

University of Washington
Department of Electrical Engineering
EE 235 Lab 2
Continuous-Time Signals and Transformations in Time

In this lab, we will use Matlab to perform transformations in time on continuous-time signals. We will also introduce students to Matlab syntax for functions, calling Matlab functions with multiple outputs, and a few useful Matlab functions.

Important concepts from Lecture you will use in Lab 2

- ☐ Performing transformations in time: time shift and time scale
- ☐ Performing multiple transformations in time using order of operations

What is expected from you in Lab 2

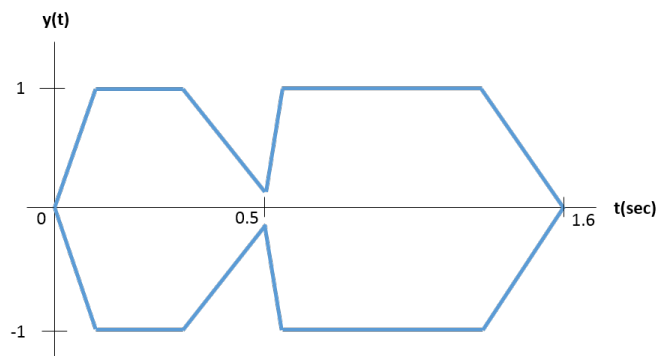
- ☐ Completion of 3 pre-lab exercises (3 points)
- ☐ Completion of 2 in-lab check offs with TA (4 points)
- ☐ Completion of a lab report (3 points)

Note: all pre-lab exercises must be completed by each student individually and turned in at the beginning of the lab. Each student will be checked off by the TA during the lab section.

PRE-LAB:

Read the Matlab Lab 2 tutorial, which reviews concepts from Lab 1 and introduces functions. This material will be needed to answer the prelab questions and in the lab.

- 1) In this problem, we'll explore time scaling. The envelope of the audio signal $y(t)$ in **train.mat** is approximated below:



- a) Sketch the envelope of signal $y(2t)$. Label the critical times $t = 0$, $t = 0.5$, and $t = 1.6$ after the time scaling. Is this an example of an audio signal speeding up or slowing down? Will the pitch be higher or lower?
- b) Repeat (a) for the signal $y(0.5t)$.
- c) To perform a time scaling and compute $y(t) = x(at)$ given an input signal $x(t)$ that has been sampled at rate F_s , we will use a function that has the following header:

$$[y, t] = \text{timescale}(x, F_s, a)$$

- i) What are the input variables of this function? What are the output variables of this function?
 - ii) How would you call this function if you wanted to time scale a signal vector y (sampled with a sampling rate of F_s) by a factor 2, the result of which you want to store in the output variables y_1 and t_{y1} ?
- 2) Using the same $y(t)$ as above to answer the following questions about time shifting.
- a) Let us first analyze the signal $y(t - 3)$:
 - i) Sketch the signal $y(t - 3)$. Make sure to label all important points, in particular, the locations of critical times $t = 0$, $t = 0.5$, and $t = 1.6$ after time shifting.
 - ii) Is this signal getting delayed or advanced?
 - b) Using your knowledge of how signals are represented on the computer using time samples, answer the following questions:
 - i) Given a vector t with values $[0, 0.5, 1, 1.5]$, how would you create a new vector that is $[t_0, t_0+0.5, t_0+1, t_0+1.5]$?
 - ii) Suppose we wish to create a vector of zeros that lasts for t_o seconds where we can vary t_o . Write the mathematical equation for the number of samples n_z needed in terms of t_o and the sampling rate variable F_s . Find n_z for the case where $t_o=3$ seconds and $F_s = 44,100$.
- 3) The goal now is to determine how much to scale and shift a transformed signal $y(t)$ in order to recover the original $x(t)$. We will derive the coefficients assuming that first we time scale $y(t)$ and then we shift the resulting signal.
- a) Consider the case where $y(t) = x(2t - 0.5)$.
 - i) The goal of the first step is to time scale $y(t)$ and produce a signal $y_1(t)$ that gets rid of the factor 2. That is, if we perform a TIME SCALE on $y(t)$ with factor " a ", then $y_1(t) = y(at) = x(2(at) - 0.5)$ which needs to equal $x(t - 0.5)$. What scaling factor " a " produces this result?
 - ii) The goal of the second step is to shift $y_1(t)$ and produce a signal $y_2(t)$ that removes the time shift 0.5. That is, if we perform a TIME SHIFT of " t_o " on $y_1(t)$, then $y_2(t) = y_1(t - t_o) = x((t - t_o) - 0.5)$ which needs to equal $x(t)$. What time shift " t_o " produces this result?
 - b) Repeat (a) for the other possible transformation $y(t) = x(0.5t - 2)$. What should the scaling factor " a " be? What should the time shift " t_o " be?

