



**Due Date: 10 pm on 19 November 2021 (Friday)**

*Notes*

No late submission will be accepted (ZERO marks!)

This project accounts for 30% of your overall grade.

	Written Report	MATLAB Code	Project demonstration in week 13 lab sections	
Part I	3%	2%	10%	
Part II	3%	2%	10%	
Total	6%	4%	20%	<b>30%</b>

**Note: Lose 12% in each part if code is not submitted before the deadline => MATLAB code and Project demonstration will be zero marks! (Project demonstration will not be arranged)**

*Objectives*

Design a simple speech recognition program in part I and design a transmitter and a receiver in part II.

- **Part I: Write a simple program to distinguish between two simple words (speech sounds)**
- **Part II: Given a mysterious channel, design a transmitter and a receiver to minimize the bit error rate**
- **Submit your self-contained runnable MATLAB code using the MATLAB via virtual barn.**
- **Please test your code using the MATLAB via the virtual barn before submission.**  
**(note: some functions might not be supported! Toolbox is not allowed to use!)**
- **ALL the MATLAB codes should be uploaded to canvas in ONE zip file (project.zip).**

**Warning:**

**All submit codes and reports will be processing for similarity check.**

**All codes and reports you submit for grading must represent the results of your own independent efforts. You can discuss methodologies with your classmates, but you must do the work independently. Students who are involved (including the copier and the helper) in an incident of plagiarism will receive a failing grade for the course and the incident will be reported for appropriate disciplinary actions.**



This project is divided into two parts.

Part I: Design a simple speech recognition program (15%)

Part II: Design a transmitter and a receiver (15%)

---

## Part I: A simple speech recognition program

Submission (online)

1. How to design your speech recognition program – a written report ( $\leq 1000$  words)
2. Submit your self-contained runnable MATLAB code (should be able to run using the MATLAB via virtual barn)

The program can distinguish between two simple words spoken by you. Based on what you have learnt from the course, you will need to compute and interpret the signals in different domains. You can use FFT to extract useful information about the two different audio signals (two simple words, i.e., on/off), design a simple speech recognition program to distinguish the two simple words.

You can choose any **English** words with just **ONE** syllable.

During the project demonstration, TA will download your submitted codes, run your code, pick one of the pre-recorded wav files in the first run and let you speak the other 4 times (randomly picked by the TA). In each run, your code should give the word detected as an output.

Note: your program should be able to **load the correct wave file, record and play your pre-recorded / spoken sound** and **give the correct output**.

Marking criteria:

1. [3%] Written report ( $\leq 1000$  words): description of your program, technique/algorithm, testing, and accuracy
2. [2%] MATLAB code
3. [10%] Project demonstration: playback the spoken sound and display the correct output

**Note:**

**Output should not be provided with pre-set code (hardcode) or you will lose all the demonstration marks (lose 10%).**

Instructions:

You can use the following codes to record your two **ENGLISH** words and save them in two separate wav files.



MATLAB codes:

*% Pre-record your words and save into wav files:*

`Fs = 44100;`

`nBits = 16;`

`nChannels = 2 ;`

`ID = -1; % default audio input device`

`recObj = audiorecorder(Fs,nBits,nChannels,ID);`

`disp('Start speaking.')`

`recordblocking(recObj, 2);`

`disp('End of Recording.');`

`y = getaudiodata(recObj);`

`filename = 'word1.wav'; % use word2.wav for the second word`

`audiowrite(filename,y,Fs);`

*% load your wav file*

`[x, Fs] = audioread('word1.wav');` *% change to word2.wav for the second word*

`sound(x,Fs);` *% playback your pre-recorded sound*

NOTE: Please make sure that you have two different wav files and you can hear your pre-recorded sound from these two wav files.

You can use the following codes to let TA pick from recording live or choose one of the wav files.



MATLAB codes:

% Let TA pick from the pre-recorded wav file or record your spoken word live:

Fs = 44100;

nBits = 16;

nChannels = 2 ;

ID = -1; % default audio input device

prompt = 'type 0 (record live), 1 (wav file 1) or 2 (wav file 2) then hit enter => ';

x = input(prompt);

if x == 0

    recObj = audiorecorder(Fs,nBits,nChannels,ID);

    disp('Start speaking.')

    recordblocking(recObj, 2);

    disp('End of Recording.');

    y = getaudiodata(recObj);

elseif x == 1

    [y, Fs] = audioread('word1.wav');

    sound(y,Fs);

elseif x == 2

    [y,Fs] = audioread('word2.wav');

    sound(y,Fs);

else

    disp('Error!');

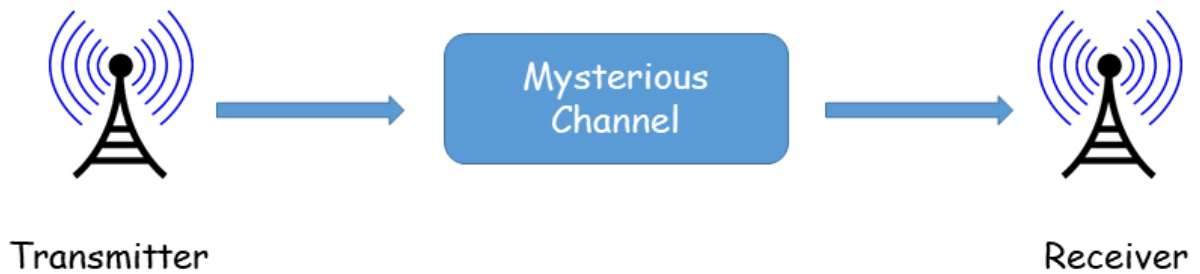
end

% y <= speech sound samples

Note: Please test your program and make sure that TA can pick one of the three choices.



