

Laboratório de Modelagem, Análise e Controle de Sistemas Não-Lineares
Departamento de Engenharia Eletrônica
Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627, 31270-901 Belo Horizonte, MG Brasil
Fone: +55 31 3499-4866 - Fax: +55 31 3499-4850



Desenvolvimento e Análise de Métodos de Reconstrução de Trajetória de Aeronaves

Plínio Marvila Fagundes

Dissertação submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Elétrica.

Orientadores: Leonardo Antônio Borges Tôrres
Luis Antonio Aguirre

Belo Horizonte, 4 de junho de 2007

Agradecimentos

Muito obrigado à minha família: minha mãe e minhas irmãs. Algumas situações em nossas vidas foram tão frustrantes e difíceis que qualquer comportamento diferente do que tivemos seria trágico. Talvez este seja o verdadeiro papel das dificuldades que a vida nos impõe: unir as pessoas que realmente se gostam e torná-las uma família de fato, sem qualquer tipo de desconfiança ou inimizade. Que sejamos sempre assim, integrantes de um grupo forte, harmonioso e bastante feliz.

Muito obrigado à Gisele: uma pessoa realmente especial, sem a qual não teria realizado nem mesmo a metade do que realizei nos últimos anos. Nas horas mais difíceis, o apoio, nas horas felizes, o motivo.

Obrigado à Solution Engenharia de Automação: empresa de grandes profissionais e grandes amigos. Os últimos dois anos foram, sem qualquer tipo de dúvida, o período de maior crescimento profissional e pessoal da minha vida. Difícil imaginar um cenário tão produtivo e repleto de realizações sem a sua participação.

Obrigado aos professores Leonardo Antônio Borges Tôrres e Luis Antonio Aguirre. Falar sobre a competência desses profissionais é tão redundante que talvez seja algum tipo de erro gramatical. Por isso, prefiro agradecer pela amizade que cultivamos e por representarem o que representam para mim.

Finalmente, agradeço a Deus pelo privilégio de conviver com essas pessoas, e nelas enxergar a Sua bondade.

Plínio Marvila Fagundes

Epígrafe

Já não vos chamo servos, porque o servo não sabe o que faz seu senhor. Mas chamei-vos amigos, pois vos dei a conhecer tudo quanto ouvi de meu Pai. Não fostes vós que me escolhestes, mas eu vos escolhi e vos constituí para que vades e produzais fruto, e o vosso fruto permaneça. Eu assim vos constituí, a fim de que tudo quanto pedirdes ao Pai em meu nome, ele vos conceda.

João 15,15-17

Toda a dificuldade pela qual passamos pode ser representada por uma função de várias variáveis. A verdadeira beleza da vida está em constatar que só passamos por eventuais pontos de máximo (locais ou não) se considerarmos apenas a dimensão do tempo. Essa constatação depende de que sejamos suficientemente inteligentes a ponto de perceber (ou aceitar) que há Quem garanta esse comportamento.

Conteúdo

Resumo	xi
Abstract	xiii
Lista de Tabelas	xviii
Lista de Figuras	xxii
Lista de Símbolos	xxiii
Lista de Abreviações	xxvii
1 Introdução	1
1.1 Motivação e justificativa	3
1.2 Relevância	4
1.3 Objetivos	5
1.4 Organização do Texto	6
2 Cinemática de Aeronaves	9
2.1 Aspectos preliminares	10
2.1.1 Representação em espaço de estados	10
2.1.2 Dinâmica de corpos rígidos	11
2.2 Sistemas de coordenadas	12
2.3 Equações de movimento	14
2.3.1 Movimento de Translação	15
2.3.2 Movimento de Rotação	20
2.3.3 Representação nos Eixos do Vento	26
2.3.4 Equações de movimento para <i>Terra Plana</i>	27

3	Reconstrução de Trajetória de Aeronaves	31
3.1	O problema a ser resolvido	33
3.2	As soluções propostas	35
3.3	Sobre os testes realizados	44
4	Métodos de Otimização	51
4.1	Introdução	51
4.2	Otimização multiobjetivo	54
4.3	Busca Direta	57
4.3.1	Método de Nelder-Mead	58
4.4	Direção de Busca	62
4.4.1	Método do Gradiente	63
4.4.2	Método do Gradiente Conjugado	65
4.4.3	Método do Gradiente Conjugado Pré-Condicionado	68
4.5	Métodos de Newton	70
4.5.1	Método de Gauss-Newton	72
4.6	Mínimos Quadrados	73
4.6.1	Método de Levenberg-Marquardt	75
4.7	Região de Confiança	77
4.7.1	Método de Powell (DogLeg)	78
4.8	Algoritmos Genéticos	81
4.8.1	Vector Evaluated Genetic Algorithm	91
4.8.2	Weight-Based Genetic Algorithm	92
4.8.3	Multiple Objective Genetic Algorithm	95
4.8.4	Non-dominated Sorting Genetic Algorithm II	97
4.8.5	Strength Pareto Evolutionary Algorithm	100
4.9	Simulated Annealing (Metropolis)	102
5	A Aplicação Desenvolvida	105
5.1	Estruturação da Aplicação	106
5.2	Funções auxiliares	108
5.3	Algoritmos Determinísticos	110
5.4	Algoritmos Estocásticos	114
5.5	Modelo	124
5.6	Interface Gráfica	136
6	Resultados e Discussão	141
6.1	Qualidade da reconstrução	141
6.1.1	Nelder-Mead, Powell e <i>Simulated Annealing</i>	142
6.1.2	Gradiente e Gradiente Conjugado	154
6.1.3	Gradiente Conjugado Pré-Condicionado e Gauss-Newton	162

6.1.4	Levenberg-Marquardt	169
6.1.5	Algoritmos genéticos	174
6.2	Tempo de execução	198
7	Conclusões e Trabalhos Futuros	201
7.1	Conclusões	201
7.2	Trabalhos Futuros	202
A	Algumas Propriedades Matemáticas	205
A.1	Rotação de coordenadas	205
A.2	Diferenciação de vetores	208
A.3	Gradiente, Jacobiana e Hessiana	211
A.3.1	Particularidades em Mínimos Quadrados	212
A.4	Inclinação e curvatura	213
A.5	Funções lineares e quadráticas	214
A.6	Matrizes	215
A.6.1	Definições	215
A.6.2	Decomposição em valores singulares	217

Resumo

Reconstrução de trajetória de vôo consiste no procedimento utilizado para se inferir a respeito da posição e da atitude de veículos, ao longo do tempo, a partir de um conjunto limitado de medições. No presente trabalho, esse problema é abordado como um problema de determinação de curvas ótimas, sob o ponto de vista da sua aproximação das séries temporais correspondentes medidas a partir de ensaios em vôo de aeronaves de asa fixa, via minimização do erro quadrático médio. As informações disponíveis para a realização desse procedimento são provenientes de unidades de medição inercial (IMU) e de sistemas de posicionamento global (GPS). Como subproduto do procedimento de reconstrução de trajetória, obtêm-se estimativas dos termos de polarização associados às medições de velocidades angulares e acelerações de translação, essenciais no processamento subsequente de dados de ensaio em vôo para fins de modelagem matemática da aeronave. Neste trabalho, apresentam-se treze algoritmos de otimização empregados para reconstrução de trajetória de aeronaves cujo desempenho é avaliado em três cenários distintos determinados pelo tipo de manobra executada, nível de contaminação de ruído e número de variáveis medidas.

Abstract

Flight path reconstruction comprises vehicle position and attitude estimation, during a certain period of time, from a limited set of measurements. In this work this procedure is viewed as an optimization problem related to the search for optimal curves, in a mean square sense, corresponding to the measured time-series, acquired from flight test data of fixed wing aircraft. The information available to execute the procedure corresponds to data from an inertial measurement unit (IMU), together with global positioning system data (GPS). As a by-product of the flight path reconstruction activity the instruments biases are also estimated, which is essential in the posterior aircraft mathematical model identification process. Thirteen optimization algorithms are presented and analysed. Their performance is evaluated in three different scenarios related to maneuver type, noise contamination and number of measured variables.

Lista de Tabelas

1.1	Ilustração e classificação dos algoritmos desenvolvidos.	5
2.1	Descrição dos conjuntos de equações diferenciais que compõem a representação no espaço de estados de um corpo rígido. . . .	14
2.2	Definições e relações vetoriais para equações de movimento . .	17
3.1	Resumo de informações a respeito dos testes realizados.	42
3.2	Descrição dos conjuntos de testes	44
4.1	Ilustração e classificação dos algoritmos desenvolvidos.	53
4.2	Exemplo de atribuição de valores aos vetores de pesos e respectivas variáveis inteiras	93
6.1	Índices de desempenho para o algoritmo de NM, conjunto de testes 1.	142
6.2	Índices de desempenho para o algoritmo de NM, conjunto de testes 2.	143
6.3	Índices de desempenho para o algoritmo de NM, conjunto de testes 3.	143
6.4	Índices de desempenho para o algoritmo de NM, conjunto de testes 4.	144
6.5	Índices de desempenho para o algoritmo de NM, conjunto de testes 5.	144
6.6	Índices de desempenho para o algoritmo de DL, conjunto de testes 1.	146
6.7	Índices de desempenho para o algoritmo de DL, conjunto de testes 2.	147
6.8	Índices de desempenho para o algoritmo de DL, conjunto de testes 3.	147
6.9	Índices de desempenho para o algoritmo de DL, conjunto de testes 4.	148

6.10	Índices de desempenho para o algoritmo de DL, conjunto de testes 5.	148
6.11	Índices de desempenho para o algoritmo de SA, conjunto de testes 1.	150
6.12	Índices de desempenho para o algoritmo de SA, conjunto de testes 2.	151
6.13	Índices de desempenho para o algoritmo de SA, conjunto de testes 3.	151
6.14	Índices de desempenho para o algoritmo de SA, conjunto de testes 4.	152
6.15	Índices de desempenho para o algoritmo de SA, conjunto de testes 5.	152
6.16	Índices de desempenho para o algoritmo de GR, conjunto de testes 1.	154
6.17	Índices de desempenho para o algoritmo de GR, conjunto de testes 2.	155
6.18	Índices de desempenho para o algoritmo de GR, conjunto de testes 3.	155
6.19	Índices de desempenho para o algoritmo de GR, conjunto de testes 4.	156
6.20	Índices de desempenho para o algoritmo de GR, conjunto de testes 5.	156
6.21	Índices de desempenho para o algoritmo de GC, conjunto de testes 1.	158
6.22	Índices de desempenho para o algoritmo de GC, conjunto de testes 2.	158
6.23	Índices de desempenho para o algoritmo de GC, conjunto de testes 3.	159
6.24	Índices de desempenho para o algoritmo de GC, conjunto de testes 4.	160
6.25	Índices de desempenho para o algoritmo de GC, conjunto de testes 5.	160
6.26	Índices de desempenho para o algoritmo de GCP, conjunto de testes 1.	162
6.27	Índices de desempenho para o algoritmo de GCP, conjunto de testes 2.	163
6.28	Índices de desempenho para o algoritmo de GCP, conjunto de testes 3.	163
6.29	Índices de desempenho para o algoritmo de GCP, conjunto de testes 4.	164

6.30	Índices de desempenho para o algoritmo de GCP, conjunto de testes 5.	164
6.31	Índices de desempenho para o algoritmo de GN, conjunto de testes 1.	166
6.32	Índices de desempenho para o algoritmo de GN, conjunto de testes 2.	167
6.33	Índices de desempenho para o algoritmo de GN, conjunto de testes 3.	168
6.34	Índices de desempenho para o algoritmo de GN, conjunto de testes 4.	168
6.35	Índices de desempenho para o algoritmo de LM, conjunto de testes 1.	170
6.36	Índices de desempenho para o algoritmo de LM, conjunto de testes 2.	170
6.37	Índices de desempenho para o algoritmo de LM, conjunto de testes 3.	171
6.38	Índices de desempenho para o algoritmo de LM, conjunto de testes 4.	172
6.39	Índices de desempenho para o algoritmo de LM, conjunto de testes 5.	172
6.40	Índices de desempenho para o algoritmo de VEGA, conjunto de testes 1.	175
6.41	Índices de desempenho para o algoritmo de VEGA, conjunto de testes 2.	175
6.42	Índices de desempenho para o algoritmo de VEGA, conjunto de testes 3.	176
6.43	Índices de desempenho para o algoritmo de VEGA, conjunto de testes 4.	177
6.44	Índices de desempenho para o algoritmo de VEGA, conjunto de testes 5.	177
6.45	Índices de desempenho para o algoritmo de WBGA, conjunto de testes 1.	179
6.46	Índices de desempenho para o algoritmo de WBGA, conjunto de testes 2.	179
6.47	Índices de desempenho para o algoritmo de WBGA, conjunto de testes 3.	180
6.48	Índices de desempenho para o algoritmo de WBGA, conjunto de testes 4.	180
6.49	Índices de desempenho para o algoritmo de WBGA, conjunto de testes 5.	181

6.50	Índices de desempenho para o algoritmo de MOGA, conjunto de testes 1.	182
6.51	Índices de desempenho para o algoritmo de MOGA, conjunto de testes 2.	183
6.52	Índices de desempenho para o algoritmo de MOGA, conjunto de testes 3.	184
6.53	Índices de desempenho para o algoritmo de MOGA, conjunto de testes 4.	184
6.54	Índices de desempenho para o algoritmo de MOGA, conjunto de testes 5.	185
6.55	Índices de desempenho para o algoritmo de NSGA, conjunto de testes 1.	186
6.56	Índices de desempenho para o algoritmo de NSGA, conjunto de testes 2.	187
6.57	Índices de desempenho para o algoritmo de NSGA, conjunto de testes 3.	188
6.58	Índices de desempenho para o algoritmo de NSGA, conjunto de testes 4.	188
6.59	Índices de desempenho para o algoritmo de NSGA, conjunto de testes 5.	189
6.60	Índices de desempenho para o algoritmo de SPEA, conjunto de testes 1.	190
6.61	Índices de desempenho para o algoritmo de SPEA, conjunto de testes 2.	191
6.62	Índices de desempenho para o algoritmo de SPEA, conjunto de testes 3.	192
6.63	Índices de desempenho para o algoritmo de SPEA, conjunto de testes 4.	192
6.64	Índices de desempenho para o algoritmo de SPEA, conjunto de testes 5.	193

Lista de Figuras

2.1	Principais sistemas de coordenadas	13
2.2	Orientação das componentes de velocidade de rotação e translação, forças e momentos externos e ângulos de ataque e de derrapagem em relação ao sistema de coordenadas ABC.	15
3.1	Manobra 1: Asa nivelada e altura aproximadamente constante.	37
3.2	Manobra 2: Asa nivelada e <i>doublet</i> de profundor.	37
3.3	Manobra 3: Rolamento e arfagem negativos seguidos de rolamento e arfagem positivos.	38
3.4	Manobra 3: Rolamento e arfagem negativos seguidos de rolamento e arfagem positivos em detalhe.	38
3.5	Comparação entre as manobras.	39
3.6	Esquema de reconstrução de trajetória via estimação de parâmetros.	43
3.7	Vetores I_1 e $I_{1,\%}$: Espaço de soluções com 2 dimensões.	45
3.8	Índice de desempenho I_2 : Espaço de soluções com 2 dimensões.	46
3.9	Índices de desempenho I_4 e $I_{4,\%}$: Espaço de soluções com 2 dimensões.	47
4.1	Comparação entre ótimo local e ótimo global.	52
4.2	Resolução de problemas conflitantes em otimização multiobjetivo.	55
4.3	Definições e tipos de problemas de otimização multiobjetivo .	56
4.4	Configuração inicial da figura geométrica do algoritmo de Nelder-Mead.	59
4.5	Configuração da figura geométrica após a etapa de <i>reflexão</i> . .	60
4.6	Configuração da figura geométrica após a etapa de <i>expansão</i> . .	60
4.7	Configuração da figura geométrica após a etapa de <i>contração externa</i>	61

4.8	Configuração da figura geométrica após a etapa de <i>contração interna</i>	61
4.9	Configuração da figura geométrica após a etapa de <i>encolhimento</i>	61
4.10	Ortogonalidade entre os vetores gradiente no ponto inicial e no ponto final da busca unidirecional.	64
4.11	Dificuldade de convergência do método do gradiente.	65
4.12	Direções de busca paralelas ao eixo do método do gradiente Conjugado	67
4.13	Direções de busca do método do gradiente Conjugado	67
4.14	Representação de um problema de estimação de parâmetros via ajuste de curvas.	73
4.15	Passo do algoritmo de Powell para $\ \delta\theta_{cp}\ \geq \Delta$	79
4.16	Passo do algoritmo de Powell para $\ \delta\theta_{cp}\ < \Delta$ e $\ \delta\theta_{gn}\ < \Delta$	80
4.17	Passo do algoritmo de Powell para $\ \delta\theta_{cp}\ < \Delta$ e $\ \delta\theta_{gn}\ \geq \Delta$	80
4.18	Fluxograma do princípio de funcionamento geral de um algoritmo genético	84
4.19	Ilustração do procedimento de reprodução via seleção proporcional.	85
4.20	Exemplo de utilização do operador de reprodução via torneio.	86
4.21	Ilustração do cenário de utilização do operador de reprodução via ranqueamento.	87
4.22	Ilustração do procedimento de reprodução via ranqueamento.	88
4.23	Ilustração do algoritmo VEGA.	92
4.24	Influência de α na linearidade da função de partilha.	94
4.25	Operador de Elitismo para o NSGA II	98
4.26	Exemplo de utilização das funções de partilha no SPEA	101
4.27	Fluxograma do princípio de funcionamento geral do algoritmo de <i>Simulated Annealing</i>	104
5.1	Arquitetura e fluxo de informação entre as entidades da aplicação.	106
5.2	Estruturação da aplicação em diretórios.	108
5.3	Biblioteca de blocos para simulação do modelo cinemático.	129
5.4	Blocos de transformação de variáveis medidas.	130
5.5	Bloco para determinação da atitude da aeronave.	131
5.6	Bloco para cálculo do vetor velocidade linear.	131
5.7	Bloco de rotação de coordenadas: NED \rightarrow ABC.	132
5.8	Bloco para cálculo de produto vetorial.	132
5.9	Bloco transformação de estados $[u \ v \ w]^T \rightarrow [V_T \ \alpha \ \beta]^T$	133
5.10	Bloco para determinação do vetor posição da aeronave via rotação de coordenadas ABC \rightarrow NED.	133

5.11	Bloco de rotação de coordenadas: $ABC \rightarrow NED$	134
5.12	Bloco de simulação do modelo cinemático de corpo rígido.	135
5.13	Ambiente de simulação para reconstrução de trajetória e estim- mação de parâmetros.	136
5.14	Interface gráfica: janela principal.	137
5.15	Interface gráfica: seleção do arquivo de dados.	137
5.16	Interface gráfica: busca pelo arquivo de dados.	138
5.17	Interface gráfica: seleção do algoritmo.	138
5.18	Interface gráfica: seleção do cenário de estimação.	139
5.19	Interface gráfica: seleção das variáveis para criação de gráficos de tendência.	139
6.1	Comparação dos índices de desempenho para o algoritmo de NM para diferentes conjuntos de teste.	145
6.2	Comparação dos índices de desempenho para o algoritmo de DL para diferentes conjuntos de teste.	149
6.3	Comparação dos índices de desempenho para o algoritmo de SA para diferentes conjuntos de teste.	153
6.4	Comparação dos índices de desempenho para o algoritmo de GR para diferentes conjuntos de teste.	157
6.5	Comparação dos índices de desempenho para o algoritmo de GC para diferentes conjuntos de teste.	161
6.6	Comparação dos índices de desempenho para o algoritmo de GCP para diferentes conjuntos de teste.	165
6.7	Comparação dos índices de desempenho para o algoritmo de GN para diferentes conjuntos de teste.	169
6.8	Comparação dos índices de desempenho para o algoritmo de LM para diferentes conjuntos de teste.	173
6.9	Comparação dos índices de desempenho para o algoritmo de VEGA para diferentes conjuntos de teste.	178
6.10	Comparação dos índices de desempenho para o algoritmo de WBGA para diferentes conjuntos de teste.	182
6.11	Comparação dos índices de desempenho para o algoritmo de MOGA para diferentes conjuntos de teste.	186
6.12	Comparação dos índices de desempenho para o algoritmo de NSGA para diferentes conjuntos de teste.	190
6.13	Comparação dos índices de desempenho para o algoritmo de SPEA para diferentes conjuntos de teste.	194
6.14	Comparação do índice $I_{4,\%}$ para os algoritmos genéticos em escala.	195

6.15	Resultado da adequação das séries temporais de a_x , a_y e a_z para valores selecionados de termos de polarização.	196
6.16	Resultado da adequação das séries temporais de a_x , a_y e a_z para valores selecionados de termos de polarização em detalhe.	196
6.17	Resultado da adequação das séries temporais de p , q e r para valores selecionados de termos de polarização em detalhe.	197
A.1	Rotação de coordenadas em um plano	207
A.2	Ilustração de movimentos de rotação entre dois sistemas de coordenadas	208

Lista de Símbolos

Neste trabalho, as matrizes são representadas por letras maiúsculas em itálico, os vetores por letras minúsculas em negrito e os escalares por letras minúsculas em itálico. Em todos os casos, utilizam-se letras do alfabeto latino e do alfabeto grego.

Os principais símbolos utilizados nesta dissertação são apresentados a seguir. Caso se trate de um símbolo utilizado para representação de mais de uma variável, apresenta-se a de maior importância para o entendimento geral do texto. Os eventuais demais significados são apresentados no contexto da sua utilização.

\mathbb{R}	Conjunto dos números reais;
\mathbb{Z}	Conjunto dos números inteiros;
$\mathbf{s}(t)$	Série temporal contínua;
$\mathbf{s}[k]$	Série temporal discreta;
s_k	k-ésimo elemento do vetor \mathbf{s} ;
\dot{s}	Derivada de s em função do tempo;
s^m	Medição de s ;
k_s	Ganho relativo à medição distorcida de s ;
Δ_s	Polarização relativa à medição distorcida de s ;
M^T	Transposta da matriz M ;
κ	Número de condicionamento;
$\ \cdot\ $	Norma Euclideana;
$\ \cdot\ _\infty$	Norma-infinito;

\triangleq	Igualdade por definição;
\triangleleft	Indicação de relação de dominância entre soluções;
\mathbf{u}	Vetor de sinais de entrada;
\mathbf{y}	Vetor de sinais de saída;
\mathbf{x}	Vetor de estados;
$\boldsymbol{\theta}$	Vetor de parâmetros;
$\hat{\boldsymbol{\theta}}$	Vetor de parâmetros estimado;
$\boldsymbol{\theta}^*$	Solução ótima;
m	Dimensão do vetor de entrada;
n	Dimensão do vetor de parâmetros;
η	Dimensão do vetor de estados;
N	Número de realizações de uma variável;
∇_f	Vetor gradiente da função f ;
$J_{\mathbf{f}}$	Matriz Jacobiana da função \mathbf{f} ;
G_f	Matriz Hessiana da função f ;
\mathbf{F}	Vetor força resultante ^a ;
\mathbf{t}	Vetor torque resultante;
\mathbf{h}	Vetor momento angular;
\mathbf{g}'	Vetor aceleração da gravidade;
\mathbf{g}_0	Norma Euclideana do vetor aceleração da gravidade;
\mathbf{p}	Vetor posição inercial;
\mathbf{v}	Vetor velocidade de translação;
$\boldsymbol{\varphi}$	Vetor posição angular (atitude);
$\boldsymbol{\omega}$	Vetor velocidade de rotação;
$\mathbf{x}_{\text{VENTO}}$	Vetor de estados nos eixos do vento;

^aExcessão à regra que determina a representação de vetores com letras minúsculas em negrito. Evita confusão com representação de função genérica: $\mathbf{f} = \mathbf{f}(\cdot)$.

x_e	Posição no eixo x do referencial NED;
y_e	Posição no eixo y do referencial NED;
z_e	Posição no eixo z do referencial NED;
u	Velocidade de translação no eixo x ;
v	Velocidade de translação no eixo y ;
w	Velocidade de translação no eixo z ;
a_x	Aceleração no eixo x ;
a_y	Aceleração no eixo y ;
a_z	Aceleração no eixo z ;
ϕ	Posição angular segundo o eixo x (rolamento);
θ	Posição angular segundo o eixo y (arfagem);
ψ	Posição angular segundo o eixo z (guinada);
p	Velocidade de rotação sobre o eixo x ;
q	Velocidade de rotação sobre o eixo y ;
r	Velocidade de rotação sobre o eixo z ;
v_T	Velocidade total nos eixos do vento;
α	Ângulo de ataque (eixos do vento);
β	Ângulo de derrapagem (eixos do vento);

Lista de Abreviações

ODE	Equação diferencial ordinária (<i>Ordinary differential equation</i>);
ECI	Sistema de coordenadas com origem no centro na Terra (<i>Earth-Centered Inertial Frame</i>);
ECEF	Sistema de coordenadas com origem no centro na Terra e de posição constante em relação à Terra (<i>Earth-Centered, Earth-Fixed</i>);
ABC	Sistema de coordenadas dos eixos do corpo da aeronave (<i>Aircraft-Body Coordinate</i>);
NED	Sistema de coordenadas Norte-Leste-Abaixo (<i>North-East-Down</i>);
GPS	Sistema de posicionamento global (<i>Global Positioning System</i>);
IMU	Unidade de medição inercial (<i>Inertial Measurement Unit</i>);
MQ	Mínimos Quadrados;
AG	Algoritmo genético;
c.g.	Centro de gravidade;
NM	Algoritmo de Nelder-Mead;
GR	Algoritmo do Gradiente;
GC	Algoritmo do Gradiente Conjugado;
GCP	Algoritmo do Gradiente Conjugado Pré-Condicionado;
GN	Algoritmo de Gauss-Newton;
LM	Algoritmo de Levenberg-Marquardt;

DL	Algoritmo de Powell (DogLeg);
SA	Algoritmo de <i>Simulated Annealing</i> (Metropolis);
VEGA	<i>Vector Evaluated Genetic Algorithm</i> ;
WBGA	<i>Weight-Based Genetic Algorithm</i> ;
MOGA	<i>Multiple Objective Genetic Algorithm</i> ;
NSGA	<i>Non-dominated Sorting Genetic Algorithm</i> ;
SPEA	<i>Strength Pareto Evolutionary Algorithm</i> ;
FPR	Reconstrução de trajetória de vôo (<i>Flight Path Reconstruction</i>).

INTRODUÇÃO

O problema de estimação de parâmetros pode ser definido de forma bastante simples. Primeiramente, considera-se conhecido um conjunto de equações dinâmicas parametrizadas capazes de descrever o comportamento do sistema investigado. Para a estimação dos parâmetros dessas equações, o sistema é estimulado por sinais de entrada específicos e infere-se a respeito dos seus valores de tal forma que as saídas do conjunto de equações (modelo) sejam, sob algum aspecto, suficientemente próximas das saídas do sistema real.

Entretanto, uma série de dificuldades pode ser encontrada quando se trata de sistemas reais. A primeira delas consiste na impossibilidade de obtenção de medições perfeitas das respostas do sistema. Problemas relacionados à existência de polarização e ruído nos sinais medidos modificam drasticamente o cenário de estimação. Com isso, o problema de estimação passa a ser responsável pela determinação de valores que, em última instância, compõem funções de probabilidade ótimas, sob algum aspecto.

Outra dificuldade consiste na própria discrepância existente entre o sistema real e o modelo identificado. Inevitavelmente, a descrição de um sistema real a partir de um modelo com parâmetros representativos de grandezas físicas não consegue ser perfeita. De fato, o processo de identificação de um modelo tipicamente não tem por objetivo descrever o sistema de forma completa, mas sim apenas os aspectos de interesse. Esse cenário insere uma confiabilidade ainda menor aos valores estimados para o vetor de parâmetros.

Uma grande quantidade de métodos e algoritmos pode ser utilizada para a solução de problemas de estimação de parâmetros. Sua utilização depende basicamente da representação do problema abordado. Aspectos como carac-

terísticas dos sinais de entrada e saída, tipo de modelo utilizado e número de parâmetros a serem estimados têm grande importância na escolha do método a ser aplicado. Além disso, essa escolha sofre grande influência do próprio motivo pelo qual o algoritmo deve ser utilizado.

Em problemas de reconstrução de trajetória, por exemplo, deve-se garantir que os valores estimados sejam capazes de produzir respostas do modelo convenientes no domínio do tempo. Isso quer dizer que não basta que as respostas do modelo e do sistema real sejam parecidas apenas no que diz respeito ao espectro de frequência, por exemplo. Deve-se garantir que as séries temporais sejam parecidas de forma a minimizar algum critério de comparação ponto a ponto.

Em problemas de adequação de curvas (*curve fitting*), deve-se garantir que a resposta do método seja ótima sob algum aspecto. Os algoritmos de reconstrução de trajetória mais comumente utilizados baseiam sua evolução em direção ao ponto ótimo na utilização de regras que garantam características das respostas como mínimo erro quadrático médio, mínima incerteza associada ao parâmetro estimado, etc.

Esses algoritmos podem ser divididos em grupos segundo diversas classificações. Cada grupo de algoritmos tende a oferecer vantagens e desvantagens, o que os torna mais ou menos apropriados para a resolução de determinada classe de problemas. No que diz respeito ao problema de reconstrução de trajetória de aeronaves, por exemplo, deve-se considerar a existência de sinais (ruidosos) provenientes de várias manobras diferentes. Além disso, o número de parâmetros a serem estimados (a dimensão do espaço de busca) também pode ser preponderante na eficiência do algoritmo analisado.

1.1 Motivação e justificativa

Nas últimas décadas, a comunidade científica vem focalizando esforços no sentido de apresentar soluções convenientes para o problema de reconstrução de trajetória. A abrangência desse interesse ultrapassa inclusive os limites de atuação do que se entende por indústria básica, com o desenvolvimento de técnicas de reconstrução de trajetória de sistemas não-lineares “*benchmark*” ou mesmo sistemas caóticos (Ruan, 2005; Meyer and Christensen, 2001). No que diz respeito à reconstrução de trajetória de aeronaves, não apenas a comunidade científica (Mulder et al., 1999) como também a indústria aeronáutica (Mendonça et al., 2005) buscam por alternativas nessa área do conhecimento e da engenharia.

Embora o interesse tenha crescido, ainda são em pequeno número as publicações destinadas à comparação de resultados obtidos com mais de um método de reconstrução de trajetória. De fato, tipicamente apresentam-se apenas os melhores resultados obtidos com um único algoritmo e quase nenhum detalhe a respeito de sua implementação, ou mesmo dos parâmetros utilizados, é apresentado. Além disso, as pesquisas e resultados obtidos na área tendem a se concentrar em experimentos com a utilização de um grupo de algoritmos relativamente restrito (Betts, 1998) ou em experimentos destinados à resolução de problemas específicos (Bar-Itzhack, 2001).

Contudo, a literatura indica uma tendência à diversificação no que diz respeito aos métodos utilizados e, conseqüentemente, à abordagem dada ao problema de reconstrução de trajetória de aeronaves nos últimos anos (Tekinalp and Bingol, 2004), (Yokoyama and Suzuki, 2005). Com isso, entende-se por relevante a apresentação de um trabalho focalizado na análise estatística e comparativa dos métodos mais comumente utilizados. Além disso, também se enfoca a inserção de novos algoritmos ao conjunto dos algoritmos aplicáveis à resolução do problema de reconstrução de trajetória de aeronaves.

1.2 Relevância

Com o aparecimento de avanços significativos no que diz respeito à utilização de determinados algoritmos, torna-se inevitável uma análise mais minuciosa do seu desempenho nos diversos cenários possíveis. Este trabalho apresenta análises de desempenho de métodos selecionados sob dois pontos de vista principais: estatístico e comparativo.

Primeiramente, devem-se caracterizar os métodos utilizados de maneira isolada, de tal forma que sejam evidenciadas suas particularidades no que diz respeito às vantagens e desvantagens de sua utilização. Para isso, deve-se obter uma caracterização estatística do comportamento dos algoritmos de acordo com o cenário de utilização. Essa análise tem grande importância por ser capaz de transformar características intrínsecas dos métodos em particularidades que os tornam mais ou menos apropriados para a resolução do problema em questão.

Uma vez que se tenha caracterizado cada um dos algoritmos segundo aspectos de interesse, torna-se importante obter algum tipo de classificação de acordo com seu desempenho. Essa comparação leva à possibilidade de inferência a respeito dos métodos mais indicados para determinados cenários de estimação / reconstrução. Com isso, pode-se inclusive justificar a predileção da comunidade científica e da indústria por uma determinada classe de métodos.

Finalmente, uma importante contribuição deste trabalho consiste na criação da aplicação gráfica para utilização dos métodos de reconstrução de trajetória selecionados. A disponibilização da aplicação cria novas perspectivas no que diz respeito a investigações na área de reconstrução de trajetória de aeronaves. De fato, provê-se um ambiente de desenvolvimento flexível não somente quanto ao conjunto de dados e aos algoritmos mas também quanto ao próprio modelo a ser utilizado.

1.3 Objetivos

O principal objetivo deste trabalho consiste na abordagem do problema de reconstrução de trajetória de aeronaves sob dois aspectos principais. O primeiro deles consiste na utilização de algoritmos com desempenho comprovado em diversas áreas do conhecimento, mas pouco utilizados para a resolução do problema proposto (Jategaonkar, 2001). O segundo aspecto consiste na análise estatística e comparativa entre os métodos utilizados para reconstrução de trajetória de aeronaves.

Primeiramente, o trabalho tem por objetivo demonstrar a possibilidade de solução do problema proposto com métodos de otimização com utilização, em princípio, não muito difundida para esse fim. Tratam-se de algoritmos amplamente difundidos em diversas áreas do conhecimento, classificados segundo a maneira com que convergem em direção ao ponto ótimo (Tabela 1.3)¹.

Tabela 1.1: Ilustração e classificação dos algoritmos desenvolvidos.

Determinísticos	Busca Direta	Nelder-Mead
	Região de Confiança	DogLeg (Powell)
	Direção de Busca	Gradiente
		Gradiente Conjugado
		Gradiente Conjugado Pré-Condicionado
	Métodos de Newton	Gauss-Newton
Mínimos Quadrados	Levenberg-Marquardt	
Estocásticos	<i>Simulated Annealing</i>	
	Genéticos	<i>Vector Evaluated Genetic Algorithm</i>
		<i>Weight-Based Genetic Algorithm</i>
		<i>Multiple Objective Genetic Algorithm</i>
		<i>Non-dominated Sorting Genetic Algorithm II</i>
		<i>Strength Pareto Evolutionary Algorithm</i>

Nesse contexto, deve-se enfatizar um aspecto importante deste trabalho. Ao contrário do que tipicamente ocorre em trabalhos relacionados, não se tem por objetivo a resolução de problemas de reconstrução de trajetória específicos ou isolados. De fato, busca-se por uma análise ampla de desempenho de algoritmos em cenários diversos com o intuito de fornecer resultados capazes

¹Para descrição detalhada dos métodos apresentados na Tabela 1.3, veja o Capítulo 4.

de auxiliar na escolha de algoritmos dentre os investigados para resolução do problema proposto.

Finalmente, este trabalho tem por objetivo criar uma aplicação (gráfica) capaz de oferecer ferramentas de análise dos sinais disponíveis e principalmente de seleção do algoritmo desejado. Em conjunto com a aplicação propriamente dita, cria-se também uma série de rotinas complementares, responsáveis pela adequação dos sinais disponíveis. Cria-se assim um ambiente de desenvolvimento capaz de oferecer ferramentas para a resolução do problema de reconstrução de trajetória de aeronaves em cenários e de maneiras até mesmo eventualmente não contemplados por este trabalho.

1.4 Organização do Texto

Esta dissertação contém sete capítulos organizados da seguinte forma:

O Capítulo 1 se trata de um capítulo introdutório e apresenta aspectos gerais do projeto e da dissertação em si.

O Capítulo 2 trata da descrição da dinâmica de aeronaves. Procedese com o detalhamento a respeito das equações que compõem o modelo dinâmico com a distinção entre grupos de variáveis relacionadas: movimentos de translação e rotação e variáveis representadas nos eixos do vento.

No Capítulo 3, o problema de reconstrução de trajetória de aeronaves é abordado. Para isso, procede-se com a definição do problema e com a descrição dos principais objetivos da sua resolução. Além disso, definem-se os cenários de reconstrução e as soluções propostas para resolução do problema.

O Capítulo 4 apresenta todos os métodos de otimização implementados. Cada algoritmo é detalhado com o intuito de ressaltar as particularidades relacionadas à sua implementação e à sua abrangência. Os métodos são agrupados segundo classificação relativa ao modo com que convergem em direção ao ponto ótimo.

O Capítulo 5 apresenta a aplicação desenvolvida. O objetivo desse capítulo é detalhar o modo de funcionamento e as funcionalidades da aplicação desenvolvida de acordo com sua concepção. Para isso, procede-se com a apresentação da estruturação da aplicação e com a descrição das rotinas implementadas.

No Capítulo 6, apresentam-se e discutem-se os resultados obtidos com os experimentos realizados. Esse capítulo tem por principal objetivo analisar comparativamente o desempenho dos algoritmos na resolução do problema proposto em conjunto com a apresentação dos resultados propriamente dita.

O Capítulo 7 apresenta conclusões gerais a respeito dos resultados obtidos. Além disso, apresentam-se possibilidades de trabalhos futuros.

Finalmente, o Apêndice A apresenta propriedades e definições matemáticas importantes para o entendimento e utilizadas no desenvolvimento do trabalho.

CINEMÁTICA DE AERONAVES

A construção de modelos matemáticos consiste em etapa fundamental na análise e representação do comportamento de sistemas dinâmicos reais. Um modelo matemático baseado em leis fundamentais da física¹ sugere a metodologia a ser utilizada em seu processo de validação, principalmente no que diz respeito às características dos sinais a serem coletados, provenientes do sistema real. Posteriormente, o modelo pode ser significativamente aprimorado, de modo a adequar-se aos sinais disponíveis.

Um dos importantes aspectos relacionados à identificação de um modelo matemático para aeronaves consiste em sua capacidade de apresentar-se como ferramenta na aferição dos sensores utilizados para navegação. Embora possua limitações (conhecidas), um modelo dinâmico deve ser capaz de descrever aspectos de interesse do comportamento do sistema investigado. Especificamente no que diz respeito à verificação dos sensores, podem-se avaliar parâmetros² responsáveis por medições imperfeitas devido a pequenos erros de alinhamento ou condições externas imprevistas.

A importância da identificação de modelos dinâmicos para esse tipo de sistema também se destaca devido aos altos custos de construção e realização de testes em uma aeronave real. Além dos aspectos relacionados às técnicas de controle de sistemas dinâmicos (Dorf and Bishop, 2001), pode-se argumentar a respeito da distribuição das forças atuantes sobre a aeronave e assim avaliar condições de segurança relacionadas ao eventual desgaste do material que compõe a fuselagem. Além disso, o modelo dinâmico identificado pode ser utilizado em simuladores de voo para treinamento de pilotos em segurança e na análise de desempenho de protótipos.

¹Esse tipo de modelagem é conhecido como *modelagem caixa branca* (Aguirre, 2004).

²Como ganho e polarização, por exemplo.

Este capítulo está organizado como se segue. Primeiramente, apresentam-se na Seção 2.1 aspectos preliminares relativos à representação de sistemas dinâmicos e à caracterização do movimento de um corpo rígido. Em seguida, procede-se com a definição dos sistemas de coordenadas utilizados para navegação de aeronaves na Seção 2.2. Finalmente, a Seção 2.3 apresenta o desenvolvimento das equações de movimento de um corpo rígido.

2.1 Aspectos preliminares

2.1.1 Representação em espaço de estados

O primeiro passo no que diz respeito à modelagem matemática de um sistema dinâmico consiste na escolha do tipo de representação a ser utilizada. Sistemas cuja complexidade exijam a análise de um grande número de variáveis (dentre as quais algumas eventualmente não-observadas) para sua completa caracterização são tipicamente representados por modelos no espaço de estados. Para o caso específico das equações de movimento de corpos rígidos, a representação no espaço de estados assume a seguinte forma geral:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (2.1)$$

em que $\mathbf{x} \in \mathbb{R}^{n \times 1}$ consiste no vetor de estados, $\mathbf{u} \in \mathbb{R}^{m \times 1}$ é o vetor de entradas e $\mathbf{f}(\cdot) : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$ é uma função vetorial tipicamente não-linear dos estados e entradas. Essa equação vetorial representa as n equações diferenciais ordinárias de primeira ordem (ODEs³)

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, \dots, x_n, u_1, \dots, u_m) \\ &\vdots \\ \dot{x}_i &= f_i(x_1, \dots, x_n, u_1, \dots, u_m) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, \dots, x_n, u_1, \dots, u_m), \end{aligned}$$

em que f_1, \dots, f_n representam diferentes funções (possivelmente não-lineares) das n variáveis de estado x_i e das m entradas u_i . Além da equação de estados (2.1), deve-se definir a equação de saída:

$$\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}), \quad (2.2)$$

³ODEs: *Ordinary Differential Equations*.

em que $\mathbf{y} \in \mathbb{R}^{p \times 1}$ consiste no vetor de saídas e $\mathbf{g}(\cdot) : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^p$ representa um vetor de funções (possivelmente não-lineares) dos estados e entradas.

O vetor de estados \mathbf{x} pode ser composto por qualquer conjunto de variáveis capazes de caracterizar completamente o *estado* do sistema. O estado de um sistema define a quantidade de energia armazenada e sua distribuição. Além disso, utilizam-se realizações mínimas⁴ para o vetor de estados, e as variáveis de saída tipicamente correspondem a quantidades para as quais têm-se elementos sensores disponíveis.

2.1.2 Dinâmica de corpos rígidos

Durante o processo de modelagem matemática de um sistema dinâmico, algumas considerações devem ser feitas. No caso específico de aeronaves, considerar o veículo como sendo um corpo rígido simplifica bastante o estudo da sua dinâmica. A classificação de um corpo como rígido implica que todos os pontos da sua estrutura mantêm posições espaciais relativas constantes durante todo o tempo.

Embora a flexão de segmentos da estrutura de uma aeronave (especialmente as asas) seja facilmente notável, até mesmo nas de menor porte, sua classificação como corpo rígido é útil na descrição do estado desse sistema com exatidão suficiente para grande parte dos casos. Além disso, a inclusão de efeitos de flexão da fuselagem implica a necessidade de utilização de equações diferenciais parciais cujas estruturas e parâmetros dependem da realização de uma série de análises significativamente mais complexas que as necessárias para as ODEs de corpos rígidos (Stevens and Lewis, 1992).

As equações de movimento de corpos rígidos podem ser separadas em equações de movimento de translação e rotação, desde que a origem do sistema de coordenadas para sua representação seja o centro de massa do corpo. O movimento de rotação é então caracterizado pelos movimentos de *rolamento*, *arfagem* e *guinada*. Esses movimentos correspondem respectivamente à rotação da estrutura da aeronave sobre os eixos x , y e z do sistema de coordenadas com origem no centro de gravidade (c.g.)⁵ da aeronave.

⁴A representação de um sistema dinâmico no espaço de estados não é única (Chen, 1999). Com isso, algumas representações podem conter variáveis não estritamente necessárias à caracterização do sistema. A presença apenas de variáveis que, simultaneamente, caracterizem completamente o sistema e sejam estritamente necessárias a essa caracterização fazem com que a representação seja classificada como mínima.

⁵Por motivos práticos, centro de gravidade e centro de massa serão utilizados como

As equações dinâmicas restantes para a caracterização do movimento de uma aeronave consistem nas componentes de translação do c.g. Com isso, o modelo matemático consiste em uma representação no espaço de estados com seis graus de liberdade: rotação e translação em x , y e z . Além disso, uma vez que se adote a hipótese de corpo rígido, não se devem considerar graus de liberdade associados a modos de flexão ou vibração de qualquer segmento da estrutura da aeronave.

A escolha das variáveis de estado a serem utilizadas é intuitiva. Embora não seja única, sua divisão em quatro grupos relacionados é a mais comum:

1. três componentes de posição para a representação da energia potencial no campo gravitacional;
2. três componentes de velocidade de translação para a representação da energia cinética de translação;
3. três componentes de velocidade de rotação para a representação da energia cinética de rotação;
4. três componentes de posição angular para a representação da posição (atitude) da aeronave relativa ao vetor gravidade.

2.2 Sistemas de coordenadas

O primeiro passo para o desenvolvimento das equações de movimento de uma aeronave consiste na definição dos sistemas de coordenadas a serem utilizados. No que diz respeito ao referencial inercial, qualquer sistema de coordenadas com velocidade angular nula e não-acelerado em relação à posição média de um conjunto de estrelas “fixas” pode ser utilizado (Halliday et al., 1996a).

Um sistema de coordenadas com origem no centro do sol e orientação constante em relação a um conjunto de estrelas fixas consiste em um referencial inercial quase perfeito. Contudo, um referencial não-ideal com características mais convenientes no que diz respeito à visualização e à análise da trajetória de veículos que se deslocam nas proximidades da superfície da Terra consiste no referencial ECI (*Earth-Centered Inertial Frame*). Esse sistema de coordenadas possui sua origem localizada no centro da Terra, com a qual translada,

sinônimos neste texto.

mas sem alterar sua orientação em relação a um conjunto de estrelas fixas.

A Figura 2.1 ilustra três dos principais sistemas de coordenadas utilizados na descrição e análise da dinâmica de aeronaves. No ponto P , define-se o sistema de coordenadas ABC (*Aircraft-Body Coordinate*). Esse referencial tem como origem o centro de gravidade da aeronave e possui alinhamento fixo em relação a eixos convenientes da fuselagem. O ponto Q consiste na origem do sistema de coordenadas NED (*North-East-Down*), que se move com a aeronave e se localiza verticalmente abaixo do seu c.g.⁶ Essa movimentação é feita de tal modo que o plano $x_{\text{NED}} \times y_{\text{NED}}$ seja sempre tangente à superfície da Terra no ponto Q . Além disso, seus eixos x e y apontam, respectivamente, para o norte e o leste da Terra, enquanto o eixo z aponta para baixo.

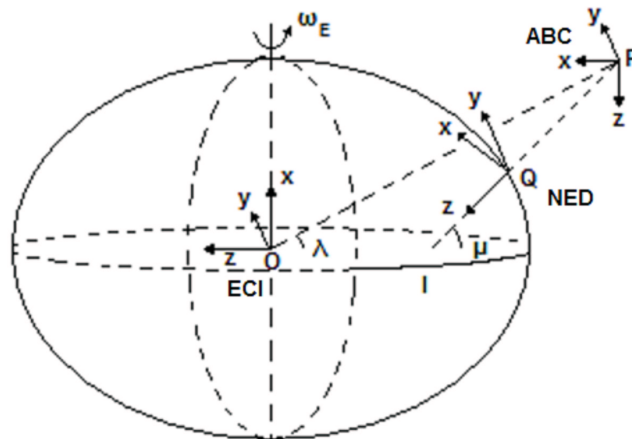


Figura 2.1: Principais sistemas de coordenadas (Stevens and Lewis, 1992) - adaptada.

A orientação dos sistemas de coordenadas ABC e NED em relação ao ECI é modificada à medida que a aeronave se desloca. Por se tratar de um referencial inercial, a orientação do sistema de coordenadas ECI pode ser definida de acordo com a conveniência. Na Figura 2.1 definiu-se o eixo x_{ECI} paralelo ao eixo de rotação da Terra e com sentido para o pólo norte.

⁶Com isso, pretende-se dizer que o vetor peso da aeronave passa pela origem do sistema de coordenadas NED caso a Terra seja modelada como uma esfera perfeita. No caso da Figura 2.1, a excentricidade considerada implica $\lambda \neq \mu$ e impede que o vetor peso da aeronave passe pelo ponto Q .

2.3 Equações de movimento

Conforme descrito na Seção 2.1, o processo de modelagem matemática de uma aeronave consiste na construção de um modelo com seis graus de liberdade: rotação e translação em x , y e z . Nesta Seção, será descrito o procedimento para identificação desse modelo, que consiste na formulação de quatro conjuntos de três equações diferenciais pela análise dos princípios físicos envolvidos na dinâmica de uma aeronave (considerada um corpo rígido).

Com o intuito de destacar a estrutura do modelo a ser identificado, repete-se na Tabela 2.1 a descrição feita na Seção 2.1 a respeito dos referidos conjuntos de equações diferenciais:

Tabela 2.1: Descrição dos conjuntos de equações diferenciais que compõem a representação no espaço de estados de um corpo rígido.

✓ Três componentes de posição para a representação da energia potencial no campo gravitacional: x_e , y_e e z_e ;
✓ Três componentes de velocidade de translação para a representação da energia cinética de translação: u , v e w ;
✓ Três componentes de velocidade de rotação para a representação da energia cinética de rotação: p , q e r ;
✓ Três componentes de posição angular para a representação da posição (atitude) da aeronave relativa ao vetor gravidade: ϕ , θ e ψ .

A Figura 2.2 ilustra parte das variáveis de estado e as forças e momentos que agem sobre a aeronave, tendo como referência o sistema de coordenadas ABC. O significado e a formulação matemática correspondentes à velocidade total V_T , o ângulo de ataque α e o ângulo de derrapagem β serão descritos com o decorrer do desenvolvimento relativo às demais variáveis.

As forças e momentos externos responsáveis pela excitação do modelo de corpo rígido, e conseqüentemente pela movimentação da aeronave em vôo, são causados pela movimentação do veículo em relação à atmosfera. Sua atuação consiste principalmente na produção das forças propulsora, de sustentação e de arrasto. Além disso, em conjunto com a deflexão das superfícies de controle, é responsável pela produção de momentos angulares. A descrição de cada uma das variáveis será feita com o desenvolvimento das equações relacionadas, nas Subseções 2.3.1, 2.3.2 e 2.3.3.

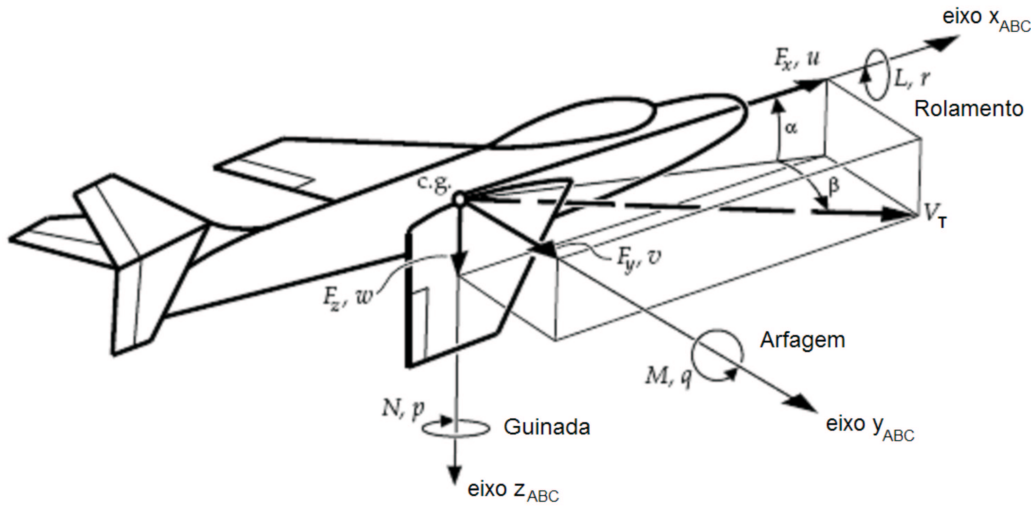


Figura 2.2: Orientação das componentes de velocidade de rotação e translação, forças e momentos externos e ângulos de ataque e de derrapagem em relação ao sistema de coordenadas ABC (Rauw, 2005).

2.3.1 Movimento de Translação

A segunda lei de Newton, aplicada ao movimento de translação, relaciona a aplicação de forças à taxa de alteração de *momentum* linear com o tempo. No que diz respeito à descrição do movimento de translação de aeronaves, as forças consistem na propulsão, nas forças aerodinâmicas e na força de atração da gravidade. Enquanto a propulsão produz forças de reação no corpo da aeronave pela impulsão do ar, as forças aerodinâmicas dependem da sua forma geométrica e do movimento da massa de ar por toda a estrutura da aeronave. Devido a isso, torna-se conveniente representar os vetores de força e velocidade no sistema de coordenadas ABC para a aplicação da segunda lei de Newton.

O movimento de translação de uma aeronave pode então ser representado, de acordo com a segunda lei de Newton, por

$$\mathbf{F}_{ABC} + Bm\mathbf{g} = \frac{d}{dt_{ECI}}(m\mathbf{v}_{abs}) \quad (2.3)$$

em que \mathbf{F}_{ABC} consiste no somatório das forças aerodinâmicas e da propulsão representado no sistema de coordenadas ABC, m é a massa da aeronave, B é uma matriz ortogonal de rotação responsável pela representação da força

gravitacional mg em ABC e \mathbf{v}_{abs} consiste no vetor de velocidade absoluta⁷ do centro de gravidade da aeronave, também representado em ABC.

Importante é observar que, para que a segunda lei de Newton possa ser aplicada, a derivada em (2.3) *deve* ser calculada com respeito a algum referencial inercial (ECI, neste caso)⁸. Considera-se também que a atração da gravidade é devida unicamente à interação entre as massas da aeronave e da Terra.

No sistema de coordenadas ECI, o vetor de posição do centro de gravidade da aeronave será representado pelo *vetor de posição inercial* $\mathbf{p}(t)$. Além disso, uma vez que as forças aerodinâmicas são produzidas pelo deslocamento relativo do ar, torna-se conveniente decompor \mathbf{v}_{abs} em duas componentes distintas: a velocidade da aeronave em relação ao ar e a velocidade absoluta da massa de ar. Assume-se também que o vetor de velocidade angular da massa de ar é exatamente igual ao da Terra⁹. A Tabela 2.2 resume as definições e ilustra relações vetoriais necessárias à aplicação da segunda lei de Newton para construção das equações de movimento de aeronaves.

O vetor de variáveis de estado de uma aeronave possui 12 componentes: 3 componentes do vetor de posição inercial \mathbf{p} , 3 componentes do vetor de velocidade relativa \mathbf{v}_{ABC} , 3 componentes do vetor de velocidade angular $\boldsymbol{\omega}_{\text{ABC}}$ e 3 variáveis relacionadas à atitude da aeronave derivadas das equações diferenciais que descrevem a transformação realizada pela matriz $B(t)$. O primeiro conjunto de equações diferenciais pode ser obtido diretamente com a determinação de $\dot{\mathbf{p}}$ como função de seus componentes

$$\dot{\mathbf{p}} = B^T \mathbf{v}_{\text{abs}} = B^T \mathbf{v}_{\text{ABC}} + \boldsymbol{\omega}_{\text{E}} \times \mathbf{p} \quad (2.4)$$

em que

$$B^T B(\boldsymbol{\omega}_{\text{E}} \times \mathbf{p}) = \boldsymbol{\omega}_{\text{E}} \times \mathbf{p}$$

por $B(t)$ tratar-se de uma matriz ortogonal (Chen, 1999).

Por se tratar da posição do veículo em relação ao referencial ECI, o vetor \mathbf{p} é composto por duas parcelas distintas: uma relativa ao raio da Terra e outra

⁷Neste contexto, o termo “absoluta” denota a medição da velocidade em relação a algum referencial inercial (como o ECI, por exemplo)

⁸Veja a seção A.2.

⁹A velocidade de rotação relativa entre toda a massa de ar e a Terra é nula.

Tabela 2.2: Definições e relações vetoriais para equações de movimento

No sistema de coordenadas ECI	
Posição do centro de gravidade da aeronave:	$\mathbf{p} \in \mathbb{R}^{3 \times 1}$
Velocidade absoluta da aeronave:	$\dot{\mathbf{p}} \in \mathbb{R}^{3 \times 1}$
Velocidade angular absoluta da Terra:	$\boldsymbol{\omega}_E \in \mathbb{R}^{3 \times 1}$
Velocidade absoluta do ar em P (origem de ABC):	$\boldsymbol{\omega}_E \times \mathbf{p}$
Matriz de rotação de ECI para ABC:	$B(t) \in \mathbb{R}^{3 \times 3}$
Aceleração de gravidade da Terra:	$\mathbf{g} \in \mathbb{R}^{3 \times 1}$
No sistema de coordenadas ABC	
Velocidade angular absoluta do referencial ABC:	$\boldsymbol{\omega}_{ABC} \in \mathbb{R}^{3 \times 1}$
Velocidade absoluta da massa de ar:	$B(\boldsymbol{\omega}_E \times \mathbf{p})$
Velocidade relativa do c.g. da aeronave em relação ao ar:	$\mathbf{v}_{ABC} \in \mathbb{R}^{3 \times 1}$
Velocidade absoluta do c.g. da aeronave:	$\mathbf{v}_{abs} \in \mathbb{R}^{3 \times 1}$
Força resultante:	$\mathbf{F}_{ABC} \in \mathbb{R}^{3 \times 1}$
Torque resultante:	$\mathbf{t}_{ABC} \in \mathbb{R}^{3 \times 1}$

relativa à distância entre o veículo e a superfície da Terra. Uma análise comparativa entre os valores dessas duas parcelas mostra que a primeira parcela ($\approx 6,37 \times 10^6$ m) é sensivelmente maior que a segunda ($\approx 30 \times 10^3$ m). Com isso, o módulo do vetor posição se manteria quase constante e igual ao raio da Terra, independentemente da distância da aeronave à superfície.

Essa análise leva à necessidade de se utilizar de algum artifício com o intuito de inserir “sensibilidade” à distância entre a aeronave e a superfície da Terra na equação (2.4). Essa modificação é efetuada com a utilização do sistema de coordenadas NED como referencial inercial, o que dá origem às denominadas equações de movimento em *Terra Plana*. Nessa abordagem, a posição é avaliada com o cálculo de \mathbf{v}_{ABC} no referencial NED e, em seguida, com a integração das suas componentes.

De fato, a consideração da hipótese de *Terra Plana* tem uma importante motivação principal: a simplificação das equações de movimento. Essas novas equações oferecem precisão suficiente para manobras realizadas durante curtos períodos de tempo, que consiste exatamente no cenário de reconstrução de trajetória. Nesses casos, garante-se que a trajetória da aeronave

corresponde a uma pequena área da superfície da Terra e, portanto, sua curvatura pode ser desconsiderada.

A equação (2.4), avaliada nas coordenadas do corpo da aeronave, define a expressão para \mathbf{v}_{abs} a ser utilizada na segunda lei de Newton:

$$\mathbf{v}_{\text{abs}} = \mathbf{v}_{\text{ABC}} + B(\boldsymbol{\omega}_{\text{E}} \times \mathbf{p}). \quad (2.5)$$

Com a substituição da equação (2.5) na equação da segunda lei de Newton (2.3), obtém-se

$$\mathbf{F}_{\text{ABC}} + Bm\mathbf{g} = \dot{m}\mathbf{v}_{\text{abs}} + m\frac{d}{dt_{\text{ECI}}}[\mathbf{v}_{\text{ABC}} + B(\boldsymbol{\omega}_{\text{E}} \times \mathbf{p})]. \quad (2.6)$$

Para a análise da dinâmica de mísseis, o termo \dot{m} tem grande importância. Contudo, no que diz respeito à dinâmica de aeronaves, a taxa de variação de massa com o tempo se torna desprezível. Torna-se suficiente avaliar a dinâmica da aeronave em intervalos de tempo fixos, nos quais o valor de m possa ser considerado constante e, devido a isso, o termo \dot{m} passará a ser desconsiderado.

Desenvolvendo-se a diferenciação indicada em (2.6) com a utilização de (A.10), (A.13) e (A.17) obtém-se

$$\frac{1}{m}\mathbf{F}_{\text{ABC}} + B\mathbf{g} = (\dot{\mathbf{v}}_{\text{ABC}} + \boldsymbol{\omega}_{\text{ABC}} \times \mathbf{v}_{\text{ABC}}) + B(\boldsymbol{\omega}_{\text{E}} \times \dot{\mathbf{p}}). \quad (2.7)$$

Uma vez que se busca pelo conjunto de equações diferenciais que descrevem o comportamento do vetor de velocidade relativa do c.g. da aeronave em relação ao ar \mathbf{v}_{ABC} , deve-se substituir o termo $\dot{\mathbf{p}}$ pelas equações de estado obtidas (2.4). O resultado dessa substituição pode ser simplificado com a utilização da propriedade distributiva do operador de multiplicação vetorial, da propriedade distributiva das matrizes de rotação submetidas a multiplicações vetoriais e da própria ortogonalidade da matriz de rotação $B(t)$ (Chen, 1999). Com o rearranjo dos termos, obtém-se

$$\dot{\mathbf{v}}_{\text{ABC}} = \frac{1}{m}\mathbf{F}_{\text{ABC}} - (\boldsymbol{\omega}_{\text{ABC}} + B\boldsymbol{\omega}_{\text{E}}) \times \mathbf{v}_{\text{ABC}} + B[\mathbf{g} - \boldsymbol{\omega}_{\text{E}} \times (\boldsymbol{\omega}_{\text{E}} \times \mathbf{p})], \quad (2.8)$$

em que a velocidade de translação \mathbf{v}_{ABC} se trata de velocidade relativa, mas a velocidade angular $\boldsymbol{\omega}_{\text{ABC}}$ consiste em uma grandeza absoluta. Devido a isso, torna-se conveniente definir $\boldsymbol{\omega}_{\text{R}}$ como a velocidade angular da aeronave em relação à Terra, conforme

$$\boldsymbol{\omega}_{\text{R}} \triangleq \boldsymbol{\omega}_{\text{ABC}} - B\boldsymbol{\omega}_{\text{E}}. \quad (2.9)$$

Com isso, a equação (2.8) pode ser reescrita como

$$\dot{\mathbf{v}}_{ABC} = \frac{1}{m}\mathbf{F}_{ABC} - (\boldsymbol{\omega}_R + 2B\boldsymbol{\omega}_E) \times \mathbf{v}_{ABC} + B[\mathbf{g} - \boldsymbol{\omega}_E \times (\boldsymbol{\omega}_E \times \mathbf{p})]. \quad (2.10)$$

Embora a equação (2.10) seja consistente, sua utilização muitas vezes não é necessária. Idealmente, o vetor de velocidade angular $\boldsymbol{\omega}_R$ deveria ser utilizado no cálculo dos efeitos aerodinâmicos devidos a movimentos de rotação da aeronave.

Contudo, a utilização de (2.10) não é realmente necessária devido ao fato de os momentos aerodinâmicos serem significativos apenas para velocidades angulares bastante superiores a $\|B\boldsymbol{\omega}_E\|_2$ (Stevens and Lewis, 1992). Essa constatação torna $\boldsymbol{\omega}_R \approx \boldsymbol{\omega}_E$ e, portanto, confirma a não-necessidade da definição de $\boldsymbol{\omega}_R$ e conseqüentemente da utilização de (2.10) em detrimento a (2.8). Além disso, os girômetros comumente utilizados medem velocidade absoluta (componentes do vetor $\boldsymbol{\omega}_{ABC}$) (Doebelin, 1990).

O último termo de (2.8) contém a aceleração gravitacional terrestre \mathbf{g} e a aceleração centrípeta devida à rotação da Terra em relação ao sistema de coordenadas ECI $\boldsymbol{\omega}_E \times \boldsymbol{\omega}_E \times \mathbf{p}$. Devido a isso, observa-se que o vetor

$$\mathbf{g}' \triangleq \mathbf{g} - \boldsymbol{\omega}_E \times (\boldsymbol{\omega}_E \times \mathbf{p}) \quad (2.11)$$

é normal à superfície da Terra¹⁰ e é comumente denominado “*vetor gravidade*”. Sua norma varia entre 9,7829 m/s² em latitude zero e 9,8345 m/s² (igual ao valor de \mathbf{g}) nos pólos.

Finalmente, observa-se que a velocidade angular absoluta da Terra (representada no sistema de coordenadas ABC), $B\boldsymbol{\omega}_E$, é desprezível se comparada com a velocidade angular absoluta do referencial ABC, $\boldsymbol{\omega}_{ABC}$, de uma aeronave. Essa constatação dá origem à versão para *Terra Plana* da equação (2.8):

$$\dot{\mathbf{v}}_{ABC} = \frac{1}{m}\mathbf{F}_{ABC} - \boldsymbol{\omega}_{ABC} \times \mathbf{v}_{ABC} + B\mathbf{g}'_0, \quad (2.12)$$

em que

$$\mathbf{g}'_0 = \begin{bmatrix} 0 \\ 0 \\ g'_0 \end{bmatrix}; \quad g'_0 = \|\mathbf{g}'\| \text{ (para } \lambda = 45^\circ) = 9,8079 \text{ m/s}^2$$

¹⁰De fato, \mathbf{g}' é normal ao elipsóide de referência.

ou, de forma detalhada,

$$\begin{aligned}\dot{u} &= \frac{F_x}{m} - (qw - rv) - g'_0 \sin \theta \\ \dot{v} &= \frac{F_y}{m} - (ru - pw) + g'_0 \cos \theta \sin \phi \\ \dot{w} &= \frac{F_z}{m} - (pv - qu) + g'_0 \cos \theta \cos \phi,\end{aligned}\quad (2.13)$$

em que

$$\mathbf{v}_{ABC} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.14)$$

2.3.2 Movimento de Rotação

Assim como descrito para o movimento de translação, a segunda lei de Newton consiste na relação necessária para determinação da taxa de variação de *momentum* angular com o tempo. Neste caso, relaciona-se a taxa de variação temporal do momento angular \mathbf{h}_{ABC} e o torque \mathbf{t}_{ABC} aplicado sobre o centro de gravidade da aeronave:

$$\mathbf{t}_{ABC} = \frac{d}{dt_{ECI}}(\mathbf{h}_{ABC}), \quad (2.15)$$

em que \mathbf{t}_{ABC} é gerado pelas superfícies de controle aerodinâmico ou por componentes de propulsão com ação fora do c.g. da aeronave, por exemplo.

Mas \mathbf{h}_{ABC} pode ser definido como uma função matricial linear dos termos de inércia e dos componentes do vetor velocidade angular $\boldsymbol{\omega}_{ABC}$ (Halliday et al., 1996b), tal que

$$\mathbf{h}_{ABC} = J\boldsymbol{\omega}_{ABC}, \quad (2.16)$$

em que J consiste na *matriz de inércia* do corpo rígido.

Com isso, uma vez que se tenha expressado o momento angular como função da matriz de inércia e da velocidade de rotação (2.16), pode-se aplicar a segunda lei de Newton ao movimento de rotação de um corpo rígido (2.15). Para isso, procede-se com a diferenciação segundo o teorema de Coriolis (A.13) e a matriz de inércia J é considerada constante, o que resulta em

$$\mathbf{t}_{ABC} = \frac{d}{dt_{ECI}}(\mathbf{h}_{ABC}) = \dot{\mathbf{h}}_{ABC} + \boldsymbol{\omega}_{ABC} \times \mathbf{h}_{ABC}$$

e, com isso,

$$J\dot{\boldsymbol{\omega}}_{ABC} = -\boldsymbol{\omega}_{ABC} \times (J\boldsymbol{\omega}_{ABC}) + \mathbf{t}_{ABC}, \quad (2.17)$$

o que implica

$$\dot{\boldsymbol{\omega}}_{ABC} = -J^{-1}(\boldsymbol{\omega}_{ABC} \times (J\boldsymbol{\omega}_{ABC})) + J^{-1}\mathbf{t}_{ABC}.$$

De fato, a matriz de inércia de uma aeronave pode sofrer variações repentinas, caso parte da carga seja lançada em vôo, ou graduais, devido ao consumo de combustível, por exemplo.

A equação (2.17) consiste na relação básica para avaliação da velocidade angular. Essa equação relaciona as derivadas das taxas de variação de velocidade angular do eixo do corpo com as próprias taxas de variação de velocidade angular do eixo do corpo (p , q e r) e com os componentes do torque exercido na aeronave. A equação (2.17) consiste no terceiro conjunto de equações diferenciais para descrição da dinâmica de corpos rígidos (Seção 2.1).

A ocorrência da inversa da matriz de inércia em (2.17) estimula a investigação a respeito do valor de J^{-1} . Devido à simetria de J , sua inversa pode ser escrita como

$$J^{-1} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_2 & k_4 & k_5 \\ k_3 & k_5 & k_6 \end{bmatrix} \quad (2.18)$$

em que

$$\begin{aligned} k_1 &= \frac{J_{yy}J_{zz} - J_{yz}^2}{\Delta} & k_2 &= \frac{J_{yz}J_{zx} + J_{xy}J_{zz}}{\Delta} \\ k_3 &= \frac{J_{xy}J_{yz} + J_{zx}J_{yy}}{\Delta} & k_4 &= \frac{J_{zz}J_{xx} - J_{zx}^2}{\Delta} \\ k_5 &= \frac{J_{xy}J_{zx} + J_{yz}J_{xx}}{\Delta} & k_6 &= \frac{J_{xx}J_{yy} - J_{xy}^2}{\Delta} \end{aligned}$$

e

$$\Delta = J_{xx}J_{yy}J_{zz} - 2J_{xy}J_{yz}J_{zx} - J_{xx}J_{yz}^2 - J_{yy}J_{zx}^2 - J_{zz}J_{xy}^2$$

Felizmente, na imensa maioria dos casos, nem todos os termos da matriz de inércia (ou de sua inversa) são necessários. A justificativa para essa afirmação encontra-se na constatação de que, tipicamente, o plano $x-z$ consiste em um plano de simetria da aeronave¹¹. Com isso, dentre todos os possíveis produtos cruzados de inércia, apenas o valor de $\int xz dm = J_{xz}$ é diferente de zero. Além disso, existe um conjunto de eixos, denominados *eixos principais*

¹¹(Travassos et al., 1980) trata da análise da dinâmica de aeronaves em que a simetria em relação ao plano $x-z$ não existe.

de inércia (Stevens and Lewis, 1992), que podem ser escolhidos de modo a garantir a característica diagonal da matriz de inércia J , independentemente da particularidade citada.

Considerando-se a simetria em relação ao plano $x - z$, obtém-se

$$J = \begin{bmatrix} J_x & 0 & -J_{xz} \\ 0 & J_y & 0 \\ -J_{xz} & 0 & J_z \end{bmatrix}; \quad J^{-1} = \frac{1}{\Gamma} \begin{bmatrix} J_z & 0 & J_{xz} \\ 0 & \frac{\Gamma}{J_y} & 0 \\ J_{xz} & 0 & J_x \end{bmatrix}, \quad (2.19)$$

em que $\Gamma = J_x J_z - J_{xz}^2$ e os subíndices duplos foram substituídos por simplicidade.

