

FELIPE ARDITTI

**Algoritmo de alocação de empuxo para sistemas DP considerando
interferência entre propulsores e casco-propulsor**

São Paulo
2013

FELIPE ARDITTI

**Algoritmo de alocação de empuxo para sistemas DP considerando interferência entre
propulsores e casco-propulsor**

Orientador: Prof. Dr. Eduardo Aoun Tannuri

São Paulo

2013

FELIPE ARDITTI

**Algoritmo de alocação de empuxo para sistemas DP considerando interferência entre
propulsores e casco-propulsor**

Orientador: Prof. Dr. Eduardo Aoun Tannuri

São Paulo
2013

FICHA CATALOGRÁFICA

Arditti, Felipe

Algoritmo de alocação de empuxo para sistemas DP considerando interferência entre propulsores e casco-propulsor / F. Arditti. -- São Paulo, 2013. 141 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Sistema de posicionamento dinâmico (Simulação) 2.Hidro-dinâmica (Interação) 3.Propulsão 4.Algoritmos I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.

ARDITTI, F. Algoritmo de alocação de empuxo para sistemas DP considerando interferência entre propulsores e casco-propulsor. São Paulo. 2013. 141 p. (Graduação) Escola Politécnica, Universidade de São Paulo, São Paulo, 2013.

À minha família que sempre
me apoia.

AGRADECIMENTOS

Ao meu orientador Prof. Dr. Eduardo Aoun Tannuri, que durante o desenvolvimento do trabalho sempre me apoiou, incentivou e esteve disposto a ajudar. Além de dividir comigo seu vasto conhecimento na área marítima, ensinou-me a pesquisar e desenvolver um trabalho científico completo.

Ao Prof. Dr. Thiago de Castro Martins pelas enriquecedoras discussões sobre otimização.

À minha família que sempre me apoia.

Aos meus amigos, que sempre estiveram dispostos a discutir e dar sugestões acerca do projeto.

Resumo

Neste trabalho desenvolveu-se um simulador para operações de Posicionamento Dinâmico (DP), em que a alocação de empuxo considera a interação entre propulsores e entre propulsor-casco. O modelo de Alocação de Empuxo desenvolvido está baseado em coordenadas polares, no qual a representação dos fenômenos de interferência é realizada através de funções de eficiência dos propulsores. Diversos aspectos de otimização como desenvolvimento de técnicas simplificadoras e tratamento de não convexidade são explorados. Diferentes técnicas de otimização foram desenvolvidas e comparadas. A partir do algoritmo de otimização que apresentou os melhores resultados em termos de potência consumida pela embarcação e tempo de processamento, desenvolveu-se o simulador, em linguagem apropriada para aplicações práticas. Por fim são realizadas simulações de operações de posicionamento dinâmico considerando-se os fenômenos de interação hidrodinâmicos e as limitações físicas dos propulsores. A análise das simulações ilustra a melhoria da capacidade de manter posição com o algoritmo proposto e a robustez do mesmo.

Palavras chave: Alocação de Empuxo. Interação. Otimização. Algoritmo. Sistema DP. Simulador.

Abstract

This work presents the development of a simulator for Dynamic Positioning (DP) operations. The thrust allocation considers the interaction between thrusters and thruster-hull. The developed thrust allocation model is based on polar coordinates in which the interaction phenomena are represented through efficiency functions of the thrusters. Several optimization aspects such as simplification techniques and treatment of non-convexity are explored. Different optimization techniques were developed and compared. From the optimization algorithm that presented the best results regarding power consumption and processing time the simulator was developed for practical applications. Finally, dynamic positioning operations are simulated considering the hydrodynamic interaction phenomena and the physical limitations of the thrusters. The evaluation of those simulations demonstrates the improvement of stationkeeping capability of the developed allocation algorithm and its robustness.

Keywords: Thrust Allocation. Interaction. Optimization. Algorithm. DP System. Simulator.

Lista de Figuras

Figura 1. 1 – Sistema DP.	17
Figura 1. 2 – Alocação de Empuxo em um navio dotado de 4 propulsores azimutais.....	19
Figura 1. 3 – Mudança de coordenadas: Coordenadas globais (a); Coordenadas do navio (b) 21	
Figura 1. 4 – Arrasto do jato d’água com o casco.....	22
Figura 1. 5 – Interferência entre propulsor e casco, choque de jato d’água com <i>pontoon</i>	23
Figura 1. 6 – Interação entre propulsores.	23
Figura 1. 7 – Propulsor principal.....	25
Figura 1. 8 – Representação do propulsor principal. (a) Sem interferência. (b) Com interferência.	26
Figura 1. 9 – Propulsor em túnel (extraída de Schottel, 2002).	26
Figura 1. 10 – Representação do propulsor principal. (a) Sem interferência. (b) Com interferência.	27
Figura 1. 11 – Propulsor azimutal.	27
Figura 1. 12 – Impossibilidade de três propulsores principais contrabalançarem as forças ambientais.	29
Figura 1. 13 – Impossibilidade de três propulsores em túnel compensarem as forças ambientais.	30
Figura 1. 14 – Condição em que um propulsor azimutal e um principal não compensam as cargas ambientais.	30
Figura 1. 15 – Sistema redundante.	31
Figura 1. 16 – Restrições de caixa de um propulsor azimutal.	34
Figura 1. 17 – Problema de otimização solucionado dentro do domínio viável com restrições ativas (a) e sem restrições ativas (b).....	36
Figura 1. 18 – Comparação entre pontos de mínimo local e mínimo global.	37
Figura 1. 19 – Exemplo de espaços viáveis convexos e não convexos.....	38
Figura 1. 20 – Métodos de solução de problemas de otimização.....	39
Figura 1. 21 – Line Search x Trust Region.....	41
Figura 1. 22 – Possíveis direções para o método Line Search.....	42
Figura 2. 1 – Método de Newton.....	46
Figura 2. 2 – Zona proibida.....	47
Figura 2. 3 – Fatores de eficiência do propulsor em coordenadas polares.	50
Figura 2. 4 – Modelo de interferência entre dois propulsores.	50
Figura 2. 5 – Estrutura do algoritmo.	51
Figura 2. 6 – Arranjo experimental para obtenção da curva de eficiência dos propulsores.	56
Figura 2. 7 – Cálculo da Força efetiva considerando o fenômeno de interferência.....	57
Figura 2. 8 – Desenho esquemático do Navio 1.....	60
Figura 2. 9 – Curvas de eficiência do Navio 1.....	60
Figura 2. 10 – Modelo da Embarcação 2.....	61
Figura 2. 11 – Curvas de eficiência dos propulsores da Embarcação 2.	61
Figura 2. 12 – Ensaio experimental da Embarcação 2.	62
Figura 3. 1 – Dinâmica da implementação inicial do Algoritmo de Alocação de Empuxo.	64

Figura 3. 2 – Dinâmica de otimização do algoritmo de otimização detalhada.....	65
Figura 3. 3 – Diagrama de blocos do FMINCON (otimizador do MATLAB).....	68
Figura 3. 4 – Evolução do propulsor 4 no processo de otimização dentro do espaço viável. ..	69
Figura 3. 5 – Posições angulares dos algoritmos de Alocação de Empuxo para o propulsor 1 em diferentes alocações de empuxo, e sua curva de eficiência.....	71
Figura 3. 6 – Evolução do propulsor 4 no processo de otimização contemplando a direção da força e o aumento de sua eficiência.....	72
Figura 3. 7 – Esquerda alocação do algoritmo de otimização do MATLAB considerando interferências.	73
Figura 3. 8 – Alocação de uma força de controle de -7N em Surge e 7N em Sway com momento nulo, para 3 algoritmos de alocação.....	74
Figura 4. 1 – Razão da Falha no Método da Direção e do Rendimento.....	80
Figura 4. 2 – Direção favorável ou desfavorável do ângulo de máxima eficiência.	80
Figura 4. 3 – Evolução do intervalo de busca do método intuitivo.....	81
Figura 4. 4 – Rotação desfavorável dos propulsores no método intuitivo.....	81
Figura 4. 5 – Solução para o problema de rotação desfavorável dos propulsores no método intuitivo.....	82
Figura 4. 6 – Dinâmica de funcionamento do Trust Region SQP.....	85
Figura 4. 7 – Plataforma de Simulação de Algoritmos de Alocação de Empuxo (AAE).	87
Figura 4. 8 – Excedente Percentual de Potência dos algoritmos desenvolvidos em relação ao algoritmo de otimização do MATLAB (FMINCON).	88
Figura 4. 9 – Comparação do Erro Percentual médio dos algoritmos.....	89
Figura 4. 10 – Comparação do Tempo médio de Processamento dos algoritmos.	90
Figura 4. 11 – Alocação dos diferentes algoritmos simulados para (FSurge = -7N, FSway = 7N e Myaw = 0N).....	92
Figura 4. 12 – Método dos gradientes reduzidos: Situação inicial.....	97
Figura 4. 13 – Método dos gradientes reduzidos: Primeiro passo.	98
Figura 4. 14 – Método dos gradientes reduzidos: Segundo passo.	99
Figura 4. 15 – Método dos gradientes reduzidos: Terceiro passo.....	100
Figura 4. 16 – Método dos gradientes reduzidos: Quarto passo.....	101
Figura 4. 17 – Logica do método do dos gradientes reduzidos.....	102
Figura 4. 18 – Solução considerando-se as desigualdades.	103
Figura 4. 19 – Funcionamento do algoritmo Simplex.....	105
Figura 4. 20 – Processo do Protocolo de Saturação.	107
Figura 4. 21 – Alocação em condições normais.	108
Figura 4. 22 – Alocação em que ocorre saturação dos propulsores.	109
Figura 4. 23 – Atuação do Protocolo de Saturação.....	110
Figura 5. 1 – Modelo da embarcação.	114
Figura 5. 2 – Planta original da embarcação.....	115
Figura 5. 3 – Curvas de eficiência dos propulsores.....	115
Figura 5. 4 – Representação das limitações físicas completas dos propulsores.	118
Figura 5. 5 – Força requerida e gerada para o caso base.	120
Figura 5. 6 – Erro percentual na direção de surge.....	121
Figura 5. 7 – Erro percentual na direção de sway.	121
Figura 5. 8 – Erro percentual na direção de yaw.....	122

Figura 5. 9 – Alocação dos propulsores (empuxo e angulo de azimuth) para o caso base.....	123
Figura 5. 10 – Variação de empuxo e ângulo de azimuth dos propulsores para o caso base.	123
Figura 5. 11 – Força requerida e gerada para o caso Desliga/Liga.	125
Figura 5. 12 – Alocação para o caso Desliga/Liga.....	126
Figura 5. 13 – Variação de empuxo e ângulo de azimuth dos propulsores para o caso Desliga/Liga.	127
Figura 5. 14 – Forças requeridas girando.	128
Figura 5. 15 – Força requerida e gerada para o caso Forças requeridas girando.....	129
Figura 5. 16 – Alocação para o caso Forças requeridas girando.....	131

Lista de Tabelas

Tabela 2. 1 – Fatores de eficiência para o propulsor azimutal.....	49
Tabela 4. 1 – Comparação de Potência requisitada pelos algoritmos.	88
Tabela 4. 2 – Comparação erro dos algoritmos.....	89
Tabela 4. 3 – Comparação do Tempo médio de Processamento dos algoritmos.....	90
Tabela 4. 4 – Definição consistente de quais desigualdades estão ativas.....	95
Tabela 4. 5 – Representação de igualdades e desigualdades.	104
Tabela 4. 6 – Comparação do Trust Region SQP com e sem o desenvolvimento do Simplex e do Active-set.	112
Tabela B. 1: Simulação completa dos Algoritmos de Alocação de Empuxo	1

Sumário

Introdução e Objetivos.....	15
1 – Fundamentos teóricos – Apresentação do Problema.....	17
1.1 – Sistema de posicionamento dinâmico (DP).....	17
1.2 – Alocação de Empuxo	19
1.3 – Padronização de Notação	21
1.4 – Estudo dos fenômenos de interferência	22
1.4.1 – Interferência entre propulsor e casco	22
1.4.2 – Interferência entre propulsores	23
1.5 – Estudo qualitativo de propulsores	25
1.5.1 – Propulsores principais.....	25
1.5.2 – Propulsores em túnel.....	26
1.5.3 – Propulsores Azimutais	27
1.6 – Otimização	29
1.6.1 – Variáveis de Projeto.....	31
1.6.2 – Função Objetivo	32
1.6.3 – Restrições	33
1.6.4 – Domínio Viável	35
1.6.5 – Mínimo Local e Global.....	36
1.6.6 – Convexidade.....	37
1.6.7 – Condições de Optimalidade de 1ª ordem	38
1.6.8 – Métodos de solução de problemas de otimização.....	38
1.6.9 – Linhas gerais de algoritmos de otimização numéricos	40
1.6.10 – Trust Region x Line Search.....	40
2 – Descrição e Análise da Implementação Inicial do Algoritmo de Alocação de Empuxo Levando em conta Interferências.....	43
2.1 – Estado da Arte.....	43
2.1.1 – Função Objetivo Quadrática, sem Fenômenos de Interação	43
2.1.2 – Potencia Exata, sem Fenômenos de Interação.....	44
2.1.3 – Potencia Exata, sem Fenômenos de Interação considerando saturação dos propulsores ...	46
2.1.4 – Potencia Exata, fenômenos de interação evitados e considerando saturação dos propulsores	47
2.2 – Algoritmo de Alocação de Empuxo levando em conta interações hidrodinâmicas	49
2.2.1 – Tratamento Matemático para Curva de Eficiência do Propulsor.....	49
2.2.2 – Estrutura Inicial do Algoritmo	51
2.2.3 – Resultados da Implementação inicial do Algoritmo	58
2.2.4 – Análise da Implementação Inicial do Algoritmo proposto	63
3 – Estudo da Dinâmica Interna do Algoritmo de Otimização	64
3.1 – Análise do Chute Inicial do Problema de Alocação de Empuxo.....	67
3.2 – Análise da Dinâmica Interna do Algoritmo de Otimização do MATLAB	68
3.3 – Análise do comportamento do FMINCON.....	73
3.3.1 – Análise do comportamento em relação interferência entre propulsores	73

3.3.2 – Análise do comportamento para regiões com baixa eficiência	74
3.4 – Relação do Comportamento do Algoritmo com suas vantagens	75
4 – Algoritmo de Alocação de Empuxo.....	76
4.1 – Problema Original de Otimização da Alocação de Empuxo.....	77
4.2 – Novos Algoritmos de Alocação de Empuxo	78
4.2.1 – Método da Direção e do Rendimento	78
4.2.2 – Método Intuitivo	80
4.2.3 – SQP - Potência Quadrática.....	82
4.2.4 – SQP Potência Quadrática + Passo 2	83
4.2.5 – Trust Region SQP.....	84
4.2.6 – Solução Problema de Programação Quadrática.....	85
4.3 – Metodologia de Simulação	86
4.3.1 – Plataforma de Simulação dos Algoritmos de Alocação de Empuxo	86
4.3.2 – Critérios Avaliados na Simulação	87
4.4 – Resultados da Simulação dos Algoritmos.....	88
4.4.1 – Análise de Potência.....	88
4.4.2 – Análise de Precisão.....	89
4.4.3 – Análise do Tempo de Processamento	90
4.5 – Análise Comparativa dos Algoritmos.....	92
4.6 – Seleção do novo Algoritmo de Alocação de Empuxo	93
4.6 – Programação Quadrática.....	94
4.7 – Análise Final	112
5 – Resultados	114
5.1 – Descrição da Embarcação	114
5.2 – <i>Time Domain Simulation</i> e Limitações Físicas.....	117
5.3 – Simulações	119
5.3.1 – Caso Base – Condições ambientais reais	120
5.3.2 – Caso Desliga/Liga.....	125
5.3.3 – Caso de Forças Requeridas Girando – T3 e T5 Cruzam “Zonas Proibidas”	128
6 – Conclusão	132
Referências Bibliográficas.....	134
Anexo A – Solução Analítica da Alocação sem Interferência e função objetivo	
quadrática.....	136
Anexo B – Tabela de comparação dos algoritmos de Alocação de Empuxo	1

Introdução e Objetivos

O sistema de Posicionamento Dinâmico (DP) tem como objetivo manter estática a posição da embarcação para operações em alto-mar, considerando-se as forças ambientais de vento, correnteza e onda. Tendo em vista a expansão de operações *offshore* devido à descoberta do pré-sal, essa tecnologia está em constante evolução.

O sistema DP utiliza os propulsores da embarcação para gerar as forças necessárias para equilibrar a resultante ambiental, funcionalidade chamada Alocação de Empuxo. Os sistemas DP disponíveis no mercado não consideram os fenômenos de interferência entre propulsores e casco-propulsor.

Motivada pela situação atual e pela relevância dos fenômenos de interferência iniciou-se o Projeto THRUST JIP, liderado pela empresa MARIN, que conta com a participação de 17 empresas de petróleo, construção naval e pesquisa, entre elas Petrobrás, Daewoo, Rolls Royce, entre outras.

No núcleo do THRUST JIP encaixa-se o objetivo deste trabalho: desenvolver um pacote computacional para a simulação e análise de operação DP considerando os fenômenos de interferência, cujo âmago é a Alocação de Empuxo. A simulação e a análise da mesma fornecem ferramentas para avaliar a melhoria da capacidade de manter a posição da embarcação.

Projetos acadêmicos de desenvolvimento do Algoritmo de Alocação de Empuxo foram desenvolvidos recentemente (ARDITTI; TANNURI, 2011 e 2012), e fornecem as ferramentas para aumentar o conhecimento técnico em um projeto de engenharia.

A fim de alcançar o objetivo proposto para esta monografia, define-se a seguinte metodologia:

Capítulo 1: Fundamentos teóricos – Apresentação do Problema. Apresenta-se o Sistema DP, a Alocação de Empuxo, os Fenômenos de Interferência, e os conceitos de otimização.

Capítulo 2: Descrição e Análise da implementação inicial do algoritmo de alocação de empuxo levando em conta interferências. Apresentam-se o estado da arte do problema de alocação de empuxo, que levou ao desenvolvimento de um algoritmo de alocação de empuxo que considera os fenômenos de interação hidrodinâmicos. Por fim apresenta-se uma análise do mesmo e suas vantagens em relação às soluções anteriores.

Capítulo 3: Estudo da Dinâmica Interna do Algoritmo de Otimização. Estudo do comportamento intrínseco do algoritmo de otimização, utilizado no desenvolvimento acadêmico da Alocação de Empuxo para desenvolver novas implementações em linguagem apropriada ao pacote computacional proposto.

Capítulo 4: Seleção do Algoritmo de Alocação de Empuxo. Implementação de propostas para o novo Algoritmo de Alocação de Empuxo, comparação no âmbito de eficiência, precisão e tempo de processamento, para seleção do algoritmo central do programa de simulação DP.

Capítulo 5: Simulação DP. Desenvolvimento do programa de simulação de operações de Posicionamento Dinâmico. Apresentação dos conceitos de *time domain simulations* e limitações físicas da embarcação. Discussão dos resultados.

Capítulo 6: Conclusão e recomendações.

1 – Fundamentos teóricos – Apresentação do Problema

1.1 – Sistema de posicionamento dinâmico (DP)

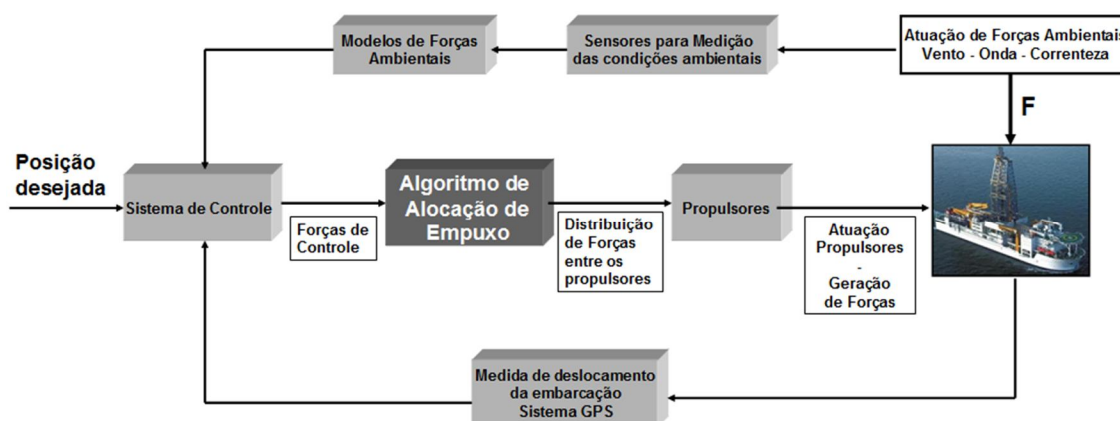
O Sistema de Posicionamento Dinâmico (*Dynamic Positioning* - DP) é uma tecnologia que começou a ser desenvolvida por volta dos anos 60. A expansão das operações petrolíferas para áreas de água profunda (*offshore*) demandava novas tecnologias.

O principal objetivo do sistema DP é manter a posição e orientação do navio, para que o mesmo possa realizar suas operações em alto mar. Note que nestas condições não é possível ou não é economicamente viável utilizar o sistema de ancoragem ou de amarração.

A assistência do Sistema DP à logística do ramo petrolífero é ampla: auxílio para perfuração precisa de poços (inclusive em condições ambientais adversas) e lançamento de dutos; segurança e confiabilidade para operações de alívio, construção e reparos de plataformas de petróleo.

Ilustrado na Figura 1.1 o Sistema DP é constituído de um *loop* que funciona continuamente para garantir o equilíbrio do navio. Quando a embarcação afasta-se da posição desejada (*Set Point*), o sistema de controle define a resultante de forças que deve ser desenvolvida pelo navio, afim de que o mesmo retorne à sua posição desejada.

Figura 1. 1 – Sistema DP.



Fonte: Imagem do navio retirada de <http://farinha0.tripod.com/id126.html>.

A posição da embarcação é definida com auxílio de Sistemas de Referenciamento de Posição, tal como o *DGPS*, e é de extrema importância para garantir a precisão da operação a ser desenvolvida. Além disso, novas tecnologias de sensoriamento permitem estimar as forças ambientais que agem sobre o navio, como ilustrado na Figura 1.1. Com esses dados e a utilização de modelos de força, é possível estimar a resultante ambiental que atua sobre o navio, e adiantar essa informação ao sistema de controle (*feedforward*) afim de aumentar a precisão do Sistema DP.

Não obstante, o Sistema DP também pode atuar em conjunto com sistemas de amarração offshore, no denominado APMS – *Assisted Position Mooring System* (Sorensen 2011). Uma variação do Sistema DP é o *Dynamic Tracking* (DT), que permite à embarcação seguir uma trajetória pré-determinada, considerando a atuação das forças ambientais (Dang, Laheij, 2009).

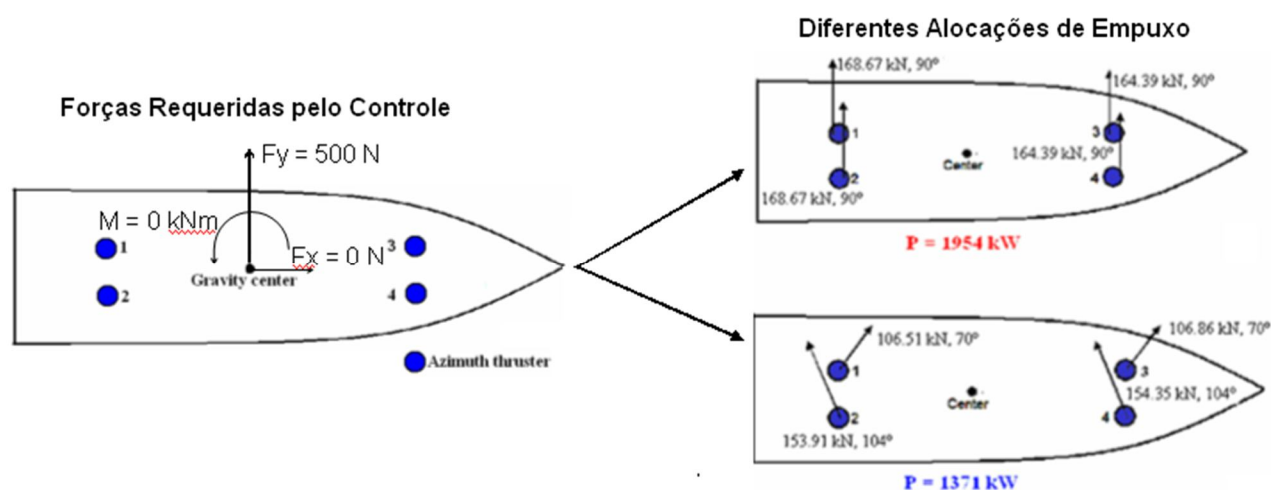
As aplicações mais recentes de Sistema DP sugerem incorporar o mesmo ao Sistema de Navegação, formando um sistema unificado para todas as aplicações da embarcação. Mais detalhes acerca do funcionamento do Sistema DP podem ser encontrados em Sorensen (2011) e Tannuri (2002).

1.2 – Alocação de Empuxo

Assim que calculadas as forças requeridas pelo sistema de controle, as mesmas devem ser geradas pela embarcação. O algoritmo de alocação de empuxo comanda os atuadores da embarcação, ou seja, seus propulsores a fim de gera-las.

A Alocação de Empuxo consiste em coordenar os propulsores de maneira otimizada, ou seja, definir o empuxo e posição (ângulo de azimuth), que cada atuador deve operar. Como ilustrado na Figura 1.2, há várias soluções para a Alocação de Empuxo, porém deve-se otimizar essa escolha a fim de minimizar a potência requerida pelo conjunto de atuadores.

Figura 1. 2 – Alocação de Empuxo em um navio dotado de 4 propulsores azimutais



Fonte: Arditti e Tannuri (2011).

Diferentes Algoritmos e Estratégias de Alocação de Empuxo foram desenvolvidos: Van Daalen et. al. (2011), De Wit (2009), Sorensen (2011), Jenssen e Realfsen (2006), Sordalen (1997), Moberg e Hellstrom (1983). O presente trabalho sugere um Algoritmo de Alocação de Empuxo e compara seu funcionamento e eficiência àqueles já existentes.

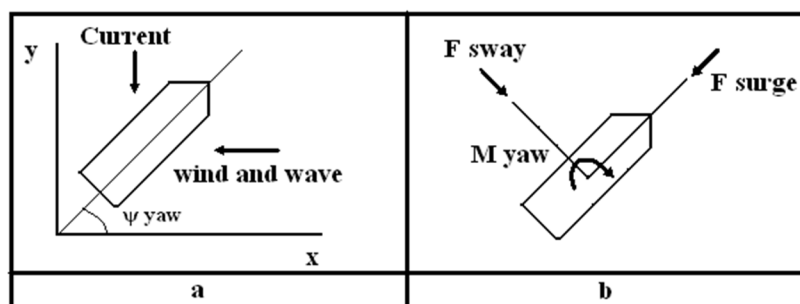
Embora o sistema DP apresente *feedback* (realimentação), o mesmo não considera condições de funcionamento do navio. Portanto, como se espera que a informação calculada no Algoritmo de Alocação de Empuxo seja confiável? A solução é incorporar ao Algoritmo de Alocação de Empuxo dados de funcionamento do navio, ou seja, fazer com que o algoritmo considere a interação entre propulsores e propulsor-casco. Dessa maneira pode-se

tornar o sistema DP mais confiável e econômico, já que permite a busca por uma alocação otimizada, que considere tais interferências.

1.3 – Padronização de Notação

Antes de estudar a interferência entre propulsores e propulsor-casco, é necessário apresentar o conjunto de coordenadas normalmente utilizado quando se trabalha com projetos navais, a fim de facilitar a compreensão das interferências. A Figura 1.3 exemplifica a mudança de coordenadas realizada nestes projetos.

Figura 1. 3 – Mudança de coordenadas: Coordenadas globais (a); Coordenadas do navio (b)



Fonte: Arditti e Tannuri (2011).

Note que o cálculo de resultantes que o navio deve desenvolver fica bastante simplificado se forem utilizadas as coordenadas do navio, uma vez que o ângulo de azimute dos propulsores é referenciado pelo eixo longitudinal do mesmo (*Surge*).

A seguir serão apresentados os fenômenos de interferência que agem sobre os propulsores. A partir do conhecimento das interações entre os propulsores e entre propulsor-casco será possível modelar um novo algoritmo de alocação de empuxo.

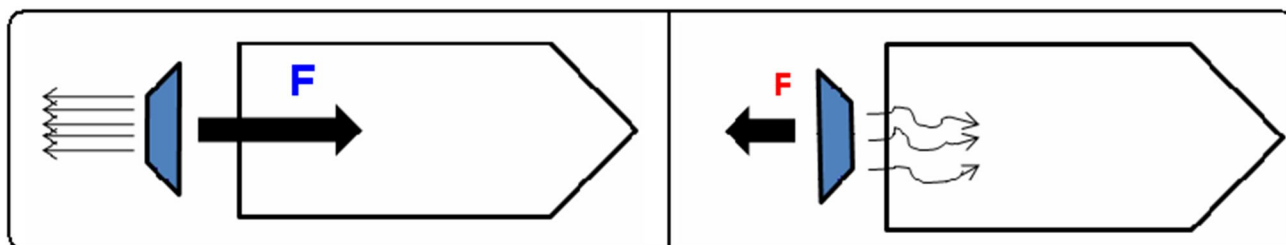
1.4 – Estudo dos fenômenos de interferência

Os principais efeitos de interferência que agem sobre os propulsores são: interação entre propulsor e casco; e interação entre propulsores (EKSTROM; BROWN, 2002).

1.4.1 – Interferência entre propulsor e casco

A interação entre o propulsor e o casco depende, principalmente da forma do casco do navio e da localização do propulsor. O principal efeito que causa a queda de rendimento do propulsor é o arrasto entre o jato d'água e o casco do navio, ilustrado na Figura 1.4.

Figura 1. 4 – Arrasto do jato d'água com o casco.



Outro efeito é causado pela proximidade de cota entre o propulsor e o casco, quanto mais próximo o atuador está do casco do navio, maior será a queda no empuxo líquido, pois haverá menos volume de água disponível ao propulsor.

Também considera-se interferência entre propulsor e casco, o encontro do jato d'água de um propulsor com o pontão (*pontoon*) de uma plataforma. Como ilustrado na Figura 1.5, esse fenômeno deve-se menos ao cisalhamento do jato d'água com casco, do que ao “choque” entre os mesmos, ou seja, a área frontal é muito mais significativa que a área lateral.

Figura 1. 5 – Interferência entre propulsor e casco, choque de jato d'água com *pontoon*.

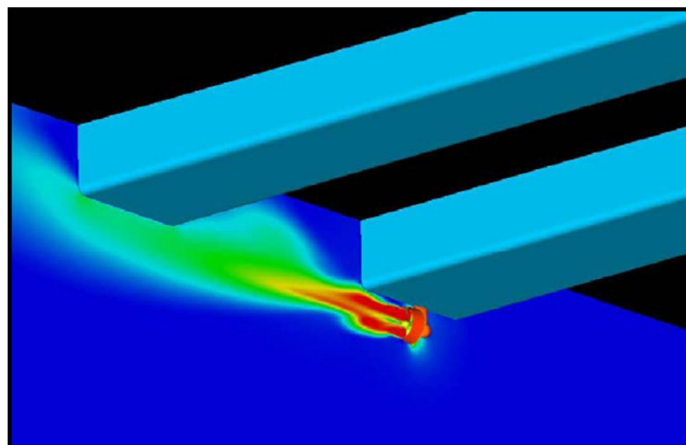


Imagem cedida pela empresa MARIN, de um estudo de CFD (*Computational Fluid Dynamics*).

Outro fenômeno que também deve ser considerado como interferência entre casco e propulsor é o arrasto do jato d'água nas paredes internas do túnel do propulsor em túnel, que será detalhado na seção 1.4.2.

O efeito Coanda está relacionado com o jato d'água proveniente do propulsor. Basicamente este efeito pode ser explicado como a união do jato d'água ao casco do navio, ou seja, ao invés do jato d'água seguir sua trajetória retilínea ele amolda-se à forma do casco. Quanto maior a velocidade do jato d'água menor é esse efeito (EKSTROM; BROWN, 2002).

Estudos recentes sugerem incorporar o leme ao sistema DP (DE WIT, 2009), ou seja, utilizá-lo como um atuador adicional, que permite aos propulsores principais gerarem empuxo no eixo de *Sway*. Embora essa abordagem não seja desenvolvida no presente trabalho, poder-se-ia considerar o leme e o propulsor principal como atuadores, e ponderar a interferência entre ambos para gerar uma resultante não paralela ao eixo de *Surge*.

1.4.2 – Interferência entre propulsores

A interferência entre dois propulsores (Figura 1. 6) ocorre quando os mesmos, normalmente propulsores azimutais, estão próximos. Quando isso ocorre há arrasto do jato d'água de um propulsor sobre outro.

Figura 1. 6 – Interação entre propulsores.

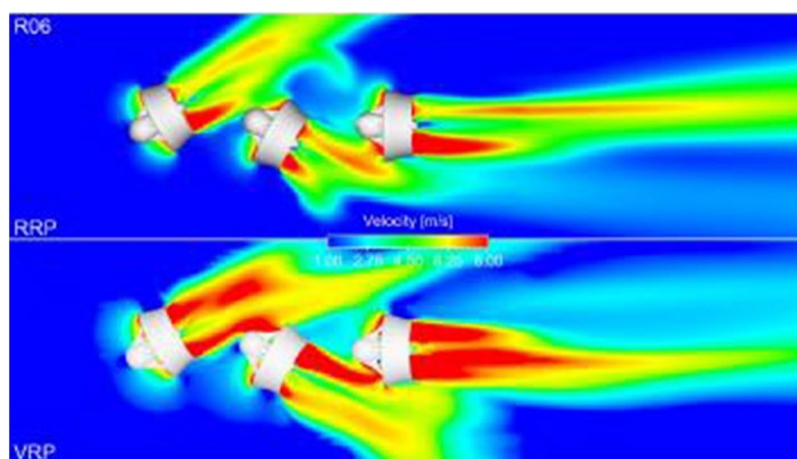


Imagem cedida pela empresa MARIN, de um estudo de CFD.

Considera-se que, para que não haja interferência entre dois propulsores azimutais, os mesmos devem estar a uma distância de 20 diâmetros (Moberg and Hellstrom, 1983). Além disso, quando um propulsor opera sob o jato d'água de outro, a queda de eficiência pode chegar a 40% (LEHN, 1980).

Agora que os principais fenômenos de interferência foram apresentados, deve-se ilustrar qualitativamente, como esse efeito pode ser representado.

1.5 – Estudo qualitativo de propulsores

Existem três tipos de propulsores: Propulsores principais; Propulsores em túnel e Propulsores Azimutais. Os mesmos são apresentados a seguir.

1.5.1 – Propulsores principais

A Figura 1.7 ilustra um propulsor principal com um leme. Lembre-se que não se considerara a utilização de lemes neste trabalho. Além disso as embarcações simuladas neste projeto não eram dotadas desse equipamento

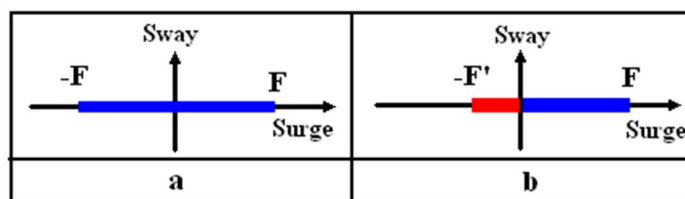
Figura 1. 7 – Propulsor principal.



Fonte: <http://www.osv-modelling.nl/page2.html>.

Os propulsores principais geram empuxo somente no eixo de *Surge*. Normalmente representa-se que tais propulsores apresentam igual eficiência independentemente do sentido. Entretanto essa representação é incorreta, pois quando o empuxo é para trás, o jato d'água vai para frente, e arrasta-se ao longo todo o comprimento do casco do navio, o que gera uma força para frente. Dessa maneira o empuxo líquido diminui (Figura 1.4). Essa representação esta ilustrada na Figura 1.8.

Figura 1. 8 – Representação do propulsor principal. (a) Sem interferência. (b) Com interferência.



1.5.2 – Propulsores em túnel

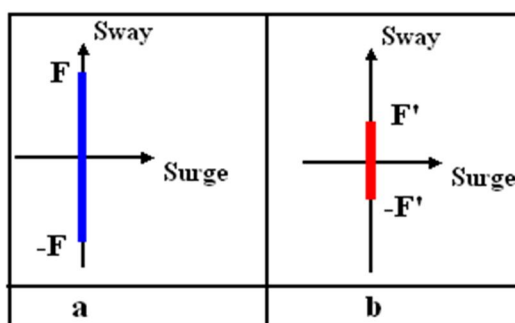
O propulsor em túnel, como o próprio nome explica, localiza-se no interior de um túnel dentro do casco do navio, Figura 1.9. Quando acionado, aspira água de um lado do túnel e expira do outro lado, gerando empuxo.

