



**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**PROGRAMA DE PÓS-GRADUAÇÃO EM**

**ENGENHARIA ELÉTRICA**

**ALGORITMO GENÉTICO, FERRAMENTA  
COMPUTACIONAL EM ESTIMATIVAS DE SINAIS  
PERIÓDICOS, APERIÓDICOS E CAÓTICOS.**

Evaldo Silva

Itajubá 2013



# **UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**PROGRAMA DE PÓS-GRADUAÇÃO EM**

**ENGENHARIA ELÉTRICA**

## **ALGORITMO GENÉTICO, FERRAMENTA COMPUTACIONAL EM ESTIMATIVAS DE SINAIS PERIÓDICOS, APERIÓDICOS E CAÓTICOS.**

Dissertação apresentada à  
Universidade Federal de Itajubá  
– Unifei, como parte dos  
requisitos para obtenção do  
título de Mestre em Engenharia  
Elétrica Área Automação e  
Sistemas Elétricos Industriais.

Orientador: Prof. Dr. Benedito  
Isaias de Lima Lopes

Itajubá 2013

Ficha catalográfica elaborada pela Biblioteca Mauá –  
Bibliotecária Margareth Ribeiro- CRB\_6/1700

S586a

Silva, Evaldo

Algoritmo genético: ferramenta computacional em estimativas de sinais periódicos, aperiódicos e caóticos / Evaldo Silva. -- Itajubá,(MG) : [s.n.], 2013.

103 p. : il.

Orientador: Prof. Dr. Benedito Isaias de Lima Lopes.  
Dissertação (Mestrado) – Universidade Federal de Itajubá.

1. Algoritmos genéticos. 2. Estimação de sinais. 3. Sinais periódicos. 4. Sinais aperiódicos. 5. Séries caóticas. I. Lopes, Benedito Isaias de Lima, orient. II. Universidade Federal de Itajubá. III. Título.



Ministério da Educação  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**  
 Criada pela Lei nº 10.435, de 24 de abril de 2002

## A N E X O II

### FOLHA DE JULGAMENTO DA BANCA EXAMINADORA PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Título da "Algoritmo Genético, Ferramenta Computacional em  
 Dissertação: Estimativas de Sinais Periódicos, Aperiódicos e Caóticos"

Autor: **Evaldo Silva**

#### JULGAMENTO

| Examinadores | Conceito  | Rubrica                                 |
|--------------|---|---|
|              | A = Aprovado - R = Reprovado - I = Insuficiente |   |
| 1º           | A   | <i>Cleverson Brighenti</i>              |
| 2º           | A   | <i>Antonio Carlos Zambroni de Souza</i> |
| 3º           | A   | <i>Benedito Isaias Lima Lopes</i>       |

#### Observações:

- (1) O Trabalho será considerado Aprovado se todos os Examinadores atribuírem conceito A.
  - (2) O Trabalho será considerado Reprovado se forem atribuídos pelos menos 2 conceitos R.
  - (3) O Trabalho será considerado Insuficiente (I) se for atribuído pelo menos um conceito R. Neste caso o candidato deverá apresentar novo trabalho. A banca deve definir como avaliar a nova versão da Dissertação.
- Este documento terá a validade de 60 (sessenta) dias a contar da data da defesa da Dissertação.

Resultado Final: Conceito: A, ou seja, aprovado

Observações: \_\_\_\_\_

Itajubá, 27 de Maio de 2013.

*Cleverson Brighenti*  
 Prof. Dr. Cleverson Brighenti  
 1º Examinador - ITA

*Antonio Carlos Zambroni de Souza*  
 Prof. Dr. Antonio Carlos Zambroni de Souza  
 2º Examinador - UNIFEI

*Benedito Isaias Lima Lopes*  
 Prof. Dr. Benedito Isaias Lima Lopes  
 1º Examinador (Orientador) - UNIFEI

*“Quem quiser chegar às estrelas não deve seguir as estradas, estas conduzem às cidades, a que se inventar um caminho, um meio, um modo.”*

*Dedico este trabalho a meus pais Vicente Silva (in memoriam) e Maria Aparecida Silva pela educação e honestidade ensinada, a minha filha Adriella Stormovski Silva pela oportunidade de educar.*

Agradecimentos:

Em primeiro lugar a DEUS por ter permitido realizar este trabalho.

Aos professores Dr. Carlos Alberto Murari Pinheiro, Dr. Antônio Carlos Zambroni de Souza, Dr. Germano Lambert Torres, Dr. Luiz Edival de Souza, Dr. José Wanderley Marangon Lima.

Ao coordenador do curso de Mestrado Prof. Dr. José Maria de Carvalho Filho e todos da administração do curso.

Ao Orientador Professor Dr. Benedito Isaias Lima Lopes pelos desafios propostos e pela orientação para enfrenta-los.

A minha filha Adriella pelo incentivo, admiração, sugestões etc.

A meus pais pelo tempo cedido para este estudo, apoio moral etc.

A meus alunos pelos incentivos.

## Resumo:

Engenharia é uma arte, e o estudo dos sinais é sem dúvida o entendimento de como a engenharia evolui, como descobre as razões e motivos do comportamento do mundo real. A engenharia possui um número incontável de ferramentas para esta análise e estudo de sinais, valendo-se de representações no tempo com uso de gráficos, que é a linguagem do engenheiro, nenhum problema passa sem um modelamento gráfico e levantamento de equações matemáticas, algébricas, uma integral ou derivada. Quando então as soluções começam a complicar uma nova ferramenta abre um novo mundo a Transformada de Laplace, a Transformada de Fourier. Problemas de difícil solução no domínio do tempo são tratados no domínio da frequência com simplicidade, em função também do uso da computação com sua contribuição na construção e aplicação de ferramentas, isto é, os programas permitem a aplicação de funções matemáticas, algébricas, transformadas, etc. Algumas melhorias podem ser propostas e usadas na solução de problemas, e este trabalho, cuja meta é apresentar uma ferramenta que possui um potencial muito grande no estudo de sinais e em um dos grandes problemas da engenharia que é a estimação de parâmetros que compõe um sinal, amplitude, frequência, fase, etc., os quais, se conhecidos, permitem repetir o sinal, modelá-lo, estudá-lo, decompô-lo, simulá-lo, etc. O Algoritmo Genético possui fácil entendimento, e sua capacidade de tratamento de multivariáveis o qualifica como uma possante ferramenta para uso na estimação de sinais periódicos, aperiódicos ou caóticos. Este estudo demonstra que o generalismo e paralelismo dos AGs permitem uma variedade de aplicações em novos produtos e novos processos, sua imunidade e robustez o qualificam como uma ferramenta de alto desempenho, com resultados precisos e baixo custo computacional.

Palavras chaves: Algoritmos genéticos, Estimação de sinais, sinais periódicos, sinais aperiódicos, séries caóticas.



## **Abstract:**

Engineering is an art and the study of signs. It is also undoubtedly the understanding of what it evolves and uncovers the reasons and motives of the behavior of the real world. Engineering has countless tools for the analysis and the study of signs, using representations in time with the usage of graphs, which are the language of the engineer. No problem is going to be solved without a graphical modeling, a survey of mathematical equations or an algebraic integral or derivative. When the solutions begin to complicate a new tool opens a new world as the Laplace transformation and the Fourier transformation. When problems are difficult to solve in time they are frequently processed with simplicity, but they also depend on the use of computers with their contribution to the construction and application of the tool. So programs allow the application of mathematical and algebraic process. Some improvements can be proposed and used to the solution of the problems. This work aims to provide a tool that has a great potential in the study of signs. One of the greatest problems of engineering is the estimation of the parameters that make up the signal, the amplitude, the frequency or the phase known. These parameters allow to repeat the signal modeling it, studying it, breaking it and simulating it. The Genetic Algorithm is easy to understand and its ability to treat multivariable qualifies makes it a powerful tool to be used in the estimation of periodic signals, aperiodic or chaotic. This study demonstrates that the generalization and the parallelism of GAs allow a variety of applications in new products and new processes. Its immunity and robustness qualify it as a high-performance tool with accurate results and low computational cost.

**Keywords:** Genetic algorithms, estimation of signals, periodic signals, aperiodic signals, chaotic series.

## Lista das Figuras

|   |    |
|---|----|
| Figura 1 Onda senoidal da rede elétrica de 60 Hz.....   | 18 |
| Figura 2 senoide de frequência ignorada próxima de 60 Hz, fase próxima 0 rd e amplitude próxima 1 pu. ....                        | 19 |
| Figura 3 Fluxograma do funcionamento do Algoritmo Genético .....  | 29 |
| Figura 4 estrutura básica de AG. Adaptado de Algoritmo Genético- Rio de Janeiro : Brasport 2006.....                              | 30 |
| Figura 5 distribuição de indivíduos no disco da roleta.....   | 37 |
| Figura 6 pontos de corte de strings com 5 bits .....  | 39 |
| Figura 7 Processo de cruzamento de um ponto adaptado de Algoritmo Genético- Rio de Janeiro : Brasport.....                        | 39 |
| Figura 8 Processo de cruzamento de dois pontos adaptado de Algoritmo Genético- Rio de Janeiro : Brasport .....                    | 40 |
| Figura 9 Processo de Mutação fonte Algoritmo Genético- Rio de Janeiro : Brasport .....  | 42 |
| Figura 10 Placa de Aquisição de dados NI-DAQ™mx USB-6009 da National Instruments ....   | 46 |
| Figura 11 diagrama de blocos do sistema de aquisição de dados.....  | 47 |
| Figura 12 sistema de aquisição de dados utilizando placa NI-DAQ™mx USB 6009 .....   | 47 |
| Figura 13 Sinal adquirido pelo sistema de aquisição NI-DAQ™mx USB. ....   | 48 |
| Figura 14 Sinal Real obtido no período de 18 ms a uma taxa de 1,25 ms .....   | 49 |
| Figura 15 Representação dos pais e filhos no processo de média simples.....   | 54 |
| Figura 16 sinal adquirido e sinal estimado para uma população de 10 indivíduos e 10 iterações. ....                               | 57 |
| Figura 17 Sinal real e sinal estimado para 100 iterações .....  | 58 |
| Figura 18 Sinal real e sinal estimado após 1000 iterações .....   | 59 |
| Figura 19 Resultados para 100 iterações e janela de ½ ciclo.....  | 60 |
| Figura 20 Resultado para janela de 1 ciclo.....   | 61 |
| Figura 21 Resultado sinal real e sinal estimado para população de 30 indivíduos após 10 iterações.....                            | 63 |
| Figura 22 Resultados sinal real e sinal estimado para população de 30 indivíduos 100 iterações .....                              | 64 |
| Figura 23 Resultado Sinal real e sinal estimado para uma população de 30 indivíduos e 1000 iterações.....                         | 65 |
| Figura 24 Resultados para uma população de 50 indivíduos 1000 iterações.....  | 66 |
| Figura 25 Resultados da FFT para 50 amostras do sinal.....  | 68 |
| Figura 26 FFT com 100 amostras do sinal de entrada .....  | 69 |
| Figura 27 FFT com 1000 amostra do sinal de entrada. ....  | 70 |
| Figura 28 sinal y(t) gerado pela equação de Mackey-Glass.....   | 71 |
| Figura 29 Sinal de entrada e sinal estimado obtido pelo AG após 100 iterações.....  | 72 |
| Figura 30 Sinal obtido após 500 iterações.....  | 73 |
| Figura 31 Resultados para execução do AG após 1000 iterações.....   | 74 |
| Figura 32 Sinal de controle para um regulador de tensão de um gerador durante processo de “carregamento” do sistema elétrico..... | 75 |
| Figura 33 Regulador de tensão de gerador K38 AVR.....   | 76 |
| Figura 34 Resultado do GA para estimação do sinal de controle de regulador de tensão.....   | 77 |
| Figura 35 Degrau.....   | 78 |

## Lista das Tabelas

|   |    |
|---|----|
| Tabela 1 Correspondências entre Genética Biológica e Algoritmo Genético.....  | 27 |
| Tabela 2 Distribuição de indivíduos no disco da roleta .....  | 36 |
| Tabela 3 indivíduos da população inicial distribuídos equidistantemente no espaço de busca. .                                     | 51 |
| Tabela 4 População de 10 Indivíduos avaliados com a respectiva aptidão, janela ¼ de ciclo....                                     | 55 |
| Tabela 5 Resultado da primeira população gerada após a aplicação dos operadores genéticos:<br>seleção, cruzamento e mutação. .... | 56 |
| Tabela 6 Resultados da execução do AG para o caso 1 : 10 indivíduos e 10 iterações .....  | 57 |
| Tabela 7 Resultados encontrados após a execução do AG. por 100 iterações .....  | 58 |
| Tabela 8 Resultados obtidos após a execução do AG por 1000 iterações.....   | 59 |
| Tabela 9 Resultados obtidos após a execução do AG para janela de ½ ciclo, 10 indivíduos e<br>100 iterações.....                   | 60 |
| Tabela 10 Resultados obtidos após a execução do AG. para uma janela de 1 ciclo 10 indivíduos<br>e 100 iterações.....              | 61 |
| Tabela 11 Resultados obtidos após a execução do AG. para uma população de 30 indivíduos por<br>10 iterações.....                  | 62 |
| Tabela 12 Resultados obtidos após a execução do AG. para uma população de 30 indivíduos por<br>100 iterações.....                 | 63 |
| Tabela 13 Resultados obtidos após a execução do AG. para uma população de 30 indivíduos por<br>1000 iterações.....                | 64 |
| Tabela 14 Resultados obtidos após a execução do AG. para uma população de 50 indivíduos por<br>1000 iterações.....                | 65 |
| Tabela 15 Resultados obtidos após a execução do AG. por 100 iterações.....  | 72 |
| Tabela 16 Resultados obtidos após a execução do AG. por 500 iterações.....  | 73 |
| Tabela 17 Resultados obtidos após a execução do AG para 1000 iterações. ....  | 74 |

## Lista das abreviaturas

A - Amplitude

AG – Algoritmo Genético

f - Frequencia

FFT – Fast Fourier Transform

FPGA – Field Programmable Gate Array

pu – por unidade.

SEP – Sistema Elétrico de Potência

TDF – Transformada Discreta de Fourier

TF – Transformada de Fourier

ONS - Operador Nacional do Sistema

ANEEL – Agencia Nacional de Energia Elétrica

## Lista de Símbolos

$x(t)$  é o sinal da rede;

$y(t)$  é o sinal calculado;

$e$  é o erro;

$A$  é a amplitude da senoide;

$f$  é a frequência do sinal ;

$\theta$  é ângulo de fase;

$\pi$  é a constante pi = 3,1415 rad

$\sin$  é a função trigonométrica seno

$A_e$  é a amplitude estimada

$f_e$  é a frequência estimada

$\theta_e$  é ângulo de fase estimado

$f_{mn}$  é filho menor

$f_{md}$  é filho médio

$f_{ma}$  é filho maior

$\alpha$  é a variável alfa

$\beta$  é a variável beta

$\tau$  é a variável tau

## Sumario

|  |    |
|--|----|
| 1 – Introdução.....  | 14 |
| 1.1 - Motivação .....  | 19 |
| 1.2 - Justificativa .....  | 20 |
| 1.3 - Organização .....  | 20 |
| 2 – Revisão Bibliográfica.....                                   | 21 |
| 3 - Algoritmos Genéticos .....                                   | 25 |
| 3.1 - Funcionamento do Algoritmo Genético.....                   | 28 |
| 3.1.1 - Representação Cromossomial .....                         | 31 |
| 3.3 –Avaliação.....  | 34 |
| 3.4 – Seleção .....  | 35 |
| 3.5 – Cruzamento .....   | 38 |
| 3.5.1 -Cruzamento de um ponto .....                              | 39 |
| 3.5.2 Cruzamento de dois pontos .....                            | 40 |
| 3.5.3 – Cruzamento pela média aritmética.....                    | 40 |
| 3.5.4 – Cruzamento pela media geométrica .....                   | 41 |
| 3.6 – Mutação.....   | 41 |
| 3.7-Formação da nova população.....                              | 42 |
| 3.8 – Elitismo .....   | 43 |
| 3.9 - Critério de parada.....                                    | 43 |
| 3.10 - Paralelismo implícito .....                               | 44 |
| 4 –Resultados.....   | 46 |
| 4.1 Aplicação do Algoritmo Genético a sinais periódicos .....    | 50 |
| 4.1.1 Aplicação de AG com populações de indivíduos pequenas..... | 51 |
| 4.1.2 Aplicação de AG com população de 30 indivíduos .....       | 62 |
| 4.1.3 Aplicação de AG com população de 50 indivíduos .....       | 65 |
| 4.2 Aplicação da Transformada de Fourier .....                   | 68 |
| 4.3 Sinais aperiódicos e caóticos.....                           | 70 |
| 4.4 Considerações Finais.....                                    | 79 |
| 5-CONCLUSÃO.....   | 80 |
| 5.1 Sugestão de Trabalhos Futuros .....                          | 81 |
| Referencias Bibliográficas .....                                 | 82 |
| Anexo A.....   | 86 |
| Anexo B .....  | 93 |

## 1 – Introdução

O homem é um animal “racional”, mas o que o qualifica como racional e os outros como irracionais?

Os primórdios da civilização humana foram anos da chamada luta pela sobrevivência, sempre havia perigos, necessidades e limitações, no entanto o homem superou através sua capacidade de se adaptar, que lhe permitiu o domínio sobre todas as criaturas e controle de situações adversas.

O uso força permitia a seleção da espécie, os mais fortes e mais preparados para aquele ambiente se reproduziam e transmitiam seus genes para as gerações futuras, até que o homem começou a usar o cérebro ou a inteligência e descobriu o poder das ferramentas: uma alavanca e um apoio permitia aumentar sua força em muitas vezes e trabalhos de difíceis execuções se tornavam fáceis com o uso da ferramenta certa, então a reprodução passou a ser privilégio dos mais adaptados que transmitiam seus genes para as gerações futuras. LIDEN, R. (2006)

A evolução da espécie foi assunto de estudo do cientista inglês Charles Darwin (1809 - 1882) que em 1859 escreveu o livro: “A Origem das espécies” no qual apresenta a teoria da evolução das espécies que diz que os indivíduos de um ecossistema competem entre si por recursos limitados, tais como, água e comida. LIDEN, R (2006).

Aqueles dentre os indivíduos de uma mesma espécie que não obtêm êxito tendem a ter uma prole menor.

Na teoria da Evolução, os indivíduos que melhor se adaptam ao ambiente são os que serão selecionados para reproduzir e gerar uma nova população. Então há a necessidade de uma avaliação da população atual para se determinar quem melhor se adaptou ao ambiente, esta avaliação é feita através de uma função de pertinência chamada de “fitness”, que é pertinente ao problema a ser solucionado. Esta ligação permite que a população inicial possa ser qualquer indivíduo, com quaisquer características dentro do espaço de busca. Por mais distante que esteja um indivíduo da solução ele terá uma nota em função da adaptabilidade ao ambiente que esta dentro do espaço de busca. Em função das melhores notas é feita a seleção dos indivíduos que participarão do cruzamento (reprodução) para geração da nova população.

Dois indivíduos selecionados (pais) são combinados (reproduzem) gerando novos indivíduos (filhos) que poderá ou não sofrer mutação segundo uma probabilidade numa tentativa de se evitar a estagnação antecipada da solução em um mínimo local.

Os indivíduos gerados (filhos) possuem as características de adaptabilidade dos pais e ainda por terem passado pelo processo de mutação, que evitam os mínimos locais, têm mais chances de obter sucesso na convergência para a solução ótima.

Um ponto chave na seleção dos indivíduos para o cruzamento é o elitismo, que consiste na seleção automática do melhor indivíduo (o de maior fitness) da população atual para fazer parte da nova população, este procedimento permite que a solução não se perca tornando a busca “cega” o que dificultaria a convergência.

A evolução da resposta é um processo de otimização que trará em última instância a melhor solução para o problema, que é determinada pela convergência do algoritmo e sua parada, que pode acontecer por vários métodos, por exemplo, número de iterações.

Os Algoritmos Genéticos (AGs) são algoritmos de busca e otimização que utilizam processos aleatórios de busca guiados por decisões probabilísticas para a obtenção de soluções ótimas ou aproximadamente ótimas, GOLDBERG (1989).

O uso de Algoritmos Genéticos para a estimação de parâmetros de um sinal periódico pode ser entendido como uma busca, num espaço limitado de multiparâmetros de uma senoide com amplitude, frequência e fase.

EL-NAGGAR & YOUSSEF (2000) propuseram um método rápido e eficiente baseado em AGs para a estimação dos parâmetros da rede para implementação de relé de frequência para a proteção do SEP (Sistema Elétrico de Potência).

Este problema real da variação da frequência da rede, constantemente monitorada por reles de frequência que usam técnicas de análise através da clássica Transformada de Fourier TF, e suas derivadas é usado neste estudo como base de aplicação da ferramenta Algoritmo Genético para sinais periódicos, devido a sua natureza um breve esclarecimento se faz necessário para o entendimento do problema.

A rede de energia elétrica que tem como base o sinal senoidal na frequência fundamental 60 Hz e é muito sensível a variações devido a alterações constantes modificações na carga distribuída e entradas/saídas de geradores uma vez que o sistema é interligado o problema afeta toda a rede.

O sistema de energia é composto pelo conjunto de geração (hidroelétricas, termoelétricas, nucleares, solares e eólicas, principalmente) de propriedade pública ou privada, espalhadas por todo o território nacional, pelo sistema de transmissão operado



pelo governo através da ONS (Operador Nacional do Sistema), e do sistema de distribuição de energia, que foi privatizado e pulverizado entre várias companhias de distribuição de energia, dentre as quais podemos citar: EDP Bandeirantes, Elektro, etc. Essa pluralidade de empresas na geração e distribuição fez com que o governo criasse um órgão regulador para intervir buscando a unidade do sistema e regulamentação das operações este órgão é ANAEL (Agencia Nacional de Energia Elétrica).

Em função das exigências dos usuários em termos de demanda e qualidade de energia, padrões de confiabilidade e continuidade foram estabelecidos e são constantemente cobrados das empresas de geração e distribuição de energia, pois a vida das empresas usuários depende em muito da energia que lhes é fornecida.

Os problemas que acometem o sistema de energia são inúmeros, pois cada terminação pode interagir de forma negativa com o sistema, assim um curto circuito nada mais é de que uma alta solicitação de corrente e provocará uma forte variação no sistema, que tentará prover através de solicitação de entrada de novos geradores no sistema que após a detecção de falha serão retirados do sistema.

Essas variações bruscas no sistema provocam uma deterioração na qualidade de energia, pois as máquinas elétricas são fortemente dependentes da frequência do sistema para gerar os campos magnéticos e suas forças magneto motrizes que provocarão o movimento dos motores, os quais devido a suas características de hardware (construtivas) e princípio de funcionamento atrelam a velocidade de operação ao parâmetro frequência da rede, que é fortemente influenciado pela entrada e saída da carga e pela entrada e saída dos geradores, a entrada de carga que solicita alta corrente provoca uma diminuição na frequência do sistema e a entrada de um gerador no sistema provoca um aumento na frequência, portanto é necessário um equilíbrio entre a energia gerada e a energia consumida.

O controle da carga cabe à necessidade do usuário, assim o sistema tem como ponto de compensação a entrada e saída dos geradores e o desacoplamento de cargas, quando estas são prejudiciais ao sistema, após todas as providências de tentativa de estabelecimento de equilíbrio dos operadores do sistema.

Há que ressaltar ainda, que o órgão regulador preza por uma energia com qualidade com respostas rápidas a variações bruscas e intempestivas provocadas pelos usuários ou pelo próprio sistema.

A qualidade em energia elétrica esta correlacionada com a estabilidade do sistema, no caso aos parâmetros que definem o sistema, tensão senoidal com amplitude e frequência

constantes (caso ideal), sendo aceitável, pequenas tolerâncias, tomando a amplitude igual a 1 pu, aceitando-se que amplitude varie 10% temos a faixa de 0,9 pu a 1,1 pu.

A frequência da rede no Brasil fixada em 60 Hz, a tolerância de desvio de frequência é de +- 0,5 Hz assim, frequência abaixo de 59,5 Hz não é tolerada pelo sistema e uma atitude deve ser tomada para restituição do equilíbrio do sistema bem como frequência acima de 60,5 Hz não é aceita pelo sistema.

Para frequência entre 59,5 Hz e 60,5 Hz o sistema consegue compensar fazendo uso dos recursos de controle dos geradores atuando na velocidade do rotor, sabe-se que a frequência chega a variar 2 Hz, portanto a faixa de variação fica de 58 Hz a 62 Hz.

Outras inúmeras interferências acometem a qualidade de energia, como exemplo: os harmônicos, os transitórios de chaveamento, o fator de potência das cargas, etc. Todas estas anomalias prejudicam o desempenho dos equipamentos ligados à rede elétrica, mas a anomalia que sem dúvida tem maior efeito sobre os desempenho dos equipamentos é a frequência, pois nos motores as velocidades tem alta dependência da frequência da rede.

Se a frequência da rede define a velocidade dos motores e a frequência se altera no decorrer do período, a velocidade vai se alterar e inúmeras aplicações que dependem da velocidade ou de uma determinada velocidade serão prejudicadas, o que pode inclusive gerar prejuízos aos fabricantes, assim, quanto mais a frequência se aproximar da frequência nominal 60 Hz e a amplitude se aproximar de 1,0 pu melhor.

Considerando o sinal da rede como uma senoide, sua representação por uma expressão matemática, é descrita na equação (1).

$$x(t) = A \sin(2\pi ft + \theta) \quad (1)$$

Aplicando as funções matemáticas e variando o tempo na equação (1) é possível gerar a forma de onda do sinal da rede. Tomando como frequência  $f = 60$  Hz, a amplitude  $A = 1$  pu e o Ângulo de fase  $\theta = 0$

Utilizando MatLab®, que é o mesmo software usado para processar os algoritmos genéticos que são utilizados nas aplicações desta dissertação, resulta na forma de onda mostrada na figura (1)

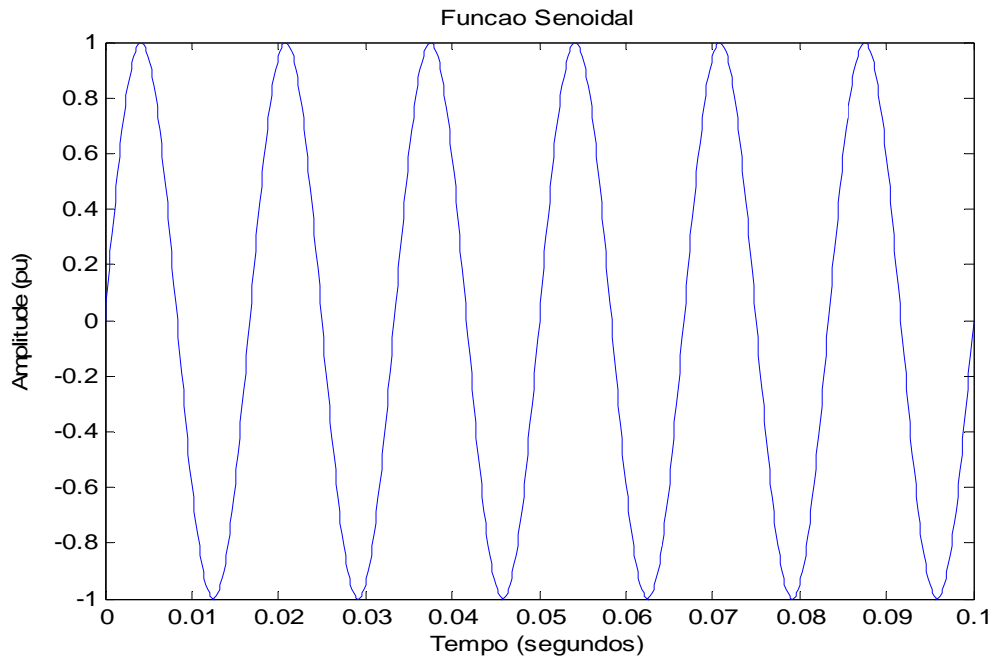


Figura 1 Onda senoidal da rede elétrica de 60 Hz

Na validação da ferramenta AG um sinal de parâmetros não conhecidos como o mostrado na figura (2) é adquirido através de um sistema de NI-DAQmx® da National Instruments™, e o arquivo de dados resultante permitirá a reprodução do sinal no MatLab® diretamente, ressaltando que os parâmetros deste sinal mostrado nesta figura(2) se aproximam dos parâmetros base mostrados na figura(1), mas não são conhecidos, serão estimados pela ferramenta e comparados com a estimativa pelos métodos tradicionais que utilizam a transformada de Fourier.