Com a simplificação da matriz de inércia (2.19), a equação (2.17) pode ser expandida e dar origem ao conjunto de três equações diferenciais que descrevem a evolução temporal dos componentes da velocidade de rotação do corpo rígido. Com a utilização de (A.16) na substituição da operação “ $\omega_{ABC} \times$ ” por

$$\Omega_{ABC} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix},$$

obtém-se

$$\begin{aligned} \Gamma \dot{p} &= J_{xz}[J_x - J_y + J_z]pq - [J_z(J_z - J_y) + J_{xz}^2]qr + J_z \bar{L} + J_{xz} N \\ J_y \dot{q} &= (J_z - J_x)pr - J_{xz}(p^2 - r^2)M \\ \Gamma \dot{r} &= [(J_x - J_y)J_x + J_{xz}^2]pq - J_{xz}[J_x - J_y + J_z]qr + J_{xz} \bar{L} + J_x N, \end{aligned} \quad (2.20)$$

em que

$$\mathbf{t}_{ABC} = \begin{bmatrix} \bar{L} \\ M \\ N \end{bmatrix}. \quad (2.21)$$

As equações (2.20) são fortemente acopladas e apresentam o denominado *efeito giroscópico*. Essa característica implica que a rotação sobre quaisquer dois eixos do corpo da aeronave produz aceleração angular sobre o terceiro eixo. Essa constatação pode ser observada com a análise direta da parcela que multiplica qr , pr e pq nas equações de \dot{p} , \dot{q} e \dot{r} , respectivamente¹².

¹²Por exemplo: $q \neq 0$ e $r \neq 0$ implicam $\dot{p} \neq 0$, na primeira equação de (2.20).

Representação da posição angular - atitude

Os sistemas de coordenadas utilizados para representação da posição de uma aeronave são, tipicamente, os representados na Figura 2.1. As orientações dos sistemas de coordenadas NED e ABC são determinadas pelas seqüências de rotações apresentadas a seguir.

ECI para NED: Um vetor representado no sistema de coordenadas ECI pode ser representado no sistema de coordenadas NED com a execução de rotações segundo os ângulos de latitude μ e longitude l :

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{NED}} = B_{\mu} B_l \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{ECI}},$$

em que B_l é responsável pela primeira rotação: positiva¹³ em torno do eixo x e B_{μ} é responsável pela segunda rotação: negativa em torno do eixo y . Devido a isso, B_l mantém o eixo x_{ECI} constante e B_{μ} mantém o novo eixo $y = y_{\text{NED}}$ inalterado:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{NED}} = \begin{bmatrix} \cos \mu & 0 & \sin \mu \\ 0 & 1 & 0 \\ -\sin \mu & 0 & \cos \mu \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos l & \sin l \\ 0 & -\sin l & \cos l \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\text{ECI}}.$$

Convencionou-se B_G como sendo a matriz de transformação do sistema de coordenadas inercial para o sistema de coordenadas geográfico:

$$B_G = B_{\mu} B_l = \begin{bmatrix} \cos \mu & 0 & \sin \mu \\ 0 & 1 & 0 \\ -\sin \mu & 0 & \cos \mu \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & \cos l & \sin l \\ 0 & -\sin l & \cos l \end{bmatrix}. \quad (2.22)$$

NED para ABC: A seqüência de rotações para a representação de um vetor no sistema de coordenadas do corpo da aeronave (ABC) inicialmente representado em um sistema de coordenadas de referência (NED) consiste em¹⁴:

1. rotação sobre o eixo z_{NED} : guinada positiva ψ ;
2. rotação sobre o novo eixo y : arfagem positiva θ ;

¹³Por convenção, rotações positivas serão tratadas como rotações segundo a “regra da mão direita”.

¹⁴Neste ponto, vale lembrar o fato de que a orientação do sistema de coordenadas ABC é ilustrada na Figura 2.2.

3. rotação sobre o novo eixo x : rolamento positivo ϕ .

Devido a isso, a transformação de um vetor originalmente representado no sistema de coordenadas NED em um vetor representado em ABC consiste na multiplicação, à esquerda, consecutiva de matrizes relacionadas à guinada, arfagem e rolamento (nessa ordem):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ABC} = B_\phi B_\theta B_\psi \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED}.$$

Segundo as definições da seção A.1 e a Figura A.1, as matrizes de rotação B_ϕ , B_θ e B_ψ podem ser definidas de modo que

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ABC} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \times \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \times \\ \times \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{NED},$$

em que os ângulos de rolamento (ϕ), arfagem (θ) e guinada (ψ) são referidos como *ângulos de Euler*.

Convenciona-se B_B como sendo a matriz de transformação do sistema de coordenadas geográfico para o sistema de coordenadas do corpo da aeronave:

$$B_B = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\ \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & \cos \phi \cos \theta \end{bmatrix}. \quad (2.23)$$

Finalmente, define-se a matriz $B(t)$ como sendo a matriz de transformação do sistema de coordenadas ECI para o sistema de coordenadas ABC:

$$B = B_B B_G, \quad (2.24)$$

o que implica

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ABC} = B(t) \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECI}.$$

A matriz $B(t)$ é a responsável pela representação da variação temporal da atitude da aeronave e depende dos ângulos de latitude e longitude (μ e l), calculados a partir de \mathbf{p} , e dos ângulos de Euler.

Equações cinemáticas de rotação

As velocidades angulares do sistema de coordenadas ABC sobre seus eixos (p , q e r) dão origem, portanto, à variação da atitude da aeronave a cada instante de tempo. A relação existente entre o vetor velocidade de rotação do eixo ABC e a atitude da aeronave corresponde à equação *Strapdown* (A.14). Isso implica que

$$\dot{B} = -\Omega B, \quad (2.25)$$

em que

$$\Omega = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}. \quad (2.26)$$

A equação *Strapdown* (2.25) resulta em um conjunto de nove equações diferenciais. Contudo, essas equações são redundantes e devem ser transformadas em um conjunto de apenas três equações diferenciais, uma vez que a cinemática relacionada à atitude da aeronave pode ser descrita pela evolução temporal dos três ângulos de Euler apenas¹⁵. Com esse objetivo, a equação (2.25) pode ser reescrita em termos das colunas de B :

$$\dot{\mathbf{b}}_i = -\boldsymbol{\omega} \times \mathbf{b}_i, \quad i = 1,2,3. \quad (2.27)$$

Utilizando-se o primeiro e o segundo elementos de $\dot{\mathbf{b}}_3$ e o primeiro elemento de $\dot{\mathbf{b}}_2$ em (2.27) obtém-se

$$\frac{d}{dt} \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} 1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad (2.28)$$

que consiste no conjunto de três equações diferenciais para a representação da evolução temporal da atitude da aeronave. De forma mais direta:

$$\begin{aligned} \dot{\phi} &= p + q \tan \theta \sin \phi + r \tan \theta \cos \phi \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= q \sin \phi / \cos \theta + r \cos \phi / \cos \theta \end{aligned} \quad (2.29)$$

ou, de forma compacta:

$$\dot{\boldsymbol{\varphi}} = \mathcal{E}(\boldsymbol{\varphi})\boldsymbol{\omega}, \quad (2.30)$$

em que $\boldsymbol{\varphi} = [\phi \ \theta \ \psi]^T$.

O conjunto de equações (2.29) será utilizado em substituição às equações em (2.25).

¹⁵Embora essa não seja a única forma de representação da atitude de uma aeronave (Stevens and Lewis, 1992).

2.3.3 Representação nos Eixos do Vento

Embora a dinâmica de uma aeronave possa ser caracterizada segundo as variáveis de estado descritas na Tabela 2.1, essa representação tipicamente não é conveniente. Motivada principalmente pela possibilidade de medição, a alternativa utilizada consiste na substituição de \mathbf{v}_{ABC} por um novo vetor de estados representado no sistema de coordenadas definido pelos eixos do vento. Esse vetor é formado pela velocidade total V_T , ângulo de ataque α e ângulo de derrapagem β de tal forma que

$$\mathbf{x}_{VENTO} = \begin{bmatrix} V_T \\ \alpha \\ \beta \end{bmatrix} \quad (2.31)$$

e seja resultado da rotação do sistema de coordenadas ABC até o sistema de coordenadas definido pelos eixos do vento. Com isso, devem-se definir duas matrizes S_α e S_β responsáveis pela rotação de ABC pelos *ângulos aerodinâmicos* α e β .

Os ângulos de ataque e derrapagem são definidos de acordo com a rotação do sistema de coordenadas ABC em torno de y_{ABC} e, em seguida, em torno do novo eixo z_{EST} (respectivamente). As rotações são feitas de maneira que o novo eixo x_{VENTO} esteja alinhado com vetor da velocidade do vento relativo (Figura 2.2). Além disso, o eixo z recebe essa nomenclatura por compor o sistema de coordenadas denominado *sistema de estabilidade*.

Os sinais de α e β são definidos de tal forma que o ângulo de ataque seja positivo caso a rotação em torno de y_{ABC} seja negativa e o ângulo de derrapagem seja positivo caso a rotação em torno de z_{EST} seja positiva. Com isso, as rotações são definidas por

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{EST} = S_\alpha \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ABC} \quad ; \quad S_\alpha = \begin{bmatrix} \cos \alpha & 0 & \text{sen } \alpha \\ 0 & 1 & 0 \\ -\text{sen } \alpha & 0 & \cos \alpha \end{bmatrix} \quad (2.32)$$

e

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{VENTO} = S_\beta \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{EST} \quad ; \quad S_\beta = \begin{bmatrix} \cos \beta & \text{sen } \beta & 0 \\ -\text{sen } \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.33)$$

Com isso,

$$\mathbf{x}_{VENTO} = S_\beta S_\alpha \mathbf{v}_{ABC} = S \mathbf{v}_{ABC} \quad (2.34)$$

em que

$$S = \begin{bmatrix} \cos \alpha \cos \beta & \sin \beta & \sin \alpha \cos \beta \\ -\cos \alpha \sin \beta & \cos \beta & -\sin \alpha \sin \beta \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}, \quad (2.35)$$

o que implica

$$\mathbf{v}_{ABC} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = S^T \mathbf{v}_{VENTO} = S^T \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} V_T \cos \alpha \cos \beta \\ V_T \sin \beta \\ V_T \sin \alpha \cos \beta \end{bmatrix}. \quad (2.36)$$

Obtém-se então a relação de transformação

$$\begin{aligned} V_T &= \sqrt{u^2 + v^2 + w^2} \\ \tan \alpha &= w/u \\ \tan \beta &= v/v_T, \end{aligned} \quad (2.37)$$

que cria o novo conjunto de variáveis representadas nos eixos de vento (v_T , α e β) a partir das componentes de velocidade de translação do c.g. da aeronave (u , v e w).

2.3.4 Equações de movimento para *Terra Plana*

Com a análise do desenvolvimento descrito nas Subseções 2.3.1, 2.3.2 e 2.3.3, observa-se a possibilidade de uma aproximação importante: consideração da superfície da Terra como sendo plana. Com isso, pode-se considerar o sistema de coordenadas NED como sendo um referencial inercial, o que dá origem a um conjunto de equações significativamente mais simples.

Embora o sistema de coordenadas NED apresente velocidade de rotação e aceleração não nulas em relação ao ECI, as acelerações associadas à Terra são desprezíveis se comparadas com as produzidas por manobras de uma aeronave. Com isso, pode-se considerar (2.12) como a equação¹⁶ para $\dot{\mathbf{v}}_{ABC}$ e a equação para $\dot{\mathbf{p}}$ (2.4) pode ser substituída por uma versão simplificada:

$$\dot{\mathbf{p}}_{NED} \triangleq \mathbf{v}_{NED} = B_B^T \mathbf{v}_{ABC} \quad (2.38)$$

ou, de forma detalhada,

$$\begin{aligned} \dot{x}_e &= u \cos \theta \cos \psi + v (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) + \\ &\quad + w (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \dot{y}_e &= u \cos \theta \sin \psi + v (\sin \phi \sin \theta \sin \psi \cos \phi \cos \psi) + \\ &\quad + w (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ \dot{z}_e &= u \sin \theta - v \sin \phi \cos \theta - w \cos \phi \cos \theta, \end{aligned}$$

¹⁶Conforme descrito na Subseção 2.3.1.

em que \mathbf{p}_{NED} consiste no vetor posição tomando-se o sistema de coordenadas geográfico (NED) como referência.

Finalmente, as equações de movimento para Terra plana, segundo (2.12), (2.17), (2.30) e (2.38) são:

$$\begin{aligned}
 \dot{\mathbf{v}}_{\text{ABC}} &= -\Omega_{\text{ABC}}\mathbf{v}_{\text{ABC}} + B_B\mathbf{g}'_0 + \frac{\mathbf{F}_{\text{ABC}}}{m} \Rightarrow \text{equação de força} \\
 \dot{\boldsymbol{\omega}}_{\text{ABC}} &= -J^{-1}\Omega_{\text{ABC}}J\mathbf{v}_{\text{ABC}} + J^{-1}\mathbf{t}_{\text{ABC}} \Rightarrow \text{equação de momento} \\
 \dot{\boldsymbol{\varphi}} &= \mathcal{E}(\boldsymbol{\varphi})\boldsymbol{\omega} \Rightarrow \text{equação de atitude} \\
 \dot{\mathbf{p}}_{\text{NED}} &= B_B^T\mathbf{v}_{\text{ABC}} \Rightarrow \text{equação de navegação}
 \end{aligned} \tag{2.39}$$

De forma detalhada, as equações (2.39) se tornam:

Equação de força:

$$\begin{cases} \dot{u} = F_x/m - (qw - rv) - g'_0 \sin \theta \\ \dot{v} = F_y/m - (ru - pw) + g'_0 \cos \theta \sin \phi \\ \dot{w} = F_z/m - (pv - qu) + g'_0 \cos \theta \cos \phi \end{cases}$$

Equação de momento:

$$\begin{cases} \Gamma \dot{p} = J_{xz}[J_x - J_y + J_z]pq - [J_z(J_z - J_y) + J_{xz}^2]qr + J_z\bar{L} + J_{xz}N \\ J_y \dot{q} = (J_z - J_x)pr - J_{xz}(p^2 - r^2)M \\ \Gamma \dot{r} = [(J_x - J_y)J_x + J_{xz}^2]pq - J_{xz}[J_x - J_y + J_z]qr + J_{xz}\bar{L} + J_xN \end{cases}$$

Equação de atitude:

$$\begin{cases} \dot{\phi} = p + q \sin \theta \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} = q \cos \phi - r \sin \phi \\ \dot{\psi} = q \sin \phi \sec \theta + r \cos \phi \sec \theta \end{cases}$$

Equação de navegação:

$$\begin{cases} \dot{x}_e = u \cos \theta \cos \psi + v (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) + \\ \quad + w (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \dot{y}_e = u \cos \theta \sin \psi + v (\sin \phi \sin \theta \sin \psi \cos \phi \cos \psi) + \\ \quad + w (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ \dot{z}_e = u \sin \theta - v \sin \phi \cos \theta - w \cos \phi \cos \theta \end{cases}$$

Com isso, o vetor de estado se torna

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_{\text{ABC}} \\ \boldsymbol{\omega}_{\text{ABC}} \\ \boldsymbol{\varphi} \\ \mathbf{p}_{\text{NED}} \end{bmatrix}, \tag{2.40}$$

em que

$$\mathbf{v}_{ABC} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}; \boldsymbol{\omega}_{ABC} = \begin{bmatrix} p \\ q \\ r \end{bmatrix}; \boldsymbol{\varphi} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}; \mathbf{p}_{NED} = \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix}. \quad (2.41)$$

RECONSTRUÇÃO DE TRAJETÓRIA DE AERONAVES

A resolução do problema de reconstrução de trajetória se baseia primordialmente na utilização conjunta do modelo cinemático e de medições obtidas a partir de instrumentos instalados na aeronave. Nesse cenário, os valores assumidos pelo vetor de estado com o decorrer do tempo definem a trajetória¹ da aeronave. Além disso, define-se por *estimação de estado* ou *reconstrução de estado* o cálculo desses valores a partir das medições provenientes da instrumentação a bordo.

A relevância da utilização de técnicas de reconstrução de trajetória se torna evidente quando um importante aspecto é observado. Uma vez que o vetor de estado tenha sido reconstruído, seus valores podem ser utilizados na identificação do modelo aerodinâmico². De fato, o vetor reconstruído auxilia não somente na determinação da estrutura do modelo aerodinâmico como, principalmente, na estimação de parâmetros de desempenho, estabilidade e controle da aeronave.

Inicialmente, as técnicas de reconstrução de trajetória de aeronaves eram utilizadas com o objetivo de determinar os valores de variáveis de difícil medição. Com o aprimoramento da instrumentação instalada, tende-se a utilizar essas técnicas mais freqüentemente com o intuito de corrigir medições distorcidas do que de efetivamente torná-las disponíveis (Mulder and Hollan-

¹No espaço de estados, portanto.

²Nesse caso, deve-se destacar o fato de que a estrutura do modelo aerodinâmico depende da aeronave e da abordagem utilizada. As modificações na estrutura do modelo aerodinâmico são observadas principalmente com a adição ou remoção de coeficientes aerodinâmicos e respectivas correções.

der, 1981).

Nesse contexto, surge uma constatação natural. Se a técnica de reconstrução de trajetória é capaz de corrigir medições imprecisas, então a comparação do resultado da reconstrução dessas variáveis com as medições propriamente ditas oferece indícios dos motivos pelos quais as medições foram imperfeitas (Gerlach, 1970). Com isso, delimita-se uma nova área de atuação das técnicas de reconstrução de trajetória de aeronaves: estimação de parâmetros de erro da instrumentação instalada. E este é um dos enfoques deste trabalho.

Além dessa abordagem, as técnicas de reconstrução de trajetória são comumente utilizadas nas seguintes áreas de pesquisa e aplicações:

- determinação de componentes da velocidade do vento a partir de sinais obtidos em grandes altitudes com turbulência (Parks et al., 1982);
- suavização das curvas das variáveis de estado estimadas - com período de amostragem, em princípio, igual ao dos sinais provenientes de GPS (Mulder et al., 1999);
- testes em aeronaves de alto desempenho - grandes ângulos de ataque e manobras rápidas (Taylor, 1982);
- análise de acidentes (Wingrove and Bach, 1983);
- modelagem estrutural de aeronaves (Klein and Batterson, 1986).

3.1 O problema a ser resolvido

Técnicas de reconstrução de trajetória de aeronaves cumprem um importante papel na validação dos instrumentos de medição utilizados em protótipos. Além de possibilitar a inferência a respeito da qualidade dos sinais medidos, a técnica é capaz de fornecer indícios a respeito das fontes de erro responsáveis pelas medições inadequadas.

Nesse contexto, os erros provenientes de eventuais desalinhamentos existentes entre girômetros e acelerômetros da Unidade de Medição Inercial (IMU³) ganham relevância. Essa afirmativa se deve principalmente ao fato de a_x, a_y, a_z, p, q e r serem consideradas variáveis de entrada do modelo cinemático, conforme as equações (3.1), (3.2) e (3.3)⁴.

$$\begin{cases} \dot{u} = a_x^m - (q^m w - r^m v) \\ \dot{v} = a_y^m - (r^m u - p^m w) \\ \dot{w} = a_z^m - (p^m v - q^m u) \end{cases} \quad (3.1)$$

$$\begin{cases} \dot{\phi} = p^m + q^m \sin \theta \tan \theta + r^m \cos \phi \tan \theta \\ \dot{\theta} = q^m \cos \phi - r^m \sin \phi \\ \dot{\psi} = q^m \sin \phi \sec \theta + r^m \cos \phi \sec \theta \end{cases} \quad (3.2)$$

$$\begin{cases} \dot{x}_e = u \cos \theta \cos \psi + v (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) + \\ \quad + w (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \\ \dot{y}_e = u \cos \theta \sin \psi + v (\sin \phi \sin \theta \sin \psi \cos \phi \cos \psi) + \\ \quad + w (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \\ \dot{z}_e = u \sin \theta - v \sin \phi \cos \theta - w \cos \phi \cos \theta \end{cases} \quad (3.3)$$

Esse contexto implica a existência de um vetor de estados de tamanho 9 definido por

$$\mathbf{x} = [u, v, w, \phi, \theta, \psi, x_e, y_e, z_e]^T, \quad (3.4)$$

um vetor de entradas de tamanho 6 definido por

$$\mathbf{u} = [p^m, q^m, r^m, a_x^m, a_y^m, a_z^m]^T \quad (3.5)$$

em que o sobrescrito “ m ” indica *variável medida*. Neste trabalho, as variáveis medidas são modeladas de tal maneira que seus valores sejam iguais ao valor

³IMU: *Inertial Measurement Unit*.

⁴Veja Tabela 2.1 na página 14 e equações (2.39) na página 28.

real da respectiva variável multiplicado por um termo de ganho e, em seguida, somado a um termo de polarização. Com isso:

$$\begin{aligned}
 a_x^m &= k_{a_x} a_x + \Delta a_x \\
 a_y^m &= k_{a_y} a_y + \Delta a_y \\
 a_z^m &= k_{a_z} a_z + \Delta a_z \\
 p^m &= k_p p + \Delta p \\
 q^m &= k_q q + \Delta q \\
 r^m &= k_r r + \Delta r
 \end{aligned} \tag{3.6}$$

o que implica um vetor de parâmetros de tamanho 12 definido por

$$\theta = [k_p, \Delta p, k_q, \Delta q, k_r, \Delta r, k_{a_x}, \Delta a_x, k_{a_y}, \Delta a_y, k_{a_z}, \Delta a_z]^T \tag{3.7}$$

e um vetor de saídas típico, de tamanho 6, definido por

$$\mathbf{y} = [x_e, y_e, z_e, v_T, \alpha, \beta]^T \tag{3.8}$$

em que, segundo (2.37),

$$\begin{cases}
 v_T &= \sqrt{u^2 + v^2 + w^2} \\
 \alpha &= \tan^{-1}(w/u) \\
 \beta &= \tan^{-1}(v/v_T).
 \end{cases}$$

3.2 As soluções propostas

Neste trabalho, o problema de reconstrução de trajetória é resolvido de forma indireta, via estimação dos termos de distorção das variáveis de entrada do modelo cinético. Para isso, decidiu-se por se utilizar um total de treze algoritmos de otimização (Capítulo 4). Dentre eles, sete consistem em algoritmos determinísticos:

1. Nelder-Mead;
2. DogLeg (Powell);
3. Gradiente;
4. Gradiente Conjugado;
5. Gradiente Conjugado Pré-Condicionado;
6. Gauss-Newton;
7. Levenberg-Marquardt,

e seis em algoritmos estocásticos:

8. *Simulated Annealing*;
9. *Vector Evaluated Genetic Algorithm*;
10. *Weight-Based Genetic Algorithm*;
11. *Multiple Objective Genetic Algorithm*;
12. *Non-dominated Sorting Genetic Algorithm II*;
13. *Strength Pareto Evolutionary Algorithm*,

em que os cinco últimos se tratam de algoritmos genéticos.

O ambiente de estimação/reconstrução criado com a implementação da aplicação (Capítulo 5) oferece a possibilidade de utilização de diversos cenários de estimação. Esses cenários podem ser diferenciados segundo os seguintes aspectos:

- algoritmo selecionado;
- dados provenientes da manobra desejada;

- número de variáveis observadas (consideradas saídas do modelo cinemático);
- número de parâmetros estimados;
- modelo utilizado.

Com o intuito de atingir os objetivos propostos, a resolução do problema proposto neste trabalho compreende quatro variações de cenário principais:

1. utilização de dados provenientes de três manobras distintas;
2. utilização de sinais com diferentes intensidades de contaminação por ruído;
3. variação no número de variáveis observadas;
4. variação no número de parâmetros estimados.

Nesse contexto, torna-se importante citar que todas as simulações realizadas utilizam dados provenientes do simulador de voo *Flightgear* (Perry, 2004) e são representadas por séries temporais de aproximadamente 1100 observações cada. Além disso, a aeronave selecionada consiste em um *Cessna 172P Skyhawk* (modelo 1981).

Quanto ao primeiro aspecto relacionado às variações de cenário abordadas, deve-se destacar que as manobras têm duração aproximada de 1 minuto e podem ser descritas por:

Manobra 1: asa nivelada e altura aproximadamente constante;

Manobra 2: asa nivelada e *doublet* de profundor;

Manobra 3: rolamento e arfagem negativos seguidos de rolamento e arfagem positivos.

As Figuras 3.1, 3.2 e 3.3 ilustram as manobras 1, 2 e 3, respectivamente. Em todos os casos, considera-se o ponto $[x_e \ y_e \ z_e]^T = \mathbf{0}$ como o ponto inicial da manobra⁵ e todos os gráficos têm medições de distância em metros. Além disso, os pontos traçados são obtidos a partir de dados de GPS, amostrados a cada 1s, após operações de transformação e rotação de coordenadas⁶.

⁵Esse ponto também é indicado com um triângulo, e o ponto final com um quadrado.

⁶Veja a rotina Modelos\Geral\pgps2ned.m na Seção 5.5.

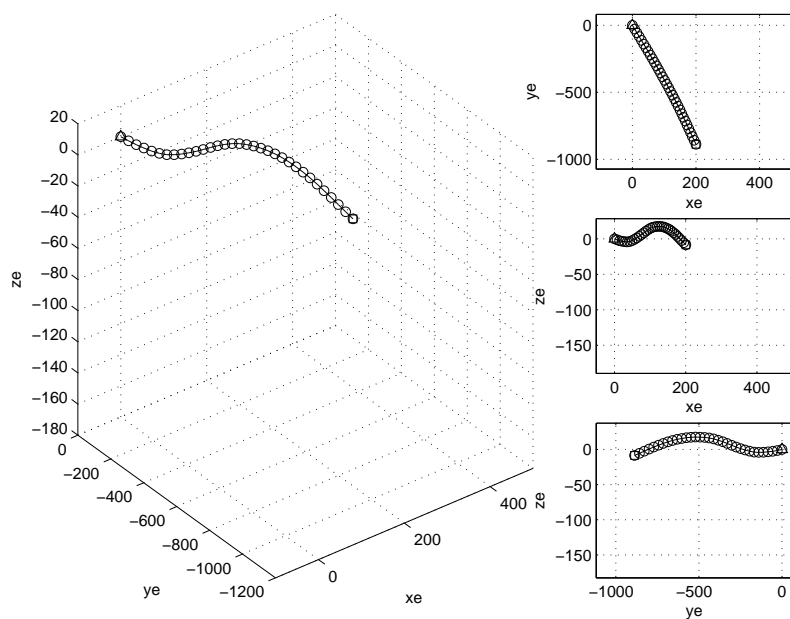


Figura 3.1: Manobra 1: Asa nivelada e altura aproximadamente constante.

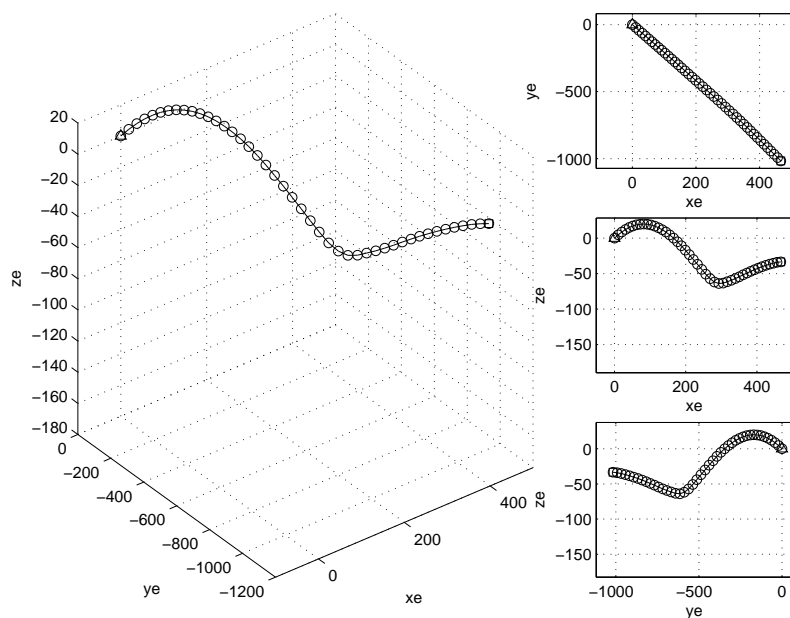


Figura 3.2: Manobra 2: Asa nivelada e *doublet* de profundor.

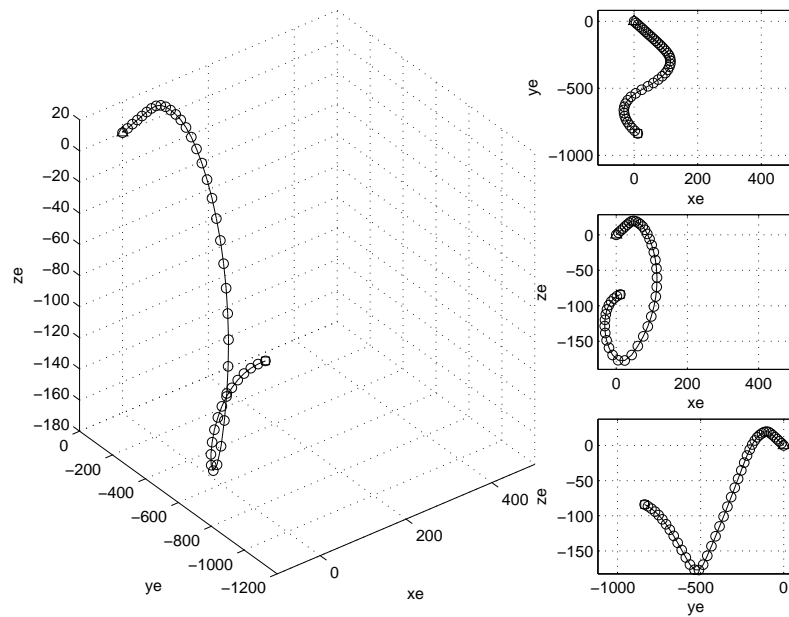


Figura 3.3: Manobra 3: Rolamento e arfagem negativos seguidos de rolamento e arfagem positivos.

A Figura 3.4 ilustra a manobra 3 em detalhe: diminuição simultânea dos valores de y_e e z_e seguida do aumento simultâneo dessas mesmas variáveis.

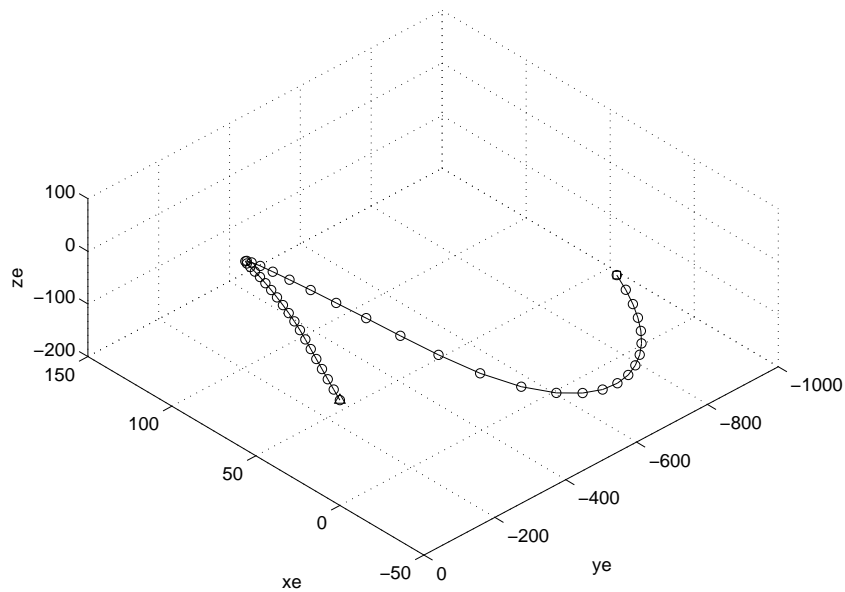


Figura 3.4: Manobra 3: Rolamento e arfagem negativos seguidos de rolamento e arfagem positivos em detalhe.

Finalmente, a Figura 3.5 ilustra uma comparação entre as três manobras, em escala.

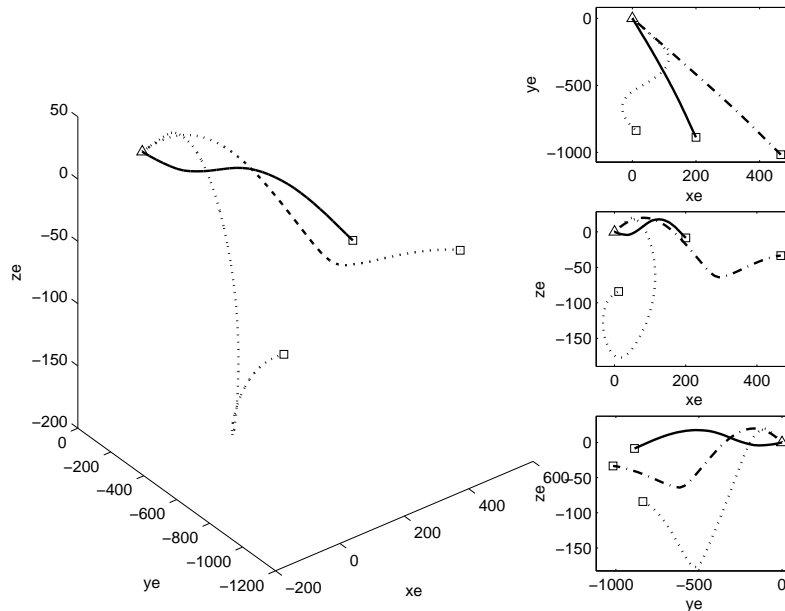


Figura 3.5: Comparação entre as manobras: manobra 1 (—); manobra 2 (— · —); manobra 3 (···).

Importante observar o motivo pelo qual utilizam-se dados provenientes de três manobras distintas. A qualidade da estimação dos parâmetros de um modelo depende fortemente das características dos seus sinais de entrada, segundo critérios de persistência de excitação (Aguirre, 2004). Devido a isso, busca-se destacar a identificabilidade dos parâmetros em cenários de excitação aproximadamente nula, excitação da dinâmica longitudinal e de excitação das dinâmicas longitudinal e látero-direcional, respectivamente.

Quanto ao segundo aspecto, utilizam-se quatro níveis de contaminação por ruído. Para isso, somam-se aos sinais de entrada, ruído branco com desvio padrão igual a 0, 1, 2 e 4 vezes o desvio padrão típico das respectivas medições provenientes de unidades de medição inercial. Para isso, consideram-se $\sigma_{acc} = 0,12 \text{ m/s}^2$ e $\sigma_{gir} = 0,07 \text{ rad/s}$ (Crista, 2004). O objetivo dessa abordagem consiste em testar a robustez dos algoritmos no que diz respeito à utilização de dados com diferentes níveis de contaminação por ruído.

Quanto ao terceiro aspecto, o número de variáveis observadas é aumentado com o intuito de inserir maior quantidade de informação no processo de estimação e assim de aumentar a eficiência dos algoritmos. Com isso, o vetor de saída

$$\mathbf{y}_1 = [x_e, y_e, z_e, v_T, \alpha, \beta]^T$$

é expandido de forma a se tornar

$$\mathbf{y}_2 = [x_e, y_e, z_e, v_T, \alpha, \beta, \phi, \theta, \psi,]^T,$$

que consiste em outro cenário típico de reconstrução de trajetória (Teixeira et al., 2005).

Finalmente, quanto ao quarto aspecto, duas configurações para o vetor de parâmetros estimados são utilizadas. Na primeira delas, apenas os valores de polarização dos girômetros e acelerômetros são estimados, de tal forma que

$$\boldsymbol{\theta}_1 = [\Delta p, \Delta q, \Delta r, \Delta a_x, \Delta a_y, \Delta a_z]. \quad (3.9)$$

Em seguida, o vetor de parâmetros é expandido conforme a equação (3.7), reescrita aqui por conveniência:

$$\boldsymbol{\theta}_2 = [k_p, \Delta p, k_q, \Delta q, k_r, \Delta r, k_{a_x}, \Delta a_x, k_{a_y}, \Delta a_y, k_{a_z}, \Delta a_z]^T.$$

Para todos os experimentos realizados, o problema de estimação de parâmetros é resolvido de forma a gerar uma massa de dados com propriedades estatísticas significativas no que diz respeito ao desempenho dos algoritmos. Isso implica dizer que os algoritmos não classificados como genéticos são executados diversas vezes (mais especificamente, 50 realizações).

Quanto aos algoritmos genéticos, a própria concepção do processo de otimização cria resultados com características estatísticas. Essa afirmativa se justifica pelo fato de esses algoritmos serem baseados na criação de conjuntos de pontos (ou populações) no espaço de objetivos e conduzir o processo de busca segundo transições e transformações de caráter probabilístico. Devido a isso, a execução de diversas realizações de um algoritmo genético não se justifica para a finalidade de interesse.

A repetição do processo de otimização é executada de tal forma que o ponto ótimo seja sempre caracterizado por valores de polarização nulos e

ganho unitário. Com isso, a única diferença existente entre as realizações consiste no ponto inicial da busca, o que confere o caráter estatístico ao experimento.

O conjunto de pontos iniciais para todas as realizações de um determinado algoritmo é criado de forma a possuir distribuição uniforme. Essa distribuição é definida no intervalo com centro no valor ideal ($\Delta = 0$ e $k = 1$) e de tamanho igual a duas vezes a variação máxima garantida pela instrumentação (Δk_{gir}^{max} , Δ_{gir}^{max} , Δk_{acc}^{max} e Δ_{acc}^{max}).

Isso implica dizer que, se o vetor de parâmetros iniciais é definido por

$$\boldsymbol{\theta}_0 = [k_{p0}, \Delta p_0, k_{q0}, \Delta q_0, k_{r0}, \Delta r_0, k_{a_x0}, \Delta a_{x0}, k_{a_y0}, \Delta a_{y0}, k_{a_z0}, \Delta a_{z0}]^T,$$

então, k_{p0} , Δp_0 , k_{a_x0} e Δa_{x0} , por exemplo, serão elementos de conjuntos com distribuição uniforme definidos em intervalos tais que

$$1 - \Delta k_{gir}^{max} < k_{p0} < 1 + \Delta k_{gir}^{max}, \quad -\Delta_{gir}^{max} < \Delta p_0 < \Delta_{gir}^{max}, \quad (3.10)$$

$$1 - \Delta k_{acc}^{max} < k_{a_x0} < 1 + \Delta k_{acc}^{max}, \quad -\Delta_{acc}^{max} < \Delta a_{x0} < \Delta_{acc}^{max}, \quad (3.11)$$

em que (Crista, 2004):

$$\Delta k_{gir}^{max} = 0,01 \text{ rad/s}, \quad \Delta_{gir}^{max} = 0,6 \text{ rad/s}, \quad (3.12)$$

$$\Delta k_{acc}^{max} = 0,01 \text{ m/s}^2, \quad \Delta_{acc}^{max} = 0,02 \text{ m/s}^2. \quad (3.13)$$

Para a avaliação de desempenho dos algoritmos, o seguinte procedimento é adotado. Primeiramente, criam-se dois vetores responsáveis pela representação do valor mais freqüentemente estimado de tal forma que:

I_1 = valor mais freqüentemente estimado (moda): valor esperado como resultado do estimador no espaço de parâmetros;

$I_{1,\%}$ = valor mais freqüentemente estimado (moda): valor esperado como resultado do estimador em um espaço de soluções normalizado (0 a 100%).

Além dos vetores I_1 e $I_{1,\%}$, utilizam-se 4 índices definidos por:

I_2 = primeira fronteira norma-infinito ocupada no espaço de soluções;

I_3 = número médio de execuções do modelo: medida do tempo de processamento médio do algoritmo;

I_4 = distância Euclideana entre o vetor de parâmetros mais frequentemente estimado pelo algoritmo (segundo I_1) e o vetor de parâmetros real no espaço de parâmetros;

$I_{4,\%}$ = distância Euclideana entre o vetor de parâmetros mais frequentemente estimado pelo algoritmo (segundo $I_{1,\%}$) e o vetor de parâmetros real no espaço de soluções normalizado (0 a 100%).

Maiores detalhades a respeito das descrições dos vetores I_1 e $I_{1,\%}$ e dos índices I_2 , I_3 , I_4 e $I_{4,\%}$ são apresentados na Seção 3.3. A Tabela 3.2 apresenta um resumo de informações a respeito dos testes realizados.

Tabela 3.1: Resumo de informações a respeito dos testes realizados.

Algoritmos	Quantidade total: 13;
	Quantidade determinísticos: 7;
	Quantidade estocásticos: 6 (5 genéticos);
	Realizações não-genéticos: 50;
	Realizações genéticos: 1.
Manobras	Quantidade: 3;
	Pontos por série temporal: 1100;
	Simulação: <i>Flightgear</i> , <i>Cessna 172P Skyhawk</i> (modelo 1981).
Ruído	$\sigma_{acc}^r = [0, 1, 2, 4]^T \sigma_{acc}$;
	$\sigma_{gir}^r = [0, 1, 2, 4]^T \sigma_{gir}$.
Var. Observadas	$\mathbf{y}_1 = [x_e, y_e, z_e, v_T, \alpha, \beta]^T$;
	$\mathbf{y}_2 = [x_e, y_e, z_e, v_T, \alpha, \beta, \phi, \theta, \psi]^T$.
Par. Estimados	$\boldsymbol{\theta}_1 = [\Delta p, \Delta q, \Delta r, \Delta a_x, \Delta a_y, \Delta a_z]^T$;
	$\boldsymbol{\theta}_2 = [k_p, \Delta p, k_q, \Delta q, k_r, \Delta r, k_{a_x}, \Delta a_x, k_{a_y}, \Delta a_y, k_{a_z}, \Delta a_z]^T$;
	Ponto ótimo: $k = 1$ e $\Delta = 0$;
	Pontos iniciais: Distribuição uniforme.
Índices	Quantidade: 4.

Finalmente, o ambiente de estimação/reconstrução pode ser representado segundo a Figura 3.6.

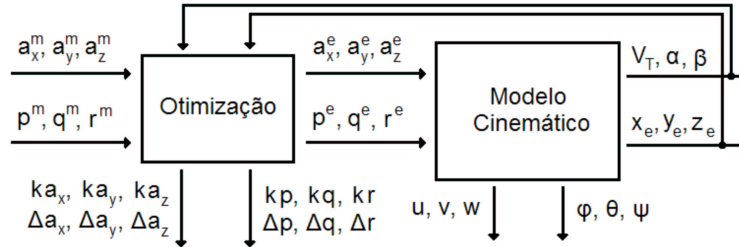


Figura 3.6: Esquema de reconstrução de trajetória via estimação de parâmetros.

Com isso, pretende-se dizer que a estratégia consiste em executar o seguinte procedimento:

1. medir os sinais da manobra executada: dos girômetros e acelerômetros (distorcidos e contaminados por ruído) e das variáveis de estado selecionadas como saídas do modelo cinético;

e, de forma recursiva,

2. reconstruir os sinais de entrada com a utilização do vetor de parâmetros estimado;
3. utilizar os sinais reconstruídos para reconstruir os estados do modelo cinético;
4. utilizar os sinais de saída do modelo cinético e suas respectivas medições para calcular novo valor do vetor de parâmetros.

Com o fim do processo, obtém-se não somente a reconstrução dos estados do modelo cinético como também os valores dos parâmetros de ganho e polarização responsáveis pela distorção dos sinais provenientes dos girômetros e acelerômetros.

3.3 Sobre os testes realizados

Um cenário de estimação é representado pela utilização de (Tabela 3.2):

- um algoritmo;
- uma manobra (conjunto de dados medidos);
- uma condição de contaminação por ruído;
- um vetor de variáveis observadas;
- um vetor de parâmetros estimados.

Este trabalho contempla a utilização de 13 algoritmos, 3 manobras, 4 condições de contaminação por ruído, 2 vetores de variáveis observadas e 2 vetores de parâmetros estimados. Essa formulação dá origem, conforme conduzida neste trabalho, a 260 diferentes cenários de reconstrução/estimação. Esses cenários são agrupados em 5 conjuntos de testes nos quais todos os algoritmos são utilizados⁷ para a resolução do problema de reconstrução de trajetória de aeronaves com os 4 níveis de contaminação por ruído especificados.

Cada conjunto de testes contempla, portanto, 52 cenários de estimação. Esses conjuntos são diferenciados segundo as características descritas na Tabela 3.3.

Tabela 3.2: Descrição dos conjuntos de testes

Conjunto	Asa Nivelada	Altitude	Vetor Estimado	Vetor Observado
1	Sim	Constante	θ_1	y_1
2	Sim	→ Variável	θ_1	y_1
3	Sim	Variável	θ_1	→ y_2
4	→ Não	Variável	θ_1	y_2
5	Não	Variável	→ θ_2	y_2

Para cada cenário de reconstrução/estimação, quatro índices são avaliados. De forma mais detalhada, os procedimentos adotados para cálculo desses índices podem ser descritos por:

⁷No caso dos algoritmos não genéticos, executam-se 50 realizações do mesmo algoritmo com o objetivo de gerar o conjunto de dados para análise estatística do seu desempenho.

I_1 : Divide-se o espaço de soluções válidas (dentro das faixas dos instrumentos) em 20^n subespaços, em que n é o número de elementos do vetor de parâmetros. Com isso, consegue-se uma resolução de 5% em cada eixo do espaço de soluções. O valor de I_1 consiste então na média dos vetores pertencentes ao subespaço com maior quantidade de soluções. A Figura 3.7 ilustra a utilização desse índice em um espaço de soluções de duas dimensões. Nesse caso, $I_1 = [0,13; 0,0783]^T$ (subespaço cinza) pois $\hat{\theta}_1 = [0,14; 0,07]^T$, $\hat{\theta}_2 = [0,13; 0,09]^T$, $\hat{\theta}_3 = [0,12; 0,075]^T$ e considera-se que as faixas para θ_1 e θ_2 são, ambas, iguais a $[0; 1]$;

$I_{1,\%}$: Representação de I_1 em um espaço normalizado (0 a 100%). Procede-se de tal forma que:

$$\begin{aligned} I_{1,\%}(i) = 0\% &\Rightarrow \hat{\theta}_i = \hat{\theta}_i^{\text{ideal}}; \\ I_{1,\%}(i) = 100\% &\Rightarrow \hat{\theta}_i = \hat{\theta}_i^{\text{ideal}} \pm \hat{\theta}_i^{\text{max}}, \end{aligned} \quad (3.14)$$

em que $\hat{\theta}_i^{\text{ideal}}$ é o valor ideal para o parâmetro estimado ($\Delta = 0$ e $k = 1$) e $\hat{\theta}_i^{\text{max}}$ é a variação máxima garantida pela instrumentação relacionada ao parâmetro em questão⁸. O objetivo dessa utilização é enfatizar a distância do vetor mais freqüentemente estimado ao vetor real. Essa distância é calculada parâmetro por parâmetro tal que, no caso da Figura 3.7 por exemplo, $I_{1,\%} = [13; 7,83]^T$.

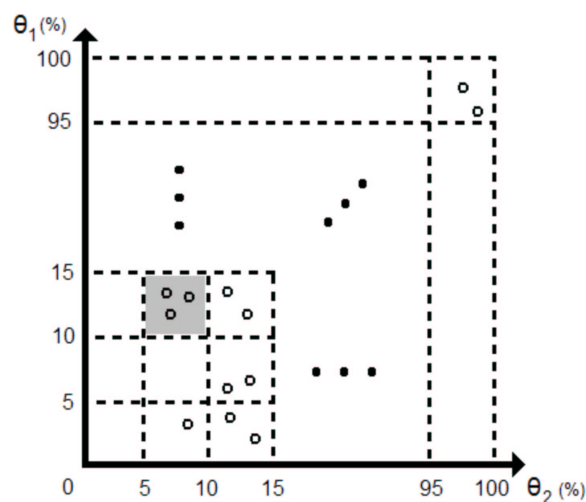


Figura 3.7: Vetores I_1 e $I_{1,\%}$: Espaço de soluções com 2 dimensões.

⁸Seção 3.2: k_{gir}^{max} , Δ_{gir}^{max} , k_{acc}^{max} e Δ_{acc}^{max} .

I_2 : Primeiramente, calcula-se a norma-infinito da diferença do vetor real com cada um dos vetores de parâmetros estimados ($\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_i\|_\infty$). Em seguida, o vetor estimado que apresentar a menor distância ao valor real (segundo a norma-infinito) é considerado como melhor estimacão segundo esse índice. Finalmente, utiliza-se $I_2 = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_i\|_\infty$. Na Figura 3.8, por exemplo, $I_2 = 10\%$ devido ao vetor estimado indicado pela seta. A importância da utilização desse índice está na necessidade de definição de um critério de quantificação da distância existente entre o vetor de parâmetros real e o estimado. Nesse caso, o uso da norma-infinito (em detrimento à norma Euclideana, $\|\cdot\|_2$) se justifica pela necessidade de inclusão apenas dos vetores estimados para os quais todos os elementos estão contidos dentro das faixas dos respectivos instrumentos. Com isso, a utilização de fronteiras definidas por hipercubos ($\|\cdot\|_\infty$) são mais convenientes que aquelas definidas por hiperesferas ($\|\cdot\|_2$);

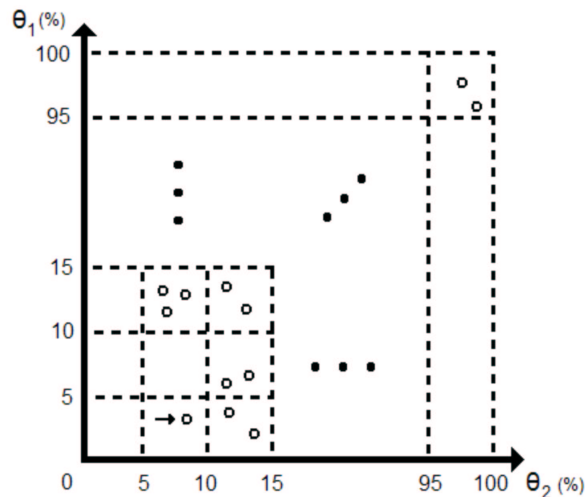


Figura 3.8: Índice de desempenho I_2 : Espaço de soluções com 2 dimensões.

I_3 : Calcula-se o número total de execuções do modelo e divide-se o resultado pelo número de realizações do algoritmo. No caso de algoritmos genéticos, o resultado é dividido pelo número de indivíduos da população. Esse índice oferece uma indicação, de forma padronizada, do tempo de execução do algoritmo uma vez que a simulação do modelo consiste na tarefa com maior custo associado no que diz respeito ao tempo de execução;

I_4 : Calcula-se a distância Euclideana entre o vetor de parâmetros mais freqüentemente estimado pelo algoritmo (segundo I_1) e o vetor de parâmetros real no espaço de soluções (Figura 3.9);

$I_{4,\%}$: Calcula-se a distância Euclideana entre o vetor de parâmetros mais freqüentemente estimado pelo algoritmo (segundo $I_{1,\%}$) e o vetor de parâmetros real no espaço de soluções normalizado (0 a 100%), segundo a Figura 3.9.

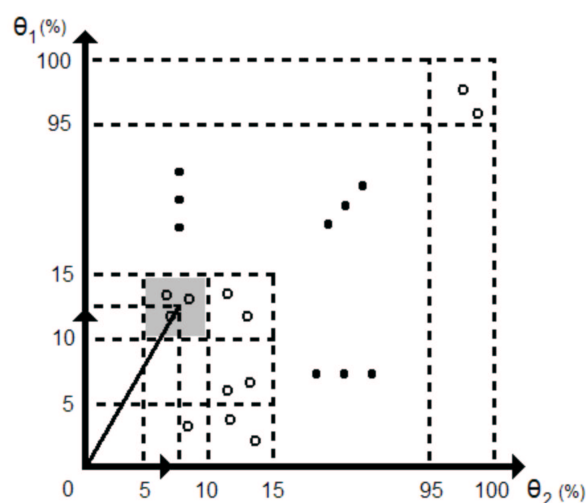


Figura 3.9: Índices de desempenho I_4 e $I_{4,\%}$: Espaço de soluções com 2 dimensões.

Os resultados obtidos são apresentados de duas maneiras distintas: sob a forma de tabelas e de gráficos. No primeiro caso, os resultados são apresentados divididos em conjuntos de testes. Apresentam-se, portanto, cinco tabelas para cada algoritmo⁹ com valores para I_1 , $I_{1,\%}$, I_2 e I_3 para os quatro cenários de contaminação por ruído selecionados.

No segundo caso, os gráficos têm por objetivo facilitar a análise comparativa de desempenho do algoritmo estudado nos diferentes cenários de estimação/reconstrução. Para isso, criam-se quatro gráficos tais que:

- gráfico a : representa o índice I_4 ;

⁹A exceção consiste no algoritmo de Gauss-Newton, cujos resultados são apresentados em apenas quatro tabelas.

- gráfico *b*: representa o índice $I_{4,\%}$;
- gráfico *c*: representa o índice I_2 ;
- gráfico *d*: representa o índice I_3 .

Os índices de desempenho criados neste trabalho têm por objetivo orientar a análise estatística e comparativa dos algoritmos na resolução do problema de reconstrução de trajetória de aeronaves. No caso dos vetores I_1 e $I_{1,\%}$, a análise é direta e seus valores representam o valor mais freqüente (ou esperado) para o estimador utilizado. Contudo, a representação de vetores em espaços de dimensão maior que três é difícil, por questões geométricas. Essa é uma das motivações para a criação dos índices I_2 , I_3 , I_4 e $I_{4,\%}$.

O índice I_2 localiza a solução mais próxima da ideal em termos da maior distância em cada um dos eixos do espaço de soluções. Para isso, calcula-se a norma-infinito de todas as soluções e aquela com menor valor associado é considerada a mais próxima da solução ideal. Apesar de não se tratar da mais freqüente, essa solução quantifica o quão próximo o estimador é capaz de chegar do resultado ideal.

O índice I_3 avalia os algoritmos quanto ao tempo de execução. Essa avaliação é feita de maneira indireta e padronizada, via contabilização do número de execuções do modelo cinemático. Esse procedimento é adotado devido à constatação de que a execução do modelo consiste na tarefa com maior custo associado no que diz respeito ao tempo de execução. Além disso, busca-se desvincular o tempo de execução com características do computador utilizado.

Além dos índices I_2 e I_3 , criam-se dois novos índices para representações em gráficos, diretamente relacionados a I_1 e $I_{1,\%}$. Enquanto o primeiro índice consiste na distância Euclideana entre o vetor real e o vetor mais freqüentemente estimado no espaço de soluções, o segundo consiste nessa mesma grandeza representada em um espaço normalizado (0 a 100%). De forma resumida tratam-se de I_4 e $I_{4,\%}$, respectivamente.

A criação desses dois novos índices (I_4 e $I_{4,\%}$) ganha importância ao se constatar uma maior facilidade na análise dos resultados em gráficos quando comparada à análise em tabelas no contexto deste trabalho. Como o desempenho geral dos algoritmos pode ser descrito por I_1 , $I_{1,\%}$ e os gráficos consistem nas melhores ferramentas de análise, boa parte da investigação a respeito do desempenho dos algoritmos nos cenários selecionados é feita com

a utilização dos índices I_4 e $I_{4,\%}$. Mais especificamente, dá-se preferência ao segundo índice ($I_{4,\%}$) com o intuito de desconsiderar diferenças de escala entre os parâmetros estimados.

Com a definição dos índices, torna-se bastante intuitivo que o bom desempenho dos algoritmos seja representado por pequenos valores de I_2 , I_3 , I_4 e $I_{4,\%}$ e, conseqüentemente, de valores de I_1 o mais próximos possível do valor ideal¹⁰.

Pelos motivos citados, o índice $I_{4,\%}$ consiste em um dos mais importantes a serem analisados. Devido a isso, inicia-se a análise de desempenho dos algoritmos com a investigação dos valores desse índice. Por uma questão de organização, inicia-se a análise pelos algoritmos determinísticos e finaliza-se com os algoritmos genéticos.

¹⁰No caso dos dados simulados, $\Delta = 0$ e $k = 1$.

MÉTODOS DE OTIMIZAÇÃO

4.1 Introdução

Um problema de otimização é caracterizado pela necessidade de determinar soluções *dominantes* (*ótimas*) dentre todas as soluções *factíveis*. Esses dois conjuntos de soluções são constituídos por pontos $\boldsymbol{\theta} \in \mathbb{R}^n$ mapeados por um vetor de funções objetivo¹ $\mathbf{c}(\boldsymbol{\theta}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

De forma geral, um problema de otimização mono-objetivo² pode ser representado por:

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \mathbb{R}^n} / \max_{\boldsymbol{\theta} \in \mathbb{R}^n} \quad & c(\boldsymbol{\theta}) & (4.1) \\ \text{sujeito a} \quad & r_i(\boldsymbol{\theta}) = 0 & i = 1, \dots, l \\ & r_i(\boldsymbol{\theta}) > 0 & i = l + 1, \dots, p \\ & \theta_k^{(L)} \leq \theta_k \leq \theta_k^{(U)} & k = 1, \dots, n \end{aligned}$$

em que a função objetivo $c(\boldsymbol{\theta})$, as l restrições de igualdade e as $(p - l)$ restrições de desigualdade são funções escalares. A existência dessas restrições constitui um problema de otimização *restrito*. Caso não existissem, tratar-se-ia de um problema de otimização *irrestrito*. Além disso, o vetor de parâmetros $\boldsymbol{\theta}$ é limitado inferiormente por $\boldsymbol{\theta}^{(L)} = [\theta_1^{(L)}, \theta_2^{(L)}, \dots, \theta_n^{(L)}]^T$ e superiormente por $\boldsymbol{\theta}^{(U)} = [\theta_1^{(U)}, \theta_2^{(U)}, \dots, \theta_n^{(U)}]^T$.

¹Neste trabalho, o problema de otimização consiste basicamente na minimização de uma função custo. Esse é o motivo pelo qual escolheu-se representar a função objetivo por $c(\boldsymbol{\theta})$ e não por $f(\boldsymbol{\theta})$, como de costume.

²Aspectos relacionados à resolução de problemas de otimização multiobjetivo são abordados na seção 4.2.

O conjunto dos pontos factíveis é formado por todos os pontos $\theta \in \mathbb{R}^n$ que satisfazem as restrições. Por sua vez, um ponto dominante θ^* consiste em um ponto factível tal que $c(\theta^*)$ é uma solução ótima (ainda que localmente) da função objetivo. Considerando-se uma região factível $\Omega(\theta^*, \delta)$ na vizinhança de um ponto ótimo, tem-se que:

$$c(\theta^*) < c(\theta) \Rightarrow \theta^* \triangleleft \theta, \quad \forall \theta \in \Omega(\theta^*, \delta), \quad \theta \neq \theta^* \quad (4.2)$$

em que $\theta^* \triangleleft \theta$ denota θ^* “domina” θ segundo a formulação do problema (neste caso, de minimização).

Grande parte dos problemas reais apresenta soluções que, embora obedeam a definição (4.2), são tipicamente indesejadas. Soluções desse tipo são denominadas ótimos locais espúrios e são bastante comuns em problemas de estimação de parâmetros. A Figura 4.1 ilustra um ponto de mínimo local em contraste com um ponto de mínimo global.

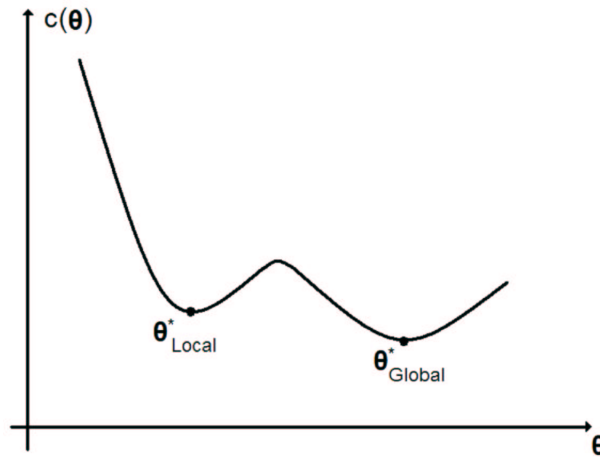


Figura 4.1: Comparação entre ótimo local e ótimo global (minimização).

De maneira geral, os métodos de otimização podem ser classificados segundo particularidades de implementação dos seus algoritmos. Essa classificação cria grandes grupos de métodos semelhantes e auxilia na compreensão do seu comportamento. Embora boa parte dos métodos seja fundamentada, de forma ponderada, em características de mais de um desses grupos, a associação se torna importante no sentido de permitir análises críticas baseadas em aspectos gerais.

Os métodos de otimização utilizados neste trabalho podem ser agrupados segundo a Tabela 1.3, reescrita aqui por conveniência:

Tabela 4.1: Ilustração e classificação dos algoritmos desenvolvidos.

Determinísticos	Busca Direta	Nelder-Mead
	Região de Confiança	DogLeg (Powell)
	Direção de Busca	Gradiente
		Gradiente Conjugado
		Gradiente Conjugado Pré-Condicionado
	Métodos de Newton	Gauss-Newton
Mínimos Quadrados	Levenberg-Marquardt	
Estocásticos	Simulated Annealing	
	Genéticos	Vector Evaluated Genetic Algorithm
		Weight-Based Genetic Algorithm
		Multiple Objective Genetic Algorithm
		Non-dominated Sorting Genetic Algorithm II
		Strength Pareto Evolutionary Algorithm

A primeira classificação consiste na distinção entre algoritmos determinísticos e estocásticos. Essa distinção se deve basicamente à maneira com que os algoritmos desses grupos conduzem o processo de otimização. Enquanto algoritmos determinísticos utilizam procedimentos cujos resultados dependem apenas das características do problema no instante da sua utilização, os estocásticos vinculam seus resultados a eventos aleatórios ou de caráter probabilístico.

Isso implica dizer, por exemplo, que várias execuções de um determinado algoritmo determinístico a partir de um mesmo ponto inicial em uma mesma massa de dados dão origem sempre o mesmo resultado. Contudo, a própria concepção de um algoritmo estocástico faz com que cada execução de um algoritmo pertencente a essa classe dê origem a um resultado distinto, ainda que se trate de um cenário como o citado.

As demais classes apresentadas na Tabela 4.1 são descritas em conjunto com os respectivos algoritmos. Nas Seções 4.3 a 4.8, apresentam-se as classes *busca direta*, *direção de busca*, *mínimos quadrados*, *região de confiança*, *métodos de Newton* e *algoritmos genéticos* respectivamente. Em conjunto com essa apresentação, procede-se com a descrição dos algoritmos em si e com a apresentação do seu desempenho na resolução do problema de reconstrução de trajetória de aeronaves segundo os critérios descritos no Capítulo 3. Na Seção 4.9, apresenta-se o único algoritmo estocástico implementado não clas-

sificado como genético: o algoritmo de *Simulated Annealing*. Finalmente, a Seção 4.2 ilustra conceitos relativos à otimização multiobjetivo e preceitos envolvidos na implementação de algoritmos com essa abordagem.

4.2 Otimização multiobjetivo

Otimização refere-se à busca ou determinação de uma ou diversas soluções factíveis correspondentes a valores extremos (mínimo/máximo) de um ou mais objetivos. Devido às propriedades extremas de soluções ótimas, a utilização de métodos de otimização possui grande importância em situações práticas como desenvolvimento em engenharia, experimentação científica e tomada de decisão empresarial, por exemplo.

Além disso, grande parte dos problemas reais envolve a determinação de soluções que satisfaçam a otimização de diversos objetivos simultaneamente. Esse cenário implica, de forma quase preponderante, a resolução de problemas conflitantes e a necessidade de geração de um conjunto de soluções igualmente eficientes e não-dominadas sob o ponto de vista dos objetivos a serem otimizados.

Esse conflito é representado na Figura 4.2, em que se ilustra um problema de otimização (minimização) multiobjetivo com duas funções $c_1(\theta), c_2(\theta) : \mathbb{R} \rightarrow \mathbb{R}$ com mínimos em θ_1^* e θ_2^* , respectivamente. Uma primeira análise do cenário indica que o ponto mínimo do problema multiobjetivo deve estar contido na região delimitada pelos ótimos individuais, $[\theta_1^*, \theta_2^*]$. Nessa região, um aumento no valor do parâmetro θ provoca uma diminuição no valor de $c_2(\theta)$ mas implica um aumento no valor de $c_1(\theta)$. O inverso ocorre quando o valor de θ é diminuído nessa região.

Os métodos desenvolvidos para a otimização de diversos objetivos simultaneamente são denominados *métodos de otimização multiobjetivo*. Devido à falta de algoritmos dedicados à solução de problemas com essa característica no passado, constituía-se prática comum transformarem-se problemas multiobjetivo em mono-objetivo com a definição de um novo objetivo modificado a partir dos objetivos originais. Contudo, essa metodologia retira a generalidade do procedimento com a transformação de um conjunto de soluções ótimas em uma única solução modificada.

Ao conjunto de soluções igualmente eficientes de um determinado pro-

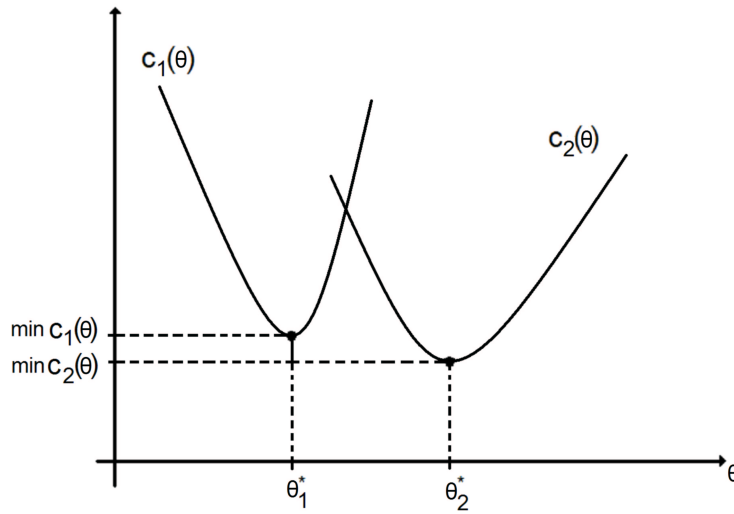


Figura 4.2: Resolução de problemas conflitantes em otimização multiobjetivo.

blema, dá-se o nome *fronteira*. À fronteira cujas soluções sejam tais que não haja pontos mais eficientes na região factível, dá-se o nome *Pareto-ótimo* (Takahashi, 2004a). Esse conjunto de soluções é tal que as seguintes condições sejam satisfeitas:

1. quaisquer duas soluções devem ser não-dominadas quando comparadas entre si;
2. qualquer solução que não pertença ao conjunto *Pareto-ótimo* deve ser dominada por ao menos uma solução pertencente a esse conjunto.

A Figura 4.3 ilustra as definições citadas para um problema de otimização com duas funções objetivo: c_1 e c_2 . Em todos os gráficos (a, b, c, d), a fronteira Pareto-ótimo é representada pelo(s) limite(s) em linha contínua da região factível. Também em todos os gráficos, as soluções 1, 2, 3 e 4 formam uma fronteira não-dominada P_1 e as soluções 5, 6 e 7 constituem uma fronteira dominada P_2 . Ilustram-se os problemas de *mínimo-mínimo* (a), *máximo-máximo* (b), *mínimo-máximo* (c) e *máximo-Mínimo* (d).

A estruturação geral de um problema de otimização multiobjetivo pode

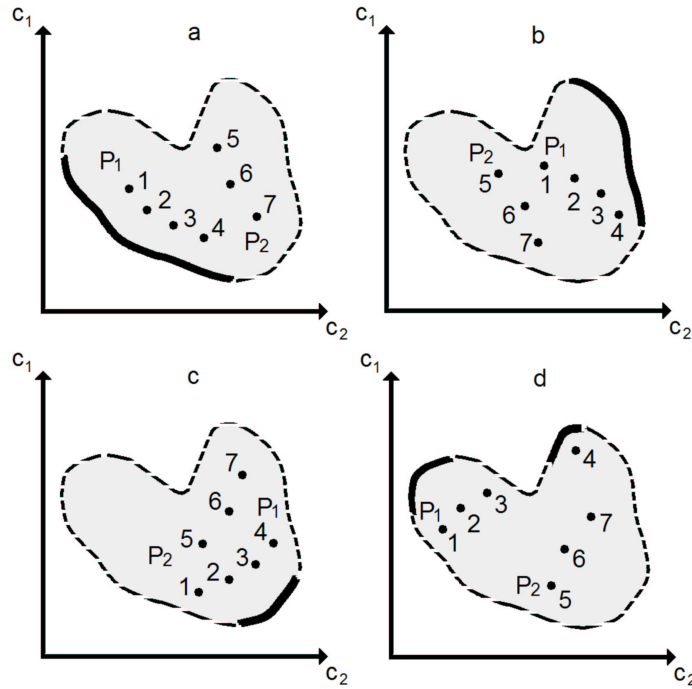


Figura 4.3: Definições e tipos de problemas de otimização multiobjetivo: a) *mínimo-mínimo*; b) *máximo-máximo*; c) *mínimo-máximo*; d) *máximo-Mínimo*. A fronteira Pareto-ótimo é representada pela linha contínua e a fronteira (não-dominada) P_1 domina a fronteira P_2 .

ser descrita por:

$$\begin{aligned}
 & \min_{\theta \in \mathbb{R}^n} / \max_{\theta \in \mathbb{R}^n} && c_j(\theta) && j = 1, \dots, m && (4.3) \\
 & \text{sujeito a} && r_i(\theta) = 0 && i = 1, \dots, l \\
 & && r_{ii}(\theta) > 0 && ii = l + 1, \dots, p \\
 & && \theta_k^{(L)} \leq \theta_k \leq \theta_k^{(U)} && k = 1, \dots, n
 \end{aligned}$$

em que $c_j(\theta)$ consiste na j -ésima função objetivo a ser minimizada ou maximizada, $\mathbf{r}(\theta) = [r_1(\theta), \dots, r_p(\theta)]^T$ representa o vetor de restrições de desigualdade e igualdade e θ consiste no vetor de parâmetros.

Além disso, qualquer algoritmo de otimização multiobjetivo deve ser implementado com o intuito de alcançar as seguintes metas:

1. obter um conjunto de soluções o mais próximo possível do conjunto *Pareto-ótimo*;

2. obter um conjunto de soluções o mais esparsas possível, de forma a garantir a diversidade das soluções.

Os algoritmos de otimização multiobjetivo podem ser classificados quanto à metodologia utilizada para obtenção do conjunto *Pareto-ótimo* em duas categorias: Algoritmos Clássicos e Algoritmos Genéticos. Os algoritmos clássicos baseiam-se em procedimentos determinísticos para a aproximação das soluções ótimas (Takahashi, 2004b). Tipicamente, esses algoritmos têm como primeiro passo a geração de uma solução aleatória inicial e, baseados em uma regra de transição pré-especificada e em informações locais, determinam uma direção de busca. Com isso, uma busca unidirecional é executada ao longo da direção definida. A melhor solução é considerada como novo ponto de partida e o procedimento se repete por um determinado número de vezes.

Os algoritmos genéticos utilizam princípios de genética e seleção naturais. Alguns desses princípios são implementados artificialmente de maneira a construir algoritmos robustos, de aplicabilidade geral e de fácil implementação. Essa categoria de algoritmos apresenta operadores representativos da genética natural como reprodução, seleção e mutação.

Neste trabalho, dentre todos os algoritmos implementados, apenas os genéticos são genuinamente algoritmos de otimização multiobjetivo. Com isso, apenas para esses algoritmos procede-se com a minimização do erro quadrático médio individualmente para cada uma das curvas a serem adequadas.

