

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**PLATAFORMA COMPUTACIONAL PARA OTIMIZAÇÃO DE
ANTENAS MULTI-LÓBULOS EFICIENTEMENTE INTEGRADAS A
SISTEMAS DE COMUNICAÇÃO SEM FIO**

RAFAEL RABELO NUNES

ORIENTADOR: MARCO ANTONIO BRASIL TERADA

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA

PUBLICAÇÃO: PPGENE.DM – 058/2012

BRASÍLIA / DF: AGOSTO / 2012

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**PLATAFORMA COMPUTACIONAL PARA OTIMIZAÇÃO DE
ANTENAS MULTI-LÓBULOS EFICIENTEMENTE INTEGRADAS A
SISTEMAS DE COMUNICAÇÃO SEM FIO**

RAFAEL RABELO NUNES

**TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA
UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.**

APROVADA POR:

**MARCO ANTONIO BRASIL TERADA, PhD, Virginia Tech – EUA
(ORIENTADOR)**

**PAULO ROBERTO DE LIRA GONDIM, Dr., PUC-RJ
(EXAMINADOR INTERNO)**

**SIDNEY CERQUEIRA BISPO DOS SANTOS, Dr., PUC-RJ
(EXAMINADOR EXTERNO)**

**PAULO HENRIQUE SALES WANDERLEY, Dr., UnB
(EXAMINADOR EXTERNO)**

**ANTONIO JOSÉ MARTINS SOARES, Dr., Unicamp
(EXAMINADOR INTERNO)**

DATA: BRASÍLIA / DF, Agosto de 2012

FICHA CATALOGRÁFICA

NUNES, RAFAEL RABELO

PLATAFORMA COMPUTACIONAL PARA OTIMIZAÇÃO DE ANTENAS MULTI-LÓBULOS EFICIENTEMENTE INTEGRADAS A SISTEMAS DE COMUNICAÇÃO SEM FIO

xiv, 109p., 210 mm x 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2012).

Tese de Doutorado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. Antenas 2. Síntese 3. Assimetria 4. Algoritmos Genéticos 5. MATLAB 6. CML

I. ENE/FT/UnB. II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

NUNES, R. R. (2012). Plataforma Computacional para Otimização de Antenas Multi-lóbulos Eficientemente Integradas a Sistemas de Comunicação Sem Fio. Tese de Doutorado em Engenharia Elétrica, Publicação PPGENE.DM-058/2012, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 109p.

CESSÃO DE DIREITOS

AUTOR: RAFAEL RABELO NUNES

TÍTULO DA TESE: Plataforma Computacional para Otimização de Antenas Multi-lóbulos Eficientemente Integradas a Sistemas de Comunicação Sem Fio.

GRAU/ANO: Doutor/2012.

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese pode ser reproduzida sem a autorização por escrito do autor.

AGRADECIMENTOS

O processo de obtenção de um Título de Doutorado raramente é uma tarefa fácil. Depois de quase seis anos da jornada entre a finalização de minha graduação e obtenção do título de Doutor muito tenho a agradecer. Primeiramente a Deus por ter me dado saúde e permissão para conduzir esse trabalho. Sei que de alguma forma eu tinha que chegar a esse ponto. Ponto em que parece que tudo começa novamente. Sinto que recomeço, olhando para um futuro ainda no mesmo ambiente só que em outra posição. Na frente de uma classe, dessa vez, conduzindo-a.

Agradeço ao Prof. Dr. Marco Terada. É raro encontrar professores como ele. Com um apetite aguçado por pesquisar, conduz sempre o seu trabalho com correção e assertividade, orientando os seus alunos com muito entusiasmo e paciência. Tive a sorte de conhecê-lo ainda em Sistemas de Comunicação 1, cursando uma disciplina optativa da Graduação. Naquela época, o método dele trabalhar já me inspirava quando sempre trazia prática para atividades tão teóricas. Já na pós-graduação tive oportunidade de conhecê-lo melhor quando comecei a trabalhar com ele. Foram congressos, viagens, artigos publicados e muitas vitórias contabilizadas pelo então Grupo de Antenas. Cada artigo aceito era sempre muito comemorado!

À minha esposa Viviane. Se não tivesse iniciado essa trajetória de pós-graduação não teria a sorte de conhecê-la. Foi nas classes de Teoria de Antenas que comecei a desfrutar de sua companhia. Ganhamos até poema! Ela que tolerou meu mau-humor quando as simulações não eram bem sucedidas. Ela que me apoiou, suportando para que eu não desistisse. Ela que me inspirou durante todo esse tempo, me lembrando de quão grandioso seria o dia da vitória. Amo-te. Hoje, ela carrega em seu ventre a Valentina, nossa primeira filha.

Sra. Tânia, minha mãe. A grande inspiradora dos estudos e a eterna busca do saber. Professora, educadora, psicóloga. Sempre preocupada e dizendo que eu me cobrava demais. Ao meu pai, sempre o meu maior ídolo, minha inspiração de retidão e ética. O grande coração da família! A ambos, o orgulho. Três filhos doutores! Aos meus irmãos, Tayse e André, pelo amor e companheirismo. Mesmo longe, sempre presentes no pensamento!

Ao amigo Marco Antonio Caldas de Figueirêdo que me ajudou a revisar esse texto. Com o seu português impecável e a sua paciência, tornou esse texto mais claro mesmo em um momento difícil que passou da sua vida.

À Kerlla pelo processo de revisão e por sempre lembrar, que nada é simples!

Ao amigo Paulo Wanderley, hoje Doutor, por ser companheiro nas horas difíceis em que a tese parecia não evoluir.

À banca por engrandecer esse trabalho.

RESUMO

PLATAFORMA COMPUTACIONAL PARA OTIMIZAÇÃO DE ANTENAS MULTI-LÓBULOS EFICIENTEMENTE INTEGRADAS A SISTEMAS DE COMUNICAÇÃO SEMFIO

Devido ao expressivo crescimento dos sistemas de comunicações sem fio, diversos trabalhos vêm sendo apresentado com o intuito de aproveitar melhor o espectro de frequências. Esse trabalho propõe uma nova forma de se projetar antenas onde se leva em consideração não apenas o seu desempenho elétrico individual, mas também parâmetros do sistema de comunicação frequentemente desconsiderados em sistemas de comunicação no qual a antena está inserida. É mostrado que em ambientes com diversos enlaces, os tipos de modulações utilizados afetam o projeto de antenas. A técnica utilizada no projeto é baseada em otimização por Algoritmos Genéticos e visa obter níveis de lóbulos predeterminados em direções desejadas. A plataforma computacional que permite o projeto das antenas foi desenvolvida parte em linguagem Java e parte em linguagem MATLAB. Sua arquitetura é dividida em três partes: um módulo que realiza a análise de antenas; um módulo que realiza a análise de sistemas de comunicação; e um módulo que realiza integração entre os dois primeiros por uma camada de otimização por Algoritmos Genéticos.

Autor: Rafael Rabelo Nunes

Orientador: Marco Antônio Brasil Terada

Programa de Pós-Graduação em Engenharia Elétrica

Brasília, Agosto de 2012

ABSTRACT

COMPUTACIONAL PLATFORM FOR THE OPTIMIZATION OF ANTENNAS EFFICIENTLY INTEGRATED TO WIRELESS COMMUNICATION SYSTEMS

The explosive growth of wireless communication systems lead to several investigations aiming a more efficient usage of the frequency spectrum. The present work introduces a new way to design antennas, considering not only their individual electrical performance, but also other system parameters not usually considered. Scenarios are discussed and analyzed in detail in which several links and antenna lobes are taken into account, assessing the influence of the digital modulation schemes in the design of antennas. An optimization layer is employed based on the Genetic Algorithms and optimizes the sidelobe levels and directions according to the desired scenario. The computational platform was developed in Java and Matlab and can be divided in three parts: antenna analysis; communication systems; and the integrating optimization layer based on the Genetic Algorithm.

Author: Rafael Rabelo Nunes

Advisor: Marco Antônio Brasil Terada

Programa de Pós-Graduação em Engenharia Elétrica

Brasília, August 2012

ÍNDICE

1.	INTRODUÇÃO	1
1.1.	ORGANIZAÇÃO DO TRABALHO	3
2.	SISTEMAS DE COMUNICAÇÕES	5
2.1.	PRINCÍPIOS BÁSICOS DE RF.....	5
2.2.	GERAÇÃO E TRANSMISSÃO DE ONDAS ELETROMAGNÉTICAS....	7
2.3.	RECEPÇÃO DE ONDAS ELETROMAGNÉTICAS	8
2.4.	ANTENAS.....	9
2.4.1.	Conjuntos de Antenas (<i>Arrays</i>) e Acoplamento Mútuo	12
2.5.	MODULAÇÃO E SINAIS DE BANDA BASE.....	13
2.5.1.	Modulação Analógica.....	14
2.5.2.	Modulação Digital	15
2.5.3.	Comunicação M-Ária.....	18
2.5.4.	Probabilidade de Erro em Modulação Digital.....	21
2.6.	MODELO DE DIVISÃO EM CAMADAS	25
3.	ALGORITMOS GENÉTICOS	28
3.1.	DEFINIÇÕES DOS ALGORITMOS GENÉTICOS	28
3.2.	FUNÇÃO <i>FITNESS</i> (<i>FUNÇÃO DE APTIDÃO</i>).....	29
3.2.1.	Função Objetivo	30

3.2.2.	<i>Fitness</i> Linearmente Proporcional	30
3.2.3.	Limite Sigma	31
3.3.	SELEÇÃO	32
3.3.1.	Método da Roleta	32
3.3.2.	Método do Torneio	33
3.3.3.	Método da Amostragem Universal Estocástica	34
3.4.	OPERADORES GENÉTICOS	34
3.4.1.	Mutação	34
3.4.2.	<i>Crossover</i>	35
4.	PLATAFORMA COMPUTACIONAL.....	38
4.1.	O JAVA E SUA PORTABILIDADE	38
4.1.1.	Classes Java	39
4.1.2.	Empacotamento de arquivos – Pacotes JAR	40
4.1.3.	<i>Applets</i> Java.....	40
4.1.4.	Java <i>Swing</i>	41
4.2.	WEBPRAC.....	41
4.3.	GRADMAX FOR WEB	42
4.4.	MATLAB	43
4.5.	CML.....	44
4.6.	AVALIAÇÃO DO CML.....	45

4.6.1.	Avaliação de simulações do 16-QAM no CML.....	45
4.6.2.	Avaliação de simulações do 64-QAM no CML.....	46
4.7.	A PLATAFORMA COMPUTACIONAL.....	48
4.7.1.	Implementação da Camada de Otimização	49
4.7.2.	Codificação de Possíveis Soluções para Antenas de Fio	50
4.7.3.	Codificação de Possíveis Soluções para Antenas Refletoras	50
4.7.4.	Implementação da Função <i>Fitness</i> pelo Método do Erro Quadrático Médio	51
4.7.5.	Implementação da Função <i>Fitness</i> pelo Método das Máscaras	52
4.7.6.	Método de Seleção	53
4.7.7.	Validação do Algoritmo Genético para Antenas de Fio	54
4.7.8.	Validação do Algoritmo Genético para Antenas Refletoras	57
4.7.9.	Validação do Algoritmo Genético para Antenas de Fio com o Método das Máscaras	61
5.	SIMULAÇÕES	64
5.1.	O CENÁRIO PROPOSTO.....	64
5.2.	PROJETO TRADICIONAL DE UMA ANTENA PARA O CENÁRIO PROPOSTO	67
5.3.	A BUSCA PELA GEOMETRIA DESEJADA COM ALGORITMOS GENÉTICOS.....	69
5.3.1.	Antena com Lóbulos Assimétricos.....	69
5.3.2.	Antena com 6,5 dB de Assimetria	72

5.3.3.	Outra Abordagem – Mudança na Função <i>Fitness</i>	74
5.4.	A UTILIZAÇÃO DE MÁSCARAS NA OTIMIZAÇÃO	76
5.4.1.	O uso de Máscaras e o Método do Erro Quadrático Médio.....	79
5.5.	OTIMIZAÇÃO DE ANTENAS REFLETORAS	81
6.	CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.....	85
	REFERÊNCIAS	87
APÊNDICE A.	MANUAL DE USO DO GRADMAX.....	91
APÊNDICE B.	BIBLIOTECA GRADMAX.JAR	98
APÊNDICE C.	MANUAL DE USO DO WEBPRAC.....	100
APÊNDICE D.	BIBLIOTECA WEBPRAC.JAR	104
APÊNDICE E.	MANUAL DE UTILIZAÇÃO DA PLATAFORMA COMPUTACIONAL	106

ÍNDICE DE TABELAS

Tabela	Página
TABELA 2-1 – CLASSIFICAÇÃO DAS ONDAS ELETROMAGNÉTICAS PELA FREQUÊNCIA.....	6
TABELA 2-2 – PROBABILIDADE DE ERRO DE SÍMBOLO PARA CADA MODULAÇÃO DIGITAL.....	24
TABELA 3-1 – TABELA HIPOTÉTICA COM INDIVÍDUOS E OS SEUS <i>FITNESS</i>	32
TABELA 4-1- COMPARATIVO ENTRE OS RESULTADOS DO CML E DA BIBLIOGRAFIA	47
TABELA 4-2 – LOCALIZAÇÃO DOS FIOS E DA ALIMENTAÇÃO DA CARDÍÓIDE	54
TABELA 4-3 - RESULTADO DO ALGORITMO GENÉTICO DO MELHOR <i>FITNESS</i>	56
TABELA 4-4 - RESULTADO DO ALGORITMO GENÉTICO DO MELHOR <i>FITNESS</i> COM USO DE MÁSCARAS	62
TABELA 5-1 – PARÂMETROS DA ANTENA COM – 20 dB PROJETADA UTILIZANDO O MÉTODO DE <i>DOLPH-CHEBYSHEV</i>	69
TABELA 5-2 - LOCALIZAÇÃO DOS FIOS E DA ALIMENTAÇÃO DA ANTENA ASSIMÉTRICA.....	70
TABELA 5-3 – LOCALIZAÇÃO DOS FIOS E DA ALIMENTAÇÃO DA ANTENA DE REFERÊNCIA PARA O GA	71
TABELA 5-4 – RESULTADO DO ALGORITMO GENÉTICO DO MELHOR <i>FITNESS</i> PARA A ANTENA ASSIMÉTRICA.....	71
TABELA 5-5 – ANTENA COM 6,5 dB, UTILIZANDO COMO BASE UMA LEVE MODIFICAÇÃO NA FASE DO FIO 2.....	73
TABELA 5-6 – RESULTADO DO ALGORITMO GENÉTICO DO MELHOR <i>FITNESS</i> PARA A ANTENA DE 6,5 dB	74
TABELA 5-7 – RESULTADO DO ALGORITMO GENÉTICO DO MELHOR <i>FITNESS</i> PARA A ANTENA DE 6,5 dB PARA OUTRA ABORDAGEM.....	75

TABELA 5-8 – RESULTADO DO ALGORITMO GENÉTICO DO MELHOR <i>FITNESS</i> PARA A ANTENA ASSIMÉTRICA.....	78
TABELA C-1 – DEFINIÇÕES DE CADA UM DOS PARÂMETROS DE UMA ANTENA REFLETORA	101
TABELA E-1 - LISTAGEM DOS ARQUIVOS DA PLATAFORMA COMPUTACIONAL.....	107
TABELA E-2 – LISTAGEM DOS ARQUIVOS DE EXEMPLOS DA PLATAFORMA COMPUTACIONAL.	107
TABELA E-3 - LISTA DOS PARÂMETROS REQUERIDOS PELA PLATAFORMA COMPUTACIONAL..	108

ÍNDICE DE FIGURAS

Figura	Página
FIG. 2.1 – PROPAGAÇÃO DE UMA ONDA ELETROMAGNÉTICA.....	5
FIG. 2.2 – COMPRIMENTO DE ONDA DE UMA ONDA ELETROMAGNÉTICA	6
FIG. 2.3 – DIAGRAMA DE BLOCOS DE UM TRANSMISSOR.....	7
FIG. 2.4 – DIAGRAMA DE BLOCOS DE UM RECEPTOR.....	9
FIG. 2.5 – GANHO E DIAGRAMA DE RADIAÇÃO DE UM DIPOLO COM 0,5M EM 300 MHz. GRÁFICO OBTIDO COM O GRADMAX FOR WEB.....	10
FIG. 2.6 - DIAGRAMA DE RADIAÇÃO DE UMA ANTENA REFLETORA ONDE É POSSÍVEL VERIFICAR O LÓBULO PRINCIPAL E OS LÓBULOS SECUNDÁRIOS	11
FIG. 2.7 – TIPOS DE MODULAÇÃO DIGITAL. (A) ASK – (B) FSK – (C) PSK	15
FIG. 2.8 - CONSTELAÇÃO DO PSK COM $D_p = 90^\circ$	17
FIG. 2.9 - CONSTELAÇÃO PSK COM $D_p = 0^\circ$	18
FIG. 2.10 – CONSTELAÇÃO DO 4-PSK OU QPSK.....	19
FIG. 2.11 – CONSTELAÇÃO DO $\pi/4$ QPSK	19
FIG. 2.12 – CONSTELAÇÃO DO M-ASK	20
FIG. 2.13 – CONSTELAÇÃO DO 16-QAM	21
FIG. 2.14 – FUNÇÃO DENSIDADE DE PROBABILIDADE PARA O RUÍDO GAUSSIANO	22
FIG. 2.15 – GRÁFICO DA FUNÇÃO $Q(y)$	23
FIG. 2.16 – PROBABILIDADE DE ERRO DE SÍMBOLO PARA O M-QAM [24].....	25
FIG. 2.17 – MODELO OSI	26

FIG. 3.1 – ILUSTRAÇÃO QUE MOSTRA COMO OS INDIVÍDUOS SÃO SELECIONADOS PELO MÉTODO DA ROLETA	33
FIG. 3.2 – EXEMPLIFICAÇÃO DO MÉTODO DO TORNEIO.....	33
FIG. 3.3 – EXEMPLO DE MUTAÇÃO REALIZADA EM PARÂMETROS BINÁRIOS	34
FIG. 3.4 – EXEMPLO DE MUTAÇÃO REALIZADA EM NÚMEROS REAIS	35
FIG. 3.5 – OPERADOR <i>CROSSOVER</i> SENDO APLICADO EM UM PONTO FIXO EM DOIS PAIS (A) E (B)	36
FIG. 3.6 – FILHOS GERADOS PELOS PAIS APÓS APLICADO O OPERADOR <i>CROSSOVER</i>	36
FIG. 3.7 – <i>CROSSOVER</i> MULTIPONTO	37
FIG. 3.8 – <i>CROSSOVER</i> REALIZADO EM CROMOSSOMOS COM GENES REAIS	37
FIG. 4.1 – EXEMPLO DE CÓDIGO DE UMA CLASSE JAVA	39
FIG. 4.2 – EXEMPLO DE CÓDIGO DE UMA CLASSE JAVA USANDO A INSTÂNCIA <i>PACKAGE</i>	40
FIG. 4.3 - DADOS DE ENTRADA PARA SIMULAÇÃO DO CML – 16-QAM	46
FIG. 4.4 - DADOS DE ENTRADA PARA SIMULAÇÃO DO CML – 64-QAM	47
FIG. 4.5 – TAXA DE ERROS DE SÍMBOLOS DO 16-QAM E 64-QAM CALCULADOS PELO CML....	48
FIG. 4.6 – EXEMPLIFICAÇÃO DO MÉTODO DAS MÁSCARAS.....	52
FIG. 4.7 – CÓDIGO DE SELEÇÃO PELO MÉTODO DA ROLETA	53
FIG. 4.8 – CÓDIGO DE SELEÇÃO DO PAI.....	54
FIG. 4.9 – DIAGRAMA DE RADIAÇÃO CALCULADO PELO GRADMAX DE UMA CARDÍOIDE.....	55
FIG. 4.10 – GRÁFICOS DA CARDÍOIDE UTILIZADA COMO REFERÊNCIA E A CALCULADA PELO ALGORITMO.....	56
FIG. 4.11 – DIAGRAMA DE RADIAÇÃO CALCULADO PELO GRADMAX DA ANTENA ENCONTRADA	

PELO ALGORITMO	57
FIG. 4.12 – EVOLUÇÃO DO <i>FITNESS</i> NA OTIMIZAÇÃO DA CARDIÓIDE.....	58
FIG. 4.13 – DIAGRAMA DE RADIAÇÃO DE UMA REFLETORA COM VARIAÇÃO DA POSIÇÃO DO ALIMENTADOR.....	58
FIG. 4.14 – DIAGRAMA DE RADIAÇÃO DA ANTENA REFLETORA BASE E A ANTENA CALCULADA	60
FIG. 4.15 – EVOLUÇÃO DO <i>FITNESS</i> NA OTIMIZAÇÃO DA REFLETORA	60
FIG. 4.16 – MÁSCARA UTILIZADA NA OTIMIZAÇÃO DA CARDIÓIDE	61
FIG. 4.17 – MÁSCARA UTILIZADA OTIMIZAÇÃO E A ANTENA CARDIÓIDE NA MESMA IMAGEM...	63
FIG. 4.18 – EVOLUÇÃO DO <i>FITNESS</i> NA OTIMIZAÇÃO.....	63
FIG. 5.1 – ILUSTRAÇÃO COM OS ENLACES PROPOSTOS	65
FIG. 5.2 - 16-QAM RECEBENDO INTERFERÊNCIA DE UMA FONTE QPSK E COM RUÍDO BRANCO ADITIVO GAUSSIANO.....	66
FIG. 5.3 – 64-QAM RECEBENDO INTERFERÊNCIA DE UM LÓBULO DE UMA TRANSMISSÃO QPSK E COM RUÍDO BRANCO ADITIVO GAUSSIANO.....	66
FIG. 5.4 – DIAGRAMA DE RADIAÇÃO DA ANTENA COM – 20 dB NO PLANO XZ PROJETADA UTILIZANDO O MÉTODO DE <i>DOLPH-CHEBYSHEV</i>	68
FIG. 5.5 – DIAGRAMA DE RADIAÇÃO CALCULADO PELO GRADMAX DA ANTENA ASSIMÉTRICA	70
FIG. 5.6 –DIAGRAMA DE RADIAÇÃO, PLANO XZ, LIMITADAS EM -30dB DOS DIAGRAMAS CALCULADOS PELO ALGORITMO	71
FIG. 5.7 – EVOLUÇÃO DO <i>FITNESS</i> PARA A SIMULAÇÃO.....	72
FIG. 5.8 - ANTENA COM 6,5 dB, UTILIZANDO COMO BASE UMA LEVE MODIFICAÇÃO NA FASE DO	

FIO 2	73
FIG. 5.9 – DIAGRAMA DE RADIAÇÃO CALCULADO PELO GRADMAX DA ANTENA COM ASSIMETRIA DE 6,5 DB	74
FIG. 5.10 – DIAGRAMA DE RADIAÇÃO CALCULADO PELO GRADMAX DA ANTENA COM ASSIMETRIA DE 6,5 DB COM <i>FITNESS</i> BASEADO NA DIFERENÇA ENTRE 0 E 180 GRAUS	76
FIG. 5.11 – MÁSCARA INSERIDA PARA O PROJETO DA ANTENA ASSIMÉTRICA PELO ALGORITMO	77
FIG. 5.12 –DIAGRAMA DE RADIAÇÃO, PLANO XZ, TRUNC -30DB DOS DIAGRAMAS CALCULADOS PELO ALGORITMO	78
FIG. 5.13 - EVOLUÇÃO DO <i>FITNESS</i> PARA A SIMULAÇÃO.....	78
FIG. 5.14 – RESULTADO OBTIDO PELA UTILIZAÇÃO CONCOMITANTE DOS DOIS MÉTODOS.....	80
FIG. 5.15 – EVOLUÇÃO DO <i>FITNESS</i> MÉDIO DE CADA GERAÇÃO PARA ESSA SIMULAÇÃO	81
FIG. 5.16 – ANTENA PADRÃO DO WEBPRAC COM ALIMENTADORES EM $X_F=0$; $Y_F=0$ E $Z_F=0$..	82
FIG. 5.17 – MÁSCARA PARA UMA REFLETORA COM ASSIMETRIA DE 4,5 DB.....	83
FIG. 5.18 – RESULTADO OBTIDO APÓS A OTIMIZAÇÃO	84
FIG. 5.19 – EVOLUÇÃO DO <i>FITNESS</i> MÉDIO PARA CADA GERAÇÃO NA OTIMIZAÇÃO	84
FIG. A.1 – TELA INICIAL DO GRADMAX	91
FIG. A.2 – ABA DE INCLUSÃO DA GEOMETRIA DA ANTENA NO GRADMAX	92
FIG. A.3 – DESENHO DA GEOMETRIA DE UM FIO PARA EXEMPLIFICAÇÃO DE USO NO GRADMAX	94
FIG. A.4 – ABA DE INCLUSÃO DE DADOS DE ALIMENTAÇÃO E CARGAS	95
FIG. A.5 – JANELA COM INFORMAÇÕES DOS SEGMENTOS DO FIO	96
FIG. A.6 – JANELA COM INFORMAÇÕES SOBRE TENSÃO, CORRENTE E IMPEDÂNCIA EM CADA	

PULSO BEM COMO O GANHO DA ANTENA CALCULADA	96
FIG. A.7 – DIAGRAMA DE RADIAÇÃO DA CARDÍOIDE, GERADA PELO GRADMAX	97
FIG. A.8 – JANELA DE EXPORTAÇÃO DA <i>STRING</i> COM OS PARÂMETROS DA GEOMETRIA INSERIDA NO GRADMAX.....	97
FIG. B.1 – EXEMPLO DE CÓDIGO JAVA UTILIZANDO A BIBLIOTECA GRADMAX.JAR.....	98
FIG. B.2 - EXEMPLO DE CÓDIGO MATLAB UTILIZANDO A BIBLIOTECA GRADMAX.JAR.....	99
FIG. C.1 – TELA INICIAL DO WEBPRAC.....	100
FIG. C.2 – GEOMETRIA DE UMA ANTENA REFLETORA COM A MARCAÇÃO DOS PARÂMETROS...	101
FIG. D.1 – CÓDIGO JAVA EXEMPLO PARA O USO DA BIBLIOTECA WEBPRAC.JAR	104
FIG. D.2 - CÓDIGO MATLAB EXEMPLO PARA O USO DA BIBLIOTECA WEBPRAC.JAR	105

ACRÔNIMOS

AM – Amplitude Modulation

ASK – Amplitude Shift Keying

AWGN – Additive White Gaussian Noise

CDMA – Code Division Multiple Access

CML – Coded Modulation Library

EIRP – Equivalent Isotropic Radiated Power

FDMA – Frequency Division Multiple Access

FEC – Forward Error Code

FM – Frequency Modulation

FSK – Frequency Shift Keying

GUI – Graphic User Interface

JVM – Java Virtual Machine

LTE – Long Term Evolution

MIMO – Multiple Input Multiple Output

MP3 – MPEG Layer 3

GRADMAX – Software para análise de antenas de fio

QAM – Quadrature Amplitude Modulation

OFDM – Orthogonal Frequency Division Multiplexing

OFDMA – Orthogonal Frequency Division Multiple Access

OOK – On-off Keying

QPSK – Quadrature Phase Shift Keying

OSI – Open Systems Interconnection

PCM – Pulse Code Modulation

PM – Phase Modulation

PSK – Phase Shift Modulation

SDMA – Space Division Multiple Access

TCP – Transport Control Protocol

TDMA – Time Division Multiple Access

WAV – Wave File

WebPRAC – Parabolic Reflector Analysis Code for Web

WiMAX – Worldwide Interoperability for Microwave Access

1. INTRODUÇÃO

O expressivo desenvolvimento da indústria de comunicação sem fio tem demandado cada vez mais um melhor aproveitamento de frequências e um aumento da velocidade de transmissão. As últimas grandes tecnologias em ascensão como o WiMAX e o LTE demonstram essa tendência com a utilização da técnica OFDM que permite atingir taxas de 5 bits/Hz de aproveitamento espectral [1]. Nesse sentido, o desenvolvimento de novas configurações físicas alinhadas com as diversas camadas dos sistemas de comunicação permite idealizar um aumento ainda maior dessas taxas de aproveitamento comportando maiores velocidades de transmissão.

Existem diversos trabalhos que mostram como informações intercamadas podem ser trocadas de forma que o sistema de comunicação trabalhe de forma mais adaptada [2-5]. Esta tese introduz uma plataforma computacional desenvolvida nas linguagens Java e MATLAB que permite o projeto de antenas integradas a sistemas de comunicações sem fio em cenários de modulações diferentes. Apesar das antenas e dos esquemas de modulação residirem no nível físico da camada OSI, a plataforma computacional pode ser estendida para a análise e projeto de antenas em cenários que envolvam outros níveis do modelo OSI.

Os aspectos de originalidade desse trabalho se baseiam em uma maneira nova de se projetar antenas. Tradicionalmente, as antenas são projetadas visando um conjunto de especificações para o seu desempenho elétrica individual, tais como um ganho desejado em determinada área de cobertura e mínima interferência fora dela [6-8].

No caso deste trabalho, o método de projeto consiste em na otimização da antena para que a mesma trabalhe eficientemente em um sistema de comunicações sem fio, levando-se em conta parâmetros sistêmicos que normalmente não são considerados, tais como as modulações digitais em uma tecnologia LTE, como exemplo [6][9][10]. É mostrado que em ambientes com diversos enlaces diferentes, uma situação que ilustra sistemas reais de comunicações, os tipos de modulações dos enlaces afetam o projeto da antena, nesse caso um conjunto de antenas de fios com lóbulos múltiplos, resultando em

uma configuração de antena diferente da projetada sem se levar em conta esses parâmetros sistêmicos.

A técnica é baseada em otimização por algoritmos genéticos, e visa obter níveis de lóbulos predeterminados em direções desejadas, diferentemente das técnicas de projeto convencionais que não levam em conta parâmetros sistêmicos, como modulações [6][11], ou mesmo das que podem considerar esses parâmetros envolvendo antenas inteligentes, pois até onde se sabe são especificadas apenas as direções desejadas de máximos e mínimos, mas não necessariamente os níveis dos lóbulos secundários (formato do diagrama de radiação) [12-15]. Outra vantagem entre a técnica proposta e a comumente encontrada na literatura de antenas inteligentes e MIMO é que neste trabalho são modeladas antenas com a máxima precisão computacional através do Método dos Momentos [7][16], levando-se em conta detalhes específicos da geometria e configuração do conjunto, incluindo efeitos de acoplamento mútuo (normalmente quando parâmetros sistêmicos, como o tipo de modulação empregado, são considerados no projeto de antenas, além de não se projetar a antena para um diagrama desejado, apenas elementos isotrópicos são empregados) [17].

A importância desta tese está em se projetar antenas melhor integradas ao seu ambiente de operação de forma sistemática e que possuam vantagens sobre as obtidas com os métodos tradicionais, incluindo os utilizados com antenas inteligentes, em termos de um melhor ganho, facilidade de construção, rapidez de adaptação a mudanças do ambiente, entre outras. A metodologia aqui desenvolvida é geral e independente da tecnologia empregada, podendo ser facilmente estendida a outros tipos de antenas além das de fios e refletoras apresentadas, bem como considerar outros parâmetros sistêmicos além das modulações digitais ou do padrão LTE, incluindo os pertencentes a outras camadas do modelo OSI. Dessa forma, introduz-se neste trabalho um processo de síntese geral que resulta em antenas melhores adaptadas ao ambiente de operação e mais fáceis de operar e construir quando comparadas com as obtidas por outras técnicas de projeto.

Todo esse trabalho é realizado por meio de uma plataforma computacional, desenvolvida em Java e MATLAB. A plataforma aqui introduzida é dividida em três

partes: um módulo de análise de antenas; um módulo de análise de sistemas de comunicação; e uma camada de otimização.

O módulo de análise de antenas é uma evolução dos *softwares* PRAC e GRADMAX, muito utilizados em [18][19]. O autor desta tese fez o trabalho de conversão desses *softwares* para a linguagem Java, cuja modernidade trouxe flexibilidade ao código produzido para a produção de uma interface amigável utilizável diretamente de uma página Web [20][21]. Além disso, foi produzida uma biblioteca que utiliza o código desses *softwares* para que a análise das antenas pudesse ser realizada em outros ambientes, como o MATLAB.

O módulo de análise de sistemas de comunicação utilizado foi o *Coded Modulation Library* (CML), utilizado na análise de sistemas de comunicações, modulações, códigos de erros e demais parâmetros sistêmicos [22]. O CML foi codificado em MATLAB e por esse motivo, os resultados da análise dos sistemas de comunicações produzidas por esse *software* puderam ser integrados com o módulo de análise de antenas. Essa integração foi realizada por meio de uma camada de otimização, codificada em MATLAB, com o uso de Algoritmos Genéticos (GA, do inglês *Genetic Algorithm*).

Por fim, cabe ressaltar a importância acadêmica desta tese. Os *softwares* de análise de antenas desenvolvidos durante a pesquisa, bem como a plataforma computacional vêm sendo utilizados como apoio no ensino de disciplinas de antenas no Departamento de Engenharia Elétrica da Universidade de Brasília além de terem sido citados como referência em livro texto da área [7].

1.1. ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado da seguinte forma. No Capítulo 2, é feita uma introdução teórica dos temas técnicos úteis para compreensão da tese incluindo informações sobre como funcionam antenas, esquemas de modulação e os sistemas de comunicação em geral. Em seguida, no Capítulo 3 é feita uma introdução teórica sobre Algoritmos Genéticos, utilizados pela camada de otimização da plataforma

computacional dessa tese. Caso o leitor já conheça os temas apresentados nesses dois capítulos, pode-se iniciar a leitura pelo capítulo seguinte.

O Capítulo 4 contém detalhes técnicos de toda a plataforma computacional. Ele inicia introduzindo ao leitor a linguagem de programação Java e todas as tecnologias dessa linguagem utilizadas nos softwares de análise e projeto de antenas refletoras (*WebPRAC*), e nos softwares de análise e projeto de antenas de fio (*GRADMAX*). Ambos os *softwares* também são tratados nesse capítulo.

Ainda no Capítulo 4, o módulo de análise de sistemas de comunicações, representado pelo CML, é introduzido, testado e comparado com resultados de bibliografias. Com isso, conseguiu-se validar que esse *software* realiza cálculos com eficiência. Por fim, ainda no Capítulo 4, a camada de otimização é apresentada e validada para o projeto de antenas conhecidas. Para isso, é realizado o projeto de uma antena de fio e de uma antena refletora conhecidas.

No Capítulo 5 são realizadas simulações referente à síntese de antenas eficientemente integradas a sistemas de comunicação. Um cenário é proposto em que há dois sistemas de comunicações diferentes, um transmitindo com modulação 16-QAM e outro transmitindo com modulação 64-QAM, em que ambos são interferidos por um outro sinal QPSK. A partir desse cenário, foi projetada uma antena eficientemente integrada a esse sistema de comunicação. Seguindo no Capítulo 5, é mostrado um cenário onde uma antena refletora também poderia ser projetada em um cenário fictício, mostrando que a plataforma computacional pode facilmente adequar-se a qualquer modelo de antenas, bastando ter um código de análise específico para o tipo de antena.

2. SISTEMAS DE COMUNICAÇÕES

Nesse capítulo serão apresentados os principais componentes de um sistema de comunicação. Assuntos como RF (Radiofrequência), Antenas, Modulação, Multiplexação e o Modelo de Comunicação em Camadas serão abordados. Cada um desses componentes será apresentado de forma simplificada apenas com o intuito de contextualizar o trabalho apresentado.

2.1. PRINCÍPIOS BÁSICOS DE RF

Uma onda eletromagnética é combinação de uma perturbação dos campos elétrico e magnético que transmitem energia. A Fig. 2.1 mostra uma onda eletromagnética com os campos elétrico e magnético propagando em uma direção qualquer. Tanto o campo elétrico quanto o magnético podem ser representados por uma senóide, com amplitude, frequência e fase. É a variação de um ou mais desses parâmetros de uma onda eletromagnética que permite o envio de informação entre pontos distantes.

