Lab 2 - Resampling, Reconstruction and Convolution

Section	L01	L02	L03	L04
Lab Date	Oct. 06	Oct. 07	Oct. 08	Oct. 09
Due Date for Report	Oct. 20	Oct. 20	Oct. 20	Oct. 20

Assessment: 5% of the total course mark.

Objectives:

- To gain experience in implementing and analyzing digital systems within `MATLAB`. In particular, you will look at time domain convolution, system impulse responses, and resampling and reconstruction of 2-dimensional images.
- To gain experience in creating projects in `CCS5` and running programs on the TMS 320 DSP processor. You need to understand the cause of aliasing and be able to observe the aliasing effects.

Assessment:

- Your grade for this lab will be based on your ability to create and work with digital signals within `MATLAB` and the TMS 320 DSP processor, and on your reporting of the results.
- Clearly label all plots and their axes (points for style will be deducted otherwise).
- Please attend the lab section to which you have been assigned.
- You can complete this lab with one lab partner.
- By the end of the lab session, you must demonstrate to your TA the MATLAB code and the aliasing effects of the sampled radar signals on the TMS 320 DSP processor.
- All MATLAB source code must be submitted on Avenue to Learn by the end of your designated lab session.
- Each pair of students should complete one lab report together. The report has to be submitted by **11:59 pm on Oct. 20, 2025**. One of the group members can submit the source code and the report.

Pre-lab:

- Carefully read through this lab description so that you know what is required.
- Read through the lecture notes so that you know how to answer the questions.
- Familiarize yourself with the `MATLAB` commands that may be required for this lab – see the list at the end of this lab description for some hints.

Experiments:

1. **Introduction to convolution**
   (a) Create the discrete-time sequence:
   $$x[n] = u[n] - u[n-10]$$
   where $u[n]$ is the unit-step function (i.e., $u[n] = 1$ for $n \geq 0$ and $u[n] = 0$ for $n < 0$). You do not have to zero-pad $x[n]$ (i.e., the vector you have should contain no zero elements). Plot $x[n]$ using `stem()` function.
   (b) Now, convolve $x[n]$ over and over:
   $$\begin{aligned} a[n] &= x[n] * x[n] \\ b[n] &= a[n] * x[n] \\ c[n] &= b[n] * x[n] \\ d[n] &= c[n] * x[n] \end{aligned}$$
   You can use the `MATLAB` function `conv()` with proper input parameters.
   (c) Plot $a[n], b[n], c[n], d[n]$ using `stem()` function.
2. **Convolution of signals and system impulse responses:**
   (a) Load the supplied acoustic impulse response of a room into `MATLAB` using the command:
   `[impr,fs] = audioread('roomIR.wav');`
   This impulse response was obtained by creating an impulsive noise at one position in the room and recording (and digitizing) the sounds arriving at another position in the room.
   (b) Plot the impulse-response waveform `impr` using the `plot()` command and listen to it using the `soundsc()` command. What can you see and hear in the impulse response?
   (c) Load the supplied speech signal into MATLAB using the command:
   `[y,fs] = audioread('convolution.wav');`
   (d) Convolve the speech signal with the impulse response, and plot and listen to the resulting signal. Describe what you see and hear, comparing it to the original speech signal `y`. Explain what the convolved signal is physically equivalent to, according to the impulse-response theory.
3. **Image resampling and reconstruction**
   So far in this course, we have primarily been considering discrete-<u>time</u> signals and systems. However, many of the same principles apply to discrete-<u>space</u> signals and systems and can consequently be applied to images.
   The 2D image you will be using in this lab is a 256-level gray-scale portable networks graphic (.png) file - `KillarneyPic.png`
   (a) Import `KillarneyPic.png`, and report its size and bytes.
   (b) Most `MATLAB` functions require arrays to be of the `double` class. Convert the imported picture into a double-class variable using **im2double()** and display it in a `MATLAB` figure using the `imshow()` function.
   (c) You will be using this image to explore different resampling and reconstruction effects. Produce new images by resampling, and in some cases reconstructing, the original as follows:
      i. Impulse sampling at $\frac{1}{5}$ of the original rate (i.e., set 4 out of every 5 samples to zero).
      ii. Downsampling by a factor of 5 (i.e., discard 4 out of every 5 samples).

iii. Zero-order hold reconstruction (back to the original rate) from the downsampled image created in part (ii) above.

iv. First-order hold reconstruction (linear interpolation—back to the original rate) from the downsampled image created in part (ii) above.

You **cannot** use MATLAB builtin functions to do any of the above operations.
This resampling and reconstruction must be done both horizontally and vertically. Since the X and Y directions are ORTHOGONAL, you can resample and reconstruct in each direction INDEPENDENTLY. Ensure that you have the expected number of rows and columns after each operation.

(d) Of the 4 resampled/reconstructed images plotted in gray-scale, which one most closely resembles the original picture? Rank the images in order of FIDELITY compared to the original.

4. **FMCW Radar Signal Processing Project with TMS320C6713 DSK**

(a) **Introduction to FMCW Radar**
This section is an introduction to the frequency-modulated continuous wave (FMCW) radar signal processing project with the digital signal processing (DSP) board TMS320C6713 DSK. This project is divided into several sections and they are assigned to the lab 2 to lab 5 of the course. By the end of the course, you will be able to use the knowledge learned in the course to process a raw FMCW radar signal and extract useful information from it.
Application of FMCW automotive radar:

i. A frequency modulated continuous wave radar or FMCW radar is a special type of radar that measures both range (distance) and range rate (speed) of moving targets. It transmits a continuous signal with varying frequency (Figure 1). This is the widely used radar type for autonomous vehicles.