Para os algoritmos não genéticos, cria-se uma função objetivo artificial única. Para isso, procede-se com a concatenação das séries temporais de interesse³ no sentido de se obter uma única série. Em seguida todos os procedimentos e cálculos são aplicados aos valores desse novo vetor de valores.

4.3 Busca Direta

Algoritmos de *Busca Direta* são assim classificados por utilizarem apenas valores da função a ser otimizada e algum critério de avaliação para o seqüenciamento das iterações (Lewis et al., 2000). Mais especificamente, utiliza-se a comparação do valor atual da função com o “melhor” até o instante, em conjunto com alguma regra de direcionamento para a determinação do próximo

³Séries temporais dos acelerômetros e girômetros da IMU.

ponto a ser avaliado. Os algoritmos de *Busca Direta* são comumente denominados algoritmos de *ordem zero*. Essa classificação se baseia no número de termos utilizados da expansão em séries de Taylor da função objetivo⁴.

Inicialmente proposta por (Hooke and Jeeves, 1961), essa classe de algoritmos se destaca pela não-necessidade de utilização de informações da derivada da função objetivo. Apesar da disponibilidade de técnicas de cálculo numérico avançadas tenderem a motivar a utilização de métodos baseados em conceitos matemáticos relativamente mais complexos, a utilização dos algoritmos de *Busca Direta* permanece bastante recorrente. Os principais motivos para isso encontram-se em sua facilidade de implementação e abrangência no que diz respeito à resolução de problemas não-lineares (Lewis et al., 2000).

4.3.1 Método de Nelder-Mead

O algoritmo de Nelder-Mead (Nelder and Mead, 1965) consiste em um dos métodos de otimização mais utilizados atualmente. Sua similaridade com o método de programação linear *Simplex* (Splendey et al., 1962) e conseqüente simplicidade de implementação são os principais motivos para sua grande utilização em problemas de otimização em diversas áreas do conhecimento, ainda que sua convergência não seja comprovada para boa parte dos casos (Lagarias et al., 1998). De fato, esse método se trata de uma extensão mais abrangente e robusta do *simplex*⁵.

O método de Nelder-Mead baseia seu processo de busca pelo ponto ótimo de uma função⁶ real $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, apenas na avaliação dessa função. Isso implica que não se utiliza qualquer tipo de informação relacionada à derivada de $f(\mathbf{x})$ e, devido a isso, pertence ao grupo dos métodos de *busca direta*. Assim como vários outros algoritmos de busca direta, o algoritmo de Nelder-Mead utiliza uma matriz simplex a cada iteração. Essa particularidade dá origem a uma figura geométrica de n dimensões ($n + 1$ vértices) responsável pela busca do ponto ótimo na hipersuperfície da função a ser otimizada. Esse é o

⁴Similarmente, o método do *Gradiente* é classificado como um algoritmo de primeira ordem e o algoritmo de *Newton*, que prevê o cálculo da primeira e segunda derivadas da função objetivo, é classificado como de segunda ordem.

⁵Prova-se essa afirmação observando-se que todas as regras para convergência do *simplex* são também aplicadas no algoritmo de Nelder-Mead. O que os diferencia é a utilização de algumas regras adicionais pelo último método.

⁶“Uma função” é colocado de forma explícita para indicar o caráter mono-objetivo do método.

motivo pelo qual esse método também é conhecido como o método da *ameba*.

A implementação do algoritmo prevê a definição de quatro coeficientes responsáveis pela movimentação da figura geométrica no espaço de busca. Tratam-se dos coeficientes de *reflexão* (ρ), *expansão* (χ), *contração* (γ) e *encolhimento* (σ). Os valores desses parâmetros devem satisfazer as seguintes restrições (Nelder and Mead, 1965):

$$\rho > 0, \chi > 1, \chi > \rho, 0 < \gamma < 1, 0 < \sigma < 1. \quad (4.4)$$

Os valores tipicamente utilizados são:

$$\rho = 1, \chi = 2, \gamma = \sigma = 1/2 \quad (4.5)$$

Uma vez definidos os coeficientes, parte-se para as iterações do algoritmo. Ao início de cada iteração k , utiliza-se uma matriz simplex $\bar{\Delta}_k$ ($(n+1) \times n$) em que cada linha corresponde a um ponto da figura geométrica de n dimensões⁷. Uma dada iteração k é composta pelos seguintes passos⁸:

1. **Ordenação/Classificação:** Ordenam-se os $n+1$ vértices representados pelas linhas de $\bar{\Delta}$ de modo que $f(\mathbf{x}_1) \triangleleft f(\mathbf{x}_2) \triangleleft \dots \triangleleft f(\mathbf{x}_{n+1})$. Desse modo, a primeira linha da matriz simplex representa o ponto com melhor avaliação da função objetivo e assim sucessivamente. A Figura 4.4 ilustra a configuração inicial da figura geométrica.

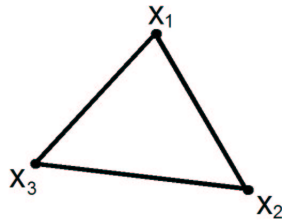


Figura 4.4: Configuração inicial da figura geométrica do algoritmo de Nelder-Mead.

2. **Reflexão:** Calcula-se o ponto de reflexão \mathbf{x}_r :

$$\mathbf{x}_r = \bar{\mathbf{x}} + \rho(\bar{\mathbf{x}} - \mathbf{x}_{n+1}), \quad (4.6)$$

⁷A definição de $\bar{\Delta}_1$, na primeira iteração, pode ser influenciada por algum conhecimento prévio da função objetivo, mas tipicamente é arbitrária.

⁸Para a exemplificação do método, será considerado um espaço de busca de duas dimensões e os sub-índices k são suprimidos por simplicidade.

em que $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ é o centróide dos n melhores pontos⁹. Caso a avaliação de $f_r = f(\mathbf{x}_r)$ implique $f_1 \triangleleft f_r \triangleleft f_n$, aceita-se o ponto \mathbf{x}_r e termina-se a iteração (Figura 4.5).

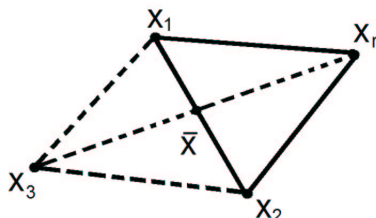


Figura 4.5: Configuração da figura geométrica após a etapa de *reflexão*.

3. **Expansão:** Se $f_r \triangleleft f_1$, então calcula-se o ponto de expansão \mathbf{x}_e :

$$\mathbf{x}_e = \bar{\mathbf{x}} + \chi(\mathbf{x}_r - \bar{\mathbf{x}}). \quad (4.7)$$

Caso a avaliação de $f_e = f(\mathbf{x}_e)$ implique $f_e \triangleleft f_r$, aceita-se o ponto \mathbf{x}_e e termina-se a iteração. Caso contrário ($f_r \triangleleft f_e$), aceita-se o ponto \mathbf{x}_r e termina-se a iteração (Figura 4.6).

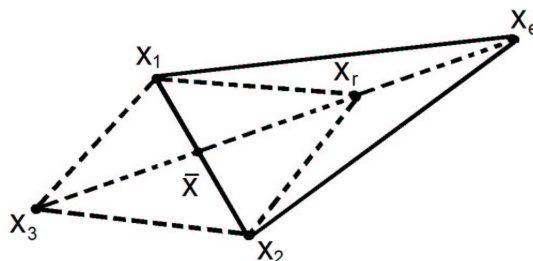


Figura 4.6: Configuração da figura geométrica após a etapa de *expansão*.

4. **Contração:** Se $f_n \triangleleft f_r$, então calcula-se o ponto de contração (\mathbf{x}_{ce} ou \mathbf{x}_{ci}) de acordo com o resultado da comparação entre f_r e f_{n+1} :

- Se $f_r \triangleleft f_{n+1}$, então procede-se com a *contração externa*:

$$\mathbf{x}_{ce} = \bar{\mathbf{x}} + \gamma(\mathbf{x}_r - \bar{\mathbf{x}}). \quad (4.8)$$

Caso a avaliação de $f_{ce} = f(\mathbf{x}_{ce})$ implique $f_{ce} \triangleleft f_r$, aceita-se o ponto \mathbf{x}_{ce} e termina-se a iteração (Figura 4.7).

⁹Uma vez que a etapa de ordenação fez com que o pior ponto fosse representado em \mathbf{x}_{n+1} .

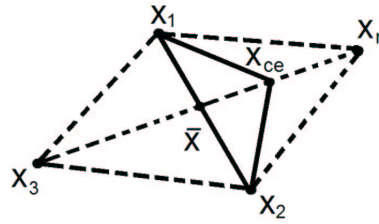


Figura 4.7: Configuração da figura geométrica após a etapa de *contração externa*.

- Se $f_{n+1} \triangleleft f_r$, então procede-se com a *contração interna*:

$$\mathbf{x}_{ci} = \bar{\mathbf{x}} - \gamma(\bar{\mathbf{x}} - \mathbf{x}_{n+1}). \quad (4.9)$$

Caso a avaliação de $f_{ci} = f(\mathbf{x}_{ci})$ implique $f_{ci} \triangleleft f_{n+1}$, aceita-se o ponto \mathbf{x}_{ci} e termina-se a iteração (Figura 4.8).

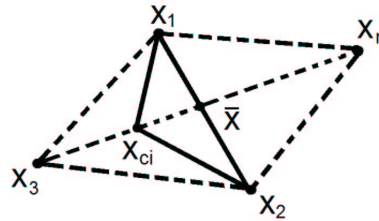


Figura 4.8: Configuração da figura geométrica após a etapa de *contração interna*.

5. **Encolhimento:** Avalia-se f para os n pontos

$$\mathbf{v}_i = \mathbf{x}_1 + \sigma(\mathbf{x}_i - \mathbf{x}_1), \quad \forall i = 2, \dots, n+1. \quad (4.10)$$

Os pontos (desordenados) da matriz simplex para a próxima iteração consistem em $\mathbf{x}_1, \mathbf{v}_1, \dots, \mathbf{v}_{n+1}$ (Figura 4.9).

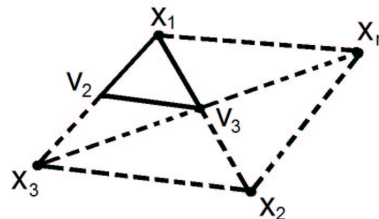


Figura 4.9: Configuração da figura geométrica após a etapa de *encolhimento*.

A análise de desempenho do algoritmo de Nelder-Mead atribui a ele três características principais:

1. implementação simples;
2. rápida convergência (quando alcançada);
3. baixa robustez.

O primeiro aspecto mencionado já havia sido citado mas merece destaque. Por se basear no método de programação linear *simplex*, o algoritmo de Nelder-Mead utiliza mecanismos bastante simples de busca por mínimos da função objetivo. A representação da matriz como um objeto geométrico de dimensão igual ao espaço de busca torna o entendimento do método bastante intuitivo.

A rápida convergência consiste em outro aspecto notável do método. Devido aos mecanismos de *expansão*, *contração* e *encolhimento*, acrescidos ao *simplex* original, a localização de um mínimo da função objetivo é feita de maneira bastante eficiente no que diz respeito ao número de iterações. Além disso, o fato de não utilizar qualquer tipo de derivada da função objetivo torna o tempo de execução de cada iteração bastante reduzido.

4.4 Direção de Busca

Embora de implementação tipicamente mais complexa que os algoritmos de *Busca Direta*, os algoritmos de *Direção de Busca* representam uma classe de métodos de otimização baseados em princípios mais intuitivos. Basicamente, utiliza-se informação da derivada da função objetivo em um ponto inicial com o intuito de determinar o próximo ponto a ser inspecionado. O novo ponto de origem passa então a ser aquele apontado pelo *vetor direção* e o processo prossegue recursivamente. Devido a isso, os algoritmos de *Direção de Busca* são comumente denominados algoritmos de *primeira ordem*.

A cada iteração de um algoritmo de *Direção de Busca*, parte-se de um ponto θ_k e procura-se por um ponto θ_{k+1} “melhor” que θ_k segundo a seguinte relação geral:

$$\theta_{k+1} = \theta_k + \alpha_k \mathbf{d}_k \quad (4.11)$$

em que α_k consiste em um parâmetro ajustável responsável pelo passo do processo de busca na direção \mathbf{d}_k , a cada iteração k .

Tipicamente, o próprio ajuste de α_k envolve a resolução de um problema de otimização¹⁰ e é definido de acordo com o algoritmo utilizado. O passo α_k é responsável pela abrangência dos métodos de *Direção de Busca*, no sentido de viabilizar sua convergência para um ótimo global (Narendra, 1991).

Por utilizarem informações de derivada, a convergência dos métodos de *Direção de Busca* depende da diferenciabilidade da função objetivo. Devido a isso, espaços de busca discretos também representam limitações para essa classe de algoritmos.

4.4.1 Método do Gradiente

Assim como qualquer algoritmo recursivo, o algoritmo do *gradiente* inicia suas iterações a partir de uma condição inicial e busca atingir algum critério de convergência à medida que processa novas iterações. O aspecto que diferencia o método do *gradiente* dentre os métodos de *direção de busca* é a maneira com que o vetor direção \mathbf{d}_k é calculado. Para isso, utiliza-se a aproximação linear local da função a ser minimizada $c(\boldsymbol{\theta})$ no ponto $\boldsymbol{\theta}_{k+1}$, dada por

$$c(\boldsymbol{\theta}_{k+1}) = c(\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k) \approx c_k + \mathbf{g}_{c_k}^T \delta\boldsymbol{\theta}_k. \quad (4.12)$$

Analisando-se (4.12), observa-se que uma maneira intuitiva de se minimizar o valor da função objetivo com a execução das iterações consiste em determinar valores de $\delta\boldsymbol{\theta}_k$ de tal modo que $\mathbf{g}_{c_k}^T \delta\boldsymbol{\theta}_k$ tenha grande valor absoluto e seja negativo. Contudo, alguma normalização deve ser imposta sobre o valor do passo $\delta\boldsymbol{\theta}_k$ com o intuito de evitar que a diminuição no valor da função objetivo $\mathbf{g}_{c_k}^T \delta\boldsymbol{\theta}_k$ seja arbitrariamente grande (Gill et al., 1981). Essa normalização deve ser tal que $\delta\boldsymbol{\theta}_k$ seja a solução do problema

$$\min_{\delta\boldsymbol{\theta}_k \in \mathbb{R}^n} \frac{\mathbf{g}_{c_k}^T \delta\boldsymbol{\theta}_k}{\|\delta\boldsymbol{\theta}_k\|}, \quad (4.13)$$

em que a solução depende da norma utilizada. Caso a norma seja definida por uma determinada matriz simétrica definida positiva P tal que

$$\|\delta\boldsymbol{\theta}_k\|_P = (\delta\boldsymbol{\theta}_k^T P \delta\boldsymbol{\theta}_k)^{\frac{1}{2}},$$

¹⁰Relativamente menos complexo, já que se trata de um problema de otimização unidimensional (sobre a direção \mathbf{d}_k). Um dos métodos comumente utilizados é o da *Seção Aurea* (He et al., 2002).

então a solução de (4.13) consiste em

$$\delta\boldsymbol{\theta}_k = -P^{-1}\mathbf{g}_{c_k}. \quad (4.14)$$

Mas, como a o método do *gradiente* prevê a utilização da norma euclidiana, a solução de (4.13) e conseqüentemente o passo do algoritmo a cada iteração são tais que

$$\alpha_k \mathbf{d}_k = \delta\boldsymbol{\theta}_k \triangleq -\alpha_k \mathbf{g}_{c_k} \quad (4.15)$$

em que α_k consiste no valor do passo dado na direção \mathbf{d}_k .

Além disso, no método do *gradiente*, considera-se que a função a ser otimizada possa ser aproximada por uma função quadrática (A.36), de modo que

$$c(\boldsymbol{\theta}) \approx \phi(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T G \boldsymbol{\theta} + \mathbf{a}^T \boldsymbol{\theta} + b \quad (4.16)$$

e a determinação do valor de α_k em (4.11) e (4.15) é feita levando-se em consideração o fato de o mínimo encontrado no processo de minimização unidirecional (na direção \mathbf{d}_k) representar o ponto no qual o vetor gradiente encontra-se perpendicular à sua posição inicial¹¹ (Figura 4.10).

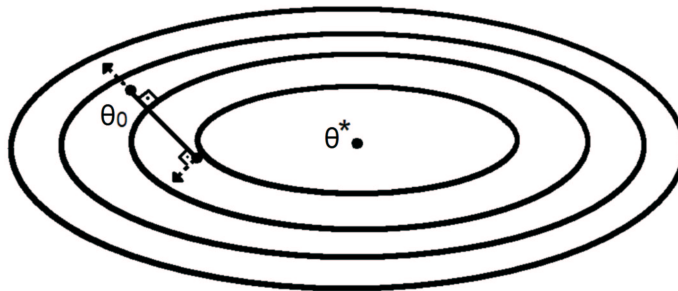


Figura 4.10: Ortogonalidade entre os vetores gradiente no ponto inicial e no ponto final da busca unidirecional.

¹¹Se \mathbf{u} e \mathbf{v} são perpendiculares, então seu produto interno $\mathbf{u} \cdot \mathbf{v} = 0$ (Santos, 2004).

Com isso e com a equação (A.39), tem-se

$$\begin{aligned}
\mathbf{g}_{\phi_{k+1}}^T \mathbf{g}_{\phi_k} &= 0 \\
(G_\phi \boldsymbol{\theta}_{k+1} + \mathbf{a})^T \mathbf{g}_{\phi_k} &= 0 \\
(G_\phi (\boldsymbol{\theta}_k + \alpha_k \mathbf{g}_{\phi_k}) + \mathbf{a})^T \mathbf{g}_{\phi_k} &= 0 \\
(G_\phi \boldsymbol{\theta}_k + \mathbf{a})^T \mathbf{g}_{\phi_k} + \alpha_k (G_\phi \mathbf{g}_{\phi_k})^T \mathbf{g}_{\phi_k} &= 0 \\
(G_\phi \boldsymbol{\theta}_k + \mathbf{a})^T \mathbf{g}_{\phi_k} &= -\alpha_k (G_\phi \mathbf{g}_{\phi_k})^T \mathbf{g}_{\phi_k} \\
\mathbf{g}_{\phi_k}^T \mathbf{g}_{\phi_k} &= -\alpha_k \mathbf{g}_{\phi_k}^T (G_\phi \mathbf{g}_{\phi_k}) \\
\alpha_k &= -\frac{\mathbf{g}_{\phi_k}^T \mathbf{g}_{\phi_k}}{\mathbf{g}_{\phi_k}^T G_\phi \mathbf{g}_{\phi_k}}
\end{aligned} \tag{4.17}$$

Por utilizar apenas os dois primeiros termos da representação em séries de Taylor da função objetivo¹², o método do *gradiente* apresenta problemas de velocidade de convergência. No caso específico de uma função cujas curvas de nível sejam representadas por elipsóides de grande excentricidade (“achata-dos”), a convergência é lenta e apresenta um comportamento oscilatório de pequenos passos (Figura 4.11). Além disso, deve-se observar que o algoritmo do *gradiente* convergirá para o ponto $\boldsymbol{\theta}^*$ desde que o ponto inicial $\boldsymbol{\theta}_0$ seja definido na bacia de atração de $\boldsymbol{\theta}^*$.

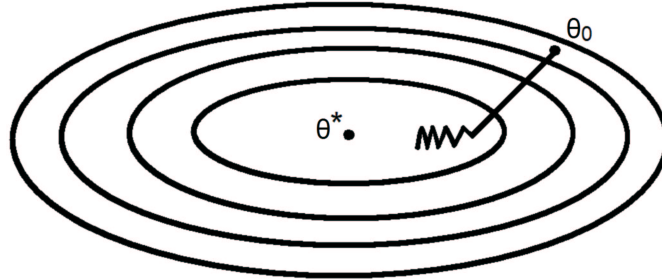


Figura 4.11: Dificuldade de convergência do método do gradiente.

4.4.2 Método do Gradiente Conjugado

O método do *Gradiente Conjugado* consiste em um aprimoramento do método do *gradiente*. A implementação dessa melhoria é baseada na observação de que o método do *gradiente* tende a evoluir ao ótimo do problema

¹²O que caracteriza o método do gradiente como um algoritmo de *primeira ordem*.

utilizando uma mesma direção diversas vezes (Figura 4.11) (Press et al., 1992). Com isso, procura-se utilizar um conjunto de direções ortogonais nas quais apenas um único passo será dado.

Para cada uma das n direções \mathbf{d}_k , o passo deve ser tal que o ponto resultante tenha mesma coordenada d_k que o ponto ótimo. Essa igualdade deve ser garantida no sistema de coordenadas definido pelos vetores $\hat{\mathbf{i}}_{d_0} = \mathbf{d}_0/\|\mathbf{d}_0\|$, $\hat{\mathbf{i}}_{d_1} = \mathbf{d}_1/\|\mathbf{d}_1\| \cdots \hat{\mathbf{i}}_{d_{n-1}} = \mathbf{d}_{n-1}/\|\mathbf{d}_{n-1}\|$.

Por exemplo, considere-se um problema cujo espaço de variáveis tenha dimensão $n = 3$ ($\boldsymbol{\theta} = [\theta_1 \ \theta_2 \ \cdots \ \theta_n]$). Considere-se também que o sistema de coordenadas formado pelos vetores $\hat{\mathbf{i}}_{d_0}$, $\hat{\mathbf{i}}_{d_1}$ e $\hat{\mathbf{i}}_{d_2}$ tenha sempre a origem no ponto atual ($\boldsymbol{\theta}_k$) e que a rotação desses eixos em relação ao sistema de coordenadas $\theta_1 \times \theta_2 \times \theta_3$ seja constante¹³. Com isso:

1. o primeiro passo deve ser tal que o ponto resultante $\boldsymbol{\theta}_1$ faça parte do plano $\mathbf{d}_1 \times \mathbf{d}_2$;
2. em seguida, o passo na direção \mathbf{d}_1 deve ser tal que o ponto resultante $\boldsymbol{\theta}_2$ esteja sobre o eixo \mathbf{d}_2 ;
3. finalmente, o terceiro passo (na direção \mathbf{d}_2) deve ser igual a distância entre os pontos $\boldsymbol{\theta}_2$ e $\boldsymbol{\theta}^*$ ($\mathbf{e}_2 = \boldsymbol{\theta}_2 - \boldsymbol{\theta}^*$).

A Figura 4.12 ilustra o procedimento para um espaço de busca de duas dimensões ($n = 2$). No caso representado, utilizam-se como direções os próprios eixos do espaço de variáveis de otimização.

A determinação do valor de α_k em (4.11) para o algoritmo do *gradiente Conjugado* resulta da utilização da definição desse método: Qualquer direção \mathbf{d}_k utilizada deve ser ortogonal ao vetor distância \mathbf{e}_{k+1} (Figura 4.12). Dessa maneira, garante-se que não haverá a necessidade de se utilizar essa direção novamente. Com isso,

$$\begin{aligned} \mathbf{d}_k^T \mathbf{e}_{k+1} &= 0 \\ \mathbf{d}_k^T (\mathbf{e}_k + \alpha_k \mathbf{d}_k) &= 0 \\ \alpha_k &= -\frac{\mathbf{d}_k^T \mathbf{e}_k}{\mathbf{d}_k^T \mathbf{d}_k}, \end{aligned} \quad (4.18)$$

em que a segunda igualdade é resultado direto de (4.11).

¹³As iterações do algoritmo não rotacionam o sistema de coordenadas $\hat{\mathbf{i}}_{d_0} \times \hat{\mathbf{i}}_{d_1} \times \hat{\mathbf{i}}_{d_2}$.

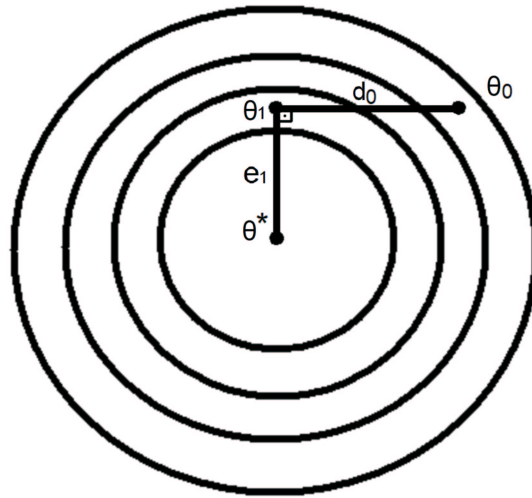


Figura 4.12: Direções de busca paralelas ao eixo do método do gradiente Conjugado

Contudo, a equação (4.18) tem pouca utilidade, uma vez que depende do valor de \mathbf{e}_k e caso esse valor fosse conhecido, não haveria problema a ser resolvido. A solução para esse impasse está na utilização de direções representadas por vetores *A-ortogonais*. Dois vetores \mathbf{d}_k e \mathbf{d}_j são considerados *A-ortogonais* ou *conjugados* se (Chen, 1999)

$$\mathbf{d}_k^T A \mathbf{d}_j = 0. \quad (4.19)$$

A Figura 4.13 ilustra o conceito de vetores *A-ortogonais*. A modificação visual é análoga ao “esticamento” do espaço em que os vetores são representados. Esse efeito é causado pela utilização de uma matriz $A \neq I$ em (4.19).

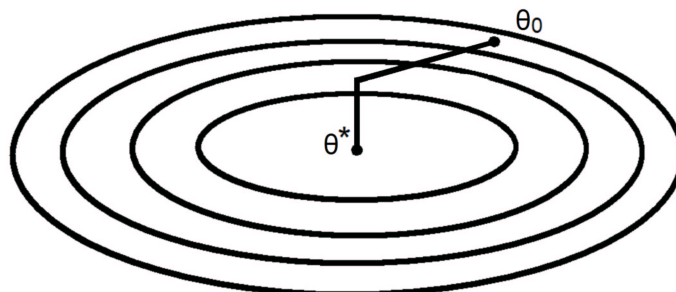


Figura 4.13: Direções de busca do método do gradiente Conjugado

Para que os conceitos utilizados pelo algoritmo continuem válidos, deve-se garantir, de forma semelhante ao que foi feito em (4.18), que os vetores \mathbf{d}_k e \mathbf{e}_{k+1} sejam A -ortogonais. De maneira similar ao que ocorre no método do *gradiente*¹⁴ essa condição de ortogonalidade equivale a determinar o ponto ótimo na direção \mathbf{d}_k :

$$\begin{aligned}\frac{d}{d\alpha}c(\boldsymbol{\theta}_{k+1}) &= 0 \\ c'(\boldsymbol{\theta}_{k+1})^T \frac{d}{d\alpha}\boldsymbol{\theta}_{k+1} &= 0 \\ -\mathbf{g}_{c_{k+1}}^T \mathbf{d}_k &= 0 \\ \mathbf{d}_k^T A\mathbf{e}_{k+1} &= 0\end{aligned}\tag{4.20}$$

Segundo (4.18), o valor do passo α_k é então definido por

$$\begin{aligned}\alpha_k &= -\frac{\mathbf{d}_k^T A\mathbf{e}_k}{\mathbf{d}_k^T A\mathbf{d}_k} \\ &= \frac{\mathbf{d}_k^T \mathbf{g}_{c_k}}{\mathbf{d}_k^T A\mathbf{d}_k},\end{aligned}$$

uma vez que, a partir de (4.20), observa-se que

$$-\mathbf{g}_{c_{k+1}}^T \mathbf{d}_k = -\mathbf{d}_k^T \mathbf{g}_{c_{k+1}} = \mathbf{d}_k^T A\mathbf{e}_{k+1} = 0,\tag{4.21}$$

o que implica $A\mathbf{e}_{k+1} = -\mathbf{g}_{c_{k+1}}$ e, portanto, $A\mathbf{e}_k = -\mathbf{g}_{c_k}$.

4.4.3 Método do Gradiente Conjugado Pré - Condicionado

Pré-condicionamento consiste em uma técnica de redução do número de condicionamento de uma determinada matriz. A utilização dessa técnica em algoritmos baseados no método do *Gradiente* tem por finalidade o aperfeiçoamento da aproximação da função objetivo por uma função quadrática. De fato, busca-se uma representação aproximadamente esférica do problema abordado.

A motivação para a utilização da técnica do pré-condicionamento é simples. Suponha-se que se deseja resolver

$$A\boldsymbol{\theta} = \mathbf{b},\tag{4.22}$$

¹⁴Note que a única diferença entre os métodos está na especificação da matriz A .

para uma matriz A mal-condicionada¹⁵ (*número de condicionamento* $\kappa(A) \gg 1$).

Nesse contexto, o valor de $\boldsymbol{\theta}$ obtido com a resolução de (4.22) estaria associado a uma grande imprecisão, uma vez que o cálculo da inversa de A depende de $\kappa(A) \approx 1$. A solução proposta pelo método para esse problema consiste na definição de uma matriz M simétrica, definida positiva, que se aproxime de A ($M \approx A$) mas que seja mais facilmente invertível ($\kappa(M) \approx 1$). Com isso, a equação (4.22) pode ser transformada em:

$$M^{-1}A\boldsymbol{\theta} = M^{-1}\mathbf{b}. \quad (4.23)$$

Resta então definir M de tal modo que $\kappa(M^{-1}A) \ll \kappa(A)$. Contudo, essa não é uma tarefa trivial, uma vez que $M^{-1}A$ geralmente não é simétrica (ou definida positiva) ainda que M e A os sejam. Mas, para toda matriz M simétrica definida positiva, existe uma matriz¹⁶ E tal que $M = EE^T$. Com isso, (4.23) pode ser transformado em:

$$E^{-1}AE^{-T}\bar{\boldsymbol{\theta}} = E^{-1}\mathbf{b}, \quad (4.24)$$

em que $\bar{\boldsymbol{\theta}} = E^T\boldsymbol{\theta}$ e resolve-se primeiro para $\bar{\boldsymbol{\theta}}$ e em seguida para $\boldsymbol{\theta}$.

A mudança de variáveis em (4.24) se justifica pela constatação de que $M^{-1}A$ e $E^{-1}AE^{-T}$ possuem os mesmos autovalores. Desse modo, garante-se $E^{-1}AE^{-T}$ simétrica definida positiva, uma vez que $M^{-1}A$ o é (por construção). A prova para essa afirmação é simples. Se \mathbf{v} é um autovetor de $M^{-1}A$ associado a um autovalor λ , então $E^T\mathbf{v}$ é um autovetor de $E^{-1}AE^{-T}$ associado a um autovalor λ :

$$\begin{aligned} (E^{-1}AE^{-T})(E^T\mathbf{v}) &= E^{-1}A\mathbf{v} = (E^TE^{-T})E^{-1}A\mathbf{v} = \\ &= E^T(E E^T)^{-1}A\mathbf{v} = E^TM^{-1}A\mathbf{v} = \lambda E^T\mathbf{v}. \end{aligned} \quad (4.25)$$

Intuitivamente, o pré-condicionamento busca deformar a representação quadrática com o intuito de torná-la mais esférica (A.43). Uma maneira ideal de se aproximar os autovalores da nova representação consistiria em definir $M = A$. Com isso, $M^{-1}A$ seria igual à matriz Identidade e, portanto, possuiria número de condicionamento unitário. Contudo, essa definição implicaria a necessidade de resolução da equação $M\boldsymbol{\theta} = \mathbf{b}$, o que, pela própria

¹⁵Subseção A.6.1

¹⁶Essa matriz não é necessariamente única e pode ser obtida via fatoração de Cholesky, por exemplo.

definição de M , não traria qualquer tipo de benefício.

O procedimento mais simples adotado para determinação de M consiste em sua definição como sendo igual a uma matriz diagonal cujos elementos sejam iguais aos elementos da diagonal de A :

$$M = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}, \quad (4.26)$$

mas, apesar de não haver dificuldades relativas à inversão de M , essa definição tipicamente resulta em uma transformação pouco significativa do problema original.

Um método mais sofisticado para determinação da matriz M consiste na utilização da *Fatoração Incompleta de Cholesky*. Esse procedimento garante a criação de uma matriz \hat{L} tal que $A = \hat{L}\hat{L}^T$ com elementos diferentes de zero nas mesmas posições que em A (Wang et al., 1993). De fato, independentemente do método utilizado para definição da matriz M , o uso de algum tipo de pré-condicionamento é prática comum na resolução de problemas com algoritmos baseados no método do *Gradiente*.

4.5 Métodos de Newton

O método de Newton consiste na principal abordagem com utilização de aproximações quadráticas da função objetivo para resolução de problemas de otimização irrestritos. Os principais motivos para a escolha da representação da função objetivo por funções quadráticas são as relativas simplicidade e abrangência atribuídas aos algoritmos.

Caso as derivadas de segunda ordem da função objetivo $c(\boldsymbol{\theta})$ (sob a forma da matriz Hessiana G_c) possam ser calculadas, uma aproximação quadrática em torno de $\boldsymbol{\theta}_k$ dessa função pode ser obtida com a utilização dos três primeiros termos da expansão em séries de Taylor de $c(\boldsymbol{\theta})$:

$$c(\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k) \approx c_k + \mathbf{g}_{c_k}^T \delta\boldsymbol{\theta}_k + \frac{1}{2} \delta\boldsymbol{\theta}_k^T G_{c_k} \delta\boldsymbol{\theta}_k, \quad (4.27)$$

em que $c_k = c(\boldsymbol{\theta}_k)$, \mathbf{g}_{c_k} é o vetor gradiente e G_{c_k} é a Hessiana de c , ambos avaliados em $\boldsymbol{\theta}_k$.

Por se tratar da aproximação quadrática em torno de um ponto específico $\boldsymbol{\theta}_k$, é conveniente expressar (4.27) como função do *passo* $\delta\boldsymbol{\theta}_k$ e não do ponto $\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k$ propriamente dito. Determinar o ótimo do lado direito de (4.27) equivale a determinar o passo ótimo $\delta\boldsymbol{\theta}_k^*$ tal que a função quadrática

$$\phi(\delta\boldsymbol{\theta}_k) \triangleq \mathbf{g}_{c_k}^T \delta\boldsymbol{\theta}_k + \frac{1}{2} \delta\boldsymbol{\theta}_k^T G_{c_k} \delta\boldsymbol{\theta}_k \quad (4.28)$$

seja otimizada. Ou, de forma compacta, deve-se resolver o problema de otimização

$$\min_{\delta\boldsymbol{\theta}_k \in \mathbb{R}^n} / \max_{\delta\boldsymbol{\theta}_k \in \mathbb{R}^n} \mathbf{g}_{c_k}^T \delta\boldsymbol{\theta}_k + \frac{1}{2} \delta\boldsymbol{\theta}_k^T G_{c_k} \delta\boldsymbol{\theta}_k. \quad (4.29)$$

A primeira condição para que $\delta\boldsymbol{\theta}_k^*$ seja um mínimo de $\phi(\delta\boldsymbol{\theta}_k)$ consiste em $\nabla\phi(\delta\boldsymbol{\theta}_k^*) = \mathbf{0}$. Mas, conforme o desenvolvimento na seção A.5

$$\nabla\phi(\delta\boldsymbol{\theta}_k) = G_{c_k} \delta\boldsymbol{\theta}_k + \mathbf{g}_{c_k}.$$

Com isso, tem-se

$$\nabla\phi(\delta\boldsymbol{\theta}_k^*) = G_{c_k} \delta\boldsymbol{\theta}_k^* + \mathbf{g}_{c_k} = \mathbf{0}, \quad (4.30)$$

o que implica que o método de Newton deve resolver o sistema linear

$$G_{c_k} \delta\boldsymbol{\theta}_k^* = -\mathbf{g}_{c_k}. \quad (4.31)$$

Um algoritmo de otimização em que $\delta\boldsymbol{\theta}_k^*$ é definido como em (4.31) é dito ser baseado no método de Newton. Além disso, a solução $\delta\boldsymbol{\theta}_k^*$ é denominada *direção de Newton*.

Finalmente, a segunda condição exige que G_{c_k} seja definida positiva em $\delta\boldsymbol{\theta}_k^*$. Caso essa condição seja satisfeita, o algoritmo converge para o mínimo da função quadrática em uma única iteração¹⁷ a partir de qualquer ponto inicial $\boldsymbol{\theta}_k$. Devido a isso, espera-se bom desempenho dos algoritmos baseados no método de Newton caso (4.27) seja de fato uma boa aproximação.

Em detrimento às características que conferem bom desempenho aos algoritmos relacionados, a formulação original do método de Newton tipicamente não é utilizada. O motivo para isso encontra-se no fato de a matriz Hessiana eventualmente não ser positiva definida (especialmente em pontos distantes da solução $\boldsymbol{\theta}_k^* = \boldsymbol{\theta}_k + \delta\boldsymbol{\theta}_k^*$). Essa é a motivação para a elaboração de métodos baseados em modificações do método de Newton, como os algoritmos de Gauss-Newton (Seção 4.5.1) e de Levenberg-Marquardt (Seção 4.6.1).

¹⁷Observe-se que a aproximação quadrática (4.27) se baseia no fato de a matriz Hessiana G_{c_k} ser constante.

4.5.1 Método de Gauss-Newton

O método de Gauss-Newton consiste em uma adaptação do algoritmo de Newton com o intuito de prevenir problemas relativos ao cálculo e particularidades¹⁸ da Hessiana da função objetivo. Essa adaptação consiste na transformação do método de Newton no sentido de resolver um problema de mínimos quadrados. Com isso, substituindo-se as definições de vetor gradiente (4.37) e matriz Hessiana (4.38) segundo o método de mínimos quadrados nas equações de Newton (4.31) obtém-se

$$(J_{f_k}^T J_{f_k} + Q_{f_k}) \delta \boldsymbol{\theta}_k^* = J_{f_k}^T \cdot f(\boldsymbol{\theta}_k) \quad (4.32)$$

O algoritmo de Gauss-Newton parte da constatação de que, a cada iteração k com o objetivo de minimizar a função quadrática $\phi(\delta \boldsymbol{\theta}_k)$ (4.28), o resíduo $\|f(\boldsymbol{\theta}_k)\|$ tende a zero à medida que $\boldsymbol{\theta}_k$ se aproxima da solução $\boldsymbol{\theta}_k^*$. Com isso, Q_{f_k} também tende a zero e a direção de Newton pode ser aproximada por

$$J_{f_k}^T J_{f_k} \delta \boldsymbol{\theta}_k^* = J_{f_k}^T f(\boldsymbol{\theta}_k). \quad (4.33)$$

Segundo (4.36), a formulação (4.33) implica a resolução de um problema de mínimos quadrados definido por

$$\min_{\delta \boldsymbol{\theta} \in \mathbb{R}^n} / \max_{\delta \boldsymbol{\theta} \in \mathbb{R}^n} \frac{1}{2} \|J_{f_k} \delta \boldsymbol{\theta}_k + f(\boldsymbol{\theta}_k)\|^2. \quad (4.34)$$

Apesar de propor a criação de uma matriz $J_{f_k}^T J_{f_k}$ para a resolução de (4.33), a maior parte das versões de algoritmos de Gauss-Newton evita esse procedimento. O motivo para isso é simples: O número de condicionamento de $J_{f_k}^T J_{f_k}$ é igual ao quadrado daquele relativo a J_{f_k} , o que implica a inserção de erros numéricos desnecessários no cálculo do vetor de direção de busca (neste caso, denominado vetor de *direção de Gauss-Newton*).

Outro problema freqüentemente enfrentado durante a utilização do algoritmo de Gauss-Newton refere-se ao *posto* da matriz Jacobiana, $\rho(J_{f_k})$. Caso J_{f_k} não possua posto completo em colunas¹⁹, a solução do problema (4.34) não é única e, tipicamente, opta-se pela solução com menor norma euclídeana. Esse vetor pode ser encontrado com a utilização de métodos de fatoração sobre J_{f_k} ou com a sua decomposição em valores singulares.

¹⁸Como o fato de não ser definida positiva.

¹⁹O que implica $\rho(J_{f_k}) < n$.

4.6 Mínimos Quadrados

Em um grande número de problemas práticos, a função objetivo $c(\boldsymbol{\theta})$ assume a forma da soma dos quadrados de funções não-lineares:

$$c(\boldsymbol{\theta}) \triangleq c_{\text{MQ}}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^N f(\boldsymbol{\theta}, t_i)^2 = \frac{1}{2} \|f(\boldsymbol{\theta}, t)\|^2 \quad (4.35)$$

em que²⁰ $f(\boldsymbol{\theta}, t_i) : \mathbb{R}^n \rightarrow \mathbb{R}$, $f(\boldsymbol{\theta}, t) : \mathbb{R}^n \rightarrow \mathbb{R}^N$, n é a dimensão de $\boldsymbol{\theta}$ e N é o tamanho da série (temporal) correspondente à simulação de $f(\boldsymbol{\theta}, t)$. Além disso, o fator $\frac{1}{2}$ é inserido para evitar o aparecimento de um fator 2 após o cálculo das derivadas de $c_{\text{MQ}}(\boldsymbol{\theta})$.

Problemas cujas soluções consistem na determinação de pontos ótimos de $c_{\text{MQ}}(\boldsymbol{\theta})$ são típicos em cenários de estimação de parâmetros via ajuste de curvas (*curve fitting*). Por exemplo, considere-se que um sistema real possua apenas um sinal de saída, $y(t)$, e que seja modelado por $\phi(\boldsymbol{\theta}, t)$. Nesse caso, a definição $f(\boldsymbol{\theta}, t_i) = \phi(\boldsymbol{\theta}, t_i) - y(t_i)$ em (4.35) vincula a estimação do vetor de parâmetros $\boldsymbol{\theta} = [\theta_1 \cdots \theta_n]^T$ à minimização de $c_{\text{MQ}}(\boldsymbol{\theta})$, conforme a Figura 4.14.

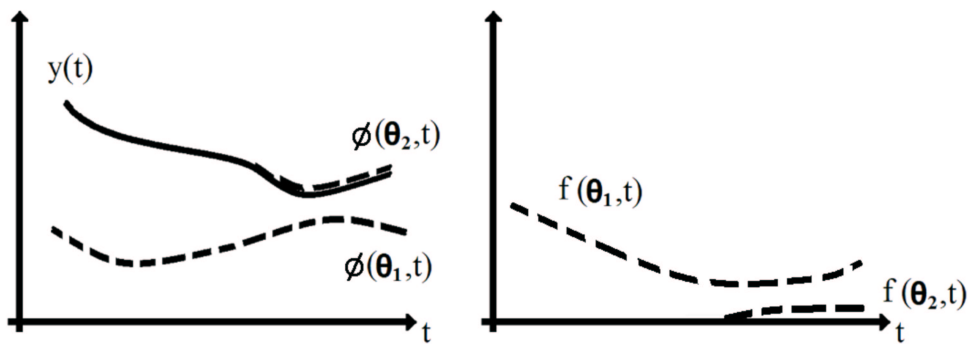


Figura 4.14: Representação de um problema de estimação de parâmetros via ajuste de curvas.

Devido à formulação (4.35), o método de estimação de parâmetros via minimização de $c_{\text{MQ}}(\boldsymbol{\theta})$ é denominado método de *Mínimos Quadrados* (Aguirre,

²⁰Uma observação sobre notação: Em (4.35), $f(\boldsymbol{\theta}, t)$ foi mantido em fonte normal, embora se trate de uma grandeza vetorial. Essa escolha é feita com o intuito de apenas utilizar letras em negrito para funções vetoriais de forma estrita. Neste caso, $f(\boldsymbol{\theta}, t)$ consiste em uma função escalar que mapeia o par $(\boldsymbol{\theta}, t_i)$ em um único valor.

2004). Esse método se baseia na resolução de um problema de otimização baseado na determinação de um vetor de parâmetros ótimo $\boldsymbol{\theta}^*$ tal que

$$\boldsymbol{\theta}^* = \arg_{\boldsymbol{\theta}} \min \|\phi(\boldsymbol{\theta}, t_i) - y(t_i)\|^2, \quad (4.36)$$

o que, de forma simplificada, implica a determinação de um vetor de parâmetros $\boldsymbol{\theta}^*$ tal que $\phi(\boldsymbol{\theta}^*, t)$ seja o mais próximo possível²¹ daquele correspondente ao sistema real responsável pela criação da série $y(t)$.

Embora o problema (4.36) possa ser resolvido por, em princípio, qualquer método de otimização irrestrito, algumas particularidades da formulação (4.35) tornam conveniente a utilização de métodos específicos para a resolução de problemas de Mínimos Quadrados. Em particular, o vetor gradiente e a matriz Hessiana de $c_{\text{MQ}}(\boldsymbol{\theta})$ podem ser definidos por (Subseção A.3.1):

$$\mathbf{g}_{c_{\text{MQ}}}(\boldsymbol{\theta}) = J_{\phi(t)}(\boldsymbol{\theta})^T f(\boldsymbol{\theta}) \quad (4.37)$$

$$G_{c_{\text{MQ}}}(\boldsymbol{\theta}) = J_{\phi(t)}(\boldsymbol{\theta})^T J_{\phi(t)}(\boldsymbol{\theta}) + Q_{\phi}(\boldsymbol{\theta}, t) \quad (4.38)$$

respectivamente, em que $J_{\phi(t)}(\boldsymbol{\theta})$ é a Jacobiana de $\phi(\boldsymbol{\theta})$ e

$$Q_{\phi}(\boldsymbol{\theta}) = \sum_{i=1}^N \frac{\partial^2 \phi(\boldsymbol{\theta}, t_i)}{(\partial \boldsymbol{\theta})^2} (\phi(\boldsymbol{\theta}, t_i) - y(t_i)),$$

o que indica uma importante característica da matriz Hessiana (4.38), específica de problemas de Mínimos Quadrados: $G_{c_{\text{MQ}}}(\boldsymbol{\theta}, t)$ contém informações relativas à primeira e à segunda derivadas da função objetivo.

Métodos de Mínimos Quadrados são tipicamente baseados na premissa de o termo $J_{\phi(t)}(\boldsymbol{\theta})^T J_{\phi(t)}(\boldsymbol{\theta})$ dominar o termo $Q_{\phi}(\boldsymbol{\theta})$, o que simplifica o cálculo da matriz Hessiana:

$$G_{c_{\text{MQ}}}(\boldsymbol{\theta}) \approx J_{\phi(t)}(\boldsymbol{\theta})^T J_{\phi(t)}(\boldsymbol{\theta}). \quad (4.39)$$

Contudo, não se pode assumir esse fato caso os resíduos $\|f(\boldsymbol{\theta}^*)\|$ observados na solução do problema possuam valores significativos²². Nesse caso, devem-se utilizar outros métodos de otimização irrestritos. Em detrimento a esse fato, boa parte dos problemas de otimização apresenta valores de resíduos suficientemente pequenos a ponto de justificarem a utilização de métodos de Mínimos Quadrados.

²¹Portanto, ótimo de acordo com a métrica definida em (4.36).

²²Por valores significativos, entende-se $\|f(\boldsymbol{\theta}^*)\|$ comparável ao maior autovalor de $J_f(\boldsymbol{\theta}^*)^T J_f(\boldsymbol{\theta}^*)$ (Gill et al., 1981).

4.6.1 Método de Levenberg-Marquardt

Inicialmente publicado por Kenneth Levenberg em 1944 (Levenberg, 1944) e aprimorado por Donald Marquardt em 1963 (Marquardt, 1963), o método de Levenberg-Marquardt tem ampla utilização nas mais diversas áreas do conhecimento científico. Trata-se de um algoritmo de otimização de classificação intermediária, simultaneamente baseado nos métodos de *Gauss-Newton* e do *Gradiente*.

Essa característica híbrida confere ao algoritmo de Levenberg-Marquardt um desempenho tendencialmente menos susceptível às condições iniciais se comparado com os métodos de *Gradiente* e de *Gauss-Newton*. Por outro lado, sua velocidade de convergência é tipicamente superada pela dos algoritmos pertencentes à última classe.

De fato, quando a solução atual está distante do ponto mínimo local, o algoritmo se comporta de forma lenta e semelhante a um algoritmo do *Gradiente*. Caso contrário, sua convergência se faz de forma mais veloz e o algoritmo se comporta como um algoritmo de *Gauss-Newton* (Franchois and Pichot, 1997).

O algoritmo de Levenberg-Marquardt é baseado na resolução de um problema de *mínimos quadrados*. A função custo correspondente à iteração k é então definida por

$$c(\hat{\boldsymbol{\theta}}_k) = \frac{1}{2} \sum_{j=1}^m s_j(\hat{\boldsymbol{\theta}}_k)^2 = \frac{1}{2} \mathbf{s}^T(\hat{\boldsymbol{\theta}}_k) \mathbf{s}(\hat{\boldsymbol{\theta}}_k) = \frac{1}{2} \|\mathbf{s}(\hat{\boldsymbol{\theta}}_k)\|^2, \quad (4.40)$$

em que s_j consiste em um índice que quantifica a qualidade da estimação do vetor de parâmetros $\boldsymbol{\theta}$, representada por $\hat{\boldsymbol{\theta}}$, no que diz respeito à função $f_j(\boldsymbol{\theta})$ ²³ (Góes et al., 2004):

$$s_j(\hat{\boldsymbol{\theta}}_k) = \sum_{i=1}^n \xi_j(i, \hat{\boldsymbol{\theta}}_k)^2 = \boldsymbol{\xi}_j^T(\hat{\boldsymbol{\theta}}_k) \boldsymbol{\xi}_j(\hat{\boldsymbol{\theta}}_k) = \|\boldsymbol{\xi}_j(\hat{\boldsymbol{\theta}}_k)\|^2, \quad (4.41)$$

em que $\boldsymbol{\xi}_j$ é o vetor de resíduos tal que $f_j(\boldsymbol{\theta}) = f_j(\hat{\boldsymbol{\theta}}) + \boldsymbol{\xi}_j$.

²³Essa formulação ilustra o caráter mono-objetivo do método de Levenberg-Marquardt. O algoritmo se baseia na minimização de uma única função objetivo: $c(\boldsymbol{\theta})$.

De forma compacta, a função objetivo $c(\hat{\boldsymbol{\theta}}_k)$ pode ser descrita por

$$\mathbf{c}(\hat{\boldsymbol{\theta}}_k) = \frac{1}{2} \sum_{j=1}^m \left(\sum_{i=1}^n \xi_j(i, \hat{\boldsymbol{\theta}}_k) \right)^2 = \frac{1}{2} \sum_{j=1}^m \left(\|\boldsymbol{\xi}_j(\hat{\boldsymbol{\theta}}_k)\|^2 \right). \quad (4.42)$$

Quanto ao algoritmo propriamente dito, o seguinte procedimento é adotado. Primeiramente, definem-se valores iniciais para o vetor de parâmetros $\boldsymbol{\theta}$ (este é o ponto de partida do algoritmo no espaço de busca). Em seguida, para cada iteração k , o vetor de parâmetros é substituído por $\hat{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_{k-1} + \delta\boldsymbol{\theta}_k$.

Para que se possa determinar o valor de $\delta\boldsymbol{\theta}_k$, $\mathbf{s}(\hat{\boldsymbol{\theta}}_k)$ é aproximada por

$$\mathbf{s}(\hat{\boldsymbol{\theta}}_k) = \mathbf{s}(\boldsymbol{\theta}_{k-1} + \delta\boldsymbol{\theta}_k) \approx \mathbf{s}(\boldsymbol{\theta}_{k-1}) + J_s \delta\boldsymbol{\theta}_k, \quad (4.43)$$

em que J_s é a matriz Jacobiana de $\mathbf{s}(\boldsymbol{\theta})$ (A.19).

Em um ponto de mínimo da função custo, tem-se $\vec{\nabla}c = 0$. A partir de (4.43), tem-se então

$$(J_s^T J_s) \delta\boldsymbol{\theta}_k = -J_s^T \mathbf{s}(\boldsymbol{\theta}_{k-1}). \quad (4.44)$$

Na equação (4.44), $\delta\boldsymbol{\theta}$ pode ser obtido invertendo-se $J_s^T J_s$. Contudo, o algoritmo de Levenberg-Marquardt insere o *fator de amortecimento* ou *regularização* $\lambda > 0$ de forma que (4.44) se torne

$$(J_s^T J_s + \lambda_k I) \delta\boldsymbol{\theta}_k = -J_s^T \mathbf{s}(\boldsymbol{\theta}_{k-1}) \quad (4.45)$$

O valor de $\lambda_k \in \mathbb{R}$ é ajustado a cada iteração e dele depende o comportamento apresentado pelo algoritmo. De forma simplificada, se a redução da função objetivo $c(\boldsymbol{\theta})$ a cada iteração for relativamente acentuada, então a atribuição de valores pequenos para λ_k garante seu comportamento similar ao de um algoritmo de *Gauss-Newton*. Caso contrário, o fator de amortecimento deve ser aumentado e seu comportamento se assemelha ao de um algoritmo do *Gradiente*.

A convergência do algoritmo de Levenberg-Marquardt também depende da atribuição de valores adequados para o fator de amortecimento (Yamashita and Fukushima, 2000). Tipicamente, o seguinte procedimento é adotado (Marquardt, 1963):

1. Define-se²⁴ λ_0 arbitrário e $\nu > 1$;

²⁴A definição de λ_0 e quão maior que 1 deve ser o valor atribuído a ν dependem do problema a ser resolvido e do conhecimento prévio acerca dele.

2. A cada iteração $k > 0$, executa-se uma iteração do algoritmo com $\lambda_k = \lambda_{k-1}$ e outra, a partir do mesmo ponto inicial $\boldsymbol{\theta}_{k-1}$, com $\lambda_k = \lambda_{k-1}/\nu$. Com isso, os seguintes procedimentos devem ser executados:

- Se ambas as avaliações da função custo forem piores que o valor inicial $c(\boldsymbol{\theta}_{k-1})$, então λ_k é incrementado por n sucessivas multiplicações por ν até que se obtenha um valor de $c(\boldsymbol{\theta}_k)$ melhor que o correspondente à iteração anterior $c(\boldsymbol{\theta}_{k-1})$. O novo fator de amortecimento é então $\lambda_k = \lambda_{k-1}\nu^n$;
- Se $\lambda_k = \lambda_{k-1}/\nu$ implicar uma redução no valor da função custo, então esse é o valor atribuído a λ_k ;
- Se $\lambda_k = \lambda_{k-1}$ implicar uma redução no valor da função custo, então mantém-se esse valor para o fator de amortecimento.

4.7 Região de Confiança

Algoritmos baseados em *Região de Confiança* vêm-se constituindo em uma alternativa bastante comum dentre os métodos de otimização. Isso se deve basicamente a propriedades que garantem sua convergência na solução dos mais diversos tipos de problemas (Shi and Shen, 2006).

Um algoritmo de *Região de Confiança* se caracteriza por utilizar informações a respeito da função objetivo $c(\boldsymbol{\theta})$ na construção de uma função quadrática $\phi(\delta\boldsymbol{\theta})$ cujo comportamento se aproxime daquele da função objetivo na vizinhança de um dado ponto $\boldsymbol{\theta}_0$ (em que $\delta\boldsymbol{\theta} \triangleq \boldsymbol{\theta} - \boldsymbol{\theta}_0$). Essa vizinhança é definida por uma hiper-esfera de raio Δ com centro em $\boldsymbol{\theta}_0$ e esse é o aspecto que justifica o nome do método.

Com isso, a otimização da função objetivo é feita com a otimização de uma função aproximada (menos complexa) $\phi(\delta\boldsymbol{\theta})$ em regiões de confiança Γ_k a cada iteração k . De forma simplificada, cada iteração tem por objetivo solucionar o problema

$$\begin{aligned} \min_{\delta\boldsymbol{\theta} \in \mathbb{R}^n} / \max_{\delta\boldsymbol{\theta} \in \mathbb{R}^n} \quad & \phi(\delta\boldsymbol{\theta}) \\ \text{sujeito a} \quad & \|\delta\boldsymbol{\theta}\| \leq \Delta_k \end{aligned} \quad (4.46)$$

O problema de otimização (4.46) ilustra a importância do valor do raio da região de confiança a cada iteração. Caso seja muito grande, a aproximação de $c(\boldsymbol{\theta})$ pela função quadrática pode não ser boa em Γ_k . Isso implica que o mínimo de $\phi(\delta\boldsymbol{\theta})$ será muito diferente daquele da função objetivo na região

pesquisada. Por outro lado, pequenos valores de Δ podem levar o algoritmo a convergir para um mínimo local, uma vez que a solução de (4.46) levaria em conta apenas uma pequena fração do espaço de busca global.

Em detrimento à existência de algoritmos para inicialização do valor do raio da região de confiança (Sartenaer, 1996), tipicamente Δ_0 é arbitrado. A cada iteração, o valor de Δ_k é definido de acordo com a qualidade da aproximação da função objetivo por $\phi(\delta\boldsymbol{\theta})$ em Γ_k .

Primeiramente, procede-se com a solução de (4.46) e obtém-se $\delta\boldsymbol{\theta}^*$. Caso a diminuição²⁵ no valor da função quadrática $\phi(\delta\boldsymbol{\theta}^*) - \phi(\mathbf{0})$ seja comparável à diminuição real da função objetivo $c(\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}^*) - c(\boldsymbol{\theta}_k)$, o novo ponto $\boldsymbol{\theta}_k + \delta\boldsymbol{\theta}^*$ é aceito. Além disso, esse ponto passa a representar o centro da nova região de confiança Γ_{k+1} e Δ é (eventualmente) aumentado. Caso contrário, diminui-se o valor do raio e resolve-se (4.46) novamente.

4.7.1 Método de Powell (DogLeg)

Assim como no caso de Levenberg-Marquardt, o algoritmo de Powell resolve problemas de otimização irrestrita conciliando os métodos do *gradiente* e de *Gauss-Newton*. A principal diferença está no fato de o algoritmo de Powell utilizar diretamente conceitos do método de *Região de Confiança*²⁶.

Com o objetivo de resolver o sub-problema (4.46), (Powell, 1970) propôs um método que consiste na utilização de dois pontos auxiliares (além do centro de região de confiança) na busca pelo mínimo da função objetivo. O primeiro ponto se trata do denominado *Cauchy point*, que consiste no mínimo de $L(\delta\boldsymbol{\theta})$ segundo o método do *gradiente*. Esse ponto é, portanto, encontrado com a minimização da função quadrática aproximada na direção $J_\phi^T \boldsymbol{\epsilon}$ (em que $\boldsymbol{\epsilon}$ consiste no erro de estimação, $\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$) e é definido por

$$\delta\boldsymbol{\theta}_{\text{cp}} = \frac{\mathbf{g}^T \mathbf{g}}{\mathbf{g}^T J_\phi^T J_\phi \mathbf{g}} \mathbf{g}. \quad (4.47)$$

O segundo ponto consiste no ponto de Gauss-Newton $\delta\boldsymbol{\theta}_{\text{gn}}$, definido pela solução de (4.33), repetido aqui por conveniência:

$$J_\phi^T J_\phi \delta\boldsymbol{\theta}_{\text{gn}} = J_\phi^T \boldsymbol{\epsilon}.$$

²⁵Para fins de exemplificação, considera-se um problema de minimização.

²⁶O algoritmo de Levenberg-Marquardt também pode ser considerado como pertencente ao grupo dos métodos de *Região de Confiança* (Ranganathan, 2004), apesar de não utilizar tais conceitos diretamente.

A cada iteração, a utilização dos pontos $\delta\boldsymbol{\theta}_{\text{cp}}$ e $\delta\boldsymbol{\theta}_{\text{gn}}$ é feita segundo o seguinte procedimento:

1. Caso o *Cauchy point* se encontre fora da região de confiança ($\|\delta\boldsymbol{\theta}_{\text{cp}}\| \geq \Delta$), o passo do algoritmo será igual à interseção entre o segmento de reta que conecta o centro da região de confiança e $\delta\boldsymbol{\theta}_{\text{cp}}$ e o limite de Γ . Isso implica

$$\delta\boldsymbol{\theta} = \frac{\Delta}{\|\delta\boldsymbol{\theta}_{\text{cp}}\|} \delta\boldsymbol{\theta}_{\text{cp}}, \quad (4.48)$$

conforme a Figura 4.15.

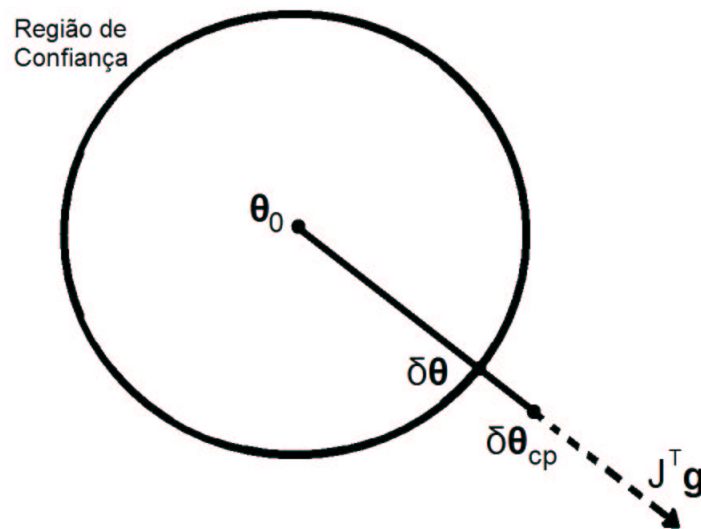


Figura 4.15: Passo do algoritmo de Powell para $\|\delta\boldsymbol{\theta}_{\text{cp}}\| \geq \Delta$.

2. Caso o *Cauchy point* e o ponto de Gauss-Newton se encontrem dentro da região de confiança ($\|\delta\boldsymbol{\theta}_{\text{cp}}\| < \Delta$ e $\|\delta\boldsymbol{\theta}_{\text{gn}}\| < \Delta$), o passo do algoritmo será igual a $\delta\boldsymbol{\theta}_{\text{gn}}$, conforme a Figura 4.16.

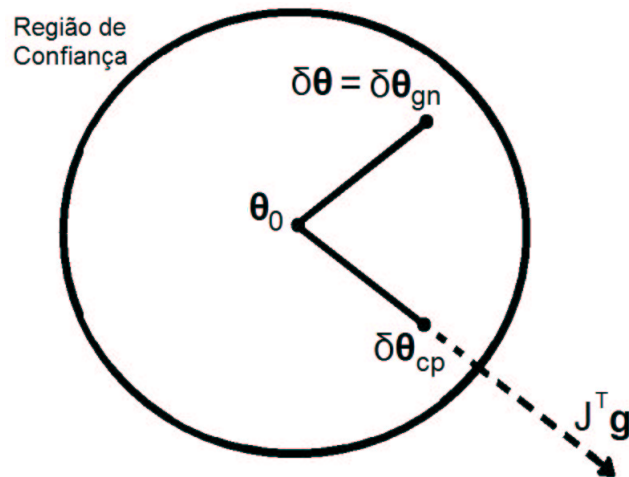


Figura 4.16: Passo do algoritmo de Powell para $\|\delta\theta_{cp}\| < \Delta$ e $\|\delta\theta_{gn}\| < \Delta$.

3. Caso o *Cauchy point* se encontre dentro da região de confiança e o ponto de Gauss-Newton encontre-se fora ($\|\delta\theta_{cp}\| < \Delta$ e $\|\delta\theta_{gn}\| \geq \Delta$), o passo do algoritmo será igual à interseção entre o segmento de reta que conecta $\delta\theta_{cp}$ e $\delta\theta_{gn}$ e o limite de Γ . Isso implica

$$\delta\theta = \delta\theta_{cp} + \beta(\delta\theta_{gn} - \delta\theta_{cp}), \quad (4.49)$$

em que β é tal que $\|\delta\theta\| = \Delta$, conforme a Figura 4.17.

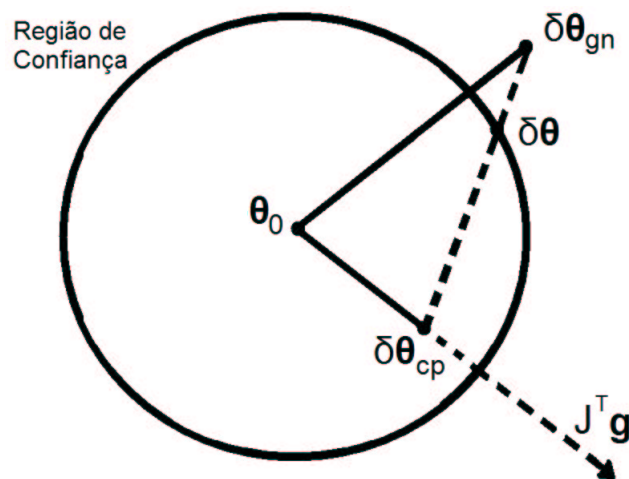


Figura 4.17: Passo do algoritmo de Powell para $\|\delta\theta_{cp}\| < \Delta$ e $\|\delta\theta_{gn}\| \geq \Delta$.

4.8 Algoritmos Genéticos

Como o próprio nome sugere, o princípio de funcionamento de um Algoritmo Genético (AG) é baseado em regras de seleção e evolução naturais. Alguns dos principais conceitos relacionados a estudos em genética são adaptados para utilização em problemas de otimização. Essa nova abordagem dá origem a algoritmos robustos e que tipicamente dependem de uma pequena quantidade de informação acerca do problema a ser resolvido (Goldberg et al., 1989).

Antes de iniciar o detalhamento a respeito dos mecanismos utilizados por um algoritmo genético, torna-se conveniente ressaltar as principais características de um algoritmo determinístico “clássico”. Esse tipo de algoritmo parte de um ponto tipicamente aleatório e, segundo alguma regra de transição pré-estabelecida e algum tipo de informação local, sugere uma direção de busca. Em seguida, procede-se com uma busca unidirecional (na direção sugerida) até que se encontre um ponto ótimo. Esse ponto torna-se então a nova solução e a resolução do problema procede com a execução do procedimento inicial de forma recursiva.

A utilização de algoritmos genéticos em detrimento aos algoritmos determinísticos clássicos é tipicamente justificada pelos seguintes limitações desse último tipo de algoritmos (Deb, 2003):

- desempenho sensível à escolha do ponto inicial;
- tendência à atração para ótimos locais (espúrios);
- ineficiência em problemas com espaços de busca discretos;
- algoritmos tipicamente especializados em classes de problemas.

O primeiro aspecto que diferencia algoritmos genéticos e determinísticos consiste no modo de representação de uma solução. Uma vez que o processo de otimização de um AG se baseia em eventos relacionados à genética, um procedimento conveniente consiste na representação das soluções em código binário. Nessa codificação, cada bit corresponde a um gene e todo o conjunto de bits da solução corresponde a um cromossomo. Assim como em genética natural, um cromossomo pode conter informações a respeito de uma ou mais “características do indivíduo”.

Por exemplo, caso se trate de um problema com duas variáveis de otimização ($\theta = [\theta_1 \ \theta_2]^T$), uma possível representação de um cromossomo s_i seria

$$s_i = \underbrace{0100}_{s_i^1} \underbrace{1010}_{s_i^2} \quad (4.50)$$

que, por se tratar de uma analogia à genética natural, é denominado *genótipo* do indivíduo θ_i . Além disso, caso os limites mínimo e máximo das variáveis de otimização fossem definidos como 0 e 31 respectivamente ($\theta_1, \theta_2 \in [0 \ 31]$), o *fenótipo* de θ_i seria $\theta_i = [4 \ 10]^T$.

No que diz respeito à representação de uma solução por um algoritmo genético, dois aspectos merecem destaque. Primeiramente, deve-se notar que não existe a necessidade de se utilizar o mesmo número de bits para a representação de cada variável de otimização. Nesse contexto, deve-se observar que, apesar de o número de bits ter forte influência sobre a qualidade da discretização do espaço de busca, pode ser escolhido de forma arbitrária ou mais conveniente.

A segunda observação é relacionada aos limites das variáveis de otimização. Evidentemente, para a representação de uma variável θ_i com ℓ_i bits, a escolha de limites mínimo e máximo tais que $\theta_i \in [0 \ 2^{\ell_i}-1]^T$ leva à discretização do espaço de busca em valores inteiros apenas. Contudo, não existe qualquer restrição associada a esses limites, o que leva a conversão ao valor real de uma variável de otimização θ_i a ser definida por

$$\theta_i = \theta_i^{\min} + \frac{\theta_i^{\max} - \theta_i^{\min}}{2^{\ell_i} - 1} v_d(s_i), \quad (4.51)$$

em que ℓ_i consiste no número de bits utilizados na representação de θ_i e $v_d(s_i)$ é o *valor decodificado*, que representa o valor de θ_i caso os limites de conversão mínimo (θ_i^{\min}) e máximo (θ_i^{\max}) fossem 0 e $2^{\ell_i}-1$, respectivamente.

Uma vez que se tenha definido o esquema de representação de cada variável de otimização, parte-se para a criação da população inicial (geração 0). Essa população é tipicamente criada de forma aleatória e é constituída por N indivíduos que representam, cada um, um ponto no espaço de variáveis.

Os indivíduos são então avaliados para cada uma das funções objetivo e, de acordo com o problema abordado, podem ser mapeados para um segundo espaço de soluções. Esse espaço tem como eixos as *funções de mérito* (ou *funções fitness*) do problema e tem a mesma dimensão do espaço de funções

objetivo²⁷. As funções de mérito tendem a ser utilizadas com o intuito de particularizar a resolução do problema ou inserir restrições mas, na maioria dos casos, são iguais às respectivas funções objetivo.

Uma vez que cada indivíduo tenha sido mapeado no espaço de soluções (*fitness* ou não), verifica-se algum tipo de condição de parada. Caso a condição seja satisfeita, todos os indivíduos são considerados soluções ótimas do problema e, em grande parte dos algoritmos, formam um conjunto de soluções não-dominadas²⁸. Caso contrário, a população é submetida aos *operadores genéticos*.

Os operadores genéticos são formados pelas operações de *reprodução*, *cruzamento* e *mutação* e são responsáveis pelo processo de busca em um algoritmo genético (Takahashi et al., 2003). De uma maneira geral, os operadores genéticos garantem a *evolução* da população em direção ao ponto ótimo (global) de maneira análoga ao que se entende por evolução de espécies biológicas. Ao fim dessa etapa, incrementa-se o contador de gerações e avalia-se a população corrente segundo as funções objetivo. Fecha-se assim o ciclo de iterações (gerações) de um algoritmo genético (Figura 4.18).

²⁷Tipicamente, utiliza-se uma função de mérito para cada função objetivo.

²⁸Um indivíduo é classificado como não-dominado sempre que não existir outro indivíduo da população que o domine segundo qualquer das funções objetivo.