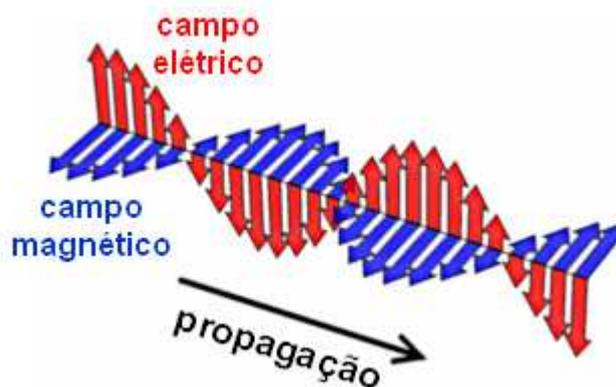


Fig. 2.1 – Propagação de uma onda eletromagnética

A frequência de oscilação “ f ”, em Hertz, da onda pode ser calculada por meio da equação (2-1), onde “ c ” é a velocidade da luz em metros por segundo e “ λ ” é o comprimento de onda em metros (distância entre dois picos ou dois vales). A Fig. 2.2 mostra alguns ciclos de uma onda onde se pode verificar a determinação do comprimento de onda.

$$f = \frac{c}{\lambda} \quad (2-1)$$

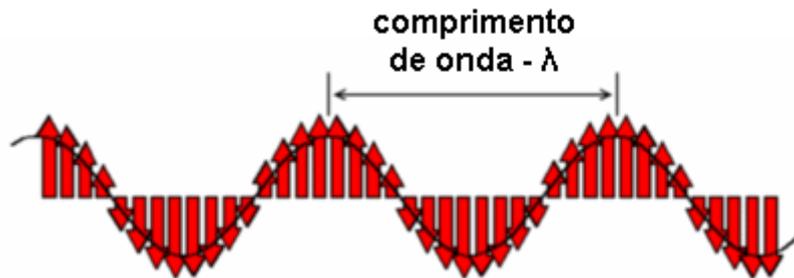


Fig. 2.2 – Comprimento de onda de uma onda eletromagnética

Da mesma forma que a luz tem a sua classificação em cores, as ondas eletromagnéticas são classificadas de acordo com a sua frequência e variam desde a VLF (*Very Low Frequency*) até a EHF (*Extremely High Frequency*). A Tabela 2-1 mostra essa classificação entre as frequências de 9 kHz até 300 GHz. Acima dessas frequências, a onda começa a tomar forma de luz infravermelha, luz visível, luz ultravioleta, raios X e raios gama.

Tabela 2-1 – Classificação das ondas eletromagnéticas pela frequência.

Nome	Abreviação	Frequências	Comprimento de Onda
Very Low Frequency	VLF	9 kHz – 30 kHz	33 km – 10 km
Low Frequency	LF	30 kHz – 300 kHz	10 km – 1 km
Médium Frequency	MF	300 kHz – 3 MHz	1 km – 100 m
High Frequency	HF	3 MHz – 30 MHz	100 m – 10 m
Very High Frequency	VHF	30 MHz – 300 MHz	10 m – 1 m
Ultra High Frequency	UHF	300 MHz – 3 GHz	1 m – 100 mm
Super High Frequency	SHF	3 GHz – 30 GHz	100 mm – 10 mm
Extremely High Frequency	EHF	30 GHz – 300 GHz	10 mm – 1 mm

Essas bandas de frequências são divididas em canais individuais que nada mais são do que pequenas fatias do espectro designados ao transmissor e ao receptor. Esses canais variam de acordo com a quantidade e o tamanho do espectro. Essas características são ditadas pelo tipo da banda e o tipo de sistemas de comunicação que serão oferecidos.

2.2. GERAÇÃO E TRANSMISSÃO DE ONDAS ELETROMAGNÉTICAS

A geração das ondas eletromagnéticas é uma responsabilidade do transmissor. Ele possui diversos subsistemas que podem ser representados de uma forma simples pelo diagrama de blocos da Fig. 2.3.

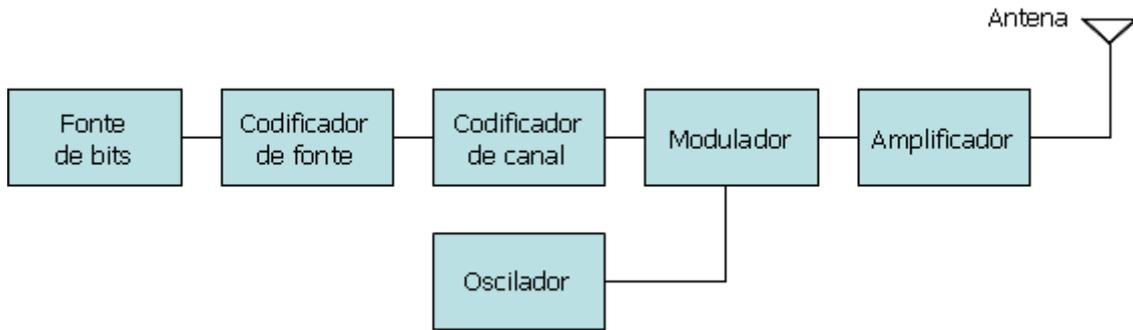


Fig. 2.3 – Diagrama de blocos de um transmissor

Cada um desses blocos tem uma função bem específica. O bloco “fonte de bits” representa a informação que se deseja transmitir que pode ser um e-mail, uma imagem, uma música, ou seja, qualquer tipo de informação.

O codificador de fonte é responsável por reduzir a redundância da fonte de bits. Na maioria das vezes, uma informação gerada tem alguma redundância que pode ser reduzida. Um dos maiores exemplos desse fato é um texto escrito. Como exemplo, pode-se exemplificar o arquivo desta tese. Se ele for submetido a um algoritmo de compactação de dados como o ZIP ou o RAR, o número de bits após esse processo será bem menor. Existem diversos codificadores de fonte e eles atingem a sua maior eficiência quando são desenvolvidos especificamente para um tipo de informação. O algoritmo ZIP, por exemplo, que pode ser muito eficiente ao reduzir as redundâncias desse texto, não é eficiente para reduzir as redundâncias existentes em um arquivo de música .WAV. Neste caso, realizar a codificação pelo algoritmo MP3 é mais eficiente.

Seguindo os blocos, tem-se um codificador de canal. Os codificadores de canal são responsáveis por adicionar redundância de forma a detectar e corrigir erros oriundos

de interferências, multi-percursos e ruído. Pode parecer um pouco incoerente colocar redundância em um estágio após uma redundância ter sido removida pelo codificador de fonte, entretanto, a redundância retirada pelo codificador de fonte não permite corrigir erros gerados por distorções no canal. Os codificadores de canal mais conhecidos são o FEC (*Forward Error Correction*) e o *Bit-interleaving*.

Após ter passado pelo codificador de canal, os bits estão prontos para serem transmitidos. O modulador tem o papel de combinar uma onda em uma frequência específica gerada pelo oscilador com os bits recebidos do codificador de canal. Existem basicamente dois tipos de modulação: a analógica e a digital, mas para esse trabalho, apenas a modulação digital é de interesse. Dentre as técnicas de modulação digital, pode-se citar o QPSK, M-QAM, ASK, PSK e FSK.

Depois de modulado, o sinal é amplificado e entregue à antena que tem o papel de irradiar e transmiti-lo pelo meio eletromagnético.

2.3. RECEPÇÃO DE ONDAS ELETROMAGNÉTICAS

A Fig. 2.4 mostra um diagrama de blocos simplificado de um receptor. A onda eletromagnética transmitida é transformada em um sinal AC pela antena e passa por um amplificador de RF. Ao chegar à antena, o sinal transmitido foi atenuado e a antena recebeu sinal de um universo infinito de outras fontes de RF, inclusive do mesmo sinal transmitido, que percorreu outro caminho e que acabou chegando em uma fase distinta.

O amplificador de RF é utilizado para trabalhar com potências extremamente baixas. Tem como objetivo aumentar a potência do sinal recebido, após ele ter sido filtrado. Esse estágio torna-se necessário, pois, a potência do sinal recebida é extremamente baixa para que possa ser detectada pelos demais estágios.

O sinal é combinado em um *mixer* com o oscilador local para selecionar a frequência desejada. O sinal é filtrado para retirar as frequências não desejadas produzidas pela combinação do sinal no *mixer* e depois o detector traduz o sinal recebido.

Esse diagrama de blocos foi bastante simplificado. Dependendo do sistema de comunicação e do fabricante do equipamento do receptor, diversos outros estágios são introduzidos de forma a obter eficiência ainda maior na recepção.

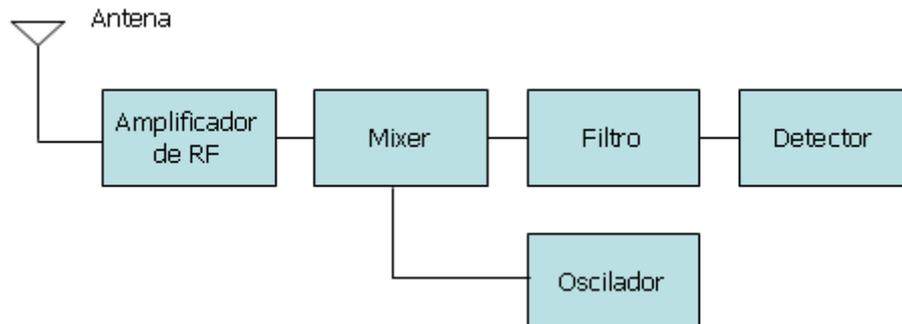


Fig. 2.4 – Diagrama de blocos de um receptor

2.4. ANTENAS

Antenas são dispositivos utilizados para transformar a energia eletromagnética existente em uma linha de transmissão em energia eletromagnética irradiada e vice-versa. Sendo assim, qualquer sistema de comunicação sem fios requer uma antena para que a onda eletromagnética seja transmitida para pontos diversos. Existem antenas de diversos tipos, formas e tamanhos. Cada antena possui diversos parâmetros que serão úteis para propósitos específicos. Pode-se dizer, então, que a escolha da antena é uma das mais importantes escolhas ao se projetar um sistema de comunicação sem fio.

Para começar a entender uma antena, leve-se em consideração uma antena chamada isotrópica, ou seja, uma que irradia energia eletromagnética uniformemente em todas as direções. Na verdade, esse tipo de antena só existe na teoria. Na natureza, é muito difícil irradiar energia eletromagnética uniformemente em todas as direções. Por analogia, uma antena isotrópica poderia ser o sol.

Assumindo toda a energia emitida por essa antena isotrópica, pode-se levar em consideração que a densidade de potência em W/m^2 no limite dessa esfera será a potência total irradiada dividida pela área dessa esfera.

A antena do mundo real mais simples é o dipolo. O dipolo nada mais é do que duas partes de iguais tamanhos de um fio que tem o comprimento ressonante na frequência desejada. A Fig. 2.5 mostra como o dipolo irradia. Esse padrão, parecido como o formato do símbolo de infinito “∞”, irradia mais em uma direção e em dois sentidos. Sendo assim, essa antena concentra energia eletromagnética mais em alguns pontos do que em outros. Por esse motivo, essa antena tem um ganho em relação a antena isotrópica que nesse caso é de 2,16 dBi; ou seja, ela irradia em seu máximo 2,16 dB em relação a antena isotrópica.

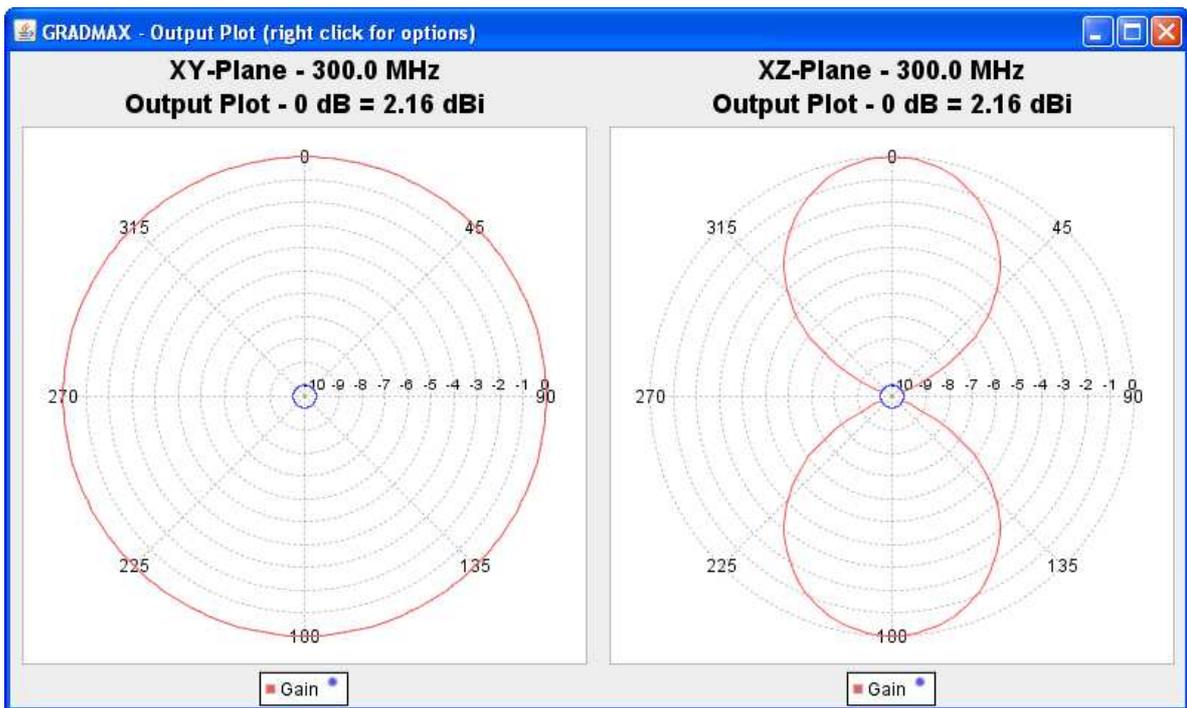


Fig. 2.5 – Ganho e diagrama de radiação de um dipolo com 0,5m em 300 MHz. Gráfico obtido com o GRADMAX for Web.

Em geral, o ganho de uma antena é comparado com a irradiação da antena isotrópica. No caso do dipolo, o ganho então é de 2,16 dBi. Entretanto, alguns fabricantes de antenas comparam o padrão de irradiação com o dipolo e expressam o ganho em dBd. No caso do dipolo, o ganho dele é então de 0 dBd. É muito importante ter atenção em qual unidade o ganho da antena está sendo medido, pois essa diferença de aproximadamente 2,1 dB influi bastante no cálculo de um enlace.

Os dipolos e as antenas isotrópicas são antenas omnidirecionais, pois irradiam energia em um campo de 360 graus em torno delas. Um outro tipo de antenas são as

direcionais que possuem o padrão de radiação como um cone. Como analogia, pode-se levar em conta que ela irradiaria como se fosse uma lanterna. Como essas antenas estão concentrando a energia em um espaço ainda menor, pode-se dizer que elas possuem um ganho ainda maior.

Existem diversos tipos de antenas direcionais, cada uma com a sua aplicação. Dentre elas, podem-se citar as antenas de fio [16], antenas de *patch*, antenas refletoras [6][21], cornetas, setoriais, entre outras. Essas antenas direcionais concentram a energia eletromagnética em uma região e essa região é chamada de lóbulo principal.

A Fig. 2.6 mostra o diagrama de radiação na polarização em uso (Copol) e na polarização cruzada (Xpol) de uma antena refletora com 4 m de diâmetro operando na frequência de 11,25 GHz.

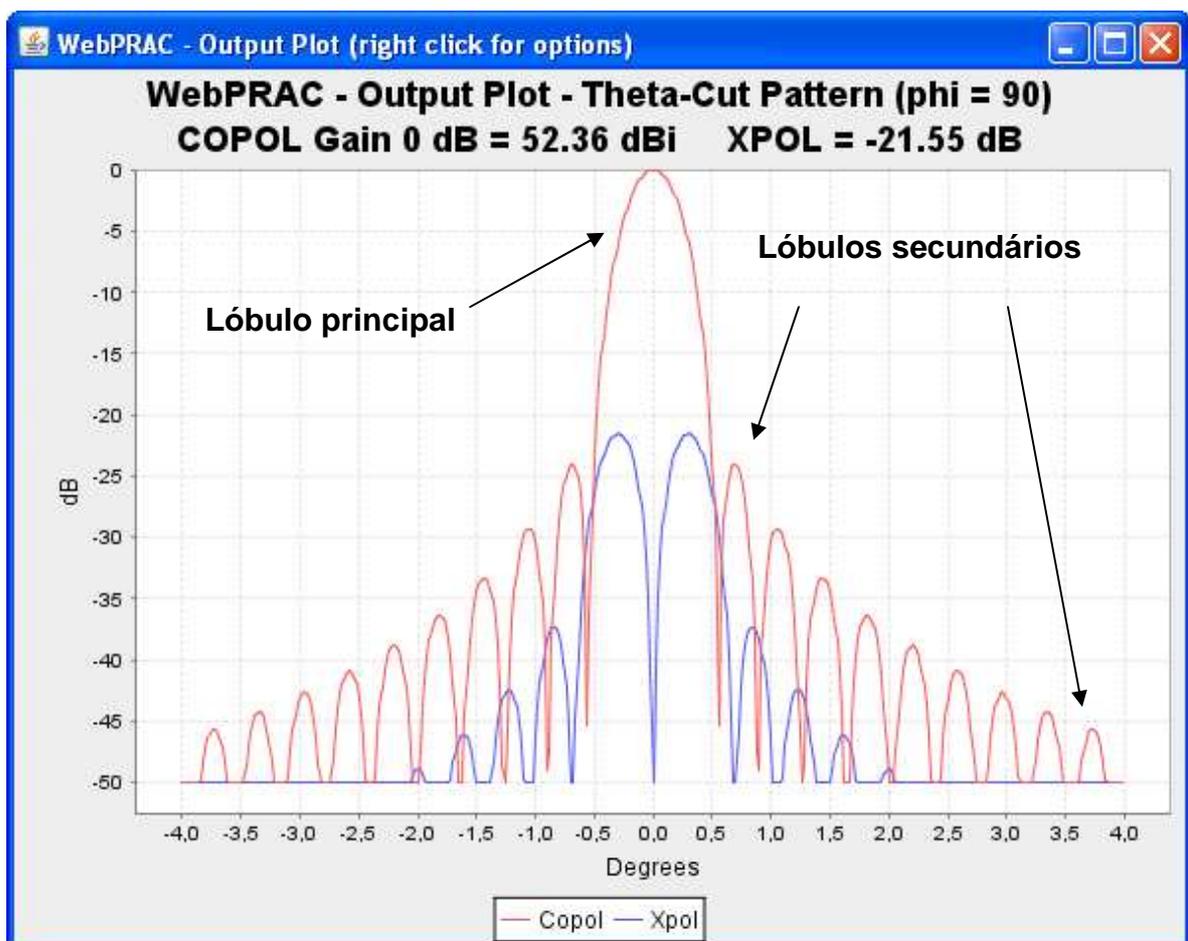


Fig. 2.6 - Diagrama de radiação de uma antena refletora onde é possível verificar o lóbulo principal e os lóbulos secundários

Para essa antena, o lóbulo principal é bem estreito sendo que ele decai 3 dB com apenas 0,25 graus. Isso significa que a energia irradiada reduz-se à metade ao se mover apenas 0,25 graus. Essa concentração da energia eletromagnética em uma direção leva com que o ganho dessa antena na direção do lóbulo principal seja de 52,36 dBi, isto é, ela concentra a energia 52,36 dB em relação ao ganho de uma antena isotrópica.

Os lóbulos secundários também estão em destaque na Fig. 2.6. Apesar dos valores desses lóbulos serem bem menores do que o ganho do lóbulo principal, -21,55 dB no caso do primeiro lóbulo secundário, eles podem interferir nos sistemas de comunicação que operam na mesma frequência de forma significativa. Em geral, os fabricantes de antenas procuram realizar projetos de forma a diminuir esses lóbulos secundários.

2.4.1. Conjuntos de Antenas (*Arrays*) e Acoplamento Mútuo

Um conjunto de antenas é definido como um grupo de antenas operando de forma conjunta na transmissão e recepção com o objetivo de maximizar a intensidade e qualidade do sinal. Para realizar a análise de conjuntos de antenas, substitui-se cada antena do conjunto por uma fonte isotrópica. Assim, o diagrama não-normalizado do conjunto é dado por um somatório ponderado de ondas esféricas, onde os pesos são as amplitudes e fases das alimentações de cada antena (ou elemento) do conjunto (2-2), onde $f(\theta, \phi)$ não está normalizada e $I_n = |I_n| < I_n$ (quantidade complexa na forma polar, em amplitude e fase), N é a quantidade de elementos do conjunto, $\beta = 2\pi / \lambda$, e R é a distância.

$$f(\theta, \phi) = \sum_{i=1}^N I_n \frac{e^{-j\beta R}}{4\pi R} \quad (2-2)$$

Assumindo que todos os elementos do conjunto são iguais, o diagrama final do conjunto pode ser obtido pelo produto do diagrama individual de cada elemento pelo conjunto isotrópico. Após a normalização, tem-se a equação (2-2), onde $g_a(\theta, \phi)$ é o

fator de elemento (diagrama de cada elemento individual) e $f(\theta, \phi)$ é o fator de conjunto AF (isotrópico e normalizado).

$$F(\theta, \phi) = g_a(\theta, \phi) f(\theta, \phi) \quad (2-3)$$

Essa aproximação é normalmente referida como Princípio da Multiplicação de Diagramas e desconsidera:

1. Os efeitos de acoplamento mútuo (interações entre os elementos no conjunto), que pode levar a erros grosseiros caso os elementos estejam muito próximos uns dos outros (impedância de entrada, diagramas e polarizações);
2. Os diagramas dos elementos nas extremidades do conjunto são bem diferentes dos imersos no conjunto;
3. Formatos conformais ou modelados, nos quais os elementos apontam para direções diferentes.

A equação (2-4), leva em conta a diferença de fase devido às distâncias diferentes entre cada elemento e o ponto de observação, onde I_i é a amplitude e fase alimentada em cada elemento, e ξ_i leva em consideração a diferença de fase devido à distância física entre os elementos para o ponto de observação. Desta forma, a diferença de fase total é devida em parte pela alimentação (onde cada elemento é alimentado com relação a uma referência) e em parte devido à geometria ou disposição física dos elementos no espaço.

$$AF(\text{norm}) = I_0 e^{j\xi_0} + I_1 e^{j\xi_1} + \dots + I_N e^{j\xi_N} \quad (2-4)$$

2.5. MODULAÇÃO E SINAIS DE BANDA BASE

Banda base é o termo a que se designa a banda de frequências de um sinal entregue por uma fonte qualquer, como, por exemplo, o sinal entregue após o codificador de canal na Fig. 2.3. Na rede de telefonia, o sinal de voz varia, em banda

base, de 0 até 4 KHz. Já para a imagem, o sinal de vídeo ocupa, em banda base, entre 0 e 4,3 MHz. Para sinais digitais, esse cálculo varia de caso a caso. Para um sinal do tipo PCM (*Pulse Code Modulation*), um sinal com R_b pulsos por segundo ocupa uma banda de 0 até R_b Hz.

Nem sempre é possível trabalhar em banda base. Em comunicações sem fio, é necessário transladar o sinal da sua banda para outra faixa de frequências para que possa ser transmitido. A esse processo se dá o nome de Modulação. Modulação, então, pode ser definida como o processo de combinar um sinal de entrada $m(t)$ e uma portadora de frequência qualquer f_c , para produzir um sinal $s(t)$ que tem sua banda de frequências centralizada na uma frequência f_c . Isso na prática é transmitir sinais em frequências diferentes da banda base. Existem dois tipos de modulação: a analógica, em que o sinal de entrada é analógico, e a modulação digital, em que o sinal de entrada é digital.

2.5.1. Modulação Analógica

A modulação analógica, a mais simples, foi utilizada nas primeiras gerações dos sistemas de comunicações. Existem basicamente dois tipos de modulação analógica: a modulação de amplitude, AM (*Amplitude Modulation*) equação (2-5), e a modulação em fase, que possui duas possibilidades: FM (*Frequency Modulation*) equação (2-6) e PM (*Phase Modulation*) equação (2-7).

Considerando um sinal $m(t)$ qualquer e que $\cos(\omega_c t)$ seja uma portadora, a equação que define cada uma das modulações analógicas pode ser representada por:

$$s(t)_{AM} = m(t) \cos(\omega_c t) \quad (2-5)$$

$$s(t)_{FM} = A \cos \left[\omega_c t + k_f \int_{-\infty}^t m(\alpha) d\alpha \right] \quad (2-6)$$

$$s(t)_{PM} = A \cos[\omega_c t + k_p m(t)] \quad (2-7)$$

2.5.2. Modulação Digital

Da mesma forma que na modulação analógica, a modulação digital utiliza-se dos mesmos parâmetros da portadora para enviar a informação. A Fig. 2.7, mostra os três tipos básicos de modulação de um sistema digital: ASK (*Amplitude Shift Keying*), FSK (*Frequency Shift Keying*) e PSK (*Phase Shift Keying*). Na figura, é possível verificar a forma de cada uma das ondas do sinal modulado de acordo com o sinal em banda base.

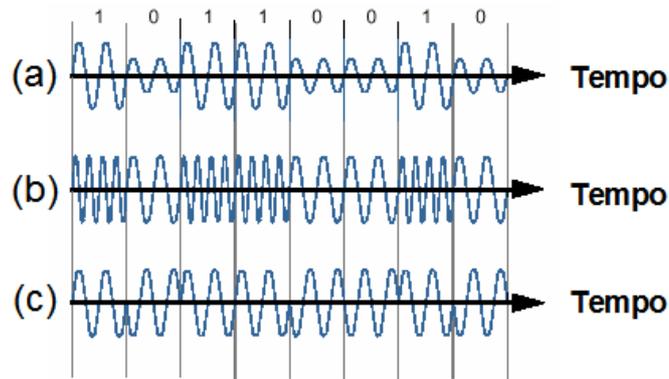


Fig. 2.7 – Tipos de modulação digital. (a) ASK – (b) FSK – (c) PSK

A forma de representação matemática de cada uma dessas modulações será descrita a seguir de forma sucinta. Para o ASK, também chamado de OOK (*On-off key*), o sinal modulado pode ser representado pela equação (2-8) onde A_c é uma constante, $m(t)$ é o sinal e $\cos(\omega_c t)$ é a portadora. Na equação (2-9), B_T é a banda ocupada por esse sinal caso seja utilizado uma filtragem de cosseno levantado onde r é o fator de *roll-off* do filtro e R é a taxa de bits em bps [22].

$$s(t) = A_c m(t) \cos(\omega_c t) \quad (2-8)$$

$$B_T = (1 + r)R \quad (2-9)$$

No caso do FSK, o sinal modulado pode ser descrito pela equação (2-10). E a banda desse sinal B_T , pode ser dada pela regra de Carson, dada pela equação descrita em (2-11), onde ΔF é o desvio de frequência e R é a taxa em bps. Caso tenha sido utilizada filtragem do tipo cosseno levantado, a equação que descreve a banda utilizada pelo sinal FSK é a equação (2-12), onde r é o fator de *roll-off* do filtro [22].

$$s(t) = A_c \cos[\omega_c + \theta(t)] = \begin{cases} A_c \cos(\omega_1 + \theta_1) & \text{bit.1} \\ A_c \cos(\omega_2 + \theta_1) & \text{bit.0} \end{cases} \quad (2-10)$$

$$B_T = 2(\Delta F + R) \quad (2-11)$$

$$B_T = 2\Delta F + (1+r)R \quad (2-12)$$

Já no PSK, o sinal pode ser representado pela equação (2-13), onde A_c é uma constante, ω_c é o inverso da frequência da portadora, D_p é uma constante e $m(t)$ é o sinal. Levando em consideração que $m(t)$ assume os valores -1 e 1, a equação (2-13) também pode ser escrita como (2-14), sabendo que $\cos(x)$ e $\sin(x)$ são funções pares e ímpares de x [22].

$$s(t) = A_c \cos[\omega_c t + D_p m(t)] \quad (2-13)$$

$$s(t) = A_c \cos(D_p m(t)) \cos(\omega_c t) - A_c \sin(D_p m(t)) \sin(\omega_c t) \quad (2-14)$$

A equação (2-14) permite uma visualização mais rápida para diferentes valores de D_p quando comparada com (2-13). Caso $D_p = 90^\circ$, a equação (2-14) pode ser simplificada e escrita por (2-15).

$$s(t) = -A_c m(t) \sin(\omega_c t) \quad (2-15)$$

A representação complexa do sinal PSK permite introduzir o conceito de constelação. Muitas vezes, o sinal digital é representado por meio de uma constelação que nada mais é do que pontos em um plano cartesiano real versus imaginário. Para o sinal PSK com $m(t)$ assumindo os valores -1 e 1, a constelação, descrita anteriormente pela equação (2-15) é representada pelos pontos da Fig. 2.8.

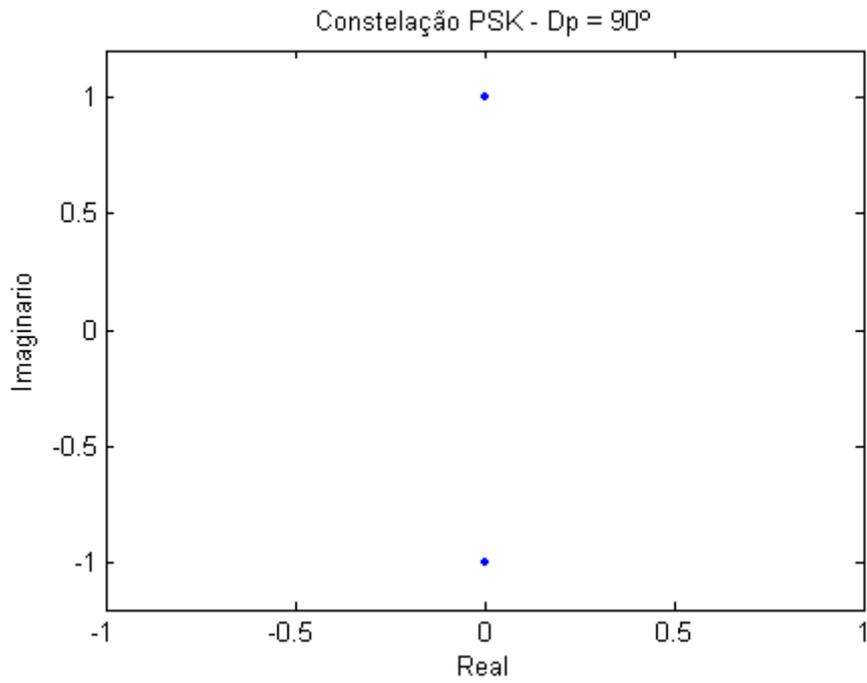


Fig. 2.8 - Constelação do PSK com $Dp = 90^\circ$

Para $Dp = 0$, o sinal PSK assume a forma:

$$S(t) = A_c m(t) \cos w_c t \quad (2-16)$$

Que no plano imaginário se torna:

$$g(t) = A_c e^{j\theta(t)} \quad (2-17)$$

Onde $\theta(t) = 0$ ou $\theta(t) = -\pi$, o que leva à constelação da Fig. 2.9.

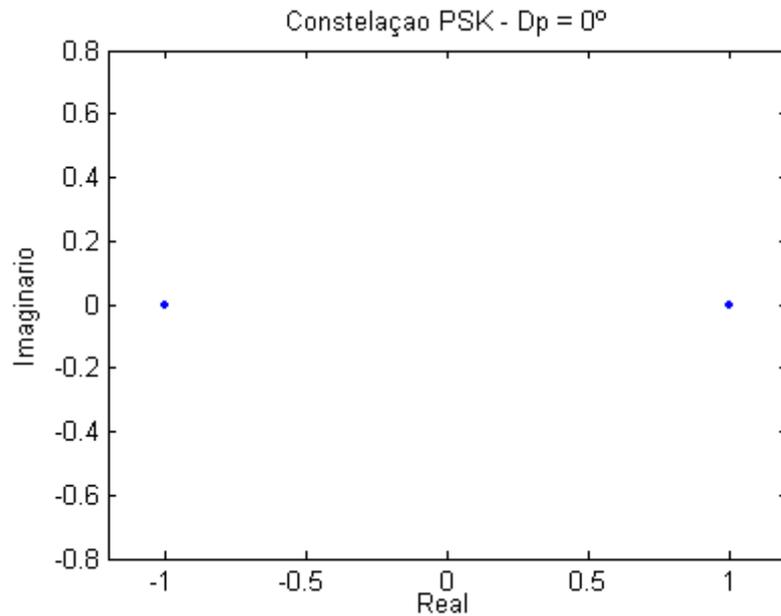


Fig. 2.9 - Constelação PSK com $D_p = 0^\circ$

2.5.3. Comunicação M-Ária

Nas técnicas ASK, FSK e PSK, $m(t)$ assume apenas dois níveis e cada um desses níveis representa os bits 0 e 1. Nas técnicas de modulação digital M-Árias, $m(t)$ assume uma maior quantidade de níveis.

Como exemplo, considere uma fonte que gere quatro níveis de sinais. Define-se a modulação 4-PSK como sendo a representada pela equação

$$g(t) = A_c e^{j\theta(t)} \quad (2-18)$$

em que $\theta(t)$ são os quatro possíveis níveis, ou no caso do PSK, as quatro possíveis fases. A constelação do 4-PSK, que também é chamado de QPSK (*Quadrature Phase Shift Keying*), é descrita pela Fig. 2.10 em que os quatro níveis são as fases de 0° , 90° , 180° e 270° . Também há outra possibilidade em que as quatro fases estão deslocadas para 45° , 135° , 225° e 315° , mostrada pela Fig. 2.11.

