

Diego Bellas Colares
Roberto de Oliveira Bezerra

**Desenvolvimento de Sistema de Comando em Tempo Real para
um Veículo Aquático Remotamente Operado (ROV)**

Monografia apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do Título de
Engenheiro

São Paulo
2012

Diego Bellas Colares
Roberto de Oliveira Bezerra

**Desenvolvimento do Sistema de Comando em Tempo Real para
um Veículo Aquático Remotamente Operado (ROV)**

Monografia apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do Título de
Engenheiro

Área de Concentração:

Engenharia Mecatrônica

Orientador:

Prof. Dr. Ettore Apolônio de Barros

São Paulo
2012

FICHA CATALOGRÁFICA

Colares, Diego Bellas

**Desenvolvimento do sistema de comando em tempo real
para um veículo aquático remotamente operado (ROV) / D.B.
Colares, R.O. Bezerra. -- São Paulo, 2012.**

84 p.

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

**1. Veículos (Operação) 2. Tempo-real 3. Simulação I. Bezerra,
Roberto de Oliveira II. Universidade de São Paulo. Escola Poli-
técnica. Departamento de Engenharia Mecatrônica e de
Sistemas Mecânicos III. t.**

AGRADECIMENTOS

Primeiramente gostaríamos de agradecer ao Prof. Dr. Ettore Apolônio de Barros pela oportunidade de nos incorporar nesse grande projeto em desenvolvimento em seu laboratório, o LVNT (Laboratório de Veículos Não Tripulados) e por toda a infraestrutura e ajuda que nos foi proporcionada para o desenvolvimento deste trabalho.

Gostaríamos de agradecer também ao Prof. Dr. Thiago de Castro Martins pelas preciosas contribuições no enriquecimento deste estudo e pelo apoio em diversos assuntos essenciais deste trabalho.

Um agradecimento especial a todos os membros do LVNT, João Lucas Dozzi Dantas, Lucas Machado de Oliveira, Persing Junior Cardenas Vivanco, Rodrigo Telles da Silva Vale, William da Silva Caetano, que nos ajudaram em diversos assuntos e dúvidas, também pela boa amizade e compartilhamento de experiências em temas de âmbito geral.

A nossas famílias e namoradas, cujo apoio em todos os momentos foi inestimável e revigorante. Dedicamos esse trabalho especialmente a todos vocês.

RESUMO

O presente trabalho visa a implementação e análise dos resultados de um sistema de comando em tempo real de um veículo remotamente operado (ROV). Para tanto são apresentados a modelagem do veículo, o desenvolvimento e documentação do simulador em tempo real, desenvolvido para a análise do veículo e todas as análises e resultados obtidos através das técnicas de otimização de alocação de empuxo e controle clássico.

Foi desenvolvido um ambiente de simulação para a análise da resposta de um veículo aquático em malha de controle aberta e posteriormente o mesmo simulador foi alterado para uma malha de controle fechada. Toda a documentação, protocolos e considerações pertinentes foram apresentados neste trabalho, assim como as respostas obtidas para as técnicas utilizadas para alocação de empuxo e sistema de controle tanto em malha aberta, quanto em malha fechada.

ABSTRACT

The present work aims at implementing and analyzing the results of a real time control system of a remotely operated vehicle (ROV). To do so will be presented the modeling of the vehicle, the development and documentation of real-time simulator developed for the analysis of the vehicle and all analyzes and results obtained using allocation of thrust optimization techniques and classic control.

It was developed a simulation environment for analyzing the response of a craft that will have an open control loop that lately was changed to a closed control loop. All documentation, protocols and relevant considerations were presented in this work, as well as the responses to the techniques used for allocation of thrust and control system in both open loop and closed loop.

SUMÁRIO

AGRADECIMENTOS	1
RESUMO	2
ABSTRACT	3
SUMÁRIO	4
LISTA DE FIGURAS	6
LISTA DE TABELAS	8
1. INTRODUÇÃO	9
2. OBJETIVOS	9
2.1. OBJETIVO GERAL	9
2.2. OBJETIVO ESPECÍFICO	10
3. METODOLOGIA	10
3.1. MODELAGEM DO ROV	10
3.1.1. Sistema de Coordenadas	12
3.1.2. Desenvolvimento do equacionamento do modelo	14
3.2. ALOCAÇÃO DE EMPUXO	18
3.2.1. Solução Para a Alocação de Empuxo	20
3.2.2. Relação entre a Velocidade do Veículo e a Propulsão Máxima	25
3.2.3. Transformação do Empuxo para Voltagem	26
4. SIMULADOR DE DINÂMICA DE UM ROV - DOCUMENTAÇÃO	27
4.1. INTRODUÇÃO AO SIMULADOR	27
4.2. OVERVIEW DO SIMULADOR – INTRODUÇÃO AO SOFTWARE	28
4.3. BLOCO MODELAGEM	30
4.3.1. Introdução ao software	30
4.4. BLOCO INTERFACE DE RESULTADOS	38
4.5. BLOCO DE COMUNICAÇÃO	39
4.5.1. Real-Time Windows Target	41
4.5.2. Protocolo de Comunicação ethernet UDP em C++ e em Simulink/MatLab (UDP/IP)	42
4.6. HOMOLOGAÇÃO DO SIMULADOR – RESULTADOS	43
4.6.1. Análise dos Resultados	45
5. PROJETO DO CONTROLADOR	46
5.1. CARACTERÍSTICAS DESEJÁVEIS DO CONTROLADOR PROJETADO	47
5.1.1. Controle robusto	47
5.1.2. Controle em tempo real	47

5.2.	ESTUDO DE CASO – NÃO LINEARIDADES DO PROJETO	48
5.2.1.	<i>Não Linearidades associadas à modelagem de um ROV</i>	<i>48</i>
5.3.	ANÁLISE DA DINÂMICA DO ROV.....	49
5.3.1.	<i>Modelo simplificado para a dinâmica do ROV.....</i>	<i>51</i>
5.3.2.	<i>Testes comparativos com o modelo simplificado</i>	<i>55</i>
5.4.	IMPLEMENTAÇÃO CONTROLE AUTOMÁTICO EM MALHA FECHADA	58
5.4.1.	<i>Sintonia pelo Método da Curva de Reação do Sistema.....</i>	<i>58</i>
5.4.2.	<i>Cálculos dos valores para o método selecionado e validação do método</i>	<i>61</i>
5.4.3.	<i>Controlador PID.....</i>	<i>62</i>
5.4.4.	<i>Análise de robustez do controlador PID.....</i>	<i>65</i>
6.	CONCLUSÕES.....	69
7.	REFERÊNCIAS BIBLIOGRÁFICAS	71
ANEXO A		
ANEXO B		

LISTA DE FIGURAS

Figura 3.1 - Sistema de Coordenadas Fixo e Móvel	12
Figura 3.2 - Sistema de Referência Inercial	13
Figura 3.3 - Movimento de "Yaw"	13
Figura 3.4 - Movimento de "Pitch"	13
Figura 3.5 - Movimento de "Roll "	14
Figura 3.6 - Disposição dos propulsores horizontais no ROV	20
Figura 3.7 - Disposição Propulsores Frontais	21
Figura 3.8 - Disposição Propulsores Traseiros	22
Figura 3.9 - Relação de empuxos verticais	24
Figura 3.10 - Relação Empuxo x Velocidade para os propulsores	25
Figura 3.11 - Relação Torque x Controle de Voltagem para o propulsor	26
Figura 4.1 - Overview do Simulador	29
Figura 4.2 - Bloco de Modelagem	31
Figura 4.3 - Modelagem do Módulo “Modelo Matemático”	32
Figura 4.4 - Modelagem do Módulo “Sistema de Coordenadas (Euler)”	33
Figura 4.5 - Modelagem do Módulo “Modelo do Ambiente”	34
Figura 4.6 - Modelagem do Módulo “Somatória de Forças”	35
Figura 4.7 - Modelagem do Cálculo da Força Hidrodinâmica	37
Figura 4.8 - Ambiente Virtual 3D	39
Figura 4.9 - Diagrama de distribuição de tarefas por PC	39
Figura 4.10 - Host PC com o programa em execução	40
Figura 4.11 - Target PC com o software de simulação em execução	41
Figura 4.12 - Gráficos Referência do ROV em movimento no meio aquático	44
Figura 4.13 - Gráficos Obtidos do ROV através da simulação do modelo apresentado	45

Figura 5.1 - Malha do Sistema de controle e dinâmica do ROV	50
Figura 5.2 - Diagrama de blocos	52
Figura 5.3 - Gráfico da resposta em V_x a um degrau de amplitude 0,5 m/s	52
Figura 5.4 - Gráfico da resposta em V_y a um degrau de amplitude 0,5 m/s	53
Figura 5.5 - Gráfico da resposta em V_z a um degrau de amplitude 0,5 m/s	54
Figura 5.6 - Gráfico da resposta em V_ψ a um degrau de amplitude 0,5 m/s	55
Figura 5.7 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. X em função do tempo	56
Figura 5.8 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. Y em função do tempo	56
Figura 5.9 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. Z em função do tempo	57
Figura 5.10 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. ψ em função do tempo	57
Figura 5.11 - Curva de reação de um sistema de primeira ordem com atraso	59
Figura 5.12 - Método para identificação dos parâmetros da curva de reação	60
Figura 5.13 - Diagrama de Blocos do controlador projetado para cada grau de liberdade separadamente	63
Figura 5.14 - Gráfico da Velocidade em X com o PID projetado	64
Figura 5.15 - Gráfico da Velocidade em Y com o PID projetado	64
Figura 5.16 - Gráfico do Posicionamento em Z com o PID projetado	65
Figura 5.17 - Gráfico do Posicionamento em ψ com o PID projetado	65
Figura 5.18 - Diagrama de blocos de um sistema genérico	66
Figura 5.19 – Barreiras de alta e baixa frequência	68

LISTA DE TABELAS

Tabela 3.1 - Siglas e seus significados para os cálculos de modelagem _____15

Tabela 5.1 - Método de Ziegler e Nichols para Curva de Reação _____60

Tabela 5.2 - Valores calculados para os graus de liberdade a serem controlados _____61

Tabela 5.3 - Valores dos ganhos dos controladores PID _____64

1. Introdução

Na era contemporânea é crescente a utilização de veículos não tripulados, sistemas que não necessitam pilotos embarcados para serem guiados e para isso são controlados a distância por meios eletrônicos e computacionais, sob a supervisão e governo humanos ou sem a sua intervenção, por meio de Controladores Lógicos Programáveis (PLC). Esses veículos são utilizados para diversas funções como sensores remotos, exploração de petróleo, vigilância aérea, transporte, ataques militares, pesquisas entre outras.

Este trabalho abordará parte do desenvolvimento de um ROV (Remote Operated Vehicle), veículo aquático para exploração e produção em grandes profundidades, dedicado a vistoriar dutos de vazão de hidrelétricas. A inspiração deste desenvolvimento vem da dificuldade de vistoriar uma hidrelétrica, pois se deve parar seu funcionamento para que seja possível a entrada de mergulhadores que farão a inspeção dos elementos submersos. Além disso, os ROV's presentes no mercado também trabalham com a usina parcialmente desligada e isso resulta em uma perda de produção energética e consequentes perdas no lucro. O ROV deste trabalho será projetado para trabalhar com a parte inspecionada desligada, mas o restante funcionando, se não plenamente, apenas com pequena redução na vazão. Essas condições de trabalho implicam que o ROV estará sujeito a diversos distúrbios e para que seja possível contornar esses distúrbios deve-se projetar um sistema de controle suficientemente robusto.

As vantagens da utilização de tais veículos vão desde evitar o uso de pessoas para essas funções, evitando que as mesmas sejam expostas aos riscos envolvidos nesta tarefa, até possibilitar o acesso a lugares que são inacessíveis ao ser humano, seja por tamanho, profundidade, temperatura entre outros fatores.

2. Objetivos

2.1. Objetivo Geral

Trata-se do desenvolvimento do sistema de comando em tempo-real para um ROV projetado para operação em ambientes com correnteza, desenvolvido no Laboratório de Veículos Não Tripulados da Universidade de São Paulo. O sistema de comando terá os modos em malha aberta e malha fechada. No modo em malha aberta, o sistema de controle traduz os comandos de movimento (direção, sentido e intensidade) gerados

pelo usuário através de um joystick em valores de rotação para os propulsores do veículo. Para tal, utiliza um algoritmo de alocação de empuxos. O sistema em malha fechada opera ao redor do sistema de alocação de empuxos tentando manter o veículo em determinada posição ou movimento em uma dada direção com o respectivo valor de velocidade.

2.2. Objetivo Específico

Nesse trabalho será desenvolvido o sistema de controle dos propulsores de um veículo aquático do tipo ROV que possui como principal função a inspeção de tubulações de hidrelétricas. Para atingir este objetivo deve-se projetar um sistema de controle em tempo real que seja suficientemente robusto para lidar com todas as incertezas envolvidas no projeto, dentre elas a modelagem de fatores ambientais como correnteza e forças hidrodinâmicas, dinâmica de atuadores e sensores, entre outros fatores.

Para estes fins foi desenvolvido, também, um simulador em tempo real para avaliação comportamental do veículo em ambiente virtual e desenvolvimento do sistema de controle a partir das informações obtidas em simulação.

3. Metodologia

3.1. Modelagem do ROV

Todo o equacionamento apresentado a seguir é apresentado em [1] e em [9], mas para entendimento do que foi feito neste trabalho os mesmos serão apresentados aqui.

A equação de movimento para um corpo rígido é obtida através da segunda lei de Newton, onde a somatória das forças externas agindo sobre o corpo é igual à taxa de variação do momento linear do mesmo em relação ao tempo. Pode-se aplicar o mesmo para o momento, onde a resultante dos momentos externos que agem no veículo é igual a taxa de variação de momento angular do mesmo em relação ao tempo.

Ou seja,

$$\Sigma \vec{F} = \frac{d}{dt}(m \cdot \vec{V})$$

$$\Sigma \vec{M} = \frac{d}{dt}\vec{H}$$

As forças externas são todas as forças agindo no veículo cuja origem não seja interna ao corpo estudado, neste estudo são consideradas as forças de gravidade, empuxo, esforços hidrodinâmicos e tração dos propulsores.

A taxa de variação de movimento e do momento angular geram as chamadas forças e momentos inerciais, respectivamente. Esses esforços são chamados de inerciais devido ao fato de dependerem exclusivamente da dinâmica do corpo, ou seja, essas forças são geradas devido a variações das velocidades lineares e angulares.

Sendo assim, aplicando as equações apresentadas acima no veículo estudado, obtemos a equação de movimento do ROV de dimensão 6:

$$[M_{ij} + A_{ij}][\dot{V}, \dot{\omega}] + F_I = F_F + F_G + F_B + F_T$$

Onde:

- M_{ij} é a matriz de inércia.
- A_{ij} é a matriz de massa adicional.
- $[\dot{V}, \dot{\omega}]$ é o vetor composto pelas acelerações lineares e angulares:

$$[\dot{V}, \dot{\omega}] = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \begin{array}{l} \rightarrow \text{componente da velocidade linear no eixo } x \\ \rightarrow \text{componente da velocidade linear no eixo } y \\ \rightarrow \text{componente da velocidade linear no eixo } z \\ \rightarrow \text{componente da velocidade angular no eixo } x \\ \rightarrow \text{componente da velocidade angular no eixo } y \\ \rightarrow \text{componente da velocidade angular no eixo } z \end{array}$$

- F_I é o vetor de esforços e momentos devido às forças inerciais compostas pelos termos de forças centrípetas e de Coriolis do sistema não inercial
- F_F é o vetor de esforços e momentos devidos às forças hidrodinâmicas atuantes no veículo
- F_G é o vetor de esforços e momentos devido à gravidade
- F_B é o vetor de esforços e momentos devido ao empuxo
- F_T é o vetor de esforços e momentos devido à tração do propulsor

3.1.1. Sistema de Coordenadas

Os cálculos de todas as forças, velocidades, acelerações, entre outros, utilizam formulações que possuem seus componentes no sistema de coordenadas fixo ao veículo. Entretanto, para alguns parâmetros desejados, como a gravidade e o empuxo, assim como melhor visualização do que está acontecendo com o veículo é necessário obter estes parâmetros em um sistema de coordenadas fixo e conhecido.

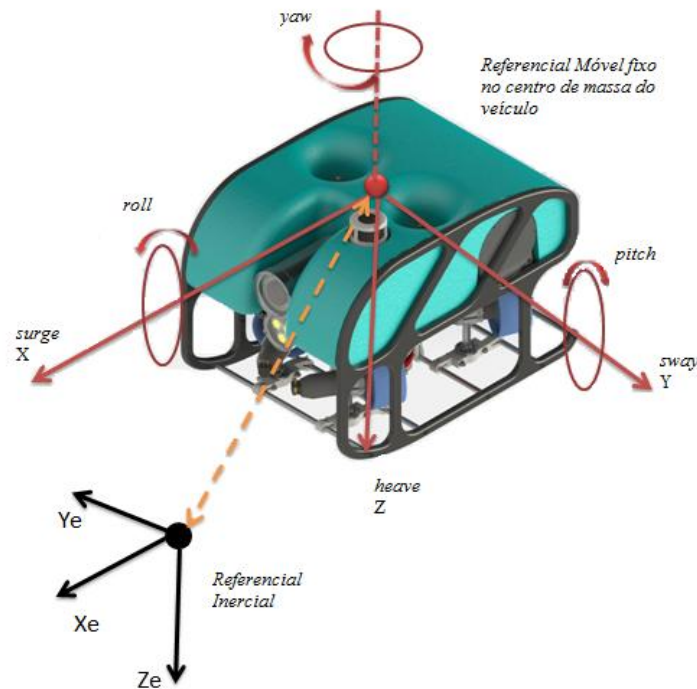


Figura 3.1 - Sistema de Coordenadas Fixo e Móvel

A figura acima ilustra a relação entre os sistemas de coordenadas fixo e móvel. Os resultados obtidos no sistema de coordenadas não-inercial não dizem muito sobre a real trajetória do veículo, portanto é necessário realizar uma correlação entre os dois sistemas de coordenadas para melhor interpretação dos dados.

Para realizar essa correlação foi utilizada a teoria de ângulos de Euler, que descreve a orientação de um corpo rígido em um espaço euclidiano tridimensional. Para definir uma orientação específica a um objeto, ele deve ser submetido a 3 rotações consecutivas descritas pelos ângulos de Euler, ou seja, compor uma matriz de rotação como um produto de 3 rotações elementares.

A transformação de coordenadas que relaciona o sistema inercial com o sistema de coordenadas ligado ao corpo pode ser representada por uma equação matricial de forma:

$$\mathbf{x} = \mathbf{\lambda} \mathbf{x}'$$

Partindo-se do sistema X, Y, Z, fixo na terra, e orientado conforme a figura 1, o sistema móvel realiza a seqüência:

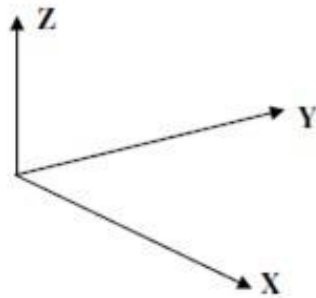


Figura 3.2 - Sistema de Referência Inercial

- 1) Rotação em torno de "Z": X, Y, Z \rightarrow x', y', z'

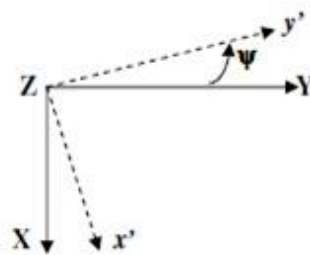


Figura 3.3 - Movimento de "Yaw"

- 2) Rotação em torno de y': x', y', z' \rightarrow x'', y'', z''

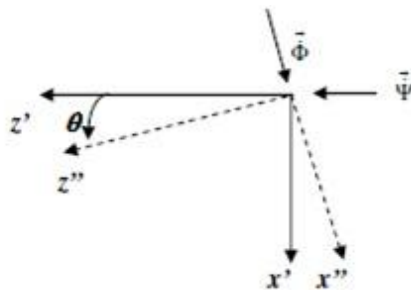


Figura 3.4 - Movimento de "Pitch"

3) Rotação em torno de x'' : $x'', y'', z'' \rightarrow x, y, z$.

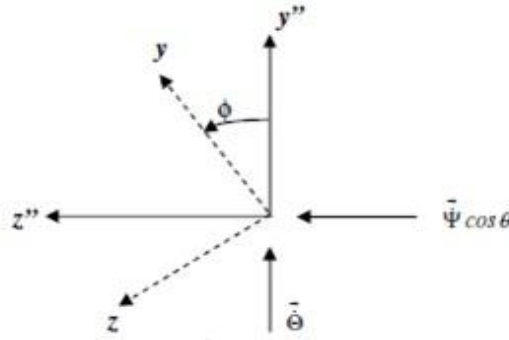


Figura 3.5 - Movimento de "Roll"

Através desta aplicação foi possível chegar à matriz de transformação T_B e à relação entre a taxa angular de Euler e a velocidade angular em cada eixo. Ambas são apresentadas a seguir:

$$T_B = \begin{bmatrix} \cos\theta \cdot \cos\psi & \sin\phi \cdot \sin\theta \cos\psi & \cos\phi \cdot \sin\theta \cdot \cos\psi \\ \cos\theta \cdot \sin\psi & \sin\phi \cdot \sin\theta \cdot \sin\psi + \cos\phi \cdot \cos\psi & \cos\phi \cdot \sin\theta \cdot \sin\psi - \sin\phi \cdot \cos\psi \\ -\sin\theta & \sin\phi \cdot \cos\theta & \cos\phi \cdot \cos\theta \end{bmatrix}$$

Portanto, definida T_B , temos que a relação entre as componentes da velocidade linear no referencial inercial e no referencial móvel:

$$[V_{xe}, V_{ye}, V_{ze}]^t = T_B \cdot [V_x, V_y, V_z]^t$$

A posição do ROV referente aos eixos estacionários pode ser calculada integrando a equação acima.

A relação entre a taxa angular de Euler e a velocidade angular é expressa a seguir:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \cdot \tan\theta & \cos\phi \cdot \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \cdot \sec\theta & \cos\phi \cdot \sec\theta \end{bmatrix} \cdot \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

3.1.2. Desenvolvimento do equacionamento do modelo

Nesta seção será explicado mais detalhadamente cada termo da equação 1, mostrando o equacionamento de cada força contida no modelo de movimentação do ROV em meio aquático.

Para entendimento das equações e seus respectivos termos, é apresentada a seguir uma tabela com a identificação de cada variável contida nas formulações.

Tabela 3.1 - Siglas e seus significados para os cálculos de modelagem

Sigla	Variável de Projeto
A_{ij}	Massa adicional
$C_{Fx}, C_{Fy}, C_{Fz}, C_{Nx}, C_{Ny}, C_{Nz}, C_{N\omega x}, C_{N\omega y}, C_{N\omega z}$	Coefficientes de forças hidrodinâmicas
G	Aceleração da gravidade
$I_{xx}, I_{yy}, I_{zz}, J_{xy}, J_{xz}, J_{yz}$	Momento de inércia e produtos de inércia
M	Massa do ROV
m_{ij}	Massa generalizada
U_x, U_y, U_z	Componentes de U (velocidade de corrente)
V_x, V_y, V_z	Componentes de V (velocidade linear)
V_{R1}, V_{R2}, V_{R3}	Componentes de V_R (velocidade relativa com o fluido)
x_B, y_B, z_B	Posição do centro de flutuação
x_G, y_G, z_G	Posição do centro de gravidade
A	Attack angle
B	Side slip angle
Γ	Side attack angle
P	Massa específica do fluido
$\omega_x, \omega_y, \omega_z$	Componente de ω (velocidade angular)
∇, ∇_R	Volume de deslocamento, Volume total do corpo

3.1.2.1. Esforços Inerciais

As forças inerciais são as geradas devido a acelerações realizadas pelo veículo, para se obter a formulação dessas forças e momentos é necessário realizar diversas operações e utilizar diversas formulações que não serão apresentadas aqui. Tais esforços são afetados também pelo efeito de massa adicional.

Quando um veículo é acelerado, o fluido próximo a ele também é acelerado e além da geração de esforços inerciais devido à inércia do veículo há, também, a geração de forças inerciais devido a essa massa de fluido acelerada.

