

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROFMAT - MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL



SAMUEL DUQUE GONÇALVES

GRUPOS MATRICIAIS APLICADOS À CINEMÁTICA
DIRETA DE BRAÇO ROBÓTICO

BELO HORIZONTE
2020

SAMUEL DUQUE GONÇALVES

**GRUPOS MATRICIAIS APLICADOS À CINEMÁTICA
DIRETA DE BRAÇO ROBÓTICO**

Dissertação apresentada ao Centro Federal de Educação Tecnológica de Minas Gerais como parte das exigências do Programa de Pós-Graduação Mestrado Profissional em Matemática em Rede Nacional, para obter o título de Mestre.

Orientadora

Marcela Richele Ferreira

Coorientação

Geraldo Magela Couto Oliveira

Banca Examinadora

Carlos Alexandre Silva

Saulo Furletti

Éden Santana Campos Amorim

BELO HORIZONTE
2020

G635g Gonçalves, Samuel Duque
 Grupos matriciais aplicados à cinemática direta de braço robótico /
 Samuel Duque Gonçalves. – 2020.
 127 f.

 Dissertação de mestrado apresentada ao Programa de Mestrado
Profissional em Matemática em Rede Nacional.
 Orientadora: Marcela Richele Ferreira.
 Coorientador: Geraldo Magela Couto Oliveira.
 Dissertação (mestrado) – Centro Federal de Educação Tecnológica de
Minas Gerais.

 1. Matemática aplicada – Teses. 2. Geometria analítica – Teses.
3. Álgebra linear – Teses. 4. Grupos matriciais – Teses. 5. Robótica – Teses.
I. Ferreira, Marcela Richele. II. Oliveira, Geraldo Magela Couto. III. Centro
Federal de Educação Tecnológica de Minas Gerais. IV. Título.

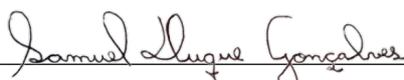
CDD 519

SAMUEL DUQUE GONÇALVES

**GRUPOS MATRICIAIS APLICADOS À CINEMÁTICA
DIRETA DE BRAÇO ROBÓTICO**

Dissertação apresentada ao Centro Federal de Educação Tecnológica de Minas Gerais como parte das exigências do Programa de Pós-Graduação Mestrado Profissional em Matemática em Rede Nacional, para obter o título de Mestre.

APROVADA: 25 de novembro de 2020.



Samuel Duque Gonçalves
(Autor)



Marcela Richele Ferreira
(Orientadora)

BELO HORIZONTE
2020

Às minhas queridas Lara e Maria Alice

Agradecimentos

Agradeço primeiramente a Deus, criador e sustentador de todas as coisas, que me proporcionou tudo que precisava para chegar até a conclusão desse trabalho. A Ele toda a Glória!

À minha amada esposa Lara minha profunda gratidão, pelo companheirismo, incentivo e amor manifestados a cada momento, não somente durante a elaboração da dissertação, mas durante todo o mestrado. Agradeço, também, à nossa filha Maria Alice pela compreensão, pelo carinho e por se preocupar pelas noites mal dormidas de seu pai. O apoio de vocês foi o combustível para eu ir até o fim.

Agradeço muito à minha orientadora Marcela pela paciência, presteza e pelas orientações precisas dadas. Ao meu coorientador Geraldo Magela, minha gratidão pelos comentários relevantes e oportunos. A experiência e o conhecimento de vocês, bem como todo o suporte dado, foram essenciais nessa trajetória.

Aos professores doutores Carlos Alexandre Silva, Saulo Furletti, Éden Santana Campos Amorim e Pedro Henrique Pereira Daldegan meus sinceros agradecimentos por comporem minha banca. Os comentários e sugestões dados foram de altíssimo valor e contribuíram grandemente para o meu conhecimento e para a qualidade final do trabalho.

Minha gratidão, também, aos demais professores e à coordenação do PROFMAT do Cefet-MG por todo o profissionalismo apresentado durante o decorrer do curso.

Por fim, aos amigos, familiares, colegas, tanto de trabalho quanto de turma, e demais pessoas que de alguma maneira contribuíram para o desenvolvimento desse trabalho, meus sinceros agradecimentos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

O desempenho dos estudantes brasileiros nas edições do Programa Internacional de Avaliação de Alunos (PISA) até 2018, com o foco em Matemática, demonstrou que estes obtiveram uma nota bem aquém da média internacional. Com base no resultado obtido na edição de 2012 do PISA, um texto apresentado pela Sociedade Brasileira de Matemática levantou, entre outras, a seguinte questão a ser respondida pelos professores: “O meu ensino deve enfatizar os conceitos matemáticos ou como esses conceitos são aplicados no mundo real?” Como resposta, foi incentivado o ensino da Matemática em sua forma aplicada, ao invés de apenas em sua forma pura. Dentre as áreas do ensino da Matemática incluídas no plano da Educação Básica, optou-se, então, por apresentar uma aplicação da Geometria Analítica e da Álgebra Linear. A proposta desse trabalho é uma aplicação de grupos algébricos matriciais na Robótica, por meio de um ambiente de programação e um simulador de robôs, como material alternativo e viável para o ensino-aprendizagem de Matemática. Dentre as atividades possíveis, este trabalho trata da Cinemática Direta de um braço robótico, que é a descrição do posicionamento e orientação do efetuador, que é o último elemento deste equipamento, tendo a informação da localização de sua base, o tamanho de seus elos e a posição de suas juntas. A aplicação dos grupos matriciais se deu por meio da notação apresentada por Denavit e Hartenberg. Como produto final, são apresentadas algumas atividades que podem ser aplicadas no ambiente escolar. Como incentivo à aplicação dos métodos sugeridos por este trabalho, foram utilizadas ferramentas gratuitas de programação e de simulação de Robótica.

Palavras-chave: Matemática Aplicada. Geometria Analítica. Álgebra Linear. Grupos Matriciais. Robótica.

Abstract

The performance of Brazilian students in the editions of the Programme for International Student Assessment (PISA) until 2018, with a focus on Mathematics, showed that they obtained a grade below the international average. Based on the result obtained in PISA of 2012, a text presented by the Brazilian Mathematical Society raised, among others, the following question to be answered by teachers: “Should my teaching emphasize mathematical concepts or how those concepts are applied in the real world?” As an answer, the teaching of mathematics in its applied form was encouraged, instead of just in its pure form. Among the areas of Mathematics education included in the basic education plan, it was decided, then, to present an application of Analytical Geometry and Linear Algebra. The purpose of this work is an application of matrix algebraic groups in Robotics, through a programming environment and a robot simulator, as an alternative and viable material for the teaching-learning of Mathematics. Among the possible activities, this work deals with the Direct Kinematics of a robotic arm, which is the description of the positioning and orientation of the end-effector, that is last element of this equipment, with information on the location of its base, the size of its links and the position of its joints. The matrix groups were applied through the notation presented by Denavit and Hartenberg. As final product, some activities that can be applied in the school environment are presented. As an incentive to apply the methods suggested by this work, free programming and robotics simulation tools were used.

Keywords: Applied Mathematics. Analytical Geometry. Linear Algebra. Matrix Groups. Robotics.

Lista de Abreviaturas

Abreviaturas utilizadas neste trabalho:

API	Interface de programação de aplicações
BNCC	Base Nacional Curricular Comum
OCDE	Organização para a Cooperação e Desenvolvimento Econômico
PCN	Parâmetros Curriculares Nacionais
PISA	Programa Internacional de Avaliação de Alunos
PROFMAT	Mestrado Profissional em Matemática
SBM	Sociedade Brasileira de Matemática

Lista de Símbolos

Símbolos e notações utilizadas neste trabalho:

α	letra grega Alfa	$[F(\vec{u})]$	vetor coluna de uma transformação linear F sobre um vetor \vec{u}
β	letra grega Beta		
γ	letra grega Gama	\forall	para todo
θ	letra grega Teta	\in	pertence a
π	letra grega Pi	\exists	existe
σ	letra grega Sigma	$;$	tal que (em expressões matemáticas)
ϕ	letra grega Fi	\mathbb{R}	conjunto dos números reais
$[A]_{ij}$	termo da i -ésima linha e da j -ésima coluna de uma matriz A	\mathbb{Z}	conjunto dos números inteiros
\mathcal{B}	base de espaço vetorial	\equiv	congruente a
\overrightarrow{AB}	vetor que liga o ponto A ao ponto B	\approx	aproximadamente igual a
\vec{v}	vetor	\parallel	paralelo a
\mathbf{v}	vetor coluna	\perp	perpendicular (ou ortogonal) a
$\ \vec{v}\ $	norma de um vetor \vec{v}	$*$	operação binária

Lista de Figuras

2.1	Produto Vetorial	38
2.2	Regra da mão direita	39
2.3	Coordenadas Esféricas	39
2.4	Isometrias diretas em \mathbb{R}^2 e \mathbb{R}^3	43
2.5	Representação das matrizes identidade	47
2.6	Rotação de 30° no plano	50
2.7	Representação de $\text{Rot}(\vec{z}, 45^\circ)$	52
2.8	Representação de $\text{Trans}(\vec{p})\text{Rot}(\vec{z}, 30^\circ)$ em $\text{SE}(3)$	56
3.1	Posicionamento do triângulo no plano	58
3.2	Posicionamento do prisma no espaço	60
3.3	Braço robótico IRB 140 da marca <i>Asea Brown Boveri</i> (ABB)	63
3.4	Braço robótico planar	65
3.5	Ambiente de programação do Octave	66
3.6	Ambiente de Simulações do CoppeliaSim	68
3.7	Braço IRB 140 no CoppeliaSim após sua configuração	69
3.8	Braço IRB 140 no CoppeliaSim, com a junta 5 selecionada na Hierarquia da Cena e descrita no Texto de Informação	70
3.9	Dimensões do braço robótico IRB 140 da ABB em milímetros	71
3.10	Posição inicial do braço robótico simulado no Octave	76
3.11	Comparação das simulações no Octave e no CoppeliaSim - Teste 1	78
3.12	Comparação das simulações no Octave e no CoppeliaSim - Teste 2	79
3.13	Comparação das simulações no Octave e no CoppeliaSim - Teste 3	80
3.14	Interface gráfica de usuário para controle do braço robótico no Octave e no CoppeliaSim	81
3.15	Interface gráfica em 3D com ângulos aleatórios	82
3.16	Interface gráfica em 2D com ângulos aleatórios	83
3.17	Teste <i>online</i> com interfaces, em três dimensões	85
3.18	Teste <i>online</i> com interfaces, em duas dimensões	86
4.1	Isometria aplicada a elipse	90
4.2	Descrindo as coordenadas do ponto D e o ângulo entre o vetor \vec{x}_f e o eixo \vec{x}	91
4.3	Braço robótico no plano com juntas em estado inicial	94
4.4	Braço robótico no plano com juntas alteradas	95
B.1	Janelas de translação e rotação de objetos no CoppeliaSim	121
B.2	Janela de propriedades de juntas e janela de <i>scripts</i> no CoppeliaSim	123
C.1	Arquivos contidos na pasta de simulação	127

Lista de Tabelas

3.1	Juntas do braço robótico	70
3.2	Elos do braço robótico	72
3.3	Resultado dos testes no Octave	77

Sumário

1	Introdução	13
2	Fundamentação Teórica	18
2.1	Noções de Álgebra Linear	18
2.1.1	Adição e Multiplicação de Matrizes	20
2.1.2	Matriz Transposta	23
2.1.3	Determinante	24
2.1.4	Matriz Inversa	30
2.1.5	Espaço Vetorial Real	34
2.1.6	Transformação Geométrica	41
2.2	Grupos Matriciais	44
2.2.1	Grupo Ortogonal Especial	45
2.2.2	Grupo Euclidiano Especial	53
3	Desenvolvimento	57
3.1	Posicionamento de um Objeto Rígido	57
3.1.1	Posicionamento no Plano	57
3.1.2	Posicionamento no Espaço	59
3.1.3	Graus de Liberdade	61
3.2	Simulação do braço robótico	61
3.2.1	Definição de braço robótico	62
3.2.2	Notação de Denavit-Hartenberg	64
3.2.3	A linguagem de programação Octave	66
3.2.4	O <i>software</i> CoppeliaSim	67
3.2.5	Descrição dos elementos do braço robótico	69
3.2.6	Montagem do braço robótico no Octave	73
3.2.7	Simulação por interface gráfica de usuário	81
4	Atividades para sala de aula	87
4.1	Atividade 1: Elipse transladada e rotacionada	88
4.2	Atividade 2: Cinemática direta no plano	91
4.3	Atividade 3: Simulação computacional de braço robótico em duas dimensões	93
4.4	Atividade 4: Utilização de interface gráfica do Octave	96
5	Considerações finais	99
	Referências	102
	Anexos	

A	Algoritmos do Octave	105
A.1	Simulação de robô 3D no Octave	105
A.2	Simulação de robô 2D no Octave	108
A.3	Código para criar a Interface Gráfica (create_gui.m)	110
A.4	Função conect.m	116
A.5	Função set_target.m	116
A.6	Função robot_plot.m	117
B	Configuração do CoppeliaSim para as simulações	120
B.1	Configuração para o levantamento das juntas e elos	120
B.2	Configuração para o início da animação	121
C	Configuração de interfaces para o Octave	126

1 Introdução

O resultado dos estudantes brasileiros na edição de 2018 da prova do Programa Internacional de Avaliação de Alunos (PISA, da sigla em inglês) demonstrou que o desempenho nacional na disciplina Matemática está abaixo da média mundial[1]. Um relatório preliminar emitido pelo Ministério da Educação constatou que, por mais que a média tenha crescido entre 2003 e 2018, tal crescimento não foi estatisticamente significativo[2].

Fundamentado nos resultados da edição de 2015 do PISA, um texto da Organização para a Cooperação e Desenvolvimento Econômico (OCDE), organizadora do teste, traduzido pela Sociedade Brasileira de Matemática (SBM) levantou, dentre outras, a seguinte questão: “O meu ensino deve enfatizar os conceitos matemáticos ou como esses conceitos são aplicados no mundo real?”[3, p. 76]. Como resposta, foi evocado o debate clássico sobre se ensinar a Matemática em sua forma pura ou em sua forma aplicada, como resolução de situações-problema do cotidiano.

Sobre a abordagem aplicada do ensino, Andrade[4] constatou, por meio de sua pesquisa, que há “a necessidade de uma nova visão diante do modelo atual sob o qual acontece o ensino da Matemática, de modo que o aluno perceba sua importância e utilidade no seu cotidiano e assim, comece a adquirir interesse por tal disciplina”.

Em um contexto de baixo desempenho escolar, Brito, Moita e Lopes[5] frisaram a importância de se “discutir alternativas metodológicas no processo de ensino com objetivo de alcançar melhores resultados de aprendizagem”.

O desenvolvimento de tecnologias e a inclusão digital podem ser facilitadores do ensino da Matemática. Peralta e Guimarães[6] ressaltaram que surge a necessidade de formar professores atuantes em contextos informatizados e essa formação deveria contemplar ações que encorajassem e subsidiassem o professor para interagir com tecnologias, transformando-a.

Ainda como resposta à pergunta levantada sobre o ensino de Matemática em sua forma Pura e Aplicada, o texto apresentado pela SBM incentiva o professor para que

Dê aos alunos uma variedade de problemas que incluam problemas contextualizados, nos quais eles precisam aplicar o conhecimento para encontrar uma solução para um problema com o qual já se depararam na vida cotidiana. Pedagogias como aprendizado baseado em projetos ou baseado em problemas apresentam ao aluno problemas do mundo real que eles precisam resolver, geralmente como um time, aplicando as habilidades que acabaram de aprender. [3, p. 86]

Para que seja realizada uma aplicação eficaz de conceitos matemáticos em ambiente de sala de aula para a resolução de problemas, interagindo com tecnologias, é necessário que os docentes sejam treinados para tal. Dentre as soluções coordenadas pela SBM que visam o aperfeiçoamento da formação de professores, pode-se destacar o mestrado profissional em Matemática, o PROFMAT. Este programa visa incentivar a pesquisa e a produção de materiais e práticas pedagógicas inovadoras para o enriquecimento do processo de ensino e aprendizagem de Matemática na escola[7].

Baseado no incentivo de promover uma prática pedagógica para a resolução de problemas e tendo em vista a necessidade de contextualizar o ensino da Matemática com atividades práticas que sejam relevantes, optou-se pela elaboração de um trabalho na área de Matemática Aplicada. Dentre as áreas do ensino, a Geometria Analítica e a Álgebra Linear foram as escolhidas para o desenvolvimento desta dissertação. Este trabalho busca abranger uma aplicação de conceitos matemáticos dessas áreas em um ambiente computacional, fazendo o uso dos conhecimentos e formação do autor em engenharia alinhados com o conteúdo e o objetivo do PROFMAT.

Mais especificamente, a situação-problema escolhida foi a cinemática direta do braço robótico, isto é, a descrição do posicionamento de seu elemento final de posse de algumas configurações relevantes prévias. O trabalho foi desenvolvido em um ambiente computacional, por meio da linguagem de programação Octave[8] integrada com o simulador de robótica CoppeliaSim[9].

O produto final apresentado nessa dissertação é uma lista de atividades com aplicações de conceitos algébricos de grupos matriciais, os quais podem ser interpretados como transformações geométricas que modificam a posição de objetos no plano e no espaço, incluindo elementos que compõem os braços robóticos.

O público-alvo dessa dissertação são os professores da Educação Básica que queiram apresentar os conceitos matemáticos contidos nesse trabalho em sua forma aplicada. Alunos de graduação de Matemática e alunos da Educação Básica sob tutoria também podem

fazer o uso dos conteúdos apresentados. Além disso, devido às diversas áreas abordadas, professores do ensino técnico, engenheiros, profissionais de computação e de outras áreas correlacionadas também podem fazer o uso do material exposto.

Como a intenção é propor um material que seja relevante para o uso em sala de aula, fez-se necessário que o conteúdo fosse adequado às exigências do Ministério da Educação. Dessa forma, as atividades foram elaboradas de acordo com os requisitos presentes na Base Nacional Curricular Comum (BNCC), “[...] um documento de caráter normativo que define o conjunto orgânico e progressivo de aprendizagens essenciais que todos os alunos devem desenvolver ao longo das etapas e modalidades da Educação Básica”[10, p. 7].

Quanto ao conteúdo para o Ensino Médio, a BNCC determina que, no que tange ao conhecimento em relação ao pensamento geométrico, os estudantes desenvolvam habilidades para interpretar e representar a localização e o deslocamento de uma figura no plano cartesiano, além de identificar transformações isométricas[10, p. 527].

Sobre o ensino de Álgebra Linear na Educação Básica, a BNCC não possui uma orientação explícita, uma vez os termos “matriz”, “determinante” e correlatos não constam no texto. No entanto, a orientação de ensinar sistemas lineares torna este estudo implícito. Cristovão e Spiller[11] acreditam que o ensino de matrizes “deve ser paralelo ao ensino de sistemas, para auxiliar na sua compreensão, como ferramenta na resolução e interpretação de problemas, e não como um tópico isolado da matemática”.

Os Parâmetros Curriculares Nacionais (PCN) apontam também que

O desenvolvimento da Geometria e o aparecimento da Álgebra marcaram uma ruptura com os aspectos puramente pragmáticos da Matemática e impulsionaram a sistematização dos conhecimentos matemáticos, gerando novos campos: Geometria Analítica, Geometria Projetiva, Álgebra Linear entre outros[12, p. 24].

Dessa forma, pode-se concluir que alguns problemas geométricos podem ser resolvidos por ferramentas algébricas e vice-versa.

Quanto ao uso de tecnologias no estudo da Matemática, a BNCC propõe que sua utilização seja realizada a partir dos anos iniciais do Ensino Fundamental e afirma ainda que “tal valorização possibilita que, ao chegarem aos anos finais, eles possam ser estimulados a desenvolver o pensamento computacional, por meio da interpretação e da elaboração de algoritmos”[10, p. 529].

O estudo de grupos matriciais, bem como das demais estruturas algébricas, não

fazem parte do conteúdo exigido pela BNCC, logo não estão no cotidiano do docente. No entanto, como esse assunto é abordado em cursos de licenciatura em Matemática, espera-se que o professor tenha alguma familiaridade com os conceitos apresentados. Uma vez que os grupos matriciais são utilizados neste trabalho como ferramentas para a cinemática direta dos braços robóticos, como um meio para esse fim, tais grupos podem ser expostos em sala apenas pelas suas propriedades e pelo seu significado geométrico, sem a necessidade de apresentar as demonstrações em um primeiro momento. E, apesar das dificuldades que possam existir, a SBM orienta também que “os professores devem cobrir os elementos fundamentais do currículo de matemática, mas ainda assim encontrar tempo para expor os alunos a problemas que promovam um entendimento conceitual e ativem suas habilidades cognitivas”[10, p. 529]. Dessa forma, o curto tempo a ser utilizado para explicar o significado geométrico das matrizes que pertencem aos grupos apresentados contribuirá para a sua utilização, tornando a explicação, então, relevante para a resolução dos problemas propostos.

Entre as dissertações mais relevantes do acervo do PROFMAT que apresentam o conteúdo de posicionamento de braços robóticos, pode-se destacar Costa[13], que enfatizou o uso da notação de Denavit-Hartenberg, a qual é exposta na Seção 3.2.2.

Mesquita[14] abordou as transformações ortogonais envolvidas na cinemática direta do braço robótico. Após apresentar uma aplicação por meio do *software* Geogebra, Carvalho[15] pôde concluir que o “[...] procedimento de modelagem e ensino de matemática por meio da robótica, em especial, do braço mecânico, permite trabalhar diversos conceitos matemáticos, relacioná-los com a prática de forma interdisciplinar e contextualizada e não apenas os citados nas situações de aprendizagens”.

Quanto à introdução do uso do Octave no estudo de matrizes, Neto afirma que esta linguagem de programação é

[...] uma ferramenta que agrega valor tanto para os estudantes quanto para professores. Os professores obtêm uma nova ferramenta, que pode ser de grande auxílio nos mais diversos conteúdos de matemática, auxiliando no despertar de um maior interesse pela matemática. Os alunos, por sua vez, ganham uma nova perspectiva sobre a matemática, além de conhecer e manusear um *software* que propicia grandes possibilidades intelectuais[16].

Nunes[17] optou por desenvolver um estudo usando linguagem de programação, com o *software* Octave, para trabalhar os conteúdos de matrizes, como alternativa metodológica,

obtendo bons resultados.

A relevância do ensino de Robótica na atualidade pode ser notado, também, pelo fato de Rousseau e Saint-Abin[18] dedicarem um capítulo do seu livro, que pertence à Coleção PROFMAT, para tratar deste assunto.

O diferencial deste trabalho é a ênfase teórica dada à vantagem da representação do posicionamento de objetos rígidos por meio de grupos matriciais e na sugestão de atividades de robótica por meio de ambientes computacionais ainda não apresentados. Vale ressaltar que os *softwares* apresentados neste trabalho são soluções gratuitas.

Além do ambiente de ensino de programação e de conceitos matemáticos, espera-se que seja desenvolvido um bom raciocínio lógico e noção espacial. Pode-se ressaltar também a possibilidade de integração com a disciplina Física, além da ludicidade do tema proposto, que pode influenciar positivamente o interesse dos alunos.

Quanto à disposição do conteúdo neste trabalho, no Capítulo 2 é apresentada a fundamentação teórica de Matemática utilizada ao longo da dissertação. Na primeira seção são expostas as noções de Álgebra Linear, incluindo matrizes, determinantes, espaço vetorial e transformações lineares e geométricas. Na segunda seção é exposto o conteúdo de grupos matriciais.

No Capítulo 3 é apresentado o desenvolvimento, sendo aplicados os conceitos expostos no capítulo anterior. Na Seção 3.1 são expostas algumas considerações sobre o posicionamento de objetos rígidos no plano e no espaço. Na Seção 3.2 é feita uma breve definição de braço robótico e, em seguida, é realizada a simulação deste equipamento em ambiente computacional.

No Capítulo 4 é apresentado o produto final dessa dissertação, que é a lista contendo quatro atividades para sala de aula, fazendo uso dos conceitos abordados neste trabalho.

Por fim, no Capítulo 5 são expostas as considerações finais acerca da elaboração desta dissertação, além de sugestões de trabalhos futuros.

2 Fundamentação Teórica

Neste capítulo são expostos os conceitos utilizados no desenvolvimento dessa dissertação. Na Seção 2.1 são abordadas noções de Álgebra Linear, conceituando matrizes, vetores, espaço vetorial e conceitos relacionados, além de transformações lineares e geométricas. Na Seção 2.2, por sua vez, são expostos os conceitos de grupos matriciais, que são aplicados no Capítulo 3.

2.1 Noções de Álgebra Linear

A fim de facilitar a compreensão das seções a seguir, vale relembrar alguns conceitos de Álgebra Linear. Inicialmente, tem-se a seguinte definição:

Definição 2.1: Sejam $m, n \in \mathbb{N}$. Define-se como matriz $A_{m \times n}$, a tabela de dimensões $m \times n$, isto é, m linhas e n colunas, onde os seus termos podem ser descritos como $a_{ij} \in \mathbb{R}$, sendo $1 \leq i \leq m$ e $1 \leq j \leq n$. Dessa forma, tem-se que:

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}$$

O índice i indica a linha da matriz e o índice j , sua coluna. Se $m = n$, diz-se que a matriz é quadrada e de ordem n .

Para facilitar a compreensão em algumas expressões, as dimensões da matriz foram suprimidas. As matrizes quadradas foram denotadas com somente um subíndice, como, por exemplo, a matriz A de ordem n , que será escrita A_n ao invés de $A_{n \times n}$. Quando houve a necessidade de numerar as matrizes, tal numeração foi escrita entre parênteses no subíndice, como, por exemplo, em $A_{(1)}$, $A_{(2)}$, $A_{(3)}$ etc.

Os termos da diagonal da matriz são aqueles das posições cujos índices são iguais, isto é, $i = j$. A matriz quadrada que possui os termos de sua diagonal com valor 1 e os demais termos 0 é chamada de matriz identidade. A matriz identidade de ordem n é denotada I_n .

Exemplo 2.1.1: A matriz $I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ é a matriz identidade de ordem 3.

Definição 2.2: Seja a matriz $A_{m \times n}$. Para $i, j \in \{1, 2, \dots, m\}$, $i \neq j$, $k \in \mathbb{R}$ e $k \neq 0$, define-se como operações elementares nas linhas as seguintes transformações:

1. Troca de linhas, denotada por $e : L_i \leftrightarrow L_j$;
2. Multiplicação de linha por uma constante não nula k , denotada $e : L_i \rightarrow kL_i$;
3. Adição de linha, onde é adicionada a uma linha L_i os termos de uma linha L_j multiplicados por uma constante k . É denotada $e : L_i \rightarrow L_i + kL_j$.

As operações realizadas nas linhas para se obter as matrizes elementares podem, também, ser realizadas nas colunas C_i e C_j , com $i, j \in \{1, 2, \dots, n\}$. Dessa forma, tem-se as operações $C_i \leftrightarrow C_j$, $C_i \rightarrow kC_i$ e $C_i \rightarrow C_i + kC_j$, de acordo com as notações apresentadas anteriormente. No entanto, neste trabalho são utilizadas apenas as operações envolvendo as linhas.

Proposição 2.3: Toda operação elementar é reversível, isto é, seja e uma operação elementar e uma matriz A , existe uma operação e' , tal que $e'(e(A)) = e(e'(A)) = A$.

Demonstração. Se $e : L_i \leftrightarrow L_j$, então $e' : L_i \leftrightarrow L_j$. Se $e : L_i \rightarrow kL_i$, então $e' : L_i \rightarrow \frac{1}{k}L_i$. Por último, se $e : L_i \rightarrow L_i + kL_j$, então $e' : L_i \rightarrow L_i - kL_j$. \square

Definição 2.4: Define-se como matriz elementar de ordem n a matriz E obtida por meio de uma operação elementar na matriz identidade I_n . Seja uma operação elementar e , tem-se, então, a notação:

$$E = e(I_n)$$

Exemplo 2.1.2: São exemplos de matrizes elementares a matriz $E_{(1)} = e_1(I_2) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$,

com $e_1 : L_1 \leftrightarrow L_2$, a matriz $E_{(2)} = e_2(I_3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix}$, com $e_2 : L_3 \rightarrow 4L_3$ e a matriz

$E_{(3)} = e_3(I_3) = \begin{bmatrix} 1 & 3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, com $e_3 : L_1 \rightarrow L_1 + 3L_2$.

Definição 2.5: As matrizes $A_{m \times n}$ e $B_{m \times n}$ são chamadas de equivalentes por linhas, denotado $A \sim B$, se existem um número finito de operações elementares que, aplicadas a uma matriz, se obtém a outra. Em símbolos, tem-se que, com $k \in \mathbb{N}$:

$$A = e_k(\dots(e_2(e_1(B))))$$

A matriz que possui dimensões $m \times 1$ é chamada de matriz coluna. A matriz que, por sua vez, possui dimensões $1 \times m$ é chamada de matriz linha.

A matriz de dimensões $m \times n$ cujos termos são 0 é chamada matriz nula, e é denotada como $0_{m \times n}$. Por exemplo, $0_{2 \times 3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ e $0_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

As matrizes contidas no interior de uma matriz são denominadas submatrizes dessa.

Exemplo 2.1.3: A matriz $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ possui como submatrizes a matriz quadrada

$B = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$, a matriz linha $C = [7 \ 8]$, a matriz coluna $D = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$ e a matriz $E = [9]$, e

pode ser escrita como $A_3 = \begin{bmatrix} B_2 & D_{2 \times 1} \\ C_{1 \times 2} & E_1 \end{bmatrix}$.

2.1.1 Adição e Multiplicação de Matrizes

Sejam as matrizes A e B de dimensões $m \times n$. A adição $C = A + B$ se dá termo a termo, ou seja, o termo geral c_{ij} é:

$$c_{ij} = a_{ij} + b_{ij},$$

onde $1 \leq i \leq m$ e $1 \leq j \leq n$, com $i, j, m, n \in \mathbb{N}$.

Exemplo 2.1.4: A adição das seguintes matrizes de ordem 3 é da forma:

$$\begin{bmatrix} 1 & -5 & 3 \\ -4 & 0 & 10 \\ -2 & 2 & 9 \end{bmatrix} + \begin{bmatrix} 0 & -2 & 5 \\ 4 & 10 & -3 \\ 7 & -2 & -5 \end{bmatrix} = \begin{bmatrix} 1+0 & -5+(-2) & 3+5 \\ -4+4 & 0+10 & 10+(-3) \\ -2+7 & 2+(-2) & 9+(-5) \end{bmatrix} = \begin{bmatrix} 1 & -7 & 8 \\ 0 & 10 & 7 \\ 5 & 0 & 4 \end{bmatrix}$$

Sejam as matrizes A , B e C , de dimensões $m \times n$, e também a matriz nula $0_{m \times n}$, de mesmas dimensões, valem as seguintes propriedades, quanto à adição de matrizes:

Comutatividade $A + B = B + A$

Associatividade $(A + B) + C = A + (B + C)$

Elemento neutro $A + 0_{m \times n} = 0_{m \times n} + A = A$

Sejam as matrizes $A_{m \times n}$ e $B_{o \times p}$, com $m, n, o, p \in \mathbb{N}$, tem-se que o produto $C = A \cdot B = AB$ é uma matriz de dimensões $m \times p$, e só é possível se $n = o$. Para os termos de A descritos por a_{ij} , com $1 \leq i \leq m$ e $1 \leq j \leq n$, e os termos de B descritos por b_{kl} , com $1 \leq k \leq n$ e $1 \leq l \leq p$, onde $i, j, k, l \in \mathbb{N}$, tem-se que o termo geral c_{rs} da matriz $C = AB$, com $1 \leq r \leq m$ e $1 \leq s \leq p$, é dado por:

$$c_{rs} = \sum_{t=1}^n a_{rt} \cdot b_{ts} \tag{2.1}$$

Sendo assim, o valor de C é obtido da seguinte forma:

$$C = A \cdot B = \begin{bmatrix} a_{11} \cdot b_{11} + a_{12} \cdot b_{21} & a_{11} \cdot b_{12} + a_{12} \cdot b_{22} & \dots & a_{11} \cdot b_{1p} + a_{12} \cdot b_{2p} \\ + \dots + a_{1n} \cdot b_{n1} & + \dots + a_{1n} \cdot b_{n2} & & + \dots + a_{1n} \cdot b_{np} \\ \\ a_{21} \cdot b_{11} + a_{22} \cdot b_{21} & a_{21} \cdot b_{12} + a_{22} \cdot b_{22} & \dots & a_{21} \cdot b_{1p} + a_{22} \cdot b_{2p} \\ + \dots + a_{2n} \cdot b_{n1} & + \dots + a_{2n} \cdot b_{n2} & & + \dots + a_{2n} \cdot b_{np} \\ \\ \vdots & \vdots & \ddots & \vdots \\ \\ a_{m1} \cdot b_{11} + a_{m2} \cdot b_{21} & a_{m1} \cdot b_{12} + a_{m2} \cdot b_{22} & \dots & a_{m1} \cdot b_{1p} + a_{m2} \cdot b_{2p} \\ + \dots + a_{mn} \cdot b_{n1} & + \dots + a_{mn} \cdot b_{n2} & & + \dots + a_{mn} \cdot b_{np} \end{bmatrix}$$

Exemplo 2.1.5: Se $A_{4 \times 3} = \begin{bmatrix} 2 & 4 & 7 \\ 1 & 0 & 3 \\ 10 & 8 & 6 \\ 11 & 9 & 5 \end{bmatrix}$ e $B_{3 \times 2} = \begin{bmatrix} 1 & 4 \\ 2 & 3 \\ 7 & 6 \end{bmatrix}$, tem-se que:

$$A \cdot B = \begin{bmatrix} 2 \cdot 1 + 4 \cdot 2 + 7 \cdot 7 & 2 \cdot 4 + 4 \cdot 3 + 7 \cdot 6 \\ 1 \cdot 1 + 0 \cdot 2 + 3 \cdot 7 & 1 \cdot 4 + 0 \cdot 3 + 3 \cdot 6 \\ 10 \cdot 1 + 8 \cdot 2 + 6 \cdot 7 & 10 \cdot 4 + 8 \cdot 3 + 6 \cdot 6 \\ 11 \cdot 1 + 9 \cdot 2 + 5 \cdot 7 & 11 \cdot 4 + 9 \cdot 3 + 5 \cdot 6 \end{bmatrix} = \begin{bmatrix} 59 & 62 \\ 22 & 22 \\ 68 & 100 \\ 64 & 101 \end{bmatrix}$$

Sejam as matrizes $A_m, B_{m \times n}, C_{n \times m}, D_n, E_m, F_{m \times n}, G_{n \times m}$ e H_n . Utilizando a notação de submatrizes, tem-se que:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} A \cdot E + B \cdot G & A \cdot F + B \cdot H \\ C \cdot E + D \cdot G & C \cdot F + D \cdot H \end{bmatrix}$$

Sejam as matrizes quadradas A, B e C , de ordem m , e também as matrizes nula 0_m e identidade I_m , de mesma ordem, tem-se as seguintes propriedades, quanto à multiplicação de matrizes:

Associatividade $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Elemento nulo $A \cdot 0_m = 0_m \cdot A = 0_m$

Elemento neutro (identidade) $A \cdot I_m = I_m \cdot A = A$

Além das propriedades listadas, a multiplicação de matrizes não é comutativa, isto é, em geral $A \cdot B \neq B \cdot A$, sendo válida a igualdade em alguns casos especiais. E quanto à multiplicação em relação à adição, têm-se a propriedade:

Distributividade $A \cdot (B + C) = A \cdot B + A \cdot C$

A multiplicação de um escalar $r \in \mathbb{R}$ por uma matriz $A_{m \times n}$ é da forma:

$$r \cdot A_{m \times n} = r \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} ra_{11} & ra_{12} & \cdots & ra_{1n} \\ ra_{21} & ra_{22} & \cdots & ra_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ra_{m1} & ra_{m2} & \cdots & ra_{mn} \end{bmatrix}$$

2.1.2 Matriz Transposta

Definição 2.6: Seja a matriz $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$. A matriz transposta de A ,

denotada A^T , é dada por:

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

Sendo assim, o elemento geral da matriz transposta é tal que:

$$[A^T]_{ij} = [A]_{ji}$$

As dimensões de A^T são $n \times m$. Se A é quadrada, tem-se que A^T também é quadrada. Os termos da diagonal da matriz quadrada mantêm-se na mesma posição em sua transposta.

Exemplo 2.1.6: A transposta da matriz $A = \begin{bmatrix} 1 & 15 & 8 \\ 7 & 12 & 2 \\ 4 & 10 & 14 \end{bmatrix}$ é a matriz

$$A^T = \begin{bmatrix} 1 & 7 & 4 \\ 15 & 12 & 10 \\ 8 & 2 & 14 \end{bmatrix}.$$

Proposição 2.7: Para a transposta do produto de matrizes vale o seguinte resultado:

$$(AB)^T = B^T A^T$$

Demonstração. Seja $[(AB)^T]_{ij}$ o termo geral da matriz $(AB)^T$. Dessa forma, tem-se que:

$$\begin{aligned}
 [(AB)^T]_{ij} &= [AB]_{ji} \\
 &= \sum_{k=1}^p A_{jk} \cdot B_{ki} \\
 &= \sum_{k=1}^p [A^T]_{kj} \cdot [B^T]_{ik} \\
 &= \sum_{k=1}^p [B^T]_{ik} \cdot [A^T]_{kj} \\
 &= [B^T A^T]_{ij}
 \end{aligned}$$

Como $[B^T A^T]_{ij}$ é o termo geral da matriz $B^T A^T$, tem-se a validade da igualdade enunciada. □

2.1.3 Determinante

Seja o natural $n \geq 1$ e o conjunto $\mathbb{N}_n = \{1, 2, 3, \dots, n\}$.

Definição 2.8: Uma função bijetiva $\sigma : \mathbb{N}_n \rightarrow \mathbb{N}_n$ é denominada uma permutação do conjunto \mathbb{N}_n e indicada como:

$$\sigma = \begin{bmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{bmatrix}$$

O conjunto de todas as permutações de \mathbb{N}_n é denominado S_n . A cardinalidade de S_n , denotada $\#S_n$, é dada por $n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1 = n!$.

Exemplo 2.1.7: Uma das permutações para $n = 3$ é

$$\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix}$$

Definição 2.9: Seja uma permutação $\sigma = \begin{bmatrix} 1 & 2 & \cdots & n \\ \sigma(1) & \sigma(2) & \cdots & \sigma(n) \end{bmatrix}$ de \mathbb{N}_n . Atribuindo ao natural r a quantidade de pares (i, j) , sendo $1 \leq i < j \leq n \in \mathbb{N}$, tais que $\sigma(i) > \sigma(j)$,

tem-se que a paridade da permutação σ , denotada $sgn(\sigma)$, é dada por

$$sgn(\sigma) = \begin{cases} 1, & \text{se } r \text{ é par} \\ -1, & \text{se } r \text{ é ímpar} \end{cases}$$

Exemplo 2.1.8: Para a permutação $\sigma = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{bmatrix}$, os pares (i, j) , tais que $\sigma(i) > \sigma(j)$, são $(1, 3)$ e $(2, 3)$, pois $2 > 1$ e $3 > 1$, respectivamente. Dessa forma, tem-se que $r = 2$, que é par, logo $sgn(\sigma) = 1$.

