sistema de bibliotecas ufg

UFG

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR AS TESES E DISSERTAÇÕES ELETRÔNICAS (TEDE) NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: **[X] Dissertação** **[ ] Tese**

2. Identificação da Tese ou Dissertação

Autor (a):	Diego Fernando Burgos Beltrán					
E-mail:	d.burgos88@hotmail.com					
Seu e-mail pode ser disponibilizado na página? [X]Sim [ ] Não						
Vínculo empregatício do autor	Nenhum					
Agência de fomento:	PAEC-OEA-GCUB			Sigla:	PAEC	
País:	Brasil	UF:	GO	CNPJ:	10.789.274/0001-65	
Título:	Algoritmos Genéticos Compactos para Estimação de Direção de Chegada e Conformação de Feixe num Arranjo de Antenas em Ambiente CDMA					
Palavras-chave:	Algoritmos genéticos; Mínimos Quadrados; Beamforming; Rastreamento; Processamento e detecção de sinais					
Título em outra língua:	Compact Genetic Algorthms for Direction of Arrival Estimation and Beamforming of an Antenna Array in a CDMA Environment					
Palavras-chave em outra língua:	Compact genetic algorithms; Least mean squares; Beamforming; Target tracking; Signal processing					
Área de concentração:	Engenharia de Computação					
Data defesa: (dd/mm/aaaa)	06/07/2015					
Programa de Pós-Graduação:	Engenharia Elétrica e de Computação					
Orientador (a):	Rodrigo Pinto Lemos					
E-mail:	lemos@ufg.br					
Co-orientador (a):*	Nenhum					
E-mail:	nenhum					

*Necessita do CPF quando não constar no SisPG

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF ou DOC da tese ou dissertação.

O sistema da Biblioteca Digital de Teses e Dissertações garante aos autores, que os arquivos contendo eletronicamente as teses e ou dissertações, antes de sua disponibilização, receberão procedimentos de segurança, criptografia (para não permitir cópia e extração de conteúdo, permitindo apenas impressão fraca) usando o padrão do Acrobat.

Diego F. Burgos

Assinatura do (a) autor (a)

Data: 08/07/2015

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

ALGORITMOS GENÉTICOS COMPACTOS PARA ESTIMAÇÃO DE DIREÇÃO DE CHEGADA E CONFORMAÇÃO DE FEIXE NUM ARRANJO DE ANTENAS EM AMBIENTE CDMA

Diego Fernando Burgos Beltrán

Orientador: Prof. Dr. Rodrigo Pinto Lemos

Goiânia
2015

DIEGO FERNANDO BURGOS BELTRÁN

ALGORITMOS GENÉTICOS COMPACTOS PARA ESTIMAÇÃO DE DIREÇÃO DE CHEGADA E CONFORMAÇÃO DE FEIXE NUM ARRANJO DE ANTENAS EM AMBIENTE CDMA

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica e de Computação, da Escola de Engenharia Elétrica, Mecânica e de Computação da Universidade Federal de Goiás, para obtenção do título de Mestre em Engenharia Elétrica e de Computação.
Área de Concentração: Engenharia de Computação.
Orientador: Prof. Dr. Rodrigo Pinto Lemos.

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO
COORDENAÇÃO DE PÓS-GRADUAÇÃO

FOLHA DE APROVAÇÃO

"Algoritmos Genéticos Compactos para Estimação de Direção de Chegada e Conformação de Feixe num Arranjo de Antenas em Ambiente CDMA"

DIEGO FERNANDO BURGOS BELTRÁN

Dissertação defendida e aprovada pela banca examinadora constituída pelos senhores:

Rodrigo Pinto Lemos – Orientador (EMC/UFG)

Leonardo Rodrigues Araújo Xavier de Menezes – FT/UnB

Flávio Henrique Teles Vieira – EMC/UFG

Leonardo da Cunha Brito – EMC/UFG

Goiânia, 06 de julho de 2015

Ficha catalográfica elaborada automaticamente
com os dados fornecidos pelo(a) autor(a), sob orientação do Sibi/UFG.

Burgos Beltrán, Diego Fernando
ALGORITMOS GENETICOS COMPACTOS PARA ESTIMAÇAO DE DIREÇAO DE CHEGADA E CONFORMAÇAO DE FEIXE NUM ARRANJO DE ANTENAS EM AMBIENTE CDMA [manuscrito] / Diego Fernando Burgos Beltrán. - 2015.
18, 95 f.: il.

Orientador: Prof. Dr. Rodrigo Pinto Lemos.
Dissertação (Mestrado) - Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) , Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Goiânia, 2015.
Bibliografia.
Inclui abreviaturas, símbolos, gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. algoritmos genéticos. 2. estimação de direção de chegada. 3. conformação de feixe. 4. mínimos quadrados. 5. rastreamento. I. Pinto Lemos, Rodrigo, orient. II. Título.

"*À minha mãe, Nydia Beltrán, por ser meu anjo guardião.*"

Agradecimentos

Primeiramente, a Deus, por todas as oportunidades que tive para meu crescimento Profissional.

Aos meus pais Nydia e Jorge, por ter me dado amor, apoio, alegrias, oportunidades e conselhos durante toda a minha vida.

À minha namorada, Cindy Rosseth, por insistir para que eu aproveitasse esta oportunidade e pela paciência que teve nos momentos de ausência para prosseguir com os meus estudos durante este curso de Mestrado.

Ao meu orientador e amigo, Prof. Dr. Eng°. Rodrigo Pinto Lemos, pela oportunidade outorgada, pela confiança depositada em mim, por sua amizade, pelos conselhos, orientações, e pela paciência.

A todos os meus professores e colegas da escola de Engenharia Elétrica, Mecânica e de Computação pelos conhecimentos compartilhados.

Resumo

Os contínuos avanços tecnológicos nas áreas da eletrônica e da programação tornaram as técnicas de processamento de sinais muito mais fáceis de implementar, permitindo a incorporação delas nos sistemas de comunicação, melhorando a performance destes. Neste trabalho desenvolve-se o problema de estimação da direção de chegada ou ângulo de incidência (DOA) de frentes de ondas eletromagnéticas sobre um arranjo linear de antenas, além da conformação de feixe (*beamforming*) do arranjo. Dentre as diversas técnicas existentes na literatura, o algoritmo de Mínima Média Quadrática (LMS, do inglês Least Mean Squared) é um método determinístico que se destaca por sua simplicidade, facilidade de implementação e a tendência de encontrar mínimos locais como resposta. Por outro lado, o Algoritmo Genético (AG) é um método heurístico que garante uma exploração mais completa de possibilidades evitando a tendência de cair em mínimos locais, mas oferecendo uma maior dificuldade de implementação, além de maior complexidade computacional. Recentemente, foi proposto o Algoritmo Genético Compacto (AGC), que é uma ferramenta que compartilha todas as virtudes dos Algoritmos Genéticos, porém sem exigir o grande custo computacional que um AG implica. Como este método ainda não foi utilizado para o controle de arranjos de antenas, este trabalho propõe utilizá-lo na estimação da DOA e *beamforming*, além de agregar-lhe uma série de modificações a fim de torná-lo mais robusto e mais completo, apesar de computacionalmente mais pesado. Neste trabalho exibe-se simulações em que o receptor adaptativo proposto é avaliado sob diferentes situações de relação sinal ruído (SNR), quantidade de fontes interferentes e velocidade de convergência. Além disso, simulam-se situações de rastreamento de usuários em movimento, onde é posta à prova a capacidade do receptor adaptar seu diagrama de radiação. Todos os testes foram feitos no ambiente de multiplicidade de acesso via divisão por códigos (CDMA), onde a única informação disponível no receptor são os códigos de espalhamento das fontes. Para conferir o funcionamento do AGC, comparou-se seu desempenho com aquele do algoritmo LMS sob as mesmas condições de simulação.

O desenvolvimento desta tese permitiu a publicação dos artigos *Adaptive Beamforming for Moving Targets Using Genetic Algorithms and a CDMA Reference Signal* no IEEE Colombian Conference on Communications and Computing COLCOM 2015 e

Adaptive Beamforming for Moving Targets Using Genetic Algorithms no IEEE Workshop on Engineering Applications WEA 2015 – International Congress on Engineering, este ultimo foi aceito para ser publicado como uma versão estendida na revista INGENIERÍA da universidade Distrital Francisco José de Caldas de Bogotá, Colômbia.

Abstract

The continuous technological advances in the areas of electronics and programming made the signal processing techniques much easier to implement, allowing them to be incorporated in the communication systems, improving their performance. This work approaches the problem of estimating direction of arrival or angle of incidence (DOA) of electromagnetic wave fronts of a linear antenna array, and of beamforming of the array. Among the various techniques that exist in the literature, the Least Mean Squared algorithm (LMS) is a deterministic method that stands out for its simplicity, ease of implementation and the tendency to find local minima. On the other hand, the Genetic Algorithm (GA) is a heuristic method that ensures more comprehensive exploration possibilities avoiding the tendency of sticking to local minima, but offering greater difficulty of implementation, and higher computational complexity. The recently proposed Compact Genetic Algorithm (cGA) is a tool that shares all the virtues of GA, but without requiring the large computational cost that a GA entails. Since this method has not yet been used for controlling antenna arrays, this paper proposes to use it as the estimation of DOA and beamforming, in addition to enhance it with a number of modifications to make it more robust and more complete, though making it computationally heavier. This work presents simulations where the proposed adaptive receiver is evaluated under different scenarios of signal to noise ratio (SNR), number of interfering sources and convergence velocity. Moreover, moving users tracking situations are simulated, where the receiver's ability to adapt its radiation pattern is tested. All tests were done in the code division multiple access (CDMA) environment, where the only information available to the receiver are the sources spreading codes. To verify the operation of the cGA, its performance was compared with that of the LMS algorithm simulation under the same simulation conditions.

The development of this thesis allowed to publish the articles named Adaptive Beamforming for Moving Targets Using Genetic Algorithms and a CDMA Reference Signal in the IEEE Colombian Conference on Communications and Computing COLCOM 2015, and Adaptive Beamforming for Moving Targets Using Genetic Algorithms in the IEEE Workshop on Engineering Applications WEA 2015 – International Congress on Engineering. The last

one was accepted as an extended version to be publish in the magazine INGENIERÍA that belongs to the Distrital Francisco José de Caldas University in Bogotá, Colombia.

Sumário

1 Introdução 23
1.1 Motivação 24
1.2 Objetivo Geral 25
1.3 Objetivos Específicos 25
1.4 Organização dos Capítulos 26
2 Modelo de Sinal e Antenas Inteligentes 28
2.1 Introdução 28
2.2 Fundamentos de Arranjos de Antenas 28
2.3 Considerações para a Modelagem 29
2.4 Equacionamento do Arranjo Linear Uniforme 30
2.5 Analogia entre ULA e Filtros FIR 33
2.6 Outras considerações sobre o sinal 34
2.7 Acesso Múltiplo em Sistemas de Comunicação 35
2.7.1 Espalhamento Espectral 36
2.7.1.1 Espalhamento Espectral por Sequencia Direta (DS-SS) 37
2.7.1.2 Códigos de Walsh 39
2.8 Modelo do Receptor Proposto 39
2.9 Vantagens das Antenas Inteligentes 42
2.10 Tipos de Conformadores de Feixe 43
3 Algoritmos Adaptativos e Genéticos 45
3.1 Introdução 45
3.2 Algoritmos e Filtros Adaptativos para Arranjos de Antenas 45
3.2.1 Algoritmo de Mínima Média Quadrática LMS 47
3.3 Algoritmos Bioinspirados 52
3.4 Computação Evolucionária 53
3.4.1 Algoritmos Genéticos 54
4 Algoritmo Genético Compacto 62
4.1 Introdução 62
4.2 Algoritmo Genético Compacto 62
4.2.1.1 A família de Algoritmos Genéticos Compactos 64
4.2.2 A Família de AGC na Estimação de DOA e Beamforming 67

4.2.2.1 Codificação do Espaço de Busca 70
5 Análise e Resultados de Desempenho de Estimação e Complexidade Computacional 71
5.1 Introdução 71
5.2 Considerações do Ambiente de Simulação 71
5.3 Bloco Sincronizador 72
5.4 Bloco Estimador de Bit 74
5.5 Bloco Algoritmo Adaptativo 76
5.5.1 Desempenho de Estimação para Fontes Estáticas Descorrelacionadas 79
5.5.2 Desempenho de Estimação para Fontes Estáticas Totalmente Correlacionadas 89
5.6 Detecção de Usuários em Movimento 97
5.6.1 Fonte de Interesse em Movimento 98
5.6.2 Todas as Fontes em Movimento 104
5.7 Análise de Resultados 109
6 Conclusões e Trabalhos futuros 112
6.1 Introdução 112
6.2 Conclusões 112
6.3 Trabalhos Futuros 113

Lista de abreviaturas e siglas

AE	Algoritmos Evolutivos.
AG	Algoritmo Genético.
AGC	Algoritmo Genético Compacto.
AGCe	Algoritmo Genético Compacto com elitismo.
AGCm	Algoritmo Genético Compacto com mutação.
AGCemr	Algoritmo Genético Compacto com elitismo, mutação e reamostragem.
AGS	Algoritmo Genético Simples.
BER	do inglês *Bit Error Rate* – Taxa de Erro de Bit.
CDMA	do inglês *Code-Division Multiple Access* – Acesso Múltiplo por Divisão em Código
CE	Computação Evolucionária.
CMA	do inglês *Constant Modulus Algorithm* – Algoritmo de Módulo Constante.
DOA	do inglês *Direction-Of-Arrival* – Direção de Chegada – Ângulo de Incidência.
DS-CDMA	do inglês *Direct Sequence Code Division Multiple Access* – Acesso Múltiplo por Divisão em Código por Sequencia Direta.
DSP	do inglês *Digital Signal Processor* – Processador Digital de Sinais.
DS-SS	do inglês *Direct Sequence Spread Spectrum* – Espalhamento Espectral por Sequencia Direta.
EDA	do inglês *Estimation of Distribution Algorithm* – Algoritmo de Estimação de distribuição.
FDMA	do inglês *Frequency Division Multiple Access* – Acesso Múltiplo por Divisão em Frequência.
IC	Inteligência Computacional.
LSM	do inglês *Least Mean Square* – Mínima Média Quadrática.
LS-CMA	do inglês *Least-Square Constant Modulus Algorithm* – Algoritmo de Mínimos Quadrados com Módulo Constante.
MODE	do Inglês *Method Of Direction Estimation* – Método de Estimação de Direção.
MODEX	do Inglês *MODE with eXtra roots* – MODE com raízes extras.
MSE	do inglês *Mean Squared Error* – Erro Quadrático Médio.

MUSIC	do inglês *MUltiple SIgnal Characterization* – Caracterização de Mútiplos Sinais.
NAMI	do inglês *Noise-alone matrix inverse* – Inverso-da-matriz-de-ruído.
PDF	do inglês *Probability Density Function* – Função densidade de Probabilidade.
RMSE	do inglês *Root Mean Square Error* – Raiz do Erro Quadrático Médio.
RF	Rádiofrequência.
RLS	do inglês, *Recursive-Least Squares* – Mínima Média Quadrática Recursivo.
SNR	do inglês *Signal-to-Noise Ratio* – Relação Sinal-Ruído.
SEAD	do Inglês SEArch of direction by Differential spectrum – Busca de Direção por Espectro Diferencial.
SDMA	do inglês *Space-Division Multiple Access* – Acesso Múltiplo por Divisão Espacial.
SIR	do inglês *Signal-to-Interference Ratio* – Relação Sinal-Interferência.
SIRN	do inglês *Signal-to-Interference plus Noise Ratio* – Relação Sinal-Interferência mais Ruído.
SPNMI	do inglês *Signal-plus-noise matrix inverse* – Inverso-da-matriz-de-sinal-ruído.
SS	do inglês, *Spread Spectrum* – Espalhamento Espectral.
TDMA	do inglês *Time Division Multiple Access* – Acesso Múltiplo por Divisão no Tempo.
ULA	do Inglês *Uniform Linear Array* – Arranjo Linear Uniforme.

Lista de símbolos

λ é o comprimento de onda.

c é a velocidade da luz no vácuo.

f_c é a frequência da portadora.

d é o espaçamento entre os sensores adjacentes no arranjo linear de sensores.

r é o espaço percorrido pela frente de onda.

θ é a direção da frente de onda eletromagnética plana.

τ é o atraso da frente de onda em relação a um deslocamento r.

φ é o defasagem da frente de onda em relação a um deslocamento r.

W é o vetor de pesos ou coeficientes do arranjo de sensores.

w_M é o coeficiente do filtro espacial referente ao M-ésimo sensor.

z_M é o M-ésimo sensor do arranjo.

$e^{-j\varphi}$ é a envoltória complexa defasada do sinal induzido nos sensores.

$x_K(t)$ é o sinal recebido pelo sensor 1 e originado pela K-ésima fonte.

$n_M(t)$ é o ruído referente ao M-ésimo sensor.

$A_{MK}(\theta)$ é a matriz de direção.

$Z_M(t)$ é a matriz de sinais induzidos nos sensores.

$N_M(t)$ é a matriz de ruído branco Gaussiano aditivo.

$X_K(t)$ é a matriz de fontes de sinal.

$a(\theta)$ é o vetor diretor.

N é o número de *Snapshots*.

Y_N é o vetor de dados (*snapshots*) na saída do arranjo (sinal + ruído).

H é o operador Hermitiano (conjugado trasposto).

$*$ é o operador conjugado.

W^H é o conjugado trasposto (Hermitiano) do vetor de pesos.

w_M* é o conjugado do peso M-ésimo peso do arranjo.

μ é a média de uma distribuição de dados.

σ^2 é a variância de uma distribuição de dados.

ω_c é a frequência temporal do filtro FIR.

T é o atraso imposto pelo filtro FIR.

σ_n^2 é a variância do ruído branco Gaussiano aditivo.

Z_{MN} é a matriz de snapshots do arranjo de sensores.

R é a sequência de referência.

$r(t)$ é o sinal de referência.

R_{zz} é a matriz de correlação dos sinais induzidos nos sensores.

R_{xx} é a matriz de correlação dos sinais gerados pelas fontes.

P_{zr} é a matriz de correlação cruzada dos sinais induzidos nos sensores do arranjo e o sinal de referência estimado.

I_M é a matriz identidade de dimensão MxM.

$P(W)$ é o espectro espacial de potência.

R_{nn} é a matriz de correlação do ruído.

$E\{\bullet\}$ é o operador esperança matemática.

W_{NAMI} é o vetor de pesos do estimador NAMI.

W_{SPNMI} é o vetor de pesos do estimador SPNMI.

p_c é a probabilidade de cruzamento do algoritmo genético.

p_m é a probabilidade de mutação do algoritmo genético.

$g_k(t)$ é a sequência de espalhamento da k-ésima fonte.

Lista de Figuras

Figura 2.1: Ilustração de um arranjo linear uniforme (ULA) com distância entre elementos igual a d. 30
Figura 2.2: Representação bidimensional de uma frente de onda incidindo em um arranjo de dois sensores. 31
Figura 2.3: Filtro temporal FIR de atrasos constantes de valor T. 34
Figura 2.4: Transmissor de Sequência Direta BPSK. 37
Figura 2.5: Diagrama de blocos do receptor CDMA proposto. 40
Figura 2.6: Resultado da correlação temporal para a detecção de atrasos. 42
Figura 3.1: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de −20 dB. 50
Figura 3.2: Representação em coordenadas polares do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de −20 dB. 50
Figura 3.3: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR de −20 dB. 51
Figura 3.4: Representação em coordenadas polares do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR −20 dB. 51
Figura 3.5: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB. 52
Figura 3.6: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB. 53
Figura 3.7: Esquema geral de um algoritmo genético em pseudocódigo. 56
Figura 3.8: Ilustração das denominações comuns para os algoritmos genéticos. 56
Figura 3.9: Representação do cruzamento uniforme para codificação binaria. 58
Figura 3.10: Representação da mutação para codificação binária. 59
Figura 3.11: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de −20 dB. 60

Figura 3.12: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR −20 dB.....60
Figura 3.13: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB.....61
Figura 3.14: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB.....61
Figura 4.1: Pseudocódigo do AGC.....65
Figura 4.2: Pseudocódigo da família de AGC's.....66
Figura 4.3: Modificação do critério de parada.....67
Figura 4.4: Mutação Binaria.....68
Figura 4.5: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 5°, 40° e 60°, usuário de interesse localizado em 5°, com SNR de −10 dB.....69
Figura 4.6: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 5°, 10° e 60°, usuário de interesse localizado em 5°, com SNR de −10 dB.....70
Figura 5.1: Erro quadrático Médio em função da relação Sinal-Ruído, do bloco sincronizador, para diferentes quantidades de usuários descorrelacionados.....73
Figura 5.2: Erro quadrático Médio em função da relação Sinal-Ruído, do bloco sincronizador, para diferentes quantidades de usuários totalmente correlacionados.....74
Figura 5.3: Taxas BER do bloco Estimador de Bit para diferentes quantidades de assinantes descorrelacionados.....75
Figura 5.4: Taxas BER do bloco Estimador de Bit para diferentes quantidades de assinantes totalmente correlacionados.....76
Figura 5.5: Testes de desempenho de direcionamento do Algoritmo Genético Compacto segundo o tamanho da população e a Relação Sinal-Rugido para três fontes estáticas descorrelacionadas localizadas em 10°, 40° e 60°.....77
Figura 5.6: Tempos médios de execução das provas de desempenho de direcionamento do Algoritmo Genético Compacto segundo o tamanho da população e a Relação Sinal-Ruído para três fontes estáticas descorrelacionadas localizadas em 10°, 40° e 60°.....78
Figura 5.7. Convergência do Erro da família de AGC's e do algoritmo LMS para três fontes descorrelacionadas localizadas em 10º, 40º e 60º.....79
Figura 5.8: RMSE do bloco Algoritmo Adaptativo para uma fonte descorrelacionada localizada em 10°.....80
Figura 5.9: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para uma fonte descorrelacionada localizada em 10°.....82

Figura 5.10: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para uma fonte descorrelacionada localizada em 10°. 82
Figura 5.11: RMSE do bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°. 83
Figura 5.12: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°. 84
Figura 5.13: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°. 85
Figura 5.14: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°. 85
Figura 5.15: RMSE do bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e -20°. 86
Figura 5.16: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e –20°. 86
Figura 5.17: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e –20°. 87
Figura 5.18: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e –20°. 87
Figura 5.19: RMSE do bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°. 88
Figura 5.20: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°. 88
Figura 5.21: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°. 89
Figura 5.22: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°. 89
Figura 5.23: RMSE do bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°. 91
Figura 5.24: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°. 91
Figura 5.25: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°. 92
Figura 5.26: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°. 92
Figura 5.27: RMSE do bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°. 93
Figura 5.28: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°. 93
Figura 5.29: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°. 94

Figura 5.30: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°.....94
Figura 5.31: RMSE do bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°.....95
Figura 5.32: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°. 96
Figura 5.33: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°.....96
Figura 5.34: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°.....97
Figura 5.35: Rastreamento da fonte de interesse em movimento por parte do algoritmo LMS com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°.....99
Figura 5.36: Resultado da saída da Relação Sinal-Interferência (SIR) do algoritmo LMS durante o rastreamento da fonte de interesse.....99
Figura 5.37: Rastreamento da fonte de interesse em movimento por parte do AGC com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°.....100
Figura 5.38: Resultado da saída da Relação Sinal-Interferência (SIR) do AGC durante o rastreamento da fonte de interesse.....100
Figura 5.39: Rastreamento da fonte de interesse em movimento por parte do AGCe com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°. 101
Figura 5.40: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCe durante o rastreamento da fonte de interesse.....102
Figura 5.41: Rastreamento da fonte de interesse em movimento por parte do AGCem com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°. 102
Figura 5.42: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCem durante o rastreamento da fonte de interesse.....103
Figura 5.43: Rastreamento da fonte de interesse em movimento por parte do AGCemr com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°. 103
Figura 5.44: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCemr durante o rastreamento da fonte de interesse.....104
Figura 5.45: Rastreamento da fonte de interesse em movimento por parte do Algoritmo LMS com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°. 105
Figura 5.46: Resultado da saída da Relação Sinal-Interferência (SIR) do Algoritmo LMS durante o rastreamento da fonte de interesse.....105
Figura 5.47: Rastreamento da fonte de interesse em movimento por parte do AGC com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°. 106
Figura 5.48: Resultado da saída da Relação Sinal-Interferência (SIR) do AGC durante o rastreamento da fonte de interesse.....106
Figura 5.49: Rastreamento da fonte de interesse em movimento por parte do AGCe com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°.....107
Figura 5.50: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCe durante o rastreamento da fonte de interesse.....107

Figura 5.51: Rastreamento da fonte de interesse em movimento por parte do AGCem com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°..... 108
Figura 5.52: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCem durante o rastreamento da fonte de interesse..... 108
Figura 5.53: Rastreamento da fonte de interesse em movimento por parte do AGCemr com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°..... 109
Figura 5.54: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCemr durante o rastreamento da fonte de interesse..... 109

Lista de Tabelas

Tabela 3.1: Termos Usados pelos Algoritmos Evolutivos (JAVIER, 2009).....55
Tabela 4.1: Membros da família de AGC's.....65

1 Introdução

A crescente demanda de dispositivos de comunicação móvel e de serviços de comunicação sem fio tem gerado uma superexploração dos recursos dos sistemas transmissores, provocando má qualidade dos serviços prestados. Para solucionar isso são implementadas novas técnicas de processamento de sinais que permitem aumentar a capacidade do canal de comunicação, evitar a interferência entre os usuários e aumentar a resistência ao ruído. Recentemente, as antenas inteligentes têm sido uma boa opção para mitigar a deficiência dos sistemas, trazendo segundo (COOPER, 2003) benefícios como melhor administração da energia de transmissão na rádio-base, diminuição da interferência e aumento da área de cobertura.

As antenas inteligentes ou arranjos de antenas (*Smart Antennas*) são conjuntos de sensores (antenas) distribuídos geometricamente e igualmente espaçados que atendem acertadamente os processos de transmissão e recepção (FERREIRA, 2005). As vantagens do arranjo de antenas quando comparado a uma antena omnidirecional é que ele permite estimar o ângulo de incidência da fonte de interesse (DOA, do inglês *Direction-Of-Arrival*), além de conformar os feixes de todos os sensores do arranjo (do inglês, *beamforming*) para apontá-lo na direção determinada e atenuar a interferência proveniente de outras fontes, ao invés de irradiar potência para todas direções (FERREIRA, 2005). Segundo (LIBERTI e RAPPAPORT, 1999 e Silva, 2009), as antenas inteligentes aplicadas nas comunicações móveis têm seis benefícios:

- Aumento na área de cobertura e melhor penetração em construções.
- Os custos iniciais de implementação num sistema sem-fio podem ser reduzidos.
- Aumento na robustez a perturbações no sistema e menor sensibilidade a falhas.
- Aumento da qualidade do *enlace* graças ao controle dos sinais multipercursos.
- Separa espacialmente os sinais, para permitir que diferentes assinantes compartilhem os mesmos recursos espectrais simultaneamente.
- Aumento da capacidade do sistema.

As antenas inteligentes encontram muitas áreas de aplicação, além das comunicações móveis celulares, como na tomografia, processamento de sinais biomédicos, geofísica e sistemas militares.

1.1 Motivação

Para estimar o DOA, direcionar o feixe na posição desejada e pôr nulos nas direções interferentes, o arranjo requer um processador digital de sinais (DSP, do inglês *Digital Signal Processor*). A etapa do DSP é realmente a parte inteligente do arranjo de sensores, porque é ele que fornece toda a informação dos sinais incidentes e adapta os parâmetros para que o arranjo irradie a potência segundo as suas indicações.

Existem diferentes métodos de controle para os arranjos, sendo que alguns deles fazem a estimação cega do DOA sem aproveitar as características próprias dos sinais recebidos. Exemplos desses métodos são o MUSIC (*Multiple Signal Characterization*) (SCHMIDT, 1979), Root-MUSIC (BARABELL, 1983), MODE (*Method Of Direction Estimation*) (ESTOICA e SHARMAN, 1990), MODEX (*MODE with eXtra roots*) (GERSHMAN e STOICA, 1999), e SEAD (*SEArch of Direction by differential spectrum*) (FERREIRA, 2005) que apresentam resultados satisfatórios mesmo para condições severas de Relação Sinal-Ruído (SNR, do inglês *Signal-to-Noise Ratio*), mas não são práticos para aplicações em tempo real devido a seus altos custos computacionais (SILVA, 2009). Por outro lado, métodos adaptativos como o algoritmo da Mínima Média Quadrático (LMS, do inglês Least Mean Square) usado por (PUTTINI, 2006), (ISLAM e MISRAN, 2008) e (MAOBING, WEI e CANHUI, 2010) aproveitam as características dos sinais recebidos, como por exemplo, o tipo de codificação, para estimar o DOA e direcionar o feixe. Estes métodos apresentam custos computacionais mais baixos em relação àqueles dos métodos de estimação cega, mas incorrem num alto risco de cair em mínimos locais, levando a falsas estimativas que entregam pontos não-ótimos tais que a taxa de erro não diminui.

Seguindo outra linha de pesquisa, em (VITALE, VESENTINI, AHMAD e HANZO, 2002) é proposta uma técnica não-convencional de *beamforming* assistido por um Algoritmo Genético (AG), conseguindo resultados muito satisfatórios quanto à relação Sinal-Interferência (SIR, do inglês *Signal-to-Interference Ratio*) e a relação Sinal-Interferência mais Ruído (SIRN, do inglês *Signal-to-Intreference plus Noise Ratio*). Similarmente, existem

outros trabalhos como os de (WOLFGANG, AHMAD, CHEN e HANZO, 2004) e (NOURI, NOORI e BAKHSHI, 2011) que também conseguem resultados bons e interessantes mas, conhecendo o alto custo computacional dos AG's, a aplicabilidade deles na estimação do DOA e *beamforming* em tempo real fica limitada.

Em (HARIK, LOBO e GOLDBERG, 1999) é desenvolvido o Algoritmo Genético Compacto (AGC), que é um algoritmo de baixo custo computacional que compartilha as mesmas virtudes dos AG convencionais. Além disso, em (GALLAGHER, VIGRAHAM e KRAMER, 2004) foram feitas uma série de modificações do AGC com o propósito de torná-lo mais robusto e aumentar sua aplicabilidade para problemas de grande porte.

Contudo, como o AGC é uma ferramenta de otimização global que até agora não foi implementada no controle dos arranjos de sensores, por sua característica da baixa complexidade computacional, propõe-se usá-lo para estimar o DOA e promover o *beamforming*.

1.2 Objetivo Geral

O alvo principal desta pesquisa é desenvolver a família de Algoritmos Genéticos Compactos (AGC's) e adaptá-la para o controle de um receptor adaptativo, com emprego de um arranjo linear de sensores, para ser aplicado em sistemas CDMA. Visando conferir a eficácia dos modelos propostos, o desempenho dos mesmos é comparado àquele do algoritmo da Mínima Média Quadratica (LMS).

1.3 Objetivos Específicos

Podem-se dividir os objetivos deste trabalho nos seguintes tópicos:

- Estudar e determinar teoricamente o problema da estimação de ângulos DOA e *beamforming*;
- Estudar a família de AGC's e reproduzir sua versão original;
- Adaptar a família de AGC's para o problema da estimação de ângulos DOA e *beamforming*;

- Reproduzir o receptor adaptativo baseado em CDMA usado por (PUTTINI, 2006);
- Avaliar o desempenho da família de AGC's e do algoritmo LMS com respeito à exatidão no direcionamento do feixe do arranjo;
- Estimar os tempos gastos para cada um dos algoritmos ao estimar o DOA e conformar o feixe.
- Avaliar estatisticamente o desempenho de estimação e o tempo gasto pelos algoritmos LMS e AGC.

1.4 Organização dos Capítulos

Visando deixar claro cada um dos assuntos abordados, esta dissertação encontra-se organizada em seis capítulos.

O capítulo 1 é usado para fins de introdução nos assuntos que foram abordados e descrever a organização deste documento.

O capítulo 2 descreve os princípios básicos do funcionamento de um arranjo linear uniforme (ULA, do Inglês *Uniform Linear Array*) adaptativo assim como os modelos de sinal e ruído com sua correspondente descrição matemática, além de apresentar algumas definições sobre as técnicas de acesso múltiplo, e uma descrição geral do receptor proposto.

No capítulo 3 são apresentadas algumas definições sobre sistemas adaptativos e a fundamentação teórica do algoritmo de Mínima Média Quadrática (LMS) e do Algoritmo Genético, centrando toda a informação na modelagem do emprego de algoritmos adaptativos para aplicação em arranjos de antenas.

O capítulo 4 expõe a fundamentação teórica do Algoritmo Genético Compacto (AGC) convencional, junto com a descrição de todas as modificações que foram no mesmo para gerar a família dos AGC's. Além disso são expostas as transformações feitas para adaptá-lo ao controle de arranjos de antenas.

No capítulo 5 são apresentados e comparados os resultados dos métodos abordados nesta dissertação, tanto para desempenho na estimação do DOA e *beamforming* como para a complexidade computacional envolvida. Além disso, desenvolve-se uma análise sobre os

resultados obtidos no rastreamento de fontes móveis para determinar o método que apresenta melhor comportamento.

Finalmente, o capítulo 6 traz as conclusões deste trabalho, com as sugestões para trabalhos futuros.

2 Modelo de Sinal e Antenas Inteligentes

2.1 Introdução

A utilização de antenas inteligentes traz grandes benefícios para as comunicações sem fio, daí a motivação de compreender seu funcionamento nos aspectos relacionados com a transmissão e recepção de sinais eletromagnéticos (FERREIRA, 2005).

Além disso, é importante fazer a modelagem matemática dos sinais que incidem no arranjo para poder relacionar suas frequências com os ângulos de incidência (SILVA, 2009), e poder determinar informação sob a localização das fontes.

2.2 Fundamentos de Arranjos de Antenas

Segundo (PUTTINI, 2006) os arranjos de antenas são um conjunto de antenas ou sensores distribuídos espacialmente em uma determinada geometria, que oferece na saída uma combinação dos sinais induzidos em cada um de seus elementos. Os sensores que conformam o arranjo são antenas omnidirecionais de baixo ganho (diretividade) e que estão conectadas por uma rede que combina os sinais de cada sensor, obtendo alta diretividade (LIBERTI e RAPPAPORT, 1999).

O termo antena adaptativa ou antena inteligente é definido segundo (LIBERTI e RAPPAPORT, 1999) como arranjos de sensores que podem ajustar dinamicamente seu padrão de irradiação para diminuir os níveis de ruído, mitigar interferências, combater os efeitos de multicaminho, e rastrear usuários em movimento. Deve-se deixar claro que o termo antena inteligente refere-se não apenas ao arranjo de sensores, mas sim a todo um sistema composto por conversores Analógico-Digital e Digital-Analógico, por um Processador Digital de Sinais (DSP) e pelo arranjo de sensores. O DSP analisa a informação fornecida pelo arranjo para determinar a posição das fontes e entrega os parâmetros para que finalmente o arranjo irradie o padrão de radiação na direção desejada.

As antenas inteligentes podem dispor seus sensores em qualquer distribuição geométrica, sendo a linear e a circular as mais usadas por motivos de simplicidade. A distribuição linear tem maior simplicidade de equacionamento, mas possui ambiguidade. A distribuição circular não possui ambiguidade, mas tem uma maior complexidade em seu equacionamento.

Nesta dissertação será apenas abordada a distribuição linear uniforme (ULA, do inglês *Uniform Linear Array*), deixando claro que os resultados podem ser estendidos para qualquer outra distribuição.

2.3 Considerações para a Modelagem

O parâmetro de maior importância na implementação de arranjos de antenas é o espaçamento entre os sensores, porque dele depende o surgimento de lóbulos indesejáveis e, portanto, a recuperação dos sinais incidentes (PUTTINI, 2006). Os lóbulos indesejáveis surgem por causa do acoplamento mútuo entre os sensores do arranjo, que se faz mais forte quando a proximidade entre os sensores é maior. Para se assegurar de que não apareçam lóbulos indesejáveis, deve-se garantir um espaçamento d entre os elementos consecutivos do arranjo de pelos menos meio comprimento de onda ($\lambda/2$) (STUTZMAN e THIELE, 1981), lembrando que o mesmo é dado por (2.1):

$$\lambda = \frac{c}{f_c} \tag{2.1}$$

onde c é a velocidade da luz no vácuo 3×10^8 m/s e f_c a frequência da portadora para ondas eletromagnéticas propagando-se no vácuo. A consideração do espaçamento de meio comprimento de onda entre os sensores do arranjo é conhecida como a versão espacial do Teorema da Amostragem (STUTZMAN e THIELE, 1981).