Pode-se considerar que a massa adicional de um veículo é uma adição de inércia a dinâmica do veículo, devido às mudanças de energia cinética do fluido próximo ao veículo. Como a inércia gerada pela massa adicional existe para todos os graus de liberdade do veículo, o termo de massa adicional é uma matriz quadrada de dimensão seis, semelhante à matriz de inércia. [9]

A obtenção dos termos da matriz de massa adicional deve ser feita verificando-se a influência separada das partes do veículo e depois somá-las aplicando os coeficientes de interferência. O cálculo dos termos da matriz de massa adicional, assim como o método utilizado para o mesmo não está no escopo deste trabalho. Sendo assim, serão apresentados apenas os resultados finais da influência do efeito de massa adicional nos esforços inerciais.

$$F_I = \begin{bmatrix} \omega_y(mV_z - mx_G\omega_y + A_{33}V_{R3}) - \omega_z(mV_y + mx_G\omega_z - mz_G\omega_x + A_{22}V_{R2}) \\ \omega_z(mV_x + mz_G\omega_y + A_{11}V_{R1}) - \omega_x(mV_z - mx_G\omega_y + A_{33}V_{R3}) \\ \omega_x(mV_y + mx_G\omega_z - mz_G\omega_x + A_{22}V_{R2}) - \omega_y(mV_x + mz_G\omega_y + A_{11}V_{R1}) \\ \omega_y(-J_{xz}\omega_x + I_{zz}\omega_z + A_{66}\omega_z) - \omega_z(-J_{xy}\omega_x + I_{yy}\omega_y + A_{55}\omega_y) + mz_G(V_z\omega_x - V_x\omega_z) + (A_{33} - A_{22})V_{R2}V_{R3} \\ \omega_z(I_{xx}\omega_x - J_{xy}\omega_y - J_{xz}\omega_z + A_{44}\omega_x) - \omega_x(-J_{xz}\omega_x + I_{zz}\omega_z + A_{66}\omega_z) + \\ + mz_G(V_z\omega_y - V_y\omega_z) + mx_G(V_x\omega_y - V_y\omega_x) + (A_{11} - A_{33})V_{R1}V_{R3} \\ \omega_x(-J_{xy}\omega_x + I_{yy}\omega_y + A_{55}\omega_y) - \omega_y(I_{xx}\omega_x - J_{xy}\omega_y - J_{xz}\omega_z + A_{44}\omega_x) + mx_G(V_x\omega_z - V_z\omega_x) + (A_{22} - A_{11})V_{R1}V_{R2} \end{bmatrix}$$

Onde as componentes de velocidade relativa são definidas como:

$$V_{R1} = V_x - U_x \quad , \quad V_{R2} = V_y - U_y \quad , \quad V_{R3} = V_z - U_z$$

Termos da matriz de massa generalizada e massa adicional são expressos a seguir:

$$[m_{ij} + A_{ij}] = \begin{bmatrix} m + A_{11} & A_{12} & A_{13} & A_{14} & mz_G + A_{15} & -my_G + A_{16} \\ A_{21} & m + A_{22} & A_{23} & -mz_G + A_{24} & A_{25} & mx_G + A_{26} \\ A_{31} & A_{32} & m + A_{33} & my_G + A_{34} & -mx_G + A_{35} & A_{36} \\ A_{41} & -mz_G + A_{42} & my_G + A_{43} & I_{xx} + A_{44} & -J_{xy} + A_{45} & -J_{xz} + A_{46} \\ mz_G + A_{51} & A_{52} & -mx_G + A_{53} & -J_{xy} + A_{54} & I_{yy} + A_{55} & -J_{yz} + A_{56} \\ -my_G + A_{61} & mx_G + A_{62} & A_{63} & -J_{xz} + A_{64} & -J_{yz} + A_{65} & I_{zz} + A_{66} \end{bmatrix}$$

3.1.2.2. Esforços Hidrodinâmicos

Há dois tipos de forças hidrodinâmicas, uma se deve a relação entre a velocidade linear e do fluido e a outra se deve a velocidade angular do veículo no meio. Ambas podem ser equacionadas da seguinte maneira:[1]

$$F_F = 0,5 \cdot \rho \cdot |V_R|^2 \cdot \nabla_R^{\frac{2}{3}} \begin{bmatrix} C_{Fx}(\alpha, \beta) \\ C_{Fy}(\beta, \gamma) \\ C_{Fz}(\alpha, \gamma) \\ C_{Nx}(\gamma) \nabla_R^{\frac{1}{3}} \\ C_{Ny}(\alpha) \nabla_R^{\frac{1}{3}} \\ C_{Nz}(\beta) \nabla_R^{\frac{1}{3}} \end{bmatrix} + 0,5 \cdot \rho \cdot \nabla_R^{\frac{5}{3}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ C_{N\omega x} \omega_x |\omega_x| \\ C_{N\omega y} \omega_y |\omega_y| \\ C_{N\omega z} \omega_z |\omega_z| \end{bmatrix}$$

Onde o *attack angle* (α), *side slip angle* (β) e *side attack angle* (γ) são definidos a seguir:

$$\alpha = \arctan\left(\frac{V_{R3}}{V_{R1}}\right) \quad , \quad \beta = \arctan\left(\frac{V_{R2}}{V_{R1}}\right) \quad , \quad \gamma = \arctan\left(\frac{V_{R3}}{V_{R2}}\right)$$

3.1.2.3. Esforços de Gravidade e Empuxo

As forças de gravidade e de empuxo são dependentes dos ângulos de Euler, portanto, para obtê-las, as matrizes de transformação devem ser utilizadas para posterior aplicação dos ângulos nas formulações dadas a seguir.

A fórmula da força gravitacional na forma vetorial, assumindo um centro de gravidade distante da origem do sistema de coordenadas não inercial, é:

$$F_G = m \cdot g \cdot \begin{bmatrix} -\sin(\theta) \\ \sin(\varphi) \cdot \cos(\theta) \\ \cos(\varphi) \cdot \cos(\theta) \\ y_G \cdot \cos(\varphi) \cdot \cos(\theta) - z_G \cdot \sin(\varphi) \cdot \cos(\theta) \\ -x_G \cdot \cos(\varphi) \cdot \cos(\theta) - z_G \cdot \sin(\theta) \\ x_G \cdot \sin(\varphi) \cdot \cos(\theta) + y_G \cdot \sin(\theta) \end{bmatrix}$$

E a fórmula para a força de empuxo no formato vetorial, assumindo seu centro de empuxo distante da origem do sistema de coordenadas não inercial, é:

$$F_B = \rho \cdot g \cdot \nabla \cdot \begin{bmatrix} \sin(\theta) \\ -\sin(\varphi) \cdot \cos(\theta) \\ -\cos(\varphi) \cdot \cos(\theta) \\ -y_G \cdot \cos(\varphi) \cdot \cos(\theta) + z_G \cdot \sin(\varphi) \cdot \cos(\theta) \\ x_G \cdot \cos(\varphi) \cdot \cos(\theta) + z_G \cdot \sin(\theta) \\ -x_G \cdot \sin(\varphi) \cdot \cos(\theta) - y_G \cdot \sin(\theta) \end{bmatrix}$$

3.1.2.4. Esforços de Propulsão

Pode-se assumir que uma força de tração gerada pelo propulsor depende exclusivamente da rotação imposta às hélices do mesmo. Na realidade há influência de outros fatores como densidade do fluido, velocidade do veículo, entre outros, entretanto como estes não serão tratados na formulação utilizada para a força de tração dos propulsores, será adotado o caso mais simples dependente apenas da rotação.

Esta componente da equação de movimento do ROV foi inicialmente modelada igual à forma mostrada em [1] para efeito de validação do simulador desenvolvido neste trabalho, porém, posteriormente, implementada a alocação de empuxo no software de controle do ROV, o cálculo dos esforços de propulsão é feito através do efeito causado pelo empuxo de cada um dos 6 propulsores, sendo enviado o empuxo calculado pelo software de controle para o simulador, nele os mesmos são compostos de forma a resultar no vetor de resultantes nos seis graus de liberdade do veículo para efeito de cálculo da somatória de forças.

Portanto os cálculos apresentados a seguir serão utilizados apenas para a validação do simulador:

$$F_T = \begin{bmatrix} F_{HR} + F_{HL} \\ F_S \\ F_V \\ -Z_S \cdot F_S \\ Z_H \cdot (F_{HL} + F_{HR}) \\ Y_H \cdot (F_{HL} - F_{HR}) \end{bmatrix}$$

Onde FHR, FHL, FS, FV são componentes da força de propulsão FT e ZH, YH e ZS são medidas construtivas dos propulsores do ROV.

Os cálculos referentes aos realizados pelo simulador para a disposição de propulsores do ROV serão apresentados na seção 3.2 a seguir através da matriz pseudo-inversa.

3.2. Alocação de Empuxo

O problema da alocação de empuxo consiste basicamente em: a partir do cálculo das forças necessárias para manter a embarcação na posição desejada, distribuí-los de maneira ótima entre os propulsores para que a resultante das forças e momento seja igual às do controlador, para tanto é necessário que ele considere as restrições funcionais tais como interferências entre propulsores, interação casco-propulsor, além

da saturação em cada propulsor. Quando o empuxo total disponível é maior do que a força comandada há várias possíveis configurações para o problema de alocação e algum novo critério deve ser incorporado para superar esta indeterminação.

O controle do posicionamento do ROV ocorre em seis graus de liberdade, como mostra a Figura 3.1. Neste trabalho, o estudo será limitado ao plano horizontal, ao longo dos movimentos de translação nos eixos x_1 (*surge*) e x_2 (*sway*) e movimento de rotação em torno do eixo x_6 (*yaw*), pois os movimentos de (*pitch*) e (*roll*) são naturalmente estáveis e não necessitam de controle, e o movimento de (*heave*) tem um relação direta com o comando, portanto sua movimentação passa por outra função para seu comando.

Serão usados para este projeto 6 propulsores, sendo 2 para movimentação vertical e 4 para a horizontal. Estes propulsores são de posição fixa, entretanto seu ângulo de azimute (ângulo com a vertical) pode ser regulado em uma pequena faixa de valores dois a dois (frontais e traseiros) e como a modelagem da alocação de empuxo em função das forças comandadas leva em conta a quantidade de propulsores, seu arranjo em relação ao centro de gravidade da embarcação e o ângulo de azimute tornam esses fatores ‘fixos’ o que possibilita a implementação da rotina de alocação de empuxo, a disposição e numeração dos propulsores horizontais são mostradas na figura a seguir.

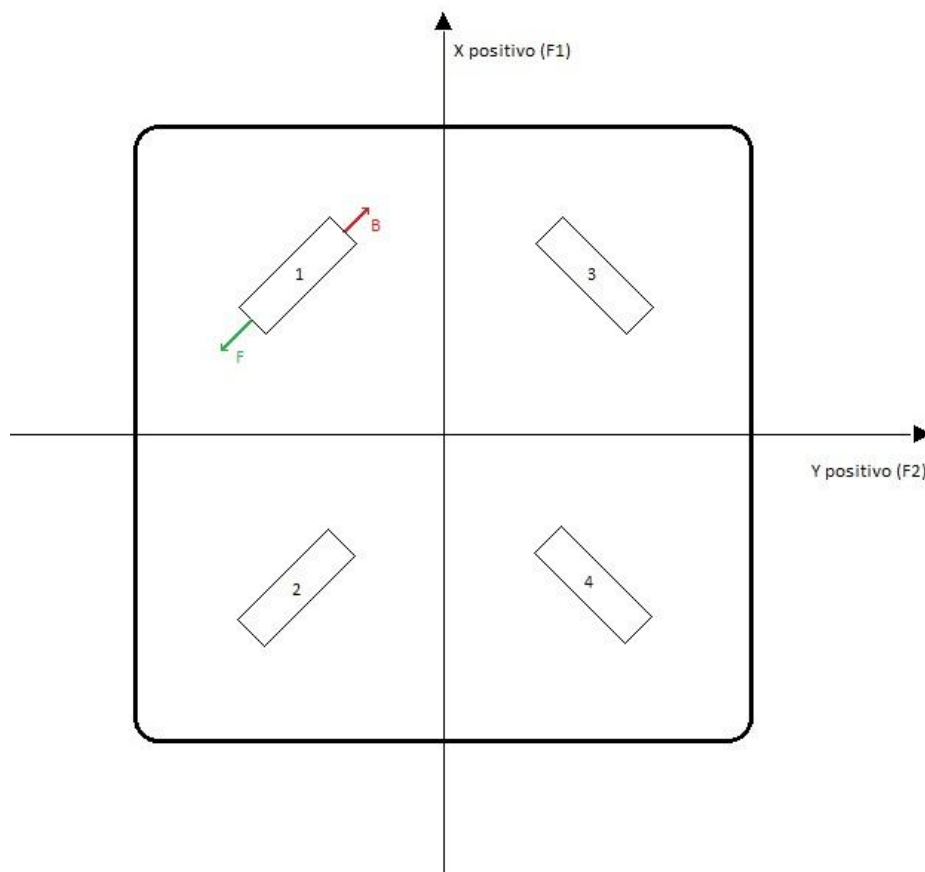


Figura 3.6 - Disposição dos propulsores horizontais no ROV

Onde F representa o movimento de *Forward* do propulsor onde ele aplica a maior força e impulsiona o ROV para frente, analogamente B representa o movimento de *Reverse*.

3.2.1. Solução Para a Alocação de Empuxo

Comumente os problemas de alocação de empuxo são tratados do mesmo modo, salvo algumas exceções que consideram mais variáveis ao equacionar o problema. O vetor de entrada do sistema com as forças e momentos enviados pelo operador através do joystick é dado pelo vetor F (lembrando que essa modelagem considera apenas a movimentação “horizontal”).

$$F = \begin{Bmatrix} F_1 \\ F_2 \\ F_6 \end{Bmatrix}$$

Onde,

F_1 : valor da força desejada em 'surge'
 F_2 : valor da força desejada em 'sway'
 F_6 : valor da momento desejado em 'yaw'

O vetor de saída deste sistema retorna diretamente as forças de cada propulsor na direção de atuação do mesmo e é dado pelo vetor T definido abaixo sendo a numeração dos propulsores utilizada é a mostrada na Figura 3.7.

$$T^T = [T_1 \quad T_2 \quad T_3 \quad T_4]$$

Onde,

T_n : valor da força para o propulsor n

A matriz 'A' torna a relação $F = A \cdot T$ válida,

$$A = \begin{bmatrix} \cos \alpha_F & \cos \alpha_B & \cos \alpha_F & \cos \alpha_B \\ \sin \alpha_F & \sin \alpha_B & -\sin \alpha_F & -\sin \alpha_B \\ \text{braço}F & \text{braço}B & -\text{braço}F & -\text{braço}B \end{bmatrix}$$

Sendo α_F e α_B os ângulos de azimutes frontais e traseiros, respectivamente, e $\text{braço}F$ e $\text{braço}B$ são as distâncias entre a linha de atuação do propulsor e o centro de massa do veículo, sendo obtidos a partir da disposição dos mesmos no veículo. Tais valores podem ser observados nas figuras 3.8 e 3.9.

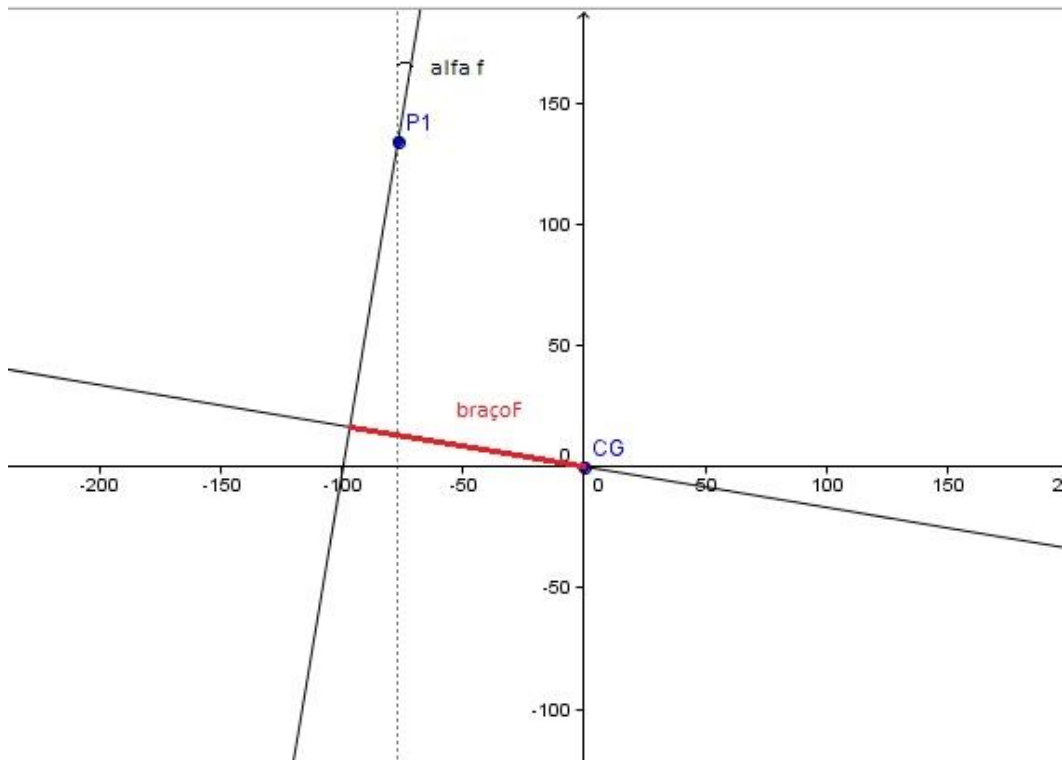


Figura 3.7 - Disposição Propulsores Frontais

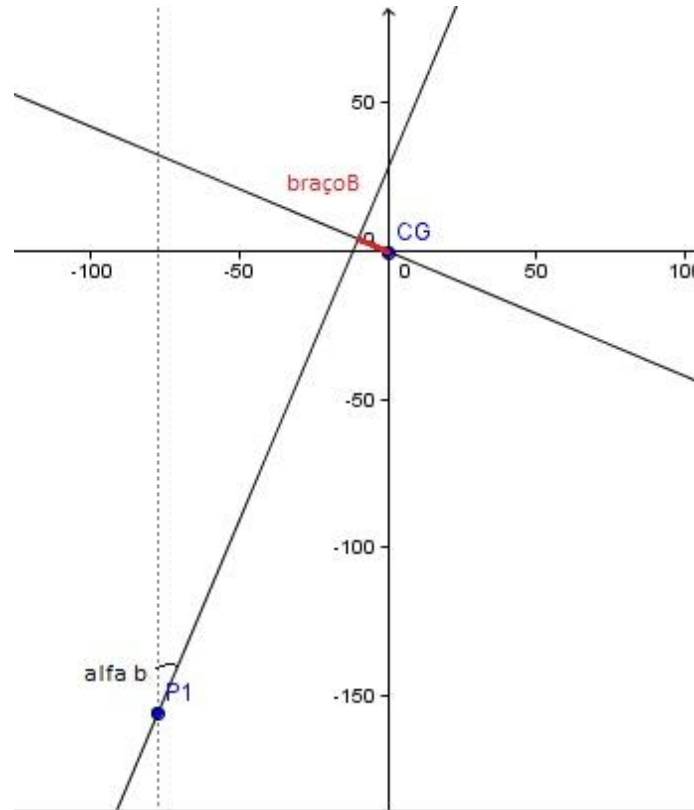


Figura 3.8 - Disposição Propulsores Traseiros

Substituindo os valores de $\alpha_F = \alpha_B = 30^\circ$ encontramos:

$$\text{braçoF} = 0,15049 \text{ m} ; \text{braçoB} = 0,00004 \text{ m}$$

Com os valores fixados a matriz de transferência A fica com os seguintes valores

$$A = \begin{bmatrix} 0,866 & 0,866 & 0,866 & 0,866 \\ 0,5 & 0,5 & -0,5 & -0,5 \\ 0,15049 & 0,00004 & -0,15049 & -0,00004 \end{bmatrix}$$

3.2.1.1. Matriz Pseudo-Inversa

A solução escolhida para resolver e implementar a alocação de empuxo foi usando a matriz pseudo-inversa. A matriz 'A' possui 3 linhas ($m=3$) e 4 colunas ($n=4$), ou seja, $A_{m \times n} = A_{3 \times 4}$, como $m < n$ será feita uma matriz inversa direita, caso $m > n$ seria usada uma matriz inversa esquerda e uma matriz inversa direita é definida de modo que $A_{\text{direita}}^{-1} = A^T (A A^T)^{-1}$ e assim garante-se que $A A_{\text{direita}}^{-1} = I_m$ e I_m tem dimensão $m=3$ e a dimensão da pseudo inversa é $A_{\text{direita}}^{-1} = A_{4 \times 3}$.

Um sistema como os mostrado ($m < n$) tem como uma possível solução para a equação matricial $T = A^T (A A^T)^{-1} F + (I - A^T (A A^T)^{-1} A) y$, onde $y \in R^n$ um vetor

arbitrário, quando múltiplas soluções são possíveis, como no caso acima, pode-se adotar aquela solução que otimiza algum critério. Por exemplo, a solução com norma euclidiana mínima é aquela que toma $y = 0$, que no caso do ROV essa solução aplicada a um sistema de mínimos quadrados minimiza a potência utilizada.

Como a posição dos propulsores é fixa no ROV, tanto matriz ‘A’ quanto a pseudo-inversa podem ser calculadas previamente e embutidas na programação e, portanto, a expressão final para a alocação de empuxo e suas dimensões finais fica:

$$\mathbf{T}_{(4 \times 1)} = (\mathbf{A}_{direita}^{-1})_{(4 \times 3)} \cdot \mathbf{F}_{(3 \times 1)}$$

Realizando os cálculos necessários, definiram-se os valores para $\mathbf{A}_{direita}^{-1}$ e são esses valores que serão programados na rotina de alocação de empuxo:

$$\mathbf{A}_{direita}^{-1} = \begin{bmatrix} 0.2887 & -0.0003 & 3.3234 \\ 0.2887 & 1.0003 & -3.3234 \\ 0.2887 & 0.0003 & -3.3234 \\ 0.2887 & -1.0003 & 3.3234 \end{bmatrix}$$

Lembrando que esses valores são para $\alpha_F = \alpha_B = 30^\circ$.

3.2.1.2. Alocação de Empuxo Para a Movimentação Vertical

A movimentação na direção z (‘heave’) não necessita uma alocação de empuxo complexa e isso se deve ao fato de sua relação ser linear, já que os dois propulsores verticais não possuem ângulos em relação a vertical e são simétricos em relação ao centro de massa. Sabendo isso, a movimentação em ‘roll’ e ‘pitch’ é praticamente nula, logo a força em cada propulsor vertical é uma porcentagem (dada pelo joystick) do valor total que é determinado a partir do máximo empuxo de cada propulsor. A figura a seguir ilustra o caso mencionado.

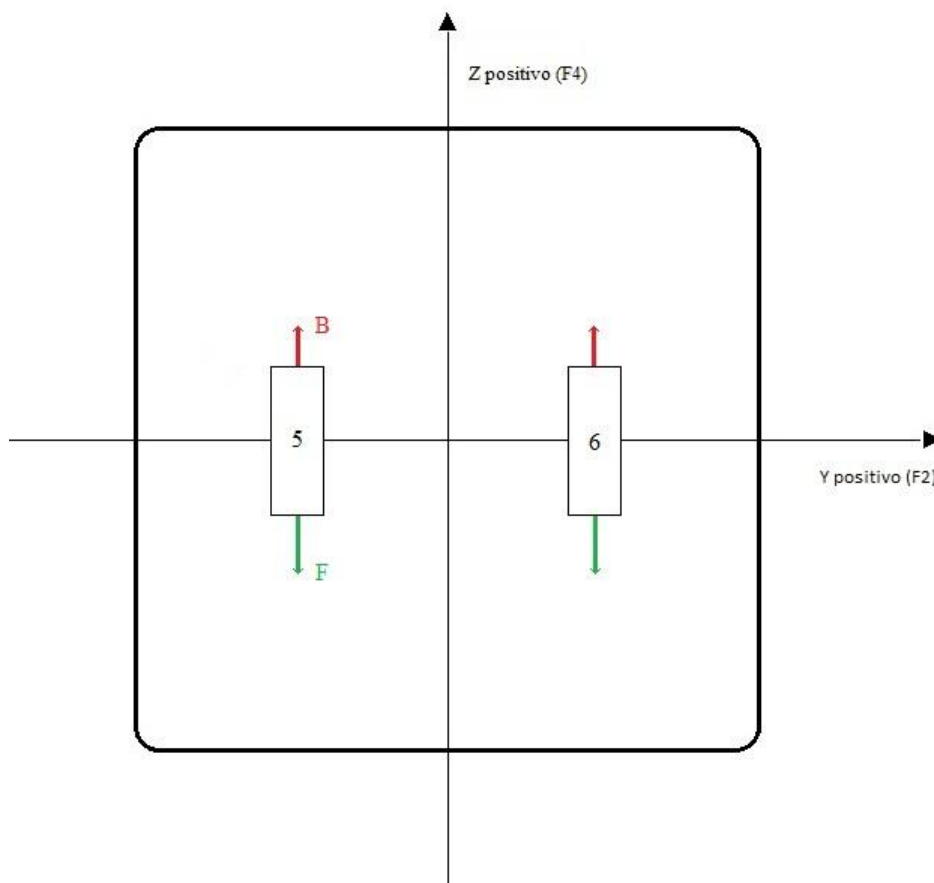


Figura 3.9 - Relação de empuxos verticais

Onde novamente F representa o movimento de “Forward” do propulsor onde ele aplica a maior força e impulsiona o ROV para ‘baixo’, analogamente B representa o movimento de “Reverse”. A alocação será então a soma das porcentagens da máxima força dos propulsores na direção a qual é desejada, em direção ao fundo ou em direção à superfície. Como o joystick manda um valor de 0 a 1, o valor do vetor de cada propulsor será:

$$H_5 = \max F \cdot F_3 \text{ se } F_3 \geq 0$$

$$H_6 = \max R \cdot F_3 \text{ se } F_3 < 0$$

Onde,

H_i : força desejada em 'z' para o propulsor i
 F_3 : força desejada em 'z' enviada pelo joystick
 $\max F$: força máxima do propulsor em 'Forward'
 $\max R$: força máxima do propulsor em 'Reverse'

Os valores de $\max R$ e $\max F$ são dados do datasheet dos propulsores e também dependem da velocidade em que o veículo está.

