

UNIVERSIDADE DA BEIRA INTERIOR - UBI

APRENDIZAGEM AUTOMÁTICA

Filtragem Adaptativa em Sinais de Eletrocardiograma

Thiago Machado

12 de Janeiro, 2021



Conteúdo

1	Introdução	1
2	Descrição da Situação Atual	2
3	Desenvolvimento	3
3.1	Filtragem Adaptativa	3
3.2	Computação Evolutiva	4
3.2.1	Algoritmo Genético	4
3.2.2	Codificação de Indivíduos	6
3.2.3	Definição de População Inicial . . .	6
3.2.4	Operador de Crossover	7
3.2.5	Operador de Mutação	8
3.2.6	Seleção de Indivíduos para a Próxima Geração	8
4	Descrição do Trabalho	9
5	Experimentação	11
5.1	Protocolo de Recolha de Dados	11
5.2	Métricas de avaliação	11
5.3	Dados Iniciais e Resultados	11
6	Conclusão	13
	Referências	14

1 Introdução

O eletrocardiograma (ECG) é uma ferramenta utilizada por profissionais da área da saúde para identificar patologias. Nele estão contidas informações diversas de um indivíduo, tais como as condições do tecido muscular cardíaco. Esta, por sua vez, pode detectar alteração em algum local do coração por meio de diferenças na morfologia da forma de onda representada no ECG. Exemplificando, é possível detectar células mortas nos ventrículos através da visualização de um complexo de ondas no sinal de ECG denominado de complexo QRS (mais detalhes na seção seguinte). As variáveis temporais do sinal de ECG também podem ser vistas como fatores de identificação de patologias, uma vez que doenças como arritmia são causadas pelo batimento cardíaco desordenado. Além disso, o ECG se faz único a cada indivíduo, possibilitando a utilização como biometria. Por esse motivo, são necessários meios de redução de ruídos, pois a presença do ruído traz uma falsa informação sobre o indivíduo, podendo falhar na tarefa de identificá-lo ou de detectar uma patologia.

Dentre os mais variados tipos de ruído que podem contaminar um sinal de ECG, merece destaque o ruído impulsivo (RI), pois a degradação por ele causada pode comprometer sobremaneira a análise do sinal. Diversos aparatos foram projetados especificamente para eliminação do RI.

2 Descrição da Situação Atual

Há vários tipos de filtros utilizados comumente no aparelho de ECG, tal como filtro passa alta, passa baixa, filtro Notch e filtros adaptativos, como Mínimo Quadrado Médio (LMS). [Ahammed, 2018] demonstra a utilização do filtro Notch para remoção do ruído de Linha de Base (0,5Hz), que normalmente é causado pela movimentação do paciente e o ruído proveniente da rede elétrica (50Hz ou 60Hz) e sequentemente, faz uma comparação com a eficácia do filtro adaptativo LMS. Utilizando a Relação-Sinal-Ruído (SNR) e o Erro Quadrático Médio (EQM) como métricas de avaliação do desempenho, o LMS é mais eficaz para eliminação do ruído. Ainda em relação aos filtros adaptativos, na pesquisa feita por [Bhogeshwar et al., 2014], foram utilizados os filtros adaptativos FIR (Finite Impulse Response) e IIR (Infinite Impulse Response) para remoção do ruído branco e o ruído de 50Hz oriundo da rede elétrica a fim de comparar a eficácia de ambos. Utilizando o SNR como métrica, concluiu-se que o filtro FIR teve um melhor desempenho.

3 Desenvolvimento

Este trabalho utiliza de conceitos de Filtragem Adaptativa combinados com os processos da Computação Evolutiva, mais especificamente o Algoritmo Genético (GA) para resolver o problema de ruídos no ECG. Tais temas serão abordados com mais profundidade nas sessões a seguir.

3.1 Filtragem Adaptativa

O processo de filtragem se trata de um bloco de M coeficientes que operam como uma "convolução" ou, com o termo mais adequado, a resposta ao impulso. A resposta ao impulso é a saída de um sistema ao se aplicar um impulso ao sinal. Por meio dessa abordagem, há dois tipos de filtros: o IIR (Infinite Impulse Response) e o FIR (Finite Impulse Response), no qual o mais conhecido é o LMS (Least Mean Square) que tem o intuito de se minimizar o erro quadrático médio.