IN-LAB:

Complete Exercises 1-3, referring to the Lab 1 and Lab 2 tutorials as needed.

EXERCISE #1: Time Scale Operation

In this exercise, we will learn how to call and test a function written in Matlab that performs time scaling. To test the time scale function, we will use the built-in Matlab file, **train.mat**, which has an envelope like you worked with in the pre-lab.

Lab Exercise:

- a) Load Matlab and create folder **Lab 2** in your U Drive to use as your working folder.
- b) Download the Matlab script **timescale.m** from the course website and save the file in your working folder. Open the file in Matlab and read through the code and comments. Note: this script will NOT be modified in this lab exercise.
- c) We will now test this function in a separate script. Open a script and call it **Ex1.m**.
- d) To start with **Ex1.m**, write the lines of code to clear all figures and close all windows.
- e) Load the data file **train.mat** via the command **load**. The file will have two variables:
 y: this is sound vector
 Fs: this is sampling rate of sound vector **y**
- f) Compute corresponding time samples of this sound vector using **y** and **Fs** and assuming that the signal starts at time 0. Store the time samples in vector **t_y**.
- g) Load a new figure window.
- h) Using a **3 x 1** figure window, plot the signal **y vs. t_y** on the 1st subplot. Adjust the limits of the x-axis to be between 0 and 4, and the limits of the y-axis to be between -2 and 2. Make sure to provide labels and a title to your plot.
- i) Run the lines of code you have so far via **Editor -> Run** or by pressing **F5**. Verify that your plot matches the envelope signal **y(t)** you analyzed in pre-lab.
- j) We will now write the code to implement the transformation for signal **y(2t)**
 - i) Write the line of code to call the function **timescale** so that you pass in the inputs **y**, the sampling rate **Fs**, and the correct time scale factor. Store the outputs of the function in the variables **y1** and **t_y1**.
 - ii) Using the same figure window you plotted **y vs. t_y**, plot **y1 vs. t_y1** on the 2nd subplot. Adjust the x-axis so that limits are between 0 and 4, and adjust the y-axis to be between -2 and 2. Label your plot appropriately.
 - iii) Run the script you have so far via **Editor -> Run** or by pressing **F5**. Verify that your plot of **y(2t)** matches your pre-lab results.

- k) Repeat (2j) for the signal $y(0.5t)$. Store your output in the variables **y2** and **t_y2**. Plot **y2** vs. **t_y2** on the same figure window as the 3rd subplot. Use the same axes limits, and label your plot appropriately. Run your script to verify that your plot of $y(0.5t)$ matches your pre-lab results.
- l) We will now play all three audio signals to confirm your answers to the pre-lab questions on whether a signal speeds up or slows down.
 - i) Using the function **sound**, write the line of code to play the original sound signal **y**. Then, in succession without pause, play the sound signal **y1** and the signal **y2**.
 - ii) Run your entire Matlab script and pay close attention to the speed of each audio signal. Verify your answers in pre-lab.
- m) Provide comments (using the symbol **%**) in your script.

Lab Check-Off #1 of 2: Demonstrate your Matlab script **Ex1.m** to the lab TA and show you that you know how to perform time scaling in Matlab.

Lab Report Question #1 of 4: Suppose a student runs the **figure** command before every call to **subplot**. When you run your script Ex1.m again, what changes do you expect to see? How will the plots change?

Lab Report Question #2 of 4: Suppose that you do the time scaling simply by changing the time vector, $t_{y2b} = 2 * t_y$, and then you plot vector **y** using times **t_y2b** rather than **t_y**. The plot would look very similar to the plot for **y2**. What would you have to change in the sound command in order to have the same sound as **y2**?

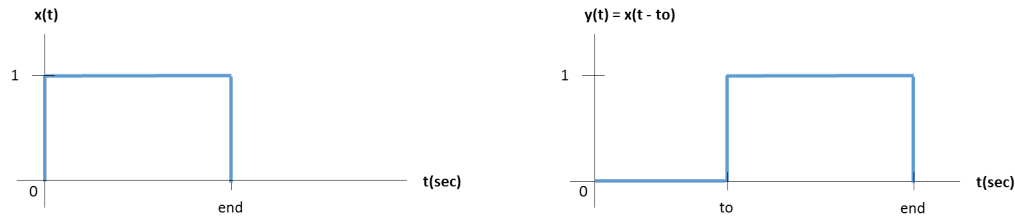
EXERCISE #2: Time Shift Operation

In this exercise, we will write our own Matlab function to perform a time shift. The built-in Matlab file, **train.mat**, will again be used to test this function. In this lab, we'll do both positive and negative time shifts, assuming that the signal to the left and right of the finite-length input signal is zero-valued. Here, we'll use a plotting time window of [-1,5].