3.2.1.3. Teste de Saturação dos Propulsores

Calculadas as forças de cada propulsor, é necessário verificar se essa força não é superior à força máxima que o propulsor fornece. Neste projeto é dado que o empuxo máximo de cada propulsor que compõe o veículo é de 8,3 kg no modo normal. Isso fornece uma força máxima em torno de 81,34 N no modo normal. No modo reverso o máximo empuxo de cada propulsor é de 3,6kg, fornecendo uma força máxima em torno de 35,28 N. Vale a pena ressaltar que esses valores são para o veículo parado, ou seja, conforme o veículo adquire velocidade há variação destes valores. Tendo isso como padrão, o valor para cada propulsor é avaliado e, se for maior do que o máximo permitido, limita-se à máxima potência permitida para evitar danos nos propulsores.

3.2.2. Relação entre a Velocidade do Veículo e a Propulsão Máxima

Tais valores para o empuxo máximo citado anteriormente aplicam-se ao veículo parado. Conforme a alteração da velocidade, esses valores mudam de acordo com uma relação fornecida pelo fabricante, relação mostrada a seguir.

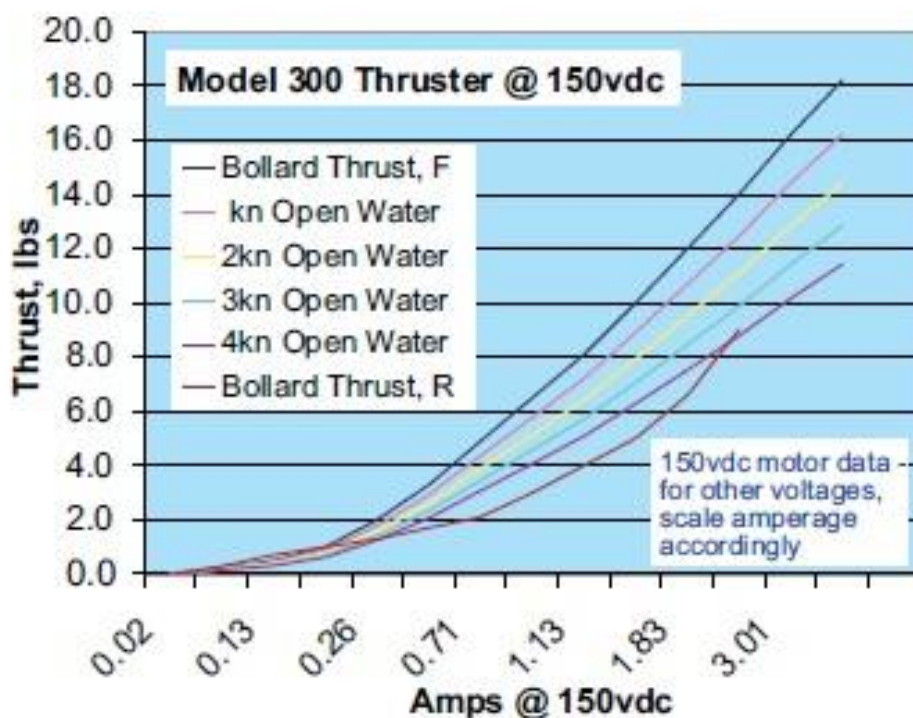


Figura 3.10 - Relação Empuxo x Velocidade para os propulsores [23]

No gráfico observa-se que para diferentes velocidades a relação entre empuxo e amperagem varia e para obter esses valores é necessária uma função que receba a velocidade do veículo e interprete qual curva de valores deve seguir.

3.2.3. Transformação do Empuxo para Voltagem

Com os valores de empuxo de cada propulsor calculados é preciso transformar estes valores em uma variação de voltagem entre $\pm 5V$ para ser enviada ao propulsor. Empuxo e variação da voltagem têm sua relação dada pelo fabricante do propulsor, apresentada a seguir.

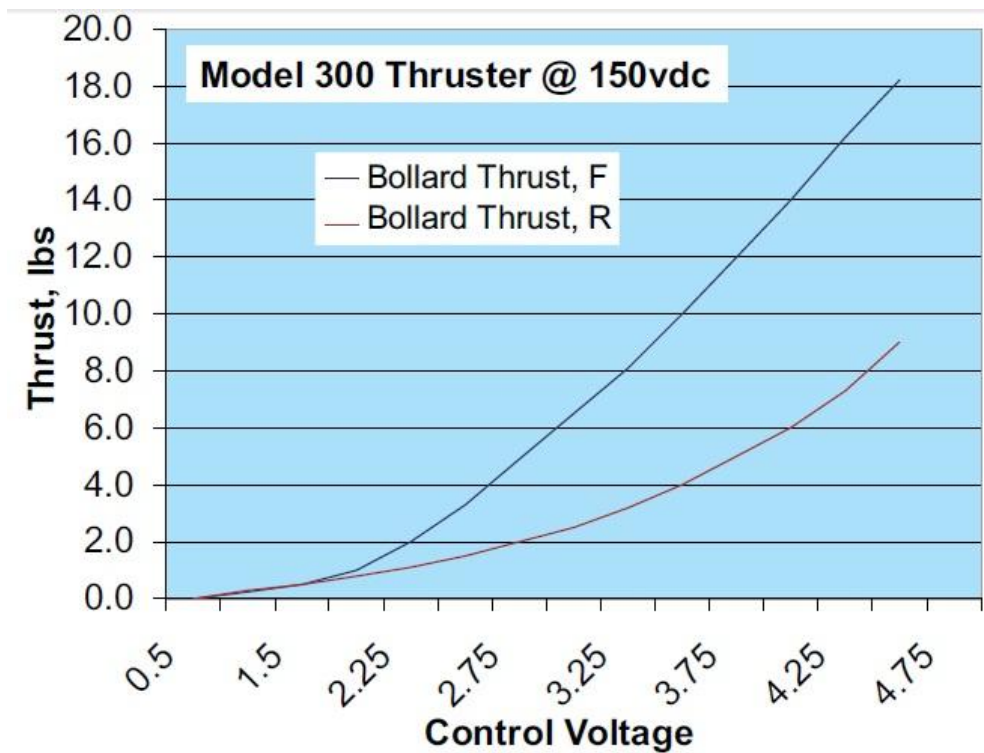


Figura 3.11 - Relação Torque x Controle de Voltagem para o propulsor [23]

O gráfico nos mostra a variação do empuxo em relação ao aumento da voltagem para os dois modos de operação do propulsor, normal (F) e reverso (R). A partir dessa relação foi possível fazer uma função que converte este valor em empuxo dado pela formulação da alocação em uma voltagem que será enviada ao ROV.

4. Simulador de Dinâmica de um ROV - Documentação

4.1. Introdução ao Simulador

O simulador desenvolvido neste trabalho tem como objetivo principal modelar todos os sistemas atuantes no ROV, ou seja, a dinâmica do veículo, do ambiente, dos sensores, dos atuadores e, através da simulação destes, obter resultados suficientemente precisos para que sejam tomadas as decisões corretas referentes à prevenção de futuros problemas que possam ocorrer com o veículo em um teste real.

O software de simulação deve possuir uma estrutura flexível, pois assim facilita-se uma futura substituição ou alteração de componentes particulares sem a alteração do software como um todo, visando um melhor desempenho do mesmo.

Para ser possível atingir este nível de flexibilidade, a arquitetura de software implementada é modular. A estrutura do software baseia-se em 3 Blocos principais, os quais possuem funções específicas e distintas, são estes módulos:

- Modelagem: este bloco contém toda modelagem envolvida na movimentação de um ROV em um meio aquático. Nele se calculam os movimentos do ROV, devido a esforços internos e externos, sendo eles dinâmicos ou estáticos e possui, também, a modelagem do meio ambiente, onde é calculada a interferência hidrodinâmica gerada pelo ambiente no movimento do ROV.
- Comunicação: este bloco contém toda a configuração necessária para a comunicação entre os dois computadores utilizados por meio do protocolo ethernet UDP, unindo assim o computador base com o computador de simulação.
- Interface: este bloco contém o ambiente de simulação 3D representativo do ROV em um meio aquático, desenvolvido em formato VRML, e um conjunto de gráficos suficientes para que o operador do ROV em simulação possa ter uma noção do que realmente está acontecendo com o veículo.

Cada um desses blocos executa diversas funções dentro do simulador e se propõe a conferir uma maior flexibilidade ao software. Entretanto essa divisão ainda não é suficiente e por isso ainda há alguns subníveis para que seja atingido o grau de flexibilidade desejado para o simulador. Entenda-se então que a nomenclatura utilizada neste trabalho para designar futuros subníveis é:

BLOCO \supset MÓDULO \supset SUBMÓDULO

Chega-se até o terceiro nível de modulação para que assim seja possível atender ao objetivo proposto de um simulador flexível o suficiente para atender a qualquer ROV.

4.2. Overview do Simulador – Introdução ao Software

Nesta seção será apresentado o software desenvolvido em ambiente Simulink. Nesta visão geral, a figura 4.1 mostra o arquivo com todos os blocos implementados, sendo cada um deles mostrado com mais detalhes posteriormente.

Através desta figura já é possível ver a facilidade da operacionalização do software, bastando iniciar a simulação. Feito isso a representação 3D inicia automaticamente e as variáveis que o usuário quiser acompanhar durante a simulação podem ser acessadas através dos blocos “scopes” mostrados na figura.

Para alteração de parâmetros e adaptação do software a outro ROV basta entrar no bloco de modelagem e fazer as alterações dos valores contidos neste bloco, feito isso basta iniciar a simulação novamente e os resultados das alterações já são observados através da interface de monitoramento.

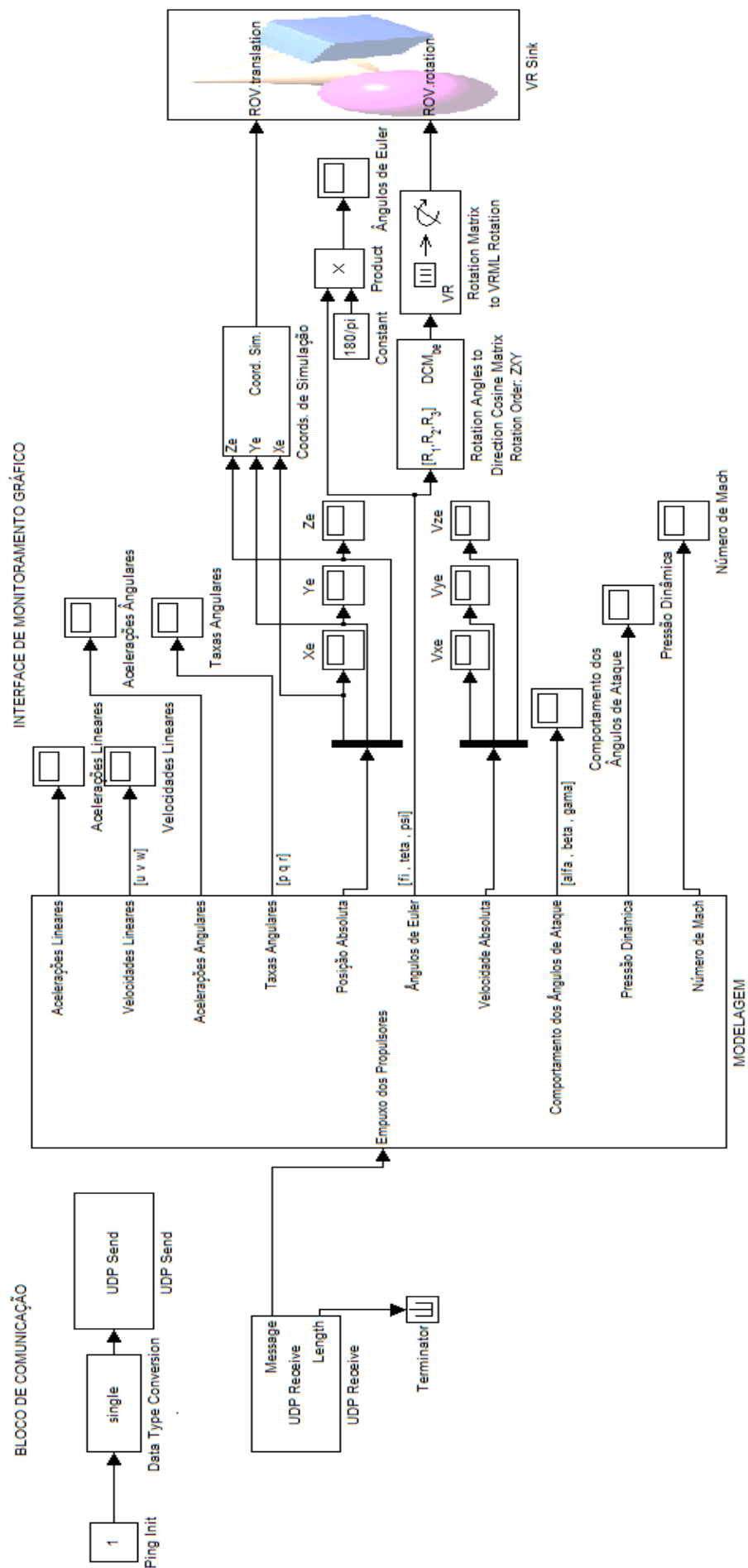


Figura 4.1 - Overview do Simulador

4.3. Bloco Modelagem

Este bloco foi desenvolvido para realizar todos os cálculos necessários para oferecer informações sobre a dinâmica do ROV, levando em consideração forças estáticas e dinâmicas sendo estas externas ou internas. Este é o bloco aonde se faz mais visível a necessidade da arquitetura proposta, pois os cálculos envolvidos nessa modelagem são complexos e, se não forem organizados, a reutilização deste simulador para outro ROV no futuro pode não ser possível devido a incapacidade de reconhecer os termos presentes nos equacionamentos ou mesmo de se adicionar ou retirar esforços.

Para reconstruir a dinâmica do veículo real é preciso considerar que todas as componentes da equação de movimento sejam implementadas o mais próximo da realidade possível, ou seja, com o mínimo de aproximações. Para isso é preciso considerar todas as forças como tendo um caráter não linear, o que confere uma dificuldade ou até impossibilidade de resolução das equações. A resolução das equações diferenciais é realizada através do método de Dormand-Prince ODE45, método este configurado no simulador por obter os resultados mais precisos.

4.3.1. Introdução ao software

Feita a apresentação dos cálculos necessários para equacionar a movimentação de um ROV em um meio aquático, serão apresentados aqui os módulos e submódulos referentes ao bloco de Modelagem do simulador desenvolvido neste trabalho. Cada módulo possui funções distintas e as interações entre eles podem ser observadas na figura 4.2. A figura a seguir mostra a estrutura do bloco de modelagem implementado em Simulink.

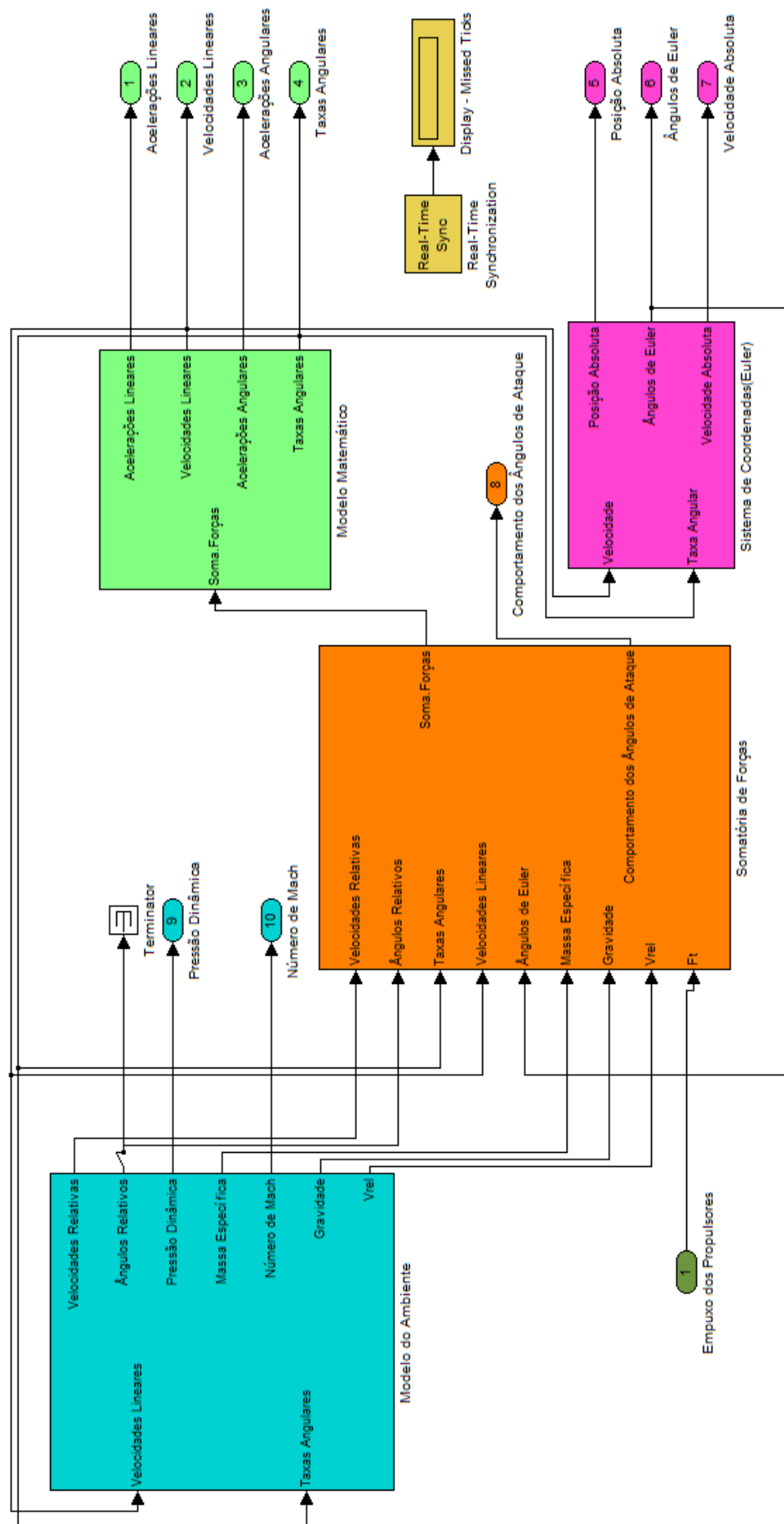


Figura 4.2 - Bloco de Modelagem

4.3.1.1. Módulo “Modelo Matemático”

Neste módulo são calculadas as acelerações angulares e lineares. Dada a matriz de inércia e recebidos do módulo “Somatória de Forças” os valores das forças atuantes no veículo. Os cálculos são feitos para as acelerações no referencial inercial, que posteriormente são numericamente integradas para se determinar as velocidades lineares e angulares. A figura abaixo mostra o conteúdo do módulo “Modelo Matemático”.

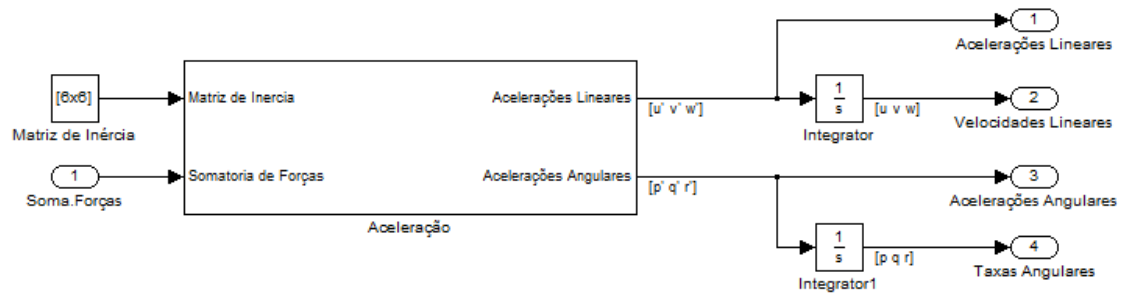


Figura 4.3 - Modelagem do Módulo “Modelo Matemático”

4.3.1.2. Módulo “Sistema de Coordenadas (Euler)”

Este módulo recebe as informações de velocidade calculadas no módulo “Modelo Matemático” e as transforma em informações absolutas. A velocidade absoluta é a que realmente importa para fins de avaliação de resposta do veículo. Integrando este valor, encontra-se a posição absoluta do veículo também.

Além disso, este módulo também é responsável por devolver os ângulos de Euler referentes às taxas angulares também calculadas no módulo “Modelo Matemático”, ângulos estes utilizados em outras contas do simulador. Vale a pena notar que no cálculo dos ângulos de Euler há um passo de atraso em relação às outras variáveis, no entanto, para a frequência utilizada para simulação, este erro é desprezível. A figura abaixo mostra o conteúdo do módulo “Sistema de Coordenadas (Euler)”.

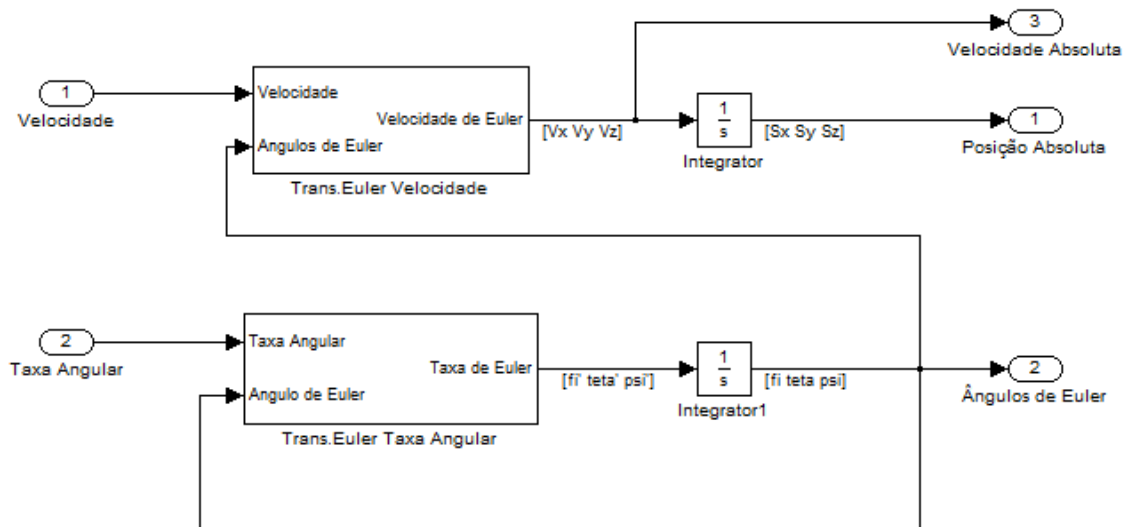


Figura 4.4 - Modelagem do Módulo “Sistema de Coordenadas (Euler)”

4.3.1.3. Módulo “Modelo do Ambiente”

Este módulo é responsável pela modelagem do ambiente aquático no qual o ROV está submetido e pelo cálculo das velocidades relativas do veículo, que são a relação entre as velocidades do veículo e as correntes do ambiente.

Para tal são definidas características físicas do fluido como velocidade do som no meio e massa específica, ambas consideradas constantes para este trabalho. O valor da aceleração gravitacional também foi definido neste módulo com valor constante.

Neste módulo existe um submódulo chamado “Modelo de Correnteza”, responsável pela modelagem mais adequada do tipo de correnteza considerada. Neste trabalho, inicialmente, não se considera nenhuma corrente, para simplificação. Porém futuramente essa modelagem será essencial para o desenvolvimento de um controle robusto e eficiente, por isso a necessidade de um submódulo, pois facilita a alteração do tipo de modelagem. Para efeito de validação através de [1], é dado que não há velocidade no fluido que contém o ROV, portanto não há correnteza modelada neste primeiro momento.

Há ainda nesse módulo o cálculo da pressão dinâmica e do número de Mach, ambos interessantes para avaliação dos esforços hidrodinâmicos aos quais o ROV está submetido. A figura abaixo mostra o conteúdo do módulo “Modelo do Ambiente”.

