

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

**SIMULADOR DE VEÍCULOS SUBAQUÁTICOS NÃO
TRIPULADOS E ESTRUTURAS DO TIPO CABO**

por

Marcos Vinicius Scholl

Dissertação de Mestrado apresentada ao Programa
de Pós-Graduação em Modelagem Computacional
da Universidade Federal do Rio Grande, como
requisito parcial para a obtenção do grau de
Mestre em Modelagem Computacional

Rio Grande, Outubro, 2017

RESUMO

O ambiente subaquático é essencial para a vida. Além de seu efeito regulador em escala global e de uma fonte de diversidade de vida, os recursos existentes em suas profundezas são essenciais para a sociedade moderna e a economia global, como é caso da indústria do petróleo e do gás *offshore*. Isso motiva o interesse da comunidade científica e o desenvolvimento da tecnologia necessária para explorar essa região relativamente desconhecida. Veículos não tripulados e estruturas como cabos/*risers* são equipamentos que tem atraído interesse especial na pesquisa de tecnologias subaquáticas. O objetivo do presente trabalho é prover uma ferramenta gráfica para análise do comportamento dinâmico de veículos subaquáticos não tripulados e de estruturas flexíveis do tipo cabo. A ferramenta fornece um cenário virtual de visualização realista tridimensional de um meio submerso, no qual animações dos equipamentos estudados são executadas, com base nos resultados de simulações numéricas de modelos dinâmicos destes. O software também é útil para analisar e planejar tarefas a serem executadas pelos veículos não tripulados. Essas são consideradas as principais contribuições desta dissertação. Inicialmente, apresenta-se a modelagem cinemática e dinâmica do veículo, desenvolvida considerando a interação entre o este, o meio aquático e a ação dos propulsores. A dinâmica de estruturas flexíveis do tipo cabo é modelada através de um processo de discretização, onde o cabo é considerado como uma cadeia cinemática serial com elos rígidos que são conectados entre si por juntas que permitem movimentos elásticos de azimute, elevação e torção. Então, utiliza-se o formalismo de Euler-Lagrange para a obtenção das equações dinâmicas do movimento. A interação entre o veículo não tripulado e o cabo, como é o caso da maioria dos veículos remotamente operados (ROV) é tratada como se o ROV fosse a carga terminal vinculada ao cabo, mas com dinâmica própria, onde são aplicadas forças externas. Para o desenvolvimento do software de animação tridimensional, foi realizado um estudo sobre modelagem e animação gráfica, tendo-se definido Python como a plataforma de programação. A interface com o usuário foi criada com o Qt *framework*, usando PyQt para isso. Para a animação, o Blender Game Engine foi selecionado. O ambiente virtual subaquático consiste em uma embarcação posicionada na origem do sistema de referência inercial, e um objeto tridimensional ROV que representa o veículo subaquático. As simulações consideram as interações entre as dinâmicas do veículo e do cabo umbilical, e os resultados são observados nas animações tridimensionais produzidas pelo software. O processo de modelagem do veículo e do cabo umbilical são apresentados juntamente com os aspectos de projeto e implementação do software. As animações produzidas foram analisadas qualitativamente, e se verificou que as simulações foram consistentes do ponto de vista físico, o que permite afirmar que o software cumpre seu objetivo de prover um ambiente virtual realista.

Palavras-chaves: Veículos subaquáticos remotamente operados. Estruturas subaquáticas do tipo cabo. Software de simulação gráfica. Modelagem cinemática e dinâmica.

ABSTRACT

The underwater environment is essential to life. Besides its regulating effect in global scale and to be a source of life diversity, the existing resources in its depths are essential to the modern society and the global economy, as in the offshore oil and gas industry. This motivates the interest of the scientific community and the development of the technology necessary to explore this relatively unknown region. Unmanned vehicles and structures like cables/risers are equipments that have drawn special interest in the underwater technology research. The objective of the present work is to provide a graphical tool for analysis of underwater unmanned vehicles and cable-like flexible structures dynamic behavior. The tool provides a realistic, tridimensional virtual scenario of a submersed environment, where animations of the studied equipments are played, based on the results of numerical simulations of the dynamic models of them. The software is also useful to analyze and plan tasks to be executed by unmanned vehicles. These are considered as the main contributions to this dissertation. At first, it is presented the kinematic and dynamic modeling of unmanned vehicles, considering the interaction between it, the aquatic environment and the action of thrusters. The dynamics of cable-like structures is modeled through a discretization process, where the cable is considered as a serial kinematic chain where rigid links are connected by joints that allow elastic movements of azimuth, elevation and torsion. Then, the Euler-Lagrange formalism is employed to obtain the dynamic equations of motion. The interaction between the unmanned vehicle and the cable, as is the case of most remotely operated vehicles (ROV) is treated as if the ROV was the terminal load attached to the cable, but with its own dynamics, where external forces are applied. To develop the tridimensional animation software, a study of graphical modeling and animation was made, and Python was defined as the programming platform. The user interface was created on the Qt framework, using the PyQt API. For animation, the Blender Game Engine was selected. The virtual underwater environment consists in a surface vessel positioned at the origin of the inertial reference system, and a ROV that represents the underwater vehicle. The simulations consider the interactions between the dynamics of the vehicle and the umbilical cable, and the results are observed in the tridimensional animations produced by the software. The modeling process of the vehicle and the tether is presented along with the project and implementation aspects of the software. The animations produced by the software were qualitatively analyzed and it was verified that the simulations were consistent to the physical aspects, thus allowing to infer that the software provides a realistic virtual environment.

Palavras-chaves: Underwater remotely operated vehicles. Cable-like underwater structures. Graphical simulation software. Kinematic and dynamic modeling.

ÍNDICE

1	INTRODUÇÃO	15
1.1	Caracterização dos Veículos Subaquáticos Não Tripulados	15
1.2	Cabos Submersos	17
1.3	Objetivo Geral	17
1.4	Objetivos Específicos	18
1.5	Bibliografia Comentada	18
1.5.1	T. I. Fossen (1994)	18
1.5.2	A. M. Tavares (2003)	19
1.5.3	E.C. Souza (2003)	19
1.5.4	C. Goulart (2007)	19
1.5.5	C. R. Rocha (2012)	19
1.5.6	I. R. P. Rocha (2013)	20
1.5.7	G. Antonelli (2014)	20
1.5.8	V. S De Oliveira (2015)	20
1.5.9	R. Aguiar Jr (2016)	21
1.6	Bibliografia Comentada Correlata à Computação Gráfica	21
1.6.1	I. H Manssour e M. Cohen (2006)	21
1.6.2	F. S. Hill e S. M. Kelley (2007)	21
1.6.3	J. L. McKesson (2012)	21
1.6.4	J. F. Hughes, A. V. Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, K. Akeley (2013)	22
2	MODELAGEM DE VEÍCULOS SUBAQUÁTICOS	23
2.1	Cinemática	23
2.1.1	Sistema de Coordenadas	23
2.1.2	Transformações por Ângulos de Euler	25
2.1.3	Matriz de transformação dos seis graus de liberdade	28
2.2	Dinâmica	29
2.2.1	Dinâmica do Corpo Rígido	29
2.2.2	Força de massa Adicional	30
2.2.3	Forças de arrasto hidrodinâmico e sustentação	31
2.2.4	Forças restauradoras	33
2.3	Veículos Subaquáticos	34
2.3.1	Modelo sem corrente oceânica	34
2.3.2	Modelo com corrente oceânica	35
2.3.3	Modelo dinâmico dos propulsores	36
2.4	Modelo dinâmico do NEROV	37
3	CONTROLE DE VEÍCULOS SUBAQUÁTICOS	39
3.1	Equação diferencial de estado	39

3.2	Linearização por Realimentação	40
3.3	Trajectoria de Referência	42
3.4	Seguimento da trajetória: controle PID	43
3.5	Controlador PID, com linearização por realimentação, implementado para o NE-ROV	44
4	ESTRUTURAS FLEXÍVEIS DO TIPO CABO	46
4.1	Modelagem do cabo	46
4.2	Cabo com Carga Terminal: Simplificação do acoplamento de um veículo	53
5	FUNDAMENTOS DA MODELAGEM E ANIMAÇÃO GRÁFICA	55
5.1	Modelagem	55
5.1.1	Modelagem Por Primitivas Geométricas	55
5.1.2	Modelagem Por Superfícies Paramétricas	57
5.2	Cenários Virtuais e Visualização	58
5.2.1	Mundo ou Universo Virtual	58
5.2.2	Câmera Sintética	59
5.2.3	Fatores Que Influenciam o Grau de Realismo de Cenas	62
5.3	Animação	64
6	PROJETO DO SIMULADOR	66
6.1	Requisitos de Projeto	66
6.2	Modelagem do Simulador	67
6.3	Ferramentas de Desenvolvimento	70
7	Aspectos de Desenvolvimento	73
7.1	Modelagem do Ambiente	73
7.2	Animação	77
7.2.1	Simulações de ROVs	78
7.2.2	Simulações de Estruturas Flexíveis do Tipo Cabo	79
7.2.3	Simulações com Interação Entre ROV e Cabo	79
7.2.4	Configuração de Simulações	80
7.3	Interface Com o Usuário	80
7.4	Gráfico tridimensional animado e interativo	83
8	Resultados	86
8.1	Introdução	86
8.2	Simulação com o Veículo Autônomo (AUV)	87
8.3	Simulação de Estrutura Flexível do Tipo Cabo	91
8.4	Simulação de um ROV com o seu Cabo Umbilical	95
8.5	Considerações Finais	100
9	Conclusões	101

LISTA DE FIGURAS

Figura 1.1: Veículos subaquático do tipo ROV e AUV, adaptado de (EDGETECH, 2017)	16
Figura 1.2: Estimativa de crescimento de mercado para veículos subaquáticos não tripulado, no período de 2015-2020, adaptado de (INDUSTRYARC, 2017)	17
Figura 2.1: Sistema de referencia, adaptado de (ROCHA, 2012).	25
Figura 2.2: Orientação do veículo em relação ao referencial inercial (ROCHA, 2013).	26
Figura 2.3: Desenho simplificado do NEROV (TAVARES, 2003)	37
Figura 3.1: Sequência de ação do controlador (TAVARES, 2003)	44
Figura 4.1: Representação cabo com carga terminal (ROCHA, 2013).	47
Figura 4.2: Sistema subaquático (BANERJEE; DO, 1995).	47
Figura 4.3: Estrutura flexível e sua representação discreta (ZANELA, 2013).	48
Figura 4.4: Estrutura flexível e sua representação discreta (ZANELA, 2013).	49
Figura 4.5: Conectando um ROV como carga terminal do cabo (ROCHA, 2013)	53
Figura 4.6: Força exercida pela carga terminal (ROV) (ROCHA, 2013)	54
Figura 5.1: Alguns sólidos primitivos e seus parâmetros dimensionais	56
Figura 5.2: Cilindros idênticos desenhados com densidades de malhas distintas, no software OpenSCAD (KINTEL; WOLF, 2017)	56
Figura 5.3: Efeito das transformações geométricas sobre um paralelepípedo	57
Figura 5.4: ROV modelado por primitivas e transformações geométricas (SORIMAJI, 2015)	57
Figura 5.5: Exemplo de superfície paramétrica Bézier (CHANG, 2015)	58
Figura 5.6: Sistemas de orientação dos eixos cartesianos, adaptado de (LEARN OPENGL ES, 2012)	59
Figura 5.7: Cenário virtual de uma floresta (MCKESSON, 2012)	60
Figura 5.8: Definição de câmeras sintéticas com diferentes orientações(MANSSOUR; COHEN, 2006)	60
Figura 5.9: Projecção do volume de visualização para um <i>viewport</i> (ou vista). Adaptado de (JAN, 2016)	61
Figura 5.10:Tronco de pirâmide da projecção perspectiva (MANSSOUR; COHEN, 2006)	61
Figura 5.11: Movimento de câmera no USim	62
Figura 5.12:Cubo renderizado como sólido (esquerda) e <i>wireframe</i> (direita)	62
Figura 5.13:Tipos de fontes de luz em computação gráfica (MANSSOUR; COHEN, 2006)	63
Figura 5.14:Exemplo de iluminação aplicadas a um cubo, adaptado de (LEARN OPENGL, 2017)	64
Figura 5.15:Exemplo de reflexão adaptado de (ECK, 2016)	64
Figura 6.1: Casos de uso do USim	68
Figura 6.2: Diagrama de classes do USim	69

Figura 7.1: Interface principal do Blender (BLENDER FOUNDATION, 2017a)	73
Figura 7.2: Ambiente de simulação	74
Figura 7.3: Geração de superfícies aquáticas a partir do <i>Ocean Modifier</i> (RUDOLPH, 2014)	75
Figura 7.4: Câmera sintética e seus possíveis movimentos (adaptado de PETERSON (2015))	75
Figura 7.5: Visão <i>wireframe</i> do ambiente sintético	76
Figura 7.6: Visão <i>wireframe</i> do ambiente sintético, com destaque para o ROV e o navio	77
Figura 7.7: Visão <i>wireframe</i> do ambiente, com o ROV e as câmeras sintéticas	77
Figura 7.8: Visualização do ambiente subaquático, renderizada pelo Blender <i>engine</i> .	78
Figura 7.9: Exemplo de dados de um arquivo de resultados de simulação de ROV . . .	79
Figura 7.10: Estrutura de dados da discretização de um cabo considerando n elos virtuais (AGUIAR JR, 2016)	79
Figura 7.11: Exemplo de arquivo XML de configuração de uma simulação de cabo . . .	80
Figura 7.12: Interface com o usuário do USim	81
Figura 7.13: USim - Configuração de software	81
Figura 7.14: Seleção de simulação e configurações de apresentação	82
Figura 7.15: Interface do Gráfico tridimensional animado e interativo	83
Figura 7.16: Gráfico tridimensional com a plotagem de uma curva paramétrica	84
Figura 7.17: Gráfico interativo, diferentes ângulos de cena, por movimento de giro . . .	85
Figura 7.18: Gráfico interativo, ferramenta zoom: a) zoom ampliado; b) zoom diminuído.	85
Figura 8.1: Sistema de coordenadas de referência	86
Figura 8.2: Resultado da simulação do NEROV em malha fechada para uma trajetória em espiral	88
Figura 8.3: Resultado da simulação do NEROV em malha fechada para uma trajetória em espiral, sem considerar perturbação dinâmica	88
Figura 8.4: a) Instante inicial; b) Instante final; animação do NEROV, renderizados por uma câmera com plano de visão alinhado com um plano vertical.	89
Figura 8.5: Animação do NEROV em diferentes instantes de animação renderizados por uma câmera em duas perspectivas	89
Figura 8.6: Movimentação do veículo, representada a partir do gráfico tridimensional animado, com frequência amostral de 10s.	90
Figura 8.7: Renderizações de instantes de uma simulação de estrutura flexível do tipo cabo com uma carga terminal: a) Instante inicial. b) Instante final	92
Figura 8.8: Simulação de estrutura flexível do tipo cabo com uma carga terminal em queda livre durante 60s com <i>frames</i> a cada 5s	93
Figura 8.9: Trajetória espacial a partir do centro de massa da carga terminal durante a animação. Início da animação (verde), fim da animação (vermelho) . . .	94

Figura 8.10:Renderizações da dinâmica do veículo com cabo umbilical nos instantes 0s a), 15s b) e 30s c).	95
Figura 8.11:Animação gráfica tridimensional da dinâmica do veículo com o cabo um- bilical, <i>frames</i> a cada 10s	97
Figura 8.12:Animação tridimensional da dinâmica do veículo e do seu cabo umbilical.	98
Figura 8.13:Trajectoria espacial do centro de massa do veículo durante a animação (início da animação (verde), fim da animação (vermelho)).	99

LISTA DE TABELAS

Tabela 1:	Notação SNAME para o movimento de veículos aquáticos ROCHA (2012)	24
Tabela 2:	Movimentos de câmera e comandos de teclado	76
Tabela 3:	Campos dos arquivos de dados de simulação de ROVs	78
Tabela 4:	Campos dos arquivos de dados de simulação de Cabos	79
Tabela 5:	Teclas seletoras das câmeras	82
Tabela 6:	Atalhos de teclado disponíveis durante a animação	83

LISTA DE SÍMBOLOS

Modelagem e controle de veículos subaquáticos

∇	Volume do veículo
\mathbf{A}	Matriz de rotação do referencial do corpo para o referencial inercial
\mathbf{A}_{mn}	Coefficientes de massa adicional
\mathbf{a}_η	Aceleração comandada expressa no referencial inercial
$\vec{\mathbf{a}}_\nu$	Vetor aceleração comandada expressa no referencial do corpo
\mathbf{B}	Matriz de propulsão
\mathbf{B}_ϕ	Matriz de rotação em torno do eixo X
\mathbf{B}_θ	Matriz de rotação em torno do eixo Y
\mathbf{B}_ψ	Matriz de rotação em torno do eixo Z
$\mathbf{C}(\vec{\nu})$	Matriz de Coriolis e centrípeta
$\mathbf{C}_A(\nu)$	Matriz de Coriolis e centrípeta de massa adicional
\mathbf{C}_d	Coefficiente de arrasto do cabo umbilical
$C_D(R_\eta\alpha)$	Coefficiente de amortecimento
$C_L(R_\eta\alpha)$	Coefficiente de sustentação
$\mathbf{C}_{RB}(\nu)$	Matriz de Coriolis e centrípeta do corpo rígido
$C_p, C_q e C_r$	Coefficiente de amortecimento ao movimento de rotação em torno de $X_0, Y_0 e Z_0$, respectivamente
$C_x, C_y e C_z$	Coefficientes de forças de arrasto e sustentação do veículo na direção, $X_0, Y_0 e Z_0$, respectivamente
$C_K, C_M e C_N$	Coefficiente do momento das forças de arrasto e sustentação em torno de $X_0, Y_0 e Z_0$, respectivamente
D_S	Coefficientes de arrasto linear
D_Q	Coefficiente de arrasto quadrático
$\mathbf{D}(\vec{\nu}_r)$	Matriz de arrasto e sustentação
D_c	Diâmetro do cabo umbilical
$\vec{\mathbf{F}}_B$	Vetor força empuxo expresso no referencial do corpo
$\vec{\mathbf{F}}_c$	Vetor força do cabo umbilical
F_D	Força de arrasto
F_L	Força de sustentação
$\vec{\mathbf{F}}_W$	Vetor força peso expresso no referencial do corpo
\mathbf{F}_W^E	Vetor peso no referencial inercial
I_X, I_Y, I_Z	Momentos de inércia
I_{XY}, I_{YZ}, I_{XZ}	Produtos de inércia
g	Aceleração da gravidade
\mathbf{g}	vetor de forças e momento de peso e empuxo
$\mathbf{J}(\vec{\eta})$	Matriz de transformação da velocidade por ângulos de Euler
$\mathbf{J}_1(\vec{\eta}_2)$	Matriz de transformação da velocidade linear por ângulos de Euler

$\mathbf{J}_2(\vec{\eta})$	Matriz de transformação da velocidade angular por ângulos de Euler
K	Constante de proporcionalidade
$\mathbf{K}_P, \mathbf{K}_I, \mathbf{K}_D$	matrizes de ganhos proporcionais, integrais e derivativos
L	Comprimento do cabo umbilical
\mathbf{M}	Matriz de inércia
\mathbf{M}_A	Matriz de inércia de massa adicional
\mathbf{M}_B	Vetor momento da força de empuxo
\mathbf{M}_c	Vetor momento do cabo umbilical
\mathbf{M}_ν	Matriz de inércia expressa no referencial inercial
\mathbf{M}_W	Momento da força peso
\mathbf{M}_{RB}	Matriz de inércia do corpo rígido
$\dot{\eta}$	o vetor de velocidades no referencial inercial
$\dot{\vec{\eta}}$	vetor posição e orientação no referencial inercial
$\vec{\eta}_1$	Vetor posição no referencial inercial
$\vec{\eta}_2$	Vetor orientação no referencial inercial
$\dot{\vec{\eta}}_2$	Vetor velocidade angular do veículo
$\vec{\mathbf{n}}(\vec{\nu}, \vec{\eta})$	Vetor de cancelamento de não linearidades
$\vec{\eta}_d$	Vetor com a posição e a orientação desejadas em cada instante de tempo
$\vec{\nu}$	Vetor de velocidades no referencial do corpo
$\vec{\nu}_1$	Vetor de velocidade linear no referencial do corpo
$\vec{\nu}_2$	Vetor velocidade angular no referencial do corpo
$\vec{\nu}_c$	Vetor velocidade da corrente no referencial do corpo
$\vec{\nu}_C^E$	Vetor velocidade da corrente no referencial inercial
$\vec{\nu}_r$	Velocidade do veículo em relação a corrente marinha
ξ	fator de amortecimento
\mathbf{P}	Matriz de inércia dos propulsores
$\vec{\mathbf{g}}(\vec{\eta})$	Vetor de forças e momentos restauradores
$\vec{\tau}$	Vetor de forças e momentos dos propulsores
$\vec{\tau}_c$	Vetor de forças e momentos do cabo umbilical
$\vec{\tau}_m$	Vetor de torques motores
$\vec{\tau}_r$	Vetor de torques resistentes
$\vec{\tau}_{WB}$	Vetor de forças e momentos de peso e empuxo
$\vec{\tau}_{DL}$	Vetor de forças e momentos de arrasto e sustentação
ω	Vetor velocidade angular
ω_n	Frequência natural não-amortecida
ω_d	Frequência natural amortecida.
θ	Ângulo de arfagem
ϕ	Ângulo de rolamento
ψ	Ângulo de guinada

Estruturas flexíveis do tipo Cabo

$a_i, 1, \dots, n$	Alturas de referência para a definição da energia potencial de cada elo
C	Matriz de coeficientes de atrito
C_e, C_a, C_t	Submatriz da matriz C , em relação aos movimentos de elevação, azimute e torção
C_{ia}, C_{ie}, C_{it}	Coefficientes de atrito devido aos ângulos de azimute, elevação, e torção
E_C	Energia Cinética
E_{C_R}	Energia Cinética devido ao movimento de rotação
E_{C_T}	Energia Cinética devido ao movimento de translação
E_P	Energia Potencial
$\vec{f}(\vec{\theta}, \dot{\vec{\theta}}), \vec{F}$	Vetor de esforços Coriolis-centrífugos
\vec{F}_{0i}	Força em cada elo
\vec{F}_{C0}	Forças externas no referencial inercial aplicadas à carga terminal
\vec{F}_C	Forças externas no referencial do corpo aplicadas à carga terminal
\vec{F}_0	Forças externas transformadas para o referencial do inercial aplicadas à carga terminal
\vec{F}_j	Vetor força no referencial da ultima articulação
$\vec{G}(\vec{\theta}, \dot{\vec{\theta}}), \vec{G}$	Vetor de esforços gravitacionais
$h_i, 1, \dots, n$	Alturas
$H_{01}, H_{02}, H_{12}n$	Matrizes de transformações homogêneas
$I_{R_{ie}}$	Momentos de inércia relativos ao movimento de elevação dos elos
$I_{R_{it}}$	Momentos de inércia relativos ao movimento de azimute dos elos
I_{T_i}	Momentos de inércia relativos ao movimento de torção dos elos
$\mathbf{I}(\vec{\theta}), \mathbf{I}$	Matriz de inércia completa
$I_e, I_a, I_t n$	Submatrizes da matriz de inércia completa
K	Matriz de constantes elásticas das articulações do cabo
$K_e, K_a, K_t n$	Submatrizes da matriz de K , em relação aos movimentos de elevação, azimute e torção
$K_{ia}, K_{ie}, K_{it} n$	Constantes elásticas devidas aos ângulos de azimute, elevação e torção
L	Lagrangiano do sistema físico
l, l_{comp}	Comprimento do cabo
$l_i, i = 1, \dots, n$	Comprimento de cada elo
m_c	Massa da carga terminal
$m_i, i = 1, \dots, n$	Massa dos elos

m_e	Massa específica do cabo
m_{ea}	Massa específica da água
M_{1n}	Produto das matrizes de rotação
n	Número de elos
$N_e, N_a, N_t n$	Submatrizes da matriz de inércia completa do cabo
R_i	Raio de cada elo
$\vec{R}_{i,i-1}, i = n, \dots, 1$	Vetor braço de alavanca
t	tempo
$\vec{\tau}_m, \vec{\tau}$	Vetor de torques externos
$T_e, T_a, T_t n$	Submatrizes da matriz de inércia completa do cabo
$\tau_i, i = n - 1, \dots, 1$	Torques em cada articulação
$\vec{\tau}_c$	Torques produzidos pelas forças sobre a carga terminal
$\vec{\tau}_{cr}$	Torque produzido pela carga terminal na ultima articulação escrito no referencial do corpo
$\vec{\tau}_{ca}$	Torque produzido pela carga terminal na ultima articulação escrito no referencial da articulação
$\tau_{ia}, \tau_{ie}, \tau_{it}, i = 1, \dots, n$	Torques em azimute, elevação e torção nas articulações
τ_{ac}, τ_{ec}	Torques em azimute e elevação na ultima articulação provocados pela carga terminal
τ_{au}, τ_{eu}	Torques de azimute e elevação na ultima articulação
X_{DL}, Y_{DL}, Z_{DL}	Forças resultantes de arrasto e sustentação
K_{DL}, M_{DL}, N_{DL}	Momentos produzidos pelas forças resultantes de arrasto e sustentação
α_i	Ângulo entre o vetor unitário e o vetor Força em cada elo
$\theta_{ia}, \theta_{ie}, \theta_{iT}$	Ângulos de azimute, elevação e torção de cada elo
$\dot{\theta}_{ia}, \dot{\theta}_{ie}, \dot{\theta}_{iT}$	Derivada em relação ao tempo das posições em azimute, elevação e torção
$\vec{\theta}$	Vetor de estado com todas as posições angulares dos elos
$\dot{\vec{\theta}}$	Derivada primeira do vetor $\vec{\theta}$ em relação a t
$\ddot{\vec{\theta}}$	Derivada segunda do vetor $\vec{\theta}$ em relação a t
ρ	Massa específica do fluido
x_0, Y_0, Z_0	Sistema no referencial inercial
x_i, Y_i, Z_i	Sistema de referência com origem na articulação i
(x_c, Y_c, Z_c)	Coordenadas do centro de massa da carga terminal i
(x_i, Y_i, Z_i)	Coordenadas do centro de massa do i -ésimo elo

LISTA DE ABREVIATURAS

API	Interface de Programação de Aplicativos (API, do inglês <i>Application Programming Interface</i>)
AUV	Veículo Subaquático Autônomo (AUV, do inglês <i>Autonomous Underwater Vehicle</i>)
BGE	<i>Blender Engine</i>
CAD	Desenho assistido por computador (CAD, do inglês <i>Computer-Aided Design</i>)
CAGR	Taxa de crescimento anual composta
CSV	Campos separados por vírgulas (CSV, do inglês <i>Comma Separated Values</i>)
DOF	Graus de Liberdade (DOF, do inglês <i>Degrees of Freedom</i>)
NEROV	Veículo Experimental Norueguês Remotamente Operado (NEROV, do inglês <i>Norwegian Experimental Remotely Operated</i>)
NTH	Instituto Norueguês de Tecnologia (NTH, do norueguês <i>Norges Tekniske Høgskole</i>)
NuMA	Núcleo de Matemática Aplicada e Controle
OO	Orientação a objetos
ONU	Organização das Nações Unidas
PD	Proporcional e integral
PID	Proporcional, integral e derivativo
UML	Linguagem de Modelagem Unificada (UML, do inglês <i>Unified Modeling Language</i>)
UUV	Veículos Subaquáticos não Tripulados (UUVs, do inglês <i>Unmanned Underwater Vehicle</i>)
USim	Simulador Subaquático (USim, do inglês <i>Underwater Simulator</i>)
UVMS	Sistema Veículo-Manipulador Subaquático (UVMS, do inglês <i>Underwater Vehicle-Manipulator System</i>)
ROV	Veículo Remotamente Operado (ROV, do inglês <i>Remotely Operated Vehicle</i>)
SNAME	Sociedade de Arquitetos Navais e Engenheiros Marinheiros (SNAME, do inglês <i>Society of Naval Architects and Marine Engineers</i>)
XML	Linguagem Extensível de Marcação (XML, do inglês <i>eXtensible Markup Language</i>)

1 INTRODUÇÃO

Aproximadamente 70% da superfície do planeta é composta por água, sendo que a maior parte provem dos oceanos. Além de essencial para a vida humana, devido aos seus recursos, existe a grande importância de suas profundezas para a economia global. Os seus recursos naturais, incluindo-se petróleo e gás, são de grande importância econômica e social. A estimativa da Organização das Nações Unidas (ONU) é que a atividade econômica global, gerada nos oceanos renda entre U\$\$ 3 e 5 trilhões, um valor expressivo, mas que pode ser ainda mais elevado, uma vez que 90% do mercado global é realizado por via marítima e mais de 30% do óleo e do gás são extraídos do alto mar (ON LINE EDITORA, 2017).

A necessidade de conhecer e explorar essas vastas regiões do planeta motiva o desenvolvimento de tecnologias como a dos Veículos Subaquáticos não Tripulados (UUVs, do inglês *Unmanned Underwater Vehicle*). Através destes, é possível investigar e operar de forma remota nos ambientes em que seres humanos não conseguem ir diretamente, seja pelas grandes profundidades, ou pelos riscos do sítio em estudo. Estes veículos tornam possíveis não apenas a observação do ambiente, mas também a interação com ele. Assim, além dos aspectos científicos, os UUV são fundamentais para a exploração dos recursos do oceano, como é o caso da extração de petróleo/gás *offshore* em grandes profundidades (SILVEIRA, 2016).

Entre as missões em que os UUV são utilizados, podem ser listadas (ROCHA, 2012):

- Inspeção de cabos, tubulações, estruturas submersas e cascos;
- Construção e manutenção de estruturas *offshore*;
- Mapeamento do leito oceânico e das calotas polares;
- Busca de destroços de acidentes;
- Vigilância de portos;
- Monitoramento de espécimes biológicos.

1.1 Caracterização dos Veículos Subaquáticos Não Tripulados

Os veículos subaquáticos não tripulados são usualmente divididos em duas categorias, a dos Veículos Remotamente Operados (ROVs, do inglês *Remotely Operated Vehicle*) e a dos Veículos Subaquáticos Autônomos (AUVs, do inglês *Autonomous Underwater Vehicle*) (SICILIANO; KHATIB, 2008). A principal diferença entre as duas categorias é em relação à tomada de decisão na execução de tarefas. Enquanto na primeira as ações são realizadas por operadores humanos a partir de teleoperação, na segunda as tarefas são previamente definidas no veículo e este as executa de forma independente de ações humanas. A Figura 1.1 demonstra em um cenário representativo a diferenciação de operação destes veículos.

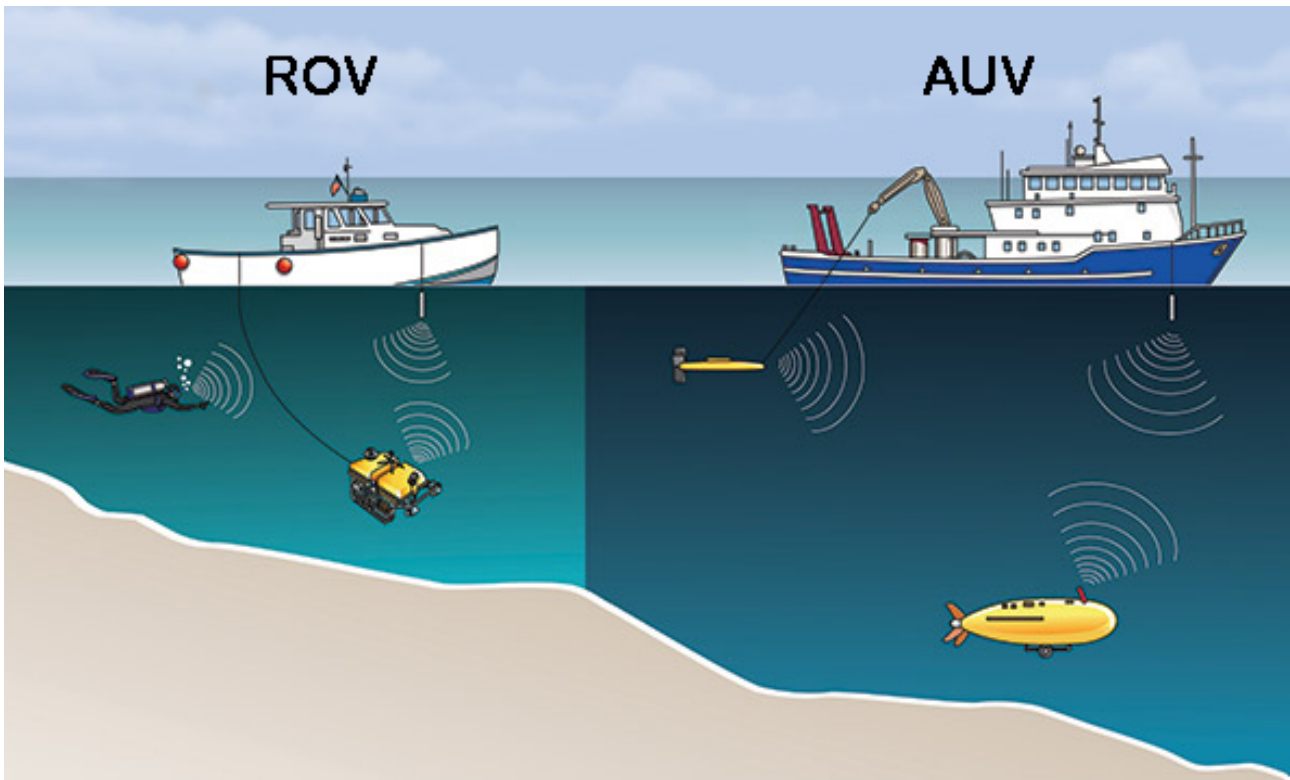


Figura 1.1: Veículos subaquático do tipo ROV e AUV, adaptado de (EDGETECH, 2017)

Um ROV se conecta a uma base de operação a partir de um cabo, a fim de possibilitar a comunicação com esta e, na maioria dos casos, fornecer energia ao veículo. Isso viabiliza a transmissão dos dados de sensores e câmeras do veículo para a base, e o envio de sinais de comando da base para o veículo. O cabo é denominado *umbilical* ou *cabo umbilical* (WHITCOMB, 2000).

Os AUVs não costumam ter umbilicais. Assim, a fonte de energia é intrínseca ao veículo, e a comunicação com a base não é contínua, exigindo grande capacidade de processamento embarcado. Isso possibilita a esse tipo de veículo um maior raio de ação e variedade de movimentos. Em missões de observação, é possível que o veículo opere por até semanas sem entrar em contato com uma base de operações ou precise de recarga de energia (WERNLI, 2001).

Os AUV podem parecer tecnologicamente mais avançados que os ROV, já que não precisam de supervisão contínua de operadores humanos. Porém, AUV ainda são limitados em relação aos tipos de tarefas que podem ser executadas, seja pela energia disponível, seja pela complexidade computacional exigida. Por esse motivo, os ROV são muito empregados em operações que exijam intervenção com o meio ou cuja complexidade demande operação humana. Assim, ambas as categorias de UUV são temas ativos de pesquisa e desenvolvimento (SOKOLOV et al., 2017). O aumento do número de aplicações subaquáticas, como a exploração de petróleo *offshore*, por exemplo, resulta num crescimento do número de veículos em operação, tanto autônomos quanto teleoperados, mesmo com fatores econômicos em baixa. Segundo INDUSTRYARC (2017), Figura 1.2, o mercado de veículos subaquáticos autônomos possui uma taxa de crescimento anual composta (CAGR) de 11,3% para os veículos autônomos (AUVs), enquanto para os veículos do tipo ROV é de 32%. Indica ainda que o mercado de operações de ROV, deve

alcançar a marca de U\$\$ 1,8 bilhão em 2020 (INDUSTRYARC, 2017).

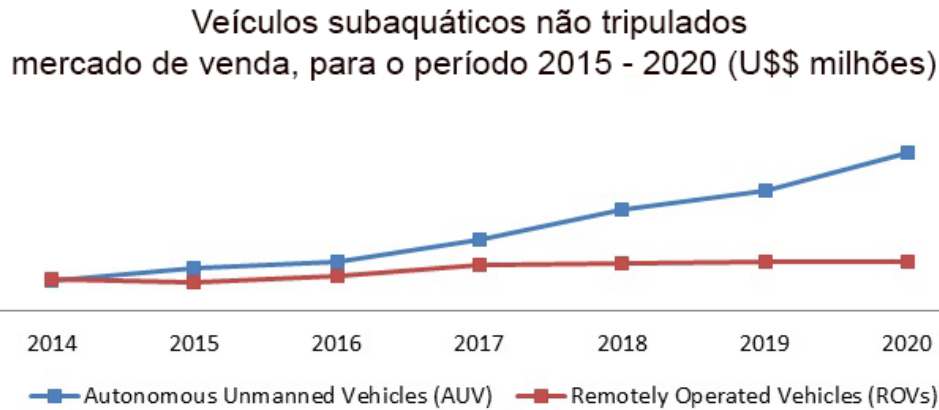


Figura 1.2: Estimativa de crescimento de mercado para veículos subaquáticos não tripulados, no período de 2015-2020, adaptado de (INDUSTRYARC, 2017)

1.2 Cabos Submersos

Cabos e estruturas flexíveis similares são muito usados no meio subaquático, em particular nas indústrias petrolífera e naval. Os cabos de amarração, ancoragem e reboque são fundamentais para o posicionamento das estruturas aquáticas. Cabos de força e comunicação são usados em ROV, por exemplo.

O projeto e a especificação de cabos constituem uma área de pesquisa em constante desenvolvimento. No caso de cabos de amarração e ancoragem, é importante determinar os esforços sofridos por estes, onde usualmente são utilizadas técnicas como análise por elementos finitos (MARZO, 2010).

O comportamento dinâmico dos cabos submetidos aos esforços causados pelo meio subaquático também é um tema de interesse. Em vários trabalhos, os esforços causados por correntes subaquáticas, o peso próprio e outros fatores ambientais são tratados de forma simplificada. Porém, a dinâmica envolvida na interação do cabo com o ambiente submerso é complexa e isso serve de motivação para o estudo e o desenvolvimento de modelos dinâmicos mais precisos, que aproximam bem a realidade física. Apesar de todo o avanço alcançado na construção de ROVs, tendo sua modelagem dinâmica bem consolidada e divulgada pela literatura, ainda existem desafios quanto ao desenvolvimento de modelos dinâmicos realistas, que representem bem o comportamento dinâmico do cabo interagindo com outras estruturas subaquáticas.

1.3 Objetivo Geral

O objetivo deste trabalho é desenvolver um simulador gráfico para veículos subaquáticos não tripulados e estruturas flexíveis do tipo cabo. O software desenvolvido considera as pesquisas

sobre a dinâmica dos cabos submersos e os efeitos destes sobre os veículos, realizadas até o momento no Núcleo de Matemática Aplicada e Controle da Universidade Federal do Rio Grande (NuMA-FURG). O software desenvolvido possibilita realizar simulações com veículos subaquáticos operando de forma autônoma ou com cabos umbilicais, com os efeitos gerados por estes nos veículos. Também é possível analisar o comportamento de cabos isolados, com ou sem carga terminal.

O software deve possibilitar a análise qualitativa de resultados das simulações a partir de visualização de animações em três dimensões, realizadas em um cenário virtual que emula o ambiente em que os objetos analisados atuam. Com isso, tem-se uma ferramenta de apoio para estudar a dinâmica desses sistemas subaquáticos, bem como para analisar e planejar tarefas a serem desenvolvidas. Por ser projetado de forma modular, o software também pode auxiliar no desenvolvimento de geradores de trajetórias e controladores para UUVs.