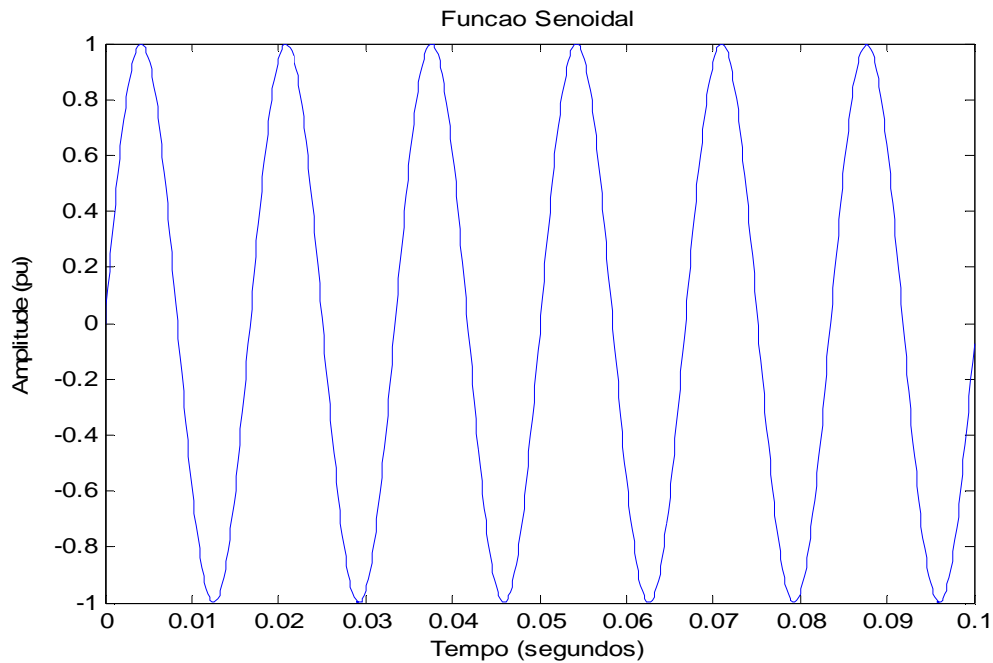


Figura 2 senoide de frequência ignorada próxima de 60 Hz, fase próxima 0 rd e amplitude próxima 1 pu.

Este problema de estimação de parâmetros de um sinal no qual a ferramenta é aplicada permitirá verificar sua atuação, eficiência, robustez e para efeito de avaliação a comparação com os métodos atuais usados na engenharia: TRANSFORMADA DE FOURIER, TF, no caso de sinais amostrados, a DFT (Discrete Fourier Transform) ou a FFT (Fast Fourier Transform) .

A estimação dos parâmetros do sinal de entrada passa por um tratamento de problema com multivariáveis, ou seja, determinar a amplitude, frequência e fase de um “sinal periódico” o que possibilita a aplicação da FFT, porque se não for periódico, falta ferramentas!

## 1.1 - Motivação

Existe a necessidade de uma ferramenta computacional que resolva problemas multivariáveis e realize estimativas de parâmetros de formação de sinais.

A proposta deste trabalho é apresentar o Algoritmo Genético como essa ferramenta que estima parâmetros e que possibilite reproduzir o sinal quer para estudo, quer para simulações, desenvolvimento de produtos ou outras necessidades da engenharia.

O desenvolvimento dessa técnica e sua aplicação em problemas reais como o problema dos relés de frequência, relatado, abre um novo caminho para indústria na criação de novos produtos que utilizem a técnica, que possam atualizar seus produtos para métodos e técnicas mais atuais, robustas e eficientes.

## **1.2 - Justificativa**

O trabalho proposto é uma inovação no método de análise de sinais hoje dependente de métodos antigos e limitados, que na maioria das vezes apresentam restrições ou baixa imunidade às condições de funcionamento.

A nova proposta vem a preencher um espaço vazio na análise dos sinais aperiódicos os quais não são tratados pelas ferramentas clássicas como Transformada de Fourier.

## **1.3 - Organização**

O trabalho está assim organizado:

Capítulo I - Introdução

Capítulo II - Revisão Bibliográficas

Capítulo III - Algoritmo Genético

Capítulo IV - Resultados

Capítulo V - Conclusão

Referencias Bibliográficas

Anexos

Certamente este estudo será de grande utilidade para professores e alunos no estudo dos sinais, e a muitos profissionais no desenvolvimento de novos produtos através de simulações mais próximas dos sinais reais.

## 2 – Revisão Bibliográfica

Este capítulo fará um apanhado de diversas aplicações dos AGs possibilitando a compreensão de quão poderosa é a ferramenta AG, de sua flexibilidade e robustez.

A estimação de parâmetros de um sinal é uma tarefa constante na engenharia e a precisão dessa estimativa qualifica a ferramenta, assim uma análise de aplicações das ferramentas de estimação permite a completa noção da abrangência do AG para os estudos dos sinais.

VARGAS (2005) apresentou um estudo baseado em AG, que realiza a estimação dos parâmetros associados a forma de onda, tais como amplitude, frequência e ângulo de fase, referente a uma proposta de relé de frequência. Neste estudo foi explorado a variação do número de indivíduos da população utilizando 6, 12, 24, 30 e 48 indivíduos, também foi experimentado os métodos de seleção por roleta e torneio, além da variação do critério de parada em 1000, 5000, 7500 e 10000 iterações para frequências de amostragem 400, 800, 1200 e 2400 Hz.

SOUZA (2008) apresentou o estudo: Algoritmos Genéticos aplicados a Proteção e Estimação de Harmônicos em Sistemas Elétricos de Potência, no qual discute a utilização da ferramenta GA na estimação de harmônicos em sistemas elétricos de potência, na estimação dos parâmetros para um relé de frequência e para um relé de distância, sendo que para o relé de frequência foi implementado um FPGA ( Field-Programmable Gate Array) com AG que possibilitou a atuação em tempo real. Em seu estudo SOUZA(2008) compara o desempenho dos AGs e dos métodos tradicionais que utilizam a FFT, nos quais é possível verificar a alta performance do AG em relação aos métodos tradicionais.

A presença de harmônicos no sinal de rede é altamente prejudicial à qualidade do sinal, segundo DUGAN et al. (1996), estimação correta das distorções harmônicas é um processo de grande importância para o correto funcionamento do sistema de energia elétrica, para tanto enumera os motivos que levam a falhas de operação, controle e proteção do sistema:

- a) Os equipamentos são sensíveis às variações nas formas de energia fornecidas. Muitos destes equipamentos possuem controles baseados em microprocessadores e dispositivos eletrônicos sensíveis a muitos tipos de distúrbios;

- b) O interesse pela racionalização e conservação da energia elétrica, visando otimizar a utilização da energia, tem aumentado o uso de equipamentos que, em muitos casos, aumentam os níveis de distorções harmônicas e podem levar o sistema a condições de ressonância;
- c) A crescente integração dos processos, significando que a falha de qualquer componente traz consequências ainda mais importantes para o sistema elétrico;
- d) As consequências resultantes de variações nas formas de onda sobre a vida útil dos componentes elétricos.

A análise das componentes harmônicas pode ser baseada nas teorias de estimação estática e dinâmica. Como exemplo de estimação estática temos os métodos baseados nos Mínimos Quadrados (MQ), a Transformada Discreta de Fourier (TDF) e a Transformada Rápida de Fourier (FFT), já os métodos de estimação dinâmica utilizam o Filtro de Kalman. BURRUS&PARKS (1985)

Segundo SOLIMAN (1997) o filtro de Kalman é um estimador ótimo recursivo e é também componentes harmônicas em SEP. Por ser baseado em estimativa dinâmica possui a capacidade de identificar, analisar e localizar harmônicos contidos em um sinal trifásico não estacionário.

O método dos MQ é uma técnica de otimização matemática que procura encontrar o melhor ajuste para um conjunto de dados tentando minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados.

A técnica dos MQ é baseada na minimização da soma do quadrado da diferença entre o valor estimado e o valor medido por uma função KAMWA & GRONDIN (1992) provam que a técnica dos MQ oferece uma estimativa razoável dos fasores.

A TDF é baseada na teoria que o sinal é composto de uma componente fundamental acrescida de uma infinita soma de harmônicos.

A TDF e a FFT são fundamentadas na teoria de transformadas ortogonais, onde a forma de onda medida consiste de uma componente fundamental acrescida de uma infinita soma de harmônicos, COOLEY, J.W. (1992).

A técnica TDF pode conduzir a erros no processo de medição, pois sofre influência dos ruídos randômicos, GIRGIS at al, (1991), e o desempenho do algoritmo pode ser afetado pelo componente CC do sinal.

A evolução da TDF permitiu a obtenção da FFT (FAST FOURIER TRANSFORM), a técnica que é comumente mais usada pela facilidade de cálculo.

A FFT foi implementada com o objetivo de diminuir a complexidade (temporal) necessária para calcular uma DFT (Transformada Discreta de Fourier), visando aplicações em tempo real.

O número de operações realizadas no cálculo da DFT através da definição é proporcional à  $N^2$ , isto é, para cada um dos  $N$  valores de  $u$ , a expansão de  $F(u)$  requer  $N$  multiplicações complexas de  $x(n)$  por  $W_N^{ux}$  além de  $(N-1)$  adições dos resultados. Alguns dos termos podem ser computados uma vez e armazenados para serem usados em operações futuras. Logo tais multiplicações de  $x(n)$  não são consideradas na implementação.

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cdot W_N^{ux} \quad (2)$$

Algumas decomposições apropriadas na equação (2) podem tornar o número de multiplicações e adições proporcionais a  $N \cdot \log_2 N$ .

Este procedimento é denominado de Transformada Rápida de Fourier ou FFT (Fast Fourier Transform).

O algoritmo da FFT é aplicável apenas quando os sinais são periódicos. CHANG et al.(2000), na tentativa de estimação dinâmica de harmônicos, propuseram o truncamento da FFT para um período. Respostas rápidas e exatas foram obtidas por um método de truncamento com superposição.

COURY et al (2011) apresentou o estudo “Algoritmo Evolutivo para a Estimação de Frequência em Sistemas Elétricos utilizando FPGAs”, no qual aborda a técnica que considera eficiente para a estimação da frequência do sinal em sistemas elétricos de Potência baseada em Algoritmos Genéticos –AGs. Neste estudo a estimação da frequência do sinal elétrico é modelada como um problema de otimização, característica intrínseca do Algoritmo Genético. Segundo COURY, a motivação para o uso de AGs nesta aplicação é sua intrínseca robustez ao ruído presente nos sinais, além da grande vantagem de utilização da estrutura paralela do FPGA em combinação com a estrutura paralela do AG que possibilitam uma redução considerável no tempo de iteração permitindo a estimação de frequência em tempo real, e sua utilização em equipamentos de proteção como relé de frequência, relé de distância e outras proteção que exijam atuações em tempo real.

EL NAGGAR & YOUSSEF (2000) utilizaram AG como ferramenta principal para a resolução de problemas de estimação de frequência, assim como SOUZA (2008) e COURY et al (2009).



MACEDO et al (2004) em seu estudo: “Uma rotina baseada em algoritmo genético para identificação de harmônicos em chaveamentos de bancos de capacitores”, aplicou AG para estimativa dos componentes de frequência presentes em uma forma de onda ruidosa provenientes de sistema elétrico de potência e comparou os resultados da aplicação de AG com os resultados obtidos da forma clássica pela transformada de Fourier (TF). Chegando a conclusão que os AGs são mais eficientes e rápidos na estimação que os métodos tradicionais.

SOUZA et al (2007) em seu estudo: “Caracterização de Componentes Harmônicas em Sistemas Elétricos via Algoritmos Genéticos”, mostra a eficiência do AGs para determinação das componentes harmônicas comuns nas duas situações analisadas: chaveamento de bancos de capacitores e energização de um transformador de potência relevando a representação real sobre a representação binária em termos de precisão das resposta mesmo em comparação com o método tradicional de Transformada de Fourier.

Souza et al, (2006) realizaram um estudo: “ Estimação de parâmetros de um sistema elétrico de potência utilizando algoritmos genéticos”, no qual mostraram que os AGs conseguem estimar as componentes harmônicas presentes em sinal elétrico de potência e também conseguem em uma segunda aplicação estimar os parâmetros Amplitude, Frequência e ângulo de fase para a atuação de um relé de frequência, neste estudo concluíram que os AGs são uma eficiente ferramenta computacional para a estimação de parâmetros de sinais periódicos.

Macêdo et al, (2004) em seu estudo : “Algoritmos evolucionários Aplicados na Estimação de Parâmetros de um Sistema Elétrico de Potência”, em que a utilização de AGs permitiu a estimação dos Parâmetros de formas de ondas de sinais de tensão e corrente, sendo ressaltada a utilização do programa GOOAL – Genetic Object Oriented Algorithm, em combinação com os dados do MatLab® e da CHESF – Companhia Hidroelétricas do São Francisco. A conclusão deste estudo relata a eficiência da ferramenta computacional na estimação destes parâmetros.

Silva, R.P.M. (2012) apresentou um trabalho, “Algoritmos Genéticos Aplicados à Medição Fasorial e Projetos de PMUs”, no qual ressalta a eficiência dos AGs na estimação fasorial e rastreamento de frequência da componente fundamental em sinais de Sistemas Elétricos de Potência (SEPs). O problema de estimação é modelado como um problema de otimização para a utilização de Algoritmos Genéticos (AGs) dedicados a busca da solução ótima. Como resultado foi desenvolvido um medidor Fasorial em FPGA combinando a técnica de Algoritmo Genético com a estrutura paralela do FPGA.

### 3 - Algoritmos Genéticos

A natureza é sabia, e um dos processos da natureza mais bem elaborado é o processo evolutivo estudado por Darwin no século XIX. Baseado em suas observações escreveu a teoria da evolução das espécies a qual preconiza que os mais capazes e que tem maior adaptabilidade ao meio são os que sobrevivem e evoluem.

O homem é um animal observador e da observação das características e princípios dos processos existentes na natureza retira inspiração para a criação de máquinas, métodos e técnicas que contribuam a realização ou facilitação do trabalho. Alguns aviões foram baseados nas características dos pássaros, os sonares foram inspirados nos morcegos.

A Inteligência Artificial também se inspira na natureza para seu desenvolvimento, a própria criação da técnica inteligente de redes neurais inspirada no cérebro humano, em neurônio.

Algoritmo Genético se baseia na teoria de Charles Darwin (1809 - 1982) que em 1859 escreveu o livro “A Origem das espécies”; onde apresenta a teoria da evolução das espécies na qual os indivíduos de um ecossistema competem entre si por recursos limitados tais como água e comida, LIDEN, R (2006). Aqueles dentre os indivíduos de uma mesma espécie que não obtém êxito tendem a ter uma prole menor, já os mais adaptados ao ambiente se reproduzem suas características genéticas para as novas populações que sofrem mutações e se adaptam mais de forma evolutiva.

Aliando a tudo isso a computação, que é uma ciência que visa a solução de problemas e que ultimamente vem se inspirando de técnicas da natureza com a finalidade de gerar soluções para os problemas práticos como exemplo disso, a criação de mais uma técnica de inteligência computacional, denominada Algoritmo Genético, uma das subáreas da Computação Evolutiva.

Os Algoritmos Genéticos são uma técnica heurística que baseada no processo de evolução e vem se mostrando uma ferramenta para soluções de grande parte dos problemas de multivariáveis.

Os algoritmos genéticos também tem grande sucesso na estimação, pois grande parte dos problemas de estimação são problemas de otimização, isto é, são problemas que a solução converge para um valor.

Algoritmos Genéticos são algoritmos de buscas adaptativas para otimização de problemas, GOLDBERG (1989)

Os Algoritmos Genéticos são procedimentos adaptativos, para a busca de soluções em espaços complexos, fundamentados na evolução biológica, com padrões de operações sobrevivência dos indivíduos que melhor se adaptam ao ambiente.

Os algoritmos genéticos sempre começam pela definição de uma população inicial, esta contém parâmetros de suma importância para a solução do problema, como a definição do número de indivíduos da população e suas características representativas das incógnitas do problema a ser solucionado.

O indivíduo elemento da população possui a representatividade das incógnitas, isto é, se a solução consiste em uma variável, a população é representada por esta variável, se é um problema de multivariáveis os indivíduos possuem partes que representam estas variáveis.

Os indivíduos podem ter sua construção de forma binária ou real, depende da representatividade das variáveis, isto é alguns problemas podem ser representados com descontinuidades e uma representação binária pode trazer vantagens, outros podem ser prejudicados por esta descontinuidade e uma representação real pode ser mais vantajosa. As definições iniciais são importantes qualitativamente, pois a representação da solução está ligada a qualidade do indivíduo, quantitativamente a solução é tanto melhor quanto mais precisa for.

Os algoritmos genéticos sempre começam com essa representatividade, pois eles não trabalham com os problemas, mas com sua representatividade através de uma codificação na construção do indivíduo.

A análise inicial do problema permite a definição do espaço de busca, espaço onde se encontra a solução e que é pesquisado pelo algoritmo genético. O espaço de busca pode ter qualquer amplitude, uma garantia necessária ao funcionamento dos AGs é que a solução faça parte do espaço de busca, isto é, esteja dentro das fronteiras do espaço de busca.

A distribuição da população dentro do espaço de busca pode ser randômica, ou uma distribuição equidistante que em alguns casos permite uma convergência mais rápida, isto é, algumas iterações são poupadas, mas em ambos os casos haverá a convergência, desde que a solução faça parte do espaço de busca.

Existe uma terminologia empregada nos AG que aproxima muito da genética natural ou biológica, algumas considerações são necessárias para o entendimento dos textos sobre os AG:

- Cromossomos e indivíduos são termos intercambiáveis que definem um elemento da população. Devido a um grande numero de AGs usarem a representação binária (bits) os cromossomos podem ser referidos muitas vezes como “strings”.
- População ou geração é o conjunto de elementos (cromossomos ou indivíduos) alocados para a solução de AG.
- População Inicial é o conjunto de elementos da primeira geração, quando o problema começa a ser resolvido.
- Como na genética biológica os cromossomos são formados por genes nos AGs que são as características do indivíduo ou cromossomo.
- Os genes podem ter um valor entre vários possíveis tanto na genética quanto nos AGs, o que é chamado de “alelos”, e a posição que um gene ocupa na genética é chamada de “locus” assim como nos AGs.
- Genótipo é a estrutura do cromossomo e fenótipo é a interação entre o conteúdo genético e o ambiente, interação que em AGs ocorre entre o conjunto de parâmetros do algoritmo.

Assim é possível construir uma tabela, em que são apresentadas as semelhanças entre a Genética Biológica e a os Algoritmos Genéticos, como a seguir é mostrado na tabela 1.

Tabela 1 Correspondências entre Genética Biológica e Algoritmo Genético

| Genética Biológica | Algoritmo Genético            |
|--------------------|-------------------------------|
| Cromossomo         | Indivíduo, string, cromossomo |
| Gene               | Característica                |
| Alelo              | Valor                         |
| Locus              | Posição                       |
| Genótipo           | Estrutura                     |
| Fenótipo           | Conjunto de parâmetros        |

Fonte Algoritmo Genético- Rio de Janeiro : LIDEN, R. 2006

Essa correspondência entre a Genética Biológica e os Algoritmos Genéticos nos permite analisar seu funcionamento comparativamente e ainda avaliar seu desempenho com relação a tempo de convergência e precisão da solução, pois temos os conceitos da genética biológica fundamentados e conhecemos os seus processos.

### 3.1 - Funcionamento do Algoritmo Genético

O funcionamento dos AGs pode ser resumido algoritmicamente através dos seguintes passos:

- a) Inicialize a população de cromossomos

Neste passo criamos uma população com características pertinentes ao problema que desejamos resolver, e definimos o espaço de busca onde a solução será procurada.

- b) Avalie cada cromossomo na população

A população criada deve ser avaliada através de uma fitness ou função custo, que premie os indivíduos melhor adaptados e puna os menos adaptados ao problema.

- c) Selecione os pais para gerar novos cromossomos.

Através de processos de seleção que podem ser roleta, torneio e outros, dois indivíduos são selecionados para o cruzamento.

- d) Aplique os operadores de recombinação a estes pais de forma a gerar os indivíduos da nova geração aplique o operador de mutação aos novos indivíduos.

Os pais selecionados no item d, são cruzados , de forma que suas características (genes) passem para seus filhos que serão gerados, e sofrerão alterações propositais para terem diversidades.

- e) Apague os velhos membros da população

Descartando os elementos antigos, uma nova população , com o mesmo numero de elementos é constituída.

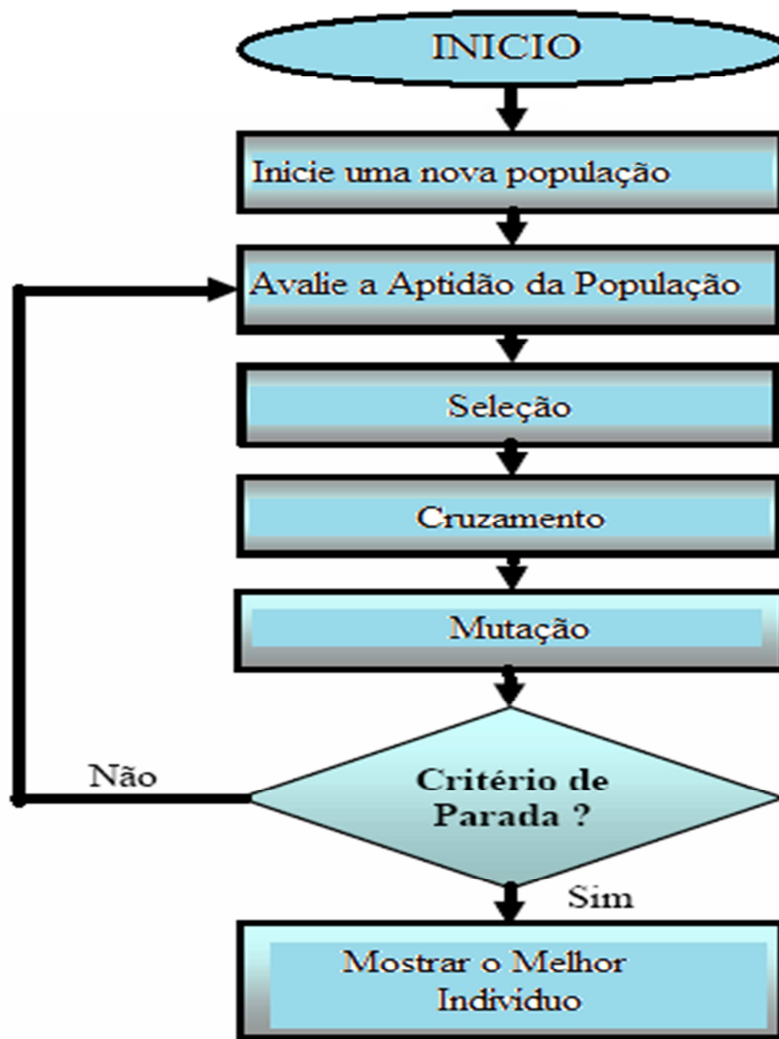
- f) Avalie todos os novos cromossomos e insira-os na população

Esta nova população é avaliada e notas são atribuídas a cada indivíduos segundo a sua adaptabilidade ao problema

- g) Se o tempo acabou, ou o melhor cromossomo satisfaz os requisitos de desempenho, retorne-o, caso contrário volte para o passo c).

O Critério de parada que pode ser o número de iterações, ou uma tolerância em torno da resposta esperada

Em forma de fluxograma teríamos a seguinte figura:



**Figura 3 Fluxograma do funcionamento do Algoritmo Genético**

A estrutura básica de funcionamento mostrada na figura 4, revela um nível maior de detalhes o funcionamento do AGs. Inicialmente colocamos uma população para ser avaliada segundo uma função de pertinência chamada de fitness que qualifica cada indivíduo da população segundo sua adaptabilidade como solução do problema. O próximo passo é a seleção dos indivíduos segundo um dos critérios de seleção de forma que os mais aptos sejam selecionados para o cruzamento, assim os pais selecionados geraram filhos que geneticamente serão mais adaptados, estes filhos podem ou não sofrer mutações, que possibilitarão a aquisição de novas e diferentes características.

Esses filhos possibilitarão a formação da nova geração de indivíduos ou nova população e será verificado se estes indivíduos atendem a solução do problema ou devem ser submetidos ao processo novamente.

O princípio fundamental de um AG é que o melhor indivíduo ou membro de uma população possui a maior probabilidade de sobrevivência, GEN & CHENG (1996).

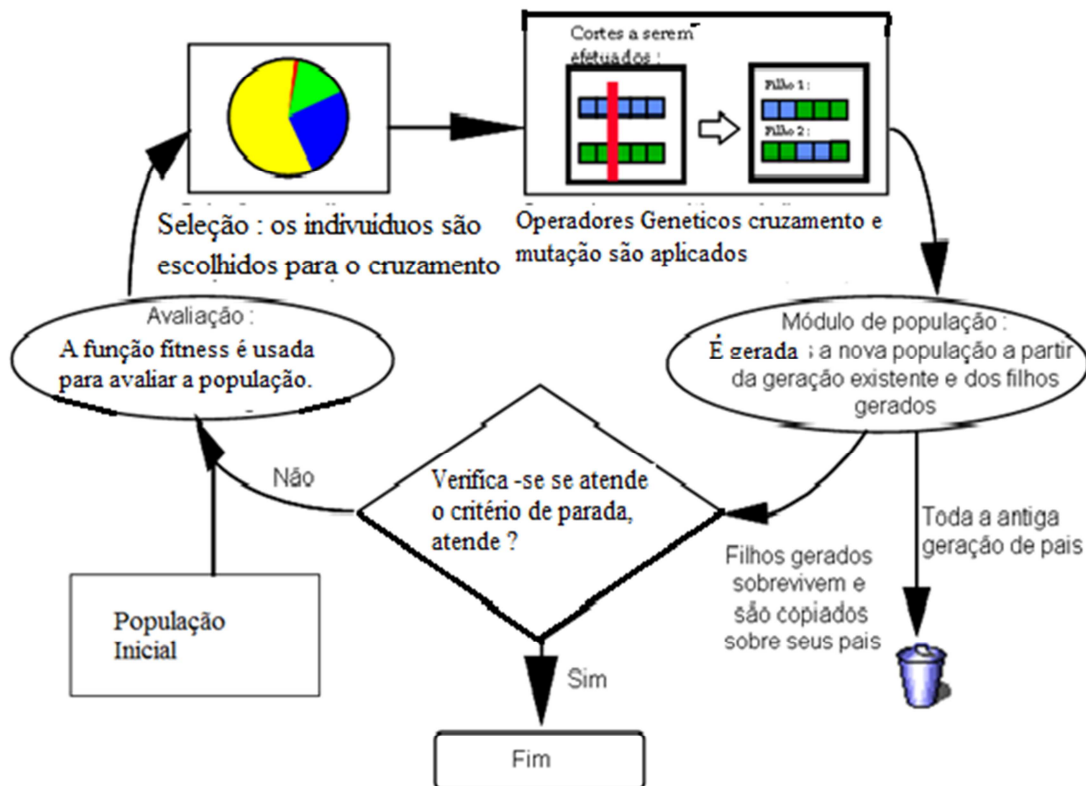


Figura 4 estrutura básica de AG. Adaptado de Algoritmo Genético- Rio de Janeiro : Brasport 2006

Como pode ser verificado a estrutura de funcionamento de um AG é extremamente simples, mas atrás desta simplicidade existe uma complexidade na obtenção de uma representação cromossomial que seja representativa para o problema e na escolha da função de avaliação que penalize os indivíduos que não atendem ao problema premie os que realmente se aproximam da solução do problema.

