

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JONATHAN CERBARO

**MODELAGEM EXTEROCEPTIVA VIA ÁLGEBRA DE CLIFFORD
PARA TRAJETÓRIA EM AMBIENTE CONFINADO DO ROBÔ
MOVEMASTER RV-M2**

DISSERTAÇÃO

CURITIBA

2022

JONATHAN CERBARO

**MODELAGEM EXTEROCEPTIVA VIA ÁLGEBRA DE
CLIFFORD PARA TRAJETÓRIA EM AMBIENTE CONFINADO
DO ROBÔ MOVEMASTER RV-M2**

**Exteroceptive Modeling Via Clifford's Algebra for Confined
Environment Trajectory of the Movemaster RV-M2 Robot**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, da Universidade Tecnológica Federal do Paraná (UTFPR), como requisito parcial para obtenção do título de “Mestre em Ciências” - Área de Concentração: Engenharia de Automação e Sistemas.

Orientador: Prof. Dr. André Schneider de Oliveira

Coorientador: Prof. Dr. João Alberto Fabro

CURITIBA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



**Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Curitiba**



JONATHAN CERBARO

**MODELAGEM EXTEROCEPTIVA VIA ÁLGEBRA DE CLIFFORD PARA TRAJETÓRIA EM AMBIENTE
CONFINADO DO ROBÔ MOVEMASTER RV-M2**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia De Automação E Sistemas.

Data de aprovação: 12 de Agosto de 2022

Dr. Andre Schneider De Oliveira, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Carlos Henrique Farias Dos Santos, Doutorado - Universidade Estadual do Oeste do Paraná (Unioeste)

Dr. Daniel Martins, Doutorado - Universidade Federal de Santa Catarina (Ufsc)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 12/08/2022.

Dedico este trabalho a meus queridos pais
Laurindo e Marliza, e ao amor da minha vida,
Caroline.

AGRADECIMENTOS

Não há maneira de expressar a gratidão que tenho pelo apoio fornecido pela minha família nesses pouco mais de dois anos que se passaram desde o início do programa de mestrado. Depois de todo o suporte que recebi durante o tempo que passei na graduação em Engenharia Eletrônica, não careci de incentivo quando o meu desejo de prosseguir com os estudos, e obter o título de Mestre, fez com que tivéssemos de lidar com a distância. Obrigado por provar repetidamente que a família está acima de tudo.

Especialmente aos meus pais, agradeço por incentivar a realização dos meus sonhos por mais excêntricos que sejam. Agradeço minha namorada pelo incondicional suporte emocional e (infinita) paciência em meus momentos de dificuldade. Aos meus amigos, agradeço pelas brincadeiras e filosofias sobre a vida, o Universo e tudo que eles contém. Aos professores, agradeço por fornecer o conhecimento e a visão de mundo acadêmico que tornaram este trabalho possível pois, como disse Isaac Newton, “se eu vi mais longe, foi por estar sobre ombros de gigantes”.

“Em um mundo devidamente automatizado e educado, as máquinas podem provar ser a verdadeira influência humanizadora. Pode ser que as máquinas façam o trabalho que torna a vida possível e que os seres humanos façam todas as outras coisas que tornam a vida agradável e valiosa.” (ASIMOV, Isaac)

RESUMO

CERBARO, Jonathan. **Modelagem Exteroceptiva Via Álgebra de Clifford Para Trajetória em Ambiente Confinado do Robô Movemaster RV-M2**. 2022. 151 f. Dissertação (Mestrado em Engenharia Elétrica e Informática Industrial) – Universidade Tecnológica Federal do Paraná. Curitiba, 2022.

Diversos tópicos envolvendo a álgebra que modela e soluciona equações de cinemática na área de manipuladores seriais possuem problemáticas ainda em aberto, apesar de já bem compreendidas. Cada vez mais a indústria exige a inovação nas técnicas de controle e sensoriamento de robôs capazes de interagir com o ambiente de maneira segura. Realiza-se neste trabalho a abordagem da álgebra de Clifford aplicada à cinemática e à descrição de um ambiente de confinamento para o manipulador Movemaster RV-M2, com elementos que permitem a generalização a outros manipuladores. É feita a relação destas técnicas com pontos, linhas e planos que permitem projetar trajetórias livres de colisão. Os experimentos são realizados com o robô após este passar pelo processo de *retrofit*, onde sensores de cor, profundidade e odometria foram instalados além de um novo controlador de código aberto compatível com ROS. Contribui-se com um algoritmo de maximização de objetivos secundários e com uma metodologia inédita de modelagem de obstáculos. Conclui-se com a validade dos métodos propostos através de exemplos realizados com o robô real e simulado, tendo a possibilidade de aplicar as técnicas discutidas em diversas contribuições futuras.

Palavras-chave: Cinemática de manipuladores. Álgebra de Clifford. *Retrofit*. Quatérnios Duais. Ambiente Confinado.

ABSTRACT

CERBARO, Jonathan. **Exteroceptive Modeling Via Clifford's Algebra for Confined Environment Trajectory of the Movemaster RV-M2 Robot**. 2022. 151 p. Dissertation (Master's Degree in Electrical Engineering and Industrial Informatics) – Federal University of Technology - Paraná. Curitiba, 2022.

Several topics involving algebra that models and solves kinematics equations in serial manipulators have problems that are still open, although they are already well understood. Increasingly, the industry demands innovation in control and sensing techniques for robots capable of interacting with the environment safely. In this work, the Clifford algebra approach is applied to kinematics and the description of a confinement environment for the Movemaster RV-M2 manipulator is carried out, with elements that allow generalization to other manipulators. These techniques are related to points, lines, and planes that project collision-free trajectories. The experiments are carried out with the robot after it goes through the *retrofit* process, where color, depth, and odometry sensors were installed in addition to a new open-source controller compatible with ROS. It contributes with an algorithm for maximizing secondary objectives and an unprecedented methodology for modeling obstacles. It concludes with the validity of the proposed methods through examples carried out with the real and simulated robot, with the possibility of applying the techniques discussed in several future contributions.

Keywords: Manipulator kinematics. Clifford Algebra. Retrofit. Dual Quaternions. Confined Environment.

LISTA DE ILUSTRAÇÕES E QUADROS

Figura 1 – Instalações anuais de robôs industriais no mundo.	16
Figura 2 – Controlador para o Movemaster modernizado por Kung e Chen.	20
Figura 3 – Manipulador Beckman Coulter ORCA original.	28
Figura 4 – Manipulador ORCA antes do <i>retrofit</i>	29
Figura 5 – Esquema de controle e comunicação ROS para o ORCA.	30
Figura 6 – Graus de liberdade do manipulador Movemaster original.	32
Figura 7 – Movemaster antes do <i>retrofit</i>	33
Figura 8 – Área de operação em laboratório do Movemaster.	33
Figura 9 – Espaço operacional original do Movemaster.	34
Figura 10 – Adaptação na conexão dos sensores do Movemaster.	35
Figura 11 – Novas conexões externas na traseira do Movemaster.	35
Figura 12 – Estação de controle do Movemaster modernizado.	37
Figura 13 – Garra do Movemaster original e modernizada.	38
Figura 14 – Suporte para as câmeras de profundidade e odometria.	38
Figura 15 – <i>Point cloud</i> capturado pela câmera D435.	39
Figura 16 – Câmera de odometria Realsense T265 em operação.	40
Figura 17 – Nova limitação para a junta 1 do Movemaster.	41
Figura 18 – Nova limitação para a junta 4 do Movemaster.	41
Figura 19 – Modelo virtual do Movemaster.	43
Figura 20 – Movimento da junta 4 refletido na junta 5.	46
Figura 21 – Validação do controlador através de tarefa de <i>pickup</i> de objeto.	47
Figura 22 – Árvore de transformações ROS do Movemaster.	47
Figura 23 – Esquema cinemático do ORCA modernizado.	51
Figura 24 – Esquema cinemático do Movemaster modernizado.	51
Figura 25 – Posicionamento do ORCA no exemplo de cinemática direta.	52
Figura 26 – Posicionamento do Movemaster no exemplo de cinemática direta.	53
Figura 27 – Exemplo de redundância do manipulador ORCA.	57
Figura 28 – Mapeamento redundante entre a velocidade das juntas e do efetuador.	57
Figura 29 – Cinemática inversa diferencial em malha fechada.	60
Figura 30 – Demonstração de convergência da cinemática diferencial (ORCA).	61
Figura 31 – Execução da trajetória circular (ORCA).	61
Figura 32 – Erro de posição na trajetória circular (ORCA).	62
Figura 33 – Trajetória com objetivo secundário (ORCA).	63
Figura 34 – Demonstração de convergência da cinemática diferencial (Movemaster).	64
Figura 35 – Execução da trajetória elíptica (Movemaster).	64
Figura 36 – Erro ao longo da trajetória elíptica (Movemaster).	65
Figura 37 – Trajetória com objetivo secundário (Movemaster).	65
Figura 38 – Trajetória das juntas corrigida pelo algoritmo proposto.	67
Figura 39 – Relação entre um ponto e um plano na Álgebra de Clifford.	74
Figura 40 – Três pontos definindo um plano na álgebra de Clifford.	76
Figura 41 – Área de trabalho montada para o experimento.	79
Figura 42 – Trajetórias desejadas para experimento prático.	81
Figura 43 – Medida da espessura do helicóide até a carcaça na junta 3.	82
Figura 44 – Configurações de leitura dos limites de operação.	84

Figura 45 – Principais blocos funcionais do experimento.	87
Figura 46 – Modelo virtual do manipulador e da área de trabalho medida.	89
Figura 47 – Trajetória elíptica processada.	89
Figura 48 – Variáveis de junta e erro durante a trajetória elíptica.	90
Figura 49 – Distância mínima até os planos durante a trajetória elíptica.	91
Figura 50 – Visualização da execução da trajetória elíptica.	92
Figura 51 – Trajetória elíptica medida pelos <i>encoders</i> , em vermelho. Trajetória desejada em verde.	92
Figura 52 – Momentos da execução da trajetória elíptica no protótipo real.	92
Figura 53 – Trajetória retilínea processada.	93
Figura 54 – Distância mínima até os planos durante a trajetória retilínea.	93
Figura 55 – Variáveis de junta e erro durante a trajetória retilínea.	94
Figura 56 – Visualização da execução da trajetória retilínea.	94
Figura 57 – Trajetória retilínea medida pelos <i>encoders</i> , em vermelho. Trajetória desejada em verde.	95
Figura 58 – Momentos da execução da trajetória em linha no protótipo real.	95
Figura 59 – Trajetória circular processada.	96
Figura 60 – Distância mínima até os planos durante a trajetória circular.	96
Figura 62 – Visualização da execução da trajetória circular.	96
Figura 61 – Variáveis de junta e erro durante a trajetória circular.	97
Figura 63 – Trajetória circular medida pelos <i>encoders</i> , em vermelho. Trajetória desejada em verde.	97
Figura 64 – Momentos da execução da trajetória circular no protótipo real.	98
Figura 65 – Rotação composta pelos ângulos de Euler ZYZ	124
Figura 66 – Representação dos ângulos Roll-Pitch-Yaw.	125
Figura 67 – Eixo torcido para visualização dos parâmetros DH.	126
Figura 68 – Exemplo de deslocamento helicoidal de um corpo rígido.	129
Figura 69 – Exemplo de deslocamento helicoidal em um manipulador.	130
Figura 70 – Organograma do bloco "CONEXÃO ROS".	148
Figura 71 – Organograma do bloco "LEITURA DOS LIMITES".	149
Figura 72 – Organograma do bloco "GERAÇÃO DE TRAJETÓRIA".	150
Figura 73 – Organograma do bloco "ENVIO AO MANIPULADOR".	151
Quadro 1 – Campos da mensagem <i>/custom_msg/set_angles</i>	31
Quadro 2 – Campos da mensagem <i>/custom_msg/status_arm</i>	31
Quadro 3 – Descrição dos componentes no controlador do Movemaster.	37
Quadro 4 – Campos da mensagem <i>/movemaster_msg/setpoint</i>	42
Quadro 5 – Campos da mensagem <i>/movemaster_msg/status</i>	42

LISTA DE TABELAS

Tabela 1 – Relação de pulsos por grau e configuração <i>home</i> do ORCA.	31
Tabela 2 – Ganhos dos controladores PID de cada junta.	38
Tabela 3 – Limites de junta do Movemaster modernizado.	42
Tabela 4 – Relação de pulsos por grau e configuração <i>home</i> do Movemaster.	45
Tabela 5 – Parâmetros DH do ORCA modernizado.	50
Tabela 6 – Parâmetros DH do Movemaster modernizado.	51
Tabela 7 – Helicóides Sucessivos do ORCA modernizado.	54
Tabela 8 – Helicóides Sucessivos do Movemaster modernizado.	54
Tabela 9 – Tabela de Multiplicação dos elementos de um quatérnio.	142

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SIGLAS

IFR	<i>International Federation of Robotics</i>
ISO	<i>International Standards Organization</i>
ROS	<i>Robot Operating System</i>
FPGA	<i>Field Programmable Gate Array</i>
RGB-D	<i>Red, Green, Blue, Distance</i>
SLAM	<i>Simultaneous Location and Mapping</i>
PD	Proporcional-derivativo
PID	Proporcional-integral-derivativo
PWM	<i>Pulse Width Modulation</i>
IMU	<i>Inertial Measurement Unit</i>
URDF	<i>Unified Robotic Description Format</i>
MTH	Matriz de Transformação Homogênea
RPY	<i>Roll-Pitch-Yaw</i>
DLS	<i>Damped Least Squares</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	MOTIVAÇÃO DO TRABALHO	15
1.2	OBJETIVOS	17
1.3	ESTRUTURA DO DOCUMENTO	18
2	ESTADO DA ARTE	19
2.1	REVISÃO DOS TRABALHOS RECENTES RELACIONADOS	19
2.2	CONSIDERAÇÕES	24
3	RETROFIT DOS MANIPULADORES	26
3.1	MANIPULADOR BECKMAN COULTER ORCA	27
3.2	MANIPULADOR MITSUBISHI MOVEMASTER RV-M2	32
3.3	CONSIDERAÇÕES	48
4	CINEMÁTICA CLÁSSICA	49
4.1	CINEMÁTICA DIRETA	50
4.2	APLICAÇÃO DOS PARÂMETROS DENAVIT-HARTENBERG	50
4.3	APLICAÇÃO DOS HELICOIDES SUCESSIVOS	53
4.4	CINEMÁTICA INVERSA DIFERENCIAL	55
4.4.1	Redundância e Objetivos Secundários	56
4.4.2	Cinemática Inversa Diferencial em Malha Fechada	59
4.4.3	Demonstrações com o Beckman Coulter ORCA	60
4.4.4	Demonstrações com o Mitsubishi Movemaster RV-M2	63
4.5	ALGORITMO PROPOSTO PARA OBJETIVO SECUNDÁRIO	66
4.6	CONSIDERAÇÕES	67
5	CINEMÁTICA ATRAVÉS DA ÁLGEBRA DE CLIFFORD	69
5.1	QUATÉRNIOS DUAIS NA CINEMÁTICA DIRETA	69
5.2	PONTOS, LINHAS E PLANOS NA ÁLGEBRA DE CLIFFORD	71
5.2.1	Pontos	71
5.2.2	Linhas	72
5.2.3	Planos	72
5.2.4	Hodge Star	72
5.2.5	Um Ponto e Um Plano	73
5.2.6	Três Pontos Como um Plano	75
5.3	CONSIDERAÇÕES	77
6	ADAPTAÇÃO DE TRAJETÓRIA EM AMBIENTE CONFINADO	78
6.1	CENÁRIO EXPERIMENTAL	78
6.2	RESULTADOS DOS EXPERIMENTOS	88
6.2.1	Medição da Área de Trabalho	88
6.2.2	Trajetoária Elíptica	89
6.2.3	Trajetoária Retilínea	93
6.2.4	Trajetoária Circular	95
6.3	ANÁLISE DOS RESULTADOS	98

6.4	CONSIDERAÇÕES	102
7	CONSIDERAÇÕES FINAIS	103
7.1	CONCLUSÕES	103
7.2	TRABALHOS FUTUROS	107
	REFERÊNCIAS	109
	APÊNDICES	119
	APÊNDICE A – MATRIZES DE TRANSFORMAÇÃO HOMOGÊNEA	120
	APÊNDICE B – REPRESENTAÇÃO MÍNIMA DE ORIENTAÇÃO . .	124
	APÊNDICE C – PARÂMETROS DENAVIT-HARTENBERG	126
	APÊNDICE D – HELICOIDES SUCESSIVOS	129
	APÊNDICE E – MATRIZ JACOBIANA NA CINEMÁTICA INVERSA DIFERENCIAL	132
	APÊNDICE F – CINEMÁTICA ATRAVÉS DA ÁLGEBRA DE CLIF- FORD	137
	APÊNDICE G – DETALHAMENTO DOS BLOCOS DO ORGANO- GRAMA DO PROJETO EXPERIMENTAL	148

1 INTRODUÇÃO

O interesse na construção de máquinas com potencial de sintetizar o comportamento humano a fim de que sejam capazes de executar tarefas cotidianas com maior eficiência data desde Aristóteles (384 a.C – 322 a.C), que imaginava a abolição da escravatura com o surgimento de máquinas automatizadas, passando por Leonardo da Vinci (1452 – 1519) que projetou em 1495 um robô humanoide articulado nos braços, boca e mandíbula (HAZARIKA; DIXIT, 2018).

Hoje, a distinção literal entre o que deve ser considerado um robô normalmente é dada por uma classificação atrelada à complexidade e à capacidade de reprogramação do dispositivo: quando uma máquina pode ser reprogramada para se adaptar à execução de diversas tarefas, então ela é comumente considerada um robô. Em contrapartida, quando uma máquina foi projetada de forma limitada a executar um determinado conjunto de tarefas, então esta é considerada uma máquina de automação fixa (CRAIG, 2012). A definição de “robô industrial” utilizada pela IFR (*International Federation of Robotics*) é a mesma que consta na ISO (*International Standards Organization*) número 8373¹: “manipulador multifuncional controlado automaticamente, reprogramável, programável em três ou mais eixos, que pode ser fixo ou móvel para uso em aplicações de automação industrial”.

Um dos tipos de robôs mais utilizado para a interação com objetos na indústria é chamado de manipulador serial. Chen et al. (2011), afirmam que o uso de manipuladores robóticos impulsionou a produtividade do setor de manufatura devido à versatilidade em sua reprogramação. Hazarika e Dixit (2018) reiteram que esta é a configuração de robô significativamente mais comum na indústria, definindo-os como sendo uma estrutura fisicamente ancorada a uma estação de trabalho constituída por uma cadeia de juntas controláveis que permite posicionar o seu efetuator, ou ferramenta, em diferentes posições e orientações dentro do espaço de trabalho.

Neste capítulo, apresenta-se a robótica no contexto do trabalho realizado, bem como suas motivações e relevância. Os objetivos geral e específicos são definidos de forma que possam ser acompanhados ao longo da leitura.

¹ Disponível em <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-3:v1:en>

1.1 MOTIVAÇÃO DO TRABALHO

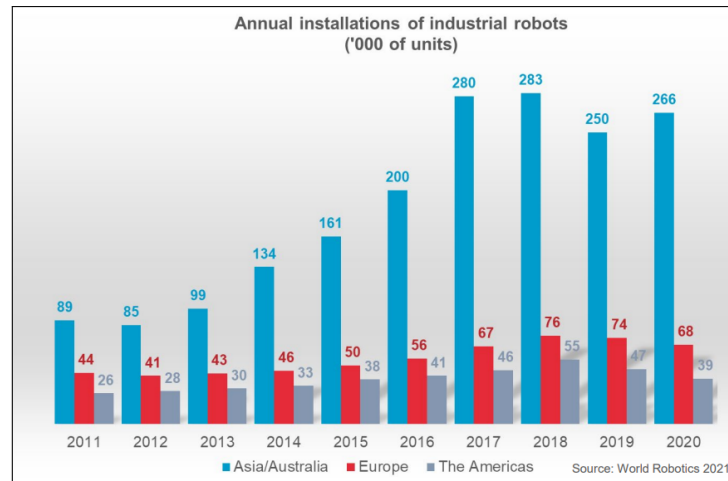
Avanços na robótica tomaram um novo foco com a pandemia de Covid-19 que atingiu o mundo no final do ano de 2019. Entre os diversos avanços que foram realizados e aplicados no contexto da pandemia, é possível exemplificar robôs móveis capazes de entregar mantimentos como comida e remédios para pessoas isoladas de forma autônoma, até manipuladores aplicados em centros cirúrgicos que precisam realizar operações com controle de movimentação, adaptando seus movimentos durante os procedimentos. Na robótica cirúrgica, o controle com repetição e precisão de manipuladores em cenários invasivos — análogos a um ambiente confinado — ajuda na realização de processos com risco reduzido e mínima intervenção humana. Estas informações são expostas na pesquisa realizada por Javaid et al. (2020).

Em linhas de montagem industriais, os manipuladores desempenham um papel importante na automação de tarefas de manuseio utilizando o sensoriamento de seu ambiente e de seu próprio mecanismo para traçar trajetórias inteligentes. As aplicações abrangem áreas de colaboração humano-robô (WANG et al., 2022), robôs colaborativos (PARK; PARK, 2008) e linhas de montagem em geral (DING et al., 2020). É possível que não seja de interesse da aplicação permitir que o robô chegue a certas configurações dentro de sua área de trabalho, seja por questões de prevenção de colisão com outros dispositivos já conhecidos ou detecção de colisão com objetos móveis. Seja qual for o caso, aplicam-se técnicas de adaptação de trajetórias que podem ser descritas com base em dados coletados por sensores do manipulador.

De forma geral, é importante para um sistema robótico ser capaz de lidar com as adversidades encontradas no seu ambiente, adaptar-se e resolver ou contornar problemas de navegação e exploração (YANG et al., 2018). Dada a característica que define um manipulador de que o mecanismo está fixo em sua plataforma de trabalho, sua mobilidade é limitada a uma área de alcance já conhecida. Torna-se necessário, então, o estudo de técnicas que permitam a construção de um sistema de visão e sensoriamento próprio e exteroceptivo de modo a permitir ao robô planejar seu movimento em situações de confinamento.

Do ponto de vista industrial, o relatório de 2021 publicado pela IFR referente a estatísticas de transporte e instalação de robôs industriais mostra o constante crescimento no emprego destas máquinas em todo o mundo, o qual é impulsionado na Ásia pela China e teve uma notável estagnação durante os anos da pandemia, conforme pode ser observado na figura 1.

No contexto matemático do tema, a cinemática de um manipulador pode atingir grande

Figura 1 – Instalações anuais de robôs industriais no mundo.

Fonte: IFR - International Federation of Robotics (2021)

complexidade algorítmica para ser solucionada, envolvendo etapas de cálculo numérico e processamento de dados sensoriais para adaptação de trajetória. Como será exposto no capítulo 2, os problemas em aberto na cinemática não são recentes em sua formulação, porém continuam sendo alvo de estudos para a obtenção de técnicas que possam tornar mais eficazes determinados pormenores dos sistemas. O problema da inversão de uma matriz Jacobiana, por exemplo, que por ser interpretado do ponto de vista puramente matemático sem associação com a robótica, é abordado em artigos do início do século XX como de Dederick (1913) e Clements (1913), porém continuam em evidência em contribuições de relevância no ano de publicação desta dissertação, como no proposto por Jesus et al. (2022).

As ferramentas algébricas para descrição e computação das equações de cinemática também estão em constante evolução. A aplicação dos quatérnios duais, os quais são derivados da álgebra de Clifford, são uma técnica promissora em eficiência computacional, elegante matematicamente e com menos problemas de representação conhecida. Seus conceitos exigem um esforço de aprendizado maior devido à sua falta de familiaridade que, apesar de um parâmetro subjetivo, é tido como verdade (KENWRIGHT, 2012). Há constante esforço em trabalhos voltados ao ensino das técnicas de representação dual-quatérniônica, com o intuito de tornar sua aplicação mais acessível, a exemplo de Montgomery-Smith e Shy (2022). Uma das contribuições pretendidas com a realização do projeto descrito nesta dissertação é o incentivo da aplicação abrangente de técnicas algébricas avançadas.

Ao longo do projeto, diversas etapas de pesquisa contribuem para a construção de um sistema de controle, visão computacional e comunicação da cinemática do manipulador

Movemaster RV-M2, o qual passa pelo processo de modernização com adição de sensoriamento e compatibilidade com ROS (*Robot Operating System*), formulado através de quatérnios duais, com contribuição na prevenção de colisões ao longo de uma trajetória pré-definida. A hipótese testada nas etapas experimentais é a possibilidade e eficácia na utilização da álgebra de Clifford para modelar matematicamente tanto o manipulador quanto o ambiente de confinamento com a finalidade de prevenir colisões em toda a estrutura utilizando ferramentas matemáticas unificadas para todo o processo. Também é exposta a contribuição relacionada à exploração de uma propriedade existente no equacionamento da cinemática diferencial, acessível ao manipular o espaço nulo da matriz Jacobiana para executar um objetivo secundário.

1.2 OBJETIVOS

Esta dissertação tem como objetivo geral construir um sistema de geração e seguimento de trajetória para um manipulador industrial modernizado de 5 graus de liberdade, dotado de sistemas exteroceptivos, capaz de evitar colisões com os limites de sua área de trabalho, aplicando uma modelagem baseada nos conceitos da álgebra de Clifford.