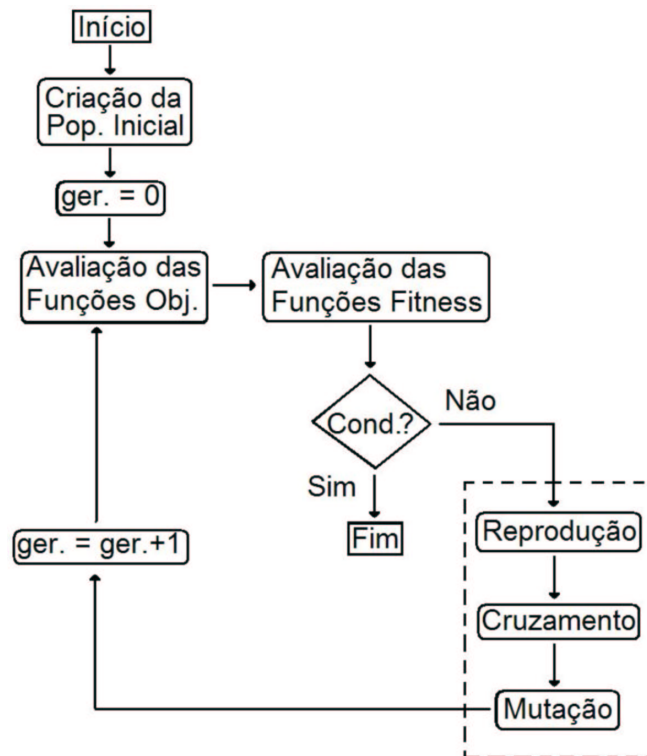


Figura 4.18: Fluxograma do princípio de funcionamento geral de um algoritmo genético

Operador de Reprodução (Seleção)

O principal objetivo do operador de reprodução é criar cópias de soluções eficientes e eliminar soluções ineficientes sem alterar o número de indivíduos da população. Nesse contexto, soluções são consideradas eficientes quando o indivíduo associado possui algum tipo de tendência a se tornar não-dominado. Devido à formulação de um algoritmo genético, essa tendência está de alguma maneira codificada em seu genótipo (4.50) e é manifestada em seu fenótipo (4.51). Dentre os tipos de operadores de reprodução mais frequentemente utilizados estão o de *seleção proporcional*, o *torneio* e o *ranqueamento*.

O operador de seleção proporcional se baseia na cópia de indivíduos eficientes de forma proporcional ao valor de alguma função de mérito (f). Primeiramente, os indivíduos são mapeados segundo a função de mérito de tal maneira que aos mais eficientes sejam atribuídos maiores valores. Em seguida, todos os valores são normalizados de tal forma que $\sum_i^N f_i = 1$ e constrói-se uma estrutura semelhante a uma roleta com perímetro unitário

(Figura 4.19).

A operação de reprodução é então realizada com a escolha de N números aleatórios relativos à posição da roleta e conseqüentemente ao indivíduo a ser copiado. Esse procedimento confere a cada indivíduo uma probabilidade de reprodução proporcional ao valor avaliado para a função de mérito.

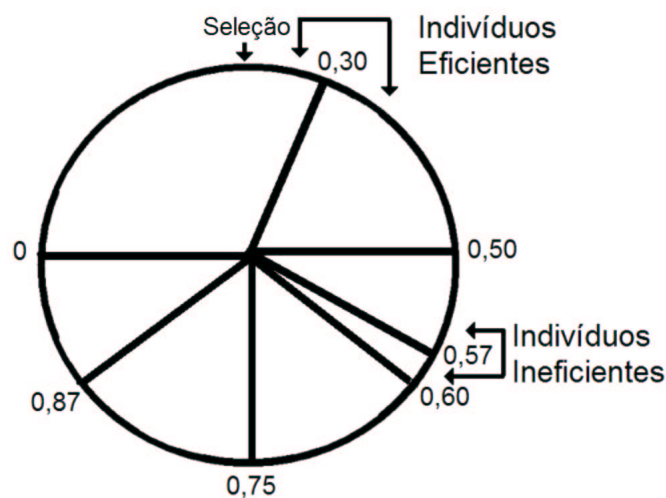


Figura 4.19: Ilustração do procedimento de reprodução via seleção proporcional.

O segundo tipo de operador de reprodução consiste no torneio. O torneio estabelece disputas entre pares de indivíduos escolhidos de forma aleatória. Ao fim de cada disputa, o indivíduo mais eficiente é copiado para uma área reservada (*mating pool*).

O procedimento completo consiste na execução desse operador duas vezes para cada indivíduo da população. Desse modo, o indivíduo mais eficiente da população produz duas cópias na área reservada e o menos eficiente desaparece da população. Além disso, esse procedimento garante a preservação do número de indivíduos da população.

A Figura 4.20 ilustra o procedimento de seleção via torneio. No exemplo, o problema de otimização consiste em determinar a aeronave de menor custo associado²⁹. Observa-se, portanto, que a aeronave de custo 90 produz duas cópias na área reservada, enquanto a de custo 370 deixa de fazer parte da

²⁹Os valores utilizados são meramente ilustrativos e não têm significado absoluto.

população a partir da geração corrente. Além disso, as aeronaves de custo intermediário produzem tipicamente uma única cópia na área reservada.

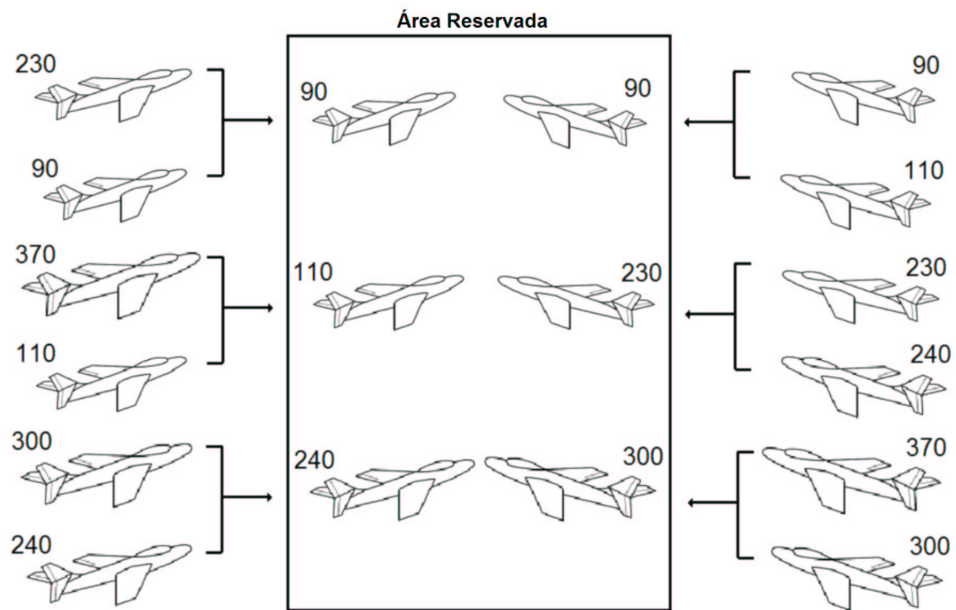


Figura 4.20: Exemplo de utilização do operador de reprodução via torneio.

Finalmente, o operador de reprodução via ranqueamento consiste em uma adaptação do operador de seleção proporcional para problemas com discrepâncias de escala. Essas discrepâncias são provenientes da existência de soluções cujas avaliações no espaço de objetivos resultem em valores consideravelmente mais eficientes que outras (Figura 4.21).

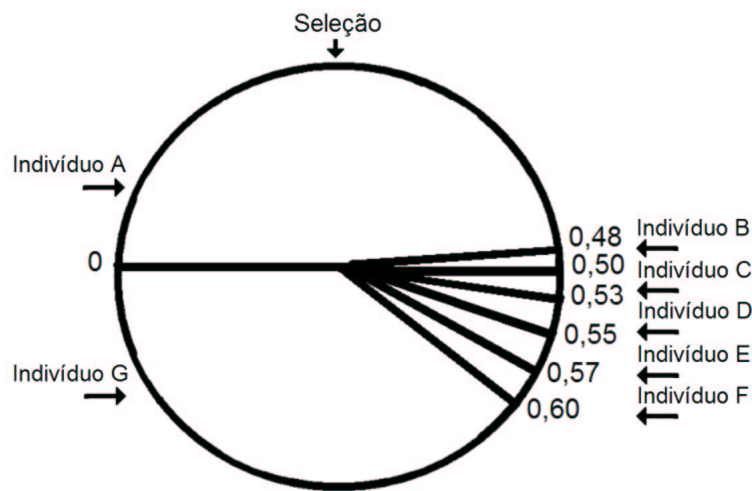


Figura 4.21: Ilustração do cenário de utilização do operador de reprodução via ranqueamento.

Em problemas desse tipo, a utilização do operador de seleção proporcional tende a preservar apenas os indivíduos mais eficientes. Esse procedimento contraria a necessidade de garantir o cumprimento das duas metas de um algoritmo de otimização multiobjetivo³⁰: *busca e diversidade*.

Devido a isso, o operador de seleção via ranqueamento atribui às soluções valores de uma função de mérito responsável pela sua classificação segundo sua eficiência. Basicamente, o procedimento consiste em ordenar as soluções segundo sua eficiência e em seguida atribuir valores de 1 a N (1 para a solução menos eficiente e N para a mais eficiente). Finalmente, procede-se com a seleção com a utilização de forma análoga ao operador de seleção proporcional (Figura 4.22).

³⁰Seção 4.2.

Operador de Mutação

Embora também tenha participação no processo de busca pelo ótimo do problema, o operador de mutação é o maior responsável pela manutenção da diversidade das soluções. Sua utilização consiste na substituição aleatória dos bits que compõem o cromossomo de um indivíduo. Tipicamente, atribuem-se à probabilidade de mutação valores inferiores a 0,01 (ou 1%). O procedimento é feito sem a necessidade de criação de pares de indivíduos (4.55).

$$\text{Indivíduo C: } 01101:1:01 \Rightarrow 01101\underline{0}01. \quad (4.55)$$

A aplicação do operador de mutação, conforme sua concepção, exige a criação de um número aleatório para cada bit de cada indivíduo. Por se tratar de um procedimento com alta complexidade computacional associada, procedimentos alternativos podem ser utilizados (Deb and Agrawal, 1999).

Operador de Elitismo

O operador de elitismo tem por função preservar os indivíduos mais eficientes (a elite) de uma população. Sua utilização pode variar sob dois aspectos principais: quanto ao modo de seleção da elite e quanto à participação da elite no processo de evolução da população.

Quanto à seleção dos indivíduos integrantes da elite, pode-se adotar um procedimento local ou global. O procedimento local consiste em, ao fim do processo de criação de novos indivíduos³³, selecionar os dois mais eficientes dentre cada conjunto de quatro indivíduos (dois antecessores e dois sucessores). Com isso, os antecessores não são necessariamente substituídos pelos seus sucessores na geração seguinte.

Por outro lado, o procedimento global consiste em, a cada geração, selecionar os N indivíduos mais eficientes dentre todos os $2N$ antecessores e sucessores. Garante-se assim a não-diminuição na eficiência dos melhores indivíduos durante o processo de evolução. Devido a isso, o procedimento global tende a acelerar o processo de busca, mas prejudica a manutenção da diversidade da população.

Quanto à participação da elite no processo de evolução, deve-se ponderar entre dois extremos. De um lado, pode-se simplesmente utilizar a elite como

³³Operações responsáveis pela criação de novos indivíduos: cruzamento e mutação.

uma espécie de repositório de indivíduos eficientes sem sua participação nas demais operações genéticas. Com isso, não se tira qualquer proveito do seu material genético durante o processo de evolução da população como um todo.

De outro lado, pode-se estabelecer que a população da nova geração (sucessores) seja constituída exclusivamente por indivíduos da elite. Esse procedimento dificulta que novos indivíduos façam parte do grupo de sucessores e o procedimento de busca não avança.

Devido a isso, costuma-se estabelecer o tamanho da população elitista como sendo igual a um pequeno percentual (10%, tipicamente) da população corrente. Permite-se então que toda a elite participe das operações genéticas em conjunto com os demais indivíduos da população sem polarizar de forma excessiva a direção do processo de busca.

4.8.1 VEGA (Vector Evaluated Genetic Algorithm)

O VEGA (*Vector Evaluated Genetic Algorithm*) consiste em uma das abordagens mais simples para resolução de problemas de otimização multiobjetivo com a utilização de algoritmos genéticos. Seu algoritmo é uma extensão direta dos algoritmos mono-objetivo e recebe esse nome por criar explicitamente um vetor de funções objetivo (Kursawe, 1991). Além disso, o método se baseia na criação de um vetor de subpopulações, o que torna sua implementação bastante particular.

Primeiramente, o algoritmo prevê a criação aleatória de tantos grupos de indivíduos quantas forem as funções objetivo do problema (m). Os N/m indivíduos de cada subpopulação são então avaliados segundo uma das funções objetivo. Em seguida, o operador de reprodução³⁴ é aplicado a cada subpopulação separadamente.

Até este ponto, o algoritmo tende a privilegiar indivíduos eficientes em funções objetivo isoladas. Devido a isso, procede-se com a aplicação do operador de cruzamento sobre toda a população. Procura-se assim criar indivíduos eficientes sob o ponto de vista do problema multiobjetivo a partir de indivíduos eficientes em sub-problemas mono-objetivo. Finalmente, o operador de mutação é aplicado em cada indivíduo.

³⁴Originalmente, utiliza-se seleção proporcional.

A Figura 4.23 ilustra a aplicação do algoritmo para uma geração.

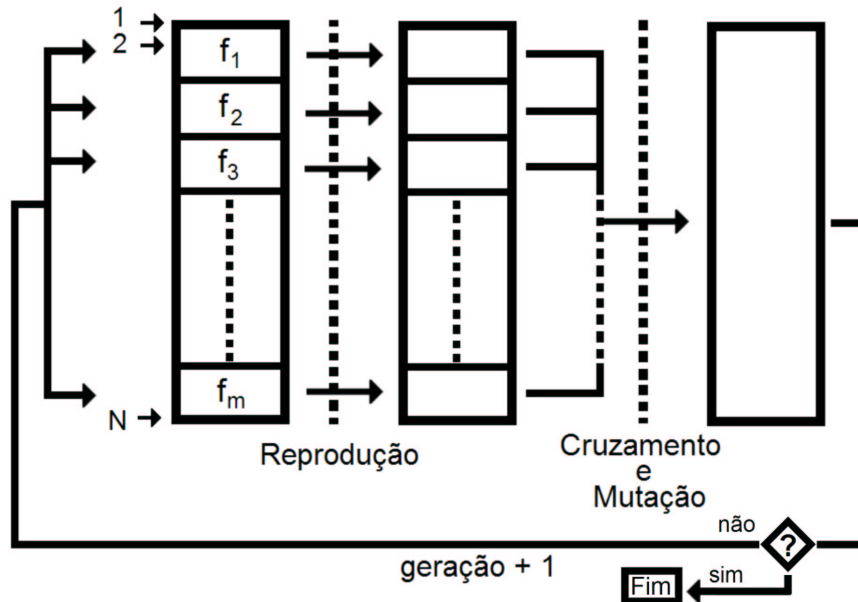


Figura 4.23: Ilustração do algoritmo VEGA.

4.8.2 WBGA (Weight-Based Genetic Algorithm)

Como o nome sugere, o WBGA (*Weight-Based Genetic Algorithm*) consiste em um algoritmo genético baseado na atribuição de pesos às funções objetivo. A classificação dos indivíduos é feita com a criação de uma função de mérito correspondente à soma entre o vetor de funções objetivo e um vetor de pesos de mesma dimensão (m).

O que diferencia o WBGA dentre os métodos baseados na atribuição de pesos às diversas funções objetivo consiste exatamente na maneira como essa etapa é executada. No caso do WBGA, atribui-se a cada indivíduo da população um vetor de pesos diferente. Com isso, o algoritmo não determina apenas uma solução não-dominada, mas tantas quanto for o número de diferentes vetores de pesos utilizados. Essa particularidade vincula a diversidade da população não-dominada à diversidade dos vetores de pesos.

Existem basicamente duas maneiras de se manter a diversidade do conjunto de vetores de pesos nesse caso. Enquanto a primeira estratégia consiste na aplicação de um critério de ranqueamento sobre os vetores de pesos, a segunda é baseada na criação de subpopulações e na atribuição de um de-

terminado vetor de pesos para cada um desses grupos de indivíduos. Pela semelhança existente entre a segunda estratégia e o algoritmo de VEGA, apenas a primeira será detalhada.

Estratégia via função de partilha

Além das n variáveis de otimização $(\theta_1, \dots, \theta_n)$, o WBGA prevê a representação de K vetores de pesos em código binário. Os m elementos de cada vetor de pesos devem ser arbitrados de forma que sua soma seja sempre igual a 1. Essa condição implica a normalização dos valores, após sua atribuição inicial, por

$$w_{N_i}^k = \frac{w_i^k}{\sum_{j=1}^m w_j^k}, \quad (4.56)$$

em que w_i^k é o i -ésimo elemento do vetor de pesos \mathbf{w}^k , na iteração k .

Tipicamente, utiliza-se uma variável inteira x^k para representar cada vetor de pesos \mathbf{w}^k . Essa variável pode assumir valores no intervalo $[1, K]$ e a escolha do valor de K determina o número de soluções não-dominadas existentes ao fim do processo de otimização. A Tabela 4.2 ilustra um exemplo de atribuição de valores aos vetores de pesos e suas respectivas variáveis inteiras para $K = 4$.

Tabela 4.2: Exemplo de atribuição de valores aos vetores de pesos e respectivas variáveis inteiras para $K = 4$

x^k	1	2	3	4
\mathbf{w}^k	$[0,2 \ 0,8]^T$	$[0,4 \ 0,6]^T$	$[0,6 \ 0,4]^T$	$[0,8 \ 0,2]^T$

Uma vez que se tenha determinado os vetores de pesos, deve-se atribuir valores de uma função de mérito para cada indivíduo θ_i da população. O valor dessa função correspondente à soma ponderada dos valores normalizados das funções objetivo:

$$r(\theta_i) = \sum_{j=1}^m w_j^i \frac{c_j(\theta_i) - c_j^{\min}}{c_j^{\max} - c_j^{\min}}, \quad (4.57)$$

em que $c_j(\theta_i)$ consiste no valor da função objetivo c_j avaliada no ponto θ_i e c_j^{\min} e c_j^{\max} correspondem aos valores mínimo e máximo dessa função.

A estratégia de utilização de uma função de partilha é então utilizada com o objetivo de garantir a diversidade da população (Krishnakumar et al., 1993). Uma vez que cada vetor de pesos \mathbf{w}^k possui uma variável inteira x^k associada, o procedimento de partilha é feito diretamente com essas variáveis. Com isso, a distância no espaço de variáveis é, neste caso, definida por

$$d_{ij} = |x^i - x^j|. \quad (4.58)$$

Além disso, define-se a *função de partilha* $Sh(d_{ij})$ por

$$Sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{\text{share}}}\right)^\alpha & , \text{ se } d \leq \sigma_{\text{share}} \\ 0 & , \text{ caso contrário} \end{cases}, \quad (4.59)$$

em que α e σ_{share} são parâmetros de ajuste. Enquanto α está relacionado com a linearidade do mapeamento (Figura 4.24), σ_{share} é responsável pela determinação dos limites de vizinhança em uma determinada fronteira³⁵.

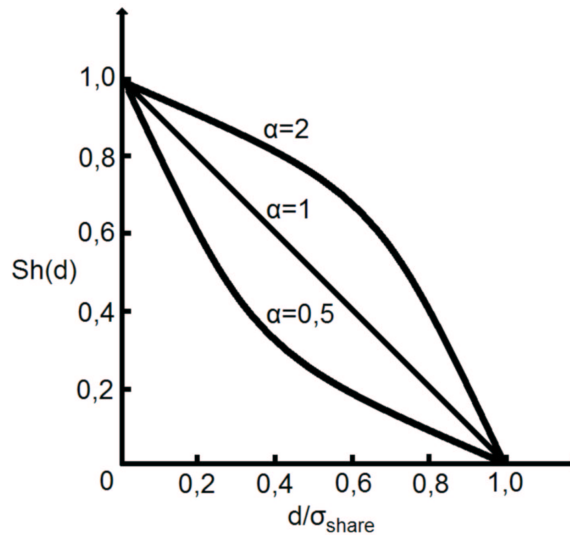


Figura 4.24: Influência de α na linearidade da função de partilha (Deb, 2003) - adaptada.

Define-se então o *contador de nicho* (nc) associado a uma solução θ_i :

$$nc(\theta_i) = \sum_{j=1}^{\mu(r(\theta_i))} Sh(d_{ij}) \quad (4.60)$$

³⁵De forma mais rigorosa, σ_{share} é um parâmetro do hiperespaço que representa uma vizinhança em uma determinada fronteira. Se a norma euclidiana for utilizada, trata-se do raio de uma hipersfera, por exemplo.

e uma nova função de mérito $r'(\boldsymbol{\theta}_i) = r(\boldsymbol{\theta}_i)/nc(\boldsymbol{\theta}_i)$ é criada. Esse é o valor utilizado para a classificação das soluções.

Importante destacar que essa degradação do valor da função de mérito tem por objetivo privilegiar soluções em regiões isoladas. A caracterização dessas regiões é feita sob o ponto de vista da sub-região definida por σ_{share} . De fato, esse parâmetro define uma vizinhança e o valor da função de mérito de cada solução é degradado de acordo com o número de soluções ali presentes.

Uma vez atribuídos os valores da função de mérito para cada indivíduo da população, procede-se com a utilização dos operadores genéticos e as gerações transcorrem segundo a Figura 4.18.

4.8.3 MOGA (Multiple Objective Genetic Algorithm)

O MOGA (*Multiple Objective Genetic Algorithm*) consiste no primeiro algoritmo de otimização genético baseado na classificação da população segundo critérios de não-dominância (Fonseca and Fleming, 1998). De fato, tanto a busca por soluções não-dominadas quanto a manutenção da diversidade da população ganham destaque nesse método. O principal aspecto que diferencia o MOGA dos demais algoritmos genéticos está na maneira com que a função de mérito é atribuída a cada indivíduo da população.

Primeiramente, cada solução é avaliada segundo sua dominância em relação às demais com a utilização de um critério de ranqueamento. Para isso, aplica-se a cada solução $\boldsymbol{\theta}_i$ uma função de mérito $r(\boldsymbol{\theta}_i)$ tal que

$$r(\boldsymbol{\theta}_i) = 1 + n_{\boldsymbol{\theta}_i}, \quad (4.61)$$

em que $n_{\boldsymbol{\theta}_i}$ consiste no número de soluções mais eficientes que $\boldsymbol{\theta}_i$. Isso implica que o valor da função de mérito associada a uma solução não-dominada $\boldsymbol{\theta}_i^*$ é $r(\boldsymbol{\theta}_i^*) = 1$.

Quanto aos possíveis valores atribuídos aos indivíduos da população, dois aspectos merecem relevância. O primeiro deles se refere ao fato de $r(\boldsymbol{\theta}_i)$ ser limitada por 1 e N (ou, de forma mais rigorosa: $r(\boldsymbol{\theta}_i) \in \{\mathbb{Z} | 1 \leq r(\boldsymbol{\theta}_i) \leq N\}$). De fato, é de se esperar que ao menos uma solução seja não-dominada e que, caso cada fronteira seja formada por apenas um indivíduo, o maior valor atribuído seja igual ao número de indivíduos da população.

O segundo aspecto está relacionado ao fato de (4.61) tipicamente não atribuir todos os possíveis valores do intervalo $[1, N]$ aos indivíduos da população. Essa situação ocorre sempre que existir alguma fronteira formada por mais de uma solução e pode causar problemas de discrepância de escala. Por exemplo, se $N - 1$ indivíduos formarem a fronteira não-dominada e apenas 1 indivíduo constitui uma eventual segunda fronteira, ter-se-á $r(\theta_1) = \dots = r(\theta_{N-1}) = 1$ e $r(\theta_N) = N$.

Devido a isso, uma nova etapa é incluída no processo de classificação dos indivíduos da população. Primeiramente, utiliza-se uma nova função de mérito tal que ao indivíduo mais eficiente seja atribuído o valor N e ao menos eficiente 1. Em seguida, atribui-se a cada indivíduo a soma dos valores de todos os indivíduos da mesma fronteira dividida pelo número de indivíduos dessa fronteira. Todo o procedimento dá origem à *função de mérito média* $r_{avg}(\theta_i)$ e pode ser implementado segundo (4.62):

$$r_{avg}(\theta_i) = N - \sum_{k=1}^{r(\theta_i)-1} \mu(k) - 0,5(\mu(r(\theta_i)) - 1), \quad (4.62)$$

em que $\mu(k)$ consiste no número de indivíduos da fronteira k .

Todo o procedimento de classificação garante a busca pela melhor solução com a atribuição de valores mais elevados aos indivíduos mais eficientes. Contudo, essa estratégia não garante diversidade entre a população não dominada³⁶. Essa meta é atingida com a criação de *nichos* nas fronteiras de soluções.

O primeiro passo para isso consiste na definição da distância (no espaço de soluções³⁷) entre dois indivíduos pertencentes a uma mesma fronteira:

$$d_{ij} = \sqrt{\sum_{k=1}^{\mu(j)} \left(\frac{c_k(\theta_i) - c_k(\theta_j)}{c_k^{\max} - c_k^{\min}} \right)^2}, \quad (4.63)$$

em que $c_k(\theta_i)$ consiste no valor da função objetivo c_k avaliada no ponto θ_i e c_k^{\min} e c_k^{\max} correspondem aos valores mínimo e máximo dessa função.

³⁶De fato, a metodologia utilizada pelos algoritmos genéticos não garante a manutenção de diversidade da população. Para isso, algum tipo de procedimento específico deve ser implementado.

³⁷Diferentemente do que ocorre no algoritmo WBGA, em que a distância é calculada no espaço de variáveis.

Define-se então uma função de partilha (4.59) com $\alpha = 1$ e, em seguida, o contador de nicho (nc) associado a uma solução θ_i é calculado segundo (4.60). Finalmente, cria-se uma nova função de mérito $r'(\theta_i)$ tal que seu valor seja igual $r_{\text{avg}}(\theta_i)/nc(\theta_i)$ e esse passa a ser o valor da solução θ_i para fins de classificação.

Embora soluções de uma mesma fronteira possuam mesmo r_{avg} , os respectivos valores de função de partilha (e portanto de contador de nicho) são diferentes. Com isso, garantem-se maiores valores de função de mérito para soluções em regiões menos povoadas de uma determinada fronteira, o que força a manutenção da diversidade da população.

Assim como qualquer algoritmo de otimização genético, o MOGA prevê a utilização dos operadores genéticos após a atribuição de funções de mérito aos indivíduos. Procede-se então com a execução das gerações conforme a Figura 4.18.

4.8.4 NSGA II

Também para o NSGA II (*Non-dominated Sorting Genetic Algorithm II*), as duas metas de um algoritmo de otimização multiobjetivo são garantidas com a utilização de funções de mérito e funções de partilha. A diferença se faz basicamente no aprimoramento dessa estratégia e na utilização do operador de Elitismo como mais uma ferramenta para um aumento de eficiência do processo de busca com a manutenção da diversidade da população (de Vasconcelos et al., 2002). Esse é o aspecto que diferencia de forma predominante a implementação do NSGA II em relação à versão não-elitista NSGA (Dias and de Vasconcelos, 2002).

A maneira com que o operador de elitismo é utilizado no NSGA II particulariza sua implementação. Ao início de cada geração, tem-se uma população P' composta por N elementos. O primeiro passo consiste na geração da elite Q com a utilização dos operadores de reprodução, cruzamento e mutação³⁸ (Figura 4.25). Em seguida, procede-se com a criação de uma população P , composta por $2N$ indivíduos, correspondente à união entre P' e Q .

O próximo passo do algoritmo consiste na ordenação da população segundo critérios de dominância. Essa etapa consiste na criação de fronteiras

³⁸Em especial, utiliza-se um operador de reprodução via torneio (*crowded tournament selection operator*) cujos detalhes são omitidos por conveniência.

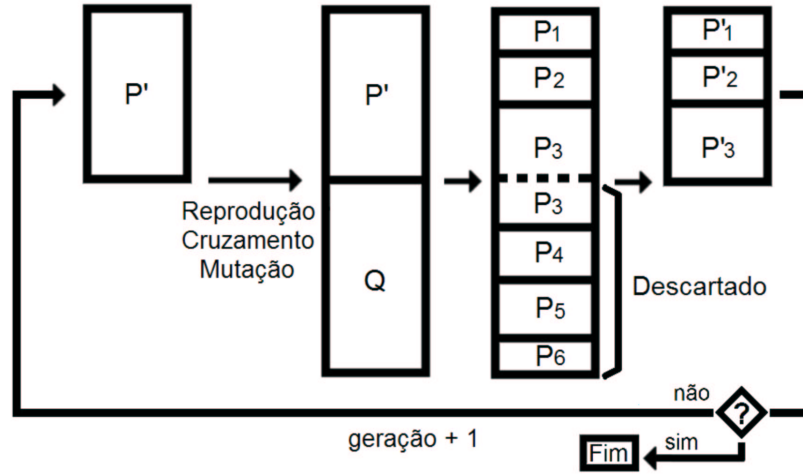


Figura 4.25: Operador de Elitismo para o NSGA II

P_j tais que

$$\begin{cases} P = \bigcup_{j=1}^{\rho} P_j \\ P_1 \triangleleft P_2 \triangleleft \dots \triangleleft P_{\rho}, \end{cases} \quad (4.64)$$

em que ρ é o número de fronteiras existentes, dados o problema proposto e a população corrente.

Uma vez que a população tenha sido classificada, tem-se que P_1 contém as soluções mais eficientes (não-dominadas), P_2 possui soluções dominadas apenas pelas integrantes de P_1 e assim sucessivamente. Com isso, torna-se natural atribuir maiores valores de funções de mérito aos indivíduos das fronteiras mais próximas da Pareto-ótimo (P_1, P_2, \dots) e menores valores aos integrantes das fronteiras mais distantes ($\dots, P_{\rho-1}, P_{\rho}$). A maneira como esse procedimento é feito caracteriza o algoritmo NSGA II.

A atribuição de valores de funções de mérito a toda a população se inicia com a fronteira não dominada (P_1). Nesse caso, atribui-se o valor $2N$ (igual ao número de indivíduos da população) a todos os indivíduos dessa fronteira:

$$r(\theta_i) = 2N \quad \forall \theta_i \in P_1 \quad (4.65)$$

Embora esse procedimento force o processo de busca em direção à fronteira Pareto-ótimo, não existe qualquer garantia quanto à manutenção da diversidade da população. Para isso, utiliza-se a estratégia de atribuição de valores de uma função de partilha de modo a privilegiar (na operação de reprodução seguinte) as soluções localizadas em regiões menos povoadas das

fronteiras.

A utilização da função é então iniciada com a definição da distância entre soluções no espaço de variáveis:

$$d_{ij} = \sqrt{\sum_{k=1}^{|P_l|} \left(\frac{\theta_k^i - \theta_k^j}{\theta_k^{\max} - \theta_k^{\min}} \right)^2} \quad (4.66)$$

em que $|P_l|$ é o número de elementos de cada vetor θ^i correspondente a um indivíduo da fronteira P_l , θ_k^i é o k -ésimo elemento do vetor θ^i e θ_k^{\min} e θ_k^{\max} correspondem aos valores mínimo e máximo que esse elemento pode assumir³⁹.

O próximo passo consiste na definição de uma função de partilha (4.59) com $\alpha = 2$. Essa formulação implica que qualquer solução θ_j a uma distância maior que σ_{share} de outra solução θ_i não tem qualquer participação em $Sh(\theta_i)$.

Em seguida, o contador de nicho (4.60) associado a cada solução θ_i da fronteira⁴⁰ é utilizado. Além disso, cria-se uma nova função de mérito $r'(\theta_i) = r(\theta_i)/nc(\theta_i)$ e esse passa a ser o valor da solução θ_i para fins de classificação.

Com isso, finaliza-se o processo de atribuição de valores de função de mérito às soluções da fronteira P_1 . O mesmo procedimento deve ser feito com as demais fronteiras a partir de P_2 . Para isso, aos indivíduos de cada nova fronteira, devem-se atribuir valores inferiores aos alcançados ao fim do processo de degradação da fronteira anterior.

Finalmente, a população P deve ser reduzida pela metade a fim de constituir uma população de antecessores P' responsável pela criação da nova elite. Essa transformação é implementada simplesmente com o descarte dos N indivíduos menos eficientes. Ao final da transformação, passa-se para a próxima geração e o algoritmo prossegue.

³⁹Nota sobre nomenclatura: Em (4.66), um indivíduo é representado por θ^i . Esse padrão difere do utilizado comumente (θ_i) com o intuito de destacar a similaridade existente entre essa equação e (4.63).

⁴⁰Neste caso, ainda se trata da primeira fronteira (não-dominada) P_1 .

4.8.5 SPEA (Strength Pareto Evolutionary Algorithm)

O SPEA (*Strength Pareto Evolutionary Algorithm*) consiste em um algoritmo genético de otimização multiobjetivo caracterizado pela maneira com que trata as soluções pertencentes e não-pertencentes à elite. Sua implementação se baseia na manutenção de uma população elitista de tamanho limitado cujos indivíduos representam as melhores soluções encontradas até a geração corrente. Essas soluções também são utilizadas no processo de evolução da população com sua participação nas operações genéticas.

O algoritmo é iniciado com a criação de uma população P com N indivíduos e a alocação da população elitista \bar{P} inicialmente vazia. Os indivíduos de \bar{P} são limitados em número por \bar{N} e têm por função representar as soluções mais eficientes até a geração corrente. A cada geração, toda a fronteira não-dominada da população é acrescentada à elite.

No SPEA, a classificação dos indivíduos pertencentes à elite é feita com a utilização de uma função de mérito específica⁴¹ $s(\boldsymbol{\theta}_i)$. Essa função é proporcional ao número ($n_{\boldsymbol{\theta}_i}$) de indivíduos da população não-elitista que o indivíduo (pertencente à elite) domina e é definida por

$$s(\boldsymbol{\theta}_i) = \frac{n_{\boldsymbol{\theta}_i}}{N + 1}, \quad (4.67)$$

em que o termo $N + 1$ é utilizado com o intuito de garantir $s(\boldsymbol{\theta}_i) \in [0, 1]$.

Por outro lado, atribui-se a cada indivíduo da população não-elitista o valor de uma função de mérito igual ao somatório dos valores atribuídos em (4.67) a todas as soluções que o dominam:

$$r(\boldsymbol{\theta}_j) = 1 + \sum_{\boldsymbol{\theta}_i \in \bar{P} \mid \boldsymbol{\theta}_i \prec \boldsymbol{\theta}_j} s(\boldsymbol{\theta}_i), \quad (4.68)$$

em que soma-se 1 ao somatório com o intuito de garantir que os indivíduos da elite possuam menores valores associados que aqueles correspondentes à população não-elitista.

A equação (4.67) implica que maiores valores serão atribuídos aos indivíduos elitistas que dominam um maior número de indivíduos da população

⁴¹A letra utilizada para a representação da função faz referência ao termo *strength*. Segundo (Zitzler and Thiele, 1998), esse termo representa a “quantificação da utilidade de uma regra”.

não-elitista. Além disso, o método identifica como indivíduos mais eficientes aqueles aos quais são atribuídos os menores valores.

Em princípio, esse procedimento pode parecer contraditório: Tipicamente, classificam-se como mais eficientes as soluções que dominam um maior número de indivíduos. Contudo, é importante observar que o SPEA se baseia no fato de que a existência de regiões intensamente povoadas pode ser detectada com a identificação de indivíduos (elitistas) com grandes valores de n_{θ_i} associados. Além disso, (4.67) e (4.68) garantem a penalização tanto de indivíduos da população não-elitista dominados simultaneamente por vários indivíduos da elite (busca) quanto daqueles dominados por uma mesma solução (diversidade).

Essa é a estratégia utilizada pelo algoritmo para atingir as duas metas da otimização multiobjetivo. A Figura 4.26 ilustra esse procedimento: De forma simplificada, privilegiam-se os indivíduos da população não-elitista localizados em regiões mais claras e menos povoadas e indivíduos da elite responsáveis pela criação de regiões menos densamente povoadas. Além disso, aos indivíduos da elite são atribuídos valores sempre menores que os correspondentes à população não-elitista.

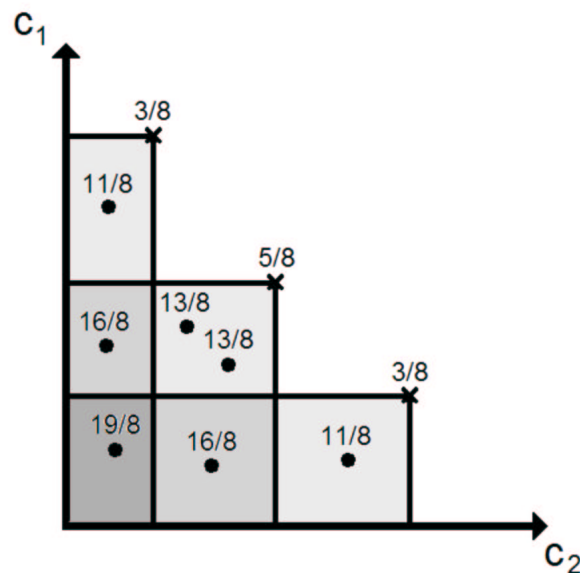


Figura 4.26: Exemplo de utilização das funções de partilha no SPEA para um problema do tipo máximo-máximo (Zitzler and Thiele, 1998) - adaptada

Além do que foi mencionado, o SPEA ainda prevê a utilização de um procedimento específico para manutenção da diversidade da população. Ao contrário do que é tipicamente utilizado por algoritmos genéticos, essa meta não é atingida com a definição de uma vizinhança a partir de uma métrica⁴². No SPEA, utiliza-se uma técnica de agrupamento (*clustering*).

O algoritmo de agrupamento utilizado é responsável não apenas pela manutenção da diversidade da população não-dominada como pela redução do número de indivíduos nessa população. Primeiramente, considera-se cada um dos \overline{N} indivíduos de \overline{P} com sendo um grupo. Em seguida as distâncias entre todos os $\binom{\overline{N}}{2}$ possíveis pares de indivíduos são calculadas.

Com isso, selecionam-se os dois indivíduos mais próximos entre si (menor distância associada) e cria-se um novo grupo que os contenha. O procedimento prossegue com o cálculo das distâncias entre todos os grupos existentes (4.69) e com a criação de novos grupos de forma recursiva até que se tenham \overline{N} grupos em \overline{P} . Com isso, a métrica utilizada para a criação desses grupos consiste em:

$$d_{C_1 C_2} = \frac{1}{|C_1||C_2|} \sum_{i \in C_1, j \in C_2} d_{ij}, \quad (4.69)$$

em que C_1 e C_2 representam dois grupos quaisquer e as distâncias podem ser definidas tanto no espaço de variáveis quanto no espaço de objetivos.

Com o fim do processo, os grupos com mais de um indivíduo são substituídos pelo indivíduo mais próximo do seu centro geomérico ou por um indivíduo criado exatamente nesse ponto.

Finalmente, a população não-elitista e a elite são unidas e os operadores de reprodução, cruzamento e mutação são aplicados. A evolução das gerações segue normalmente até que se atinja alguma condição de parada.

4.9 Simulated Annealing (Metropolis)

O método de *simulated annealing* (SA)⁴³ consiste em um método de busca local aprimorado no sentido de possibilitar a determinação do ótimo global do problema abordado. Esse aprimoramento é caracterizado pela aceitação

⁴²Como o σ_{share} , por exemplo.

⁴³Apesar de haver traduções para o português, o nome original será utilizado por sua ampla utilização mesmo em textos de língua não-inglesa.

(eventual) de movimentos de degradação do processo de busca com o intuito de evitar a atração por ótimos locais. O algoritmo se inspira no processo físico de obtenção de estruturas cristalinas pelo super-aquecimento e conseqüente resfriamento lento da substância, o que induz ao estado menos energético possível do sistema (Kirkpatrick et al., 1983).

O algoritmo estocástico de otimização implementado pelo SA é baseado em duas estratégias principais. A primeira delas consiste na utilização do *algoritmo de Metropolis* (Metropolis et al., 1953). Esse algoritmo permite que o processo de busca seja degradado em algumas iterações com o intuito de evitar a atração por pontos de ótimo local. Sua implementação é baseada nos seguintes passos:

1. a partir de um estado θ_i de energia $E(\theta_i)$, gera-se um estado θ_j de energia $E(\theta_j)$ com a inserção de algum tipo de perturbação no sistema;
2. se a variação de energia $\Delta E = E(\theta_j) - E(\theta_i)$ for negativa ou nula, então θ_j é aceito como estado atual do sistema;
3. se a variação de energia for positiva, então a probabilidade de θ_j ser aceito como estado atual do sistema é definida por:

$$p = e^{-\left(\frac{E(\theta_j) - E(\theta_i)}{k_B T}\right)}, \quad (4.70)$$

em que T é a temperatura do sistema e k_B é a *constante de Boltzmann*.

Para o processo físico, a obtenção de uma estrutura cristalina de baixa energia depende da diminuição gradativa (e principalmente lenta) da temperatura do sistema. Esse procedimento é realizado com o intuito de garantir que o sistema atinja equilíbrio térmico em cada temperatura. No caso do algoritmo de Metropolis, o procedimento é implementado com a execução de diversas transições de estado para um mesmo valor do parâmetro T .

Essa metodologia utilizada para diminuição do valor de T consiste na segunda estratégia utilizada pelo SA. De forma prática, essa diminuição é feita assim que se observe que as transições de estado já não mais implicam grandes diminuições no valor da função objetivo (energia do sistema). Com isso, e segundo (4.70), a cada novo valor de T diminui-se a probabilidade de aceitação de um estado de maior energia que o corrente.

Assim como qualquer algoritmo iterativo, a recursão do SA depende da avaliação de alguma condição de parada. Essa condição consiste tipicamente

em um número máximo de iterações. A Figura 4.27 ilustra o algoritmo do *simulated annealing*.

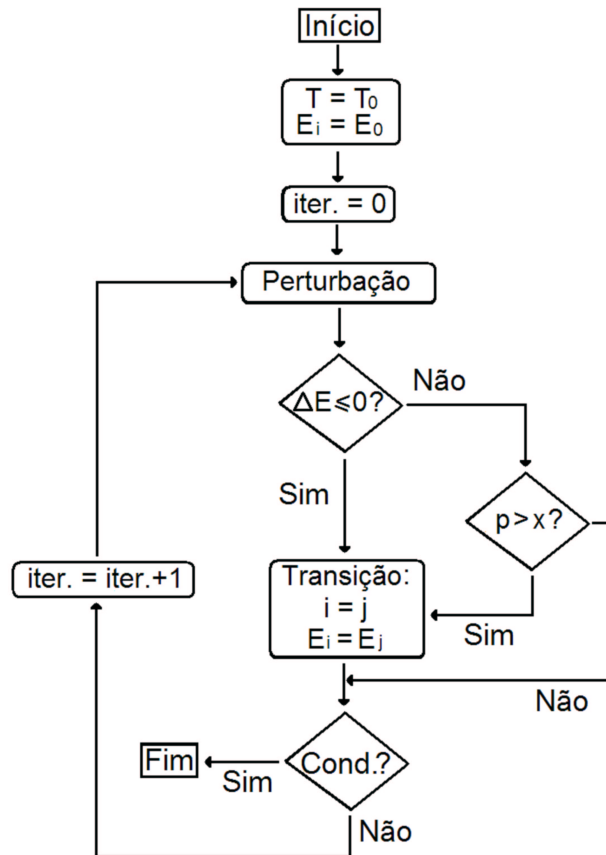


Figura 4.27: Fluxograma do princípio de funcionamento geral do algoritmo de *Simulated Annealing*. x consiste em uma variável aleatória com distribuição normal tal que $x \in [0, 1]$.

De maneira geral, o SA apresenta uma série de particularidades no que diz respeito à sua implementação. Dentre elas estão a determinação do valor inicial da temperatura T_0 , o número de transições de estado (ou o valor médio de ΔE) a partir do qual deve-se diminuir o valor de T e mesmo a estratégia de redução desse parâmetro. Quanto ao último aspecto, utiliza-se tipicamente um fator de redução (entre 0,8 e 0,99) cujo valor exato é obtido de forma experimental.

A APLICAÇÃO DESENVOLVIDA

Todo o desenvolvimento responsável pela geração dos resultados apresentados neste trabalho dá origem a uma aplicação única. Essa aplicação tem por objetivo não apenas reunir todas as rotinas desenvolvidas como também oferecer um ambiente flexível e amigável para investigações a respeito de estimação de parâmetros e reconstrução de trajetória de sistemas dinâmicos¹. Trata-se de uma aplicação desenvolvida em ambiente Matlab/Simulink® estruturada de forma a garantir a funcionalidade de todos os métodos selecionados sem que haja a necessidade de maiores interferências do usuário final na aplicação em si.

Este capítulo tem por objetivo detalhar o modo de funcionamento e as funcionalidades da aplicação desenvolvida de acordo com sua concepção. Na Seção 5.1, apresentam-se as duas formas adotadas para estruturação funcional da aplicação: em camadas de rotinas relacionadas e em diretórios de classes de rotinas. Na seção 5.2, apresentam-se funções auxiliares, responsáveis por atividades periféricas na aplicação. Nas Seções 5.3 e 5.4, apresentam-se descrições das funções de todos os algoritmos determinísticos e estocásticos implementados, respectivamente. Na Seção 5.5 detalham-se os modelos utilizados para a resolução do problema de reconstrução de trajetória de aeronaves. Finalmente, a Seção 5.6 ilustra as funcionalidades da interface gráfica responsável pela troca de informação com o usuário.

¹Importante observar a generalização dos cenários em que a aplicação pode ser utilizada. Justificativas para essa afirmação são fornecidas nas Seções subseqüentes.

5.1 Estruturação da Aplicação

O desenvolvimento de aplicações em que várias entidades devem ter acesso a um conjunto de funcionalidades compartilhadas apresenta aspectos bastante particulares. O principal deles consiste na necessidade de se garantir que o acesso a determinadas funcionalidades da aplicação seja restrito a determinados conjuntos de entidades (Fagundes, 2004). Com isso, garante-se a organização da aplicação de forma funcional, de tal maneira a possibilitar a disponibilização de informação entre as entidades de forma segura e padronizada.

Com o intuito de viabilizar a comunicação entre as diversas entidades que compõem a aplicação, projetou-se uma arquitetura composta por três camadas principais. Essa arquitetura é a responsável pela organização da aplicação segundo o fluxo de informação entre as rotinas e é ilustrada na Figura 5.1.

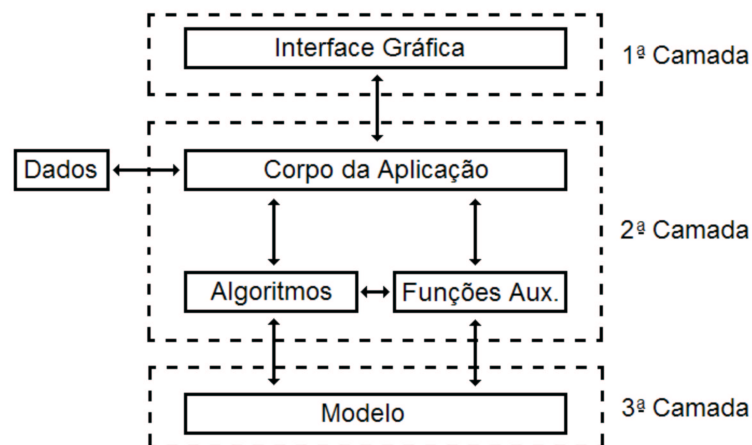


Figura 5.1: Arquitetura e fluxo de informação entre as entidades da aplicação.

A arquitetura ilustrada na Figura 5.1 garante uma série de vantagens no que diz respeito à disponibilização das funcionalidades da aplicação:

- 1ª Camada:
 - Toda operação de troca de informação com o usuário é feita por esta camada. Garante-se, portanto, facilidade no que diz respeito à inserção e recepção de informação por parte do usuário. Além disso, garante-se um único tipo de acesso à 2ª Camada: via 1ª Camada;

- 2^a Camada:
 - Esta é a camada de aquisição e processamento de dados da aplicação. Seu objetivo é intermediar o acesso aos dados de entrada e ao modelo utilizados. Esta camada é também, portanto, responsável pela criação dos resultados finais a serem exibidos pela interface gráfica;
 - A coordenação das atividades desempenhadas pela 2^a Camada é efetuada por uma rotina principal, indicada por “Corpo da Aplicação”. Essa rotina é responsável pela transformação dos comandos do usuário em execuções das funções de algoritmos e das funções auxiliares. Enquanto as funções de algoritmos implementam diretamente os métodos selecionados, as funções auxiliares são responsáveis por atividades periféricas, tais como a verificação de erros gerais e da integridade dos dados de entrada;
 - O acesso à 3^a Camada pode ser feito de duas maneiras: via funções de algoritmos ou via funções auxiliares;

- 3^a Camada:
 - Esta camada contém o modelo cujos parâmetros devem ser estimados. Por se tratar da camada mais inferior da arquitetura, consiste também na mais freqüentemente executada. Devido a isso, deve ser projetada de modo a possuir acesso facilitado pela 2^a e ser capaz de ser executada de forma mais eficiente possível. Caso contrário todo o processo de estimação / reconstrução se torna ineficiente.

Além da criação de uma arquitetura em camadas para a aplicação, criou-se também uma estruturação em diretórios responsável pelo agrupamento de rotinas relacionadas (Figura 5.2). Com isso, podem-se separar funções responsáveis pelo tratamento dos dados de entrada daquelas responsáveis pela execução dos algoritmos propriamente ditos, por exemplo. Além do caráter organizacional, essa medida aumenta ainda mais a capacidade de expansão da ferramenta, uma vez que modulariza a aplicação em um nível bastante fundamental.

A estruturação da aplicação conforme descrito cria um ambiente de desenvolvimento para investigações a respeito do desempenho de algoritmos em cenários bastante diversos. Podem-se, por exemplo, utilizar manobras (representadas pelos sinais de entrada e saída) ou mesmo algoritmos e modelos diferentes dos contemplados por este trabalho. De fato, a aplicação desenvolvida viabiliza a possibilidade de se resolverem problemas não relacionados com reconstrução de trajetória ou até mesmo com dinâmica de aeronaves.

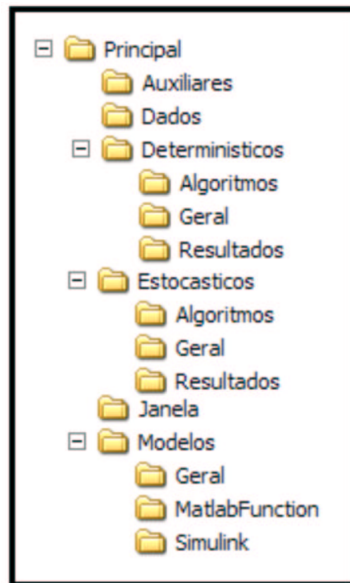


Figura 5.2: Estruturação da aplicação em diretórios.

Nas Seções 5.2 a 5.5, uma breve descrição a respeito de cada uma das rotinas implementadas é dada. Por conveniência, não se pretende entrar em maiores detalhes a respeito da sua implementação. Com isso, disponibilizam-se apenas a interface e o texto relativo ao espaço de ajuda (*help*) de cada um dos arquivos do Matlab (*m-files*). Essas Seções organizam as rotinas simultaneamente segundo os critérios de estruturação descritos nas Figuras 5.1 e 5.2. Enquanto a primeira classificação é feita pela própria divisão em Seções, a segunda é alcançada com a determinação da localização de cada uma das rotinas nos diretórios selecionados.

5.2 Funções auxiliares

As funções auxiliares têm por objetivos principais o tratamento de erros gerais e a verificação de integridade (e tratamento) dos sinais de entrada. Além disso, também cumprem o importante papel de intermediar e possibilitar a comunicação entre as três camadas em que a aplicação foi dividida. A descrição dessas rotinas é ilustrada a seguir.

pircaminhos.m

```
function pircaminhos(flagir)
Inserir/remover caminhos para os diretórios que contêm os algoritmos
(Windows, Unix-like)
```

Entrada:

```
flagir = 0 => insere; flagir = 1 => remove
```

Auxiliares\plogeventos.m

```
function plogeventos(evento,flagts)
Imprime log de eventos
```

Entrada:

```
evento = evento a ser impresso (pode ser numerico ou string)
flagts = 0 => não imprime timestamp (default)
       = 1 => imprime timestamp
```

Auxiliares\pleituracsv.m

```
function [tmedido,xmedido,Ts] = pleituracsv(nomearqcsv,viniend,retmed)
Le dados de arquivo csv
```

Entrada:

```
nomearqcsv = nome do arquivo a ser lido
viniend = vetor de pontos inicial e final a serem considerados das series
         lidas do arquivo csv
retmed = flag de indicação de correção (retirada) de medicações
        inconsistentes (veja pretmedicoes)*
        = 0 => não retira medicações inconsistentes
        = 1 => retira medicações inconsistentes
```

Saída:

```
tmedido = vetor de tempo (1ª coluna do arquivo)
xmedido = matriz de variáveis medidas (cada coluna corresponde
         a uma variável)
Ts = período de amostragem
```

*Obs: Opcional (default = 0)

Auxiliares\pdist.m

```
function s = pdist(v1,v2)
Calcula distância Euclidiana entre os pontos p1 e p2.
```

Entrada:

```
v1 = vetor coluna (n posições) que representa o 1º ponto no espaço de n dimensões
v2 = vetor coluna (n posições) que representa o 2º ponto no espaço de n dimensões
```

Saída:

```
s = distância entre v1 e v2
```

Obs: Se v1 e v2 forem matrizes nxm, então calcula-se a distância entre cada um dos pares de pontos v1(:,i) e v2(:,i). Com isso, s se torna um vetor (m linhas) de distâncias.

Auxiliares\pretmedicoes.m

```
function [tfim,Dfim] = pretmedicoes(tini,Dini)
Retira medicoes repetidas (tini(i) == tini(i+1)) ou
inconsistentes (tini(i) > tini(i+1))

Entrada:
  tini = vetor de tempo inicial (vetor coluna)
  Dini = matriz de dados inicial (cada coluna deve ser uma variavel)

Saida:
  tfim = vetor de tempo final (vetor coluna)
  Dfim = matriz de dados final (cada coluna eh uma variavel)
```

Auxiliares\prmse.m

```
function ind = prmse(E1,E2,p)
Retorna RMSE para quantificar semelhanca entre atratores

Entrada:
  Ex = estado do atrator x
      (Matriz em que cada coluna consiste em uma variavel de estado)
  p = percentual da trajetoria descartada para que sejam consideradas
      apenas orbitas em regime estacionario (opcional, default = 0)

Saida
  ind = indice
```

5.3 Algoritmos Determinísticos

As rotinas relacionadas à implementação dos algoritmos de otimização determinísticos podem ser divididas em dois grupos principais: funções gerais e funções de algoritmos. Enquanto as funções gerais consistem em rotinas de apoio, principalmente relacionadas à construção de estruturas de dados específicas, as funções de algoritmos se tratam da implementação dos métodos de otimização propriamente dita. A seguir apresentam-se, primeiramente, as funções gerais.

Deterministicos\Geral\pvec2mat.m

```
function M = pvec2mat(v,ncol)
Cria ncol colunas em M com os valores de v
```

Deterministicos\Geral\pmat2vec.m

```
function v = pmat2vec(M)
Concatena colunas de M verticalmente (cria vetor coluna)
```

Deterministicos\Geral\pgradvec.m

```
function g = pgradvec(delta,cellerro)
Retorna o gradiente da funcao s(p) a partir de perturbacoes

Entrada:
delta = valor infinitesimal (sugestao: 1e-6) tal que o gradiente de s(p)
      com relacao aa variavel i (g_i) possa ser aproximado por:
      g_i = (s(p + delta*e_i) - s(p))/delta
      em que e_i eh um vetor coluna com zero em todas as length(p)
      posicoes exceto a i-esima e p eh o vetor de parametros
cellerro = celula de matrizes tal que cellerro{1} = s(p), cellerro{i} = s(p + delta*e_i)
          Veja help cell, para maiores informacoes

Obs: s(p + delta*e_i) = avaliacao da funcao s em um ponto afastado de p por delta*e_i

Saida:
g = matriz gradiente
   cada coluna corresponde a uma variavel
```

Deterministicos\Geral\pperturbasist.m

```
function [yerropar,ysim,fcount] = pperturbasist(f,p,delta,vindsp,xmedido,varargin)
Retorna celula com perturbacoes do sistema f

Entrada:
f = funcao cujos parametros devem ser estimados
p = vetor de parametros de f
delta = valor infinitesimal (sugestao: 1e-6) de perturbacao de s(p)
vindsp = indices do vetor p que serao perturbados
xmedido = matriz de medicoes (atrator) com a qual cada simulacao de f deve ser
          comparada para criacao da funcao objetivo
          Cada coluna deve corresponder a uma variavel
parfun1,parfun2,.. = parametros da funcao objetivo f
                    (na ordem em que aparecem na interface da function)

Saida:
yerropar = celula de matrizes tal que yerropar{1} = s(p), yerropar{i} = s(p + delta*e_i)
          Veja help cell, para maiores informacoes
ysim = vetor coluna (apos pmat2vec, se necessario) resultante da simulacao
      de f com o vetor de parametros p
fcount = numero de avaliacoes da funcao f

Obs: s(p + delta*e_i) = avaliacao da funcao s em um ponto afastado de p por delta*e_i
```

Uma vez definidas as rotinas de apoio, parte-se para a implementação dos algoritmos de otimização propriamente ditos. As denominadas funções de algoritmos garantem sua correta utilização de duas maneiras principais: com a utilização das rotinas de apoio e com a disponibilização de uma interface padronizada.

No primeiro caso, garante-se a utilização de dados organizados em estruturas de dados convenientes e padronizadas. No segundo caso, garante-se a

eficiência na troca de informação com a 1ª Camada da aplicação, independentemente do algoritmo a ser utilizado. Especificamente no que diz respeito à garantia de padronização de comunicação com as funções de algoritmos, deve-se definir uma interface com parâmetros de entrada e saída padronizados. Esse padrão é apresentado a seguir.

Interface padrão dos algoritmos de otimização determinísticos
<pre>function [p,vp,sdiag] = algoritmo(p0,f,tol,maxit,vindsfobj,tmedido,xmedido,parfun1,parfun2,...) Descricao do algoritmo Entrada: p0 = ponto inicial do algoritmo f = funcao cujos parametros devem ser estimados tol = tolerancia (valor maximo de qualquer dos componentes do vetor gradiente = norm(g,inf)) maxit = numero maximo de iteracoes vindsfobj = vetor de indices das funcoes objetivo a serem minimizadas (por exemplo, vindsfobj = 2 implica que o algoritmo vai tentar estimar apenas o segundo elemento do vetor de parametros) tmedido = vetor de tempo com o qual o modelo f deve ser simulado xmedido = matriz de medicoes (atrator) com a qual cada simulacao de f deve ser comparada para criacao da funcao objetivo Cada coluna deve corresponder a uma variavel parfun1,parfun2,... = parametros da funcao f (na ordem em que aparecem na interface da function)* *Obs: A funcao deve ,*obrigatoriamente*, ter ao menos 1 argumento: - 0 vetor de parametros. Caso possua mais argumentos, o primeiro argumento *deve* ser o vetor de parametros Saida: p = vetor de parametros estimado vp = matriz correspondente ao vetor de vetores de parametros vp ilustra a evolucao da estimacao (a cada iteracao) (cada linha corresponde a uma iteracao) sdiag = estrutura de diagnostico: sdiag(1) = numero de avaliacoes da funcao f e conseqentemente da funcao obj (embora f nao seja a funcao obj) sdiag(2) = numero de iteracoes executadas sdiag(3) = numero de reinicializacoes sdiag(4) = tempo transcorrido (em segundos) durante execucao sdiag(5) = string de observacao</pre>

As exceções ao padrão estabelecido consistem nos algoritmos de *Nelder-Mead* e *Powell (DogLeg)*. No primeiro caso, trata-se de um algoritmo de características bastantes particulares, especialmente no que diz respeito à caracterização da evolução do processo de otimização².

²Descrita por modificações na posição e forma do objeto geométrico de dimensão igual ao espaço de busca

Quanto ao algoritmo de *Powell*, o fato de se tratar de um método baseado em *região de confiança* leva à necessidade de se estabelecer o raio inicial da região de busca (Δ_0), o que implica a criação de um parâmetro especial para este algoritmo. Importante observar que se tratam de exceções no que diz respeito apenas aos parâmetros de entrada. Todos os algoritmos possuem os mesmos parâmetros de saída.

Deterministicos\Algoritmos\pneldermead.m

```
function [p,vp,sdiag] = pneldermead(p0,f,tol,maxit,maxrestart,vindsfobj,parfun1,parfun2,..)
Executa algoritmo de Nelder-Mead
```

Entrada:

```
p0 = vertices do simplex inicial (n x n+1)
    em que n = size(p0,1) = numero de parametros a serem estimados
fobj = funcao objetivo*
tol = tolerancia (maxima distancia euclideana entre o melhor
    e o pior vertice do simplex)
maxit = numero maximo de iteracoes
maxrestart = numero maximo de reinicializacoes
parfun1,parfun2,.. = parametros da funcao objetivo f
    (na ordem em que aparecem na interface da function)**
```

*Nota: Um dos parametros dessa funcao eh a funcao objetivo. O motivo para isso estah no fato de que o algoritmo de Nelder-Mead nao foi concebido originalmente com o intuito de resolver problemas de estimacao de parametros (como eh o caso dos algoritmos de Levenberg-Marquardt e DogLeg, por exemplo). Sua concepcao se baseia na criacao de uma metodologia em que um objeto de dimensao n+1 "caminhe" sobre a superficie da funcao objetivo.

**Obs: A funcao deve ,*obrigatoriamente*, ter ao menos 1 argumento:
 - 0 vetor de parametros.
 Caso possua mais argumentos, o primeiro argumento *deve* ser o vetor de parametros

Saida:
 padrao

Deterministicos\Algoritmos\pdogleg.m

```
function [p,vp,sdiag] =
pdogleg(p0,f,tol,maxit,vindsfobj,tmedido,xmedido,delta0,parfun1,parfun2,..)
Executa algoritmo de DogLeg (Powell)
```

Entrada:

```
delta0 = raio inicial da regioao de confianca
    (sugestao: delta0 = 1e-5*(1 + norm(p0)))
```

Saida:
 padrao

Todos os demais algoritmos possuem exatamente a mesma interface. Devido a isso, apenas seus protótipos e descrições serão citados.

Deterministicos\Algoritmos\pgradiente.m
--

<pre>function [p,vp,sdiag] = pgradiente(p0,f,tol,maxit,vindsfobj,tmedido,xmedido,parfun1,parfun2,...) Executa algoritmo do Gradiente</pre>
--

Deterministicos\Algoritmos\pgradconj.m

<pre>function [p,vp,sdiag] = pgradconj(p0,f,tol,maxit,vindsfobj,tmedido,xmedido,parfun1,parfun2,...) Executa algoritmo do Gradiente Conjugado</pre>

Deterministicos\Algoritmos\pgradconjpre.m
--

<pre>function [p,vp,sdiag] = pgradconjpre(p0,f,tol,maxit,vindsfobj,tmedido,xmedido,parfun1,parfun2,...) Executa algoritmo do Gradiente Conjugado Pre-Condicionado</pre>

Deterministicos\Algoritmos\pgaussnewton.m
--

<pre>function [p,vp,sdiag] = pgaussnewton(p0,f,tol,maxit,vindsfobj,tmedido,xmedido,parfun1,parfun2,...) Executa algoritmo de Gauss-Newton</pre>

Deterministicos\Algoritmos\plevmar.m

<pre>function [p,vp,sdiag] = plevmar(p0,f,tol,maxit,vindsfobj,tmedido,xmedido,parfun1,parfun2,...) Executa algoritmo de Levenberg-Marquardt</pre>

5.4 Algoritmos Estocásticos

Assim como no caso determinístico, as rotinas relacionadas à implementação dos algoritmos de otimização estocásticos podem ser divididas em funções gerais e funções de algoritmos. A distinção de funcionalidades desses dois grupos segue aquela apresentada para os algoritmos determinísticos. No caso dos algoritmos estocásticos, uma importante particularidade deve ser observada. A existência de algoritmos genéticos demanda a implementação de uma série de funções auxiliares com o intuito de implementar os operadores genéticos. A implementação desses operadores (e de algumas rotinas de apoio) é ilustrada a seguir.

Estocasticos\Geral\pclusterlim.m

```
function [Pfim,Ffim] = pclusterlim(Pini,Fini,n)
Retorna n individuos selecionados segundo a tecnica
de agrupamento (cluster) para manutencao da diversidade
no conjunto de individuos (segunda meta na otimizacao
multiobjetivo)

Entrada:
Pini = Matriz com populacao inicial (a ser avaliada)
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini
n = Numero de individuos a serem selecionados

Saida:
Pfim = Matriz com populacao nao dominada
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pfim

Obs:
Tanto em Pini quanto em Fini, cada coluna deve corresponder a um
individuo
```

Estocasticos\Geral\pcriapesos.m

```
function w = pcriapesos(m,npesos)
Retorna matriz (vetor de vetores) de pesos

Entrada:
m = numero de funcoes objetivo
npesos = numero de vetores de pesos

Saida
w = matriz (vetor de vetores) de pesos
Cada coluna corresponde a um vetor de pesos i tal que sum(w(:,i)) = 1
w tem dimensao m x npesos
```

Estocasticos\Geral\pcriarank.m

```
function r = pcriarank(Fini)
Retorna rank dos individuos cujos valores das funcoes objetivo estao
representados em Fini (classifica individuos segundo dominancia)

Entrada:
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini
Cada coluna consiste em um individuo

Saida:
r = vetor com o rank de cada individuo
```

Estocasticos\Geral\pcruzamento.m

```
function Mpopfim = pcruzamento(Mpopini,pcros,nvar,tcros)
Retorna genotipo de uma populacao inicial apos sofrer cruzamento

Entrada:
Mpopini = genotipo da populacao inicial
          (cada coluna consiste em um individuo representado pelos seus
           size(Mpopini,1) genes)
pcros = probabilidade de cruzamento
nvar = numero de variaveis de otimizacao
tcros = tipo de cruzamento
        (1 => 1 ponto; 2 => 2 pontos; 3 => ponto/variavel; 4 = uniforme)

Saida:
Mpopfim = genotipo da populacao apos sofrer cruzamento
```

Estocasticos\Geral\pdistancia.m

```
function d = pdistancia(Pini,Fini,indiv,flagesp,flagtipo)
Retorna distancias entre individuo indiv e todos os outros individuos

Entrada:
Pini = matriz com fenotipo da populacao
Fini = matriz com avaliacoes das funcoes para cada individuo
indiv = indice do individuo cujas distancias devem ser calculadas*
        Portanto, indiv eh o indice de uma coluna de Pini (e Fini)
flagesp = flag que indica em que espaco as distancias devem ser calculadas
          0 => espaco de objetivos (default); 1 => espaco de variaveis
flagtipo = flag que indica o tipo de norma a ser utilizada (default = 2)
           Para flagtipo inteiro positivo qualquer (p) => norma_p
           Para flagtipo = inf => max(abs)
           Para flagtipo = -inf => min(abs)

*indiv pode tambem ser um vetor de indices de individuos.
Nesse caso, d serah uma matriz em que cada linha i corresponde
a um individuo indiv(i).

Saida:
d = vetor linha de distancias (dimensao 1 x size(Pini,2))

Obs: Em todas as matrizes (Pini,Fini) cada coluna consiste em um individuo
```

Estocasticos\Geral\pdomina.m

```
function [Pfim,Ffim] = pdomina(Pini,Fini)
Retorna individuos nao dominados

Entrada:
Pini = Matriz com populacao inicial (a ser avaliada)
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini

Saida:
Pfim = Matriz com populacao nao dominada
Ffim = Matriz com avaliacoes das funcoes para cada individuo de Pfim

Obs: Em todas as matrizes (Pini,Fini,Pfim,Ffim) cada coluna consiste em um individuo
```


Estocasticos\Geral\pelitismo.m

```
function [Pfim,Pelitfim,Felitfim,avaliar] = pelitismo(Pini,Pelitini,Fini,Felitini,npopext)
Executa tarefas elitistas
```

Entrada:

```
Pini = Matriz com fenotipo da populacao online inicial
Pelitini = Matriz com fenotipo da populacao offline (elite) inicial
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini
Felitini = Matriz com avaliacoes das funcoes para cada individuo de Pelitini
npopext = Tamanho desejado para a populacao offline
```

Saida:

```
Pfim = Matriz com fenotipo da populacao online final
Pelitfim = Matriz com fenotipo da populacao offline (elite) final
FxIfim = Matriz com avaliacoes das funcoes para cada individuo de MID final
Felitfim = Matriz com avaliacoes das funcoes para cada individuo de MIDelit final
avaliar = 1 => Necessidade de se avaliar novamente as funcoes de otimizacao
```

Obs: Em todas as matrizes (Pini,Pelitini,Fini,Felitini,Pfim,Pelitfim,Felitfim) cada coluna consiste em um individuo

Estocasticos\Geral\pfen2gen.m

```
function Mpop = pfen2gen(Pini,nbits,Lvar)
Transforma individuos decodificados (fenotipo)
em individuos com codificacao binaria (genotipo)
```

Entrada:

```
Pini = Matriz de individuos
    Linha i: Caracteristica/variavel i de todos os individuos
    Coluna j: Conjunto de caracteristicas/variaveis do individuo j
nbits = Numero de bits por caracteristica/variavel
Lvar = Matriz 2xv contendo um vetor coluna com os limites
    [xmin; xmax] para cada variavel
```

Saida:

```
Mpop = Pini com decodificacao binaria (cada individuo consiste em uma coluna)
```

Obs: Razoavelmente bem testado para numeros negativos

Estocasticos\Geral\pgen2fen.m

```
function Pfim = pgen2fen(Mpop,nvar,Lvar)
Transforma individuos com codificacao binaria (genotipo)
em individuos decodificados (fenotipo)
```

Entrada:

```
Mpop = Matriz de individuos
    Linha i: Gene/bit i de todos os individuos
    Coluna j: Conjunto de genes/bits do individuo j
nvar = Numero de variaveis de otimizacao (cromossomos)
Lvar = Matriz 2xv contendo um vetor coluna com os limites
    [xmin; xmax] para cada variavel
```

Saida:

```
Pfim = Mpop decodificado (cada individuo consiste em uma coluna)
```

Estocasticos\Geral\pmutacao.m

```
function Mpopfim = pmutacao(Mpopini,pmut,nvar,tmut)
Retorna genotipo de uma populacao inicial apos sofrer mutacao

Entrada:
Mpopini = genotipo da populacao inicial
          (cada coluna consiste em um individuo representado pelos seus
           size(Mpopini,1) genes)
pmut = probabilidade de mutacao
nvar = numero de variaveis de otimizacao
tmut = tipo de mutacao
       (1 => bit a bit; 2 => bit/variavel; 3 => bit/individuo)

Saida:
Mpopfim = genotipo da populacao apos sofrer mutacao
```

Estocasticos\Geral\pretiguais.m

```
function [Pfim,Ffim] = pretiguais(Pini,Fini,tol)
Retira individuos cuja distancia euclideana no espaco
de objetivos seja menor que tol

Entrada:
Pini = Matriz com populacao inicial (a ser avaliada)
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini
tol* = Tolerancia

*Obs: Opcional (default = 0.1)

Saida:
Pfim = Matriz com populacao nao dominada
Ffim = Matriz com avaliacoes das funcoes para cada individuo de Pfim

Obs: Em todas as matrizes (Pini,Fini,Pfim,Ffim) cada coluna consiste em um individuo
```

Estocasticos\Geral\pselprop.m

```
function [Pfim,Ffim] = pselprop(Pini,Fini)
Retorna individuos selecionados via selecao proporcional
e suas respectivas funcoes objetivo

Entrada:
Pini = Matriz com populacao inicial (a ser avaliada)
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini

Saida:
Pfim = Matriz com populacao nao dominada
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pfim

Obs:
Tanto em Pini quanto em Fini, cada coluna deve corresponder a um
individuo
```

Estocasticos\Geral\pnsga.m

```
function Fshar = pnsga(Pini,Fini,nindiv,Lvar,sharini,flagesp,flagtipo)
Retorna valores da funcao de partilha devido aa divisao dos individuos em
fronteiras segundo NSGA

Entrada:
Pini = matriz com fenotipo da populacao
Fini = matriz com avaliacoes das funcoes para cada individuo
nindiv = numero de individuos em Pini sobre os quais o operador
        nsga deve ser aplicado (Pini(:,1:nindiv))
Lvar = limites para espaco de variaveis
        Matriz 2xv contendo um vetor coluna com os limites
        [xmin; xmax] para cada variavel
sharini = valor de inicializacao da funcao de partilha: melhores individuos
        (sugestao: sharini = 100)
flagesp = flag que indica em que espaco as distancias devem ser calculadas
        0 => espaco de objetivos (default); 1 => espaco de variaveis
flagtipo = flag que indica o tipo de norma a ser utilizada (default = 2)
        Para flagtipo inteiro positivo qualquer (p) => norma_p
        Para flagtipo = inf => max(abs)
        Para flagtipo = -inf => min(abs)

Saida:
Fshar = vetor linha com os valores da funcao de partilha para cada
        individuo

Obs: Em todas as matrizes (Pini,Fini,Fshar) cada coluna consiste em um individuo
```

Estocasticos\Geral\pselsus.m

```
function [Pfim,Ffim] = pselsus(Pini,Fini)
Retorna individuos selecionados via selecao SUS
("Stochastic Universal Sampling") e suas respectivas
fitness function*

*Obs: Pode levar alguns segundos.

Entrada:
Pini = Matriz com populacao inicial (a ser avaliada)
Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini

Saida:
Pfim = Matriz com populacao nao dominada
Ffim = Matriz com avaliacoes das funcoes para cada individuo de Pfim

Obs:
Tanto em Pini quanto em Fini, cada coluna deve corresponder a um
        individuo
```

Estocasticos\Geral\pgerapop.m

```
function Mpop = pgerapop(npop, ngenes)
Gera populacao de individuos

Entrada:
  npop = Numero de individuos (tamanho da populacao)
  ngenes = Numero de genes por individuo

Saida:
  Mpop = Matriz de individuos
        Linha i: Gene/bit i de todos os individuos
        Coluna j: Conjunto de genes/bits do individuo j

Obs: Na proxima versao, Mpop sera transversa
```

Estocasticos\Geral\ptorneio.m

```
function [Pfim,Ffim] = ptorneio(Pini,Fini,n)
Retorna n individuos* vencedores de n torneios binarios
e suas respectivas funcoes objetivo

*Obs: Os pares de competidores sao escolhidos
aleatoriamente.

Entrada:
  Pini = Matriz com populacao inicial (a ser avaliada)
  Fini = Matriz com avaliacoes das funcoes para cada individuo de Pini
  n = Numero de torneios (e consequentemente o numero de individuos de Pfim)

Saida:
  Pfim = Matriz com populacao nao dominada
  Ffim = Matriz com avaliacoes das funcoes para cada individuo de Pfim

Obs: Em todas as matrizes (Pini,Fini,Pfim,Ffim), cada coluna corresponde
a um individuo
```

O único algoritmo estocástico implementado não classificado como genético consiste no método de *Simulated Annealing (Metropolis)*. Devido a isso, trata-se do único algoritmo com interface não padronizada. A função que representa a implementação desse método é apresentada a seguir.