1.4 Objetivos Específicos

Para o desenvolvimento do simulador, foram definidos os seguintes objetivos específicos:

- Estudar a modelagem física de veículos subaquáticos e cabos, considerando aspectos dinâmicos e cinemáticos;
- Executar simulações numéricas, de forma a avaliar o comportamento dinâmico de veículos subaquáticos;
- Efetuar simulações numéricas, de forma a avaliar o comportamento dinâmico de estruturas flexíveis do tipo cabo;
- Projetar o software simulador;
- Implementar o simulador;
- Testar e analisar as simulações e analisar os resultados obtidos.

1.5 Bibliografia Comentada

O trabalho foi iniciado com uma revisão da literatura sobre a modelagem de veículos subaquáticos e de estruturas submersas flexíveis do tipo cabo. A seguir, são comentados os principais trabalhos anteriores que nortearam o desenvolvimento da pesquisa.

1.5.1 T. I. Fossen (1994)

Guidance and Control of Ocean Vehicles

Este livro é referência da maioria dos trabalhos que abordam a modelagem e o controle de veículos subaquáticos. Apresenta um estudo aprofundado sobre o controle de diversos veículos

marinhos. No Capítulo 2 é apresentada a modelagem desses veículos, considerando aspectos dinâmicos e cinemáticos. O Capítulo 3 trata da modelagem das perturbações externas, como ventos, correntes oceânicas e ondas. O Capítulo 4 descreve o estudo do controle e da estabilidade de veículos subaquáticos. No Capítulo 5 é discutido o estado da arte, com técnicas de modelagem para navios. Por fim, os dois últimos Capítulos, 6 e 7, tratam do estudo de controle automático de navios e outras embarcações de superfície de alta velocidade.

1.5.2 A. M. Tavares (2003)

Um Estudo Sobre a Modelagem e o Controle de Veículos Subaquáticos Não Tripulados

Esta dissertação apresenta um abrangente estudo teórico sobre as modelagens dinâmica e cinemática de veículos subaquáticos não tripulados. São apresentados e comparados métodos de modelagem cinemática por ângulos de Euler, parâmetros de Euler e parâmetros de Rodrigues. A modelagem dinâmica considera a interação entre a estrutura do veículo e o meio aquático e a ação dos propulsores sobre o veículo. Efetuam-se simulações em malha aberta e em malha fechada com um controle do tipo PID, utilizado em conjunto com a técnica linearização por realimentação.

1.5.3 E.C. Souza (2003)

Modelagem e Controle de Veículos Submarinos Não Tripulados

Esta dissertação desenvolve o emprego de algumas técnicas de controle em veículos submarinos e apresenta a modelagem dos propulsores levando em consideração efeitos eletromecânicos de motores elétricos e hidrodinâmicos causados pela interação entre a água e o propulsor. Além disso, apresenta um modelo de cabo umbilical de um veículo operado remotamente, com o objetivo de reproduzir os efeitos de distúrbio presentes quando operado sob condições reais. O modelo resultante foi utilizado nas simulações para avaliação do desempenho sistema controlado.

1.5.4 C. Goulart (2007)

Modelagem, Simulação e Controle de um Veículo Submarino de Operação Remota

Este trabalho aborda a modelagem dinâmica do ROV LUMA, veículo desenvolvido com objetivo de inspecionar dutos de adução em barragens de usinas hidrelétricas. É desenvolvido um simulador de ROVs para visualizar o comportamento dinâmico do veículo. Outra contribuição importante refere-se a obtenção, de forma experimental, das características hidrodinâmicas do veículo, como as massas adicionais, a força de arraste e o torque restaurador além dos centros de gravidade e de flutuação do veículo.

1.5.5 C. R. Rocha (2012)

Planejamento de movimento de sistemas robóticos de intervenção subaquática baseado na teoria dos helicoides

Esta tese estuda o problema do planejamento de movimento de sistemas veículo-manipulador subaquáticos (UVMS, do inglês *Underwater Vehicle-Manipulator System*) em tarefas de intervenção. Abrange o desenvolvimento da dinâmica, cinemática e simulação de veículo subaquático, além do manipulador robótico. O trabalho utiliza a análise cinemática baseada na teoria dos helicoides, teoria dos grafos e ferramentas derivadas para definir modelos cinemáticos dos UVMS em diferentes cenários de execução de tarefas de intervenção. Um *framework* computacional é projetado para facilitar a implementação desses modelos. Também são apresentadas simulações de uso dessa estrutura em diferentes cenários de operação.

1.5.6 I. R. P. Rocha (2013)

Modelagem Matemática de Cabos Umbilicais e Veículos Subaquáticos Não Tripulados

A presente dissertação apresenta um formalismo acoplado a modelagem dinâmica do cabo umbilical e a de um ROV. Para desenvolver o modelo dinâmico do veículo considerou-se a interação do veículo com o meio aquático e a ação dos propulsores. Foi modelada a perturbação provocada pelo cabo a partir do formalismo discreto. Foram feitas simulações em que o ROV é conectado a extremidade livre do cabo, sendo, portanto a sua carga terminal e possuindo dinâmica própria, influenciada principalmente pelos seus atuadores. Estas simulações mostraram que os resultados obtidos eram fisicamente esperados. Porém, o acoplamento não foi generalizado, ou seja, o mesmo foi conseguido apenas para casos particulares de modelos de cabo com no máximo cinco elos rígidos na cadeia poliarticulada.

1.5.7 G. Antonelli (2014)

Underwater Robots

Este livro, aborda os principais aspectos de controle em tarefas de manipulação subaquáticas. É discutido um modelo matemático com impacto significativo na estratégia de controle. O problema de controlar um veículo submarino autônomo de 6 graus de liberdade (ou DOF, do inglês *Degrees of Freedom*) é profundamente investigado e um levantamento de detecção/estratégias de tolerância à falhas para veículos subaquáticos não tripulados é fornecido. A cinemática inversa, o controle dinâmico e a interação para os sistemas de veículo-manipulador subaquático são discutidos.

1.5.8 V. S De Oliveira (2015)

Algoritmos genéricos para a geração de modelos dinâmicos de cabos umbilicais para aplicações subaquáticas

A dissertação trata de desenvolver uma proposta de acoplamento das dinâmicas de cabos umbilicais e veículos subaquáticos do tipo ROV, a partir de algoritmos genéricos capazes de gerar modelos dinâmicos para cabos contendo n números de elos. A partir das simulações é analisado o comportamento da carga terminal e do cabo umbilical sob a influência de forças e torques aplicados à carga.

1.5.9 R. Aguiar Jr (2016)

Desenvolvimento de uma Interface para Simulação com Animação 3D de Estruturas Flexíveis do Tipo Cabo

A dissertação trata de desenvolver um simulador em três dimensões para a dinâmica de estruturas flexíveis do tipo cabo. A modelagem dinâmica de estruturas flexíveis do tipo cabo é obtida a partir do formalismo discreto, utilizando-se de algoritmos genéricos capazes de gerar modelos dinâmicos para cabos contendo n números de elos. Com base nas simulações virtuais, é analisado o comportamento da carga terminal e do cabo umbilical sob a influência de forças e torques aplicados à carga.

1.6 Bibliografia Comentada Correlata à Computação Gráfica

1.6.1 I. H Manssour e M. Cohen (2006)

Introdução à Computação Gráfica

Este tutorial abrange os principais conceitos e definições da Computação Gráfica, abordando técnicas de realismo, animação e visualização. Exemplos de aplicações gráficas são apresentados como motivação para o estudo e pesquisa nesta área. Todos os passos do processo de visualização são descritos de forma objetiva através da apresentação de imagens. Questões em aberto e tendências para o desenvolvimento de aplicações gráficas também são abordadas.

1.6.2 F. S. Hill e S. M. Kelley (2007)

Computer Graphics Using OpenGL (3rd Edition)

Este livro apresenta os conceitos básicos e técnicas de computação gráfica interativa moderna, auxiliando na redação de programas de aplicação prática. Combina os princípios e as principais técnicas em computação gráfica com exemplos. Atualiza o tratamento de hardware gráfico e algoritmos. Discute o desenvolvimento de videogames. Enfatiza os gráficos interativos. E além de descrever conceitualmente a computação gráfica, explica a matemática subjacente, mostrando como traduzir a matemática para o código do programa.

1.6.3 J. L. McKesson (2012)

Learning Modern 3D Graphics Programming

Este livro oferece uma instrução de nível iniciante sobre o que é programação gráfica, ensinando a renderização programável para programadores gráficos iniciantes. O livro aborda a maioria dos conceitos básicos de renderização 3D. Ele está dividido em uma série de assuntos gerais onde cada assunto contém vários capítulos numerados chamados de tutoriais. Cada tutorial descreve vários conceitos relacionados. Em praticamente todos os casos, cada conceito é demonstrado por um conjunto de código complementar.

1.6.4 J. F. Hughes, A. V. Dam, M. McGuire, D. F. Sklar, J. D. Foley, S. K. Feiner, K. Akeley (2013)

Computer Graphics: Principles and Practice (3rd Edition)

Neste livro os autores explicam os princípios, bem como a matemática e o conhecimento computacional subjacente para se trabalhar com computação gráfica. Os primeiros capítulos mostram imediatamente como criar imagens 2D e 3D, apoiando a experimentação. Capítulos posteriores abrangem uma ampla gama de tópicos, nos quais demonstram abordagens mais sofisticadas. As seções sobre a prática atual de computação gráfica mostram como aplicar determinados princípios em situações comuns, tais como a forma de aproximar uma solução ideal no hardware disponível ou como representar uma estrutura de dados de forma mais eficiente. Os tópicos são reforçados por exercícios, problemas de programação e projetos práticos.

2 MODELAGEM DE VEÍCULOS SUBAQUÁTICOS

Neste capítulo é desenvolvida a modelagem cinemática de veículos subaquáticos, como os UUVs, relacionando posições, velocidades e acelerações entre os referenciais inercial e do corpo, sem considerar as forças atuantes sobre o corpo.

2.1 Cinemática

A cinemática é a parte da mecânica que estuda e descreve os movimentos, sem se preocupar com as suas causas. Seu objetivo é descrever apenas como se movem os corpos. A parte que se preocupa com as causas do movimento é a dinâmica. A ideia de movimento é descrita pela física como a variação, em função do tempo, da posição de um corpo em relação a outro corpo que serve de referência. O movimento é sempre relativo a um referencial. Quando a posição de um corpo varia, em relação a um dado referencial, durante um intervalo de tempo qualquer, diz-se que há movimento. Porém, se a posição do corpo não varia, em relação a um referencial, durante um intervalo de tempo, diz-se que esse corpo está em repouso. Outro conceito que depende fundamentalmente do referencial adotado é o de trajetória. A trajetória de um corpo pode ser entendida como o caminho que ele percorreu durante sucessivos instantes de tempo, ao longo de seu movimento.

Um dos fundamentos da cinemática que dita a mobilidade de um sistema mecânico traz o conceito de graus de liberdade (DOF - *Degrees of Freedom*), em que a mobilidade é determinada de acordo com o número de graus de liberdade que um sistema possui. Os DOF são iguais ao número de parâmetros independentes necessários para determinar uma única posição no espaço em qualquer instante de tempo. Imaginando um ambiente em três dimensões, para se determinar a posição e a orientação de um corpo rígido é necessário obter suas coordenadas lineares (x, y, z) e suas coordenadas angulares (ϕ, θ, ψ) e, portanto, para determinar a posição e orientação deste corpo rígido, as variáveis necessárias são $(x, y, z, \phi, \theta, \psi)$. Portanto, este corpo possui 6 graus de liberdade (6-DOF). Qualquer corpo rígido em três dimensões possui 6-DOF. Os tipos de movimentos que um corpo rígido pode realizar são rotação e translação, ou uma combinação de ambos.

2.1.1 Sistema de Coordenadas

A postura (atitude) de um corpo rígido é definida em relação a um sistema de coordenadas inercial associado a Terra, ou seja, o referencial inercial possui origem em um ponto fixo em relação a Terra. Para a obtenção da postura do corpo rígido é necessário, a partir da referência inercial, obter a sua orientação e, para isso, é definida outra estrutura de referência, no caso a referência móvel, ou do corpo, e que possui origem no centro de gravidade do corpo rígido FOSSEN (1994). Na área naval, é usual adotar a notação definida pela *Society of Naval Architects and Marine Engineers* (SNAME) para postura e movimento. Ela é apresentada na

Tabela 1. Segundo essa notação, o movimento linear se dá segundo as direções de avanço (*surge*), balanço (*sway*) e afundamento (*heave*), enquanto o movimento angular ocorre segundo os eixos de rolagem (*roll*), arfagem (*pitch*) e guinada (*yaw*) (SNAME, 1950; TAVARES, 2003).

Tabela 1: Notação SNAME para o movimento de veículos aquáticos ROCHA (2012)

Postura		Eixo		Velocidade		
Posição	η_1	x_0	avanço (<i>surge</i>)	u_0	ν_1	Linear
		y_0	balanço (<i>sway</i>)	v_0		
		z_0	afundamento (<i>heave</i>)	w_0		
Orientação	η_2	ϕ_0	rolagem (<i>roll</i>)	p_0	ν_2	Algular
		θ_0	arfagem (<i>pitch</i>)	s_0		
		ψ_0	guinada (<i>yaw</i>)	r_0		

onde:

u velocidade linear na direção X_0 (eixo longitudinal), movimento de avanço (*surge*);

v velocidade linear na direção Y_0 (eixo transversal), movimento de balanço (*sway*);

w velocidade linear na direção Z_0 (eixo normal), movimento de oscilação (*heave*);

p velocidade angular na direção X_0 , movimento de rolamento (*roll*);

s velocidade angular na direção Y_0 , movimento de arfagem (*pitch*);

r velocidade angular na direção Z_0 , movimento de guinada (*yaw*).

A velocidade linear ($\vec{\nu}_1$) e a velocidade angular ($\vec{\nu}_2$) podem ser expressas na forma vetorial:

$$\vec{\nu}_1 = [u, v, w]^T, \vec{\nu}_2 = [p, q, r]^T, \vec{\nu} = [\vec{\nu}_1^T, \vec{\nu}_2^T]^T \quad (2.1)$$

Os seguintes vetores são expressos no referencial inercial:

$$\eta_1 = [\dot{x}, \dot{y}, \dot{z}]^T, \eta_2 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T e, \dot{\eta} = [\eta_1, \eta_2]^T \quad (2.2)$$

onde η_1 é o vetor velocidade linear e η_2 é o vetor que expressa a taxa de variação da orientação do veículo, relativa ao sistema inercial.

A modelagem de veículos subaquáticos é obtida por dois sistemas de referências, conforme indica a Figura 2.1. O referencial inercial que possui origem em um ponto fixo em relação a Terra, e o referencial do corpo, ou referencial móvel, que possui origem no centro de gravidade do corpo rígido e seus eixos coordenados coincidentes com os eixos principais do corpo rígido (FOSSSEN, 1994).

As transformações de um referencial para o outro podem ser feitas em ângulos de Euler.

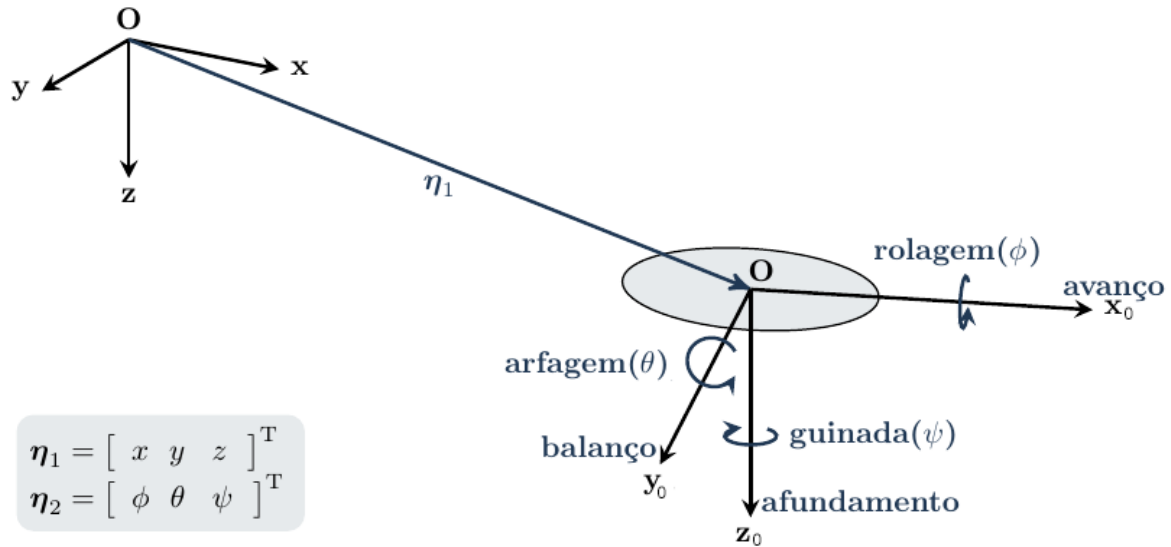


Figura 2.1: Sistema de referencia, adaptado de (ROCHA, 2012).

2.1.2 Transformações por Ângulos de Euler

Rotações principais O método de transformação por Ângulos de Euler consiste em alinhar os referenciais através de rotações sucessivas em torno de cada um dos três eixos coordenados. Essas rotações são denominadas de rotações principais, e os ângulos associados a elas são denominados Ângulos de Euler. Existem três matrizes associadas às rotações principais, que são:

Matriz de rotação em torno do eixo X de um ângulo ϕ representado por \mathbf{B}_ϕ :

$$\mathbf{B}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}; \quad (2.3)$$

Matriz de rotação em torno do eixo Y de um ângulo θ , representada por \mathbf{B}_θ :

$$\mathbf{B}_\theta = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix}; \quad (2.4)$$

Matriz de rotação em torno do eixo Z de um ângulo ψ , representada por \mathbf{B}_ψ :

$$\mathbf{B}_\psi = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad (2.5)$$

Os ângulos ϕ , θ e ψ associados às três rotações são denominados os ângulos de Euler e descrevem a orientação de um corpo rígido em relação a um referencial inercial.

Velocidades

Linear Quando conhecidas as matrizes de rotação principais, a transformação da velocidade linear entre referenciais se dá a partir de sucessivas rotações principais. Para isso, é fundamental respeitar a ordem das rotações. Existem diversas convenções a respeito da ordem das rotações. Neste trabalho é adotada a convenção *xyz* usada por (FOSSEN, 1994), para veículos marinhos.

Supondo-se que a velocidade linear no referencial inercial seja conhecida $\dot{\vec{\eta}}_1 = [\dot{x}, \dot{y}, \dot{z}]^T$, para obter-se a velocidade no referencial do corpo é preciso identificar um sistema de eixos paralelos aos eixos do referencial inercial X_3, Y_3, Z_3 e com origem coincidente com a origem do referencial do corpo X_0, Y_0, Z_0 . A velocidade linear em X_3, Y_3, Z_3 é igual à velocidade no referencial inercial.

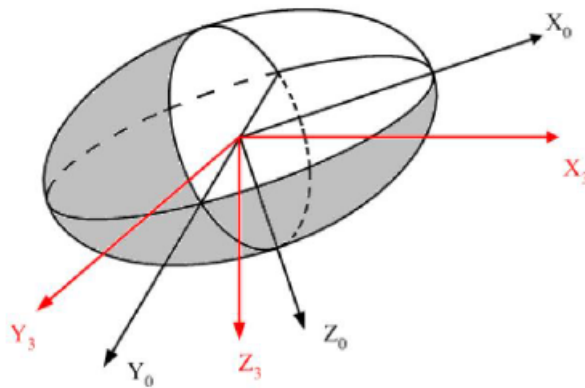


Figura 2.2: Orientação do veículo em relação ao referencial inercial (ROCHA, 2013).

Para efetuar a transformação da velocidade do referencial inercial para o referencial do corpo são feitas rotações de X_3, Y_3, Z_3 em torno de cada um de seus eixos. Pela convenção adotada, é feito inicialmente uma rotação em torno do eixo Z_3 , obtendo-se assim um novo sistema de eixos X_2, Y_2, Z_2 . Após, o referencial X_2, Y_2, Z_2 é rotacionado sobre o eixo Y_2 , gerando o referencial X_1, Y_1, Z_1 . Finalmente, rotacionando X_1, Y_1, Z_1 em torno de X_1 chega-se no referencial do corpo X_0, Y_0, Z_0 .

O produto das matrizes $\mathbf{B}_\phi \mathbf{B}_\theta \mathbf{B}_\psi$ é definido por:

$$\mathbf{J}_1^T(\vec{\eta}_2) = \mathbf{B}_\phi \mathbf{B}_\theta \mathbf{B}_\psi \quad (2.6)$$

de onde se obtêm a seguinte expressão:

$$\vec{v}_1 = \mathbf{J}_1^T(\vec{\eta}_2) \dot{\vec{\eta}}_1 \quad (2.7)$$

na qual a matriz $\mathbf{J}_1^T(\vec{\eta}_2)$ corresponde a matriz de transformação da velocidade linear do referencial inercial para o referencial do corpo (TAVARES, 2003). Efetuando-se o produto das matrizes de rotação $\mathbf{B}_\phi \mathbf{B}_\theta \mathbf{B}_\psi$ obtêm-se a forma final da matriz $\mathbf{J}_1^T(\vec{\eta}_2)$:

$$\mathbf{J}_1^T(\vec{\eta}_2) = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \cos \psi \cos \phi + \sin \phi \sin \theta \sin \psi & \cos \theta \sin \phi \\ \sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta & -\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.8)$$

A matriz $\mathbf{J}_1^T(\vec{\eta}_2)$ é ortogonal FOSSEN (1994), portanto:

$$\mathbf{J}_1^{-1}(\vec{\eta}_2) = \mathbf{J}_1^T(\vec{\eta}_2) \quad (2.9)$$

Então, a transformação da velocidade linear no referencial do corpo para o referencial inercial, é descrita por:

$$\dot{\vec{\eta}}_1 = \mathbf{J}_1^T(\vec{\eta}_2) \vec{\nu}_1 \quad (2.10)$$

A matriz de transformação da velocidade linear do referencial do corpo para o referencial inercial tem a forma:

$$\mathbf{J}_1^T(\vec{\eta}_2) = \begin{bmatrix} \cos \psi \cos \theta & -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta \\ \sin \psi \cos \theta & \cos \psi \cos \phi + \sin \phi \sin \theta \sin \psi & -\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.11)$$

Angular O processo de transformação da velocidade angular difere do processo utilizado para a transformação da velocidade linear, o qual objetiva determinar a relação entre a velocidade angular no referencial do corpo $\vec{\nu}_2 = [p, q, r]^T$, e a taxa de variação dos ângulos de Euler, $\dot{\vec{\eta}}_2 = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$.

Os mesmos sistemas de eixos anteriores são considerados. A velocidade angular do sistema $X_3Y_3Z_3$, Figura 2.2, em torno do eixo Z_3 é expressa no mesmo referencial como $\vec{\omega}_3 = [0, 0, \dot{\psi}]^T$. Utilizando-se das matrizes de rotação, obtêm-se a velocidade angular no referencial do corpo em torno do eixo Z_3 .

Após, é realizado a transformação da velocidade angular do sistema $X_2Y_2Z_2$, em torno de Y_2 , $\vec{\omega}_2 = [0, \dot{\theta}, 0]^T$, para o referencial do corpo. E por fim, a transformação da velocidade angular no sistema $X_1Y_1Z_1$ em torno do eixo X_1 , $\vec{\omega}_1 = [\dot{\phi}, 0, 0]^T$.

O vetor velocidade é então obtido com a soma das três velocidades angulares encontradas no referencial do corpo (TAVARES, 2003):

$$\vec{\nu}_2 = [\dot{\phi}, 0, 0]^T + \mathbf{B}_\phi [0, \dot{\theta}, 0]^T + \mathbf{B}_\phi \mathbf{B}_\theta [0, 0, \dot{\psi}]^T \quad (2.12)$$

Obtendo as operações entre as matrizes de rotação e os vetores de velocidades angulares, têm-se a velocidade angular no referencial do corpo como produto da matriz de transformação

pelo vetor taxa de variação dos Ângulos de Euler no referencial inercial, resultando em:

$$\vec{\nu}_2 = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \dot{\vec{\eta}}_2 \quad (2.13)$$

Para a transformação do referencial do corpo para o referencial inercial é necessário inverter a matriz $\vec{\nu}_2$:

$$\vec{\nu}_2^{-1} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \quad (2.14)$$

Uma característica da matriz de transformação da velocidade angular é que ela não é definida para $\theta = \pm 90^\circ$, que corresponde ao veículo efetuando um movimento de subida ou descida com eixo longitudinal posicionado na posição vertical. Porém, essa singularidade não chega a representar um problema, uma vez que, normalmente, os veículos subaquáticos são projetados de maneira que os centros de gravidade e de flutuação estejam suficientemente afastados para que o ângulo permaneça pequeno (DOMINGUEZ, 1989).

Determina-se a transformação da velocidade angular do referencial do corpo para o referencial inercial por:

$$\dot{\vec{\eta}}_2 = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \vec{\nu}_2 \quad (2.15)$$

Determinando a matriz de transformação em:

$$\mathbf{J}_2(\dot{\vec{\eta}}_2) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix}; \quad (2.16)$$

Resultando em:

$$\dot{\vec{\eta}}_2 = \mathbf{J}_2(\vec{\eta}_2) \vec{\nu}_2 \quad (2.17)$$

2.1.3 Matriz de transformação dos seis graus de liberdade

A transformação das coordenadas do referencial do corpo para o referencial inercial, para os seis graus de liberdade, simultaneamente, pode ser descrita como (ROCHA, 2013):

$$\dot{\vec{\eta}} = J(\vec{\eta}) \vec{\nu} \quad (2.18)$$

onde:

$$\dot{\vec{\eta}} = \begin{bmatrix} \dot{\vec{\eta}}_1 \\ \dot{\vec{\eta}}_2 \end{bmatrix}; \mathbf{J}(\vec{\eta}) = \begin{bmatrix} \mathbf{J}_1(\vec{\eta}_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{J}_2(\vec{\eta}_2) \end{bmatrix} e, \vec{\nu} = \begin{bmatrix} \vec{\nu}_1 \\ \vec{\nu}_2 \end{bmatrix} \quad (2.19)$$

2.2 Dinâmica

Essa seção desenvolve a modelagem dinâmica de veículos subaquáticos, que considera as interações da estrutura com o meio aquático, como força de massa adicional, peso, empuxo, arrasto, sustentação, efeitos da corrente oceânica, e no caso do UUV, também a ação dos propulsores. Integrando veículo com o cabo, é considerado a tração do cabo umbilical.

2.2.1 Dinâmica do Corpo Rígido

A dinâmica consiste em estabelecer relações entre causa e efeito para o movimento de um corpo material, que pode ser interpretado como ponto material ou como corpo rígido, dependendo das dimensões e da sua distribuição de massa.

Como o veículo possui dimensões não desprezíveis, sua massa não está concentrada em um único ponto, mas sim distribuída pela sua estrutura e, portanto, devendo ser tratado como um corpo rígido.

A representação da dinâmica para um corpo rígido possui termos adicionais em relação à dinâmica do ponto material que estão relacionados com o movimento de rotação do corpo em torno dos seus eixos. Estes termos adicionais correspondem à força centrípeta (centrífugos) e à força de Coriolis. As equações matriciais da dinâmica de translação e rotação do corpo rígido podem ser escritas como (YUH, 2000; FOSSEN, 1994):

$$\mathbf{M}_{RB} \dot{\vec{\nu}} + \mathbf{C}_{RB}(\vec{\nu}) \vec{\nu} = \vec{\tau}_{RB} \quad (2.20)$$

onde \mathbf{M}_{RB} é a matriz de inércia, \mathbf{C}_{RB} é a matriz de esforços do tipo Coriolis e centrípeta, $\vec{\nu} = [u, v, w, p, q, r]^T$ é o vetor velocidade e, $\vec{\tau}_{RB} = [X_{RB}, Y_{RB}, Z_{RB}, K_{RB}, M_{RB}, N_{RB}]^T$ é o vetor resultante de todas as forças e momentos aplicados ao veículo, sendo que todas as grandezas estão no referencial do corpo.

Matriz de Inércia A matriz de inércia \mathbf{M}_{RB} é constituída pela massa (m), momentos de inércia (I_X, I_Y, I_Z), no referencial do corpo, produtos de inércia (I_{XY}, I_{YZ}, I_{XZ}) e coordenadas do centro de gravidade no referencial do corpo $\vec{\mathbf{r}}_G = [X_G, Y_G, Z_G]^T$. É uma matriz positiva

definida e possui uma única parametrização possível (FOSSEN, 1994; FJELLSTAD, 1994):

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mZ_G & -mY_G \\ 0 & m & 0 & -mZ_G & 0 & mX_G \\ 0 & 0 & m & mY_G & -mX_G & 0 \\ 0 & -mZ_G & mY_G & I_X & -I_{XY} & -I_{XZ} \\ mZ_G & 0 & -mX_G & -I_{YX} & I_Y & -I_{YZ} \\ -mY_G & mX_G & 0 & -I_{ZX} & -I_{ZY} & I_Z \end{bmatrix} \quad (2.21)$$

Matriz de Coriolis e centrípeta A matriz de Coriolis e centrípeta do corpo rígido pode ter várias parametrizações, diferentemente da matriz de inércias. Uma parametrização possível possui a forma (GOULART, 2007):

$$\mathbf{C}_{RB}(\vec{\nu}) = \begin{bmatrix} 0_{3 \times 3} & \mathbf{C}_1 \\ -\mathbf{C}_1^T & \mathbf{C}_2 \end{bmatrix} \quad (2.22)$$

onde

$$\mathbf{C}_1 = \begin{bmatrix} m(Y_G q + Z_G r) & -m(X_G q - w) & -m(X_G r + \nu) \\ -m(Y_G p + w) & m(Z_G r + X_G p) & -m(Y_G p - u) \\ -m(Y_G p - \nu) & -m(Z_G q + u) & m(X_G p + Y_G q) \end{bmatrix} \quad (2.23)$$

$$\mathbf{C}_2 = \begin{bmatrix} 0 & -I_{YZq} - I_{XZp} - I + Zr & I_{YZr} + I_{XYp} - I_Y q \\ I_{YZq} + I_{XZp} - I_Z r & 0 & -I_{XZr} - I_{XYq} + I_X p \\ I_{YZr} - I_{XZp} + I_Y q & I_{XZr} + I_{XYq} - I_X p & 0 \end{bmatrix} \quad (2.24)$$

Com essa parametrização, pode-se concluir que a matriz \mathbf{C}_{RB} é antissimétrica, ou seja:

$$\mathbf{C}_{RB}(\vec{\nu}) = -\mathbf{C}_{RB}(\vec{\nu})^T \quad (2.25)$$

2.2.2 Força de massa Adicional

Os esforços de massa adicional estão relacionados com o fluido que envolve o corpo do veículo quando este está submerso. Quando ele se desloca, deve vencer a inércia do fluido, além da sua própria inércia, fazendo com que o veículo aparente possuir uma massa superior a sua massa real. Para FOSSEN (1994), a expressão do esforço devido a massa adicional $\vec{\tau}_a$ em função das matrizes de inércia \mathbf{M}_A e de Coriolis centrípeta $\mathbf{C}_A(\vec{\nu})$ pode ser obtida da seguinte maneira:

$$\vec{\tau}_a = -\mathbf{M}_A \dot{\vec{\nu}} - \mathbf{C}_A(\vec{\nu}) \vec{\nu} \quad (2.26)$$

onde $\vec{\tau}_a = [X_A, Y_A, Z_A, K_A, M_A, N_A]^T$ é o vetor de forças e momentos de massa adicional, e $\vec{\nu} = [u, \nu, p, q, r]^T$ é o vetor velocidade.

Matriz de inércia de massa adicional Para o caso de veículos com três planos de simetria, xz, yz, xy , considera-se a matriz de inércia de massa adicionada \mathbf{M}_A como diagonal. Seus coeficientes estão associadas às acelerações, variações dos esforços externos e a geometria do veículo. Devido a complexidade de obtenção dos coeficientes para todos os estados de operação possíveis do veículo, adota-se coeficientes constantes (FOSSSEN, 1994). Portanto, a matriz de inércia possui a seguinte parametrização:

$$\mathbf{M}_A = \begin{bmatrix} A_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{66} \end{bmatrix} \quad (2.27)$$

Matriz de Coriolis e centrípeta de massa adicional A matriz de Coriolis e centrípeta de massa adicional $\mathbf{C}_A(\vec{\nu})$ possui termos dependentes das componentes do vetor velocidade do veículo ($\vec{\nu}$), e pode apresentar diferentes parametrizações. Uma possível parametrização é:

$$\mathbf{C}_A(\vec{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & A_{33}w & -A_{22}\nu \\ 0 & 0 & 0 & -A_{33}\nu & 0 & A_{11}u \\ 0 & 0 & 0 & A_{22}\nu & -A_{11}u & 0 \\ 0 & A_{33}w & -A_{22}\nu & 0 & A_{66}r & -A_{55}q \\ -A_{33}w & 0 & A_{11}u & -A_{66}r & 0 & A_{44}p \\ A_{22}\nu & -A_{11}u & 0 & A_{55}q & -A_{44}p & 0 \end{bmatrix} \quad (2.28)$$

2.2.3 Forças de arrasto hidrodinâmico e sustentação

O esforço hidrodinâmico devido ao arrasto e sustentação são dependentes de dimensões e forma do corpo do veículo, rugosidade da sua superfície e velocidade de operação. A força de arrasto é paralela à velocidade do veículo em relação ao fluido e a força de sustentação é ortogonal à velocidade em relação ao fluido. Existem diferentes modelos para identificação das forças de arrasto e sustentação. TAVARES (2003) observou dois modelos aplicados aos veículos (NEROV e Tatuí), o modelo de DOMINGUEZ (1989) e o de FOSSSEN (1994). Dominguez apresenta um modelo considerado completo, porém mais lento, enquanto Fossen é considerado mais simples, porém, mais rápido. TAVARES (2003) constatou que não houve diferenças significativas entre os modelos, optando-se pelo modelo de Fossen, devido ao menor tempo de simulação.

A força de arrasto F_D pode ser descrita da seguinte forma (TAVARES, 2003; FOSSSEN, 1994):

$$F_D = D_S\nu + D_Q|\nu|\nu \quad (2.29)$$

onde F_D é a força de arrasto, D_S e D_Q são coeficientes de arrasto linear e coeficiente de arrasto

quadrático, respectivamente, e ν é a velocidade do veículo em relação ao fluido.

A força de sustentação é definida por F_L :

$$F_L = \frac{1}{2} \rho C_L(R_\eta \alpha) A(\alpha) |\nu| \nu \quad (2.30)$$

onde $C_L(R_\eta \alpha)$ é o coeficiente de sustentação, ρ é a massa específica do fluido e $A(\alpha)$ é a área frontal projetada do veículo.

As equações das forças e momentos obtidos em função do arrasto e sustentação, produzidas nas direções X_0, Y_0, Z_0 podem ser obtidas pelas seguintes equações:

$$\begin{aligned} X_{DL} &= \frac{1}{2} \rho U^2 \nabla^{2/3} C_x(\alpha, \beta) \\ Y_{DL} &= \frac{1}{2} \rho U^2 \nabla^{2/3} C_y(\beta, \gamma) \\ Z_{DL} &= \frac{1}{2} \rho U^2 \nabla^{2/3} C_z(\alpha, \gamma) \end{aligned} \quad (2.31)$$

$$\begin{aligned} K_{DL} &= \frac{1}{2} \rho U^2 \nabla C_k(\gamma) + \frac{1}{2} \rho \nabla^{5/3} C_p p |p| \\ M_{DL} &= \frac{1}{2} \rho U^2 \nabla C_m(\alpha) + \frac{1}{2} \rho \nabla^{5/3} C_q q |q| \\ N_{DL} &= \frac{1}{2} \rho U^2 \nabla C_n(\alpha) + \frac{1}{2} \rho \nabla^{5/3} C_r r |r| \end{aligned} \quad (2.32)$$

onde as equações (2.31) são as forças resultantes de arrasto e sustentação, e as equações (2.32) são os momentos produzidos pelas forças resultantes de arrasto e sustentação, ambas equações respectivamente nas direções X_0, Y_0 e Z_0 . A massa específica do fluido é ρ , ∇ é o volume do veículo, U é o módulo da velocidade linear do veículo, (C_p, C_q e C_r) definem os coeficientes de amortecimento de rotação de veículo, (C_x, C_y e C_z) são os coeficientes das forças de arrasto e sustentação, (C_k, C_m e C_n) são os coeficientes dos momentos produzidos pelas forças de arrasto e sustentação, α é o ângulo de ataque, β é o ângulo de deriva e γ é o ângulo de ataque lateral.

As equações das forças e momentos obtidos em função do arrasto e sustentação, podem ser descritas em formato matricial-vetorial:

$$\vec{\tau}_{DL} = -\mathbf{D}(\vec{\nu}) \vec{\nu} \quad (2.33)$$

onde $\vec{\tau}_{DL} = [X_{DL}, Y_{DL}, Z_{DL}, K_{DL}, M_{DL}, N_{DL}]^T$, $\mathbf{D}(\vec{\nu})$ é a matriz de arrasto e sustentação e $\vec{\nu} = [u, v, w, p, q, r]^T$ é o vetor de velocidade no referencial do corpo.

A matriz $\mathbf{D}(\vec{\nu})$ de arrasto e sustentação possui o seguinte formato (TAVARES, 2003; DOMINGUEZ, 1989):

$$\mathbf{D}(\vec{\nu}) = \begin{bmatrix} \nabla^{\frac{2}{3}} u C_x(\alpha, \beta) & \nabla^{\frac{2}{3}} v C_x(\alpha, \beta) & \nabla^{\frac{2}{3}} w C_x(\alpha, \beta) & 0 & 0 & 0 \\ \nabla^{\frac{2}{3}} u C_y(\beta, \gamma) & \nabla^{\frac{2}{3}} v C_y(\beta, \gamma) & \nabla^{\frac{2}{3}} w C_y(\beta, \gamma) & 0 & 0 & 0 \\ \nabla^{\frac{2}{3}} u C_z(\alpha, \gamma) & \nabla^{\frac{2}{3}} v C_z(\alpha, \gamma) & \nabla^{\frac{2}{3}} w C_z(\alpha, \gamma) & 0 & 0 & 0 \\ \nabla u C_k(\gamma) & \nabla v C_k(\gamma) & \nabla w C_k(\gamma) & \nabla^{\frac{5}{3}} C_p |p| & 0 & 0 \\ \nabla u C_m(\alpha) & \nabla v C_m(\alpha) & \nabla w C_m(\alpha) & 0 & \nabla^{\frac{5}{3}} C_q |q| & 0 \\ \nabla u C_n(\beta) & \nabla v C_n(\beta) & \nabla w C_n(\beta) & 0 & 0 & \nabla^{\frac{5}{3}} C_r |r| \end{bmatrix} \quad (2.34)$$

2.2.4 Forças restauradoras

A ação gravitacional (ou peso) e o empuxo hidrostático correspondem as forças restaurativas. Elas atuam sempre na direção vertical do referencial inercial e possuem sentidos contrários.

Os vetores da força peso f_W e da força empuxo f_B , no referencial do veículo são expressos por:

$$\vec{f}_W = \mathbf{J}_1^{-1}[0, 0, W]^T \quad (2.35)$$

$$\vec{f}_B = \mathbf{J}_1^{-1}[0, 0, B]^T \quad (2.36)$$

onde \mathbf{J}_1^{-1} é a matriz inversa de transformação \mathbf{J}_1 , e é definida como:

$$\mathbf{J}_1 = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \phi + \cos \psi \sin \theta \sin \phi & \cos \psi \cos \phi + \sin \phi \sin \theta \sin \psi & \cos \theta \sin \phi \\ \sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta & -\cos \psi \sin \phi + \sin \theta \sin \psi \cos \phi & \cos \theta \sin \phi \end{bmatrix} \quad (2.37)$$

onde ψ , θ e ϕ são os ângulos em relação aos eixos do referencial inercial.

