Nottingham Trent University

School of Science and Technology
Department of Engineering

Contents

1 Details of Module and Team

MODULE CODE	ENGG30201
MODULE TITLE	Machine Learning for Engineers
MODULE LEADER	Prof Harris Tsimenidis
TUTOR(S)	N/A
COURSEWORK TITLE	Adaptive Filtering
LEARNING OUTCOMES ASSESSED	K3, K5, S1, S2, S3, S4
CONTRIBUTION TO MODULE	50%
DATE SET	March 11, 2025
DATE OF SUBMISSIION	April 4, 2025 at 14:30
METHOD OF SUBMISSION	Dropbox
DATE OF SUMMATIVE FEEDBACK	Two weeks after the submission deadline
METHOD OF FORMATIVE FEEDBACK	Provided on progress in lab sessions

Late submissions and NECs

Work handed in up to five working days late will be given a maximum Grade of Low Third/ Pass whilst work that arrives more than five working days will be given a mark of Zero.

Please note if you are repeating the work and are capped in your grade, you will receive a Zero grade if the work is submitted at all late.

Work will only be accepted beyond the five working day deadline if satisfactory evidence, for example, an NEC is provided. **https://www.ntu.ac.uk/studenthub/my-course/student-handbook/submit-a-notification-of-extenuating-circumstances**

Group work

If appropriate describe the process involved in grading where group work and NECs are involved or delete.

Breaches of Academic Integrity

To ensure that you are not accused of any breaches of Academic Integrity, look at the NOW page Plagiarism and Academic Integrity at NTU. for guidance.

The University views plagiarism and collusion as serious academic irregularities and there are a number of different penalties which may be applied to such offences. The Quality handbook a section on such breaches, which outlines the penalties and states that **plagiarism** includes:

Presenting someone else's ideas as your own **(including text, graph, diagrams, videos etc.)** in a substantial proportion of your work, with or without consent, by incorporating it into assessment without full acknowledgement, including:

Self-plagiarism: reproducing or representing work for assessment without proper attribution and attempting to gain credit for this work where credit has already been received.

Paraphrasing: rephrasing a source's ideas without proper attribution.

Mosaic plagiarism/patchworking: weaving phrases and text from several sources into your own work; and/or adjusting sentences without quotation marks or attribution.

Source-based plagiarism: providing inaccurate or incomplete information about sources such that they cannot be found.

Computer code plagiarism: copying or adapting source code without permission from and attribution to the original creator.

Whereas **collusion** includes:
working with other students on an assessment meant for individual submission Sharing your work with other students enabling them to plagiarise your ideas.

Please remember submitting portions of work already assessed for the same learning outcomes is **Self-Plagiarism** and is also a serious academic irregularity. Penalties for Breaches of academic integrity range from capped or zero grades for elements of modules, to dismissal from the course and termination of studies.

Chat GPT and other AI-powered language models.
It is important to note when using any AI platform that they generate the most common responses to questions, not necessarily the correct ones. They also fabricate evidence. The material they produce is not your own words. Assessments require you answer questions giving your own view and in your own words. The outputs from platforms such as Chat GPT do not provide that.

By presenting such material as your own words you are violating Academic Integrity policy, a matter that NTU takes very seriously.

The skills you develop during your time with us allow you to interrogate material and evaluate it, important skills in all careers. Generative AI does not allow you to develop these.

If you have utilised such platforms, you must retain any outputs from them to provide evidence your work is your own in the case of suspected breaches of Academic Integrity.

2 What Learning Outcomes are assessed?

See table in Section 1.

3 What are my Deadlines and how much does this assessment contribute to my Module Grade?

See table in Section 1.

