

ES3C5: Signal Processing
Coursework Lab Assignment Briefing Sheet v1.0
Submission Deadline:
12pm (noon) on Thursday 1 December 2022

Module Leader: Dr Adam Noel (D205, adam.noel@warwick.ac.uk)
Autumn 2022

Target Completion Time for Average Performance: 14–18 hours (not including time to learn the corresponding lecture material)

Note: This assignment is worth 30 % of your module mark.

Any changes made to this briefing will be listed here.

Health warning: this assignment involves the generation and filtering of audio signals. While the processing specified in this assignment should not make your signal louder than a conversational volume, 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.

1 Purpose

The following module learning objectives are applicable for this assignment:

- **Model** signals, filters and processes using computer packages.
- **Design** signal processing systems.
- **Apply** signal processing systems to classify signals and extract information.
- **Evaluate** signals and systems using laboratory test and measurement equipment.

The focus of this assignment is Parts 2 and 3 of the module, in particular content related to **digital filtering** and **signal estimation**.

2 Sessions and Support

There are **2 sets of in-person lab sessions** as scheduled in Tabula to be held in A2.05a/b:

- Friday 18 November and Friday 25 November 1:00pm–3:00pm
- Friday 18 November and Friday 25 November 3:00pm–5:00pm

These sessions will begin and end promptly on time and you must attend your own sessions.

If you would like to change either of your sessions: You must find someone to swap with and both of you must email the module leader to confirm that you agree. Otherwise, your entry at the other session may be refused.

Extra support can be obtained during the module leader’s support and feedback sessions in D2.05:

- Mondays 1:00pm–1:50pm
- Tuesdays 3:30pm–4:20pm

Finally, there is a **support forum for all Lab Assignments** ([link](#)) under the “Key Module Resources” heading on Moodle. Emailed questions will generally be re-directed to the forum unless the question is truly individual.

Email/forum cutoff: In fairness to all students, the module leader will not respond to Assignment queries (either forum or email) sent after 10pm on Wednesday 30 November until after the submission deadline, unless there is an internal (i.e., university-side) problem with submissions.

3 Academic Misconduct

Your submission **must** be your own. **Plagiarism of any kind is not acceptable and can result in a mark of zero.** We have previously identified many cases of academic misconduct (including plagiarism) that have resulted in penalties for the assignments in this module.

It is expected that you will discuss the assignment with your classmates. However, you should keep the following **best practices** in mind to avoid having your work flagged for academic misconduct:

- Write your own code, comments, and written answers. Start from the provided templates.
- You should not adapt code from external sources, but if you do then they should be cited.
- Do not write your implementation together with another student and submit it as your own work. This includes talking through your implementation line-by-line. Such cases count as academic dishonesty.
- Do not share your assignment files with another student. Allowing your work to be copied, even if it is modified, counts as plagiarism.
- Do not use the files of another student or other source as a starting point for your own submission. This still counts as plagiarism and can be flagged by our academic integrity processes, even if you make changes!

4 General Instructions

Heads up regarding names: These instructions include *precise* case-sensitive guidance regarding names of files, directories, functions, and variables. These must be followed *exactly* so that the marking scripts run correctly. **There will be points deducted for any deviations.** This may seem pedantic, but precise naming is standard practice in programming and is essential for writing code that other people can rely on.

The instructions to complete this lab assignment are as follows:

1. Download the zip file [ES3C5_2022_2023_lab_data.zip \(link\)](#) from the ES3C5 Moodle Page under the Assessment Information heading. Extract **four** files from the zip file:

- ES3C5_2022_2023_lab_template.m
- ES3C5_2022_2023_lab_template.docx
- u<ID>_lab.txt
- ES3C5_2022_2023_lab_recording.pdf

where <ID> is your student number.

2. Download your signals file `u<ID>_lab_signals.mat` from the [ES3C5 Coursework Lab Assignment Signals Files page \(link\)](#) under the Assessment Information heading, where <ID> is your student number.

- Note: This is a separate download so that you do not need to download everyone else's signals (over 100 MB).

