) was created from another signal x(t) by first scaling it by a factor of 3 and then delaying it by 1. In other words, y(t)=x(3(t-1)). Write an equation to undo the transformation and recover x(t) from y(t).

Download the template notebook for lab 2 (TemplateL2.ipynb) and rename it Lab2-XYZ, where XYZ are the initials of the lab partners. Your Lab 2 notebook should be organized according to this template.

Lab Assignments

This lab has 3 assignments. Each should be given a separate code cell in your Notebook, and each should be associated with a markdown cell with subtitle and discussion. As in Lab 1, your notebook should start with a markdown title and overview cell, which should be followed by an import cell that has the import statements for all assignments.

For assignments 1 and 2, you will use the train signal from Lab 1. Make sure you have "train32.wav" saved to your working directory. For assignment 3, your TA will assign you an audio file to upload.

```
%matplotlib notebook
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import scipy
from scipy.io import wavfile as wav
from scipy import signal as sig
import simpleaudio as sa
import decimal
```

You will be working with arrays, audio files and plotting, so you may want to refer back to the Lab 1 background discussion. Be sure to include the necessary import statements. Your input cell should look like:

Assignment 1: Time Scaling Audio Signals

We will now implement and test a **timescale** function. This assignment will have four parts, A-D, each of which should be indicated with comments, following the guidelines in the **Lab 2 template**.

A. Replicate the **timescale** function in Section 5 of the Lab 2 background document, and save it in its own cell, as indicated in the **Lab 2 template**.

B. Write the necessary code to do the following. Feel free to use a loop for repeated steps:

    a. Load "**train32.wav**" and save its signal as **y** and sampling rate as **fs**. Recall from lab 1 that this file only has one channel.

    b. Compute the time samples **t_y** using **y** and **fs**. Refer to Lab 1 if needed.

C. Use the timescale function to obtain w(t)=y(2t) and v(t)=y(0.5t)

    a. Create **w(t)** using **a=2** and store the outputs of the **timescale** function as **w** and **t_w.**

    b. Create **v(t)** using **a=0.5** and store the outputs of the **timescale** function as **v** and **t_v.**

    c. Play all three signals and verify that they sound different in the ways you predicted in the prelab. Warning: the **timescale** function will give you a result with float format and you need int16 for playing the audio file, so you will have to convert the format.

D. Load a new figure, and using a 3x1 subplot.

    a. Plot the signal **y** vs. **t_y** on the 1st subplot. Adjust your x-axis to [0, 4], y-axis to [-30000, 30000]. Label the axes and title your plot appropriately.

    b. Repeat for the signal **w** vs. **t_w** on the 2nd subplot.

    c. Repeat for the signal **v** vs. **t_v** on the 3rd subplot.

If you've implemented your code correctly, your figure should look something like:
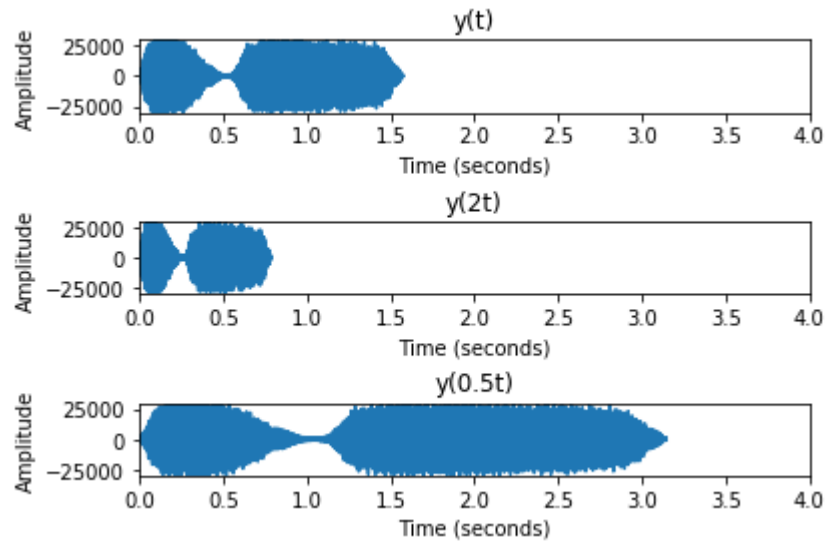
Assignment Check-Off #1 of 2: Demonstrate this Assignment to the lab TA by playing the three versions of the train signal.

Report discussion: What would happen if you used a time scaling factor of a=-1? Suppose a student runs the figure command before every call to subplot. When you run your script, what changes do you expect to see? How will the plots change?

Assignment 2: Time Shift Operation

We will now implement and test a **timescale** function. This assignment will have four parts, A-D, each of which should be indicated with comments, following the guidelines in the **Lab 2 template**.

A. Write a function called **timeshift** that takes as input: a signal **x**, the sampling frequency **fs** (in Hz), and a time shift **t0** (in seconds). The function should implement y(t) = x(t+t0) and produce as output the portion of the shifted signal starting at time 0. Assume that the original signal has value zero outside the time window. Your function should:

    i.   Find the integer shift **n0** given **t0** and **fs**.

    ii.   Use conditional control that tests whether the time shift is positive or negative

        a. For a time delay, create **y** by concatenating a zero vector with the original signal. (The output should be longer than the original signal.)

        b. For a time advance, create **y** by copying the portion of the starting from **n0** and then appending **n0** zeroes at the end of the signal. (The output should be the same length as the original signal.)

iii. Based on the length of the final signal and the sampling frequency, create a time vector that corresponds to the output signal length, starting at 0.

iv. Return the new signal and the time vector.