4 What am I required to do in the assessment?

This coursework is a mini project which applies machine leaning techniques to synthesize and analyse data series as encountered in real-world engineering scenarios. You are asked to implement several adaptive filter techniques in Python programming language to solve a specific engineering problem and report the code development, implementation and results. Adaptive filters are a specialised form of machine learning, particularly relevant in real-time signal-processing applications. They use supervised and online learning to update their parameters iteratively based on input data. The objective is to obtain a model that learns to minimize an error or cost function. Learning is achieved using gradient-based optimisation such as the Least-mean Squares (LMS) algorithm, which is also fundamental in training neural networks and other machine learning models. The main focus of this coursework will be placed on interference cancellation in electrocardiogram (ECG) signals using adaptive filters.

4.1 Problem Statement

4.1.1 Measurement Setup

Fig. 1 illustrates the setup for measuring an ECG signal. In such scenario, external interference from power-line noise (typically at 50 or 60 Hz) or electromagnetic interference (EMI) from nearby devices such as infusion pumps, monitors, and fluorescent lighting can obscure the ECG signal. In order to cancel the interference signal, an adaptive filter is trained using a reference electrode placed on a non-active site, such as the patient's arm or leg, to measure the background noise (e.g., power-line hum) without the cardiac signal. The challenge lies in developing an adaptive filtering algorithm capable of distinguishing and attenuating interference noise, while preserving the critical details of ECG signals. By addressing this challenge, the system aims to enhance the ECG quality, by transforming noisy ECG recordings into clear, interpretable signals, ensuring high-quality data for effective healthcare.

4.1.2 Modelling of the Unknown Coupling Systems

The unknown coupling systems are modelled as Finite Impulse Response (FIR) filters with specified transfer function (TF), i.e. the interference-to-electrode-1 system the TF is given us

$$H_1(z) = 1 - 0.2z^{-2} + 0.3z^{-3} \tag{1}$$

while for the interference-to-electrode-2 we have

$$H_2(z) = 1 - 0.6z^{-2} + 0.3z^{-4} - 0.2z^{-7} \tag{2}$$

where z^{-n} represents the delay operation (see Seminar 5). The coupling systems model the propagation of the interference from the source to the electrodes.