Foi necessário, ao longo dos estudos e da realização dos experimentos, o foco em mais de uma área de concentração. Isso se deve ao fato de que o objetivo geral envolve a sincronia de etapas intermediárias, cada qual permitindo uma parcela de contribuição acadêmica. Os objetivos parciais descritos ao longo do documento foram organizados de modo a estabelecer a relevância das etapas intermediárias que levam ao objetivo final. O estudo da literatura relacionada permitiu a identificação dos temas recorrentes, o que levou ao estudo aprofundado das álgebras que envolvem projetos de cinemática. Para execução destes conceitos, é necessária a criação de uma estação de trabalho modernizada, o que levou ao processo de *retrofit* de dois manipuladores. Todos os conceitos são unificados com os sistemas de visão computacional para a principal contribuição deste trabalho: uma maneira inédita de representação espacial utilizando a álgebra de Clifford, sendo aplicada neste caso à geração e adaptação de trajetória de um manipulador.

Ao longo dos capítulos desta dissertação, descreve-se a completude dos objetivos específicos do projeto de pesquisa como etapas intermediárias à realização do objetivo geral. Estes objetivos são:

1. Elaborar a revisão do estado da arte presente na literatura relacionada, com ênfase nos dispositivos utilizados no experimento e nas técnicas algébricas utilizadas em trabalhos

recentes;

2. Realizar a modernização de dois manipuladores industriais para fins de experimentações práticas, compatíveis com o ROS, de código aberto, em uma bancada de laboratório que permita projetos de pesquisa futuros;
3. Revisar, aplicar e analisar técnicas de cinemática clássica em operações com manipuladores industriais, propondo um algoritmo para controlar o limite físico das juntas dentro da cinemática diferencial;
4. Estabelecer os princípios da álgebra de Clifford para cinemática, aplicando em situações comparativas aos métodos clássicos, e aplicar estes conceitos na representação de pontos e planos para descrição de um ambiente confinado através de sensores exteroceptivos;
5. Realizar experimentos de percepção externa do manipulador tanto em simulação quanto no protótipo real, de modo que as técnicas exploradas sejam aplicadas na adaptação de trajetória e modelagem do ambiente.

1.3 ESTRUTURA DO DOCUMENTO

A dissertação é estruturada de modo que os objetivos específicos sejam cumpridos ao longo de cada capítulo. A iniciar pelo Capítulo 2, realiza-se a revisão do estado da arte com a análise de trabalhos relacionados. O Capítulo 3 apresenta a descrição do processo de *retrofit* dos manipuladores utilizados durante o período de pesquisa e experimentação. O Capítulo 4 contém a aplicação de conceitos clássicos de cinemática onde também são explicitadas suas limitações. No capítulo 5 apresenta-se a utilização da álgebra de Clifford na cinemática do manipulador, além da apresentação das relações que permitem descrever o ambiente de trabalho com pontos e planos. No Capítulo 6 é apresentado o projeto que engloba todas as etapas parciais em ambiente tanto real quanto simulado. A dissertação se encerra com as considerações finais e a descrição de possíveis trabalhos futuros no Capítulo 7. As informações conceituais relevantes ao entendimento do projeto estão disponíveis nos apêndices, de forma a facilitar a leitura para familiares ao conteúdo e possibilitar a compreensão dos demais.

2 ESTADO DA ARTE

Este capítulo contém a análise dos trabalhos considerados de relevância publicados nas áreas de interesse do projeto, com foco nas publicações mais recentes. Pesquisas voltadas à álgebra de Clifford, a manipuladores com graus de redundância, planejamento de trajetórias livres de colisão, cinemática com limitações nas juntas, aplicação de *point cloud* na robótica e convenções matemáticas estão entre os temas abordados. O objetivo dos comentários presentes neste capítulo é demonstrar a atual relevância acadêmica dos temas estudados ao longo do programa de mestrado.

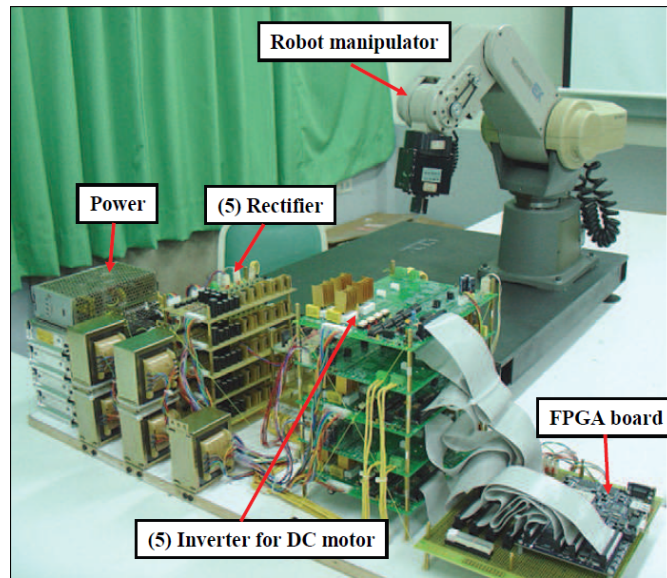
2.1 REVISÃO DOS TRABALHOS RECENTES RELACIONADOS

As principais áreas de concentração presentes no projeto desta dissertação são direcionadas à visão computacional através de detecção de cor e profundidade, aplicação de técnicas de cinemática a manipuladores e modelagem do ambiente para prevenção de colisões. Ao longo da implementação das técnicas foi necessária a reforma de dois manipuladores. Estes são descritos em detalhes ao longo do capítulo 3, onde demonstra-se que o principal alvo dos experimentos práticos é o manipulador Movemaster RV-M2, da fabricante Mitsubishi, referenciado a partir deste ponto do texto apenas como “Movemaster”. Apesar de ser um modelo antigo e desatualizado em referência a sensoriamento, controle e comunicação moderna, há aplicações onde busca-se a revitalização do robô.

No projeto de Huang e Wu (2009), o Movemaster recebe um novo controlador construído em chips FPGA (*Field Programmable Gate Array*) e um sistema de visão computacional para auxiliar na manipulação de objetos. O sistema também utiliza a cinemática através de parâmetros DH e lógica Fuzzy para gerar trajetórias para o efetuador. Outro processo de modernização do controlador foi feito em Kung e Chen (2008), onde novamente sistemas FPGA foram implementados para substituir o controlador original. Neste caso, a robustez do sistema de potência e das placas utilizadas na época resultaram num sistema grande, que apesar de funcional ocupa espaço considerável. O sistema resultante é mostrado na figura 2. Este sistema pode ser comparado em quesitos de portabilidade ao desenvolvido no capítulo 3.

Outras aplicações deste manipulador remetem ao uso em ambiente educacional, como nos trabalhos de Duong e Jaksic (2018) e Jaksic e Duong (2018), onde o robô foi usado com o

Figura 2 – Controlador para o Movemaster modernizado por Kung e Chen.



Fonte: Kung e Chen (2008)

intuito de ensinar conceitos de robótica em laboratório. Em trabalho mais recente, Huang et al. (2020) também realizam a modernização do Movemaster com lógica FPGA e controlador Fuzzy, desta vez aplicando a um sistema de controle mestre-escravo teleoperado.

De acordo com DeWitt et al. (2015), *exterocepção* envolve a percepção de estímulos externos que permitem que um organismo — no caso deste trabalho, um robô — tenha a capacidade de se adaptar ao seu ambiente, sendo uma das formas de relacionar os dados medidos por sensores a um conceito de consciência. A adição de sensoriamento exteroceptivo no formato de câmeras de cor ou profundidade é relevante para sistemas que interagem com o ambiente ou manipulam objetos. O sensoriamento utilizado nos primeiros trabalhos utiliza o Microsoft Kinect, o qual tem seu uso na robótica revisado no trabalho de El-laithy et al. (2012). O Kinect unificou a percepção de ambiente e a robótica fornecendo simultaneamente dados de cor e profundidade no formato de nuvem de pontos RGB-D (do inglês *Red, Green, Blue, Distance*, em tradução: vermelho, verde, azul e distância), conhecido como *point cloud*, e logo tornou-se compatível com o ROS (BOREN; COUSINS, 2011). O uso de *point clouds* na robótica tem clara aplicação em algoritmos de localização e mapeamento simultâneo (SLAM - *Simultaneous Location and Mapping*), onde combinado ao ROS gera trabalhos como o de Ruiz et al. (2013). Em aplicações envolvendo manipuladores seriais, o uso de nuvens de pontos permite a detecção de obstáculos na área de trabalho, como em Rakprayoon et al. (2011), onde a instalação de sensores de profundidade permitem a operação de um manipulador com a circulação segura de humanos. Neste caso, o sensor é instalado longe do manipulador em si, porém agindo como um

sistema integrado.

Os sistemas que utilizam Kinect logo tornaram-se obsoletos com o surgimento de câmeras dedicadas às aplicações em robótica como as da série Realsense. Na análise feita por Ahn et al. (2019), as câmeras como a modelo Realsense D435 permitem melhores desempenhos que o Kinect nos quesitos de faixa de preço, tamanho, peso, ruído e precisão de distância. As aplicações das câmeras D435 e D415, ambas modelos similares da Realsense, foram analisadas no trabalho de Tadic et al. (2019), onde há ênfase na sua aplicação na robótica.

Em trabalhos recentes, Liu et al. (2021) utilizam nuvens de pontos para estimar a pose de objetos e de um manipulador de seis graus de liberdade, onde é proposto um sistema de processamento de imagem utilizando os dados coletados. O termo “pose” refere-se à posição tri-dimensional de um objeto, envolvendo também a orientação sobre os seus eixos x , y e z (CRAIG, 2012). Chen et al. (2018) utilizam técnicas semelhantes para que um manipulador acoplado a uma base móvel possa identificar a pose de objetos para melhor manipulação. Neste caso também é feito um estudo relativo à métrica de manipulabilidade, a qual pode ser entendida como um objetivo secundário do robô. Outra aplicação da exterocepção através de nuvens de pontos é feita por Wong et al. (2021), onde a análise é feita em um manipulador serial redundante para melhorar a postura em uma atividade de coleta de objeto.

A combinação de ferramentas de geração de *point cloud* com rede de comunicação ROS também encontra aplicação em realidade aumentada, como descrito por Ostanin et al. (2020). Os autores utilizaram a representação do ambiente através de nuvens de pontos de forma que o usuário pôde controlar o manipulador em seu modelo virtual, onde algoritmos de geração de trajetória e desvio de colisão foram aplicados.

Além da possibilidade de modelar o ambiente com câmeras de cor e profundidade, a adição de um sistema proprioceptivo de odometria auxilia no monitoramento e controle dos movimentos realizados pelo robô, como no artigo de Bayer e Faigl (2019). Os autores combinam a câmera D435 com uma câmera de odometria modelo T265. A T265 é capaz de fornecer informações de orientação e deslocamento através do processamento de imagens coletadas do ambiente, permitindo informar os movimentos do robô a uma rede ROS. Neste caso a aplicação é feita em um robô móvel.

Um projeto semelhante na detecção de estruturas no ambiente aplicadas à robótica é visto na publicação feita por Choi et al. (2017). Neste trabalho o sensoriamento foi utilizado para modelar o ambiente no formato de planos matemáticos. A estratégia permitiu a criação de

trajetórias livres de colisão para robôs móveis. A aplicação demonstrou o potencial do uso de modelagens de ambiente no formato de planos, porém sem aplicação direta de suas propriedades algébricas.

A álgebra de Clifford permite modelar e relacionar elementos geométricos, podendo ser uma forma de descrição simplificada da percepção do ambiente. A álgebra de Clifford é uma ferramenta matemática extensa que engloba diversos campos da geometria e da representação de corpos rígidos. Seu uso é revisado em detalhes ao longo do capítulo 5. A publicação utilizando a representação de Clifford para pontos, linhas e planos feita por Huo et al. (2017) emprega esta ferramenta para computar o local de limitações mecânicas em um manipulador de seis graus de liberdade, utilizando linhas como eixos de movimentação. Suas conclusões demonstram a simplicidade da generalização permitida pelas fórmulas desta álgebra.

Trabalhos relevantes envolvendo a álgebra de Clifford na modelagem de objetos foram feitos em duas instâncias, citadas como Bayro-Corrochano e Lasenby (1995) e Bayro-Corrochano e Sommer (1995). No primeiro caso, um capítulo de livro foi dedicado à exposição da modelagem de objetos de forma genérica utilizando a álgebra de Clifford. Seu estudo expôs relações matemáticas para soluções de modelagem, ao oposto de focar na solução de problemas específicos. A publicação posterior utiliza as relações para demonstrar o potencial de aplicação em prevenção de colisões e detecção e manuseio de objetos. Apesar das literaturas serem claras quanto à aplicabilidade e simplicidade de uso, as notações utilizando bases geradoras facilmente torna-se confusa. Os artigos de Bayro-Corrochano e Falcón (2005), e posteriormente de Franchini et al. (2010), abordam estas dificuldades fornecendo uma revisão conceitual com exemplos visuais dos elementos da álgebra de Clifford. Os experimentos concluem que é justificada a aplicação em problemáticas dentro da robótica, e com o potencial de aplicação em visão computacional devido à sua elegância frente às laborosas formulações envolvendo o uso de matrizes.

A relação da álgebra de Clifford com o tratamento de fenômenos físicos estende-se além da visão computacional, como no processamento de imagens para detecção de movimentos de acordo com Lasenby et al. (1998), onde aplicam-se técnicas de reconstrução de objetos detectados em imagens. Recentemente, aplicações voltadas à eletrônica de potência surgiram nos artigos de Montoya e Eid (2021) e Montoya et al. (2021). Aplicações gerais na física são dadas no livro de Doran et al. (2003), passando por eletromagnetismo e física quântica e até mesmo redes neurais como nos trabalhos de Buchholz et al. (2007) e Hitzer (2013). A revisão dos elementos históricos são relatados nas teses de Oliveira (2011) e Breuils (2018).

Nos trabalhos de Kenwright (cita-se: Kenwright (2012) e Kenwright (2013)) é demonstrado como é possível realizar a definição completa dos quatérnios duais sem nenhuma relação com a álgebra de Clifford, sendo estas referências que, apesar de não serem as primeiras a abordar o tema, são didáticas para um estudo que pode se tornar confuso ao primeiro contato. O próprio autor reitera que o uso dos quatérnios duais apenas não é a norma dentro da cinemática por questões de familiaridade subjetiva dos métodos clássicos.

Os quatérnios duais são aplicados no estudo de Dooley e McCarthy (1993) para manipuladores colaborativos; de Daniilidis (1999) para mensurar as transformações de eixos de referência em um sistema com câmeras; aplicados na solução da cinemática inversa de um robô redundante por Gan et al. (2008); por Perez e McCarthy (2003) para a cinemática de manipuladores com restrições de movimento; na definição de um operador que combina quatérnios duais com a notação Denavit-Hartenberg no artigo de Radavelli et al. (2014b) e nas transformadas de manipuladores paralelos por Thiruvengadam et al. (2021), onde encontra-se extensa revisão matemática com ilustrações das propriedades. Johansen et al. (2019) compilam os resultados obtidos em diversas áreas do conhecimento utilizando quatérnios duais. Definir um quatérnio dual como caso especial da álgebra de Clifford é, no entanto, parte essencial da criação de um sistema unificado para representação da cinemática e do ambiente através de pontos, linhas e planos, como é discutido ao longo desta dissertação.

As propriedades de pontos, linhas e planos são aplicadas em exemplos de determinação de pose de manipuladores no capítulo de livro escrito por Radavelli et al. (2014a). Nesta referência é possível encontrar a descrição feita com notações de quatérnios duais, evitando a notação de bases geradoras que costuma ser confusa segundo os autores.

É possível utilizar a álgebra de Clifford para analisar regiões de singularidade. Para manipuladores paralelos, os trabalhos de Ma et al. (2018) e Cheng e Ji (2019) trazem soluções analíticas e as comparam com as clássicas soluções Jacobianas. Para manipuladores seriais, metodologias utilizando quatérnios duais na identificação das regiões singulares são descritas por Thiruvengadam e Miller (2020) e Zaplana et al. (2021), onde pode ser notada a complexidade matemática que os métodos permitem atingir.

No artigo de formato *survey* publicado por Hitzer et al. (2013) há a revisão das aplicações da álgebra de Clifford e seus avanços feitos desde o ano 1995, apesar de ser possível encontrar trabalhos envolvendo cinemática ao caso dos quatérnios duais desde os artigos de Funda e Paul (1990) e Ge e McCarthy (1990). A coleção de contribuições citadas neste trabalho englobam

redes neurais, processamento de imagem, visão computacional, cinemática e dinâmica robótica com aplicações em problemas de controle. O *survey* publicado por Breuils et al. (2022) é uma atualização das referências apresentadas por Hitzer, o qual é co-autor deste novo artigo. O intuito desta publicação foi revisar as aplicações da álgebra de Clifford nos 10 anos que se passaram desde a revisão anterior. As novas contribuições revisadas continuam envolvendo as mesmas áreas de estudo, porém com o uso de novas bibliotecas desenvolvidas em linguagens de programação usuais como M (para Matlab), Java, Javascript, C++ e Python. Há uma ênfase na contínua aplicação em robótica, especialmente em visão e cinemática. O fato de uma revisão bibliográfica contendo uma extensa lista de aplicações publicada no mesmo ano desta dissertação, 2022, evidencia o claro interesse acadêmico nos estudos da álgebra de Clifford aplicada a problemas de manipuladores robóticos.

2.2 CONSIDERAÇÕES

Os trabalhos de modernização envolvendo o Movemaster mostram que este manipulador ainda desperta interesse em aplicações educacionais que podem ser estendidas ao âmbito industrial. Apesar de ser recorrente a modernização do controlador e a implementação de sistemas de visão computacional, nenhum dos trabalhos menciona compatibilidade com ROS. Além disso, os controladores normalmente ocupam grande volume, o que pode dificultar a montagem de estações de trabalho. O espaço para contribuições se dá na construção de sistemas atualizados para aplicações que não envolvam apenas o meio acadêmico, mas que possam ser expandidas a outros manipuladores robustos em cenários industriais.

A utilização de câmeras de cor e profundidade que geram dados no formato de *point cloud* é um tema recorrente em aplicações de mapeamento, especialmente quando compatíveis com o ROS. O emprego das câmeras Realsense para esta função é a atual tecnologia mais empregada em literatura.

Ao que se refere a ferramentas matemáticas para descrição cinemática de manipuladores, a álgebra de Clifford fornece as técnicas mais eficientes e utilizadas nos trabalhos recentes, tornando as aplicações clássicas obsoletas a não ser por sua dificuldade de aprendizado - a não confundir com a dificuldade de implementação e clareza de resultados, a qual a literatura mostra ser a metodologia de preferência em todos os aspectos.

Em nenhum dos trabalhos revisados foi observada a aplicação de um sistema de representação da cinemática e do ambiente de forma unificada na álgebra de Clifford, com sistemas

de visão computacional e adaptação de trajetórias para prevenção de colisões compatível com o ROS. Não é clamada completude na análise devido à vasta quantidade de trabalhos existentes, como pode ser observado nos *surveys* de Hitzer et al. (2013) e Breuils et al. (2022), porém as aplicações e resultados desta dissertação tem clara contribuição aos trabalhos já existentes.

3 RETROFIT DOS MANIPULADORES

As técnicas e algoritmos desenvolvidos ao longo deste projeto aplicam-se no ambiente algébrico da cinemática. Para verificar o comportamento do robô fora do ambiente simulado é necessário preparar um cenário real que permita a validação dos resultados. As abordagens propostas são implementadas em camadas tradicionalmente proprietárias e fechadas de controladores comerciais de manipuladores, o que motivou a modernização e concepção de uma arquitetura de controle totalmente aberta e compatível com os padrões atuais da área de robótica. Além disso, o ambiente prático leva as incertezas do mundo real, as quais não podem ser observadas no simulador.

Este capítulo trata da descrição dos manipuladores usados tanto nas etapas intermediárias quanto finais de experimentação prática. Duas estruturas passaram pelo processo de modernização, ou *retrofit*: um manipulador planar de três graus de liberdade, denominado ORCA, e um manipulador de cinco graus de liberdade modelo Movemaster RV-M2. Ambos foram adaptados para trabalhar dentro de uma rede de tópicos ROS com sistemas de percepção. O processo de *retrofit* consiste em reformar robôs desatualizados tecnologicamente para sistemas e técnicas mais recentes, permitindo a continuidade de aplicações didáticas curriculares unindo o baixo custo de estruturas já presentes em laboratório e a multidisciplinaridade da tarefa de reforma (LYMAN et al., 2019).

A metodologia para o *retrofit* de um manipulador é específica a cada caso. A abordagem depende do estado da máquina disponível e de que sistemas serão adicionados e modernizados. Em linhas gerais, deseja-se ter acesso em pontos terminais a todas as informações internas do manipulador, isto é, deve ser feito o levantamento — através de inspeção ou por meio de manuais — de como é o funcionamento do sistema de posicionamento das juntas e de que tipos de sensoriamento o manipulador já é equipado originalmente.

Uma vez que estabelecem-se os pontos do circuito original que fornecem as informações internas do robô, também deve-se encontrar os pontos que recebem os comandos de movimentação, bem como as alimentações dos motores. Isso permite a execução de tarefas básicas de envio e recebimento de dados, ou seja, permite a implementação de técnicas de controle de junta. Estas etapas envolvem uma modernização que visa a troca de controladores proprietários por sistemas abertos ou de desenvolvimento particular. Caso a situação em questão não envolva a troca dos sistemas de controle, também pode ser necessário o desenvolvimento de ferramentas

que entendam ou modifiquem a linguagem de comunicação existente — novamente, isso depende da estrutura, estado e objetivo do manipulador, portanto não é possível generalizar a metodologia a todos os casos.

Ao seguir com a modernização dos manipuladores, deve-se tomar as devidas precauções com as mudanças de parâmetros cinemáticos. A exemplo do que será relatado ao longo deste capítulo, a inserção de sistemas que não englobam o projeto original do robô podem alterar seus parâmetros cinemáticos, como a troca da ferramenta de efetuador. Também há o fato de que comumente os manipuladores são projetados de forma que seja evitada ou dificultada a colisão da estrutura consigo própria, o que pode não se manter com a adição de câmeras ou outros sensores do tipo. Para este tipo de cenário, o projeto deve levar em conta como o robô lida com sensoriamentos de fim-de-curso ou auto-colisão, também sendo necessário identificar os pontos do circuito que contém tal informação para que seja tratada externamente. Os manipuladores descritos neste capítulo passam por esta alteração.

Este capítulo contém a descrição dos processos adotados para o caso disponível em laboratório, o qual envolveu a troca do controlador proprietário por um sistema aberto desenvolvido ao longo do programa. Relata-se, então, os processos intermediários e como se dá o funcionamento do sistema final modernizado.

3.1 MANIPULADOR BECKMAN COULTER ORCA

O manipulador ORCA (Beckman Culter) tem como propósito original aplicações na área farmacêutica para manipulação de utensílios de laboratório, a exemplo do trabalho de Lindhagen et al. (2008). Sua estrutura original possui cinco graus de liberdade e uma ferramenta do tipo garra em seu efetuador, podendo ser observado na figura 3. O manipulador também é dotado de controlador próprio e é classificado como parte de um sistema de analisadores químicos automatizados (MedWOW, 2022).

Originalmente, os graus de liberdade do manipulador ORCA são referidos pela seguinte nomenclatura, seguindo numeração descrita na figura 3 e conforme catalogado em MedWOW (2022):

1. *rail*: Junta prismática para ajuste da posição linear da base;
2. *height*: Junta rotacional para a altura;
3. *reach*: Junta rotacional para o alcance;

Figura 3 – Manipulador Beckman Coulter ORCA original.



Fonte: Adaptado de MedWOW (2022).

4. *bend*: Junta rotacional para a inclinação;
5. *twist*: Junta rotacional para movimentos de centrifugação ou ajuste de orientação;
6. *grip*: efetuador com garra.

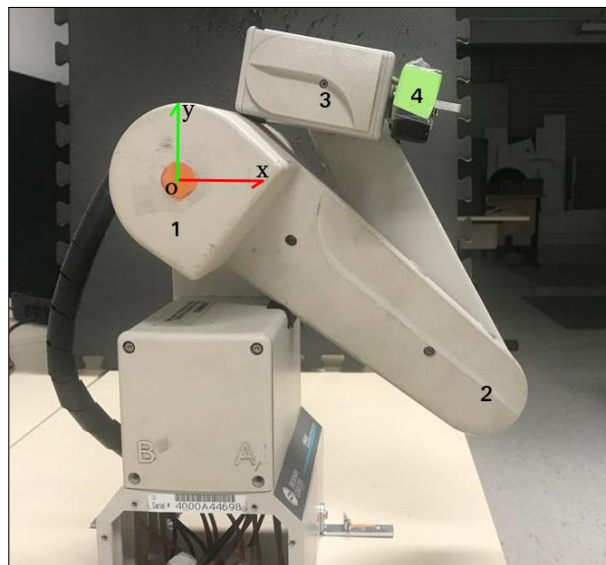
O protótipo disponível para o presente projeto não conta com a junta prismática e a junta responsável pelo *twist* não está operante. Esta situação faz com que o manipulador seja limitado aos graus de *height*, *reach* e *bend*, tornando-se planar e com três graus de liberdade. O controlador não está disponível, além de ser de interesse criar um controlador próprio de fonte aberta para envio de comandos através do ROS.