Das equações (2.35) e (2.36) obtêm-se as seguintes componentes de módulo, respectivamente:

$$W = m \cdot g \quad (2.38)$$

$$B = \rho g \nabla \quad (2.39)$$

As forças de peso W e empuxo B dependem da massa m e do volume do veículo ∇ , de massa específica do fluido ρ e da aceleração da gravidade g . A força peso atua sobre o centro de gravidade do veículo, onde sua posição é definida pelo vetor $\vec{r}_G = [x_G, y_G, z_G]^T$, já a força de empuxo que atua no centro do empuxo (centro do volume) possui seu vetor posição definido por $\vec{r}_B = [X_B, y_B, Z_B]^T$. Os vetores posição estão no referencial do corpo. Então, o momento do peso \vec{M}_W e o momento do empuxo \vec{M}_B em relação a origem do veículo são respectivamente:

$$\vec{M}_W = \vec{r}_G \times \vec{f}_W \quad (2.40)$$

$$\vec{M}_B = \vec{r}_B \times \vec{f}_B \quad (2.41)$$

Das equação (2.35) até a equação (2.41), obtêm-se a expressão geral das forças e momentos

restauradores no referencial do corpo (FOSSEN, 1994; TAVARES, 2003):

$${}^t\vec{\tau}_{WB} = \begin{bmatrix} \vec{f}_W + \vec{f}_B \\ \vec{M}_W + \vec{M}_B \end{bmatrix} = - \begin{bmatrix} (W - B) \sin \theta \\ -(W - B) \cos \theta \sin \phi \\ -(W - B) \cos \theta \cos \phi \\ -(y_G W - y_B B) \cos \theta \cos \phi + (z_G W - z_B B) \cos \theta \sin \phi \\ (z_G W - z_B B) \sin \theta + (x_G W - x_B B) \cos \theta \cos \phi \\ -(x_G W - x_B B) \cos \theta \sin \phi - (y_G W - y_B B) \sin \theta \end{bmatrix} \quad (2.42)$$

2.3 Veículos Subaquáticos

Veículos são modelados como um corpo rígido que possui movimentação de seis graus de liberdade (6-DOF), três lineares (direção dos eixos coordenados) e três angulares (rotação em torno do eixo) (ROCHA, 2013). Esses movimentos sofrem ação de diversas forças externas, como corrente marítima, ondas e tração do cabo umbilical. A dinâmica do veículo é presente no referencial do corpo, e obtida com transformações das velocidades para o referencial inercial.

2.3.1 Modelo sem corrente oceânica

A literatura propõem diversos modelos dinâmicos de veículos. FOSSEN (1994) detalha o desenvolvimento sobre a dinâmica e cinemática. As abordagens mais recentes para o modelo presente em FOSSEN (1994); ANTONELLI (2013) que representa o equilíbrio das forças e torques atuando no veículo é dada por:

$$\mathbf{M}_{RB} \dot{\vec{v}} + \mathbf{C}_{RB}(\vec{v})\vec{v} = \vec{t}_{RB} \quad (2.43)$$

Na equação (2.43) \mathbf{M}_{RB} é a matriz de inércia, \mathbf{C}_{RB} é a matriz de esforços do tipo Coriolis e centrífugos, $\vec{v} = [u, v, w, p, q, r]^T$ é o vetor velocidade e, $\vec{t}_{RB} = [X_{RB}, Y_{RB}, Z_{RB}, K_{RB}, M_{RB}, N_{RB}]^T$ é o vetor resultante de todas as forças e momentos aplicados ao veículo. Todos os termos da equação estão em relação ao sistema do corpo. A equação pode ser reescrita como:

$$(\mathbf{M}_{RB} + \mathbf{M}_A) \dot{\vec{v}} + (\mathbf{C}_{RB}(\vec{v}) + \mathbf{C}_A(\vec{v}))\vec{v} + \mathbf{D}(\vec{v})\vec{v} + \vec{g} = \vec{t} \quad (2.44)$$

Na equação (2.44), \mathbf{M}_A e $\mathbf{C}_A(\vec{v})$ são contribuições devidas à massa adicional na matriz de inércia e nos termos Coriolis e centrípeta, respectivamente. $\mathbf{D}(\vec{v})\vec{v}$ são os termos relativos ao arrasto e à sustentação, e \vec{g} contém os esforços devidos ao peso e ao empuxo.

Definindo: $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$ e $\mathbf{C}(\vec{v}) = \mathbf{C}_{RB}(\vec{v}) + \mathbf{C}_A(\vec{v})$ e considerando as relações cinemáticas entre os referenciais móvel (do corpo) e inercial, o modelo do veículo, expresso no referencial do corpo, pode ser escrito na seguinte forma:

$$\mathbf{M} \dot{\vec{v}} + \mathbf{C}(\vec{v})\vec{v} + \mathbf{D}(\vec{v})\vec{v} + \vec{g} = \vec{t} \quad (2.45)$$

onde:

- \mathbf{M} é matriz de inércia;
- $\mathbf{C}(\vec{\nu})$ é matriz Coriolis e centrípeta;
- $\mathbf{D}(\vec{\nu})$ é matriz de arrasto e sustentação;
- $\vec{\mathbf{g}}$ é o vetor de forças e momentos restauradores;
- $\vec{\boldsymbol{\tau}}$ é o vetor de forças e momentos dos propulsores;
- $\vec{\nu}$ é o vetor de velocidades no referencial do corpo.

2.3.2 Modelo com corrente oceânica

Quando considerado corrente oceânica \vec{v}_C^E a velocidade da corrente pode ser indicada, no referencial inercial pelo vetor $\vec{v}_C^E = [u_C^E, v_C^E, w_C^E, 0, 0, 0]^T$, desprezando o movimento de rotação provocado pela corrente. Passando o vetor para o referencial do corpo, utilizando Ângulos de Euler, têm-se:

$$\vec{\nu}_C = \mathbf{J}^{-1}(\vec{\eta})\vec{v}_C^E \quad (2.46)$$

O vetor $\vec{\nu}_C = [u_C, v_C, w_C, 0, 0, 0]^T$ corresponde a velocidade da corrente no referencial do corpo. A diferença entre a velocidade do veículo $\vec{\nu}$ e a velocidade da corrente $\vec{\nu}_C$ é denominada velocidade relativa e indicada por $\vec{\nu}_r$.

$$\vec{\nu}_r = \vec{\nu} - \vec{\nu}_C \quad (2.47)$$

Substituindo a velocidade relativa à corrente no lugar da velocidade do veículo no modelo dinâmico expresso na equação (2.45), obtém-se a seguinte equação:

$$\mathbf{M}\dot{\vec{\nu}}_r + \mathbf{C}(\vec{\nu}_r)\vec{\nu}_r + \mathbf{D}(\vec{\nu}_r)\vec{\nu}_r + \vec{\mathbf{g}}(\vec{\eta}) = \vec{\boldsymbol{\tau}} \quad (2.48)$$

Considerando $\vec{\nu} = \vec{\nu}_r + \vec{\nu}_c$, e que a taxa de variação da corrente no referencial pode ser considerada nula, ou seja, $\dot{\vec{\nu}}_c = 0$, têm-se que $\dot{\vec{\nu}} = \dot{\vec{\nu}}_r$. Dessa forma o modelo dinâmico com a corrente oceânica é descrito pelas equações:

$$\begin{aligned} \mathbf{M}\dot{\vec{\nu}} + \mathbf{C}(\vec{\nu}_r)\vec{\nu}_r + \mathbf{D}(\vec{\nu}_r)\vec{\nu}_r + \vec{\mathbf{g}}(\vec{\eta}) &= \vec{\boldsymbol{\tau}} && \text{(dinâmica)} \\ \dot{\vec{\eta}} &= \mathbf{J}(\vec{\eta})\vec{\nu} && \text{(cinemática)} \end{aligned} \quad (2.49)$$

onde:

- \mathbf{M} é matriz de inércia;
- $\mathbf{C}(\vec{\nu}_r)$ é matriz de Coriolis e centrípeta;
- $\mathbf{D}(\vec{\nu}_r)$ é matriz de arrasto e sustentação;

- $\vec{g}(\vec{\eta})$ é o vetor de forças e momentos restauradores;
- $\boldsymbol{\eta}$ é o vetor de posições e orientações no referencial inercial ($\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^T$);
- $\dot{\boldsymbol{\eta}}$ é o vetor de velocidades no referencial inercial ($\dot{\boldsymbol{\eta}} = [\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$);
- $\vec{\tau}$ é o vetor de forças e momentos dos propulsores;
- $\mathbf{J}(\vec{\eta})$ é a matriz de transformação por ângulos de Euler;
- $\vec{\nu}$ é o vetor de velocidades no referencial do corpo ($\boldsymbol{\nu} = [u, v, w, p, q, r]^T$) em relação a corrente marinha.

2.3.3 Modelo dinâmico dos propulsores

É necessário observar que o modelo dinâmico do veículo contempla a modelagem dinâmica dos propulsores, que é a força e momentos de propulsão decorrente da ação de cada propulsor e são expressos como funções da velocidade dos propulsores, descrito como:

$$\vec{\tau} = \mathbf{B}\vec{u} \quad (2.50)$$

onde $\vec{\tau}$ é o vetor de forças e momentos de propulsão, sendo p o número de propulsores, \vec{u} é um vetor de dimensão $p \times 1$ com elementos $u_i = |n_i| n_i$ e \mathbf{B} uma matriz $6 \times p$, em que os seus elementos estão distribuídos em função da localização dos seus propulsores.

A equação para a dinâmica dos propulsores pode ser descrita como:

$$\mathbf{P}\dot{\vec{\eta}} = \vec{\tau}_m - \vec{\tau}_r \quad (2.51)$$

onde, \mathbf{P} é uma matriz diagonal que contém os momento de inércia dos propulsores, $\dot{\vec{\eta}}$ é o vetor de aceleração angular das hélices, $\vec{\tau}_m$ é o vetor de torques motores e $\vec{\tau}_r$ é o vetor de torques resistentes.

Acoplando a dinâmica dos propulsores na dinâmica do veículo, equação (2.49), obtêm-se uma novo modelo dinâmico, descrito por:

$$\begin{aligned} \mathbf{M}\dot{\vec{\nu}} + \mathbf{C}(\vec{\nu}_r)\vec{\nu}_r + \mathbf{D}(\vec{\nu}_r)\vec{\nu}_r + \vec{g}(\vec{\eta}) &= \vec{\tau} && \text{(dinâmica)} \\ \dot{\vec{\eta}} &= \mathbf{J}(\vec{\eta})\vec{\nu} && \text{(cinemática)} \\ \mathbf{P}\dot{\vec{\eta}} &= \vec{\tau}_m - \vec{\tau}_r && \text{(propulsores)} \end{aligned} \quad (2.52)$$

A integração conjunta das três equações (2.52) obtêm as posições lineares e angulares $\boldsymbol{\eta}$, do veículo subaquático no referencial inercial. Os detalhes da modelagem de veículos subaquáticos podem ser encontrados em TAVARES (2003).

Com os conceitos teóricos apresentados, pode-se desenvolver modelos dinâmicos para veículos subaquáticos, independentemente de suas dimensões ou formas. Dentre as definições necessárias para um veículo subaquático alguns parâmetros se diferem entre os veículos, principalmente os

parâmetros devido ao formato do casco, número de propulsores, utilização ou não de lemes de controle e existência ou não de cabo umbilical.

Este trabalho, descreve o desenvolvimento do modelo dinâmico do veículo denominado NEROV (*Norwegian Experimental Remotely Operated*) (FOSSEN; BALCHEN et al., 1991; FOSSEN, 1994; FJELLSTAD, 1994), pois os seus parâmetros são conhecidos e difundidos na literatura. O trabalho de modelagem, acompanhado de simulações computacionais, descrito por TAVARES (2003), é apresentado na próxima seção.

2.4 Modelo dinâmico do NEROV

Esta seção apresenta a modelagem do veículo subaquático denominado NEROV.

O NEROV, Figura 2.3, é um veículo autônomo que foi projetado e construído pelo Departamento de Engenharia Cibernética do Instituto Norueguês de Tecnologia (NTH), com intuito de desenvolver e testar controle para veículos subaquáticos (FOSSEN; BALCHEN et al., 1991).

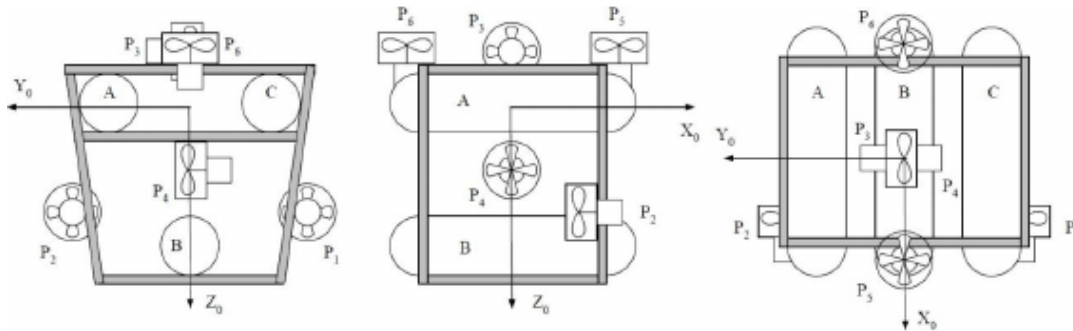


Figura 2.3: Desenho simplificado do NEROV (TAVARES, 2003)

Segundo a Figura 2.3, que apresenta o desenho simplificado do veículo e com base nas características descritas por FOSSEN (1994), têm-se que o NEROV é um veículo com seis propulsores e controle ativo nos seis graus de liberdade.

O modelo dinâmico utilizado para modelar o veículo NEROV é descrito de forma resumida pela equação geral (2.52), definida por:

$$\begin{aligned}
 \mathbf{M}\dot{\vec{\nu}} + \mathbf{C}(\vec{\nu}_r)\vec{\nu}_r + \mathbf{D}(\vec{\nu}_r)\vec{\nu}_r + \vec{\mathbf{g}}(\vec{\eta}) &= \vec{\tau} && \text{(dinâmica)} \\
 \dot{\vec{\eta}} &= \mathbf{J}(\vec{\eta})\vec{\nu} && \text{(cinemática)} \\
 \mathbf{P}\ddot{\vec{\eta}} &= \vec{\tau}_m - \vec{\tau}_r && \text{(propulsores)}
 \end{aligned} \tag{2.53}$$

onde:

- \mathbf{M} é matriz de inércia;
- $\mathbf{C}(\vec{\nu}_r)$ é matriz Coriolis e centrípeta;
- $\mathbf{D}(\vec{\nu}_r)$ é matriz de arrasto e sustentação;
- $\vec{\mathbf{g}}(\vec{\eta})$ é o vetor de forças e momentos restauradores;

- $\boldsymbol{\eta}$ é o vetor de posições e orientações no referencial inercial ($\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^T$);
- $\dot{\boldsymbol{\eta}}$ é o vetor de velocidades no referencial inercial ($\dot{\boldsymbol{\eta}} = [\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$);
- $\vec{\boldsymbol{\tau}}$ é o vetor de forças e momentos dos propulsores;
- $\mathbf{J}(\vec{\boldsymbol{\eta}})$ é a matriz de transformação por ângulos de Euler;
- $\vec{\boldsymbol{v}}$ é o vetor de velocidades no referencial do corpo ($\boldsymbol{v} = [u, v, w, p, q, r]^T$) em relação a corrente marinha.

O modelo apresentado pelas equações (2.53), possui como entrada os torques motores, que dependem da corrente elétrica alimentada nos propulsores. O vetor de forças e momentos dos propulsores é determinado com as velocidades de rotação induzidas nos propulsores pelos torques motores aplicados. Como resultado do modelo para veículos subaquáticos, equações (2.53), têm-se de saída o vetor velocidade relativa, que é expresso no referencial do corpo, e o vetor posição e orientação, expresso no referencial inercial.

O modelo dinâmico de um veículo subaquático é complexo, ele é não linear, multivariável e acoplado, já que os seis graus de liberdade estão acoplados. A dinâmica do veículo nos seis graus de liberdade é expressa por seis equações diferenciais de segunda ordem.

Com base no modelo, TAVARES (2003) apresenta a modelagem sobre as forças e momentos de propulsão, que descreve na forma matricial-vetorial, equação 2.50, onde $\vec{\boldsymbol{\tau}} = [X_p, Y_p, Z_p, K_p, M_p, N_p]^T$, $\vec{\boldsymbol{u}}$ é o vetor de controle e \mathbf{B} é a matriz de propulsão, respectivamente definidos como:

$$\vec{\boldsymbol{u}} = [u_1, u_2, u_3, u_4, u_5, u_6]^T \quad (2.54)$$

$$\mathbf{B} = \begin{bmatrix} T_{|n|n} & -T_{|n|n} & 0 & 0 & 0 & 0 \\ 0 & 0 & T_{|n|n} & T_{|n|n} & 0 & 0 \\ 0 & 0 & 0 & 0 & T_{|n|n} & -T_{|n|n} \\ Q_{|n|n} & Q_{|n|n} & -z_{p3}T_{|n|n} & -z_{p4}T_{|n|n} & y_{p5}T_{|n|n} & -y_{p6}T_{|n|n} \\ z_{p1}T_{|n|n} & -z_{p2}T_{|n|n} & Q_{|n|n} & -Q_{|n|n} & -x_{p5}T_{|n|n} & x_{p6}T_{|n|n} \\ -y_{p1}T_{|n|n} & y_{p2}T_{|n|n} & x_{p3}T_{|n|n} & x_{p4}T_{|n|n} & Q_{|n|n} & Q_{|n|n} \end{bmatrix} \quad (2.55)$$

Como resultado do modelo de veículos subaquáticos e a modelagem referente ao NEROV, TAVARES (2003) apresenta diversas simulações em malha aberta que permitem o entendimento do seu comportamento dinâmico.

3 CONTROLE DE VEÍCULOS SUBAQUÁTICOS

O controle de veículos subaquáticos de posição e orientação supervisiona as coordenadas do veículo, para determinar a sua posição e orientação, como em situações que se deseja mantê-lo parado, ou se deseja levar o veículo até determinada posição e mantê-lo enquanto este realiza determinada tarefa. As trajetórias de referência para o controle, em alguns casos, são geradas pelo próprio operador do veículo que o controla a partir de um *joystick*. Existe ainda o controle automático, onde o operador somente define as posições e orientações desejadas para o veículo. Quando os sistemas de controle são em malha fechada, utilizam sensores par auxiliar o controle, obtendo assim, um desempenho melhor do que os sistemas de malha aberta (FOSSEN, 1994). Os veículos utilizam além do controle de posição e orientação, o controle de velocidade e controle para o seguimento de trajetória (FJELLSTAD, 1994). O projeto de controlador de veículos subaquáticos é complexo, HSU (2000) e ANTONELLI; LEONESSA (2008) citam como principais fatores que dificultam o controle, as não linearidades do veículo, dinâmica variável com o comportamento do veículo e os inúmeros distúrbios causados pelas correntes oceânicas.

3.1 Equação diferencial de estado

As modelagens cinemática e dinâmica demonstram que a interação entre o veículo e o fluido podem ser modelados a partir de duas equações diferenciais de primeira ordem, na forma matricial-vetorial, fornecendo a aceleração do veículo em relação ao fluido, no referencial do corpo, e a aceleração absoluta do veículo, no referencial inercial. Adicionando outra equação diferencial de primeira ordem, fornece as acelerações angulares dos propulsores, como visto na equação, 2.52. Agrupando essas três equações diferenciais, forma-se uma equação diferencial de estado, com a seguinte forma:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\boldsymbol{\nu}}_r \\ \dot{\boldsymbol{\eta}} \\ \dot{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} \mathbf{M}^{-1} [\boldsymbol{\tau} + \boldsymbol{\tau}_c - \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \mathbf{g}(\boldsymbol{\eta})] \\ \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu} \\ \mathbf{P}^{-1}(\boldsymbol{\tau}_m - \boldsymbol{\tau}_r) \end{bmatrix} \quad (3.1)$$

onde $\mathbf{x} = [\boldsymbol{\nu}_r^T, \boldsymbol{\eta}^T, \mathbf{n}^T]^T$ é o vetor de estado do sistema, em que $\boldsymbol{\nu}_r$ é o vetor velocidade do veículo considerando o fluido e expresso no referencial do corpo, $\boldsymbol{\eta}$ o vetor com as coordenadas de posição e orientação, no referencial inercial, e \mathbf{n} sendo o vetor velocidade dos propulsores.

As demais grandezas têm os seguintes significados:

- \mathbf{M} é matriz de inércia do veículo e \mathbf{P} a matriz de inércia dos propulsores;
- $\mathbf{C}(\boldsymbol{\nu}_r)$ é matriz Coriolis e centrípeta;
- $\mathbf{D}(\boldsymbol{\nu}_r)$ é matriz de arrasto e sustentação;
- $\boldsymbol{\nu}$ é o vetor de velocidades no referencial do corpo em relação a corrente marinha.

- $\vec{\tau}$, $\vec{\tau}_c$ e $\vec{g}(\vec{\eta})$ são vetores de forças e momentos produzidos por, respectivamente, propulsores, cabo umbilical, peso e empuxos;
- $\mathbf{J}(\vec{\eta})$ é a matriz de transformação do referencial do corpo para o referencial inercial por ângulos de Euler;
- $\vec{\tau}_m$ é o vetor de torques motores e $\vec{\tau}_r$ é o vetor torques resistentes, associados aos propulsores;

A simulação é realizada através da integração da equação diferencial de estado. A complexidade da equação traz a necessidade de se utilizar um método numérico, como Método de Euler ou algum dos Métodos de Runge-Kutta. TAVARES (2003) descreve a opção utilizada para a solução numérica com uma modificação do método Runge-Kutta, onde efetua a alteração da função *ode45* de forma a simular um controle discreto. A função *ode45* utiliza um integrador Runge-Kutta de 4a e 5a ordem. A simulação é desenvolvida sobre o software MATLAB (MATrix LABoratory) (MATHWORKS, 2015), que trata-se de um software interativo de alta performance voltado para o cálculo numérico.

3.2 Linearização por Realimentação

Esta seção apresenta o desenvolvimento de um sistema de controle de posição e orientação a partir da linearização por realimentação (SLOTINE et al., 1991; FOSSEN, 1994).

Linearização por realimentação é uma técnica de controle utilizada em sistemas não lineares. A ideia por trás desse método de controle consiste na transformação algébrica do sistema dinâmico não linear, mediante uma lei de controle adequada, em um sistema linear em malha fechada, possibilitando o uso das técnicas de controle linear, mais usuais e conhecidas, no projeto do controlador. Esta técnica também pode ser entendida como a escolha de uma estratégia de controle que permita a transformação do sistema dinâmico original em um sistema dinâmico equivalente, porém mais simples (SLOTINE et al., 1991).

Para realizar a linearização por realimentação aplica-se no veículo forças e momentos que cancelem os termos não lineares responsáveis pelo acoplamento dinâmico.

A lei de controle utilizada é a seguinte (FOSSEN, 1994):

$$\vec{\tau} = \mathbf{M}\vec{a}_v + \vec{n}(\vec{v}, \vec{\eta}) \quad (3.2)$$

onde \vec{a}_v é o vetor de aceleração comandada, \vec{v} é o vetor velocidade do veículo e $\vec{n}(\vec{v}, \vec{\eta})$ é o vetor de cancelamento das não linearidades do sistema, expressos no referencial do corpo.

Os termos não lineares e acoplados associados a veículos subaquáticos são: forças e momentos de Coriolis e centrípeta; forças e momentos de arrasto e sustentação; forças e momentos restauradores. Portanto, o vetor de cancelamento das não linearidades, da equação 3.2, é dado por:

$$\vec{\mathbf{n}}(\vec{\nu}, \vec{\eta}) = \mathbf{C}(\vec{\nu})\vec{\nu} + \mathbf{D}(\vec{\nu})\vec{\nu} + \vec{\mathbf{g}}(\vec{\eta}) \quad (3.3)$$

Como não é possível tornar o sistema perfeitamente linear devido as não-linearidades como os distúrbios ambientais, que não podem ser perfeitamente previstos em razão das diferenças entre o modelo adotado e o sistema físico real, o que se faz é tornar a resposta tão linear quanto possível. Ao desprezar a corrente marinha, a dinâmica do veículo é representada pelas seguintes equações diferenciais:

$$\mathbf{M}\dot{\vec{\nu}} + \vec{\mathbf{n}}(\vec{\nu}, \vec{\eta}) = \vec{\tau} \quad (3.4)$$

$$\dot{\vec{\eta}} = \mathbf{J}(\vec{\eta})\vec{\nu} \quad (3.5)$$

Ao substituir 3.4 em 3.5 chega-se na equação dinâmica com a lei de controle incluída, resultando em:

$$\mathbf{M}(\dot{\vec{\nu}} - \vec{\mathbf{a}}_v) = 0 \quad (3.6)$$

Como deseja-se efetuar o posicionamento do veículo no referencial inercial, a aceleração comandada expressa no referencial do corpo ($\vec{\mathbf{a}}_v$) não é inicialmente conhecida. Primeiramente é necessário definir uma trajetória de referência no referencial inercial, em função da posição e orientação inicial e da posição e orientação final desejadas do veículo. A aceleração comandada no referencial inercial é determinada de forma a acompanhar a aceleração de referência. Então, a aceleração comandada é transformada do referencial inercial para o referencial do corpo.

Derivando-se a equação 3.5 em relação ao tempo e isolando a aceleração $\dot{\vec{\nu}}$ no lado esquerdo da igualdade obtém-se (TAVARES, 2003):

$$\dot{\vec{\nu}} = \mathbf{J}^{-1}(\vec{\eta})[\ddot{\vec{\eta}} - \dot{\mathbf{J}}(\vec{\eta})\vec{\nu}] \quad (3.7)$$

Substituindo da equação 3.7 na 3.6, resulta a seguinte equação:

$$\mathbf{M}\mathbf{J}^{-1}(\vec{\eta})[\ddot{\vec{\eta}} - (\dot{\mathbf{J}}(\vec{\eta})\vec{\nu} + \mathbf{J}(\vec{\eta})\vec{\mathbf{a}}_v)] = 0 \quad (3.8)$$

Da equação 3.8, o termo $(\ddot{\vec{\eta}})$ que é o primeiro entre colchetes, expressa a aceleração do veículo, já o segundo termo que está entre parenteses expressa a aceleração comandada, expressa no referencial inercial, que é dado por:

$$\mathbf{a}_\eta = \dot{\mathbf{J}}(\vec{\eta})\vec{\nu} + \mathbf{J}(\vec{\eta})\vec{\mathbf{a}}_v \quad (3.9)$$

A pré-multiplicação de ambos os lados da equação 3.8 por $\mathbf{J}^{-T}(\vec{\eta})$ e a utilização de 3.9, permitem a obtenção da equação da dinâmica com a lei de controle expressa no referencial inercial, sob a forma (TAVARES, 2003):

$$\mathbf{M}_\eta(\ddot{\vec{\eta}} - \vec{\mathbf{a}}_v) = 0 \quad (3.10)$$

onde a equação 3.10 é sistema linear e desacoplado, resultante da realimentação em que $\mathbf{M}_\eta = \mathbf{J}^{-T}(\vec{\eta})\mathbf{M}\mathbf{J}^{-1}(\vec{\eta})$ é a matriz de inércia expressa no referencial inercial (FOSSSEN, 1994).

3.3 Trajetória de Referência

A trajetória de referência pode ser determinada a partir da solução de uma equação diferencial de segunda ordem, expressa por:

$$\ddot{\vec{\eta}}_d + 2\xi\omega_n\dot{\vec{\eta}}_d + \omega_n^2\vec{\eta}_d = \omega_n^2\vec{r}_\eta \quad (3.11)$$

onde $\vec{\eta}_d$ é o vetor com posição e orientação desejada a cada instante de tempo, \vec{r}_η é o vetor com posição e orientação final desejada, o coeficiente de amortecimento é representado por ξ e a frequência natural não-amortecida é representada por ω_n (TAVARES, 2003).

Definir esses parâmetros é uma tarefa importante, uma vez que pequenos valores de amortecimento produzem grande oscilação em torno da posição final desejada e valores de frequência não-amortecida muito altos produzem pequeno tempo de subida (FRANKLIN et al., 1994). O tempo de subida pequeno pode exigir correntes demasiadamente elevadas nos atuadores. Portanto, para que não ocorra um sobre-sinal significativo, TAVARES (2003) define um coeficiente de amortecimento em torno de 0,90. O tempo de subida t_r pode ser tomado como uma função linear do erro inicial de posição, por meio da seguinte expressão:

$$t_r = K\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \quad (3.12)$$

onde K é uma constante de proporcionalidade e $\Delta x, \Delta y, \Delta z$ são diferenças entre a posição inicial e a posição final desejada. O valor de K é determinado através de diversas simulações, de acordo com as correntes nos atuadores, procurando-se identificar um valor que atribuisse correntes aceitáveis para os atuadores (ausência de saturação). A primeira aproximação para K é definida com base na velocidade de cruzeiro do veículo.

Devido à necessidade de deslocamentos laterais e angulares do veículo, o valor final de K deve ser maior do que o da aproximação inicial.

Definido o tempo de subida t_r e o fator de amortecimento ξ , encontra-se a frequência natural não amortecida ω_n (OGATA; SEVERO, 1998).

$$\omega_n = \frac{\pi - \tan^{-1} \frac{\sqrt{1-\xi^2}}{\xi}}{t_r\sqrt{1-\xi^2}} \quad (3.13)$$

Com base na solução analítica da equação 3.11 descrita por OGATA; SEVERO (1998), tem-se que $\vec{\eta}_0$ é o vetor com a posição e orientação inicial do veículo e que $\omega_d = \omega_n\sqrt{1-\xi^2}$ é a frequência natural amortecida.

As seguintes equações expressam o vetor velocidade desejada $\dot{\vec{\eta}}_d$ e o vetor aceleração desejada $\ddot{\vec{\eta}}_d$:

$$\dot{\vec{\eta}}_d = \left[\xi \omega_n e^{-\xi \omega_n t} \left(\cos \omega_d t + \frac{\xi}{\sqrt{1-\xi^2}} \sin \omega_d t \right) + e^{-\xi \omega_n t} \left(\omega_d \sin \omega_d t - \frac{\xi \omega_d}{\sqrt{1-\xi^2}} \cos \omega_d t \right) \right] (\mathbf{r}_\eta - \boldsymbol{\eta}_0) \quad (3.14)$$

$$\ddot{\vec{\eta}}_d = \omega_n^2 \vec{\mathbf{r}}_\eta - 2\xi \omega_n \dot{\vec{\eta}}_d - \omega_n^2 \vec{\eta}_d \quad (3.15)$$

As equações 3.14 e 3.15 fornecem os vetores posição, velocidade e aceleração de referência, respectivamente, que possuem coordenadas relativas aos movimentos linear e angular do veículo (TAVARES, 2003).

3.4 Seguimento da trajetória: controle PID

A subseção descreve o desenvolvimento de um controlador do tipo proporcional, integral e derivativo (PID) para o seguimento da trajetória de referência. O controle PID é uma técnica de controle de processos que une as ações derivativa, integral e proporcional, fazendo com que o sinal de erro seja minimizado pela ação proporcional, zerado pela ação integral e obtido com uma velocidade antecipativa pela ação derivativa.

O erro $\tilde{\vec{\eta}}$ no seguimento de trajetória de referencia $\vec{\eta}_d$ é definido por:

$$\tilde{\vec{\eta}} = \vec{\eta} - \vec{\eta}_d \quad (3.16)$$

Com a utilização da lei de controle descrita pela equação 3.10, deseja-se que a aceleração do veículo no referencial inercial seja igual a aceleração comandada no referencial inercial $\ddot{\vec{\eta}} = \ddot{\vec{\eta}}_d$. Portanto, da equação 3.16 pode-se concluir que:

$$\ddot{\vec{\mathbf{a}}}_\eta = \ddot{\vec{\eta}}_d + \ddot{\tilde{\vec{\eta}}} \quad (3.17)$$

A dinâmica do erro pode ser controlada por uma ação proporcional, integral e derivativa:

$$\ddot{\tilde{\vec{\eta}}} + \mathbf{K}_D \dot{\tilde{\vec{\eta}}} + \mathbf{K}_P \tilde{\vec{\eta}} + \mathbf{K}_I \int \tilde{\vec{\eta}} dt = 0 \quad (3.18)$$

onde \mathbf{K}_P , \mathbf{K}_I e \mathbf{K}_D são matrizes diagonais com os ganhos, respectivamente, proporcionais, integrais e derivativos (PID).

Das equações 3.17 e 3.18 obtém-se a aceleração comandada no referencial inercial, descrita por:

$$\ddot{\vec{\mathbf{a}}}_\eta = \ddot{\vec{\eta}}_d - \mathbf{K}_D \dot{\tilde{\vec{\eta}}} - \mathbf{K}_P \tilde{\vec{\eta}} - \mathbf{K}_I \int \tilde{\vec{\eta}} dt \quad (3.19)$$

O modelo dinâmico do veículo está expresso no referencial do corpo, já a aceleração comanda

está expressa no referencial inercial \vec{a}_η , portanto, a aceleração comandada deve ser transformada também para o referencial do corpo. Essa transformação de referenciais é feita isolando-se a aceleração comandada \vec{a}_v na equação 3.7, resultando em:

$$\vec{a}_v = \mathbf{J}^{-1}(\vec{\eta})[\vec{a}_\eta - \dot{\mathbf{J}}(\vec{\eta})\vec{v}] \quad (3.20)$$

A Figura 3.1 resume, de forma esquemática, a sequência utilizada na estrutura de controle.

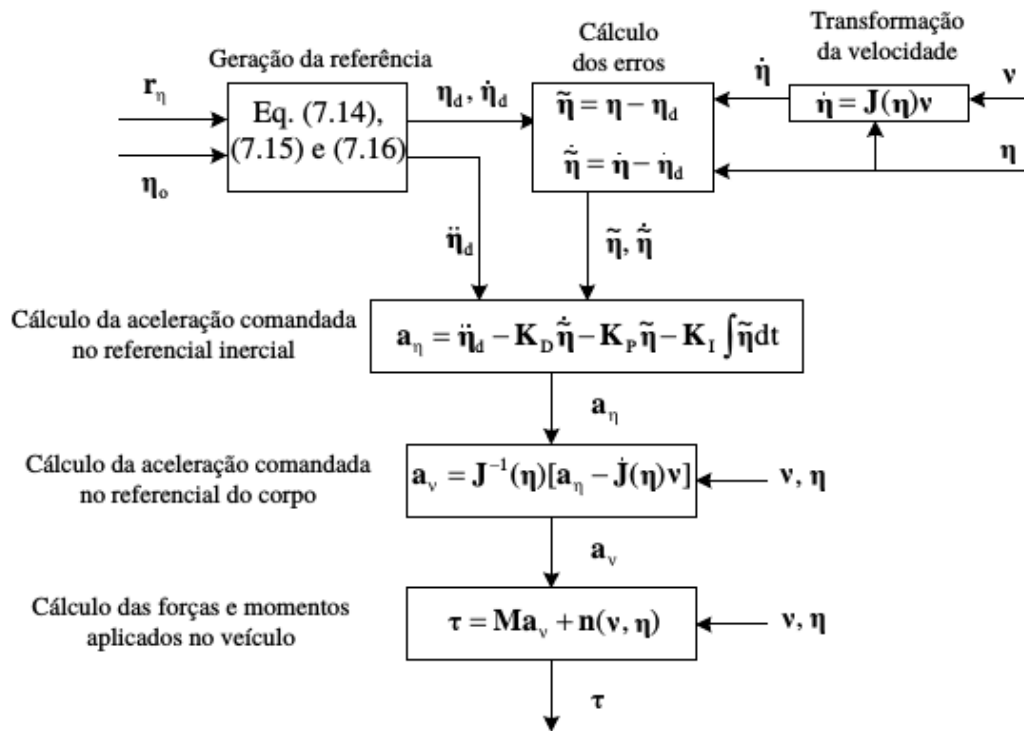


Figura 3.1: Sequência de ação do controlador (TAVARES, 2003)

O erro de posição e a sua derivada em relação ao tempo são facilmente determinados, uma vez que todo o estado do sistema é suposto conhecido. A integral do erro é determinada pelo processo proposto por GOMES; BIER (1998), o qual utiliza uma "janela de esquecimento". Trata-se da consideração de um número fixo de valores de erros para comporem o somatório que substitui a integral na implementação prática da lei de controle, na forma primeiro que entra corresponde ao último que sai.

3.5 Controlador PID, com linearização por realimentação, implementado para o NEROV

O veículo NEROV é descrito na seção 2.4. As velocidades dos propulsores do NEROV são controladas diretamente pelas tensões aplicadas aos mesmos (TAVARES, 2003). Já as velocidades dos propulsores, determinam as forças e momentos que controlam o veículo.

Do veículo, o vetor de controle \vec{u} é descrito pela equação 2.54 e a matriz de propulsão \mathbf{B} pela equação 2.55 TAVARES (2003), de onde x_{pi} , y_{pi} e z_{pi} representam as coordenadas do centro de

empuxo do *i-ésimo* propulsor.

Como o NEROV possui seis propulsores, a sua matriz de propulsão \mathbf{B} tem dimensão 6×6 . O que permite que o vetor de controle $\vec{\mathbf{u}}$ seja definido a partir da inversão da matriz de propulsão \mathbf{B} . Portanto, obtém-se a seguinte expressão:

$$\vec{\mathbf{u}} = \mathbf{B}^{-1} \vec{\mathbf{r}} \quad (3.21)$$

Conhecendo as componentes do vetor de controle, pode-se obter a velocidade do *i-ésimo* hélice, dado pela seguinte equação:

$$n_i = \text{sign}(\vec{\mathbf{u}}_i) \sqrt{|\vec{\mathbf{u}}_i|} \quad (i = 1, 2, 3, 4, 5, 6) \quad (3.22)$$

Considerando o propulsor como um motor de corrente contínua, e assim, desprezando-se o termo (queda de tensão na resistência da armadura), pode-se obter a tensão que deve ser aplicada ao *i-ésimo* motor para que ele desenvolva a velocidade desejada, definido pela equação:

$$V_{ai} = 2\pi K_m n_i \quad (3.23)$$

onde:

- V_{ai} é a tensão aplicada;
- n_i é a velocidade desejada;
- K_m é a constante eletromagnética do motor.

Os demais parâmetros do modelo do NEROV e resultados de simulação, estão descritos por (TAVARES, 2003).

4 ESTRUTURAS FLEXÍVEIS DO TIPO CABO

Neste capítulo será apresentada a formulação de Euler-Lagrange, utilizada na modelagem matemática da dinâmica de estruturas flexíveis do tipo cabo, considerando-o fixo em sua extremidade superior e livre na outra, na qual é colocada uma carga de massa m_c .

São consideradas duas situações distintas: o cabo fixo a uma plataforma móvel em uma das extremidades e livre na outra (carga terminal livre); o cabo fixo à plataforma móvel em uma extremidade e livre na outra com uma carga terminal na qual são aplicados forças e torques. Considera-se a plataforma móvel (que pode ser um navio ou uma embarcação) com possibilidade de movimento apenas na vertical, simulando assim as oscilações que ocorrem na superfície do mar.

O modelo dinâmico do cabo necessita ser determinado de forma precisa, para que o sistema de controle seja confiável, porém a sua determinação é uma tarefa complexa, dado as não-linearidades presentes no sistema, a flexibilidade estrutural do cabo no ambiente tridimensional e a influência de interações do cabo com o meio onde está submerso.

As equações da dinâmica serão obtidas a partir da abordagem discreta que divide o cabo (estrutura contínua) em partes rígidas (elos) conectadas por elementos flexíveis chamados de articulações elásticas fictícias, originalmente desenvolvida por PEREIRA (2010), inspirado no modelo proposto por GOMES; ROSA; ALBERTINI (2006).