A construção de um filtro LMS dá início em uma série de coeficientes que são multiplicados pelas amostras de entrada na medida que o sinal de entrada avança dentro do sistema, como mostrado na figura 1. A seguir, as multiplicações são somadas, resultando em uma amostra \mathbf{Y} . Essa amostra é comparada uma amostra de um sinal desejado, resultando em um valor de erro.

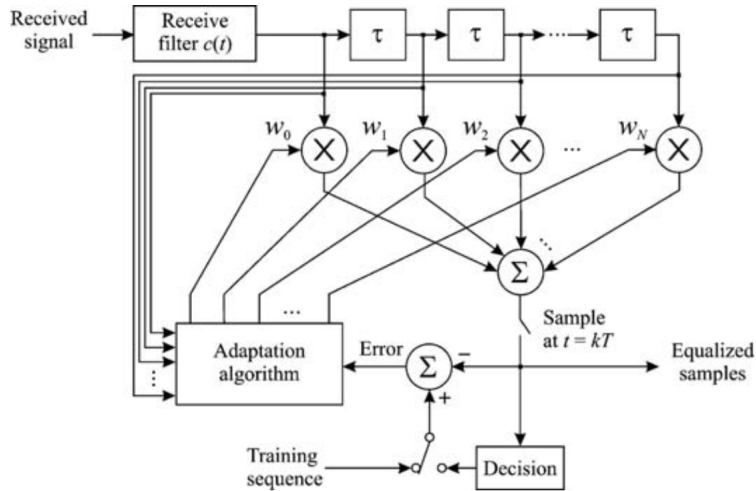


Figura 1: Representação de um filtro FIR.

O filtro assume a posição de treinamento para fazer as alterações nos coeficientes do filtro até que o erro quadrático médio seja aproximadamente zero. O filtro, então, assume que o sinal de treinamento seja igual o sinal desejado e muda para a posição de decisão. Nesse momento, o sinal de entrada é comparado com o sinal equalizado que foi produzido anteriormente e um novo erro é produzido [Guimaraes, 2010].

3.2 Computação Evolutiva

Os sistemas baseados em computação evolutiva possuem uma população de soluções potenciais, aplicam processos de seleção baseados na adaptação de um indivíduo e também empregam outros operadores “genéticos”. Os principais propostas de computação evolutiva encontradas nas literaturas utilizadas nesse documento são:

- algoritmos genéticos;
- estratégias evolutivas;
- programação evolutiva.

3.2.1 Algoritmo Genético

Os GAs possuem a função de formalizar matematicamente e explicar processos de adaptação em sistemas naturais e desenvolver sistemas artificiais que retenham mecanismos originais encontrados em sistemas naturais. Os GAs empregam o *crossover* e a *mutação*¹. Uma extensão dos GAs é a *programação genética*, que consiste em evoluir programas de computador baseados nos conceitos de evolução natural. Atualmente, a programação genética vem sendo empregada para resolver uma gama de problemas complexos, assim como na síntese de circuitos elétricos ou na definição de arquiteturas de redes neurais artificiais.

Os GAs empregam a terminologia originada da teoria da evolução e da genética. O processo da evolução se dá por um método de pesquisa de

¹A mutação surge quando dois pais geram um filho com características que não estavam presentes em nenhum dos pais. Embora o nome pareça algo negativo, é extremamente importante para geração de novos genes que são melhores adaptados ao ambiente.

buscas em um amplo espaço de soluções para o problema. Esta busca requer um equilíbrio entre dois objetivos aparentemente opostos: o aproveitamento das melhores solução e a exploração do espaço de busca (exploitation x exploration)[Michalewicz and Schoenauer, 1996]. Este equilíbrio ainda está muito longe se considerar as seguintes técnicas de busca:

- métodos de otimização clássicos, como o método do gradiente que só aproveita a melhor solução na busca sem realizar uma pesquisa no espaço de busca.
- métodos de busca aleatória que, no caso, realiza uma pesquisa ampla no espaço de busca mas ignora regiões promissoras do espaço.