Definição 2.10: Determinante é uma função que atribui um número real a uma matriz quadrada. Uma forma de se obter o valor do determinante de uma matriz

$A_n = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$, denotado $\det(A) = |A|$, é pela equação de Leibniz a seguir:

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix} = \sum_{\sigma \in S_n} \left(sgn(\sigma) \prod_{i=1}^n a_{i\sigma(i)} \right) \quad (2.2)$$

Exemplo 2.1.9: O determinante da matriz quadrada $A_1 = \begin{bmatrix} a_{11} \end{bmatrix}$ é a_{11} , pois só existe uma permutação em \mathbb{N}_1 , a saber $\sigma = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, logo $r = 0$ e $sgn(\sigma) = 1$, obtendo, então, que:

$$\det(A) = 1 \cdot a_{11} = a_{11}$$

Exemplo 2.1.10: Para calcular o determinante da matriz $B = \begin{bmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \end{bmatrix}$, de ordem 2, basta observar que:

1. As permutações em \mathbb{N}_2 são $\sigma_1 = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix}$ e $\sigma_2 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$, isto é, $S_2 = \{\sigma_1, \sigma_2\}$. Então, $r_1 = 0$ e $r_2 = 1$ e, por fim, $sgn(\sigma_1) = 1$ e $sgn(\sigma_2) = -1$.

2. Utilizando a Equação 2.2, vale que:

$$\det(B) = 1 \cdot b_{11} \cdot b_{22} + (-1) \cdot b_{12} \cdot b_{21} = b_{11}b_{22} - b_{12}b_{21}$$

Exemplo 2.1.11: Encontrar o determinante da matriz $C = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$.

Como a matriz C é de ordem 3, serão listadas todas as permutações σ de \mathbb{N}_3 , com seus respectivos valores de $sgn(\sigma)$:

$$\begin{array}{l} \sigma_1 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \quad sgn(\sigma_1) = 1 \\ \sigma_2 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 1 & 2 & 3 \end{bmatrix} \quad sgn(\sigma_2) = 1 \\ \sigma_3 = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad sgn(\sigma_3) = 1 \\ \sigma_4 = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix} \quad sgn(\sigma_4) = -1 \\ \sigma_5 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \\ 1 & 2 & 3 \end{bmatrix} \quad sgn(\sigma_5) = -1 \\ \sigma_6 = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix} \quad sgn(\sigma_6) = -1 \end{array}$$

Aplicando a Equação 2.2, o determinante da matriz C é dado por:

$$\begin{aligned} \det(C) &= 1 \cdot c_{11} \cdot c_{22} \cdot c_{33} + 1 \cdot c_{12} \cdot c_{23} \cdot c_{31} + 1 \cdot c_{13} \cdot c_{21} \cdot c_{32} + \\ &\quad (-1) \cdot c_{11} \cdot c_{23} \cdot c_{32} + (-1) \cdot c_{12} \cdot c_{21} \cdot c_{33} + (-1) \cdot c_{13} \cdot c_{22} \cdot c_{31} \\ \det(C) &= c_{11}c_{22}c_{33} + c_{12}c_{23}c_{31} + c_{13}c_{21}c_{32} - (c_{11}c_{23}c_{32} + c_{12}c_{21}c_{33} + c_{13}c_{22}c_{31}) \end{aligned}$$

Exemplo 2.1.12: Aplicando a Equação 2.2 à matriz $D = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix}$, é possível

verificar que seu determinante é:

$$\begin{aligned} \det(D) &= + d_{11} (d_{22}d_{33}d_{44} + d_{23}d_{34}d_{42} + d_{24}d_{32}d_{43} - d_{22}d_{34}d_{43} - d_{23}d_{32}d_{44} - d_{24}d_{33}d_{42}) \\ &\quad - d_{12} (d_{21}d_{33}d_{44} + d_{23}d_{34}d_{41} + d_{24}d_{31}d_{43} - d_{21}d_{34}d_{43} - d_{23}d_{31}d_{44} - d_{24}d_{33}d_{41}) \\ &\quad + d_{13} (d_{21}d_{32}d_{44} + d_{22}d_{34}d_{41} + d_{24}d_{31}d_{42} - d_{21}d_{34}d_{42} - d_{22}d_{31}d_{44} - d_{24}d_{32}d_{41}) \\ &\quad - d_{14} (d_{21}d_{32}d_{43} + d_{22}d_{33}d_{41} + d_{23}d_{31}d_{42} - d_{21}d_{33}d_{42} - d_{22}d_{31}d_{43} - d_{23}d_{32}d_{41}) \end{aligned}$$

Proposição 2.11: O determinante da matriz transposta é da forma:

$$\det(A^T) = \det(A)$$

Demonstração. A cada iteração do somatório da Equação de Leibniz 2.2 é utilizado um elemento de cada linha e de cada coluna da matriz, o que é equivalente a dizer que é usado um elemento de cada coluna e cada linha da transposta. Sendo assim, o determinante da matriz é o mesmo que o da sua transposta. \square

Ao se aplicar a Equação 2.2 à matriz identidade I_n é possível observar que a única permutação que não possui termos nulos no produtório é $\sigma = \begin{bmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{bmatrix}$ e, neste caso, $\text{sgn}(\sigma) = 1$. Conclui-se, então, que:

$$\det(I_n) = 1 \cdot 1^n = 1$$

Proposição 2.12: Seja a matriz $A_{m \times n}$ e a matriz elementar E_m , originada da aplicação da operação elementar e à matriz identidade I_m . Então a matriz EA resulta da aplicação da operação e na matriz A . Em símbolos, têm-se que:

$$e(I)A = EA = e(A)$$

Demonstração. Seja o produto $EA = F$. Serão analisados três casos, de acordo com as operações elementares apresentadas:

1. $e : L_i \leftrightarrow L_j$

Ao se multiplicar E por A , de acordo com a Equação 2.1, os elementos da matriz $F = EA$ são da forma:

$$f_{rs} = \sum_{t=1}^m e_{rt} \cdot a_{ts}$$

De acordo com a forma dos elementos da matriz E , com $i, j, s \in \{1, 2, \dots, m\}$, tem-se que:

- Para toda linha $r \in \{1, 2, \dots, m\}$, com $r \neq i$ e $r \neq j$, tem-se que o

elemento $e_{rr} = 1$ e $e_{rs} = 0$, para $r \neq s$. Dessa forma, tem-se que:

$$f_{rs} = \sum_{t=1}^m e_{rt} \cdot a_{ts} = e_{rr} \cdot a_{rs} = a_{rs}$$

- Para a linha i , tem-se que $e_{ii} = 0$, $e_{ij} = 1$ e $e_{is} = 0$, para $s \neq j$. Então, tem-se que:

$$f_{is} = \sum_{t=1}^m e_{it} \cdot a_{ts} = e_{ij} \cdot a_{js} = a_{js}$$

- Para a linha j , tem-se que $e_{jj} = 0$, $e_{ji} = 1$ e $e_{js} = 0$, para $s \neq i$. Logo, tem-se que:

$$f_{js} = \sum_{t=1}^m e_{jt} \cdot a_{ts} = e_{ji} \cdot a_{is} = a_{is}$$

Sendo assim, a i -ésima linha de F é igual a j -ésima linha de A , a j -ésima linha de F é igual a i -ésima linha de A e as demais linhas são iguais em F e A . Conclui-se, então, que o produto EA é igual à aplicação da operação $e : L_i \leftrightarrow L_j$ à matriz A .

2. $e : L_i \rightarrow kL_i$

Aplicando a Equação 2.1 à matriz $F = EA$, tem-se:

- Para toda linha $r \in \{1, 2, \dots, m\}$, com $r \neq i$, tem-se que o elemento $e_{rr} = 1$ e $e_{rs} = 0$, para $r \neq s$. Dessa forma, tem-se que os elementos de F são da forma:

$$f_{rs} = \sum_{t=1}^m e_{rt} \cdot a_{ts} = e_{rr} \cdot a_{rs} = a_{rs}$$

- Para a linha i , tem-se que $e_{ii} = k$, $e_{is} = 0$, para $s \neq i$. Então, tem-se que:

$$f_{is} = \sum_{t=1}^m e_{it} \cdot a_{ts} = e_{ii} \cdot a_{is} = ka_{is}$$

Dessa forma, a i -ésima linha de F é igual a k vezes a i -ésima linha de A , e as demais linhas são iguais em F e em A . Conclui-se, então, que F é igual à aplicação da operação $e : L_i \rightarrow kL_i$ à matriz A .

3. $e : L_i \rightarrow L_i + kL_j$

Por fim, ao se aplicar a Equação 2.1 à matriz $F = EA$, tem-se:

- Para toda linha $r \in \{1, 2, \dots, m\}$, com $r \neq i$, tem-se que o elemento

$e_{rr} = 1$ e $e_{rs} = 0$, para $r \neq s$. Dessa forma, tem-se que os elementos de F são da forma:

$$f_{rs} = \sum_{t=1}^m e_{rt} \cdot a_{ts} = e_{rr} \cdot a_{rs} = a_{rs}$$

- Para a linha i , tem-se que $e_{ii} = 1$, $e_{ij} = k$ e $e_{is} = 0$, para $s \neq i$ e $s \neq j$. Então, tem-se que:

$$f_{is} = \sum_{t=1}^m e_{it} \cdot a_{ts} = e_{ii} \cdot a_{is} + e_{ij} \cdot a_{js} = a_{is} + ka_{js}$$

Então, a i -ésima linha de F é igual a i -ésima linha de A somada a k vezes a i -ésima linha de A , e as demais linhas são iguais em F e em A . Conclui-se, finalmente, que o produto $E \cdot A$ é igual à aplicação da operação $e : L_i \rightarrow L_i + kL_j$ à matriz A .

□

Proposição 2.13: Sejam as matrizes elementares $E_{(1)}$, $E_{(2)}$ e $E_{(3)}$, obtidas a partir das operações elementares $L_i \leftrightarrow L_j$, $L_i \rightarrow k_1 L_i$ e $L_i \rightarrow k_2 L_j$, respectivamente, com $i, j \in \{1, 2, \dots, n\}$ e $k_1, k_2 \in \mathbb{R}$. Têm-se, então, que:

1. $\det(E_{(1)}) = -1$
2. $\det(E_{(2)}) = k_1$
3. $\det(E_{(3)}) = 1$

Demonstração. Ao aplicar a Equação 2.2 às matrizes elementares $E_{(1)}$, $E_{(2)}$ e $E_{(3)}$, tem-se os seguintes resultados, respectivamente:

1. Ao se permutar duas linhas L_i e L_j , a permutação que não possui valor nulo é da forma $\sigma = \begin{bmatrix} 1 & \dots & i & \dots & j & \dots & n \\ 1 & \dots & j & \dots & i & \dots & n \end{bmatrix}$, com $i, j \in \{1, 2, \dots, n\}$ e $i < j$, sem perda de generalidade. Nesse caso, de acordo com a Definição 2.9, como $\sigma(i) = j > i = \sigma(j)$, tem-se que $r = 1$ e, portanto, $\text{sgn}(\sigma) = -1$. Dessa forma, conclui-se que $\det(E_{(1)}) = -1$.
2. Na única permutação não nula $\sigma = \begin{bmatrix} 1 & 2 & \dots & n \\ 1 & 2 & \dots & n \end{bmatrix}$, com $\text{sgn}(\sigma) = 1$, a i -ésima linha possui o valor k_1 , o que implica que ao produtório é atribuído esse valor. Dessa forma, $\det(E_{(2)}) = k_1$.

3. A permutação $\sigma = \begin{bmatrix} 1 & \dots & i & \dots & j & \dots & n \\ 1 & \dots & j & \dots & i & \dots & n \end{bmatrix}$, com $\text{sgn}(\sigma) = 1$, é um resultado não nulo para o somatório. Como a permutação que utilizasse a j -ésima coluna da i -ésima linha utilizaria valores nulos da j -ésima linha, não existe outra permutação não nula. Sendo assim, têm-se que $\det(E_{(3)}) = 1$. \square

2.1.4 Matriz Inversa

Definição 2.14: Seja a matriz quadrada $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{bmatrix}$, de termo geral a_{ij} .

O número real A_{ij} , chamado cofator do elemento a_{ij} de A , é o determinante da matriz quadrada de ordem $(m - 1)$ formada pelos termos de A , exceto a linha i e a coluna j , multiplicado por $(-1)^{i+j}$.

Exemplo 2.1.13: Na matriz $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$, o cofator A_{21} , referente ao termo $a_{21} = 4$,

é dado por:

$$A_{21} = (-1)^{2+1} \begin{vmatrix} 2 & 3 \\ 8 & 9 \end{vmatrix} = 6$$

Uma outra maneira de calcular os determinantes de uma matriz de ordem $n \geq 2$, por meio do uso de cofatores, é utilizando o teorema de Laplace, onde, escolhida uma linha i , é realizado o somatório do produto de todos os termos desta linha por seus cofatores em todas as colunas. O teorema de Laplace é descrito pela seguinte equação:

$$\det(A) = \sum_{j=1}^n a_{ij} A_{ij} \tag{2.3}$$

Ao se aplicar a Equação 2.3 à matriz de ordem 4 do Exemplo 2.1.12, tomando a linha $i = 1$, encontra-se:

$$\begin{aligned}
 \det(D) &= \sum_{j=1}^4 d_{1j} D_{1j} = d_{11} \begin{pmatrix} (-1)^{1+1} \begin{vmatrix} d_{22} & d_{23} & d_{24} \\ d_{32} & d_{33} & d_{34} \\ d_{42} & d_{43} & d_{44} \end{vmatrix} \\ + d_{12} \begin{pmatrix} (-1)^{1+2} \begin{vmatrix} d_{21} & d_{23} & d_{24} \\ d_{31} & d_{33} & d_{34} \\ d_{41} & d_{43} & d_{44} \end{vmatrix} \\ + d_{13} \begin{pmatrix} (-1)^{1+3} \begin{vmatrix} d_{21} & d_{22} & d_{24} \\ d_{31} & d_{32} & d_{34} \\ d_{41} & d_{42} & d_{44} \end{vmatrix} \\ + d_{14} \begin{pmatrix} (-1)^{1+4} \begin{vmatrix} d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \\ d_{41} & d_{42} & d_{43} \end{vmatrix} \end{pmatrix} \\
 &= +d_{11} (d_{22}d_{33}d_{44} + d_{23}d_{34}d_{42} + d_{24}d_{32}d_{43} - d_{22}d_{34}d_{43} - d_{23}d_{32}d_{44} - d_{24}d_{33}d_{42}) \\
 &\quad -d_{12} (d_{21}d_{33}d_{44} + d_{23}d_{34}d_{41} + d_{24}d_{31}d_{43} - d_{21}d_{34}d_{43} - d_{23}d_{31}d_{44} - d_{24}d_{33}d_{41}) \\
 &\quad +d_{13} (d_{21}d_{32}d_{44} + d_{22}d_{34}d_{41} + d_{24}d_{31}d_{42} - d_{21}d_{34}d_{42} - d_{22}d_{31}d_{44} - d_{24}d_{32}d_{41}) \\
 &\quad -d_{14} (d_{21}d_{32}d_{43} + d_{22}d_{33}d_{41} + d_{23}d_{31}d_{42} - d_{21}d_{33}d_{42} - d_{22}d_{31}d_{43} - d_{23}d_{32}d_{41})
 \end{aligned}$$

O resultado encontrado é o mesmo do Exemplo 2.1.12, obtido utilizando a Equação 2.2, de Leibniz.

Definição 2.15: Entende-se como matriz adjunta de uma matriz quadrada A , denotada $\text{adj}(A)$, a matriz transposta dos cofatores de cada termo de A , ou seja:

$$\text{adj}(A) = \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{m1} \\ A_{12} & A_{22} & \cdots & A_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1m} & A_{2m} & \cdots & A_{mm} \end{bmatrix}$$

Quanto ao determinante da matriz adjunta, vale a seguinte equação:

$$\det(\text{adj}(A)) = (\det(A))^{n-1}$$

Definição 2.16: Seja a matriz quadrada A , de ordem m . A matriz inversa de A , denotada A^{-1} , é a que satisfaz:

$$A \cdot A^{-1} = A^{-1} \cdot A = I_m$$

Proposição 2.17: Toda matriz elementar E é invertível, e sua inversa E^{-1} também é elementar.

Demonstração. Seja a matriz $E = e(I)$. Dessa forma, de acordo com a Proposição 2.3, existe uma operação elementar e' , tal que $e'(e(I)) = e'(E) = I$. De forma similar, podemos escrever que $E' \cdot E = I$, de acordo com a Proposição 2.12. Assim, $E' = E^{-1}$, isto é, inversa de E , e, como e' é uma operação elementar, E^{-1} é uma matriz elementar. \square

Proposição 2.18: Uma matriz A ser invertível é equivalente a, ou A ser elementar, ou A puder ser escrita como um produto de matrizes elementares. Nesse caso, tem-se que $A \sim I$ e $\det(A) \neq 0$. Se forem aplicadas a mesma sequência de operações elementares que transformam A em I na própria matriz I , será obtida a inversa A^{-1} .

O cálculo da inversa de uma matriz A se dá por meio da equação a seguir:

$$A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$$

Se uma matriz A é invertível, o determinante de sua inversa é dada por:

$$\det(A^{-1}) = \frac{1}{\det(A)} \tag{2.4}$$

Exemplo 2.1.14: A matriz $A = \begin{bmatrix} 2 & 1 & 0 \\ 4 & -2 & 7 \\ 5 & 3 & -1 \end{bmatrix}$ possui determinante 1, logo é invertível.

A inversa de A é $A^{-1} = \begin{bmatrix} -19 & 1 & 7 \\ 39 & -2 & -14 \\ 22 & -1 & -8 \end{bmatrix}$, pois $A \cdot A^{-1} = I_3$.

Proposição 2.19: Seja a matriz elementar E e a matriz quadrada A , de mesma ordem. Sendo $\det(E) = d$, é válido que:

$$\det(EA) = d \det(A)$$

Teorema 2.20 (de Binet): Dadas as matrizes quadradas A e B de mesma ordem, tem-se que:

$$\det(AB) = \det(A) \det(B)$$

Demonstração. São considerados os seguintes casos:

1. A_n não é invertível.

Se $\det(A) = 0$, então $\det(AB) = 0$, pois, caso contrário, haveria uma matriz C tal que $(AB)C = I_n$, o que não é possível, uma vez que, por associatividade da multiplicação, a matriz BC seria inversa de A . Logo, tem-se que:

$$\begin{aligned} 0 &= 0 \det(B) \\ \Rightarrow \det(AB) &= \det(A) \det(B) \end{aligned}$$

2. A_n é invertível

Uma vez que $\det(A) \neq 0$, de acordo com a Proposição 2.18, essa pode ser escrita como um produto de matrizes elementares, obtendo:

$$A = E_{(1)}E_{(2)} \dots E_{(k)} \tag{2.5}$$

Dessa forma, para $\det(E_{(i)}) = d_i, \forall i \in \{1, 2, \dots, k\}$, aplicando o resultado da Proposição 2.19 sucessivamente, o determinante de A é:

$$\begin{aligned} \det(A) &= \det(E_{(1)}(E_{(2)} \dots E_{(k)})) \\ \det(A) &= d_1 \det(E_{(2)}(E_{(3)} \dots E_{(k)})) \\ &\vdots \\ \det(A) &= d_1 d_2 \dots d_k \end{aligned} \tag{2.6}$$

Escrevendo A na forma da Equação 2.5, aplicando a Proposição 2.19 e a associativa da multiplicação de matrizes, e, por fim, substituindo o valor determinante de A obtido na Equação 2.6, pode-se concluir que:

$$\begin{aligned} \det(AB) &= \det(E_{(1)}(E_{(2)} \dots E_{(k)}B)) \\ &= d_1 \det(E_{(2)}(E_{(3)} \dots E_{(k)}B)) \\ &\vdots \\ &= d_1 d_2 \dots d_k \det(B) \\ &= \det(A) \det(B) \end{aligned} \quad \square$$

2.1.5 Espaço Vetorial Real

Definição 2.21: O espaço \mathbb{R}^n é o conjunto de n -uplas ordenadas (x_1, x_2, \dots, x_n) reais. Em símbolos, tem-se que:

$$\mathbb{R}^n = \{(x_1, x_2, \dots, x_n); x_1, x_2, \dots, x_n \in \mathbb{R}\}$$

Seja o segmento orientado \overrightarrow{AB} , que liga o ponto $A = (a_1, a_2, \dots, a_n)$ ao ponto $B = (b_1, b_2, \dots, b_n)$, com $A, B \in \mathbb{R}^n$. Os segmentos equipolentes a \overrightarrow{AB} são aqueles que possuem:

- O mesmo comprimento de \overrightarrow{AB} ;
- A mesma direção de \overrightarrow{AB} , isto é, são paralelos ou colineares;
- O mesmo sentido de \overrightarrow{AB} .

Definição 2.22: Define-se como vetor $\vec{v} = \overrightarrow{AB}$, o conjunto de todos os segmentos orientados equipolentes a \overrightarrow{AB} . Cada segmento orientado equipolente a \overrightarrow{AB} é chamado representante de \vec{v} .

Por padrão, o representante do vetor \vec{v} a ser utilizado é aquele cujo ponto inicial é o ponto $O = (0, 0, 0, \dots, 0)$, com n zeros. Dessa forma, suas coordenadas são:

$$\vec{v} = (v_1, v_2, \dots, v_n),$$

onde $v_i = b_i - a_i, \forall i \in \{1, 2, \dots, n\}$, sendo a_i e b_i coordenadas dos pontos A e B , respectivamente.

O comprimento do vetor, também chamado de norma, é a distância entre as extremidades do vetor, e pode ser dado por:

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$$

Os vetores cuja norma tem valor 1 são chamados de vetores unitários.

Denomina-se o como vetor nulo, denotado como $\vec{0}$, aquele que possui norma nula e, por isso, não possui direção nem sentido.

Para algumas operações, os vetores da forma $\vec{v} = (v_1, v_2, \dots, v_n)$ foram escritos como matrizes coluna, chamados então de vetores coluna, da forma $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = [v_1 \ v_2 \ \dots \ v_n]^T$.

Fazendo o uso dessa notação, a norma de um vetor é dada por:

$$\|\vec{v}\| = \sqrt{\mathbf{v}^T \cdot \mathbf{v}} = \sqrt{[v_1, v_2, \dots, v_n] \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}} = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$$

Sendo assim, como $\|\vec{v}\| \geq 0$, é válido que:

$$\|\vec{v}\|^2 = \mathbf{v}^T \cdot \mathbf{v} \tag{2.7}$$

Sejam os vetores $\vec{u} = (u_1, u_2, \dots, u_n)$ e $\vec{v} = (v_1, v_2, \dots, v_n)$, e o escalar $\rho \in \mathbb{R}$. O vetor $\vec{r} = \vec{u} + \vec{v}$ é da forma $\vec{r} = (u_1 + v_1, u_2 + v_2, \dots, u_n + v_n)$. O vetor $\vec{s} = \rho\vec{u}$ é da forma $\vec{s} = (\rho u_1, \rho u_2, \dots, \rho u_n)$.

Definição 2.23: Pode-se somar um ponto $A = (a_1, a_2, \dots, a_n)$ com um vetor $\vec{v} = (v_1, v_2, \dots, v_n)$, resultando em um ponto $B = (a_1 + v_1, a_2 + v_2, \dots, a_n + v_n)$. Essa operação significa fazer uma translação do ponto A na direção, sentido e norma do vetor \vec{v} , o que será explicitado posteriormente na Definição 2.40.

Definição 2.24: Sejam os vetores $\vec{u} = (u_1, u_2, \dots, u_n)$ e $\vec{v} = (v_1, v_2, \dots, v_n)$. Define-se como produto escalar a operação entre vetores que resulta em um escalar, da forma:

$$\vec{u} \cdot \vec{v} = u_1 \cdot v_1 + u_2 \cdot v_2 + \dots + u_n \cdot v_n$$

Definição 2.25: A ortogonalidade entre dois vetores se dá quando o produto escalar entre dois vetores é nulo. Sendo assim, se $\vec{u} \cdot \vec{v} = 0$, diz-se que os vetores são ortogonais, e denota-se $\vec{u} \perp \vec{v}$.

Se dois vetores unitários são ortogonais, diz-se que são ortonormais.

Definição 2.26: O conjunto não vazio V , de vetores n -dimensionais com coordenadas reais, é definido como espaço vetorial real, com as seguintes propriedades:

- Quanto à adição de vetores $\vec{u}, \vec{v}, \vec{w} \in V$:

$$\text{Comutatividade} \quad \vec{u} + \vec{v} = \vec{v} + \vec{u}$$

$$\text{Associatividade} \quad (\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w})$$

$$\text{Elemento neutro} \quad \vec{u} + \vec{0} = \vec{0} + \vec{u} = \vec{u}$$

$$\text{Elemento inverso} \quad \exists -\vec{u}; \vec{u} + (-\vec{u}) = \vec{0}$$

- Quanto à multiplicação por escalar, com $r, s \in \mathbb{R}$ e $\vec{u} \in V$:

$$\text{Associatividade} \quad r(s\vec{u}) = (rs)\vec{u}$$

$$\text{Elemento neutro} \quad 1\vec{u} = \vec{u}1 = \vec{u}$$

- Quanto à adição de escalares e à multiplicação por escalar, com $r, s \in \mathbb{R}$ e $\vec{u}, \vec{v} \in V$:

$$\text{Distributiva de escalar pela soma de vetores} \quad r(\vec{u} + \vec{v}) = r\vec{u} + r\vec{v}$$

$$\text{Distributiva de soma de escalares por vetor} \quad (r + s)\vec{u} = r\vec{u} + s\vec{u}$$

A representação dos espaços vetoriais é feita por uma letra maiúscula e, quando a dimensão do espaço vetorial for relevante, este será notado com o expoente da dimensão. Por exemplo, o espaço vetorial V^3 é formado por vetores da forma $\vec{v} = (v_1, v_2, v_3)$, com $v_1, v_2, v_3 \in \mathbb{R}$.

Definição 2.27: Um vetor $\vec{v} \in V^n$ é uma combinação linear dos vetores $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$, se existem escalares $k_1, k_2, \dots, k_n \in \mathbb{R}$ tais que:

$$\vec{v} = k_1\vec{v}_1 + k_2\vec{v}_2 + \dots + k_n\vec{v}_n$$

Um conjunto de vetores de V^n é linearmente independente quando é válida a equivalência:

$$k_1\vec{v}_1 + k_2\vec{v}_2 + \dots + k_n\vec{v}_n = \vec{0} \iff k_1, k_2, \dots, k_n = 0$$

Caso haja um escalar $k_i \neq 0$, com $i \in \{1, 2, \dots, n\}$, em que seja válido que $k_1\vec{v}_1 + k_2\vec{v}_2 + \dots + k_n\vec{v}_n = \vec{0}$, diz-se que há uma dependência linear entre os vetores.

Definição 2.28: Em V^n , os vetores $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$, nesta ordem, formam uma base do espaço vetorial se são linearmente independentes e todo vetor $\vec{v} \in V^n$ pode ser escrito como combinação linear de $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$.

Uma base $\mathcal{B} = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$ é dita ortonormal se $\|\vec{e}_1\| = \|\vec{e}_2\| = \dots = \|\vec{e}_n\| = 1$ e ainda $\vec{e}_i \perp \vec{e}_j$, para todo $i, j \in \{1, 2, \dots, n\}$, com $i \neq j$. Se ainda $\vec{e}_1 = (1, 0, \dots, 0)$, $\vec{e}_2 = (0, 1, \dots, 0)$, \dots , $\vec{e}_n = (0, 0, \dots, 1)$, diz-se que a base \mathcal{B} é canônica.

Exemplo 2.1.15: Os vetores $\vec{e}_1 = (1, 0, 0)$, $\vec{e}_2 = (0, 1, 0)$ e $\vec{e}_3 = (0, 0, 1)$ formam a base canônica de V^3 , pois qualquer vetor do tipo $\vec{v} = (a, b, c) \in V^3$, com $a, b, c \in \mathbb{R}$, pode ser escrito como $\vec{v} = a\vec{e}_1 + b\vec{e}_2 + c\vec{e}_3$ e ainda $\|\vec{e}_1\| = \|\vec{e}_2\| = \|\vec{e}_3\| = 1$, $\vec{e}_1 \perp \vec{e}_2$, $\vec{e}_2 \perp \vec{e}_3$ e $\vec{e}_1 \perp \vec{e}_3$.

Para localizar um ponto P no espaço tridimensional, é necessário fazer uso da Definição 2.29 de Sistemas de Coordenadas a seguir, conforme apresentado por Boulos e Camargo[19]:

Definição 2.29: Sejam um ponto O do espaço tridimensional e uma base ortonormal $\mathcal{B} = \{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$ de V^3 . Define-se como sistema de coordenadas ortogonal de V^3 o par (O, \mathcal{B}) . Ao ponto $O = (0, 0, 0)$ se dá o nome de origem, e, sendo os pontos $A = O + \vec{e}_1$, $B = O + \vec{e}_2$ e $C = O + \vec{e}_3$, os vetores \vec{OA} , \vec{OB} e \vec{OC} correspondem aos eixos \vec{x} , \vec{y} e \vec{z} , respectivamente.

Dessa forma, todo vetor $\vec{v} \in V^3$ pode ser representado por um terno (a, b, c) . Esse representante é o que possui como ponto inicial a origem do sistema e a outra extremidade no ponto de coordenadas (a, b, c) , sendo representado pela combinação linear $\vec{v} = a\vec{x} + b\vec{y} + c\vec{z}$.

De forma similar, o ponto $O = (0, 0)$ e uma base ortonormal de V^2 descrevem um sistema de coordenadas do plano cartesiano, com os eixos \vec{x} e \vec{y} .

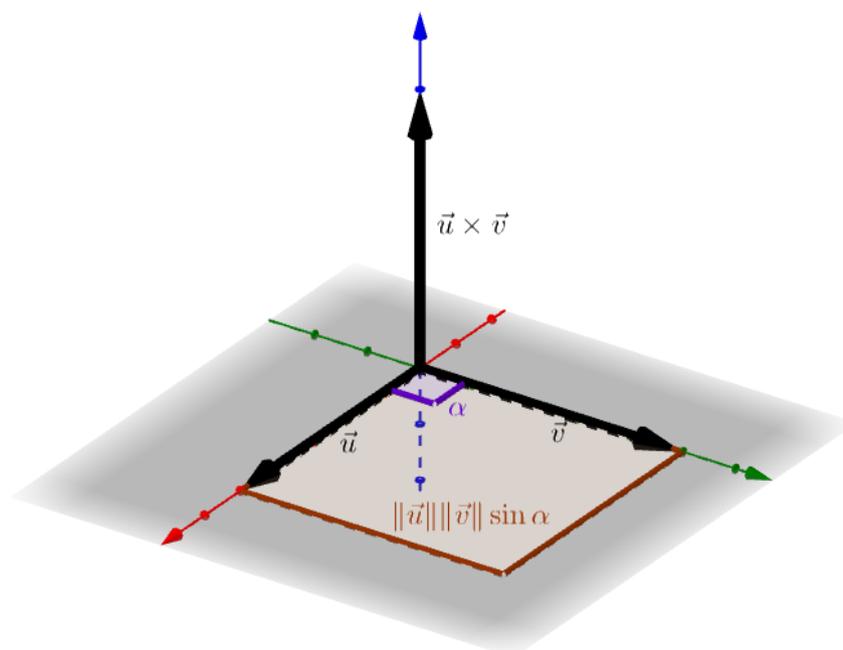
Definição 2.30: Sejam os vetores $\vec{u}, \vec{v} \in V^3$, e os seus paralelos unitários $\vec{u}_1, \vec{v}_1 \in V^3$, respectivamente. Define-se como ângulo entre os vetores, o comprimento do arco da circunferência de raio unitário contida no plano $O\vec{u}_1\vec{v}_1$ entre os vetores \vec{u}_1 e \vec{v}_1 . Como existem dois arcos entre os vetores, opta-se por representar o ângulo pelo de menor valor. Tal comprimento é denotado em radianos.

Uma outra maneira de representar os ângulos é por graus. Sendo ω a representação do ângulo em radianos e α a representação em graus, é válido que:

$$\frac{\omega}{\pi} = \frac{\alpha}{180^\circ}$$

Definição 2.31: Sejam os vetores $\vec{u}, \vec{v} \in V^3$. Define-se como produto vetorial de \vec{u} por \vec{v} , denotado por $\vec{u} \times \vec{v}$, o vetor cuja norma é $\|\vec{u}\| \|\vec{v}\| \sin \alpha$, sendo α o menor ângulo entre \vec{u} e \vec{v} , direção normal ao plano $O\vec{u}\vec{v}$, e sentido como o descrito na Figura 2.1.

Figura 2.1: Produto Vetorial



Fonte: Elaborado pelo autor.

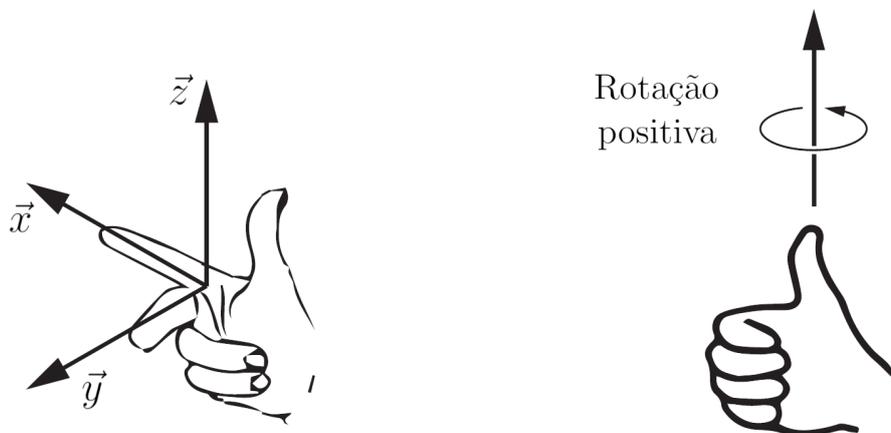
Uma base ortonormal $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ de V^3 é dita positiva se $\vec{e}_1 \times \vec{e}_2 = \vec{e}_3$, e negativa se $\vec{e}_1 \times \vec{e}_2 = -\vec{e}_3$. Uma representação de uma base positiva é dada pela “regra da mão direita”, que pode ser visto na Figura 2.2a.

Outra maneira de se representar objetos no espaço é por meio de coordenadas esféricas, as quais são definidas a seguir.

Definição 2.32: Define-se como coordenadas esféricas de um ponto P no espaço, a tripla ordenada (ρ, θ, ϕ) , onde ρ , também chamado de raio, é a distância do ponto P à origem, θ é o ângulo entre o segmento OP e o eixo \vec{z} e ϕ é ângulo entre o eixo \vec{x} e a projeção de OP no plano $O\vec{x}\vec{y}$. As coordenadas esféricas são representadas na Figura 2.3.

Figura 2.2: Regra da mão direita

- (a) Base ortogonal $(\vec{x}, \vec{y}, \vec{z})$ positiva (b) Rotação positiva em torno de um eixo

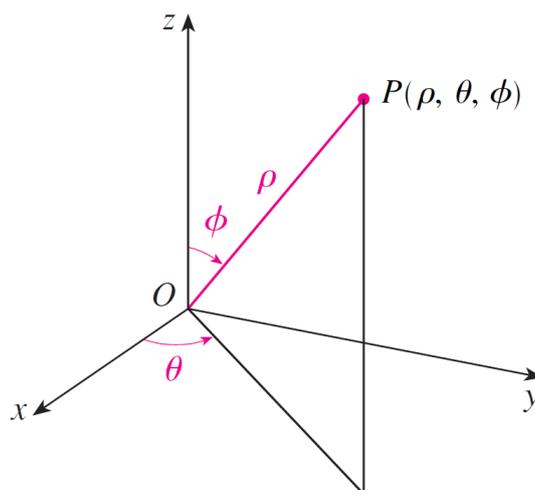


Fonte: Lynch e Park[20], com adaptações do autor.

Definição 2.33: Sejam os espaços vetoriais U sobre \mathbb{R}^n e V sobre \mathbb{R}^m . Uma função $F : U \rightarrow V$ é chamada transformação linear se, e somente se, $\forall u_1, u_2 \in U$ e $\rho \in \mathbb{R}$, vale que:

1. $F(u_1 + u_2) = F(u_1) + F(u_2)$
2. $F(\rho u_1) = \rho F(u_1)$

Figura 2.3: Coordenadas Esféricas



Fonte: Stewart[21].

Proposição 2.34: Sejam os espaços vetoriais U e V , de dimensões n e m , respectivamente. A transformação linear $F : U \rightarrow V$ pode ser descrita por uma matriz de dimensões $m \times n$.

Demonstração. Sejam as bases $\mathcal{B}_1 = \{u_1, u_2, \dots, u_n\}$ e $\mathcal{B}_2 = \{v_1, v_2, \dots, v_m\}$ de U e V , respectivamente. Dessa forma, pode-se determinar, de forma única, números $a_{ij} \in \mathbb{R}$, com $1 \leq i \leq m$ e $1 \leq j \leq n$, tais que:

$$F(\vec{u}_j) = a_{1j}v_1 + a_{2j}v_2 + \dots + a_{mj}v_m \quad (2.8)$$

Escrevendo um vetor $\vec{u} = (k_1, k_2, \dots, k_n)$, pertencente a U , da forma $\vec{u} = k_1u_1 + k_2u_2 + \dots + k_nu_n$, com $k_i \in \mathbb{R}$ para $1 \leq i \leq n$, tem-se que, de acordo com a Definição 2.33 e com a Equação 2.8:

$$\begin{aligned} F(\vec{u}) &= k_1F(\vec{u}_1) + k_2F(\vec{u}_2) + \dots + k_nF(\vec{u}_n) \\ &= k_1(a_{11}v_1 + \dots + a_{m1}v_m) + \dots + k_n(a_{1n}v_1 + \dots + a_{mn}v_m) \\ &= (a_{11}k_1 + \dots + a_{1n}k_n)v_1 + \dots + (a_{m1}k_1 + \dots + a_{mn}k_n)v_m \end{aligned}$$

Utilizando a notação de vetor coluna e escrevendo da forma matricial, tem-se que:

$$\begin{aligned} [F(\vec{u})] &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} \\ &= A \cdot \mathbf{u} \end{aligned}$$

A matriz A apresentada descreve a transformação linear. □

Definição 2.35: Um escalar λ é definido como autovalor de uma transformação linear $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, se existir um vetor \mathbf{v} não-nulo tal que $A\mathbf{v} = \lambda\mathbf{v}$. Ao vetor \mathbf{v} dá-se o nome de autovetor.