4.1 Modelagem do cabo

A modelagem do cabo se baseia no método de dividir o cabo em pequenos elementos (elos), conectados por articulações fictícias flexíveis que permitem três movimentos livres e independentes, denominados de azimute, elevação e torção (ZANELA, 2013). Estes movimentos são sempre relativos ao elo anterior da cadeia articulada (PEREIRA, 2010).

Cada articulação fictícia tem natureza elástica e, portanto, três constantes elásticas com seus respectivos amortecimentos que constituem parâmetros que definem a natureza física da articulação.

Considera-se um cabo como estrutura flexível, fixo em sua extremidade superior, articulado a uma base podendo ser móvel e em sua extremidade inferior uma massa m_c , (Figura 4.1) como carga terminal. Estas estruturas flexíveis possuem diversas aplicações em sistemas subaquáticos, como, por exemplo, em sistema de produção de extrato de petróleo (Figura 4.2).

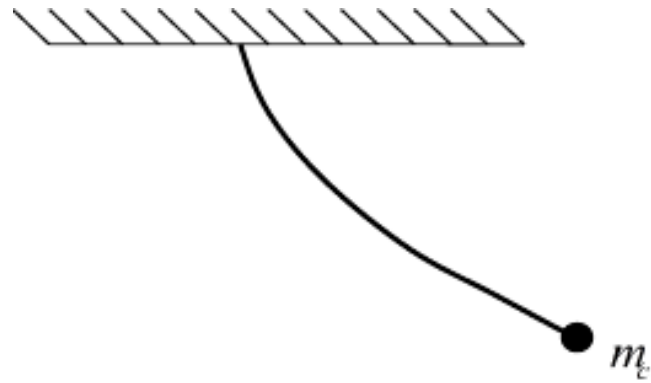


Figura 4.1: Representação cabo com carga terminal (ROCHA, 2013).

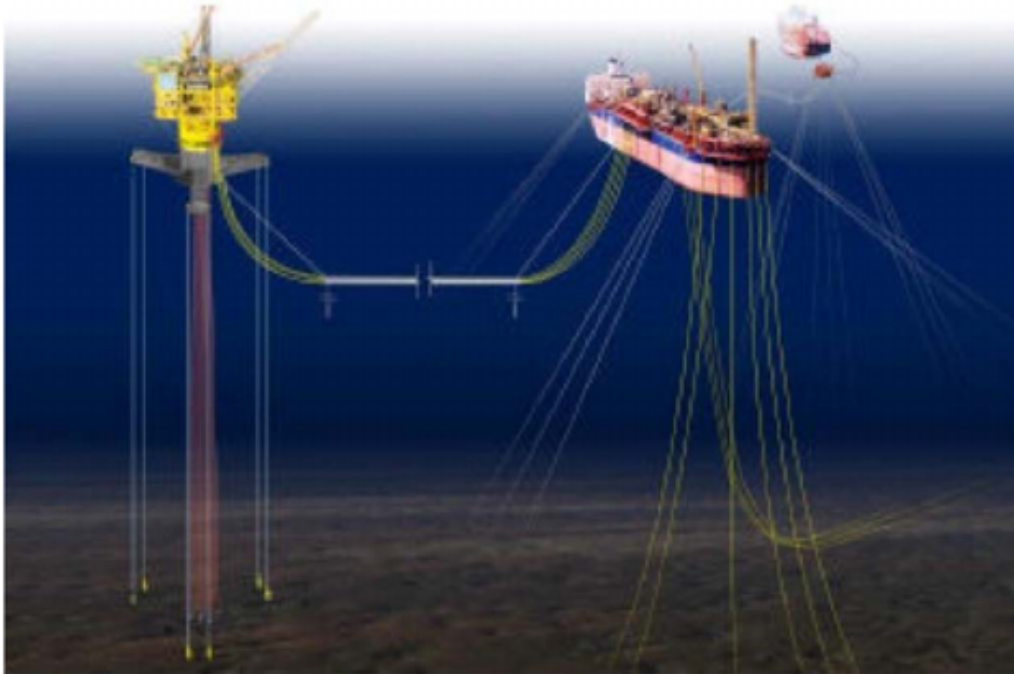


Figura 4.2: Sistema subaquático (BANERJEE; DO, 1995).

Para um cabo fixo na sua extremidade superior e livre na outra, é necessário que a orientação e a posição da extremidade livre deste cabo sejam conhecidas. A orientação da extremidade livre do cabo é definida por intermédio de rotações puras e a posição por translações.

Considera-se o cabo representado por pequenos elos de comprimentos $l_1, l_2, l_3, \dots, l_n$, conectados através de articulações fictícias, como mostra na Figura 4.3.

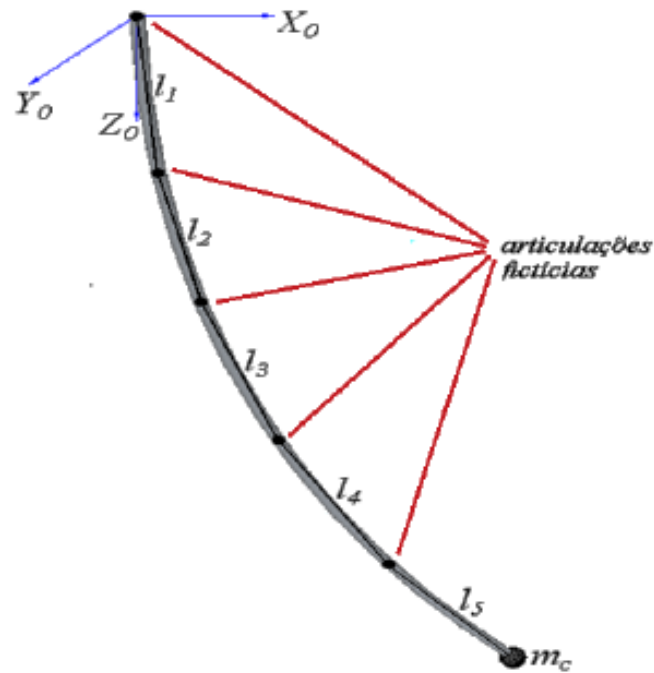


Figura 4.3: Estrutura flexível e sua representação discreta (ZANELA, 2013).

Para determinar-se a posição de cada elo e da carga terminal, é associado um sistema de referencia na extremidade superior do cabo (fixa), denominado $X_0Y_0Z_0$, e um sistema de eixos coordenados para cada articulação $X_iY_iZ_i$, com $i \in \mathbb{N}$ e $1 \leq i \leq n - 1$. Em cada articulação são considerados os ângulos de azimute, elevação e torção do cabo. Assim, na i -ésima articulação o ângulo de azimute θ_{ia} é formado pela projeção do elo no plano horizontal X_iY_i com sentido positivo do eixo X_i , o ângulo de elevação θ_{ie} corresponde ao ângulo entre a reta suporte do elo e o eixo vertical Z_i e o ângulo de torção θ_{iT} e corresponde a rotação em torno do próprio eixo do elo, conforme a Figura 4.4.

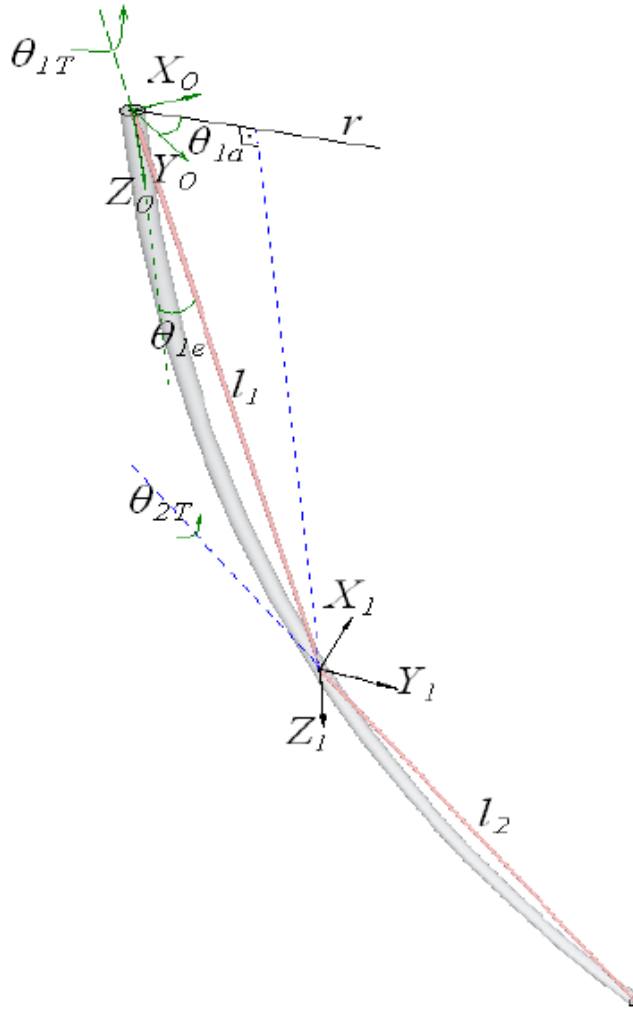


Figura 4.4: Estrutura flexível e sua representação discreta (ZANELA, 2013).

Os elos têm massas concentradas nos centros de massa, com coordenadas (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) , ... , (x_n, y_n, z_n) que possuem massas, respectivamente, $m_1, m_2, m_3, \dots, m_n$ enquanto (x_c, y_c, z_c) são as coordenadas do centro de massa da carga de massa m_c . Em cada articulação são consideradas três constantes elásticas, ou seja, na i -ésima articulação são consideradas as constantes elásticas k_{ia}, k_{ie}, k_{iT} , devido aos ângulos de azimute, elevação e torção respectivamente (PEREIRA, 2010; ZANELA, 2013; OLIVEIRA, 2015).

Para determinar o Lagrangeano do sistema, a Energia Cinética é definida por:

$$E_C = E_{C_R} + E_{C_T} \quad (4.1)$$

onde E_{C_R} é a energia cinética proveniente da rotação do cabo e E_{C_T} é devido ao movimento de translação do cabo.

A energia cinética devido a rotação do cabo é definida por:

$$E_{C_R} = \frac{1}{2} \left[\sum_{i=1}^n \left(I_{R,i} \sum_{j=1}^i \dot{\theta}_{a,j}^2 + I_{T,i} \dot{\theta}_{T,i}^2 \right) + I_{T,c} \dot{\theta}_{T,n}^2 \right] \quad (4.2)$$

onde:

- $I_{R_{ie}} \Rightarrow$ são os momentos de inércia relativos ao movimento de elevação;
- $\theta_{ie} \Rightarrow$ são os ângulos em elevação;
- $I_{R_{ia}} \Rightarrow$ são os momentos de inércia relativos ao movimento de azimute;
- $\theta_{ia} \Rightarrow$ são os ângulos em azimute;
- $I_{T_i}, I_{T_c} \Rightarrow$ são os momentos de inércia relativos ao movimento de torção;
- $\theta_{iT} \Rightarrow$ são os ângulos de torção;

Salienta-se que a energia cinética de translação é mais significativa que a de rotação e, portanto, é dada ênfase à energia cinética de translação neste formalismo de modelagem (PEIREIRA, 2010).

A energia cinética devido ao movimento de translação do cabo é definida por:

$$E_{CT} = \frac{1}{2} \left[\sum_{i=1}^n (m_i (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2)) + m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2) \right] \quad (4.3)$$

onde $\frac{1}{2}m_i (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2)$ e $\frac{1}{2}m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2)$ são energias cinéticas devidas aos movimentos das massas m_i dos elos e m_c da carga terminal, respectivamente. \dot{x}_i , \dot{y}_i e \dot{z}_i são os componentes da velocidade linear do elo de acordo com o sistema de coordenadas de referência .

A energia potencial devido ao peso do cabo é definida por:

$$E_P = \frac{1}{2} \left[\sum_{i=1}^n k_{e,i} (\theta_{e,i} - \theta_{e,i-1})^2 + \sum_{i=1}^n k_{a,i} (\theta_{a,i} - \theta_{a,i-1})^2 + \sum_{i=1}^n k_{T,i} (\theta_{T,i} - \theta_{T,i-1})^2 \right] + \sum_{i=1}^n m_i g h_i \quad (4.4)$$

onde g é a aceleração da gravidade, h_i é a altura do centro de massa i^{th} e $m_i g h_i$ é a energia potencial gravitacional. Esta altura pode ser genericamente expressa como:

$$h_i = \sum_{j=1}^{i-1} a_j + \frac{l_i}{2} (1 - \cos \theta_{e,i}) \quad (4.5)$$

onde a_i é a projeção vertical do comprimento l_i do elo i^{th} . A altura da carga do terminal h_c , pode ser genericamente expressa como:

$$h_c = \sum_{i=1}^{n-1} a_i + \frac{l_n}{2} (1 - \cos \theta_{e,n}) \quad (4.6)$$

Finalmente, para o elo entre o cabo e a base, $\theta_{e,0} = \theta_{a,0} = \theta_{T,0} = 0$.

O Lagrangeano deste sistema, definido como $L = Ec - Ep$, pode ser expresso como função de coordenadas angulares somente após algum rearranjo do sistema de equações como:

$$L = L_1 + L_2 + L_3 + L_4 \quad (4.7)$$

Os termos L_1 , L_2 , L_3 e L_4 são definidos como:

$$\begin{aligned} L_1 = & \sum_{i=1}^n \left\{ \frac{1}{2} I_{Re,i} \dot{\theta}_{e,i}^2 + \frac{1}{2} I_{Ra,i} \dot{\theta}_{a,i}^2 + \frac{1}{2} I_{RT,i} \dot{\theta}_{T,i}^2 - \right. \\ & \frac{1}{2} K_{e,i} (\theta_{e,i} - \theta_{e,(i-1)})^2 - \frac{1}{2} K_{a,i} (\theta_{a,i} - \theta_{a,(i-1)})^2 \\ & \left. - \frac{1}{2} K_{T,i} (\theta_{T,i} - \theta_{T,(i-1)})^2 - \right. \\ & \left. m_i g \left[\sum_{j=1}^{i-1} a_j + \frac{l_j}{2} (1 - \cos \theta_{e,j}) \right] \right\} + \\ & \frac{1}{2} I_{Ce} \dot{\theta}_{e,n}^2 + \frac{1}{2} I_{Ca} \dot{\theta}_{a,n}^2 + \frac{1}{2} I_{CT} \dot{\theta}_{T,n}^2 - \\ & m_c g \sum_{i=1}^n l_i (1 - \cos \theta_{e,i}) \end{aligned} \quad (4.8)$$

$$\begin{aligned} L_2 = & \sum_{i=1}^n \frac{m_i}{2} \left\{ \frac{l_i}{2} \left[\sin \theta_{e,i} \cos \left(\sum_{j=1}^i \theta_{a,j} \right) \sum_{j=1}^i \dot{\theta}_{a,j} + \right. \right. \\ & \left. \left. \cos \theta_{e,i} \sin \left(\sum_{j=1}^i \theta_{a,j} \right) \dot{\theta}_{e,i} \right] + \right. \\ & \left. \sum_{j=1}^{i-1} l_j \left[\sin \theta_{e,j} \cos \left(\sum_{b=1}^j \theta_{a,b} \right) \sum_{b=1}^j \dot{\theta}_{a,b} + \right. \right. \\ & \left. \left. \cos \theta_{e,j} \sin \left(\sum_{b=1}^j \theta_{a,b} \right) \dot{\theta}_{e,j} \right] \right\}^2 \end{aligned} \quad (4.9)$$

$$\begin{aligned}
L3 = & \sum_{i=1}^n \frac{m_i}{2} \left\{ \frac{l_i}{2} \left[-\sin \theta_{e,i} \sin \left(\sum_{j=1}^i \theta_{a,j} \right) \sum_{j=1}^i \dot{\theta}_{a,j} + \right. \right. \\
& \left. \left. \cos \theta_{e,i} \cos \left(\sum_{j=1}^i \theta_{a,j} \right) \dot{\theta}_{e,i} \right] + \right. \\
& \left. \sum_{j=1}^{i-1} l_j \left[-\sin \theta_{e,j} \sin \left(\sum_{b=1}^j \theta_{a,b} \right) \sum_{b=1}^j \dot{\theta}_{a,b} + \right. \right. \\
& \left. \left. \cos \theta_{e,j} \cos \left(\sum_{b=1}^j \theta_{a,b} \right) \dot{\theta}_{e,j} \right] \right\}^2 + \\
& \sum_{i=1}^n \frac{m_i}{2} \left[-\frac{l_i}{2} \sin \theta_{e,i} \dot{\theta}_{e,i} - \sum_{j=1}^{i-1} \sin \theta_{e,j} \dot{\theta}_{e,j} \right]^2
\end{aligned} \tag{4.10}$$

$$\begin{aligned}
L4 = & \frac{1}{2} m_c \left\{ \sum_{i=1}^n l_b \left[\sin \theta_{e,i} \cos \left(\sum_{j=1}^i \theta_{a,j} \right) \sum_{j=1}^i \dot{\theta}_{a,j} + \right. \right. \\
& \left. \left. \cos \theta_{e,i} \sin \left(\sum_{j=1}^i \theta_{a,j} \right) \dot{\theta}_{e,i} \right] \right\}^2 + \\
& \frac{1}{2} m_c \left\{ \sum_{i=1}^n l_b \left[-\sin \theta_{e,i} \sin \left(\sum_{j=1}^i \theta_{a,j} \right) \sum_{j=1}^i \dot{\theta}_{a,j} + \right. \right. \\
& \left. \left. \cos \theta_{e,i} \cos \left(\sum_{j=1}^i \theta_{a,j} \right) \dot{\theta}_{e,i} \right] \right\}^2 + \\
& \frac{1}{2} m_c \left[\sum_{i=1}^n l_b \sin \theta_{e,i} \dot{\theta}_{e,i} \right]^2
\end{aligned} \tag{4.11}$$

A partir do Lagrangeano anteriormente obtido, é possível calcular a equação de Euler-Lagrange para todo o sistema de $3n$ graus de liberdade, resultando no modelo dinâmico representado na equação 4.12:

$$\mathbf{I}(\vec{\theta}) \ddot{\vec{\theta}} + \mathbf{C} \dot{\vec{\theta}} + \mathbf{K} \vec{\theta} + \vec{f}(\vec{\theta}, \dot{\vec{\theta}}) + \vec{G}(\vec{\theta}, \dot{\vec{\theta}}) = \vec{\tau}_m \tag{4.12}$$

onde $\mathbf{I}(\vec{\theta})$ é a matriz de inércia, \mathbf{C} é a matriz de coeficientes de atrito, \mathbf{K} é a matriz de constantes elásticas, $\vec{f}(\vec{\theta}, \dot{\vec{\theta}})$ é o vetor de esforços Coriolis-centrífugos, $\vec{G}(\vec{\theta}, \dot{\vec{\theta}})$ é o vetor de esforços gravitacionais e $\vec{\tau}_m$ é o vetor de torques externos. Todas as matrizes tem dimensão $[3n \times 3n]$, enquanto todos os vetores possuem dimensão $[3n \times 1]$.

Os elementos das matrizes e vetores do modelo dinâmico do cabo são desenvolvidos por GOMES; ZANELA; PEREIRA (2016).

4.2 Cabo com Carga Terminal: Simplificação do acoplamento de um veículo

O caso de um veículo conectado por um cabo umbilical pode ser estendido a partir do modelo de um cabo com carga terminal, considerando que a carga é um corpo com forças/torques atuando sobre ele devido a seus propulsores. A Figura 4.5 ilustra esta situação, onde a força resultante atua no centro de massa do veículo, do tipo ROV, onde o sistema de coordenadas $X_C Y_C Z_C$ tem sua origem.

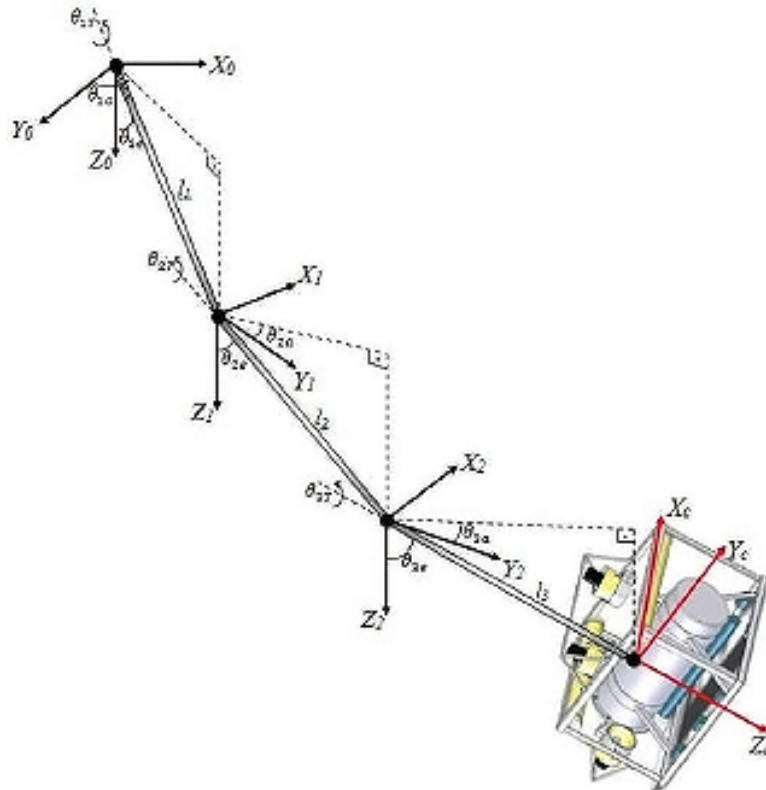


Figura 4.5: Conectando um ROV como carga terminal do cabo (ROCHA, 2013)

A força exercida no veículo, representada por \vec{F}_C , é gerada pelos propulsores para mover o veículo ao longo de uma trajetória de referência e é expresso de acordo com o sistema de referência no ROV (corpo), como demonstrado na Figura 4.6.

Esta força também produz um torque, expresso na equação 4.13. Nesta equação, $\vec{\tau}_C$ é expresso de acordo com o sistema de coordenadas $X_n Y_n Z_n$. \mathbf{r} é a distância entre a última junta virtual do cabo e o centro de massa do veículo.

$$\vec{\tau}_C = \vec{F}_C \times \vec{r} \quad (4.13)$$

A força/torque produzida no último elo do cabo é propagada para todos os outros elos. Além disso, as forças resultantes geradas pelo comportamento dinâmico do cabo afetam o ROV, de acordo com o sistema de coordenadas $X_n Y_n Z_n$, é expresso na equação 4.14.

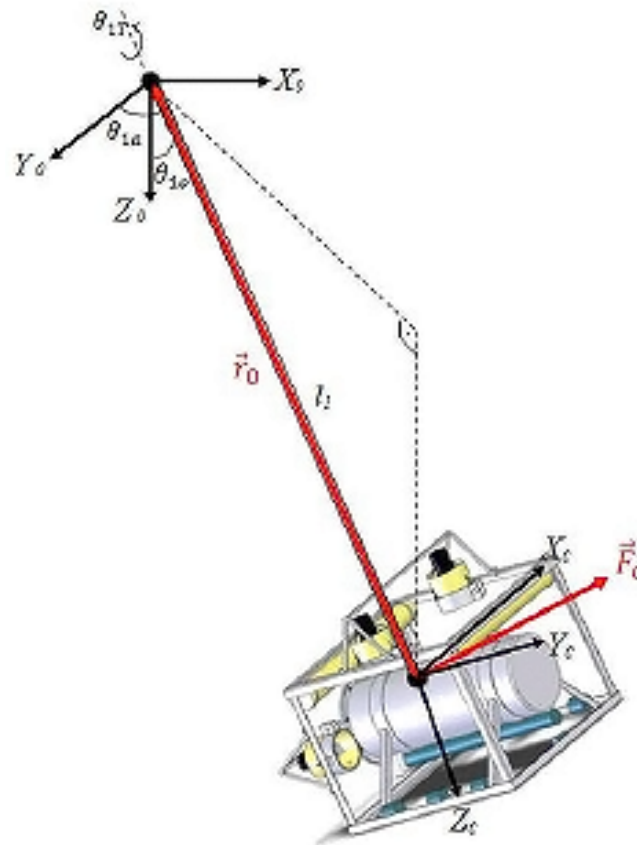


Figura 4.6: Força exercida pela carga terminal (ROV) (ROCHA, 2013)

$$\vec{\tau}_n = \vec{\tau}_C + \vec{\tau}_{C0} \quad (4.14)$$

A relação entre as forças no cabo e no veículo é estabelecida quando $\vec{\tau}_n$ é aplicado na dinâmica do ROV, equação 2.48, igualando a $\vec{\tau}_u$.

O acoplamento da dinâmica da carga terminal com a dinâmica do cabo, assim como os elementos das matrizes e vetores do modelo dinâmico das forças e torques na carga terminal, são desenvolvidos por OLIVEIRA (2015).

5 FUNDAMENTOS DA MODELAGEM E ANIMAÇÃO GRÁFICA

A computação gráfica é uma tecnologia cuja crescente aplicação nos diversos ramos de atividade motiva a sua evolução tanto em software quanto em hardware. Isso se reflete na área científica, no uso de ferramentas CAD (*Computer-Aided Design*, ou Projeto Assistido por Computador), por exemplo.

Outra aplicação importante da computação gráfica é em simulações, onde a modelagem gráfica complementa a modelagem física/numérica, permitindo visualizar resultados de forma estática ou dinâmica. Além de representações virtuais animadas dos ambientes e objetos simulados, é possível apresentar dados em diferentes formas de visualização, o que acrescenta um importante recurso de análise quantitativa e qualitativa às simulações.

5.1 Modelagem

No caso das simulações, a modelagem gráfica consiste em definir a representação das entidades que fazem parte do sistema a ser simulado, bem como a interação entre estas, o ambiente simulado e eventualmente com o usuário do software. De acordo com a natureza do sistema, das entidades, e do grau de realismo visual a ser obtido, diferentes abordagens podem ser utilizadas. Em situações onde as entidades consistem em corpos rígidos, por exemplo, o modelamento por primitivas é uma opção adequada e relativamente rápida de processar. No caso de entidades fluidas, como a água, o modelamento pode empregar superfícies paramétricas variantes no tempo, além de ser necessário considerar efeitos do próprio fluido, como translucidez e interação com a luz. Assim, é comum se trabalhar com mais de uma abordagem de modelagem (MANSSOUR; COHEN, 2006).

5.1.1 Modelagem Por Primitivas Geométricas

Na modelagem por primitivas, uma entidade é definida por objetos geométricos cujas formas são matematicamente definíveis, como cilindros, paralelepípedos, esferas, sólidos de revolução e poliedros. Esses objetos podem ter suas posturas (posições e orientações) determinadas em relação a outros objetos que compõem a entidade ou em relação a um sistema de referência da entidade, que também serve para definir a posição e orientação desta em relação ao ambiente da simulação.

Cada sólido tem parâmetros que definem suas dimensões, como a altura e o diâmetro de um cilindro, por exemplo. Estes são definidos em função de um referencial do próprio objeto, como mostrado na Figura 5.1.

Com base nos parâmetros, gera-se uma malha de pontos conectados (*mesh*), que é usada para desenhar o objeto no cenário. A densidade de pontos é um fator que afeta a qualidade do resultado final. Esta densidade é função da resolução do dispositivo gerador da representação

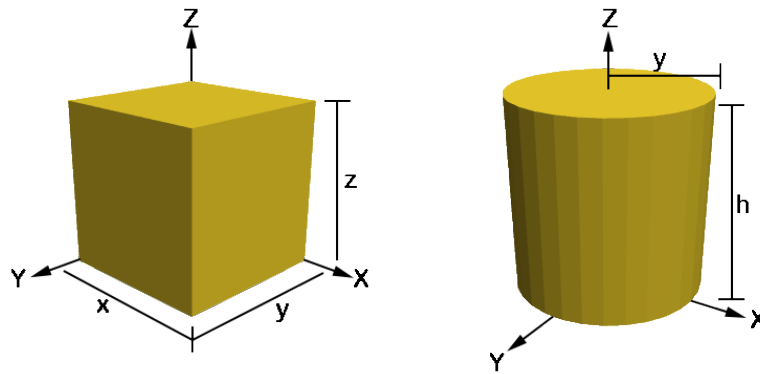


Figura 5.1: Alguns sólidos primitivos e seus parâmetros dimensionais

gráfica, da capacidade de processamento do processador e do quão realista a cena deve parecer (JOHN F. HUGHES ANDRIES VAN DAM, 2014). Um exemplo de como a malha de pontos afeta o resultado final pode ser observado na Figura 5.2, onde o mesmo cilindro foi desenhado com fatores de densidade diferentes.



Figura 5.2: Cilindros idênticos desenhados com densidades de malhas distintas, no software OpenSCAD (KINTEL; WOLF, 2017)

As dimensões das primitivas podem ser afetadas por transformações geométricas como escalas e distorções (ou cisalhamento, do inglês *shearing*), aplicadas a cada ponto da malha da primitiva geométrica. Essas transformações ainda podem ser empregadas para definir a postura do objeto, usando translações e rotações em relação a um referencial. Uma representação muito empregada para tais transformações é a matriz de transformação homogênea. É possível compor as transformações por multiplicações das matrizes homogêneas, o que permite aplicar várias transformações em uma única operação sobre todos os pontos da malha da primitiva (HILL; KELLEY, 2007). A Figura 5.3 exemplifica o efeito de algumas transformações sobre um paralelepípedo.

A escolha do referencial em relação ao qual as transformações geométricas são definidas é importante, pois o comportamento dinâmico da entidade pode ser representado diretamente por estas transformações. No caso de entidades rígidas, o referencial da entidade pode ser o escolhido, de forma que as matrizes serão função da posição e orientação deste referencial em relação ao ambiente simulado. No caso da entidade ter partes móveis entre si, pode-se representar estas em relação a outras partes da entidade, o que acontece com robôs manipuladores,

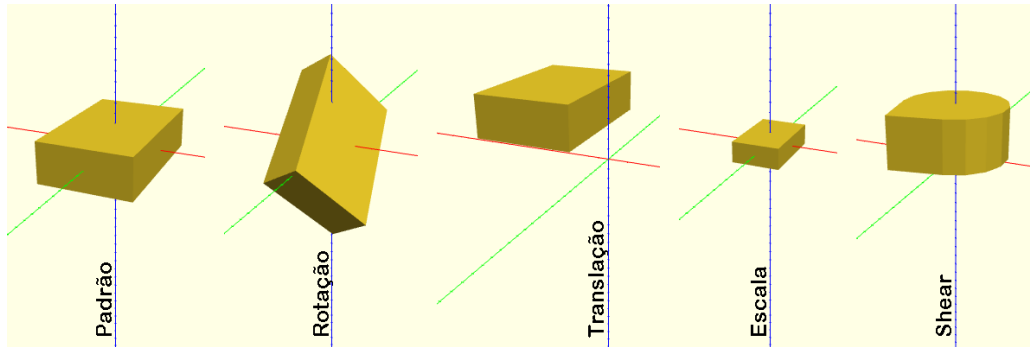


Figura 5.3: Efeito das transformações geométricas sobre um paralelepípedo

por exemplo, onde um elo do robô tem sua postura determinada em função do elo ao qual ele está vinculado.

Um exemplo de representação em relação ao referencial da entidade é mostrada na Figura 5.4, que apresenta um ROV geometricamente formado pela combinação de primitivas como cilindros, paralelepípedos, esferas e poliedros.

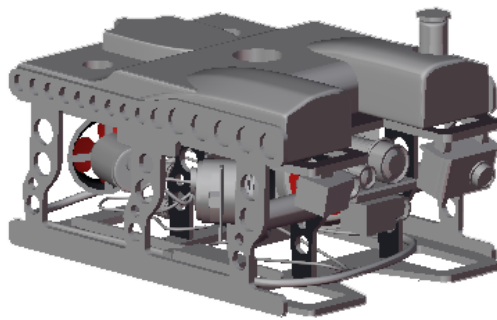


Figura 5.4: ROV modelado por primitivas e transformações geométricas (SORIMAJI, 2015)

5.1.2 Modelagem Por Superfícies Paramétricas

Curvas e superfícies paramétricas são bastante empregadas no desenvolvimento de formas que devam ter fluidez, como no projeto de estruturas com propriedades aerodinâmicas ou hidrodinâmicas, e onde a estética se alia a essas características, como no caso da indústria automotiva. Em certos casos, as primitivas geométricas podem ser substituídas por volumes delimitados por superfícies paramétricas para aumentar o grau de realismo do modelo, como por exemplo em formas orgânicas, como braços e pernas de seres vivos. Além desses casos, a possibilidade de modificação dinâmica dessas superfícies é interessante para objetos flexíveis e que mudem a sua forma ao longo do movimento, como é o caso de tecidos ou pele.

Assim como para as primitivas geométricas, as técnicas de representação paramétrica de curvas e superfícies depende muito do dispositivo que irá gerar a imagem e da qualidade desejada. O tempo necessário para o processamento e desenho dos objetos é função da quantidade de parâmetros e da densidade da malha de pontos a ser gerada. Se for se pensar em animação, esses fatores são ainda mais críticos, uma vez que deve-se considerar que uma quantidade grande

de imagens deve ser gerada em curtos espaços de tempo, a não ser que esta seja gerada *offline*.

Uma tipo de representação bastante empregado na modelagem de superfícies é a de Bézier, que define uma superfície bicúbica através de um conjunto de 3 equações que interpolam os pontos de controle (16, no total) para gerar uma superfície contínua e suave, como a exemplificada na Figura 5.5 (CHANG, 2015). Apesar de haverem modelos matemáticos mais detalhados, como os NURBS (*Non Uniform Rational Basis Spline*), as superfícies de Bézier usualmente são mais rápidas em termos de processamento.

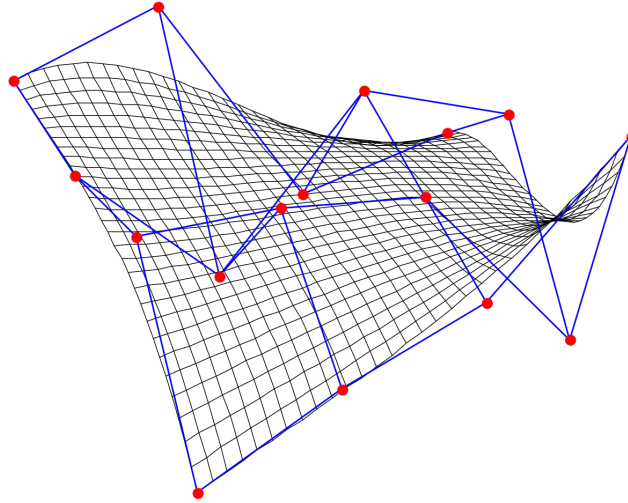


Figura 5.5: Exemplo de superfície paramétrica Bézier (CHANG, 2015)

O uso de transformações geométricas nas superfícies paramétricas tem efeito análogo ao de aplicação em primitivas geométricas, visto que ambos os tipos de modelagem acabam por gerar pontos que sofrerão as referidas transformações.

No caso da modelagem deste trabalho, as superfícies paramétricas são empregadas para representação da superfície da água e para a geração de efeitos de ondulação da imagem em função do cenário subaquático, quando isso é desejado.

5.2 Cenários Virtuais e Visualização

Para simulações gráficas de sistemas dinâmicos, é importante ter uma forma de representar e organizar as entidades que o compõem, de maneira que se possa gerar a visualização adequada e coerente das cenas. Estas podem ser definidas como a configuração das entidades do sistema em relação ao mundo em um instante da simulação, definida pelos resultados da simulação numérica para um determinado instante de tempo.

5.2.1 Mundo ou Universo Virtual

O ambiente do sistema simulado é comumente chamado de mundo ou universo, de acordo com a literatura e com o software de modelagem gráfica/animação utilizado. Pode-se afirmar que o mundo é definido por um conjunto de objetos, entre eles as entidades do sistema simulado.

Para definir a postura de cada objeto, o mundo virtual deve ter um sistema de coordenadas de referência, usualmente cartesiano (JOHN F. HUGHES ANDRIES VAN DAM, 2014; HILL; KELLEY, 2007). O sistema pode ser dextrógiro (baseado na regra da mão direita) ou levógiro (baseado na regra da mão esquerda), conforme a Figura 5.6.

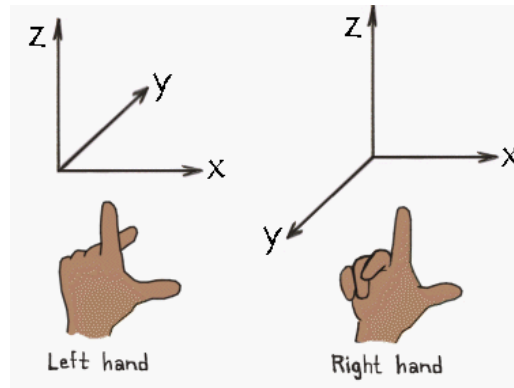


Figura 5.6: Sistemas de orientação dos eixos cartesianos, adaptado de (LEARN OPENGL ES, 2012)

Para cada objeto no mundo, deve-se ter uma representação deste e a sua postura. Esta pode ser definida por diferentes maneiras, como por um vetor de posição e orientação por ângulos de Euler, por exemplo. Porém, a mais usual é por matrizes de transformação homogênea, que também são usadas para modificar os objetos e são facilmente combinadas por operações de multiplicação de matrizes. Além disso, os hardwares de processamento gráfico atuais usualmente processam nativamente esse tipo de transformação (JOHN F. HUGHES ANDRIES VAN DAM, 2014).

No caso de objetos compostos, como no caso de definição por primitivas gráficas, as transformações homogêneas provêm uma forma hierárquica de composição de transformações, sistematizando a forma de posicionamento de cada componente do objeto em relação ao seu referencial e em relação ao mundo. Um exemplo disso é apresentado na Figura 5.7, que mostra um cenário virtual de uma floresta, onde cada árvore é formada por primitivas gráficas como cones e cilindros (MCKESSON, 2012).

5.2.2 Câmera Sintética

Um mundo virtual pode ter uma grande quantidade de objetos a serem processados a cada cena. Isso implica no uso dos resultados de um instante de simulação para definir a posição, orientação, e mesmo forma, dos objetos do mundo. Com isso, observa-se que o referido processo demanda muito processamento. O conceito de *câmera sintética* auxilia no processo de geração das cenas (ou *renderização*), por auxiliar na definição de que objetos devem ser efetivamente processados a cada instante (HILL; KELLEY, 2007; MCKESSON, 2012).

A câmera sintética consiste na determinação da postura em que um observador se encontra no mundo virtual. Além disso, define-se um volume de visualização, que delimita o que o observador é capaz de observar no mundo. Assim, uma mesma cena pode gerar diferentes

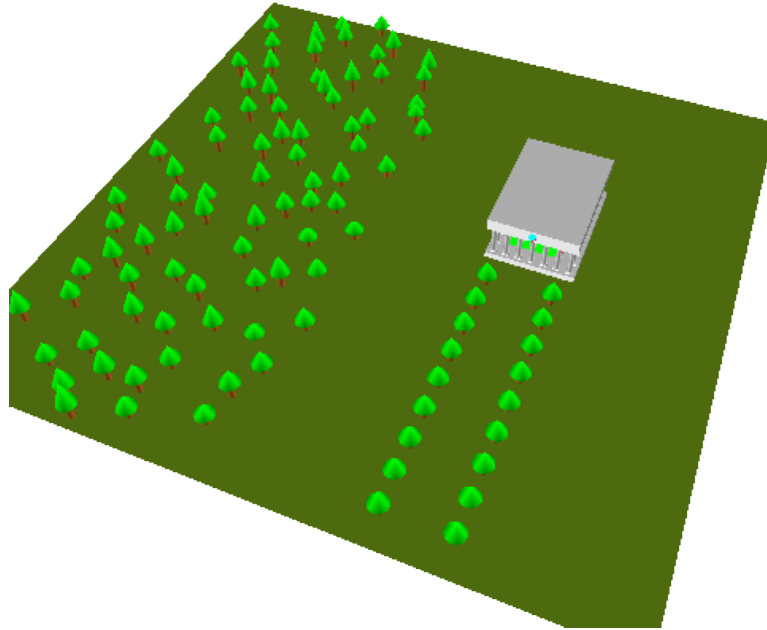


Figura 5.7: Cenário virtual de uma floresta (MCKESSON, 2012)

visualizações, de acordo com o observador. A Figura 5.8 ilustra duas possíveis posturas de um observador no mundo virtual, que irão gerar visualizações distintas da mesma cena.