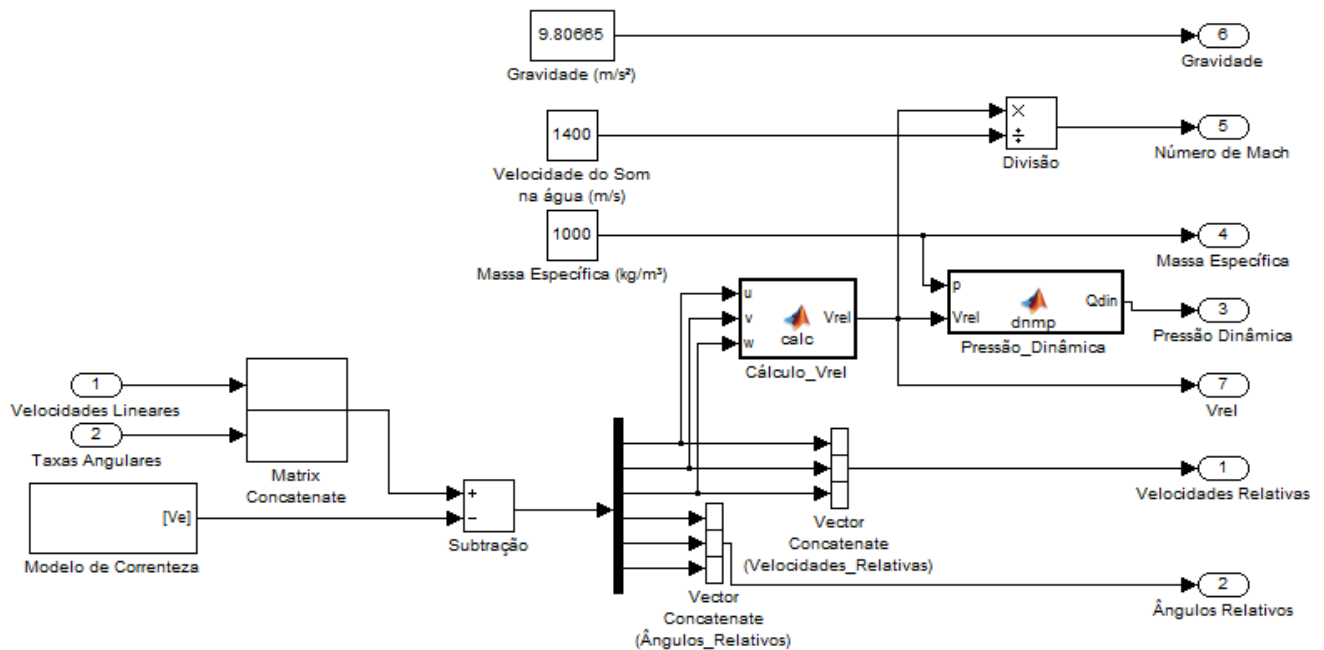


Figura 4.5 - Modelagem do Módulo “Modelo do Ambiente”

4.3.1.4. Módulo “Somatória de Forças”

Este módulo recebe as informações necessárias dos outros módulos e é responsável pelo cálculo de todas as forças contidas na equação fundamental de movimento do ROV, calculando assim a somatória das forças agentes sobre o veículo.

Os cálculos das forças foram separados em 3 diferentes localidades, sendo cada uma delas referentes ao cálculo de uma força em específico. Essa implementação foi feita utilizando um bloco do tipo Matlab Function, que permite inserir uma programação em código Matlab dentro dos blocos do Simulink. Isso pode ser observado na figura 4.6 que mostra o conteúdo do módulo “Somatória de Forças”.



A função “Cálculo das Forças Inercial, Gravitacional e de Empuxo” como o nome já diz é responsável pelo cálculo das forças mencionadas. No caso a saída desta função é o resultado da somatória:

$$F_1 = -F_I + F_G + F_B$$

A outra função chamada “Acoplamento de Movimento” é a função responsável por receber os valores de empuxo de cada um dos propulsores e alocá-los no veículo conforme a disposição geométrica dos propulsores no veículo, resultando nas forças nos seis graus de liberdade do veículo. Esse acoplamento será melhor explicado a diante, no capítulo referente ao projeto do controlador.

Por último o subsistema chamado “Cálculo Força Hidrodinâmica” que como o nome diz, é responsável pelo cálculo da força hidrodinâmica vigente sobre o veículo. Este subsistema foi necessário devido a dificuldade de realizar uma modelagem eficiente para o cálculo dos coeficientes de arrasto atuantes sobre o veículo. Os coeficientes utilizados neste trabalho estão contidos em gráficos em [1] e são referentes ao veículo Ishidera. Como os valores contidos nos gráficos não foram dados, primeiramente foi feita uma aproximação numérica dos dados, gerando um polinômio de oitavo grau que correspondia aos valores, porém essa aproximação apresentou um comportamento insatisfatório e para que os resultados do cálculo dos coeficientes hidrodinâmicos fossem confiáveis foi necessário inserir na modelagem tabelas com os valores medidos nos gráficos contidos em [1]. Com estas tabelas, entra-se com os valores dos ângulos de ataque do veículo (attack angle, side slip angle e side attack angle) e o algoritmo interpola os valores contidos nessa tabela para convergir para o ótimo valor do coeficiente. Esse tipo de abordagem apresentou resultados muito satisfatórios, porém introduziu restrições aos movimentos do veículo. Há um algoritmo de extrapolação de valores baseado nos valores tabelados, porém nem sempre esses valores são confiáveis. Entretanto o *range* de graus tabelados são suficientes para atender a quase totalidade dos movimentos que um ROV executa, sendo apenas movimentos agressivos como composições de velocidades com altos valores problemáticas no simulador. A figura 4.7 mostra a modelagem do software contendo as tabelas e a estruturação do calculo das força hidrodinâmica.

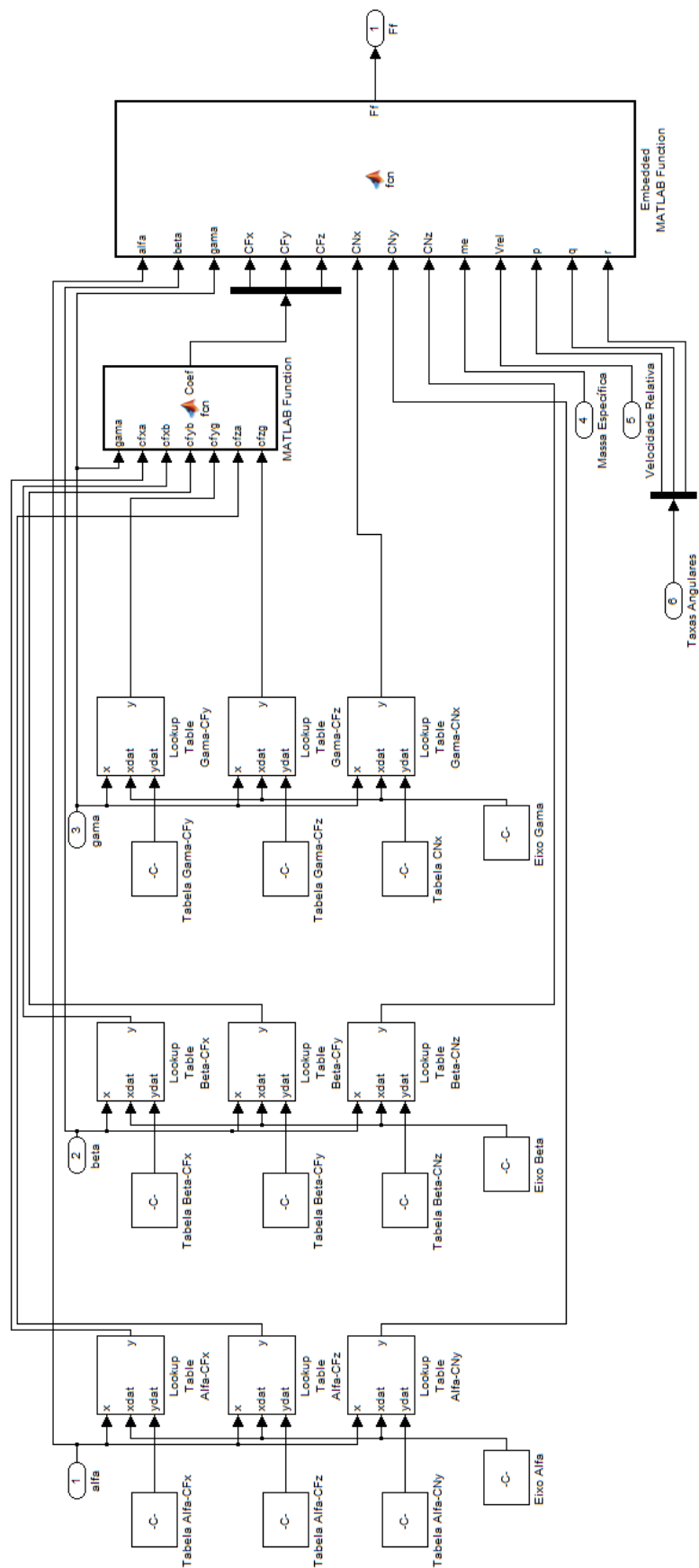


Figura 4.7 - Modelagem do Cálculo da Força Hidrodinâmica

4.4. Bloco Interface de Resultados

Este bloco se torna extremamente necessário e útil como um meio de auxiliar no processo de ‘debug’ dos algoritmos implementados e auxiliar na visualização dos dados de saída do sistema, tanto durante quanto após o término da execução da simulação.

A visualização durante a simulação permite verificar se o controlador ou o veículo estão tendo o comportamento esperado e para isso foi desenvolvido um sistema de plotagem em função do tempo através de ‘Scopes’ no Simulink, funções estas que permitem a visualização de gráficos do comportamento das variáveis ao longo do tempo e foi feito também um programa para visualização 3D do veículo.

A visualização por plotagem em função do tempo é o tipo mais simples, mas também a mais esclarecedora, pois através dela observa-se a variação de um ou mais parâmetros no tempo. A implementação destas funções gráficas é simples e podem ser adicionados quantos gráficos forem necessários para a monitoração do sistema simulado.

O software de visualização 3D do veículo foi desenvolvido em linguagem VRML (Virtual Reality Modeling Language) através do software V-Realm Builder, software oferecido pela própria Mathworks que, apesar de oferecer um ambiente gráfico simples, oferece uma confiabilidade boa com relação à navegação do veículo. Essa visualização 3D se faz necessária porque mesmo que o acompanhamento gráfico dos parâmetros em função do tempo seja eficiente, ainda não é possível visualizar facilmente todas as variáveis em conjunto atuando no veículo em navegação. Foi criado um ambiente virtual aonde é possível acompanhar a orientação do veículo e sua movimentação em seus 6 graus de liberdade através de uma representação fiel aos movimentos calculados pelo simulador.

Na figura a seguir é apresentada a interface de visualização 3D.

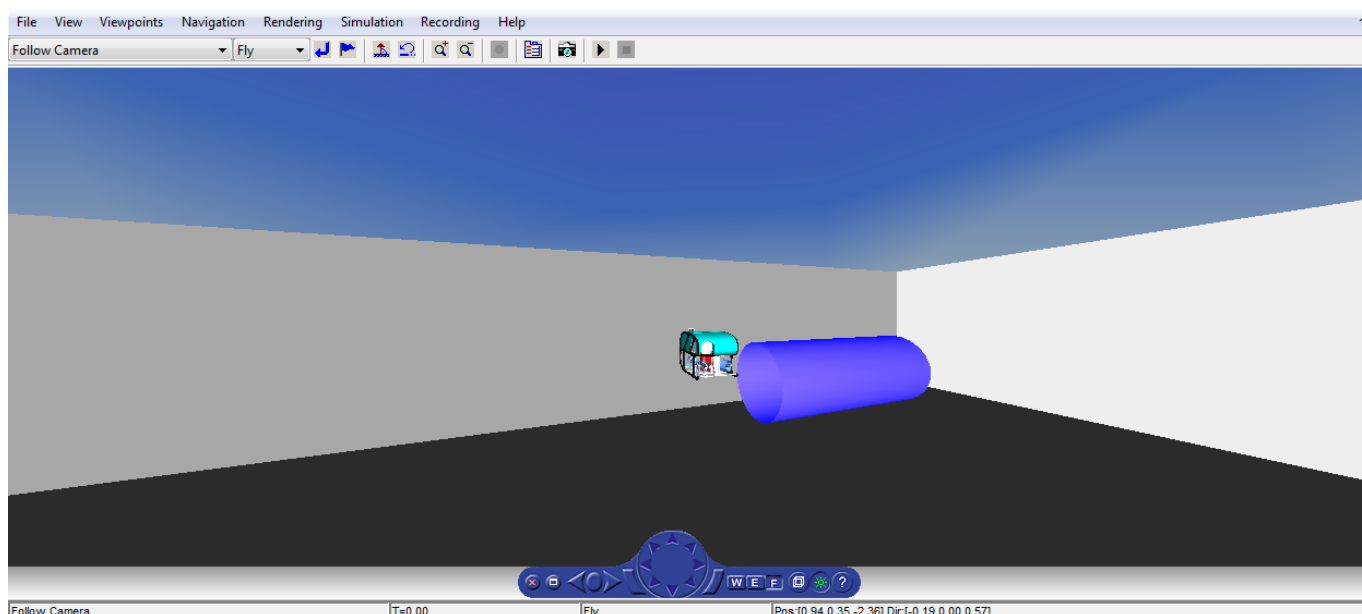


Figura 4.8 - Ambiente Virtual 3D

4.5. Bloco de Comunicação

Para desenvolver este bloco, primeiro foi preciso definir como estão divididas as funções em cada bloco do software Simulador e em qual dos dispositivos de hardware disponíveis cada um dos softwares desenvolvidos seriam alocados. O diagrama a seguir confere a noção necessária para esse entendimento.

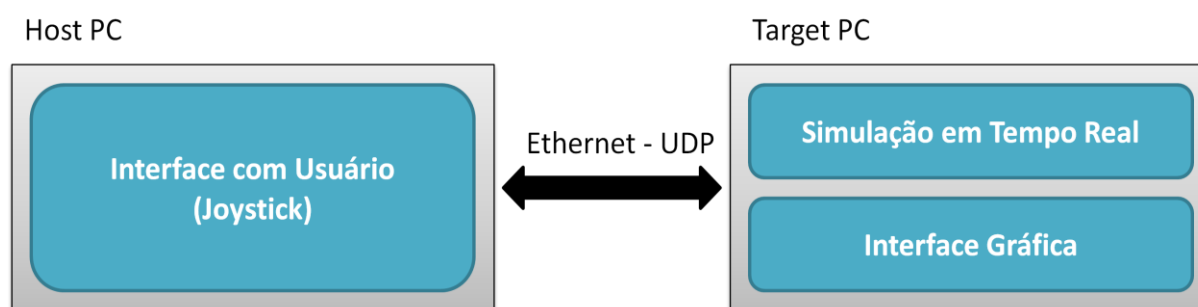


Figura 4.9 - Diagrama de distribuição de tarefas por PC

O bloco de comunicação é responsável pelo envio e recepção de dados entre o hardware que está executando a simulação em tempo real e o hardware responsável pela geração de sinais de controle. Essa implementação consiste em configurar máscaras e portas de rede entre os dois computadores envolvidos no simulador. O Host PC possui um protocolo escrito em linguagem C++ para a realização da comunicação. Já o Target PC que é responsável pela simulação em tempo real, feita em ambiente Simulink, possui

a configuração da comunicação através de “blocos de comunicação” contidos na biblioteca do próprio Simulink que devem ser configurados com as mesmas informações de envio e recebimento contidas no protocolo escrito no programa em C++.

As figuras 4.10 e 4.11 mostram o hardware utilizado neste trabalho. Um detalhe que mostra uma das vantagens do protocolo de comunicação implementado, o protocolo ethernet UDP, é a distância que os computadores se encontram, sendo sua comunicação feita através de um roteador no laboratório, ou seja, essa comunicação permite que os computadores que a compõe possam estar a grandes distâncias.

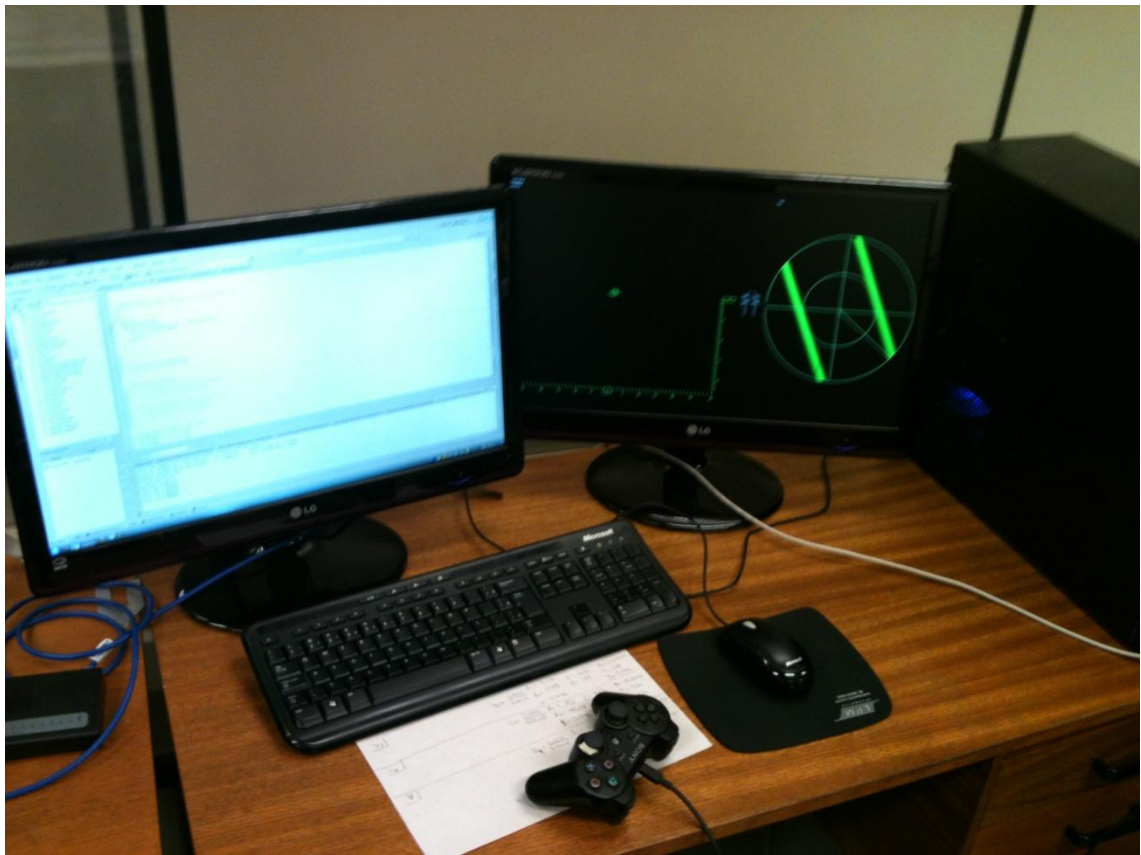


Figura 4.10 - Host PC com o programa em execução



Figura 4.11 - Target PC com o software de simulação em execução

4.5.1. Real-Time Windows Target

O ambiente oferecido pelo Real-Time Windows Target oferece ao Simulink a função de simular em tempo real seus modelos em um PC com ambiente Microsoft Windows, oferece também ‘blocos’ que podem ser conectados a diversos tipos de dispositivos de I/O. Esse ambiente permite que o usuário crie e controle sistemas em tempo real para prototipagem rápida e mesmo sistemas hardware-in-the-loop. O Real-Time Windows Target oferece diversos tipos de protocolos de comunicação para serem implementados, mas considerando que serão utilizados dois computadores, somente dois protocolos foram considerados opções: a comunicação serial RS232 e a comunicação ethernet UDP (User Datagram Protocol). Apesar das duas soluções terem suas vantagens, o protocolo que será utilizado é o ethernet UDP, pelos motivos apresentados a seguir:

- Alta velocidade de transferência: a conexão ethernet possui uma velocidade máxima de transferência de dados muito mais rápida do que a da comunicação serial, sendo da ordem de 100 Mbps. Este fato ajuda consideravelmente sistemas

em tempo real, pois evita atrasos no envio e recebimento de dados e nem limita a transferência de dados.

- Grandes distâncias de comunicação: é possível utilizar gateways e repetidores para aumentar a distância entre os dois desktops utilizados, sendo possível utilizar uma rede privada ou a internet para comunicação.
- A comunicação ethernet utilizada é a básica por UDP, onde os bits são empacotados e enviados manualmente aos outros blocos, sendo posteriormente desempacotados e lidos.

4.5.2. Protocolo de Comunicação ethernet UDP em C++ e em Simulink/MatLab (UDP/IP)

Para realizar a comunicação entre o Host PC e o Target PC foi escolhido implementar o protocolo ethernet. A camada de transporte possui dois protocolos que são o UDP (User Datagram Protocol) e TCP (Transmission Control Protocol). O protocolo UDP realiza apenas a multiplexação para que várias aplicações possam acessar o sistema de comunicação de forma coerente, já o protocolo TCP realiza, além da multiplexação, uma série de funções para tornar a comunicação entre origem e destino mais confiável, são responsabilidades do protocolo TCP: o controle de fluxo, o controle de erro, o sequenciamento e a multiplexação de mensagens. Há aplicações em que é preferível entregar os dados o mais rapidamente possível, mesmo que algumas informações se percam no caminho. É o caso, por exemplo, das transmissões de vídeo pela internet (streaming), onde a perda de um pacote de dados não interromperá a transmissão. O mesmo ocorre neste caso, pois o envio de dados do simulador é feito através de variações no direcional do joystick, porém caso haja alguma perda de dados associada à transmissão, essa perda não terá um grande impacto associado aos movimentos do veículo no simulador, pois a velocidade em que se alteram as informações no direcional analógico do joystick é muito inferior à velocidade de transmissão de dados do software de aquisição, portanto há uma taxa de repetição de dados muito alta, atenuando a perda de eventuais pacotes de dados enviados ao simulador.

A implementação do protocolo de comunicação no programa de interface do usuário foi feita em C++, a comunicação foi feita através de um socket(interface de programação de aplicação (API), que fornece a sintaxe através da qual os serviços

oferecidos por um protocolo podem ser chamados em um Sistema Operacional). Na programação das interfaces socket, o modelo utilizado é o cliente-servidor, neste modelo, o servidor faz uma abertura passiva, esperando conexões de um cliente, e o cliente faz a abertura ativa da conexão. No caso deste trabalho, o programa desenvolvido em C++ é o servidor e o problema desenvolvido em Matlab é o cliente. A abertura passiva consiste na criação de um socket e na preparação do servidor para uma conexão, sem efetivamente estabelecê-la. A porta usada pelo servidor deve ter um número conhecido para que o cliente saiba onde tentar a conexão, no caso foi utilizado a porta '20000'. As informações necessárias são fornecidas através de uma estrutura de endereçamento. Depois de criado o socket, o mesmo é associado a uma porta UDP, o que é feito pela função bind(). O UDP, por ser um protocolo não orientado à conexão, como o próprio nome diz, não estabelece uma conexão antes de enviar os dados: quando a aplicação tem informação a transmitir, ela simplesmente a envia. Logo, o programa UDP só fará uso das funções recvfrom() e sendto() (que anexa um cabeçalho contendo o endereço IP do destino e o número da porta aos dados), lembrando que ambos os computadores tem IP fixos e esse endereço por não variar pode ser escrito previamente na rotina de comunicação.

O protocolo foi implementado no Bloco de Comunicação do simulador, no caso o Simulink possui um bloco de comunicação UDP onde só é preciso configurar as mesmas informações de porta e IP que o Host PC, assim quando o Host PC enviar um sinal para iniciar a comunicação, o simulador recebe e envia um valor para inicializar a comunicação entre eles.

4.6. Homologação do Simulador – Resultados

Através da modelagem apresentada e do software implementado em Simulink, foram configurados os dados do ROV Ishidera[1] para conferir se os resultados obtidos são compatíveis com os resultados apresentados em [1], em caso afirmativo pode-se afirmar que o simulador está modelado de forma correta, podendo assim ser modificado para os dados de outro ROV tendo a garantia de que os resultados apresentados são confiáveis e compatíveis com a realidade.

Serão apresentados a seguir os gráficos referência fornecidos em [1] que foram gerados por uma simulação e garantidos através de testes em tanque.