Proposição 2.36: Seja a matriz quadrada A e os seus autovalores $\lambda_1, \lambda_2, \dots, \lambda_n$. Tem-se a seguinte propriedade:

$$\det(A) = \lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_n$$

Definição 2.37: Sejam as bases $\mathcal{B} = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n\}$ e $\mathcal{B}' = \{\vec{v}'_1, \vec{v}'_2, \dots, \vec{v}'_n\}$, ambas do espaço vetorial V^n . Define-se como mudança de base, a transformação linear F , tal que $F(\vec{v}_i) = \vec{v}'_i \forall i \in \{1, 2, \dots, n\}$.

2.1.6 Transformação Geométrica

Definição 2.38: Define-se como objeto rígido um conjunto de pontos que formam uma figura, quando no plano, ou um sólido, quando no espaço.

Definição 2.39: Define-se como transformação geométrica a função cujo domínio e a imagem são conjuntos de pontos. Essa função é bijetiva, ou seja, para uma dada transformação f , para cada ponto A no domínio, existe apenas um único ponto B tal que $f(A) = B$ e, para cada ponto D do contra-domínio existe único ponto C tal que $f(C) = D$.

Definição 2.40: Isometrias são as transformações geométricas que preservam as distâncias e os ângulos. As isometrias se classificam entre diretas e opostas, sendo que as diretas são as que preservam a orientação e as opostas, as que a invertem.

As isometrias diretas em \mathbb{R}^3 são as seguintes:

- **Translação:** É o deslocamento do objeto no espaço, para alguma direção e sentido definidos. Para tal transformação, soma-se a cada ponto do objeto rígido, com coordenadas no espaço cartesiano, as coordenadas de um vetor que representa a transformação. Dessa forma, para cada dois pontos $A = (a_1, a_2, a_3)$ e $B = (b_1, b_2, b_3)$ do objeto rígido, serão somados as coordenadas do vetor $\vec{v} = (v_1, v_2, v_3)$ resultando nos pontos $A' = (a_1 + v_1, a_2 + v_2, a_3 + v_3)$ e $B' = (b_1 + v_1, b_2 + v_2, b_3 + v_3)$, de forma que $\overrightarrow{AA'} \equiv \overrightarrow{BB'} \equiv \vec{v}$ e $\overrightarrow{AA'} \parallel \overrightarrow{BB'}$.

Sendo assim, a transformação geométrica $\Omega : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, que representa a translação descrita por um vetor \vec{v} , é da forma:

$$\Omega(\vec{u}) = \vec{u} + \vec{v}$$

Para representar uma translação no espaço é utilizada a notação $\text{Trans}(\vec{v})$, onde \vec{v} é o vetor pertencente a V^3 que representa a translação.

- **Rotação:** É o giro do objeto em torno de um eixo representado por um vetor que passa pela origem.

A representação da rotação se dará por meio da notação $\text{Rot}(\vec{v}, \alpha)$, onde \vec{v} é o eixo

da rotação e α o ângulo da rotação.

A notação da transformação linear associada à transformação geométrica de rotação é apresentada na Proposição 2.51.

O sentido de rotação positivo pode ser descrito pela “regra da mão direita”, conforme a Figura 2.2b. Com isso, é válido dizer que:

$$\text{Rot}(\vec{v}, -\alpha) = \text{Rot}(-\vec{v}, \alpha)$$

- Translação e rotação: É a composição simultânea da translação e da rotação apresentadas anteriormente. É interpretado como uma translação seguida de uma rotação, nessa ordem.

A notação adotada é $\text{Trans}(\vec{v}) \text{Rot}(\vec{z}, \alpha)$, sendo \vec{v} o vetor que caracteriza a translação, \vec{z} o eixo da rotação e α o ângulo da rotação.

A representação por símbolos da operação da transformação geométrica de translação e rotação é apresentada na Proposição 2.57.

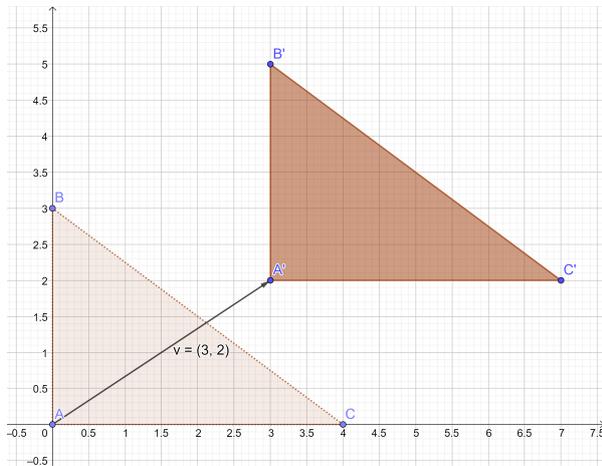
A Figura 2.4 ilustra alguns exemplos de isometrias diretas no plano e no espaço, com as notações adequadas.

Quando a translação ocorre no plano, a notação também é $\text{Trans}(\vec{v})$. O que muda é que $\vec{v} \in V^2$. A rotação, por sua vez, é denotada apenas $\text{Rot}(\alpha)$, pois se dá em torno da origem $O = (0, 0)$.

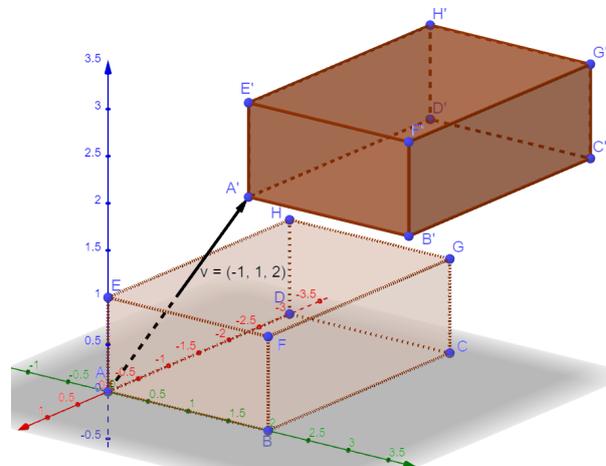
Este trabalho aborda as isometrias diretas, uma vez que essas transformações podem ser aplicadas a objetos rígidos, conforme é apresentado na Seção 3.1.

Figura 2.4: Isometrias diretas em \mathbb{R}^2 e \mathbb{R}^3

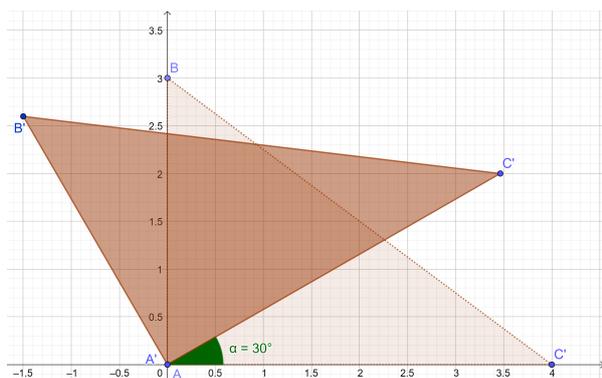
(a) $\text{Trans}(\vec{v})$, com $\vec{v} = (3, 2)$



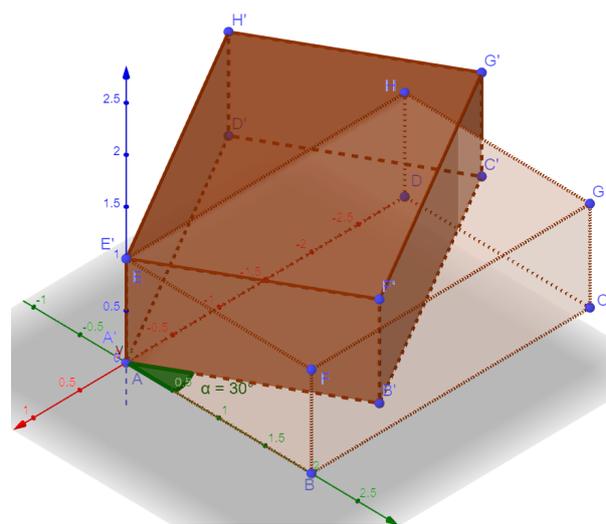
(b) $\text{Trans}(\vec{v})$, com $\vec{v} = (-1, 1, 2)$



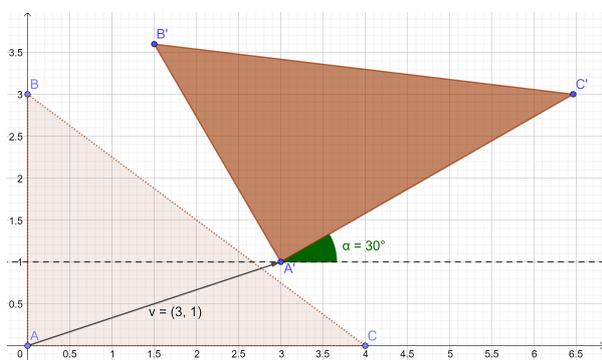
(c) $\text{Rot}(\alpha)$, com $\alpha = 30^\circ$



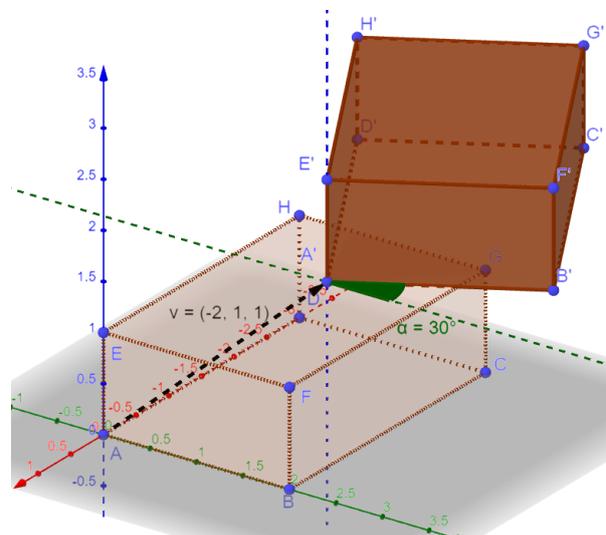
(d) $\text{Rot}(\vec{z}, \alpha)$, com $\alpha = 30^\circ$



(e) $\text{Trans}(\vec{v}) \text{Rot}(\alpha)$, com $\vec{v} = (3, 1)$ e $\alpha = 30^\circ$



(f) $\text{Trans}(\vec{v}) \text{Rot}(\vec{z}, \alpha)$, com $\vec{v} = (-2, 1, 1)$ e $\alpha = 30^\circ$



Fonte: Elaborado pelo autor.

2.2 Grupos Matriciais

Nessa seção é feita a apresentação do conceito de grupo algébrico e também são relacionados os grupos matriciais que foram utilizados no desenvolvimento deste trabalho.

Definição 2.41: Grupos são conjuntos em que dois elementos se associam por meio de uma operação binária para formar um terceiro, e esta operação é dotada de certas propriedades. Por exemplo, seja o grupo \mathbb{G} , os elementos $a, b, c \in \mathbb{G}$, e a operação binária $*$, têm-se, então, as seguintes propriedades:

Fechamento $a * b \in \mathbb{G}$

Associatividade $(a * b) * c = a * (b * c)$

Elemento neutro $\exists n \in \mathbb{G} ; a * n = n * a = a$

Elemento inverso $\exists a^{-1} \in \mathbb{G} ; a * a^{-1} = a^{-1} * a = n$

Exemplo 2.2.1: O conjunto dos números inteiros \mathbb{Z} é um grupo quanto à operação binária adição (+). Além das propriedade de fechamento e de associatividade, o elemento neutro é o $0 \in \mathbb{Z}$, e o inverso de $k \in \mathbb{Z}$ é $-k \in \mathbb{Z}$.

Exemplo 2.2.2: O conjunto $\{1\}$ é o menor grupo quanto à operação binária multiplicação (\cdot), pois o elemento 1 é neutro na multiplicação e inverso multiplicativo de si mesmo.

Antes de apresentar os grupos matriciais utilizados nesse trabalho, faz-se necessário apresentar a seguinte definição:

Definição 2.42: Define-se como $M_n(\mathbb{R})$ o conjunto das matrizes reais quadradas de ordem n .

Quanto à adição, o conjunto $M_n(\mathbb{R})$ é um grupo, com elemento neutro 0_n e, como elemento inverso de uma matriz $A_n \in M_n(\mathbb{R})$, a matriz $-A_n \in M_n(\mathbb{R})$. No entanto, quanto à multiplicação, $M_n(\mathbb{R})$ não é um grupo, pois não são todas as matrizes de ordem n que possuem inverso multiplicativo.

Definição 2.43: O subconjunto de $M_n(\mathbb{R})$ com matrizes invertíveis de ordem n , dotado da operação multiplicação, é definido como grupo linear real, denotado $GL_n(\mathbb{R})$. A representação simbólica é:

$$GL_n(\mathbb{R}) = \{A \in M_n; \det(A) \neq 0\}$$

Exemplo 2.2.3: A matriz $A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix} \in M_3$ também pertence a GL_3 , pois $\det(A) = -18 \neq 0$.

Definição 2.44: Define-se como grupo ortogonal de dimensão n o subgrupo de GL_n em que as matrizes transpostas são iguais às inversas de cada elemento. A representação simbólica é:

$$O(n) = \{Q_n \in GL_n(\mathbb{R}); Q_n^{-1} = Q_n^T\}$$

Proposição 2.45: Seja $Q \in O(n)$. Têm-se, então, que $\det(Q) = \pm 1$.

Demonstração. Pela definição, tem-se que $Q \cdot Q^T = I$. Dessa forma, se d é o determinante de Q , então, pelo Teorema de Binet 2.20 e pela Proposição 2.11, conclui-se que:

$$\begin{aligned} \det(Q \cdot Q^T) &= \det(I) \\ \det(Q) \det(Q^T) &= \det(I) \\ d \cdot d &= 1 \\ d &= \sqrt{1} \\ d &= \pm 1 \end{aligned} \quad \square$$

Os grupos matriciais utilizados no desenvolvimento desse trabalho são o Grupo Ortogonal Especial e o Grupo Euclidiano Especial, apresentados nas Subseções 2.2.1 e 2.2.2, respectivamente. As propriedades de ambos os grupos se dá quanto à operação de multiplicação.

2.2.1 Grupo Ortogonal Especial

Definição 2.46: Também conhecido como grupo das matrizes rotacionais, o grupo ortogonal especial é o subgrupo de $O(n)$ com matrizes de determinante 1. O grupo é denotado como $SO(n)$ e, por símbolos, tem-se que:

$$SO(n) = \{R_n \in GL_n(\mathbb{R}); R_n^{-1} = R_n^T \text{ e } \det(R_n) = 1\}$$

No espaço vetorial V^2 têm-se que $R \in SO(2)$ é da forma:

$$R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}$$

onde tem-se que os vetores unitários $\vec{x} = (r_{11}, r_{21})$ e $\vec{y} = (r_{12}, r_{22})$, com a relação $\hat{x} \perp \hat{y}$, formam uma base ortonormal de V^2 .

No espaço vetorial V^3 , por sua vez, têm-se que $R \in SO(3)$ é da forma:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

onde o terno de vetores $\vec{x} = (r_{11}, r_{21}, r_{31})$, $\vec{y} = (r_{12}, r_{22}, r_{32})$ e $\vec{z} = (r_{13}, r_{23}, r_{33})$ formam uma base ortonormal de V^3 .

As propriedades de grupo de $SO(n)$, ou seja, fechamento, associatividade, elemento neutro e elemento inverso, são demonstradas nas Proposições 2.47, 2.48, 2.49 e 2.50 a seguir, respectivamente.

Proposição 2.47 (Fechamento em $SO(n)$): Se $R_{(1)} \in SO(n)$ e $R_{(2)} \in SO(n)$, então $R_{(1)}R_{(2)} \in SO(n)$.

Demonstração. Ao multiplicar $R_{(1)}R_{(2)}$ pela sua transposta, tem-se que:

$$\begin{aligned} (R_{(1)}R_{(2)}) (R_{(1)}R_{(2)})^T &\stackrel{(a)}{=} (R_{(1)}R_{(2)}) (R_{(2)}^T R_{(1)}^T) \\ &\stackrel{(b)}{=} R_{(1)} (R_{(2)}R_{(2)}^T) R_{(1)}^T \\ &\stackrel{(c)}{=} R_{(1)}R_{(1)}^T \\ &= I_n \end{aligned}$$

A igualdade (a) vem da Proposição 2.7, a (b) vem da associatividade da multiplicação de matrizes e a igualdade (c) vem do fato da transposta ser igual à inversa em $SO(n)$, logo $R_{(2)}R_{(2)}^T = I_n$, que é elemento neutro na multiplicação.

Como a multiplicação pela transposta resultou em uma matriz identidade, pode-se concluir que a matriz $R_{(1)}R_{(2)}$ é ortogonal, pois $(R_{(1)}R_{(2)})^T = (R_{(1)}R_{(2)})^{-1}$. Uma vez que $\det(R_{(1)}) = \det(R_{(2)}) = 1$, pelo Teorema de Binet 2.20, tem-se que $\det(R_{(1)}R_{(2)}) = 1$. Dessa forma, conclui-se que $R_{(1)}R_{(2)} \in SO(n)$. \square

Proposição 2.48 (Associatividade de $SO(n)$): Se $R_{(1)}, R_{(2)}, R_{(3)} \in SO(n)$, então $(R_{(1)}R_{(2)})R_{(3)} = R_{(1)}(R_{(2)}R_{(3)})$.

Demonstração. O resultado é obtido aplicando a propriedade de associatividade da multiplicação de matrizes. □

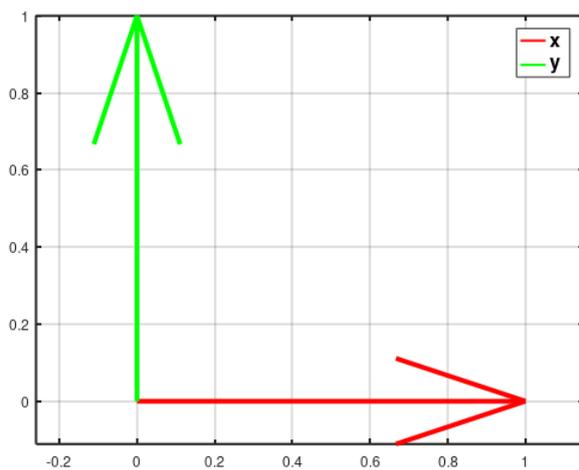
Proposição 2.49 (Elemento Neutro de $SO(n)$): Se $R \in SO(n)$, então existe o elemento $I_n \in SO(n)$ tal que $R \cdot I_n = I_n \cdot R = R$.

Demonstração. O produto segue da propriedade de elemento neutro da multiplicação de matrizes. Basta, então, verificar se $I_n \in SO(n)$, o que é verdade, pois $\det(I_n) = 1$ e $I_n^{-1} = I_n^T$. □

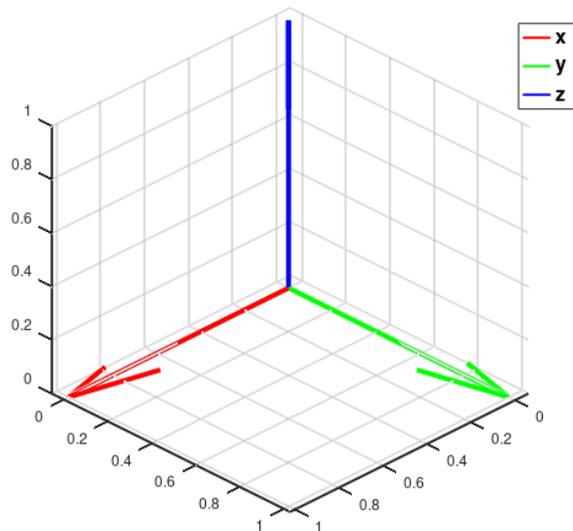
Por definição, as matrizes pertencentes a $SO(2)$ e $SO(3)$ representam bases ortonormais de V^2 e V^3 , respectivamente. Logo essas matrizes podem representar geometricamente o sistema de coordenadas, uma vez que tal representação depende de uma base, conforme a Definição 2.29. Por exemplo, as matrizes I_2 e I_3 , que representam as bases canônicas do plano e do espaço, nessa ordem, podem ser representadas conforme exposto na Figura 2.5.

Figura 2.5: Representação das matrizes identidade

(a) I_2 no plano.



(b) I_3 no espaço.



Fonte: Elaborado pelo autor.

As matrizes diferentes de I_2 e I_3 , mas que também pertencem a $SO(2)$ e $SO(3)$, possuem uma representação similar no plano e no espaço, respectivamente, mantendo as ortogonalidades. Porém no plano os eixos não coincidem com os representados por I_2 , e, no espaço, dois ou três eixos não coincidem com os representados por I_3 .

Proposição 2.50 (Elemento Inverso de $SO(n)$): Se $R \in SO(n)$, então existe $R^{-1} \in SO(n)$, tal que $R \cdot R^{-1} = I_n$.

Demonstração. Como $\det(R) = 1$, então a matriz possui inverso multiplicativo. O produto segue, então, da propriedade de elemento inverso da multiplicação de matrizes. Basta, então, verificar se $R^{-1} \in SO(n)$. De acordo com a Equação 2.4, tem-se que $\det(R^{-1}) = \frac{1}{\det(R)} = \frac{1}{1} = 1$, e, como $R^{-1} \cdot (R^{-1})^{-1} = R^{-1} \cdot R = I$, conclui-se que $R^{-1} \in SO(n)$. \square

Proposição 2.51: Toda matriz $R \in SO(n)$ descreve uma mudança de base em V^n .

Demonstração. Seja a base $\mathcal{B} = \{\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n\}$ representada pela matriz $R_{(1)} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_n \end{bmatrix} \in SO(n)$, no qual cada \mathbf{e}_i representa o vetor coluna correspondente ao i -ésimo vetor da base, com $1 \leq i \leq n$. De acordo com a Proposição 2.47, o produto $RR_{(1)}$ também pertence a $SO(n)$, logo descreve uma base ortonormal de V^n . Escrevendo da forma $RR_{(1)} = \begin{bmatrix} \mathbf{e}'_1 & \mathbf{e}'_2 & \dots & \mathbf{e}'_n \end{bmatrix}$, tem-se o seguinte resultado:

$$\begin{bmatrix} \mathbf{e}'_1 & \mathbf{e}'_2 & \dots & \mathbf{e}'_n \end{bmatrix} = R \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_n \end{bmatrix}$$

Pela Definição 2.37, é possível afirmar, então, que a multiplicação à esquerda por R significa uma transformação linear de mudança de base. Caso $\mathbf{e}'_i = \mathbf{e}_i \forall i \in \{1, 2, \dots, n\}$, então $R = I_n$. \square

Corolário 2.52: Sejam as matrizes $R_{(1)}, R_{(2)} \in SO(n)$. Existe uma matriz $R \in SO(n)$, tal que $R_{(1)} = RR_{(2)}$.

Demonstração. O resultado é obtido quando $R = R_{(1)}R_{(2)}^{-1}$. \square

Os resultados da Proposição 2.51 e do Corolário 2.52 afirmam não somente que uma matriz $R \in SO(n)$ representa uma transformação linear de mudança de base, mas que toda mudança de base possui uma matriz representante pertencente a este grupo. Caso $R = I$, é realizada uma transformação identidade, a qual todo vetor permanece inalterado.

Quando a notação ${}_A R \in SO(n)$ é utilizada, significa que a matriz representa uma transformação do sistema de coordenadas de referência, cuja base é a matriz identidade, para o sistema de coordenadas A . A notação ${}_B^A R \in SO(n)$, por sua vez, significa a transformação do sistema de coordenadas A para o sistema de coordenadas B .

A implicação geométrica da mudança de base ocorrida ao se multiplicar um vetor à esquerda por uma matriz pertencente a $SO(n)$ é verificada no resultado a seguir:

Proposição 2.53: Seja um vetor $\mathbf{u} = \mathbf{R} \cdot \mathbf{v}$, onde $\mathbf{u}, \mathbf{v} \in V^n$ e $\mathbf{R} \in SO(n)$. É válido que $\|\mathbf{u}\| = \|\mathbf{v}\|$.

Demonstração. Como $\|\mathbf{u}\| \geq 0$, fazendo uso da Equação 2.7, tem-se que:

$$\|\mathbf{u}\|^2 = \mathbf{u}^T \cdot \mathbf{u} = (\mathbf{R} \cdot \mathbf{v})^T \cdot (\mathbf{R} \cdot \mathbf{v}) = \mathbf{v}^T \cdot \mathbf{R}^T \cdot \mathbf{R} \cdot \mathbf{v} = \mathbf{v}^T \cdot \mathbf{v} = \|\mathbf{v}\|^2 \quad \square$$

O resultado apresentado pela Proposição 2.53 afirma que a multiplicação à esquerda por ${}^A_B\mathbf{R}$ não altera a norma do vetor. A representação geométrica dessa operação consiste na alteração do sistema de coordenadas do vetor, que muda do sistema A para o B. Como $\mathbf{R}0_{n \times 1} = 0_{n \times 1}$ e todo vetor de V^n possui a origem como ponto inicial, a operação de multiplicar à esquerda por ${}^A_B\mathbf{R}$ representa a rotação do vetor, mantendo a origem inalterada.

No plano, a matriz $\mathbf{R} \in SO(2)$ representa a transformação geométrica $\text{Rot}(\alpha)$, sendo α o ângulo de rotação em torno da origem. Sendo assim, a matriz \mathbf{R} é da seguinte forma:

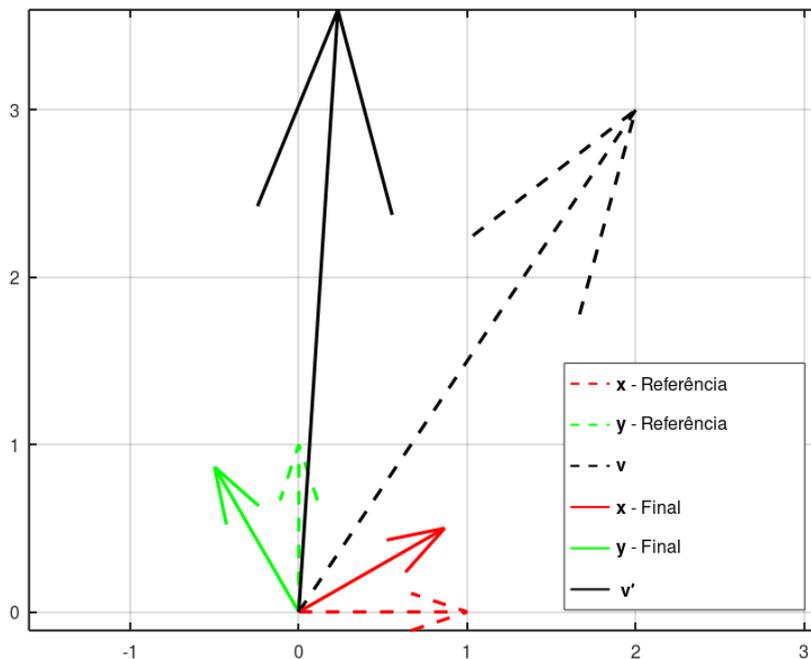
$$\mathbf{R} = \begin{bmatrix} \cos \alpha & -\text{sen } \alpha \\ \text{sen } \alpha & \cos \alpha \end{bmatrix} \quad (2.9)$$

Exemplo 2.2.4: Seja o vetor $\mathbf{v} = [2, 3]^T$. Ao se multiplicar este vetor à esquerda pela matriz $\mathbf{R} = \begin{bmatrix} \cos 30^\circ & -\text{sen } 30^\circ \\ \text{sen } 30^\circ & \cos 30^\circ \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \in SO(2)$ é obtido o vetor $\mathbf{v}' = \mathbf{R}\mathbf{v} = \left[\sqrt{3} - \frac{3}{2}, 1 - 3\frac{\sqrt{3}}{2} \right]^T \approx [0,232 \ 3,598]^T$. A representação geométrica dessa multiplicação, demonstrada pela Figura 2.6, apresenta uma rotação do vetor em 30° em torno da origem, isto é, $\text{Rot}(30^\circ)$, segundo a notação de transformação geométrica no plano.

Para verificar o significado da matriz $\mathbf{R} \in SO(3)$ como uma rotação no espaço, faz-se necessário verificar os resultados a seguir.

Proposição 2.54: O módulo dos autovalores de uma transformação linear descrita por uma matriz $\mathbf{R} \in SO(n)$ é 1.

Figura 2.6: Rotação de 30° no plano



Fonte: Elaborado pelo autor.

Demonstração. A Definição 2.35, quanto a autovalores e autovetores, afirma que:

$$R\mathbf{v} = \lambda\mathbf{v}$$

Considerando a norma dos vetores, tem-se que:

$$\begin{aligned} \|R\mathbf{v}\|^2 &= \|\lambda\mathbf{v}\|^2 \\ \|R\mathbf{v}\|^2 &= |\lambda|^2\|\mathbf{v}\|^2 \end{aligned}$$

Uma vez que, pela Proposição 2.53, $\|R\mathbf{v}\|^2 = \|\mathbf{v}\|^2$, segue que:

$$\|\mathbf{v}\|^2 = |\lambda|^2\|\mathbf{v}\|^2 \Rightarrow |\lambda|^2 = 1$$

Como o módulo é sempre não-negativo, conclui-se que $|\lambda| = 1$. □

Teorema 2.55 (Teorema de Rotação de Euler): Qualquer rotação no espaço que mantém constante um ponto, também mantém constante um eixo. Em outras palavras, se $R \in \text{SO}(3)$, então existe um vetor não nulo \mathbf{v} , tal que $R\mathbf{v} = \mathbf{v}$.

Demonstração. Utilizando a definição de autovetor e autovalor, \mathbf{v} é o autovetor

associado ao autovalor 1. Logo, é necessário provar a existência deste autovalor unitário.

A equação $R\mathbf{v} = \lambda\mathbf{v}$, que define autovetores e autovalores, pode ser escrita da forma:

$$R\mathbf{v} = \lambda I\mathbf{v} \Rightarrow (R - \lambda I)\mathbf{v} = \mathbf{0}_{3 \times 1}$$

Essa última equação somente tem solução não-trivial, isto é, com \mathbf{v} não-nulo, se o polinômio característico $p(\lambda) = \det(R - \lambda I)$ possuir raízes reais.

Como o polinômio é de terceiro grau, ele possui ao menos uma raiz real. Levando em conta que $\det(R) = 1$, e as Proposições 2.36 e 2.54, tem-se os seguintes casos:

1. O polinômio possui três raízes reais $\lambda_1, \lambda_2, \lambda_3$. Como as raízes possuem módulo unitário e o produto das raízes é igual a 1, ou as raízes são $(-1, -1, 1)$ ou são $(1, 1, 1)$;
2. O polinômio possui uma raiz real λ_1 e duas raízes complexas conjugadas λ_2 e λ_2^* . Como $\lambda_2 \cdot \lambda_2^* = \lambda_2^2 = 1$, tem-se que $\lambda_1 = 1$.

Em ambos os casos existe a raiz 1. Sendo assim, como R possui um autovalor unitário, existe um autovetor \mathbf{v} tal que $R\mathbf{v} = \mathbf{v}$. □

Geometricamente, o vetor pertencente a V^3 que permanece inalterado ao ser multiplicado à esquerda por uma matriz de rotação $R \in SE(3)$, conforme demonstrado pelo Teorema de Euler 2.55, é o eixo da rotação no espaço. O sentido do vetor define o sentido da rotação, conforme a regra da mão direita apresentada na Figura 2.2b.

As rotações no espaço que são mais relevantes para o desenvolvimento desse trabalho são as que acontecem em torno dos vetores que representam os eixos \vec{x} , \vec{y} e \vec{z} do sistema de coordenadas. As matrizes ${}_A R = \text{Rot}(\vec{x}, \alpha)$, ${}_B R = \text{Rot}(\vec{y}, \beta)$ e ${}_C R = \text{Rot}(\vec{z}, \gamma)$, com ${}_A R, {}_B R, {}_C R \in SO(3)$, são representadas da seguinte forma, respectivamente:

$${}_A R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\text{sen } \alpha \\ 0 & \text{sen } \alpha & \cos \alpha \end{bmatrix}$$

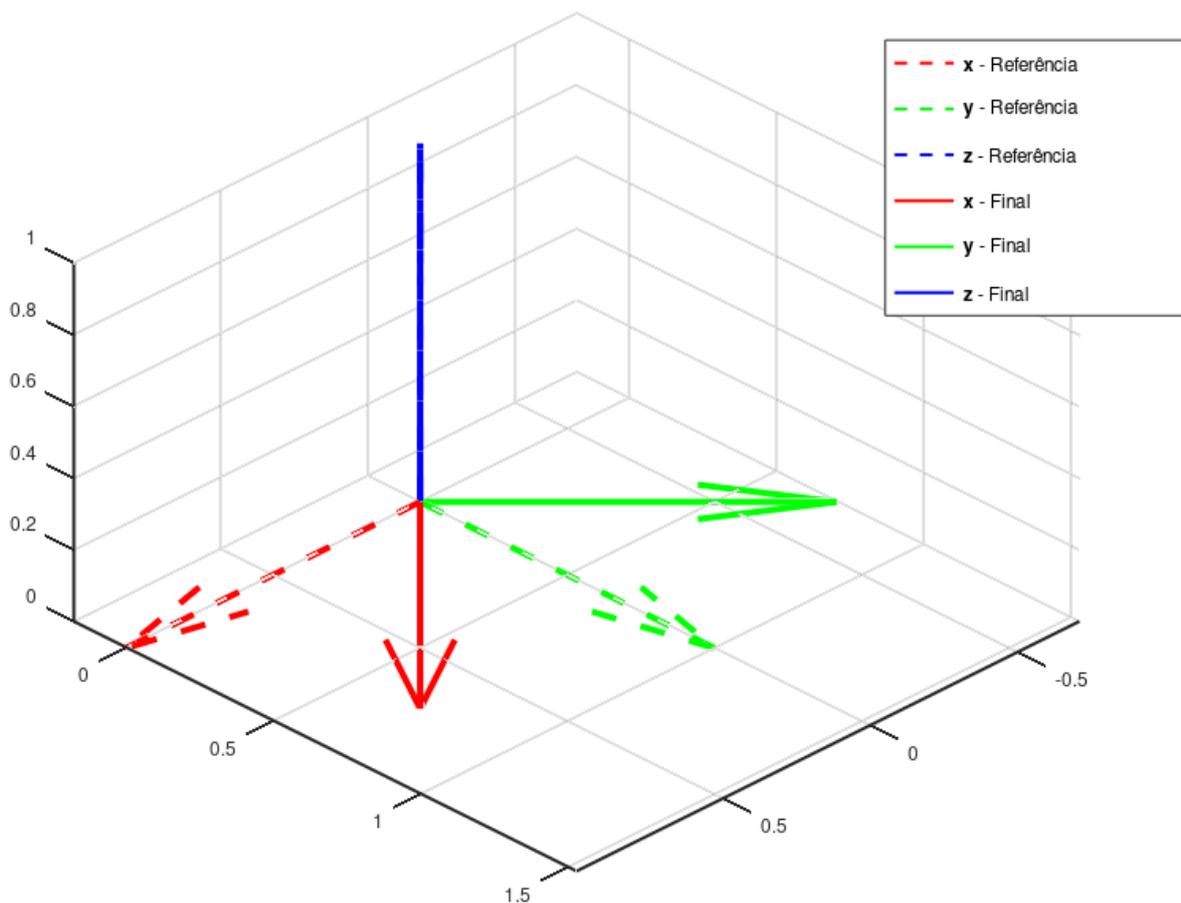
$${}_B\mathbf{R} = \begin{bmatrix} \cos \beta & 0 & \text{sen } \beta \\ 0 & 1 & 0 \\ -\text{sen } \beta & 0 & \cos \beta \end{bmatrix}$$

$${}_C\mathbf{R} = \begin{bmatrix} \cos \gamma & -\text{sen } \gamma & 0 \\ \text{sen } \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Exemplo 2.2.5: A rotação de 45° em torno do eixo \vec{z} , representado como $\text{Rot}(\vec{z}, 45^\circ)$, exemplificado pela Figura 2.7, é dado pela seguinte matriz:

$${}_A\mathbf{R} = \begin{bmatrix} \cos 45^\circ & -\text{sen } 45^\circ & 0 \\ \text{sen } 45^\circ & \cos 45^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \approx \begin{bmatrix} 0,707 & -0,707 & 0 \\ 0,707 & 0,707 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Figura 2.7: Representação de $\text{Rot}(\vec{z}, 45^\circ)$



Fonte: Elaborado pelo autor.

2.2.2 Grupo Euclidiano Especial

Definição 2.56: Denotado por $SE(n)$, o grupo euclidiano especial é aquele que contém matrizes que descrevem as isometrias diretas de um espaço n -dimensional.

A representação se dá por matrizes de dimensões $(n + 1) \times (n + 1)$, chamadas matrizes de transformação homogênea, que são da forma:

$$T = \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times n} & 1 \end{bmatrix}$$

onde $R \in SO(n)$ e $\mathbf{p} \in V^n$.

Proposição 2.57: Todas as isometrias diretas do plano e do espaço podem ser descritas por uma matriz pertencente a $SE(2)$ e $SE(3)$, respectivamente.

Demonstração. Escrevendo o vetor $\vec{v} = (v_1, v_2) \in V^2$ como $\mathbf{v}' = [\mathbf{v}^T \ 1]^T = [v_1 \ v_2 \ 1]^T$, ao se multiplicar $T \in SE(2)$ por \mathbf{v}' , obtém-se:

$$T \cdot \mathbf{v}' = \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times n} & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} = \begin{bmatrix} R \cdot \mathbf{v} + \mathbf{p} \\ 1 \end{bmatrix}$$

O vetor $[R \cdot \mathbf{v} + \mathbf{p}]^T$ é o vetor \mathbf{v} após uma transformação de translação descrita por \mathbf{p} e uma rotação descrita por R . Se a transformação for de somente translação, tem-se que $R = I_2$, e se for somente de rotação, tem-se que $\mathbf{p} = [0 \ 0]^T$.

A demonstração para o espaço, com $T \in SE(3)$, é análoga. \square

Dessa forma, a transformação geométrica $\Omega : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, que representa isometrias diretas no espaço, é da forma:

$$[\Omega(\mathbf{v})] = T\mathbf{v}, \tag{2.10}$$

onde $T \in SE(3)$, e os vetores coluna $[\Omega(\mathbf{v})]$ e \mathbf{v} possuem quatro elementos, sendo os três primeiros as coordenadas de \mathbb{R}^3 e o quarto, o valor 1, ou seja, são da forma $[v_1, v_2, v_3, 1]^T$.