3. Rename the template files as follows:

- ES3C5_2022_2023_lab_template.m to `u<ID>_lab.m` (where <ID> is your student number; don't forget the "u" prefix here!) and place the file in your MATLAB working directory (this should be a directory where you have write privileges on your computer).
- ES3C5_2022_2023_lab_template.docx to `u<ID>_lab.docx`.

4. Open `u<ID>_lab.m` and rename the function title in the first line from `ES3C5_2022_2023_lab_template()` to `u<ID>_lab()` (where <ID> is your student number). Do not change the output argument **Answers** or any of the code in the top-level function. You will be entering all of your code in the *subfunctions*. Also, do not change the output arguments of the subfunctions or add any input arguments.

5. Complete the prescribed steps in the *Recording Supplement* ([ES3C5_2022_2023_lab_recording.pdf](#)) to record the audio data that you will use to answer the problems in Question 1. This data will be stored to the file `u<ID>_lab_Audio.mat`.

6. Complete the problems listed below in Section 6 and refer to `u<ID>_lab.txt` for the functions and parameter values that you must use. You will also need to load signals from `u<ID>_lab_signals.mat` and there is already a call to do this as needed in the template code. Enter the required code in the corresponding *subfunctions* in `u<ID>_lab.m`; you should

not need to write your own functions. The *code must calculate the answers using **your** assigned parameters unless otherwise indicated.* Take care to use parameters with the correct units for each MATLAB function that you call. For example, you may need to do conversions between radian and Hz frequencies, or convert gains between linear and dB forms.

7. When a question includes plotting a figure, please account for the following:
 - (a) Every figure should be initiated by creating a new blank figure and your code should not close any figures.
 - (b) The plot function calls must be *inside* the subfunction code for the question, but details such as labels can be added after your solution function runs.
 - (c) Carefully consider appropriate parameter ranges and axis scaling, e.g., is it better for a given axis to be linear or logarithmic?
 - (d) **ALL** submitted plots must have labelled axes.
 - (e) Figures must include your student number and question number at the top.
 - (f) Each figure requested should be placed in u<ID>_lab.docx as a png image.
8. Include readable comments in your code that briefly summarise your work, especially critical steps. See the subfunction Q0 below (and in u<ID>_lab.m) as a template example:

```
function c0 = Q0()
% Assign answer to c0 (double value)

% Define triangle lengths
a0 = 2; % 1st side
b0 = 1; % 2nd side

% Find length of hypotenuse
c0 = sqrt(a0^2 + b0^2); % Pythagorean theorem to find 3rd side

end
```

9. Run your code file *from the MATLAB command line* and write the output to a structure named **Answers**. Save **Answers** to a mat file named u<ID>_lab_Answers.mat. You can do these steps with the following code:

```
Answers = u<ID>_lab; % Runs assignment code
save('u<ID>_lab_Answers', 'Answers') % Writes output to file
```

10. Place your code file u<ID>_lab.m, signals file u<ID>_lab_signals.mat, audio file u<ID>_lab_Audio.mat, output file u<ID>_lab_Answers.mat, and document u<ID>_lab.docx into **1 folder** named u<ID>_lab, and compress it into a single zip-file called u<ID>_lab.zip (where <ID> is your student number). The zip-file is what you submit to Tabula.

- Be sure to place the files into the named folder *before* zipping. This ensures that your code is nested in its own subdirectory when unzipping.

5 Recording Supplement

Refer to `ES3C5_2022_2023_lab_recording.pdf` for instructions on recording and saving your audio.

5.1 Specification of the Data to Record

The (imaginary) premise is that, in the grand tradition of number-based songs (e.g., 867-5309, 46 and 2, One), you are recording your student number to use as a sample in a song. You will record your voice saying (or singing?) only “<ID>,” i.e., **all 7 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`. Refer to `ES3C5_2022_2023_lab_recording.pdf` for additional guidance.

6 Questions (Total: 100 points)

Please answer all of the following questions with reference to `u<ID>_lab.txt` for your specific parameter values. Remember that all code to solve the following problems must be included in `u<ID>_lab.m`. Written answers in `u<ID>_lab.docx` are required where specified, including copies of output figures.