Figura 1. 9 – Propulsor em túnel (extraída de Schottel, 2002).



A representação para o propulsor em túnel (Figura 1.10) é semelhante àquela do propulsor principal. As diferenças são: o empuxo é no eixo *Sway*, e há perdas de eficiência iguais nos dois sentidos, pois o propulsor em túnel é simétrico e o arrasto d'água nas paredes internas do túnel é igual nos dois sentidos.

Figura 1. 10 – Representação do propulsor principal. (a) Sem interferência. (b) Com interferência.



1.5.3 – Propulsores Azimutais

A Figura 1.11 ilustra um propulsor azimutal. Diferentemente dos propulsores principais e em túnel, o propulsor azimutal tem dois graus de liberdade. É possível comandar o empuxo que o mesmo deve desenvolver e seu ângulo de azimute, uma vez que esse propulsor pode girar 360°.

Figura 1. 11 – Propulsor azimutal.

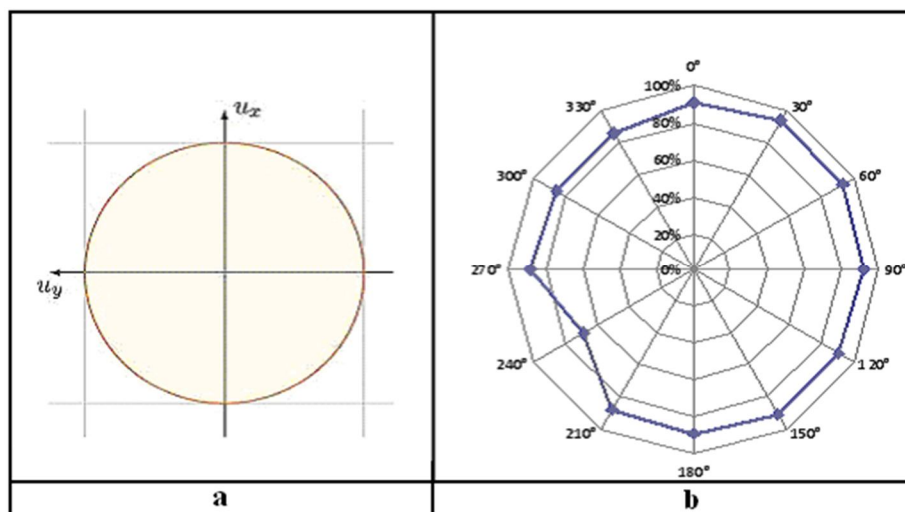


Fonte: extraída de Marine Propulsion International, 1998.

O propulsor azimutal é o mais representativo quando se avalia fenômenos de interferência. A Figura 1.12 (a) mostra a representação sem interferência (caso ideal) de um

propulsor azimuthal. A Figura 1.12 (b) ilustra a interação entre o propulsor e outras partes do navio.

Figura 1. 12 - (a) Representação do propulsor azimuthal sem considerar o fenômeno de interferência; (b) Eficiência do propulsor azimuthal com fenômeno de interferência.



Fonte: De Wit, (2009). Arditti e Tannuri (2011).

De acordo com a Figura 1.12, a eficiência do propulsor é intrinsecamente dependente de sua posição de funcionamento (ângulo de azimuth), portanto deve ser desenvolvido um novo modelo de alocação em coordenadas polares.

Note que em termos de modelagem os propulsores principais e em túnel podem ser modelados como propulsores azimuthais posicionados (sempre) em 0° e 90°, respectivamente.

1.6 – Otimização

Tendo estudado qualitativamente os métodos de representação da eficiência dos atuadores, e como abordar o problema de alocação de empuxo (em coordenadas polares), é necessário conhecer as ferramentas e métodos para otimizar o mesmo.

Antes de apresentar os fundamentos de otimização é necessário compreender a razão do problema de Alocação de Empuxo estar sujeito à mesma.

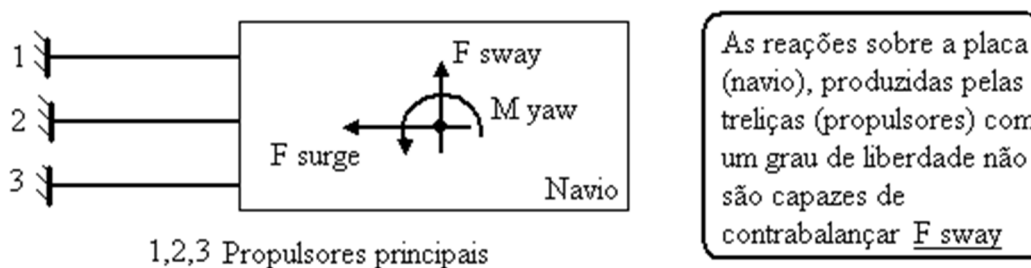
Note que para manter o navio estacionário é preciso contrabalançar as forças ambientais que atuam sobre a embarcação. Pode-se encará-las como vento, onda e correnteza ou como sugerido na Figura 1. 3b: Força de *Surge*, Força de *Sway* e Momento de *Yaw*. Portanto há três graus de liberdade envolvidos no problema.

A premissa básica para a resolução deste problema é: são necessários atuadores com três graus de liberdade linearmente independentes para solucionar o sistema de alocação de empuxo. Isso não significa, necessariamente, que são necessários apenas de três atuadores com um grau de liberdade (propulsores principais ou em túnel) ou de um atuador com um grau de liberdade e outro com dois (propulsor azimutal).

Abaixo se justifica tal premissa. Considere o sistema equivalente que representa o navio como uma placa plana; os propulsores de um grau de liberdade como treliças e; os propulsores azimutais como apoios fixos:

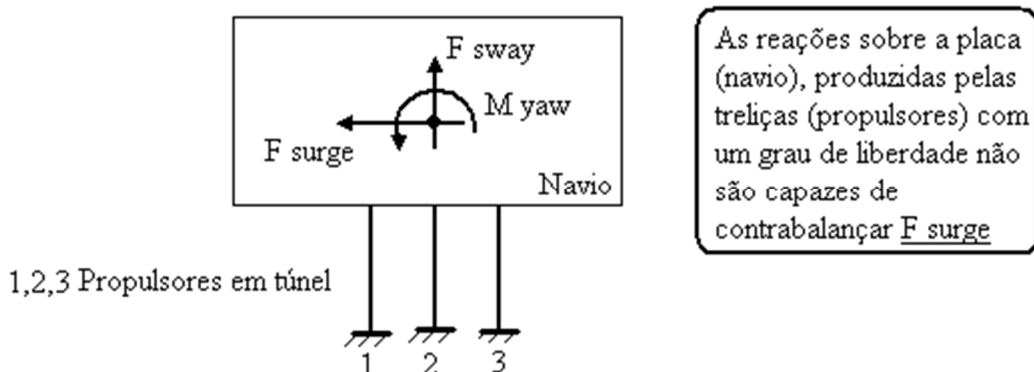
- Com três propulsores principais é possível controlar as forças no eixo de *Surge* e o Momento (se os mesmos não estiverem alinhados ao eixo de *Surge*). Mas é impossível equilibrar a resultante ambiental no eixo de *Sway*, Figura 1.13.

Figura 1. 12 – Impossibilidade de três propulsores principais contrabalançarem as forças ambientais.



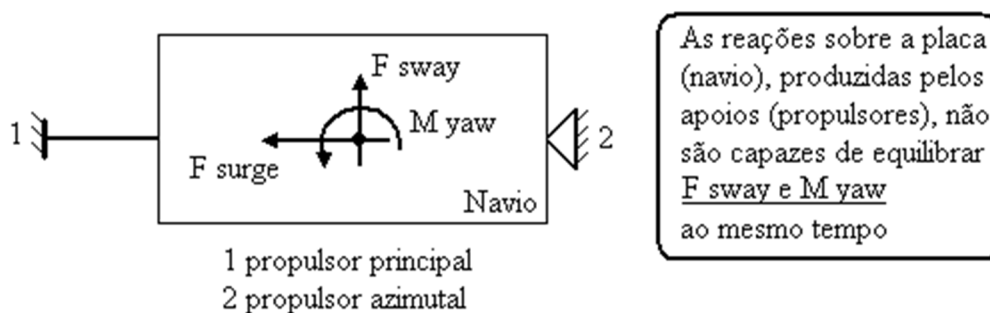
- Com três propulsores em túnel também não é possível equilibrar o sistema. Não há como compensar F_{Surge} , Figura 1.14.

Figura 1. 13 – Impossibilidade de três propulsores em túnel compensarem as forças ambientais.



- Quando se utiliza um propulsor azimutal e um propulsor principal, há condições nas quais não é possível compensar as cargas ambientais, pois as reações na direção de *Sway* e no momento de *Yaw* são linearmente dependentes. Figura 1.15.

Figura 1. 14 – Condição em que um propulsor azimutal e um principal não compensam as cargas ambientais.



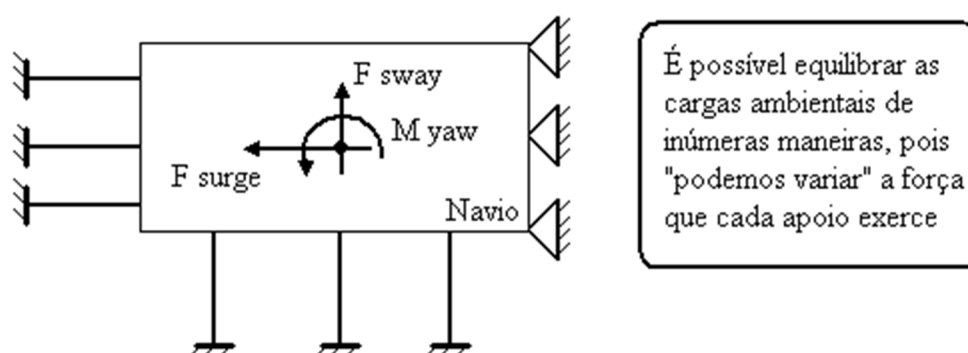
Sem dúvida ao utilizar-se propulsores em túnel junto com principais podemos obter três graus de liberdade linearmente independentes com três atuadores. Também podemos obter três graus de liberdade linearmente independentes deslocando-se para cima ou para baixo o apoio fixo da Figura 1.15.

Embora seja possível resolver o problema de Alocação de Empuxo com as configurações expressas acima, isso não é recomendável pelas seguintes razões:

- Se for utilizado o número mínimo de atuadores, é provável que os mesmos não suportem cargas ambientais relativamente baixas, devido à sua saturação, pois os propulsores têm limites de geração de empuxo.
- Caso haja falha em qualquer um dos propulsores, não será possível compensar as cargas ambientais nem manobrar a embarcação.

Portanto adicionar propulsores às configurações básicas (número mínimo de atuadores, que somam três graus de liberdade LI), sistema torna-se redundante, como mostra a Figura 1.16. Porém a redundância permite que a embarcação opere em condições mais adversas, ou continue funcional mesmo com falha de algum dos propulsores.

Figura 1. 15 – Sistema redundante.



De acordo com a Figura 1.16, um sistema redundante permite um conjunto infinito de soluções. Portanto a melhor maneira de escolher essa solução é otimizar algum quesito. Para tanto é necessário definir um problema de otimização e a ferramenta que será utilizada para resolvê-lo.

1.6.1 – Variáveis de Projeto

O conceito de variáveis de projeto é simples de entender, porém aplicá-lo corretamente é vital para o sucesso da otimização (Silva, 2011). As variáveis de projeto

permitem as manipulações do sistema de otimização. Note que tais manipulações devem estar presentes no escopo do problema.

No Algoritmo de Alocação de Empuxo é possível manipular o empuxo produzido pelos propulsores, bem como ângulo de azimute dos propulsores azimutais para minimizar a potência requerida pelo navio. É importante notar que as variáveis do problema são contínuas, ou seja, podem assumir qualquer valor dentro de um intervalo limitado. Isso auxilia bastante a otimização, que fica bastante simplificada se comparada com outras de variáveis discretas.

Poder-se-ia afirmar que mudar a posição da embarcação pode diminuir as cargas ambientais sobre a mesma, e consequentemente diminuir a potência necessária para manter o navio estático, portanto o ângulo de *Yaw* (Figura 1.3) deveria ser uma variável de projeto. Porém a Alocação de Empuxo diz respeito apenas à geração de forças comandadas pelo controle. Embora variar a posição da embarcação possa, de fato possa minimizar a potência desenvolvida pela mesma, essa tarefa é responsabilidade do sistema de controle ou do comandante da embarcação.

1.6.2 – Função Objetivo

A Função Objetivo é o centro do problema de otimização, pois é a função que define a eficiência do projeto (SILVA, 2011). No caso, desse trabalho o objetivo é minimizar a potência requerida pela embarcação para mantê-la estática.

A diminuição da potência garante economia de combustível, o que significa economia de dinheiro e menor emissão de poluentes. Além disso, se a potência for utilizada de maneira otimizada, a embarcação poderá operar em condições mais adversas.

A principal característica da Função Objetivo é ser função das variáveis de projeto, de maneira direta ou indireta. O trabalho com vários propulsores requer a minimização da potência do conjunto de atuadores. A matriz de pesos deve refletir a maneira como cada propulsor consome potência. Isso ficará mais claro adiante, quando Função Objetivo do algoritmo de alocação de empuxo for definida.

1.6.3 – Restrições

As restrições definem a liberdade do programa de otimização. Ou seja, ao manipular as variáveis de projeto, visando a minimização da função objetivo devem-se respeitar as restrições laterais (de caixa), de desigualdade e de igualdade.

1.6.3.1 – Restrições laterais

As restrições laterais, também conhecidas como restrições de caixa, definem os limites superiores e inferiores das variáveis de projeto. Ou seja, tais restrições estabelecem o intervalo de manipulação das variáveis. Normalmente ilustram-se as restrições de caixa com símbolos: \geq e \leq .

A incorporação dessas restrições neste projeto é simples. Para propulsores de um grau de liberdade, dos quais se controla apenas o empuxo, basta limitar sua variação ao intervalo de empuxo mínimo e empuxo máximo. Esses dados são obtidos diretamente do manual dos propulsores.

Para propulsores azimutais, deve-se limitar seu funcionamento em torno de seus dois graus de liberdade (empuxo e ângulo de azimuth). Isso significa que o propulsor pode operar entre 0° e 360° e; entre seu empuxo mínimo (0 N) e seu empuxo máximo T_{\max} . As Equações 1.1 e 1.2 ilustram as restrições do tipo caixa aplicadas a um propulsor azimutal, e a Figura 1.17 elucida graficamente essas equações.

$$0^\circ \leq \alpha \leq 360^\circ \quad (1.1)$$

$$0 \leq T \leq T_{\max} \quad (1.2)$$

Figura 1. 16 – Restrições de caixa de um propulsor azimuthal.



Note que a restrição de caixa referente à alocação entre 0° e 360° deve ser abandonada para evitar problemas de singularidade.

1.6.3.2 – Restrições de igualdade

As restrições de igualdade definem os requisitos, que o algoritmo de otimização deve respeitar, durante a minimização da Função Objetivo. Ou seja, trata-se de um sistema que contém combinações das variáveis de projeto igualadas a um vetor de valores constantes.

A forma dessas equações é ilustrada no sistema de Equações 1.3, onde $h_i(x)$ representa combinações das variáveis de projeto e V_i são valores constantes:

$$\begin{cases} h_1(x) = V_1 \\ h_2(x) = V_2 \\ \dots \end{cases} \quad (1.3)$$

As restrições de igualdade são utilizadas para equiparar as forças designadas pelo controle ao conjunto de forças geradas pelos atuadores, considerando a perda de eficiência dos mesmos, devido aos fenômenos de interferência.

1.6.3.3 – Restrições de desigualdade

As restrições de desigualdade são bastante semelhantes às restrições de igualdade, o que as diferencia é a utilização do sinal \geq ao invés de $=$. A forma dessas equações é ilustrada no sistema de Equações 1.4, onde $g_i(x)$ representa combinações das variáveis de projeto e W_i são valores constantes.

$$\begin{cases} g_1(x) \geq W_1 \\ g_2(x) \geq W_2 \\ \dots \end{cases} \quad (1.4)$$

As restrições de desigualdade são as que trazem maior dificuldade à otimização, pois podem estar ativas ou inativas (SILVA 2011).

1.6.3.4 – Restrições ativas e inativas

Ao final da rotina de otimização deve-se obter o valor mínimo da Função Objetivo com todas as Restrições respeitadas. Porém haverá restrições de desigualdade (incluindo as laterais) que estarão inativas, ou seja, as variáveis estarão no meio do intervalo de manipulação, o que significa dizer que seu valor não será igual ao seu limite superior ou inferior (para restrições laterais).

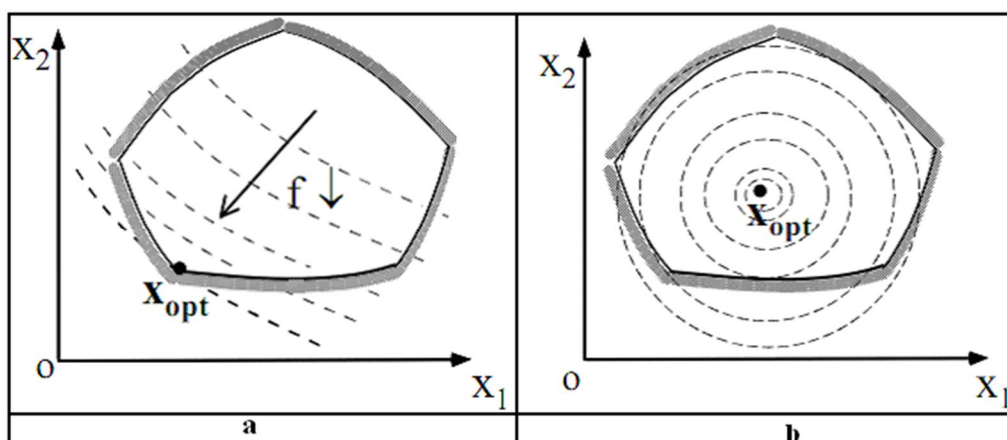
Esse conceito é importante, pois revela qual das restrições tem papel limitante na otimização, o que permite melhorar o desempenho do sistema se for possível relaxar tais restrições. Note que as restrições de igualdade, como o próprio nome revela, deverão estar sempre ativas.

1.6.4 – Domínio Viável

Uma vez definido o problema de otimização: **Variáveis de Projeto, Função Objetivo e Restrições**. O Próximo passo é definir a região onde é possível buscar sua solução (SILVA, 2011). A parte do domínio em que as restrições são respeitadas é denominada domínio viável, e o restante denomina-se domínio inviável.

A Figura 1.18 ilustra problemas de otimização bastante didáticos que contêm todos os assuntos de otimização discutidos até o momento.

Figura 1. 17 – Problema de otimização solucionado dentro do domínio viável com restrições ativas (a) e sem restrições ativas (b).



Fonte: Silva (2011).

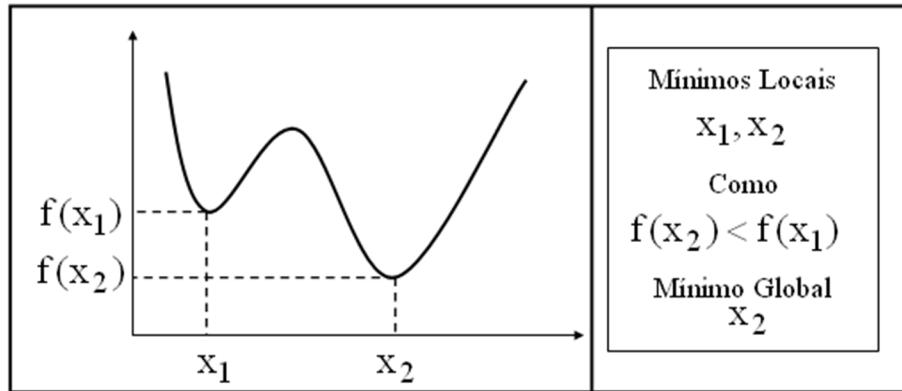
A Figura 1.18a ilustra um problema de otimização bidimensional, cujas variáveis de projeto são X_1 e X_2 . Representam-se as curvas de nível da Função Objetivo. O espaço viável definido pelas restrições é o interior da polítope e a solução ótima do problema é encontrada com restrições ativas.

A Figura 1.18b, bastante semelhante ao problema de otimização da Figura 1.18a tem as mesmas variáveis de projeto, apresenta outra Função Objetivo, e o mesmo espaço viável. Porém a solução, que está no interior do espaço viável não ativa nenhuma das restrições.

1.6.5 – Mínimo Local e Global

A Figura 1.18 apresenta o conceito de mínimo global e local para uma função de uma variável. Os pontos de mínimo local têm o menor valor da função para os pontos ao seu redor. O ponto de mínimo global apresenta o menor valor da função. É importante apresentar estes conceitos, pois os algoritmos numéricos normalmente buscam somente os pontos de mínimo local (Nocedal and Wright 2006).

Figura 1. 18 – Comparação entre pontos de mínimo local e mínimo global.



1.6.6 – Convexidade

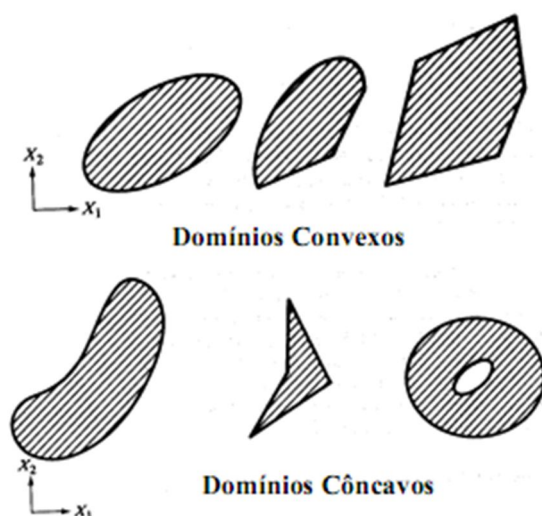
A Convexidade é uma propriedade importante dos problemas de otimização, a mesma garante que o ponto de mínimo local do problema convexo é único. Dessa forma o ponto de mínimo local também é o ponto de mínimo global (Nocedal and Wright 2006). Entretanto para que o problema seja convexo a função objetivo e o domínio devem ser convexos.

Para que a função Objetivo seja convexa sua Matriz Hessiana (Eq. 1.5) deve ser positiva semi-definida, ou positiva definida. Em termos práticos isso significa que os autovalores da Matriz Hessiana devem ser todos maiores ou iguais a 0.

$$\text{Hess} = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \frac{\partial^2 F}{\partial x_1 \partial x_3} \\ \frac{\partial^2 F}{\partial x_1 \partial x_2} & \frac{\partial^2 F}{\partial x_2^2} & \frac{\partial^2 F}{\partial x_2 \partial x_3} \\ \frac{\partial^2 F}{\partial x_1 \partial x_3} & \frac{\partial^2 F}{\partial x_2 \partial x_3} & \frac{\partial^2 F}{\partial x_3^2} \end{bmatrix} \quad (1.5)$$

Para que o domínio seja convexo as restrições de desigualdade (Eq. 1.4) devem ser côncavas. E as restrições de igualdade devem ser lineares. Fisicamente isso significa que em um domínio convexo, se unirmos dois pontos em seu interior por um segmento de reta (\bar{r}), todos os pontos de (\bar{r}) estão no interior do domínio viável.

Figura 1. 19 – Exemplo de espaços viáveis convexos e não convexos.



Fonte: adaptado de Silva (2011).

1.6.7 – Condições de Optimalidade de 1ª ordem

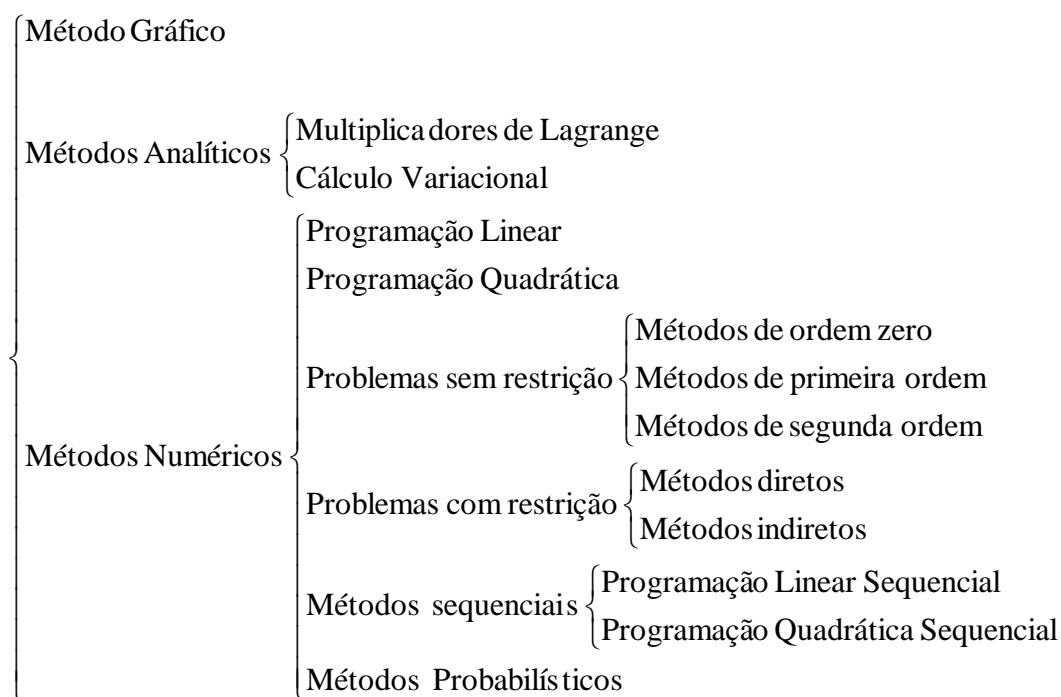
As condições de necessárias para um ponto ser candidato a mínimo local em problemas com restrições são as condições de Karuch-Kuhn-Tucker (KKT), Eq. 1.6. No caso de um problema convexo, caso essas condições sejam respeitadas, encontrar-se-á o ponto de mínimo global. Note que as condições oferecem as equações necessárias para resolver o problema de otimização, e dessa forma podem ser utilizadas para obter-se a solução analítica do mesmo.

$$\begin{aligned}
 \nabla_x L(x^*, \lambda^*) &= 0 \\
 h(x^*) &= 0 \\
 g(x^*) &\geq 0 \\
 \lambda^* &\geq 0
 \end{aligned}
 \tag{1.6}$$

1.6.8 – Métodos de solução de problemas de otimização

Existem vários métodos de solucionar problemas de otimização, que são apresentados na Figura 1.21. A seguir uma breve discussão acerca dos mesmos é realizada. Para maiores informações consulte Silva (2011) e Nocedal e Wright (2006).

Figura 1. 20 – Métodos de solução de problemas de otimização.



O método gráfico consiste em representar o espaço viável (as restrições) e curvas de nível da Função Objetivo e buscar os pontos ótimos (Figura 1.18). Embora seja uma metodologia eficiente é muito simplória e não permite solucionar problemas com mais de dois graus de liberdade.

Os métodos analíticos são os mais consistentes, ou seja, seus resultados são confiáveis, pois estão embasados em soluções puramente matemáticas. Entretanto se o problema de otimização for moderadamente complexo não é possível utilizar tais métodos. Destaca-se nessa classe: o método dos multiplicadores de Lagrange e o Cálculo Variacional, bastante utilizado em problemas estruturais.

Os métodos numéricos apresentam uma gama de técnicas de otimização. O método de programação linear é bastante difundido, porém só é aplicável para problemas de otimização lineares (com Função Objetivo e Restrições lineares).

Os problemas sem restrição fornecem grande liberdade de solução. A diferença entre os métodos de ordem zero, primeira ordem e segunda ordem é a quantidade de informações da Função Objetivo. Nos métodos de ordem zero só há valores da Função Objetivo, enquanto no método de primeira ordem dispõe-se também do gradiente da mesma.

Os problemas de otimização não lineares com restrições são bastante complexos. Há duas abordagens para resolvê-los: caminhar dentro do espaço viável com informações sobre o gradiente da Função Objetivo e verificar se a cada passo houve melhora na solução (métodos diretos) ou; transformar um problema com restrições em um problema sem restrições (métodos indiretos), e aplicar a metodologia de problemas sem restrição para resolvê-lo.

Os métodos sequenciais aproximam as não linearidades das equações por funções lineares. Dessa forma resolvem-se problemas simplificados em intervalos confiáveis (onde a aproximação de Taylor é válida). Soluciona-se sucessivamente o problema até que haja convergência.

Os métodos probabilísticos tentam solucionar os problemas de otimização através de soluções randômicas. Também há uma vertente desses métodos que aposta em basear-se em fenômenos da natureza, como a genética, para solucionar o problema de otimização.

Embora seja um pouco cedo para justificar a escolha do algoritmo de otimização por um método numérico, adianta-se que esse é a melhor metodologia para lidar com o problema da alocação de empuxo considerando a complexidade dos fenômenos de interação. A justificativa para a escolha deste método esta na seção 2.2.9.

1.6.9 – Linhas gerais de algoritmos de otimização numéricos

Em linhas gerais os algoritmos de otimização numéricos funcionam da seguinte maneira. Inicia-se com um ponto (“chute”) inicial e procede-se com uma sequência de melhores estimativas para o problema (iterações). Geralmente as novas iterações dependem da função objetivo, da última iteração e em alguns casos de iterações anteriores. O algoritmo continua as iterações até que não haja como fazer progresso ou até que se chegue a uma solução com suficiente precisão (Nocedal and Wright 2006).

Basicamente existem duas estratégias para realizar as iterações: *Trust Region* e *Line Search*. Note que existem vários algoritmos de otimização numérico, mas eles são variações desses modelos.

1.6.10 – Trust Region x Line Search

A estratégia Trust Region (Fig. 1.22) consiste em definir uma região (normalmente circular) em torno da última solução. Note que o raio da região é o tamanho máximo do passo

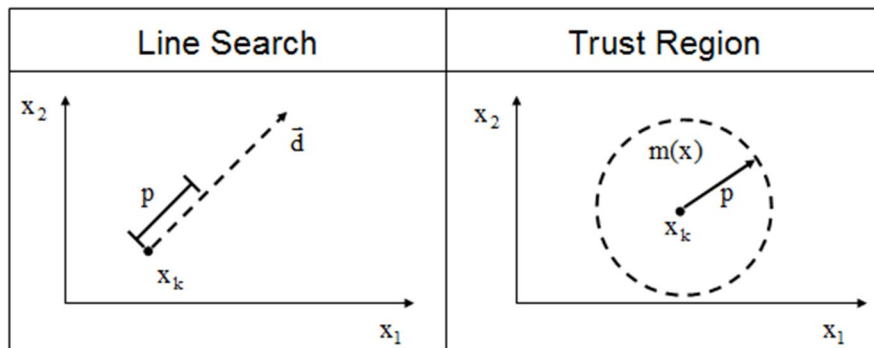
para a próxima iteração. A estratégia de definir uma região confiável permite que a função objetivo e as restrições sejam aproximadas por funções mais simples. Portanto basta resolver o problema ilustrado na Eq. 1.7, em que m_k é uma aproximação da função objetivo (Nocedal and Wright 2006).

$$\min m_k(x_k + p) \quad (1.7)$$

Na estratégia Line Search (Fig. 1.22) escolhe-se uma direção que minimize a função objetivo \vec{d} e define-se o passo que se dá nessa direção a fim de resolver o problema descrito na Eq. 1.8.

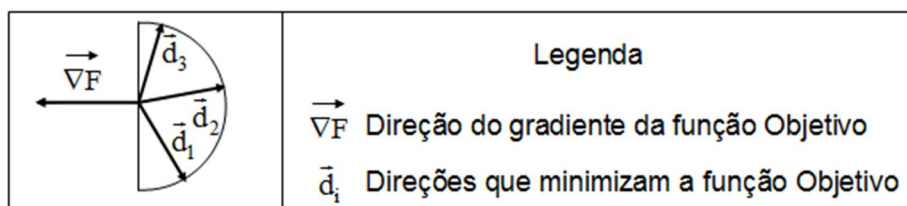
$$\min f(x_k + p \cdot \vec{d}) \quad (1.7)$$

Figura 1. 21 – Line Search x Trust Region.



Note que as direções que minimizam a função objetivo estão contidas na direção contrária ao gradiente da função podendo-se variar 90° em relação à mesma, conforme ilustrado pela Fig. 1.23 (Nocedal and Wright 2006).

Figura 1. 22 – Possíveis direções para o método Line Search



2 – Descrição e Análise da Implementação Inicial do Algoritmo de Alocação de Empuxo Levando em conta Interferências

Inicialmente apresenta-se o estado da arte da alocação de empuxo e seu desenvolvimento. Esse desenvolvimento levou a alocação de empuxo a um novo patamar, no qual os fenômenos de interação hidrodinâmicos que agem sobre os propulsores são considerados. O algoritmo de alocação de empuxo que considera tais interações é descrito e analisado a fim de verificar seu funcionamento e vantagens, o que permitira aplicá-lo num software para simulação de operações de posicionamento dinâmico.

2.1 – Estado da Arte

A alocação de empuxo é um assunto maduro, portanto o problema e suas soluções evoluíram de maneira consistente com o crescimento do mercado de posicionamento dinâmico. Diferentes representações e abordagens são apresentadas nessa seção. A evolução do problema e de suas soluções é apresentada de maneira a ilustrar a evolução de sua complexidade.

2.1.1 – Função Objetivo Quadrática, sem Fenômenos de Interação

Esse problema foi o primeiro passo no desenvolvimento de algoritmos de alocação de empuxo. A função objetivo que representa a potencia consumida pelos propulsores, ilustrada na Eq. 2.1 é simplificada para uma função quadrática. A mesma é uma aproximação da relação real entre empuxo e consumo energético. c_i representa a constante de potencia e T_i o empuxo desenvolvido por cada propulsor.

$$P = \sum_{i=1}^N c_i \cdot T_i^2 \quad (2.1)$$

As restrições impostas a este problema são a geração das forças requeridas (F_x, F_y, M_z) pelo sistema de controle, sem considerar os fenômenos de interação. F_{x_i} e F_{y_i} são as forças desenvolvidas por cada propulsor, x_i e y_i são a posição longitudinal e lateral do propulsores, respectivamente.

$$R: \begin{cases} \sum_{i=1}^N F_{x_i} - F_x = 0 \\ \sum_{i=1}^N F_{y_i} - F_y = 0 \\ \sum_{i=1}^N F_{y_i} \cdot (x_i) - \sum_{i=1}^N F_{x_i} \cdot (y_i) - M_z = 0 \end{cases} \quad (2.2)$$

A solução deste problema é bastante simples e conhecida. Utiliza-se o método dos multiplicadores de lagrange, a fim de obter um sistema linear que é facilmente resolvido. Quando os propulsores são iguais é possível utilizar diretamente a matriz pseudo-inversa para resolver esse sistema.

Superficialmente, o método dos multiplicadores de lagrange une a função objetivo às restrições, multiplicando as mesmas por novas variáveis (λ), essa nova função é denominada lagrangeano (L), Eq 2.3. Isso relaxa o problema uma vez que não há mais equações de igualdade. O passo seguinte é definir a condição de otimização, representada na Eq. 2.4, na qual o gradiente do lagrangeano deve ser 0 (X são as variáveis de projeto).

$$L = Obj - \lambda_i \cdot h_i \quad (2.3)$$

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial \lambda} = 0 \quad (2.4)$$

2.1.2 – Potencia Exata, sem Fenômenos de Interação

O segundo passo na evolução de problemas de alocação de empuxo foi utilizar a relação exata entre o empuxo desenvolvido e a potencia necessária para gera-lo, Eq. 2.5. Note

que essa relação é válida para as condições em que a embarcação está numa condição estacionária e não há correnteza (*bollard pull*).

$$P = \sum_{i=1}^N c_i \cdot T_i^{\frac{3}{2}} \quad (2.5)$$

As restrições são as mesmas que as representadas na Eq. 2.2. Ou seja, os fenômenos de interação hidrodinâmicos não são considerados.

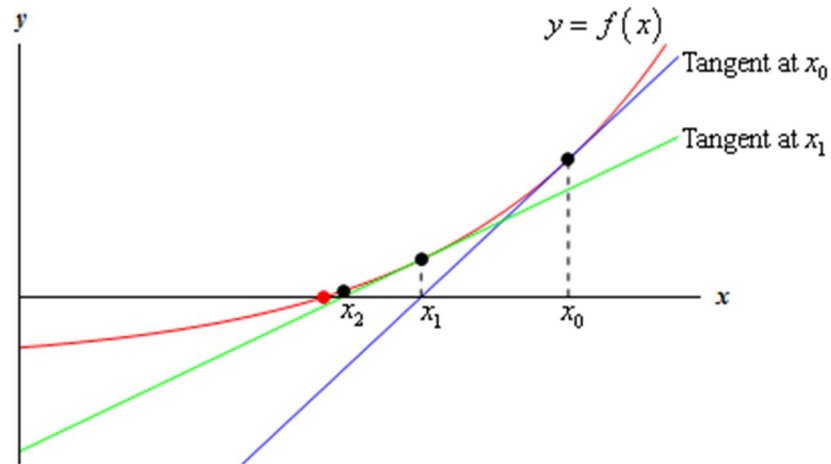
Esse problema também é bastante conhecido e apresenta soluções eficientes:

Aproximar a potência por uma equação quadrática (Eq. 2.6), e resolver esse problema como descrito na seção 2.1.1.

$$P = a_0 + a_1 \cdot T + a_2 \cdot T^2 \quad (2.6)$$

Utilizar o método de Newton. O método de Newton representado na Figura 2.1, é aplicado no sobre o gradiente do lagrangeano (Eq. 2.4), pois para obter-se um ponto de mínimo o mesmo deve ser nulo. Superficialmente, o método de Newton consiste em: a cada iteração dar um passo na direção contrária ao gradiente da função (Hessiana do Lagrangeano), até que o gradiente seja nulo ou até que um critério de parada seja atingido.