## Part II: Design a transmitter and a receiver



### Introduction

By now, you have learnt many important concepts and theories about signals and communications. In this part, you will be given a mysterious communication channel. You will need to build your own transmitter and receiver to send and receive some text messages through this mysterious channel. The main goal is to minimize the bit error rate.

### Submission (online)

1. How to design your transmitter and receiver – a written report ( $\leq 1000$  words)
2. Submit your self-contained runnable MATLAB code (should be able to run using the MATLAB via virtual barn)

During the project demonstration, your **codes** will be run and tested with some unknown text messages. Your code should give the minimum bit error rate ( $\approx 0$ ).

### Marking criteria:

1. [3%] Written report ( $\leq 1000$  words): descriptions of your transmitter/receiver, problems/issues found, methodology used and testing method
2. [2%] MATLAB code
3. [10%] Project Demonstration: the bit error rate

**Note: Bit error rate or the number of error bits should not be provided with pre-set code (hardcode) or you will lose all the demonstration marks (lose 10%).**

Instructions: Download the txrx.p, compute\_BER.p files and a sample message (txtext.mat) file from the canvas page.

Open MATLAB and make sure that you can find these files in the 'Current Folder'.

You may reuse code or functions that you have previously developed during the lab work over the semester.



**Note:**

**The length of the transmitted samples is limited by 200000 samples.**

**The amplitude of the transmitted waveform should be between 0 and 1.**

**Exceed these two limits will lose all the demonstration points (10%).**

Instructions:

1. Double click txttext.mat files => **textmsg** should appear in the MATLAB Workspace  
**Note: the length of the textmsg will be at most 500 characters.**  
**Reserve the variable 'textmsg' in your code or your code cannot be run!**  
**DO NOT** load the txttext.mat inside your transmitter function!
2. Implement your transmitter
3. Use the **txrx** function to transmit and receive your waveform => **rx\_wave = txrx(tx\_wave);** **The given txrx function should be used in your transmitter.**
4. The **rx\_wave** will be saved to a rxwave.mat file.
5. Clear all the variable in your workspace >> **clear all**
6. Implement your receiver and run your receiver
7. Load your rxwave.mat file inside your receiver function so that **rx\_wave** will appear in the workspace
8. Find the received bit sequence (**rx\_bs**) and decode the bit sequence into text message (**rx\_msg**).
9. Use this function **compute\_BER(rx\_bs, rx\_msg)**. **The given compute\_BER function should be used in your receiver function.**

This **compute\_BER** function will display the transmitted and the received messages in text format; it will also compute the bit error rate; and calculate the number of error bits.  
Note: the transmitted messages should be stored in 'txttext.mat' when using the **compute\_BER** function.

**Note: The transmitter and receiver should be written/coded in two separate MATLAB functions/files.**

Calculations:

Bit Error Rate (BER): use **compute\_BER(rx\_bs, rx\_msg)**;

where **rx\_bs** => the decoded bit sequence received at the receiver

**rx\_msg** => the decoded text message from **rx\_bs**



## Project Demonstration:

Each student will be assigned a time slot (max. 10 minutes\*) => if you are late, you will need to wait until all the students have done their demonstrations!

**\*all parts should be able to run within this time slot, no extended time will be given!**

If you do not show up, you will lose 10% (Project Demonstration) in each part.

### Process:

TA will download your submitted codes

**Note: MATLAB code cannot be corrected or changed!  
Functions/files cannot be added!**

⇒ **NO changes can be made!**

**If your code give errors/cannot be run, you will lose the 10% (demonstration) in each part!**

Let TA know which files should be run in part I and part II

## Part I [10%]: let TA know which two words were chosen

- TA will run your code and pick one of the wav files (1 time)
- TA will let you speak your chosen words 4 times (randomly picked by the TA).
- In each run, your code should playback your pre-recorded sound/spoken sound and give the word detected as an output.

Note: your program should be able to **load the correct wav file, record and play your pre-recorded / spoken sound** and **give the correct output**.

### Possible mark deduction:

If your program **CANNOT** load the correct wav file/record your spoken sound, you will get zero marks in the demonstration (lose **10%**).

If your program cannot playback your pre-recorded / spoken sound, you will lose **1%** in each run.

If your program cannot show the correct output, you will lose **1%** in each run.



**Part II [10%]: TA will load an unknown text message (the same format with different text and length as the sample text message given) `textmsg` to your workspace**

- TA will run your transmitter function
- TA will clear all the variable in your workspace
- TA will run your receiver function
- TA will check the transmitted and received message and check the bit error rate and the number of error bits.

(Max. 2 runs and count the best run only)

Marks*	10%	9%	8%	7%	6%	5%	4%
BER#	0	> 0 – 0.0014	> 0.0014 – 0.0025	> 0.0025 – 0.0039	> 0.0039 – 0.005	> 0.005 – 0.0075	> 0.0075
No. of error bits.#	0	1-5	6-10	11-15	16-20	21-30	> 30

\*not including the mark deduction

#whichever lower

Possible mark deduction:

**Your code should follow the limit as stated.**

**Exceed the specified limit of either one will lose all the demonstration points (10%).**

**Project Report:**

Please write your report including part I and part II and submit in **ONE pdf** file.

**NOTE: In each part, only the first 1000 words will be marked!**

**Part I: Design a simple speech recognition program**

- descriptions of your program;
- technique/algorithm;
- testing and accuracy

**Part II: Design a transmitter and a receiver**

- descriptions of your transmitter/receiver;
- problems/issues found;
- methodology used and testing method





Submission Summary:

**Due Date: 10 pm on 19 November 2021 (Friday)**

Project report – PDF format (ONE FILE)

Self-contained runnable MATLAB codes – in ONE zip file – project.zip (including all the necessary functions/files)

--- End of Project ---