Estocasticos\Algoritmos\psimann.m

```
function [p,vp,sdiag] =
    psimann(p0,fobj,maxit,Lvar,temp0,fresfr,rp,nptemp,vindsfobj,parfun1,parfun2,..)
Executa algoritmo de Simulated Annealing (Metropolis)

Entrada:
p0 = ponto inicial do algoritmo
fobj = funcao objetivo
maxit = numero maximo de iteracoes
      (neste caso, trata-se do numero de ciclos de resfriamento)
Lvar = limites para espaco de variaveis
      Matriz 2xv contendo um vetor coluna com os limites
      [xmin; xmax] para cada variavel
temp0 = temperatura inicial (sugestao: temp0 = 10)
fresfr = fator de resfriamento (sugestao: fresfr = 0.92)
        fator de reducao da temperatura a cada
        iteracao (ciclo de resfriamento)
rp = raios de busca no espaco de parametros
     valor maximo do passo aleatorio ("random walk")
     (sugestao: rp = 0.1*ones(size(p0)))
nptemp = numero de passos por temperatura (sugestao: nptemp = 10)
parfun1,parfun2,.. = parametros da funcao objetivo f
                    (na ordem em que aparecem na interface da function)**

**Obs: A funcao deve ,*obrigatoriamente*, ter ao menos 1 argumento:
      - 0 vetor de parametros.
      Caso possua mais argumentos, o primeiro argumento *deve* ser o
      vetor de parametros
```

Quanto à implementação dos algoritmos genéticos, uma importante característica deve ser observada. Se, por um lado, a implementação de todos os algoritmos genéticos é suficientemente parecida a ponto de se poder definir uma interface padronizada, por outro lado, possuem implementação bastante particular, e isso impede a padronização de sua interface se comparada com a dos algoritmos determinísticos. Tendo isso em vista, define-se a seguinte interface padrão para as funções que implementam algoritmos genéticos:

Interface padrão dos algoritmos de otimização genéticos

```
function [p, vp, sdiag] =
    algoritmo(fobj, maxit, Lvar, npop, npopex, nbit, pcros, pmut, tcros, tmut, parfun1, parfun2.. )
Descricao do algoritmo

Entrada:
fobj = funcao objetivo*
maxit = numero maximo de iteracoes (geracoes, neste caso)
Lvar = limites para espaco de variaveis
      Matriz 2xv contendo um vetor coluna com os limites
      [xmin; xmax] para cada variavel
npop = tamanho maximo da populacao interna
npopex = tamanho maximo da populacao externa
nbit = numero de bits por variavel de otimizacao (sugestao: nbit = 12)
pcros = probabilidade de cruzamento (sugestao: pcros = 0.9)
pmut = probabilidade de mutacao (sugestao: pcros = 0.01)
tcros = tipo de cruzamento: 1 = 1 Pt; 2 = 2Pt; 3 = Pt/var; 4 = uniforme
tmut = Tipo de mutação: 1 = bit a bit; 2 = bit/variavel; 3 = bit/individuo
parfun1, parfun2,.. = parametros da funcao objetivo f
                    (na ordem em que aparecem na interface da function)**

*Nota: Note que, para os algoritmos geneticos, a funcao objetivo deve ser
passada como parametro. Isso se deve aa propria definicao de
algoritmo genetico de utilizar uma populacao (que se modifica
dinamicamente) com o intuito de localizar pontos otimos de uma
funcao objetivo.

**Obs: A funcao deve ,*obrigatoriamente*, ter ao menos 1 argumento:
- 0 vetor de parametros.
Caso possua mais argumentos, o primeiro argumento *deve* ser o
vetor de parametros

Saida:
padrao
```

As exceções ao padrão estabelecido consistem nos algoritmos de *WBGA* e *MOGA*. Em ambos os casos, particularidades relativas às regras utilizadas para a evolução dos algoritmos em direção ao ótimo da função analisada levam à necessidade de parâmetros especiais: o número de vetores de pesos (n_{pesos}) para o *WBGA* e o raio das hiperesferas que representam cada nicho (σ_{share}) para o *MOGA*.

Estocasticos\Algoritmos\pwbga.m

```
function [p, vp, sdiag] =
  pwbga(fobj, maxit, Lvar, npop, npopex, nbit, pcros, pmut, tcros, tmut, npesos, parfun1, parfun2..)
Executa algoritmo genetico Weight-Based Genetic Algorithm (WBGA)

Entrada:
  tcros = tipo de cruzamento: 1 = 1 Pt; 2 = 2Pt; 3 = Pt/var; 4 = uniforme
          Padrao WBGA => uniforme (tcros = 4)
  tmut = Tipo de mutação: 1 = bit a bit; 2 = bit/variavel; 3 = bit/individuo
          Padrao WBGA => bit/individuo (tmut = 3)
  npesos = numero de vetores de pesos

Saida:
  padrao
```

Estocasticos\Algoritmos\pmoga.m

```
function [p, vp, sdiag] =
  pmoga(fobj, maxit, Lvar, npop, nbit, pcros, pmut, tcros, tmut, sigmash, parfun1, parfun2..)
Executa algoritmo genetico Multiple-Objective Genetic Algorithm (MOGA)

Entrada:
  tcros = tipo de cruzamento: 1 = 1 Pt; 2 = 2Pt; 3 = Pt/var; 4 = uniforme
          Padrao MOGA => Pt/var (tcros = 3)
  tmut = Tipo de mutação: 1 = bit a bit; 2 = bit/variavel; 3 = bit/individuo
          Padrao MOGA => bit/individuo (tmut = 3)
  sigmash = raio das hiperesferas que representam cada nicho
            (sugestao: sigmash = 0.5)

Saida:
  padrao
```

Todos os demais algoritmos possuem exatamente a mesma interface. Devido a isso, apenas seus protótipos e descrições serão citados. Importante observar que, neste caso, explicitam-se os tipos de operadores de cruzamento e mutação mais comumente utilizados para cada algoritmo.

Estocasticos\Algoritmos\pvega.m

```
function [p, vp, sdiag] =
  pvega(fobj, maxit, Lvar, npop, npopex, nbit, pcros, pmut, tcros, tmut, parfun1, parfun2..)
Executa algoritmo genetico Vector Evaluated Genetic Algorithm (VEGA)

Entrada:
  tcros = tipo de cruzamento: 1 = 1 Pt; 2 = 2Pt; 3 = Pt/var; 4 = uniforme
          Padrao VEGA => uniforme (tcros = 4)
  tmut = Tipo de mutação: 1 = bit a bit; 2 = bit/variavel; 3 = bit/individuo
          Padrao VEGA => bit/individuo (tmut = 3)

Saida:
  padrao
```

Estocasticos\Algoritmos\pemnsga.m

```
function [p,vp,sdiag] =
pemnsga(fobj,maxit,Lvar,npop,npopex,nbit,pcros,pmut,tcros,tmut,parfun1,parfun2..)
Executa algoritmo genetico Non-Dominated Sorting Genetic Algorithm II (NSGA II)

Entrada:
tcros = tipo de cruzamento: 1 = 1 Pt; 2 = 2Pt; 3 = Pt/var; 4 = uniforme
      Padrao NSGA2 => nao indicado
tmut = Tipo de mutação: 1 = bit a bit; 2 = bit/variavel; 3 = bit/individuo
      Padrao VEGA => nao indicado

Saida:
padrao
```

Estocasticos\Algoritmos\pspea.m

```
function [p,vp,sdiag] =
pspea(fobj,maxit,Lvar,npop,npopex,nbit,pcros,pmut,tcros,tmut,parfun1,parfun2..)
Executa algoritmo genetico Strength Pareto Evolutionary Algorithm (SPEA)

Entrada:
tcros = tipo de cruzamento: 1 = 1 Pt; 2 = 2Pt; 3 = Pt/var; 4 = uniforme
      Padrao SPEA => nao indicado
tmut = Tipo de mutação: 1 = bit a bit; 2 = bit/variavel; 3 = bit/individuo
      Padrao SPEA => nao indicado

Saida:
padrao
```

5.5 Modelo

As rotinas responsáveis pela representação do modelo cujos parâmetros devem ser estimados podem ser divididas em três categorias:

1. funções gerais;
2. função do modelo;
3. diagramas Simulink;

Assim como descrito para os algoritmos de otimização, as funções gerais têm por principais objetivos a verificação de inconsistências nos sinais (de entrada e saída do modelo) e da existência de erros gerais de execução das rotinas. Neste caso, a funcionalidade de tratamento dos dados se torna evidente quando se observa a existência de rotinas de rotação de coordenadas. As funções gerais são listadas a seguir.

Modelos\Geral\pcriaestados.m

```
function [imu,linvel,inertial,wind,gps,lindist] =
  pcriaestados(xmedido,ftmin,dgradin,ftmout,dgradout,flagplot)
  Cria estados a partir das medicoes (com conversoes tipicas)

Entrada:
  xmedido = matriz de medicoes*
  ftmin = flag de indicacao de unidades de xmedido (feet ou metro)
          ftmin = 0 => feet / ftmin = 1 => metro
  dgradin = flag de indicacao de unidades de xmedido (grau ou radiano)
          dgradin = 0 => grau / dgradin = 1 => radiano
  ftmout = flag de indicacao de unidades de imu, linvel, inertial, wind e gps
          (feet ou metro)
          ftmout = 0 => feet / ftmout = 1 => metro
  dgradout = flag de indicacao de unidades de imu, linvel, inertial, wind e gps
            (grau ou radiano)
            dgradout = 0 => grau / dgradout = 1 => radiano
  flagplot** = 0 => nao traca graficos
              = 1 => traca graficos

Saida:
  imu = struct com medicoes relativas aa unidade de medicao inercial:
        ax, ay, az, p, q, r
  linvel = struct com medicoes de velocidades lineares:
          u, v, w
  inertial = struct com medicoes dos angulos de Euler:
            phi, theta, psi
  gps = struct com medicoes provenientes de gps:
        h, lat, lgt, odo
  lindist = struct com variaveis provenientes medicoes via gps (veja pgps2ned):
          xe, ye, ze

*Obs: Matriz com 25 colunas com as seguintes (e respectivas) medicoes:
      | ax | ay | az | p | q | r | u | v | w | phi | theta | psi |
      | udot | vdot | wdot | phidot | thetadot | psidot | vt | alpha | beta |
      | h | lat | lgt | odo |
      **Caso nao se disponha de qualquer dessas variaveis, basta preencher a coluna
      correspondente com zeros.

**Obs: Parametro opcional (default = 0).
```

Modelos\Geral\pecef2ned.m

```
function vned = pecef2ned(x_ecef,y_ecef,z_ecef,lat,lgt)
  Rotaciona vetor representado no referencial ECEF de tal forma que
  seja representado no referencial NED

Entrada:
  vecef = vetor coluna representado no referencia ECEF (3 posicoes = [x; y; z])
          Caso se trate de uma serie temporal, basta concatenar os vetores
          em uma matriz vecef tal que cada linha corresponda a um instante
          de tempo e cada coluna corresponda a uma variavel (x, y e z)
  lat, lgt = latitude e longitude do ponto (obtido via gps, p.ex.)
          Caso se trate de uma serie temporal, basta criar vetores lat e
          lgt tal que cada elemento corresponda a um instante de tempo

Saida:
  vned = vetor representado no referencial NED
```

Modelos\Geral\pgps2ecef.m

```
function vecef = pgps2ecef(vgps)
Transforma vetor de medicoes via GPS em vetor representado
no referencial ECEF

Entrada:
  vgps = vetor coluna representado de medicoes via GPS
        (3 posicoes = [altura; latitude; longitude])
        Caso se trate de uma serie temporal, basta concatenar os vetores
        em uma matriz vgps tal que cada linha corresponda a um instante
        de tempo e cada coluna corresponda a uma variavel

Saida:
  vecef = vetor representado no referencial ECEF
```

Modelos\Geral\pgps2ned.m

```
function vned = pgps2ned(vgps)
Transforma vetor de medicoes via GPS em vetor representado
no referencial NED (via transformacao GPS -> ECEF e
rotacao ECEF -> NED)

Entrada:
  vgps = vetor coluna representado de medicoes via GPS
        (3 posicoes = [altura; latitude; longitude])
        Caso se trate de uma serie temporal, basta concatenar os vetores
        em uma matriz vgps tal que cada linha corresponda a um instante
        de tempo e cada coluna corresponda a uma variavel

Saida:
  vned = vetor representado no referencial NED

Obs: Para maiores informacoes, refira-se a pgps2ecef e pecef2ned
```

Modelos\Geral\pganhobias.m

```
function [kacc,bacc,sigmaacc,kgir,bgir,sigmagir] = pganhobias(nomeimu)
Retorna valores de ganho e polarizacao da IMU especificada

Entrada:
  nomeimu = nome (string) da IMU desejada

Saida*:
  kacc = ganho maximo dos acelerometros
  bacc = polarizacao maxima dos acelerometros
  sigmaacc = desvio padrao do ruido de medicao tipico dos acelerometros
  kgir = ganho maximo dos girometros
  bgir = polarizacao maxima dos girometros
  sigmagir = desvio padrao do ruido de medicao tipico dos girometros

*Obs: Esses valores sao tais que
      acc_medida = (1 +- kacc)*acc_real +- bacc
      e
      gir_medida = (1 +- kgir)*gir_real +- bgir
**Alem disso, consideram-se medicoes em m/s2 e graus/s
```

Modelos\Geral\pnormaliza.m

```
function En = pnormaliza(E)
Retorna matriz de estados normalizada

Entrada:
  E = matriz de estados nao-normalizada

Saida:
  En = matriz de estados normalizada
      (os valores de cada coluna variam entre -1 e 1)

Obs: Tanto para E quanto para En, cada coluna corresponde aa serie
      temporal de um estado
```

Modelos\Geral\prmsemocd.m

```
function ind = prmsemocd(Pini,E0,tmedido,xmedido,options,monomulti,vindsobs)
Retorna o rmse entre a simulacao do modelo cinematico simulado com
cada uma das colunas de Pini e xmedido

Entrada:
  Pini = matriz (de individuos*) de 12 linhas com os parametros
        kp, bp, kq, bq, kr, br, kax, bax, kay, bay, kaz, baz.
        Cada coluna deve ser tal que as linhas sejam essas respectivamente, da 1a aa 12a
        (cada coluna corresponde a um individuo*)
  E0 = condicoes iniciais
  tmedido = vetor de tempo com o qual o modelo cinematico deve ser simulado
  xmedido = atrator com o qual a simulacao deve ser comparada para calculo do rmse
           Cada coluna deve corresponder a uma variavel
           (1a: vt, 2a: alpha, 3a: beta, 4a: xe, 5a: ye, 6a: ze)
  options = <use options = Ein>, em que:*
           Ein = uma matriz com vetores correspondentes aas variaveis
                (estados) de entrada do modelo cinematico, tal que
                Ein = [vpm vqm vrm vaxm vaym vazm vgm]
  monomulti = flag para indicacao de problema mono ou multiobjetivo (default = 0):
              = 0 => mono-objetivo: A cada coluna de Pini (a cada individuo*), atribui-se
                                   um unico indice independente do numero de funcoes
                                   objetivo (= numero de colunas de xmedido).
                                   Esse indice sera a soma dos size(xmedido,2) indices.
              = 1 => multiobjetivo: A cada coluna de Pini (a cada individuo*), atribuem-se
                                   size(xmedido,2) indices
  vindsobs = vetor de indices de variaveis observadas (default = [1 2 3 4 5 6])
            Por exemplo, caso se deseje calcular o rmse observando-se apenas as
            variaveis vt, alpha, beta e ze, basta fazer vindsobs = [1 2 3 6]

*Obs: A nomenclatura utilizada nesses comentarios se aplica aos algoritmos geneticos

Saida
  ind = indice correspondente ao rmse entre xmedido e a simulacao
```

Modelos\Geral\pvarglobais.m

```
pvarglobais
Declaracao de variaveis globais: Necessario para utilizacao do modelo via simulink

Obs.: pvarglobais nao se trata de fato de uma function, mas apenas um m-file.
```

A função do modelo consiste na representação da estratégia adotada para flexibilização da aplicação quanto ao modelo utilizado. Trata-se de uma rotina com interface padronizada que tem por principal objetivo encapsular a execução do modelo a ser utilizado. Com isso, consegue-se garantir que qualquer modelo pode ser utilizado, desde que se mantenha a interface da rotina que o encapsula. A função do modelo utilizada para reconstrução de trajetória e estimação de parâmetros de aeronaves neste trabalho é apresentada a seguir.

```

Modelos\MatlabFunction\pmodcshell.m

function [tsim,Esim] = pmodcshell(p,t,E0,options,vindsobs,varruído)
Funcao que encapsula a integracao do Modelo Cinematico - aplicado a
Dinamica de Aeronaves - (veja pmodelocinematico.mdl)

Entrada:
p = parametros do sistema (valores de ganho e polarizacao) =
    [kp; bp; kq; bq; kr; br; kax; bax; kay; bay; kaz; baz; kg; bg; ..
    kvt; bvt; kalpha; balpha; kbeta; bbeta; kxe; bxe; kye; bye; kze; bze; ..
    kphi; bphi; ktheta; btheta; kpsi; bpsi; ku; bu; kv; bv; kw; bw]
t = vetor de tempo para integracao
E0 = condicoes iniciais de cada um dos estados (vetor linha) =
    [wind.vt(1) wind.alpha(1) wind.beta(1) ..
    linvel.u(1) linvel.v(1) linvel.w(1) ..
    inertial.psi(1) inertial.theta(1) inertial.phi(1) ..
    lindist.xe(1) lindist.ye(1) lindist.ze(1)]
options = <use options = Ein>, em que:*
    Ein = uma matriz com vetores correspondentes aas variaveis
    (estados) de entrada do modelo cinematico, tal que
    Ein = [vpm vqm vrm vaxm vaym vazm vgm]
vindsobs = vetor de indices de variaveis observadas
varruído = variancia do ruido de medicao a ser inserido apos
    a simulacao (default = varruído = 0)

Saida
Esim = estados calculados apos integracao - cada coluna corresponde a um estado
    (outvt, outalpha, outbeta, outxe, outye, outze, outphi, outtheta, outpsi,
    outu, outv, outw)
tsim = vetor de tempo utilizado por ode45 para integracao
    (pode ser diferente de t)

*Obs: Parametro utilizado, em principio, apenas para funcoes cuja
integracao seja feita com a utilizacao (explicita) da funcao ode45.
Neste caso a integracao eh feita via simulink, o que dispensa a
utilizacao desse parametro com essa utilidade. Contudo, com o objetivo
de padronizar a interface das funcoes desenvolvidas, eh utilizado
como matriz de variaveis (estados) de entrada do modelo cinematico.

Obs: Embora a criacao desta funcao pareca de pequena relevancia, tem
grande contribuicao na flexibilizacao da aplicacao como um todo no que
diz respeito aa capacidade de generalizar a utilizacao dos algoritmos
implementados para, em principio, qualquer funcao sob a forma de uma function.

```

O terceiro elemento utilizado para representação do modelo consiste nos diagramas Simulink. Além da facilidade de implementação, esse ambiente oferece uma importante vantagem: a possibilidade de modularização do modelo de forma bastante intuitiva. O primeiro passo para a implementação do modelo consiste na definição de blocos com funcionalidades específicas. Esses blocos compõem uma biblioteca de funções, representada na Figura 5.3.

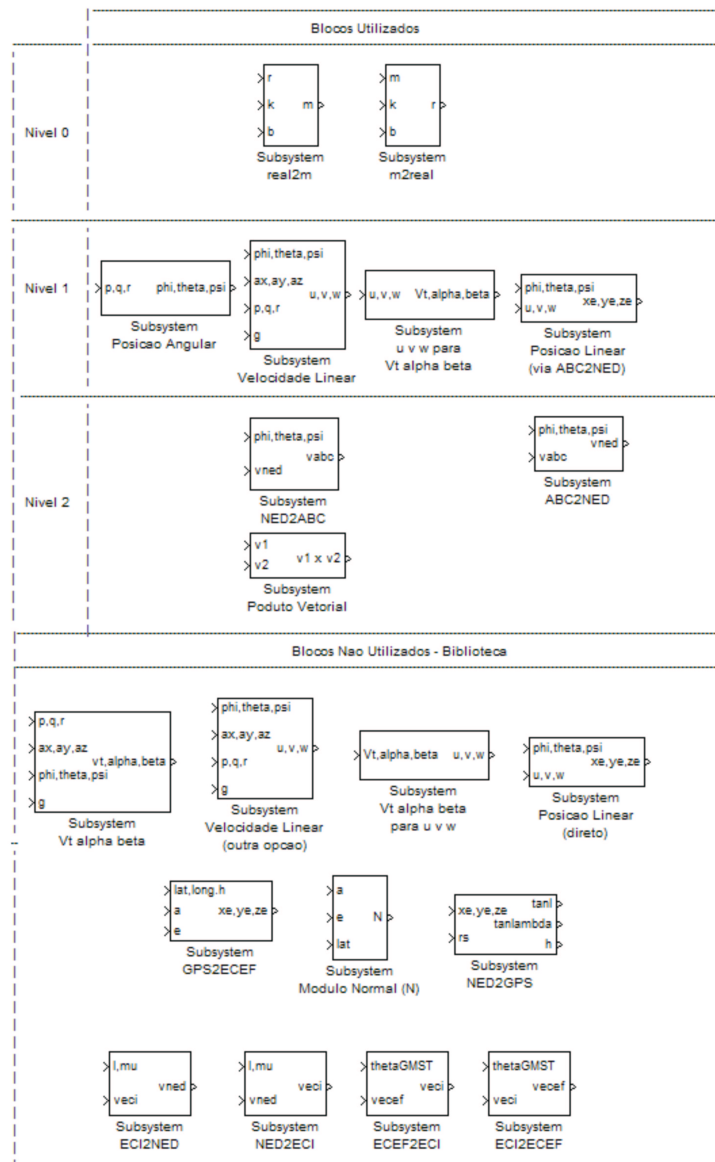


Figura 5.3: Biblioteca de blocos para simulação do modelo cinemático.

Importante observar na Figura 5.3 a divisão dos blocos em duas categorias: “Blocos Utilizados” e “Blocos Não Utilizados - Biblioteca”. Com o desenvolvimento do projeto, tornou-se conveniente utilizar algumas estratégias em detrimento de outras. Essa mudança de perspectiva demandou a criação de novos blocos e, conseqüentemente, a não utilização de outros. Contudo, os blocos não utilizados cumprem funções equivalentes às dos seus substitutos, e este é o motivo pelo qual foram mantidos no projeto.

Além disso, deve-se observar que os blocos utilizados são separados em níveis. Essa classificação busca facilitar o entendimento a respeito da finalidade do bloco. Mais especificamente, o próprio alinhamento dos blocos ilustra a existência de encapsulamentos de tal forma que blocos de níveis superiores utilizam os blocos de níveis inferiores (alinhados abaixo) para desempenharem suas funções³.

Por conveniência apenas os blocos utilizados no projeto serão detalhados. Os dois primeiros consistem nos blocos responsáveis pela inserção dos parâmetros (estimados) no modelo de acordo com a equação (5.4). De fato, enquanto o bloco ilustrado na Figura 5.4.a implementa a transformação “valor real” \rightarrow “medição” e a 5.4.b implementa a transformação “medição” \rightarrow “valor real”.

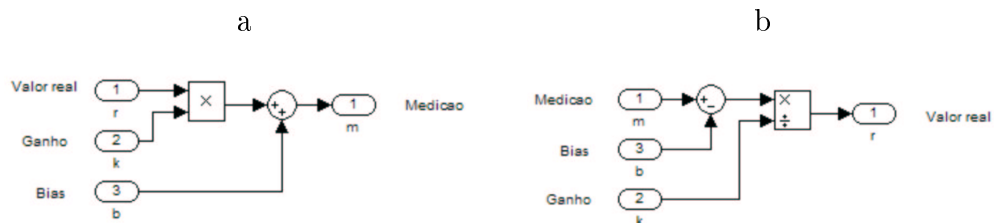


Figura 5.4: Blocos de transformação de variáveis medidas. a) Transformação “valor real” \rightarrow “medição”; b) Transformação “medição” \rightarrow “valor real”

Em seguida, ilustra-se na Figura 5.5 bloco responsável pela determinação da atitude da aeronave: equação (2.30), página 25.

³De fato, essa regra deve ser aplicada apenas do “Nível 1” em diante.

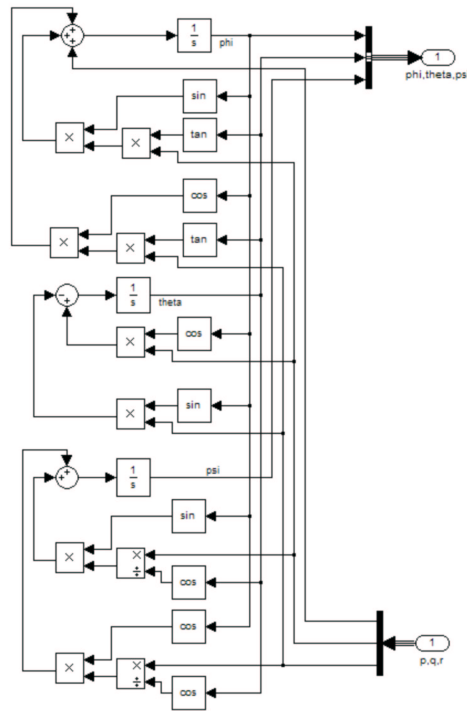


Figura 5.5: Bloco para determinação da atitude da aeronave.

A Figura 5.6 ilustra a implementação do bloco responsável pelo cálculo do vetor velocidade linear da aeronave: equação (2.12). Para isso, utiliza dois blocos auxiliares: um bloco de rotação de coordenadas $NED \rightarrow ABC$ (Figura 5.7) e outro que implementa a operação de produto vetorial (Figura 5.8).

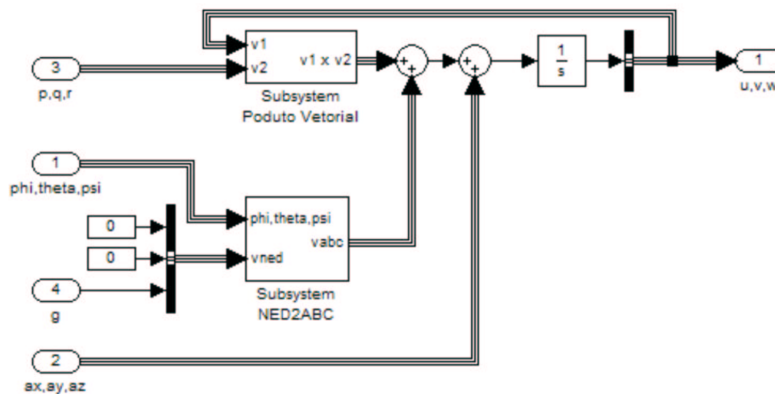


Figura 5.6: Bloco para cálculo do vetor velocidade linear.

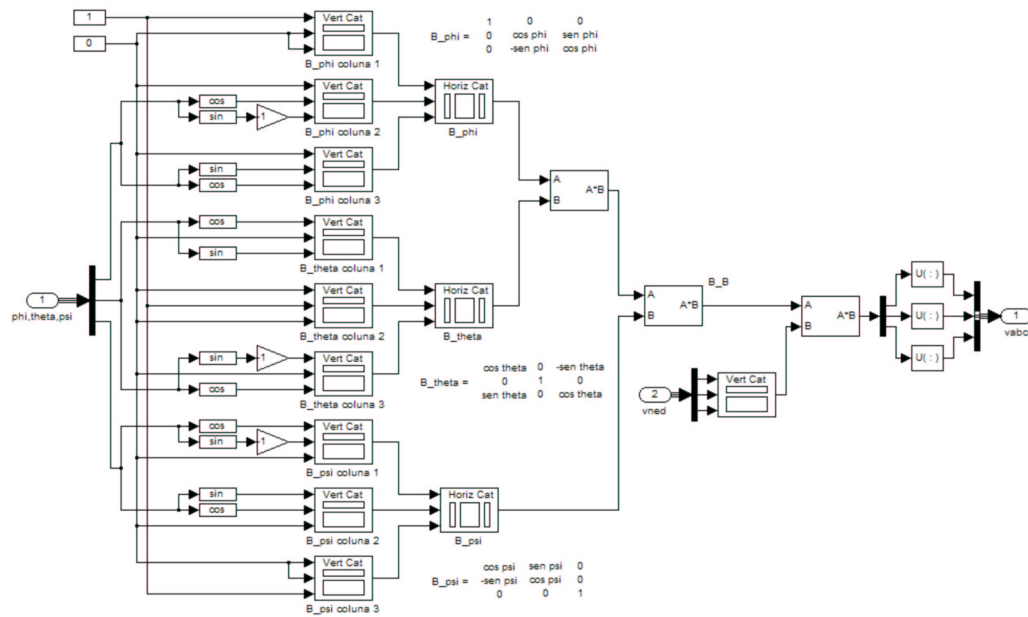


Figura 5.7: Bloco de rotação de coordenadas: NED \rightarrow ABC.

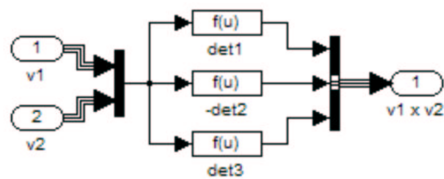


Figura 5.8: Bloco para cálculo de produto vetorial.

O bloco seguinte se trata do responsável pela determinação das variáveis representadas no eixo do vento (Figura 5.9). Para isso, utiliza-se a equação (2.38) com o intuito de transformar o vetor velocidade linear \mathbf{v}_{ABC} em \mathbf{x}_{VENTO} .

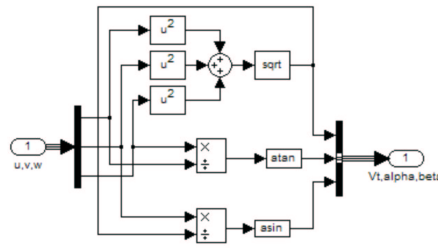


Figura 5.9: Bloco transformação de estados $[u \ v \ w]^T \rightarrow [V_T \ \alpha \ \beta]^T$.

Finalmente, ilustra-se o bloco responsável pela determinação do vetor posição \mathbf{p}_{NED} da aeronave na Figura 5.10. Para isso, utiliza-se um bloco de rotação de coordenadas $\text{ABC} \rightarrow \text{NED}$, representado na Figura 5.11.

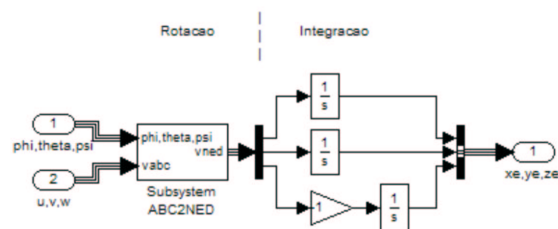


Figura 5.10: Bloco para determinação do vetor posição da aeronave via rotação de coordenadas $\text{ABC} \rightarrow \text{NED}$.

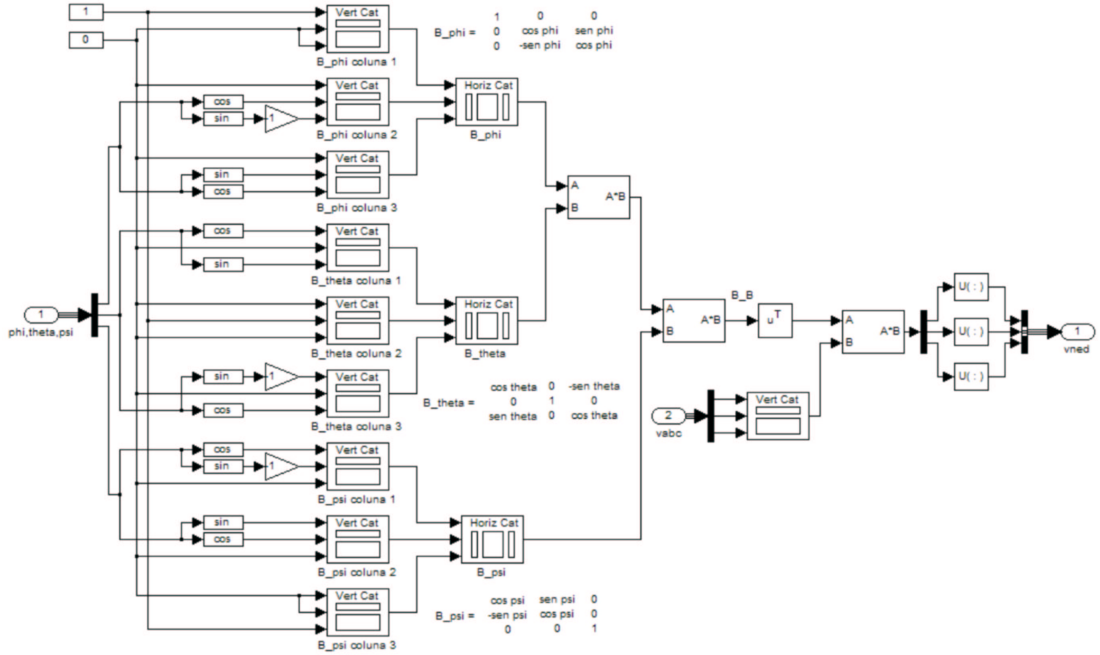


Figura 5.11: Bloco de rotação de coordenadas: ABC \rightarrow NED.

Os blocos representados nas Figuras 5.5, 5.6, 5.9 e 5.10 compõem o modelo cinemático de corpo rígido utilizado pela aplicação para reconstrução de trajetória e estimação de parâmetros. Esse modelo implementa o cenário descrito pelas equações (5.1) e (5.2):

$$\mathbf{u} = [p^m, q^m, r^m, a_x^m, a_y^m, a_z^m, g^m]^T \quad (5.1)$$

$$\mathbf{x} = [u, v, w, \phi, \theta, \psi, x_e, y_e, z_e]^T \quad (5.2)$$

A utilização em conjunto desses blocos dá origem ao bloco “Modelo Cinemático”, representado na Figura 5.12.

Finalmente, deve-se definir um cenário de estimação. Esse cenário deve prever a inserção dos parâmetros ganho e polarização, conforme descrito pela equação 5.3, também reescrito aqui por conveniência:

$$\theta = \begin{bmatrix} k_p, \Delta p, k_q, \Delta q, k_r, \Delta r, k_{a_x}, \Delta a_x, k_{a_y}, \Delta a_y, k_{a_z}, \Delta a_z, \\ k_g, \Delta g, k_u, \Delta u, k_v, \Delta v, k_w, \Delta w, k_\phi, \Delta \phi, k_\theta, \Delta \theta, k_\psi, \Delta \psi, \\ k_{x_e}, \Delta x_e, k_{y_e}, \Delta y_e, k_{z_e}, \Delta z_e, k_{v_T}, \Delta v_T, k_\alpha, \Delta \alpha, k_\beta, \Delta \beta \end{bmatrix}^T \quad (5.3)$$

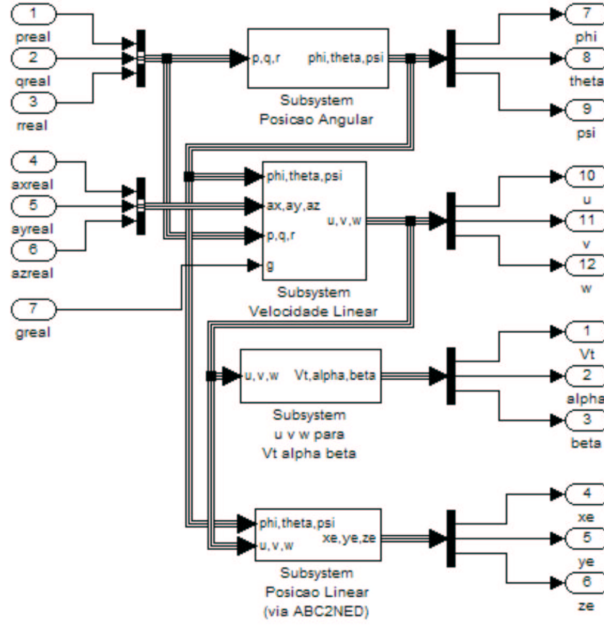


Figura 5.12: Bloco de simulação do modelo cinemático de corpo rígido.

tal que

$$\begin{aligned}
 p^m &= k_p p + \Delta p; & q^m &= k_q q + \Delta q; & r^m &= k_r r + \Delta r; \\
 a_x^m &= k_{a_x} a_x + \Delta a_x; & a_y^m &= k_{a_y} a_y + \Delta a_y; & a_z^m &= k_{a_z} a_z + \Delta a_z; \\
 u^m &= k_u u + \Delta u; & v^m &= k_v v + \Delta v; & w^m &= k_w w + \Delta w; \\
 \phi^m &= k_\phi \phi + \Delta \phi; & \theta^m &= k_\theta \theta + \Delta \theta; & \psi^m &= k_\psi \psi + \Delta \psi; \\
 x_e^m &= k_{x_e} x_e + \Delta x_e; & y_e^m &= k_{y_e} y_e + \Delta y_e; & z_e^m &= k_{z_e} z_e + \Delta z_e; \\
 p^m &= k_p p + \Delta p; & q^m &= k_q q + \Delta q; & r^m &= k_r r + \Delta r; \\
 v_T^m &= k_{v_T} v_T + \Delta v_T; & \alpha^m &= k_\alpha \alpha + \Delta \alpha; & \beta^m &= k_\beta \beta + \Delta \beta; \\
 g^m &= k_g g + \Delta g;
 \end{aligned} \tag{5.4}$$

Esse cenário é descrito na Figura 5.13. Importante observar a existência de blocos de transformação “medição” → “valor real” para as variáveis de entrada e de “valor real” → “medição” para as variáveis de saída. A evolução dos algoritmos de otimização se faz exatamente com a alteração dos valores dos parâmetros (representados pelo vetor de parâmetros estimados “vparest”) e a observação das variáveis de saída.

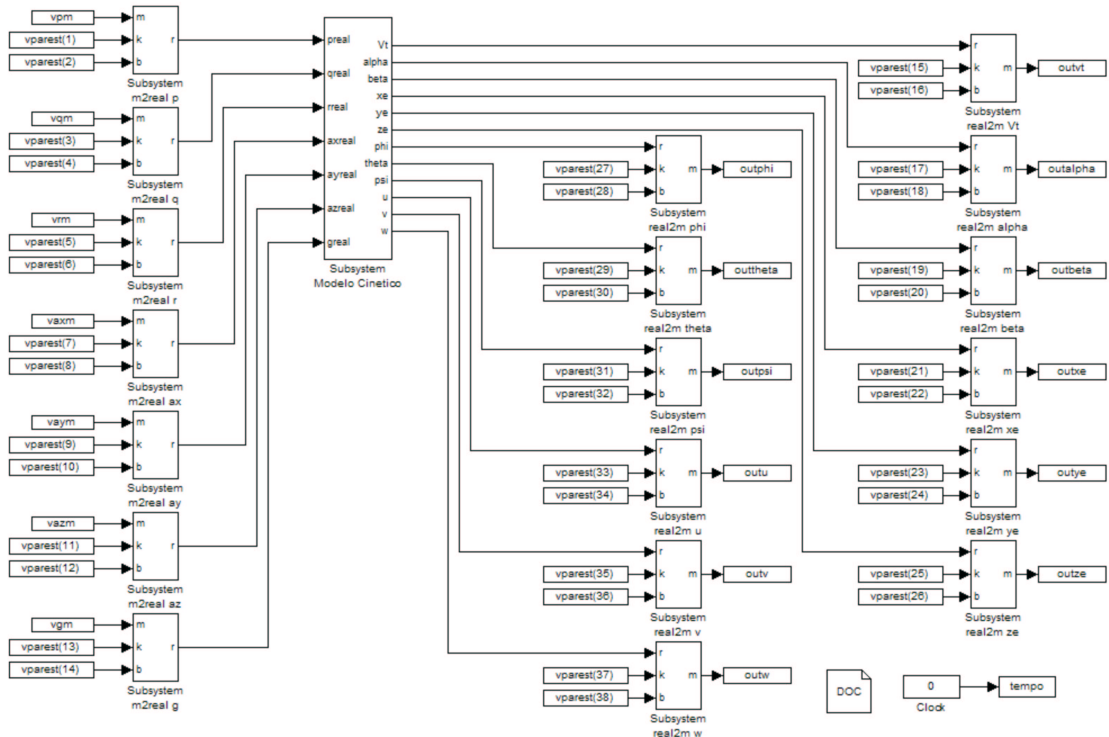


Figura 5.13: Ambiente de simulação para reconstrução de trajetória e estimação de parâmetros.

5.6 Interface Gráfica

A interface gráfica consiste em um conjunto de janelas responsáveis pela disponibilização das funcionalidades da aplicação para o usuário. O objetivo final da sua implementação consiste em conciliar o amplo acesso às diversas funcionalidades implementadas com a simplicidade de operação.

A janela principal consiste em quadros de indicação e botões de navegação. Os quadros de indicação são responsáveis por dar orientação ao usuário quanto às suas seleções atuais. Especificamente, disponibilizam-se um quadro de indicação do algoritmo selecionado e outro relativo ao arquivo de dados a ser utilizado. A janela principal é ilustrada na Figura 5.14.

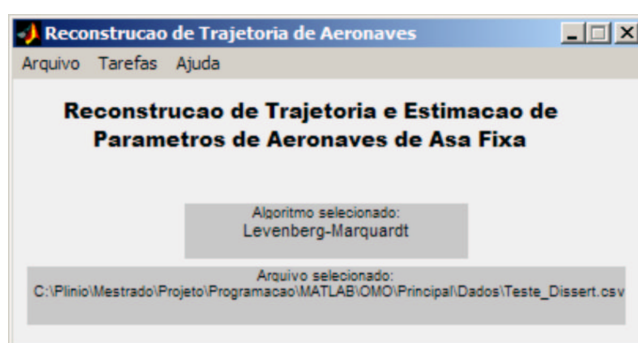


Figura 5.14: Interface gráfica: janela principal.

Os botões de navegação têm por função oferecer ferramentas para configuração do experimento a ser realizado. O primeiro passo para isso consiste na determinação do arquivo de dados a ser utilizado⁴. A Figura 5.15 ilustra esse procedimento.

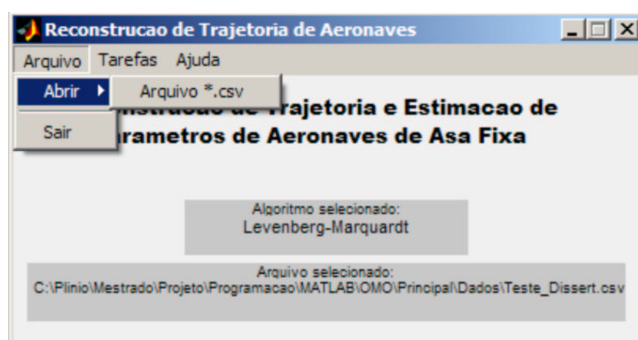


Figura 5.15: Interface gráfica: seleção do arquivo de dados.

Em seguida, abre-se uma janela de busca pelo arquivo de dados do próprio sistema operacional (Figura 5.16). Com o fim desse procedimento, o arquivo selecionado é indicado no quadro inferior da janela principal.

⁴Para maiores informações a respeito da organização e padronização do arquivo de dados, deve-se observar o campo de ajuda da função “Modelos\Geral\pcriaestados.m” (Seção 5.5).

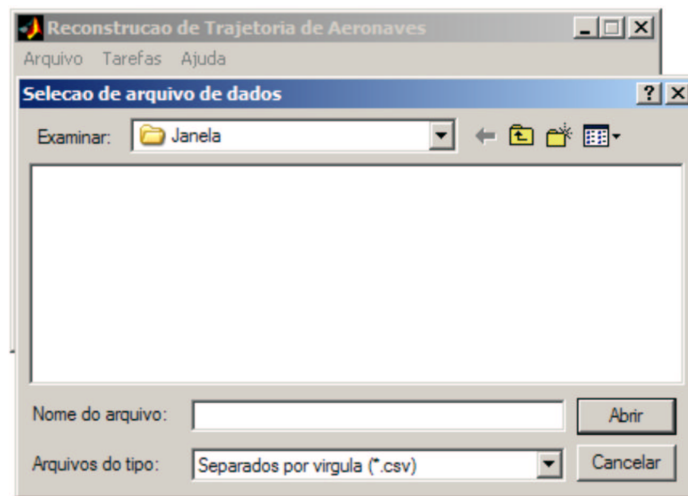


Figura 5.16: Interface gráfica: busca pelo arquivo de dados.

O segundo passo consiste na seleção do algoritmo desejado. Com o fim desse procedimento, o nome do algoritmo selecionado é indicado no quadro superior da janela principal. A Figura 5.17 ilustra esse procedimento.

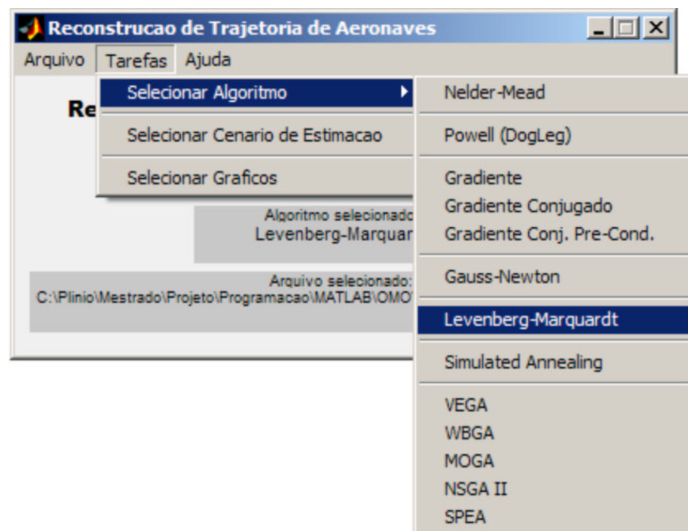


Figura 5.17: Interface gráfica: seleção do algoritmo.

O próximo passo consiste na seleção do cenário de estimação. Esse cenário é representado pelos parâmetros a serem estimados e as variáveis observadas para esse fim, conforme ilustra a Figura 5.18. Uma vez que todos os passos

tenham sido cumpridos, basta pressionar o botão “Iniciar” e o algoritmo inicia o processo de estimação / reconstrução de trajetória.



Figura 5.18: Interface gráfica: seleção do cenário de estimação.

A aplicação também disponibiliza uma ferramenta de análise dos sinais de entrada e saída do modelo via gráficos de tendência. Para isso, basta selecionar a opção “Selecionar Gráficos” em “Tarefas”, conforme ilustra a Figura 5.19.

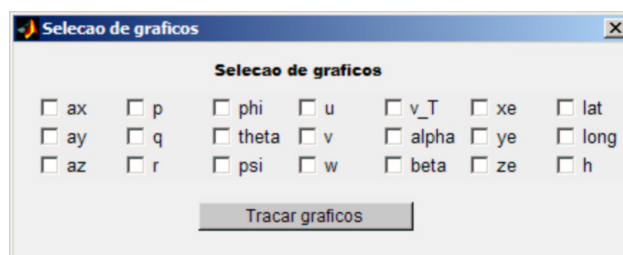


Figura 5.19: Interface gráfica: seleção das variáveis para criação de gráficos de tendência.

RESULTADOS E DISCUSSÃO

Os resultados obtidos neste trabalho e a discussão a seu respeito são apresentados neste capítulo. O motivo para a criação de um capítulo específico para esse fim é simples. Boa parte das análises é feita de forma mais conveniente se aplicada a grupos de situações. Não se vêem aspectos positivos na distribuição dessas análises em capítulos e seções distintas. De fato, esse procedimento levaria à criação de descrições eventualmente muito parecidas e desvincularia (pela disposição no texto) resultados semelhantes.

Devido a isso, este capítulo apresenta os resultados obtidos e a discussão a seu respeito segundo dois aspectos principais: a qualidade da reconstrução (Seção 6.1) e o tempo de execução (Seção 6.2). Em ambos os casos, as análises são feitas de maneira a criar grupos de algoritmos com desempenho semelhante na resolução do problema proposto.

6.1 Qualidade da reconstrução

Nesta seção, os algoritmos são analisados segundo a qualidade da reconstrução obtida com a sua utilização na resolução do problema de reconstrução de trajetória de aeronaves. Para isso, criam-se cinco grupos de algoritmos com desempenho semelhante:

1. Nelder-Mead, Powell e *Simulated Annealing* (item 6.1.1);
2. Gradiente e Gradiente Conjugado (item 6.1.2);
3. Gradiente Conjugado Pré-Condicionado e Gauss-Newton (item 6.1.3);
4. Levenberg-Marquardt (item 6.1.4);
5. Algoritmos genéticos (item 6.1.5).

6.1.1 Nelder-Mead, Powell e *Simulated Annealing*

A análise dos resultados obtidos com a utilização do algoritmo de Nelder-Mead (NM) mostra pequena sensibilidade tanto no que diz respeito à manobra utilizada quanto ao nível de contaminação dos sinais de entrada por ruído. Apesar de haver alguma tendência à diminuição de desempenho com o aumento do índice da manobra e com o aumento do nível de ruído, os resultados obtidos com o índice $I_{4,\%}$ não são decisivos a esse respeito.

Um dos motivos para esse comportamento pode estar vinculado ao fato de o processo de otimização ser baseado na interpretação de resultados particulares (instantâneos) e em tomadas de decisão que dependem da configuração atual da matriz simplex. Com isso, o processo depende mais das configurações iniciais (e correntes) da matriz simplex do que propriamente de informações a respeito da dinâmica do sistema.

Tabela 6.1: Índices de desempenho para o algoritmo de NM, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NM,C1,0\sigma}$	$\hat{\theta}_{NM,C1,1\sigma}$	$\hat{\theta}_{NM,C1,2\sigma}$	$\hat{\theta}_{NM,C1,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{NM,C1,0\sigma,\%}$	$\hat{\theta}_{NM,C1,1\sigma,\%}$	$\hat{\theta}_{NM,C1,2\sigma,\%}$	$\hat{\theta}_{NM,C1,4\sigma,\%}$
I_2	30,0%	20,0%	30,0%	20,0%
I_3	168,54	172,10	175,90	179,10

Detalhamento da Tabela 6.1:

$$\left\{ \begin{array}{l} \hat{\theta}_{NM,C1,0\sigma} = [-0,04138; 0,16552; 0,08276; 0,00690; -0,00414; -0,01379]^T \\ \hat{\theta}_{NM,C1,0\sigma,\%} = [3,448; 13,793; 6,897; 17,241; 10,345; 34,483]^T \\ \hat{\theta}_{NM,C1,1\sigma} = [0,00000; 0,16552; 0,04138; 0,00690; -0,00552; -0,00138]^T \\ \hat{\theta}_{NM,C1,1\sigma,\%} = [0,000; 13,793; 3,448; 17,241; 13,793; 3,448]^T \\ \hat{\theta}_{NM,C1,2\sigma} = [-0,53793; 0,16552; 0,20690; -0,01517; 0,00690; -0,00552]^T \\ \hat{\theta}_{NM,C1,2\sigma,\%} = [44,828; 13,793; 17,241; 37,931; 17,241; 13,793]^T \\ \hat{\theta}_{NM,C1,4\sigma} = [-0,37241; 0,16552; 0,08276; -0,00966; 0,00966; 0,00000]^T \\ \hat{\theta}_{NM,C1,4\sigma,\%} = [31,034; 13,793; 6,897; 24,138; 24,138; 0,000]^T \end{array} \right.$$

Tabela 6.2: Índices de desempenho para o algoritmo de NM, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NM, C2, 0\sigma}$	$\hat{\theta}_{NM, C2, 1\sigma}$	$\hat{\theta}_{NM, C2, 2\sigma}$	$\hat{\theta}_{NM, C2, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{NM, C2, 0\sigma, \%}$	$\hat{\theta}_{NM, C2, 1\sigma, \%}$	$\hat{\theta}_{NM, C2, 2\sigma, \%}$	$\hat{\theta}_{NM, C2, 4\sigma, \%}$
I_2	35,0%	30,0%	20,0%	25,0%
I_3	175,62	179,66	176,78	175,12

Detalhamento da Tabela 6.2:

$$\left\{ \begin{array}{l} \hat{\theta}_{NM, C2, 0\sigma} = [-0,24828; 0,08276; -0,16552; 0,00276; 0,00138; -0,00828]^T \\ \hat{\theta}_{NM, C2, 0\sigma, \%} = [20,690; 6,897; 13,793; 6,897; 3,448; 20,690]^T \\ \hat{\theta}_{NM, C2, 1\sigma} = [-0,28966; 0,16552; -0,12414; 0,00138; -0,00138; -0,01379]^T \\ \hat{\theta}_{NM, C2, 1\sigma, \%} = [24,138; 13,793; 10,345; 3,448; 3,448; 34,483]^T \\ \hat{\theta}_{NM, C2, 2\sigma} = [-0,16552; 0,16552; 0,08276; 0,01103; -0,00276; -0,01241]^T \\ \hat{\theta}_{NM, C2, 2\sigma, \%} = [13,793; 13,793; 6,897; 27,586; 6,897; 31,034]^T \\ \hat{\theta}_{NM, C2, 4\sigma} = [-0,20690; 0,16552; -0,12414; 0,00690; -0,00138; 0,00000]^T \\ \hat{\theta}_{NM, C2, 4\sigma, \%} = [17,241; 13,793; 10,345; 17,241; 3,448; 0,000]^T \end{array} \right.$$

Tabela 6.3: Índices de desempenho para o algoritmo de NM, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NM, C3, 0\sigma}$	$\hat{\theta}_{NM, C3, 1\sigma}$	$\hat{\theta}_{NM, C3, 2\sigma}$	$\hat{\theta}_{NM, C3, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{NM, C3, 0\sigma, \%}$	$\hat{\theta}_{NM, C3, 1\sigma, \%}$	$\hat{\theta}_{NM, C3, 2\sigma, \%}$	$\hat{\theta}_{NM, C3, 4\sigma, \%}$
I_2	35,0%	50,0%	45,0%	35,0%
I_3	189,98	188,70	191,52	193,22

Detalhamento da Tabela 6.3:

$$\left\{ \begin{array}{l} \hat{\theta}_{NM, C3, 0\sigma} = [-0,12414; -0,12414; -0,33103; 0,00276; -0,01103; -0,00828]^T \\ \hat{\theta}_{NM, C3, 0\sigma, \%} = [10,345; 10,345; 27,586; 6,897; 27,586; 20,690]^T \\ \hat{\theta}_{NM, C3, 1\sigma} = [-0,08276; -0,12414; -0,45517; -0,01379; 0,00276; -0,01103]^T \\ \hat{\theta}_{NM, C3, 1\sigma, \%} = [6,897; 10,345; 37,931; 34,483; 6,897; 27,586]^T \\ \hat{\theta}_{NM, C3, 2\sigma} = [-0,08276; 0,04138; -0,04138; -0,00414; -0,00966; -0,00966]^T \\ \hat{\theta}_{NM, C3, 2\sigma, \%} = [6,897; 3,448; 3,448; 10,345; 24,138; 24,138]^T \\ \hat{\theta}_{NM, C3, 4\sigma} = [0,00000; 0,00000; -0,24828; -0,01379; -0,01103; -0,01379]^T \\ \hat{\theta}_{NM, C3, 4\sigma, \%} = [0,000; 0,000; 20,690; 34,483; 27,586; 34,483]^T \end{array} \right.$$

Tabela 6.4: Índices de desempenho para o algoritmo de NM, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NM, C4, 0\sigma}$	$\hat{\theta}_{NM, C4, 1\sigma}$	$\hat{\theta}_{NM, C4, 2\sigma}$	$\hat{\theta}_{NM, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{NM, C4, 0\sigma, \%}$	$\hat{\theta}_{NM, C4, 1\sigma, \%}$	$\hat{\theta}_{NM, C4, 2\sigma, \%}$	$\hat{\theta}_{NM, C4, 4\sigma, \%}$
I_2	40,0%	45,0%	25,0%	40,0%
I_3	199,60	195,82	199,46	201,40

Detalhamento da Tabela 6.4:

$$\left\{ \begin{array}{l} \hat{\theta}_{NM, C4, 0\sigma} = [0,33103; 0,00000; 0,04138; 0,00138; 0,00138; -0,00276]^T \\ \hat{\theta}_{NM, C4, 0\sigma, \%} = [27,586; 0,000; 3,448; 3,448; 3,448; 6,897]^T \\ \hat{\theta}_{NM, C4, 1\sigma} = [0,04138; 0,12414; -0,12414; 0,00828; -0,00690; -0,00276]^T \\ \hat{\theta}_{NM, C4, 1\sigma, \%} = [3,448; 10,345; 10,345; 20,690; 17,241; 6,897]^T \\ \hat{\theta}_{NM, C4, 2\sigma} = [0,20690; 0,00000; -0,45517; -0,00828; -0,00690; -0,00828]^T \\ \hat{\theta}_{NM, C4, 2\sigma, \%} = [17,241; 0,000; 37,931; 20,690; 17,241; 20,690]^T \\ \hat{\theta}_{NM, C4, 4\sigma} = [0,04138; 0,00000; -0,45517; -0,00966; 0,00276; -0,01931]^T \\ \hat{\theta}_{NM, C4, 4\sigma, \%} = [3,448; 0,000; 37,931; 24,138; 6,897; 48,276]^T \end{array} \right.$$

Tabela 6.5: Índices de desempenho para o algoritmo de NM, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NM, C5, 0\sigma}$	$\hat{\theta}_{NM, C5, 1\sigma}$	$\hat{\theta}_{NM, C5, 2\sigma}$	$\hat{\theta}_{NM, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{NM, C5, 0\sigma, \%}$	$\hat{\theta}_{NM, C5, 1\sigma, \%}$	$\hat{\theta}_{NM, C5, 2\sigma, \%}$	$\hat{\theta}_{NM, C5, 4\sigma, \%}$
I_2	80,0%	50,0%	65,0%	50,0%
I_3	219,02	224,82	217,60	223,18

Detalhamento da Tabela 6.5:

$$\left\{ \begin{array}{l} \hat{\theta}_{NM, C5, 0\sigma} = [0,00000; 0,08276; -0,49655; 0,00828; 0,00138; -0,00966; \\ \quad 0,99655; 0,99931; 1,00552; 1,00069; 1,00276; 0,99586]^T \\ \hat{\theta}_{NM, C5, 0\sigma, \%} = [0,000; 6,897; 41,379; 20,690; 3,448; 24,138; \\ \quad 17,241; 3,448; 27,586; 3,448; 13,793; 20,690]^T \\ \hat{\theta}_{NM, C5, 1\sigma} = [0,12414; 0,04138; -0,20690; -0,00552; -0,00276; -0,00414; \\ \quad 0,99793; 0,99655; 1,00207; 0,99931; 0,99310; 1,00345]^T \\ \hat{\theta}_{NM, C5, 1\sigma, \%} = [10,345; 3,448; 17,241; 13,793; 6,897; 10,345; \\ \quad 10,345; 17,241; 10,345; 3,448; 34,483; 17,241]^T \\ \hat{\theta}_{NM, C5, 2\sigma} = [0,04138; 0,00000; -0,53793; 0,00000; 0,00552; -0,00552; \\ \quad 0,99379; 1,00138; 0,99586; 0,99310; 0,99862; 1,00207]^T \\ \hat{\theta}_{NM, C5, 2\sigma, \%} = [3,448; 0,000; 44,828; 0,000; 13,793; 13,793; \\ \quad 31,034; 6,897; 20,690; 34,483; 6,897; 10,345]^T \\ \hat{\theta}_{NM, C5, 4\sigma} = [0,37241; 0,08276; -0,37241; -0,01103; -0,00690; -0,01793; \\ \quad 0,99793; 0,99793; 1,00138; 1,00000; 1,00069; 0,99517]^T \\ \hat{\theta}_{NM, C5, 4\sigma, \%} = [31,034; 6,897; 31,034; 27,586; 17,241; 44,828; \\ \quad 10,345; 10,345; 6,897; 0,000; 3,448; 24,138]^T \end{array} \right.$$

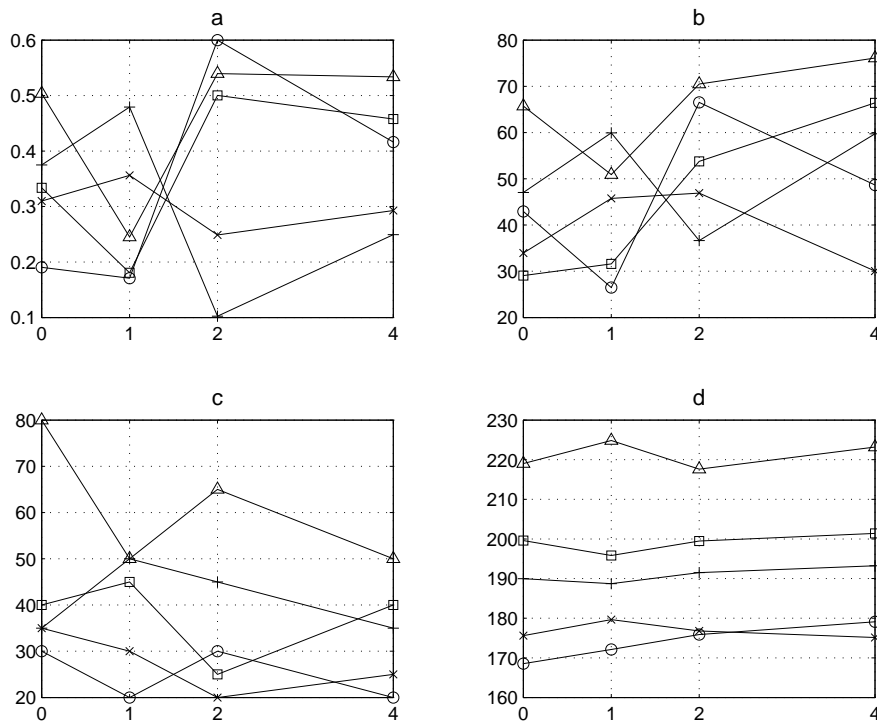


Figura 6.1: Comparação do desempenho do algoritmo de NM para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: x; 3: +; 4: \square e 5: \triangle .

Assim como o algoritmo de Levenberg-Marquardt, o algoritmo de Powell (DL) concilia a utilização dos métodos do gradiente e de Gauss-Newton. Contudo essa utilização é baseada na localização do *cauchy point* e do ponto de Gauss-Newton em relação à região de confiança.

Com isso, o algoritmo tende a apresentar comportamento dependente da posição desses pontos e do valor do raio da região de confiança. O fato de utilizar tomadas de decisão fortemente dependentes do ponto estimado atual (assim como no algoritmo de Nelder-Mead) pode-se tratar de um dos motivos pelos quais o algoritmo apresenta pouca sensibilidade tanto a variações do nível de contaminação dos sinais de entrada por ruído quanto da manobra utilizada.