Visando simplificar a análise dos arranjos de antenas, foram adotadas as mesmas simplificações usadas em (PUTTINI, 2006):

- O espaçamento entre os sensores do arranjo é pequeno o suficiente para considerar que todos eles recebem sinais de mesma amplitude, mas com fases diferentes, além de que não há diferenças no ângulo de incidência das ondas em diferentes sensores do arranjo;

- Existe um número finito de sinais incidentes, ou seja, todos os sinais recebidos são compostos por um número finito de ondas planas.
- As fontes que originam os sinais encontram-se na região de campo distante, ou região de Fraunhofer (BALANIS, 1982), relativamente ao comprimento de onda, de tal forma que as ondas incidentes no arranjo são planas;
- Não existe acoplamento mútuo entre os elementos do arranjo;
- A largura de faixa do sinal incidente é pequena se comparada com a frequência da portadora, tratando-se, de sinais faixa estreita;
- Os sensores são antenas isotrópicas.

2.4 Equacionamento do Arranjo Linear Uniforme

Considere-se a incidência sobre um arranjo linear uniforme, composto por *M* sensores igualmente espaçados por uma distância *d*, de frentes de onda planas numa direção de chegada θ, emitidas por *K* fontes localizadas na região de campo distante, conforme é mostrado na Figura 2.1.

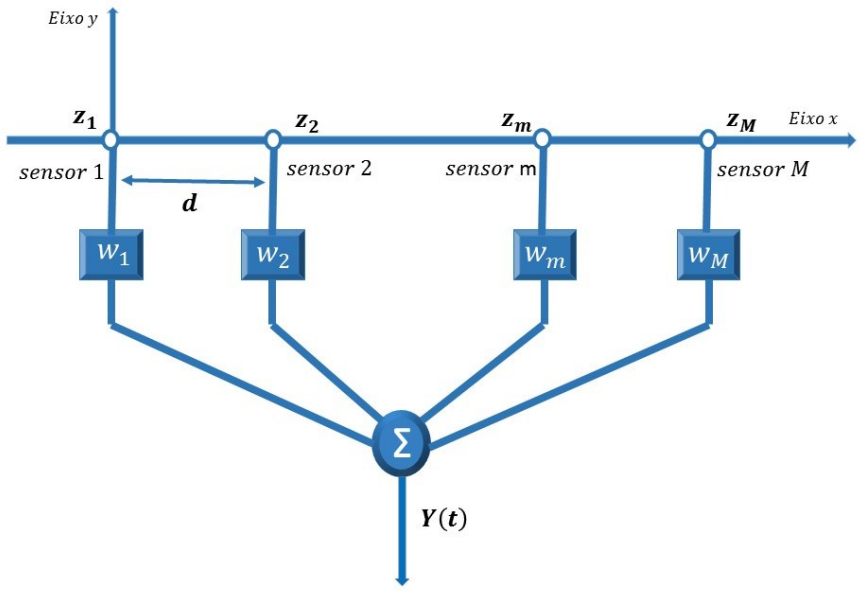

Figura 2.1: Ilustração de um arranjo linear uniforme (ULA) com distância entre elementos igual a *d*.

Analisando-se a Figura 2.2 pode-se determinar que a distância que separa duas frentes de onda que viajam com uma velocidade *c* incidentes em dois sensores consecutivos do arranjo é igual a $r = d.sen(\theta)$, ou seja, a frente de onda que incide no sensor 2 chega após percorrer uma distância *r* e com um atraso e defasagem dados por (2.2) e (2.3) respectivamente:

$$\tau = \frac{d \cdot \sin\theta}{c} \qquad (2.2)$$

$$\varphi = 2\pi \cdot \frac{d \cdot \sin\theta}{\lambda} \qquad (2.3)$$

onde o espaçamento *d* entre os sensores deve cumprir o critério da versão espacial do teorema de amostragem, ou seja, *d* tem que ser igual a $\lambda/2$ com a finalidade de evitar o acoplamento entre sensores.

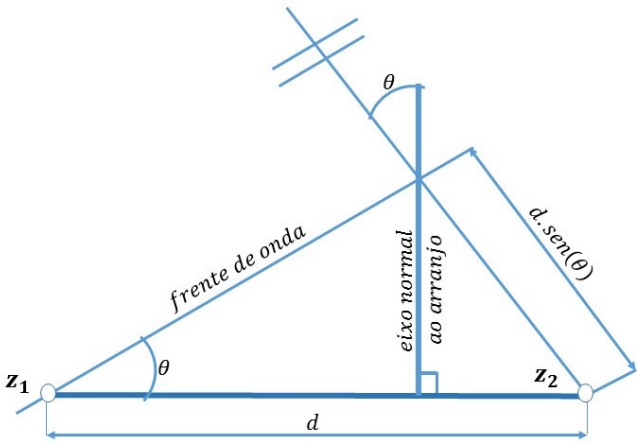

Figura 2.2: Representação bidimensional de uma frente de onda incidindo em um arranjo de dois sensores.

Note-se que na Figura 2.1 é mostrado o vetor de pesos $W = [w_1\ w_2\ \ldots\ w_M]^T$, no qual o peso w_M corresponde ao fator de ponderação associado ao *M*-ésimo sensor (PUTTINI, 2006).

Na modelagem é possível descrever os sinais induzidos no conjunto de sensores a partir do sinal induzido num deles, para tanto toma-se como referência o sensor designado pelo índice 1 (z_1). Agora, considerando φ como a defasagem imposta a dois sensores consecutivos quaisquer e sendo $x(t)$ o sinal induzido no sensor de referência, pode-se concluir

que o sinal induzido no sensor 2 (z_2) é uma versão defasada de $x(t)$, com envoltória complexa $e^{-j\varphi}$, ou seja, $x(t).e^{-j\varphi}$.

Para tornar mais realista a modelagem, considera-se que o sistema está sob a influência de ruído branco Gaussiano aditivo, próprio do meio de transmissão e que é inserido em cada sensor do arranjo. Considerando que $n_M(t)$ é o ruído referente ao M-ésimo sensor, que se trata de um arranjo linear uniforme e que o sistema está sob os efeitos de uma única fonte de sinal, as envoltórias podem ser modeladas pela seguinte equação (PUTTINI, 2006).

$$\begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_M(t) \end{bmatrix} = \begin{bmatrix} 1 \\ e^{-j\varphi} \\ \vdots \\ e^{-j(M-1)\varphi} \end{bmatrix} \cdot x(t) + \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_M(t) \end{bmatrix} \quad (2.4)$$

Para generalizar a expressão (2.4), considera-se o sistema sob a influência de K fontes de sinal, conforme é exposto em (2.5), onde $x_K(t)$ corresponde ao sinal recebido pelo sensor 1 e originado pela K-ésima fonte.

$$\begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_M(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ e^{-j\varphi_1} & e^{-j\varphi_2} & \cdots & e^{-j\varphi_K} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-j(M-1)\varphi_1} & e^{-j(M-1)\varphi_2} & \cdots & e^{-j(M-1)\varphi_K} \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_K(t) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ n_2(t) \\ \vdots \\ n_M(t) \end{bmatrix} \quad (2.5)$$

A expressão (2.5) também pode ser escrita como indica (2.6):

$$Z_M(t) = A_{MK}(\theta) \cdot X_K(t) + N_M(t) \quad (2.6)$$

onde a matriz $A_{MK}(\theta)$ é conhecida como matriz de direção (SILVA, 2009), e representa a forma como o arranjo responde para uma dada direção θ (PUTTINI, 2006). A matriz $A_{MK}(\theta)$ possui estrutura de Vandermonde (SILVA, 2009), com colunas linearmente independentes e posto cheio, igual ao número de sinais, caso os sinais incidentes sejam totalmente independentes (ALVES, 2004). Segundo (PUTTINI, 2006) cada coluna da matriz de direção é usualmente chamada vetor diretor (do inglês, *steering vectors*). A matriz de direção pode ser descrita conforme (2.7), (2.8) e (2.9), designando o vetor referente ao sinal da k-ésima fonte por $a(\theta_k)$ (PUTTINI, 2006).

$$A(\theta) = [a(\theta_1) \quad a(\theta_2) \quad \ldots \quad a(\theta_K)] \quad (2.7)$$

$$\theta = [\theta_1 \quad \theta_2 \quad \ldots \quad \theta_K]^T \quad (2.8)$$

$$a(\theta_K) = [1 \quad e^{-j\varphi_K} \quad \ldots \quad e^{-j(M-1)\varphi_K}]^T \quad (2.9)$$

Para fazer qualquer processamento é preciso primeiro ter a versão discreta dos sinais induzidos nos sensores, por isso os sinais recebidos são amostrados considerando um

conversor analógico-digital na saída de cada sensor. Segundo (PUTTINI, 2006), o conjunto de vetores de dados resultantes da amostragem ocorrida em cada elemento do arranjo, num certo instante de tempo, recebe o nome de matriz de *Snapshots*. As expressões que descrevem um arranjo linear uniforme com *N Snapshots* são dadas por (2.10) e (2.11).

$$\begin{bmatrix} z_1(1) \dots z_1(N) \\ z_2(1) \cdots z_2(N) \\ \vdots \quad \ddots \quad \vdots \\ z_M(1) \cdots z_M(N) \end{bmatrix} = A_{MK}(\theta). \begin{bmatrix} x_1(1) \dots x_1(N) \\ x_2(1) \cdots x_2(N) \\ \vdots \quad \ddots \quad \vdots \\ x_k(1) \cdots x_k(N) \end{bmatrix} + \begin{bmatrix} n_1(1) \dots n_1(N) \\ n_2(1) \cdots n_2(N) \\ \vdots \quad \ddots \quad \vdots \\ n_M(1) \cdots n_M(N) \end{bmatrix} \quad (2.10)$$

$$Z_{MN} = A_{MK}(\theta). X_{KN} + N_{MN} \quad (2.11)$$

Segundo (PUTTINI, 2006), no modelo apresentado cada um dos sensores recebe sinais de amplitude unitária, com direção de chegada θ e com ganho $a(\theta)$. Para originar a saída do arranjo faz-se uma combinação linear dos sinais induzidos nos sensores do arranjo, ponderada pelo vetor de pesos *W*, daí que a modelagem do arranjo seja feita em função da defasagem φ e do vetor de pesos *W* conforme é expresso em (2.12).

$$Y_N = W^H. Z_{MN} = \sum_{i=1}^{M} w_i^*. z_i(t) = z_1(t). \sum_{i=1}^{M} w_i^* . e^{-j(i-1)\varphi} \quad (2.12)$$

Por meio da alteração dos pesos, o arranjo consegue modificar o diagrama de radiação. O controle dos pesos é feito através de algoritmos adaptativos que fazem com que o arranjo trabalhe como um filtro espacial, beneficiando uma determinada posição de interesse e atenuando a influência das fontes interferentes.

2.5 Analogia entre ULA e Filtros FIR

Segundo (PUTTINI, 2006), existe uma semelhança entre o modelo proposto para os filtros espaciais com arranjos de antenas e os filtros temporais FIR. Na expressão (2.13) apresenta-se a saída do filtro temporal FIR de atrasos constantes *nT* da Figura 2.3.

$$y(nT) = W^H \cdot z(nT) = \sum_{i=1}^{M} w_i^* \cdot z_i(nT) = z_1(t) \cdot \sum_{i=1}^{M} w_i^* \cdot e^{-j \cdot w_c (i-1) \cdot nT} \quad (2.13)$$

É possível ter uma analogia completa dos filtros FIR e dos arranjos lineares uniformes de sensores comparando as expressões (2.12) e (2.13), considerando que o filtro espacial encontra-se sincronizado para uma única portadora f_c, e além disso, fazendo-se a

correspondência entre a frequência temporal ω_c e o atraso T do filtro FIR com o seno da direção de chegada e o espaçamento d dos sensores do arranjo respectivamente (PUTTINI, 2006). Com essa analogia, as antenas adaptativas podem usar técnicas desenvolvidas para os filtros FIR.

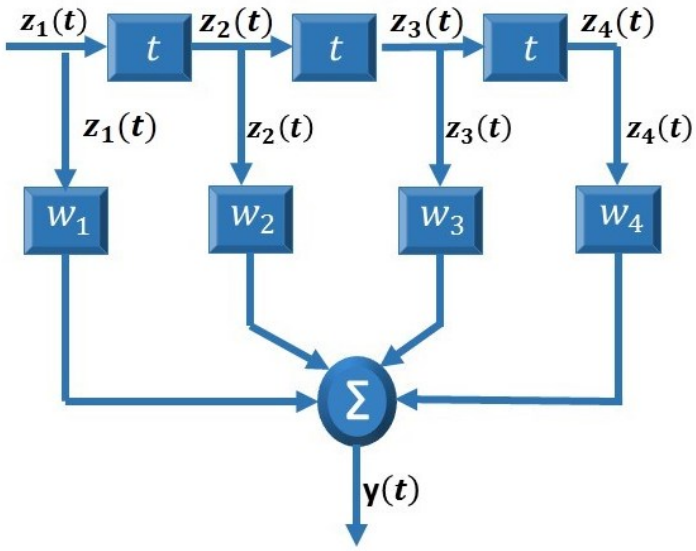

Figura 2.3: Filtro temporal FIR de atrasos constantes de valor T.

2.6 Outras considerações sobre o sinal

Considerando que a antena inteligente faz o processamento sobre a versão discreta dos sinais recebidos e pela natureza estocástica dos sinais induzidos nos sensores, a modelagem do sistema é feita através de variáveis estatísticas. Os processos estocásticos não podem ser descritos através de uma função ou expressão matemática determinística. Em vez disso são representados por meio de funções de densidade de probabilidade (PDF, do inglês *Probability Density Function*), onde parâmetros como a média μ e a variância σ^2 têm maior importância, já que eles descrevem o nível ao redor do qual flutua o processo e a medida de espalhamento do processo ao redor da média, respectivamente. Para simplificar mais a modelagem, os processos são considerados estacionários em sentido amplo, o que significa que têm média e variância constante e valor quadrático médio finito (PUTTINI, 2006).

Os conformadores adaptativos requerem um sinal de referência para fazer a estimação. Esse sinal pode ser estimado supondo-se que o modelo é ergódigo, o que quer dizer que o modelo é um processo estocástico estacionário no sentido amplo com estimadores não

polarizados e onde a média converge para a solução ótima quando a quantidade de amostras tende a infinito (HAYKIN, 2002).

A matriz de correlação dos sinais induzidos nos sensores R_{zz}, a matriz de correlação dos sinais gerados pelas fontes R_{xx}, a matriz de correlação cruzada dos sinais induzidos nos sensores e o sinal de referência P_{zr}, são de muita importância para a estimação. Agora, tendo definido a Z_{MN} em 2.10 e supondo uma sequência de referência R, produto da amostragem do sinal de referencia $r(t)$, e considerando a variância do ruído branco igual a σ_n^2, usam-se os *snapshots* para determinar as matrizes conforme as expressões (2.14), (2.15) e (2.16).

$$R_{zz} = E\{Z_{MN} \cdot Z_{MN}^H\} = A_{MK} \cdot R_{xx} \cdot A_{MK}^H + \sigma_n^2 \cdot I_M \cong \frac{1}{N} \cdot Z_{MN} \cdot Z_{MN}^H \quad (2.14)$$

$$R_{xx} = E\{X \cdot X^H\} \quad (2.15)$$

$$P_{zr} = E\{Z_{MN} \cdot r(t)^H\} \cong \frac{1}{N} \cdot Z_{MN} \cdot R^H \quad (2.16)$$

Modelando os sinais induzidos nos sensores como processos estacionários de média nula e para um determinado vetor de pesos W, a potência média espectral ou espectro espacial de potência pode ser expresso como em (2.17), que geralmente é determinada em função das direções de cobertura da antena (PUTTINI, 2006).

$$P(W) = E\{y(t) \cdot y(t)^*\} = W^H \cdot R_{zz} \cdot W \quad (2.17)$$

2.7 Acesso Múltiplo em Sistemas de Comunicação

Quando se fala de Acesso Múltiplo refere-se à capacidade de um sistema de comunicação de transmitir diversos sinais simultaneamente pelo mesmo canal de comunicação. Existem quatro domínios sobre os quais se pode implementar o acesso múltiplo de usuários, os quais são: frequência, tempo, espaço e código (PUTTINI, 2006).

A técnica de Acesso Múltiplo por Divisão em Frequência (FDMA, do inglês *Frequency Division Multiple Access*) é vastamente usada em sistemas de comunicação analógicos, onde se garante a ortogonalidade dos sinais atribuindo a cada usuário uma faixa de frequência permanente, ou seja, o canal de comunicação pode conter múltiplos sinais separados espectralmente (SKLAR, 2001), daí a limitação que encontra esta técnica na utilização do espectro de frequências, já que restringe o uso a uma determinada quantidade de usuários em função da largura de faixa exigida para cada um deles.

No caso do Acesso Múltiplo por Divisão no Tempo (TDMA, do inglês *Time Division Multiple Access*) a ortogonalidade dos sinais é garantida atribuindo a cada um dos usuários todo o espectro durante um curto período de tempo chamado *time slot* (SKLAR, 2001). Esta técnica é amplamente usada em sistemas de comunicação sem fio como o *Global System for Mobile Communications* (GSM), AMPS digital (IS - 136) e *Personal Access Communications System* (PACS) (PUTTINI, 2006).

O Acesso Múltiplo por Divisão no Espaço (SDMA, do inglês *Space Division Multiple Access*) é uma técnica própria dos arranjos de antenas e dos satélites, onde as múltiplas antenas que os compõem permitem acesso simultâneo de fontes localizadas em regiões diferentes. No caso das antenas inteligentes, são exploradas as características espaciais dos sinais induzidos nos sensores para apontar o diagrama de radiação na direção da fonte de interesse e impor nulos nas direções das fontes interferentes.

Graças à introdução dos dispositivos eletrônicos digitais nos sistemas de comunicação, o Acesso Múltiplo por Divisão no Código (CDMA, do inglês *Code Division Multiple Access*) tornou-se uma técnica amplamente utilizada devido a usar modulação para espalhar o sinal sobre todo o espectro de frequência, aumentando a capacidade de utilização do canal, reduzindo a interferência e acrescentando a imunidade ao ruído.

Nos sistemas de comunicação é possível empregar mais de uma técnica de acesso múltiplo, como por exemplo, nesta dissertação é usada a técnica SDMA aplicada pelo arranjo de antenas que recebe sinais com espalhamento espectral por sequência direta (DS-SS, do inglês *Direct Sequence Spread Spectrum*).

2.7.1 Espalhamento Espectral

Como foi dito na seção anterior o Espalhamento Espectral (SS, do inglês *Spread Spectrum*) é uma técnica de modulação que "espalha" o sinal que vai ser transmitido ao longo de todo o espectro de frequências ou em uma faixa muito longa de frequências, apresentando benefícios como supressão da interferência, diminuição da potência e claro o múltiplo acesso. As técnicas de espalhamento espectral estão classificadas em duas categorias: Salto em Frequência (FH, do inglês *Frequency Hopping*) e Sequência Direta (DS, do inglês *Direct Sequence*) que respectivamente geram diferentes tipos de codificação como CDMA por salto

em frequência (FH-CDMA) e CDMA por sequência direta (DS-CDMA). No desenvolvimento desta dissertação, apenas o DS-CDMA foi considerado.

2.7.1.1 Espalhamento Espectral por Sequencia Direta (DS-SS)

No espalhamento espectral por sequencia direta, o sinal que contém a informação $x(t)$ de largura de faixa B_1 é multiplicado por um Sinal ou Código de Espalhamento $g(t)$ composto por símbolos chamados *chips* e de largura de faixa B_2 para obter uma versão espalhada do sinal $x(t)$. Considerando que o sinal $x(t)$ é um sinal de faixa estreita comparada com a largura de faixa do código de espalhamento $g(t)$ ($B_2 >> B_1$), a multiplicação do sinal e do código de espalhamento $x(t)g(t)$ entrega um sinal de largura de faixa aproximadamente igual à largura de faixa do código de espalhamento B_2, conforme demostra (SKLAR, 2001).

Os sinais ou códigos de espalhamento são sequências pseudoaleatórias com densidade espectral similar à do ruído branco de fundo, e podem ser geradas de modos diferentes, como por exemplo, as sequências de Walsh aplicadas nesta dissertação. As sequências de espalhamento são amostradas com taxas muito maiores que as taxas da sequência de dados original. Essas taxas estão diretamente relacionadas com o período dos *chips* T_c e com o período dos bits T_b, e a razão entre estes períodos determinam o ganho do processo de codificação ou Fator de Espalhamento (SF, do inglês *Spread Factor*), ou seja, $SF = T_b/T_c$.

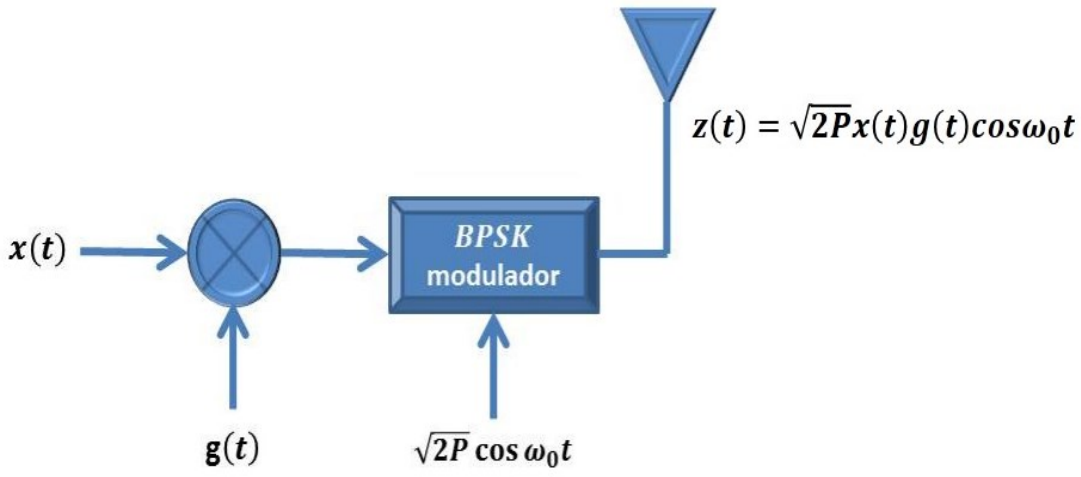

Figura 2.4: Transmissor de Sequência Direta BPSK.

No processo de transmissão conforme é mostrado na Figura 2.4 primeiro é espalhado o sinal para logo ser modulado em BPSK (*Binary Phase-Shift Key*) de envoltória constante, frequência em radianos ω_0 e potência P. O sinal transmitido pelo meio é dado por (2.18) considerando uma portadora ideal.

$$z(t) = \sqrt{2P}x(t)g(t)\cos\omega_0 t \tag{2.18}$$

No receptor o sinal *z(t)* é recebido com influência de ruído branco gaussiano aditivo $n(t)$ e com atraso conforme mostra expressão (2.19), lembrado que $\omega_0 = 2\pi f_c$, e sendo f_c a frequência em Hertz da portadora, A o ganho do sinal e φ o ângulo de fase aleatório no intervalo (0, 2π).

$$z(t-\tau_0) = A\sqrt{2P}x(t-\tau_0)g(t-\tau_0)\cos[2\pi f_c(t-\tau_0)+\varphi] + n(t) \tag{2.19}$$

Para recuperar o sinal original no receptor é necessário gerar uma réplica da sequência de espalhamento com certa defasagem τ_0, considerando o atraso com que é recebido o sinal $z(t)$. O alvo dessa defasagem é sincronizar o sinal incidente com a sequência de espalhamento, já que só é possível recuperar a mensagem com $z(t)$ e $g(t)$ sincronizados por causa da ortogonalidade que existe entre os códigos de cada usuário, conforme se expõe em (2.20).

$$g_k(t)\cdot g_h(t) = \begin{cases} 0, se\ k \neq h \\ 1, se\ k = h \end{cases} \tag{2.20}$$

No receptor, logo de fazer o desespalhamento do sinal é feita a demodulação, conseguindo assim a estimativa do sinal $z(t)$, e cumprindo exatamente o processo inverso feito pelo transmissor. Nas expressões (2.21) e (2.22) é mostrado o desespalhamento do sinal incidente no receptor, onde vale a pena ressaltar que a influência do ruído e a interferência são anuladas graças à ortogonalidade das sequências de espalhamento.

$$z(t-\tau_0)g(t-\tau_0) = A\sqrt{2P}x(t-\tau_0)g(t-\tau_0)g(t-\tau_0)\cos[2\pi f_c(t-\tau_0)+\varphi] + n(t)g(t-\tau_0) \tag{2.21}$$

$$z(t-\tau_0) = A\sqrt{2P}x(t-\tau_0)\cos[2\pi f_c(t-\tau_0)+\varphi] \tag{2.22}$$

Finalmente o sinal obtido logo da demodulação é uma versão atrasada e atenuada do sinal $x(t)$, mas no receptor já é conhecido o atraso graças ao sincronizador, daí que simplesmente se retira o atraso e amplifica o sinal demodulado.

Nesta dissertação é feita só uma pequena introdução dos sistemas CDMA, já que esta é uma técnica amplamente conhecida e muito aplicada, mas pode-se encontrar maior informação em (SKLAR, 2001), (TORRIERI, 2005), (ZIGANGIROV, 2004) e (VITERBI, 1995).

2.7.1.2 Códigos de Walsh

Os códigos de Walsh são sequências binarias ortogonais geradas usando a matriz de Hadamard conforme mostra (2.23), e utilizadas para separar usuários que compartilham um mesmo canal ou para separar canais dentro de uma célula (ZOOGHBY, 2005).

$$W_{2N} = \begin{bmatrix} W_N & W_N \\ W_N & -W_N \end{bmatrix} \quad (2.23)$$

Os códigos de Walsh têm um comprimento de 2^N, onde N é um número inteiro. O comprimento representa a quantidade de usuários por canal. Supondo que N é igual 6, as dimensões da matriz de códigos é de 64x64, o que significa que cada código de Walsh conta com 64 *chips* e que em esse canal só podem ser alocados 64 usuários.

Para garantir que os códigos de Walsh sejam aplicados com sucesso, tem que ser garantida a sincronização, com uma precisão de pelo menos um período de *chip* T_c.

2.8 Modelo do Receptor Proposto

A Figura 2.5 apresenta o diagrama de blocos do receptor proposto, deixando claro que é uma versão simplificada e não foram descritos elementos como sintonizador de frequência, modulador e demodulador, amplificadores de baixo ruído e conversores Analógico – Digital. As simplificações feitas no receptor e o fato que o sistema opera em banda base não restringem os resultados das simulações, já que em condições reais de operação o modelo integrará um sistema de recepção muito mais robusto.

O sistema proposto está integrado por um arranjo de antenas linear e uniforme composto por sensores isotrópicos que recepciona os sinais dos usuários alocados no canal. O sinal $z(t)$ que incide no arranjo é composto pela superposição dos sinais emitidos pelos usuários e é induzido em cada sensor do arranjo com uma pequena defasagem. Todos os elementos do arranjo são ponderados pelo vetor de pesos W, o que significa que cada sensor tem um peso w_M associado, que faz possível a separação espacial desejada.

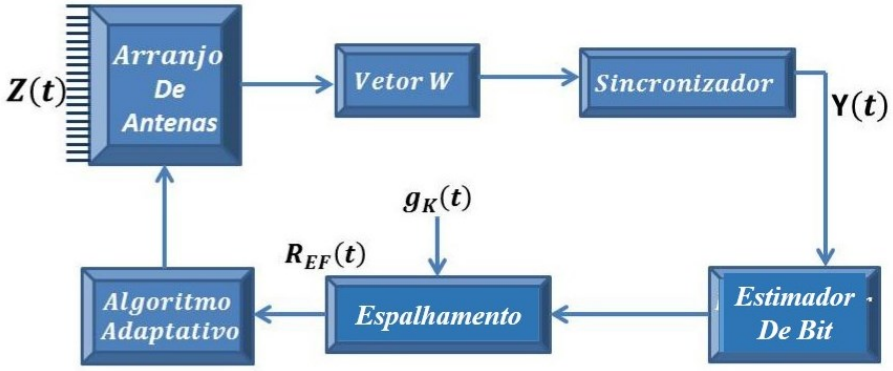

Figura 2.5: Diagrama de blocos do receptor CDMA proposto.

Conforme foi mencionado anteriormente, a sincronização entre a sequência de espalhamento e o sinal de interesse deve ser garantida para poder recuperar a informação que contém, daí que o sincronismo torna-se um fator crítico nos sistemas CDMA. Prevendo isso, o modelo proposto conta com um bloco sincronizador encarregado de detectar os atrasos dos sinais incidentes e de alinhar estes com as sequências de espalhamento.

O sincronizador proposto é só um correlator temporal que estima o valor absoluto da correlação entre a média dos sinais induzidos nos sensores e o código de espalhamento que atua sobre o sinal de interesse. O sincronizador primeiro estima a média de cada determinada quantidade de amostras dos dados induzidos nos sensores para reduzir a influência do ruído, logo estima a correlação com a sequência de Walsh que espalhou o sinal de interesse.

Nesta dissertação os sinais sem espalhamento têm um comprimento de 10 bits, os códigos de Walsh têm comprimento de 64 *chips* e o sinal recebido pelo receptor tem um comprimento de 6400 amostras por causa do espalhamento e da taxa de amostragem. Como foram quantizadas 10 amostras por bit, a média do sinal é feito cada 10 amostras obtendo um vetor de médias de 640 posições com o qual se calcula o valor absoluto da correlação de cada 64 amostras com o código de Walsh. O valor absoluto da correlação de cada bloco de 64 amostras com o código de espalhamento tem um comprimento de 128 posições. Para aumentar a resolução da estimativa do valor absoluto da correlação se sumam as primeiras 64 posições com as segundas 64 posições ficando um vetor de só 64 posições, como todo o processo é repetido para cada grupo de 64 posições, cada estimativa do valor absoluto da correlação vai ter um vetor de 64 posições. Os vetores resultantes são alocados nas linhas de uma matriz, representando o valor absoluto da correlação de cada bit de informação com o código de espalhamento. Posteriormente, com cada coluna dessa matriz é estimada a média,

obtendo um vetor de 64 posições, onde o máximo valor de pico desse vetor representa o atraso do sinal. Porém, como o atraso tem que ser determinado em termos dos chips, multiplica-se a localização do maior pico pela quantidade de amostras por bit do sinal original.

A Figura 2.6 ilustra a média do valor absoluto da correlação temporal entre o código de espalhamento e o sinal recebido no pelo arranjo. A curva mostra o sincronismo entre o sinal espalhado e o código de espalhamento versus as posições dos atrasos. No caso da Figura 2.6, o maior pico do valor médio dos valores absolutos da correlação do sinal espalhado com o código de espalhamento está localizado na posição 23 mas, considerando que a taxa de amostragem é de 10 amostras por bit, a localização real do atraso é 230 *chips*.

No capítulo cinco são apresentados os resultados do desempenho do sincronizador proposto para diferentes valores de SNR e quantidade de usuários.

No sistema proposto, o vetor de pesos *W* é atualizado a cada 10 bits de informação, daí é necessário atualizar, também a cada 10 bits, o sinal de referência para apontar o feixe do arranjo quando a fonte do sinal de interesse esteja em movimento. A primeira estimação dos primeiros 10 bits é feita de maneira cega, já que ainda não se conhece a informação a respeito da localização da fonte, tal que o vetor *W* é inicializado com todos seus pesos iguais a um (*W = 1*), de forma que o arranjo inicie a busca apontando na direção de zero graus ($\theta = 0°$). Tão logo o Algoritmo Adaptativo convirja, o vetor de pesos é atualizado e o bloco Estimador processa os dados escolhidos para estimar outros 10 bits.

Graças às propriedades dos sistemas que usam espalhamento espectral, o receptor proposto dispõe de um bloco Estimador de Bit, encarregado de gerar uma sequência de bits espalhados que sirvam como sinal de referência para o bloco do Algoritmo Adaptativo.

O algoritmo implementado para a estimação dos bits consiste simplesmente em estimar a média aritmética de cada subconjunto de 640 amostras do conjunto de 6400 amostras recebidas pelo arranjo. As equações (2.24) e (2.25) descrevem a metodologia usada pelo estimador de bit. Este estimador de bit está baseado no estimador de bit desenvolvido por (PUTTINI, 2006). Os resultados do desempenho do estimador são expostos no capítulo cinco.

$$\mu_x = \frac{1}{640} \cdot \sum_{j=1}^{640} amostras(j) \qquad (2.24)$$

$$bit = \begin{cases} +1, se\ \mu_x \geq 0 \\ -1, se\ \mu_x < 0 \end{cases} \quad (2.25)$$

Finalmente, o Algoritmo Adaptativo é usado para o controle do arranjo de antenas, exatamente na estimação dos pesos do arranjo, adaptando-os dinamicamente de acordo com o sinal de referência estimado pelo bloco estimador de bit. Os algoritmos Adaptativos empregados foram o algoritmo LMS e a família de Algoritmos Genéticos Compactos (AGC's). Maiores detalhes da implementação e desempenho destes algoritmos serão expostos em capítulos subsequentes.

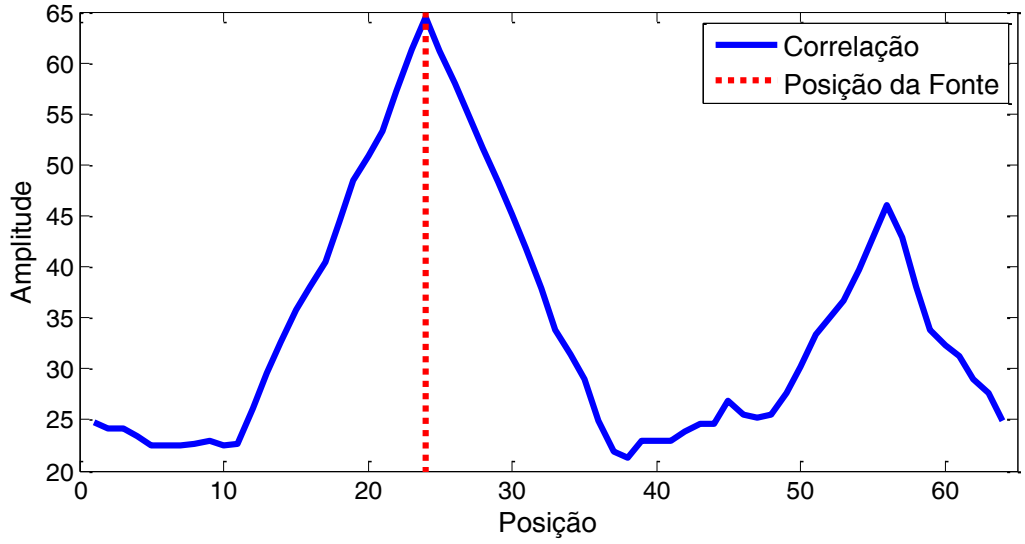

Figura 2.6: Resultado da correlação temporal para a detecção de atrasos.

2.9 Vantagens das Antenas Inteligentes

As antenas inteligentes têm a capacidade de separar espacialmente os sinais recebidos direcionando eletronicamente lóbulos de alto ganho para cada usuário de interesse e colocando nulos nas direções das fontes interferentes. Isso permite uma forma de compartilhamento no domínio espacial, ou seja, um Acesso Múltiplo por Divisão Espacial (SDMA, do inglês *Space-Division Multiple Access*), o qual representa uma das maiores vantagens dos arranjos de sensores e pelo qual podem ser chamados de filtros espaciais. Segundo (PUTTINI, 2006), os arranjos têm as seguintes vantagens:

- Funcionam para qualquer comprimento de onda, graças à mudança em tempo real dos parâmetros do arranjo feita pelos algoritmos adaptativos de controle.
- O desempenho dos sistemas de comunicação é otimizado, graças ao direcionamento dos feixes ser feito via *software*, o que implica que os lóbulos podem ser atribuídos em tempo real.
- Diminui os efeitos do desvanecimento por multipercurso, isso garantindo a ausência de correlação dos sinais induzidos nos sensores.
- A capacidade de discriminação espacial aumenta com o tamanho da abertura espacial do arranjo.

2.10 Tipos de Conformadores de Feixe

Segundo (VAN VEEN e BUCKLEY, 1988), os arranjos de sensores podem ser classificados em duas categorias: Independentes dos dados e Estatisticamente Ótimos (Adaptativos), onde a primeira classe faz referência aos arranjos que estabelecem pesos constantes aplicados nos sensores, dependendo da posição de interesse. Dentro deste grupo encontram-se o Conformador Clássico ou Convencional e o Conformador com Nulos Direcionais. Por outro lado, dentro do grupo dos conformadores Estatisticamente Ótimos encontram-se os Adaptadores em Bloco e os Adaptadores Contínuos.

Os Adaptadores em Bloco trabalham com uma determinada quantidade de dados ou *snapshots* para buscar a solução ótima dessas amostras. Neste grupo, enquadram-se o conformador Inverso-da-matriz-de-ruído (NAMI), inverso-da-matriz-de-sinal-mais-ruído (SPNMI) e o filtro de Wiener-Hopf (PUTTINI, 2006).