### 3.1.1 - Representação Cromossomial

A representação cromossomial é fundamental para a solução do problema, pois AGs não trabalham com os problemas e sim com suas representação, se esta não representa o problema de maneira fidedigna a solução não será aplicável ao problema.

A representação cromossomial consiste em uma maneira de traduzir a informação do problema em uma maneira viável de ser processada pelo computador, dentro dessa estrutura de funcionamento do AG. Se a representação escolhida consegue representar adequadamente o problema, maior a qualidade dos resultados obtidos pelo processamento do AG.

Uma questão que deve ser evitada é a tendência a adaptar o problema a nossa representação, normalmente isso leva a complicações, pois cada problema é único e consequentemente tem a sua representação diferente de outra representação que um dia resolveu outro problema.

A representação pode ser:

- Binária;
- Real;
- Inteira.

Cada cromossomo é composto de genes, e os genes são partes indivisíveis da representação, devem ser avaliados como partes do cromossomo.

Algumas regras devem ser observadas e a primeira delas é básica, trata-se da complexidade da representação, o que deve ser evitado. A representação deve ser a mais simples possível, isto possibilita respostas mais diretas consequentemente mais rápidas.

As soluções proibidas ao problema, se houverem, não devem fazer parte do espaço de soluções do problema, portanto não devem ter uma representação.

Se houver imposições de algum tipo elas devem estar implícitas na representação, isto significa que as imposições devem conduzir as respostas e fazer parte da solução do problema.

A representação binária foi a originalmente adotada por Holland em seu livro seminal, trata-se de uma representação de fácil entendimento e implementável sendo considerada a mais simples e mais usada pelos praticantes dos AGs. LIDEN, R (2006) .

Na representação cromossomial binária cada bit representa um gene e o conceito representado pelo bit ou conjunto de bits é inerente ao problema.



O cromossomo, portanto nada mais é que uma sequência de bits, uma string.

Os números (inteiros ou reais) podem ser representados pela representação binária, bastando que se saia duas características desta representação:

- 1) A faixa de valores que este número pode representar
- 2) A precisão que se deseja desta representação.

Uma fórmula para converter os binários para reais, sabendo-se as características da representação é dada pela seguinte expressão:

$$Real = \inf_i + \left( \frac{\sup_i - \inf_i}{2^i - 1} \right) * r_i \quad (3)$$

Onde:

$\inf_i$  é o limite inferior da representação;

$\sup_i$  é o limite superior da representação;

$r_i$  é o valor da conversão binária para inteiro.

Assim uma string composta pelos seguintes binários;

$$S = 0001000001110011$$

$$S = \overset{gene1}{00010000} \overset{gene2}{01110011}$$

Em que cada gene é representado por 8 bits, tem-se:

$$r_1 = 00010000 = 16$$

$$r_2 = 01110011 = 115$$

Supondo que a faixa de representação do gene 1 seja de -2 a +2

$$Real_1 = (-2) + \left( \frac{2 - (-2)}{2^8 - 1} \right) * 16 \quad (4)$$

$$Real = -1,749$$

Supondo que a faixa de representação do gene 2 seja de 0 a +2

$$Real_2 = (0) + \left(\frac{2-(0)}{2^8-1}\right) * 115$$

$$Real_2 = 0,902$$

## 3.2 - População Inicial

A inicialização dos cromossomos no início do GA é chamada de formação da população inicial e apesar de ser simples e os valores em si não serem importante envolve algumas características que interferem no funcionamento do AG.

A população inicial é aquela que será a primeira a ser avaliada pela fitness, função de avaliação, e deve conter características pertinentes ao problema.

A primeira característica a ser observada é o espaço de busca, a população deve estar contida no espaço de busca.

O espaço de busca é a região onde o AG irá pesquisar em busca da solução do problema, que garantidamente deve estar dentro do espaço de busca.

A população inicial, a primeira população do AG tem uma característica muito importante para a resolução do problema, trata-se do número de indivíduos da população, este número está intimamente ligado ao sucesso do AG. Um número pequeno de indivíduos pode prejudicar a qualidade da população, pois não há diversidade de pais, conseqüentemente não haverá diversidades de filhos, e tanto a convergência, quanto o tempo de convergência pode ser prejudicado. Um grande número de indivíduos prejudica o tempo de processamento do algoritmo, pois varias etapas exigem que haja operações com cada indivíduo como, por exemplo, a avaliação, todos devem ser avaliados a cada geração ou iteração.

Portanto, existe um número ótimo de indivíduos para a solução do problema ou existe uma faixa de números em que a solução é obtida sem prejuízo de qualidade da resposta ou tempo de processamento.

A inicialização dos indivíduos da população inicial a priori não deve ter muita importância, grande parte dos autores aconselham a inicialização randômica, a qual irá garantir uma distribuição randômica dos indivíduos dentro do espaço de busca. Uma distribuição equidistante também pode ser usada, sendo que em alguns casos contribui

para a redução do tempo de execução do AG na solução do problema, porém não contribui significativamente, conforme foi verificado em vários experimentos.

O programa a seguir dá uma ideia de inicialização randômica de uma representação binária na linguagem java.

```
private void inicializa Elemento(int tamanho)
{
    int i;
    this.valor="";
    for(i=0;i<tamanho;++i)
    {
        if (java.lang.Math.random()<0.5)
        {
            this.valor=this.valor+"0";
        }
        Else
        {
            this.valor=this.valor+"1";
        }
    }
}
```

A string resultado da operação será preenchida com 0 ou 1 conforme o gerador randômico processar as suas saídas.

### 3.3 –Avaliação

Uma vez constituída a População Inicial, esta precisa ser avaliada e nesta avaliação está todo o sucesso do AG, pois uma má avaliação pode levar o AG a não encontrar a solução do problema.

A função de avaliação em muitos casos é a única ligação real entre a representação do problema e o problema real devido a generalidade dos AGs.

A função de avaliação, também chamada de função custo ou “fitness”, penaliza os indivíduos que se afastam da solução e premia os que mais se aproximam da solução, isto é feito pela atribuição de uma nota a cada indivíduo, essa nota está baseada no resultado

da avaliação é proporcional a sua capacidades de serem soluções para o problema, assim o pior avaliado recebe a pior nota e o melhor avaliado receberá a melhor nota.

O resultado da avaliação é extremamente importante, pois a avaliação define os indivíduos que mais se adaptaram aquele ambiente, as melhores respostas para o problema naquela geração, portanto define os indivíduos que tem condições de ser escolhidos para o cruzamento os quais poderão transmitir suas características para a próxima geração.

A função de avaliação deve conter o conhecimento sobre o problema, pois a mesma propiciará a seleção do melhor elemento daquela geração e permitirá a classificação dos demais elementos da geração para que se saiba quais podem participar do cruzamento.

### **3.4 – Seleção**

A seleção é o mecanismo que permitirá ao Algoritmo Genético a aproximação do processo de evolução natural da natureza, através da escolha dos mais aptos (os que obtiveram as melhores notas da avaliação) os quais terão mais chances de realizarem o cruzamento e passar seus genes para os filhos que farão parte da nova população.

Existem diversos métodos de seleção, os principais métodos usados:

- Método da Roleta;
- Método do Torneio;
- Método de Amostragem Estocástica Uniforme;
- Seleção Local ;
- Seleção por Ranking;
- Seleção Truncada.

O processo de seleção mais simples é o conhecido como roleta, que consiste de uma roleta viciada na qual um determinado número de indivíduos são distribuídos conforme a sua aptidão, ou seja, conforme a sua nota na avaliação.

Os indivíduos com maiores notas ocupam regiões maiores do disco da roleta e os indivíduos com menores notas ocupam regiões menores, mas participam da rodada, uma vez dividido todo o espaço do disco entre os participantes da rodada é executada a

roleta. O giro da roleta será simulado por uma rotina que executa um gerador randômico que é programado nos AGs e seu resultado funciona como o ponteiro da roleta, marcando o ganhador da rodada que será o primeiro participante do cruzamento, o processo se repete com os restantes que são novamente distribuídos no espaço da roleta e o resultado fornece o segundo participante do cruzamento.

A distribuição dos indivíduos no disco da roleta obedece ao critério de proporcionalidade em relação a sua aptidão, mas isso não garante o sucesso na escolha. Ao executar o giro da roleta a probabilidade passa a fazer parte do AG e muitas vezes o indivíduo que possui as maiores chances de ser escolhido não é, pois a roleta é aleatória e pode parar em outro indivíduo, neste processo o indivíduo com grandes notas tem grandes possibilidades de ser escolhido, pois o espaço ocupado é proporcional a nota da avaliação, a tabela 2 mostra uma distribuição de indivíduos sua participação em porcentagem e em graus.

Tabela 2 Distribuição de indivíduos no disco da roleta

| Indivíduo | Avaliação | Participação(%) | Participação (°) |
|-----------|-----------|-----------------|------------------|
| 000010    | 10        | 7,14            | 25,71            |
| 001010    | 20        | 14,29           | 51,43            |
| 011011    | 80        | 57,14           | 205,71           |
| 010111    | 30        | 21,43           | 77,15            |
| Total     | 140       | 100%            | 360°             |

A distribuição de indivíduos no disco da roleta é proporcional a avaliação do indivíduo logo o indivíduo 011011 que obteve a nota de avaliação 80 ita ter uma participação maior no disco da roleta ocupando 57,14% da área do disco, isto faz com que sua possibilidade de escolha seja muito maior que o indivíduo 000010, que obteve a nota 10 e sua participação no disco será de apenas 7,14% da área do disco, mas com possibilidade de escolha .

Cada indivíduo é então alocado no disco ocupando uma área proporcional à sua avaliação, conforme mostrado na fig. 5 a seguir.

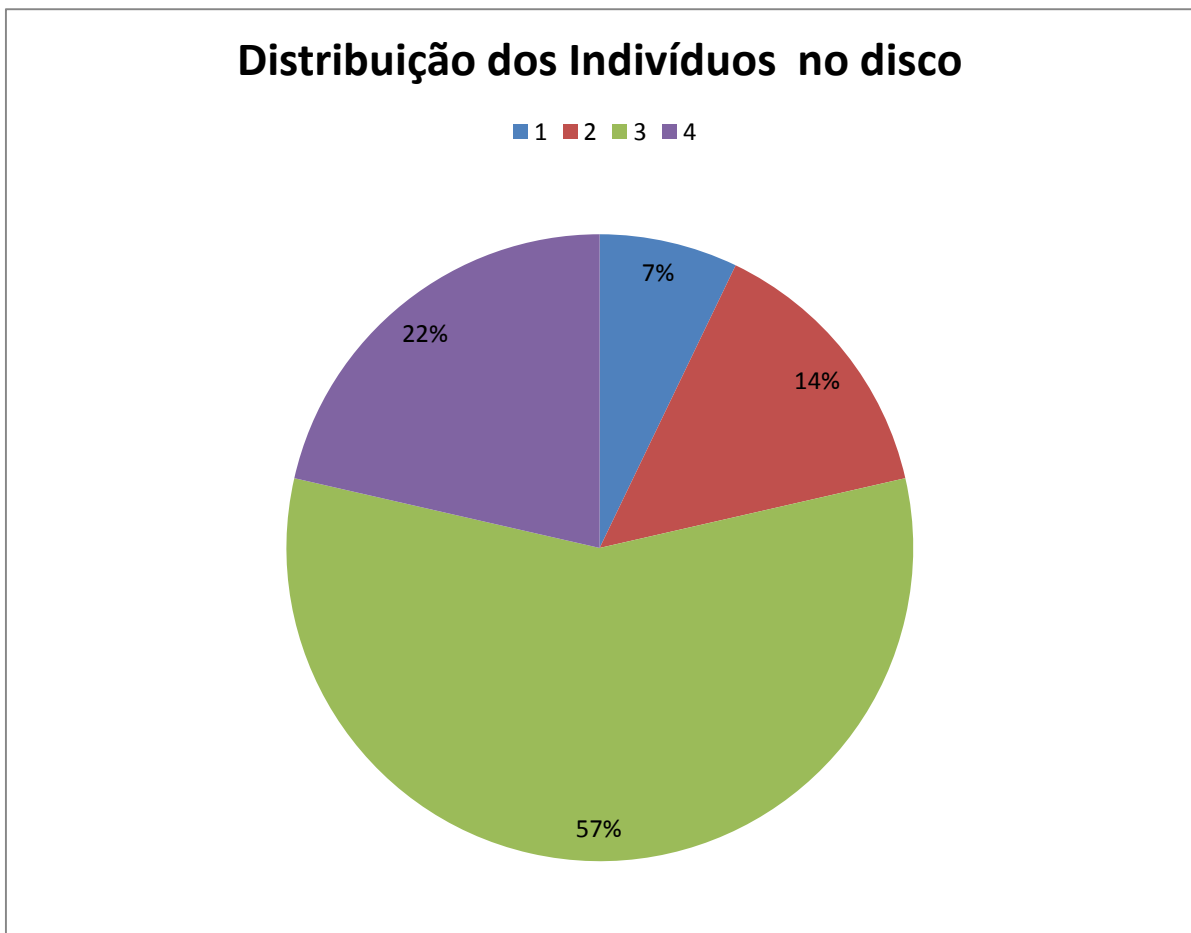


Figura 5 distribuição de indivíduos no disco da roleta

Como o indivíduo que recebeu a maior nota ocupa o maior espaço na roleta, isto implica em suas chances de escolha ser maior que dos demais indivíduos, assim como na natureza, quem melhor se adapta tem mais chances.

Implementação a roleta em Java:

```

1. public int roleta() {
2.     int i;
3.     double aux=0;
4.     calculaSomaAvaliacoes();
5.     double limite=Math.random()*this.somaAvaliacoes;
6.     for(i=0;
       ((i<this.populacao.size())&&(aux<limite))
       ;++i) {
7.         aux+=((ElementoGA)populacao.get(i)).getAvaliacao();

```

```
8.      }  
9.      i--;  
10.     return(i);  
11.     }
```

O programa utiliza a função randômica para girar a roleta, esta função torna a rodada totalmente imparcial, e também não repetitiva, pois a cada rodada o resultado será diferente.

Outro método de seleção é o torneio que consiste em tomar 4 (quatro) indivíduos da população por uma escolha randômica, montar um torneio dois a dois, e neste torneio o que tem a maior aptidão vence, assim o indivíduo com maior nota não será perdido, a cada par será selecionado um pai, que participará do cruzamento

O método do torneio funciona muito bem para a representação real, sendo o mais indicado para esta representação. Este procedimento tem apresentado melhores resultados que os outros operadores de seleção para uma grande gama de aplicações, GOLDBERG (2002) , DEB (2002) e DE JONG(2006) .

### **3.5 – Cruzamento**

O cruzamento é a transmissão das características para os filhos da próxima geração, o processo de cruzamento na natureza é a procriação.

O processo de cruzamento pode acontecer por diversos métodos, no caso da representação binária o cruzamento acontece com troca de material genético dos pais.

O cruzamento mais simples é cruzamento de um ponto, mas pode ocorrer cruzamento de várias formas, sempre lembrando que o cruzamento tem por finalidade gerar uma nova população que recebe as características dos pais, que foram selecionados para o cruzamento por apresentarem um bom desempenho como solução do problema.

### 3.5.1 -Cruzamento de um ponto

O cruzamento de um ponto é feito pela escolha de um ponto de corte na string, sendo que uma string com  $n$  bits possui  $n-1$  pontos de corte, as strings são então separadas no ponto de corte e as partes a direita do corte são trocadas e concatenadas as partes a esquerda do corte, sendo que o filhos tem partes do pai1 e do pai2

A fig. 6 mostra os pontos de corte de uma string de 5 bits, consequentemente 4 pontos de corte.

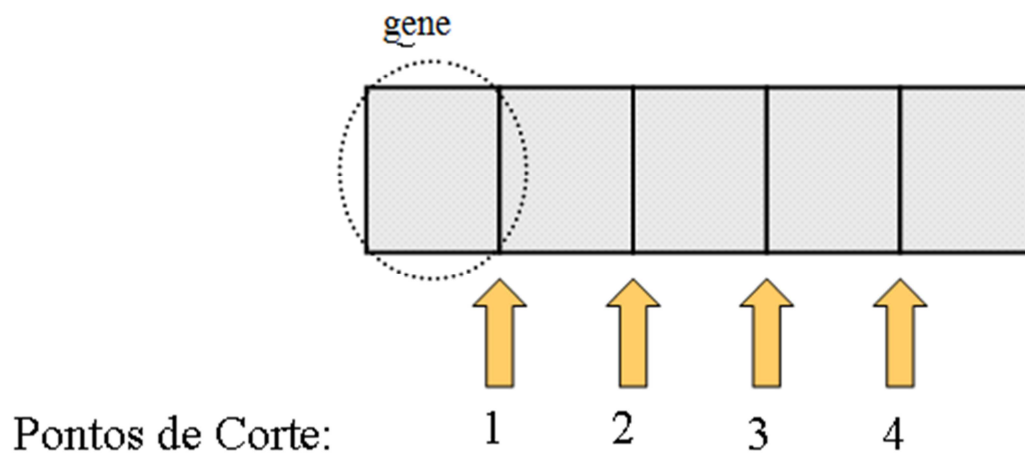


Figura 6 pontos de corte de strings com 5 bits

Uma vez selecionado o ponto de corte é realizada a separação do cromossomo em duas partes, que serão combinadas para formar os indivíduos filhos, ilustrado na fig. 7.

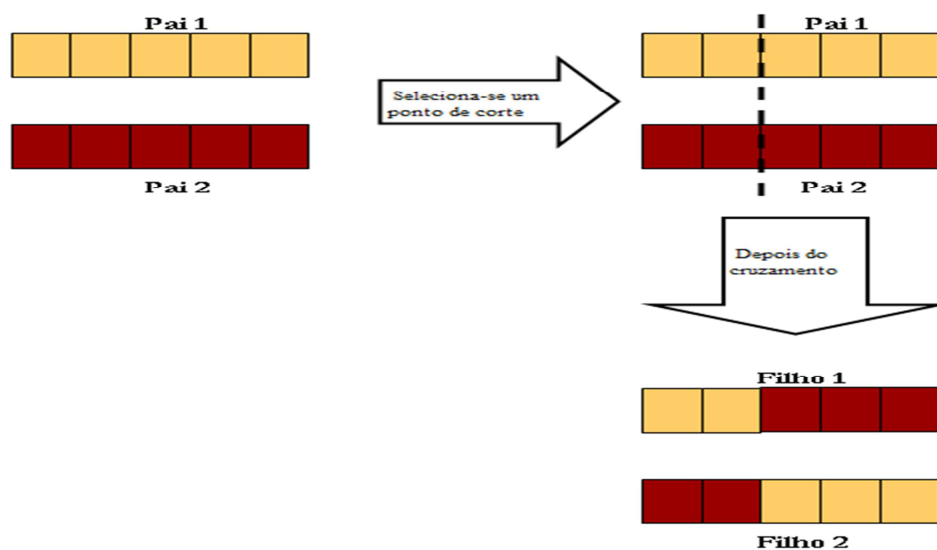


Figura 7 Processo de cruzamento de um ponto adaptado de Algoritmo Genético- LIDEN 2006.



### 3.5.2 Cruzamento de dois pontos

O Cruzamento pode ser em mais de um ponto, como no cruzamento de dois pontos, neste caso são selecionados dois pontos de corte na string e o material é trocado entre os pais, como mostrado na fig. 8.

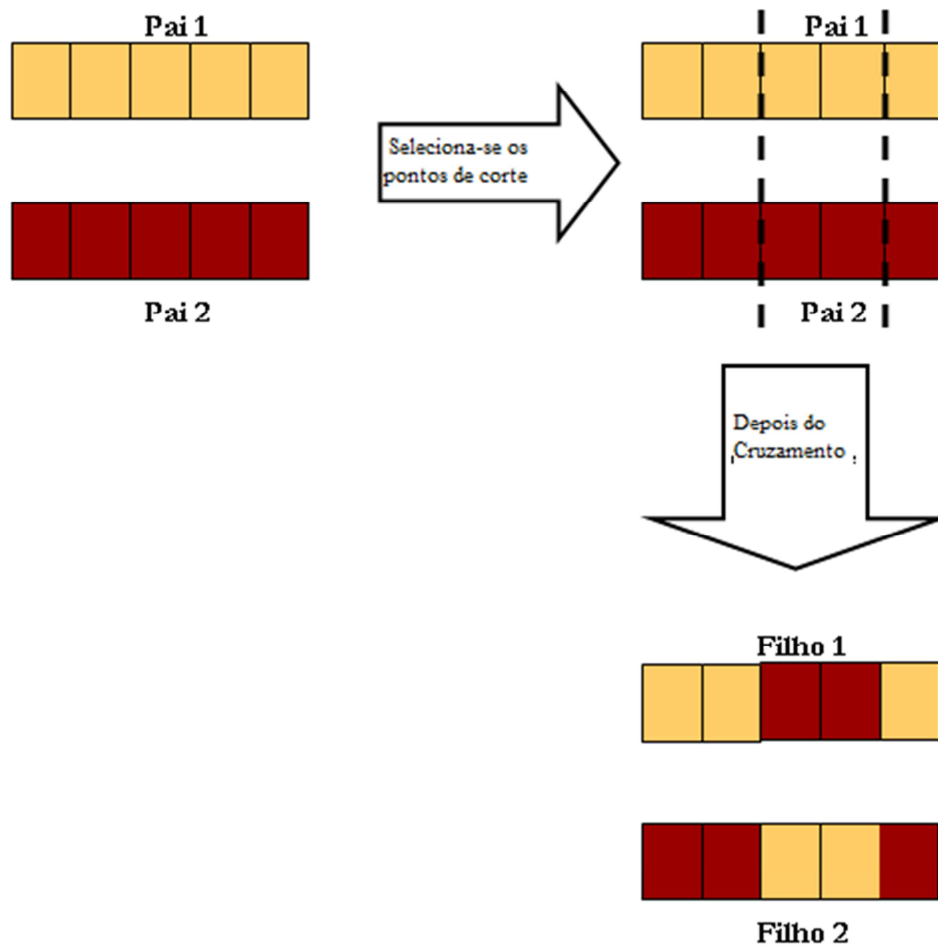


Figura 8 Processo de cruzamento de dois pontos adaptado de Algoritmo Genético- LIDEN 2006

### 3.5.3 – Cruzamento pela média aritmética

O cruzamento pode ser feito pela média aritmética entre os pais, basicamente o cruzamento visa a transferência de características dos pais para os filhos, quer sejam explícitas ou implícitas.

A equação (6) mostra o processo de média aritmética que consiste na soma de dois indivíduos e divisão por dois.

$$Filho = \frac{Pai1 + Pai2}{2} \quad (6)$$

### 3.5.4 – Cruzamento pela media geométrica

O Cruzamento pela média geométrica é obtido pela raiz quadrada da soma dos quadrados dos pais. Este tipo de cruzamento também transmitem as características e tendências dos pais para as novas gerações.

A equação (7) mostra a equação matemática que executa a média geométrica

$$Filho = \sqrt{Pai1 * Pai2} \quad (7)$$

### 3.6 – Mutação

Após o cruzamento acontece o processo de mutação que consiste em modificar o gene segundo uma baixa probabilidade, o que confere ao AG a característica de não estagnar em mínimos locais.

Esta baixa probabilidade implica que a mutação não ocorre para todos os indivíduos, apenas ocorre para alguns indivíduos, pois a mutação tem como finalidade evitar a parada na evolução de uma resposta em uma boa solução que ainda não evoluiu totalmente.

Os genes são alterados para valores aleatórios ou não, apenas modicam seus valores de forma a criar a diversidade que pode, às vezes, ser perdida no processo de seleção ou cruzamento.

A probabilidade de mutação deve ser, ao contrário da probabilidade de cruzamento muito pequena, pois não deve interferir na direção de convergência, isto é, se a probabilidade de mutação for grande a alteração no indivíduo pode modifica-lo de tal forma que ele perca as características que o levou a se adaptar até a geração anterior, ou

em outra hipótese, transformar a busca em uma busca cega ou em uma caminhada randômica (random walk).

Normalmente a probabilidade de mutação é menor que 10% enquanto que a probabilidade de cruzamento é maior que 90%, SOUZA, 2008.

A fig. 9 mostra o processo de mutação.

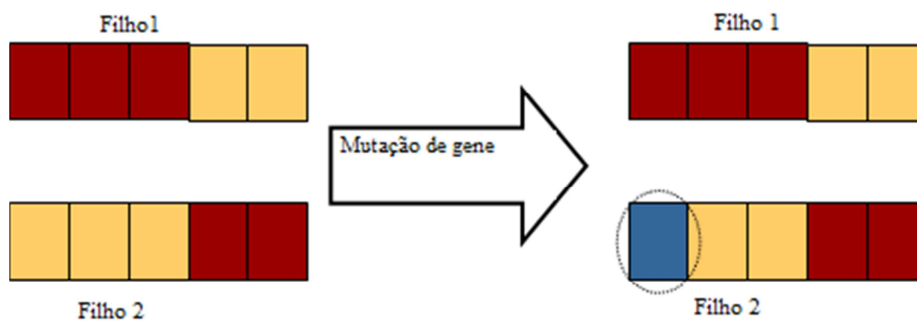


Figura 9 Processo de Mutação fonte Algoritmo Genético- LIDEN (2006)

O gene que sofreu mutação vai assumir um valor, no caso binário 0 ou 1.

A mutação tem a finalidade de evitar a estagnação, isto é, evita que um mínimo local prevaleça e seja tomado como solução de um problema. Este processo complementa os processos de operadores genéticos.

### 3.7-Formação da nova população

A nova população deve ter o mesmo número de indivíduos que a população anterior, já foram feitas tentativas de aproximação maior entre a natureza e os AGs. Aumento na população ou mesmo diminuição como na natureza acontecem, pois as populações estão variando constantemente com nascimentos, mortes e imigrações e essas variações acabam interferindo na solução do problema.

Essa alteração na população interfere na estrutura do AG e de certa forma, torna esta estrutura muito complexa, pois exige controle de outros parâmetros, caminhando em sentido contrário aos princípios dos AGs, que é a simplicidade e evolução natural.

Um AG em que o tempo de processamento seja importante pode ter esse artifício, mas nos casos de FPGAs temos limitações de hardware, pois os indivíduos são posições de memórias, implicando em espaços físicos dentro do dispositivo.

### 3.8 – Elitismo

O elitismo é uma premiação ao indivíduo que obteve a melhor avaliação nas últimas iterações e, se não houver indivíduos que se aproximem mais da solução não haverá indivíduos com melhores avaliações e, conseqüentemente o elite será a solução do problema.

O elemento elite como premio automaticamente fará parte da nova população, pois ele pode ser a resposta do problema, isso proporciona a segurança de que a melhor solução não será perdida no processo. O elemento elite deve participar da seleção e no caso dos torneios ele sempre será um dos pais, pois possui a melhor nota na avaliação, porém pode não ser convocado para participar do torneio uma vez que a convocação é executada por uma escolha randômica.

A participação do elemento elite no processo de cruzamento possibilita a transmissão de gens de parte da solução ou do caminho para a solução para o filho, isto é uma grande vantagem na orientação da busca, principalmente se o processo de iterações se encontra em uma fase avançada.

### 3.9 - Critério de parada

O Critério de parada é quem define o momento que o AG chegou ao final e pode acontecer por três motivos:

- a) Número de iterações máximas;
- b) Não alternância do elemento elite;
- c) Tolerância no erro.

O momento da parada do AG é quando a solução encontrada para o problema será apresentada como a melhor resposta do AG, este momento é muito importante porque ele finaliza a operação do AG, permitindo conhecer o tempo de processamento do AG, sua eficiência e eficácia.