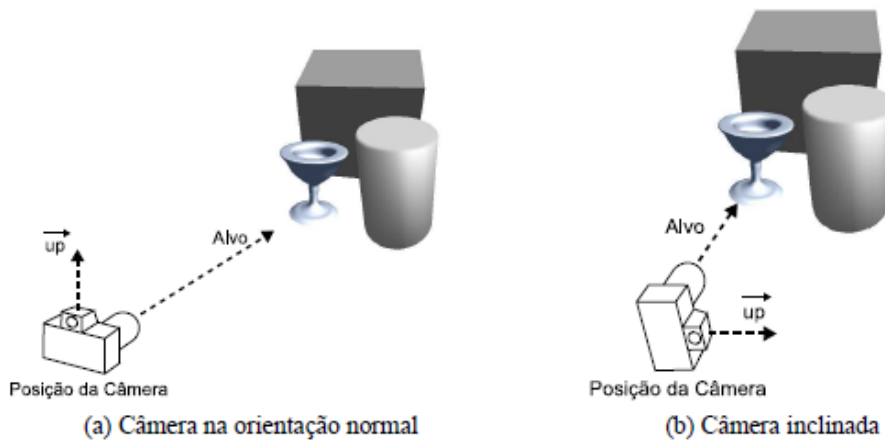


Figura 5.8: Definição de câmeras sintéticas com diferentes orientações (MANSSOUR; COHEN, 2006)

O volume de visualização do observador auxilia na definição de quais objetos presentes no mundo devem ser efetivamente renderizados para um determinado instante de uma cena, o que afeta significativamente a quantidade de processamento gráfico. Além disso, o volume de visualização é usado para realizar a projeção dos elementos do espaço tridimensional (do mundo virtual) para o espaço bidimensional (do dispositivo de visualização) (JOHN F. HUGHES ANDRIES VAN DAM, 2014), como demonstrado na Figura 5.9.

A projeção pode ser ortográfica ou perspectiva. A primeira é mais empregada em softwares para desenhos técnicos, como CADs, onde o volume é definido como um paralelepípedo. A projeção perspectiva é mais utilizada em sistemas de renderização de ambientes, devido ao fato

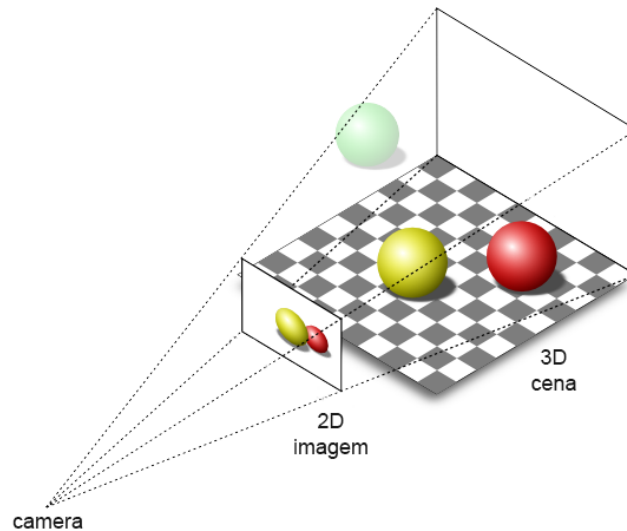


Figura 5.9: Projeção do volume de visualização para um *viewport* (ou vista). Adaptado de (JAN, 2016)

de ser mais próxima à que o olho humano realiza ao captar uma imagem e, portanto, parece mais natural ao observador (JOHN F. HUGHES ANDRIES VAN DAM, 2014). Nesta, define-se um tronco de pirâmide, o que faz com que objetos mais distantes pareçam menores, como na Figura 5.10 (MANSSOUR; COHEN, 2006).

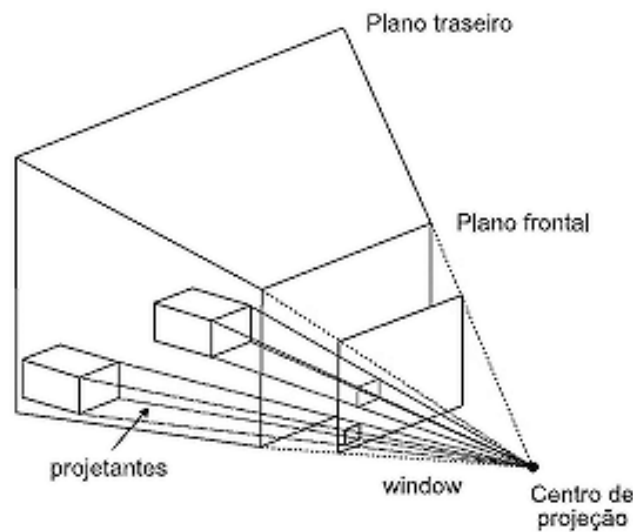


Figura 5.10: Tronco de pirâmide da projeção perspectiva (MANSSOUR; COHEN, 2006)

Diferentes efeitos podem ser obtidos graças ao conceito de câmera sintética. Um deles é o de estereoscopia, onde duas câmeras sintéticas são posicionadas como os olhos de um ser humano, gerando imagens planas levemente diferentes, que ao serem visualizadas com os instrumentos adequados (por exemplo, óculos especiais) geram um efeito tridimensional. Outro efeito interessante é o de simular o movimento do observador, fazendo com que a câmera mude de postura ao longo do tempo, Figura 5.11.

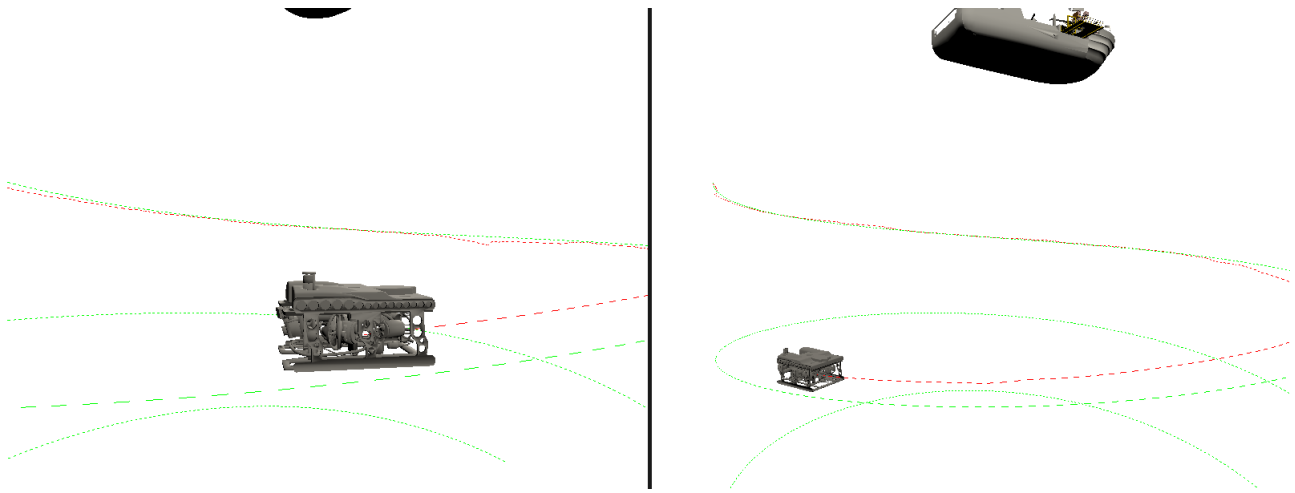


Figura 5.11: Movimento de câmera no USim

5.2.3 Fatores Que Influenciam o Grau de Realismo de Cenas

A definição da forma dos objetos e da postura do observador no mundo virtual são decisivos para a geração das visualizações das cenas. Porém, o realismo obtido em uma renderização é função não apenas do grau de geração das malhas das superfícies dos objetos, mas também das características dessas superfícies e da iluminação presente no mundo virtual.

Ao se renderizar um objeto de uma cena, seja definido por primitivas geométricas, seja por superfícies paramétricas, se obtém uma malha (*mesh*) definida por pontos que se conectam em arestas, que definem faces planares.

Uma face pode ter uma cor ou uma textura de preenchimento. No primeiro caso, cada ponto interno à face seria da cor definida para esta. No segundo caso, a textura consiste em um padrão de imagem que tenta emular os padrões de superfície dos materiais renderizados, e cada ponto da face teria uma cor definida pela cor do padrão da textura.

Utilizando apenas as propriedades dos objetos, é possível renderizar os objetos de uma cena através de *wireframes* (ou armação) ou por planos preenchidos. Um exemplo pode ser visto na Figura 5.12, onde um cubo pe renderizado com uma cor sólida ou pelo traçado de suas arestas apenas.

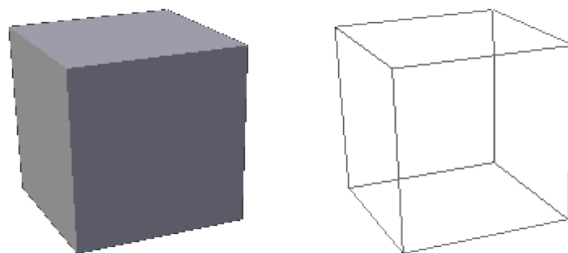


Figura 5.12: Cubo renderizado como sólido (esquerda) e *wireframe*(direita)

A renderização por *wireframes* costuma ser mais rápida, pois trata de gerar a imagem através

de segmentos de reta. Já a renderização por planos preenchidos é mais onerosa, pela quantidade de pontos a mais que tem de desenhar. Porém, ela pode gerar imagens de difícil compreensão, pois as faces de mesma cor podem acabar confundindo o observador. A renderização por *wireframes*, apesar de menos realista, é capaz de dar uma maior sensação de profundidade, quando não são consideradas as questões de iluminação na renderização das cenas, como pode-se observar na Figura 5.12.

O uso de fontes de luz no processo de geração de visualizações de cenas é de grande importância para a correta percepção de formas e para definir um maior grau de realismo nas cenas. Afinal, a percepção de cores do olho humano é função de como a luz é refletida pelos objetos visualizados. Assim, as faces dos objetos não aparentam necessariamente ter a mesma cor, em função da forma como este reflete a luz incidente, da cor dessa luz e do tipo/posição da fonte luminosa.

As fontes de luz em um mundo virtual podem ser do tipo ambiente, direcional ou *spot*. A Figura 5.13 apresenta as características desses três tipos de fontes.

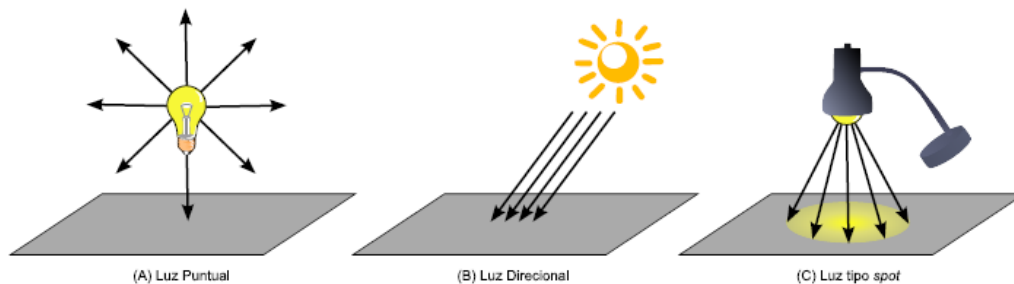


Figura 5.13: Tipos de fontes de luz em computação gráfica (MANSSOUR; COHEN, 2006)

A luz ambiente é considerada uniforme em todas as direções do ambiente, usualmente empregada para um mínimo de visualização da forma de um objeto. Uma fonte de luz direcional é tratada como distante o suficiente para se considerar que todos os raios desta incidem paralelamente sobre uma superfície. Já uma fonte do tipo *spot* é considerada como próxima ao objeto iluminado, formando um cone de luz, ou seja, os raios de luz incidem em ângulos diferentes sobre uma superfície.

Quando a iluminação é considerada no processo de renderização de visualizações de uma cena, a cor de um ponto de uma face é definida a partir de como os raios das fontes de luz incidem sobre essa face, e da cor do ponto em si. Essa interação é representada por diferentes modelos físicos, conforme (MCKESSON, 2012; HILL; KELLEY, 2007).

Para definir um modelo de iluminação correto, é comum definir propriedades adicionais às faces, denominadas de materiais, como as cores ambiente, difusa e especular, bem como o índice de reflexão/espalhamento (*shininess*) da luz. A Figura 5.14 exemplifica essas propriedades.

A cor ambiente define a cor que o objeto deve ter sob o efeito da luz ambiente do cenário, mantendo normalmente a própria cor do objeto. Já a luz difusa representa a cor que esse objeto apresenta quando submetido a uma fonte de luz que não a ambiente. A cor especular define a cor que o objeto emite quando refletindo diretamente uma fonte de luz, cuja área de reflexão é

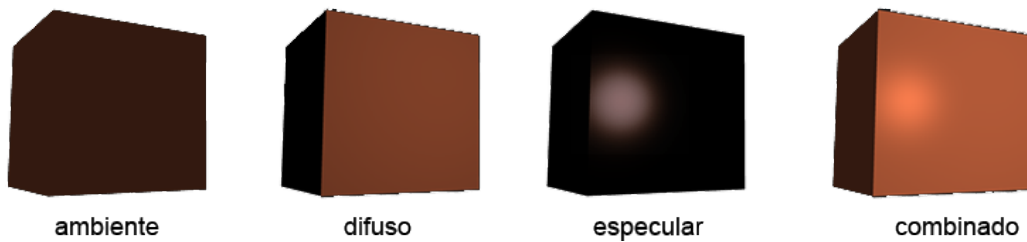


Figura 5.14: Exemplo de iluminação aplicadas a um cubo, adaptado de (LEARN OPENGL, 2017)

definida pelo fator de reflexão/espalhamento especular.

Essas cores são combinadas para cada ponto da face, e ainda são afetadas pelas cores das fontes de luz que incidem sobre o objeto, e o ângulo entre os raios de luz e a normal da face, como mostra a Figura 5.15.

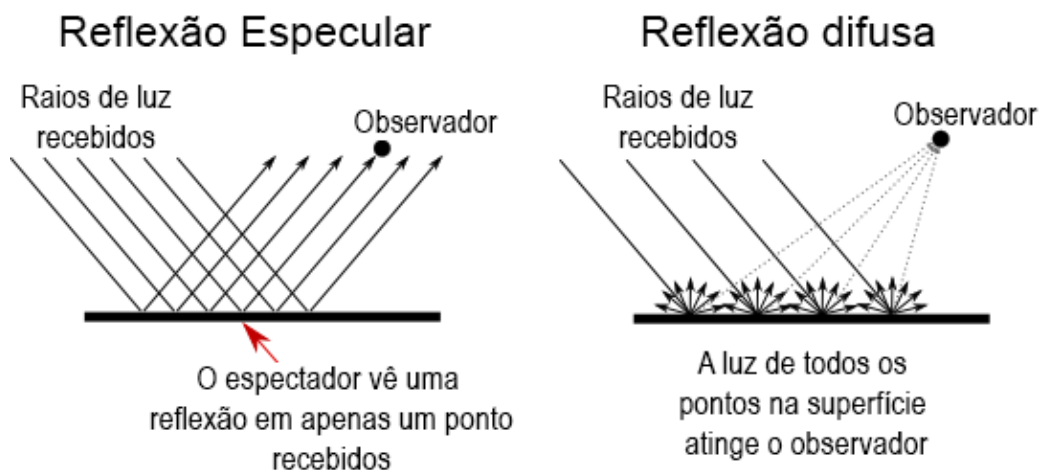


Figura 5.15: Exemplo de reflexão adaptado de (ECK, 2016)

Observa-se que esse processo é feito para cada objeto dentro do volume de visualização da câmera sintética, para cada face visível deste objeto, sendo assim calculada a cor para cada pixel da imagem correspondente a um ponto da face iluminado por uma ou mais fontes de luz do mundo virtual.

5.3 Animação

Uma vez feita a modelagem do mundo virtual e de seus elementos, como as entidades do sistema, as câmeras e fontes de luz, é possível renderizar as cenas de acordo com os volumes de visualização desejados.

Uma animação é função do conjunto de resultados numéricos da simulação, onde para cada instante são calculados os estados do sistema, o que define a postura, forma e cor de cada entidade. Ao se gerar visualizações para cada instante de tempo da simulação, obtém-se um conjunto de quadros (ou *frames*), que apresentados sequencialmente na velocidade correta passará a ilusão de continuidade de movimento.

Uma animação pode ocorrer *online*, onde cada quadro é gerado à medida que os estados são calculados/disponibilizados, ou *offline*, em que todos os quadros são previamente renderizados para após serem apresentados na sequência. O primeiro caso pode gerar atrasos na animação, devido à complexidade da geração dos quadros na frequência mínima para criar a ilusão de movimento. Já o segundo caso, apesar de produzir um resultado com uma animação mais fluída, não pode sofrer modificações durante o processo de animação. Neste último, seria possível gerar arquivos de vídeo que poderiam ser utilizados independentes do software de simulação.

O tipo de animação a ser utilizado é função da interatividade desejada para a simulação. Além disso, é necessário ter a capacidade de processamento numérico e gráfico adequados para cada tipo. Observa-se que uma animação *online* irá requerer mais computação por instante de tempo que uma animação *offline*. Porém, ela pode ser importante para casos em que o usuário deve interagir com a simulação, como em sistemas de treinamento simulado, por exemplo.

6 PROJETO DO SIMULADOR

O processo de projeto e as escolhas para o desenvolvimento do simulador são apresentados neste capítulo.

6.1 Requisitos de Projeto

Com base na revisão da literatura do Capítulo 1, pode-se observar que este trabalho encaixa-se em uma linha de pesquisa do Laboratório de Matemática Aplicada e Controle, de modelagem de sistemas subaquáticos. Assim, os desenvolvimentos prévios são a base para o levantamento dos requisitos do projeto.

Em uma rápida identificação de características do simulador, observou-se que trabalhos anteriores focaram no modelamento numérico de veículos subaquáticos e de estruturas flexíveis do tipo cabo. Alguns dos trabalhos ainda trouxeram a ideia de serem um software *per se*, ou seja, relativamente autônomos, com uma interface com o usuário, sem que este tivesse de trabalhar com códigos/programação. O trabalho de (AGUIAR JR, 2016) foi o primeiro, da linha de pesquisa do laboratório, focado na modelagem gráfica e animação de estruturas flexíveis do tipo cabo, utilizando os resultados de simulações numéricas de trabalhos anteriores. Esse trabalho concentrou-se em desenvolver um animador tridimensional para estruturas do tipo cabo somente, não contendo ambientação realista e a possibilidade de movimento no entorno do cabo durante a animação. O presente trabalho é uma expansão e evolução de (AGUIAR JR, 2016).

A modelagem gráfica de ROVs e das interações destes com os cabos umbilicais, porém, não havia sido considerada, apesar dos trabalhos do grupo já terem resultados significativos em termos de modelagem numérica. Assim, esta foi a finalidade definida para o software, que foi nomeado USim (de *Underwater Simulator*). Em função dessa finalidade, dos desenvolvimentos anteriores (alguns dos quais, gerando os resultados a serem visualizados através do USim) e das características desejadas para o software, foram identificados os requisitos (FOWLER, 2014).

Como requisitos funcionais, tem-se:

Gerar visualizações realistas O principal recurso que o software deve prover é um meio de representar o ambiente subaquático e os elementos do sistema dinâmico a ser simulado, como os ROVs e os cabos, bem como gerar visualizações deste ambiente virtual de uma forma visual realista. Diferentes visualizações de uma mesma cena poderiam ser obtidas através de pontos de vistas (ou câmeras) diferentes, estáticos ou móveis ao longo do tempo.

Realizar animações Além de gerar visualizações de cenas do ambiente subaquático e dos elementos do sistema dinâmico simulado, o software deve ser capaz de animá-lo, a partir da rápida sucessão de visualizações de cenas configuradas pelos resultados de cada instante de uma simulação numérica.

Prover informações complementares Além de visualização do ambiente virtual, o software deve ser disponibilizar meios de complementar a análise, como através da geração de um rastro do movimento do ROV e apresentar sistemas de coordenadas auxiliares nos elementos do sistema. Além desses recursos inerentes ao ambiente virtual, deve ser possível observar gráficos relativos às variáveis da simulação desejadas.

Compatível com dados de simulações numéricas O software deve ser capaz de ler dados gerados pelas simulações numéricas correspondentes às dinâmicas de ROVs, cabos, e da interação entre estes, previamente desenvolvidos no NuMA-FURG (usualmente em MATLAB).

Prover uma interface com o usuário Para que o usuário do software possa configurar os parâmetros de visualização e animação, bem como definir os arquivos numéricos que devem ser usados, o USim deve ter uma interface que permita realizar essas funções, bem como outras relacionadas à execução da animação, como movimentar câmeras e capturar/exportar instantâneos de cenas.

Ser extensível Como os sistemas podem mudar quanto ao número de objetos e à natureza destes, o USim deve evoluir organicamente. Para tanto, deve ser capaz de ser facilmente extensível, através da agregação de modelos de ROVs, cabos e outros possíveis elementos de sistemas subaquáticos de forma sistemática e padronizada.

Em termos de requisitos não funcionais, foram identificados:

Executar em diferentes plataformas A questão de multiplataforma é interessante para possibilitar o uso do software em diferentes ambientes computacionais utilizados no meio acadêmico e de pesquisa.

Compatibilidade com softwares de modelagem A fim de agilizar o processo de modelagem dos ambientes virtuais, seria desejável que o simulador fosse compatível com formatos de objetos de softwares de modelagem/CAD comumente empregados para essa finalidade.

Ser aberto e gratuito A questão de ser aberto é vinculada ao requisito funcional de ser extensível. Dessa forma, os usuários, ao terem acesso ao código do simulador e à sua estrutura, podem contribuir para a evolução deste, provendo modelos de novos objetos e suporte a diferentes hardwares. O fato de ser gratuito, por sua vez, possibilitará o acesso a mais usuários, auxiliando na divulgação científica das pesquisas do laboratório.

6.2 Modelagem do Simulador

A análise dos requisitos procurou determinar um projeto de software que os satisfizessem tanto em termos de operação quanto em implementação. Para o primeiro item, foi feita uma modelagem de como o usuário interagiria com o simulador, através de um diagrama de casos de uso. O segundo item foi analisado e modelado através de um diagrama de classes,

ambos parte da UML (do inglês *Unified Modeling Language*, ou Linguagem de Modelagem Unificada(BOOCH; RUMBAUGH; JACOBSON, 2006).

Os casos de uso descrevem formas de interação entre atores e o sistema. Um ator pode tanto ser o papel que um usuário exerce quando utilizando o software como outro sistema/software que interage com o sistema modelado de alguma forma (BOOCH; RUMBAUGH; JACOBSON, 2006). O diagrama da Figura 6.1 apresenta as interações dos usuários com o USim. Observa-se também que há um ator externo, chamado Simulador Numérico, que se relaciona com o USim pelo fornecimento dos resultados da simulação numérica.

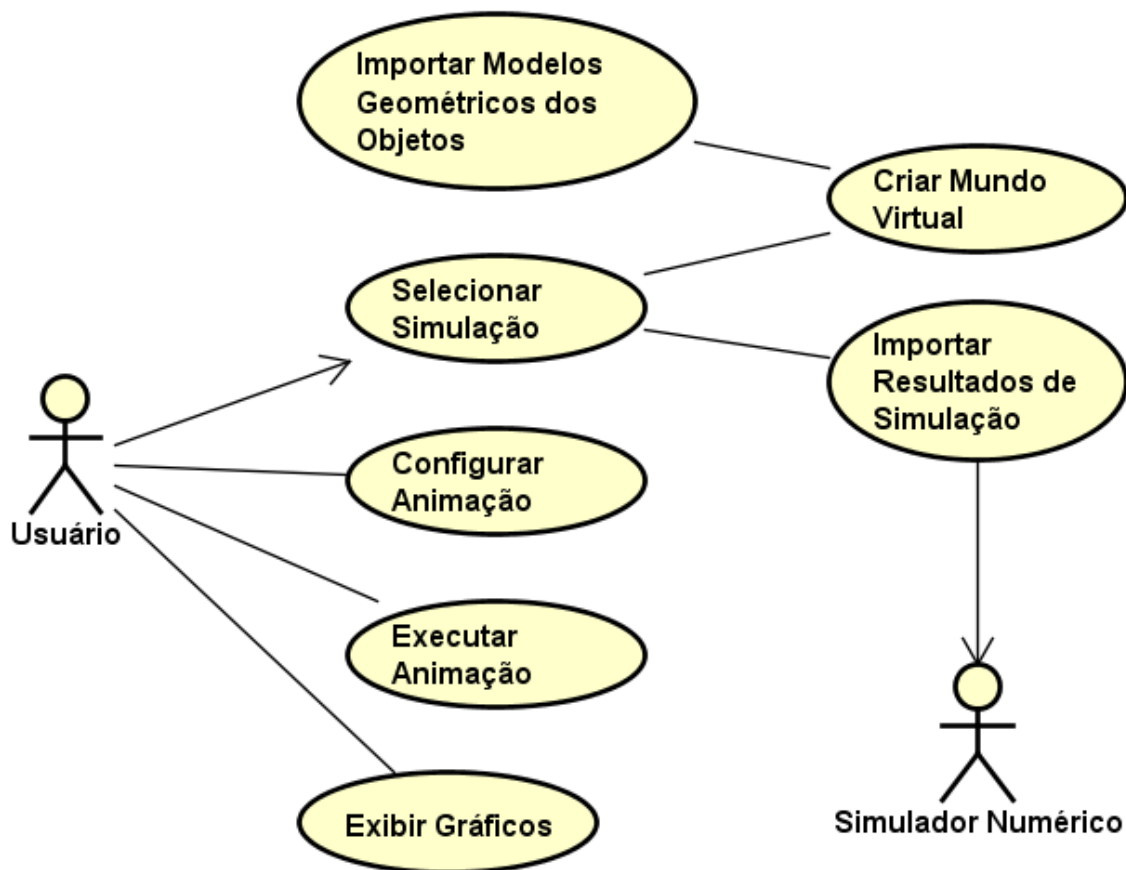


Figura 6.1: Casos de uso do USim

O diagrama descreve as funcionalidades do sistema do ponto de vista do usuário, que inicia a interação com o sistema selecionando uma animação, onde então é possível para ele, configurar e executar essa animação, bem como, exibir o gráfico a ela associado. Ao selecionar uma animação, ele importa os dados a partir de um módulo que atualmente é externo, pois está em desenvolvimento, que provê os resultados numéricos da simulação. Então o animador cria o mundo virtual, que por sua vez, importa os modelos geométricos da simulação do usuário.

O diagrama de classes é classificado como estrutural, apresentando uma visão estática de como os componentes do software estão organizados e se relacionam(FOWLER, 2014). Ele é adequado para o paradigma da orientação a objetos (OO), que foi adotado para o desenvolvimento do USim em função das características modulares, extensíveis e evolutivas que as

ferramentas baseadas neste paradigma oferecem.

Os principais componentes de software do USim estão presentes no diagrama de classes da Figura 6.2. Pode-se observar que o software é organizado em pacotes, cada qual contendo classes que tenham responsabilidades correlatas. Os relacionamentos entre objetos (instâncias das classes) são representados pelas linhas e seus terminadores.

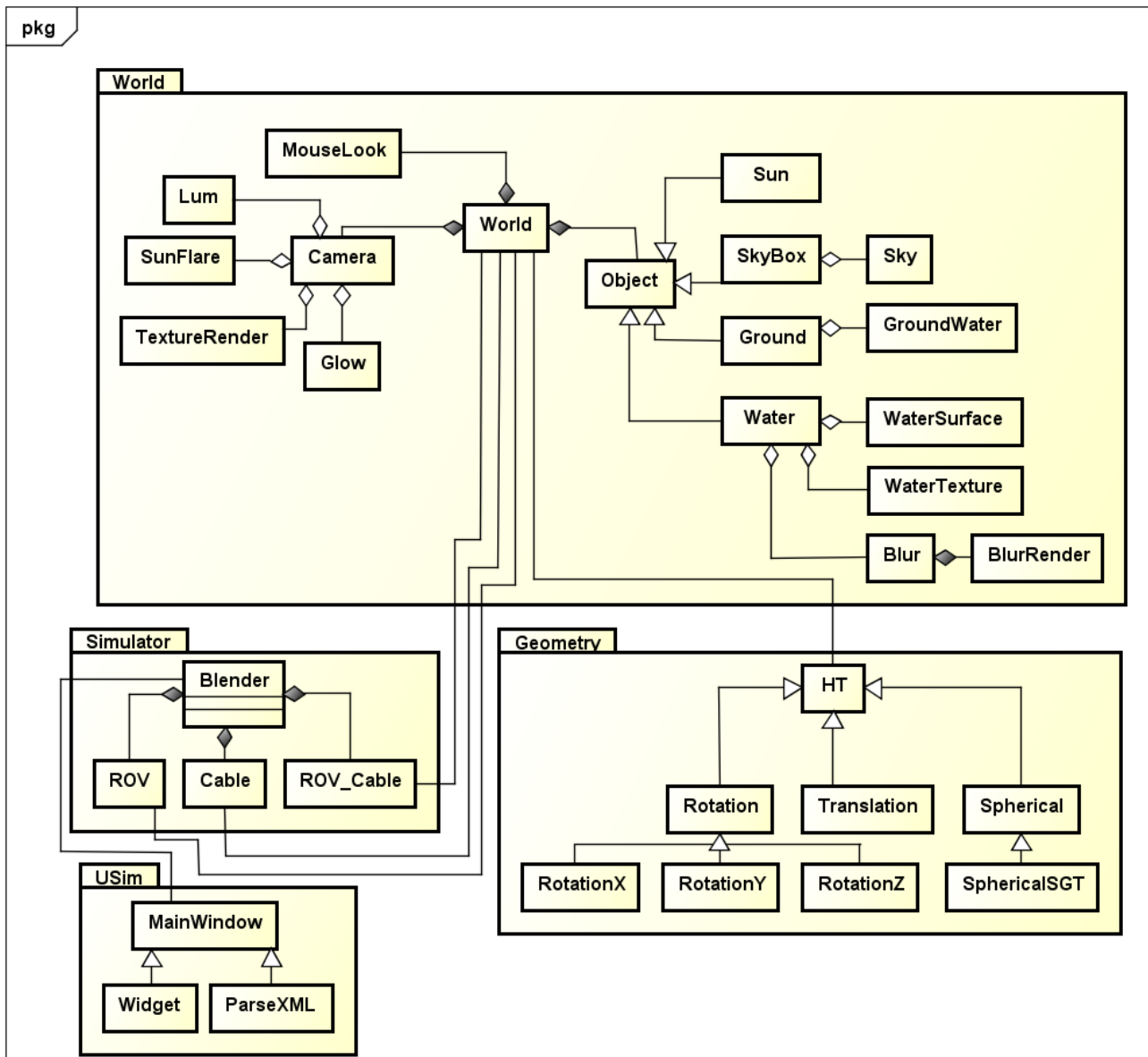


Figura 6.2: Diagrama de classes do USim

O pacote *USim* contém as classes que gerenciam a execução do aplicativo e a interface gráfica com usuário. A interface gráfica e o fluxo de execução são geridos por uma instância de *MainWindow*. Esta agrega diversos *Widgets* gráficos para formar a interface com o usuário. Uma instância de *ParseXML* é responsável pela leitura e processamento dos arquivos de configuração da simulação.

O ambiente virtual e seus componentes são modelados por instâncias das classes do pacote *World*. Uma classe com o mesmo nome, *World*, é um *container* de instâncias de objetos do

ambiente, como água (*Water*), céu (*SkyBox*), terra (*Ground*) e câmeras (*Camera*), estas com definições de iluminação e parâmetros de renderização de cenas, além de posição e orientação. A classe *Object* é usada como base para definir objetos de cena, tanto importados de ferramentas de modelagem quanto criados a partir de primitivas geométricas/superfícies paramétricas. Em função do ambiente de renderização/animação escolhido, essas classes ainda permitem a interação com o usuário, como a movimentação de câmeras utilizando mouse, por exemplo.

O pacote *Simulator* contém classes que representam os componentes ativos da simulação como ROVs (*ROV*), Cabos (*Cable*) e a interação entre estes (*ROV_Cable*). Instâncias dessas classes são agregadas ao mundo (instância de *World*). O renderizador das cenas e animação é representado pela classe *Blender*, que faz a ligação da aplicação com o motor gráfico (*engine*) de mesmo nome. O motivo de escolha desta solução será explicado na próxima seção.

O pacote *Geometry* foi originalmente desenvolvido em AGUIAR JR (2016), para simulação de estruturas do tipo cabo. Ele contém um conjunto de classes que representam matrizes de transformação homogêneas específicas, todas realizações da classe base *HT*. Assim, estão disponíveis classes para translações puras (*Translation*) e rotações segundo os eixos coordenados X (*RotationX*), Y (*RotationY*) e Z (*RotationZ*). Transformações de orientação específicas, como em torno de um eixo (*RotationZ*), e baseadas em ângulos de azimute e inclinação também estão definidas (*Spherical* e *SphericalSGT*). Todas as transformações podem ser compostas a partir da multiplicação entre as respectivas instâncias, permitindo representar movimentos/posturas complexos a partir de transformações mais simples (ou primitivas)(AGUIAR JR, 2016).

As classes *Spherical* e *SphericalSGT* são utilizadas na determinação de postura dos elos virtuais que compõem a representação gráfica dos cabos. A partir dos valores de elevação, azimute e torção para cada elo virtual a cada instante da simulação numérica(ZANELA, 2013), são calculadas as matrizes de transformação homogênea que posicionam cada elo em relação à base do cabo de forma recursiva, ou seja, a postura de um elo i é resultado da transformação de *SphericalSGT* específica desse elo multiplicada pelas transformações dos elos anteriores de 1 a $i - 1$.

Apesar de se desejar evitar utilizar componentes de software específicos no momento da modelagem conceitual, tornou-se necessário fazer algumas definições em função das soluções já utilizadas em trabalhos anteriores, como o de AGUIAR JR (2016), bem como de experiências prévias com algumas ferramentas de desenvolvimento, que nortearam, naquele momento, a própria estrutura do software.

De todo modo, o processo de escolha de ferramentas de desenvolvimento, abordado a seguir, foi iniciado quase em paralelo com a modelagem, pois algumas possíveis soluções demandavam experimentar exemplos de códigos para avaliação de adequação aos requisitos.

6.3 Ferramentas de Desenvolvimento

As escolhas das ferramentas utilizadas para o desenvolvimento do software de simulação foram baseadas na necessidade de atendimento dos requisitos identificados, bem como na pos-

sibilidade de aproveitamento de trabalhos anteriores do grupo, como o de AGUIAR JR (2016), que criou uma ferramenta para visualização de simulações de cabos.

A linguagem de programação base adotada para o desenvolvimento do USim foi a Python (VAN ROSSUM et al., 2007), em função de suas características que a colocam entre uma linguagem de programação de aplicações e um sistema de computação algébrica (ou CAS, do inglês *Computer Algebra System*). Assim, é possível pensar no desenvolvimento de aplicações *standalone*, que não dependem de ambientes específicos para a sua execução (ROCHA, 2012). Além disso, ela satisfaz os requisitos não funcionais, como a questão da multiplataforma e possibilidade do software ser *open source* e gratuito.

Em aplicações científicas, Python e seu *ecossistema* oferecem um conjunto de APIs diversas, com uma grande base de conhecimentos formada por empresas e usuários, usualmente disponível gratuitamente. Essas APIs costumam sofrer atualizações/manutenções constantes, colaborando para o aumento do desempenho e estabilidade das soluções. A facilidade de criar interfaces com APIs desenvolvidas em outras linguagens (*bindings*), como C/C++ e FORTRAN, e com aplicações de terceiros, como Blender, Inkscape, LabView, entre outros, ampliam as soluções para criação de aplicações. Em relação ao desenvolvimento de interfaces gráficas, existem diferentes opções para a criação de telas e *widgets*, como wx, GTK, Qt, FLTK, e MFC/Windows (PYTHON SOFTWARE FOUNDATION, 2017). Destas, foi escolhido o Qt, por ser multiplataforma, com apresentação consistente entre sistemas operacionais; estável; ter APIs completas como o PyQt (RIVERBANK COMPUTING, 2017); e prover ferramentas para a prototipação rápida de interfaces gráficas.

Para a renderização de cenários do mundo virtual, pesquisou-se por uma solução que possibilitasse representar mundos e gerar visualizações em um nível mais alto que o uso direto de OpenGL (SHREINER et al., 2013), utilizado no trabalho de AGUIAR JR (2016). Apesar de efetivo, o uso desta API de baixo nível demandou o desenvolvimento de quase toda a parte de modelagem e iluminação. Assim, buscou-se por um *motor gráfico* que tivesse um *binding* Python, tivesse recursos de importação de objetos desenvolvidos em softwares de modelagem/CAD, permitisse a construção dos cenários com base nesses objetos e renderizasse as cenas com base no conceito de câmeras sintéticas/iluminação com velocidade adequada. Entre os motores estudados, foi escolhido o Blender Game Engine, parte do software *Blender* (BLENDER FOUNDATION, 2016). Este *engine* realiza a interface entre o USim e o Blender, permitindo ao primeiro utilizar o modelador para renderizar mundos representados de acordo com o modelo do Blender.

Em termos de processamento matemático, foram utilizadas as APIs Numpy, SciPy e Matplotlib. As primeiras fornecem as funcionalidades numéricas similares às encontradas na maioria dos softwares de álgebra computacional (DEVELOPERS, 2013). O Matplotlib é uma API para a construção de gráficos científicos a partir de estruturas de dados matemáticas do Numpy (HUNTER, 2007). Além de prover um conjunto amplo de tipos de gráficos bidimensionais e tridimensionais, com gerenciadores de visualização, esta API permite a integração as APIs gráficas, como o Qt, e exportação de imagens em diferentes formatos gráficos.

Embora tanto o interpretador Python quanto as APIs empregadas no desenvolvimento ser instaladas individualmente, foi utilizada a distribuição *Python(x,y)* (RAYBAUT; NYO, 2015). Esta distribuição é voltada para o uso científico de Python em sistemas operacionais Microsoft Windows. Além do interpretador, ela acrescenta um conjunto de APIs científicas empregados em diferentes áreas, bem como um ambiente integrado de desenvolvimento chamado *Spyder*, que permite o uso do Python tanto de forma interativa quanto programática.

As ferramentas utilizadas permitiram o desenvolvimento da primeira versão do USim, que fornece um modelo de ROV (o NEROV) e modelos diversos de cabos, consistentes com as simulações dos trabalhos que geram os resultados numéricos para as simulações.

7 ASPECTOS DE DESENVOLVIMENTO

7.1 Modelagem do Ambiente

A construção do mundo virtual onde as simulações/animações ocorrem é feita a partir da criação dos objetos a serem simulados, como os elementos da simulação – ROVs e estruturas flexíveis do tipo cabo – e de objetos de cena, necessários para definir um certo grau de realismo nas cenas. Além destes, objetos auxiliares são eventualmente empregados, como sistemas de coordenadas auxiliares. A renderização de cenas também demanda objetos como câmeras sintéticas e fontes de luz.