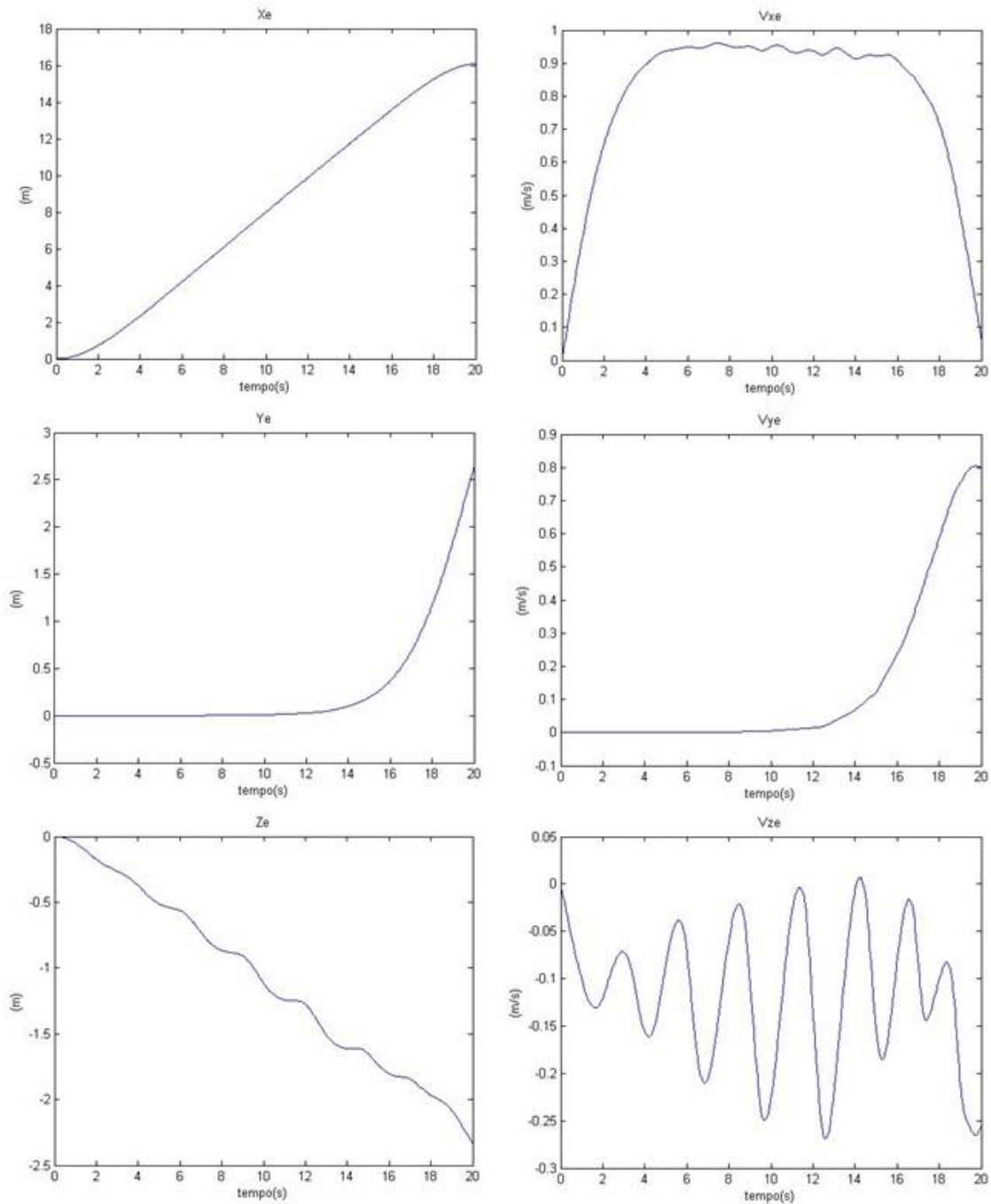


Figura 4.12 - Gráficos Referência do ROV em movimento no meio aquático

Os gráficos apresentados são referentes às posições absolutas do ROV no meio aquático (coordenadas X_e , Y_e , Z_e) e às velocidades absolutas do ROV (componentes V_{xe} , V_{ye} , V_{ze}). Apresentados os gráficos de referência, serão apresentados a seguir os gráficos obtidos com a simulação do modelo desenvolvido neste projeto.

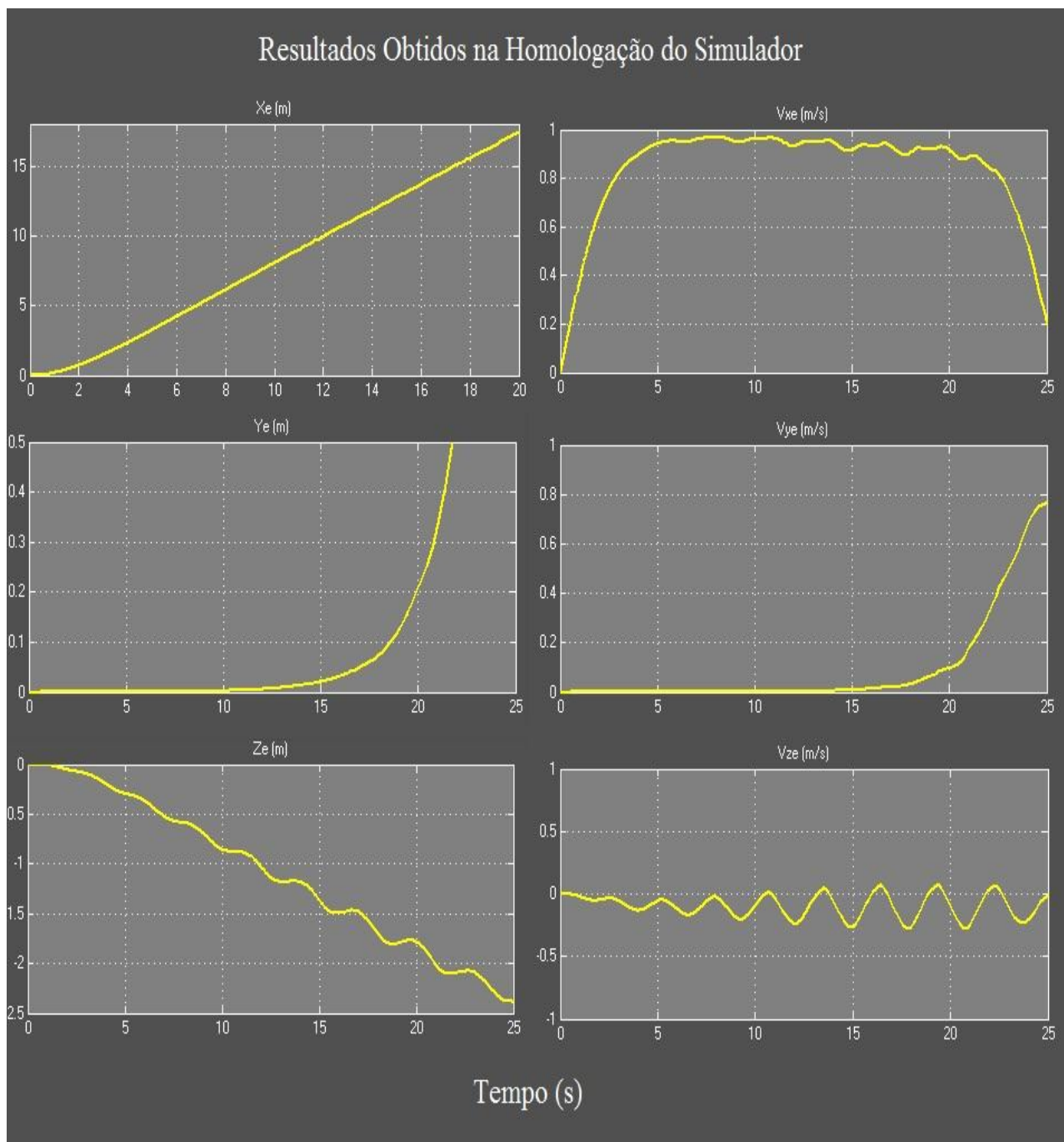


Figura 4.13 - Gráficos Obtidos do ROV através da simulação do modelo apresentado

Os gráficos apresentados acima são os gráficos obtidos através das funções Scope contidas no simulador, apresentados na mesma ordem que os gráficos referência para facilitar a comparação dos resultados.

4.6.1. Análise dos Resultados

Através da análise dos conjuntos de gráficos apresentados, pode-se concluir que o modelo desenvolvido neste trabalho simula corretamente o comportamento de um

veículo ROV em movimentação num meio aquático. Pode-se concluir isto comparando os gráficos referenciais contidos em [1] e os gráficos obtidos através da simulação do modelo.

Foi verificado também o efeito do tamanho do Step de simulação nos resultados obtidos, sendo este um parâmetro de simulação fundamental para uma boa resolução dos resultados. Os gráficos referentes a Vze não estão a primeira vista muito parecidos devido a diferenças na escala do eixo y, mas se alterada essa escala verifica-se que ambos possuem o mesmo comportamento.

Portanto através destes resultados podemos garantir que em futuras simulações, os dados obtidos das velocidades e das posições absolutas são confiáveis e reais, podendo ser utilizados para tomadas de decisão no projeto do controlador a ser desenvolvido.

5. Projeto do Controlador

Nesta Seção serão descritas todas as etapas do projeto do sistema de controle para o ROV, desenvolvido para inspecionar dutos submersos. O sistema de navegação adotado necessita de quatro malhas de controle: a primeira para o controle da velocidade de movimento em *surge*, a segunda para o controle da velocidade de movimento em *sway*, a terceira para o controle da profundidade e a quarta para o controle do rumo. Estando todos os controladores habilitados, o operador realizará o controle manual do veículo para posicionamento e inspeção no duto.

Inicialmente, nesta seção, são apresentados os conceitos guias para o projeto e implementação do sistema de controle desejado. Em seguida, é feita uma análise das não linearidades associadas à modelagem de um ROV e como as mesmas foram tratadas neste trabalho. Apresentados estes conceitos iniciais, é descrito o mecanismo para realizar o controle da dinâmica do ROV. Mecanismo composto pelo controlador, pelo desacoplamento dos quatro graus de liberdade (*surge*, *sway*, *heave* e *yaw*) do ROV e da dinâmica em si, mostrando a malha a ser controlada.

5.1. Características desejáveis do controlador projetado

5.1.1. Controle robusto

Controle robusto é um ramo da teoria de controle que explicitamente lida com a incerteza na sua abordagem para projeto do controlador. Métodos de controle robustos são projetados para funcionar adequadamente, enquanto parâmetros incertos ou distúrbios estão dentro de um conjunto (normalmente compacto). Métodos robustos visam alcançar desempenho robusto e / ou estabilidade na presença de erros de modelagem limitados. Em contraste com uma política de controle adaptativo, uma política de controle robusto é estático, em vez de adaptar a medições de variações, o controlador é projetado para trabalhar assumindo que certas variáveis serão desconhecidas, mas, por exemplo, delimitadas. Este tópico também não foi estudado profundamente ainda.

5.1.2. Controle em tempo real

De acordo com a definição dada pelo IEEE POSIX Standard (Portable Operation system Interface for Computer Environments) “um sistema de tempo real é aquele em que a correção de um resultado não depende apenas da correção lógica através do cálculo, mas também do momento em que o resultado se torna disponível”. Daí interpreta-se que o tempo é uma das entidades mais importantes desse sistema e que há restrições de tempo associadas às tarefas do sistema, onde essas tarefas devem controlar ou reagir a eventos que estão acontecendo no “mundo real”, ou seja, em “tempo real”.

Deve notar-se que computação em tempo real não é equivalente a computação rápida. Computação rápida visa obter os resultados o mais rapidamente possível, enquanto computação em tempo real visa obter os resultados em um prazo determinado de tempo dentro de definidas tolerâncias de tempo. Assim, um prazo pode ser associado a uma tarefa que tem que satisfazer esta restrição de tempo especificando o início ou o seu tempo de conclusão. Se a tarefa tem que cumprir o prazo, porque de outra forma irá causar erros fatais ou consequências indesejáveis, a tarefa é chamada de “hard real time task”. Do contrário, se o cumprimento do prazo é desejável, mas não obrigatório, a tarefa é chamada de “soft real time task”. Por extensão podemos classificar restrições também como “hard/soft time constraints. [12]

5.2. Estudo de caso – Não linearidades do Projeto

Um sistema não linear apresenta um comportamento muito mais rico e complexo que os sistemas lineares, que tem como principal característica a relação linear entrada e saída. Fato esse que é observado ao se aplicar uma variação na entrada e essa variação é notada proporcionalmente em sua saída (linearidade e superposição), porém um sistema dito não linear não só viola essa propriedade, mas pode também possuir outras características particulares que se comportam como não linearidades, tornando-o não linear:

- Linearidade e Superposição: a soma dos efeitos de duas entradas não coincide com o efeito da soma destas entradas.
- Múltiplos pontos de equilíbrio: Um ponto de equilíbrio localmente assintoticamente estável de um sistema linear é globalmente assintoticamente estável. Pontos de equilíbrio de sistemas não lineares podem ser localmente assintoticamente estáveis, mas não serem globalmente assintoticamente estáveis.
- Sistemas lineares assintoticamente estáveis são necessariamente estáveis BIBO (ou entrada-saída). Em outras palavras, entradas limitadas provocam sempre saídas limitadas. Uma entrada ou distúrbio limitado pode até instabilizar um sistema não linear, mesmo que ele seja globalmente assintoticamente instável.
- O estado de um sistema não linear nunca diverge em tempo finito, pois as soluções possuem crescimento (ou decrescimento) exponencial. O módulo $kx(t)$ do estado de um sistema não linear pode tender a infinito em tempo finito.
- Um sistema não linear pode não possuir a propriedade de unicidade de suas soluções. Em outras palavras, para uma condição inicial fixa, podem existir duas ou mais soluções da equação passando por esta condição inicial.
- Caos: O comportamento caótico que pode ocorrer em sistemas não lineares nunca ocorre em sistemas lineares.

Se o sistema apresentar um ou vários desses comportamentos ele é dito um sistema não linear.

5.2.1. Não Linearidades associadas à modelagem de um ROV

O comportamento dinâmico dos veículos submarinos é, essencialmente, não linear, como em muitos dos casos que a engenharia aborda. Muitas vezes essas não linearidades são suficientemente complexas para que não haja uma solução exata e as

mesmas devem ser solucionadas por algoritmos de aproximação. Nesse trabalho a modelagem apresentada anteriormente não aborda problemas de não linearidades associados aos esforços modelados e há também esforços não lineares cuja modelagem não foi sequer abordada.

Dentre os fatores não lineares envolvidos estão a modelagem do cabo umbilical (cabo que conecta o veículo à base de operação), a dinâmica dos propulsores (saturação dos mesmos, dinâmica do motor CC, característica tensão x empuxo quadrática e zona morta), fatores não lineares da dissipação hidrodinâmica, entre outros.

Destes mencionados, apenas um foi tratado nesta modelagem, que é a dissipação hidrodinâmica, calculada com os termos quadráticos das velocidades na formulação da força de esforços hidrodinâmicos. A modelagem dos propulsores, como dito anteriormente, não foi abordada, pois os dados necessários para esta modelagem não estão disponíveis e estimá-los através de ensaios está fora do escopo deste trabalho. Para compensar este erro atribuído à não linearidade do propulsor, atribui-se essa incerteza ao controlador projetado aumentando a robustez do sistema modelado.

A modelagem do cabo umbilical também não foi abordada neste trabalho, também por ser de grande complexidade e demandar um tempo de cálculo e implementação muito grande para o escopo do projeto. Tal modelagem foi bem explicada em [17] e pode ser aplicada para uma modelagem mais precisa e confiável. Esta modelagem é realmente importante para um veículo do tipo ROV quando submetido a um ambiente com correnteza, mas como o ROV deste trabalho entrará neste ambiente com uma gaiola protetora rígida, o efeito da força pelo cabo umbilical fica consideravelmente diminuído, podendo ser compensado através da robustez do controlador projetado.

5.3. Análise da Dinâmica do ROV

O ROV é um sistema multivariável com quatro entradas dadas pelo Joystick (isto é, comandos para os quatro propulsores) e quatro saídas (graus de liberdade que podem ser controlados). O controlador atua entre os sinais e referência enviados pelo Joystick e o mecanismo de desacoplamento de variáveis. O mecanismo de desacoplamento irá permitir realizar o controle independente de cada grau de liberdade. Em outras palavras, o sistema multivariável pode ser visto como um conjunto de quatro sistemas monovariáveis. Considerações adicionais permitem obter modelos linearizados simplificados para a dinâmica de cada um dos graus de liberdade de interesse.

A malha de controle mencionada acima pode ser vista no sistema ilustrado na figura 5.1 a seguir.

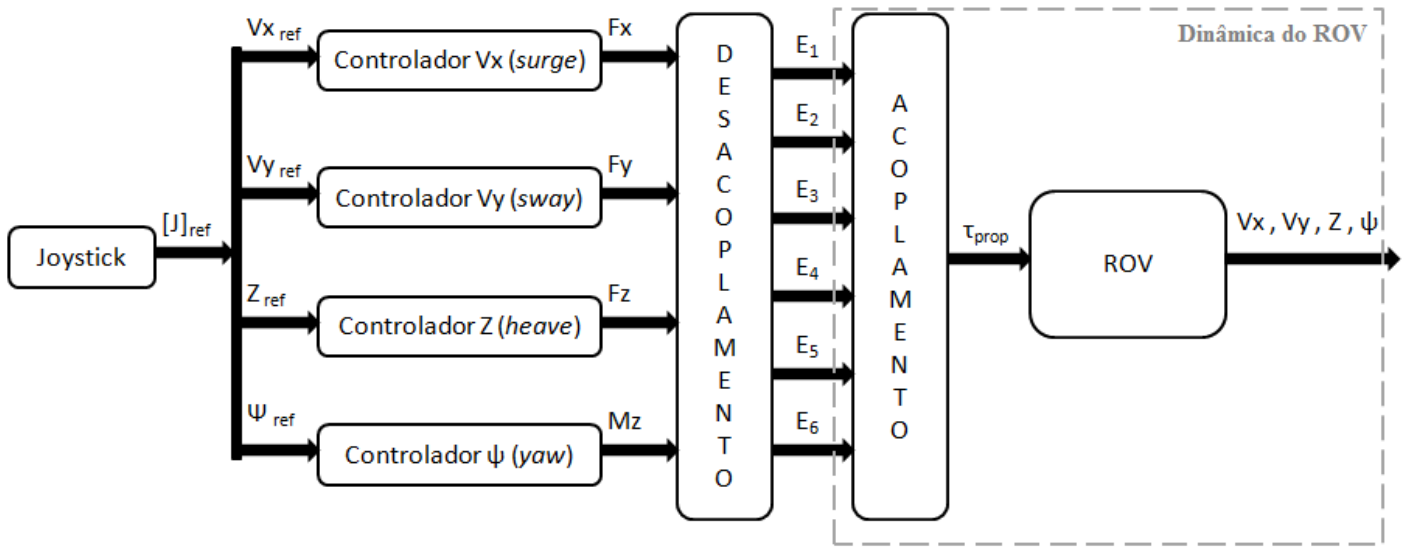


Figura 5.1 - Malha do Sistema de controle e dinâmica do ROV

- Desacoplamento

Como explicado no Capítulo 3, a dinâmica do ROV é bastante acoplada. Para realizar movimentos translacionais ou rotacionais em torno de um dos seus eixos, é necessário em geral que um conjunto de propulsores atue simultaneamente. Por exemplo, devido à configuração dos propulsores do ROV, para execução de um movimento translacional em x é necessário acionar os propulsores 1, 2, 3 e 4.

Com o mecanismo de desacoplamento completo, é possível projetar independentemente um controlador monovariável para cada grau de liberdade no qual se deseja atuar [20], isto é, o sinal de entrada do i-ésimo grau de liberdade atuará somente na dinâmica correspondente.

No figura 5.1 apresentam-se os sinais de entrada e saída correspondentes a essa função no diagrama, onde:

- F_x, F_y, F_z, M_z : são as forças de controle desejadas para execução no veículo.
- E_i : são os empuxos desejados em cada i-ésimo propulsor existente no veículo.

- Acoplamento

No diagrama apresentado em 5.1, o bloco de acoplamento é responsável por situar as forças de empuxo enviadas pela matriz de desacoplamento no veículo, resultando na real dinâmica do veículo, uma vez que o posicionamento da força de empuxo no veículo pode causar momentos e forças resultantes desse posicionamento projetado.

5.3.1. Modelo simplificado para a dinâmica do ROV

Foi apresentado neste trabalho a modelagem de um ROV em um meio aquático, todas as considerações utilizadas e todas as descon siderações que foram feitas para este modelo que foi utilizado para a criação de um simulador suficientemente confiável. Porém, com toda a matemática envolvida e todas as aproximações feitas, fica muito complexa a determinação de uma função exata do sistema dinâmico de um ROV e essa função é de vital importância para a análise do controle a ser implementado.

Para que fosse possível chegar a uma função de transferência próxima a do sistema em estudo, foram impostas entradas do tipo degrau em cada um dos graus de liberdade a serem controlados, separadamente. Através das formulações matemáticas e teoremas de controle contidos nas bibliografias [21] e [22] foi possível observar pelo comportamento gráfico das variáveis a serem controladas um comportamento de um sistema de primeira ordem e, através dos valores contidos em cada um dos gráficos simulados, é possível determinar um modelo simplificado da função de transferência suficientemente próximo para um estudo mais detalhado.

Para tal foi utilizada a seguinte formulação:

$$G(s) = \frac{K_{planta}}{\tau s + 1}$$

Onde:

- K_{planta} é o ganho associado à planta para a resposta do sistema a uma entrada do tipo degrau imposta a ele.
- τ é a constante de tempo associada à resposta do sistema a uma entrada degrau imposta a ele.

Para a determinação de tais valores utilizam-se as formulações a seguir, para melhor entendimento de tais formulações é apresentado o diagrama a seguir.

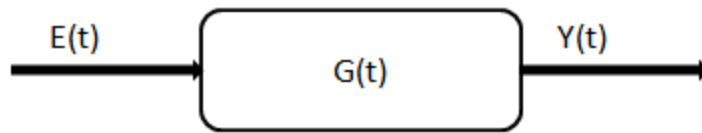


Figura 5.2 - Diagrama de blocos

$$K_{planta} = \frac{\lim_{t \rightarrow \infty} Y(t)}{E(t)}$$

$$\tau = 0,632 \lim_{t \rightarrow \infty} Y(t)$$

Os gráficos de resposta para uma entrada do tipo degrau para cada um dos quatro graus de liberdade separadamente são apresentados a seguir com seus respectivos valores de K_{planta} e τ e a função de transferência aproximada associada.

5.3.1.1. Velocidade em x (surge)

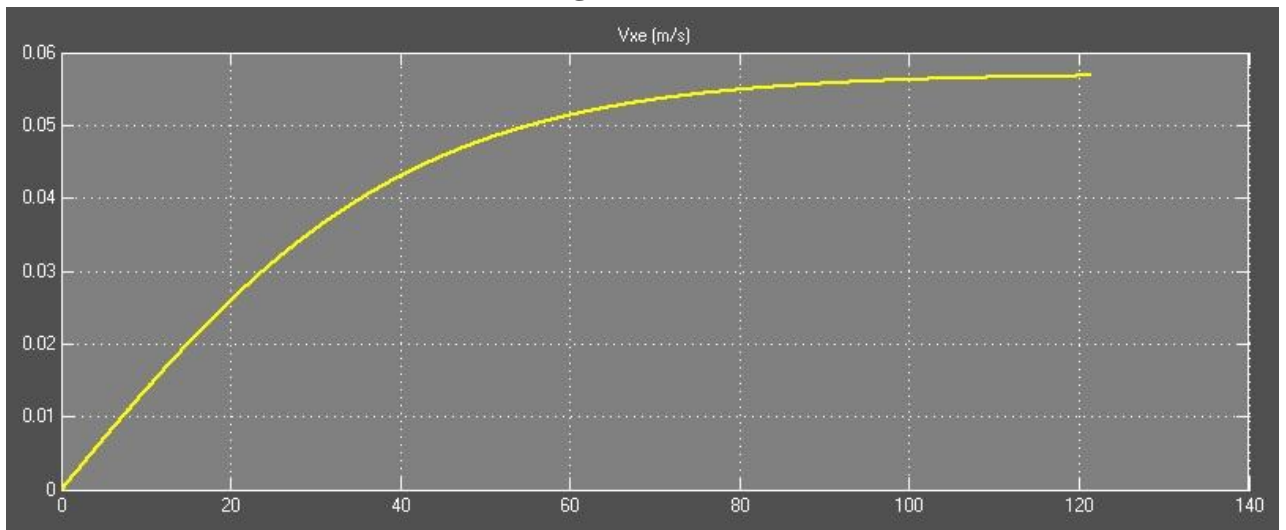


Figura 5.3 - Gráfico da resposta em Vx a um degrau de amplitude 0,5 m/s

Para o gráfico acima, é possível identificar os seguintes valores:

$$K_{planta} = 0,1136 \text{ m/s}$$

$$\tau = 30,15 \text{ s}$$

Para tais valores chega-se à função de transferência aproximada:

$$G(s) = \frac{0,1136}{30,15s + 1}$$

5.3.1.2. Velocidade em y (sway)



Figura 5.4 - Gráfico da resposta em Vy a um degrau de amplitude 0,5 m/s

Para o gráfico acima, é possível identificar os seguintes valores:

$$K_{\text{planta}} = 0,1426 \text{ m/s}$$

$$\tau = 21,48 \text{ s}$$

Para tais valores chega-se à função de transferência aproximada:

$$G(s) = \frac{0,1426}{21,48s + 1}$$

5.3.1.3. Posição em z (heave)

Para a identificação da função de transferência aproximada da posição foi primeiramente estimada a função de transferência para a velocidade, cujo comportamento se assemelha ao de uma função de primeira ordem. Para obter a função de transferência para a posição basta colocar um integrador na função.