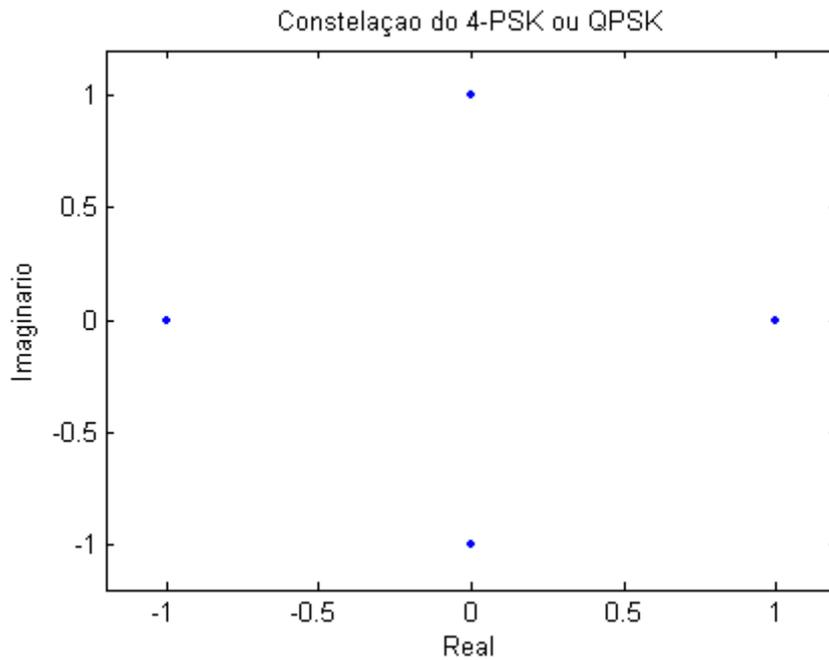


Fig. 2.10 – Constelação do 4-PSK ou QPSK

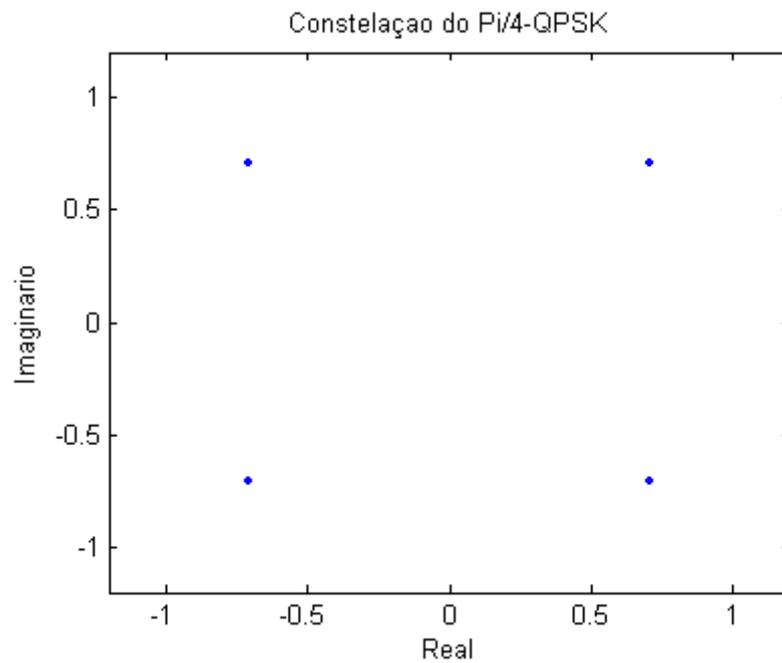


Fig. 2.11 – Constelação do $\pi/4$ QPSK

O M-PSK (*M*-Ary Phase Shift Keying) pode também ser gerado por duas portadoras em quadratura para cada um dos ângulos $\theta_1, \theta_2, \dots, \theta_M$, do sinal MPSK. Nessa fórmula geral, *M* pode assumir 2^n níveis, em que *n* é o número de símbolos.

$$g(t) = A_c e^{j\theta(t)} = x(t) + jy(t) \quad (2-19)$$

$$x_i = A_c \cos\theta_i \quad (2-20)$$

$$y_i = A_c \sin\theta_i \quad (2-21)$$

O M-ASK é a mesma modulação descrita anteriormente pela equação (2-8) do ASK, entretanto nesse caso, $m(t)$ assume mais do que dois níveis. A Fig. 2.12 mostra a constelação do 4-ASK.

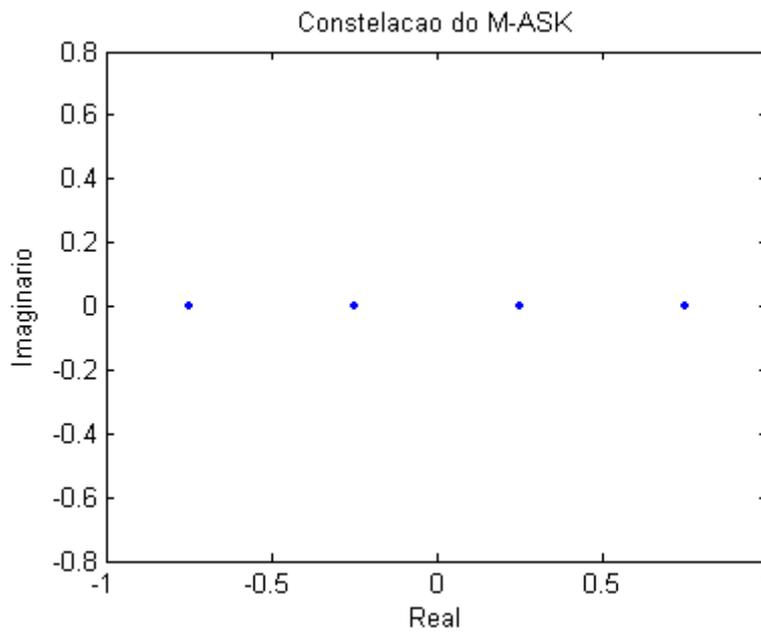


Fig. 2.12 – Constelação do M-ASK

O QAM é outro tipo de modulação M-Ária bastante utilizado. Essa modulação é uma combinação do M-ASK e do M-PSK. A fórmula geral do sinal QAM é dada pelas equação (2-22), onde $i = 1, 2, \dots, M$, $r_i = \sqrt{a_i^2 + b_i^2}$, $\theta_i = \tan^{-1}(b_i / a_i)$. No M-ASK, $\theta_i = 0$ para todos os símbolos sendo apenas r_i diferente. No M-PSK, r_i é constante para todos os símbolos e apenas θ_i é diferente. Como no M-QAM tanto r_i quanto θ_i são diferentes, esse se torna o motivo pelo qual o M-QAM é uma combinação dos dois. A equação

(2-23) é uma outra forma de representar o sinal M-QAM e graficamente, a Fig. 2.13, mostra a constelação do M-QAM quando M=16.

$$\begin{aligned} s(t) &= p'(t)(a_i \cos \omega_c t + b_i \sin \omega_c t) \\ &= p'(t)[r_i \cos(\omega_c t + \theta_i)] \end{aligned} \quad (2-22)$$

$$g(t) = x(t) + jy(t) = R(t)e^{j\theta(t)} \quad (2-23)$$

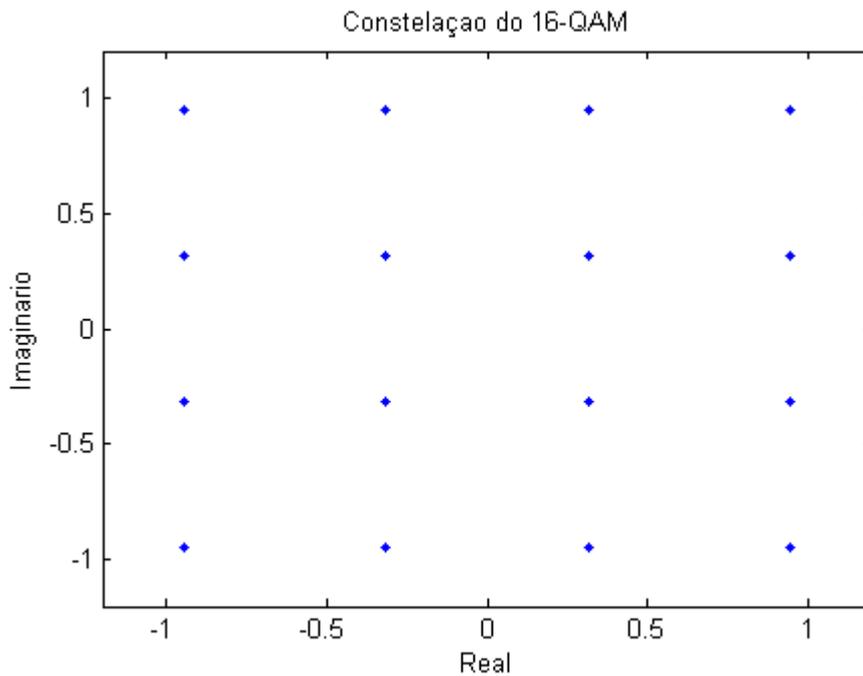


Fig. 2.13 – Constelação do 16-QAM

2.5.4. Probabilidade de Erro em Modulação Digital

Ao ser transmitido por um canal, o sinal é distorcido e sofre interferências. A seguir, será feita uma breve descrição do processo de detecção de um sinal digital, assumindo que a interferência em um sinal com ruído branco aditivo gaussiano (AWGN), detalhado melhor por [24].

Considere um canal AWGN em que os bits 0 e 1 sejam enviados com probabilidade iguais utilizando um pulso positivo e um pulso negativo respectivamente.

O pulso enviado correspondente a 1 é $p(t)$ o pulso enviado correspondente a 0 é $-p(t)$. O valor máximo desse pulso deve ser A_p em um instante $t = T_p$.

Pelo fato do canal ser ruidoso, os sinais recebidos serão $\pm p(t) + n(t)$, onde $n(t)$ é o ruído. Para o ruído gaussiano, a função densidade de probabilidade é dada pela equação (2-24), onde σ_n é o desvio padrão dessa densidade. Pela simetria do problema, leva-se em consideração que o limiar de detecção do pulso é 0, ou seja, se a soma do pulso com o ruído for um valor menor que 0, considera-se que o bit recebido é 0, caso for um valor maior que zero, considera-se que o bit recebido é 1. Pelo fato da amplitude do ruído ser infinita, $(-\infty, \infty)$, o valor amostrado $-A_p + n$ pode ocasionalmente ser positivo, levando a uma falsa interpretação. Da mesma forma, $A_p + n$ pode ocasionalmente ser negativo, levando também a uma falsa interpretação. Se um 0 for transmitido, ele será detectado como 1 se $-A_p + n > 0$, ou seja, se $n > A_p$.

A Fig. 2.14 mostra o gráfico da equação (2-24), a marcação dos valores máximos dos pulsos $-A_p$ e A_p e também uma área sombreada com a probabilidade de erro se um 0 for transmitido ($n > A_p$).

$$P_n(n) = \frac{1}{\sigma_n \sqrt{2\pi}} e^{-n^2/2\sigma_n^2} \quad (2-24)$$

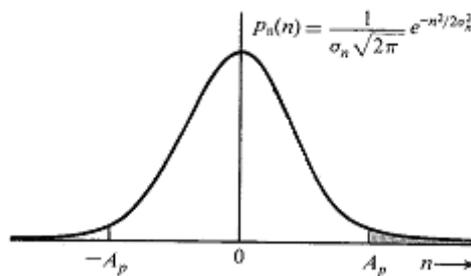


Fig. 2.14 – Função densidade de probabilidade para o ruído gaussiano

Sendo assim, a probabilidade de erro nesse caso, será a área da região sombreada, definida pela equação (2-25).

$$p_e = \frac{1}{\sigma_n \sqrt{2\pi}} \int_{A_p}^{\infty} e^{-x^2/2\sigma_n^2} dx \quad (2-25)$$

A equação (2-25) também pode ser escrita por (2-26), levando em consideração (2-27). A função $Q(y)$ é então definida. O seu gráfico é traçado na Fig. 2.15.

$$p_e = Q\left(\frac{A_p}{\sigma_n}\right) \quad (2-26)$$

$$Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^{\infty} e^{-x^2/2} dx \quad (2-27)$$

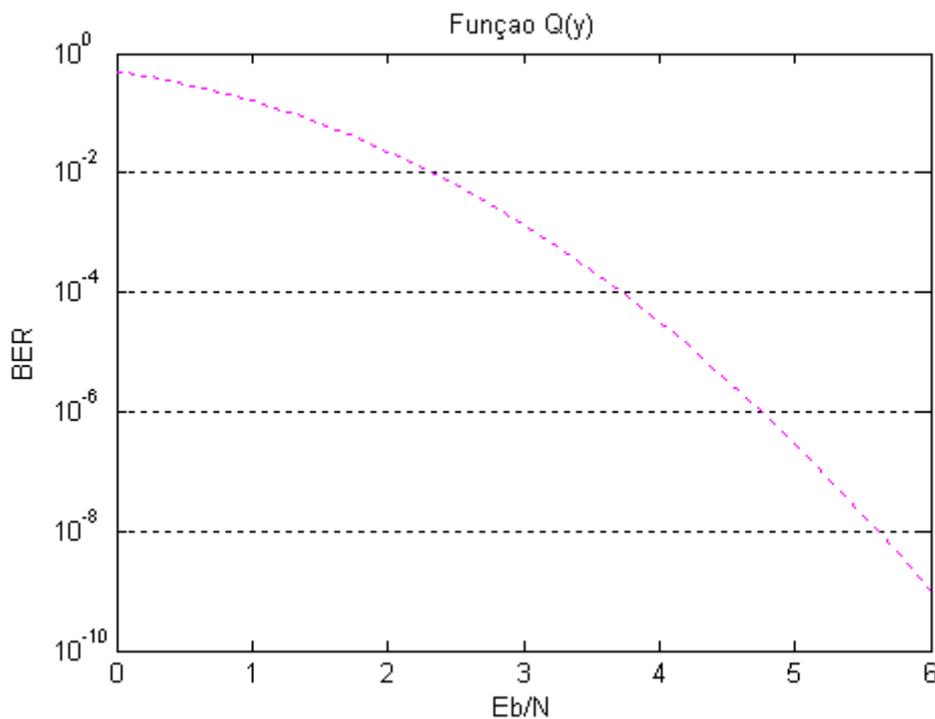


Fig. 2.15 – Gráfico da função Q(y)

Geralmente, a probabilidade de erro em modulação digital é descrita com apoio da função $Q(y)$. Além disso, a probabilidade de erro de símbolo pode ser expressa em relação a um parâmetro base, chamado de Energia de bit, ou simplesmente E_b . O valor

de E_b é calculado com base na energia de cada um dos pulsos dos bits enviados e está detalhada em [24]. A energia de cada pulso $g(t)$ é dada pela equação (2-28).

$$E_g = \int_{-\infty}^{\infty} g(t)^2 dt \quad (2-28)$$

Sendo assim, a probabilidade de erro de símbolo para cada uma das modulações listadas, em função de $Q(y)$ e em função da Energia de bit E_b é dada pela Tabela 2-2 [24] [25].

Tabela 2-2 – Probabilidade de erro de símbolo para cada modulação digital

ASK (Detecção Coerente)	$P_b = Q\left(\sqrt{\frac{E_b}{N}}\right)$
FSK (Detecção Coerente, com desvio de frequência Δf otimizado)	$P_b = Q\left(\sqrt{\frac{1,217E_b}{N}}\right)$
PSK	$P_b = Q\left(\sqrt{\frac{2E_b}{N}}\right)$
M-PSK	$P_b \approx 2Q\left(\sqrt{\frac{2E_b \log_2 M}{N}} \sin \frac{\pi}{M}\right)$
M-QAM $M = 2^k$	$P_{eM} \leq 4Q\left(\sqrt{\frac{3kE_b}{(M-1)N}}\right)$

A Fig. 2.16 mostra a equação do M-QAM traçada para três casos. Nela é possível verificar a probabilidade de erro de símbolos para 4-QAM, 16-QAM e 64-QAM. Esse tipo de gráfico é bastante comum em comunicações digitais e esses resultados foram utilizados nesse trabalho para verificar a confiabilidade dos simuladores utilizados.

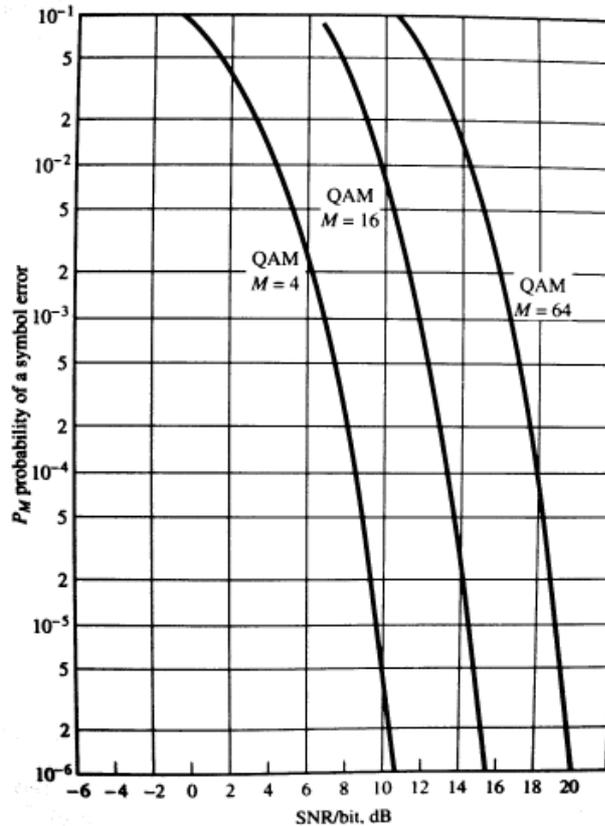


Fig. 2.16 – Probabilidade de erro de símbolo para o M-QAM [24].

2.6. MODELO DE DIVISÃO EM CAMADAS

O modelo de divisão em camadas pode ser considerado um dos fatores que contribuiu para o grande avanço atingido pelos sistemas de comunicações modernos nas últimas décadas. Por meio desse modelo, tornou-se possível dividir o processo de comunicação de dados em várias etapas em que cada uma realizasse o seu trabalho sem se preocupar com a etapa anterior ou a seguinte. Antes da existência desse modelo, todas as camadas eram desenvolvidas diferentemente em qualquer sistema de comunicação que era produzido. Sendo assim, não havia reaproveitamento de funções realizadas por cada uma das camadas [26].

Essa ideia de se reutilizar as funções executadas por um sistema de comunicação começou a tomar forma nos final dos anos 70, quando a ISO (*International Organization for Standardization*) desenvolveu o modelo OSI (*Open Systems Interconnection*) que previa que os sistemas de comunicações deveriam estar divididos

em 7 camadas, quais sejam: camada de aplicação, camada de apresentação, camada de sessão, camada de transporte, camada de rede, camada de enlace e por fim, a camada física. Cada uma dessas 7 camadas têm uma função bem específica e pode-se resumir no seguinte:

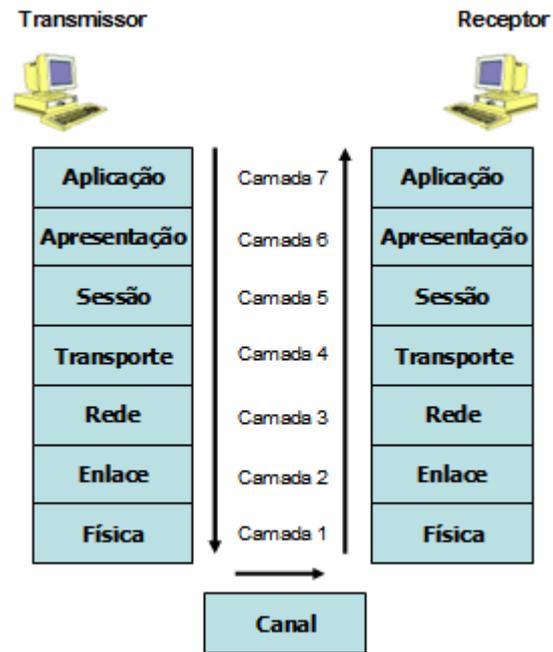


Fig. 2.17 – Modelo OSI

Camada de Aplicação. É onde reside a funcionalidade final para o sistema de comunicação. Nela existem protocolos como HTTP (*HyperText Transfer Protocol*), base da *Web*, além de possuir protocolos de comunicação instantânea, voz sobre IP, entre outros.

Camada de Apresentação. Essa camada é responsável por prover serviços que permitam que as aplicações de comunicação interpretem o significado dos dados trocados. Dentre esses serviços estão a compressão de dados e a codificação de dados, assim como a descrição de dados.

Camada de Sessão. Provê a delimitação e a sincronização da troca de dados, incluindo os meios de construir um esquema de pontos de verificação e de recuperação.

Camada de Transporte. Responsável pelo controle de entrega de informações. Dependendo do tipo de aplicação que será transportado, pode concentrar serviços de entrega garantida de pacotes, controle de fluxo fim-a-fim ou caso esse tipo de serviço não seja requerido, os serviços dessa camada podem se resumir à entrega não garantida de pacotes e sem controle de fluxo e congestionamento, o que traz a agilidade necessária em aplicações de tempo real.

Camada de Rede. Responsável pela movimentação de pacotes. Determina a rota e o caminho que a informação, dividida em datagramas, deverá seguir. Essas rotas podem ser criadas estaticamente ou dinamicamente por meio de protocolos de roteamento. Essa camada, então, é a responsável por entregar um pacote fim a fim.

Camada de Enlace. Responsável por detectar e corrigir erros que possam ter sido gerados na camada física, delimitação do tamanho dos quadros e controle de fluxo.

Camada Física. Responsável pela transmissão de bits brutos pelo canal de comunicação. Define características de dispositivos elétricos, ópticos e magnéticos. Em seu nível mais superior, contém como os bits serão interpretados, o que é chamado de modulação.

Alguns trabalhos mostram caminhos de evolução dos sistemas de comunicação em que informações inter-camadas podem ser utilizadas na otimização de sistemas de comunicações, entretanto, nesses casos, dados da camada física é compartilhada para as camadas superiores [27].

Os resultados apresentados por este trabalho permanecerão na camada física, entretanto, a plataforma computacional desenvolvida permite simulações intercamadas bastando que cenários sejam previstos e simulados.

3. ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (GAs, do inglês *Genetic Algorithms*) têm sido vastamente utilizados na solução de diversos problemas de engenharia em que a quantidade de soluções possíveis é muito grande ou é complexa. Os seus princípios foram inicialmente propostos por John Holland [28] na década de 1970 e desenvolvidos em seguida, por ele e seus alunos.

GAs são inspirados no mecanismo de seleção natural de Darwin em que indivíduos melhores adaptados ao meio tendem a sobreviver. Com isso, esses indivíduos têm maior possibilidade de gerar descendentes, o que faz que, depois de diversas gerações, restem os indivíduos melhores adaptados.

No paradigma dos GAs, considera-se que cada indivíduo seja uma possível solução do problema. No início, são geradas diversas possíveis soluções, de forma aleatória ou de forma conduzida (caso se tenha melhor conhecimento do problema). A qualidade dessas soluções é medida por uma função *fitness* que verifica a adequação da solução ao resultado esperado. Quanto maior o valor dessa função, maior será a probabilidade dos indivíduos serem eleitos “pais” que gerarão filhos para próximas gerações. Com o passar de diversas gerações, restam os melhores indivíduos, ou seja, as melhores soluções para o problema [28-31].

3.1. DEFINIÇÕES DOS ALGORITMOS GENÉTICOS

Para o bom entendimento dos algoritmos genéticos, faz-se necessário apresentar algumas definições que são inspiradas na biologia.

Genes e Cromossomos. Assim como no processo de seleção natural, os genes são o elemento primário dos algoritmos genéticos. Geralmente são representações de parâmetros das soluções utilizadas na otimização. Para os algoritmos genéticos, um conjunto de genes é considerado um cromossomo. Os cromossomos podem ser representados por uma cadeia de bits (0s e 1s), uma cadeia de números reais [33] ou então a combinação de ambos. Na biologia, um indivíduo geralmente possui diversos

cromossomos, entretanto, por simplicidade, pode-se dizer que, em GAs, um cromossomo é o mesmo que um indivíduo ou uma possível solução para o problema.

Populações e Gerações. Nos algoritmos genéticos, uma gama de soluções em um instante de tempo é considerada uma população. Essas populações podem ser consideradas como a unidade de iteração que o algoritmo genético utiliza para realizar a busca pela solução ótima. Cada iteração realizada pelo algoritmo genético também pode ser chamada de geração.

Pais e Filhos. Após o processo de inicialização em que a população inicial é criada, são escolhidos pares de indivíduos, utilizando processos de seleção que serão discutidos posteriormente. Cada par desses indivíduos (pais) gera outros dois indivíduos (filhos). O processo de geração dos filhos envolve operações de *crossover* e mutação, que também serão discutidas posteriormente. Essas operações são utilizadas para gerar diversidade na população sem que as características otimizadas em gerações anteriores não sejam perdidas.

Outros conceitos de algoritmos genéticos serão tratados com maiores detalhes nas páginas seguintes.

3.2. FUNÇÃO *FITNESS* (FUNÇÃO DE APTIDÃO)

A função *fitness* dá uma medida de adequação para cada indivíduo da população do GA. Pode-se dizer que a função *fitness* e a codificação de soluções na forma de cromossomos são as únicas ligações entre o GA e o problema de otimização envolvido. Como cada indivíduo é avaliado por essa função, é natural que o resultado dessa função seja diferente para cada indivíduo. É esse valor que será utilizado pelos métodos de seleção para escolher os pais que gerarão filhos para as próximas gerações.

Para que a função *fitness* seja aplicada de forma bem sucedida, são necessários alguns requisitos básicos:

- a. O valor retornado pela função *fitness* deve, de alguma forma, ser proporcional à qualidade da solução/indivíduo;
- b. A função *fitness* deve ser preferencialmente um valor positivo.

Para exemplificar como funciona a função *fitness*, suponha um problema de otimização simples por GA em que o objetivo principal é encontrar um segredo que guarda uma sequência correta de 10 números. A função *fitness* poderia ser definida para retornar a quantidade de números da sequência que estivesse correta. Essa função simples atenderia muito bem aos dois requisitos expostos anteriormente, ou seja, nesse caso, o valor retornado pela função seria proporcional à qualidade da solução e ainda, o valor retornado seria sempre positivo.

É claro que em problemas reais, nem sempre é fácil escrever uma função que atenda a esses requisitos. Por esse motivo, existem algumas formas de se escrever o relacionamento entre a qualidade da solução, que aqui será chamada de função objetivo, e a função *fitness*.

3.2.1. Função Objetivo

Esta primeira forma é exatamente o que foi feito no exemplo acima. A função *fitness* simplesmente mapeia o resultado da função objetivo. Nesse caso, pode-se escrever matematicamente que

$$f_i = o_i \quad (3-1)$$

onde f_i é a função *fitness* e o_i é a função objetivo.

3.2.2. Fitness Linearmente Proporcional

Esta outra forma permite maior controle sobre a função *fitness*, matematicamente ela é escrita pela equação (3-2).

$$f_i = ao_i + b \quad (3-2)$$

Nesse caso, tem-se que a e b são constantes escolhidas com o objetivo de permitir o equilíbrio entre a média da função objetivo e a média da função *fitness*. Esse

método pode reduzir o efeito do chamado “*genetic drift*” que é um efeito gerado pela aleatoriedade do processo reprodutivo que pode gerar indivíduos com um bom *fitness* mas que, no entanto, não foram resultados do processo de evolução propriamente dito [30].

Esse formato também permite que o GA minimize a função objetivo. Um GA simples, que tenha as funções de seleção, *crossover* e mutação implementadas (que serão mencionadas posteriormente), são funções maximizadoras por natureza. Para encontrar um mínimo, pode-se utilizar esse método fazendo com que a constante *b* seja o valor máximo esperado da função objetiva. Escrevendo a função *fitness* como mostrado em (3-3), esse objetivo será atingido.

$$f_i = Max - ao_i \quad (3-3)$$

Apesar disso, há outras formas de encontrar um mínimo sem que haja necessidade de alterar a função *fitness* para esse fim. Esse método será discutido posteriormente no tópico de seleção baseada no método do torneio.

3.2.3. Limite Sigma

Esse método evita de forma mais elegante o aparecimento de valores negativos na função *fitness*, além de dispensar a escolha de constantes que exigem conhecimento prévio do problema. Esse método é representado pela equação (3-4) onde *c* é um inteiro pequeno, \bar{o} é a média e σ é o desvio padrão dos valores da função objetiva.

$$f_i = o_i - (\bar{o} - c\sigma) \quad (3-4)$$

Para prevenir valores negativos na função *fitness*, qualquer valor em que $f < 0$ é arbitrariamente atribuído zero. Não é utilizado o valor do desvio padrão em indivíduos que possuem o valor do *fitness* menores que a constante *c*.

3.3. SELEÇÃO

A seleção é um dos princípios básicos de funcionamento dos GAs. A seleção vai permitir que, depois de muitas gerações, os indivíduos mais aptos sejam preferencialmente escolhidos para que gerem descendentes. Como mencionado anteriormente, os GAs começam com uma população inicial, que pode ter sido escolhida aleatoriamente (quando não se tem conhecimento da influência de cada gene no resultado da solução) ou então escolhida de forma conduzida (quando se conhece, a priori, a região da solução).

De forma que o processo de seleção privilegie os indivíduos mais adaptados, é atribuída uma nota para cada cromossomo da população. De acordo com essa nota, podem ser utilizados diversos métodos para se escolher quais os indivíduos comporão a próxima geração. Serão detalhados três métodos: o método da roleta, o método do torneio e o método da amostragem universal estocástica.

3.3.1. Método da Roleta

O método da roleta é o mais simples e o mais utilizado. Esse nome é utilizado em analogia a uma roleta de jogos de azar, em que cada indivíduo recebe uma fatia da roleta proporcionalmente ao seu *fitness*. De forma a exemplificar esse fato, considere que, em um determinado momento, existam sete indivíduos com os *fitness* e seus valores relativos (em relação a toda a população) listados na Tabela 3-1.

Tabela 3-1 – Tabela hipotética com indivíduos e os seus *fitness*

Indivíduo	<i>Fitness</i>	<i>Fitness Relativo</i>
1	3	0,075
2	10	0,250
3	4	0,100
4	8	0,200
5	2	0,050
6	7	0,175
7	6	0,150

Nesse caso, a escolha do indivíduo seria o equivalente a “girar a roleta” ilustrada na Fig. 3.1, em que cada indivíduo recebe fatia proporcional ao valor do *fitness*. Ou seja, o indivíduo 2 por exemplo, teria 25 % de chance de ser escolhido. Com isso, os indivíduos são selecionados até compor toda a geração.

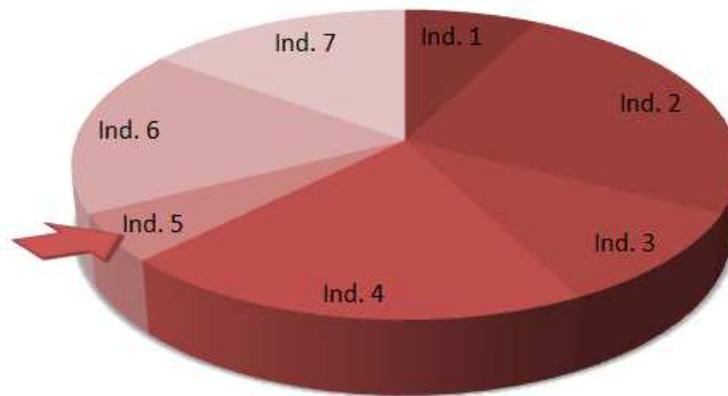


Fig. 3.1 – Ilustração que mostra como os indivíduos são selecionados pelo método da roleta

3.3.2. Método do Torneio

No método do torneio são escolhidos um número n de indivíduos e o indivíduo que estiver melhor adaptado (maior *fitness*) será o indivíduo escolhido. O método é repetido até que a população seja totalmente preenchida. Um valor típico para n é 3. A Fig. 3.2 exemplifica a seleção por torneio para os mesmos indivíduos da Tabela 3-1 em que $n = 3$.

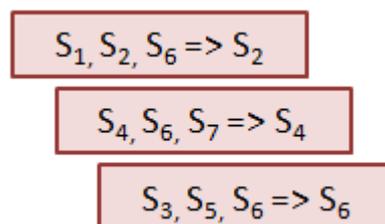


Fig. 3.2 – Exemplificação do Método do Torneio

O método do torneio também pode ser utilizado de forma a escolher o menor *fitness*. Isso pode ser necessário quando o menor valor do *fitness* significa que o indivíduo está mais bem adaptado.

3.3.3. Método da Amostragem Universal Estocástica

O método da amostragem universal estocástica (do inglês, SUS – *Stochastic Universal Sampling*) é baseado no método da roleta. A diferença é que se escolhem n indivíduos de uma só vez por meio de colocação de n agulhas igualmente espaçadas. Assim, em vez de girar n vezes a roleta gira apenas uma vez.

3.4. OPERADORES GENÉTICOS

Após os possíveis pais terem sido selecionados pelos métodos de seleção listados anteriormente, faz-se necessário realizar algumas operações para que a próxima população seja diferente da anterior mas que mantenha características dos seus geradores. Isso é importante para que a população se diversifique e que, ao mesmo tempo, continue com características adaptativas adquiridas em gerações anteriores. Para isso, existem dois operadores: a mutação e o *crossover*.

3.4.1. Mutação

A mutação é o operador utilizado para introduzir novos elementos na população. Alterando arbitrariamente um ou mais parâmetros da estrutura é introduzida diversidade genética, fazendo com que seja possível mudar levemente a direção de busca, permitindo que seja resolvido o problema dos mínimos locais. Com isso, a probabilidade de não se chegar a qualquer ponto do espaço de busca é quase zero.

A Fig. 3.2 exemplifica a mutação realizada em um cromossomo, com representação em 6 bits. Em (a), tem-se o estado antes da mutação e em (b) o estado depois da mutação. Como pode ser notada, a mutação fez com que o segundo bit fosse alterado, e com isso, houve modificação do parâmetro.

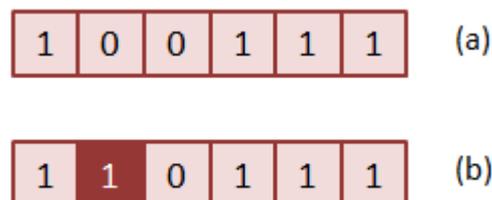


Fig. 3.3 – Exemplo de mutação realizada em parâmetros binários

A mutação pode ser aplicada aos indivíduos com taxa de $(0 \leq P_m \leq 1)$, mas geralmente utiliza-se um valor pequeno $(0,001 \leq P_m \leq 0,01)$ por se tratar de um operador secundário.

O operador de mutação também pode ser utilizado em cromossomos com representação real. Suponha que, em um determinado problema de otimização existam três parâmetros que possam variar de 0 a 3. Com isso, a mutação é realizada com probabilidade P_m sorteando um novo valor entre 0 e 3 e substituindo o parâmetro antigo pelo novo assim como mostrado pela Fig. 3.4.

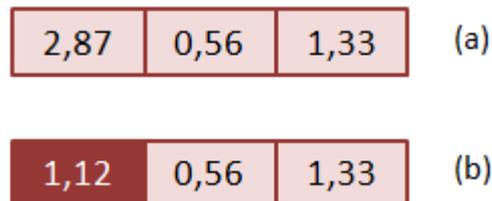


Fig. 3.4 – Exemplo de mutação realizada em números reais

3.4.2. Crossover

O *crossover* é o responsável pela recombinação das características dos pais durante a geração de um novo indivíduo para a próxima população. Após terem sido escolhidos dois indivíduos geradores, é aplicado esse operador com probabilidade P_c entre $(0 \leq P_c \leq 1)$. Por ser o operador predominante, ele é aplicado na prática com taxas que variam entre $(0,60 \leq P_c \leq 0,99)$.

Crossover em um ponto. A forma mais simples de realizar essa combinação é escolher um ponto e realizar a troca entre os dois pais a partir desse local o que é mostrado na Fig. 3.5. Com isso, é possível gerar dois filhos formados pela combinação dos pais, Fig. 3.6.

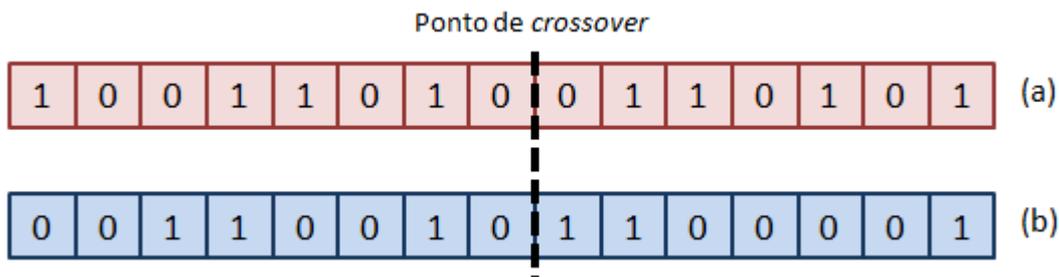


Fig. 3.5 – Operador *crossover* sendo aplicado em um ponto fixo em dois pais (a) e (b)

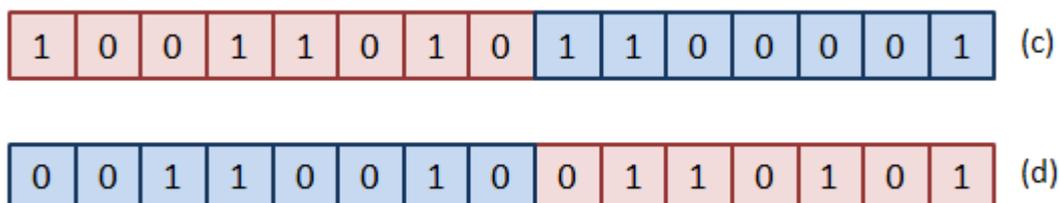


Fig. 3.6 – Filhos gerados pelos pais após aplicado o operador *crossover*

O ponto escolhido para realizar o *crossover* pode ser fixado ou escolhido de forma aleatória. Entretanto, a aleatoriedade na escolha do ponto permite maior grau de liberdade nessa operação, principalmente em casos no qual não se tem certeza da influência de cada parâmetro na solução.