Como, segundo a Definição 2.29, um sistema de referência pode ser representado por um par composto por um ponto e uma base ortonormal, isto é, (O, \mathcal{B}) , ao se aplicar uma translação ao ponto O e uma rotação, que representa uma mudança de base em \mathcal{B} , é realizada, então, uma mudança do sistema de referência.

As quatro proposições a seguir provam as propriedades de grupo de $SE(n)$.

Proposição 2.58 (Fechamento em $SE(n)$): Se ${}_A T \in SE(n)$ e ${}_B T \in SE(n)$ então ${}_A T {}_B T \in SE(n)$.

Demonstração. Ao se multiplicar as matrizes, obtém-se:

$$\begin{aligned} {}_A T {}_B T &= \begin{bmatrix} {}_A R & \mathbf{p}_A \\ 0_{1 \times n} & 1_1 \end{bmatrix} \cdot \begin{bmatrix} {}_B R & \mathbf{p}_B \\ 0_{1 \times n} & 1_1 \end{bmatrix} \\ &= \begin{bmatrix} {}_A R \cdot {}_B R + \mathbf{p}_A \cdot 0_{1 \times n} & {}_A R \cdot \mathbf{p}_B + \mathbf{p}_A \cdot 1_1 \\ 0_{1 \times n} \cdot {}_B R + 1_1 \cdot 0_{1 \times n} & 0_{1 \times n} \cdot \mathbf{p}_B + 1_1 \cdot 1_1 \end{bmatrix} \\ &= \begin{bmatrix} {}_A R {}_B R & {}_A R \mathbf{p}_B + \mathbf{p}_A \\ 0_{1 \times n} & 1_1 \end{bmatrix} \end{aligned}$$

Conforme a Proposição 2.47, a submatriz ${}_A R {}_B R \in SO(n)$. Como a submatriz ${}_A R \mathbf{p}_B + \mathbf{p}_A$ tem dimensões $n \times 1$, pode-se concluir que ${}_A T {}_B T \in SE(n)$. \square

Proposição 2.59 (Associatividade de $SE(n)$): Se ${}_A T \in SE(n)$, ${}_B T \in SE(n)$ e ${}_C T \in SE(n)$, então $({}_A T {}_B T) {}_C T = {}_A T ({}_B T {}_C T)$.

Demonstração. O resultado é obtido aplicando a propriedade de associatividade da multiplicação de matrizes. \square

Proposição 2.60 (Elemento Neutro de $SE(n)$): Se $T \in SE(n)$, então existe o elemento $I_{n+1} \in SE(n)$ tal que $T \cdot I_{n+1} = I_{n+1} \cdot T = T$.

Demonstração. Como a matriz $T \in SE(n)$ possui ordem $n + 1$, o produto segue da propriedade de elemento neutro da multiplicação de matrizes. Basta, então, verificar se $I_{n+1} \in SE(n)$, o que é verdade, pois $I_{n+1} = \begin{bmatrix} I_n & 0_{n \times 1} \\ 0_{1 \times n} & 1_1 \end{bmatrix}$, e, como $I_n \in SO(n)$ conforme a Proposição 2.49, tem-se que $I_{n+1} \in SE(n)$. \square

Em $SE(3)$ a matriz I_4 significa o sistema de referência dos eixos das coordenadas reais \vec{x} , \vec{y} e \vec{z} . Significa, também, aplicar nenhuma translação e nenhuma rotação ao sistema de coordenadas. A Figura 2.5 representa I_4 em $SE(3)$.

Proposição 2.61 (Elemento Inverso de $SE(n)$): Se $T = \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times n} & 1_1 \end{bmatrix} \in SE(n)$, existe $T^{-1} \in SE(n)$, tal que $T \cdot T^{-1} = I_{n+1}$.

Demonstração. O produto segue da propriedade de elemento inverso da multiplicação de matrizes. Basta, então, verificar se $T^{-1} \in SE(n)$. Suponha a existência de tal matriz da forma $T^{-1} = \begin{bmatrix} R' & \mathbf{p}' \\ 0_{1 \times n} & 1_1 \end{bmatrix} \in SE(n)$. Sendo assim, tem-se que:

$$\begin{aligned} TT^{-1} &= \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times n} & 1_1 \end{bmatrix} \cdot \begin{bmatrix} R' & \mathbf{p}' \\ 0_{1 \times n} & 1_1 \end{bmatrix} \\ &= \begin{bmatrix} R \cdot R' + \mathbf{p} \cdot 0_{1 \times n} & R \cdot \mathbf{p}' + \mathbf{p} \cdot 1_1 \\ 0_{1 \times n} \cdot R' + \mathbf{p} \cdot 0_{1 \times n} & 0_{1 \times n} \cdot \mathbf{p}' + 1_1 \cdot 1_1 \end{bmatrix} \\ &= \begin{bmatrix} R \cdot R' & R \cdot \mathbf{p}' + \mathbf{p} \cdot 1_1 \\ 0_{1 \times n} & 1_1 \end{bmatrix} \end{aligned} \quad (2.11)$$

Dessa forma, para que $R \cdot R' = I_n$, tem-se que $R' = R^T$, conforme a Proposição 2.50. Para que $R \cdot \mathbf{p}' + \mathbf{p} \cdot 1_1 = 0_{n \times 1}$, tem-se que $R \cdot \mathbf{p}' = -\mathbf{p} \Rightarrow \mathbf{p}' = -R^T \mathbf{p}$.

Por fim, conclui-se que, para a matriz $T = \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times n} & 1_1 \end{bmatrix}$, a sua inversa é

$$T^{-1} = \begin{bmatrix} R^T & -R^T \mathbf{p} \\ 0_{1 \times n} & 1_1 \end{bmatrix}. \quad \square$$

Proposição 2.62: Sejam as matrizes $T_{(1)} = \begin{bmatrix} I_n & \mathbf{p}_{(1)} \\ 0_{1 \times n} & 1_1 \end{bmatrix}$ e $T_{(2)} = \begin{bmatrix} I_n & \mathbf{p}_{(2)} \\ 0_{1 \times n} & 1_1 \end{bmatrix}$, ambas pertencentes a $SE(n)$, as quais representam as translações $\text{Trans}(\mathbf{p}_{(1)})$ e $\text{Trans}(\mathbf{p}_{(2)})$, respectivamente. A matriz $T = T_{(1)} \cdot T_{(2)}$ equivale à translação $\text{Trans}(\mathbf{p}_{(1)} + \mathbf{p}_{(2)})$.

Proposição 2.63: A matriz $T = \begin{bmatrix} R & \mathbf{p} \\ 0_{1 \times n} & 1_1 \end{bmatrix} \in SE(n)$, com $R \neq I_n$, $n = \{2, 3\}$ e o vetor \mathbf{p} não nulo, tem-se que T equivale à aplicação de uma translação e uma rotação, nessa ordem.

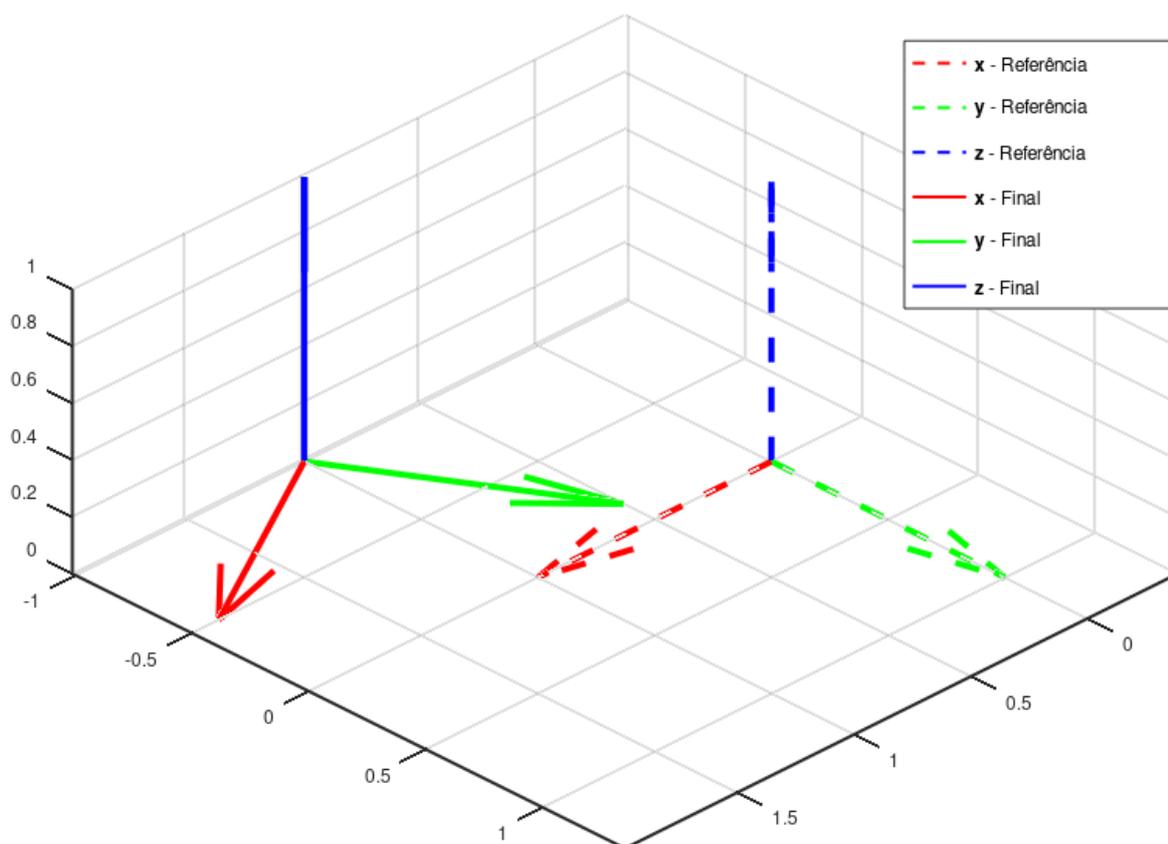
Demonstração. A prova se dá ao multiplicar as matrizes equivalentes às transformações geométricas, pois $\begin{bmatrix} I & \mathbf{p} \\ 0_{1 \times n} & 1_1 \end{bmatrix} \cdot \begin{bmatrix} R & 0_{n \times 1} \\ 0_{1 \times n} & 1_1 \end{bmatrix} = T$. \square

Exemplo 2.2.6: A matriz ${}_A T \in SE(3)$ que equivale às transformações

Trans (\vec{p}) Rot ($\vec{z}, 30^\circ$), onde $\vec{p} = (1, -1, 0)$, exemplificada pela Figura 2.8, é dada por:

$${}^A T = \begin{bmatrix} \cos 30^\circ & -\text{sen } 30^\circ & 0 & 1 \\ \text{sen } 30^\circ & \cos 30^\circ & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 2.8: Representação de Trans (\vec{p}) Rot ($\vec{z}, 30^\circ$) em SE (3)



Fonte: Elaborado pelo autor.

3 Desenvolvimento

Nesse capítulo são utilizados os conteúdos apresentados na fundamentação teórica em atividades práticas, sendo a primeira a de posicionamento de objetos rígidos no plano e no espaço, e a segunda, a descrição de um braço robótico no espaço, por meio de uma abordagem computacional.

3.1 Posicionamento de um Objeto Rígido

O problema abordado nessa seção consiste em especificar quantas e quais são as informações necessárias para descrever o posicionamento de um objeto tanto no plano quanto no espaço.

Como exemplo, para a análise no plano é considerado um triângulo escaleno, onde os lados possuem dimensões inalteráveis a , b e c , opostos aos vértices A , B e C , respectivamente. Para a realização da análise no espaço é utilizado um prisma, cuja face da base seja congruente ao triângulo analisado no plano, e a altura desprezível.

Para que seja possível ter referencial no posicionamento, este foi realizado no plano e no espaço cartesianos, isto é, em \mathbb{R}^2 e \mathbb{R}^3 , respectivamente.

3.1.1 Posicionamento no Plano

O posicionamento do objeto no plano se inicia com a escolha livre de um ponto no plano cartesiano para posicionar o ponto A . Essa escolha envolve duas variáveis, que representam as coordenadas do ponto A no plano OXY de referência. Essas variáveis serão representadas como x_A e y_A .

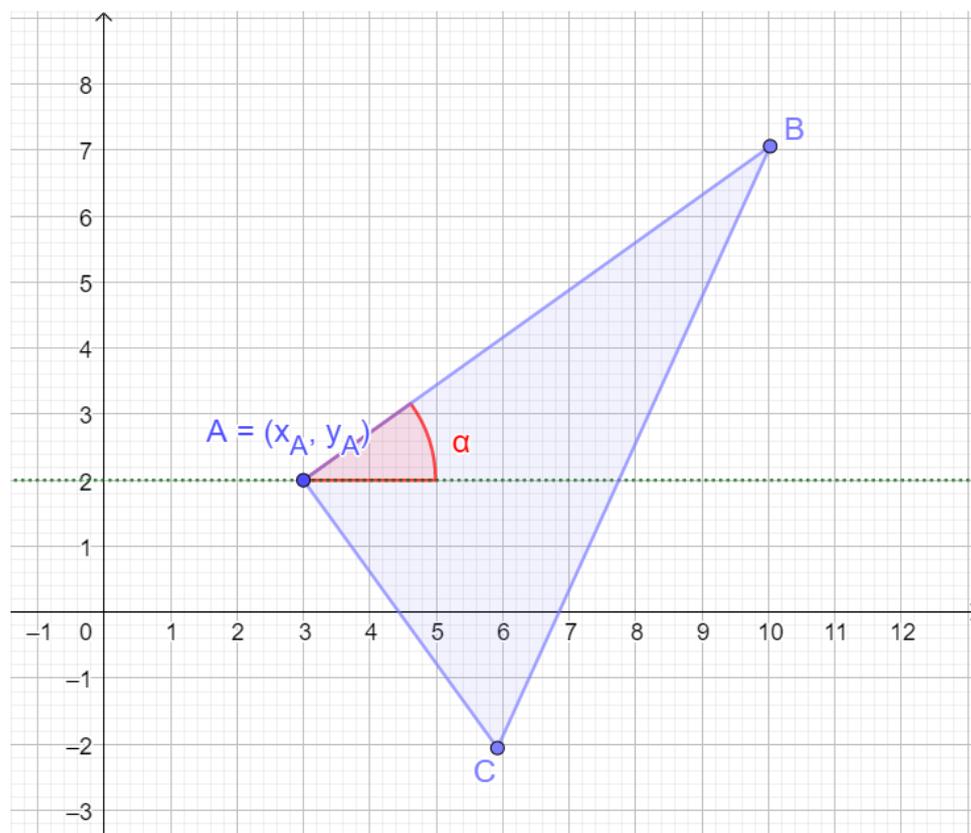
Uma vez definido o ponto A , é posicionado o ponto B do objeto. É fácil verificar que a escolha da posição do ponto B não é livre, isto é, não se pode tomar quaisquer coordenadas x_B e y_B , pois é necessário satisfazer a restrição $d(A,B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} = c$.

Dessa maneira, para que a restrição seja facilmente cumprida, é utilizada uma variável α , que representa o ângulo compreendido entre a reta r , paralela ao eixo OX que

passa por A , e a semirreta \overrightarrow{AB} .

Uma vez fixados os pontos A e B , o ponto C possui somente um local para ser posicionado, obedecendo às restrições $d(C,A) = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2} = b$ e $d(C,B) = \sqrt{(x_B - x_C)^2 + (y_B - y_C)^2} = a$. Para o posicionamento do objeto no plano não é considerada a opção de virá-lo em torno do eixo AB , uma vez que exigiria um movimento em uma terceira dimensão. A Figura 3.1 exemplifica o posicionamento final do triângulo no plano.

Figura 3.1: Posicionamento do triângulo no plano



Fonte: Elaborado pelo autor.

Conclui-se, então, que bastam três variáveis para representar o posicionamento de um objeto no plano, a saber as duas coordenadas de um ponto qualquer do sólido e uma variável que representa um giro deste no plano.

Para utilizar a notação do grupo euclidiano especial $SE(2)$, conforme a Seção 2.2.2, tem-se que:

- O referencial do objeto é o ponto A , que passa a ser a origem do sistema de referência;
- O deslocamento inicial do ponto A no plano é dado pelo vetor $\vec{p} = (x_A, y_A)$;

- O giro do objeto em torno do ponto A é no valor do ângulo α ;

Dessa forma, a matriz de transformação homogênea $T \in SE(2)$ equivalente ao deslocamento é a seguinte:

$$T = \begin{bmatrix} \cos \alpha & -\sin \alpha & x_A \\ \sin \alpha & \cos \alpha & y_A \\ 0 & 0 & 1 \end{bmatrix}$$

3.1.2 Posicionamento no Espaço

Semelhantemente ao que foi feito no plano, ao iniciarmos o posicionamento do prisma no espaço, são fixados um de três pontos não-colineares pertencentes ao sólido. Tomando um ponto A aleatório, os dados necessários para seu posicionamento são as suas coordenadas no espaço (x_A, y_A, z_A) .

Após fixarmos o ponto A , deve-se posicionar um ponto B , pertencente ao sólido. Como a distância entre A e B é constante, há uma restrição na escolha do local de posicionamento do ponto B , que tem que estar contido na esfera com centro em A e com raio $d(A,B) = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2 + (z_B - z_A)^2} = c$.

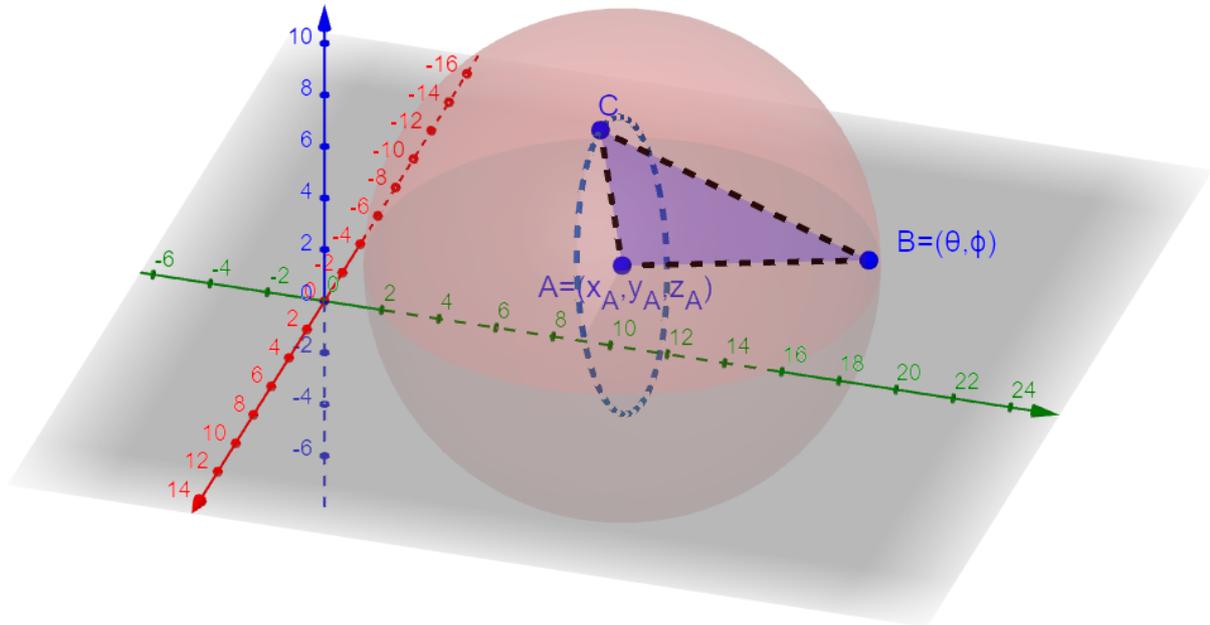
De forma a facilitar a representação, são utilizados dois ângulos, θ e ϕ , que representam as referentes coordenadas esféricas, conforme a Definição 2.32, do ponto B em relação aos vetores paralelos a \vec{x} , \vec{y} e \vec{z} que passam no ponto A . O comprimento do segmento AB é a coordenada ρ restante.

Uma vez fixado os pontos A e B , é posicionado, então, um ponto C , também contido no prisma, e não colinear a AB . É possível observar que o ponto C necessariamente tem que estar contido no círculo de raio $d(AB,C)$ e centro em C' , que é a interseção da reta perpendicular a AB que passa por C e a própria reta AB . Esse círculo é a interseção entre as esferas com centro em A e raio $d(A,C)$ e com centro em B e raio $d(B,C)$, uma vez que, pelo fato de A , B e C não serem colineares, $d(A,B) < d(A,C) + d(B,C)$. Para posicionar o ponto no círculo, basta que haja uma representação por meio de um ângulo γ , em relação a um plano qualquer que contenha AB .

Após fixar os pontos A , B e C , é realiza uma tentativa, então, de fixar um quarto ponto D , contido no sólido. No entanto, por haver três restrições, isto é, a distância do ponto D a qualquer um dos três pontos já fixados, resta apenas uma posição para o quarto ponto, e qualquer variável associada ao seu posicionamento seria redundante. Dessa forma,

pode-se afirmar que, para posicionar um objeto sólido no espaço, bastam se utilizar seis variáveis, sendo três coordenadas e três ângulos. A Figura 3.2 exemplifica o posicionamento final do prisma no espaço.

Figura 3.2: Posicionamento do prisma no espaço



Fonte: Elaborado pelo autor.

Utilizando a notação do grupo euclidiano especial $SE(3)$, conforme a Seção 2.2.2, tem-se que:

- A origem do sistema de referência do objeto no espaço é o ponto A ;
- O eixo \vec{y} do objeto, denotado \vec{y}_o é colinear a \overrightarrow{AB} e o eixo \vec{z} do objeto, denotado \vec{z}_o é colinear a \overrightarrow{AC} ;
- O deslocamento inicial do ponto A no espaço é dado pelo vetor $\vec{p} = (x_A, y_A, z_A)$;
- Se forem feitas as conversões $\alpha = 90^\circ - \phi$ e $\beta = 90^\circ - \theta$, os ângulos α , β e γ representarão rotações sucessivas em torno dos eixos \vec{x}_o , \vec{y}_o' e \vec{z}_o'' , respectivamente, onde \vec{y}_o' é o eixo \vec{y}_o após a primeira rotação em torno de \vec{x}_o , e \vec{z}_o'' é o eixo \vec{z}_o após as duas primeiras rotações sucessivas. Os ângulos α , β e γ , que descrevem as três rotações sucessivas, são denominados Ângulos de Euler[22].

Dessa forma, a matriz de transformação homogênea $T \in SE(3)$ equivalente ao deslocamento é a seguinte:

$$\begin{aligned}
T = & \begin{bmatrix} 1 & 0 & 0 & x_A \\ 0 & 1 & 0 & y_A \\ 0 & 0 & 1 & z_A \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha & -s\alpha & 0 \\ 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\beta & 0 & s\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\gamma & -s\gamma & 0 & 0 \\ s\gamma & c\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
& \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta & x_A \\ c\alpha s\gamma + c\gamma s\alpha s\beta & c\alpha c\gamma - s\alpha s\beta s\gamma & -c\beta s\alpha & y_A \\ s\alpha s\gamma - c\alpha c\gamma s\beta & c\gamma s\alpha + c\alpha s\beta s\gamma & c\alpha c\beta & z_A \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

em que $c\alpha = \cos\alpha$, $s\alpha = \sin\alpha$ etc.

3.1.3 Graus de Liberdade

Graus de liberdade, na Mecânica, é o de conjunto de variáveis que representam a posição e o deslocamento de um objeto rígido ou de um sistema em um espaço n -dimensional.

Tomando como referência o posicionamento de objetos rígidos realizado nas seções anteriores, é válido que:

- No plano são necessárias $2 + 1 = 3$ variáveis para representar a posição do objeto, totalizando 3 graus de liberdade;
- No espaço tridimensional são necessárias $3 + 2 + 1 = 6$ variáveis para posicionar o objeto, totalizando 6 graus de liberdade;

3.2 Simulação do braço robótico

Nessa seção são apresentados os conceitos relativos a braços robóticos, a notação de Denavit-Hartenberg e o processo de simulação por meio de ambiente computacional.

O objetivo principal é comparar a representação do efetuador de um braço robótico por meio da notação de Denavit-Hartenberg, ou seja, por um produto de matrizes de $SE(3)$, com a posição e orientação do efetuador de um braço real. A representação dada é por meio do ambiente computacional da linguagem de programação Octave e a verificação da posição do braço, por meio do simulador de robótica CoppeliaSim, ambos apresentados nas próximas subseções.

3.2.1 Definição de braço robótico

O termo robô foi originado da palavra tcheca *robota*, que significa trabalho escravo, e surgiu pela primeira vez na obra de ficção “R.U.R.”, do escritor Karel Čapek, em 1920. No enredo, humanoides são criados a partir de matéria orgânica, a fim de exercerem atividades serviçais a humanos.

O autor que ganhou mais fama com o tema robótica, a partir do ano 1950, foi Isaac Asimov, com os livros “Eu, Robô”, “O Homem Bicentenário”, os livros da série Robôs entre outros.

Dentre alguns robôs conhecidos na ficção, pode-se destacar a empregada doméstica Rosie, do desenho animado Os Jetsons, os droides C-3PO e R2-D2, da saga Star Wars, o Bender, da série Futurama, Wall-E, do filme homônimo, entre outros.

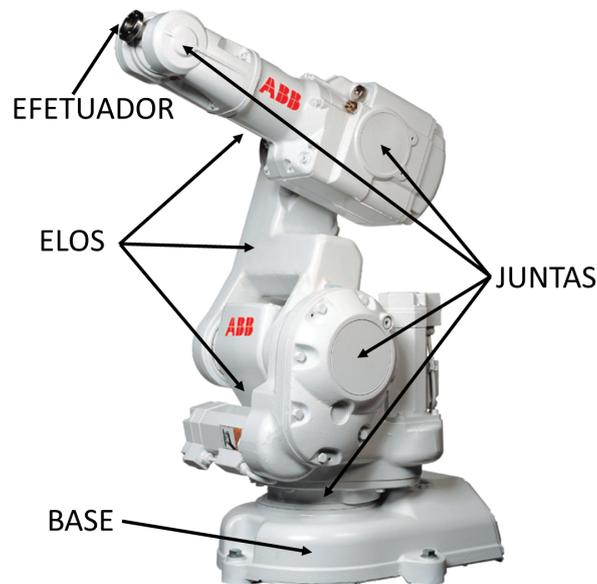
Na vida real, no entanto, o conceito de robôs não é unânime. Enquanto alguns consideram robô qualquer equipamento automatizado que realize uma tarefa, o que incluiria, por exemplo, as máquinas de lavar roupas, o conceito mais aplicado se restringe aos equipamentos que podem ser reprogramados, de forma a executar funções que não foram pré-programadas em sua construção.

Por mais que haja robôs no ambiente doméstico, como aspiradores de pó automáticos, segundo Mihelj et al.[23], os sistemas robóticos mais úteis e eficientes, atualmente, são os manipuladores robóticos industriais, que podem substituir os trabalhadores humanos em trabalhos difíceis ou monótonos, ou onde o homem pudesse se deparar com condições perigosas.

O manipulador robótico mais comum é o braço robótico, que é um equipamento mecânico similar aos membros superiores humanos, com articulações e partes sólidas, podendo executar tarefas diversas. À junção das partes do braço robótico se dá o nome de cadeia aberta, pois podem ser representados por um conjunto de mecanismos simples ligados, que possuem movimentação independente. Os componentes que formam a cadeia do braço robótico, conforme o exemplo da Figura 3.3, são os seguintes:

- Base: onde o braço é fixado, podendo ser classificada como fixa, quando instalada em uma estrutura imóvel, ou móvel, quando é instalada em um carro, trilho ou algum outro equipamento que se desloque;
- Elos: objetos rígidos responsáveis por dar comprimento e sustentação ao braço;

Figura 3.3: Braço robótico IRB 140 da marca *Asea Brown Boveri* (ABB)



Fonte: ABB[24], com marcações do autor.

- Juntas: mecanismos que transmitem movimento ao conjunto. Estão instaladas entre os elos, mudando a posição relativa entre estes. Elas são classificadas quanto ao tipo de movimentação que exercem, sendo as de movimento linear chamadas de juntas deslizantes ou prismáticas e as de movimento rotacional, juntas rotacionais ou de revolução;
- Efetuador: elemento final da cadeia. É onde é instalada a ferramenta que exercerá a atividade final proposta pelo braço robótico. Alguns exemplos de ferramentas comuns instaladas nos efetuadores são garras, eletroímãs, ventosas, parafusadeiras, equipamentos de soldagem, entre outros.

Define-se como cinemática direta o uso de recursos matemáticos para se localizar o efetuador do braço robótico, fazendo o uso de informações da localização da base, do tamanho dos elos e da posição das juntas.

Ao conjunto de todas as posições e orientações as quais o braço robótico pode alcançar é dado o nome de espaço de trabalho.

Além da cinemática direta, outras áreas de estudo de braços robóticos são os seguintes:

- Cinemática inversa, na qual, dada uma posição para o efetuador, é verificada o conjunto de soluções de posição das juntas para alcançá-la, caso seja possível;

- Dinâmica, onde é verificado o torque, momento, e demais forças envolvidas na tarefa do braço robótico;
- Controle, que lida com a maneira a qual é possível enviar comandos para o braço, de forma a alterar a posição de juntas, e/ou operar as ferramentas acopladas em seu efetuador;
- Automação, que é a área que lida com a sequência de tarefas a serem executadas, bem como na geração de trajetórias, tendo em consideração algumas variáveis, como o tempo e a resposta de sensores externos à máquina.

O estudo de caso desse trabalho se ateu à cinemática direta do braço robótico.

3.2.2 Notação de Denavit-Hartenberg

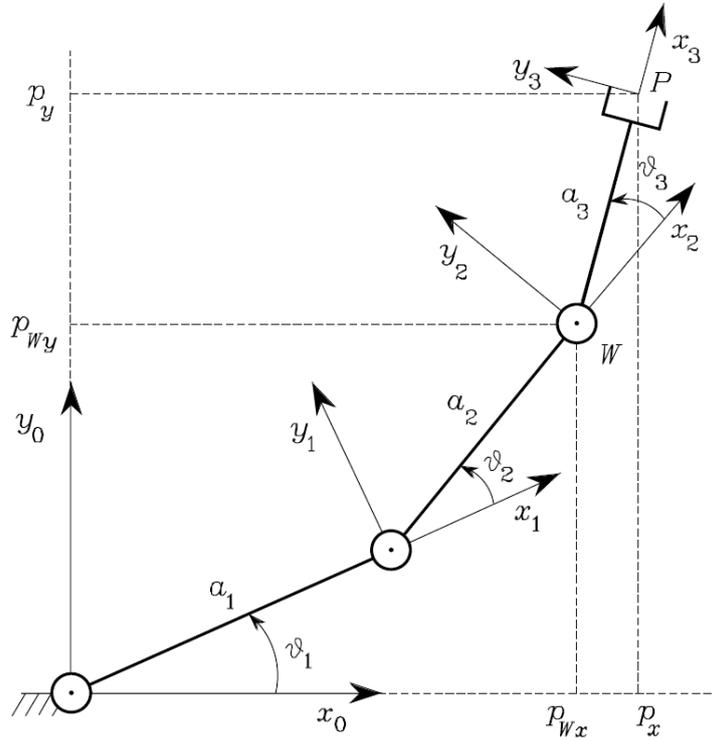
Denavit e Hartenberg[25] apresentaram uma proposta de utilização de matrizes para representar mecanismos. Em suma, propuseram a utilização de matrizes para a descrição de mecanismos, incluindo seus componentes, contendo as informações necessárias para, não somente descrever seu posicionamento, mas também sua orientação.

A representação de um elo de um braço robótico se dá por meio de uma translação com valor constante. As juntas rotacionais são representadas por uma rotação com ângulo variável e as prismáticas, como uma translação com comprimento variável. O sistema de coordenadas adotado na descrição de um elemento leva em consideração o sistema de coordenadas de todos os elementos anteriores reunidos.

Por exemplo, o braço robótico planar apresentado por Siciliano et al.[26], presente na Figura 3.4, possui três elos e três juntas rotacionais, e o ponto P descreve a posição do final da cadeia, isto é, seu efetuador. Segundo a notação de Denavit-Hartenberg, a cada um desses elementos é atribuída uma matriz de transformação homogênea.

Como o exemplo é no plano, a matriz de rotação que equivale a cada junta é similar à matriz apresentada na Equação 2.9, só variando o ângulo em questão. Com as juntas em estado inicial, isto é, com valor 0° atribuído aos ângulos ϑ_1 , ϑ_2 e ϑ_3 da Figura 3.4, os elos são representados como translações ao longo do eixo \vec{x} . A descrição de cada elemento acontece, então, da seguinte forma:

Figura 3.4: Braço robótico planar



Fonte: Siciliano et al.[26].

$$\begin{aligned}
 \text{Junta 1: } T_{(J1)} &= \begin{bmatrix} \cos \vartheta_1 & -\text{sen } \vartheta_1 & 0 \\ \text{sen } \vartheta_1 & \cos \vartheta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{Elo 1: } T_{(E1)} &= \begin{bmatrix} 1 & 0 & a_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 \text{Junta 2: } T_{(J2)} &= \begin{bmatrix} \cos \vartheta_2 & -\text{sen } \vartheta_2 & 0 \\ \text{sen } \vartheta_2 & \cos \vartheta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{Elo 2: } T_{(E2)} &= \begin{bmatrix} 1 & 0 & a_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
 \text{Junta 3: } T_{(J3)} &= \begin{bmatrix} \cos \vartheta_3 & -\text{sen } \vartheta_3 & 0 \\ \text{sen } \vartheta_3 & \cos \vartheta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{Elo 3: } T_{(E3)} &= \begin{bmatrix} 1 & 0 & a_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Ainda segundo a notação, a matriz $T_{(Ef)}$ que representa o efetuador do braço robótico, descrevendo o sistema de coordenadas ao qual o ponto P é a origem, é dada pelo produto:

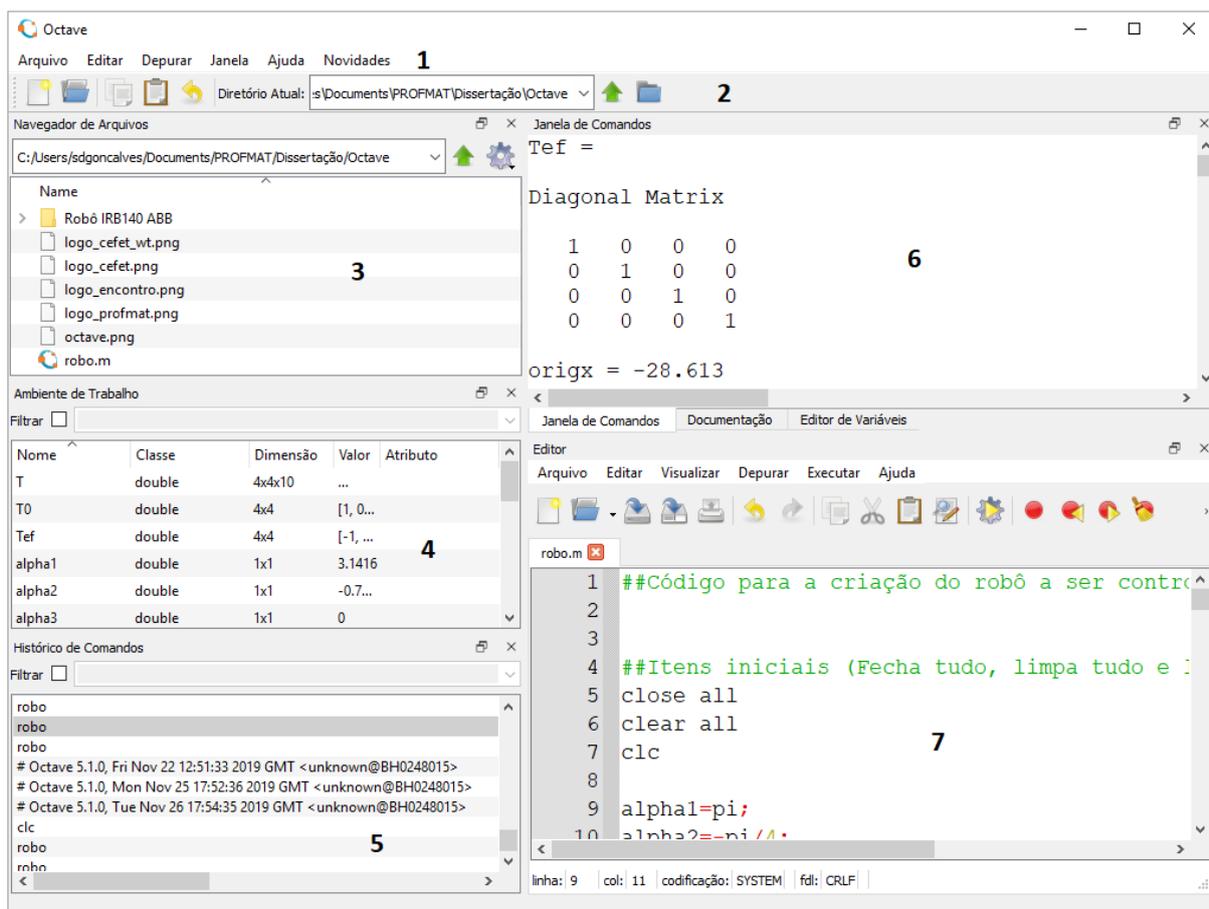
$$T_{(Ef)} = T_{(J1)}T_{(E1)}T_{(J2)}T_{(E2)}T_{(J3)}T_{(E3)}$$

3.2.3 A linguagem de programação Octave

O Octave é uma linguagem de programação de alto nível, destinada inicialmente a cálculos numéricos, mas que possui muitas ferramentas para resolver problemas comuns de Álgebra Linear[8]. Além da sintaxe simples e da facilidade de manipulação de matrizes, a escolha da linguagem de programação Octave se mostrou conveniente devido ao seu bom ambiente de programação, sua fácil customização, as ferramentas de plotagem de gráficos e a criação de interface gráfica de usuário. A versão do Octave utilizada é a 5.2.0, em um computador com sistema operacional *Microsoft Windows 10*.

O ambiente de programação do Octave, ilustrado pela Figura 3.5, possui, basicamente, os seguintes itens em sua interface gráfica:

Figura 3.5: Ambiente de programação do Octave



Fonte: Captura de tela do *software* Octave com anotações do autor.

1. Barra de menus, com as opções “Arquivo”, “Editar” etc;
2. Barra de ferramentas, com os botões “Novo”, “Abrir” etc;

3. Navegador de Arquivos, onde é exibida a pasta ativa e os arquivos contidos nela;
4. Ambiente de Trabalho, que concentra as variáveis globais que foram declaradas e tem valor atribuído;
5. Histórico de Comandos, onde ficam registradas as últimas linhas de comando;
6. Janela de Comandos, onde são escritos as linhas de comandos a serem executados uma a uma, e exibidas o progresso da execução de comandos elaborados no Editor;
7. Editor, onde são escritos os códigos a serem executados, e que podem ser salvos.