Os Adaptadores Contínuos estimam os pesos à medida que novos *snapshots* são coletados de modo que o vetor de pesos convirja na solução ótima (PUTTINI, 2006). Neste grupo, encontram-se o filtro de Mínima Média Quadrática (LMS) e o filtro de Mínima Média Quadrática Recursivo (RLS, do inglês *Recursive-Least Squares*).

Considerando que a classificação feita por (VAN VEEN e BUCKLEY, 1988), foi feita quando o desenvolvimento dos métodos heurísticos não era muito conhecido, vale a pena agregar uma nova categoria na classificação. Assim, nesta dissertação é agregada a classe dos

conformadores Heurísticos, que são métodos que fazem uma estimação a partir de estratégias evolutivas. Dentro desta classe podem ser enquadrados os Algoritmos Genéticos.

3 Algoritmos Adaptativos e Genéticos

3.1 Introdução

Os algoritmos adaptativos são algoritmos que, como seu nome indica, têm a capacidade de adaptar-se a diferentes circunstâncias do entorno no qual trabalham. Os algoritmos adaptativos modificam seu comportamento durante o tempo de execução para atender as mudanças produzidas no sistema e entregar a melhor resposta possível de acordo com o problema em estudo.

Por outro lado, o algoritmo Genético (AG) é uma ferramenta de otimização muito poderosa e eficiente, comumente aplicada em diversas áreas da engenharia. Apesar de seu amplo uso, ele pode apresentar alta complexidade computacional, dependendo da função que vai ser otimizada. Isto exige maiores tempos de execução ou demanda mais recursos de *hardware*.

De acordo com o apresentado no capítulo 2, os algoritmos adaptativos são altamente usados no problema de estimação da direção de chegada DOA e na conformação de feixe, daí a importância de conhecer mais a fundo seu comportamento. No caso dos algoritmos genéticos, existem trabalhos que utilizam os AG's na estimação do DOA como (YAN e ZHU, 2000), (LI e LU, 2002), (VITALE, VESENTINI, AHMAD e HANZO, 2002) e (WOLFGANG, AHMAD e HANZO, 2004), que propõem uma forma não convencional de tratar o problema do DOA e *beamforming*. Por tudo isso, é necessário conhecer a estrutura do AG para introduzir o leitor na proposta desta dissertação.

3.2 Algoritmos e Filtros Adaptativos para Arranjos de Antenas

Os algoritmos adaptativos são normalmente utilizados em comunicações para estimar os coeficientes dos filtros adaptativos e, no caso dos arranjos de sensores, graças à analogia com os filtros FIR, produzem filtros espaciais onde os coeficientes do filtro são os pesos complexos do arranjo. No filtro espacial se requer que os coeficientes sejam reajustados

automaticamente em cada iteração durante a fase de aprendizagem, enquanto as condições do meio mudam. Obviamente, a finalidade do ajuste dos pesos do arranjo é otimizar o filtragem espacial.

O objetivo dos algoritmos adaptativos é minimizar o erro quadrático médio (MSE, do inglês *Mean Squared Error*) entre um sinal de referência e um sinal gerado pelo algoritmo a partir dos pesos do arranjo e dos sinais induzidos nos sensores. A expressão que define o erro é dada por:

$$\varepsilon(t) = Y_N - r(t) = W^H \cdot Z_{MN} - r(t) \quad (3.1)$$

Segundo (PUTTINI, 2006), a resposta dos algoritmos adaptativos pode ser descrita pelas seguintes equações:

$$E\{|\varepsilon(t)|^2\} = E\{|Y_N - r(t)|^2\} = E\{|W^H \cdot Z_{MN} - r(t)|^2\} \quad (3.2)$$

$$MSE = E\{|\varepsilon(t)|^2\} = E\{|Y_N|^2 - 2 \cdot W^H \cdot Z_{MN} \cdot r(t)^H + |r(t)|^2\} \quad (3.3)$$

$$MSE = W^H \cdot R_{zz} \cdot W - 2 \cdot W^H \cdot E\{Z_{MN} \cdot r(t)^H\} + E\{|r(t)|^2\} \quad (3.4)$$

$$MSE = E\{|r(t)|^2\} + W^H \cdot R_{zz} \cdot W - 2 \cdot W^H \cdot P_{zr} \quad (3.5)$$

Os algoritmos usados em arranjos de sensores são normalmente divididos em dois tipos, os cegos e não-cegos. Os algoritmos cegos dependem das características próprias e conhecidas dos sinais recebidos para determinar a saída, ou seja, estes algoritmos tentam recuperar aquelas características dos sinais para produzir estimativas com taxas de erro baixas. Entre os algoritmos cegos encontram-se o algoritmo de módulo constante (CMA, do inglês *Constant Modulus Algorithm*) e o algoritmo de mínimos quadrados com módulo constante (LS-CMA, do inglês *Least-Square Constant Modulus Algorithm*). No caso dos estimadores não-cegos ou treinados, precisa-se de um sinal de referência recebido previamente ou estimado pelo próprio sistema para determinar a saída. Entre os algoritmos não-cegos enquadram-se o filtro de Wiener-Hopf, o algoritmo LMS e o algoritmo RLS.

No desenvolvimento desta dissertação, considerou-se apenas a reprodução do algoritmo LMS como algoritmo de referência para avaliar a desempenho dos novos métodos de estimação propostos.

3.2.1 Algoritmo de Mínima Média Quadrática LMS

O LMS pertence à família dos algoritmos com gradiente estocástico, também é conhecido como solução adaptativa do filtro de Wiener. A importância do algoritmo LMS reside na sua simplicidade, porque não precisa conhecer as matrizes de correlação dos dados (R_{zz} e P_{zr}) nem precisa fazer inversões de matriz. Em vez disso, ele faz a estimativa instantânea do gradiente com base em um sinal de erro que surge da comparação entre um sinal de referência e um sinal estimado pelo algoritmo. O algoritmo LMS é aplicável nos ULA graças à analogia destes com os filtros FIR, fazendo possível que as características do algoritmo LMS sejam aplicadas nos arranjos de sensores. O pseudocódigo do algoritmo LMS geral pode ser resumido por (HAYES, 1996):

$$W_0 = [0\ 0\ \ldots 0] \tag{3.6}$$

Para n = 0, 1, 2, 3, ...

$$y(n) = W_n^T \cdot X(n) \tag{3.7}$$

$$e(n) = d(n) - y(n) \tag{3.8}$$

$$W_{n+1} = W_n + \mu \cdot e(n)X^*(n) \tag{3.9}$$

sendo $e(n)$ o sinal de erro, $y(n)$ o sinal estimado pelo algoritmo, $d(n)$ o sinal de referência, W_n o vetor de coeficientes do filtro, W_0 o vetor inicial de coeficientes e μ o passo ou parâmetro de correção do algoritmo.

O algoritmo, a partir do seu vetor de coeficientes inicial W_0 e dos dados de entrada, estima o sinal $y(t)$ para compará-lo com o sinal de referência $d(t)$ e estabelecer o erro $e(t)$. Logo, atualiza-se o vetor de coeficientes do filtro com seu valor anterior, o erro $e(t)$ e o passo μ, repetindo-se este processo até se atender o critério de parada.

É preciso ressaltar a importância que tem o passo μ no desempenho do algoritmo, já que ele é o que gradua o valor segundo o qual o vetor de pesos será atualizado, portanto, ele está diretamente relacionado com a velocidade de convergência e a precisão do algoritmo. Um valor muito grande do μ inviabiliza a convergência do modelo, conseguindo erros de estimação altos, enquanto que um valor de μ muito pequeno pode provocar um aumento desnecessário de processamento. Segundo (HAYKIN, 2002), para garantir a convergência do algoritmo, o parâmetro μ deve cumprir a condição (3.10).

O vetor de coeficientes W_n normalmente é inicializado com todos seus valores iguais a zero (W_0 = [0 0 0 ... 0]) mas, sabendo-se que colocando valores diferentes de zero no vetor inicial de pesos o algoritmo converge mais rapidamente, pode-se inicializa-lo com os valores desejados.

$$0 < \mu < \frac{2}{\lambda_{max}} \tag{3.10}$$

No caso dos arranjos de antenas, é necessário fazer umas modificações no algoritmo LMS geral mostrado nas expressões (3.11), (3.12), (3.13) e (3.14), para poder atender os requisitos de bidimensionalidade do problema da estimação da DOA e *beamforming*. As equações que modelam o algoritmo LMS usado no controle de um arranjo linear uniforme de sensores aplicado nesta dissertação, estão baseadas no trabalho (PUTTINI, 2006) e são sumarizadas a seguir:

$$\widehat{\boldsymbol{w}}_0 = [1 \quad 1 \quad \ldots \quad 1] \tag{3.11}$$

Para $i = 0, 1, 2, 3, \ldots, n$

$$\boldsymbol{y}^i = \widehat{\boldsymbol{w}}_i^H . \boldsymbol{Z}^i \tag{3.12}$$

$$\boldsymbol{\varepsilon_i} = \boldsymbol{y}^i - \boldsymbol{r}_i \tag{3.13}$$

$$\widehat{\boldsymbol{w}}_{i+1} = \widehat{\boldsymbol{w}}_i + \mu . \boldsymbol{Z}^i . \boldsymbol{\varepsilon}_i^H \tag{3.14}$$

sendo $\widehat{\boldsymbol{w}}_0$ o vetor de pesos inicial do arranjo, $\boldsymbol{y^i}$ a sequência gerada pelo algoritmo na iteração i e de dimensão N, $\boldsymbol{r_i}$ a sequência de referência na iteração i, $\boldsymbol{Z^i}$ a matriz de *snapshots* na iteração i e de dimensões MxN, $\boldsymbol{\varepsilon_i}$ o vetor de erro na iteração i e μ o passo ou parâmetro de correção.

No algoritmo LMS usado no controle do ULA, troca-se o sinal de entrada pela matriz de *snapshots* dos sinais induzidos nos sensores, e no cálculo do erro é feita uma normalização da sequência gerada pelo algoritmo com a finalidade de aumentar a velocidade de convergência do algoritmo. A escolha do vetor inicial de pesos é de extrema importância no desempenho do algoritmo, já que quanto mais distante de seus valores ótimos leva mais tempo convergir até eles.

Normalmente, na prática o vetor de pesos é inicializado de tal forma que o arranjo de sensores aponte na direção de θ = 0°, o que significa que o vetor tem que ser inicializado com todos seus valores igual a um (W_0 = [1 1 1 ... 1]). Do mesmo modo que no algoritmo LMS geral, o parâmetro μ continua sendo de extrema importância na performance do algoritmo.

Portanto, a escolha de seu valor no caso de arranjos de antenas está restrita a (HAYKIN, 2002):

$$0 < \mu < \frac{1}{traço(R_{zz})} \quad (3.15)$$

Segundo (PUTTINI, 2006), cada elemento da diagonal de R_{zz} é igual à potência média no sensor correspondente, o que significa que o traço da matriz R_{zz} equivale à potência média medida em qualquer dos sensores vezes a quantidade de sensores do arranjo.

No algoritmo LMS aplicado no controle do ULA, o vetor de pesos é estimado por meio de um processo iterativo de treinamento e mantido constante durante a transmissão ou recepção da informação (PUTTINI, 2006). O tempo durante o qual o vetor de pesos é mantido constante equivale ao tempo necessário para amostrar uma matriz de *N snapshots*, já que na janela de tempo seguinte serão recebidos novos *snapshots* com informação de uma possível mudança de posição da fonte de interesse. Portanto os pesos serão atualizados a partir de seu valor anterior para a nova posição da fonte (PUTTINI, 2006).

Nas Figuras 3.1 e 3.2 é mostrado o espectro espacial de potência em coordenadas cartesianas e polares respectivamente de um ULA composto por dez sensores controlado pelo algoritmo LMS, que recebe sinais de quatro fontes descorrelacionadas que incidem com a mesma potência localizadas em –20°, 10°, 40° e 60° com uma SNR de –20 dB, sendo a fonte de interesse a localizada em 10°. É notável que o arranjo consegue apontar o maior lóbulo de seu diagrama de radiação na direção da fonte de interesse e não consegue anular o feixe nas direções das fontes interferentes. Porém, o sistema mostra um bom desempenho considerando a baixa SNR usada na simulação. Considerando-se outro caso em que no arranjo incidem sinais descorrelacionados de fontes localizadas em –20°, 10°, 15° e 60° com uma SNR de –20 dB, onde a fonte de interesse é de novo a fonte localizada em 10°. Nesta simulação existe uma fonte interferente muito próxima da fonte de interesse, o que dificulta o direcionamento do arranjo na direção desejada. Entretanto, o maior ganho é apontado na direção de interesse, apesar de beneficiar também o sinal da fonte interferente vizinha. Quanto às outras fontes interferentes, o algoritmo LMS não conseguiu posicionar nulos espectrais em suas direções, conforme mostram as Figuras 3.3 e 3.4.

Melhorando a relação Sinal-Ruído a 0 dB, o algoritmo LMS melhora seu desempenho, já que para fontes afastadas consegue apontar o maior ganho do arranjo na direção desejada e também atenuar em grande medida o ganho dos lóbulos que apontam nas direções

interferentes, A Figura 3.5 expõe o espectro espacial de potência de um ULA de dez sensores isotrópicos para as mesmas fontes descorrelacionadas afastadas, localizadas em –20°, 10°, 40° e 60°, sendo a fonte de 10° a fonte de interesse.

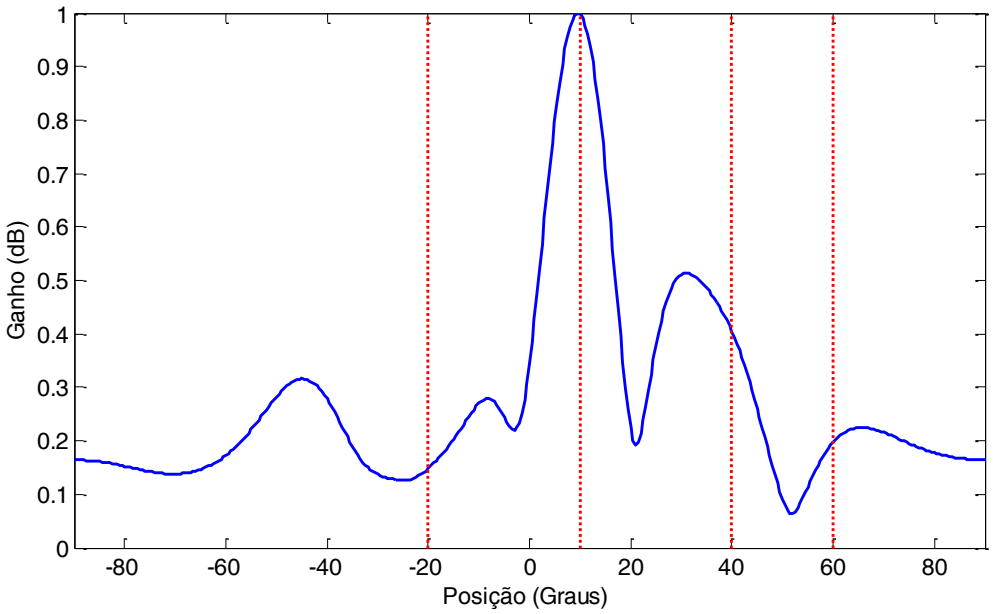

Figura 3.1: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de –20 dB.

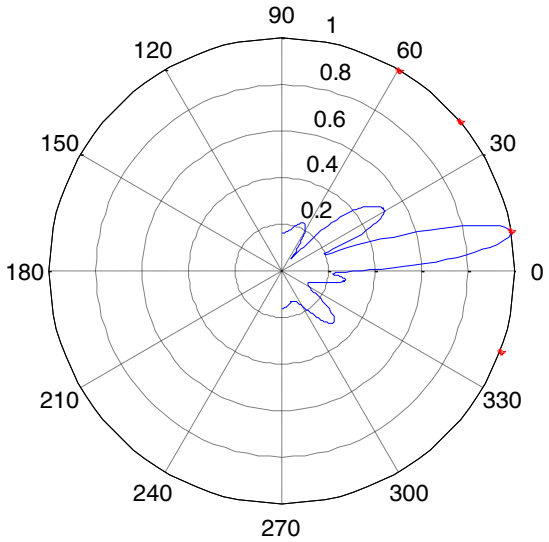

Figura 3.2: Representação em coordenadas polares do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de –20 dB.

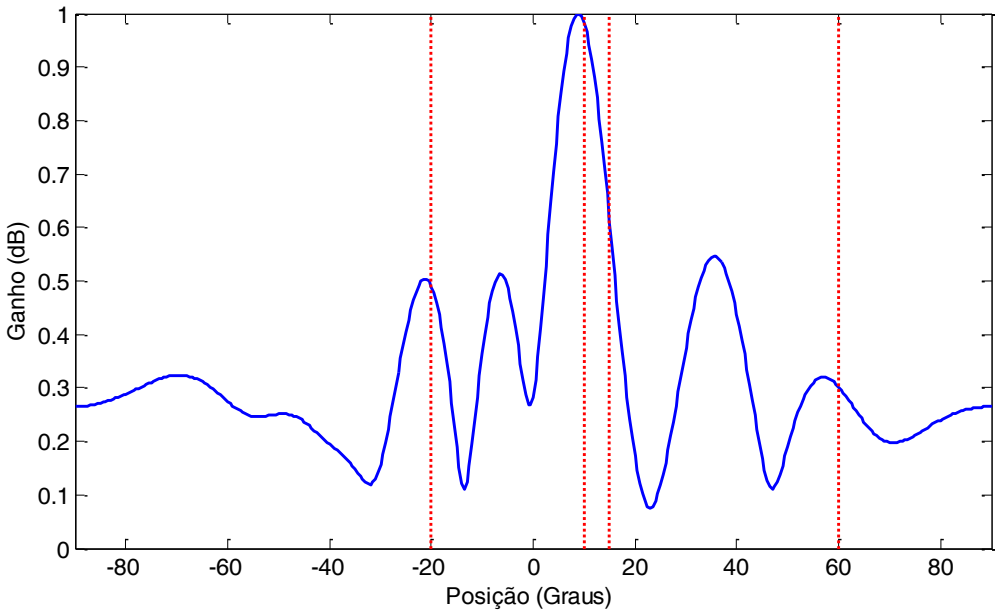

Figura 3.3: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR de −20 dB.

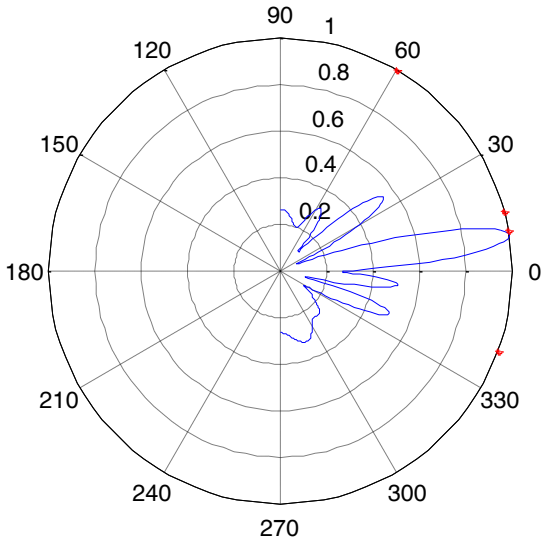

Figura 3.4: Representação em coordenadas polares do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR −20 dB.

Como pode ser percebido, o algoritmo LMS usado para estimação de fontes vizinhas (Figura 3.6) também apresenta uma grande melhoria no seu desempenho com o aumento na

SNR, conseguindo novamente apontar o lóbulo principal do arranjo na direção de interesse e atenuando os feixes que apontam para as fontes interferentes, com exceção da fonte interferente vizinha que é favorecida pelo feixe que aponta para fonte de interesse. A Figura 3.6 ilustra o comportamento do algoritmo LMS quando no canal há fontes descorrelacionadas vizinhas, que estão localizadas em −20°, 10°, 15° e 60° e que incidem em um ULA de dez sensores isotrópicos com SNR de 0 dB, a fonte de interesse é a fonte locada em 10°.

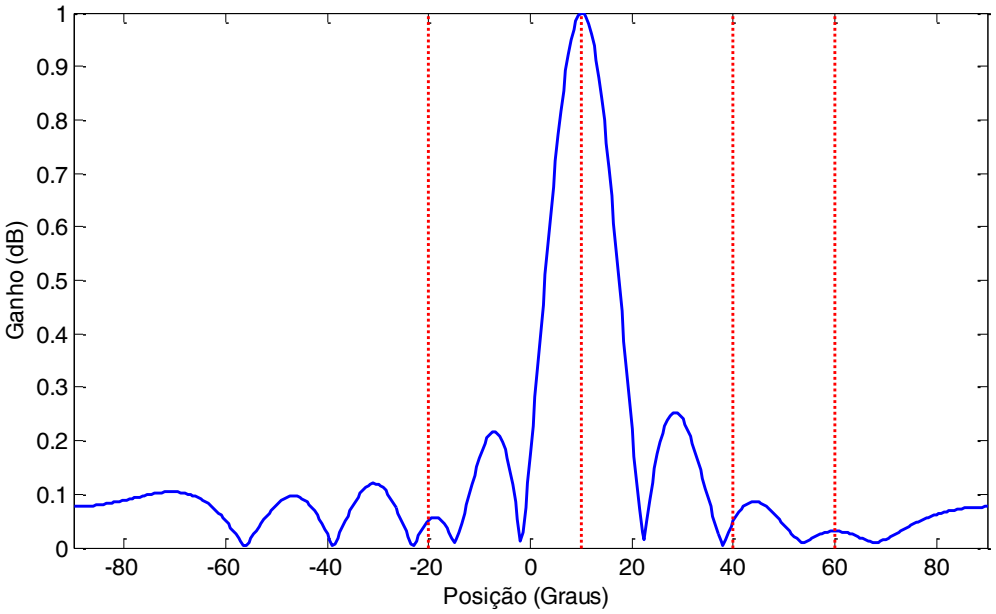

Figura 3.5: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB.

3.3 Algoritmos Bioinspirados

Os algoritmos bioinspirados são técnicas computacionais utilizadas na otimização de problemas de grande complexidade. Estes algoritmos são inspirados em processos naturais, daí que o objetivo destes algoritmos é modelar os processos naturais para adaptá-los computacionalmente e resolver problemas de otimização muito difíceis de modelar matematicamente.

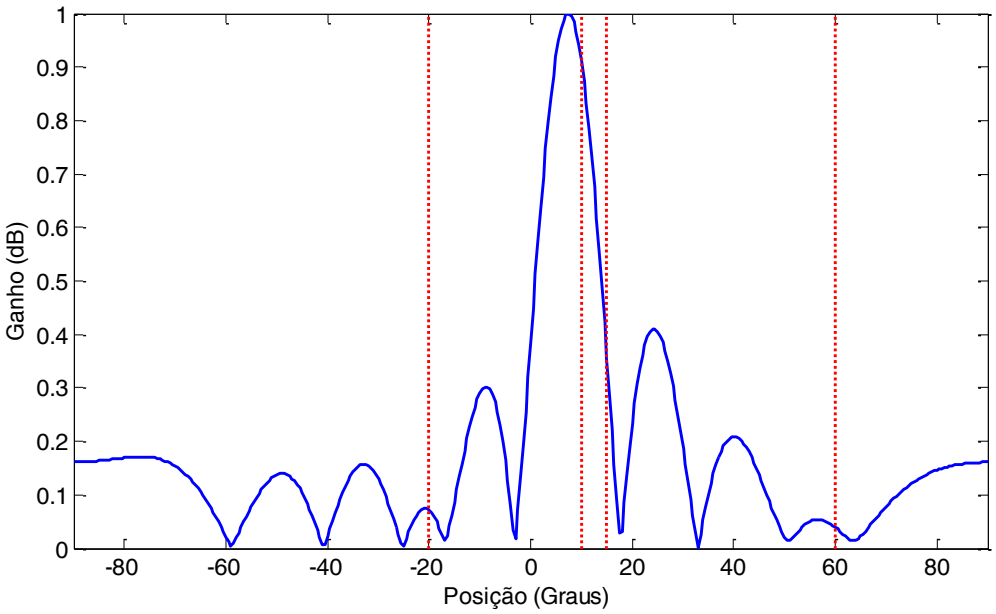

Figura 3.6: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo LMS, para quatro sinais descorrelacionadas de mesma potência localizadas em −20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB.

3.4 Computação Evolucionária

Segundo (SILVA, 2008) a Computação Evolucionária (CE) é um ramo da Inteligência Computacional (IC) que utiliza técnicas inspiradas na evolução natural, ou seja, são processos adaptativos que imitam processos seletivos naturais e fundamentam-se na sobrevivência do melhor indivíduo que representa uma solução potencial do problema tratado (JAVIER, 2009). Os métodos usados na CE são conhecidos como Algoritmos Evolutivos (AE) e estes algoritmos trabalham sobre populações que evoluem de geração em geração através de operadores genéticos adaptados ao problema (JAVIER, 2009 *apud* HIDALGO e CERVIGÓN, 2002).

Dentre a computação evolucionária existem diferentes tipos de algoritmos evolutivos, mas o princípio do seu funcionamento é o mesmo para todos. O princípio dirige os algoritmos evolutivos é a seleção natural, onde uma população de indivíduos é submetida à influência de um ambiente no qual sobrevivem apenas os mais aptos. Logo, os indivíduos sobreviventes originam uma nova população com melhores características para adaptar-se às condições do seu *habitat*, ou seja, cada nova geração é composta por indivíduos mais fortes ou mais aptos. Em termos de programação, dada uma determinada função para ser maximizada ou

minimizada (ambiente), pode-se gerar aleatoriamente um conjunto de possíveis soluções (população) e aplicá-las a função objetivo como medida de *fitness*. Baseados nesse *fitness*, alguns dos melhores candidatos são selecionados para passar à seguinte geração e, a partir deles, gerar uma nova população. A nova população é gerada aplicando-se operadores de recombinação e/ou mutação. A recombinação é um operador genético onde dois ou mais candidatos, chamados pais, são selecionados para gerar uma determinada quantidade de novos candidatos ou filhos. Já a mutação se aplica só em um candidato, mudando uma ou várias das suas características para originar um novo candidato. Aplicando-se os operadores genéticos, produz-se um novo conjunto de candidatos que competem, com base no seu *fitness* e possivelmente na sua idade, com os candidatos antigos por um lugar na geração seguinte (JAVIER, 2009). Este é um processo interativo que finaliza quando o melhor candidato é escolhido ou quando são cumpridos os critérios de parada.

Nos algoritmos evolutivos, muitos de seus processos são estocásticos. Durante o processo de seleção, os melhores indivíduos têm uma probabilidade maior de serem escolhidos que os menos aptos, mas até os indivíduos mais fracos têm a possibilidade de chegarem a ser pais (JAVIER, 2009). No caso da mutação acontece o mesmo, já que as partes que serão alteradas para produzir um indivíduo são escolhidas aleatoriamente, bem como as novas partes que vão substitui-as.

Na computação evolucionária são usados continuamente termos da biologia, por sua inspiração na natureza. A Tabela 3.1 apresenta os termos mais usados nesta área junto com seu correspondente conceito computacional.

3.4.1 Algoritmos Genéticos

Os Algoritmos Genéticos (AG's) são algoritmos evolutivos inspirados nos mecanismos de seleção natural e na genética (SILVA, 2008), e são considerados como algoritmos de busca heurística.

TERMO	CONCEPTO COMPUTACIONAL (AE)
Cromossomo	É uma estrutura de dados que contém uma cadeia de parâmetros de desenho ou genes. Esta estrutura de dados pode ser representada, por exemplo, por uma cadeia de bits ou um arranjo de inteiros.
Gene	É uma subseção de um cromossomo que usualmente codifica o valor de um único parâmetro.
Genótipo	Codificação que pode ser binária ou de outro tipo dos parâmetros que representam uma solução do problema a resolver.
Fenótipo	É a decodificação do cromossomo, ou seja, os valores obtidos depois de passar da representação codificada à usada pela função objetivo.
Individuo	É um membro único da população de potenciais soluções do problema. Cada indivíduo contém um cromossomo que representa uma possível solução do problema.
Fitness	Valor que é atribuído a cada indivíduo e que indica quão bom ele é se comparado com os demais.
Geração	Cada iteração da medida de *fitness* e criação de uma nova população por meio de operadores de reprodução.
Operador Genético	É o mecanismo que influencia a forma como é passada a informação genética dos pais para os filhos.
Cruzamento	Operador que forma um novo cromossomo combinando partes de cada um dos cromossomos de seus pais.
Mutação	Operador que forma um novo cromossomo através de alterações dos valores dos genes de um só cromossomo pai.
Elitismo	Mecanismo utilizado para garantir que os cromossomos dos membros mais aptos de uma população passem para a seguinte geração sem serem alterados pelos operadores genéticos.

Tabela 3.1: Termos Usados pelos Algoritmos Evolutivos (JAVIER, 2009).

Segundo (SILVA, 2008) um AG é uma simulação computacional de um modelo de um sistema natural, onde os indivíduos de uma população evoluem no recorrer das gerações através de um processo repetitivo de seleção e reprodução. Nos AG's a seleção é feita por

métodos probabilísticos e a reprodução pelos operadores de cruzamento sexual e mutação, onde o operador principal é o cruzamento e o secundário é a mutação. O pseudocódigo de um AG é mostrado na Figura 3.7.

1. Gera população Inicial aleatoriamente.
   $pop \coloneqq gera_pop(l, n);$
2. Calcular o fitness da população.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ n\ \boldsymbol{do}$
   $fitness(i) \coloneqq avalia_fitness(pop(i));$
3. Selecionar probabilisticamente com base no fitness os pais.
   $pais \coloneqq seleção(fitness);$
4. Gera seguinte população.
   $nova_pop \coloneqq cruzamento_mutação(pais);$
5. Verificar a convergência do algoritmo.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ n\ \boldsymbol{do}$
   $\boldsymbol{if}\ fitness(i) = valor_desejado\ \boldsymbol{do}$
   $\boldsymbol{Stop};$
   $\boldsymbol{else\ do}$
   $\boldsymbol{return\ to}\ step\ 2;$

Parâmetros do AG.

n: $tamanho\ da\ população$

l: $longitude\ do\ cromossomo$

Figura 3.7: Esquema geral de um algoritmo genético em pseudocódigo.

Nos AG, os indivíduos também podem ser chamados fenótipos e contar com mais de um cromossomo. Os cromossomos formam os chamados genomas ou genótipos, que simplesmente são uma representação de uma solução. Normalmente, os cromossomos usam codificação binária, mas podem usar qualquer outra codificação. Na codificação binária cada bit do cromossomo é chamado alelo. As variáveis do problema são representadas por conjuntos de alelos, onde estes conjuntos de alelos são conhecidos como genes. A Figura 3.8 ilustra os conceitos anteriores.

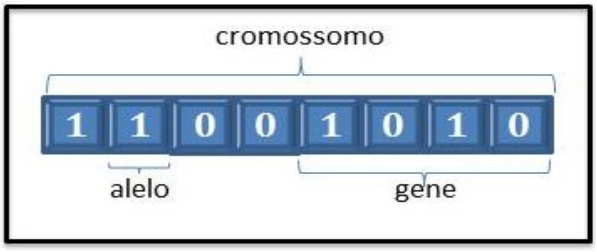

Figura 3.8: Ilustração das denominações comuns para os algoritmos genéticos.

Os algoritmos genéticos iniciam com uma população de indivíduos gerados aleatoriamente, que logo são avaliados para determinar os indivíduos pais que vão se reproduzir para gerar uma nova população. Conforme foi dito anteriormente, a seleção dos indivíduos pais obedece ao princípio da seleção natural ou princípio Darwiniano da sobrevivência do melhor adaptado. Portanto, os indivíduos mais fortes têm maior probabilidade de passar à seguinte geração. A aptidão dos indivíduos é medida por meio da função que se quer otimizar, daí esta função ser conhecida como função de *fitness* ou função objetivo. Os indivíduos selecionados como pais se reproduzirão através de operadores genéticos como cruzamento (*crossover*) e mutação. Todo este processo é iterativo e termina só quando o melhor indivíduo é encontrado ou quando a quantidade de gerações máximas estimadas pelo programador for atingida. A Figura 3.7 expõe o pseudocódigo do processo evolutivo dos AG's.

O processo de seleção é fundamental nos AG's porque com ele são eleitos os indivíduos que produzirão a nova geração. Normalmente, a seleção é do tipo probabilística, onde todos os indivíduos inclusive os menos aptos têm a possibilidade de serem pais. Por outro lado, na seleção de tipo extintiva, unicamente os indivíduos mais aptos têm a oportunidade de serem pais. Existem três grandes classes de seleção: seleção proporcional, seleção mediante torneio e seleção de estado uniforme. Dentro de algumas delas existem diferentes métodos. Na seleção proporcional, por exemplo, encontram-se os métodos da roleta, sobrante estocástico, universal estocástica e amostragem determinística. Dentro da seleção diante torneio encontram-se as técnicas determinísticas e probabilísticas. Existem outras técnicas de seleção pouco usadas como: a seleção disruptiva, a seleção por hierarquias não-lineares e a seleção competitiva.

O cruzamento é o processo mediante o qual novos indivíduos são criados a partir da informação contida em dois ou mais pais. O operador de cruzamento é considerado como a principal característica dos algoritmos genéticos. Geralmente, este operador é empregado de forma probabilística com uma percentagem de cruzamento p_c, este valor comumente está entre os limites de 0.5 e 1.0 (JAVIER, 2009). Quando os pais são escolhidos, é gerado um número aleatório entre 0.0 e 1.0, e logo se determina se este valor é superior que o limiar p_c para realizar o cruzamento, caso contrário é usado outro método para gerar novos indivíduos. Existem diferentes técnicas para efetuar o cruzamento como o *cruzamento de um ponto* ou *cruzamento de n-pontos* entre outros.

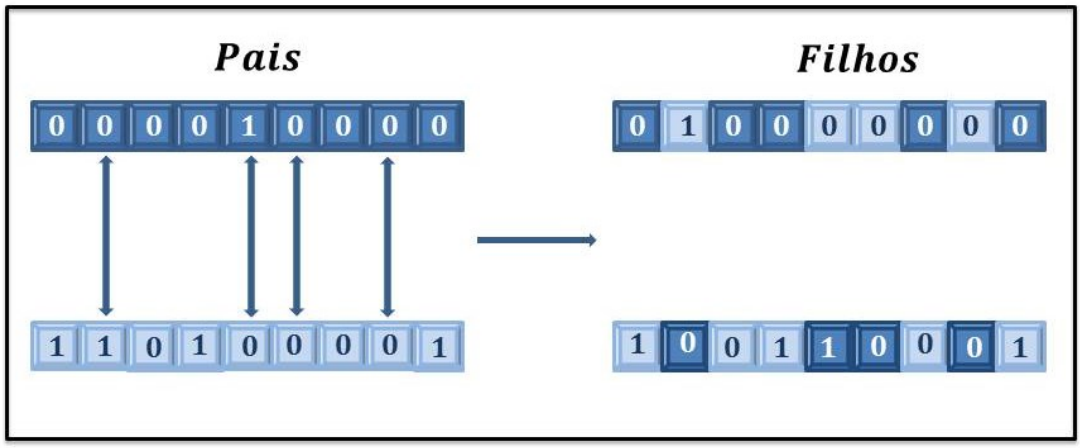

Figura 3.9: Representação do cruzamento uniforme para codificação binaria.

A mutação é um operador genético que a partir de um único pai gera um filho graças as modificações aleatórias feitas sobre seu próprio genótipo ou representação codificada. Assim como no cruzamento, a mutação usa uma probabilidade de mutação p_m para definir se a mutação é usada ou não. A mutação é muito usada em codificação binária, mas pode ser usada em qualquer tipo de codificação.

Segundo (JAVIER, 2009 *apud* GOLDBERG, 1989), os algoritmos genéticos são diferentes dos algoritmos tradicionais em quatro aspectos:

- Baseiam-se em uma codificação dos parâmetros de otimização, e não dos parâmetros em si;
- A busca é feita sobre uma população de soluções, e não sobre uma solução única;
- Não necessitam de nenhum conhecimento derivado do problema, apenas de uma forma de avaliação do resultado;
- Usam transições probabilísticas e não determinísticas.

Na Figura 3.11 é mostrado o espectro espacial de potência em coordenadas cartesianas de um ULA composto por dez sensores isotrópicos e controlado pelo algoritmo Genético, que recebe sinais de quatro fontes descorrelacionadas, incidentes com a mesma potência e localizadas em –20°, 10°, 40° e 60° com uma SNR de –20 dB, sendo a fonte de interesse a localizada em 10°. Observa-se que o arranjo aponta o maior lóbulo de seu diagrama de

radiação na direção da fonte de interesse, porém com resolução menor que aquela obtida com o algoritmo LMS. Além disso, não consegue impor nulos para as fontes interferentes, mas, em geral o comportamento do AG é aceitável para fontes afastadas considerando a baixa SNR.