Figura 2. 1 – Método de Newton.



2.1.3 – Potencia Exata, sem Fenômenos de Interação considerando saturação dos propulsores

A potencia é considerada conforme a Eq. 2.5 (relação exata).

As restrições são as mesmas que as representadas na Eq. 2.2.

A saturação dos propulsores é representada conforme a Eq. 2.7, que ilustra o empuxo máximo que cada propulsor pode gerar.

$$I: \{T_i \leq T_{max_i} \quad (2.7)$$

A solução convencional a este problema de saturação é o pós-processamento. Inicialmente o problema é resolvido sem que sejam consideradas as desigualdades (conforme um dos métodos apresentados em 2.1.2). Avalia-se caso um ou mais propulsores devem gerar um empuxo superior à sua saturação. Caso isso ocorra, define-se que esse propulsor desenvolveu seu empuxo máximo, e uma nova alocação de empuxo é realizada, considerando-se como um dado o empuxo desenvolvido pelo propulsor saturado.

2.1.4 – Potencia Exata, fenômenos de interação evitados e considerando saturação dos propulsores

A potencia é considerada conforme a Eq. 2.5 (relação exata).

As restrições são as mesmas que as representadas na Eq. 2.2, pois as regiões com interação hidrodinâmica são evitadas.

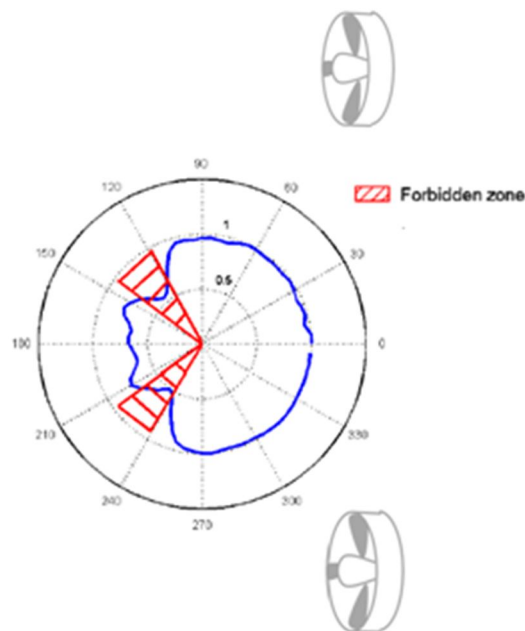
A saturação dos propulsores é representada conforme a Eq. 2.7.

E os propulsores não podem ser posicionados em regiões onde ocorre significativa interação hidrodinâmica, chamada zona proibida. A Eq. 2.8 a representação matemática de uma zona proibida de 20° em torno do ângulo $\alpha_{forbidden}$.

$$I: \left\{ (\alpha_i - \alpha_{forbidden})^2 \geq 10^2 \right. \quad (2.8)$$

A Figura 2.2 ilustra a eficiência de um propulsor, destacando as zonas proibidas, onde a eficiência é significativamente inferior devido à interação do mesmo com outros propulsores.

Figura 2. 2 – Zona proibida.



A solução deste problema é bastante semelhante ao método de pós-processamento apresentado na seção 2.1.3. Inicialmente resolve-se o problema de alocação independentemente da desigualdade 2.5. Caso algum dos propulsores seja posicionado no interior de sua respectiva zona proibida, o mesmo é direcionado para a fronteira mais próxima da mesma, e essa informação é considerada constante na alocação seguinte.

Nota-se, portanto que os fenômenos de interação são considerados na alocação de empuxo, ao evita-los. Embora essa solução evite interações hidrodinâmicas significativas, a mesma não considera de fato a eficiência do propulsor em diferentes ângulos de azimute nem os fenômenos hidrodinâmicos em si, o que poderia melhorar significativamente a precisão do sistema DP. A seção 2.2 apresenta o algoritmo de alocação de empuxo desenvolvido por Arditti e Tannuri (2011 e 2012), que de fato representa e considera as interações hidrodinâmicas entre propulsores e entre propulsor e casco.

2.2 – Algoritmo de Alocação de Empuxo levando em conta interações hidrodinâmicas

2.2.1 – Tratamento Matemático para Curva de Eficiência do Propulsor

A fim de implementar os dados de interferência no algoritmo de alocação de empuxo, utilizam-se fatores de eficiência referentes a cada ângulo de azimuth do propulsor azimuthal. Como ilustrado no estudo qualitativo de interferência isso é necessário, pois a eficiência é função da posição angular do propulsor.

$$\eta = f(\alpha) \quad (2.9)$$

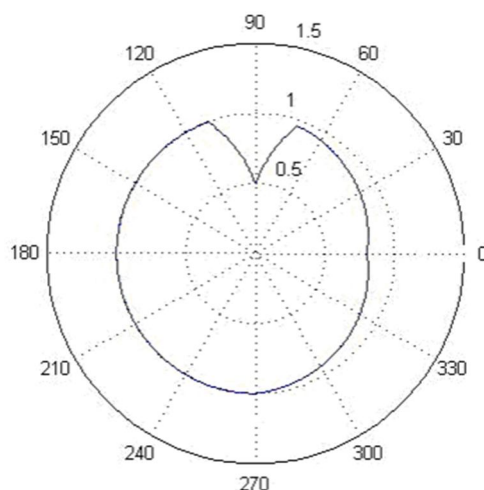
A Tabela 2.1 e Figura 2.3 ilustram a aplicação desse método.

Tabela 2. 1 – Fatores de eficiência para o propulsor azimuthal.

Ângulo	Eficiência	Ângulo	Eficiência	Ângulo	Eficiência	Ângulo	Eficiência
0°	0,80	90°	0,50	180°	1,00	270°	1,00
10°	0,82	100°	0,75	190°	1,00	280°	0,98
20°	0,84	110°	1,00	200°	1,00	290°	0,96
30°	0,87	120°	1,00	210°	1,00	300°	0,93
40°	0,89	130°	1,00	220°	1,00	310°	0,91
50°	0,91	140°	1,00	230°	1,00	320°	0,89
60°	0,93	150°	1,00	240°	1,00	330°	0,87
70°	0,96	160°	1,00	250°	1,00	340°	0,84
80°	0,75	170°	1,00	260°	1,00	350°	0,82

Fonte: Arditti e Tannuri (2011).

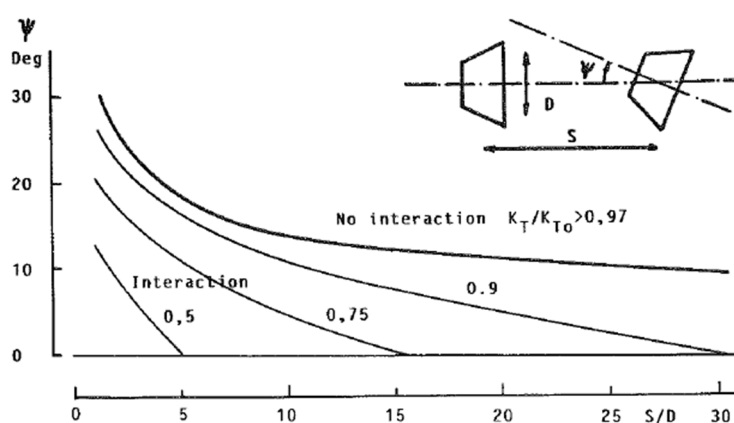
Figura 2. 3 – Fatores de eficiência do propulsor em coordenadas polares.



Fonte: Arditti e Tannuri (2011).

A curva de eficiência da Fig. 2.3 foi construída a partir de dados bibliográficos, porém poder-se-ia utilizar um modelo para construí-la, como é descrito na seção 2.2.2.7. Para interações entre propulsores foi utilizado o modelo sugerido por Moberg e Hellstrom (1983); ilustrado na Figura 2.4.

Figura 2. 4 – Modelo de interferência entre dois propulsores.



Fonte: Moberg e Hellstrom (1983).

Para iterações entre propulsor e casco o modelo de interferência adotado é descrito em Ekstrom e Brown (2002) e Moberg e Hellstrom (1983). A perda de eficiência quando o jato d'água percorre todo o casco é considerada como de 20%, e quando o jato d'água é

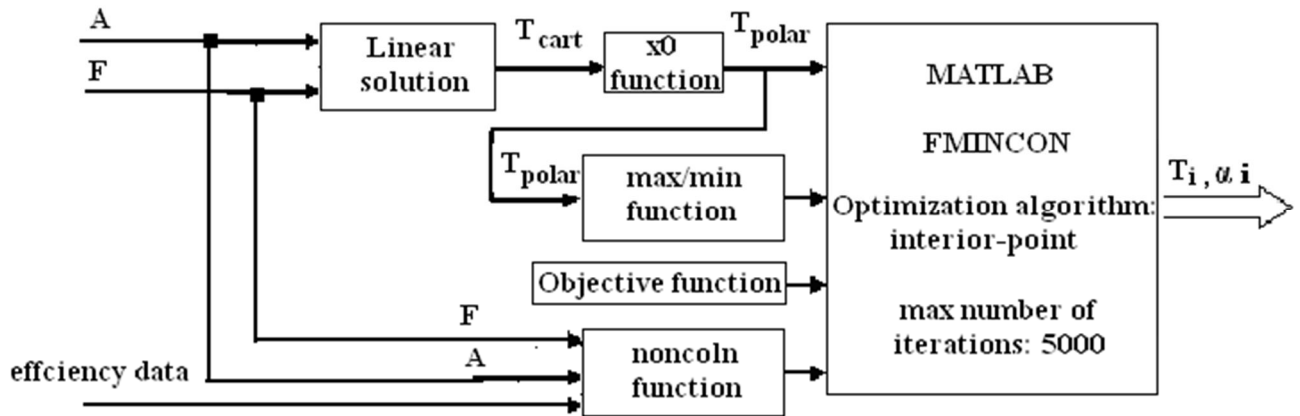
transversal ao casco não há queda na eficiência do propulsor. Note que os efeitos de interferência foram somados (Figura 2.3), pois não são mutuamente excludentes.

Após a definição do modelo de tratamento da eficiência dos propulsores, curva de eficiência, apresenta-se a estrutura do Algoritmo de Alocação Empuxo sugerido.

2.2.2 – Estrutura Inicial do Algoritmo

A estrutura do Algoritmo de Alocação de Empuxo (Arditti and Tannuri 2011) implementado inicialmente é apresentada na Figura 2.5. Cada bloco será explicado a seguir:

Figura 2. 5 – Estrutura do algoritmo.



Fonte: Arditti e Tannuri (2011).

2.2.2.1 – Vetores F , T_{cart} e T_{polar}

O vetor F contém as forças requeridas pelo sistema de controle. As forças geradas pelo conjunto de atuadores da embarcação, levando-se em conta os fenômenos de interferência, deve igualar esse vetor.

$$\mathbf{F} = \begin{bmatrix} F_{surge} \\ F_{sway} \\ M_{yaw} \end{bmatrix} \quad (2.10)$$

\mathbf{T}_{cart} contém a solução inicial da alocação em coordenadas cartesianas para cada propulsor.

$$\mathbf{T}_{\text{cart}} = \begin{bmatrix} F_{\text{surge}} \\ F_{\text{sway}} \end{bmatrix} \quad (2.11)$$

$\mathbf{T}_{\text{polar}}$ contém a solução inicial em coordenadas polares de cada propulsor. É o “chute inicial” do algoritmo de alocação.

$$\mathbf{T}_{\text{polar}} = \begin{bmatrix} T_{\text{empuxo}} \\ \alpha_{\text{angulo de trabalho}} \end{bmatrix} \quad (2.12)$$

2.2.2.2 – Matriz A

A Matriz \mathbf{A} relaciona o empuxo de cada propulsor (\mathbf{T}_{cart}) com sua contribuição ao empuxo resultante em cada direção, e depende unicamente da localização relativa ao centro de gravidade da embarcação (DE WIT, 2009). É de grande importância ao algoritmo de alocação de empuxo, pois contribui para a construção do sistema de forças geradas pelos atuadores da embarcação.

A primeira linha da matriz representa a direção *Surge*; a segunda linha a direção *Sway* e; a terceira linha o momento de *Yaw*. Ilustra-se na Eq. 2.13 a construção da matriz \mathbf{A} para um propulsor azimutal, onde X e Y representam a localização do propulsor nas coordenadas da embarcação. A matriz tem duas colunas, pois o propulsor gera empuxo nas direções de *Surge* e *Sway*, e é mais simples representá-lo como a soma de um propulsor principal e um propulsor em túnel.

$$\mathbf{A}_A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -Y & X \end{bmatrix} \quad (2.13)$$

2.2.2.3 – Linear Solution

Essa função resolve o problema de alocação sem levar em conta o problema de interferência. É o “chute inicial” necessário ao problema de otimização. *pinv* gera a matriz pseudo-inversa.

$$\mathbf{T}_{cart} = pinv(\mathbf{A}) \times \mathbf{F} \quad (2.14)$$

A solução \mathbf{T}_{cart} gera a alocação com menor consumo de potência para a Equação 2.15 (como ilustrado no anexo B), que é bastante similar com consumo de potência real apresentado na seção da Função Objetivo, ou seja, é um bom "chute" inicial. n representa o número de propulsores.

$$P(power) = \sum_{i=1}^n (T_i)^2 \quad (2.15)$$

Algoritmos numéricos de otimização requerem uma “solução inicial”, pois trabalham em espaços N dimensionais (onde N é o número de variáveis relevantes ao problema), e através de iterações a melhor solução é encontrada.

$$N = N^o(prop\ principais) + N^o(prop\ em\ túnel) + 2 \cdot N^o(prop\ Azimutais) \quad (2.16)$$

2.2.2.4 – X0 function

Essa função transforma a solução as coordenadas cartesianas em coordenadas polares (Equação 2.17). No caso de um propulsor azimutal, a função recebe $FSurge$ e $FSway$ e devolve o empuxo total T (Equação 2.18), e o ângulo de trabalho do propulsor α (Equação 2.19). Para propulsores principais e propulsores em túnel, essa mudança de coordenadas não é necessária, pois a eficiência dos mesmos depende apenas do sentido do empuxo, portanto seu sinal (+’ significa empuxo para frente; -’ significa empuxo à ré).

$$\mathbf{T}_{\text{polar}} = X0(\mathbf{T}_{\text{cart}}) \quad (2.17)$$

$$T = \sqrt{F_{\text{surge}}^2 + F_{\text{sway}}^2} \quad (2.18)$$

$$\alpha = \arctg\left(\frac{F_{\text{sway}}}{F_{\text{surge}}}\right) \quad (2.19)$$

2.2.2.5 – max/min function ou restrições laterais do problema de alocação

Essa função define os limites de alocação, portanto o ângulo máximo e mínimo de trabalho dos propulsores azimutais, e os limites de empuxo máximo e mínimo. Como discutido nos fundamentos teóricos, definem-se as restrições de caixa conforme o intervalo de trabalho dos atuadores.

O código básico dessa função é apresentado nas Equações 2.20 e 2.21. Note que os valores máximos de empuxo representam 80% de seu valor máximo real, isso deve ser feito por medidas de segurança (MOBERG; HELLSTROM, 1983).

$$T_{\min} \leq T_i \leq 0,8 \cdot T_{\text{Max}} \quad (2.20)$$

$$0 \leq \alpha_i \leq 2\pi \quad (2.21)$$

2.2.2.6 – Objective Function

A Função Objetivo é aquela que deve ser minimizada pelo algoritmo, portanto deve-se tornar mínima a potência consumida pelo navio durante sua operação DP. Segundo Tannuri (2002), a relação entre potência e empuxo esta expressa na Equação 2.22, onde c_i é um valor constante.

$$P(\text{power}) \propto (T_i)^{\frac{3}{2}} \Rightarrow P(\text{power}) = c_i \cdot (T_i)^{\frac{3}{2}} \quad (2.22)$$

A Equação 2.22 expressa a potência de um único propulsor. O objetivo da otimização é minimizar a potência geral consumida pelo conjunto de atuadores. Caso os propulsores

sejam diferentes, é necessário construir uma matriz de peso, que atribua as constantes de potencia aos seus respectivos propulsores.

$$Obj(c_i, T_i) = \sum_{i=1}^n \left[c_i \cdot (T_i)^{\frac{3}{2}} \right] = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 \\ 0 & 0 & c_3 & 0 \\ 0 & 0 & 0 & c_4 \end{bmatrix} \begin{bmatrix} (T_1)^{\frac{3}{2}} \\ (T_2)^{\frac{3}{2}} \\ (T_3)^{\frac{3}{2}} \\ (T_4)^{\frac{3}{2}} \end{bmatrix} \quad (2.23)$$

2.2.2.7 – $\eta(\alpha)$

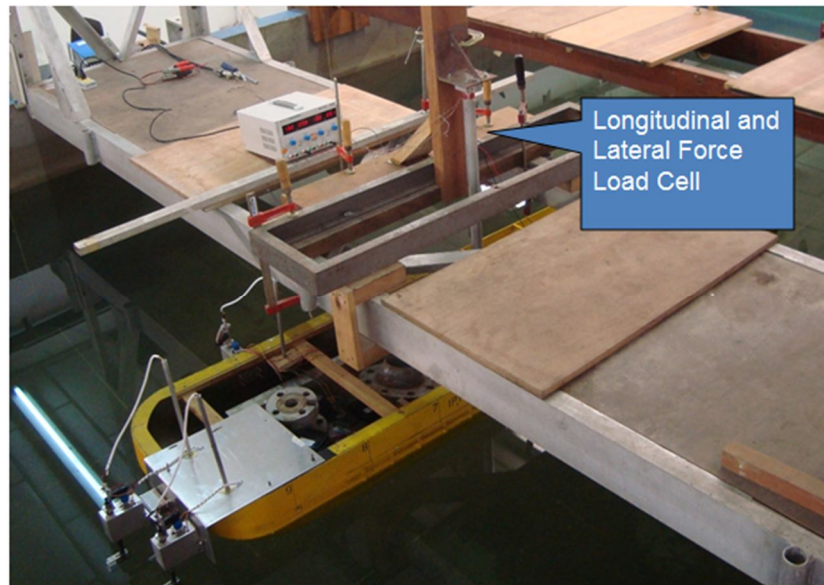
η é a função de interferência ilustrada na Figura 2.2. Embora fosse mais correto definir $\eta(\alpha, n)$ onde n é a rotação do propulsor, pois muitos dos efeitos de interferência poderiam depender da velocidade do jato d'água (como explicado no item 1.3), utiliza-se apenas a dependência com ângulo de propulsão α . Isso se deve ao fato da eficiência ter variado muito pouco com diferentes rotações nos testes práticos (ARDITTI; TANNURI, 2011).

Outra razão é que os fatores de eficiência (η) servem para comparar os diferentes ângulos de propulsão, necessários à Alocação de Empuxo, portanto obter-se-á uma alocação otimizada mesmo que a rotação do propulsor caia (todos os fatores de eficiência diminuirão de maneira proporcional a n^R , em que R estabelece uma relação entre a rotação do motor e a velocidade do jato d'água). O equilíbrio do navio será garantido pela robustez do sistema DP, que conta com realimentação em seu controle de alocação. Dessa maneira, a comparação relativa entre a eficiência dos diferentes ângulos de alocação de cada propulsor é mais importante que seus valores globais.

Além disso, caso fosse definida a função $\eta(\alpha, n)$, o algoritmo se tornaria muito complexo (tendo em vista que o empuxo também depende da rotação n) e não haveria como solucionar o problema de alocação.

Os dados de interferência podem ser encontrados na bibliografia, ou devem-se realizar testes com o modelo de embarcações, como ilustrado no arranjo experimental da Fig. 2.6.

Figura 2. 6 – Arranjo experimental para obtenção da curva de eficiência dos propulsores.



As Equações 2.24 e 2.26 ilustram como obter a eficiência em ensaios experimentais (ARDITTI; TANNURI, 2012):

$$K_{t0} = \frac{T_0}{\rho n^2 D^4} \quad (2.24)$$

T é o empuxo; ρ é a densidade da água; D é o diâmetro de propulsor;

K_{t0} pode ser determinado a partir de testes com o propulsor desacoplado ao navio.

$$K_t = \frac{T}{\rho n^2 D^4} \quad (2.25)$$

K_t é determinado através de testes do propulsor acoplado ao navio.

Finalmente:

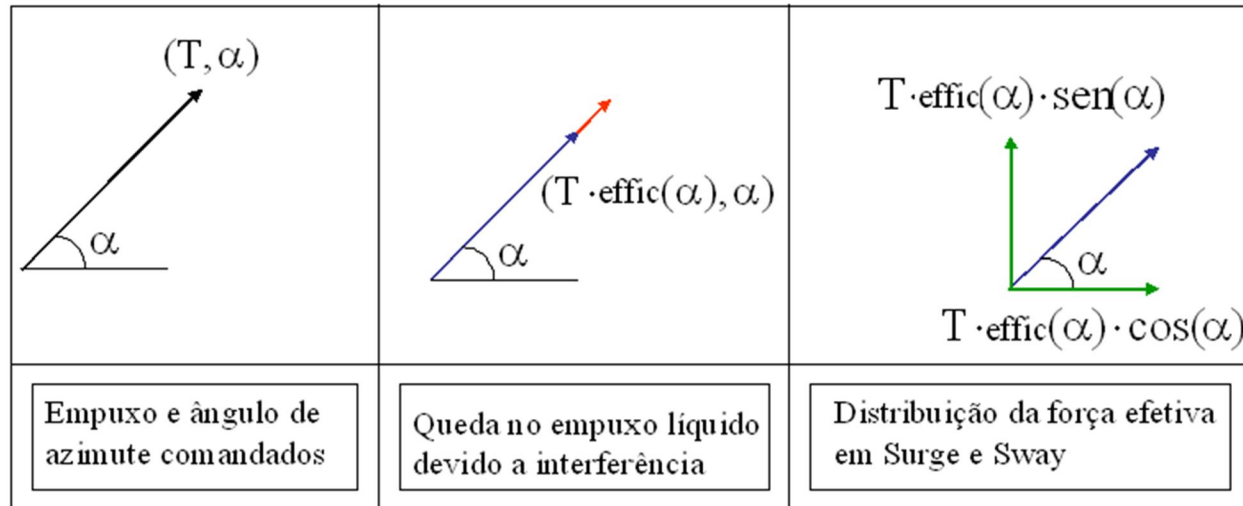
$$\eta = \frac{K_t}{K_{t0}} \quad (2.26)$$

2.2.2.8 – *noncoln function*

Essa função garante que o navio suportará as cargas ambientais. A função não linear de restrições (*noncoln*, Manual do MATLAB 2008), contém um sistema com as três equações da força requerida pela embarcação (direções *Surge* e *Sway*; e momento de *Yaw*), considerando o problema de interferência.

Neste caso trata-se da força efetiva exercida pelos propulsores, ou seja, considera-se a perda de eficiência relacionada aos fenômenos de interferência. A Figura 2.7 ilustra como é a distribuição de empuxo efetiva para um caso geral (note que a função *effic* é a eficiência do propulsor em função de sua posição angular):

Figura 2. 7 – Cálculo da Força efetiva considerando o fenômeno de interferência.



Determinadas as forças efetivas pode-se montar o sistema de Equações 2.27 que representa a geração de forças da embarcação. Portanto a solução desse sistema com o menor valor da função objetivo, respeitando-se as limitações dos propulsores, garantirá o equilíbrio da embarcação com o menor consumo de energia.

$$R: \begin{cases} \sum_{i=1}^N F_{x_i} \cdot \eta_i - F_x = 0 \\ \sum_{i=1}^N F_{y_i} \cdot \eta_i - F_y = 0 \\ \sum_{i=1}^N F_{y_i} \cdot \eta_i \cdot (x_i) - \sum_{i=1}^N F_{x_i} \cdot \eta_i \cdot (y_i) - M_z = 0 \end{cases} \quad (2.27)$$

2.2.2.9 – FMINCON function (Manual do MATLAB 2008)

De acordo com a divisão de métodos de solução de problemas de otimização, listados na fundamentação teórica, deve-se optar por um método de solução não analítico devido à complexidade do problema, ilustrado pela curva de eficiência do propulsor (Tabela 2.1) combinado a não linearidade da função objetivo.

Note que o método gráfico embora eficiente seja ineficaz neste projeto, devido ao elevado número de variáveis de projeto (mínimo 4 como discutida na seção 1.5).

Métodos randômicos também não são cabíveis a este projeto, pois o número elevado de variáveis contínuas dificulta o acerto de uma solução ótima, e o tempo de processamento seria muito elevado.

Dessa forma opta-se por trabalhar-se com o método numérico de problemas com restrições, que consegue lidar bem com a complexidade das restrições de igualdade (*noncoln*) e da função objetivo.

O algoritmo de alocação é executado pela função FMINCON do programa MATLAB. Essa função resolve o sistema não linear de restrições (*noncoln*), através da estimativa inicial (T_{polar}), das limitações dos propulsores (*max/min function*), buscando o menor valor da função objetivo ($Obj(T)$).

2.2.3 – Resultados da Implementação inicial do Algoritmo

2.2.3.1 – Algoritmos de Alocação de Empuxo simulados

A implementação inicial do Algoritmo de Alocação de Empuxo foi testado e comparado com dois algoritmos de Alocação de Empuxo padrão (ARDITTI; TANNURI, 2011, 2012):

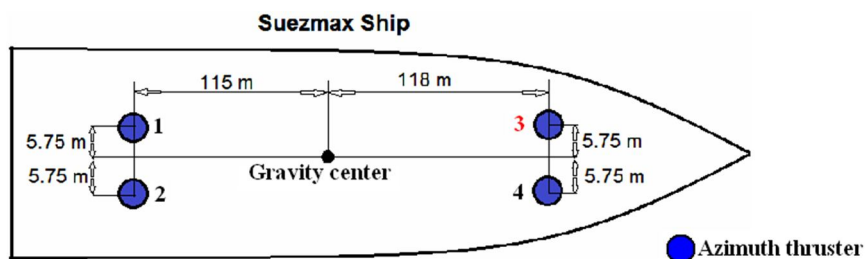
1. Simple Lagrangian Allocation (S.L.A.): Simple Alocação de Empuxo, utilizando o método da matriz pseudo-inversa (normalmente utilizado em simulações da embarcação).
2. Algoritmo de Zona Proibida: Algoritmo de Alocação de Empuxo que não permite posicionar os propulsores em zonas onde o jato d'água de um propulsor acerte outro propulsor. Ou seja, define-se uma zona de $\pm 20^\circ$ em torno da direção que liga dois propulsores relativamente próximos. Além disso, há um algoritmo recursivo interno que garante que a Alocação de Empuxo respeite as forças de controle. Esse algoritmo recursivo interno emula um sistema de controle, que considera os fenômenos de interferência (baseado nas curvas de eficiência da embarcação), isso é necessário pois a concepção deste algoritmo considera apenas uma simplificação do problema de interferência entre propulsores, deixando de lado, todas as outras fontes de perda de eficiência.

2.2.3.2 – Embarcações Testadas

Utilizaram-se duas embarcações para compara os algoritmos de Alocação de Empuxo:

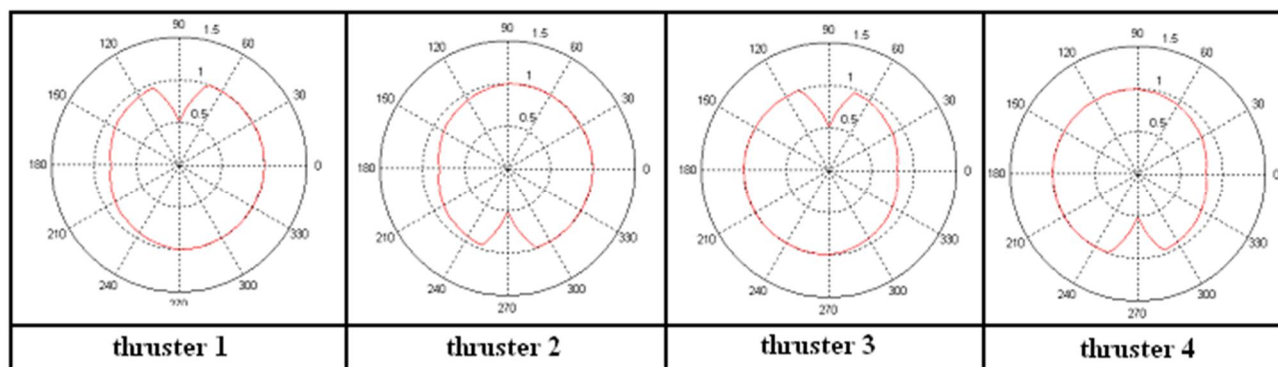
O Navio 1 (Fig. 2.5) é uma FPSO - Unidade flutuante de armazenamento e transferência (em inglês *Floating Production Storage and Offloading*). Esse navio não tem modelo, portanto suas curvas de eficiência (Fig. 2.6) foram construídas a partir de dados bibliográficos (ARDITTI; TANNURI, 2011).

Figura 2. 8 – Desenho esquemático do Navio 1.



Fonte: Arditti e Tannuri (2011).

Figura 2. 9 – Curvas de eficiência do Navio 1.



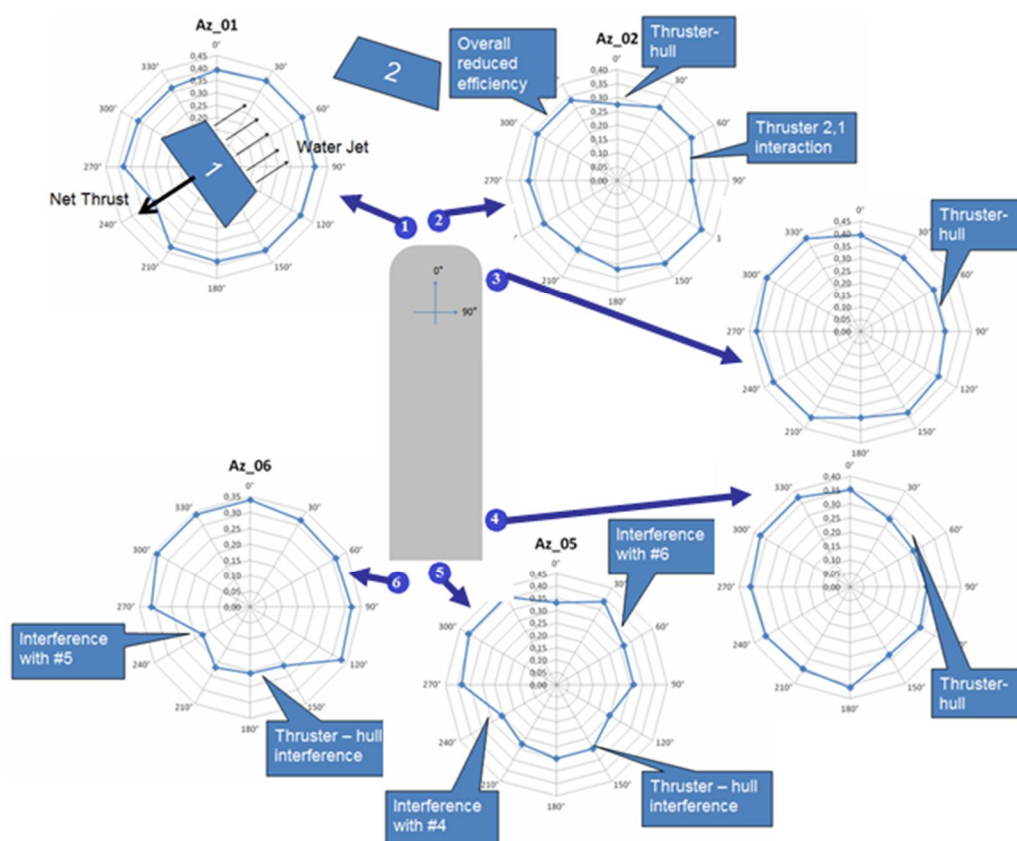
Fonte: Arditti e Tannuri (2011).

A Embarcação 2 é utilizada como navio guindaste. Uma foto do modelo dessa embarcação esta ilustrada na Fig. 2.7. As curvas de eficiência de seus propulsores foram determinadas a partir de simulações com o modelo no Tanque de Provas da escola Politécnica da USP (Arditti and Tannuri 2012) e são apresentadas na Fig. 2.8.

Figura 2. 10 – Modelo da Embarcação 2.



Figura 2. 11 – Curvas de eficiência dos propulsores da Embarcação 2.



Fonte: Arditti e Tannuri (2012).

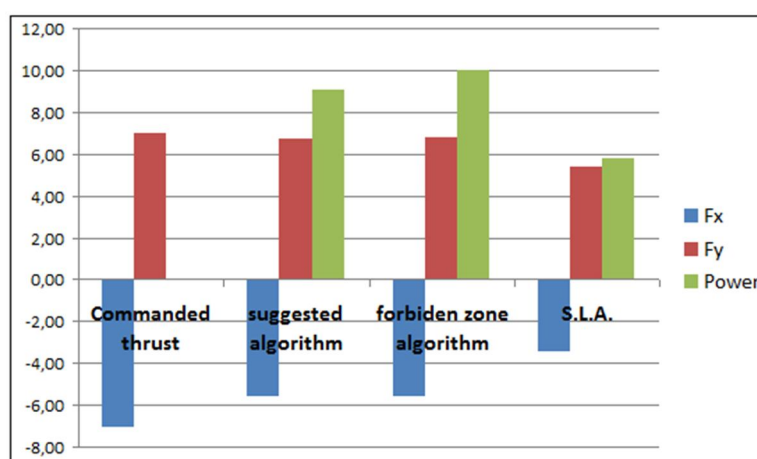
2.2.3.3 – Resultados das Simulações

Constatou-se que a implementação inicial do Algoritmo de Alocação de Empuxo considerando interferência realiza uma alocação objetiva e consegue lidar com qualquer forma de interação.

As simulações computacionais ilustram significativa economia no consumo de combustível do Navio 1, podendo-se atingir 129000 litros de economia de diesel por ano para uma frota de 10 embarcações, ao comparar-se o S.L.A. e o Algoritmo de Zonas Proibidas. Além disso, a janela de operação, que representa a condição extrema em que a embarcação pode operar, aumentou em 5% (ARDITTI; TANNURI, 2011).

O ensaios experimentais da Embarcação 2 ilustraram a necessidade de considerar os fenômenos de interferência na alocação. Observe que na Figura 2.9, os dois algoritmos que levam em conta as interações chegaram muito mais próximos de atingir a força de controle, ou seja, ao incluir os dados de interferência na alocação o sistema torna-se significativamente mais preciso e seguro.

Figura 2. 12 – Ensaio experimental da Embarcação 2.



Fonte: Arditti e Tannuri (2012).

Embora os dois algoritmos (proposto e de zonas proibidas) sejam igualmente precisos nota-se claramente que o algoritmo desenvolvido é mais eficiente. Na simulação da Fig. 2.9 observou-se economia de 10,3 %. Na média dos ensaios a economia ficou em 5%.

2.2.4 – Análise da Implementação Inicial do Algoritmo proposto

O algoritmo comportou-se da maneira esperada, sendo sempre superior aos Algoritmos de Alocação de Empuxo utilizados atualmente. Nota-se que o sistema fica mais preciso ao considerar o sistema com interferência, e conseqüentemente mais seguro, pois a força comandada é alocada mais rapidamente, prevenindo comportamentos inesperados da embarcação.

Além disso, o ganho econômico fica evidente quando se compara o consumo de combustível entre os algoritmos. Não obstante, esse ganho econômico é mais pronunciado se for considerado o custo de oportunidade referente ao aumento da janela de operação da embarcação, o que implica na possibilidade de operar por mais tempo.

Embora os resultados tenham sido excelentes uma análise objetiva do ponto de vista de otimização revela que o algoritmo proposto não é convexo, ou seja, não há como garantir que os resultados atingidos sejam os melhores possíveis. Isso se explica no fato de que num problema não convexo não há como garantir que um ponto de mínimo local seja o ponto de mínimo global.