Vale ressaltar que o Octave é um *software* livre e seu ambiente de programação é distribuído gratuitamente.

3.2.4 O *software* CoppeliaSim

De forma a verificar se a descrição do posicionamento e da orientação do braço robótico por meio de grupos matriciais, segundo a notação de Denavit-Hartenberg, podem representar uma descrição fiel, foi realizada a comparação com um exemplar de braço robótico. No entanto, ao invés de utilizar um braço robótico real, o estudo foi feito em um modelo de braço robótico disponível em um ambiente de simulação.

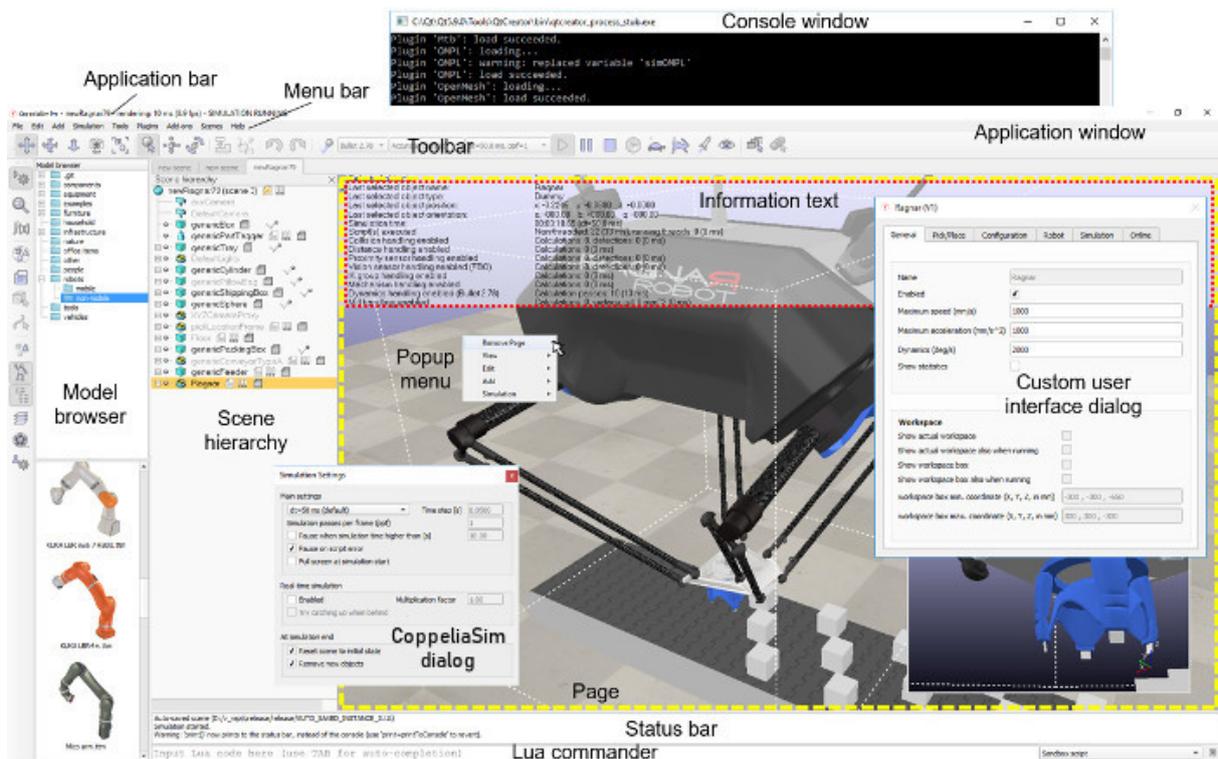
Foi utilizado, então, o simulador de robô CoppeliaSim[9], que possui ambiente de desenvolvimento integrado e é baseado em uma arquitetura de controle distribuída, em que cada objeto pode ser controlado individualmente. O controle pode ser feito tanto por *scripts* embarcados no próprio *software*, quanto externamente, por meio de ferramentas, como, por exemplo, interface de programação de aplicações (API, da sigla em inglês) escrita na linguagem Octave.

A versão do CoppeliaSim utilizada foi a Edu, número 4.1.0, em computador com sistema operacional *Microsoft Windows* 10. Por mais que não haja requisitos básicos para o uso do *software*, este foi utilizado em um computador com 8 gigabytes de memória RAM, processador com 2,5 gigahertz de frequência e placa de vídeo dedicada de 1 gigabyte.

Dentre as aplicações do CoppeliaSim informadas pelo desenvolvedor, pode-se destacar o uso em educação relacionada à robótica.

A janela da aplicação do CoppeliaSim, conforme ilustrado na Figura 3.6, possui, entre outros, os seguintes elementos relevantes a este trabalho:

Figura 3.6: Ambiente de Simulações do CoppeliaSim



Fonte: Coppelia Robotics[9].

- Barra de menus (*Menu bar*), onde são acessadas a maioria das funcionalidades do simulador. Possui as opções “*File*”, “*Edit*”, “*Add*” etc;
- Barras de ferramentas (*Toolbar*), onde estão os botões de ferramentas que são acessadas mais frequentemente;
- Navegador de modelos (*Model browser*), onde é possível acessar as pastas com os modelos de robôs, bem como de equipamentos, ferramentas e demais componentes da simulação. Ao selecionar a pasta, são exibidas miniaturas dos itens abaixo;
- Hierarquia da cena (*Scene hierarchy*), que é a lista de nomes dos itens contidos na cena;
- Página (*Page*), que é onde são exibidos os elementos que compõem a simulação. Durante a execução da simulação, é o local onde é possível ver as animações;
- Texto de informação (*Information text*), localizado logo acima da página da simulação, com informações sobre o item selecionado na Hierarquia da cena;

- Editor de comandos Lua (*Lua Commander*), localizado na posição extrema inferior. É onde se insere comandos na linguagem de programação Lua, que é a embarcada no simulador.

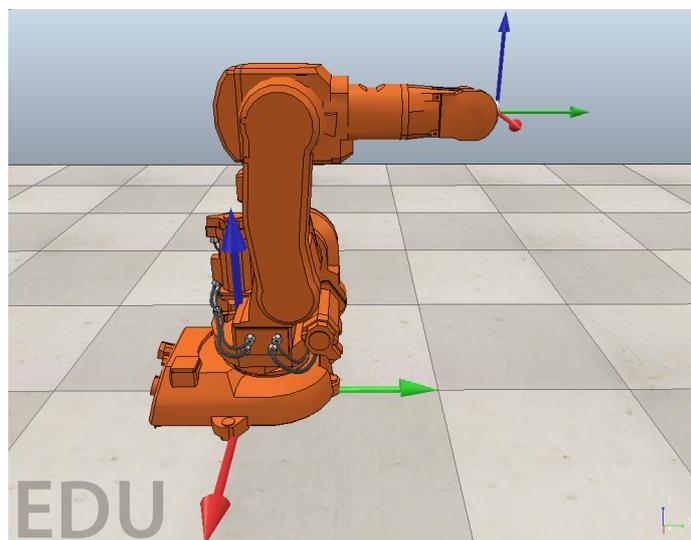
A versão Edu do CoppeliaSim é distribuída gratuitamente, possibilitando seu uso em ambiente acadêmico sem grandes investimentos.

3.2.5 Descrição dos elementos do braço robótico

Após a apresentação das ferramentas computacionais que foram utilizadas, para iniciar a simulação por meio de matrizes, fez-se necessário escolher um modelo de braço robótico para o estudo de casos. Dentre os modelos disponíveis no ambiente de simulação do CoppeliaSim, optou-se pelo IRB 140, da marca *Asea Brown Boveri* (ABB), por ter sua ficha de dados disponível na *internet*[24] e por possuir 6 graus de liberdade, o que significa que, dentro do seu espaço de trabalho, qualquer posição e orientação pode ser alcançada. A Figura 3.3 ilustra o braço em questão.

Uma vez feita a escolha do braço, foi realizado o levantamento das juntas e dos elos do braço robótico por meio das dimensões descritas pelo manual fornecido pelo fabricante e pela análise dos elementos que compõem o braço no modelo fornecido pelo CoppeliaSim. Para analisar o modelo do simulador, alinhando-o ao eixo \vec{y} , é necessário realizar as etapas de configuração no ambiente de simulação descritas no Anexo B. A Figura 3.7 representa o braço robótico após a configuração.

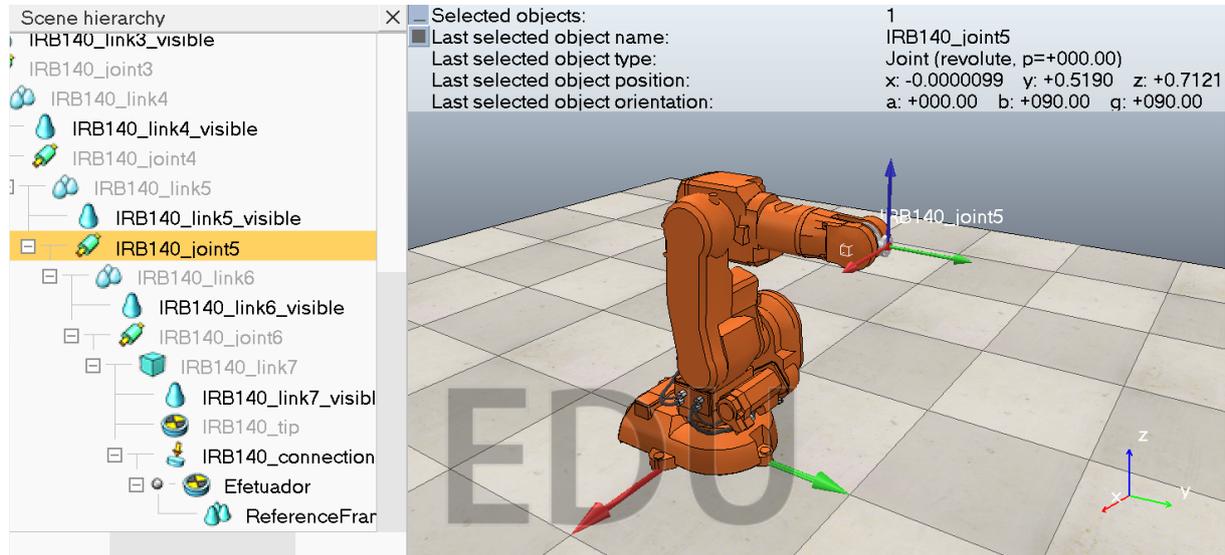
Figura 3.7: Braço IRB 140 no CoppeliaSim após sua configuração



Fonte: Elaborado pelo autor.

A justificativa para o alinhamento do braço robótico com o eixo \vec{y} é explicada quando a simulação for por meio de interface gráfica de usuário, na Subseção 3.2.7.

Figura 3.8: Braço IRB 140 no CoppeliaSim, com a junta 5 selecionada na Hierarquia da Cena e descrita no Texto de Informação



Fonte: Elaborado pelo autor.

Com a Hierarquia da cena expandida, é possível verificar todos os itens que compõem o modelo. Para o levantamento da posição das juntas do braço no espaço, basta clicar na junta correspondente (“IRB140_joint5”, por exemplo) e verificar as coordenadas no Texto de Informação, como descrito na Figura 3.8. A Tabela 3.1 exibe as informações das juntas, utilizando os dados do manual do fabricante e do modelo do CoppeliaSim.

Número da Junta	Coordenadas das Juntas (m)	Alcance de Trabalho	Escala da Simulação	Valor Inicial	Eixo da Rotação
1	(0 0,069 0)	360°	-180° a +180°	0°	\vec{z}
2	(0 0,139 0,352)	200°	-100° a +100°	0°	\vec{x}
3	(0 0,139 0,712)	280°	-140° a +140°	0°	\vec{x}
4	(0 0,365 0,712)	Ilimitado	-180° a +180°	0°	\vec{y}
5	(0 0,519 0,712)	230°	-115° a +115°	-90°	\vec{x}
6	(0 0,565 0,712)	Ilimitado	-180° a +180°	0°	\vec{y}

Tabela 3.1: Juntas do braço robótico

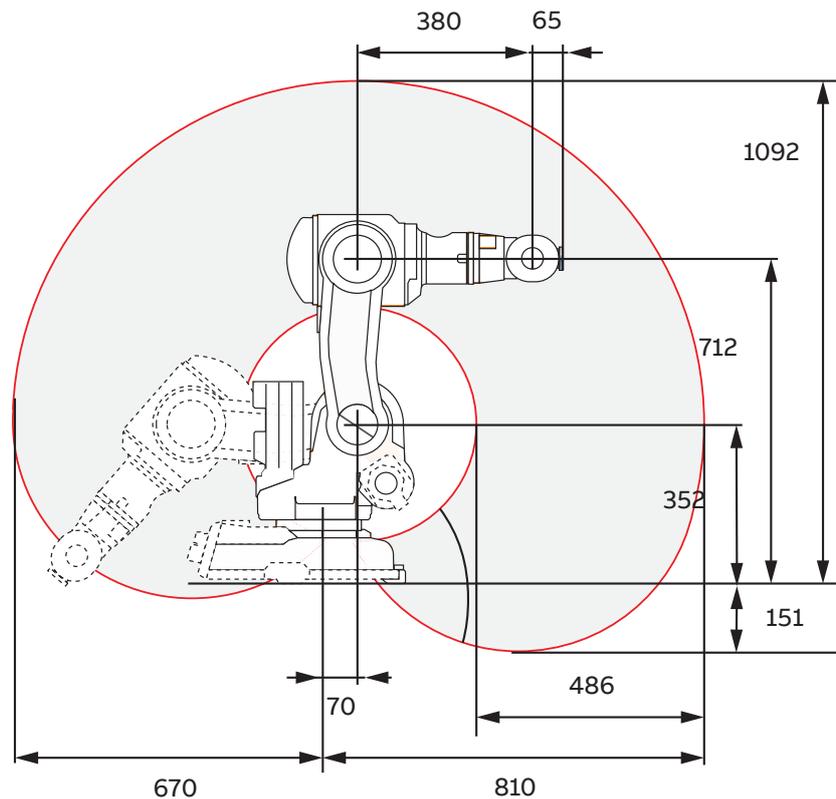
Por mais que as juntas 4 e 6 possuam movimentação ilimitada, optou-se por utilizar a faixa de -180° a $+180^\circ$, para que não haja posições alcançáveis com ângulos diferentes.

A matriz de transformação homogênea $T_{(J_i)} = \begin{bmatrix} R_{(i)} & 0_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}$, com $1 \leq i \leq 6$, corresponde à i -ésima junta do braço robótico, sendo a submatriz $R_{(i)}$ a matriz de rotação,

em torno do eixo exibido na última coluna da Tabela 3.1, que descreve a movimentação da junta.

A obtenção dos vetores que descrevem os elos que estão entre juntas adjacentes, segundo a notação de Denavit-Hartenberg, é por meio da diferença das coordenadas de tais juntas. O primeiro elo é descrito pela diferença entre a junta 1 e o ponto de origem, e o último, entre o ponto (0 0,584 0,712), referente à conexão do efetuador (“IRB140_connection” na Hierarquia da cena), e a junta 6. Comparando com vetores calculados com as dimensões disponíveis no manual do braço robótico, conforme descrito na Figura 3.9, obtém-se a Tabela 3.2, onde os elos são apresentados, com os elementos que os limitam e os vetores de suas translações correspondentes, com coordenadas em metros.

Figura 3.9: Dimensões do braço robótico IRB 140 da ABB em milímetros



Fonte: ABB[24].

A matriz de transformação homogênea $T_{(E_i)} = \begin{bmatrix} I_3 & \mathbf{p}^{(i)} \\ 0_{1 \times 3} & 1 \end{bmatrix}$, com $1 \leq i \leq 7$, corresponde ao i -ésimo elo do braço robótico, sendo o vetor $\mathbf{p}^{(i)}$ o descrito na última coluna da Tabela 3.2.

Após a descrição das juntas e dos elos, a descrição do efetuador do braço robótico,

Número do Elo	Elementos que Limitam	Vetor Correspondente (m)
1	Origem e Junta 1	$[0 \ 0,069 \ 0,000]^T$
2	Junta 1 e Junta 2	$[0 \ 0,070 \ 0,352]^T$
3	Junta 2 e Junta 3	$[0 \ 0,000 \ 0,360]^T$
4	Junta 3 e Junta 4	$[0 \ 0,226 \ 0,000]^T$
5	Junta 4 e Junta 5	$[0 \ 0,154 \ 0,000]^T$
6	Junta 5 e Junta 6	$[0 \ 0,046 \ 0,000]^T$
7	Junta 6 e Ponto de Conexão	$[0 \ 0,019 \ 0,000]^T$

Tabela 3.2: Elos do braço robótico

segundo a notação de Denavit-Hartenberg para a cadeia aberta, é dada por:

$$\mathbf{T}_{(Ef)} = \mathbf{T}_{(E1)}\mathbf{T}_{(J1)}\mathbf{T}_{(E2)}\mathbf{T}_{(J2)}\mathbf{T}_{(E3)}\mathbf{T}_{(J3)}\mathbf{T}_{(E4)}\mathbf{T}_{(J4)}\mathbf{T}_{(E5)}\mathbf{T}_{(J5)}\mathbf{T}_{(E6)}\mathbf{T}_{(J6)}\mathbf{T}_{(E7)} \quad (3.1)$$

No entanto, para simplificar a notação, foi utilizado o resultado apresentado na Proposição 2.63. Dessa forma, a matriz de transformação que descreve o braço robótico é:

$$\mathbf{T}_{(Ef)} = \mathbf{T}_{(1)}\mathbf{T}_{(2)}\mathbf{T}_{(3)}\mathbf{T}_{(4)}\mathbf{T}_{(5)}\mathbf{T}_{(6)}\mathbf{T}_{(7)}, \quad (3.2)$$

onde, sendo α_i o ângulo descrito pela i -ésima junta, tem-se que:

$$\mathbf{T}_{(1)} = \mathbf{T}_{(E1)}\mathbf{T}_{(J1)} = \begin{bmatrix} \cos \alpha_1 & -\text{sen } \alpha_1 & 0 & 0 \\ \text{sen } \alpha_1 & \cos \alpha_1 & 0 & 0,069 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{(2)} = \mathbf{T}_{(E2)}\mathbf{T}_{(J2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_2 & -\text{sen } \alpha_2 & 0,070 \\ 0 & \text{sen } \alpha_2 & \cos \alpha_2 & 0,352 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{(3)} = \mathbf{T}_{(E3)}\mathbf{T}_{(J3)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_3 & -\text{sen } \alpha_3 & 0 \\ 0 & \text{sen } \alpha_3 & \cos \alpha_3 & 0,360 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{(4)} = T_{(E4)}T_{(J4)} = \begin{bmatrix} \cos \alpha_4 & 0 & \sin \alpha_4 & 0 \\ 0 & 1 & 0 & 0,226 \\ -\sin \alpha_4 & 0 & \cos \alpha_4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{(5)} = T_{(E5)}T_{(J5)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_5 & -\sin \alpha_5 & 0,154 \\ 0 & \sin \alpha_5 & \cos \alpha_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{(6)} = T_{(E6)}T_{(J6)} = \begin{bmatrix} \cos \alpha_6 & 0 & \sin \alpha_6 & 0 \\ 0 & 1 & 0 & 0,046 \\ -\sin \alpha_6 & 0 & \cos \alpha_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{(7)} = T_{(E7)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0,019 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2.6 Montagem do braço robótico no Octave

Antes de iniciar a programação no ambiente do Octave, é necessário criar uma pasta específica no sistema, a qual deve ser selecionada no Navegador de Arquivos para que todos os arquivos criados sejam salvos nela. No Editor, selecionar Arquivo/Novo Script, para que seja aberto um editor de texto, onde o código é inserido. Selecionar, então, o cursor do mouse na primeira linha do editor, para início da escrita.

A sintaxe do Octave é simples e não requer que se insira um “;” ao final de cada linha. No entanto, a não inserção acarreta em exibição de cada linha compilada na Janela de Comandos, o que pode gerar excesso de informação.

Ao se inserir o caractere “#”, o texto escrito após este é considerado comentário e, portanto, não é compilado. Para o início da escrita, é comumente utilizado os seguintes comandos, com o seu significado escrito no comentário da linha:

```
clear all; #Apaga as variáveis do Ambiente de Trabalho
close all; #Fecha todas as figuras abertas
```

```
clc; #Limpa a Janela de Comandos
```

Em seguida, são declaradas as variáveis correspondentes aos ângulos das juntas, e atribuído o valor 0°. Dessa forma, tem-se que:

```
j1=0; #Ângulo da junta 1  
j2=0; #Ângulo da junta 2  
j3=0; #Ângulo da junta 3  
j4=0; #Ângulo da junta 4  
j5=0; #Ângulo da junta 5  
j6=0; #Ângulo da junta 6
```

Logo após, para declarar a primeira matriz de transformação homogênea, usa-se o seguinte código:

```
T(:, :, 1)=[cosd(j1), -sind(j1), 0, 0;  
sind(j1), cosd(j1), 0, 70;  
0, 0, 1, 0;  
0, 0, 0, 1];
```

No código, a variável T possui três dimensões numéricas, sendo a primeira referente às linhas da matriz, a segunda referente às colunas, e a terceira, um número sequencial, que vai de 1 a 7. A escrita T(:, :, 1) significa que são atribuídos valores para as linhas e colunas da matriz número 1.

Os valores da matriz são delimitados por colchetes, e são declarados linha a linha, sendo as linhas separadas por “;” e os valores separados por “,”. As funções `sind(j1)` e `cosd(j1)` representam as funções seno e cosseno da variável j1, respectivamente, com argumentos em graus.

A declaração das demais matrizes de transformação homogêneas foram omitidas, de forma a dar sequência.

A matriz de transformação homogênea do efetuador, que é o produto de todas as matrizes dos elos e juntas, é obtida por meio de um laço, isto é, de uma estrutura em que o mesmo código é executado repetidamente. Dessa forma, tem-se que:

```
Tef=eye(4);  
for ind=1:7
```

```
Tef=Tef*T(:, :, ind);  
pef(:, ind)=Tef(1:3, 4);  
end;
```

A linha `Tef=eye(4);` atribui à matriz `Tef` um valor inicial, que é a matriz identidade de ordem 4.

O laço descrito em seguida descreve que, para a variável de índice `ind` variando de 1 a 7, são executadas as duas linhas de comando a seguir até o final do laço descrito pelo comando `end;`, totalizando 7 iterações.

O comando `Tef=Tef*T(:, :, ind);` atribui à variável `Tef` o valor da iteração anterior (ou a matriz identidade, caso seja a primeira iteração) multiplicado pela “`ind`-ésima” matriz `T`, que foi declarada anteriormente.

A linha de comando `pef(:, ind)=Tef(1:3, 4);` atribui os valores das posições de cada junta à variável `pef`. Esse valor é obtido do vetor que significa a translação da matriz de transformação homogênea após cada multiplicação que ocorre no laço.

Ao final desse código, a matriz `Tef` deve possuir dimensões 4×4 , sendo a matriz de transformação homogênea referente ao efetuador do braço robótico. A variável `pef`, por sua vez, deve possuir dimensões 3×7 , sendo que os 6 primeiros vetores coluna representam as coordenadas das juntas, em ordem, e o último vetor coluna representa as coordenadas do efetuador.

Para testar o código escrito até então, basta clicar, no Editor, em Executar/Salvar Arquivo e Executá-lo. É aberta, então, uma janela para salvar o arquivo, caso essa seja a primeira vez que este comando é executado. Após salvar, o código é compilado e as variáveis declaradas aparecem no Ambiente de Trabalho.

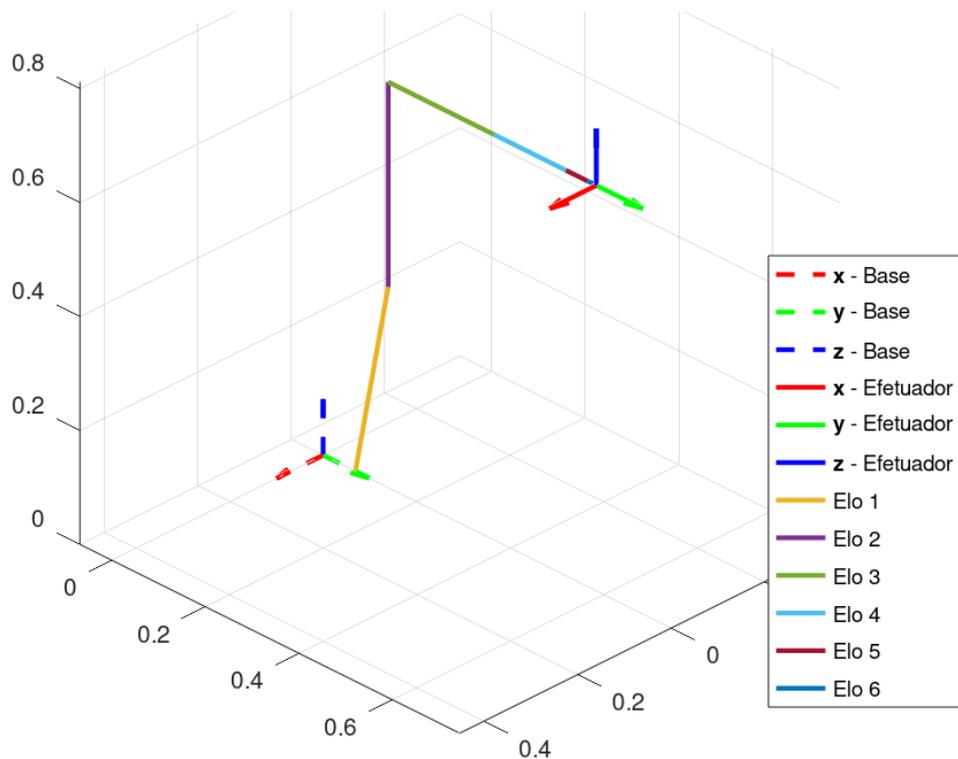
Ao digitar o nome da variável `Tef` na Janela de Comandos e teclar *Enter*, é exibido o seguinte resultado:

```
Tef=  
1.00000 0.00000 0.00000 0.00000  
0.00000 1.00000 0.00000 0.58400  
0.00000 0.00000 1.00000 0.71200  
0.00000 0.00000 0.00000 1.00000
```

A tabela de dados exibida é a matriz de transformação homogênea do efetuador. Pode-se observar que a submatriz de rotação, pertencente a $SO(3)$, é a matriz identidade, logo a orientação do efetuador é a mesma da referência. O vetor coluna que representa a translação exibe as coordenadas do efetuador e é o mesmo que foi visto no CoppeliaSim durante a montagem e elaboração da Tabela 3.2.

O código completo do Octave, não somente para simular as matrizes referentes ao braço robótico, mas também plotar gráficos em três dimensões que representam os elos e os vetores ortonormais da referência da base do braço e do efetuador, está disponível no Anexo A.1, com os devidos comentários. Após a execução do código, com os ângulos das juntas com o valor de 0° , obtém-se a Figura 3.10.

Figura 3.10: Posição inicial do braço robótico simulado no Octave



Fonte: Elaborado pelo autor.

Para verificar se o braço robótico simulado no Octave é uma boa representação do braço robótico IRB 140 da marca ABB, é realizada uma comparação com o braço deste modelo no CoppeliaSim, utilizando as mesmas posições das juntas em ambos os ambientes.

Para alterar a posição do braço robótico no Octave, basta alterar os valores atribuídos para o ângulo das juntas, e executar o código novamente. Um botão da barra de ferramentas do Editor que também funciona como Salvar Arquivo e Executá-lo é o botão 

homônimo.

Para o início da comparação, foram escolhidos, de forma aleatória, ângulos para as juntas do braço robótico. A Tabela 3.3 apresenta os ângulos utilizados nos testes e a matriz de transformação homogênea do efetuador ao final de cada teste.

Número do Teste	Ângulos das Juntas	Matriz de Transformação Homogênea do Efetuador			
1	j1=90; j2=45;	-0.25882	-0.68301	0.68301	-0.22689
	j3=-30; j4=-90;	0.00000	0.70711	0.70711	0.11496
	j5=-45; j6=180;	-0.96593	0.18301	-0.18301	0.71681
		0.00000	0.00000	0.00000	1.00000
2	j1=180; j2=0;	-0.28869	0.64086	-0.71131	0.04166
	j3=90; j4=-45;	0.71131	0.64086	0.28869	0.04066
	j5=115; j6=45;	0.64086	-0.42262	-0.64086	1.06453
		0.00000	0.00000	0.00000	1.00000
3	j1=0; j2=-90;	0.50000	0.70711	0.50000	0.04596
	j3=135; j4=90;	0.14645	0.50000	-0.85355	0.80020
	j5=45; j6=-45;	-0.85355	0.50000	0.14645	0.65320
		0.00000	0.00000	0.00000	1.00000

Tabela 3.3: Resultado dos testes no Octave

Com o braço robótico do CoppeliaSim já previamente parametrizado pelas etapas descritas no Anexo B, basta iniciar a simulação em “*Simulation/Start simulation*” para possibilitar a animação. Ao iniciar, o braço está com todas as juntas com o ângulo 0° e é aberta uma janela contendo a matriz de transformação homogênea do efetuador.

Para alterar o valor das juntas durante a animação, basta inserir o comando abaixo no Editor de comandos Lua e teclar *Enter*:

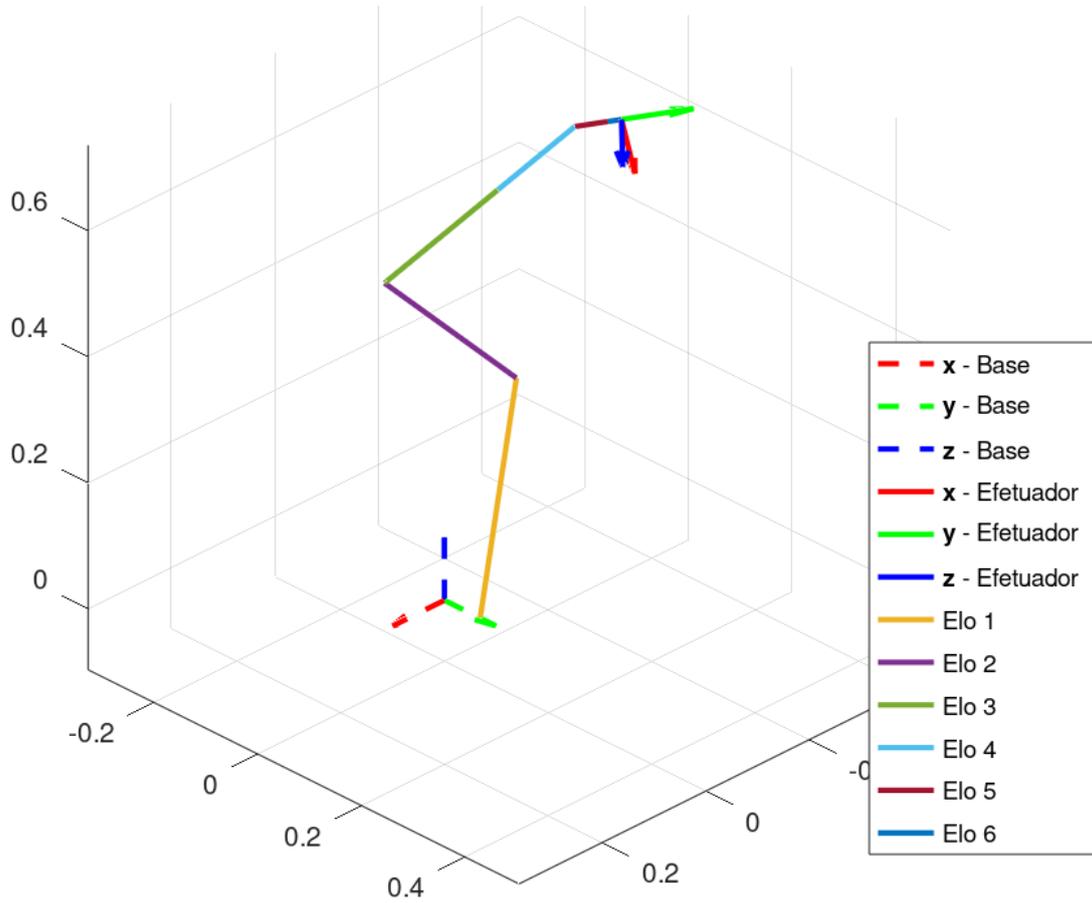
```
sim.setJointTargetPosition(sim.getObjectHandle('IRB140_joint1'),90*math.pi/180)
```

No comando do exemplo acima é alterado o valor da junta 1 para 90° , que é convertido para radianos.

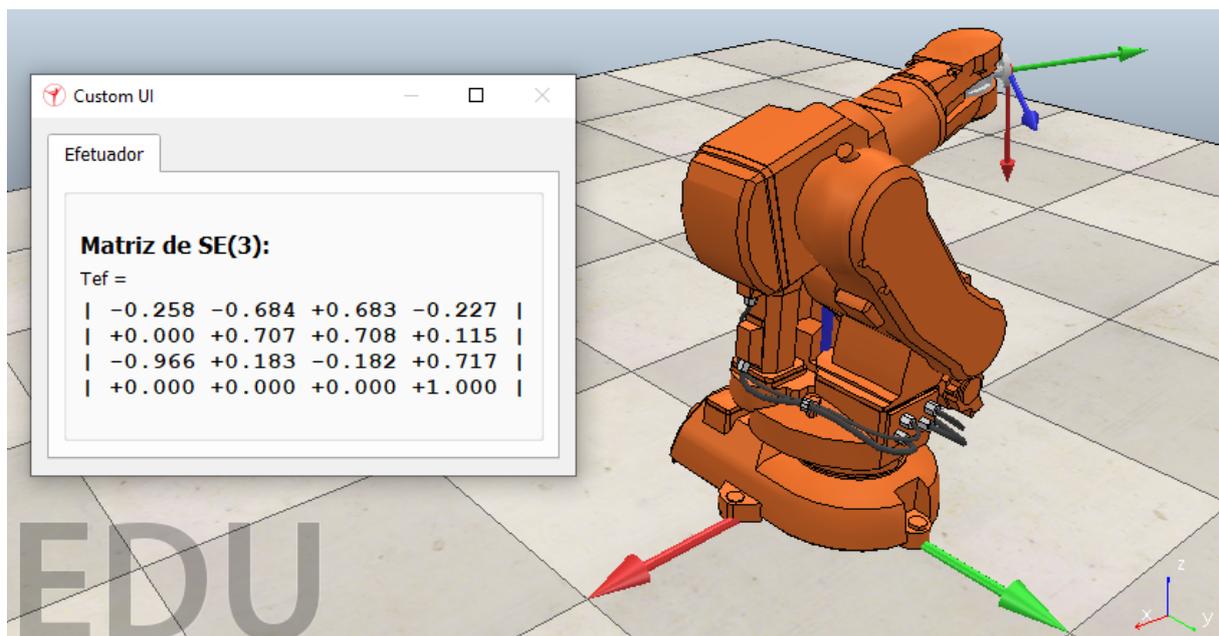
Após a alteração de cada junta, foi comparado tanto o posicionamento e a orientação das representação gerada pelo Octave e pelo CoppeliaSim, quanto as matrizes de transformação homogênea. As Figuras 3.11, 3.12 e 3.13 exibem as orientações e as matrizes do CoppeliaSim referentes aos testes 1, 2 e 3, respectivamente. As matrizes do CoppeliaSim podem ser comparadas com as matrizes exibidas na Tabela 3.3.

Figura 3.11: Comparação das simulações no Octave e no CoppeliaSim - Teste 1

(a) Teste 1 no Octave



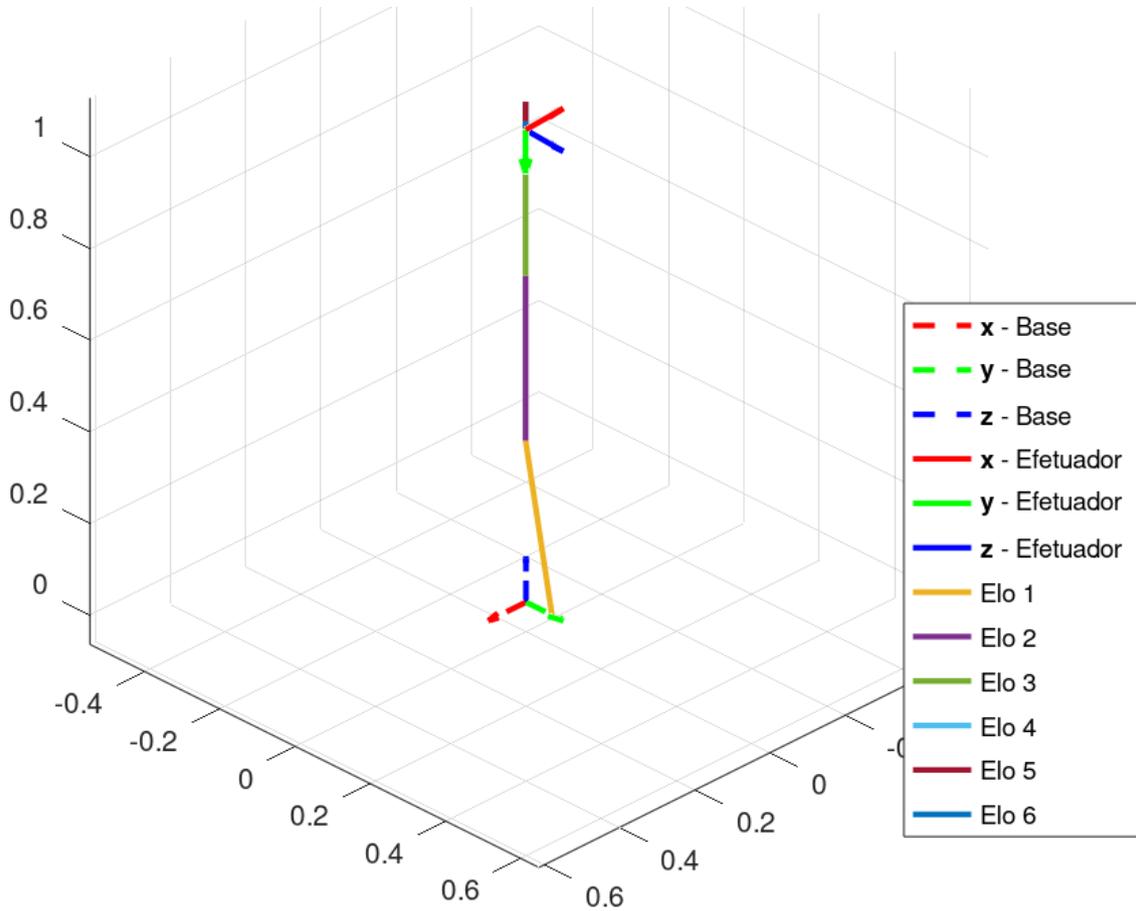
(b) Teste 1 no CoppeliaSim



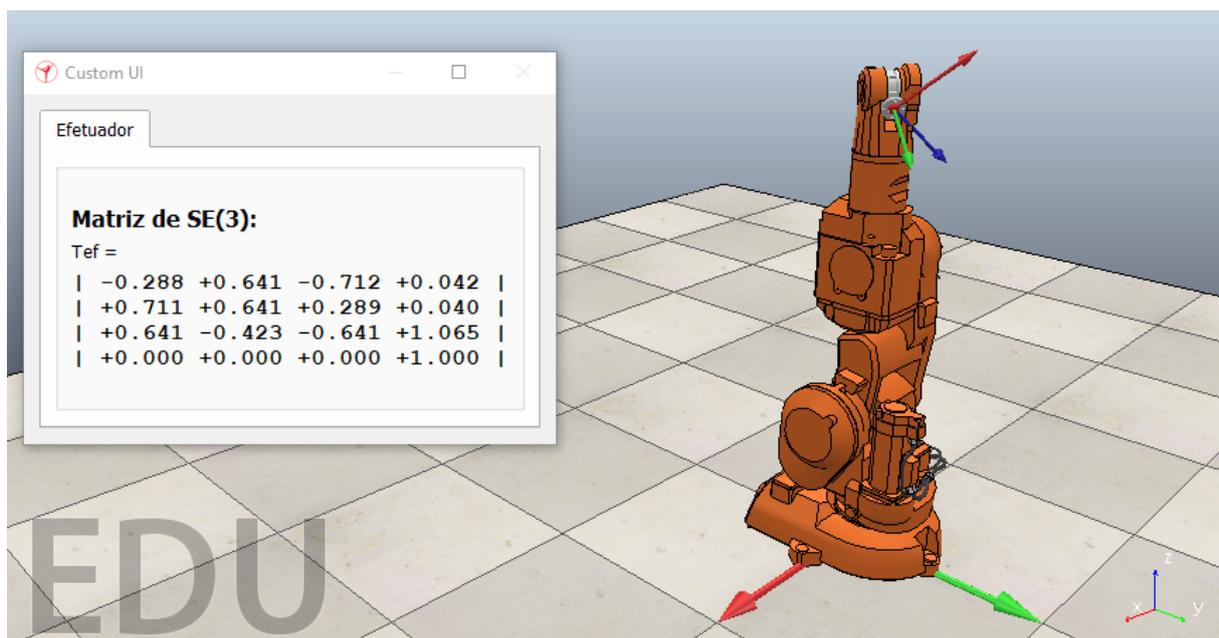
Fonte: Elaborado pelo autor.

