**Machine Learning for Electrode Motion artefact removal in ECG signals**

A red and white logo

Description automatically generated

Student ID : 40178580

Date : 02/01/2024

Supervisor : Professor Sean Mcloone

# **Project Specification**

Table 1 indicates the relevant sub-field that are employed in this project.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Control | X | Embedded Systems |  | High Frequency Electronics |  | Microelectronics |
|  | Electric Power | X | Software | X | Connected Health |  | MEMS |
|  | Cyber-Security |  | Wireless Communications | X | Signal/Image Processing | X | Intelligent Systems |
|  | Digital Design |  | Sensor Networks | X | Data Analytics | X | Electronics |

Table 1 Shows the sub-fields that are relevant to this MSc project

The electrocardiogram (ECG) is a non-invasive method to measure the electrical activity of the heart and can be used to diagnose heart disease. According to the World Health Organisation (WHO), chronic heart disease was the number one cause of death from 2000 – 2019. Long term ECG monitoring is currently the gold standard for diagnosing cardiovascular diseases (CVDs), however obtaining reliable long-term measurements of the ECG signal is challenging because patients are required to collect their ECG signal remotely on a wearable device. Wearable devices are inherently contaminated with noise which can supress the essential pathological biomarkers and, in some cases, render the ECG completely unusable. ECG signals can be contaminated by many types of noise including: 1) Baseline Wander, 2) Powerline Interference, 3) Electromyographic and 4) Electrode Motion artefacts. Most of these noise sources can be reduced through the use of time and frequency domain digital filters. However, the frequency spectrum of electrode motion noise overlaps with the frequency spectrum of a typical ECG signal making it very difficult to remove in the time and frequency domain. The objective of this project is to explore if AI/ machine learning can be used to learn the characteristics of, and correct for, electrode motion induced noise on ECG signals.

**Objectives**

1. Conduct a literature review on ECG motion artefact removal algorithms to identify the different approaches that exist, and the challenges involved in developing effective methods and assessing their performance.
2. Using available online sources (e.g. Physionet1) compile an ECG dataset which can be used to investigate and assess ECG motion artifact removal algorithms.
3. Develop a model for generating synthetic motion artefacts in clean ECG signals and use it to create a reference dataset for ground truth comparisons.
4. Investigate machine learning approaches to reducing electrode motion noise on ECG signals.
5. Develop and implement a candidate approach using Python or Matlab and validate its performance on the datasets from (2) and (3).
6. Compare the performance of the developed algorithm against alternative baseline algorithms from the literature.

**MEng Extension**

1. Explore advanced deep learning concepts (e.g., transfer learning, data augmentation) to enhance the performance of models and/or develop and evaluate alternative machine approaches for motion artefact removal.
2. Provide a rigorous assessment of all approaches developed with regard to real-time/embedded system implementation constraints.

**Learning Outcomes**

At the end of the project the student will be able to demonstrate:

1. A good understanding of ECG denoising algorithms.
2. A working knowledge of machine learning/ deep learning models and associated development tools
3. Enhanced programming skills in Python or Matlab, particularly with regard to algorithm development and signal processing.

Contents

[**1** **Project Specification** 2](#_Toc157514461)

[**1.** **Introduction** 5](#_Toc157514462)

[**1.1** **Example Filters** 5](#_Toc157514463)

[**1.1.1** **High-Pass Finite Impulse Response (FIR) Filter.** 5](#_Toc157514464)

[**1.2** **High-Pass Infinite Impulse Response (IIR) Filter.** 6](#_Toc157514465)

[**1.2.1** **Moving Average (MA) Filter.** 6](#_Toc157514466)

[**1.2.2** **Moving Median Filter** 6](#_Toc157514467)

[**1.2.3** **Wavelet Transform Denoising** 6](#_Toc157514468)

[**1.2.4** **Empirical Mode Decomposition** 6](#_Toc157514469)

[**1.2.5** **Adaptive Filter** 7](#_Toc157514470)

[**1.3** **Dataset** 7](#_Toc157514471)

[**1.4** **Models** 9](#_Toc157514472)

[**2** **Methodology** 10](#_Toc157514473)