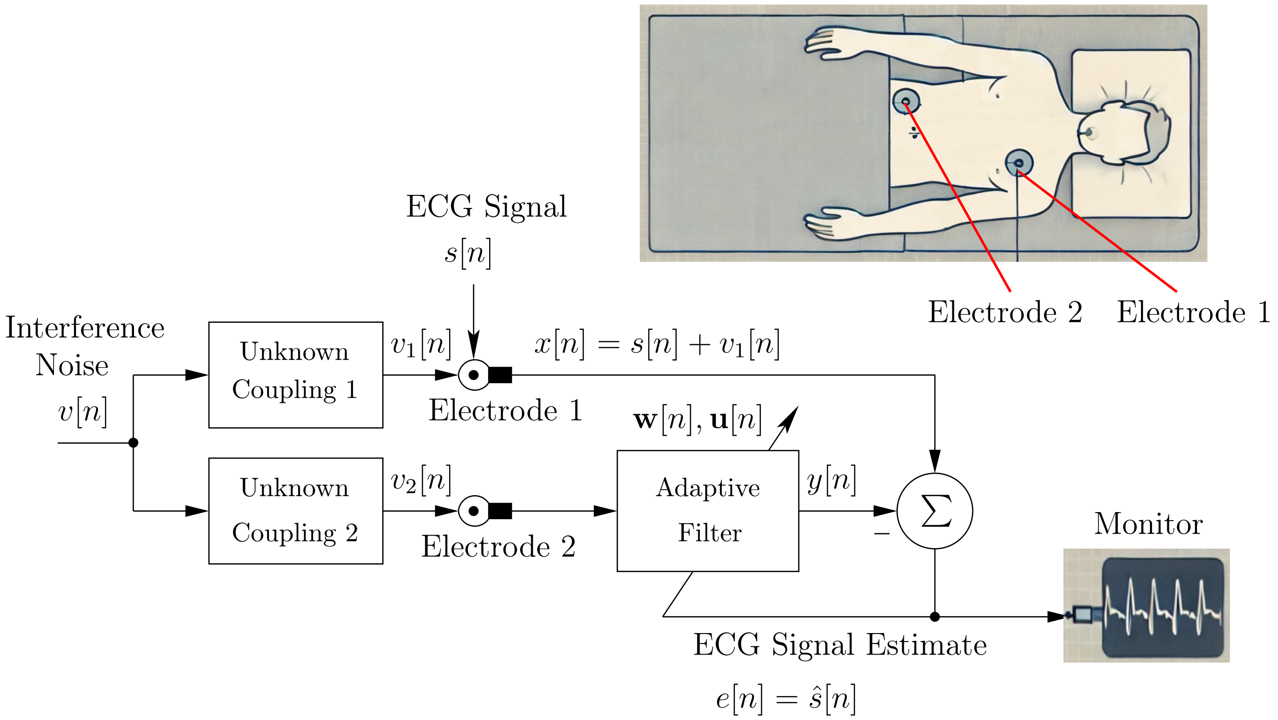

Figure 1: ECG signal measurement setup.

4.1.3 Modelling of the ECG Signal

An ECG pulse as shown in Fig. 2 can be represented as a sum of Gaussian functions and a sinusoidal baseline drift, that is

$$s[n] = s_p[n] + s_q[n] + s_r[n] + s_s[n] + s_t[n] + s_b[n] \tag{3}$$

where T_s is the sampling time is seconds. The six terms are referred to as

- The P wave represents atrial depolarisation. This is the electrical activity associated with the contraction of the atria (upper chambers of the heart), that is

  $$s_p[n] = A_p \exp\left[-\frac{(nT_s - t_p)^2}{\sigma_p^2}\right]$$

  with magnitude $A_p = 0.2$, centered at $t_p = 0.2$ and width $\sigma_p^2 = 0.02$.

- The Q wave represents the beginning of ventricular depolarisation, which is the process of electrical activation of the heart's ventricles, that is

  $$s_q[n] = A_q \exp\left[-\frac{(nT_s - t_q)^2}{\sigma_q^2}\right]$$

  with $A_q = -0.1$, $t_q = 0.45$, and $\sigma_q^2 = 0.002$.

- The R wave reflects the electrical activation of the heart's pumping chambers and is given as

  $$s_r[n] = A_r \exp\left[-\frac{(nT_s - t_r)^2}{\sigma_r^2}\right]$$

  with $A_r = 1.0$, $t_r = 0.5$, and $\sigma_r^2 = 0.001$.

- The S wave represents the final phase of ventricular depolarization, primarily involving the posterior and basal portions of the heart, that is

$$s_s[n] = A_s \exp\left[-\frac{(nT_s - t_s)^2}{\sigma_s^2}\right]$$

$A_s = -0.4$, $t_s = 0.53$, and $\sigma_s^2 = 0.002$.

- The T wave represents ventricular repolarization, the process by which the ventricles reset their electrical state after contracting, i.e.

$$s_t[n] = A_t \exp\left[-\frac{(nT_s - t_t)^2}{\sigma_t^2}\right]$$

$A_t = 0.4$, $t_t = 0.75$, and $\sigma_t^2 = 0.02$.

- The last term, called the baseline drift, refers to the slow, wandering movement of the baseline (isoelectric line) on ECG tracing. It is given as

$$s_b[n] = A_b \sin(2\pi f_b n T_s)$$

with magnitude $A_b = 0.02$, and frequency $f_b = 0.5\,\mathrm{Hz}$.

For the remainder of the assignment we assume the sampling frequency is $f_s = 1\ \mathrm{kHz}$ and the sampling time is $T_s = 1/f_s = 1\ \mathrm{ms}$.