Para contextualização, cabe aqui uma breve descrição do que é e no que se aplica o ROS, do inglês *Robot Operating System*, ou Sistema Operacional de Robôs, é um conjunto de bibliotecas de *software* e ferramentas que ajudam a criar aplicações em robótica (Open Robotics, 2021). O ROS é uma rede de pacotes que trabalham tanto na parte de programação quanto na comunicação entre agentes de um sistema, chamados de “nós”, podendo estes ser robôs, sensores ou até mesmo rotinas executadas em um *prompt* de comando. Estes nós comunicam-se através de “mensagens” que são enviadas e lidas em um sistema de “tópicos”. Cada nó que publica informações, o faz em um tópico criado especificamente para armazenar esta mensagem. Os nós que precisam de informações para realizar suas tarefas o fazem inscrevendo-se neste

tópico. Cada nó do sistema pode ter relações de posicionamento no ambiente entre si, cuja cadeia é denominada “árvore de transformações”. O ROS fornece ferramentas de diversas áreas da tecnologia desenvolvidos para robótica, sempre utilizando código aberto para desenvolvimento. Maiores detalhes podem ser verificados no site oficial da plataforma¹.

A figura 4 apresenta o protótipo do ORCA utilizado para o *retrofit* em sua configuração de juntas inicial, ou *home*. A origem do sistema de coordenadas estabelecido também está representado.

Figura 4 – Manipulador ORCA antes do *retrofit*.



Fonte: Autoria própria.

Como a cadeia cinemática do ORCA disponível é diferente da original, refere-se às juntas como um análogo de braço antropomórfico, conforme numeração da figura 4 e seguinte descrição:

1. Ombro: Junta rotacional para a altura;
2. Cotovelo: Junta rotacional para o alcance;
3. Punho: Junta rotacional para a inclinação;
4. Garra: efetuador com garra a ser adaptada.

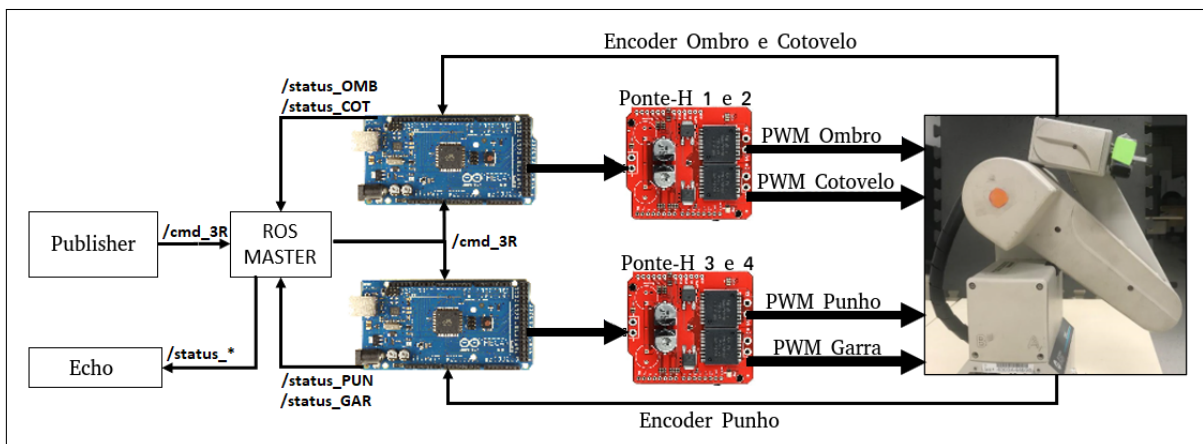
O projeto inclui um sistema de controle de ângulo de junta utilizando técnicas de controle proporcional-derivativo (PD) digital. Foram utilizados dois microcontroladores Arduino

¹ Disponível em: <https://www.ros.org/>.

e cada um foi acoplado a um *shield*, que possui duas ponte-H modelo VNH3SP30 cada (ST MICROELECTRONICS, 2017). Um Arduino é responsável pelo controle das juntas Ombro e Cotovelo e o outro controla Punho e Garra.

Os *setpoints* dos ângulos são enviados através do tópico ROS `/cmd_3R`, onde são processados por *subscribers* em cada Arduino. Estes ângulos são convertidos para a notação de pulsos de *encoder*, a qual serve de métrica de posição das juntas medida através dos *encoders* de cada motor. A velocidade das juntas é controlada via PWM (*Pulse Width Modulation*) como saída do controle PD. Ao receber a informação dos *encoders*, o controlador é atualizado e a informação é publicada nos tópicos ROS de monitoramento `/status_*`, onde entende-se por *** como um tópico para cada junta. Esta informação pode ser visualizada por qualquer sistema capaz de se inscrever no ROS Master (um nó *echo*, por exemplo). Um esquemático do sistema de controle pode ser observado na figura 5.

Figura 5 – Esquema de controle e comunicação ROS para o ORCA.



Fonte: Autoria própria.

As mensagens ROS para os tópicos `/cmd_3R` são do tipo `/custom_msg/set_angles` e para os tópicos `/status_*` são do tipo `/custom_msg/status_arm`. Estas mensagens não fazem parte do pacote padrão do ROS e foram criadas para este projeto. O quadro 1 lista os campos da mensagem tipo `/custom_msg/set_angles` e o quadro 2 faz o mesmo para a mensagem tipo `/custom_msg/status_arm`, ambos acompanhados do detalhamento de aplicação de cada campo.

A contagem de pulsos dos *encoders* é realizada através de interrupções nos Arduino. Devido às características do processador e ao estado do dispositivo foi necessário calibrar a relação de pulsos dos *encoders* para graus de cada junta. O procedimento foi feito por meio de repetidas medições a cada junta com um ângulo de referência de 90 graus. Posicionando cada junta em uma referência de 0 graus e realizando a contagem dos pulsos até os 90 graus

Quadro 1 – Campos da mensagem /custom_msg/set_angles.

Nome do campo	Tipo	Aplicação
set_OMB	float32	Define o <i>setpoint</i> da junta Ombro, em graus.
set_COT	float32	Define o <i>setpoint</i> da junta Cotovelo, em graus.
set_PUN	float32	Define o <i>setpoint</i> da junta Punho, em graus.
set_GAR	bool	Define o <i>setpoint</i> da Garra, sendo <code>true</code> para garra aberta e <code>false</code> para garra fechada.
RESET	bool	Faz o manipulador retornar à configuração <i>home</i> caso o valor seja <code>true</code> . Deve ser a primeira mensagem ao energizar o sistema.
RETRY	bool	Faz o mesmo que a função RESET e em seguida repete a última configuração. Utilizada para lidar com erros de operação onde uma nova tentativa é necessária.
EMERGENCY_STOP	bool	Quando uma mensagem é enviada com este campo preenchido com valor <code>true</code> , o sistema para imediatamente.

Fonte: Autoria própria.

Quadro 2 – Campos da mensagem /custom_msg/status_arm.

Nome do campo	Tipo	Aplicação
junta	string	Texto que define a identificação da junta.
pulsos_setpoint	float32	Valor do atualmente buscado pelo controlador.
pulsos_contados	float32	Valor atual da variável de controle.
output_P	float32	Saída da parcela proporcional do controlador.
output_I	float32	Saída da parcela integral do controlador. Atualmente desativada.
output_D	float32	Saída da parcela derivativa do controlador.
output_PID	float32	Somatório das saídas do controlador, representando o PWM.
loop_time	float32	Tempo de integração/derivação do controlador.
IsDone	bool	Se <code>true</code> , a junta está posicionada no <i>setpoint</i> dentro da tolerância.

Fonte: Autoria própria.

foi possível obter a contagem de pulsos realizada. Em uma média de 10 medições para cada junta, obteve-se a relação descrita na tabela 1. A garra não possui controle via *encoders*. Também apresentam-se os ângulos da configuração *home*, seguindo o eixo cartesiano da figura 4.

Tabela 1 – Relação de pulsos por grau e configuração *home* do ORCA.

Junta	Pulsos por grau	<i>home</i>
Ombro	200	-40°
Cotovelo	290	145°
Punho	148	-133°

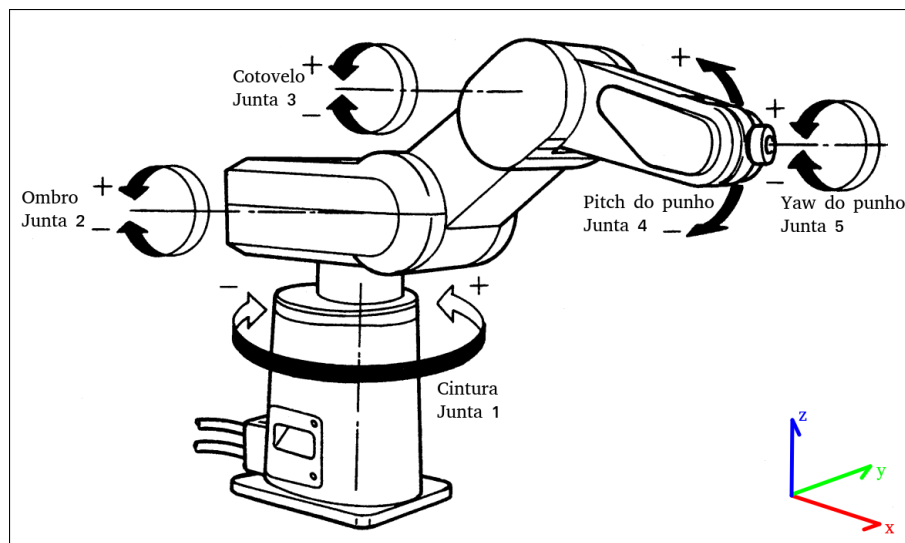
Fonte: Autoria própria.

O manipulador agora é capaz de se comunicar com uma rede ROS e utiliza controladores *open-source* para realizar seus movimentos. O ORCA modernizado foi utilizado no decorrer deste projeto para realizar testes de performance e validar algoritmos nas suas versões simplificadas, conforme será discorrido ao longo dos capítulos seguintes. A natureza didática de um manipulador planar de 3 graus de liberdade foi utilizada de maneira intermediária à aplicação avançada de conceitos no manipulador Movemaster, o qual tem foco nos principais resultados.

3.2 MANIPULADOR MITSUBISHI MOVEMASTER RV-M2

O manipulador Movemaster RV-M2 é um manipulador serial de cinco graus de liberdade utilizado em tarefas gerais de manipulação de objetos, a exemplo dos trabalhos realizados por Huang e Wu (2009) e Kung e Chen (2008). Sua estrutura original possui cinco graus de liberdade definidos conforme figura 6. Neste documento as juntas do Movemaster serão referenciadas pelo seu número de 1 a 5.

Figura 6 – Graus de liberdade do manipulador Movemaster original.



Fonte: Adaptado de Mitsubishi Electric Corporation (1990)

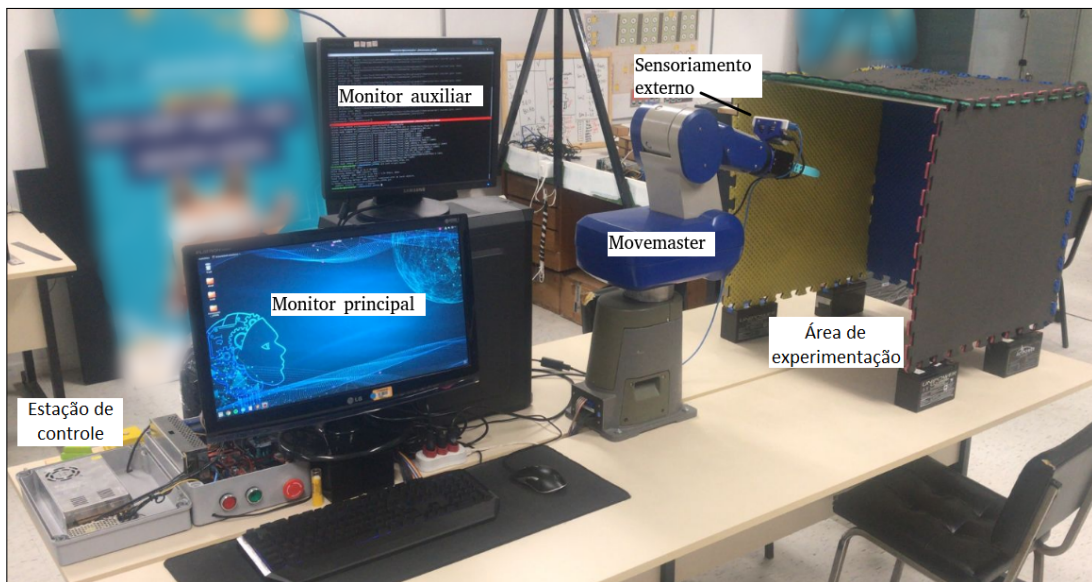
O protótipo disponível em laboratório encontrava-se fora de condições de uso e passou por um processo de *retrofit*, recuperando seu funcionamento e adicionando tecnologias de coleta de dados exteroceptivos. A figura 7 contém o Movemaster em seu estado de recebimento, antes da realização das melhorias. O processo de modernização seguiu com a reforma da pintura da carcaça do manipulador, apenas com objetivos estéticos. A estrutura foi fixada em uma bancada de laboratório de forma que permita a realização de todos os experimentos. O posicionamento do robô e do computador onde o controle é realizado é apresentado na figura 8.

Figura 7 – Movemaster antes do retrofit.



Fonte: Autoria própria.

Figura 8 – Área de operação em laboratório do Movemaster.

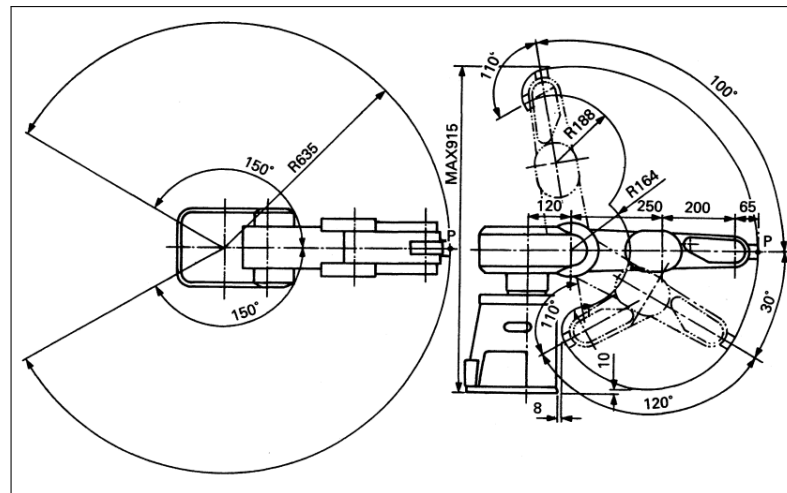


Fonte: Autoria própria.

O robô possui sensores do tipo fim-de-curso em todas as juntas, o que permite o controle sem o risco de colisões com a própria estrutura. Esta característica pode ser observada na figura 9, onde o manual descreve o espaço operacional do Movemaster. Nota-se, no entanto, que o volume de alcance do robô não toca o chão caso não exista uma ferramenta acoplada, porém esse risco existe ao adicionar elementos ao efetuador.

O processo de modernização iniciou com a remoção de placas internas do manipulador de modo que a conexão direta com os sensores e motores tornou-se exposta. Com o auxílio de testes de condutividade, realizou-se o mapeamento dos conectores de interesse. O objetivo desta etapa é ter acesso aos atuadores e sensores para conectá-los a uma unidade de controle

Figura 9 – Espaço operacional original do Movemaster.



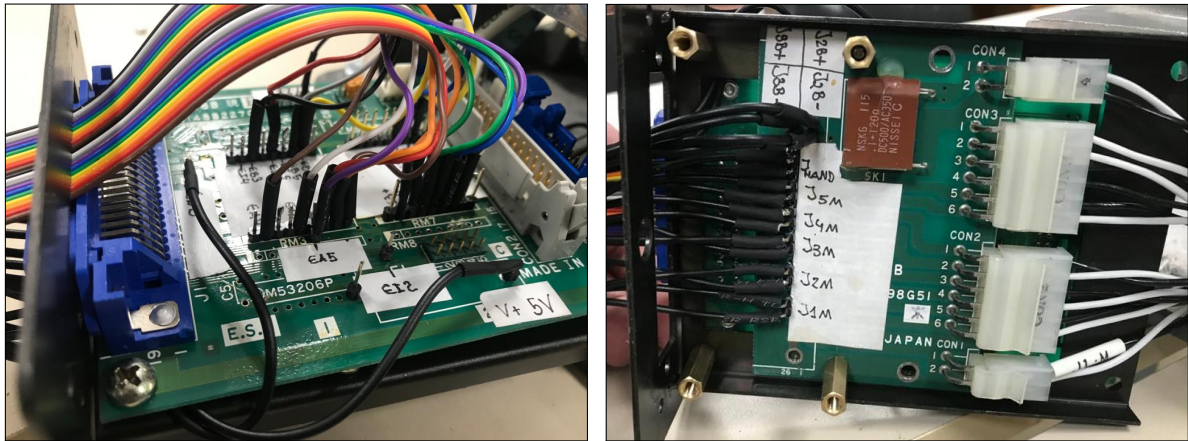
Fonte: Mitsubishi Electric Corporation (1990)

desenvolvida em laboratório. As conexões de interesse são listadas a seguir, onde informações básicas de construção são obtidas no manual da Mitsubishi Electric Corporation (1990):

- Duas conexões para PWM de cada motor. A velocidade dos motores é controlada pelo *duty cycle* de um PWM e a direção é controlada alterando qual lado do motor recebe GND e qual recebe o sinal;
- Pinos de alimentação positivo e GND dos *encoders*, os quais devem ser energizados com 5V contínuos;
- Pinos de alimentação positivo e GND dos sensores fim-de-curso, sendo dois sensores para cada junta, exceto a garra. Esta alimentação também é feita com 5V contínuos;
- Pinos de sinal A e B dos *encoders* de cada junta, exceto a garra. São cinco *encoders* de quadratura, o que permite acessar a informação de quanto uma junta rotacionou e também sua direção;
- Pinos de sinal dos sensores fim-de-curso, sendo dois sensores por junta, exceto a garra. O pino de sinal deve ser conectado a um resistor de *pull-up*, uma vez que a transição do sinal binário do sensor não é linear;
- Alimentação positiva e negativa dos freios das juntas 2 e 3, os quais devem ser alimentados com 12V contínuos para que o freio seja liberado. A frenagem é binária, não permitindo amortecimento.

A figura 10(a) mostra a placa interna onde as conexões dos sensores convergem e suas respectivas alimentações. Originalmente estas conexões não são acessíveis, uma vez que são ligadas ao cabo que leva ao controlador da fabricante. A soldagem apresentada conecta os sensores fim-de-curso e *encoders* a um cabo confeccionado que leva a informação ao controlador. De forma semelhante, os cabos que levam o PWM aos motores e a alimentação dos freios foram ligados a um cabo externo, conforme figura 10(b). A figura 11 mostra como a saída dos cabos foi adaptada para contornar a ausência do cabo original. A fiação possui 80 cm externos ao robô até a estação de controle.

Figura 10 – Adaptação na conexão dos sensores do Movemaster.



(a) Placa de conexão dos sensores.

(b) Placa de conexão dos motores e freios.

Fonte: Autorial própria.

Figura 11 – Novas conexões externas na traseira do Movemaster.



Fonte: Autorial própria.

Devido à simplicidade de programação e ao baixo custo, utilizou-se microcontroladores Arduino para realizar o processamento do controle do manipulador. Foram instalados três Arduinos acoplados a *shields* de Ponte-H da mesma forma que o realizado com o ORCA. A estação de controle foi montada de modo a comportar os três microcontroladores conectados a um *hub* USB.

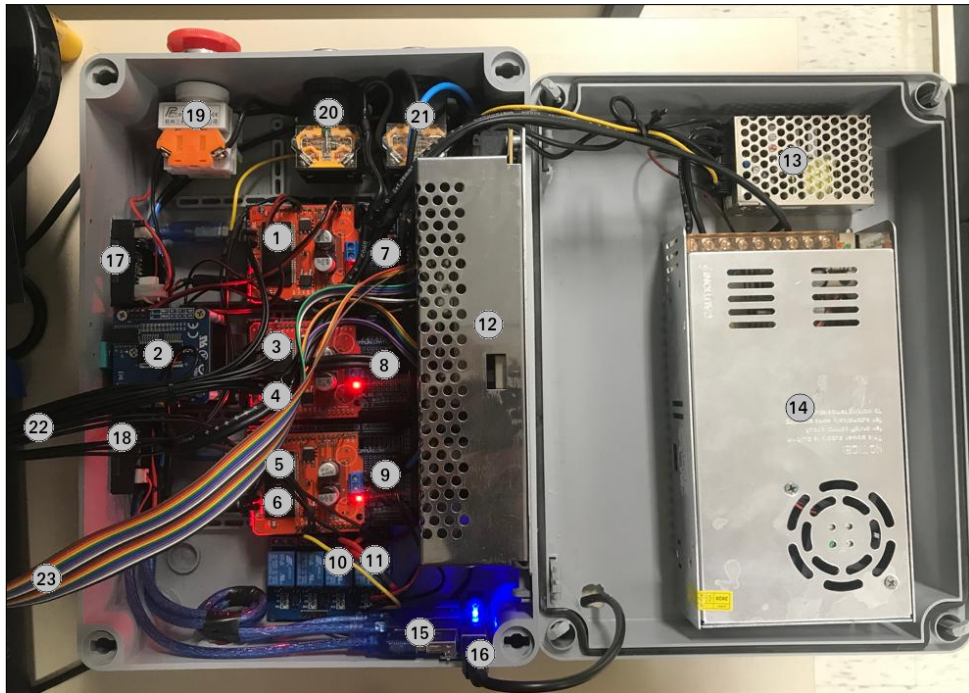
Foi instalada uma fonte de 12V para alimentação dos motores. Apesar dos motores serem fabricados para trabalhar com 36V, este valor os faz trabalhar em uma velocidade alta em relação ao pretendido em experimento. Foi escolhida uma alimentação de 12V pois mesmo com 100% de *duty cycle* o movimento da estrutura não oferece riscos ao operador. Esta alimentação, no entanto, não é suficiente para movimentar livremente a junta 2, pois ela precisa suportar todo o peso do manipulador dependendo da pose do mesmo.

Para solucionar o problema de movimentação da junta 2, optou-se por instalar uma fonte de alimentação de 24V dedicada a ela. Acarreta desta alteração que a ponte-H responsável por controlar este motor deve ser trocada, pois as pontes VNH3SP30 suportam alimentação até 16V, segundo documentação do fabricante ST Microelectronics (2017). Este problema foi solucionado com a instalação de duas meia-ponte-H modelo BTS7960, de documentação disponibilizada pelo fabricante Infineon (2004), a qual suporta alimentação de até 45V e conduz até 45A. A alteração da ponte-H permitiu trabalhar com maior liberdade na janela do PWM, uma vez que o sinal máximo leva uma alimentação de 24V para o motor, resultando em movimentação da junta de acordo com o esperado para experimentos práticos.

A estação do controlador foi construída com um botão de emergência de fácil acesso para o operador, bem como um multímetro e um amperímetro que permitem visualizar a tensão e corrente atual, com o objetivo de informar situações que exigem ação emergencial para prevenção de danos ou acidentes. Também há botões que acionam de maneira independente as fontes de 12V e 24V, bem como uma fonte de alimentação de 5V, instalada para realizar a alimentação independente dos sensores.

Ainda no controlador, uma placa de relés foi instalada para controlar o acionamento dos freios presentes nas juntas 2 e 3. Os relés são alimentados com 12V e um sinal digital proveniente dos Arduinos é responsável por acionar ou não os freios. Quando os relés não estão conduzindo, os freios estão acionados. A figura 12 mostra a estrutura do controlador que comporta os blocos funcionais aqui descritos. A descrição dos componentes presentes no controlador é apresentada no quadro 3.

Figura 12 – Estação de controle do Movemaster modernizado.



Fonte: Autoria própria.

Quadro 3 – Descrição dos componentes no controlador do Movemaster.

Item	Descrição
1 a 6	Ponte-H das juntas 1 a 5 e garra
7 a 9	Conjunto de resistores <i>pull-up</i> .
10 e 11	Relés dos freios das juntas 2 e 3.
12	Fonte de alimentação de 12V.
13	Fonte de alimentação de 5V.
14	Fonte de alimentação de 24V.
15	Hub USB para os Arduinos.
16	Porta USB para comunicação com o PC.
17	Display voltímetro.
18	Display amperímetro.
19	Botão de emergência.
20 e 21	Botões para acionar as fontes de alimentação.
22	Cabos de alimentação dos motores, freios e sensores.
23	Cabos de sinal dos sensores.

Fonte: Autoria própria.