[**3** **Conclusion** 11](#_Toc157514474)

[**4** **References** 11](#_Toc157514475)

# **Introduction**

There are 4 major sources of noise in ECG signals, these are 1) Baseline Wander (BW), 2) Powerline Interference (PL), 3) Muscle Artefact (MA) and 4) **Electrode Motion (EM)**. BW and PL have a relatively unique frequency content and thus are easily removed by simple digital filters (Eg notch, low-pass). MA and EM noise are more challenging to remove as they can have a wide frequency content that overlaps with that of the ECG signal, specifically the PQRST complex. Adaptive filters are primarily used for these type of noise sources, however these require a reference noise signal which needs to be estimated, providing a potential source of error.

Electrode motion noise in Electrocardiogram (ECG) signals refers to the interference caused by the movement of electrodes attached to the skin. When electrodes move, even slightly, they can pick up additional electrical activity not related to the heart's electrical signals. This extra activity is seen as noise in the ECG signal, which can distort the true reading. It's particularly problematic during physical activity or if the electrode doesn't adhere well to the skin. This noise appears as irregular spikes or a fuzzy baseline in the ECG trace, making it challenging to accurately interpret the heart's electrical activity. Managing electrode motion noise is crucial for ensuring reliable ECG readings, especially in scenarios requiring patient movement or long-term monitoring.

Electrode Motion is difficult to remove from an ECG signal for several reasons. Firstly, the signal attenuated by electrode motion can be very similar to the ECG signal itself, this similarity can make it difficult to remove through the use of digital filters which use both time and frequency domain characteristics to separate the two signals. The variability of EM noise also adds to the difficulty to remove the noise, this comes from the wide range of movements a subject can undergo such as running, walking, jumping or any other from of activity. Each activity will return a different characteristic shape of noise. Electrode placement can also vary the shape of the noise signal. Finally, individual differences such as skin type, amount of hair and other factors influencing the electrode-skin contact will affect the amount of EM noise added during movement, it is important that algorithms can deal with this problem effectively.

## **Example Filters**

### **High-Pass Finite Impulse Response (FIR) Filter.**

Described by the following discrete difference equation:

(1)

Where is the coefficient of the filter, M-1 is the order of the filter, M is the length of the filter, x(n) is the input signal and y(n) is the filtered signal.

## **High-Pass Infinite Impulse Response (IIR) Filter.**

Described by the following discrete difference equation:

(2)

Where and are the coefficients of the filter, N is the order of the filter, x(n) is the input signal and y(n) is the filtered signal.

### **Moving Average (MA) Filter.**

Described by the following discrete difference equation:

(3)

(4)

Where denotes the noisy ECG signal, denotes the estimated noise, is the ECG signal after denoising, M is the filter length.

### **Moving Median Filter**

Same as MA but uses median instead of average.

### **Wavelet Transform Denoising**

A diagram of a flowchart

Description automatically generated

Figure 1 Demonstration of 3-level discrete wavelet transform decomposition.

### **Empirical Mode Decomposition**

Described by :

(5)

Where represents the intrinsic mode function and represents the residual signal.

### **Adaptive Filter**

## **A diagram of a computer algorithm Description automatically generatedDataset**

Figure 2 Block diagram of adaptive filter.

The raw data used in this experiment comes from 2 different databases/tools available on PhysioNet. PhysioNet is an open source repository of freely available medical research data, managed by the MIT Laboratory for computational physiology. The first database used is the MIT-BIH Noise Stress Test Database (NSTDB) []. This database includes 12 thirty-minute ECG recordings and 3 thirty-minute recordings of noise typical in ambulatory ECG recordings. The electrodes were placed in such a position such that the subjects ECG was not visible. The three noise sources available are: 1) Baseline Wander, 2) Muscle Artefact and 3) Electrode Motion. For the purpose of this experiment, the EM noise file was extracted and added to clean ECG signals as discussed later in the report.

Due to the need for clean reference signals, this experiment utilises a PhysioNet tool called ‘ECGSYN’, which is a software package that can be used to generate realistic ECG waveforms. Since the software is based on 1st ODE’s originating from the heart, the signal produced should in theory, be completely devoid of noise. This is a crucial step which other researchers have not considered. The software also allows for the adjustment of waveform parameters (Ie amplitude, angle, width) which enables the creation of a large and diverse dataset.