Save the function it in its own cell, as indicated in the **Lab 2 template**.

B. Use the function to create **y(t+0.5)** and **y(t-2)**. Plot the shifted signals with the original in a 3x1 plot: **y(t)**, **y(t+0.5)**, and **y(t-2)**. The **x-axis** should be between 0 and 4 for all three plots. Label axes and title the plot.

C. Play all three signals. For the signal that has been advanced, you should be able to hear that part of the sound is missing, since we have not preserved those samples.

Assignment Check-Off #2 of 2: Demonstrate this Assignment by showing the plot to the lab TA and playing the three signals.

Report discussion: There is a trivial case that you should ideally test for. If the shift is zero, then the output is the original signal. If the shift is an advance bigger than the original signal, then the output will be zero. Comment on whether your current implementation correctly handles these cases and whether there a better implementation.

3.0 Recovering Popular TV/Movie Audio File

Mayday, mayday! The ECE head TA is in dire need of succor. His favorite audio files of popular culture quotes that form the bulk of his personality have been tampered with by a mischievous undergraduate! Please help restore his wit by undoing the evil student's transformations in time and recover the signals in the TA's prized possession of audio files.

The professor is fairly certain that this was a harmless prank, and the student used a transformation that would not be too difficult recover, with transformation factors of 1/2 or 2. Therefore, the audio signal either went through a sequence of transformations to produce **y(t) = x(2t – 0.5)** or **y(t) = x(0.5t – 2)**. Your job is to implement the inverse transformation for both cases using the functions you created in assignments 1 & 2, and choose the original based on listening to the audio.

There are several audio files to fix. Your TA will assign each team a sound file to download; save it to your workspace folder.

A. Using time scaling then time shift, implement transformations to recover x(t) for the two options: a) **y(t) = x(2t – 0.5)**, and b) **y(t) = x(0.5t – 2)**. For the first case, call the output signal **ya** and the corresponding time samples **ta**. For the second case, use **yb** and **tb**, respectively.

B. Play both of the transformed signals, **ya** and **yb**. Determine which is most human like, and from there determine which was the transformation actually performed by the

mischievous student: **y(t) = x(2t – 0.5)** or **y(t) = x(0.5t – 2)**. Save the recovered (corrected) signal.

C. Start a new figure. Plot the distorted audio in a **2x1** subplot figure window, after computing the time sample **t_y**. Plot the recovered signal in the **2nd** subplot of the figure.

Assignment Check-Off: No assignment check-offs for this part, but you can play the corrected signal for the TA to get confirmation that you have it right.

Report discussion: Specify which transformation was used by the mischievous student, and identify the movie or TV show your quote came from. Below are some hints and fun facts about each quote.

S1: This quote comes from a popular science-fiction movie franchise directed by James Cameron that chronicles the ongoing battle between humans and Skynet. The film series stars a former Governor of California. Five films have been made so far with a sixth film is expected in 2019.

S2: This quote comes from the third entry in a popular movie franchise based on a series of British fantasy books chronicling the life of a boy wizard and his friends. The director won a 2014 Academy Award for Best Director for Gravity, which was the head TA's favorite movie from 2013.

S3: This quote comes from a 2016 standalone spinoff film to a popular space opera film series. The movie ended up being an unexpected prequel!

S4: This quote comes from a hugely popular HBO fantasy drama series. The head TA is a huge fan and waited 6+ hours in line to attend its San Diego Comic-Con program panel and was elated when HBO handed out free gift bags.

S5: This quote comes from an AMC post-apocalyptic horror drama TV series. At the 2013 San Diego Comic-Con, the head TA and a group of friends started standing in line at 4am to gain entrance in to the show's program panel that started at 2pm!

S6: This quote comes from the 55th Disney animated feature film about an unlikely friendship between a rabbit and a fox working to solve a crime. The movie is so good that its creators won a Golden Globe Award for best animated film.

S7: This quote comes from an animated sequel to a classic fish tale from Pixar. It is voiced by a famous talk show host and comedian recently honored by a Presidential Medal of Freedom.

S8: This quote comes from a web-slinging superhero movie trilogy that has been rebooted so many times that it is hard to keep count. Regardless, the head TA is a huge fan and has plans to watch every movie this superhero is in.

S9: This quote comes from a television sitcom on CBS about a group of four socially awkward geeks (ahem, intellectuals) living in Pasadena, CA. The word you hear is a regularly used catch phrase on the show that means "fooled you!"

S10: This quote comes from a popular, female-led book-turned-movie set in the fictional country of "Panem." The head TA was too excited to see the film and only finished 75% of the book before seeing it. He reports being surprised how many things the movie left out!

Team Report

When you've tested and cleaned up all your code (remember, you should only submit code for the Assignments, each in their own cell), go to 'File' then 'Download as', then select '.ipynb'. The file you download is a Notebook that your TA will be able to open and grade for you, once you

submit it on Canvas. Remember, only one notebook per team! Make sure that your notebook is titled Lab2-XYZ.ipynb, where XYZ are the initials of the lab partners. You may want to also download the file as pdf to have a nicer documentation of your records.

Submit via Canvas: i) the .ipynb file, and ii) the recovered audio file from assignment 3.