ENEL420 - Genetic Algorithms in Digital Signal Processing

Department of Electrical and Computer Engineering
University of Canterbury

Luke Trenberth (ID: 47277086)
Hassan Alhujhoj (ID: 35352633)

October 16, 2020

Abstract

I would like to say some bullshit here that summaries my report in an interesting way.

Contents

1 Introduction

Genetic Algorithms (GA) are inspired by the mechanism of natural selection where the strongest and fittest individuals would likely be the winners in a competing environment. Genetic Algorithm is used as a direct analogy of such natural evolution where it presumes that a potential solution of a problem is an individual and can be represented by a set of parameters. These sets of parameters are regarded as the genes of a chromosome and can be structured by a string of values in binary form. A fitness value is used to reflect the degree of goodness of the chromosome for the problem which would be highly related with its objective value [1].

History has shown that the fitter chromosome tends to yield good quality offspring which means a better solution to the problem. Practically, a population pool of chromosomes must be randomly set initially. The size of this population varies from one problem to the other. Each cycle of genetic operation is termed as an evolving process where a subsequent generation is created from the chromosomes in the current population. This evolving process can only be succeeded if a group of those chromosomes, which are generally called "parents" or a collection term "mating pool" are selected. The genes of the parents are then mixed to produce offspring in the next generation. From this manipulation of genes process, the "better" chromosome will create a larger number of offspring, and thus has a higher chance of survival in the subsequent generation, emulating the survival-of-the-fittest mechanism in nature [1].

To make sure the desired termination criterion is reached, the cycle of evolution is repeated. The offspring of the previous generation are reinserted into the model, further yielding higher quality offsprings [1].

There are two fundamental operators that facilitate the evolution cycle: Crossover and Mutation. Both operators are required for such a process even though the selection routine. To further illustrate the crossover procedure, the one-point crossover mechanism is shown in Figure 1. Genes are exchanged between parents to form offspring. Mutations are randomly generated after crossover with a small probability of occurrence.

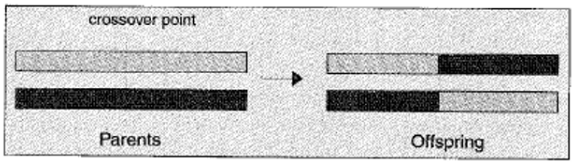


Figure 1: Interference frequencies present in the ECG signal.

2 Background

2.1 Digital Signal Processing of ECG Signals

In assignment one, a noisy ECG signal with 1024Hz sampling frequency was provided to be filtered. The assignment required the implementation of a notch filter with either an FIR or IIR filter. An FIR or IIR notch filter was suited to filter this ECG signal since there were a clear two interference frequencies present within the frequency spectrum of the ECG signal. These interference frequencies were identified to be $f_1 = 31.456Hz$ and $f_2 = 74.36Hz$ as shown in Figure 2. Interference frequencies from other ECG signals can be found to be between $30Hz \leqslant f \leqslant 100Hz$. It should be noted that the first peak in Figure 2 is the DC component due to the use FFT to get the frequency response of

the time domain ECG signal.

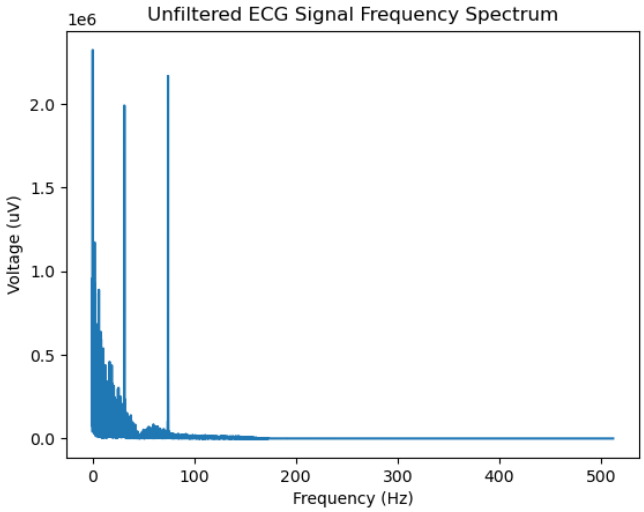


Figure 2: Interference frequencies present in the ECG signal.

