

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

DESENVOLVIMENTO DE UMA INTERFACE PARA A SIMULAÇÃO COM
ANIMAÇÃO TRIDIMENSIONAL DE ESTRUTURAS FLEXÍVEIS DO TIPO CABO

ROBERTO AGUIAR JUNIOR

Dissertação apresentada à Comissão do Programa de Pós-Graduação em Modelagem Computacional da Universidade Federal do Rio Grande, como requisito parcial para a obtenção do título de Mestre em Modelagem Computacional.

Orientador: Prof. Dr. Sebastião Cícero Pinheiro
Gomes

Coorientador: Prof. Dr. Carlos Rodrigues Rocha

Rio Grande, fevereiro de 2016

*Dedico este trabalho à
minha esposa e ao meu filho.*

AGRADECIMENTOS

Primeiramente, agradeço a Deus, por ter me dado forças para desenvolver este trabalho e guiado meus passos ao longo desta caminhada.

Ao meu orientador, Prof. Dr. Sebastião Cícero Pinheiro Gomes, pelo excelente trabalho que vem desenvolvendo na sua área de atuação. Agradeço por toda compreensão, colaboração e dedicação nesta dissertação.

Ao meu coorientador Prof. Dr. Carlos Rodrigues Rocha, por todo auxílio, tempo despendido e atenção dada às minhas dúvidas. Obrigado me conduzir no desenvolvimento deste trabalho.

Ao Programa de Pós-Graduação em Modelagem Computacional, pela oportunidade de realizar este trabalho.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela concessão de recursos ao longo do desenvolvimento deste trabalho.

À minha família, pelo apoio e participação em todos os momentos de minha vida, em especial a minha esposa Daniela e meu filho Samuel por estarem ao meu lado em todos os momentos, compreendendo a minha falta de tempo e por este período não poder estar presente na vida deles como gostaria.

A meu amigo Felipe Herrmann por ter sido uma pessoa que me incentivou ao longo desta jornada permitindo que eu chegasse até aqui. Muito Obrigado.

Aos colegas do Núcleo de Matemática Aplicada (NuMA) Jéssica Campos, Thiago Ávila e Vanessa Oliveira, pela convivência e companheirismo ao longo da etapa que se encerra.

A todos os colegas do Programa de Pós-Graduação em Modelagem Computacional que tive o prazer de conviver, Carlos Ferreira e Fabiane Harter, pela parceria e auxílio no início do meu Mestrado.

Aos membros da Banca Examinadora, avaliadores desta Dissertação, pela disponibilidade para tal.

A todos os professores que, de alguma forma, contribuíram para a realização desta Dissertação.

RESUMO

Aplicações tecnológicas voltadas ao ambiente subaquático têm atraído a atenção da comunidade científica, especialmente no caso de pesquisas relacionadas à exploração do petróleo e gás *offshore*. Destas aplicações, destacam-se diversos tipos de estruturas flexíveis como cabos e *risers*, bem como veículos subaquáticos não tripulados, como é o caso dos ROVs (Remotely Operated Vehicles, ou Veículos Remotamente Operados). A presente dissertação insere-se neste contexto. Inicialmente, foram pesquisados formalismos de modelagem dinâmica de estruturas flexíveis do tipo cabo, bem como de modelos dinâmicos de ROV. Escolheu-se um formalismo discreto para a modelagem dinâmica de cabos, o qual aproxima o cabo por diversos elos rígidos conectados entre si por juntas esféricas, cada uma permitindo movimentos rotacionais em três graus de liberdade: elevação, azimute e torção. Foram desenvolvidos algoritmos para gerar os modelos dinâmicos de forma automática, em função do número de elos usados para aproximar de forma discreta a flexibilidade contínua. O ambiente a ser simulado consiste em um cabo fixo em uma de suas extremidades, como uma embarcação na superfície, que a conecta a um ROV na outra extremidade. Este é considerado a carga terminal do cabo, que é vinculado a ele por uma junta esférica. Uma vez testada a consistência dos resultados de simulação, foi desenvolvido um *software* que gera uma visualização em três dimensões dos resultados, objetivo principal da presente dissertação. O *software* foi desenvolvido utilizando-se a linguagem de programação Python e a biblioteca gráfica OpenGL (*Open Graphics Library*). Bibliotecas foram desenvolvidas para criar a modelagem do mundo virtual com visualização por câmeras sintéticas. A interface do usuário foi baseada no *framework* Qt, através da biblioteca PyQt. Analisaram-se, de forma quantitativa, as animações tridimensionais realizadas no aplicativo desenvolvido e constatou-se que as simulações apresentaram-se de acordo com o esperado fisicamente.

Palavras-chave: estruturas flexíveis, modelagem dinâmica, algoritmos genéricos, desenvolvimento, animações tridimensionais, Python, OpenGL.

ABSTRACT

Technological applications related to the underwater environment have attracted the attention of the scientific community, especially in the case of research related to the oil/gas offshore exploration. From these applications, flexible structures as cables and risers, and unmanned underwater vehicles, such as ROVs (Remotely Operated Vehicles) are of particular interest. The present dissertation is inserted in this context. Initially, dynamic modeling formalisms of flexible structures as cables and ROV vehicles were studied. It was chosen a discrete formalism for dynamic modeling of cables, where a cable is represented by a serial chain of rigid links connected by spherical joints, each one allowing rotational movement in three degrees of freedom: elevation, azimuth and twist. Algorithms to automatically generate dynamic models as function of the number of links used to approximate the discrete serial chain to the real continuous flexibility were developed. The environment to be simulated consist in a cable that is fixed at one end, as in a vessel in the water surface, connecting it to a ROV at the other end. This is considered as the cable terminal load, and it is linked to the cable by a spherical joint. Once the consistency of the simulation results was validated, a software was developed which aims to generate three dimensional views of the environment. This is the main objective of this dissertation. The software was developed based on the Python programming language and the OpenGL (Open Graphics Library) graphics library. To graphically model the environment, libraries were developed to represent virtual worlds viewed by synthetic cameras. The user interface was based on the Qt framework, using the PyQt library. Three dimensional animations generated by the software were qualitatively analyzed and it was observed that the simulations behaved as it was physically expected.

Key-words: flexible structures, dynamic modelling, generic algorithms, software development, three-dimensional animation, Python, OpenGL.

SUMÁRIO

LISTA DE SÍMBOLOS	x
LISTA DE FIGURAS	xii
LISTA DE TABELAS	xiv
1. INTRODUÇÃO	1
1.1 OBJETIVOS	2
<i>1.1.1 Objetivo Geral.....</i>	<i>2</i>
<i>1.1.2 Objetivos Específicos</i>	<i>2</i>
1.2 REVISÃO BIBLIOGRÁFICA	2
<i>1.2.1 Modelagem de Cabos.....</i>	<i>3</i>
<i>1.2.2 Desenvolvimento do Software</i>	<i>6</i>
<i>1.2.3 Biblioteca OpenGL.....</i>	<i>9</i>
1.3 ROTEIRO DO TRABALHO.....	10
2. ALGORITMOS GENÉRICOS.....	12
2.1 INTRODUÇÃO	12
2.2 MODELAGEM ANALÍTICA DE ESTRUTURAS FLEXÍVEIS DO TIPO CABO...	12
2.3 CABO COM TRÊS ELOS	25
2.4 CABO COM QUATRO ELOS	29
2.5 ALGORITMOS GENÉRICOS	34
<i>2.5.1 Algoritmo Genérico para a Matriz de Inércia</i>	<i>35</i>
<i>2.5.2 Algoritmo Genérico para as Matrizes de Constantes Elásticas e de Coeficientes de Atrito.....</i>	<i>39</i>
<i>2.5.3 Algoritmo Genérico para o Vetor de Esforços Coriolis-centrifugo.....</i>	<i>40</i>
<i>2.5.4 Algoritmo Genérico para o Vetor Gravitacional.....</i>	<i>45</i>
2.6 SIMULAÇÃO NUMÈRICA	46
2.7 CONSIDERAÇÕES	48
3. METODOLOGIA E TECNOLOGIA DE DESENVOLVIMENTO.....	49
3.1 INTRODUÇÃO.....	49
3.2 DESCRIÇÃO DO PROBLEMA	49
3.3 METODOLOGIA DE DESENVOLVIMENTO DO SOFTWARE	51

3.3.1 Pacote cablesim.....	55
3.3.2 Pacotes gui e Qt	55
3.3.3 Pacote world	56
3.3.4 Pacote geometry.....	57
3.4 FERRAMENTAS DE DESENVOLVIMENTO DO APLICATIVO	61
3.4.1 Python.....	62
3.4.2 OpenGL.....	64
3.4.3 PyQt.....	65
3.5 CONSIDERAÇÕES	66
4.IMPLEMENTAÇÃO DO APLICATIVO	67
4.1 INTRODUÇÃO.....	67
4.2 ARQUIVO DE CABEÇALHO	67
4.3 ANIMAÇÃO	69
4.3.1 Desenvolvendo o mundo virtual.....	70
4.3.2 Execução da animação.....	72
4.4 CALIBRAÇÃO	74
4.5 GRÁFICO.....	74
4.6 INTERFACE DO USÁRIO	74
4.6.1 Classes e métodos Qt utilizadas	75
4.6.2 Funcionalidades	77
4.7 CONSIDERAÇÕES	84
5. RESULTADOS.....	85
5.1 INTRODUÇÃO.....	85
5.2 PRIMEIRA ANIMAÇÃO	85
5.3 SEGUNDA ANIMAÇÃO	89
5.4 TERCEIRA ANIMAÇÃO.....	93
5.5 QUARTA ANIMAÇÃO.....	97
5.6 CONSIDERAÇÕES	101
6. CONCLUSÕES	102
7. REFERENCIAS BIBLIOGRÁFICAS	104
APÊNDICE A - ELEMENTOS DAS MATRIZES E VETORES PARA O CABO	
DIVIDIDO EM TRÊS E QUATRO ELÓS.....	108

APÊNDICE A.1 – ELEMENTOS DAS MATRIZES E VETORES PARA O CABO DIVIDIDO EM TRÊS ELOS	108
APÊNDICE A.2 – ELEMENTOS DAS MATRIZES E VETORES PARA O CABO DIVIDIDO EM QUATRO ELOS	116
APÊNDICE B - ESPAÇO DE TRABALHO TRIDIMENSIONAL	133

LISTA DE SÍMBOLOS

$a_i, 1, \dots, n$	<i>Alturas de referencia para a definição da energia potencial de cada elo.</i>
C	<i>Matriz de coeficientes de atrito da dinâmica do cabo.</i>
C_e, C_a, C_t	<i>Submatrizes de matriz C, em relação aos movimentos de elevação, azimute e torção.</i>
C_{ia}, C_{ie} e C_{iT}	<i>Coefficientes de atrito devido aos ângulos de azimute, elevação e torção.</i>
E_c	<i>Energia cinética.</i>
E_{CR}	<i>Energia Cinética devida ao movimento de rotação.</i>
E_{CT}	<i>Energia Cinética devida ao movimento de translação.</i>
E_P	<i>Energia Potencial.</i>
$\vec{f}(\vec{\theta}, \dot{\vec{\theta}}), \vec{F}$	<i>Vetor Coriolis-centrífugos.</i>
g	<i>Aceleração gravitacional.</i>
$\vec{G}(\vec{\theta}, \dot{\vec{\theta}}), \vec{G}$	<i>Vetor de esforços gravitacionais.</i>
$h_i, i = 1, \dots, n$	<i>Alturas.</i>
$H_{0_1}, H_{0_2}, H_{1_2}$	<i>Matrizes de transformações homogêneas.</i>
$I_{R_{ie}}$	<i>Momentos de inércia relativos ao movimento de elevação dos elos.</i>
$I_{R_{ia}}$	<i>Momentos de inércia relativos ao movimento de azimute dos elos.</i>
I_{T_i}	<i>Momentos de inércia devido ao movimento de torção dos elos.</i>
$I(\vec{\theta}), I$	<i>Matriz de inércia completa.</i>
I_e, I_a, I_t	<i>Submatrizes da matriz de inércia completa.</i>
K	<i>Matriz de constantes elásticas das articulações do cabo.</i>
K_e, K_a e K_t	<i>Submatrizes da matriz K, em relação aos movimentos de elevação, azimute e torção.</i>
K_{ia}, K_{ie} e K_{iT}	<i>Constantes elásticas devidas aos ângulos de azimute, elevação e torção</i>
L	<i>Lagrangeano do sistema físico.</i>
l, l_c	<i>Comprimento do cabo.</i>
$l_i, i = 1, \dots, n$	<i>Comprimento de cada elo.</i>
m_c	<i>Massa da carga terminal.</i>
n	<i>Números de elos.</i>
N_e, N_a e N_t	<i>Submatrizes de matriz de inércia completa do cabo.</i>
t	<i>Tempo.</i>

\vec{T}_m, \vec{T}	<i>Vetor de torques externos.</i>
T_e, T_a e T_t	<i>Submatrizes da matriz de inércia completa do cabo.</i>
$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{ie}} \right)$	<i>Derivada total em relação ao tempo.</i>
$\frac{\partial L}{\partial \theta_{ie}}, \frac{\partial L}{\partial \theta_{ia}}, \frac{\partial L}{\partial \theta_{it}}$	<i>Derivadas parciais do Lagrangeano.</i>
$\theta_{ia}, \theta_{ie}, \theta_{it}$	<i>Ângulos de azimute, elevação e torção de cada elo.</i>
$\dot{\theta}_{ia}, \dot{\theta}_{ie}, \dot{\theta}_{it}$	<i>Derivadas em relação ao tempo das posições em azimute, elevação e torção.</i>
$\vec{\theta}$	<i>Vetor de estado com todas as posições angulares dos elos.</i>
$\dot{\vec{\theta}}$	<i>Derivada primeira do vetor $\vec{\theta}$ em relação a t.</i>
X_0, Y_0, Z_0	<i>Sistema no referencial inercial.</i>
X_i, Y_i, Z_i	<i>Sistema de referência com origem na articulação i.</i>
(x_c, y_c, z_c)	<i>Coordenadas do centro de massa de carga terminal.</i>
(x_i, y_i, z_i)	<i>Coordenadas do centro de massa do i-ésimo elo.</i>

LISTA DE FIGURAS

Figura 1.1- Envio de dados para GPU (ANGEL e SHREINER [7]).	10
Figura 2.1- Representação esquemática do cabo umbilical conectado a uma massa mc.	12
Figura 2.2- Representação esquemática do cabo umbilical conectado a um ROV.	13
Figura 2.3 – Estrutura flexível e sua representação discreta.	14
Figura 2.4 – Ângulos de azimute, elevação e torção (PEREIRA [1]).	14
Figura 2.5 – Ângulos de azimute, elevação e torção na i-ésima articulação (ROCHA [10]).	15
Figura 2.6 – Cabo com quatro (4) elos (PEREIRA [2]).	17
Figura 2.7 – Representação geométrica dos dois primeiros sistemas de referência (PEREIRA [2]).	19
Figura 2.8 – Representação geométrica do cabo dividido em quatro articulações fictícias (ZANELA [1]).	25
Figura 2.9 – Representação geométrica do cabo dividido em quatro articulações fictícias (ZANELA [1]).	30
Figura 2.10 - Simulação considerando-se 16 elos para aproximar a flexibilidade contínua.	48
Figura 3.1 – Representação de uma linha do arquivo TXT.	50
Figura 3.2 – Casos de uso do software CableSim.	53
Figura 3.3 – Diagrama de classe do software CableSim.	54
Figura 4.1 – Formato XML do arquivo do cabeçalho.	69
Figura 4.2 – Representação das coordenadas cartesianas.	71
Figura 4.3 – Exemplo que ilustra a criação do mundo virtual contendo 4 elos.	71
Figura 4.4 – Exemplo que ilustra a posição de um cabo com 4 elos no instante de tempo $t_0 = 0$.	73
Figura 4.5 – Layout da janela principal do aplicativo CableSim (QT [34]).	76
Figura 4.6 – Janela principal do aplicativo CableSim.	78
Figura 4.7 – Diálogo Sobre.	79
Figura 4.8 – Diálogo que gerencia o cabeçalho do arquivo.	80
Figura 4.9 – Janela principal da animação com a opção coordenadas habilitada.	81
Figura 4.10 – Janela que apresenta o gráfico da carga terminal com base nos dados do arquivo padrão da simulação.	82
Figura 4.11 – Visualização do cabo a partir de seis ângulos distintos.	84

Figura 5.1 – Primeira animação (1, 5 fps): cabo com 32 metros de comprimento dividido em 8 elos de tamanho 4 metros.....	88
Figura 5.2 – Gráfico do deslocamento da carga terminal da primeira animação.....	89
Figura 5.3 – Segunda animação (13,75 fps): cabo com 22 metros de comprimento dividido em 22 elos de tamanho 1 metro.....	92
Figura 5.4 – Gráfico do deslocamento da carga terminal da segunda animação.	93
Figura 5.5 – Terceira animação (7,5 fps): cabo com 32 metros de comprimento dividido em 32 elos de tamanho 1 metro.....	96
Figura 5.6– Gráfico do deslocamento da carga terminal da terceira animação.	97
Figura 5.7 – Quarta animação (13,75 fps): cabo com 36 metros de comprimento dividido em 36 elos de tamanho 1 metro.....	100
Figura 5.8 – Gráfico do deslocamento da carga terminal da quarta animação.	101
Figura B.1– Sistema SRU em três dimensões utilizando OpenGL (COLARES [36]).....	133
Figura B.2 – Câmera sintética do OpenGL posicionada de duas formas diferentes (COHEN e MANSSOUR [32]).....	134
Figura B.3 – Projeção de paralelepípedos (COHEN e MANSSOUR [32]).....	135

LISTA DE TABELAS

Tabela 2.1 – Parâmetros físicos usado na simulação.	47
Tabela 3.1 – Tabela comparativa das linguagens C/C++, Fortran e Python (adaptada do artigo de RASHED e AHSAN [17]).....	63
Tabela 3.2 – Tabela do IEEE com a classificação das linguagens mais populares (adaptada IEEE [43]).	64
Tabela 3.3 – Tabela comparativa das bibliotecas Qt [26], Microsoft Foundation Class (MICROSOFT [44]), GTK [45]e Swing (ORACLE [46]).	65
Tabela 4.1 – Parâmetros de uma simulação.	68
Tabela 4.2 – Configuração das opções de retorno da janela Arquivo.....	77
Tabela 4.3 – Teclas de atalho do CableSim.	78
Tabela 5.1 – Parâmetros configurados no CableSim para a primeira animação.....	86
Tabela 5.2 – Componentes da câmera sintética utilizados na primeira animação.	86
Tabela 5.3 – Parâmetros configurados no CableSim para a segunda animação.	90
Tabela 5.4 – Componentes da câmera sintética utilizados na segunda animação.....	90
Tabela 5.5 – Parâmetros configurados no CableSim para a terceira animação.	94
Tabela 5.6 – Componentes da câmera sintética utilizados na terceira animação.....	94
Tabela 5.7 – Parâmetros configurados no CableSim para a quarta animação.	98
Tabela 5.8 – Componentes da câmera sintética utilizados na quarta animação.....	98
Tabela 5.9 – Parâmetros referentes ao desempenho das simulações realizadas neste Capítulo.	101

1. INTRODUÇÃO

O estudo referente à modelagem de estruturas flexíveis do tipo cabo está em constante desenvolvimento por possuir diversas aplicações fortemente relevantes, principalmente nas indústrias de petróleo (ZANELA [1]). Os cabos umbilicais estão presentes em plataformas petrolíferas *offshore*, plataformas e unidades flutuantes de produção e unidades móveis marítimas de perfuração. Neste meio, um exemplo de aplicação consiste em cabos conectados entre estruturas flutuantes e veículos subaquáticos não tripulados do tipo ROV (*Remotely Operated Vehicle*).

Atualmente, têm-se desenvolvido trabalhos sobre a modelagem dinâmica de estruturas flexíveis do tipo cabo utilizando-se o formalismo de Euler-Lagrange. PEREIRA [2], baseando-se no formalismo discreto (*Lumped Mass Approach*) criou um método que obtém o modelo dinâmico de estruturas flexíveis do tipo cabo. Esse método supõe um cabo formado por pequenos elos rígidos conectados por articulações fictícias esféricas elásticas sendo que cada articulação permite três movimentos livres: elevação, azimute e torção. ZANELA [1] criou algoritmos genéricos que, independentes do número de elos que o cabo é dividido, permitem gerar automaticamente as equações dos modelos dinâmicos para as estruturas flexíveis.

Para validar seus algoritmos genéricos, ZANELA [1] utilizou o *software* SIMMODEL CAB desenvolvido por FABBER e VIEIRA [3], que gera automaticamente os elementos das matrizes e vetores do modelo dinâmico de cabos. Esse sistema foi criado utilizando como base a tese de doutorado de PEREIRA [2] e uma caixa de ferramentas do MATLAB que faz manipulação simbólica de variáveis. E essa manipulação torna-se muito pesada computacionalmente conforme se aumenta o número de elos do cabo. O MATLAB (MATHWORKS [4]) é um aplicativo que possui um custo de licença alto e o aplicativo SIMMODEL CAB fica vinculado diretamente a essa despesa.

O propósito desta dissertação é apresentar os resultados obtidos a partir dos algoritmos genéricos em forma de simulações gráficas tridimensionais, conforme o número de elos em que o cabo é dividido, e considerando-se também a utilização de aplicativos com licença gratuita e sem vínculo com outros softwares, apresentando assim, uma nova forma de visualização e análise das simulações.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral desse trabalho é a construção de um *software* de visualização em três dimensões, que permita a animação da simulação dinâmica de estruturas flexíveis do tipo cabo a partir dos algoritmos genéricos de ZANELA [1].

1.1.2 Objetivos Específicos

Têm-se os seguintes objetivos específicos:

- Realizar um estudo sobre os algoritmos genéricos criados por ZANELA [1]. Pesquisar linguagens de programação e interfaces de programação de aplicativo (API) gráficas apropriadas para a realização de simulações a fim de determinar qual é a melhor linguagem e API para criar o *software*. Definir as ferramentas de desenvolvimento para o desempenho ao realizar as animações de cabos umbilicais.
- Realizar um estudo aprofundado dos recursos importantes da linguagem e API e implementar uma cena em três dimensões permitindo a visualização do resultado da simulação da modelagem dinâmica das estruturas flexíveis do tipo cabo na forma de emulação de um cenário virtual do ambiente simulado.
- Executar dentro da cena simulações informando a quantidade de elos escolhidos e tamanho de cada um deles analisando qualitativamente os resultados obtidos.

1.2 REVISÃO BIBLIOGRÁFICA

A seguir apresentam-se breves resumos dos trabalhos que serviram de referência para o entendimento do problema, permitindo a implementação deste projeto de dissertação. A revisão foi dividida em três subseções: a primeira aborda a modelagem dinâmica de cabos umbilicais, a segunda trata da modelagem e desenvolvimento do *software* e a última faz uma

explicação sobre a biblioteca OpenGL (KHRONOS [5]) utilizada na implementação deste trabalho.

1.2.1 Modelagem de Cabos

- **A. E. L. PEREIRA (1999)**

Um Estudo sobre Modelagem Matemática de Estruturas Flexíveis.

Neste trabalho de dissertação foi desenvolvida e validada a técnica do formalismo Discreto, que estudou a modelagem dinâmica de estruturas flexíveis no plano, sendo esta estrutura articulada em uma extremidade e livre na outra. A técnica considera uma estrutura flexível dividida em duas, três ou quatro partes rígidas conectadas por articulações elásticas e permite apenas o movimento de elevação no espaço bidimensional.

- **A. E. L. PEREIRA (2010)**

O método da decomposição de Adomian aplicado à interação fluido-estrutura de um cabo.

A tese apresenta um estudo sobre a interação fluido-estrutura de um cabo submerso em um fluido, acoplando a modelagem da dinâmica do cabo com o movimento do fluido. Com base no formalismo de Euler-Lagrange foi criado um formalismo discreto que obtém as equações de movimento de cabos umbilicais. Para essa formulação é proposto que a flexibilidade contínua é representada por uma aproximação discreta, formada de elos rígidos conectados por articulações elásticas fictícias, permitindo três movimentos distintos: azimute, elevação e torção, no espaço tridimensional. É considerado que o cabo é cilíndrico e propõe o escoamento sobre um cilindro circular para o cálculo do movimento do fluido. Esse movimento é determinado resolvendo-se analiticamente as equações de Navier-Stokes através do Método da Decomposição de Adomian.

- **E. B. ZANELA (2013)**

Modelagem Analítica de Estruturas do Tipo Cabo para Aplicações Subaquáticas.

Esse trabalho de dissertação aplica o formalismo Discreto para a modelagem dinâmica de estruturas flexíveis tridimensionais do tipo cabo, tomando como referência o formalismo

de Euler-Lagrange. O método supõe um cabo formado por pequenos elos rígidos conectados por articulações fictícias esféricas elásticas no qual cada parte permite três movimentos livres: elevação, azimute e torção. Com base no exposto, foram criados algoritmos genéricos, independentes do número de elos que o cabo é dividido, que permitem gerar automaticamente as equações dos modelos dinâmicos para as estruturas flexíveis. Foi utilizado também na dissertação um aplicativo, desenvolvido usando uma toolbox de manipulação simbólica de variáveis do MATLAB, que validou os algoritmos genéricos.

- **I. B. P. ROCHA (2013)**

Modelagem Matemática de Cabos Umbilicais e Veículos Subaquáticos Não Tripulados.

A dissertação propõe um formalismo para a unificação da dinâmica de veículos subaquáticos não tripulados do tipo ROV (*Remotely Operated Vehicle*) e a dinâmica do cabo umbilical. A modelagem física do veículo considera aspectos dinâmicos e cinemáticos. Para desenvolver o modelo dinâmico foi considerada a interação entre o veículo e o meio aquático e a ação dos propulsores. Para a modelagem do cabo foi utilizado um formalismo discreto que trata a flexibilidade contínua do cabo aproximada de forma discreta, formando pequenas hastes rígidas conectadas por articulações fictícias que permitem os movimentos elásticos de azimute, elevação e torção. O acoplamento da dinâmica aloca o ROV na extremidade livre do cabo, tratando-o como uma carga terminal com dinâmica própria e influenciada pelos seus atuadores. O modelo foi validado de acordo com os resultados obtidos das simulações do cabo dividido em no máximo cinco elos.

- **T. Á. POUZADA (2015)**

Algoritmos Genéricos para a Geração Automática de Modelos de Cabos em Três Situações Práticas de Aplicações Subaquáticas.

Essa dissertação mostra o desenvolvimento de algoritmos genéricos para a geração automática do modelo dinâmico de estruturas flexíveis do tipo cabo acoplado a plataformas móveis, considerando a quantidade de elos nos quais se deseja fazer a divisão do cabo umbilical. O trabalho utiliza o Método do formalismo Discreto para desenvolver a modelagem dinâmica de cabos. Esse método consiste em dividir o cabo em elos rígidos

conectados por articulações elásticas fictícias. As estruturas móveis e o cabo foram analisados considerando-se três situações distintas. Na primeira situação, uma das extremidades está articulada a uma estrutura flutuante e a outra conectada a uma carga terminal que fica livre. Na segunda o que a diferencia da primeira é que a carga terminal passa a ser uma estrutura fixa ao fundo do oceano. Na terceira, as extremidades do cabo são articuladas a duas estruturas distintas e móveis. Os algoritmos genéricos obtidos nos três casos foram validados a partir de simulações.

- **V. S. D. OLIVEIRA (2015)**

Algoritmos Genéricos para a Geração de Modelos Dinâmicos de Cabos Umbilicais e Veículos Subaquáticos.

A dissertação apresenta o desenvolvimento de algoritmos genéricos para acoplar a modelagem de veículos subaquáticos não tripulados do tipo ROV (*Remotely Operated Vehicle*) com a modelagem dinâmica de estruturas flexíveis do tipo cabo, chamados, neste trabalho, de modelo global. A situação estudada trata um cabo articulado a uma estrutura fixa em uma extremidade e a um ROV na outra, sendo aplicadas a ele forças externas que transferem os torques do mesmo para o cabo, ou seja, é considerado que a carga terminal (ROV) possui dinâmica própria e influencia na dinâmica do cabo. A modelagem das estruturas flexíveis do tipo cabo utiliza o formalismo discreto, a partir do formalismo de Euler-Lagrange, que supõe um cabo formado por pequenos elos rígidos conectados por articulações fictícias flexíveis, permitindo movimentos em elevação, azimute e torção. O mesmo método de modelagem das estruturas flexíveis é utilizado para a modelagem de acoplamento da dinâmica do veículo subaquático com a dinâmica do cabo umbilical. A partir de um estudo da influência das forças do ROV na dinâmica do cabo foram desenvolvidos os algoritmos genéricos geradores do modelo global para um número qualquer de elos escolhidos para a dinâmica do cabo. Os algoritmos foram validados de forma qualitativa visualizando e analisando o comportamento das simulações utilizando animações tridimensionais.

1.2.2 Desenvolvimento do *Software*

- **P. F. DUBOIS (2007)**

Python: Batteries Included.

No artigo, são apresentados os recursos importantes da linguagem de programação Python e o crescente papel que desempenha em projetos científicos. Python é open source e possui uma grande quantidade de bibliotecas com suporte para a área da ciência da computação. Ele contém também diversas ferramentas de terceiros e módulos adicionais, tais como: SciPy/NumPy, IPython, IPython, MayaVi, SWIG, Boost/Python, PyCXX e f2py. A linguagem permite realizar várias tarefas como, por exemplos: criar sites, calcular a transformada rápida de Fourier, trabalhar com programação paralela e distribuída. Mesmo Python sendo uma linguagem interpretada, ela pode ser suficiente rápida para a programação científica e ainda, interativa.

- **T. E. OLIPHANT (2007)**

Python for Scientific Computing.

O artigo aborda aspectos relevantes para justificar o crescente uso da linguagem de programação Python no meio científico. Explica os motivos pelos quais cientistas e engenheiros utilizam Python para desenvolver aplicações científicas, sendo um deles, o fato da linguagem ser de alto nível. No artigo também são apresentadas características detalhadas sobre as razões pelas quais o Python tem se destacando como uma plataforma científica: a licença é open source, pode ser executado em vários sistemas operacionais, é uma linguagem que pode interagir com outros *softwares*, tem uma grande quantidade de bibliotecas, possui ferramentas para desenvolvimento de interface gráfica, entre outras.

- **J. K. MILLMAN; M. AIVAZIS (2011)**

Python for Scientists and Engineers.

No artigo são apresentadas as vantagens da utilização da linguagem de programação Python para pesquisa científica e também faz uma breve explicação sobre bibliotecas e ferramentas usadas na computação científica. SciPy é uma forte comunidade que inclui um

crescente grupo de cientistas, engenheiros e pesquisadores que usam, estendem e promovem o uso do Python para a pesquisa científica. Essa comunidade possui bibliotecas, tais como, NumPy, Matplotlib e SciPy que ajudam o Python a tornar-se um ambiente altamente produtivo para a computação científica moderna.

- **S. V. D. WALT; S. C. COLBERT; G. VAROQUAUX (2011)**

The NumPy Array: A Structure for Efficient Numerical Computation.

A linguagem Python fornece um conjunto de estruturas de alto nível de manipulação de dados, tais como listas e dicionários. Dentre as estruturas citadas, a linguagem possui o módulo NumPy que é uma biblioteca para manipulação de matrizes uni e multidimensionais. O artigo aborda as seguintes três técnicas para melhorar o desempenho do módulo NumPy: cálculos vetorizados, evitar a copia de dados para a memória e minimizar a quantidade de operações. Em qualquer linguagem de scripts, como por exemplo, Python, o uso excessivo de laços de repetições implica em um baixo desempenho da linguagem. A biblioteca NumPy permite a realização de operações matemáticas com matrizes utilizando o recurso de vetorização sendo que estas, são realizadas em C e conseqüentemente, ocorre uma melhora significativamente na velocidade dos cálculos. O módulo NumPy possui recursos eficientes para otimizar o uso de memória ao trabalhar-se com a manipulação de matrizes, tais como: broadcasting e o mapeamento de memória. Numpy pode solucionar problemas numéricos na ciência utilizando um código de alto nível, transparente e fácil gerenciamento.

- **E. ANGEL; D. SHREINER (2011)**

Teaching a Shader-Based Introduction to Computer Graphics.

O artigo faz uma abordagem sobre o ensino da interface de programação de aplicativos OpenGL (*Open Graphics Library*) nas aulas introdutórias de computação gráfica nos cursos de ciência da computação, engenharia e matemática. OpenGL é uma biblioteca gráfica de código aberto multiplataforma que mudou radicalmente e incluiu novos recursos de *hardware*. São apresentadas as concepções antigas da biblioteca e são expostos exemplos da nova forma de programação com OpenGL, que inclui os pipelines programáveis. Entre as diversas vantagens abordadas pelo artigo, uma delas é o desempenho considerável quando comparado com o modelo antigo. Espera-se que com a inclusão dos novos recursos, tais como

shaders programáveis, o ensino do OpenGL moderno nas aulas iniciais de computação gráfica gere vantagens para novos estudantes da área.

- **M. G. RASHED; R. AHSAN (2012)**

Python in Computational Science: Applications and Possibilities.

Mesmo Python sendo uma linguagem de programação de propósito geral, o artigo expõe as principais características que torna o Python ideal para a computação científica. Mostra também um gráfico com o crescente número de usuários da linguagem e apresenta um comparativo entre as linguagens Fortran, C/C++ e Python, destacando as vantagens de Python para uso na ciência computacional. Rapidez na reutilização do código-fonte, possuir diferentes arquiteturas para a computação paralela, ter vários *frameworks* disponíveis para a interface gráfica com o usuário e conter as principais bibliotecas científicas, são as vantagens detalhadas no artigo. Apesar de Python ser um pouco mais lento que outras linguagens, ela é de fácil aprendizado, possui uma estrutura de dados de alto nível e é orientada a objetos. Para concluir, o artigo destaca três principais benefícios da utilização do Python na ciência computacional.

- **G. REINA; T. MÜLHER; T. ERTL (2014)**

Incorporating Modern OpenGL into Computer Graphics Education.

O artigo explana sobre a metodologia de ensino utilizada pela Universidade de Stuttgart para atualizar seu curso de ciência de computação para incorporar o OpenGL moderno. Foi desenvolvido um *framework* que utiliza o OpenGL legado como base de estudos para os alunos compreenderem a OpenGL moderno. O artigo aborda conceitos introdutórios importantes do OpenGL moderno comparando com definições antigas do OpenGL, incluindo um exemplo para renderização de um triângulo através do *framework* desenvolvido. A nova metodologia de ensino se mostrou eficaz conforme o retorno obtido pelos alunos da universidade.

1.2.3 Biblioteca OpenGL

De acordo com as leituras realizadas na subseção 1.2.2 identificou-se a necessidade de se ampliar o estudo da biblioteca gráfica OpenGL devido ao fato de alterações significativas na estrutura da mesma.

Segundo ANGEL e SHREINER [7], a partir da versão 3.1 a biblioteca OpenGL passou por diversas mudanças para se adequar e suportar os novos recursos de hardware, introduzindo o *pipeline* com *shaders* programáveis, ou seja, o *pipeline* programável. Diante deste novo recurso do OpenGL, chamado atualmente de OpenGL moderno, todas as aplicações requerem *shaders* programáveis, que são programas desenvolvidos na linguagem GLSL (*OpenGL Shading Language*). Os *shaders* são executados diretamente no processador de renderização de gráficos em tempo real, chamado de GPU (*Graphics Processing Unit*). O GPU está cada vez mais presente em aplicações de propósito geral por possuir desempenho significativo no processamento paralelo de vértices (*Vertex Shader*) e fragmentos, chamado também de *pixels*, (*Fragment Shader*).

Pipeline é um conjunto de funções que são chamadas pela biblioteca OpenGL para transformar cenas tridimensionais em bidimensionais para apresentar no monitor. O processamento do *pipeline* gráfico se dá através do envio de primitivas geométricas, normalmente coordenadas tridimensionais de triângulos, para a GPU (OWENS [8]). A aplicação, por meio da API OpenGL, envia os dados de vértices e texturas para o processamento de primitivas. Esses dados são agrupados em matrizes e adicionados a um objeto Vertex buffer e enviados em blocos para a GPU. Todo o processamento é acionado pela execução da função Draw arrays que envia as matrizes através do *pipeline*. Após os vértices que são enviados para *vertex shader* para processamento geométrico. Vértices são montados em objetos e testado para inclusão no volume visualização antes de ser enviado para a fase de rasterização. Na rasterização são determinadas quais as áreas de pixels (fragmentos) da tela são cobertas (pintadas). A cor do pixel é calculada com base nas características da cena, como sua posição e efeitos de iluminação. Esse processamento é realizado pelo *fragment shader*. Após o processamento do OpenGL a imagem é encaminhada para a tela ou para área de memória: *frame buffer*. A Figura 1.1 mostra esse processo.

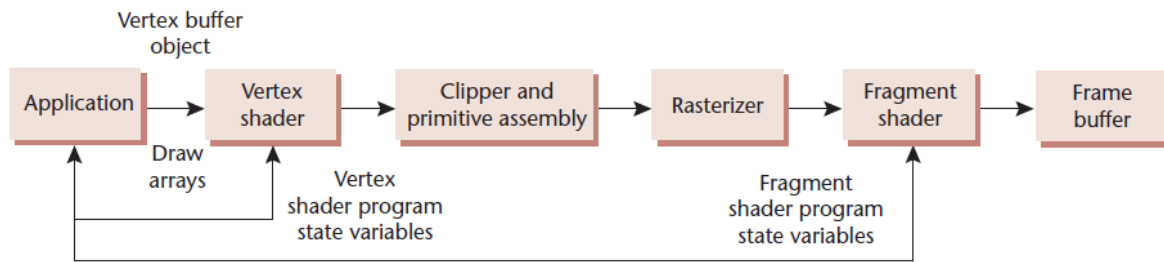


Figura 1.1- Envio de dados para GPU (ANGEL e SHREINER [7]).

O *vertex shader* é executado uma vez por vértice e sua função é efetuar operações matemáticas de transformações (rotação, translação e escala) dos vértices no espaço tridimensional. O *fragment shader* é executado uma vez por pixel e realiza cálculos de cor, iluminação e textura (ROSSANT [9]).

1.3 ROTEIRO DO TRABALHO

Esta dissertação é organizada como se segue: neste capítulo apresentou-se a introdução sobre o assunto que é abordado no trabalho, a justificativa, os objetivos e a revisão bibliográfica básica.

O Capítulo 2 mostra a fundamentação teórica para o desenvolvimento do modelo dinâmico de estruturas flexíveis do tipo que cabo, bem como dos algoritmos genéricos de ZANELA [1] para um número qualquer de elos escolhido.

No Capítulo 3 apresenta-se o problema que envolve a dissertação, a metodologia de desenvolvimento adotada para a modelagem do software, bem como a linguagem de desenvolvimento e as bibliotecas utilizadas para o desenvolvimento do aplicativo.

O Capítulo 4 define todas as informações que viabilizam uma animação tridimensional com base em simulações de cabos umbilicais. Explica-se também a técnica de animação adotada para solucionar o problema do projeto e mostra-se o produto resultante das implementações realizadas neste trabalho.

No Capítulo 5 é possível se observar as animações resultantes de simulações realizadas com base nos algoritmos genéricos. Detalham-se os frames de cada simulação conforme se aumenta o número de elos em que o cabo umbilical é dividido.

O Capítulo 6 contém as conclusões do trabalho realizado, informando sua motivação, como foi desenvolvido, os resultados obtidos e os projetos futuros.

Três apêndices complementam o texto principal. O Apêndice A.1 apresenta as equações dos elementos das matrizes e vetores para o cabo dividido em três elos, enquanto o Apêndice A.2 mostra as equações dos elementos das matrizes e vetores para o cabo dividido em quatro elos. O Apêndice B explana conceitos referentes à visualização de objetos e cenas em um ambiente tridimensional.

2. ALGORITMOS GENÉRICOS

2.1 INTRODUÇÃO

Neste capítulo é apresentado o formalismo discreto, proposto por PEREIRA [2] em tese de doutorado, que obtém as equações de movimento de cabos umbilicais a partir do formalismo de Euler-Lagrange. Com base no formalismo discreto, ZANELA [1] desenvolveu os algoritmos genéricos, que independentemente do número de elos em que o cabo é dividido, permitem gerar automaticamente as equações dos modelos dinâmicos de estruturas flexíveis do tipo cabo. Esses algoritmos também serão abordados neste capítulo.

2.2 MODELAGEM ANALÍTICA DE ESTRUTURAS FLEXÍVEIS DO TIPO CABO

Para esta modelagem considera-se um cabo com a extremidade superior articulada a uma plataforma fixa e a extremidade inferior é conectada a uma massa m_c conforme mostra o esquemático mostrado na Figura 2.1.

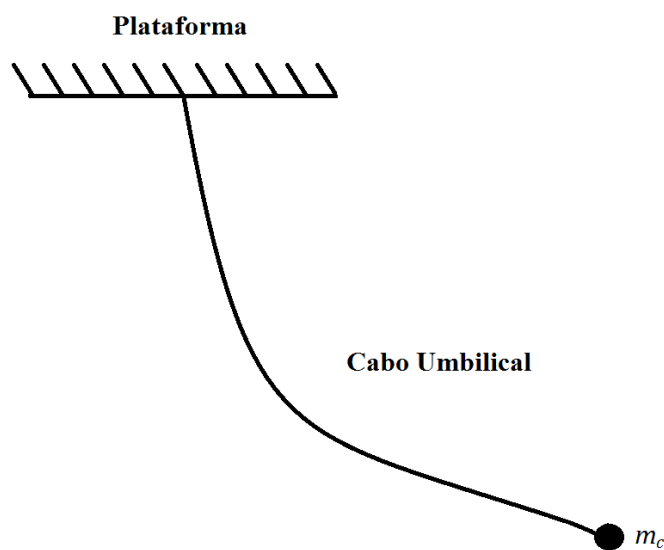


Figura 2.1- Representação esquemática do cabo umbilical conectado a uma massa m_c .

Outra situação prática envolvendo estruturas flexíveis em que cabos submersos podem ser utilizados ocorre quando a extremidade superior é articulada a uma base fixa e a outra é conectada a veículos subaquáticos não tripulados, conforme é mostrado na Figura 2.2.

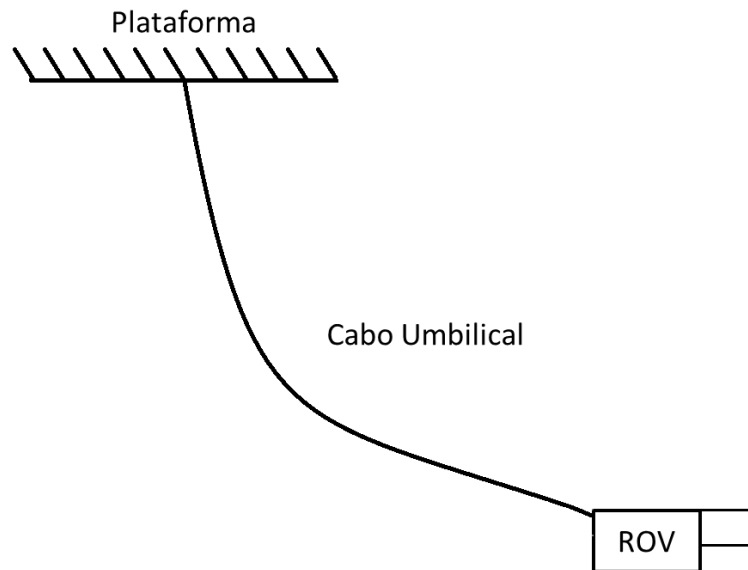


Figura 2.2- Representação esquemática do cabo umbilical conectado a um ROV.

Para os casos em que a extremidade superior do cabo é fixa a uma base, é necessário que a orientação e a posição da extremidade livre do cabo sejam conhecidas. A orientação é dada através de rotações e a posição por meio de translações.

A modelagem propõe que o cabo é formado por pequenas hastes rígidas, chamadas de elos, conectadas por articulações fictícias, ou juntas. Estas permitem três movimentos rotacionais livres e independentes, nomeados de azimute, elevação e torção, que são sempre relativos ao elo anterior da cadeia articulada (PEREIRA [2]).

Com o formalismo discreto, pode ser determinado o Lagrangeano do sistema de forma algorítmica, independente do número de elos que se deseja dividir o cabo. Na modelagem discreta, o cabo de flexibilidade contínua é dividido em n elos rígidos de comprimento $l_1, l_2, l_3, \dots, l_n$, conectados por articulações fictícias, conforme é mostrado na Figura 2.3.

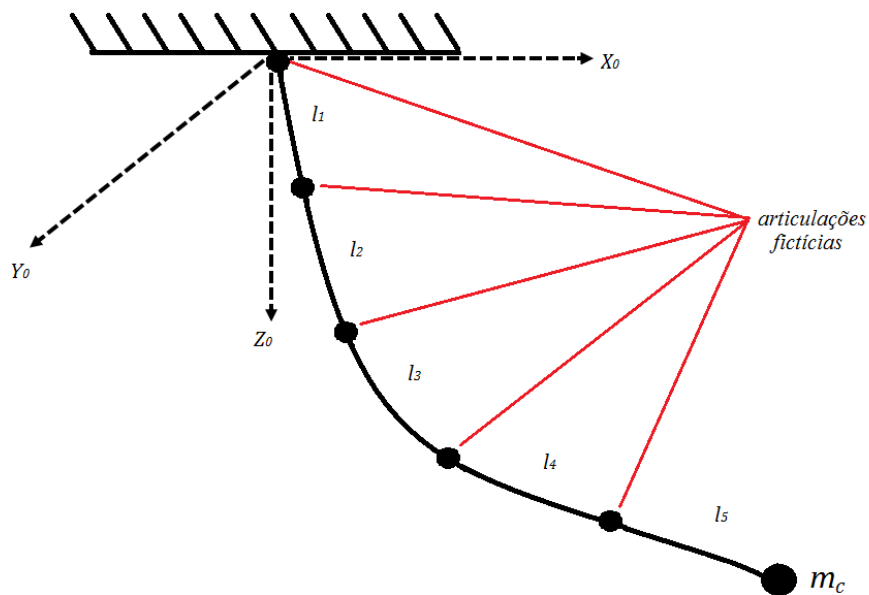


Figura 2.3 – Estrutura flexível e sua representação discreta.

Os elos possuem massas centradas nos seus centros de massas, ou seja, (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) , ..., (x_n, y_n, z_n) , com massas $m_1, m_2, m_3, \dots, m_n$, respectivamente, e (x_c, y_c, z_c) são as coordenadas do centro de massa da carga de massa m_c . Para cada junta i do cabo consideram-se os ângulos de azimute, elevação e torção, representados por $\theta_{ia}, \theta_{ie}, \theta_{iT}$, respectivamente, conforme a representação mostrada na Figura 2.4.

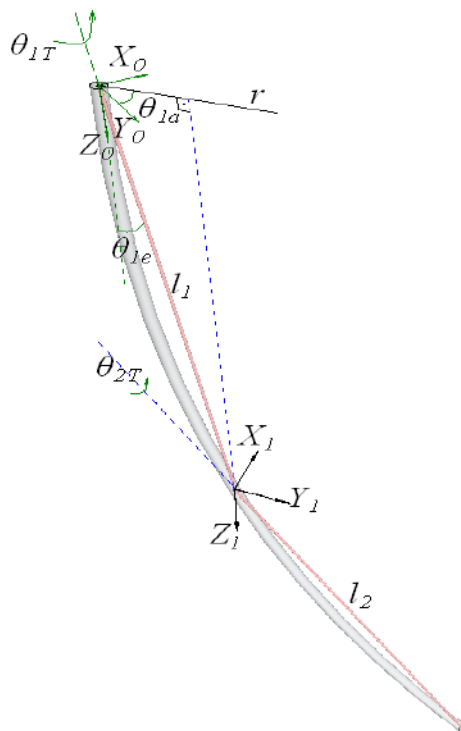


Figura 2.4 – Ângulos de azimute, elevação e torção (PEREIRA [1]).

A posição de cada elo é determinada a partir de um sistema de referência inicial na extremidade superior do cabo, chamado de sistema $X_0Y_0Z_0$, e um sistema de eixos coordenados para cada junta $X_iY_iZ_i$, com $i \in \mathbb{N}$ e $1 \leq i \leq n - 1$, de acordo com a Figura 2.5. O referencial $X_iY_iZ_i$ é obtido considerando que o eixo O_iZ_i é paralelo ao eixo OZ_0 do referencial inicial na direção vertical, o eixo O_iY_i é paralelo à projeção da parte rígida anterior a i -ésima articulação (projeção no plano horizontal) e o eixo O_iX_i é ortogonal ao eixo O_iY_i .