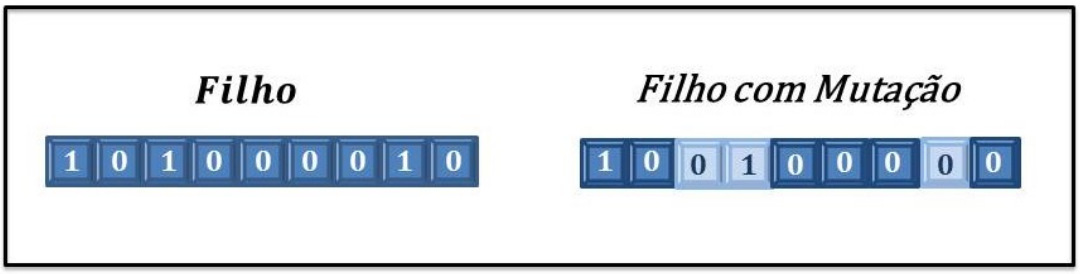

Figura 3.10: Representação da mutação para codificação binária.

Agora, considere-se que no ULA incidem sinais descorrelacionadas de fontes localizadas em –20°, 10°, 15° e 60° com uma SNR de –20 dB, onde a fonte de interesse é de novo a fonte localizada em 10°. No canal existe uma fonte interferente vizinha da fonte de interesse que dificulta o direcionamento do arranjo na direção desejada. O resultado da Figura 3.12 mostra que o AG não consegue anular a fonte interferente vizinha pois, apesar de apontar seu feixe na direção da fonte de interesse, a baixa resolução posiciona o lóbulo principal também na fonte interferente vizinha e produz lóbulos indesejáveis de alto ganho que irradiam potência desnecessariamente.

Aumentando a SNR até 0 dB, o arranjo controlado pelo AG apresenta melhor desempenho para fontes afastadas, já conseguindo apontar seu feixe na direção desejada e reduzindo o feixe que aponta para as fontes interferentes, tal que o lóbulo principal tem uma melhor resolução. Para a simulação de fontes afastadas, a fonte de interesse foi localizada em 10° e as fontes interferentes em –20° 40° e 60°. A Figura 3.13 mostra o espectro espacial de potência em coordenadas cartesianas desta simulação.

No caso de fontes vizinhas, o sistema consegue apontar seu maior feixe na direção de interesse com maior resolução e reduzir o feixe nas direções das fontes interferentes, mas não consegue impor um nulo na fonte interferente vizinha, conforme mostra a Figura 3.14. Para esta simulação como nas simulações anteriores, a fonte de interesse está localizada em 10° e as fontes interferentes em –20° 15° e 60° com SNR de 0 dB.

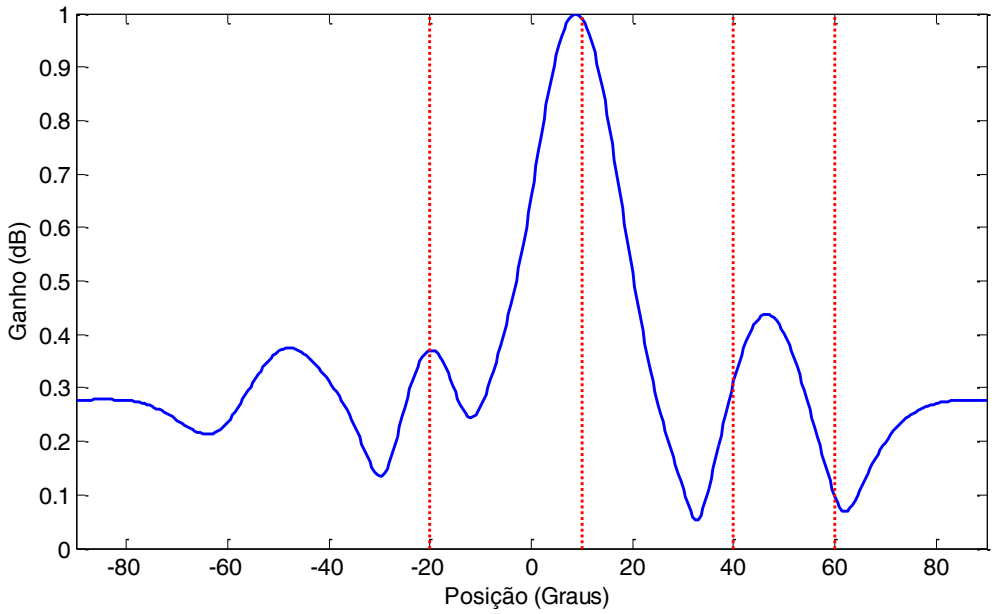

Figura 3.11: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de –20 dB.

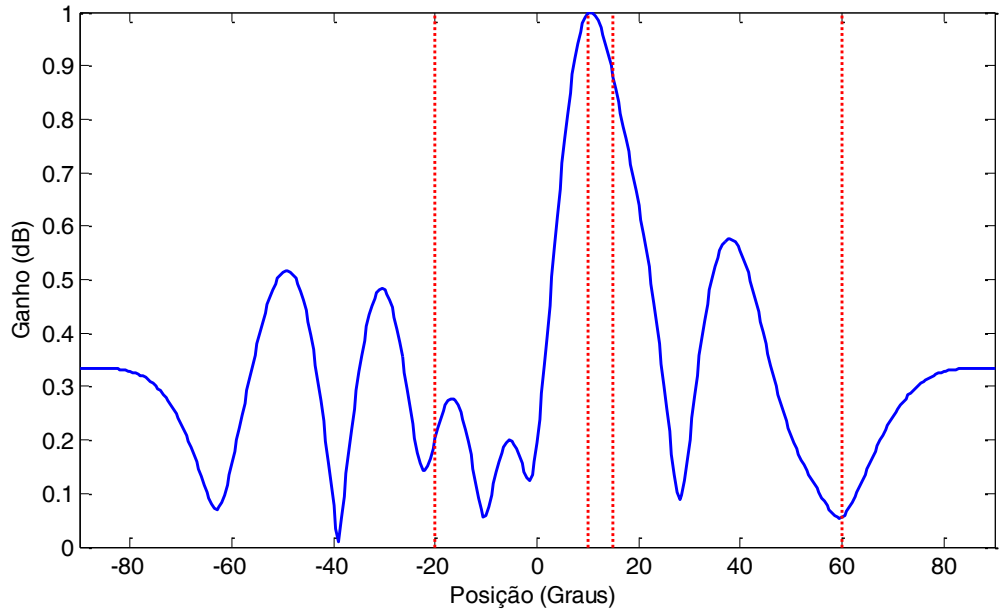

Figura 3.12: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR –20 dB.

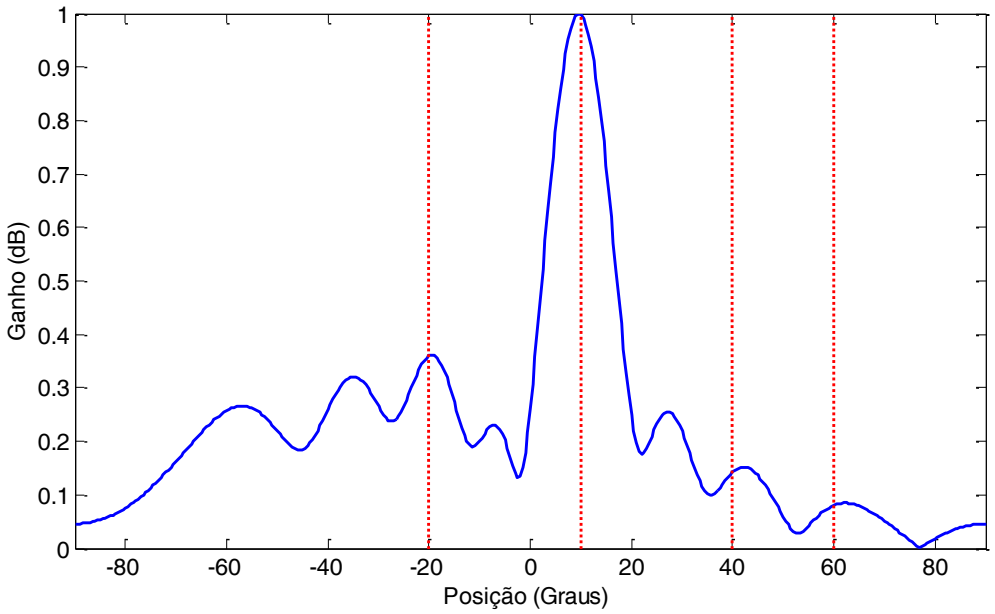

Figura 3.13: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 10°, 40° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB.

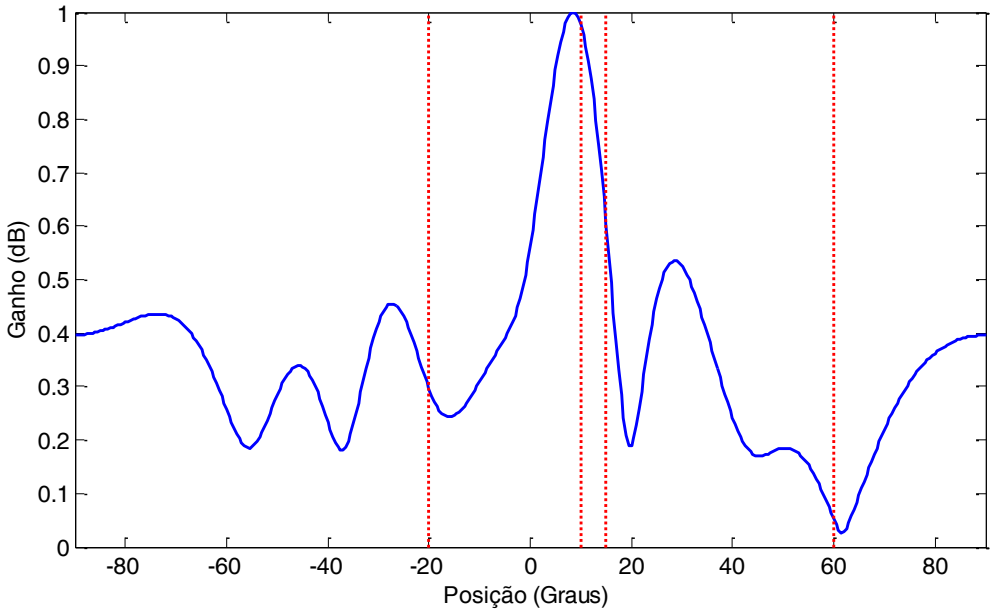

Figura 3.14: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 10°, 15° e 60°, usuário de interesse localizado em 10°, com SNR de 0 dB.

4 Algoritmo Genético Compacto

4.1 Introdução

Entre os algoritmos evolutivos destaca-se o Algoritmo Genético Compacto (AGC) que apresenta características que o tornam uma ferramenta de extrema aplicabilidade, já que as barreiras de implementação de *software* e *hardware* que limitam outros algoritmos genéticos desaparecem neste caso.

Considerando que até o momento não existem registros sobre a utilização do AGC no problema da estimação da DOA e *beamforming,* a proposta feita nesta dissertação apresenta uma novidade na área das antenas inteligentes.

No presente capítulo é descrita a família de Algoritmos Genéticos Compactos e a forma como foi adaptada ao problema da estimação da DOA e *beamforming*.

4.2 Algoritmo Genético Compacto

Segundo (SILVA, 2008 *apud* HE e YAO, 2002), o tamanho da população é de altíssima relevância para o desempenho dos algoritmos baseados em estratégias evolutivas, já que o incremento da população aumenta a probabilidade de primeiro acerto no algoritmo. Entretanto isso pode causar um grande impacto no tempo de execução, ou seja, um tamanho maior da população melhora a exatidão da resposta do algoritmo mas aumenta o consumo dos recursos de *hardware* disponíveis, tornando o processo lento. Esta limitação motivou a pesquisa de novas técnicas que compartilham as mesmas vantagens dos AG mas com custos computacionais mais baixos, conhecidos como Algoritmos de Estimação de Distribuição (EDA, do inglês *Estimation of Distribution Algorithm*).

Os EDA's são um novo ramo da computação evolucionária que baseia seu funcionamento em distribuições probabilísticas no lugar dos operadores genéticos, como cruzamento e mutação. Segundo (LARRAÑAGA e LOZANO, 2002) em vez de usar operadores genéticos, são feitas amostragens de acordo com uma distribuição probabilística,

obtida a partir de um grupo de indivíduos selecionados na população. Os EDA's geram uma população inicial aleatoriamente e, em seguida, selecionam um grupo de indivíduos para estimar uma distribuição probabilística. A distribuição probabilística é usada para gerar os indivíduos que substituirão parte ou toda a população atual. Este processo iterativo se repete até que o critério de parada seja cumprido. Os EDA's utilizam diferentes tipos de distribuições probabilísticas, em (LOZANO, LARRAÑAGA, INZA e BENGOETXEA, 2006) são expostos alguns tipos de EDA's.

O algoritmo genético compacto é um EDA que, segundo (HARIK, 1999), foi inspirado no modelo de passo aleatório (*random walk*), introduzido por ele mesmo em (HARIK, 1997). O AGC representa a população como uma distribuição de probabilidade discretizada sobre um conjunto de soluções (HARIK, 1999), com o objetivo de simular o comportamento de um AG com menos espaço de memória. A forma como o AGC manipula esta distribuição imita o comportamento de um Algoritmo Genético simples (AGS) de ordem um (HARIK, 1999).

O AGC não armazena a população para gerar descendência, em vez disso usa um vetor de probabilidade sobre todo o cromossomo para representar a população, como faz o algoritmo de aprendizagem incremental baseado na população (PBIL, do inglês *Population-based Incremental Learning*) desenvolvido por (BALUJA, 1994). O vetor de probabilidade é inicializado com valores iguais a 0.5 e cada valor é deslocado até 0.0 ou 1.0, conforme avança o processo.

Na Figura 4.1 é mostrado o pseudocódigo do AGC, onde seus parâmetros são o tamanho da população n e o comprimento do cromossomo l. A população é representada pelo vetor l-dimensional p, onde cada posição do vetor $p[i]$ representa a probabilidade do *bit* da i-ésima posição. Os dois candidatos são gerados à partir do vetor de probabilidade p, e o valor de *fitness* de cada candidato é calculado com a função de avaliação que se deseja otimizar. Após selecionar os candidatos ganhador e perdedor conforme seus valores de *fitness,* o vetor de probabilidade é atualizado de acordo com o indivíduo ganhador para produzir candidatos com maior aptidão na geração seguinte. Este processo é repetido até cumprir o critério de parada que pode ser alguma taxa de erro do problema considerado ou quando cada componente do vetor $p[i]$ seja igual a um (1) ou zero (0).

4.2.1.1 A família de Algoritmos Genéticos Compactos

A família de AGC foi introduzida em (GALLAGHER, VIGRAHAM e KRAMER, 2004) com o alvo de aumentar o poder de busca do AGC desenvolvido em (HARIK, 1999). As modificações introduzidas melhoram a capacidade de busca do AGC para problemas de otimização de grande ordem, sem aumentar significativamente a complexidade computacional. Segundo (GALLAGHER, VIGRAHAM e KRAMER, 2004), mesmo com as modificações propostas, a família de AGC pode ser implementada em *hardware* sem dificuldade, o que aumenta a aplicabilidade em diferentes áreas da engenharia.

A primeira das modificações propostas em (GALLAGHER, VIGRAHAM e KRAMER, 2004) é a introdução de elitismo para garantir a permanência do melhor indivíduo ao longo das gerações. Para isso, foi introduzido um novo passo no AGC, onde só o indivíduo perdedor é substituído para o torneio seguinte. O passo (4.26) representa a implementação do elitismo no AGC conforme é mostrado na Figura 4.2. A segunda modificação é a inserção de mutação, agregando um torneio secundário no AGC, onde são comparados os *fitness* do indivíduo ganhador e uma versão mutante dele mesmo. O ganhador deste torneio substitui o ganhador anterior por um novo candidato para o próximo torneio. A mutação é implementada considerando que ela implica o uso de elitismo, por tanto, a mutação deve ocorrer antes do elitismo, conforme é exposto no passo (4.25) da Figura 4.2. Finalmente, é introduzida a reamostragem do campeão, para evitar que uma função objetivo ruidosa mantenha um indivíduo ruim erroneamente. Isso permite também localizar os ótimos globais de funções objetivo dinâmicas, que trocam a posição do seu ótimo global ao longo do tempo. Portanto reduz-se a taxa de erro, já que sem reamostragem é possível que o algoritmo não detecte as mudanças da função objetivo, mantendo resultados ou indivíduos sub-óptimos. Para a reamostragem, é necessário modificar o torneio de tal forma a reavaliar o campeão a cada determinada quantidade de iterações. O algoritmo completo é mostrado na Figura 4.2.

Nesta dissertação, todos os membros da família de AGC são usados na estimação DOA e conformação de feixe, mas para isso é preciso definir a função objetivo, o critério de parada e adaptar o algoritmo para este problema conforme será mostrado mais na frente.

Com as modificações expostas, podem ser feitas diferentes combinações para formar AGC com distintos tipos de operadores biológicos. As combinações que podem ser feitas são expostas na tabela 4.1.

1. Inicializar vetor de probabilidade p.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ l\ \boldsymbol{do}\ p[i] \coloneqq 0.5;$
2. Gerar dois candidatos com base em p.
   $a \coloneqq gerar(p);$
   $b \coloneqq gerar(p);$
3. Determinar o ganhador e o perdedor.
   $ganhador, perdedor \coloneqq torneio(a, b);$
4. Atualizar p com respeito ao candidato ganhador.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ l\ \boldsymbol{do}$
   $\quad \boldsymbol{if}\ ganhador[i] \neq perdedor[i]\ \boldsymbol{then}$
   $\quad\quad \boldsymbol{if}\ ganhador[i] = 1\ \boldsymbol{then}$
   $\quad\quad\quad p[i] \coloneqq p[i] + \frac{1}{n};$
   $\quad\quad \boldsymbol{else}$
   $\quad\quad\quad p[i] \coloneqq p[i] - \frac{1}{n};$
5. Verificar se o vetor tem convergido.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ l\ \boldsymbol{do}$
   $\quad \boldsymbol{if}\ p[i] > 0\ and\ p[i] < 1\ \boldsymbol{then}$
   $\quad\quad \boldsymbol{return\ to}\ step\ 2;$
   Parâmetros do AGC.
   n: $tamanho\ da\ população.$
   l: $longitude\ do\ cromossomo.$

Figura 4.1: Pseudocódigo do AGC.

	Elitismo	Mutação	Reamostragem	Comentários
AGC	Não	Não	Não	É o AGC simples mostrado na Figura 4.1.
AGCe	Sim	Não	Não	AGC com elitismo
AGCm	Sim	Sim	Não	AGC com elitismo e mutação
AGCemr	Sim	Sim	Sim	AGC com elitismo, mutação e Reamostragem.

Tabela 4.1: Membros da família de AGC's.

1. Inicializar o vetor de probabilidade p.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ l\ \boldsymbol{do}\ p[i] \coloneqq 0.5;$
2. Gera dois candidatos com base em p.
   $a \coloneqq gerar(p);$
   $b \coloneqq gerar(p);$
3. Determinar o ganhador e o perdedor.
   $ganhador, perdedor \coloneqq torneio(a, b);$
4. Atualizar p com respeito ao candidato ganhador.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ l\ \boldsymbol{do}$
   $\boldsymbol{if}\ ganhador[i] \neq perdedor[i]\ \boldsymbol{then}$
   $\boldsymbol{if}\ ganhador[i] = 1\ \boldsymbol{then}$
   $p[i] \coloneqq p[i] + \frac{1}{n};$
   $\boldsymbol{else}$
   $p[i] \coloneqq p[i] - \frac{1}{n};$

4.25. Mutar e avaliar o ganhador.
   $\boldsymbol{if}\ \ fitness(a) > fitness(b)\ \boldsymbol{then}$
   $\{c \coloneqq mutar(a); avaliar(c);$
   $\boldsymbol{if}\ fitness(c) > fitness(a)\ \boldsymbol{then}$
   $a \coloneqq c;\}$
   $\boldsymbol{else}$
   $\{c \coloneqq mutar(b); avaliar(c);$
   $\boldsymbol{if}\ fitness(c) > fitness(b)\ \boldsymbol{then}$
   $b \coloneqq c;\}$

4.26. Trocar um indivíduo com base em p.
   $\boldsymbol{if}\ ganhador == a\ \boldsymbol{then}\ b \coloneqq genar(p);$
   $\boldsymbol{else}\ a \coloneqq gerar(p);$

5. Verificar se o vetor tem convergido.
   $\boldsymbol{for}\ i \coloneqq 1\ \boldsymbol{to}\ l\ \boldsymbol{do}$
   $\boldsymbol{if}\ p[i] > 0\ and\ p[i] < 1\ \boldsymbol{then}$
   $\boldsymbol{return\ to}\ step\ 3;$

Parâmetros da família dos AGCs

n: $tamanho\ da\ popula$ção.
l: $longitude\ do\ cromossomo.$

Figura 4.2: Pseudocódigo da família de AGC's.

4.2.2 A Família de AGC na Estimação de DOA e Beamforming

Para a implementação da família de AGC, primeiro se define a função objetivo que se deseja otimizar. Em nosso caso, ela é a mesma usada pelo algoritmo LMS, ou seja, é o erro entre o sinal de referência gerado pelo sistema e o sinal de dados estimado a partir dos *snapshots* e pesos do arranjo, expressa em (3.13), que é reescrita como:

$$\boldsymbol{\varepsilon_i} = \boldsymbol{y}^i - \boldsymbol{r}_i \tag{4.1}$$

O segundo parâmetro modificado para garantir boas estimativas, com baixa complexidade computacional para implementação em tempo real, é o critério de parada do AGC. O critério de parada do AGC foi substituído por um limiar que tem que ser superado pelo valor de *fitness* do indivíduo ganhador. Nesta dissertação, o valor de *fitness* dos candidatos é estimado pelo erro quadrático médio (MSE, do inglês *Mean Square Error*) da função objetivo, a qual serve para determinar a aptidão de cada indivíduo e, assim, decidir qual deles é melhor. Considerou-se que o AGC usa 640 *snapshots* para otimizar a função objetivo. O passo 5 da Figura 4.2 é mudado pelo critério de parada e exposto na Figura 4.3.

5. Verificar se o vetor tem convergido.

$$\boldsymbol{if}\ \frac{\sum_{N=1}^{640}\left[\left(\boldsymbol{y}^i - \boldsymbol{r}_i\right)^2\right]}{640} > valor_desejado\ \boldsymbol{then}$$

$$\boldsymbol{return\ to}\ step\ 3;$$

Figura 4.3: Modificação do critério de parada.

Após a modificação destes parâmetros, adapta-se o AGC para o problema de estimação da DOA e *beamforming*, definindo o tamanho do vetor de probabilidade, a função para gerar indivíduos ou cromossomos e o tipo de mutação que se vai usar. O tamanho do vetor de probabilidade é definido mais na frente na seção 4.2.2.1 Codificação do Espaço de Busca.

Para gerar os cromossomos se implementou uma função que faz um *sorteio* com cada elemento do vetor de probabilidade para originar cada alelo do cromossomo, ou seja, se faz

uma estimativa pseudoaleatória de cada bit do vetor que representa o cromossomo com base no valor de probabilidade de cada membro do vetor de probabilidade. Este *sorteio* inicia quando o vetor de probabilidade está inicializado em 0.5, entregando um vetor (cromossomo) de zeros e uns. Este processo se realiza à medida que o vetor de probabilidade é atualizado e vai aos poucos convergindo para a resposta certa, aumentando a probabilidade de obter cromossomos que otimizem a função objetivo, conforme é mostrado nos pseudocódigos das Figuras 4.1 e 4.2.

No projeto foi implementada mutação binária, que usa uma máscara binária gerada aleatoriamente com comprimento igual ao tamanho do cromossomo. Com esta máscara é feita uma operação logica XOR ou OR exclusivo com o cromossomo ganhador, para estimar a versão mutante dele mesmo e determinar qual das versões do cromossomo têm melhor aptidão. Na Figura 4.4 é exposto o processo de mutação.

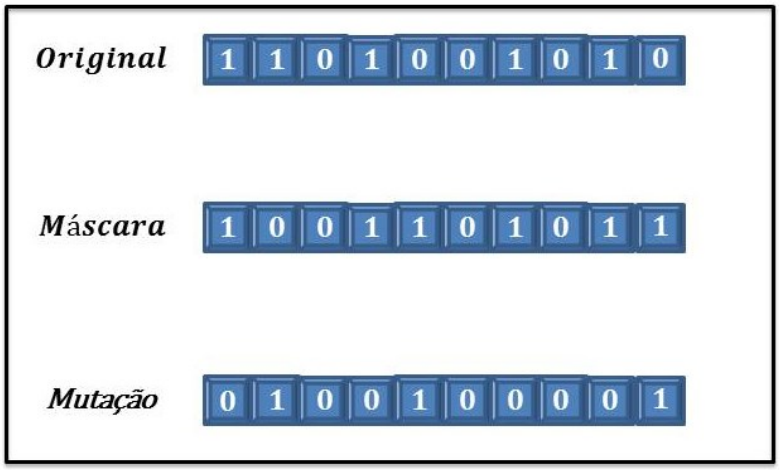

Figura 4.4: Mutação Binaria.

Os resultados das simulações da família de AGC para diferentes valores de parâmetros e para diferentes ambientes de simulação são expostos no capítulo 5, onde se testa o desempenho da família de AGC quanto a exatidão e complexidade computacional.

O espectro de um arranjo de 10 sensores isotrópicos controlado pelo Algoritmo Genético Compacto, para a estimação da DOA e *beamforming,* para quatro fontes descorrelacionadas afastadas, situadas em -20° 5° 40° e 60°, com relação Sinal-Ruído de –10 dB é mostrado na Figura 4.5. Como os valores do espectro foram normalizados, o eixo das

ordenadas representa o ganho imposto à direção em questão. Como pode ser percebido na Figura 4.5, o AGC posicionou um ponto de ganho máximo na posição de interesse, e atenuou o ganho dos lóbulos que apontam nas posições das fontes interferentes. Já no caso de fontes vizinhas, o AGC não conseguiu separar a fontes de interesse e a fonte interferente vizinha, ele aponta o ganho máximo do arranjo na posição de interesse mas beneficia também o sinal interferente vizinho. Por outro lado, o ganho nas direções das outras fontes interferentes não é anulado. A Figura 4.6 apresenta a representação gráfica do espectro espacial de potência do mesmo arranjo linear uniforme de 10 sensores, com utilização do AGC para a estimação da DOA e *beamforming* para fontes vizinhas, localizadas em –20° 5° 10° e 60° com SNR de –10 dB.

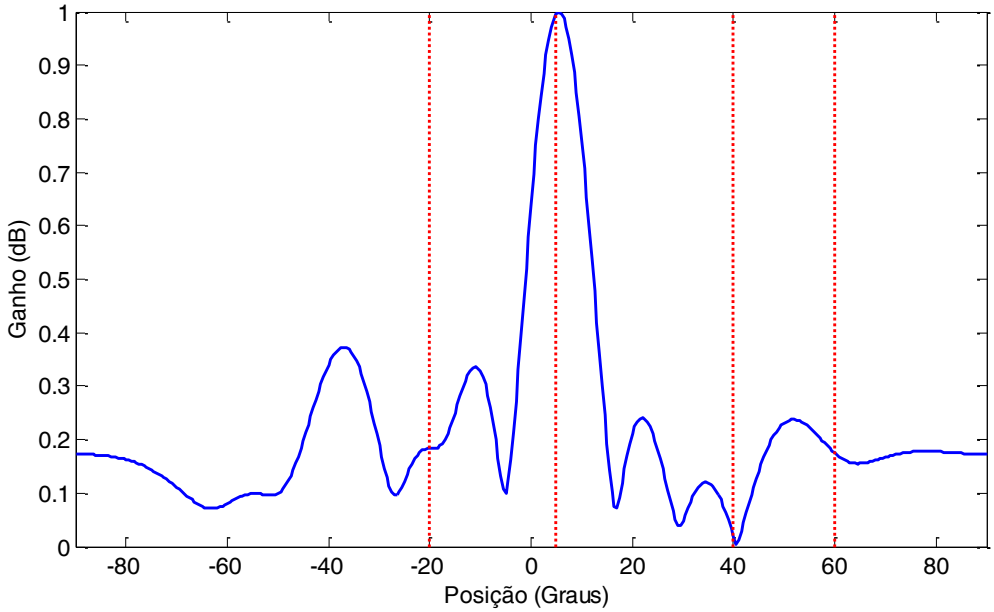

Figura 4.5: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 5°, 40° e 60°, usuário de interesse localizado em 5°, com SNR de –10 dB.

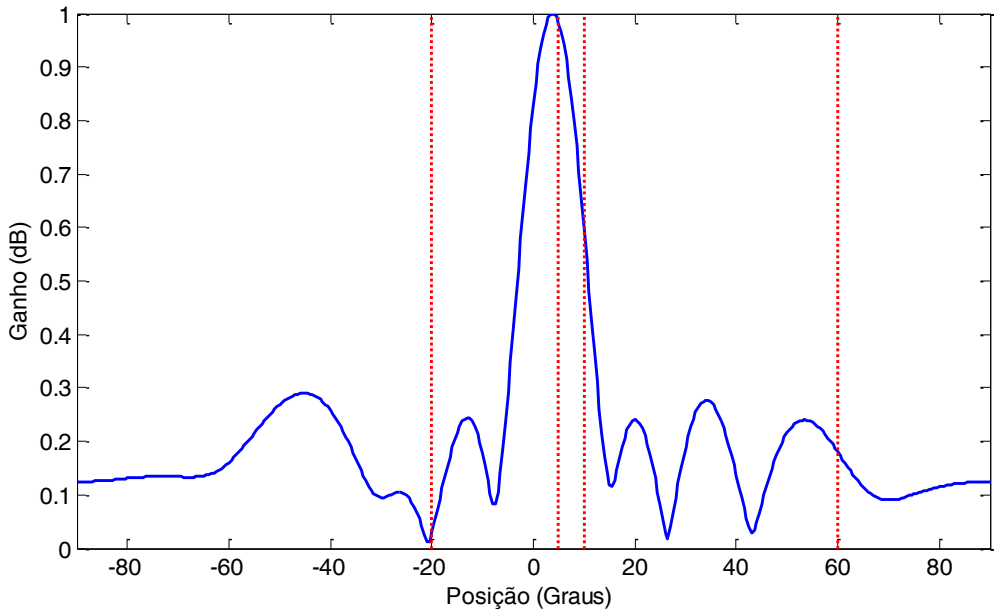

Figura 4.6: Representação em coordenadas cartesianas do espectro espacial de potência de um ULA controlado pelo algoritmo Genético, para quatro sinais descorrelacionadas de mesma potência localizadas em –20°, 5°, 10° e 60°, usuário de interesse localizado em 5°, com SNR de –10 dB.

4.2.2.1 Codificação do Espaço de Busca

Nesta dissertação foi definido um vetor de probabilidade de 420 posições, portanto considera-se que os cromossomos têm esse mesmo tamanho, onde cada bloco de 42 bits representa um peso complexo. Os 42 bits estão distribuídos da seguinte forma; um bit para representar o sinal da parte real do peso, vinte bits para representar a magnitude da parte real, um bit para representar o sinal da parte imaginária e os vinte bits restantes representam a magnitude da parte imaginária do peso. O objetivo de usar um vetor de probabilidade dessa dimensão é garantir alta resolução nas estimativas do algoritmo.

5 Análise e Resultados de Desempenho de Estimação e Complexidade Computacional

5.1 Introdução

Nas áreas tanto das telecomunicações como da computação, a complexidade computacional de um algoritmo é um parâmetro que determina a sua aplicabilidade e a sua viabilidade na otimização de diferentes tipos de problemas. Os testes propostos para o algoritmo LMS e a família de AGC são feitos para diferentes situações de SNR, quantidade de fontes interferentes, fontes estáticas e em movimento, e correlação entre as fontes. Além disso, para a família de AGC são testadas estas situações modificando seus parâmetros e os operadores genéticos.

5.2 Considerações do Ambiente de Simulação

No ambiente de simulação, são alocadas diferentes quantidades de usuários ou assinantes que enviam sinais binários de 10 bits de informação, amostrados à taxa de 10 amostras/bit e codificados com sequências de Walsh de 64 *chips*. Portanto, o comprimento total dos sinais transmitidos é de 6400 *chips*. Nos sinais são inseridos atrasos aleatórios com valores entre $1xT_c$ e $64xT_c$, ou seja, o atraso mínimo é de um período de *chip* e o atraso máximo é de 64 períodos de *chip*. Isso significa que os sinais sofrem um atraso máximo de um período de bit T_b. Se esclarece que o tipo de codificação utilizado nesta tese não restringe os resultados conseguidos.

O sinal $Z(t)$ que incide no arranjo corresponde à superposição de todos os sinais que são enviados pelo canal de comunicação mais a influência de ruído branco Gaussiano aditivo complexo de variância igual a um. Considerou-se que todos os sinais têm a mesma potência. No receptor trabalhou-se com janelas de dados, que são conjuntos de amostras utilizadas nas iterações dos algoritmos para a convergência do receptor adaptativo. As janelas usadas foram blocos de 100 *snapshots* para cada sensor do arranjo. Por exemplo, se a quantidade de

sensores do arranjo é 10, cada janela totaliza uma matriz de 1000 amostras (10x100) para fazer a estimação, ou seja, que a quantidade de amostras por janela depende da quantidade de sensores do arranjo já que a quantidade de *snapshots* por janela é fixa. Todas as simulações foram feitas com as considerações aqui adotadas, exceto quando especificada outra configuração do ambiente.

O principal alvo desta dissertação é avaliar o desempenho do bloco do algoritmo adaptativo, mais concretamente o desempenho da família de AGC. Assim, nas simulações, primeiro cada bloco é testado individualmente para conferir seu desempenho. A ideia é que no momento dos testes nenhum bloco dependa de resultados de estimações de outros blocos, uma vez que uma estimativa errada em um bloco afetaria o processamento do bloco seguinte, refletindo no desempenho global. Posteriormente, o receptor é testado por completo, integrando cada um dos blocos, com a finalidade de simular um comportamento do receptor da maneira mais próxima da realidade.

Considerando que as simulações incluem testes sobre a velocidade de convergência dos algoritmos usados, é importante mencionar as características de *hardware* e *software* do equipamento que foi usado para fazer as simulações, para ter uma ideia clara da velocidade que alcançam os algoritmos neste equipamento e que podem alcançar em outros com melhores características. As simulações foram feitas usando MATLAB® 2013a instalado num computador Macbook Air com processador Intel® Core I5, 4 GB de memória RAM, 120 GB de disco rígido com tecnologia *flash*, usando sistema operacional Apple® Mavericks (OS X 10.9.4).

5.3 Bloco Sincronizador

Conforme foi exposto no capítulo 2, o bloco sincronizador é um correlator temporal que estima a correlação entre a média dos dados induzidos nos sensores e o código de Walsh de interesse. O sincronizador procura estimar o atraso da fonte de interesse e sincronizar os dados recebidos pelo arranjo considerando o atraso detectado. O desempenho do bloco sincronizador é testado em relação à quantidade de fontes alocadas no canal de comunicação e à SNR. A Figura 5.1 mostra os resultados das simulações para diferentes quantidades de fontes descorrelacionadas alocadas no canal de comunicação.

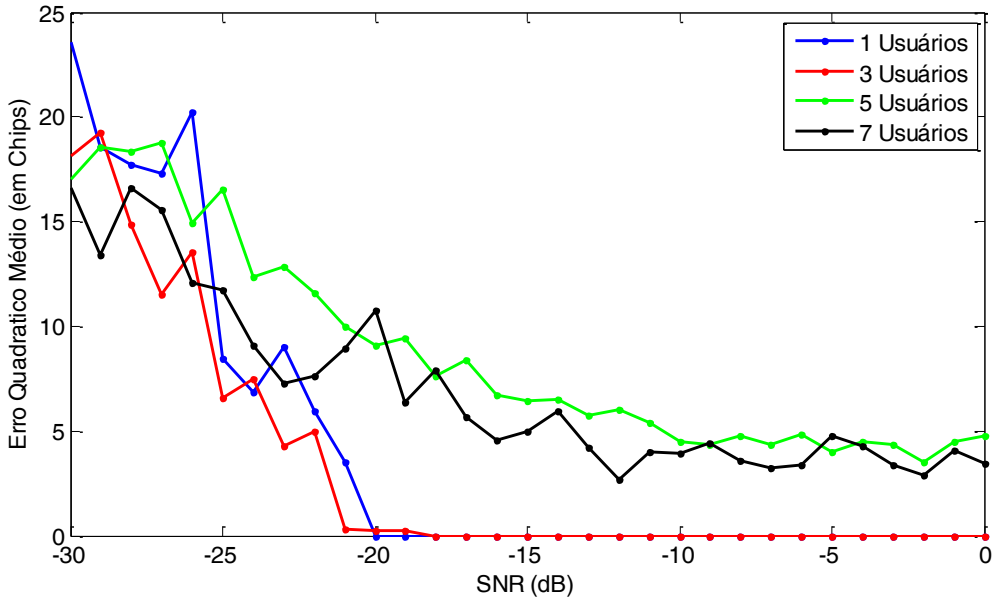

Figura 5.1: Erro quadrático Médio em função da relação Sinal-Ruído, do bloco sincronizador, para diferentes quantidades de usuários descorrelacionados.