Tabela 6.6: Índices de desempenho para o algoritmo de DL, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{DL, C1, 0\sigma}$	$\hat{\theta}_{DL, C1, 1\sigma}$	$\hat{\theta}_{DL, C1, 2\sigma}$	$\hat{\theta}_{DL, C1, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{DL, C1, 0\sigma, \%}$	$\hat{\theta}_{DL, C1, 1\sigma, \%}$	$\hat{\theta}_{DL, C1, 2\sigma, \%}$	$\hat{\theta}_{DL, C1, 4\sigma, \%}$
I_2	55,0%	55,0%	45,0%	40,0%
I_3	134,96	130,20	130,76	130,90

Detalhamento da Tabela 6.6:

$$\left\{ \begin{array}{l} \hat{\theta}_{DL, C1, 0\sigma} = [0,34759; -0,24828; 0,04966; 0,00331; -0,01655; -0,00497]^T \\ \hat{\theta}_{DL, C1, 0\sigma, \%} = [28,966; 20,689; 4,138; 8,276; 41,380; 12,414]^T \\ \hat{\theta}_{DL, C1, 1\sigma} = [-0,37241; 0,57931; 0,20690; -0,00414; 0,01103; -0,01655]^T \\ \hat{\theta}_{DL, C1, 1\sigma, \%} = [31,034; 48,276; 17,241; 10,345; 27,586; 41,379]^T \\ \hat{\theta}_{DL, C1, 2\sigma} = [-0,49242; -0,21104; 0,28138; -0,00469; -0,00704; -0,00235]^T \\ \hat{\theta}_{DL, C1, 2\sigma, \%} = [41,035; 17,587; 23,448; 11,725; 17,587; 5,862]^T \\ \hat{\theta}_{DL, C1, 4\sigma} = [-0,29793; 0,19862; -0,29793; -0,01545; 0,01324; 0,00993]^T \\ \hat{\theta}_{DL, C1, 4\sigma, \%} = [24,827; 16,552; 24,827; 38,621; 33,103; 24,827]^T \end{array} \right.$$

Tabela 6.7: Índices de desempenho para o algoritmo de DL, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{DL,C2,0\sigma}$	$\hat{\theta}_{DL,C2,1\sigma}$	$\hat{\theta}_{DL,C2,2\sigma}$	$\hat{\theta}_{DL,C2,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{DL,C2,0\sigma,\%}$	$\hat{\theta}_{DL,C2,1\sigma,\%}$	$\hat{\theta}_{DL,C2,2\sigma,\%}$	$\hat{\theta}_{DL,C2,4\sigma,\%}$
I_2	70,0%	60,0%	45,0%	45,0%
I_3	130,76	140,98	131,74	131,04

Detalhamento da Tabela 6.7:

$$\left\{ \begin{array}{l} \hat{\theta}_{DL,C2,0\sigma} = [-0,08276; 0,37241; -0,37241; -0,01103; -0,01793; 0,01655]^T \\ \hat{\theta}_{DL,C2,0\sigma,\%} = [6,897; 31,034; 31,034; 27,586; 44,828; 41,379]^T \\ \hat{\theta}_{DL,C2,1\sigma} = [-0,20690; 0,33103; 0,37241; -0,01379; -0,00690; 0,00690]^T \\ \hat{\theta}_{DL,C2,1\sigma,\%} = [17,241; 27,586; 31,034; 34,483; 17,241; 17,241]^T \\ \hat{\theta}_{DL,C2,2\sigma} = [-0,20690; 0,41379; -0,08276; -0,01931; -0,00276; 0,00828]^T \\ \hat{\theta}_{DL,C2,2\sigma,\%} = [17,241; 34,483; 6,897; 48,276; 6,897; 20,690]^T \\ \hat{\theta}_{DL,C2,4\sigma} = [-0,04138; 0,20690; -0,53793; 0,01655; -0,00828; 0,01655]^T \\ \hat{\theta}_{DL,C2,4\sigma,\%} = [3,448; 17,241; 44,828; 41,379; 20,690; 41,379]^T \end{array} \right.$$

Tabela 6.8: Índices de desempenho para o algoritmo de DL, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{DL,C3,0\sigma}$	$\hat{\theta}_{DL,C3,1\sigma}$	$\hat{\theta}_{DL,C3,2\sigma}$	$\hat{\theta}_{DL,C3,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{DL,C3,0\sigma,\%}$	$\hat{\theta}_{DL,C3,1\sigma,\%}$	$\hat{\theta}_{DL,C3,2\sigma,\%}$	$\hat{\theta}_{DL,C3,4\sigma,\%}$
I_2	60,0%	50,0%	50,0%	65,0%
I_3	128,80	124,60	124,04	135,10

Detalhamento da Tabela 6.8:

$$\left\{ \begin{array}{l} \hat{\theta}_{DL,C3,0\sigma} = [-0,12414; 0,12414; -0,08276; -0,01241; -0,01793; -0,01655]^T \\ \hat{\theta}_{DL,C3,0\sigma,\%} = [10,345; 10,345; 6,897; 31,034; 44,828; 41,379]^T \\ \hat{\theta}_{DL,C3,1\sigma} = [-0,12414; -0,04138; -0,49655; -0,01655; 0,01517; -0,01241]^T \\ \hat{\theta}_{DL,C3,1\sigma,\%} = [10,345; 3,448; 41,379; 41,379; 37,931; 31,034]^T \\ \hat{\theta}_{DL,C3,2\sigma} = [-0,20690; 0,28966; -0,53793; -0,01241; 0,01103; 0,00000]^T \\ \hat{\theta}_{DL,C3,2\sigma,\%} = [17,241; 24,138; 44,828; 31,034; 27,586; 0,000]^T \\ \hat{\theta}_{DL,C3,4\sigma} = [-0,04138; 0,53793; -0,20690; -0,01931; -0,01379; 0,00138]^T \\ \hat{\theta}_{DL,C3,4\sigma,\%} = [3,448; 44,828; 17,241; 48,276; 34,483; 3,448]^T \end{array} \right.$$

Tabela 6.9: Índices de desempenho para o algoritmo de DL, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{DL, C4, 0\sigma}$	$\hat{\theta}_{DL, C4, 1\sigma}$	$\hat{\theta}_{DL, C4, 2\sigma}$	$\hat{\theta}_{DL, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{DL, C4, 0\sigma, \%}$	$\hat{\theta}_{DL, C4, 1\sigma, \%}$	$\hat{\theta}_{DL, C4, 2\sigma, \%}$	$\hat{\theta}_{DL, C4, 4\sigma, \%}$
I_2	45,0%	35,0%	65,0%	35,0%
I_3	123,20	127,82	142,24	130,20

Detalhamento da Tabela 6.9:

$$\left\{ \begin{array}{l} \hat{\theta}_{DL, C4, 0\sigma} = [-0,20690; 0,28966; -0,24828; -0,00552; 0,01931; 0,00552]^T \\ \hat{\theta}_{DL, C4, 0\sigma, \%} = [17,241; 24,138; 20,690; 13,793; 48,276; 13,793]^T \\ \hat{\theta}_{DL, C4, 1\sigma} = [-0,28966; 0,37241; -0,16552; -0,00552; 0,00552; -0,01931]^T \\ \hat{\theta}_{DL, C4, 1\sigma, \%} = [24,138; 31,034; 13,793; 13,793; 13,793; 48,276]^T \\ \hat{\theta}_{DL, C4, 2\sigma} = [-0,20690; 0,16552; -0,49655; -0,00966; -0,00828; -0,01793]^T \\ \hat{\theta}_{DL, C4, 2\sigma, \%} = [17,241; 13,793; 41,379; 24,138; 20,690; 44,828]^T \\ \hat{\theta}_{DL, C4, 4\sigma} = [-0,16552; 0,24828; -0,04138; -0,01655; 0,01241; 0,01517]^T \\ \hat{\theta}_{DL, C4, 4\sigma, \%} = [13,793; 20,690; 3,448; 41,379; 31,034; 37,931]^T \end{array} \right.$$

Tabela 6.10: Índices de desempenho para o algoritmo de DL, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{DL, C5, 0\sigma}$	$\hat{\theta}_{DL, C5, 1\sigma}$	$\hat{\theta}_{DL, C5, 2\sigma}$	$\hat{\theta}_{DL, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{DL, C5, 0\sigma, \%}$	$\hat{\theta}_{DL, C5, 1\sigma, \%}$	$\hat{\theta}_{DL, C5, 2\sigma, \%}$	$\hat{\theta}_{DL, C5, 4\sigma, \%}$
I_2	75,0%	80,0%	75,0%	75,0%
I_3	240,50	248,56	244,66	237,12

Detalhamento da Tabela 6.10:

$$\left\{ \begin{array}{l} \hat{\theta}_{DL, C5, 0\sigma} = [-0,37241; 0,04138; -0,08276; -0,00828; 0,00138; 0,00000; \\ \quad 1,00483; 0,99241; 0,99172; 0,99586; 1,00690; 1,00759]^T \\ \hat{\theta}_{DL, C5, 0\sigma, \%} = [31,034; 3,448; 6,897; 20,690; 3,448; 0,000; \\ \quad 24,138; 37,931; 41,379; 20,690; 34,483; 37,931]^T \\ \hat{\theta}_{DL, C5, 1\sigma} = [-0,28966; 0,24828; -0,57931; 0,01241; 0,00966; -0,01517; \\ \quad 0,99793; 1,00759; 0,99379; 0,99310; 1,00897; 0,99310]^T \\ \hat{\theta}_{DL, C5, 1\sigma, \%} = [24,138; 20,690; 48,276; 31,034; 24,138; 37,931; \\ \quad 10,345; 37,931; 31,034; 34,483; 44,828; 34,483]^T \\ \hat{\theta}_{DL, C5, 2\sigma} = [-0,24828; 0,28966; -0,37241; -0,00690; -0,01793; -0,00966; \\ \quad 1,00621; 1,00276; 0,99379; 0,99793; 0,99310; 0,99655]^T \\ \hat{\theta}_{DL, C5, 2\sigma, \%} = [20,690; 24,138; 31,034; 17,241; 44,828; 24,138; \\ \quad 31,034; 13,793; 31,034; 10,345; 34,483; 17,241]^T \\ \hat{\theta}_{DL, C5, 4\sigma} = [-0,08276; -0,20690; 0,00000; -0,00414; -0,01793; 0,00690; \\ \quad 0,99655; 1,00621; 0,99655; 1,00000; 0,99310; 0,99379]^T \\ \hat{\theta}_{DL, C5, 4\sigma, \%} = [6,897; 17,241; 0,000; 10,345; 44,828; 17,241; \\ \quad 17,241; 31,034; 17,241; 0,000; 34,483; 31,034]^T \end{array} \right.$$

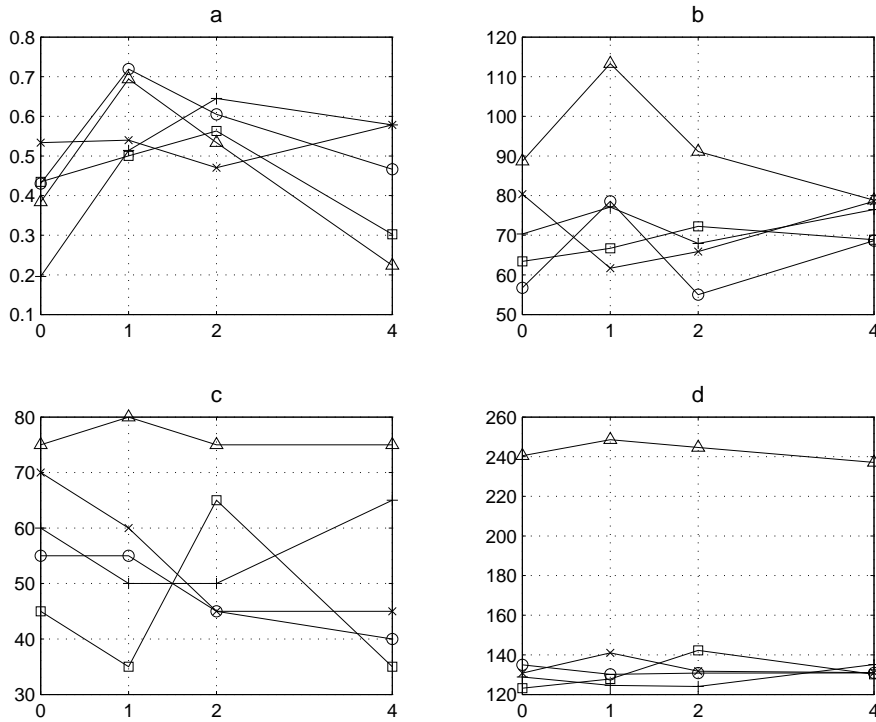


Figura 6.2: Comparação do desempenho do algoritmo de DL para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: x; 3: +; 4: \square e 5: \triangle .

O único algoritmo estocástico não classificado como genético consiste no Simulated Annealing (SA). Os resultados obtidos com esse algoritmo dão origem à mesma análise feita para os métodos de Nelder-Mead e de Powell. Deve-se observar, também para o algoritmo de Simulated Annealing, a existência de critérios de tomada de decisão fortemente influenciados pelo estágio da estimação. Nesse caso a decisão de transição entre estados depende basicamente da comparação entre o valor da função no ponto atual e no ponto avaliado e, de forma indireta, do valor do parâmetro T (temperatura).

Tabela 6.11: Índices de desempenho para o algoritmo de SA, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SA, C1, 0\sigma}$	$\hat{\theta}_{SA, C1, 1\sigma}$	$\hat{\theta}_{SA, C1, 2\sigma}$	$\hat{\theta}_{SA, C1, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{SA, C1, 0\sigma, \%}$	$\hat{\theta}_{SA, C1, 1\sigma, \%}$	$\hat{\theta}_{SA, C1, 2\sigma, \%}$	$\hat{\theta}_{SA, C1, 4\sigma, \%}$
I_2	55,0%	0,0%	0,0%	0,0%
I_3	401,00	401,00	401,00	401,00

Detalhamento da Tabela 6.11:

$$\left\{ \begin{array}{l} \hat{\theta}_{SA, C1, 0\sigma} = [-0,57931; -0,08276; 0,24828; -0,01931; 0,02000; -0,01931]^T \\ \hat{\theta}_{SA, C1, 0\sigma, \%} = [48,276; 6,897; 20,690; 48,276; 50,000; 48,276]^T \\ \hat{\theta}_{SA, C1, 1\sigma} = [-0,57931; -0,08276; 0,24828; -0,01931; -0,01931; 0,02000]^T \\ \hat{\theta}_{SA, C1, 1\sigma, \%} = [48,276; 6,897; 20,690; 48,276; 48,276; 50,000]^T \\ \hat{\theta}_{SA, C1, 2\sigma} = [-0,57931; -0,08276; 0,24828; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA, C1, 2\sigma, \%} = [48,276; 6,897; 20,690; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{SA, C1, 4\sigma} = [-0,57931; -0,08276; 0,24828; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA, C1, 4\sigma, \%} = [48,276; 6,897; 20,690; 48,276; 48,276; 48,276]^T \end{array} \right.$$

Tabela 6.12: Índices de desempenho para o algoritmo de SA, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SA,C2,0\sigma}$	$\hat{\theta}_{SA,C2,1\sigma}$	$\hat{\theta}_{SA,C2,2\sigma}$	$\hat{\theta}_{SA,C2,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{SA,C2,0\sigma,\%}$	$\hat{\theta}_{SA,C2,1\sigma,\%}$	$\hat{\theta}_{SA,C2,2\sigma,\%}$	$\hat{\theta}_{SA,C2,4\sigma,\%}$
I_2	60,0%	0,0%	70,0%	70,0%
I_3	401,00	401,00	401,00	401,00

Detalhamento da Tabela 6.12:

$$\left\{ \begin{array}{l} \hat{\theta}_{SA,C2,0\sigma} = [-0,20690; 0,08276; -0,04138; -0,01931; 0,02000; -0,01931]^T \\ \hat{\theta}_{SA,C2,0\sigma,\%} = [17,241; 6,897; 3,448; 48,276; 50,000; 48,276]^T \\ \hat{\theta}_{SA,C2,1\sigma} = [0,00000; 0,16552; 0,41379; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA,C2,1\sigma,\%} = [0,000; 13,793; 34,483; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{SA,C2,2\sigma} = [-0,20690; 0,16552; -0,08276; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA,C2,2\sigma,\%} = [17,241; 13,793; 6,897; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{SA,C2,4\sigma} = [-0,41379; 0,12414; 0,00000; -0,01931; 0,02000; -0,01931]^T \\ \hat{\theta}_{SA,C2,4\sigma,\%} = [34,483; 10,345; 0,000; 48,276; 50,000; 48,276]^T \end{array} \right.$$

Tabela 6.13: Índices de desempenho para o algoritmo de SA, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SA,C3,0\sigma}$	$\hat{\theta}_{SA,C3,1\sigma}$	$\hat{\theta}_{SA,C3,2\sigma}$	$\hat{\theta}_{SA,C3,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{SA,C3,0\sigma,\%}$	$\hat{\theta}_{SA,C3,1\sigma,\%}$	$\hat{\theta}_{SA,C3,2\sigma,\%}$	$\hat{\theta}_{SA,C3,4\sigma,\%}$
I_2	80,0%	85,0%	60,0%	0,0%
I_3	401,00	401,00	401,00	401,00

Detalhamento da Tabela 6.13:

$$\left\{ \begin{array}{l} \hat{\theta}_{SA,C3,0\sigma} = [0,28966; 0,45517; -0,53793; -0,01931; 0,02000; -0,01931]^T \\ \hat{\theta}_{SA,C3,0\sigma,\%} = [24,138; 37,931; 44,828; 48,276; 50,000; 48,276]^T \\ \hat{\theta}_{SA,C3,1\sigma} = [0,41379; 0,49655; -0,57931; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA,C3,1\sigma,\%} = [34,483; 41,379; 48,276; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{SA,C3,2\sigma} = [0,49655; 0,04138; -0,57931; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA,C3,2\sigma,\%} = [41,379; 3,448; 48,276; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{SA,C3,4\sigma} = [0,28966; 0,45517; -0,57931; 0,02000; 0,02000; -0,01931]^T \\ \hat{\theta}_{SA,C3,4\sigma,\%} = [24,138; 37,931; 48,276; 50,000; 50,000; 48,276]^T \end{array} \right.$$

Tabela 6.14: Índices de desempenho para o algoritmo de SA, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SA, C4, 0\sigma}$	$\hat{\theta}_{SA, C4, 1\sigma}$	$\hat{\theta}_{SA, C4, 2\sigma}$	$\hat{\theta}_{SA, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{SA, C4, 0\sigma, \%}$	$\hat{\theta}_{SA, C4, 1\sigma, \%}$	$\hat{\theta}_{SA, C4, 2\sigma, \%}$	$\hat{\theta}_{SA, C4, 4\sigma, \%}$
I_2	80,0%	0,0%	30,0%	0,0%
I_3	401,00	401,00	401,00	401,00

Detalhamento da Tabela 6.14:

$$\left\{ \begin{array}{l} \hat{\theta}_{SA, C4, 0\sigma} = [0,33103; 0,49655; -0,12414; 0,02000; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA, C4, 0\sigma, \%} = [27,586; 41,379; 10,345; 50,000; 48,276; 48,276]^T \\ \hat{\theta}_{SA, C4, 1\sigma} = [0,33103; 0,12414; -0,12414; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA, C4, 1\sigma, \%} = [27,586; 10,345; 10,345; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{SA, C4, 2\sigma} = [0,24828; -0,04138; -0,12414; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{SA, C4, 2\sigma, \%} = [20,690; 3,448; 10,345; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{SA, C4, 4\sigma} = [0,33103; 0,49655; -0,12414; 0,02000; 0,02000; -0,01931]^T \\ \hat{\theta}_{SA, C4, 4\sigma, \%} = [27,586; 41,379; 10,345; 50,000; 50,000; 48,276]^T \end{array} \right.$$

Tabela 6.15: Índices de desempenho para o algoritmo de SA, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SA, C5, 0\sigma}$	$\hat{\theta}_{SA, C5, 1\sigma}$	$\hat{\theta}_{SA, C5, 2\sigma}$	$\hat{\theta}_{SA, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{SA, C5, 0\sigma, \%}$	$\hat{\theta}_{SA, C5, 1\sigma, \%}$	$\hat{\theta}_{SA, C5, 2\sigma, \%}$	$\hat{\theta}_{SA, C5, 4\sigma, \%}$
I_2	0,0%	0,0%	0,0%	0,0%
I_3	401,00	401,00	401,00	401,00

Detalhamento da Tabela 6.15:

$$\left\{ \begin{array}{l} \hat{\theta}_{SA, C5, 0\sigma} = [0,33103; 0,49655; -0,16552; -0,01931; -0,01931; -0,01931; \\ \quad 0,99034; 0,99034; 1,01000; 1,01000; 1,01000; 0,99034]^T \\ \hat{\theta}_{SA, C5, 0\sigma, \%} = [27,586; 41,379; 13,793; 48,276; 48,276; 48,276; \\ \quad 48,276; 48,276; 50,000; 50,000; 50,000; 48,276]^T \\ \hat{\theta}_{SA, C5, 1\sigma} = [0,16552; 0,04138; -0,12414; -0,01931; -0,01931; -0,01931; \\ \quad 0,99034; 0,99034; 1,01000; 0,99034; 0,99034; 1,01000]^T \\ \hat{\theta}_{SA, C5, 1\sigma, \%} = [13,793; 3,448; 10,345; 48,276; 48,276; 48,276; \\ \quad 48,276; 48,276; 50,000; 48,276; 48,276; 50,000]^T \\ \hat{\theta}_{SA, C5, 2\sigma} = [0,04138; 0,16552; -0,57931; -0,01931; 0,02000; -0,01931; \\ \quad 1,01000; 0,99034; 1,01000; 1,01000; 1,01000; 0,99034]^T \\ \hat{\theta}_{SA, C5, 2\sigma, \%} = [3,448; 13,793; 48,276; 48,276; 50,000; 48,276; \\ \quad 50,000; 48,276; 50,000; 50,000; 50,000; 48,276]^T \\ \hat{\theta}_{SA, C5, 4\sigma} = [0,33103; 0,49655; -0,57931; -0,01931; -0,01931; 0,02000; \\ \quad 0,99034; 0,99034; 1,01000; 0,99034; 0,99034; 0,99034]^T \\ \hat{\theta}_{SA, C5, 4\sigma, \%} = [27,586; 41,379; 48,276; 48,276; 48,276; 50,000; \\ \quad 48,276; 48,276; 50,000; 48,276; 48,276; 48,276]^T \end{array} \right.$$

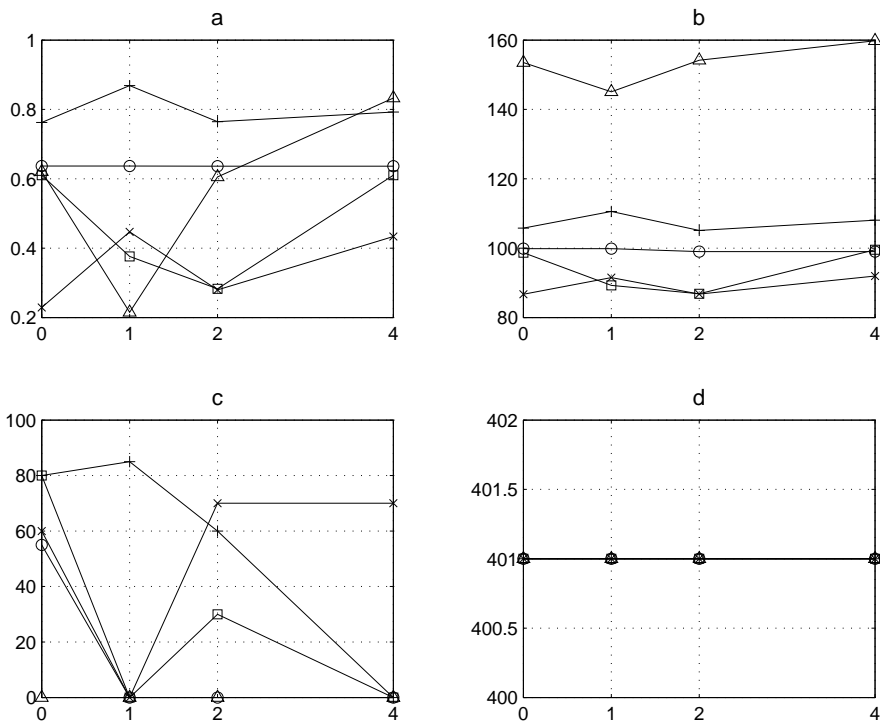


Figura 6.3: Comparação do desempenho do algoritmo de SA para diferentes conjuntos de teste segundo: a) a distância Euclideana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: x; 3: +; 4: □ e 5: △.

6.1.2 Gradiente e Gradiente Conjugado

Os resultados obtidos com o algoritmo do gradiente (GR) são, sob certo aspecto, surpreendentes. Em detrimento ao que se espera, o algoritmo tende a apresentar melhor desempenho para conjuntos de testes com menores índices. Essa afirmativa implica dizer que os melhores resultados desse algoritmo são conseguidos com sinais de entrada incapazes de excitar as dinâmicas longitudinal e látero-direcional da aeronave de forma apropriada.

Uma das justificativas para esse comportamento pode ser encontrada ao se analisar o problema sob outro ponto de vista. A concepção do algoritmo do gradiente pressupõe a resolução de problemas aproximadamente lineares. Mas os sinais responsáveis pela excitação das dinâmicas longitudinal e látero-direcional de uma aeronave devem, supostamente, excitar a dinâmica não-linear do modelo cinemático. Com isso, a excitação da dinâmica não-linear do modelo cinemático pode constituir o motivo pelo qual o algoritmo do gradiente se comporta conforme descrito.

Tabela 6.16: Índices de desempenho para o algoritmo de GR, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GR, C1, 0\sigma}$	$\hat{\theta}_{GR, C1, 1\sigma}$	$\hat{\theta}_{GR, C1, 2\sigma}$	$\hat{\theta}_{GR, C1, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{GR, C1, 0\sigma, \%}$	$\hat{\theta}_{GR, C1, 1\sigma, \%}$	$\hat{\theta}_{GR, C1, 2\sigma, \%}$	$\hat{\theta}_{GR, C1, 4\sigma, \%}$
I_2	55,0%	35,0%	75,0%	55,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.16:

$$\left\{ \begin{array}{l} \hat{\theta}_{GR, C1, 0\sigma} = [-0,35172; 0,24621; -0,31655; -0,01524; 0,00000; 0,00235]^T \\ \hat{\theta}_{GR, C1, 0\sigma, \%} = [29,311; 20,517; 26,379; 38,104; 0,000; 5,862]^T \\ \hat{\theta}_{GR, C1, 1\sigma} = [-0,46345; 0,19862; -0,13242; -0,00552; -0,00882; -0,01434]^T \\ \hat{\theta}_{GR, C1, 1\sigma, \%} = [38,621; 16,552; 11,034; 13,793; 22,069; 35,862]^T \\ \hat{\theta}_{GR, C1, 2\sigma} = [-0,35379; 0,35379; -0,51103; -0,00393; 0,00524; -0,00787]^T \\ \hat{\theta}_{GR, C1, 2\sigma, \%} = [29,482; 29,482; 42,587; 9,828; 13,103; 19,656]^T \\ \hat{\theta}_{GR, C1, 4\sigma} = [-0,18621; 0,37242; -0,24828; -0,01449; -0,00207; -0,01449]^T \\ \hat{\theta}_{GR, C1, 4\sigma, \%} = [15,518; 31,035; 20,689; 36,207; 5,172; 36,207]^T \end{array} \right.$$

Tabela 6.17: Índices de desempenho para o algoritmo de GR, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GR,C2,0\sigma}$	$\hat{\theta}_{GR,C2,1\sigma}$	$\hat{\theta}_{GR,C2,2\sigma}$	$\hat{\theta}_{GR,C2,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GR,C2,0\sigma,\%}$	$\hat{\theta}_{GR,C2,1\sigma,\%}$	$\hat{\theta}_{GR,C2,2\sigma,\%}$	$\hat{\theta}_{GR,C2,4\sigma,\%}$
I_2	75,0%	55,0%	60,0%	60,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.17:

$$\left\{ \begin{array}{l} \hat{\theta}_{GR,C2,0\sigma} = [-0,49655; 0,16552; -0,41379; -0,01931; -0,01793; -0,00138]^T \\ \hat{\theta}_{GR,C2,0\sigma,\%} = [41,379; 13,793; 34,483; 48,276; 44,828; 3,448]^T \\ \hat{\theta}_{GR,C2,1\sigma} = [-0,64552; 0,21518; -0,21518; -0,01256; -0,01256; 0,01076]^T \\ \hat{\theta}_{GR,C2,1\sigma,\%} = [53,793; 17,931; 17,931; 31,379; 31,379; 26,897]^T \\ \hat{\theta}_{GR,C2,2\sigma} = [-0,37241; 0,20690; -0,57931; 0,00138; -0,00414; -0,01655]^T \\ \hat{\theta}_{GR,C2,2\sigma,\%} = [31,034; 17,241; 48,276; 3,448; 10,345; 41,379]^T \\ \hat{\theta}_{GR,C2,4\sigma} = [-0,43034; 0,21518; -0,75310; 0,00000; 0,00179; 0,00000]^T \\ \hat{\theta}_{GR,C2,4\sigma,\%} = [35,862; 17,931; 62,759; 0,000; 13,449; 0,000]^T \end{array} \right.$$

Tabela 6.18: Índices de desempenho para o algoritmo de GR, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GR,C3,0\sigma}$	$\hat{\theta}_{GR,C3,1\sigma}$	$\hat{\theta}_{GR,C3,2\sigma}$	$\hat{\theta}_{GR,C3,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GR,C3,0\sigma,\%}$	$\hat{\theta}_{GR,C3,1\sigma,\%}$	$\hat{\theta}_{GR,C3,2\sigma,\%}$	$\hat{\theta}_{GR,C3,4\sigma,\%}$
I_2	80,0%	85,0%	65,0%	75,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.18:

$$\left\{ \begin{array}{l} \hat{\theta}_{GR,C3,0\sigma} = [-0,41379; 0,24828; -0,24828; 0,01931; -0,01931; -0,01241]^T \\ \hat{\theta}_{GR,C3,0\sigma,\%} = [34,483; 20,690; 20,690; 48,276; 48,276; 31,034]^T \\ \hat{\theta}_{GR,C3,1\sigma} = [0,16552; 0,20690; -0,57931; 0,00000; -0,01793; -0,01931]^T \\ \hat{\theta}_{GR,C3,1\sigma,\%} = [13,793; 17,241; 48,276; 0,000; 44,828; 48,276]^T \\ \hat{\theta}_{GR,C3,2\sigma} = [-0,45517; 0,20690; -0,12414; -0,01931; -0,01931; 0,00276]^T \\ \hat{\theta}_{GR,C3,2\sigma,\%} = [37,931; 17,241; 10,345; 48,276; 48,276; 6,897]^T \\ \hat{\theta}_{GR,C3,4\sigma} = [-0,28966; 0,20690; -0,57931; -0,00828; -0,00552; -0,00138]^T \\ \hat{\theta}_{GR,C3,4\sigma,\%} = [24,138; 17,241; 48,276; 20,690; 13,793; 3,448]^T \end{array} \right.$$

Tabela 6.19: Índices de desempenho para o algoritmo de GR, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GR, C4, 0\sigma}$	$\hat{\theta}_{GR, C4, 1\sigma}$	$\hat{\theta}_{GR, C4, 2\sigma}$	$\hat{\theta}_{GR, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{GR, C4, 0\sigma, \%}$	$\hat{\theta}_{GR, C4, 1\sigma, \%}$	$\hat{\theta}_{GR, C4, 2\sigma, \%}$	$\hat{\theta}_{GR, C4, 4\sigma, \%}$
I_2	65,0%	85,0%	80,0%	80,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.19:

$$\left\{ \begin{array}{l} \hat{\theta}_{GR, C4, 0\sigma} = [-0,39103; 0,08690; -0,52138; -0,01738; -0,01593; -0,01593]^T \\ \hat{\theta}_{GR, C4, 0\sigma, \%} = [32,586; 7,242; 43,448; 43,448; 39,828; 39,828]^T \\ \hat{\theta}_{GR, C4, 1\sigma} = [-0,27311; 0,18207; -0,63724; 0,02124; 0,01213; -0,01063]^T \\ \hat{\theta}_{GR, C4, 1\sigma, \%} = [22,759; 15,172; 53,104; 53,104; 30,345; 26,552]^T \\ \hat{\theta}_{GR, C4, 2\sigma} = [-0,57103; 0,42827; -0,19035; 0,01268; 0,01745; -0,02062]^T \\ \hat{\theta}_{GR, C4, 2\sigma, \%} = [47,586; 35,689; 15,862; 31,724; 43,621; 51,552]^T \\ \hat{\theta}_{GR, C4, 4\sigma} = [-0,54620; 0,44689; -0,64552; 0,01159; -0,01986; 0,00166]^T \\ \hat{\theta}_{GR, C4, 4\sigma, \%} = [45,517; 37,241; 53,794; 28,966; 49,655; 4,138]^T \end{array} \right.$$

Tabela 6.20: Índices de desempenho para o algoritmo de GR, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GR, C5, 0\sigma}$	$\hat{\theta}_{GR, C5, 1\sigma}$	$\hat{\theta}_{GR, C5, 2\sigma}$	$\hat{\theta}_{GR, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{GR, C5, 0\sigma, \%}$	$\hat{\theta}_{GR, C5, 1\sigma, \%}$	$\hat{\theta}_{GR, C5, 2\sigma, \%}$	$\hat{\theta}_{GR, C5, 4\sigma, \%}$
I_2	85,0%	90,0%	80,0%	90,0%
I_3	650,00	650,00	650,00	650,00

Detalhamento da Tabela 6.20:

$$\left\{ \begin{array}{l} \hat{\theta}_{GR, C5, 0\sigma} = [-0,49655; 0,45517; -0,08276; -0,01793; 0,00828; -0,01655; \\ \quad 0,99310; 0,99448; 0,99862; 0,99103; 0,99586; 1,00345]^T \\ \hat{\theta}_{GR, C5, 0\sigma, \%} = [41,379; 37,931; 6,897; 44,828; 20,690; 41,379; \\ \quad 34,483; 27,586; 6,897; 44,828; 20,690; 17,241]^T \\ \hat{\theta}_{GR, C5, 1\sigma} = [0,41379; 0,08276; -0,53793; -0,01379; 0,01379; -0,01241; \\ \quad 0,99793; 1,00276; 0,99034; 0,99103; 1,00621; 0,99517]^T \\ \hat{\theta}_{GR, C5, 1\sigma, \%} = [34,483; 6,897; 44,828; 34,483; 34,483; 31,034; \\ \quad 10,345; 13,793; 48,276; 44,828; 31,034; 24,138]^T \\ \hat{\theta}_{GR, C5, 2\sigma} = [0,00000; 0,16552; -0,04138; 0,00552; -0,01517; 0,00000; \\ \quad 1,00483; 0,99448; 1,00138; 1,00414; 0,99517; 0,99172]^T \\ \hat{\theta}_{GR, C5, 2\sigma, \%} = [0,000; 13,793; 3,448; 13,793; 37,931; 0,000; \\ \quad 24,138; 27,586; 6,897; 20,690; 24,138; 41,379]^T \\ \hat{\theta}_{GR, C5, 4\sigma} = [-0,20690; 0,33103; -0,24828; 0,00000; 0,00690; 0,01793; \\ \quad 0,99931; 0,99793; 0,99448; 0,99310; 1,00828; 1,00069]^T \\ \hat{\theta}_{GR, C5, 4\sigma, \%} = [17,241; 27,586; 20,690; 0,000; 17,241; 44,828; \\ \quad 3,448; 10,345; 27,586; 34,483; 41,379; 3,448]^T \end{array} \right.$$

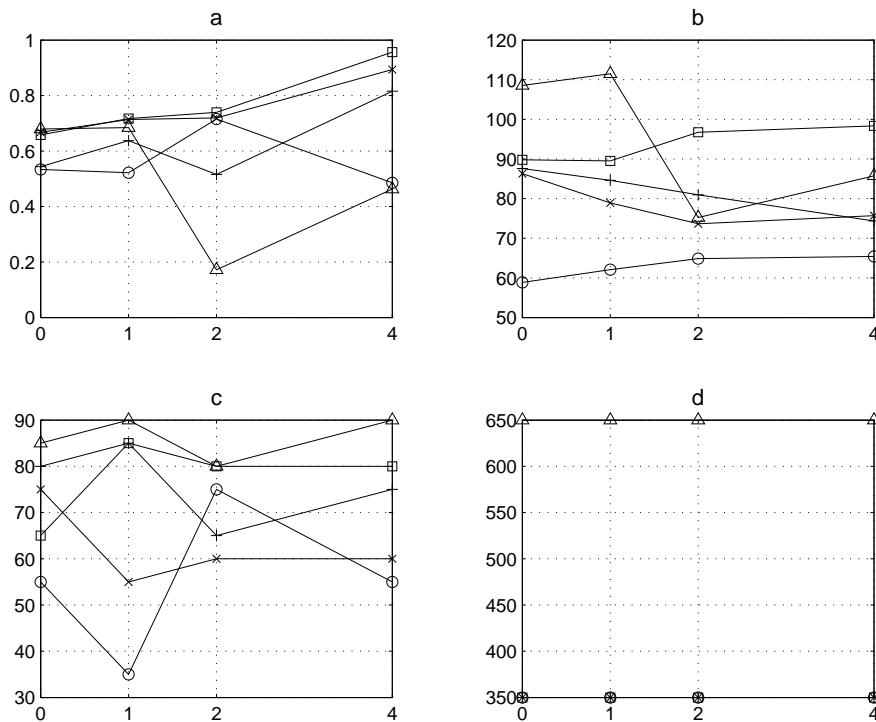


Figura 6.4: Comparação do desempenho do algoritmo de GR para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: ×; 3: +; 4: □ e 5: △.

A análise dos resultados obtidos com o algoritmo do gradiente conjugado (GC) segue aquela relativa ao algoritmo do gradiente. A distinção se faz em um desempenho geral um pouco melhor neste caso.

Tabela 6.21: Índices de desempenho para o algoritmo de GC, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GC,C1,0\sigma}$	$\hat{\theta}_{GC,C1,1\sigma}$	$\hat{\theta}_{GC,C1,2\sigma}$	$\hat{\theta}_{GC,C1,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GC,C1,0\sigma,\%}$	$\hat{\theta}_{GC,C1,1\sigma,\%}$	$\hat{\theta}_{GC,C1,2\sigma,\%}$	$\hat{\theta}_{GC,C1,4\sigma,\%}$
I_2	65,0%	55,0%	60,0%	60,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.21:

$$\left\{ \begin{array}{l} \hat{\theta}_{GC,C1,0\sigma} = [-0,07448; -0,22345; -0,48414; 0,00497; -0,00993; -0,00248]^T \\ \hat{\theta}_{GC,C1,0\sigma,\%} = [6,207; 18,621; 40,345; 12,414; 24,827; 6,207]^T \\ \hat{\theta}_{GC,C1,1\sigma} = [-0,33517; 0,14897; -0,48414; 0,00124; 0,00248; -0,00869]^T \\ \hat{\theta}_{GC,C1,1\sigma,\%} = [27,931; 12,414; 40,345; 3,103; 6,207; 21,724]^T \\ \hat{\theta}_{GC,C1,2\sigma} = [-0,31655; 0,31655; -0,42207; 0,00469; -0,00938; -0,00938]^T \\ \hat{\theta}_{GC,C1,2\sigma,\%} = [26,379; 26,379; 35,172; 11,724; 23,448; 23,448]^T \\ \hat{\theta}_{GC,C1,4\sigma} = [-0,26069; -0,11173; -0,33517; -0,01614; -0,00248; -0,01117]^T \\ \hat{\theta}_{GC,C1,4\sigma,\%} = [21,724; 9,311; 27,931; 40,345; 6,207; 27,931]^T \end{array} \right.$$

Tabela 6.22: Índices de desempenho para o algoritmo de GC, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GC,C2,0\sigma}$	$\hat{\theta}_{GC,C2,1\sigma}$	$\hat{\theta}_{GC,C2,2\sigma}$	$\hat{\theta}_{GC,C2,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GC,C2,0\sigma,\%}$	$\hat{\theta}_{GC,C2,1\sigma,\%}$	$\hat{\theta}_{GC,C2,2\sigma,\%}$	$\hat{\theta}_{GC,C2,4\sigma,\%}$
I_2	65,0%	75,0%	75,0%	70,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.22:

$$\left\{ \begin{array}{l} \hat{\theta}_{GC, C2, 0\sigma} = [0,20690; -0,28966; -0,37241; 0,01379; -0,01931; 0,01241]^T \\ \hat{\theta}_{GC, C2, 0\sigma, \%} = [17,241; 24,138; 31,034; 34,483; 48,276; 31,034]^T \\ \hat{\theta}_{GC, C2, 1\sigma} = [-0,24828; 0,20690; 0,12414; 0,01793; -0,01793; -0,00138]^T \\ \hat{\theta}_{GC, C2, 1\sigma, \%} = [20,690; 17,241; 10,345; 44,828; 44,828; 3,448]^T \\ \hat{\theta}_{GC, C2, 2\sigma} = [-0,37241; 0,20690; 0,20690; -0,01793; 0,00138; -0,01379]^T \\ \hat{\theta}_{GC, C2, 2\sigma, \%} = [31,034; 17,241; 17,241; 44,828; 3,448; 34,483]^T \\ \hat{\theta}_{GC, C2, 4\sigma} = [-0,34759; -0,09931; -0,24828; -0,01159; -0,00497; -0,01820]^T \\ \hat{\theta}_{GC, C2, 4\sigma, \%} = [28,966; 8,276; 20,689; 28,966; 12,414; 45,517]^T \end{array} \right.$$

Tabela 6.23: Índices de desempenho para o algoritmo de GC, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GC, C3, 0\sigma}$	$\hat{\theta}_{GC, C3, 1\sigma}$	$\hat{\theta}_{GC, C3, 2\sigma}$	$\hat{\theta}_{GC, C3, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{GC, C3, 0\sigma, \%}$	$\hat{\theta}_{GC, C3, 1\sigma, \%}$	$\hat{\theta}_{GC, C3, 2\sigma, \%}$	$\hat{\theta}_{GC, C3, 4\sigma, \%}$
I_2	60,0%	70,0%	90,0%	65,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.23:

$$\left\{ \begin{array}{l} \hat{\theta}_{GC, C3, 0\sigma} = [-0,57931; -0,04138; -0,37241; -0,01793; -0,00552; -0,01517]^T \\ \hat{\theta}_{GC, C3, 0\sigma, \%} = [48,276; 3,448; 31,034; 44,828; 13,793; 37,931]^T \\ \hat{\theta}_{GC, C3, 1\sigma} = [0,00000; -0,26897; -0,16138; -0,01793; -0,01613; -0,01076]^T \\ \hat{\theta}_{GC, C3, 1\sigma, \%} = [0,000; 22,413; 13,449; 44,828; 40,344; 26,897]^T \\ \hat{\theta}_{GC, C3, 2\sigma} = [-0,41379; 0,20690; -0,57931; -0,00828; -0,01793; -0,01379]^T \\ \hat{\theta}_{GC, C3, 2\sigma, \%} = [34,483; 17,241; 48,276; 20,690; 44,828; 34,483]^T \\ \hat{\theta}_{GC, C3, 4\sigma} = [-0,54620; 0,24828; -0,64552; -0,00662; -0,01324; -0,00166]^T \\ \hat{\theta}_{GC, C3, 4\sigma, \%} = [45,517; 20,689; 53,794; 16,552; 33,103; 4,138]^T \end{array} \right.$$

Tabela 6.24: Índices de desempenho para o algoritmo de GC, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GC, C4, 0\sigma}$	$\hat{\theta}_{GC, C4, 1\sigma}$	$\hat{\theta}_{GC, C4, 2\sigma}$	$\hat{\theta}_{GC, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{GC, C4, 0\sigma, \%}$	$\hat{\theta}_{GC, C4, 1\sigma, \%}$	$\hat{\theta}_{GC, C4, 2\sigma, \%}$	$\hat{\theta}_{GC, C4, 4\sigma, \%}$
I_2	90,0%	70,0%	75,0%	70,0%
I_3	350,00	350,00	350,00	350,00

Detalhamento da Tabela 6.24:

$$\left\{ \begin{array}{l} \hat{\theta}_{GC, C4, 0\sigma} = [0,46345; -0,36414; -0,46345; -0,00110; -0,00993; -0,01434]^T \\ \hat{\theta}_{GC, C4, 0\sigma, \%} = [38,621; 30,345; 38,621; 2,758; 24,827; 35,862]^T \\ \hat{\theta}_{GC, C4, 1\sigma} = [-0,57931; -0,24828; -0,24828; 0,01379; -0,00552; 0,01655]^T \\ \hat{\theta}_{GC, C4, 1\sigma, \%} = [48,276; 20,690; 20,690; 34,483; 13,793; 41,379]^T \\ \hat{\theta}_{GC, C4, 2\sigma} = [-0,57931; -0,28966; -0,49655; -0,00552; -0,01793; 0,01517]^T \\ \hat{\theta}_{GC, C4, 2\sigma, \%} = [48,276; 24,138; 41,379; 13,793; 44,828; 37,931]^T \\ \hat{\theta}_{GC, C4, 4\sigma} = [-0,48413; 0,37656; -0,10759; -0,00897; -0,02151; 0,00718]^T \\ \hat{\theta}_{GC, C4, 4\sigma, \%} = [40,344; 31,379; 8,966; 22,413; 53,793; 17,931]^T \end{array} \right.$$

Tabela 6.25: Índices de desempenho para o algoritmo de GC, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GC, C5, 0\sigma}$	$\hat{\theta}_{GC, C5, 1\sigma}$	$\hat{\theta}_{GC, C5, 2\sigma}$	$\hat{\theta}_{GC, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{GC, C5, 0\sigma, \%}$	$\hat{\theta}_{GC, C5, 1\sigma, \%}$	$\hat{\theta}_{GC, C5, 2\sigma, \%}$	$\hat{\theta}_{GC, C5, 4\sigma, \%}$
I_2	80,0%	80,0%	75,0%	65,0%
I_3	650,00	650,00	650,00	650,00

Detalhamento da Tabela 6.25:

$$\left\{ \begin{array}{l} \hat{\theta}_{GC, C5, 0\sigma} = [0,24828; 0,04138; -0,08276; 0,00000; -0,01241; -0,01655; \\ \quad 0,99379; 0,99448; 0,99517; 0,99241; 1,00069; 0,99172]^T \\ \hat{\theta}_{GC, C5, 0\sigma, \%} = [20,690; 3,448; 6,897; 0,000; 31,034; 41,379; \\ \quad 31,034; 27,586; 24,138; 37,931; 3,448; 41,379]^T \\ \hat{\theta}_{GC, C5, 1\sigma} = [-0,08276; 0,41379; -0,16552; -0,01931; -0,01103; -0,00276; \\ \quad 0,99310; 1,00552; 1,00207; 1,00759; 1,00207; 1,00690]^T \\ \hat{\theta}_{GC, C5, 1\sigma, \%} = [6,897; 34,483; 13,793; 48,276; 27,586; 6,897; \\ \quad 34,483; 27,586; 10,345; 37,931; 10,345; 34,483]^T \\ \hat{\theta}_{GC, C5, 2\sigma} = [-0,24828; 0,45517; -0,24828; -0,01793; -0,01793; -0,00966; \\ \quad 0,99034; 1,00345; 0,99241; 0,99310; 0,99862; 1,00207]^T \\ \hat{\theta}_{GC, C5, 2\sigma, \%} = [20,690; 37,931; 20,690; 44,828; 44,828; 24,138; \\ \quad 48,276; 17,241; 37,931; 34,483; 6,897; 10,345]^T \\ \hat{\theta}_{GC, C5, 4\sigma} = [0,24828; -0,12414; -0,33103; 0,00966; -0,00552; -0,01241; \\ \quad 0,99172; 0,99448; 1,00690; 0,99379; 0,99793; 0,99586]^T \\ \hat{\theta}_{GC, C5, 4\sigma, \%} = [20,690; 10,345; 27,586; 24,138; 13,793; 31,034; \\ \quad 41,379; 27,586; 34,483; 31,034; 10,345; 20,690]^T \end{array} \right.$$

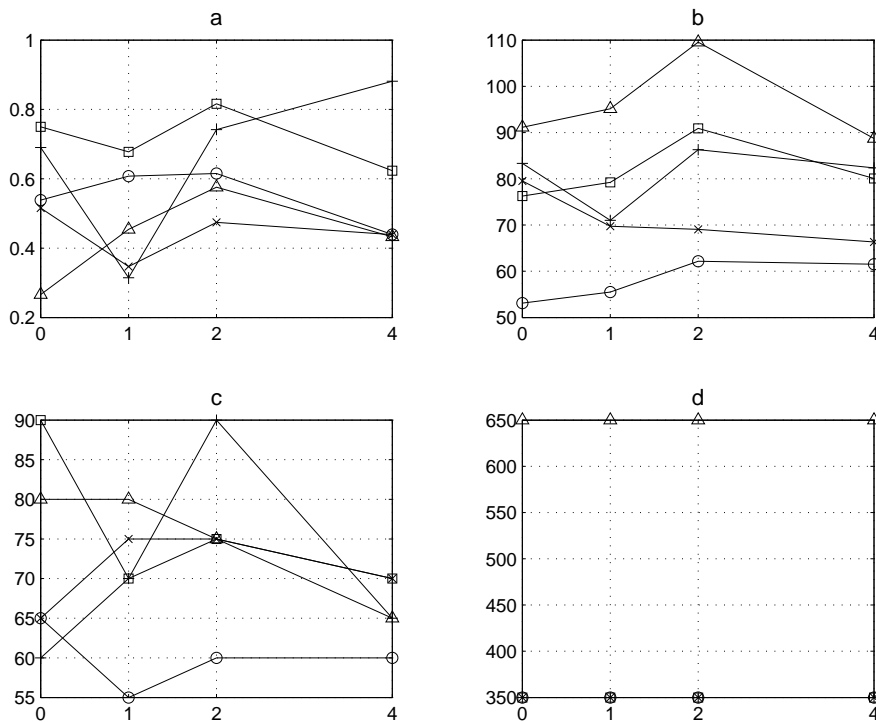


Figura 6.5: Comparação do desempenho do algoritmo de GC para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: x; 3: +; 4: □ e 5: △.

6.1.3 Gradiente Conjugado Pré-Condicionado e Gauss-Newton

Os resultados obtidos com o algoritmo do gradiente conjugado pré-condicionado (GCP) destoam em relação aos alcançados com os do gradiente e gradiente conjugado. Apesar de, assim como nos dois casos anteriores, haver certa tendência a um pior desempenho com o aumento do nível de contaminação por ruído, os resultados indicam pouca sensibilidade do algoritmo em relação à manobra utilizada.

Com o transcorrer das realizações, nota-se a tentativa de utilização de vetores com valores muito distantes do ideal. Esse fato dá origem a pontos muito dispersos no espaço de soluções e prejudica o desempenho do algoritmo. Uma das justificativas para esse cenário pode-se tratar de uma determinação pouco adequada da matriz de pré-condicionamento. Nesse caso, o ponto a ser determinado pelo algoritmo pode não mais se tratar do ponto ótimo do problema. Com isso, o algoritmo tende a divergir.

Tabela 6.26: Índices de desempenho para o algoritmo de GCP, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GCP,C1,0\sigma}$	$\hat{\theta}_{GCP,C1,1\sigma}$	$\hat{\theta}_{GCP,C1,2\sigma}$	$\hat{\theta}_{GCP,C1,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GCP,C1,0\sigma,\%}$	$\hat{\theta}_{GCP,C1,1\sigma,\%}$	$\hat{\theta}_{GCP,C1,2\sigma,\%}$	$\hat{\theta}_{GCP,C1,4\sigma,\%}$
I_2	85,0%	0,0%	0,0%	75,0%
I_3	1976,66	1809,92	1739,64	1778,00

Detalhamento da Tabela 6.26:

$$\left\{ \begin{array}{l} \hat{\theta}_{GCP,C1,0\sigma} = [0,20690; 0,57931; 0,53793; -0,00138; -0,00276; -0,01931]^T \\ \hat{\theta}_{GCP,C1,0\sigma,\%} = [17,241; 48,276; 44,828; 3,448; 6,897; 48,276]^T \\ \hat{\theta}_{GCP,C1,1\sigma} = [0,20690; 0,57931; -0,45517; -0,01931; -0,01103; -0,01241]^T \\ \hat{\theta}_{GCP,C1,1\sigma,\%} = [17,241; 48,276; 37,931; 48,276; 27,586; 31,034]^T \\ \hat{\theta}_{GCP,C1,2\sigma} = [0,20690; 0,57931; 0,12414; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GCP,C1,2\sigma,\%} = [17,241; 48,276; 10,345; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GCP,C1,4\sigma} = [0,20690; 0,57931; -0,49655; -0,01793; -0,00276; -0,00138]^T \\ \hat{\theta}_{GCP,C1,4\sigma,\%} = [17,241; 48,276; 41,379; 44,828; 6,897; 3,448]^T \end{array} \right.$$

Tabela 6.27: Índices de desempenho para o algoritmo de GCP, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GCP,C2,0\sigma}$	$\hat{\theta}_{GCP,C2,1\sigma}$	$\hat{\theta}_{GCP,C2,2\sigma}$	$\hat{\theta}_{GCP,C2,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GCP,C2,0\sigma,\%}$	$\hat{\theta}_{GCP,C2,1\sigma,\%}$	$\hat{\theta}_{GCP,C2,2\sigma,\%}$	$\hat{\theta}_{GCP,C2,4\sigma,\%}$
I_2	0,0%	55,0%	0,0%	0,0%
I_3	1974,14	1790,60	1844,50	1724,10

Detalhamento da Tabela 6.27:

$$\left\{ \begin{array}{l} \hat{\theta}_{GCP,C2,0\sigma} = [0,37241; 0,28966; -0,49655; -0,01793; -0,01931; 0,00690]^T \\ \hat{\theta}_{GCP,C2,0\sigma,\%} = [31,034; 24,138; 41,379; 44,828; 48,276; 17,241]^T \\ \hat{\theta}_{GCP,C2,1\sigma} = [0,45517; -0,37241; -0,04138; -0,00138; -0,00276; -0,01379]^T \\ \hat{\theta}_{GCP,C2,1\sigma,\%} = [37,931; 31,034; 3,448; 3,448; 6,897; 34,483]^T \\ \hat{\theta}_{GCP,C2,2\sigma} = [0,45517; -0,41379; -0,04138; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GCP,C2,2\sigma,\%} = [37,931; 34,483; 3,448; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GCP,C2,4\sigma} = [0,45517; -0,37241; -0,04138; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GCP,C2,4\sigma,\%} = [37,931; 31,034; 3,448; 48,276; 48,276; 48,276]^T \end{array} \right.$$

Tabela 6.28: Índices de desempenho para o algoritmo de GCP, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GCP,C3,0\sigma}$	$\hat{\theta}_{GCP,C3,1\sigma}$	$\hat{\theta}_{GCP,C3,2\sigma}$	$\hat{\theta}_{GCP,C3,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GCP,C3,0\sigma,\%}$	$\hat{\theta}_{GCP,C3,1\sigma,\%}$	$\hat{\theta}_{GCP,C3,2\sigma,\%}$	$\hat{\theta}_{GCP,C3,4\sigma,\%}$
I_2	85,0%	70,0%	0,0%	0,0%
I_3	1758,12	1906,24	1844,50	1724,10

Detalhamento da Tabela 6.28:

$$\left\{ \begin{array}{l} \hat{\theta}_{GCP,C3,0\sigma} = [-0,20690; 0,20690; -0,24828; -0,01931; 0,01379; -0,00552]^T \\ \hat{\theta}_{GCP,C3,0\sigma,\%} = [17,241; 17,241; 20,690; 48,276; 34,483; 13,793]^T \\ \hat{\theta}_{GCP,C3,1\sigma} = [0,20690; 0,20690; -0,33103; -0,01655; -0,00966; 0,00000]^T \\ \hat{\theta}_{GCP,C3,1\sigma,\%} = [17,241; 17,241; 27,586; 41,379; 24,138; 0,000]^T \\ \hat{\theta}_{GCP,C3,2\sigma} = [0,45517; -0,41379; -0,04138; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GCP,C3,2\sigma,\%} = [37,931; 34,483; 3,448; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GCP,C3,4\sigma} = [0,45517; -0,37241; -0,04138; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GCP,C3,4\sigma,\%} = [37,931; 31,034; 3,448; 48,276; 48,276; 48,276]^T \end{array} \right.$$

Tabela 6.29: Índices de desempenho para o algoritmo de GCP, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GCP,C4,0\sigma}$	$\hat{\theta}_{GCP,C4,1\sigma}$	$\hat{\theta}_{GCP,C4,2\sigma}$	$\hat{\theta}_{GCP,C4,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GCP,C4,0\sigma,\%}$	$\hat{\theta}_{GCP,C4,1\sigma,\%}$	$\hat{\theta}_{GCP,C4,2\sigma,\%}$	$\hat{\theta}_{GCP,C4,4\sigma,\%}$
I_2	85,0%	55,0%	0,0%	0,0%
I_3	1758,12	1790,60	1784,02	1824,90

Detalhamento da Tabela 6.29:

$$\left\{ \begin{array}{l} \hat{\theta}_{GCP,C4,0\sigma} = [-0,20690; 0,24828; -0,20690; -0,01931; 0,01379; -0,00552]^T \\ \hat{\theta}_{GCP,C4,0\sigma,\%} = [17,241; 20,690; 17,241; 6,897; 3,448; 13,793]^T \\ \hat{\theta}_{GCP,C4,1\sigma} = [0,45517; -0,37241; -0,04138; -0,01931; -0,01931; -0,01379]^T \\ \hat{\theta}_{GCP,C4,1\sigma,\%} = [37,931; 31,034; 3,448; 34,483; 48,276; 34,483]^T \\ \hat{\theta}_{GCP,C4,2\sigma} = [0,49655; 0,41379; -0,37241; -0,00276; -0,00138; -0,01931]^T \\ \hat{\theta}_{GCP,C4,2\sigma,\%} = [41,379; 34,483; 31,034; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GCP,C4,4\sigma} = [-0,20690; -0,04138; -0,37241; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GCP,C4,4\sigma,\%} = [17,241; 3,448; 31,034; 48,276; 48,276; 48,276]^T \end{array} \right.$$

Tabela 6.30: Índices de desempenho para o algoritmo de GCP, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GCP,C5,0\sigma}$	$\hat{\theta}_{GCP,C5,1\sigma}$	$\hat{\theta}_{GCP,C5,2\sigma}$	$\hat{\theta}_{GCP,C5,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GCP,C5,0\sigma,\%}$	$\hat{\theta}_{GCP,C5,1\sigma,\%}$	$\hat{\theta}_{GCP,C5,2\sigma,\%}$	$\hat{\theta}_{GCP,C5,4\sigma,\%}$
I_2	0,0%	0,0%	0,0%	0,0%
I_3	2967,64	2967,64	2967,64	2967,64

Detalhamento da Tabela 6.30:

$$\left\{ \begin{array}{l} \hat{\theta}_{GCP, C5, 0\sigma} = [0,49655; -0,08276; -0,04138; 0,00966; -0,00138; -0,00552; \\ \quad 1,00207; 0,99517; 0,99034; 0,99034; 0,99034; 0,99586]^T \\ \hat{\theta}_{GCP, C5, 0\sigma, \%} = [41,379; 6,897; 3,448; 24,138; 3,448; 13,793; \\ \quad 10,345; 24,138; 48,276; 48,276; 48,276; 20,690]^T \\ \hat{\theta}_{GCP, C5, 1\sigma} = [0,49655; -0,12414; 0,24828; 0,00966; -0,00276; -0,00552; \\ \quad 0,99379; 0,99172; 0,99172; 0,99034; 1,00483; 0,99034]^T \\ \hat{\theta}_{GCP, C5, 1\sigma, \%} = [41,379; 10,345; 20,690; 24,138; 6,897; 13,793; \\ \quad 31,034; 41,379; 41,379; 48,276; 24,138; 48,276]^T \\ \hat{\theta}_{GCP, C5, 2\sigma} = [-0,12414; -0,12414; -0,04138; 0,00966; -0,00276; -0,00552; \\ \quad 0,99724; 0,99517; 1,00414; 0,99034; 0,99034; 0,99034]^T \\ \hat{\theta}_{GCP, C5, 2\sigma, \%} = [10,345; 10,345; 3,448; 24,138; 6,897; 13,793; \\ \quad 13,793; 24,138; 20,690; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GCP, C5, 4\sigma} = [-0,12414; -0,08276; 0,16552; 0,01103; -0,00138; -0,00552; \\ \quad 1,00552; 0,99034; 0,99655; 0,99034; 0,99034; 1,00069]^T \\ \hat{\theta}_{GCP, C5, 4\sigma, \%} = [10,345; 6,897; 13,793; 27,586; 3,448; 13,793; \\ \quad 27,586; 48,276; 17,241; 48,276; 48,276; 3,448]^T \end{array} \right.$$

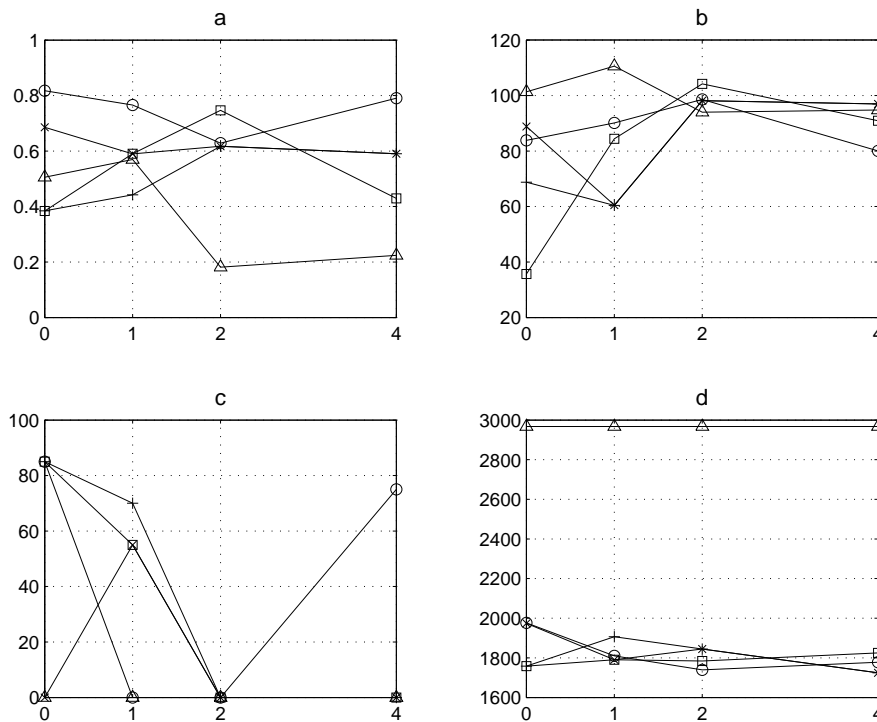


Figura 6.6: Comparação do desempenho do algoritmo de GCP para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: ×; 3: +; 4: □ e 5: △.

Resultados bastante semelhantes ocorrem com a utilização do algoritmo de Gauss-Newton (GN). De fato, esse algoritmo não converge para qualquer cenário de estimação de parâmetros de polarização e ganho simultaneamente¹. Essa constatação torna o algoritmo de Gauss-Newton o que piores resultados apresentou neste trabalho. Nesse contexto, torna-se importante observar que os valores iguais a 0% para I_2 nas Tabelas 6.31, 6.32, 6.33 e 6.34 correspondem a situações em que a primeira fronteira norma-infinito se encontra fora do limite de 100%.

Embora os algoritmos de Gauss-Newton e do gradiente conjugado pré-condicionado em princípio não apresentem grandes semelhanças, os motivos para a divergência desses algoritmos podem ser entendidos de maneira similar. Ambos os algoritmos, em última instância, resolvem o problema

$$A\theta^* = \mathbf{b}$$

em que a distinção se faz apenas na definição da matriz A , segundo as equações (4.24) e (4.33). Com isso, assim como problemas de determinação da matriz de pré-condicionamento, dificuldades de estimação da matriz Hessiana tendem a criar pontos muito dispersos no espaço de soluções. Essa pode-se tratar de uma das justificativas para o mau desempenho geral do algoritmo de Gauss-Newton neste trabalho.

Tabela 6.31: Índices de desempenho para o algoritmo de GN, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GN, C1, 0\sigma}$	$\hat{\theta}_{GN, C1, 1\sigma}$	$\hat{\theta}_{GN, C1, 2\sigma}$	$\hat{\theta}_{GN, C1, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{GN, C1, 0\sigma, \%}$	$\hat{\theta}_{GN, C1, 1\sigma, \%}$	$\hat{\theta}_{GN, C1, 2\sigma, \%}$	$\hat{\theta}_{GN, C1, 4\sigma, \%}$
I_2	0,0%	0,0%	0,0%	0,0%
I_3	140,00	140,00	140,00	140,00

¹Esse é o motivo pelo qual não se apresentam resultados desse algoritmo para o conjunto de testes 5.

Detalhamento da Tabela 6.31:

$$\left\{ \begin{array}{l} \hat{\theta}_{GN,C1,0\sigma} = [0,20690; 0,57931; -0,49655; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C1,0\sigma,\%} = [17,241; 48,276; 41,379; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C1,1\sigma} = [0,20690; 0,57931; -0,57931; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C1,1\sigma,\%} = [17,241; 48,276; 48,276; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C1,2\sigma} = [0,20690; 0,57931; -0,57931; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C1,2\sigma,\%} = [17,241; 48,276; 48,276; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C1,4\sigma} = [0,20690; 0,57931; -0,16552; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C1,4\sigma,\%} = [17,241; 48,276; 13,793; 48,276; 48,276; 48,276]^T \end{array} \right.$$

Tabela 6.32: Índices de desempenho para o algoritmo de GN, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GN,C2,0\sigma}$	$\hat{\theta}_{GN,C2,1\sigma}$	$\hat{\theta}_{GN,C2,2\sigma}$	$\hat{\theta}_{GN,C2,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GN,C2,0\sigma,\%}$	$\hat{\theta}_{GN,C2,1\sigma,\%}$	$\hat{\theta}_{GN,C2,2\sigma,\%}$	$\hat{\theta}_{GN,C2,4\sigma,\%}$
I_2	0,0%	0,0%	0,0%	75,0%
I_3	140,00	140,00	140,00	140,00

Detalhamento da Tabela 6.32:

$$\left\{ \begin{array}{l} \hat{\theta}_{GN,C2,0\sigma} = [0,37241; -0,53793; -0,12414; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C2,0\sigma,\%} = [31,034; 44,828; 10,345; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C2,1\sigma} = [0,53793; 0,20690; -0,49655; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C2,1\sigma,\%} = [44,828; 17,241; 41,379; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C2,2\sigma} = [0,28966; -0,28966; -0,53793; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C2,2\sigma,\%} = [24,138; 24,138; 44,828; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C2,4\sigma} = [0,41379; 0,28966; -0,53793; -0,01103; 0,00276; -0,01655]^T \\ \hat{\theta}_{GN,C2,4\sigma,\%} = [34,483; 24,138; 44,828; 27,586; 6,897; 41,379]^T \end{array} \right.$$

Tabela 6.33: Índices de desempenho para o algoritmo de GN, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GN,C3,0\sigma}$	$\hat{\theta}_{GN,C3,1\sigma}$	$\hat{\theta}_{GN,C3,2\sigma}$	$\hat{\theta}_{GN,C3,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GN,C3,0\sigma,\%}$	$\hat{\theta}_{GN,C3,1\sigma,\%}$	$\hat{\theta}_{GN,C3,2\sigma,\%}$	$\hat{\theta}_{GN,C3,4\sigma,\%}$
I_2	0,0%	0,0%	0,0%	0,0%
I_3	140,00	140,00	140,00	140,00

Detalhamento da Tabela 6.33:

$$\left\{ \begin{array}{l} \hat{\theta}_{GN,C3,0\sigma} = [0,45517; 0,16552; -0,24828; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C3,0\sigma,\%} = [37,931; 13,793; 20,690; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C3,1\sigma} = [0,04138; -0,08276; -0,49655; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C3,1\sigma,\%} = [3,448; 6,897; 41,379; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C3,2\sigma} = [0,45517; -0,04138; -0,41379; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C3,2\sigma,\%} = [37,931; 3,448; 34,483; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C3,4\sigma} = [-0,57931; 0,20690; -0,49655; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C3,4\sigma,\%} = [48,276; 17,241; 41,379; 48,276; 48,276; 48,276]^T \end{array} \right.$$

Tabela 6.34: Índices de desempenho para o algoritmo de GN, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{GN,C4,0\sigma}$	$\hat{\theta}_{GN,C4,1\sigma}$	$\hat{\theta}_{GN,C4,2\sigma}$	$\hat{\theta}_{GN,C4,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{GN,C4,0\sigma,\%}$	$\hat{\theta}_{GN,C4,1\sigma,\%}$	$\hat{\theta}_{GN,C4,2\sigma,\%}$	$\hat{\theta}_{GN,C4,4\sigma,\%}$
I_2	0,0%	0,0%	0,0%	0,0%
I_3	140,00	140,00	140,00	140,00

Detalhamento da Tabela 6.34:

$$\left\{ \begin{array}{l} \hat{\theta}_{GN,C4,0\sigma} = [0,41379; 0,16552; -0,53793; -0,01931; -0,01931; 0,01103]^T \\ \hat{\theta}_{GN,C4,0\sigma,\%} = [34,483; 13,793; 44,828; 48,276; 48,276; 27,586]^T \\ \hat{\theta}_{GN,C4,1\sigma} = [-0,45517; -0,33103; -0,04138; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C4,1\sigma,\%} = [37,931; 27,586; 3,448; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C4,2\sigma} = [-0,04138; 0,20690; -0,57931; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C4,2\sigma,\%} = [3,448; 17,241; 48,276; 48,276; 48,276; 48,276]^T \\ \hat{\theta}_{GN,C4,4\sigma} = [-0,37241; 0,00000; -0,45517; -0,01931; -0,01931; -0,01931]^T \\ \hat{\theta}_{GN,C4,4\sigma,\%} = [31,034; 0,000; 37,931; 48,276; 48,276; 48,276]^T \end{array} \right.$$

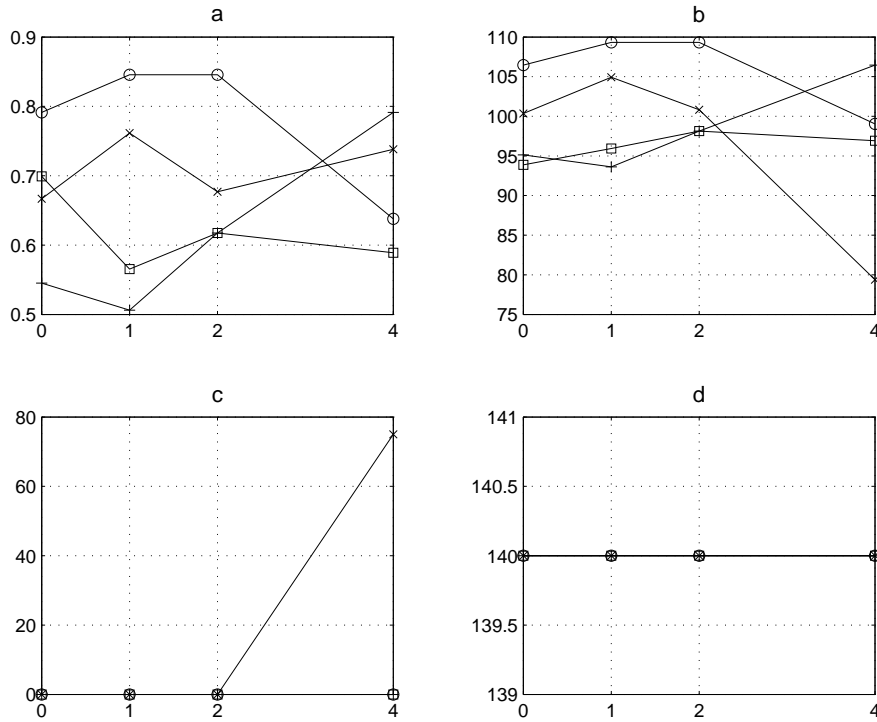


Figura 6.7: Comparação do desempenho do algoritmo de GN para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: \times ; 3: +; 4: \square e 5: \triangle .

6.1.4 Levenberg-Marquardt

O algoritmo de Levenberg-Marquardt (LM) apresenta, dentre todos os algoritmos analisados, os resultados mais condizentes com a expectativa inicial. Além da diminuição de desempenho causada pelo aumento do nível de contaminação dos sinais de entrada por ruído, nota-se uma distinção de qualidade de estimação/reconstrução no que diz respeito à manobra utilizada.

Por se tratar de um algoritmo que utiliza estratégias baseadas nos métodos do gradiente e de Gauss-Newton (em situações específicas), o algoritmo de Levenberg-Marquardt consegue tratar de forma mais adequada toda a informação proveniente da excitação da dinâmica não-linear do modelo cinemático. De fato, seu desempenho geral é melhor que o obtido com a utilização desses dois métodos isoladamente.