Portanto, os GAs fornecem uma solução equilibrada entre aproveitamento das melhores soluções e exploração do espaço de busca. O método que se faz isso é guardando uma população de soluções candidatas enquanto métodos alternativos processam um único ponto no espaço de busca a cada constante. Além disso, estes algoritmos não são considerados determinísticos e nem puramente aleatórios pois combinam passos de variações aleatórias com seleção, polarizadas pelos valores de adequação (*fitness*) atribuído ao indivíduo.

O processo de busca é, portanto, multi-direcional, através da manutenção de soluções candidatas, e encorajando a troca de informação entre as direções. A cada geração, soluções relativamente “boas” se reproduzem, enquanto que soluções relativamente “ruins” são eliminadas. Para fazer a distinção entre diferentes soluções, é empregada uma função de avaliação ou de adaptabilidade (*fitness*) que simula o papel da pressão exercida pelo ambiente sobre o indivíduo. Seguindo a estrutura básica de um algoritmo evolutivo, podemos descrever um GA como segue [Michalewicz and Schoenauer, 1996]:

- durante a iteração t , um GA guarda uma população de soluções potenciais (indivíduos, cromossomos, lista de atributos ou vetores) $P(t) = \{x_1^t, \dots, x_i^t\}$;
- cada solução x_i^t é avaliada e produz sua medida de adaptação (*fitness*);
- uma nova população (iteração $t + 1$) é então formada privilegiando a participação dos indivíduos mais adaptados;
- alguns membros da nova iteração passam por novas alterações, por meio de *crossover* e mutação para formar novas soluções potenciais;

- esse processo se repete até que se atinja um número pré determinado de iterações seja atingido ou até que um fitness esperado seja alcançado.

Um GA para um problema particular deve ter os seguintes componentes:

- uma representação genética para soluções potenciais (processo de codificação);
- uma maneira de criar uma população inicial de soluções potenciais;
- uma função de avaliação que faz o papel da pressão ambiental, capaz de classificar as soluções pela sua capacidade de resolver o problema;
- operadores genéticos;
- valores para os diversos parâmetros usados pelo GA (tamanho da população, probabilidades de aplicação dos operadores genéticos, etc).

3.2.2 Codificação de Indivíduos

A etapa de codificação é um dos processos mais críticos na definição de GA, pois uma escolha errada pode levar a uma convergência prematura do algoritmo. A estrutura de um cromossomo deve representar uma solução como um todo e deve a fazer da maneira mais simples possível.

Em problemas de otimização restrita, a codificação utilizada pode fazer com que os indivíduos alterados por *crossover*/mutação sejam inválidos. Por esse motivo, cuidados especiais devem ser tomados na escolha da codificação correta.

Em um modelo clássico, a codificação que geralmente é utilizada é o de arranjo binário, com tamanho fixo, assim como demonstrado por [Holland et al., 1992]. Porém esse modelo ainda tem algumas limitações, como em problemas com alta dimensionalidade e precisão.

3.2.3 Definição de População Inicial

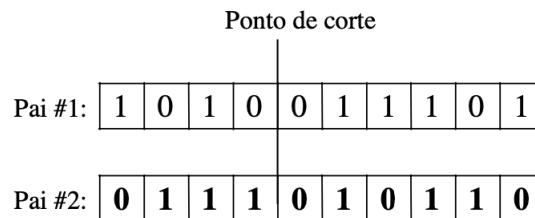
O método mais comum da criação de população é a de inicialização aleatória dos indivíduos. Se alguma informação inicial a respeito do problema estiver disponível, pode ser utilizado na criação da população. Por exemplo, na codificação binária, se é conhecido que o número de 0's é maior que o número

de 1's, a inicialização pode ser feita mesmo que não seja conhecido a proporção. Porém, isso não deve ser utilizado em problemas de restrição, pois pode gerar indivíduos inválidos.