Como pode ser percebido na figura acima, o sincronizador consegue estimar o atraso da fonte de interesse com exatidão a partir de −18 dB, quando existem até três usuários alocados no canal. Por outro lado, o sincronizador apresenta resultados discutíveis quando há cinco ou mais assinantes no canal de comunicação. Isto se deve à simplicidade da implementação do sincronizador, mas, deve-se esclarecer que o sincronizador não é o objetivo principal desta dissertação. Ele foi implementado só para simular um sistema receptor completo semelhante aos receptores usados por sistemas reais. Contudo, esse fato não discrimina os resultados obtidos pelos blocos subsequentes.

No caso onde todos os sinais alocados no canal são totalmente correlacionados o sincronizador não conseguiu detectar o atraso do sinal de interesse em nenhum dos casos, apresentando estimativas com taxas de RMSE (do inglês, *Root Mean Square Error*) de até 80 *chips*. Mesmo com poucos assinantes no canal, foram observados comportamentos muito instáveis, conforme mostra a Figura 5.2. Recorda-se que nos sistemas reais de comunicação são usados sincronizadores muito mais robustos e nesta dissertação o sincronizador não é o principal alvo, daí a sua simplicidade de implementação.

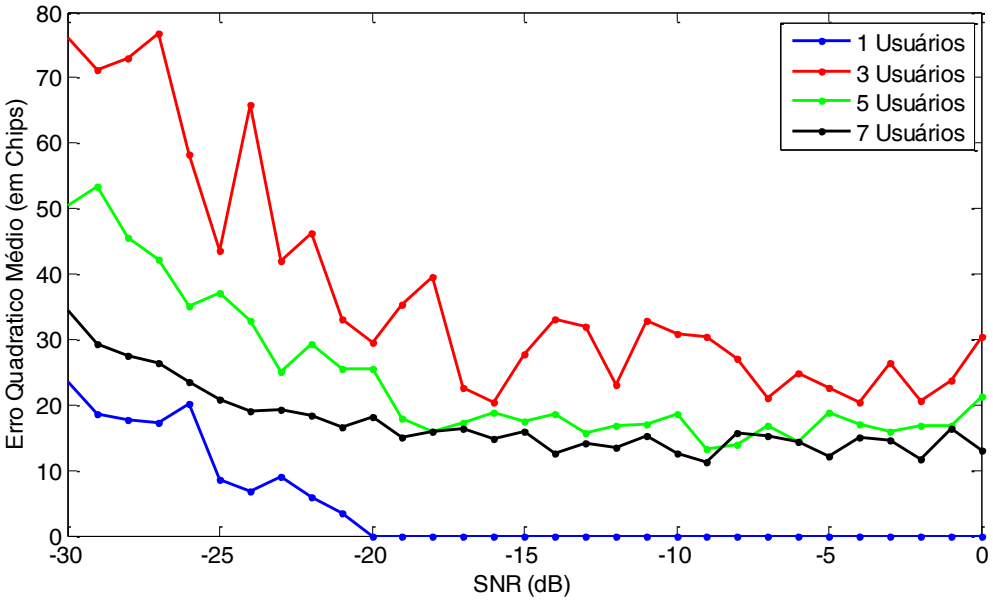

Figura 5.2: Erro quadrático Médio em função da relação Sinal-Ruído, do bloco sincronizador, para diferentes quantidades de usuários totalmente correlacionados.

5.4 Bloco Estimador de Bit

O bloco estimador de bit aproveita as características do espalhamento espectral para estimar os bits e gerar o sinal de referência. Utiliza-se a ortogonalidade entre as sequências de espalhamento para rejeitar a influência dos sinais interferentes sobre o sinal de interesse. Assim, o bloco estimador de bit limpa os dados recebidos para, em seguida, estimar os bits de acordo com as expressões (2.24) e (2.25). Então, os bits são espalhados pela sequência de Walsh de interesse para gerar o sinal de referência usado pelo algoritmo adaptativo. O desempenho do bloco estimador de bit é mostrado na Figura 5.3.

Nos testes realizados para o bloco estimador de bit, foi considerado que todos os sinais alocados no canal são descorrelacionados e que só o sinal de interesse está sincronizado, enquanto os outros sinais alocados no canal apresentam atrasos. Assim, eles podem apresentar taxas maiores de correlação que afetam o desempenho do estimador de bit, já que na ausência de sincronismo entre todos os sinais, perde-se a ortogonalidade nas sequências de Walsh. Ainda assim, o estimador de bit demonstrou desempenho satisfatório, já que conseguiu baixas taxas de erro de bit (BER) (do inglês, *Bit Error Rate*) para SNR muito pequenas. O estimador consegue inclusive taxas BER nulas para SNR à partir de –26 dB, embora esse valor aumente com a quantidade de assinantes no canal.

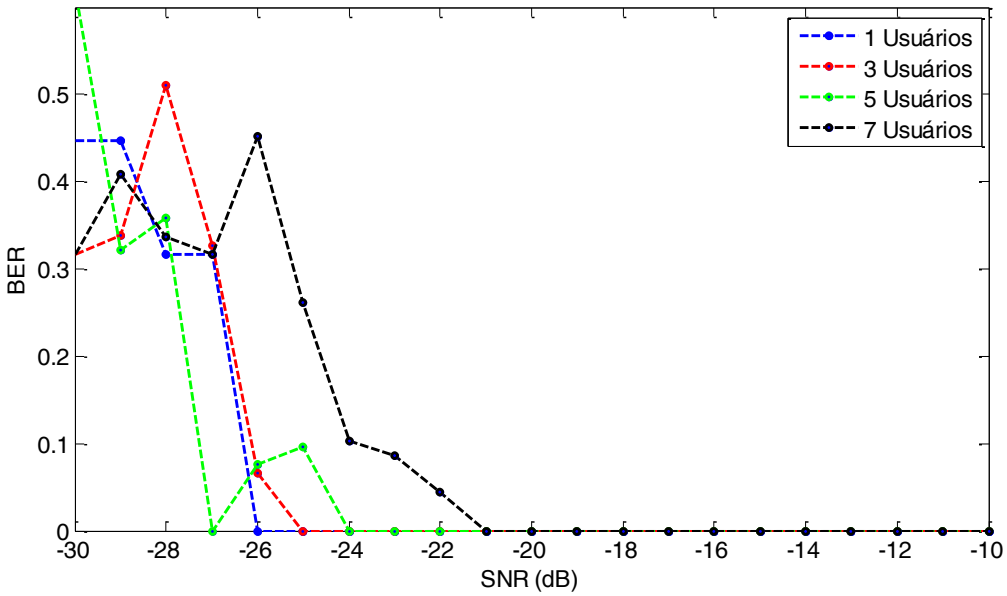

Figura 5.3: Taxas BER do bloco Estimador de Bit para diferentes quantidades de assinantes descorrelacionados.

Quando os sinais alocados no canal são totalmente correlacionados, as taxas BER são maiores devido à alta correlação que prejudica a ortogonalidade entre os sinais. Os resultados mostrados na Figura 5.4 são obtidos para fontes totalmente correlacionadas, onde só uma delas está sincronizada. Estes resultados expõem taxas BER muito elevadas que chegam até valores de 0.5, o que significa que a metade dos bits estimados pelo bloco estimador de bit estão errados. Estas taxas BER não diminuem conforme aumenta o SNR, em vez disso apresentam comportamentos muito instáveis e nunca conseguem valores nulos.

Conforme mostra a Figura 5.4, o aumento na quantidade de assinantes alocados no canal afeta diretamente o desempenho do estimador de bit, à medida que mais usuários totalmente correlacionados levam a maiores taxas de correlação entre os sinais induzidos nos sensores e a sequência de espalhamento de interesse. Isso afeta a principal vantagem dos sistemas CDMA que é a redução de interferência entre símbolos, daí fica mais difícil “limpar” o sinal para estimar o sinal de referência.

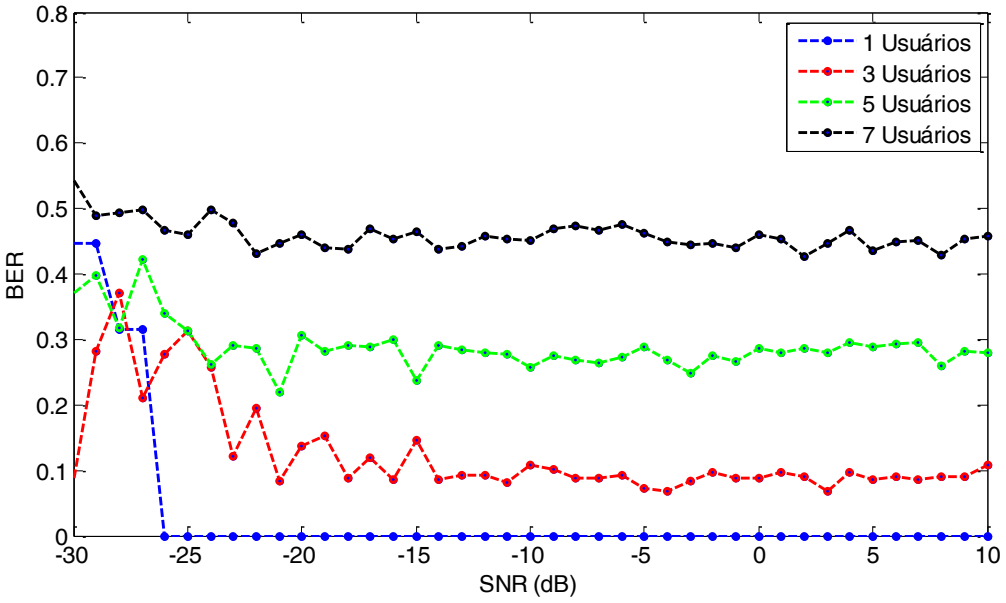

Figura 5.4: Taxas BER do bloco Estimador de Bit para diferentes quantidades de assinantes totalmente correlacionados.

5.5 Bloco Algoritmo Adaptativo

O bloco do algoritmo adaptativo é o bloco mais importante do receptor e é o objetivo principal desta dissertação. Conforme foi descrito nos capítulos anteriores, o algoritmo adaptativo é quem realiza a estimação dos pesos complexos que controlam o arranjo de antenas para poder apontar o feixe na direção de interesse. Foram realizados testes para determinar a precisão no direcionamento do feixe e a velocidade de convergência da família de AGC. Para conferir a viabilidade desta proposta, os resultados foram comparados com os aqueles obtidos pelo algoritmo LMS. O LMS é uma das técnicas mais usadas para o controle de arranjos de antenas, dada sua facilidade de implementação e, embora seja considerado de convergência lenta, é um algoritmo muito rápido. Assim, o algoritmo LMS foi usado justamente como método de referência porque ele não depende de uma população para convergir, bastando um valor inicial para fazer as estimativas e além é um dos métodos mais usados na literatura. Em princípio, devido a sua simplicidade, espera-se que ele seja mais rápido que um AG convencional, uma vez que à medida que aumenta a população, a velocidade de convergência do AG diminui devido ao aumento do esforço computacional para processar todos os indivíduos.

De acordo com o exposto anteriormente, a população determina a exatidão e a velocidade de convergência dos AG. Assim, convém determinar o tamanho da população que fornece as melhores estimativas com a maior velocidade de convergência possível para o AGC. A Figura 5.5 mostra o RMSE de mil estimativas de DOA e *beamforming* do arranjo controlado pelo AGC para diferentes tamanhos de populações e sob quatro SNR diferentes. O AGC foi programado segundo nosso critério para gerar no máximo 300 gerações, já que uma quantidade maior de gerações pode desacelerar o algoritmo. As estimativas são feitas para três fontes estáticas descorrelacionadas localizadas em 10°, 40° e 60°.

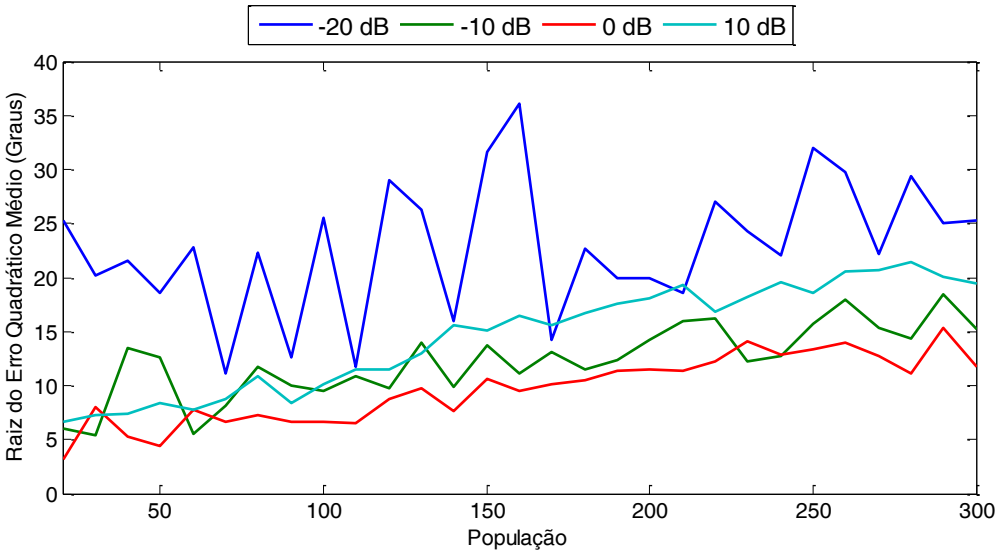

Figura 5.5: Testes de desempenho de direcionamento do Algoritmo Genético Compacto segundo o tamanho da população e a Relação Sinal-Rugido para três fontes estáticas descorrelacionadas localizadas em 10°, 40° e 60°.

A Figura acima mostra que as melhores estimativas são conseguidas com populações pequenas. Por exemplo, para uma SNR de –20 dB, o menor erro é conseguido quando o AGC tem uma população de 70 indivíduos. Para uma SNR de 10 dB, o melhor direcionamento é conseguido quando o AGC tem população de 20 indivíduos. Porém, o RMSE não é o único critério que se deve considerar para decidir qual tamanho de população é mais conveniente. Também se considera a velocidade de convergência. A Figura 5.6 expõe os tempos médios de execução das estimativas mostradas na Figura 5.5.

A Figura 5.6 mostra que, quando a SNR é baixa, o tamanho da população não apresenta maior perturbação nos tempos médios de execução do AGC, por outro lado, quando a SNR é

alta uma população pequena pode acelerar a convergência do AGC. Com uma população pequena o vetor de probabilidade tem mudanças maiores, o que permite gerar bons candidatos rapidamente. Junto com a boa qualidade dos dados recebidos, é mais simples determinar o sinal que minimiza o erro atendendo com maior velocidade o critério de parada, ou seja, a estimação do *beamforming* é mais rápida. Isso não ocorre em SNR baixas porque a má qualidade ou pouca resolução dos dados induzidos nos sensores dificulta a identificação de um sinal que minimize a função objetivo, apesar de o AGC ter gerado bons candidatos.

Contudo, os testes mostram que os melhores resultados em termos de RMSE foram obtidos para uma população de 20 indivíduos em todos os casos, exceto para a SNR de -20 dB que conseguiu o melhor resultado com uma população de 70 indivíduos. Conforme foi especificado anteriormente, os tempos de execução não foram muito afetados pelo tamanho da população em SNR baixas. Porém, para SNR de 10 dB, a população determina a velocidade de convergência do AGC. Assim, escolheu-se a população de 20 indivíduos para fazer os testes.

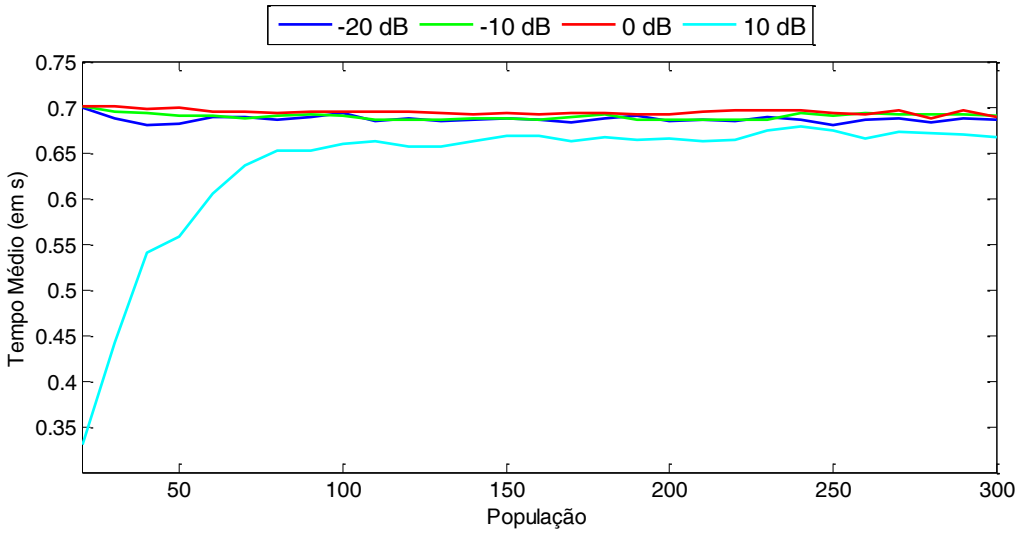

Figura 5.6: Tempos médios de execução das provas de desempenho de direcionamento do Algoritmo Genético Compacto segundo o tamanho da população e a Relação Sinal-Ruído para três fontes estáticas descorrelacionadas localizadas em 10°, 40° e 60°.

Um critério muito importante em todo tipo de algoritmo é o jeito como o AGC consegue a convergência, já que isto pode determinar a sua velocidade e a sua aplicabilidade em sistemas reis. Na Figura 5.7 são expostas as curvas de convergência do erro da família de AGC's e do método de referência.

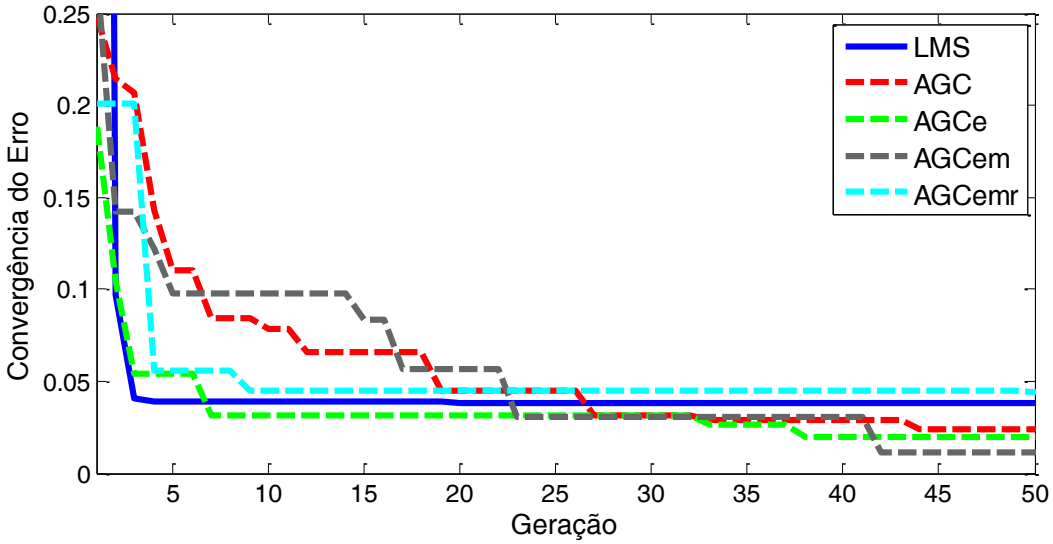

Figura 5.7. Convergência do Erro da família de AGC's e do algoritmo LMS para três fontes descorrelacionadas localizadas em 10º, 40º e 60º.

Na figura acima são mostrados apenas os resultados da convergência do erro para 50 gerações da família de AGC na estimação da DOA e *beamforming* de três fontes estáticas descorrelacionadas localizadas em 10º, 40º e 60º com uma SNR de 10 dB, isto com o objetivo de deixar as curvas dos erros mais visíveis para e poder comparar e avaliar mais comodamente a convergência dos algoritmos. Conforme mostra a Figura 5.7, o algoritmo LMS inicialmente faz uma mudança rápida, mas logo só consegui fazer alterações muito pequenas que torna-lo lento, por outro lado, a família de AGC faz uma série de mudanças escalonadas nas quais é possível evidenciar que o erro fica estático por várias gerações formando uns vales, mas apesar de isso a família de AGC consegue uma convergência mais rápida graças à sua diversidade genética. Se bem que os algoritmos conseguem uma convergência rápida, alguns dos membros da família mostram uns vales mais longos que impedem que algoritmo aumente a sua exatidão, como é o caso do AGCe e do AGCemr.

5.5.1 Desempenho de Estimação para Fontes Estáticas Descorrelacionadas

O desempenho do bloco Algoritmo Adaptativo foi medido em relação à sua capacidade de apontar corretamente o lóbulo principal do diagrama de radiação na direção de interesse, ou seja, mediu-se o RMSE dos direcionamentos obtidos nos testes. Também mediu-se a

capacidade de rejeitar as fontes interferentes, ou seja, a relação Sinal Interferência (SIR, do inglês *Signal-to-Interference Ratio*). Além disso, a velocidade de convergência é mais outro critério importante para determinar a viabilidade de um algoritmo, já que determina a sua aplicabilidade em sistemas reais, tendo sido escolhida como critério de desempenho.

Nos testes foram consideradas diferentes quantidades de fontes alocadas no canal, diferentes SNR e diferentes tipos de AGC. Os resultados obtidos foram comparados com aqueles do algoritmo LMS. Primeiramente, testou-se a estimação de DOA e *beamforming* obtido com os algoritmos para uma única fonte localizada em 10°, para uma faixa de SNR de –40 dB até 10 dB, fazendo-se mil experimentos para cada um deles. Tanto essa faixa de SNR quanto essa quantidade de experimentos foram mantidas para todos os testes. A Figura 5.8 exibe os RMSE das mil estimações feitas para cada SNR com cada um dos algoritmos implementados.

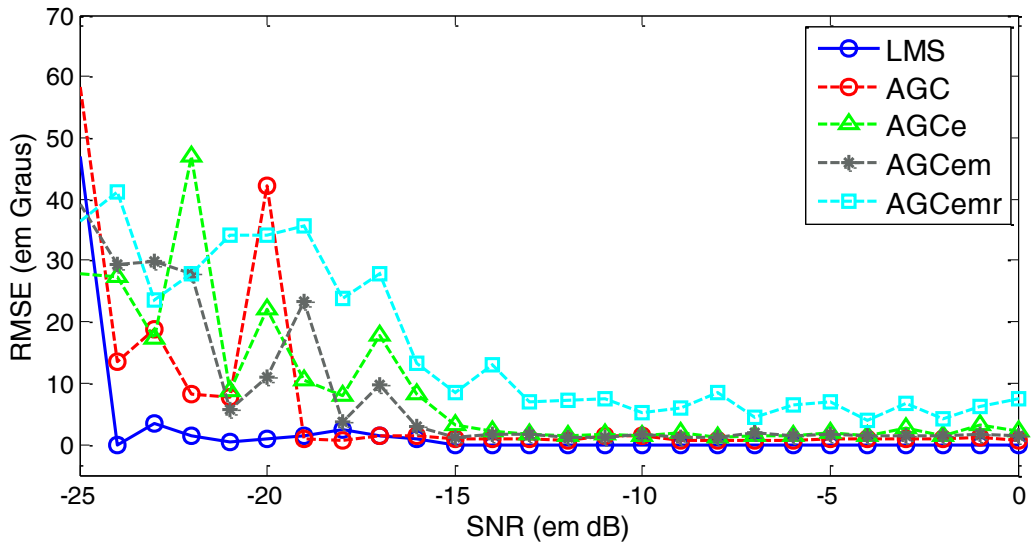

Figura 5.8: RMSE do bloco Algoritmo Adaptativo para uma fonte descorrelacionada localizada em 10°.

Na Figura 5.8 são mostrados só os resultados obtidos de –25 dB até 0 dB, uma vez que os comportamentos dos algoritmos foram instáveis e apresentaram altas taxas de RMSE para SNR menores que –25 dB. Por outro lado, para SNR maiores que 0 dB, os algoritmos apresentam comportamentos estáveis, facilitando a visualização dos resultados. A Figura 5.8 mostra que todos os algoritmos alcançaram taxas RMSE próximas de um grau, com exceção do AGCemr que não conseguiu se aproximar das taxas de RMSE conseguidas pelos outros

algoritmos. O algoritmo LMS alcançou erros baixos para SNR a partir de −24 dB, mas conseguiu erro nulo[1] a partir de −15 dB. Nenhum dos algoritmos da família de AGC obteve taxas nulas de RMSE, mas conseguiram taxas de erro próximas de um grau. O AGC foi que convergiu para valores mais baixos de SNR, alcançando erro unitário para −19 dB, enquanto que o AGCe e o AGCem conseguem taxas de erro próximas da unidade para −15 dB.

Dado que neste caso o canal só tem alocada uma única fonte, não é possível avaliar o desempenho dos algoritmos quanto à mitigação de interferências. Assim, foi usada a velocidade de convergência para determinar qual dos algoritmos tem o melhor desempenho sob aquelas condições de simulação. A Figura 5.9 expõe os tempos médios para cada SNR dos direcionamentos feitos pelos algoritmos na Figura 5.8. Na Figura 5.9 são mostrados os tempos médios dos mil experimentos feitos para cada SNR com o objetivo de mostrar como evoluíram os tempos médios de execução dos algoritmos. À partir desses tempos médios, foram feitos histogramas para determinar a moda[2] dos tempos de execução de cada algoritmo e determinar o algoritmo mais rápido. A Figura 5.10 mostra o histograma dos tempos médios de todos os algoritmos implementados.

As Figura 5.9 e 5.10 mostram que toda a família de AGC convergiu mais rapidamente que o algoritmo LMS, sendo o AGCe o mais rápido de toda a família. Os tempos médios de convergência do algoritmo LMS não diminuíram com o aumento da SNR, enquanto que os tempos médios das simulações da família de AGC diminuíram em SNR altas. Segundo os resultados da Figura 5.10, as modas dos tempos médios dos testes foram 0.86 segundos para o algoritmo LMS, para o AGC 0.70 segundos, para o AGCe 0.40 segundos, para o AGCem 0.71 segundos e para o AGCemr 0.70 segundos.

[1] Quando nesta tese se fala de erro nulo ou de erro zero, faz referência a zero (0) mesmo.
[2] Moda é um termo usado em estatística para referir ao valor que ocorre com maior frequência num conjunto de dados ou numa distribuição de dados.

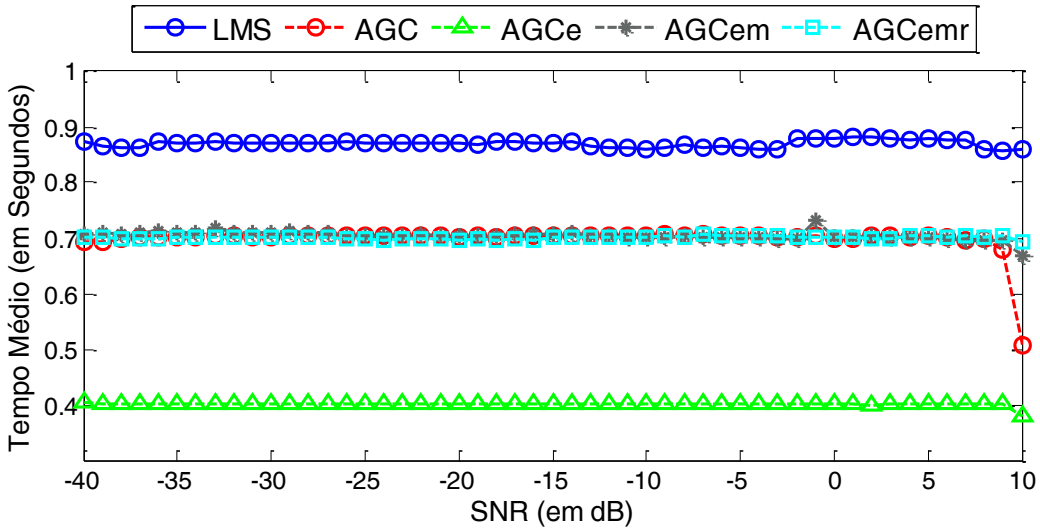

Figura 5.9: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para uma fonte descorrelacionada localizada em 10°.

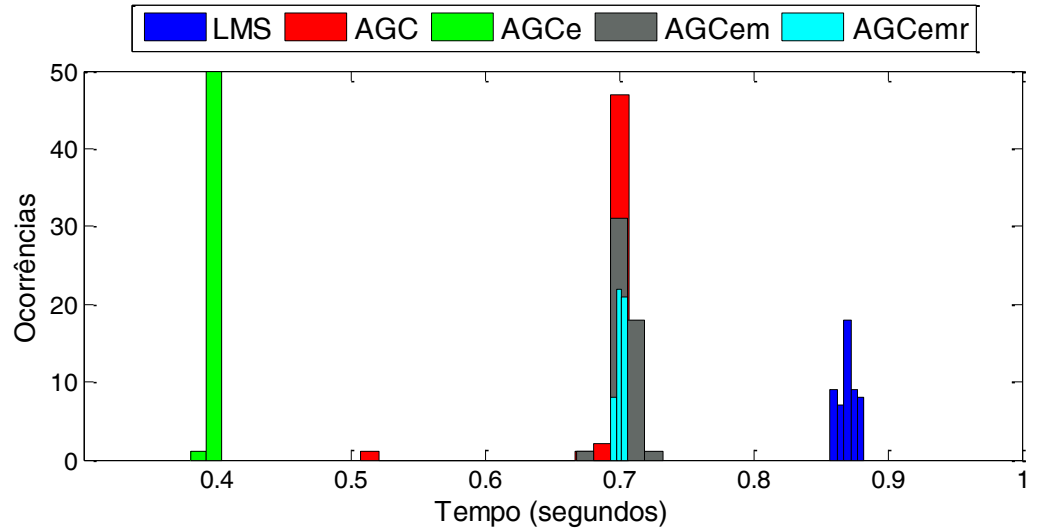

Figura 5.10: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para uma fonte descorrelacionada localizada em 10°.

Agora são apresentados os testes dos algoritmos conformando o feixe para três fontes localizadas em 10°, 40° e 60°, sendo a fonte de 10° a fonte de interesse. A Figura 5.11 mostra o RMSE das estimações feitas pelos algoritmos, na qual novamente o algoritmo LMS foi aquele que apresentou maior exatidão, conseguindo valores de RMS inferiores a um grau a partir de –22 dB e taxas nulas de RMSE a partir de –13 dB. O algoritmo que conseguiu acompanhar mais de perto o desempenho do LMS foi o AGC que alcançou taxas de RMSE próximas da unidade a partir de –18 dB. Por outro lado, os outros membros da família não

conseguiram se aproximar das taxas de erro conseguidas pelo AGC e pelo LMS, somente o AGCe conseguiu taxas de RMSE entre cinco e dois graus.

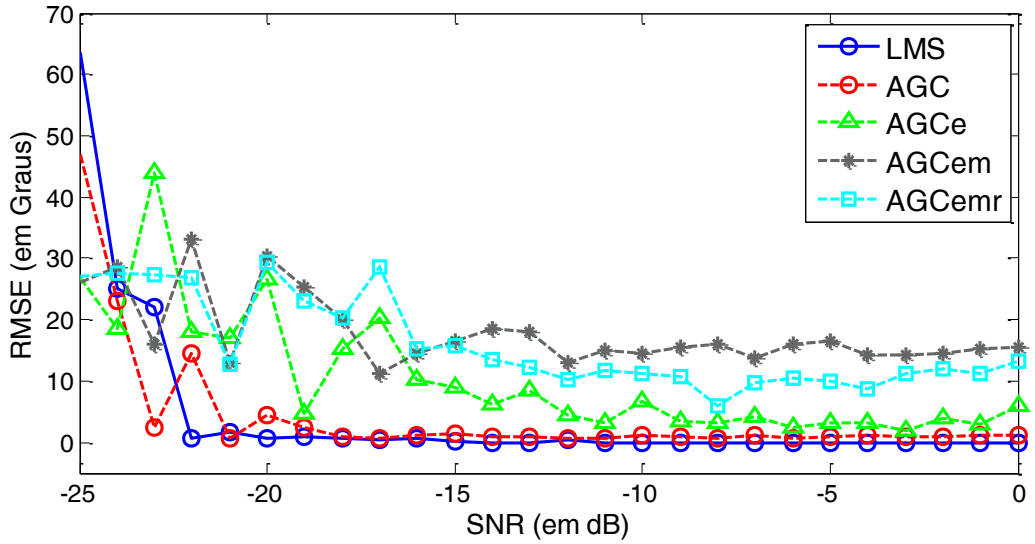

Figura 5.11: RMSE do bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°.

Para conseguir uma verdadeira filtragem espacial é preciso que o maior feixe seja apontado na direção da fonte de interesse e que os diagrama de radiação seja anulado nas direções das fontes interferentes. Assim, foi medida a capacidade de rejeição dos algoritmos para determinar seu desempenho. A Figura 5.12 mostra os níveis de relação Sinal-Interferência (SIR, do inglês *Signal-to-Interference Ratio*) conseguidos pelos algoritmos nos direcionamentos da figura anterior. Os resultados de SIR obtidos mostram que o algoritmo LMS consegue maiores níveis que os demais algoritmos a partir de –22 dB, enquanto que toda a família de AGC's tem maiores níveis de SIR para SNR menores que –23 dB. O AGC foi o único que conseguiu se aproximar da capacidade de rejeição do algoritmo LMS.

Além da capacidade de rejeição, o tempo de convergência tem um papel fundamental no desempenho de um algoritmo, já que este critério determina a sua aplicabilidade em sistemas em tempo real. A Figura 5.13 mostra como evoluíram os tempos de execução de todos os métodos sob diferentes SNR. A família de AGC conseguiu tempos de convergência menores que o método de referência, sendo o mais rápido o AGCe, enquanto que o resto de algoritmos da família apresentou tempos de convergência similares, que diminuíram quando a SNR era alta. Conforme mostra a Figura 5.14 a moda dos tempos para o algoritmo LMS foi de 0.86 segundos, para o AGC é de 0.69 segundos, para o AGCe é de 0.37 segundos, para o

AGCem é de 0.68 segundos e para o AGCemr é de 0.67 segundos, mostrando que o AGCe foi o membro mais rápido de toda a família e que os outros algoritmos apresentaram tempos de convergência similares, mas menores que os tempos do algoritmo LMS.

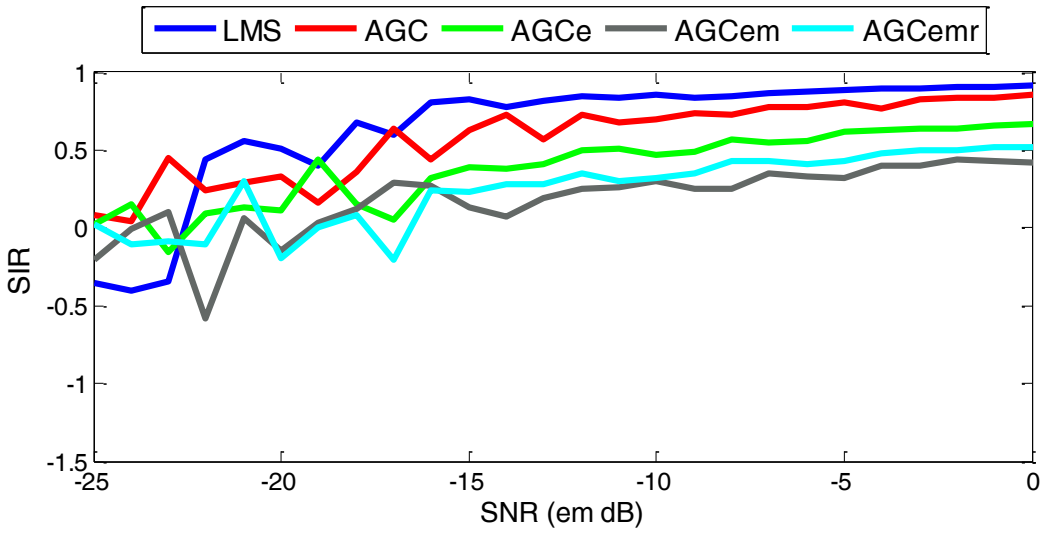

Figura 5.12: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°.