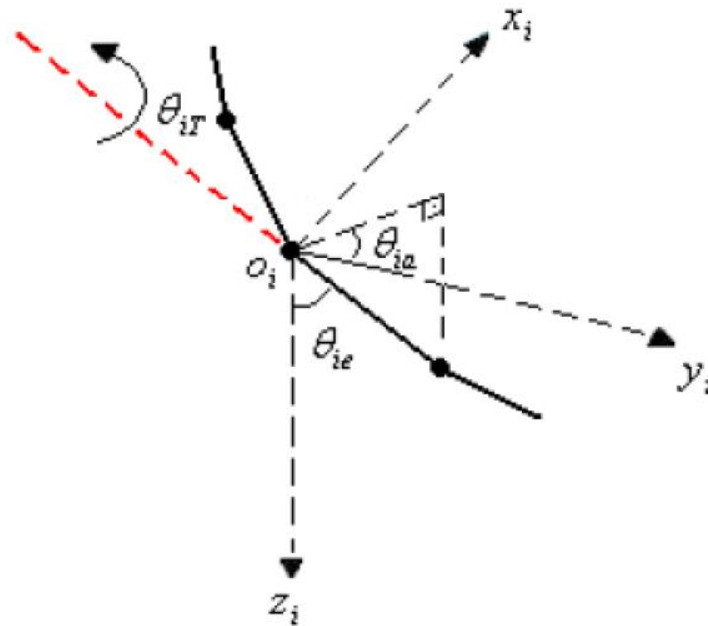


Figura 2.5 – Ângulos de azimute, elevação e torção na i -ésima articulação (ROCHA [10]).

A natureza física de cada articulação é definida por parâmetros que são constituídos por três constantes elásticas k_{ia} , k_{ie} e k_{it} que são referentes aos ângulos azimute, elevação e torção, respectivamente.

O Lagrangeano do sistema é dado por

$$L = E_C - E_P \quad (2.1)$$

onde L é o Lagrangeano do sistema, E_C é a energia cinética e E_P é a energia potencial.

A energia cinética é definida por

$$E_C = E_{C_R} + E_{C_T}, \quad (2.2)$$

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

A energia cinética devido ao movimento de rotação é definida pela Equação 2.3 como

$$\begin{aligned}
 E_{C_R} = & \frac{1}{2} I_{R_{1e}} \dot{\theta}_{1e}^2 + \frac{1}{2} I_{R_{2e}} \dot{\theta}_{2e}^2 + \frac{1}{2} I_{R_{3e}} \dot{\theta}_{3e}^2 + \dots + \frac{1}{2} I_{R_{ne}} \dot{\theta}_{ne}^2 + \\
 & + \frac{1}{2} I_{R_{1a}} \dot{\theta}_{1a}^2 + \frac{1}{2} I_{R_{2a}} \dot{\theta}_{2a}^2 + \frac{1}{2} I_{R_{3a}} \dot{\theta}_{3a}^2 + \dots + \frac{1}{2} I_{R_{na}} \dot{\theta}_{na}^2 + \\
 & + \frac{1}{2} I_{T_1} \dot{\theta}_{1T}^2 + \frac{1}{2} I_{T_2} \dot{\theta}_{2T}^2 + \frac{1}{2} I_{T_3} \dot{\theta}_{3T}^2 + \dots + \frac{1}{2} (I_{T_n} + I_{T_c}) \dot{\theta}_{nT}^2,
 \end{aligned} \tag{2.3}$$

onde $\theta_{1e}, \theta_{2e}, \theta_{3e}, \dots, \theta_{ne}$ são os ângulos de elevação, $\theta_{1a}, \theta_{2a}, \theta_{3a}, \dots, \theta_{na}$ são os ângulos de azimute, $\theta_{1T}, \theta_{2T}, \theta_{3T}, \dots, \theta_{nT}$ são os ângulos de torção, $I_{R_{1e}}, I_{R_{2e}}, I_{R_{3e}}, \dots, I_{R_{ne}}$ são os momentos de inércia relativos ao movimento de elevação, $I_{R_{1a}}, I_{R_{2a}}, I_{R_{3a}}, \dots, I_{R_{na}}$ são os momentos de inércia relativos ao movimento de azimute e $I_{T_1}, I_{T_2}, I_{T_3}, \dots, I_{T_n}, I_{T_c}$ são os momentos de inércia relativos aos movimentos de torção.

Nesse formalismo de modelagem a energia cinética rotacional é desprezada, pois a energia cinética de translação é muito mais importante que a rotacional.

A energia cinética devido ao movimento de translação é definida pela Equação 2.4,

$$\begin{aligned}
 E_{C_T} = & \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2) + \\
 & + \frac{1}{2} m_3 (\dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2) + \dots + \frac{1}{2} m_n (\dot{x}_n^2 + \dot{y}_n^2 + \dot{z}_n^2) + \\
 & + \frac{1}{2} m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2)
 \end{aligned} \tag{2.4}$$

onde $\frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2)$, $\frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2)$, $\frac{1}{2} m_3 (\dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2)$, ..., $\frac{1}{2} m_n (\dot{x}_n^2 + \dot{y}_n^2 + \dot{z}_n^2)$ e $\frac{1}{2} m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2)$ são as energias cinéticas relativas aos movimentos das massas $m_1, m_2, m_3, \dots, m_n$ dos elos e da massa m_c da carga terminal, respectivamente.

A energia potencial é definida por

$$\begin{aligned}
 E_P = & \frac{1}{2} k_{1e} \theta_{1e}^2 + \frac{1}{2} k_{2e} (\theta_{2e} - \theta_{1e})^2 + \frac{1}{2} k_{3e} (\theta_{3e} - \theta_{2e})^2 + \dots + \\
 & + \frac{1}{2} k_{ne} (\theta_{ne} - \theta_{(n-1)e})^2 + \frac{1}{2} k_{1a} \theta_{1a}^2 + \frac{1}{2} k_{2a} (\theta_{2a} - \theta_{1a})^2 + \\
 & + \frac{1}{2} k_{3a} (\theta_{3a} - \theta_{2a})^2 + \dots + \frac{1}{2} k_{na} (\theta_{na} - \theta_{(n-1)a})^2 + \frac{1}{2} k_{1T} \theta_{1T}^2 + \\
 & + \frac{1}{2} k_{2T} (\theta_{2T} - \theta_{1T})^2 + \frac{1}{2} k_{3T} (\theta_{3T} - \theta_{2T})^2 + \dots + \frac{1}{2} k_{nT} (\theta_{nT} - \theta_{(n-1)T})^2 +
 \end{aligned} \tag{2.5}$$

$$+ m_1gh_1 + m_2gh_2 + m_3gh_3 + \dots + m_ngh_n$$

onde $k_{1e}, k_{2e}, k_{3e}, \dots, k_{ne}$ são as constantes elásticas nas articulações referentes aos ângulos de elevação, $k_{1a}, k_{2a}, k_{3a}, \dots, k_{na}$ são constantes elásticas nas articulações referentes aos ângulos de azimute, $k_{1T}, k_{2T}, k_{3T}, \dots, k_{nT}$ são constantes elásticas nas articulações referentes aos ângulos de torção e $m_1gh_1 + m_2gh_2 + m_3gh_3 + \dots + m_ngh_n$ é a energia potencial gravitacional, sendo $h_1, h_2, h_3, \dots, h_n$ as alturas, que são definidas por

$$\begin{aligned} h_1 &= \frac{a_1}{2} \Rightarrow h_1 = \frac{l_1}{2}(1 - \cos \theta_{1e}), \\ h_2 &= a_1 + \frac{a_2}{2} \Rightarrow h_2 = a_1 + \frac{l_2}{2}(1 - \cos \theta_{2e}), \\ h_3 &= a_1 + a_2 + \frac{a_3}{2} \Rightarrow h_3 = a_1 + a_2 + \frac{l_3}{2}(1 - \cos \theta_{3e}), \\ &\vdots \\ h_n &= \sum_{i=1}^{n-1} a_i + \frac{a_n}{2} \Rightarrow h_n = \sum_{i=1}^{n-1} a_i + \frac{l_n}{2}(1 - \cos \theta_{ne}) \end{aligned} \quad (2.6)$$

sendo $a_0 = 0, a_1 = l_1(1 - \cos \theta_{1e}), a_2 = l_2(1 - \cos \theta_{2e}), \dots, a_n = l_n(1 - \cos \theta_{ne})$, como pode ser visto na Figura 2.6 a representação das alturas em função dos ângulos de elevação.

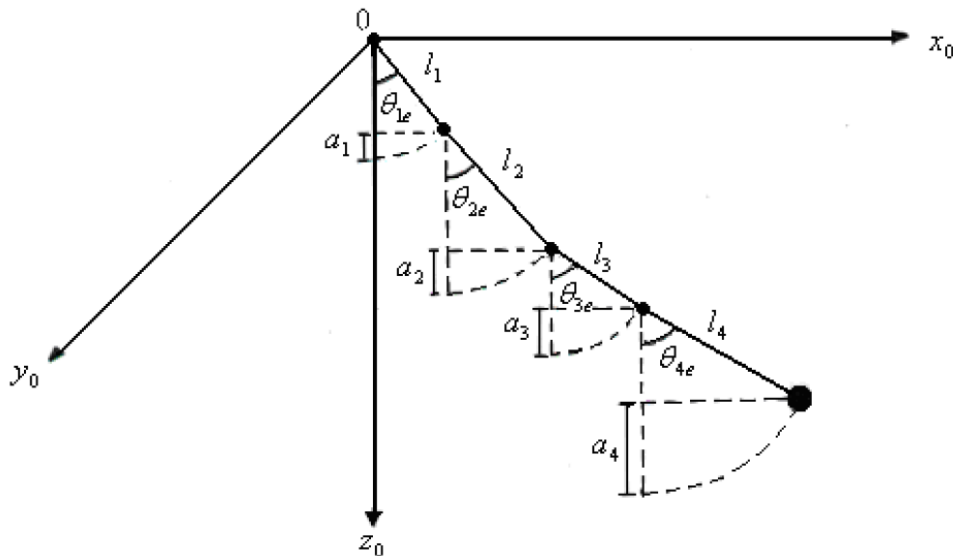


Figura 2.6 – Cabo com quatro (4) elos (PEREIRA [2]).

Substituindo (2.2), (2.3), (2.4), (2.5) e (2.6) no Lagrangeano (2.1), obtém-se

$$L = \frac{1}{2}I_{R_{1e}}\dot{\theta}_{1e}^2 + \frac{1}{2}I_{R_{2e}}\dot{\theta}_{2e}^2 + \frac{1}{2}I_{R_{3e}}\dot{\theta}_{3e}^2 + \dots + \frac{1}{2}I_{R_{ne}}\dot{\theta}_{ne}^2 + \quad (2.7)$$

$$\begin{aligned}
& + \frac{1}{2} I_{R_{1a}} \dot{\theta}_{1a}^2 + \frac{1}{2} I_{R_{2a}} \dot{\theta}_{2a}^2 + \frac{1}{2} I_{R_{3a}} \dot{\theta}_{3a}^2 + \dots + \frac{1}{2} I_{R_{na}} \dot{\theta}_{na}^2 + \\
& + \frac{1}{2} I_{T_1} \dot{\theta}_{1T}^2 + \frac{1}{2} I_{T_2} \dot{\theta}_{2T}^2 + \frac{1}{2} I_{T_3} \dot{\theta}_{3T}^2 + \dots + \frac{1}{2} (I_{T_n} + I_{T_c}) \dot{\theta}_{nT}^2 + \\
& + \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2) + \frac{1}{2} m_3 (\dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2) + \\
& + \dots + \frac{1}{2} m_n (\dot{x}_n^2 + \dot{y}_n^2 + \dot{z}_n^2) + \frac{1}{2} m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2) + \\
& - \frac{1}{2} k_{1e} \theta_{1e}^2 - \frac{1}{2} k_{2e} (\theta_{2e} - \theta_{1e})^2 - \frac{1}{2} k_{3e} (\theta_{3e} - \theta_{2e})^2 - \dots + \\
& - \frac{1}{2} k_{ne} (\theta_{ne} - \theta_{(n-1)e})^2 - \frac{1}{2} k_{1a} \theta_{1a}^2 - \frac{1}{2} k_{2a} (\theta_{2a} - \theta_{1a})^2 + \\
& - \frac{1}{2} k_{3a} (\theta_{3a} - \theta_{2a})^2 - \dots - \frac{1}{2} k_{na} (\theta_{na} - \theta_{(n-1)a})^2 - \frac{1}{2} k_{1T} \theta_{1T}^2 + \\
& - \frac{1}{2} k_{2T} (\theta_{2T} - \theta_{1T})^2 - \frac{1}{2} k_{3T} (\theta_{3T} - \theta_{2T})^2 - \dots - \frac{1}{2} k_{nT} (\theta_{nT} - \theta_{(n-1)T})^2 + \\
& - m_1 g \left[\frac{l_1}{2} (1 - \cos \theta_{1e}) \right] - m_2 g \left[l_1 (1 - \cos \theta_{1e}) + \frac{l_2}{2} (1 - \cos \theta_{2e}) \right] + \\
& - m_3 g \left[l_1 (1 - \cos \theta_{1e}) + l_2 (1 - \cos \theta_{2e}) + \frac{l_3}{2} (1 - \cos \theta_{3e}) \right] - \dots + \\
& - m_n g \left[\sum_{i=1}^{n-1} l_i (1 - \cos \theta_{ie}) + \frac{l_n}{2} (1 - \cos \theta_{ne}) \right]
\end{aligned}$$

De (2.7) e considerando $\theta_0 = 0$ e $\sum_{i=1}^{b-1} a_i = 0$, quando $b = 1$, tem-se

$$\begin{aligned}
L = \sum_{b=1}^n & \left\{ \frac{1}{2} I_{R_{be}} \dot{\theta}_{be}^2 + \frac{1}{2} I_{R_{ba}} \dot{\theta}_{ba}^2 + \frac{1}{2} I_{T_b} \dot{\theta}_{bT}^2 + \frac{1}{2} m_b (\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2) + \right. \\
& - \frac{1}{2} k_{be} (\theta_{be} - \theta_{(b-1)e})^2 - \frac{1}{2} k_{ba} (\theta_{ba} - \theta_{(b-1)a})^2 + \\
& \left. - \frac{1}{2} k_{bT} (\theta_{bT} - \theta_{(b-1)T})^2 - m_b g \left[\sum_{i=1}^{b-1} a_i + \frac{l_b}{2} (1 - \cos \theta_{be}) \right] \right\} + \\
& + \frac{1}{2} I_{T_c} \dot{\theta}_{nT}^2 + \frac{1}{2} m_c (\dot{x}_c^2 + \dot{y}_c^2 + \dot{z}_c^2)
\end{aligned} \tag{2.8}$$

Para determinar as coordenadas dos centros de massa dos elos e da carga terminal utilizam-se transformações homogêneas, sendo estas transformações descritas no Apêndice A da tese de PEREIRA [2]. A primeira articulação fictícia está localizada na origem do sistema de referência $X_0Y_0Z_0$ e os ângulos de elevação θ_{1e} e de azimute θ_{1a} são considerados conforme as Figura 2.4 e Figura 2.7.

As coordenadas da segunda articulação $(x_{0_1}, y_{0_1}, z_{0_1})$ em relação ao sistema de referência $X_0Y_0Z_0$ são obtidas da seguinte forma:

$$\begin{cases} x_{0_1} = l_1 \operatorname{sen} \theta_{1e} \operatorname{sen} \theta_{1a} \\ y_{0_1} = l_1 \operatorname{sen} \theta_{1e} \cos \theta_{1a} \\ z_{0_1} = l_1 \cos \theta_{1e} \end{cases} \quad (2.9)$$

As coordenadas do centro de massa do primeiro elo são:

$$\begin{cases} x_1 = \frac{l_1}{2} \operatorname{sen} \theta_{1e} \operatorname{sen} \theta_{1a} \\ y_1 = \frac{l_1}{2} \operatorname{sen} \theta_{1e} \cos \theta_{1a} \\ z_1 = \frac{l_1}{2} \cos \theta_{1e} \end{cases} \quad (2.10)$$

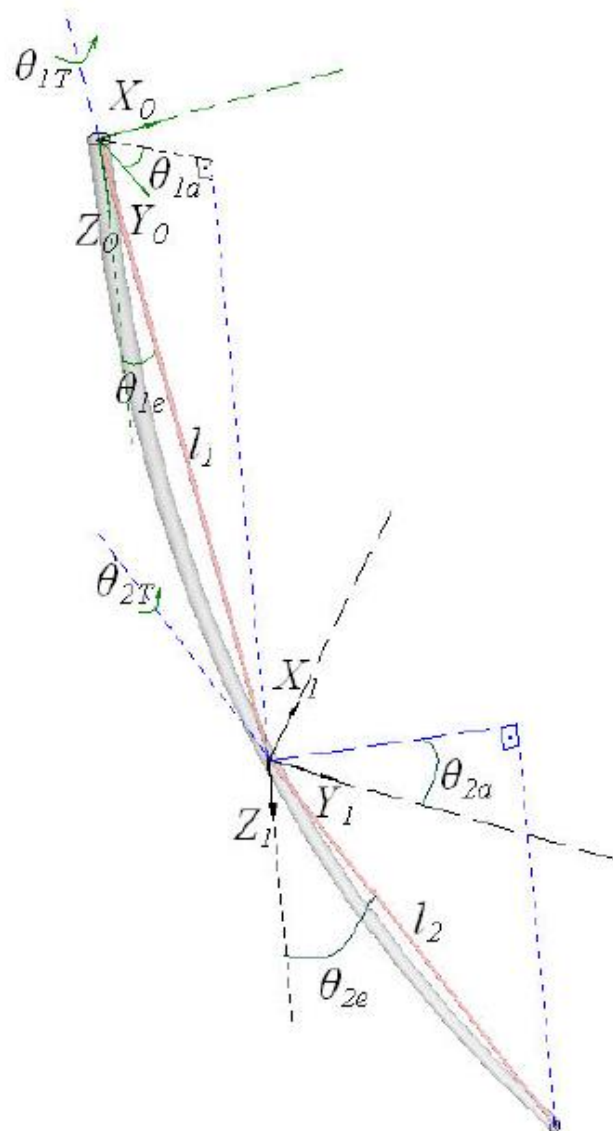


Figura 2.7 – Representação geométrica dos dois primeiros sistemas de referência (PEREIRA [2]).

De acordo com a Figura 2.7 a segunda articulação é adicionada a um novo sistema de referência $X_1Y_1Z_1$.

A matriz homogênea de rotação de um ângulo θ_{1a} em torno do eixo OZ_0 em relação ao referencial $X_0Y_0Z_0$ é definida por:

$$H_{0_1} = \begin{bmatrix} [R_{z,\theta_{1a}}]_{3 \times 3} & \begin{pmatrix} x_{0_1} \\ y_{0_1} \\ z_{0_1} \end{pmatrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

E a matriz de rotação de um ângulo θ em torno do eixo OZ é dada por:

$$R_{z,\theta} = \begin{bmatrix} \cos \theta & \text{sen } \theta & 0 \\ -\text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

A matriz de transformação homogênea que relaciona os sistemas $X_0Y_0Z_0$ e $X_1Y_1Z_1$, obtida considerando-se as equações (2.09), (2.11) e (2.12), tem a seguinte forma:

$$H_{0_1} = \begin{bmatrix} \cos \theta_{1a} & \text{sen } \theta_{1a} & 0 & l_1 \text{sen } \theta_{1e} \text{sen } \theta_{1a} \\ -\text{sen } \theta_{1a} & \cos \theta_{1a} & 0 & l_1 \text{sen } \theta_{1e} \cos \theta_{1a} \\ 0 & 0 & 1 & l_1 \cos \theta_{1e} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

Para o segundo elo considera-se o sistema de referência $X_2Y_2Z_2$ na terceira articulação, sendo, as coordenadas desta articulação em relação ao referencial $X_1Y_1Z_1$ da seguinte forma:

$$\begin{cases} x_{1_2} = l_2 \text{sen } \theta_{2e} \text{sen } \theta_{2a} \\ y_{1_2} = l_2 \text{sen } \theta_{2e} \cos \theta_{2a} \\ z_{1_2} = l_2 \cos \theta_{2e} \end{cases} \quad (2.14)$$

a matriz homogênea de rotação de um ângulo θ_{2a} em torno do eixo O_1Z_1 em relação ao referencial $X_1Y_1Z_1$ é definida por:

$$H_{1_2} = \begin{bmatrix} [R_{z,\theta_{2a}}]_{3 \times 3} & \begin{pmatrix} x_{1_2} \\ y_{1_2} \\ z_{1_2} \end{pmatrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

e a transformação homogênea que relaciona os sistemas $X_1Y_1Z_1$ e $X_2Y_2Z_2$ possui a seguinte forma:

$$H_{1_2} = \begin{bmatrix} \cos \theta_{2a} & \text{sen } \theta_{2a} & 0 & l_2 \text{ sen } \theta_{2e} \text{ sen } \theta_{2a} \\ -\text{sen } \theta_{2a} & \cos \theta_{2a} & 0 & l_2 \text{ sen } \theta_{2e} \cos \theta_{2a} \\ 0 & 0 & 1 & l_2 \cos \theta_{2e} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

Obtém-se a posição da terceira articulação fictícia e as coordenadas do centro de massa do segundo elo em relação ao referencial $X_0Y_0Z_0$ efetuando-se o produto das matrizes H_{0_1} e H_{1_2} , ou seja, $H_{0_2} = H_{0_1} \cdot H_{1_2}$, tendo H_{0_2} a seguinte forma:

$$\begin{bmatrix} \cos(\theta_{1a} + \theta_{2a}) & \text{sen}(\theta_{1a} + \theta_{2a}) & 0 & l_2 \text{ sen } \theta_{2e} \text{ sen}(\theta_{1a} + \theta_{2a}) + l_1 \text{ sen } \theta_{1e} \text{ sen } \theta_{1a} \\ -\text{sen}(\theta_{1a} + \theta_{2a}) & \cos(\theta_{1a} + \theta_{2a}) & 0 & l_2 \text{ sen } \theta_{2e} \cos(\theta_{1a} + \theta_{2a}) + l_1 \text{ sen } \theta_{1e} \cos \theta_{1a} \\ 0 & 0 & 1 & l_2 \cos \theta_{2e} + l_1 \cos \theta_{1e} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

As coordenadas do centro de massa do segundo elo em relação ao referencial $X_0Y_0Z_0$ são dadas por:

$$\begin{cases} x_2 = \frac{l_2}{2} \text{ sen } \theta_{2e} \text{ sen}(\theta_{1a} + \theta_{2a}) + l_1 \text{ sen } \theta_{1e} \text{ sen } \theta_{1a} \\ y_2 = \frac{l_2}{2} \text{ sen } \theta_{2e} \cos(\theta_{1a} + \theta_{2a}) + l_1 \text{ sen } \theta_{1e} \cos \theta_{1a} \\ z_2 = \frac{l_2}{2} \cos \theta_{2e} + l_1 \cos \theta_{1e} \end{cases} \quad (2.18)$$

Conforme se adicionam elos na modelagem, a matriz homogênea, as coordenadas do centro de massa da k -ésima parte rígida e as coordenadas da carga terminal seguem um padrão de crescimento indicando que é possível generalizar o processo de generalização dessas equações.

A matriz homogênea generalizada é definida como:

$$H_{0_k} = \begin{bmatrix} \cos\left(\sum_{i=1}^k \theta_{ia}\right) & \sin\left(\sum_{i=1}^k \theta_{ia}\right) & 0 & l_k \sin \theta_{ke} \sin\left(\sum_{i=1}^k \theta_{ia}\right) + \sum_{j=1}^{k-1} \left[l_j \sin \theta_{je} \sin\left(\sum_{i=1}^j \theta_{ia}\right) \right] \\ -\sin\left(\sum_{i=1}^k \theta_{ia}\right) & \cos\left(\sum_{i=1}^k \theta_{ia}\right) & 0 & l_k \sin \theta_{ke} \cos\left(\sum_{i=1}^k \theta_{ia}\right) + \sum_{j=1}^{k-1} \left[l_j \sin \theta_{je} \cos\left(\sum_{i=1}^j \theta_{ia}\right) \right] \\ 0 & 0 & 1 & l_k \cos \theta_{ke} + \sum_{j=1}^{k-1} l_j \cos \theta_{je} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

com $k = 1, 2, \dots, n$.

As coordenadas do centro de massa do k -ésimo elo são definidos da seguinte forma:

$$\begin{cases} x_k = \frac{l_k}{2} \sin \theta_{ke} \sin\left(\sum_{i=1}^k \theta_{ia}\right) + \sum_{j=1}^{k-1} \left[l_j \sin \theta_{je} \sin\left(\sum_{i=1}^j \theta_{ia}\right) \right] \\ y_k = \frac{l_k}{2} \sin \theta_{ke} \cos\left(\sum_{i=1}^k \theta_{ia}\right) + \sum_{j=1}^{k-1} \left[l_j \sin \theta_{je} \cos\left(\sum_{i=1}^j \theta_{ia}\right) \right] \\ z_k = \frac{l_k}{2} \cos \theta_{ke} + \sum_{j=1}^{k-1} l_j \cos \theta_{je} \end{cases} \quad (2.20)$$

com $k = 1, 2, \dots, n$.

As coordenadas da carga terminal são definidas como:

$$\begin{cases} x_c = \sum_{j=1}^n \left[l_j \sin \theta_{je} \sin\left(\sum_{i=1}^j \theta_{ia}\right) \right] \\ y_c = \sum_{j=1}^n \left[l_j \sin \theta_{je} \cos\left(\sum_{i=1}^j \theta_{ia}\right) \right] \\ z_c = \sum_{j=1}^n l_j \cos \theta_{je} \end{cases} \quad (2.21)$$

As derivadas das coordenadas (2.20) e (2.21) em relação ao tempo, são dadas, respectivamente, por:

$$\left\{ \begin{array}{l} \dot{x}_k = \frac{l_k}{2} \left[\text{sen } \theta_{ke} \cos \left(\sum_{i=1}^k \theta_{ia} \right) \sum_{i=1}^k \dot{\theta}_{ia} + \cos \theta_{ke} \text{sen} \left(\sum_{i=1}^k \theta_{ia} \right) \dot{\theta}_{ke} \right] + \\ \quad + \sum_{j=1}^{k-1} l_j \left[\text{sen } \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \\ \dot{y}_k = \frac{l_k}{2} \left[-\text{sen } \theta_{ke} \text{sen} \left(\sum_{i=1}^k \theta_{ia} \right) \sum_{i=1}^k \dot{\theta}_{ia} + \cos \theta_{ke} \cos \left(\sum_{i=1}^k \theta_{ia} \right) \dot{\theta}_{ke} \right] + \\ \quad + \sum_{j=1}^{k-1} l_j \left[-\text{sen } \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \\ \dot{z}_k = -\frac{l_k}{2} \text{sen } \theta_{ke} \dot{\theta}_{ke} - \sum_{j=1}^{k-1} l_j \text{sen } \theta_{je} \dot{\theta}_{je} \end{array} \right. \quad (2.22)$$

e

$$\left\{ \begin{array}{l} \dot{x}_c = \sum_{j=1}^n l_j \left[\text{sen } \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \\ \dot{y}_c = \sum_{j=1}^n l_j \left[-\text{sen } \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \\ \dot{z}_c = -\sum_{j=1}^n l_j \text{sen } \theta_{je} \dot{\theta}_{je} \end{array} \right. \quad (2.23)$$

Para obter-se a expressão geral do Lagrangeano, substituem-se as coordenadas cartesianas (2.22) e (2.23) na equação (2.8), conforme segue:

$$L = L_1 + L_2 + L_3 + L_4 + L_5 \quad (2.24)$$

sendo:

$$\begin{aligned}
L_1 &= \sum_{b=1}^n \left\{ \frac{1}{2} I_{R_{be}} \dot{\theta}_{be}^2 + \frac{1}{2} I_{R_{ba}} \dot{\theta}_{ba}^2 + \frac{1}{2} I_{T_b} \dot{\theta}_{bT}^2 - \frac{1}{2} k_{be} (\theta_{be} - \theta_{(b-1)e})^2 + \right. \\
&\quad \left. - \frac{1}{2} k_{ba} (\theta_{ba} - \theta_{(b-1)a})^2 - \frac{1}{2} k_{bT} (\theta_{bT} - \theta_{(b-1)T})^2 + \right. \\
&\quad \left. - m_b g \left[\sum_{i=1}^{b-1} a_i + \frac{l_b}{2} (1 - \cos \theta_{be}) \right] \right\} \\
L_2 &= \sum_{b=1}^n \frac{1}{2} m_b \left\{ \frac{l_b}{2} \left[\text{sen } \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] + \right. \\
&\quad \left. + \sum_{j=1}^{b-1} l_j \left[\text{sen } \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \right\}^2 \\
L_3 &= \sum_{b=1}^n \frac{1}{2} m_b \left\{ \frac{l_b}{2} \left[-\text{sen } \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] + \right. \\
&\quad \left. + \sum_{j=1}^{b-1} l_j \left[-\text{sen } \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \right\}^2 \\
L_4 &= \sum_{b=1}^n \frac{1}{2} m_b \left[-\frac{l_b}{2} \text{sen } \theta_{be} \dot{\theta}_{be} - \sum_{j=1}^{b-1} l_j \text{sen } \theta_{je} \dot{\theta}_{je} \right]^2 + \frac{1}{2} I_{T_c} \dot{\theta}_{nT}^2
\end{aligned}$$

e

$$\begin{aligned}
L_5 &= \frac{1}{2} m_c \left\{ \sum_{b=1}^n l_b \left[\text{sen } \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] \right\}^2 + \\
&\quad + \frac{1}{2} m_c \left\{ \sum_{b=1}^n l_b \left[-\text{sen } \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] \right\}^2 + \\
&\quad + \frac{1}{2} m_c \left[-\sum_{b=1}^n l_b \text{sen } \theta_{be} \dot{\theta}_{be} \right]^2
\end{aligned}$$

onde $\theta_0 = 0$ e $\sum_{i=1}^{b-1} a_i = 0$, quando $b = 1$.

2.3 CABO COM TRÊS ELOS

Para a modelagem dinâmica da estrutura flexível dividida em três elos considera-se um cabo com a extremidade superior articulada a uma estrutura fixa e a extremidade inferior conectada a uma massa m_c . O cabo de flexibilidade contínua é dividido em três partes rígidas de comprimento l_1, l_2 e l_3 , conectados por articulações fictícias, conforme Figura 2.8.

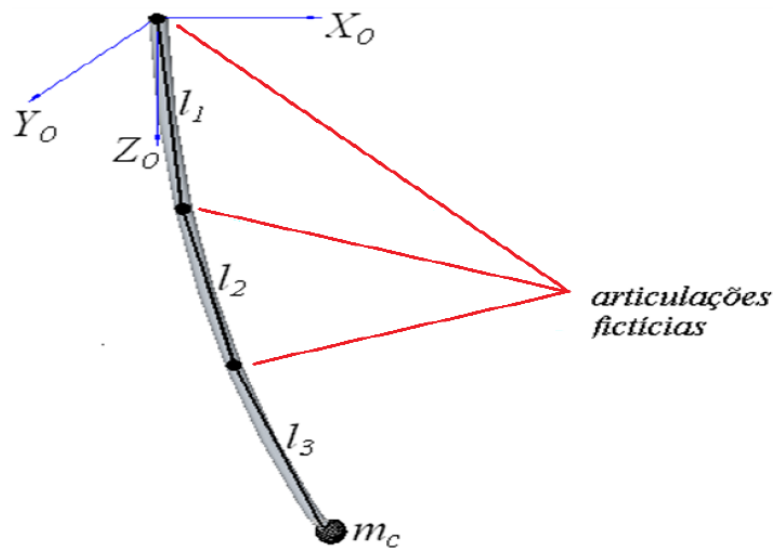


Figura 2.8 – Representação geométrica do cabo dividido em quatro articulações fictícias (ZANELA [1])

A obtenção do Lagrangeano é dada por (2.24):

$$L = L_1 + L_2 + L_3 + L_4 + L_5 ,$$

sendo:

$$L_1 = \sum_{b=1}^3 \left\{ \frac{1}{2} I_{R_{be}} \dot{\theta}_{be}^2 + \frac{1}{2} I_{R_{ba}} \dot{\theta}_{ba}^2 + \frac{1}{2} I_{T_b} \dot{\theta}_{bT}^2 - \frac{1}{2} k_{be} (\theta_{be} - \theta_{(b-1)e})^2 + \right. \\ \left. - \frac{1}{2} k_{ba} (\theta_{ba} - \theta_{(b-1)a})^2 - \frac{1}{2} k_{bT} (\theta_{bT} - \theta_{(b-1)T})^2 + \right. \\ \left. - m_b g \left[\sum_{i=1}^{b-1} a_i + \frac{l_b}{2} (1 - \cos \theta_{be}) \right] \right\}$$

$$\begin{aligned}
L_2 &= \sum_{b=1}^3 \frac{1}{2} m_b \left\{ \frac{l_b}{2} \left[\text{sen } \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] + \right. \\
&\quad \left. + \sum_{j=1}^{b-1} l_j \left[\text{sen } \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \right\}^2 \\
L_3 &= \sum_{b=1}^3 \frac{1}{2} m_b \left\{ \frac{l_b}{2} \left[-\text{sen } \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] + \right. \\
&\quad \left. + \sum_{j=1}^{b-1} l_j \left[-\text{sen } \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \right\}^2 \quad (2.25) \\
L_4 &= \sum_{b=1}^3 \frac{1}{2} m_b \left[-\frac{l_b}{2} \text{sen } \theta_{be} \dot{\theta}_{be} - \sum_{j=1}^{b-1} l_j \text{sen } \theta_{je} \dot{\theta}_{je} \right]^2 + \frac{1}{2} I_{Tc} \dot{\theta}_{4T}^2,
\end{aligned}$$

e

$$\begin{aligned}
L_5 &= \frac{1}{2} m_c \left\{ \sum_{b=1}^3 l_b \left[\text{sen } \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] \right\}^2 + \\
&\quad + \frac{1}{2} m_c \left\{ \sum_{b=1}^3 l_b \left[-\text{sen } \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] \right\}^2 + \\
&\quad + \frac{1}{2} m_c \left[-\sum_{b=1}^3 l_b \text{sen } \theta_{be} \dot{\theta}_{be} \right]^2
\end{aligned}$$

onde $\theta_0 = 0$ e $\sum_{i=1}^{b-1} a_i = 0$, quando $b = 1$.

Aplicando-se a equação de Euler-Lagrange a cada uma das variáveis do Lagrangeano, tem-se as seguintes equações:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{1e}} \right) - \frac{\partial L}{\partial \theta_{1e}} = T_{\theta_{1e}} - (c_{1e} + c_{2e}) \dot{\theta}_{1e} + c_{2e} \dot{\theta}_{2e} \quad (2.26)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{2e}} \right) - \frac{\partial L}{\partial \theta_{2e}} = T_{\theta_{2e}} + c_{2e} \dot{\theta}_{1e} - (c_{2e} + c_{3e}) \dot{\theta}_{2e} + c_{3e} \dot{\theta}_{3e} \quad (2.27)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{3e}} \right) - \frac{\partial L}{\partial \theta_{3e}} = T_{\theta_{3e}} - c_{3e} \dot{\theta}_{3e} + c_{3e} \dot{\theta}_{2e} \quad (2.28)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{1a}} \right) - \frac{\partial L}{\partial \theta_{1a}} = T_{\theta_{1a}} - (c_{1a} + c_{2a}) \dot{\theta}_{1a} + c_{2a} \dot{\theta}_{2a} \quad (2.29)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{2a}} \right) - \frac{\partial L}{\partial \theta_{2a}} = T_{\theta_{2a}} + c_{2a} \dot{\theta}_{1a} - (c_{2a} + c_{3a}) \dot{\theta}_{2a} + c_{3a} \dot{\theta}_{3a} \quad (2.30)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{3a}} \right) - \frac{\partial L}{\partial \theta_{3a}} = T_{\theta_{3a}} - c_{3a} \dot{\theta}_{3a} + c_{3a} \dot{\theta}_{2a} \quad (2.31)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{1T}} \right) - \frac{\partial L}{\partial \theta_{1T}} = T_{\theta_{1T}} - (c_{1T} + c_{2T}) \dot{\theta}_{1T} + c_{2T} \dot{\theta}_{2T} \quad (2.32)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{2T}} \right) - \frac{\partial L}{\partial \theta_{2T}} = T_{\theta_{2T}} + c_{2T} \dot{\theta}_{1T} - (c_{2T} + c_{3T}) \dot{\theta}_{2T} + c_{3T} \dot{\theta}_{3T} \quad (2.33)$$

e

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{3T}} \right) - \frac{\partial L}{\partial \theta_{3T}} = T_{\theta_{3T}} - c_{3T} \dot{\theta}_{3T} + c_{3T} \dot{\theta}_{2T} \quad (2.34)$$

onde c_{1e}, c_{2e} e c_{3e} , são os coeficientes de atrito devido ao movimento angular de elevação, c_{1a}, c_{2a} e c_{3a} , os coeficientes de atrito de azimute, c_{1T}, c_{2T} e c_{3T} , os coeficientes de atrito de torção, $T_{\theta_{ie}}, T_{\theta_{ia}}$ e $T_{\theta_{iT}}$, com $i = 1, 2$ e 3 , são os torques externos atuantes em cada articulação fictícia em azimute, elevação e torção, respectivamente. Os torques externos são independentes da carga terminal e surgem por diversos motivos, como por exemplo, correntes marítimas e força de empuxo.

Aplicando-se as equações (2.26) a (2.34), obtém-se o seguinte sistema de equações diferenciais de segunda ordem:

$$I(\vec{\theta}) \ddot{\vec{\theta}} + C \dot{\vec{\theta}} + K \vec{\theta} + \vec{f}(\vec{\theta}, \dot{\vec{\theta}}) + \vec{G}(\vec{\theta}, \dot{\vec{\theta}}) = \vec{T}_m \quad (2.35)$$

onde $\vec{\theta} = [\theta_{1e} \ \theta_{2e} \ \theta_{3e} \ \theta_{1a} \ \theta_{2a} \ \theta_{3a} \ \theta_{1T} \ \theta_{2T} \ \theta_{3T}]^T$. Em notação matricial, para o cabo dividido em 3 elos, o modelo dinâmico é definido por:

$$[I_{9 \times 9}] [\ddot{\vec{\theta}}_{9 \times 1}] + [C_{9 \times 9}] [\dot{\vec{\theta}}_{9 \times 1}] + [K_{9 \times 9}] [\vec{\theta}_{9 \times 1}] + [\vec{F}_{9 \times 1}] + [\vec{G}_{9 \times 1}] = [\vec{T}_{9 \times 1}] \quad (2.36)$$

onde I é a matriz de inércia, C é a matriz dos coeficientes de atrito, K é a matriz das constantes elásticas, \vec{F} é o vetor coriolis-centrífugo (esforços não-lineares), \vec{G} é o vetor gravitacional e \vec{T} é o vetor dos torques externos.

As equações que representam os elementos das matrizes e vetores do modelo dinâmico para o cabo dividido em 3 elos são expressas no Apêndice A.1.

A seguir mostram-se as matrizes de inércia I , dos coeficientes de atrito C e das constantes elásticas K , respectivamente:

$$I = \begin{bmatrix} I_{11} & I_{12} & I_{13} & I_{14} & I_{15} & I_{16} & 0 & 0 & 0 \\ I_{21} & I_{22} & I_{23} & I_{24} & I_{25} & I_{26} & 0 & 0 & 0 \\ I_{31} & I_{32} & I_{33} & I_{34} & I_{35} & 0 & 0 & 0 & 0 \\ I_{41} & I_{42} & I_{43} & I_{44} & I_{45} & I_{46} & 0 & 0 & 0 \\ I_{51} & I_{52} & I_{53} & I_{54} & I_{55} & I_{56} & 0 & 0 & 0 \\ I_{61} & I_{62} & I_{63} & I_{64} & I_{65} & I_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I_{77} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{88} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{99} \end{bmatrix},$$

$$C = \begin{bmatrix} C_{11} & C_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{32} & C_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & C_{45} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{54} & C_{55} & C_{56} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{65} & C_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{77} & C_{78} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{87} & C_{88} & C_{89} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{98} & C_{99} \end{bmatrix}$$

e

$$K = \begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ K_{21} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{32} & K_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{44} & K_{45} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{54} & K_{55} & K_{56} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{65} & K_{66} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_{77} & K_{78} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_{87} & K_{88} & K_{89} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{98} & K_{99} \end{bmatrix}$$

A seguir apresentam-se os vetores coriolis-centrífugo \vec{F} , gravitacional \vec{G} e de torques externos \vec{T} , respectivamente:

$$\vec{F} = [f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6 \quad 0 \quad 0 \quad 0]^T$$

$$\vec{G} = [g_1 \quad g_2 \quad g_3 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

e

$$\vec{T} = [T_{1e} \quad T_{2e} \quad T_{3e} \quad T_{1a} \quad T_{2a} \quad T_{3a} \quad T_{1T} \quad T_{2T} \quad T_{3T}]^T$$

2.4 CABO COM QUATRO ELOS

Para a modelagem dinâmica da estrutura flexível dividida em quatro elos considera-se um cabo com a extremidade superior articulada a uma estrutura fixa e a extremidade inferior conectada a uma massa m_c . O cabo de flexibilidade contínua é dividido em quatro partes rígidas de comprimento l_1, l_2, l_3 e l_4 , conectados por articulações fictícias, conforme Figura 2.9.

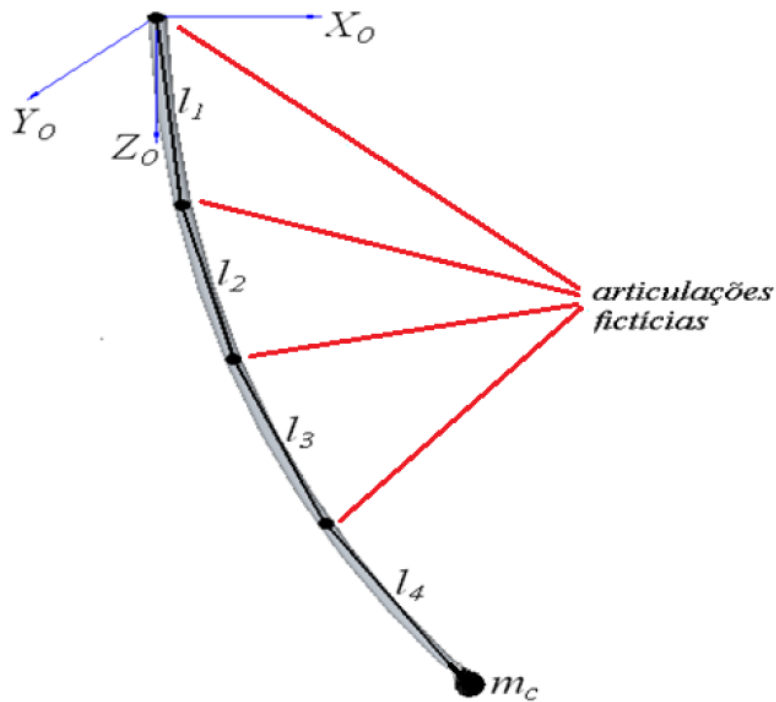


Figura 2.9 – Representação geométrica do cabo dividido em quatro articulações fictícias (ZANELA [1]).

A obtenção do Lagrangeano é dada por (2.24):

$$L = L_1 + L_2 + L_3 + L_4 + L_5$$

sendo:

$$L_1 = \sum_{b=1}^4 \left\{ \frac{1}{2} I_{R_{be}} \dot{\theta}_{be}^2 + \frac{1}{2} I_{R_{ba}} \dot{\theta}_{ba}^2 + \frac{1}{2} I_{T_b} \dot{\theta}_{bT}^2 - \frac{1}{2} k_{be} (\theta_{be} - \theta_{(b-1)e})^2 + \right. \\ \left. - \frac{1}{2} k_{ba} (\theta_{ba} - \theta_{(b-1)a})^2 - \frac{1}{2} k_{bT} (\theta_{bT} - \theta_{(b-1)T})^2 + \right. \\ \left. - m_b g \left[\sum_{i=1}^{b-1} a_i + \frac{l_b}{2} (1 - \cos \theta_{be}) \right] \right\}$$

$$L_2 = \sum_{b=1}^4 \frac{1}{2} m_b \left\{ \frac{l_b}{2} \left[\sin \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \sin \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] + \right. \\ \left. + \sum_{j=1}^{b-1} l_j \left[\sin \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \sin \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \right\}^2$$

$$\begin{aligned}
L_3 = & \sum_{b=1}^4 \frac{1}{2} m_b \left\{ \frac{l_b}{2} \left[-\text{sen } \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] + \right. \\
& \left. + \sum_{j=1}^{b-1} l_j \left[-\text{sen } \theta_{je} \text{sen} \left(\sum_{i=1}^j \theta_{ia} \right) \sum_{i=1}^j \dot{\theta}_{ia} + \cos \theta_{je} \cos \left(\sum_{i=1}^j \theta_{ia} \right) \dot{\theta}_{je} \right] \right\}^2 \quad (2.37) \\
L_4 = & \sum_{b=1}^4 \frac{1}{2} m_b \left[-\frac{l_b}{2} \text{sen } \theta_{be} \dot{\theta}_{be} - \sum_{j=1}^{b-1} l_j \text{sen } \theta_{je} \dot{\theta}_{je} \right]^2 + \frac{1}{2} I_{T_c} \dot{\theta}_{4T}^2,
\end{aligned}$$

e

$$\begin{aligned}
L_5 = & \frac{1}{2} m_c \left\{ \sum_{b=1}^4 l_b \left[\text{sen } \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] \right\}^2 + \\
& + \frac{1}{2} m_c \left\{ \sum_{b=1}^4 l_b \left[-\text{sen } \theta_{be} \text{sen} \left(\sum_{i=1}^b \theta_{ia} \right) \sum_{i=1}^b \dot{\theta}_{ia} + \cos \theta_{be} \cos \left(\sum_{i=1}^b \theta_{ia} \right) \dot{\theta}_{be} \right] \right\}^2 + \\
& + \frac{1}{2} m_c \left[-\sum_{b=1}^4 l_b \text{sen } \theta_{be} \dot{\theta}_{be} \right]^2
\end{aligned}$$

onde $\theta_0 = 0$ e $\sum_{i=1}^{b-1} a_i = 0$, quando $b = 1$.

Aplicando-se a equação de Euler-Lagrange a cada uma das variáveis do Lagrangeano, tem-se as seguintes equações:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{1e}} \right) - \frac{\partial L}{\partial \theta_{1e}} = T_{\theta_{1e}} - (c_{1e} + c_{2e}) \dot{\theta}_{1e} + c_{2e} \dot{\theta}_{2e} \quad (2.38)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{2e}} \right) - \frac{\partial L}{\partial \theta_{2e}} = T_{\theta_{2e}} + c_{2e} \dot{\theta}_{1e} - (c_{2e} + c_{3e}) \dot{\theta}_{2e} + c_{3e} \dot{\theta}_{3e} \quad (2.39)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{3e}} \right) - \frac{\partial L}{\partial \theta_{3e}} = T_{\theta_{3e}} - c_{3e} \dot{\theta}_{3e} + c_{3e} \dot{\theta}_{2e} + c_{4e} \dot{\theta}_{3e} - c_{4e} \dot{\theta}_{4e} \quad (2.40)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{4e}} \right) - \frac{\partial L}{\partial \theta_{4e}} = T_{\theta_{4e}} - c_{4e} \dot{\theta}_{3e} + c_{4e} \dot{\theta}_{4e} \quad (2.41)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{1a}} \right) - \frac{\partial L}{\partial \theta_{1a}} = T_{\theta_{1a}} - (c_{1a} + c_{2a})\dot{\theta}_{1a} + c_{2a}\dot{\theta}_{2a} \quad (2.42)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{2a}} \right) - \frac{\partial L}{\partial \theta_{2a}} = T_{\theta_{2a}} + c_{2a}\dot{\theta}_{1a} - (c_{2a} + c_{3a})\dot{\theta}_{2a} + c_{3a}\dot{\theta}_{3a} \quad (2.43)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{3a}} \right) - \frac{\partial L}{\partial \theta_{3a}} = T_{\theta_{3a}} - c_{3a}\dot{\theta}_{3a} + c_{3a}\dot{\theta}_{2a} - c_{4a}\dot{\theta}_{3a} + c_{4a}\dot{\theta}_{4a} \quad (2.44)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{4a}} \right) - \frac{\partial L}{\partial \theta_{4a}} = T_{\theta_{4a}} + c_{4a}\dot{\theta}_{3a} - c_{4a}\dot{\theta}_{4a} \quad (2.45)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{1T}} \right) - \frac{\partial L}{\partial \theta_{1T}} = T_{\theta_{1T}} - (c_{1T} + c_{2T})\dot{\theta}_{1T} + c_{2T}\dot{\theta}_{2T} \quad (2.46)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{2T}} \right) - \frac{\partial L}{\partial \theta_{2T}} = T_{\theta_{2T}} + c_{2T}\dot{\theta}_{1T} - (c_{2T} + c_{3T})\dot{\theta}_{2T} + c_{3T}\dot{\theta}_{3T} \quad (2.47)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{3T}} \right) - \frac{\partial L}{\partial \theta_{3T}} = T_{\theta_{3T}} - c_{3T}\dot{\theta}_{3T} + c_{3T}\dot{\theta}_{2T} - c_{4T}\dot{\theta}_{3T} + c_{4T}\dot{\theta}_{4T} \quad (2.48)$$

e

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{4T}} \right) - \frac{\partial L}{\partial \theta_{4T}} = T_{\theta_{4T}} + c_{4T}\dot{\theta}_{3T} - c_{4T}\dot{\theta}_{4T} \quad (2.49)$$

onde c_{1e}, c_{2e}, c_{3e} e c_{4e} , são os coeficientes de atrito devido ao movimento angular de elevação, c_{1a}, c_{2a}, c_{3a} e c_{4a} , os coeficientes de atrito de azimute, c_{1T}, c_{2T}, c_{3T} e c_{4T} , os coeficientes de atrito de torção, $T_{\theta_{ie}}, T_{\theta_{ia}}$ e $T_{\theta_{iT}}$, com $i = 1, 2, 3$ e 4 , são os torques externos atuantes em cada articulação fictícia em azimute, elevação e torção, respectivamente.