O primeiro critério é o número de iteração. É um número estimado pelo programador do AG que tenta usar sua experiência para estimar um número de iterações nas quais seja possível o AG convergir para a solução do problema. Normalmente este número é

superestimado, pois é preferível um erro no tempo de processamento do que na solução do problema. Uma estimativa insuficiente provoca o término da execução do AG. antes da convergência e conseqüentemente a solução pode não ter evoluído ao ponto de ser a melhor solução para o problema, por outro lado a tentativa de uma superestimava pode prejudicar o tempo de processamento.

Existe um compromisso entre o tempo de processamento e a qualidade da solução que varia de problema para problema em função da função de avaliação e da natureza do problema, fazendo com que nem sempre seja fácil estimar o número de iterações necessárias à convergência da solução.

O segundo critério de parada é o que verifica a não alternância do elemento elite por um determinado tempo. O elemento elite é a solução do problema e se este não se altera por determinado número de iterações, é devido à solução ter sido encontrada e nenhuma avaliação melhor que a do elemento elite tenha sido encontrada no processamento do AG. De certa forma apenas transferimos o problema do número de iterações, pois agora é necessário estimar o número de iterações que a resposta deve permanecer inalterada, para que se possa afirmar que ela é uma resposta ao problema.

O terceiro critério de parada é o mais correto, porém sua implementação mais trabalhosa, trata-se de estabelecer uma tolerância ao erro, uma margem que quando atingida permita que o AG suspenda a evolução e forneça a solução atual (elemento elite), esta margem define o compromisso ente qualidade e tempo de programação. Em aplicações em tempo real, como no caso do FPGAs, em que o tempo de processamento é um fator limitante de atuação este critério de parada é bastante útil.

### **3.10 - Paralelismo implícito**

Os AGs possuem um paralelismo implícito, o processo de avaliação é um processo paralelo, cada individuo é avaliado em cada iteração. No processo de seleção existe um processo paralelo, pois um número determinado de indivíduos é selecionado e o processo de cruzamento em que cada par de pais geram os filhos em paralelo a outros pares de pais.

O paralelismo pode ser explorado por dispositivos como os FPGAs que possuem mecanismos paralelos, isso faz com que o tempo de iteração seja reduzido e a sua utilização otimizada, permitindo o uso em aplicações em tempo real.

A estrutura paralela dos AGs pode ser processada de forma sequencial, a única diferença será no tempo de processamento do algoritmo, que em certas aplicações, não tem influência e permite a sua utilização.

## 4 – Resultados

Inicialmente o AG é aplicado na análise de um sinal de tensão da rede elétrica doméstica 127V60Hz. Normalmente uma análise Fourier em relés de frequência, onde a estimação da frequência da rede é necessária para a atuação do relé e a manutenção da estabilidade do sistema elétrico não permitindo-se variar acima de (+) 0,5 Hz ou abaixo de (-) 0,5 Hz da frequência nominal.

A aquisição do sinal de entrada é feita através do sistema de aquisição de dados da National Instruments NI USB-6009 que possui oito entradas, 14-bit, Multifunção E/S que fornece um arquivo de dados que serão usados pelo Algoritmo Genético e FFT no MatLab®.

A fig. 10 apresenta o sistema de aquisição, composto de um computador, o sistema NI-DAQ™mx para USB que é usado para a aquisição do sinal real.

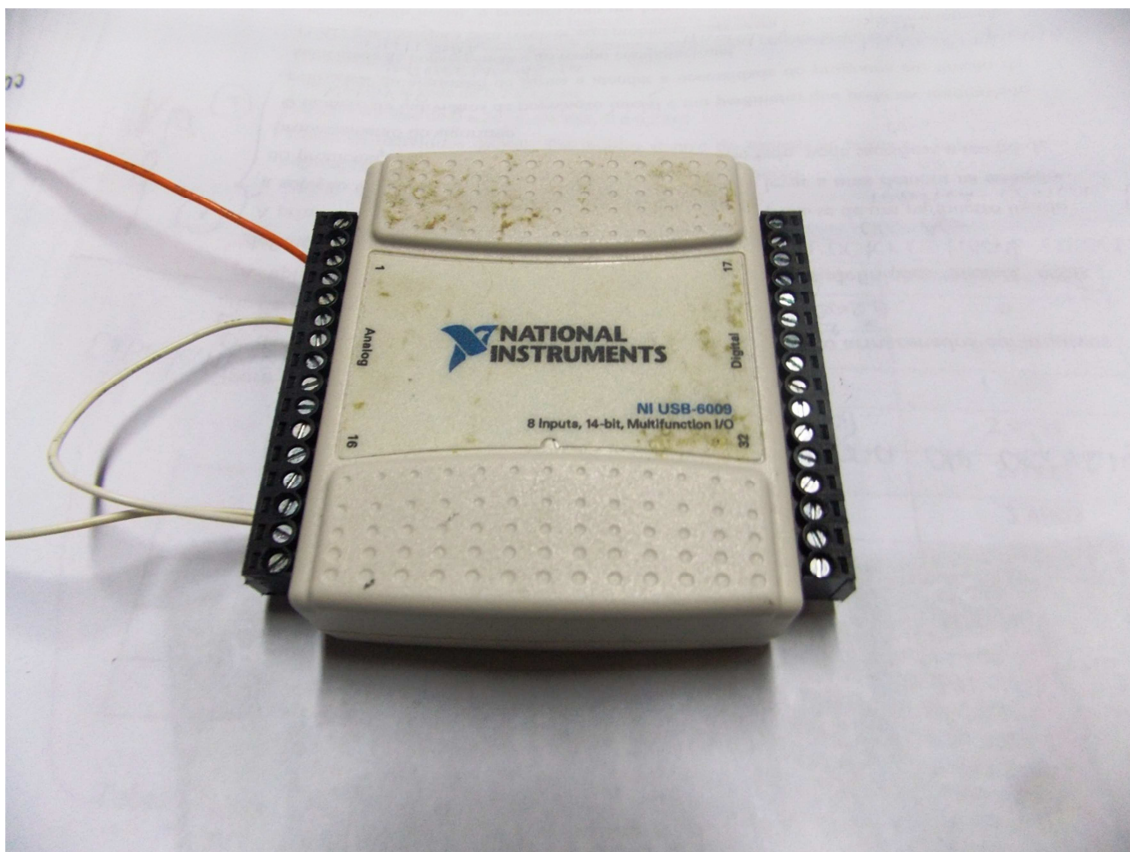


Figura 10 Placa de Aquisição de dados NI-DAQ™mx USB-6009 da National Instruments

A fig. 11 apresenta o diagrama de blocos do sistema de aquisição, composto de um computador, o sistema NI-DAQ™mx para USB que é usado para a aquisição do sinal real.

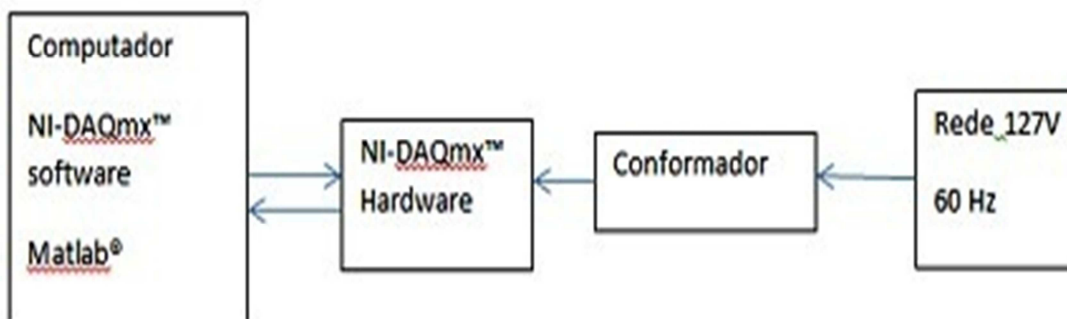


Figura 11 diagrama de blocos do sistema de aquisição de dados

O sistema de aquisição de dados reais foi implementado conforme a fig. 12, um computador notebook Samsung, com software MatLab® e o software da National NI-DAQmx™, a placa de aquisição NI-DAQmx™, a placa conformadora de sinal e o sinal.

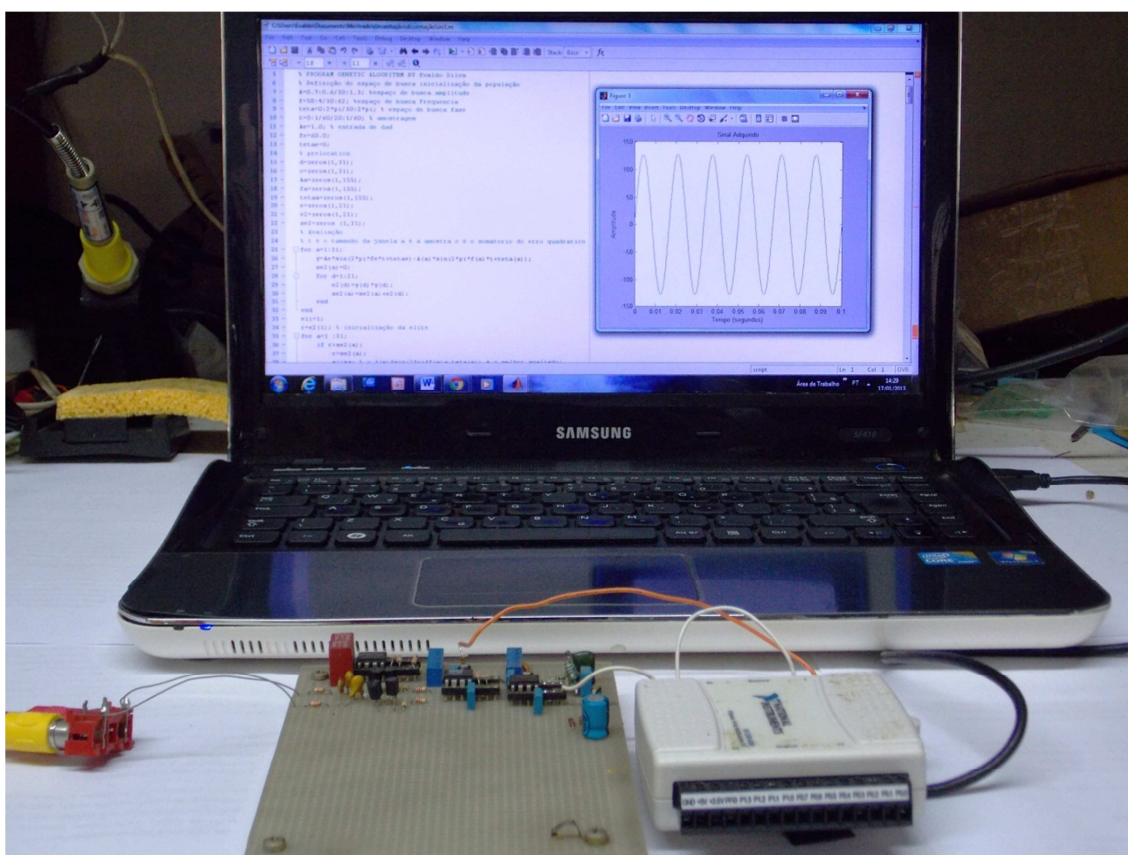


Figura 12 sistema de aquisição de dados utilizando placa NI-DAQ™mx USB 6009



A placa conformadora foi montada com intuito de limitar o sinal real dentro da faixa de operação de entrada da placa NI USB 6009, realizando também uma filtragem do tipo passa baixa do sinal, com objetivo de eliminar os componentes harmônicos do sinal real.

O sistema de aquisição de dados utilizando a placa da National Instruments possibilitou a criação de um arquivo de dados que foi utilizado pelo Matlab® na geração de figuras e processamento do AG.

O sinal de entrada adquirido foi uma senoide de 60 Hz conforme mostra a fig. 13.

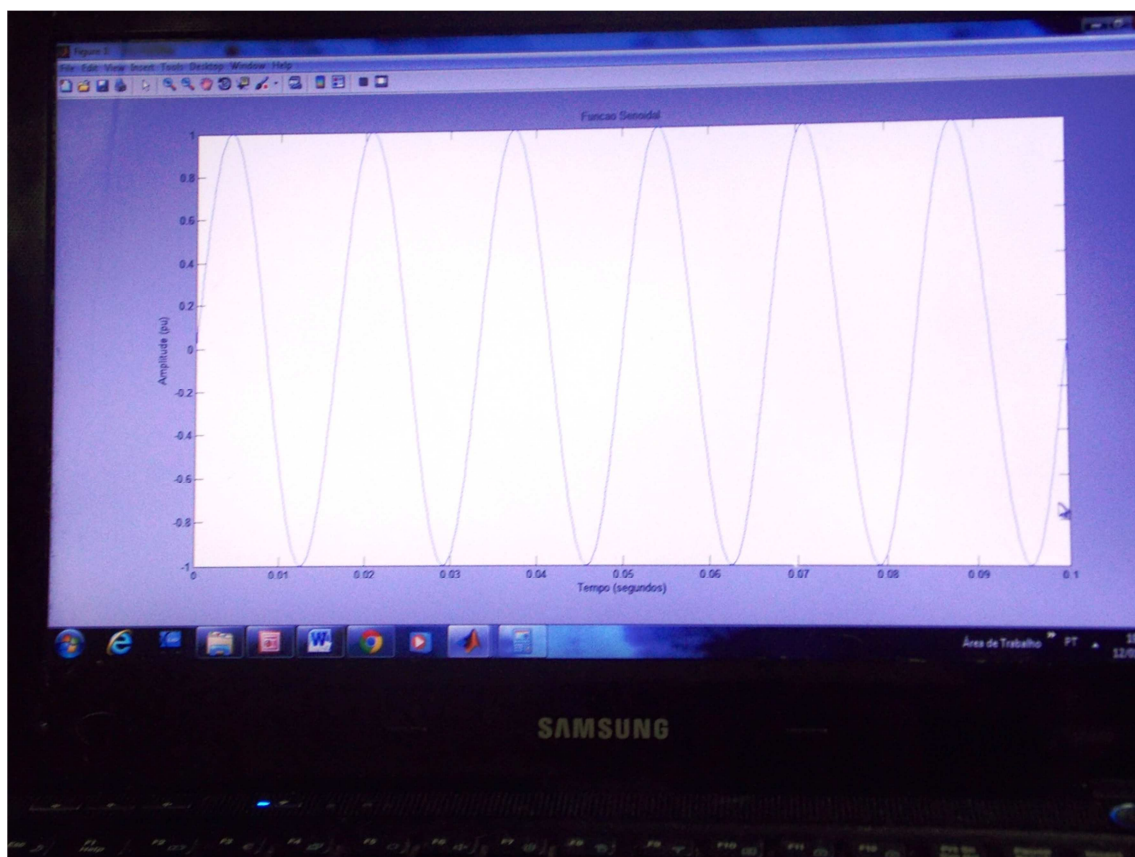


Figura 13 Sinal adquirido pelo sistema de aquisição NI-DAQ™mx USB.

A equação representativa da forma de onda do sinal adquirido foi mostrada em (1).

$$x(t) = A * \sin(2 * \pi * f * t + \theta) \quad (1)$$

Onde

$x(t)$  é a amostra do sinal

$A$  é amplitude do sinal

$\sin$  a função matemática seno

$\pi$  é a constante pi 3,1415 rad

$\Theta$  é o ângulo de fase do sinal

$t$  é a taxa de amostragem

Os valores foram normalizados de acordo com :

Amplitude da senoide  $A= 1$  pu

Frequência da senoide  $f = 60\text{Hz}$

Ângulo de fase  $\theta= 0$  rd.

Taxa de amostragem do sinal será  $t= 1/800$

Portanto de acordo com Nyquist, um sinal amostrado só pode ser recuperado se a frequência de amostragem for 2 vezes superior a maior frequência presente no sinal, para o caso de 60 Hz uma taxa de amostragem de 1/120 é suficiente para recuperar a forma de onda do sinal.

Os pontos amostrados no intervalo estão marcados na figura.

Assim o sinal gerado será conforme a fig. 14.

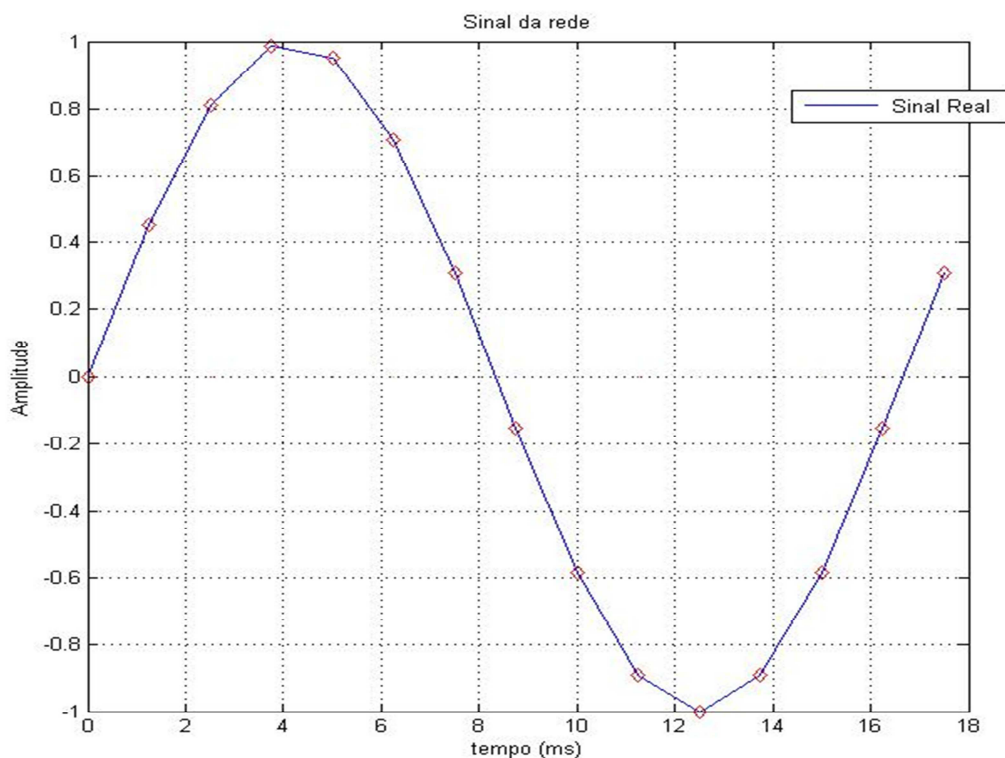


Figura 14 Sinal Real obtido no período de 18 ms a uma taxa de 1,25 ms

Estes valores são usados como entrada de dados para o AG (algoritmo genético), trata-se um sinal real adquirido pelo sistema A/D utilizado.

A opção por uma taxa de amostragem 1/800, se deve ao fato de o sinal obtido se aproximar o mais possível de uma senoide pura, portanto, a possibilidade de ruídos ou harmônicos são mitigados, o que permite uma análise das características do sinal a ser estimado sem a influência de variáveis externas que poderiam interferir na análise do resultado.

Para os casos práticos conforme é discutido apropriadamente no tratamento do sinal, utiliza-se de recursos para adequar a entrada às necessidades da aplicação, valendo-se de filtros e métodos de regularização do sinal como limitação de amplitude através de um AGC (Automatic Gain Control, controle automático do ganho), conformador dentro da faixa de excursão do conversor A/D, bem como da limitação de banda do sinal de entrada, entre outros.

Cabe lembrar que os dados dos sinais são armazenados em arquivos em formatos apropriados para análises e posterior utilização por programas.

## **4.1 Aplicação do Algoritmo Genético a sinais periódicos**

A aplicação do AG algoritmo genético depende de algumas definições iniciais, essas são características usadas para aplicação do AG.

1. A primeira é o tipo de representação: binária ou real, deve-se avaliar qual representação é mais adequada ao problema.
2. A segunda é o tamanho da população inicial. Trata-se de um parâmetro ligado a solução do problema, uma população pequena pode levar a uma convergência rápida do problema, sacrificando a precisão da resposta, que será mais precisa se a população inicial for maior, no entanto, uma população grande demais, pode elevar o tempo de processamento do algoritmo.

O número de indivíduos da população inicial é um parâmetro que pode ser manipulado pelo usuário de forma a atender à necessidade do programa em função da velocidade do processo de convergência e tempo computacional.

No AG utilizado para resolver o problema inicial apresentado, foi utilizado três casos para a população inicial: o primeiro com um baixo número de indivíduos 10 apenas; o segundo, com um número considerado adequado, 30 indivíduos, e o terceiro com um

número excessivo de indivíduos 50, segundo varias experiências de execução dos AGs para este problema.

Outro parâmetro extremamente importante é a definição do espaço de busca, esse parâmetro deve garantir que a solução esteja dentro de certos limites, porém a interferência na velocidade de convergência é pequena, pois a população de indivíduos é distribuída dentro do espaço de busca de maneira equidistante.

Os seguintes parâmetros limites são usados neste estudo:

Frequência da rede de 58 a 62 Hz;

Amplitude da rede de 0,7 a 1,3 pu;

Ângulo de fase de 0 a  $2\pi$  rd, ou seja, 0 a 6,28rd.

#### 4.1.1 Aplicação de AG com populações de indivíduos pequenas

Inicialmente é realizado o estudo de o AG com população pequena 10 indivíduos, este caso é bem detalhado o que permite analisar o desempenho e funcionamento do AG.

Como os indivíduos são distribuídos dentro do espaço de busca adotado, pode-se descrever para o caso 1 os indivíduos da população inicial conforme a tabela 3.

Tabela 3 indivíduos da população inicial distribuídos equidistantemente no espaço de busca.

| Individuo | Amplitude (Ae) | Frequência (fe) | angulo de fase( $\theta_e$ ) |
|-----------|----------------|-----------------|------------------------------|
| 1         | 0.70           | 58.00           | 0                            |
| 2         | 0.77           | 58.44           | 0.69                         |
| 3         | 0.83           | 58.89           | 1.39                         |
| 4         | 0.90           | 59.33           | 2.09                         |
| 5         | 0.97           | 59.78           | 2.79                         |
| 6         | 1.03           | 60.22           | 3.49                         |
| 7         | 1.10           | 60.67           | 4.18                         |
| 8         | 1.17           | 61.11           | 4.88                         |
| 9         | 1.23           | 61.56           | 5.58                         |
| 10        | 1.30           | 62.00           | 6.28                         |

Alguns pontos podem ser ressaltados, a população inicial foi distribuída dentro do espaço de busca de maneira equidistante, mas poderia ser randomicamente distribuída, sem prejuízo para a convergência do AG. Essa população inicial deve ser avaliada, isto é sua aptidão para adaptação ao ambiente deve ser verificada, assim a avaliação permitirá atribuir uma nota a cada indivíduo.

A avaliação é feita em função de ajuste ou fitness que consiste na comparação do sinal de entrada com o sinal estimado composto por cada indivíduo, e desta comparação resultará em um erro o qual minimizado corresponderá ao sinal de entrada.

A função fitness ou erro é apresentada na equação 2:

$$e(t) = x(t) - y(t) \quad (2)$$

Onde:

$e(t)$  = erro

$x(t)$  = sinal de entrada medido (sinal senoidal de amplitude 1.0 pu, frequência 60 Hz e ângulo de fase 0 rd)

$y(t)$  = sinais estimados usando os indivíduos da população inicial ( família de senoides geradas pelos parâmetros da tabela segundo a equação 3:

$$y(t)_i = Ae_i * \sin(2 * \pi * fe_i * t + \theta e_i) \quad (3)$$

Onde:

$y(t)_i$  = enésimo sinal estimado;

$Ae_i$  = amplitude do individuo i;

$fe_i$  = frequência do individuo i;

$\theta e_i$  = ângulo de fase do individuo i.

Adota-se os valores do erro quadrático calculado segundo a equação (4):

$$e^2(t)_i = (x(t) - y(t)_i)^2 \quad (4)$$

Se a comparação for feita ponto a ponto a precisão do método é ruim e o resultado distorcido. Para um aumento na precisão, se considera mais amostras no processo, e o erro passa a ser calculado sobre um pedaço do sinal. Uma janela de tempo permite a inclusão de mais amostras e equação do erro fica:

$$e^2(t)_i = \sum_{t=j}^n (x(t) - y(t)_i)^2 \quad (5)$$

Na equação (5) pode-se ver a vantagem de trabalhar com erro quadrático, pois caso o erro comum for utilizado, pode-se perder ou alterar o resultado dependendo da direção do erro, um erro positivo é anulado por um erro negativo, assim pode não haver erro. O erro quadrático elimina a direção do erro e o somatório mostra o erro acumulado no período da janela de tempo.

O tamanho da janela de tempo influencia na convergência, poucas amostras resultam em uma precisão menor, no caso as janelas testadas são de ¼ de ciclo, ½ ciclo e 1 ciclo da frequência nominal de 60 Hz.

O próximo passo na aplicação do algoritmo genético é a seleção dos melhores indivíduos para posterior cruzamento. Esta seleção pode ser feita pelo método da roleta onde os indivíduos ocupam áreas inversamente proporcionais ao erro e o disco é girado por um método aleatório. Esse resultado aponta para um dos elementos da roleta, sendo o elemento o pai 1, e o pai 2 também será selecionado pelo mesmo método. Ambos os pais participam do cruzamento.

Na seleção por torneio 4 elementos são selecionados a partir da população e suas aptidões comparadas 2 a 2 onde, de cada grupo é selecionado um pai e estes participam do cruzamento.

O cruzamento é a geração da nova população que traz a herança genética dos pais, pois o processo combina as características selecionadas dos pais devido a suas aptidões.

No caso em questão, o cruzamento é realizado pela média simples entre os pais, gerando 3 filhos:

- a) um filho menor (fmn);
- b) um filho médio(fmd);
- c) um filho maior(fma)

Este cruzamento é ilustrado na fig.15.



Figura 15 Representação dos pais e filhos no processo de média simples.

O filho médio é gerado pela média segundo a equação 6:

$$fmd = \frac{pai1 + pai2}{2} \quad (6)$$

O filho menor é gerado pelo rebatimento da média no pai 1 conforme a equação 7:

$$fmn = pai1 - (fmd - pai1) \quad (7)$$

O filho maior é gerado pelo rebatimento da média no pai 2 conforme a equação 8:

$$fma = pai2 + (fmd - pai2) \quad (8)$$

Para o estudo, porém gerou-se 15 filhos para uma população de apenas 10 indivíduos, porém uma posição é reservada para o indivíduo elite, que é o elemento que mais se aproxima da solução do problema, portanto, nove posições são disponíveis e uma nova seleção é realizada e o método de seleção aleatório fornece os nove elementos entre os quinze indivíduos gerados.

Na sequência, os elementos gerados passam por um processo de mutação que é a alteração das características de forma proposital, isto é, adicionando-se ou retirando-se um valor constante ao valor do indivíduo sempre se preocupando em mantê-lo dentro do espaço de busca. Essa mutação ocorre segundo uma probabilidade que deve estar entre 1% e 10% e tem por finalidade evitar os mínimos locais que podem ocorrer no processo de convergência do algoritmo.

Após essa fase a nova população deve ser avaliada e todo o processo é repetido até que se encontre a solução, que é o elemento elite da última iteração.

Utilizando uma janela de  $\frac{1}{4}$  de ciclo, significando que 4 amostras do sinal de entrada serão utilizadas para serem comparadas com quatro amostras de cada um dos 10 indivíduos, gerando um erro que será elevado ao quadrado, proporcionando a avaliação da população inicial em relação ao sinal de entrada.

A população 10 indivíduos é mostrada na tabela 4 juntamente com o resultado da avaliação de cada indivíduo.