Considerando que um dos objetivos é testar a resposta dos algoritmos com o aumento das fontes alocadas no canal, alocamos cinco fontes estáticas descorrelacionadas localizadas em 10°, 40°, 60°, 20° e -20°, sendo a fonte de 10° a de interesse. Os resultados dos testes estão apresentados nas Figuras 5.15, 5.16, 5.17 e 5.18.

Na Figura 5.15 são mostrados os valores de RMSE conseguidos pelos métodos de estimação, revelando que outra vez o método de referência consegue a maior exatidão no direcionamento do lóbulo principal embora não alcance taxas nulas de RMSE, conseguindo valores próximos de 0.5 graus a partir de −16 dB. Este desempenho foi acompanhado pelo AGC que conseguiu erros próximos de 2 graus acima de −12 dB. Os outros métodos não conseguiram acompanhar estes desempenhos.

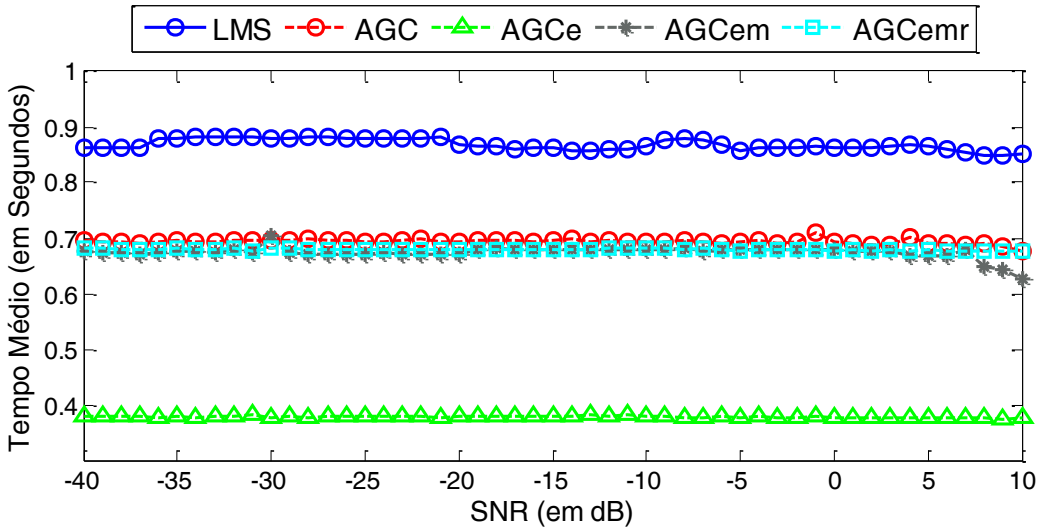

Figura 5.13: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°.

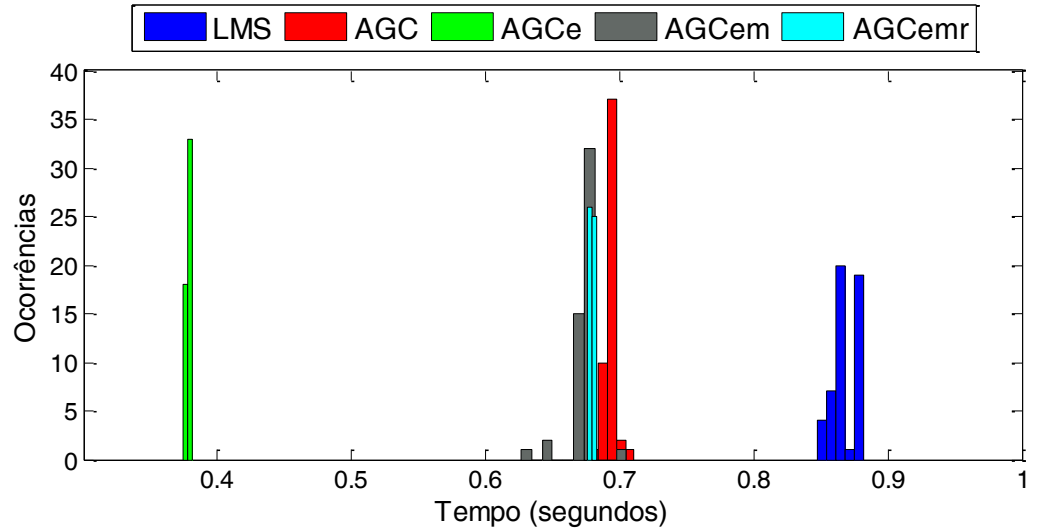

Figura 5.14: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para três fontes descorrelacionadas localizadas em 10°, 40° e 60°.

No caso da rejeição de interferências, nenhum dos algoritmos conseguiu se aproximar dos níveis de SIR conseguidos pelo método de referência (LMS). Novamente o AGC foi o método que conseguiu os melhores resultados de rejeição das fontes, mas ainda assim não conseguiu filtrar com a mesma exatidão que o método de referência conforme mostra a Figura 5.16.

Em relação aos os tempos de convergência, manteve-se a tendência dos tempos de toda a família de AGC serem menores que os tempos de convergência do método de referência conforme mostra a Figura 5.17. Além disso, as modas dos tempos dos algoritmos se mantiveram praticamente constantes, uma vez que as mudanças experimentadas com o aumento das fontes alocadas no canal foram muito pequenas. Neste caso as modas dos tempos para cada método, conforme o histograma da Figura 5.18, foram 0.87 segundos para o algoritmo LMS, para o AGC 0.70 segundos, para o AGCe 0.39 segundos, para AGCem 0.66 segundos e para o AGCemr 0.68 segundos.

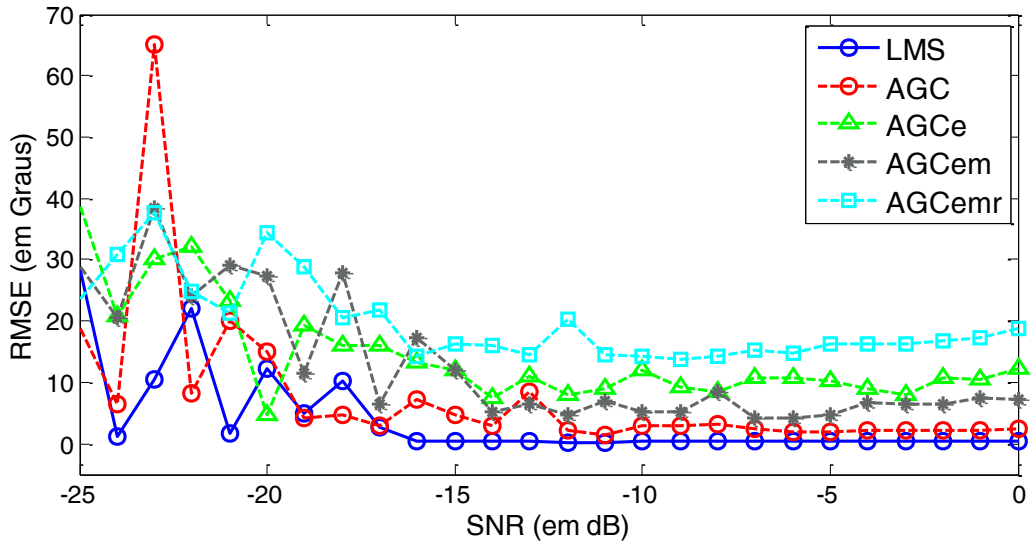

Figura 5.15: RMSE do bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e -20°.

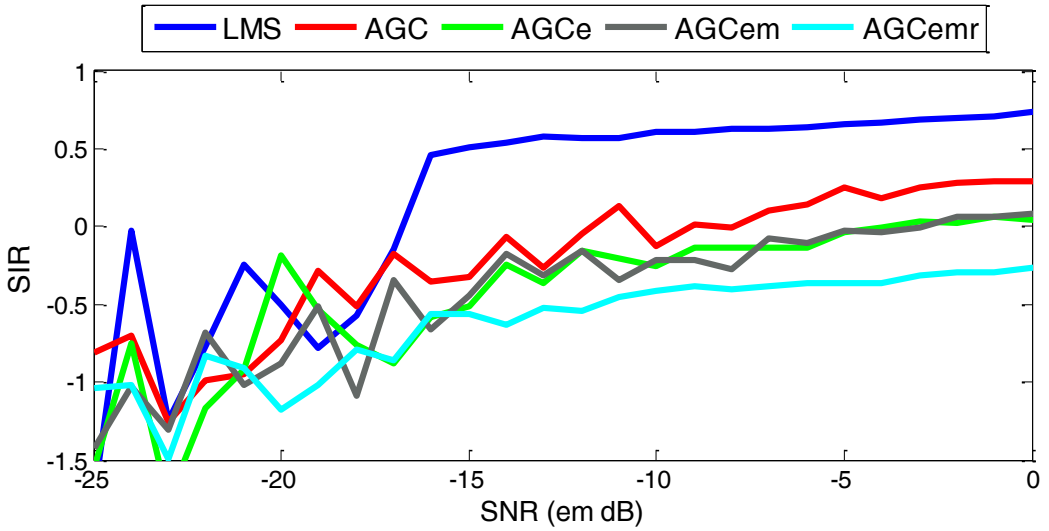

Figura 5.16: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e –20°.

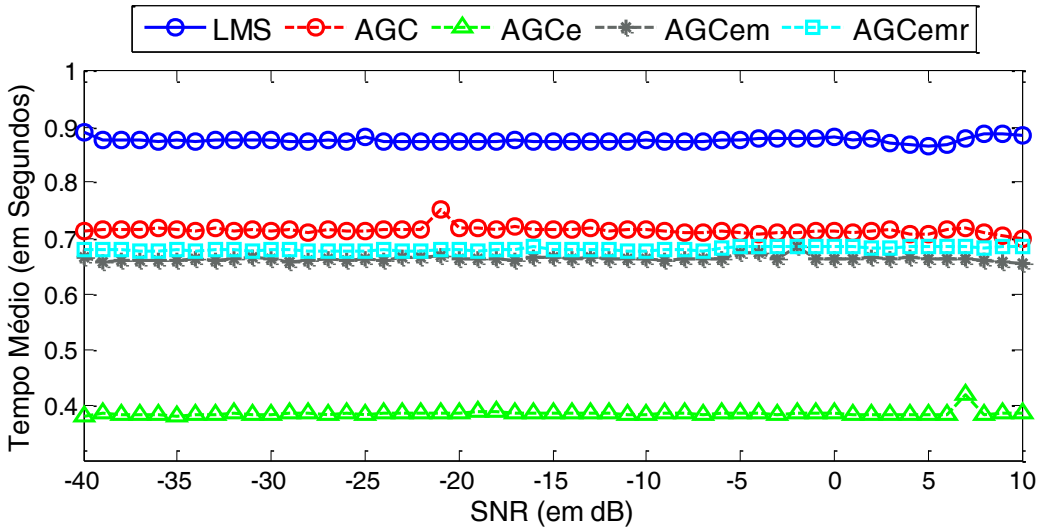

Figura 5.17: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e –20°.

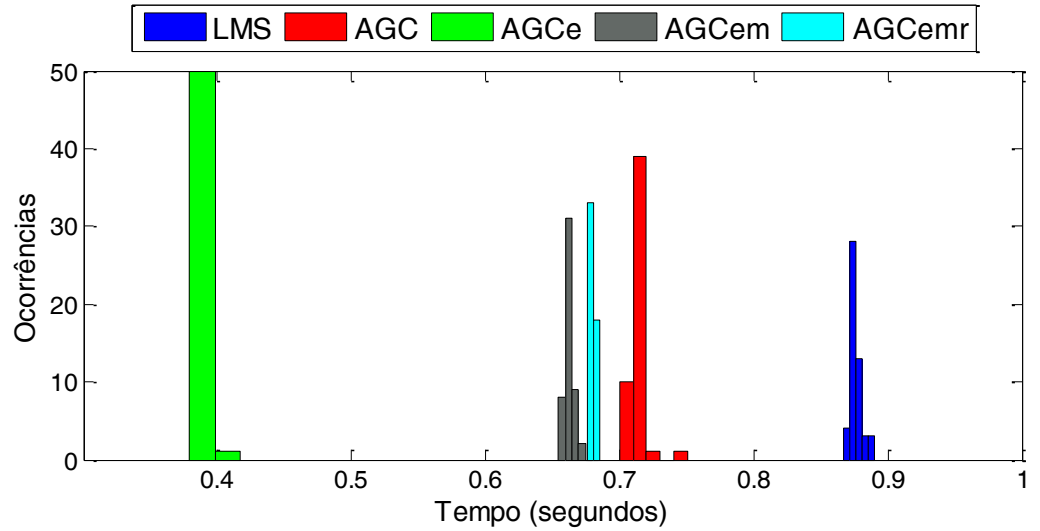

Figura 5.18: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para cinco fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20° e –20°.

Finalmente, para terminar a avaliação de desempenho dos algoritmos para fontes descorrelacionadas, alocamos no canal sete fontes estáticas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°. A Figura 5.19 mostra que novamente o AGC foi o método que mais se aproximou do método de referência, mas com um comportamento um pouco instável que oscilou entre dois e três graus de erro acima de –16 dB. O algoritmo LMS não conseguiu taxas nulas de RMSE, mas atingiu RMSE próximo de 0.5 graus a partir de –16 dB.

Rejeitando as fontes interferentes, a família de AGC's não foi tão efetiva como o algoritmo LMS. Neste caso, o AGC e o AGCem tiveram comportamentos similares mas que não lograram alcançar os níveis de SIR obtidos pelo algoritmo LMS. Estas condutas são expostas na Figura 5.20.

No caso dos tempos médios de convergência os resultados foram praticamente os mesmos, conforme mostra a Figura 5.21. Todos os métodos foram mais rápidos que o algoritmo LMS com modas de 0.69 segundos para o AGC, 0.38 segundos para o AGCe, 0.7 segundos para o AGCem e 0.66 segundos para o AGCemr, enquanto que o algoritmo LMS teve uma moda de 0.88 segundos conforme mostra a Figura 5.22.

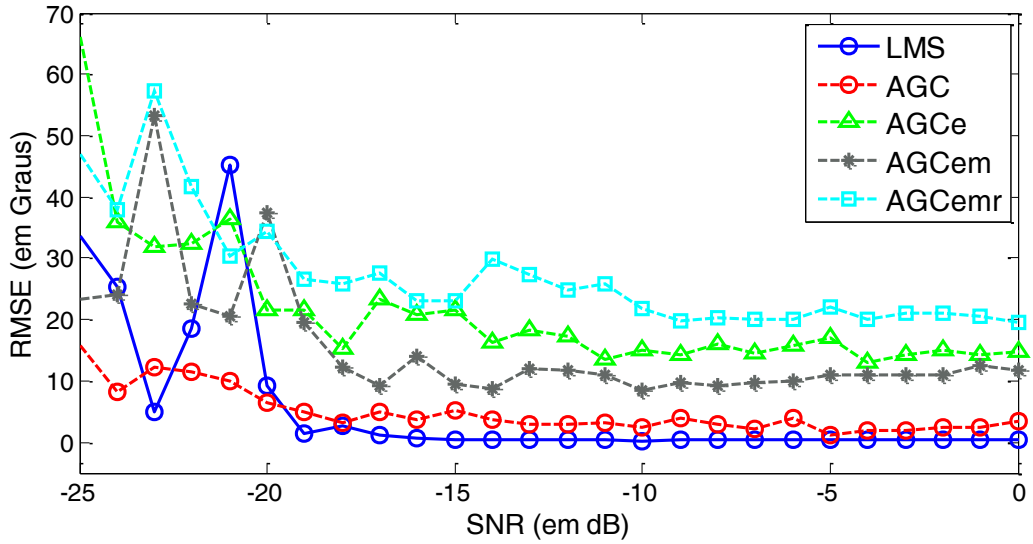

Figura 5.19: RMSE do bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°.

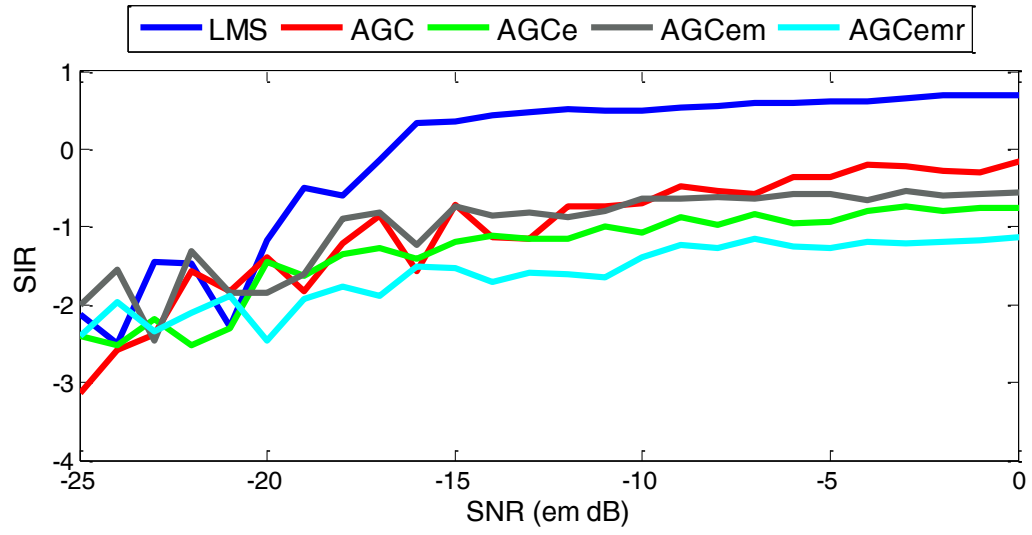

Figura 5.20: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°.

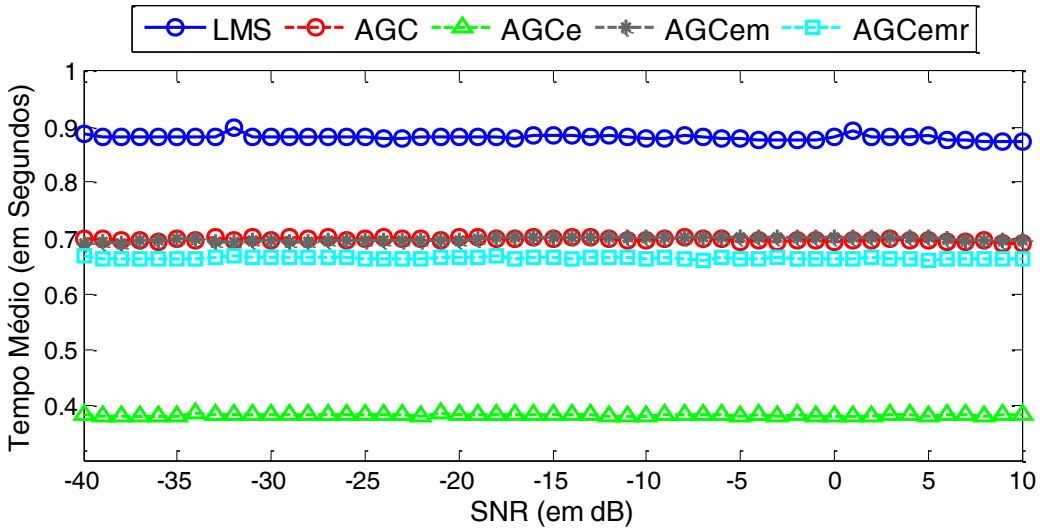

Figura 5.21: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°.

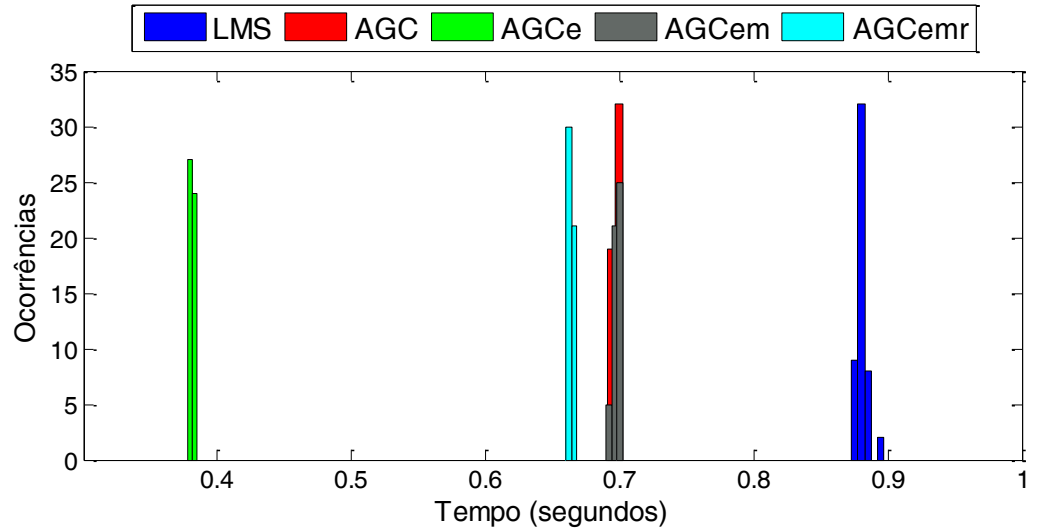

Figura 5.22: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para sete fontes descorrelacionadas localizadas em 10°, 40°, 60°, 20°, –20°, –40° e –60°.

5.5.2 Desempenho de Estimação para Fontes Estáticas Totalmente Correlacionadas

Para estes testes foram feitas as mesmas considerações que para os testes com fontes estáticas descorrelacionadas.

Primeiro, deve-se esclarecer que os sinais alocados no canal eram totalmente correlacionados, ou seja, que os sinais enviados tinham o mesmo sinal de mensagem e o mesmo código de espalhamento, porém com diferentes atrasos. Para determinar o RMSE dos testes, considerou-se que os algoritmos definiram um lóbulo principal apontando para uma das fontes sem importar a posição dela, isto é, que apontaram o maior ganho na direção da fonte que era mais fácil de identificar. Assim, para calcular o erro se determinou para qual direção estava apontado o maior feixe do arranjo e se identificou qual era a fonte mais próxima desse direcionamento. A diferença entre a direção na qual apontava o maior lóbulo e a posição da fonte mais próxima foi definida como erro de direcionamento.

O primeiro teste foi feito para duas fontes estáticas totalmente correlacionadas localizadas em 10° e 40°. O desempenho do algoritmo LMS de novo foi o melhor, obtendo RMSE próximos de zero graus desde –21 dB. Da mesma forma que para fontes descorrelacionadas, o AGC foi o método que mais se aproximou do método de referência com valores próximos de um grau a partir da SNR de –17 dB. Os AGCe e AGCem conseguiram taxas de erro baixas, porém instáveis e não tão baixas como aquelas dos outros métodos. Os resultados de RMSE dos algoritmos estão expostos na Figura 5.23.

Os níveis de SIR conseguidos por todos os métodos foram bastante satisfatórios, pois embora todas as fontes tivessem alta correlação, os algoritmos conseguiram definir um só lóbulo com alto ganho. Apesar de todos os algoritmos terem apresentado altos níveis de SIR, o algoritmo LMS atingiu valores que nenhum dos métodos conseguiu alcançar. Estes resultados são mostrados na Figura 5.24.

Agora no tocante aos tempos médios de convergência, a média se manteve conforme mostram as Figuras 5.25 e 5.26. Os tempos de toda a família de AGC foram menores que o do método de referência. Já o histograma da Figura 5.26 apresenta as modas dos tempos médios de convergência: 0.86 segundos para o algoritmo LMS, 0.70 segundos para o AGC, 0.37 segundos para o AGCe, 0.71 segundos para o AGCem, e 0.65 segundos para o AGCemr.

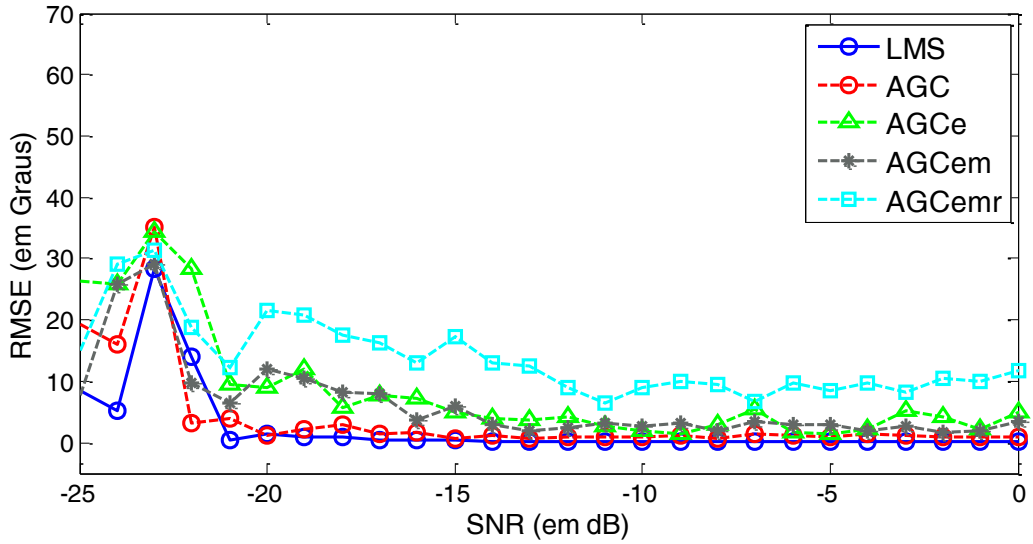

Figura 5.23: RMSE do bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°.

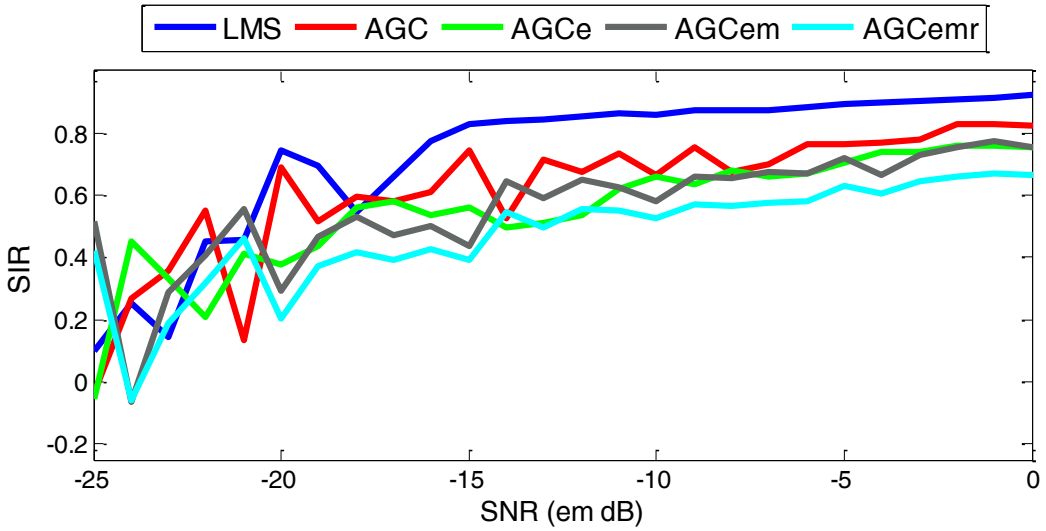

Figura 5.24: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°.

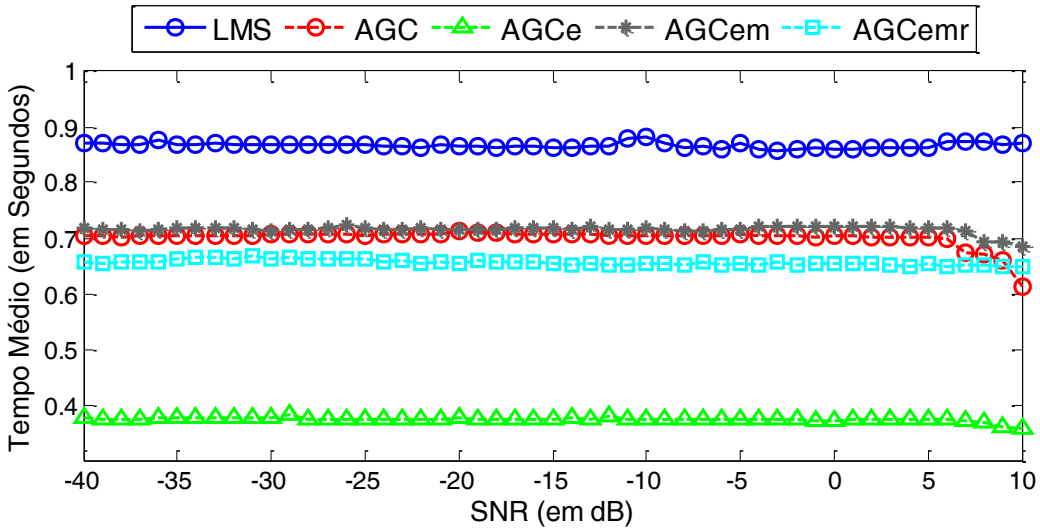

Figura 5.25: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°.

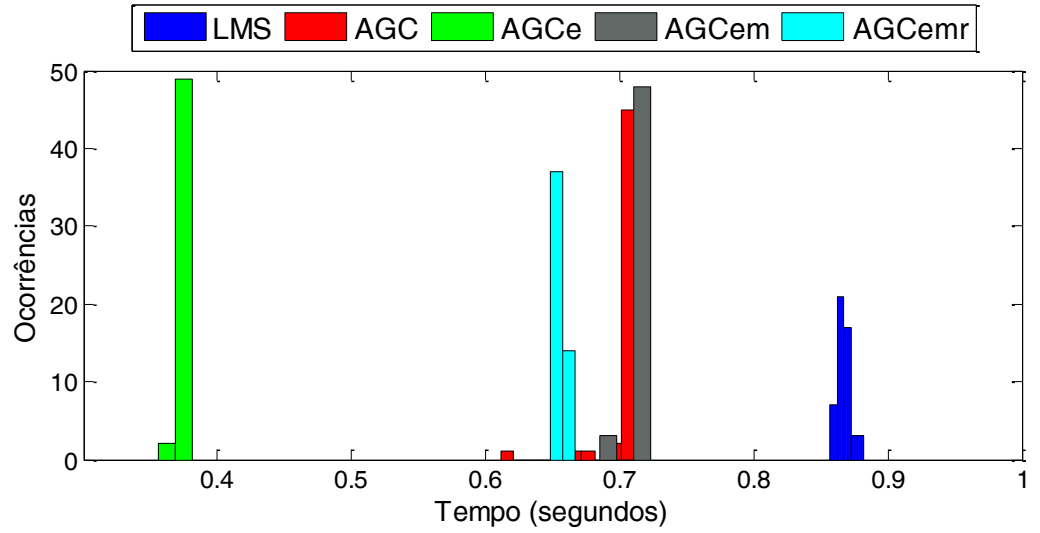

Figura 5.26: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para duas fontes totalmente correlacionadas localizadas em 10° e 40°.

Conforme mostra a Figura 5.27, aumentando a quantidade de fontes para três, o algoritmo LMS continuou sem conseguir taxas nulas de erro, mas seus valores de RMSE ficaram muito próximos de zero acima de –17 dB. Igualmente, AGC acompanhou o desempenho do algoritmo LMS com taxas de RMSE unitárias acima de –15 dB. Os AGCe e AGCem conseguiram taxas de erro próximas de cinco graus acima de –16 dB, mas com comportamento instável.

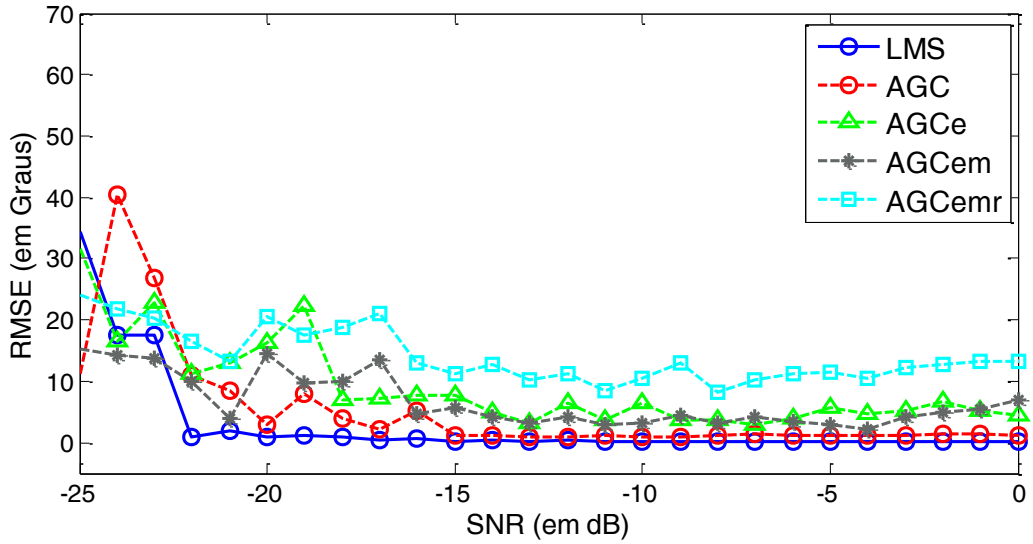

Figura 5.27: RMSE do bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°.

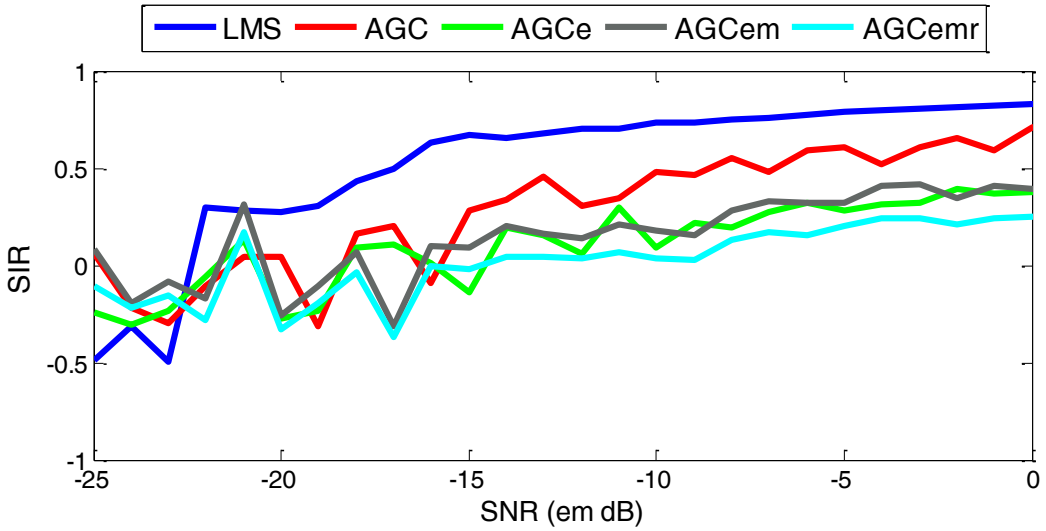

Figura 5.28: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°.

A Figura 5.28 mostra que os algoritmos conseguiram apontar um lóbulo de alto ganho em direção de uma das fontes, mas não conseguiram impor nulos nas direções das outras fontes, isto era de se esperar pela alta correlação dos sinais.

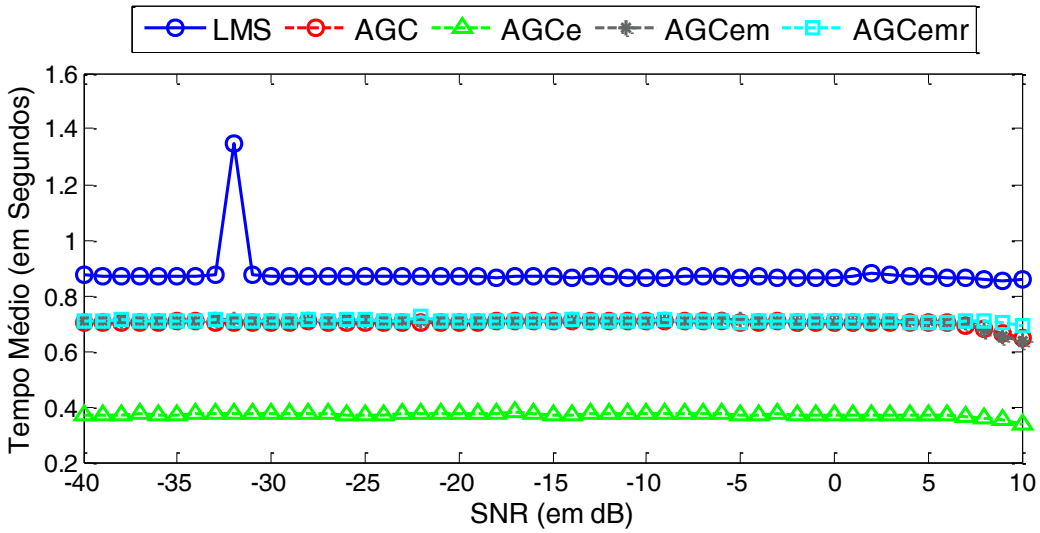

Figura 5.29: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°.