One method to filter these two frequencies was to reject them with either a window or Parks-McClellan filters.

2.2 Genetic Algorithms

Since the emergence of Darwin's theory of natural selection, GA has become a powerful tool and used in various applications. The GAs are also known as optimisation algorithms. Optimisation algorithms have two major classes. These two classes are classified as calculus-based techniques and enumerative techniques. Calculus-based optimisation algorithms employ the gradient-directed searching mechanism to solve error surfaces or differentiable surfaces of an objective function. A common misuse of an objective function can occur with an ill-defined or multimodal objective function. This can lead to obtaining a local optima instead of a global optima. Such use of objective functions are common in signal processing [2].

Genetic Algorithms have a general cycle known as GA cycle, shown in Figure 3. This cycle is a searching process based on the laws of natural selection and genetics. A simple GA consists of three operations: Selection, Genetic Operation and Replacement. GA population comprises of a group of chromosomes from which candidates can be selected for the solution of a problem. The population, initially, is generated randomly. The fitness values of all chromosomes are evaluated by calculating the objective function in a decoded form (phenotype). to generate the offspring by the defined genetic operations, a particular group of chromosomes (parents) are selected from the total population. The fitness of the offspring is evaluated in a similar fashion to their parent chromosomes. The chromosomes in the current population are then replaced by their offspring based on a certain replacement strategy defined by the user [2].

This GA cycle is then repeated until a desired termination criterion is reached. For example a predefined number of generations is produced. If all goes well and according to this process of simulated evolution, the best chromosomes is the final population can become a highly evolved solution to the problem. Generally, GA follows the following process [2].

1. Randomly generate an initial population $X(O) := (x_1, x_2, ..., x_N)$.

2. Compute the fitness $F(x_i)$ of each chromosome x_i in the current population $X(t)$;
3. Create new chromosomes $X_r(t)$ by mating current chromosomes, applying mutation and recombination (crossover) as the parent chromosomes mate;
4. Delete numbers of the population to make room for the new chromosomes;
5. Compute the fittness of $X_r(t)$, and insert these into the popuation;
6. $t := t + 1$, if not (end-test) go to step 3, or else stop an dreturn the best chromosome.