***Crossover* mutipontos.** Pode-se ampliar a liberdade do *crossover* permitindo que ele seja realizado em mais de um ponto. Na verdade, essa é apenas uma generalização do caso anterior que é mostrado na Fig. 3.7, onde (a) e (b) são os pais e (c) e (d) são os filhos gerados após o *crossover* em dois pontos.

Schemata. Também conhecido como máscara ou cruzamento uniforme. Esse método não utiliza pontos de cruzamento específicos, mas sim, uma máscara que indica quais os parâmetros que devem ser modificados. Ele deve ser utilizado quando já se tem conhecimento anterior sobre a influência de alguns parâmetros, e por esse motivo se deseja os mantê-los fixos.

***Crossover* em genes reais.** Cabe-se ressaltar que, apesar dos exemplos anteriores terem sido realizados com parâmetros baseados em *bits*, o *crossover* também

pode ser realizado em parâmetros de números reais. A mecânica é a mesma e está exemplificada na Fig. 3.8 em que os pais estão em (a) e (b) e os filhos em (c) e (d).

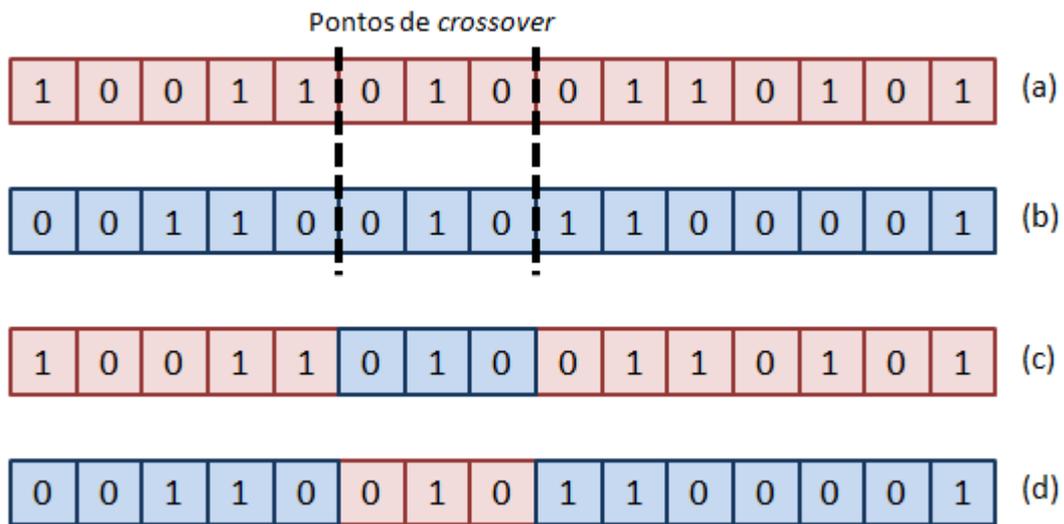


Fig. 3.7 – *Crossover* multiponto

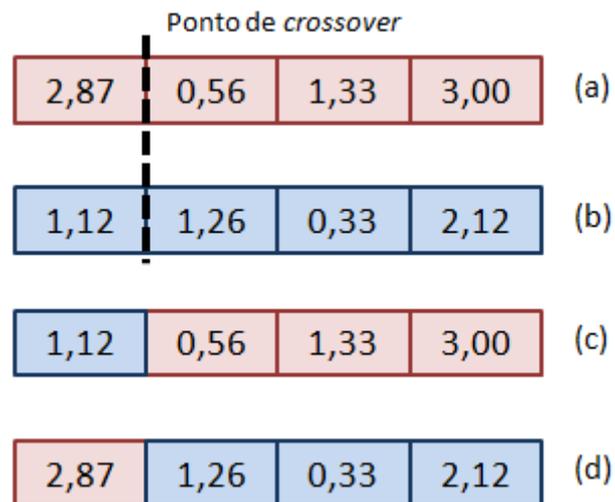


Fig. 3.8 – *Crossover* realizado em cromossomos com genes reais

4. PLATAFORMA COMPUTACIONAL

Esse trabalho utilizou alguns softwares para realizar a síntese de antenas e a análise de sistemas de comunicação. A seguir, será detalhado cada um dos sistemas, *softwares* e bibliotecas utilizados para o desenvolvimento desse trabalho.

4.1. O JAVA E SUA PORTABILIDADE

Anunciada formalmente pela então *Sun Microsystems* em 1995, a linguagem de programação Java, ou somente Java, foi um projeto bem sucedido idealizado nos anos 1990 por Patrick Naughton, Mike Sheridan e James Gosling. Naquela época, a linguagem que era chamada simplesmente de *Oak* (carvalho, em português) já tinha como objetivo a portabilidade. Os três pesquisadores tinham a percepção que em um futuro breve os computadores, equipamentos eletrônicos e eletrodomésticos iriam de alguma forma se interagir e por esse motivo havia a necessidade de uma linguagem portátil que permitisse que o mesmo programa pudesse ser utilizado em equipamentos diferentes. Desde então a linguagem se disseminou bastante chegando a nove milhões de desenvolvedores no ano de 2012 com mais de três bilhões de dispositivos que rodam aplicativos com essa linguagem. Mais informações sobre o histórico da linguagem em [33-35].

Ao contrário de outras linguagens de programação de alto nível, como o C, C++, ou Pascal, em que o código é compilado e traduzido para a linguagem de máquina, na linguagem Java, o código quando compilado é traduzido para uma linguagem intermediária chamada de *bytecode*. Essa linguagem intermediária é interpretada por uma Máquina Virtual Java (JVM – *Java Virtual Machine*) que tem o papel de ler o *bytecode* traduzindo-o para instruções de baixo nível de acordo com a plataforma em que a máquina virtual está executando. Dessa forma, é introduzida uma camada extra que permite a portabilidade, algo similar como o que foi feito com as redes de comunicação, através do Modelo de Comunicação em Camadas, comentado no Capítulo 2 [37].

Além da portabilidade, o Java também incluiu o conceito de programação orientada a objetos, o que trouxe modernidade. Atualmente, o Java é considerado muito mais do que uma linguagem de programação. Ele é toda uma infraestrutura que permite desenvolver aplicativos para praticamente qualquer ambiente, desde para pequenos dispositivos móveis, pequenos aplicativos para páginas web ou até grandes soluções para grandes corporações. Não é objetivo desta tese esgotar esse assunto, mesmo porque há bibliografia extensa publicada sobre a linguagem Java. Deseja-se nesse ponto detalhar algumas das tecnologias utilizadas dessa linguagem de programação na construção, utilização e entendimento desse *framework* produzido nos últimos anos para análise e síntese de antenas de acordo com os parâmetros dos sistemas de comunicação.

4.1.1. Classes Java

As classes Java são as unidades elementares para se construir qualquer programa em Java. Os códigos fontes são arquivos textos, geralmente com a extensão “.java” e que quando compiladas, são traduzidas para a linguagem intermediária *bytecode* e armazenadas em arquivos “.class”.

A Fig. 4.1 mostra um exemplo simples de uma classe chamada “Wire”, que possui as variáveis reais do x, y e z inicial e final das posições de um fio.

```
class Wire {  
    double x1,x2,y1,y2,z1,z2;  
}
```

Fig. 4.1 – Exemplo de código de uma classe Java

Existem alguns pontos importantes sobre as classes Java que devem ser pontuados:

1. Os nomes das classes devem iniciar-se por um caractere em maiúsculo;
2. O nome do arquivo “.java” deve coincidir com o nome da classe;
3. O código Java é *case-sensitive*, ou seja, caracteres maiúsculos são diferentes de caracteres minúsculos.

4.1.2. Empacotamento de arquivos – Pacotes JAR

Os pacotes JAR são uma forma de simplificar a distribuição de um programa, aplicativo ou componente Java. Além de conter toda uma estrutura de classes Java, os pacotes JARs podem conter imagens, sons e tudo o que for necessário para a execução de um aplicativo ou componente. Uma das vantagens deste empacotamento é que os JARs na verdade são arquivos compactados no formato ZIP, bastando renomeá-los para a extensão “.zip” para conseguir abri-los e explorá-los em aplicativos como o *WinRAR* ou *Winzip*.

Os aplicativos *Netbeans* e *Eclipse*, utilizados para desenvolver aplicativos Java, possuem suporte nativo a criação de arquivos JARs. Para sinalizar que uma classe pertence a um pacote JAR, deve-se utilizar o modificador *package* no início da classe, conforme listado na Fig. 4.2. Nesse exemplo, é sinalizado que a classe *Wire* pertencerá ao pacote *GRADMAX.jar*.

```
package GRADMAX;
class Wire {
    double x1,x2,y1,y2,z1,z2;
}
```

Fig. 4.2 – Exemplo de código de uma classe Java usando a instância *package*

4.1.3. Applets Java

Os *applets* são programas escritos em Java que podem ser incluídos em uma página web com a mesma facilidade que se inclui uma imagem. Quando se usa essa tecnologia, o código do *applet* é transferido para o computador e executado localmente pela máquina virtual Java do *browser* automaticamente sem que haja intervenção do usuário. Esse fato exige algumas restrições de segurança o que faz com que os *applets* sejam executados em uma *sandbox* que restringe acesso a sistemas de arquivos e à área de transferência para proteger os computadores de códigos maliciosos [38]. Isto impede, que, por exemplo, um arquivo seja salvo diretamente em disco pelo programa em execução. Apesar dessas restrições, os *applets* são bastante úteis, possuindo aplicações em diversas áreas, sendo usado como facilitador para o ensino de diversas disciplinas

como física, química, eletromagnetismo, criptografia, entre outras. Alguns exemplos estão listados em [39][40].

4.1.4. Java Swing

Java *Swing* é uma funcionalidade do Java que permite o desenvolvimento de aplicativos Java com GUI (*Graphics Users Interface*). Essa funcionalidade permite a inclusão de janelas, botões, textos, tabelas, entre outros e podem ser utilizados em conjunto com Java *Applets*. Sendo assim, os miniaplicativos Java que estão sendo executados em uma janela web podem conter todas as funcionalidades de um aplicativo comum bastando utilizar Java *swings*.

4.2. WEBPRAC

O PRAC (*Parabolic Reflector Analysis Code*) é um software amigável para a análise de antenas parabólicas que produz resultados com alta precisão. Ele já foi muito utilizado no ensino de antenas, em pesquisas e também pela indústria atingindo resultados notáveis, sendo comparável com softwares proprietários. Por ser um software gratuito, ele pode ser utilizado e distribuído sem nenhum custo.

O PRAC implementa a solução da integral de campo distante, descrita pela equação (4-1) – em que E^{rad} é o campo distante no ângulo Θ e ϕ – utilizando o método de Jacobi-Bessel [42].

$$\mathbf{E}^{\text{rad}}(\theta, \phi) = -jk\eta \frac{e^{-jkr}}{4\pi r} (\mathbf{I} - \hat{\mathbf{R}}\hat{\mathbf{R}}) \bullet \iint_{s'} \mathbf{J}(\mathbf{r}') e^{jkr' \bullet \hat{\mathbf{R}}} ds' \quad (4-1)$$

O WebPRAC é uma evolução do PRAC. O código do PRAC foi primeiramente escrito em Pascal e por esse motivo, só era possível ser executado em ambiente do sistema operacional DOS. Desde o Windows Millenium Edition, o sistema DOS não é mais utilizado dentro da família Microsoft. Quando há necessidade de utilizá-lo, é criada uma instância de emulação do DOS e logo em seguida, executado o código do PRAC. Essa emulação gera algumas incompatibilidades que dificultavam a utilização do software. Além disso, reescrever o mesmo código para linguagens atuais permitem que ele seja integrado com algoritmos inteligentes de forma a realizar projetos

dinâmicos de antenas utilizando como exemplo, a taxa de rejeição sistêmica como parâmetro norteador do projeto. Esses foram pontos motivadores para que o código do PRAC fosse reescrito na linguagem Java aliada com a tecnologia dos Applets Java [6][21].

Um manual de uso do WebPRAC está disponível no Apêndice APÊNDICE C e um manual de uso da biblioteca WebPRAC.jar, utilizada para integrar o WebPRAC com outros aplicativos está disponível no Apêndice APÊNDICE D. Recentemente, o WebPRAC foi citado como software de apoio ao ensino de antenas refletoras em [7].

4.3. GRADMAX FOR WEB

O GRAMAX *for Web* é um software desenvolvido primeiramente em Pascal e convertido para a linguagem Java que realiza análise de antenas e conjuntos de fios [16]. Entre as suas funcionalidades, pode-se citar: cálculo da impedância de entrada em todos os pontos de alimentação; cálculo do ganho máximo da estrutura dos fios; diagrama de radiação da estrutura de fios; otimização de conjuntos de monopolos; possibilidade de simulação do plano terra (plano XY).

Uma das vantagens desse *software* em relação a outros é que seu código leva em consideração o acoplamento mútuo, o que não ocorre em outros códigos de análise de conjuntos de antenas. Esse fato permite que se tenha maior precisão no projeto e análise de antenas, o que mostra um ganho em relação a outros *softwares*.

Além disso, o GRADMAX também analisa a correta irradiação eletromagnética realizada por cada um dos elementos, o que também não é facilmente encontrado em outros softwares de conjuntos já que em geral, é considerado que cada elemento do conjunto seja isotrópico [43][44]. Apesar de ter sido mostrado em alguns artigos que conjuntos isotrópicos são suficientes para uma análise qualitativa efetiva, mesmo assim, não garante que em uma síntese de um conjunto isso se torne suficiente [45].

O código do GRADMAX é baseado no método dos momentos introduzido pelo MININEC (*Mini Numerical Electromagnetics Code*). Além de analisar antenas, o

GRADMAX realiza, por padrão, dois tipos de otimizações: na primeira, altera-se o ângulo dos fios para aumentar o ganho na direção x. Na segunda, altera-se o espaçamento de monopolos para também aumentar o ganho na direção x [46].

Por ter sido escrito com a tecnologia de *Applets* Java, não se faz necessária a instalação do GRADMAX no computador do usuário. Para utilizá-lo, deve-se ter instalado na máquina do usuário o suporte a aplicativos Java e depois acessar o endereço da web descrito com o navegador preferido. Um manual de utilização do GRADMAX está disponível no Apêndice APÊNDICE A e um manual de utilização da biblioteca GRADMAX.jar, utilizada para integração do GRADMAX com outros sistemas, está disponível no Apêndice APÊNDICE B.

O trabalho de conversão do GRADMAX da linguagem Pascal para a linguagem Java não foi imediato e esbarrou em alguns problemas que retardaram bastante o trabalho. Um *bug* da função *round* do Pascal retornava valores incorretamente quando estava executando em modo *runtime* dentro do compilador *Turbo Pascal*. A inconsistência dos valores apresentados fazia com que os valores calculados em Java e em Pascal não fossem os mesmos, o que gerava dúvidas se o código traduzido para Java estava correto. Após algumas pesquisas na internet, verificou-se que realmente havia problemas com o funcionamento da função *round*. Para contornar o problema, escreveu-se uma função chamada “arredonda” que utilizava a função *trunc* do Pascal e assim, não foi mais necessário utilizar a função *round*. Após essa alteração, os valores calculados pelo Pascal foram exatamente os mesmos calculados pelo Java.

Recentemente, o GRADMAX foi citado como software de apoio ao ensino de antenas de fio em [7].

4.4. MATLAB

O MATLAB (MATrix LABoratory) é um software de alto desempenho voltado para o cálculo numérico. Esse *software* foi desenvolvido no fim dos anos 1970 com o intuito de realizar cálculos matriciais. Ele teve tanta aceitação que se difundiu e evoluiu por toda a comunidade acadêmica.

Além de realizar cálculos matriciais, hoje é possível implementar códigos de processamento de sinais, processamento de imagens, construção de gráficos, entre outras coisas. O MATLAB permite ainda a integração com códigos desenvolvidos em outras linguagens de programação como o C, Java e Fortran. Além disso, existem diversas bibliotecas e rotinas já desenvolvidas disponíveis na Internet.

Um dos fatores de escolha do MATLAB foi essa fácil possibilidade de integração, aliadas com um vasto repositório de códigos do MATLAB, o que permitiu ganhar velocidade na produção de códigos e na modelagem de sistemas. Os códigos escritos para MATLAB são escritos em linguagem própria e conhecidos como códigos M, ou simplesmente M.

Existem diversos manuais e tutoriais sobre o MATLAB, e por isso esse texto não abordará como esse *software* deve ser utilizado ou como ele funciona.

4.5. CML

O CML (*Coded Modulation Library*) é uma biblioteca escrita para rodar em MATLAB, desenvolvida pela Iterative Solutions que tem como objetivo simular de forma precisa e eficiente sistemas de comunicações modernos. Essa biblioteca não tem nenhum custo, pois é um software livre (*Open Source*) que permite a sua modificação e redistribuição de acordo com os termos da *GNU Lesser General Public* [22] [47]. Além disso, como está escrita em MATLAB, é possível uma integração com o Java, o que significa integração com o GRADMAX e WebPRAC.

Essa biblioteca permite a simulação de vários parâmetros sistêmicos como Modulação, Codificação de Canal, modelos de propagação, entre outros. É possível assim, montar os parâmetros de um enlace e simular o comportamento em relação a diversas figuras de mérito como o BER (*Bit Error Rate*), a capacidade de canal, a probabilidade de desvanecimento de blocos de bits, a capacidade em termos de banda, entre outros.

Além de permitir montar um enlace com parâmetros previamente definidos pelo usuário, essa biblioteca já possui previamente configurados, valores dos parâmetros de

padrões de sistemas de comunicação modernos como WiMAX (*Worldwide Interoperability for Microwave Access*), LTE (*Long Term Evolution*) e o DVB (*Digital Video Broadcasting*), o que torna possível simular rapidamente o comportamento desses sistemas sem que seja necessário consultar documentos de referências desses padrões para uma simulação concisa.

Como saída dessas simulações, o CML permite que sejam escolhidas diversas figuras de mérito para que se use como comparação entre os resultados obtidos. Como exemplo, pode-se comparar a taxa de BER (*Bit Error Rate*) por E_b/N_0 de um sistema WiMax com a mesma taxa de BER por E_b/N_0 em um sistema LTE. Além do BER, outros valores de referência podem ser utilizados como figura de mérito.

4.6. AVALIAÇÃO DO CML

Antes de utilizar o CML verificou-se a confiabilidade dessa biblioteca comparando o resultado dos cálculos produzidos por ela com resultados de publicações acadêmicas disponíveis. Para isso foram propostas duas simulações diferentes utilizando dois esquemas de modulação. Em uma simulação, foi proposto um enlace utilizando o esquema de modulação 16-QAM e em outra, um enlace com o esquema de modulação 64-QAM. Como resultado, o CML forneceu curvas de *SER* (*Symbol Error Rate*) por E_b/N que foram comparadas com resultados de bibliografias [24].

4.6.1. Avaliação de simulações do 16-QAM no CML

Para avaliação do 16-QAM em um canal AWGN (*Additive White Gaussian Noise*), foram utilizados os parâmetros do CML listados na

Fig. 4.3. Esses parâmetros são o registro número “3” do arquivo “uncodedscenarios.m” disponível pela biblioteca. Esses parâmetros na verdade traduzem que será simulada a relação E_b/N variando de 0 a 20 dB, com modulação 16-QAM, código de Gray sobre um canal AWGN.

```

record = 3;
sim_param(record).comment = 'Uncoded 16-QAM in AWGN w/ gray labeling';
sim_param(record).sim_type = 'uncoded';
sim_param(record).SNR = [0:0.5:20];
sim_param(record).SNR_type = 'Eb/No in dB';
sim_param(record).framesize = 100000;
sim_param(record).modulation = 'QAM';
sim_param(record).mod_order = 16;
sim_param(record).mapping = 'gray';
sim_param(record).channel = 'AWGN';
sim_param(record).bicm = 1;
sim_param(record).demod_type = 0;
sim_param(record).linetype = 'm: ';
sim_param(record).legend = sim_param(record).comment;
sim_param(record).filename = strcat( data_directory, 'QAM16AWGN.mat');
sim_param(record).reset = 0;
sim_param(record).max_trials = 10000*ones( size(sim_param(record).SNR) );
sim_param(record).minBER = 1e-6;
sim_param(record).max_frame_errors = 220*ones( size(sim_param(record).SNR) );
sim_param(record).save_rate = 100;

```

4.6.2. Avaliação de simulações do 64-QAM no CML

Da mesma forma realizada anteriormente, foi avaliado o comportamento do 64-QAM em um canal AWGN. Foram utilizados os parâmetros do CML listados na Fig. 4.4. Esses parâmetros são o registro número “12” do arquivo “uncodedscenarios.m” disponível na biblioteca. Esses parâmetros na verdade traduzem que será simulada a relação E_b/N variando de 0 a 20 dB, com modulação 64-QAM, código de Gray sobre um canal AWGN.

A Fig. 4.5 e a Tabela 4-1 mostram os resultados obtidos com essa simulação onde é possível verificar a taxa de erro de símbolos para a modulação 16-QAM e para a modulação 64-QAM. A Fig. 2.16 mostra essa mesma taxa de erro calculada matematicamente de acordo com as equações propostas por [24]. Os valores numéricos da Fig. 2.16 foram transcritos para a Tabela 4-1 na coluna “Bibliografia”.

Para fins de comparação, considere uma taxa de erro de símbolos de 10^{-4} . Segundo os resultados mostrados na Fig. 2.16, tem-se para o 16-QAM uma E_b/N de 13 dB e para o 64-QAM uma E_b/N de 18 dB. Esses resultados são bem próximos dos

obtidos com o CML (Fig. 4.5). Sendo assim, considerou-se essa biblioteca com bons resultados em relação à bibliografia publicada para as simulações que serão realizadas.

```

record = 12;
sim_param(record).comment = 'Uncoded 64-QAM in AWGN w/ gray labeling';
sim_param(record).sim_type = 'uncoded';
sim_param(record).SNR = [0:0.5:20];
sim_param(record).SNR_type = 'Eb/No in dB';
sim_param(record).framesize = 1e4;
sim_param(record).modulation = 'QAM';
sim_param(record).mod_order = 64;
sim_param(record).mapping = 'gray';
sim_param(record).channel = 'AWGN';
sim_param(record).bicm = 1;
sim_param(record).demod_type = 0;
sim_param(record).linetype = 'b: ';
sim_param(record).legend = sim_param(record).comment;
sim_param(record).filename = strcat( data_directory, 'QAM64AWGN.mat');
sim_param(record).reset = 0;
sim_param(record).max_trials = 1e5*ones( size(sim_param(record).SNR) );
sim_param(record).minBER = 1e-5;
sim_param(record).max_frame_errors = 20*ones( size(sim_param(record).SNR) );
sim_param(record).save_rate = 100;

```

Fig. 4.4 - Dados de entrada para simulação do CML – 64-QAM

Tabela 4-1- Comparativo entre os resultados do CML e da Bibliografia

Eb/N	Taxa de Erro 16-QAM		Taxa de Erro 64-QAM	
	Calculado – CML	Bibliografia	Calculado – CML	Bibliografia
8 dB	0.0366	0.500	-	-
10 dB	0.007	0.007	-	-
12 dB	0.0006	0.0007	0.0578	0.0500
14 dB	$1,0482 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	0.0130	0.0100
16 dB	-	-	0.0015	0.0020
18 dB	-	-	$5 \cdot 10^{-5}$	$7 \cdot 10^{-5}$

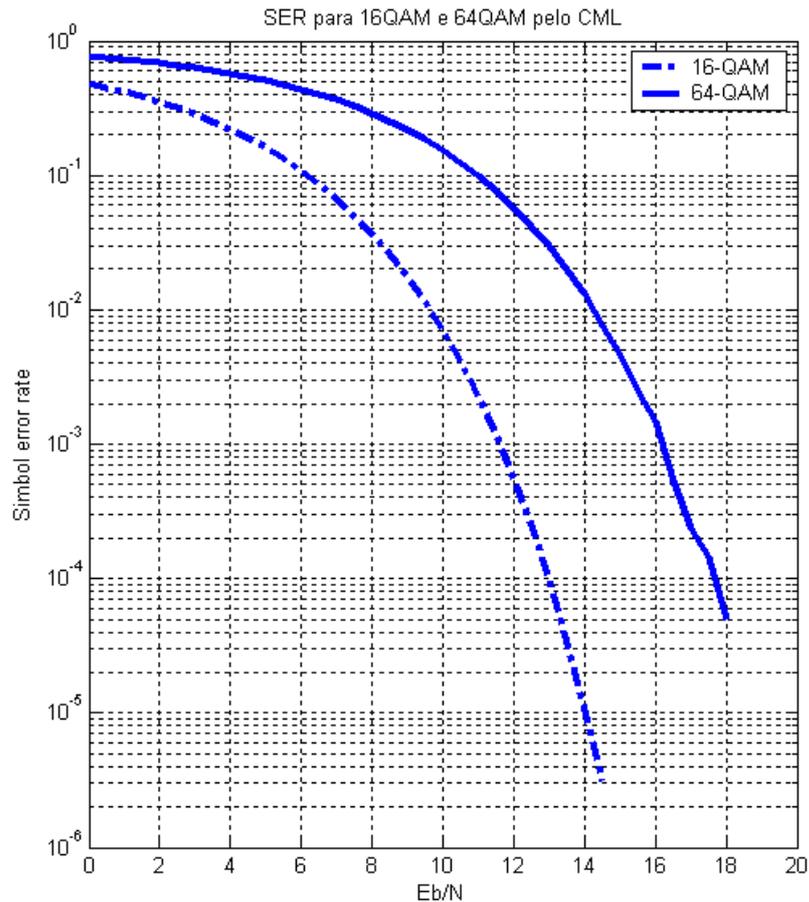


Fig. 4.5 – Taxa de erros de símbolos do 16-QAM e 64-QAM calculados pelo CML

4.7. A PLATAFORMA COMPUTACIONAL

A plataforma computacional é base fundamental desta tese. Ela é composta por:

- Algoritmo de análise de antenas
- Algoritmo de análise de sistemas de comunicação (CML);
- Camada de otimização.
-

Para integrar estes três componentes, era preciso convergir todos os códigos para um mesmo ambiente. Após algumas análises, decidiu-se utilizar o MATLAB como plataforma de integração entre os componentes, pois:

- Dois códigos para análise de antenas (GRADMAX e WebPRAC) escritos em linguagem Java, facilmente integrados ao MATLAB

conforme descrito pelos Apêndices APÊNDICE B e APÊNDICE D, estavam disponíveis;

- O CML está naturalmente disponível em MATLAB, apesar de parte do código estar escrito em bibliotecas com linguagem C;
- A camada de otimização ainda não estava construída. O MATLAB se mostrou um ambiente que poderia fornecer todas as funcionalidades para a construção do algoritmo de otimização.

Em seguida, será descrito como foi realizada a implementação da camada de otimização. Um manual de utilização da plataforma computacional está disponível no Apêndice APÊNDICE E.

4.7.1. Implementação da Camada de Otimização

A implementação da camada de otimização partiu do pressuposto de se utilizar um algoritmo genético ordinário, sem inclusão de mecanismos complexos de seleção ou *fitness* em um primeiro momento.

A camada foi escrita em dois núcleos: um núcleo responsável pela otimização de antenas de fio, codificado no arquivo “ga_core.m” e outro núcleo responsável pela otimização de antenas refletoras codificado no arquivo “ga_core_reflector.m”. Ambos os núcleos são responsáveis pelas tarefas de um algoritmo genético comum, como:

- Formar a população inicial;
- Executar as rotinas de cálculo de adequação (*fitness*) de cada indivíduo;
- Controlar rotinas de seleção;
- Controlar rotinas *crossover* e mutação;
- Verificar critérios de parada.

Além disso, os núcleos também são responsáveis por rotinas como:

- Gerar arquivo texto de resposta do algoritmo;

- Integrar com os algoritmos de análise de antenas (GRADMAX e WebPRAC)

4.7.2. Codificação de Possíveis Soluções para Antenas de Fio

A codificação das possíveis soluções para antenas de fio é realizada utilizando genes reais, em que cada gene é a tensão ou a fase de um ponto de alimentação da antena. Para isso, a otimização de antenas de fio requer que seja definido um conjunto de antenas base com indicações dessas alimentações. Esse conjunto de antenas pode ser projetado utilizando a interface de *applets* Java do GRADMAX como apoio - conforme manual no Apêndice APÊNDICE A. Após a definição do conjunto, deve ser utilizada a função *Export* para extração da *string* base, que é então inserida no algoritmo de otimização pelo parâmetro *string*, conforme descrito no Apêndice APÊNDICE E.

As informações inseridas são lidas, e então o algoritmo identifica automaticamente quantos *genes* serão necessários para a otimização. A primeira alimentação é sempre utilizada como referência, com tensão em $1 V$ e fase em 0 graus e todas as outras alimentações são considerados *genes* do algoritmo genético.

Assim, se a antena base possui cinco pontos de alimentação, o primeiro ponto será utilizado como referência e os demais quatro serão utilizados na otimização, sendo que, para cada um, haverá dois genes com representação real: um para a tensão e outro para a fase.

4.7.3. Codificação de Possíveis Soluções para Antenas Refletoras

A codificação de possíveis soluções para antenas refletoras torna-se um pouco mais simples do que a codificação para as antenas de fio, já que para as refletoras ainda não foi implementado no WebPRAC a possibilidade de realizar análise de conjunto de antenas.

Da mesma forma que a otimização de antenas de fio, a otimização de antenas refletoras requer que seja definida uma antena base que pode ser projetada utilizando a interface de *applets* Java do WebPRAC como apoio - conforme manual no Apêndice APÊNDICE C. Após a definição da antena, deve ser utilizada a função *Export* para

extração da *string* base, que é então inserida no algoritmo de otimização pelo parâmetro *string*, como mostrado no Apêndice APÊNDICE E.

As informações são lidas e são utilizados como genes codificados como parâmetros reais as posições x, y e z do alimentador.

4.7.4. Implementação da Função *Fitness* pelo Método do Erro Quadrático Médio

Conforme visto no Capítulo 3, a função *fitness* procura medir o nível de adequação de um indivíduo como solução do algoritmo genético. A equação (4-2) mostra uma implementação para o *fitness* sendo uma variação da função objetivo descrito em 3.2.2. Nesse caso, é utilizado o erro quadrático médio, em que *n* é a quantidade de pontos do diagrama de radiação, *ant_calc* é um vetor com os pontos do diagrama de radiação calculado para a solução e *ant_des* é um vetor com o diagrama de radiação desejado para a solução. A função da exponenciação é simplesmente para retirar valores negativos gerados da diferença entre *ant_calc* e *ant_des*. O valor de *K* é uma constante que deve ser ajustado de acordo com cada caso para evitar valores negativos.

$$fitness = n - \frac{\sum_{i=1}^n (ant_calc(i) - ant_des(i))^2}{K} \quad (4-2)$$

Por esse método, o valor do *fitness* será *n* quando o diagrama de radiação da antena calculada for exatamente igual ao diagrama de radiação da antena desejada. Quanto maior o *fitness*, mais próximo o diagrama de radiação calculado estará do diagrama de radiação desejado.

A plataforma computacional utilizará esse método para o *fitness* quando o parâmetro *tipo_fitness* receber o valor 1. Para antenas de fio, a quantidade de pontos *n* será sempre 180 (diagrama de radiação polar, com cada ponto representando 2 graus). Para antenas refletoras, a quantidade de pontos *n* dependerá da quantidade de pontos configurada na *string* WebPRAC.

4.7.5. Implementação da Função *Fitness* pelo Método das Máscaras

Outra forma de implementar a função *fitness* é por meio do uso de máscaras, conforme usado em [48]. Nesse caso, a função *fitness* é representada pela equação (4-3), onde n é a quantidade de pontos do diagrama de radiação, ant_calc é um vetor com os pontos do diagrama de radiação calculado para a solução e ant_des é um vetor com a máscara para a solução. Ao contrário do método anterior, em que a equação é aplicada em todos os pontos, nesse método, essa equação é utilizada apenas nos k pontos em que o diagrama calculado excede a máscara.

$$fitness = n - \sum_{i=1}^k ant_des(i) - ant_calc(i) \quad (4-3)$$

A Fig. 4.6 traz uma representação de uso dessa equação em que se destaca a região em que k pontos do diagrama da antena calculada excedem a máscara. Nesses k pontos são utilizados a equação (4-3). Os outros $(n-k)$ pontos são desconsiderados pois considera-se que já estão adequados à máscara.

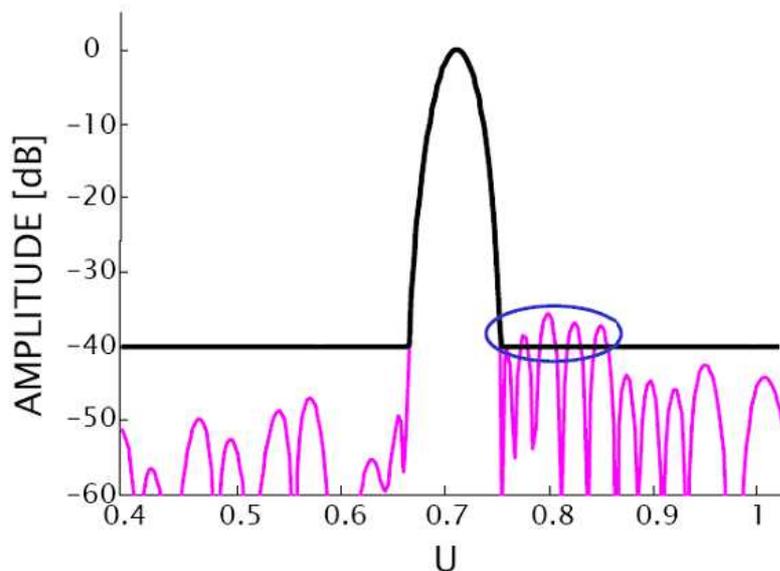


Fig. 4.6 – Exemplificação do método das máscaras

O valor do *fitness* será proporcional ao nível de adequação da antena calculada a máscara atingindo o valor máximo n quando o diagrama de radiação da antena calculada estiver totalmente adequado à máscara.

A plataforma computacional utilizará esse método para o *fitness* quando o parâmetro *tipo_fitness* receber o valor 2. Para antenas de fio, a quantidade de pontos n será sempre 180 (diagrama de radiação polar, com cada ponto representando 2 graus). Para antenas refletoras, a quantidade de pontos n dependerá da quantidade de pontos configurada na *string* WebPRAC.

Por fim, a equação (4-4) mostra uma variação da equação (4-3), em que se retira os valores em dB do diagrama de radiação além de normalizá-lo. Efetivamente é esta equação que está implementada na plataforma computacional.

$$fitness = n - \sum_{i=1}^k \frac{\log_{10}(-ant_des(i)) - \log_{10}(-ant_calc(i))}{\log_{10}(-ant_des(i))} \quad (4-4)$$

4.7.6. Método de Seleção

É utilizado o método da roleta, descrito no Capítulo 3.3.1, como método de seleção. A sua implementação está mostrada na Fig. 4.7. O vetor *array_fit* armazena o resultado da avaliação da função *fitness* para cada um dos indivíduos da geração atual. A variável *totalFit* armazena o somatório de todos os *fitness*.

```

%preparação da roleta
if (totalFit==0) then
    for j=1:pop,slice(j)=1.0/pop; end
else
    for j=1:pop,slice(j)=array_fit(j)/totalFit;end
end
marcas(1)=slice(1);
for j=2:pop,marcas(j)=marcas(j-1)+slice(j); end

```

Fig. 4.7 – Código de seleção pelo método da roleta

Como pode ser visto no código, o vetor *slice* armazena a fatia a qual aquele indivíduo terá direito no método da roleta. O vetor *marcas* é utilizado como apoio, sendo a frequência relativa acumulada do vetor *slice*.

