ES3C5: Signal Processing
Coursework Lab Assignment Recording Supplement
v1.2

Module Leader: Dr Viji Ahanathapillai (D211,
viji.ahanathapillai@warwick.ac.uk)

Update History:

- **v1.1 2023-11-13**: Updated note regarding copying of template code.
- **v1.2 2024-11-4**: Updated document to make 5 second recording.

1 Introduction

This document provides supplementary guidance on making a 5-second audio recording for ES3C5.

Health warning: please take care when listening to your data after you have changed it (e.g., listening to filter output). While the processing specified for the assignment should not make your audio significantly louder than a conversational level, a typo or other mistake in your code could manipulate your data in unintended ways and make your audio samples dangerously loud. Take precautions when listening to processed audio signals for the first time and whenever changes are made to the digital systems manipulating your audio.

2 Data Recording

This guidance will go through how to record an audio file to use in MATLAB. There are 4 possible approaches:

1. Record to MATLAB using your own personal computer and microphone (preferred method 1a). This method is preferred because you record directly into MATLAB and do not need to borrow equipment for your recording.
2. Record to MATLAB at a lab computer using your own personal microphone and headphones (preferred method 1b). This method is preferred because you record directly into MATLAB and do not need to borrow equipment for your recording.
3. Record to your smartphone and then import to MATLAB (preferred method 2). This is the 2nd preferred method because it is harder for the staff to support and you need to import your data. However, you still do not need to borrow equipment for your recording.

4. Record to MATLAB at a lab computer using borrowed equipment (**not** preferred method). We have a limited number of microphones that can be used for recording in the lab. This is **not** preferred because we do not have enough equipment for everyone to have their own set. You may need to wait until a microphone becomes available.

No matter which method you use, you will need to generate the same kind of data and save it to the data file `u<ID>_lab_Audio.mat` (where `<ID>` is your student number).

2.1 Specification of the Data to Record

Before elaborating on the different recording methods, we first establish what you need to record as this will be common for all methods. The (imaginary) premise is that the University has decided to add voice codes as an option in two-factor authentication, so you can state all the digits of your student number in order to access University systems. You will record your voice saying all the digits of your student number. This recording must be recorded to a **single channel (mono)** at a **sampling frequency of 22.05 kHz** and **must be exactly 5 seconds long**. If the original recording is longer than 5 seconds, then you must shorten it to 5 seconds before saving to `u<ID>_lab_Audio.mat`.

2.2 Recording with MATLAB

If you are going to record with MATLAB, then the microphone should be plugged in and configured **BEFORE OPENING MATLAB**. Otherwise, you will likely need to restart MATLAB, because it needs to detect input devices when it launches. If you are going to record with a smartphone, then you can skip to the following subsection ("Recording with a Smartphone). Otherwise, follow the steps for your recording use case:

1. **Built-in Microphone:** If your microphone is already configured on your computer (e.g., a built-in laptop microphone), and you already know that it works, then you should not need to change any settings and you can proceed to open MATLAB. However, if you find that your recording is very quiet, then you may want to adjust your microphone settings to increase the gain.

2. **Lab Computer:** If you are using a lab computer, then you should speak with a lab demonstrator or the module leader. We will have a dedicated lab computer for you to make your recording. Guidance will be provided in the lab.

3. **Personal Desktop:** If you are using a standalone microphone on your own computer, and you have previously used the microphone on that computer, then you should be able to plug it in as usual and then open MATLAB. If you have not previously used the microphone, then you may need to configure your audio settings to detect the microphone and have sufficient gain.

Now that MATLAB is open, complete the following steps to make your recording:

1. Check that your audio is on by clicking on the speaker icon. Be sure that it is not on mute. If you are wearing headphones, then set the volume level to 20 % or less. You can increase it later if needed, but you should be able to hear normal audio outputs at this level.