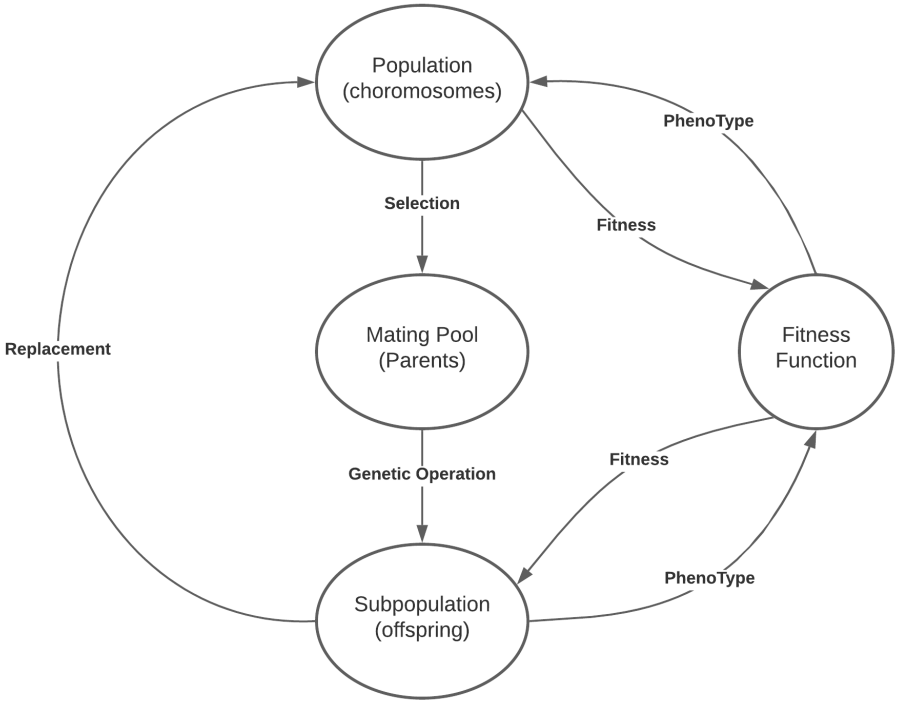


Figure 3: Genetic Algorithms cycle. Adapted from [2]

2.2.1 Crossover

Crossover is a GA operator which is a recombination operator that combines subparts of two parents chromosomes to produce offsprings that contain some parts of both parents' genetic material. Crossover is considered by many GA practitioners to be the determining factor that distinguishes the GA from all other optimisation algorithms [2].

2.2.2 Mutation

Mutation is another operator that introduces variations into the chromosomes. This variation can be local or global. Mutation can occur some occasionally but can randomly alters the value of a string position. A randomly generated bit can replace any bit of the chromosome bitstring mutating the original bit sequence of the parents [2].

2.3 Applications

Genetic Algorithms are widely used and can be applied in many applications such as in analog signal processing, digital signal processing, DSP MCU scheduling, FIR filter design, PID tuning and PID

control, etc [3, 4, 5, 6, 7, 8].

In [3], GA were used to control a BLDC motor with PI speed controller that is then processed by a Digital Signal Processor (DSP). In a real control system, the processing time in a cycle might be token long when the data communication is processed. The implementation of a GA-PI control design improves the performance of the PI control of the BLDC motor according to [3]. In this study, the BLDC motor fitness is evaluated based on the transfer function of the motor which is then used to evaluate the chromosomes population

3 Method

In this assignment, genetic algorithms were used to design a bandstop Finite Impulse Response (FIR) filter for Assignment 1's ECG data. To design a filter, the filtered frequencies, the bandwidth and band transition width were required characteristics. These were set as the genes for the filter and each chromosome contained these parameters. A population was then initialised with p_{size} chromosomes, each containing the genes to create a filter. For each chromosome, the filter created by its characteristics, returning a fitness score, with higher scores representing higher quality filters. The genetic algorithm used the fitness scores of the previous generation to generate a new generation, where the highest n_{parent} scores become the parents of the succeeding generation. utilising genetic crossover, variation and mutation. An identical process is then conducted on the succeeding generation. As the number of iterations increases, the population fitness increases. Once $N_{generations}$ have been reached, the optimal filter coefficients are returned.

3.1 Digital Signal Processing Data

Due to accessible data samples, the ECG signals from Assignment 1 have been analysed. This data contained two interference frequencies between $f = 30 - 100Hz$. The interference frequencies must be identified and removed from the data. Each chromosome had a set of genes representing a filter property. The genes were represented as decimal numbers, known as decimal encoding, which is useful in scenarios that require a high level of accuracy.

A python class was created to analyse the signal. This class was able to convert the time domain signal to the frequency domain, giving a plot of the signal. This class was able to conduct filtering techniques such as Window Filtering, Frequency Sampling and Parks-McLellan Filtering. Plots of filtered and unfiltered data was generated. This allowed the data to be analysed with a level of abstraction.