Tabela 4 População de 10 Indivíduos avaliados com a respectiva aptidão, janela  $\frac{1}{4}$  de ciclo.

| Indivíduo | Amplitude(Ae) | Frequência(fe) | Ang. Fase( $\theta_e$ ) | Aptidão |
|-----------|---------------|----------------|-------------------------|---------|
| 1         | 0.70          | 58             | 0                       | 0.19    |
| 2         | 0.77          | 58.44          | 0.69                    | 0.72    |
| 3         | 0.83          | 58.89          | 1.39                    | 2.63    |
| 4         | 0.90          | 59.33          | 2.09                    | 5.28    |
| 5         | 0.97          | 59.78          | 2.79                    | 7.47    |
| 6         | 1.03          | 60.22          | 3.49                    | 7.99    |
| 7         | 1,10          | 60.67          | 4.18                    | 6.45    |
| 8         | 1.17          | 61.11          | 4.88                    | 3.67    |
| 9         | 1.23          | 61.56          | 5.58                    | 1.13    |
| 10        | 1.30          | 62             | 6.28                    | 0.18    |

Nota-se que o indivíduo 10 obteve a melhor nota, pois o erro quadrático foi o menor, portanto, este indivíduo é selecionado como elite, com direito a participar da próxima população, sem passar pelo processo de seleção após a formação da nova geração.

O próximo passo é a seleção de 4 indivíduos aleatoriamente da população de 10 indivíduos; e estes indivíduos vão competir com suas notas para a seleção dos futuros pais. É interessante frisar que a escolha é aleatória, mas uma vez selecionado o indivíduo é confrontado com outro indivíduo e sua característica ou aptidão é determinante na escolha como pai.

Para o caso de uma população pequena pode acontecer de os dois pais selecionados serem os mesmos indivíduos, o que levaria à inexistência do cruzamento, um dos pais é substituído pelo elemento elite, que vence o torneio e é escolhido. Isto não acarreta problema, pois o elemento elite é o que mais se aproxima da solução do problema.



Uma vez selecionado os dois pais para o cruzamento suas características darão origem a um novo indivíduo, que terá os genes dos pais e a tendência a se adaptar é maior que a de seus pais.

O processo de cruzamento usando a média simples entre os dois pais, permite a geração de um filho entre os pais e, expandindo-se para explorar as vizinhanças, o processo produz 3 filhos ( filho menor (fmn), filho médio (fmd) e filho maior (fma)) sendo que estes indivíduos são sempre formados com as características de adaptabilidade dos pais.

O filho médio (fmd) é o resultado da média simples entre os dois pais, pela subtração da média e o pai menor encontra-se um valor chamado delta que é subtraído do pai menor, gerando o filho menor e, adicionado ao pai maior, gerando o filho maior. O processo de cruzamento acontece por um numero de vezes suficiente para se ter uma nova população, no caso será executado 5 vezes, gerando 15 novos elementos.

Como o elemento elite já é parte da nova população, 9 outros elementos devem ser selecionados para compor os 10 indivíduos da nova população, isto é feito por um processo aleatório.

Esta primeira população gerada não sofrerá mutação, que é uma alteração proposital visando evitar a estagnação do processo em mínimos locais. A mutação ocorre a partir da segunda geração em diante, segundo uma pequena probabilidade, menor que 10% e o processo de cruzamento ao contrário da mutação, com uma probabilidade alta, cerca de 90%. Este processo resulta na tabela 5.

Tabela 5 Resultado da primeira população gerada após a aplicação dos operadores genéticos: seleção, cruzamento e mutação.

| Indivíduo | Amplitude | Frequência | Ângulo de fase |
|-----------|-----------|------------|----------------|
| 1         | 1,30      | 62         | 6,28           |
| 2         | 1,30      | 62         | 6,28           |
| 3         | 1,20      | 61,33      | 5,23           |
| 4         | 0,97      | 59,78      | 2,79           |
| 5         | 1,30      | 62         | 6,28           |
| 6         | 1,07      | 60,44      | 3,83           |
| 7         | 1,20      | 61,33      | 5,23           |
| 8         | 0,70      | 58         | 0              |
| 9         | 1,03      | 60,22      | 3,49           |
| 10        | 1,30      | 62         | 6,28           |

O processo de convergência do algoritmo genético está diretamente ligado ao número de iterações ou ciclos de geração de populações, quanto maior o número de iterações mais o resultado se aproxima da solução. Para um pequeno número de iterações o erro é grosseiro e, portanto diminui com o aumento do número de iterações. Para o exemplo, os valores encontrados após 10 iterações para o caso 1 são mostrados na tabela 6.

Tabela 6 Resultados da execução do AG para o caso 1 : 10 indivíduos e 10 iterações

| Individuo      | Amplitude | Frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,89      | 58,90      | 0,27           | 0.05    |
| Erro           | 0,11      | 1,1        | 0,27           |         |
| Erro %         | 18,33     | 27,5       | 4,30           |         |

Na tabela 6 também será mostrado o erro percentual que é calculado segundo a equação 9

$$E\% = \frac{V_{esp} - V_{enc}}{V_{esp}} \cdot 100 \quad (9)$$

Onde:

$V_{esp}$  = valor esperado

$V_{enc}$  = valor encontrado

O gráfico das curvas do sinal adquirido e do sinal estimado mostra com maior clareza o resultado após 10 iterações, conforme a fig. 16.

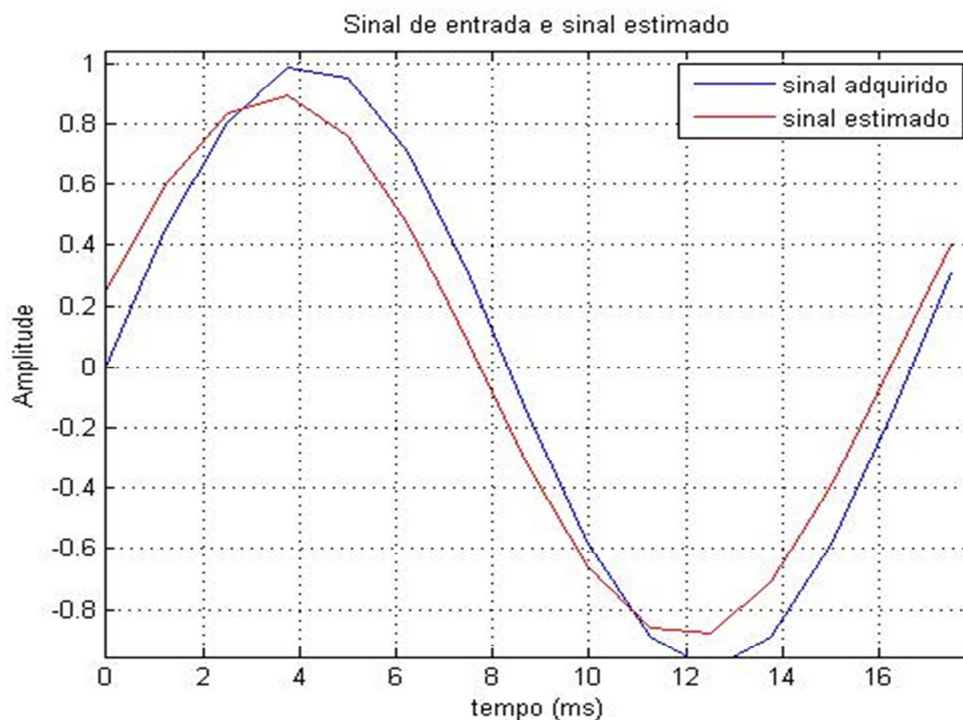


Figura 16 sinal adquirido e sinal estimado para uma população de 10 indivíduos e 10 iterações.

Para 100 iterações o algoritmo realiza 100 gerações de populações e a possibilidade de encontrar indivíduos melhor adaptados é maior, possibilitando um resultado mais preciso, isto é, o valor estimado é mais próximo do valor real. Os resultados são mostrados na tabela 7.

Tabela 7 Resultados encontrados após a execução do AG. por 100 iterações

| Indivíduo      | Amplitude | Frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,92      | 59,40      | 0,02           | 0.0007  |
| Erro           | 0,08      | 0,60       | 0,02           |         |
| Erro %         | 1,3       | 15         | 1              |         |

Após 100 iterações notamos uma convergência maior, isto é, o sinal estimado se aproxima mais do sinal real, conforme ilustrado na fig. 17.

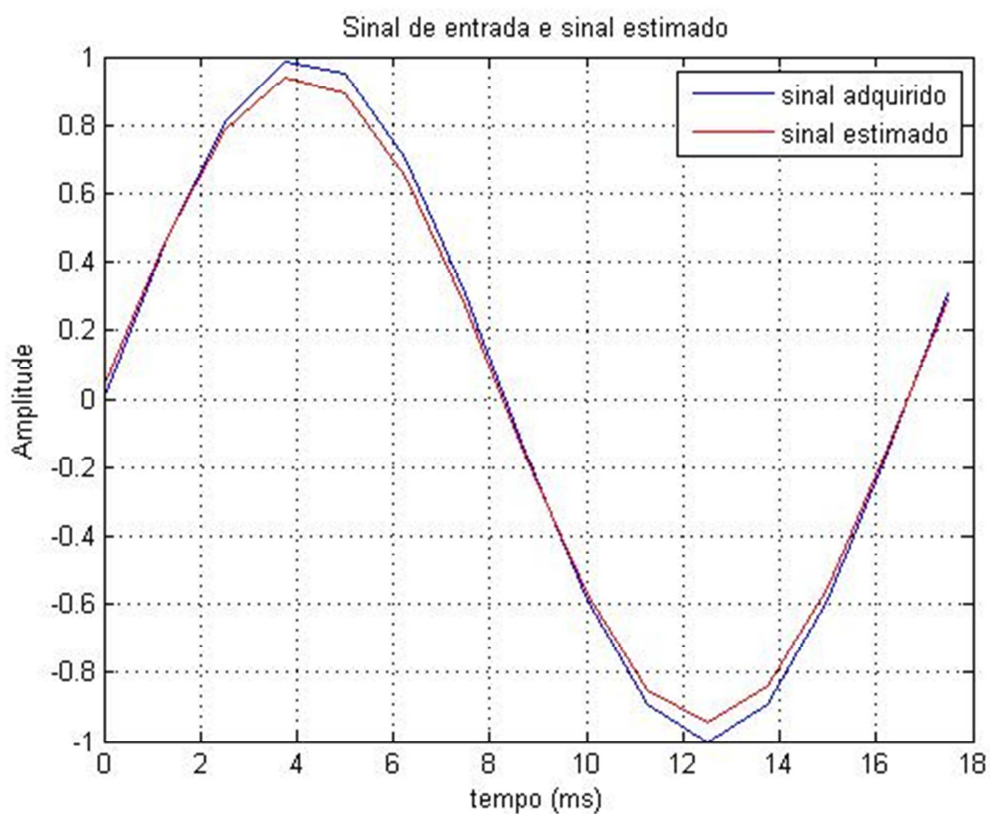


Figura 17 Sinal real e sinal estimado para 100 iterações

Após 1000 iterações o sinal estimado encontrado é muito próximo do sinal real e o erro é muito pequeno, muitas vezes o processo se estabiliza antes das 1000 iterações se

completarem, isto é, a melhor solução é encontrada em uma determinada iteração e após esta nenhuma outra apresenta resultado melhor. Os resultados estão exemplificados na tab.8.

Tabela 8 Resultados obtidos após a execução do AG por 1000 iterações.

| Individuo      | Amplitude | frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 1,01      | 60,01      | 0,01           | 0.0001  |
| Erro           | 0,01      | 0,01       | 0,01           |         |
| Erro %         | 0,47      | 0,2        | 0              |         |

O gráfico da resposta para 1000 iterações mostra uma precisão muito grande do sinal estimado em relação ao sinal real. A figura 18 mostra o sinal real e o sinal estimado após 1000 iterações.

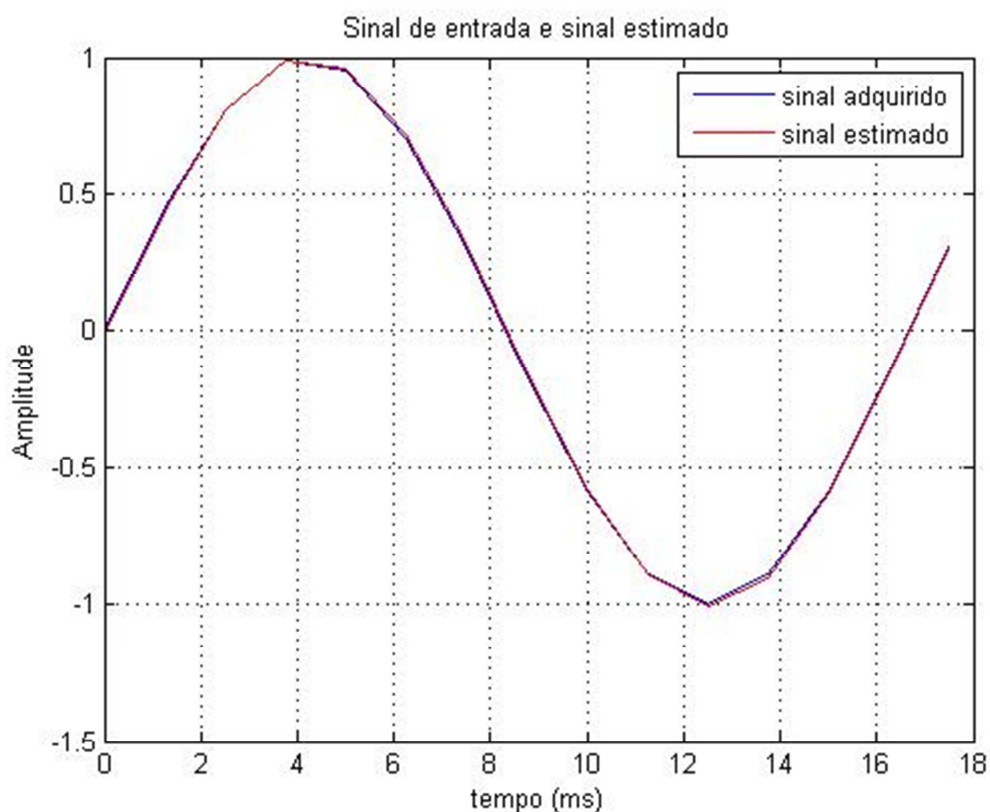


Figura 18 Sinal real e sinal estimado após 1000 iterações

Aumentando a janela de amostragem de  $\frac{1}{4}$  de ciclo para  $\frac{1}{2}$  ciclo a precisão aumenta, pois o número de amostras do sinal que é comparado possui mais características do sinal real, o que permite uma adaptação melhor do algoritmo aumentando a precisão da resposta.

A tabela 9 mostra os resultados obtidos para a janela de  $\frac{1}{2}$  ciclo, 10 indivíduos e 100 iterações.

Tabela 9 Resultados obtidos após a execução do AG para janela de  $\frac{1}{2}$  ciclo, 10 indivíduos e 100 iterações.

| Indivíduo      | Amplitude | Frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,98      | 59,84      | 0,12           | 0,005   |
| Erro           | 0,02      | 0,16       | 0,12           |         |
| Erro %         | 0,88      | 4          | 2              |         |

O sinal estimado se aproxima mais do sinal real, o erro que é a diferença entre o sinal real e o sinal estimado diminui, conforme pode ser verificado na fig. 19.

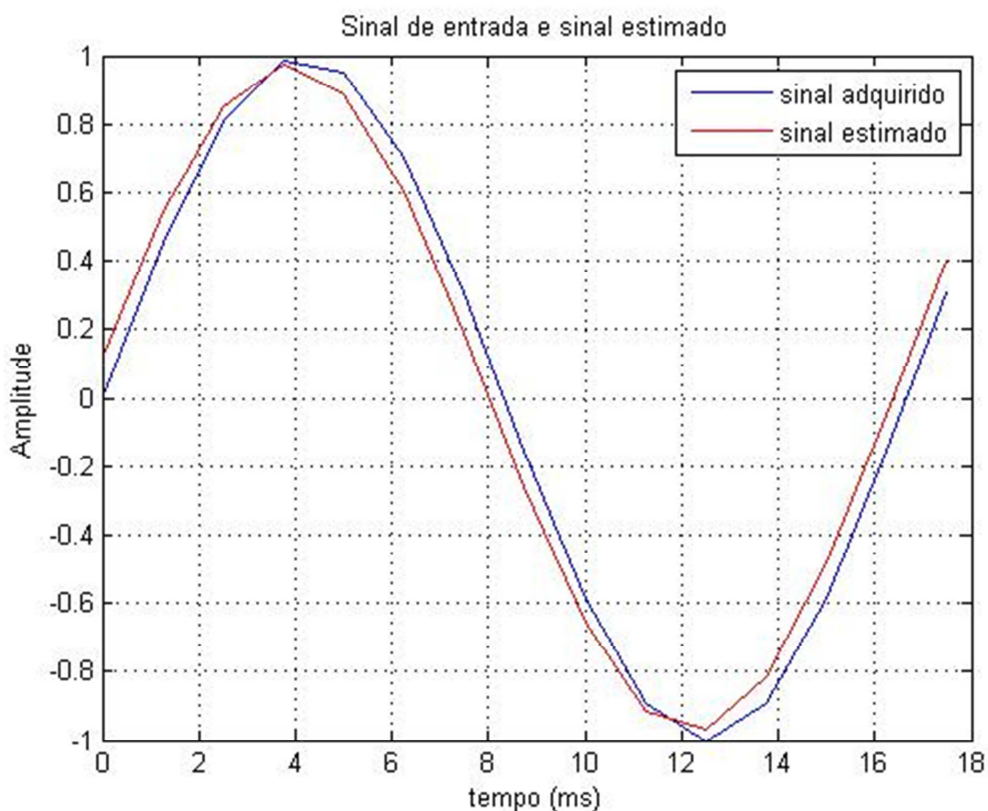


Figura 19 Resultados para 100 iterações e janela de  $\frac{1}{2}$  ciclo

Para uma janela de 1 ciclo conforme afirmado anteriormente a precisão é aumentada, pois mais características do sinal de entradas são usadas na avaliação dos indivíduos.

A tabela 10 mostra os resultados obtidos para a janela de 1 ciclo, 10 indivíduos e 100 iterações.

Tabela 10 Resultados obtidos após a execução do AG. para uma janela de 1 ciclo 10 indivíduos e 100 iterações.

| Indivíduo      | Amplitude | frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,98      | 59,86      | 0,08           | 0,0038  |
| Erro           | 0,02      | 0,14       | 0,08           |         |
| Erro %         | 0,88      | 3,49       | 1,32           |         |

O sinal real e o sinal estimado produzem um erro muito menor que o erro de uma janela de  $\frac{1}{4}$  de ciclo mostrado na tab. 7.

A fig. 20 mostra o resultado para uma população de 10 indivíduos obtidos na tabela 10.

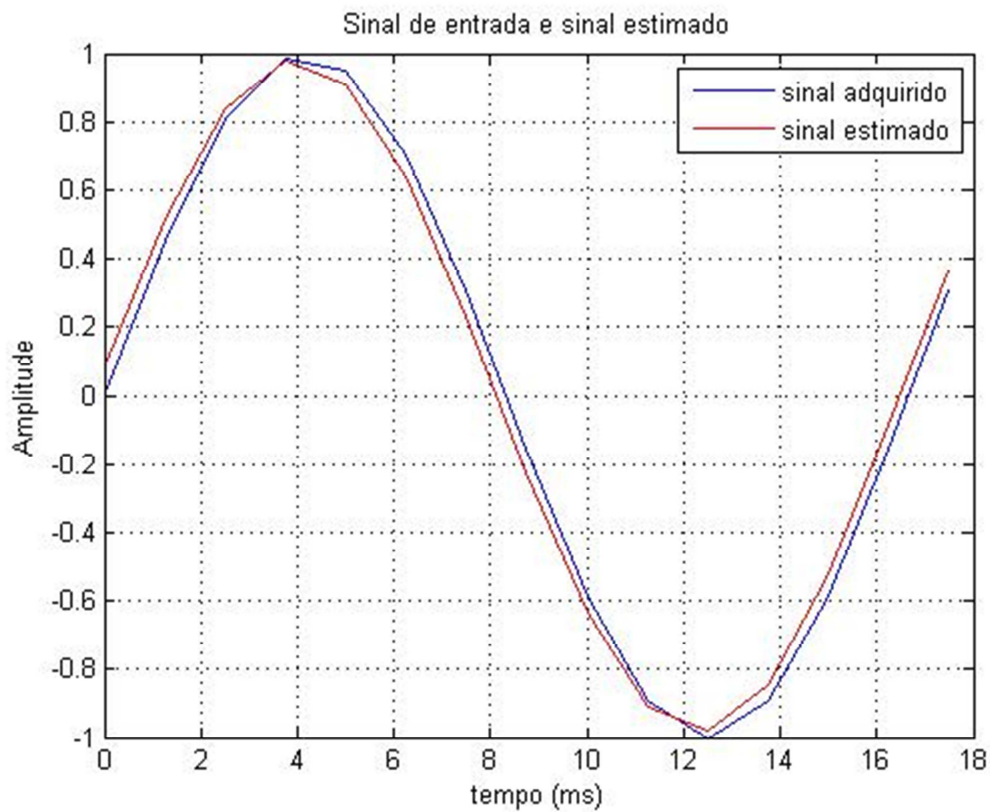


Figura 20 Resultado para janela de 1 ciclo

Note que a janela maior permite uma maior precisão, significando que mais amostras são usadas no processo de avaliação e a complexidade do processo aumenta mas nos leva a uma precisão maior.

#### 4.1.2 Aplicação de AG com população de 30 indivíduos

O tamanho da população está diretamente relacionado com o tempo de processamento, por um lado melhora a precisão por outro aumenta o tempo de processamento, pois as fases de avaliação, seleção, cruzamento e mutação que são realizadas para cada indivíduo aumenta em função do aumento de indivíduos em cada fase, aumentando o número de operações em cada iteração, conseqüentemente aumenta o tempo de processamento geral do AG.

O numero ideal de indivíduos de uma população é um número empírico que deve ser estimado pelo programador do AG, sendo que cada problema tem as suas características próprias e conseqüentemente tem o números de indivíduos próprio para este problema que varia para outro problema.

Os resultados para o segundo caso para uma população de 30 indivíduos são mostrados na tab. 11.

Tabela 11 Resultados obtidos após a execução do AG. para uma população de 30 indivíduos por 10 iterações.

| Individuo      | Amplitude | frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,98      | 59,71      | 0,08           | 0,0038  |
| Erro           | 0,02      | 0,29       | 0,08           |         |
| Erro %         | 0,88      | 7,19       | 1,32           |         |

O gráfico do resultado da fig. 21 mostra que o sinal estimado é bem mais do preciso que com a população de 10 indivíduos em 10 iterações.

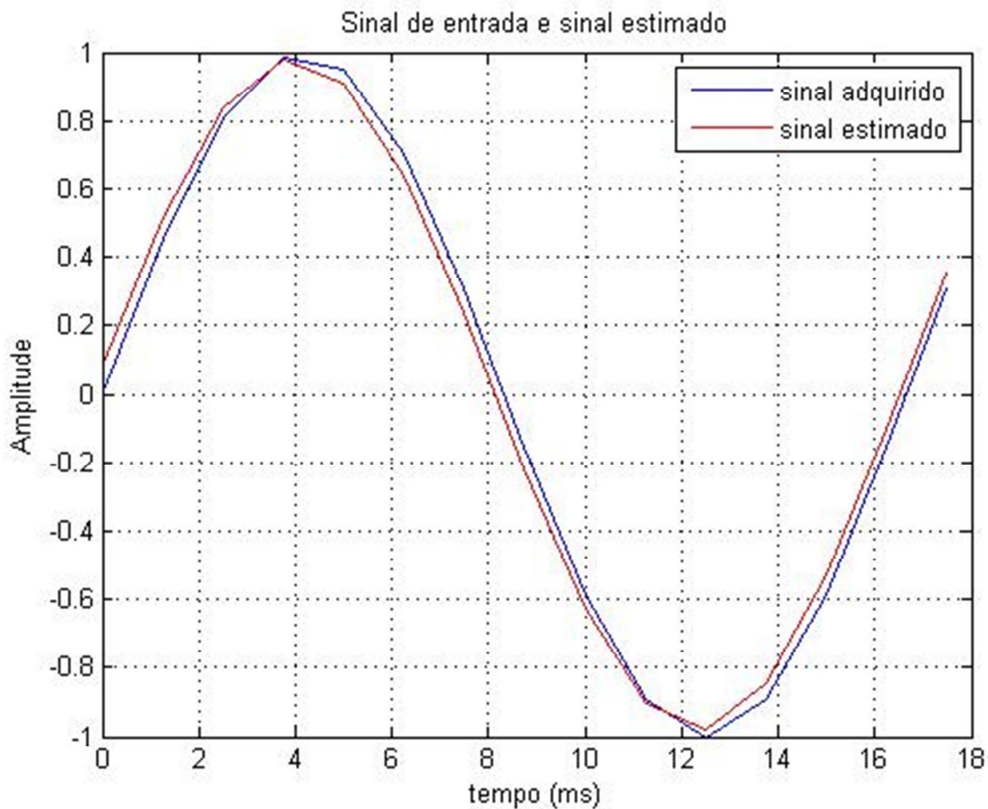


Figura 21 Resultado sinal real e sinal estimado para população de 30 indivíduos após 10 iterações.

A execução do AG com um número maior de iterações proporciona um resultado mais aproximado entre o sinal real e o sinal estimado, conforme mostra a tab. 12

Tabela 12 Resultados obtidos após a execução do AG. para uma população de 30 indivíduos por 100 iterações.

| Indivíduo      | Amplitude | Frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,99      | 59,75      | 0,00           | 0,0003  |
| Erro           | 0,01      | 0,25       | 0,00           |         |
| Erro %         | 0,5       | 6.35       | 0              |         |

O gráfico do sinal real e do sinal estimado mostrado na fig.22 ilustra uma precisão bem melhor em comparação com a população de 10 indivíduos, para o mesmo número de iterações, no caso 100 iterações.



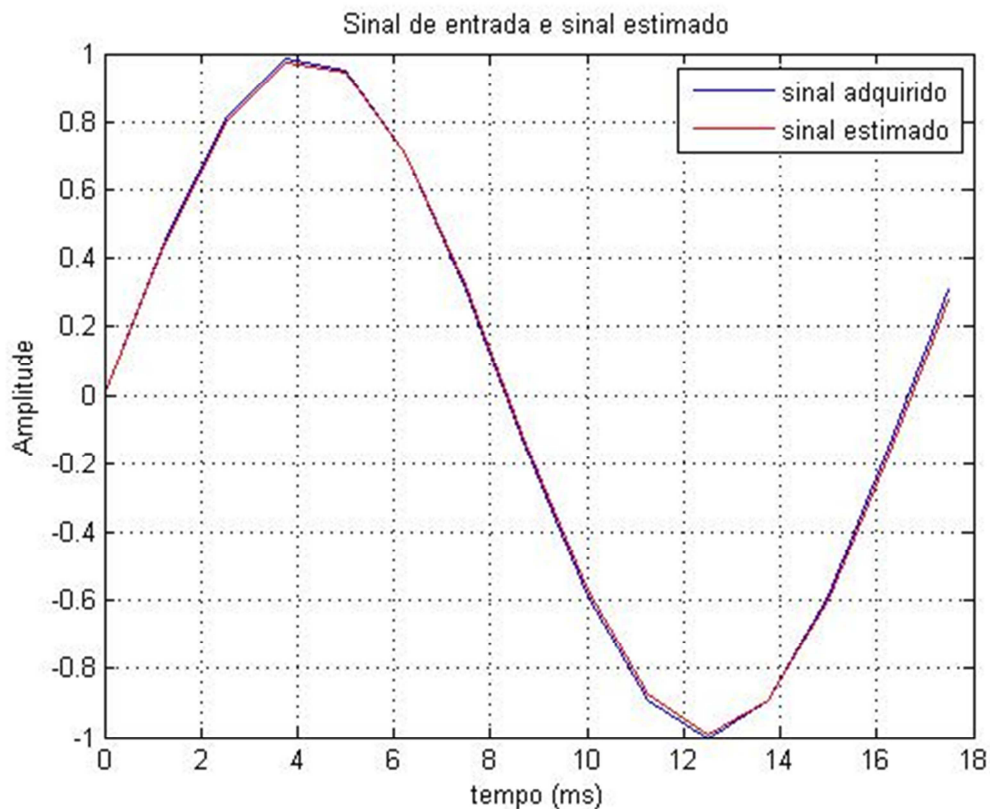


Figura 22 Resultados sinal real e sinal estimado para população de 30 indivíduos 100 iterações

Aumentando mais o número de iterações, para 1000 iterações, a aproximação se torna ainda maior, e o resultado mais próximo do sinal de entrada, conforme observado na Tab. 13.