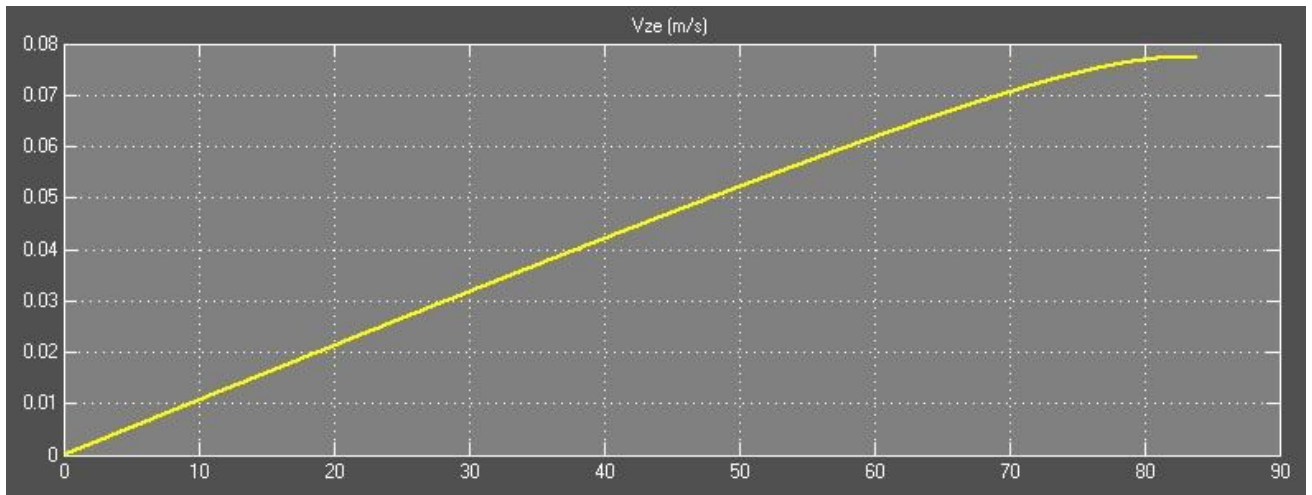


Figura 5.5 - Gráfico da resposta em Vz a um degrau de amplitude 0,5 m/s

Para o gráfico acima, é possível identificar os seguintes valores:

$$K_{planta} = 0,1546 \text{ m/s}$$

$$\tau = 46,68 \text{ s}$$

Para tais valores chega-se à função de transferência aproximada:

$$G(s) = \frac{0,1546}{s(46,68s + 1)}$$

5.3.1.4. Velocidade em ψ (yaw)

Para a identificação da função de transferência aproximada da posição foi primeiramente estimada a função de transferência para a velocidade, cujo comportamento se assemelha ao de uma função de primeira ordem. Para obter a função de transferência para a posição basta colocar um integrador na função.

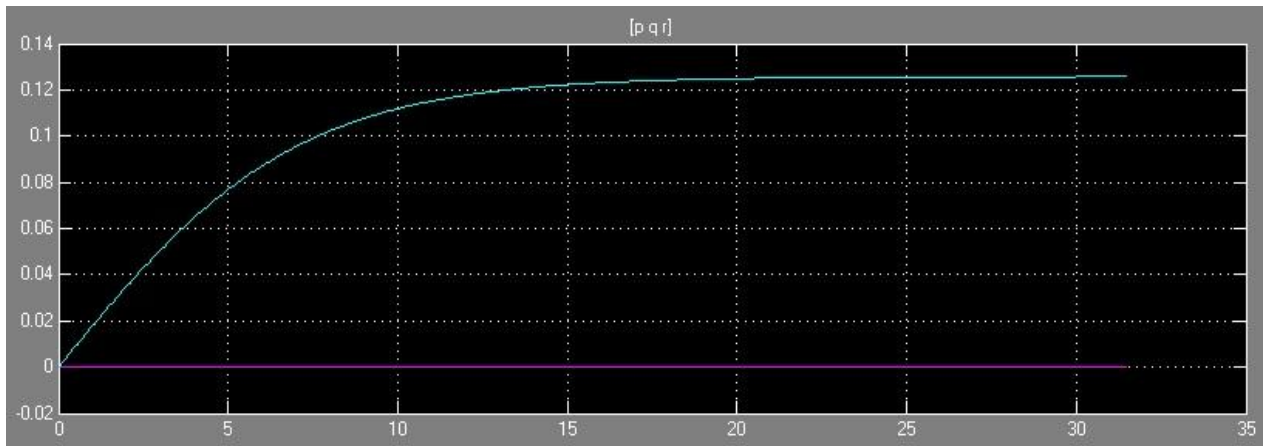


Figura 5.6 - Gráfico da resposta em $V\psi$ a um degrau de amplitude 0,5 m/s

Para o gráfico acima, é possível identificar os seguintes valores:

$$K_{\text{planta}} = 0,2512 \text{ m/s}$$

$$\tau = 5,22 \text{ s}$$

Para tais valores chega-se à função de transferência aproximada:

$$G(s) = \frac{0,2512}{s(5,22s + 1)}$$

5.3.2. Testes comparativos com o modelo simplificado

Com o objetivo de validar o modelo simplificado, foram realizados testes comparativos com o modelo completo. Para estes testes, ambos os modelos de dinâmica de ROVs foram implementados e excitados por sinais tipo degrau com a amplitude de 0,5 m/s para cada grau de liberdade.

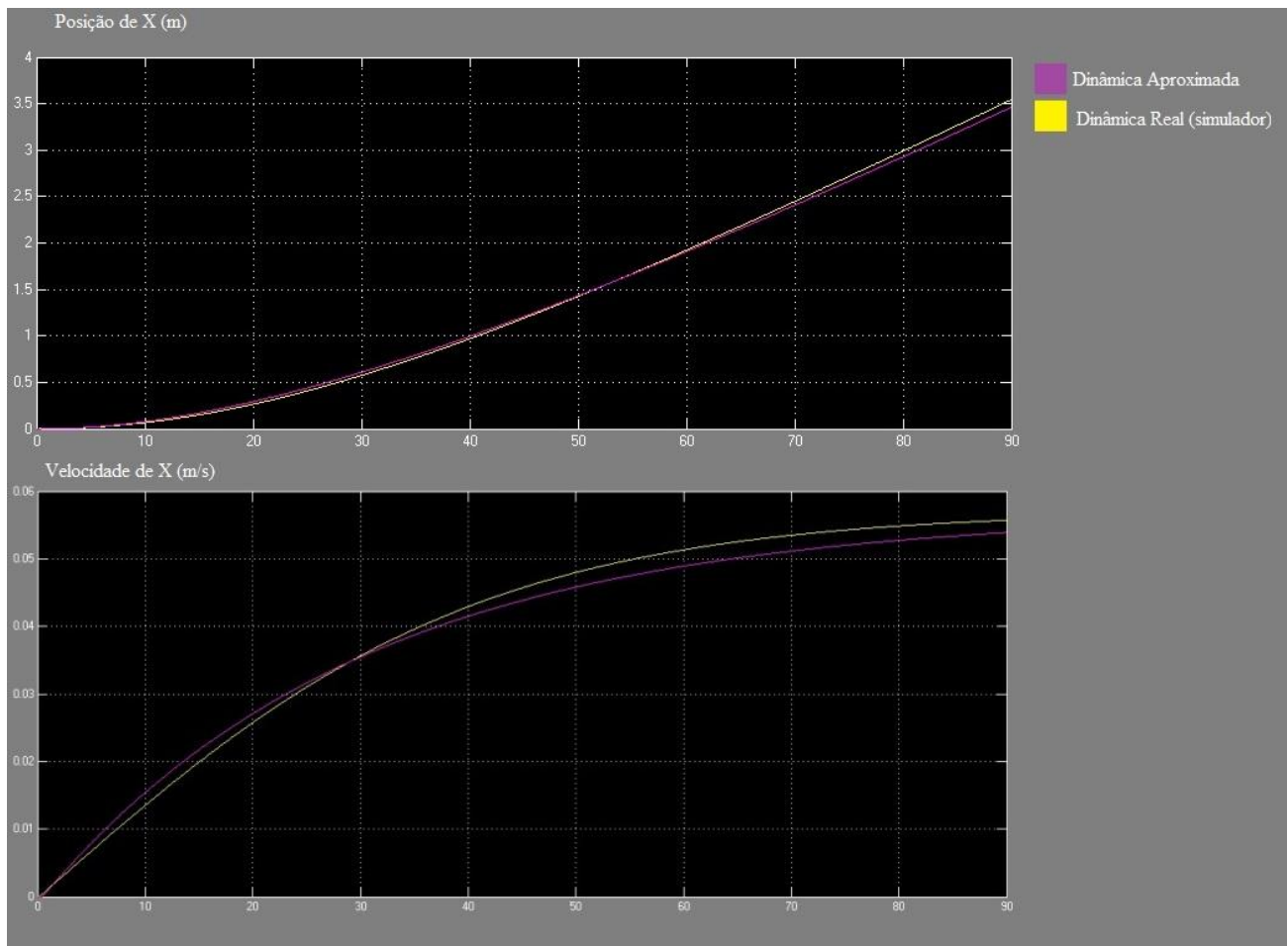


Figura 5.7 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. X em função do tempo

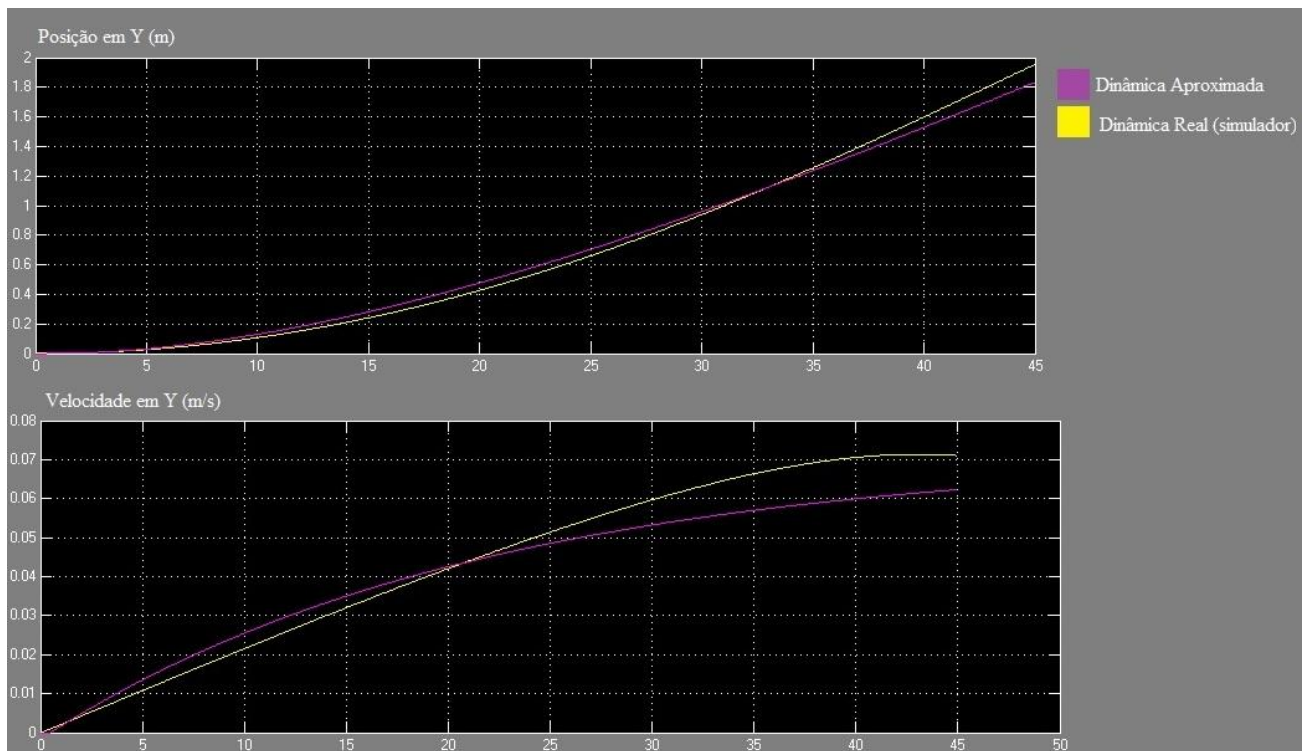


Figura 5.8 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. Y em função do tempo

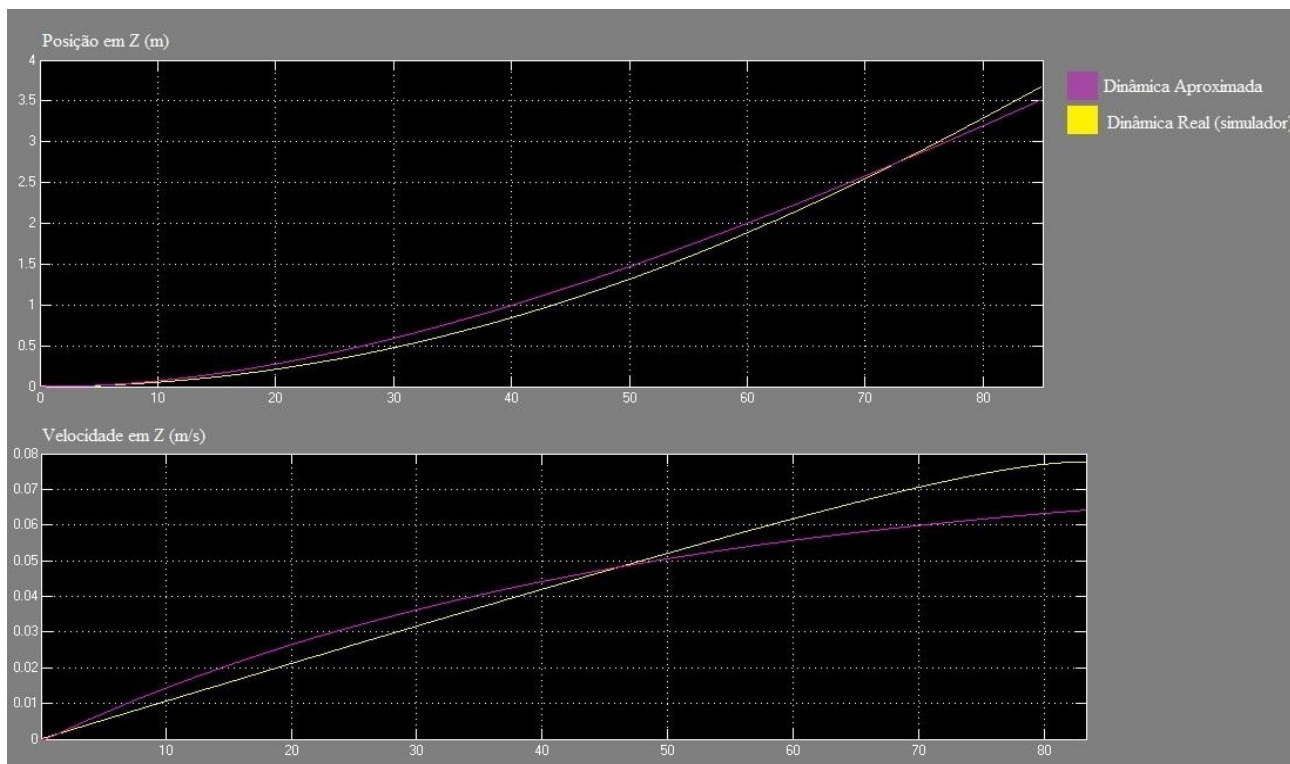


Figura 5.9 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. Z em função do tempo

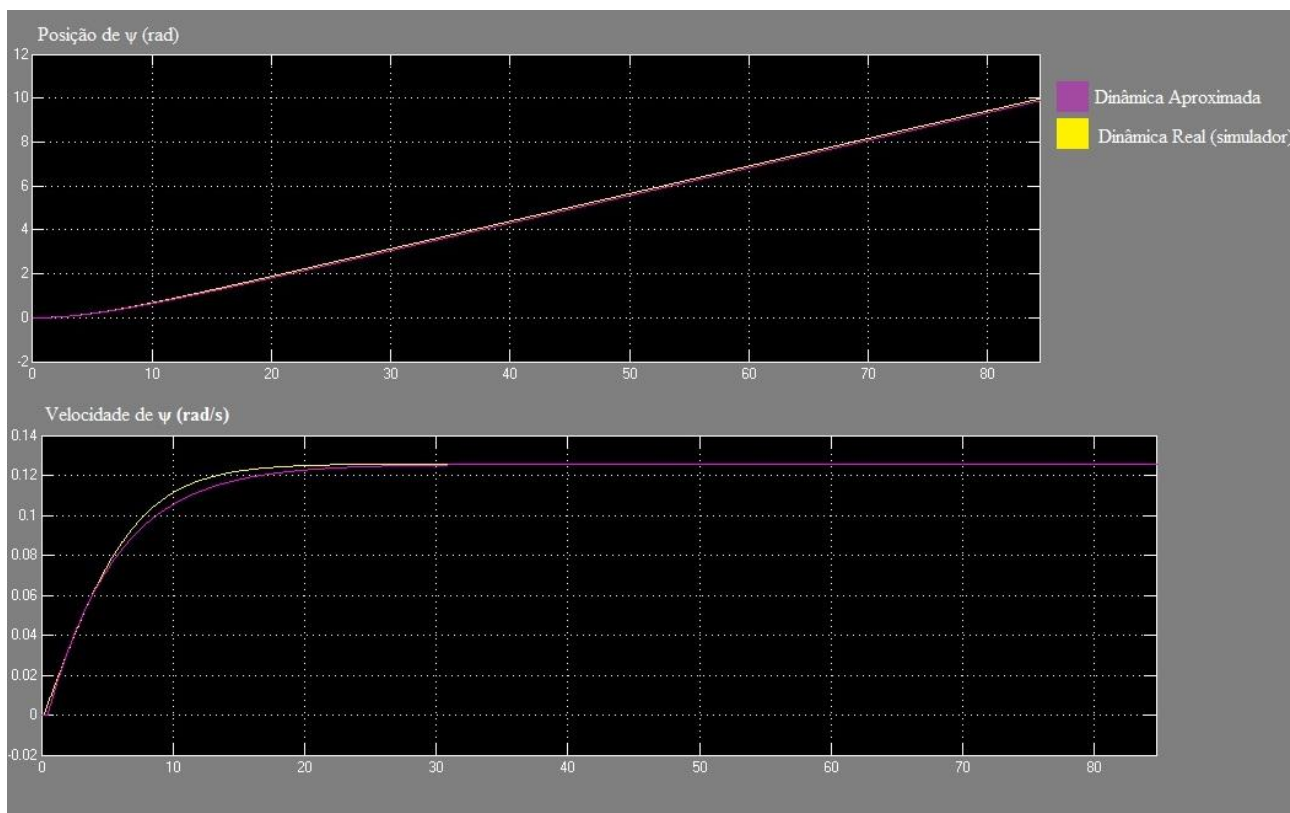


Figura 5.10 - Comparação entre os gráficos da dinâmica real e aproximada do G.L. ψ em função do tempo

É possível observar que, para baixas velocidades, o comportamento do sistema simplificado se assemelha muito ao do sistema completo. O que, portanto, valida a utilização destas simplificações para o projeto e análise do controlador em projeto. As diferenças notadas entre ambas as curvas se deve, principalmente, à ausência do efeito do arrasto hidrodinâmico no modelo simplificado.

5.4. Implementação Controle automático em Malha Fechada

Para o ROV deste trabalho foi adotado um algoritmo linear bem conhecido, o PID. Estes sistemas de posicionamento dinâmico devem manter os valores de V_x , V_y , Z e ψ bem próximos às referências fornecidas pelo operador ou obtidas nos momentos em que as malhas de controle foram acionadas.

O ROV possui sensores de velocidade responsáveis pela realimentação do sistema de controle de V_x e V_y e sensores de profundidade e rumo, responsáveis pela realimentação dos sistemas de controle de Z e ψ respectivamente.

Para o desenvolvimento do sistema de controle fechou-se a malha do sistema dinâmico do ROV, tendo esta nova malha de controle, utilizou-se o método de sintonia pelo Método da Curva de Reação do Sistema, proposto por Ziegler e Nichols, para fazer o projeto inicial do controlador do sistema. O método foi retirado da referência [21].

5.4.1. Sintonia pelo Método da Curva de Reação do Sistema

Este método foi proposto por Ziegler e Nichols para a sintonia on-line de controladores. Ele se baseia num único teste experimental que deve ser realizado com o sistema em malha aberta (controlador no modo manual).

Produz-se um sinal do tipo degrau com amplitude M na saída do controlador e a resposta $c(t)$ do sistema é registrada. O gráfico desta resposta a degrau é chamada de curva de reação do sistema. O método se aplica apenas no caso em que a resposta a degrau da planta em malha aberta tem o aspecto indicado na figura abaixo, típica de um sistema de primeira ordem com atraso.

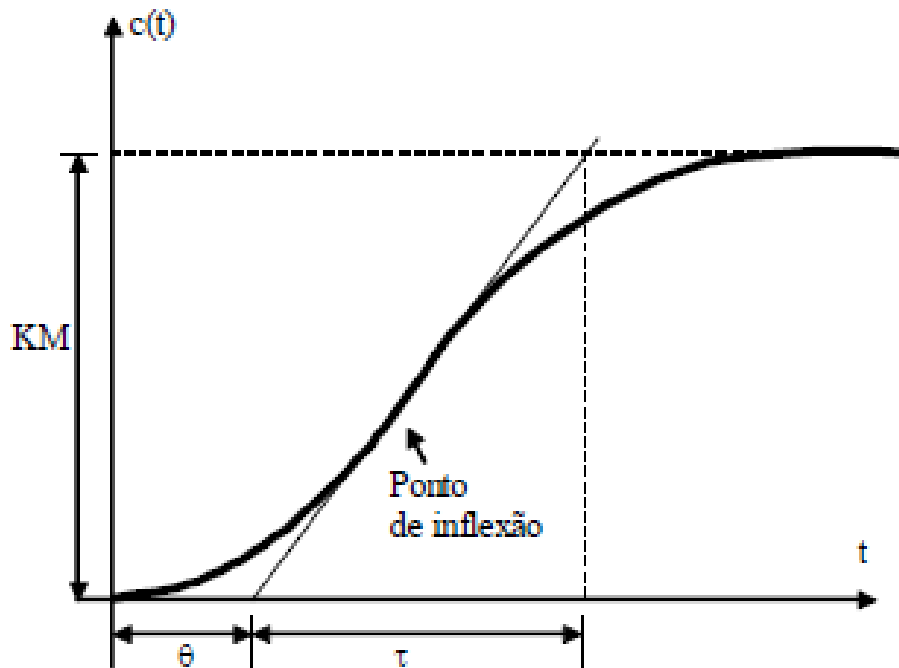


Figura 5.11 - Curva de reação de um sistema de primeira ordem com atraso

θ e τ são chamados na literatura, respectivamente, de tempo de retardo e constante de tempo. O ganho do sistema é calculado por

$$K = \frac{c(\infty)}{M}$$

Onde M é a amplitude do sinal degrau de entrada.

Muitas vezes, a determinação da tangente da curva de reação, e por consequência dos tempos θ e τ , possui grande incerteza. Nestes casos, podem-se utilizar métodos de aproximação entre a resposta de um sistema de primeira ordem com atraso e a curva de reação do sistema. Um destes métodos se baseia na definição dos tempos t_1 e t_2 , nos quais a curva de reação atinge 28,3% de $c(\infty)$ e 63,2% de $c(\infty)$ respectivamente.

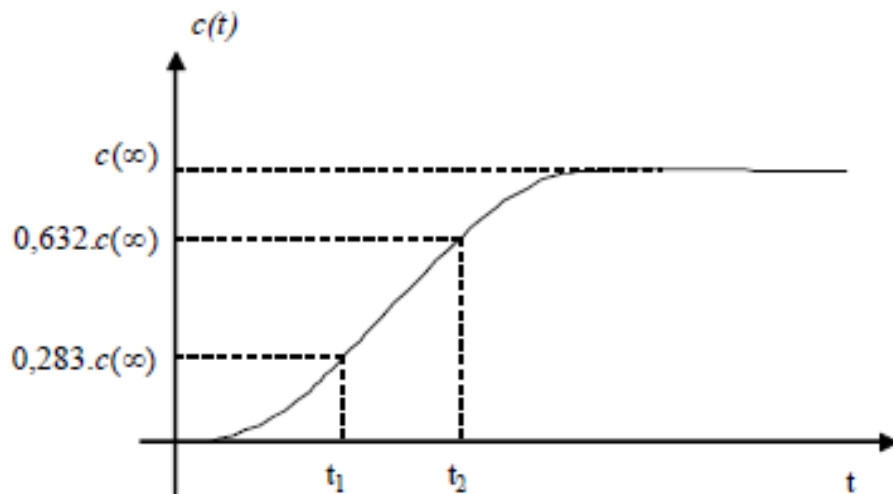


Figura 5.12 - Método para identificação dos parâmetros da curva de reação

Pode-se mostrar que uma boa aproximação de θ e τ é dada por:

$$\tau = \frac{3}{2}(t_2 - t_1) \quad \text{e} \quad \theta = t_2 - \tau$$

Os valores dos parâmetros do compensador devem ser escolhidos conforme indicado na tabela a seguir, segundo Ziegler e Nichols.