A seguir prova-se que a modelagem atual do problema torna-o não convexo. As condições de convexidade são:

1. Matriz Hessiana positiva definida ou positiva semi-definida. Essa condição é respeitada, pois a Matriz Hessiana é positiva semi-definida apresentando o aspecto ilustrado na Eq. 2.28 (repetindo-se o padrão para a diagonal principal e o resto da Matriz é preenchido com 0, para mais de um propulsor):

$$\text{Hess} = \begin{bmatrix} \frac{\partial^2 F}{\partial T^2} & \frac{\partial^2 F}{\partial T \partial \alpha} \\ \frac{\partial^2 F}{\partial T \partial \alpha} & \frac{\partial^2 F}{\partial \alpha^2} \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 4 \cdot \sqrt{T} & 0 \\ 0 & 0 \end{bmatrix} \quad (2.28)$$

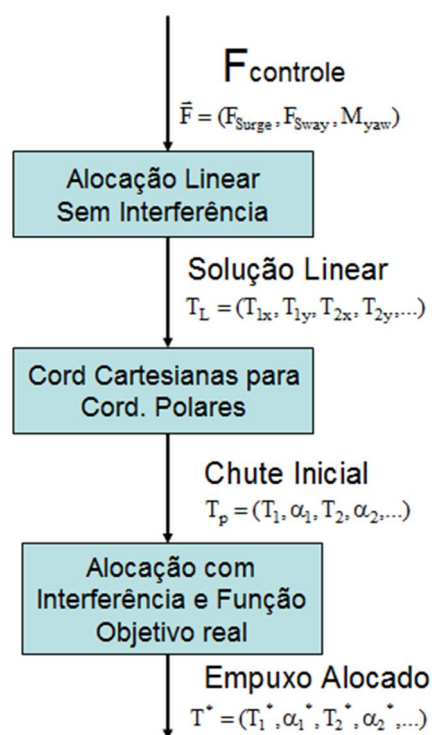
2. Espaço viável convexo. Essa condição não é respeitada. Embora as equações de desigualdade (*max/min function*) sejam todas lineares (portanto côncavas), as restrições de desigualdade (noncoln) não são lineares.

3 – Estudo da Dinâmica Interna do Algoritmo de Otimização

O estudo da dinâmica Interna da implementação inicial do Algoritmo de Alocação de Empuxo tem por objetivo: Retratar seu funcionamento para que seja possível melhorá-lo e torna-lo aplicável em um pacote computacional, ou seja, em um produto que não pode depender do *software* MATLAB, que é pago.

A implementação inicial do Algoritmo de Alocação de Empuxo tem sua estrutura retratada na Fig. 2.5. Inicialmente será apresentada a dinâmica do Algoritmo de Alocação de Empuxo (Fig. 3.1) de maneira mais simplificada e cada uma das funções do algoritmo será detalhada quanto ao seu funcionamento intrínseco.

Figura 3. 1 – Dinâmica da implementação inicial do Algoritmo de Alocação de Empuxo.

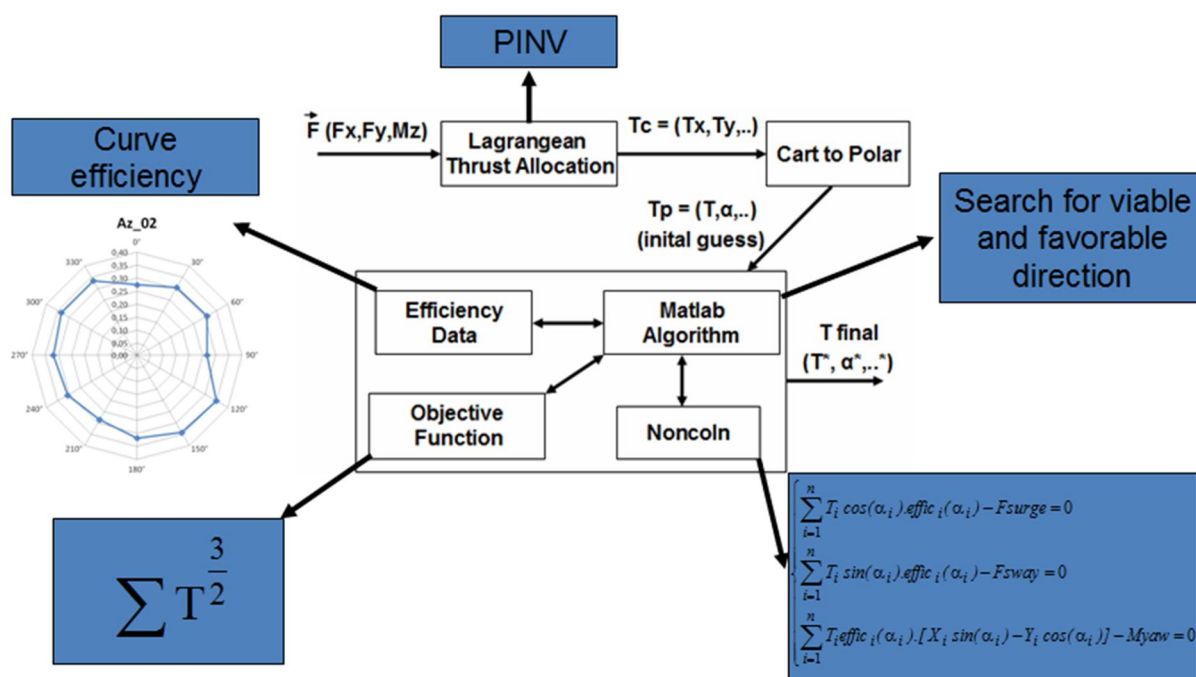


Visto que o Algoritmo de Alocação de Empuxo desenvolvido pode ser dividido em 3 blocos principais descreve-se a função de cada um destes blocos.

1. Alocação Linear sem Interferência: Simples Alocação de Empuxo para gerar as forças de controle, considerando-se o sistema sem Interferência e com Função Objetivo Quadrática.
2. Cord. Cartesianas para Cord. Polares: Simples passagem da solução linear para coordenadas polares, para gerar o chute inicial de alocação, uma vez que a mesma é realizada em coordenadas polares.
3. Alocação com Interferência e Função Objetivo real: Sistema de otimização que recebe o chute inicial de alocação em coordenadas polares, e gera as forças de controle considerando os fenômenos de Interferência e a Função Objetivo real.

A Fig. 3.2 apresenta os 3 blocos do Algoritmo de Alocação de Empuxo fornecendo maiores detalhes quanto à maneira como cada um é processado. Note que o bloco inicial é resolvido utilizando-se pseudo-inversa de A. O segundo bloco tem uma solução bastante simples. Por fim o terceiro bloco representa o sistema mais complexo, que é resolvido (no estágio inicial de desenvolvimento) pelo programa MATLAB e a função FMINCON.

Figura 3. 2 – Dinâmica de otimização do algoritmo de otimização detalhada.



Fonte: Arditti e Tannuri (2012).

A seguir analisa-se a escolha do chute inicial do Algoritmo de Alocação de Empuxo e da dinâmica interna do algoritmo de otimização do MATLAB. A passagem de coordenadas cartesianas para polares é simples e foi descrita no capítulo anterior.

3.1 – Análise do Chute Inicial do Problema de Alocação de Empuxo

O chute inicial do Algoritmo de Alocação de Empuxo é realizado pela solução do sistema sem interferência considerando-se uma função objetivo quadrática. O anexo A ilustra que essa solução é obtida através da Matriz pseudo-inversa da Matriz A.

O chute inicial não considera os fenômenos de interferência pois se trata simplesmente da primeira iteração do algoritmo, que visa apenas encontrar um ponto próximo à solução final do problema. A consideração dos fenômenos de interferência tornaria essa etapa lenta sem qualquer necessidade.

Opta-se por resolver o problema considerando-se a potência quadrática (T^2), ao invés da potência real ($T^{\frac{3}{2}}$) por duas razões:

1. As soluções do sistema sem interferência para a potência quadrática e para potência real são muito próximas.
2. Resolver o problema considerando-se a potência real é mais complexa e requer maior tempo de processamento.

Destaca-se que não é escolhido um chute inicial aleatório, pois os algoritmos de otimização dependem bastante da primeira iteração para resolver o problema, e uma escolha aleatório poderia posicionar o sistema muito distante da solução ótima, o que poderia incorrer em elevado tempo de processamento até chegar-se a solução ótima, ou mesmo, levar a última iteração para um mínimo local distante do mínimo global.

Além disso, alguns dos algoritmos desenvolvidos no próximo capítulo requerem uma solução inicial válida, ou seja, que resolva o sistema de forças da embarcação (F_{surge} , F_{sway} e M_{yaw}). Para obtê-la basta dividir o empuxo alocado sem interferência pela eficiência correspondente ao ângulo de azimuth do mesmo, conforme a Eq. 3.1.

$$T_{\text{feasible}} = \frac{T_0}{\text{effic}(\alpha_0)} \quad (3.1)$$

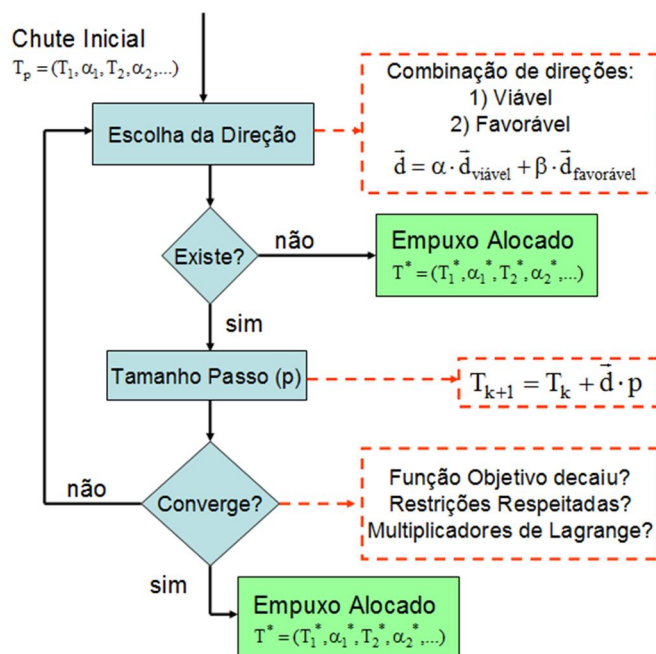
3.2 – Análise da Dinâmica Interna do Algoritmo de Otimização do MATLAB

O algoritmo de otimização utilizado pelo MATLAB é o FMINCON. O mesmo é capaz de resolver problemas de otimização complexos (função objetivo e restrições não lineares), através de um método numérico de 1ª ordem.

A otimização ocorre em várias iterações conforme ilustrado na Figura 3.2. O algoritmo do MATLAB a cada iteração tenta diminuir o valor da função objetivo, respeitando as forças de controle (equilíbrio das forças ambientais), considerando as curvas de eficiência de cada propulsor.

A fim de minimizar a função objetivo o algoritmo de otimização do MATLAB (Fig. 3.3) utiliza um método Trust Region (seção 1.6.10) no qual busca a direção ideal para a presente iteração a partir do método de direções viáveis e favoráveis. Calcula-se também o tamanho ideal do passo para a presente iteração. E ao final de cada iteração avalia-se a convergência do problema conforma as condições KKT (1.6.7).

Figura 3. 3 – Diagrama de blocos do FMINCON (otimizador do MATLAB).

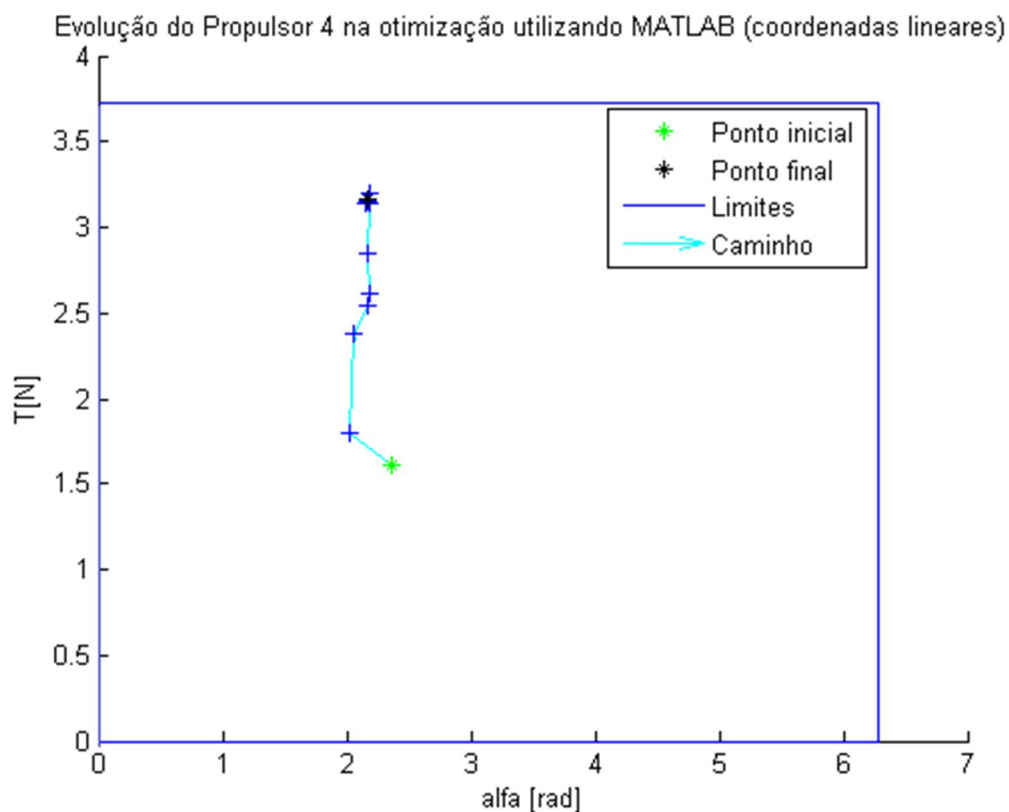


O método das direções viáveis e favoráveis consiste em combinar linearmente as mesmas para obter um direção ótima. Abaixo se define o conceito de direção viável e direção favorável.

1. Direções viáveis são mudanças nas variáveis de projeto que permitam que o algoritmo continue respeitando as restrições (Noncoln e desigualdades), ou seja, como as variáveis podem ser manipuladas no espaço viável.
2. Direções favoráveis são as manipulações que podemos realizar nas variáveis de projeto, de maneira que o valor da função Objetivo decaia.

A Figura 3.4 ilustra a evolução de um propulsor em uma Alocação de Empuxo utilizando-se o algoritmo FMINCON. O algoritmo escolhe apenas direções e posições viáveis, tanto que o mesmo mantém as variáveis dentro de seu respectivo espaço viável.

Figura 3. 4 – Evolução do propulsor 4 no processo de otimização dentro do espaço viável.



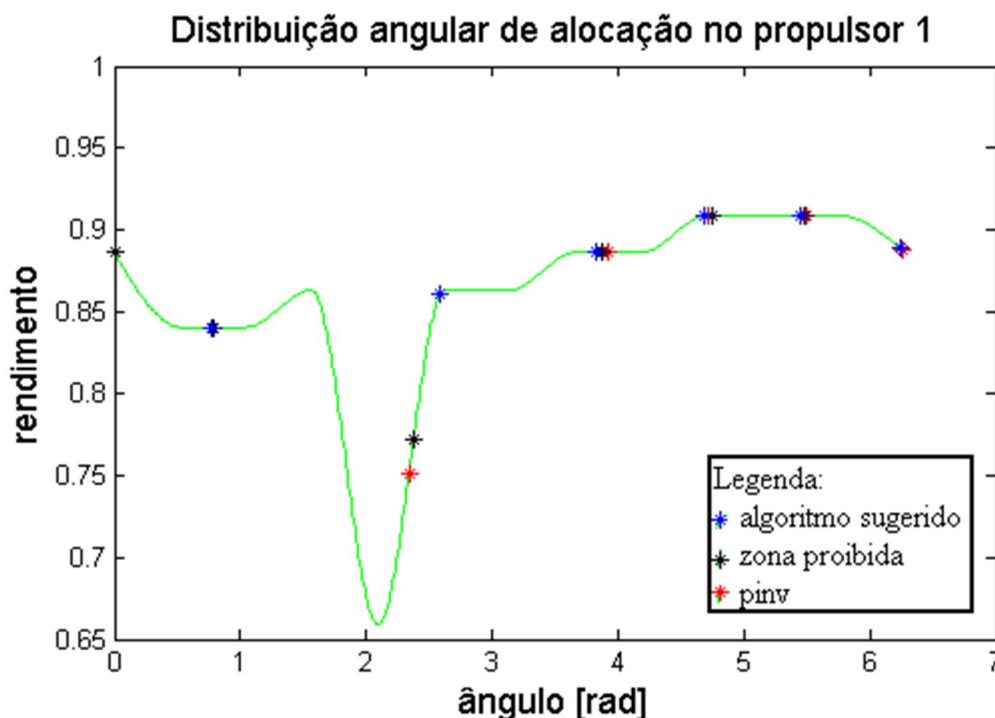
Ao obter a direção ótima o algoritmo dá um passo (variação das variáveis de projeto), e verifica se a função objetivo diminuiu e se as restrições seguem respeitadas. Caso o valor da função objetivo tenha diminuído e ainda não tenham sido respeitadas as condições KKT, o algoritmo repete o processo até que não haja mais direções viáveis e favoráveis ou até que o decrescimento da função objetivo se torne desprezível.

Note que ao encontrar um ponto onde não há direções favoráveis e viáveis o algoritmo encontra um ponto estacionário. Se este ponto for um mínimo local não há como provar que se trata do mínimo global, pois o problema não é convexo (restrições de igualdade considerando-se a curva de eficiência não são lineares).

Uma análise qualitativa revela que a direção favorável se refere ao decrescimento da função objetivo, e está intrinsecamente relacionada ao crescimento da função de eficiência dos propulsores. Por outro lado, a direção favorável refere-se a não distanciar-se muito dos ângulos viáveis à geração de empuxo na direção da força comandada pelo sistema de controle (ARDITTI; TANNURI, 2011).

Essa argumentação pode ser exemplificada ao analisar-se a Fig. 3.5, que ilustra o posicionamento do propulsor 1 da Embarcação 2, realizado por diferentes algoritmos de Alocação de Empuxo, sobre a curva de eficiência do atuador.

Figura 3. 5 – Posições angulares dos algoritmos de Alocação de Empuxo para o propulsor 1 em diferentes alocações de empuxo, e sua curva de eficiência.



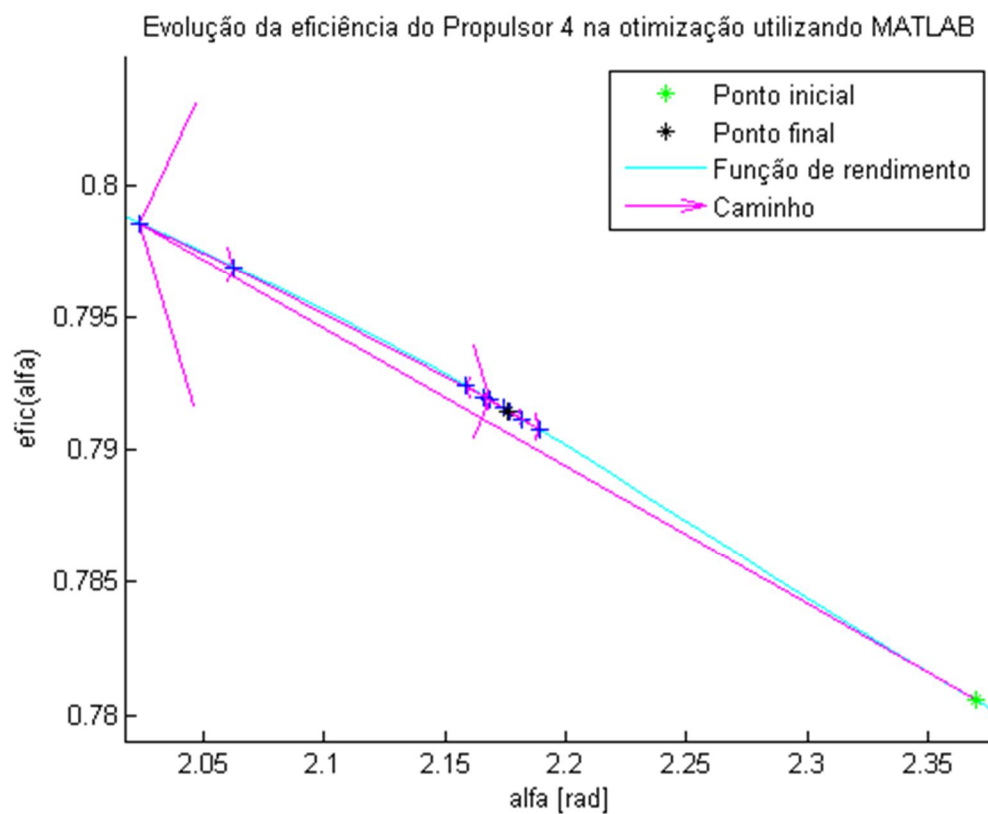
Fonte: Simulação realizada em Arditti e Tannuri (2012).

Fica claro que ao algoritmo sugerido sempre posiciona seu atuador em ângulos favoráveis, onde o rendimento é superior aos outros algoritmos. Além disso, também é possível observar que os ângulos escolhidos pelo algoritmo são viáveis, pois estão próximos dos ângulos necessários para gerar as forças de controle.

Se o algoritmo optasse apenas por ângulos favoráveis, o mesmo iria sempre para o ponto de máximo da sua curva de eficiência, o que não seria viável, uma vez que esses ângulos podem ser muito distantes da direção da força que se deseja gerar.

Isso ocorre, pois se trabalha com um sistema em coordenadas polares em que a força efetiva é definida pelo rendimento do propulsor e pela direção do mesmo. Portanto deve-se contemplar na otimização tanto a curva de eficiência dos propulsores, como as funções senoidais que determinam a direção do empuxo. Isso está ilustrado na Fig. 3.6 em que se verifica que o FMINCON posiciona o propulsor em um ângulo de azimuth que contempla a direção da força a ser desenvolvida (próxima à posição do ângulo inicial), e busca uma direção favorável (crescimento do rendimento).

Figura 3. 6 – Evolução do propulsor 4 no processo de otimização contemplando a direção da força e o aumento de sua eficiência.



Ao compreender a dinâmica do FMINCON é possível compreender como se comporta (“pensa”) o algoritmo e desenvolver técnicas de otimização simplificadoras, que permitam o desenvolvimento de um algoritmo de otimização próprio.

3.3 – Análise do comportamento do FMINCON

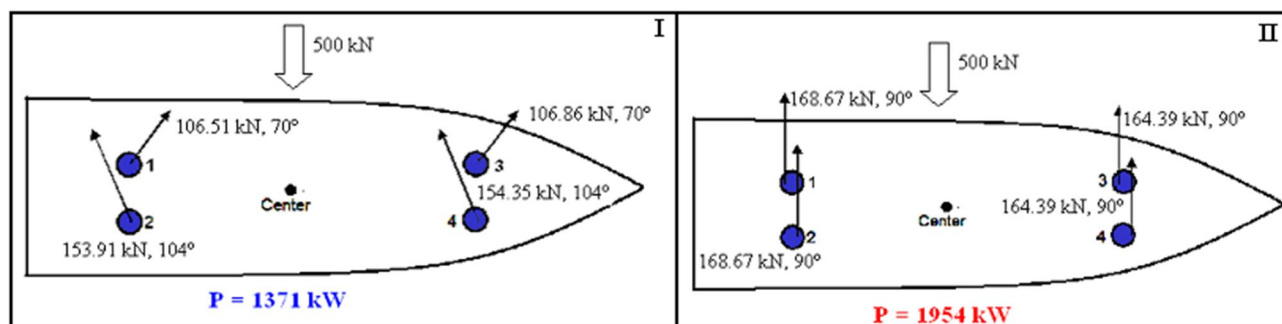
Essa seção tem por objetivo retratar como é a tomada de decisão do FMINCON do ponto de vista comportamental. Pretende-se revelar como a dinâmica interna do algoritmo atua sobre o comportamento qualitativo da Alocação de Empuxo, com esse objetivo dois estudos de caso são apresentados.

3.3.1 – Análise do comportamento em relação interferência entre propulsores

A Fig. 3.7 ilustra a Alocação de Empuxo de 500 kN na direção de Sway para o Navio 1. O algoritmo que considera os fenômenos de interferência sai da zona de interação entre os propulsores (1 e 2; 3 e 4) sem que seja definida uma zona proibida. Por outro lado o algoritmo que não considera os fenômenos de interferência lança jatos de água de um propulsor sobre outro, o que diminui muito sua eficiência, e leva a um gasto energético 43% maior.

Figura 3. 7 – Esquerda alocação do algoritmo de otimização do MATLAB considerando interferências.

Direita alocação S.L.A.



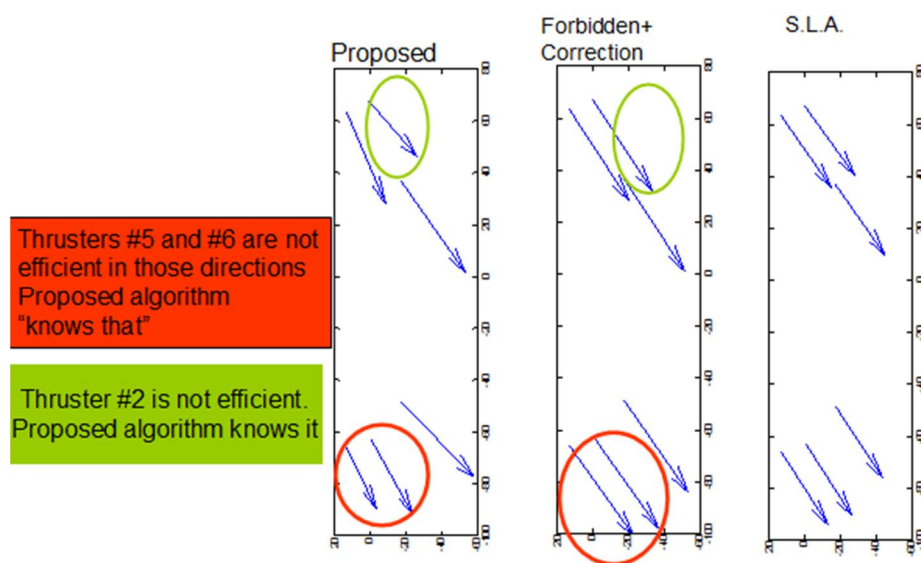
Fonte: Arditti e Tannuri (2011).

O algoritmo de alocação de empuxo saiu da zona de interação entre propulsores por tratar-se de uma zona com valores baixos na curva de eficiência. Uma análise do ponto de vista qualitativo revela que a zona de interferência entre propulsores é viável, mas não favorável. Dessa forma, constatou-se que o algoritmo consegue lidar bem com a zona proibida sem a necessidade de se restringir a sua liberdade de posicionamento dos propulsores (como é feito no algoritmo de alocação com zonas proibidas).

3.3.2 – Análise do comportamento para regiões com baixa eficiência

A Fig. 3.8 ilustra a alocação de empuxo do algoritmo FMINCON (considerando interferências), Zona Proibida e S.L.A. para uma força de controle de ($F_{\text{surge}} = -7\text{N}$, $F_{\text{sway}} = 7\text{N}$, $M = 0$). A Alocação de Empuxo do FMINCON economiza 10,3% em relação ao Algoritmo de Zona Proibida, enquanto que o S.L.A. não gera a força de controle (mesma alocação da Fig. 2.9).

Figura 3. 8 – Alocação de uma força de controle de -7N em Surge e 7N em Sway com momento nulo, para 3 algoritmos de alocação.



Fonte: Arditti e Tannuri (2012).

A implementação inicial do Algoritmo de Alocação de Empuxo requisitou menos força dos propulsores 2, 5 e 6, pois os mesmos apresentavam baixa eficiência. Do ponto de vista qualitativo pode-se inferir que não havia como mudar a posição desses propulsores o que implicaria em desviá-los significativamente da linha de ação da força de controle, ou seja, esse era o espaço viável. Como não havia como melhorar o rendimento desses propulsores a solução intuitiva e correta é exigir menos empuxo dos mesmos e o algoritmo proposto toma essa atitude enquanto o algoritmo de zona proibida não o faz.

3.4 – Relação do Comportamento do Algoritmo com suas vantagens

O princípio do comportamento do algoritmo FMINCON é:

1. Evitar zonas de interação entre propulsores (não favorável).
2. Requisitar menor empuxo de propulsores com baixa eficiência.

Esse princípio garante os resultados ilustrados na seção 2.4, ou seja, maior economia de combustível e maior precisão durante a operação DP.

4 –Algoritmo de Alocação de Empuxo

Neste capítulo apresentam-se novas implementações para o algoritmo de otimização do MATLAB, que permitem o desenvolvimento do pacote computacional para simulação de operação DP. Inicialmente analisa-se do ponto de vista de otimização o problema da Alocação de Empuxo, discutem-se técnicas simplificadoras, apresentam-se os novos algoritmos desenvolvidos, os critérios para a seleção dos mesmos, por fim as simulações que sustentam a seleção do novo algoritmo de alocação de empuxo.

Resumidamente o objetivo deste capítulo é desenvolver um novo algoritmo de alocação de empuxo, tendo como base os resultados, dinâmica interna e comportamento do FMINCON.

4.1 – Problema Original de Otimização da Alocação de Empuxo

O problema original de Alocação de Empuxo é modelado conforme a Eq. 4.1.

$$\begin{aligned}
 P &= \sum_{i=1}^N c_i \cdot T_i^{\frac{3}{2}} \\
 R: &\begin{cases} \sum_{i=1}^N T_i \cdot \cos(\alpha_i) \cdot \eta_i - F_{surge} = 0 \\ \sum_{i=1}^N T_i \cdot \sin(\alpha_i) \cdot \eta_i - F_{sway} = 0 \\ \sum_{i=1}^N T_i \cdot \sin(\alpha_i) \cdot \eta_i \cdot (x_i) - \sum_{i=1}^N T_i \cdot \cos(\alpha_i) \cdot \eta_i \cdot (y_i) - M_{yaw} = 0 \end{cases} \quad (4.1) \\
 I: &\{T_i \leq T_{max_i}
 \end{aligned}$$

Ao observar-se o problema destacam-se dois complicadores para o algoritmo de otimização que será desenvolvido.

1. A Função objetivo com $T^{\frac{3}{2}}$.
2. As restrições de igualdade com T multiplicando $\cos(\alpha) \cdot \eta$, que pode ser uma função qualquer, pois o rendimento é definido a partir de uma *spline* que interpola os pontos em que foi aferida a eficiência do propulsor.

Note que é muito mais difícil lidar com a segunda complicação.

4.2 – Novos Algoritmos de Alocação de Empuxo

Tendo em vista a complexidade do problema diferentes técnicas de otimização foram utilizadas para abordá-lo. Cinco algoritmos foram desenvolvidos e são apresentados a seguir.

4.2.1 – Método da Direção e do Rendimento

A ideia deste algoritmo é otimizar cada um dos propulsores separadamente para definir seu ângulo de azimuth, e depois resolver o problema original de otimização com os ângulos definidos. Esse algoritmo é puramente intuitivo e não tem nenhuma fundamentação matemática que garanta sua eficiência, o mesmo foi constituído baseado somente na análise comportamento do FMINCON para a Alocação de Empuxo.

Baseado na análise de posicionamento viável e favorável dos propulsores da embarcação trabalha-se com as seguintes hipóteses:

1. Um ângulo viável deve estar próximo à linha de ação de força, e é possível determiná-lo com o chute inicial da implementação inicial do Algoritmo de Alocação de Empuxo (solução com o método da Matriz Pseudo-inversa).
2. Um ângulo favorável é aquele em que a eficiência do propulsor é alta.

Unindo as duas hipóteses decidiu-se que a otimização de cada propulsor para definir seu ângulo de atuação deve ser realizada maximizando a Eq. 4.2 de cada propulsor. Note que α_0 é o ângulo da solução da Matriz pseudo-inversa.

$$\begin{aligned} \max_{\alpha_i} \text{effic}_i(\alpha_i) \cdot \cos(|\alpha_{0i} - \alpha_i|) \\ 0 \leq \alpha_i \leq 2\pi \end{aligned} \quad (4.2)$$

A solução deste problema embora analiticamente não seja tão simples, do ponto de vista numérico é resolvida variando α com elevada precisão e encontrando o máximo valor da função por comparações.

Definida a parte inicial do problema que é encontrar os ângulos de alocação, o problema de otimização passa a ser descrito pela Eq. 4.3. Lembre-se que o ângulo de azimuth não é mais uma variável no problema.

$$\begin{aligned}
 & \text{Min}_T \sum_{i=1}^n T_i^{\frac{3}{2}} \\
 & \text{considerando} \\
 & \begin{cases} \sum_{i=1}^n T_i \cdot \cos(\alpha_i) \cdot \text{effic}_i(\alpha_i) - F_{\text{surge}} = 0 \\ \sum_{i=1}^n T_i \cdot \sin(\alpha_i) \cdot \text{effic}_i(\alpha_i) - F_{\text{sway}} = 0 \\ \sum_{i=1}^n T_i \cdot \text{effic}_i(\alpha_i) \cdot [X_i \cdot \sin(\alpha_i) - Y_i \cdot \cos(\alpha_i)] - M_{\text{yaw}} = 0 \end{cases} \\
 & \text{ou} \\
 & \begin{cases} \sum_{i=1}^n T_i \cdot \text{cte}_{1i} - F_{\text{surge}} = 0 \\ \sum_{i=1}^n T_i \cdot \text{cte}_{2i} - F_{\text{sway}} = 0 \\ \sum_{i=1}^n T_i \cdot \text{cte}_{3i} - M_{\text{yaw}} = 0 \end{cases} \tag{4.3} \\
 & 0 \leq T \leq T_{\text{max}}
 \end{aligned}$$

A solução deste problema pode ser feita utilizando-se as condições KKT, e basta resolver um sistema que tem apenas uma não linearidade (\sqrt{T}). Esse sistema é resolvido numericamente (utilizando-se o método de Newton-Raphson) e obtém-se o empuxo requisitado por cada propulsor.

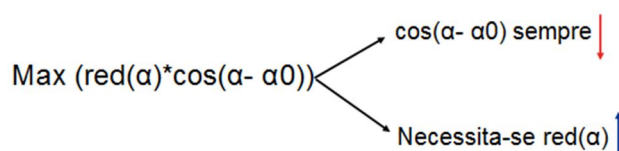
Note que o problema de otimização do empuxo de cada propulsor é convexo, pois tem a Matriz Hessiana definida Positiva e somente restrições lineares.

Esse algoritmo foi simulado e os resultados obtidos, do ponto de vista de economia de potência não foram bons. Observou-se que os ângulos de azimuth praticamente não mudavam, e não permitiam uma melhora na alocação.

Esse comportamento é explicado quando se observa a Eq. 4.2. Note que a função $\cos(|\alpha_{0i} - \alpha_i|)$ diminui significativamente conforme α_i se afasta de α_{0i} , portanto o

rendimento deve crescer consideravelmente para que o ângulo de azimuth mude, e essa não é uma característica da curva de eficiência. A Fig. 4.1 ilustra a situação.

Figura 4. 1 – Razão da Falha no Método da Direção e do Rendimento.



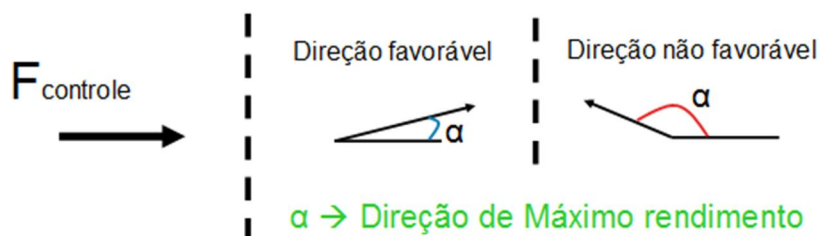
Portanto esse método já é descartado.

4.2.2 – Método Intuitivo

O método intuitivo é uma tentativa de melhorar o Método da Direção e do Rendimento. Como a função limitante do primeiro método é $\cos(|\alpha_{0i} - \alpha_i|)$, opta-se por realizar a otimização do propulsor (definição do seu ângulo de azimuth) baseado somente no seu rendimento. O segundo passo após a definição do ângulo dos propulsores mantém-se o mesmo.

Entretanto o ângulo de azimuth de máxima eficiência dos propulsores pode ser favorável ou desfavorável em relação à direção da força de controle, como ilustrado na Fig. 4.2.

Figura 4. 2 – Direção favorável ou desfavorável do ângulo de máxima eficiência.