Figura 3.12: Comparação das simulações no Octave e no CoppeliaSim - Teste 2

(a) Teste 2 no Octave



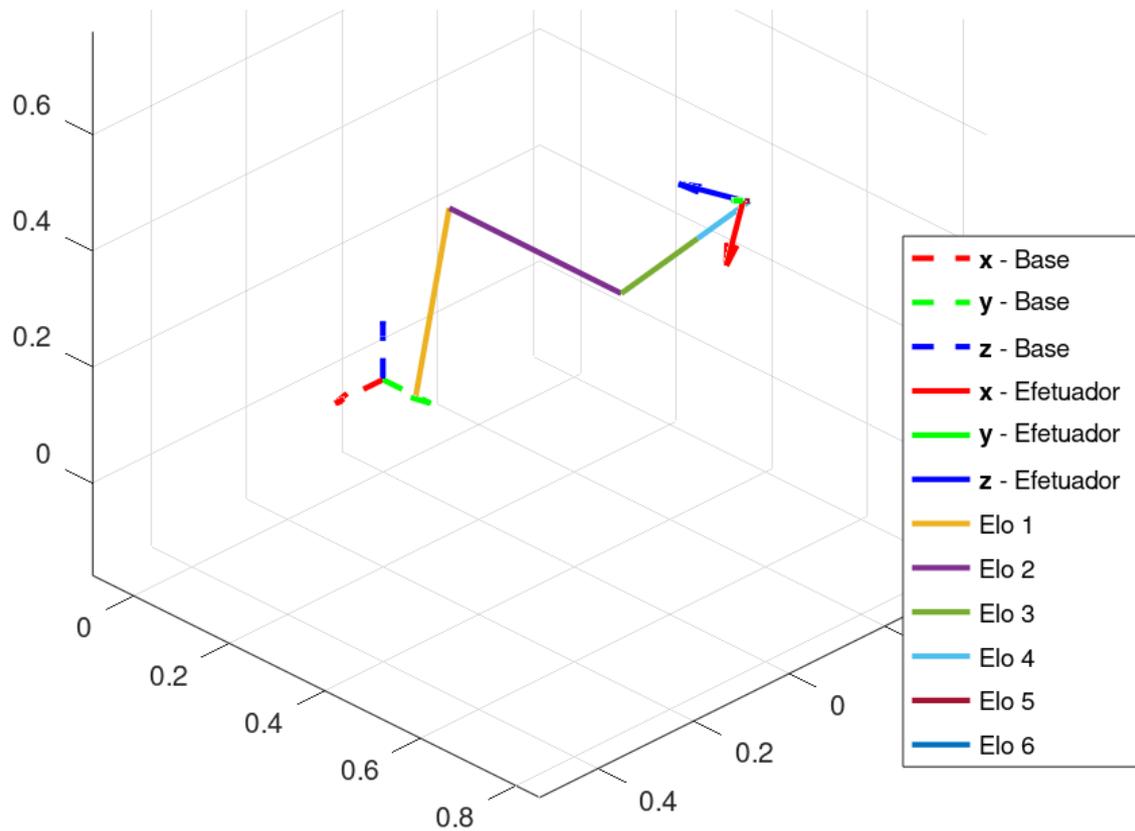
(b) Teste 2 no CoppeliaSim



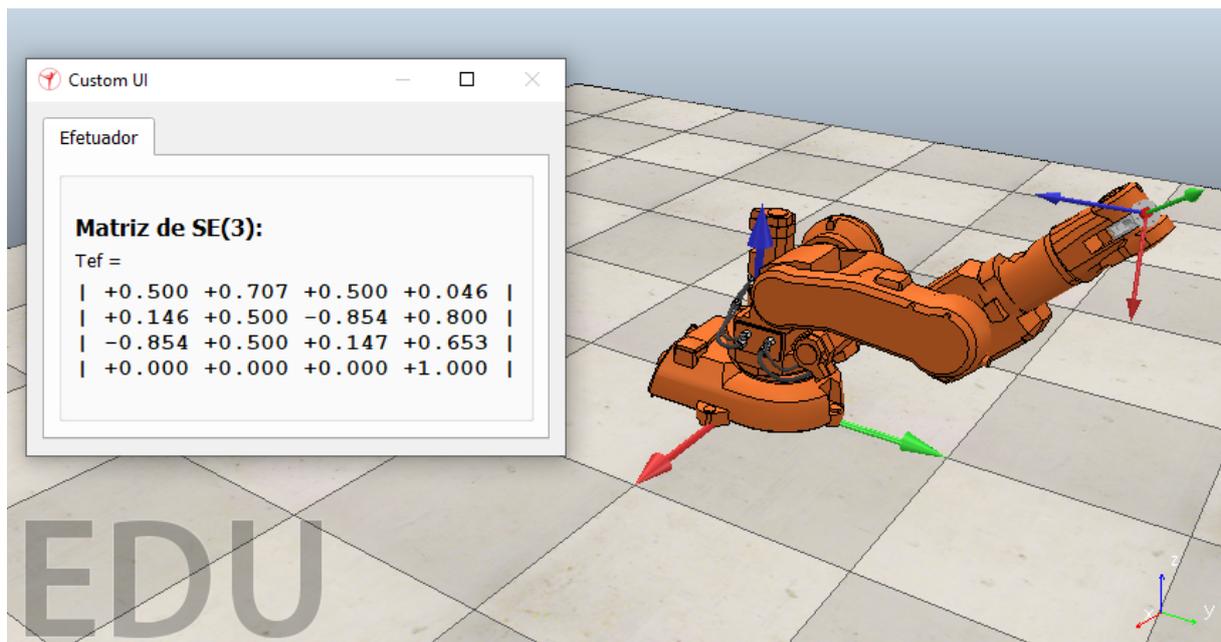
Fonte: Elaborado pelo autor.

Figura 3.13: Comparação das simulações no Octave e no CoppeliaSim - Teste 3

(a) Teste 3 no Octave



(b) Teste 3 no CoppeliaSim



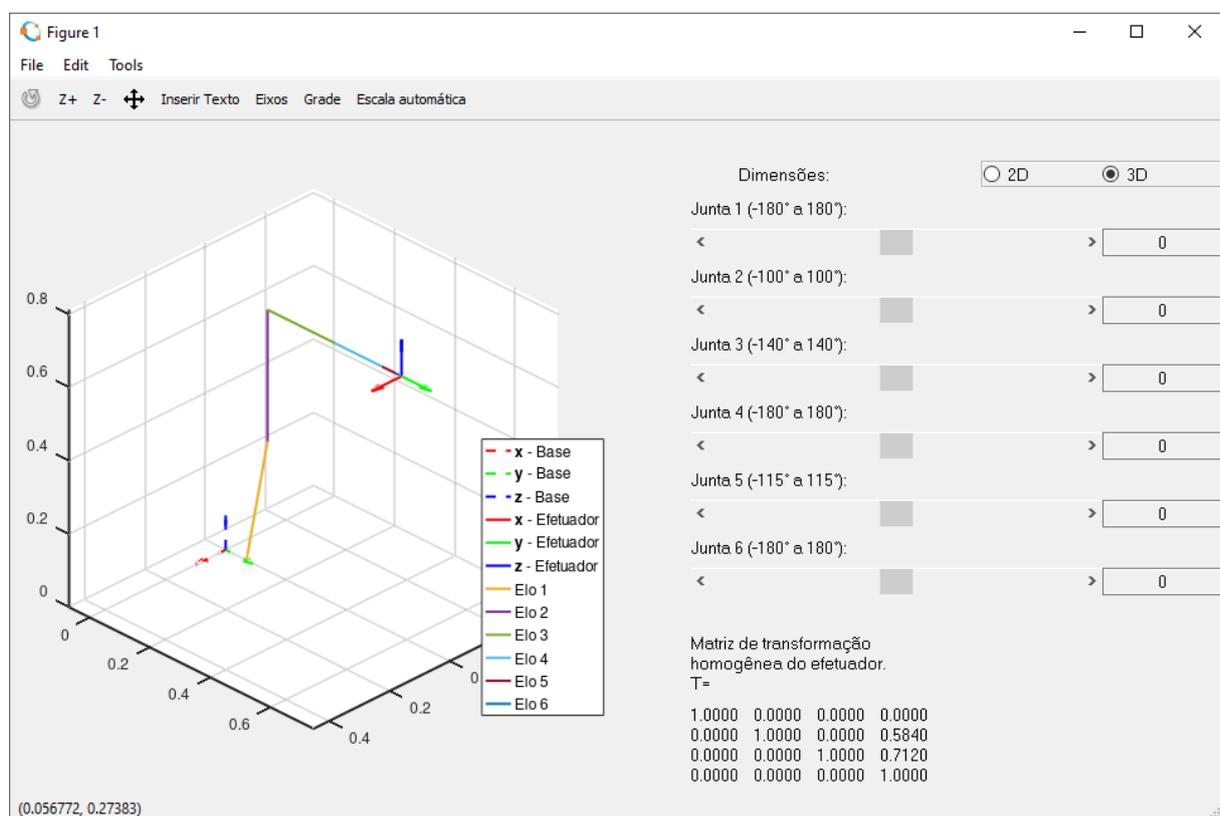
Fonte: Elaborado pelo autor.

3.2.7 Simulação por interface gráfica de usuário

De modo a facilitar a alteração dos ângulos das juntas tanto no Octave quanto no CoppeliaSim, foi desenvolvida uma interface gráfica de usuário no Octave que comunica com o CoppeliaSim, enviando comando de alteração dos ângulos do braço robótico. A criação das interfaces é descrita no Anexo C.

A Figura 3.14 exibe a interface gráfica de usuário em seu estado inicial, onde é possível observar a representação do braço robótico, similar ao apresentado na Figura 3.10, e os controles das juntas com o valor inicial, isto é, 0° . Na posição inferior é exibida a matriz de transformação homogênea do efetuador do braço robótico.

Figura 3.14: Interface gráfica de usuário para controle do braço robótico no Octave e no CoppeliaSim



Fonte: Elaborado pelo autor.

Na parte superior direita estão localizados botões de opção, onde é possível escolher se o controle do braço robótico será em duas ou três dimensões. A opção inicial é “3D”, que habilita o controle de todas as juntas. A opção “2D” será abordada mais adiante.

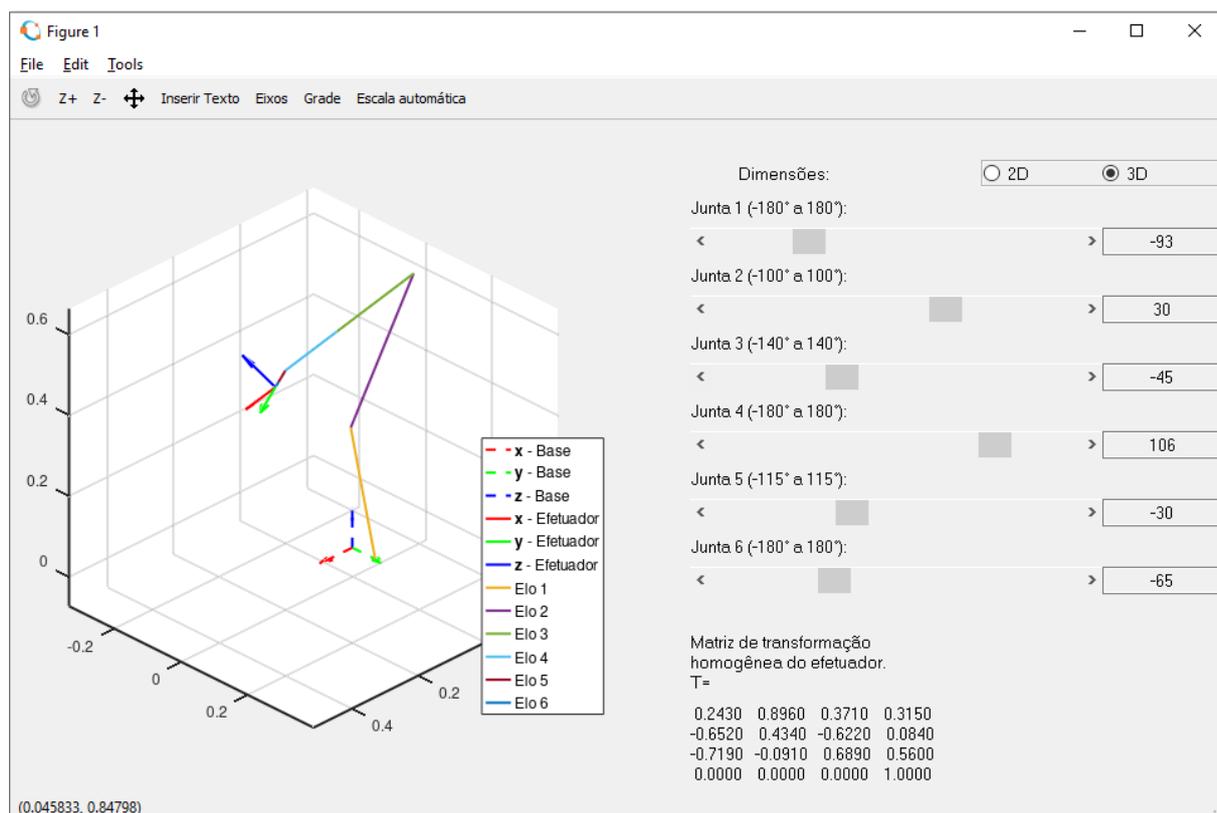
O início do código da interface gráfica apresenta a seguinte linha de comando:

```
h.online=true; #true=conectado; false=desconectado
```

Esta linha de código habilita a conexão ao CoppeliaSim se for atribuído o valor `true`, que é o valor lógico verdadeiro, à variável `h.online`. Caso se queira executar o código de forma *offline*, basta atribuir o valor `false` à variável. Para o desenvolvimento deste trabalho, optou-se, inicialmente, pelo teste *offline*.

Na interface gráfica, a alteração das juntas pode se dar tanto por meio da rolagem das barras horizontais, quanto pela caixa de texto posicionada à direita dessa barra. Os valores limites de cada junta, em graus, estão escritos entre parênteses e são os mesmos descritos na Tabela 3.1. Ao se alterar a posição da barra de rolagem, é alterado o valor na caixa de texto correspondente, e vice-versa. Em seguida é atualizada a representação e a matriz de transformação homogênea do efetuador exibida. Por exemplo, na Figura 3.15, é exibido um teste da interface com ângulos aleatórios.

Figura 3.15: Interface gráfica em 3D com ângulos aleatórios

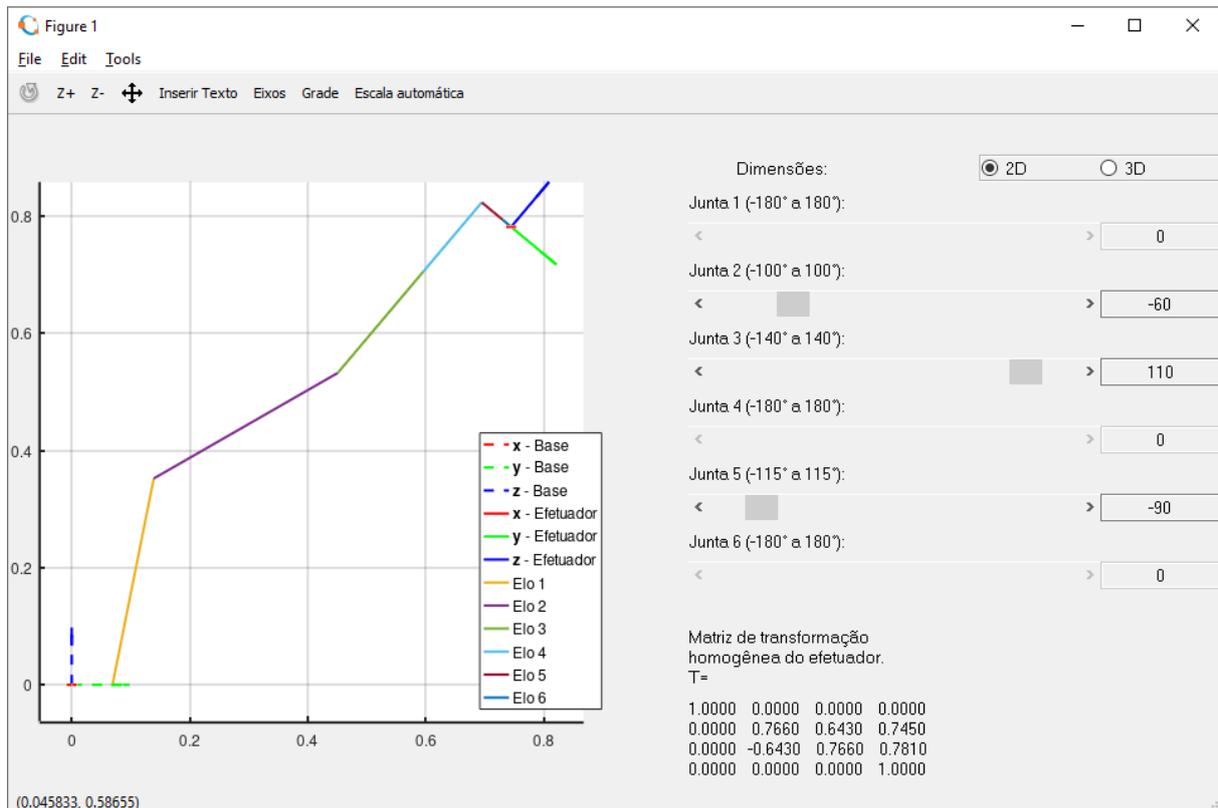


Fonte: Elaborado pelo autor.

O botão “2D” altera a opção para duas dimensões, desabilitando o controle das juntas 1, 3 e 6 e mudando a forma de apresentação do braço robótico, de forma a interpretá-lo como um braço planar. As juntas 2, 4 e 5, que continuaram alteráveis, são as que possuem rotação em torno do eixo \vec{x} , conforme descrito na Tabela 3.1. Esse foi o motivo de

alinhar o braço robótico ao eixo \vec{y} , pois, caso estivesse alinhado ao eixo \vec{x} como é o padrão do modelo, as rotações positivas não seriam em torno deste eixo, mas sim em torno de $-\vec{y}$, o que dificultaria a visualização em duas dimensões com o padrão de rotação estabelecido. Escolhendo valores aleatórios para os ângulos, com o controle em duas dimensões, obtém-se a configuração descrita pela Figura 3.16.

Figura 3.16: Interface gráfica em 2D com ângulos aleatórios



Fonte: Elaborado pelo autor.

A matriz de transformação homogênea do efetuador obtida na Figura 3.16 é a seguinte:

$$T = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.7660 & 0.6430 & 0.7450 \\ 0.0000 & -0.6430 & 0.7660 & 0.7810 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$$

É possível observar que, como as juntas rotacionam em torno do eixo \vec{x} , o primeiro vetor coluna da matriz é sempre $[1 \ 0 \ 0 \ 0]^T$, bem com o primeiro vetor linha é $[1 \ 0 \ 0 \ 0]$,

independente dos ângulos escolhidos. Assim sendo, é fácil assimilar as simulações como sendo em duas dimensões, com as matrizes pertencentes a $SE(2)$, como no exemplo a seguir:

```
T=  
0.7660 0.6430 0.7450  
-0.6430 0.7660 0.7810  
0.0000 0.0000 1.0000
```

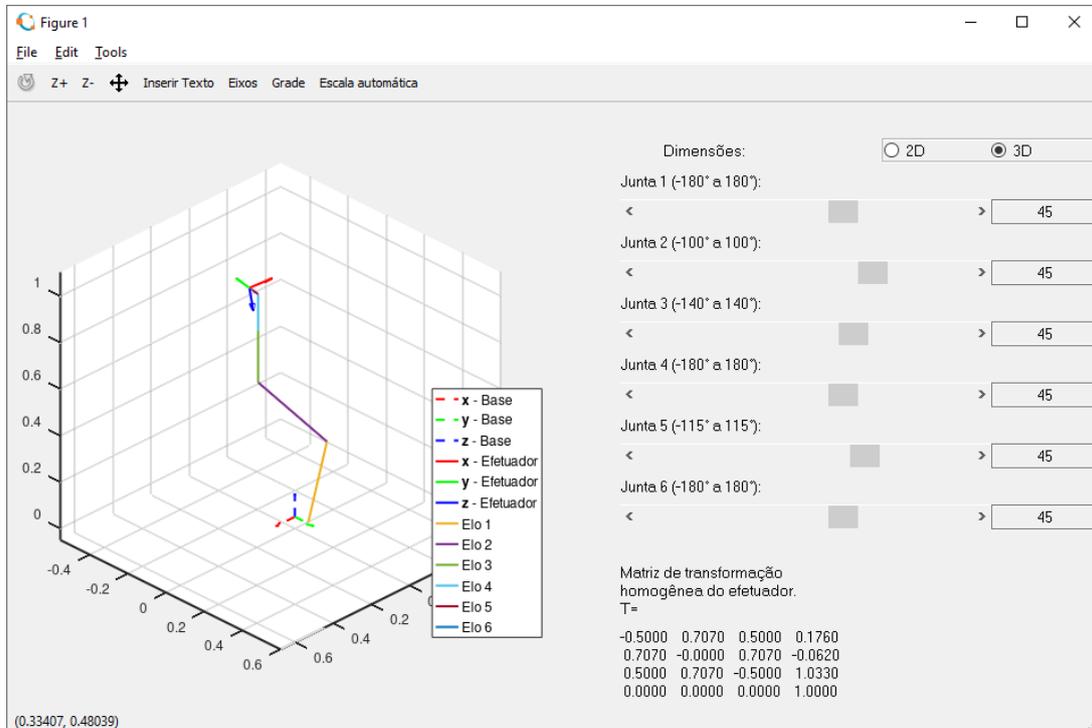
Para executar a interface gráfica no modo *online*, isto é, enviando comandos ao CoppeliaSim, é necessário iniciar a simulação no CoppeliaSim em “*Simulation/Start simulation*”. Logo em seguida, executar o código do Octave referente à interface gráfica, após atribuir o valor `true` à variável `h.online`.

No teste inicial, é marcado o controle em “3D” na interface gráfica atribuindo, por exemplo, 45° a todas as juntas, o que resulta na Figura 3.17. No segundo teste, com o controle em “2D”, foram atribuídos os valores, aleatoriamente, de -30° , 30° e -90° às juntas 2, 4 e 5, respectivamente, conforme ilustrado na Figura 3.18.

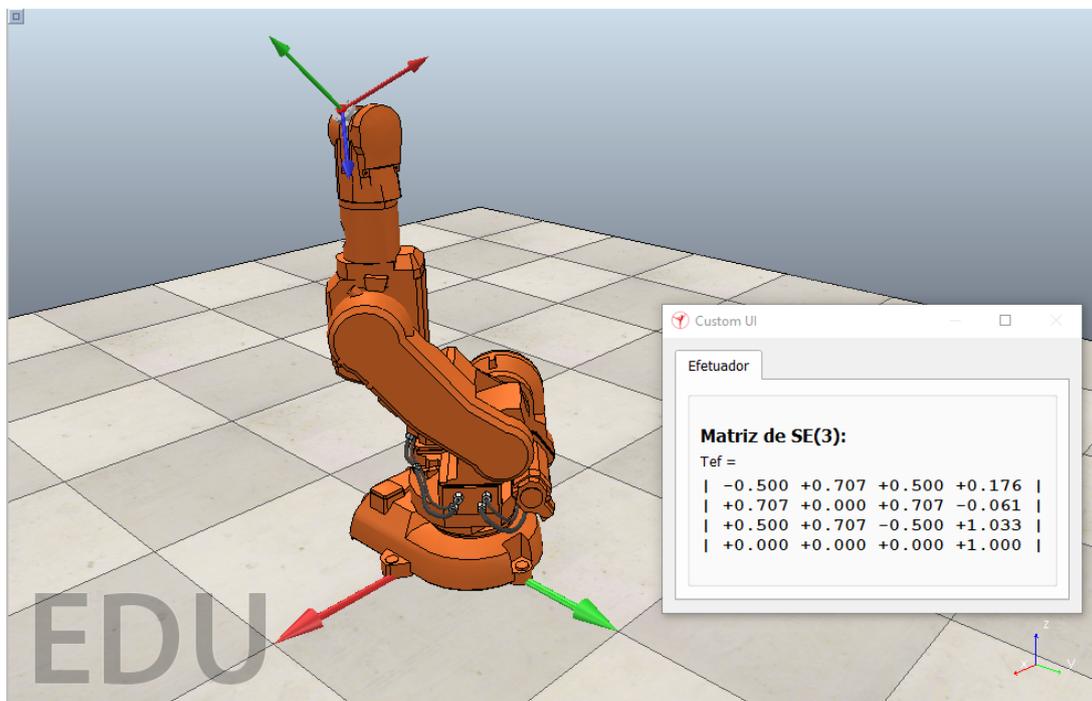
Comparando as matrizes de transformação obtidas nos dois testes, tanto do Octave quanto do CoppeliaSim, e, também, as demais matrizes obtidas em testes não registrados, foi possível verificar que o erro é muito baixo, na ordem de milésimos. As diferenças obtidas podem ser atribuídas ao fato do levantamento das coordenadas das juntas ter sido feito com apenas três casas decimais. Outro motivo pode ser os eixos de rotação das juntas do modelo do CoppeliaSim não serem paralelas aos do sistema de coordenadas e também pelo fato de a simulação alterar os valores com o passar do tempo, mesmo não havendo alterações nas juntas e na base. Além disso, um fator que deve ser levado em consideração é o fato de, no cálculo de raízes, senos e cossenos, o Octave trabalhar com número finito de casas decimais.

Figura 3.17: Teste *online* com interfaces, em três dimensões

(a) Teste *online* com interfaces em três dimensões no Octave



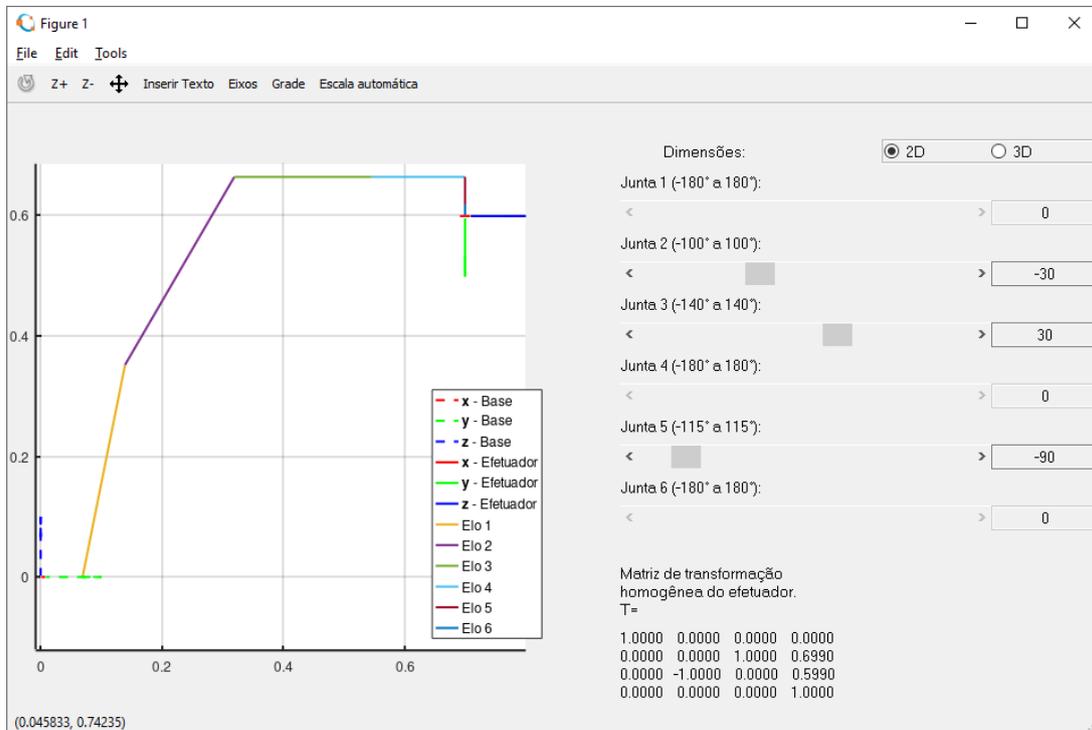
(b) Teste *online* com interfaces em três dimensões no CoppeliaSim



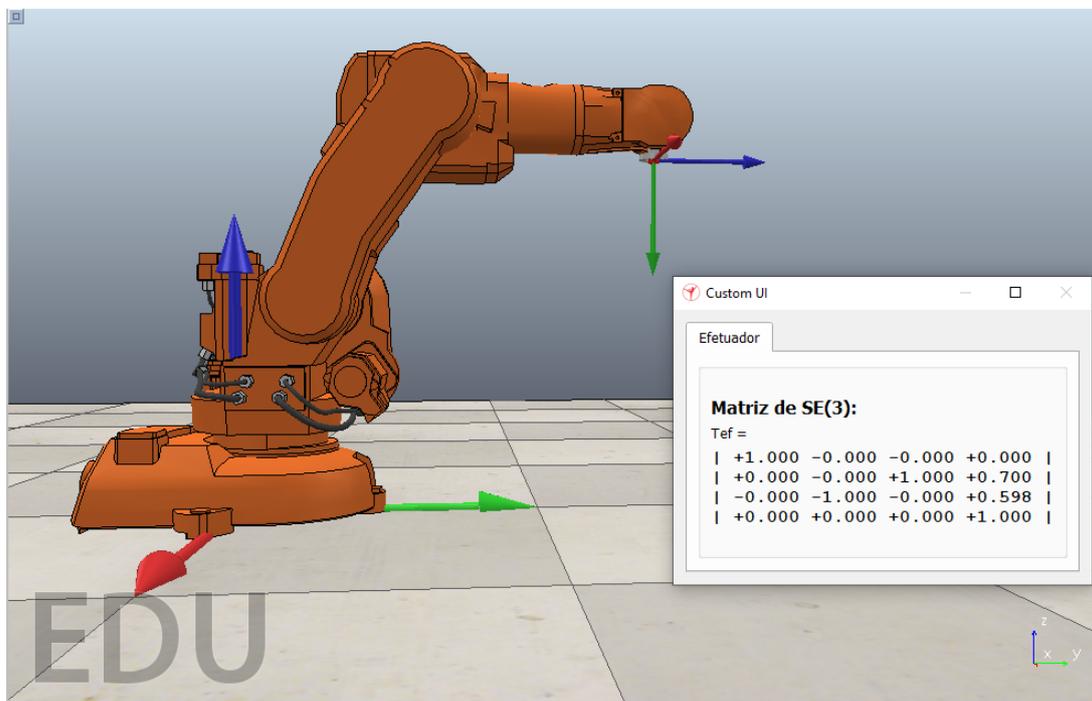
Fonte: Elaborado pelo autor.

Figura 3.18: Teste *online* com interfaces, em duas dimensões

(a) Teste *online* com interfaces em duas dimensões no Octave



(b) Teste *online* com interfaces em duas dimensões no CoppeliaSim



Fonte: Elaborado pelo autor.

4 Atividades para sala de aula

Neste capítulo é apresentado o produto final dessa dissertação. Trata-se de algumas atividades possíveis de serem aplicadas em sala de aula, utilizando o conteúdo exposto até então. Sua elaboração foi baseada nas exigências da Base Nacional Curricular Comum[10].

A lista não faz parte de uma sequência de didática, logo cada atividade pode ser aplicada individualmente, independente da ordem, conforme a intenção do professor. No entanto, como o grau de dificuldade e de aprofundamento no assunto é progressivo, o professor pode apresentar os exercícios de forma sequencial, o que contribui, inclusive, para o aumento do interesse do aluno no assunto.

As habilidades da BNCC[10, p. 543] utilizadas na elaboração da lista de atividade são as seguintes:

EM13MAT301 Resolver e elaborar problemas do cotidiano, da Matemática e de outras áreas do conhecimento, que envolvem equações lineares simultâneas, usando técnicas algébricas e gráficas, com ou sem apoio de tecnologias digitais.

EM13MAT405 Utilizar conceitos iniciais de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática.

EM13MAT105 Utilizar as noções de transformações isométricas (translação, reflexão, rotação e composições destas) e transformações homotéticas para construir figuras e analisar elementos da natureza e diferentes produções humanas (fractais, construções civis, obras de arte, entre outras).

EM13MAT308 Aplicar as relações métricas, incluindo as leis do seno e do cosseno ou as noções de congruência e semelhança, para resolver e elaborar problemas que envolvem triângulos, em variados contextos.

4.1 Atividade 1: Elipse transladada e rotacionada

Ano(s): 1° ao 3° anos do Ensino Médio

Habilidade(s) da BNCC: EM13MAT105 e EM13MAT308

Recursos: Quadro-negro e impressão da atividade (opcional)

Objetivo: Utilizar grupos matriciais para localizar vetores no plano

Descrição:

Quando é feito o estudo de cônicas em Geometria, como elipse e círculo, é comum se deparar com estas figuras transladadas em relação à origem ou rotacionadas em torno do seu centro. Por exemplo, a elipse construída pela Figura 4.1a e descrita pela equação

$$244x^2 - 384xy - 2368x + 356y^2 + 224y + 5899 = 0$$

é a elipse descrita pela Figura 4.1b e pela equação

$$4x^2 + 20y^2 = 125$$

após uma translação por um vetor $\vec{v} = (8, 4)$ e uma rotação em torno de seu centro por um ângulo $\alpha = 36,87^\circ$.

A atividade consiste em identificar as coordenadas dos pontos da lista abaixo, referentes à elipse da Figura 4.1b, na elipse transladada e rotacionada:

1. $A = (0, 2,5)$ - Vértice
2. $B = (3,35, -2)$
3. $C = (0, 0)$ - Centro
4. $F_1 = (-5, 0)$ - Foco 1
5. $F_2 = (-5, 0)$ - Foco 2

As coordenadas do ponto C são obtidas por meio da soma das coordenadas $(0, 0)$ do ponto com as coordenadas $(8, 4)$ do vetor. Espera-se que os alunos usem as funções seno e cosseno do ângulo para identificar as coordenadas dos pontos restantes. Após fazer alguns exercícios, será possível identificar uma repetição na forma em que as multiplicações são feitas.

O objetivo é demonstrar que as transformações geométricas de translação e rotação podem ser representadas por uma matriz de transformação homogênea pertencente a $SE(2)$.

A translação é dada pelo vetor $\mathbf{p} = \begin{bmatrix} 8 \\ 4 \end{bmatrix}$, pertencente a V^2 . A rotação, por sua vez, pode ser descrita pela matriz de rotação $R = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$, pertencente a $SO(2)$.