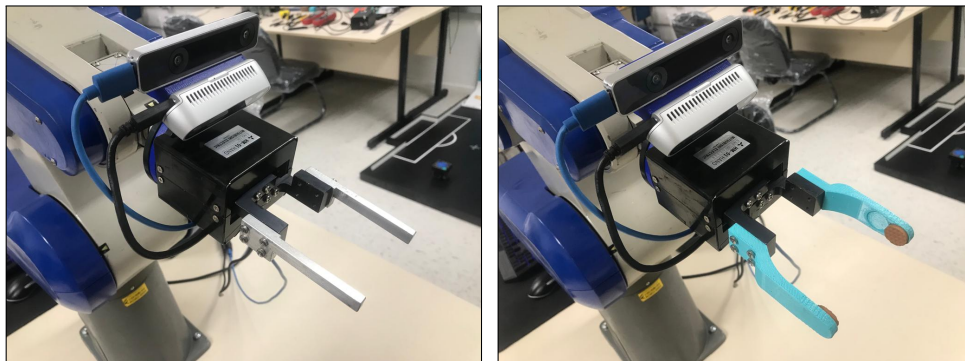
O sistema de controle é realizado por cada uma das juntas independentemente dentro dos Arduinos. A técnica utilizada foi um controlador PID básico, onde é possível agir e alterar o valor do PWM de maneira proporcional, integral e derivativa ao erro atual. Os ganhos foram definidos de acordo com testes realizados em bancada de modo que os movimentos fossem suaves em aceleração e desaceleração — ajustando a parcela derivativa — além de ser capaz de vencer a inércia para pequenos ajustes — ajustando a parcela integral — e garantindo erro nulo em regime permanente. A tabela 2 mostra os ganhos k para as parcelas P, I e D de cada junta.

Tabela 2 – Ganhos dos controladores PID de cada junta.

Junta	kP	kI	kD
J_1	0,06	0	0,3
J_2	0,03	0,02	0,5
J_3	0,03	0,04	0,6
J_4	0,03	0,05	0,05
J_5	1	0	0

Fonte: Autoria própria.

O efetuador do Movemaster permite que seja acoplado uma garra, a qual foi adaptada através de impressão 3D em material ABS. A alteração da estrutura da garra visa uma melhoria na capacidade do robô de interagir com objetos, tendo um alcance e um ângulo máximo de abertura maior que o original. A ferramenta original é apresentada na figura 13(a), enquanto a adaptação instalada é mostrada na figura 13(b).

Figura 13 – Garra do Movemaster original e modernizada.**(a) Original.****(b) Modernizada, em impressão 3D.**

Fonte: Autoria própria.

Parte do objetivo da modernização do Movemaster é dar ao robô capacidade de percepção do ambiente e de si mesmo. Para isso, foi projetado um suporte adaptador para uma câmera de profundidade e uma de odometria. A figura 14 mostra o suporte impresso em ABS instalado.

Figura 14 – Suporte para as câmeras de profundidade e odometria.

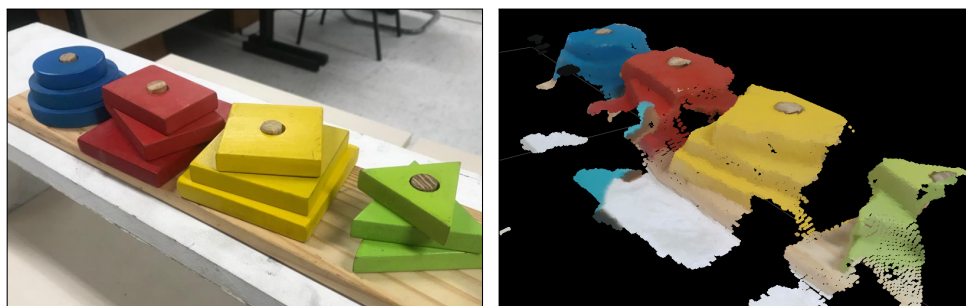
Fonte: Autoria própria.

A câmera de profundidade é do modelo Realsense D435, produzida pela Intel Corporation (2022a). Esta câmera realiza a coleta de informações RGB-D, ou seja, dados de cor e profundidade. A resolução do sensor de profundidade é de, no máximo, 1280×720 a até 90 quadros por segundo e da imagem RGB pode chegar a 1920×1080 a até 30 quadros por segundo. Esta câmera foi escolhida para o projeto devido ao seu alcance. Como o manipulador é fixo em sua área de trabalho, não é necessário utilizar câmeras de longo alcance. O alcance de precisão segundo o fabricante é de 0,3 m até 3 m, compatível com a área operacional do Movemaster.

Para utilizar a D435 com a rede ROS, basta utilizar o pacote *realsense2*² desenvolvido e mantido pela própria Intel Corporation, o qual também é de código aberto. Ao especificar no arquivo *launcher* a variável *enable_pointcloud* como `true`, a câmera D435 passa a publicar no tópico */d400/depth/color/points* no formato *sensor_msgs/PointCloud2*, contendo a informação de cor e profundidade simultânea do ambiente. Por padrão do pacote, também são publicados tópicos separados para os dados de profundidade e imagem RGB, aplicáveis dependendo da atividade realizada.

A capacidade de capturar um *point cloud* do ponto de vista do efetuador faz com que o manipulador tenha a percepção do ambiente em que atua. Esta informação é utilizada neste projeto para delimitar o espaço de trabalho com limitações externas, conforme será demonstrado no Capítulo 6. Na figura 15 é ilustrado um exemplo da captura de *point cloud*, utilizando o pacote de visualização padrão do ROS, o Rviz.

Figura 15 – *Point cloud* capturado pela câmera D435.



(a) Cenário real.

(b) *Point cloud* capturado.

Fonte: Autoria própria.

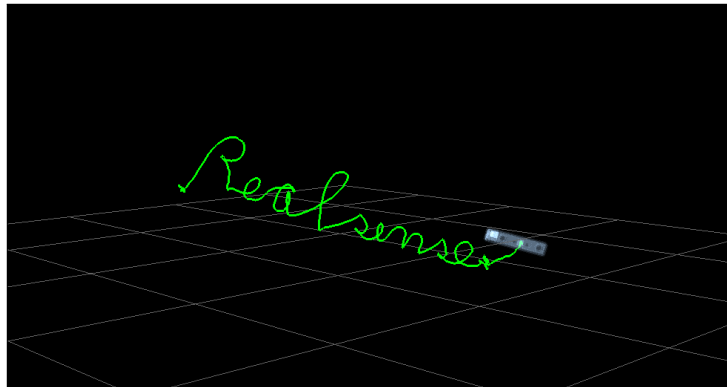
Para a parcela proprioceptiva do manipulador, a estrutura original já conta com o sistema de *encoders* em cada junta. É possível saber a localização de todas as juntas e do efetuador apenas com a informação do ângulo das juntas utilizando técnicas de cinemática direta. Esta informação é acurada porém é dependente da precisão dos parâmetros cinemáticos e é afetada por

² Disponível em <https://github.com/IntelRealSense/realsense-ros>

imperfeições mecânicas e de processamento de dados. Como ação para aprimorar a capacidade de percepção da própria estrutura, foi acoplado ao efetuador uma câmera de odometria.

A câmera de odometria modelo Realsense T265, também produzida pela Intel Corporation (2022b), usa técnicas de SLAM para rastrear o deslocamento através do processamento das imagens capturadas, associado a um módulo IMU (do inglês: *Inertial Measurement Unit*, ou Unidade de Medição Inercial). Dessa forma, ao acoplar a câmera junto ao efetuador, informações de deslocamento e orientação são acessíveis durante a execução das tarefas. O pacote *realsense2* também fornece compatibilidade entre a T265 e o ROS, fazendo com que as informações de odometria sejam publicadas no tópico `/t265/odom/sample` no formato `nav_msgs/Odometry`. O intuito de utilizar a T265 é verificar a odometria do efetuador, permitindo o monitoramento em tempo de execução. A figura 16 mostra um exemplo da odometria sendo publicada, onde o fabricante exemplifica o funcionamento da câmera escrevendo o nome do produto no ar, o qual é visualizado no computador.

Figura 16 – Câmera de odometria Realsense T265 em operação.



Fonte: Grunnet-Jepsen et al. (2019)

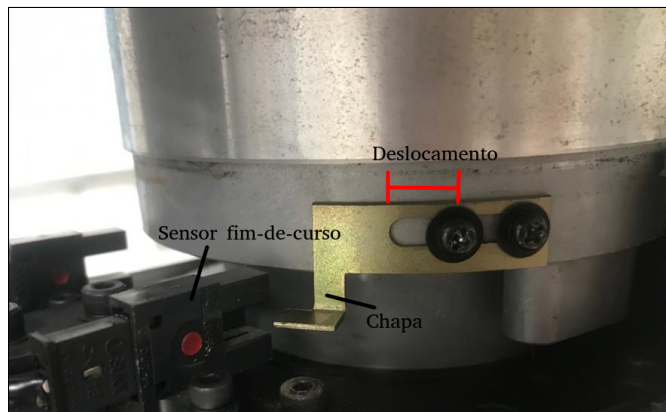
Para utilizar as câmeras D435 e T265 numa rede ROS simultaneamente, é necessário especificar os detalhes de inicialização e nomenclatura de nós e tópicos através do arquivo *launcher*. O pacote *realsense2* possui o arquivo `rs_d400_and_t265.launch`, o qual detecta e inicializa automaticamente ambas as câmeras. O arquivo é pré-configurado e, por ser de código aberto, serve de base para o projeto final desta dissertação.

Como consequência da instalação das câmeras junto do efetuador, surge o risco de colisão com a própria estrutura ao movimentar a junta 4. Outra situação envolvendo riscos de colisão é a área de trabalho onde o manipulador está fixo, como foi mostrado na figura 8. Destas observações surge a necessidade de alterar os limites de junta.

Como o computador e o operador ficam localizados na parte posterior do Movemaster,

deve-se limitar os movimentos horário e anti-horário da junta 1 para uma área segura. Ao observar a instalação dos sensores de fim-de-curso originais, nota-se que é possível inverter a forma como as chapas de acionamento estão acopladas. A figura 17 mostra a alteração realizada na fixação da chapa que aciona ambos os fim-de-curso da junta 1, o que faz com que os sensores sejam ativados com um ângulo de junta menor em ambas as direções. Mesmo com a limitação devido ao acionamento do sensor, os algoritmos desenvolvidos ao longo deste projeto limitam a junta 1 em $\pm 90^\circ$, por razões de segurança.

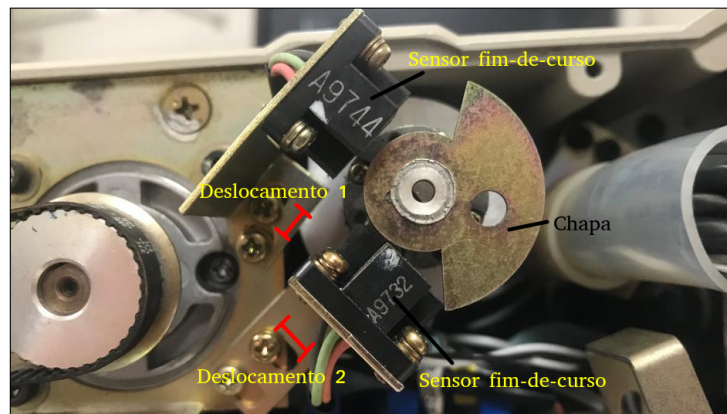
Figura 17 – Nova limitação para a junta 1 do Movemaster.



Fonte: Autoria própria.

De forma semelhante, as chapas de acionamento dos sensores de fim-de-curso também podem ser movidas de modo que a rotação no sentido anti-horário da junta 4 acione o sensor com um ângulo menor. Esta alteração muda o limite da junta para $\pm 100^\circ$, evitando a colisão do suporte das câmeras com o corpo do manipulador. A alteração do sensor é mostrada na figura 18. Como já destacado, não há maneira direta de evitar a colisão com a bancada na altura da base, onde a posição z do efetuador é $z_{\text{eff}} \leq 0$. Para tal caso, é necessária a ação do controlador.

Figura 18 – Nova limitação para a junta 4 do Movemaster.



Fonte: Autoria própria.

Com as alterações realizadas, os limites de junta para o Movemaster modernizado são dadas conforme tabela 3, em graus. Os sentidos horário e anti-horário são definidos de acordo com a regra da mão direita aplicada aos eixos cartesianos representados na figura 6. A substituição da garra original também gera alteração nos parâmetros cinemáticos do robô, os quais serão descritos em detalhes no Capítulo 4.

Tabela 3 – Limites de junta do Movemaster modernizado.

Junta	Limite horário	Limite anti-horário
J_1	-100°	100°
J_2	-30°	100°
J_3	-120°	0°
J_4	-100°	100°
J_5	-180°	180°

Fonte: Autoria própria.

Um pacote de mensagens customizadas foi criado para a comunicação do robô com a rede ROS, de forma semelhante ao realizado com o ORCA. Os quadros 4 e 5 listam os campos das mensagens do tipo *movemaster_msg/setpoint* e */movemaster_msg/status*, respectivamente.

Quadro 4 – Campos da mensagem */movemaster_msg/setpoint*.

Nome do campo	Tipo	Aplicação
set_1	float32	Define o <i>setpoint</i> da junta 1, em graus.
set_2	float32	Define o <i>setpoint</i> da junta 2, em graus.
set_3	float32	Define o <i>setpoint</i> da junta 3, em graus.
set_4	float32	Define o <i>setpoint</i> da junta 4, em graus.
set_5	float32	Define o <i>setpoint</i> da junta 5, em graus.
set_GRIP	bool	Define o <i>setpoint</i> da Garra, sendo <i>true</i> para garra aberta e <i>false</i> para garra fechada.
GoHome	int8	Definida como um inteiro que aciona diferentes rotinas: caso setada como 1, o robô volta à configuração <i>home</i> (reset), caso setada em 2, retorna à configuração <i>home</i> e repete a última ação enviada (retry), caso 0, não executa nenhuma ação.
emergency_stop	bool	Quando uma mensagem é enviada com este campo preenchido com valor <i>true</i> , o sistema para imediatamente.

Fonte: Autoria própria.

Quadro 5 – Campos da mensagem */movemaster_msg/status*.

Nome do campo	Tipo	Aplicação
joint	string	Texto que define a identificação da junta.
setpoint	float32	Valor do atualmente buscado pelo controlador.
pulse_count	float32	Valor atual da variável de controle.
error	float32	Erro atual do sistema de controle.
output	float32	Valor do PWM.
control_loop	float32	Tempo de integração/derivação do controlador
IsDone	bool	Se <i>true</i> , está dentro da tolerância do <i>setpoint</i> .

Fonte: Autoria própria.

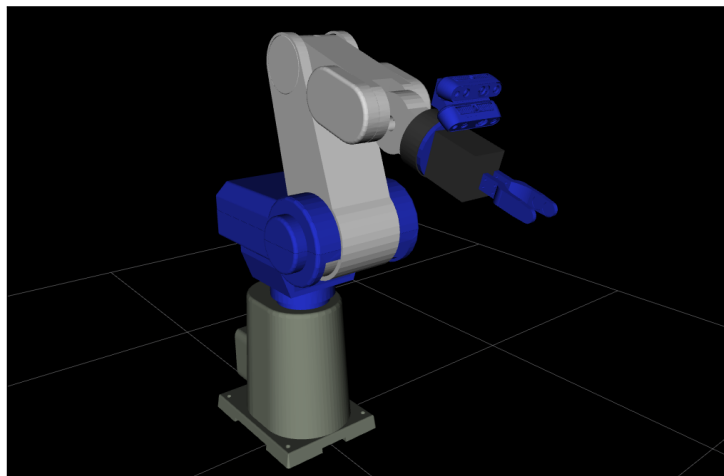
O *firmware* do controlador conta com sistemas de controle PID digital para cada junta. Diferente do realizado no ORCA, utilizou-se as parcelas proporcional, integral e derivativa para obter um controle suavizado do deslocamento das juntas até seu *setpoint*.

Um sistema de *feedback* visual foi implementado para monitorar o comportamento do robô através do Rviz. O objetivo é ter uma representação virtual semelhante ao modelo real do Movemaster que replique seus movimentos. No visualizador é possível adicionar elementos gráficos que fornecem informações de desempenho, como a trajetória executada pelo efetuador, por exemplo.

Não foi possível encontrar um modelo virtual pronto para o Movemaster RV-M2, porém o modelo do robô Movemaster RV-M1, disponibilizado por Walus (2018), possui a mesma estrutura cinemática, mudando apenas em questões de dimensão. Os arquivos do projeto original são fornecidos no formato .CATIA, portanto cada peça foi convertida para o formato .SKP, para edição simplificada e livre no *software Sketchup*³.

Após a adaptação das dimensões do modelo para a versão RV-M2, cada parte móvel foi exportada no formato .STL em um arquivo separado. O objetivo de ter cada parte móvel em um arquivo .STL é que dessa forma é possível associar o objeto 3D com um *link* na árvore de transformações do ROS através de um arquivo URDF (do inglês *Unified Robotic Description Format*, ou Formato Unificado de Descrição Robótica). Arquivos URDF contém a descrição das juntas e *links* de robôs compatível com a árvore de transformação do ROS, além de permitir a associação de modelos 3D que podem ser visualizados no Rviz. A figura 19 mostra o modelo gerado no Rviz através de descrição URDF.

Figura 19 – Modelo virtual do Movemaster.



Fonte: Autoria própria.

³ Disponível em: <https://www.sketchup.com/products/sketchup-for-web>

Criou-se um nó ROS na linguagem C++ responsável pela atualização da árvore de transformações com os valores do ângulo das juntas fornecido pelos *encoders*. O nó *state_publisher* subscreve-se aos tópicos */status_**, pelos quais recebe o valor dos ângulos de junta e os publica em um tópico chamado *joint_state*, composto por mensagens do tipo *sensor_msgs/JointState*. Mensagens desse tipo são interpretadas pelo modelo URDF e visualizadas diretamente no Rviz, possibilitando a atualização visual do modelo.

Este arquivo também é o responsável pela inserção dos *links /odom* e */Base* na árvore de transformação. O *link /odom* representa o deslocamento do ponto de origem do manipulador em relação à origem global. O *link /Base* representa o deslocamento da origem do sistema de referência do robô até o ponto de referência para o *link* da base, na junta 1. Para o caso do presente projeto, uma vez que o robô é fixo e sua origem está localizada no mesmo ponto que a junta 1, os três *links* coincidem. Optou-se por manter esta estrutura de transformadas para permitir trabalhos futuros onde deslocamentos podem estar envolvidos.

Além disso, o nó *state_publisher* também está subscrito no tópico */t265/odom/sample*. Ao ler as informações de odometria da câmera T265, este nó gera a transformada de */odom* até um novo *frame* denominado *virtual_effector*. Através de medições diretamente realizadas no robô real, sabe-se que o efetuador encontra-se a uma distância fixa da câmera T265 de 3,75 cm ao longo de x e 6,25 cm ao longo de $-z$ (com referência ao *frame* da câmera de odometria). Essa transformação permite saber qual é a posição do efetuador de acordo com a câmera T265, gerando uma posição de efetuador virtual. Estes dados podem ser visualizados e comparados com a mesma informação fornecida pelos *encoders* para fins de validação.

Também foi desenvolvido na linguagem C++ um nó ROS responsável pela inicialização das câmeras em momento oportuno, chamado *camera_launcher*. Ao energizar o sistema e conectar o ROS Master, o manipulador executa a sua função de RESET, o que garante o início das operações em sua configuração inicial.

O nó *camera_launcher* resolve o problema de inicialização subscrevendo-se nos tópicos */status_**. Somente após verificar que o campo `IsDone` de todas as juntas contém o valor `true`, o nó envia ao terminal o comando para executar o arquivo *camera.launch*, o qual contém as configurações de inicialização das câmeras.

O arquivo *camera.launch* é uma adaptação dos *launchers* originais contidos no pacote *realsense2*. A inicialização das câmeras é feita com os parâmetros padrão do fabricante, alterando apenas os seguintes campos para a aplicação atual:

- `enable_pointcloud`: definido como `false` por padrão. Alterado para `true` para habilitar a publicação do *PointCloud2*;
- `clip_distance`: não definido por padrão. Foi alterado para 0.70, o que significa que a nuvem de pontos ignora todos os valores de distância maior que 0,7 m. Distâncias maiores que isso estão fora da área de interesse no manipulador e representam ruído na aplicação.

Como a inicialização do sistema ROS depende da execução de uma série de nós, criou-se um arquivo geral que inicializa o sistema completo, chamado *movemaster.launch*. Portanto, dentro deste arquivo são executados os nós:

- *Arduino_(1, 2 e 3)*: Inicia os nós dos três Arduinos dentro do controlador;
- *state_publisher*: Atualiza o valor das juntas na árvore de transformação;
- *robot_state_publisher*: Interpreta o arquivo URDF e coloca o robô na árvore de transformação do ROS. Este nó é acompanhado do parâmetro *robot_description*, contendo o caminho para o arquivo URDF;
- *camera_launcher*: Nó responsável pelo monitoramento da rotina de RESET. Inicializa as câmeras quando o robô está na configuração *home*.

Quando os comandos contendo novos ângulos de junta são recebidos pelo controlador, é necessário realizar a conversão para pulsos dos *encoders*. Apesar da informação estar disponível em manual, devido à condição do manipulador e ao fato da contagem de pulsos ser realizada através de interrupções no Arduino, o valor da constante de conversão diverge do original. A calibração foi feita de maneira análoga ao ORCA, realizando a média de uma sequência de 10 contagens de pulsos ao longo de um ângulo de referência de 90° para cada junta.

As constantes de conversão de cada *encoder*, bem como os ângulos que formam a configuração *home* para o Movemaster estão descritos na tabela 4.

Tabela 4 – Relação de pulsos por grau e configuração *home* do Movemaster.

Junta	Pulsos por grau	<i>home</i>
J_1	185	0°
J_2	228	106°
J_3	186	-123°
J_4	155	0°
J_5	116	0°

Fonte: Autoria própria.

Uma característica mecânica observada é a relação existente entre as juntas 4 e 5. Por características próprias do projeto do Movemaster, ao realizar uma rotação na junta 4, o movimento é repetido na junta 5. Em outras palavras, ao realizar um movimento que adiciona θ graus à junta 4, os mesmos θ graus são adicionados à junta 5, porém não reciprocamente. Essa característica pode ser observada nas figuras 20(a) e 20(b), onde é apresentado um deslocamento aplicado na junta 4 que se reflete de maneira indesejada na junta 5.

Figura 20 – Movimento da junta 4 refletido na junta 5.



(a) Pose antes de executar o movimento.

(b) Rotação de 90° na junta 4, adicionados à junta 5.

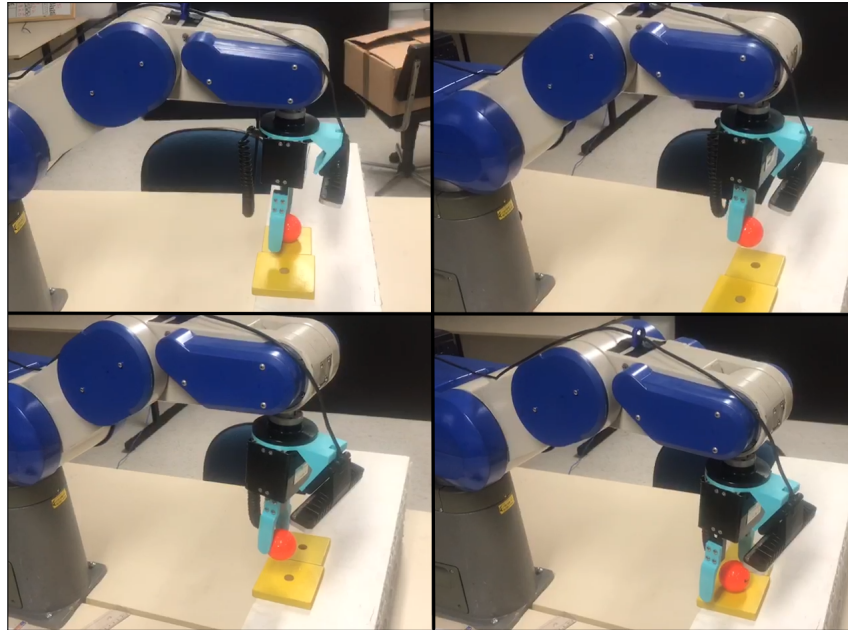
Fonte: Autoria própria.

Devido a esta peculiaridade, para qualquer situação envolvendo o envio de *setpoint* através do ROS, o valor para a junta 5 sempre deve ter o valor da junta 4 subtraído. A exemplo, caso se deseje que a junta 4 assuma 45° e a junta 5 assuma 60°, deve-se enviar os 45° para o tópico `/set_4`, porém deve-se mandar 15° para o tópico `/set_5`. Este não é o caso de cálculos envolvendo cinemática, uma vez que tal situação existe apenas no modelo físico.

Para validar o funcionamento do controlador e do sistema de comunicação ROS do Movemaster, realizou-se um experimento de *pickup*, onde o manipulador pegou uma bola de golfe e a posicionou em outro suporte. As posições de origem e destino são conhecidas e não houve nenhum cálculo cinemático envolvido. Tal demonstração teve como intuito provar o correto funcionamento e calibração do controlador. Imagens da execução do experimento são representadas na figura 21.