25 points are designated as being part of “**First Class Problems**”. These questions are deliberately more challenging than the other assignment questions. While support can be sought to clarify what these questions are asking, instructors and lab demonstrators will not provide guidance on *how* to answer them.

Up to **30 Points** in total *can be deducted* for poor coding style. Poor style includes:

- Displaying messages to the command line (e.g., by omitting semi-colons for output suppression)
- Insufficient use of code comments to explain key steps and demonstrate *your* understanding of your implementation. However, succinctness is still appreciated.

Here are the questions:

1. **(39 Points)** This question concerns the generation and processing of an audio signal. Please complete the following:
 - (a) (4 points) Ensure that you have completed the appropriate steps in the recording guidance and have saved your audio recording vector with the correct specification to the file `u<ID>_lab_Audio.mat`. It is needed to proceed with Question 1. Load the recording into the `Q1` subfunction. The subfunction has template code that you just need to update to do this.
 - *Note: There is nothing to calculate here; this part is being evaluated by the correctness of your audio and loading it into the `Q1` subfunction.*

- (b) (9 points) Design an FIR low pass digital filter for your audio signal with the windowing method. Complete the following:
- Using the window information tables in the lecture slides or notes, determine the shortest window type and length that will give a pass band ripple parameter below `rp` dB at frequencies less than `fp1`, and that has a stop band defined by frequencies above `fstop1`. Remember that the sampling rate being used is $f_s = 22.05$ kHz. Assign window type as a *case-sensitive character array* (either `'rectwin'`, `'hann'`, `'hamming'`, or `'blackman'`) to `Q1.w`. Assign window length L to `Q1.L`.
 - Note: you do **not** need to trial different parameters to meet the specification; you can assume that the tables are accurate.*
 - Use the `fir1` function and the corresponding window function to determine the *unscaled* impulse response $h_b[n]$. Use a cutoff frequency that is exactly halfway between `fp1` and `fstop1`. Assign FIR impulse response vector $h_b[n]$ to `Q1.hb`.
 - Use your FIR low pass digital filter with the `filter` function (and not any function variant). Assign the low pass filter output to `Q1.audioLow`.
- (c) (8 points) You wish to make your voice recording sound more interesting. Consider applying a reverberation (reverb) effect that will simulate being in a large room. You will use an all-pass reverberator with the following z-domain transfer function:

$$H_c(z) = \frac{-A + z^{-D}}{1 - Az^{-D}},$$

where A is the relative strength of the reverb effect and D is the baseline delay (measured in number of *samples*) associated with the reverb. Complete the following:

- Using the reverb strength `A1` and the delay *time* `t1` (in *seconds*), determine the corresponding transfer function. Assign the delay D (in number of *samples*) to `Q1.D`. Scale the b and a vectors for $H_c(z)$ such that the first element of a is exactly 1. Assign b and a vectors for $H_c(z)$ to `Q1.b` and `Q1.a`, respectively.
 - Pass your filtered audio signal `Q1.audioLow` (**not** the original audio) through the reverberator using the `filter` function. Assign the reverberated output to `Q1.audioReverb`.
- (d) (6 points) Your recording has become contaminated by a noise source. The contents of the noise source are in the vector `n1` in your signals file `u<ID>_lab_signals.mat`.
- Add the noise source vector `n1` to your reverberated output `Q1.audioReverb`. Assign the noisy signal to `Q1.audioNoisy`.
 - Calculate the fast Fourier transform (FFT) of `Q1.audioNoisy`. Assign the FFT to vector `Q1.FFT`.
 - See the Extra box 14.2 “Double-Sided vs Single-Sided FFT” in the lecture notes to be aware that the MATLAB help examples present the FFT different from how you are expected to calculate it. To answer this question you must find the double-sided FFT (which we cover) and not the single-sided FFT (which we do not cover).
 - Plot the *magnitude* of this FFT output as a function of Hz frequency (i.e., a domain of [0,22050] Hz should be visible). Ensure that the figure is sufficiently labelled

as directed by the general instructions, including a label for the largest noise frequency component. Save the FFT magnitude plot as a png file and include it in `u<ID>_lab.docx`.