A modelagem desses objetos é criada com o Blender, cujo *engine* gráfico foi escolhido como ferramenta de renderização/animação. O software de modelagem contém um conjunto de ferramentas para a criação de objetos gráficos e para visualização, como modelagem geométrica, iluminação, câmeras sintéticas e definição de texturas. Ele já incorpora um módulo que trata de evitamento de colisão, onde dois objetos sólidos não compartilham o mesmo espaço em caso de colisão. Animações podem ser especificadas por linhas do tempo e scripts (BLENDER FOUNDATION, 2017a). A Figura 7.1 apresenta a interface do software.

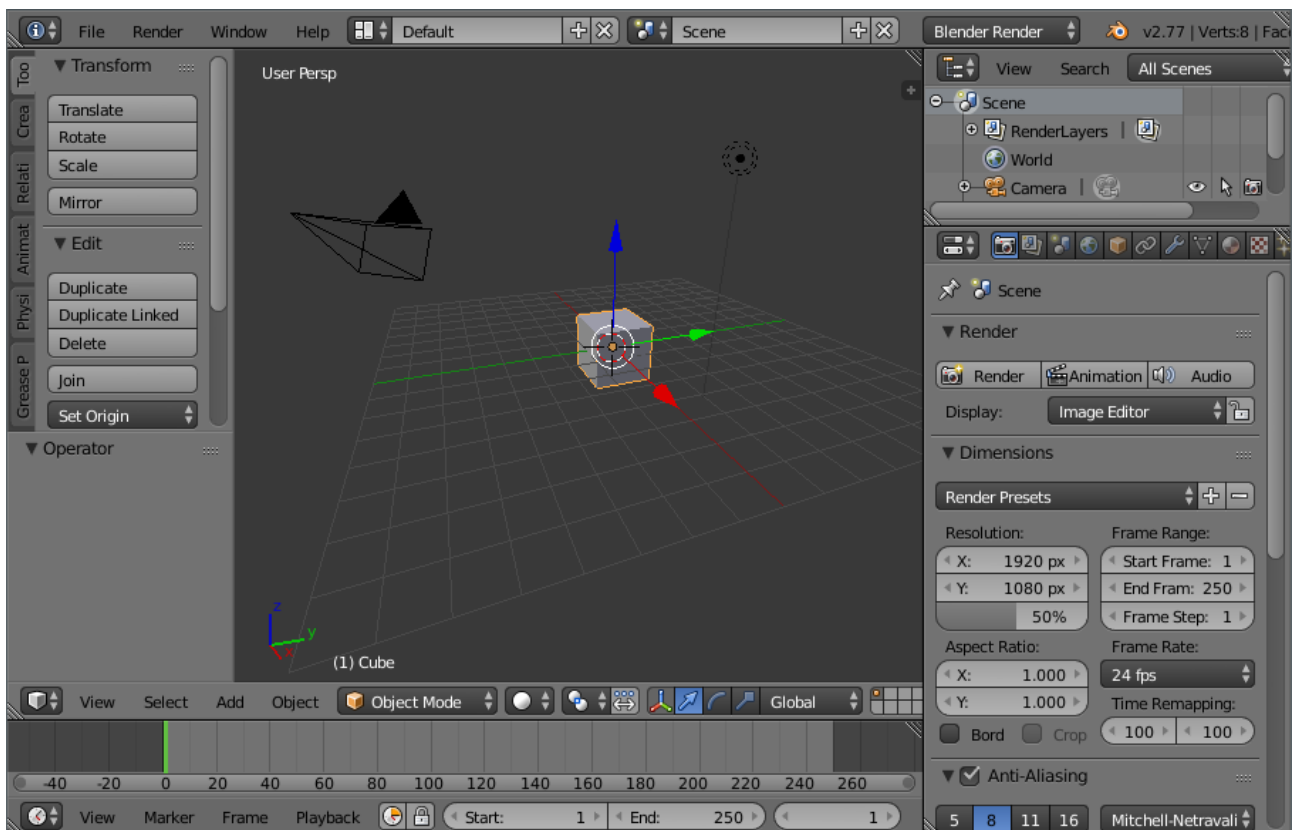


Figura 7.1: Interface principal do Blender (BLENDER FOUNDATION, 2017a)

Partindo de um mundo vazio, ou seja, um volume definido, devem ser inseridos objetos que componham o ambiente da simulação. No caso do USim, o cenário é formado por um céu e uma massa de terra, que delimitam o ambiente. A iluminação é fornecida por um Sol. Um

corpo d'água completa a simulação, sendo onde os elementos do sistema dinâmico simulado se movimentam. Esses objetos são criados através de primitivas do Blender, como planos, cubos e cilindros, que podem ser posteriormente editados para ajustar formas ao que se deseja.

O céu é definido por um objeto *SkyBox*, que é um cubo com textura que dá a impressão de um céu azul, com poucas nuvens. A terra é definida por um objeto *Ground*, criado a partir de um plano cuja malha de pontos é posteriormente modificada (ou esculpida) para gerar os efeitos de relevo. O solo do fundo do corpo d'água é definido por um objeto *GroundWater*, que define a cor e textura do fundo do corpo d'água. Isso é observável na Figura 7.2, juntamente com os efeitos de água.



Figura 7.2: Ambiente de simulação

A superfície da água é representada por um objeto *Water*, que é um plano dinamicamente modificado através de um script de animação chamado *Ocean Modifier* (BLENDER FOUNDATION, 2017b). Este trabalha a modificação da malha de um plano considerando-o uma superfície paramétrica Bézier, de forma a reproduzir o movimento típico de uma superfície de um corpo d'água. A Figura 7.3 apresenta um exemplo de superfície gerada pelo script. A superfície é configurável pela variação da resolução, do tamanho de ondas e da velocidade/direção do vento. Ainda é possível definir como a luz é refletida pela superfície, como se observa na Figura 7.2.

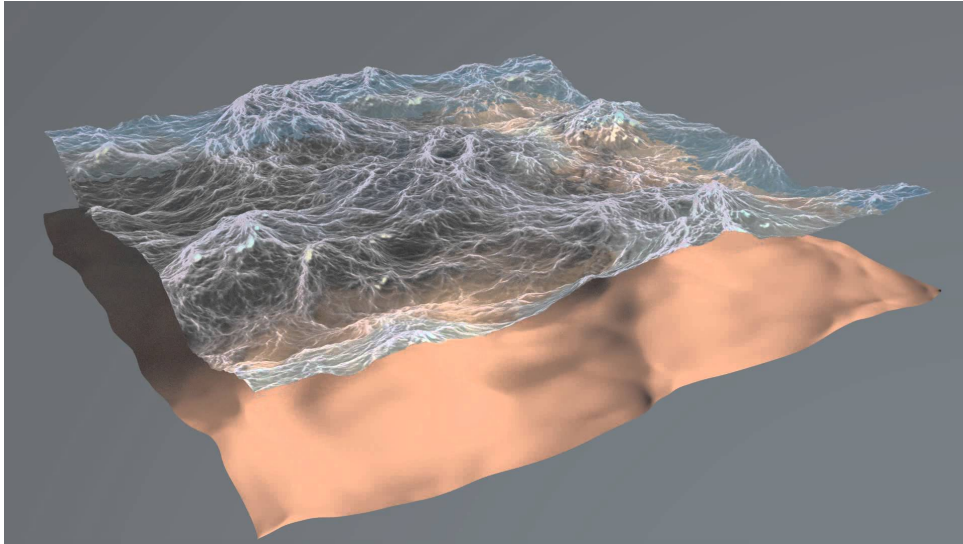


Figura 7.3: Geração de superfícies aquáticas a partir do *Ocean Modifier* (RUDOLPH, 2014)

A luz do cenário é fornecida por um objeto *Sun*, definido como uma fonte de luz direcional, simulando o Sol.

As câmeras sintéticas são objetos já existentes no Blender, que as utiliza justamente para a visualização de cenas sob diferentes pontos de vista. Essas câmeras, além de poderem ser posicionadas em qualquer ponto do mundo virtual, ainda podem ser comandadas pelo usuário através de uma interface inerente a elas fornecida pelo software. A Figura 7.4 identifica os graus de liberdade em que a câmera pode se movimentar. O usuário movimenta a câmera através do mouse e do teclado. A Tabela 2 relaciona as teclas de ação para cada movimento.



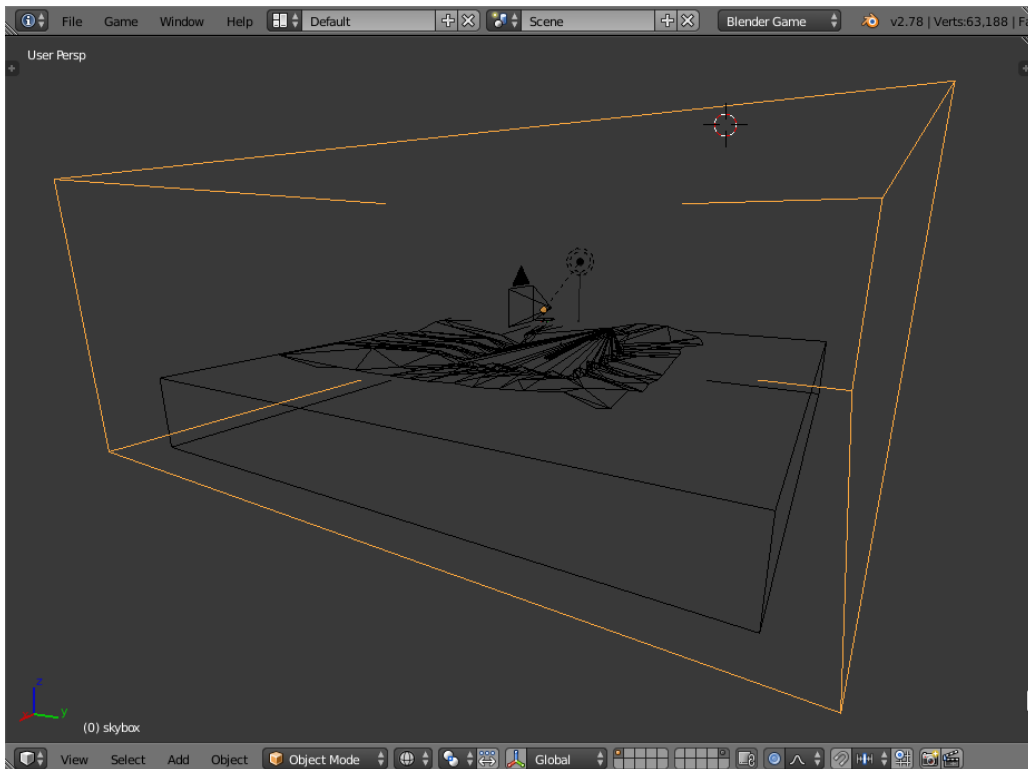
Figura 7.4: Câmera sintética e seus possíveis movimentos (adaptado de PETERSON (2015))

Uma visão dos elementos do cenário correspondentes ao ambiente é apresentada na Figura 7.5, onde os objetos são renderizados em *wireframes*, de forma que é possível observar a delimitação do ambiente pelo *SkyBox* (em contorno laranja) e o terreno, em contorno preto.

Outra visão em *wireframes* é apresentada na Figura 7.6, onde é possível observar o navio utilizado como base e indicador do sistema de coordenadas de referência do ambiente, e o ROV da simulação. A Figura 7.7 mostra outra visualização em *wireframe* do ROV e do navio, incluindo as câmeras utilizadas no simulador (representadas por pirâmides).

Tabela 2: Movimentos de câmera e comandos de teclado

Comando	Ação	Movimento
Mouse	Movimento Livre	Roll
		Pitch
		Yaw
Teclado	Tecla W	Frente
	Tecla S	Atrás
	Tecla D	Direita
	Tecla A	Esquerda
	Seta Acima	Acima
	Seta Abaixo	Abaixo

Figura 7.5: Visão *wireframe* do ambiente sintético

O *Blender Engine* (ou BGE) é o renderizador embutido no USim. Ele é uma especialização do *engine* que o Blender utiliza no processo de modelagem, otimizado para animações em tempo real, criando continuamente cenas e permitindo a interação do usuário ao longo do processo. Nesse modo, a renderização passa a considerar cores e texturas das superfícies dos objetos, bem como o efeito de iluminação sobre eles. A Figura 7.2 foi renderizada utilizando o BGE. Outro exemplo é mostrado na Figura 7.8, com uma câmera posicionada no fundo com foco no ROV, onde se pode ver o efeito de ondas na superfície da água.

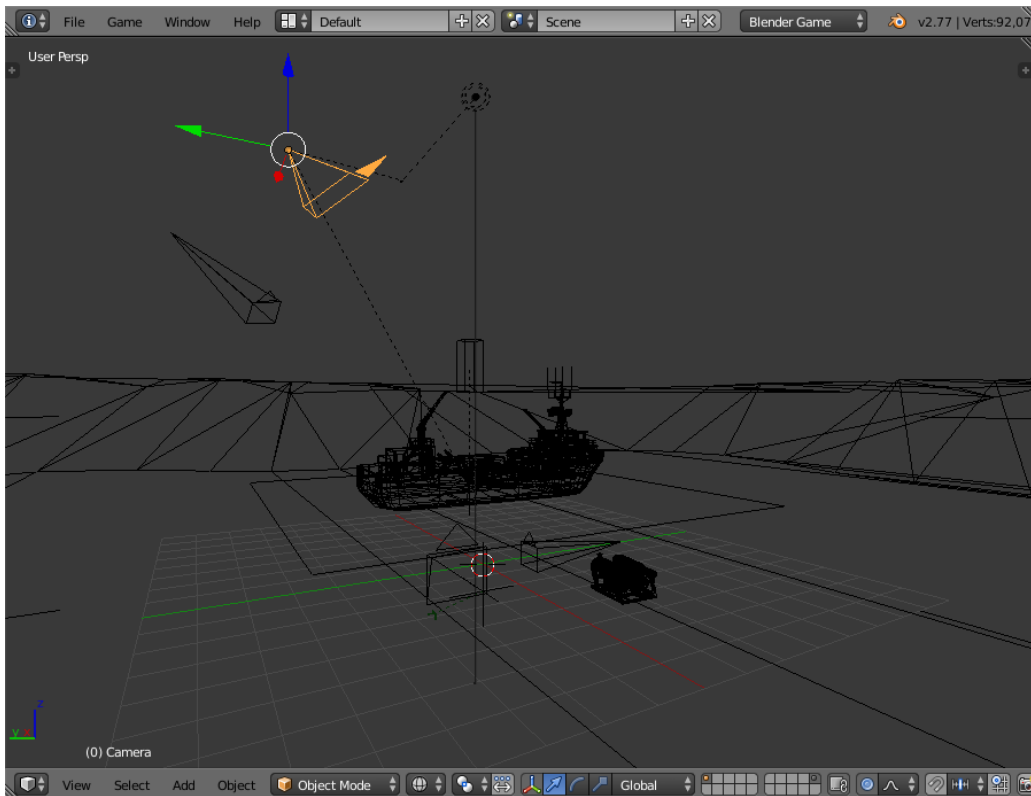


Figura 7.6: Visão *wireframe* do ambiente sintético, com destaque para o ROV e o navio

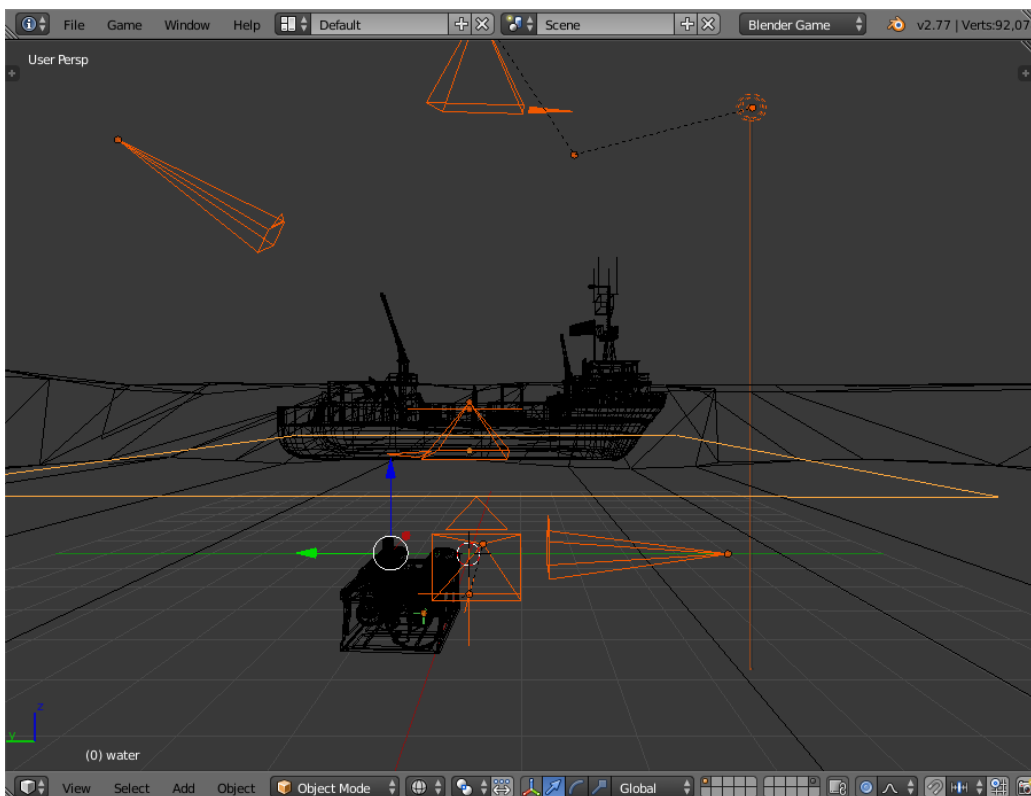


Figura 7.7: Visão *wireframe* do ambiente, com o ROV e as câmeras sintéticas

7.2 Animação

Uma vez criado o mundo virtual, é possível realizar as animações a partir dos resultados das simulações numéricas. Para tanto, as simulações devem exportar esses resultados em um arquivo

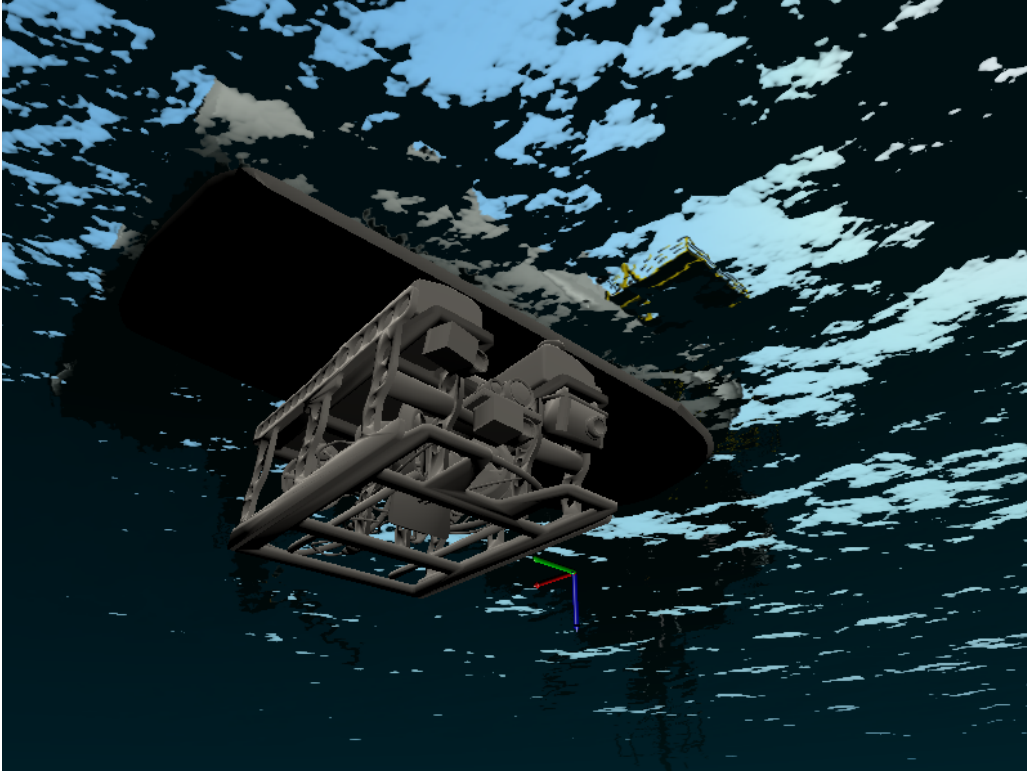


Figura 7.8: Visualização do ambiente subaquático, renderizada pelo Blender *engine*

texto, onde cada linha corresponde a um instante da simulação. Os parâmetros correspondem às posições e orientações dos objetos no cenário virtual, ou no caso dos cabos, às posturas relativas de cada elo virtual.

7.2.1 Simulações de ROVs

No caso de simulações da dinâmica de ROVs livres, o arquivo deve fornecer a posição e orientação para cada instante da simulação. Cada linha do arquivo corresponde a um instante, contendo as coordenadas do centro de massa do ROV em relação ao referencial inercial e os ângulos *roll-pitch-yaw*, separados por espaços. Este formato textual de estruturação de dados é usualmente conhecido como *csv* (campos separados por vírgulas, ou *comma separated values*, em inglês), e é utilizado em situações em que se deseje compartilhar dados entre diferentes aplicações. A Tabela 3 apresenta a ordem de cada linha de dados do arquivo de simulação, onde x , y e z são as coordenadas do veículo e ϕ , θ e ψ são os ângulos de rolagem, arfagem e guinada (ou *roll*, *pitch* e *yaw*) respectivamente.

Tabela 3: Campos dos arquivos de dados de simulação de ROVs

Posição			Orientação		
x	y	z	ϕ	θ	ψ

A simulação pode ser em malha aberta, ou em malha fechada, onde um controlador é utilizado para fazer com que o ROV siga uma trajetória desejada. Neste caso, a simulação

deve gerar um arquivo texto contendo os dados da trajetória desejada, no mesmo formato do arquivo de dados de postura do ROV. A Figura 7.9 ilustra um trecho de um arquivo de dados de simulação.

0.0152	9.9966	30.0117	0.0060	0.0010	1.5874
0.0589	9.9895	30.0191	0.0114	0.0025	1.6040
0.1289	9.9809	30.0247	0.0170	0.0022	1.6231
0.2220	9.9740	30.0299	0.0231	0.0045	1.6444
0.3353	9.9642	30.0348	0.0280	0.0061	1.6626

Figura 7.9: Exemplo de dados de um arquivo de resultados de simulação de ROV

7.2.2 Simulações de Estruturas Flexíveis do Tipo Cabo

A animação do cabo baseia-se utiliza a mesma estrutura de dados de arquivos texto de AGUIAR JR (2016), que importa o resultado de simulações numéricas de ZANELA (2013). Nele, cada linha corresponde a um instante da simulação, contendo os valores dos ângulos de elevação, azimute e torção para cada elo virtual utilizado na discretização do cabo simulado. Esses dados são sintetizados na Tabela 4.

Tabela 4: Campos dos arquivos de dados de simulação de Cabos

Ângulos			Tempo
Elevação	Azimute	Torção	Segundos
n	n	n	t

A última coluna de cada linha contém o instante de tempo da simulação. A Figura 7.10 demonstra o conjunto de parâmetros para um conjunto de n elos virtuais.

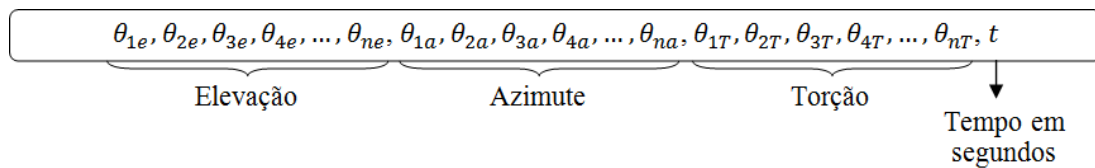


Figura 7.10: Estrutura de dados da discretização de um cabo considerando n elos virtuais (AGUIAR JR, 2016)

7.2.3 Simulações com Interação Entre ROV e Cabo

Nesta simulação, considera-se o ROV como a carga terminal do cabo. Assim, o cabo pode sofrer esforços devido a dinâmica do cabo. O arquivo texto de parâmetros do cabo mantém a estrutura de dados com cabos apenas. O ROV é considerado como um elo virtual vinculado ao último elo do cabo.

7.2.4 Configuração de Simulações

Além dos arquivos de dados, utiliza-se um arquivo de configuração dos parâmetros como a quantidade de instantes da simulação, o passo de animação, a quantidade e o comprimento dos elos.

O passo é definido para que a animação seja executada no tempo definido para a simulação. No caso da quantidade de instantes ser maior do que a possível para a renderização em tempo real de simulação, o passo pode ser ajustado para que alguns dos instantes de simulação possam ser descartados.

Esses parâmetros a dados complementares da simulação são armazenados em um arquivo texto, estruturados no formato XML (*eXtensible Markup Language*). Parâmetros auxiliares são acrescentados, para melhor descrever a simulação, como uma descrição e os caminhos dos arquivos de simulação. Um exemplo de arquivo de configuração é mostrado na Figura 7.11.

```
<?xml version='1.0' encoding='UTF-8'?>
<USim>
  <About autor="Marcos Vinicius Scholl" comments="Cabeçalho da animação 3D de estruturas flexíveis
do tipo cabo" email="marcos.vinicius.scholl@gmail.com" status="beta" version="20170801-1400" />
  <Description>Simulação Padrão: 10 segundos Elevação + Azimute + Torsão</Description>
<Header>
  <project name="project">FURG - PPGMC</project>
  <step name="step">2</step>
  <size_elo name="sizeElo">4.0</size_elo>
  <number_elo name="numberElos">4</number_elo>
  <number_points name="numberPoints">2001</number_points>
  <file_header name="file_header">/USim/simulations/cable/1elos4sim10s.xml</file_header>
  <file_simulation name="file_simulation">/USim/simulations/cable/1elos4sim10s.txt</file_simulation>
</Header>
</USim>
```

Figura 7.11: Exemplo de arquivo XML de configuração de uma simulação de cabo

A biblioteca *ElementTree* é utilizada para o processamento dos arquivos de configuração (PYTHON SOFTWARE FOUNDATION, 2015).

7.3 Interface Com o Usuário

A tela inicial do USim é apresentada na Figura 7.12. Nela, o usuário deve escolher uma das opções de cenário de simulação possíveis, que seriam simulação de um ROV, simulação de uma estrutura flexível do tipo cabo, ou simular um ROV conectado por um cabo a uma base.

Para organizar os arquivos das diferentes simulações, o software tem estruturados uma árvore de diretórios onde os arquivos para cada tipo de simulação devem estar armazenados, conforme mostrado na Figura 7.13. Para cada arquivo de simulação numérica (nomeados com extensão .txt) deve haver um arquivo de parâmetros da simulação (com mesmo nome dos dados, mas com extensão .xml).

Ao escolher o tipo de cenário, o Usim relaciona as simulações disponíveis no diretório correspondente para o usuário escolher qual será carregada para animação, na janela apresentada na Figura 7.14. Opções como gerar rastro do ROV ou da carga terminal, e mostrar sistemas de coordenadas são selecionáveis para aparecer durante a animação.

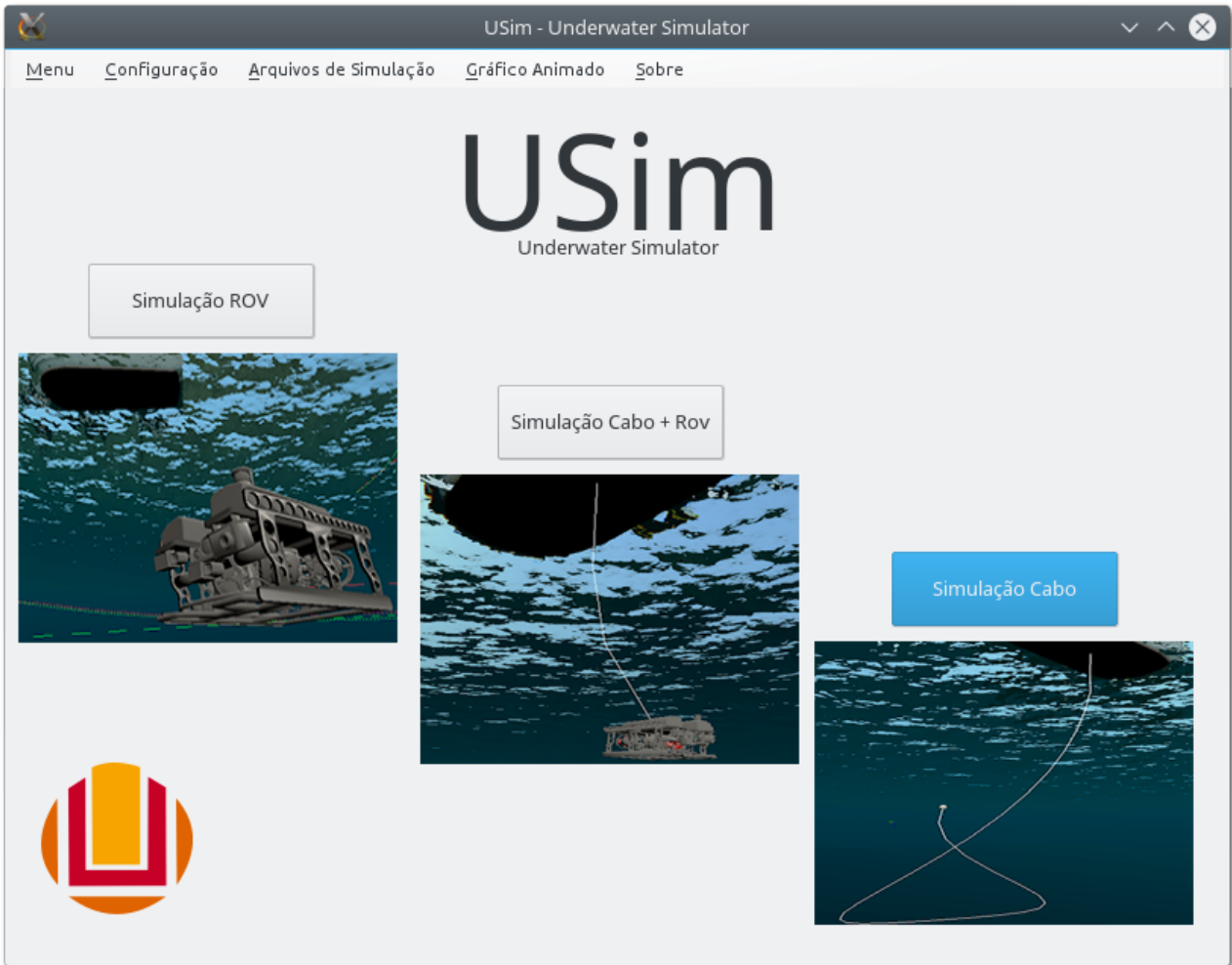


Figura 7.12: Interface com o usuário do USim

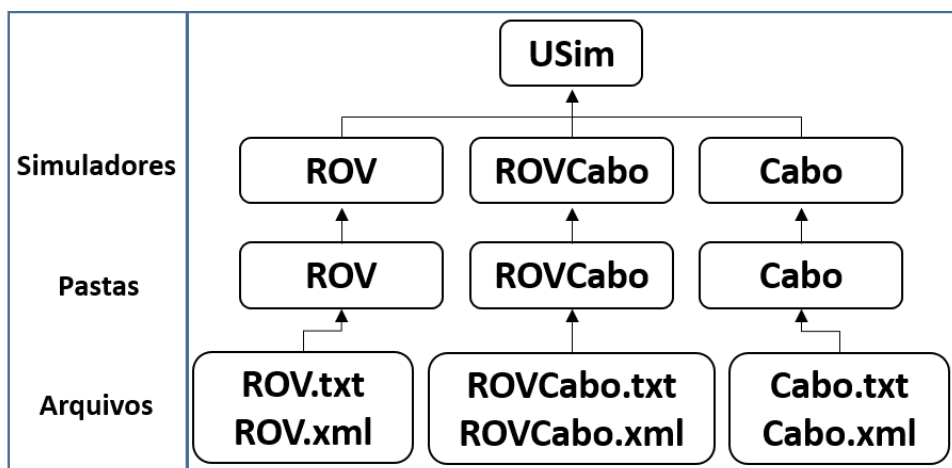


Figura 7.13: USim - Configuração de software

Após o usuário definir a simulação e as opções de visualização, o software carrega os dados dos arquivos, processa-os e apresenta a janela do cenário da simulação, iniciando a animação no ambiente virtual.

Durante a simulação, são disponibilizadas três câmeras. O usuário pode ativar a visualização

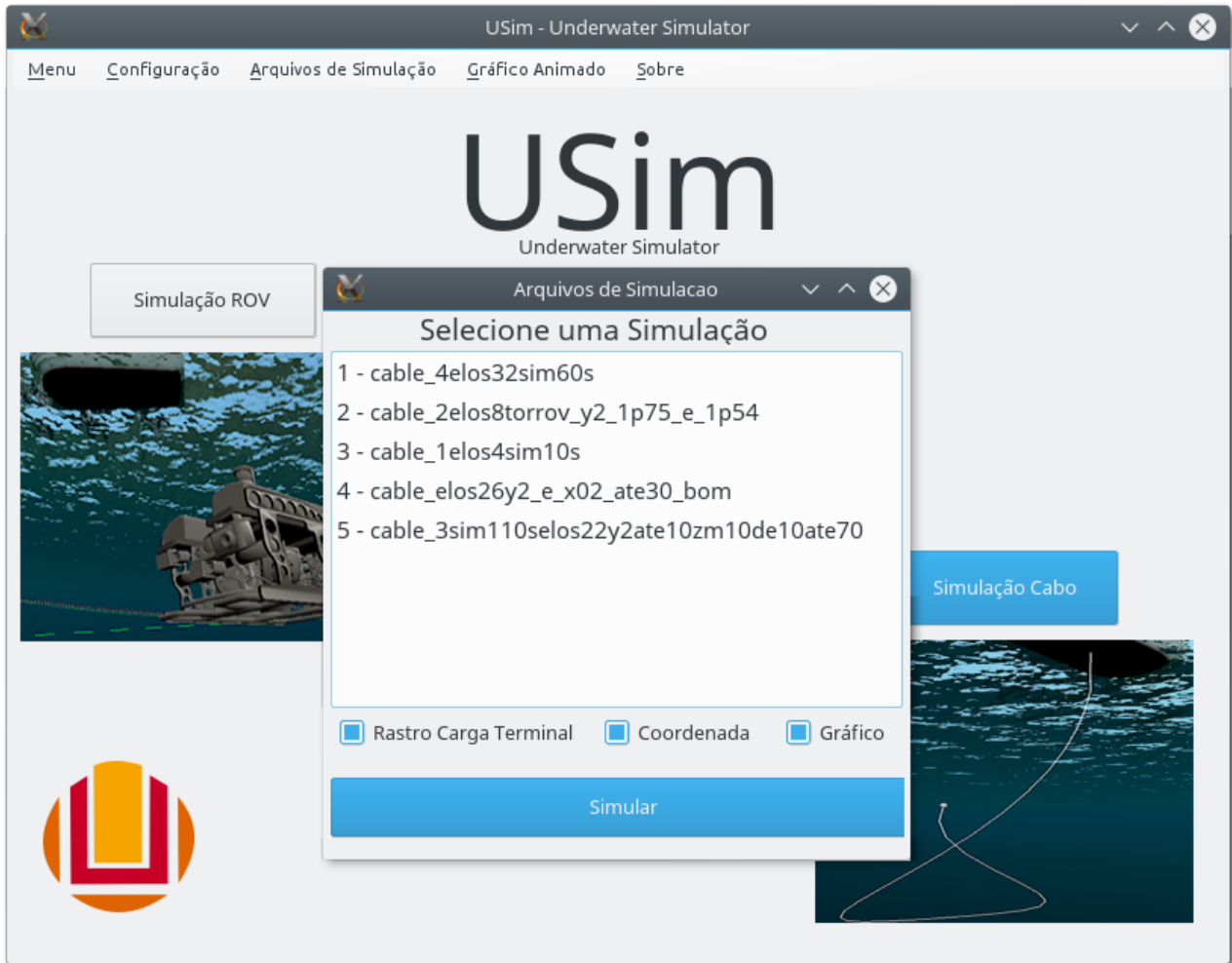


Figura 7.14: Seleção de simulação e configurações de apresentação

desejada a qualquer momento da animação. Uma das câmeras segue o objeto de simulação, enquanto outra é fixada no fundo do ambiente apontando para a superfície. A terceira câmera pode ser movida livremente pelo usuário. A Tabela 5 define as teclas de ação que alternam entre as câmeras disponíveis.

Tabela 5: Teclas seletoras das câmeras

Câmera	Tecla
Livre	Tecla Espaço
Profundidade	Tecla Num 0
Tracker	Tecla Num 3

Alguns atalhos de teclado estão disponíveis para o usuário, a fim de controlar o ritmo da animação. Estes são apresentados na Tabela 6. A captura de telas e a gravação das animações resultam na criação de arquivos em uma pasta específica na árvore de diretórios do USim, que podem ser utilizadas posteriormente pelo usuário.

Tabela 6: Atalhos de teclado disponíveis durante a animação

Teclas de atalhos durante a animação	
Atalho	Ação
Esc	Finalizar animação
Enter	Pausar/Iniciar animação
P	Captura de Tela
R	Gravação de Tela

7.4 Gráfico tridimensional animado e interativo

Além do desenvolvimento do animador em ambiente realista, foi desenvolvido um animador tridimensional interativo em modo gráfico, como uma ferramenta auxiliar a análise e exibição da animação. Este gráfico não contém ambientação, apenas é composto por um ambiente espacial tridimensional com escala em todos os seus eixos, compondo uma cena em grade da escala determinada.

A Figura 7.15 apresenta o gráfico tridimensional, onde percebe-se que ele proporciona uma sensação de cena 3D. Isso ocorre pelo fato da utilização do *toolkit* de plotagem 3D do *matplotlib* (HUNTER, 2007), o *mplot3d*, que adiciona recursos de uma terceira dimensão para gráfico, fornecendo um objeto de eixos que pode criar uma projeção 3D.

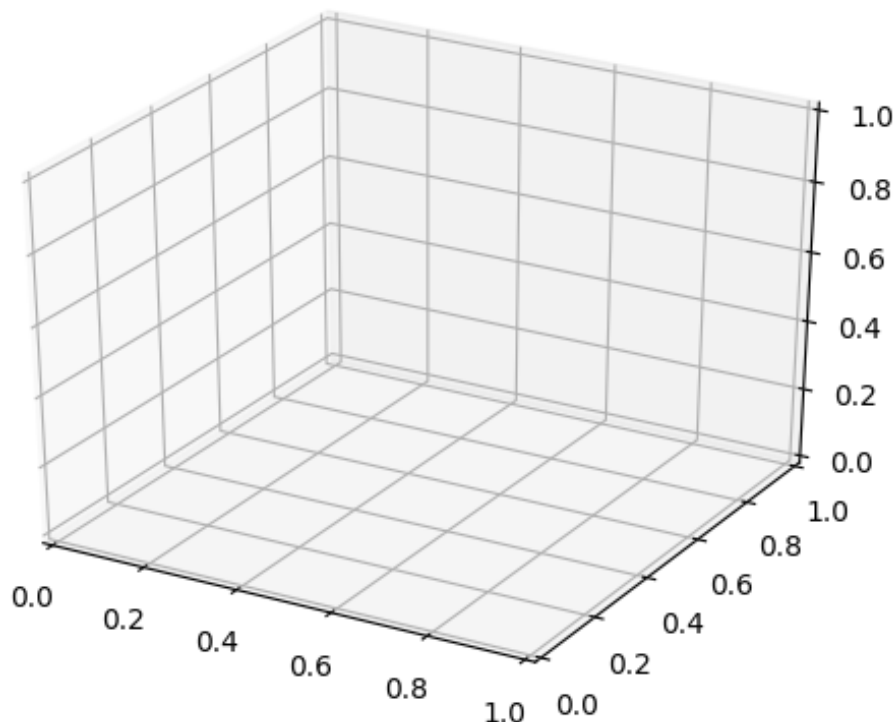


Figura 7.15: Interface do Gráfico tridimensional animado e interativo

Com uma cena tridimensional, o *toolkit* possibilita que seja plotado dados de três dimensões nesta cena, onde é possível determinar a elevação e o azimute em que a projeção é apresentada.