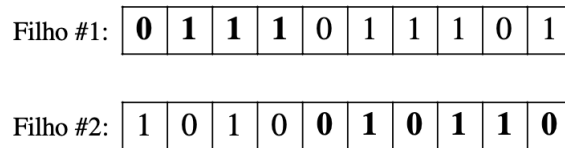
3.2.4 Operador de Crossover

O crossover (ou recombinação) é a união de dois indivíduos que gera um novo indivíduo com características herdadas dos pais. A ideia por trás do operador de *crossover* é a de troca de informação por parte das soluções candidatas. No GA clássico é atribuída uma probabilidade de crossover fixa aos indivíduos da população.

O crossover mais utilizado atualmente é o crossover de um ponto. Ele se baseia em dois indivíduos (pais) que, a partir de seus cromossomos, são gerados mais dois indivíduos (filhos). Para gerar os filhos, seleciona-se um mesmo ponto de corte aleatoriamente nos cromossomos dos pais, e os segmentos de cromossomo criados a partir do ponto de corte são trocados. Considere, por exemplo, dois indivíduos selecionados como pais a partir da população inicial de um GA e suponhamos que o ponto de corte escolhido (aleatoriamente) encontra-se entre as posições 4 e 5 dos cromossomos dos pais:



Após o crossover, teremos os seguintes filhos:



Há, ainda, vários outros tipos de operadores de crossover que também são muito utilizados atualmente. Tais como o *crossover de dois pontos* ou o *crossover uniforme*. Porém, nenhum deles se mostra mais eficiente que o outro para resolver problemas genéricos. Cada um possui uma eficiência

em um caso e ineficiência nos demais casos, portanto é de escolha pessoal, dependendo do problema que se encontra.

3.2.5 Operador de Mutação

O operador de mutação modifica um ou mais genes de um cromossomo proveniente dos indivíduos pais. A probabilidade de ocorrer a mutação de um gene é denominada *taxa de mutação*. Normalmente, são atribuídos valores pequenos para essa variável. A ideia por trás da taxa de mutação é de criar uma variabilidade extra na população sem destruir o progresso já obtido com a busca.

3.2.6 Seleção de Indivíduos para a Próxima Geração

Dentre os métodos utilizados para seleção da próxima geração, o mais clássico é o chamado de *roulette wheel*, na qual consiste em medir o *fitness* dos indivíduos e dividir pelo somatório do *fitness* de todos os indivíduos. Assim, todos os indivíduos são colocados em probabilidade de aparecer na próxima geração. Porém, através dessa técnica, o indivíduo com maior *fitness* pode não vir a aparecer. Há, também, a uma técnica chamada de *elitista*, pois além de observar os *fitness* de todos os indivíduos, ele ainda garante que o indivíduo de maior *fitness*.

Os possíveis demais mecanismos de seleção são:

- Seleção por diversidade: são selecionados os indivíduos mais diversos da população.
- Seleção bi-clássica: são selecionados os $P\%$ melhores indivíduos e os $(100 - P)\%$ piores indivíduos da população.
- Seleção aleatória: São selecionados N indivíduos aleatoriamente, podendo, ainda, ser classificado como:
 - Salvacionista: seleciona-se o melhor indivíduo e os demais aleatoriamente.
 - Não-salvacionista: seleciona-se aleatoriamente todos os indivíduos.

4 Descrição do Trabalho

O início do trabalho foi feito utilizando os primeiros conceitos da filtragem adaptativa. A diferença está na forma com que os coeficientes são atualizados. Neste trabalho, a forma com que os coeficientes são atualizado é através do GA.

o GA se baseia em uma sequência bem definida de passos para ser implementada, dessa forma, foram seguidos os seguintes caminhos:

- Codificação: Foi utilizada a codificação real, de forma que os números foram representados por eles mesmo.
- Definição de população inicial: Foi gerada uma população de popsize indivíduos de valores aleatórios entre -1 e 1 .
- Operadores genéticos: Foi utilizado uma taxa de crossover e mutação.