Aplicando-se as equações (2.38) a (2.49), obtém-se o seguinte sistema de equações diferenciais de segunda ordem:

$$I(\vec{\theta})\ddot{\vec{\theta}} + C\dot{\vec{\theta}} + K\vec{\theta} + \vec{f}(\vec{\theta}, \dot{\vec{\theta}}) + \vec{G}(\vec{\theta}, \dot{\vec{\theta}}) = \vec{T}_m \quad (2.50)$$

onde $\vec{\theta} = [\theta_{1e} \ \theta_{2e} \ \theta_{3e} \ \theta_{4e} \ \theta_{1a} \ \theta_{2a} \ \theta_{3a} \ \theta_{4a} \ \theta_{1T} \ \theta_{2T} \ \theta_{3T} \ \theta_{4T}]^T$. Em notação matricial, para o cabo dividido em 4 elos, o modelo dinâmico é definido por:

$$[I_{12 \times 12}] [\ddot{\theta}_{12 \times 1}] + [C_{12 \times 12}] [\dot{\theta}_{12 \times 1}] + [K_{12 \times 12}] [\theta_{12 \times 1}] + [\vec{F}_{12 \times 1}] + [\vec{G}_{12 \times 1}] = [\vec{T}_{12 \times 1}] \quad , (2.51)$$

onde I é a matriz de inércia, C é a matriz dos coeficientes de atrito, K é a matriz das constantes elásticas, \vec{F} é o vetor coriolis-centrífugo (esforços não-lineares), \vec{G} é o vetor gravitacional e \vec{T} é o vetor dos torques externos.

As equações que representam os elementos das matrizes e vetores do modelo dinâmico para o cabo dividido em 4 elos são expressas no Apêndice A.2.

A seguir mostram-se as matrizes de inércia I , dos coeficientes de atrito C e das constantes elásticas K , respectivamente:

$$I = \begin{bmatrix} I_{11} & I_{12} & I_{13} & I_{14} & I_{15} & I_{16} & I_{17} & I_{18} & 0 & 0 & 0 & 0 \\ I_{21} & I_{22} & I_{23} & I_{24} & I_{25} & I_{26} & I_{27} & I_{28} & 0 & 0 & 0 & 0 \\ I_{31} & I_{32} & I_{33} & I_{34} & I_{35} & I_{36} & I_{37} & I_{38} & 0 & 0 & 0 & 0 \\ I_{41} & I_{42} & I_{43} & I_{44} & I_{45} & I_{46} & I_{47} & 0 & 0 & 0 & 0 & 0 \\ I_{51} & I_{52} & I_{53} & I_{54} & I_{55} & I_{56} & I_{57} & I_{58} & 0 & 0 & 0 & 0 \\ I_{61} & I_{62} & I_{63} & I_{64} & I_{65} & I_{66} & I_{67} & I_{68} & 0 & 0 & 0 & 0 \\ I_{71} & I_{72} & I_{73} & I_{74} & I_{75} & I_{76} & I_{77} & I_{78} & 0 & 0 & 0 & 0 \\ I_{81} & I_{82} & I_{83} & 0 & I_{85} & I_{86} & I_{87} & I_{88} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{99} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{10,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{11,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I_{12,12} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{32} & C_{33} & C_{34} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{43} & C_{44} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & C_{56} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{65} & C_{66} & C_{67} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{76} & C_{77} & C_{78} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{87} & C_{88} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{99} & C_{9,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{10,9} & C_{10,10} & C_{10,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{11,10} & C_{11,11} & C_{11,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & C_{12,11} & C_{12,12} \end{bmatrix}$$

e

$$K = \begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ K_{21} & K_{22} & K_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_{32} & K_{33} & K_{34} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{43} & K_{44} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{55} & K_{56} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{65} & K_{66} & K_{67} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{76} & K_{77} & K_{78} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & K_{87} & K_{88} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{9,9} & K_{9,10} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{10,9} & K_{10,10} & K_{10,11} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{11,10} & K_{11,11} & K_{11,12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_{12,11} & K_{12,12} \end{bmatrix}$$

A seguir apresentam-se os vetores coriolis-centrífugo \vec{F} , gravitacional \vec{G} e de torques externos \vec{T} , respectivamente:

$$\vec{F} = [f_1 \quad f_2 \quad f_3 \quad f_4 \quad f_5 \quad f_6 \quad f_7 \quad f_8 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

$$\vec{G} = [g_1 \quad g_2 \quad g_3 \quad g_4 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$$

e

$$\vec{T} = [T_{1e} \quad T_{2e} \quad T_{3e} \quad T_{4e} \quad T_{1a} \quad T_{2a} \quad T_{3a} \quad T_{4a} \quad T_{1T} \quad T_{2T} \quad T_{3T} \quad T_{4T}]^T$$

2.5 ALGORITMOS GENÉRICOS

Em ZANELA [1] foram desenvolvidas as equações analíticas do modelo dinâmico para 2, 3 e 4 elos permitindo uma análise e identificação de um padrão de crescimento genérico das matrizes e vetores do sistema dinâmico. Essa análise permitiu o desenvolvimento de algoritmos genéricos que possibilitam gerar de forma automática o modelo dinâmico para um número qualquer de elos escolhidos para dividir o cabo. A seguir serão apresentados os algoritmos genéricos para as matrizes de inércia, das constantes elásticas e dos coeficientes de atrito e também os vetores coriolis-centrífugo e de esforços gravitacionais.

2.5.1 Algoritmo Genérico para a Matriz de Inércia

A matriz completa considerando-se o cabo dividido n elos, neste modelo dinâmico, possui o seguinte padrão de crescimento genérico:

$$M = \begin{bmatrix} I_e & N_e & T_e \\ I_a & N_a & T_a \\ I_t & N_t & T_t \end{bmatrix} \quad (2.52)$$

onde os índices e , a e t indicam elevação, azimute e torção, respectivamente, I , N e T são submatrizes quadradas de ordem n , sendo n o número de elos.

Como a matriz de inércia é simétrica, conclui-se que:

$$I_a = N_e^T, I_t = T_e^T \text{ e } N_t = T_a^T \quad (2.53)$$

Neste caso, é necessário obter-se apenas os algoritmos genéricos das submatrizes I_e , N_e , N_a , T_e , T_a e T_t .

2.5.1.1 Elementos das submatrizes T

Na tese de PEREIRA [2], o formalismo proposto considera os movimentos de torção desacoplados em relação aos movimentos de azimute e elevação, conseqüentemente as submatrizes T_e e T_a são nulas e de acordo com as igualdades de (2.53), as submatrizes I_t e N_t também são nulas.

Como a submatriz T_t é diagonal, tem-se:

$$T_t(i, i) = \sum_{k=i}^n J_{tk} \quad (2.54)$$

onde J_{tk} é o momento de inércia de rotação sobre o eixo longitudinal do elo k , com $i = 1, \dots, n$ e

$$T_t(n, n) = J_{tn} + J_{tc} \quad (2.55)$$

sendo J_{tc} é equivalente ao momento de inércia sobre o eixo longitudinal de rotação da carga terminal conectada à extremidade livre do cabo.

2.5.1.2 Elementos da submatriz I_e

Os elementos da submatriz I em relação à elevação (I_e) são obtidos conforme seguinte o algoritmo:

para $i = 1:n$,

para $j = i + 1:n$,

$$I_e(i, j) = l_i l_j \left(\frac{m_j}{2} + m_c + \sum_{k=j+1}^n m_k \right) \left[\cos \theta_{ie} \cos \theta_{je} \cos \left(\sum_{k=i+1}^j \theta_{ka} \right) + \right. \\ \left. + \sin \theta_{ie} \sin \theta_{je} \right]; \quad (2.56)$$

fim para,

$$I_e(i, i) = l_i^2 \left[\frac{m_i}{4} + m_c + \sum_{k=i+1}^n m_k \right];$$

fim para,

Observa-se que os elementos da diagonal principal da submatriz I_e coincidem com os elementos da diagonal principal da matriz completa M , ou seja, $I_e(i, i) = M(i, i)$ para $i = 1, \dots, n$. Nota-se ainda que a submatriz I_e é simétrica e, conseqüentemente, a equação (2.56) calcula apenas os elementos da diagonal principal e os que estão acima desta diagonal principal.

Caso queira-se considerar a parcela de energia cinética rotacional do movimento de elevação aos elementos da diagonal principal da submatriz I_e , acrescentam-se os momentos de inércia rotacionais sobre os eixos transversais aos elos e passando pelos seus respectivos centros de massa, conforme a seguinte equação:

$$I_e(i, i) = I_e(i, i) + I_{R_{ei}} \quad (2.57)$$

onde $I_{R_{ei}}$ é o momento de inércia rotacional de elevação relativo ao elo i .

2.5.1.3 Elementos da submatriz N_e

Em ZANELA [1] foi identificado que matriz completa M apresenta a seguinte característica:

$$N_e(n, n) = M(n, 2n) = 0$$

onde n é o número de elos.

Também foi verificado que a submatriz N_e possui duplicação de termos, conforme se observa a seguir:

$$N_e(i, i) = N_e(i, i + 1) \quad ,$$

sendo i o número de linhas da submatriz N_e .

O algoritmo para gerar os elementos da submatriz N_e é definido como:

```

para i = 1:n,
  para j = 1:n,
    se i ≠ j,
      S = 0;
      para k = j:n,
        se k ≠ i,
          se i > k,
            γ = 1;
          senão
            γ = -1;
          fim se,

          v = [ik];
          s1 = 0;
          para g = max(v) + 1:n,
            s1 = s1 + m(g);
          fim para,

          s2 = 0;
          para g = mix(v) + 1:max(v),
            s2 = s2 + θga;
          fim para,

          S = S + γlk [  $\frac{m_k}{2} + m_c + s_1$  ] cos(θie) sen(θke) sen(s2);
        fim se,

```

(2.58)

```

    fim para,
         $N_e(i, j) = l_i S;$ 
    fim se,
    fim para,
    fim para,

    para  $i = 1:n - 1,$ 
         $N_e(i, i) = N_e(i, i + 1);$ 
    fim para,

 $N_e(n, n) = 0;$ 

```

2.5.1.4 Elementos da submatriz N_a

Os elementos de N_a podem ser obtidos conforme o seguinte algoritmo:

```

    para  $i = 1:n,$ 
        para  $j = i:n,$ 

             $S1 = 0;$ 
            para  $k = j:n,$ 
                 $S1 = S1 + l_k^2 \left[ \frac{m_k}{4} + m_c + \sum_{h=k+1}^n m_h \right] \text{sen}^2(\theta_{ke});$ 
            fim para,

             $S2 = 0;$ 
             $v = [ij];$ 
            para  $k = \text{mix}(v) : n - 1,$ 

                 $S3 = 0;$ 
                para  $g = j:n,$ 
                    se  $g > k,$ 
                         $\beta = \text{calculabeta}(i, j, k);$ 
                         $S3 = S3 +$ 
                             $+ \beta l_g \left[ \frac{m_g}{2} + m_c + \sum_{h=g+1}^n m_h \right] \text{sen}(\theta_{ke}) \text{sen}(\theta_{ge}) \cos \left( \sum_{h=k+1}^g \theta_{ha} \right);$ 
                    fim se,
                fim para,

                 $S2 = S2 + l_k S3;$ 
            fim para,

            se  $(i = n) \text{ e } (j = n),$ 

```

(2.59)

$S2 = 0;$
fim se,

$N_a(i, j) = S1 + S2;$
 $N_a(j, i) = N_a(i, j);$

fim para,
fim para,

sendo $calculabeta(i, j, k)$ a função que calcula o parâmetro β e possui o seguinte algoritmo:

função $\beta = calculabeta(i, j, k)$
se $(i = 1)$ e $(j = 1)$ ou $(k \geq j)$,
 $\beta = 2;$ (2.60)
senão
 $\beta = 1;$
fim se,

A submatriz N_a é simétrica visto que sua diagonal principal pertence a diagonal da matriz M . Nota-se no algoritmo (2.59) que a i varia de 1 a n e j de i até n permitindo o cálculo dos elementos da diagonal principal e acima dela. Os elementos abaixo da diagonal principal são calculados no final do algoritmo na linha $N_a(j, i) = N_a(i, j);$.

Caso queira-se considerar a parcela de energia cinética rotacional do movimento de azimute aos elementos da diagonal principal da submatriz N_a , acrescentam-se os momentos de inércia rotacionais de azimute, considerados constantes neste formalismo:

$$N_a(i, i) = N_a(i, i) + I_{R_{ai}} \quad , \quad (2.61)$$

onde $I_{R_{ai}}$ é o momento de inércia rotacional de azimute relativo ao elo i .

2.5.2 Algoritmo Genérico para as Matrizes de Constantes Elásticas e de Coeficientes de Atrito

As matrizes de constantes elásticas K e de coeficientes de atrito C possuem as mesmas regras de geração e têm o seguinte formato:

$$K = \begin{bmatrix} K_e & 0 & 0 \\ 0 & K_a & 0 \\ 0 & 0 & K_t \end{bmatrix}; C = \begin{bmatrix} C_e & 0 & 0 \\ 0 & C_a & 0 \\ 0 & 0 & C_t \end{bmatrix}, \quad (2.62)$$

onde K_e , K_a e K_t são submatrizes de constantes elásticas relativas aos movimentos de elevação, azimute e torção, respectivamente, C_e , C_a e C_t são submatrizes de coeficientes de atrito relativos aos movimentos de elevação, azimute e torção, respectivamente.

Conforme visto em (2.62), as matrizes são diagonais por bloco e cada bloco é formado por submatrizes $n \times n$, sendo n o número de elos.

O algoritmo a seguir permite gerar os termos de cada uma das submatrizes de ambas as matrizes K e C :

```

para i = 1:n,
  para j = 1:n,
     $K_e(i,j) = 0;$ 

    se (i = j) e (i < n),
       $K_e(i,j) = K_{ei} + K_{ei+1};$ 
    fim se,

    se (i = j) e (i = n),
       $K_e(i,j) = K_{ei};$ 
    fim se,

    se j = i + 1,
       $K_e(i,j) = K_{ej};$ 
    fim se,

     $K_e(i,j) = K_e(j,i);$ 
  fim para,
fim para,

```

(2.63)

2.5.3 Algoritmo Genérico para o Vetor de Esforços Coriolis-centrifugo

Considera-se o vetor coriolis-centrífugo formado por $3n$ termos, sendo n o número de elos. Os primeiros n componentes são relativos aos movimentos de elevação, os n componentes intermediários são relativos aos movimentos de azimute e os últimos n componentes são relativos aos movimentos de torção. O vetor coriolis-centrífugo possui o seguinte formato:

$$\vec{F}(\vec{\theta}, \dot{\vec{\theta}}) = [f_1 \ f_2 \ \dots \ f_n \ f_{n+1} \ f_{n+2} \ \dots \ f_{2n} \ f_{2n+1} \ f_{2n+2} \ \dots \ f_{3n}]^T, \quad (2.64)$$

onde os elementos $[f_1 \ f_2 \ \dots \ f_n]^T$ referem-se aos movimentos de elevação, os elementos $[f_{n+1} \ f_{n+2} \ \dots \ f_{2n}]^T$ são referentes aos movimentos de azimute e $[f_{2n+1} \ f_{2n+2} \ \dots \ f_{3n}]^T$ são relativos aos movimentos de torção, sendo este último composto por elementos nulos. Considerando-se o exposto, serão apresentados os algoritmos genéricos para a determinação dos componentes de elevação e de azimute nas próximas subseções, respectivamente.

- Componentes de Elevação

Para determinar os elementos $[f_1 \ f_2 \ \dots \ f_n]^T$, considera-se que cada componente f_i da equação (2.64), com $i = 1, \dots, n$, é composto por dois termos, da seguinte forma:

$$f_i = v_1 + v_2. \quad (2.65)$$

O termo v_1 possui a seguinte equação:

$$v_1 = \alpha l_i^2 \left[\frac{m_i}{4} + m_c + \sum_{g=i+1}^n m_g \right] \text{sen}(\theta_{ie}) \cos(\theta_{ie}) \left\{ \left(\sum_{g=1}^{i-1} \dot{\theta}_{ga} \sum_{h=g+1}^i \dot{\theta}_{ha} \right) + \right. \\ \left. + \frac{1}{|\alpha|} \left(\sum_{g=1}^i \dot{\theta}_{ga}^2 \right) \right\}, \quad (2.66)$$

sendo α determinado na forma:

$$\begin{aligned} & \text{se } i = 1, \\ & \quad \alpha = -1; \\ & \text{senão} \\ & \quad \alpha = -2; \\ & \text{fim se,} \end{aligned} \quad (2.67)$$

com $i = 1, \dots, n$.

O vetor $\vec{z} = [z_1 \ z_2 \ \dots \ z_n]^T$ é definido a partir da seguinte equação:

$$\vec{z} = \vec{R} + W\vec{P} \quad (2.71)$$

sendo \vec{R} um vetor $n \times 1$ que contém termos funções dos l_i^2 , W é uma matriz $n \times q$ cujos elementos são funções dos ângulos e das velocidades angulares de azimute e elevação e \vec{P} é um vetor $q \times 1$ que contém produtos do tipo $l_i l_j$, para $j > i$. A variável n é o número de elos e q é obtido em função de n conforme o seguinte algoritmo:

$q = 0$;

para $i = 1:n$,

para $j = i + 1:n$,

$q = q + 1$;

fim para,

fim para,

2.72)

Cada termo R_i do vetor \vec{R} é dado por:

$$R_i = 2 \sum_{k=1}^n \left\{ l_k^2 \left(\frac{m_k}{4} + m_c + \sum_{g=k+1}^n m_g \right) \text{sen}(\theta_{ke}) \cos(\theta_{ke}) \left(\sum_{g=1}^k \dot{\theta}_{ga} \right) \dot{\theta}_{ke} \right\} \quad (2.73)'$$

com $i = 1, \dots, n$.

Os termos P_u do vetor \vec{P} possuem a seguinte forma:

$u = 0$;

para $i = 1:n$,

para $j = i + 1:n$,

$u = u + 1$;

$$P_u = 2l_i l_j \left[\frac{m_j}{2} + m_c + \sum_{g=j+1}^n m_g \right]; \quad (2.74)$$

fim para,

fim para,

com $u = 1, \dots, q$.

A matriz W é definida conforme segue:

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1q} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nq} \end{bmatrix} . \quad (2.75)$$

Cada elemento da matriz W é dado por:

$$w_{k,u} = a_{k,u} + b_{k,u} + c_{k,u} , \quad (2.76)$$

onde os elementos $a_{k,u}$, $b_{k,u}$ e $c_{k,u}$ são calculados a partir do seguinte algoritmo:

para $k = 1:n$,

$u = 0$;

para $i = 1:n - 1$,

para $j = i + 1:n$,

$u = u + 1$;

se $j \geq k$,

se $i < k$,

$$c_{k,u} = \text{sen}(\theta_{ie}) \text{sen}(\theta_{je}) \text{sen} \left(\sum_{g=i+1}^j \theta_{ga} \right) \vartheta_1 \left(\dot{\theta}_e, \dot{\theta}_a, i, j \right); \quad (2.77)$$

senão,

$$c_{k,u} = \text{sen}(\theta_{ie}) \text{sen}(\theta_{je}) \text{sen} \left(\sum_{g=i+1}^j \theta_{ga} \right) \vartheta_2 \left(\dot{\theta}_e, \dot{\theta}_a, i, j \right);$$

fim se,

senão,

$c_{k,u} = 0$;

fim se,

se $(k = 1)$ ou $(k \geq 2)$ e $(k < n)$ e $(i \geq k)$,

$$b_{k,u} = \text{sen}(\theta_{ie}) \cos(\theta_{je}) \cos\left(\sum_{g=i+1}^j \theta_{ga}\right) \left[\left(\sum_{g=1}^j \dot{\theta}_{ga}\right) \dot{\theta}_{je}\right];$$

senão,

$$b_{k,u} = 0;$$

fim se,

se $j > k$,

$$a_{k,u} = \cos(\theta_{ie}) \text{sen}(\theta_{je}) \cos\left(\sum_{g=i+1}^j \theta_{ga}\right) \left[\left(\sum_{g=1}^j \dot{\theta}_{ga}\right) \dot{\theta}_{ie}\right];$$

senão,

$$a_{k,u} = 0$$

fim se,

$$w_{k,u} = a_{k,u} + b_{k,u} + c_{k,u};$$

fim para,

fim para,

fim para,

onde $\vartheta_1(\dot{\theta}_e, \dot{\theta}_a, i, j)$ é calculado da seguinte forma:

$$\vartheta_1 = - \left\{ \left(\sum_{g=1}^{j-1} \dot{\theta}_{ga} \right) \left(\sum_{h=g+1}^j \dot{\theta}_{ha} \right) + \frac{1}{2} \left[\left(\sum_{g=i+1}^j \dot{\theta}_{ga}^2 \right) + \dot{\theta}_{je}^2 - \dot{\theta}_{ie}^2 \right] \right\} \quad (2.78)$$

e para obter-se $\vartheta_2(\dot{\theta}_e, \dot{\theta}_a, i, j)$, tem-se:

$$\vartheta_2 = - \left\{ \left(\sum_{g=1}^{i-1} \dot{\theta}_{ga} \right) \left(\sum_{h=g+1}^i \dot{\theta}_{ha} \right) + \frac{1}{2} \left[\left(\sum_{g=1}^i \dot{\theta}_{ga}^2 \right) + \dot{\theta}_{ie}^2 \right] \right\} \quad (2.79)$$

2.5.4 Algoritmo Genérico para o Vetor Gravitacional

O vetor gravitacional possui a seguinte forma:

$$\vec{G}(\vec{\theta}) = [g_1 \ g_2 \ \dots \ g_n \ g_{n+1} = 0 \ \dots \ g_{2n} = 0 \ g_{2n+1} = 0 \ \dots \ g_{3n} = 0]^T \quad (2.80)$$

Observa-se que apenas os primeiros n elementos apresentam valores diferentes de zero e são determinados de acordo com o seguinte algoritmo:

$$\begin{aligned}
 & \text{para } i = 1: n, \\
 & \quad g_i = l_i \left(\frac{m_i}{2} \sum_{k=i+1}^n m_k \right) g \operatorname{sen} \theta_{ie}; \\
 & \text{fim para,}
 \end{aligned} \tag{2.81}$$

2.6 SIMULAÇÃO NUMÉRICA

Foi realizada uma simulação considerando-se os mesmos parâmetros físicos adotados no trabalho de ZANELA [1]. Os parâmetros do modelo utilizados são mostrados na Tabela 2.1. Ressalta-se que estes parâmetros foram estipulados a priori, pois não se dispõe, até o presente momento, de um suporte experimental para a identificação paramétrica. No tempo inicial ($t = 0$), todas as coordenadas de posição e velocidade são inicializadas com zero. As simulações foram realizadas aplicando-se um torque de elevação igual a 80 Nm , um torque de torção igual a 20 Nm e um torque de azimute igual a 40 Nm , aplicado após os primeiros dois segundos. Todos os torques foram aplicados na última articulação, antes da carga terminal. O movimento é planar durante os primeiros dois segundos. Após este tempo, o torque em azimute é aplicado e o movimento do cabo torna-se tridimensional. Este tipo de simulação foi adotado porque é semelhante ao caso de um veículo subaquático do tipo ROV atuando como carga terminal e que aplica torques na última articulação. O cabo foi considerado submerso em água e, portanto, sujeito ao arrasto hidrodinâmico e às forças de empuxo. O arrasto hidrodinâmico é modelado de forma simples, ou seja, proporcional ao quadrado da velocidade relativa entre o fluido e a estrutura. A Figura 2.10 mostra os resultados dessa simulação. Os ângulos de elevação, azimute e torção são vistos nos três gráficos superiores. Os três gráficos intermediários mostram aspectos tridimensionais da evolução temporal desses ângulos, enquanto que os dois últimos gráficos mostram as coordenadas de posição em X, Y e Z, bem como a trajetória da carga terminal no espaço tridimensional.

Tabela 2.1 – Parâmetros físicos usado na simulação.

Parâmetros	Valores numéricos	Significado físico
L_c	24(m)	Comprimento do cabo
n	16	Número de elos
r_i	0.01(m)	Raio de cada elo (constante)
m_e	7850 (kg/m ³)	Massa específica do cabo
m_c	16 (kg)	Carga terminal
l_i	$\frac{L_c}{n}$ (m)	Comprimento de cada elo
m_i	$\pi r_i^2 l_i m_e$ (kg)	Massa de cada elo
I_{ie}	$\frac{m_i}{12} l_i^2$ (kgm ²)	Inércia rotacional de elevação
I_{ia}	$I_{ie}[(1 - \gamma) \sin\theta_{ie} + \gamma]$ (kgm ²)	Inércia rotacional de azimute
γ	0.1	Coefficiente adimensional
I_{iT}	$\frac{m_i}{2} r_i^2$ (kgm ²)	Inércia rotacional de torção
k_{ie}	$\frac{5 n E I_{cs}}{6 L_c} \left(\frac{Nm}{rd}\right)$	Constante elástica de elevação
E	8e10 (Nm ²)	Módulo de Young
I_{cs}	$\frac{1}{4} \pi r_i^4$ (m ⁴)	Momento de área de seção reta
k_{ia}	Zero	Constante elástica de azimute
k_{iT}	$2.5 K_{ie} \left(\frac{Nm}{rd}\right)$	Constante elástica de torção
c_e	$33n \left(\frac{Nms}{rd}\right)$	Coefficiente de atrito de elevação
c_a	$33n \left(\frac{Nms}{rd}\right)$	Coefficiente de atrito de azimute
c_t	$0.83n \left(\frac{Nms}{rd}\right)$	Coefficiente de atrito de torção
c_v	$200 \left(\frac{Ns^2}{m^2}\right)$	Coefficiente de arrasto hidrodinâmico
E_{ct}	0.95 $m_c g$ (N)	Empuxo na carga terminal

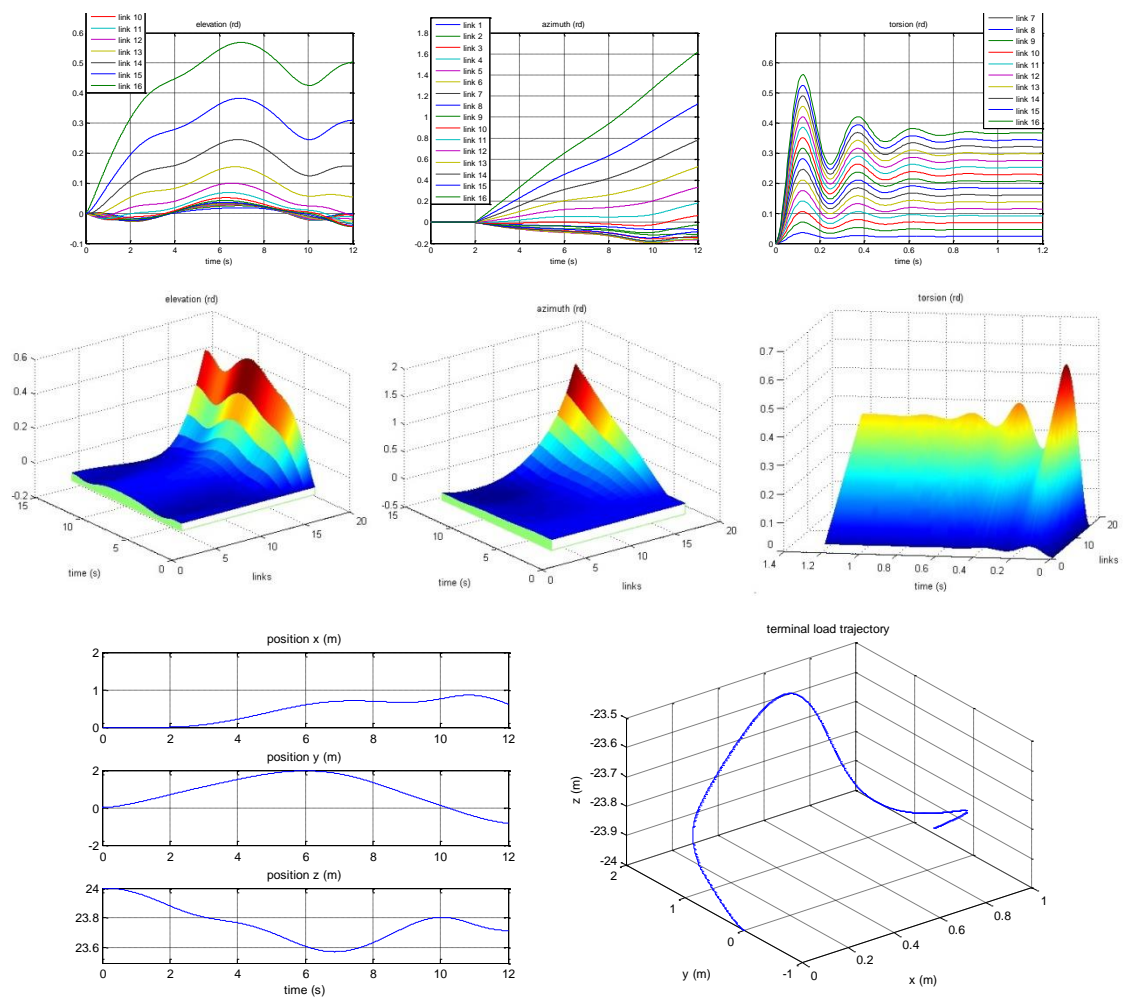


Figura 2.10 - Simulação considerando-se 16 elos para aproximar a flexibilidade contínua.

2.7 CONSIDERAÇÕES

Foi utilizado na dissertação de ZANELA [1] um aplicativo, desenvolvido usando um *toolbox* do MATLAB para manipulação simbólica de variáveis, para efetuar a validação dos resultados obtidos com os algoritmos genéricos. Maiores detalhes sobre a modelagem e a validação dos resultados podem ser verificadas em ZANELA [1]. No trabalho encontram-se também simulações utilizando os algoritmos.

Apesar de fornecerem informações interessantes sobre a simulação, os aspectos gráficos vistos na Figura 2.10 não permitem visualizar como se comportou dinamicamente todo o cabo no espaço tridimensional. Esta foi a principal razão que motivou o desenvolvimento do software de animação que se apresenta a partir do próximo capítulo.

3. METODOLOGIA E TECNOLOGIA DE DESENVOLVIMENTO

3.1 INTRODUÇÃO

No Capítulo 2 foi apresentada a modelagem dinâmica de estruturas flexíveis do tipo cabo considerando-se o formalismo de Euler-Lagrange como base para a criação do formalismo discreto. Na formulação abordada considera-se o cabo formado por pequenos elos rígidos conectados por articulações fictícias sendo que cada elo do cabo possui três movimentos livres: azimute, elevação e torção. Também foram apresentados os algoritmos genéricos que permitem gerar automaticamente as equações dos modelos dinâmicos para estruturas flexíveis para qualquer número de elos em que o cabo é dividido.

Esse capítulo descreve o problema principal motivador do trabalho, expõe a metodologia adotada e apresenta as ferramentas utilizadas para o desenvolvimento do *software*.

3.2 DESCRIÇÃO DO PROBLEMA

Como exposto na introdução do Capítulo 1, ZANELA [1] validou de forma comparativa seus algoritmos, utilizando um aplicativo que gera de forma automática o modelo dinâmico de cabos. O aplicativo, chamado de SIMMODEL CAB, foi desenvolvido por FABER e VIEIRA [3] e baseia-se na tese de doutorado de PEREIRA [2]. O SIMMODEL CAB foi criado utilizando-se um *toolbox* do MATLAB que efetua manipulação simbólica de variáveis, o qual tem seu tempo de processamento aumentado consideravelmente conforme se aumenta o número de elos do cabo. Devido ao crescimento do tempo de processamento, o SIMMODEL CAB não suporta a geração das equações para o cabo dividido com mais de seis elos, o que foi comprovado pelos travamentos do aplicativo ao realizar simulações com o número de elos igual ou superior a sete.

A implementação dos algoritmos genéricos foi realizada utilizando o programa MATLAB que possui um custo de licença alto. Para os algoritmos genéricos ainda não foi desenvolvida nenhuma solução para simulação com animação tridimensional, sendo este trabalho, o primeiro a implementar uma plataforma tridimensional com base nos algoritmos genéricos de ZANELA [1].

Antes de iniciar o desenvolvimento do *software*, chamado de CableSim, foi necessário determinar quais são os dados de entrada para então pensar em como efetuar o processamento e por fim gerar os resultados. Neste capítulo será dada ênfase no processo de obtenção dos dados de entrada através da importação dos mesmos gerados a partir dos algoritmos de ZANELA [1]. O processamento será apresentado no Capítulo 4 e os resultados, no Capítulo 5.

A animação do cabo ocorre identificando-se a posição de cada elo em cada instante de tempo da simulação, sendo que esta posição é determinada através de três movimentos livres: elevação, azimute e torção. Informações sobre o tempo da animação também são importantes para que se tenha uma aproximação entre o tempo da simulação e o tempo real.

Com base no disposto, criou-se um padrão para que o CableSim importe de forma correta os dados essenciais para realizar a simulação do arquivo no formato TXT. O padrão é constituído de colunas que estão diretamente relacionadas com a quantidade de elos e aos movimentos angulares, ou seja, as colunas do arquivo são formadas pela quantidade de elos multiplicada por três (número de ângulos dos movimentos livres, elevação, azimute e torção, respectivamente). O mesmo deve possuir também uma última coluna que se refere ao tempo em cada instante da simulação. A Figura 3.1 mostra a representação da estrutura do arquivo TXT com a ordem da disposição das colunas referente a uma linha do mesmo.

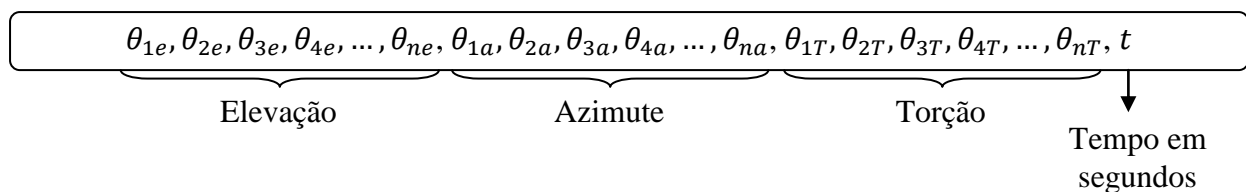


Figura 3.1 – Representação de uma linha do arquivo TXT.

A variável n é o número de elos em que o cabo é dividido. As primeiras colunas referem-se aos movimentos angulares de elevação, as colunas seguintes constam os movimentos angulares de azimute, em seguida estão dispostos os movimentos angulares de torção.

Já a quantidade de linhas do arquivo é utilizada para manipular os *frames* da animação, sendo que a última consta o tempo total da animação.

Considerando-se o ambiente e o problema exposto, na próxima seção é explicada a metodologia de desenvolvimento utilizada para dar início a solução do problema.

3.3 METODOLOGIA DE DESENVOLVIMENTO DO SOFTWARE

Definir uma metodologia para o desenvolvimento do *software* é uma fase inicial importante para promover qualidade do produto de *software* final. Portanto, a seguir apresentam-se as quatro etapas realizadas para a produção do aplicativo CableSim.

A primeira etapa foi a determinação do modelo de processo de *software* adequado para o problema que envolve o projeto. O modelo é uma representação resumida de um processo de *software*, o qual fornece informações parciais sobre o aplicativo (SOMMERVILLE [19]). Optou-se pelo desenvolvimento incremental que aborda intercaladamente, as atividades de especificação, desenvolvimento e validação. Nesse modelo são adicionadas funcionalidades a cada versão do sistema.

A segunda etapa adotada é a especificação dos requisitos do *software*, ou seja, determinar quais são as tarefas que o programa necessita realizar, os serviços que ele deve oferecer e as restrições referentes ao seu funcionamento (SOMMERVILLE [19]). Os requisitos descrevem as tarefas que o sistema deve realizar e podem ser em nível de usuário, ou seja, relatados de uma forma mais abstrata ou em um nível mais específico, geralmente bem mais detalhado e utilizado por profissionais da área de desenvolvimento de aplicativos. Abaixo observam-se os requisitos elencados para o CableSim, considerando-se que os requisitos foram descritos em nível de usuário por se tratar de um *software* acadêmico. O sistema deve:

- Possibilitar a importação dos dados gerados pelos algoritmos genéricos implementados no MATLAB.
- Ser capaz de efetuar animações de acordo com os dados informados pelo usuário juntamente com as informações importadas. Este requisito é de suma importância para o projeto, visto que se refere ao objetivo geral do mesmo.
- Possuir uma interface em que o usuário informe os dados adicionais da animação, tais como: tamanho do cabo, quantidade de elos e tempo de animação.
- Permitir a aproximação entre o tempo da animação e o tempo real decorrido.
- Gerar o gráfico referente ao deslocamento da carga terminal.

- Possibilitar a adição de coordenadas cartesianas em cada elo para uma melhor visualização do posicionamento do elo e o cabo como um todo.

A terceira etapa adotada trata de apresentar os casos de uso do aplicativo, que é uma forma simples de demonstrar as pessoas envolvidas e as interações com o aplicativo. Eles descrevem as funcionalidades de um sistema de *software* além de serem aplicados para captar o comportamento pretendido com o mesmo. E ainda facilitam a comunicação com o usuário solicitante do aplicativo, pois ao elaborar o diagrama não é necessário se preocupar com detalhes de implementações. Os casos de uso podem ser textuais ou em forma de diagramas e fazem parte da Linguagem de Modelagem Unificada (UML, do inglês *Unified Modelling Language*). Segundo BALZERT [20], UML é uma linguagem de notação gráfica usada para criar modelos de projeto de *software*. A UML possui diversos tipos de diagramas que auxiliam o desenvolvimento de vários tipos de sistemas de *software*. De acordo com a pesquisa de ERICKSON e SIAU [21], os seguintes quatro diagramas destacam-se na preferências dos usuários da linguagem UML por serem suficientes para demonstrar a essência de um aplicativo: classe, casos de uso, sequência e estado. Outros diagramas também são destacados como importantes no artigo, são eles: componentes, atividade, colaboração, implementação e objetos. Neste trabalho, por ser acadêmico, são elaborados os diagramas de casos de uso e de classes. A seguir é apresentado o diagrama de casos de uso do projeto e após fala-se sobre o diagrama de classe do trabalho.

Um caso de uso abrange a interação entre atores e sistemas e um ator representa um conjunto de papéis que o usuário do sistema de *software* desempenha na vida real (BOOCH [22]). Os casos de usos foram elaborados com o intuito de demonstrar que têm dois atores envolvidos no projeto (Figura 3.2), ou seja, o ator Usuário que é o responsável pela utilização do aplicativo e o MATLAB que é o ator externo, ele fornece os dados da animação.

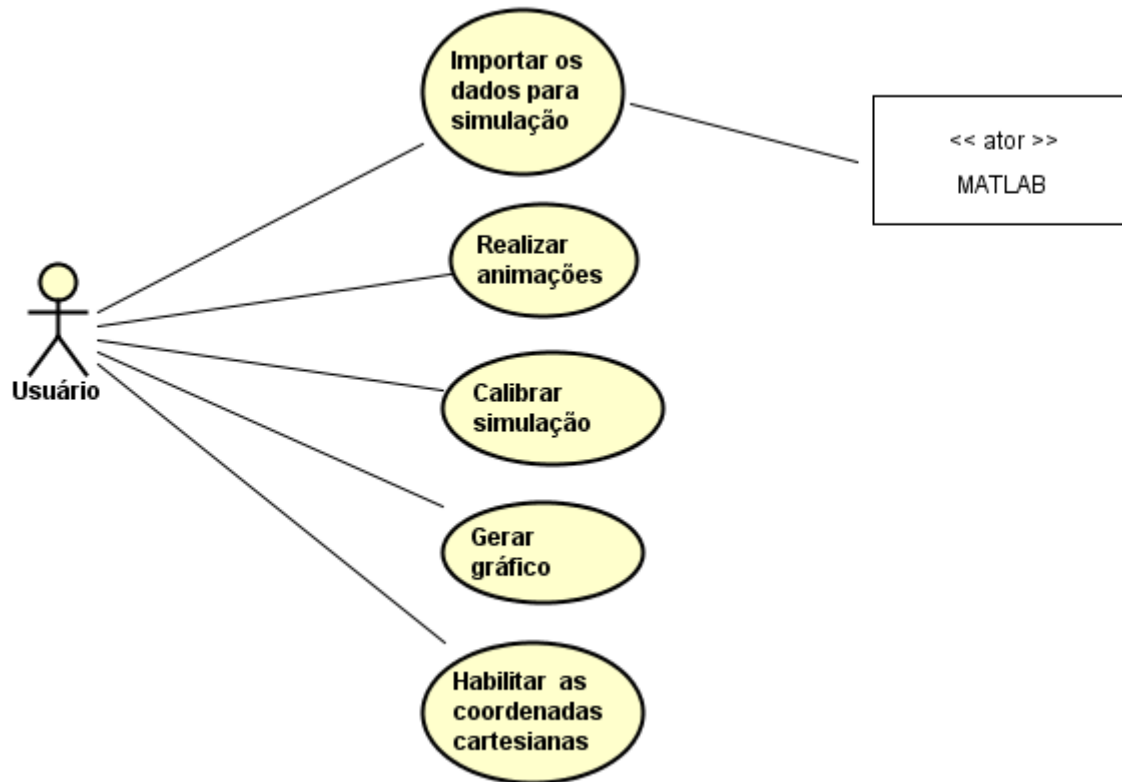


Figura 3.2 – Casos de uso do software CableSim.

A última etapa consiste no desenvolvimento do sistema utilizando-se o paradigma da orientação a objetos, visto que esse paradigma oferece diversas vantagens, como por exemplo, simplificar o desenvolvimento, facilitar a reutilização do código e agilizar a manutenção do mesmo. Nesse paradigma, um objeto possui atributos e operações que representam um artefato do mundo real a ser modelado e, conseqüentemente, simplifica o desenvolvimento do produto final, o aplicativo. Objetos são independentes se todas as operações realizadas nos atributos de um objeto forem programadas somente no próprio objeto, e este fato facilita a reutilização e a manutenção do código. O aplicativo resultante da utilização da orientação a objetos é formado por unidades menores e independentes que reduz o nível de complexidade do sistema tornando fácil e rápido o seu desenvolvimento e a sua manutenção (SCHACH [23]).

A base da orientação a objetos é a classe, pois objetos são criados a partir de classes, ou seja, eles são instâncias de classe e compartilham um conjunto de características semelhantes. As operações são os mesmos em cada instância, mas os atributos de cada objeto são diferentes (BARRY [24]). Uma classe define o comportamento (operações), a estrutura (atributos) e o relacionamento para uma coleção de objetos. Além do diagrama de casos de

uso da linguagem UML, optou-se por elaborar também o diagrama classes devido a importância dada ao diagrama no artigo de ERICKSON e SIAU [21]. O diagrama de classe mostra as classes e as associações entre essas classes e é muito usado para o desenvolvimento de um modelo de sistema orientado a objetos. As associações são ligações entre classes indicando um relacionamento entre elas. (SOMMERVILLE [19]). A Figura 3.3 mostra o diagrama de classe do sistema CableSim incluindo o diagrama de pacotes. O diagrama de pacotes permite agrupar, com o objetivo de organizar, as classes do projeto (LARMAN [25]), permitindo detalhar a estrutura do aplicativo a um nível acima das classes. As classes encontram-se representadas nos seguintes pacotes envolvidos no projeto: Qt, gui, world, cablesim e geometry.

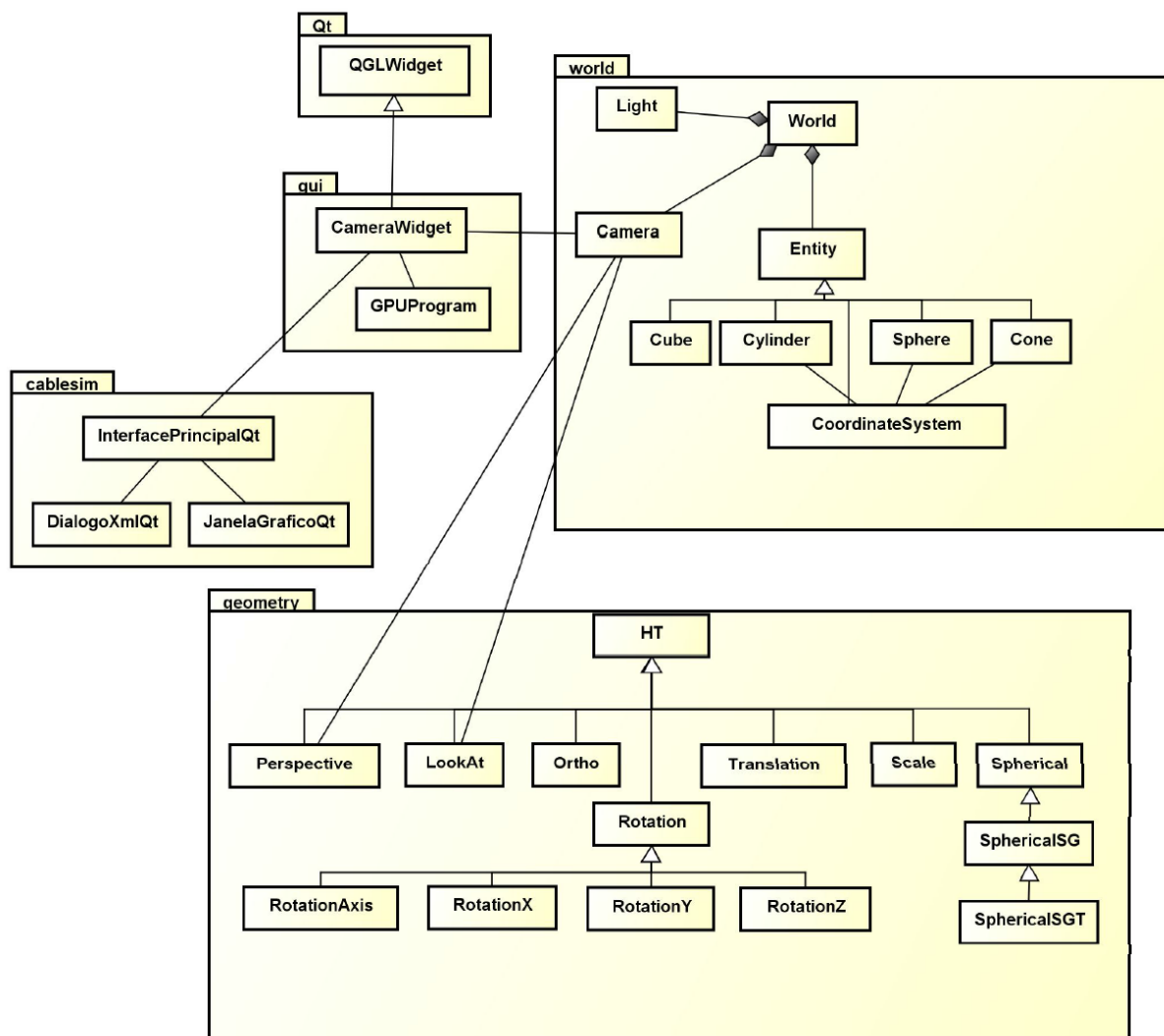


Figura 3.3 – Diagrama de classe do software CableSim.