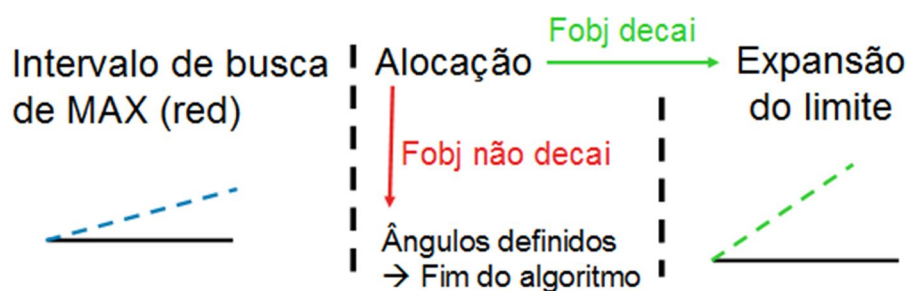


A solução deste problema é limitar o intervalo de busca da máxima eficiência para regiões próximas à direção inicial do propulsor (chute inicial da implementação inicial do Algoritmo de Alocação de Empuxo). Conforme o ângulo de máximo rendimento é encontrado

realiza-se a alocação (definição de forças) e verifica-se se a função objetivo melhorou em relação à última tentativa e toma-se a seguinte decisão (Fig.4.3):

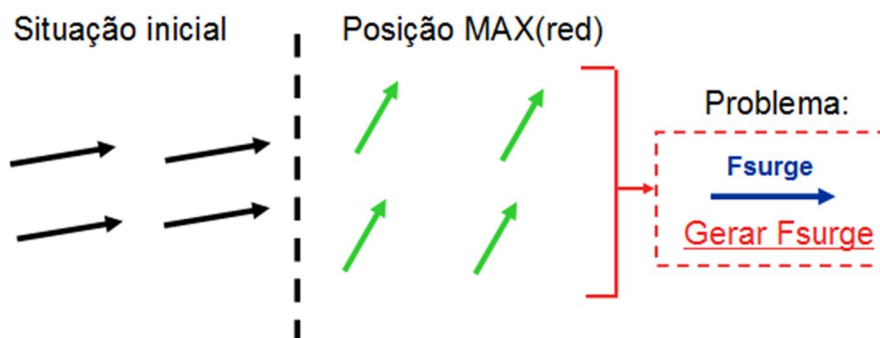
1. Caso a função objetivo esteja com um valor inferior o intervalo de busca é expandido.
2. Caso a função objetivo tenha crescido o algoritmo termina e fica com a penúltima solução (antes do crescimento da função).

Figura 4. 3 – Evolução do intervalo de busca do método intuitivo.



Entretanto é possível adiantar outro problema que pode ocorrer no método intuitivo, que é a rotação desfavorável dos propulsores (ilustrado na Fig. 4.4). Esse problema seria causado se todos ou a maioria dos propulsores girasse apenas em um sentido, o que dificultaria a geração de forças na direção de Surge ou Sway.

Figura 4. 4 – Rotação desfavorável dos propulsores no método intuitivo.



A solução para o problema de rotação desfavorável é definir que para cada propulsor que gira num sentido outro girará no sentido contrário, conforme ilustrado na Fig. 4.5.

Figura 4. 5 – Solução para o problema de rotação desfavorável dos propulsores no método intuitivo.



Porém a programação deste método é muito confusa e não há qualquer garantia matemática de que há alguma otimização ocorrendo. Embora intuitivamente esse método seja coerente, ele não é confiável e apresenta um problema intrínseco igual ao método da Direção e do Rendimento:

A otimização dos propulsores separadamente é ineficaz, pois não considera a otimização do sistema global que apresentar pequenas pioras em alguns propulsores e melhoras significativas em outros. Além disso, a separação dos propulsores não permite que o algoritmo de otimização posicione em conjunto todos os propulsores para gerar as forças de controle, que obviamente é um requisito para uma boa otimização.

Tendo em vista a ineficácia dos métodos intuitivos os novos algoritmos de otimização desenvolvidos têm forte fundamentação matemática.

4.2.3 – SQP - Potência Quadrática

Esse algoritmo de otimização simplifica o problema de duas maneiras:

1. A função objetivo considerada é $\sum T^2$, que é semelhante à função objetivo real e de solução mais simples.
2. As restrições de igualdade são linearizadas utilizando-se o método de Taylor (Eq. 4.4).

$$g_{\text{linear}} = g(x_0) + \nabla(x_0) \times [x - x_0] \quad (4.4)$$

Que quando aplicado à restrição gera a Eq. 4.5:

$$\sum_{i=1}^n \left[(T_i - T_{ki}) \cdot \cos(\alpha_{ki}) \cdot \eta(\alpha_{ki}) + T_{ki} \cdot (\alpha_i - \alpha_{ki}) \cdot \frac{\partial(\cos(\alpha_{ki}) \cdot \eta_i(\alpha_{ki}))}{\partial \alpha} \right] + g(T_k, \alpha_k) = 0 \quad (4.5)$$

A segunda derivada da Eq. 4.5 é realizada numericamente, conforma a Eq. 4.6.

$$\frac{\partial(\cos(\alpha_{ki}) \cdot \eta_i(\alpha_{ki}))}{\partial \alpha} = \frac{\cos(\alpha_{ki} + \Delta) \cdot \eta_i(\alpha_{ki} + \Delta) - \cos(\alpha_{ki} - \Delta) \cdot \eta_i(\alpha_{ki} - \Delta)}{2\Delta} \quad (4.6)$$

Existe a necessidade de aplicar limites móveis (inequações para limitar a mudança de valor nas variáveis), ou seja, limites para as variáveis em torno da iteração anterior, pois a matriz Hessiana é positiva semi-Definida (NOCEDAL; WRIGHT; 2006). Do ponto de vista intuitivo é necessário aplicar os limites móveis para a variação dos ângulos α , que fazem parte das restrições de igualdade, mas não da função objetivo, não mudarem o valor para uma região em que a linearização não é válida.

Com as simplificações o problema torna-se de solução significativamente mais simples, função objetivo quadrática e somente restrições (de igualdade e desigualdade lineares), ou seja, um problema de programação quadrática sequencial convexo.

Note que esse problema classifica-se na classe de métodos Trust Region em que é necessário resolver iterativamente até que os limites móveis atinjam valores significativamente pequenos.

4.2.4 – SQP Potência Quadrática + Passo 2

Esse método utiliza a solução do método SQP - Potência Quadrática para definir os ângulos de alocação. E o passo 2 é a solução do sistema de Equações 4.3, que considera os ângulos de azimuth constantes e a função objetivo real.

Note que as duas etapas são problemas convexos e que esse algoritmo também é da classe Trust Region, cuja solução implica em resolver o problema quadrático até que os limites móveis convirjam.

4.2.5 – Trust Region SQP

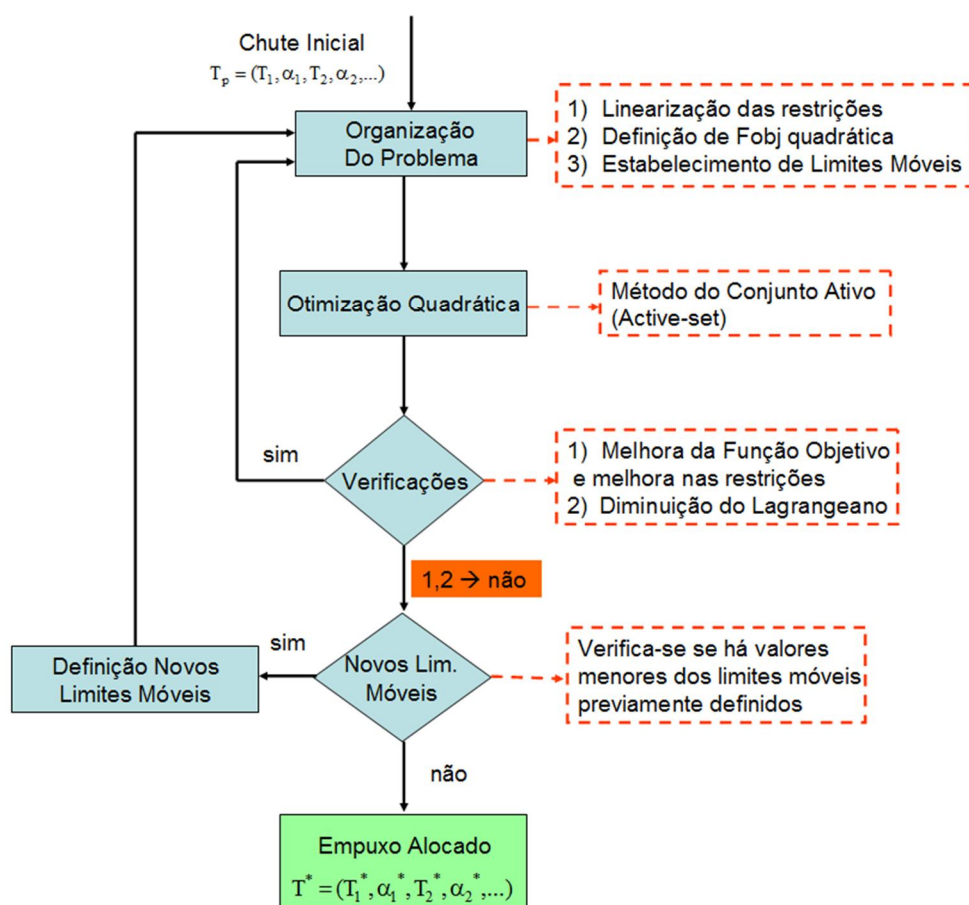
O Trust Region SQP também lineariza as equações de igualdade conforme o método de Taylor (Eq. 4.5). Além disso, a função objetivo real também é aproximada pelo método de Taylor com o objetivo de gerar uma função quadrática, conforme a Eq. 4.7:

$$\text{Potência} = \nabla(T_0) \times T + \frac{1}{2} \nabla^2(T_0) \times T^2 \quad (4.7)$$

Note que ao comparar-se a Eq. 4.7 com a aproximação de Taylor estão faltando termos. Porém esses termos que estão faltando são constantes, e termos constantes na função objetivo não afetam a otimização (SILVA 2011). Além disso, a nova função objetivo apresenta termos lineares, mas isso não faz com que ele deixe de ser quadrática.

Novamente temos um problema de programação quadrática sequencial (PQS), convexo e da classe Trust Region, cuja solução implica em resolver o problema quadrático até que os limites móveis convirjam. A Fig. 4.6 ilustra a dinâmica do funcionamento do algoritmos Trust Region SQP.

Figura 4. 6 – Dinâmica de funcionamento do Trust Region SQP.



4.2.6 – Solução Problema de Programação Quadrática

Como o problema de programação quadrática está presente nos três algoritmos que serão simulados e comparados ao FMINCON, é necessário resolvê-lo. Porém antes de resolvê-lo é necessário analisar quais dos algoritmos desenvolvidos será selecionado. Portanto nessa fase de desenvolvimento, para a solução dos problemas de programação quadrática utilizou-se uma rotina do Matlab.

4.3 – Metodologia de Simulação

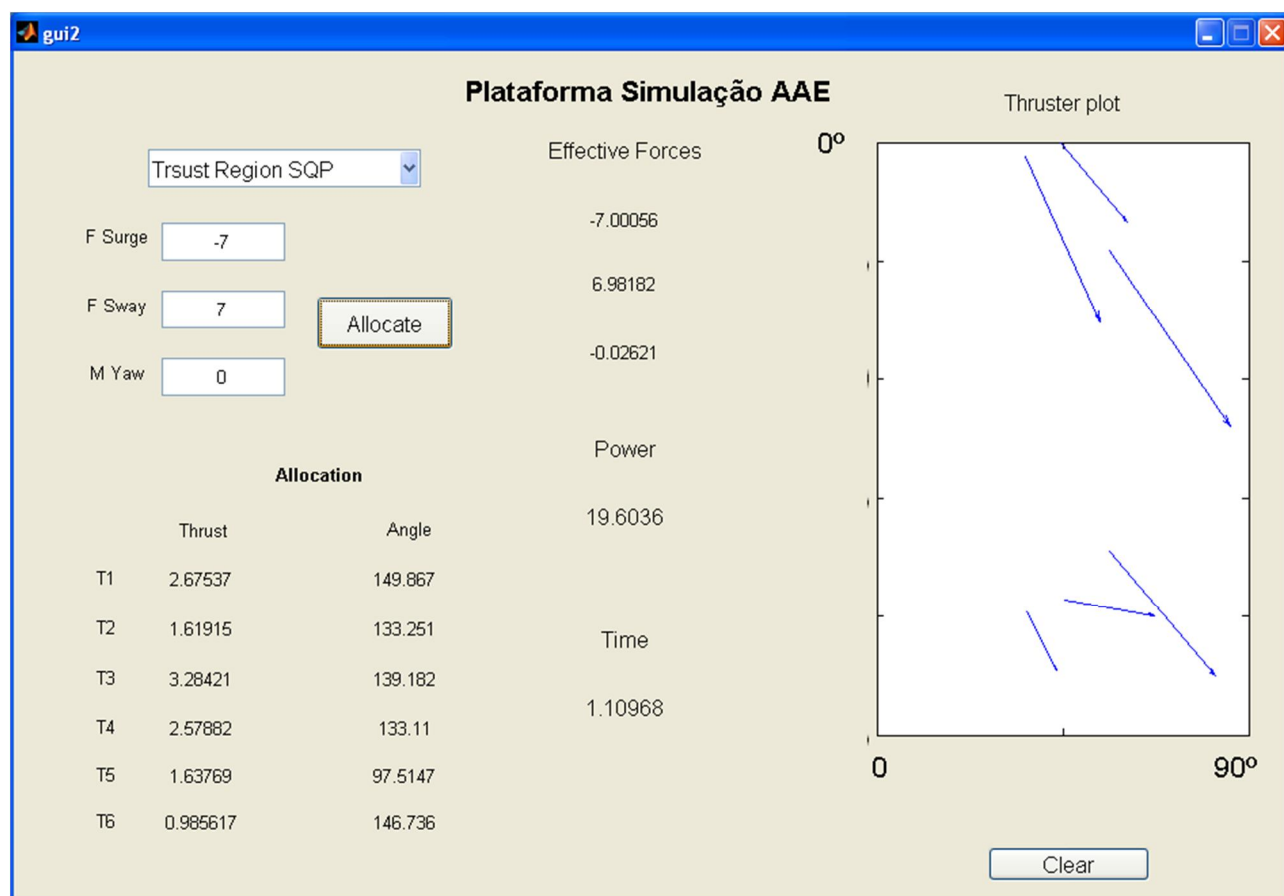
4.3.1 – Plataforma de Simulação dos Algoritmos de Alocação de Empuxo

A plataforma de simulação é um instrumento que permite comparar o funcionamento dos diferentes Algoritmos de otimização de Alocação de Empuxo. A plataforma esta ilustrada na Fig. 4.8, e suas funcionalidades são listadas, como segue:

1. Seleção do Algoritmo de Alocação de Empuxo (AAE) na lista de opções (onde está selecionado o Trust Region SQP).
2. Definição das Forças de Controle.
 - a) Força Longitudinal – F Surge
 - b) Força Transversal – F Sway
 - c) Momento – M Yaw
3. Comandar que a Alocação de Empuxo seja realizada (botão “Allocate”).
4. Informação do empuxo (“Thrust”) e ângulo (“Angle”) desenvolvido por cada um dos seis propulsores.
5. Força Efetiva desenvolvida pelos propulsores (“Effective Forces”) na direção Longitudinal, Transversal e Momento.
6. Potência requerida para gerar as forças requisitadas (“Power”).
7. Tempo de Processamento do Algoritmo para resolver o problema de Otimização da Alocação de Empuxo (“Time”).

Plotagem dos vetores de empuxo gerados por cada propulsor (“Thrust”, “Angle”) a partir de sua posição relativa à Embarcação.

Figura 4. 7 – Plataforma de Simulação de Algoritmos de Alocação de Empuxo (AAE).



4.3.2 – Critérios Avaliados na Simulação

Os critérios avaliados na simulação dos algoritmos são:

1. Potência – Comparar a eficiências dos algoritmos
2. Precisão – Verificar que os algoritmos respeitam as forças de controle
3. Tempo de Processamento – Verificar se o tempo de processamento não torna inviável a aplicação do algoritmo.

Os critérios 2 e 3 são requisitos obrigatórios, ou seja, todos os algoritmos devem ser precisos e apresentar Tempo de processamento aceitável. O critério 1 é de seleção, ou seja, dos algoritmos precisos e de processamento rápido escolher-se-á o mais eficiente.

4.4 – Resultados da Simulação dos Algoritmos

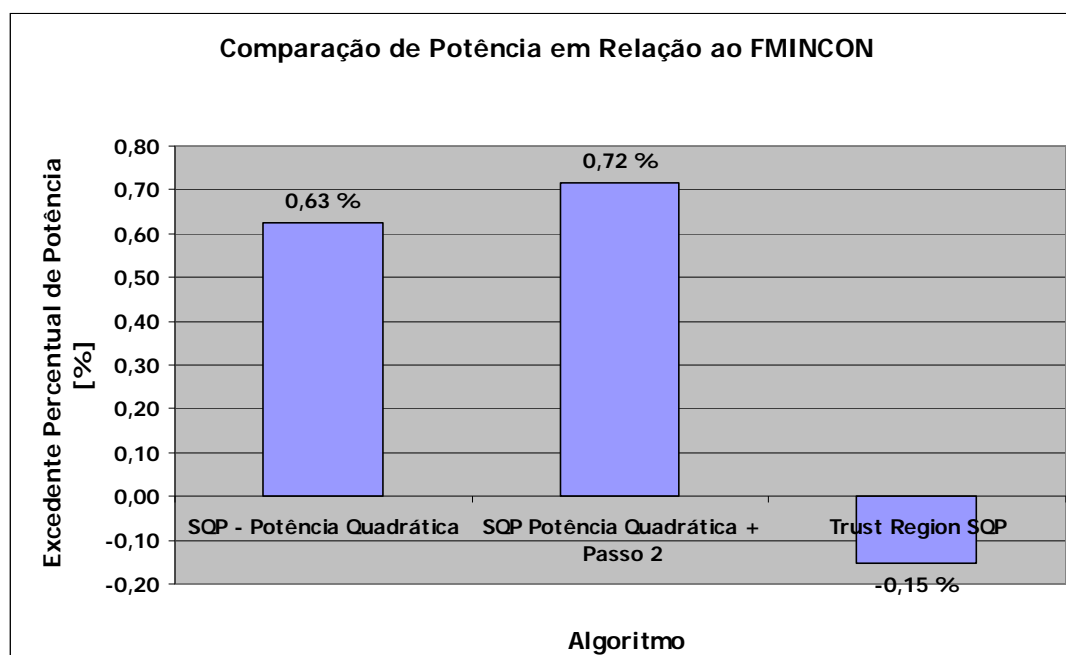
4.4.1 – Análise de Potência

A Tabela 4.1 e a Fig. 4.9 apresentam o consumo médio de potência de cada um dos algoritmos para as diferentes forças de controle comandadas (conforme a Tabela B1).

Tabela 4. 1 – Comparação de Potência requisitada pelos algoritmos.

Análise de Potência		
Algoritmo	Potência Média	Potencia em relação FMINCON
FMINCON	18,6816	0,00 %
SQP - Potência Quadrática	18,7985	0,63 %
SQP Potência Quadrática + Passo 2	18,8155	0,72 %
Trust Region SQP	18,6534	-0,15 %

Figura 4. 8 – Excedente Percentual de Potência dos algoritmos desenvolvidos em relação ao algoritmo de otimização do MATLAB (FMINCON).



O Algoritmo Trust Region SQP é mais eficiente que o algoritmo do MATLAB, economizando cerca de 0,15%. Não obstante, o SQP - Potência Quadrática e o SQP Potência

Quadrática + Passo 2 são um pouco menos eficientes, consumindo 0,63% e 0,72% mais energia, respectivamente.

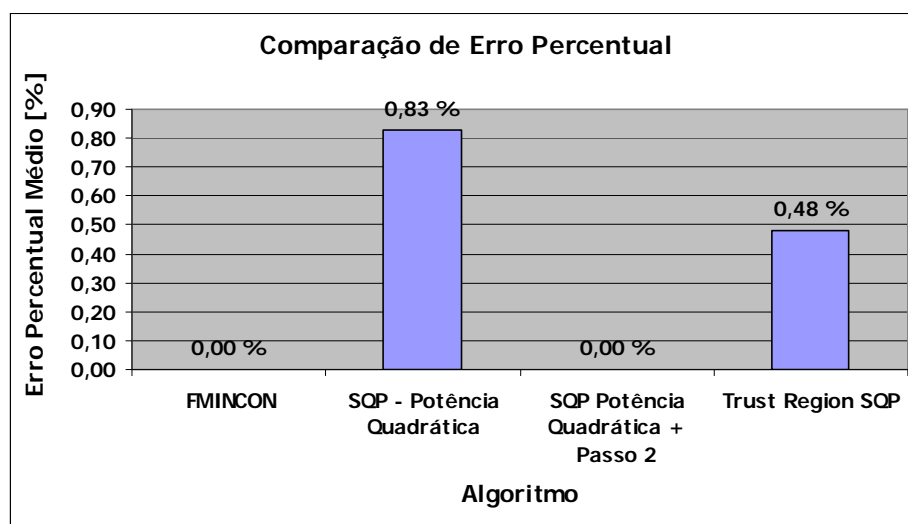
4.4.2 – Análise de Precisão

A Tabela 4.2 e a Fig. 4.10 apresentam o Erro percentual médio de geração da força de controle dos algoritmos (conforme a Tabela B.1).

Tabela 4. 2 – Comparação erro dos algoritmos.

Análise de Erro	
Algoritmo	Erro percentual médio em relação à Fcontrole
FMINCON	0,00 %
SQP – Potência Quadrática	0,83 %
SQP Potência Quadrática + Passo 2	0,00 %
Trust Region SQP	0,48 %

Figura 4. 9 – Comparação do Erro Percentual médio dos algoritmos.



Os algoritmos SQP - Potência Quadrática e Trust Region SQP geram as forças de controle com erros percentuais médios de 0,83% e 0,48%, respectivamente. Enquanto o FIMINCON e o SQP Potência Quadrática + Passo 2 geram as forças de controle sem erro. Porém em termos práticos, os erros encontrados são insignificantes em relação aos fenômenos

associados à embarcação, como a precisão de posicionamento dos propulsores e a geração de empuxo.

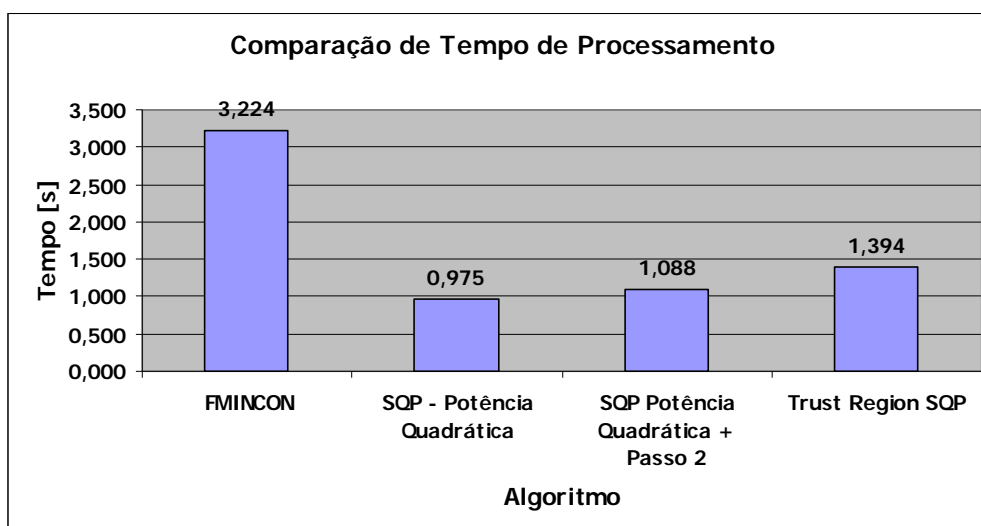
4.4.3 – Análise do Tempo de Processamento

A Tabela 4.3 e a Fig. 4.11 apresentam o Tempo médio de Processamento dos algoritmos (conforme a Tabela B.1).

Tabela 4. 3 – Comparação do Tempo médio de Processamento dos algoritmos.

Análise de Tempo		
Algoritmo	Tempo Médio	Tempo em relação FMINCON
FMINCON	3,224 s	Referência
SQP - Potência Quadrática	0,975 s	30,23 %
SQP Potência Quadrática + Passo 2	1,088 s	33,75 %
Trust Region SQP	1,394 s	43,25 %

Figura 4. 10 – Comparação do Tempo médio de Processamento dos algoritmos.



Nota-se que o FMINCON é significativamente mais lento que os Algoritmos de Alocação de Empuxo desenvolvidos. Entretanto, mesmo o tempo de processamento do FMINCON é bastante inferior aos fenômenos associados à embarcação, como

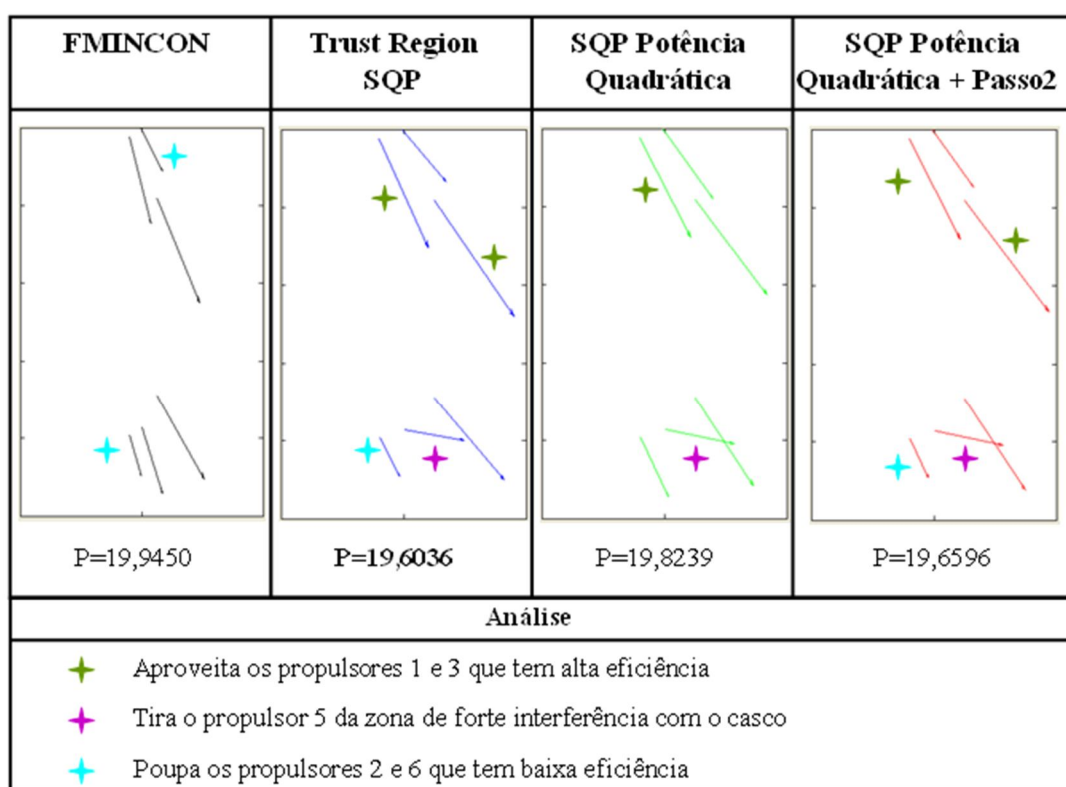
posicionamento angular $\frac{\partial \alpha_i}{\partial t}$ e variação no empuxo $\frac{\partial T_i}{\partial t}$ dos propulsores. Ou seja, o Tempo de Processamento dos algoritmos é insignificante em relação ao intervalo entre os *loops* de alocação.

4.5 – Análise Comparativa dos Algoritmos

Uma vez que foram comparados os algoritmos desenvolvidos em relação aos requisitos técnicos do projeto, vale a pena avaliar comportamento dos mesmos a fim de justificar os resultados encontrados.

A Fig. 4.12 apresenta a alocação dos diferentes algoritmos simulados para ($FSurge = -7N$, $FSway = 7N$ e $Myaw = 0N$), bem como uma análise das vantagens e desvantagens comparativas para cada algoritmo.

Figura 4. 11 – Alocação dos diferentes algoritmos simulados para ($FSurge = -7N$, $FSway = 7N$ e $Myaw = 0N$).



As vantagens que alguns algoritmos aproveitaram foram: a alta eficiência dos propulsores 1 e 3 nessas posições; movimentar o propulsor 5 para tira-lo de uma zona de baixa eficiência devida à interferência com o casco e; requisitar pouca força dos propulsores 2 e 6 que apresentavam baixa eficiência.

Os ângulos dos propulsores dos algoritmos SQP - Potência Quadrática e SQP Potência Quadrática + Passo 2 são os mesmos como era de se esperar, pois o segundo algoritmo utiliza o ângulo de alocação do primeiro.

Além disso, o Tempo de Processamento do SQP Potência Quadrática + Passo 2 é superior ao do SQP - Potência Quadrática como esperado, pois ele o contém dentro de sua solução.

O SQP Potência Quadrática + Passo 2 explora melhor a alocação de força que o SQP - Potência Quadrática, pois considera a função objetivo correta.

Todos os algoritmos desenvolvidos são mais agressivos no posicionamento dos propulsores do que o FMINCON. Isso se explica no processo de otimização dos mesmos, pois ao realizar a linearização das restrições de igualdade o ângulo α multiplica o empuxo T_0 , ou seja, do ponto de vista qualitativo é como se o ângulo pudesse gerar força ao mesmo tempo em que não aumenta o valor da função objetivo. Portanto intuitivamente o algoritmo “pensa” em variar significativamente o ângulo para alcançar a força de controle sem afetar a função objetivo. Por isso justifica-se matematicamente e intuitivamente a necessidade dos limites móveis, pois caso contrário os ângulos de azimuth variariam de maneira exagerada.

O Trust Region SQP apresenta resultados melhores quando comparado ao SQP Potência Quadrática + Passo 2 em termos de eficiência (valor final da função objetivo). Isso se explica no fato do primeiro realizar a alocação em passo único já considerando a função objetivo real, enquanto o segundo mesmo que considere, no Passo 2, a funções objetivo real, apresenta perdas intrínsecas à separação do problema em posicionamento dos propulsores e geração de força.

4.6 – Seleção do novo Algoritmo de Alocação de Empuxo

Tendo em vista que todos os algoritmos satisfazem os requisitos de Tempo de processamento e Erro na geração das Forças de Controle. E que nenhum aspecto negativo desse algoritmo foi levantado na análise comparativa dos algoritmos. Opta-se pelo Algoritmo com menor consumo de Potência, o Trust Region SQP.

Uma vez selecionado o algoritmo de alocação de empuxo, é necessário resolver o problema apresentado na seção 4.2.6.

4.6 – Programação Quadrática

A solução de problemas de otimização com função objetivo quadrática e restrições lineares é bastante conhecida. Existem softwares eficientes para a solução deste tipo de problema no mercado. Há também softwares livres que são dedicados a este específico problema de otimização. Entretanto, para o desenvolvimento de uma ferramenta que possa ser utilizada de maneira prática e eficiente por empresas e laboratórios de avaliação de operações DP, essas opções apresentam características desfavoráveis.

A utilização de um software pago, de maneira geral encarece de maneira significativa um produto. Além disso, a implementação de um software pago, dentro de outro software possivelmente comercial atrela dificuldades de negociação e carga de responsabilidade.

A utilização de um software livre dedicado à solução de problemas de otimização quadráticos, requer a adição de diversas bibliotecas e modelagem do problema para os moldes do produto selecionado. A adição das bibliotecas pode gerar problemas de compatibilidade, dependendo do sistema operacional instalado. Adiante, qualquer problema de adaptação ao software livre, ou de mau funcionamento do mesmo podem prejudicar o desempenho do simulador desenvolvido.

Portanto, levando em conta os aspectos descritos e a quantidade de trabalho necessária para ajustar o problema de otimização quadrático a um programa externo e suas bibliotecas, decide-se por desenvolver um programa que resolva este problema de otimização.

4.6.1 – Solução Convencional

O Método convencional para a solução de problemas de programação quadráticos é simples. Basta definir o Lagrangeano, que consiste em unir a função objetivo às igualdades e desigualdades da seguinte maneira:

$$L = \underbrace{\left\{ Obj - (\lambda_{igualdade})_i \cdot h_i \right\}}_{\text{parte 1}} - \underbrace{\left\{ (\lambda_{desigualdade})_i \cdot g_i \right\}}_{\text{parte 2}} \quad (4.8)$$

O passo seguinte é resolver uma série de problemas de otimização quadráticos em que a parte 1, que consiste da função objetivo e as restrições de igualdade, sempre estão presentes; e a parte 2, desigualdades estão em parte presentes. Após a solução de cada subproblema verifica-se se as condições KKT foram respeitadas, e se as restrições de desigualdade que não foram consideradas foram respeitadas; caso isso ocorra, encontra-se o mínimo global.

Um método consistente de realizar esse processo é ativar e desativar as desigualdades, definindo os valores de $\lambda_{desigualdade}$. Quando a desigualdade não está ativa, $(\lambda_{desigualdade})_i = 0$, quando a mesma está ativa, ela faz parte do problema e $(\lambda_{desigualdade})_i$ é automaticamente calculada, pelas seguintes equações que resolvem analiticamente o problema de otimização:

$$\frac{\partial L}{\partial T} = \frac{\partial L}{\partial \lambda} = 0 \quad (4.9)$$

Normalmente utiliza-se uma tabela, conforme ilustrado abaixo para controlar quais restrições de desigualdade estão ativas. Note que na primeira linha todas as restrições estão inativas, na segunda linha apenas a primeira desigualdade está ativa e assim por diante. Em resumo, cada linha representa um subproblema da otimização, no qual, as restrições inativas não são consideradas, e as desigualdades ativas são consideradas igualdades. Para maiores informações sobre o método convencional de solução de problemas de otimização quadráticos, ver Silva (2011).

Tabela 4. 4 – Definição consistente de quais desigualdades estão ativas.

	$\lambda_{desigualdade_1}$	$\lambda_{desigualdade_2}$	$\lambda_{desigualdade_3}$
Subproblema 1	0	0	0
Subproblema 2	$\neq 0$	0	0
:	:	:	:
Subproblema n	$\neq 0$	$\neq 0$	$\neq 0$

Embora o método seja consistente, relativamente simples, e garante que caso exista um mínimo global, o mesmo será encontrado, há um fator que complica sua implementação.

O numero de subproblemas cresce exponencialmente com a quantidade de desigualdades. Demonstra-se: cada desigualdade pode estar em apenas 2 estados, ativa ou inativa. O numero de subproblemas refere-se às possíveis combinações de das desigualdades (ativas e inativas). Portanto, caso existam n desigualdades, e constatando-se que cada um pode estar em apenas 2 estados, conclui-se que existem 2^n subproblemas de otimização possíveis.

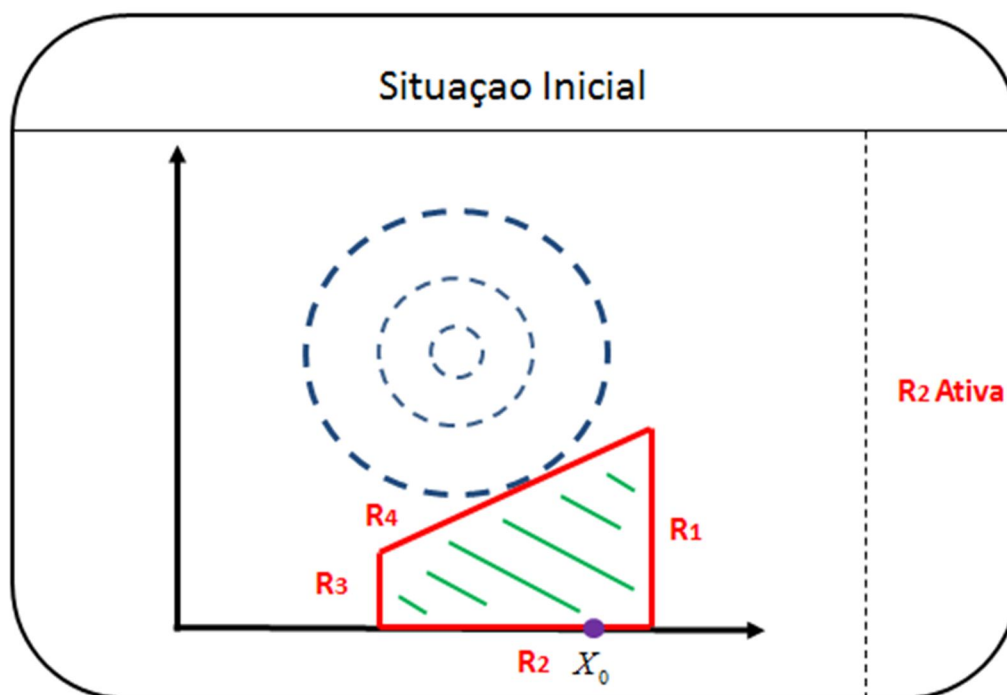
Dessa forma, ao considerar uma embarcação com 6 propulsores azimutais, ou seja, 12 variáveis de projeto $\{T, \alpha\}$, e que cada variável apresenta 2 limites móveis (superior e inferior), encontram-se 24 desigualdades. Portanto para resolver esse problema de otimização, seriam necessários resolver $2^{24} = 16.777.216$ subproblemas de otimização. Essa constatação torna inviável a aplicação do método convencional. Portanto um método diferente deve ser utilizado para resolver o problema de otimização quadrático dentro da rotina de alocação de empuxo.