Em seguida, a função *ga_selecionapai* é executada para selecionar um pai para a próxima geração, Fig. 4.8.

```
function pai=ga_selecionapai(marcas,pop);
    roleta=rand();
    for i=1:pop,
        if roleta<marcas(i)
            pai=i;
            break;
        end
    end
end
```

Fig. 4.8 – Código de seleção do pai

4.7.7. Validação do Algoritmo Genético para Antenas de Fio

Em um primeiro momento, desejou-se testar o algoritmo para validar se ele era capaz de realizar otimizações já conhecidas. O GRADMAX tem como antena padrão a Cardióide, que é uma antena que tem o diagrama de radiação com um formato de um coração em uma das direções. Essa antena é construída com dois fios. O primeiro é alimentado em 1 V e 0 graus, o segundo foi alimentado em 3,06 V e -54,66 graus de fase. A posição de cada um dos fios está listada na Tabela 4-2 e todos esses dados podem ser obtidos no GRADMAX. A Fig. 4.9 mostra esse diagrama nos planos XY e XZ, ambos limitados em -30 dB.

Tabela 4-2 – Localização dos fios e da alimentação da Cardióide

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,0m	-0,25m - 0,25m	1,0	0,0
Fio 2	0,25m – 0,25m	0,0 – 0,0m	-0,25m - 0,25m	3,06	-54,66

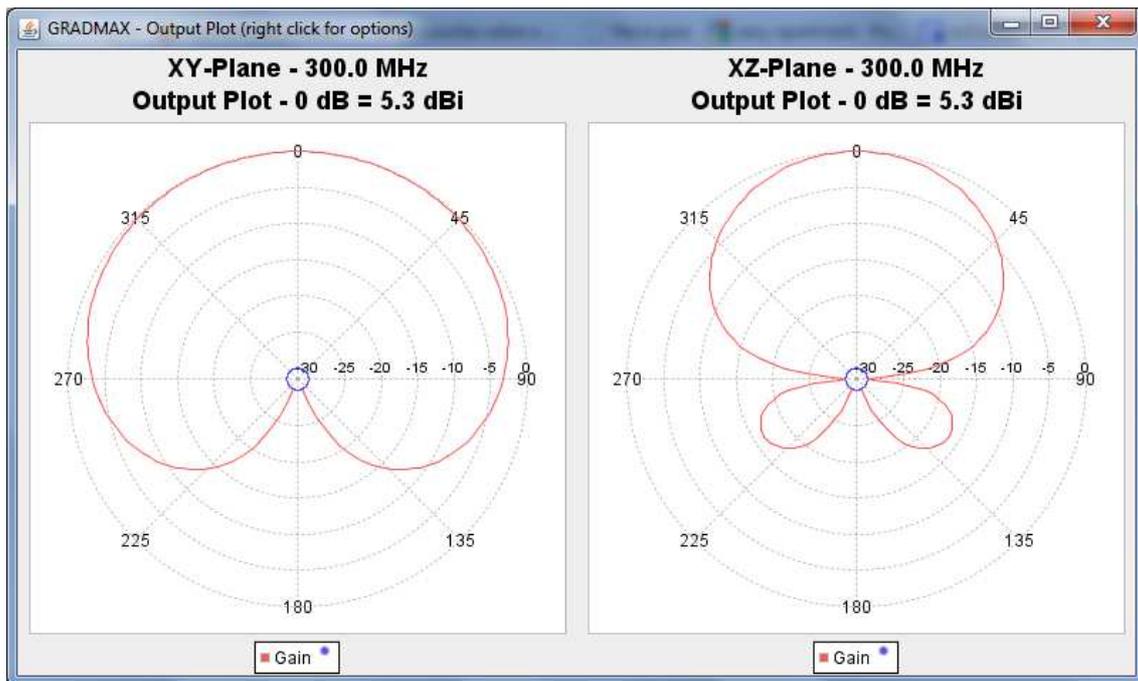


Fig. 4.9 – Diagrama de radiação calculado pelo GRADMAX de uma Cardióide

Para validação do algoritmo e verificar se ele era capaz de encontrar a Cardióide, fez-se o seguinte:

1. Definiu-se o diagrama da Cardióide como antena desejada utilizando os pontos do diagrama de radiação no plano XZ e os aplicando na variável *antena_desejada*, que é utilizada pela função *fitness*.
2. Definiram-se os parâmetros do Algoritmo Genético:
 - a. Probabilidade de *crossover*: 30%
 - b. População inicial: 30 antenas
 - c. Probabilidade de mutação: 1%
 - d. Número máximo de gerações: 10000;
3. Definiram-se os parâmetros da Plataforma Computacional:
 - a. Critério de parada: 99% do valor máximo do *fitness*
 - b. Forma de cálculo do *fitness*: cálculo pelo erro quadrático médio. $Tipo_fitness = 1$.
 - c. Divisor K: 100
 - d. Tensão máxima para os genes que representam os alimentadores: 10 V

A escolha dos valores de mutação e crossover foram baseados em resultados apresentados em [32]. Após a definição dos parâmetros, disponíveis no arquivo `exemple01_cardioid_tf1.m` da plataforma, rodou-se o algoritmo genético e após a geração 327 foi encontrado um indivíduo com o valor de fitness em 179,13 de um máximo de 180 (caso em que há total coincidência entre todos os pontos do diagrama desejado e o calculado). O resultado listado na Tabela 4-3 mostra os parâmetros encontrados para esse caso para a alimentação dos fios.

Tabela 4-3 - Resultado do algoritmo genético do melhor *fitness*

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,0m	-0,25m - 0,25m	1,0	0,0
Fio 2	0,25m – 0,25m	0,0 – 0,0m	-0,25m - 0,25m	3,034	-51,434

Como pode ser verificado, o resultado obtido é muito próximo do esperado. A alimentação desejada para o segundo fio era de 3,06 V e o algoritmo calculou 3,03 V. Para a fase, desejava-se -54,66 graus e encontrou-se -51,43 graus. A Fig. 4.10 mostra o gráfico desses dois resultados em uma mesma imagem. Como o resultado é bem próximo, os gráficos praticamente coincidem em todos os pontos. Em azul, tem-se a Cardióide utilizada como referência e, em vermelho, tem-se o diagrama calculado pelo algoritmo de otimização, ambos no plano XZ, limitando em -30 dB.

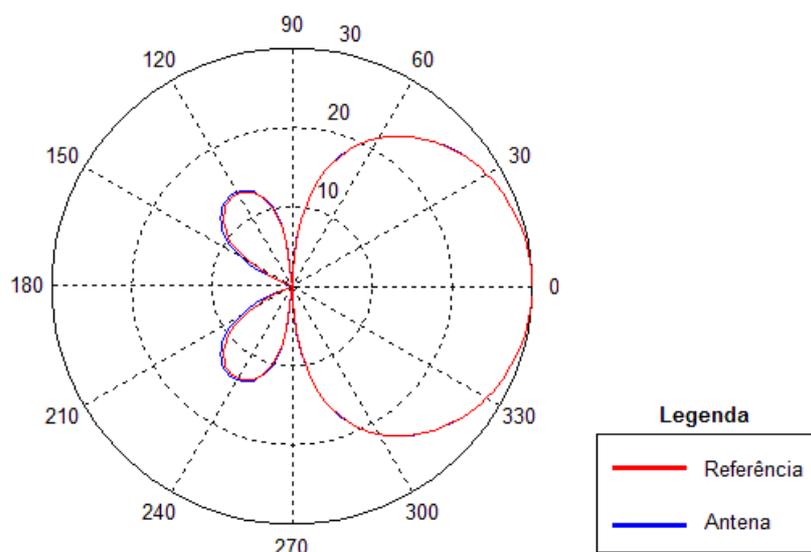


Fig. 4.10 – Gráficos da Cardióide utilizada como referência e a calculada pelo algoritmo

A Fig. 4.11 mostra o diagrama de radiação dessa antena calculada. Como se pode notar, é exatamente o diagrama da Cardióide. Houve diferença apenas no ganho, que nesse caso foi de $5,24\text{ dBi}$ contra $5,30\text{ dBi}$ da antena anterior.

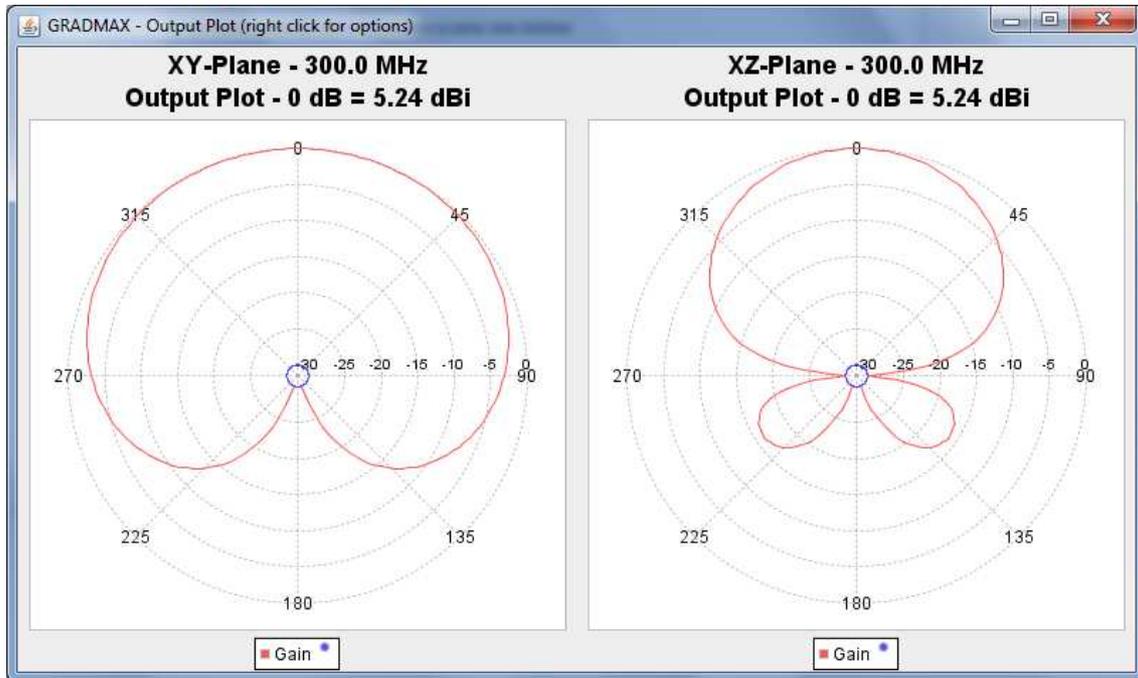


Fig. 4.11 – Diagrama de radiação calculado pelo GRADMAX da antena encontrada pelo algoritmo

Esse exemplo permitiu validar o funcionamento do GA que, em meio a tantas possibilidades, conseguiu encontrar o exato diagrama de radiação da Cardióide confirmando a correta implementação do algoritmo.

Outra informação interessante para avaliar a otimização é a evolução média do *fitness* geração após geração. A Fig. 4.12 mostra um gráfico com essa informação. Como pode ser visto, nas primeiras gerações há um salto muito grande da assertividade do *fitness*, até que em um determinado momento o valor médio do *fitness* se estabiliza variando próximo de um valor.

4.7.8. Validação do Algoritmo Genético para Antenas Refletoras

Da mesma forma, desejou-se validar se o algoritmo era capaz de realizar otimizações de antenas refletoras. O WebPRAC possui como antena padrão um refletor

de 4m que tem o diagrama de radiação mostrado pela Fig. 2.6. Modificou-se a posição do alimentador desse refletor, colocando $X_f = 1,3$; $Y_f = 1,7$ e $Z_f = 4,3$, resultando no diagrama de radiação mostrado pela Fig. 4.13.

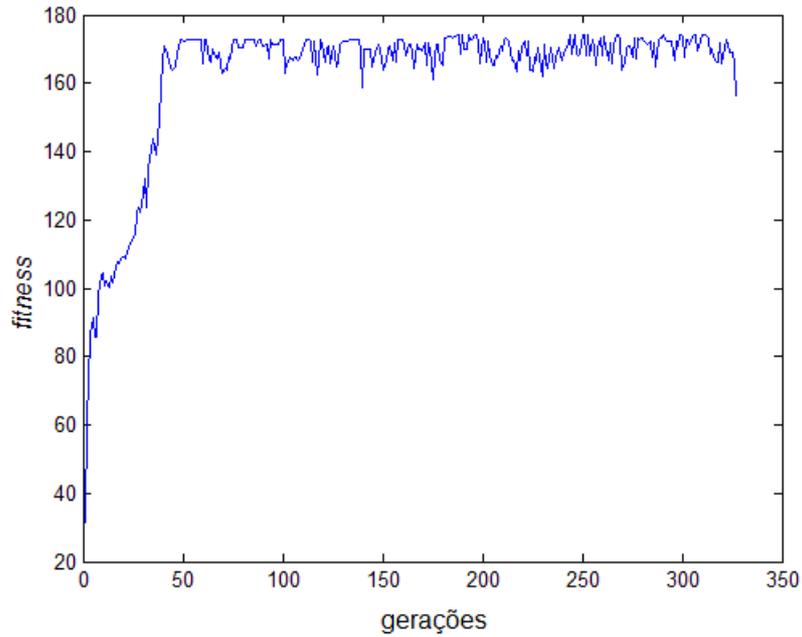


Fig. 4.12 – Evolução do *fitness* na otimização da Cardióide

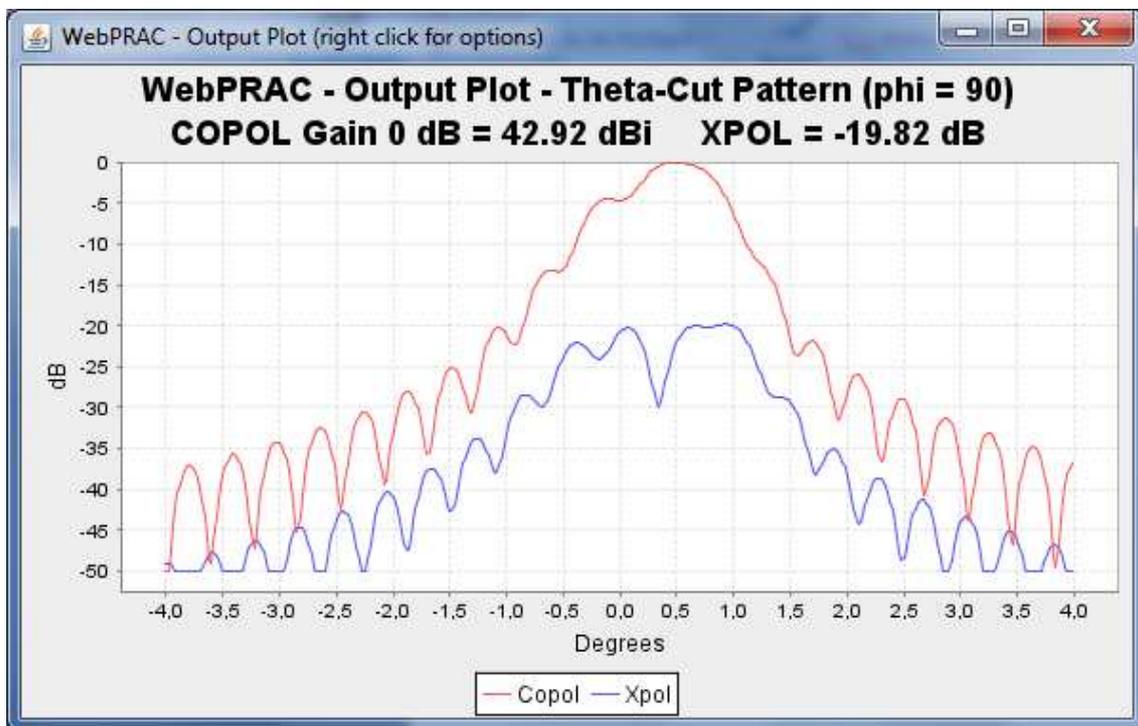


Fig. 4.13 – Diagrama de radiação de uma refletora com variação da posição do alimentador

Para validação do algoritmo e verificar se ele era capaz de encontrar a antena refletora da Fig. 4.13, fez-se o seguinte:

1. Definiu-se o diagrama da Refletora como antena desejada utilizando os pontos do diagrama de radiação no plano COPOL e os aplicando na variável *antena_desejada*, que é utilizada pela função *fitness*.
2. Definiram-se os parâmetros do Algoritmo Genético:
 - a. Probabilidade de *crossover*: 30%
 - b. População inicial: 30 antenas
 - c. Probabilidade de mutação: 3%
 - d. Número máximo de gerações: 10000;
3. Definiram-se os parâmetros da Plataforma Computacional
 - a. Critério de parada: 99% do valor máximo do *fitness*
 - b. Forma de cálculo do *fitness*: cálculo pelo erro quadrático médio.
Tipo_fitness = 1.
 - c. Divisor K: 100
 - d. Variação máxima para o posicionamento dos alimentadores: 2 m

Da mesma forma que na otimização da antena de fio, a escolha dos valores de *crossover* e mutação foram baseados em [32]. Após a definição dos parâmetros, disponíveis no arquivo *exemple02_reflector400m_tf1.m* da plataforma, rodou-se o algoritmo genético, e após a geração 151 foi encontrado um indivíduo com o valor de *fitness* em 396,61 de um máximo de 400 (caso em que há total coincidência entre todos os pontos do diagrama desejado e o calculado, lembrando que para o refletor padrão do WebPRAC são retornados 400 pontos para o diagrama de radiação).

O resultado para as posições do alimentador foi: $X_f = 1,51$; $Y_f = 1,61$; $Z_f = 4,48$. Os valores ficaram bem próximos da antena base. O diagrama de radiação das duas antenas sobreposto pode ser visto na Fig. 4.14. Nele, é possível verificar o diagrama COPOL e XPOL da antena base, em azul, e o diagrama de radiação COPOL da antena calculada, em verde.

A Fig. 4.15 mostra a evolução do *fitness* para essa otimização. É possível verificar que o *fitness* médio evolui gradualmente até a geração 151, onde o critério de parada é atingido e a otimização finalizada.

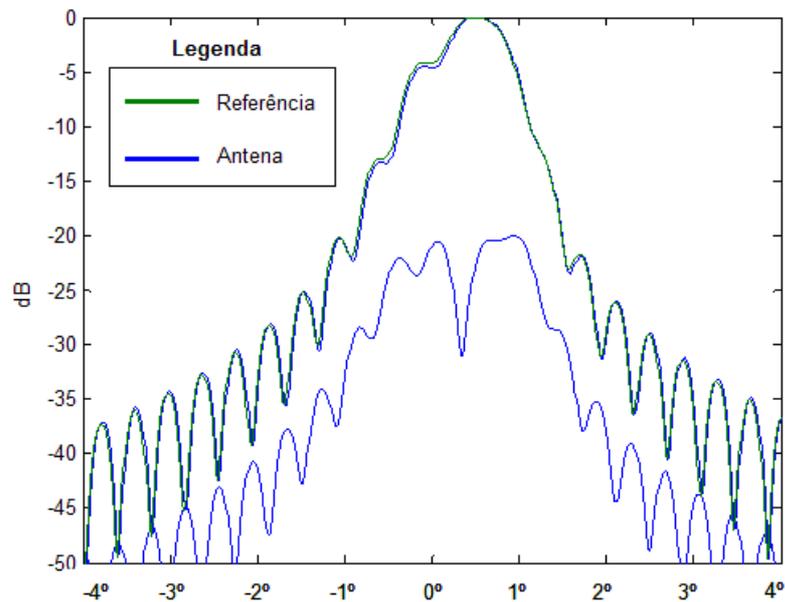


Fig. 4.14 – Diagrama de radiação da antena refletora base e a antena calculada

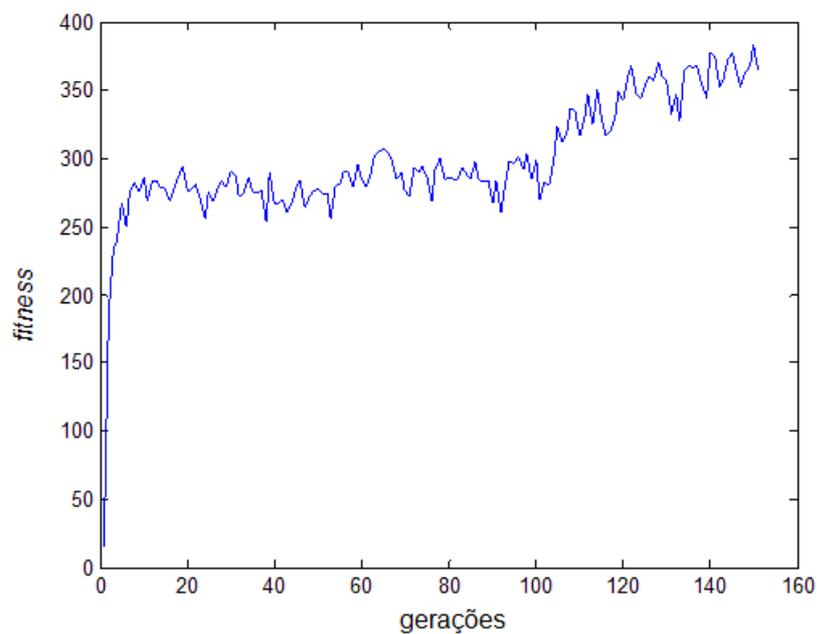


Fig. 4.15 – Evolução do *fitness* na otimização da refletora

4.7.9. Validação do Algoritmo Genético para Antenas de Fio com o Método das Máscaras

Nas duas seções anteriores foram otimizadas antenas de fio e refletoras utilizando o erro quadrático médio como base para o cálculo do *fitness*. Nessa seção, deseja-se validar o método das máscaras, introduzido na seção 4.7.5 que pode funcionar como alternativa no projeto de antenas. Por simplicidade, a Cardióide da seção 4.7.7 será utilizada nessa validação.

A máscara utilizada pelo algoritmo pode ser projetada utilizando um aplicativo de planilha eletrônica comum, como o *Excel*. Como nesse caso já se conhece o diagrama de radiação da antena desejada, torna-se simplificado o projeto da máscara, pois bastou utilizar o diagrama de radiação da antena desejada atribuindo para os pontos da máscara valores um pouco acima dos valores do diagrama da Cardióide. Com isso, chegou-se à máscara apresentada na Fig. 4.16, lembrando que essa máscara foi desenhada para a Cardióide utilizando como referência o plano XZ.

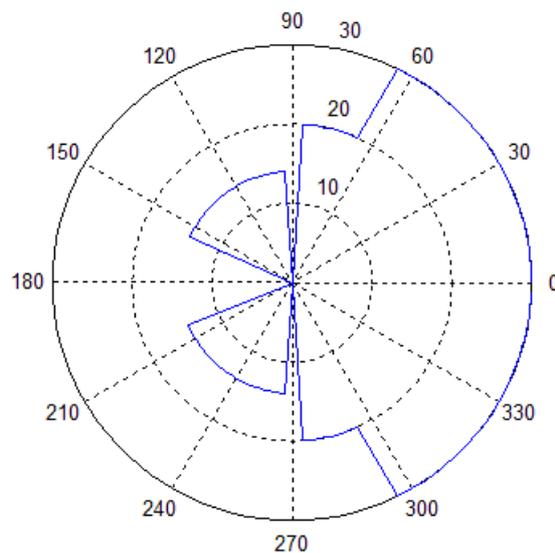


Fig. 4.16 – Máscara utilizada na otimização da Cardióide

Seguindo o processo de otimização, definiram-se os seguintes parâmetros para o Algoritmo Genético:

- a. Probabilidade de *crossover*: 30%

- b. População inicial: 30 antenas
- c. Probabilidade de mutação: 3%
- d. Número máximo de gerações: 10000;

E os seguintes parâmetros da Plataforma Computacional:

- a. Critério de parada: 99% do valor máximo do *fitness*
- b. Forma de cálculo do *fitness*: método das máscaras. *Tipo_fitness* = 2.
- c. Tensão máxima para os genes que representam os alimentadores: 10 V

Após a definição dos parâmetros, disponíveis no arquivo *exemple03_cardioid_tf2.m* da plataforma, rodou-se o algoritmo genético e após a geração 26 foi encontrado um indivíduo com o valor de *fitness* em 179,69 de um máximo de 180 (caso em que o diagrama da antena está totalmente inserido dentro da máscara). O resultado listado na Tabela 4-4 mostra os parâmetros encontrados para esse caso para a alimentação dos fios.

Tabela 4-4 - Resultado do algoritmo genético do melhor *fitness* com uso de máscaras

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,0m	-0.25m - 0,25m	1,0	0,0
Fio 2	0,25m – 0,25m	0,0 – 0,0m	-0.25m - 0,25m	2,8047	-50,318

Como pode ser verificado, o resultado obtido não é tão próximo quanto ao resultado obtido no erro quadrático médio. Alimentação da Cardióide para o segundo fio era de 3,06 V e o algoritmo calculou 2,80 V. Para a fase, a Cardióide possuía -54,66 graus e encontrou-se -50,31 graus. Apesar de o resultado ter sido levemente diferente, o diagrama de radiação se mostrou muito satisfatório, pois esteve quase que totalmente inserido na máscara conforme pode ser visto na Fig. 4.17.

O método das máscaras adiciona uma grau maior de liberdade na otimização, permitindo que uma maior diversidade de soluções seja contemplada. Na maioria das situações, se deseja um projeto de antena que atenda certos limiares de radiação em determinadas direções, e não um diagrama que atenda a níveis exatos de radiação em

determinadas direções. Esse método se mostrou bastante satisfatório para isso. A Fig. 4.18 traz a evolução do valor do *fitness* médio de cada geração durante a otimização.

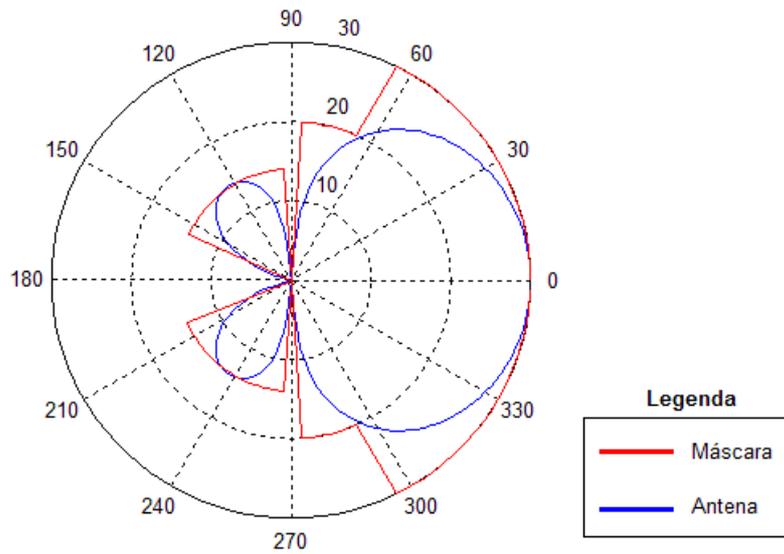


Fig. 4.17 – Máscara utilizada otimização e a antena Cardióide na mesma imagem

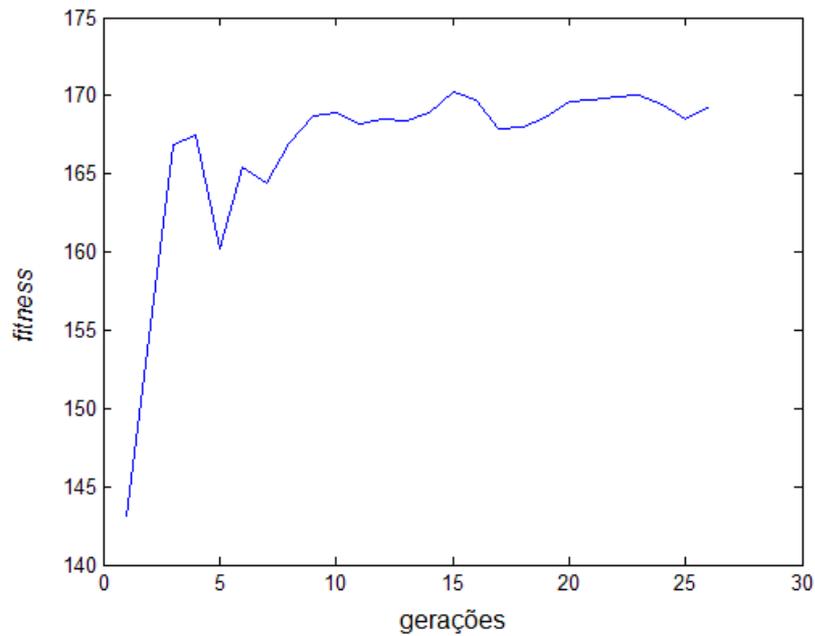


Fig. 4.18 – Evolução do *fitness* na otimização

5. SIMULAÇÕES

Conforme já exposto anteriormente, tradicionalmente as antenas são projetadas visando um conjunto de especificações para o seu desempenho individual. Nesse capítulo será proposto um cenário envolvendo métodos de modulação diferentes e como se pode sintetizar uma geometria de antena levando em consideração parâmetros de um sistema de comunicação, nesse caso a modulação. A simulação inicia-se propondo um cenário que é construído em conjunto com a ferramenta CML. Em seguida, é mostrado um método tradicional para o projeto de antenas nesse cenário específico, seção 5.2 e em continuação, é apresentado nas seções 5.3 e 5.4, como a plataforma computacional foi desenvolvida e utilizada para o projeto de antenas mais integradas a esse ao cenário proposto.

Destacado desse cenário, a seção 5.5 mostra como a plataforma também poderia ser utilizada para o projeto de uma antena refletora.

5.1. O CENÁRIO PROPOSTO

Considere um cenário no qual uma antena transmita com modulação QPSK em um enlace ponto a ponto. Considere também que essa antena possua o lóbulo principal e tenha dois lóbulos laterais interferindo em dois outros enlaces, sendo um 16-QAM e outro 64-QAM, como mostrado pela Fig. 5.1. Deseja-se analisar como uma antena poderia ser construída para que trabalhasse de forma eficiente dentro desse cenário.

Para isso, foi utilizada uma simulação simples do CML, construída pelo código *UncodedScenarios.m* disponível na ferramenta. Nele, são simulados cenários que não utilizam códigos de erro, pois se utiliza o esquema de modulação de forma pura, gerando um fluxo de bits aleatório e o transmitindo por um canal ruidoso AWGN (*Additive White Gaussian Noise*), cuja codificação está no arquivo *CmlChannel.m*. A interferência QPSK foi introduzida nesse arquivo por meio de um fluxo de bits modulados nesse esquema. As simulações foram então realizadas e os resultados obtidos são as curvas de erro de bit por E_b/N_0 mostradas nas Fig. 5.2 e Fig. 5.3 (foram

consideradas como interferências tanto o sinal QPSK como o ruído branco). Todos os enlaces operam na frequência de 300 MHz.

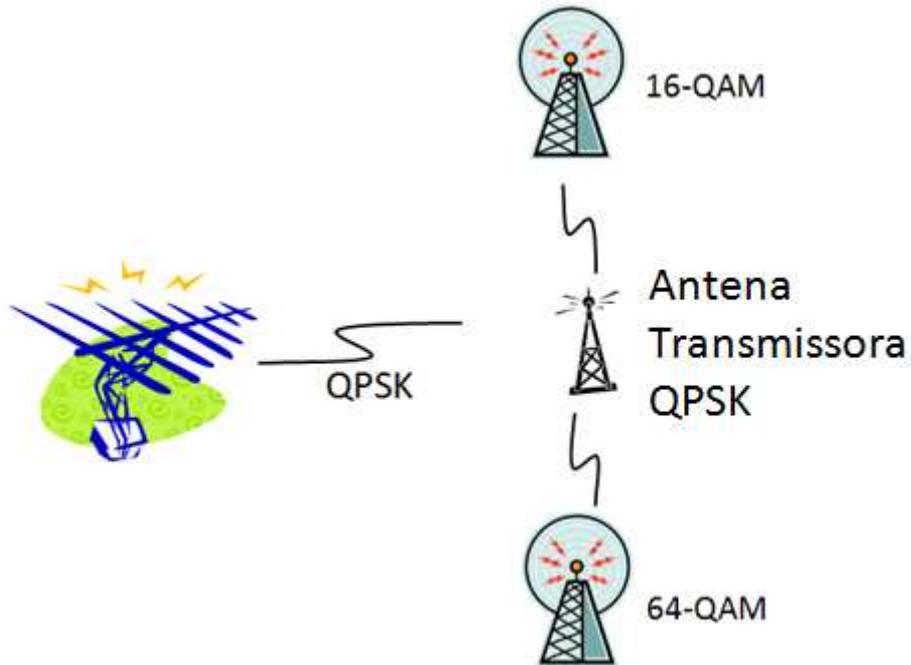


Fig. 5.1 – Ilustração com os enlaces propostos

Sem perda de generalidade, considere para este exemplo que o BER aceitável é de 10^{-3} . De acordo com a Fig. 5.3, foi possível verificar que a interferência máxima permitida no caso do 64-QAM seria de 18 dB. Já no caso do enlace 16-QAM, a interferência máxima seria de 11,5 dB, Fig. 5.2. Sendo assim, tem-se que as interferências máximas permitidas nos dois enlaces têm uma diferença de 6,5 dB, o que sugere que uma antena construída para esse cenário deveria ser assimétrica para atingir esse limite máximo de interferência.

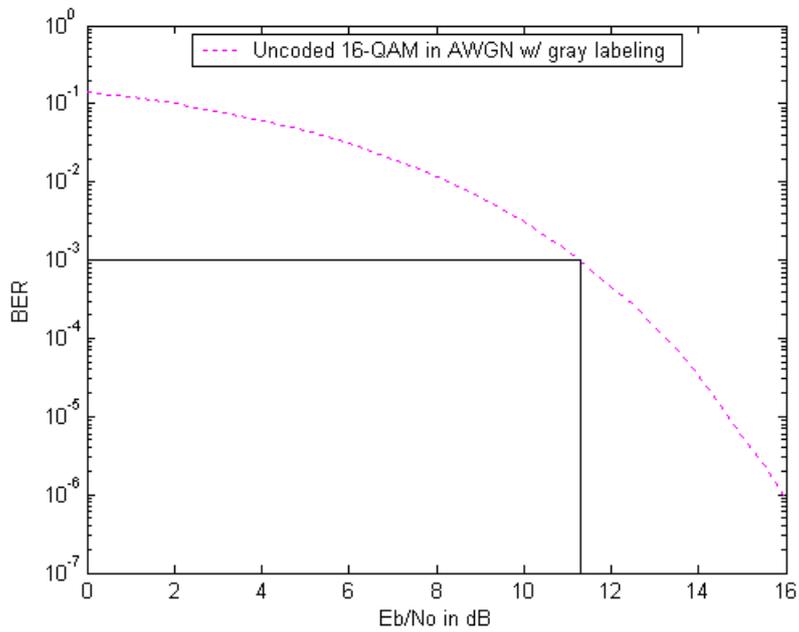


Fig. 5.2 - 16-QAM recebendo interferência de uma fonte QPSK e com ruído branco aditivo gaussiano

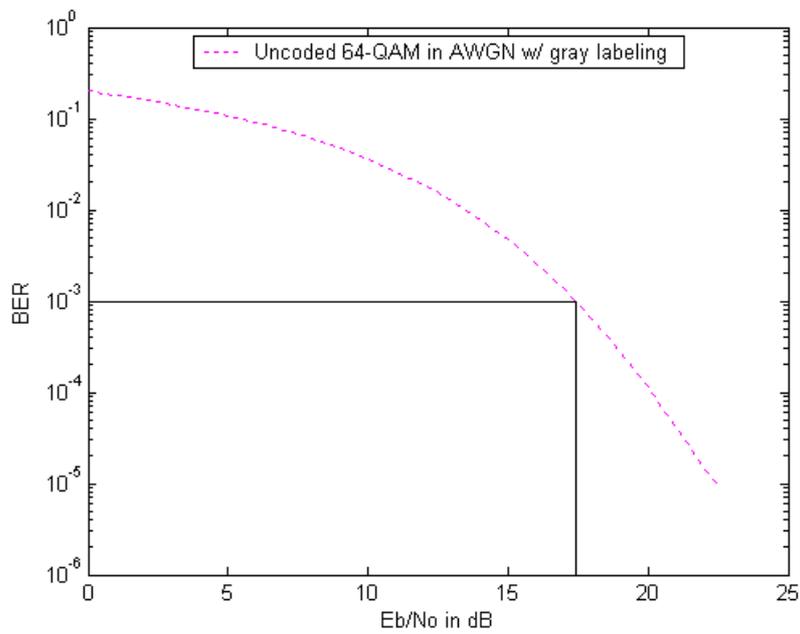


Fig. 5.3 – 64-QAM recebendo interferência de um lóbulo de uma transmissão QPSK e com ruído branco aditivo gaussiano

5.2. PROJETO TRADICIONAL DE UMA ANTENA PARA O CENÁRIO PROPOSTO

No cenário proposto na seção anterior, o enlace principal QPSK interfere em dois outros enlaces, um 16-QAM e outro 64-QAM. Em um projeto tradicional de antenas, geralmente tentaria se maximizar o ganho na direção do enlace principal minimizando os lóbulos laterais para reduzir a interferência em outros sistemas de comunicação. Existem diversos métodos que permitem realizar projetos de antenas com esse requisito e um deles, é conhecido como método de *Dolph-Chebyshev* que realiza a síntese de antenas utilizando conjuntos lineares de antenas [7].