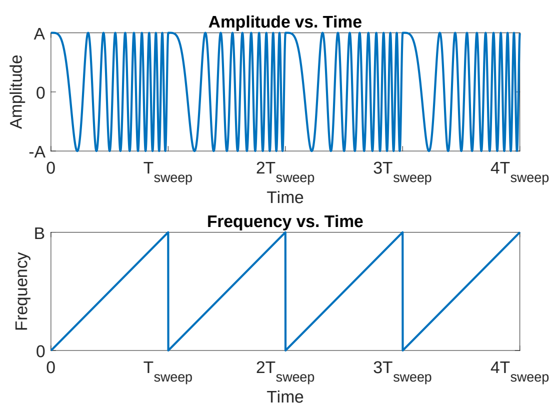

Figure 1: Transmitted signal of FMCW radar.

ii. Figure 2 illustrates a scenario of an ego-car (platform) installed with a front-facing FMCW radar. The radar detects the range and range rate of the cars ahead to make driving decisions (e.g., brake, lane change, etc.). FMCW radar usually gives angle information with multiple antennas in an array. In our project, for simplicity, only a single antenna is considered, and only the range and range rate of the targets can be obtained.

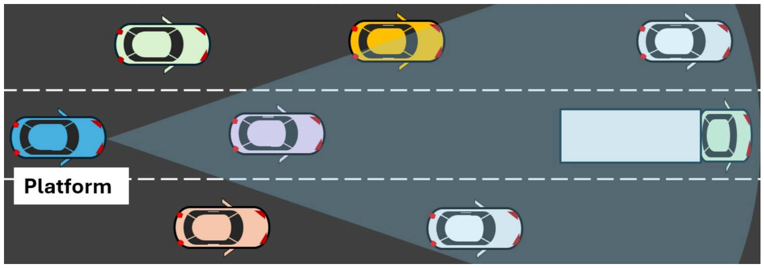

Figure 2: Driving scenario with a platform with front-facing radar.

Note that Figure 2 provides a general idea of the scenario and does not accurately describe the case that you will be working on. Every group have a unique data set representing different scenarios.
The antenna transmits the signal continuously, and the Electromagnetic (EM) wave hits objects in the field of view and then reflects back to the radar receiver. The signal shown in Figure 1 consists of multiple sweeps of the signals.
The bandwidth of the signal, B, is decided based on the required range resolution. In our data set, 1 m range resolution is considered. The number of samples, $N_{samples}$, in a single sweep with sampling frequency $f_{sampling}$ is determined by

$$N_{samples} = t_{sweep} \times f_{sampling} \tag{1}$$

Where t_{sweep} is the time of a single sweep, and it is determined by the maximum detection range and the speed of light.
The important parameters of the transmitted signal are summarized in the table below.

Parameter	**Value**
Time of a single sweep (t_{sweep})	3.3356 μs
Bandwidth (B)	150 MHz
Sampling Rate ($f_{sampling}$)	150 MHz
Number of samples per sweep ($N_{samples}$)	500
Number of sweeps (N_{Sweeps})	8

(b) **The DSP board**
The TMS320C6713 DSK board is a board designed by Texas Instruments specifically for high-performance digital signal processing applications. It has an on-board 16-bit audio stereo codec (the Texas Instruments AIC23B) that serves both as an A/D and a D/A converter.

(c) **Lab Procedure**

i. Several steps are required to compile, link and load the code for the DSP processors. These steps are outlined in the document "Y:/COE4TL4/Configuring CCS5.pdf".

- Create a project with the name "RadarProcessor" following the steps in "Configuring CCS5.pdf".
- In step Part 3, use "Y:/COE4TL4/RadarProcessor/src" as the source code directory.

ii. Include the following paths to "include search path"

- "Y:/COE4TL4/RadarProcessor/include"

- "Y:/COE4TL4/RadarProcessor/data/groupXX/no_barrier". Use your group number for "XX". Each group will use a different data set, and they are provided under this path.

  Note: please do not copy the files under "groupXX" and "include" folders to your project folder. Including paths is a better option so that your project can work even if the files under "groupXX" or "include" folder got updated.

iii. Compile and run the project "RadarProcessor". This program simply plot the pre-stored transmitted signal on the oscilloscope.

iv. Observe the signals on the oscilloscope, and explain why the waveform of the transmitted signal is different from the waveform in Figure 1.

v. Downsample the transmitted signal with a factor of 2 by modifying the interrupt function. Describe what you observed on the oscilloscope and explain.

vi. Modify the interrupt function to plot the received signal (without downsampling) instead of the transmitted signal. Observe the received signal and see if you can determine the number of targets in the coverage. You have to look for peaks to find the targets.

REPORT: The report should contain

- Any mathematical calculations or derivations carried out
- `MATLAB` plots of results with brief descriptions
- Answers to questions

You **do not** need to include the `MATLAB` code in the report. However, you have to submit the `MATLAB` code separately.

POTENTIALLY USEFUL `MATLAB` COMMANDS:

Note that this is not an exhaustive list! You are not required to incorporate all of these in your scripts.

help topic	helpwin	figure	plot	stem
histogram	subplot	hold on	xlabel	ylabel
legend	title	function	clear	close
clc	zeros	ones	cos	exp
abs	round	max	min	find
if	for	end	real	imag
angle	unwrap	phase	audioinfo	audioread
audiowrite	soundsc			