Tabela 13 Resultados obtidos após a execução do AG. para uma população de 30 indivíduos por 1000 iterações.

| Indivíduo      | Amplitude | Frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,99      | 59,93      | 0              | 0,0001  |
| Erro           | 0,01      | 0,07       | 0              |         |
| Erro %         | 0,5       | 0,72       | 0              |         |

O gráfico para população de 30 indivíduos, mostrado na fig. 23 permite constatar uma precisão muito boa, pois o erro é muito pequeno.

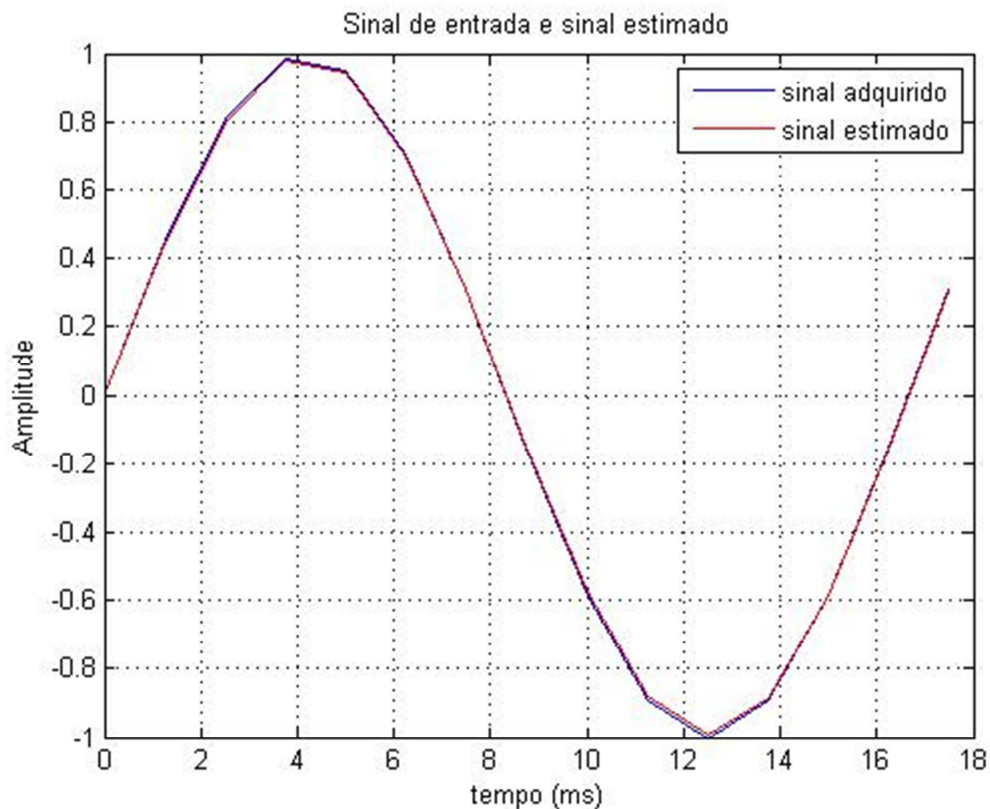


Figura 23 Resultado Sinal real e sinal estimado para uma população de 30 indivíduos e 1000 iterações.

O sinal estimado está praticamente idêntico ao sinal adquirido provando que os Ags são um excelentes estimadores de sinais periódicos.

### 4.1.3 Aplicação de AG com população de 50 indivíduos

O último caso analisado é o caso de uma população grande com um número considerado mais que o suficiente para a solução do problema, conforme foi verificado em várias tentativas em laboratório.

O número de iterações a ser considerado para esta análise é de 1000 iterações, que nos permitirá uma maior definição das vantagens e desvantagens de cada um dos casos. A Tab. 14 mostra os resultados

Tabela 14 Resultados obtidos após a execução do AG, para uma população de 50 indivíduos por 1000 iterações.

| Individuo      | Amplitude | Frequência | Ângulo de fase | Aptidão |
|----------------|-----------|------------|----------------|---------|
| Sinal Real     | 1         | 60         | 0              |         |
| Sinal estimado | 0,99      | 59,99      | 0              | 0,00    |
| Erro           | 0,01      | 0,01       | 0              |         |
| Erro %         | 0,5       | 0,3        | 0              |         |

A fig. 24 mostra os resultados para uma população de 50 indivíduos e 1000 iterações, neste estudo uma grande precisão entre o sinal real e sinal estimados foi obtida.

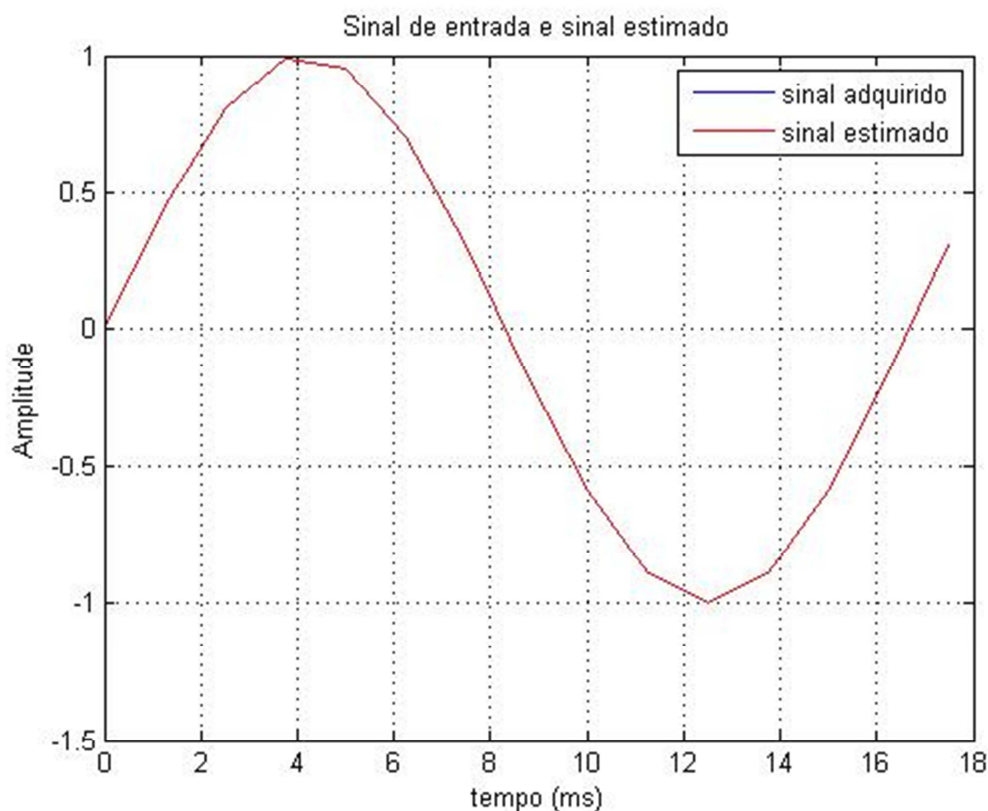


Figura 24 Resultados para uma população de 50 indivíduos 1000 iterações.

A fig. 24 mostra que foi atingida uma excelente precisão, mas analisando em função do tempo de processamento do algoritmo, o tempo de execução do AG para uma população de 10 indivíduos é muito menor que o tempo de processamento de um AG para uma população de 30 indivíduos que por sua vez é menor que o o tempo de execução de um AG para uma população de 50 indivíduos, isto se deve ao fato que cada indivíduo deve ser avaliado, participar da seleção, do cruzamento e da mutação. O aumento no número de indivíduos provoca o aumento no tempo de execução do algoritmo.

Outro parâmetro que **pode** melhorar o desempenho do algoritmo genético é a mutação. No caso de multivariáveis a mutação deve ser coerente com o espaço de busca de cada variável, 0,1 para a amplitude, 0,3 para a frequência e 0,5 para o ângulo de fase permitem atingir o resultado em um número menor de iterações. Estes valores foram

experimentados em função de suas proporcionalidades em relação a faixa de variação (range) de cada variável.

Os resultados para estas alterações podem acontecer antes do número final de iterações programado, a atividade do algoritmo cessa no momento que o elemento elite é a solução do problema, pois uma vez encontrada a solução não é possível evoluir mais.

O critério de parada é um dos problemas do AG, comumente são adotados 3 critérios de parada:

- a) número de iterações ;
- b) a imutabilidade do indivíduo elite;
- c) a tolerância.

O número de iterações pode ser usado como critério de parada, mas deve-se tomar cuidado para não superestimar ou subestimar o número de iterações necessárias para a convergência do AG, isto significa que, para que o elemento elite seja uma resposta aceitável para o problema é necessário um tempo computacional, e se este tempo for maior do que o tempo necessário há desperdício de tempo computacional, nem sempre aceitável, devido a condições como aplicações em tempo real, etc.

A imutabilidade do elemento elite como critério de parada é um critério que se baseia na alternância do elemento, visto que quando houver a convergência esse elemento não se altera mais, pois a solução foi encontrada, sendo também um critério que requer certo cuidado, pois terá que ser suficiente para garantir que a convergência é máxima.

O critério de convergência que usa a tolerância ou que estabelece o erro aceitável, é o critério que permite a parada imediata após atingir o valor desejado de erro ou nota máxima do elemento evitando o processamento de confirmação dos critérios anteriores que envolve um desperdício de tempo computacional.

A preocupação com o tempo de execução do AG tem fundamentos para aplicações em tempo real, onde as respostas devem acompanhar as entradas em termos de tempo de processamento SOUZA (2008) apresentou um estudo sobre a utilização de FPGA para aplicações em tempo real devido a capacidade de processamento paralelo das células dos FPGAs, e da natureza paralela do AG, que combinadas propiciaram uma alta eficiência à proteção de linha de transmissão do sistema real.

O programa utilizado no MatLab® para a execução dos AGs para sinais periódicos encontra-se no anexo B.

## 4.2 Aplicação da Transformada de Fourier

Até o presente momento a aplicação do AG foi para análise de um sinal repetitivo no tempo, isto é, um sinal periódico, para a qual o AG apresentou um bom desempenho na estimação do sinal. Para efeito de comparação deste desempenho a análise do mesmo sinal de entrada foi feito um estudo por um método tradicional, no caso, a Transformada de Fourier que é frequentemente utilizada para a estimação de parâmetros de sinais periódicos, principalmente usada para a estimação da frequência com o uso do matlab®.

Como o sinal de entrada é um sinal discreto a transformada de Fourier aplicada é a Transformada Discreta de Fourier, TDF, que é eficiente na análise de sinais amostrados, sendo inclusive base para vários equipamentos de proteção do sistema elétrico em função de desvios de frequência, como os relés de frequência.

A TDF possui uma implementação bastante simples e apresenta resultados consideravelmente eficientes. Para o sinal de entrada  $x(t)$  aquisitado pela placa NI-DAQ™mx USB, amostrado a uma taxa de 1/800 para o caso de 50 amostras, o seguinte resultado encontrado é mostrado na fig. 25.

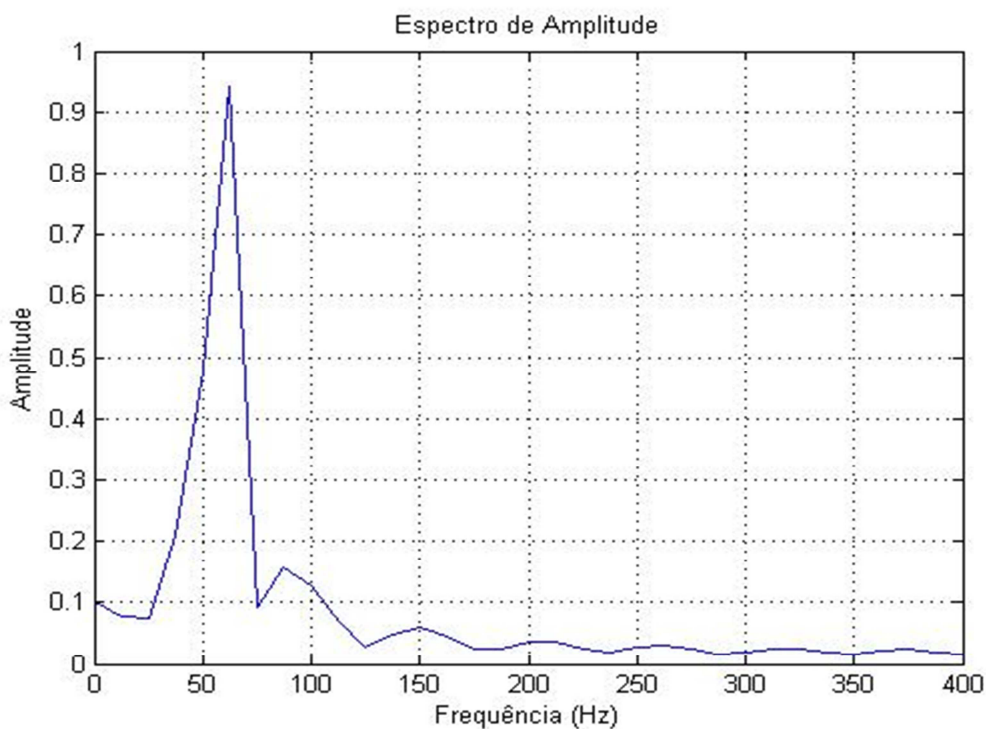


Figura 25 Resultados da FFT para 50 amostras do sinal

A precisão é comprometida, mas é possível perceber que a frequência foi estimada corretamente.

Aumentando-se o número de amostras, aumenta-se a qualidade da resposta obtida, a que se levar em conta que o sinal é uma senoide pura e que os sinais no mundo real possuem harmônicos e ruídos, presentes, portanto, na resposta da DFT. A estimativa da amplitude é prejudicada em função do próprio processo de cálculo da DFT.

A qualidade da resposta no processo deixa a desejar, pois não permite determinar a frequência com precisão, portanto aumentou-se o número de amostras para 100 amostras, a precisão melhorou, mas ainda é baixa, os seguintes resultados são apresentados na fig. 26.

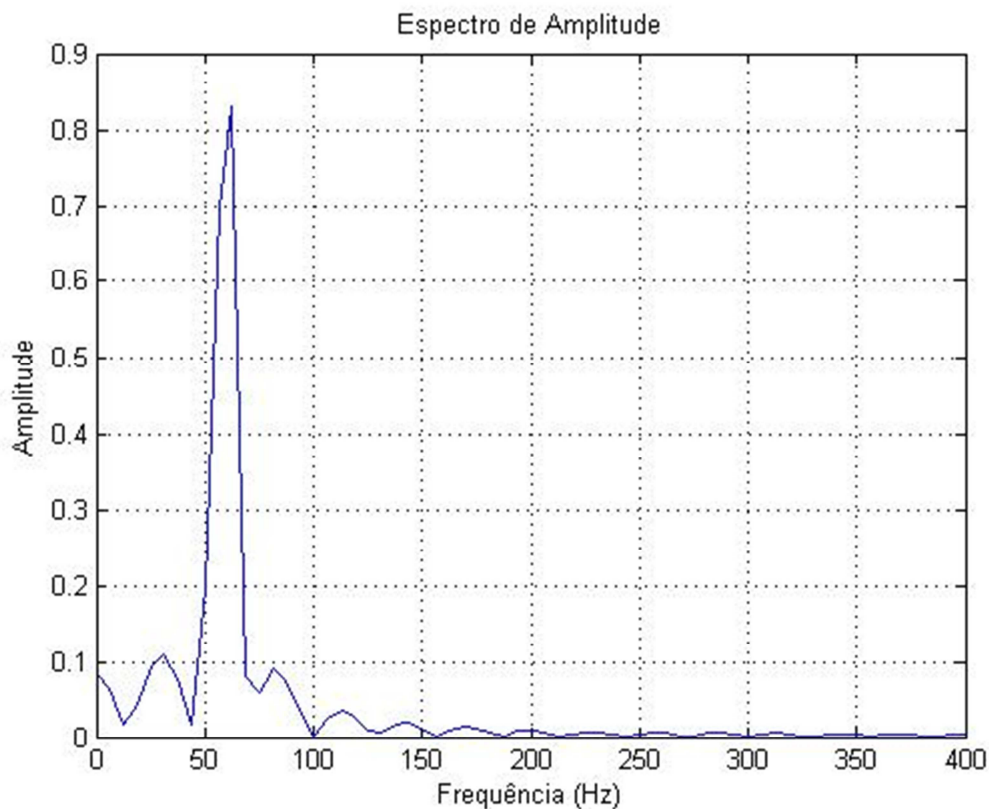


Figura 26 FFT com 100 amostras do sinal de entrada

Os resultados começam a ser mais precisos com 1000 amostras, conforme apresentado na fig. 27.

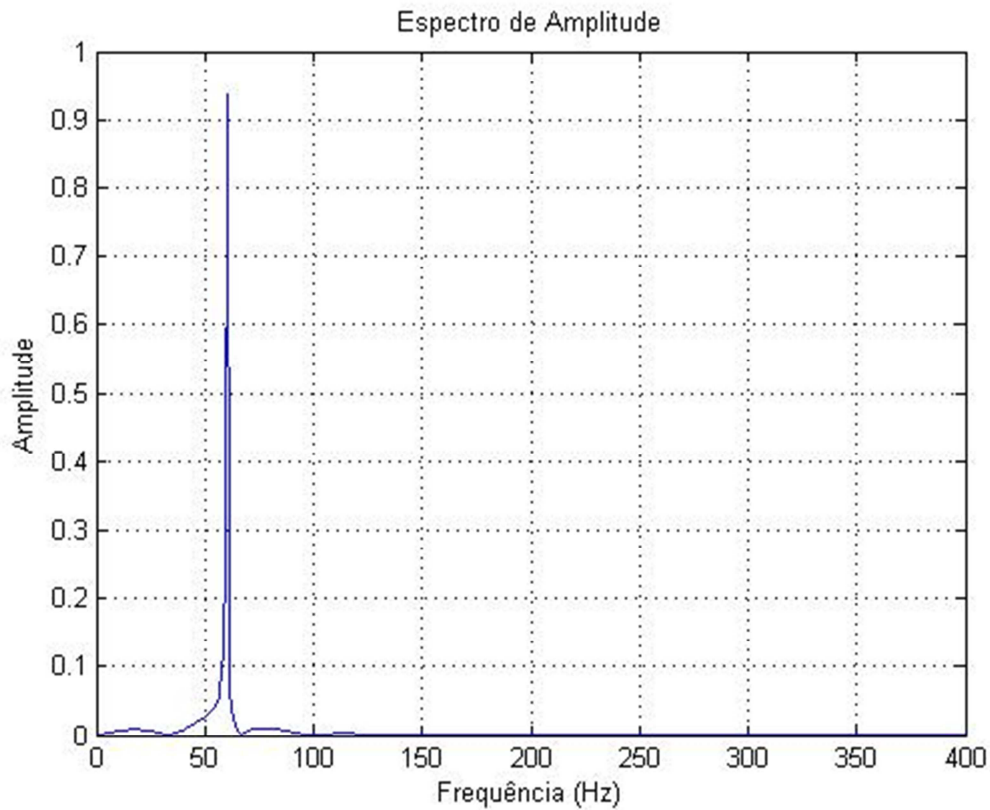


Figura 27 FFT com 1000 amostra do sinal de entrada.

A partir de 1000 amostra tem-se uma boa precisão, é possível verificar o resultado 60 Hz, mas cabe ressaltar que o processo torna-se lento, pois o tempo computacional aumenta proporcionalmente com número de amostras.

### 4.3 Sinais aperiódicos e caóticos

Os sinais nem sempre são comportados e repetitivos, o mundo real está repleto de sinais de vão desde um simples degrau unitário a uma série caótica, passando por sinais de toda a espécie.

Para o caso de sinais aperiódicos, sinais não repetitivos no tempo, as transformadas de Fourier não mais são aplicáveis, mas o AG consegue resolver.

Todo o desempenho do AG está associado a escolha de uma função fitness coerente com o problema.

Liden (2000) relata que o grande problema do AG é a representação do problema no AG, dos indivíduos e seus genes. Na busca de uma fitness que represente o sinal aperiódico vamos estudar as séries temporais de Mackey-Glass.

As séries temporais de Mackey-Glass são produzidas por uma equação diferencial com atrasos de tempo ( $\Delta$ ) (MACKEY; GLASS, 1977) conforme equação 9

$$\frac{dy(t)}{dt} = \beta y(t) + \frac{\alpha y(t-\Delta)}{(1+y(t-\Delta))^{10}} \quad (9)$$

A série de Mackey-Glass é gerada a partir da discretização da Equação (9), usando o método de Euler.

Para um sinal  $y(t)$  a equação da série fica:

$$y(t) = \beta y(t-1) + \frac{\alpha y(t-\tau)}{(1+y(t-\tau))^{10}} \quad (10)$$

Para os valores de  $\beta=0,2$ ,  $\alpha=0,9$  e  $\tau=15$ , usando Matlab®, gera-se a fig. 28.

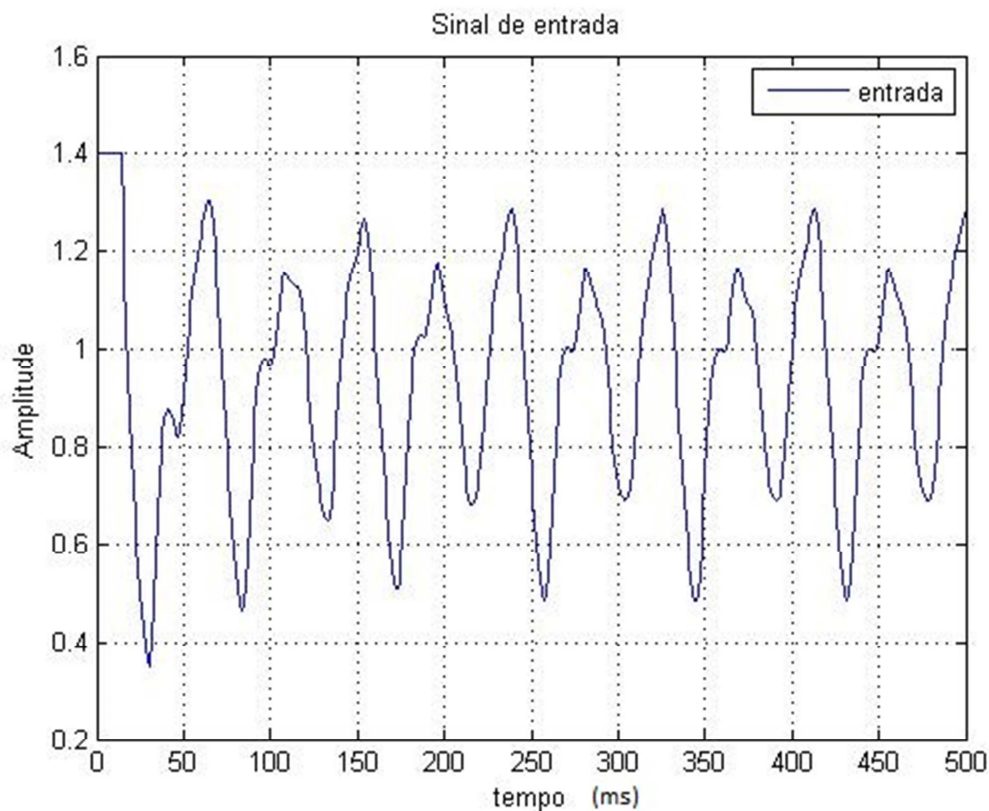


Figura 28 sinal  $y(t)$  gerado pela equação de Mackey-Glass



O Algoritmo genético consegue identificar os parâmetros geradores da série e reproduzi-la com exatidão, lembrando que o critério de parada é que determina a precisão, portanto quanto maior o número de iterações, mais indivíduos são gerados e tendem a ser solução para o problema.

Como o critério é de 100 iterações o sinal estimado se aproxima do sinal de entrada, mas sua precisão é de baixa qualidade, pois o algoritmo ainda não gerou número suficiente de indivíduos para que haja a convergência, portanto o erro é grosseiro e fica evidenciado tanto na tabela como na figura.

Tabela 15 Resultados obtidos após a execução do AG. por 100 iterações.

|                  | A    | $\beta$ | T  |
|------------------|------|---------|----|
| Sinal de entrada | 0,9  | 0,2     | 15 |
| Sinal de saída   | 0.88 | 0,24    | 15 |
| Erro             | 0,02 | 0,04    | 0  |
| Erro %           | 2    | 4       | 0  |

Os resultados do processamento do AG para 100 iterações são mostrados na fig. 29.

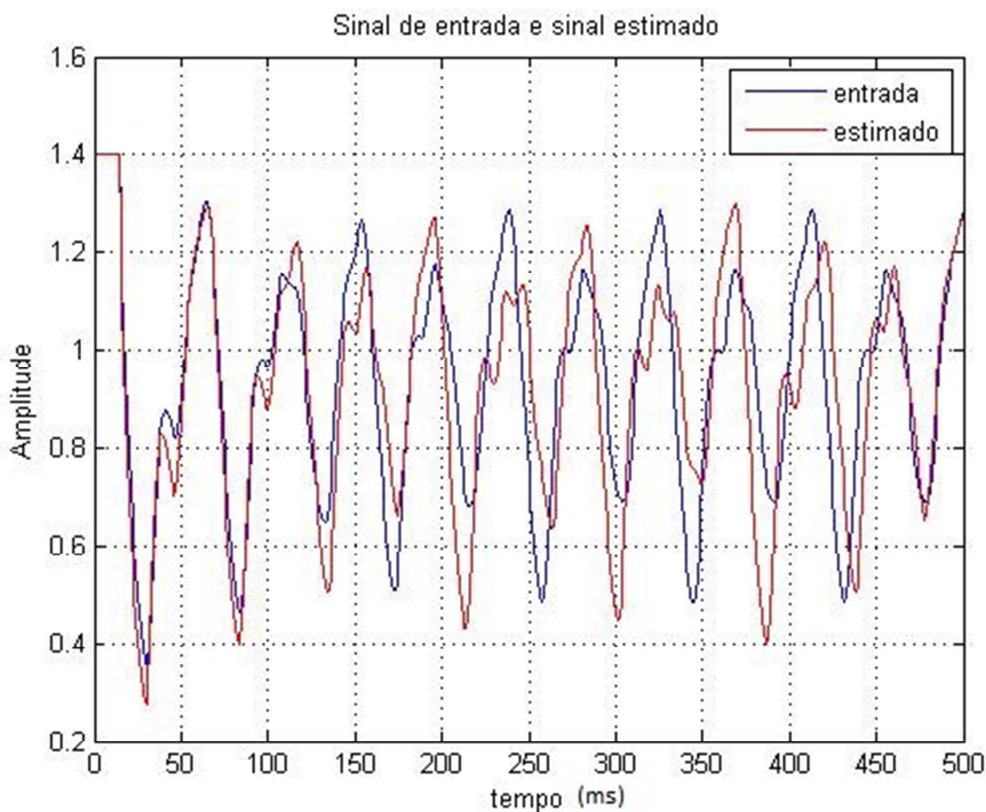


Figura 29 Sinal de entrada e sinal estimado obtido pelo AG após 100 iterações.

O resultado não apresentou uma precisão razoável, mas próximo da solução do problema, ampliando o numero de iterações para 500 iterações, os resultados são apresentados na tabela 16.

