School of Engineering
The University of Warwick

General Assignment Information

Module Code	ES3C5
Module Title	Signal Processing
Assignment Title	Coursework Lab Assignment
Assignment Weighting / Credits	30%
Submission Deadline	Week 16, Thursday, 22 January 2026, 12 noon (midday)

Intended Learning Outcomes (ILOs) Assessed

Intended Learning Outcome(s) (ILOs)	**Task**
LO2: Apply signal processing systems to classify signals and extract information.	Use coded signal processing systems to process data
LO3: Critique practical issues behind signal processing and information retrieval	Provide comments/ critical analysis and reasoning behind the parameter selections and result observations
LO4: Design signal processing system to meet a specification	Propose system performance specifications to meet a desired signal processing aim, then propose and implement signal processing systems to meet the defined specifications.
LO5: Model signals, filters and processes using computer packages	Write code to represent signals and systems and perform signal processing tasks. Plot pertinent results.
LO6: Measure and analyse real-world signals	Collect real-world data to use as an input signal to the digital systems that you implement.

Notes:

- A total mark below 30% indicates that the ILOs have not all been met at threshold level;
- A total mark in the range 30 – 38% indicates that the ILOs have all been partially met to at least threshold level;
- A total mark of at least 40% indicates that the ILOs have all been met.

Submission Details

The submission details for this assignment are:

- **Deadline:** 12pm Noon Week 16, Thursday, 22 January 2026, 12 (midday)
- **Method:** Tabula
- **Format of submission:** *5 files including a MATLAB code m-file and a written report docx-file (details in the briefing)*
- **Submission length:** The written report should be no more than 6 pages in length. Individual written questions have maximum answer lengths (e.g., 5 lines of text); longer answers will only be marked up to the maximum indicated.
- **Formatting instructions:** A template m-file and docx-file are provided and must be used. Answers in the docx-file should be written in 11-point Calibri font with 25 mm left and right margins.

Note: Submissions should be of an appropriate file size and students are responsible for ensuring that work is uploaded successfully before the deadline. If there are technical issues when submitting online, please contact the Engineering Student Office (eng.eso@warwick.ac.uk).

Guidance and Referencing Style

It is serious Academic Misconduct to pass off the work of others (including peers or AI-based chatbots such as ChatGPT) as your own and you should not permit colleagues to copy from you. Sources must be appropriately and properly acknowledged every time reference is made to another's work, using the Harvard Referencing system. Failure to do so amounts to plagiarism which breaches university regulations and falls short of the Academic Integrity expected in the department and university.

Find out more about the School of Engineering Referencing System here:

https://warwick.ac.uk/fac/sci/eng/eso/undergraduate_students/guidance/handbook/skills/shb-2-04

There are also other types of academic offences including duplication or 'self-plagiarism'. Refer to https://warwick.ac.uk/fac/arts/history/students/undergraduate/assess-plagiarism/ for further details.

Style and Formatting Guide

Submissions should follow the style and format of the provided m-file and docx-file templates.

Assignment Feedback

You will receive feedback in two ways:

1. **Individual written feedback** that lists your score for each part of each question and brief comments on areas where improvements could have been made.

2. A **cohort feedback presentation** will review common strengths and weaknesses and summarise the solution methodology for each question.

The Brief

The full assignment briefing follows.

ES3C5: Signal Processing
Coursework Lab Assignment Briefing Sheet
Autumn 2025

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

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

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 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. Relevant Content

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

2. Support

The module support forum for all Lab Assignments under the "Key Module Resources" heading on Moodle can be used to post questions. Feel free to post a question there or to check it for previous queries. You can also email the module leader. Queries posted to the forum or sent via email will be responded to within 2 business days.