Lab Exercise:

- a) Download and open the file **timeshift.m** from the course website. Unlike in Exercise #1, we will be writing a few lines of code in this file.
- b) We will now fill in the first part of the code for **timeshift.m**. We will simulate a time shift $x(t-t_0)$ by adding zeros to the beginning or end of the audio signal, depending on the time shift. Use the variable **t0** for t_0 in Matlab (**t0**>0 for delay, and **t0**<0 for advance), and assume **t0** is given in seconds.
 - i. The number of zeros required to pad the signal before or after the original one will therefore depend on **|t0|** (**abs(t0)** in matlab). Using your equation from pre-lab, compute the number of zeros needed to last **|t0|** seconds given the sampling rate is **Fs**. Store this result in the variable **nz**. Use the function **ceil** to round your answer to an integer.

- ii. Using the **zeros** function, create a column vector of zeros called **z** with the number of rows equal to the number of zeros **nz** required to shift the signal.
- c) We will start with a time shift to the right under the “if” statement.



- i) Insert the zeros before the signal by concatenating **z** and **x** to form the output vector **y**. Note that your output signal will be longer than your input signal by **nz** samples.
- ii) Compute the corresponding time samples of output vector **y** using the sampling rate **Fs** and assuming that the first sample is at time 0. Store this result in the variable **t**. Make sure that the vectors **t** and **y** have the same length.
- c) Now let us perform a time shift to the left under the “else” statement. For a time shift to the left, we pass a **negative** value for **t0**.
 - i) Insert the zeros after the signal by concatenating **x** and **z** to form the output vector **y**. Again, your output signal will be longer than your input signal by **nz** samples.
 - ii) Compute the corresponding time samples of output vector **y** using the sampling rate **Fs** and assuming that the first sample is at time **t0**. Store this result in the variable **t**. Make sure that the vectors **t** and **y** have the same length.
- d) Complete your edits of **timeshift.m** by adding a function header comment, as well as inline comments. Note that the output signal will be different in length than the input signal.
- e) We will now test your function by writing a test script **Ex2.m**. Open up a new script and call it **Ex2.m**. To start in **Ex2.m**, first clear all variables and close all figures.
- f) Load data file **train.mat** via the command **load**.
- g) Using the same procedure in **Ex1.m**, plot the audio signal **y** as the 1st plot in a 3 x 1 subplot figure window. Make sure you compute the corresponding time samples **t_y** before plotting. Adjust the limits of the x-axis to be between -1 and 5. The y-axis does not need to be adjusted. Make sure to label all your axes and title your plot.
- h) We will now write the code to perform the time shift **y(t - 3)**
 - i) Call the function **timeshift** so that you pass in the inputs **y**, the sampling rate **Fs**, and the correct time shift amount needed. Store outputs in the variables **y1** and **t_y1**.
 - ii) Using the same figure window you plotted **y** vs. **t_y**, plot **y1** vs. **t_y1** on the 2nd subplot. Adjust the x-axis so that limits are between -1 and 5. Label your plot appropriately.
 - iii) Run the script you have so far via **Editor -> Run** or by pressing **F5**. Verify that your plot of **y(t - 3)** matches your pre-lab results.

- i) Repeat (2i) for the signal $y(t + 0.5)$. Store your output in the variables **y2** and **t_y2**. Plot **y2 vs. t_y2** as the 3rd subplot on the same figure window. Make sure to run your script to verify that your plot of $y(t + 0.5)$ matches your pre-lab results.
- j) We will now play all three audio signals to confirm your answers to pre-lab questions on whether a signal is delayed or advanced. Using the function **sound**, play the original sound signal **y**. Then, play the sound signal **y1**. You should notice the start-time delay. (You won't notice anything when you play the sound signal **y2** since we don't have a concept of negative when we're listening.) Run your entire Matlab script and pay close attention when you play the three audio signals. Verify your answers to pre-lab.
- k) Provide in-line comments for your script.

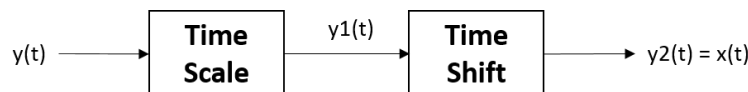
Lab Check-Off #2 of 2: Demonstrate your Matlab script **Ex2.m** to the lab TA.

EXERCISE #3: Recovering Popular TV/Movie Audio File

Mayday, mayday, mayday, [insert your name here]! A former EE235 TA's favorite audio clips of popular tv/movie quotes have been tampered with by an evil former student using what s/he knew about transformations in time. Your mission is to undo the evil former student's transformations in time and recover the signals in the TA's prized possession of audio files.