Por fim, na figura 22 descreve-se a árvore de transformações dos *frames* de referência que modelam o robô no ROS. Destacam-se os *frames* criados pelos pacotes da Realsense para o funcionamento das câmeras bem como cada junta do manipulador e o suporte da câmera impresso em ABS. Esta árvore de transformações representa uma cadeia de relações do tipo pai

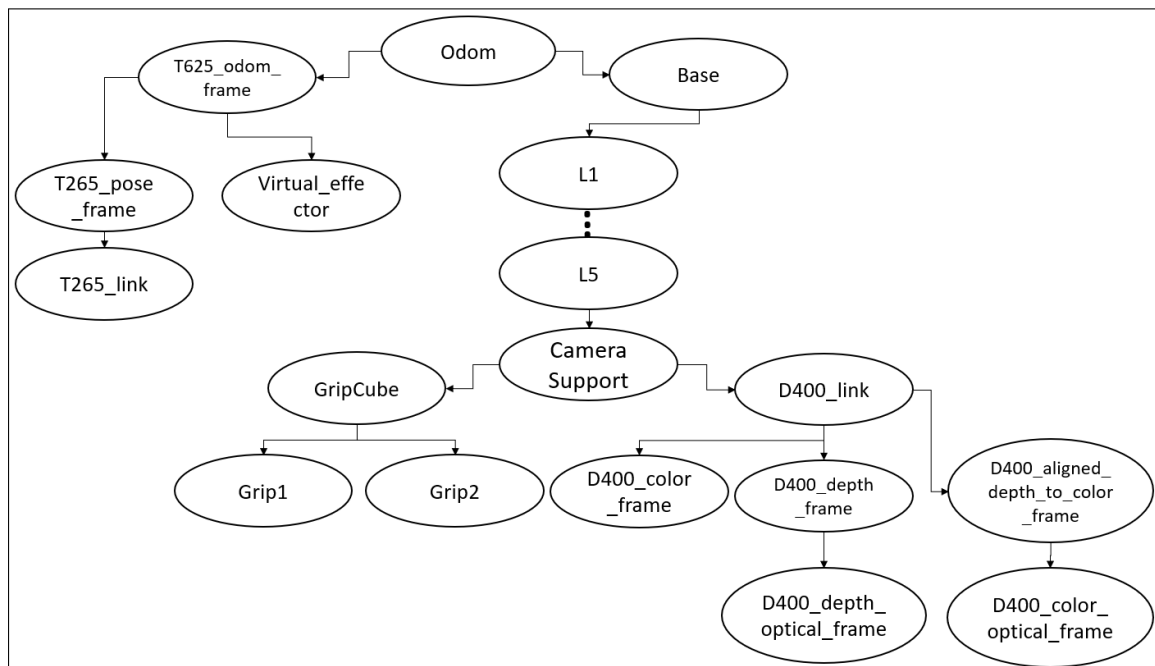
Figura 21 – Validação do controlador através de tarefa de *pickup* de objeto.



Fonte: Autoria própria.

e filho, onde cada nó — ou seja, cada elemento — está relacionado a um dos *frames* da cadeia cinemática. Cada nó tem um pai e quantos filhos forem necessários para descrever o sistema por completo. Esta transformação permite relacionar *frames* fundamentais, como a posição da base do robô em relação a alguma referência global, até detalhes de dependência como os *frames* dos sensores e lentes da câmera.

Figura 22 – Árvore de transformações ROS do Movemaster.



Fonte: Autoria própria.

3.3 CONSIDERAÇÕES

Este capítulo teve como objetivo apresentar os manipuladores utilizados em ambiente de experimentação prática. Ambos foram modernizados de forma a permitir seu controle através de uma rede ROS com mensagens customizadas e com controladores *open-source*.

A natureza didática do ORCA, contendo três graus de liberdade dispostos de forma planar, permitiu a elaboração gradual das técnicas estudadas ao longo do projeto. Após seu uso nas etapas intermediárias, o manipulador foi direcionado a um projeto paralelo, permitindo foco nas aplicações com o Movemaster.

O manipulador Movemaster possui características mais complexas do ponto de vista cinemático uma vez que pode deslocar-se ao longo do ambiente tridimensional. A modernização de sua estrutura conta com a instalação de sistemas de percepção próprio e exteroceptiva como câmeras de odometria, profundidade e imagem.

4 CINEMÁTICA CLÁSSICA

A cinemática, na área da robótica, é o estudo dos mecanismos de um robô sem considerar parcelas como forças, torques e imperfeições mecânicas que causam o movimento (CRAIG, 2012). Uma maneira de se descrever matematicamente a pose de um objeto é realizando a sua transposição sobre um sistema de coordenadas cartesianas de referência. Assim, define-se que: um corpo rígido é um ponto associado a um sistema cartesiano de referência - um *frame* - o qual descreve sua configuração espacial. Um corpo rígido livre no espaço tridimensional pode transladar e rotacionar em qualquer dos eixos x , y e z de seu *frame* de referência, o que implica na necessidade de definir seis variáveis para descrevê-lo por completo (LYNCH; PARK, 2017).

Neste capítulo, é feita a aplicação prática dos conceitos da notação clássica para manipuladores, chamada Denavit-Hartenberg, utilizando as Matrizes de transformação Homogênea em exemplos práticos. Os mesmos exemplos são computados utilizando a notação alternativa dos Helicoides Sucessivos, onde discorre-se sobre as diferenças entre as abordagens. Tal contextualização faz-se necessária dado o objetivo de estabelecer seu uso e limitações para que as ferramentas avançadas sejam propriamente descritas no capítulo 5.

Também são expostos os conceitos de cinemática diferencial com a aplicação da matriz Jacobiana e a derivação de uma equação para objetivos secundários. O capítulo encerra com a demonstração de exemplos práticos das equivalências e diferenças entre as duas convenções utilizando dois manipuladores reais de estruturas diferentes. Ao analisar os resultados, propõe-se uma variação do algoritmo de maximização de distância angular para limites de junta.

Os conceitos que fundamentam as aplicações da cinemática direta deste capítulo foram organizados no formato de apêndices no final deste documento. Nas seções seguintes são detalhados apenas conceitos necessários para a contextualização do tema da dissertação, de modo a simplificar a dissertação para os leitores familiarizado com tais conceitos.

No apêndice A são revisados os conceitos de representação geométrica fundamental e descreve-se a formulação das Matrizes de Transformação Homogênea (MTH). No apêndice B revisa-se a representação mínima de orientação através de ângulos de Euler, expondo a limitação que é abordada através dos quatérnios unitários. Os apêndices C e D contém a fundamentação das notações utilizadas neste capítulo para a descrição de manipuladores: parâmetros DH e Helicóides Sucessivos, respectivamente. Para a definição matemática e propriedades da matriz Jacobiana, o leitor deve consultar o apêndice E.

4.1 CINEMÁTICA DIRETA

Uma aplicação dos conceitos de transformação de *frames* de referência é no cálculo da posição e orientação atual da ferramenta de um manipulador. O cálculo da posição e orientação cartesiana de um efetuador, tendo conhecimento de todas suas variáveis de junta, chama-se cinemática direta (CRAIG, 2012). Para aplicar os conceitos de transformação a um manipulador, é necessário descrever sua estrutura matematicamente utilizando parâmetros que se comportam de maneira análoga ao modelo físico.

Existem duas formas comumente aplicadas na descrição de uma cadeia serial de corpos rígidos e estas levam a metodologias diferentes para a solução da cinemática direta. A primeira metodologia e a mais amplamente difundida é a convenção de Denavit-Hartenberg, e uma alternativa que recentemente tem ganhando destaque em trabalhos científicos é a metodologia dos Helicóides Sucessivos.

4.2 APLICAÇÃO DOS PARÂMETROS DENAVIT-HARTENBERG

Nesta seção é realizada a demonstração de aplicação dos Parâmetros DH, explicados em detalhes no apêndice C. O objetivo é obter exemplos que podem ser comparados à abordagem alternativa dos Helicóides Sucessivos utilizando como exemplo os manipuladores modernizados em laboratório.

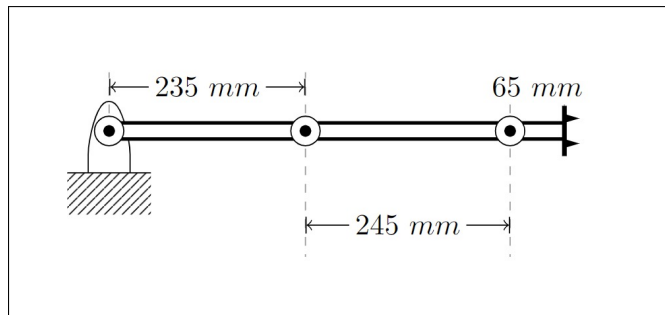
Para exemplificar a aplicação dos parâmetros DH, apresenta-se as tabelas referentes a duas estruturas de manipulador, os quais são aplicados neste projeto em ambiente prático e que passaram pelo processo de *retrofit* conforme descrito no Capítulo 3. A tabela 5 apresenta os parâmetros do manipulador modelo Beckman Coulter ORCA cujo esquemático dos parâmetros está representado na figura 23. A tabela 6 apresenta os Parâmetros DH do manipulador modelo Mitsubishi Movemaster RV-M2 cujo esquemático dos parâmetros está representado na figura 24. As medidas de ângulo são dadas em radianos e as de distância em milímetros. Nota-se que os valores de θ não são fixos, o que significa que esta é a variável de junta para todas as juntas.

Tabela 5 – Parâmetros DH do ORCA modernizado.

Junta	α	a	θ	d
J_1	0	235	θ_1	0
J_2	0	245	θ_2	0
J_3	0	65	θ_3	0

Fonte: Autoria própria.

Figura 23 – Esquema cinemático do ORCA modernizado.



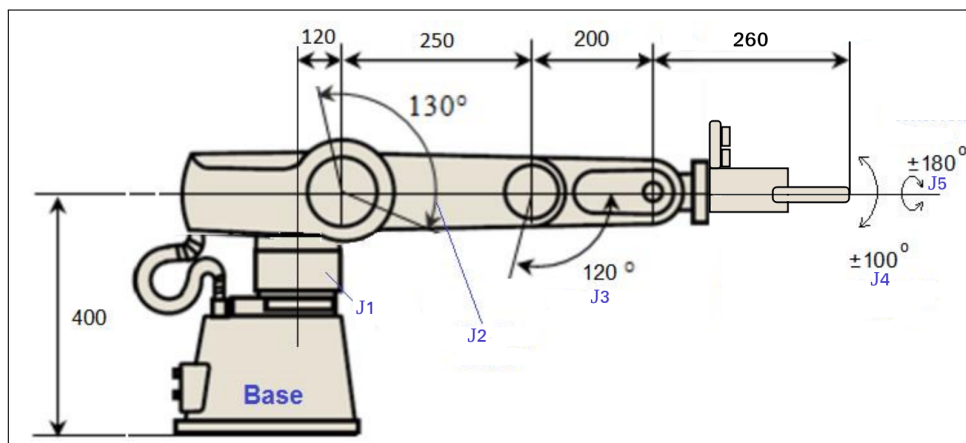
Fonte: Autoria própria.

Tabela 6 – Parâmetros DH do Movemaster modernizado.

Junta	α	a	θ	d
J_1	$\pi/2$	120	θ_1	0
J_2	0	250	θ_2	0
J_3	0	200	θ_3	0
J_4	$\pi/2$	0	$\theta_4 + \pi/2$	0
J_5	0	0	θ_5	250

Fonte: Autoria própria.

Figura 24 – Esquema cinemático do Movemaster modernizado.



Fonte: Adaptado de Duong e Jaksic (2018)

Exemplificando a aplicação, tem-se o manipulador ORCA modernizado na configuração de ângulos de junta $q = [0,3491, 1,5708, -1.9199]$ rad. Deseja-se aplicar as MTH na convenção dos parâmetros DH para descobrir a posição e orientação do efetuador. A Matriz de Transformação Homogênea, a posição cartesiana e orientação do efetuador encontrados foram:

$$H = \begin{bmatrix} 1 & 0 & 0 & 318,140 \\ 0 & 1 & 0 & 309,321 \\ 0 & 0 & 1 & 0,000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$p' = \begin{bmatrix} 318,140 \\ 309,321 \end{bmatrix} mm \quad (2)$$

$$R_{RPY} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} rad \quad (3)$$

Tais resultados estão de acordo com o observado no manipulador físico, onde utilizou-se a rede ROS para enviar os ângulos de junta desejados. O manipulador posicionou-se conforme figura 25 e sua posição foi medida manualmente 10 vezes por duas pessoas diferentes, onde a média dos valores de posição obtido foi $p' = [335, 330]$ mm. Esta medida representa um erro normalizado de 5,98% no modelo real, o qual pode ser atribuído às imperfeições mecânicas, de controlador e de modelagem do manipulador.

Figura 25 – Posicionamento do ORCA no exemplo de cinemática direta.



Fonte: Autoria própria.

De mesma forma, os parâmetros DH foram aplicados através das MTH no Movemaster. O objetivo é encontrar a pose do efetuador com uma configuração de variáveis de junta $q = [\pi/6, \pi/3, -\pi/2, -\pi/6, \pi/2]^T$ rad. Os resultados encontrados para H , p' e R_{RPY} foram:

$$H = \begin{bmatrix} 0,500 & -0,750 & 0,433 & 470,429 \\ -0,866 & -0,433 & 0,250 & 271,602 \\ 0,000 & -0,500 & -0,866 & 300,000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$p' = \begin{bmatrix} 470,429 \\ 271,602 \\ 300,000 \end{bmatrix} \text{ mm} \quad (5)$$

$$R_{RPY} = \begin{bmatrix} -1,047 \\ 0,000 \\ -2,618 \end{bmatrix} \text{ rad} \quad (6)$$

Através da rede ROS, os ângulos q foram enviados ao manipulador físico que posicionou-se como demonstrado na figura 26, onde através da média de medições concluiu-se que a posição do efetuador é $p' = [477, 275, 293]^T$ mm, representando um erro normalizado de 1,02%, decorrente dos erros acumulados por imprecisões mecânicas, de controlador e de modelagem.

Figura 26 – Posicionamento do Movemaster no exemplo de cinemática direta.



Fonte: Autoria própria.

4.3 APLICAÇÃO DOS HELICÓIDES SUCESSIVOS

Os parâmetros de Helicóides Sucessivos dos manipuladores utilizados neste projeto podem ser extraídos ao observar os esquemas cinemáticos das figuras 23 e 24. Levando em consideração a pose de referência onde $\theta = 0$ para todas as juntas, obtêm-se os parâmetros do manipulador ORCA descritos na tabela 7, onde o ponto de referência do efetuador está em

$p = [655, 0, 0]^T$ mm. O mesmo se aplica ao manipulador Movemaster, descrito na tabela 8, onde o ponto de referência do efetuador está em $p = [820, 0, 400]^T$ mm.

Tabela 7 – Helicóides Sucessivos do ORCA modernizado.

Junta	s_i	s_{0_i}
J_1	(0, 0, 1)	(0, 0, 0)
J_2	(0, 0, 1)	(245, 0, 0)
J_3	(0, 0, 1)	(485, 0, 0)

Fonte: Autoria própria.

Tabela 8 – Helicóides Sucessivos do Movemaster modernizado.

Junta	s_i	s_{0_i}
J_1	(0, 0, 1)	(0, 0, 400)
J_2	(0, -1, 0)	(120, 0, 400)
J_3	(0, -1, 0)	(370, 0, 400)
J_4	(0, -1, 0)	(570, 0, 400)
J_5	(1, 0, 0)	(570, 0, 400)

Fonte: Autoria própria.

As etapas da cinemática utilizando Helicóides Sucessivos permanecem iguais à convenção DH, uma vez feitas as alterações na estrutura da matriz H . Desta forma, H pode ser aplicado à equação 50 — ver apêndice D — para obter a pose do efetuador e tendo, assim, a solução da cinemática direta.

Para exemplificar esta modelagem, repete-se o experimento relatado na sub-seção 4.2. A matriz H obtida através dos Helicóides Sucessivos no manipulador ORCA é dada pela equação 7. Os demais resultados obtidos são idênticos aos obtidos com os parâmetros DH e estão omitidos por concisão.

$$H = \begin{bmatrix} 1 & 0 & 0 & -336,860 \\ 0 & 1 & 0 & 309,321 \\ 0 & 0 & 1 & 0,000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Da mesma forma, o experimento foi repetido para o Movemaster. A Matriz de Transformação Homogênea H obtida é apresentada na equação 8.

$$H = \begin{bmatrix} 0,433 & 0,750 & 0,500 & -84,641 \\ 0,250 & 0,433 & -0,866 & 413,012 \\ -0,866 & 0,500 & 0,000 & 1010,140 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Observa-se equivalência nos resultados das duas abordagens de parametrização. Neste caso cabe a escolha da metodologia a ser empregada. A comparação dos métodos é feita no trabalho de Li et al. (2021), onde há uma diferenciação de notação nos helicoides utilizados. A conclusão é que os parâmetros DH possuem um processo de cálculo mais simples porém uma modelagem mais complicada e produz singularidades com mais facilidade. Os Helicoides Sucessivos por outro lado permitem a computação da cinemática utilizando apenas dois *frames* de referência, sendo a base e o efetuador. Além disso, a representação helicoidal torna-se geometricamente clara ao observar os parâmetros. As conclusões obtidas pelos autores mostra que a escolha de um ou outro método pela complexidade de implementação é uma métrica subjetiva. Conclusão semelhante é encontrada na contribuição de Cardona et al. (2020), onde os autores afirmam a simplicidade de representar um manipulador pelos Helicoides Sucessivos. Uma conclusão qualitativa é fornecida por Ribeiro et al. (2008), onde é demonstrado que a escolha de uma pose de referência utilizando Helicoides Sucessivos pode permitir a construção de uma matriz Jacobiana simplificada.

Para este projeto - e para futuras implementações - utiliza-se a notação por Helicoides Sucessivos. Apesar da métrica de facilidade de implementação ser relativa, observa-se em literatura que esta convenção de parametrização está diretamente relacionada à álgebra dos quatérnios duais, como observado nas contribuições feitas por Chevallier (1991), Gouasmi (2012) e Özgür e Mezouar (2016). Os quatérnios duais são parte fundamental deste trabalho e são explorados ao longo do capítulo 5 e a notação unificada de toda a implementação é um dos subprodutos das contribuições aqui almejadas.

4.4 CINEMÁTICA INVERSA DIFERENCIAL

As definições que fundamentam a cinemática direta foram apresentadas no decorrer deste capítulo. Tais equações permitem determinar a pose de um manipulador em função das variáveis de junta para cada uma das metodologias usuais. O problema complementar a este é chamado de *cinemática inversa*, onde o objetivo é conhecer o conjunto de valores para as variáveis de junta que satisfaz determinada pose do *frame* do efetuador. Para facilitar a compreensão da cinemática inversa utilizada neste projeto, a qual exige o entendimento de diferenciação de equações, faz-se necessário elaborar o significado de terminologias comumente empregadas e explanar conceitos pouco claros presentes na literatura.

Para calcular o conjunto de variáveis de junta que, se aplicadas a um manipulador,

levam o efetuador e uma determinada pose, primeiramente é necessário definir qual é a relação matemática entre o movimento das juntas e o movimento do efetuador. Estas relações são representadas através de equações cinemáticas diferenciais que relacionam a influência da variação de cada junta à variação da pose do efetuador através da denominada matriz Jacobiana (SCIAVICCO; SICILIANO, 2012).

Neste ponto cabe salientar a definição de que a *variação* das juntas causa uma *variação* no efetuador, e isso está diretamente ligado ao conceito de diferenciação das equações cinemáticas já conhecidas. O ponto que pode não ficar claro na literatura é o conceito de *velocidade* das juntas. Como lida-se com a variação de grandezas que representam ângulos e distâncias, é corriqueiro relacionar suas derivadas com velocidades angulares e lineares respectivamente. Esta afirmação depende do caso: pode-se utilizar a diferenciação para projetar movimentos ao longo de uma trajetória com um tempo definido por um perfil de velocidades, porém aplica-se o mesmo conceito para cálculos de *variação instantânea*, sem haver um tempo ou velocidade necessariamente atrelados.

Um exemplo do uso de velocidades instantâneas no sentido apenas matemático e sem atrelar ao tempo é dado no artigo de Buss e Kim (2005), onde os autores comparam diversos métodos de solução para a cinemática inversa porém utilizando métodos iterativos, onde cada iteração busca reduzir o erro em relação à pose desejada sem exigir um cumprimento de trajetória dentro de limites físicos. Neste tipo de aplicação, apenas as variáveis de junta finais interessam para o manipulador e a trajetória física é apenas consequência das características mecânicas e do controlador.

Nesta dissertação, independente de se estar aplicando a diferenciação para cálculos de velocidade de trajetória ou para convergência do efetuador, o termo empregado para a derivada de ângulos ou posições será sempre *velocidade angular ou linear*, respectivamente. A desambiguação deve ficar evidente a depender do contexto. As notações e definições contidas nesta seção são dadas de acordo com o livro de Sciavicco e Siciliano (2012), exceto se explicitado o contrário.

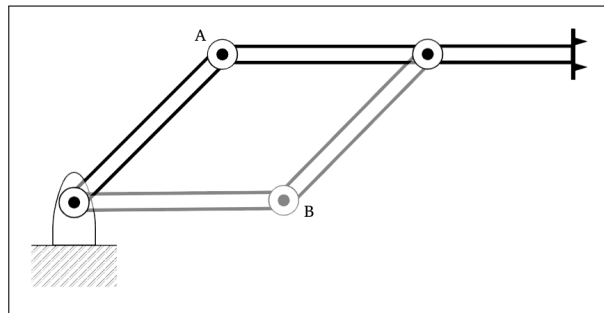
4.4.1 Redundância e Objetivos Secundários

Um manipulador é redundante quando o número de graus de liberdade é maior que o número de variáveis necessárias para descrever o espaço operacional. Um exemplo de manipulador redundante é o ORCA, apresentado na figura 23, quando o objetivo é apenas controlar a

posição do efetuador sem levar em consideração sua orientação.

Uma vez que o ORCA é um manipulador planar, define-se sua posição apenas em termos das coordenadas x e y . Um manipulador com três graus de liberdade e duas variáveis sendo controladas é considerado redundante, e neste caso é possível que existam duas ou mais soluções para a mesma posição, conforme demonstrado na figura 27. A configuração A é atingida com os ângulos $q = [\pi/4 \ - \ \pi/4, \ 0]^T$ rad e a configuração B tem os ângulos $q = [0,0087, \ 0,7852, \ -0,7942]^T$ rad. Ambas resultam num efetuador posicionado em $p = [583,2, \ 173,2]^T$ mm, como pode ser verificado através das técnicas de cinemática direta já revisadas neste capítulo e nos apêndices A a D.

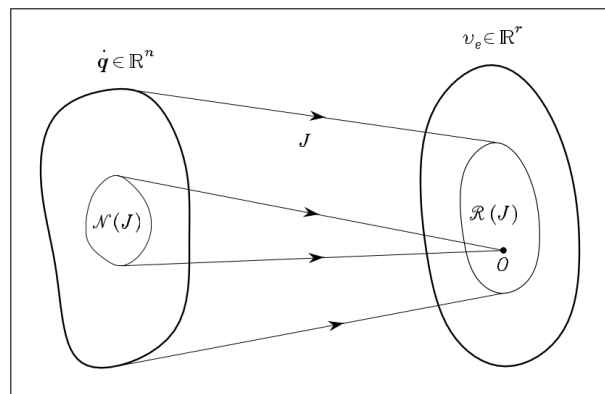
Figura 27 – Exemplo de redundância do manipulador ORCA.



Fonte: Autoria própria.

Do ponto de vista da cinemática diferencial, uma maneira de visualizar a redundância de um manipulador é através da figura 28. O conjunto do lado esquerdo da figura representa o domínio da velocidade das juntas \dot{q} , o qual tem dimensão de n graus de liberdade do manipulador. O conjunto do lado direito descreve o domínio da velocidade do efetuador v_e , cuja dimensão é de r variáveis sendo controladas. J é a matriz Jacobiana que define o mapeamento entre os dois domínios.

Figura 28 – Mapeamento redundante entre a velocidade das juntas e do efetuador.



Fonte: Sciavicco e Siciliano (2012)

A redundância descrita através da figura 28 é observada com a indicação de que o subespaço $N(J)$ é mapeado para um ponto de valor zero no espaço das velocidades do efetuador. Este espaço é denominado *espaço nulo* pois todas as variações aplicadas às juntas dentro deste espaço não geram movimento no efetuador. O subespaço $R(J)$ é chamado *espaço de alcance* e representa todas as velocidades instantâneas possíveis para a configuração atual do manipulador.

A existência de um subespaço $N(J) \neq \emptyset$ permite o desenvolvimento de técnicas que utilizam os graus de liberdade redundantes para executar tarefas secundárias, conforme descrito por Sciavicco e Siciliano (2012). Os autores demonstram que é possível reescrever as equações de cinemática inversa para obter uma extensão que engloba as soluções redundantes, obtendo a equação 9. O uso da formulação pseudoinvertida para a inversão da Jacobiana introduz inconsistências matemáticas no decorrer do cálculo, uma vez que esta abordagem resulta em uma aproximação dos valores que de fato representam a inversão de matrizes de dimensão genérica. Apesar destas limitações estarem presentes cumulativamente em cada iteração do cálculo, o fato de isto ocorrer dentro de um sistema de malha fechada faz com que o erro do sistema tenda a zero caso isso seja possível.

$$\dot{q} = J^\dagger v_e + (I_n - J^\dagger J) \dot{q}_0 \quad (9)$$

Há dois termos na equação 9. O primeiro representa a minimização local da velocidade das juntas e o segundo é chamado de *solução homogênea*. O termo $(I_n - J^\dagger J)$ resulta no espaço nulo de J e indica redundância quando seus elementos são diferentes de zero. Nesta formulação, caso exista redundância, o segundo termo é não-nulo e o valor de \dot{q}_0 é adicionado ao espaço das velocidades do efetuador, devendo este ser projetado para maximizar um objetivo secundário.