Como já foi explicado nesta seção, as classes representam os objetos da realidade modelada e são compostas por atributos, operações (também chamados de métodos) e relacionamentos, sendo que o objetivo do diagrama de classe deste projeto é apresentar a visão geral do aplicativo (a Figura 3.3 mostra as classes com essas simplificações). A seguir são exploradas as classes organizadas por pacotes para situar melhor o leitor sobre o assunto em questão trabalhado.

3.3.1 Pacote cablesim

A classe `InterfacePrincipalQt` é responsável por viabilizar o acesso a todas as funcionalidades do aplicativo. Nela foi aplicado um dos mecanismos importantes da orientação a objetos, chamado de herança, que é implementado através da técnica de generalização. Na modelagem de sistemas a generalização ocorre no momento em que uma classe de nível inferior herda características comuns de outra classe de nível superior. Essas características são os atributos e operações. Essa classe herda as funcionalidades dos *widgets* do *framework* Qt utilizados para a criação da interface do usuário. O Qt é abordado na seção 3.4 e as funcionalidades bem como os demais métodos são apresentadas no capítulo 4 por se tratarem de implementações do software.

3.3.2 Pacotes gui e Qt

Neste pacote estão presentes as classes `CameraWidget` e `GPUProgram`. A classe `CameraWidget` é uma generalização da classe `QGLWidget` pertencente ao pacote Qt. A classe `QGLWidget` do *framework* Qt (QT [26]) é responsável por renderizar o gráfico OpenGL (Open Graphics Library) (KHRONOS [5]), ou seja, fornece as funcionalidades necessárias para exibir gráficos OpenGL integrados em uma aplicação Qt.

Como `CameraWidget` é uma subclasse de `QGLWidget`, ela implementa os seguintes três métodos vitais para executar as tarefas essenciais do OpenGL: `initializeGL()`, `paintGL()` e `resizeGL()`. O primeiro método inicializa as funções OpenGL para então possibilitar a renderização de cenas, portanto é o primeiro método a ser executado após a inicialização da interface gráfica do usuário. Esse método faz a renderização da câmera para mundo virtual. O segundo método, `paintGL()`, processa a cena OpenGL

configurando a projeção de perspectiva da câmera. Já o último método, `resizeGL()`, configura as dimensões da janela de visualização dos objetos OpenGL sendo acionado quando o objeto *widget* é criado ou redimensionado.

Além dos três métodos principais da classe `CameraWidget`, a classe reimplementa outros dois, referentes ao movimento do mouse na área do *widget*, são eles: `mousePressEvent()`, que identifica momento no qual o mouse está pressionado e está programado para retornar a posição do mouse quando o clique ocorrer no *widget* de desenho. O `mouseMoveEvent()` reconhece o movimento do mouse sobre o *widget* e está implementado para realizar a aproximação e o afastamento da câmera em relação aos objetos que compõem a cena, mas precisamente, o cabo umbilical.

A classe `GPUProgram` faz o gerenciamento entre os arquivos *shaders*, criados na linguagem GLSL do OpenGL, e o GPU. Essa classe efetua a compilação do programa *shader* a partir dos arquivos vértice e fragment *shader*. `GPUProgram` recebe o nome dos arquivos vértice e fragment da classe `CameraWidget` justificando a associação existente entre elas.

3.3.3 Pacote world

O pacote world é formado pelas classes `World`, `Light`, `Camera`, `Entity`, `Cube`, `Cylinder`, `Sphere`, `Cone` e `CoordinateSystem`.

`World` é uma classe container sendo sua utilidade armazenar e organizar instâncias das classes `Light`, `Camera` e `Entity`. Esses tipos de classe geralmente implementam um conjunto mínimo de funcionalidades padronizadas, tais como criar, inserir e remover um objeto no container, além de fornecer acesso aos objetos armazenados e informar o tamanho da lista de elementos existentes no container. Portanto a classe `World` é criada neste contexto.

De acordo com a Figura 3.3 é possível observar que as classes `World`, `Light`, `Camera` e `Entity` estão relacionadas por um tipo especial de associação, chamada de composição, e indica que ao criar a classe `World`, as classes `Light`, `Camera` e `Entity` são coincidentes. O mesmo ocorre na eliminação da classe `World`, as classes `Light`, `Camera` e `Entity` também são eliminadas.

`Light` gerencia a iluminação do mundo enquanto que a classe `Camera` representa a câmera sintética com possibilidade de se mover pelo mundo.

`Entity` é a classe base para a construção dos objetos que formam o mundo virtual a ser visualizado pelas câmeras. Ela possui métodos para inicializar e desenhar as entidades que compõem o mundo. As classes `Cube`, `Cylinder`, `Sphere`, `Cone` e `CoordinateSystem` são especializações da classe `Entity` e estendem as funcionalidades da mesma.

A classe `Cube` define cubos e paralelepípedos, pois podem ser informadas as dimensões da figura geométrica. `Cylinder` e `Cone` produz a geometria de um cilindro e um cone, respectivamente, nos quais, parâmetros podem ser adicionados, tais como: diâmetro, altura, cor e o número de faces que ele terá. A classe `Sphere` cria uma esfera de acordo com o diâmetro informado. `CoordinateSystem` define o sistema de coordenadas cartesiano utilizando as classes `Cylinder`, `Cone` e `Sphere`.

3.3.4 Pacote geometry

`HT` é a classe que representa uma matriz de transformações homogênea de dimensão 4×4 . Essa é a classe suporte para as transformações homogêneas, ou seja, a partir dela são derivadas as demais classes do pacote `geometry`, que são explicadas nesta subseção. As classes possuem um método especial, chamado de construtor, que inicializa o estado de um objeto. O método construtor da classe `HT` define uma matriz identidade conforme equação (3.1). Outro método importante desta classe é o `__mul__()` que permite realizar a multiplicações entre matrizes de transformação homogênea e de matrizes de transformação homogênea com vetores de dimensão 4×1 . Com a classe `HT` é possível ainda, obter a inversa da transformação homogênea.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

As classes `Translation`, `Scale`, `LookAt`, `Perspective`, `Ortho`, `Rotation` e `Spherical` são derivadas de `HT` e, portanto, herdam os métodos desta classe e criam uma matriz de transformação homogênea de tamanho 4×4 .

O construtor da classe `Translation` recebe os valores das variáveis x , y , z para serem adicionados na matriz de transformação homogênea de translação e assim gerar uma nova posição. Na equação (3.2), considera-se a matriz identidade (3.1) e os valores da nova posição.

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Já a classe `Scale` possui um construtor que recebe os valores das variáveis x , y , z para serem calculados na matriz de transformação homogênea de escala. Na equação (3.3), considera-se a matriz identidade (3.1) e os valores da nova escala.

$$\begin{bmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

`LookAt` refere-se a uma representação da matriz de transformação homogênea de posicionamento da câmera de acordo com a Figura B.2 do Apêndice B. Esta classe é responsável por transformar as coordenadas do mundo virtual para as coordenadas da câmera. Além do construtor, ela implementa os seguintes métodos: `_updateHT()`, `position()`, `move()`, `advance()`, `target()`, `up()`, `pan()` e `tilt()`. O método construtor define uma matriz de transformação homogênea de acordo com os argumentos posição, alvo e o vetor *up*. O método `_updateHT()` atualiza essa matriz, `position()` define a posição da câmera no mundo e `move()` move a câmera com relação ao sistema de orientação da câmera ou do mundo. O método `advance()` afasta ou aproxima a câmera do alvo. Já `target()`, define a posição do alvo e o `up()` determina a orientação da câmera em relação ao mundo. Os métodos `pan()` e `tilt()` efetuam a rotação da câmera ao longo dos eixos Y e X , respectivamente, da própria câmera.

`Perspective` e `Ortho` referem-se às projeções ortográfica e perspectiva, respectivamente, e armazenam as matriz de transformação homogêneas de cada projeção.

A classe `Rotation` é a base das matrizes de transformação homogênea de rotação e quando instanciada pelo método construtor, cria a matriz identidade conforme a equação

(3.1). Ela possui as seguintes especializações: `RotationX`, `RotationY`, `RotationZ` e `RotationAxis`.

`RotationX`, `RotationY` e `RotationZ` referem-se a matriz de transformação homogênea de rotação em torno dos eixos X, Y e Z, respectivamente. O construtor de cada uma dessas classes recebe o valor do ângulo de rotação, em radianos, que é utilizado no método `rotation()` de cada uma dessas classes para definir a matriz de transformação homogênea. A classe `RotationX` determina a matriz de transformação homogênea a partir do ângulo de rotação em torno do eixo X, medido do eixo Z para Y, no sentido anti-horário, conforme a equação (3.4).

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

onde θ é o ângulo de rotação em torno do eixo X. A classe `RotationY` determina a matriz de transformação homogênea a partir do ângulo de rotação em torno do eixo Y, medido do eixo X para Z, no sentido anti-horário, conforme a equação (3.5).

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

onde θ é o ângulo de rotação em torno do eixo Y. A classe `RotationZ` determina a matriz de transformação homogênea a partir do ângulo de rotação em torno do eixo Z, medido do eixo X para Y, no sentido anti-horário, conforme a equação (3.6).

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

onde θ é o ângulo de rotação em torno do eixo Z.

Já a classe `RotationAxis` efetua a rotação em torno de um eixo arbitrário armazenando a matriz de transformação homogênea em torno deste eixo. O método construtor desta classe recebe o valor do ângulo de rotação, em radianos, e o vetor de dimensão 3×1

contendo as coordenadas (x, y, z) do eixo. O método `rotation()` define a matriz de transformação homogênea de acordo com o ângulo de rotação e o vetor conforme equação (3.7).

$$\begin{bmatrix} x^2(1 - \cos \theta) + \cos \theta & xy(1 - \cos \theta) - z \sin \theta & xz(1 - \cos \theta) + y \sin \theta & 0 \\ xy(1 - \cos \theta) + z \sin \theta & y^2(1 - \cos \theta) + y \cos \theta & yz(1 - \cos \theta) - x \sin \theta & 0 \\ xz(1 - \cos \theta) - y \sin \theta & yz(1 - \cos \theta) + x \sin \theta & z^2(1 - \cos \theta) + \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

onde θ é o ângulo de rotação em torno do eixo arbitrário e x, y e z são as coordenadas desse eixo.

As classes `Spherical`, `SphericalSG` e `SphericalSGT` são especializações de `HT` e têm significativa importância no projeto por determinar a posição e orientação dos elos virtuais que representam o cabo em função dos valores de elevação, azimute e torção, gerados a partir dos algoritmos genéricos de ZANELA [1]. Elas herdam os métodos da classe `HT` e criam uma matriz de transformação homogênea de tamanho 4×4 .

Ao inicializar a classe `Spherical`, a matriz transformação homogênea é calculada com base nos ângulos de azimute e elevação. O azimute consiste em uma rotação em torno do eixo Z , no sentido anti-horário, medidos de X para Y . A elevação é uma rotação em torno do eixo Y , também no sentido anti-horário, medidos de Z para X . O método `_updateHT()` calcula a matriz de rotação em função dos ângulos de azimute e elevação recebidos através do método construtor da classe `Spherical`. A equação (3.8) mostra a matriz pertencente ao método `_updateHT()`.

$$\begin{bmatrix} \cos \theta_a \cos \theta_e & -\sin \theta_a & \cos \theta_a \sin \theta_e & 0 \\ \sin \theta_a \cos \theta_e & \cos \theta_a & \sin \theta_a \sin \theta_e & 0 \\ -\sin \theta_e & 0 & \cos \theta_e & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.8)$$

onde θ_a e θ_e são os ângulos de azimute e elevação, respectivamente.

A classe `SphericalSG` é uma especialização da classe `Spherical` em função da modificação necessária para ser utilizada com a simulação de cabos subaquáticos. `SphericalSG` representa uma matriz de transformação homogênea de rotação com base nos ângulos, em radianos, de azimute e elevação. Nesta classe, os ângulos de azimute permitem a rotação em torno do eixo Z , no sentido anti-horário, medidos de Y para X diferentemente do

método da classe `Spherical`, que a medição é feita de X para Y . Os ângulos de elevação ocorrem em torno do eixo Y , também no sentido anti-horário, medidos de Z para X . A equação (3.9) mostra a matriz do método `_updateHT()` da classe `SphericalSG`.

$$\begin{bmatrix} -\operatorname{sen} \theta_a \cos \theta_e & -\cos \theta_a & -\operatorname{sen} \theta_a \operatorname{sen} \theta_e & 0 \\ \cos \theta_a \cos \theta_e & -\operatorname{sen} \theta_a & \cos \theta_a \operatorname{sen} \theta_e & 0 \\ -\operatorname{sen} \theta_e & 0 & \cos \theta_e & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.9)$$

onde θ_a e θ_e são os ângulos de azimute e elevação, respectivamente.

A última classe do pacote `geometry`, a `SphericalSGT`, é uma instância da classe principal (`InterfacePrincipalQt`) da aplicação para efetuar a animação. Sua função é criar uma matriz de transformação homogênea de rotação com base nos ângulos, em radianos, de azimute e elevação e também a transformação geométrica de translação. Da mesma forma que `SphericalSG`, a classe `SphericalSGT` calcula a rotação dos ângulos de azimute em torno do eixo Z , no sentido anti-horário, medidos de Y para X e a rotação dos ângulos de elevação em torno do eixo Y , também no sentido anti-horário, medidos de Z para X . Como esta classe também processa a translação, seu construtor recebe as coordenadas de posição do elo para incluir na matriz de transformação. A equação (3.10) mostra a matriz do método `_updateHT()` desta classe.

$$\begin{bmatrix} \cos \theta_a & \operatorname{sen} \theta_a \cos \theta_e & \operatorname{sen} \theta_a \operatorname{sen} \theta_e & x \\ -\operatorname{sen} \theta_a & \cos \theta_a \cos \theta_e & \cos \theta_a \operatorname{sen} \theta_e & y \\ 0 & -\operatorname{sen} \theta_e & \cos \theta_e & z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3.10)$$

onde θ_a e θ_e são os ângulos de azimute e elevação, respectivamente, e x, y, z as coordenadas de posição de um elo.

3.4 FERRAMENTAS DE DESENVOLVIMENTO DO APLICATIVO

Nesta seção é definida a linguagem de programação utilizada para o desenvolvimento do *software* bem como justificando seu emprego na aplicação. Também são detalhadas as

bibliotecas científicas, de visualização, as gráficas e as de interface com o usuário; definindo-as e justificando-as na escolha para o desenvolvimento do aplicativo.

3.4.1 Python

O desenvolvimento do aplicativo foi realizado na plataforma de programação Python (PYTHON [27]), sendo que esta escolha se deu em função de estudos realizados levando-se em conta as seguintes características: desempenho da linguagem de programação no meio científico, por possuir facilidade de modificação, o fato da linguagem ser multiplataforma, permitir ser desenvolvida usando o paradigma da orientação a objetos, e também possuir licença gratuita.

Além de ser orientada a objetos, possui fácil integração com outras linguagens de programação como, por exemplo, C e Fortran (BORGES [31]).

Python é uma plataforma de desenvolvimento que possui bibliotecas matemáticas eficientes para a computação científica, como as que foram utilizadas no desenvolvimento do CableSim: NumPy (NUMPY [28]) e Matplotlib (MATPLOTLIB [29]). Além dessas bibliotecas de terceiros utilizou-se também o módulo *math* existente na própria linguagem. O módulo NumPy, muito utilizado neste trabalho, é uma biblioteca muito utilizada para manipulação de matrizes uni e multidimensionais.

Numpy permite a realização de operações matemáticas com matrizes utilizando o recurso de vetorização que resulta em uma maior velocidade dos cálculos, visto que as operações são realizadas por bibliotecas binárias maduras e otimizadas para essa finalidade.

Matplotlib é um pacote gráfico 2D usado para o desenvolvimento de aplicativos em Python, permitindo a geração de imagens com qualidade de publicação através de sua interface gráfica básica de usuário, que possibilita exportar os gráficos nos principais formatos de imagens (JPG, JPEG, PNG), em Portable Document Format (PDF) e Scalable Vector Graphics (SVG). Também é possível utilizar a biblioteca Matplotlib na maioria dos sistemas operacionais, tais como, Microsoft Windows, Macintosh's OS X e distribuições Linux (HUNTER [30]). Além das vantagens mencionadas, a biblioteca possui uma caixa de ferramenta, chamada mplot3d, que possibilita a plotagem de gráficos tridimensionais.

Python também apresenta vantagens quando comparado com linguagens legadas, como C/C++ e Fortran, conforme mostra a Tabela 3.1, como é o que descreve o artigo de RASHED e AHSAN [17] destacando o seu uso na ciência computacional. Como Python é

uma linguagem de uso geral, os programas desenvolvidos nesta linguagem apresentam execução um pouco mais lenta que programas criados na linguagem Java, mas podem ser desenvolvidos 3 a 5 vezes mais rápidos do que os aplicativos em Java.

Tabela 3.1 – Tabela comparativa das linguagens C/C++, Fortran e Python (adaptada do artigo de RASHED e AHSAN [17]).

Linguagem	C/C++	Fortran	Python
Utilização	Aplicação, Sistema	Aplicação, Computação Numérica	Aplicativos, Geral, Web, <i>Scripting</i>
Paradigmas	Imperativa, procedural, orientada a objetos (C++)	Genérica, imperativa, orientada a objetos, procedural	Orientada a aspectos, funcional, imperativa, orientada a objeto, reflexiva
Tipificação	forte	forte	forte
Segurança de tipos	insegura	segura	segura
Expressão de tipos	explícita	explícita	implícita
Verificação de tipo	estático	estático	dinâmico
E/S segura	Não	Não	Sim
Proporção de declarações	1 / 2.5	2	6

É importante ressaltar que todas as bibliotecas citadas até o momento possuem licença gratuita e podem ser integradas facilmente com a linguagem Python.

A Tabela 3.2 mostra a classificação das linguagens de programação mais populares segundo pesquisa realizada pelo Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE [41]) e a linguagem Python está classificada na 4ª posição nos anos de 2014 e 2015. Python é usada frequentemente por desenvolvedores para gerar ferramentas de análise de engenharia e softwares de animação (DIAKOPOULOS e CASS [42]).

Também considerou-se a possibilidade de gerar uma aplicação standalone, ou seja, que possa ser executada independente de outros softwares.

Tabela 3.2 – Tabela do IEEE com a classificação das linguagens mais populares (adaptada IEEE [43]).

Classificação	Linguagem	Tipo de Aplicação	Pontos no ano de:	
			2014	2015
1 ^a	Java	Web, Mobile, Desktop, Aplicações Científicas	100.0	100.0
2 ^a	C	Mobile, Desktop, Aplicações Científicas, Dispositivo de Controle	99.3	99.9
3 ^a	C++	Mobile, Desktop, Aplicações Científicas, Dispositivo de Controle	95.5	99.4
4 ^a	Python	Web, Desktop, Aplicações Científicas	93.5	96.5
5 ^a	C#	Web, Mobile, Desktop, Aplicações Científicas	92.4	91.3
6 ^a	R	Desktop, Aplicações Científicas	74.3	84,8
7 ^a	PHP	Web	84,8	84,5
8 ^a	JavaScript	Web, Mobile	84.5	83.0
9 ^a	Ruby	Web, Desktop, Aplicações Científicas	78.9	76.2
10 ^a	Matlab	Desktop, Aplicações Científicas	72.8	72.4

3.4.2 OpenGL

Para facilitar a programação de aplicativos gráficos utiliza-se uma biblioteca ou um pacote gráfico, que é um conjunto de rotinas gráficas básicas. OpenGL é uma biblioteca multiplataforma *open source* de instruções gráficas e de modelagem bidimensional (2D) e tridimensional (3D) (COHEN e MANSSOUR [32]).

Para realizar a modelagem do mundo virtual do CableSim optou-se pelo OpenGL, que pertence a um consórcio de empresas voltadas para o estudo de hardwares e softwares gráficos, além de ser uma biblioteca multiplataforma com suporte aos novos recursos de hardware. O estudo realizado sobre o OpenGL moderno foi explanado na subseção 1.2.3 dessa dissertação.

3.4.3 PyQt

Para desenvolver a interface do usuário, foi utilizada a biblioteca PyQt (RIVERBANK [33]), a qual utiliza a linguagem de programação Python e o *framework* Qt (QT [34]). Qt é um *framework* de desenvolvimento que possui ferramentas apropriadas para facilitar a criação de aplicativos e interfaces de área de trabalho mais sofisticadas. Com PyQt é possível desenvolver aplicações para diversas plataformas, como Windows, Linux, Mac OS X e sistemas baseados em Unix (SUMMERFIELD [35]).

Além de ser multiplataforma, Qt possui bindings para outras linguagens permitindo assim a portabilidade no desenvolvimento do software, está em constante atualização e têm-se grande base de conhecimento disponível. Essas características foram importantes para a escolha do Qt como interface gráfica do CableSim. A Tabela 3.3 mostra características pesquisadas nas páginas oficiais dos desenvolvedores.

Tabela 3.3 – Tabela comparativa das bibliotecas Qt [26], Microsoft Foundation Class (MICROSOFT [44]), GTK [45] e Swing (ORACLE [46]).

Características	QT [26]	Microsoft Foundation Class (MICROSOFT [44])	GTK [45]	Swing (ORACLE [46])
Base de Conhecimento: Experiência	Sim	Não	Não	Sim
Multiplataforma	Sim	Não	Sim	Sim
Bindings	Sim	Sim	Sim	Não
Constante Atualização	Sim	Sim	Não	Sim

Qt possui os principais recursos para a criação de aplicações GUI (Graphical User Interface, ou Interface Gráfica com o Usuário) oferecendo importantes elementos UI (User Interface – Interface com Usuário) padrões, chamados de *widgets*, como janelas, botões, controles deslizantes, barra de menus, caixas de texto, entre outros (QT [34]).

3.5 CONSIDERAÇÕES

Neste capítulo o problema que envolve essa dissertação foi exposto e também determinou-se a estrutura principal do arquivo de dados utilizado como entrada para alimentar a animação tridimensional. Com a metodologia apresentada é possível perceber com mais detalhes quais são as funcionalidades essenciais aplicadas na implementação do aplicativo, além de se ter o entendimento da estrutura de classes que envolve o projeto, bem como as principais equações que resolvem o problema de integração entre os dados gerados pelos algoritmos genéricos de ZANELA [1]. Ainda neste capítulo, adota-se Python como a linguagem de desenvolvimento do software de animação incluindo suas bibliotecas apropriadas para a computação científica. Descreve-se ainda a biblioteca gráfica OpenGL para a animação tridimensional do cabo.

No capítulo 4 descreve-se os processos utilizados na implementação dos algoritmos genéricos utilizando-se Python e OpenGL e detalha-se as etapas que envolvem a animação do cabo. Explica-se também a solução adotada para identificar os arquivos utilizados na realização de simulações, e apresenta-se a aplicação em si, ou seja, a interface com as funcionalidades do software de animação desenvolvidas com o framework Qt.

4. IMPLEMENTAÇÃO DO APLICATIVO

4.1 INTRODUÇÃO

Neste capítulo serão discutidas as implementações que norteiam este projeto de dissertação a partir do problema exposto, ou seja, será explanado o processamento das informações capturadas de acordo com a estrutura definida no capítulo anterior juntamente com a metodologia e as ferramentas adotadas para o desenvolvimento do aplicativo.

4.2 ARQUIVO DE CABEÇALHO

No Capítulo 3 criou-se uma estrutura padrão para os dados importantes identificados e apresentados na Figura 3.1 para viabilizar uma animação tridimensional. Esses dados são os ângulos de elevação, azimute e torção e o tempo da animação, armazenados, de acordo com a estrutura padrão, em um arquivo no formato TXT. Além destas informações, percebeu-se que outras seriam necessárias para executar uma simulação, tais como: número de pontos do arquivo, passo, quantidade de elos e o tamanho do elo. Estas informações são definidas nesta dissertação como parâmetros da animação.

O parâmetro número de pontos do arquivo refere-se à quantidade de linhas que o arquivo de dados da simulação possui. Em cada linha do arquivo de dados encontram-se os ângulos tridimensionais referentes à posição de cada elo em um determinado instante de tempo. No software CableSim, as linhas são utilizadas para manipular os *frames* da animação.

Conforme explica COHEN e MANSSOUR [32], uma animação é o processo de mostrar uma sequência de imagens em alta velocidade, suavemente diferentes, de forma que o observador tenha a sensação de movimento contínuo desconsiderando as imagens individuais. A velocidade em que as imagens são exibidas é chamada de taxa de exibição e a unidade de medida é qps (quadros por segundo) ou fps (*frames* por segundo).

O passo determina a quantidade de linhas que será apresentada na animação. O passo é calculado utilizando-se o conceito de Progressão Aritmética gerando assim uma sequência numérica a partir do segundo elemento, pois o primeiro termo é sempre o número 1, ou seja, a primeira linha do arquivo TXT sempre é executada na animação. Portanto o passo determina o número que é utilizado para somar ao seu antecessor formando a sequência de execução da animação. Por exemplo, se o passo for 1, todas as linhas do arquivo de dados serão animadas,

caso o passo seja 2, serão animadas apenas as linhas ímpares do arquivo, tais como: 1, 3, 5, 7, 9, Caso o passo seja igual a 3, a animação ocorre considerando-se um pulo de execução de 3 em 3.

O parâmetro quantidade de elos refere-se à quantidade de elos em que o cabo umbilical é dividido. O parâmetro tamanho do elo especifica o tamanho de cada elo na unidade de medida em metros. Com essas duas informações o sistema calcula o tamanho do cabo.

Estes quatro parâmetros também são essenciais para executar uma animação. Mesmo que alguns parâmetros se consigam obter do arquivo de dados, optou-se por armazená-los separadamente com o objetivo de organizar o processo de animação, facilitar a interface com o usuário e também minimizar o acesso a esse arquivo agilizando assim a animação. O armazenamento desses parâmetros dá-se em um arquivo no formato XML implementado utilizando-se a biblioteca ElementTree (EFFBOT [37]).

XML (*eXtensible Markup Language*) é uma linguagem de marcação extensível recomendada pela empresa W3C [38] para criação de documentos organizados hierarquicamente.

ElementTree é uma biblioteca utilizada para criar arquivos no formato XML, possui fácil integração com a linguagem de programação Python e ainda, têm licença livre (ELEMENTTREE [39]). Com esta biblioteca foram implementados métodos para efetuar a leitura e a gravação de arquivos XML contendo dos parâmetros do arquivo do cabeçalho. A principal classe da biblioteca é a `Element`, pois ela é responsável por instanciar um objeto `ElementTree`. Outra classe importante usada para manipular o arquivo XML é a classe `SubElement`, porque ela permite adicionar subelementos aos elementos principais da arquitetura XML. A Figura 4.1 mostra um arquivo XML criado pelo CABLESIM com base nos dados apresentados na Tabela 4.1, a fim de apresentar o formato da estrutura do arquivo.

Tabela 4.1 – Parâmetros de uma simulação.

Parâmetros	Valor
Passo	3
Tamanho do Elo (m)	1.0
Quantidade de Elos	10
Número de Pontos	12001

```

-<AnimationCable>
  <About autor="Roberto Aguiar Jr." comments="Cabeçalho da animação 3D de estruturas flexíveis do tipo cabo"
  email="raguiar.jr@gmail.com" status="beta" version="20151024-1400"/>
  -<Description>
    Simulação Padrão: 60 segundos Elevação + Azimuth + Torsion
  </Description>
  -<Header>
    <project name="project">FURG - PPGMC</project>
    <step name="step">3</step>
    <size_elo name="sizeElo">1.0</size_elo>
    <number_elo name="numberElos">10</number_elo>
    <number_points name="numberPoints">12001</number_points>
  -<file_header name="file_header">
    E:/Documentos/NetBeansProjects/PPGMC/FURG/src/cablesim_v18/simulation/ppgmc_furg_header_sim10e60sEAT.xml
  </file_header>
  -<file_simulation name="file_simulation">
    E:/Documentos/NetBeansProjects/PPGMC/FURG/src/cablesim_v18/simulation/ppgmc_furg_sim10e60sEAT.txt
  </file_simulation>
  </Header>
</AnimationCable>

```

Figura 4.1 – Formato XML do arquivo do cabeçalho.

Para facilitar a identificação dos arquivos XML e TXT adicionaram-se três parâmetros: descrição, caminho do arquivo do cabeçalho da simulação e caminho do arquivo da simulação. O parâmetro descrição é utilizado para especificar anotações sobre a simulação e está representada no arquivo XML pela tag <Description>. A tag <file_header> armazena o caminho e o nome do arquivo do cabeçalho. O nome e o caminho do arquivo da simulação são guardados na tag <file_simulation>.

4.3 ANIMAÇÃO

Tendo-se definido os arquivos XML e TXT que contêm os parâmetros e os dados da simulação, respectivamente, têm-se as informações imprescindíveis para se efetuar uma animação no ambiente tridimensional.

Como exposto no Capítulo 3, a classe `InterfacePrincipalQt` é responsável por gerenciar e viabilizar a animação das simulações, portanto ela possui diversos métodos implementados para gerenciar este processo. Aqui são explicados os mais importantes que envolvem a animação. Antes de efetivamente executar uma simulação, o software cria o mundo virtual para então, realizar uma animação, ganhando assim desempenho computacional.

4.3.1 Desenvolvendo o mundo virtual

O mundo virtual é formado por um ambiente subaquático considerando-se um cabo fixo na extremidade superior e conectado a um ROV na parte inferior (Figura 2.2) e por um sistema de referencia inercial definido horizontalmente no plano XY e verticalmente com eixo Z para baixo utilizando a regra da mão direita para representar o sentido de giro.

A classe `InterfacePrincipalQt` utiliza o método `readXml()` para efetuar a leitura do arquivo XML desenvolvido na biblioteca `ElementTree`. Após essa leitura, tem-se os parâmetros da simulação da simulação, dentre eles, o nome e o caminho do arquivo `TXT` que contém dos dados da simulação que são obtidos com o método `loadBody()`. O método `loadDefaultHeader()` armazena nos atributos da classe `InterfacePrincipalQt` todas as informações necessárias para a criação do ambiente de animação, como: passo, tamanho do elo, quantidade de elos, número de pontos, ângulos de elevação, azimute e torção, e o tempo da animação.

O método `drawWorld()` realiza a criação do mundo virtual através das classes `CameraWidget`, `Camera`, `Cylinder`, `Cube`, `CoordinateSystem` e `World`, que acessam os atributos da classe `InterfacePrincipalQt`. Portanto, `drawWorld()` é responsável por criar o mundo e adicionar todos os elementos envolvidos na animação neste mundo. Os principais atributos que compõem a câmera, posição, alvo e orientação (*up*), são informados na sua criação. O tamanho do cilindro é representado pelo tamanho do elo e é determinado na classe `Cylinder`. No geral, cria-se a câmera para renderização da cena (`CameraWidget`), a câmera sintética do mundo virtual (`Camera`), criam-se os demais objetos da cena (`Cylinder`, `Cube` e `CoordinateSystem`) e após adicionam-se todas as criações ao mundo por meio da classe `World`, que cria uma lista de objetos e o mundo virtual é visível no aplicativo evitando que os dados da simulação sejam acessados enquanto a animação é executada. Neste *software* o cabo umbilical é formado por um conjunto de cilindros sendo que cada cilindro representa um elo. O ROV é desenhado com dois paralelepípedos chamados, neste trabalho, de blocos. As coordenadas cartesianas são formadas pela junção das seguintes figuras geométricas: cone, cilindro e esfera. O sistema de coordenadas da Figura 4.2 é uma representação do sistema inercial, com Z para baixo e X na direção do observador. A visualização é feita por uma câmera sintética posicionada de frente para o plano YZ . Uma câmera sintética é um artifício utilizado para projetar no plano uma parte do mundo virtual como se fosse visto por um observador para dar realismo à cena, como

se uma pessoa de fato estivesse observando, usualmente a projeção é realizada com o efeito de perspectiva. O apêndice B detalha essas definições e um dos efeitos notáveis é uma distorção em relação ao fundo, em direção do ponto de fuga.

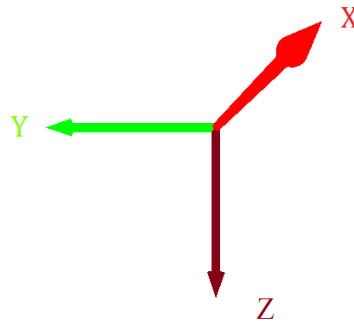


Figura 4.2 – Representação das coordenadas cartesianas.

A Figura 4.3 mostra o resultado da criação dos objetos do mundo virtual contendo as coordenadas cartesianas, quatro elos e os blocos que representam a carga terminal.

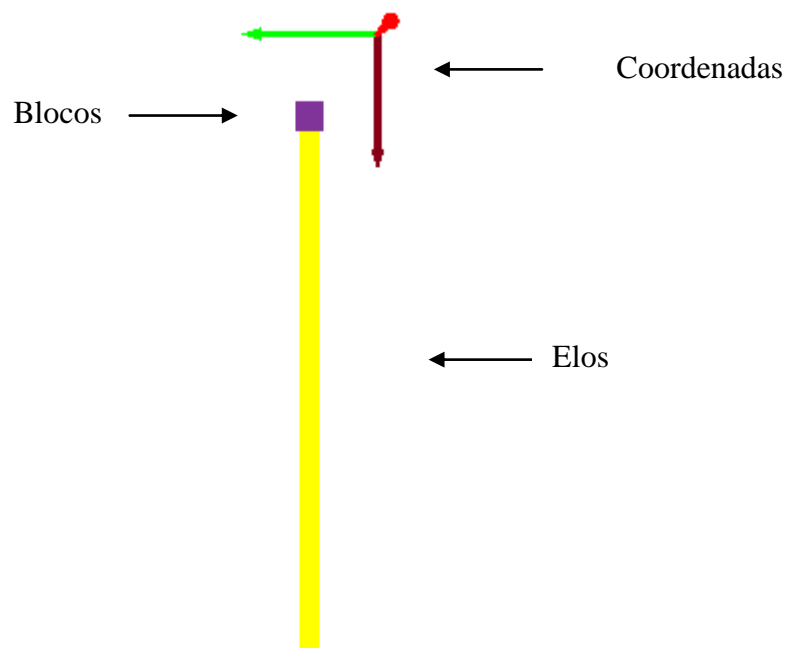


Figura 4.3 – Exemplo que ilustra a criação do mundo virtual contendo 4 elos.

Os parâmetros da câmera compreendem uma posição, um alvo que representa o local que o observador está olhando, o vetor *up* que se refere à orientação da câmera, um ângulo de

campo de visão, que no caso deste trabalho, foi definido como 90° e os limites de distância do campo de visão.

4.3.2 Execução da animação

Os principais métodos da classe `InterfacePrincipalQt` que implementam a animação são `animationWorld()` e `instantCable()`.

A animação é gerenciada pela classe `QTimer` do *framework* Qt. Essa classe implementa um temporizador, que neste projeto é o principal responsável por executar os *frames* da animação. `QTimer` executa repetitivamente o método `animationWorld()` que por sua vez executa o método `instantCable()`. Portanto, o método `animationWorld()` controla os *frames* da animação de acordo com o parâmetro passo presente na classe `InterfacePrincipalQt` através de um atributo. Já o método `instantCable()` calcula as coordenadas (x, y, z) de cada elo em um instante de tempo t . Esse instante de tempo t varia entre t_0 e t_{np} , onde np é o número de pontos da simulação. Ao mesmo tempo em que são calculadas as coordenadas de cada elo aplicam-se as transformações homogêneas de rotação e translação utilizando-se a classe `SphericalSGT` de acordo com os ângulos de elevação e azimute e as coordenadas calculadas de cada elo no instante t . Ainda no mesmo instante t calcula-se a transformação homogênea de rotação no eixo Z com a classe `RotationZ` do pacote `geometry`.

No instante de tempo $t_0 = 0$, o aplicativo desenha o cabo de acordo com os parâmetros tamanho do elo e quantidade de elos e também os ângulos de elevação, azimute e torção. A Figura 4.4 mostra o exemplo de uma simulação com 4 elos de tamanho 4 metros cada um em $t_0 = 0$.

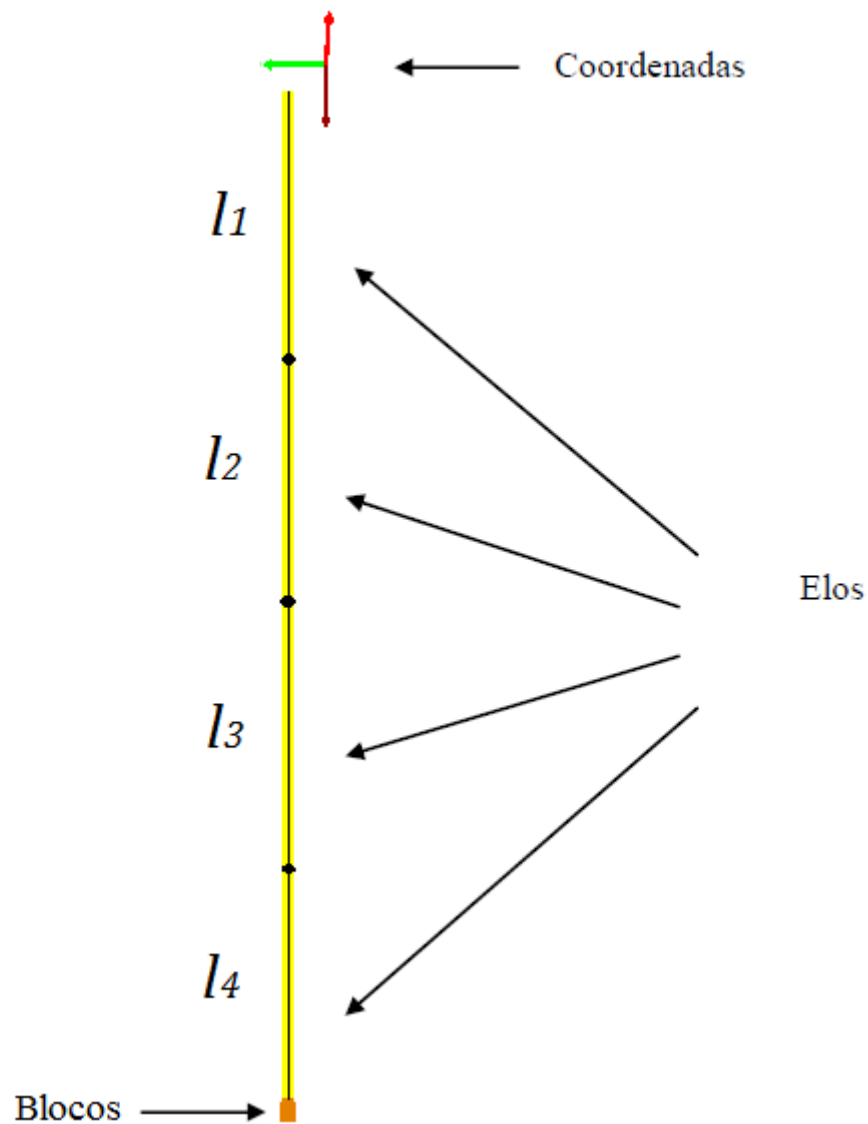


Figura 4.4 – Exemplo que ilustra a posição de um cabo com 4 elos no instante de tempo $t_0 = 0$.

O mundo virtual é composto por uma lista de objetos, no caso do atual projeto, são figuras geométricas. Para minimizar os recursos de hardware ao executar uma animação, implementou-se o método `update()` que gerencia a lista de objetos do mundo virtual. Em cada instante t , o método altera posição de cada objeto da lista. Então os objetos, são desenhados na cena uma única vez por animação evitando a utilização de memória sem necessidade.

4.4 CALIBRAÇÃO

Efetuar-se testes nas animações das simulações e identificou-se a necessidade de implementar o método `calcularCalibracao()` na classe `InterfacePrincipalQt`. O método identifica a quantidade de *frames* ideal para execução da animação e efetua a alteração do passo no arquivo XML desta simulação. A alteração é efetuada pelo método `writeXml()` implementado com a biblioteca `ElementTree` e suas funções são criar e atualizar um arquivo XML. Para programar o método `calcularCalibracao()` utilizou-se a classe `QTimer`, com o objetivo de calcular o tempo real em que a simulação foi executada considerando-se os ângulos e os 10 primeiros segundos dos dados da simulação a ser calibrada. O passo é alterado se o resultado da razão entre o tempo real da simulação multiplicado pelo passo atual e o tempo do arquivo for maior que zero e diferente do passo atual.

4.5 GRÁFICO

Com base na biblioteca `Matplotlib`, implementou-se o método `plotarGrafico()` na classe `JanelaGraficoQt`. Esse método é responsável por plotar o gráfico referente ao deslocamento da carga terminal (ROV). O método recebe os dados em uma matriz `NumPy` da classe `InterfacePrincipalQt`. Dentre os recursos disponíveis na biblioteca `Matplotlib`, destacam-se as classes `Figure`, `FigureCanvasQTAgg` e `NavigationToolbar2QTAgg`.

A classe `Figure` fornece os recursos essenciais para plotar o gráfico. `FigureCanvasQTAgg` é responsável por renderizar o gráfico na classe `JanelaGraficoQt`, e a classe `NavigationToolbar2QTAgg` permite a adição da barra de ferramentas de navegação no gráfico.

4.6 INTERFACE DO USÁRIO

A interface do usuário foi desenvolvida usando a biblioteca `PyQt` conforme explicado no Capítulo 3. As janelas do aplicativo foram implementadas utilizando-se o software `Qt`

Designer (QT [40]), que é uma ferramenta que permite criar interfaces gráficas de área de trabalho utilizando-se o *framework* Qt.

4.6.1 Classes e métodos Qt utilizadas

O aplicativo é composto por três janelas sendo duas especializações da classe `QMainWindow` do Qt e a outra derivada da classe `QDialog`. Com a classe `QMainWindow` foi desenvolvida a janela principal da aplicação e a que gera um gráfico sobre o deslocamento da carga terminal que é abordado na próxima subseção. A janela utilizada para a criação do arquivo XML apresentado na seção 4.2 foi desenvolvida utilizando-se a classe `QDialog`. Outras classes utilizadas no projeto são a `QAction` e a `QMessageBox`. `QAction` é responsável pela interação com o usuário, sendo esta uma classe importante para o aplicativo e a classe `QMessageBox` é utilizada para apresentar mensagens ao usuário. Para manipular os arquivos foi utilizada a classe `QFileDialog`, que permite a interação com o sistema operacional.

A classe `QMainWindow` fornece uma estrutura completa para a construção de uma interface principal do usuário possibilitando o gerenciamento de todas as janelas do aplicativo. Ela permite usar diversos objetos no layout, como os que foram incluídos no CableSim: `QWidget`, `QMenuBar`, `QToolBar` e `QStatusBar`.

O *layout* da janela principal do aplicativo é implementada conforme ao que se apresenta na Figura 4.5.

A classe `QWidget` é a base para outros objetos do *framework* e é nesta área que o cabo é desenhado e animado dentro do aplicativo CableSim. Ela permite a interação com o usuário através de eventos gerados a partir do mouse e teclado diretamente no *widget* ou até mesmo indiretamente acessando outros objetos da classe `QMainWindow`.

Com a classe `QMenuBar` foi adicionado na aplicação os menus importantes para tornar funcional a mesma. Utilizou-se a classe `QToolBar` para incluir botões de acesso rápido as funcionalidades do CableSim.

A classe `QStatusBar` implementa uma barra horizontal ideal para mostrar mensagens de interação com o usuário. O método `showMessage()` é o responsável por apresentar essas mensagens. Para limpar as mensagens utiliza-se o método `clearMessage()`.

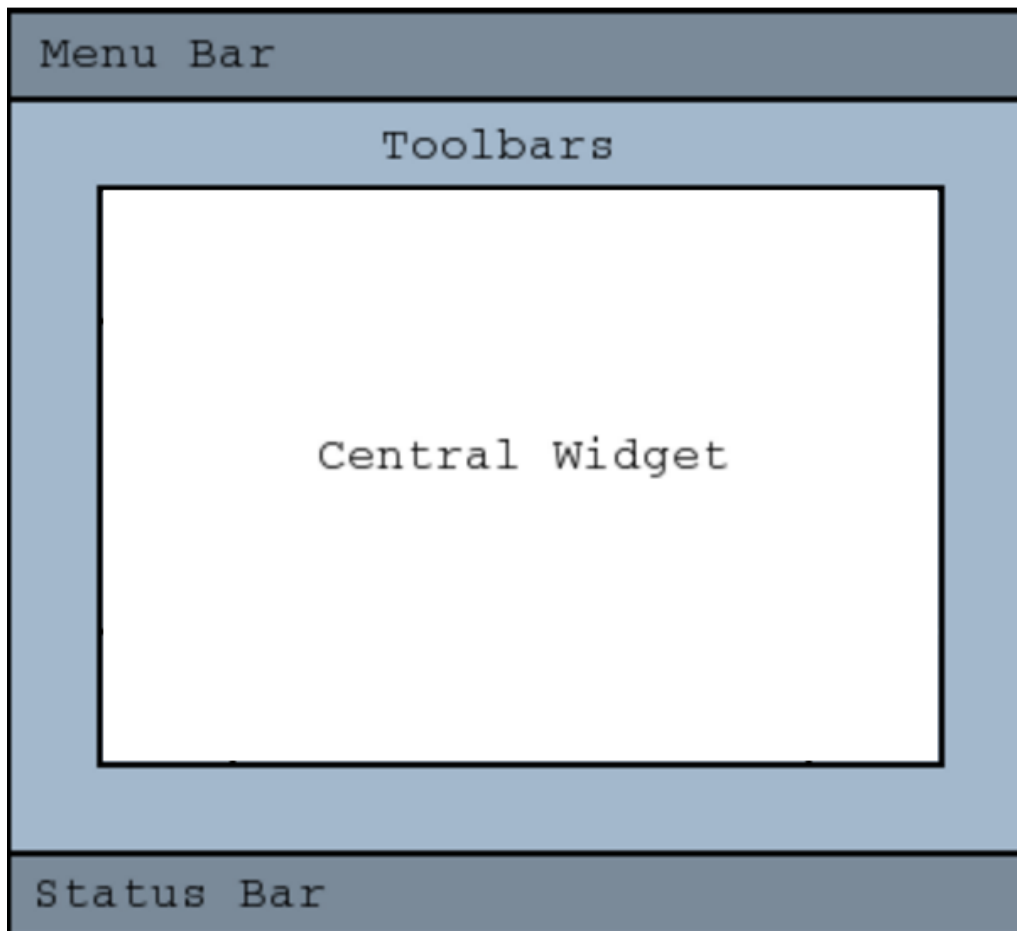


Figura 4.5 – Layout da janela principal do aplicativo CableSim (QT [34]).

Para viabilizar ao usuário a geração e leitura do arquivo XML responsável por armazenar os dados do cabeçalho da animação, optou-se por desenvolver a janela Arquivo utilizando-se a classe `QDialog`. Essa classe é a base para gerenciar janelas do tipo diálogo. Diálogos são janelas que requerem uma breve comunicação o usuário, ou seja, diálogos são utilizados para a realização de tarefas de curta duração. O método `setModal()` usado na aplicação, permite deixar o diálogo como modal, isto é, a janela permanece visível não permitindo que o usuário interaja com as outras janelas do *software* até que a mesma seja finalizada. Isso possibilitou capturar a interação do usuário com o uso de um recurso disponível somente neste tipo de janela que é configurar valores padrões de retorno de acordo as opções selecionadas pelo usuário nesta janela. A Tabela 4.2 apresenta os valores implementados, sendo que as funcionalidades configuradas são explicadas na próxima subseção.

Tabela 4.2 – Configuração das opções de retorno da janela Arquivo.

Valor do retorno	Função – Significado
0	A janela diálogo foi fechada.
1	O usuário acionou a opção de animação.
2	O usuário acionou a opção de calibragem do arquivo XML com base no arquivo de dados.

A classe `QAction` fornece recursos necessário para captar as ações do usuário nos *widgets* implementados, como as barras de ferramentas e os menus. Para as ações desenvolvidas através dos menus e barras de ferramentas foi empregado o sinal `triggered()`. Em Qt, os *widgets* possui um conjunto de sinas possibilitando a interação com o usuário e precisam estar relacionados com um *slot*. Slots são métodos que devem ser declarados para serem vinculados aos sinais (SUMMERFIELD [35]).