Tabela 6.35: Índices de desempenho para o algoritmo de LM, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{LM, C1, 0\sigma}$	$\hat{\theta}_{LM, C1, 1\sigma}$	$\hat{\theta}_{LM, C1, 2\sigma}$	$\hat{\theta}_{LM, C1, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{LM, C1, 0\sigma, \%}$	$\hat{\theta}_{LM, C1, 1\sigma, \%}$	$\hat{\theta}_{LM, C1, 2\sigma, \%}$	$\hat{\theta}_{LM, C1, 4\sigma, \%}$
I_2	65,0%	50,0%	70,0%	75,0%
I_3	205,80	200,20	213,08	206,22

Detalhamento da Tabela 6.35:

$$\left\{ \begin{array}{l} \hat{\theta}_{LM, C1, 0\sigma} = [0,49655; 0,24828; -0,37241; 0,01517; -0,01103; 0,01103]^T \\ \hat{\theta}_{LM, C1, 0\sigma, \%} = [41,379; 20,690; 31,034; 37,931; 27,586; 27,586]^T \\ \hat{\theta}_{LM, C1, 1\sigma} = [0,49655; 0,28966; -0,33103; -0,00414; -0,00966; -0,01379]^T \\ \hat{\theta}_{LM, C1, 1\sigma, \%} = [41,379; 24,138; 27,586; 10,345; 24,138; 34,483]^T \\ \hat{\theta}_{LM, C1, 2\sigma} = [0,41379; 0,34482; -0,27586; -0,01609; 0,01264; -0,01264]^T \\ \hat{\theta}_{LM, C1, 2\sigma, \%} = [34,483; 28,736; 22,988; 40,230; 31,609; 31,609]^T \\ \hat{\theta}_{LM, C1, 4\sigma} = [0,49655; 0,28966; -0,37241; -0,01103; -0,01241; -0,01793]^T \\ \hat{\theta}_{LM, C1, 4\sigma, \%} = [41,379; 24,138; 31,034; 27,586; 31,034; 44,828]^T \end{array} \right.$$

Tabela 6.36: Índices de desempenho para o algoritmo de LM, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{LM, C2, 0\sigma}$	$\hat{\theta}_{LM, C2, 1\sigma}$	$\hat{\theta}_{LM, C2, 2\sigma}$	$\hat{\theta}_{LM, C2, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{LM, C2, 0\sigma, \%}$	$\hat{\theta}_{LM, C2, 1\sigma, \%}$	$\hat{\theta}_{LM, C2, 2\sigma, \%}$	$\hat{\theta}_{LM, C2, 4\sigma, \%}$
I_2	70,0%	30,0%	30,0%	55,0%
I_3	221,20	210,00	230,02	198,24

Detalhamento da Tabela 6.36:

$$\left\{ \begin{array}{l} \hat{\theta}_{LM, C2, 0\sigma} = [0,22345; -0,14897; -0,03724; -0,01613; -0,01737; -0,01241]^T \\ \hat{\theta}_{LM, C2, 0\sigma, \%} = [18,621; 12,414; 3,103; 40,345; 43,448; 31,035]^T \\ \hat{\theta}_{LM, C2, 1\sigma} = [0,20690; 0,57931; 0,08276; 0,00000; 0,00690; -0,00828]^T \\ \hat{\theta}_{LM, C2, 1\sigma, \%} = [17,241; 48,276; 6,897; 0,000; 17,241; 20,690]^T \\ \hat{\theta}_{LM, C2, 2\sigma} = [0,20690; 0,57931; -0,16552; 0,00690; -0,01379; -0,00828]^T \\ \hat{\theta}_{LM, C2, 2\sigma, \%} = [17,241; 48,276; 13,793; 17,241; 34,483; 20,690]^T \\ \hat{\theta}_{LM, C2, 4\sigma} = [0,20690; 0,57931; -0,20690; -0,01655; 0,00552; -0,00966]^T \\ \hat{\theta}_{LM, C2, 4\sigma, \%} = [17,241; 48,276; 17,241; 41,379; 13,793; 24,138]^T \end{array} \right.$$

Tabela 6.37: Índices de desempenho para o algoritmo de LM, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{LM, C3, 0\sigma}$	$\hat{\theta}_{LM, C3, 1\sigma}$	$\hat{\theta}_{LM, C3, 2\sigma}$	$\hat{\theta}_{LM, C3, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{LM, C3, 0\sigma, \%}$	$\hat{\theta}_{LM, C3, 1\sigma, \%}$	$\hat{\theta}_{LM, C3, 2\sigma, \%}$	$\hat{\theta}_{LM, C3, 4\sigma, \%}$
I_2	45,0%	65,0%	70,0%	25,0%
I_3	215,74	217,14	210,42	211,82

Detalhamento da Tabela 6.37:

$$\left\{ \begin{array}{l} \hat{\theta}_{LM, C3, 0\sigma} = [-0,37241; 0,03724; -0,29793; -0,01365; -0,01489; -0,00869]^T \\ \hat{\theta}_{LM, C3, 0\sigma, \%} = [31,035; 3,103; 24,827; 34,138; 37,24; 21,724]^T \\ \hat{\theta}_{LM, C3, 1\sigma} = [-0,57931; 0,04138; -0,28966; -0,00828; -0,01517; -0,01655]^T \\ \hat{\theta}_{LM, C3, 1\sigma, \%} = [48,276; 3,448; 24,138; 20,690; 37,931; 41,379]^T \\ \hat{\theta}_{LM, C3, 2\sigma} = [-0,37241; 0,04138; -0,41379; -0,00414; 0,01931; 0,00276]^T \\ \hat{\theta}_{LM, C3, 2\sigma, \%} = [31,034; 3,448; 34,483; 10,345; 48,276; 6,897]^T \\ \hat{\theta}_{LM, C3, 4\sigma} = [-0,04138; 0,08276; -0,49655; -0,01241; 0,00276; -0,01517]^T \\ \hat{\theta}_{LM, C3, 4\sigma, \%} = [3,448; 6,897; 41,379; 31,034; 6,897; 37,931]^T \end{array} \right.$$

Tabela 6.38: Índices de desempenho para o algoritmo de LM, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{LM, C4, 0\sigma}$	$\hat{\theta}_{LM, C4, 1\sigma}$	$\hat{\theta}_{LM, C4, 2\sigma}$	$\hat{\theta}_{LM, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{LM, C4, 0\sigma, \%}$	$\hat{\theta}_{LM, C4, 1\sigma, \%}$	$\hat{\theta}_{LM, C4, 2\sigma, \%}$	$\hat{\theta}_{LM, C4, 4\sigma, \%}$
I_2	80,0%	65,0%	70,0%	60,0%
I_3	217,28	203,28	214,48	214,90

Detalhamento da Tabela 6.38:

$$\left\{ \begin{array}{l} \hat{\theta}_{LM, C4, 0\sigma} = [-0,14897; 0,04966; -0,14897; -0,01076; 0,00414; -0,00993]^T \\ \hat{\theta}_{LM, C4, 0\sigma, \%} = [12,414; 4,138; 12,414; 26,897; 10,345; 24,827]^T \\ \hat{\theta}_{LM, C4, 1\sigma} = [0,02483; 0,04966; -0,14897; -0,00827; -0,01076; -0,00910]^T \\ \hat{\theta}_{LM, C4, 1\sigma, \%} = [2,069; 4,138; 12,414; 20,690; 26,897; 22,759]^T \\ \hat{\theta}_{LM, C4, 2\sigma} = [-0,03104; 0,09310; -0,12414; -0,01034; -0,01448; -0,00207]^T \\ \hat{\theta}_{LM, C4, 2\sigma, \%} = [2,586; 7,759; 10,345; 25,862; 36,207; 5,173]^T \\ \hat{\theta}_{LM, C4, 4\sigma} = [0,04966; -0,02483; -0,12414; -0,01159; -0,00993; -0,01159]^T \\ \hat{\theta}_{LM, C4, 4\sigma, \%} = [4,138; 2,069; 10,345; 28,966; 24,827; 28,966]^T \end{array} \right.$$

Tabela 6.39: Índices de desempenho para o algoritmo de LM, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{LM, C5, 0\sigma}$	$\hat{\theta}_{LM, C5, 1\sigma}$	$\hat{\theta}_{LM, C5, 2\sigma}$	$\hat{\theta}_{LM, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{LM, C5, 0\sigma, \%}$	$\hat{\theta}_{LM, C5, 1\sigma, \%}$	$\hat{\theta}_{LM, C5, 2\sigma, \%}$	$\hat{\theta}_{LM, C5, 4\sigma, \%}$
I_2	75,0%	85,0%	85,0%	75,0%
I_3	358,02	363,48	363,48	358,02

Detalhamento da Tabela 6.39:

$$\left\{ \begin{array}{l} \hat{\theta}_{LM, C5, 0\sigma} = [-0,24828; 0,28966; -0,04138; -0,01655; -0,01379; -0,00414; \\ \quad 0,99379; 0,99034; 0,99793; 0,99034; 0,99862; 0,99310]^T \\ \hat{\theta}_{LM, C5, 0\sigma, \%} = [20,690; 24,138; 3,448; 41,379; 34,483; 10,345; \\ \quad 31,034; 48,276; 10,345; 48,276; 6,897; 34,483]^T \\ \hat{\theta}_{LM, C5, 1\sigma} = [-0,04552; 0,09104; -0,09104; -0,01972; 0,01063; -0,01063; \\ \quad 1,09165; 1,08937; 1,09621; 1,09165; 1,09469; 1,09317]^T \\ \hat{\theta}_{LM, C5, 1\sigma, \%} = [3,793; 7,587; 7,587; 49,311; 26,552; 26,552; \\ \quad 41,724; 53,104; 18,965; 41,724; 26,552; 34,137]^T \\ \hat{\theta}_{LM, C5, 2\sigma} = [-0,04552; 0,09104; -0,09104; -0,01972; 0,01063; -0,01063; \\ \quad 1,09165; 1,08937; 1,09621; 1,09165; 1,09469; 1,09317]^T \\ \hat{\theta}_{LM, C5, 2\sigma, \%} = [3,793; 7,587; 7,587; 49,311; 26,552; 26,552; \\ \quad 41,724; 53,104; 18,965; 41,724; 26,552; 34,137]^T \\ \hat{\theta}_{LM, C5, 4\sigma} = [-0,24828; 0,28966; -0,04138; -0,01655; -0,01379; -0,00414; \\ \quad 0,99379; 0,99034; 0,99793; 0,99034; 0,99862; 0,99310]^T \\ \hat{\theta}_{LM, C5, 4\sigma, \%} = [20,690; 24,138; 3,448; 41,379; 34,483; 10,345; \\ \quad 31,034; 48,276; 10,345; 48,276; 6,897; 34,483]^T \end{array} \right.$$

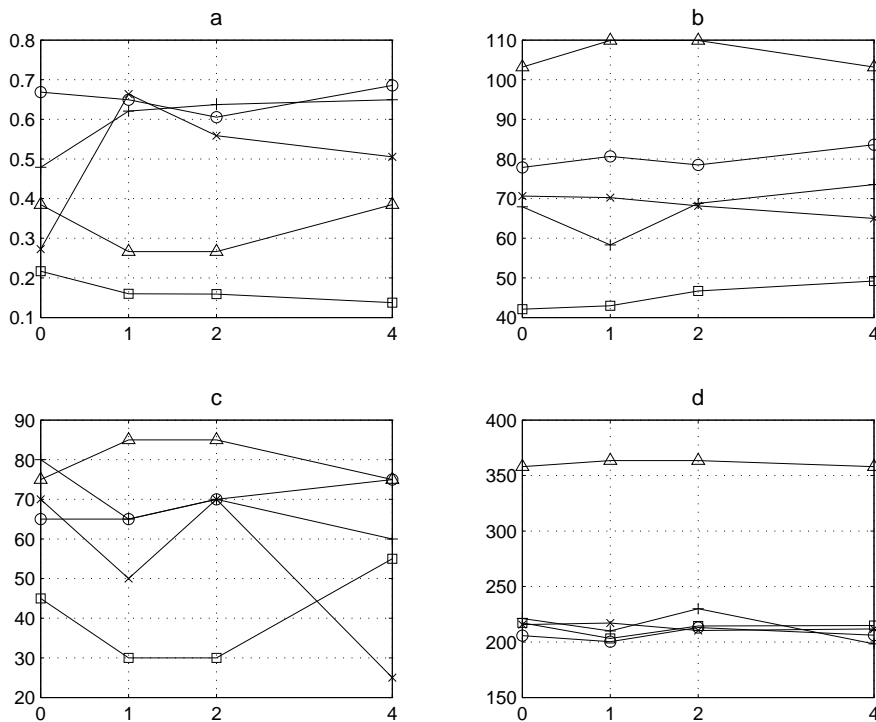


Figura 6.8: Comparação do desempenho do algoritmo de LM para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: x; 3: +; 4: \square e 5: \triangle .

6.1.5 Algoritmos genéticos

Quanto aos algoritmos genéticos, a principal conclusão consiste na possibilidade da sua divisão em dois grupos de algoritmos com desempenhos semelhantes segundo $I_{4,\%}$: o primeiro grupo contendo os algoritmos de VEGA e WBGA e o segundo com MOGA, NSGA II e SPEA. Essa divisão se deve basicamente à metodologia empregada em cada algoritmo para determinação do ponto ótimo da função objetivo.

De fato, a metodologia de um algoritmo genético consiste basicamente nos operadores utilizados. Com isso em vista, nota-se que os algoritmos de VEGA, WBGA, MOGA, NSGA II e SPEA apresentam, nessa ordem, uma evolução na utilização de estratégias representadas por operadores genéticos. Essa evolução pode ser observada a seguir.

1. VEGA: baseado na criação de subpopulações para otimização de objetivos isolados. Operadores utilizados: reprodução, cruzamento, mutação;
2. WBGA: baseado na soma ponderada das funções objetivo. Operadores utilizados: reprodução, cruzamento, mutação e ranqueamento (segundo função de partilha ou criação de subpopulações);
3. MOGA: primeiro algoritmo de otimização a utilizar critérios de não-dominância para classificação da população. Operadores utilizados: reprodução, cruzamento, mutação, ranqueamento e criação de nichos;
4. NSGA II: utiliza critérios de não-dominância para classificação da população com o uso de elitismo. Operadores utilizados: reprodução, cruzamento, mutação, ranqueamento, criação de nichos e elitismo;
5. SPEA: utiliza critérios de não-dominância para classificação da população com o uso de elitismo e mantém a diversidade da população com técnica de agrupamento. Operadores utilizados: reprodução, cruzamento, mutação, ranqueamento, criação de nichos, elitismo e *clustering*.

Além disso, com a análise comparativa dos algoritmos genéticos segundo $I_{4,\%}$, nota-se a possibilidade de classificação desses métodos segundo seu desempenho. Essa análise leva à definição do algoritmo de VEGA como o de pior desempenho seguido do WBGA e, com desempenho significativamente melhor, os algoritmos de MOGA, NSGA II e SPEA, respectivamente. Essa conclusão é conseqüência da análise direta dos gráficos b , que ilustram o comportamento da distância do vetor estimado com maior frequência, em que os

algoritmos de VEGA, WBGA, MOGA, NSGA II e SPEA apresentam distâncias percentuais médias de 50%, 40%, 20%, 17% e 13% respectivamente.

Tabela 6.40: Índices de desempenho para o algoritmo de VEGA, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{VEGA,C1,0\sigma}$	$\hat{\theta}_{VEGA,C1,1\sigma}$	$\hat{\theta}_{VEGA,C1,2\sigma}$	$\hat{\theta}_{VEGA,C1,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{VEGA,C1,0\sigma,\%}$	$\hat{\theta}_{VEGA,C1,1\sigma,\%}$	$\hat{\theta}_{VEGA,C1,2\sigma,\%}$	$\hat{\theta}_{VEGA,C1,4\sigma,\%}$
I_2	50,0%	25,0%	25,0%	35,0%
I_3	28252,00	28864,00	28558,00	29476,00

Detalhamento da Tabela 6.40:

$$\left\{ \begin{array}{l} \hat{\theta}_{VEGA,C1,0\sigma} = [-0,04138; -0,37241; -0,20690; 0,00276; 0,00828; -0,00414]^T \\ \hat{\theta}_{VEGA,C1,0\sigma,\%} = [3,448; 31,034; 17,241; 6,897; 20,690; 10,345]^T \\ \hat{\theta}_{VEGA,C1,1\sigma} = [-0,16552; -0,16552; 0,08276; -0,00690; -0,00138; 0,00828]^T \\ \hat{\theta}_{VEGA,C1,1\sigma,\%} = [13,793; 13,793; 6,897; 17,241; 3,448; 20,690]^T \\ \hat{\theta}_{VEGA,C1,2\sigma} = [-0,08276; -0,08276; -0,16552; -0,00414; 0,00828; 0,00000]^T \\ \hat{\theta}_{VEGA,C1,2\sigma,\%} = [6,897; 6,897; 13,793; 10,345; 20,690; 0,000]^T \\ \hat{\theta}_{VEGA,C1,4\sigma} = [-0,16552; 0,04138; -0,24828; -0,00966; 0,00414; 0,00138]^T \\ \hat{\theta}_{VEGA,C1,4\sigma,\%} = [13,793; 3,448; 20,690; 24,138; 10,345; 3,448]^T \end{array} \right.$$

Tabela 6.41: Índices de desempenho para o algoritmo de VEGA, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{VEGA,C2,0\sigma}$	$\hat{\theta}_{VEGA,C2,1\sigma}$	$\hat{\theta}_{VEGA,C2,2\sigma}$	$\hat{\theta}_{VEGA,C2,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{VEGA,C2,0\sigma,\%}$	$\hat{\theta}_{VEGA,C2,1\sigma,\%}$	$\hat{\theta}_{VEGA,C2,2\sigma,\%}$	$\hat{\theta}_{VEGA,C2,4\sigma,\%}$
I_2	35,0%	20,0%	30,0%	25,0%
I_3	29068,00	29476,00	29374,00	29476,00

Detalhamento da Tabela 6.41:

$$\left\{ \begin{array}{l} \hat{\theta}_{VEGA, C2, 0\sigma} = [0,08276; 0,00000; -0,08276; 0,00000; -0,00966; -0,00138]^T \\ \hat{\theta}_{VEGA, C2, 0\sigma, \%} = [6,897; 0,000; 6,897; 0,000; 24,138; 3,448]^T \\ \hat{\theta}_{VEGA, C2, 1\sigma} = [0,04138; 0,08276; -0,08276; 0,00000; 0,00414; -0,00552]^T \\ \hat{\theta}_{VEGA, C2, 1\sigma, \%} = [3,448; 6,897; 6,897; 0,000; 10,345; 13,793]^T \\ \hat{\theta}_{VEGA, C2, 2\sigma} = [0,12414; 0,24828; 0,04138; 0,00138; 0,00000; 0,00414]^T \\ \hat{\theta}_{VEGA, C2, 2\sigma, \%} = [10,345; 20,690; 3,448; 3,448; 0,000; 10,345]^T \\ \hat{\theta}_{VEGA, C2, 4\sigma} = [-0,12414; 0,08276; -0,20690; -0,00552; 0,00690; -0,00414]^T \\ \hat{\theta}_{VEGA, C2, 4\sigma, \%} = [10,345; 6,897; 17,241; 13,793; 17,241; 10,345]^T \end{array} \right.$$

Tabela 6.42: Índices de desempenho para o algoritmo de VEGA, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{VEGA, C3, 0\sigma}$	$\hat{\theta}_{VEGA, C3, 1\sigma}$	$\hat{\theta}_{VEGA, C3, 2\sigma}$	$\hat{\theta}_{VEGA, C3, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{VEGA, C3, 0\sigma, \%}$	$\hat{\theta}_{VEGA, C3, 1\sigma, \%}$	$\hat{\theta}_{VEGA, C3, 2\sigma, \%}$	$\hat{\theta}_{VEGA, C3, 4\sigma, \%}$
I_2	55,0%	65,0%	40,0%	50,0%
I_3	30772,00	30448,00	30124,00	30988,00

Detalhamento da Tabela 6.42:

$$\left\{ \begin{array}{l} \hat{\theta}_{VEGA, C3, 0\sigma} = [0,24828; -0,16552; -0,41379; 0,01103; -0,00276; 0,00000]^T \\ \hat{\theta}_{VEGA, C3, 0\sigma, \%} = [20,690; 13,793; 34,483; 27,586; 6,897; 0,000]^T \\ \hat{\theta}_{VEGA, C3, 1\sigma} = [0,08276; 0,24828; -0,45517; 0,00552; -0,01241; -0,00690]^T \\ \hat{\theta}_{VEGA, C3, 1\sigma, \%} = [6,897; 20,690; 37,931; 13,793; 31,034; 17,241]^T \\ \hat{\theta}_{VEGA, C3, 2\sigma} = [0,08276; -0,12414; -0,33103; 0,01103; -0,00690; -0,00414]^T \\ \hat{\theta}_{VEGA, C3, 2\sigma, \%} = [6,897; 10,345; 27,586; 27,586; 17,241; 10,345]^T \\ \hat{\theta}_{VEGA, C3, 4\sigma} = [0,37241; -0,12414; -0,24828; 0,00690; -0,00276; 0,00966]^T \\ \hat{\theta}_{VEGA, C3, 4\sigma, \%} = [31,034; 10,345; 20,690; 17,241; 6,897; 24,138]^T \end{array} \right.$$

Tabela 6.43: Índices de desempenho para o algoritmo de VEGA, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{VEGA, C4, 0\sigma}$	$\hat{\theta}_{VEGA, C4, 1\sigma}$	$\hat{\theta}_{VEGA, C4, 2\sigma}$	$\hat{\theta}_{VEGA, C4, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{VEGA, C4, 0\sigma, \%}$	$\hat{\theta}_{VEGA, C4, 1\sigma, \%}$	$\hat{\theta}_{VEGA, C4, 2\sigma, \%}$	$\hat{\theta}_{VEGA, C4, 4\sigma, \%}$
I_2	55,0%	60,0%	60,0%	55,0%
I_3	2908,00	2908,00	2908,00	2908,00

Detalhamento da Tabela 6.43:

$$\left\{ \begin{array}{l} \hat{\theta}_{VEGA, C4, 0\sigma} = [0,12414; 0,20690; -0,41379; 0,00138; -0,00414; 0,00552]^T \\ \hat{\theta}_{VEGA, C4, 0\sigma, \%} = [10,345; 17,241; 34,483; 3,448; 10,345; 13,793]^T \\ \hat{\theta}_{VEGA, C4, 1\sigma} = [0,12414; -0,41379; -0,45517; -0,01379; -0,00966; 0,01379]^T \\ \hat{\theta}_{VEGA, C4, 1\sigma, \%} = [10,345; 34,483; 37,931; 34,483; 24,138; 34,483]^T \\ \hat{\theta}_{VEGA, C4, 2\sigma} = [0,33103; -0,24828; 0,04138; -0,01103; 0,01517; -0,00276]^T \\ \hat{\theta}_{VEGA, C4, 2\sigma, \%} = [27,586; 20,690; 3,448; 27,586; 37,931; 6,897]^T \\ \hat{\theta}_{VEGA, C4, 4\sigma} = [-0,04138; -0,16552; -0,41379; 0,00138; -0,00966; 0,00966]^T \\ \hat{\theta}_{VEGA, C4, 4\sigma, \%} = [3,448; 13,793; 34,483; 3,448; 24,138; 24,138]^T \end{array} \right.$$

Tabela 6.44: Índices de desempenho para o algoritmo de VEGA, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{VEGA, C5, 0\sigma}$	$\hat{\theta}_{VEGA, C5, 1\sigma}$	$\hat{\theta}_{VEGA, C5, 2\sigma}$	$\hat{\theta}_{VEGA, C5, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{VEGA, C5, 0\sigma, \%}$	$\hat{\theta}_{VEGA, C5, 1\sigma, \%}$	$\hat{\theta}_{VEGA, C5, 2\sigma, \%}$	$\hat{\theta}_{VEGA, C5, 4\sigma, \%}$
I_2	80,0%	65,0%	75,0%	65,0%
I_3	2908,00	2908,00	2800,00	29692,00

Detalhamento da Tabela 6.44:

$$\left\{ \begin{array}{l} \hat{\theta}_{VEGA, C5, 0\sigma} = [0,20690; 0,33103; -0,37241; 0,01103; -0,01517; 0,00000; \\ \quad 0,99862; 0,99655; 0,99793; 0,99448; 0,99793; 0,99103]^T \\ \hat{\theta}_{VEGA, C5, 0\sigma, \%} = [17,241; 27,586; 31,034; 27,586; 37,931; 0,000; \\ \quad 6,897; 17,241; 10,345; 27,586; 10,345; 44,828]^T \\ \hat{\theta}_{VEGA, C5, 1\sigma} = [0,49655; -0,41379; -0,28966; 0,00000; 0,01517; 0,01241; \\ \quad 1,00207; 1,00759; 1,00690; 0,99655; 1,00483; 0,99517]^T \\ \hat{\theta}_{VEGA, C5, 1\sigma, \%} = [41,379; 34,483; 24,138; 0,000; 37,931; 31,034; \\ \quad 10,345; 37,931; 34,483; 17,241; 24,138; 24,138]^T \\ \hat{\theta}_{VEGA, C5, 2\sigma} = [0,49655; 0,45517; -0,24828; -0,01241; -0,01379; 0,00828; \\ \quad 1,00759; 0,99793; 0,99862; 1,00483; 0,99379; 1,00138]^T \\ \hat{\theta}_{VEGA, C5, 2\sigma, \%} = [41,379; 37,931; 20,690; 31,034; 34,483; 20,690; \\ \quad 37,931; 10,345; 6,897; 24,138; 31,034; 6,897]^T \\ \hat{\theta}_{VEGA, C5, 4\sigma} = [0,28966; -0,28966; -0,49655; -0,01379; -0,00414; 0,00966; \\ \quad 1,00690; 0,99379; 0,99586; 1,00207; 1,00552; 0,99724]^T \\ \hat{\theta}_{VEGA, C5, 4\sigma, \%} = [24,138; 24,138; 41,379; 34,483; 10,345; 24,138; \\ \quad 34,483; 31,034; 20,690; 10,345; 27,586; 13,793]^T \end{array} \right.$$

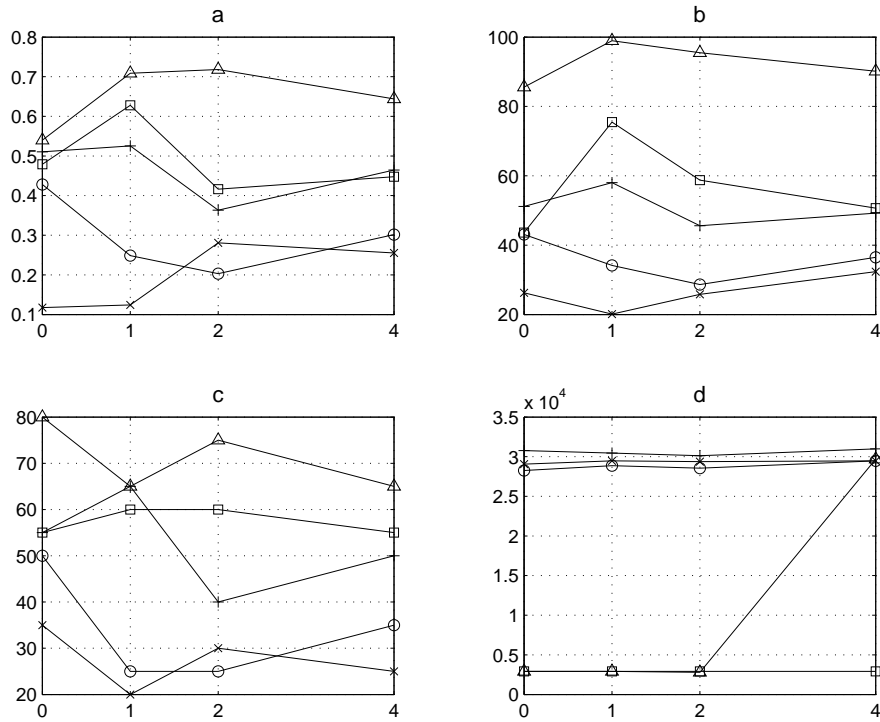


Figura 6.9: Comparação do desempenho do algoritmo de VEGA para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: ×; 3: +; 4: □ e 5: △.

Tabela 6.45: Índices de desempenho para o algoritmo de WBGA, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{WBGA, C1, 0\sigma}$	$\hat{\theta}_{WBGA, C1, 1\sigma}$	$\hat{\theta}_{WBGA, C1, 2\sigma}$	$\hat{\theta}_{WBGA, C1, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{WBGA, C1, 0\sigma, \%}$	$\hat{\theta}_{WBGA, C1, 1\sigma, \%}$	$\hat{\theta}_{WBGA, C1, 2\sigma, \%}$	$\hat{\theta}_{WBGA, C1, 4\sigma, \%}$
I_2	65,0%	55,0%	50,0%	30,0%
I_3	19900,00	19900,00	19900,00	19900,00

Detalhamento da Tabela 6.45:

$$\left\{ \begin{array}{l} \hat{\theta}_{WBGA, C1, 0\sigma} = [0,04138; -0,49655; 0,00000; -0,00552; 0,00552; -0,00690]^T \\ \hat{\theta}_{WBGA, C1, 0\sigma, \%} = [3,448; 41,379; 0,000; 13,793; 13,793; 17,241]^T \\ \hat{\theta}_{WBGA, C1, 1\sigma} = [-0,41379; 0,20690; -0,24828; -0,00414; -0,00276; -0,00276]^T \\ \hat{\theta}_{WBGA, C1, 1\sigma, \%} = [34,483; 17,241; 20,690; 10,345; 6,897; 6,897]^T \\ \hat{\theta}_{WBGA, C1, 2\sigma} = [-0,37241; -0,08276; 0,08276; -0,00552; 0,00276; -0,00414]^T \\ \hat{\theta}_{WBGA, C1, 2\sigma, \%} = [31,034; 6,897; 6,897; 13,793; 6,897; 10,345]^T \\ \hat{\theta}_{WBGA, C1, 4\sigma} = [0,00000; 0,04138; 0,20690; 0,00276; -0,00138; -0,00828]^T \\ \hat{\theta}_{WBGA, C1, 4\sigma, \%} = [0,000; 3,448; 17,241; 6,897; 3,448; 20,690]^T \end{array} \right.$$

Tabela 6.46: Índices de desempenho para o algoritmo de WBGA, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{WBGA, C2, 0\sigma}$	$\hat{\theta}_{WBGA, C2, 1\sigma}$	$\hat{\theta}_{WBGA, C2, 2\sigma}$	$\hat{\theta}_{WBGA, C2, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{WBGA, C2, 0\sigma, \%}$	$\hat{\theta}_{WBGA, C2, 1\sigma, \%}$	$\hat{\theta}_{WBGA, C2, 2\sigma, \%}$	$\hat{\theta}_{WBGA, C2, 4\sigma, \%}$
I_2	20,0%	60,0%	60,0%	55,0%
I_3	19900,00	19900,00	19900,00	19900,00

Detalhamento da Tabela 6.46:

$$\left\{ \begin{array}{l} \hat{\theta}_{WBGA, C2, 0\sigma} = [-0,16552; 0,04138; -0,16552; -0,00414; -0,00414; 0,00414]^T \\ \hat{\theta}_{WBGA, C2, 0\sigma, \%} = [13,793; 3,448; 13,793; 10,345; 10,345; 10,345]^T \\ \hat{\theta}_{WBGA, C2, 1\sigma} = [-0,41379; 0,41379; -0,16552; -0,00276; 0,00414; 0,00552]^T \\ \hat{\theta}_{WBGA, C2, 1\sigma, \%} = [34,483; 34,483; 13,793; 6,897; 10,345; 13,793]^T \\ \hat{\theta}_{WBGA, C2, 2\sigma} = [0,41379; 0,20690; 0,45517; -0,00138; -0,00276; 0,00276]^T \\ \hat{\theta}_{WBGA, C2, 2\sigma, \%} = [34,483; 17,241; 37,931; 3,448; 6,897; 6,897]^T \\ \hat{\theta}_{WBGA, C2, 4\sigma} = [0,28966; 0,12414; 0,37241; 0,00552; -0,00690; 0,00276]^T \\ \hat{\theta}_{WBGA, C2, 4\sigma, \%} = [24,138; 10,345; 31,034; 13,793; 17,241; 6,897]^T \end{array} \right.$$

Tabela 6.47: Índices de desempenho para o algoritmo de WBGA, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{WBGA, C3, 0\sigma}$	$\hat{\theta}_{WBGA, C3, 1\sigma}$	$\hat{\theta}_{WBGA, C3, 2\sigma}$	$\hat{\theta}_{WBGA, C3, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{WBGA, C3, 0\sigma, \%}$	$\hat{\theta}_{WBGA, C3, 1\sigma, \%}$	$\hat{\theta}_{WBGA, C3, 2\sigma, \%}$	$\hat{\theta}_{WBGA, C3, 4\sigma, \%}$
I_2	55,0%	20,0%	35,0%	40,0%
I_3	19900,00	19900,00	19900,00	19900,00

Detalhamento da Tabela 6.47:

$$\left\{ \begin{array}{l} \hat{\theta}_{WBGA, C3, 0\sigma} = [0,20690; 0,41379; -0,04138; 0,00000; -0,00138; 0,00276]^T \\ \hat{\theta}_{WBGA, C3, 0\sigma, \%} = [17,241; 34,483; 3,448; 0,000; 3,448; 6,897]^T \\ \hat{\theta}_{WBGA, C3, 1\sigma} = [0,00000; 0,12414; -0,20690; 0,00138; -0,00414; 0,00000]^T \\ \hat{\theta}_{WBGA, C3, 1\sigma, \%} = [0,000; 10,345; 17,241; 3,448; 10,345; 0,000]^T \\ \hat{\theta}_{WBGA, C3, 2\sigma} = [-0,04138; 0,28966; 0,16552; 0,00552; 0,00552; 0,00552]^T \\ \hat{\theta}_{WBGA, C3, 2\sigma, \%} = [3,448; 24,138; 13,793; 13,793; 13,793; 13,793]^T \\ \hat{\theta}_{WBGA, C3, 4\sigma} = [-0,28966; 0,33103; -0,12414; 0,00552; -0,00276; 0,00552]^T \\ \hat{\theta}_{WBGA, C3, 4\sigma, \%} = [24,138; 27,586; 10,345; 13,793; 6,897; 13,793]^T \end{array} \right.$$

Tabela 6.48: Índices de desempenho para o algoritmo de WBGA, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{WBGA, C4, 0\sigma}$	$\hat{\theta}_{WBGA, C4, 1\sigma}$	$\hat{\theta}_{WBGA, C4, 2\sigma}$	$\hat{\theta}_{WBGA, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{WBGA, C4, 0\sigma, \%}$	$\hat{\theta}_{WBGA, C4, 1\sigma, \%}$	$\hat{\theta}_{WBGA, C4, 2\sigma, \%}$	$\hat{\theta}_{WBGA, C4, 4\sigma, \%}$
I_2	55,0%	40,0%	60,0%	35,0%
I_3	1900,00	1900,00	1900,00	1900,00

Detalhamento da Tabela 6.48:

$$\left\{ \begin{array}{l} \hat{\theta}_{WBGA, C4, 0\sigma} = [0,33103; 0,41379; -0,04138; -0,00276; -0,00414; 0,00000]^T \\ \hat{\theta}_{WBGA, C4, 0\sigma, \%} = [27,586; 34,483; 3,448; 6,897; 10,345; 0,000]^T \\ \hat{\theta}_{WBGA, C4, 1\sigma} = [-0,20690; 0,28966; 0,16552; 0,00276; -0,00138; 0,00138]^T \\ \hat{\theta}_{WBGA, C4, 1\sigma, \%} = [17,241; 24,138; 13,793; 6,897; 3,448; 3,448]^T \\ \hat{\theta}_{WBGA, C4, 2\sigma} = [0,28966; 0,41379; -0,16552; -0,00414; 0,00276; 0,00276]^T \\ \hat{\theta}_{WBGA, C4, 2\sigma, \%} = [24,138; 34,483; 13,793; 10,345; 6,897; 6,897]^T \\ \hat{\theta}_{WBGA, C4, 4\sigma} = [0,12414; 0,28966; 0,12414; 0,00276; -0,00138; 0,00138]^T \\ \hat{\theta}_{WBGA, C4, 4\sigma, \%} = [10,345; 24,138; 10,345; 6,897; 3,448; 3,448]^T \end{array} \right.$$

Tabela 6.49: Índices de desempenho para o algoritmo de WBGA, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{WBGA, C5, 0\sigma}$	$\hat{\theta}_{WBGA, C5, 1\sigma}$	$\hat{\theta}_{WBGA, C5, 2\sigma}$	$\hat{\theta}_{WBGA, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{WBGA, C5, 0\sigma, \%}$	$\hat{\theta}_{WBGA, C5, 1\sigma, \%}$	$\hat{\theta}_{WBGA, C5, 2\sigma, \%}$	$\hat{\theta}_{WBGA, C5, 4\sigma, \%}$
I_2	30,0%	10,0%	55,0%	45,0%
I_3	1900,00	1900,00	1900,00	1900,00

Detalhamento da Tabela 6.49:

$$\left\{ \begin{array}{l} \hat{\theta}_{WBGA, C5, 0\sigma} = [0,00000; 0,00000; -0,16552; 0,00276; -0,00690; 0,00138; \\ \quad 0,99586; 1,00345; 1,00000; 0,99931; 0,99862; 1,00345]^T \\ \hat{\theta}_{WBGA, C5, 0\sigma, \%} = [0,000; 0,000; 13,793; 6,897; 17,241; 3,448; \\ \quad 20,690; 17,241; 0,000; 3,448; 6,897; 17,241]^T \\ \hat{\theta}_{WBGA, C5, 1\sigma} = [-0,08276; 0,08276; 0,08276; 0,00414; -0,00276; 0,00276; \\ \quad 0,99931; 1,00000; 1,00138; 1,00000; 0,99793; 0,99793]^T \\ \hat{\theta}_{WBGA, C5, 1\sigma, \%} = [6,897; 6,897; 6,897; 10,345; 6,897; 6,897; \\ \quad 3,448; 0,000; 6,897; 0,000; 10,345; 10,345]^T \\ \hat{\theta}_{WBGA, C5, 2\sigma} = [0,24828; 0,41379; -0,20690; 0,00000; 0,00552; 0,00138; \\ \quad 1,00207; 1,00276; 1,00069; 0,99931; 0,99724; 1,00276]^T \\ \hat{\theta}_{WBGA, C5, 2\sigma, \%} = [20,690; 34,483; 17,241; 0,000; 13,793; 3,448; \\ \quad 10,345; 13,793; 3,448; 3,448; 13,793; 13,793]^T \\ \hat{\theta}_{WBGA, C5, 4\sigma} = [0,28966; 0,33103; -0,08276; 0,00138; 0,00138; 0,00138; \\ \quad 0,99655; 0,99862; 1,00138; 0,99724; 0,99931; 0,99931]^T \\ \hat{\theta}_{WBGA, C5, 4\sigma, \%} = [24,138; 27,586; 6,897; 3,448; 3,448; 3,448; \\ \quad 17,241; 6,897; 6,897; 13,793; 3,448; 3,448]^T \end{array} \right.$$

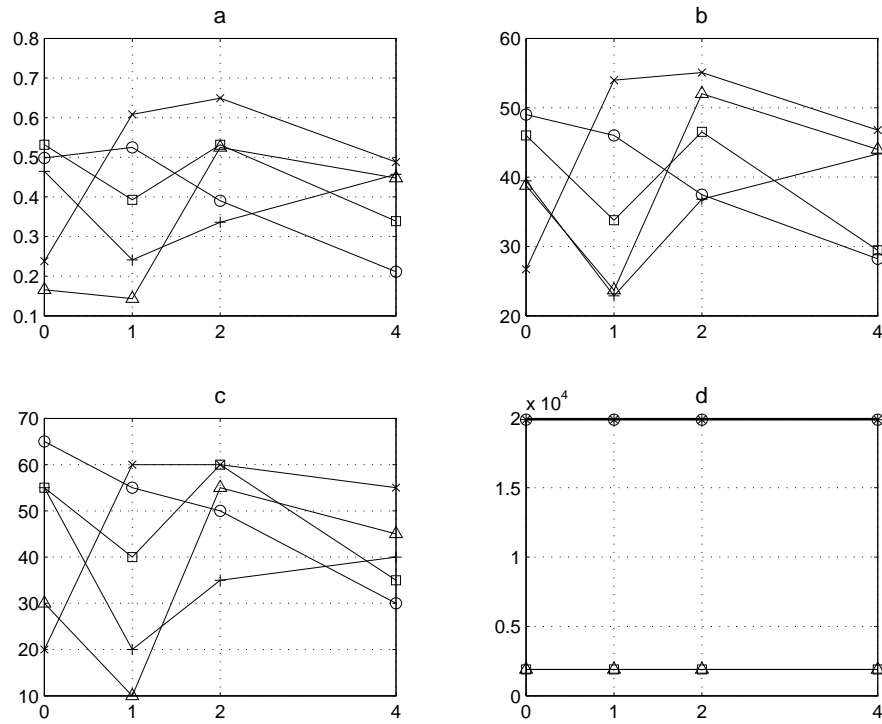


Figura 6.10: Comparação do desempenho do algoritmo de WBGA para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: ×; 3: +; 4: □ e 5: △.

Tabela 6.50: Índices de desempenho para o algoritmo de MOGA, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{MOGA,C1,0\sigma}$	$\hat{\theta}_{MOGA,C1,1\sigma}$	$\hat{\theta}_{MOGA,C1,2\sigma}$	$\hat{\theta}_{MOGA,C1,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{MOGA,C1,0\sigma,\%}$	$\hat{\theta}_{MOGA,C1,1\sigma,\%}$	$\hat{\theta}_{MOGA,C1,2\sigma,\%}$	$\hat{\theta}_{MOGA,C1,4\sigma,\%}$
I_2	10,0%	30,0%	5,0%	5,0%
I_3	15000,00	15000,00	15000,00	15000,00

Detalhamento da Tabela 6.50:

$$\left\{ \begin{array}{l} \hat{\theta}_{MOGA, C1, 0\sigma} = [0,16552; 0,00000; 0,00000; 0,00138; 0,00000; 0,00276]^T \\ \hat{\theta}_{MOGA, C1, 0\sigma, \%} = [13,793; 0,000; 0,000; 3,448; 0,000; 6,897]^T \\ \hat{\theta}_{MOGA, C1, 1\sigma} = [0,12414; -0,24828; -0,20690; 0,00414; 0,00276; -0,00552]^T \\ \hat{\theta}_{MOGA, C1, 1\sigma, \%} = [10,345; 20,690; 17,241; 10,345; 6,897; 13,793]^T \\ \hat{\theta}_{MOGA, C1, 2\sigma} = [0,04138; -0,12414; -0,08276; -0,00138; -0,00276; -0,00276]^T \\ \hat{\theta}_{MOGA, C1, 2\sigma, \%} = [3,448; 10,345; 6,897; 3,448; 6,897; 6,897]^T \\ \hat{\theta}_{MOGA, C1, 4\sigma} = [0,12414; -0,12414; -0,08276; 0,00276; 0,00000; -0,00414]^T \\ \hat{\theta}_{MOGA, C1, 4\sigma, \%} = [10,345; 10,345; 6,897; 6,897; 0,000; 10,345]^T \end{array} \right.$$

Tabela 6.51: Índices de desempenho para o algoritmo de MOGA, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{MOGA, C2, 0\sigma}$	$\hat{\theta}_{MOGA, C2, 1\sigma}$	$\hat{\theta}_{MOGA, C2, 2\sigma}$	$\hat{\theta}_{MOGA, C2, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{MOGA, C2, 0\sigma, \%}$	$\hat{\theta}_{MOGA, C2, 1\sigma, \%}$	$\hat{\theta}_{MOGA, C2, 2\sigma, \%}$	$\hat{\theta}_{MOGA, C2, 4\sigma, \%}$
I_2	10,0%	15,0%	10,0%	40,0%
I_3	15000,00	15000,00	15000,00	15000,00

Detalhamento da Tabela 6.51:

$$\left\{ \begin{array}{l} \hat{\theta}_{MOGA, C2, 0\sigma} = [0,08276; 0,00000; -0,04138; 0,00414; 0,00138; -0,00276]^T \\ \hat{\theta}_{MOGA, C2, 0\sigma, \%} = [6,897; 0,000; 3,448; 10,345; 3,448; 6,897]^T \\ \hat{\theta}_{MOGA, C2, 1\sigma} = [0,16552; -0,12414; -0,08276; -0,00414; -0,00414; -0,00414]^T \\ \hat{\theta}_{MOGA, C2, 1\sigma, \%} = [13,793; 10,345; 6,897; 10,345; 10,345; 10,345]^T \\ \hat{\theta}_{MOGA, C2, 2\sigma} = [0,04138; -0,08276; -0,12414; 0,00414; -0,00414; -0,00138]^T \\ \hat{\theta}_{MOGA, C2, 2\sigma, \%} = [3,448; 6,897; 10,345; 10,345; 10,345; 3,448]^T \\ \hat{\theta}_{MOGA, C2, 4\sigma} = [0,16552; -0,12414; -0,16552; -0,00966; 0,00138; 0,00138]^T \\ \hat{\theta}_{MOGA, C2, 4\sigma, \%} = [13,793; 10,345; 13,793; 24,138; 3,448; 3,448]^T \end{array} \right.$$

Tabela 6.52: Índices de desempenho para o algoritmo de MOGA, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{MOGA, C3, 0\sigma}$	$\hat{\theta}_{MOGA, C3, 1\sigma}$	$\hat{\theta}_{MOGA, C3, 2\sigma}$	$\hat{\theta}_{MOGA, C3, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{MOGA, C3, 0\sigma, \%}$	$\hat{\theta}_{MOGA, C3, 1\sigma, \%}$	$\hat{\theta}_{MOGA, C3, 2\sigma, \%}$	$\hat{\theta}_{MOGA, C3, 4\sigma, \%}$
I_2	5,0%	5,0%	0,0%	5,0%
I_3	15000,00	15000,00	15000,00	15000,00

Detalhamento da Tabela 6.52:

$$\left\{ \begin{array}{l} \hat{\theta}_{MOGA, C3, 0\sigma} = [0,04138; -0,04138; -0,12414; 0,00000; 0,00000; -0,00138]^T \\ \hat{\theta}_{MOGA, C3, 0\sigma, \%} = [3,448; 3,448; 10,345; 0,000; 0,000; 3,448]^T \\ \hat{\theta}_{MOGA, C3, 1\sigma} = [0,04138; 0,00000; -0,04138; 0,00414; 0,00000; 0,00000]^T \\ \hat{\theta}_{MOGA, C3, 1\sigma, \%} = [3,448; 0,000; 3,448; 10,345; 0,000; 0,000]^T \\ \hat{\theta}_{MOGA, C3, 2\sigma} = [0,04138; -0,04138; -0,08276; -0,00138; 0,00138; 0,00276]^T \\ \hat{\theta}_{MOGA, C3, 2\sigma, \%} = [3,448; 3,448; 6,897; 3,448; 3,448; 6,897]^T \\ \hat{\theta}_{MOGA, C3, 4\sigma} = [0,12414; -0,12414; -0,08276; 0,00414; -0,00138; -0,00138]^T \\ \hat{\theta}_{MOGA, C3, 4\sigma, \%} = [10,345; 10,345; 6,897; 10,345; 3,448; 3,448]^T \end{array} \right.$$

Tabela 6.53: Índices de desempenho para o algoritmo de MOGA, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{MOGA, C4, 0\sigma}$	$\hat{\theta}_{MOGA, C4, 1\sigma}$	$\hat{\theta}_{MOGA, C4, 2\sigma}$	$\hat{\theta}_{MOGA, C4, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{MOGA, C4, 0\sigma, \%}$	$\hat{\theta}_{MOGA, C4, 1\sigma, \%}$	$\hat{\theta}_{MOGA, C4, 2\sigma, \%}$	$\hat{\theta}_{MOGA, C4, 4\sigma, \%}$
I_2	10,0%	15,0%	35,0%	15,0%
I_3	15000,00	15000,00	15000,00	15000,00

Detalhamento da Tabela 6.53:

$$\left\{ \begin{array}{l} \hat{\theta}_{MOGA, C4, 0\sigma} = [0,04138; 0,08276; -0,12414; -0,00138; -0,00138; 0,00000]^T \\ \hat{\theta}_{MOGA, C4, 0\sigma, \%} = [3,448; 6,897; 10,345; 3,448; 3,448; 0,000]^T \\ \hat{\theta}_{MOGA, C4, 1\sigma} = [0,16552; -0,04138; -0,08276; 0,00138; 0,00000; -0,00138]^T \\ \hat{\theta}_{MOGA, C4, 1\sigma, \%} = [13,793; 3,448; 6,897; 3,448; 0,000; 3,448]^T \\ \hat{\theta}_{MOGA, C4, 2\sigma} = [0,28966; 0,00000; 0,08276; -0,00138; -0,00138; -0,00276]^T \\ \hat{\theta}_{MOGA, C4, 2\sigma, \%} = [24,138; 0,000; 6,897; 3,448; 3,448; 6,897]^T \\ \hat{\theta}_{MOGA, C4, 4\sigma} = [0,16552; -0,04138; -0,08276; -0,00276; 0,00000; 0,00276]^T \\ \hat{\theta}_{MOGA, C4, 4\sigma, \%} = [13,793; 3,448; 6,897; 6,897; 0,000; 6,897]^T \end{array} \right.$$

Tabela 6.54: Índices de desempenho para o algoritmo de MOGA, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{MOGA, C5, 0\sigma}$	$\hat{\theta}_{MOGA, C5, 1\sigma}$	$\hat{\theta}_{MOGA, C5, 2\sigma}$	$\hat{\theta}_{MOGA, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{MOGA, C5, 0\sigma, \%}$	$\hat{\theta}_{MOGA, C5, 1\sigma, \%}$	$\hat{\theta}_{MOGA, C5, 2\sigma, \%}$	$\hat{\theta}_{MOGA, C5, 4\sigma, \%}$
I_2	10,0%	0,0%	15,0%	20,0%
I_3	15000,00	15000,00	15000,00	15000,00

Detalhamento da Tabela 6.54:

$$\left\{ \begin{array}{l}
 \hat{\theta}_{MOGA, C5, 0\sigma} = [0,12414; 0,04138; 0,00000; 0,00000; 0,00276; -0,00414; \\
 \quad 1,00000; 0,99931; 1,00138; 1,00276; 1,00069; 1,00069]^T \\
 \hat{\theta}_{MOGA, C5, 0\sigma, \%} = [10,345; 3,448; 0,000; 0,000; 6,897; 10,345; \\
 \quad 0,000; 3,448; 6,897; 13,793; 3,448; 3,448]^T \\
 \hat{\theta}_{MOGA, C5, 1\sigma} = [0,04138; 0,08276; 0,00000; 0,00000; 0,00000; -0,00138; \\
 \quad 0,99931; 1,00000; 0,99931; 1,00000; 1,00000; 0,99931]^T \\
 \hat{\theta}_{MOGA, C5, 1\sigma, \%} = [3,448; 6,897; 0,000; 0,000; 0,000; 3,448; \\
 \quad 3,448; 0,000; 3,448; 0,000; 0,000; 3,448]^T \\
 \hat{\theta}_{MOGA, C5, 2\sigma} = [0,04138; -0,04138; -0,12414; -0,00414; 0,00000; 0,00000; \\
 \quad 1,00207; 0,99931; 0,99724; 1,00207; 0,99931; 0,99793]^T \\
 \hat{\theta}_{MOGA, C5, 2\sigma, \%} = [3,448; 3,448; 10,345; 10,345; 0,000; 0,000; \\
 \quad 10,345; 3,448; 13,793; 10,345; 3,448; 10,345]^T \\
 \hat{\theta}_{MOGA, C5, 4\sigma} = [0,12414; 0,00000; -0,08276; 0,00000; 0,00552; 0,00552; \\
 \quad 0,99931; 1,00207; 0,99793; 0,99793; 1,00069; 1,00138]^T \\
 \hat{\theta}_{MOGA, C5, 4\sigma, \%} = [10,345; 0,000; 6,897; 0,000; 13,793; 13,793; \\
 \quad 3,448; 10,345; 10,345; 10,345; 3,448; 6,897]^T
 \end{array} \right.$$

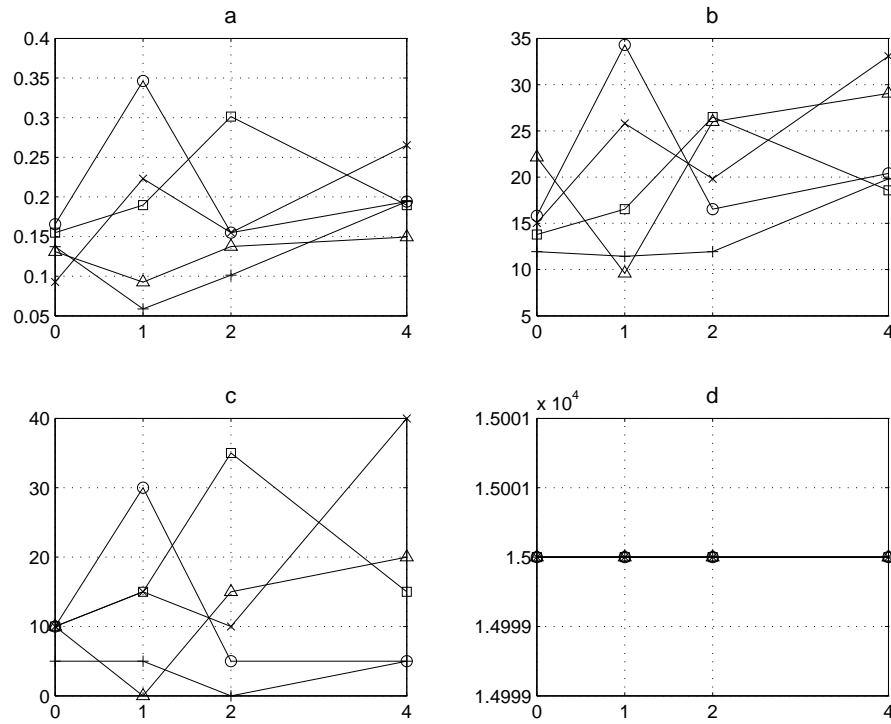


Figura 6.11: Comparação do desempenho do algoritmo de MOGA para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: x; 3: +; 4: \square e 5: \triangle .

Tabela 6.55: Índices de desempenho para o algoritmo de NSGA, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NSGA,C1,0\sigma}$	$\hat{\theta}_{NSGA,C1,1\sigma}$	$\hat{\theta}_{NSGA,C1,2\sigma}$	$\hat{\theta}_{NSGA,C1,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{NSGA,C1,0\sigma,\%}$	$\hat{\theta}_{NSGA,C1,1\sigma,\%}$	$\hat{\theta}_{NSGA,C1,2\sigma,\%}$	$\hat{\theta}_{NSGA,C1,4\sigma,\%}$
I_2	10,0%	0,0%	20,0%	0,0%
I_3	18000,00	18000,00	18000,00	18000,00

Detalhamento da Tabela 6.55:

$$\left\{ \begin{array}{l} \hat{\theta}_{NSGA, C1, 0\sigma} = [-0,12414; 0,12414; 0,08276; 0,00414; 0,00138; -0,00138]^T \\ \hat{\theta}_{NSGA, C1, 0\sigma, \%} = [10,345; 10,345; 6,897; 10,345; 3,448; 3,448]^T \\ \hat{\theta}_{NSGA, C1, 1\sigma} = [-0,04138; 0,00000; 0,08276; 0,00138; -0,00138; -0,00276]^T \\ \hat{\theta}_{NSGA, C1, 1\sigma, \%} = [3,448; 0,000; 6,897; 3,448; 3,448; 6,897]^T \\ \hat{\theta}_{NSGA, C1, 2\sigma} = [-0,20690; 0,04138; 0,00000; -0,00138; 0,00138; -0,00414]^T \\ \hat{\theta}_{NSGA, C1, 2\sigma, \%} = [17,241; 3,448; 0,000; 3,448; 3,448; 10,345]^T \\ \hat{\theta}_{NSGA, C1, 4\sigma} = [0,04138; -0,04138; 0,08276; -0,00276; -0,00138; 0,00276]^T \\ \hat{\theta}_{NSGA, C1, 4\sigma, \%} = [3,448; 3,448; 6,897; 6,897; 3,448; 6,897]^T \end{array} \right.$$

Tabela 6.56: Índices de desempenho para o algoritmo de NSGA, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NSGA, C2, 0\sigma}$	$\hat{\theta}_{NSGA, C2, 1\sigma}$	$\hat{\theta}_{NSGA, C2, 2\sigma}$	$\hat{\theta}_{NSGA, C2, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{NSGA, C2, 0\sigma, \%}$	$\hat{\theta}_{NSGA, C2, 1\sigma, \%}$	$\hat{\theta}_{NSGA, C2, 2\sigma, \%}$	$\hat{\theta}_{NSGA, C2, 4\sigma, \%}$
I_2	15,0%	15,0%	5,0%	20,0%
I_3	18000,00	18000,00	18000,00	18000,00

Detalhamento da Tabela 6.56:

$$\left\{ \begin{array}{l} \hat{\theta}_{NSGA, C2, 0\sigma} = [0,00000; -0,08276; -0,04138; -0,00552; 0,00138; 0,00276]^T \\ \hat{\theta}_{NSGA, C2, 0\sigma, \%} = [0,000; 6,897; 3,448; 13,793; 3,448; 6,897]^T \\ \hat{\theta}_{NSGA, C2, 1\sigma} = [0,08276; -0,16552; 0,00000; 0,00138; -0,00276; 0,00000]^T \\ \hat{\theta}_{NSGA, C2, 1\sigma, \%} = [6,897; 13,793; 0,000; 3,448; 6,897; 0,000]^T \\ \hat{\theta}_{NSGA, C2, 2\sigma} = [0,00000; 0,08276; 0,00000; -0,00138; 0,00000; -0,00276]^T \\ \hat{\theta}_{NSGA, C2, 2\sigma, \%} = [0,000; 6,897; 0,000; 3,448; 0,000; 6,897]^T \\ \hat{\theta}_{NSGA, C2, 4\sigma} = [-0,20690; 0,12414; 0,00000; -0,00138; -0,00414; 0,00138]^T \\ \hat{\theta}_{NSGA, C2, 4\sigma, \%} = [17,241; 10,345; 0,000; 3,448; 10,345; 3,448]^T \end{array} \right.$$

Tabela 6.57: Índices de desempenho para o algoritmo de NSGA, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NSGA, C3, 0\sigma}$	$\hat{\theta}_{NSGA, C3, 1\sigma}$	$\hat{\theta}_{NSGA, C3, 2\sigma}$	$\hat{\theta}_{NSGA, C3, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{NSGA, C3, 0\sigma, \%}$	$\hat{\theta}_{NSGA, C3, 1\sigma, \%}$	$\hat{\theta}_{NSGA, C3, 2\sigma, \%}$	$\hat{\theta}_{NSGA, C3, 4\sigma, \%}$
I_2	15,0%	10,0%	20,0%	5,0%
I_3	18000,00	18000,00	18000,00	18000,00

Detalhamento da Tabela 6.57:

$$\left\{ \begin{array}{l} \hat{\theta}_{NSGA, C3, 0\sigma} = [-0,04138; 0,08276; 0,12414; 0,00552; -0,00138; 0,00138]^T \\ \hat{\theta}_{NSGA, C3, 0\sigma, \%} = [3,448; 6,897; 10,345; 13,793; 3,448; 3,448]^T \\ \hat{\theta}_{NSGA, C3, 1\sigma} = [-0,04138; -0,12414; -0,12414; 0,00000; 0,00276; 0,00138]^T \\ \hat{\theta}_{NSGA, C3, 1\sigma, \%} = [3,448; 10,345; 10,345; 0,000; 6,897; 3,448]^T \\ \hat{\theta}_{NSGA, C3, 2\sigma} = [0,04138; -0,12414; 0,20690; 0,00414; 0,00000; -0,00276]^T \\ \hat{\theta}_{NSGA, C3, 2\sigma, \%} = [3,448; 10,345; 17,241; 10,345; 0,000; 6,897]^T \\ \hat{\theta}_{NSGA, C3, 4\sigma} = [-0,08276; -0,04138; 0,04138; 0,00000; 0,00276; -0,00138]^T \\ \hat{\theta}_{NSGA, C3, 4\sigma, \%} = [6,897; 3,448; 3,448; 0,000; 6,897; 3,448]^T \end{array} \right.$$

Tabela 6.58: Índices de desempenho para o algoritmo de NSGA, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NSGA, C4, 0\sigma}$	$\hat{\theta}_{NSGA, C4, 1\sigma}$	$\hat{\theta}_{NSGA, C4, 2\sigma}$	$\hat{\theta}_{NSGA, C4, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{NSGA, C4, 0\sigma, \%}$	$\hat{\theta}_{NSGA, C4, 1\sigma, \%}$	$\hat{\theta}_{NSGA, C4, 2\sigma, \%}$	$\hat{\theta}_{NSGA, C4, 4\sigma, \%}$
I_2	5,0%	10,0%	15,0%	5,0%
I_3	18000,00	18000,00	18000,00	18000,00

Detalhamento da Tabela 6.58:

$$\left\{ \begin{array}{l} \hat{\theta}_{NSGA, C4, 0\sigma} = [0,04138; 0,08276; 0,04138; 0,00138; -0,00138; -0,00276]^T \\ \hat{\theta}_{NSGA, C4, 0\sigma, \%} = [3,448; 6,897; 3,448; 3,448; 3,448; 6,897]^T \\ \hat{\theta}_{NSGA, C4, 1\sigma} = [0,12414; -0,04138; -0,08276; -0,00276; 0,00000; -0,00414]^T \\ \hat{\theta}_{NSGA, C4, 1\sigma, \%} = [10,345; 3,448; 6,897; 6,897; 0,000; 10,345]^T \\ \hat{\theta}_{NSGA, C4, 2\sigma} = [0,04138; 0,04138; -0,04138; 0,00276; 0,00552; 0,00000]^T \\ \hat{\theta}_{NSGA, C4, 2\sigma, \%} = [3,448; 3,448; 3,448; 6,897; 13,793; 0,000]^T \\ \hat{\theta}_{NSGA, C4, 4\sigma} = [-0,04138; -0,04138; 0,12414; 0,00000; -0,00276; 0,00138]^T \\ \hat{\theta}_{NSGA, C4, 4\sigma, \%} = [3,448; 3,448; 10,345; 0,000; 6,897; 3,448]^T \end{array} \right.$$

Tabela 6.59: Índices de desempenho para o algoritmo de NSGA, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{NSGA, C5, 0\sigma}$	$\hat{\theta}_{NSGA, C5, 1\sigma}$	$\hat{\theta}_{NSGA, C5, 2\sigma}$	$\hat{\theta}_{NSGA, C5, 4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{NSGA, C5, 0\sigma, \%}$	$\hat{\theta}_{NSGA, C5, 1\sigma, \%}$	$\hat{\theta}_{NSGA, C5, 2\sigma, \%}$	$\hat{\theta}_{NSGA, C5, 4\sigma, \%}$
I_2	15,0%	5,0%	0,0%	25,0%
I_3	18000,00	18000,00	18000,00	18000,00

Detalhamento da Tabela 6.59:

$$\left\{ \begin{array}{l}
 \hat{\theta}_{NSGA, C5, 0\sigma} = [-0,08276; 0,04138; -0,12414; -0,00414; 0,00138; -0,00276; \\
 \quad 1,00138; 1,00069; 1,00000; 0,99724; 0,99931; 0,99862]^T \\
 \hat{\theta}_{NSGA, C5, 0\sigma, \%} = [6,897; 3,448; 10,345; 10,345; 3,448; 6,897; \\
 \quad 6,897; 3,448; 0,000; 13,793; 3,448; 6,897]^T \\
 \hat{\theta}_{NSGA, C5, 1\sigma} = [-0,08276; -0,04138; 0,00000; -0,00138; 0,00000; -0,00138; \\
 \quad 0,99862; 0,99862; 1,00000; 1,00138; 0,99862; 1,00138]^T \\
 \hat{\theta}_{NSGA, C5, 1\sigma, \%} = [6,897; 3,448; 0,000; 3,448; 0,000; 3,448; \\
 \quad 6,897; 6,897; 0,000; 6,897; 6,897; 6,897]^T \\
 \hat{\theta}_{NSGA, C5, 2\sigma} = [0,04138; -0,04138; 0,08276; -0,00138; 0,00000; 0,00138; \\
 \quad 1,00000; 1,00000; 0,99931; 1,00000; 0,99862; 1,00138]^T \\
 \hat{\theta}_{NSGA, C5, 2\sigma, \%} = [3,448; 3,448; 6,897; 3,448; 0,000; 3,448; \\
 \quad 0,000; 0,000; 3,448; 0,000; 6,897; 6,897]^T \\
 \hat{\theta}_{NSGA, C5, 4\sigma} = [0,00000; 0,00000; 0,00000; 0,00138; -0,00138; 0,00414; \\
 \quad 0,99793; 1,00276; 0,99931; 1,00000; 0,99931; 0,99655]^T \\
 \hat{\theta}_{NSGA, C5, 4\sigma, \%} = [0,000; 0,000; 0,000; 3,448; 3,448; 10,345; \\
 \quad 10,345; 13,793; 3,448; 0,000; 3,448; 17,241]^T
 \end{array} \right.$$

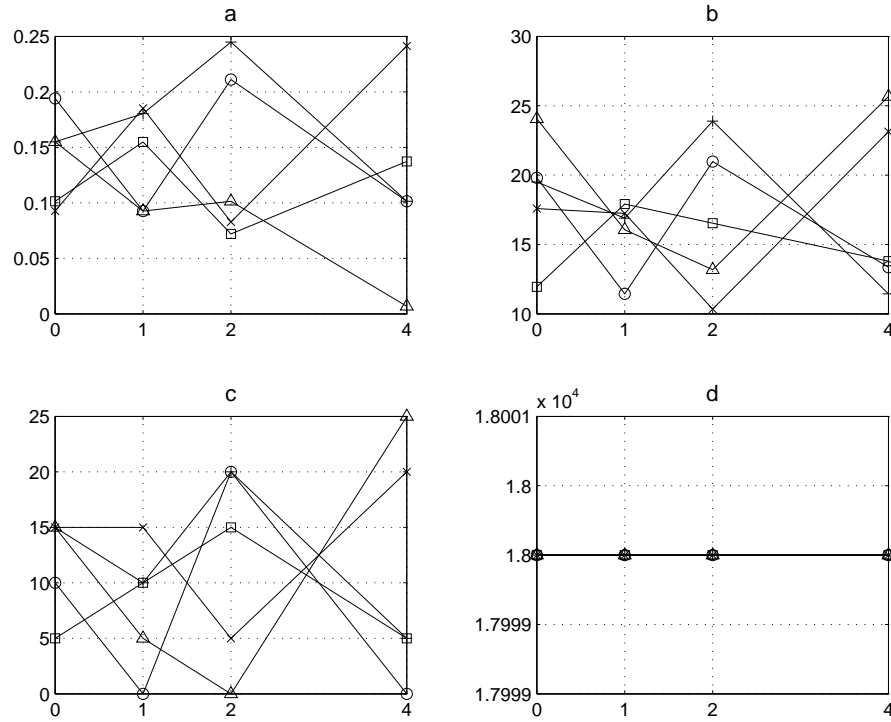


Figura 6.12: Comparação do desempenho do algoritmo de NSGA para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: \times ; 3: +; 4: \square e 5: \triangle .

Tabela 6.60: Índices de desempenho para o algoritmo de SPEA, conjunto de testes 1.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SPEA,C1,0\sigma}$	$\hat{\theta}_{SPEA,C1,1\sigma}$	$\hat{\theta}_{SPEA,C1,2\sigma}$	$\hat{\theta}_{SPEA,C1,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{SPEA,C1,0\sigma,\%}$	$\hat{\theta}_{SPEA,C1,1\sigma,\%}$	$\hat{\theta}_{SPEA,C1,2\sigma,\%}$	$\hat{\theta}_{SPEA,C1,4\sigma,\%}$
I_2	5,0%	0,0%	5,0%	5,0%
I_3	73800,00	121680,00	73680,00	73620,00

Detalhamento da Tabela 6.60:

$$\left\{ \begin{array}{l} \hat{\theta}_{SPEA, C1, 0\sigma} = [-0,04138; 0,08276; -0,04138; 0,00000; 0,00138; 0,00138]^T \\ \hat{\theta}_{SPEA, C1, 0\sigma, \%} = [3,448; 6,897; 3,448; 0,000; 3,448; 3,448]^T \\ \hat{\theta}_{SPEA, C1, 1\sigma} = [-0,04138; -0,08276; -0,04138; 0,00276; 0,00138; -0,00138]^T \\ \hat{\theta}_{SPEA, C1, 1\sigma, \%} = [3,448; 6,897; 3,448; 6,897; 3,448; 3,448]^T \\ \hat{\theta}_{SPEA, C1, 2\sigma} = [0,08276; -0,08276; -0,04138; -0,00138; 0,00138; 0,00138]^T \\ \hat{\theta}_{SPEA, C1, 2\sigma, \%} = [6,897; 6,897; 3,448; 3,448; 3,448; 3,448]^T \\ \hat{\theta}_{SPEA, C1, 4\sigma} = [-0,08276; -0,08276; 0,08276; -0,00138; -0,00138; 0,00414]^T \\ \hat{\theta}_{SPEA, C1, 4\sigma, \%} = [6,897; 6,897; 6,897; 3,448; 3,448; 10,345]^T \end{array} \right.$$

Tabela 6.61: Índices de desempenho para o algoritmo de SPEA, conjunto de testes 2.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SPEA, C2, 0\sigma}$	$\hat{\theta}_{SPEA, C2, 1\sigma}$	$\hat{\theta}_{SPEA, C2, 2\sigma}$	$\hat{\theta}_{SPEA, C2, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{SPEA, C2, 0\sigma, \%}$	$\hat{\theta}_{SPEA, C2, 1\sigma, \%}$	$\hat{\theta}_{SPEA, C2, 2\sigma, \%}$	$\hat{\theta}_{SPEA, C2, 4\sigma, \%}$
I_2	0,0%	5,0%	0,0%	0,0%
I_3	120420,00	74640,00	74460,00	73560,00

Detalhamento da Tabela 6.61:

$$\left\{ \begin{array}{l} \hat{\theta}_{SPEA, C2, 0\sigma} = [0,00000; -0,04138; 0,04138; 0,00000; 0,00138; 0,00138]^T \\ \hat{\theta}_{SPEA, C2, 0\sigma, \%} = [0,000; 3,448; 3,448; 0,000; 3,448; 3,448]^T \\ \hat{\theta}_{SPEA, C2, 1\sigma} = [0,12414; 0,00000; -0,04138; -0,00138; -0,00414; -0,00138]^T \\ \hat{\theta}_{SPEA, C2, 1\sigma, \%} = [10,345; 0,000; 3,448; 3,448; 10,345; 3,448]^T \\ \hat{\theta}_{SPEA, C2, 2\sigma} = [0,00000; 0,08276; 0,00000; -0,00138; -0,00138; -0,00138]^T \\ \hat{\theta}_{SPEA, C2, 2\sigma, \%} = [0,000; 6,897; 0,000; 3,448; 3,448; 3,448]^T \\ \hat{\theta}_{SPEA, C2, 4\sigma} = [0,08276; 0,04138; -0,04138; 0,00138; 0,00000; 0,00138]^T \\ \hat{\theta}_{SPEA, C2, 4\sigma, \%} = [6,897; 3,448; 3,448; 3,448; 0,000; 3,448]^T \end{array} \right.$$

Tabela 6.62: Índices de desempenho para o algoritmo de SPEA, conjunto de testes 3.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SPEA,C3,0\sigma}$	$\hat{\theta}_{SPEA,C3,1\sigma}$	$\hat{\theta}_{SPEA,C3,2\sigma}$	$\hat{\theta}_{SPEA,C3,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{SPEA,C3,0\sigma,\%}$	$\hat{\theta}_{SPEA,C3,1\sigma,\%}$	$\hat{\theta}_{SPEA,C3,2\sigma,\%}$	$\hat{\theta}_{SPEA,C3,4\sigma,\%}$
I_2	0,0%	0,0%	5,0%	15,0%
I_3	75000,00	74940,00	74700,00	75000,00

Detalhamento da Tabela 6.62:

$$\left\{ \begin{array}{l} \hat{\theta}_{SPEA,C3,0\sigma} = [-0,04138; 0,08276; 0,08276; 0,00138; -0,00276; 0,00138]^T \\ \hat{\theta}_{SPEA,C3,0\sigma,\%} = [3,448; 6,897; 6,897; 3,448; 6,897; 3,448]^T \\ \hat{\theta}_{SPEA,C3,1\sigma} = [0,00000; -0,04138; 0,08276; -0,00276; -0,00138; -0,00276]^T \\ \hat{\theta}_{SPEA,C3,1\sigma,\%} = [0,000; 3,448; 6,897; 6,897; 3,448; 6,897]^T \\ \hat{\theta}_{SPEA,C3,2\sigma} = [0,04138; 0,04138; 0,08276; -0,00276; -0,00138; 0,00138]^T \\ \hat{\theta}_{SPEA,C3,2\sigma,\%} = [3,448; 3,448; 6,897; 6,897; 3,448; 3,448]^T \\ \hat{\theta}_{SPEA,C3,4\sigma} = [-0,16552; 0,00000; 0,08276; 0,00138; 0,00138; 0,00000]^T \\ \hat{\theta}_{SPEA,C3,4\sigma,\%} = [13,793; 0,000; 6,897; 3,448; 3,448; 0,000]^T \end{array} \right.$$

Tabela 6.63: Índices de desempenho para o algoritmo de SPEA, conjunto de testes 4.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SPEA,C4,0\sigma}$	$\hat{\theta}_{SPEA,C4,1\sigma}$	$\hat{\theta}_{SPEA,C4,2\sigma}$	$\hat{\theta}_{SPEA,C4,4\sigma}$
$I_{1,\%}$	$\hat{\theta}_{SPEA,C4,0\sigma,\%}$	$\hat{\theta}_{SPEA,C4,1\sigma,\%}$	$\hat{\theta}_{SPEA,C4,2\sigma,\%}$	$\hat{\theta}_{SPEA,C4,4\sigma,\%}$
I_2	5,0%	5,0%	0,0%	10,0%
I_3	75000,00	74700,00	114600,00	127500,00

Detalhamento da Tabela 6.63:

$$\left\{ \begin{array}{l} \hat{\theta}_{SPEA,C4,0\sigma} = [-0,12414; -0,04138; -0,08276; -0,00138; 0,00000; -0,00276]^T \\ \hat{\theta}_{SPEA,C4,0\sigma,\%} = [10,345; 3,448; 6,897; 3,448; 0,000; 6,897]^T \\ \hat{\theta}_{SPEA,C4,1\sigma} = [0,08276; 0,00000; -0,12414; 0,00276; -0,00138; 0,00276]^T \\ \hat{\theta}_{SPEA,C4,1\sigma,\%} = [6,897; 0,000; 10,345; 6,897; 3,448; 6,897]^T \\ \hat{\theta}_{SPEA,C4,2\sigma} = [0,08276; 0,04138; 0,00000; 0,00000; 0,00138; 0,00000]^T \\ \hat{\theta}_{SPEA,C4,2\sigma,\%} = [6,897; 3,448; 0,000; 0,000; 3,448; 0,000]^T \\ \hat{\theta}_{SPEA,C4,4\sigma} = [0,08276; -0,04138; -0,12414; 0,00000; 0,00138; -0,00414]^T \\ \hat{\theta}_{SPEA,C4,4\sigma,\%} = [6,897; 3,448; 10,345; 0,000; 3,448; 10,345]^T \end{array} \right.$$

Tabela 6.64: Índices de desempenho para o algoritmo de SPEA, conjunto de testes 5.