O método baseia-se nas séries de *Fourier* definidas nas equações (5-1) e (5-2) e nos polinômios de *Chebyshev*, definidos pela equação (5-3). Nessas equações, $w = \cos\theta$, i_m são os elementos de corrente, d é o espaçamento entre os elementos do conjunto, N é a quantidade de elementos, λ é o comprimento de onda. A equação (5-1) é utilizada caso o número de elementos do conjunto linear for par e a equação (5-2) é utilizada caso o número de elementos do conjunto linear for ímpar.

$$f(w) = \sum_{m=-N}^N i_m e^{j2\pi m(d/\lambda)w} \quad (5-1)$$

$$f(w) = \sum_{m=1}^N \left(i_m e^{-j\pi(2m-1)(d/\lambda)w} + i_m e^{-j\pi(2m-1)(d/\lambda)w} \right) \quad (5-2)$$

$$T_n(x) = \begin{cases} (-1)^n \cosh(n \cosh^{-1}|x|) & x < -1 \\ \cos(n \cos^{-1} x) & -1 < x < 1 \\ \cosh(n \cosh^{-1} x) & x > 1 \end{cases} \quad (5-3)$$

O método não será detalhado, pois o objetivo é apenas mostrar um exemplo de como projetar uma antena para o cenário proposto.

Voltando ao cenário, tem-se que o pior caso de interferência do sinal QPSK seria no enlace 64-QAM, já que esse esquema de modulação é mais sensível. Sendo assim, o projeto tradicional deve levar isso em consideração. De acordo com a Fig. 5.3, é possível verificar que é necessário uma relação E_b/N de pelo menos 18 dB para que o sistema de comunicação opere com um BER de 10^{-3} . Com isso, os lóbulos laterais que interferem o sistema de comunicação em 64-QAM deveriam ser menores que 18 dB.

Em [49], tem-se um exemplo de um conjunto projetado com o método de *Dolph-Chebyshev* em que a diferença entre o lóbulo principal e o lateral é de -20 dB. O diagrama de radiação foi traçado pelo GRADMAX, e mostrado na Fig. 5.4 onde é possível ver que os lóbulos laterais estão no máximo em -20 dB no plano XZ. A localização e a alimentação dos fios dessa antena estão listados na Tabela 5-1.

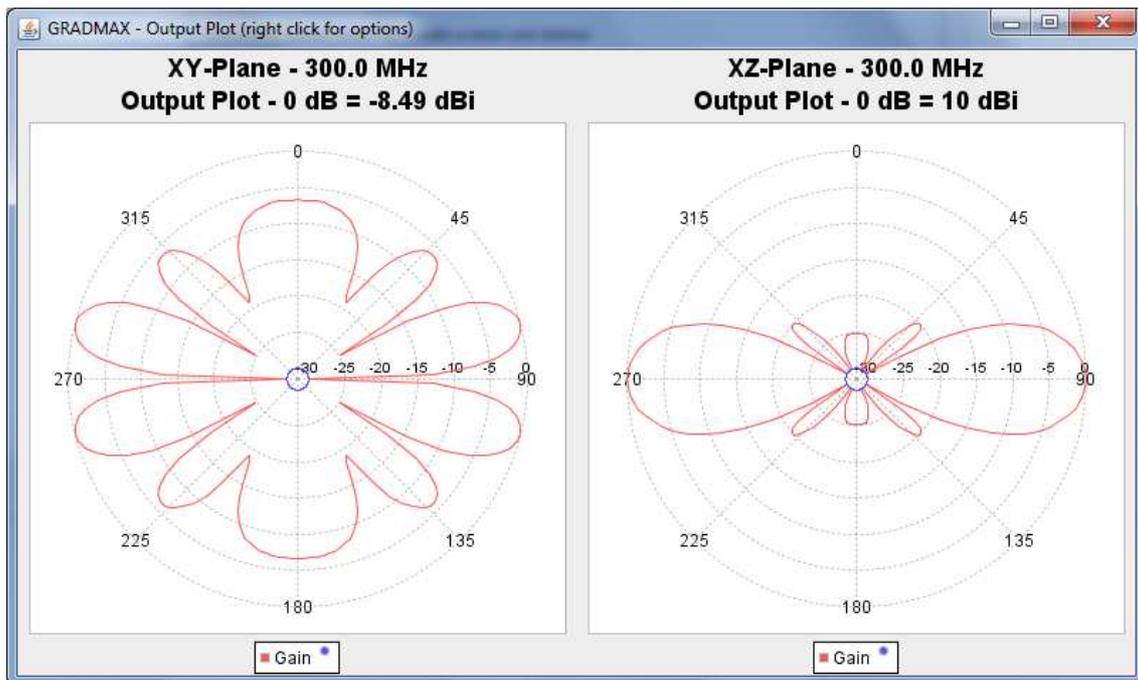


Fig. 5.4 – Diagrama de radiação da antena com – 20 dB no plano XZ projetada utilizando o método de *Dolph-Chebyshev*

Tabela 5-1 – Parâmetros da antena com – 20 dB projetada utilizando o método de *Dolph-Chebyshev*

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	-0,25 – 0,25m	0,0 – 0,0m	1,0	0,0
Fio 2	0,5 – 0,5m	-0,25 – 0,25m	0,0 – 0,0m	1,61	0,0
Fio 3	1,0 – 1,0m	-0,25 – 0,25m	0,0 – 0,0m	1,93	0,0
Fio 4	1,5 – 1,5m	-0,25 – 0,25m	0,0 – 0,0m	1,61	0,0
Fio 5	2,0 – 2,0m	-0,25 – 0,25m	0,0 – 0,0m	1,0	0,0

Essa antena atende exatamente ao requisito tradicional de projeto já que nesse caso, esse sistema de comunicação não interferirá os outros dois, pois o nível de radiação em ambas as direções não afetam as taxas de erro dos enlaces 16-QAM e 64-QAM. Em geral, esse método é utilizado para se projetar antenas para cenários como esse.

5.3. A BUSCA PELA GEOMETRIA DESEJADA COM ALGORITMOS GENÉTICOS

Aqui se sugere uma nova maneira de projetar antenas para o cenário proposto. Nesse modo, é admitido maior grau de liberdade no projeto da antena, deixando que se interfira os enlaces 16-QAM e 64-QAM até os limites em que o BER permaneça abaixo de 10^{-3} . Para realizar a síntese dessa antena assimétrica, pretende-se utilizar a plataforma computacional descrita no Capítulo 4.

5.3.1. Antena com Lóbulos Assimétricos

Primeiramente, buscou-se na literatura uma antena que pudesse representar uma assimetria próxima da desejada mesmo que a diferença entre o lóbulo principal e o secundário não fosse de 18 dB (enlace de 64-QAM). Nessa primeira etapa, o que importa é buscar antena com assimetria de 6,5 dB. Em [50], encontrou-se exemplos de conjuntos de antenas simples, com apenas dois elementos, que atenderiam parcialmente a esse requisito.

Os parâmetros dessa antena com lóbulos assimétricos estão listados na Tabela 5-2. Esses parâmetros foram inseridos no GRADMAX e o diagrama de radiação é mostrado pela Fig. 5.5. Como pode ser notado, há uma diferença entre os lóbulos do plano XZ em 0 e 180 graus que foi medida em 4,5 dB.

Tabela 5-2 - Localização dos fios e da alimentação da antenna assimétrica

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,5m	0,0 – 0,0m	1,0	0,0
Fio 2	0,75m – 0,75m	0,0 – 0,5m	0,0 – 0,0m	1,0	-45,0

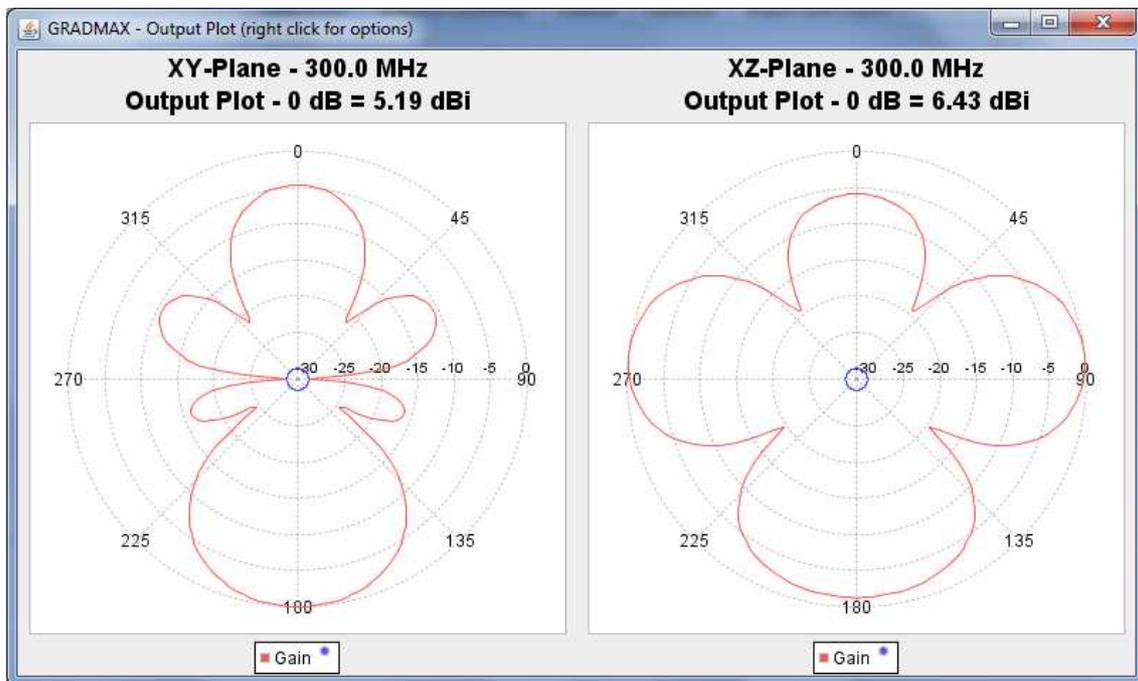


Fig. 5.5 – Diagrama de radiação calculado pelo GRADMAX da antenna assimétrica

Desejou-se avaliar se a plataforma computacional era capaz de encontrar o diagrama de radiação da Fig. 5.5, mesmo que os parâmetros de alimentação do fio fossem diferentes. Para isso, foi aplicada no GA a antenna base da Tabela 5-3, em que a tensão e a fase do fio 2 foram genes do GA.

Para o GA, o tamanho da população foi de 30 indivíduos, *crossover* em um ponto com probabilidade de 30%, probabilidade de mutação de 1%, função *fitness* definido pelo método do erro quadrático médio, equação (4-2), critério de parada em 99% do *fitness* máximo. Todos os parâmetros estão disponíveis no arquivo *exemple04_arrl_tf1.m* da plataforma computacional.

Tabela 5-3 – Localização dos fios e da alimentação da antena de referência para o GA

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,5m	0,0 – 0,0m	1,0	0,0
Fio 2	0,75m – 0,75m	0,0 – 0,5m	0,0 – 0,0m	Gene 1	Gene 2

Após 124 gerações, obtiveram-se os resultados da

Tabela 5-4 e o diagrama de radiação traçado na Fig. 5.6. O valor da função *fitness* calculado foi de 178,70 para um máximo de 180, ou seja, pode-se dizer que a antena encontrada pelo GA é bem próxima da antena de [50]. A Fig. 5.7 mostra a evolução do *fitness* médio da geração para esse cenário.

Tabela 5-4 – Resultado do algoritmo genético do melhor *fitness* para a antena assimétrica

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,5m	0,0 – 0,0m	1,0	0,0
Fio 2	0,75m – 0,75m	0,0 – 0,5m	0,0 – 0,0m	0,599	-40,262

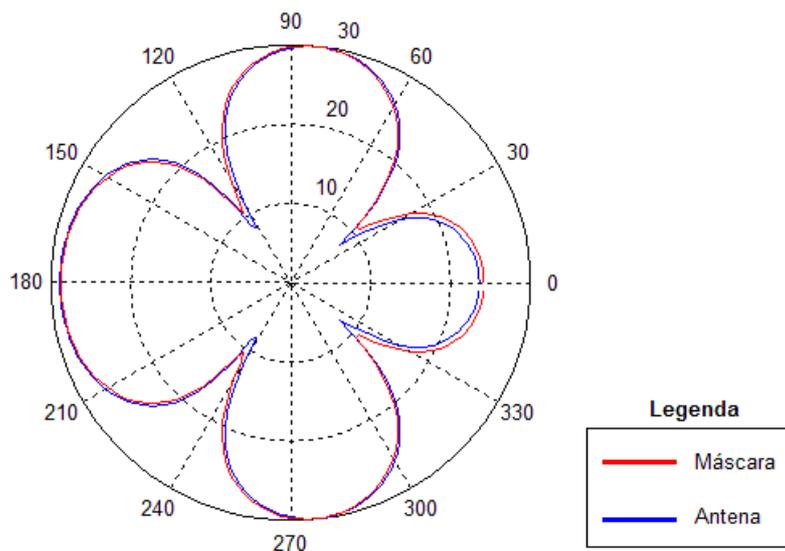


Fig. 5.6 –Diagrama de radiação, plano XZ, limitadas em -30dB dos diagramas calculados pelo algoritmo

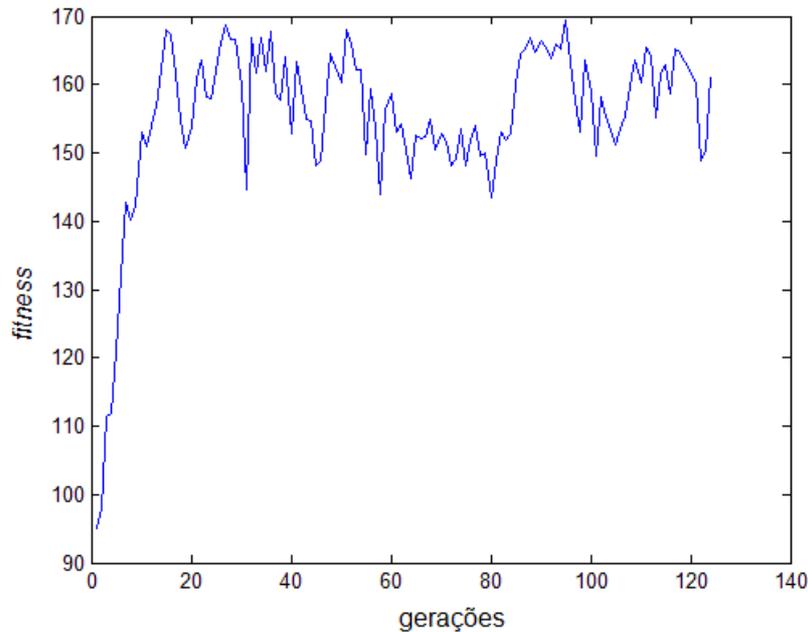


Fig. 5.7 – Evolução do *fitness* para a simulação

5.3.2. Antena com 6,5 dB de Assimetria

Apesar de se ter projetado uma antena com uma assimetria de aproximadamente 4,5 dB, o que se deseja é encontrar uma antena com 6,5 dB de assimetria para os enlaces 16-QAM e 64-QAM o que seria o limite máximo de interferência para um BER de 10^{-3} . Para isso, tentou-se alterar o parâmetro “*antena_desejada*”, utilizado na função *fitness* em (4-2), aumentando a assimetria da antena da Fig. 5.5 de 4,5 dB para 6,5 dB em uma planilha eletrônica. Com esse novo diagrama aplicado na função *fitness*, tentou-se rodar o GA com os mesmos 30 indivíduos na população, 30% de probabilidade de *crossover*, 1% de mutação. Após 5000 gerações, não se obteve sucesso em encontrar um diagrama com 6,5 dB de assimetria.

Aumentou-se então o número de elementos, o espaçamento entre os elementos, mas todas as tentativas não tiveram sucesso. Em um determinado momento, chegou-se a utilizar até 16 elementos, também sem sucesso.

Em outro momento, ao modificar a fase do segundo fio, verificou-se que a assimetria entre os lóbulos aumentava. Ao atingir a fase de -60,7035 graus, a assimetria entre os dois lóbulos era de exatamente 6,5 dB, Fig. 5.8, Tabela 5-5. Isso permitiu

chegar a conclusão que a forma de implementação do *fitness* estivesse prejudicando a otimização da antena.

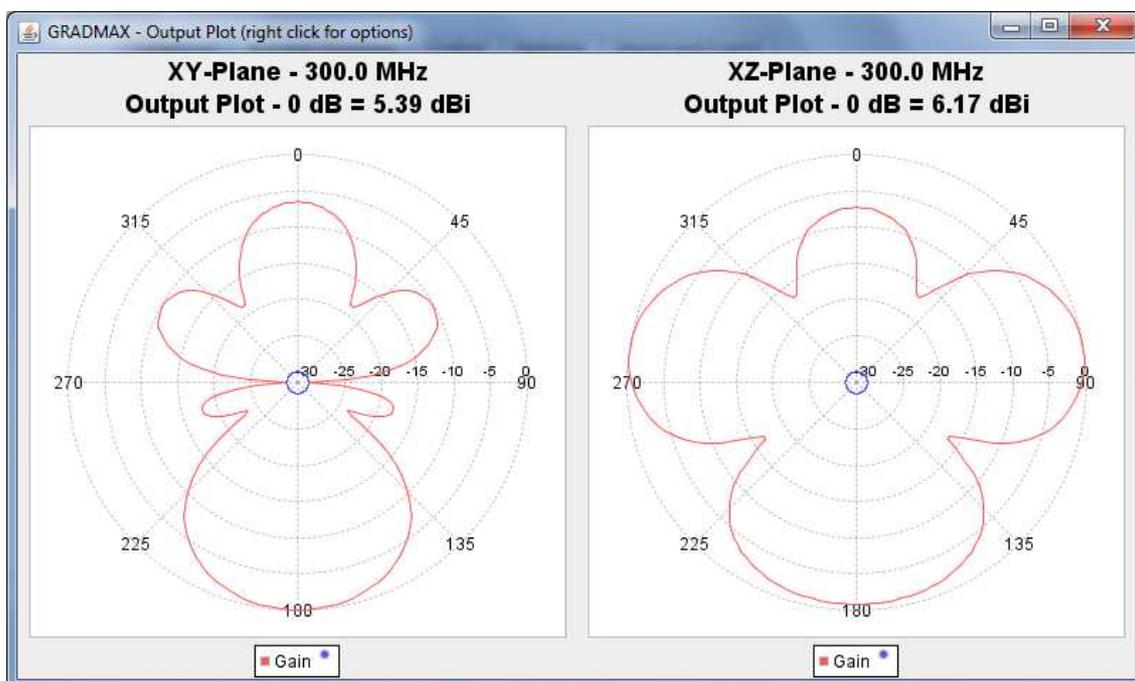


Fig. 5.8 - Antena com 6,5 dB, utilizando como base uma leve modificação na fase do fio 2

Tabela 5-5 – Antena com 6,5 dB, utilizando como base uma leve modificação na fase do fio 2

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,5m	0,0 – 0,0m	1,0	0,0
Fio 2	0,75m – 0,75m	0,0 – 0,5m	0,0 – 0,0m	1,0	-60,7035

Para confirmar a hipótese que o *fitness* estaria prejudicando, utilizou-se essa antena como “*antena_desejada*” e executou-se o GA da mesma forma realizada nas otimizações anteriores. Os parâmetros dessa simulação estão disponíveis no arquivo *exemple05_arrl65db_tf1.m*.

Após 34 gerações, obteve-se um valor de *fitness* de 178,69 de um máximo de 180. Sendo assim, pode-se dizer que a antena com 6,5 dB foi encontrada. Os parâmetros dessa nova antena estão relacionados na Tabela 5-6 e o seu diagrama de radiação está traçado na Fig. 5.9.

Tabela 5-6 – Resultado do algoritmo genético do melhor *fitness* para a antena de 6,5 dB

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,5m	0,0 – 0,0m	1,0	0,0
Fio 2	0,75m – 0,75m	0,0 – 0,5m	0,0 – 0,0m	0,902	-62,398

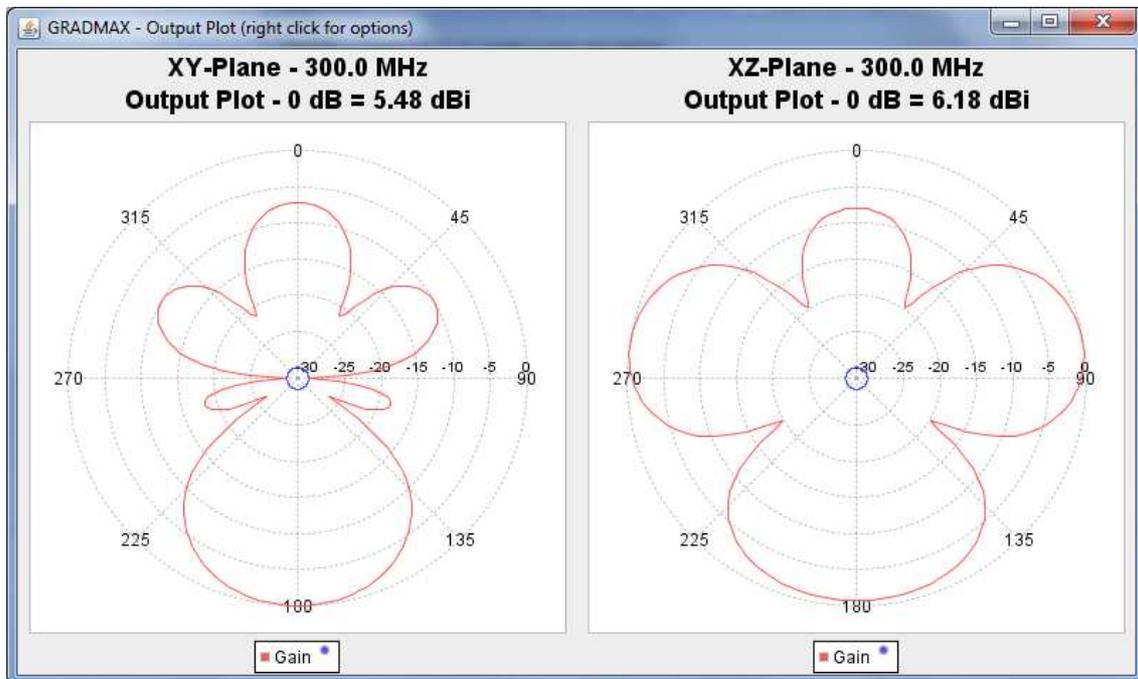


Fig. 5.9 – Diagrama de radiação calculado pelo GRADMAX da antena com assimetria de 6,5 dB

Esse fato permite realizar uma reflexão: só foi possível encontrar uma antena com assimetria de 6,5 dB depois de se ter como referência uma antena parecida. Ao que tudo indica, isso está relacionado como a forma que o *fitness* está pontuando cada uma das antenas de cada estágio do GA. Isso será analisado em seguida.

5.3.3. Outra Abordagem – Mudança na Função *Fitness*

Como mencionado no capítulo sobre GAs, Capítulo 3, o *fitness* é a parte que faz a ligação entre o problema a ser resolvido e a camada de otimização do GA. Na abordagem do item anterior, foi possível encontrar a antena com 6,5 dB de diferença literalmente fazendo uma comparação entre uma antena desejada e as antenas obtidas (indivíduos) pelo GA. Exigir que o algoritmo encontre exatamente o diagrama desejado pode fazer com que antenas que poderiam atender ao requisito de assimetria de 6,5 dB nos dois lóbulos possam ser descartados, já que o cálculo do *fitness* não as privilegia.

Sendo assim, faz-se necessário reescrever o *fitness* de forma que se encontre uma antena que atenda ao requisito de assimetria. Para isso, propõe-se então que o *fitness* seja calculado de acordo com a equação (5-4), em que *antena*(180) e *antena*(0) são os ganhos da antena a ser avaliada em 180 e 0 graus.

$$f = 180 - (\textit{antena}(180) - \textit{antena}(0) - 6,5)^2 \quad (5-4)$$

Feito isso, executou-se o GAs utilizando basicamente os mesmos parâmetros, ou seja, 30 indivíduos na população, 30% de probabilidade de *crossover*, 1% de mutação e os parâmetros da Tabela 5-3 como antena de referência. Após 5000 iterações, foi encontrada a antena da Fig. 5.10. A diferença entre os lóbulos assimétricos de 0 e 180 graus é de exatamente 6,5 dB.

A Tabela 5-7 mostra os parâmetros da antena da Fig. 5.10. Como pode ser notado, existe uma pequena diferença entre os diagramas obtidos pelas duas abordagens. Entretanto, nas duas abordagens a diferença entre os diagramas é de 6,5 dB. Esse resultado mostrou que reescrever o *fitness* permitiu que uma antena com a assimetria de 6,5 dB fosse utilizada sem que fosse necessário utilizar outra antena como referência o que trouxe um ganho para o algoritmo.

Tabela 5-7 – Resultado do algoritmo genético do melhor *fitness* para a antena de 6,5 dB para outra abordagem

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,5m	0,0 – 0,0m	1,0	0,0
Fio 2	0,75m – 0,75m	0,0 – 0,5m	0,0 – 0,0m	1,3993	-88.8134

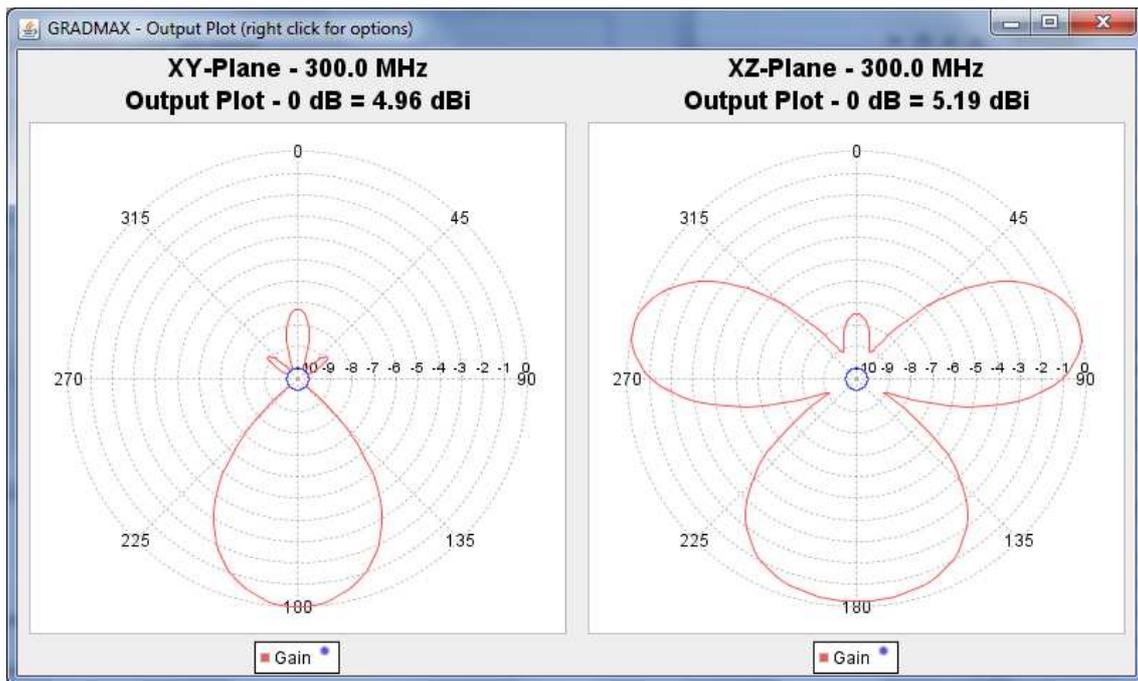


Fig. 5.10 – Diagrama de radiação calculado pelo GRADMAX da antena com assimetria de 6,5 dB com *fitness* baseado na diferença entre 0 e 180 graus

Para projetar a antena com todos os requisitos levantados, a função *fitness* terá que ser evoluída para que se encontre a antena com os exatos requisitos desejados. O que foi feito no item anterior foi simplesmente analisar a diferença entre os dois lóbulos laterais. Entretanto, não foi levada em consideração a diferença entre o lóbulo principal e os secundários.

5.4. A UTILIZAÇÃO DE MÁSCARAS NA OTIMIZAÇÃO

Para realizar a otimização de forma eficiente, implementou-se função *fitness* para analisar a adequação das antenas de acordo com máscaras, conforme a seção 4.7.5.

Assim, projetou-se a máscara em planilha eletrônica de acordo com a Fig. 5.11. Como pode-se verificar, a máscara está desenhada apenas duas regiões com pontos de ± 30 graus e entre 125 e 235 graus. Essas duas regiões possuem assimetria de 8 dB, aproximadamente. Os outros pontos da máscara foram preenchidos com o valor “-1000” para indicar ao algoritmo pontos em que a antena pode variar livremente.

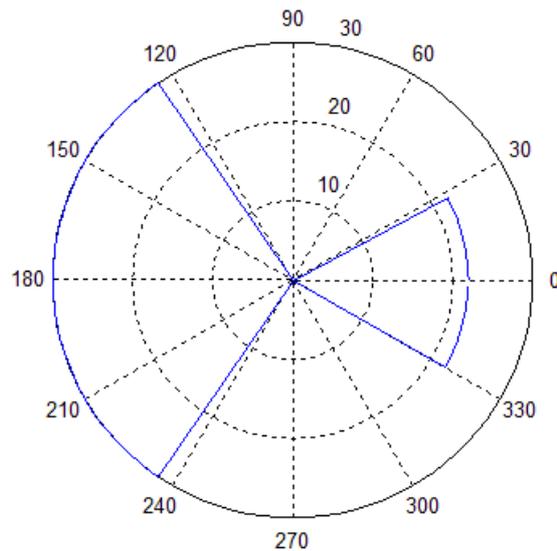


Fig. 5.11 – Máscara inserida para o projeto da antena assimétrica pelo algoritmo

De posse do vetor da máscara, que foi atribuído ao parâmetro *antena_desejada* do algoritmo genético, executou-se o algoritmo com as seguintes abaixo, que estão todas disponíveis no arquivo *exemplo06_arrl_ass_mask_tf2.m* da plataforma computacional:

- a. Probabilidade de *crossover*: 30%
- b. População inicial: 30 antenas
- c. Probabilidade de mutação: 3%
- d. Número máximo de gerações: 10000;

Definiu os parâmetros da Plataforma Computacional:

- a. Critério de parada: 99% do valor máximo do *fitness*
- b. Forma de cálculo do *fitness*: método das máscaras. *Tipo_fitness* = 2.
- c. Tensão máxima para os genes que representam os alimentadores: 10 V

Após 22 gerações, obteve-se os resultados da Tabela 5-8 e o diagrama de radiação traçado na Fig. 5.12. O valor da função *fitness* calculado foi de 87 para um máximo de 87, ou seja, a antena calculada estava completamente inserida dentro da máscara - para otimização por máscaras, pontos definidos abaixo de -1000 são

desconsiderados no cálculo. A Fig. 5.13 mostra a evolução do *fitness* médio de cada geração para esse cenário.

Tabela 5-8 – Resultado do algoritmo genético do melhor *fitness* para a antena assimétrica

	X (inicial – final)	Y (inicial – final)	Z (inicial – final)	Tensão (V)	Fase (Graus)
Fio 1	0,0 – 0,0m	0,0 – 0,5m	0,0 – 0,0m	1,0	0,0
Fio 2	0,75m – 0,75m	0,0 – 0,5m	0,0 – 0,0m	0,599	-40,262

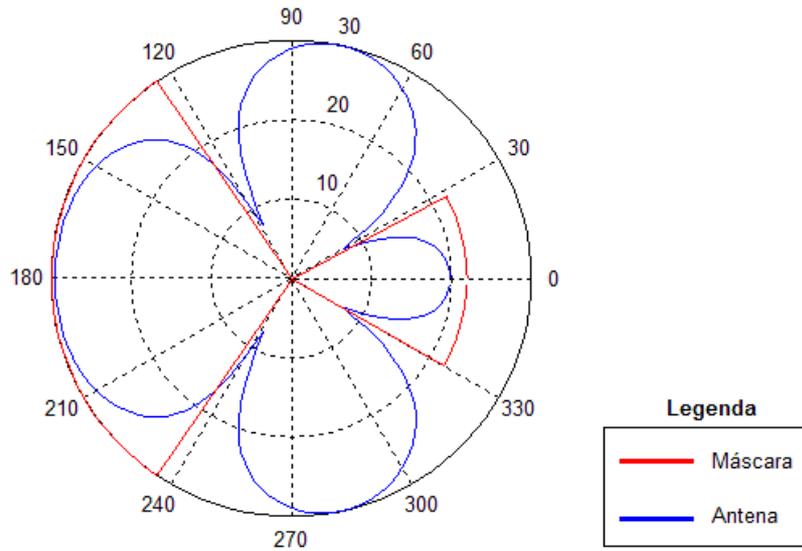


Fig. 5.12 –Diagrama de radiação, plano XZ, trunc -30dB dos diagramas calculados pelo algoritmo

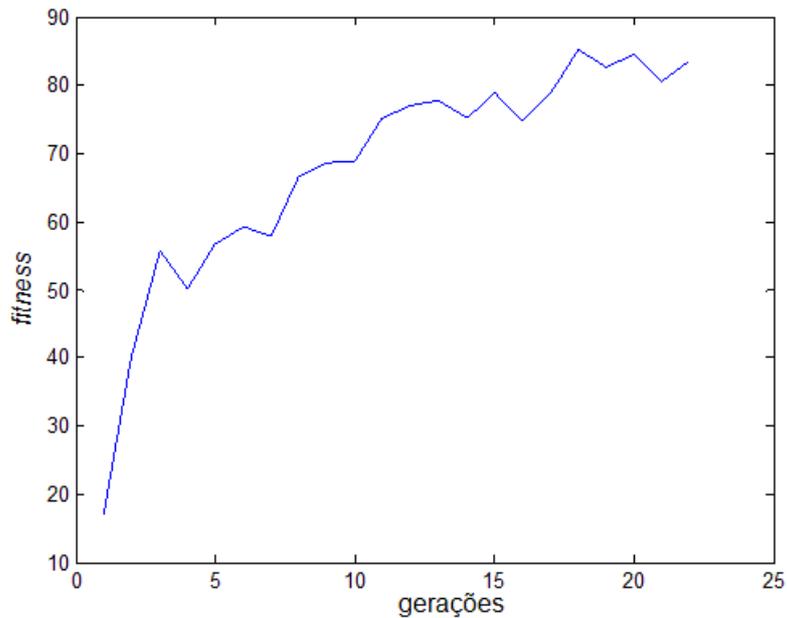


Fig. 5.13 - Evolução do *fitness* para a simulação

Como ser visto no resultado apresentado da Fig. 5.12, a antena otimizada atende aos requisitos da máscara, pois todos os pontos do diagrama estão abaixo do limiar da máscara – lembrando que a região não contemplada pela máscara não é considerada na otimização (pontos da máscara ≤ -1000 dB). Apesar de atender aos requisitos da máscara, é possível notar que a assimetria é de aproximadamente 10 dB o que não atende ao requisito desejado de assimetria de 6,5 dB.