`QMessageBox` permitir criar pequenas caixas de diálogos do tipo modal para informar ou solicitar respostas ao usuário. Dessa classe, foram utilizados os métodos `about()`, `critical()` e `information()`. O método `about()` apresenta uma mensagem com as informações do projeto, `critical()` foi usado para mostrar erros críticos do sistema e o método `information()` para avisar sobre as diversas situações que podem ocorrer ao utilizar o aplicativo.

A classe `QFileDialog` fornece uma caixa de diálogo com recursos para abrir arquivos e diretórios implementados no método `getOpenFileName()` e para salvar arquivos como o método `getSaveFileName()`.

4.6.2 Funcionalidades

A janela principal possui as barras de menus, ferramentas e *status* e área de desenho e animação, como pode ser visto na Figura 4.6.

A barra de menus possui os seguintes menus: Arquivo, Ferramentas e Ajuda. As ações (itens de menu) dos menus Arquivo e Ferramentas estão disponíveis também na barra de ferramentas localizada abaixo da barra de menus. A barra de ferramentas é composta por botões de ações e o que diferencia os itens de menu dos botões de ações são as teclas de atalho que estão configuradas no itens de menus. A relação de teclas de atalho está disponível na Tabela 4.2.

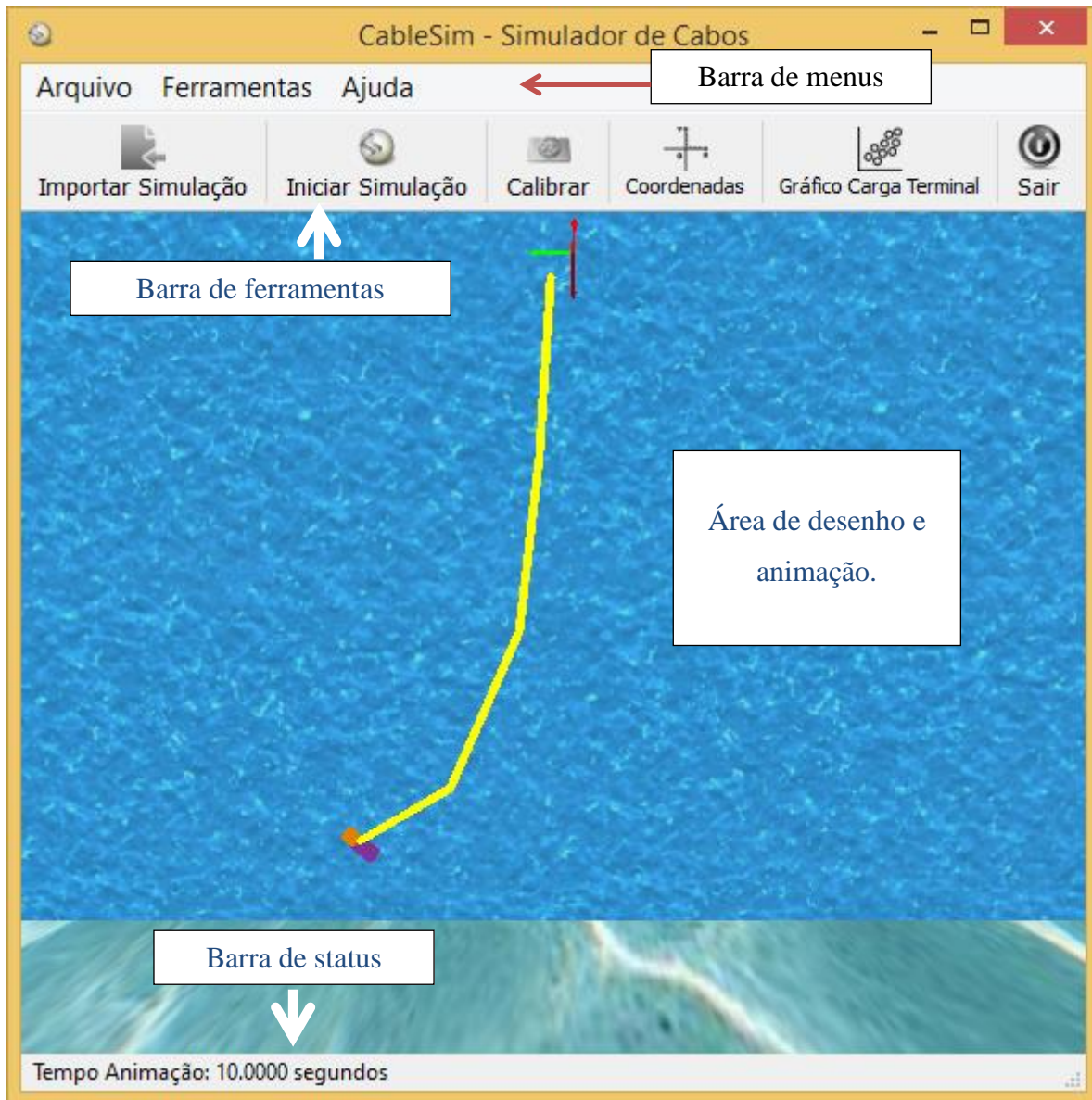


Figura 4.6 – Janela principal do aplicativo CableSim.

Tabela 4.3 – Teclas de atalho do CableSim.

Funcionalidade	Tecla de Atalho
Importar Simulação	Ctrl + I
Iniciar Simulação	Ctrl + S
Sair	Ctrl + Q
Calibrar	Ctrl + C
Coordenadas	Ctrl + O
Gráfico Carga Terminal	Ctrl + G
Sobre	F2

O menu Ajuda possui a funcionalidade Sobre responsável por mostrar as informações do projeto, como: título do trabalho, aluno e os professores orientador e coorientador (Figura 4.7).

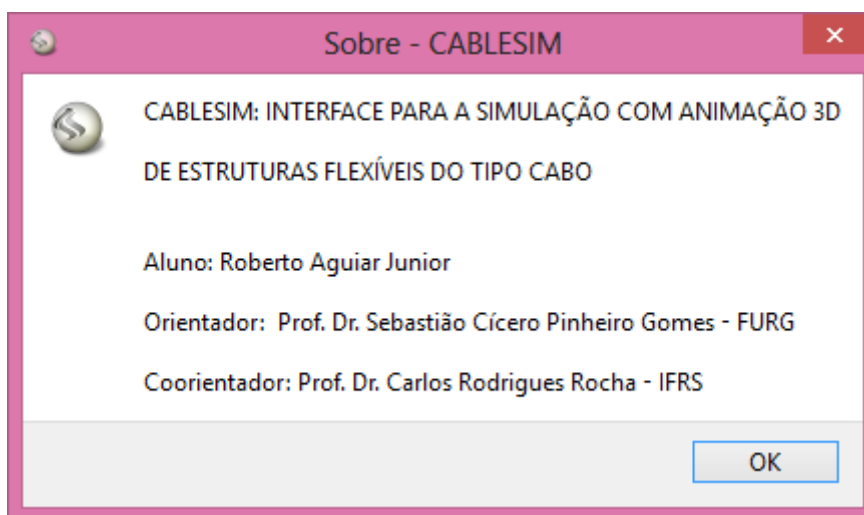


Figura 4.7 – Diálogo Sobre.

A barra de ferramentas é composta pelos botões de ações Importar Simulação, Iniciar Simulação, Calibrar, Coordenadas, Gráfico Carga Terminal e Sair. A seguir detalham-se as funções de cada um dos botões dentro do aplicativo iniciado pelo botão Iniciar Simulação por ser a principal funcionalidade do *software*.

- Iniciar Simulação: Executa uma animação tridimensional de acordo com os parâmetros informados em um arquivo no formato XML. Para facilitar a utilização do aplicativo, ele inicializa um arquivo de simulação padrão. Esse arquivo pode ser alterado acessando-se o botão Importar Simulação.
- Importar Simulação: abre uma janela de diálogo contendo os parâmetros da simulação atual, conforme mostra a Figura 4.8. Esta janela é responsável por gerenciar a manipulação dos arquivos do cabeçalho (XML) e dos dados (TXT).

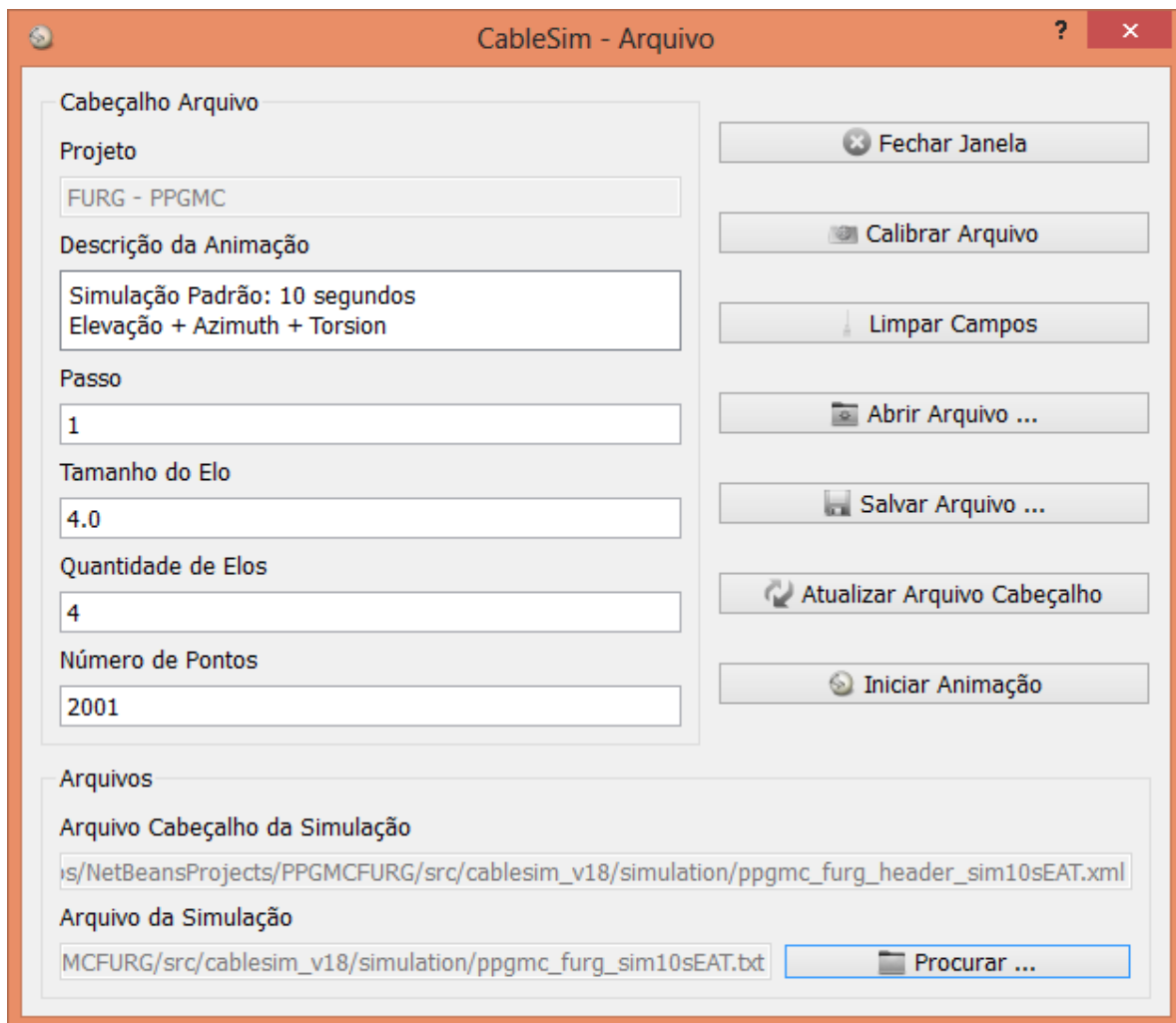


Figura 4.8 – Diálogo que gerencia o cabeçalho do arquivo.

Nesta janela, a partir do botão Salvar Arquivo é possível gerar um arquivo no formato XML com os parâmetros informados dos campos deste diálogo. O botão Abrir Arquivo permite acessar os arquivos (XML) armazenados no computador. A opção Atualizar Arquivo Cabeçalho efetua as modificações realizadas na janela de diálogo mesmo arquivo registrado no campo Arquivo Cabeçalho da Simulação. Desta janela ainda é possível iniciar uma simulação (botão Iniciar Simulação) ou efetuar uma calibragem acionando o botão Calibrar arquivo.

- Calibrar: Efetua a aproximação do tempo estipulado no arquivo de dados com o tempo real da simulação e altera o passo diretamente no arquivo XML. Esta alteração se faz necessário para futuras simulações com o mesmo arquivo no mesmo computador em que o sistema sendo executado.
- Coordenadas: é uma opção disponível que permite ao usuário visualizar as coordenadas cartesianas em cada elo do cabo umbilical (Figura 4.9).

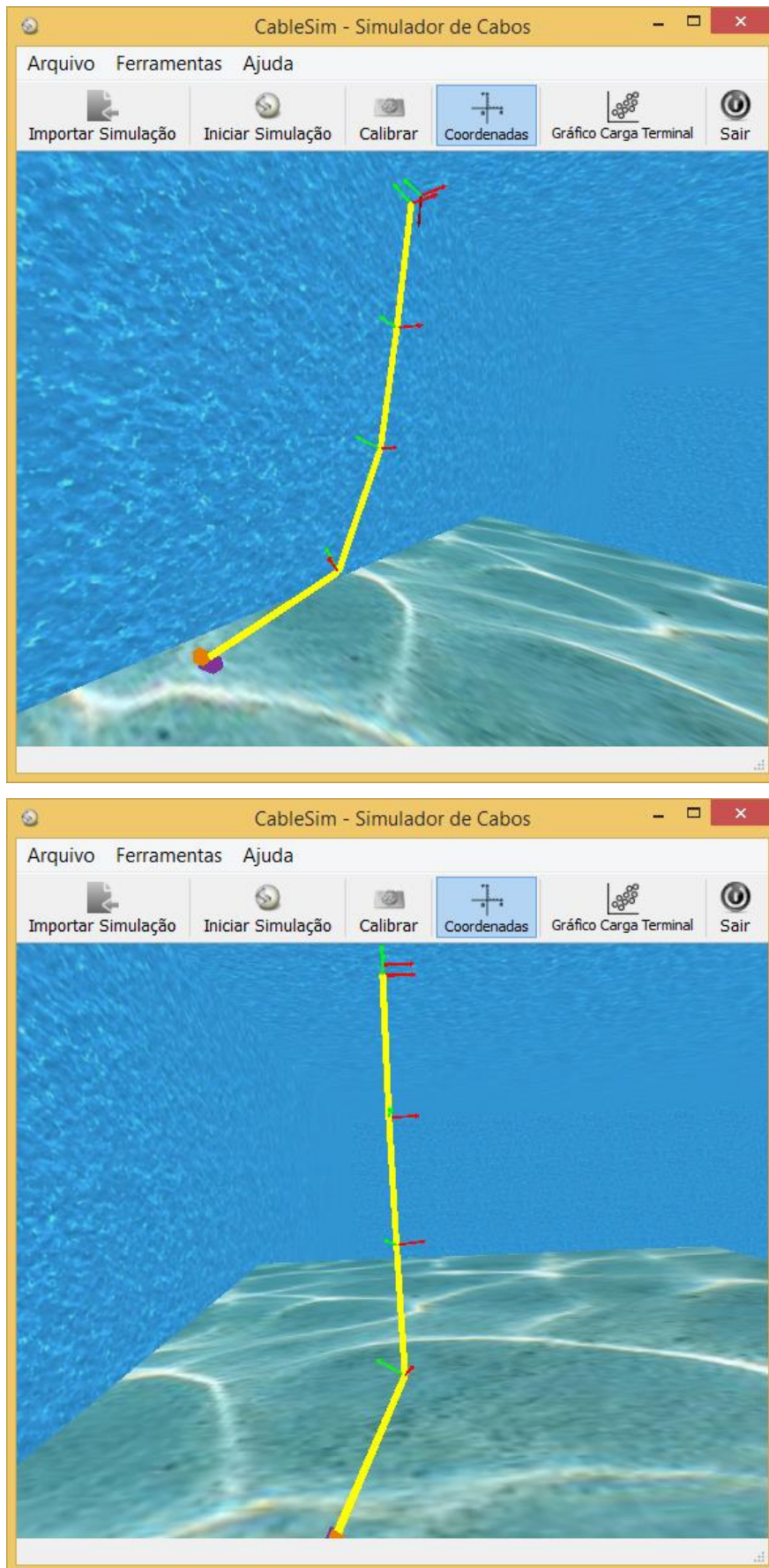


Figura 4.9 – Janela principal da animação com a opção coordenadas habilitada.

- Gráfico Carga Terminal: Janela onde é gerado um gráfico referente ao deslocamento da carga terminal por meio dos dados da animação atual. A Figura 4.10 apresenta um exemplo gerado utilizando-se os dados obtidos do arquivo padrão de simulação carregado pelo aplicativo.

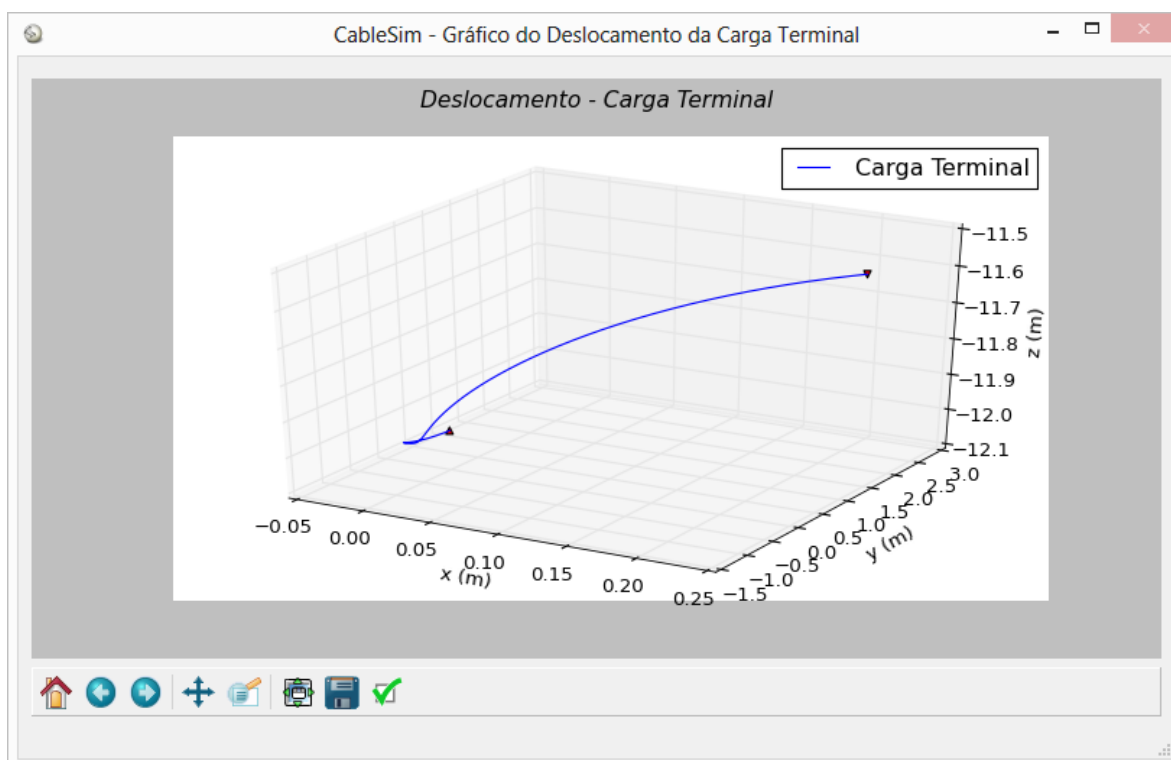


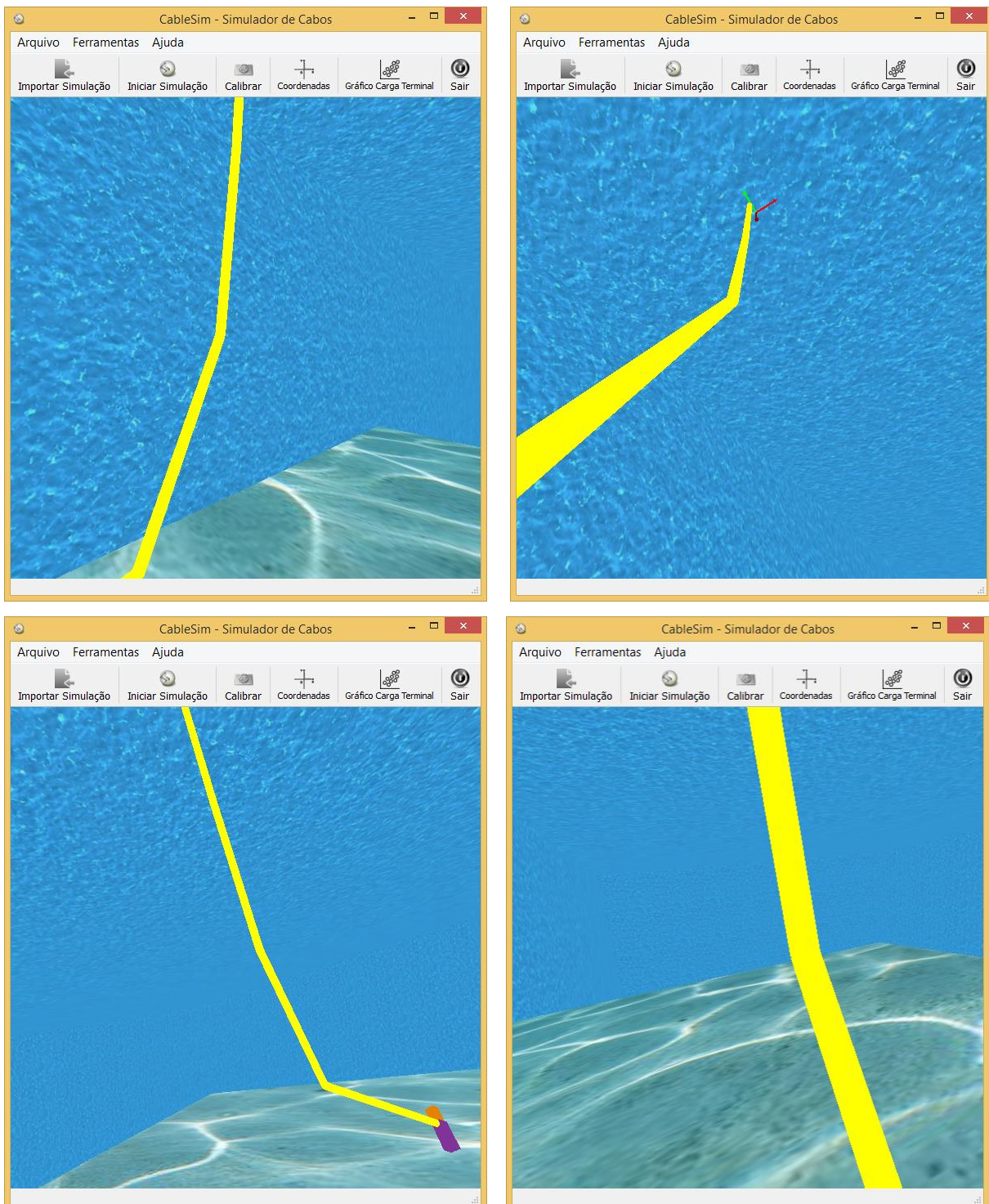
Figura 4.10 – Janela que apresenta o gráfico da carga terminal com base nos dados do arquivo padrão da simulação.

A janela do gráfico possui uma barra de ferramentas padrão da biblioteca Matplotlib, que permite realizar tarefas, como salvar o gráfico gerado nos principais formatos: imagens (JPG, JPEG, PNG), *Portable Document Format* (PDF) e *Scalable Vector Graphics* (SVG). Além disto, é possível configurar as dimensões de visualização do gráfico atual.

- Sair: finaliza o software de simulação.

A interação com o usuário ocorre também por intermédio da barra de *status*, pois é ela que mostra o tempo da animação registrado no arquivo de dados (TXT), além de ser usada para apresentar os processos envolvidos na montagem e atualização do mundo virtual.

Área de desenho e animação, além de mostrar o andamento da animação, permite que o usuário aproxime ou afaste a câmera sintética dando a impressão que é o cabo que está sofrendo esta alteração. Esta funcionalidade pode ser executada acionando o botão do meio do mouse. Também é possível girar a câmera em torno do alvo acionando-se o botão esquerdo ou direito do mouse. A *Figura 4.11* mostra o resultado da visualização quando utilizado o mouse na área de desenho, possibilitando observar os diversos ângulos formados nos elos do cabo.



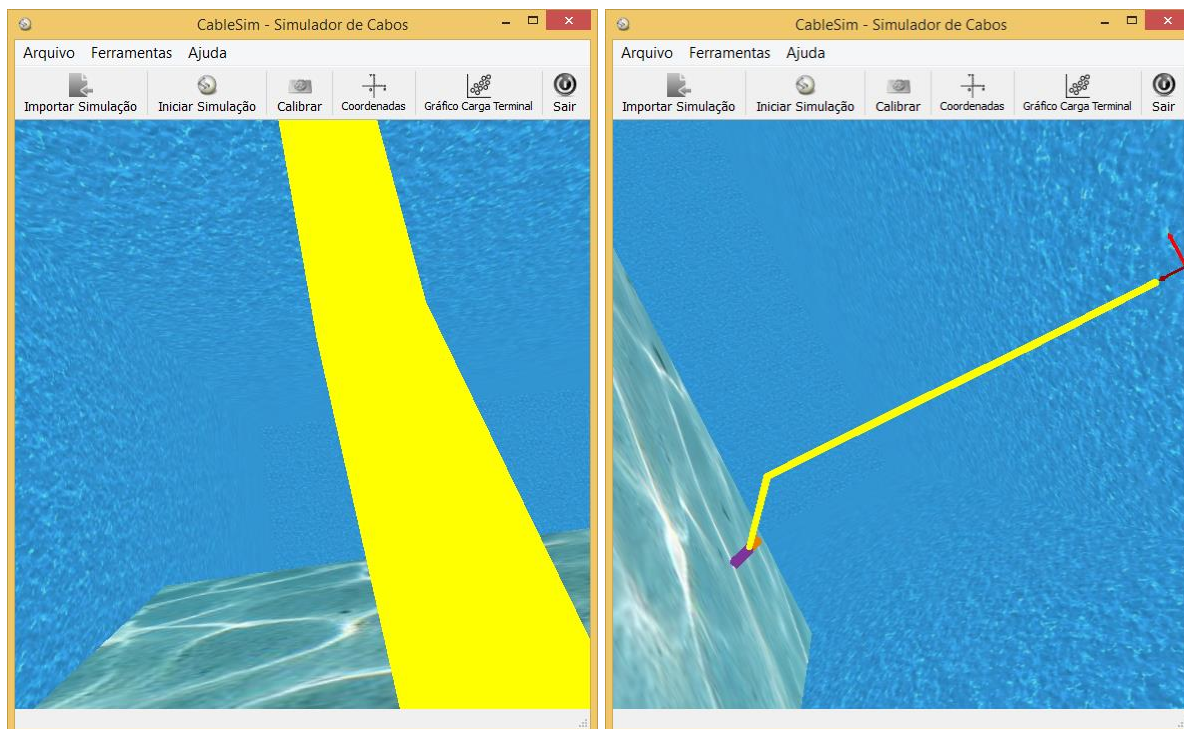


Figura 4.11 – Visualização do cabo a partir de seis ângulos distintos.

4.7 CONSIDERAÇÕES

Neste capítulo foram apresentados os dois arquivos necessários para realizar uma animação tridimensional das simulações de estruturas flexíveis do tipo cabo. No arquivo XML são encontrados os parâmetros da simulação e o arquivo TXT contém os ângulos de elevação, azimute e torção e tempo da animação. Mostrou-se a técnica de animação adotada para a implementação do cabo no mundo virtual, bem como os recursos de calibração e geração do gráfico. Ainda foi abordado questões de usabilidade do software mostrando suas funcionalidades, tais como, observar o comportamento do cabo no espaço tridimensional.

No próximo capítulo mostram-se os resultados obtidos com o aplicativo através de animações de simulações.

5. RESULTADOS

5.1 INTRODUÇÃO

Neste capítulo, são mostradas as animações referentes a quatro simulações realizadas no aplicativo desenvolvido, permitindo a análise do comportamento do cabo no espaço tridimensional conforme o número de elos considerados em cada simulação aumenta. O cabo foi considerado submerso em água e, portanto, está sujeito ao arrasto hidrodinâmico, as correntes subaquáticas e às forças de empuxo. Para todas as simulações apresentadas neste Capítulo considera-se um cabo com comprimento de 1.200 metros e diâmetro de 1 cm, sendo 600 kg a massa da extremidade inferior. Demais parâmetros físicos usados para as mesmas simulações são mostrados na Tabela 2.1.

As animações são apresentadas a partir de captura de *frames* com o objetivo de considerar o movimento do cabo durante sua evolução temporal. Em cada uma das simulações é exibido também o gráfico da trajetória da carga terminal no ambiente tridimensional.

As simulações foram realizadas utilizando-se um computador com processador Intel Core i5-2430M de 2ª geração, 2,40 Ghz, 4GB de memória SDRAM DDR3 e placa de vídeo Radeon HD 6470M com 1024 MB de memória DDR3.

5.2 PRIMEIRA ANIMAÇÃO

Para a primeira animação considera-se uma simulação com tempo de 12 segundos, sendo que os parâmetros configurados no CableSim são informados na Tabela 5.1 e os componentes configurados na câmera sintética são mostrados na Tabela 5.2.

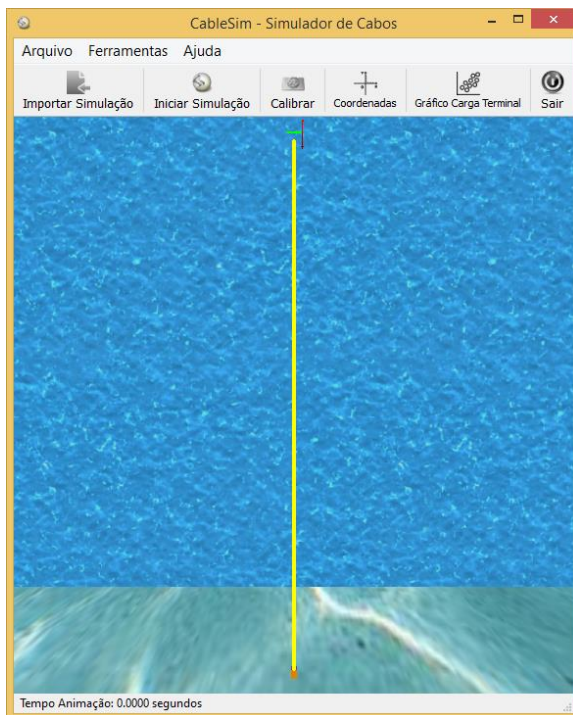
A Figura 5.1 mostra o movimento do cabo a uma velocidade de 1,5 frames por segundo (fps). Na Figura 5.1a, tem-se o cabo esticado no tempo $t = 0$. Também é possível visualizar nas imagens entre a Figura 5.1b e a Figura 5.1i o movimento do cabo de acordo com ângulos de elevação, azimute e torção e a trajetória espacial da carga terminal devido à aplicação de forças no referencial do corpo na direção Y de intensidades 245 N e 215,6 N. Com a aplicação dessas forças percebe-se a rotação do ROV em torno do cabo.

Tabela 5.1 – Parâmetros configurados no CableSim para a primeira animação.

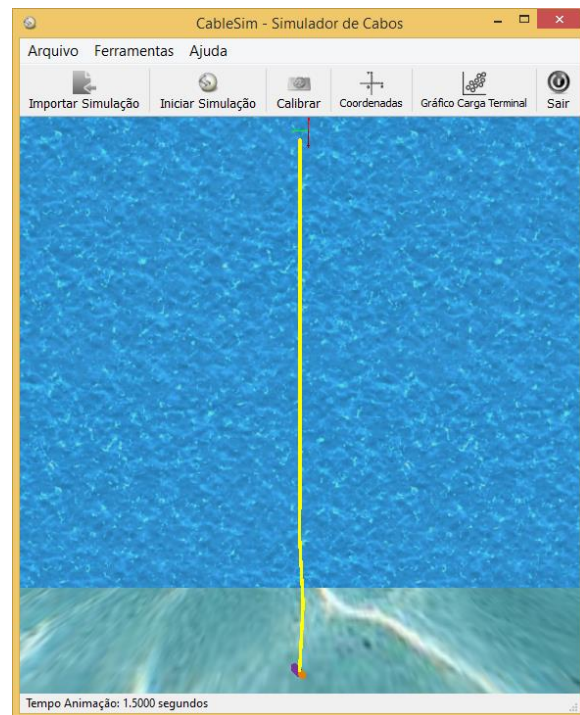
Parâmetros	Valor
Tamanho do cabo	32 m
Quantidade de elos	8
Diâmetro do cabo	20 cm
Tempo real	12 s
Passo de integração	6 ms
Velocidade	200 fps
Passo: Pulo de execução	1
Número de Pontos: Quantidade de linhas do arquivo TXT	2001

Tabela 5.2 – Componentes da câmera sintética utilizados na primeira animação.

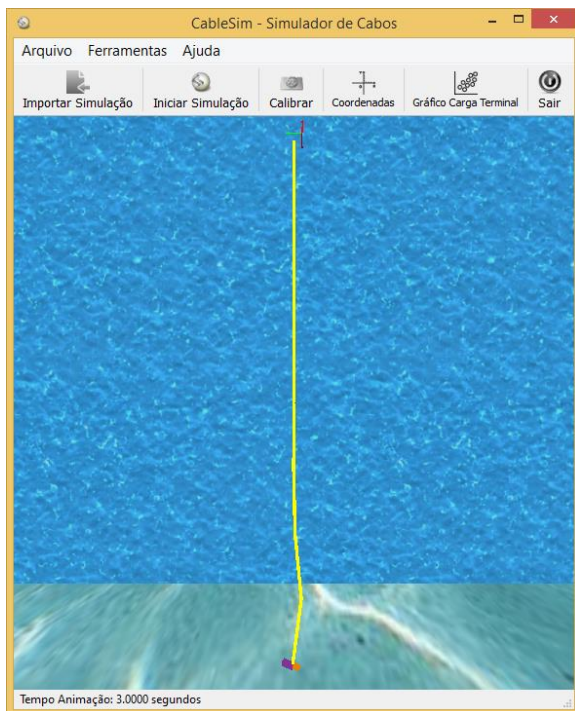
Câmera	
Posição	(1.0, 0.0, 16.0)
Alvo	(0.0, 0.0, 16.0)
Orientação	(0.0, 0.0, -1.0)
Ângulo Azimute	84,22°
Ângulo Elevação	0°



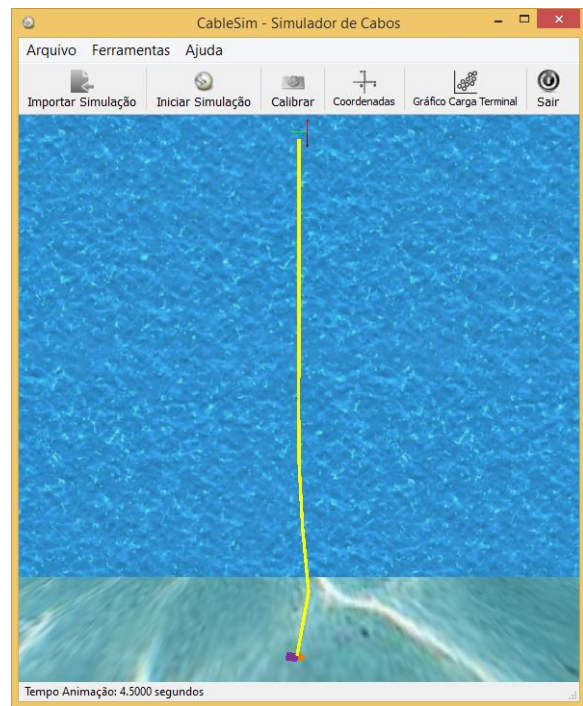
(a) Área de trabalho no tempo $t = 0$ s.



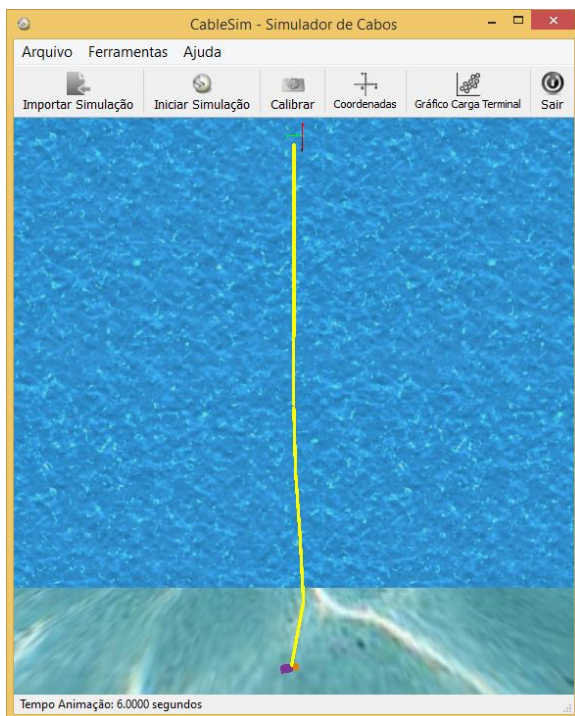
(b) Área de trabalho no tempo $t = 1,5$ s



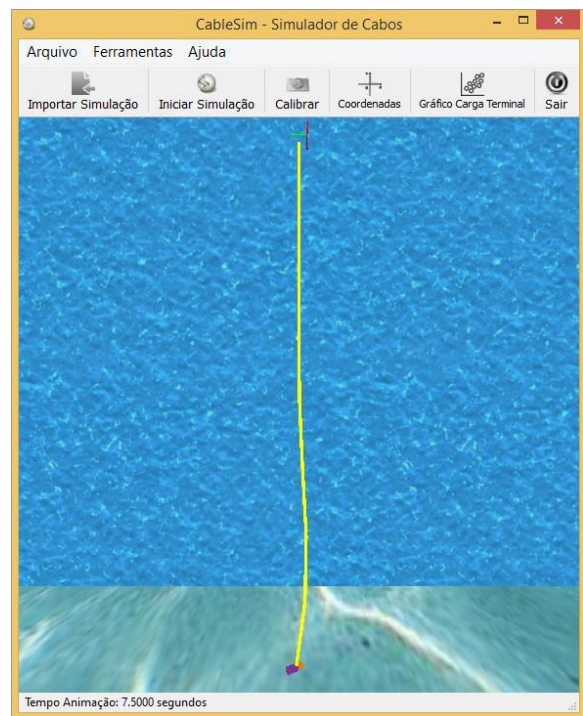
(c) Área de trabalho no tempo $t = 3$ s



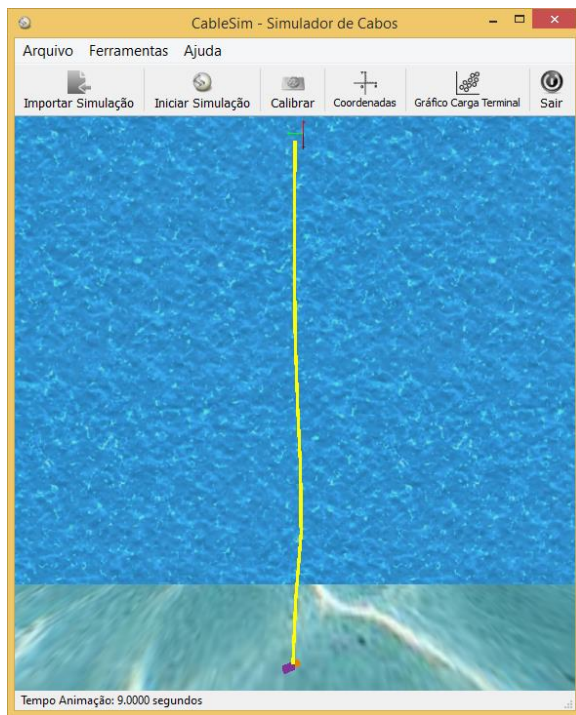
(d) Área de trabalho no tempo $t = 4,5$ s



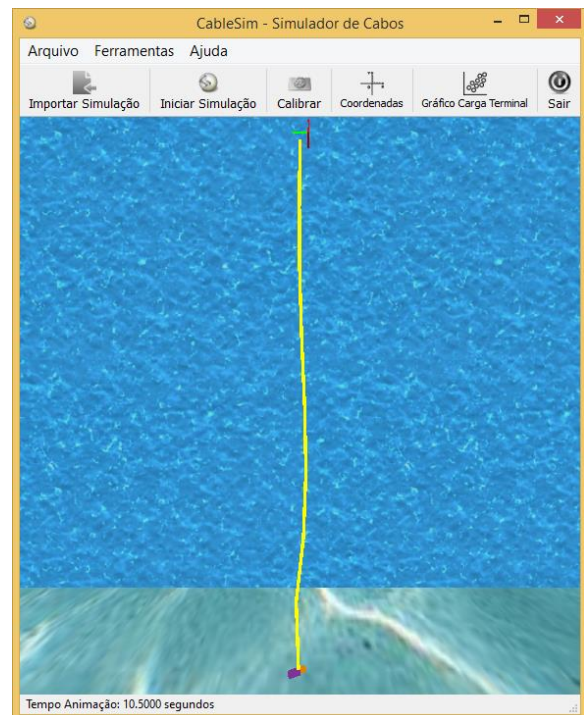
(e) Área de trabalho no tempo $t = 6$ s



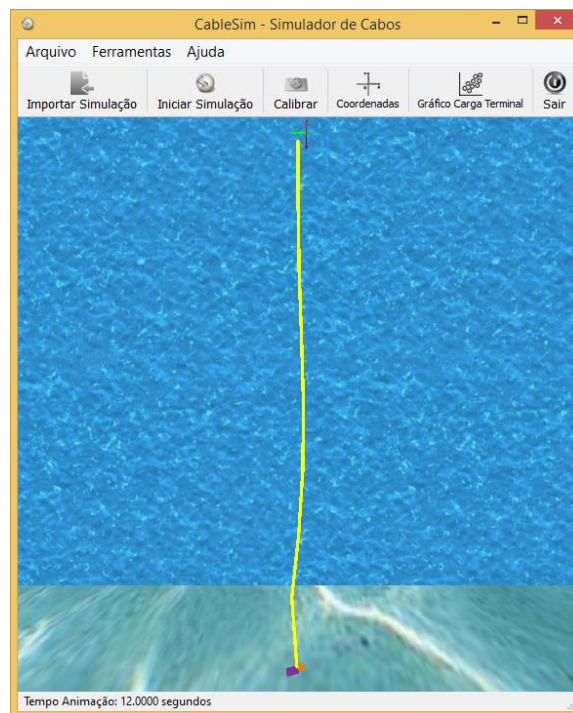
(f) Área de trabalho no tempo $t = 7,5$ s



(g) Área de trabalho no tempo $t = 9$ s



(h) Área de trabalho no tempo $t = 10,5$ s



(i) Área de trabalho no tempo $t = 12$ s

Figura 5.1 – Primeira animação (1, 5 fps): cabo com 32 metros de comprimento dividido em 8 elos de tamanho 4 metros.

Na Figura 5.2 mostra-se o gráfico da trajetória da carga terminal no espaço tridimensional.

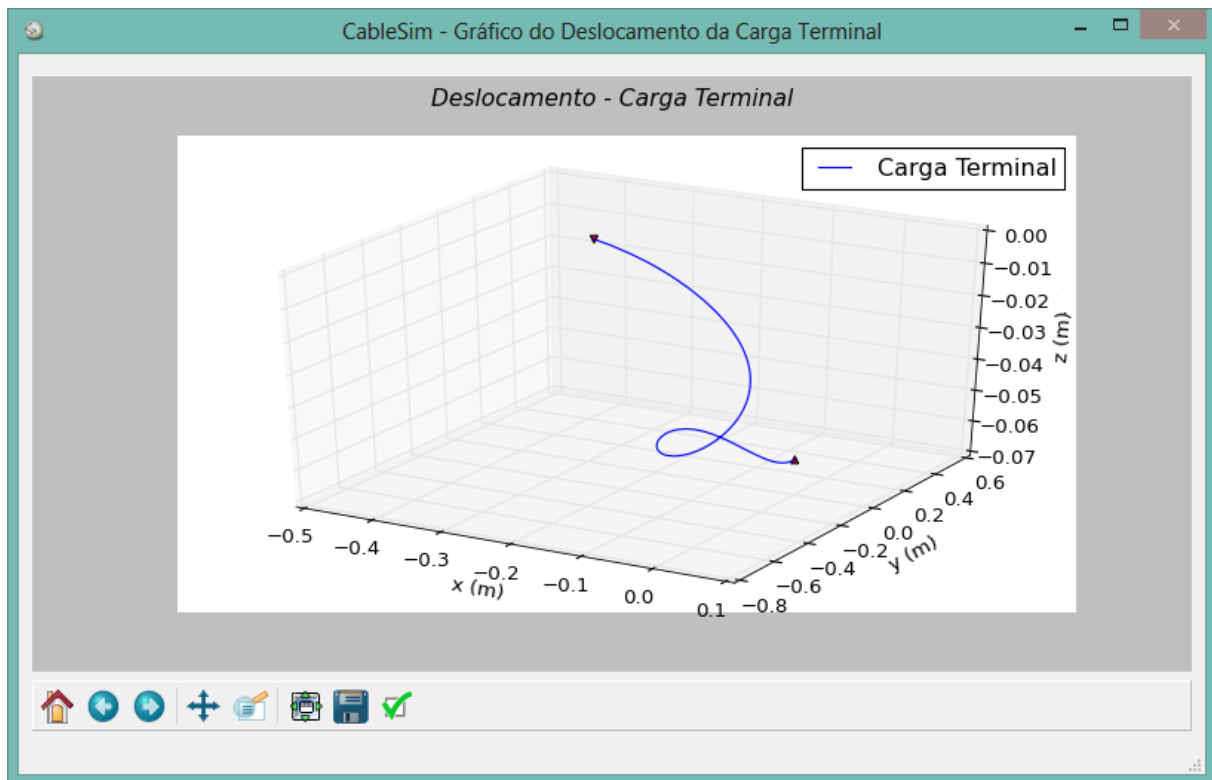


Figura 5.2 – Gráfico do deslocamento da carga terminal da primeira animação.

5.3 SEGUNDA ANIMAÇÃO

Essa simulação possui 110 segundos de duração, os parâmetros configurados no CableSim são informados na Tabela 5.3 e os componentes configurados na câmera sintética são mostrados na Tabela 5.4.

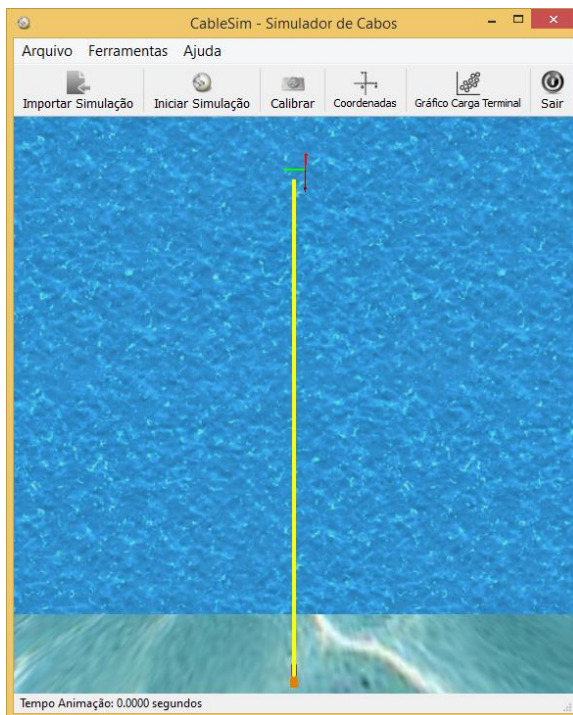
A Figura 5.3 mostra a animação do cabo a uma velocidade de 13,75 fps e o seu movimento conforme os ângulos de elevação, azimute e torção. A Figura 5.3a apresenta o cabo esticado no tempo $t = 0$. Nos primeiros 10 segundos aplicou-se uma força positiva no referencial inercial de intensidade 280 N na direção Y . Entre a Figura 5.3b e a Figura 5.3f observa-se o comportamento do cabo quando se aplicou uma força negativa também no referencial inercial de intensidade 1.400 N na direção Y no período de tempo entre 10 e 70 segundos. Após 70 segundos todas as forças são anuladas e é possível perceber através das Figura 5.3g a Figura 5.3i que o ROV cai até esticar totalmente o cabo.