3.2 Fitness Function

As each characteristic will produce a different gain, it was important to quantify how effective each filter was. The Signal to Noise Ratio (SNR) is considered a useful metric for choosing the most optimal filtering coefficients. For each chromosome, a Parks-McLellan filter was generated, and then applied to the original data. The filtered signal power was then calculated by considering the variance in the signal. The difference between the filtered power and the original signal power was then considered the noise power, and the signal to noise power was calculated. Signals with higher SNRs were considered more optimal filtering, and therefore the SNR of the signal was determined to be an appropriate measure of the sample's fitness.

$$SNR = var(y_0) - var(y) \tag{1}$$

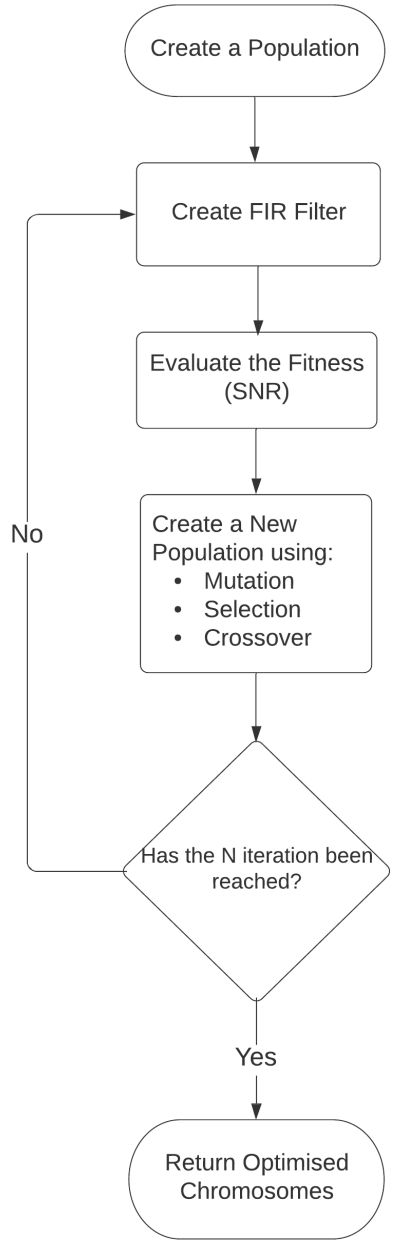


Figure 4: Flowchart diagram of the use of GA to process ECG signals.

3.3 Population Selection

Parent chromosomes were chosen by taking the chromosomes with the largest fitness scores in the previous generation, known as elitest selection. Chromosomes with the largest fitness scores have the most successful genes and were incorporated into the successor generation gene pool. For the analysed ECG data set, a population of 20 chromosomes was used, with four parents succeeding to the following generation. The effect of modifying each of these results on the success of the algorithm and the number of iterations for success was investigated.

Successive chromosomes were created through a combination of genetic mutation, and crossover of the parent chromosomes. Genetic mutation was incorporated into the genetic algorithm by random variation in the genetic sequences, using numpy's $np.randint()$ function. This allowed the

frequency genes to span the entire frequency range. Genetic crossover was implemented to find optimal combinations of known genes in the gene pool. This mixes genes from parent chromosomes. The effectiveness of found gene combinations was assessed in the next generation, ideally finding combinations of chromosomes that yield higher fitness. Any combinations with higher fitness than the original are then given a higher selection priority.

3.4 Finite Impulse Response (FIR) Filtering Techniques

3.4.1 Window Filter

Plz write something here.