O objetivo secundário é definido conforme equação 10, onde $w(q)$ é uma função representando qualquer informação dada em função das variáveis de junta. O valor de k_0 é definido empiricamente para ponderar a influência do ganho secundário na equação. Os autores também salientam que o objetivo secundário é executado *em respeito* ao objetivo primário, ou seja, este será violado caso sua realização force o aumento do erro no objetivo primário.

$$\dot{q}_0 = k_0 \left(\frac{\partial w(q)}{\partial q} \right)^T \quad (10)$$

O projeto da função $w(q)$ é totalmente dependente da aplicação. Objetivos secundários de relevância prática são listados:

- *Manipulabilidade do efetuador*, definida conforme equação 11. Esta métrica torna-se nula em configurações singulares e maximizar seu valor significa afastar-se de singularidades.

$$w(q) = \sqrt{\det(J(q)J^T(q))} \quad (11)$$

- *Distância dos limites mecânicos das juntas*, definida conforme equação 12, onde q_{iM} , q_{im} e \bar{q}_i indicam os valores máximo, mínimo e médio para a variável de junta q_i , respectivamente.

$$w(q) = -\frac{1}{2n} \sum_{i=1}^n \left(\frac{q_i - \bar{q}_i}{q_{iM} - q_{im}} \right)^2 \quad (12)$$

- *Distância de um obstáculo*, definida conforme equação 13, onde o objetivo é maximizar a distância entre um ponto $p(q)$ qualquer na estrutura do manipulador até um obstáculo centralizado no ponto o .

$$w(q) = \min_{p,o} \|p(q) - o\| \quad (13)$$

Discorreu-se nesta seção sobre os conceitos necessários para a definição completa da cinemática inversa diferencial. As informações revisadas serão organizada na seção seguinte no formato de sistema de controle de modo a gerar a solução prática da cinemática inversa.

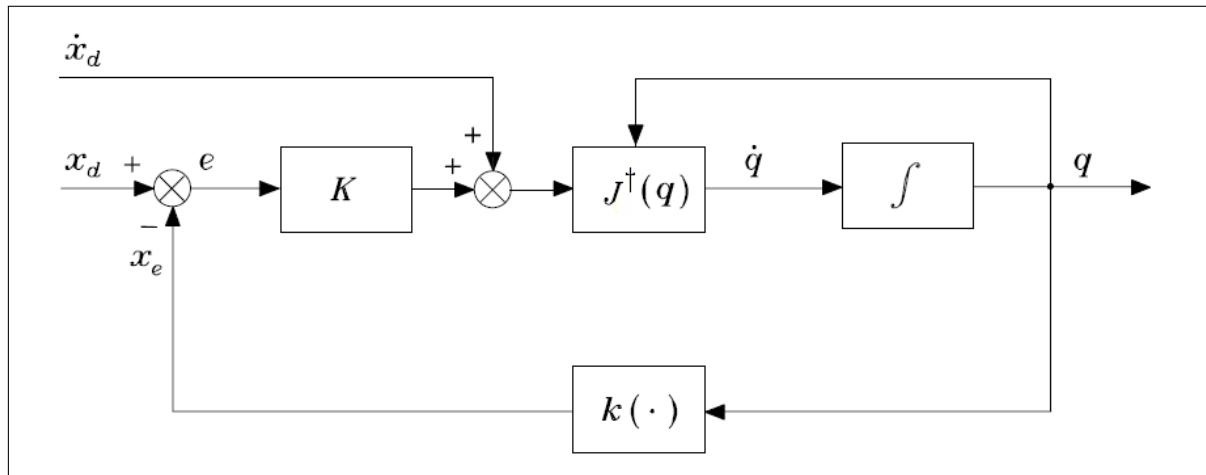
4.4.2 Cinemática Inversa Diferencial em Malha Fechada

Como as equações da cinemática inversa diferencial são definidas através de uma operação de integração de funções não-lineares no espaço discreto, introduzindo o fenômeno de *drift*, naturalmente há um erro associado à solução de malha aberta apresentada na equação 66 — ver apêndice E.— Em situações práticas é necessário realizar um controle de malha fechada para que o erro tenda a zero ao longo da trajetória ou ao longo das iterações de convergência de cálculo.

O diagrama de blocos na figura 29 descreve o sistema de controle de erro para solução numérica da cinemática inversa. O sistema possui duas entradas, sendo x_d a pose desejada para o efetuador e \dot{x}_d a sua velocidade instantânea. O bloco K indica um vetor de ganhos para o vetor de erro de pose e . As entradas passam pela inversão da Jacobiana, onde neste caso está sendo utilizada a pseudoinversa da Jacobiana Geométrica. Este ponto do sistema gera a velocidade das

juntas \dot{q} , a qual passa por um integrador para fornecer a saída desejada com os novos valores para as variáveis de junta q . Como estes valores possuem um erro associado, a cinemática direta é computada no bloco $k(\cdot)$. A pose resultante do efetuador x_e é realimentada negativamente à entrada para fechar a malha de controle.

Figura 29 – Cinemática inversa diferencial em malha fechada.



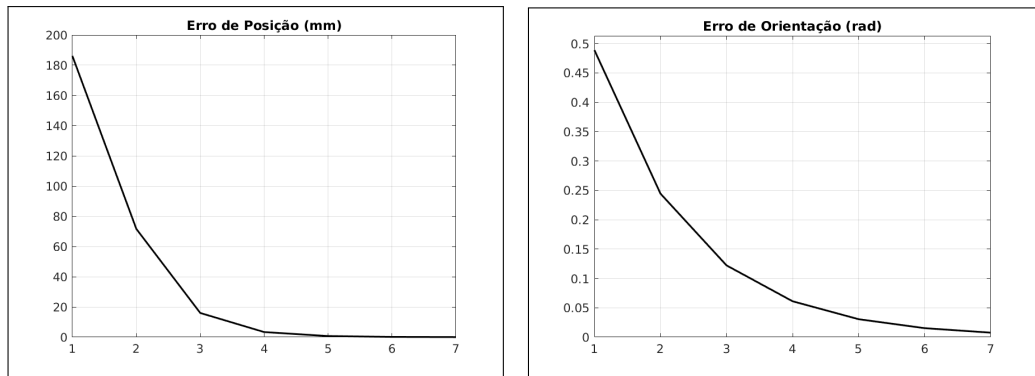
Fonte: Adaptado de Sciavicco e Siciliano (2012)

Com a finalidade de demonstrar a unificação de todos os conceitos descritos ao longo deste capítulo em cenário prático, realizou-se experimentos de geração de trajetória para os manipuladores ORCA e Movemaster. Os parâmetros cinemáticos de ambos foram descritos neste capítulo ao longo da seção 4.1. Todos os algoritmos, bem como a geração dos resultados gráficos, foram implementados no *software* Matlab.

4.4.3 Demonstrações com o Beckman Coulter ORCA

O experimento 1 realizado com o ORCA modernizado visa aplicar o sistema de malha fechada para solucionar a cinemática inversa de maneira iterativa. O manipulador encontra-se inicialmente em sua configuração *home*, onde as juntas posicionam-se ao energizar o sistema, tal que $q_i = [-0,6981 \quad 2,5307 \quad -2,3213]^T$ rad. O efetuador deve convergir para a pose $eff = [260, 180, 0]^T$ mm com sua orientação em torno do eixo z em $\phi = 0$ rad. Não houve necessidade de utilizar o amortecimento da Jacobiana, logo $\lambda = 0$. A condição de parada do algoritmo é quando o erro de posição $\|e_{pos}\| \leq 1$ mm e o erro de orientação $\|e_{ori}\| \leq 0,01$ rad. A figura 30 apresenta a norma dos erros de posição (a) e orientação (b) atingidas após o algoritmo realizar 7 iterações, o que o levou a atingir a condição de parada, sendo este o resultado desta demonstração.

Figura 30 – Demonstração de convergência da cinemática diferencial (ORCA).



(a) Convergência de posição.

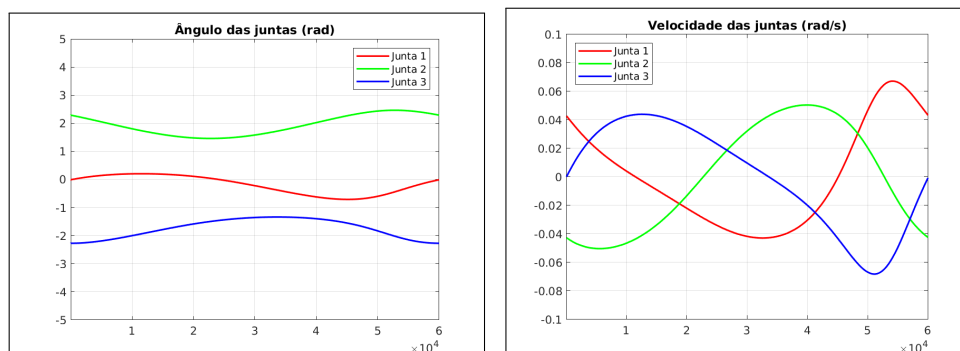
(b) Convergência de orientação.

Fonte: Autoria própria.

O objetivo do experimento 2 realizado com o ORCA modernizado é executar uma trajetória circular centrada no ponto $c = [360, 180]^T$ mm com raio de 100 mm. O efetuador deve realizar uma volta completa em 60 s sem perder a orientação inicial. O tempo de integração é de 1 ms. O vetor de ganhos do erro de pose é definido como $K = [800, 800, 500]^T$. O efetuador está posicionado no ponto $p_i = [260, 180]^T$ mm orientado a $\phi = 0$ rad, o que torna o erro inicial $e_0 = 0$ mm.

Como resultado desta demonstração, as figuras 31(a) e 31(b) descrevem o ângulo e a velocidade das juntas durante a execução da trajetória, respectivamente. Na figura 32 apresenta-se o erro de posição, onde o erro de orientação encontra-se suprimido uma vez que este tem magnitude próxima de zero, abaixo da escala de 10^{-10} rad, o que compromete a visualização. Nota-se o caráter do erro, que mesmo com características oscilatórias, mantém-se em valores dentro da tolerância.

Figura 31 – Execução da trajetória circular (ORCA).

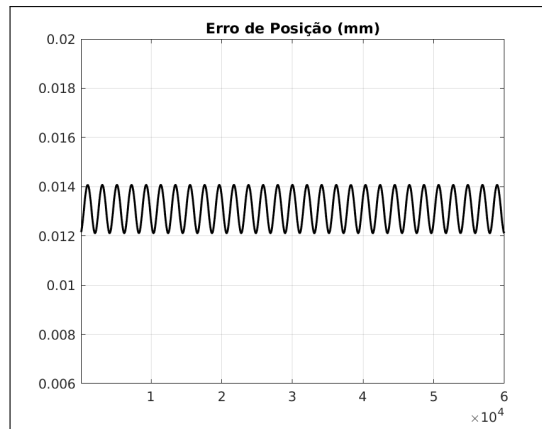


(a) Ângulo das juntas.

(b) Velocidade das juntas.

Fonte: Autoria própria.

Figura 32 – Erro de posição na trajetória circular (ORCA).



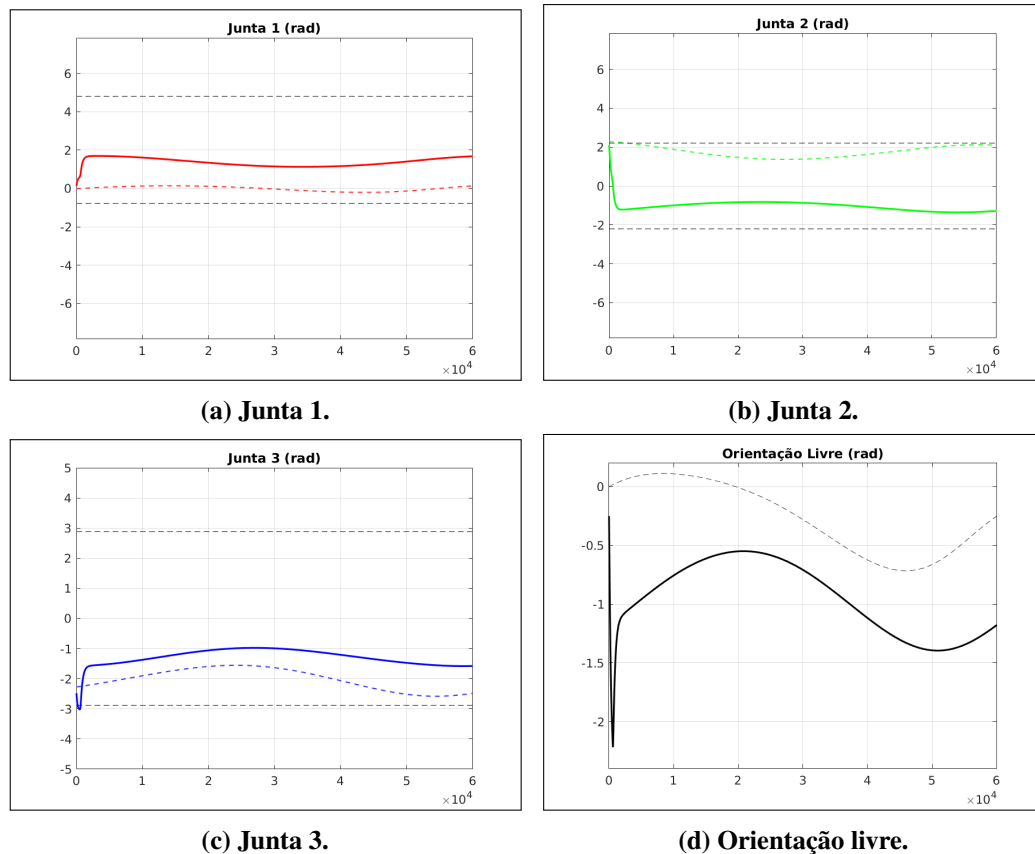
Fonte: Autoria própria.

O experimento 3 com o ORCA objetiva aplicar o objetivo secundário de maximizar a distância dos limites mecânicos das juntas. Para isso, repete-se o experimento 2 porém sem manter o controle da orientação, tornando o manipulador redundante. O ganho secundário utilizado é $k_0 = 250$. Conforme estabelecido na equação 10, é necessário computar a derivada parcial da equação de objetivo secundário para que esta seja aplicada na cinemática diferencial. Ao realizar a derivada da equação 12 com relação à variável de junta q_i , obtém-se a equação 14.

$$\dot{w}(q) = \frac{q_{im} - 2q_i + q_{iM}}{6(q_{im} - q_{iM})^2} \quad (14)$$

Os resultados deste experimento podem ser observados nas figura 33(a-c), onde observa-se o comportamento das juntas 1, 2 e 3 (linhas contínuas coloridas) em relação aos seus limites físicos (linhas tracejadas de mesma cor). Na figura 33(d) demonstra-se que para este caso a orientação segue livre de restrições, sendo apenas uma consequência dos cálculos de posição angular das juntas. As linhas tracejadas em preto na figura 33 representam o caso em que não há o reforço de um objetivo secundário, ou seja, $k_0 = 0$, permitindo assim observar o efeito do algoritmo na linha contínua ao alterar o valor deste ganho de acordo com o enunciado do experimento. Estes resultados demonstram o correto funcionamento dos algoritmos em uma estrutura simples e didática como o manipulador planar 3R. Um conjunto análogo de experimentos foi realizado no manipulador Movemaster.

Figura 33 – Trajetória com objetivo secundário (ORCA).



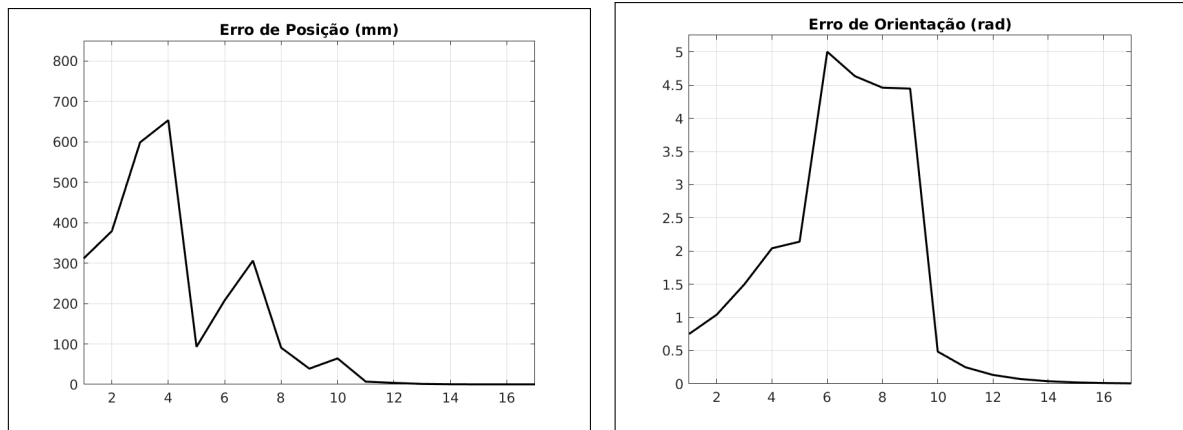
Fonte: Autoria própria.

4.4.4 Demonstrações com o Mitsubishi Movemaster RV-M2

Todos os experimentos realizados com o manipulador ORCA foram repetidos com o Movemaster, dados os ajustes descritos nesta subseção para tornar o cenário análogo. Para o experimento 1, a convergência deve ser realizada para o ponto $p = [440, 0, 200]^T$ mm com orientação $\phi = -\pi/3$ rad no eixo y e $\gamma = 0$ rad no eixo x . A orientação é dada em relação ao *frame* do efetuador, sabendo que a orientação no eixo z não pode ser controlada pela arquitetura deste manipulador. O robô encontra-se em sua configuração *home*, onde as juntas posicionam-se ao energizar o sistema, ou seja, $q_i = [0, 1,8500, -2,1468, 0, 0]^T$ rad. A constante de amortecimento da Jacobiana é $\lambda = 0,1$ para todas as juntas. A condição de parada do algoritmo é quando o erro de posição $\|e_{pos}\| \leq 1$ mm e o erro de orientação $\|e_{ori}\| \leq 0,01$ rad.

A figura 34 apresenta a norma dos erros de posição (a) e orientação (b) ao longo das 17 iterações necessárias para atingir a condição de parada, sendo este o resultado desta demonstração. Tal resultado valida o algoritmo base que será aprimorado na inicialização das juntas nos experimentos descritos no Capítulo 6.

Figura 34 – Demonstração de convergência da cinemática diferencial (Movemaster).



(a) Convergência de posição.

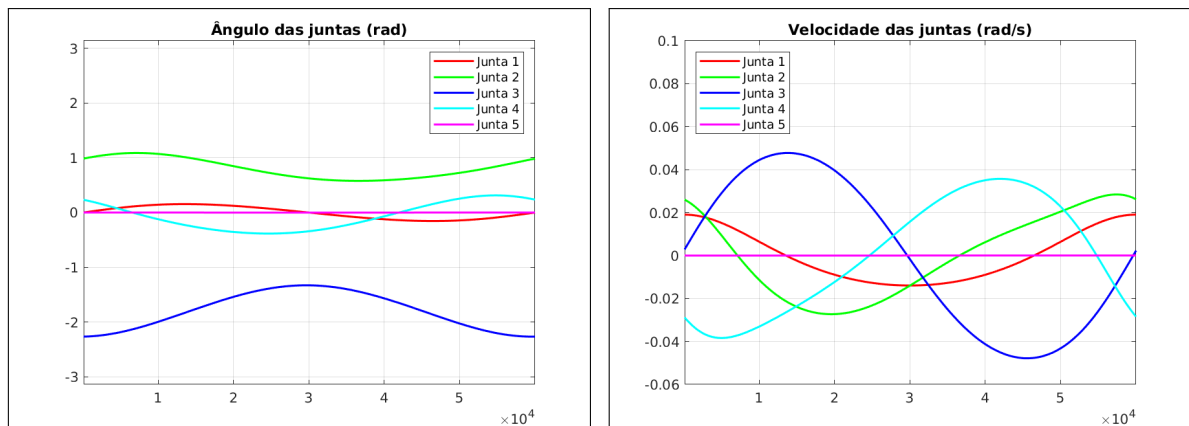
(b) Convergência de orientação.

Fonte: Autoria própria.

O experimento 2 executado no Movemaster visa executar uma trajetória em forma de elipse centrada no ponto $c = [520, 0, 200]^T$ mm com raios de 80 mm ao longo de x , 80 mm ao longo de y e 50 mm ao longo de z . O efetuador deve realizar uma volta completa em 60 s sem perder a orientação inicial. O tempo de integração é de 1 ms. O vetor de ganhos para o erro de pose é $K = [1800, 1800, 1800, 1200, 1200]^T$. O efetuador está posicionado no ponto $p_i = [440, 0, 200]^T$ mm orientado a $\phi = -\pi/3$ rad no eixo y e a $\gamma = 0$ rad no eixo x , o que torna o erro inicial $e_0 = 0$.

Como resultado desta demonstração, na figura 35, descreve-se o ângulo e a velocidade das juntas durante a execução da trajetória, respectivamente. Na figura 36 apresenta-se o erro de posição (a), em escala de ordem 10^{-3} e de orientação (b), em escala de ordem 10^{-5} . Os valores mantêm-se abaixo da tolerância, podendo ser considerados nulos na prática.

Figura 35 – Execução da trajetória elíptica (Movemaster).

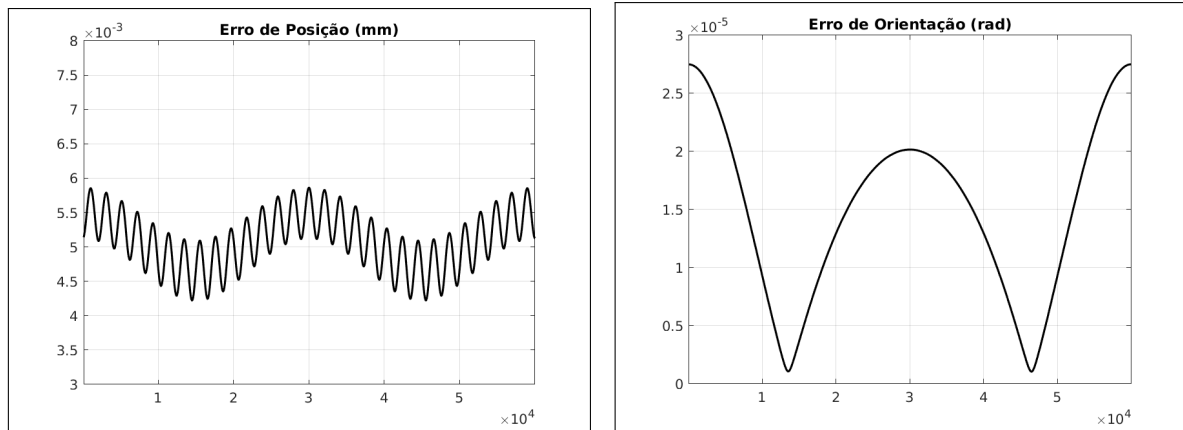


(a) Ângulo das juntas.

(b) Velocidade das juntas.

Fonte: Autoria própria.

Figura 36 – Erro ao longo da trajetória elíptica (Movemaster).



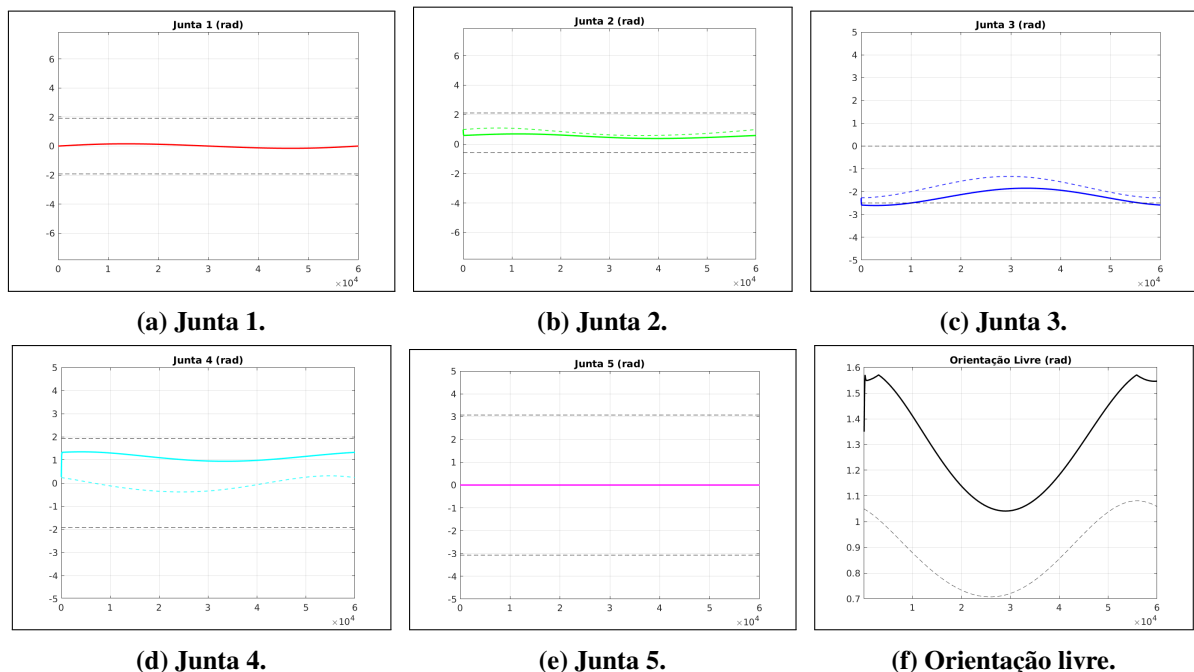
(a) Erro de posição.