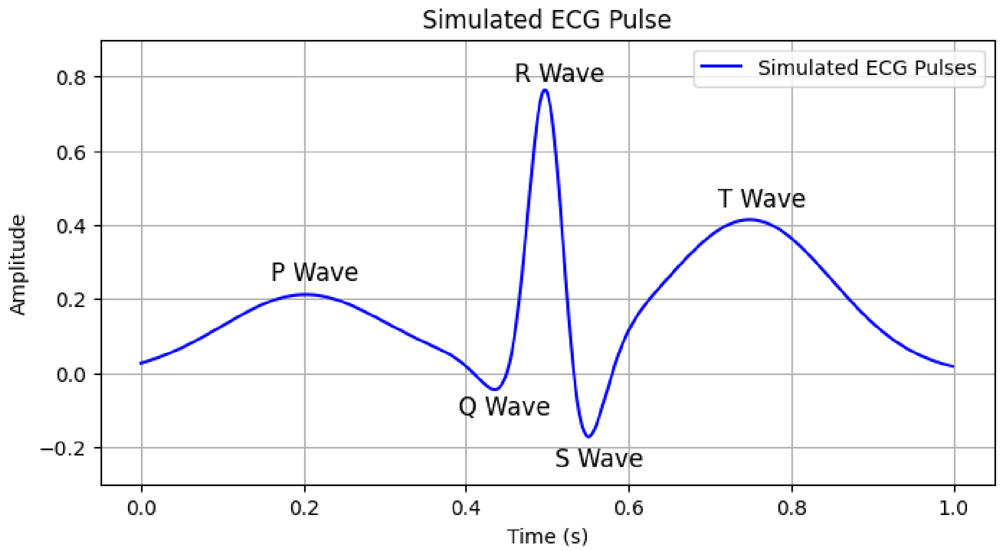

Figure 2: ECG signal pulse.

4.1.4 Modelling of the Interference

To model the effect of hum interference in ECG signals, we can simulate how powerline interference at 50 Hz contaminates the ECG waveform. The hum signal is modelled as

$$v[n] = A_h \sin(2\pi f_h n T_s + \phi) \tag{4}$$

where $A_h = 1$ V is the hum magnitude, $f_h = 50$ Hz is the frequency in Hz, and T_s is the sampling time in seconds. Furthermore, ϕ is a random phase uniformly distributed between $[0,\ 2\pi)$. The corrupted ECG signal is then given as

$$\begin{aligned} x[n] =& s[n] + v_1[n] \\ =& s[n] + h_1[n] * v[n] \end{aligned} \tag{5}$$

where $*$ denotes the discrete convolution operation. Furthermore, as shown in Fig. 1, the reference signal that is the input to the adaptive filter is given as

$$v_2[n] = h_2[n] * v[n] \tag{6}$$

4.2 Code Development Deliverables

4.2.1 Deliverables Set 1: Modelling and data series synthesis [32 Points]

Furthermore, in this part, you need to complete the following deliverables:

1. Using Eq. (3), write a Python function that takes as input the pulse duration in seconds and sampling frequency in Hz and generates a single ECG pulse.
2. Generate a signal $s[n]$ with 100 ECG pulses (100 s) and plot the last 3 s of the $s[n]$.
3. Compute and plot the magnitude spectrum of $s[n]$ and zoom the plot in the area between 0 and 20 Hz. Use an appropriate dimension for the Fast Fourier Transform length. You do not need to use the full signal $s[n]$ to compute the spectrum. Scale the x-axis appropriately to reflect frequency in Hz.
4. Using the `stem` plotting function, plot the impulse responses of the two unknown coupling systems, $h_1[n]$ and $h_2[n]$. Scale the x-axis appropriately to reflect time in seconds.
5. Compute and plot the frequency responses (magnitude and phase) of $H_1(z)$ and $H_2(z)$. Scale the x and y axes in Hz and dB, respectively.
6. Write a Python function to generate and filter the hum noise $v[n]$ with the coupling systems $h_1[n]$ and $h_2[n]$ to produce the coupled noise sequences $v_1[n]$ and $v_2[n]$.
7. Compute and plot the histograms of the coupled noise sequences $v_1[n]$ and $v_2[n]$ and compute the sample mean and standard deviation estimates of $v_1[n]$ and $v_2[n]$.
8. Compute the noise-corrupted ECG signal $x[n]$, and plot it against the original clean ECG signal $s[n]$. Plot the last 3 s of the two signals.