A Figura 7.16 mostra o resultado de plotagem de uma curva paramétrica.

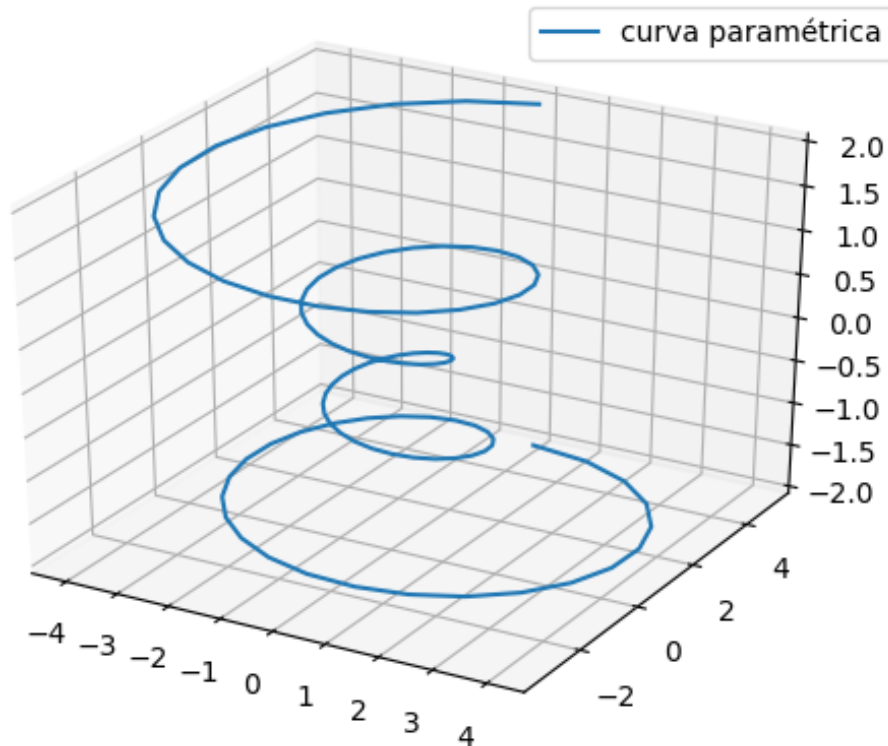


Figura 7.16: Gráfico tridimensional com a plotagem de uma curva paramétrica

Após a definição de uma cena em projeção 3D é possível apresentar alguns resultados das animações do USim, como por exemplo, a trajetória percorrida por um AUV. Esse resultado é obtido na forma de gráfico estático, mas com a possibilidade de apresentar os resultados de animação em tempo de simulação, com a utilização desta cena tridimensional gráfica. Isso tornou-se possível com a utilização da API *animation* que é provida pelo *matplotlib*, e faz com que o gráfico seja representado como um objeto animado. Essa característica ocorre com a utilização da função *FuncAnimation* presente em *animation* e que seguindo um intervalo de tempo pré-determinado, atualiza o conteúdo presente no gráfico.

Do momento inicial de execução do gráfico o seu conteúdo sofre atualizações até que o seu estágio de atualização final seja alcançado. Com isso, torna-se possível apresentar uma animação dentro desta cena.

Afim de possibilitar a interação do usuário com o gráfico, o *matplotlib* proporciona um *backend* interativo para a projeção gráfica, onde oferece a capacidade de girar e ampliar a cena 3D. Pode-se girar a cena 3D simplesmente clicando e arrastando a cena (Figura 7.17). O zoom é feito clicando com o botão direito do mouse na cena e arrastando o mouse para cima e para baixo (Figura 7.18). Outras funções também estão disponíveis, como editar a escala dos eixos, texto do gráfico, e ainda permitindo que o usuário capture imagens do cenário que está sendo apresentado.

Essas características transformam o animador gráfico tridimensional interativo em uma ferramenta relevante a análise dos resultados de simulação, bem como, uma ferramenta auxiliar de animação, contemplando as principais funcionalidades de uma animação tridimensional. O

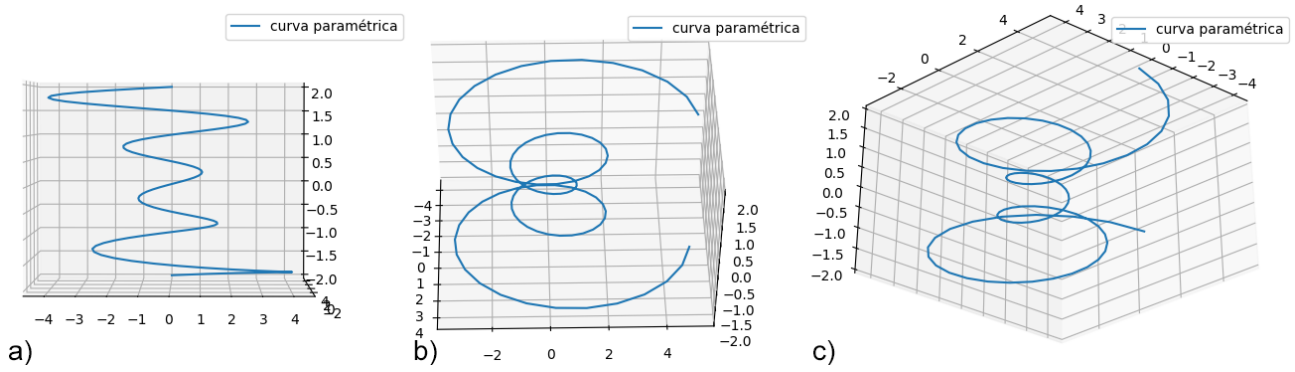


Figura 7.17: Gráfico interativo, diferentes ângulos de cena, por movimento de giro

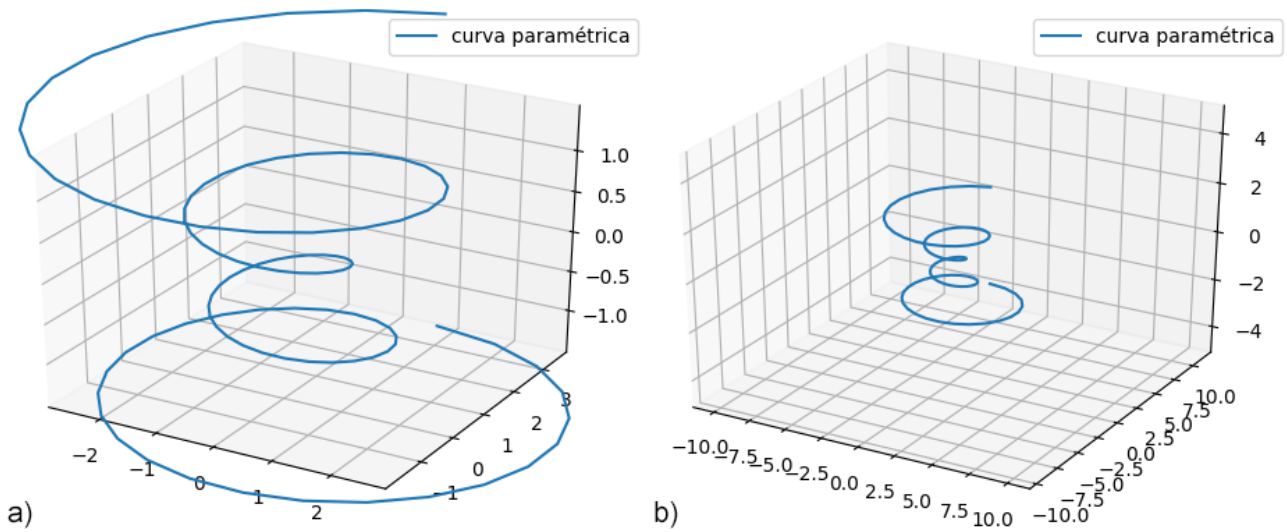


Figura 7.18: Gráfico interativo, ferramenta zoom: a) zoom ampliado; b) zoom diminuído.

Capítulo 8 quando apresenta os resultados, utiliza-se desta ferramenta para demonstrar as animações, bem como, para gerar os gráficos de trajetórias.

Toda simulação animada pelo USim, gera um arquivo de dados que fica disponível ao animador gráfico. Ao usuário é permitido animar a simulação com o uso desta ferramenta, a partir da interface principal do USim. Toda animação exibida, gera um arquivo de vídeo que é armazenado em uma pasta específica presente no contexto do software.

Resultados de simulações para os três cenários de simulação do USim são apresentados e discutidos no próximo Capítulo.

8 RESULTADOS

8.1 Introdução

Neste capítulo são apresentados os resultados de simulações com animações gráficas referentes aos três casos de aplicação considerados: apenas o veículo subaquático sem cabo, seguindo uma trajetória de referência previamente especificada, como resultados de simulação com controle em malha fechada; cabo como estrutura flexível, conectado a uma estrutura fixa em uma de suas extremidades, enquanto na outra extremidade há uma carga terminal livre, como resultado de uma simulação em malha aberta; veículo subaquático considerado como carga terminal na extremidade livre de um cabo.

Os animadores são descritos nos capítulos 6 e 7 e permitem a análise do comportamento de veículo subaquático e estrutura flexível do tipo cabo no espaço tridimensional subaquático. As animações são apresentadas a partir de captura de *frames* com o objetivo de considerar o movimento da simulação durante sua evolução temporal. Em cada uma das simulações é exibido também o seu resultado em gráfico tridimensional interativo, e caso a simulação considere uma carga terminal, a trajetória da carga terminal também é apresentada.

A efetividade do simulador gráfico pode ser avaliada a partir dos casos apresentados a seguir. Todas as animações ocorrem no mesmo mundo sintético, apenas diferindo nos elementos do sistema apresentados. O cenário tem seu sistema de coordenadas de referência inercial identificado por um navio. A origem do sistema de referência se situa no centro do plano horizontal do navio que tangencia a superfície da água. As coordenadas X , Y e Z são identificadas pelas cores verde, vermelho e azul, respectivamente. Observa-se que o eixo z neste cenário é orientado na direção do leito subaquático, ou seja, apontado para o centro da Terra. A Figura 8.1 apresenta o sistema de coordenadas já no cenário virtual.

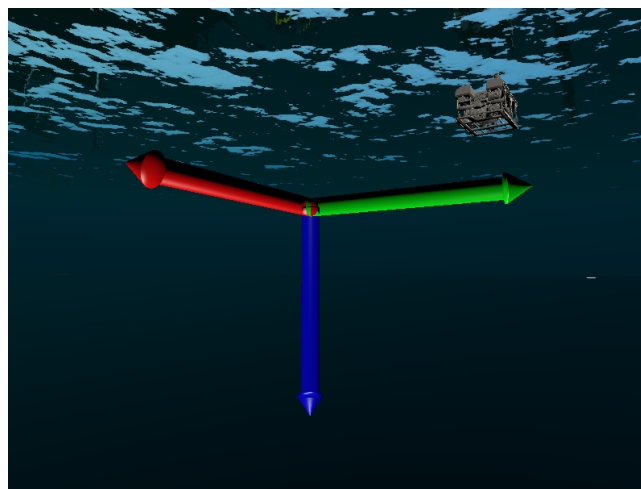


Figura 8.1: Sistema de coordenadas de referência

As simulações foram realizadas utilizando-se um computador com processador Intel Core i5-2450M de 2ª geração, 2.50 Ghz, 8GB de memória SDRAM DDR3 e placa de vídeo GeForce

GT 540M com 1024 MB de memória DDR3.

8.2 Simulação com o Veículo Autônomo (AUV)

No caso de simulações de veículos subaquáticos a animação consiste em movimentar o AUV ao longo da trajetória de referência especificada no espaço tridimensional, trajetória esta que fornece a posição do centro de massa do veículo e a sua orientação em ângulos de Euler para cada instante de tempo. A posição real do veículo é obtida da simulação numérica. O software desenvolvido mostra estas duas trajetórias enquanto realiza a animação em três dimensões do movimento do veículo, permitindo assim ao usuário ter uma boa noção sobre o desempenho do controle projetado e simulado. Além disso, gera-se um rastro de trajetória a partir dos dados de posição do veículo anteriores ao atualmente sendo renderizado, possibilitando ao usuário acompanhar a evolução do movimento do AUV. Para sistemas em malha aberta, apenas a curva de trajetória do veículo é apresentada.

Para ilustrar os resultados desse tipo de animação, executou-se uma simulação numérica considerando-se os modelos cinemático e dinâmico do NEROV (apresentado no capítulo 3). Em uma das simulações, foi considerada uma corrente subaquática na direção x de $2,5m/s$, inserindo-se assim um maior nível de dificuldade para o controle projetado manter um bom desempenho. Definiu-se uma trajetória de referência em espiral, ou seja, o veículo deveria descer em direção ao fundo do mar percorrendo uma trajetória espacial em espiral. A simulação numérica foi realizada com um passo de integração de $0.05s$, durante $80s$. Foi utilizado um controle do tipo proporcional e derivativo (PD) com linearização por realimentação, conforme definido no capítulo 3.2.

A Figura 8.2 mostra a trajetória espacial do centro de massa do veículo (em vermelho), bem como a trajetória espacial de referência (em azul). Neste caso, considerou-se a corrente subaquática. Percebe-se que o controle não foi eficiente no seguimento da referência em razão da perturbação dinâmica exercida pela corrente oceânica.

Já a Figura 8.3 mostra a mesma trajetória, agora sem considerar a corrente subaquática. Neste caso percebe-se que o controle atua com eficiência no seguimento da trajetória do veículo, uma vez que não ocorre perturbação dinâmica sendo exercida em seu corpo.

As Figuras 8.4 e 8.5 correspondem a renderizações dos instantes inicial e final da simulação no ambiente virtual do USim. Cada figura é renderizada por câmeras sintéticas em diferentes posturas.

Para se ter uma melhor compreensão da simulação e dos resultados apresentados, a Figura 8.6 apresenta o movimento do AUV desde o instante inicial $t = 0s$ até o instante final $t = 80s$, obtido através do gráfico tridimensional animado presente no USim, como uma ferramenta auxiliar à visualização da animação.

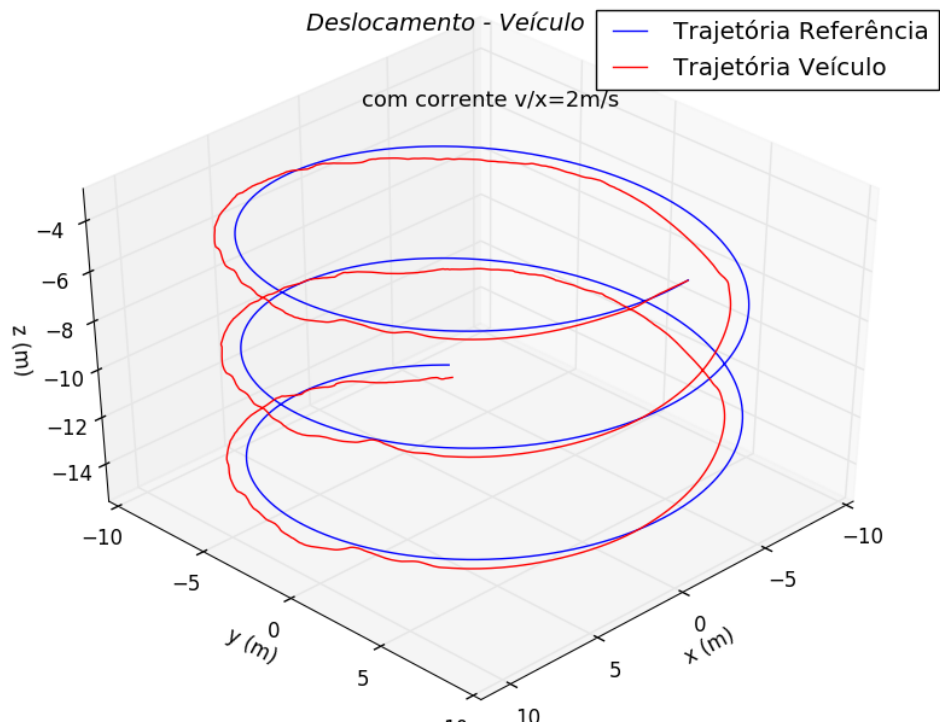


Figura 8.2: Resultado da simulação do NEROV em malha fechada para uma trajetória em espiral

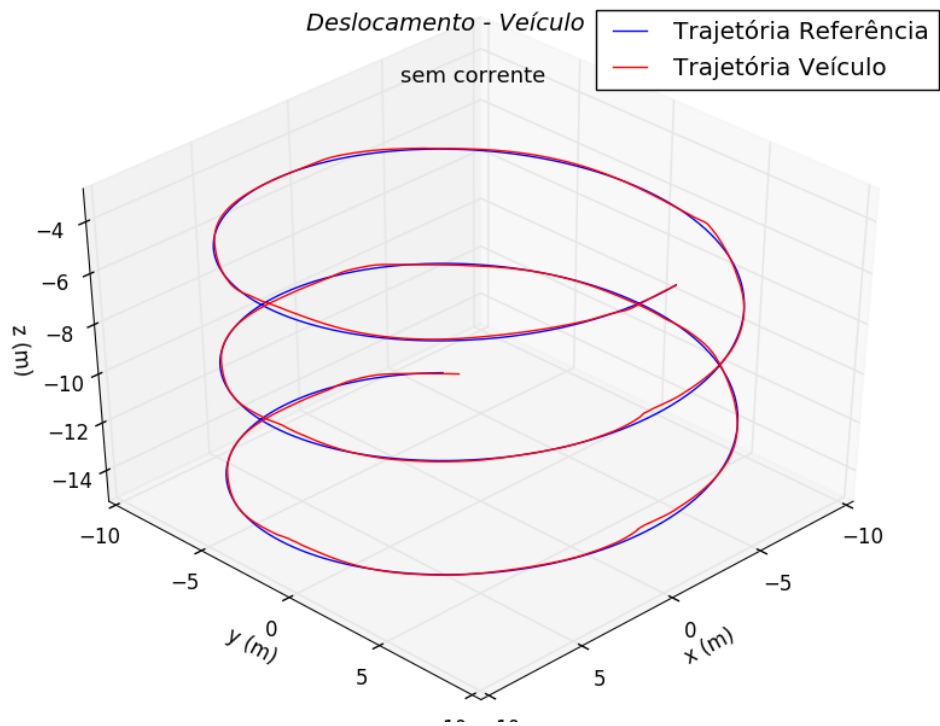


Figura 8.3: Resultado da simulação do NEROV em malha fechada para uma trajetória em espiral, sem considerar perturbação dinâmica

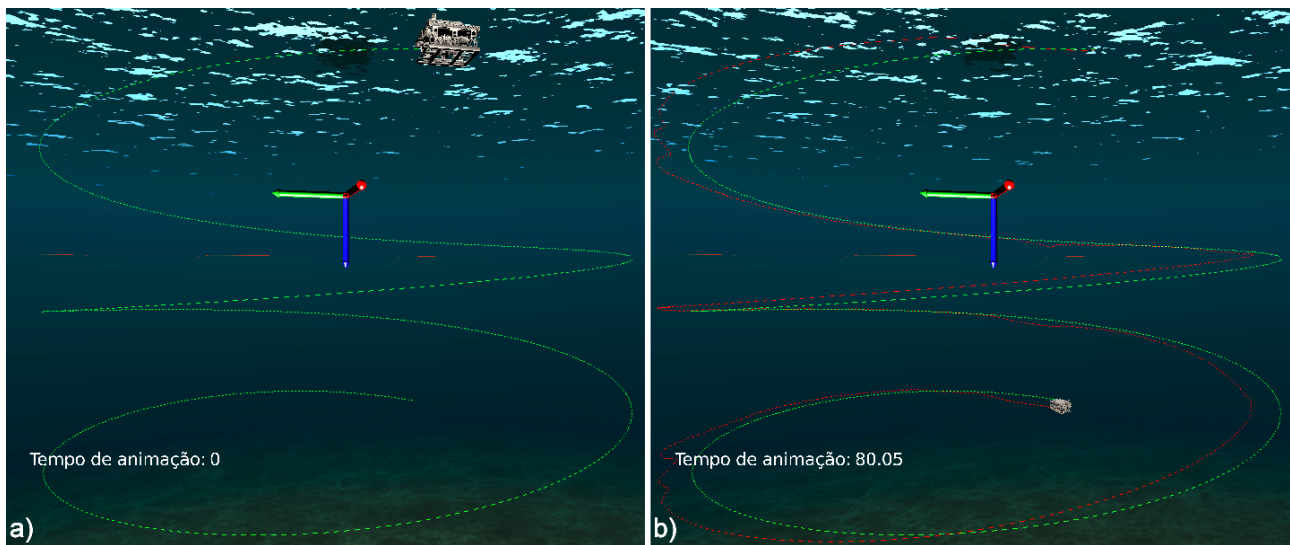


Figura 8.4: a) Instante inicial; b) Instante final; animação do NEROV, renderizados por uma câmera com plano de visão alinhado com um plano vertical.

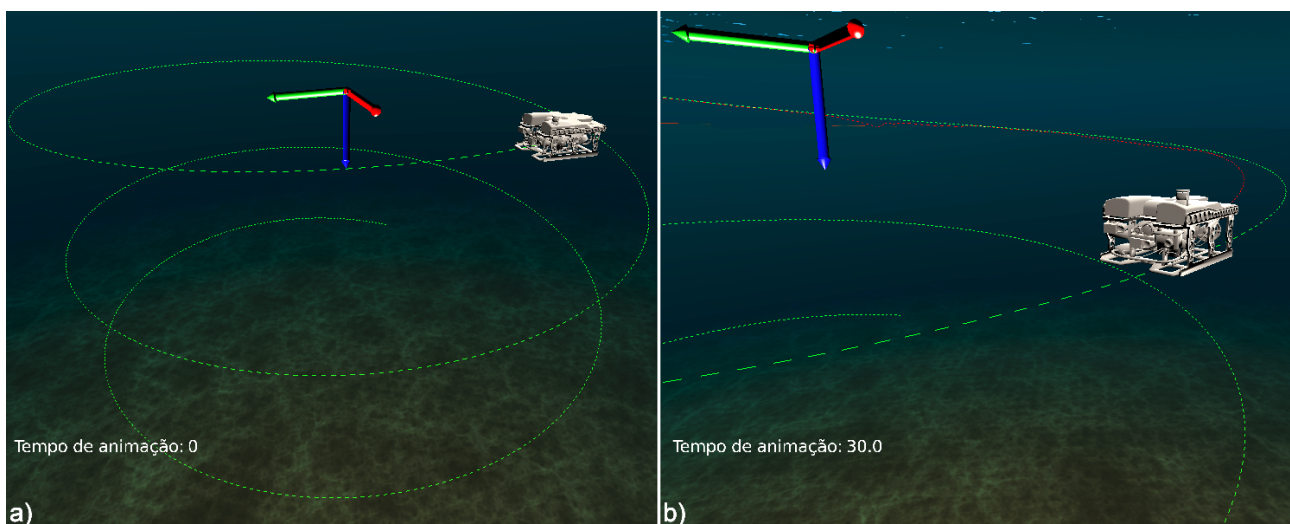


Figura 8.5: Animação do NEROV em diferentes instantes de animação renderizados por uma câmera em duas perspectivas

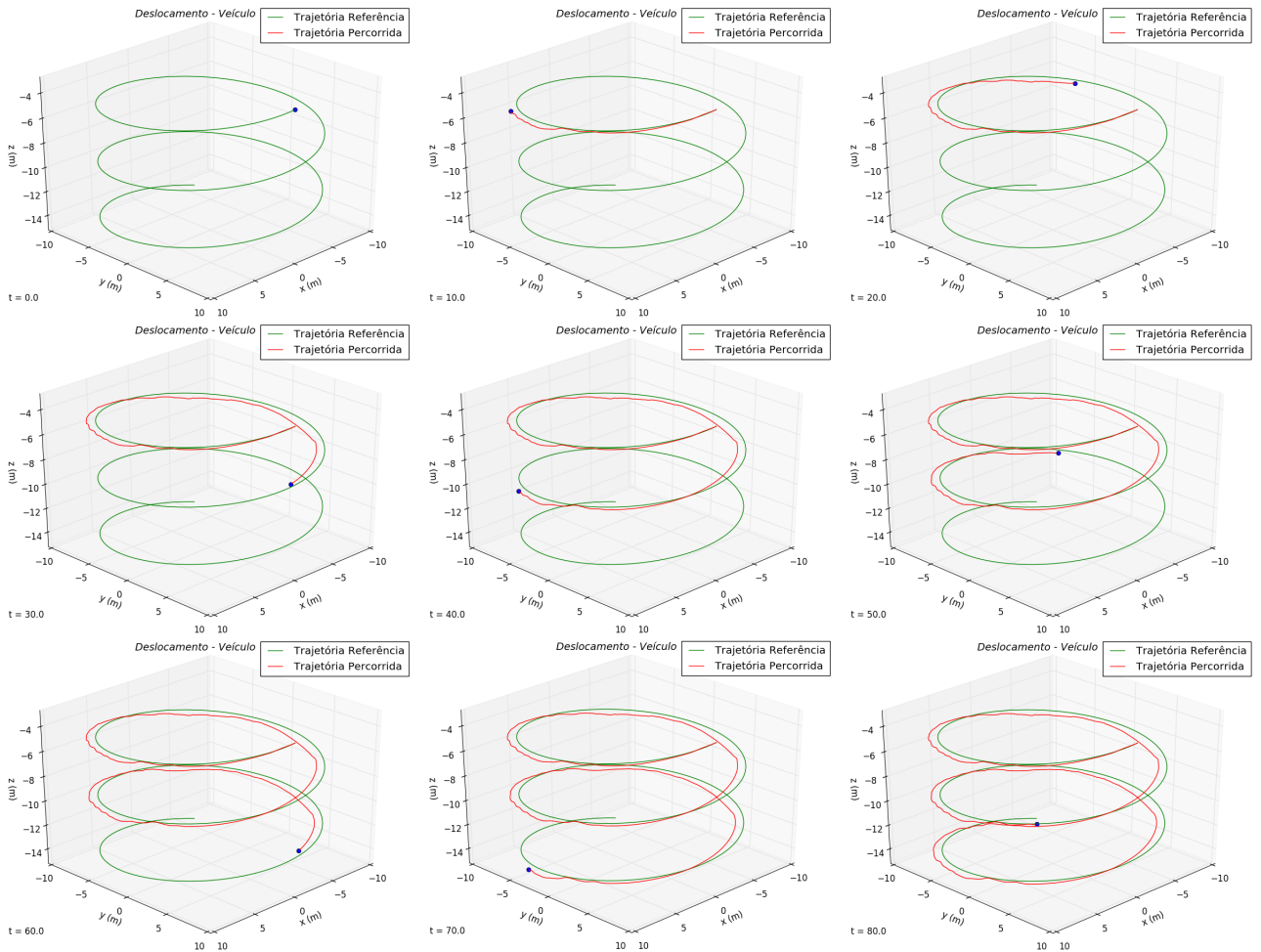


Figura 8.6: Movimentação do veículo, representada a partir do gráfico tridimensional animado, com frequência amostral de 10s.

8.3 Simulação de Estrutura Flexível do Tipo Cabo

Em simulações de cabos, a animação consiste em renderizar a estrutura flexível como um conjunto de elos cilíndricos de mesmo comprimento, conectados entre si. A orientação de cada elo é definida pelos resultados da simulação numérica, com ângulos de azimute, elevação e torção. As posições de um elo i são calculadas em função da postura de seus antecessores.

A simulação consiste em um cabo conectado a uma estrutura fixa em uma de suas extremidades, enquanto na outra extremidade há uma carga terminal livre. Portanto, caso se utilize uma carga terminal na simulação, esta é representada por um esfera solidária ao último elo virtual do cabo. É possível visualizar o movimento dessa carga a partir de uma curva traçada no cenário virtual, criada a partir das posições dessa carga a cada instante da animação.

O estudo de caso para dinâmica de cabos utiliza os resultados de uma simulação numérica de um cabo de $1200m$ de comprimento, com uma massa específica de $7850(kg/m^3)$. A carga terminal considerada possui massa de $600kg$. O cabo é discretizado em 32 elos de mesmo comprimento. A simulação está sob efeito de arrasto hidrodinâmico, como cabo submerso, partindo do repouso e de uma posição especial determinada, possibilitando assim uma situação de queda livre.

No início da simulação, a carga está em uma profundidade de $200m$, e é considerada em queda livre durante o tempo de $60s$ da simulação. O passo utilizado foi de $0.005s$. A Figura 8.7 apresenta duas renderizações dessa simulação. Na da esquerda (Figura 8.7a), em instante inicial ($t = 0.5s$), a câmera está posicionada mais próximo ao espaço em que o cabo se movimenta, sendo possível observar a esfera que representa a carga terminal. Na renderização da direita, (Figura 8.7b), em instante final ($t = 60s$), é possível observar o movimento da carga terminal por uma linha pontilhada, a partir de um ponto de vista mais afastado do espaço da simulação.

A Figura 8.8 apresenta o movimento do cabo e da carga terminal com *frames* mostrados a cada $5s$. O cabo está em queda livre e animação possibilita mostrar uma boa sensação de realidade física, indicando que o resultado da simulação está coerente com o fisicamente esperado.

A figura 8.9 mostra a trajetória espacial do centro de massa da carga terminal no espaço tridimensional. O triângulo (verde) corresponde ao início da animação e o triângulo invertido (vermelho) corresponde ao final da animação.

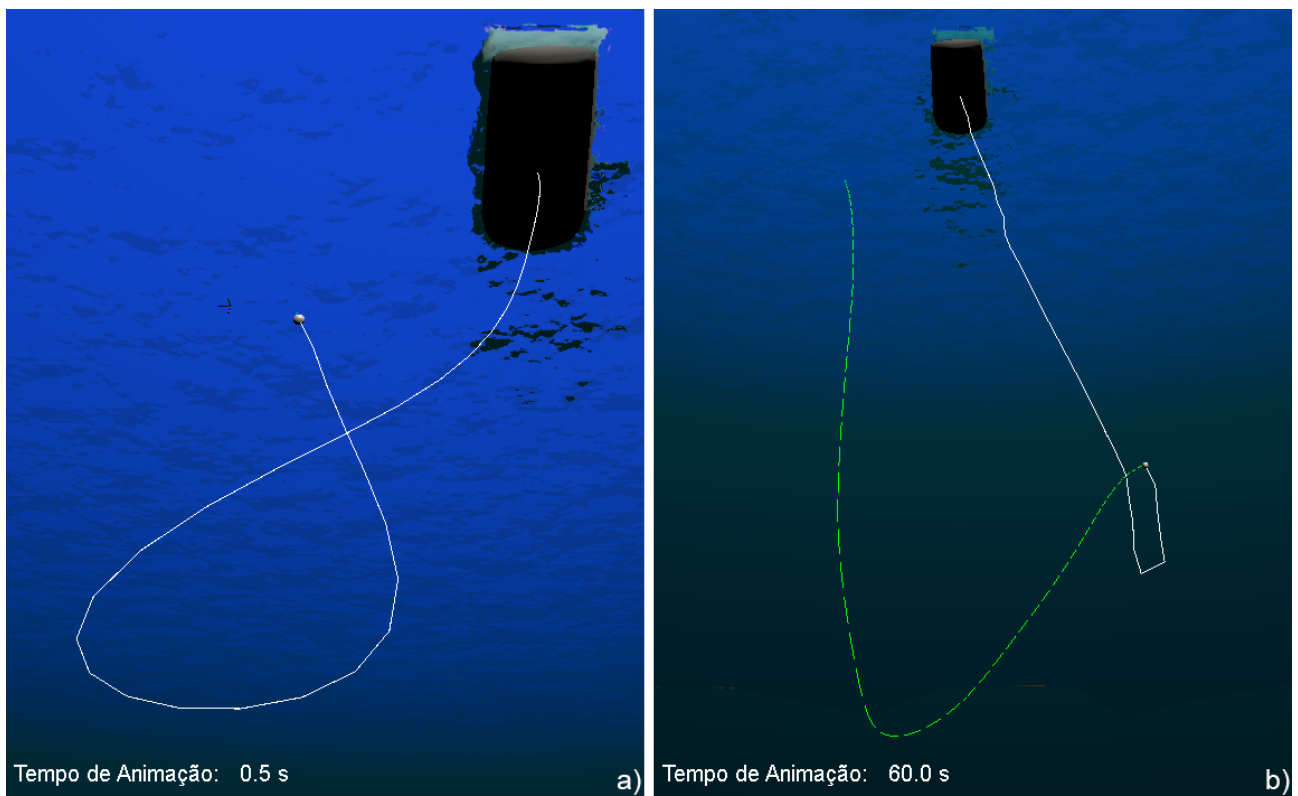


Figura 8.7: Renderizações de instantes de uma simulação de estrutura flexível do tipo cabo com uma carga terminal: a) Instante inicial. b) Instante final

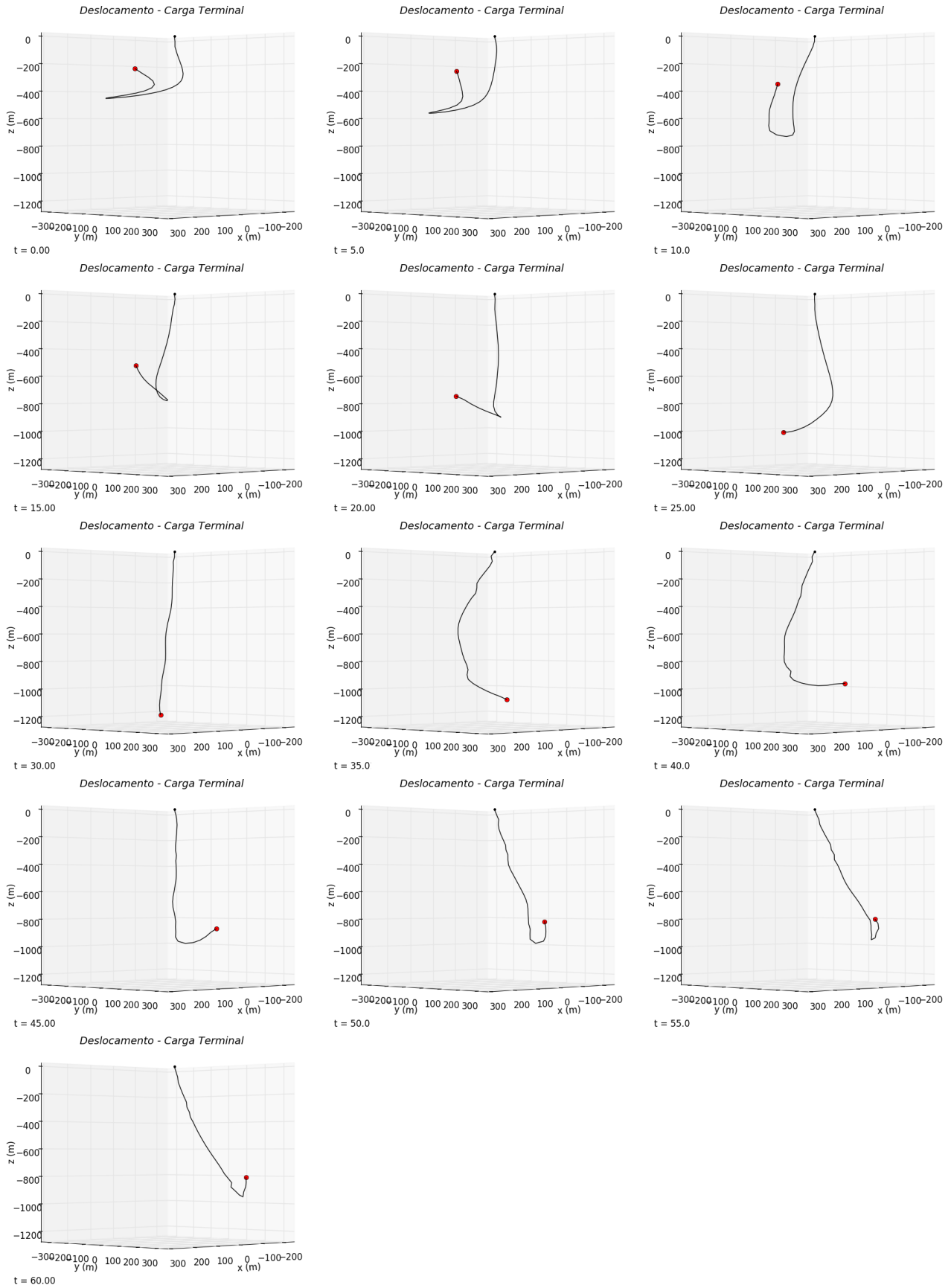


Figura 8.8: Simulação de estrutura flexível do tipo cabo com uma carga terminal em queda livre durante 60s com *frames* a cada 5s

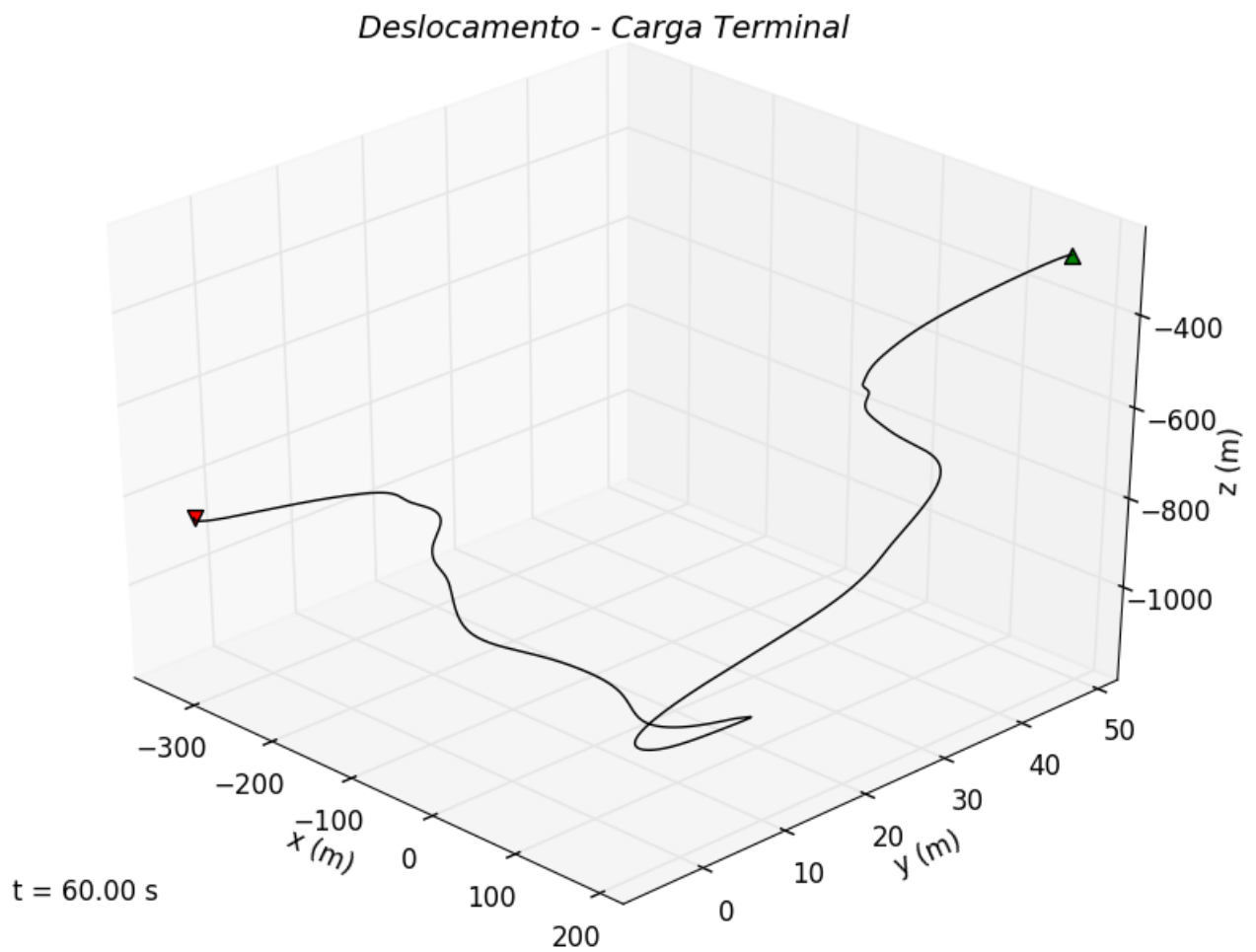


Figura 8.9: Trajetória espacial a partir do centro de massa da carga terminal durante a animação. Início da animação (verde), fim da animação (vermelho)

8.4 Simulação de um ROV com o seu Cabo Umbilical

Esse tipo de simulação é muito similar à simulação de estruturas do tipo cabo com carga terminal. A diferença é que a esfera utilizada para representar a carga é substituída pelo modelo do ROV, que é considerado solidário ao último elo, considerando-se assim as forças e torques aplicadas na carga terminal. Essas forças também geram torques na própria carga, e como estão no referencial do corpo, é como se aplicadas por atuadores de um ROV, modelo descrito no capítulo 4.2.