5.4.1. O uso de Máscaras e o Método do Erro Quadrático Médio

Foi mostrado que cada um dos métodos de cálculo do fitness tem suas vantagens. Ao aliar ambos, chega-se ao modelo ideal. Nesse próximo exemplo, foi utilizado o método das máscaras concomitantemente ao método do erro quadrático médio para avaliar a diferença de assimetria entre o nível de radiação em 0 graus e 180 graus.

O método apresentado foi simples: foi adicionado ao cálculo do *fitness* uma penalização calculada pela equação (5-4), que tem como objetivo garantir que a diferença da assimetria esteja o mais próximo possível de 6,5 dB.

Executou-se o algoritmo com os parâmetros abaixo, disponíveis no arquivo *exemple07_arrl_ass_mask_tf2_and_diff.m* da plataforma computacional.:

- a. Probabilidade de *crossover*: 30%
- b. População inicial: 30 antenas
- c. Probabilidade de mutação: 3%
- d. Número máximo de gerações: 10000;

Definiu os parâmetros da Plataforma Computacional:

- a. Critério de parada: 99,5% do valor máximo do *fitness*
- b. Forma de cálculo do *fitness*: método das máscaras aliado ao método do erro quadrático médio. *Tipo_fitness* = 3.
- c. Tensão máxima para os genes que representam os alimentadores: 10 V

Após 1139 gerações, o resultado foi o da Fig. 5.14. Nesse caso, a diferença entre os lóbulos ficou em 6,71 dB o que é bem próximo do desejado. O valor do *fitness* nesse caso ficou em 86,63 de um total de 87. A Fig. 5.15 mostra a evolução do *fitness* médio para cada geração nessa situação.

Nota-se que o diagrama obtido pode ser ainda melhorado para o uso prático, como por exemplo, reduzindo-se um dos lóbulos principais que no presente cenário é sobressalente. No entanto, é importante indicar que o exemplo teve propósitos ilustrativos, e que sem perda de generalidade, demonstrou a possibilidade de síntese de antenas melhor integradas a parâmetros normalmente não levados em consideração no projeto de antenas em sistemas de comunicações, nesse caso específico as modulações digitais.

Esse resultado permitiu mostrar a flexibilidade existente na plataforma computacional no projeto de antenas integrando-as efetivamente a sistemas de comunicações sem fio modernos.

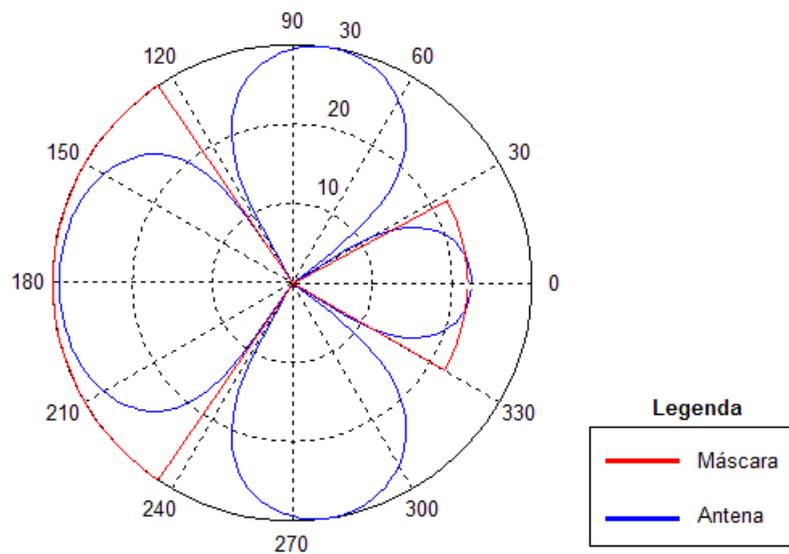


Fig. 5.14 – Resultado obtido pela utilização concomitante dos dois métodos

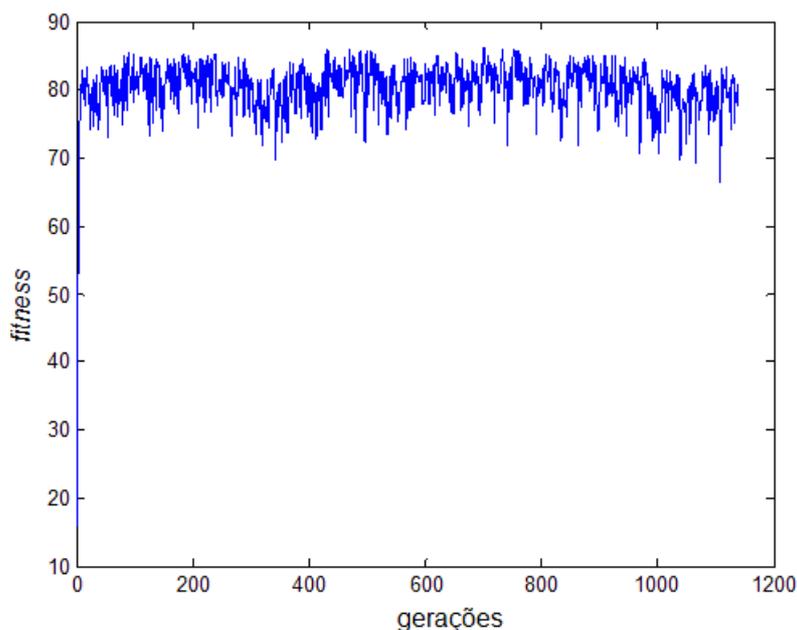


Fig. 5.15 – Evolução do *fitness* médio de cada geração para essa simulação

5.5. OTIMIZAÇÃO DE ANTENAS REFLETORAS

Por fim, deseja-se mostrar uma forma de otimizar antenas refletoras efetivamente em sistemas de comunicações sem fio por meio da plataforma. O método é exatamente o mesmo mostrado para as antenas de fio, podendo-se utilizar o método do erro quadrático médio, o método das máscaras ou ambos ao mesmo tempo.

Para isso, considere que, em um sistema de comunicação sem fio qualquer, deseja-se integrar uma antena refletora que possua assimetria dos lóbulos laterais de 4,5 dB. Essa antena é conhecida e é obtida colocando posicionando o alimentador em $X_f = 0$; $Y_f = 1$ e $Z_f = 0$ para a antena padrão do WebPRAC, Fig. 5.16. Nela, há uma assimetria entre o lóbulo lateral esquerdo e o lóbulo lateral direito de aproximadamente 4,5 dB. Suponha que essa antena não seja conhecida e que simplesmente, seja desenhada uma máscara, como a da Fig. 5.17. Considere que essa antena esteja operando em 11,25 GHz.

De posse da máscara, executou-se o algoritmo com os seguintes parâmetros, disponíveis no arquivo *exemple08_reflector400m_tf2.m* da plataforma computacional.

- a. Probabilidade de *crossover*: 30%
- b. População inicial: 30 antenas
- c. Probabilidade de mutação: 3%
- d. Número máximo de gerações: 100000;

Definiu os parâmetros da Plataforma Computacional:

- a. Critério de parada: 100% do valor máximo do *fitness*
- b. Forma de cálculo do *fitness*: método das máscaras. *Tipo_fitness* = 2.
- c. Variação máxima do posicionamento dos alimentadores: 2 m

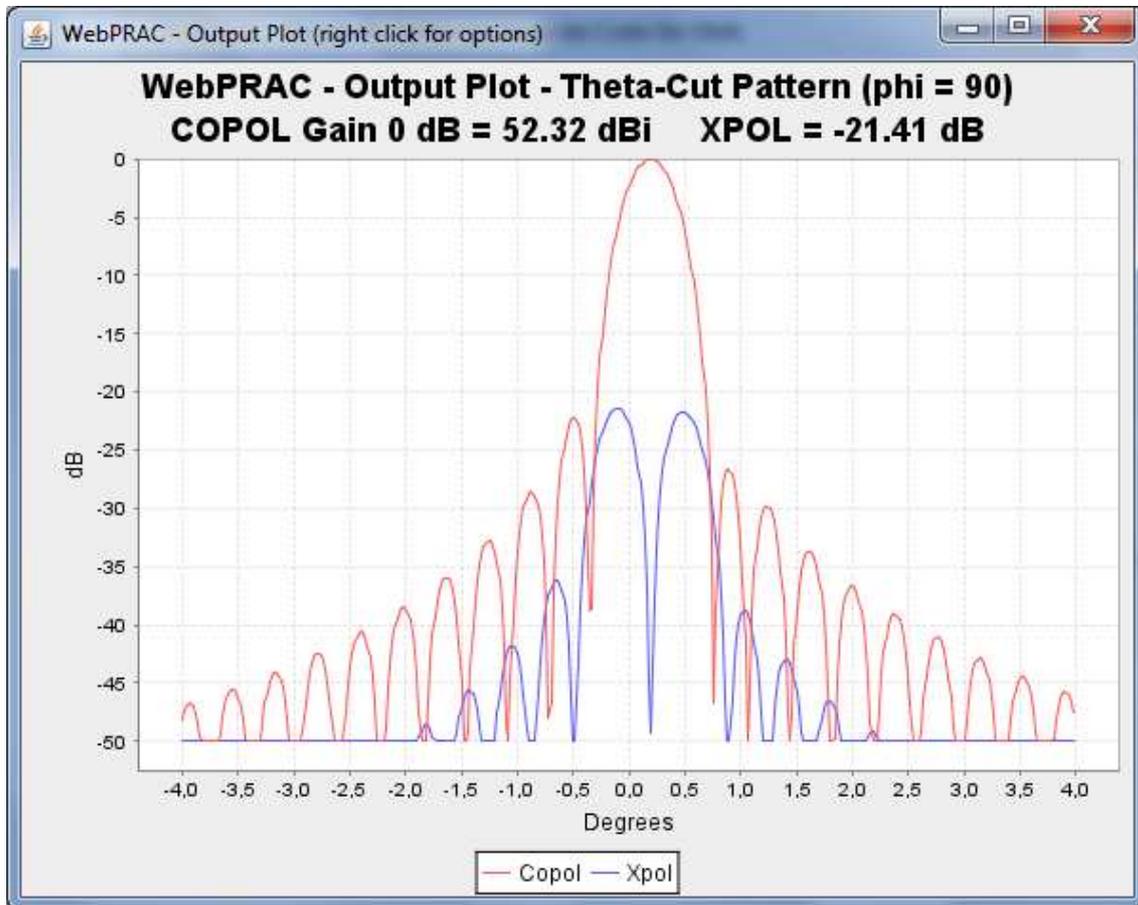


Fig. 5.16 – Antena padrão do WebPRAC com alimentadores em $X_f=0$; $Y_f=0$ e $Z_f=0$

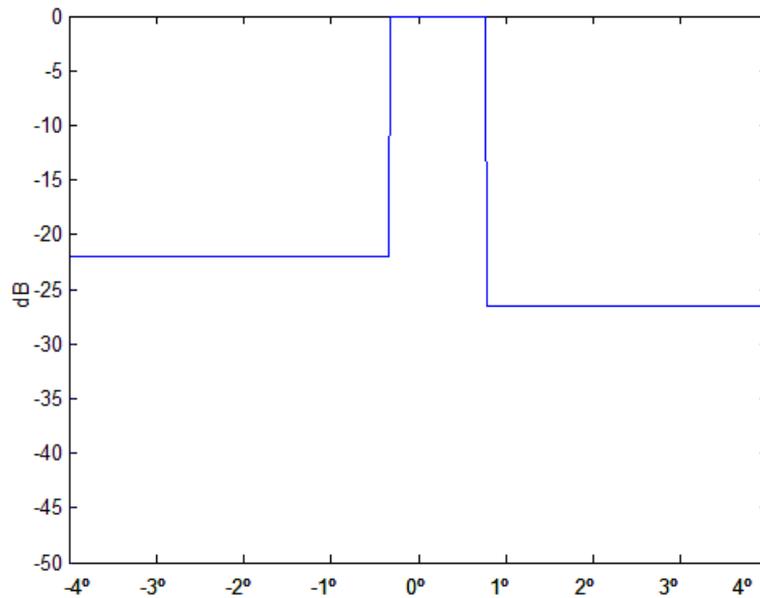


Fig. 5.17 – Máscara para uma refletora com assimetria de 4,5 dB

Após 7852 gerações, o resultado foi o da Fig. 5.18. Nesse caso, a diferença entre os lóbulos ficou em 7,41 dB. O valor do *fitness* nesse caso ficou em 400 de um total de 400 significando que a antena ficou totalmente inserida dentro da máscara apesar da diferença dos lóbulos não ter ficado em 4,5 dB, como determinado na máscara. Caso haja necessidade que a diferença entre os lóbulos fique em 4,5 dB, pode-se adotar a técnica dos erros quadráticos médios em conjunto com a máscara para atingir esse resultado. A Fig. 5.19 mostra a evolução do *fitness* para essa situação. Nesse caso, cabe comentar um pouco mais sobre a evolução da função *fitness*. É possível notar que para esse caso, a função ficou constante praticamente durante todo o processo de otimização, tendo um início em aproximadamente 350 pontos e estabilizando-se por volta da geração 500. Isso aconteceu devido ao nível de exigência do critério de parada que demandou assertividade de 100% da função *fitness*. Caso houvesse maior relaxamento desse valor, o número de gerações seria menor.

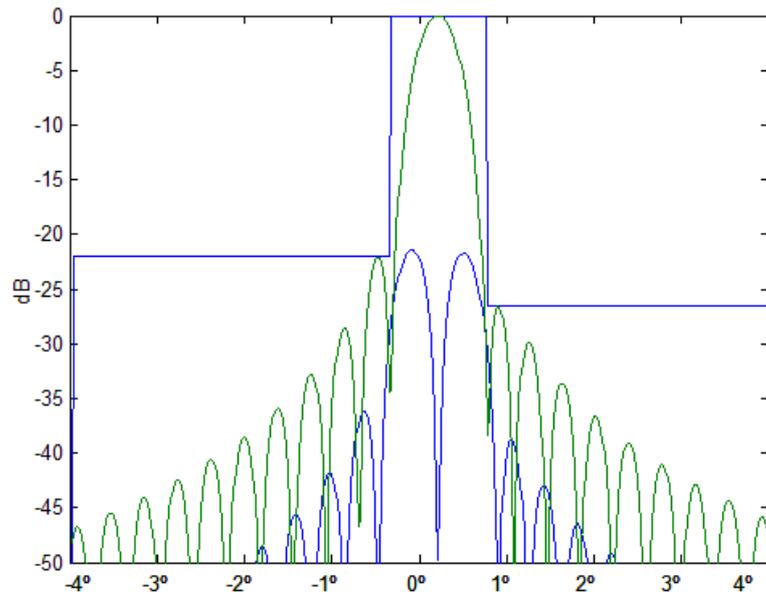


Fig. 5.18 – Resultado obtido após a otimização

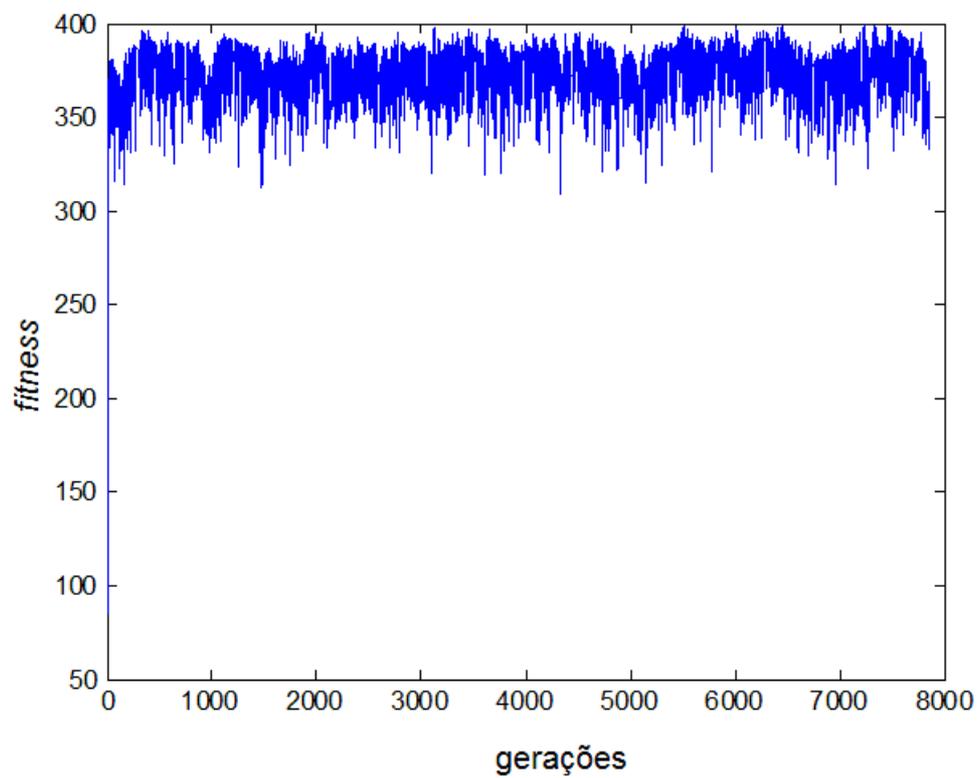


Fig. 5.19 – Evolução do *fitness* médio para cada geração na otimização

6. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Ao fim deste trabalho, foi mostrada uma nova forma de se projetar antenas. Aqui, levou-se em consideração a modulação dos sistemas de comunicação de forma que a antena trabalhe de forma eficiente, não levando em conta apenas seu desempenho individual. As antenas foram analisadas com máxima precisão computacional através do método dos momentos, o que diferiu de outros trabalhos publicados. Deve-se ainda considerar que a irradiação real de cada um dos elementos dos conjuntos de antenas foi calculada. Em outros trabalhos publicados [12][14][43][45], considera-se que cada elemento do conjunto seja uma antena isotrópica.

No cenário apresentado, foi projetada uma antena de fio com assimetria de 6,5 dB entre os lóbulos interferentes em outros dois sistemas de comunicação, um em 16-QAM e outro em 64-QAM. Como pôde ser visto, a obtenção dessa antena só foi possível após o uso do *fitness* prevendo não só a diferença entre os lóbulos laterais, mas também que a antena se comportasse dentro de limites da máscara. A obtenção dessa antena pelo método do erro quadrático médio não foi possível pelo fato de diversas antenas que poderiam ser elegíveis para esse cenário serem desconsideradas pela baixa pontuação obtida pelo *fitness*. Isso mostrou que apesar da operação da plataforma computacional ser simplificada, com o uso de arquivos texto na colocação dos parâmetros, a escolha da geometria base nunca é uma tarefa trivial e requer conhecimento prévio de antenas.

Apesar disso, a plataforma computacional se mostrou uma ferramenta eficaz no projeto de antenas em cenários específicos, sendo facilmente extensível para análise de outros pontos além de permitir a otimização em tempo real, permitindo a construção mais fácil de antenas e com menos gasto de energia. Sugere-se como trabalho futuro a simulação de um ambiente OFDM, utilizado pelo padrão LTE, para verificar como se pode aplicar o conceito apresentado nesta tese no projeto de antenas, levando em consideração um cenário com a modulação OFDM além de possibilidades referentes a modulação adaptativa, existente em sistemas de comunicação modernos. Para isso,

deve-se alterar o módulo de análise de sistema de comunicação para um que analise o padrão LTE. Aponta-se aqui, que já existe um módulo desenvolvido em MATLAB e baseado no CML, construído pela Universidade de Viena que analisa o padrão LTE [51].

Também é deixado como trabalho futuro evoluir a camada de otimização. Nesse trabalho foi utilizado apenas um Algoritmo Genético simples, mas sabe-se que o poder do Algoritmo Genético pode ser aumentado com o uso de técnicas como elitismo, mutação e *crossover* adaptativos, entre outros. Com isso, em algumas situações, pode-se obter uma maior velocidade na convergência do *fitness*.

Outro ponto que ainda pode ser melhorado são os *softwares* de análise de antenas. Quanto ao WebPRAC, ainda não é possível realizar a análise de conjuntos de antenas refletoras. Assim, seria interessante a evolução desse software para permitir essa possibilidade.

A interface Web de ambos os *softwares* de análise de antenas pode ser melhorada, permitindo uma visualização real do diagrama de radiação a medida que os parâmetros da antena são alterados. Essa evolução pode ajudar no ensino de antenas, mostrando como a alteração de cada parâmetro da antena pode influenciar no diagrama, ganho, polarização cruzada, entre outros.

REFERÊNCIAS

- [1] R. Rabelo, “Redes de Acesso Metropolitanas Sem-fio”. Congresso de Iniciação Científica, 2005. Brasília-DF.
- [2] Shakkottai, S., Rappaport, T. S., Karlsson, P. C, “Cross-layer design for wireless networks”, IEEE Communications Magazine, v. 41, p. 74-80, Oct. 2003.
- [3] Srivastava, V., Motani, M., “Cross-layer design: a survey and the road ahead”, IEEE Communications Magazine, v. 43, p. 112-119, Dec. 2005
- [4] Qingwen L., Shengli Z., Giannakis, G. B., “Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless link”, IEEE Transactions on Wireless Communications, v. 3, p. 1746-1755, Sept. 2004.
- [5] Aboutorab, N., Mohammadi A., “A cross-layer design of wireless IP systems using effective bandwidth and MQAM adaptive modulation, Telecommunications Systems, v. 46, p. 343-351, 2011.
- [6] R. Rabelo, M. Terada, “Taxa de Rejeição Sistêmica no Projeto de Antenas: Um Novo Conceito Multi-Camadas em Redes de Comunicação Sem Fio”, MOMAG 2010. V1, p. 326-330.
- [7] W. Stutzman, G. Thiele, “*Antenna Theory and Design*”, 3rd Edition
- [8] M. Terada, “*Reflector Antennas*”, Wiley Encyclopedia of RF and Microwave Engineering, John Wiley & Sons, 2005, v. 5, p. 4450-4474.
- [9] S. Sesia, I. Toufik, M. Baker, “*LTE - The UMTS Long Term Evolution: From Theory to Practice*”, John Wiley & Sons, 2009.
- [10] H. Holma, A. Toskala, “*LTE for UMTS OFDMA and SC-FDMA Based Radio Access*”, John Wiley & Sons, 2009.
- [11] O. Bucci, G. D’Elia, G. Mazzella, G. Panariello, “Antenna Pattern Synthesis: A New General Approach”, Proceedings of the IEEE, Vol. 82, No. 3, March 1994, p. 358-371.
- [12] J.H.Winters, “Smart Antennas for Wireless Systems”, Personal Communications, IEEE, v. 5, p. 23-37, 2002.
- [13] D. Correia, A.J.M. Soares, M.A.B. Terada, “Optimization of Gain, Impedance and Bandwidth in Yagi-Uda Antennas Using Genetic Algorithm”, 1999 SBMO/IEEE

- MTT-S, AP-S and LEOS International Microwave and Optoelectronics Conference, Rio de Janeiro – RJ, August 9-12, 1999.
- [14] Haupt, R. L., “Thinned arrays using genetic algorithms”, IEEE Transactions on Antennas and Propagation, July 1994.
- [15] G. K. Mahanti, N. N. Pathak, P. L. Mahanti, “Synthesis of Thinned Linear Antenna Arrays with Fixed Sidelobe Level using Real-Coded Genetic Algorithm”, Progress in Electromagnetics Research, PIER 75, p. 319-328, 2007.
- [16] R. Rabelo, M. Terada, “Analysis and Optimization of Wire Antennas over the Internet”. IEEE Antennas and Propagation Magazine, v. 52, p. 188-194, 2010.
- [17] A.H.El Zooghby, C.G.Christodoulou, M. Georgiopolous, “A neural network-based smart antenna for multiple source tracking”, IEEE Transactions on Antennas and Propagation, 2002, v.48, p. 768-776.
- [18] M. Terada, W. L. Stutzman, “Computer-Aided Design of Reflector Antennas”, Microwave Journal, 38, August 1995, p. 64-73.
- [19] M. Terada, A. Soares, F. Silva, S. Fonseca, “GRADMAX: Um Programa para Análise e Otimização de Antenas de Fios”, TELEM092 (*Software Educational*): X Simpósio Brasileiro de Telecomunicações e V Simpósio Brasileiro de Microondas, Brasília-DF, pp. 15-18, Julho 1992.
- [20] R. Rabelo, M. Terada and W. Stutzman, “Analysis of Reflector Antennas through the World Wide Web”, IEEE Antennas and Propagation Magazine, v. 49, p. 113-116, Apr. 2007.
- [21] R. Rabelo, M.Terada and W.Stutzman, “WebPRAC: A Reflector Antenna Analysis Code for the Internet”, 2008. IEEE Antennas and Propagation Symposium, 2008, San Diego – California. Proceedings of the 2008 IEEE Antennas and Propagation Symposium 2008. v.1, p. 1-4.
- [22] Coded Modulation Library – <http://www.iterativesolutions.com> acessado em 01 de Agosto de 2011.
- [23] Couch, Leon W., “Digital and Analog Communication Systems”, Editora Prentice Hall, Sexta Edição, 2001.
- [24] B.P.Lathi, “Modern Digital and Analog Communications Systems”, Oxford University Press, Terceira Edição, 1998.
- [25] J.G.Proakis, M.Salehi, “Communications Systems Engineering”, Prentice Hall, Segunda Edição, 2001.

- [26] A. S. Tanenbaum, “Redes de Computadores”, Tradução da Quarta Edição, Editora Campus.
- [27] S. Shakkotai, T. S. Rappaport, P. C. Karlsson, T. Sonera, “Cross-Layer Design for Wireless Networks”, IEEE Communications Magazine, Oct/2003, p. 74-80.
- [28] Holland, John H. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, Mich.: University of Michigan Press 2d ed. 1992.
- [29] S. Rezende, “*Sistemas Inteligentes: Fundamentos e Aplicações*”, Editora Manole, 1ª Edição, 2003.
- [30] K. F. Man, K. S. Tang, S. Kwong, “*Genetic Algorithms: Concepts and Designs*”, v.1, Springer, 1999.
- [31] M. Mitchell, “*An introduction to genetic algorithms*”, MIT Press, 1998.
- [32] Y. Rahmat-Samii, E. Michielssen, “*Electromagnetic Optimization by Genetic Algorithms*”, John Wiley & Sons, 1999.
- [33] K. Deep, K. N. Das, “Performance improvement of real coded genetic algorithm with Quadratic Approximation based hybridisation”, International Journal of Intelligent Defence Support Systems, Volume 2, p. 319-334, 2009.
- [34] Java Reference – Acessado em 04 de Julho de 2012 - <http://www.oracle.com/technetwork/java/index.html>
- [35] Java History Index – Acessado em 04 de Julho de 2012 - <http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>
- [36] J. Gosling, B. Joy, G. Steele, G. Bracha, “Java™ Language Specification”, 3rd Edition. Editora Addison-Wesley. ISBN: 0321246780
- [37] *How the Java Virtual Machine Works*, acessado em 04 de Julho de 2012 - <http://www.codeproject.com/Articles/30422/How-the-Java-Virtual-Machine-JVM-Works>
- [38] The Original Applet Sandbox - acessado em 08 de Julho de 2012 - <http://www.securingjava.com/chapter-two/>.
- [39] *Demonstration Applet and Applications* - acessado em 08 de Julho de 2012 - <http://docs.oracle.com/javase/1.5.0/docs/relnotes/demos.html>.
- [40] Applets para o ensino de eletromagnetismo – acessado em 08 de Julho de 2012 - <https://sites.google.com/site/professorpifer/Home/fisica-mix/eletromagnetismo>

- [41] *What is Java Swing* - acessado em 09 de Julho de 2012 - <http://docs.oracle.com/javase/tutorial/ui/overview/intro.html>
- [42] E.A.Paes, M.A.B.Terada, “Análise de refletores offset com polarização circular”, Projeto Final de Graduação, Dezembro/1997 – Universidade de Brasília.
- [43] H. Gazzah, “Optimum Antenna Arrays for Isotropic Direction Finding”, IEEE Transactions on Aerospace and Electronic Systems, v.47, p. 1482-1489, April 2011.
- [44] D. Tse and P. Viswanath, “*Fundamentals of Wireless Communication*”. Cambridge University Press”, 2005.
- [45] H. Yordanov, P. Russer, M. Ivrlac, J. Nossek, “Arrays of isotropic radiators – a field-theoretic justification”, 2009 International ITG Workshop on Smart Antenna, Berlin, Germany, 2009.
- [46] A. Julian, J. C. Logan, J. W. Rockway, “MININEC: A mini-numerical electromagnetics code”, Final Report, Naval Ocean Systems Center, San Diego, CA. Communications Research and Technology Division, Apr. 1981 – May 1982.
- [47] Lesser GPL License - <http://www.fsf.org/licenses/licenses/lgpl.html> acessado em 01 de Agosto de 2011.
- [48] S. G. Santarelli, D. E. Goldberg, T. Yu, “Optimization of a Constrained Feed Network for an Antenna Array Using Simple and Competent Genetic Algorithm Techniques”, Session WMSA in 2004 Genetic and Evolutionary Computation Conference.
- [49] W. Stutzman, G. Thiele, “*Antenna Theory and Design*”, 1st Edition
- [50] G. Hall, “*The ARRL Antenna Book*”, published by The Antenna Radio Relay League, 1988.
- [51] C. Mehlhruer, M. Wrulich, J. C. Ikuno, D. Bosanska, M. Rupp, “Simulating the Long Term Evolution Physical Layer”, 17th European Signal Processing Conference, 2009.

APÊNDICE A. MANUAL DE USO DO GRADMAX

Para usar o GRADMAX por meio de Applets Java, basta acessar o endereço www.ene.unb.br/~terada/antennas e em seguida, escolher a opção de antenas de fio. Após essa escolha, deverá aparecer a Fig. A.1 caso tenha instalado no computador uma máquina virtual Java.

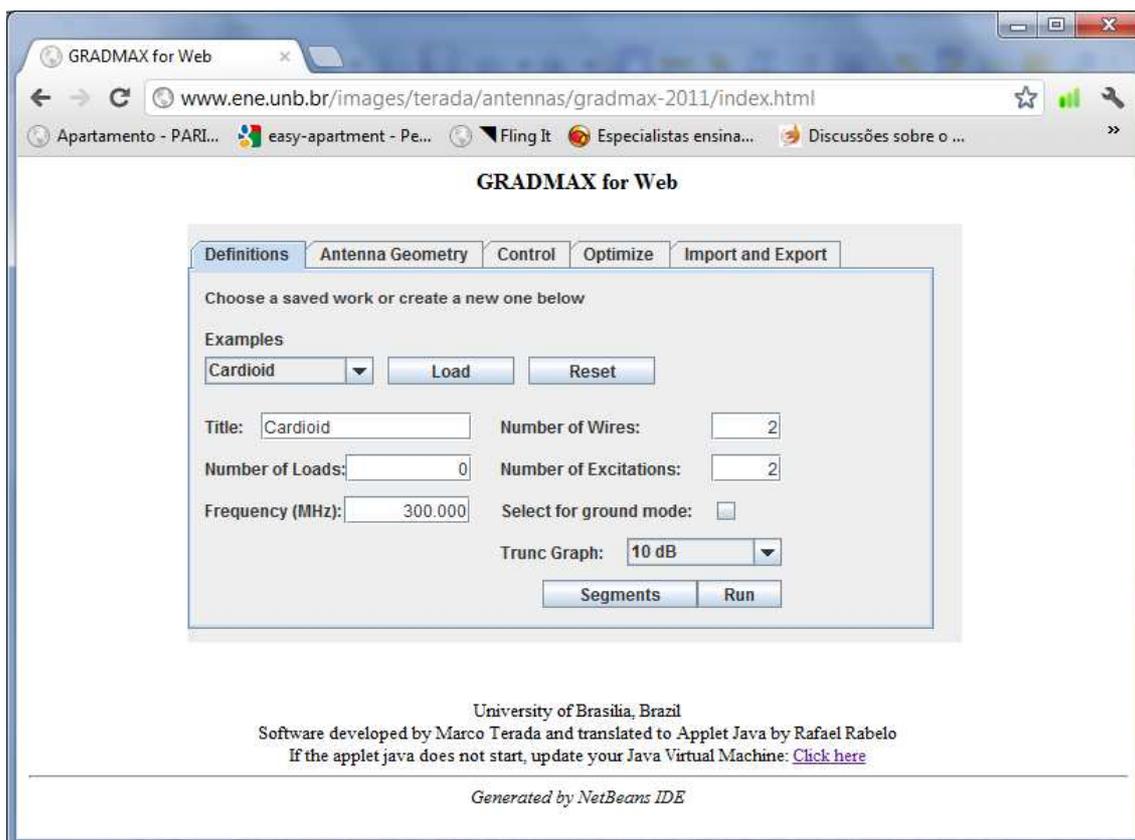


Fig. A.1 – Tela inicial do GRADMAX

Por padrão, existem algumas antenas pré-configuradas no sistema que podem ser utilizadas como exemplos. São elas: *Cardioid*, *Dipole*, *Dipole with Load*, *V-Dipole*, *Monopole* e *Yagi*. Pode-se escolher qualquer delas e clicar no botão *Load* para carregar os dados dessa antena. Nas próximas imagens, todos os parâmetros são referentes ao primeiro exemplo, *Cardioid*. Os dados serão carregados em diversos campos, que podem ser acessados por diversas abas. A primeira delas é a *Definitions*, onde é possível inserir, editar ou consultar informações básicas da antena como, por exemplo:

- *Title*: título da antena;

- *Number of wires*: quantidade de fios que comporá o conjunto a ser analisado.
- *Number of loads*: quantidade de cargas que comporá o conjunto a ser analisado;
- *Number of excitations*: quantidade de alimentações que comporá o conjunto a ser analisado;
- *Frequency (MHz)*: a frequência, em MHz em que será analisada a antena para fins de cálculo de ganho, diagrama de radiação, entre outros parâmetros;
- *Ground mode*: quando selecionado, essa opção permite simular perfeitamente o plano terra (plano xy);
- *Trunc graph*: opção que permite selecionar o limiar de corte do gráfico a ser traçado. Por padrão, o sistema deixa selecionado em -10 dB, considerando que o diagrama de radiação da antena tem máximo em 0 dB.

A aba *Antenna Geometry*, Fig. A.2, permite inserir, consultar ou editar a geometria da antena, incluindo as coordenadas X, Y e Z para cada fio – representadas pelos campos X1 e X2 (x inicial e final), Y1 e Y2 (y inicial e final) e Z1 e Z2 (z inicial e final) -. Todos os valores inseridos nessa aba estão na unidade m (metros).