Tabela 16 Resultados obtidos após a execução do AG. por 500 iterações.

|                  | A    | B    | $\tau$ |
|------------------|------|------|--------|
| Sinal de entrada | 0,9  | 0,2  | 15     |
| Sinal de saída   | 0.89 | 0,21 | 15     |
| Erro             | 0,01 | 0,01 | 0      |
| Erro %           | 1    | 1,5  | 0      |

A análise da fig. 30 composta dos resultados obtidos pelo AG após 500 iterações, permite verificar uma convergência mais apropriada:

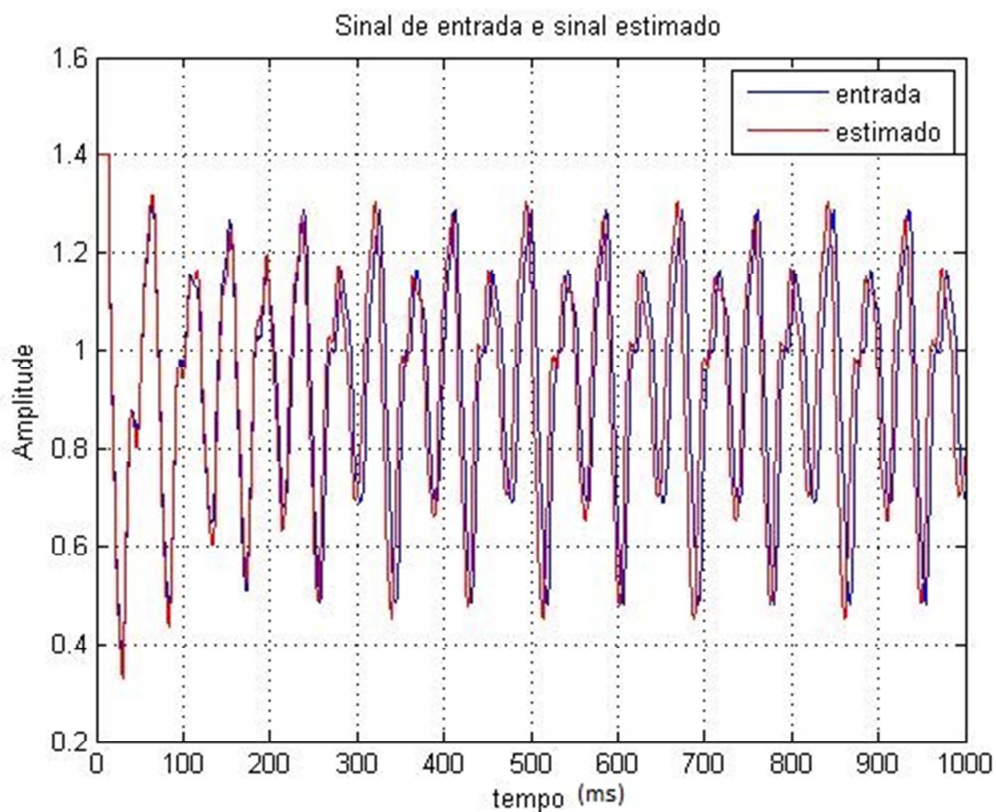


Figura 30 Sinal obtido após 500 iterações.

Ampliando ainda mais o número de iterações do AG, obtém-se resultados mais precisos como mostrados na tabela 17, esta noção de precisão esta ligada a redução do erro,

significando que os parâmetros encontrados permitem a construção de um sinal muito próximo do sinal adquirido processado pelo AG.

Tabela 17 Resultados obtidos após a execução do AG para 1000 iterações.

|                  | A     | B     | T  |
|------------------|-------|-------|----|
| Sinal de entrada | 0,9   | 0,2   | 15 |
| Sinal de saída   | 0.899 | 0,201 | 15 |
| Erro             | 0,001 | 0,001 | 0  |
| Erro %           | 0,1   | 0,1   | 0  |

O aumento no número de iterações aumenta a precisão, o que é mostrado na fig. 31 que apresenta os resultados da execução do AG para 1000 iterações:

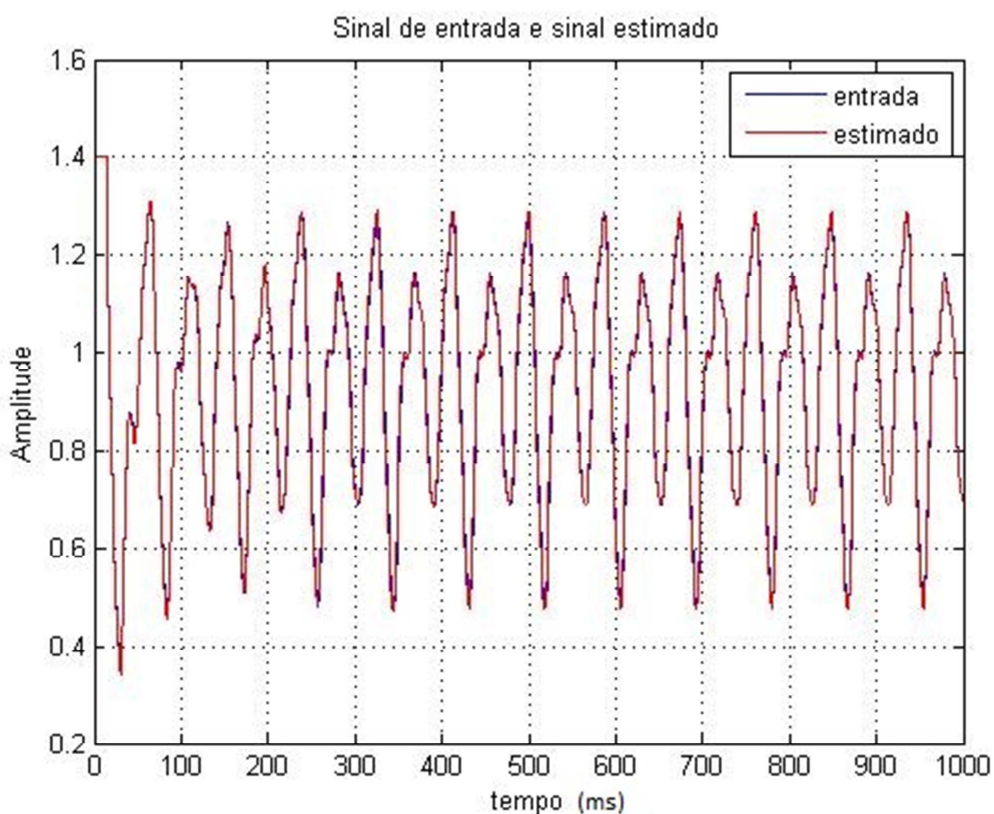


Figura 31 Resultados para execução do AG após 1000 iterações.

O algoritmo genético foi utilizado na modelagem de sinais de controle como o sinal abaixo, Fig. 32, o qual seria impossível de estimar pelos processos usando a TF. Provando que o AG é uma ferramenta capaz de estimar qualquer sinal do mundo real, como este, que é um sinal de controle de um regulador de tensão de um gerador durante

um “carregamento” do sistema elétrico, em outras palavras, durante o uso do gerador para alimentar uma carga resistiva.

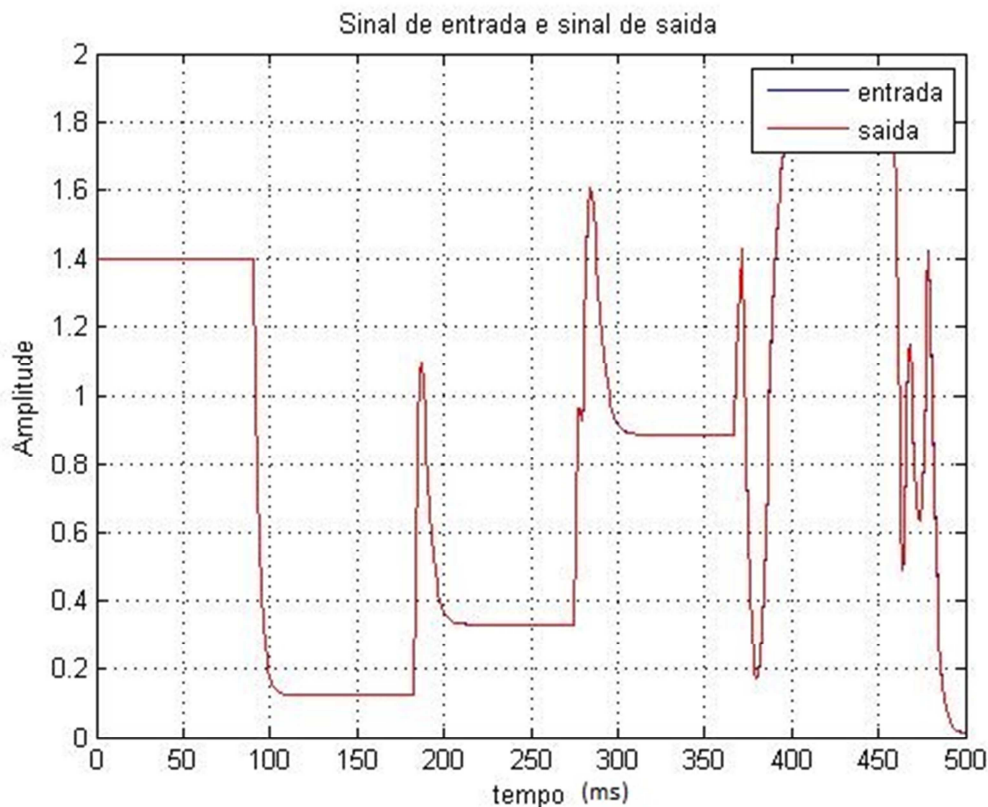


Figura 32 Sinal de controle para um regulador de tensão de um gerador durante processo de “carregamento” do sistema elétrico

A aquisição do sinal de controle e sua repetibilidade a partir do sinal estimado possibilita o estudo da atuação do controlador, dos tempos limites e valores das atuações dentre outros parâmetros possíveis de se analisar.

Se um sinal pode ser repetido ele pode ser estudado, analisado e simulado, pois seus parâmetros de formação foram descobertos e permitem a sua recuperação a qualquer instante.

A ferramenta AG neste estudo busca um sinal qualquer, do mundo real, sem as características de periodicidades que impõe restrições aos métodos de análise clássicos e mostra resultados precisos.

A fig. 33 mostra a placa do controlador de onde foi capturado o sinal utilizando o sistema de aquisição de dados da National Instruments NI USB-6009 DAQmx® e o sistema de aquisição de dados utilizado neste trabalho.

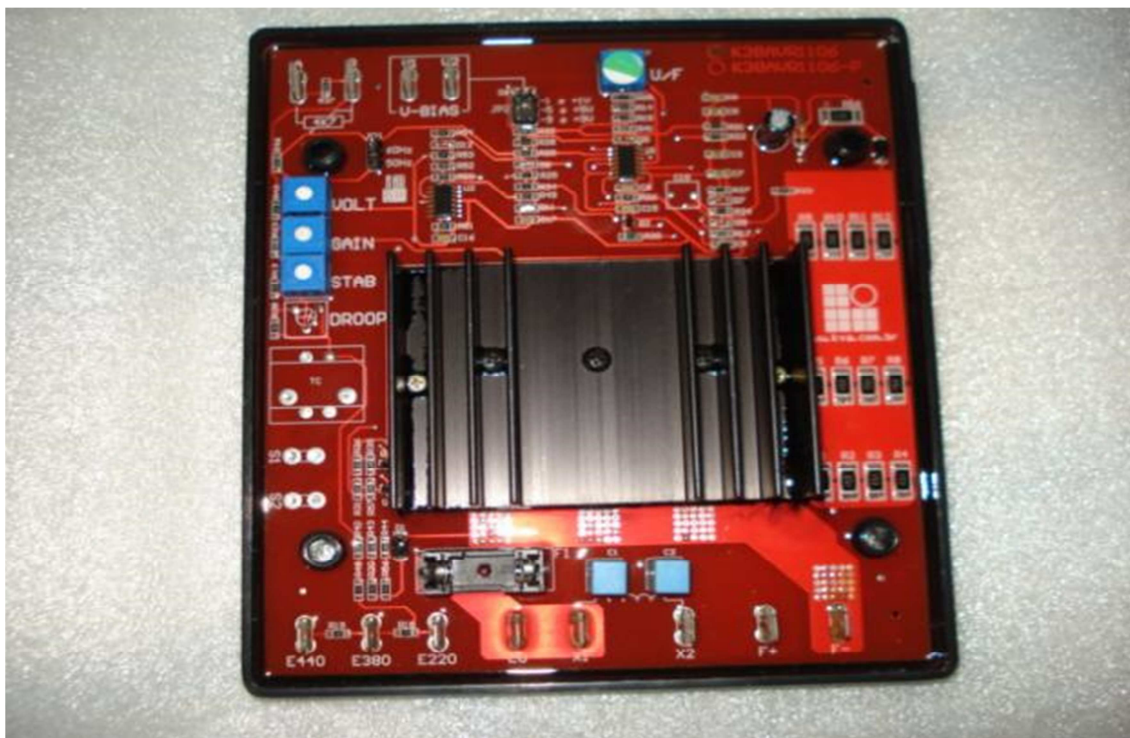


Figura 33 Regulador de tensão de gerador K38 AVR

Outra aquisição onde é perfeitamente visível o adaptabilidade do controle a situação de carregamento do sistema elétrico, mostrado na figura 34.

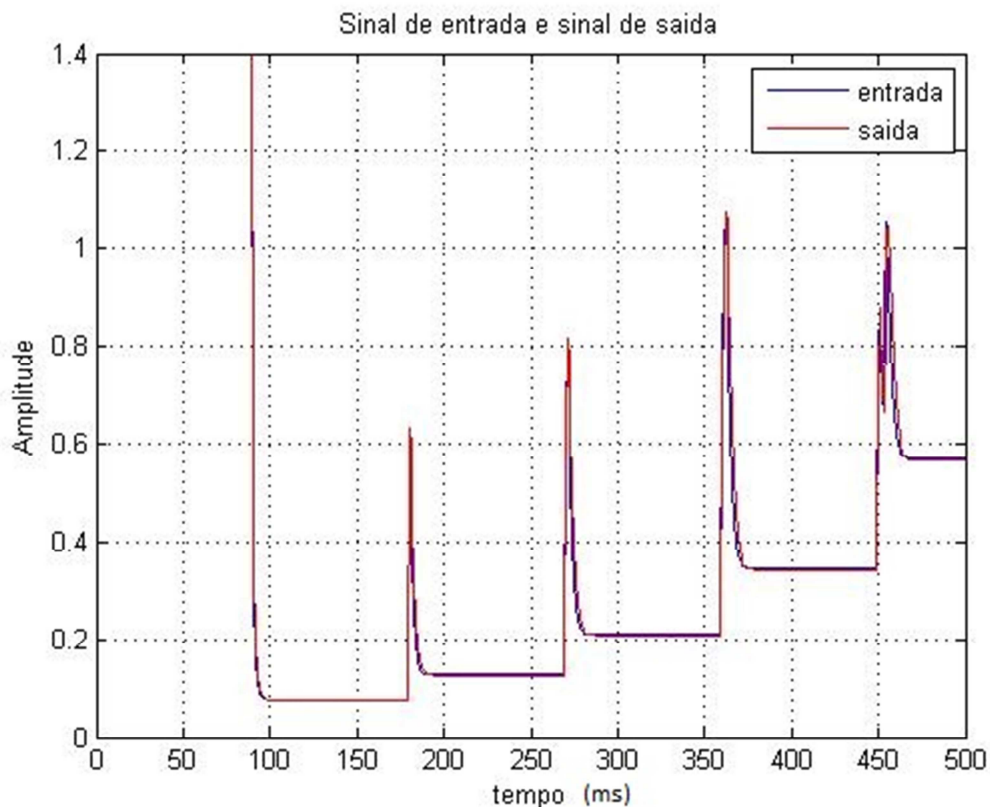


Figura 34 Resultado do GA para estimação do sinal de controle de regulador de tensão.

Os Ag. Fornece os resultados que permitem os estudos sobre os tempos de adaptação de sinais de controle do regulador de tensão durante as diversas situações de carregamento do sistema elétrico alimentado pelo gerador. Outro sinal, importante nos estudo dos sinais, que pode ser estimado pelo algoritmo genético é a função degrau. Pode-se analisar detalhes como o tempo de subida, inclinação, entre outros parâmetros.

A fig. 35 nos mostra um sinal de comutação entre níveis TTL em que o sinal é representado por um degrau, é possível verificar, por exemplo, o tempo de subida, ou seja, o tempo de comutação entre o estado baixo e o estado alto, este tempo tem importância fundamental na eletrônica digital, pois define a compatibilidade de componentes eletrônicos das diversas famílias de chips, que com suas características podem ou não fazerem parte do circuito.

A compatibilidade dos componentes eletrônicos repercute diretamente no funcionamento correto dos circuitos, assim um elemento incompatível em termos da velocidade de comutação pode levar todo o circuito a ter um comportamento indevido e prejudicar o desempenho do circuito.

A fig. 35 mostra o sinal de comutação de um nível baixo para o nível alto.

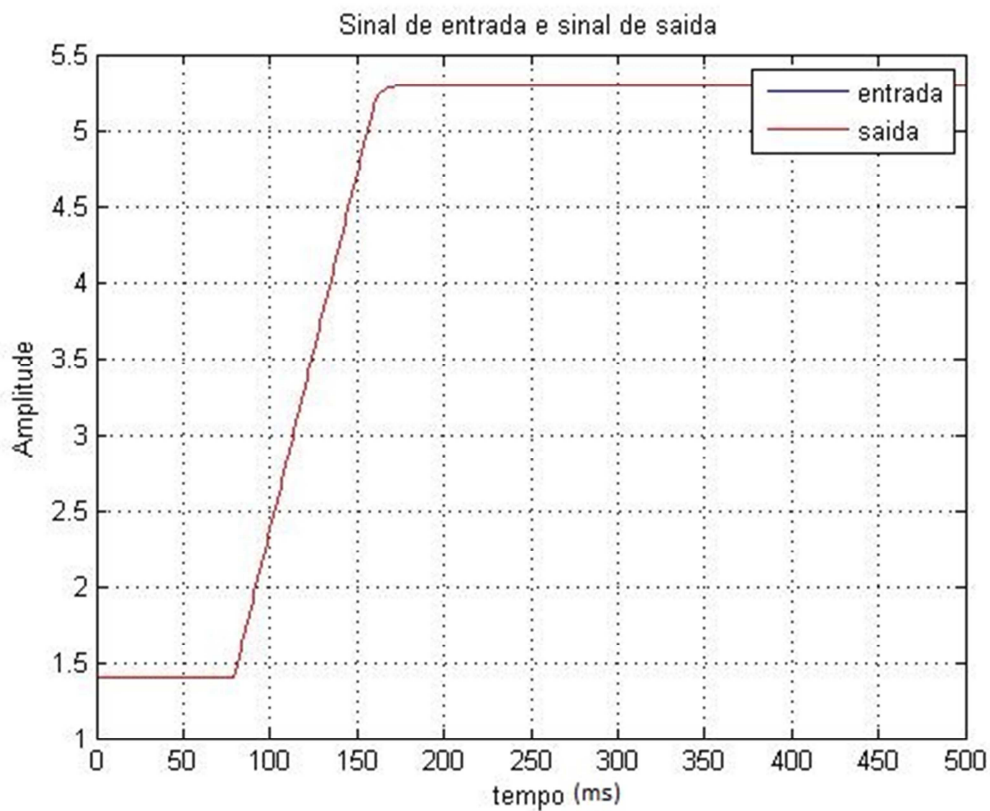


Figura 35 Degrau

A precisão da saída em relação à entrada se deve a uma fitness perfeitamente ajustada ao problema em uma estrutura de fácil implementação e baixo esforço.

O anexo A apresenta o programa utilizado no MatLab® para a execução do AG para sinais aperiódicos e caóticos.

## 4.4 Considerações Finais

Avaliando a aplicação da ferramenta algoritmo genético no estudo de sinais é possível verificar que se trata de uma ferramenta robusta em função dos resultados obtidos, atende as necessidades para os sinais periódicos visto que, em poucas iterações consegue chegar ao valor exato de um sinal periódico como uma senoide, adquirida por um sistema de aquisição de dados como o apresentado neste capítulo.

Em comparação com os métodos clássicos que utilizam a TF para detecção destes mesmos parâmetros os AGs, apresentam vantagens sensíveis, pois sua precisão é muito maior e o tempo de processamento é menor apresentado um ganho de qualidade e uma redução no tempo de processamento da resposta, uma vez que qualidade da resposta, para a FFT, começa a ficar boa a partir de 1000 amostras, levando em consideração que as respostas de amplitude e fase são prejudicadas pelo próprio processo de cálculo da FFT GIRGIS(1991) .

No caso de sinais aperiódicos os AGs apresentam excelentes resultados no tratamento dos sinais, pois permitem uma estimação precisa do sinal de entrada, com baixo tempo de processamento, assim uma escolha da fitness adequada permite estimar sinais de aperiódicos e até mesmos caóticos, provando que a ferramenta tem alta flexibilidade no tratamento de sinais.



## 5-CONCLUSÃO

Os Algoritmos genéticos mostram-se uma ferramenta de grande aplicabilidade nas estimativas e estudos dos sinais, sua aplicação em determinados casos provou ser muito superior aos métodos tradicionais como as FFT. No caso dos sinais periódicos e nos casos de sinais aperiódicos e caóticos provou ser uma ferramenta para esta carência da engenharia.

A estrutura simples dos AGs. permite ao usuário a aplicação modular, sua rápida implementação e execução para os diversos problemas com pequenas adaptações.

O grande problema dos AGs é a perfeita representação dos problemas e a busca de uma fitness, que seja pertinente premiando os indivíduos adaptados e punindo os indivíduos que desviam da solução do problema.

O Algoritmos Genéticos são ferramentas de otimização, portanto, sua resposta é uma alternativa ótima para a solução do problema, mas devido as probabilidades envolvidas no processo as respostas para duas execuções não são necessariamente idênticas.

Um fato bastante importante de ressaltar é a possibilidade de implementação dos AGs em dispositivos FPGAs, que combinadas as características de paralelismos permitem o desenvolvimento de produtos que operam em tempo real com desempenho superior aos equipamentos que usam os métodos tradicionais.

A utilização da ferramenta AG para estimativa dos sinais provou ser um poderoso recurso para determinar os sinais periódicos e aperiódicos presentes na grande gama de sinais encontrados em vários equipamentos, e ou processos de equipamentos facilitando a sua reprodução, conseqüentemente seu estudo.

O tempo de processamento de um AG, a precisão da resposta e o número de indivíduos da população são fatores a se levar em conta em uma aplicação, pois o tempo de execução do AG está ligado proporcionalmente ao número de indivíduos e ao número de iterações do AG e ambos ligados a precisão das respostas.

Os AGs se apresentam como uma alternativa para a estimação de sinais periódicos e como uma solução para estimação no caso de sinais aperiódicos e caóticos, conforme mostrado nas aplicações discutidas neste trabalho.

## **5.1 Sugestão de Trabalhos Futuros**

Como sugestão para trabalhos futuros pode ser feita a implementação em um sistema com multiprocessadores que melhorem o tempo de processamento, beneficiando-se da estrutura paralela do AG. e da divisão de tarefas nos núcleos processadores, devido ao baixo custo dos processadores possibilitando uma estrutura em que cada indivíduo fosse tratado por um processador específico que realizasse todas as fases do processo de cálculos e apresentasse os resultados para um processador central, num processo de computação distribuída e paralela minimizando sensivelmente o tempo de processamento e melhorando a capacidade de processamento, aproveitando os recursos dos processadores e a estrutura dos AGs.

## Referencias Bibliográficas

LIDEN, Ricardo - Algoritmo Genético –Rio de Janeiro : Brasport 2006;

VARGAS, Elis Tápia. “Uma proposta de relé digital de frequência baseado em algoritmos genéticos”. 2005. Dissertação (Mestrado em Engenharia Elétrica) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2005. Disponível em: <http://www.teses.usp.br/teses/disponiveis/18/18133/tde-17112006-112206/>. Acesso em: 2013-02-13.

SOUZA, S.A., Oleskovicz, M., Coury, D.V., Silva, T.V., Delbem, A.C.B. and Simões, E. V. (2008) “ FPGA implementation of genetic algorithm for frequency estimation in power systems”, in Proc. IEEE PES General Meeting, PitTsburgh, pp 1-6.

DUGAN,R.C., McGRANAGHAN, M.F., BEATY, H. W. (1996) “Eletrical Power System Quality”- McGraw-Hill.

BURRUS, C.S. e PARKS, T.W. (1985)” DFT/FFT and Convolution Algorithm: Teory and implementation”- Wiley- Intercience.

GIRGIS, A.A.; BIN CHANG, W.; MAKRAN, E.B. (1991).”A Digital Recursive Measurement Scheme for On-Line Tracking of Power System Harmonics.” IEEE Transactions on Power Delivery. 6(3): Pages: 1152-1160

CHANG, Y.N.; HSIEH, Y.C.; MOO, C.S. (2000). “Truncation Effects of FFT on Estimation of Dynamic Harmonics in Power System.” Power System Technology. International Conference on PowerCon 2000, Vol.: 3. Pages: 1155 – 1160

COURY,D.V., Oleskovicz, M.,Delbem, A. C.B., Simões, E.V., Silva, T. V., Barbosa, D. and Carvalho, J.R.(2011) “ Algoritmo Evolutivo para a Estimção da Frequencia em Sistemas Elétricos utilizando FPGAS” Revista Controle & Automação/ Vol.22 no.5/ Setembro e Outubro 2011 pp 495 – 505.

El-Naggar, K.M. and Youssef, H.K.M. (2000) “ A genetic based algorithm for frequency-relaying applications” *Electric Power Systems Research*, vol. 55, pp. 173-178.

COURY, D.V., Oleskovicz, M., Delbem, A. C.B., Simões, E.V., Silva, T. V., Barbosa, D. and Carvalho, J.R. (2009) “ A Genetic Based Algorithm for Frequency Relaying using FPGAs” *Proc. IEEE PES General Meeting, Calgary*, pp 1-6.

MACEDO, Renata Araripe de ; SILVA FILHO, Donato da ; COURY, Denis Vinicius ; CARNEIRO, A. A. F. M. ; CARVALHO, André Carlos Ponce de Leon Ferreira de . “Uma Rotina Baseada em Algoritmos Genéticos para Identificação de Harmônicos em Chaveamentos de Bancos de Capacitores”. In: *XV Congresso Brasileiro de Automática*, 2004, Gramado - RS, 2004.

SOUZA, S.A., OLESKOVICZ, M., e COURY D.V. “ Caracterização de Componentes Harmônicas em Sistema Elétrico via Algoritmos Genéticos” *Sociedade Brasileira de Qualidade em Energia Elétrica. SBQEE 2007 – Santos-SP*.

SOUZA, S.A., Macêdo, R.A., Vargas E.T., Coury, D.V. e Oleskovicz, M.” *Estimação de Parâmetros de um Sistema Elétrico de Potência Utilizando Algoritmos Genéticos” IEEE Latin America Transactions, Vol 4, No.1, March 2006*.

SOUZA, Silvio Aparecido de. “Algoritmos genéticos aplicados à proteção e estimação de harmônicos em sistemas elétricos de potência”. 2008. Tese (Doutorado em Sistemas Elétricos de Potência) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2008. Disponível em: <http://www.teses.usp.br/teses/disponiveis/18/18154/tde-17032009-150821/>. Acesso em: 2013-02-15.

MACÊDO, R.A., FILHO, D.S. e COURY, D.V. “Algoritmos Evolucionários Aplicados na Estimação de Parâmetros de um Sistema Elétrico de Potencia” 2006 Disponível em : [http://www.labplan.ufsc.br/congressos/SBSE/anais/096\\_sbse2006\\_final.pdf](http://www.labplan.ufsc.br/congressos/SBSE/anais/096_sbse2006_final.pdf) acesso em: 2013-02-15

SILVA, Raphael Philipe Mendes da. “Algoritmos genéticos aplicados à estimação fasorial em sistemas elétricos de potência”. 2012. Dissertação (Mestrado em Sistemas Elétricos de Potência) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2012. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18154/tde-05102012-091637/>>. Acesso em: 2013-02-13.

KAMWA, I., GRONDIN, R. (1992) “Fast Adaptive Schemes for Tracking Voltage Phasor and local Frequency in Power Transmission and Distribution Systems” IEEE Transaction on Power Delivery 7(2):789-794

COOLEY, J.W. (1992) “How FFT Gains Acceptance”. IEEE Signal Processing Magazine v.9, n.1, p 10-13.