Sendo assim, a matriz pertencente a $SE(2)$ que descreve a transformação geométrica é:

$$T = \begin{bmatrix} \cos 36,87^\circ & -\sin 36,87^\circ & 8 \\ \sin 36,87^\circ & \cos 36,87^\circ & 4 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0,8 & -0,6 & 8 \\ 0,6 & 0,8 & 4 \\ 0 & 0 & 1 \end{bmatrix}$$

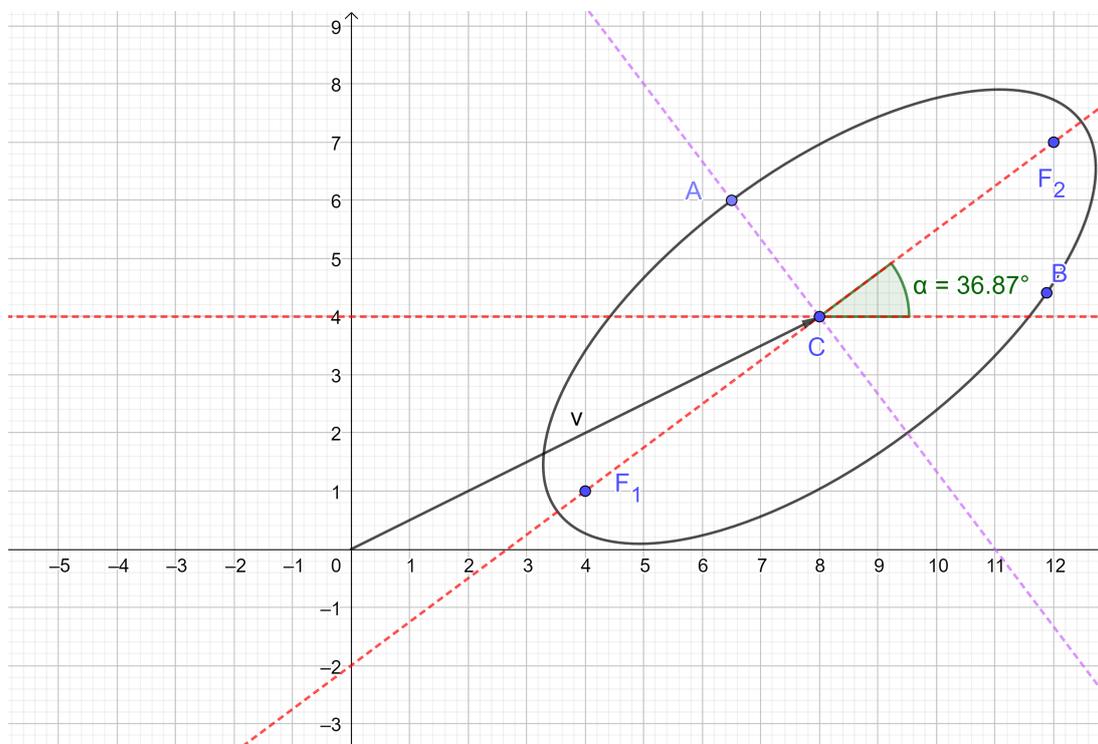
Para encontrar a coordenada dos pontos após a transformação, basta considerar os pontos como vetores com centro na origem, escrevê-los como vetores coluna adicionando o valor 1, e multiplicar à esquerda por T . Resolvendo a primeira questão é possível obter:

$$\mathbf{a}' = T\mathbf{a} = \begin{bmatrix} 0,8 & -0,6 & 8 \\ 0,6 & 0,8 & 4 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2,5 \\ 1 \end{bmatrix} = \begin{bmatrix} 6,5 \\ 6 \\ 1 \end{bmatrix}$$

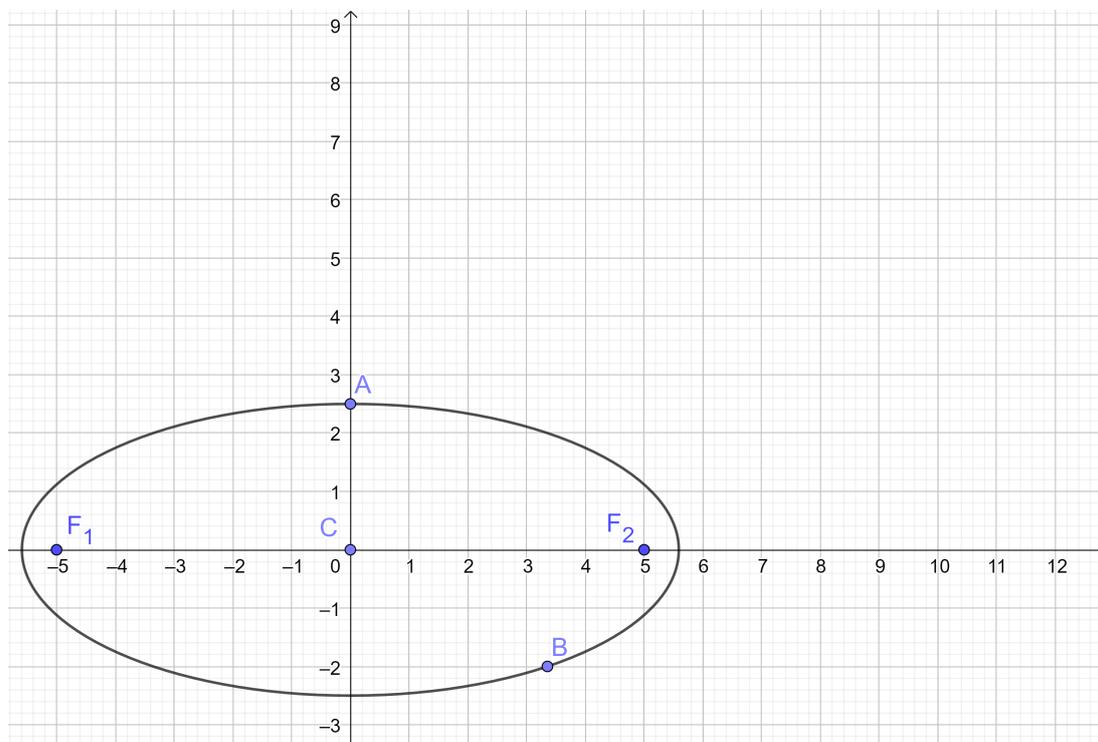
Dessa forma, sabe-se que as coordenadas do ponto A após a transformação geométrica são $(6,5, 6)$.

Figura 4.1: Isometria aplicada a elipse

(a) Elipse transladada e rotacionada



(b) Elipse centrada na origem e com eixos paralelos aos do sistema cartesiano



Fonte: Elaborado pelo autor.

4.2 Atividade 2: Cinemática direta no plano

Ano(s): 1° ao 3° anos do Ensino Médio

Habilidade(s) da BNCC: EM13MAT301, EM13MAT105 e EM13MAT308

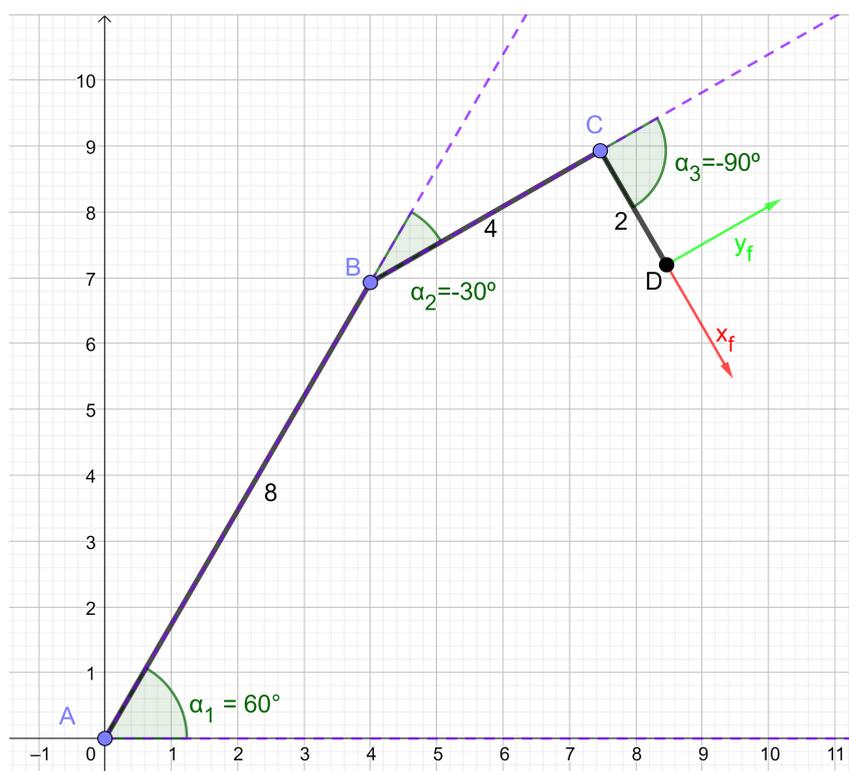
Recursos: Quadro-negro e impressão da atividade (opcional)

Objetivo: Descrever pontos e vetores no plano por meio de grupos matriciais

Descrição:

A Figura 4.2 exibe os pontos A, B, C e D no plano cartesiano. A atividade proposta envolve a descrição da localização do ponto D e o ângulo entre o vetor \vec{x}_f e o eixo \vec{x} do sistema de referência. O ponto A está localizado na origem e os comprimentos dos segmentos de reta e os ângulos relevantes para o levantamento estão descritos na figura.

Figura 4.2: Descrevendo as coordenadas do ponto D e o ângulo entre o vetor \vec{x}_f e o eixo \vec{x}



Fonte: Elaborado pelo autor.

Espera-se que os alunos utilizem propriedades geométricas, como Teorema de Pitágoras, Teorema de Tales, seno e cosseno, além de ângulos congruentes, complementares e suplementares, para resolver este problema.

Por meio de Geometria Algébrica é possível escrever as coordenadas (x_D, y_D) do

ponto D em função dos ângulos, obtendo, então:

$$x_D = 8 \cos \alpha_1 + 4 \cos (\alpha_1 + \alpha_2) + 2 \cos (\alpha_1 + \alpha_2 + \alpha_3)$$

$$y_D = 8 \sin \alpha_1 + 4 \sin (\alpha_1 + \alpha_2) + 2 \sin (\alpha_1 + \alpha_2 + \alpha_3)$$

O ângulo β , que é entre o vetor \vec{x}_f e o eixo \vec{x} , pode ser descrito por:

$$\beta = \alpha_1 + \alpha_2 + \alpha_3$$

No entanto, pode-se interpretar a Figura 4.2 como um braço robótico, descrevendo-o por meio de matrizes pertencentes a SE (2). Os pontos A, B e C descrevem as posições das juntas, sendo o ponto A também a base do braço, os segmentos AB, BC e CD descrevem os elos e o ponto D representa o efetuador, o qual terá sua posição e orientação a serem descobertos por meio da cinemática direta. Dessa forma, a matriz T que descreve o sistema é a seguinte:

$$T = \begin{bmatrix} \cos \alpha_1 & -\sin \alpha_1 & 0 \\ \sin \alpha_1 & \cos \alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha_2 & -\sin \alpha_2 & 8 \\ \sin \alpha_2 & \cos \alpha_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \alpha_3 & -\sin \alpha_3 & 4 \\ \sin \alpha_3 & \cos \alpha_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

O resultado deste produto é a matriz de transformação homogênea que descreve o efetuador do braço robótico, e é da forma:

$$T = \begin{bmatrix} \cos \beta & -\sin \beta & x_D \\ \sin \beta & \cos \beta & y_D \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

4.3 Atividade 3: Simulação computacional de braço robótico em duas dimensões

Ano(s): 1° ao 3° anos do Ensino Médio

Habilidade(s) da BNCC: EM13MAT301, EM13MAT405, EM13MAT105 e EM13MAT308

Recursos: Quadro-negro, impressão da atividade (opcional) e computador com o *software* Octave instalado

Objetivo: Fazer uso de recursos computacionais para simular braço robótico, utilizando grupos matriciais

Descrição:

Essa atividade consiste em simular o braço robótico descrito na atividade anterior no ambiente de programação do Octave. Para isso, utiliza-se o Editor do *software*, onde é inserido o código.

Inicialmente, faz-se necessário declarar as variáveis das juntas, que são ângulos em graus. O valor atribuído inicialmente é:

```
j1=0; j2=0; j3=0;
```

Em seguida são declaradas as quatro matrizes, conforme Equação 4.1. A primeira matriz, por exemplo, é da seguinte forma:

```
##Matriz 1
T(:, :, 1)=[cosd(j1), -sind(j1), 0;
            sind(j1), cosd(j1), 0;
            0, 0, 1];
```

Para multiplicar as matrizes, a fim de obter o resultado com a matriz de transformação homogênea do efetuador, é utilizado o código a seguir, que realiza um laço como o do exemplo da Seção 3.2.6.

```
T0=eye(3);
Tef=T0;
for ind=1:4
    Tef=Tef*T(:, :, ind);
```

```

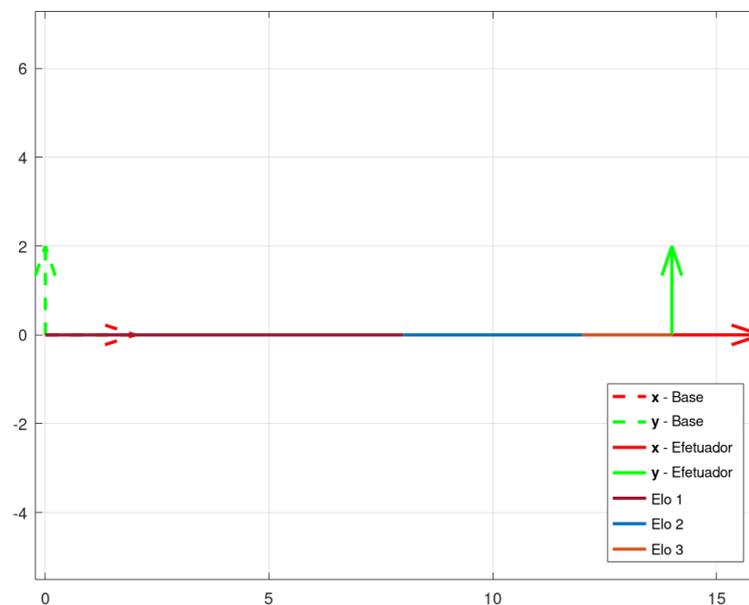
pef(:, ind) = Tef(1:3, 4);
end;

```

O código completo da simulação, que possibilita não somente a declaração das variáveis, mas também a plotagem da representação do braço robótico, está disponível no Anexo A.2.

Ao executar o código completo, atribuindo o valor de 0° para as juntas, obtém-se o resultado exibido na Figura 4.3.

Figura 4.3: Braço robótico no plano com juntas em estado inicial



Fonte: Elaborado pelo autor.

A matriz de transformação homogênea exibida nessa simulação é facilmente entendida, pois a submatriz de rotação é I_2 , o que demonstra a mesma orientação da base, e o vetor coluna de translação possui coordenadas $(14, 0)$, que é a soma dos elos em direção ao eixo \vec{x} , totalizando $8 + 4 + 2 = 14$. A matriz é a seguinte:

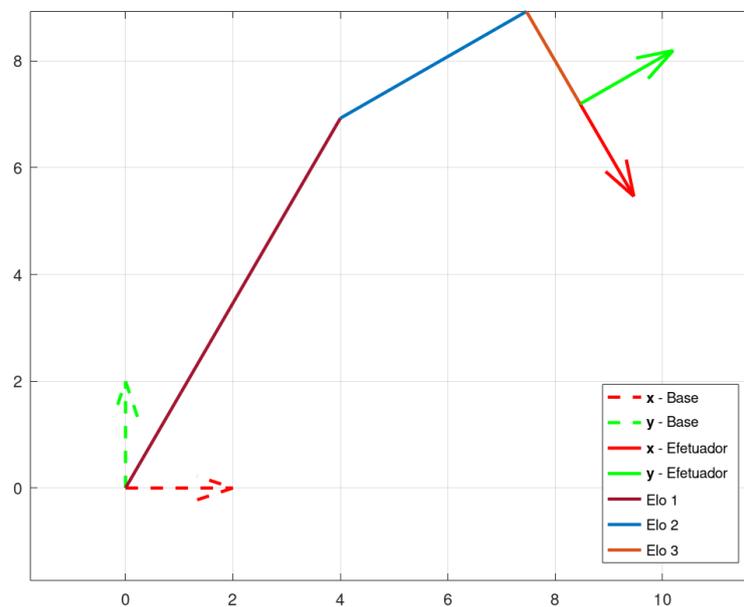
$$\begin{aligned}
T_{ef} &= \\
&\begin{bmatrix} 1 & 0 & 14 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

Em seguida, atribui-se os valores do exemplo da atividade anterior às juntas 1, 2 e

3, isto é, 60, -30 e -90 , respectivamente. Ao se executar o código, é exibida a Figura 4.4, semelhante à Figura 4.2, e a matriz de transformação homogênea é a seguinte:

$$T_{ef} = \begin{pmatrix} 0.50000 & 0.86603 & 8.46410 \\ -0.86603 & 0.50000 & 7.19615 \\ 0.00000 & 0.00000 & 1.00000 \end{pmatrix}$$

Figura 4.4: Braço robótico no plano com juntas alteradas



Fonte: Elaborado pelo autor.

Ao comparar o resultado com a matriz descrita na Equação 4.2, tem-se que as coordenadas do ponto D são $(8,4641 \ 7,19615)$ e o ângulo β , que é o entre o vetor \vec{x}_f e o eixo \vec{x} , é dado por:

$$\begin{cases} \cos \beta = 0,5 \\ \text{sen } \beta = -0,866 \end{cases} \Rightarrow \beta = -60^\circ$$

4.4 Atividade 4: Utilização de interface gráfica do Octave

Ano(s): 1° ao 3° anos do Ensino Médio

Habilidade(s) da BNCC: EM13MAT301, EM13MAT405, EM13MAT105 e EM13MAT308

Recursos: Quadro-negro, impressão da atividade (opcional) e computador com os *softwares* Octave e CoppeliaSim (opcional) instalados, e, de preferência, com todos os arquivos de simulação disponíveis

Objetivo: Utilizar interface gráfica do *software* Octave para assimilar os conceitos de grupos e interpretar braços robóticos simulados

Descrição:

Esta atividade consiste em utilizar a interface apresentada na Seção 3.2.7.

Devido à alta complexidade, sugere-se que a aula seja preparada com certa antecedência, para que o professor tenha desenvoltura suficiente nos *softwares* e seja capaz de responder a eventuais questionamentos dos alunos.

O CoppeliaSim, nessa atividade, é utilizado apenas como representação do braço robótico e pode ser dispensado, caso o professor opte por isso. A desvantagem de não utilizá-lo está no fato de que a interpretação do gráfico da interface como braço robótico exigirá uma abstração maior dos alunos. Caso não seja usado, é necessário alterar o início do código “create_gui” da interface gráfica, de forma a deixar seguinte linha de comando:

```
h.online=false; #true=conectado; false=desconectado
```

Em sala de aula, o professor deve apresentar inicialmente as partes principais do braço robótico estudado, ressaltando o movimento de cada uma de suas seis juntas. É imprescindível que os alunos entendam o significado de base, juntas, elos e efetuador para continuar a atividade.

Ao executar o código da interface gráfica de usuário, conforme explicado nos passos do Anexo C, uma janela da interface é aberta, como pode ser observado na Figura 3.14. Nessa janela, os alunos têm a opção de alterar o valor do ângulo das juntas, de forma a aumentar sua percepção de que o gráfico representa o braço robótico.

A matriz de transformação homogênea apresentada na interface, bem como no

CoppeliaSim, é da forma:

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

O aluno deve conseguir localizar a submatriz de rotação e a submatriz (vetor coluna) de translação. As coordenadas (x, y, z) do vetor de translação representam as coordenadas do efetuador do braço robótico.

Com o controle em “2D” selecionado, o aluno deve alterar os valores das juntas e observar sua mudança no gráfico. Após entender o significado dos elementos do gráfico, este deve responder as seguintes perguntas:

1. Qual é o valor dos ângulos das juntas para que o braço robótico atinja a posição mais alta verticalmente? E a posição mais esticada horizontalmente?
2. Qual é a condição dos valores dos ângulos das juntas para que o eixo \vec{y} (verde) do efetuador esteja alinhado com eixo \vec{y} da base? E qual a condição para que o eixo \vec{y} do efetuador esteja alinhado com eixo $-\vec{z}$ (oposto ao azul) da base?
3. Se a coordenada z é negativa, o que isso representa no braço robótico?
4. O que o valor nulo da coordenada x representa? O que o valor $r_{11} = 1$ representa?
5. Caso esteja usando o CoppeliaSim, explicar por que existem posições de juntas que não são alcançáveis pelo braço robótico do simulador.

A resposta da primeira pergunta deve levar em consideração que a posição mais alta na vertical é obtida quando a coordenada z do vetor de translação está com seu valor máximo, e na horizontal quando a coordenada y está com valor absoluto máximo.

O valor nulo de x representa que o braço está situado no plano OYZ . O valor $r_{11} = 1$ representa que o vetor \vec{x} do efetuador, mesmo omitido, está alinhado com o vetor \vec{x} da base.

A última pergunta expõe o problema das restrições. É necessário explicar que o braço robótico possui um ambiente de trabalho e existem posições que não são alcançadas pelo efetuador.

Alterando o controle para “3D”, é possível verificar que as juntas 1, 4 e 6 são habilitadas, e o ângulo de vista do braço robótico é alterado. Antes de alterar quaisquer ângulos, deve-se responder as seguintes questões:

6. É possível interpretar o braço robótico em um plano ao se alterar apenas o valor dos ângulos das juntas 2, 3 e 5, deixando as demais com o valor zero?
7. Qual é o valor dos ângulos das juntas para que o braço robótico atinja a posição mais alta verticalmente? O valor de quais juntas não alteram o resultado?
8. Qual é o valor das juntas para que o braço robótico fique mais esticado horizontalmente, paralelo ao plano da base? Quais juntas não alteram o resultado?
9. Caso esteja usando o CoppeliaSim, quais as juntas não possuem limite de atuação (rotacionam livremente)?

A pergunta número 7 é respondida de forma similar à primeira parte da pergunta 2. No entanto, a pergunta 8 possui uma outra interpretação uma vez que a junta 1, que rotaciona o braço em torno da base, não altera o quanto o braço está esticado horizontalmente. Dessa forma, a condição é satisfeita quando $\sqrt{x^2 + y^2}$ está com valor máximo.

Ao responder a última questão, deve-se fazer um paralelo com o braço humano, que possui restrições nas “juntas”.

5 Considerações finais

A principal motivação ao elaborar essa dissertação foi abordar conceitos matemáticos em sua forma aplicada, produzindo um trabalho que enriquecesse o processo de ensino-aprendizagem de Matemática e apresentasse indicações de práticas pedagógicas em um assunto interessante e relevante. O estudo da cinemática direta dos braços robóticos se mostrou uma boa opção de aplicação. Para tal, foi necessário realizar uma revisão bibliográfica aprofundada no assunto, não somente em fontes da área de Matemática para o estudo da Geometria e da Álgebra envolvidas, mas também em livros de engenharia, no que tange à Robótica.

O estudo da Geometria Analítica na descrição dos elementos do braço robótico permitiu interpretar segmentos de reta e ângulos de uma maneira diferente. A intenção ao abordar os conceitos de Álgebra Linear nesse trabalho, por sua vez, foi de apresentar uma abordagem que dá uma outra interpretação às operações elementares com matrizes, como realizar transformações geométricas a objetos.

A representação de braços robóticos utilizando matrizes de transformação homogênea se mostrou uma opção adequada, pois faz uso de conceitos algébricos de grupos. Essas estruturas, com suas propriedades, garantem que a representação do mecanismo se dê de forma simples, precisa e facilmente alterável. Cada fator de um produto de matrizes representa um ou dois elementos do braço robótico e, como o resultado desse produto também é uma matriz com a mesma característica, a localização do efetuador é claramente interpretada.

O avanço cada vez mais intenso de tecnologias e o desafio de inovação no ensino de Matemática motivaram a aplicação dos conceitos geométricos e algébricos por meio da computação. O ambiente de programação apresentado se mostrou uma ferramenta eficaz para a manipulação de matrizes e para a elaboração de representações gráficas. O simulador de Robótica, por sua vez, possibilitou a interpretação dos dados obtidos em animações, bem como na assimilação do que cada translação e rotação significam nesse

estudo. O fato dos *softwares* serem disponibilizados de forma gratuita fez com que sua escolha se tornasse atrativa. Por mais que os erros obtidos na simulação, na ordem de milésimos, possam ser significativos no ambiente industrial, como estes ocorreram por causas alheias às práticas adotadas, o resultado pode ser considerado bem-sucedido para a análise acadêmica.

A lista de atividades apresentada não é exaustiva. Trata-se de sugestões a serem avaliadas pelos docentes e, caso sejam adequadas ao ensino dado no momento, podem ser administradas de modo a ter uma aplicação no ambiente de sala de aula. O professor é encorajado a usar sua criatividade e ir além, produzindo atividades inspiradas no conteúdo apresentado nesse trabalho. Não havendo impacto no conteúdo básico a ser exposto aos alunos, pode-se, inclusive, realizar a abordagem em três dimensões, fazendo o uso, não somente da linguagem de programação proposta, mas também do simulador de robótica. Caso a escola possua braços robóticos reais, as atividades práticas podem ser estendidas da sala de aula para os laboratórios.

O assunto Robótica não se detém ao ambiente acadêmico. O crescente avanço de tecnologias no ambiente industrial tem exigido profissionais cada vez mais qualificados, que possuam conhecimento nas mais diversas áreas. A aplicação desse conteúdo de Robótica pode aguçar a curiosidade dos alunos, promover um ambiente acadêmico desafiador e lúdico, além de ressaltar possíveis aptidões por essa área, o que refletirá em um maior anseio pelo estudo de disciplinas correlatas.

O advento da pandemia da doença COVID-19, a qual alcançou o Brasil em 2020, tornou o cenário da educação desafiador em todos os sentidos. O Ministério da Educação e os professores tiveram que se reinventar para poder ministrar algum conteúdo aos alunos. Nesta dissertação o impacto foi maior no que tange à validação das atividades propostas em sala de aula, uma vez que não foi possível encontrar ambientes com tempo suficiente para a administração do conteúdo. Espera-se, então, que a lista seja utilizada em outro momento por algum docente e os resultados compartilhados para futuras consultas. Uma opção de estudo de caso a ser realizada também é a análise de um braço robótico real, reproduzindo-o virtualmente.

Na área teórica, sugere-se a abordagem da cinemática inversa, que é o estudo das posições de juntas necessárias para que o efetuador atinja uma posição e orientação específicos. Pode-se, inclusive, abordar a dinâmica envolvida, estudando forças, torques e

demais conceitos de Física relacionados, considerando as restrições que acontecem com o choque de peças do próprio equipamento.

Tendo em vista a proposta inicial do trabalho, o objetivo do programa de mestrado e a aplicação do conteúdo absorvido pelo autor, entende-se que esse trabalho cumpriu seus objetivos.

Referências

- 1 OECD. *PISA 2018 country-specific overviews: Brazil*. 2018. Disponível em: <https://www.oecd.org/pisa/publications/PISA2018_CN_BRA.pdf>. Acesso em: 25-01-2021.
- 2 BRASIL: MINISTÉRIO DA EDUCAÇÃO. *Brasil no Pisa 2018 (Relatório nacional)*. 2019. Disponível em: <http://download.inep.gov.br/acoes_internacionais/pisa-/documentos/2019/relatorio_PISA_2018_preliminar.pdf>. Acesso em: 25-01-2021.
- 3 SOCIEDADE BRASILEIRA DE MATEMÁTICA. *10 Questões para Professores de Matemática... e como o PISA Pode Ajudar a Respondê-las*. OCDE/IMPA, 2018. 104 p. Disponível em: <<https://www.sbm.org.br/destaque/10-questoes-para-professores-de-matematica>>. Acesso em: 25-01-2021.
- 4 ANDRADE, C. C. de. *O ensino da matemática para o cotidiano*. Trabalho de Conclusão de Curso (Especialização) — Universidade Tecnológica Federal do Paraná, Medianeira, 2013. Disponível em: <<http://repositorio.roca.utfpr.edu.br/jspui/handle/1/4286>>. Acesso em: 25-01-2021.
- 5 BRITO, R. S.; MOITA, F. M. G. da S. C.; LOPES, M. da C. Robótica educacional: Desafios e possibilidades no trabalho interdisciplinar entre matemática e física. *Ensino da Matemática em Debate*, v. 5, n. 1, p. 27–44, 2018. Disponível em: <<https://revistas.pucsp.br/index.php/emd/article/view/36687>>. Acesso em: 25-01-2021.
- 6 PERALTA, D.; GUIMARÃES, E. A robótica na escola como postura pedagógica interdisciplinar: o futuro chegou para a educação básica? *Revista Brasileira de Informática na Educação*, v. 26, p. 30, 01 2018.
- 7 SOCIEDADE BRASILEIRA DE MATEMÁTICA. *PROFMAT: Uma reflexão e alguns resultados*. 2017. Disponível em: <https://www.profmtat-sbm.org.br/wp-content/uploads/sites/23/2017/07/PROFMAT-relatorio_DIGITAL.pdf>. Acesso em: 25-01-2021.
- 8 GNU OCTAVE. *GNU OCTAVE: About*. 2020. Disponível em: <<https://www.gnu.org/software/octave/about.html>>. Acesso em: 25-01-2021.
- 9 COPPELIA ROBOTICS. *CoppeliaSim User Manual: User interface*. 2020. Disponível em: <<https://www.coppeliarobotics.com/helpFiles/index.html>>. Acesso em: 25-01-2021.
- 10 BRASIL: MINISTÉRIO DA EDUCAÇÃO. *Base Nacional Comum Curricular (BNCC)*. 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf>. Acesso em: 25-01-2021.

- 11 CRISTOVÃO, E. M.; SPILLER, L. K. P. C. *Matrizes, determinantes e sistemas ou sistemas, matrizes e determinantes?* 2009. Disponível em: <http://alb.org.br/arquivo-morto/edicoes_anteriores/anais17/txtcompletos/sem07/COLE_4002.pdf>. Acesso em: 25-01-2021.
- 12 BRASIL: MINISTÉRIO DA EDUCAÇÃO. *Parâmetros Curriculares Nacionais (PCN): Matemática / Secretaria de Educação Fundamental – Brasília: MEC/SEF.* 1997. Disponível em: <<http://portal.mec.gov.br/seb/arquivos/pdf/livro03.pdf>>. Acesso em: 25-01-2021.
- 13 COSTA, C. G. da. *Utilização de matrizes no estudo de orientação e posição de um braço robótico por meio das coordenadas de Denavit-Hartenberg.* Dissertação (Mestrado) — Universidade Federal de Goiás, 2014. Disponível em: <https://sca.proformat-sbm.org.br/sca_v2/get_tcc3.php?id=61463>. Acesso em: 25-01-2021.
- 14 MESQUITA, L. C. *Transformações Ortogonais Aplicada à Cinemática de Robôs.* Dissertação (Mestrado) — Universidade Federal do Piauí, 2016. Disponível em: <https://sca.proformat-sbm.org.br/sca_v2/get_tcc3.php?id=94902>. Acesso em: 25-01-2021.
- 15 CARVALHO, R. N. de. *Ensino de matemática através da robótica: movimento do braço mecânico.* Dissertação (Mestrado) — Fundação Universidade Federal de Rondônia, 2013. Disponível em: <https://sca.proformat-sbm.org.br/sca_v2/get_tcc3.php?id=51154>. Acesso em: 25-01-2021.
- 16 NETO, J. R. da S. *A utilização do software Octave na interpretação geométrica das operações com matrizes no ensino médio.* Dissertação (Mestrado) — Universidade Federal Rural do Semi-Árido, 2019. Disponível em: <https://sca.proformat-sbm.org.br/sca_v2/get_tcc3.php?id=160252002>. Acesso em: 25-01-2021.
- 17 NUNES, D. M. *O uso da linguagem de programação de computadores no ensino de matemática: Alternativa metodológica para uma integração disciplinar.* Dissertação (Mestrado) — Universidade Estadual do Sudoeste da Bahia, 2016. Disponível em: <https://sca.proformat-sbm.org.br/sca_v2/get_tcc3.php?id=94325>. Acesso em: 25-01-2021.
- 18 ROUSSEAU, C.; SAINT-ABIN, Y. *Matemática e Atualidade.* 1ª. ed. Rio de Janeiro: SBM, 2015. v. 5. 326 p. (Coleção Profmat, v. 5). ISBN 978-85-8337-031-4.
- 19 BOULOS, P.; CAMARGO, I. de. *Geometria Analítica: um tratamento vetorial.* 2. ed. São Paulo: Makron Books do Brasil, 1987. 385 p.
- 20 LYNCH, K. M.; PARK, F. C. *Modern Robotics: Mechanics, Planning, and Control.* Cambridge, Reino Unido: Cambridge University Press, 2017. v. 2. 544 p. ISBN 9781107156302.
- 21 STEWART, J. *Cálculo Volume 2: Tradução da 7ª edição norte-americana.* São Paulo: Cengage Learning, 2013. ISBN 978-85-221-1463-4.
- 22 COPPELIA ROBOTICS. *Euler Angles.* 2020. Disponível em: <<https://www.coppeliarobotics.com/helpFiles/en/eulerAngles.htm>>. Acesso em: 25-01-2021.
- 23 MIHELJ, M. et al. *Robotics.* Cham, Suíça: Springer International Publishing, 2019. ISBN 9783319729114.

-
- 24 ABB. *IRB 140. Small, powerful, and fast 6-axes robot*. 2019. Disponível em: <<https://new.abb.com/products/robotics/industrial-robots/irb-140>>. Acesso em: 25-01-2021.
- 25 DENAVIT, J.; HARTENBERG, R. S. A kinematic notation for lower pair mechanisms based on matrices. *Journal of applied mechanics*, v. 77, n. 2, p. 215–221, 1955.
- 26 SICILIANO, B. et al. *Robotics: Modelling, Planning and Control*. 1. ed. Londres: Springer-Verlag London, 2009. 632 p. (Advanced Textbooks in Control and Signal Processing). ISBN 978-1-84628-642-1.

Anexo A

Algoritmos do Octave

Este anexo exhibe os códigos do Octave utilizados ao longo dessa dissertação. Eles foram apresentados no trabalho utilizando a mesma fonte padrão do ambiente de programação.

A utilização deste código exige a instalação do ambiente de programação na máquina e um conhecimento mínimo da linguagem de programação empregada. Todos os códigos foram escritos no Editor de Texto do Octave e salvos em uma pasta do sistema, utilizando o sistema operacional *Microsoft Windows 10*, em um computador com 8 gigabytes de memória RAM e processador com 2,5 gigahertz de frequência. A versão do Octave utilizada é a 5.2.0.

A Seção [A.1](#) deste anexo exhibe o código utilizado na Subseção [3.2.6](#) do Capítulo [3](#). A Seção [A.2](#), por sua vez, contém o código utilizado na atividade presente na Seção [4.3](#), no Capítulo [4](#).

A Seção [A.3](#) apresenta o código utilizado para criar a interface gráfica de usuário, que foi aplicada na Subseção [3.2.7](#). As funções que são chamadas por esse código estão nas Seções [A.4](#), [A.5](#) e [A.6](#). A etapa para a criação da interface, bem como a descrição dessas funções, está contida no Anexo [C](#).

A.1 Simulação de robô 3D no Octave

```
## Código para plotar o braço robótico em 3D
## Samuel Duque Gonçalves - 2020
close all;
clear all;
clc;
##Iniciar as juntas atribuindo valor 0
j1=0; j2=0; j3=0; j4=0; j5=0; j6=0;

##Matriz 1
T(:, :, 1)=[cosd(j1), -sind(j1), 0, 0;
            sind(j1), cosd(j1), 0, 0.069;
            0, 0, 1, 0;
            0, 0, 0, 1];

##Matriz 2
T(:, :, 2)=[1, 0, 0, 0;
            0, cosd(j2), -sind(j2), .07;
```

```
0, sind(j2), cosd(j2), .3521;
0, 0, 0, 1];

##Matriz 3
T(:, :, 3)=[1, 0, 0, 0;
0, cosd(j3), -sind(j3), 0;
0, sind(j3), cosd(j3), .36;
0, 0, 0, 1];

##Matriz 4
T(:, :, 4)=[cosd(j4), 0, sind(j4), 0;
0, 1, 0, .226;
-sind(j4), 0, cosd(j4), 0;
0, 0, 0, 1];

##Matriz 5
T(:, :, 5)=[1, 0, 0, 0;
0, cosd(j5), -sind(j5), .154;
0, sind(j5), cosd(j5), 0;
0, 0, 0, 1];

##Matriz 6
T(:, :, 6)=[cosd(j6), 0, sind(j6), 0;
0, 1, 0, 0.046;
-sind(j6), 0, cosd(j6), 0;
0, 0, 0, 1];

##Matriz 7
T(:, :, 7)=[1, 0, 0, 0;
0, 1, 0, .019;
0, 0, 1, 0;
0, 0, 0, 1];

##Multiplicar as matrizes
T0=eye(4);
Tef=T0;
for ind=1:7
    Tef=Tef*T(:, :, ind);
    pef(:, ind)=Tef(1:3, 4);
end;

#Plotar os vetores e suas legendas
origx = quiver3(0, 0, 0, 0.1*T0(1, 1), 0.1*T0(2, 1), ...
    0.1*T0(3, 1), 'Color', 'red', ...
    'linestyle', '--', 'linewidth', 1);
hold on #Mantem os plots na mesma figura
```

```

origx_line = line([0,0.1*T0(1,1)], [0,0.1*T0(2,1)], ...
  [0,0.1*T0(3,1)], 'Color', 'red', 'linestyle', '--', ...
  "visible", "off", "displayname", '\bfx\rm - Base');

origy = quiver3(0,0,0,0.1*T0(1,2),0.1*T0(2,2), ...
  0.1*T0(3,2), 'Color', 'green', ...
  'linestyle', '--', "linewidth", 1);
origy_line = line([0,0.1*T0(1,2)], [0,0.1*T0(2,2)], ...
  [0,0.1*T0(3,2)], 'Color', 'green', 'linestyle', '--', ...
  "visible", "off", "displayname", '\bfy\rm - Base');

origz = quiver3(0,0,0,0.1*T0(1,3),0.1*T0(2,3), ...
  0.1*T0(3,3), 'Color', 'blue', ...
  'linestyle', '--', "linewidth", 1);
origz_line = line([0,0.1*T0(1,3)], [0,0.1*T0(2,3)], ...
  [0,0.1*T0(3,3)], 'Color', 'blue', 'linestyle', '--', ...
  "visible", "off", "displayname", '\bfz\rm - Base');

efx = quiver3(pef(1,ind),pef(2,ind),pef(3,ind), ...
  0.1*Tef(1,1),0.1*Tef(2,1),0.1*Tef(3,1), ...
  'Color', 'red', "linewidth", 1);
efx_line = line([pef(1,ind),0.1*Tef(1,1)], ...
  [pef(2,ind),0.1*Tef(2,1)], [pef(3,ind),0.1*Tef(3,1)], ...
  'Color', 'red', 'linestyle', '--', ...
  "visible", "off", "displayname", '\bfx\rm - Efetuador');

efy = quiver3(pef(1,ind),pef(2,ind),pef(3,ind), ...
  0.1*Tef(1,2),0.1*Tef(2,2),0.1*Tef(3,2), ...
  'Color', 'green', "linewidth", 1);
efy_line = line([pef(1,ind),0.1*Tef(1,2)], ...
  [pef(2,ind),0.1*Tef(2,2)], [pef(3,ind),0.1*Tef(3,2)], ...
  'Color', 'green', 'linestyle', '--', ...
  "visible", "off", "displayname", '\bfy\rm - Efetuador');

efz = quiver3(pef(1,ind),pef(2,ind),pef(3,ind), ...
  0.1*Tef(1,3),0.1*Tef(2,3),0.1*Tef(3,3), ...
  'Color', 'blue', "linewidth", 1);
efz_line = line([pef(1,ind),0.1*Tef(1,3)], ...
  [pef(2,ind),0.1*Tef(2,3)], [pef(3,ind),0.1*Tef(3,3)], ...
  'Color', 'blue', 'linestyle', '--', ...
  "visible", "off", "displayname", '\bfz\rm - Efetuador');

#Plotar os elos como linhas
for ind=2:7
  line([pef(1,ind-1),pef(1,ind)], ...
  [pef(2,ind-1),pef(2,ind)], [pef(3,ind-1),pef(3,ind)], ...