Email/forum cutoff: *In fairness to all students, the module leader will not respond to Assignment queries (either forum or email) sent after* ***5pm on Wednesday 21 January 2026*** *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.** Many cases of academic misconduct (including plagiarism) have previously resulted in penalties for the assignments in this module.
It is expected that you may 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 or use files from another student or other source as a starting point for your own submission. Copying or allowing your work to be copied, even if it is modified, still counts as plagiarism and can be flagged by our academic integrity processes.

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 marks deducted for any deviations. Precise naming is standard practice in programming to ensure integration.

The instructions to complete this lab assignment are as follows:

1. Download the four files from **Coursework Lab Assignment Data** from the ES3C5 Moodle Page under the Assessment Information heading.
(a) ES3C5_2025_2026 _lab_template.m
(b) ES3C5_2025_2026 _lab_template.docx
(c) u<ID>_lab.txt
(d) 2025_2026_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 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_2025_2026_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_2025_2026_lab_template.docx to u<ID>_labreport.docx.

4. Open u<ID>_lab.m and rename the function title in the first line from ES3C5_2025_2026_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. Do not change the input or output arguments of the subfunctions.

5. Complete the prescribed steps in the Recording Supplement (2025_2026_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 5 and refer to u<ID>_lab.txt for the functions and parameter values that you must use. You will need to load signals from u<ID>_lab_signals.mat and there are already calls to do this in the template code. Enter the required code in the corresponding subfunctions in u<ID>_lab.m; you should not write your own functions, and you should only use functions from toolboxes that we cover in the module. The code must calculate the answers using your assigned parameters unless otherwise indicated. Take care to use parameters with the correct units.

7. When a question includes plotting a figure, please account for the following:
(a) Always start by creating a new blank figure. Do not close or overwrite existing figures.
(b) The plot function calls must be inside the question subfunction code.

(c) Carefully consider appropriate parameter ranges to view all necessary detail.
(d) ALL submitted plots must have labelled axes. You can add labels after your code runs.
(e) Any data point label should include the parameter name, value, and units.
(f) Figures must include your student number and question number at the top.
(g) ALL submitted plots must be placed in u<ID>_labreport.docx as png images.

8. Include readable comments in your code that briefly summarise your work, especially critical steps. See the subfunction Q0 in u<ID>_lab.m as a template example.

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. For example, you can run your code as follows (remember to correct the apostrophes if copying):

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

10. Place all necessary files 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). Submit the zip-file 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. Questions (Total: 80 Marks)

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

Certain parts of the assessment are deliberately designed to be more challenging (First Class Problems). The instructor will not provide guidance on how to answer them. Clarifications can be made regarding what these questions are asking.

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

- Writing output 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.

Question 1 (35 marks)

(a) (5 marks) Initial Audio Recording and Loading

- Record an audio clip with a length of at least 5 seconds and a sample rate of 22.05kHz. If the original recording is longer than 5 seconds, then shorten it to 5 seconds
- Save the audio recording as a vector with appropriate specifications to the file u<ID>_lab_Audio.mat.
- Load this audio into the Q1 variable, ensuring it is correctly assigned to Q1.audioInput.

Note: The correctness of the recording process and loading of the data will be evaluated.

(b) (8 marks) Noise Contamination and FFT Analysis

Your audio recording will now be contaminated with a synthetic fixed-frequency noise signal that you will generate yourself.

i. Generate a sinusoidal noise signal with a fixed noise frequency n1 Hz provided in the u<ID>_lab.txt file, using the same length and sample rate as your original audio (Q1.audioInput). Assign this noise signal to Q1.noiseSignal.
ii. Add this noise signal to your clean audio Q1.audioInput, and assign the noisy signal to Q1.audioNoisy.
iii. Calculate the fast Fourier transform (FFT) of the noisy signal Q1.audioNoisy (do not apply any scaling). Assign the FFT result to Q1.FFTNoisy.
iv. Plot the magnitude of this FFT output as a function of frequency in Hz, with a visible frequency range of [0, 22050) Hz.
v. Label the plot with the largest frequency component in the noise and save it as a .png file to be included in u<ID>_labreport.docx.