Tabela 5.1 - Método de Ziegler e Nichols para Curva de Reação

Controlador	K_P	T_I	T_D
P	$\frac{1}{K} \frac{\tau}{\theta}$	∞	0
PI	$0,9 \frac{1}{K} \frac{\tau}{\theta}$	$\theta/0.3$	0
PID	$1,2 \frac{1}{K} \frac{\tau}{\theta}$	2θ	0.5θ

Essas relações para a sintonia do controlador foram obtidas empiricamente com o objetivo de conseguir uma taxa de decaimento da ordem de $1/4$.

O método da curva de reação do processo apresenta as seguintes vantagens:

- É preciso realizar apenas um único experimento, não sendo necessário um processo de tentativa e erro.

- Os parâmetros do controlador são calculados de maneira simples.

Contudo esse método apresenta algumas desvantagens:

- O experimento deve ser realizado em malha aberta. Portanto, se uma mudança significativa nas condições de operação ocorre durante o teste, nenhuma ação corretiva é executada e os resultados podem ser bastante distorcidos.
- A obtenção precisa dos parâmetros θ e τ pode ser difícil se a medida da resposta do sistema se apresenta afetada de ruído ou se um simples registrador de papel é utilizado.
- Este método tende a ser sensível aos erros de calibração do controlador, em oposição ao método de Ziegler-Nichols da oscilação mantida, que é menos sensível a erros de calibração em K_P , uma vez que neste caso o ganho é ajustado durante o experimento.
- A resposta do sistema tende a ser oscilatória, dado que o método foi desenvolvido para produzir uma taxa de decaimento de $1/4$.
- O método não se aplica a sistema que tenham uma resposta oscilatória em malha aberta uma vez que esta não tem a forma padrão apresentada na figura anterior.
- O método apresenta bons resultados apenas quando a relação θ/τ estiver entre 0,1 e 0,5.

5.4.2. Cálculos dos valores para o método selecionado e validação do método

Apresentado o método utilizado para a sintonia do controlador para cada um dos graus de liberdade é preciso verificar se o mesmo apresentará resultados satisfatórios. Para tal a tabela a seguir mostra os valores calculados para cada um dos graus de liberdade a serem controlados e a relação θ/τ para efeito de confiabilidade do método a ser utilizado.

Tabela 5.2 - Valores calculados para os graus de liberdade a serem controlados

	V_x	V_y	V_z	V_ψ
t_1	12,3	9,22	20	2,06
t_2	30,15	21,48	46,68	5,22
K	0,1136	0,1426	0,1546	0,2512
θ	3,375	3,09	6,66	0,48
τ	26,775	18,39	40,02	4,74
θ/τ	0,1261	0,1680	0,1664	0,1013

Como pode ser observado na tabela acima, todos os valores da taxa θ/τ estão contidos entre 0,1 e 0,5. Pode-se então utilizar este método para determinar os ganhos do controlador a ser projetado.

5.4.3. Controlador PID

Supondo-se que os graus de liberdade do veículo foram desacoplados, como descrito anteriormente, então é possível controlar cada um dos graus de liberdade independentemente.

O controlador PD é o algoritmo mais simples que se pode usar para um sistema tipo duplo integrador (modelo simplificado para a dinâmica de ROVs)[15]. Entretanto, para projeto inicial foi decidido utilizar um controle do tipo PID, pois a inclusão do termo integrativo no controlador diminui o tempo de subida e elimina o erro estacionário da resposta do sistema controlado. Esses dois fatores são desejáveis no ROV deste projeto, principalmente para os controles de rumo e profundidade, pois para que o operador possa ter total controle do posicionamento do veículo dentro de um duto é fundamental que não haja erro em regime, já a resposta mais rápida do sistema é desejável, mas não essencial. Seu efeito se faz importante durante tomadas de decisão urgentes em que o operador precisa de uma ação rápida para manobrar o ROV.

Para efeito de análise, considera-se inicialmente o controlador PID formulado abaixo:

$$C(s) = K_P + K_D s + \frac{K_I}{s}$$

Onde:

- K_P é o ganho proporcional
- K_D é o ganho derivativo
- K_I é o ganho integral

A figura a seguir ilustra o diagrama de blocos do controlador a ser projetado.

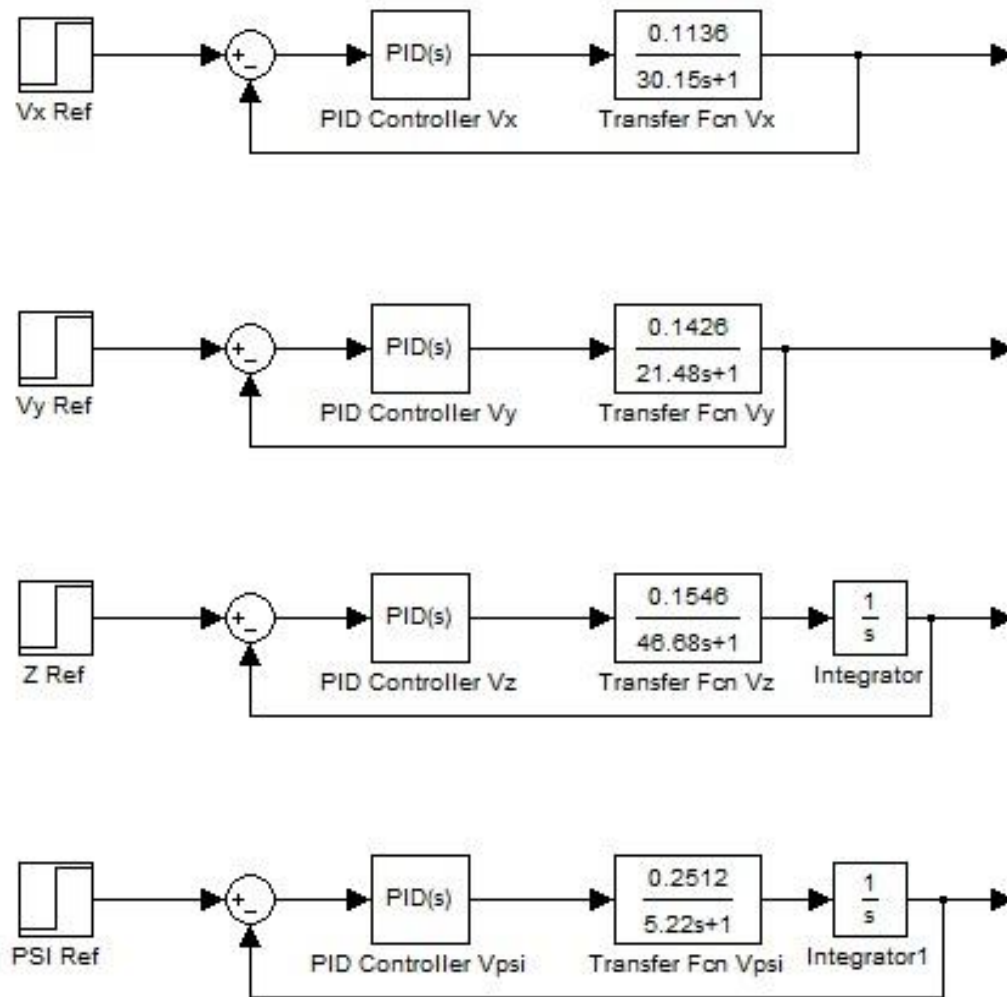


Figura 5.13 - Diagrama de Blocos do controlador projetado para cada grau de liberdade separadamente

Note que esta modelagem não leva em consideração a dinâmica dos sensores, sendo os mesmos considerados como uma realimentação perfeita unitária.

Como o equacionamento é análogo para todos os graus de liberdade, será mostrado o equacionamento da malha de controle de V_x e o mesmo pode ser replicado para os outros graus de liberdade.

Utilizando o método apresentado na Seção 5.4.1 foram encontrados os valores de K_P , K_D e K_I para cada um dos controladores do projeto. Sendo os mesmos apresentados na tabela a seguir.

Tabela 5.3 - Valores dos ganhos dos controladores PID

	V_x	V_y	Z	ψ
K_P	83,8028	50,0824	46,6417	47,1736
K_I	12,4152	8,1039	3,5016	49,1391
K_D	141,4172	77,3773	155,3169	11,3217

Os ganhos apresentados na tabela 5.3 foram implementados no algoritmo de controle para verificação da resposta do ROV submetido a uma entrada degrau novamente.

5.4.3.1. Resultados Obtidos com Controlador PID em Malha Fechada

Para os ganhos calculados na tabela 5.3 o sistema controlado apresenta as seguintes respostas:

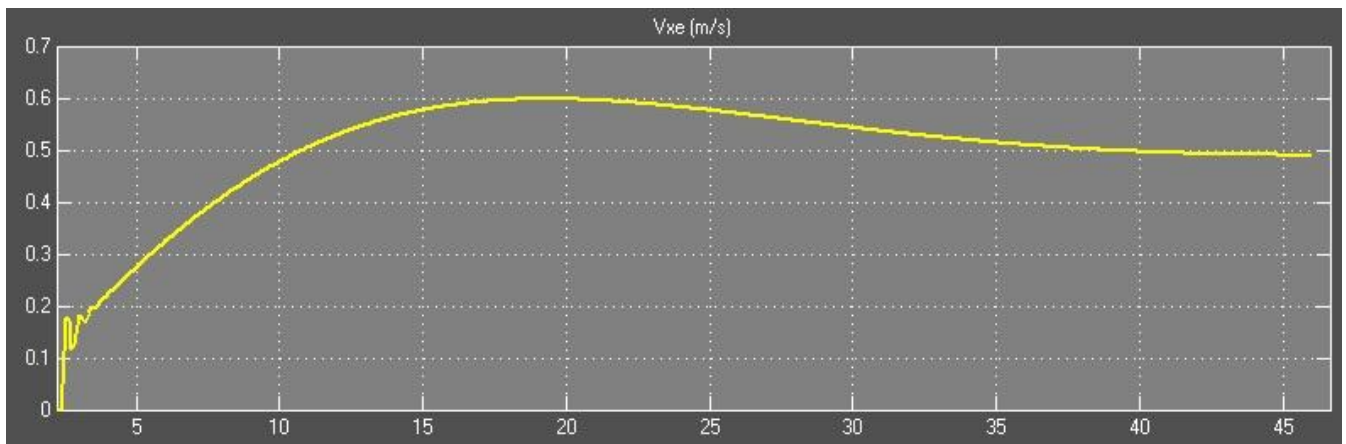


Figura 5.14 - Gráfico da Velocidade em X com o PID projetado

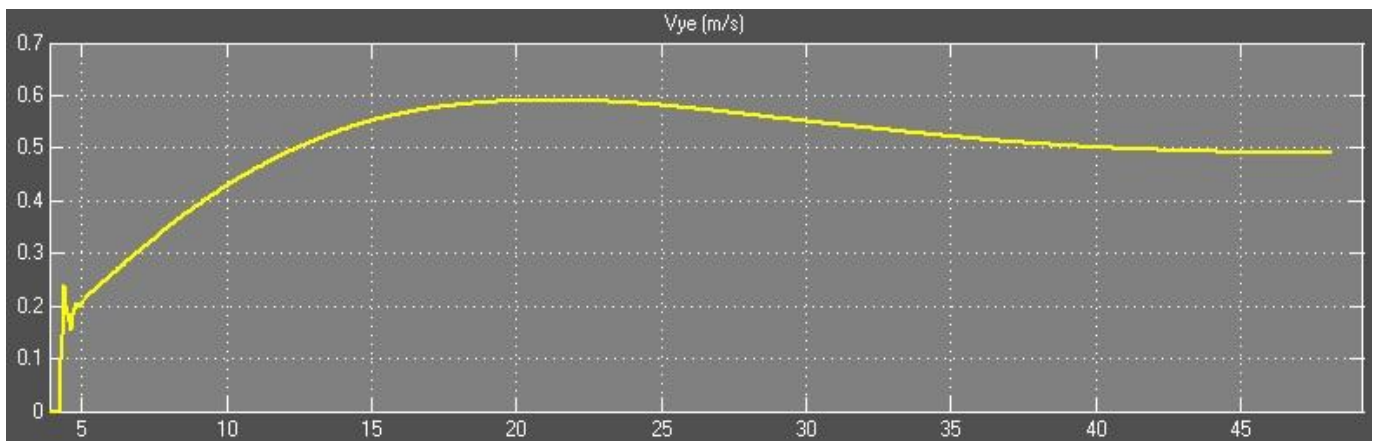


Figura 5.15 - Gráfico da Velocidade em Y com o PID projetado

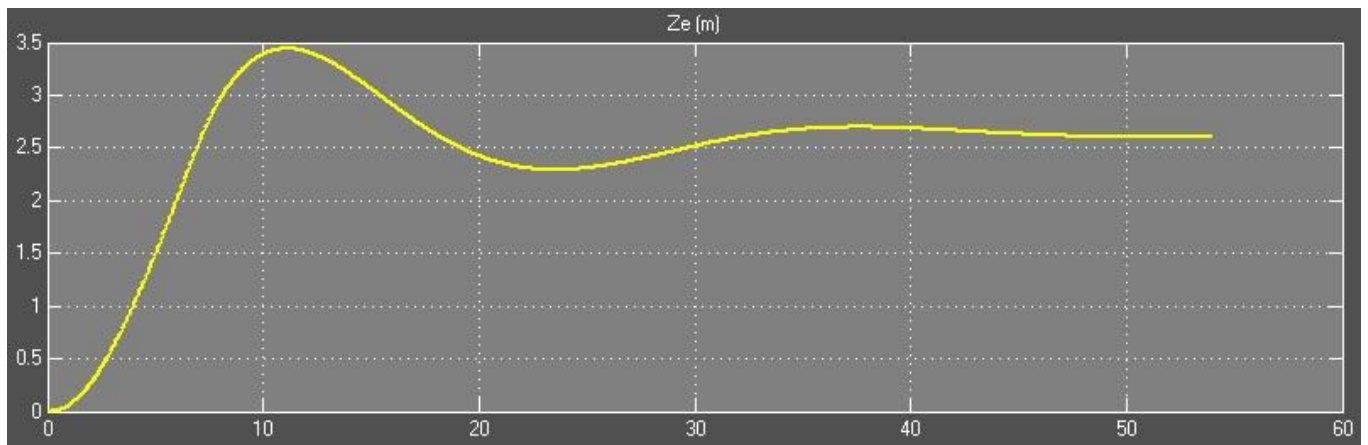


Figura 5.16 - Gráfico do Posicionamento em Z com o PID projetado

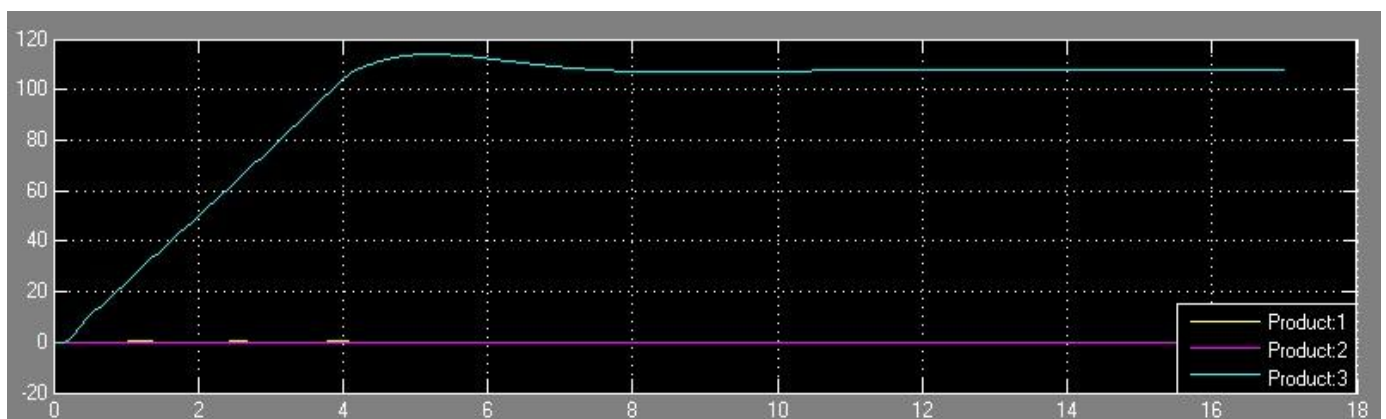


Figura 5.17 - Gráfico do Posicionamento em ψ com o PID projetado

Para o controlador calculado pelo Método da Curva de Reação do Sistema, que é um método muito simples de sintonização de controles PID, os resultados obtidos pelo simulador são satisfatórios. Apresentam um sobressinal da ordem de 10% a 20% e um tempo de estabilização em torno de 40 segundos para as velocidades em X e Y e profundidade Z, já a rotação em ψ apresenta um tempo de estabilização menor, da ordem de 8 segundos.

5.4.4. Análise de robustez do controlador PID

O controlador sintonizado para cada grau de liberdade apresentado anteriormente forneceu resultados satisfatórios para o caso analisado, que foi o de uma entrada do tipo degrau em cada grau de liberdade independentemente, porém esse controlador não foi analisado no quesito de robustez, ou seja, não se sabe se esse controlador é suficientemente robusto para aplicação em todos os casos.

Para fins de análise de robustez, a malha de controle será analisada conforme a teoria apresentada em [13], onde se analisa a robustez de um sistema através

sensibilidade $S(s)$ e sensibilidade complementar $T(s)$ do sistema para a frequência de operação do sistema.

5.4.4.1. Estabilidade do Sistema

Para estudar a estabilidade partimos de uma sistema genérico como o mostrado abaixo:

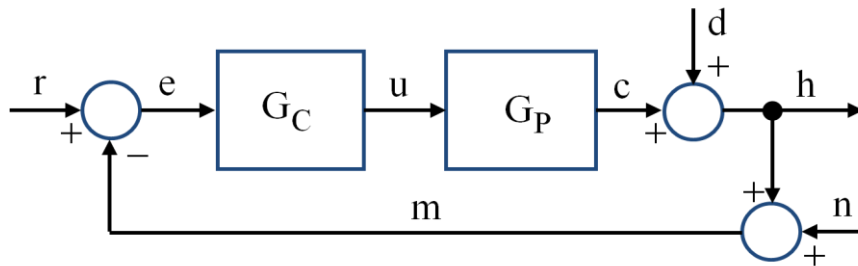


Figura 5.18 - Diagrama de blocos de um sistema genérico

Onde,

- G_C é a função de transferência do controlador
- G_P é a função de transferência da planta
- r é o sinal de Referência
- e é o sinal de Erro
- d é a perturbação refletida na saída da planta
- n é o erro (ruído) de medida
- h é o sinal de saída

Como mostrado em [13] uma maneira de analisar a estabilidade do sistema nominal (SISO) é através do critério de Nyquist, que pode ser enunciado da seguinte forma:

“uma condição necessária e suficiente para a estabilidade do sistema em malha fechada é que o número de voltas do Diagrama de Nyquist de $G_C(j\omega).G_P(j\omega)$ em torno do ponto $-1+j0$, no sentido anti-horário, seja igual ao número de polos instáveis de malha aberta” [21].

Já a análise do Desempenho do Sistema Nominal é estudado a partir de 4 premissas:

- Acompanhamento do sinal de referência

Partimos de $d(j\omega) = n(j\omega) = 0$ e obtemos a relação entre o sinal do erro e o sinal de referência de tal modo que $\frac{|e(j\omega)|}{|r(j\omega)|} < \alpha_r(\omega)$ onde $\alpha_r \gg 1$ é um dado do projeto, isso impõe que o ganho da função transferência de malha deve ser “grande” na região em que o sinal de referência tem a parte mais significativa de sua energia.

- Rejeição de perturbações

Agora partimos de $r(j\omega) = n(j\omega) = 0$ e obtemos a relação entre o sinal do erro e a perturbação de tal modo que $\frac{|e(j\omega)|}{|d(j\omega)|} < \alpha_d(\omega)$ onde $\alpha_d \ll 1$ é outro dado do projeto, isso impõe que o ganho da função transferência de malha deve ser “grande” na região em que as perturbações tem a maior parte de sua energia.

- Rejeição do erro de medida

Agora partimos de $r(j\omega) = d(j\omega) = 0$ e obtemos a relação entre o sinal do erro e o erro de medida tal modo que $\frac{|e(j\omega)|}{|n(j\omega)|} < \alpha_n(\omega)$ onde $\alpha_n \leq 1$ é outro dado do projeto, isso impõe que o ganho da função transferência de malha deve ser “pequeno” na faixa de frequências onde o erro de medida é “grande”.

- Insensibilidade a variações na planta

Supondo a planta sem perturbações e erros de medida temos o seguinte sistema:

$$y(j\omega) = \frac{G_P(j\omega) \cdot G_C(j\omega)}{1 + G_P(j\omega) \cdot G_C(j\omega)} r(j\omega)$$

Consideramos uma variação $\delta G_P(s)$ na função transferência da planta, de maneira que a saída passe a ser $y(j\omega) + \delta y(j\omega)$.

Tomando $\frac{|\delta y(j\omega)/y(j\omega)|}{|\delta g(j\omega)/G_P(j\omega)|} \leq \alpha_\delta(\omega)$ onde $\alpha_\delta \ll 1$ é outro dado do projeto, isso impõe que o ganho da função transferência de malha deve ser “grande” na região de frequências em que $\delta g(j\omega)/G_P(j\omega)$ é mais significativa .

Analisando as 4 medidas citadas notamos que a rejeição do erro de medida (que requer $G_C(j\omega) \cdot G_P(j\omega)$ “pequena”) conflita com as outras 3 premissas (que requer $G_C(j\omega) \cdot G_P(j\omega)$ “grande”) e assim o diagrama de Bode do ganho da função de transferência

de malha deve estar situado entre as barreiras de baixa e alta frequência como na figura 5.19.

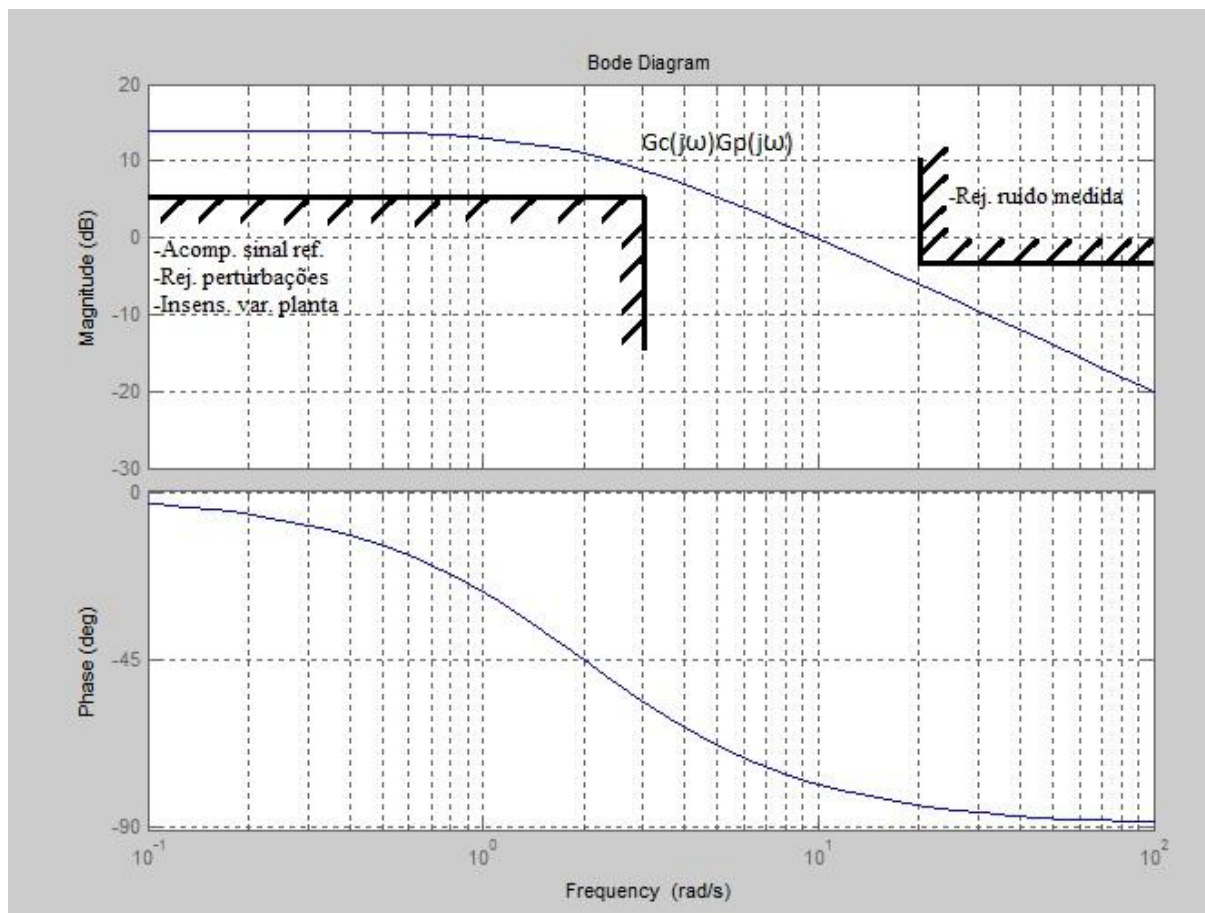


Figura 5.19 - Barreiras de alta e baixa frequência[13]

6. Conclusões

Neste trabalho foi apresentada a implementação de um modelo dinâmico para um veículo submarino não tripulado, a criação de um simulador em tempo real para a execução de testes sobre o mesmo e a utilização deste mesmo simulador para o desenvolvimento de um algoritmo de controle através de métodos de estratégia de controle clássicos. Os resultados obtidos foram analisados e submetidos a análises de estabilidade robusta.