```

```
    "linewidth",2,"displayname",['Elo ',num2str(ind-1)]);  
end;  
  
hold off;  
  
#Configurar legenda, eixos e fontes  
axis equal;  
set(gca, "linewidth", 2, "fontsize", 12);  
leg=legend ("location", "southeast");  
set(leg, "fontsize", 12);
```

A.2 Simulação de robô 2D no Octave

```
## Código para plotar o braço robótico em 2D  
## Samuel Duque Gonçalves - 2020  
close all;  
clear all;  
clc;  
  
##Declaração das juntas do braço  
##Inicialmente atribuído o valor 0  
j1=0; j2=0; j3=0;  
  
##Matriz 1  
T(:, :, 1)=[cosd(j1), -sind(j1), 0;  
    sind(j1), cosd(j1), 0;  
    0, 0, 1];  
  
##Matriz 2  
T(:, :, 2)=[cosd(j2), -sind(j2), 8;  
    sind(j2), cosd(j2), 0;  
    0, 0, 1];  
  
##Matriz 3  
T(:, :, 3)=[cosd(j3), -sind(j3), 4;  
    sind(j3), cosd(j3), 0;  
    0, 0, 1];  
  
##Matriz 4  
T(:, :, 4)=[1, 0, 2;  
    0, 1, 0;  
    0, 0, 1];  
  
##Multiplicar as matrizes  
T0=eye(3);  
Tef=T0;
```

```

for ind=1:4
    Tef=Tef*T(:, :, ind);
    pef(:, ind)=Tef(1:2, 3);
end;

#Plotar os vetores e suas legendas
origx = quiver(0,0,2*T0(1,1),2*T0(2,1),...
    'Color', 'red','linestyle','--',"linewidth",2);
hold on
origx_line = line([0,2*T0(1,1)], [0,2*T0(2,1)],...
    'Color','red','linestyle','--',"visible","off",...
    "displayname",' \bfx\rm - Base',"linewidth",2);
origy = quiver(0,0,2*T0(1,2),2*T0(2,2),...
    'Color', 'green','linestyle','--',"linewidth",2);
origy_line = line([0,2*T0(1,2)], [0,2*T0(2,2)],...
    'Color', 'green','linestyle','--',"visible","off",...
    "displayname",' \bfy\rm - Base',"linewidth",2);

efx = quiver(pef(1,4),pef(2,4),2*Tef(1,1),2*Tef(2,1),...
    'Color', 'red','linestyle','-',"linewidth",2);
efx_line = line([pef(1,4),2*Tef(1,1)],...
    [pef(2,4),2*Tef(2,1)],...
    'Color', 'red','linestyle','-',"visible","off",...
    "displayname",' \bfx\rm - Efetuador',"linewidth",2);
efy = quiver(pef(1,4),pef(2,4),2*Tef(1,2),2*Tef(2,2),...
    'Color', 'green','linestyle','-',"linewidth",2);
efy_line = line([pef(1,4),2*Tef(1,2)],...
    [pef(2,4),2*Tef(2,2)],...
    'Color', 'green','linestyle','-',"visible","off",...
    "displayname",' \bfy\rm - Efetuador',"linewidth",2);

#Plotar os elos como linhas
for ind=2:4
    line([pef(1,ind-1),pef(1,ind)],...
        [pef(2,ind-1),pef(2,ind)],...
        "linewidth",2,"displayname",['Elo ', num2str(ind-1)]);
end;

hold off;

#Configurar legenda, eixos e fontes
axis equal;
set(gca, "linewidth", 2, "fontsize", 12);
leg=legend("location", "southeast");
set(leg, "fontsize", 12);

```

A.3 Código para criar a Interface Gráfica (create_gui.m)

```
## Código para criar a Interface Gráfica de Usuário
## Samuel Duque Gonçalves - 2020
##Códigos iniciais
clear all
close all
clear h
clc
graphics_toolkit qt

##Conectar ao CoppeliaSim?
h.online=true;#true=conectado; false=desconectado
##Se for trabalhar online, executar função para conexão
if h.online
    conect
endif;

#Inicializar variáveis:
h.j1=0; h.j2=0; h.j3=0; h.j4=0; h.j5=0; h.j6=0;
h.alfa=0; h.beta=0; h.gama=0;
h.Tef=eye(4);

##Criar tela com gráfico
figure(1, 'position',...
[get(0,"screensize")(3:4)/4 get(0,"screensize")(3:4)/2]);
h.ax = axes ("position", [0 0.5 0.5 0.5]);

##Função para atualizar a plotagem
function update_plot (obj, init = false)
h = guidata (obj);
replot = false;
recalc = false;

switch (gcbo)
    case {h.linecolor_radio_2D}
        set (h.linecolor_radio_3D, "value", 0);
        h.j1=0; h.j4=0; h.j6=0;
        set (h.joint1_slider,"Value",0, "enable", "off");
        set (h.joint4_slider,"Value",0, "enable", "off");
        set (h.joint6_slider,"Value",0, "enable", "off");
        set (h.joint1_edit,"string","0", "enable", "off");
        set (h.joint4_edit,"string","0", "enable", "off");
        set (h.joint6_edit,"string","0", "enable", "off");
        recalc = true;
```

```
case {h.linecolor_radio_3D}
    set (h.linecolor_radio_2D, "value", 0);
    set (h.joint1_slider, "enable", "on");
    set (h.joint4_slider, "enable", "on");
    set (h.joint6_slider, "enable", "on");
    set (h.joint1_edit, "enable", "on");
    set (h.joint4_edit, "enable", "on");
    set (h.joint6_edit, "enable", "on");
case {h.joint1_slider}
    set (h.joint1_edit, "string",...
        num2str(get (h.joint1_slider, "value")))
    recalc = true;
case {h.joint2_slider}
    set (h.joint2_edit, "string",...
        num2str(get (h.joint2_slider, "value")));
    recalc = true;
case {h.joint3_slider}
    set (h.joint3_edit, "string",...
        num2str(get (h.joint3_slider, "value")));
    recalc = true;
case {h.joint4_slider}
    set (h.joint4_edit, "string",...
        num2str(get (h.joint4_slider, "value")));
    recalc = true;
case {h.joint5_slider}
    set (h.joint5_edit, "string",...
        num2str(get (h.joint5_slider, "value")));
    recalc = true;
case {h.joint6_slider}
    set (h.joint6_edit, "string",...
        num2str(get (h.joint6_slider, "value")));
    recalc = true;
case {h.joint1_edit}
    set (h.joint1_slider, "Value",...
        str2num(get (h.joint1_edit, "String")));
    recalc = true;
case {h.joint2_edit}
    set (h.joint2_slider, "Value",...
        str2num(get (h.joint2_edit, "String")));
    recalc = true;
case h.joint3_edit
    set (h.joint3_slider, "Value",...
        str2num(get (h.joint3_edit, "String")));
    recalc = true;
case {h.joint4_edit}
    set (h.joint4_slider, "Value",...
        str2num(get (h.joint4_edit, "String")));
    recalc = true;
```

```

case {h.joint5_edit}
    set (h.joint5_slider, "Value",...
        str2num(get (h.joint5_edit, "String")));
    recalc = true;
case {h.joint6_edit}
    set (h.joint6_slider, "Value",...
        str2num(get (h.joint6_edit, "String")));
    recalc = true;
endswitch

if (recalc || init) #No slider ou no inicio
    h.j1=get (h.joint1_slider, "value");
    h.j2=get (h.joint2_slider, "value");
    h.j3=get (h.joint3_slider, "value");
    h.j4=get (h.joint4_slider, "value");
    h.j5=get (h.joint5_slider, "value");
    h.j6=get (h.joint6_slider, "value");
    h.Tef=robot_plot(h.j1,h.j2,h.j3,h.j4,h.j5,h.j6);

    if h.online
        set_target(h.j1,h.j2,h.j3,h.j4,h.j5,h.j6);
    endif;

    set (h.se3_matrix, "String",...
        num2str((round(1000*h.Tef))/1000, "%10.4f"));
    if (get (h.linecolor_radio_2D, "value")==1)
        view (90, 0);
    else
        view (135, 30)
    endif
endif

endfunction

## Modo de controle
h.linecolor_label = uicontrol ("style", "text",
    "units", "normalized",
    "string", "Dimensões:",
    "horizontalalignment", "left",
    "position", [0.6 0.9 0.35 0.04]);

gp = uibuttongroup ( "Position", [ 0.8 0.9 .2 .04]);
h.linecolor_radio_2D=uicontrol (gp,"style", "radiobutton",
    "units", "normalized",
    "string", "2D",
    "callback", @update_plot,

```

```
"value", 0,
"position", [0 0 1 1]);

h.linecolor_radio_3D=icontrol (gp,"style", "radiobutton",
"units", "normalized",
"string", "3D",
"callback", @update_plot,
"value", 1,
"position", [0.5 0 1 1]);

#Descrição das juntas
h.joint1_label = uicontrol ("style", "text",
"units", "normalized",
"string", "Junta 1 (-180° a 180°):",
"horizontalalignment", "left",
"position", [0.56 0.85 0.34 0.04]);

h.joint2_label = uicontrol ("style", "text",
"units", "normalized",
"string", "Junta 2 (-100° a 100°):",
"horizontalalignment", "left",
"position", [0.56 0.75 0.34 0.04]);

h.joint3_label = uicontrol ("style", "text",
"units", "normalized",
"string", "Junta 3 (-140° a 140°):",
"horizontalalignment", "left",
"position", [0.56 0.65 0.34 0.04]);

h.joint4_label = uicontrol ("style", "text",
"units", "normalized",
"string", "Junta 4 (-180° a 180°):",
"horizontalalignment", "left",
"position", [0.56 0.55 0.34 0.04]);

h.joint5_label = uicontrol ("style", "text",
"units", "normalized",
"string", "Junta 5 (-115° a 115°):",
"horizontalalignment", "left",
"position", [0.56 0.45 0.34 0.04]);

h.joint6_label = uicontrol ("style", "text",
"units", "normalized",
"string", "Junta 6 (-180° a 180°):",
"horizontalalignment", "left",
"position", [0.56 0.35 0.34 0.04]);
```

```
##Slider de controle das juntas
h.joint1_slider = uicontrol ("style", "slider",
    "units", "normalized",
    "string", "slider",
    'Min', -180,
    'Max', 180,
    "callback", @update_plot,
    "value", 0,
    "position", [0.56 0.8 0.34 0.04]);

h.joint2_slider = uicontrol ("style", "slider",
    "units", "normalized",
    "string", "slider",
    'Min', -100,
    'Max', 100,
    "callback", @update_plot,
    "value", 0,
    "position", [0.56 0.7 0.34 0.04]);

h.joint3_slider = uicontrol ("style", "slider",
    'Min', -140,
    'Max', 140,
    "units", "normalized",
    "string", "slider",
    "callback", @update_plot,
    "value", 0,
    "position", [0.56 0.6 0.34 0.04]);

h.joint4_slider = uicontrol ("style", "slider",
    "units", "normalized",
    "string", "slider",
    'Min', -180,
    'Max', 180,
    "callback", @update_plot,
    "value", 0,
    "position", [0.56 0.5 0.34 0.04]);

h.joint5_slider = uicontrol ("style", "slider",
    "units", "normalized",
    "string", "slider",
    'Min', -115,
    'Max', 115,
    "callback", @update_plot,
    "value", 0,
    "position", [0.56 0.4 0.34 0.04]);
```

```
h.joint6_slider = uicontrol ("style", "slider",
    "units", "normalized",
    "string", "slider",
    'Min', -180,
    'Max', 180,
    "callback", @update_plot,
    "value", 0,
    "position", [0.56 0.3 0.34 0.04]);

##edit de controle das juntas
h.joint1_edit = uicontrol ("style", "edit",
    "units", "normalized",
    "string", "0",
    "callback", @update_plot,
    "position", [0.9 0.8 0.1 0.04]);

h.joint2_edit = uicontrol ("style", "edit",
    "units", "normalized",
    "string", "0",
    "callback", @update_plot,
    "position", [0.9 0.7 0.1 0.04]);

h.joint3_edit = uicontrol ("style", "edit",
    "units", "normalized",
    "string", "0",
    "callback", @update_plot,
    "position", [0.9 0.6 0.1 0.04]);

h.joint4_edit = uicontrol ("style", "edit",
    "units", "normalized",
    "string", "0",
    "callback", @update_plot,
    "position", [0.9 0.5 0.1 0.04]);

h.joint5_edit = uicontrol ("style", "edit",
    "units", "normalized",
    "string", "0",
    "callback", @update_plot,
    "position", [0.9 0.4 0.1 0.04]);

h.joint6_edit = uicontrol ("style", "edit",
    "units", "normalized",
    "string", "0",
    "callback", @update_plot,
    "position", [0.9 0.3 0.1 0.04]);
```

```

h.se3_matrix_title = uicontrol ("style", "text",
    "units", "normalized",
    "string", {"Matriz de transformação";...
    "homogênea do efetuador."; "T="},
    "horizontalalignment", "left",
    "position", [0.56 0.16 0.34 0.08]);

h.se3_matrix = uicontrol ("style", "text",
    "units", "normalized",
    "string", num2str((round(1000*h.Tef))/1000, "%10.4f"),
    "horizontalalignment", "left",
    "position", [0.56 0 0.34 0.16]);

set(gcf,"color", get(0,"defaultuicontrolbackgroundcolor"))
guidata (gcf, h)
update_plot (gcf, true);

```

A.4 Função conect.m

```

## Código para conectar ao CoppeliaSim
## Samuel Duque Gonçalves - 2020
function conect()

sim=remApiSetup();
simxFinish(-1);
clientID=simxStart('127.0.0.1',19999,true,true,5000,5);

if (clientID>-1)
    disp('Conectado ao servidor de API remoto');
else
    disp('Falha ao conectar ao servidor de API remoto');
end

endfunction

```

A.5 Função set_target.m

```

## Código para enviar comandos
## para as juntas no CoppeliaSim
## Samuel Duque Gonçalves - 2020

function set_target(j1,j2,j3,j4,j5,j6)
    j=[j1,j2,j3,j4,j5,j6];
sim=remApiSetup();
simxFinish(-1);
    clientID=simxStart('127.0.0.1',19999,true,true,5000,5);
    [r,hand(1)]=simxGetObjectHandle(clientID,...

```

```

        'IRB140_joint1', sim.simx_opmode_blocking);
[r,hand(2)]=simxGetObjectHandle(clientID,...
        'IRB140_joint2', sim.simx_opmode_blocking);
[r,hand(3)]=simxGetObjectHandle(clientID,...
        'IRB140_joint3', sim.simx_opmode_blocking);
[r,hand(4)]=simxGetObjectHandle(clientID,...
        'IRB140_joint4', sim.simx_opmode_blocking);
[r,hand(5)]=simxGetObjectHandle(clientID,...
        'IRB140_joint5', sim.simx_opmode_blocking);
[r,hand(6)]=simxGetObjectHandle(clientID,...
        'IRB140_joint6', sim.simx_opmode_blocking);

for i=1:6
    simxSetJointTargetPosition(clientID,hand(i),...
        deg2rad(j(i)),sim.simx_opmode_oneshot);
end
endfunction

```

A.6 Função robot_plot.m

```

## Código para plotar o braço robótico
## Samuel Duque Gonçalves - 2020

function Tef=robot_plot(j1,j2,j3,j4,j5,j6)

##Matriz 1
T(:, :, 1)=[cosd(j1), -sind(j1), 0, 0;
    sind(j1), cosd(j1), 0, 0.069;
    0, 0, 1, 0;
    0, 0, 0, 1];

##Matriz 2
T(:, :, 2)=[1, 0, 0, 0;
    0, cosd(j2), -sind(j2), .07;
    0, sind(j2), cosd(j2), .3521;
    0, 0, 0, 1];

##Matriz 3
T(:, :, 3)=[1, 0, 0, 0;
    0, cosd(j3), -sind(j3), 0;
    0, sind(j3), cosd(j3), .36;
    0, 0, 0, 1];

##Matriz 4
T(:, :, 4)=[cosd(j4), 0, sind(j4), 0;
    0, 1, 0, .226;
    -sind(j4), 0, cosd(j4), 0;

```

```

    0,0,0,1];

##Matriz 5
T(:, :, 5)=[1,0,0,0;
    0,cosd(j5),-sind(j5),.154;
    0,sind(j5),cosd(j5),0;
    0,0,0,1];

##Matriz 6
T(:, :, 6)=[cosd(j6),0,sind(j6),0;
    0,1,0,0.046;
    -sind(j6),0,cosd(j6),0;
    0,0,0,1];

##Matriz 6
T(:, :, 7)=[1,0,0,0;
    0,1,0,.019;
    0,0,1,0;
    0,0,0,1];

##Multiplicar as matrizes
T0=eye(4);
Tef=T0;
for ind=1:7
    Tef=Tef*T(:, :, ind);
    pef(:, ind)=Tef(1:3,4);
end;

#Plotar os vetores e suas legendas
origx = quiver3(0,0,0,0.1*T0(1,1),0.1*T0(2,1),...
    0.1*T0(3,1),'Color','red',...
    'linestyle','--',"linewidth",1);
hold on #Mantem os plots na mesma figura
origx_line = line([0,0.1*T0(1,1)],[0,0.1*T0(2,1)],...
    [0,0.1*T0(3,1)], 'Color','red','linestyle','--',...
    "visible","off","displayname",'\\bfx\\rm - Base');

origy = quiver3(0,0,0,0.1*T0(1,2),0.1*T0(2,2),...
    0.1*T0(3,2), 'Color','green',...
    'linestyle','--',"linewidth",1);
origy_line = line([0,0.1*T0(1,2)],[0,0.1*T0(2,2)],...
    [0,0.1*T0(3,2)], 'Color','green','linestyle','--',...
    "visible","off","displayname",'\\bfy\\rm - Base');

origz = quiver3(0,0,0,0.1*T0(1,3),0.1*T0(2,3),...
    0.1*T0(3,3), 'Color','blue',...

```

```

    'linestyle','--',"linewidth",1);
origz_line = line([0,0.1*T0(1,3)],[0,0.1*T0(2,3)],...
    [0,0.1*T0(3,3)], 'Color','blue','linestyle','--',...
    "visible","off","displayname",'\\bfz\\rm - Base');

efx = quiver3(pef(1,7),pef(2,7),pef(3,7),...
    0.1*Tef(1,1),0.1*Tef(2,1),0.1*Tef(3,1),...
    'Color', 'red',"linewidth",1);
efx_line = line([pef(1,7),0.1*Tef(1,1)],...
    [pef(2,7),0.1*Tef(2,1)],[pef(3,7),0.1*Tef(3,1)],...
    'Color', 'red','linestyle','-','...',
    "visible","off","displayname",'\\bfx\\rm - Efetuador');

efy = quiver3(pef(1,7),pef(2,7),pef(3,7),...
    0.1*Tef(1,2),0.1*Tef(2,2),0.1*Tef(3,2),...
    'Color', 'green',"linewidth",1);
efy_line = line([pef(1,7),0.1*Tef(1,2)],...
    [pef(2,7),0.1*Tef(2,2)],[pef(3,7),0.1*Tef(3,2)],...
    'Color', 'green','linestyle','-','...',
    "visible","off","displayname",'\\bfy\\rm - Efetuador');

efz = quiver3(pef(1,7),pef(2,7),pef(3,7),...
    0.1*Tef(1,3),0.1*Tef(2,3),0.1*Tef(3,3),...
    'Color', 'blue',"linewidth",1);
efz_line = line([pef(1,7),0.1*Tef(1,3)],...
    [pef(2,7),0.1*Tef(2,3)],[pef(3,7),0.1*Tef(3,3)],...
    'Color', 'blue','linestyle','-','...',
    "visible","off","displayname",'\\bfz\\rm - Efetuador');

#Plotar os elos como linhas
for ind=2:7
    line([pef(1,ind-1),pef(1,ind)],...
        [pef(2,ind-1),pef(2,ind)],[pef(3,ind-1),pef(3,ind)],...
        "linewidth",2,"displayname",['Elo ',num2str(ind-1)]);
end;

hold off;

#Configurar legenda, eixos e fontes
axis equal;
set(gca, "linewidth", 2, "fontsize", 12);
leg=legend("location","southeast");
set(leg, "fontsize", 12);

endfunction

```

Anexo B

Configuração do CoppeliaSim para as simulações

As etapas descritas neste anexo permitem controlar o braço robótico IRB 140 da marca ABB no ambiente de simulação do CoppeliaSim. Apesar de apresentar um escopo longo, a parametrização possui baixa complexidade em suas ações. A versão do CoppeliaSim utilizada é a Edu, número 4.1.0, em um computador com sistema operacional *Microsoft Windows 10*, com 8 gigabytes de memória RAM e processador com 2,5 gigahertz de frequência.

Uma vez aberto o CoppeliaSim, a parametrização se dará por meio de duas etapas, descritas nas seções abaixo. A primeira etapa consiste em deixar o braço em condições para que seja feito o levantamento da posição de seus itens. A segunda, por sua vez, habilita o controle do braço robótico durante a animação, utilizando códigos específicos a serem inseridos no simulador ou por meio de interface de programação de aplicação.

B.1 Configuração para o levantamento das juntas e elos

Essa seção apresenta os passos necessários para centralizar o braço robótico IRB 140 na origem do sistema, alinhando-o ao eixo \vec{y} e atribuindo o valor de 0° às juntas. As ações a serem feitas são as seguintes:

1. No Navegador de modelos, na pasta robots/non-mobile, encontrar o braço robótico “ABB IRB 140.ttm”, selecionar e arrastar para a página da simulação. Ao soltar na página, abrirá uma janela contendo informações do braço robótico e, para fechá-la, basta clicar em “OK”;

2. Para centralizar com a origem da simulação, selecionar “IRB140” na Hierarquia da

cena e clicar no botão “Object/ item shift”  , na barra de ferramentas. Na janela que abrir, na aba “Position”, marcar “Relative to: World”, e preencher o valor de 0 metros para as coordenadas X e Y, conforme ilustrado na Figura B.1a, e fechar a janela;

3. Para rotacionar o braço de forma a orientá-lo com o eixo \vec{y} , selecionar “IRB140”

na Hierarquia da cena e clicar selecionar “Object/item rotate”  , na barra de

ferramentas. Na janela que abrir, na aba “*Orientation*”, marcar “*Relative to: World*”, e preencher o valor de 90 graus para o ângulo γ , que significa a rotação de 90° em torno do eixo \vec{z} , fazendo o alinhamento mudar de \vec{x} para \vec{y} , conforme ilustrado na Figura B.1b. Fechar a janela em seguida;

4. Para orientar a base do braço ao eixo \vec{y} , clicar com o botão direito do *mouse* em “IRB140” na Hierarquia da cena e clicar em “*Edit/Reorient bounding box/with reference frame of world*”;
5. Como o ângulo inicial da quinta junta é -90° , para zerá-lo basta expandir o braço robótico “IRB140” na Hierarquia da Cena clicando em “+”, dar um duplo clique em “IRB140_joint5” e, na aba “*Joint*”, em “*Position [deg]*”, preencher o valor 0, teclando *Enter* em seguida. Logo em seguida, fechar a janela de propriedades do objeto.

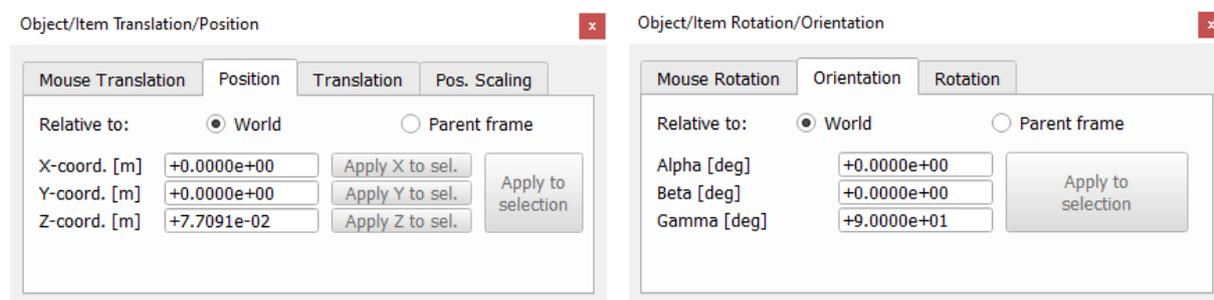
Para descobrir a posição das juntas e da conexão do efetuador, basta selecionar os itens na Hierarquia da Cena e ver as coordenadas no Texto de Informação.

Após a realização dos passos acima, ao se executar a simulação em “*Simulation/Start Simulation*” a junta 5 retorna a -90° e, ao clicar no braço robótico, é aberta a interface gráfica de usuário do modelo da ABB, que permite controlar o braço robótico por meio de barras de rolagem. No entanto, esse meio de controle não é utilizado neste trabalho. Para habilitar, então, a alteração das juntas durante a animação, faz-se necessário realizar a configuração da seção a seguir.

Figura B.1: Janelas de translação e rotação de objetos no CoppeliaSim

(a) Janela de “*Object/item shift*”

(b) Janela de “*Object/item rotate*”



Fonte: Elaborado pelo autor.

B.2 Configuração para o início da animação

Nessa seção são exibidas as ações necessárias para habilitar o controle por meio do Editor de comandos Lua e também por interfaces de programação de aplicativos (APIs), além de criar uma janela que informa a matriz de transformação homogênea do efetuador durante a animação. Após o início dessas etapas, não é mais possível iniciar a simulação e controlar o braço por meio da interface gráfica de usuário embarcada da ABB. As ações são as seguintes:

1. Na Hierarquia da Cena, selecionar e deletar os itens “IRB140_target”, “IRB140_manipulationSphere” e “IRB140_manipulationSphereBase”. Com isso a esfera verde que estava no braço robótico irá desaparecer;

- No Navegador de modelos, na pasta *other*, encontrar a representação dos eixos “reference.frame.ttm” e arrastar para a página da simulação. Para acoplar a referência ao efetuador, apertar a tecla Ctrl do teclado e selecionar “ReferenceFrame” e “IRB140_connection” na Hierarquia da cena e clicar no botão “*Assemble/Disassemble*”



. Logo após, selecionar apenas o “ReferenceFrame” e clicar em “*Object/item*



shift”. Na janela que abrir, na aba “*Position*”, marcar “*Relative to: Parent frame*”, e preencher o valor de 0 metros para as coordenadas X, Y e Z, e fechar a janela. Por último, dar um duplo-clique no nome do “ReferenceFrame” na Hierarquia de Cena e alterar para “Efetuador” e teclar *Enter*. Para aumentar os eixos, expandir o “Efetuador” na Hierarquia da Cena clicando em “+” e dar duplo-clique no item ReferenceFrame_shape. Na janela que abrir, na aba *Shape* clicar no botão *View/modify geometry*. Em seguida, na nova janela, em *Scaling factors* preencher o valor 2 os eixos X, Y e Z, e clicar no botão *Apply* correspondente, e fechar todas as janelas.

Para finalizar, clicar no botão “*Object/item shift*”  e, na janela que abrir, na aba *Pos. Scaling*, marcar *Parent frame*, preencher 2 para X, Y e Z, e clicar no botão *Scale position* maior apenas uma vez, fechando a janela em seguida;

- Inserir, novamente, um “reference.frame.ttm” na Página de animação, para acoplar à origem do sistema. Em seguida, selecionar apenas o “ReferenceFrame” e clicar

em “*Object/item shift*” . Na janela que abrir, na aba “*Position*”, marcar “*Relative to: World*”, e preencher o valor de 0 metros para as coordenadas X, Y e Z, e fechar a janela. Dar um duplo-clique no nome do “ReferenceFrame” na Hierarquia de Cena e alterar para “Origem” e teclar *Enter*. Para aumentar os eixos, expandir a “Origem” na Hierarquia da Cena clicando em “+” e dar duplo-clique no item ReferenceFrame_shape0. Na janela que abrir, na aba *Shape* clicar no botão *View/modify geometry*. Em seguida, na nova janela, em *Scaling factors* preencher o valor 5 os eixos X, Y e Z, e clicar no botão *Apply* correspondente, e fechar todas as

janelas. Por último, clicar no botão “*Object/item shift*”  e, na janela que abrir, na aba *Pos. Scaling*, marcar *Parent frame*, preencher 5 para X, Y e Z, e clicar no botão *Scale position* maior apenas uma vez, fechando a janela em seguida;

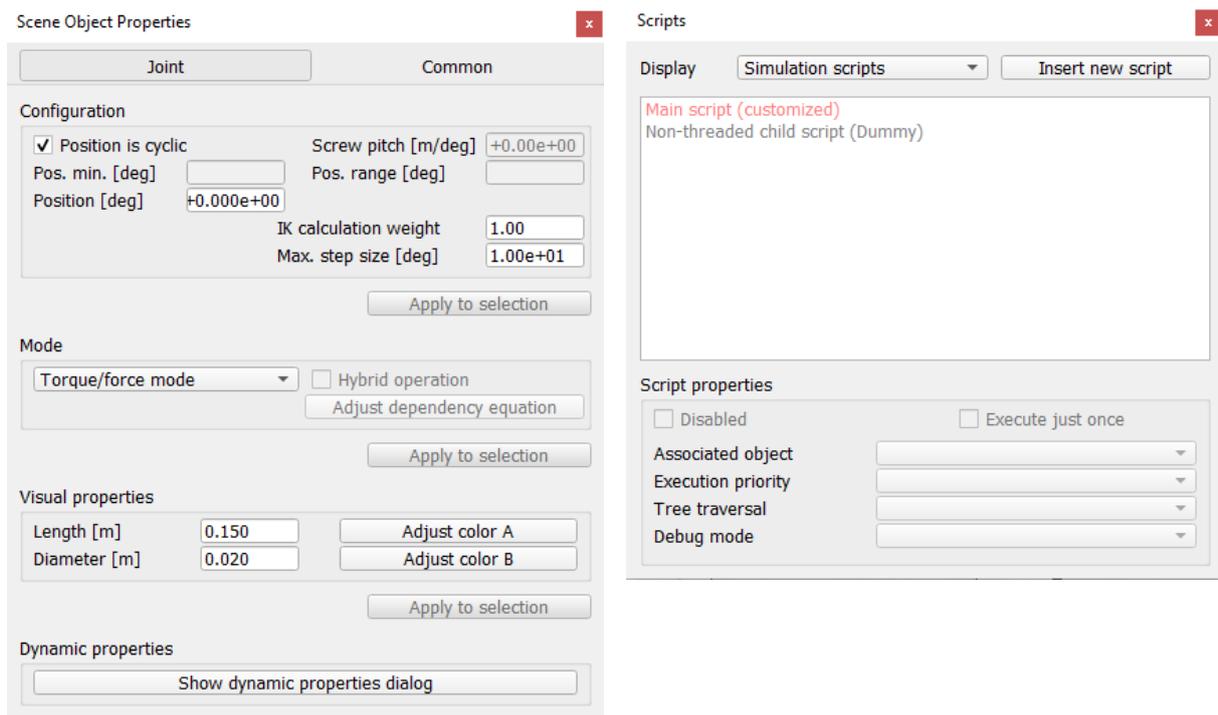
- Dar um duplo-clique no ícone da junta 1, isto é, da “IRB140_joint1” e na janela que abrir, na aba “*Joint*”, no item “*Mode*”, selecionar a opção “*Torque/force mode*” na lista suspensa, como demonstrado na Figura B.2a, e fechar a janela em seguida. Repetir esse passo para todas as seis juntas do braço robótico;
- Clicar em “*Add/Dummy*”, para criar um objeto ao qual será adicionado o código para o controle do braço robótico;
- Clicar em “*Tools/Scripts*” e, na janela que abrir, selecionar “*Non-threaded child script (IRB140)*” e teclar *delete* para apagar o código embarcado do modelo da ABB. Em seguida, clicar “*Insert new script*”, adicionando um *script* do tipo “*Child script (non-threaded)*” e clicar em *OK*. Logo após, selecionar “*Non-threaded child script*

(*unassociated*)” e, no “*Script properties*”, no item “*Associated object*”, selecionar o item “*Dummy*”, o qual foi criado no passo anterior, como demonstrado na Figura B.2b. Por último, dar um duplo-clique no *script* e, na página que abrir, apagar o conteúdo e colar o código a seguir:

Figura B.2: Janela de propriedades de juntas e janela de *scripts* no CoppeliaSim

(a) Janela de propriedades de juntas

(b) Janela de *scripts*



Fonte: Elaborado pelo autor.

```
function sysCallInit()
    simRemoteApi.start(19999)
    xml = [[
    <ui closeable="false" on-close="closeEventHandler" resizable="true">
    <tabs>
    <tab title="Efetuador">
    <group>
    <label text="<big> Matriz de SE(3):</big>"id="6008"
    wordwrap="false" style="font-weight: bold;"/>
    <label text="Tef = " wordwrap="false" />
    <label text="" id="1234" wordwrap="false" />
    </group>
    <stretch />
    </tab>
    </tabs>
```

```
</ui>
]]
ui=simUI.create(xml)
    base=sim.getObjectHandle('Origem')
    ee=sim.getObjectHandle('Efetuador')
    calcSE3(ui)
end

function createSE3string()
    local g=createSE3matrix()
    local out="<b><big><tt>"
    for i=1,4,1 do
        out=out.."| "
        for j=1,4,1 do
            out=out..string.format(" %+.6.3f",g[i][j])
        end
        out=out.." | <br>"
    end
    out=out.."</tt></big></b>"
    return out
end

function createSE3matrix()
    local matrix=sim.getObjectMatrix(ee,base)
    local g =
        for i=1,4,1 do
            g[i] =
                for j=1,4,1 do
                    if (i<=3) then
                        g[i][j] = matrix[4*(i-1)+j]
                    else
                        if (j<=3) then
                            g[i][j] = 0
                        else
                            g[i][j] = 1
                        end
                    end
                end
            end
        end
    return g
end

function calcSE3(ui)
    out=createSE3string()
    simUI.setLabelText(ui,1234,out)
end
```

```
function sysCallActuation()
    calcSE3(ui)
end
```

```
function sysCallCleanup()
    simUI.destroy(ui)
    simRemoteApi.stop(19999)
end
```

Após a realização dos passos acima, é habilitado o comando por APIs e pelo comando abaixo, a ser escrito no Editor de Comandos Lua do próprio CoppeliaSim:

```
sim.setJointTargetPosition(sim.getObjectHandle('IRB140_joint1'),90*math.pi/180)
```

No exemplo anterior, é enviado o comando de alteração do ângulo da junta 1 para 90°. Caso a junta já esteja nesse ângulo, o braço não se movimenta.

Após a configuração realizada, recomenda-se salvar o arquivo em alguma pasta do sistema, clicando em “*File/Save scene*”. O nome utilizado para o arquivo nesse trabalho foi `robot_arm.ttt`.

Anexo C

Configuração de interfaces para o Octave

Este anexo contém as etapas necessárias para configurar o Octave de modo a habilitar duas interfaces, que são elementos que possibilitam intermediar sistemas distintos. As interfaces são as seguintes:

Interface Gráfica de Usuário aplicação que permite a interação do usuário com o Octave de forma visual e intuitiva, enviando comando por meio de elementos (botões, barras, caixas de texto etc) e recebendo informações através de gráficos e textos.

Interface de Programação de Aplicação conjunto de rotinas e funções que permitem a comunicação entre *softwares*. No caso, habilita o envio de comandos do Octave para o CoppeliaSim.

Antes de criar a interface gráfica de usuário, é necessária a criação de uma pasta no sistema para reunir todos os arquivos necessários. O sistema operacional utilizado foi o *Microsoft Windows 10*, em um computador com 8 gigabytes de memória RAM e processador com 2,5 gigahertz de frequência.

O código `create_gui.m`, disponível na Seção [A.3](#), cria a interface gráfica e precisa estar na pasta em questão. Ao ser executado, este código chama as funções externas a seguir, as quais também precisam estar na pasta referida:

connect conecta a interface gráfica de usuário ao CoppeliaSim. É executado somente quando o código de criação da interface é iniciado e a variável `h.online` receber o valor `true`. Para que seja bem sucedida a conexão, é necessário que a simulação do CoppeliaSim esteja executando antes de executar a criação da interface, exibindo o texto “Conectado ao servidor de API remoto” na Janela de Comandos. Caso haja erro na conexão, é exibido o texto “Falha ao conectar ao servidor de API remoto”. O código da função `connect` está na Seção [A.4](#) do Anexo [A](#);

set_target envia os ângulos que serão alvos a serem alcançados pelo braço robótico do CoppeliaSim. É executado quando há alterações dos ângulos na interface gráfica, somente se a variável `h.online` tiver recebido o valor `true` e a conexão, por meio da função `connect`, tiver sido bem-sucedida. Na Seção [A.5](#) do Anexo [A](#) está contido o código da função `set_target`;

robot_plot este código é similar ao apresentado na seção [A.1](#), e tem o objetivo de atualizar a figura contida na interface gráfica. O código da função `set_target` está na Seção [A.6](#) do Anexo [A](#);

Além dos quatro arquivos descritos anteriormente, para habilitar a interface de programação de aplicação é necessário que haja, na mesma pasta, os seguintes arquivos:

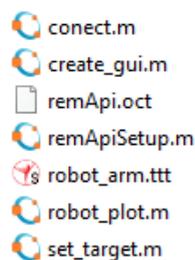
- remApiSetup.m, disponível na pasta C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu/programming/remoteApiBindings/octave/octave, ou pasta similar onde esteja instalado o programa;
- remApi.oct.

O arquivo remApi.oct não é disponibilizado nas pastas do CoppeliaSim. Para que ele seja criado é essencial a realização dos seguintes passos:

1. Criar uma pasta provisória no sistema. O nome dado a esta pasta, para que não haja confusão, é de “Provisória”;
2. Copiar para esta pasta “Provisória” os seguintes itens:
 - Arquivo remoteApi.dll, contido na pasta C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu/programming/remoteApiBindings/lib/lib/Windows, ou similar;
 - Todos os arquivos contidos na pasta C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu/programming/include, ou similar;
 - Todos os arquivos contidos na pasta C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu/programming/remoteApi, ou similar. Ao fazê-lo, irá abrir uma janela perguntando sobre sobrescrever o arquivo license.txt, o que pode ser feito sem problemas;
 - Os arquivos buildWin.m e remApi.cc contidos na pasta C:/Program Files/CoppeliaRobotics/CoppeliaSimEdu/programming/remoteApiBindings/octave, ou similar;
3. Em seguida, executar o arquivo buildWin.m no Octave. Com a Janela de Comandos selecionada, pressionar qualquer tecla do teclado quando aparecer o texto “Press any key to start”. O arquivo remApi.oct aparecerá na pasta “Provisória” após a execução do código.

Por fim, os arquivos necessários para realizar as simulações deste trabalho, incluindo o arquivo robot_arm.ttt criado no Anexo B, são os descritos pela Figura C.1, organizados de forma alfabética na pasta de exemplo.

Figura C.1: Arquivos contidos na pasta de simulação



Fonte: Elaborado pelo autor.