Tabela 5.3 – Parâmetros configurados no CableSim para a segunda animação.

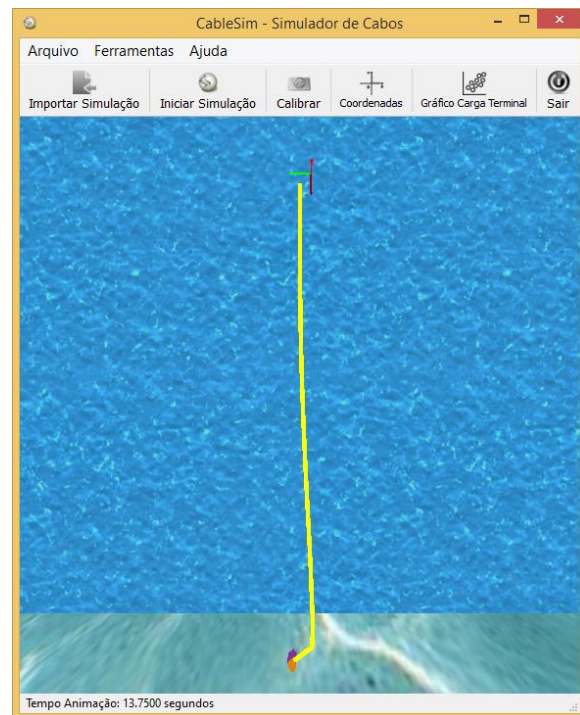
Parâmetros	Valor
Tamanho do cabo	22 m
Quantidade de elos	22
Diâmetro do cabo	20 cm
Tempo real	88 s
Passo de integração	8 ms
Velocidade	125 fps
Passo: Pulo de execução	2
Número de Pontos: Quantidade de linhas do arquivo TXT	22001

Tabela 5.4 – Componentes da câmera sintética utilizados na segunda animação.

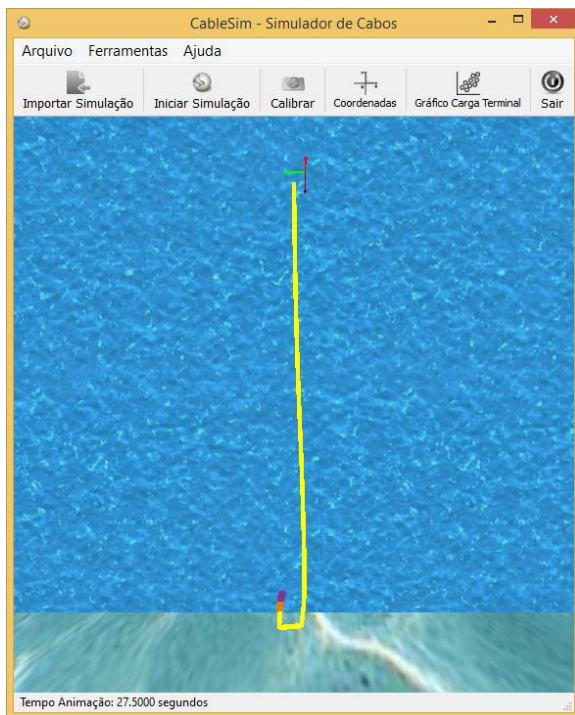
Câmera	
Posição	(1.0, 0.0, 10.0)
Alvo	(0.0, 0.0, 10.0)
Orientação	(0.0, 0.0, -1.0)
Ângulo Azimute	84,22°
Ângulo Elevação	0°



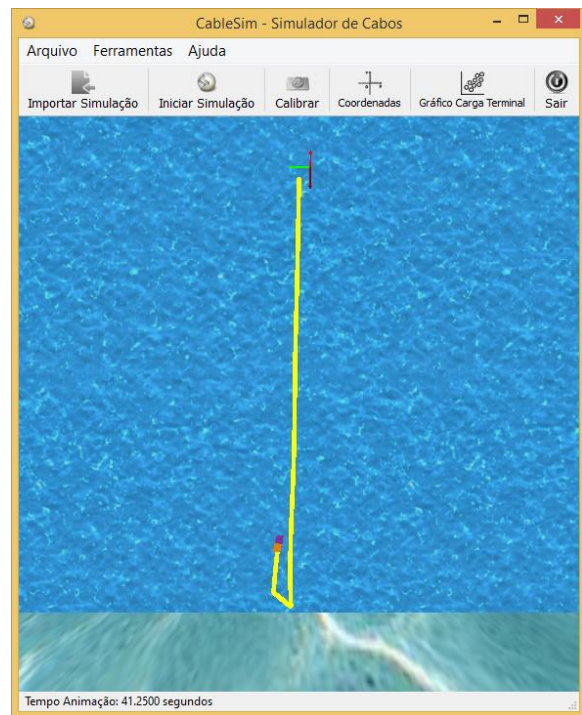
(a) Área de trabalho no tempo $t = 0$ s.



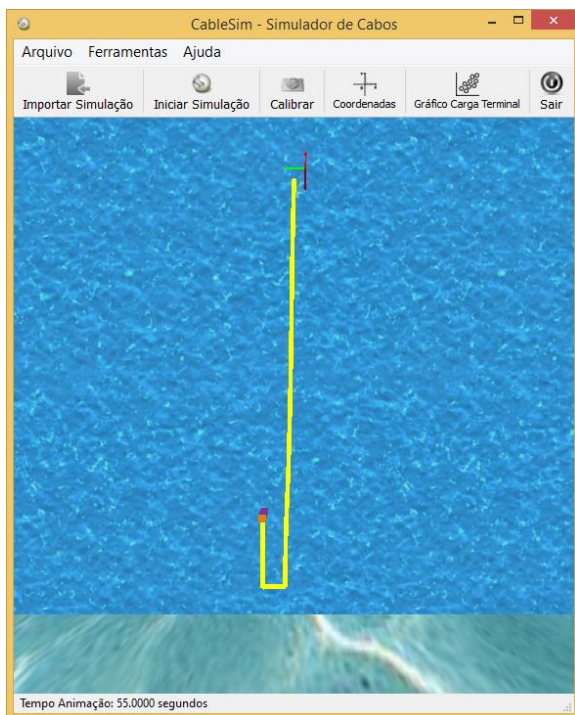
(b) Área de trabalho no tempo $t = 13,75$ s.



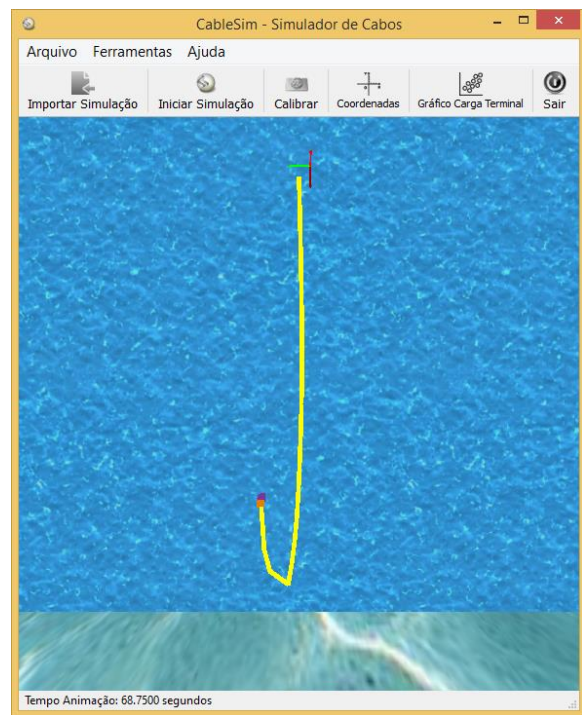
(c) Área de trabalho no tempo $t = 27,5$ s.



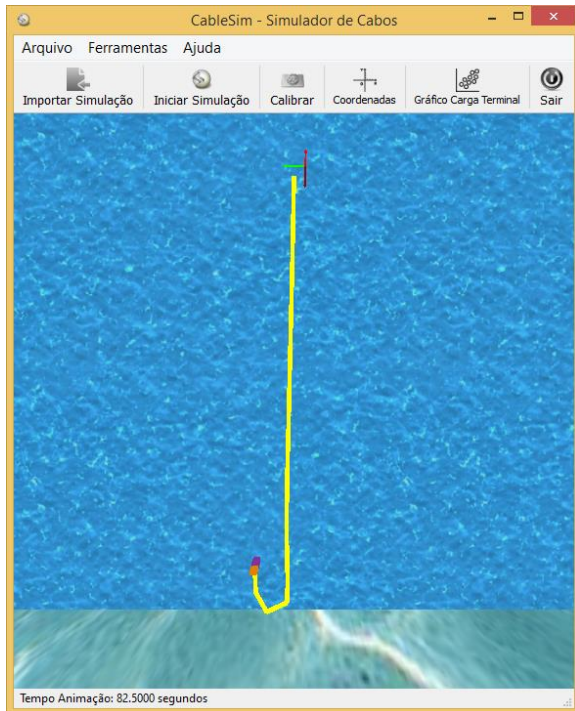
(d) Área de trabalho no tempo $t = 41,25$ s.



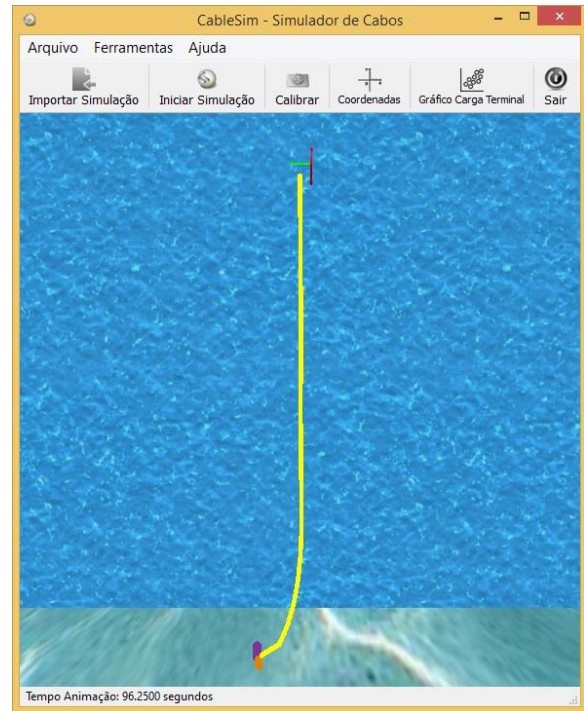
(e) Área de trabalho no tempo $t = 55$ s.



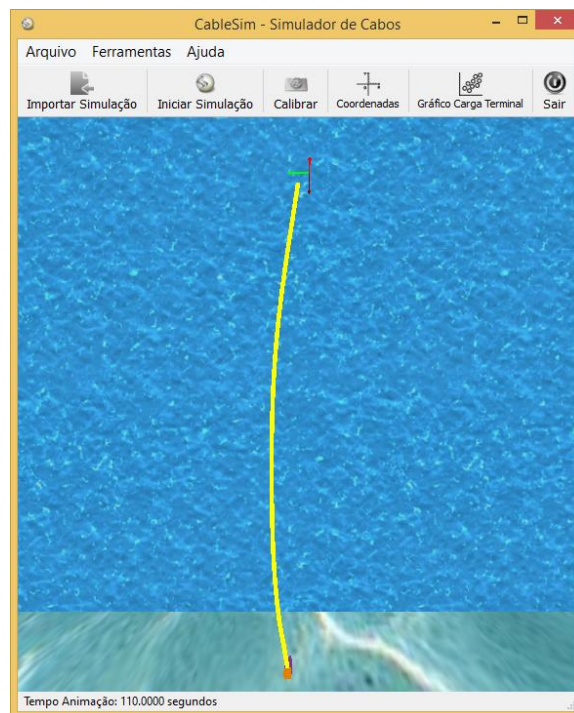
(f) Área de trabalho no tempo $t = 68,75$ s.



(g) Área de trabalho no tempo $t = 82,5$ s.



(h) Área de trabalho no tempo $t = 96,25$ s.



(i) Área de trabalho no tempo $t = 110$ s.

Figura 5.3 – Segunda animação (13,75 fps): cabo com 22 metros de comprimento dividido em 22 elos de tamanho 1 metro.

Na Figura 5.4 mostra-se o gráfico da trajetória da carga terminal no espaço tridimensional.

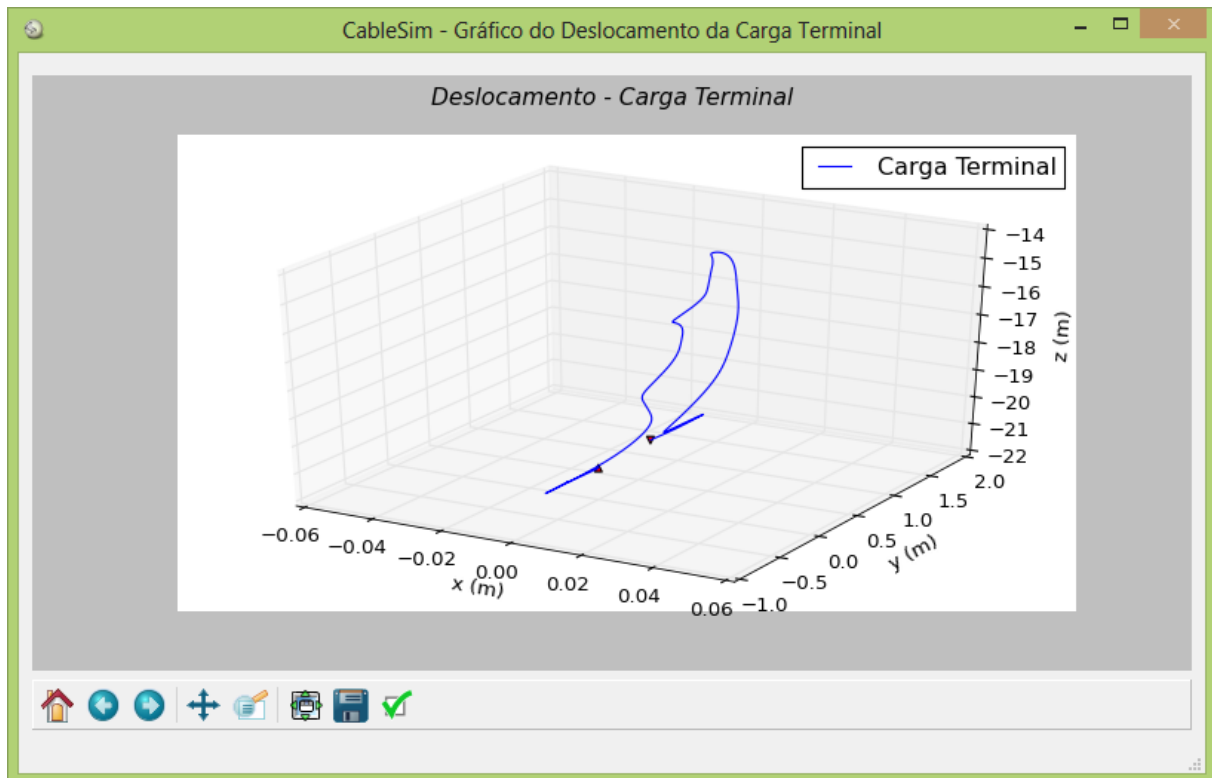


Figura 5.4 – Gráfico do deslocamento da carga terminal da segunda animação.

5.4 TERCEIRA ANIMAÇÃO

Para a terceira animação considera-se uma simulação com tempo de 60 segundos, sendo que os parâmetros configurados no CableSim são informados na Tabela 5.5 e os componentes configurados na câmera sintética são mostrados na Tabela 5.6.

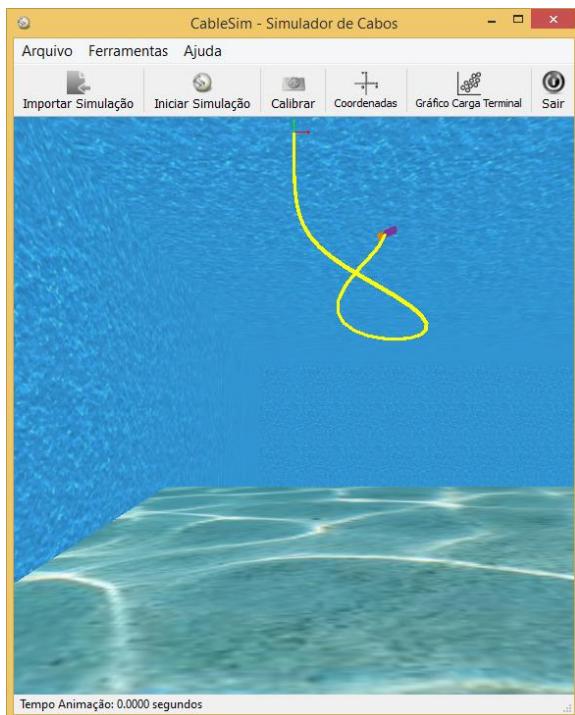
A Figura 5.5 apresenta o movimento do cabo a uma velocidade de 7,5 fps. Na Figura 5.5a, o cabo é solto no tempo $t = 0$ e nas imagens entre a Figura 5.5b e a Figura 5.5i tem-se uma sensação de que o cabo está caindo em queda livre, sem atuação de forças, até ficar totalmente esticado em $t \cong 28,8$. Em seguida ele sobe devido ao impacto causado quando esticado. Nesta animação utilizam-se os ângulos de elevação e azimute e neste caso, as colunas do arquivo TXT referentes aos ângulos de torção apresentam-se zeradas.

Tabela 5.5 – Parâmetros configurados no CableSim para a terceira animação.

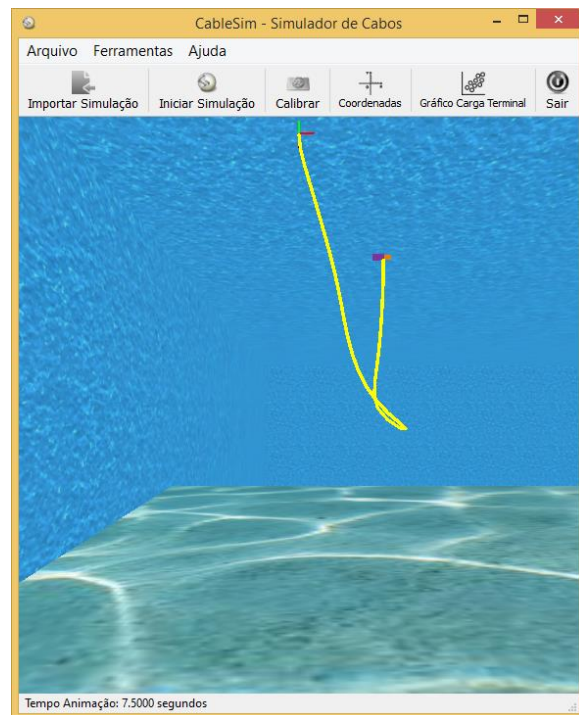
Parâmetros	Valor
Tamanho do cabo	32 m
Quantidade de elos	32
Diâmetro do cabo	20 cm
Tempo real	56 s
Passo de integração	9 ms
Velocidade	107,15 fps
Passo: Pulo de execução	2
Número de Pontos: Quantidade de linhas do arquivo TXT	12001

Tabela 5.6 – Componentes da câmera sintética utilizados na terceira animação.

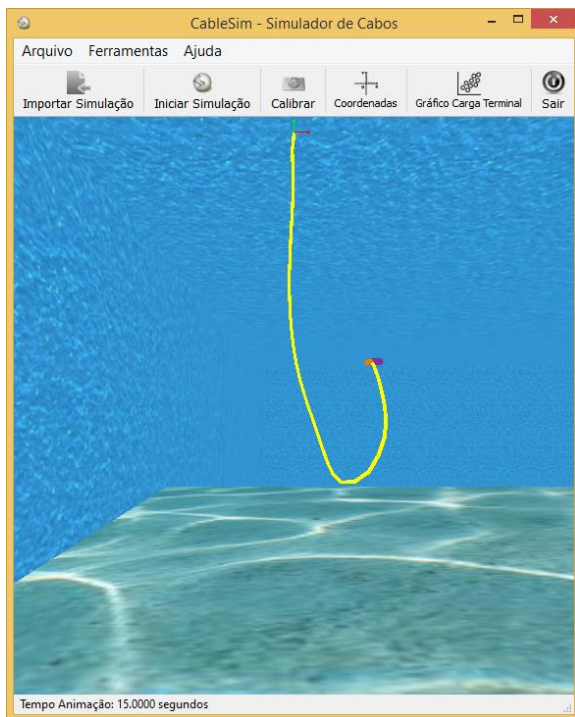
Câmera	
Posição	(0.0, 1.0, 16.5)
Alvo	(0.0, 0.0, 16.5)
Orientação	(0.0, 0.0, -1.0)
Ângulo Azimute	5,71°
Ângulo Elevação	0°



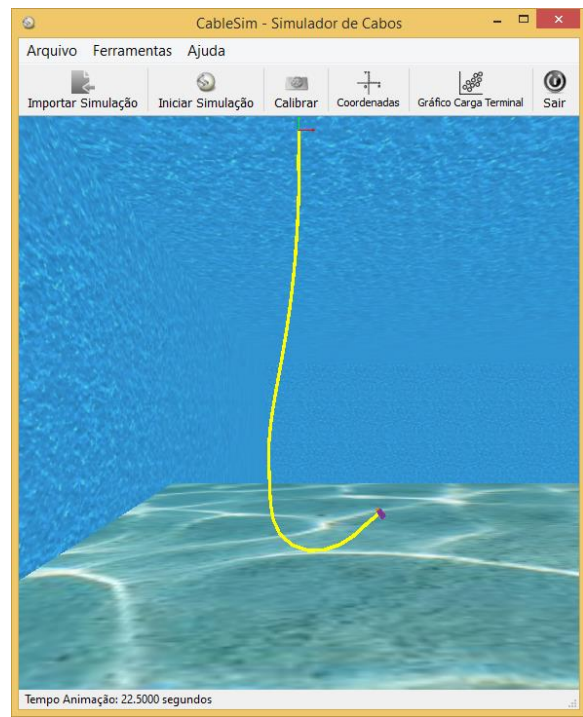
(a) Área de trabalho no tempo $t = 0$ s.



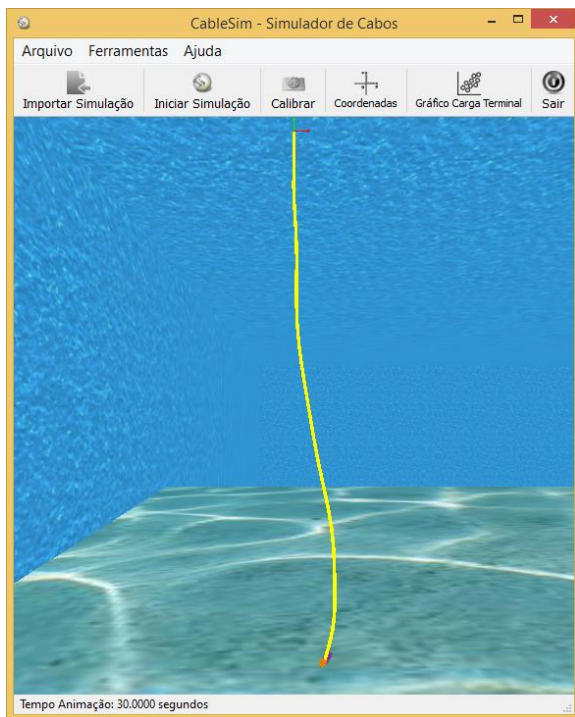
(b) Área de trabalho no tempo $t = 7,5$ s.



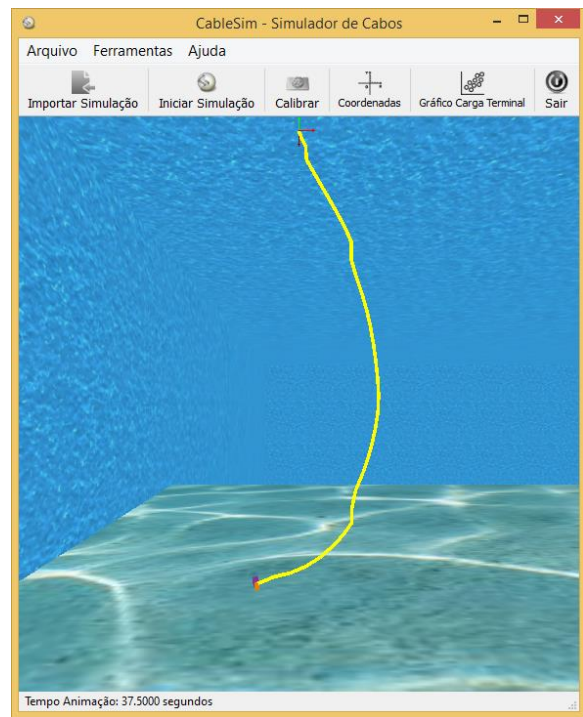
(c) Área de trabalho no tempo $t = 15$ s.



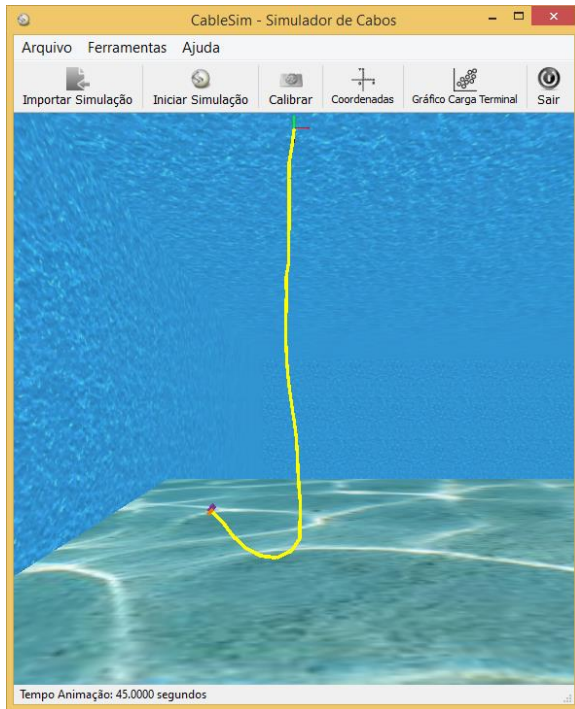
(d) Área de trabalho no tempo $t = 22,5$ s.



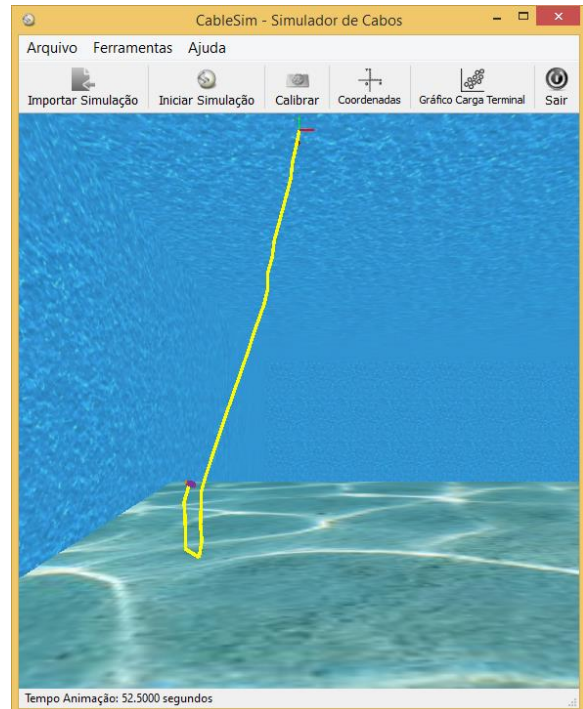
(e) Área de trabalho no tempo $t = 30$ s.



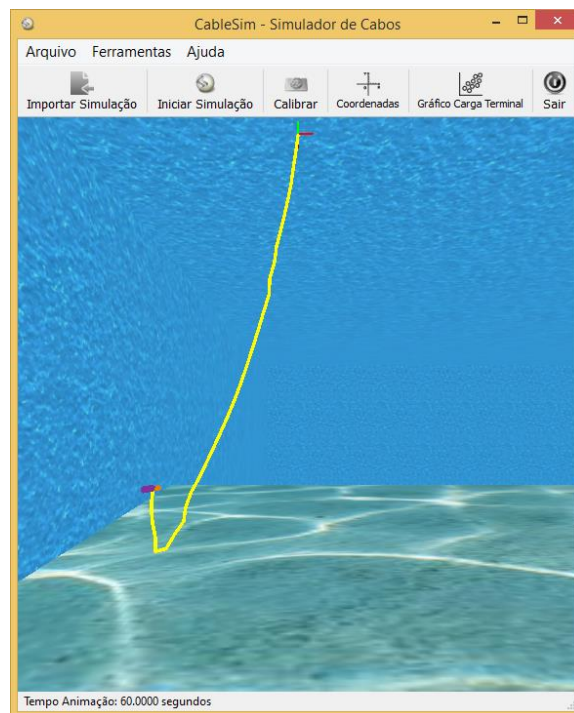
(f) Área de trabalho no tempo $t = 37,5$ s.



(g) Área de trabalho no tempo $t = 45$ s.



(h) Área de trabalho no tempo $t = 52,5$ s.



(i) Área de trabalho no tempo $t = 60$ s.

Figura 5.5 – Terceira animação (7,5 fps): cabo com 32 metros de comprimento dividido em 32 elos de tamanho 1 metro.

Na Figura 5.6 é possível visualizar a trajetória da carga terminal no espaço tridimensional.

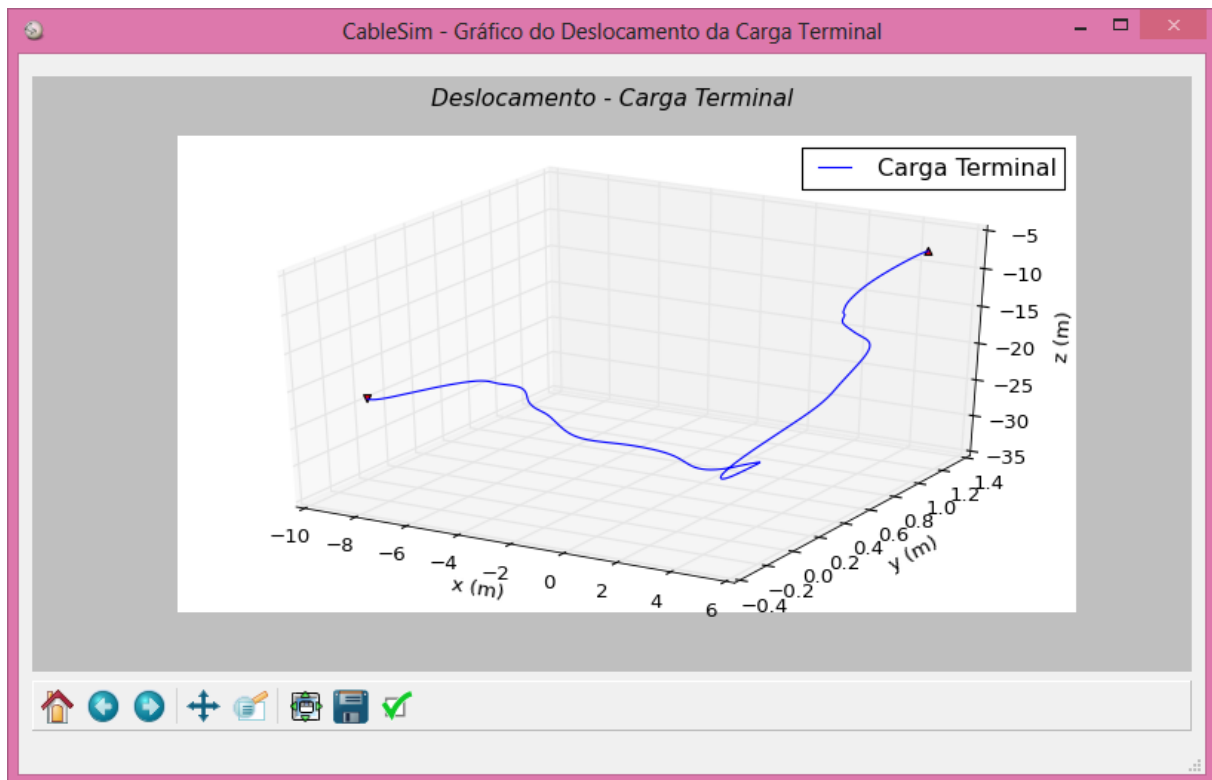


Figura 5.6– Gráfico do deslocamento da carga terminal da terceira animação.

5.5 QUARTA ANIMAÇÃO

Para a última animação considera-se uma simulação com tempo de 110 segundos, sendo que os parâmetros configurados no CableSim são informados na Tabela 5.7 e os componentes configurados na câmera sintética são mostrados na Tabela 5.8.

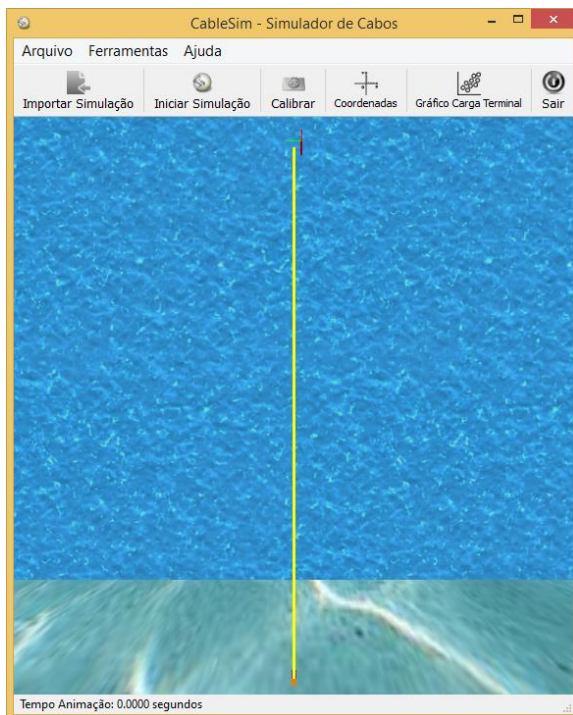
A Figura 5.7 mostra a animação do cabo a uma velocidade de 13,75 fps e o seu movimento conforme os ângulos de elevação e azimute. A Figura 5.7^a apresenta o cabo esticado no tempo $t = 0$. Nos primeiros 10 segundos aplicou-se uma força positiva no referencial inercial de intensidade 280 N na direção Y . Entre a Figura 5.3b e a Figura 5.3f observa-se o comportamento do cabo quando se aplicou uma força negativa também no referencial inercial de intensidade 1.400 N na direção Y no período de tempo entre 10 e 60 segundos. Após 60 segundos uma força positiva, ainda no referencial inercial, de intensidade 1.400 N na direção Z foi aplicada até 110 segundos e as imagens entre a Figura 5.7g e a Figura 5.7i mostram o movimento do cabo neste instante de tempo.

Tabela 5.7 – Parâmetros configurados no CableSim para a quarta animação.

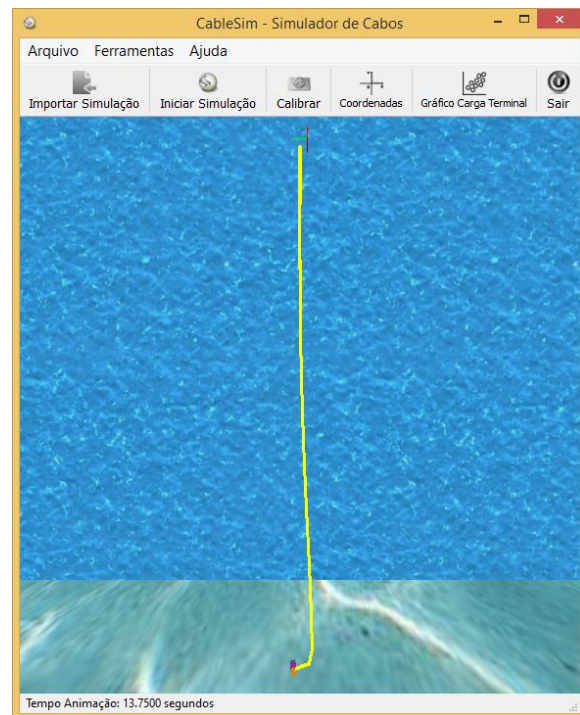
Parâmetros	Valor
Tamanho do cabo	36 m
Quantidade de elos	36
Diâmetro do cabo	20 cm
Tempo real	110 s
Passo de integração	22 ms
Velocidade	45,84 fps
Passo: Pulo de execução	4
Número de Pontos: Quantidade de linhas do arquivo TXT	22001

Tabela 5.8 – Componentes da câmera sintética utilizados na quarta animação.

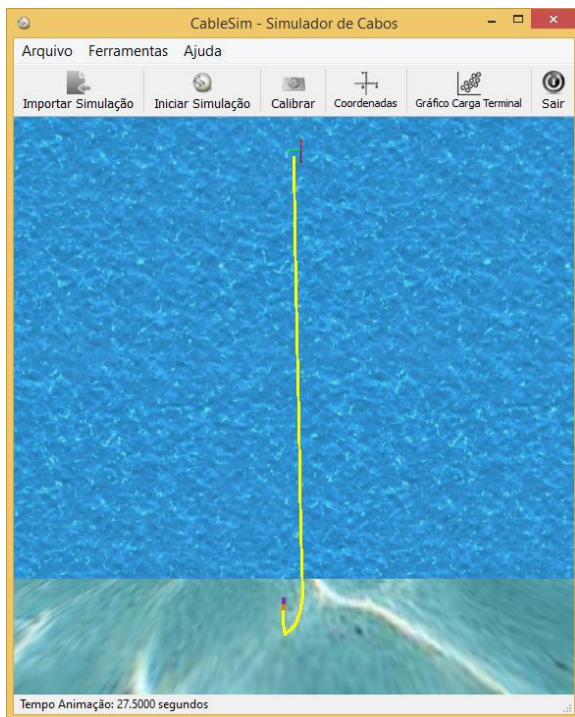
Câmera	
Posição	(1.0, 0.0, 17.5)
Alvo	(0.0, 0.0, 17.5)
Orientação	(0.0, 0.0, -1.0)
Ângulo Azimute	84,22°
Ângulo Elevação	0°



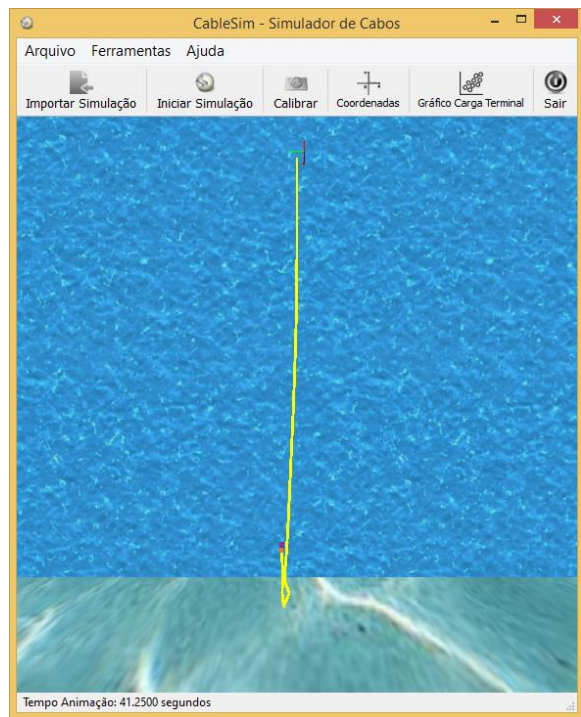
(a) Área de trabalho no tempo $t = 0$ s.



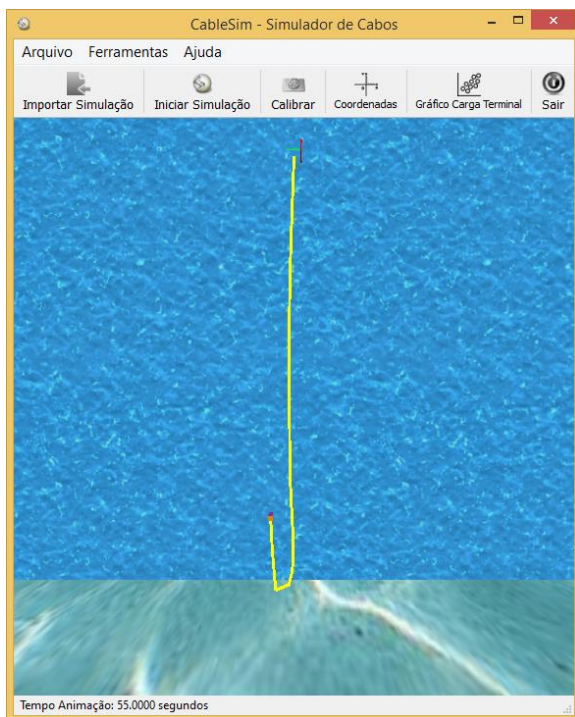
(b) Área de trabalho no tempo $t = 13,75$ s.



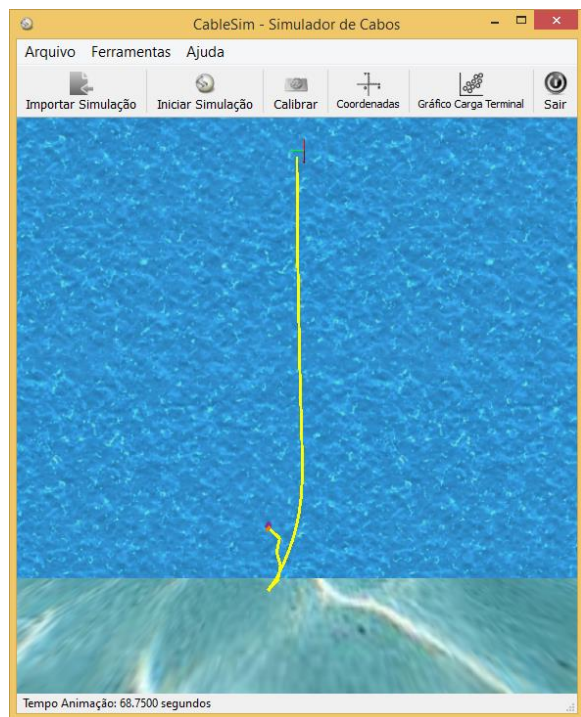
(c) Área de trabalho no tempo $t = 27,5$ s.



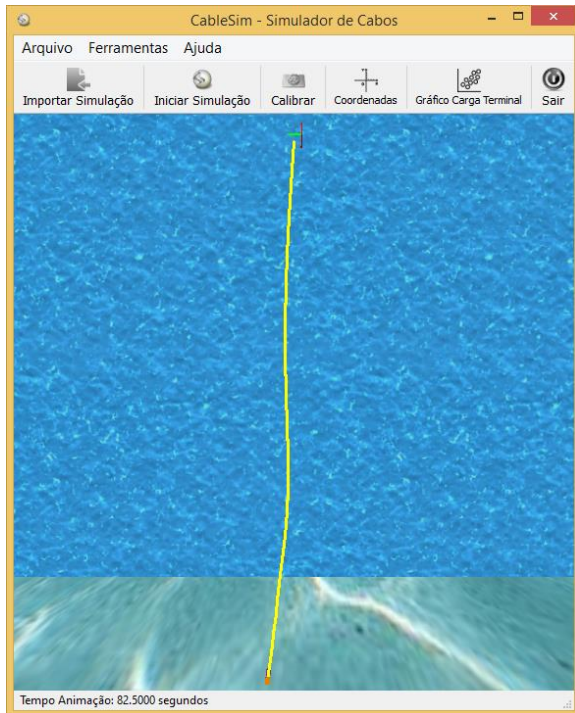
(d) Área de trabalho no tempo $t = 41,25$ s.



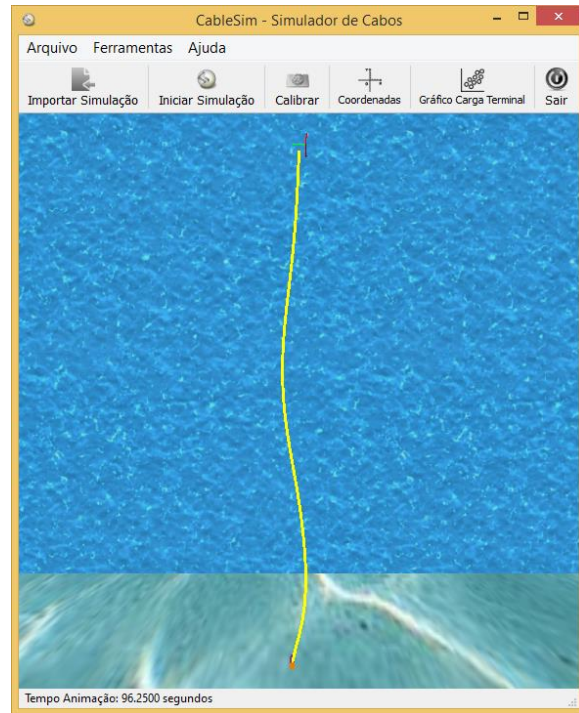
(e) Área de trabalho no tempo $t = 55$ s.



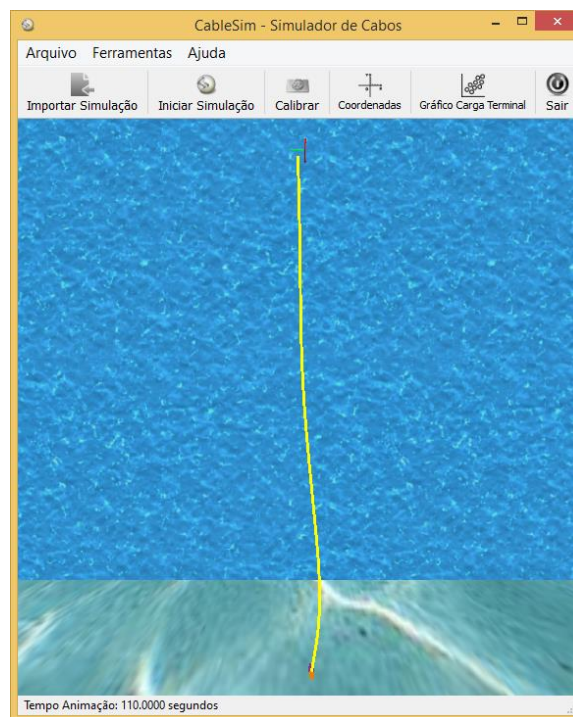
(f) Área de trabalho no tempo $t = 68,75$ s.



(g) Área de trabalho no tempo $t = 82,5$ s.



(h) Área de trabalho no tempo $t = 96,25$ s.



(i) Área de trabalho no tempo $t = 110$ s.

Figura 5.7 – Quarta animação (13,75 fps): cabo com 36 metros de comprimento dividido em 36 elos de tamanho 1 metro.

Na Figura 5.8 é possível visualizar a trajetória da carga terminal no espaço tridimensional.

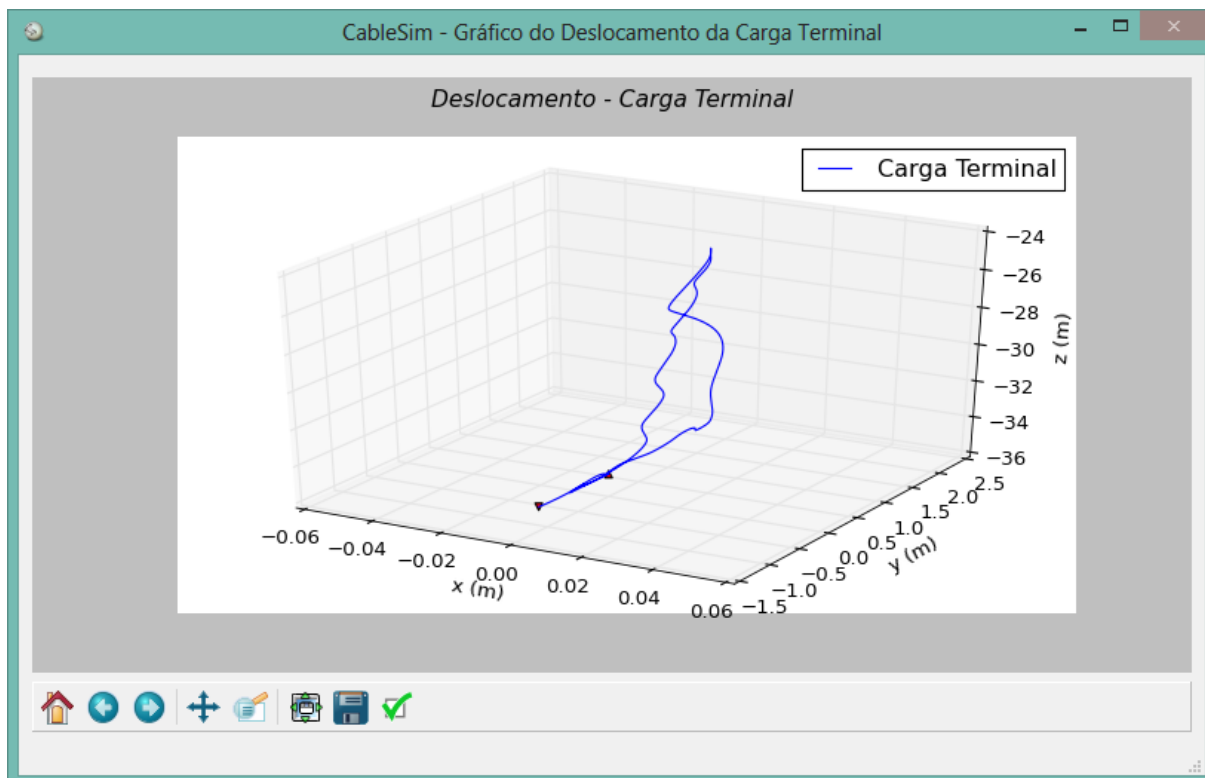


Figura 5.8 – Gráfico do deslocamento da carga terminal da quarta animação.