Além disso, o modo em que o crossover foi implementado foi utilizando o conceito de *Piscina de Cruzamento*. Esse conceito se baseia em uma população que passa por diversos torneios, selecionando 2 ou mais indivíduos de forma aleatória e o fitness desses indivíduos é testado para compor a piscina de cruzamento. Vale ressaltar que os indivíduos que são selecionados não deixam a população inicial, portanto, podem ser selecionados mais de uma vez, sendo repetidos na piscina. Esse processo é repetido até que a piscina esteja cheia com o mesmo tamanho da população inicial. O crossover, então é feito selecionando os indivíduos da piscina de cruzamento aos pares, produzindo pares de filhos, assim como mostra na figura 2

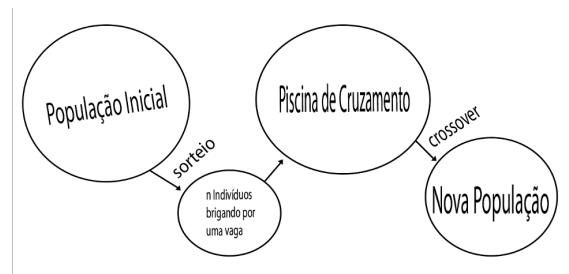


Figura 2: Representação da forma com que ocorre a Piscina de Cruzamento

A forma com que o cruzamento é feito é por meio de uma aleatoriedade selecionada. Os pais tem seus valores dos respectivos coeficientes comparados

e o coeficiente do filho tem um valor de coeficiente formado aleatoriamente entre os valores dos coeficientes dos pais com uma margem de erro de α para menos ou para mais, assim como mostra a figura 3. Repete-se esse procedimento para os demais coeficientes, formando um novo indivíduo não igual a nenhum dos pais. Além disso, ainda há uma probabilidade de o filho sofrer uma mutação em um número de 0 a n cromossomos.

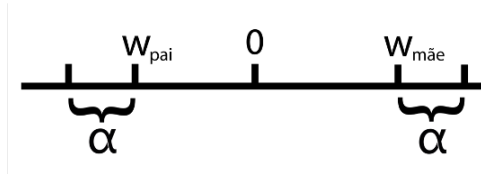


Figura 3: Caption

Esses filhos formarão uma nova população, que substituirá a população inicial. O número de gerações de filhos criados depende do desempenho dos filhos ou de um número de gerações pré definidos.

5 Experimentação

5.1 Protocolo de Recolha de Dados

A recolha de dados será feita através de uma função na biblioteca *scipy.signal*, capaz de gerar um sinal aproximado ao de um ECG real. A taxa de amostragem é de 800 amostras por segundo, num período de 6 segundos. Afim de classificar os filtros, o sinal de ECG foi adicionado sem o ruído, que foi adicionado posteriormente, com uma forma conhecida (que, no caso, é uma senoide). Logo após, foi gerado uma senoide de baixa amplitude e frequência para, sendo adicionado ao sinal, demonstrar o efeito do ruído de flutuação presente no ECG.

5.2 Métricas de avaliação

A métrica de avaliação de desempenho será utilizando o Signal to Noise Ratio (SNR) que é a razão entre a potência do sinal e a potência do ruído.

5.3 Dados Iniciais e Resultados

Os testes serão feitos utilizando o IDE *Spyder*, utilizando a linguagem *Python*.

Inicialmente, foi utilizado um filtro de 4 coeficientes.

O GA utilizou de uma população de 10 indivíduos, uma taxa de crossover de 90%, mutação em 10%. O α tem um valor de 0,2.

Segundo [Bhogeshwar et al., 2014], os filtros FIR (Finite Impulse Response) demonstraram o melhor resultado para eliminação de ruído no ECG, por esse motivo, foi escolhido o filtro Butterworth, com *ordem* = 2 com a finalidade de comparação com o modelo utilizado nesse artigo.

A figura 4 demonstra os sinais gerados, bem como o resultado da utilização dos filtros Butterworth e o filtro adaptativo.

Como método de classificação dos filtros, foi-se utilizado o SNR (Signal to Noise Ratio), portanto foi calculado o SNR médio dos mesmos sinais, como demonstrado na figura 5 afim de comparar os métodos utilizados.