Definitions Antenna Geometry Control Optimize Import and Export

Geometry of each wire (meters):

Wire 1/2

NS: 10 X1: 0.0 Y1: 0.0 Z1: -0.25 C1: 0

R: 0.0050 X2: 0.0 Y2: 0.0 Z2: 0.25 C2: 0

< Previous Wire Next Wire >

Fig. A.2 – Aba de inclusão da geometria da antena no GRADMAX

No campo *NS* é possível definir a quantidade de segmentos que o fio será dividido. É recomendado um valor entre 10 e 20 segmentos para cada λ , onde λ é o comprimento de onda. Esse valor é utilizado pelo método dos momentos para calcular as integrais de corrente que resultarão no diagrama de radiação da antena. Quanto maior o número de segmentos, maior será a precisão do cálculo, entretanto, deve-se levar em consideração que quanto maior o número de segmentos menor será o desempenho do *software*. Além disso, é importante lembrar que um fio de pequeno comprimento (operando em frequências de 2,5 GHz, por exemplo), quando dividido em 50 ou mais segmentos, fatalmente gerará problemas numéricos de *overflow e/ou underflow* o que impossibilita a análise da antena. A divisão dos fios em segmentos também é importante para a identificação dos pulsos. Para o GRADMAX, os pulsos estão localizados no centro do fio. Sendo assim, um número par de segmentos gerará número ímpar de pulsos.

No campo *R*, pode-se definir o raio do fio. É recomendável que o valor do raio seja menor ou igual a $\lambda/100$.

Por fim, os campos *C1* e *C2* são os pontos de conexão para o *i*-ésimo fio. É altamente recomendado que se desenhe o diagrama para minimizar a possibilidade de erros ao inserir essa informação no GRADMAX. O campo *C1* deve ser preenchido com informações de conexão do ponto inicial do fio, *E1*, e o campo *C2*, com informações de conexão do ponto final do fio, *E2*. Para o preenchimento desses campos deve-se levar em conta a ordem de inclusão dos fios no GRADMAX, pois uma conexão só existe com um fio previamente inserido.

Os possíveis valores para os campos *C1* e *C2* são:

- 0: quando inserido 0 (zero), não haverá conexão no ponto;
- -*i*: significa que haverá conexão com o plano terra para o *i*-ésimo fio;
- *k*: se o ponto final do *i*-ésimo fio está conectado ao inicial *k*-ésimo fio (na Fig. XX, fio *E1* até *E2* ou *E2* até *E1*), onde $k < i$;
- -*k*: se o ponto inicial do *i*-ésimo fio está conectado ao inicial *k*-ésimo fio (na Fig. XX, *E1* até *E1* ou *E2* até *E2*), onde $k < i$.

Para tornar mais fácil o entendimento de como utilizar os pontos de conexão, considere a geometria da Fig. A.3. Nessa geometria existem três fios. O fio 1 está

conectado ao plano terra e os outros dois, estão conectados com o primeiro por um ponto.

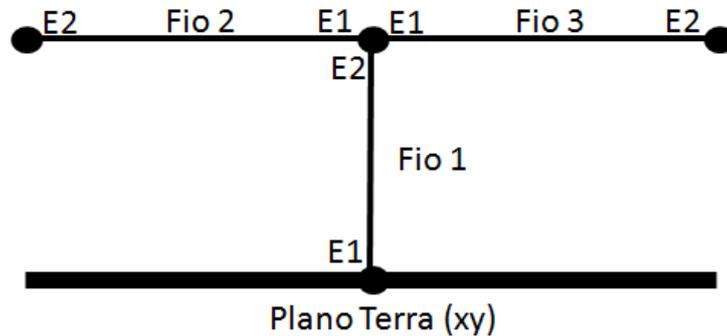


Fig. A.3 – Desenho da geometria de um fio para exemplificação de uso no GRADMAX

Para o exemplo da Fig. A.3, tem-se que os valores para C1 e C2 devem ser:

- Fio 1: C1 = -1 (conexão ao plano terra do ponto E1); C2 = 0 (apesar de estar conectado ao Fio 2 e Fio 3, até esse momento o segundo e o terceiro fio não estavam especificados, portanto deve-se considerar esse valor 0).
- Fio 2: C1 = 1 (E1 do fio 2 se conecta com o fio 1); C2 = 0 (Não há conexão)
- Fio 3: C1 = 1 (E1 do fio 3 se conecta com o fio 1), ou ainda, outra possibilidade seria C1 = -2 (E1 do fio 2 se conecta com E1 do fio 3); C2 = 0 (Não há conexão do ponto E2).

Continuando na aba *Definitions*, tem-se ainda dois botões muito importantes que serão detalhados posteriormente: *Run* e *Segments*. Antes disso, cabe-se detalhar melhor as informações da aba *Control*, Fig. A.4. Nessa aba, é possível configurar as informações sobre as alimentações (*excitations*) e as cargas (*loads*). Quanto às alimentações, deve-se entrar com o número do pulso (*Pulse Number*) além de informações de tensão (*Voltage Magnitude*) e fase, em graus (*Phase degree*).

O número do pulso será o ponto onde será realizada a alimentação. Na aba anterior, foi definido que o primeiro fio da *Cardioid* seria dividido em 10 segmentos. Como um número par de segmentos gera um número ímpar de pulsos, tem-se que o quinto pulso encontra-se exatamente no centro do fio. No exemplo da Fig. A.5, pode-se verificar que o quinto pulso está sendo alimentado com 1 V de tensão e com diferença de 0 graus de fase.

A informação do número do pulso também é utilizada para localizar as cargas. Após escolhido o número do pulso, pode-se colocar o valor da resistência (*resistance*) e reatância (*reactance*).

The image shows a software interface with five tabs: 'Definitions', 'Antenna Geometry', 'Control', 'Optimize', and 'Import and Export'. The 'Control' tab is active. It contains two sections: 'Excitation 1/2' and 'Load 1/0'. In the 'Excitation 1/2' section, there are three input fields: 'Pulse Number' with the value '5', 'Voltage Magnitude' with '1.0', and 'Phase (degree)' with '0.0'. Below these are two buttons: '< Previous Excitation' and 'Next Excitation >'. The 'Load 1/0' section has three input fields: 'Pulse Number' with '0.0', 'Resistance::' with '0.0', and 'Reactance:' with '0.0'. Below these are two buttons: '< Previous Load' and 'Next Load >'. The interface has a light gray background and blue buttons.

Fig. A.4 – Aba de inclusão de dados de alimentação e cargas

Muitas das vezes pode-se tornar difícil encontrar o número do pulso para ser usado na aba *Control*, principalmente quando há muitos fios no mesmo diagrama. De forma a facilitar, existe o botão *Segments* da aba *Definitions*. Ao clicar nesse botão, aparece uma janela como a mostrada na Fig. XX. Nessa janela, pode-se verificar como será a divisão de cada um dos fios o que torna praticamente imediata, a escolha do número do pulso a ser utilizado para a alimentação ou carga.

Voltando à aba *Definitions*, há o botão *Run*. Esse botão deve ser utilizado após se ter inserido todos os dados da geometria. Ao clicar nele, aparecerá um resumo com os resultados de tensão, corrente e impedância para cada um dos pulsos que foram alimentados ou carregados, Fig. A.6. Em seguida, será apresentado o diagrama de radiação da geometria nos planos xy e xz além dos ganhos em cada direção, Fig. A.7.

Seguindo nas funcionalidades do GRADMAX, tem-se a aba *Optimize*. Essa aba permite otimizar a geometria levando em consideração dois parâmetros, o ângulo, que alterará os ângulos dos fios para otimizar o ganho na direção x, e a distância, que altera a separação dos monopolos para otimizar o ganho na direção x. Ao escolher uma das duas formas de otimização o GRADMAX procurará a melhor geometria para cada caso.

Antenna Geometry

X	Coordinates		Raio	Conection		Pulse No
	Y	Z		C1	C2	
0	0	-0.2	0.005	0	1	1
0	0	-0.15	0.005	1	1	2
0	0	-0.1	0.005	1	1	3
0	0	-0.05	0.005	1	1	4
0	0	0	0.005	1	1	5
0	0	0.05	0.005	1	1	6
0	0	0.1	0.005	1	1	7
0	0	0.15	0.005	1	1	8
0	0	0.2	0.005	1	0	9
0.25	0	-0.2	0.005	0	2	10
0.25	0	-0.15	0.005	2	2	11
0.25	0	-0.1	0.005	2	2	12
0.25	0	-0.05	0.005	2	2	13

Fig. A.5 – Janela com informações dos segmentos do fio

Results

?

Pulse Number 5
Voltage = (1;0 j)
Current = (0,022;0,001 j)
Impedance = (45,322;-1,848 j)

Pulse Number 14
Voltage = (1,77;-2,496 j)
Current = (0,002;-0,02 j)
Impedance = (134,684;76,987 j)

x-direction:
Original Gain = 0 dBi
Gain after 0 iteration(s) = 5,295 dBi

Ok

Fig. A.6 – Janela com informações sobre tensão, corrente e impedância em cada pulso bem como o ganho da antena calculada

Por fim, tem-se a aba *Import and Export*. Nela é possível importar ou exportar dados de uma geometria. Como mencionado anteriormente no capítulo sobre Java, os Applets Java não conseguem alterar informações no sistema de arquivos do computador. Assim, não é possível permitir que um Applet Java salve um arquivo com

as informações da geometria no computador em que ele está executando. Assim, para contornar a necessidade de salvar uma geometria para posterior uso, foi adicionada essa funcionalidade que exporta todas as informações da geometria em uma cadeia de caracteres separadas por ponto e vírgula, “;”.

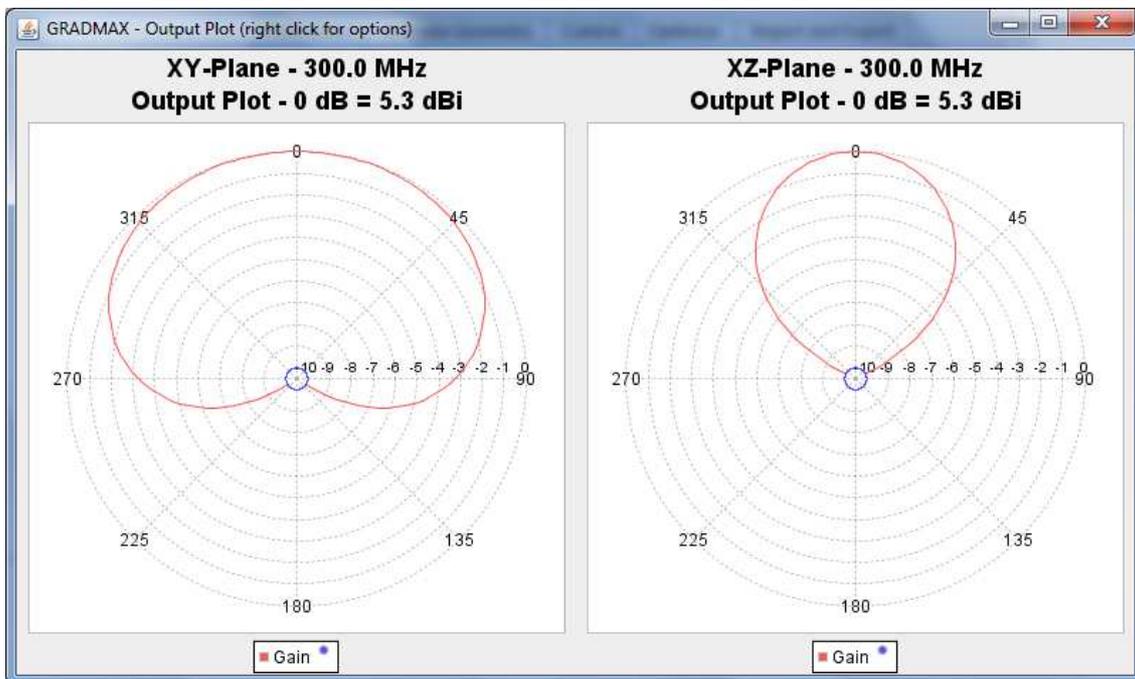


Fig. A.7 – Diagrama de radiação da Cardióide, gerada pelo GRADMAX

Para exemplificar, tem-se que ao clicar no botão *Export* da aba *Import and Export* após ter carregado o exemplo *Cardioid*, o GRADMAX abrirá uma janela com a seguinte cadeia de caracteres, Fig XX. Todos os dados da geometria estarão disponíveis para que possam ser armazenados localmente no computador para posterior importação nessa mesma aba.

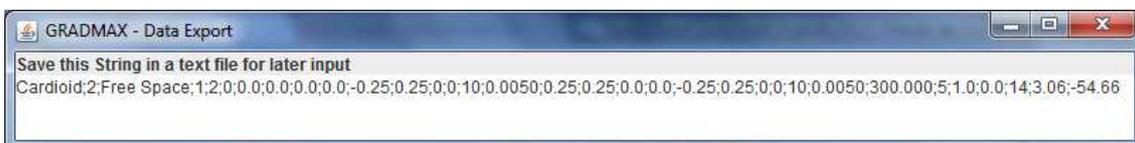


Fig. A.8 – Janela de exportação da *string* com os parâmetros da geometria inserida no GRADMAX

APÊNDICE B. BIBLIOTECA GRADMAX.JAR

O GRADMAX também pode ser utilizado para a análise de geometrias de fio por meio da biblioteca GRADMAX.jar diretamente em códigos Java. Para esse fim, deve-se importar a biblioteca, instanciar a classe *Wires* e em seguida, executar o método *evaluate*, que recebe três informações como parâmetro:

- String exportada pela função Export no Applet Java, que representa a geometria da antena;
- Valor, em dB, em que o gráfico deve ser truncado. No exemplo, 30 significa que o gráfico será truncado em -30dB;
- Plano o qual deve ser calculado o diagrama. Para plano xz, colocar o valor "0". Para o plano xy, colocar o valor "1".

Após executar o método, será devolvido um vetor com 180 pontos que representará o gráfico polar do plano xy ou xz. Cada ponto representa 2 graus no diagrama.

Na Fig. B.1, tem-se exemplo de um código em Java que utiliza a biblioteca GRADMAX.jar.

```
import GRADMAX.Wires;

public static void main(String args[]) {
    Wires wire = new Wires();
    double[] response;
    double dbtrunc = 30;
    String geometry = "Cardioid;2;Free Space;1;2;0;0.0;0.0;0.0;0.0;-
0.25;0.25;0;0;10;0.0050;0.25;0.25;0.0;0.0;-
0.25;0.25;0;0;10;0.0050;300.000;5;1.0;0.0;14;3.06;-54.66";
    response = wire.evaluate(geometry,dbtrunc,0);
}
```

Fig. B.1 – Exemplo de código Java utilizando a biblioteca GRADMAX.jar

Essa mesma biblioteca pode ser utilizada integrada por meio do Matlab. Antes disso, deve-se realizar a seguinte configuração:

1. Fechar o MATLAB;
2. Copiar o arquivo GRADMAX.jar para a pasta \$MATLABROOT/java/jar, onde \$MATLABROOT é a pasta onde o Matlab foi instalado;
3. Abrir o arquivo \$MATLABROOT/toolbox/local/classpath.txt e adicionar a linha \$matlabroot/java/jar/GRADMAX.jar , considerando que nesse caso, o .jar gerado é o arquivo GRADMAX.jar. Se estiver no Windows 7 ou posterior e o editor de texto mencionar que o arquivo está sendo utilizado por outro programa, provavelmente você não tem permissão para gravar diretamente na pasta do Matlab. Para contornar esse problema, salve o arquivo primeiramente em uma pasta qualquer e copie-o em seguida para a pasta indicada;
4. Abrir o MATLAB;
5. Instanciar a classe e utilizá-la da mesma forma que se tivesse sido utilizada em um código Java. A Fig. B.2, mostra um exemplo de um script .M para executar o método *evaluate* da classe *Wires*.

```
import GRADMAX.*

wires=Wires()
result=wires.evaluate('Cardioid;2;Free Space;1;2;0;0.0;0.0;0.0;0.0;-
0.25;0.25;0;0;10;0.0050;0.25;0.25;0.0;0.0;-
0.25;0.25;0;0;10;0.0050;300.000;5;1.0;0.0;14;3.06;-54.66',30,0);
```

Fig. B.2 - Exemplo de código MATLAB utilizando a biblioteca GRADMAX.jar

APÊNDICE C. MANUAL DE USO DO WEBPRAC

Para usar o WebPRAC por meio de Applets Java, basta acessar o endereço www.ene.unb.br/~terada/antennas e em seguida, escolher a opção de antenas refletoras. Após essa escolha, deverá aparecer a Fig. C.1 caso tenha instalado no computador uma máquina virtual Java.

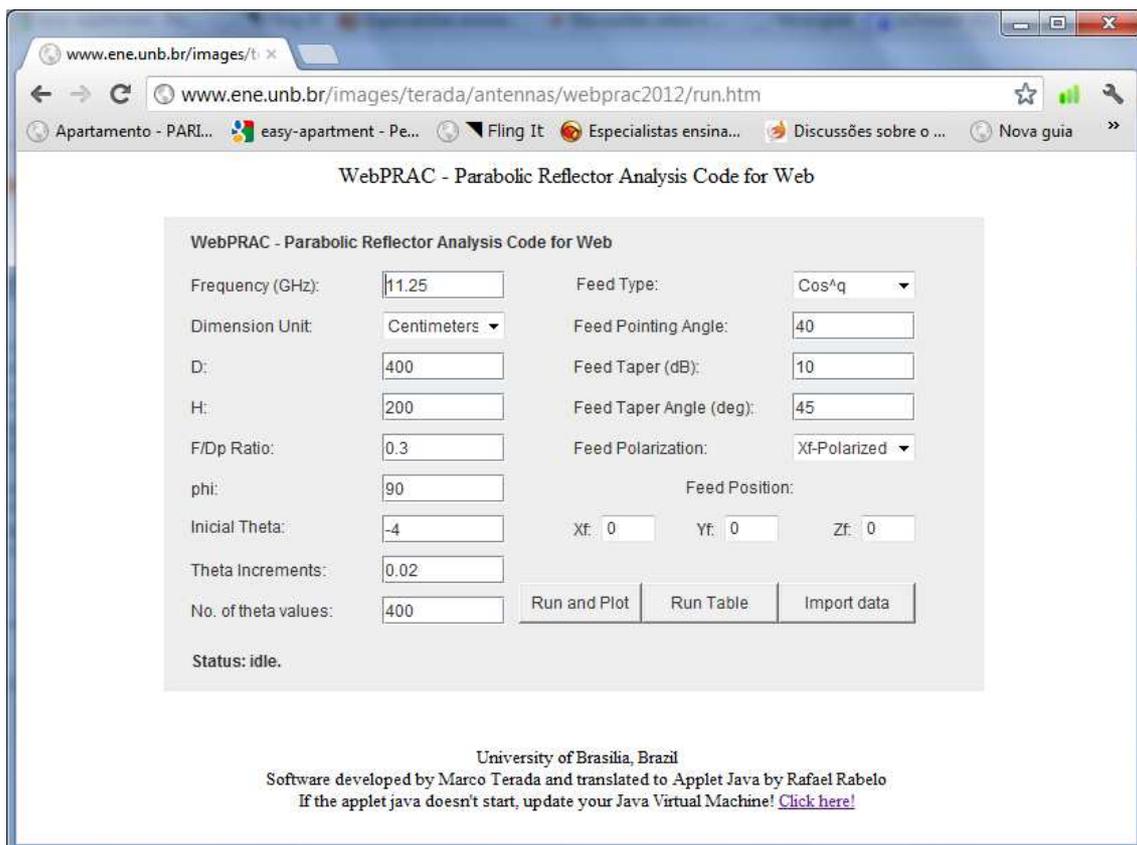


Fig. C.1 – Tela inicial do WebPRAC

O uso do WebPRAC é bem simples sendo basicamente essa tela inicial onde devem ser preenchidos alguns parâmetros e em seguida, utilizar o botão *Run and Plot* ou *Run Table* para obter os resultados. Apesar da facilidade na inclusão dos parâmetros, deve-se conhecer a geometria de uma antena refletora para utilizar o WebPRAC com eficiência, Fig. C.2. Os principais pontos e ângulos dessa geometria estão descritos na Tabela C-1.

Tabela C-1 – Definições de cada um dos parâmetros de uma antena refletora

Símbolo	Definição
D	Diâmetro da abertura circular projetada do refletor parabólico principal ($D = D_p$ para uma parabólica de eixos simétricos)
D_p	Diâmetro da abertura circular projetada do parabolóide gerador
H	Distância de offset = distância do eixo de simetria ao canto inferior do refletor
F	Distância focal do parabolóide
Ponto F	Ponto focal
Ponto A	Ápice do refletor gerador
Ponto B	Ponto no refletor principal que corresponde a bissetriz do ângulo subtendido pelo refletor quando visto do alimentador
Ponto C	Ponto no refletor principal que corresponde ao centro da abertura projetada
Ponto P	Ponto no refletor principal correspondente ao pico do diagrama do alimentador
ψ_f	Ângulo do pico do diagrama do alimentador em relação ao eixo de simetria; o alimentador aponta diretamente para o ponto P
ψ_B	Valor de ψ_f que corresponde a bissetriz do ângulo subtendido pelo refletor
ψ_C	Valor de ψ_f quando o alimentador aponta para o ponto C do refletor, que corresponde ao centro de abertura
ψ_s	Metade do ângulo subtendido pelo refletor quando visto do ponto focal
$(\psi_U - \psi_L)$	Ângulo compreendido pelo refletor parabólico principal, visto pelo ponto focal.

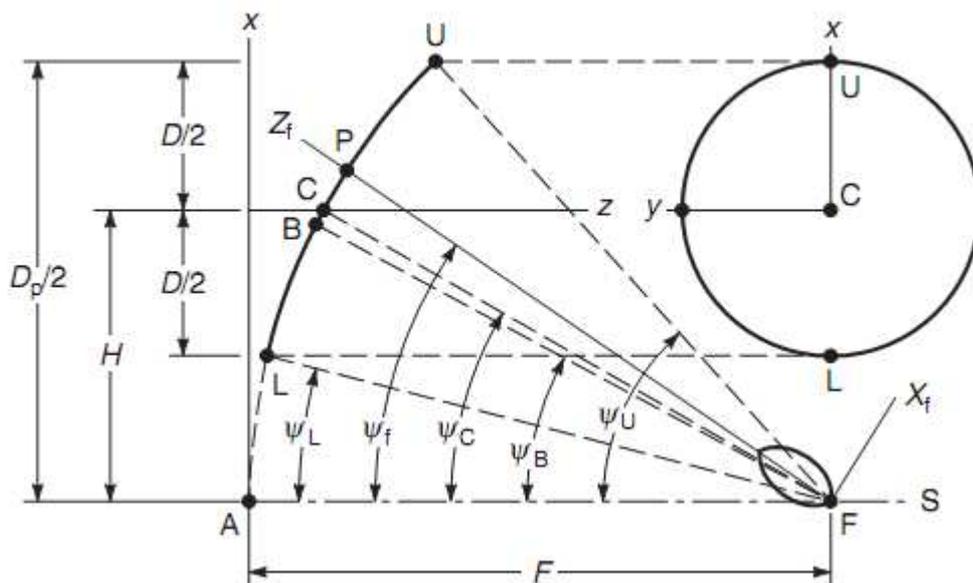


Fig. C.2 – Geometria de uma antena refletora com a marcação dos parâmetros

Outra informação importante a conhecer referente ao WebPRAC é o modelo de alimentação. Atualmente, o WebPRAC implementou apenas o modelo $\cos^q(\Theta_n)$. Essa expressão aproxima o padrão de radiação de cornetas corrugadas cônicas de forma muito satisfatória

Todos os parâmetros necessários para se realizar a análise de uma antena refletora estão disponíveis. Atualmente só está disponível o modelo de alimentação $\cos^q(\Theta_n)$.

- *Frequency*: frequência em que a antena funcionará. Expressa em Ghz.
- *Dimension Unit*: unidade em que os dados serão incluídos. Pode-se escolher cm, m, polegadas ou pés.
- *D*: o diâmetro da antena, expressa na unidade selecionada no item anterior.
- *H*: a altura da antena em relação ao eixo S.
- *F/Dp Ratio*: a razão entre o valor de F e o valor de Dp.
- *Phi*: Ângulo de visualização
- *Inicial theta*: Refere-se ao valor do primeiro ângulo que será feita a análise do ganho.
- *Theta Increments*: Refere-se ao incremento de ângulo em relação ao theta inicial.
- *N. of theta values*: Refere-se ao número de pontos que serão calculados. Combinado com os valores de Inicial Theta e Theta Increments, esse valor definirá os ângulos thetas que serão calculados.
- *Feed Type*: Refere-se a modelagem matemática do alimentador. Atualmente, apenas o modelo \cos^q está implementado.
- *Feed Pointing Angle*: Refere-se ao ângulo do alimentador em relação ao plano S. Ângulo ψ_f .
- *Feed Taper*: Refere-se ao valor em dB do modelo \cos^q .
- *Feed Taper Angle*: Refere-se ao valor do anglo, em graus do modelo \cos^q .
- *Xf, Yf e Zf*: Posição do alimentador em metros nas coordenadas X, Y e Z respectivamente.

Após preenchido os valores, deve-se clicar no botão *Run and Plot*, para realizar a análise da antena, Fig. C.1. Essa análise tem como resultado o diagrama de radiação

da antena na co-polarização (vermelho) e o diagrama de radiação na polarização cruzada (azul). Além disso, é apresentado o ganho da antena e o nível da polarização cruzada com relação ao ganho.

No botão *Run Table*, é disponibilizada uma funcionalidade que permite analisar a antena e exportar os dados para outro software, como o Excel, Matlab, etc. Com isso, é possível traçar os diagramas de co-polarização e polarização cruzada em outro ambiente. Nessa janela também pode ser obtida uma *string* com todos os parâmetros preenchidos na janela principal do *software* para que possam ser posteriormente importados pela função *Import data*. Essa funcionalidade funciona da mesma forma que o GRADMAX.

APÊNDICE D. BIBLIOTECA WEBPRAC.JAR

Da mesma forma que o GRADMAX, o WebPRAC também pode ser utilizado para a análise de antenas refletoras utilizando a biblioteca WebPRAC.jar diretamente em códigos Java. Para esse fim, deve-se importar a biblioteca, instanciar a classe *SingleReflector* e em seguida, executar o método *evaluate*, que recebe duas informações como parâmetro:

- String exportada pela função *Run Table* no *Applet* Java, que contém informações sobre a geometria da antena, ângulo inicial do diagrama (campo *Inicial Theta* no *Applet* Java), resolução do diagrama (campo *Theta Increments* no *Applet* Java), além da quantidade de pontos do diagrama (campo *N. of theta values* no *Applet* Java).
- Polarização do alimentador. Para polarização-y, deve receber valor “0”. Para polarização-x, deve receber valor “1”.

Após executar o método, será devolvido um vetor com duas colunas e com a quantidade de linhas definida em “*N. of theta values*”, em que, na primeira coluna conterà a co-polarização e na segunda a polarização cruzada. Na Fig. D.1, tem-se exemplo de um código em Java que utiliza a biblioteca WebPRAC.jar.

```
import WebPRAC.SingleReflector;

public static void main(String args[]) {
    SingleReflector singleReflector = new SingleReflector();
    double[][] response;
    String geometry = "No
title;400.0;200.0;0.0;0.0;0.0;11.25;0.3;90.0;-
4.0;0.02;400.0;10.0;45.0;40.0";
    response = singleReflector.evaluate(geometry,0);
}
```

Fig. D.1 – Código Java exemplo para o uso da biblioteca WebPRAC.jar

Essa mesma biblioteca pode ser utilizada integrada por meio do Matlab. Antes disso, deve-se realizar a seguinte configuração:

1. Fechar o MATLAB;
2. Copiar o arquivo WebPRAC.jar para a pasta \$MATLABROOT/java/jar, onde \$MATLABROOT é a pasta onde o Matlab foi instalado;
3. Abrir o arquivo \$MATLABROOT/toolbox/local/classpath.txt e adicionar a linha \$matlabroot/java/jar/WebPRAC.jar, considerando que nesse caso, o .jar gerado é o arquivo WebPRAC.jar. Se estiver no Windows 7 ou posterior e o editor de texto mencionar que o arquivo está sendo utilizado por outro programa, provavelmente você não tem permissão para gravar diretamente na pasta do Matlab. Para contornar esse problema, salve o arquivo primeiramente em uma pasta qualquer e copie-o em seguida para a pasta indicada;
4. Abrir o MATLAB;
5. Instanciar a classe e utilizá-la da mesma forma que se tivesse sido utilizada em um código Java. A Fig XX, mostra um exemplo de um script .M para executar o método *evaluate* da classe *SingleReflector*.

```
import WebPRAC.*

singleReflector=SingleReflector()
result=singleReflector.evaluate('No
title;400.0;200.0;0.0;0.0;0.0;11.25;0.3;90.0;-
4.0;0.02;400.0;10.0;45.0;40.0',0);
```

Fig. D.2 - Código MATLAB exemplo para o uso da biblioteca WebPRAC.jar

APÊNDICE E. MANUAL DE UTILIZAÇÃO DA PLATAFORMA COMPUTACIONAL

Para utilizar a plataforma computacional deve-se primeiramente seguir os passos indicados nos Apêndices APÊNDICE B e APÊNDICE D para acoplar as bibliotecas de análise de antenas ao MATLAB e descompactar o arquivo “plataforma.ene.unb.grupoantenas.zip” em uma pasta qualquer do computador apontando o MATLAB para essa pasta. A Tabela E-1 detalha as funções de cada arquivo resultante dessa descompactação.

Como quase todas as plataformas desenvolvidas para o MATLAB, o funcionamento dessa plataforma também é baseado em arquivos textos com a extensão “.m”. Todos os parâmetros necessários para o início da otimização devem ser definidos dentro desses arquivos. Os arquivos `ga_cardioid.m` e `ga_reflector400m.m` podem servir como exemplo de parâmetros a serem incluídos para otimização de antenas de fio e antenas refletoras respectivamente. Cada parâmetro existente nesses arquivos é detalhado na Tabela E-3 onde a coluna R indica os parâmetros utilizados em otimização de antenas refletoras e a coluna F indica os parâmetros utilizados em otimização de antenas de fio.

Após a definição de todos os parâmetros, pode-se iniciar a otimização executando o nome do arquivo no *prompt* do MATLAB. Nesse momento, será determinada a primeira geração do algoritmo genético com a quantidade de indivíduos definidos no parâmetro `pop`. Logo em seguida, é gerado um arquivo “.txt” que, além de armazenar todos os parâmetros da otimização, armazenará os indivíduos com as melhores adequações (*fitness*) encontradas geração após geração.

Para cada indivíduo, também são registrados as *strings* do GRADMAX e WebPRAC com os parâmetros para permitir que os diagramas de radiação possam ser visualizados por meio dos *Applets* Java antes mesmo da otimização ter finalizado.

Tabela E-1 - Listagem dos arquivos da Plataforma Computacional

Arquivo	Definição
core_evaluateString.m	Executado pelo núcleo da plataforma para analisar uma <i>string</i> que
fitness.m	Algoritmo que analisa o <i>fitness</i> do diagrama de radiação de uma geometria.
ga_cardioid.m	Exemplo de otimização de antenas de fio utilizando o exemplo “ <i>Cardioid</i> ” do GRADMAX
ga_core.m	Núcleo do algoritmo genético para antenas de fio
ga_core_reflector.m	Núcleo do algoritmo genético para antenas refletores
ga_reflector400m.m	Exemplo de otimização de antenas refletoras utilizando o refletor de 4m (exemplo inicial do WebPRAC).
ga_selecionapai.m	Algoritmo que implementa o método da roleta para seleção da próxima geração do algoritmo genético
gradmax2matlab.m	Converte os dados calculados pela biblioteca GRADMAX.jar e os converte para variáveis do MATLAB.
gradmaxstring.m	Calcula o diagrama de radiação de uma geometria de fio utilizando os dados da variável <i>string</i>
matlab2gradmax.m	Converte variáveis do MATLAB para uma <i>string</i> do GRADMAX
matlab2webprac.m	Converte variáveis do MATLAB para uma <i>string</i> do WebPRAC
plotstring.m	Traça o gráfico do resultado atual
webprac2matlab.m	Converte os dados calculados pela biblioteca WebPRAC.jar e os converte para variáveis do MATLAB.
webpracstring.m	Calcula o diagrama de radiação de um refletor utilizando os dados da variável <i>string</i>

Tabela E-2 – Listagem dos arquivos de exemplos da Plataforma Computacional

Arquivo	Definição
exemplo01_cardioid_tf1.m	Exemplo utilizado na seção 4.7.7.
exemplo02_reflector400m_tf1.m	Exemplo utilizado na seção 4.7.8.
exemplo03_cardioid_tf2.m	Exemplo utilizado na seção 4.7.9.
exemplo04_arr1_tf1.m	Exemplo utilizado na seção 5.3.1.
exemplo05_arr1_65db_tf1.n	Exemplo utilizado na seção 5.3.2.
exemplo06_arr1_ass_mask_tf2.m	Exemplo utilizado na seção 5.4.
exemplo07_arr1_ass_mask_tf2_and_diff.m	Exemplo utilizado na seção 5.4.1.
exemplo08_reflector_400m_tf2.m	Exemplo utilizado na seção 5.5.

Tabela E-3 - Lista dos parâmetros requeridos pela Plataforma Computacional

Parâmetro	Definição	Ref	Fio
String	String da antena base que será utilizada na otimização. <ul style="list-style-type: none"> No caso das antenas de fio trata-se da geometria utilizando como base a <i>string</i> gerada no GRADMAX. Nesse caso, será otimizado o valor da amplitude e fase da alimentação a partir da segunda alimentação. A alimentação do primeiro pulso é utilizada como referência em 1 V e 0 graus. No caso de antenas refletoras trata-se da geometria utilizando como base a <i>string</i> gerada no WebPRAC. Nesses caso, será otimizado as posições x, y e z do alimentador. 	X	X
gerações	Número máximo de gerações admitidas na otimização.	X	X
stop_criteria	Critério de parada para a otimização. Caso escolher 1, a otimização parará somente quando finalizar todas as gerações ou atingir ao maior valor possível do <i>fitness</i> . Menor que 1 significa a porcentagem sob o melhor <i>fitness</i> que haverá a parada. 0.99 significa 99% do melhor <i>fitness</i> .	X	X
prob_cross	Probabilidade de crossover. Informação utilizada pelo algoritmo genético.	X	X
prob_mutacao	Probabilidade de mutação. Informação utilizada pelo algoritmo genético.	X	X
Pop	Tamanho da população. Informação utilizada pelo algoritmo genético.	X	X
Tensao	Valor limite para a tensão de alimentação nos pulsos dos fios. Caso receber o valor 10, entende-se que o algoritmo permitirá que a tensão do pulso a partir do segundo fio poderá variar de 0 V a 10 V.	N/A	X
dimensão	Valor limite para o deslocamento da posição x, y e z do alimentador da antena refletora.	X	N/A

tipo_fitness	<p>Forma como o fitness será calculado. Até o presente momento, pode receber três valores.</p> <ul style="list-style-type: none"> • Valor 1: caso receba 1, o fitness será calculado de acordo com o método dos mínimos quadrados detalhado na seção 4.7.4. • Valor 2: caso receba 2, o <i>fitness</i> será calculado utilizando o método da máscara detalhado no 4.7.5. 	X	X
k_value	Variável que determina o divisor no caso do fitness = 1.	X	X
antena_desejada	Diagrama de alvo ou máscara. Caso tipo_fitness = 1, trata-se do diagrama de radiação alvo; caso tipo_fitness = 3, trata-se da máscara.	X	X
tipo_antena	<p>Tipo de antena a ser otimizado</p> <ul style="list-style-type: none"> • Valor 1: antenas de fio • Valor 2: antenas refletoras 	X	X
Trunc	Limite em -dB dos diagramas produzidos pelo GRADMAX e/ou WebPRAC	X	X
diagrama_desejado	<ul style="list-style-type: none"> • Para antenas de fio: diagrama_desejado=0 (plano XZ), diagrama_desejado=1 (plano XY) • Para antenas parabólicas: diagrama_desejado=0 (COPOL), diagrama_desejado=1 (XPOL) 	X	X