5.6 CONSIDERAÇÕES

Como pode ser observado nas simulações realizadas, o software atende a necessidade de efetuar animações tridimensionais com desempenho de processamento aceitável, pois conforme o número de elos em que o cabo é dividido aumenta, o tempo de animação da simulação é muito próximo do tempo previsto no arquivo de dados importado dos algoritmos genéricos de ZANELA [1]. Com o objetivo de facilitar a análise das quartas simulações realizadas neste Capítulo, a Tabela 5.9 contém os parâmetros de desempenho das mesmas.

Ademais, a estrutura flexível projetada em três dimensões proporciona uma sensação de realidade juntamente com o passo implementado para aproximar os tempos de simulações.

Tabela 5.9 – Parâmetros referentes ao desempenho das simulações realizadas neste Capítulo.

Parâmetros	Animações			
	Primeira	Segunda	Terceira	Quarta
Quantidade de elos	8	22	32	36
Tempo da simulação (segundos)	12	110	60	110
Tempo real (segundos)	12	88	56	110
Velocidade (fps)	200	125	107,15	45,84

6. CONCLUSÕES

Neste trabalho, realizou-se o desenvolvimento de uma interface que permite analisar o comportamento dinâmico de estruturas flexíveis do tipo cabo a partir de animações tridimensionais. O que motivou o trabalho foi a possibilidade de visualizar o movimento do cabo umbilical no espaço tridimensional, tendo como base a teoria apresentada no Capítulo 2. Os principais estudos realizados para atingir o objetivo desta dissertação, além de compreender a modelagem dinâmica de estruturas flexíveis do tipo cabo foram a linguagem de programação e a biblioteca gráfica.

A linguagem de programação adequada para a manipulação de matrizes de forma eficiente foi um pré-requisito considerado na escolha do Python. Recursos, tais como integração com bibliotecas gráficas e de visualizações também contribuíram para essa escolha. Investiu-se um tempo de estudo no OpenGL moderno, pois ele é o principal responsável por realizar as animações implementadas no CableSim.

O início do projeto deu-se com a escolha da metodologia e da modelagem de desenvolvimento adequadas para viabilizar o objetivo proposto. O paradigma da orientação a objetos proporcionou não só o desenvolvimento do *software*, mas também a implementação de bibliotecas que podem ser utilizadas para modelar um mundo virtual com visualização por câmeras sintéticas. É importante ressaltar que estas bibliotecas podem ser reutilizadas em outros projetos que envolvam o ambiente de programação Python e OpenGL.

Como produto principal resultante desta dissertação, tem-se o aplicativo CableSim, que implementa as funcionalidades essenciais para a realização de animações tridimensionais de estruturas flexíveis do tipo cabo, permitindo assim uma melhor análise dos resultados de uma simulação. Isto pode ser contemplado com as simulações realizadas no capítulo anterior. A análise ocorre por meio da geração do gráfico referente ao deslocamento da carga terminal e também pelo recurso da área de desenho, o qual aproxima e afasta a câmera sintética possibilitando uma melhor visualização do cabo em uma simulação.

Os algoritmos genéricos que permitem gerar de forma automática os elementos dos vetores e matrizes do modelo dinâmico constituem uma poderosa ferramenta de modelagem dinâmica de estruturas flexíveis do tipo cabo, importante para diversas aplicações, principalmente vinculadas à indústria do petróleo *offshore*. Cabos de ancoragem, umbilicais, *risers*, são exemplos de estruturas flexíveis cujos modelos podem ser desenvolvidos com a teoria utilizada na presente dissertação. Porém, toda essa ferramenta de modelagem

necessitava de uma outra ferramenta que permitisse uma visualização mais completa e eficiente dos resultados oriundos das simulações numéricas. O CableSim veio suprir essa necessidade de visualização. Animações tridimensionais com o CableSim permitiram a percepção de uma intensa sensação de realidade física.

Por fim, como trabalhos futuros, destacam-se funcionalidades inerentes a questões de usabilidade do software, como implementar um manual diretamente na aplicação, e outras referentes ao OpenGL, como gerenciar melhor a visualização dos objetos da cena e aplicar textura no cabo.

7. REFERENCIAS BIBLIOGRÁFICAS

- [1] ZANELA, E. B. **Modelagem Analítica de Estruturas do Tipo Cabo para Aplicações Subaquáticas**. Rio Grande, RS, 2013. 137 p. Dissertação (Mestrado em Engenharia Oceânica), FURG – Universidade Federal do Rio Grande.
- [2] PEREIRA, A. E. L. **O Método da Decomposição de Adomian Aplicado à Interação Fluido-estrutura de um Cabo**. Porto Alegre, RS, 2010. 178 p. Tese (Doutorado em Matemática Aplicada), UFRGS – Universidade Federal do Rio Grande do Sul.
- [3] FABER, I.; VIEIRA, A. C. **Gerador Automático de Modelos Dinâmicos de Cabos Umbilicais Subaquáticos Utilizando Formulação Discreta**, Trabalho de Conclusão de Curso, Universidade Federal do Rio Grande, Curso de Engenharia de Computação, Rio Grande, (2009).
- [4] MATHWORKS. **MATLAB Documentation**. Disponível em: <<http://www.mathworks.com/help/matlab/index.html>>. Acesso em 01 de agosto de 2014.
- [5] KHRONOS Group. **OpenGL – The Industry's Foundation for High Performance Graphics**. Disponível em < <https://www.khronos.org/opengl/>>. Acesso em 10 de agosto de 2015.
- [6] PEREIRA, A. E. L. **Um Estudo sobre Modelagem Matemática de Estruturas Flexíveis**. Rio Grande, RS, 1999. 144 p. Dissertação (Mestre em Matemática Aplicada), FURG – Universidade Federal do Rio Grande.
- [7] ANGEL, E.; SHREINER, D. **Teaching a Shader-Based Introduction to Computer Graphics**. IEEE Computer Graphics and Applications, vl.31, n.2, pp. 9-13, 2011.
- [8] OWENS, J. D.; et al. **GPU Computing**. Proceedings of the IEEE, vl.96, n.5, pp. 879-899, 2008.
- [9] ROSSANT, C. **The Power of Shaders in Real-Time Graphics Programming**. Disponível em: <<http://cyrille.rossant.net/shaders-opengl/>>. Acesso em 04 de fevereiro de 2015.
- [10] ROCHA, I. B. P. **Modelagem Matemática de Cabos Umbilicais e Veículos Subaquáticos Não Tripulados**. Rio Grande, RS, 2013. 116 p. Dissertação (Mestrado em Modelagem Computacional), FURG – Universidade Federal do Rio Grande.

- [11] POUZADA, T. Á. **Algoritmos Genéricos para a Geração Automática de Modelos de Cabos em Três Situações Práticas de Aplicações Subaquáticas**. Rio Grande, RS, 2015. 145 p. Dissertação (Mestrado em Engenharia Oceânica), FURG – Universidade Federal do Rio Grande.
- [12] OLIVEIRA, V. S. D. **Algoritmos Genéricos para a Geração de Modelos Dinâmicos de Cabos Umbilicais e Veículos Subaquáticos**. Rio Grande, RS, 2015. 179 p. Dissertação (Mestrado em Modelagem Computacional), FURG – Universidade Federal do Rio Grande.
- [13] DUBOIS, P. F. **Python: Batteries Included**. Computing in Science & Engineering, vl.9, n.3, pp. 7-9, 2007.
- [14] OLIPHANT, T. E. **Python for Scientific Computing**. Computing in Science & Engineering, vl.9, n.3, pp. 10-20, 2007.
- [15] MILLMAN, K. J.; AIVAZIS, M. **Python for Scientists and Engineers**. Computing in Science & Engineering, vl.13, n.2, pp. 9-12, 2011.
- [16] WALT, S. V. D.; COLBERT, S. C.; VAROQUAUX, G. **The NumPy Array: A Structure for Efficient Numerical Computation**. Computing in Science & Engineering, vl.13, n.2, pp. 22-30, 2011.
- [17] RASHED, M. G.; AHSAN, R. **Python in Computational Science: Applications and Possibilities**. International Journal of Computer Applications, vl.46, n.20, pp. 26-30, 2012.
- [18] REINA, G.; MÜLHER, T.; ERTL, T. **Incorporating Modern OpenGL into Computer Graphics Education**. IEEE Computer Graphics and Applications, vl.34, n.4, pp. 16-21, 2012.
- [19] SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
- [20] BALZERT, H. **UML 2: compacto**. Rio de Janeiro: Elsevier, 2008.
- [21] ERICKSON, J.; SIAU, K. **Theoretical and practical complexity of modeling methods**. Communications of the ACM, vl.50, n.8, pp. 46-51, 2007.
- [22] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Elsevier, 2005.
- [23] SCHACH, S. R. **Engenharia de software: os paradigmas clássico e orientado a**

- objetos. 7. ed. São Paulo: McGraw-Hill, 2010.
- [24] BARRY, P. **Use a cabeça Python**. Rio de Janeiro, RJ: Alta Books, 2012.
- [25] LARMAN, C. **Utilizando Uml e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. 3. ed. Porto Alegre: Bookman, 2007.
- [26] QT Company. **Qt**. Disponível em: <<http://www.qt.io/>>. Acesso em 10 de janeiro de 2015.
- [27] PYTHON *Software* Foundation. **Linguagem de programação Python**. Disponível em <<http://www.python.org/>>. Acesso em 11 de julho de 2014.
- [28] NUMPY Developers. **About Numpy**. Disponível em <<http://www.numpy.org/>>. Acesso em 25 de agosto de 2014.
- [29] MATPLOTLIB Development Team. **Matplotlib 1.5.1 documentation**. Disponível em <<http://matplotlib.org/>>. Acesso em 25 de agosto de 2014.
- [30] HUNTER, J. D. **Matplotlib: A 2d Graphics Environment**. *Computing in Science & Engineering*, vl.9, n.3, pp. 90-95, 2007.
- [31] BORGES, L. E. **Python para Desenvolvedores**. 2. ed. Rio de Janeiro, 2010. Disponível em: <http://ark4n.files.wordpress.com/2010/01/python_para_desenvolvedores_2ed.pdf>. Acesso em 10 de julho de 2014.
- [32] COHEN, M.; MANSSOUR, I. H. **OpenGL: Uma Abordagem Prática e Objetiva**. São Paulo: Novatec Editora Ltda, 2006.
- [33] RIVERBANK Computing Limited. **PyQt**. Disponível em: <<https://riverbankcomputing.com/news>>. Acesso em 26 de agosto de 2014.
- [34] QT Documentation. **Qt Features Overview**. Disponível em: <<http://doc.qt.io/qt-4.8/qt-overview.html>>. Acesso em 20 de janeiro de 2015.
- [35] SUMMERFIELD, M. **Rapid GUI programming with Python and Qt: The Definitive Guide to PyQt Programming**. United States of America: Pearson Education, 2007.
- [36] COLARES, A. F. D. O. **Desenvolvimento de Interface Gráfica como Suporte para Soluções Numéricas das Equações de Maxwell em Coordenadas Gerais-3D**.

- Belém, PA, 2011. 145 p. Dissertação (Mestrado em Engenharia Elétrica), UFPA – Universidade Federal do Pará.
- [37] EFFBOT. **effbot.org**. Disponível em: <<http://effbot.org/>>. Acesso em 25 de outubro de 2015.
- [38] W3C Brasil. **W3C Brasil - Consórcio World Wide Web Escritório Brasil**. Disponível em: <<http://www.w3c.br/Home/WebHome>>. Acesso em 10 de outubro de 2015.
- [39] ELEMENTTREE Overview. **ElementTree Overview**. Disponível em: <<http://effbot.org/zone/element-index.htm>>. Acesso em 25 de outubro de 2015.
- [40] QT Documentation. **Qt Designer Manual**. Disponível em: <<http://qt-project.org/doc/qt-4.8/designer-manual.html>>. Acesso em 21 de agosto de 2014.
- [41] IEEE Spectrum. **IEEE Spectrum: Technology, Engineering, and Science News**. Disponível em: <<http://spectrum.ieee.org/>>. Acesso em 04 de abril de 2016.
- [42] DIAKOPOULOS, N.; CASS, S. **Interactive: The Top Programming Languages 2015**. Disponível em: <<http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>>. Acesso em 04 de abril de 2016.
- [43] IEEE Spectrum. **The 2015 Top Ten Programming Languages**. Disponível em: <<http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>>. Acesso em 04 de abril de 2016.
- [44] MICROSOFT Windows. **MFC Desktop Applications**. Disponível em: <[https://msdn.microsoft.com/en-us/library/d06h2x6e\(v=VS.120\).aspx](https://msdn.microsoft.com/en-us/library/d06h2x6e(v=VS.120).aspx)>. Acesso em 04 de abril de 2016.
- [45] GTK + Team. **The GTK+ Team**. Disponível em: <<http://www.gtk.org/>>. Acesso em 04 de abril de 2016.
- [46] ORACLE. **Oracle Technology Network for Java Developers**. Disponível em: <<http://www.oracle.com/technetwork/java/index.html>>. Acesso em 04 de abril de 2016.

APÊNDICE A - ELEMENTOS DAS MATRIZES E VETORES PARA O CABO DIVIDIDO EM TRÊS E QUATRO ELOS

APÊNDICE A.1 – ELEMENTOS DAS MATRIZES E VETORES PARA O CABO DIVIDIDO EM TRÊS ELOS

Os elementos da matriz de inércia I são dados por:

$$I_{11} = l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_c \right) + I_{R_{1e}}$$

$$I_{12} = l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) (\cos \theta_{2e} \cos \theta_{1e} \cos \theta_{2a} + \text{sen } \theta_{2e} \text{ sen } \theta_{1e})$$

$$I_{13} = l_1 l_3 \left(\frac{m_3}{2} + m_c \right) (\cos \theta_{3e} \cos \theta_{1e} \cos(\theta_{2a} + \theta_{3a}) + \text{sen } \theta_{3e} \text{ sen } \theta_{1e})$$

$$I_{14} = -l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) (\text{sen } \theta_{2e} \cos \theta_{1e} \text{ sen } \theta_{2a}) + \\ -l_1 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a})]$$

$$I_{15} = -l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) (\text{sen } \theta_{2e} \cos \theta_{1e} \text{ sen } \theta_{2a}) + \\ -l_1 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a})]$$

$$I_{16} = -l_1 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a})]$$

$$I_{17} = I_{18} = I_{19} = 0$$

$$I_{21} = I_{12}$$

$$I_{22} = l_2^2 \left(\frac{m_2}{4} + m_3 + m_c \right) + I_{R_{2e}}$$

$$I_{23} = l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\cos \theta_{3e} \cos \theta_{2e} \cos \theta_{3a} + \text{sen } \theta_{3e} \text{ sen } \theta_{2a})$$

$$I_{24} = l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) (\text{sen } \theta_{1e} \cos \theta_{2e} \text{ sen } \theta_{2a}) + \\ -l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen } \theta_{3e} \cos \theta_{2e} \text{ sen } \theta_{3a})$$

$$I_{25} = -l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen } \theta_{3e} \cos \theta_{2e} \text{ sen } \theta_{3a})$$

$$I_{26} = -l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen } \theta_{3e} \cos \theta_{2e} \text{ sen } \theta_{3a})$$

$$I_{27} = I_{28} = I_{29} = 0$$

$$I_{31} = I_{13}$$

$$I_{32} = I_{23}$$

$$I_{33} = l_3^2 \left(\frac{m_3}{4} + m_c \right) + I_{R_{3e}}$$

$$I_{34} = l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen } \theta_{2e} \cos \theta_{3e} \text{ sen } \theta_{3a}) + \\ + l_1 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{1e} \text{ sen } \theta_{3e} \text{ sen}(\theta_{2a} + \theta_{3a})]$$

$$I_{35} = l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen } \theta_{2e} \cos \theta_{3e} \text{ sen } \theta_{3a})$$

$$I_{36} = I_{37} = I_{38} = I_{39} = 0$$

$$I_{41} = I_{14}$$

$$I_{42} = I_{24}$$

$$I_{43} = I_{34}$$

$$I_{44} = l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_c \right) (\text{sen}^2 \theta_{1e}) + l_2^2 \left(\frac{m_2}{4} + m_3 + m_c \right) (\text{sen}^2 \theta_{2e}) + \\ + l_3^2 \left(\frac{m_3}{4} + m_c \right) (\text{sen}^2 \theta_{3e}) + 2l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) (\text{sen } \theta_{1e} \text{ sen } \theta_{2e} \cos \theta_{2a}) + \\ + 2l_1 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{1e} \text{ sen } \theta_{3e} \cos(\theta_{2a} + \theta_{3a})] + \\ + 2l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen } \theta_{2e} \text{ sen } \theta_{3e} \cos \theta_{3a}) + I_{R_{1a}}$$

$$I_{45} = l_2^2 \left(\frac{m_2}{4} + m_3 + m_c \right) (\text{sen}^2 \theta_{2e}) + l_3^2 \left(\frac{m_3}{4} + m_c \right) (\text{sen}^2 \theta_{3e}) + \\ + l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) (\text{sen } \theta_{1e} \text{ sen } \theta_{2e} \cos \theta_{2a}) + \\ + l_1 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{1e} \text{ sen } \theta_{3e} \cos(\theta_{2a} + \theta_{3a})] + \\ + 2l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen } \theta_{2e} \text{ sen } \theta_{3e} \cos \theta_{3a})$$

$$I_{46} = l_3^2 \left(\frac{m_3}{4} + m_c \right) (\text{sen}^2 \theta_{3e}) + l_1 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{1e} \text{ sen } \theta_{3e} \cos(\theta_{2a} + \theta_{3a})] + \\ + l_2 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen } \theta_{2e} \text{ sen } \theta_{3e} \cos \theta_{3a}]$$

$$I_{47} = I_{48} = I_{49} = 0$$

$$I_{51} = I_{15}$$

$$I_{52} = I_{25}$$

$$I_{53} = I_{35}$$

$$I_{54} = I_{45}$$

$$\begin{aligned}
 I_{55} &= l_2^2 \left(\frac{m_2}{4} + m_3 + m_c \right) (\text{sen}^2 \theta_{2e}) + l_3^2 \left(\frac{m_3}{4} + m_c \right) (\text{sen}^2 \theta_{3e}) \\
 &\quad + 2l_2 l_3 \left(\frac{m_3}{2} + m_c \right) (\text{sen} \theta_{2e} \text{sen} \theta_{3e} \text{sen} \theta_{3a}) + I_{R_{2a}} \\
 I_{56} &= l_3^2 \left(\frac{m_3}{4} + m_c \right) [\text{sen}^2 \theta_{3e}] + l_2 l_3 \left(\frac{m_3}{2} + m_c \right) [\text{sen} \theta_{2e} \text{sen} \theta_{3e} \cos \theta_{3a}] \\
 I_{57} &= I_{58} = I_{59} = 0 \\
 I_{61} &= I_{16} \\
 I_{62} &= I_{26} \\
 I_{63} &= I_{36} \\
 I_{64} &= I_{46} \\
 I_{65} &= I_{56} \\
 I_{66} &= l_3^2 \left(\frac{m_3}{4} + m_c \right) (\text{sen}^2 \theta_{3e}) + I_{R_{3a}} \\
 I_{67} &= I_{68} = I_{69} = 0 \\
 I_{71} &= I_{72} = I_{73} = I_{74} = I_{75} = I_{76} = 0 \\
 I_{77} &= I_{T1} \\
 I_{78} &= I_{79} = 0 \\
 I_{81} &= I_{82} = I_{83} = I_{84} = I_{85} = I_{86} = I_{87} = 0 \\
 I_{88} &= I_{T2} \\
 I_{89} &= 0 \\
 I_{91} &= I_{92} = I_{93} = I_{94} = I_{95} = I_{96} = I_{97} = I_{98} = 0 \\
 I_{99} &= I_{T3} + I_{Tc}
 \end{aligned}$$

Os elementos da matriz das constantes de atrito C são definidos por:

$$\begin{aligned}
 C_{11} &= C_{1e} + C_{2e}; \\
 C_{12} &= -C_{2e}; \\
 C_{13} &= C_{14} = C_{15} = C_{16} = C_{17} = C_{18} = C_{19} = 0; \\
 C_{21} &= C_{12}; \\
 C_{22} &= C_{2e} + C_{3e}; \\
 C_{23} &= -C_{3e}; \\
 C_{24} &= C_{25} = C_{26} = C_{27} = C_{28} = C_{29} = 0; \\
 C_{31} &= C_{13};
 \end{aligned}$$

$$\begin{aligned}C_{32} &= C_{23}; \\C_{33} &= C_{3e}; \\C_{34} &= C_{35} = C_{36} = C_{37} = C_{38} = C_{39} = 0; \\C_{41} &= C_{42} = C_{43} = 0; \\C_{44} &= C_{1a} + C_{2a}; \\C_{45} &= -C_{2a}; \\C_{46} &= C_{47} = C_{48} = C_{49} = 0; \\C_{51} &= C_{52} = C_{53} = 0; \\C_{54} &= C_{45}; \\C_{55} &= C_{2a} + C_{3a}; \\C_{56} &= -C_{3a}; \\C_{57} &= C_{58} = C_{59} = 0; \\C_{61} &= C_{62} = C_{63} = C_{64} = 0; \\C_{65} &= C_{56}; \\C_{66} &= C_{3a}; \\C_{67} &= C_{68} = C_{69} = 0; \\C_{71} &= C_{72} = C_{73} = C_{74} = C_{75} = C_{76} = 0; \\C_{77} &= C_{1T} + C_{2T}; \\C_{78} &= -C_{2T}; \\C_{79} &= 0; \\C_{81} &= C_{82} = C_{83} = C_{84} = C_{85} = C_{86} = 0; \\C_{87} &= C_{78}; \\C_{88} &= C_{2T} + C_{3T}; \\C_{89} &= -C_{3T}; \\C_{91} &= C_{92} = C_{93} = C_{94} = C_{95} = C_{96} = C_{97} = 0; \\C_{98} &= C_{89}; \\C_{99} &= C_{3T};\end{aligned}$$

Os elementos da matriz das constantes de elásticas K são dados por:

$$\begin{aligned}K_{11} &= K_{1e} + K_{2e}; \\K_{12} &= -K_{2e};\end{aligned}$$

$$K_{13} = K_{14} = K_{15} = K_{16} = K_{17} = K_{18} = K_{19} = 0;$$

$$K_{21} = K_{12};$$

$$K_{22} = K_{2e} + K_{3e};$$

$$K_{23} = -K_{3e}; K_{24} = K_{25} = K_{26} = K_{27} = K_{28} = K_{29} = 0;$$

$$K_{31} = K_{13};$$

$$K_{32} = K_{23};$$

$$K_{33} = K_{3e};$$

$$K_{34} = K_{35} = K_{36} = K_{37} = K_{38} = K_{39} = 0;$$

$$K_{41} = K_{42} = K_{43} = 0;$$

$$K_{44} = K_{1a} + K_{2a};$$

$$K_{45} = -K_{2a};$$

$$K_{46} = K_{47} = K_{48} = K_{49} = 0;$$

$$K_{51} = K_{52} = K_{53} = 0;$$

$$K_{54} = K_{45};$$

$$K_{55} = K_{2a} + K_{3a};$$

$$K_{56} = -K_{3a};$$

$$K_{57} = K_{58} = K_{59} = 0;$$

$$K_{61} = K_{62} = K_{63} = K_{64} = 0;$$

$$K_{65} = K_{56};$$

$$K_{66} = K_{3a};$$

$$K_{67} = K_{68} = K_{69} = 0;$$

$$K_{71} = K_{72} = K_{73} = K_{74} = K_{75} = K_{76} = 0;$$

$$K_{77} = K_{1T} + K_{2T};$$

$$K_{78} = -K_{2T};$$

$$K_{79} = 0;$$

$$K_{81} = K_{82} = K_{83} = K_{84} = K_{85} = K_{86} = 0;$$

$$K_{87} = K_{78};$$

$$K_{88} = K_{2T} + K_{3T};$$

$$K_{89} = -K_{3T};$$

$$K_{91} = K_{92} = K_{93} = K_{94} = K_{95} = K_{96} = K_{97} = 0;$$

$$K_{98} = K_{89};$$

$$K_{99} = K_{3T};$$

Os elementos do vetor coriolis-centrífugo \vec{F} são definidos conforme segue:

$$\begin{aligned}
 f_1 = & -2l_1l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) \left\{ \cos \theta_{1e} \left[\cos \theta_{2e} \sin \theta_{2a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} + \right. \right. \\
 & \left. \left. + \sin \theta_{2e} \cos \theta_{2a} (\dot{\theta}_{1a} \dot{\theta}_{2a} + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{2e}^2}{2} \right)) \right] \right\} - \frac{1}{2} (\sin \theta_{1e} \cos \theta_{2e}) \dot{\theta}_{2e}^2 + \\
 & -2l_1l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{1e} \left[\cos \theta_{3e} \sin(\theta_{2a} + \theta_{3a}) (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} + \right. \right. \\
 & \left. \left. + \sin \theta_{3e} \cos(\theta_{2a} + \theta_{3a}) \left((\dot{\theta}_{1a} \dot{\theta}_{2a} + \dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{3a}) + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} \right) \right) \right] \right\} + \\
 & -\frac{1}{2} [\sin \theta_{1e} \cos \theta_{3e}] \dot{\theta}_{3e}^2 \left\} - l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_c \right) \sin \theta_{1e} \cos \theta_{1e} \dot{\theta}_{1a}^2
 \end{aligned}$$

$$\begin{aligned}
 f_2 = & -2l_1l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) \left\{ \cos \theta_{2e} \left[-\cos \theta_{1e} \sin \theta_{2a} \dot{\theta}_{1a} \dot{\theta}_{1e} + \right. \right. \\
 & \left. \left. + \sin \theta_{1e} \cos \theta_{2a} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{1e}^2}{2} \right) \right] \right\} - \frac{1}{2} \cos \theta_{1e} \sin \theta_{2e} \dot{\theta}_{1e}^2 \left\} + \right. \\
 & -2l_1l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{2e} \left[\cos \theta_{3e} \sin \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} + \right. \right. \\
 & \left. \left. + \sin \theta_{3e} \cos \theta_{3a} \left((\dot{\theta}_{1a} \dot{\theta}_{2a} + \dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{3a}) + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} \right) \right) \right] \right\} + \\
 & -\frac{1}{2} \sin \theta_{2e} \cos \theta_{3e} \dot{\theta}_{3e}^2 \left\} - 2l_2^2 \left(\frac{m_2}{4} + m_3 + m_c \right) \sin \theta_{2e} \cos \theta_{2e} \left[\dot{\theta}_{1a} \dot{\theta}_{2a} + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2}{2} \right) \right]
 \end{aligned}$$

$$\begin{aligned}
 f_3 = & -2l_1l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{3e} \left[-\cos \theta_{1e} \sin(\theta_{2a} + \theta_{3a}) \dot{\theta}_{1a} \dot{\theta}_{1e} + \right. \right. \\
 & \left. \left. + \sin \theta_{1e} \cos(\theta_{2a} + \theta_{3a}) \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{1e}^2}{2} \right) \right] \right\} - \frac{1}{2} \cos \theta_{1e} \sin \theta_{3a} \dot{\theta}_{1e}^2 \left\} + \right. \\
 & -2l_2l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{3e} \left[-\cos \theta_{2e} \sin \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} + \sin \theta_{2e} \cos \theta_{3a} (\dot{\theta}_{1a} \dot{\theta}_{2a} + \right. \right. \\
 & \left. \left. + \frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{1e}^2}{2}) \right] \right\} - \frac{1}{2} \cos \theta_{2e} \sin \theta_{3e} \dot{\theta}_{2e}^2 \left\} - 2l_3^2 \left(\frac{m_3}{4} + m_c \right) \left\{ \sin \theta_{3e} \cos \theta_{3e} \left[(\dot{\theta}_{1a} \dot{\theta}_{2a} + \right. \right. \right. \\
 & \left. \left. + \dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{3a}) + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2}{2} \right) \right] \right\}
 \end{aligned}$$

$$\begin{aligned}
 f_4 = & -2l_1l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) \left\{ \cos \theta_{1e} \sin \theta_{2e} \cos \theta_{2a} \dot{\theta}_{1a} \dot{\theta}_{1e} + \right. \\
 & + \sin \theta_{1e} \cos \theta_{2e} \cos \theta_{2a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} - \sin \theta_{1e} \sin \theta_{2e} \sin \theta_{2a} [\dot{\theta}_{1a} \dot{\theta}_{2a} + \\
 & \left. + \left(\frac{\dot{\theta}_{2a}^2 + \dot{\theta}_{1e}^2 + \dot{\theta}_{2e}^2}{2} \right) \right\} + 2l_1l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{1e} \sin \theta_{3e} \cos(\theta_{2a} + \theta_{3a}) \dot{\theta}_{1a} \dot{\theta}_{1e} + \right. \\
 & + \sin \theta_{1e} \cos \theta_{3e} \cos(\theta_{2a} + \theta_{3a}) (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} + \\
 & - \sin \theta_{1e} \sin \theta_{3e} \sin(\theta_{2a} + \theta_{3a}) [(\dot{\theta}_{1a} \dot{\theta}_{2a} + \dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{3a}) + \\
 & \left. + \left(\frac{\dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 - \dot{\theta}_{1e}^2 + \dot{\theta}_{3e}^2}{2} \right) \right\} + 2l_2l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{2e} \sin \theta_{3e} \cos \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} + \right. \\
 & + \sin \theta_{2e} \cos \theta_{3e} \cos \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} + \\
 & \left. - \sin \theta_{2e} \sin \theta_{3e} \sin \theta_{3a} \left[\dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{3a} + \left(\frac{\dot{\theta}_{3a}^2 - \dot{\theta}_{2e}^2 + \dot{\theta}_{3e}^2}{2} \right) \right] \right\} + \\
 & + 2l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_c \right) (\sin \theta_{1e} \cos \theta_{1e} \dot{\theta}_{1a} \dot{\theta}_{1e}) + \\
 & + 2l_2^2 \left(\frac{m_2}{4} + m_3 + m_c \right) [\sin \theta_{2e} \cos \theta_{2e} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e}] + \\
 & + 2l_3^2 \left(\frac{m_3}{4} + m_c \right) [\sin \theta_{3e} \cos \theta_{3e} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e}] \\
 f_5 = & 2l_1l_2 \left(\frac{m_2}{2} + m_3 + m_c \right) \left\{ \cos \theta_{1e} \sin \theta_{2e} \cos \theta_{2a} \dot{\theta}_{1a} \dot{\theta}_{1e} + \right. \\
 & + \sin \theta_{1e} \sin \theta_{2e} \sin \theta_{2a} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{1e}^2}{2} \right) \left. \right\} + \\
 & + 2l_1l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{1e} \sin \theta_{3e} \cos(\theta_{2a} + \theta_{3a}) \dot{\theta}_{1a} \dot{\theta}_{1e} + \right. \\
 & + \sin \theta_{1e} \sin \theta_{3e} \sin(\theta_{2a} + \theta_{3a}) \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{1e}^2}{2} \right) \left. \right\} + \\
 & + 2l_2l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{2e} \sin \theta_{3e} \cos \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} + \right. \\
 & + \sin \theta_{2e} \cos \theta_{3e} \cos \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} + \\
 & \left. - \sin \theta_{2e} \sin \theta_{3e} \sin \theta_{3a} \left[(\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{3a} + \left(\frac{\dot{\theta}_{3a}^2 - \dot{\theta}_{2e}^2 + \dot{\theta}_{3e}^2}{2} \right) \right] \right\} + \\
 & + 2l_2^2 \left(\frac{m_2}{4} + m_3 + m_c \right) [\sin \theta_{2e} \cos \theta_{2e} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e}] + \\
 & + 2l_3^2 \left(\frac{m_3}{4} + m_c \right) [\sin \theta_{3e} \cos \theta_{3e} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e}]
 \end{aligned}$$

$$\begin{aligned}
 f_6 = & 2l_1l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{1e} \sin \theta_{3e} \cos(\theta_{2a} + \theta_{3a}) \dot{\theta}_{1a} \dot{\theta}_{1e} + \right. \\
 & \left. + \sin \theta_{1e} \sin \theta_{3e} \sin(\theta_{2a} + \theta_{3a}) \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{1e}^2}{2} \right) \right\} + \\
 & + 2l_2l_3 \left(\frac{m_3}{2} + m_c \right) \left\{ \cos \theta_{2e} \sin \theta_{3e} \cos \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} + \right. \\
 & \left. + \sin \theta_{2e} \sin \theta_{3e} \sin \theta_{3a} \left[\dot{\theta}_{1a} + \dot{\theta}_{2a} + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{2e}^2}{2} \right) \right] \right\} + \\
 & + 2l_3^2 \left(\frac{m_3}{4} + m_c \right) \left[\sin \theta_{3e} \cos \theta_{3e} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} \right] \\
 f_7 = & f_8 = f_9 = 0
 \end{aligned}$$

Os elementos do vetor gravitacional \vec{G} são dados por:

$$\begin{aligned}
 g_1 &= l_1 \left(\frac{m_1}{2} + m_2 + m_3 \right) g \sin \theta_{1e} \\
 g_2 &= l_2 \left(\frac{m_2}{2} + m_3 \right) g \sin \theta_{2e} \\
 g_3 &= l_3 \left(\frac{m_3}{2} \right) g \sin \theta_{3e} \\
 g_4 &= g_5 = g_6 = g_7 = g_8 = g_9 = 0
 \end{aligned}$$

Os elementos do vetor de torques externos \vec{T} são:

$$\begin{aligned}
 T_{m_1} &= T_{\theta_{1e}}; \\
 T_{m_2} &= T_{\theta_{2e}}; \\
 T_{m_3} &= T_{\theta_{3e}}; \\
 T_{m_4} &= T_{\theta_{1a}}; \\
 T_{m_5} &= T_{\theta_{2a}}; \\
 T_{m_6} &= T_{\theta_{3a}}; \\
 T_{m_7} &= T_{\theta_{1T}}; \\
 T_{m_8} &= T_{\theta_{2T}}; \\
 T_{m_9} &= T_{\theta_{3T}};
 \end{aligned}$$

APÊNDICE A.2 – ELEMENTOS DAS MATRIZES E VETORES PARA O CABO DIVIDIDO EM QUATRO ELOS

Os elementos da matriz de inércia I são dados por:

$$\begin{aligned}
 I_{11} &= l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_4 + m_c \right) + I_{R_{1e}} \\
 I_{12} &= l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\cos \theta_{2e} \cos \theta_{1e} \cos \theta_{2a} + \text{sen } \theta_{2e} \text{ sen } \theta_{1e}] \\
 I_{13} &= l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{3e} \cos \theta_{1e} \cos(\theta_{2a} + \theta_{3a}) + \text{sen } \theta_{3e} \text{ sen } \theta_{1e}] \\
 I_{14} &= l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{4e} \cos \theta_{1e} \cos(\theta_{2a} + \theta_{3a} + \theta_{4a}) + \text{sen } \theta_{4e} \text{ sen } \theta_{1e}] \\
 I_{15} &= -l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\text{sen } \theta_{2e} \cos \theta_{1e} \text{ sen } \theta_{2a}] + \\
 &\quad -l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a})] \\
 &\quad -l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})] \\
 I_{16} &= -l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\text{sen } \theta_{2e} \cos \theta_{1e} \text{ sen } \theta_{2a}] + \\
 &\quad -l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a})] + \\
 &\quad -l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})] \\
 I_{17} &= -l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a})] + \\
 &\quad -l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})] \\
 I_{18} &= -l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})] \\
 I_{19} &= I_{1,10} = I_{1,11} = I_{1,12} = 0 \\
 I_{21} &= I_{12} \\
 I_{22} &= l_2^2 \left(\frac{m_2}{4} + m_3 + m_4 + m_c \right) + I_{R_{2e}} \\
 I_{23} &= l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) + [\cos \theta_{3e} \cos \theta_{2e} \cos \theta_{3a} + \text{sen } \theta_{3e} \text{ sen } \theta_{2e}] \\
 I_{24} &= l_2 l_4 \left(\frac{m_4}{2} + m_c \right) + [\cos \theta_{4e} \cos \theta_{2e} \cos(\theta_{3a} + \theta_{4a}) + \text{sen } \theta_{4e} \text{ sen } \theta_{2e}]
 \end{aligned}$$

$$I_{25} = l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) (\text{sen } \theta_{1e} \cos \theta_{2e} \text{sen } \theta_{2a}) + \\ - l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{2e} \text{sen } \theta_{3a}] + \\ - l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{2e} \text{sen}(\theta_{3a} + \theta_{4a})]$$

$$I_{26} = -l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) (\text{sen } \theta_{3e} \cos \theta_{2e} \text{sen } \theta_{3a}) + \\ - l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{2e} \text{sen}(\theta_{3a} + \theta_{4a})]$$

$$I_{27} = -l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) (\text{sen } \theta_{3e} \cos \theta_{2e} \text{sen } \theta_{3a}) + \\ - l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{2e} \text{sen}(\theta_{3a} + \theta_{4a})]$$

$$I_{28} = -l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{2e} \text{sen}(\theta_{3a} + \theta_{4a})]$$

$$I_{29} = I_{2,10} = I_{2,11} = I_{2,12} = 0$$

$$I_{31} = I_{13}$$

$$I_{32} = I_{23}$$

$$I_{33} = l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) + I_{R_{3e}}$$

$$I_{34} = l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{4e} \cos \theta_{3e} \cos \theta_{4a} + \text{sen } \theta_{4e} \text{sen } \theta_{3e}]$$

$$I_{35} = l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) \text{sen } \theta_{1e} \cos \theta_{3e} \text{sen}(\theta_{2a} + \theta_{3a}) + \\ + l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen } \theta_{2e} \cos \theta_{3e} \text{sen} \theta_{3a}] + \\ - l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{3e} \text{sen} \theta_{4a}]$$

$$I_{36} = l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen } \theta_{2e} \cos \theta_{3e} \text{sen} \theta_{3a}] + \\ - l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{3e} \text{sen} \theta_{4a}]$$

$$I_{37} = -l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{3e} \text{sen} \theta_{4a}]$$

$$I_{38} = -l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{4e} \cos \theta_{3e} \text{sen} \theta_{4a}]$$

$$I_{39} = I_{3,10} = I_{3,11} = I_{3,12} = 0$$

$$I_{41} = l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{1e} \cos \theta_{4e} \cos(\theta_{2a} + \theta_{3a} + \theta_{4a}) \text{sen } \theta_{1e} \text{sen } \theta_{4e}]$$

$$I_{42} = l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{2e} \cos \theta_{4e} \cos(\theta_{3a} + \theta_{4a}) + \text{sen } \theta_{2e} \text{ sen } \theta_{4e}]$$

$$I_{43} = l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{3e} \cos \theta_{4e} \cos \theta_{4a} + \text{sen } \theta_{3e} \text{ sen } \theta_{4e}]$$

$$I_{44} = l_4^2 \left(\frac{m_4}{4} + m_c \right) + I_{R4e}$$

$$I_{45} = l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{1e} \cos \theta_{4e} \text{ sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})] + \\ + l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{2e} \cos \theta_{4e} \text{ sen}(\theta_{3a} + \theta_{4a})] + \\ + l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{4e} \text{ sen } \theta_{4a}]$$

$$I_{46} = l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{2e} \cos \theta_{4e} \text{ sen}(\theta_{3a} + \theta_{4a})] + \\ + l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{4e} \text{ sen } \theta_{4a}]$$

$$I_{47} = l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen } \theta_{3e} \cos \theta_{4e} \text{ sen } \theta_{4a}]$$

$$I_{48} = I_{49} = I_{4,10} = I_{4,11} = I_{4,12} = 0$$

$$I_{51} = -l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\cos \theta_{1e} \text{ sen } \theta_{2e} \text{ sen } \theta_{2a}] + \\ - l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{1e} \text{ sen } \theta_{3e} \text{ sen}(\theta_{2a} + \theta_{3a})] + \\ - l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{1e} \text{ sen } \theta_{4e} \text{ sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})]$$

$$I_{52} = l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\cos \theta_{2e} \text{ sen } \theta_{1e} \text{ sen } \theta_{2a}] + \\ - l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{2e} \text{ sen } \theta_{3e} \text{ sen } \theta_{3a}] + \\ - l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{2e} \text{ sen } \theta_{4e} \text{ sen}(\theta_{3a} + \theta_{4a})]$$

$$I_{53} = l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{3e} \text{ sen } \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a})] + \\ + l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{3e} \text{ sen } \theta_{2e} \text{ sen } \theta_{3a}] + \\ - l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{3e} \text{ sen } \theta_{4e} \text{ sen } \theta_{4a}]$$

$$I_{54} = l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{4e} \text{ sen } \theta_{1e} \text{ sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})] + \\ + l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{4e} \text{ sen } \theta_{2e} \text{ sen}(\theta_{3a} + \theta_{4a})] +$$

$$+l_3l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{4e} \operatorname{sen} \theta_{3e} \operatorname{sen} \theta_{4a}]$$

$$\begin{aligned} I_{55} = & l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_4 + m_c \right) [\operatorname{sen}^2 \theta_{1e}] + \\ & + l_2^2 \left(\frac{m_2}{4} + m_3 + m_4 + m_c \right) [\operatorname{sen}^2 \theta_{2e}] + \\ & + l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) [\operatorname{sen}^2 \theta_{3e}] + \\ & + l_4^2 \left(\frac{m_4}{4} + m_c \right) [\operatorname{sen}^2 \theta_{4e}] + \\ & + 2l_1l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\operatorname{sen} \theta_{2e} \operatorname{sen} \theta_{1e} \cos \theta_{2a}] + \\ & + 2l_1l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\operatorname{sen} \theta_{3e} \operatorname{sen} \theta_{1e} \cos(\theta_{2a} + \theta_{3a})] + \\ & + 2l_1l_4 \left(\frac{m_4}{2} + m_c \right) [\operatorname{sen} \theta_{1e} \operatorname{sen} \theta_{4e} \cos(\theta_{2a} + \theta_{3a} + \theta_{4a})] + \\ & + 2l_2l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\operatorname{sen} \theta_{3e} \operatorname{sen} \theta_{2e} \cos \theta_{3a}] + \\ & + 2l_2l_4 \left(\frac{m_4}{2} + m_c \right) [\operatorname{sen} \theta_{4e} \operatorname{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\ & + 2l_3l_4 \left(\frac{m_4}{2} + m_c \right) [\operatorname{sen} \theta_{4e} \operatorname{sen} \theta_{3e} \cos \theta_{4a}] + I_{R1a} \end{aligned}$$

$$\begin{aligned} I_{56} = & l_2^2 \left(\frac{m_2}{4} + m_3 + m_4 + m_c \right) [\operatorname{sen}^2 \theta_{2e}] + \\ & + l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) [\operatorname{sen}^2 \theta_{3e}] + \\ & + l_4^2 \left(\frac{m_4}{4} + m_c \right) [\operatorname{sen}^2 \theta_{4e}] + \\ & + l_1l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\operatorname{sen} \theta_{2e} \operatorname{sen} \theta_{1e} \cos \theta_{2a}] + \\ & + l_1l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\operatorname{sen} \theta_{3e} \operatorname{sen} \theta_{1e} \cos(\theta_{2a} + \theta_{3a})] + \\ & + l_1l_4 \left(\frac{m_4}{2} + m_c \right) [\operatorname{sen} \theta_{1e} \operatorname{sen} \theta_{4e} \cos(\theta_{2a} + \theta_{3a} + \theta_{4a})] + \\ & + 2l_2l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\operatorname{sen} \theta_{3e} \operatorname{sen} \theta_{2e} \cos \theta_{3a}] + \\ & + 2l_2l_4 \left(\frac{m_4}{2} + m_c \right) [\operatorname{sen} \theta_{4e} \operatorname{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\ & + 2l_3l_4 \left(\frac{m_4}{2} + m_c \right) [\operatorname{sen} \theta_{4e} \operatorname{sen} \theta_{3e} \cos \theta_{4a}] \end{aligned}$$

$$\begin{aligned}
 I_{57} = & l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) [\text{sen}^2 \theta_{3e}] + \\
 & + l_4^2 \left(\frac{m_4}{4} + m_c \right) [\text{sen}^2 \theta_{4e}] + \\
 & + l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{3e} \text{sen} \theta_{1e} \cos(\theta_{2a} + \theta_{3a})] + \\
 & + l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{1e} \text{sen} \theta_{4e} \cos(\theta_{2a} + \theta_{3a} + \theta_{4a})] + \\
 & + l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{3e} \text{sen} \theta_{2e} \cos \theta_{3a}] + \\
 & + l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\
 & + 2 l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{3e} \cos \theta_{4a}]
 \end{aligned}$$

$$\begin{aligned}
 I_{58} = & l_4^2 \left(\frac{m_4}{4} + m_c \right) [\text{sen}^2 \theta_{4e}] + \\
 & + l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{1e} \text{sen} \theta_{4e} \cos(\theta_{2a} + \theta_{3a} + \theta_{4a})] + \\
 & + l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\
 & + l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{3e} \cos \theta_{4a}]
 \end{aligned}$$

$$I_{59} = I_{5,10} = I_{5,11} = I_{5,12} = 0$$

$$\begin{aligned}
 I_{61} = & -l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\cos \theta_{1e} \text{sen} \theta_{2e} \text{sen} \theta_{2a}] + \\
 & -l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{1e} \text{sen} \theta_{3e} \text{sen}(\theta_{2a} + \theta_{3a})] + \\
 & -l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{1e} \text{sen} \theta_{4e} \text{sen}(\theta_{2a} + \theta_{3a} + \theta_{4a})]
 \end{aligned}$$

$$\begin{aligned}
 I_{62} = & -l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{2e} \text{sen} \theta_{3e} \text{sen} \theta_{3a}] + \\
 & -l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{2e} \text{sen} \theta_{4e} \text{sen}(\theta_{3a} + \theta_{4a})]
 \end{aligned}$$

$$\begin{aligned}
 I_{63} = & l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{3e} \text{sen} \theta_{2e} \text{sen} \theta_{3a}] + \\
 & -l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{2e} \text{sen} \theta_{4e} \text{sen} \theta_{4a}]
 \end{aligned}$$

$$\begin{aligned}
 I_{64} = & l_2 l_4 (m_4 + m_c) [\cos \theta_{4e} \text{sen} \theta_{2e} \text{sen} (\theta_{3a} + \theta_{4a})] + \\
 & + l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{4e} \text{sen} \theta_{3e} \text{sen} \theta_{4a}]
 \end{aligned}$$

$$\begin{aligned}
 I_{65} = & l_2^2 \left(\frac{m_2}{4} + m_3 + m_4 + m_c \right) [\text{sen}^2 \theta_{2e}] + \\
 & + l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) [\text{sen}^2 \theta_{3e}] + \\
 & + l_4^2 \left(\frac{m_4}{4} + m_c \right) [\text{sen}^2 \theta_{4e}] + \\
 & + l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) [\text{sen} \theta_{1e} \text{sen} \theta_{2e} \cos \theta_{2a}] + \\
 & + l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{1e} \text{sen} \theta_{3e} \cos(\theta_{2a} + \theta_{3a})] + \\
 & + l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{1e} \text{sen} \theta_{4e} \cos(\theta_{2a} + \theta_{3a} + \theta_{4a})] + \\
 & + 2l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{3e} \text{sen} \theta_{2e} \cos \theta_{3a}] + \\
 & + 2l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\
 & + 2l_3 l_4 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{3e} \cos \theta_{4a}]
 \end{aligned}$$