2. You should become familiar with the documentation for the following MATLAB functions (no toolboxes required):

- `audiorecorder`
- `recordblocking`
- `play`
- `getaudiodata`

3. From the MATLAB command window or a separate script (**not** `u<ID>_lab.m`!), run the following steps of code in order:

   (a) You will create an audio "object" in MATLAB. This includes the settings that we will record with (i.e., sample rate and quantisation). To capture enough detail without using too much memory, we will sample at 22050 Hz and with 8 bits (1 byte) per sample. We achieve this with the following:

   ```
   recObj = audiorecorder(22050, 8, 1); % 22.05kHz, 8 bits per
       sample, 1 channel
   recObj.StartFcn = 'disp(''Start speaking.'')';
   recObj.StopFcn = 'disp(''End of recording.'')';
   ```

   This code also adds displays so that you will know when recording starts and stops. *Note: The template code shown is intended to be copied directly into MATLAB. However, be sure that "sample, 1 channel" does not appear on its own line, and you will need to replace the apostrophes as shown with single quotes (below the @ symbol on a standard UK keyboard).*

   (b) You will use the audio object to record your voice stating the specified message. You will give yourself 5 seconds and you can repeat this step as many times as you need to be satisfied with the quality of the recording, but **DO NOT** change the recording time of 5 seconds. Run the following code:

   ```
   recordblocking(recObj, 5); % Record for 5 seconds (do not change)
   ```

   The `recordblocking` function prohibits other commands from running while recording. Alternatively, the `record` function enables recording while allowing code to continue running.

   (c) You will play back the recording. For now we will do this with the `play` function:

   ```
   play(recObj); % Play recorded audio
   ```

   Ensure that you can clearly hear your voice and that the audio does not break up. Some interference (e.g., from background noise in the lab) is OK (and expected). If you wish to re-record, then you can go back to the previous step to write over the recording. If you don't hear anything, check that the PC audio is not muting your speakers or headphones and that the microphone settings are configured properly.

   (d) You need to convert the audio recording into a format that you can readily use for signal processing. You can extract the audio to a vector of double floating point numbers as follows:

   ```
   audioRaw = getaudiodata(recObj); % Extract audio to vector
   ```

   (e) You can now use the vector `audioRaw`, which has 110250 elements (why?), in any regular MATLAB function. But most importantly you should **SAVE** it. Create a .mat file to store this vector:

```
save('u<ID>_lab_Audio.mat', 'audioRaw');
```

where <ID> is your student number. Backup the file u<ID>_lab_Audio.mat by *multiple* means (e.g., email it to yourself, put in on a cloud service, copy it to a USB drive, etc.). **Do not lose this file** as you need its contents and you need to make changes to it.

2.3 Recording with a Smartphone

If you are going to record with a smartphone, then please read this subsection. If you are going to record with MATLAB, then please read all of the instructions in the previous subsection "Recording with MATLAB".

There are many free apps that can record audio to a smartphone. Try to use one that lets you set the audio quality as this will help to control file sizes. A recommended free app for Android is "Super Recorder" by HappyBees (unfortunately we do not have an iPhone recommendation as of this writing). It is straightforward in "Super Recorder" to record at 22.05 kHz over 1 channel (mono) at 128 kbps (a suitable data rate) for at least 5 seconds, export the audio file in .aac format to a cloud storage service such as OneDrive or Dropbox, and then import directly into MATLAB. Once the .aac file is in your MATLAB working directory, double click the file from the "Current Folder" interface in MATLAB. There should be a `data` vector and a scalar `fs` containing the audio and sampling frequency, respectively. Confirm that `fs` is exactly 22050 and that `data` has at least 110250 elements (i.e., it is 5 seconds in length). Copy the `data` vector to `audioRaw` (case-sensitive). Save the vector as follow:

```
audioRaw = audioRaw(1:110250); % Trim recording to 5 seconds
save('u<ID>_lab_Audio.mat', 'audioRaw');
```

where as always <ID> is your student number. Backup the file u<ID>_lab_Audio.mat by *multiple* means (e.g., email it to yourself, put in on a cloud service, etc.). **Do not lose this file** as you need its contents to complete and submit the assignment.

2.4 Listening to Your Recording

Now that you have your recording in MATLAB as a vector of double precision floating point numbers (whether you recorded using MATLAB or a smartphone), you can playback the vector using the `sound` function. If it is your first time listening to the data, consider keeping your headphones away from your ears in case the volume is too high. You need to provide the sampling frequency to `sound` as the second argument (also try listening if you change the frequency to some other value):

```
sound(audioRaw, 22050); % Play audio from vector
```