(b) erro de orientação.

Fonte: Autoria própria.

De forma análoga ao realizado com o ORCA, o experimento 3 objetiva remover o controle de orientação do Movemaster, tornando-o redundante, e maximizar a distância para os limites de junta. O ganho secundário utilizado é $k_0 = 500$. Os resultados podem ser observados na figura 37(a-e), onde observa-se o comportamento das juntas em relação a seus limites físicos. Na figura 37(f) observa-se a orientação livre de restrições durante a trajetória. As linhas contínuas indicam o comportamento das juntas com a maximização de objetivo secundário e as linhas tracejadas caso $k_0 = 0$.

Figura 37 – Trajetória com objetivo secundário (Movemaster).



(a) Junta 1.

(b) Junta 2.

(c) Junta 3.

(d) Junta 4.

(e) Junta 5.

(f) Orientação livre.

Fonte: Autoria própria.

Observa-se ao longo da figura 37 que apenas as juntas 2, 3 e 4 apresentaram comportamento diferente ao ativar o objetivo secundário. Tal comportamento pode ser analisado do ponto de vista do espaço nulo da Jacobiana. Com o auxílio do Matlab é possível obter o espaço nulo a qualquer momento com a função $null(J)$ e observa-se que apenas as linhas 2, 3 e 4 são diferentes de zero. Dessa forma, apenas estas juntas colaboram para a maximização do objetivo secundário.

Também é possível observar que o algoritmo apresenta falhas para este caso. A junta 3 transita fora da região limite, o que é impossível na prática. A junta 4 tem um desempenho mais próximo do limite com o objetivo acionado do que se apenas seguisse a trajetória normalmente, o que é contrário à ideia do algoritmo. Para solucionar este problema é proposto um algoritmo como uma das contribuições desta dissertação.

4.5 ALGORITMO PROPOSTO PARA OBJETIVO SECUNDÁRIO

Para solucionar a limitação observada no experimento 3 realizado com o Movemaster, propõe-se um algoritmo que age individualmente em cada junta que violar seu limite. O algoritmo proposto deve ser executado após o cálculo dos novos valores de q na cinemática inversa. Trata-se de um laço para verificar o estado de cada junta em relação aos seus limites independentemente. Caso a junta esteja fora de seus limites, o ganho k_0 é aumentado *apenas para esta junta*. Em oposição a tratar o ganho secundário como um escalar, trata-se como um vetor de dimensão n onde cada elemento está associado a uma junta.

Algoritmo 1 – Proposta para objetivo secundário.

```

1: para  $i = 1$  to  $n$  faça
2:   se  $\theta_i > \theta_{max_i}$  então
3:      $\theta_i = \theta_{max_i}$ 
4:      $k_{0_i} = 1,01k_{0_i}$ 
5:   finaliza se
6:   se  $\theta_i < \theta_{min_i}$  então
7:      $\theta_i = \theta_{min_i}$ 
8:      $k_{0_i} = 1,01k_{0_i}$ 
9:   finaliza se
10:   $k_0 = \text{diag}(k_0)$ 
11: finaliza para

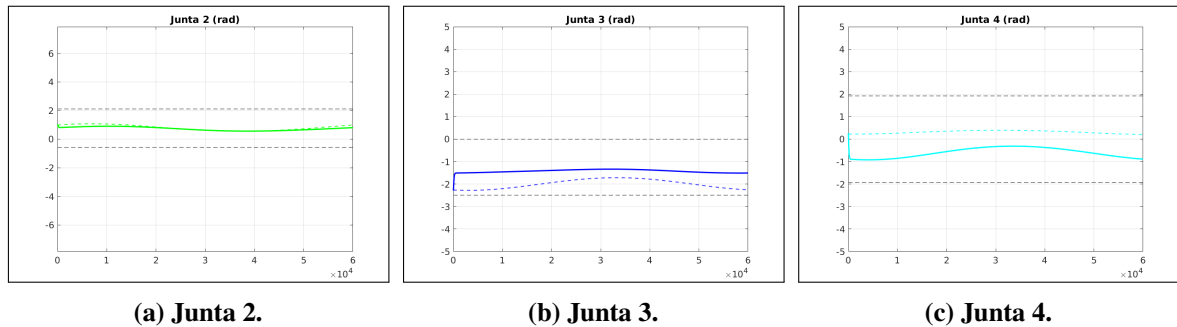
```

Fonte: Autoria própria.

Quando o algoritmo acusa que a junta i está fora de seu limite, k_{0_i} é aumentado em 1%. Tal variação é suficiente para gerar os resultados desejados, os quais são observados na figura 38, onde ao utilizar a técnica descrita, obtém-se o comportamento das juntas representados como

linhas coloridas contínuas. As linhas coloridas tracejadas representam o algoritmo clássico e as tracejadas em preto são os limites físicos das juntas. As demais juntas são omitidas uma vez que não são afetadas por esta técnica. O perfil de erro de posição permanece o mesmo, porém o comportamento das juntas agora é compatível com o desejado em um manipulador real, onde os limites não são violados.

Figura 38 – Trajetória das juntas corrigida pelo algoritmo proposto.



Fonte: Autoria própria.

Tais resultados validam o sistema de geração de trajetória no espaço das juntas. O algoritmo proposto é, portanto, válido para a implementação dos experimentos a serem descritos no Capítulo 6.

4.6 CONSIDERAÇÕES

Neste capítulo foi realizada a aplicação dos conceitos fundamentais da cinemática em manipuladores robóticos, partindo das notações elementares, definição das operações de transformação de *frames* de referência e estabelecendo as equações de cadeias cinemáticas.

A cinemática direta foi utilizada nos termos de parâmetros DH e Helicóides Sucessivos, duas técnicas de representação matemática das estruturas robóticas que possuem diferenças fundamentais apesar de expressar resultados análogos.

O capítulo seguiu com uma seção sobre a cinemática inversa diferencial onde discorreu-se sobre a formulação do cálculo numérico em termos de velocidade de junta. Métodos para tornar os sistemas robustos a singularidades foram descritos. O entendimento do mapeamento entre o espaço das juntas para o espaço operacional levou ao raciocínio sobre a exploração do espaço redundante da Jacobiana, o que permitiu discorrer sobre os objetivos secundários.

O algoritmo de malha fechada para a cinemática inversa diferencial foi descrito neste capítulo, onde todos os conceitos previamente apresentados foram demonstrados em experimen-

tos com dois manipuladores seriais de estrutura diferente. Observou-se uma limitação no sistema de controle de objetivos secundários ao controlar os limites de junta do Movemaster.

Um algoritmo foi proposto para solucionar o problema de controle secundário, o qual foi submetido a teste prático e validado, sendo este a primeira contribuição desta dissertação.

5 CINEMÁTICA ATRAVÉS DA ÁLGEBRA DE CLIFFORD

Em face das limitações introduzidas pelas técnicas clássicas de cinemática, discorre-se neste capítulo sobre a álgebra de Clifford, tendo ênfase em um de seus subgrupos, o qual compreende os quatérnios duais, e na possibilidade de utilizar suas propriedades na representação de pontos, linhas e planos. Os conceitos fundamentais dos quatérnios duais na área de cinemática, além das definições da álgebra de Clifford e como elas se relaciona na robótica, estão descritos em maiores detalhes no apêndice F. Recomenda-se ao leitor que não está familiarizado com estes conceitos a leitura do apêndice antes de prosseguir neste capítulo.

Neste capítulo, discorre-se sobre a aplicação dos quatérnios duais como alternativa às MTH. Em seguida, apresentam-se relações que permitem descrever objetos presentes na área de trabalho do manipulador utilizando a álgebra de Clifford, também demonstrando exemplos numéricos.

5.1 QUATÉRNIOS DUAIS NA CINEMÁTICA DIRETA

O objetivo desta seção é aplicar a álgebra dual-quatérniônica na solução da cinemática direta, substituindo totalmente a aplicação das MTH. Para tal, é necessário criar um algoritmo capaz de solucionar a transformação de *frames* sucessivamente na cadeia cinemática, porém em termos da álgebra de Clifford. O algoritmo deve fornecer a solução para a equação 15, a qual representa a solução da cinemática direta através de quatérnios duais. O detalhamento desta equação encontra-se no apêndice F.

$$\xi' = (h_n \dots (h_2 (h_1 (\xi) h_1^* h_2^*) \dots h_n^*)) \quad (15)$$

Para tal, criou-se uma função no Matlab para simplificar a construção dos elementos de transformação h_n do manipulador Movemaster. A função `make_h` recebe como parâmetros de entrada o eixo de rotação `axis` no formato de um vetor \mathbb{R}^3 unitário; o ângulo de rotação sobre este eixo `theta` em radianos; um vetor de translação ou posição `transl`; e um parâmetro `order`, definido como RT para indicar rotações seguidas de translações (é possível utilizar TR porém não há aplicação nesta dissertação).

Caso a convenção seja feita pelos parâmetros DH:

- $\text{axis} = [0, 0, 1]^T$. Todas as juntas são definidas com rotação ao longo de z ;
- $\text{theta} = \theta_i$. É o valor da variável de junta q do *link* i ;
- $\text{transl} = [a_i \cos(\theta_i), a_i \sin(\theta_i), 0]^T$. Definido como as projeções nos eixos x e y do link de comprimento a_i , sendo a_i um dos parâmetros DH. A projeção no eixo z é sempre zero porque esse é o eixo de rotação da junta.

Caso a convenção seja feita por Helicoides Sucessivos:

- $\text{axis} = s_i$. Parâmetro que indica o eixo de rotação da junta i ;
- $\text{theta} = \theta_i$. É o valor da variável de junta q do *link* i ;
- $\text{transl} = d$. O vetor d é a parcela de translação definida conforme equação 62, onde os elementos são obtidos a partir da computação da rotação de Rodrigues (equação 61) — ver apêndice D.

Ao gerar a solução da equação 15 utilizando tanto os parâmetros DH quanto os Helicóides Sucessivos para ambos manipuladores ORCA e Movemaster, a solução encontrada para posição e orientação é idêntica ao relatado ao longo dos experimentos descritos no Capítulo 4. Os resultados estão omitidos por questão de concisão uma vez que são os mesmos valores já apresentados.

As vantagens no uso dos quatérnios duais na solução de equações cinemáticas são descritos na literatura relacionada. Kenwright (2012) afirma que os quatérnios duais representam uma quebra da norma dentro da cinemática, fornecendo soluções elegantes, unívocas e de fácil interpretação, carregando informações de translação e rotação em uma única variável de estado. O emprego desta álgebra não é trivial e é pouco utilizada em comparação às matrizes de transformação homogênea devido à sua familiaridade e rápido aprendizado.

Além da interpretação subjetiva fornecida por Kenwright (2012), as vantagens no uso desta álgebra são avaliados no quesito de quantidade de operações matemáticas necessárias no trabalho de Aspragathos e Dimitros (1998), onde é demonstrado que exige-se menos cálculos para computar um quatérnio dual do que uma matriz de transformação homogênea. Oliveira (2011) mostra um experimento onde foi utilizado um processador dedicado à contagem de ciclos de *clock* para medir o desempenho computacional entre as operações de MTH e de quatérnios duais. O custo computacional da cinemática sendo solucionada por quatérnios duais é acumulado ao

longo das iterações de *loops* de controle onde a posição do efetuador é verificada constantemente. A redução do custo computacional já provada em literatura, associada à possibilidade de extrair ângulos de orientação sem riscos de indefinições matemáticas, podendo realizar a transformação completa de cadeias cinemáticas com variáveis de estado compactas, faz com que a álgebra dual-quaterniônica seja a técnica de escolha para o sistema desenvolvido neste projeto.

5.2 PONTOS, LINHAS E PLANOS NA ÁLGEBRA DE CLIFFORD

A aplicabilidade da álgebra de Clifford na cinemática não se limita apenas a rotações e translações com quatérnios duais. Elementos matemáticos como pontos, linhas e planos podem ser descritos e manipulados ao aplicar as propriedades de $Cl(0,3,1)$, o grupo dos movimentos de corpos rígidos do qual os quatérnios duais são caso especial, conforme relatado no apêndice F. Selig (2000) publicou o trabalho original que descreve tais relações, as quais são revisadas nesta seção com exemplos. Todas as relações aqui descritas são referenciadas a Selig (2000), a não ser que explicitado o contrário.

5.2.1 Pontos

Apresenta-se no apêndice F a representação de um ponto no espaço tridimensional na notação de quatérnio dual na equação 99. Uma notação alternativa para o ponto é descrita conforme equação 16. Nota-se a utilização de elementos de grau ímpar, portanto já não se pode classificar estas relações como pertencentes apenas ao grupo dos quatérnios duais, mas sim considerando toda a álgebra $Cl(0,3,1)$.

$$p = e_1e_2e_3 + xe_2e_3e + ye_3e_1e + ze_1e_2e \quad (16)$$

Esta representação é diferente da utilizada na equação 99 e é utilizada por Selig (2000) para levar a relações simplificadas envolvendo linhas e planos. O autor também menciona que uma representação igualmente válida tem o coeficiente do elemento $e_1e_2e_3$ como -1 ao invés de $+1$. Isso levaria à interpretações que não tem significado físico.

5.2.2 Linhas

Uma forma de definir matematicamente uma linha no espaço tridimensional é através de dois vetores: um vetor unitário v que aponte na direção da linha e um vetor de momento $u = r \times v$, sendo r um ponto qualquer que coincida com a linha. Nesta relação, nota-se que u e v são ortogonais.

Para representar uma linha na álgebra de Clifford, utiliza-se a forma descrita na equação 17.

$$\ell = (v_x e_2 e_3 + v_y e_3 e_1 + v_z e_1 e_2) + (u_x e_1 e + u_y e_2 e + u_z e_3 e) \quad (17)$$

Esta relação cumpre os requisitos para que u e v sejam ortogonais e v seja unitário. Também há uma relação de direcionalidade, uma vez que ℓ e $-\ell$ resultam na mesma linha porém com direção oposta.

5.2.3 Planos

Um plano pode ser definido matematicamente através de um vetor unitário n e um escalar d representando a distância perpendicular até a origem do sistema, respeitando a relação $n \cdot r = d$, em que r é um ponto qualquer que coincida com o plano.

Na álgebra de Clifford, um plano pode ser representado na forma descrita na equação 18, onde o vetor $n = [n_x, n_y, n_z]^T$ indica a norma do plano.

$$\pi = n_x e_1 + n_y e_2 + n_z e_3 + de \quad (18)$$

Nesta relação também há indicação de direcionalidade, uma vez que π e $-\pi$ resultam no mesmo plano os quais podem ser interpretados como contendo suas faces invertidas.

5.2.4 Hodge Star

O operador Hodge Star, ou *Operador Dual de Hodge* é um operador de mapeamento, descrito por Hodge e Pedoe (1947), o qual retorna o dual de elementos algébricos. No grupo do movimento de corpos rígidos não é possível realizar o mapeamento entre cada elemento e seu dual uma vez que o processo envolve multiplicar as bases por si mesmas. Os elementos que contém a base geradora e tornam-se nulos uma vez que $e^2 = 0$ por definição.

O conceito de um Operador Dual de Hodge é amplo e utilizado em áreas da matemática que envolvem mapeamento, topologia e análise de elementos finitos. A análise detalhada das definições que envolvem este operador são extensas e seu estudo requer um projeto dedicado e direcionado à matemática, o qual fica fora do escopo desta dissertação. Para o presente propósito, basta a compreensão das relações descritas nesta seção, uma vez que este operador está presente apenas nas etapas intermediárias dos cálculos que resultam em valores facilmente visualizados em representação tridimensional. A correta aplicação é demonstrada em exemplos numéricos relacionados aos experimentos práticos. O leitor que deseja obter informações detalhadas sobre a ampla área de estudos que envolve este operador deve consultar os trabalhos de Tarhasaari et al. (1999) Dray (1999), Hirani et al. (2013) e Mohamed et al. (2016).

Selig (2000) define este operador para as bases geradoras em $Cl(0,3,1)$ e estende o conceito a toda a álgebra demandando linearidade e partindo do requisito que $a(\star a) = e_1 e_2 e_3 e$. O Hodge Star de cada base geradora é dado conforme:

$$\begin{array}{ll}
 \star 1 = e_1 e_2 e_3 e & \star e_1 e_2 e_3 e = 1 \\
 \star e_1 = e_2 e_3 e & \star e_2 e_3 e = -e_1 \\
 \star e_2 = e_3 e_1 e & \star e_3 e_1 e = -e_2 \\
 \star e_3 = e_1 e_2 e & \star e_1 e_2 e = -e_3 \\
 \star e = -e_1 e_2 e_3 & \star e_1 e_2 e_3 = e \\
 \star e_1 e_2 = e_3 e & \star e_3 e = e_1 e_2 \\
 \star e_2 e_3 = e_1 e & \star e_1 e = e_2 e_3 \\
 \star e_3 e_1 = e_2 e & \star e_2 e = e_3 e_1
 \end{array}$$

Definir um operador Hodge Star válido para $Cl(0,3,1)$ é fundamental para estabelecer as relações geométricas entre pontos, linhas e planos. Nas seções seguintes, estas relações são aplicadas para a definição de equações que resultam nos objetos tridimensionais que podem ser visualizados e manipulados nos experimentos de cinemática.

5.2.5 Um Ponto e Um Plano

É possível estabelecer relações entre um ponto e um plano descritos na álgebra de Clifford conforme equações 16 e 18. Para determinar a distância mínima l entre um ponto e um plano qualquer, utiliza-se a relação $l = n \cdot r - d$, sendo que esta distância é positiva caso o ponto esteja do mesmo lado do plano que o vetor normal aponta.

Em termos de Clifford, tem-se a equação 19, onde nota-se que a relação retorna o dobro da distância como coeficiente de $e_1e_2e_3e$.

$$\pi p - p\pi = 2le_1e_2e_3e \quad (19)$$

Apesar de ser possível extrair l diretamente da equação 19, também é possível expressar l de forma explícita conforme equação 20.

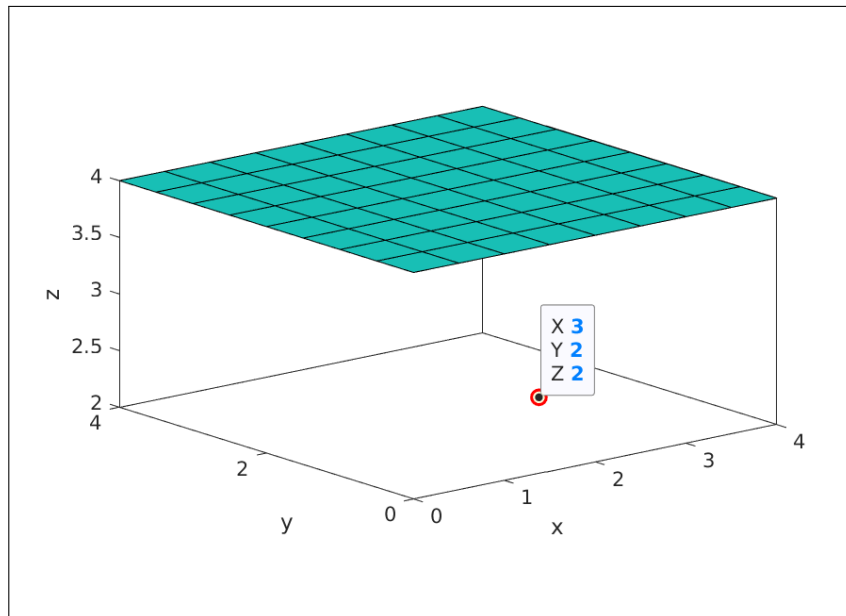
$$l = \frac{1}{2} \star (\pi p - p\pi) \quad (20)$$

Caso a distância l seja zero, significa que o ponto está em intersecção com o plano, portanto para este caso a condição descrita na equação 21 é verdadeira.

$$\pi p - p\pi = 0 \quad (21)$$

Para exemplificar esta relação, toma-se o ponto $r = [1, 3, 4]^T$ que intersecta um plano cuja normal é $n = [0, 0, -1]^T$. Através da álgebra de Clifford é possível computar a distância mínima entre o ponto $p = [3, 2, 2]^T$ e este plano. O cenário é ilustrado na figura 39.

Figura 39 – Relação entre um ponto e um plano na Álgebra de Clifford.



Fonte: Autoria própria.

Descrevendo o ponto p em Clifford, tem-se:

$$p = e_1e_2e_3 + 2e_1e_2e - 2e_1e_3e + 3e_2e_3e \quad (22)$$

Descrevendo o plano em Clifford, tem-se:

$$\pi = -e_3 - 4e \quad (23)$$

A equação 24 indica o resultado obtido ao substituir o ponto p e o plano π na equação 19. Uma vez que o coeficiente do elemento de grau $e_1e_2e_3e$ é o dobro da distância mínima entre o ponto e o plano, conclui-se que a solução é $l = 2$, o que está de acordo com a formulação do problema representada na figura 39.

$$\pi p - p\pi = 4e_1e_2e_3e \quad \therefore l = 2 \quad (24)$$

Para o exemplo aqui elaborado, utilizou-se um plano ortogonal ao eixo z por duas razões: esta configuração facilita a visualização dos valores para fins de validação, e um plano como o apresentado é análogo a uma das limitações na área de trabalho para um manipulador em ambiente confinado. Como π está descrito dentro do grupo do movimento de corpos rígidos, é possível aplicar transformações de rotação e translação utilizando a equação 95 a planos da mesma maneira que o realizado com sistema de referência na cinemática.

5.2.6 Três Pontos Como um Plano

Dado um conjunto de três pontos não co-lineares no espaço tridimensional, é possível traçar um plano unívoco que os englobe simultaneamente. Para definir o elemento que representa este plano, parte-se da relação descrita na equação 25.

$$\begin{aligned} \sigma = & (\star p_1)(\star p_2)(\star p_3) - (\star p_2)(\star p_1)(\star p_3) \\ & + (\star p_2)(\star p_3)(\star p_1) - (\star p_3)(\star p_2)(\star p_1) \\ & + (\star p_3)(\star p_1)(\star p_2) - (\star p_1)(\star p_3)(\star p_2) \end{aligned} \quad (25)$$

A solução torna-se degenerada quando os pontos são co-lineares, tornando impossível definir um plano único e fazendo com que $\sigma = 0$. O plano definido por três pontos na álgebra de Clifford é descrito como equação 26.

$$\pi = (\star \sigma) / \sqrt{(\star \sigma)(\star \sigma)^*} \quad (26)$$

Um ponto importante a destacar na equação 26 é que o termo do qual se está extraindo a raiz quadrada sempre irá resultar em um escalar. Não é possível extrair a raiz quadrada de

elementos de álgebras com bases nilpotentes ($e^2 = 0$), porém para este caso a base nilpotente sempre será zero, permitindo o cálculo normalmente.

Para validar a relação entre três pontos e um plano, tome-se os pontos $p_1 = [1, 2, 3]^T$, $p_2 = [2, -2, 3]^T$, $p_3 = [-1, 1, 0]^T$. Estes pontos são descritos em Clifford conforme:

$$p_1 = e_1e_2e_3 + 3e_1e_2e - 2e_1e_3e + e_2e_3e \quad (27)$$

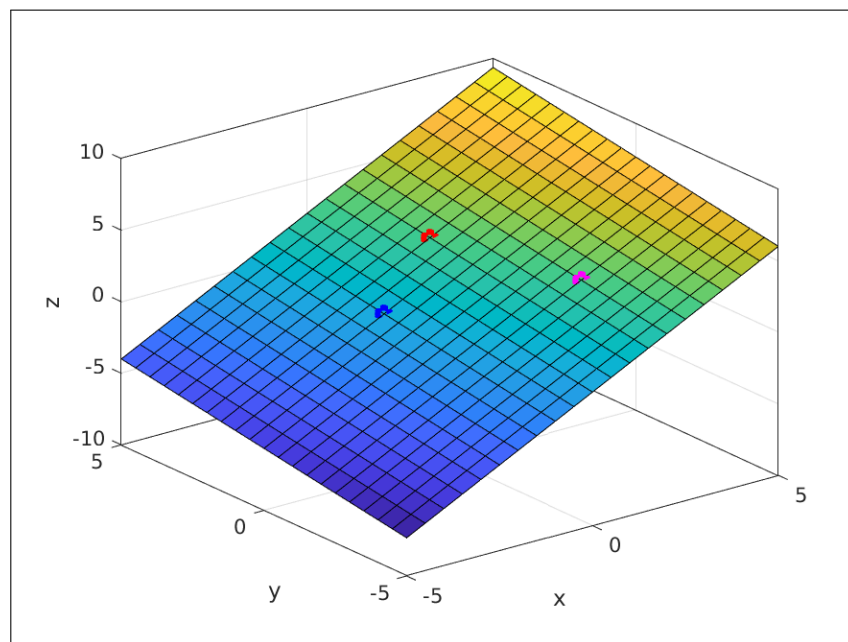
$$p_2 = e_1e_2e_3 + 3e_1e_2e + 2e_1e_3e + 2e_2e_3e \quad (28)$$

$$p_3 = e_1e_2e_3 - e_1e_3e - e_2e_3e \quad (29)$$

Ao substituir a representação dos pontos na equação 26, obtém-se o resultado descrito na equação 30. Nota-se que o resultado está de acordo com a notação de um plano em Clifford, o qual é reproduzido na figura 40, onde é possível verificar que, de fato, os pontos p_1 , p_2 e p_3 intersectam o plano π .

$$\pi = -0,7845e_1 - 0,1961e_2 + 0,5883e_3 + 0,5883e \quad (30)$$

Figura 40 – Três pontos definindo um plano na álgebra de Clifford.