(c) (12 marks) FIR Filter design to isolate the largest noise frequency

Design a Finite Impulse Response (FIR) filter to suppress (i.e., primarily filter out) the most prominent noise frequency component in your noisy signal.

i. *Target specification:* In no more than 5 lines, justify a target FIR filter design to suppress the noise frequency component, including the desired filter type (low-pass, high-pass, band-stop, etc.), passband/stopband edges, and any gain requirements.
ii. *Filter design:* Using the target specification, design the FIR filter. Assign the impulse response of the filter to Q1.h.
iii. *Frequency response plot:* Use the freqz function to plot the magnitude of the filter's frequency response. Save the plot and ensure it is well-labelled to validate the design, including frequency cutoffs and passband/stopband. Include the plot in u<ID>_labreport.docx.
iv. *Design validation:* In no more than 5 lines, briefly comment on whether the plot validates your design. In addition to your comment, quantify the attenuation (in dB) at the noise frequency and state whether it meets your target specification.
v. *Test Filter design:* Pass the noisy audio signal Q1.audioNoisy through your filter using the filter function and assign the output to Q1.filteredAudio.
vi. Label the plot with the noisy signal and filtered signal and save it as a .png file to be included in u<ID>_labreport.docx.

(d) (10 marks) Apply an Effect of Your Choice

Select one audio effect (e.g., chorus, flanger, phaser, or echo) and apply it to your clean audio signal Q1.audioInput.

i. Design and code the effect (you may adapt a known difference equation). Ensure the processed signal has the same length as the input. Save the output as Q1.audioCustom. *Hint: The index variable must be a positive integer.*
ii. Compare the original and processed signals using a time-domain plot. Save the plot as .png and include it in u<ID>_labreport.docx.
iii. In no more than 5 lines, briefly explain: Why you chose this effect? What parameters you selected (e.g., delay time, modulation frequency, feedback)? How the processed audio sounds different from the original?

Question 2 (20 marks)

You are tasked with designing an IIR low-pass filter to remove high-frequency noise from the output of a biomedical signal acquisition system. The system operates at a sampling frequency of *fs* kHz. The signal of interest is below a cutoff frequency of *fc* Hz, and the filter should provide maximum attenuation (in dB) for frequencies beyond the stopband edge *fsb* Hz, while preserving signal components below the cutoff.

The parameters are provided in the u<ID>_lab.txt file:

(a) Filter Design (10 Marks)

i. Design appropriate IIR low pass filter (Butterworth or Chebyshev) to meet the given specification.
ii. In no more than 5 lines, justify your filter type selection based on the requirements in u<ID>_labreport.docx.
iii. Use MATLAB to calculate the filter order and the filter coefficients (b and a). Assign the filter order to Q2.filterOrder and coefficients to Q2.b and Q2.a.

iv. In no more than 5 lines, explain how the selected order and cutoff affect signal distortion near the transition band, computational complexity and feasibility

(b) Frequency Response and Filter Validation (10 Marks)

i. Plot the magnitude response and phase response of the filter using MATLAB's freqz function. Clearly label the pass band and stop band frequencies on the plot. Save and include the plot in u<ID>_labreport.docx.
ii. Annotate the plot with the stop band attenuation at fsb Hz and verify whether the attenuation meets the design requirement of maximum attenuation (provided in dB) at the stop band frequency and include in u<ID>_labreport.docx.
iii. In no more than 5 lines, briefly comment in u<ID>_labreport.docx on the effectiveness of the filter in isolating the low-frequency signal from high-frequency noise. What parameters (e.g., order, cutoff, ripple) could be adjusted to enhance performance? Comment on the Trade-offs

Question 3 (25 marks)