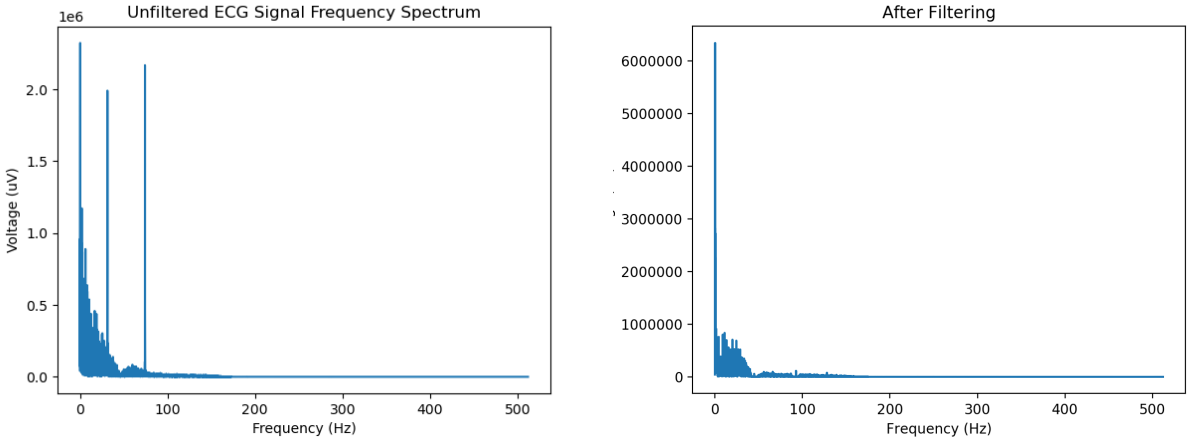
3.4.2 Parks-McClellan Filter

Filters were initially designed using the Parks-McClellan algorithm. This algorithm finds the optimal Chebyshev FIR filter using iteration. For each interference frequency identified, a Chebychev filter was created. All filters were then convolved to the original data frequency responses. Ideally, this removed the interference at the given frequencies from the signal without affecting the remaining frequency response of the signal.

4 Results

4.1

When the Genetic Algorithm was run using a Parks-McLellan filter, the algorithm identified the optimal genes for maximum SNR. With the filter population size of $p_{size} = 20$, and a maximum number of iterations of $n = 1000$, a filtered output was produced. As seen in Figure 5, the interference frequencies found in the original signal have been removed from the frequency plot. When data from other files was tested, these were also successful without changing parameters. This proved the fitness function was robust as interference frequencies were identified in various conditions without needing data training.



(a) Before filtering.

(b) After filtering.

Figure 5: The filtered output of ECG signal 1, using $p_{size} = 20$, $n = 1000$.

When contrasting the number of iterations with the population frequency, it was found the maximum fitness consistently reached the maximum population fitness within 50 iterations. However,

when the population size was decreased, a larger number of iterations was required to reach optimum fitness. Using $p_{size} = 20$, $n = 50$ gave reasonable filter results with a quick calculation time.

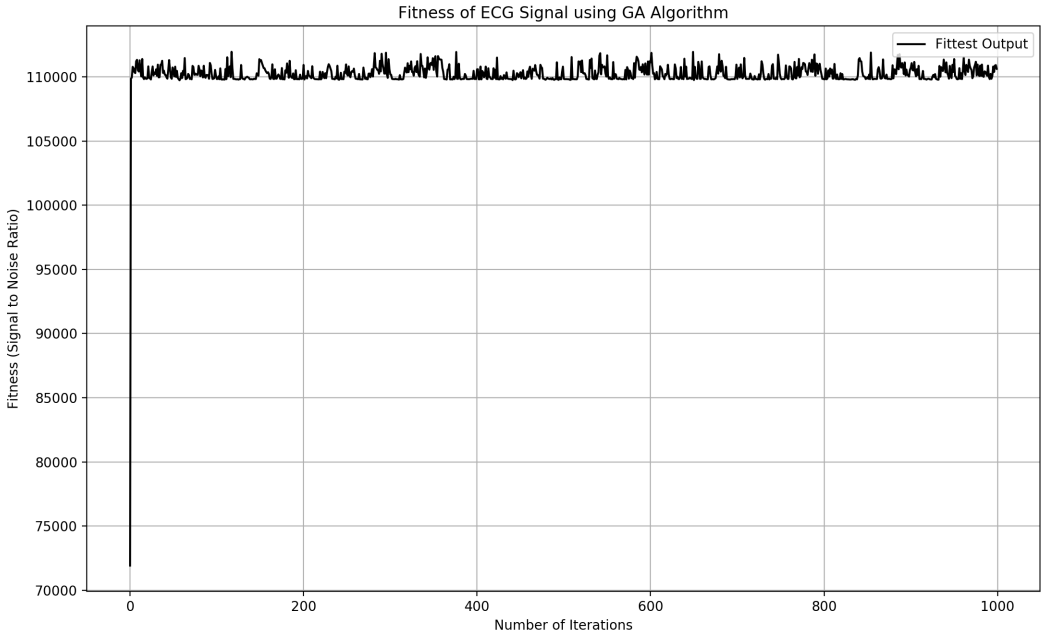


Figure 6: The population fitness of ECG signal 1 through iterations, using $p_{size} = 20$, $n = 1000$.