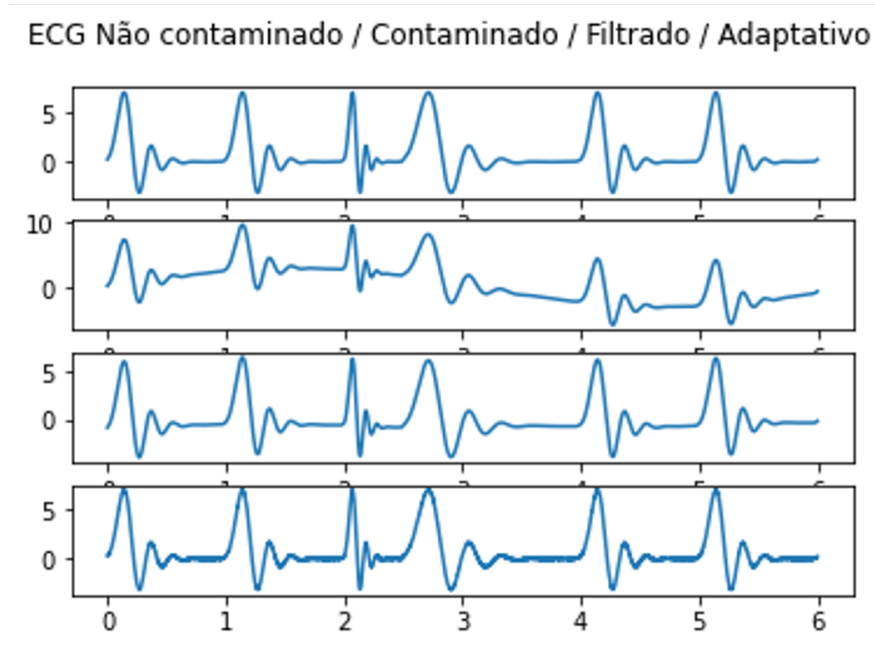


Figura 4: Representação do sinal de ECG gerado, na primeira linha. Na segunda linha, o ECG corrompido com o ruído de flutuação, na terceira linha, o ECG filtrado utilizando o filtro Butterworth com ordem = 2 e, na quarta linha, o sinal filtrado utilizando a filtragem adaptativa.

```
SNR medio ECG: 1.0635714647351495
SNR medio X: 2.122031734874304
SNR medio Y Filtrado (ButterWorth): 0.9336390304221497
SNR medio Y Filtrado (Adaptativo): 1.0628726868051128
```

Figura 5: SNR médio dos sinais de ECG. Na primeira linha, o ECG não contaminado pelo ruído, na segunda linha o ECG contaminado pelo ruído, na terceira linha, o sinal de ECG filtrado usando o Butterworth e, na quarta linha, utilizando o filtro adaptativo.

6 Conclusão

Segundo [Bhogeshwar et al., 2014], os filtros FIR (Finite Impulse Response) demonstraram o melhor resultado para eliminação de ruído no ECG. Por esse motivo, esse trabalho focou em uma nova maneira de filtrar o ECG através de um novo filtro FIR.

Como visto nos resultados mencionados, o filtro projetado possui a maior semelhança com o sinal original, esse fator pode ser observado através do SNR dos sinais. Porém, em uma situação real não há um sinal desejado para comparações, portanto conclui-se que foi utilizada a função de *equalizador*, como menciona [Guimaraes, 2010].

Para trabalhos futuros, é possível uma nova forma de calcular o erro através de um sinal estimado, dessa forma, este trabalho terá mais relevância, pois se mostrou promissor em sua tarefa. Além disso, também comprova a flexibilidade da computação evolutiva em resolver diversos problemas.

Referências

- [Ahammed, 2018] Ahammed, K. (2018). Removal of structured noise and base line wander from ecg signals via lms adaptive and fixed notch filter. *European Journal of Engineering Research and Science*, 3:12.
- [Bhogeshwar et al., 2014] Bhogeshwar, S. S., Soni, M. K., and Bansal, D. (2014). Design of simulink model to denoise ecg signal using various iir fir filters. In *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, pages 477–483.
- [Guimaraes, 2010] Guimaraes, D. A. (2010). *Digital transmission: a simulation-aided introduction with VisSim/Comm*. Springer Science & Business Media.
- [Holland et al., 1992] Holland, J. H. et al. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.
- [Michalewicz and Schoenauer, 1996] Michalewicz, Z. and Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*, 4(1):1–32.