4.6.2 – Método dos gradientes reduzidos (*Active-set*)

O método dos gradientes reduzidos (*Active-set*) é o mais indicado, pois o funcionamento do mesmo segue a seguinte premissa: Verificar de maneira iterativa quais restrições de desigualdade são de fato restritivas, ou seja, não permitem que seja encontrada uma solução melhor, e que portanto devem estar ativas. As iterações não requerem percorrer todo o conjunto de possíveis desigualdades.

Esse método é explicado com a ajuda de um exemplo para facilitar sua compreensão. Figura 4.12 ilustra a situação inicial de um problema de otimização com função objetivo quadrática (curvas de nível em azul), quatro restrições de desigualdade (em vermelho) que definem o espaço viável (área hachurada) e uma solução inicial (X_0) sobre R_2 . Verifica-se neste caso que R_2 esta ativa.

Figura 4. 12 – Método dos gradientes reduzidos: Situação inicial.

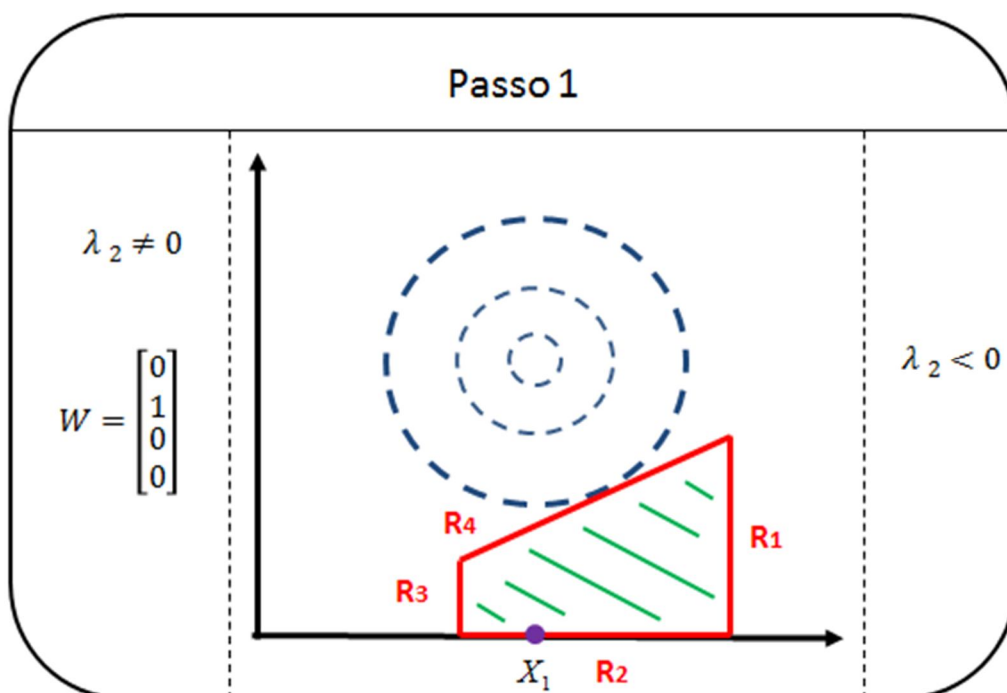


O primeiro passo é verificar quais restrições estão ativas, neste caso apenas R_2 . Tendo em vista que apenas a segunda restrição está ativa, o vetor W (conjunto das restrições ativas, ou Conjunto Ativo), recebe o valor 1 apenas em sua segunda linha. Isso indica que nesta iteração, o problema é resolvido considerando-se apenas a função objetivo e a segunda restrição, ou seja, a desigualdade referente à segunda restrição, é considerada uma igualdade.

A solução deste problema é simples, e requer apenas a aplicação da seguinte condição de otimização (Eq. 4.9). Essa condição resulta em um sistema linear, que fornece um valor para as variáveis de projeto e para as variáveis de Lagrange (λ).

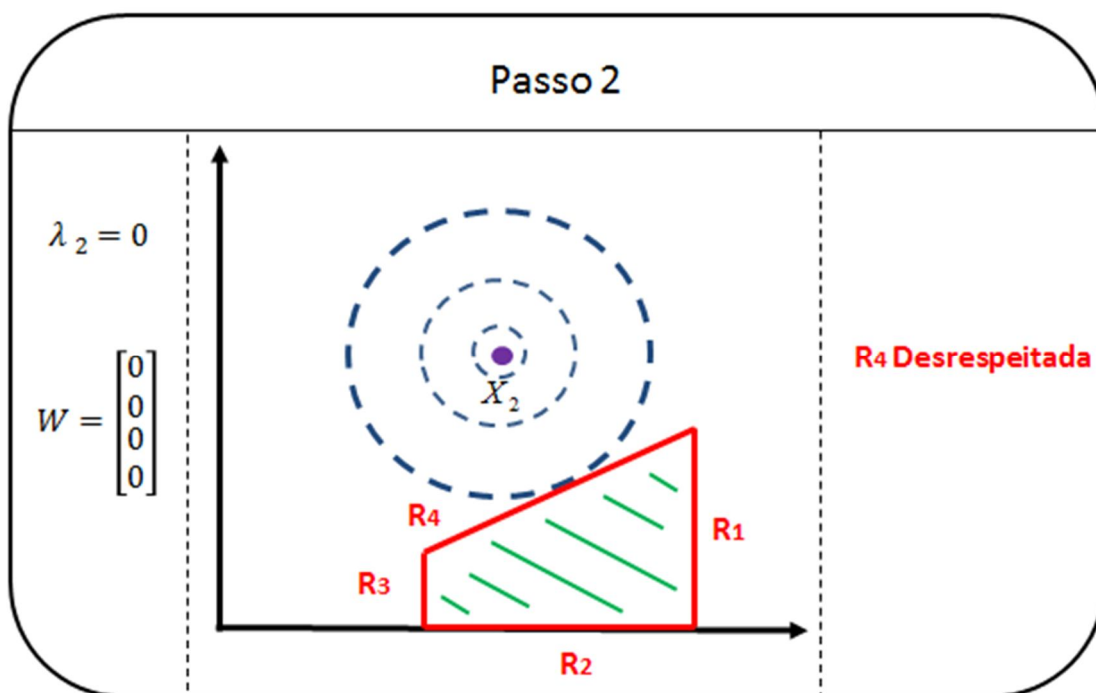
Ao resolver esse problema, ilustrado na Figura 4.13 determina-se uma nova solução X_1 e os valores para as variáveis de Lagrange (λ), que são 0 para todas as restrições não ativas, e neste caso verifica-se ($\lambda_2 < 0$), que não corresponde às condições KKT, portanto não configura X_1 como um ponto de mínimo local.

Figura 4. 13 – Método dos gradientes reduzidos: Primeiro passo.



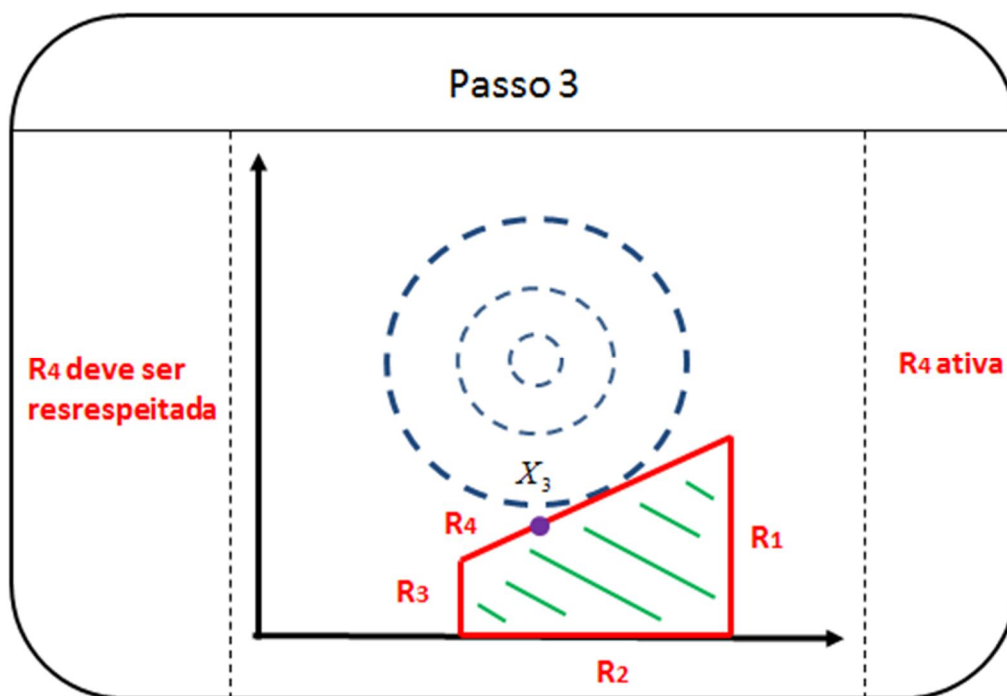
Tendo em vista que R_2 não é uma restrição que limita a busca pelo ponto de mínimo $\lambda_2 < 0$, retira-se a mesma do conjunto ativo, e uma nova otimização é realizada. Neste segundo passo ilustrado em Figura 4.14 soluciona-se o problema de alocação levando-se em conta apenas a função objetivo. A solução encontrada X_2 não respeita R_4 .

Figura 4. 14 – Método dos gradientes reduzidos: Segundo passo.



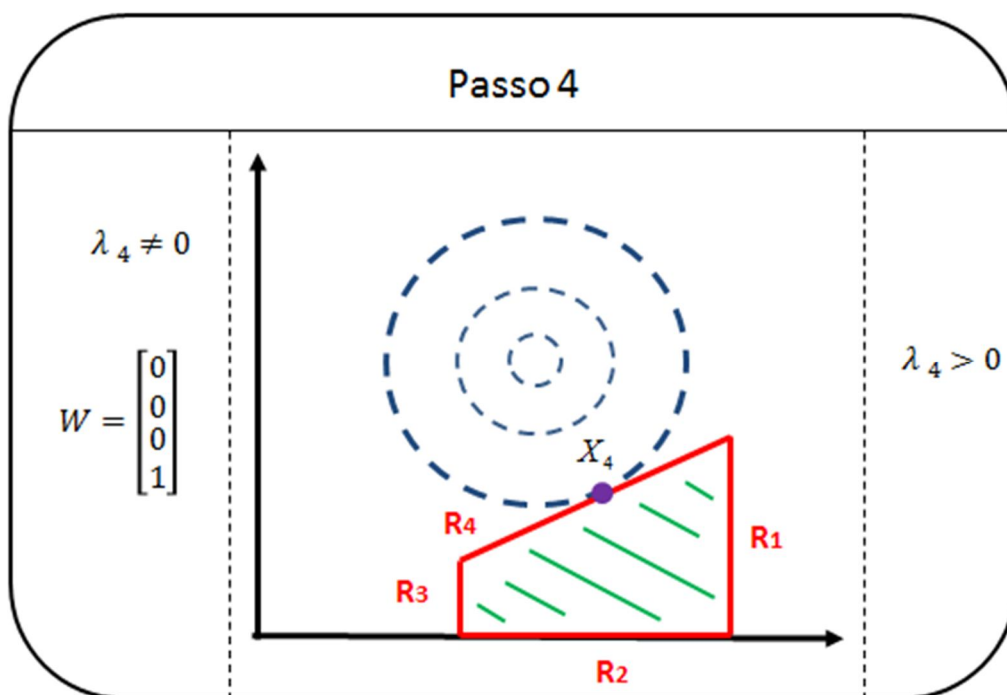
Note que a ultima iteração desrespeitou uma restrição do problema. Como ilustrado na Figura 4.15, essa iteração garante que R_4 seja respeitada. Isso é realizado diminuindo o tamanho do passo dado na ultima iteração de maneira que o mesmo seja o maior possível respeitando as restrições do problema. Ou seja, $\overrightarrow{X_2 - X_1}$ define a direção do passo nessa iteração (note que o ponto inicial desse passo é X_1), e o tamanho do passo é o maior possível levando-se em conta as restrições do problema. Isso resulta na solução X_3 que respeita R_4 .

Figura 4. 15 – Método dos gradientes reduzidos: Terceiro passo.



Note que a solução X_3 torna R_4 ativa, portanto a mesma deve ser adicionada ao conjunto ativo W . Isso implica na solução do problema considerando-se a função objetivo e R_4 como uma igualdade. A solução desse problema é simples e derivada de um sistema linear, baseado na condição de otimização. Note que a solução X_4 tem $\lambda \geq 0$, o que respeita as condições KKT, e garante que esse ponto é o mínimo global.

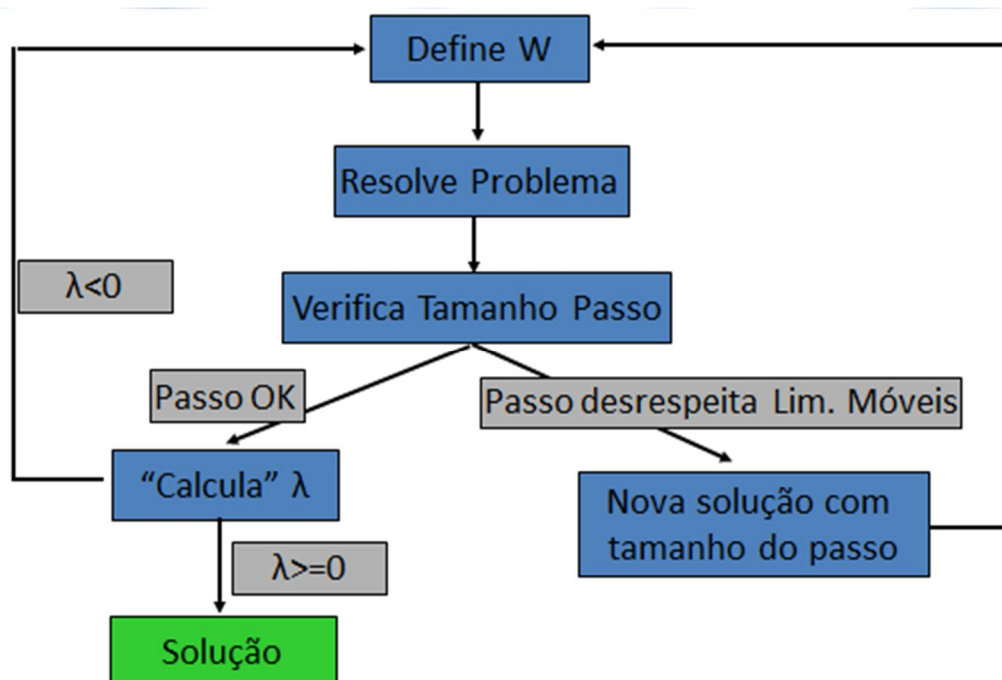
Figura 4. 16 – Método dos gradientes reduzidos: Quarto passo.



O loop que representa a lógica do método dos gradientes reduzidos está representado em Figura 4.17.

- 1) **DEFINE W:** O primeiro passo é definir o conjunto ativo, ou seja, o vetor W , que assinala quais restrições de desigualdade devem ser consideradas nessa iteração.
- 2) **RESOLVE PROBLEMA:** A seguir resolve-se o problema considerando o conjunto de restrições de desigualdade ativas, a função objetivo e as restrições de igualdade.
- 3) **VERIFICA TAMANHO PASSO:** Ao obter-se a nova solução verifica-se se a mesma respeita todas as restrições de desigualdade.
 - a. Caso respeite verifica-se λ . **CALCULA λ**
 - i. Caso $\lambda \geq 0$, o que configura a aceitação das condições KKT e configura a mesma como ponto de mínimo.
 - ii. Caso contrário, a restrição que apresenta o menor valor de λ é descartada do conjunto ativo para a próxima iteração.
 - b. Caso o passo não respeite as desigualdades (limites móveis da alocação de empuxo), obtém-se uma nova solução, na direção que liga a solução anterior à solução recém obtida. A essa se designa o nome: **NOVA Solução COM TAMANHO DO PASSO.**

Figura 4. 17 – Logica do método do dos gradientes reduzidos.



As principais funções referentes do loop do Active-set são:

4.6.2.1 – Define W

Verifica sobre quais restrições de desigualdade a solução atual encontra-se. E desconsidera alguma desigualdade que apresente $\lambda < 0$, da iteração anterior.

4.6.2.2 – Resolve Problema

O método para a solução do problema, que contem a função objetivo, as restrições de igualdade e desigualdade e o vetor W (que contem as desigualdades que devem ser consideradas igualdades) é o seguinte:

Monta-se o sistema linear geral de equações, derivado da condição de otimização, da seguinte maneira:

- a. Para $\frac{\partial L}{\partial T} = 0$ utilizam-se todas as desigualdades independente de W.
- b. Para $\frac{\partial L}{\partial \lambda_{igualdade}} = 0$ utilizam-se todas as desigualdades independente de W.

c. $\frac{\partial L}{\partial (\lambda_{desigualdade})_i} = 0$

i. Se $W_i = 0$, a equação utilizada é $(\lambda_{desigualdade})_i = 0$

ii. Se $W_i = 1$, ou seja, essa restrição deve estar ativa, utiliza-se a equação de desigualdade como uma igualdade.

O sistema linear resolvido fornece novos valores para a alocação (T, α) , $\lambda_{igualdade}$ e $\lambda_{desigualdade}$.

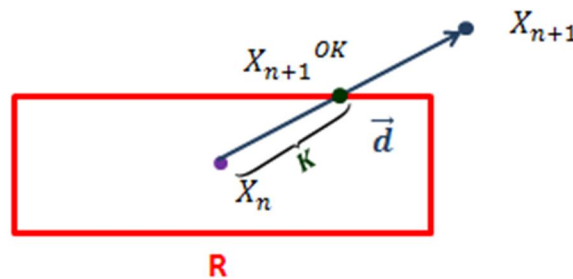
4.6.2.3 – Verifica tamanho passo

Verifica se todas as desigualdades são respeitadas.

4.6.2.3 – Nova solução com tamanho do passo

O funcionamento desta função está ilustrado na Figura 4.18. Inicialmente calcula-se o vetor \vec{d} , que liga a solução anterior à nova. A partir de X_n , verifica-se qual a maior distância que pode ser percorrida (K) até atingir a primeira desigualdade. Percorre-se K, na direção \vec{d} , e obtém-se a nova solução X_{n+1}^{OK} , que respeita todas as desigualdades.

Figura 4. 18 – Solução considerando-se as desigualdades.



Para maiores informações sobre o método dos gradientes reduzidos ver: Nocedal e Wright (2006).

Entretanto esse algoritmo apresenta uma falha intrínseca. Caso não seja possível resolver as igualdades, ou seja, as equações referentes às forças requeridas pelo sistema de

controle, considerando-se as desigualdades (limites móveis), ao montar o sistema linear geral montado na função **Resolve Problema** nota-se que o mesmo é um Sistema Impossível (SI). Essa situação é comum em casos em que as condições ambientais são significativamente adversas, e as forças requisitadas excedem a saturação (T_{max}) dos propulsores da embarcação.

Portanto, numa situação adversa, permitir que o controlador tente resolver um sistema impossível é perigoso, pois o mesmo pode gerar respostas aleatórias, ou simplesmente não funcionar. Isso poderia causar em acidentes. Dessa forma, é necessário estabelecer um protocolo para a situação em que não é possível gerar as forças requisitadas considerando-se a saturação dos propulsores (**Protocolo de Saturação**). Além disso, é necessário verificar os casos em que esse protocolo deve ser seguido, ou seja, necessita-se desenvolver um algoritmo que verifique se as igualdades podem ser resolvidas considerando-se as desigualdades.

4.6.3 – Verificação de congruência de Igualdades e Desigualdades

A verificação que deve ser realizada é se existe uma solução para as igualdades dentro do espaço viável definido pelas desigualdades. A Tabela 4.5 representa claramente as equações que devem ser consideradas e seu sentido físico e matemático.

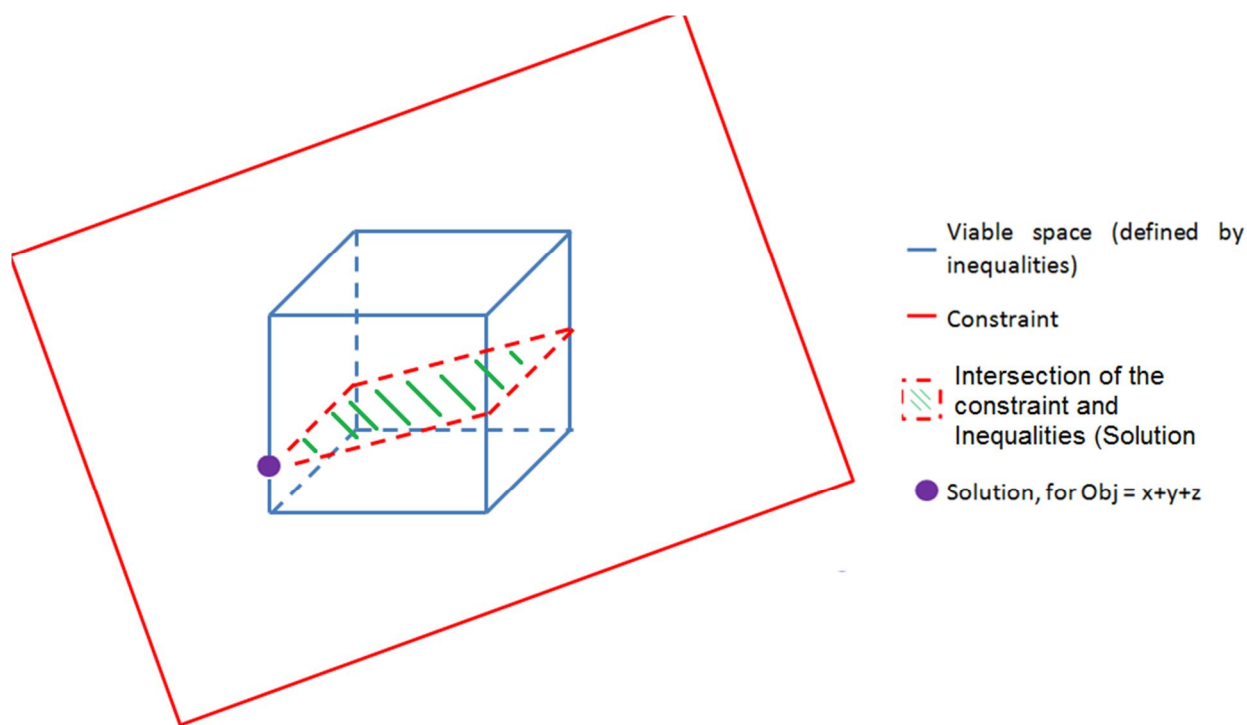
Tabela 4. 5 – Representação de igualdades e desigualdades.

Fisicamente	Equações	Notação Matemática
Forças Requeridas	$\left\{ \begin{array}{l} \sum_{i=1}^N F_{x_i} \cdot \eta_i - F_{surge} = 0 \\ \sum_{i=1}^N F_{y_i} \cdot \eta_i - F_{sway} = 0 \\ \sum_{i=1}^N F_{y_i} \cdot \eta_i \cdot (x_i) - \sum_{i=1}^N F_{x_i} \cdot \eta_i \cdot (y_i) - M_{yaw} = 0 \end{array} \right.$	Igualdades
Saturação e Limites Móveis	$\left\{ \begin{array}{l} \text{Saturacao} \{ T_i \leq T_{max_i} \\ \text{Limites Moveis} \left\{ \begin{array}{l} \Delta T_i \leq \Delta T_{max_i} \\ \Delta \alpha_i \leq \Delta \alpha_{max_i} \end{array} \right. \end{array} \right.$	Desigualdades

O Simplex é um algoritmo de otimização para problemas lineares bastante conhecido e utilizado. Note, que a verificação é realizada em equações lineares, pois as forças requeridas foram linearizadas conforme a aplicação da programação quadrática sequencial e os limites móveis e saturação são restrições de caixa, por definição lineares.

A interpretação matemática do funcionamento do Simplex é ilustrado na Figura 4.19. Note que as igualdades (forças requeridas) podem ser representadas por planos, pois são lineares. Ademais, os limites móveis e saturação podem ser representados como uma caixa, pois os mesmos sempre definem os limites inferior e superior de cada variável. Se existe uma intersecção entre o plano e a caixa, significa que existe um subespaço que contem a solução do problema.

Figura 4. 19 – Funcionamento do algoritmo Simplex.



Portanto baseado no conceito do Simplex a verificação da possibilidade de resolver o sistema de igualdades e desigualdades pode ser resolvida da seguinte maneira.

- 1) Consistentemente, escolhem-se $2n - 3$ desigualdades (sendo $2n$ o numero de variáveis) que devem ser respeitadas.

- 2) Atribui-se tais valores a essas variáveis.
- 3) Resolve-se o sistema de 3 igualdades, que apresenta 3 incógnitas (as 3 variáveis as quais não foram atribuídas os valores respectivos de suas desigualdades).
 - a. Caso essas 3 variáveis respeitem suas equações de desigualdade conclui-se que é possível resolver as igualdades considerando-se as desigualdades.
 - b. Caso contrario, escolhe-se consistentemente uma nova combinação de desigualdades.
- 4) Se todas as combinações possíveis de desigualdades foram testadas e não foi possível resolver as igualdades trata-se de um sistema impossível, e aplica-se o protocolo de saturação.

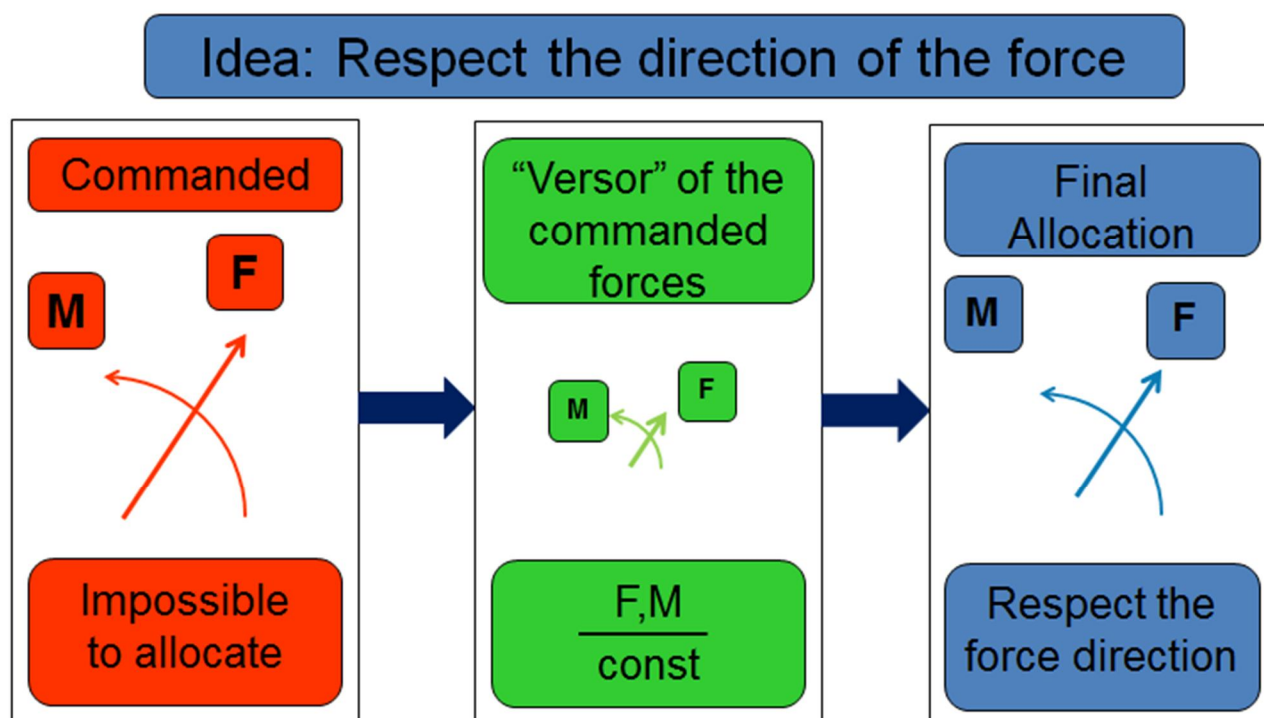
4.6.4 – Protocolo de Saturação

O Protocolo de Saturação é utilizado quando não é possível gerar as forças requeridas considerando-se a saturação dos propulsores, que limitam os limites móveis. Uma solução segura em condições adversas, é que a direção das forças geradas seja a mesma que as forças comandadas, o que evita o comportamento errático e inesperado da embarcação.

O processo que garante que a direção das forças geradas seja o mesmo que o das forças requeridas esta representado na Figura 4.20.

- 1) Inicialmente constata-se que não é possível gerar as forças requeridas.
- 2) Define-se o Versor das forças comandadas, que nada mais é que dividir o vetor das forças requeridas por um valor constante. Tendo em vista que o Versor das forças comandadas apresenta valores de ordem de grandeza inferior aos da força comandada, os mesmos podem ser gerados com relativa facilidade pelos propulsores da embarcação.
- 3) Finalmente, multiplica-se o empuxo de cada propulsor por um mesmo valor constante. Esse valor constante satura os propulsores que estão mais próximos de seus valores máximos de empuxo. Garantindo assim, que a força comandada terá a mesma direção da força requerida.

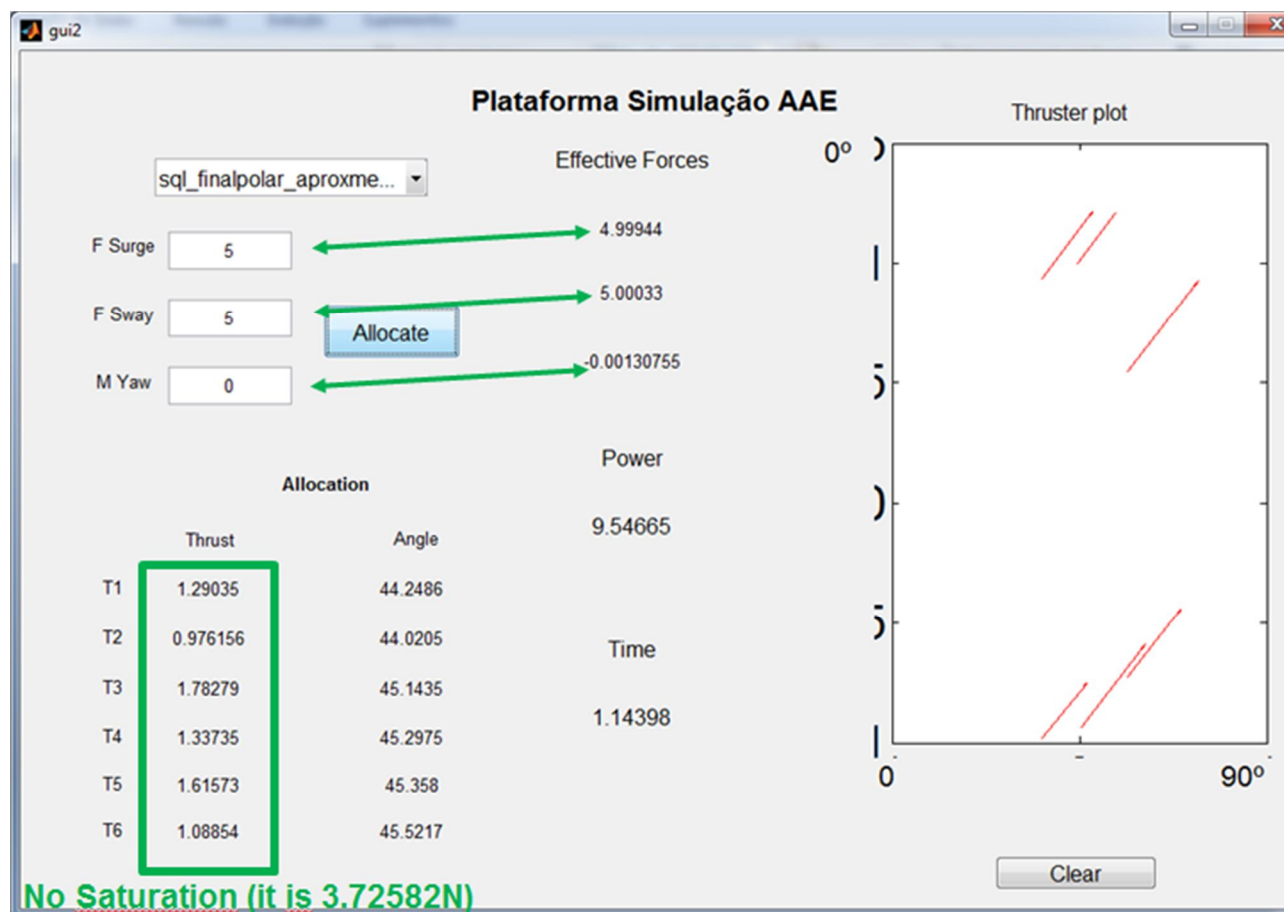
Figura 4. 20 – Processo do Protocolo de Saturação.



As Figuras a seguir ilustram o funcionamento do Protocolo de Saturação.

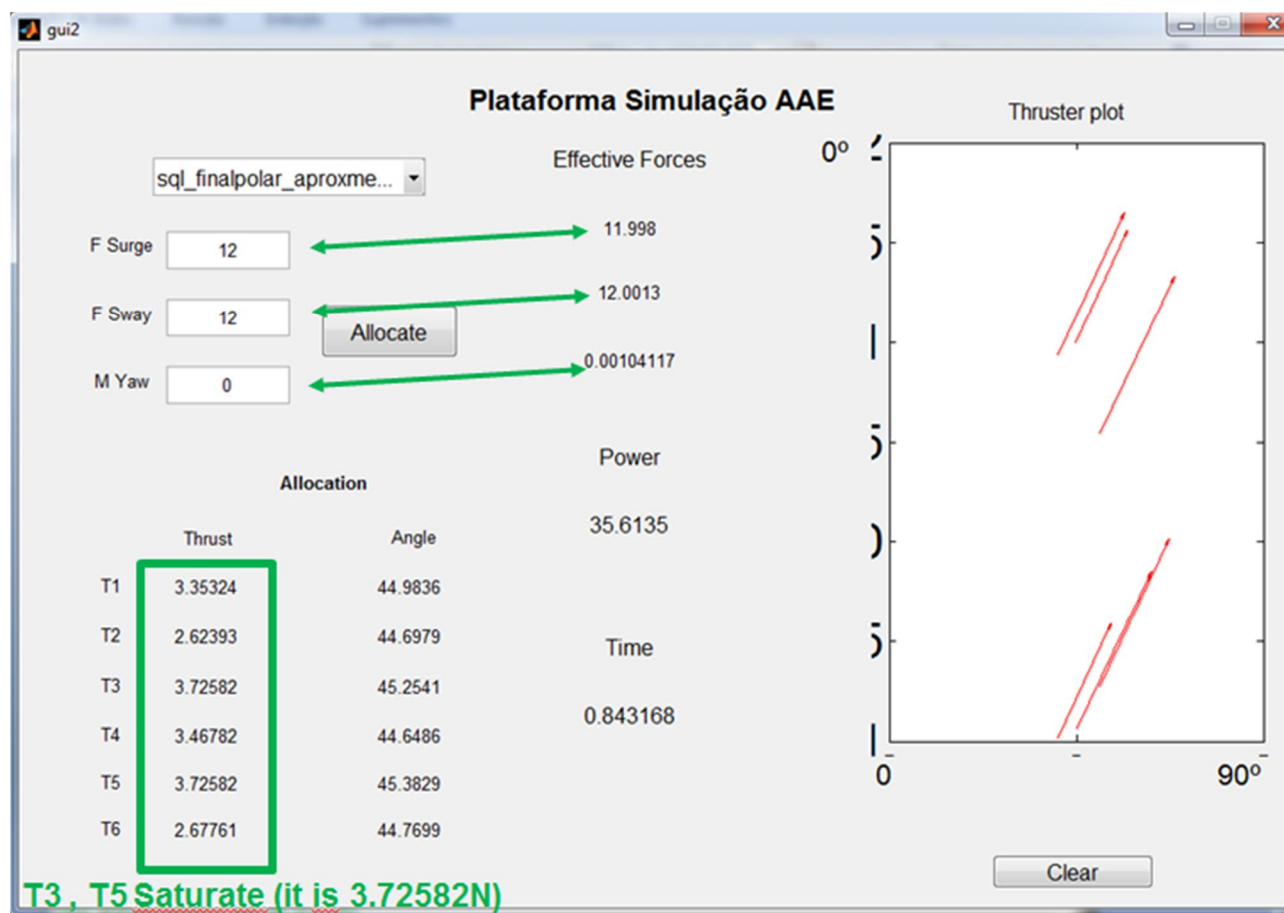
A Figura 4.21 ilustra uma situação normal em que não é necessário acionar o Protocolo de Saturação.

Figura 4. 21 – Alocação em condições normais.