Note 1: For all plots use appropriate labelling of axes, title, legend and grid.

Note 2: Each deliverable is equally weighted.

4.2.2 Deliverables Set 2: Noise Cancellation using Adaptive Filtering [36 Points]

The ECG signal can be enhanced by using a Least Mean Squared (LMS) adaptive filter to cancel the coupled engine noise using the structure shown in Fig. 1. In this part, you need to complete the following deliverables:

1. Declare and initialise the following variables (hyperparameters):

(a) The learning rate η.

(b) Array variables of size M to store the LMS filter weights $\mathbf{w}[n]$ and memory $\mathbf{u}[n]$.

(c) Array variables of size N for the LMS filter output $y[n]$ and error signal $e[n]$.

2. Implement the steps of the LMS algorithm as outlined in Lecture 5 and run the algorithm for the setup in Fig. 1.

3. Plot the last 3 s of the error signal $e[n] = \hat{s}[n]$ against the same segment of the original ECG signal $s[n]$. Use subplots for this purpose.

4. Compute and plot the last 3 s of the instantaneous squared error $\epsilon_{v_1}[n]$ in dB between the LMS filter output $y[n] = \hat{v}_1[n]$ and the corresponding segment of the coupled unknown noise $v_1[n]$ given as

$$\epsilon_{v_1}[n] = 20\log_{10}\left(|y[n] - v_1[n]|\right)$$

5. Compute and plot the instantaneous squared error $\epsilon[n]$ in dB between the clean ECG signal $s[n]$ and the LMS filter error $e[n] = \hat{s}[n]$ given as

$$\epsilon[n] = 20\log_{10}\left(|s[n] - e[n]|\right)$$

Focus the plot on the first 10 s of $\epsilon[n]$ to demonstrate the convergence.

6. Design an averaging FIR filter with $K = 50$ coefficients to filter the signal $\epsilon[n]$ and plot the magnitude and phase responses of its transfer function given as

$$H_{av}(z) = \frac{1}{K}\sum_{k=0}^{K} b_k z^{-k}$$

where the all coefficients are $b_k = 1$. The impulse response $h_{av}[n]$ of this filter is an array of K ones scaled by $1/K$.

7. Apply the averaging filter to the signal $\epsilon[n]$ and plot the result over the graph from the previous task.

8. Optimise the hyperparameters of the adaptive system, such the learning rate η and filter size M. Use the MSE error to drive your optimisation to achieve fast convergence and low MSE levels. Include the optimized results in your report.

9. Compute the heart rate using the corrupted signal $x[n]$ and the equalised signal $e[n]$ via the FFT spectrum method. To achieve this, increase the frequency range to $[0, 60]$ Hz and plot the magnitude spectrum of two signals $x[n]$ and $e[n]$. Explain the results obtained by the FFT method compared to the adaptive filter.

Note 3: Each deliverable is equally weighted.

4.3 Deliverables Set 3: Report [32 Points]

The report should be at least **2500 words** excluding references and embedded code segments, and should include the following points:

1. Title Page

   - Assignment Title: A clear and descriptive title.

- Student Information: Name, ID number, course name, and instructor's name.
- Submission Date: The date the report is submitted.

2. Abstract that summarises the work and results achieved.
3. Background and methodology related to adaptive filtering and noise cancellation.
4. Derivation of the range for the learning rate η.
5. Documentation of the developed code.
6. Discussion of results.
7. Conclusions.
8. Relevant references and their correct citation in the text
(Use IEEE style of referencing).

Note 4: Each deliverable is equally weighted.