A simulação utilizada para ilustrar o tipo de animação desse caso consiste no veículo posicionado inicialmente a $300m$ de profundidade, conectado ao navio por um cabo de umbilical. O cabo é discretizado em 16 elos de mesmo comprimento ($18.75m$). A simulação teve duração de $120s$, com passo de integração igual a $0.0005s$.

A Figura 8.10 apresenta três instantes da simulação. A renderização da esquerda 8.10a) corresponde ao instante inicial, enquanto a central 8.10b), ao instante $15s$. A renderização da direita 8.10c) é no instante $30s$.

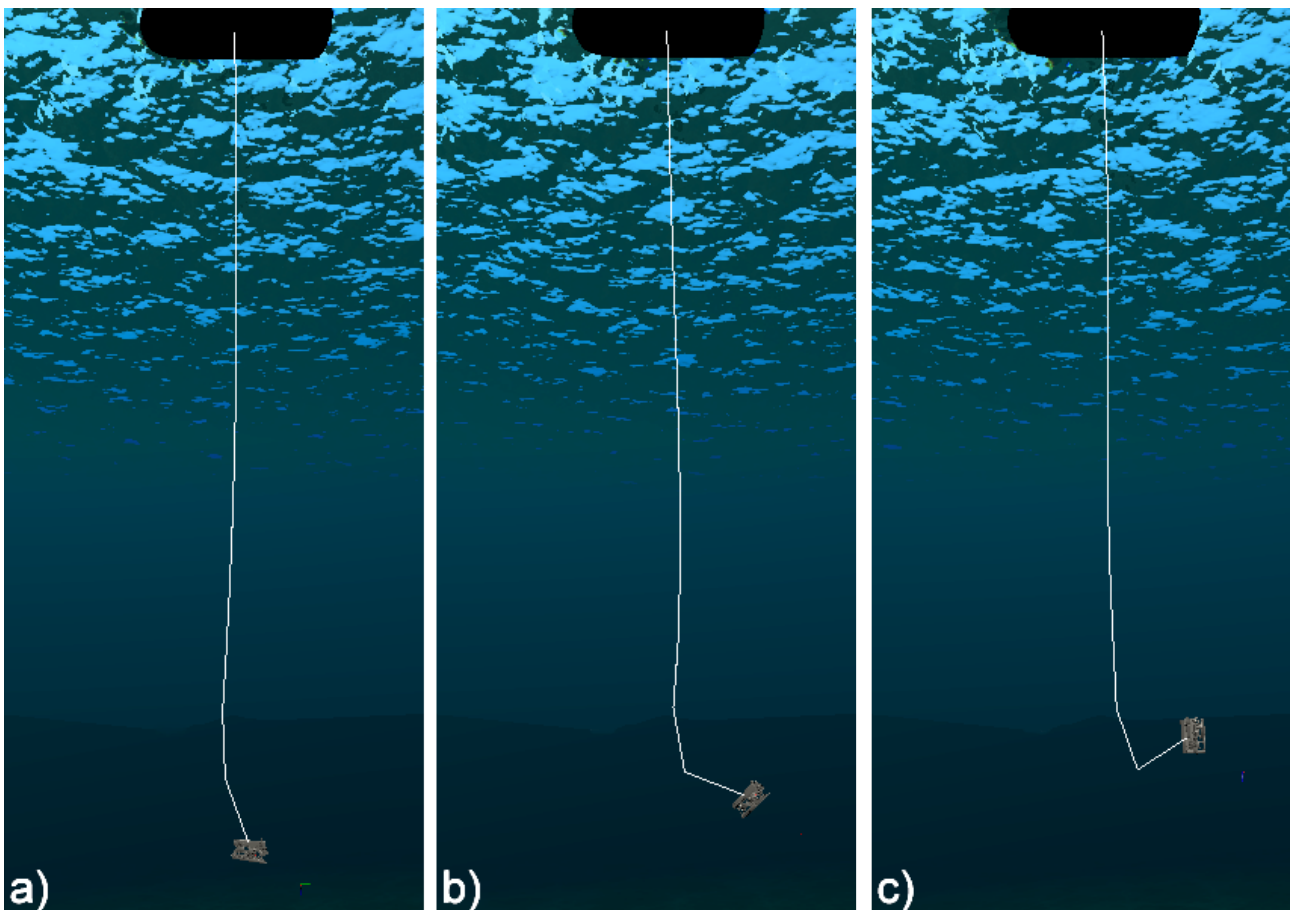


Figura 8.10: Renderizações da dinâmica do veículo com cabo umbilical nos instantes $0s$ a), $15s$ b) e $30s$ c).

Na Figura 8.11, tem-se o movimento do Cabo com interações com o veículo, com *frames* mostrados a cada $10s$.

Inicialmente, forças foram aplicadas ao ROV na direção inercial Y durante os primeiros 10s. Após estes segundos iniciais, forças são aplicadas na direção inercial vertical, fazendo com que o veículo suba em direção à superfície d'água. Após 80s, os atuadores são desligados e o movimento do veículo passa a ser em queda livre a partir de então, embora sob a ação das forças hidrodinâmicas.

Já a Figura 8.12 mostra essa mesma animação em ambiente tridimensional do USim compreendendo diferentes momentos de animação. A primeira captura da animação corresponde ao instante 65s e prossegue a diferentes instantes até o instante 110s ser alcançado. Com essa visualização, pode-se perceber o movimento de rotação do veículo, resultando também na torção do cabo.

A Figura 8.13 mostra a trajetória espacial do veículo descrita pelo seu centro de massa. O triângulo (verde) corresponde ao início da animação e o triângulo invertido (vermelho) corresponde ao final da animação.

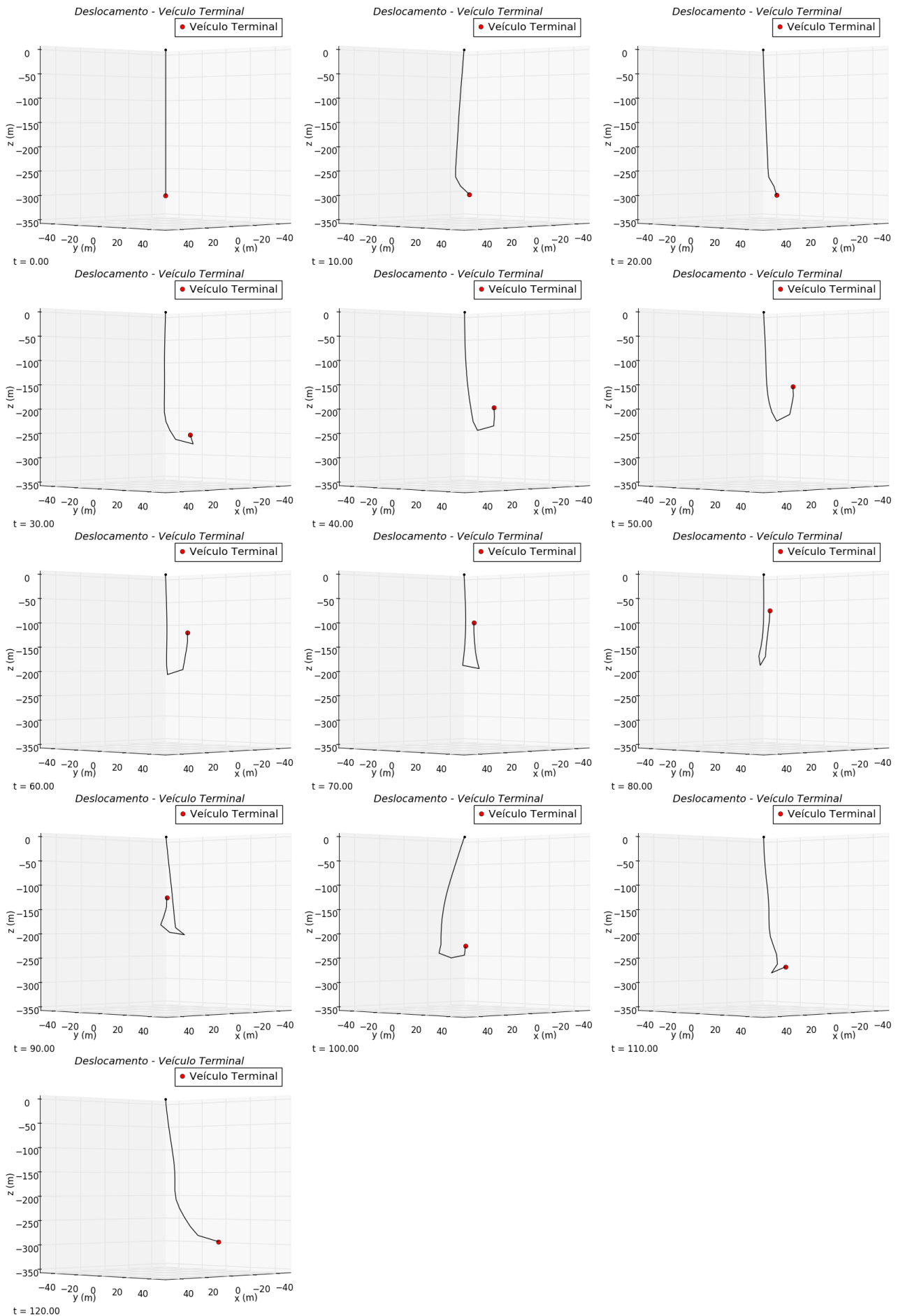


Figura 8.11: Animação gráfica tridimensional da dinâmica do veículo com o cabo umbilical, frames a cada 10s

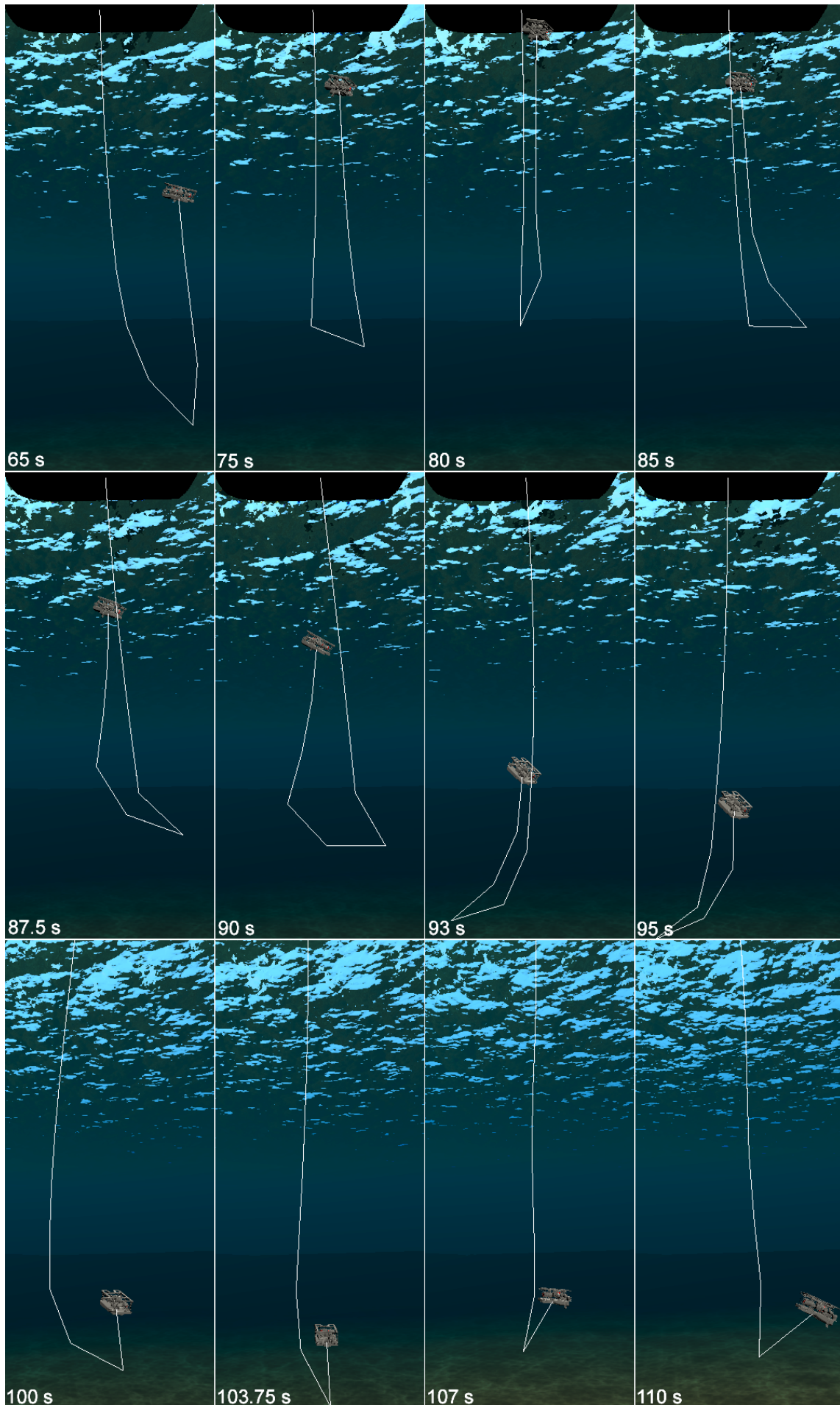


Figura 8.12: Animação tridimensional da dinâmica do veículo e do seu cabo umbilical.

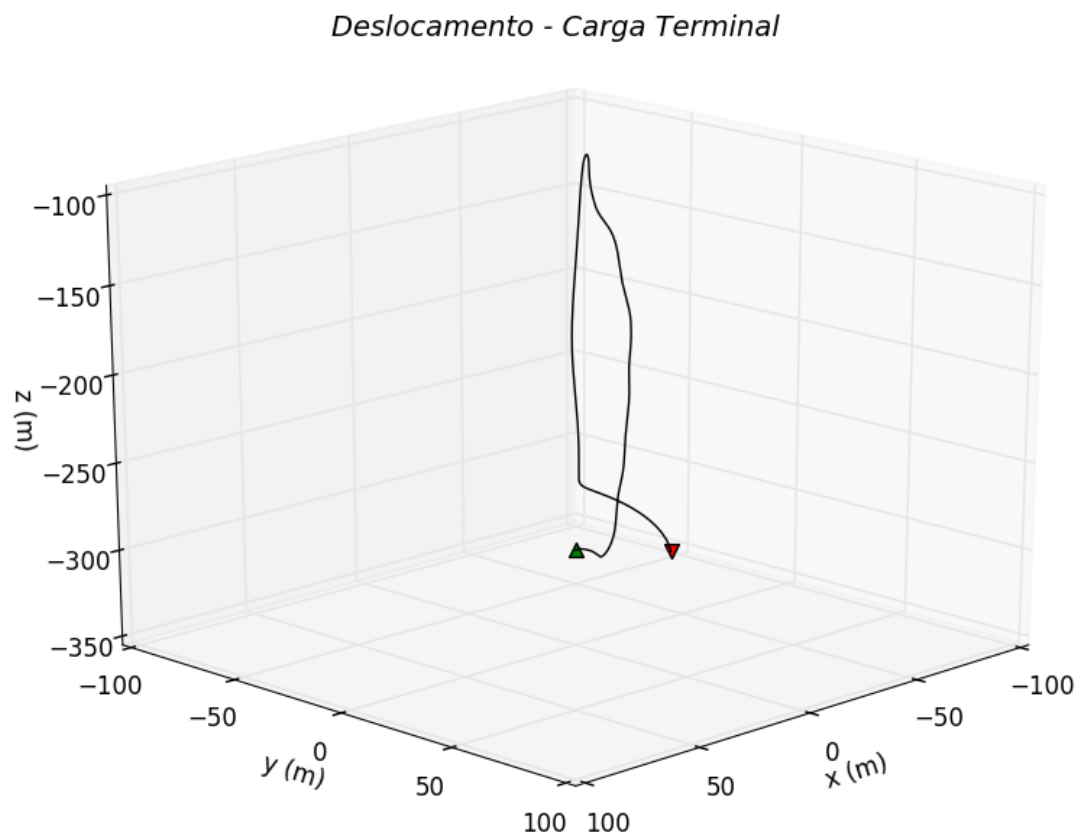


Figura 8.13: Trajetória espacial do centro de massa do veículo durante a animação (início da animação (verde), fim da animação (vermelho)).

8.5 Considerações Finais

O software consegue gerar renderizações de cenários virtuais para os diferentes tipos de simulações com um grau de realismo que depende apenas da qualidade da modelagem geométrica dos elementos da simulação.

O desempenho das animações depende do hardware gráfico empregado para a renderização. Quando o passo da simulação é pequeno, o grande número de instantes resultantes afeta a animação, que para poder exibir cenas a cada instante pode acabar levando um tempo maior do que o tempo de simulação. Por esse motivo existe a definição de passo de animação, que é uma medida de quantos instantes são efetivamente renderizados, de forma que a animação leve o mesmo tempo da simulação. Um procedimento de calibração é utilizado para definir esse passo de animação, calculado como a relação entre o tempo da simulação e o tempo da animação considerando todos os instantes. Esse fator estabelece que é aceitável uma perda de informações de renderização a fim de que esta ocorra no tempo da simulação.

Com base nas simulações apresentadas, é possível concluir que o software cumpre os objetivos inicialmente propostos no projeto.

9 CONCLUSÕES

Neste capítulo apresentam-se as conclusões finais sobre os resultados obtidos na presente dissertação. Diversas observações sobre cada etapa do trabalho desenvolvido já foram apresentadas ao longo dos capítulos. Entretanto, as principais conclusões verificadas em toda a dissertação são resumidas no presente capítulo, complementadas por comentários que possibilitem uma melhor interpretação dos resultados obtidos.

Os trabalhos relativos às modelagens cinemática e dinâmica contidos na presente dissertação foram desenvolvidos em dissertações anteriores, realizadas no Núcleo de Matemática Aplicada e Controle (IMEF-FURG), nos programas de pós-graduação em Modelagem Computacional e Engenharia Oceânica, da FURG. Tais modelos referem-se a estruturas flexíveis do tipo cabo e ainda, a veículos subaquáticos não tripulados, tendo sido amplamente validados e testados em trabalhos anteriores, de forma que vieram compor a base matemática do presente trabalho.

Conforme demonstrado no capítulo 1, a indústria *offshore* voltada à exploração de petróleo e gás em ambiente subaquático vem crescendo muito nos últimos anos, despertando interesses de estudos e pesquisas visando o seu constante desenvolvimento. Pesquisas envolvendo estruturas flexíveis do tipo cabo estão mais focadas em análises estáticas, visando o dimensionamento de tais estruturas, conhecendo-se os esforços a que elas estarão submetidas no ambiente subaquático. Entretanto, quando se considera análises dinâmicas no espaço tridimensional, a bibliografia existente na literatura não é tão abundante, principalmente em razão da complexidade inerente ao desenvolvimento de modelos dinâmicos no espaço tridimensional. Além da complexidade natural do problema de modelagem, há dificuldades inclusive para visualizar os resultados de simulações, de forma a permitir uma análise mais consciente dos resultados. É exatamente nesse contexto que se insere o foco da presente dissertação, que envolveu a realização de estudos visando o desenvolvimento de um simulador que permite visualizar os resultados de simulações a partir de animações em três dimensões, fundamental para a análise do comportamento dinâmico de tais estruturas. O simulador ainda permite a visualização de animações com veículos subaquáticos não tripulados, autônomos, ou seja, sem cabos conectando os veículos à superfície e, neste contexto, podem ser testadas leis de controle para veículos subaquáticos, com uma excelente estrutura de visualização dos resultados de simulações. O simulador desenvolvido contém ainda um módulo que permite unificar o veículo subaquático e o seu cabo umbilical, provendo animações que permitem aferir as interações dinâmicas entre cabo e veículo. Em resumo, o simulador desenvolvido na presente dissertação permite a visualização de resultados a partir de animações tridimensionais em três situações possíveis: estruturas flexíveis do tipo cabo em diversas situações de aplicações subaquáticas; veículos subaquáticos autônomos em movimento controlado no espaço tridimensional; junção dinâmica entre veículo e cabo umbilical, no caso de veículos subaquáticos controlados remotamente. O desenvolvimento do simulador, bem como o estudo necessário à viabilização desse desenvolvimento, pretendem ser as maiores contribuições da presente dissertação.

O simulador desenvolvido foi, portanto, o principal objetivo e foco a ser perseguido no trabalho que resultou na presente dissertação. A concretização desse objetivo só foi possível graças a estudos realizados, divididos nos seguintes módulos:

- formalismo discreto para a modelagem dinâmica de estruturas flexíveis do tipo cabo;
- modelagem cinemática de veículos subaquáticos a partir da utilização dos ângulos de Euler;
- modelagem dinâmica de veículos subaquáticos;
- controle de veículos subaquáticos (PID com linearização por realimentação);
- junção das dinâmicas do veículo subaquático e do cabo umbilical.

O estudo realizado envolvendo a modelagem dinâmica de veículos subaquáticos mostrou que há uma grande dificuldade na identificação dos parâmetros do modelo, que dependem muito da forma geométrica do veículo, disposição dos atuadores, etc. Para testar o simulador desenvolvido foi utilizado o modelo dinâmico do veículo NEROV, bastante conhecido na literatura de robótica subaquática. A principal vantagem desse veículo é que ele possui atuadores nos três planos de simetria e é controlado nos seus seis graus de liberdade. Esse modelo dinâmico já foi amplamente testado em trabalhos anteriores, de forma que se mostrou ideal para ser utilizado no simulador desenvolvido na presente dissertação.

Para simular a dinâmica do veículo subaquático sem a presença do cabo umbilical foram disponibilizados no simulador desenvolvido dois módulos, consistindo em resultados de malha aberta e de malha fechada. No caso da malha aberta, não há trajetórias de referência a seguir, apenas tensões são aplicadas aos atuadores e estes geram empuxos que movimentam o veículo. O simulador mostra o deslocamento do veículo no espaço tridimensional e ainda, mostra uma curva relativa ao deslocamento do centro de massa do veículo, numa espécie de rastro deixado pelo veículo na sua trajetória espacial. Quando utilizado no módulo malha fechada, o simulador mostra a trajetória de referência no espaço tridimensional, além de mostrar o deslocamento do centro de massa do veículo durante a animação do seu movimento. Esta forma de visualização dos resultados da simulação é essencial para aferir o desempenho do controle, pois além de mostrar a animação tridimensional do veículo como um corpo sólido, mostra as curvas espaciais de referência e de saída (centro de massa), dando uma boa noção sobre os erros cometidos no controle do veículo. Considera-se que essa é uma das contribuições importantes da presente dissertação.

Para realizar o controle em malha fechada optou-se por usar um PD (proporcional e derivativo), em conjunto com a técnica de linearização por realimentação. Essa técnica de controle consegue realizar o desacoplamento dinâmico entre os seis graus de liberdade do sistema, de forma a permitir o projeto individual de controle para cada atuador, de forma independente. Porém, a ausência da componente integral na lei de controle a torna menos robusta a perturbações externas. Foram consideradas duas situações distintas, com e sem a presença de uma

corrente oceânica de $2,5m/s$ como fonte de perturbação dinâmica. A trajetória de referência escolhida foi em espiral, com a referência da posição angular em torno do eixo Z (direção vertical) tangenciando a trajetória em espiral de referência para o centro de massa do veículo. O simulador consegue mostrar de forma bastante eficaz os resultados, indicando a degradação do desempenho do controle quando se considera a perturbação dinâmica da corrente subaquática.

No caso da dinâmica de cabos, o simulador permite testar dois casos distintos: apenas o cabo fixo a uma plataforma em uma das extremidades e livre na outra; cabo fixo a uma plataforma em uma das extremidades e com um veículo subaquático do tipo ROV na extremidade livre. A complexidade dos movimentos envolvidos nos vários graus de liberdade é muito alta, de forma que, sem o simulador desenvolvido, ficaria muito difícil interpretar os resultados, ou seja, a visualização de animações tridimensionais é essencial para a análise dos resultados das simulações numéricas. Deve ser remarcado que o formalismo de modelagem dinâmica utiliza vários elos rígidos conectados por articulações fictícias elásticas que permitem, para cada articulação, movimentos em três graus de liberdade: elevação, azimute e torção. Trata-se, portanto, de uma longa cadeia de corpos rígidos poliarticulados representando o cabo, de forma que ficaria muito difícil interpretar os resultados de simulações sem as animações tridimensionais que o simulador desenvolvido possibilita.

Especificamente sobre o software para a animação tridimensional, inicialmente elaborou-se um breve resumo sobre os fundamentos da modelagem e animação gráfica, no qual os principais conceitos deste tipo de modelagem são apresentados. A partir destes conceitos é possível criar objetos e cenários que permitem realizar uma animação gráfica em três dimensões. Posteriormente aborda-se o projeto e as escolhas para o desenvolvimento do simulador bem como os aspectos do seu desenvolvimento. O simulador gráfico desenvolvido contém a construção de um cenário tridimensional para analisar o comportamento dinâmico de veículos subaquáticos não tripulados e estruturas flexíveis do tipo cabo, com base em animações realizadas em um ambiente virtual que busca uma aproximação com o mundo real. O projeto e escolhas de ferramentas de desenvolvimento buscou satisfazer requisitos que visam a disponibilização do simulador à comunidade acadêmica de forma que esta não apenas possa utilizá-lo, mas estendê-lo, contribuindo para a sua evolução a partir do acréscimo de novos modelos de veículos, implementação de diferentes controladores e ferramentas para a definição de tarefas a serem realizadas pelos ROVs. Além das animações em três dimensões, gráficos animados mostrando as trajetórias de referência e de saída (centro de massa do veículo) foram desenvolvidas e anexadas como ferramenta auxiliar ao simulador USim, permitindo uma melhor observação do desempenho de leis de controle.

Em todas as situações utilizadas como teste, tanto na dinâmica de cabos quanto na de veículos subaquáticos, os resultados obtidos mostraram-se conforme o esperado fisicamente e, para se chegar a essa conclusão, o simulador desenvolvido foi fundamental por permitir uma melhor visualização do comportamento dinâmico dos sistemas físicos testados.

O trabalho realizado na presente dissertação se encaixa em uma linha de pesquisa do Laboratório de Matemática Aplicada e Controle da Universidade Federal do Rio Grande, que trata

da dinâmica de sistemas subaquáticos. Os trabalhos anteriores fundamentaram e motivaram o desenvolvimento do presente simulador.

A principal contribuição desta dissertação consiste no desenvolvimento do simulador gráfico USim. Além de trabalhar com os veículos, a ferramenta é importante para o estudo de cabos de ancoragem, umbilicais e *risers*, muito utilizados na exploração de petróleo *offshore*, por exemplo. A animação da interação entre o ROV e cabo umbilical é uma inovação deste software.

Considera-se que uma contribuição indireta é o estudo de caso de um *engine* gráfico livre e disponível (Blender) e sua integração a um software de simulação, que pode estimular o uso em outros trabalhos.

Em termos de trabalhos futuros derivados da presente dissertação pode ser citado a extensão a outros modelos de ROVs, além do NEROV, uma vez que o USim foi projetado com a ideia de ser extensível e modular. Outra demanda importante seria a integração dos algoritmos de simulação numérica ao software, unificando os desenvolvimentos em um produto. Por fim, evoluções da interface com o usuário para configurar as simulações, definir tarefas e simular a operação remota de ROVs são possibilidades que podem advir deste trabalho.

Referências

- AGUIAR JR, R. **Desenvolvimento de uma Interface para Simulação com Animação 3D de Estruturas Flexíveis do Tipo Cabo**. 2016. Dissertação (Mestrado em Ciência da Computação) — Dissertação (Mestrado em Modelagem Computacional), Universidade Federal do Rio Grande.
- ANTONELLI, G. **Underwater Robots: motion and force control of vehicle-manipulator systems**. [S.l.]: Springer Berlin Heidelberg, 2013. (Springer Tracts in Advanced Robotics).
- ANTONELLI, G.; LEONESSA, A. Underwater robots: motion and force control of vehicle-manipulator systems. **IEEE Control Systems Magazine**, [S.l.], 2008.
- BANERJEE, A. K.; DO, V. N. Deployment control of a cable connecting a ship to an underwater vehicle. **Applied Mathematics and Computation**, [S.l.], v.70, n.2-3, p.97–116, 1995.
- BLENDER FOUNDATION. **Blender - a 3D modelling and rendering package**. Blender Institute, Amsterdam: Blender Online Community, 2016.
- BLENDER FOUNDATION. **Blender 2.78 Manual - User Interface**. https://docs.blender.org/manual/ko/dev/interface/window_system/introduction.html, 2017.
- BLENDER FOUNDATION. **Blender 2.78 Manual - Ocean Modifier**. <https://docs.blender.org/manual/ko/dev/modeling/modifiers/simulate/ocean.html>, 2017.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. [S.l.]: Elsevier Brasil, 2006.
- CHANG, L. **The Software Behind Frank Gehry's Geometrically Complex Architecture**. 2015.
- DEVELOPERS, N. NumPy. **NumPy Numpy. Scipy Developers**, [S.l.], 2013.
- DOMINGUEZ, R. Simulação e controle de um veículo submarino de operação remota. **Master's thesis, Programa de Engenharia Elétrica, COPPE/UFRJ, Rio de Janeiro**, [S.l.], 1989.
- ECK, D. J. **Introduction to Computer Graphics**. 1.1.ed. <http://math.hws.edu/graphicsbook/>: Department of Mathematics and Computer Science - Hobart and William Smith Colleges, 2016.
- EDGETECH. **Offshore AUV/ ROV Sonar Systems**. <https://www.edgetech.com/applications/offshore-auv-rov-sonar-systems>, 2017.

- FJELLSTAD, O.-E. **Control of Unmanned Underwater Vehicles in Six Degrees of Freedom**: a quaternion feedback approach. dr. ing. 1994. Tese (Doutorado em Ciência da Computação) — thesis, Rep. 1994: 92-W, Department of Engineering Cybernetics, The Norwegian Institute of Technology, Trondheim, Norway.
- FOSSSEN, T. **Guidance and control of ocean vehicles**. [S.l.]: Wiley, 1994.
- FOSSSEN, T. I.; BALCHEN, J. G. et al. The NEROV autonomous underwater vehicle. In: OCEANS, 1991. **Proceedings...** [S.l.: s.n.], 1991. v.91, p.1142–1155.
- FOWLER, M. **UML Essencial**: um breve guia para linguagem padrão. [S.l.]: Bookman Editora, 2014.
- FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A.; POWELL, J. D. **Feedback control of dynamic systems**. [S.l.]: Addison-Wesley Reading, MA, 1994. v.3.
- GOMES, S. C. P.; BIER, C. C. Estudo sobre trajetórias de controle para robôs manipuladores. In: XII CONGRESSO BRASILEIRO DE AUTOMÁTICA, 1998. **Anais...** [S.l.: s.n.], 1998.
- GOMES, S. C. P.; ROSA, V. S. da; ALBERTINI, B. C. Active control to flexible manipulators. **IEEE/ASME Transactions on Mechatronics**, [S.l.], v.11, n.1, p.75–83, 2006.
- GOMES, S. C.; ZANELA, E. B.; PEREIRA, A. E. Automatic generation of dynamic models of cables. **Ocean Engineering**, [S.l.], v.121, p.559 – 571, 2016.
- GOULART, C. Modelagem, simulação e controle de um veículo submarino de operação remota. **Rio de Janeiro**, [S.l.], 2007.
- HILL, F. S.; KELLEY, S. M. **Computer graphics**: using opengl (3rd edition). [S.l.]: Prentice hall Upper Saddle River, NJ, 2007. v.2.
- HSU, L. Avaliação experimental da modelagem e simulação da dinâmica de um veículo submarino de operação remota. **Sociedade Brasileira de Automática**, [S.l.], 2000.
- HUNTER, J. D. Matplotlib: a 2d graphics environment. **Computing In Science & Engineering**, [S.l.], v.9, n.3, p.90–95, 2007.
- INDUSTRYARC. **Unmanned Underwater Vehicles (UUV) Market**: by type (auv, rov, hybrid), by applications (defense, scientific and academic research, commercial exploration, retrieval system), by rov class (work, observation) - with forecast (2017 - 2022). [S.l.]: IndustryARC Research Reports, 2017.
- JAN, M. **Pixels and voxels, the long answer**. 2016.
- JOHN F. HUGHES ANDRIES VAN DAM, M. M. D. F. S. J. D. F. S. K. F. K. A. **Computer Graphics**: principles and practice (3rd edition). [S.l.]: Pearson Education, 2014.

- KINTEL, M.; WOLF, C. **OpenSCAD, The Programmers Solid 3D CAD Modeller**. 2017.
- LEARN OPENGL. **Learn OpenGL - Lighting: basic lighting**. <https://learnopengl.com/#!Lighting/Basic-Lighting>, 2017.
- LEARN OPENGL ES. **Understanding OpenGL's Matrices - left-handed coordinate system**. <http://www.learnopengles.com/tag/left-handed-coordinate-system/>, 2012.
- MANSSOUR, I. H.; COHEN, M. Introdução à Computação Gráfica. **RITA**, [S.l.], v.13, n.2, p.43–68, 2006.
- MARZO, G. R. D. **Aplicação do método dos elementos finitos na análise de tensões induzidas em cabos umbilicais**. 2010. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo.
- MATHWORKS. **MATLAB version 8.6 (R2015b)**. Natick, Massachusetts: The MathWorks Inc., 2015.
- MCKESSON, J. L. Learning Modern 3D Graphics Programming. **Arcsynthesis.org**, [S.l.], v.17, 2012.
- OGATA, K.; SEVERO, B. **Engenharia de controle moderno**. [S.l.]: Prentice Hall do Brasil, 1998.
- OLIVEIRA, V. S. de. **Algoritmos genéricos para a geração de modelos dinâmicos de cabos umbilicais e veículos subaquáticos**. 2015. Dissertação (Mestrado em Ciência da Computação) — Dissertação (Mestrado em Modelagem Computacional), Universidade Federal do Rio Grande.
- ON LINE EDITORA. **Guia A Profundez do Oceano Ed.01: segredos e mistérios do fundo do mar**. [S.l.]: On Line Editora, 2017. (Profundez do Oceano).
- PEREIRA, A. E. L. **O método da decomposição de Adomian aplicado à interação fluido-estrutura de um cabo**. 2010. Tese (Doutorado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.
- PETERSON, D. **The Phase One XF Story – Past, Present & Future**. 2015.
- PYTHON SOFTWARE FOUNDATION. **The Python Standard Library - The ElementTree XML API**. Disponível em <https://docs.python.org/3/library/xml.etree.elementtree.html>. Acesso em 25/06/17.
- PYTHON SOFTWARE FOUNDATION. **Graphical User Interface FAQ - Python 3.6.2 documentation**. Disponível em <https://docs.python.org/3.6/faq/gui.html>. Acesso em 01/07/17.

- RAYBAUT, P.; NYO, G. **Python(x,y) - the scientific Python distribution**. Disponível em <https://python-xy.github.io/>. Acesso em 24/06/17.
- RIVERBANK COMPUTING. **Riverbank — Software — PyQt — What is PyQt?** Disponível em: <https://riverbankcomputing.com/software/pyqt/>. Acesso em 13/06/2017.
- ROCHA, C. R. **Planejamento de movimento de sistemas robóticos de intervenção subaquática baseado na teoria dos helicoides**. 2012. Dissertação (Mestrado em Ciência da Computação) — .
- ROCHA, I. B. P. **Modelagem matemática de cabos umbilicais e veículos subaquáticos não tripulados**. 2013. Dissertação (Mestrado em Ciência da Computação) — Dissertação (Mestrado em Modelagem Computacional), Universidade Federal do Rio Grande.
- RUDOLPH, T. **Blender Animation - Ocean Modifier Simulation (Cycles Render)**. 2014.
- SHREINER, D.; SELLERS, G.; KESSENICH, J. M.; LICEA-KANE, B. M. **OpenGL Programming Guide: the official guide to learning opengl, version 4.3**. 8th.ed. [S.l.]: Addison-Wesley Professional, 2013.
- SICILIANO, B.; KHATIB, O. **Springer Handbook of Robotics**. [S.l.]: Springer Berlin Heidelberg, 2008. (Springer Handbook of Robotics).
- SILVEIRA, E. da. Veículos subaquáticos - Protótipos de robôs para uso embaixo d'água são testados na exploração de petróleo e pesquisas oceanográficas. **Revista Pesquisa FAPESP**, [S.l.], n.244, p.62–65, 2016.
- SLOTINE, J.-J. E. et al. **Applied nonlinear control**. [S.l.]: Pearson, 1991. v.199, n.1.
- SNAME. Nomenclature for treating the motion of a submerged body through a fluid. **The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin No.**, [S.l.], p.1–5, 1950.
- SOKOLOV, S.; ZHILENKOV, A.; NYRKOV, A.; CHERNYI, S. The Use Robotics for Underwater Research Complex Objects. In: **Computational Intelligence in Data Mining**. [S.l.]: Springer, 2017. p.421–427.
- SORIMAJI, T. **ROV - Remotely operated underwater vehicle**. 2015.
- TAVARES, A. M. **Um estudo sobre a modelagem e o controle de veículos subaquáticos não tripulados**. 2003. Dissertação (Mestrado em Ciência da Computação) — .
- VAN ROSSUM, G. et al. Python Programming Language. In: **USENIX ANNUAL TECHNICAL CONFERENCE, 2007**. **Anais...** [S.l.: s.n.], 2007. v.41, p.36.

- WERNLI, R. L. Recent US Navy underwater vehicle projects. In: 2001. **Anais...** Proceedings of the 24th Marine Facilities Panel, 2001.
- WHITCOMB, L. L. Underwater robotics: out of the research laboratory and into the field. In: ROBOTICS AND AUTOMATION, 2000. PROCEEDINGS. ICRA'00. IEEE INTERNATIONAL CONFERENCE ON, 2000. **Anais...** [S.l.: s.n.], 2000. v.1, p.709–716.
- YUH, J. Design and Control of Autonomous Underwater Robots: a survey. **Autonomous Robots**, [S.l.], v.8, n.1, p.7–24, 2000.
- ZANELA, E. B. **Modelagem analítica de estruturas do tipo cabo para aplicações subaquáticas**. 2013. Tese (Doutorado em Ciência da Computação) — Universidade Federal de Rio Grande.