5 Discussion

The Genetic Algorithm has been effective at identifying and filtering the interference frequencies. These algorithms allow quick identification of required parameters without a comprehensive understanding of the system. Genetic Algorithms provide flexible, robust solutions to filter design. These can also be applied to with varying parameters without requiring mathematical derivations. Unlike other iterative methods, Genetic Algorithms can span the entire solution space to find optimal solutions, and when given appropriate parameters, the system will not misidentify suboptimal solutions in local maximums. However, Genetic Algorithms require a fitness function to assess the quality of each potential solution quickly. This can be difficult in systems that require significant time for data gathering. GA's are also computationally intensive.

The GA used to design filter coefficients emulates these properties. Once a population and fitness function was created, and appropriate system parameters were identified, the algorithm was consistently able to return high-quality results. By inspection, these solutions identified the optimal results to high degrees of accuracy. However, the algorithm took a reasonable time to produce results. This may not be practical in other applications.

To improve the filtering capabilities, the filtering method could be incorporated into the chromosomes, which would allow the program to identify the optimal filter technique. This would automate the type of filter, allowing the program to optimise for the type of filter as well as the filter characteristics.

6 Conclusion

Reinstate the stuff you've talked about in the report. Don't introduce new materials in here.

7 References

[1] K. F. Man and K. S. Tang, "Genetic algorithms for control and signal processing," in *Proceedings of the IECON'97 23rd International Conference on Industrial Electronics, Control, and Instrumentation (Cat. No.97CH36066)*, vol. 4, 1997, pp. 1541–1555 vol.4. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.canterbury.ac.nz/document/664911

[2] W. K. Tang, K. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *Signal Processing Magazine, IEEE*, vol. 13, pp. 22 – 37, 12 1996. [Online]. Available: https://ieeexplore.ieee.org/document/543973

[3] G. Y. Chen and J. Perng, "Pi speed controller design based on ga with time delay for bldc motor using dsp," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2017, pp. 1174–1179. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.canterbury.ac.nz/document/8015983

[4] P. Fleming and C. Fonseca, "Genetic algorithms in control systems engineering," *IFAC Proceedings Volumes*, vol. 26, no. 2, Part 2, pp. 605 – 612, 1993, 12th Triennal Wold Congress of the International Federation of Automatic control. Volume 2 Robust Control, Design and Software, Sydney, Australia, 18-23 July. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S147466701749015X

[5] D. Cao, A. Modiri, G. Sureka, and K. Kiasaleh, "Dsp implementation of the particle swarm and genetic algorithms for real-time design of thinned array antennas," *IEEE Antennas and Wireless Propagation Letters*, vol. 11, pp. 1170–1173, 2012. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.canterbury.ac.nz/document/6317130

[6] P. Fabijanski and R. Lagoda, "On-line pid controller tuning using genetic algorithm and dsp pc board," in *2008 13th International Power Electronics and Motion Control Conference*, 2008, pp. 2087–2090. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.canterbury.ac.nz/document/4635574

[7] T. Miyata, H. Asou, and N. Aikawa, "Design method for fir filter with variable multiple elements of stopband using genetic algorithm," in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, 2018, pp. 1–4. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.canterbury.ac.nz/document/8631793

[8] R. W. Amphlett and D. R. Bull, "Genetic algorithm based dsp multiprocessor scheduling," in *1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96*, vol. 2, 1996, pp. 253–256 vol.2. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.canterbury.ac.nz/document/540400