The overall process can be broken into 3 steps;

1) Extraction and pre-processing of the EM noise file.

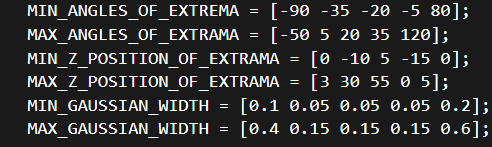
2) Generation of clean ECG database.

3) Corrupting each clean ECG with relative amounts of the EM file.

To extract and process the EM noise file, the EM.dat file was downloaded from PhysioNet database. After this, the .dat file was read into MATLAB and saved as a .mat file for easier processing, the sampling frequency was also read in. The EM noise file also contained a small amount of Baseline Wander, this would potentially cause issues when training the algorithm and such, a Chebyshev Type II Bandpass filter was used to mitigate this potential error source. The maximum frequency of the baseline noise was set to 1Hz, and the maximum frequency of electrode motion was set to 50Hz. A A screen shot of a computer code

Description automatically generated20th order filter, with 50dB attenuation was used.

The processed noise file was then saved in .mat format.

To produce a clean ECG dataset, the methodology is slightly more complex. The first step was to determine the minimum and maximum thresholds of each parameter within the ECG morphology, that is, the angles of extrema, the amplitudes, and the gaussian width of each of the five features within the ECG signal (P, Q, R, S, T). These min/max values have been selected arbitrarily for now, however it is my aim to find a suitable reference for these to be set off. Currently, the following are used:

Where each element in each array refers to the waveform feature (P, Q, R, S, T). Following this, Latin Hypercube Sampling is used as a sampling method to generate 10,000 random samples of each parameter, this number can be increased to expand the database. Latin Hypercube sampling produces random samples between the defined limits and thus, each ECG signal produced should satisfy the following requirements:

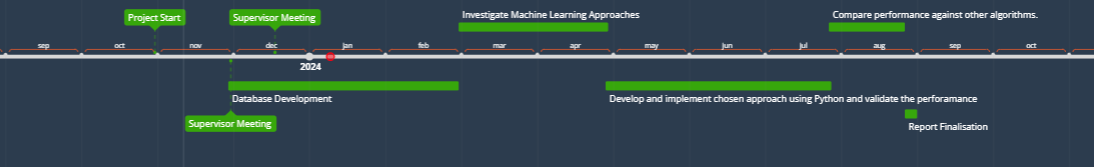
1. Each ECG signal will be a different morphology.
2. Each ECG should have waveforms representative of a real ECG.

The final step of the database creation process was to synthetically corrupt each clean ECG with different amounts and shapes of the noise file that was extracted in the 1st step. Again, this was performed on MATLAB by creating a function that is directed at both the single noise file, and 10,000 clean ECG’s. The SNR levels were set to be 0, 6, 12, 18 and 24 dB, this covers a good range from high noise levels to low noise levels. The 30 minute noise file was segmented and then in an attempt to expand the noise characteristics, an auto-regressive model (Arburg) was applied to generate coefficients of the reference noise signal. The new estimated noise signals were scaled to apply desired SNR levels to the ECG signals. Further details are provided later in this report.

## **Models**

There have been a few attempts in the past to develop Intelligent algorithms to remove Electrode Motion noise from an ECG signal. Namely, Brophy et all [] proposed a deep learning framework where they employ a custom loss function. Results look promising, however performance was not yet good enough to be used over standard digital filters. The primary focus of this report will be around the development of a deep learning framework, similar to the reference paper, however a new loss function will be utilised called Wavelet Energy Based Diagnostic Distortion (WEDD) [] which should provide a better means of capturing the diagnostic information within the ECG signals. Furthermore, the reference paper utilises 38 ECG records (That are not entirely without noise) to train their algorithm. This experiment will use a much larger training dataset, with perfectly clean ECG signals in an attempt to improve on previous work. All model development will be completed using the Pytorch framework.