EL-NAGGAR, K. M., YOUSSEF K. M. (2000) “A Genetic Based Algorithm for a frequency-Relaying” Applications Electric Power System Research v.55, p 173-178

GOLDBERG, D. E. (1989) “Genetic Algorithms in Search Optimization and Machine Learning, Reading”, M.A, Addison Wesley.

SOLIMAN, S. A., EL-NAGGAR, K. M., AL-KANDARI, A. (1997) “kalman filtering Based Algorithm for Low Frequency Power Systems Sub-Harmonics Identification”, International Power Energy and Systems, vol. 17, n. 1, pp 38-42.

GEN, M., CHENG R. (1996) “Genetic Algorithm and Engineering Design”, New York, John Wiley & Sons

GOLDBERG, D. E., (2002) “Design innovation: lessons from and for component genetic algorithms”, Kluwer Academic Publishers, Norwell, USA.

DEB, K. (2002) “Multi-objetive optimizations using evolutionary algorithms”, John Wiley & Sons.

DE JONG, K. A. (2006) "Evolutionary computation: a unified approach", MIT Press.

## Anexo A

Programa usado para série caótica

```

%%
%
% PROGRAM GENETIC ALGORITHM BY Evaldo Silva
% Inicialização
ereli=0;
er=zeros(1,30);
er2=zeros(1,30);
vt=zeros(1,1000);
x=zeros(1,1000);
alfaw=zeros(1,30);
alfawa=zeros(1,90);
betaw=zeros(1,30);
betawa=zeros(1,90);
taw=zeros(1,30);
tawa=zeros(1,90);
b=zeros(1,30);
c=zeros(1,30);
d=zeros(1,30);
y=zeros(1,30);
xw=zeros(1,30);
xwa=zeros(1,150);
clear;
% geração caotica
% entrada de dados tau é o target
%display(' Informe o tau');
tau=15;
%display ('informe o xi');
xi=1.4;
alfa=.9;
beta=.19;
Np=500;
% Geração de Sinal
for t=1:Np
    vt(t)=t-1;
    if t<=tau
        x(t)=xi;
    else
        x(t)=alfa*x(t-1)+(beta*x(t-tau))/(1+x(t-tau)^10);
    end
end
%grafico da curva
plot(vt,x); grid;
%Geração da população inicial

for i=1:30; % espaço de busca x=1 até 100
    taw(i)=randi(1000);
end
taw;
for i=1:30 % espaço de busca de y=0.01 até 5.0
    xw(i)=randi(500)/100;
end
xw;
for i=1:30 % espaço de busca de y=0.1 até 10.0
    alfaw(i)=randi(10000)/1000;
end
end

```

```

%alfaw(1)=0.9;
for i=1:30 % espaço de busca de y=0.1 até 10.0
    betaw(i)=randi(10000)/1000;
end
%betaw(1)=0.2;
%inicialização do contador para gerar índice iteração
ct1=0;
% loop de iteração definição do espaço da função que será analisado
% para outros espaços altera o parametro inicial e final de z
    for z=100:Np-100;

        % avaliação
            er2(1)=1;
            for a=1:30;
                ert2(a)=0;
                for t=z:z+50;
                    if z>taw(a);
                        b(a)=alfaw(a).*x(t-1);
                        c(a)=betaw(a).*x(t-taw(a));
                        d(a)=1+x(t-taw(a)).^10;
                        y(a)=b(a)+c(a)./d(a);
                        er(a)=x(t)-y(a);
                        % avaliação do fitness de cada individuo da população
                        er2(a)=er(a).^2;
                        else
                            er2(a)=2;
                        end
                        ert2(a)=ert2(a)+er2(a);
                    end
                end
            end
            % seleção da elite
            eli=1;
            ereali=ert2(1);
            for a=1:30;
                if ereali>ert2(a);
                    eli=(a);
                    ereali=ert2(a);
                    ct1=z-100;
                    ct1,alfaw(1),betaw(1),taw(1),ert2(1),
                end
            end
            %Cruzamento
            % Seleção por torneio
            for a=1:30;
                p1=randi(30); % seleção de quatro elementos
                p2=randi(30);
                p3=randi(30);
                p4=randi(30);
                % seleção de pai 1
                if ert2(p1)>=ert2(p2); % comparação das avaliações dos
elementos
                    sp1=p2;
                else
                    sp1=p1;
                end
                % seleção de pai 2
                if ert2(p3)>=ert2(p4); % comparação das avaliações do
elementos
                    sp2=p4;
                else
                    sp2=p3;
                end
            end
        end
    end
end

```



```

        end
        % ordenamento crescente
        if alfaw(sp1)>alfaw(sp2);
            spaux=sp2;
            sp2=sp1;
            sp1=spaux;
        end
        %calculo de media entre os pais
        deltalafaw=alfaw(sp2)-alfaw(sp1);
        % calculo do filho médio
        filmedalfaw=alfaw(sp1)+(deltalafaw/2);
        %calculo do filho menor
        filmenalfaw=filmedalfaw-deltalafaw;
        % Proteção contra transbordo do espaço de busca
        if filmenalfaw<0.01;
            filmenalfaw=0.01;
        end
        % calculo do filho maior
        filmaialfaw=filmedalfaw+deltalafaw;
        % Proteção contra transbordo do espaço de busca
        if filmaialfaw>=10;
            filmaialfaw=10;
        end
        % armazenamento do resultado do cruzamento
        ind=a*3;
        % Alfa
        alfawa(ind-2)=filmedalfaw;
        alfawa(ind-1)=filmenalfaw;
        alfawa(ind)=filmaialfaw;
    end
    % Seleção por torneio
    for a=1:30;
        p1=randi(30); % seleção de quatro elementos
        p2=randi(30);
        p3=randi(30);
        p4=randi(30);
        % seleção de pai 1
        if ert2(p1)>=ert2(p2); % comparação das avaliações dos
elementos
            sp1=p2;
        else
            sp1=p1;
        end
        % seleção de pai 2
        if ert2(p3)>=ert2(p4); % comparação das avaliações do
elementos
            sp2=p4;
        else
            sp2=p3;
        end
        % ordenamento crescente
        if betaw(sp1)>betaw(sp2);
            spaux=sp2;
            sp2=sp1;
            sp1=spaux;
        end
        %calculo de media entre os pais
        deltbetaw=betaw(sp2)-betaw(sp1);
        % calculo do filho médio
        filmedbetaw=betaw(sp1)+deltbetaw/2;
        %calculo do filho menor

```

```

    filmenbetaw=filmedbetaw-deltbetaw;
% Proteção contra transbordo do espaço de busca
    if filmenbetaw<0.01;
        filmenbetaw=0.01;
    end
    % calculo do filho maior
    filmaibetaw=filmedbetaw+deltbetaw;
% Proteção contra transbordo do espaço de busca
    if filmaibetaw>=10;
        filmaibetaw=10;
    end
% armazenamento do resultado do cruzamento
    ind=a*3;
% Beta
    betawa(ind-2)=filmedbetaw;
    betawa(ind-1)=filmenbetaw;
    betawa(ind)=filmaibetaw;
end
% Seleção por torneio
for a=1:30;
    p1=randi(30); % seleção de quatro elementos
    p2=randi(30);
    p3=randi(30);
    p4=randi(30);
    % seleção de pai 1
    if ert2(p1)>=ert2(p2); % comparação das avaliações dos
elementos
        sp1=p2;
    else
        sp1=p1;
    end
    % seleção de pai 2
    if ert2(p3)>=ert2(p4); % comparação das avaliações do
elementos
        sp2=p4;
    else
        sp2=p3;
    end
    % ordenamento crescente
    if taw(sp1)>taw(sp2);
        spaux=sp2;
        sp2=sp1;
        sp1=spaux;
    end
    %calculado de media entre os pais
    delttaw=taw(sp2)-taw(sp1);
    % calculo do filho médio
    filmedtaw=taw(sp1)+uint32(delttaw/2);
    %calculado do filho menor
    filmentaw=filmedtaw-delttaw;
% Proteção contra transbordo do espaço de busca
    if filmentaw<1;
        filmentaw=1;
    end
    % calculo do filho maior
    filmaitaw=filmedtaw+delttaw;
% Proteção contra transbordo do espaço de busca
    if filmaitaw>=1000;
        filmaitaw=1000;
    end
    % armazenamento do resultado do cruzamento

```

```

        ind=a*3;
    % Tau
        tawa(ind-2)=filmedtaw;
        tawa(ind-1)=filmentaw;
        tawa(ind)=filmaitaw;
end
alfaw(1)=alfaw(eli);
betaw(1)=betaw(eli);
taw(1)=taw(eli);
% seleção da nova população
for a=2:30;
    b=randi(90);
    alfaw(a)=alfawa(b);
    b=randi(90);
    betaw(a)=betawa(b);
    b=randi(90);
    taw(a)=tawa(b);
end
%mutações
for a=2:30;
    % mutação definida em 10%
    xm=randi(100);
    if xm<=10;
        if alfaw(a)<=5;
            alfaw(a)= alfaw(a)+1;
        else
            alfaw(a)=alfaw(a)-1;
        end
    end
    % mutação definida em 10%
    xm=randi(100);
    if xm<=10;
        if betaw(a)<=5;
            betaw(a)= betaw(a)+1;
        else
            betaw(a)=betaw(a)-1;
        end
    end
    % mutação definida em 10%
    xm=randi(100);
    if xm<=10;
        if tawa(a)<=50;
            taw(a)= taw(a)+10;
        else
            taw(a)=taw(a)-10;
        end
    end
end
end
end
% Calculo do xi
% a única maneira que consegui imaginar é retornar até o momento em
% que a amostra atual é igual a amostra anterior isto significa que
% a amostra esta sobre a influencia apenas de xi
% partindo do ultimo valor pesquisado em direção ao inicio da série
% quando encontrar o xi aplica-se o algoritmo genético...
%
flag=0;
for ct=1:Np;
    if flag==0;
        if x(z-ct-1)==x(z-ct);
            flag=1;
        end
    end
end

```

```

for c=1:20;
    for a=1:30; % avaliação 31 indivíduos
        er(a)=x(z-ct)-xw(a);
        er2(a)=er(a)^2;
    end
% seleção da elite
eli=1;
ereli=er2(1);
for a=1:30;
    if ereli>er2(a);
        eli=(a);
        ereli=er2(a);
    end
end
%Cruzamento
% Seleção por torneio
for a=1:30;
    p1=randi(30); % seleção de quatro elementos
    p2=randi(30);
    p3=randi(30);
    p4=randi(30);
% seleção de pai 1
    if er2(p1)>=er2(p2); % comparação das
avaliações
        %dos elementos
        sp1=p2;
        else
        sp1=p1;
        end
% seleção de pai 2
    if er2(p3)>=er2(p4); % comparação das
avaliações
        %dos elementos
        sp2=p4;
        else
        sp2=p3;
        end
% ordenamento crescente
    if xw(sp1)>xw(sp2);
        spaux=sp2;
        sp2=sp1;
        sp1=spaux;
    end
%calculo de media entre os pais
    deltxw=xw(sp2)-xw(sp1);
% calculo do filho médio
    filmedxw=xw(sp1)+(deltxw/2);
%calculo do filho menor
    filmenxw=filmedxw-deltxw;
% Proteção contra transbordo do espaço de busca
    if filmenxw<0.01;
        filmenxw=0.01;
    end
% calculo do filho maior
    filmaixw=filmedxw+deltxw;
% Proteção contra transbordo do espaço de busca
    if filmaixw>=5;
        filmaixw=5;
    end
% armazenamento do resultado do cruzamento
    ind=a*5;

```

```

        xwa(ind-4)=filmenxw;
        xwa(ind-3)=xw(sp1);
        xwa(ind-2)=filmedxw;
        xwa(ind-1)=xw(sp2);
        xwa(ind)=filmaixw;
    end
    % mutação definida em 10%
    for a=1:150;
        xym=randi(100);
        if xym<=10;
            if xwa(a)<=2.5;
                xwa(a)= xwa(a)+0.5;
            else
                xwa(a)=xwa(a)-0.5;
            end
        end
    end
    % transferencia do elite para nova população
    xw(1)=xw(eli);
    % seleção da nova população
    for a=2:30;
        b=randi(150);
        xw(a)=xwa(b);
    end
end
end
end
    % publicação dos resultados
    display ('xi=');xi,xw(1),
    display ('tau=');tau,taw(1),
    display ('alfa');alfa,alfaw(1),
    display ('beta');beta,betaw(1),
    % Geração de Sinal
    for t=1:1000
        vd(t)=t-1;
        if t<=tau
            xo(t)=xi;
        else
            xo(t)=alfa*xo(t-1)+(beta*xo(t-tau))/(1+xo(t-tau)^10);
        end
    end
    %grafico da curva
    plot(vd,xo,'b'); grid;
    for t=1:1000
        vd(t)=t-1;
        if t<=taw(1)
            xu(t)=xw(1);
        else
            xu(t)=alfaw(1)*xu(t-1)+(betaw(1)*xu(t-taw(1)))/(1+xu(t-
            taw(1))^10);
        end
    end
end
hold
%grafico da curva
plot(vd,xu,'r'); grid;

```

## Anexo B

Programa usado para estimativa dos sinais periódicos

```

%%
clear;
clc;
%
%
% PROGRAM GENETIC ALGORITHM BY Evaldo Silva
% Definição do espaço de busca inicialização da população
A=0.7:0.6/29:1.3; %espaço de busca amplitude
%A
f=58:4/29:62; %espaço de busca Freqüencia
%f
teta=0:2*pi/29:2*pi; % espaço de busca fase
%teta
ta=800;
t=0:1/ta:14/ta; % amostragem
%t
Ae=1; % entrada de dados
fe=60;
tetae=0;
% prelocation
d=zeros(1,30);
c=zeros(1,30);
Aa=zeros(1,150);
fa=zeros(1,150);
tetaa=zeros(1,150);
e=zeros(1,30);
e2=zeros(1,30);
se2=zeros (1,30);
% Avaliação
% t é o tamanho da janela a é a amostra c é o somatorio do erro
quadrático
for a=1:30;
    y=Ae*sin(2*pi*fe*t+tetae)-A(a)*sin(2*pi*f(a)*t+teta(a));
    se2(a)=0;

    for d=1:14;
        e2(d)=y(d)*y(d);
        se2(a)=se2(a)+e2(d);
    end
end
se2
eli=1;
r=se2(1); % inicialização da elite
for a=1:30;
    if r>se2(a);
        r=se2(a);
        eli=a; % o A(e)*sin(2*pi*f(e)+ teta(e)) é o melhor avaliado;
    end
end
eli
r
'população'
A
f

```

teta

```

% Seleção por torneio
for a=1:15;
    p1=randi(30); % seleção de quatro elementos
    p2=randi(30);
    while p2==p1
        p2=randi(30);
    end
    p3=randi(30);
    p4=randi(30);
    while p3==p4
        p4=randi(30);
    end
    % seleção de pai 1
    if se2(p1)>=se2(p2); % comparação das avaliações dos elementos
        sp1=p2;
    else
        sp1=p1;
    end

    % seleção de pai 2
    if se2(p3)>=se2(p4); % comparação das avaliações do elementos
        sp2=p4;
    else
        sp2=p3;
    end
% ordenamento crescente
    if A(sp1)>A(sp2);
        spaux=sp2;
        sp2=sp1;
        sp1=spaux;
    end

    %calcudo de delta da amplitude
    deltA=A(sp2)-A(sp1);
    if deltA==0
        deltA=.1;
    end
    filmedA=A(sp1)+deltA/2;
    filmenA=filmedA-deltA;

    % Proteção contra transbordo do espaço de busca
    if filmenA<0.7;
        filmenA=0.7;
    end
    filmaiA=filmedA+deltA;

    % Proteção contra transbordo do espaço de busca
    if filmaiA>1.3;
        filmaiA=1.3;
    end
    ind=a*3;
    % Amplitude
    Aa(ind-2)=filmenA;
    Aa(ind-1)=filmedA;
    Aa(ind)=filmaiA;
Aa;
% sp2
% sp1

```

```

%A(sp2)
%A(sp1)
%deltaA
%filmenA
%filmedA
%filmaiA
% Calculo da frequencia

    p1=randi(30); % seleção de quatro elementos
    p2=randi(30);
    while p2==p1
        p2=randi(30);
    end
    p3=randi(30);
    p4=randi(30);
    while p3==p4
        p4=randi(30);
    end
    % torneio
% seleção de pai 1
    if se2(p1)>=se2(p2); % comparação das avaliações dos elementos
        sp1=p2;
    else
        sp1=p1;
    end

    % seleção de pai 2
    if se2(p3)>=se2(p4); % comparação das avaliações do elementos
        sp2=p4;
    else
        sp2=p3;
    end

    % ordenamento crescente
    if f(sp1)>f(sp2);
        spaux=sp2;
        sp2=sp1;
        sp1=spaux;
    end
%calcudo de delta da frequencia
    deltf=f(sp2)-f(sp1);
    if deltf==0
        deltf=.1;
    end
    filmedf=f(sp1)+deltf/2;
    filmenf=filmedf-deltf;

    % Proteção contra transbordo do espaço de busca
    if filmenf<58;
        filmenf=58;
    end
    filmaif=filmedf+deltf;

    % Proteção contra transbordo do espaço de busca
    if filmaif>62;
        filmaif=62;
    end
ind=a*3;
    % Frequencia
    fa(ind-2)=filmenf;

```



```

        fa(ind-1)=filmedf;
        fa(ind)=filmaif;
    fa;
% sp2
% sp1
%f(sp2)
%f(sp1)
%deltf
%filmenf
%filmedf
%filmaif
% calculo da fase

    p1=randi(30); % seleção de quatro elementos
    p2=randi(30);
    while p2==p1
        p2=randi(30);
    end
    p3=randi(30);
    p4=randi(30);
    while p3==p4
        p4=randi(30);
    end
    % torneio
% seleção de pai 1
    if se2(p1)>=se2(p2); % comparação das avaliações dos elementos
        sp1=p2;
    else
        sp1=p1;
    end

    % seleção de pai 2
    if se2(p3)>=se2(p4); % comparação das avaliações do elementos
        sp2=p4;
    else
        sp2=p3;
    end

    % ordenamento crescente
    if teta(sp1)>teta(sp2);
        spaux=sp2;
        sp2=sp1;
        sp1=spaux;
    end
%calcudo de delta da fase
    deltteta=teta(sp2)-teta(sp1);
    if deltteta==0
        deltteta=0.1;
    end
    filmedteta=teta(sp1)+deltteta/2;
    filmenteta=filmedteta-deltteta;

    % Proteção contra transbordo do espaço de busca
    if filmenteta<0;
        filmenteta=0;
    end
    filmaiteta=filmedteta+deltteta;

    % Proteção contra transbordo do espaço de busca
    if filmaiteta>2*pi;

```

```

        filmaiteteta=2*pi;
    end
    %armazenamento da população auxiliar
    ind=a*3;
    % Fase
    tetaa(ind-2)=filmenteta;
    tetaa(ind-1)=filmedteta;
    tetaa(ind)=filmaiteteta;
tetaa;
% sp2
% sp1
%teta(sp2)
%teta(sp1)
%deltteta
%filmenteta
%filmedteta
%filmaiteteta
end
% transferencia do elite para nova população
A(1)=A(eli);
f(1)=f(eli);
teta(1)=teta(eli);
Ael=A(1);
fel=f(1);
tetael=teta(1);
% seleção da nova população
for a=2:30;
    b=randi(45);
    A(a)=Aa(b);
    f(a)=fa(b);
    teta(a)=tetaa(b);
end

%inicialização do contador de iteração
x=0;
rm=1;

% avaliação do fitness de cada individuo da população
while x<100
    %      'nova população'
%A
%f
%teta
%pause;
% loop de iteração
    % tamanho da janela deslizante 4 amostras
    t=t+1/ta;
    for a=1:30;
y=Ae*sin(2*pi*fe*t+tetae)-A(a)*sin(2*pi*f(a)*t+teta(a));
se2(a)=0;
        for b=1:14;
            e2(b)=y(b)*y(b);
            se2(a)=se2(a)+e2(b);
        end
    end

r=se2(1); % inicialização da elite

for a=1:30;
    if r>se2(a);
        r=se2(a);
    end
end

```

```

        eli=a; % o  $A(e) \cdot \sin(2 \cdot \pi \cdot f(e) + \text{teta}(e))$  é o melhor avaliado
    end
end
% se2
% eli
% r
% x
if r<rm; rm=r;r,x,Ael=A(eli),fel=f(eli),tetael=teta(eli),end
% graficos de desempenho
% figure;
% grid on;
% subplot(2,1,1);
% plot (t,s);
% subplot(2,1,2);
% plot (t,se),

% Seleção por torneio
for a=1:15;
    p1=randi(30); % seleção de quatro elementos
    p2=randi(30);
    while p2==p1
        p2=randi(30);
    end
    p3=randi(30);
    p4=randi(30);
    while p3==p4
        p4=randi(30);
    end
    % torneio
% seleção de pai 1
    if se2(p1)>=se2(p2); % comparação das avaliações dos elementos
        sp1=p2;
    else
        sp1=p1;
    end

    % seleção de pai 2
    if se2(p3)>=se2(p4); % comparação das avaliações do elementos
        sp2=p4;
    else
        sp2=p3;
    end

    % ordenamento crescente
    if A(sp1)>A(sp2);
        spaux=sp2;
        sp2=sp1;
        sp1=spaux;
    end
    %calcudo de delta da amplitude
    deltA=A(sp2)-A(sp1);
    if deltA==0
        deltA=0.6/30;
    end
    filmedA=A(sp1)+deltA/2;
    filmenA=filmedA-deltA;

    % Proteção contra transbordo do espaço de busca
    if filmenA<0.7;
        filmenA=0.7;
    end
end

```

```

filmaiA=filmedA+deltA;

% Proteção contra transbordo do espaço de busca
if filmaiA>1.3;
    filmaiA=1.3;
end
%
% sp2
% sp1
% A(sp2)
% A(sp1)
% delatA
% filmedA
% filmenA
% filmaiA
% armazenamento do resultado do cruzamento
ind=a*3;
% Amplitude
Aa(ind-2)=filmenA;
Aa(ind-1)=filmedA;
Aa(ind)=filmaiA;
Aa;
p1=randi(30); % seleção de quatro elementos
p2=randi(30);
while p2==p1
    p2=randi(30);
end
p3=randi(30);
p4=randi(30);
while p3==p4
    p4=randi(30);
end
% torneio
% seleção de pai 1
if se2(p1)>=se2(p2); % comparação das avaliações dos elementos
    sp1=p2;
    else
    sp1=p1;
end

% seleção de pai 2
if se2(p3)>=se2(p4); % comparação das avaliações do elementos
    sp2=p4;
    else
    sp2=p3;
end
% ordenamento crescente
if f(sp1)>f(sp2);
    spaux=sp2;
    sp2=sp1;
    sp1=spaux;
end
%calculo de delta da frequencia
deltf=f(sp2)-f(sp1);
if deltf==0
    deltf=4/30;
end
filmedf=f(sp1)+deltf/2;
filmenf=filmedf-deltf;

% Proteção contra transbordo do espaço de busca
if filmenf<58;

```

```

        filmenf=58;
    end

    filmaif=filmedf+deltf;

    % Proteção contra transbordo do espaço de busca
    if filmaif>62;
        filmaif=62;
    end
%     sp2
%     sp1
%     f(sp2)
%     f(sp1)
%     deltf
%     filmedf
%     filmenf
%     filmaif
%     % armazenamento do resultado do cruzamento
ind=a*3;
    % Frequencia
    fa(ind-2)=filmenf;
    fa(ind-1)=filmedf;
    fa(ind)=filmaif;
fa;
%     calculo da fase
p1=randi(30); % seleção de quatro elementos
p2=randi(30);
while p2==p1
    p2=randi(30);
end
p3=randi(30);
p4=randi(30);
while p3==p4
    p4=randi(30);
end
% torneio
% seleção de pai 1
if se2(p1)>=se2(p2); % comparação das avaliações dos elementos
    sp1=p2;
    else
    sp1=p1;
end

% seleção de pai 2
if se2(p3)>=se2(p4); % comparação das avaliações do elementos
    sp2=p4;
    else
    sp2=p3;
end

% ordenamento crescente
if teta(sp1)>teta(sp2);
    spaux=sp2;
    sp2=sp1;
    sp1=spaux;
end
% calculo de delta da fase
deltteta=teta(sp2)-teta(sp1);
if deltteta<0
    deltteta=-deltteta;
end

```

```

if deltteta==0
    deltteta=0.0001;% evita divisão por zero
end
filmedteta=teta(sp1)+deltteta/2;
if filmedteta > 2*pi
    filmedteta=filmedteta-2*pi;
end
filmenteta=filmedteta-deltteta;

% Proteção contra transbordo do espaço de busca
if filmenteta<0;
    filmenteta=filmenteta+2*pi;
end

filmaiteta=filmedteta+deltteta;

% Proteção contra transbordo do espaço de busca
if filmaiteta>2*pi;
    filmaiteta=filmaiteta-2*pi;
end
sp2
sp1
teta(sp2)
teta(sp1)
filmedteta
filmenteta
    filmaiteta
%armazenamento do resultado do cruzamento
ind=a*3;
% Fase
tetaa(ind-2)=filmenteta;
tetaa(ind-1)=filmedteta;
tetaa(ind)=filmaiteta;
tetaa;
end
% Aa
% fa
% tetaa
% mutação
for a=1:45
    xm=randi(100);
    if xm<=10;
        if Aa(a)<=1;
            Aa(a)= Aa(a)+0.6/20;
            if fa(a)<=60;
                fa(a)= fa(a)+4/20;
                if tetaa(a)<pi;
                    tetaa(a)=tetaa(a)+2*pi/20;
                else
                    tetaa(a)=tetaa(a)-2*pi/20;
                end
            else
                fa(a)=fa(a)-4/20;
                if tetaa(a)<pi;
                    tetaa(a)=tetaa(a)+2*pi/20;
                else
                    tetaa(a)=tetaa(a)-2*pi/20;
                end
            end
        else
            Aa(a)=Aa(a)-0.6/20;
        end
    end
end

```

```

        if fa(a)<=60;
            fa(a)= fa(a)+4/20;
            if tetaa(a)<pi;
                tetaa(a)=tetaa(a)+2*pi/20;
            else
                tetaa(a)=tetaa(a)-2*pi/20;
            end
        else
            fa(a)=fa(a)-4/20;
            if tetaa(a)<pi;
                tetaa(a)=tetaa(a)+2*pi/20;
            else
                tetaa(a)=tetaa(a)-2*pi/20;
            end
        end
    end
end
end
end
end
% 'mutação'
% Aa
% fa
% tetaa
% transferencia do elite para nova população
A(1)=A(eli);
f(1)=f(eli);
teta(1)=teta(eli);
plot(t,Ae*sin(2*pi*fe*t+tetae),t,A(1)*sin(2*pi*f(1)*t+teta(1)))
drawnow
% seleção da nova população
for a=2:30;
    b=randi(45);
    A(a)=Aa(b);
    b=randi(45);
    f(a)=fa(b);
    b=randi(45);
    teta(a)=tetaa(b);
end
% incremento do numero de iterações
x=x+1;
end
Ae,
Ael,
erA=abs(Ae-Ael);
erA,
errA=(1-(0.6-erA)/0.6)*100;
errA,
fe,
fel,
erf=abs(fe-fel);
erf,
errf=(1-(4-erf)/4)*100;
errf,
tetae,
tetael,
erte=abs(tetae-tetael);
erte,
errte=(1-(2*pi-erte)/(2*pi))*100;
errte,
x=0;
y=0;
t=0:1/ta:14/ta;

```

```
x=Ae*sin (2*pi*fe*t + tetae);  
y=Ael*sin(2*pi*fel*t+tetael);  
plot (t,x,'b',t,y,'r');  
title('Sinal de entrada e sinal estimado');  
xlabel('tempo (ms)');  
ylabel('Amplitude');  
legend('sinal adquirido ','sinal estimado');  
grid on;
```