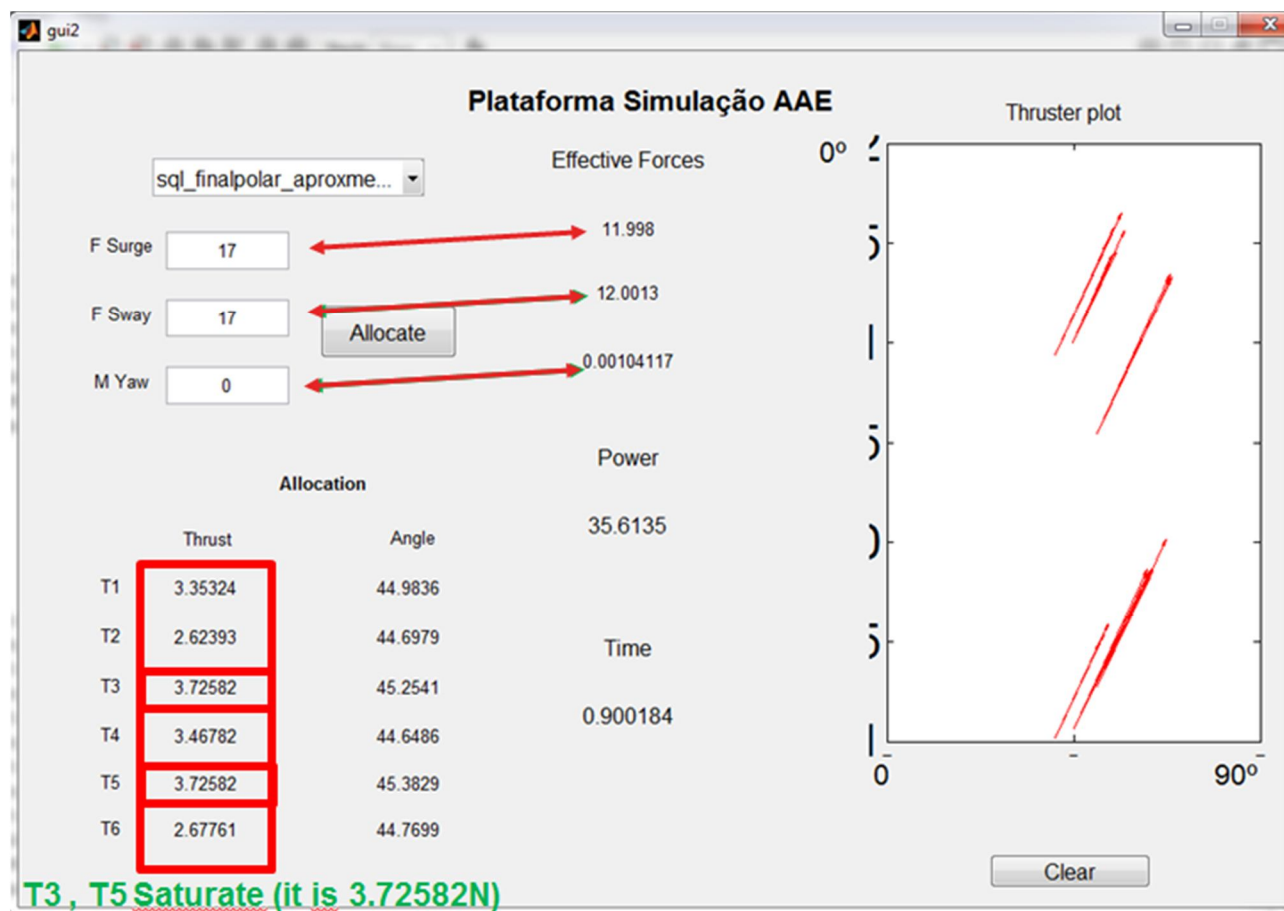
A Figura 4.22 ilustra uma situação em que os propulsores estão saturados, mas que não foi necessário acionar o Protocolo de Saturação.

Figura 4. 22 – Alocação em que ocorre saturação dos propulsores.



A Figura 4.23 ilustra uma situação em que os propulsores estão saturados, pois foi necessário acionar Protocolo de Saturação. Note que as forças efetivas são inferiores as forças requeridas.

Figura 4. 23 – Atuação do Protocolo de Saturação.



Finalmente é possível constatar que nos 3 últimos exemplos a direção das forças era a mesma e que o Protocolo de Saturação garante que as forças geradas pela embarcação, caso não seja iguais às forças requeridas, tem a mesma direção da mesma. Existem outras soluções e propostas para as condições em que não é possível alocar as forças requeridas, entretanto a solução escolhida preza pela segurança e pelo comportamento previsível da embarcação.

Outro Protocolo de Saturação bastante utilizado é priorizar a geração do momento em detrimento das outras direções (Surge e Sway). Essa escolha preza por manter a embarcação com a mesma direção, mesmo que ocorram deslocamentos laterais e longitudinais, que a princípio não elevam as forças aplicadas sobre a mesma, considerando que a priori a

embarcação opera na condição *weathervane*, na qual as forças aplicadas sobre a mesma são as menores possíveis. Normalmente utilizam-se pesos para definir a prioridade relativa de cada direção ($P_{Surge}, P_{Sway}, P_{Yaw}$). Uma solução interessante que trabalha com essa ideia pode ser encontrada em Skjetne e Kjerstad (2013), onde o autor trabalha com dois conceitos básicos, o *null-space* e a aplicação da matriz pseudo-inversa.

4.7 – Analise Final

A fim de verificar se os algoritmos Simplex e o método dos gradientes reduzidos, implementados na alocação de empuxo, apresentam os resultados satisfatórios previamente vistos no Trust Region SQP, novas simulações foram realizadas. Os resultados da Tab. 4.6 apresentam uma rápida comparação entre a alocação realizada com o Trust Region SQP, com a otimização quadrática resolvida pela rotina do Matlab em relação à implementação do Active-set e simplex. Note que em média a Potencia requisitada diminui marginalmente, enquanto o erro caiu 50% em relação à implementação anterior. O tempo médio de processamento também é inferior, mas vale ressaltar que as simulações foram realizadas em computadores diferentes. De qualquer forma o tempo de processamento esta de acordo com a implementação em softwares de alocação de empuxo em tempo real.

Tabela 4. 6 – Comparação do Trust Region SQP com e sem o desenvolvimento do Simplex e do Active-set.

Força Requerida	Trust Region SQP			Simplex e Active-set implementados		
$[F_x, F_y, M_z]$	Tempo	Potência	erro	Tempo	Potência	erro
[7; 7; 0]	1,002	15,8125	0,09%	0,834	15,8425	0,09%
[-7; 7; 0]	1,286	19,6036	0,31%	0,641	19,6296	0,07%
[-7; -7; 0]	1,362	19,0812	0,51%	0,765	19,0578	0,21%
[7; -7; 0]	1,234	19,3467	0,66%	0,665	18,8336	0,39%
[-10; 0; 0]	2,176	19,8260	0,55%	0,463	19,8461	0,38%
[10; 0; 0]	1,491	16,9549	0,25%	0,749	16,9791	0,15%
[0; -10; 0]	1,210	19,9492	1,01%	0,772	20,0326	0,52%
Médias	1,394	18,6534	0,48%	0,698	18,6030	0,26%
Comparação				-50%	-0%	-46%

Portanto a implementação do Active-set, Simplex e Protocolo de Saturação finalizam o algoritmo de alocação de empuxo. Os resultados obtidos pelas simulações revelam que o mesmo é capaz de lidar com a saturação dos propulsores e os fenômenos de interação. Note que o desenvolvimento deste ultimo conjunto de rotinas mantém a eficiência do algoritmo de alocação previamente demonstrada, e elimina a dependência de qualquer rotina exclusiva do software MATLAB.

O capítulo seguinte apresenta a aplicação do algoritmo de alocação de empuxo desenvolvido em software MATLAB para a simulação de operação DP no domínio do tempo (*Time Domain Simulation*).

5 – Resultados

5.1 – Descrição da Embarcação

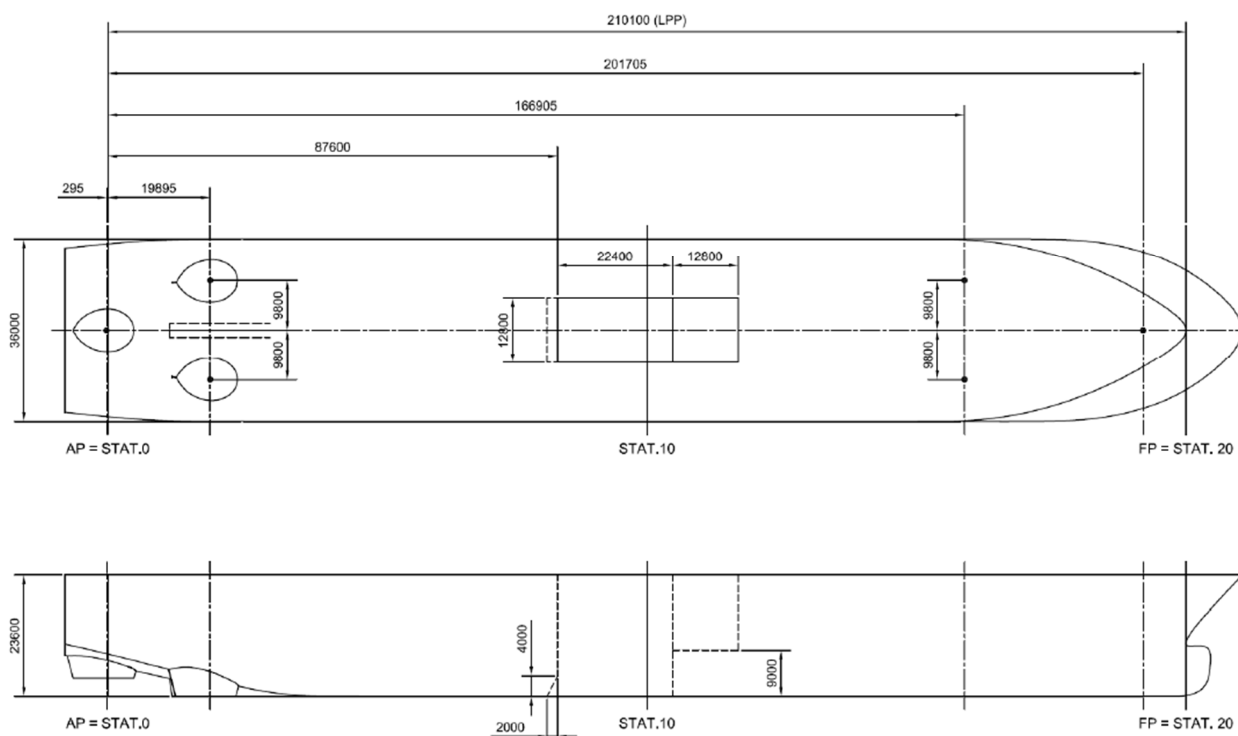
A Fig. 5.1 apresenta o modelo da embarcação utilizado nas simulações de operação DP. Trata-se de um navio de perfuração com 6 propulsores azimutais agrupados em 2 clusters triangulares: na proa e na popa.

Figura 5. 1 – Modelo da embarcação.



A Fig. 5.2 apresenta a planta da embarcação com suas medidas originais. Note que é possível determinar a posição dos propulsores em relação ao centro gravitacional da mesma.

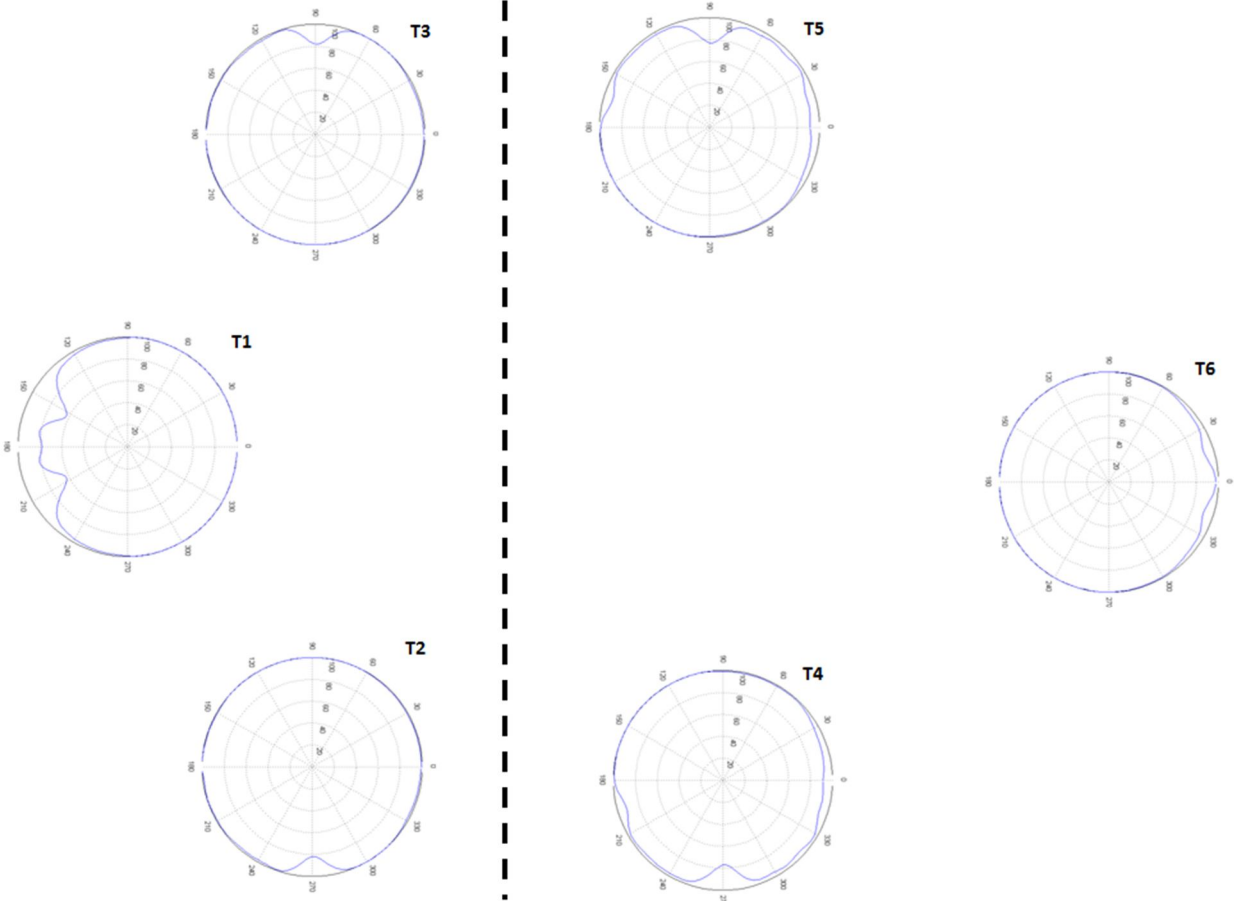
Figura 5. 2 – Planta original da embarcação.



As curvas de eficiência de cada propulsor são apresentadas na

Figura 5. 3. Note que o posicionamento das curvas de eficiência é similar ao posicionamento dos propulsores no casco da embarcação, portanto é possível relacionar os fenômenos de interferência com a posição dos propulsores. Note que o *cluster* de propulsores na popa esta distante do *cluster* de propulsores na proa.

Figura 5. 3 – Curvas de eficiência dos propulsores.



5.2 – *Time Domain Simulation* e Limitações Físicas

Com o objetivo de simular uma operação DP, introduz-se o conceito de *time domain simulation* e limitações físicas.

A simulação das operações DP é realizada da seguinte maneira:

- a) Dados de entrada: Um conjunto de forças requeridas por um sistema de controle, considerando-se intervalos temporais discretos.
- b) Em cada passo uma alocação de empuxo é calculada levando-se em conta como condição inicial a solução do passo anterior.

Dessa maneira a simulação ocorre no domínio do tempo (*Time Domain Simulation*).

Adiante, a fim de aproximar o modelo de simulação da realidade, as limitações físicas dos propulsores devem ser consideradas. As mesmas referem-se à variação máxima do ângulo de azimute do propulsor num determinado espaço de tempo, ou seja, a velocidade angular máxima do mesmo; E a máxima variação de RPM dos propulsores, o que limita a variação do empuxo desenvolvido.

O cálculo da limitação física dos propulsores em cada passo depende do estado anterior dos mesmos. As Eq. 5.1 a 5.3 ilustram o como é determinada a limitação de variação de ângulo de azimute dos propulsores. Note que t_{step} é o intervalo temporal de cada passo, Lb é o limite inferior e Ub é o limite superior.

$$\Delta\alpha_{max} = \dot{\alpha}_{max} \cdot t_{step} \quad (5.1)$$

$$Lb = \alpha_{prev} - \Delta\alpha_{max} \quad (5.2)$$

$$Ub = \alpha_{prev} + \Delta\alpha_{max} \quad (5.3)$$

Para o calculo da máxima variação do empuxo inicialmente determina-se sua rotação atual, conforme a Eq. 5.4. Note que C_{T0} representa o coeficiente de empuxo do propulsor.

$$n_{current} = \sqrt{\frac{T}{C_{T0} \cdot \rho \cdot D^4}} \quad (5.4)$$

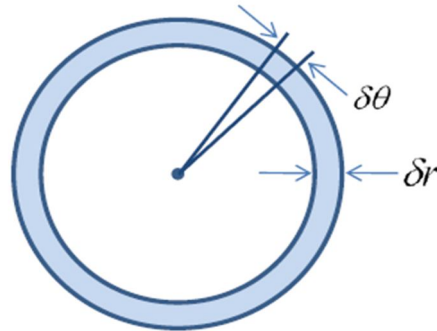
Em seguida aplica-se a limitação física relativa à máxima variação de RPM (\dot{n}_{max}), e determinam-se os limites superiores (Ub) e inferiores (Lb) da variação de empuxo.

$$Lb = C_{T0} \cdot \rho \cdot D^4 \cdot (n_{current} - \dot{n}_{max})^2 \quad (5.5)$$

$$Ub = C_{T0} \cdot \rho \cdot D^4 \cdot (n_{current} + \dot{n}_{max})^2 \quad (5.6)$$

A Figura 5. 4 ilustra graficamente a atuação das limitações físicas dos propulsores.

Figura 5. 4 – Representação das limitações físicas completas dos propulsores.



Finalmente, a implementação das limitações físicas no algoritmo é realizada de maneira simples. Manipula-se os limites moveis de alocação de maneira que os mesmos respeitem a saturação dos propulsores, sua limitada variação de RPM e ângulo de azimuth.

5.3 – Simulações

Essa seção contém 3 simulações de operações DP para a embarcação apresentada. Note que todas as simulações consideram os fenômenos de interação hidrodinâmicos (representados pelas curvas de eficiência), as limitações físicas dos propulsores e são realizados no domínio do tempo (*Time domain simulation*). Cada caso será apresentado, em seguida os resultados são expostos, e imediatamente discutidos.

Inicialmente um caso base é apresentado, a fim de ilustrar o comportamento da embarcação em condições ambientais reais. Em seguida, casos específicos de situações complexas são descritos e discutidos a fim de ilustrar a robustez do algoritmo de alocação de empuxo desenvolvido para operações de posicionamento dinâmico.

5.3.1 – Caso Base – Condições ambientais reais

5.3.1.1 – Condições

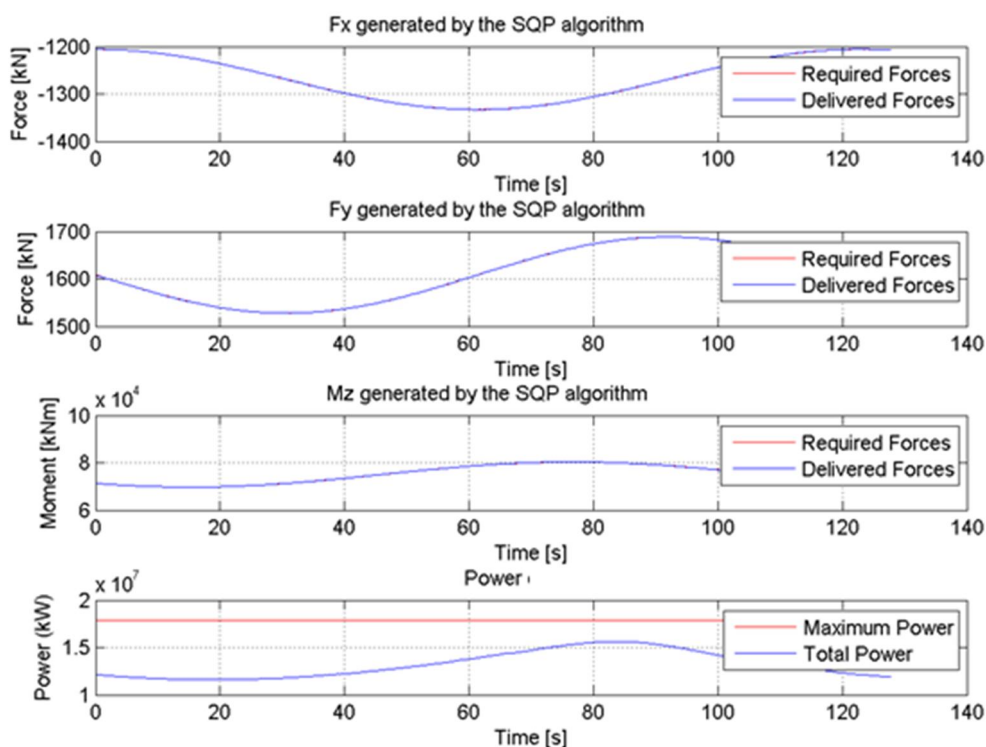
As forças requeridas pelo sistema de controle são baseadas em uma simulação de operação de perfuração realizada com o modelo da embarcação, considerando-se condições ambientais típicas.

5.3.1.2 – Discussão dos Resultados

A Figura 5. 5 apresenta as forças requeridas e geradas pela embarcação para condições ambientais reais. Fica claro que o algoritmo de alocação de empuxo conseguiu gerar as forças requeridas pelo sistema de controle (a linha azul esta sobre a linha vermelha).

Além disso, note que a curva de potencia tem formato bastante semelhante à curva de momento, portanto conclui-se que o momento é a direção que requer maior atenção do sistema de geração de forças. Adiante, note que a embarcação ainda possui alguma reserva de potencia que poderia ser utilizada caso fosse necessário.

Figura 5. 5 – Força requerida e gerada para o caso base.



As figuras a seguir ilustram o erro percentual na geração de forças nas direções longitudinal, lateral e momento. Como esperado (Figura 5. 5), os erros são insignificantes.

Figura 5. 6 – Erro percentual na direção de surge.

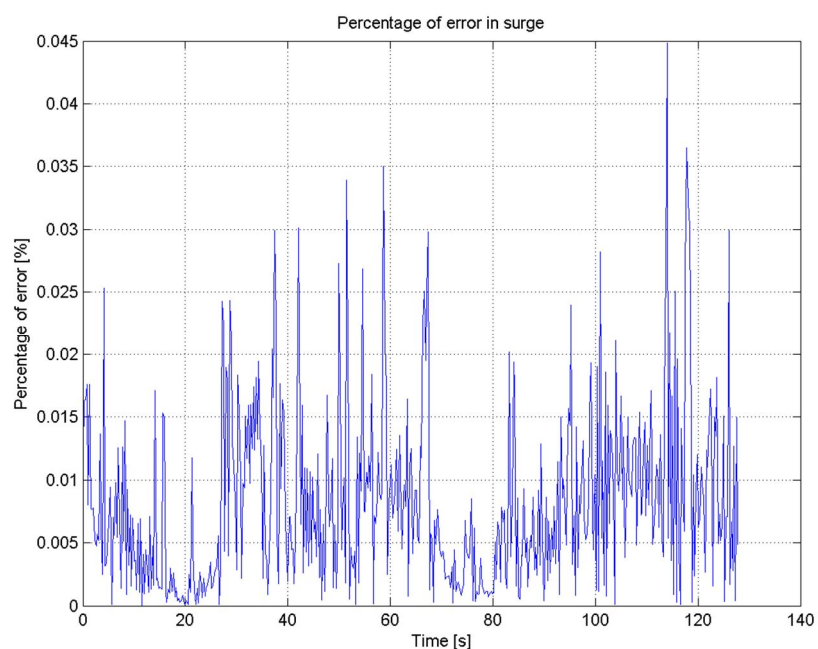


Figura 5. 7 – Erro percentual na direção de sway.

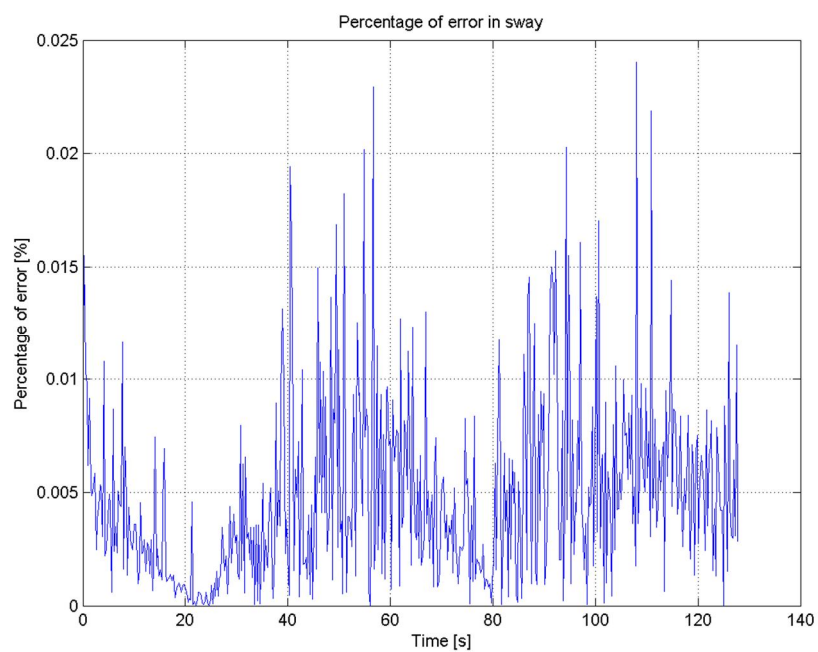
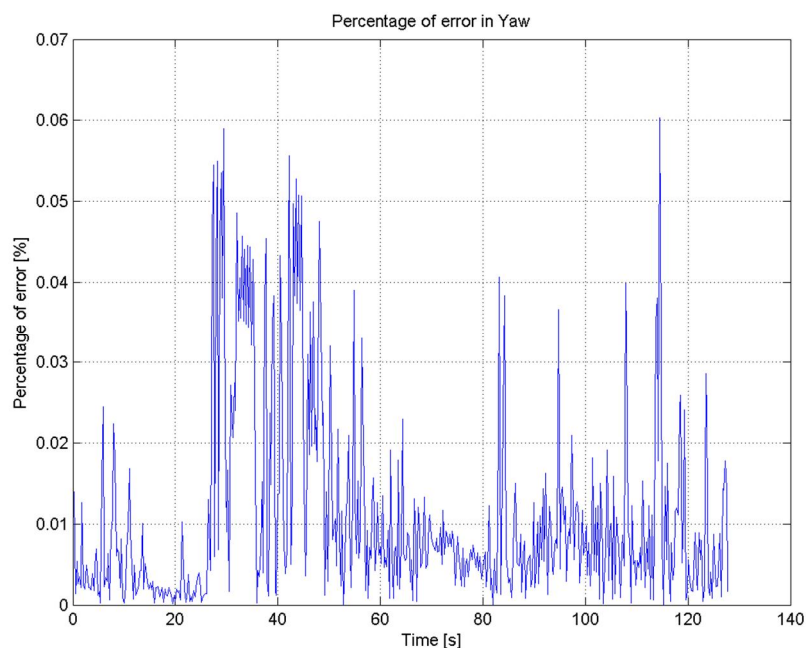


Figura 5. 8 – Erro percentual na direção de yaw.



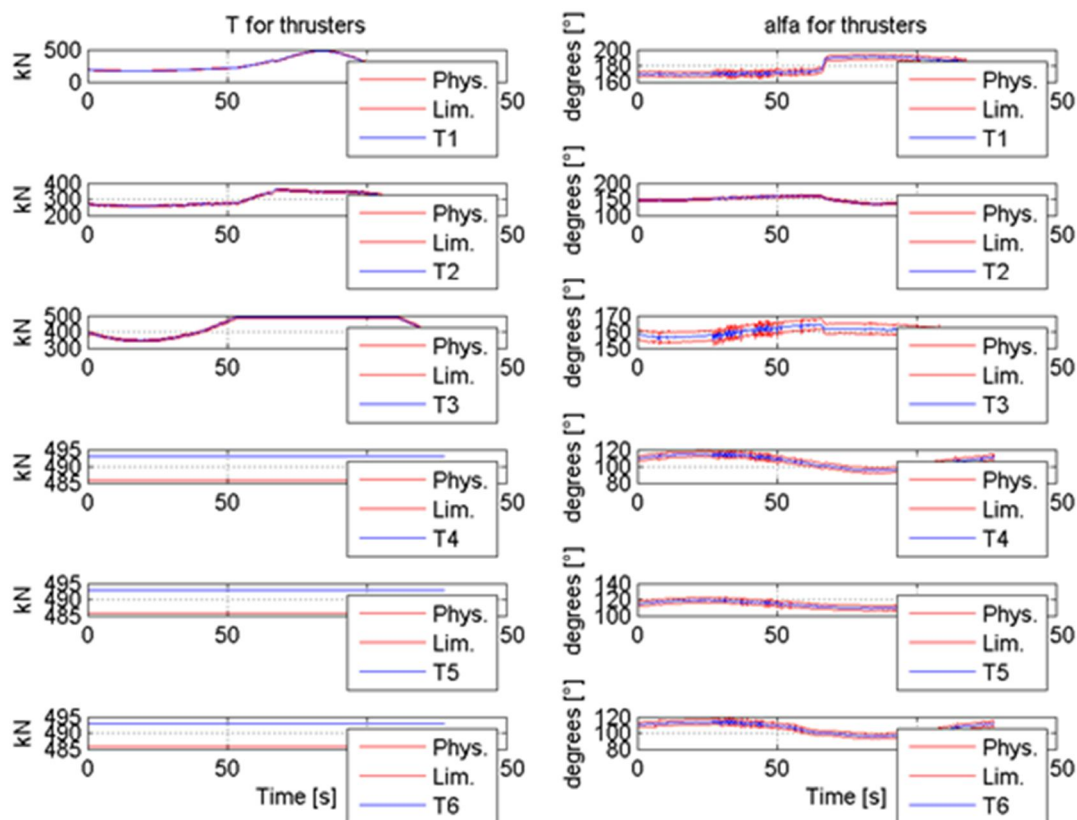
A Figura 5. 9 apresenta a alocação dos propulsores durante a simulação do caso base. Note que os propulsores T4, T5 e T6 estão saturados durante todo o intervalo de simulação. Isso ocorre, pois os mesmos são os principais responsáveis pela geração de momento. Além disso, conclui-se que este cenário é quase crítico, mas mesmo assim o algoritmo de alocação de empuxo conseguiu gerar as forças requeridas pelo sistema de controle.

Note que os ângulos de azimuth apresentam movimentos (em baixa frequência) bastante similares à curva de momento. Esse resultado é esperado, pois os propulsores devem seguir a força crítica aplicada sobre a embarcação.

O propulsor T1 é uma exceção, pois o mesmo encontra-se em uma zona de baixa eficiência (Figura 5. 3) e seu comportamento ilustra que este tenta ficar em ângulos de azimuth que proporcionem maior eficiência. Além disso, por volta de 60 s, T1 vai o mais rápido possível (considerando-se as limitações físicas) para um ângulo de aproximadamente 190° . Essa alocação não contribui para a geração da força lateral requerida, mas contribui com a geração de momento, o que ilustra novamente a importância dessa direção para esta situação.

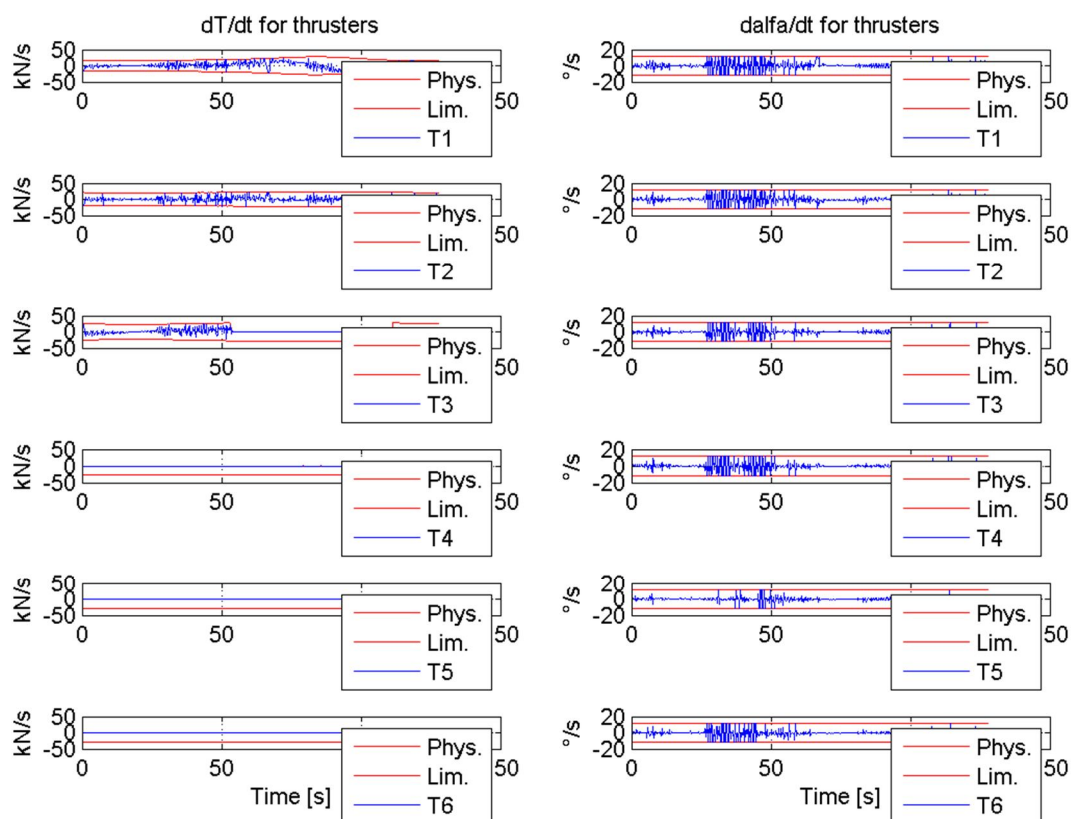
Finalmente, note que as curvas vermelhas ilustram as limitações físicas dos propulsores, que são calculadas com base na saturação dos mesmos, máxima variação de RPM e máxima variação de ângulo de azimute.

Figura 5. 9 – Alocação dos propulsores (empuxo e ângulo de azimute) para o caso base.



A Figura 5. 10 apresenta a variação de empuxo e ângulo de azimute verificadas na alocação de empuxo para o caso base. Note que as linhas vermelhas representam a limitação física dos propulsores e estão de acordo com a Figura 5. 9. Além disso, verifica-se que não é possível aumentar o empuxo dos propulsores T4, T5 e T6 durante todo o intervalo de simulação, pois os mesmos já estão saturados.

Figura 5. 10 – Variação de empuxo e ângulo de azimute dos propulsores para o caso base.



5.3.1.3 – Conclusão

A partir do desenvolvimento do algoritmo de alocação de empuxo foi possível simular uma operação DP baseada em dados ambientais reais, considerando as limitações físicas dos propulsores e os fenômenos de interação entre propulsores e entre propulsor e casco. Embora as condições fossem críticas para a embarcação (T4, T5 e T6 estão saturados durante todo o intervalo de simulação), as forças requeridas pelo sistema de controle foram geradas com elevada precisão.

Ainda, nota-se que os propulsores evitam as zonas de menor eficiência, que correspondem aos ângulos de azimute em que há interação entre propulsores, portanto conclui-se que não é necessário definir zonas proibidas para os propulsores, pois as mesmas são evitadas automaticamente porque apresentam baixa eficiência.