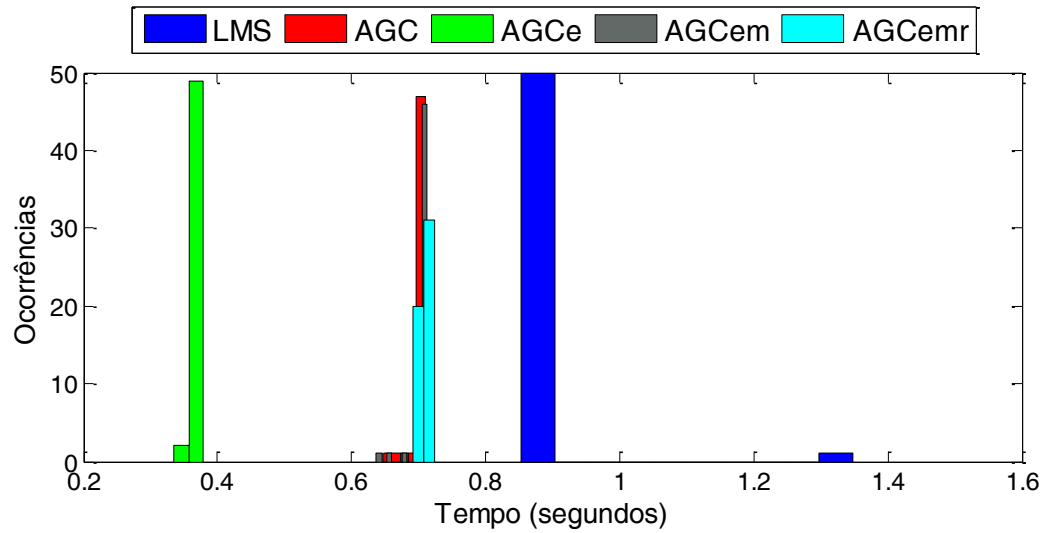

Figura 5.30: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para três fontes totalmente correlacionadas localizadas em 10°, 40° e 60°.

Os tempos médios mostrados nas Figuras 5.29 e 5.30 continuaram com os mesmos comportamentos, tendo sido toda a família de AGC mais rápida que o método de referência e com as mesmas modas nos tempos, apesar de um valor de pico no tempo médio do algoritmo LMS para SNR próxima de –30 dB.

Finalmente, alocamos no canal quatro fontes. Com o aumento do número de fontes, o limiar de SNR do método de referência se deslocou até –18 dB. Por outro lado, as taxas de

RMSE do AGC aumentaram, oscilando entre um e dois graus. Os outros métodos aumentam ligeiramente suas taxas de erro. As taxas de RMSE para quatro fontes são mostradas na Figura 5.31.

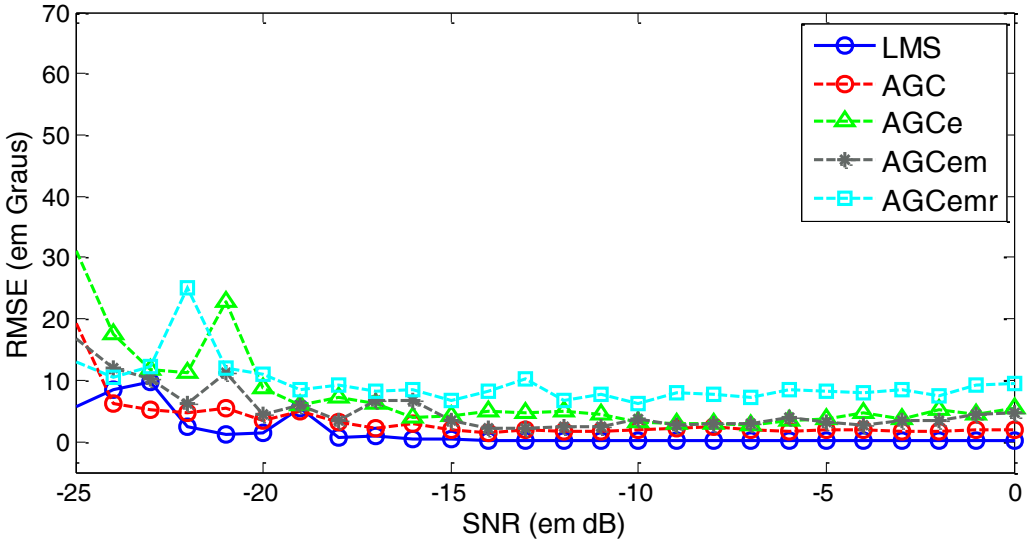

Figura 5.31: RMSE do bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°.

Os níveis de SIR da família de AGC diminuíram drasticamente, enquanto que os níveis do algoritmo LMS decresceram até quase a metade. Isto significa que os lóbulos laterais não foram anulados, devido à pela dificuldade para separar os sinais do canal. Os níveis de SIR dos algoritmos são mostrados na Figura 5.32.

Os tempos de todos os métodos apresentaram os mesmos comportamentos, com exceção do AGC que apresenta um aumento na velocidade a partir de –5 dB. Isto pode significar que, pela alta correlação, o algoritmo está gerando falsos positivos e por essa razão aumentaram-se ligeiramente as taxas de erro e diminuíram-se os níveis de SIR. Nas Figuras 5.33 e 5.34 mostram-se estes comportamentos.

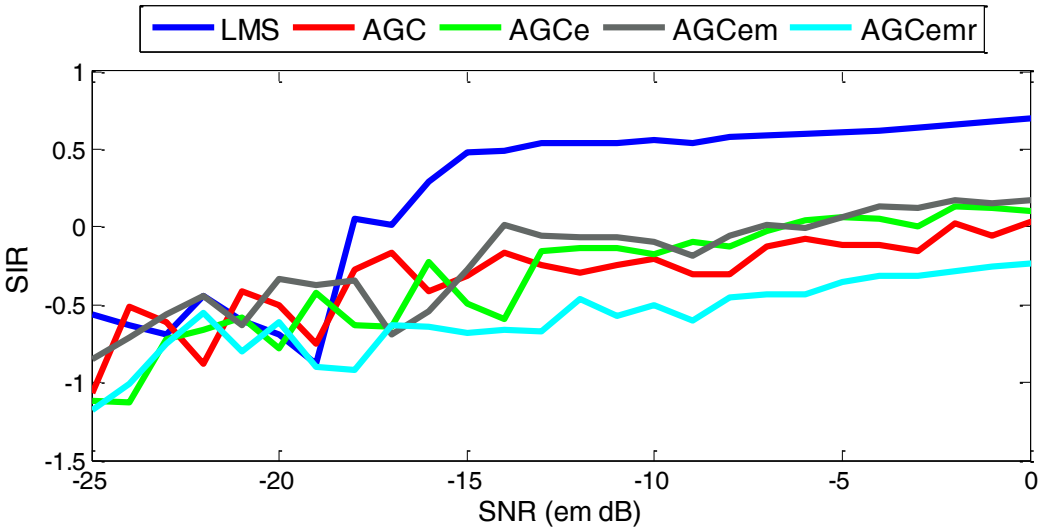

Figura 5.32: Resultado da saída da Relação Sinal-Interferência (SIR) do bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°.

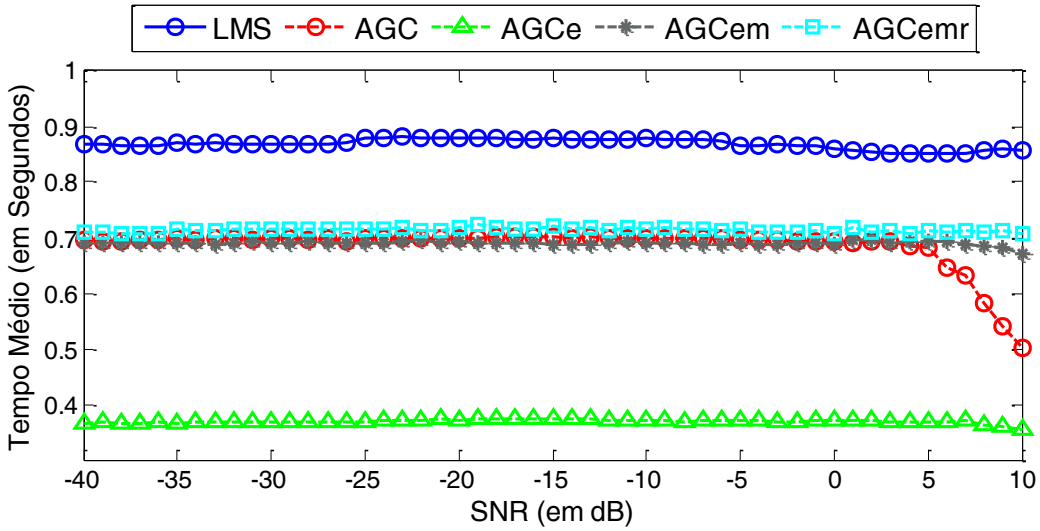

Figura 5.33: Tempos médios de execução para cada SNR das estimações de conformação de feixe feitas pelo bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°.

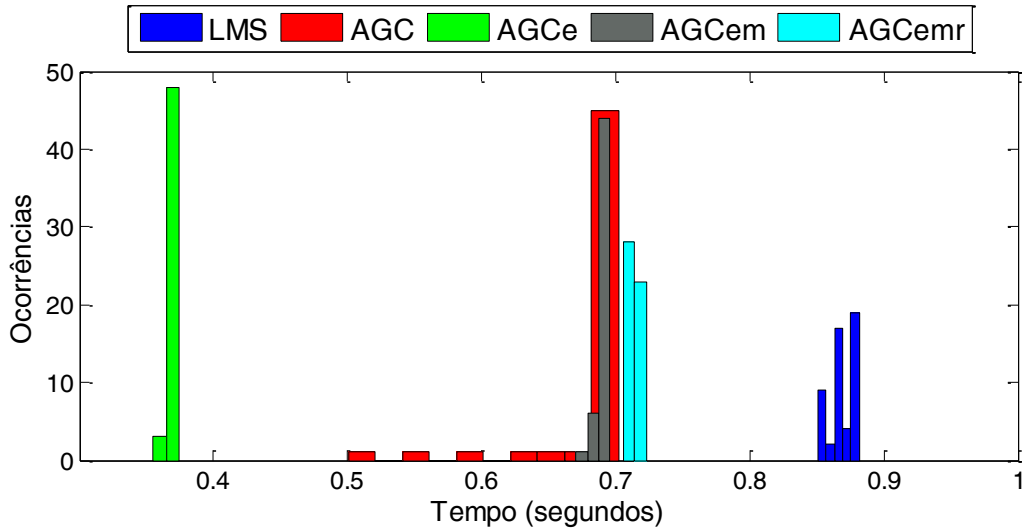

Figura 5.34: Histograma dos tempos médios de convergência do bloco Algoritmo Adaptativo para quatro fontes totalmente correlacionadas localizadas em 10°, 40°, 60° e -20°.

5.6 Detecção de Usuários em Movimento

Agora serão apresentados alguns resultados muito interessantes correspondentes à detecção da DOA e conformação do feixe quando as fontes estão se movimentando. Conforme já descrito no Capítulo 2, no bloco estimador de bit do sistema receptor proposto foi implementada a família de AGC para fins de cálculo do vetor de pesos W do arranjo. Mostrou-se que, para fontes estáticas, o algoritmo converge para a solução ótima assim que consegue o candidato que proporcione o menor valor da Raiz quadrada do erro quadrático médio (RMSE) conforme é mostrado nas seções anteriores.

No entanto, caso os usuários estejam em movimento, o cálculo dos pesos por parte do algoritmo adaptativo corresponde a uma verdadeira adaptação do algoritmo, já que tem que estimar os pesos de uma nova posição das fontes com base nos pesos estimados na última convergência. O método aplicado a nosso método de referência (*algoritmo LMS*) é o mesmo utilizado em (PUTTINI, 2006), onde o vetor de pesos W é atualizado para cada nova janela de dados de informação, a partir de seu último valor obtido pela convergência do algoritmo adaptativo. Enquanto que para a família de algoritmos propostos nesta tese (*família de AGC*), o vetor de pesos W é reiniciado para cada nova janela de dados, fazendo-se uma estimativa nova sem considerar o vetor de pesos calculado na estimação anterior para cada nova posição

da fonte, sendo esta uma justificativa das oscilações apresentadas nos resultados conseguidos, mas que não restringem ou afetam o ótimo desempenho conseguido pelo método proposto.

Para não restringir os resultados obtidos nas estimações de fontes em movimento ao desempenho dos blocos sincronizador e estimador de bit, se assume que o sinal de interesse está sincronizado e que o sinal de referência é estimado com exatidão, de modo que os resultados da estimação dependam só do algoritmo adaptativo. Nas simulações se considera que as fontes se movimentam com velocidade angular constante de 0.2 Graus/segundo sem apresentar mudanças abruptas e que o vetor de pesos é atualizado a cada 10 bits de informação, além disso o efeito Doppler é levado em conta para a frequência da fonte em movimento, mas, a mudança na frequência é praticamente nula podendo-se considerar depreciável.

Visando verificar a performance dos algoritmos propostos em ambientes de acesso múltiplo, são testadas diferentes situações para fontes em movimento. Os resultados conseguidos com a família de algoritmos propostos são comparados com os resultados do algoritmo LMS que é considerado como o método de referência.

5.6.1 Fonte de Interesse em Movimento

Primeiramente, na Figura 5.35, encontram-se representados o percurso da fonte de interesse ao longo do tempo, a posição das fontes interferentes e o direcionamento do feixe feito pelo algoritmo LMS. Nesta simulação só a fonte de interesse se movimenta a partir da posição inicial –20° até 40°, com SNR de 10 dB, com interferência de quatro fontes estáticas descorrelacionadas localizadas em –60°, –30°, 0° e 45°. Como pode ser percebido na Figura 5.35, o arranjo foi capaz de rastrear a fonte de interesse com eficiência aceitável, mas apresentou dificuldade na estimação quando a fonte de interesse passou perto de uma fonte interferente. A convergência lenta do algoritmo LMS permitiu fazer só 378 estimações da posição da fonte, conseguindo um RMSE de 3.30 graus. Por outro lado, o algoritmo LMS apresentou um comportamento estável e foi eficiente ao mitigar as fontes interferentes alcançando taxas de SIR altas, exceto quando a fonte de interesse se movimentou perto de uma fonte interferente, conforme mostra a Figura 5.36.

Da mesma forma, a Figura 5.37 mostra o percurso da fonte de interesse ao longo do tempo, a posição das fontes interferentes e o direcionamento do feixe feito pelo AGC, para uma fonte que se movimenta a partir da posição inicial –20° até 40°, com SNR de 10 dB, com interferência de quatro fontes estáticas descorrelacionadas localizadas em –60°, –30°, 0° e 45°. Estas condições de simulação são mantidas para todos os testes da família de AGC's.

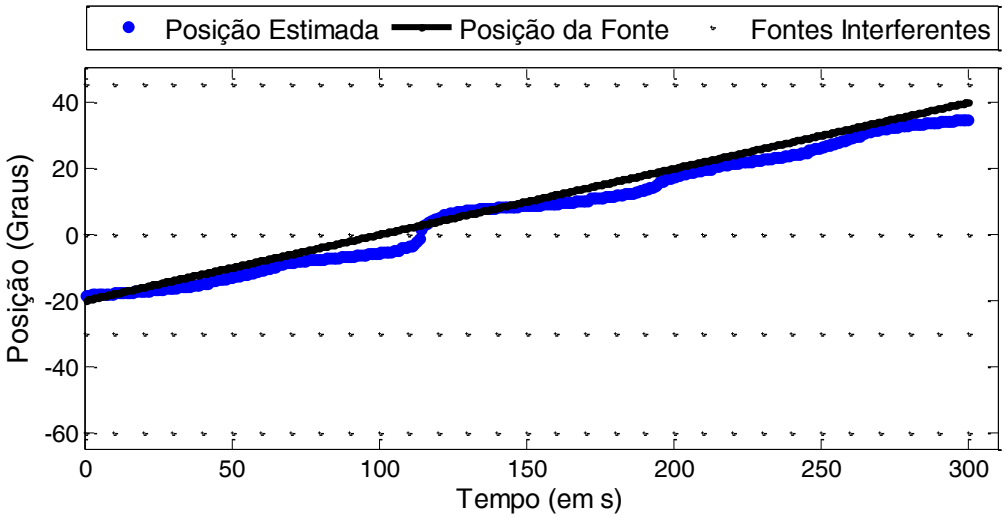

Figura 5.35: Rastreamento da fonte de interesse em movimento por parte do algoritmo LMS com quatro fontes interferentes estáticas localizadas em –65°, –30°, 0° e 45°.

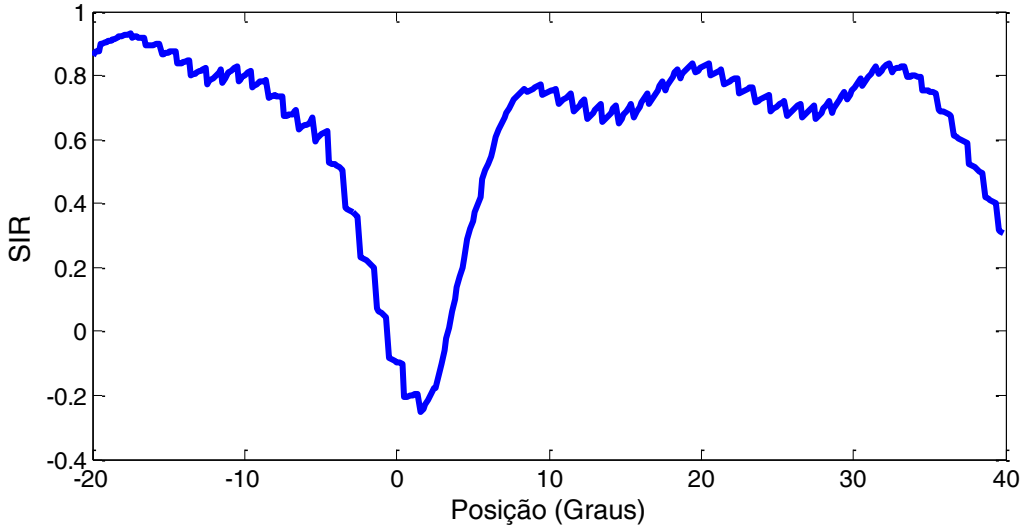

Figura 5.36: Resultado da saída da Relação Sinal-Interferência (SIR) do algoritmo LMS durante o rastreamento da fonte de interesse.

A Figura 5.37 mostra que o rastreamento do AGC foi um pouco instável, mas pela sua velocidade de convergência conseguiu acompanhar de perto a fonte em movimento, isto

permitiu uma maior quantidade de estimativas, neste caso 420, com um RMSE de 2.24 graus, aproximadamente um grau menor que o conseguido pelo algoritmo LMS. Ele apresentou uma leve degradação no desempenho apenas quando a fonte de interesse passou perto de uma fonte interferente, mas comparado com o algoritmo LMS, ele teve melhor comportamento. Quanto à capacidade de rejeitar as fontes interferentes, o AGC também apresentou um comportamento instável, mas os níveis de SIR foram comparáveis com os obtidos pelo algoritmo LMS, conforme a Figura 5.38. Estes resultados demostraram que o AGC mostrou melhor desempenho para este tipo de rastreamento, sendo mais adequado para aplicações em tempo real.

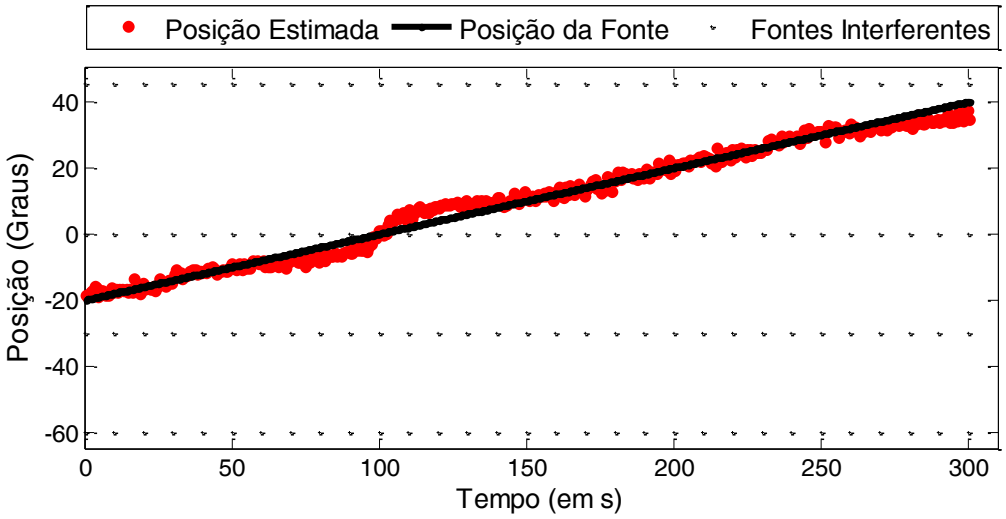

Figura 5.37: Rastreamento da fonte de interesse em movimento por parte do AGC com quatro fontes interferentes estáticas localizadas em –65°, –30°, 0° e 45°.

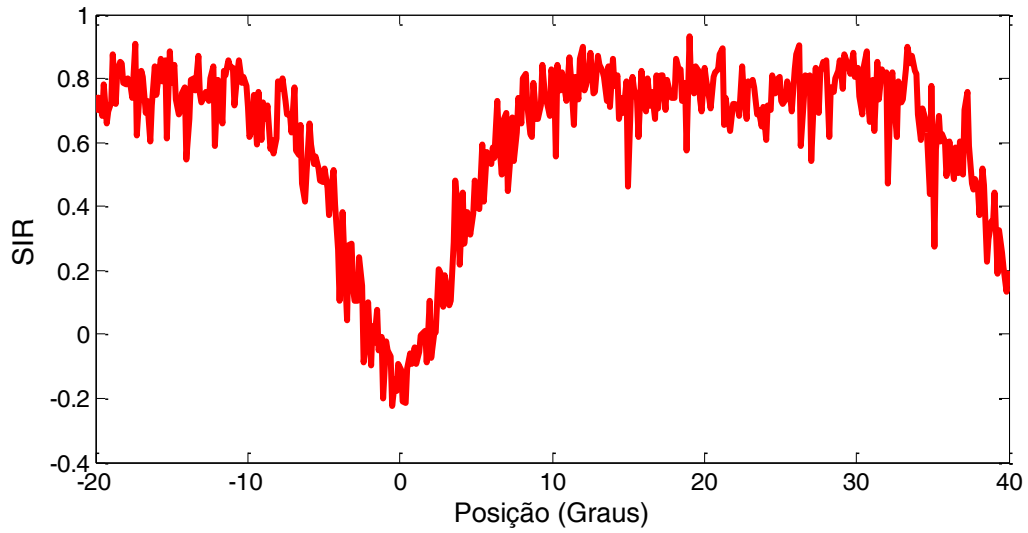

Figura 5.38: Resultado da saída da Relação Sinal-Interferência (SIR) do AGC durante o rastreamento da fonte de interesse.

Contrário ao bom desempenho apresentado pelo AGC, o restante da família de AGC não cumpriu com as expectativas de desempenho. Isto era previsível pelos resultados obtidos por estes algoritmos para fontes estáticas. A Figura 5.39 exibe os resultados das estimações feitas pelo Algoritmo Genético Compacto com Elitismo (AGCe). Nela as estimações de direcionamento ficaram muito dispersas e houve uma grande quantidade de estimativas longe da posição real da fonte. A grande velocidade de convergência permitiu gerar 777 estimativas, quase o dobro das feitas pelo AGC, mas a qualidade dos direcionamento não foi boa, com RMSE de 13.44 graus. Além disso, a capacidade de mitigar as fontes interferentes também foi reduzida, já que se conseguiu taxas de SIR menores que aquelas obtidas pelo AGC. A Figura 5.40 mostra os níveis de SIR obtidos pelo AGCe.

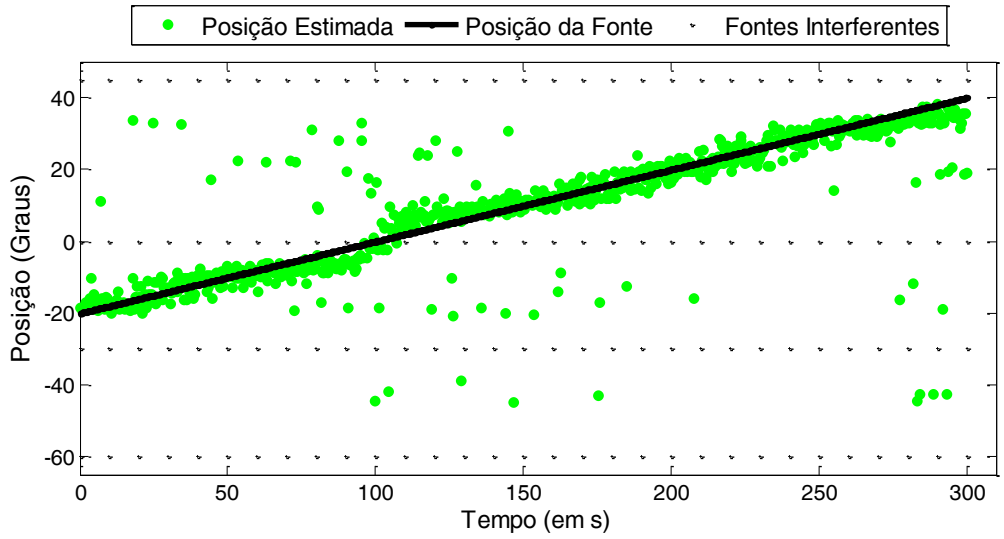

Figura 5.39: Rastreamento da fonte de interesse em movimento por parte do AGCe com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°.

Do mesmo modo, o Algoritmo Genético Compacto com Elitismo e Mutação (AGCem) não apresentou bom acompanhamento da fonte em movimento, já que também apontou muito distante da posição real da fonte, o que logicamente aumentou o RMSE e diminuiu os níveis de SIR. O RMSE conseguido pelo AGCem foi de 11.68 graus para um total de 412 estimativas. Os resultados do rastreamento do AGCem são expostos na Figura 5.41 e os níveis de SIR conseguidos são mostrados na Figura 5.42.

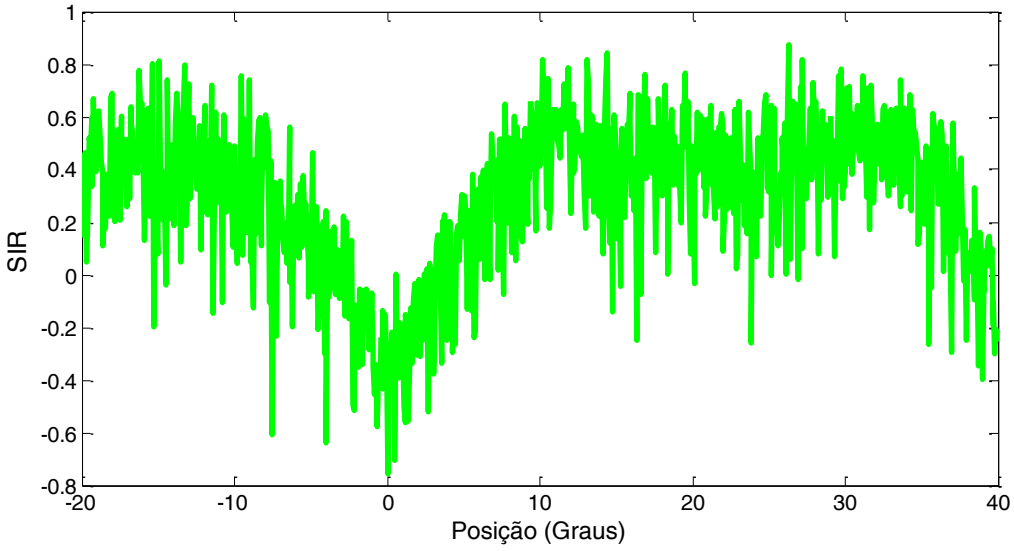

Figura 5.40: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCe durante o rastreamento da fonte de interesse.

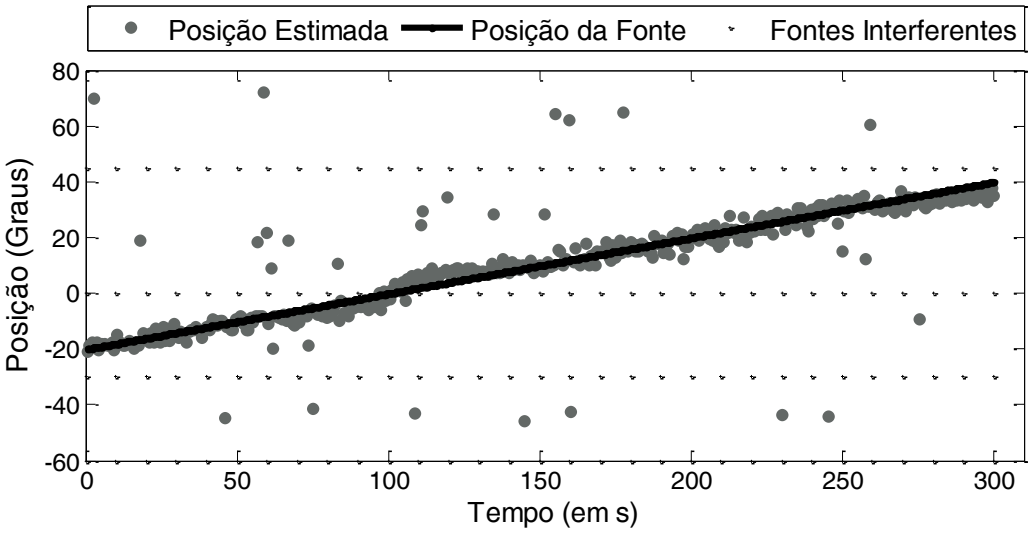

Figura 5.41: Rastreamento da fonte de interesse em movimento por parte do AGCem com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°.

Finalmente, o Algoritmo Genético Compacto com Elitismo, Mutação e Reamostragem (AGCemr) apresenta o pior desempenho da família de AGC, conseguindo um RMSE de 16.50 graus com 418 estimativas. A alta taxa de erro implica baixos níveis de SIR, logo o AGCemr não conseguiu rejeitar eficientemente as fontes interferentes. As Figuras 5.43 e 5.44 mostram respectivamente as estimativas de direcionamento e os níveis de SIR conseguidos.

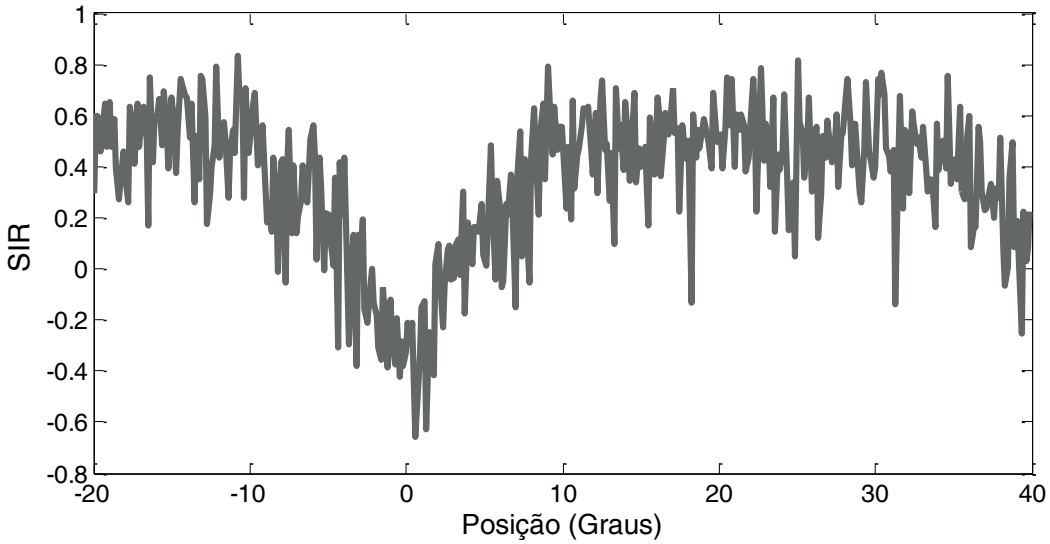

Figura 5.42: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCem durante o rastreamento da fonte de interesse.

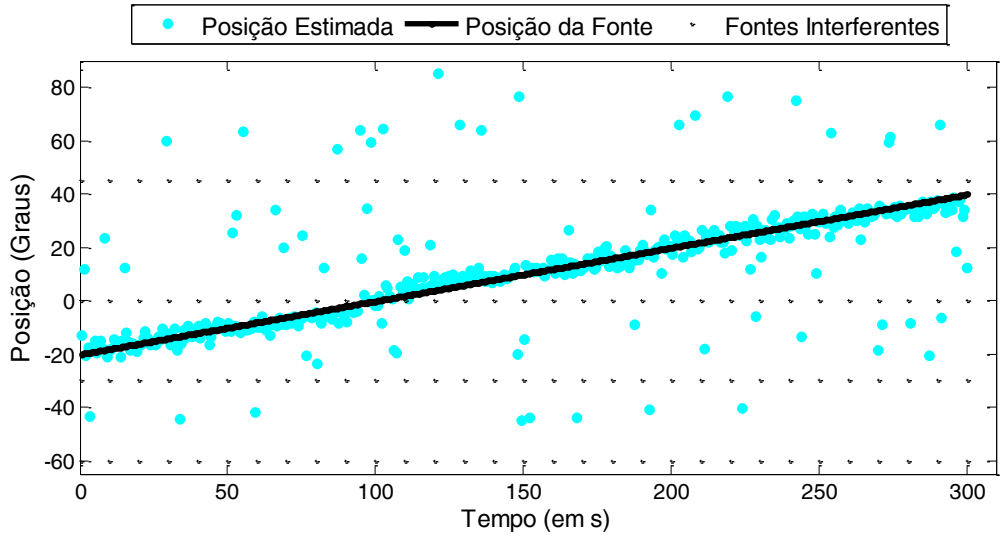

Figura 5.43: Rastreamento da fonte de interesse em movimento por parte do AGCemr com quatro fontes interferentes estáticas localizadas em −65°, −30°, 0° e 45°.

Deve-se ressaltar que, apesar de que alguns dos algoritmos apresentarem taxas de erro altas, os níveis de SIR conseguidos pela maioria deles foram aceitáveis ante a literatura.

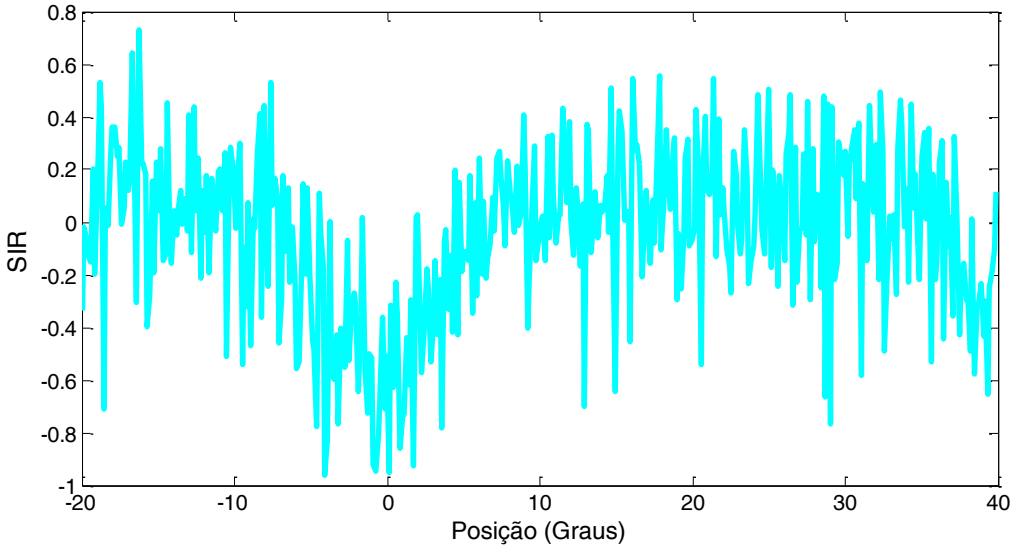

Figura 5.44: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCemr durante o rastreamento da fonte de interesse.

5.6.2 Todas as Fontes em Movimento

Para estes testes considerou-se que todas as fontes alocadas no canal se comunicação se movimentavam com velocidade constante de 0.2 graus/segundo e sem mudanças abruptas conforme foi dito anteriormente. A fonte de interesse novamente se movimentou a partir da posição inicial –20° até 40°, com SNR de 10 dB, com interferência de quatro fontes descorrelacionadas localizadas inicialmente em –60°, –30°, 0° e 45° e que se movimentaram em diferentes direções. Estas considerações foram mantidas para todas as simulações.