- (e) (12 points; “**First Class Problem**”) Design an FIR filter to remove the largest noise frequency component.
- In no more than **4** lines of text in `u<ID>_lab.docx`, describe and justify a target FIR filter performance specification. The specification should be sufficient to assess a candidate design by inspection of the magnitude of its frequency response.
 - Design a filter that meets your target specification. Assign the filter impulse response to `Q1.he`.
 - Test your filter design. Pass your noisy audio signal `Q1.audioNoisy` through your designed filter using the `filter` function. Assign the filtered output to `Q1.audioNotNoisy`.
 - Create a suitable plot that assesses the test of your design. Ensure that the figure is sufficiently labelled as directed by the general instructions. Save the plot as a png file and include it in `u<ID>_lab.docx`.
 - Briefly (in no more than **2** lines of text) explain in `u<ID>_lab.docx` whether your plot validates the design.

2. (21 Points) This question concerns interference and aliasing.

You are an engineer at SpaceZ and are receiving signals from the new deep space satellite deployed by SpaceY. You are tuning your terrestrial hardware using transmissions from the satellite and have collected a sequence of time-series data `y2` that was sampled at frequency `f2s`. You are expecting to observe a signal of frequency `f_satellite` from the satellite.

- (a) (8 points) Determine the spectra associated with `y2` as follows:
- Calculate the fast Fourier transform (FFT) of `y2`. Assign the FFT output to vector `Q2.FFT`.
 - See the Extra box 14.2 “Double-Sided vs Single-Sided FFT” in the lecture notes to be aware that the MATLAB help examples present the FFT different from how you are expected to calculate it. To answer this question you must find the double-sided FFT (which we cover) and not the single-sided FFT (which we do not cover).
 - Each element in the FFT is associated with a frequency. Create the vector of frequencies associated with `Q2.FFT`. Assign the vector of frequencies to `Q2.fRange`.
 - Plot the *magnitude* of `Q2.FFT` as a function of Hz frequency (i.e., a domain of `[0,f2s)` Hz should be visible). Ensure that the figure is sufficiently labelled as directed by the general instructions. Include labels that identify **all** spectral peaks associated with target frequency `f_satellite`. Save the FFT magnitude plot as a png file and include it in `u<ID>_lab.docx`.
- (b) (5 points) It should be apparent from your FFT magnitude plot in Q2(a) that there are unexpected signal components at other frequencies. You can assume that there is no aliasing. Thus, based on the plot in Q2(a), at what distinct frequencies do you expect there to be interfering sinusoids? Briefly (in no more than **3** lines of text) explain

your approach in `u<ID>_lab.docx`. Assign the unexpected frequencies (each rounded to nearest Hz) to row vector `Q2.fInter`.

- (c) (8 points; “**First Class Problem**”) Suppose that one of the unexpected signal components in `y2` could be an alias. Complete the following:
- Calculate the lowest frequency signal (in Hz and rounded to the nearest Hz) that could lead to any of the unexpected signal components as an alias. Briefly (in no more than **3** lines of text) explain your approach in `u<ID>_lab.docx`. Assign this frequency (in Hz) to `Q2.fFirst`.
 - Create a signal with frequency `Q2.fFirst` and plot its FFT to demonstrate that it would result in the alias as predicted. Ensure that the figure is sufficiently labelled as directed by the general instructions. Save the FFT magnitude plot as a png file and include it in `u<ID>_lab.docx`.
 - Briefly (in no more than **3** lines of text) propose a suitable solution in `u<ID>_lab.docx` to prevent this signal from creating the alias without having any other impact on `y2`.

3. (20 Points) This question concerns estimation of an unknown signal.

You are a control engineering consultant and are visiting the factory of NewWave Mobile to inspect a heating chamber that will be used to stress-test their new drone communication chips. Based on the chamber specification, you expect that the transient heating profile will behave according to the general equation provided in `u<ID>_lab.txt` where `y3(t)` is the temperature in degrees Celsius and `w3(t)` is the noise signal.