During testing of a hydraulic pressure line in a factory, engineers observe that when the valve is opened, the line pressure does not rise smoothly. Instead, the pressure response shows a combination of a first-order rise and a repeating oscillation caused by valve stick–slip effect. The response is modelled according to the general equation provided in u<ID>_lab.txt, where p(t) is the pressure in bar and N(t) is the measurement noise at time t in seconds. You have collected noisy pressure readings in column vector P in your signals file u<ID>_lab_signals.mat. Your data were collected starting from time t = 0 at the constant interval Ts in your parameter file. Please complete the following

(a) **Construct the Linear Model (5 marks)**
Construct the linear model observation matrix Θ needed to estimate the unknown parameters A, B, and C (in that respective order) from the given equation. Assign this observation matrix to Q3.Obs.

i. In no more than 5 lines, briefly comment in u<ID>_labreport.docx explaining why constructing an accurate observation matrix Θ is essential in estimating parameters in linear models. If you suspect that one of the terms in the model might not be capturing the true behaviour of the system, how could this impact the parameter estimates for A, B and C? What would you consider doing to improve the model accuracy?

(b) **Parameter Estimation (4 marks)**
Apply linear model estimation to estimate the unknown parameters A, B, and C. Assign the estimated parameters in the same order to the 3 × 1 column vector Q3.param.

(c) **Pressure Prediction (2 marks)**
Use the estimated parameters to predict the pressure readings for the same time points as the noisy sensor data. Assign the predicted pressure data to the column vector Q3.yHat.

(d) **Model Error Calculation (1 marks)**
Calculate the mean squared error (MSE) between the predicted pressure data and the noisy sensor data. Assign the MSE to the scalar Q3.mse.

(e) **Frequency Domain Analysis (8 marks)**
Perform a spectral comparison of the sensor readings and the predicted pressures:

i. Compute the double-sided Fast Fourier Transform (FFT) of the noisy pressure data. Assign the FFT result to Q3.yFFT.
ii. Compute the FFT of the predicted pressure data and assign the result to Q3.yHatFFT.
iii. Create a frequency vector associated with the FFT results. Assign the frequency vector to Q3.fRange.
iv. Plot the magnitude spectra of both Q3.yFFT and Q3.yHatFFT on the same axes, over the range [0, 1/Ts] Hz. Label all key spectral peaks and axes. Save the FFT magnitude plot as a .png file and include it in u<ID>_labreport.docx
v. In no more than 5 lines, in u<ID>_labreport.docx, briefly comment on the accuracy of the frequency-domain model. Would a time-domain plot be helpful in verifying model accuracy?

(f) **(5 marks)** In no more than 5 lines, comment on the physical meaning of each estimated parameter in relation to the pressure response and valve behaviour. Suppose the residuals (measured – predicted) show a strong pattern at another frequency. What does this imply about the adequacy of the model structure? How could you modify the model to improve its accuracy?

6. 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:

1. u<ID>_lab.m (where <ID> is your student number), containing your code.
2. u<ID>_lab_signals.mat (where <ID> is your student number), containing your assigned signals. This file should be unchanged from the version that you downloaded.
3. u<ID>_lab_Audio.mat (where <ID> is your student number), containing your original audio recording.
4. u<ID>_lab_Answers.mat (where <ID> is your student number), containing the answers generated by your code.
5. u<ID>_labreport.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 22 January 2026**, otherwise standard late penalties will be applied. Applications for extension without penalty must be made via Tabula.

7. Notes on Marking

Each problem will be marked according to:

- Correctness of the audio recordings. Your submitted code must not play any audio.
- Correctness of the calculated answers, figure contents, and written answers.
    - Incorrect answers will not automatically penalise you in later problems.
    - Written answers will not be read beyond the number of lines specified in the question. A line of math will be read as 1 line.
- Implementation of the code used to arrive at your answers. All 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:
    - Trying to use a MATLAB toolbox that we do not introduce in the module.
    - Trying to read from a parameter before it has been created.
    - Making changes to the format of the output argument structure.