We need to first prep and train you in the art of undoing transformations in time, [insert your name here]. Luckily, we know the evil student failed his/her midterm question on time reversal, so we believe that the student only used time scaling and time shifting. (It would be noticeable if there's a reverse.) The TA is also certain the student used transformation values of factors 0.5 or 2.0 because $2.0 / 0.5 = 4.0$, which is the grade the former student was trying to get. Therefore, the audio signal either went through a sequence of transformations to produce $y(t) = x(2t - 0.5)$ or $y(t) = x(0.5t - 2)$.

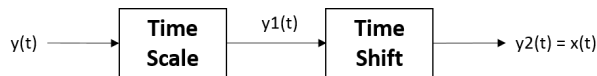
Regardless of which output is correct, the transformations in time can be undone by performing the transformations in reverse order. If we say the signal $x(t)$ first went through a TIME SHIFT and then a TIME SCALE to produce $y(t)$, then to undo the transformation, we need to perform a TIME SCALE and then a TIME SHIFT on $y(t)$ to recover $x(t)$ as shown below:



Lab Exercise: We will now try to undo the transformations in time by testing the values you computed in pre-lab.

- a) Ask the TA to assign you one of the 10 Matlab sound files on the class website. Download the file in to your workspace folder.

- b) Open up a new script file and call it **Ex3.m**. Clear all variables and close all figures.
- c) Load your Matlab data file. The Matlab data file will again have two variables: **y** and **Fs**.
- d) Plot the distorted audio signal **y** as the 1st subplot in a **2 x 1** subplot figure window. Make sure to compute the corresponding time sample **t_y** before plotting. No need to adjust the axes, but make sure to label and title your plot.
- e) Listen to the file and guess which transformation was used based on what speech would sound like when it was speeded up or slowed down.
- f) We will now undo the operations for **y(t) = x(at - b)** using the same order of operations shown below from pre-lab:



- i) Undo the time scale operation on audio signal vector **y** by calling the **timescale** function and passing the appropriate arguments using your transformation values from pre-lab. Call the output signal **y1** and the corresponding time samples **t1**.
- ii) Undo the time shift operation on audio signal **y1** by calling the **timeshift** function using your transformation values from pre-lab. Call the output signal **y2** and the corresponding times samples **t2**.
- g) Run your script and make sure you do not have any errors. Using the function **sound**, play the resulting signal to determine whether you have an audible result.
- h) Repeat (f) for the other transformation if you don't think you guessed correctly.
- i) Once you have figured out which recovered signal is correct, go back to your script **Ex3.m** and plot that signal as the 2nd subplot and save the audio file. No need to adjust the axes. Just make sure to label and title your plot.
- j) Provide comments for your script.

Lab Report Question #3 of 4: What is the quote in the sound file? If you are having trouble identifying some of the words, feel free to ask your TA for help. Can you identify which popular TV show or movie that quote came from? Here are some hints and fun facts about each signal that you can Google to answer the lab report question.

- **S1:** This quote comes from a popular TV show on the air in 2015.
- **S2:** This quote comes from the third entry in a popular movie franchise based on a series of British fantasy books. The director won a 2014 Academy Award for Best Director for Gravity.

- **S3:** This quote comes from the sequel to a reboot of another popular science fiction action franchise, set in the year 2259 and directed by J.J. Abrams.
- **S4:** This quote comes from a hugely popular 2003 animated fish tale from Pixar, and is voiced by an equally popular TV talk show host. A sequel to the film was released in 2016.
- **S5:** This quote comes from a popular superhero movie trilogy from Sony Pictures that was recently rebooted in 2012 with a whole new director and cast of actors. This web-slinging super hero was featured in a sequel released back in May 2014.
- **S6:** This quote comes from a hugely popular, female-led book-turned-movie set in the fictional country of "Panem."
- **S7:** This quote comes from an AMC post-apocalyptic horror drama TV series.
- **S8:** This quote comes from another superhero movie franchise set in the fictional Gotham City. The actor behind the voice of this sound file won an Oscar for his role as the film's main villain.
- **S9:** This quote comes from a popular ABC TV show that completed its 121-episode run in 2010. This science fiction adventure series follows the survivors of a commercial passenger jet crash on a mysterious tropical island.
- **S10:** This quote comes from a CW TV show based on a green costumed crime-fighter from DC Comics.

Lab Report Question #4 of 4: Suppose a student accidentally flips the order of function calls in step (f) when performing the time transformations, and calls **timeshift** first before calling **timescale**. Clearly, the output of the sequence of transformations will no longer be $x(t)$. What will the output be in terms of signal x ? Define your answer mathematically, and also describe answer in words. As an example, the answer mathematically could be $x(0.5t)$, and the description in words would be the signal x slows down by factor 2.

LAB REPORT:

Turn in a PDF of your lab report (one per team) online via the link on the class website. A sample format for the lab report is on the class website. You may vary from this format but please include the basic sections. The report is due prior to the start of your next lab section. Lab turn-in times will be checked, so no late lab reports will be accepted unless arranged in advance with the instructor.