The performance of the developed algorithms will be compared to that of standard digital filters mentioned in section 1.1. The performance will be established using a variety of objective metrics, specifically SNR improvement, Cross-Correlation and WEDD. Visual inspection of the denoised ECG signals will also be looked at.

Figure X shows the full project planning timeline, as well as the current position.

**Current Position**

A screenshot of a computer

Description automatically generatedAll work on this project is being tracked and version controlled using Github. A repository has been created [X] that stores all code and documentation for the project, this enables effective tracking of databases, models and reports. Sourcetree has been used as a graphical user interface (GUI) for Git which enables both developers and reviewers to easily look at changes throughout the project life cycle. All commitments to code and reports have been described in the corresponding repository commit, this is shown in the figure below:

As of 09/01/24, The project contains a mixture of MATLAB and Python code. MATLAB has been used to generate the reference database on which the model will be trained, this is due to MATLABs enhanced ability for signal processing problems where noise is being modelled and added to clean ECG signals. MATLABs ability to work with matrices makes this relatively simple. Python is used to develop machine learning models, using the reference database. To date, only a reference deep learning model has been added to the repository. Python has numerous frameworks specifically designed for implementing AI models (Pytorch, Tensorflow) and so has been selected as an appropriate language to use.

# **Methodology**

This section aims to dive further into the data processing techniques used in the dataset development, since this stage is almost complete.

## **Data Processing**

The following flow diagram presents a high level overview of the data processing steps followed to generate the noisy database used for training various intelligent algorithms:

**Figure 3 Flow Diagram over dataset creation**

**generatingNoisyEcgDatabase.m**

**createTrainingDataset.m**

**Single 30 minute noise record output as .mat file in unique directory**

**ECG signals output as .mat files in a unique directory**

**createSyntheticCleanEcgSignals.m**

**extractAndPreProcessNoiseSignal.m**

### **Extraction and pre-processing of noise file.**

A blue line graph with white text

Description automatically generated with medium confidenceElectrode motion noise was downloaded from Physionets Noise Stress Test database (NSTDB) [], along with the sampling frequency of the noise. The figure below shows the raw noise signal itself:

**Figure 4 Electrode Motion noise obtained from Physinet. The sampling frequency is 360Hz,**

A red line graph on a white background

Description automatically generatedThe ECG signal was then resampled to 500Hz using MATLAB’s native resample.m function, this was done to ensure all processing in the database creation is complete at 500Hz. This noise signal shown in figure 4 was then passed through a 20th order Chebyshev Type II bandpass filter to remove the baseline noise that can be seen, the cut-off frequencies were set at 1Hz and 50Hz such that only EM noise remained in the signal. The processed signal can be seen below:

**Figure 5 EM noise after resampling and bandpass filter.**

Finally, this noise file is saved in .mat format in a unique folder.

### **Creation of synthetic clean ECG signals.**

To create a dataset of clean ECG signals, Physionets ECGSYN tool was utilised. This tool allows a user to generate a clean ECG signal where they can choose the morphology of the beats. In this experiment, and as with the majority of Intelligent models, it is important that the algorithm is trained on as large a database as possible, with as many different characteristic shapes of the time series data.

ECGSYN allows for the alteration of the following features:

1. Angles of Extrema – Deflection angle of each waveform.
2. Z Position of Extrema – Amplitude of each waveform.
3. Gaussian Width – Width of each waveform.

This experiment defined the maximum/minimum angle of extrema possible as:

This experiment defined the maximum/minimum Z of extrema possible as:

This experiment defined the maximum/minimum width possible as:

Where the arrays refer to the parameter setting for a specific feature ([P, Q, R, S, T]).

Latin Hypercube sampling [] was employed to generate 10,000 random samples between the defined limits. This ensured that each signal generated had different morphologies, and that each morphology generated was realistic. By realistic we mean the waveforms are technically possible to be generated by a human heart.

It was also necessary to generate signals with different Heart Rates (HR), the available HR’s were set as :

We generate 10,000 signals per HR and thus, the final dataset will have 60,000 ECG signals. Each clean ECG signal is stored in .mat format in a unique directory.