Para desenvolver o sistema de simulação deste projeto, foram utilizados softwares de mercado da Mathworks, e a realização do mesmo sem estas ferramentas seria de uma complexidade muito maior. As bibliotecas de funções contidas nestes softwares são excelentes e foram capazes de gerar um modelo confiável.

Para o desenvolvimento do software em linguagem C++, responsável pela interface com o usuário através do Joystick, foi necessário um estudo mais aprofundado sobre as bibliotecas, principalmente na parte da implementação do protocolo de comunicação ethernet UDP. O desenvolvimento do projeto do simulador no geral excedeu as expectativas em sua complexidade e demandou um tempo de implementação suficiente para que nem todos os objetivos de análise de robustez do sistema de controle puderam ser atingidos. Entretanto, os resultados alcançados neste trabalho foram de fato satisfatórios, dadas as respostas apresentadas anteriormente.

A arquitetura proposta para o simulador do ROV alcançou seu objetivo máximo que era uma implementação de fácil manipulação para posterior utilização na modelagem de diferentes ROVs. Isso pode ser vivenciado em seu manuseio, na alteração de parâmetros físicos do modelo dado em [1] para os parâmetros do ROV deste projeto. Infelizmente não foi possível obter resultados confiáveis sobre o ROV deste trabalho, pois fatores essenciais na modelagem, como os coeficientes de arrasto hidrodinâmico, não estão disponíveis para implementação e ainda precisam ser estimados, portanto, para suprir estes valores foram utilizados os valores de [1], o que também resulta em uma limitação deste software de simulação, pois os ângulos de ataque do veículo ficam limitados aos valores contidos em [1]. Foi feito um algoritmo de extrapolação de valores para casos em que o veículo exceda estes limites, mas o mesmo não apresentou resultados tão satisfatórios quanto desejados.

Outro fator que foi observado no decorrer deste trabalho foi o aumento significativo no custo computacional para a execução do simulador em tempo real. Isso se deve em partes ao Real-Time Windows Target, pois ele utiliza o próprio kernel da CPU para realizar seu processamento e isso consome uma grande capacidade de processamento. Esse fato se verificou de tal forma que no decorrer do projeto foi necessário trocar o hardware de simulação para que os resultados do simulador não fossem comprometidos.

Homologado o simulador e sabendo de suas particularidades foi possível controlar o mesmo em malha aberta através de simulação. Devido a não simetria na matriz de inércia do veículo, assim como a pequena diferença entre os centros de gravidade e empuxo (o que definiu forças restaurativas baixas), o controle em malha aberta apresenta resultados não muito precisos. Porém quando implementado o controle em malha fechada, o simulador apresentou resultados excelentes.

Neste trabalho foi possível verificar a importância de uma modelagem eficiente e precisa e o quanto o projeto de um controle com estabilidade e robustez suficientes podem auxiliar no projeto de ROVs. Fazendo com que um ROV só seja submetido a um ambiente hostil depois de muito estudo sobre o seu comportamento e isso evita acidentes catastróficos.

7. Referências Bibliográficas

- [1] ISHIDERA, H., *Simulation and Experiment of Automatic Controlled ROV*, Mitsui Ocean Development & Engineering Co., Ltd., Tokio, 1986.
- [2] SOUZA, E. C., *Otimização da alocação de empuxo em sistemas de posicionamento dinâmico*, São Paulo, 2005. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo.
- [3] CARLTON, J., *Marine Propellers and Propulsion*, Elsevier, 2ª Edição, 2007.
- [4] GRANJA, B. B., *Alocação de Empuxo Para Sistema DP: Efeito da Utilização de Lemes de Alto Desempenho*, SOBENA, Rio de Janeiro, 2010.
- [5] TANNURI, E. A., *Sistemas de Posicionamento Dinâmico: Projeto, Análise e Novos Desenvolvimentos*, São Paulo, 2009. Tese de Livre Docência – Escola Politécnica, Universidade de São Paulo.
- [6] DE WIT, C., *Optimal Thrust Allocation Methods for Dynamic Positioning of Ships*, Delf Institute of Applied Mathematics, the Netherlands, 2009.
- [7] VAN DAALEN, E.F.G., *A Generic Optimization Algorithm for the Allocation of DP Actuators*, OMAE, Rotterdam, The Netherlands, 2011.
- [8] TANNURI, E. A., *Desenvolvimento de Metodologia de Projeto de Sistema de Posicionamento Dinâmico Aplicado a Operações em Alto-Mar*, São Paulo 2002. Tese de Doutorado – Escola Politécnica, Universidade de São Paulo.
- [9] DANTAS, J.L.D., *Software de Simulação Hardware-in-the-Loop para a Simulação do Sistema de Navegação e Controle de Veículos Autônomos Submarinos*, São Paulo, 2008. Escola Politécnica, Universidade de São Paulo.
- [10] FREIRE, L.O., *Simulador em Tempo Real de um Veículo Submarino*, São Paulo, 2010. Escola Politécnica, Universidade de São Paulo.
- [11] FABEKOVIC, Z., ESKINJA, Z., VUKIC, Z., *Micro ROV Simulator*, Zagreb, 2007.
- [12] A.GAMBIER, *Real-time Control Systems: A Tutorial*, Mannheim, 2004.
- [13] CRUZ, J.J., *Controle Robusto Multivariável*, EDUSP, 1996.

- [14] SLOTINE, J.J.E. e LI, W., *Applied nonlinear control*, Prentice-Hall, Inc., 1991.
- [15] GOULART, C., *Modelagem, Simulação e Controle de um veículo submarino de operação remota*, Rio de Janeiro, 2007. Universidade Federal do Rio de Janeiro.
- [16] AVILA, J.P.J, *Modelagem e Identificação de parâmetros hidrodinâmicos de um Veículo Robótico Submarino*, São Paulo, 2008. Escola Politécnica, Universidade de São Paulo.
- [17] SOUZA, E.C, *Modelagem e Controle de Veículos Submarinos Não Tripulados*, São Paulo, 2003. Escola Politécnica, Universidade de São Paulo.
- [18] PRACZYK, T., SZYMAK, P., *Decision system for a team of autonomous underwater vehicles*, Gdynia, Poland, 2011.
- [19] CUNHA, J. P. V. S., LIZARRALDE, F. C., COSTA, R. R., HSU, L., SMITH Jr., R., WOLLMANN Jr., D. & SANT'ANNA, A. C. C. M. (1994), *Sistema de posicionamento dinâmico para um veículo submarino de operação remota*, X Congresso Brasileiro de Automática pp. 1309–1314.
- [20] CUNHA, J. P. V. S., COSTA, R. R. & HSU, L. (1991), *Input/output variable structure position control of a remotely operated underwater vehicle*, in 'Robots in Unstructured Environments', Vol. 2, Programa de Eng. Elétrica, COPPE/UFRJ, Fifth International Conference on Advanced Robotics, Pisa, Italy, pp. 1305–1310.
- [21] OGATA, K., *Engenharia de Controle Moderno*, Pearson, 5ª Edição, 2010.
- [22] SILVA, D. S., *Identificação de Funções de Transferência utilizando como entrada um Degrau*, Ilha Solteira, 2008. Universidade Estadual Paulista.
- [23] Datasheet Model 300 DC Brushless Thrusters, TECNADYNE, <http://www.tecnadyne.com/>

ANEXO A

Código Software desenvolvido em C++

```
#include "thruster.h"
#include <iostream>
#include <cmath>
#include <QTcpSocket>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <cstring>
using namespace std;

//declaração das variáveis publicas para a comunicação
float buffer [100];
int bufferSize = sizeof(buffer);
int mensagem_length;
int mensagem_length2;
int number_bytes_sent;
int socketIdSND = socket(AF_INET, SOCK_DGRAM, 0);
int socketIdRCV = socket(AF_INET, SOCK_DGRAM, 0);
sockaddr_in serverAddr , clientAddr;
sockaddr &serverAddrCast = (sockaddr &) serverAddr;
sockaddr &clientAddrCast = (sockaddr &) clientAddr;

//integral da posição em Z e PSY
float posicaoZ = 0;
float posicaopsy = 0;

//auxiliar para a parte integral do controlador
float integraX = 0;
float integraY = 0;
float integraZ = 0;
float integraPSY = 0;

//auxiliar para a parte integral do controlador
float difantX =0;
float difantY =0;
```

```
float difantZ =0;
float difantPSY =0;
```

```
//CONSTANTES DO CONTROLADOR
```

```
float TDX = 1.6875;
float TIX = 6.75;
float TDY = 1.545;
float TIY = 6.78;
float TDZ = 3.33;
float TIZ = 6.32;
float TDPSY = 0.225;
float TIPSY = 0.9;
```

```
float KPvelX = 83.8028;
float KDvelX = KPvelX*TDX;
float KlvelX = KPvelX/TIX;
```

```
float KPvelY = 50.0824;
float KDvelY = KPvelY*TDY;
float KlvelY = KPvelY/TIY;
```

```
float KPposZ =46.6417;
float KDposZ =KPposZ*TDZ;
float KlposZ =KPposZ/TIZ;
```

```
float KPposPSY =50.6369;
float KDposPSY =KPposPSY*TDPSY;
float KlposPSY =KPposPSY/TIPSY;
```

```
//tempo de aquisição do joystick esta 40ms
float TD = 0.04;
```

```
void init_com (){
```

```
//endereço do Server para o bind
serverAddr.sin_family = AF_INET ;
serverAddr.sin_port = htons (20000);
serverAddr.sin_addr.s_addr = INADDR_ANY;
```

```
//BIND estabelece a conexão e a verifica socket de envio
bind (socketIdSND , &serverAddrCast, sizeof(serverAddr)) ;
printf("Conexão Estabelecida\n");
```

```
//recebe do cliente
socklen_t size = sizeof(clientAddr);
mensagem_length = recvfrom (socketIdSND, buffer,bufferSize,0,&clientAddrCast,&size ) ;
```

```
//imprime o que recebeu do cliente (teste de conexão)
if(mensagem_length<=0){
    printf("Erro\n");
}
else{
    printf("Mensagem length %d\n",mensagem_length);
    for ( int i = 0 ; i < 4 ; i++ )
        printf("valor {%d} %f \n",i,buffer[i]);
}
for ( int i = 0 ; i < (mensagem_length) ; i++ )
    buffer[i]= 0;
```

```
//endereço do cliente
clientAddr.sin_family = AF_INET;
clientAddr.sin_port = htons (25000);
clientAddr.sin_addr.s_addr = inet_addr("143.107.99.198");
bzero(&(clientAddr.sin_zero),8);

} //init_com
```

//FUNÇÃO DO CONTROLADOR PID

```
float controlador (float joystick, float KP, float KD, float KI, float *joystickanterior, float *integral){
```

```
float proppart;
float derivpart;
float integralpart;
float controlpart;
```

```
proppart = KP*joystick;
derivpart = KD*(joystick - *joystickanterior)/TD;
```

```
integralpart = *integral + TD*joystick;
```

```
controlpart = proppart + derivpart + KI*integralpart;
```

```
*joystickanterior = joystick;
```

```
*integral = integralpart;
```

```
return (controlpart);
```

```
}
```

//FUNÇÃO PRINCIPAL

void thrusterlocation (float force[4], float thruster[6]){

int i = 0;

int j = 0;

float matrixpsdA [4][3];

//pseudo Inversa da matriz A;

float thrustermaxF;

//maximo de propulsao de cada propulsor para frente

float thrustermaxR;

//maximo de propulsao de cada propulsor para atras

float vel[4];

float velref[4];

float inter;

//variavel de auxilio vel[2] + velref[2]el[0]

float inter1;

float forcesim [6];

float tmaxXF = 281.77;

float tmaxXR = 122.21;

float tmaxY = 70.56;

float tmaxMZ = 83.8028;

float vmaxX = 1.35;

float vmaxY = 0.77;

float vmaxZ = 1.09;

float vmaxMZ = 0.755;

float erroX;

float erroY;

float erroZ;

float erroPSY;

float posicaoZref;

float posicaopsyref;

float forcecontrol[4];

//zera os vetores de velocidade antes de iniciar o ciclo

for(i=0; i<4; i++) {

vel[i]=0;

velref[i]=0;

}

```

//recebe as velocidades do veiculo do simulador
socklen_t size = sizeof(clientAddr);
mensagem_length = recvfrom (socketIdSND, velref,32,0,&clientAddrCast,&size ) ;

//valor setado para o veiculo parado
thrustermaxF = 81.34;
thrustermaxR = -35.28;

//valor setado para alfas iguais a 30 graus
matrixpsdA[0][0] = 0.2887;
matrixpsdA[0][1] = -0.0003;
matrixpsdA[0][2] = 3.3234;
matrixpsdA[1][0] = 0.2887;
matrixpsdA[1][1] = 1.0003;
matrixpsdA[1][2] = -3.3234;
matrixpsdA[2][0] = 0.2887;
matrixpsdA[2][1] = 0.0003;
matrixpsdA[2][2] = -3.3234;
matrixpsdA[3][0] = 0.2887;
matrixpsdA[3][1] = -1.0003;
matrixpsdA[3][2] = 3.3234;

// pois os botoes superiores vao de 0 a 2
force[3] = force[3]/2;

//Velocidade
vel[0] = 0.5*force[0];      // Velocidade em x (m/s)
vel[1] = 0.5*force[1];      // Velocidade em y (m/s)
vel[2] = 0.5*force[3];      // Velocidade em z (m/s)
vel[3] = 0.5*force[2];      // Velocidade de rotaç o em z (rad/s)

// Integral da velocidade em Z para obter posi o Z de referencia
posicaoZ += vel[2]*TD;
// Integral da velocidade de rota o em Z para obter posi o psy
posicaopsy += vel[3]*TD;

//erro entre as velocidade do simulador e a do controle

erroX = vel[0] - velref[0];
erroY = vel[1] - velref[1];
erroZ = posicaoZ + velref[2]; //vel[2] + velref[2];
erroPSY = vel[3] - velref[3];

```



```
//CONTROLADOR
```

```
forcecontrol[0] = controlador(erroX, KPvelX, KDvelX, KIvelX, &difantX, &integraX);  
forcecontrol[1] = controlador(erroY, KPvelY, KDvelY, KIvelY, &difantY, &integraY);  
forcecontrol[3] = controlador(erroZ, KPposZ, KDposZ, KIposZ, &difantZ, &integraZ);  
forcecontrol[2] = controlador(erroPSY, KPposPSY, KDposPSY, KIposPSY, &difantPSY,  
&integraPSY);
```

```
//"OTIMIZAÇÃO"
```

```
//função que transforma a porcentagem do joystick em um valor em newtons
```

```
// if(force[0]>=0)    force[0]= force[0]*tmaxXF;  
// else if(force[0]<0)    force[0]= force[0]*tmaxXR;  
// force[1]= force[1]*tmaxY;  
// force[2]= force[2]*tmaxMZ;
```

```
//ALOCAÇÃO (SURGE - SWAY - YAW)
```

```
//multiplica a força dada pelo joystick com a pseudoinversa
```

```
i=0;  
j=0;  
inter= 0;  
for(i=0;i<4;i++){  
    for(j=0;j<3;j++){  
        inter= matrixpsdA [i][j] * forcecontrol[j];  
        thruster [i] += inter;  
        inter=0;  
    }  
}
```

```
thruster[4] = forcecontrol[3]/2;  
thruster[5] = forcecontrol[3]/2;
```

```
//FORÇAS ENVIADAS PARA O SIMULADOR
```

```
forcesim[0] = thruster[0];  
forcesim[1] = thruster[1];  
forcesim[2] = thruster[2];  
forcesim[3] = thruster[3];  
forcesim[4] = thruster[4];  
forcesim[5] = thruster[5];
```

```

// TESTE DE SATURAÇÃO
i=0;
for(i=0;i<6;i++){
    if(thruster[i]> thrustermaxF)    thruster[i] = thrustermaxF;
    else if(thruster[i]< -thrustermaxF)    thruster[i] = -thrustermaxF;
}

//NORMALIZAÇÃO.
for(i=0;i<6;i++){thruster[i]=thruster[i]/81.34;}

//envia para o cliente
number_bytes_sent = sendto(socketIdSND,forcesim,48,0,&clientAddrCast,sizeof(clientAddr));

printf("ERROZ = %f", erroZ);

return;
}

```

ANEXO B

Tabelas de Valores dos Gráficos contidos em [1], utilizadas para a implementação da modelagem dos coeficientes de arrasto hidrodinâmico.

Pontos medidos através das medições no ângulo de ataque α

α	CFx	CFz	CNy
-90,00	0	1	0
-86,90	-0,0625	1	0
-83,79	-0,125	1	-0,0625
-80,69	-0,1875	1	-0,0625
-77,59	-0,21875	1	-0,1125
-74,48	-0,25	1	-0,1125
-71,38	-0,28125	1	-0,10625
-68,27	-0,3125	1	-0,0875
-65,17	-0,3125	0,9375	-0,075
-62,07	-0,375	0,9375	-0,06875
-58,96	-0,375	0,9375	-0,0625
-55,86	-0,40625	0,9375	-0,0625
-52,76	-0,4375	0,90625	-0,0625
-49,65	-0,45625	0,9	-0,0625
-46,55	-0,4625	0,8875	-0,0625
-43,45	-0,46875	0,875	-0,0625
-40,34	-0,5	0,84375	-0,0625
-37,24	-0,5	0,8125	-0,0625
-34,14	-0,5125	0,75	-0,05625
-31,03	-0,53125	0,6875	-0,05
-27,93	-0,55	0,5625	-0,03125
-24,83	-0,5625	0,5	-0,0125
-21,72	-0,5625	0,4375	0
-18,62	-0,5625	0,375	0
-15,52	-0,5625	0,3125	0
-12,41	-0,56875	0,25	0,00625
-9,31	-0,56875	0,1875	0,00625
-6,21	-0,56875	0,125	0,00625
-3,10	-0,56875	0	0,0125
0,00	-0,575	0	0,04375

3,23	-0,58125	-0,09375	0,0625
6,45	-0,59375	-0,125	0,0625
9,68	-0,59375	-0,21875	0,0625
12,90	-0,59375	-0,28125	0,0625
16,13	-0,59375	-0,375	0,0625
19,35	-0,59375	-0,4375	0,06875
22,58	-0,59375	-0,46875	0,06875
25,81	-0,59375	-0,5625	0,06875
29,03	-0,58125	-0,625	0,06875
32,26	-0,5625	-0,75	0,06875
35,48	-0,55	-0,78125	0,06875
38,71	-0,53125	-0,84375	0,0625
41,94	-0,5	-0,90625	0,0625
45,16	-0,46875	-0,9375	0,0625
48,39	-0,4375	-1,03125	0,0625
51,61	-0,40625	-1,0625	0,05625
54,84	-0,375	-1,15625	0,05625
58,06	-0,35625	-1,1875	0,04375
61,29	-0,3375	-1,25	0,03125
64,52	-0,3125	-1,28125	0,01875
67,74	-0,29375	-1,3125	0
70,97	-0,25	-1,3125	0
74,19	-0,21875	-1,3375	0
77,42	-0,20625	-1,35625	0
80,65	-0,1875	-1,375	0,00625
83,87	-0,15	-1,375	0,0125
87,10	-0,125	-1,375	0,03125
90,32	-0,09375	-1,375	0,05625
93,55	-0,0625	-1,375	0,0625
100,00	0	-1,375	0,0625

Pontos medidos através das medições no ângulo de ataque β

β	CFx	CFy	CNz
-150,00	0,71928	0,53946	-0,0333
-146,67	0,6993	0,60606	-0,05328
-143,33	0,666	0,666	-0,0666
-140,00	0,6327	0,71928	-0,0666
-136,67	0,5994	0,7326	-0,0666
-133,33	0,5661	0,7992	-0,07326
-130,00	0,5328	0,83916	-0,07326
-126,67	0,4662	0,8658	-0,07992
-123,33	0,3996	0,87246	-0,07992
-120,00	0,3663	0,91242	-0,07326
-116,67	0,333	0,9324	-0,07326
-113,33	0,27972	0,9324	-0,07326
-110,00	0,25308	0,9657	-0,0666
-106,67	0,1998	0,98568	-0,0666
-103,33	0,1665	0,999	-0,04662
-100,00	0,1332	0,999	-0,03996
-96,67	0,08658	1,01232	-0,02664
-93,33	0	1,04562	-0,00666
-90,00	-0,0666	1,05228	0
-86,67	-0,0999	1,05228	0
-83,33	-0,13986	1,0323	0
-80,00	-0,18648	1,02564	0
-76,67	-0,2331	1,01232	0
-73,33	-0,25974	1,00566	0,00666
-70,00	-0,27972	0,999	0,01998
-66,67	-0,333	0,97902	0,0333
-63,33	-0,3663	0,95904	0,05328
-60,00	-0,3996	0,9324	0,05994
-56,67	-0,4329	0,91908	0,0666
-53,33	-0,4662	0,89244	0,0666
-50,00	-0,51948	0,8658	0,0666
-46,67	-0,5328	0,8325	0,07326
-43,33	-0,55944	0,7992	0,07326
-40,00	-0,57942	0,74592	0,07326
-36,67	-0,58608	0,6993	0,07326
-33,33	-0,5994	0,62604	0,07326
-30,00	-0,5994	0,5661	0,0666
-26,67	-0,5994	0,4995	0,0666
-23,33	-0,5994	0,4329	0,0666
-20,00	-0,5994	0,333	0,0666
-16,67	-0,5994	0,2997	0,0666
-13,33	-0,5994	0,2664	0,05328

-10,00	-0,5994	0,17316	0,0333
-6,67	-0,5994	0,1332	0
-3,33	-0,5994	0,0666	0
0,00	-0,5994	0	0
3,33	-0,5994	-0,0666	0
6,67	-0,5994	-0,1332	0
10,00	-0,5994	-0,17316	-0,0333
13,33	-0,5994	-0,2664	-0,05328
16,67	-0,5994	-0,2997	-0,0666
20,00	-0,5994	-0,333	-0,0666
23,33	-0,5994	-0,4329	-0,0666
26,67	-0,5994	-0,4995	-0,0666
30,00	-0,5994	-0,5661	-0,0666
33,33	-0,5994	-0,62604	-0,07326
36,67	-0,58608	-0,6993	-0,07326
40,00	-0,57942	-0,74592	-0,07326
43,33	-0,55944	-0,7992	-0,07326
46,67	-0,5328	-0,8325	-0,07326
50,00	-0,51948	-0,8658	-0,0666
53,33	-0,4662	-0,89244	-0,0666
56,67	-0,4329	-0,91908	-0,0666
60,00	-0,3996	-0,9324	-0,05994
63,33	-0,3663	-0,95904	-0,05328
66,67	-0,333	-0,97902	-0,0333
70,00	-0,27972	-0,999	-0,01998
73,33	-0,25974	-1,00566	-0,00666
76,67	-0,2331	-1,01232	0
80,00	-0,18648	-1,02564	0
83,33	-0,13986	-1,0323	0
86,67	-0,0999	-1,05228	0
90,00	-0,0666	-1,05228	0
93,33	0	-1,04562	0,00666
96,67	0,08658	-1,01232	0,02664
100,00	0,1332	-0,999	0,03996
103,33	0,1665	-0,999	0,04662
106,67	0,1998	-0,98568	0,0666
110,00	0,25308	-0,9657	0,0666
113,33	0,27972	-0,9324	0,07326
116,67	0,333	-0,9324	0,07326
120,00	0,3663	-0,91242	0,07326
123,33	0,3996	-0,87246	0,07992
126,67	0,4662	-0,8658	0,07992
130,00	0,5328	-0,83916	0,07326

133,33	0,5661	-0,7992	0,07326
136,67	0,5994	-0,7326	0,0666
140,00	0,6327	-0,71928	0,0666
143,33	0,666	-0,666	0,0666
146,67	0,6993	-0,60606	0,05328
150,00	0,71928	-0,53946	0,0333

Pontos medidos através das medições no ângulo de ataque γ

γ	CFy	CFz	CNx
-30,00	-0,7056	0,5292	-0,06468
-26,67	-0,7644	0,4116	-0,0588
-23,33	-0,78792	0,3528	-0,0588
-20,00	-0,7938	0,294	-0,04116
-16,67	-0,8232	0,2646	-0,02352
-13,33	-0,83496	0,2058	0
-10,00	-0,85848	0,1764	0,01176
-6,67	-0,87612	0,1176	0,04704
-3,33	-0,882	0,05292	0,05292
0,00	-0,89376	0	0,0588
3,33	-0,95625	-0,0625	0,0625
6,67	-0,96875	-0,125	0,0625
10,00	-1	-0,1875	0,05
13,33	-1,00625	-0,33125	0,0125
16,67	-1,00625	-0,46875	0
20,00	-1,00625	-0,55	-0,00625
23,33	-1,00625	-0,6875	-0,03125
26,67	-1	-0,71875	-0,0625