Índice	Ruído			
	0	σ	2σ	4σ
I_1	$\hat{\theta}_{SPEA, C5, 0\sigma}$	$\hat{\theta}_{SPEA, C5, 1\sigma}$	$\hat{\theta}_{SPEA, C5, 2\sigma}$	$\hat{\theta}_{SPEA, C5, 4\sigma}$
$I_{1, \%}$	$\hat{\theta}_{SPEA, C5, 0\sigma, \%}$	$\hat{\theta}_{SPEA, C5, 1\sigma, \%}$	$\hat{\theta}_{SPEA, C5, 2\sigma, \%}$	$\hat{\theta}_{SPEA, C5, 4\sigma, \%}$
I_2	10,0%	5,0%	5,0%	10,0%
I_3	75000,00	74940,00	120480,00	74940,00

Detalhamento da Tabela 6.64:

$$\left\{ \begin{array}{l}
 \hat{\theta}_{SPEA, C5, 0\sigma} = [0,12414; -0,04138; 0,00000; 0,00276; 0,00000; 0,00000; \\
 \quad 0,99931; 0,99793; 1,00138; 1,00000; 1,00069; 1,00069]^T \\
 \hat{\theta}_{SPEA, C5, 0\sigma, \%} = [10,345; 3,448; 0,000; 6,897; 0,000; 0,000; \\
 \quad 3,448; 10,345; 6,897; 0,000; 3,448; 3,448]^T \\
 \hat{\theta}_{SPEA, C5, 1\sigma} = [0,04138; 0,00000; -0,08276; 0,00138; 0,00414; 0,00138; \\
 \quad 1,00000; 1,00069; 1,00138; 1,00000; 1,00069; 0,99862]^T \\
 \hat{\theta}_{SPEA, C5, 1\sigma, \%} = [3,448; 0,000; 6,897; 3,448; 10,345; 3,448; \\
 \quad 0,000; 3,448; 6,897; 0,000; 3,448; 6,897]^T \\
 \hat{\theta}_{SPEA, C5, 2\sigma} = [-0,08276; 0,00000; -0,04138; 0,00000; 0,00000; 0,00000; \\
 \quad 0,99862; 0,99862; 1,00000; 1,00069; 0,99862; 1,00000]^T \\
 \hat{\theta}_{SPEA, C5, 2\sigma, \%} = [6,897; 0,000; 3,448; 0,000; 0,000; 0,000; \\
 \quad 6,897; 6,897; 0,000; 3,448; 6,897; 0,000]^T \\
 \hat{\theta}_{SPEA, C5, 4\sigma} = [0,12414; 0,12414; 0,00000; -0,00276; 0,00000; -0,00138; \\
 \quad 1,00207; 1,00069; 1,00207; 1,00138; 1,00207; 1,00069]^T \\
 \hat{\theta}_{SPEA, C5, 4\sigma, \%} = [10,345; 10,345; 0,000; 6,897; 0,000; 3,448; \\
 \quad 10,345; 3,448; 10,345; 6,897; 10,345; 3,448]^T
 \end{array} \right.$$

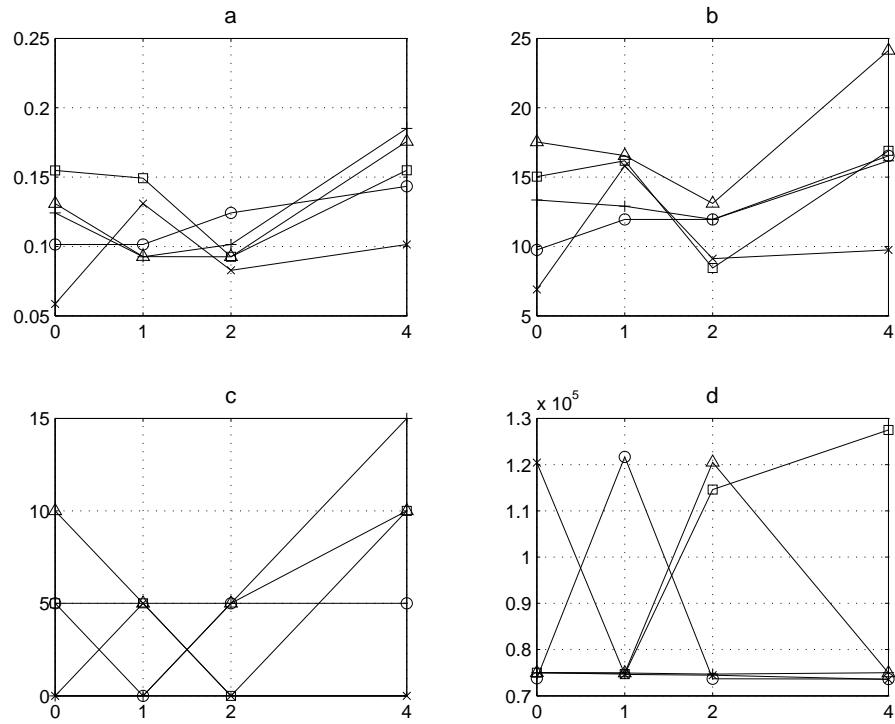


Figura 6.13: Comparação do desempenho do algoritmo de SPEA para diferentes conjuntos de teste segundo: a) a distância Euclidiana entre vetor real e vetor estimado a) no espaço de soluções e b) no espaço normalizado (0 a 100%), c) I_2 e d) I_3 . Conjuntos de teste 1: o; 2: ×; 3: +; 4: □ e 5: △.

A Figura 6.14 ilustra a possibilidade de classificação dos algoritmos genéticos segundo a qualidade de reconstrução associada à sua utilização. Essa análise é feita com a apresentação dos valores correspondentes ao índice $I_{4,\%}$.

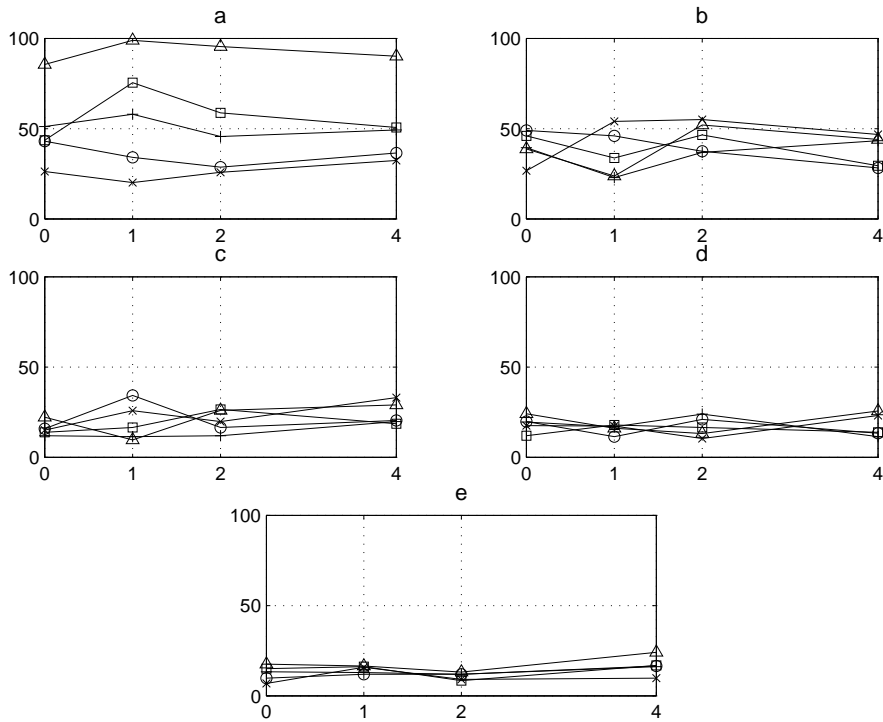


Figura 6.14: Comparação do índice $I_{4,\%}$ para os algoritmos genéticos em escala: a) VEGA; b) WBGA; c) MOGA; d) NSGA II; e) SPEA.

Com o intuito de ilustrar o significado dos valores obtidos para os índices de desempenho, as Figuras 6.15 e 6.16 apresentam exemplos de resultados do procedimento de adequação de curvas para os sinais provenientes dos acelerômetros da unidade de medição inercial (a_x , a_y e a_z). Em todos os casos, o eixo horizontal consiste no tempo em segundos e as acelerações são representadas em m/s^2 . Os gráficos são organizados de tal forma que:

- a) – implica a_x ; –· implica $a_x + 0,25 \Delta_{acc}^{max}$; ··· implica $a_x + 0,50 \Delta_{acc}^{max}$;
- b) – implica a_x ; –· implica $a_x + 0,75 \Delta_{acc}^{max}$; ··· implica $a_x + \Delta_{acc}^{max}$;
- c) – implica a_y ; –· implica $a_y + 0,25 \Delta_{acc}^{max}$; ··· implica $a_y + 0,50 \Delta_{acc}^{max}$;
- d) – implica a_y ; –· implica $a_y + 0,75 \Delta_{acc}^{max}$; ··· implica $a_y + \Delta_{acc}^{max}$;
- e) – implica a_z ; –· implica $a_z + 0,25 \Delta_{acc}^{max}$; ··· implica $a_z + 0,50 \Delta_{acc}^{max}$;
- f) – implica a_z ; –· implica $a_z + 0,75 \Delta_{acc}^{max}$; ··· implica $a_z + \Delta_{acc}^{max}$.

No caso da Figura 6.15, deve-se observar que sua principal função é comparar a distância entre as curvas com os valores absolutos das medições (ainda que a proximidade entre as curvas impeça a sua visualização).

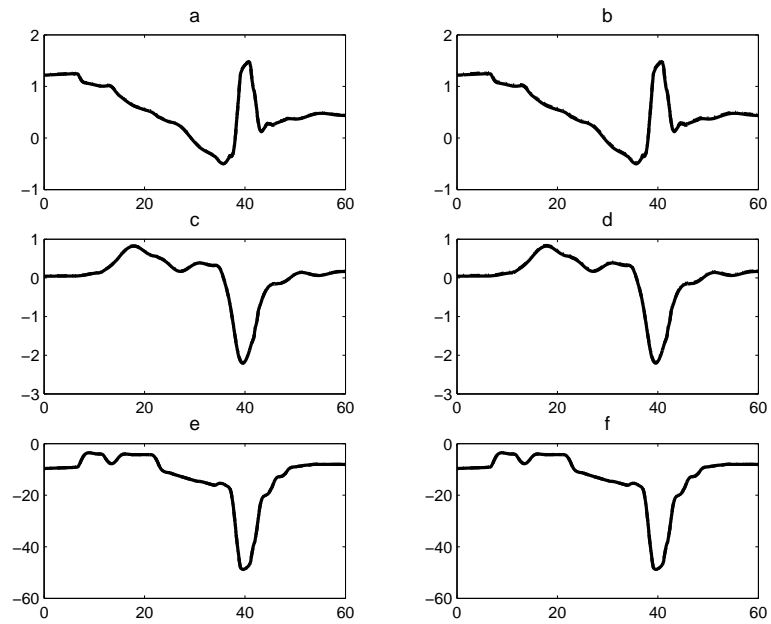


Figura 6.15: Resultado da adequação das séries temporais de a_x , a_y e a_z para valores selecionados de termos de polarização.

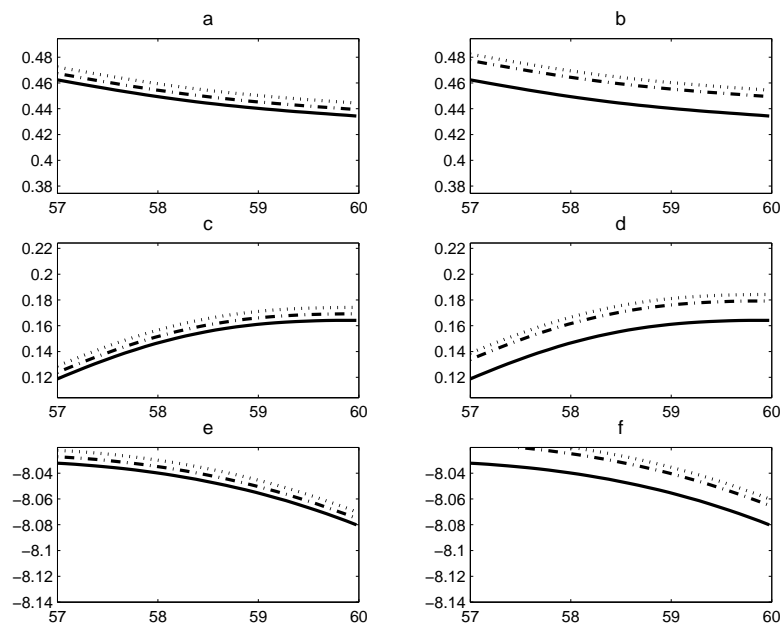


Figura 6.16: Resultado da adequação das séries temporais de a_x , a_y e a_z para valores selecionados de termos de polarização em detalhe.

Além disso, apresentam-se exemplos de resultados do procedimento de adequação de curvas para os sinais provenientes dos girômetros (p , q e r) na Figura 6.17. Em todos os casos, o eixo horizontal consiste no tempo em segundos e as velocidades de rotação são representadas em rad/s. Os gráficos são organizados de tal forma que:

- a) – implica p ; \cdots implica $p + 0,25 \Delta_{gir}^{max}$; \cdots implica $p + 0,50 \Delta_{gir}^{max}$;
b) – implica p ; \cdots implica $p + 0,75 \Delta_{gir}^{max}$; \cdots implica $p + \Delta_{gir}^{max}$;
c) – implica q ; \cdots implica $q + 0,25 \Delta_{gir}^{max}$; \cdots implica $q + 0,50 \Delta_{gir}^{max}$;
d) – implica q ; \cdots implica $q + 0,75 \Delta_{gir}^{max}$; \cdots implica $q + \Delta_{gir}^{max}$;
e) – implica r ; \cdots implica $r + 0,25 \Delta_{gir}^{max}$; \cdots implica $r + 0,50 \Delta_{gir}^{max}$;
f) – implica r ; \cdots implica $r + 0,75 \Delta_{gir}^{max}$; \cdots implica $r + \Delta_{gir}^{max}$.

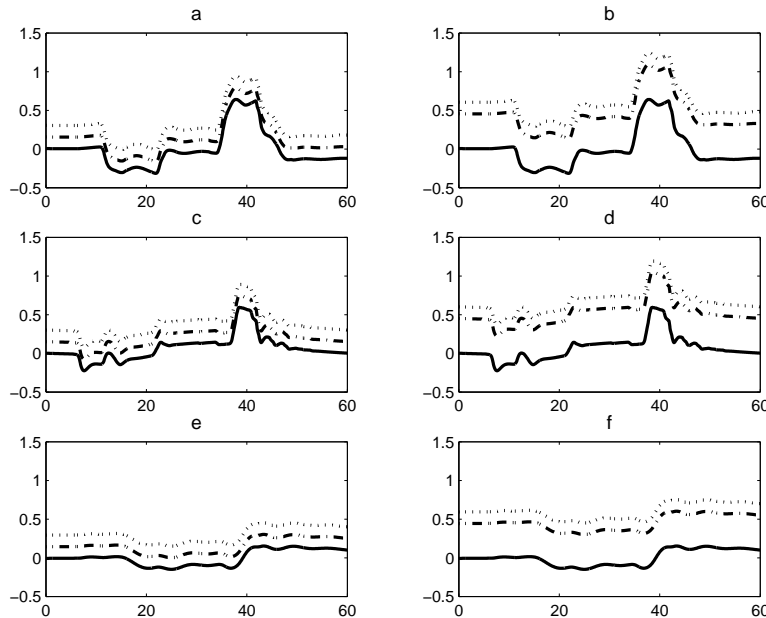


Figura 6.17: Resultado da adequação das séries temporais de p , q e r para valores selecionados de termos de polarização.

Ainda com o intuito de exemplificar os valores obtidos para os índices de desempenho, deve-se observar que um valor de $I_{4,\%} = 100\%$, por exemplo, implica que um elemento i do vetor de parâmetros tem valor médio percentual tal que

$$I_1(i) = \frac{100}{\sqrt{n}} \quad (6.1)$$

em que n é a dimensão do vetor de parâmetros. Com isso, para o caso de um vetor de parâmetros de dimensão seis, tem-se $I_1(i) \approx 40,8$, em média.

6.2 Tempo de execução

Quanto à análise do índice I_3 , a primeira distinção se deve à grande diferença existente entre os algoritmos genéticos e os não genéticos. Enquanto os não genéticos tendem a executar o modelo cinemático algumas centenas de vezes, os algoritmos genéticos o fazem várias dezenas de milhares de vezes. Esse aspecto certamente desmotiva a utilização de algoritmos genéticos para resolução de problemas de reconstrução de trajetória de aeronaves.

Com o intuito de ressaltar esse aspecto, considere-se por exemplo, um cenário de reconstrução com número de execuções do modelo cinemático igual a 10.000. A simulação do modelo cinemático em um computador com um processador de 3GHz e com 1GB de memória RAM², leva cerca de 800ms. Isso implica um tempo de processamento de cerca de 2h e 15min.

Deve-se também observar que a execução dos operadores genéticos e das estratégias utilizadas por esse tipo de algoritmo (como a contagem das distâncias de cada indivíduo com todos os demais, no caso do SPEA), tipicamente dobra o tempo médio de uma execução do modelo cinemático. Além disso, boa parte dos algoritmos genéticos estudados neste trabalho apresenta valores significativamente maiores que o apresentado: o SPEA, por exemplo, implica 120.000 execuções em boa parte dos cenários avaliados. Nesse caso extremo, o tempo de processamento chega às 53h e 20min, em média.

Quanto aos algoritmos não genéticos, o tempo de processamento tende a ser proporcional ao tamanho do vetor de parâmetros estimado. Com isso, o conjunto de testes 5 (em que tanto os parâmetros de polarização quanto os de ganho são estimados) tende a ser processado em um tempo significativamente maior que os demais.

Ainda no que diz respeito a esse grupo de algoritmos, três merecem destaque: os métodos de GCP, SA e NM. No primeiro caso, observa-se um tempo de processamento significativamente maior que os demais. Essa diferença de comportamento se deve basicamente à concepção do método no que diz respeito às execuções do modelo para cálculo da matriz Jacobiana.

O procedimento de correção da matriz Jacobiana (eventualmente mal-condicionada) custa tipicamente pouco mais de três vezes o tempo de processamento da Jacobiana em si. Esse é motivo do aparecimento de um fator 4

²Configuração do computador utilizado neste trabalho.

na comparação entre os tempos de execução dos algoritmos de GR e GC com o de GCP.

No caso do algoritmo de SA, observa-se uma particularidade: o tempo de processamento permanece constante independentemente do cenário de reconstrução. A justificativa para esse comportamento se deve à metodologia utilizada pelo método para transição entre estados de maiores ou menores energias associadas.

De forma desvinculada à aceitação ou não do estado seguinte, o algoritmo simplesmente executa o modelo cinemático com o novo vetor de parâmetros. Isso implica que a execução do modelo cinemático pelo algoritmo uma vez a cada tentativa de transição de estados, o que, de fato, dá essa característica ao SA.

Finalmente, os valores do índice I_3 para o algoritmo de NM devem ser analisados. Para isso, deve-se primeiramente observar que a etapa de encolhimento é a única para a qual mais de um ponto da hipersuperfície da função objetivo é avaliado (especificamente, n em um total de $n + 1$). Portanto, a execução dessa etapa é a responsável pela diferença no número de execuções do modelo cinemático.

É intuitivo pensar que quanto melhor o cenário de estimação, maior será o número de encolhimentos sofridos pela figura geométrica de n dimensões ($n + 1$ vértices) representada pela matriz simplex. Essa afirmativa se deve ao simples fato de a matriz simplex tender a entrar em colapso (volume nulo) no ponto ótimo da função objetivo. Em outras palavras, quanto melhor o cenário de estimação (representado pelos sinais de vôo medidos) maior será a eficiência do algoritmo em achar o ponto ótimo da hipersuperfície da função objetivo e, conseqüentemente, maior será o número de encolhimentos executados.

Com isso, é interessante observar que, embora não apresente grandes variações (como ocorre para boa parte dos demais algoritmos), o tempo de execução do método de NM reflete o nível de excitação das dinâmicas (longitudinal e látero-direcional) da aeronave. Essa característica relaciona, em princípio, a dificuldade (ou facilidade) de convergência do algoritmo de NM com o grau de identificabilidade do vetor de parâmetros estimados.

Finalmente, deve-se analisar o motivo pelo qual os termos de ganho são inseridos apenas no conjunto de testes 5. Com o transcorrer dos testes exe-

cutados neste trabalho, observou-se uma forte sensibilidade das funções objetivo a eventuais modificações nos valores de ganho. De fato, a inserção desses parâmetros no vetor a ser estimado causa a ocorrência de uma grande quantidade de estimações sabidamente erradas. Essas estimações são identificadas pela existência de valores estimados fora das faixas dos respectivos instrumentos.

Alguns testes de sensibilidade foram feitos e também se observou que a dificuldade de convergência dos algoritmos não depende exclusivamente do aumento da dimensão do espaço de busca com a inclusão de termos de ganho. De fato, ainda que se trate de um espaço de dimensão seis (ou mesmo três), os algoritmos encontram basicamente o mesmo tipo de dificuldade durante o processo de otimização. Esse é o motivo pelo qual o conjunto de testes 5 representa tipicamente o pior cenário de estimação analisado neste trabalho.

CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusões

As conclusões provenientes da análise dos resultados obtidos neste trabalho podem ser divididas em três grupos principais:

1. relativo aos algoritmos genéticos;
2. relativo aos algoritmos baseados no cálculo de derivadas;
3. relativo à inclusão dos termos de ganho no vetor de parâmetros.

No que diz respeito aos algoritmos genéticos, deve-se observar que, apesar dos bons resultados relacionados à qualidade da reconstrução se comparados aos algoritmos não genéticos, o custo associado ao tempo de execução desmotivava a sua utilização. De fato, a utilização dos algoritmos genéticos para a resolução do problema de reconstrução de trajetória de aeronaves deve ser precedida de uma verificação da necessidade de um melhor desempenho na diminuição dos erros de estimação, ainda que esse procedimento implique um aumento significativo do tempo de execução.

Quanto aos algoritmos baseados no cálculo de derivadas, deve-se observar que, por esse cálculo se referir a valores da função objetivo, não são descritos por uma fórmula analítica. Devido a isso, utiliza-se um procedimento de aproximação por diferenças finitas, o que diminui sensivelmente a confiabilidade nos resultados obtidos. Portanto, a utilização de algoritmos com essa característica deve ser feita sempre com esse aspecto em vista.

Finalmente, no que diz respeito à inclusão dos termos de ganho no vetor de parâmetros, observa-se uma diminuição de desempenho dos algoritmos devido a dois motivos principais:

- aumento da dimensão do espaço de parâmetros: por se tratarem de algoritmos de otimização, esse aumento está tipicamente associado a uma maior dificuldade de convergência dos algoritmos para o ótimo global do problema a ser resolvido;
- grande sensibilidade das funções objetivo à variação dos termos de ganho: com o transcorrer das realizações dos algoritmos para os cenários de reconstrução utilizados, observa-se que pequenas modificações nos valores dos parâmetros relativos aos termos de ganho provocam grandes alterações nos valores das funções objetivo. Esse aspecto causa a divergência de boa parte dos algoritmos quando utilizados nesse contexto.

7.2 Trabalhos Futuros

A finalização deste projeto cria a possibilidade de desenvolvimento de novos trabalhos relacionados em diversas direções. Essas novas abordagens podem contemplar, por exemplo:

- maior particularização na implementação dos algoritmos de Nelder-Mead e Levenberg-Marquardt: com a análise dos resultados obtidos e tendo em vista o tipo de funcionalidade utilizada por cada algoritmo, observa-se que alguma melhoria ainda pode ser feita com o intuito de se aprimorarem esses algoritmos para a resolução do problema de reconstrução de trajetória de aeronaves.
No primeiro caso, o bom desempenho e a relativa simplicidade de implementação criam um cenário bastante propício para a implementação de futuras melhorias. Quanto ao algoritmo de Levenberg-Marquardt, apesar de utilizar um procedimento de aproximação de derivadas por diferenças finitas, consiste em um dos métodos mais difundidos para a resolução dos mais diversos tipos de problemas de otimização e, recentemente, também no contexto de reconstrução de trajetória de aeronaves;
- atribuição de diferentes pesos aos diversos objetivos: com o intuito de se privilegiar a adequação de determinados pares de curvas em detrimento à adequação de outros pares de curvas. Com essa abordagem, pode-se definir valores de erro relativamente maiores quando associados a problemas de inadequação da curva de altitude em relação a todas os demais sinais de saída do modelo cinemático, por exemplo;

-
- utilização da aplicação em novos contextos (principal motivação para a sua criação):
 - algoritmos de estimação de estados: amplamente utilizados na resolução do problema de reconstrução de trajetória de aeronaves;
 - algoritmos de estimação de parâmetros.

ALGUMAS PROPRIEDADES MATEMÁTICAS

A.1 Rotação de coordenadas

Problemas relacionados à rotação de coordenadas surgem com bastante frequência quando se trata da navegação de veículos. No que diz respeito às aeronaves, torna-se particularmente importante a capacidade de se representarem vetores em novos sistemas de coordenadas cartesianos. Devido a isso, algum procedimento deve ser adotado com o intuito de possibilitar a representação de um vetor originalmente pertencente a um sistema de coordenadas 1 em um novo sistema de coordenadas 2. A relação entre $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^{3 \times 1}$ pode ser expressa por

$$\mathbf{v}_2 = R\mathbf{v}_1, \tag{A.1}$$

em que $R \in \mathbb{R}^{3 \times 3}$ consiste em uma matriz responsável pela orientação do sistema de coordenadas 2 em relação ao 1.

Uma característica intuitiva da rotação de coordenadas consiste em sua propriedade de garantir que o processo de rotação não altere o comprimento¹ dos vetores rotacionados. Com isso, deve-se garantir que

$$\|\mathbf{v}_2\|_2 - \|\mathbf{v}_1\|_2 = \sqrt{\mathbf{v}_2^T \mathbf{v}_2} - \sqrt{(R\mathbf{v}_1)^T R\mathbf{v}_1} = \sqrt{\mathbf{v}_2^T \mathbf{v}_2} - \sqrt{\mathbf{v}_1^T R^T R \mathbf{v}_1} = 0,$$

o que implica

$$R^T R = I = R R^T. \tag{A.2}$$

¹Por simplicidade, o comprimento de um vetor será sempre considerado igual à sua norma euclidiana.

A equação (A.2) consiste na definição de matrizes ortogonais (Chen, 1999), o que torna o cálculo de sua inversa particularmente simples: $R^{-1} = R^T$. Além disso, essa equação também garante que as colunas (e linhas) da matriz R compõem um conjunto ortonormal de vetores.

$$R = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3], \quad \mathbf{r}_i^T \mathbf{r}_j = \begin{cases} 0, & i \neq j \\ 1, & i = j. \end{cases} \quad (\text{A.3})$$

Uma importante constatação a partir de (A.3) é que os vetores unitários ortogonais paralelos aos eixos x , y e z do espaço tridimensional A (\mathbf{i}_A , \mathbf{j}_A e \mathbf{k}_A respectivamente) correspondem à primeira, segunda e terceira colunas da matriz de rotação (nessa ordem) no novo sistema de coordenadas B .

$$\begin{aligned} \mathbf{r}_1 &= R \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = R\mathbf{i}_A; & \mathbf{r}_2 &= R \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = R\mathbf{j}_A; & \mathbf{r}_3 &= R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = R\mathbf{k}_A \\ & \Rightarrow \mathbf{r}_1 = \mathbf{i}_B; & \mathbf{r}_2 &= \mathbf{j}_B; & \mathbf{r}_3 &= \mathbf{k}_B. \end{aligned}$$

Uma matriz de rotação preserva as propriedades do produto vetorial dos eixos do sistema de coordenadas. Devido a isso, os vetores que compõem as colunas da matriz de rotação também satisfazem as seguintes propriedades:

$$\begin{aligned} \mathbf{r}_1 \times \mathbf{r}_2 &= \mathbf{r}_3 \\ \mathbf{r}_2 \times \mathbf{r}_3 &= \mathbf{r}_1 \\ \mathbf{r}_3 \times \mathbf{r}_1 &= \mathbf{r}_2. \end{aligned}$$

Utilizando-se essas propriedades, pode-se mostrar (Chen, 1999) que uma matriz de rotação respeita a propriedade distributiva conforme ilustrado em (A.4).

$$R(\mathbf{u} \times \mathbf{v}) = (R\mathbf{u}) \times (R\mathbf{v}). \quad (\text{A.4})$$

De forma mais direta: O resultado da rotação de um produto vetorial é igual ao produto vetorial dos mesmos vetores previamente rotacionados. A importância dessa constatação está no fato de que se torna indiferente a ordem relativa da aplicação das operações de rotação e produto vetorial.

A orientação de um sistema de coordenadas em relação a outro pode ser descrita por uma seqüência de rotações sobre seus eixos. Por exemplo, considerando-se que o sistema de coordenadas 2 consiste no sistema de coordenadas 1 rotacionado de $\theta > 0$ sobre o eixo dos z (Figura A.1), deseja-se obter o valor de $\mathbf{v}_2 = [x_2 \ y_2 \ z_2]^T$ dado que se conhece $\mathbf{v}_1 = [x_1 \ y_1 \ z_1]^T$.

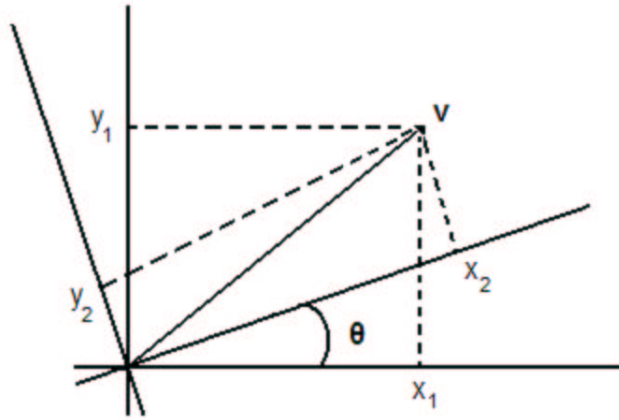


Figura A.1: Rotação de coordenadas em um plano

A matriz de rotação necessária deve, intuitivamente, aproximar-se de I à medida que θ tende a zero. Além disso, deve conter termos do tipo $\sin \theta$ e $\cos \theta$ e não deve modificar o componente em z do vetor a ser rotacionado. A análise geométrica do problema leva a

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (\text{A.5})$$

Interessante observar em (A.5) que a mudança no sentido de rotação ($\theta < 0$) implica a utilização de R^T como matriz de rotação. Além disso, a constatação de que a ordem das operações de rotação influencia a orientação do sistema de coordenadas resultante induz à necessidade de especificação de uma seqüência para essas operações.

A prática na indústria aeronáutica define as rotações sobre os eixos z , y e x ordenados dessa maneira. Com isso, a representação de um vetor em um novo sistema de coordenadas implica

$$\mathbf{v}_2 = R_x R_y R_z \mathbf{v}_1 \quad (\text{A.6})$$

e, devido a isso, a operação inversa torna-se

$$\mathbf{v}_1 = (R_x R_y R_z)^{-1} \mathbf{v}_2 = R_z^T R_y^T R_x^T \mathbf{v}_2. \quad (\text{A.7})$$

A.2 Diferenciação de vetores

Para a aplicação das leis de Newton relacionadas à cinemática de corpos rígidos, torna-se necessário definir a taxa de variação de uma quantidade vetorial. Mais especificamente, deve-se calcular a derivada no tempo de um vetor de velocidade absoluta e representar o resultado em outro sistema de coordenadas (2.3). O novo sistema de coordenadas pode possuir velocidade angular não-nula sobre seus próprios eixos ou com relação ao referencial inercial (Figura A.2). Devido a isso, a derivada em questão corresponderá à projeção instantânea da derivada absoluta no sistema de coordenadas em rotação.

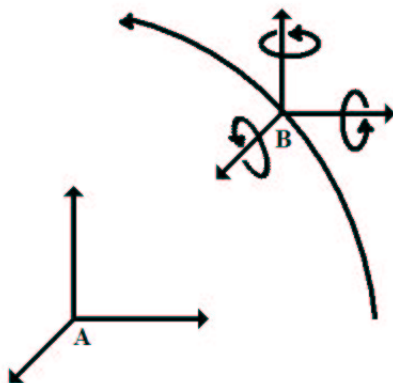


Figura A.2: Ilustração de movimentos de rotação entre dois sistemas de coordenadas

A notação adotada para definir a operação de diferenciação de vetores deve ser tal que tanto o sistema de coordenadas no qual a derivada é calculada quanto aquele no qual o resultado deve ser representado devem ser explicitados. Para isso, o subíndice no operador de diferenciação d/dt será utilizado para indicar o sistema de coordenadas no qual a derivada é calculada e o subíndice no vetor a ser derivado indicará aquele em que o resultado será representado. Caso se trate do mesmo sistema de coordenadas, os subíndices serão omitidos.

Se \mathbf{v} consiste em um vetor do espaço tridimensional, então

$$\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}.$$

Com isso

$$\frac{d}{dt}(\mathbf{v}) \triangleq \dot{\mathbf{v}} \triangleq \dot{v}_x \mathbf{i} + \dot{v}_y \mathbf{j} + \dot{v}_z \mathbf{k}. \quad (\text{A.8})$$

Caso os sistemas de coordenadas sejam diferentes e considere-se R uma matriz de rotação que representa vetores do espaço A no espaço B então

$$\mathbf{v}_B = R\mathbf{v}_A, \quad (\text{A.9})$$

e uma expressão para a derivada de \mathbf{v}_B em relação ao sistema de coordenadas A pode ser representada por

$$\frac{d}{dt_A}(\mathbf{v}_B) = R\dot{\mathbf{v}}_A. \quad (\text{A.10})$$

Além disso, segundo (A.9) e a regra da cadeia para diferenciação de produtos, tem-se que

$$\frac{d}{dt_B}(\mathbf{v}_B) = \dot{\mathbf{v}}_B = \dot{R}\mathbf{v}_A + R\dot{\mathbf{v}}_A, \quad (\text{A.11})$$

em que $\dot{R} \neq \mathbf{0}$ se $R = R(t)$ e, portanto, A possui velocidade de rotação não-nula em relação ao sistema de coordenadas B .

Com isso, a substituição de (A.11) em (A.10) implica

$$\frac{d}{dt_A}(\mathbf{v}_B) = \dot{\mathbf{v}}_B - \dot{R}\mathbf{v}_A. \quad (\text{A.12})$$

A equação (A.12) indica um importante resultado. A derivada de um vetor em relação a um sistema de coordenadas diferente daquele no qual é representado possui duas parcelas. A primeira parcela consiste na derivada com respeito ao sistema de coordenadas no qual o vetor é representado e a segunda em um termo diferente de zero apenas se o sistema de coordenadas no qual o resultado deve ser representado apresentar velocidade angular não-nula em relação ao sistema de coordenadas original. Com isso, a diferenciação de um vetor em relação a um sistema de coordenadas diferente daquele no qual é representado é diferente do caso em que se trata do mesmo sistema de coordenadas apenas se os sistemas de coordenadas apresentarem velocidade angular relativa diferente de zero.

Uma versão alternativa da equação (A.12), segundo o *teorema de Coriolis*, consiste em

$$\frac{d}{dt_A}(\mathbf{v}_B) = \dot{\mathbf{v}}_B + \boldsymbol{\omega}_B \times \mathbf{v}_B, \quad (\text{A.13})$$

em que $\boldsymbol{\omega}_B$ representa a velocidade angular relativa do sistema de coordenadas B em relação ao A . Com isso, todos os vetores são representados no mesmo sistema de coordenadas.

A comparação entre as equações (A.12) e (A.13) indica a possibilidade de existência de uma relação cinemática entre a taxa de variação da matriz de rotação \dot{R} e o vetor de velocidade angular $\boldsymbol{\omega}_B$. De fato, substituindo-se o produto vetorial em (A.13) pela matriz de produto vetorial Ω_B e com a utilização da equação (A.9) na substituição de \mathbf{v}_B , após a comparação do resultado com (A.12) obtém-se

$$\dot{R} = -\Omega_B R, \quad (\text{A.14})$$

o que implica

$$\Omega_B = R\dot{R}^T. \quad (\text{A.15})$$

A prova da existência de uma matriz Ω capaz de substituir um produto vetorial $\mathbf{v} \times \mathbf{w}$ é direta. Considerando-se

$$\mathbf{v} \times \mathbf{w} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{vmatrix} = \begin{vmatrix} v_2 & v_3 \\ w_2 & w_3 \end{vmatrix} \mathbf{i} + \begin{vmatrix} v_1 & v_3 \\ w_1 & w_3 \end{vmatrix} \mathbf{j} + \begin{vmatrix} v_1 & v_2 \\ w_1 & w_2 \end{vmatrix} \mathbf{k},$$

em que $\mathbf{v} = [v_1 \ v_2 \ v_3]^T$ e $\mathbf{w} = [w_1 \ w_2 \ w_3]^T$, tem-se

$$\mathbf{v} \times \mathbf{w} = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \triangleq \Omega \mathbf{w}. \quad (\text{A.16})$$

A equação (A.14) consiste em uma relação cinemática fundamental na análise da rotação relativa entre dois sistemas de coordenadas. Essa equação relaciona as duas representações de variações angulares responsáveis pela descrição do movimento do sistema de coordenadas B em relação ao A . Associa-se a taxa de rotação relativa do sistema de coordenadas B (em R) com o vetor de velocidade angular medido no mesmo sistema de coordenadas (em Ω_B).

A importância da equação *strapdown* (A.14) se faz na possibilidade de sua integração numérica em sistemas de navegação inercial a bordo e assim de calcular a atitude do veículo (Santana et al., 2004). Girômetros alinhados aos eixos do sistema de coordenadas do veículo medem as velocidades angulares em Ω_B e a matriz de rotação é calculada como uma função do tempo.

A informação a respeito da atitude contida na matriz de rotação é então utilizada para ajustar as medições de aceleração provenientes dos acelerômetros fixados ao corpo do veículo.

Finalmente, deve-se definir a diferenciação de um produto vetorial. Seu desenvolvimento provém da utilização da regra da cadeia para diferenciação de operações de multiplicação.

$$\frac{d}{dt}(\mathbf{u} \times \mathbf{v}) = \dot{\mathbf{u}} \times \mathbf{v} + \mathbf{u} \times \dot{\mathbf{v}}. \quad (\text{A.17})$$

A.3 Gradiente, Jacobiana e Hessiana

O vetor de dimensão n composto pelas derivadas parciais de uma determinada função escalar $c(\boldsymbol{\theta})$ em relação a $\boldsymbol{\theta} \in \mathbb{R}^{n \times 1}$ é denominado *vetor gradiente* de $c(\boldsymbol{\theta})$:

$$\nabla c(\boldsymbol{\theta}) \triangleq \mathbf{g}_c(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \frac{\partial c}{\partial \theta_1} \\ \frac{\partial c}{\partial \theta_2} \\ \vdots \\ \frac{\partial c}{\partial \theta_n} \end{bmatrix}. \quad (\text{A.18})$$

Quando se trata de uma função vetorial $\mathbf{c}(\boldsymbol{\theta}) = [c_1(\boldsymbol{\theta}) \ c_2(\boldsymbol{\theta}) \ \cdots \ c_m(\boldsymbol{\theta})]^T$, estende-se o conceito do vetor gradiente sob a forma de uma matriz, denominada *matriz Jacobiana* de $\mathbf{c}(\boldsymbol{\theta})$:

$$J_c(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \mathbf{g}_{c_1}^T \\ \mathbf{g}_{c_2}^T \\ \vdots \\ \mathbf{g}_{c_m}^T \end{bmatrix} = \begin{bmatrix} \frac{\partial c_1}{\partial \theta_1} & \frac{\partial c_1}{\partial \theta_2} & \cdots & \frac{\partial c_1}{\partial \theta_n} \\ \frac{\partial c_2}{\partial \theta_1} & \frac{\partial c_2}{\partial \theta_2} & \cdots & \frac{\partial c_2}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial c_m}{\partial \theta_1} & \frac{\partial c_m}{\partial \theta_2} & \cdots & \frac{\partial c_m}{\partial \theta_n} \end{bmatrix}. \quad (\text{A.19})$$

Finalmente, a representação da segunda derivada de uma função escalar $c_i(\boldsymbol{\theta})$ assume a forma de uma matriz quadrada $n \times n$. A Hessiana de $c_i(\boldsymbol{\theta})$ é definida por:

$$\nabla^2 c_i(\boldsymbol{\theta}) \triangleq \nabla(\nabla c_i(\boldsymbol{\theta}))^T \triangleq G_{c_i}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial^2 c_i}{\partial \theta_1^2} & \frac{\partial^2 c_i}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 c_i}{\partial \theta_1 \partial \theta_n} \\ \frac{\partial^2 c_i}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 c_i}{\partial \theta_2^2} & \cdots & \frac{\partial^2 c_i}{\partial \theta_2 \partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 c_i}{\partial \theta_n \partial \theta_1} & \frac{\partial^2 c_i}{\partial \theta_n \partial \theta_2} & \cdots & \frac{\partial^2 c_i}{\partial \theta_n^2} \end{bmatrix}. \quad (\text{A.20})$$

A.3.1 Particularidades em Mínimos Quadrados

Em problemas de Mínimos Quadrados, a função objetivo $c(\boldsymbol{\theta})$ assume a forma

$$c(\boldsymbol{\theta}) \triangleq c_{\text{MQ}}(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^N f(\boldsymbol{\theta}, t_i)^2 = \frac{1}{2} \|f(\boldsymbol{\theta}, t)\|^2, \quad (\text{A.21})$$

em que $f(\boldsymbol{\theta}, t_i) : \mathbb{R}^n \rightarrow \mathbb{R}$, $f(\boldsymbol{\theta}, t) : \mathbb{R}^n \rightarrow \mathbb{R}^N$, n é a dimensão de $\boldsymbol{\theta}$ e N é o tamanho da série (temporal) correspondente à simulação de $f(\boldsymbol{\theta}, t)$.

Com isso, a representação de segunda ordem da expansão em séries de Taylor de $c_{\text{MQ}}(\boldsymbol{\theta})$ em torno de um ponto $\bar{\boldsymbol{\theta}}$ no espaço de parâmetros implica

$$c_{\text{MQ}}(\boldsymbol{\theta}) = c_{\text{MQ}}(\bar{\boldsymbol{\theta}}) + \mathbf{g}_{c_{\text{MQ}}}(\bar{\boldsymbol{\theta}}) \cdot (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) + \frac{1}{2} \cdot G_{c_{\text{MQ}}}(\bar{\boldsymbol{\theta}}) \cdot (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^2. \quad (\text{A.22})$$

Mas $f(\boldsymbol{\theta}, t)$ é tipicamente definido como o erro (de estimação) cometido ao se tentar aproximar o comportamento de um modelo, $\phi(\boldsymbol{\theta}, t)$, ao de um sistema real cuja saída seja representada por $y(t)$, por exemplo:

$$f(\boldsymbol{\theta}, t) = \phi(\boldsymbol{\theta}, t) - y(t). \quad (\text{A.23})$$

Com a definição de $c_{\text{MQ}}(\boldsymbol{\theta})$ no terceiro termo da equação (A.21), obtém-se

$$\begin{aligned} \mathbf{g}_{c_{\text{MQ}}}(\boldsymbol{\theta}) &= \frac{\partial c_{\text{MQ}}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \\ &= \sum_{i=1}^N \frac{\partial f(\boldsymbol{\theta}, t_i)}{\partial \boldsymbol{\theta}} f(\boldsymbol{\theta}, t_i). \end{aligned}$$

Mas, de acordo com (A.23),

$$\frac{\partial f(\boldsymbol{\theta}, t)}{\partial \boldsymbol{\theta}} = \frac{\partial \phi(\boldsymbol{\theta}, t)}{\partial \boldsymbol{\theta}}, \quad (\text{A.24})$$

o que, com (A.23), implica

$$\mathbf{g}_{c_{\text{MQ}}}(\boldsymbol{\theta}) = \sum_{i=1}^N \frac{\partial \phi(\boldsymbol{\theta}, t_i)}{\partial \boldsymbol{\theta}} (\phi(\boldsymbol{\theta}, t_i) - y(t_i)). \quad (\text{A.25})$$

De maneira análoga ao que foi descrito para o cálculo do vetor gradiente, a matriz Hessiana $G_{c_{\text{MQ}}}(\boldsymbol{\theta})$ se torna

$$G_{c_{\text{MQ}}}(\boldsymbol{\theta}) = \sum_{i=1}^N \frac{\partial \phi(\boldsymbol{\theta}, t_i)}{\partial \boldsymbol{\theta}} \frac{\partial \phi(\boldsymbol{\theta}, t_i)}{\partial \boldsymbol{\theta}} + \sum_{i=1}^N \frac{\partial^2 \phi(\boldsymbol{\theta}, t_i)}{(\partial \boldsymbol{\theta})^2} (\phi(\boldsymbol{\theta}, t_i) - y(t_i)), \quad (\text{A.26})$$

em que se utiliza o resultado obtido em (A.25) e a regra da *derivada do produto*.

Finalmente, com a utilização da definição de matriz Jacobiana (A.19) e com os resultados obtidos em (A.25) e em (A.26), o vetor gradiente e a matriz Hessiana são representados por

$$\mathbf{g}_{c_{MQ}}(\boldsymbol{\theta}) = J_{\phi(t)}(\boldsymbol{\theta})^T f(\boldsymbol{\theta}) \quad (\text{A.27})$$

$$G_{c_{MQ}}(\boldsymbol{\theta}) = J_{\phi(t)}(\boldsymbol{\theta})^T J_{\phi(t)}(\boldsymbol{\theta}) + Q_{\phi}(\boldsymbol{\theta}, t) \quad (\text{A.28})$$

respectivamente, em que $J_{\phi(t)}(\boldsymbol{\theta})$ é a Jacobiana de $\phi(\boldsymbol{\theta})$ e

$$Q_{\phi}(\boldsymbol{\theta}) = \sum_{i=1}^N \frac{\partial^2 \phi(\boldsymbol{\theta}, t_i)}{(\partial \boldsymbol{\theta})^2} (\phi(\boldsymbol{\theta}, t_i) - y(t_i)),$$

Importante destacar a diferença entre as definições de matriz Jacobiana nas equações (A.19) e (A.27). No segundo caso (Gill et al., 1981),

$$J_{\phi(t)}(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \mathbf{g}_{\phi(t_1)}^T \\ \mathbf{g}_{\phi(t_2)}^T \\ \vdots \\ \mathbf{g}_{\phi(t_N)}^T \end{bmatrix} = \begin{bmatrix} \frac{\partial \phi(t_1)}{\partial \theta_1} & \frac{\partial \phi(t_1)}{\partial \theta_2} & \cdots & \frac{\partial \phi(t_1)}{\partial \theta_n} \\ \frac{\partial \phi(t_2)}{\partial \theta_1} & \frac{\partial \phi(t_2)}{\partial \theta_2} & \cdots & \frac{\partial \phi(t_2)}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \phi(t_N)}{\partial \theta_1} & \frac{\partial \phi(t_N)}{\partial \theta_2} & \cdots & \frac{\partial \phi(t_N)}{\partial \theta_n} \end{bmatrix}. \quad (\text{A.29})$$

A.4 Inclinação e curvatura

Boa parte dos métodos de otimização determinísticos prevêm a determinação do mínimo de uma função $c(\boldsymbol{\theta})$ ao longo de uma direção \mathbf{d} a partir de um ponto inicial $\boldsymbol{\theta}_0$. Qualquer ponto ao longo dessa direção pode ser descrito por

$$\boldsymbol{\theta} = \boldsymbol{\theta}_0 + \alpha \mathbf{d} \quad (\text{A.30})$$

em que ao ponto ótimo $\boldsymbol{\theta}^*$ corresponde um *passo* ótimo α^* .

Para a determinação de α^* , tipicamente utilizam-se a primeira e a segunda derivadas da função objetivo em relação a α na direção \mathbf{d} . Nesse contexto, define-se

$$\frac{\partial c(\boldsymbol{\theta})}{\partial \alpha} = \mathbf{d}^T \mathbf{g}_c \quad \text{e} \quad \frac{\partial^2 c(\boldsymbol{\theta})}{\partial \alpha^2} = \mathbf{d}^T \mathbf{G}_c \mathbf{d} \quad (\text{A.31})$$

como sendo a *inclinação* e a *curvatura* de $c(\boldsymbol{\theta})$ em relação a α ao longo da direção \mathbf{d} respectivamente.

A.5 Funções lineares e quadráticas

Uma função linear $l(\boldsymbol{\theta}) : \mathbb{R}^n \rightarrow \mathbb{R}$ pode ser descrita por:

$$l(\boldsymbol{\theta}) = a_1\theta_1 + \cdots + a_n\theta_n + b = \sum_{i=1}^n a_i\theta_i + b = \mathbf{a}^T \boldsymbol{\theta} + b, \quad (\text{A.32})$$

em que $b \in \mathbb{R}$ e $\mathbf{a} \in \mathbb{R}^n$ são constantes.

O cálculo do gradiente de uma função linear implica

$$\nabla l(\boldsymbol{\theta}) = \nabla(\mathbf{a}^T \boldsymbol{\theta} + b) = \nabla(\mathbf{a}^T \boldsymbol{\theta}), \quad (\text{A.33})$$

uma vez que $\nabla b = 0$.

Para a resolução de (A.33), deve-se levar em conta que, para dois vetores \mathbf{u} e \mathbf{v} de mesma dimensão,

$$\nabla(\mathbf{u}^T \mathbf{v}) = \nabla(\mathbf{u}^T) \mathbf{v} + \nabla(\mathbf{v}^T) \mathbf{u}. \quad (\text{A.34})$$

Além disso, deve-se observar que

$$\nabla(\boldsymbol{\theta}^T) = [\nabla\theta_1 \ \cdots \ \nabla\theta_n] = \begin{bmatrix} \frac{\partial\theta_1}{\partial\theta_1} & \frac{\partial\theta_2}{\partial\theta_1} & \cdots & \frac{\partial\theta_n}{\partial\theta_1} \\ \frac{\partial\theta_1}{\partial\theta_2} & \frac{\partial\theta_2}{\partial\theta_2} & \cdots & \frac{\partial\theta_n}{\partial\theta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\theta_1}{\partial\theta_n} & \frac{\partial\theta_2}{\partial\theta_n} & \cdots & \frac{\partial\theta_n}{\partial\theta_n} \end{bmatrix} = I.$$

Com isso, obtém-se o valor do gradiente de $l(\boldsymbol{\theta})$:

$$\nabla l(\boldsymbol{\theta}) = \mathbf{g}_l = \underbrace{\nabla(\mathbf{a}^T)}_{\mathbf{0}} \boldsymbol{\theta} + \underbrace{\nabla(\boldsymbol{\theta}^T)}_I \mathbf{a} = \mathbf{a}. \quad (\text{A.35})$$

Como consequência, tem-se que a Hessiana de uma função linear consiste em uma matriz nula ($G_l = 0$).

No que diz respeito às funções quadráticas, considere-se uma função $\phi(\boldsymbol{\theta}) : \mathbb{R}^n \rightarrow \mathbb{R}$ tal que

$$\phi(\boldsymbol{\theta}) = \frac{1}{2} \boldsymbol{\theta}^T G \boldsymbol{\theta} + \mathbf{a}^T \boldsymbol{\theta} + b, \quad (\text{A.36})$$

em que $b \in \mathbb{R}$, $\mathbf{a} \in \mathbb{R}^n$ e $G \in \mathbb{R}^{n \times n}$ são constantes e G é simétrica.

O gradiente de uma função quadrática é

$$\nabla\phi(\boldsymbol{\theta}) = \nabla\left(\frac{1}{2}\boldsymbol{\theta}^T G\boldsymbol{\theta}\right) + \nabla(\mathbf{a}^T\boldsymbol{\theta}) + \nabla(b). \quad (\text{A.37})$$

Utilizando-se (A.34) com $\mathbf{u} = \boldsymbol{\theta}$ e $\mathbf{v} = G\boldsymbol{\theta}$, tem-se

$$\nabla\left(\frac{1}{2}\boldsymbol{\theta}^T G\boldsymbol{\theta}\right) = \frac{1}{2}(\nabla(\boldsymbol{\theta}^T)G\boldsymbol{\theta} + \nabla((G\boldsymbol{\theta})^T)\boldsymbol{\theta}) = \frac{1}{2}(G + G^T)\boldsymbol{\theta}. \quad (\text{A.38})$$

Mas G é simétrica por construção ($G = G^T$). Com os resultados obtidos em (A.35), obtém-se o gradiente de uma função quadrática como

$$\nabla\phi(\boldsymbol{\theta}) = \mathbf{g}_\phi = G_\phi\boldsymbol{\theta} + \mathbf{a}, \quad (\text{A.39})$$

que consiste em uma função linear e, portanto, torna a matriz Hessiana constante.

Uma importante consequência da equação (A.39) éS que, dados dois pontos $\boldsymbol{\theta}_a$ e $\boldsymbol{\theta}_b$ e os valores do gradiente da função quadrática avaliado nesses pontos, tem-se

$$\mathbf{g}_\phi^a - \mathbf{g}_\phi^b = G_\phi(\boldsymbol{\theta}_a - \boldsymbol{\theta}_b). \quad (\text{A.40})$$

Isso implica que a Hessiana mapeia diferenças entre pontos em diferenças entre gradientes.

A.6 Matrizes

A.6.1 Definições

Definição 1: Uma matriz simétrica M $n \times n$ é definida positiva se, e somente se, qualquer das seguintes condições for verdadeira:

1. todos os autovalores de M são positivos;
2. todos os menores principais iniciais² de M são positivos;
3. existe uma matriz não-singular N $n \times n$ tal que $M = N^T N$.

²Menores principais iniciais: *Leading principal minors* (Chen, 1999).

Definição 1.1: Os menores principais de uma matriz M $n \times n$ são definidos como os determinantes de todas as submatrizes de M cujas diagonais coincidem com a diagonal de M . Por exemplo, se

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}, \quad (\text{A.41})$$

então seus menores principais são:

$$\begin{array}{ccc} m_{11}, & m_{22}, & m_{33}, \\ \left| \begin{array}{cc} m_{11} & m_{12} \\ m_{21} & m_{22} \end{array} \right|, & \left| \begin{array}{cc} m_{11} & m_{13} \\ m_{31} & m_{33} \end{array} \right|, & \left| \begin{array}{cc} m_{22} & m_{23} \\ m_{32} & m_{33} \end{array} \right|, \\ & \left| \begin{array}{ccc} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{array} \right|. \end{array}$$

Definição 1.2: Os menores principais *iniciais* de uma matriz quadrada M $n \times n$ são definidos como os determinantes das submatrizes obtidas com a exclusão das últimas k linhas e colunas de M , com $k = n - 1, \dots, 0$. Por exemplo, se M é definido como em (A.41), então os menores principais iniciais de M são:

$$m_{11}, \quad \left| \begin{array}{cc} m_{11} & m_{12} \\ m_{21} & m_{22} \end{array} \right|, \quad \left| \begin{array}{ccc} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{array} \right|.$$

Definição 2: De forma análoga à Definição 1, uma matriz simétrica M $n \times n$ é semidefinida positiva se, e somente se, qualquer das seguintes condições for verdadeira:

1. todos os autovalores de M são não-negativos;
2. todos os menores principais iniciais de M são não-negativos;
3. existe uma matriz N $n \times n$ singular ou uma matriz N $m \times n$ com $m < n$ tal que $M = N^T N$.

Definição 3: O número de condicionamento $\kappa(M) \geq 1$ de uma matriz H $n \times n$ é definido por

$$\kappa(M) = \|M\|_p \|M^{-1}\|_p \quad (\text{A.42})$$

em que p indica o tipo de norma considerada (1, 2, ∞).

Definição 3.1: Uma matriz M $n \times n$ é dita mal-condicionada se $\kappa(M) \gg 1$. Isso implica M (aproximadamente) singular.

Definição 4: Define-se *esférica* uma matriz diagonal D $n \times n$ com n autovalores repetidos:

$$D = \begin{bmatrix} \lambda & 0 & \cdots & 0 \\ 0 & \lambda & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda \end{bmatrix} \quad (\text{A.43})$$

Definição 5: O *posto* de uma matriz H $m \times n$, $\rho(H)$ é definido como o número de colunas (ou linhas) linearmente independentes de H .

Definição 5.1: Uma matriz H $m \times n$ possui posto completo de colunas se todas as suas colunas são linearmente independentes entre si. O análogo se aplica à definição correspondente às linhas de H .

Definição 6: Definido apenas para matrizes quadradas, o *traço* de uma matriz M $n \times n$ é definido por:

$$\text{Tr}(M) = \sum_{i=1}^n m_{ii} \quad (\text{A.44})$$

A partir dessa definição, as seguintes propriedades do operador traço podem ser verificadas:

$$\text{Tr}(\alpha M) = \alpha \text{Tr}(M) \quad (\text{A.45})$$

$$\text{Tr}(M + N) = \text{Tr}(M) + \text{Tr}(N) \quad (\text{A.46})$$

$$\text{Tr}(MN) = \text{Tr}(NM) \quad (\text{A.47})$$

$$\text{Tr}(\mathbf{v}\mathbf{w}^T) = \mathbf{v}^T \mathbf{w} \quad (\text{A.48})$$

$$\text{Tr}(A\mathbf{v}\mathbf{w}^T) = \mathbf{w}^T A\mathbf{v} \quad (\text{A.49})$$

$$\text{Tr}(MNPQ) = \text{Tr}(NPQM) = \text{Tr}(PQMN) = \text{Tr}(QMNP) \quad (\text{A.50})$$

em que α é um escalar, \mathbf{v} e \mathbf{w} são vetores (coluna) e N , P e Q são matrizes quadradas, da mesma dimensão de M .

A.6.2 Decomposição em valores singulares

Sejam as matrizes $H \in \mathbb{R}^{m \times n}$ e $M \triangleq H^T H$. Tem-se, portanto, que M é $n \times n$, simétrica e semidefinida positiva. Considere-se que M possua r

autovalores positivos. Com isso, os autovalores de M consistem em:

$$\lambda_1^2 \geq \lambda_2^2 \geq \cdots \lambda_r^2 > 0 = \lambda_{r+1} = \cdots = \lambda_n.$$

Se $\tilde{n} = \min(m, n)$, então os valores

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_r > 0 = \lambda_{r+1} = \cdots = \lambda_{\tilde{n}}$$

são denominados *valores singulares* de H e são tipicamente listados em ordem decrescente.

Bibliografia

- Aguirre, L. A. (2004). *Introdução à Identificação de Sistemas: Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, Belo Horizonte, 2 edition.
- Bar-Itzhack, I. Y. (2001). Classification of algorithms for angular velocity estimation. *Journal of Guidance, Control, and Dynamics*, 24(2):214–218.
- Betts, J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207.
- Chen, C.-T. (1999). *Linear System Theory and Design*. Oxford University Press, New York, 3 edition.
- Crista, C. C. T. (2004). Crista inertial measurement unit (imu) - interface/operation document.
- de Vasconcelos, J. A., Adriano, R. L. S., Vieira, D. A. G., Souza, G. F. D., and Azevedo, H. S. (2002). Nsga with elitism applied to solve multiobjective optimization problems. *Journal of Microwaves and Optoelectronics*, 2(6):59–69.
- Deb, K. (2003). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, New York.
- Deb, K. and Agrawal, S. (1999). Understanding interactions among genetic algorithm parameters. *Foundations of Genetic Algorithms*, 5:265–286.
- Dias, A. H. F. and de Vasconcelos, J. A. (2002). Multiobjective genetic algorithms applied to solve optimization problems. *IEEE Transactions on Magnetics*, 38(2):1133–1136.
- Doebelin, E. O. (1990). *Measurement Systems: Application and Design*. McGraw-Hill Book, 4 edition.

- Dorf, R. C. and Bishop, R. H. (2001). *Sistemas de Controle Modernos*. Livros Técnicos e Científicos Editora, 8 edition.
- Fagundes, P. M. (2004). Medição inteligente e monitoramento de pressão em caixa espiral de turbinas hidráulicas e tubos de sucção. Projeto fim de curso, Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, Belo Horizonte, MG, Brasil.
- Fonseca, C. M. and Fleming, P. J. (1998). Multiobjective optimization and multiple constraint handling with evolutionary algorithms ii: Application example. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):38–47.
- Franchois, A. and Pichot, C. (1997). Microwave imaging-complex permittivity reconstruction with a levenberg-marquardt method. *IEEE Transactions on Antennas and Propagation*, 45(5):203–215.
- Gerlach, O. H. (1970). Determination of performance and stability parameters from non-steady flight test manoeuvres. *National business aircraft meeting*, (700236).
- Góes, L. C. S., Hemerly, E. M., Maciel, B. C. O., Neto, W. R., Mendonça, C. B., and Hoff, J. (2004). Aircraft parameter estimation using output-error methods. *Inverse Problems, Design and Optimization Symposium*.
- Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical Optimization*. Academic Press, 1 edition.
- Goldberg, D. E., Korb, B., and Deb, K. (1989). Messy genetic algorithms: Motivation analysis and first results. *Complex Systems*, 5(3):493–530.
- Halliday, D., Resnick, R., and Krane, K. S. (1996a). *Física 1 - Mecânica*, volume 1. Livros Técnicos e Científicos Editora, 4 edition.
- Halliday, D., Resnick, R., and Krane, K. S. (1996b). *Física 2 - Movimento Ondulatório e Gravitação*, volume 1. Livros Técnicos e Científicos Editora, 4 edition.
- He, C., Zheng, Y. F., and Ahalt, S. C. (2002). Object tracking using the Gabor wavelet transform and the golden section algorithm. *IEEE Transactions on Multimedia*, 4(4):528–538.
- Hooke, R. and Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *J. Assoc. Comput. Mach.*, 8:212–229.

- Jategaonkar, R. V. (2001). Estima - a modular and integrated software tool for parameter estimation and simulation of dynamics systems. Manual de usuário, DLR, Braunschweig. Versão 1.0.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*.
- Klein, V. and Batterson, J. G. (1986). Aerodynamic parameters estimated from flight and wind tunnel data. *Journal of Aircraft*, 23(4):306–312.
- Krishnakumar, K., Swaminathan, R., and Montgomery, L. (1993). Multiple optimal solutions for structural control using genetic algorithms with niching. *Journal of Guidance, Control, and Dynamics*, 17(6):1374–1377.
- Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In Schwefel, H. P. and Manner, R., editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496, pages 193–197, Berlin, Germany. Springer-Verlag.
- Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–147.
- Levenberg, K. (1944). A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168.
- Lewis, M., Torczon, V., and Trosset, M. W. (2000). Direct search methods: Then and now. *J. Comp. Appl. Math.*, 124:191–207.
- Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441.
- Mendonça, C. B., Hemerly, E. M., and Curvo, M. (2005). Reconstrução de trajetória de aeronaves com identificação paramétrica em modelo não-linear. Technical report, Empresa Brasileira de Aeronáutica S.A., Av. Brigadeiro Faria Lima, 2170, 12270 901 São José dos Campos, SP.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091.
- Meyer, R. and Christensen, N. (2001). Fast bayesian reconstruction of chaotic dynamical systems via extended kalman filtering. *Physical Review E*, 65:016206.

- Mulder, J. A., Chu, Q. P., Sridhar, J. K., Breeman, J. H., and Laban, M. (1999). Non-linear aircraft flight path reconstruction review and new advances. *Progress in Aerospace Sciences*, 35:673–726.
- Mulder, J. A. and Hollander, J. G. (1981). Status of dynamic flight test technology - model identification for flight simulations. *Business aircraft meeting of the Society of Automotive Engineers*, (810597).
- Narendra, K. S. (1991). Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Transactions On Neural Networks*, 2(2):252–262.
- Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7:308–313.
- Parks, E. K., Bach, R. E., and Wingrove, R. C. (1982). Analysis of the nature and cause of turbulence upset using airline flight records. *Symposium of the Society of Flight Test Engineers*.
- Perry, A. R. (2004). The flightgear flight simulator.
- Powell, M. (1970). A hybrid method for nonlinear equations. *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–144.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992). *Numerical Recipes in C - The Art of Scientific Computing*. Press Syndicate of the University of Cambridge, 2 edition.
- Ranganathan, A. (2004). The Levenberg-Marquardt algorithm.
- Rauw, M. O. (2005). Fdc 1.4 - a simulink toolbox for flight dynamics and control analysis. Technical report, <http://www.dutchroll.org>.
- Ruan, H. (2005). *State Estimation of Chaotic Stochastic Systems with Applications to Chaotic Communication*. PhD thesis, Marquette University, Milwaukee, Wisconsin, U.S.A.
- Santana, D. D. S., Campos, V. A., Furukawa, C. M., and Ama, N. M. (2004). Estimaco de trajetrias utilizando sistema de navegaco inercial *strap-down*. Technical report, Universidade de So Paulo.
- Santos, R. J. (2004). Um curso de geometria analtica e lgebra linear.

- Sartenaer, A. (1996). Automatic determination of an initial trust region in nonlinear programming. Technical report, Department of Mathematics, Facultés Universitaires N.D. de la Paix, 61 rue de Bruxelles, B-5000, Namur, Belgium.
- Shi, Z.-J. and Shen, J. (2006). On memory gradient method with trust region for unconstrained optimization. *Numerical Algorithms*, 41:173–196.
- Spears, W. M. (1998). *The Role of Mutation and Recombination in Evolutionary Algorithms*. PhD thesis, George Mason University.
- Splendey, W., Hext, G. R., and Himsworth, F. R. (1962). Sequential application of simplex designs in optimization and evolutionary operation. *Technometrics*, 4:441–461.
- Stevens, B. L. and Lewis, F. L. (1992). *Aircraft Simulation and Control*. John Wiley & Sons, New York.
- Takahashi, R. H. C. (2004a). Caracterização analítica e formulação do problema de geração das solução de pareto.
- Takahashi, R. H. C. (2004b). Geração de soluções eficientes.
- Takahashi, R. H. C., de Vasconcelos, J. A., Ramirez, J. A., and Krahenbuhl, L. (2003). A multiobjective methodology for evaluating genetic operators. *IEEE Transactions on Magnetics*, 39(3):1321–1324.
- Taylor, L. W. (1982). Applications of parameter estimation in the study of spinning airplanes. *American Institute of Aeronautics and Astronautics*, (82-1309).
- Teixeira, B. O. S., Tôrres, L. A. B., de Oliveira, P. H. I. A., and Aguirre, L. A. (2005). Flight path reconstruction using the unscented kalman filter algorithm. *18th International Congress of Mechanical Engineering*.
- Tekinalp, O. and Bingol, M. (2004). Simulated anealing form missile optimization: Developping method and formulation techniques. *Journal of Guidance, Control, and Dynamics*, 27(4):616–626.
- Travassos, R. H., Gupta, N. K., Iliffe, K. W., and Maine, R. (1980). Determination of an oblique wing aircraft's aerodynamic characteristics. *AIAA Flight Mech. Conf.*, page 608.

- Wang, X., Gallivan, K., and Bramley, R. (1993). Incomplete cholesky factorization with sparsity pattern modification. Technical report, Indiana University / University of Illinois, Bloomington / Urbana.
- Wingrove, R. C. and Bach, R. E. (1983). Analysis of general aviation accident using atc radar records. *Journal of Aircraft*, 20(10):872–876.
- Yamashita, N. and Fukushima, M. (2000). On the rate of convergence of the levenberg-marquardt method. Technical report, Department of Applied Mathematics and Physics, Kyoto University, Kyoto 606-8501, Japan.
- Yokoyama, N. and Suzuki, S. (2005). Modified genetic algorithm for constrained trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 28(1):139–144.
- Zitzler, E. and Thiele, L. (1998). An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Swiss Federal Institute of Technology (ETH), Gloriastrasse 35, CH-8092 Zurich, Switzerland.