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 2 : Algorithm to construct clean ECG signals.** | | | |
|  | ***Input***:  HR\_TO\_GENERATE  MIN\_ANGLES\_OF\_EXTREMA  MAX\_ANGLES\_OF\_EXTREMA  MIN\_Z\_POSITION\_OF\_EXTRAMA  MAX\_Z\_POSITION\_OF\_EXTRAMA  MIN\_GAUSSIAN\_WIDTH  MAX\_GAUSSIAN\_WIDTH | | |
|  | Output: A database of clean ECG signals with different morphologies and Heart Rates. | | |
| **1** | Sampling 🡨 Generate 10,000 random sample parameters of our inputs. | | |
| **2** | For (iHeartRate = 1 : numberOfHeartRates) // Loop through all heart rates defined. | | |
| **3** |  | Define the HR to generate 10,000 signals for. | |
| **4** |  | for (iSignal = 1 : numberOfSignalsPerHR) // Loop through each of the 10000 signals | |
| **5** |  |  | **Define parameter settings:** *Waveform angle, waveform amplitude, width size.* |
| **6** |  |  | ***Try:***  *Call ECGSYN function with specific parameter settings.*  ***Catch:***  ***Continue***  ***end*** |
| **7** |  |  | ***Save :*** *Save the signal with HR and signal number in filename.* |
| **8** | ***Save :*** *Save the parameter setting matrix.* | | |

Figure 6 Algorithm design of clean ECG signal database.

### **Creation of noisy database.**

At this stage, there exists a 30 minute EM noise file contained in a directory, and 60, 000 ECG signals contained in another directory. The aim of this processing step was to combine the noise and ECG signals such that a noisy signal was produced for each ECG shape. This experiment generated noise at Signal To Noise Ratio’s of 0, 6, 12, 18 and 24 dB. 30 second ECG strips were generated and 10 noisy signals were generated per clean ECG, details of this are provided below.

The noise signal is first read into MATLAB, we then loop through 30 second segments of the 30 minute noise file and model further shapes of the noise, this is done to extend the noise shapes that may not be contained in the 30 minute strip. This is done using an auto-regressive model [] which is used to predict further noise shapes.

We then iterate through each clean ECG signal generated and read in the corresponding R-Peak positions. (These are returned from ECGSYN). The R-peak locations are used calculate the peak to peak amplitude and power, which in turn allow the specified SNR levels to be determined.

The overall algorithm can be seen below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 3 : Algorithm to construct Noisy ECG signals.** | | | |
|  | ***Input***: Path to noise signal  Path to clean ECG signals  Sampling frequency  ECG Length  Number of noise sections  SNR Levels  Number of generated noise signals. | | |
|  | Output: A database of noisy ECG signals with different morphologies and Heart Rates. | | |
| **1** | Load Noise Data 🡨 Read in 30 minute noise signal | | |
| **2** | Calculate Minimum and Maximum ECG length. | | |
| **3** |  | Define the HR to generate 10,000 signals for. | |
| **4** |  | for (iSignal = 1 : numberOfSignalsPerHR) // Loop through each of the 10000 signals | |
| **5** |  |  | **Define parameter settings:** *Waveform angle, waveform amplitude, width size.* |
| **6** |  |  | ***Try:***  *Call ECGSYN function with specific parameter settings.*  ***Catch:***  ***Continue***  ***end*** |
| **7** |  |  | ***Save :*** *Save the signal with HR and signal number in filename.* |
| **8** | ***Save :*** *Save the parameter setting matrix.* | | |

Future work comprises of development and testing of Intelligent algorithms.

There were a number of problems encountered in the database development. The parameter setting stage when generating clean signals has been difficult as they have been set randomly based on a reasonable estimation, however in practice these may not be realistic.

There is no guarantee that the training dataset will provide useful when we develop an algorithm and apply it to unseen data. There are a lot of pre-processing techniques, and a few assumptions that could mean the training data is not representative of real ECG signals. However this is unknown until we test the algorithm.

# **Conclusion**

# **References**

[1] **https://archive.physionet.org/physiobank/database/macecgdb/**

[X] [ben120-web/MSc-Project: This repository will contain the codebase used to develop various models to remove electrode motion noise from ECG signals using learning models. (github.com)](https://github.com/ben120-web/MSc-Project)