You have collected noisy thermometer readings in column vector `y3` in your signals file `u<ID>_lab_signals.mat`. Your data were collected starting from time $t = 0$ at the constant interval `Ts3` in your parameter file. Please complete the following:

- (3 points) Construct the linear model observation matrix Θ needed to estimate the unknown parameters `A3`, `B3`, and `C3` (in that respective order). Assign Θ to `Q3.0bs`.
- (2 points) Apply linear model estimation to estimate the unknown parameters `A3`, `B3`, and `C3`. Assign these three constants in the same order to the 3×1 column vector `Q3.param`.
- (1 point) Use the parameter estimates to predict the temperature *for the same times* at which the sensor values were taken. Assign the predicted temperatures to column vector `Q3.yHat`.
- (1 point) Calculate the mean square error of the predicted temperatures. Assign the scalar mean square error to `Q3.mse`.
- (5 points) Compare the measured temperatures and predicted values in the time domain as follows:
 - Compare the predictions with the actual sensor values on the same axes in a single time-domain plot, where the y-axis is the temperature in degrees Celsius. Save the plot as a png file and include it in `u<ID>_lab.docx`.

- ii. Briefly (in no more than **2** lines of text) comment in `u<ID>_lab.docx` on the possible accuracy of the model as shown in this plot.
- (f) (8 points) Compare the spectra of the temperature sensor readings with those of the estimated temperatures, as follows:
- i. Calculate the fast Fourier transform (FFT) of `y3`. Assign the FFT output to vector `Q3.yFFT`.
 - See the Extra box 14.2 “Double-Sided vs Single-Sided FFT” in the lecture notes to be aware that the MATLAB help examples present the FFT different from how you are expected to calculate it. To answer this question you must find the double-sided FFT (which we cover) and not the single-sided FFT (which we do not cover).
 - ii. Each element in the FFT is associated with a frequency. Create the vector of frequencies associated with `Q3.FFT`. Assign the vector of frequencies to `Q3.fRange`.
 - iii. Calculate the fast Fourier transform (FFT) of `Q3.yHat`. Assign the FFT output to vector `Q3.yHatFFT`.
 - iv. Plot the *magnitude* of *both* FFT outputs as a function of Hz frequency (i.e., a domain of $[0, 1/Ts3)$ Hz should be visible). Include **both** sets of spectra on the **same** axes. Ensure that the figure is sufficiently labelled as directed by the general instructions, including labels that identify all relevant spectral peaks. Save the FFT magnitude plot as a png file and include it in `u<ID>_lab.docx`.
 - v. Briefly (in no more than **3** lines of text) comment in `u<ID>_lab.docx` on the possible accuracy of the model as shown in this plot.

4. (20 Points) This question concerns the estimation of FIR filter coefficients.

You work for NewWave Mobile and you are performing wireless propagation testing in Coventry where a new 5G base station will be installed. Since there are many cloudy days, you are measuring the impact of signal reflections when the base station is communicating with a mobile user. You have determined that there are four primary signal paths of interest: the direct path from the base station and three paths that reflect off of the clouds. The reflections arrive later than the direct signal, such that the total received signal voltage `y4[n]` is the output of a noisy difference equation of the form provided in `u<ID>_lab.txt`, where `w4[n]` is the noise signal.

You have collected samples of the received signal voltage (in Volts) `y4[n]` based on the transmission of a pseudo-random pilot sequence `x4[n]`. These are given as the column vectors `y4` and `x4`, respectively, in your signals file `u<ID>_lab_signals.mat`. Please answer the following:

- (a) (4 points) Construct the linear model 4-column observation matrix Θ needed to estimate the unknown FIR filter coefficients `A4`, `B4`, `C4`, and `D4` (in that respective order). Assign the 4-column Θ matrix to `Q4.Obs`.
- (b) (3 points) There was a sensor malfunction in your recording equipment while measuring `y4[n]`, such that the variance of `w4[n]` increased by a constant factor of `k4` for the last `s4` samples (as described in `u<ID>_lab.txt`). Determine an appropriate weight column

vector $W[n]$ that can be used to perform weighted least squares estimation using the entire data set. Assign $W[n]$ to `Q4.W`.