O algoritmo LMS exibiu um bom desempenho neste teste, conseguindo apontar o feixe mesmo quando a fonte de interesse passou perto de uma fonte interferente. O RMSE conseguido pelo algoritmo LMS foi de 3.26 graus, similar ao RMSE obtido nos testes em que apenas a fonte de interesse se movimentava. A quantidade de estimativas foi de 377, também similar às estimativas feitas no teste passado. Os níveis de SIR conseguidos foram altos, só diminuindo quando a fonte interferente passou perto de uma fonte interferente, situação na que o lóbulo principal também beneficiava a fonte interferente. A Figura 5.45 mostra as estimativas da posição e a Figura 5.46 mostra o SIR da simulação.

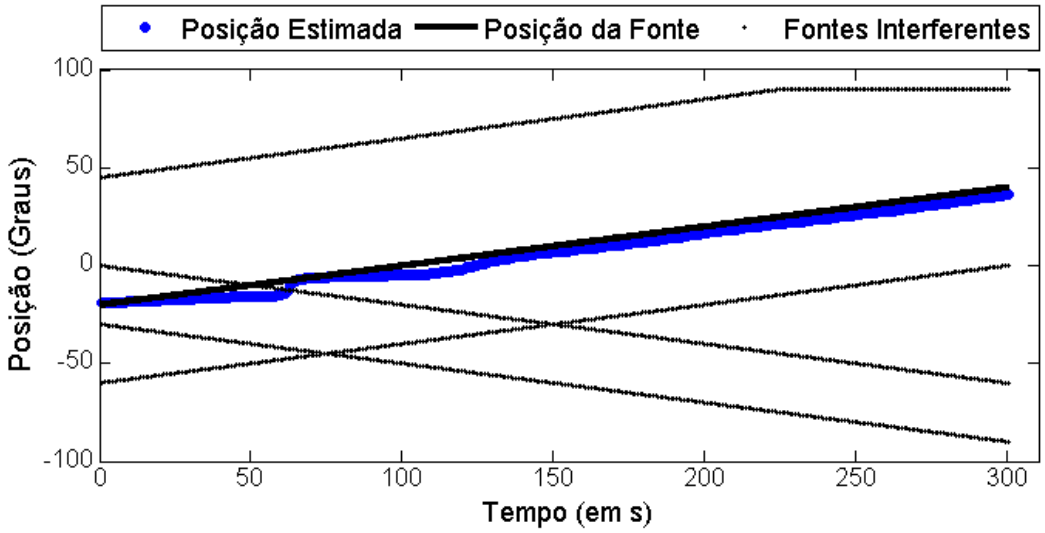

Figura 5.45: Rastreamento da fonte de interesse em movimento por parte do Algoritmo LMS com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°.

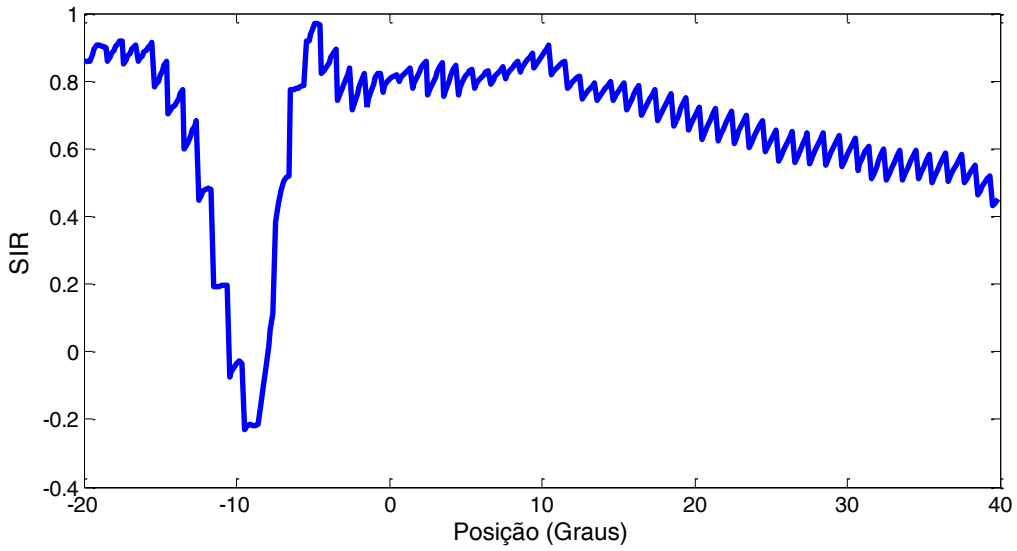

Figura 5.46: Resultado da saída da Relação Sinal-Interferência (SIR) do Algoritmo LMS durante o rastreamento da fonte de interesse.

O desempenho do AGC nestes testes foi surpreendente, conseguindo rastrear a fonte interferente com significativa precisão, obtendo um RMSE de 1.68 graus com 417 estimativas, além de ter conseguido rejeitar efetivamente as fontes interferentes móveis. A Figura 5.47 mostra a maior exatidão do rastreamento e a Figura 5.48 expõe a efetividade na filtragem.

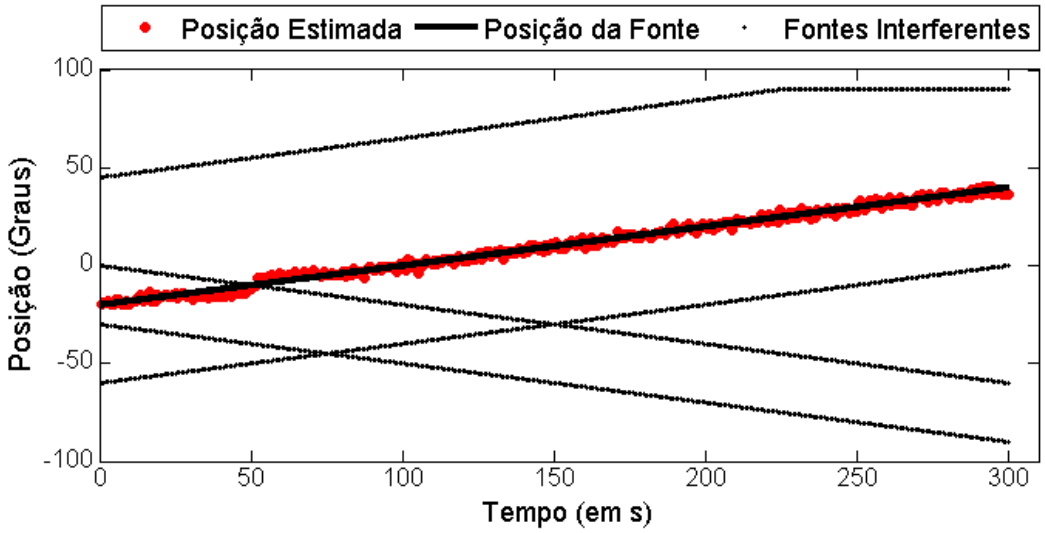

Figura 5.47: Rastreamento da fonte de interesse em movimento por parte do AGC com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°.

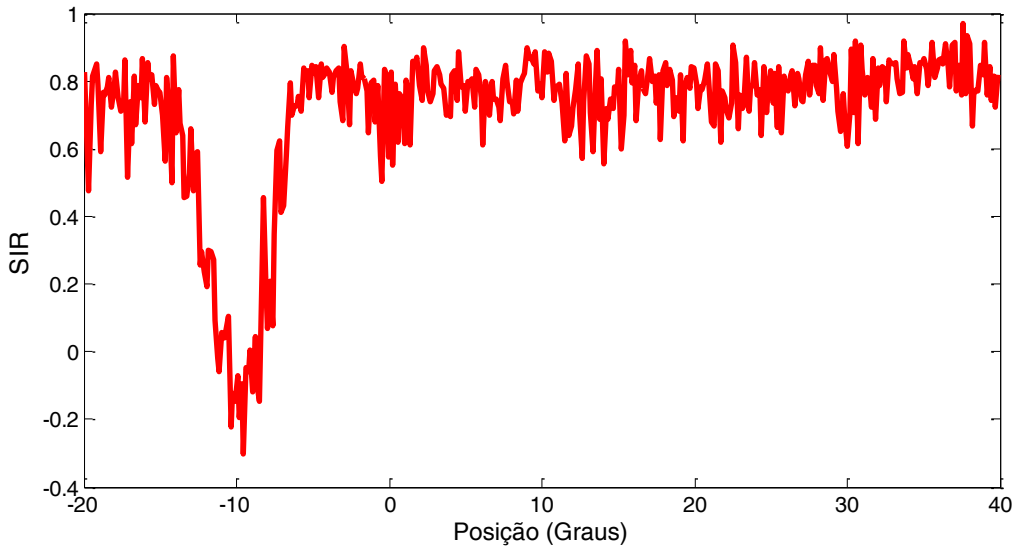

Figura 5.48: Resultado da saída da Relação Sinal-Interferência (SIR) do AGC durante o rastreamento da fonte de interesse.

Do mesmo jeito que com os testes da seção anterior, o AGCe apresentou a maior quantidade de estimativas, mas isso precisamente não indicou que a qualidade de rastreamento fosse boa. Pelo contrário, a taxa de RMSE aumentou consideravelmente, alcançando o valor de 12.52 graus com 779 estimativas. A rejeição das fontes interferentes também apresentou uma piora. As estimações da posição e os níveis de SIR alcançados são expostos nas Figuras 5.49 e 5.50 respectivamente.

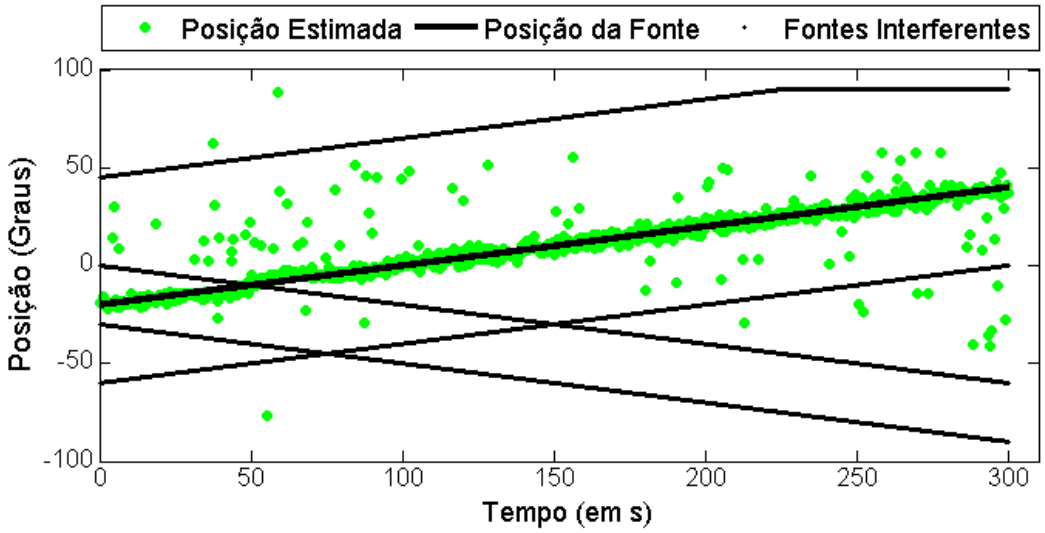

Figura 5.49: Rastreamento da fonte de interesse em movimento por parte do AGCe com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°.

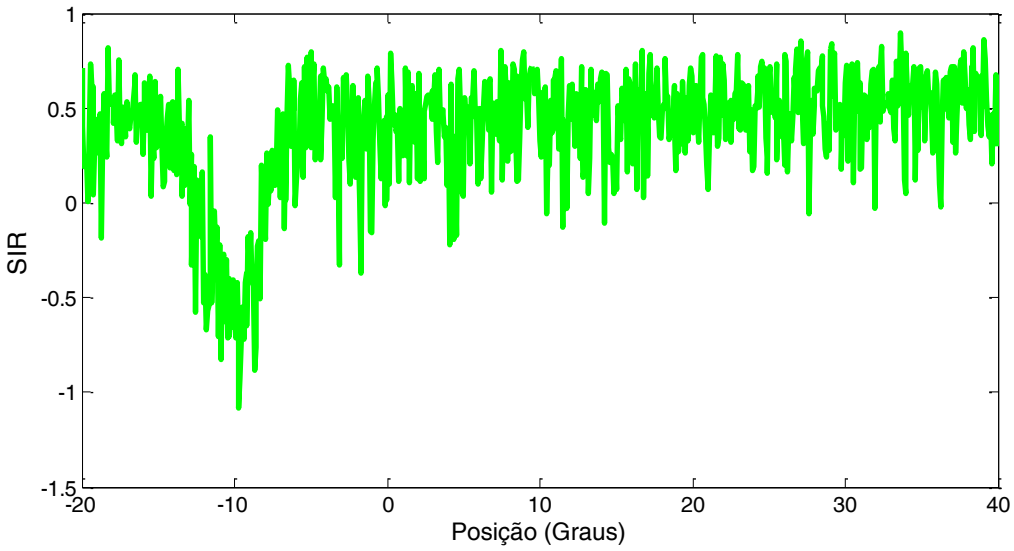

Figura 5.50: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCe durante o rastreamento da fonte de interesse.

O AGCem apresentou RMSE de 10.37 graus com 409 estimativas e certamente errou menos que o AGCe, mas ficou longe de um ótimo desempenho. A Figura 5.51 mostra o rastreamento do algoritmo. No caso da rejeição das fontes o AGCem mostrou melhoria aumentando níveis de SIR conforme mostra a Figura 5.52.

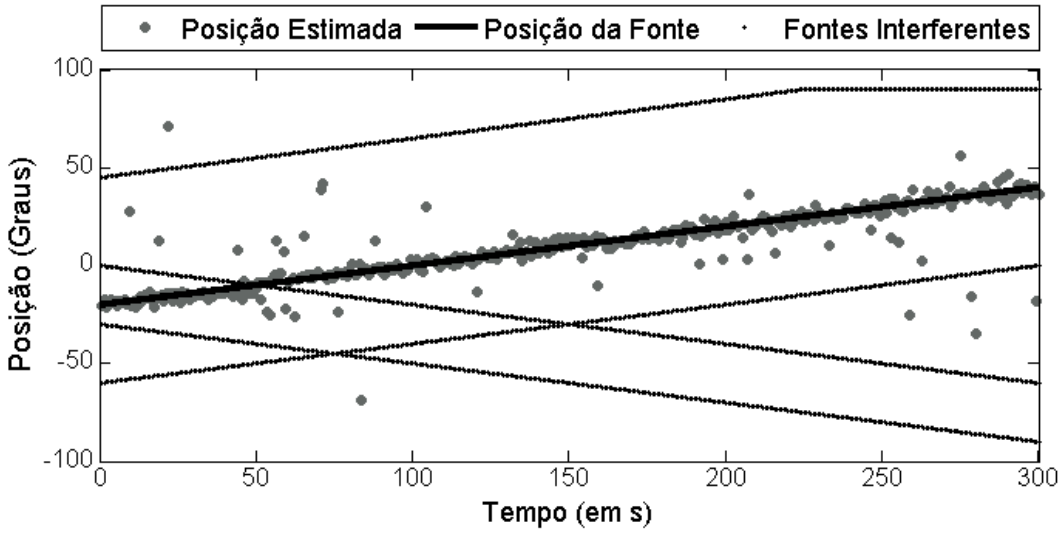

Figura 5.51: Rastreamento da fonte de interesse em movimento por parte do AGCem com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°.

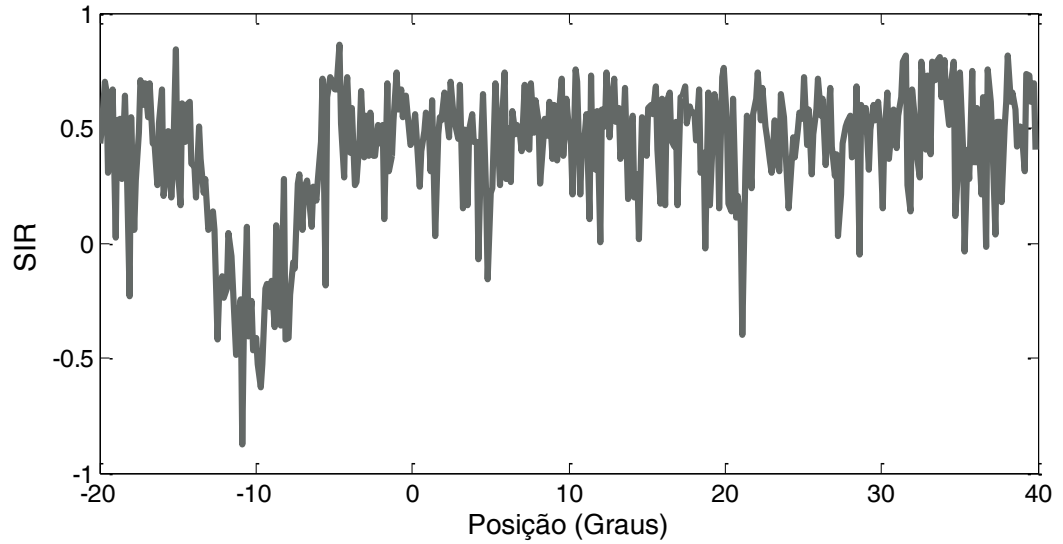

Figura 5.52: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCem durante o rastreamento da fonte de interesse.

Por último, o AGCemr novamente conseguiu a pior taxa de erro, com um RMSE de 16.23 graus com 435 estimativas. A alta taxa de erro refletiu na instabilidade dos níveis de SIR conseguidos pelo algoritmo. A Figura 5.53 mostra ainda que o algoritmo conseguiu acompanhar o movimento da fonte de interesse, mas apresentou uma grande quantidade de estimativas erradas, que logicamente impediram uma filtragem espacial ótima das fontes interferentes, implicando instabilidade e baixos níveis de SIR, conforme mostra a Figura 5.54.

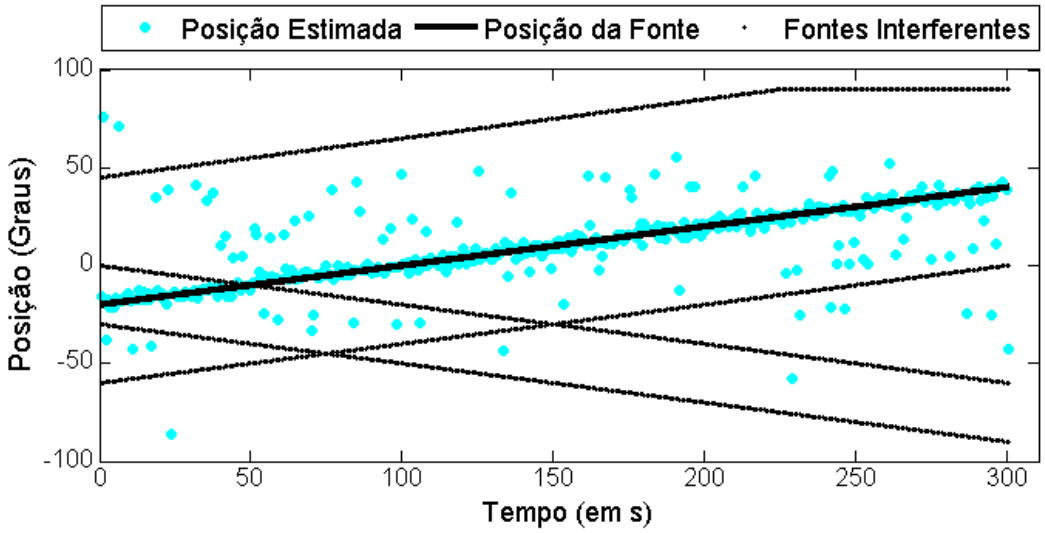

Figura 5.53: Rastreamento da fonte de interesse em movimento por parte do AGCemr com quatro fontes interferentes móveis que iniciam seu movimento em −65°, −30°, 0° e 45°.

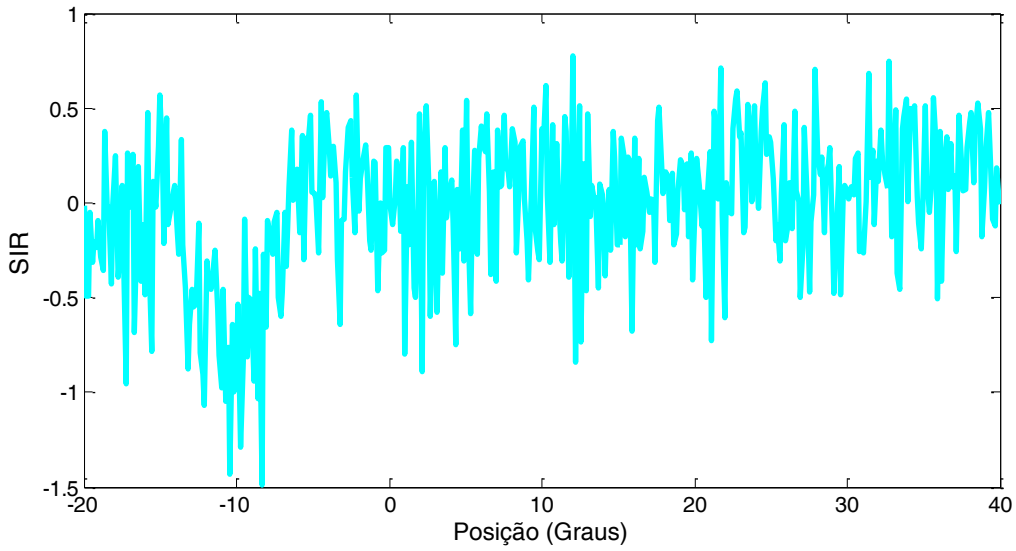

Figura 5.54: Resultado da saída da Relação Sinal-Interferência (SIR) do AGCemr durante o rastreamento da fonte de interesse.

5.7 Análise de Resultados

Primeiramente, reitera-se que o desempenho do bloco sincronizador está diretamente relacionado com a sua simplicidade de implementação. O desempenho na detecção de atrasos para fontes descorrelacionadas é muito limitado, uma vez que ele foi capaz de detectar o atraso da fonte de interesse de forma aceitável quando no canal são alocados no máximo três

fontes. Isso se deve ao fato de que todos os sinais inseridos no canal apresentam atrasos, o que implica em perda de ortogonalidade e aumento da correlação dos sinais. Além disso, o aumento da quantidade de sinais no canal, aumenta ainda mais a correlação, dificultando a detecção do atraso da fonte de interesse por parte do sincronizador. No caso de fontes totalmente correlacionadas, o sincronizador não conseguiu detectar o atraso da fonte de interesse em nenhum dos casos testados, isso aconteceu justamente devido à alta correlação existente entre as fontes e à falta de ortogonalidade entre os sinais.

Os resultados conseguidos pelo bloco estimador de bit para fontes descorrelacionadas foram muito satisfatórios, conseguindo taxas BER nulas para todos os testes realizados apesar de apenas a fonte de interesse estar sincronizada. Por outro lado, os resultados para fontes totalmente correlacionadas apresentaram taxas BER superiores a 50%, diretamente proporcionais à quantidade de fontes alocadas no canal. Entretanto, estes resultados já eram esperados, uma vez que o bloco estimador de bit funciona sob o princípio da ortogonalidade entre os sinais nos sistemas CDMA. Ao não se cumprir este princípio, a interferência intersimbólica aumenta, dificultando a recuperação do sinal de interesse.

Para o bloco Algoritmo Adaptativo, foram feitos testes adicionais para determinar primeiramente o tamanho ideal da população a ser implementado na família de AGC para conformação de feixe do arranjo. Os resultados determinaram que uma população de 20 indivíduos no AGC conseguiu as menores taxas de RMSE na maioria dos casos. Adicionalmente, observou-se que o tamanho da população tem relação direta com a velocidade de convergência do AGC, o que é mais evidente quando a SNR era de 10 dB. A explicação para este resultado é que uma população com mais indivíduos promove mais mudanças no vetor de probabilidades, acelerando a convergência do AGC. Por outro lado, uma população maior implicaria numa maior quantidade de gerações para assegurar a convergência do vetor de probabilidade e garantir baixas taxas de RMSE, provocando uma desaceleração na velocidade de convergência do AGC.

Para fontes descorrelacionas, o algoritmo LMS conseguiu taxas de RMSE nulas quando no canal foram alocadas até três fontes e, com o aumento no número de fontes, o erro do algoritmo LMS aumentou ligeiramente. O algoritmo LMS apresentou bom comportamento anulando as fontes interferentes, tal que os níveis de SIR foram maiores em todos os testes realizados, diminuindo com o aumento na quantidades de fontes no canal. Por outro lado, este algoritmo obteve os tempos médios de convergência mais altos de todos os métodos estudados, os quais não diminuíram com o aumento da SNR. O AGC foi o único algoritmo

que conseguiu desempenho semelhante ao do LMS, com taxas de RMSE maiores porém muito próximas das obtidas pelo método de referência. Os níveis de SIR alcançados pelo AGC ficaram próximos daqueles do LMS para até três fontes no canal. À medida que se aumentou o número de assinantes, os níveis de SIR diminuíram, indicando que o algoritmo não conseguiu impor nulos nas direções das fontes interferentes. Os tempos médios de execução do AGC foram menores do que os do LMS e parecidos com aqueles gastos pelo AGCem e o AGCemr. A exatidão nas estimativas de conformação de feixe do AGCe, AGCem e do AGCemr foi afetada pela utilização de elitismo nos algoritmos, já que o elitismo preserva o melhor indivíduo obtido na primeira geração e atualiza o vetor de probabilidade com base nele. Esse indivíduo só é trocado quando o algoritmo gera um candidato melhor que ele mas, caso o algoritmo demore para encontrar um melhor candidato, o vetor de probabilidade continua sendo atualizado em termos desse individuo, aumentando o risco de gerar clones do indivíduo ganhador, levando a otimização para um mínimo local. A introdução de reamostragem na família de AGC tem a função de impedir a geração de clones, mas a taxa de reamostragem usada nos testes foi muito alta e os resultados foram afetados.

Para o caso de fontes estáticas correlacionadas, os algoritmos implementados no bloco Algoritmo Adaptativo, conseguiram desempenhos muito parecidos com os alcançados para fontes estáticas descorrelacionadas. A exceção foi o algoritmo LMS que não conseguiu valores nulos de RMSE. No tocante ao SIR e aos tempos médios de convergência, manteve-se o comportamento observado para fontes descorrelacionadas.

Os resultados obtidos para fontes estáticas estão refletidos nos resultados conseguidos para fontes em movimento, tal que o AGC mostrou o melhor desempenho no rastreamento da fonte de interesse em todas as situações de fontes em movimento, ou seja, o comportamento do AGC é muito melhor em situações de fontes em movimento apesar de reiniciar o seu vetor de pesos W a cada nova posição da fonte de interesse.

6 Conclusões e Trabalhos futuros

6.1 Introdução

A estimação de ângulos de incidência (DOA) e a conformação de feixe (*beamforming*) são campos que apresentam muita aplicabilidade em diferentes áreas da tecnologia e que ainda podem ser mais exploradas. Uma das principais razões pelas quais se restringe o uso de sistemas de conformação de feixe é o alto custo computacional, que implica baixas velocidades de convergência, e que, portanto, impede que sejam utilizados em aplicações em tempo real.

Diante desta perspectiva, o algoritmo da Mínima Média Quadrática (LMS) possui um desempenho de estimação excelente. Apesar disso, a sua característica de convergência lenta limita a sua aplicabilidade em sistemas de rastreamento de alvos em movimento. Para tentar superar esta limitação, foram aplicados métodos heurísticos como a família de Algoritmos Genéticos Compactos (AGC's).

Neste trabalho, foi apresentada uma adaptação da família de AGC's para o problema da conformação de feixe, tal que estes algoritmos apresentaram uma complexidade computacional significativamente menor, mantendo uma qualidade competitiva nas estimativas.

6.2 Conclusões

Nos resultados de desempenho de estimação apresentados no Capítulo 5, pôde-se verificar que o desempenho do algoritmo Genético Compacto (AGC) proposto conseguiu se aproximar daquele do algoritmo da Mínima Média Quadrática (LMS) em todos os casos de fontes estáticas descorrelacionadas e correlacionadas, conseguindo taxas de RMSE e limiares de Relação Sinal-Ruído (SNR) marginalmente maiores. Observou-se, entretanto, que as alterações do AGC não surtiram bom resultado na precisão do direcionamento do arranjo. No tocante à capacidade de rejeição das fontes interferentes, só o AGC conseguiu níveis de Relação Sinal-Interferência (SIR) semelhantes aos alcançados pelo algoritmo LMS.

Entretanto, enquanto os níveis de SIR do método de referência diminuíram com o aumento do número de fontes, os níveis obtidos pelo AGC foram mais afetados. Contudo, a família de AGC possibilitou significativa redução no tempo de estimação, o que indica seu uso em sistemas de direcionamento em tempo real.

Tendo em conta esta aplicação, foram simulados casos de fontes em movimento, constatando-se que o desempenho de estimação do AGC foi superior aos dos outros métodos utilizados, rastreando com maior exatidão o alvo, conseguindo convergir para uma maior quantidade de estimativas que o algoritmo LMS e alcançando as menores taxas de RMSE dentre todos os algoritmos estudados.

6.3 Trabalhos Futuros

Um estudo acerca da relação entre o tamanho do arranjo, a resolução e a complexidade computacional da família de Algoritmos Genéticos Compactos (AGC) é recomendado, com os objetivos de aumentar a exatidão das estimativas, diminuir a Raiz do Erro Quadrático Médio RMSE e aumentar os níveis de SIR.

Ainda é necessário determinar os parâmetros adequados para cada algoritmo da família de AGC's, já que cada um deles têm características diferentes que precisam ser aproveitadas. Por isso, é proposto um estudo que avalie o desempenho dos algoritmos para diferentes combinações de parâmetros, de tal forma a determinar os parâmetros ótimos que levem à melhor relação custo benefício.

Uma pesquisa interessante que pode ser desenvolvida é utilizar a família de AGC para a estimação cega dos Ângulos de Incidência num arranjo de antenas, já que neste trabalho só se considerou estimação com conhecimento da sequência de espalhamento em sistemas CDMA. Além disso, podem ser testados outros métodos determinísticos de convergência rápida como, por exemplo, o algoritmo da Mínima Média Quadrática Recursiva (RLS), comparando-o com os método aqui estudados. Por outro lado, é proposto um estudo de detecção de fontes em movimento utilizando o AGC, mas usando o histórico do vetor de probabilidade ou o histórico do vetor de pesos das posições anteriores da fonte de interesse.

Referências bibliográficas

ALVES, C. A. *Análise Teórica e Experimental de Métodos de Estimação de DOA e de Estimação de Frequências com Alta Resolução*. 2004. 129 f. Tese (Doutorado) – Universidade Estadual de Campinas, Campinas, 2004.

BALANIS, C. A. *Antenna Theory – Analysis and Design*. Harper and Row, New York, 1982.

BALUJA, S. *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Carniegi Mellon Univ. Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994.*

COOPER, M. Antennas get smart. In: *Scientific American Magazine.* [S. 1.: s. n], 2003.

ESTOICA, P., SHARMAN, K. *Novel Eigenanalysis Method for Direction Estimation*. IEEE Transaction on Acoustics, Speech and Signal Processing, v. 137, p. 19-26, fev. 1990.

FERREIRA, Y. R. *Métodos de Estimação de Ângulos DOA*. 2005. Dissertação (Mestrado) – Programa de Mestrado em Engenharia Elétrica e de Computação, Universidade federal de Goiás, Goiânia, 2005.

GALLAGHER, J. C.; VIGRAHAM, S.; KRAMER, G. *A Family of Compact Genetic Algorithm for Intrinsic Evolvable Hardware*. IEEE Transactions, v. 8, p. 111-126, 2004.

GERSHMAN, A.; STOICA, P. *New MODE-based Techniques for Direction Finding with an Improved Threshold Performances*, Signal Processing, v. 76, p. 221-235, ago. 1999.

GODARA, L. C. *Application of Antenna Arrays to Mobile Communications, Part II: Beamforming and Direction-of-Arrival Considerations*. Proceedings of IEEE, v. 85, p. 1195-1245, n. 8, 1997.

HARIK, G. R; LOBO F. G.; GOLDBERG, D. E. *The Compact Genetic Algorithm*. IEEE Transactions, v. 3, p. 287-297, 1999.

HAYES, M. H. *Statistical Digital Signal Processing and Modeling*. 1ed. John Wiley & Sons New York UEA, 1996.

HAYKIN, S. *Adaptive Filter Theory*. Prentice Hall, Nova Jersey, EUA, 2002.

HE, J.; YAO, X. *From an individual to a population: an analysis of the first hitting time population-based evolutionary algorithms*. IEEE Transactions on Evolutionary Computation, v. 6, n. 5, p. 495-511, 2002.

HIDALGO PÉREZ, J. I.; CERVIGÓN RÜCKAUER, C. *Una Revisión de los Algoritmos y sus Aplicaciones*. Madrid, España: [s. n.], 2002.

ISLAM, M. T.; MISRAN, N. *MI-NLMS Adaptive Beamforming Algorithm for OFDM System*. IEEE Antennas and Propagations Conference LAPC, p. 369-372, 2008.

JAVIER, A. L. *Diseño e Implementación en Hardware de un Algoritmo Bio-Inspirado*. 2009. Tesis (Maestría) – Programa de Maestría en Ciencias de la Computación, Instituto Politécnico Nacional, México, DF, 2009.

LARRAÑAGA, P.; LOZANO, J. A. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA: Kluwer, 2002.

LI, M.; LU, Y. *Genetic Algorithm Based Maximum Likelihood DOA Estimation*. IEEE IET Conference Publications, p. 502-506, 2002.

LIBERTI, J. C., RAPPAPORT, T. S. *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*. Upper Saddle River: Prentice-Hall, 1999.

LOZAO, J. A.; LARRAÑAGA, P.; INZA, I.; BENGOETXEA, E. Towards a New Evolutionary Computation. Springer, Warsaw, 2006.

MAOBING H.; WEI, T.; CANHUI, C. *A New Variable Step-Size LMS Adaptive Filtering Algorithm for Beamforming Technology*. IEEE Antennas Propagation and EM Theory ISAPE International Symposium, v. 9, p. 101-104, 2010.

NOURI, N.; NOORI, N.; BAKHSHI, H. *Joint power allocation and beamforming in relay-assisted cognitive radio networks*. IEEE Asia-Pacific Conference, v. 17, p. 22-27, 2011.

PUTTINI, S. B. *Emprego de Antenas Adaptativas para Estimação de Dados em Ambiente CDMA*. 2006. Dissertação (Mestrado) – Programa de Mestrado em Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 2006.

SCHIDT, R. O. *Multiple Emitter Location and Signal parameter Estimation*. RADC Spectrum Estimation Workshop, 1979, p. 243-258.

SILVA, H. V. L. *Redução da Complexidade Computacional do Método de estimação de Ângulos de Incidência Através da Diferença Entre os Valores Singulares da Matriz de Covariância Espacial*. 2009. Dissertação (Mestrado) – Programa de Mestrado em Engenharia Elétrica e de Computação, Universidade Federal de Goiás, Goiânia, 2009.

SILVA, R. R. *Estudo e Aplicação de um Algoritmo Genético Compacto Usando Elitismo e Mutação*. 2008. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná, Curitiba, 2008.

SKLAR, B. *Digital Communications Fundamentals and Applications*. 2 ed. Prentice Hall, Los Angeles, USA, 2001.

STUTZMAN, W. L.; THIELE, G. A. *Antenna Theory and Design*. 1 ed. John Wiley & Sons, New York USA, 1981.

TORRIERI, D. *Principles of Spread-Spectrum Communication Systems*. 1 ed. Springer, Boston, USA, 2005.

VAN VEEN, B. D.; BUCKLEY, K. M. *Beamforming: A Versatile Approach to Spatial Filtering*. IEEE ASSP Magazine, 1988.

VITALE, M.; VESENTINI, G.; AHMAD N. N.; HANZO, L. *Genetic Algorithm Assisted Adaptive Beamforming*. IEEE Vehicular Technology Conference, v. 56, p. 601-605, vol. 1, 2002.

VITERBI, A. J. *CDMA Principles of Spread Spectrum Communication*. 1 ed. Addison-Wesley Publishing Company, Massachusetts, UEA, 1995.

WOLFGANG, A.; AHMAD, N. N.; CHEN, S.; HANZO, L. *Genetic algorithm assisted error probability optimization for beamforming*. IEEE Electronics Letters, vol. 40, p. 320-322, 2004.

YAN, W.; ZHU, Z. *Maximum Likelihood DOA by Real-Valued Genetic Algorithm*. IEEE National Aerospace Conference, p. 633-636, 2000.

ZIGANGIROV, K. *Theory of Code Division Multiple Access Communication*. 1 ed. IEEE, USA, 2004.