Note 5: Individual submissions are also allowed but you still need to address all the deliverables of the assignment.

5 What are my assessment criteria? (What do I have to achieve for each grade?)

For the Assessment Criteria see additional file in **NOW**.

6 Submission Details

- The submission deadline is **April 4, 2025 at 14:30 .**
- One group member needs to submit the report in PDF format and the python code files in ipynb format from Colab all compressed in a zip file.
- Use the following format for the zip file name: **ENGG30201_Group_XX.zip**
- Each group member needs to submit a peer assessment form.
- Use the following format for the peer assessment file name:
ENGG30201_Group_XX_Student_ID

7 The Moderation Process and Peer Assessment

All assessments are subject to a two-stage moderation process. Firstly, any details related to the assessment (e.g., clarity of information and the assessment criteria) are considered by an independent person (usually a member of the module team). Secondly, the grades awarded are considered by the module team to check for consistency and fairness across the cohort for the piece of work submitted.

Furthermore, the peer assessment form will ensure that each group member is evaluated fairly and provide an opportunity for constructive feedback. It is expected to encourage self-reflection and accountability, which are essential for successful teamwork. The peer assessment form is confidential and will only be used to assess group dynamics and individual contributions, and feedback will be anonymized when shared with the group. Moreover, the peer assessment scores may lead to a $\pm 5\%$ adjustment of the final mark at the discretion of the module leader and moderator.

8 Can I get formative feedback before submitting? If so, how?

During the lab sessions, whilst you are working on the coursework, you will be given formative feedback from your Module Leader using in-lab exercises. On Wednesday afternoons, there is also the opportunity to book appointments to discuss outside of class time.

9 What extra support could I look for myself?

In addition to the activities planned in the module, students can explore additional online information available on the links below:

- `https://www.w3schools.com/python/`
- `https://scikit-learn.org/stable/`
- `https://www.tensorflow.org/learn`

10 Is there feedback from other work that would be useful?

- Presentation Skills: You may apply lessons from past presentations to improve the delivery and clarity of your current work.
- Workflow Strategies: Reflect on how you managed your time in past courseworks and apply effective strategies (e.g., breaking tasks into smaller steps, setting deadlines) to the current assignment.
- Critical Analysis: Use past critical thinking exercises to evaluate arguments, identify biases, or develop well-reasoned conclusions.
- Academic Integrity: Ensure that any reuse of past materials complies with academic integrity policies. Avoid self-plagiarism as you have already received credit for your previous work.
- Group Work Insights: Reflect on past group projects to improve communication, delegation, and collaboration.
- Peer Feedback: Use insights from peer reviews in past courseworks to enhance the quality of your current work.

11 How and when do I submit this assessment?

The deadline for the report submission is April 4, 2025 at 14:30 via Dropbox in NOW. The Dropbox will appear a couple of days before the deadline.

12 How and when will I get summative feedback?

After you have submitted the coursework, you will receive specific feedback regarding your coursework submission together with your awarded grade when it is returned to you. Clearly, feedback provided with your coursework is only for developmental purposes so that you can improve for the next assessment or in subject-related modules.

13 What skills might this work evidence to employers?

This module and the work on two courseworks will offer substantial proof of diverse academic, technical, and interpersonal abilities, including proficiency in Python programming, organisational skills, analytical thinking, reflective practices, and competence in report writing. Moreover, it provides prospective employers with explicit demonstrations of your capacity to handle engineering datasets intelligently and to discern their business significance. Within the report, you will cover crafting a scientific-style document, conducting a literature review, executing programming tasks, employing proper referencing techniques, and accurately labelling figures, documenting and discussing code implementation and scientific results.

Most of these skills are valuable assets for job applications or inclusion in your Skills Portfolio. Likewise, the hands-on lab activities offer multiple instances suitable for incorporation into your Skills Portfolio, such as proficiency in Python programming, machine learning methods, and data analysis.