5.3.2 – Caso Desliga/Liga

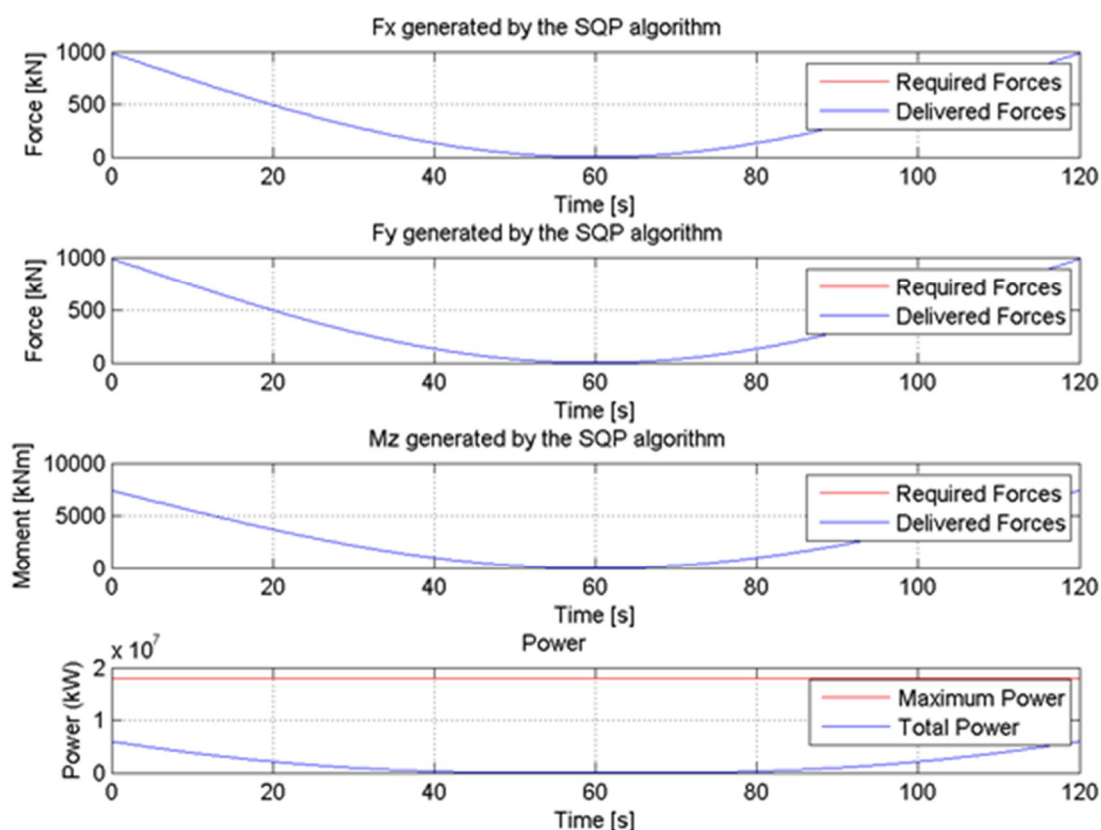
5.3.2.1 – Condições

Neste caso estuda-se o comportamento do algoritmo de alocação de empuxo para a situação em que os propulsores da embarcação são desligados, e em seguida ligados. Ou seja, as forças requeridas pelo sistema de controle decaem até valores nulos e em seguida aumentam.

5.3.2.2 – Discussão dos Resultados

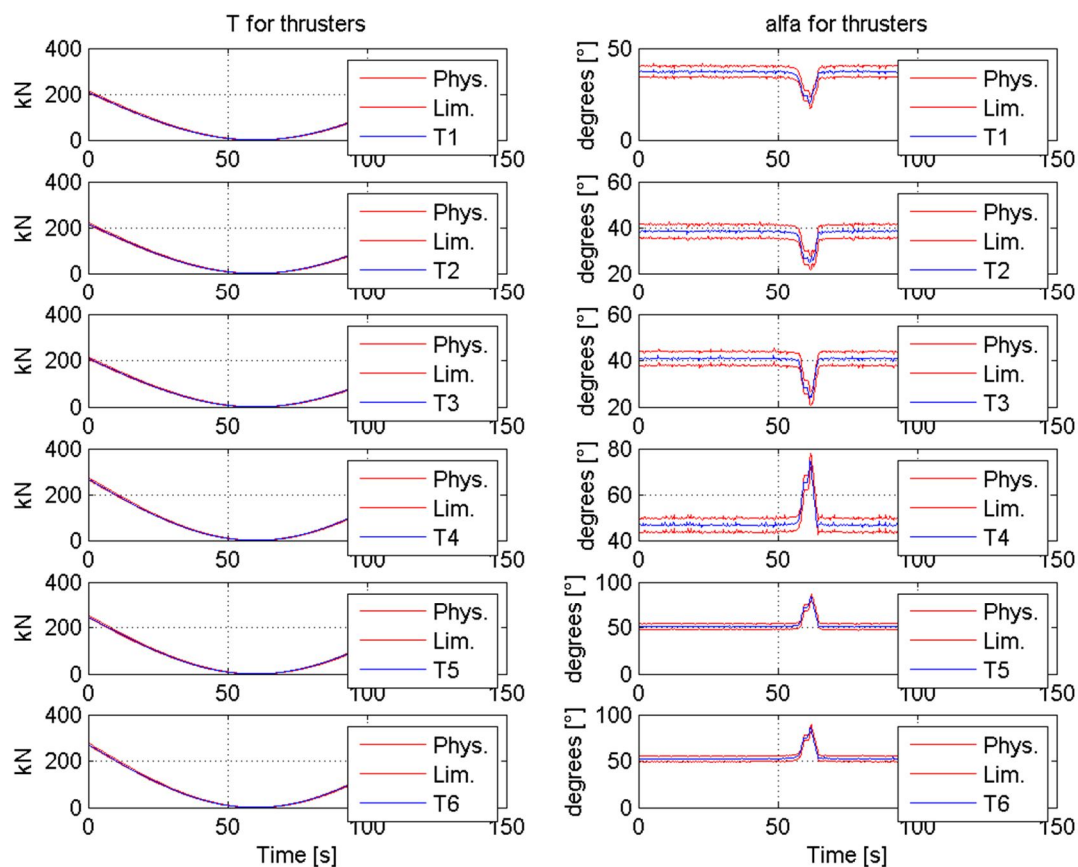
A Figura 5. 11 apresenta as forças requeridas e geradas para o caso em que as mesmas são inicializadas em um valor elevado, diminuem até Zero, e em seguida, aumentam até o valor inicial. Note que durante todo o intervalo de simulação o algoritmo de alocação de empuxo conseguiu gerar as forças requeridas.

Figura 5. 11 – Força requerida e gerada para o caso Desliga/Liga.



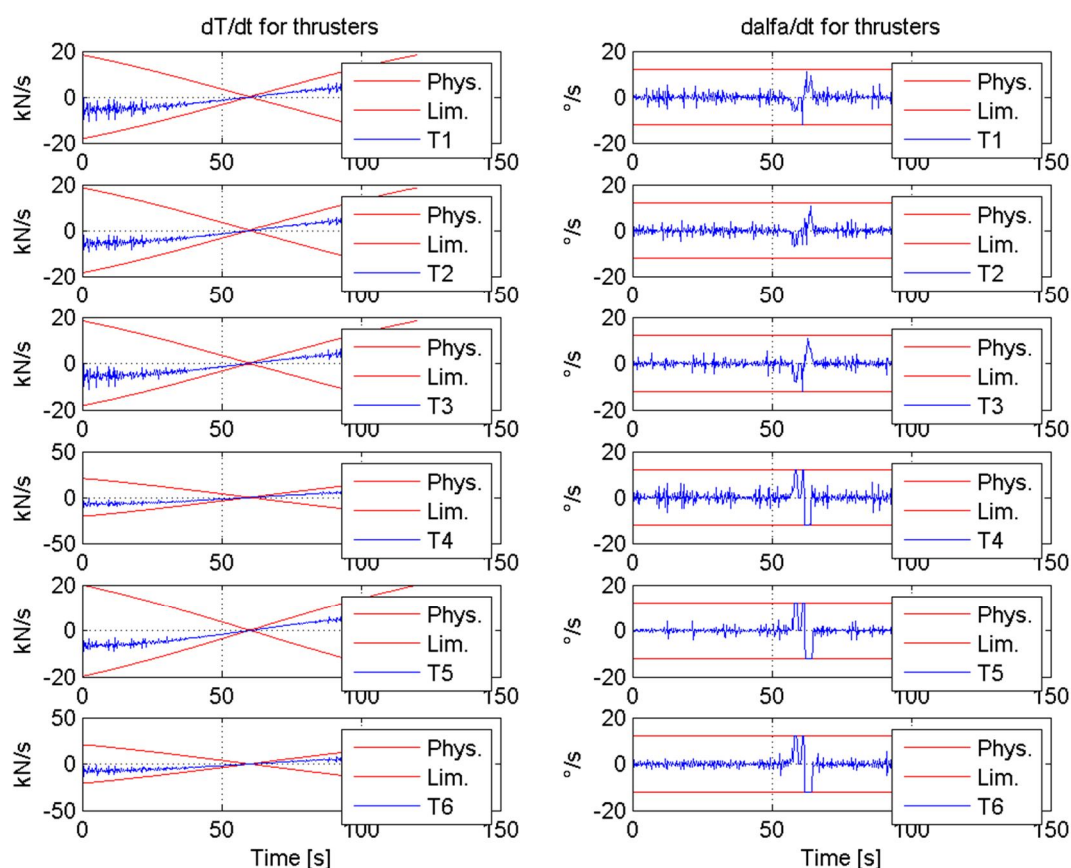
A Figura 5. 12 apresenta a alocação de empuxo para o caso Desliga/Liga. Note que o empuxo gerado por cada propulsor tem a mesma forma das forças requeridas, o que é um comportamento esperado.

Figura 5. 12 – Alocação para o caso Desliga/Liga.



Neste caso é interessante estudar a variação das variáveis, ilustrado na Figura 5. 13. Note que conforme o empuxo diminui a sua variação máxima também diminui. Isso pode ser explicado conforme as Eq.s 5.5 e 5.6, pois o empuxo depende da rotação do propulsor elevada ao quadrado; portanto se a rotação diminui, a mesma variação em RPM, leva a um intervalo menor de geração de empuxo. Desta forma, explica-se a forma da curva de limitação física do empuxo dos propulsores.

Figura 5. 13 – Variação de empuxo e ângulo de azimuth dos propulsores para o caso Desliga/Liga.



5.3.2.3 – Conclusão

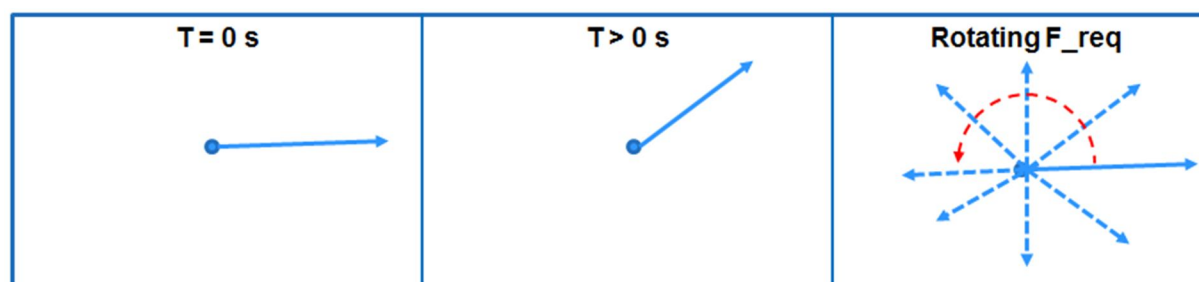
Essa simulação ilustrou aspectos interessantes relativos às limitações físicas dos propulsores. Além disso, o algoritmo de alocação de empuxo lidou de maneira correta com o fenômeno Desliga/Liga simulado, e gerou com precisão as forças requeridas pelo sistema de controle.

5.3.3 – Caso de Forças Requeridas Girando – T3 e T5 Cruzam “Zonas Proibidas”

5.3.3.1 – Condições

As forças requeridas neste caso não se baseiam em simulações com o modelo. As mesmas estão girando conforme ilustrado na Figura 5. 14. Essa situação pode ocorrer durante manobras da embarcação.

Figura 5. 14 – Forças requeridas girando.

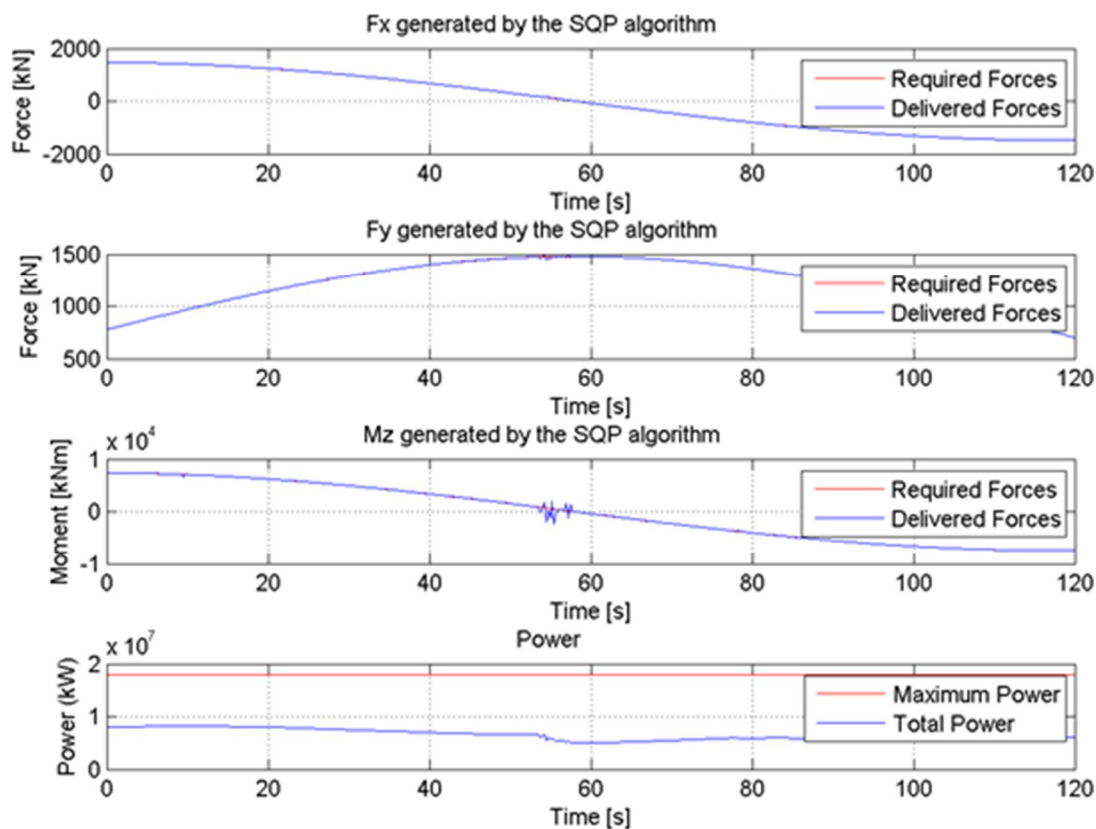


O objetivo desta simulação é verificar se os propulsores cruzarão as zonas de baixa eficiência (onde há interação entre propulsores), e que normalmente são definidas como “zonas proibidas”. Além disso, verifica-se de que maneira ocorre essa passagem.

5.3.3.2 – Discussão dos Resultados

A Figura 5. 15 apresenta as forças geradas para o caso em que as forças requeridas estão girando. Verifica-se que as forças geradas atendem as forças requeridas durante todo o intervalo de simulação, porém vale a pena olhar com mais detalhes como isso ocorre.

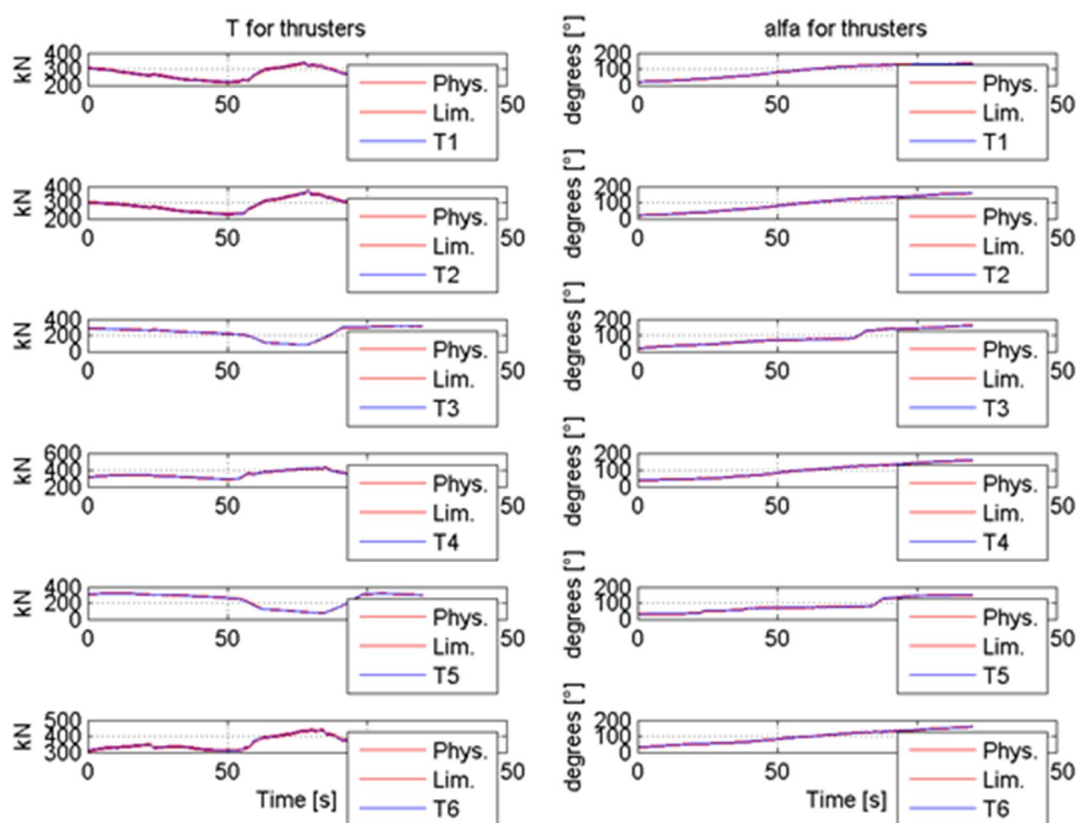
Figura 5. 15 – Força requerida e gerada para o caso Forças requeridas girando.



A Figura 5. 16 apresenta as alocações para esse caso. Há 4 eventos que merecem destaque e que ilustram como os propulsores T3 e T5 cruzam as zonas de baixa eficiência.

- 1) Por volta de 55 s, T3 e T5 começam a diminuir seus respectivos empuxos. Em contrapartida, os outros propulsores aumentam seus empuxos para compensar o comportamento de T3 e T5.
- 2) A diminuição de empuxo de T3 e T5 ocorre, pois T3 e T5 estão parados na entrada da chamada “zona proibida” e estão se distanciando da direção da força requerida. Os outros propulsores seguem a força requerida.
- 3) Por volta de 75 s, T3 e T5 cruzam a zona onde apresentam baixa eficiência, com pequenos valores de empuxo e o mais rápido possível (considerando-se suas limitações físicas), o que resulta em economia de potencia, pois os mesmos encontram-se em zona de baixa eficiência. Além disso, ao cruzarem essa zona de baixa eficiência que representa a interação entre propulsores com pequenos valores de empuxo e de maneira rápida, o jato d’água jogado contra os outros propulsores não é muito significativo.
- 4) Após cruzarem as zonas de baixa eficiência T3 e T5, “correm” atrás da direção da força requerida, aumentando seus respectivos empuxos, pois os mesmos apresentam boa eficiência e direção cada vez mais próxima à da força requerida. Em contrapartida, os outros propulsores diminuem seus respectivos empuxos a fim de minimizar o consumo de energia da embarcação.

Figura 5. 16 – Alocação para o caso Forças requeridas girando.



5.3.3.3 – Conclusão

Essa simulação ilustra o processo do cruzamento de zonas de baixa eficiência, que em outros algoritmos são tratadas como zonas proibidas. Verifica-se como este processo é realizado de maneira automática pelo algoritmo de alocação de empuxo e de maneira otimizada.

- 1) Parada na entrada da zona de baixa eficiência.
- 2) Diminuição do empuxo.
- 3) Cruzamento o mais rápido possível.
- 4) Corrida atrás das forças requeridas

Esse processo é bastante eficiente, pois ao diminuir o empuxo gerado antes de entrar na zona de baixa eficiência, a alocação requer menor consumo de energia, e o jato d'água jogado sobre os outros propulsores não é muito significativo.

6 – Conclusão

Os resultados da implementação inicial do Algoritmo de Alocação de Empuxo permitem concluir que os fenômenos de interferência são importantes e devem ser considerados no Sistema DP.

O estudo da dinâmica do algoritmo inicial permitiu conhecer e prever seu funcionamento, bem como desenvolver técnicas de otimização simplificadoras. Essas diferentes técnicas de otimização foram utilizadas para desenvolver novos algoritmos de Alocação de Empuxo.

Os novos algoritmos foram testados, comparados e analisados, quanto à sua Eficiência, Precisão, Tempo de processamento e Dinâmica intrínseca. Por fim selecionou-se o algoritmo Trust Region SQP, que em nível superior é da classe *Trust Region* e em nível inferior (método dos gradientes reduzidos para programação quadrática) é da classe *Line Search*.

O Trust Region SQP respeita os requisitos essenciais de Precisão e Tempo de Processamento, e é mais eficiente do que todos os algoritmos simulados, incluindo o FMINCON do MATLAB.

Além disso, todas as comparações com os algoritmos de alocação de Empuxo disponíveis no mercado continuam válidas:

Alocação Objetiva

- Consegue resolver o problema para qualquer fenômeno de Interferência

Sistema Confiável e seguro

- Inclusão da Interferência torna o sistema mais próximo da realidade
- Poupa propulsores avariados

Ganhos Econômicos

- Economia Combustível
- Aumento Janela de Operação

Por fim, toda a arquitetura do algoritmo de alocação de empuxo foi configurada para o desenvolvimento de um simulador de operações DP. Note que as simulações realizadas incluem as limitações físicas dos propulsores e ocorrem no domínio do tempo (*time domain simulations*), além de considerar os fenômenos de interação.

A simulação em condições ambientais reais ilustra que o algoritmo de alocação de empuxo é confiável para operações DP. Os erros na geração de força foram insignificantes e a embarcação conseguiu manter sua posição, levando em conta suas limitações físicas, saturação dos propulsores, e os fenômenos de interação.

O caso em que 2 propulsores cruzaram suas respectivas zonas de baixa eficiência (normalmente tratadas como “zonas proibidas”), mostra que o algoritmo de alocação de empuxo automaticamente evita tais zonas, até que valha a pena cruzá-las. Além disso, o cruzamento dessas zonas de baixa eficiência é realizado com baixo empuxo e com a maior velocidade possível, o que economiza energia e previne que um propulsor jogue um jato d’água significativo sobre um segundo propulsor.

Portanto os resultados das simulações de operações DP, mostram que o algoritmo de alocação de empuxo desenvolvido pode lidar com fenômenos de interação hidrodinâmicos, considerando as limitações físicas dos propulsores. Espera-se que essas melhoras na estratégia de alocação de empuxo levem ao aumento da precisão de operações DP e *stationkeeping* de embarcações.

Não obstante, note que o algoritmo de alocação de empuxo foi utilizado em 2 embarcações e em diversas situações complexas. Portanto o mesmo mostra-se uma alternativa robusta.

Algumas melhorias e ideias que podem ser desenvolvidas em trabalhos futuros nessa área são:

- 1) Melhorar o tratamento de saturação dos propulsores. Há diversas estratégias disponíveis na bibliografia que podem ser implementadas e comparadas. Destacam-se os seguintes métodos: dicotomia, loops de realocação e definição de um novo problema de otimização que lida apenas com minimizar os erros na geração de forças.
- 2) Incluir fenômenos de interação hidrodinâmicos mais complexos como, a interação entre propulsor e correnteza.
- 3) Realizar simulações DP em tanques de prova.

Referências Bibliográficas

ARDITTI, F.; TANNURI, E. A. **Thrust Allocation Algorithm for DP Systems considering the Interference between thrusters and thrust-hull.** In COBEM 2011. Natal, Brazil. 2011.

ARDITTI, F.; TANNURI, E. A. **Experimental Analysis of a Thrust Allocation Algorithm for DP Systems Considering the Interference between Thrusters and Thruster-Hull.** MCMC 2012.

COZIIN, J.L., HALLMANN, R. **The wake flow behind azimuthing thrusters: Measurements in open water, under a plate and under a barge.** In Proceedings of the ASME 2012. 31st International Conference on Ocean, Offshore and Arctic Engineering OMAE2012. Rio de Janeiro, Brazil.

DANG, J., LAHEIJ, H. **Hydrodynamic Aspects of Steerable Thrusters.** Dynamic Positioning Conference, Marine Technology Society.

DE WIT, C. **Optimal Thrust Allocation Methods for Dynamic Positioning of Ships,** Holland. Dissertação (mestrado) – Delft Institute of Applied Mathematics, Delft University of Technology. 2009.

EKSTROM, L.; BROWN, D. T. **Interactions Between Thrusters Attached to a Vessel Hull.** In: Proceedings of OMAE 02, 21st International Conference on Offshore Mechanics and Arctic Engineering, Oslo, Norway. 2002.

JENSSEN, N. A., REALFSEN, B., **Power Optimal Thruster Allocation.** Dynamic Positioning Conference. Marine Technology Society.

LEHN, E. **Thruster Interaction Effects,** NSFI, Marintek, R102-80, Trondheim, Norway. 1980.

MOBERG, S.; HELLSTRÖM, S.A., **Dynamic Positioning of a four column semi-submersible. Model testes of interaction forces and a philosophy about optimum strategy when operating thrusters,** In Proceedings of second International Symposium on Ocean Engineering and Ship Handling, pp 443-480. 1983.

NOCEDAL, J.; WRIGHT S. J., **Numerical Optimization.** Second Edition. Ed. Springer. Estados Unidos. 2006.

SILVA, E. C. N. **Otimização Aplicada ao Projeto de Sistemas Mecânicos**. Apostila do curso de Otimização Aplicada ao Projeto de Sistemas Mecânicos (PMR 5215), oferecido no curso de Engenharia Mecatrônica da Escola Politécnica da USP. Apostila disponível no site: <http://sites.poli.usp.br/d/pmr5215/notas.html>. Consultada em 2011.

SKJETNE, R. and KJERSTAD, O. K. **Recursive nullspace-based control allocation with strict prioritization for marine craft**. Japão 2013.

SORDALEN, O. J. **Optimal thrust allocation for marine vessels**, Control Engineering Practice, vol 5 (9), pp 1223-1231.

Sorensen, A. J. **A survey of dynamic positioning control systems**. Annual Reviews in control, vol 35 (1), pp 123-136.

TANNURI, E.A. **Desenvolvimento de Metodologia de Projeto de Sistema de Posicionamento Dinâmico Aplicado a Operações em Alto-Mar**, São Paulo. 1v. Dissertação (doutorado) – Escola Politécnica, Universidade de São Paulo. 2002.

THRUST JIP. Informações sobre o projeto disponíveis em: <http://www.marin.nl/web/JIPs-Networks/Public/TRUST.htm>. Consultado em 20/06/2012.

VAN DAALEN, E. F. G. COZIEN, J. L. LOUSSOUARN, C., HEMKER, P. W. **A generic optimization algorithm for the allocation of DP**. 2011. In: Proceedings of the ASME 2011 30th International Conference on Ocean, Offshore and Arctic Engineering OMAE. Rotterdam, The Netherlands.

Anexo A – Solução Analítica da Alocação sem Interferência e função objetivo quadrática

Esse anexo tem por objetivo ilustrar a solução analítica ao problema de otimização da Alocação com Função Objetivo Quadrática. Essa solução é o chute inicial do problema de otimização completo (com Função Objetivo ajustada e curvas de interferência).

Como ilustrado na seção 3.3.1, os propulsores azimutais podem ser considerados como a união de um propulsor principal a um propulsor em túnel. Dessa forma trabalha-se considerando forças com índices ímpares como propulsores principais ou parte de propulsores azimutais e; forças com índices pares como propulsores em túnel ou complementos de propulsores azimutais.

Considera-se A função objetivo é de forma quadrática (Equação A.1).

$$\text{Obj} = F_1^2 + F_2^2 + F_3^2 + F_4^2 + \dots + F_n^2 \quad (\text{A.1})$$

As restrições ao problema de alocação são a equivalência entre as forças geradas pelos propulsores às cargas ambientais ($FSurge$; $Fsway$; $MYaw$).

$$\begin{cases} h_1 : F_1 + F_3 + F_5 + \dots + F_{n-1} - F_{surge} = 0 \\ h_2 : F_2 + F_4 + F_6 + \dots + F_n - F_{sway} = 0 \\ h_3 : F_1 \cdot (-y_1) + F_2 \cdot x_2 + F_3 \cdot (-y_3) + F_4 \cdot x_4 + F_{n-1} \cdot (-y_n) + F_n \cdot x_n - Myaw = 0 \end{cases} \quad (\text{A.2})$$

Note que a Equação A.2 é equivalente à equação A.3:

$$[A] \times [F_i] = \begin{bmatrix} F_{surge} \\ F_{sway} \\ Myaw \end{bmatrix} \quad (\text{A.3})$$

Aplica-se o método de Lagrange, descrito em Silva (2011). Onde se define a função Lagrangeana (L), que contém a Função Objetivo mais as restrições multiplicadas pelas

variáveis λ_i . Essas variáveis são conhecidas como variáveis de relaxação e facilitam a solução, pois o problema torna-se uma otimização sem restrição:

$$L = \text{Obj} + \lambda_1 \cdot h_1 + \lambda_2 \cdot h_2 + \lambda_3 \cdot h_3 \quad (\text{A.4})$$

Para obter-se o valor da Função Objetivo (A.1) é necessário obter:

$$\left\{ \frac{\partial L}{\partial F_i} = 0 = \frac{\partial L}{\partial \lambda_i} \right. \quad (\text{A.5})$$

Note que $\frac{\partial L}{\partial \lambda_i}$ recupera as restrições do problema.

Aplicando a metodologia descrita na Equação A.5 obtêm-se:

$$\left\{ \begin{array}{l} \text{Parte superior} \\ 2 \cdot F_1 + \lambda_1 - y_1 \cdot \lambda_3 = 0 \\ 2 \cdot F_2 + \lambda_2 + x_2 \cdot \lambda_3 = 0 \\ 2 \cdot F_3 + \lambda_1 - y_3 \cdot \lambda_3 = 0 \\ 2 \cdot F_4 + \lambda_2 + x_4 \cdot \lambda_3 = 0 \\ \dots \\ 2 \cdot F_{n-1} + \lambda_1 - y_{n-1} \cdot \lambda_3 = 0 \\ 2 \cdot F_n + \lambda_2 + x_n \cdot \lambda_3 = 0 \\ \text{Parte inferior} \\ [A] \times [F_i] = \begin{bmatrix} \text{Fsurge} \\ \text{Fsway} \\ \text{Myaw} \end{bmatrix} \end{array} \right. \quad (\text{A.6})$$

Para tornar a solução mais clara, daqui a diante considera-se apenas dois propulsores azimutais (F_1, F_2, F_3, F_4) , que é o número mínimo que permite otimização, pois desta maneira, o sistema de Equações A.3 tem infinitas soluções. Note que o processo de solução será análogo para demais propulsores. Dessa forma obtém-se:

$$\begin{cases}
 \text{Superior} \\
 2 \cdot F_1 + \lambda_1 - y_1 \cdot \lambda_3 = 0 \\
 2 \cdot F_2 + \lambda_2 + x_2 \cdot \lambda_3 = 0 \\
 2 \cdot F_3 + \lambda_1 - y_3 \cdot \lambda_3 = 0 \\
 2 \cdot F_4 + \lambda_2 + x_4 \cdot \lambda_3 = 0 \\
 \text{Inferior} \\
 F_1 + F_3 - F_{\text{surge}} = 0 \\
 F_2 + F_4 - F_{\text{sway}} = 0 \\
 F_1 \cdot (-y_1) + F_2 \cdot x_2 + F_3 \cdot (-y_3) + F_4 \cdot x_4 - Myaw = 0
 \end{cases} \quad (A.7)$$

Considere inicialmente as equações $\frac{\partial L}{\partial F_i} = 0$, na parte superior do sistema de Equações A.7. Veja como é fácil eliminar λ_1 e λ_2 , basta subtrair uma equação de índice ímpar de outra, e realizar o mesmo processo entre as equações de índices pares.

$$\begin{cases}
 2F_1 - 2F_3 + \lambda_3 \cdot y_3 - \lambda_3 \cdot y_1 = 0 \\
 2F_2 - 2F_4 + \lambda_3 \cdot x_2 - \lambda_3 \cdot x_4 = 0 \\
 F_1 + F_3 - F_{\text{surge}} = 0 \\
 F_2 + F_4 - F_{\text{sway}} = 0 \\
 F_1 \cdot (-y_1) + F_2 \cdot x_2 + F_3 \cdot (-y_3) + F_4 \cdot x_4 - Myaw = 0
 \end{cases} \quad (A.8)$$

Também é possível eliminar λ_3 combinando as duas primeiras equações:

$$\begin{cases} 2F_2 \cdot (y_3 - y_1) + 2F_4 \cdot (y_1 - y_3) + 2F_3 - 2F_1 = 0 \\ F_1 + F_3 - F_{\text{surge}} = 0 \\ F_2 + F_4 - F_{\text{sway}} = 0 \\ F_1 \cdot (-y_1) + F_2 \cdot x_2 + F_3 \cdot (-y_3) + F_4 \cdot x_4 - \text{Myaw} = 0 \end{cases} \quad (\text{A.9})$$

Escrevendo A.9 na forma matricial:

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -y_1 & x_2 & -y_3 & x_4 \\ -1 & y_3 - y_1 & 1 & y_1 - y_3 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} F_{\text{surge}} \\ F_{\text{sway}} \\ \text{Myaw} \\ 0 \end{bmatrix} \quad (\text{A.10})$$

Para resolver o sistema basta inverter a primeira matriz:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \text{pinv}(A) \end{bmatrix} & \begin{matrix} \text{don't care} \\ \text{don't care} \\ \text{don't care} \\ \text{don't care} \end{matrix} \end{bmatrix} \begin{bmatrix} F_{\text{surge}} \\ F_{\text{sway}} \\ \text{Myaw} \\ 0 \end{bmatrix} \quad (\text{A.11})$$

Desta forma demonstra-se a melhor alocação para Função Objetivo quadrada é multiplicar a matriz pseudo-inversa de A pelas forças do meio ambiente. Note que a última coluna não tem relevância.

Note que a matriz pseudo-inversa de A também é obtida segundo a Equação (A.12)

$$\text{pinv}(A) = A^+ = A^T (AA^T)^{-1} \quad (\text{A.12})$$

Anexo B – Tabela de comparação dos algoritmos de Alocação de Empuxo

Apresentam-se as simulações de diferentes Forças de Controle (F_x , F_y , M_z), as Forças Efetivas (F_x , F_y , M_z) geradas por cada um dos algoritmos, seu Tempo de processamento, o valor da Função Objetivo (Potência) na solução e o Erro Percentual ao atender as Forças de Controle.

Tabela B. 1: Simulação completa dos Algoritmos de Alocação de Empuxo

F controle			FMINCON			SQP - Potência Quadrática								
F_x	F_y	M_z	F_x	F_y	M_z	Tempo	Potência	Erro de Força	F_x	F_y	M_z	Tempo	Potência	Erro de Força
7	7	0	7,000	7,000	0,000	2,819	15,8124	0 %	7,003	6,997	0,013	1,137	15,8721	0,14 %
-7	7	0	-7,000	7,000	0,000	3,132	19,945	0 %	-6,983	7,011	0,001	0,911	19,8239	0,21 %
-7	-7	0	-7,000	-7,000	0,000	3,367	19,0804	0 %	-6,909	-7,057	-0,031	1,078	19,0529	1,28 %
7	-7	0	7,000	-7,000	0,000	2,730	18,8584	0 %	7,006	-6,865	-0,124	0,762	19,8251	1,89 %
-10	0	0	-10,000	0,000	0,000	3,139	19,8517	0 %	-9,992	0,006	-0,012	1,012	19,9786	0,26 %
0	10	0	0,000	10,000	0,000	3,994	16,9581	0 %	0,013	9,999	0,001	0,952	17,0148	0,15 %
0	-10	0	0,000	-10,000	0,000	3,389	20,2653	0 %	0,090	-9,953	0,051	0,971	20,0224	1,88 %

SQP Potência Quadrática + Passo 2						Trust Region SQP					
F_x	F_y	M_z	Tempo	Potência	Erro de Força	F_x	F_y	M_z	Tempo	Potência	Erro de Força
7,000	7,000	0,000	1,334	15,8294	0 %	7,000	7,000	-0,013	1,002	15,8125	0,09 %
-7,000	7,000	0,000	1,042	19,6596	0 %	-7,000	6,982	-0,026	1,286	19,6036	0,31 %
-7,000	-7,000	0,000	1,184	19,1459	0 %	-6,977	-7,002	-0,046	1,362	19,0812	0,51 %
7,000	-7,000	0,000	0,841	20,0313	0 %	6,931	-6,997	-0,020	1,234	19,3467	0,66 %
-10,000	0,000	0,000	1,108	19,8816	0 %	-9,991	0,016	0,030	2,176	19,826	0,55 %
0,000	10,000	0,000	1,048	16,9603	0 %	0,019	10,000	0,006	1,491	16,9549	0,25 %
0,000	-10,000	0,000	1,060	20,2002	0 %	-0,023	-9,957	0,035	1,210	19,9492	1,01 %