$$\begin{aligned}
 I_{66} = & l_2^2 (m_3 + m_4 + m_c) [\text{sen}^2 \theta_{2e}] + \\
 & + l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) [\text{sen}^2 \theta_{3e}] + \\
 & + l_4^2 \left(\frac{m_4}{4} + m_c \right) [\text{sen}^2 \theta_{4e}] + \\
 & + 2l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{3e} \text{sen} \theta_{2e} \cos \theta_{3a}] + \\
 & + 2l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\
 & + 2l_3 l_4 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{3e} \cos \theta_{4a}] + I_{R2a}
 \end{aligned}$$

$$\begin{aligned}
 I_{67} = & l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) [\text{sen}^2 \theta_{3e}] + \\
 & + l_4^2 \left(\frac{m_4}{4} + m_c \right) [\text{sen}^2 \theta_{4e}] + \\
 & + l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{3e} \text{sen} \theta_{2e} \cos \theta_{3a}] + \\
 & + l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\
 & + 2l_3 l_4 \left(\frac{m_4}{2} + m_4 + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{3e} \cos \theta_{4a}]
 \end{aligned}$$

$$\begin{aligned}
 I_{68} = & l_4^2 \left(\frac{m_4}{4} + m_c \right) [\text{sen}^2 \theta_{4e}] + l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{2e} \cos(\theta_{3a} + \theta_{4a})] + \\
 & + l_3 l_4 \left(\frac{m_3}{2} + m_4 + m_c \right) [\text{sen} \theta_{4e} \text{sen} \theta_{3e} \cos \theta_{4a}]
 \end{aligned}$$

$$I_{69} = I_{6,10} = I_{6,11} = I_{6,12} = 0$$

$$I_{71} = -l_1 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{1e} \sin \theta_{3e} \sin(\theta_{2a} + \theta_{3a})] +$$

$$-l_1 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{1e} \sin \theta_{4e} \sin(\theta_{2a} + \theta_{3a} + \theta_{4a})]$$

$$I_{72} = -l_2 l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) [\cos \theta_{2e} \sin \theta_{3e} \sin \theta_{3a}] +$$

$$-l_2 l_4 \left(\frac{m_4}{2} + m_c \right) [\cos \theta_{2e} \sin \theta_{4e} \sin(\theta_{3a} + \theta_{4a})]$$

$$I_{73} = -l_3 l_4 (m_4 + m_c) [\cos \theta_{3e} \sin \theta_{4e} \sin \theta_{4a}]$$

$$I_{74} = l_3 l_4 (m_4 + m_c) [\cos \theta_{4e} \sin \theta_{3e} \sin \theta_{4a}]$$

$$I_{75} = I_{57}$$

$$I_{76} = I_{67}$$

$$I_{77} = l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) [\sin^2 \theta_{3e}] +$$

$$+ l_4^2 \left(\frac{m_4}{4} + m_c \right) [\sin^2 \theta_{4e}] +$$

$$+ 2l_2 l_3 \left(\frac{m_4}{2} + m_c \right) [\sin \theta_{4e} \sin \theta_{3e} \cos \theta_{4a}] + I_{R3a}$$

$$I_{78} = l_4^2 \left(\frac{m_4}{4} + m_c \right) [\sin^2 \theta_{4e}] +$$

$$+ l_3 l_4 \left(\frac{m_4}{2} + m_c \right) [\sin \theta_{4e} \sin \theta_{3e} \cos \theta_{4a}]$$

$$I_{79} = I_{7,10} = I_{7,11} = I_{7,12} = 0$$

$$I_{81} = I_{18}$$

$$I_{82} = I_{28}$$

$$I_{83} = I_{38}$$

$$I_{84} = I_{48} = 0$$

$$I_{85} = I_{58}$$

$$I_{86} = I_{68}$$

$$I_{87} = I_{78}$$

$$I_{88} = l_4^2 \left(\frac{m_4}{4} + m_c \right) [\sin^2 \theta_{4e}] + I_{R4a}$$

$$I_{89} = I_{8,10} = I_{8,11} = I_{8,12} = 0$$

$$I_{91} = I_{92} = I_{93} = I_{94} = I_{95} = I_{96} = I_{97} = I_{98} = I_{99} = I_{9,10} = I_{9,11} = 0$$

$$I_{9,12} = I_{T1}$$

$$I_{10,1} = I_{10,2} = I_{10,3} = I_{10,4} = I_{10,5} = I_{10,6} = I_{10,7} = I_{10,8} = I_{10,9} = 0$$

$$I_{10,10} = I_{T2}$$

$$I_{10,11} = I_{10,12} = 0$$

$$I_{11,1} = I_{11,2} = I_{11,3} = I_{11,4} = I_{11,5} = I_{11,6} = I_{11,7} = I_{11,8} = I_{11,9} = I_{11,10} = 0$$

$$I_{11,11} = I_{T3}$$

$$I_{11,12} = 0$$

$$I_{12,1} = I_{12,2} = I_{12,3} = I_{12,4} = I_{12,5} = I_{12,6} = I_{12,7} = I_{12,8} = I_{12,9} = I_{12,10} = I_{12,11} = 0 \quad .$$

$$I_{12,12} = I_{T4} + I_{Tc}$$

Os elementos da matriz das constantes de atrito C são definidos por:

$$C_{11} = C_{1e} + C_{2e};$$

$$C_{12} = -C_{2e};$$

$$C_{13} = C_{14} = C_{15} = C_{16} = C_{17} = C_{18} = C_{19} = C_{1,10} = C_{1,11} = C_{1,12} = 0;$$

$$C_{21} = C_{12};$$

$$C_{22} = C_{2e} + C_{3e};$$

$$C_{23} = -C_{3e};$$

$$C_{24} = C_{25} = C_{26} = C_{27} = C_{28} = C_{29} = C_{2,10} = C_{2,11} = C_{2,12} = 0;$$

$$C_{31} = C_{13};$$

$$C_{32} = C_{23};$$

$$C_{33} = C_{3e} + C_{4e};$$

$$C_{34} = -C_{4e};$$

$$C_{35} = C_{36} = C_{37} = C_{38} = C_{39} = C_{3,10} = C_{3,11} = C_{3,12} = 0;$$

$$C_{41} = C_{42} = 0;$$

$$C_{43} = C_{34};$$

$$C_{44} = -C_{4e};$$

$$C_{45} = C_{46} = C_{47} = C_{48} = C_{49} = C_{4,10} = C_{4,11} = C_{4,12} = 0;$$

$$C_{51} = C_{52} = C_{53} = C_{54} = 0;$$

$$C_{55} = C_{1a} + C_{2a};$$

$$C_{56} = -C_{2a};$$

$$C_{57} = C_{58} = C_{59} = C_{5,10} = C_{5,11} = C_{5,12} = 0;$$

$$C_{61} = C_{62} = C_{63} = C_{64} = 0;$$

$$C_{65} = C_{56};$$

$$C_{66} = C_{2a} + C_{3a};$$

$$C_{67} = -C_{3a};$$

$$C_{68} = C_{69} = C_{6,10} = C_{6,11} = C_{6,12} = 0;$$

$$C_{71} = C_{72} = C_{73} = C_{74} = C_{75} = 0;$$

$$C_{76} = C_{67};$$

$$C_{77} = C_{3a} + C_{4a};$$

$$C_{78} = -C_{4a};$$

$$C_{79} = C_{7,10} = C_{7,11} = C_{7,12} = 0;$$

$$C_{81} = C_{82} = C_{83} = C_{84} = C_{85} = C_{86} = 0;$$

$$C_{87} = C_{78};$$

$$C_{88} = C_{4a};$$

$$C_{89} = C_{8,10} = C_{8,11} = C_{8,12} = 0;$$

$$C_{91} = C_{92} = C_{93} = C_{94} = C_{95} = C_{96} = C_{97} = C_{98} = 0;$$

$$C_{99} = C_{1T} + C_{2T};$$

$$C_{9,10} = -C_{2T};$$

$$C_{9,11} = C_{9,12} = 0;$$

$$C_{10,1} = C_{10,2} = C_{10,3} = C_{10,4} = C_{10,5} = C_{10,6} = C_{10,7} = C_{10,8} = 0;$$

$$C_{10,9} = C_{9,10};$$

$$C_{10,10} = C_{2T} + C_{3T};$$

$$C_{10,11} = -C_{3T};$$

$$C_{10,12} = 0;$$

$$C_{11,1} = C_{11,2} = C_{11,3} = C_{11,4} = C_{11,5} = C_{11,6} = C_{11,7} = C_{11,8} = C_{11,9} = 0;$$

$$C_{11,10} = C_{10,11};$$

$$C_{11,11} = C_{3T} + C_{4T};$$

$$C_{11,12} = -C_{4T};$$

$$C_{12,1} = C_{12,2} = C_{12,3} = C_{12,4} = C_{12,5} = C_{12,6} = C_{12,7} = C_{12,8} = C_{12,9} = C_{12,10} = 0;$$

$$C_{12,11} = C_{11,12};$$

$$C_{12,12} = C_{4T}$$

Os elementos da matriz das constantes de elásticas K são dados por:

$$K_{11} = K_{1e} + K_{2e};$$

$$K_{12} = -K_{2e};$$

$$K_{13} = K_{14} = K_{15} = K_{16} = K_{17} = K_{18} = K_{19} = K_{1,10} = K_{1,11} = K_{1,12} = 0;$$

$$K_{21} = K_{12};$$

$$K_{22} = K_{2e} + K_{3e};$$

$$K_{23} = -K_{3e};$$

$$K_{24} = K_{25} = K_{26} = K_{27} = K_{28} = K_{29} = K_{2,10} = K_{2,11} = K_{2,12} = 0;$$

$$K_{31} = 0;$$

$$K_{32} = K_{23};$$

$$K_{33} = K_{3e} + K_{4e};$$

$$K_{34} = -K_{4e};$$

$$K_{35} = K_{36} = K_{37} = K_{38} = K_{39} = K_{3,10} = K_{3,11} = K_{3,12} = 0;$$

$$K_{41} = K_{42} = 0;$$

$$K_{43} = K_{34};$$

$$K_{44} = K_{4e};$$

$$K_{45} = K_{46} = K_{47} = K_{48} = K_{49} = K_{4,10} = K_{4,11} = K_{4,12} = 0;$$

$$K_{51} = K_{52} = K_{53} = K_{54} = 0;$$

$$K_{55} = K_{1a} + K_{2a};$$

$$K_{56} = -K_{2a};$$

$$K_{57} = K_{58} = K_{59} = K_{5,10} = K_{5,11} = K_{5,12} = 0;$$

$$K_{61} = K_{62} = K_{63} = K_{64} = 0;$$

$$K_{65} = K_{56};$$

$$K_{66} = K_{2a} + K_{3a};$$

$$K_{67} = K_{3a};$$

$$K_{68} = K_{69} = K_{6,10} = K_{6,11} = K_{6,12} = 0;$$

$$K_{71} = K_{72} = K_{73} = K_{74} = K_{75} = 0;$$

$$K_{76} = -K_{3a};$$

$$K_{77} = K_{3a} + K_{4a};$$

$$K_{78} = -K_{4a};$$

$$K_{79} = K_{7,10} = K_{7,11} = K_{7,12} = 0;$$

$$K_{81} = K_{82} = K_{83} = K_{84} = K_{85} = K_{86} = 0;$$

$$K_{87} = K_{78};$$

$$K_{88} = K_{4a};$$

$$K_{89} = K_{8,10} = K_{8,11} = K_{8,12} = 0;$$

$$K_{91} = K_{92} = K_{93} = K_{94} = K_{95} = K_{96} = K_{97} = K_{98} = 0;$$

$$K_{99} = K_{1T} + K_{2T};$$

$$K_{9,10} = -K_{2T};$$

$$K_{9,11} = K_{9,12} = 0;$$

$$K_{10,1} = K_{10,2} = K_{10,3} = K_{10,4} = K_{10,5} = K_{10,6} = K_{10,7} = K_{10,8} = 0;$$

$$K_{10,9} = K_{9,10};$$

$$K_{10,10} = K_{2T} + K_{3T};$$

$$K_{10,11} = -K_{3T};$$

$$K_{10,12} = 0;$$

$$K_{11,1} = K_{11,2} = K_{11,3} = K_{11,4} = K_{11,5} = K_{11,6} = K_{11,7} = K_{11,8} = K_{11,9} = 0;$$

$$K_{11,10} = K_{10,11};$$

$$K_{11,11} = K_{3T} + K_{4T};$$

$$K_{11,12} = -K_{4T};$$

$$K_{12,1} = K_{112,2} = K_{12,3} = K_{12,4} = K_{12,5} = K_{12,6} = K_{12,7} = K_{12,8} = K_{12,9} = K_{12,10} = 0;$$

$$K_{12,11} = K_{11,12};$$

$$K_{12,12} = K_{4T}$$

Os elementos do vetor coriolis-centrífugo \vec{F} são definidos conforme segue:

$$\begin{aligned} f_1 = & -l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_4 + m_c \right) \{ \text{sen } \theta_{1e} \cos \theta_{1e} \dot{\theta}_{1a}^2 \} + \\ & -2l_1 l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) \{ \cos \theta_{1e} [\text{sen } \theta_{2e} \cos \theta_{2a} (\dot{\theta}_{1a} \dot{\theta}_{2a} + \\ & + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{2e}^2}{2} \right)] + \cos \theta_{2e} \text{sen } \theta_{2a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e}] - \frac{1}{2} \cos \theta_{2e} \text{sen } \theta_{1e} \dot{\theta}_{2e}^2 \} + \end{aligned}$$

$$\begin{aligned}
 & -2l_1l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) \left\{ \cos \theta_{1e} \left[\text{sen} \theta_{3e} \cos(\theta_{2a} + \theta_{3a}) \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} \right) + \right. \right. \\
 & \left. \left. + (\dot{\theta}_{1a}\dot{\theta}_{2a} + \dot{\theta}_{1a}\dot{\theta}_{3a} + \dot{\theta}_{2a}\dot{\theta}_{3a}) \right] + \cos \theta_{3e} \text{sen}(\theta_{2a} + \theta_{3a})(\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a})\dot{\theta}_{3e} \right\} + \\
 & -\frac{1}{2} \cos \theta_{3e} \text{sen} \theta_{1e} \dot{\theta}_{3e} \left. \right\} - 2l_1l_4 \left(\frac{m_4}{4} + m_c \right) + \\
 f_2 = & -2l_2^2 \left(\frac{m_4}{4} + m_3 + m_4 + m_c \right) \text{sen} \theta_{2e} \cos \theta_{2e} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2}{2} + \dot{\theta}_{1a}\dot{\theta}_{2a} \right) + \\
 & -2l_1l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) \left\{ \cos \theta_{2e} \left[\text{sen} \theta_{1e} \cos \theta_{2a} + \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{1e}^2}{2} \right) + \right. \right. \\
 & \left. \left. - \cos \theta_{1e} \text{sen} \theta_{2a} \dot{\theta}_{1a}\dot{\theta}_{1e} \right] - \frac{1}{2} \cos \theta_{1e} \text{sen} \theta_{2e} \dot{\theta}_{1e}^2 \right. \\
 & - 2l_2l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) \left\{ \cos \theta_{2e} \left[\text{sen} \theta_{3e} \cos \theta_{3a} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} \right) + \right. \right. \\
 & \left. \left. + (\dot{\theta}_{1a}\dot{\theta}_{2a} + \dot{\theta}_{1a}\dot{\theta}_{3a} + \dot{\theta}_{2a}\dot{\theta}_{3a}) \right] + \cos \theta_{3e} \text{sen} \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a})\dot{\theta}_{3e} \right\} + \\
 & -\frac{1}{2} \text{sen} \theta_{2e} \cos \theta_{3e} \dot{\theta}_{3e}^2 \left. \right\} + \\
 & - 2l_2l_4 \left(\frac{m_4}{2} + m_c \right) \left\{ \cos \theta_{2e} \left[\text{sen} \theta_{4e} \cos(\theta_{3a} + \theta_{4a}) \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{4a}^2 + \dot{\theta}_{4e}^2}{2} + \right. \right. \right. \\
 & \left. \left. + (\dot{\theta}_{1a}\dot{\theta}_{2a} + \dot{\theta}_{1a}\dot{\theta}_{3a} + \dot{\theta}_{1a}\dot{\theta}_{4a} + \dot{\theta}_{2a}\dot{\theta}_{3a} + \dot{\theta}_{2a}\dot{\theta}_{4a} + \dot{\theta}_{3a}\dot{\theta}_{4a}) + \right. \right. \\
 & \left. \left. + \cos \theta_{4e} \text{sen}(\theta_{3a} + \theta_{4a})(\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a} + \dot{\theta}_{4a})\dot{\theta}_{4e} \right] - \frac{1}{2} \text{sen} \theta_{2e} \cos \theta_{4e} \dot{\theta}_{4e} \right\} \\
 f_3 = & -2l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) \left\{ \text{sen} \theta_{3e} \cos \theta_{3e} \left[\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2}{2} + \right. \right. \\
 & \left. \left. + (\dot{\theta}_{1a}\dot{\theta}_{2a} + \dot{\theta}_{1a}\dot{\theta}_{3a} + \dot{\theta}_{2a}\dot{\theta}_{3a}) \right] \right\} + \\
 & - 2l_1l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) \left\{ \cos \theta_{1e} \left[\text{sen} \theta_{3e} \cos(\theta_{2a} + \theta_{3a}) \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} \right) + \right. \right. \\
 & \left. \left. + \cos \theta_{3e} \text{sen}(\theta_{2a} + \theta_{3a})(\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a})\dot{\theta}_{3e} \right] - \frac{1}{2} \cos \theta_{3e} \text{sen} \theta_{1e} \dot{\theta}_{3e}^2 \right\} + \\
 & - 2l_2l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) \left\{ \cos \theta_{2e} \left[\text{sen} \theta_{3e} \cos \theta_{3a} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} + \right. \right. \right. \\
 & \left. \left. + \dot{\theta}_{1a}\dot{\theta}_{2a} + \dot{\theta}_{1a}\dot{\theta}_{3a} + \dot{\theta}_{2a}\dot{\theta}_{3a} \right] - \cos \theta_{3e} \text{sen} \theta_{3a} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a})\dot{\theta}_{3e} \right\} \\
 & - \frac{1}{2} \cos \theta_{2e} \text{sen} \theta_{3e} \dot{\theta}_{2e}^2 \left. \right\}
 \end{aligned}$$

$$\begin{aligned}
 & -2l_3l_4 \left(\frac{m_4}{2} + m_c \right) \left\{ \cos \theta_{1e} \left[\text{sen} \theta_{3e} \cos \theta_{4a} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} + \right. \right. \right. \\
 & \left. \left. \left. + (\dot{\theta}_{1a} \dot{\theta}_{2a} + \dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{3a}) \right) - \cos \theta_{3e} \text{sen} \theta_{4a} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{4e} \right] + \right. \\
 & \left. - \frac{1}{2} \cos \theta_{3e} \text{sen} \theta_{4e} \dot{\theta}_{3e}^2 \right\} \\
 f_4 = & -2l_4^2 \left(\frac{m_4}{4} + m_c \right) \left\{ \text{sen} \theta_{4e} \cos \theta_{4e} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{4a}^2}{2} + \right. \right. \\
 & \left. \left. + (\dot{\theta}_{1a} \dot{\theta}_{2a} + \dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{1a} \dot{\theta}_{4a} + \dot{\theta}_{2a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{4a} + \dot{\theta}_{3a} \dot{\theta}_{4a}) \right) \right\} + \\
 & -2l_1l_4 \left(\frac{m_4}{4} + m_c \right) \left\{ \cos \theta_{4e} \left[\text{sen} \theta_{1e} \cos (\theta_{2a} + \theta_{3a} + \theta_{4a}) (\dot{\theta}_{1a}^2 + \dot{\theta}_{1e}^2) + \right. \right. \\
 & \left. \left. - \cos \theta_{1e} \text{sen} (\theta_{2a} + \theta_{3a} + \theta_{4a}) \dot{\theta}_{1a} \dot{\theta}_{1e} \right] - \frac{1}{2} \cos \theta_{1e} \text{sen} \theta_{1e} \dot{\theta}_{1e}^2 \right\} + \\
 & -2l_2l_4 \left(\frac{m_4}{2} + m_c \right) \left\{ \cos \theta_{4e} \left[\text{sen} \theta_{2e} \cos (\theta_{3a} + \theta_{4a}) \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3e}^2}{2} + \dot{\theta}_{1a} \dot{\theta}_{2a} \right) + \right. \right. \\
 & \left. \left. - \cos \theta_{2e} \text{sen} (\theta_{3a} + \theta_{4a}) (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} \right] - \frac{1}{2} \cos \theta_{2e} \text{sen} \theta_{4e} \dot{\theta}_{3e}^2 \right\} + \\
 & -2l_3l_4 \left(\frac{m_4}{2} + m_c \right) \left\{ \cos \theta_{4e} \left[\text{sen} \theta_{3e} \cos \theta_{4a} \left(\frac{\dot{\theta}_{1a}^2 + \dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 + \dot{\theta}_{3e}^2}{2} + \right. \right. \right. \\
 & \left. \left. \left. + (\dot{\theta}_{1a} \dot{\theta}_{2a} + \dot{\theta}_{1a} \dot{\theta}_{3a} + \dot{\theta}_{2a} \dot{\theta}_{3a}) \right) + \right. \right. \\
 & \left. \left. - \cos \theta_{3e} \text{sen} \theta_{4a} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} \right] - \frac{1}{2} \cos \theta_{3e} \text{sen} \theta_{4e} \dot{\theta}_{3e}^2 \right\} \\
 f_5 = & 2l_1^2 \left(\frac{m_1}{4} + m_2 + m_3 + m_4 + m_c \right) \left\{ \text{sen} \theta_{1e} \cos \theta_e \dot{\theta}_{1a} \dot{\theta}_{2e} \right\} + \\
 & + 2l_2^2 \left(\frac{m_2}{4} + m_3 + m_4 + m_c \right) \left\{ \text{sen} \theta_{2e} \cos \theta_{2e} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} \right\} + \\
 & + 2l_3^2 \left(\frac{m_3}{4} + m_4 + m_c \right) \left\{ \text{sen} \theta_{3e} \cos \theta_{3e} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} \right\} + \\
 & + 2l_4^2 \left(\frac{m_4}{4} + m_c \right) \left\{ \text{sen} \theta_{4e} \cos \theta_{4e} (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a} + \dot{\theta}_{4a}) \dot{\theta}_{3e} \right\} + \\
 & + 2l_1l_2 \left(\frac{m_2}{2} + m_3 + m_4 + m_c \right) \left\{ -s\theta_{1e}s\theta_{2e}s\theta_{2a} \left(\frac{\dot{\theta}_{2a}^2 - \dot{\theta}_{1e}^2 + \dot{\theta}_{2e}^2}{2} + \dot{\theta}_{1a} \dot{\theta}_{2a} \right) + \right. \\
 & \left. + \text{sen} \theta_{1e} \cos \theta_{2e} \cos \theta_{2a} (\dot{\theta}_{1a} + \dot{\theta}_{2a}) \dot{\theta}_{2e} + \cos \theta_{1e} \text{sen} \theta_{2e} \cos \theta_{2a} \dot{\theta}_{1a} \dot{\theta}_{1e} \right\} + \\
 & + 2l_1l_3 \left(\frac{m_3}{2} + m_4 + m_c \right) \left\{ -s\theta_{1e}s\theta_{3e}s(\theta_{2a} + \theta_{3a}) \left(\frac{\dot{\theta}_{2a}^2 + \dot{\theta}_{3a}^2 - \dot{\theta}_{1e}^2 + \dot{\theta}_{3e}^2}{2} + \right. \right. \\
 & \left. \left. + \dot{\theta}_{1a} \dot{\theta}_{2a} + \dot{\theta}_{2a} \dot{\theta}_{3a} + \dot{\theta}_{1a} \dot{\theta}_{3a} \right) + \text{sen} \theta_{1e} \cos \theta_{3e} \cos (\theta_{2a} + \theta_{3a}) (\dot{\theta}_{1a} + \dot{\theta}_{2a} + \dot{\theta}_{3a}) \dot{\theta}_{3e} + \right. \\
 & \left. + \cos \theta_{1e} \text{sen} \theta_{3e} \cos (\theta_{2a} + \theta_{3a}) \dot{\theta}_{1a} \dot{\theta}_{1e} \right\} +
 \end{aligned}$$

$$\begin{aligned}
 & +2l_1l_4\left(\frac{m_4}{2}+m_c\right)\{-s\theta_{1e}s\theta_{4e}s(\theta_{2a}+\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{2a}^2+\dot{\theta}_{3a}^2+\dot{\theta}_{4a}^2-\dot{\theta}_{1e}^2+\dot{\theta}_{4e}^2}{2}+\right. \\
 & +\dot{\theta}_{1a}\dot{\theta}_{2a}+\dot{\theta}_{1a}\dot{\theta}_{3a}+\dot{\theta}_{1a}\dot{\theta}_{4a}+\dot{\theta}_{2a}\dot{\theta}_{3a}+\dot{\theta}_{2a}\dot{\theta}_{4a}+\dot{\theta}_{3a}\dot{\theta}_{4a})+ \\
 & +\operatorname{sen}\theta_{1e}\cos\theta_{4e}\cos(\theta_{2a}+\theta_{3a}+\theta_{4a})\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a}\right)\dot{\theta}_{4e}+ \\
 & \left.+\cos\theta_{1e}\operatorname{sen}\theta_{4e}\cos(\theta_{2a}+\theta_{3a}+\theta_{4a})\dot{\theta}_{1a}\dot{\theta}_{1e}\right\}+ \\
 & +2l_2l_3\left(\frac{m_3}{2}+m_4+m_c\right)\{-s\theta_{2e}s\theta_{3e}s\theta_{3a}\left(\frac{\dot{\theta}_{3a}^2-\dot{\theta}_{2e}^2+\dot{\theta}_{3e}^2}{2}\left(\dot{\theta}_{1a}\dot{\theta}_{2a}\right)\dot{\theta}_{3a}\right)+ \\
 & +\cos\theta_{2e}\operatorname{sen}\theta_{3e}\cos\theta_{3a}\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}\right)\dot{\theta}_{2e}\}+ \\
 & +\operatorname{sen}\theta_{2e}\cos\theta_{3e}\cos\theta_{3a}\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}\right)\dot{\theta}_{3e}\}+ \\
 & +2l_2l_4\left(\frac{m_4}{2}+m_c\right)\{-s\theta_{2e}s\theta_{4e}s(\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{3a}^2+\dot{\theta}_{4a}^2-\dot{\theta}_{2e}^2+\dot{\theta}_{4e}^2}{2}+\right. \\
 & +\dot{\theta}_{1a}\dot{\theta}_{3a}+\dot{\theta}_{1a}\dot{\theta}_{4a}+\dot{\theta}_{2a}\dot{\theta}_{3a}+\dot{\theta}_{2a}\dot{\theta}_{4a}+\dot{\theta}_{3a}\dot{\theta}_{4a})+ \\
 & +\cos\theta_{2e}\operatorname{sen}\theta_{4e}\cos(\theta_{3a}+\theta_{4a})\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}\right)\dot{\theta}_{2e}+ \\
 & \left.+\operatorname{sen}\theta_{2e}\cos\theta_{4e}\cos(\theta_{3a}+\theta_{4a})\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a}\right)\dot{\theta}_{4e}\right\}+ \\
 & +2l_3l_4\left(\frac{m_4}{2}+m_c\right)\{-s\theta_{3e}s\theta_{4e}s\theta_{4a}\left(\frac{\dot{\theta}_{4a}^2-\dot{\theta}_{3e}^2+\dot{\theta}_{4e}^2}{2}+\right. \\
 & +\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}\right)\dot{\theta}_{4a})+\operatorname{sen}\theta_{3e}\cos\theta_{4e}\cos\theta_{4a}\theta_{4a}\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a}\right)\dot{\theta}_{4e}\}+ \\
 & \left.+\cos\theta_{3e}\operatorname{sen}\theta_{4e}\cos(\theta_{2a}+\theta_{3a}+\theta_{4a})\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}\right)\dot{\theta}_{3e}\right\} \\
 f_6 = & 2l_2^2\left(\frac{m_2}{4}+m_3+m_4+m_c\right)\{\operatorname{sen}\theta_{2e}\cos\theta_{2e}\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}\right)\dot{\theta}_{2e}\}+ \\
 & +2l_3^2\left(\frac{m_3}{4}+m_4+m_c\right)\{\operatorname{sen}\theta_{3e}\cos\theta_{3e}\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}\right)\dot{\theta}_{3e}\}+ \\
 & +2l_4^2\left(\frac{m_4}{4}+m_c\right)\{\operatorname{sen}\theta_{4e}\cos\theta_{4e}\left(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a}\right)\dot{\theta}_{4e}\}+ \\
 & +2l_1l_2\left(\frac{m_2}{2}+m_3+m_4+m_c\right)\{\cos\theta_{1e}s\theta_{2e}\cos\theta_{2a}\dot{\theta}_{1a}\dot{\theta}_{1e}+ \\
 & +\operatorname{sen}\theta_{1e}\operatorname{sen}\theta_{2e}\operatorname{sen}\theta_{2a}\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{1e}^2}{2}\right)\}+ \\
 & +2l_1l_3\left(\frac{m_3}{2}+m_4+m_c\right)\{\cos\theta_{1e}s\theta_{3e}\cos(\theta_{2a}+\theta_{3a})\dot{\theta}_{1a}\dot{\theta}_{1e}+ \\
 & +\operatorname{sen}\theta_{1e}\operatorname{sen}\theta_{3e}\operatorname{sen}(\theta_{2a}+\theta_{3a})\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{1e}^2}{2}\right)\}+
 \end{aligned}$$

$$\begin{aligned}
 &+2l_1l_4\left(\frac{m_4}{4}+m_c\right)\left\{\cos\theta_{1e}\theta_{4e}\cos(\theta_{2a}+\theta_{3a}+\theta_{4a})\dot{\theta}_{1a}\dot{\theta}_{1e}+\right. \\
 &\left.+\operatorname{sen}\theta_{1e}\operatorname{sen}\theta_{4e}\operatorname{sen}(\theta_{2a}+\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{1e}^2}{2}\right)\right\}+ \\
 &+2l_2l_3\left(\frac{m_3}{2}+m_4+m_c\right)\left\{-s\theta_{2e}s\theta_{3e}s\theta_{3a}\left[\left(\frac{\dot{\theta}_{3a}^2+\dot{\theta}_{3e}^2-\dot{\theta}_{2e}^2}{2}\right)+\dot{\theta}_{1a}\dot{\theta}_{2a}\right]+ \right. \\
 &+\cos\theta_{2e}s\theta_{3e}\cos\theta_{3a}(\dot{\theta}_{1a}+\dot{\theta}_{2a})\dot{\theta}_{2e}+ \\
 &+\operatorname{sen}\theta_{2e}\cos\theta_{3e}\cos\theta_{3a}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a})\dot{\theta}_{3e}\left.\right\}+ \\
 &+2l_2l_4\left(\frac{m_4}{2}+m_c\right)\left\{-s\theta_{2e}s\theta_{4e}s(\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{3a}^2+\dot{\theta}_{4a}^2-\dot{\theta}_{2e}^2+\dot{\theta}_{4e}^2}{2}+\right.\right. \\
 &+\dot{\theta}_{1a}\dot{\theta}_{2a}+\dot{\theta}_{1a}\dot{\theta}_{3a}+\dot{\theta}_{1a}\dot{\theta}_{4a}+\dot{\theta}_{1a}\dot{\theta}_{4a}+\dot{\theta}_{2a}\theta_{4a}+\dot{\theta}_{3a}\dot{\theta}_{4a})+ \\
 &+\cos\theta_{2e}\operatorname{sen}\theta_{4e}\cos(\theta_{3a}+\theta_{4a})(\dot{\theta}_{1a}+\dot{\theta}_{2a})\dot{\theta}_{2e}+ \\
 &+\operatorname{sen}\theta_{2e}\cos\theta_{4e}\cos(\theta_{3a}+\theta_{4a})(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a})\dot{\theta}_{4e}\left.\right\}+ \\
 &+2l_3l_4\left(\frac{m_4}{2}+m_c\right)\left\{-\operatorname{sen}\theta_{3e}\operatorname{sen}\theta_{4e}\operatorname{sen}\theta_{4a}\left(\frac{\dot{\theta}_{4a}^2-\dot{\theta}_{3e}^2+\dot{\theta}_{4e}^2}{2}+\right.\right. \\
 &+\dot{\theta}_{1a}\dot{\theta}_{2a}+\dot{\theta}_{2a}\dot{\theta}_{3a}+\dot{\theta}_{1a}\dot{\theta}_{3a})+(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a})\dot{\theta}_{4a}+ \\
 &+\operatorname{sen}\theta_{3e}\cos\theta_{4e}\cos\theta_{4a}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a})\dot{\theta}_{4e}\left.\right\} \\
 &+\cos\theta_{3e}\operatorname{sen}\theta_{4e}\cos\theta_{4a}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a})\dot{\theta}_{3e}\left.\right\} \\
 f_7 = &-2l_3^2\left(\frac{m_3}{4}+m_4+m_c\right)\left\{\operatorname{sen}\theta_{3e}\cos\theta_{3e}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a})\dot{\theta}_{3e}\right\}+ \\
 &+2l_4^2\left(\frac{m_4}{4}+m_c\right)\left\{\operatorname{sen}\theta_{4e}\cos\theta_{4e}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a})\dot{\theta}_{4e}\right\}+ \\
 &+2l_1l_3\left(\frac{m_3}{2}+m_4+m_c\right)\left\{\operatorname{sen}\theta_{1e}s\theta_{3e}\operatorname{sen}(\theta_{2a}+\theta_{3a})\left(\frac{\dot{\theta}_{2a}^2+\dot{\theta}_{1e}^2}{2}\right)+\right. \\
 &+\cos\theta_{1e}\operatorname{sen}\theta_{3e}\cos(\theta_{2a}+\theta_{3a})\dot{\theta}_{1a}\dot{\theta}_{1e}\left.\right\}+ \\
 &+2l_1l_4\left(\frac{m_4}{2}+m_c\right)\left\{s\theta_{1e}s\theta_{4e}\operatorname{sen}(\theta_{2a}+\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{1e}^2}{2}\right)+\right. \\
 &+\cos\theta_{1e}s\theta_{4e}\cos(\theta_{2a}+\theta_{3a}+\theta_{4a})\dot{\theta}_{1a}\dot{\theta}_{1e}\left.\right\}+ \\
 &+2l_2l_3\left(\frac{m_3}{2}+m_4+m_c\right)\left\{s\theta_{2e}s\theta_{3e}s\theta_{3a}\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{2e}^2+\dot{\theta}_{2e}^2}{2}+\dot{\theta}_{1a}\dot{\theta}_{2a}\right)+\right. \\
 &+\cos\theta_{2e}\operatorname{sen}\theta_{3e}\cos\theta_{3a}(\dot{\theta}_{1a}+\dot{\theta}_{2a})\dot{\theta}_{2e}\left.\right\}+
 \end{aligned}$$

$$\begin{aligned}
 &+2l_2l_4\left(\frac{m_4}{2}+m_c\right)\{s\theta_{2e}s\theta_{4e}s(\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{2a}^2+\dot{\theta}_{2e}^2}{2}+\dot{\theta}_{1a}\dot{\theta}_{2a}\right)+ \\
 &+\cos\theta_{2e}\sin\theta_{4e}\cos(\theta_{3a}+\theta_{4a})(\dot{\theta}_{1a}+\dot{\theta}_{2a})\dot{\theta}_{2e}\}+ \\
 &+2l_3l_4\left(\frac{m_4}{2}+m_c\right)\{-\sin\theta_{3e}\sin\theta_{4e}\sin\theta_{4a}\left(\frac{\dot{\theta}_{4a}^2-\dot{\theta}_{3e}^2+\dot{\theta}_{4e}^2}{2}+(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a})\dot{\theta}_{4a}\right)+ \\
 &+\cos\theta_{3e}\sin\theta_{4e}\cos\theta_{4a}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a})\dot{\theta}_{3e}\}+ \\
 &+\sin\theta_{3e}\cos\theta_{4a}\cos\theta_{4a}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a})\dot{\theta}_{4e}\}
 \end{aligned}$$

$$\begin{aligned}
 f_8 = &-2l_4^2\left(\frac{m_4}{2}+m_c\right)\{\cos\theta_{4e}\sin\theta_{4e}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a}+\dot{\theta}_{4a})\dot{\theta}_{4e}\}+ \\
 &-2l_1l_4\left(\frac{m_4}{2}+m_c\right)\{-s\theta_{1e}s\theta_{4e}\sin(\theta_{2a}+\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{1e}^2}{2}\right)+ \\
 &-\cos\theta_{1e}s\theta_{4e}\cos(\theta_{2a}+\theta_{3a}+\theta_{4a})\dot{\theta}_{1a}\dot{\theta}_{1e}\}+ \\
 &-2l_2l_4\left(\frac{m_4}{2}+m_c\right)\{-s\theta_{2e}s\theta_{4e}s(\theta_{3a}+\theta_{4a})\left(\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{2a}^2+\dot{\theta}_{2e}^2}{2}+\dot{\theta}_{1a}\dot{\theta}_{2a}\right)+ \\
 &-\cos\theta_{2e}\sin\theta_{4e}\cos(\theta_{3a}+\theta_{4a})(\dot{\theta}_{1a}+\dot{\theta}_{2a})\dot{\theta}_{2e}\}+ \\
 &-2l_3l_4\left(\frac{m_4}{2}+m_c\right)\{-\sin\theta_{3e}\sin\theta_{4e}\sin\theta_{4a}\left((\dot{\theta}_{1a}\dot{\theta}_{2a}+\dot{\theta}_{1a}\dot{\theta}_{3a}+\dot{\theta}_{2a}\dot{\theta}_{3a})+ \right. \\
 &\left. +\frac{\dot{\theta}_{1a}^2+\dot{\theta}_{2a}^2+\dot{\theta}_{3a}^2+\dot{\theta}_{3e}^2}{2}\right)-\cos\theta_{3e}\sin\theta_{4e}\cos\theta_{4a}(\dot{\theta}_{1a}+\dot{\theta}_{2a}+\dot{\theta}_{3a})\dot{\theta}_{3e}\}
 \end{aligned}$$

$$f_9 = f_{10} = f_{11} = f_{12} = 0.$$

Os elementos do vetor gravitacional \vec{G} são dados por:

$$g_1 = l_1\left(\frac{m_1}{2}+m_2+m_3+m_4\right)g\sin\theta_{1e};$$

$$g_2 = l_2\left(\frac{m_2}{2}+m_3+m_4\right)g\sin\theta_{2e};$$

$$g_3 = l_3\left(\frac{m_3}{2}+m_4\right)g\sin\theta_{3e};$$

$$g_4 = l_4\left(\frac{m_4}{2}\right)g\sin\theta_{4e};$$

$$g_5 = g_6 = g_7 = g_8 = g_9 = g_{10} = g_{11} = g_{12} = 0$$

Os elementos do vetor de torques externos \vec{T} são:

$$T_{m_1} = T_{\theta_{1e}};$$

$$T_{m_2} = T_{\theta_{2e}};$$

$$T_{m_3} = T_{\theta_{3e}};$$

$$T_{m_4} = T_{\theta_{4e}};$$

$$T_{m_5} = T_{\theta_{1a}};$$

$$T_{m_6} = T_{\theta_{2a}};$$

$$T_{m_7} = T_{\theta_{3a}};$$

$$T_{m_8} = T_{\theta_{4a}};$$

$$T_{m_9} = T_{\theta_{1T}};$$

$$T_{m_{10}} = T_{\theta_{2T}};$$

$$T_{m_{11}} = T_{\theta_{3T}};$$

$$T_{m_{12}} = T_{\theta_{4T}}$$

APÊNDICE B - ESPAÇO DE TRABALHO TRIDIMENSIONAL

Utiliza-se para abordagem deste apêndice o livro de COHEN e MANSSOUR [32] visto que ele traz todos os conceitos necessários para se falar da visualização 3D utilizada neste trabalho.

A visualização de objetos, em OpenGL (KHRONOS [5]), ocorre em uma cena tridimensional, a qual representa virtualmente um cenário real. Um espaço de trabalho tridimensional possui três eixos ortogonais entre si (X, Y, Z) que passa pela origem $(0,0,0,0,0)$ no Sistema de Referência do Universo (SRU), sistema do OpenGL. A Figura B.1 mostra os sistema adotado pelo OpenGL.

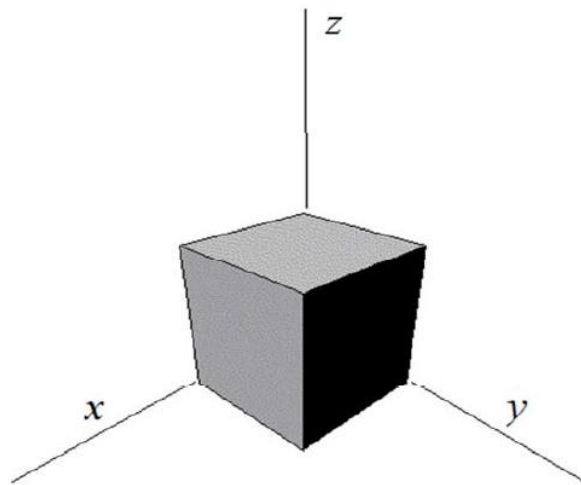


Figura B.1– Sistema SRU em três dimensões utilizando OpenGL (COLARES [36]).

Todos os comandos OpenGL são utilizados com base neste sistema. Para mostrar a cena com seus objetos contidos nela nos monitores é necessária a transformação das coordenadas do sistema SRU para o Sistema de Referência de Tela (SRT). Os dispositivos de saída, como os monitores, processam tais saídas em 2D e a origem é localizada no canto superior esquerdo da tela. O processo de conversão do sistema SRU para SRT exige muitas etapas e para visualizar a cena 3D e projetá-la em 2D é necessário especificar de que forma os recursos, como a câmera sintética e a projeção são definidos no OpenGL.

A câmera sintética refere-se ao processo de visualização 3D, ou seja, define o posicionamento dos objetos pertencentes a uma cena 3D no sistema SRU. Para isto são necessárias operações de transformações geométricas, tais como translação, escala e rotação.

Adicionam-se as constantes de deslocamento a todas as coordenadas das primitivas gráficas que compõem a cena e os objetos realiza-se a translação, alterando assim a posições dos mesmos. Para o processo de escala, multiplica-se o valor da escala por todas as coordenadas das primitivas que compõem a cena e os objetos. Já na rotação utiliza-se o valor do ângulo como referencia para girar a cena e os objetos. Nesta operação, são realizados cálculos com relação ao seno e o cosseno do ângulo de rotação aplicando-se a todas as coordenadas das primitivas gráficas que compõem a cena e os objetos.

Uma câmera sintética possui componentes importantes que facilita a visualização 3D. O observador é um destes componentes, ele define o local que se deseja exibir a cena, ou seja, os objetos podem ser vistos de lugares diferentes. O observador é composto por posição e orientação, isto é, define onde ele está e para onde está olhando. A posição da câmera é dada por um ponto (x,y,z) em relação ao sistema SRU assim como os objetos pertencentes à cena. A orientação da câmera é definida pelo ponto Alvo (x,y,z) e um vetor, chamado de up. A Figura B.2a indica que os objetos são processados da mesma forma que estão dispostos na cena. Já na Figura B.2b, os objetos aparecem inclinados 90° para a esquerda.

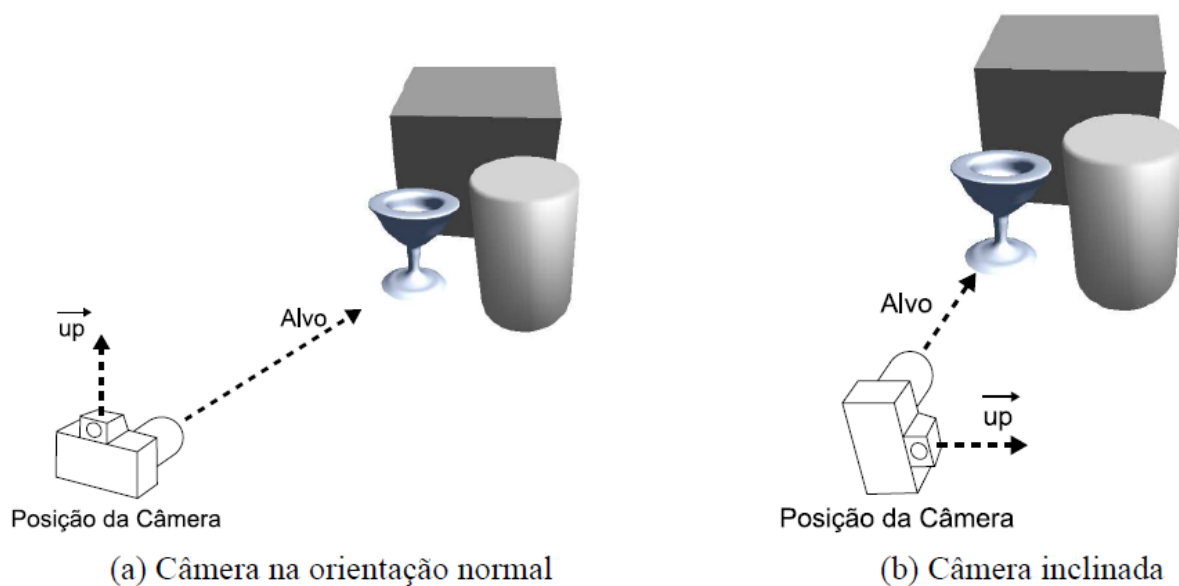


Figura B.2 – Câmera sintética do OpenGL posicionada de duas formas diferentes (COHEN e MANSSOUR [32]).

Outro componente essencial da biblioteca OpenGL é a projeção que é responsável por mapear os objetos 3D para imagens 2D para serem projetados nos dispositivos de saídas, como o monitor, por exemplo.

Os objetos são representados por vértices e a projeção destes objetos ocorre por raios de projeção, chamados de projetantes. Raios de projeção são seguimentos de retas que passam através de cada vértice do objeto e interseccionam um plano de projeção. As projeções são classificadas em duas: Paralela Ortográfica e Perspectiva.

Na primeira projeção, as projetantes são paralelas entre si passando pelos pontos que definem os objetos e interseccionam o plano com um ângulo de 90° , conforme Figura B.3a.

Na segunda projeção, perspectiva, as projetantes derivam de um único ponto que está a uma distância finita do plano de projeção e passam pelos pontos que definem os objetos (Figura B.3b). Esta projeção é muito utilizada por exibir melhor os objetos do mundo virtual.

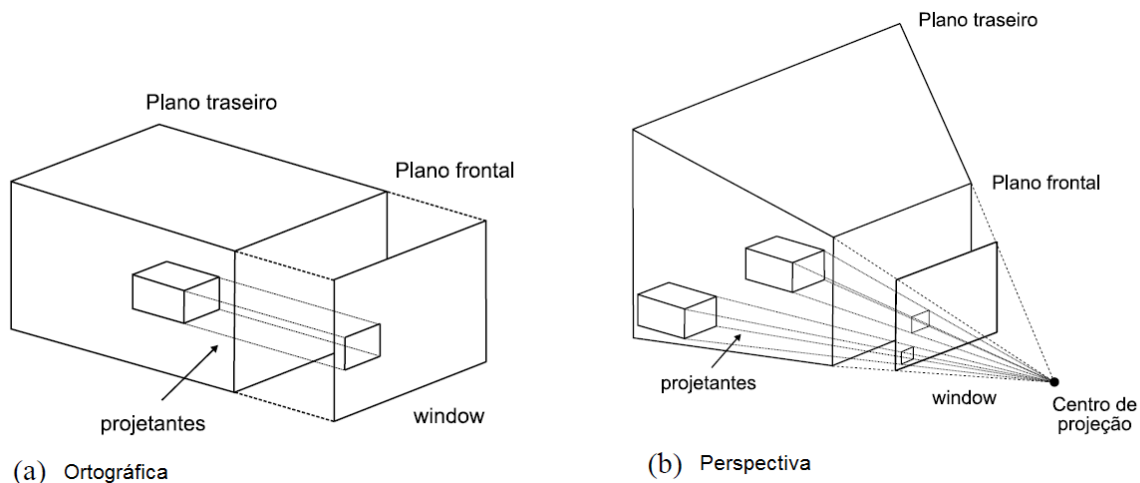


Figura B.3 – Projeção de paralelepípedos (COHEN e MANSSOUR [32]).

Na projeção ortográfica não acontecem alterações nas dimensões do objeto, pois apenas ocorre a omissão de uma das componentes de cada vértice. Já a projeção perspectiva representa melhor a realidade projetada por aumentar o tamanho de objetos localizados mais próximos do ponto central de projeção e diminuir o tamanho dos objetos mais afastados deste ponto central de projeção.