- (c) (2 point) Apply linear model estimation (using *weighted* least squares) to estimate the FIR filter coefficients `A4`, `B4`, `C4`, and `D4`. Assign these four constants in the same order to the 4×1 column vector `Q4.param`.
- (d) (1 point) Use the FIR filter coefficient estimates to predict the voltage *for the same pilot input sequence* `x4[n]`. Assign the predicted voltages to column vector `Q4.yHat`.
- (e) (5 points) Compare the measured voltages and predicted values in the time domain as follows:
 - i. Compare the predictions with the actual wireless sensor voltages on the same axes in a single time-domain plot, where the x-axis is the sequence index and the y-axis is the voltage. Save the plot as a png file and include it in `u<ID>_lab.docx`.
 - ii. Briefly (in no more than **3** lines of text) comment in `u<ID>_lab.docx` on the possible accuracy of the predictions as shown in this plot.
- (f) (5 points; **“First Class Problem”**) Suppose that the specified noise increase factor `k4` was measured incorrectly. Assuming that `Q4.yHat` is correct, estimate the correct value of `k4` to the nearest 0.1. Briefly (in no more than **3** lines of text) explain your approach in `u<ID>_lab.docx`. Assign your estimate of `k4` to `Q4.k4Est`.

7 Submission of Deliverables

Place your files into a folder named `u<ID>_lab` (where `<ID>` is your student number). Zip the folder and call it `u<ID>_lab.zip`. The folder should contain:

- `u<ID>_lab.m` (where `<ID>` is your student number), containing your code.
- `u<ID>_lab_signals.mat` (where `<ID>` is your student number), containing your assigned signals.
- `u<ID>_lab_Audio.mat` (where `<ID>` is your student number), containing your original audio recording.
- `u<ID>_lab_Answers.mat` (where `<ID>` is your student number), containing the answers generated by your code.
- `u<ID>_lab.docx` (where `<ID>` is your student number), containing your written answers and all of your plot figures.

Be sure that `u<ID>_lab.m` runs correctly with no other custom files required.

You must submit the zip-file to **Tabula**. It is *your* responsibility to double-check your submission after it is uploaded to ensure that the correct files were included. You may refer to the [Wrong File Upload Policy](#) as published by the ESO.

Your assignment **must be submitted by 12pm (noon) on Thursday 1 December 2022**, otherwise standard late penalties will be applied. Applications for extension without penalty must be made via Tabula.

8 Notes on Marking

Each problem will be marked according to:

- Correctness of the audio recordings. **Your submitted code must not play any audio when it runs.**
- Correctness of the calculated answers, figure contents, and written answers. Please note that incorrect answers for earlier problems will not automatically penalise you in later problems, e.g., using an incorrect answer from part (a) in your solution to part (b) can still earn points.
 - Written answers will not be read beyond the number of lines specified in the question. A line of math will read as 1 line.
- Implementation of the code used to arrive at your answers. Any relevant calculations should be performed within your code.
- Presentation of labelled plots. Take care that plots include all information requested, *including your student number and the question number in **every** figure title.*
- Code running efficiently as expected by the marking scripts. This includes using correct naming, correct directory structure, and your code running without errors. Expect that any lines of code causing an error will be commented out and then the rest of the code executed as is.

Common reasons for execution errors to occur:

- Trying to use a MATLAB toolbox that we do not introduce in the module.
- Trying to read from a parameter that does not yet exist. This is most likely if you try implementing in a script and then copy your solution into the subfunctions.
- Making changes to the format of the output argument structure.

There is a total of **100 points** available. Up to 30 points can be deducted for poor coding style (i.e., if you have no suppression of output to command window or make insufficient use of comments).

9 Assignment Weighting

This assignment is worth **30 %** of your module mark.