Fonte: Autoria própria.

Os exemplos numéricos descritos nesta seção estão relacionados aos experimentos descritos ao longo do Capítulo 6. Utilizou-se a biblioteca *Clifford Multivector Toolbox*¹ para Matlab (SANGWINE; HITZER, 2022). Esta *toolbox*, descrita em artigo e mantida pelos autores Sangwine e Hitzer (2017), traz todas as operações de interesse em qualquer assinatura de álgebra $Cl(p,q,r)$ e permite futuras explorações das suas aplicações em cinemática.

5.3 CONSIDERAÇÕES

Neste capítulo discorreu-se sobre a álgebra de Clifford com foco no Grupo do Movimento de Corpos Rígidos $Cl(0,3,1)$ e em seu caso especial de subálgebra $Cl^+(0,3,1)$ denominada álgebra Dual-Quaterniônica. As propriedades dos quatérnios duais foram aplicadas para a solução da cinemática direta, onde foi demonstrada a possibilidade de aplicação através dos parâmetros DH ou via Helicoides Sucessivos. A representação de orientação através de um quatérnio permite a extração dos ângulos de orientação sem a presença de singularidades de representação. Associando este quatérnio a uma representação dual de translação, tem-se uma variável de estado única e compacta que descreve as transformações de corpos rígidos.

Uma vez que toda a abordagem cinemática pode ser descrita através de elementos associados a helicóides, ou seja, solucionando as equações cinemáticas com quatérnios duais em um manipulador descrito pelo método dos Helicóides Sucessivos, torna-se de interesse de projeto estender a álgebra de Clifford para toda a área de trabalho. Ao expor as relações apresentadas por Selig (2000) para descrever pontos, linhas e planos, tem-se as ferramentas necessárias para modelar objetos no ambiente de trabalho do manipulador. Outras relações com objetos descritos na álgebra de Clifford existem porém não são abordados neste documento por questões de concisão, mantendo a ênfase nas relações aplicáveis ao projeto final.

Ao completar as revisões e aplicações de todas as ferramentas matemáticas no decorrer deste capítulo, tornam-se válidas suas aplicações em projeto.

¹ Disponível em <https://clifford-multivector-toolbox.sourceforge.io/>

6 ADAPTAÇÃO DE TRAJETÓRIA EM AMBIENTE CONFINADO

Como objetivo geral desta dissertação, foi implementado um cenário experimental onde as técnicas de cinemática estudadas ao longo do programa puderam ser aplicadas e analisadas. Neste capítulo, a cinemática através da álgebra de Clifford é combinada com o processamento próprio e exteroceptivo de um manipulador modernizado cumprindo trajetórias em ambiente confinado.

O Capítulo é dividido em três sessões: primeiramente o cenário dos experimentos é descrito. Em seguida, detalha-se a abordagem para obtenção dos resultados, encerrando com sua análise. Os experimentos são realizados em cenário real e simulado.

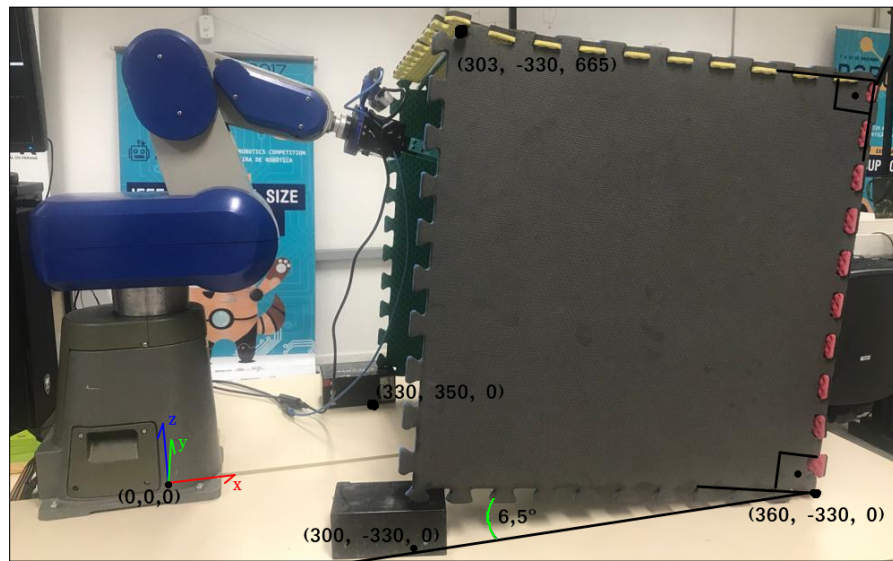
6.1 CENÁRIO EXPERIMENTAL

O objetivo do experimento é obter uma trajetória no espaço das juntas do robô Move-master modernizado, a qual cumpra requisitos de posicionamento para a ferramenta do efetuador sem que as partes intermediárias da estrutura colidam com os limites do espaço operacional real. Os limites deste espaço devem ser menores que o espaço operacional fisicamente possível do robô, ou seja, com limitações dentro de seu alcance. Os limites são mensurados por uma câmera de profundidade e processados via Matlab. A descrição matemática dos planos que formam a área de atuação, bem como a cinemática do manipulador, são realizados via álgebra de Clifford. Ao longo da seção, os passos do algoritmo são expostos para explicar a abordagem. Ao finalizar, apresenta-se um organograma para visualização do algoritmo com maior clareza.

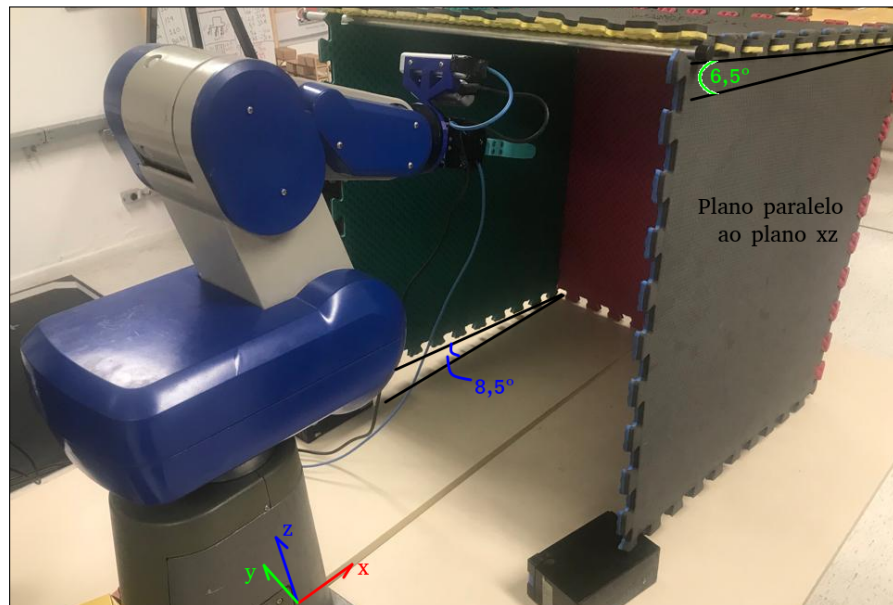
Uma área de trabalho limitada foi montada na frente do local de fixação do Movemaster. A área é móvel e feita de material EVA para que possa ser modificada com novos formatos e limites conforme for de interesse do experimento. Para comparar o algoritmo em diferentes trajetórias, utilizou-se a mesma configuração de área de trabalho. As coordenadas cartesianas para os limites do confinamento são definidas conforme destacado na figura 41, onde os pontos dos vértices definem a área de atuação.

Antes de realizar o processamento dos dados, define-se uma trajetória para o efetuador, a qual consiste em um conjunto de pontos gerados no *software* Matlab. A equação 31 é a estrutura genérica utilizada para geração de trajetórias que apresentam diferentes propriedades ao alterar seus parâmetros. Os vetores $c = [c_x, c_y, c_z]^T$ e $r = [r_x, r_y, r_z]^T$ indicam o centro e o

Figura 41 – Área de trabalho montada para o experimento.



(a) Área de trabalho, vista 1.



(b) Área de trabalho, vista 2.

Fonte: Autoria própria.

raio da trajetória, respectivamente. Considera-se t como o parâmetro que permite particionar a trajetória em seus pontos intermediários, uma vez que definiu-se o intervalo de integração como $\Delta t = 0,001$. A variável t pode ser tratada como tempo de execução, porém para fins de generalização, t assume a função de dar sentido à matemática em termos de geração dos pontos da trajetória. O valor de $T = 30$ tem a função de particionar a trajetória em função de t , podendo ser interpretado como uma constante que define a quantidade de pontos ou como o tempo total da trajetória.

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ \vartheta \\ \psi \end{bmatrix} = \begin{bmatrix} c_x + r_x \cos(2\pi t/T + \pi) \\ c_y + r_y \sin(-2\pi t/T + \pi) \\ c_z + r_z \sin(-2\pi t/T + \pi) \\ \vartheta \\ \psi \end{bmatrix} \quad (31)$$

Conforme exposto no Capítulo 4, para computar a cinemática diferencial é necessário definir o perfil de velocidades instantâneas das juntas. Este perfil é obtido derivando a equação 31 em função de t . O resultado é apresentado na equação 32, a qual serve como perfil de velocidade para todos os cenários propostos.

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{p}_z \\ \dot{\vartheta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \pi \sin(\pi t)/4 \\ \pi \cos(\pi t)/4 \\ \pi \cos(\pi t)/4 \\ 0 \\ 0 \end{bmatrix} \quad (32)$$

Os três cenários propostos são definidos como segue:

1. Uma elipse cujo centro está em $c = [520, 0, 230]^T$ mm. Os raios ao longo dos eixos foram definidos como $r = [60, 280, 140]$ mm. O efetuador inicia a trajetória no ponto mais próximo da base e realiza uma volta completa no sentido horário. O efetuador deve manter um ângulo Pitch $\vartheta = -\pi/3$ rad e Yaw $\psi = 0$ rad. A matriz da equação 33 descreve a trajetória, a qual está ilustrada na figura 42(a).

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ \vartheta \\ \psi \end{bmatrix} = \begin{bmatrix} 520 + 60 \cos(2\pi t/30 + \pi) \\ 280 \sin(-2\pi t/30 + \pi) \\ 230 + 140 \sin(-2\pi t/30 + \pi) \\ -\pi/3 \\ 0 \end{bmatrix} \quad (33)$$

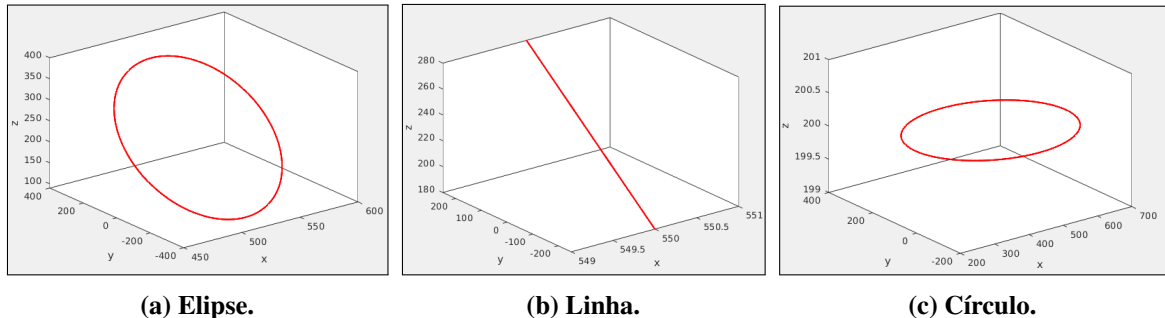
2. Uma reta ligando os pontos $A = [550, 250, 280]$ e $B = [550, -250, 180]$. O efetuador deve manter um ângulo Pitch $\vartheta = 0$ rad e Yaw $\psi = \pi/2$ rad. A matriz da equação 34 descreve o equacionamento utilizado para gerar a trajetória, a qual está ilustrada na figura 42(b).

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ \vartheta \\ \psi \end{bmatrix} = \begin{bmatrix} 550 \\ 250 \operatorname{sen}(-2\pi t/30 + \pi) \\ 230 + 500 \operatorname{sen}(-2\pi t/30 + \pi) \\ 0 \\ \pi/2 \end{bmatrix} \quad (34)$$

3. Um círculo paralelo ao plano yz , de centro $c = [480, 100, 200]$ mm e raio $r = [220, 220, 0]$ mm. O efetuador deve manter um ângulo Pitch $\vartheta = -1.2217$ rad e Yaw $\psi = 0.5236$ rad. A matriz da equação 35 descreve o equacionamento utilizado para gerar a trajetória, a qual está ilustrada na figura 42(c).

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ \vartheta \\ \psi \end{bmatrix} = \begin{bmatrix} 480 + 220 \cos(2\pi t/30 + \pi) \\ 100 + 220 \operatorname{sen}(-2\pi t/30 + \pi) \\ 200 \\ -1.2217 \\ 0.5236 \end{bmatrix} \quad (35)$$

Figura 42 – Trajetórias desejadas para experimento prático.



(a) Elipse.

(b) Linha.

(c) Círculo.

Fonte: Autoria própria.

Para que os experimentos possam ser realizados do ponto de vista de regime estacionário, em todos os cenários o erro inicial deve ser zerado antes da execução da trajetória. Para isso a pose inicial do manipulador é calculada utilizando o método iterativo, partindo da sua configuração *home*. O ponto inicial p_0 de cada trajetória pretendida é:

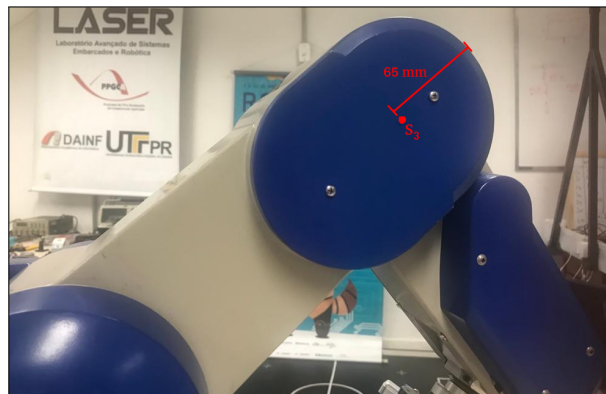
1. $p_0 = [460, 0, 230]^T$ mm, para a elipse;
2. $p_0 = [550, 0, 230]^T$ mm, para a reta;
3. $p_0 = [260, 100, 200]^T$ mm, para o círculo;

Cada uma das três trajetórias é executada em três casos distintos:

1. Trajetória com controle de orientação: neste caso, a trajetória é executada respeitando a restrição de orientação proposta. O espaço nulo da Jacobiana torna-se um conjunto vazio $N(J) = \emptyset$, uma vez que cinco variáveis estão sendo controladas em um manipulador de cinco graus de liberdade, e não é possível aplicar algoritmos de objetivo secundário. Este cenário não pode ser executado no manipulador físico, uma vez que a estrutura do Movemaster tem dificuldades em manter a orientação desejada sem causar auto-colisões e violação dos limites de junta, os quais não podem ser reforçados matematicamente;
2. Trajetória sem controle de orientação e com maximização da distância aos limites físicos: o controle da orientação é desativado. Passa a existir um espaço nulo não-vazio para a Jacobiana, $N(J) \neq \emptyset$ e o objetivo secundário torna-se aplicável;
3. Trajetória sem controle de orientação com algoritmo proposto: repete-se o caso anterior, porém utilizando o algoritmo proposto para maximizar a distância até o limite das juntas.

A espessura da estrutura física do Movemaster não é modelada na cinemática, portanto para definir a distância de segurança, foi medida a espessura da junta 3 em relação ao seu centro matemático - a posição do helicóide s_3 .o helicóide deve se manter a 65 mm dos obstáculos para evitar colisões. A figura 43 mostra a medição realizada. Considera-se uma região com risco *critical* a 80 mm e com risco *warning* a 110 mm dos obstáculos. Nenhuma parte do manipulador deve estar presente a distâncias menores que *critical*. A região *warning* descreve distâncias imediatamente maiores que *critical*, onde o manipulador pode navegar porém com medidas de segurança.

Figura 43 – Medida da espessura do helicóide até a carcaça na junta 3.



Fonte: Autoria própria.

Os algoritmos descritos são generalizáveis a qualquer configuração cinemática de manipulador, bastando alterar a quantidade de graus de liberdade e seus parâmetros cinemáticos. O Movemaster, no entanto, possui uma estrutura que permite simplificações na computação da trajetória livre de colisões. Como a única junta que rotaciona em torno do eixo z do sistema de referência é a junta 1, e todo o restante do manipulador comporta-se de maneira planar, a única forma de haver colisão com os limites $\pm y$ é através do efetuador. O mesmo acontece para colisões com a plataforma de montagem na altura $z = 0$ mm.

Por ser o objetivo do experimento, a trajetória desejada para o efetuador é totalmente conhecida. Isso permite simplificar o tratamento dos limites com um pré-processamento aqui denominado *trimming* nos limites $\pm y$ e $-z$. Os pontos da trajetória que levam o efetuador a distâncias menores que `warning` são substituídos por seu valor em um laço que percorre todos os pontos gerados.

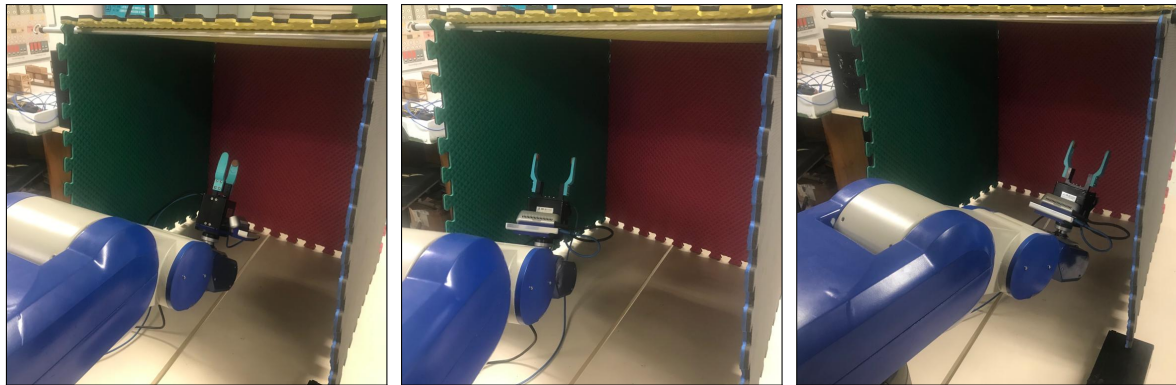
Esta simplificação não pode ser aplicada à região de colisão no limite $+z$. A região da estrutura mais próxima da área superior do confinamento normalmente não é o efetuador. De fato, não há um controle direto do posicionamento da junta mais próximo da região crítica, uma vez que seu posicionamento é consequência das exigências do controle do efetuador. O *trimming* dos limites superiores é realizado ao longo do cálculo da cinemática diferencial, onde é possível acessar a posição do helicóide mais próximo da colisão e ajustar a trajetória de acordo.

O primeiro comando ROS enviado é para posicionar o manipulador na configuração de leitura de *point cloud*. Os ângulos de junta $q = [0, -30, -5, 95, -90]^T$, dados em graus, posicionam o Movemaster como mostra a figura 44(a). Realiza-se uma rotação de 90° na junta 5 para permitir uma área de captura maior do limite em $+z$. Ao reposicionar a junta 5 em 0° e definindo um ângulo de $\pm 10^\circ$ à junta 1, conforme mostrado nas figuras 44(b) e 44(c), a câmera pode capturar as nuvens de pontos que representam os limites em $\pm y$, respectivamente. Os dados são publicados no tópico ROS pela D435 e são lidos e armazenados pelo algoritmo em Matlab. Considera-se ambos os limites em $\pm x$ como fora de alcance.

O algoritmo recebe a árvore de transformações do ROS, o que permite realizar a transformação dos dados da nuvem de pontos coletada pela D435 de modo que sejam visualizados e processados nas mesmas coordenadas do mundo real. O passo seguinte é realizado utilizando ferramentas da *Computer Vision Toolbox*¹. O *Point Cloud* publicado é dado no formato *sensor_msgs/PointCloud2* e a *toolbox* trabalha com dados no formato *pointcloud*. Dessa forma os dados

¹ Disponível em: <https://www.mathworks.com/help/vision/index.html>. (The MathWorks, Inc., 2022)

Figura 44 – Configurações de leitura dos limites de operação.



(a) Leitura $+z$.

(b) Leitura $+y$.

(c) Leitura $-y$.

Fonte: Autoria própria.

são convertidos através da função *ptCloud()* para realizar o processamento.

O objetivo do processamento das nuvens de pontos é filtrar os dados de modo que mantenha-se apenas o que é correspondente ao plano, ou seja, ao limite da área de atuação que foi medido. A *toolbox* conta com a função *pcfitplane()*, a qual permite uma série de configurações para detectar planos em um *pointcloud*. A forma mais simples da função tem como parâmetro de entrada o próprio *pointcloud*, um parâmetro de distância máxima entre os pontos para considerá-los como pertencentes ou não ao plano, definida em 5 mm, e uma distância angular máxima entre os pontos, definida como 4° . Esta função retorna como resultado uma variável do tipo *pointcloud* contendo apenas os pontos pertencentes a um plano.

Outro parâmetro de entrada que permite maior seletividade dos dados filtrados é a norma de referência: como é de conhecimento *a priori* que a área de confinamento é aproximadamente um cuboide, a norma dos planos pode ser norteada. Ao medir o plano $+z$, por exemplo, alimenta-se a função com a indicação de norma $n = [0, 0, -1]$. Isso sinaliza ao algoritmo que o plano processado deve ser o mais próximo de uma norma apontando na direção $-z$, o que evita casos onde o processamento de dados identifica mais de um plano na mesma nuvem de pontos. Para medir o plano em $+y$ utiliza-se $n = [0, -1, 0]$ e em $-y$ utiliza-se $n = [0, 1, 0]$.

Como neste ponto do algoritmo há uma nuvem de pontos que representa um plano, realiza-se a parametrização dentro da álgebra de Clifford. Três pontos quaisquer são selecionados de forma aleatória pelo Matlab para modelar um plano utilizando a equação 26. Uma vez que a etapa de filtro determina que estes pontos de fato pertencem ao mesmo plano, não é necessário escolher quais especificamente devem ser utilizados, uma vez que o equacionamento apenas exige que eles não sejam co-lineares. Na rara possibilidade dos pontos escolhidos serem co-lineares, o

cálculo do plano resulta em $\pi = 0$ e um novo conjunto de pontos é utilizado. As variáveis que armazenam os planos de Clifford, bem como as operações necessárias são realizadas através da biblioteca *Clifford Multivector Toolbox* para Matlab (SANGWINE; HITZER, 2022), já descrita no Capítulo 5.

As informações referentes à localização dos planos são publicados em um tópico ROS, sendo lidas e tratadas pelo nó criado na linguagem C++ denominado *marker_publisher*, dedicado a esta tarefa. Este nó publica mensagens no formato *visualization_msgs/Marker*, as quais são lidas e representadas de maneira visual no Rviz. Com estas informações é possível ver no ambiente virtual a representação do Movemaster e da sua área de operação.

Após essas etapas o algoritmo inicia seu processo de geração de trajetória no espaço das juntas, ou seja, executa a cinemática inversa diferencial através de quatérnios duais e utilizando a convenção dos Helicoides Sucessivos para todos os pontos da trajetória. Os valores dos ângulos de junta iniciais são definidos conforme resultado da inicialização por erro zero. O controle é feito em malha fechada para minimizar o efeito de *drift* de integração.

Como será detalhado na seção 6.2, a não-linearidade do algoritmo dificulta a manutenção da estabilidade mesmo com diferentes combinações de ganho. Isso exige a implementação de uma rotina de convergência da cinemática inversa, executada entre o cálculo de cada ponto da trajetória: o *loop* de controle apenas segue para o ponto seguinte quando encontra uma configuração para o manipulador dentro de um erro de posição de 10 mm.

Após o cálculo de cada ponto da trajetória, o algoritmo verifica a distância mínima entre as juntas do manipulador e o plano de Clifford em $+z$ utilizando a relação entre pontos e planos descrita na equação 20. A posição das juntas é obtida através do cálculo das camadas parciais da equação de transformação dual-quatérniônica já apresentada na equação 15. Para uma estrutura genérica de manipulador, é necessário computar a distância mínima até todos os planos gerados, porém como já demonstrado nesta seção, o Movemaster exige este cálculo apenas para $+z$. Para fins de experimentação e demonstração dos conceitos estudados, não é relevante que o manipulador siga exatamente a trajetória original: o desvio da trajetória desejada e o método como isto é computado foi definido de forma que se torne evidente a contribuição principal do trabalho, a qual é voltada à modelagem e identificação de obstáculos através da álgebra de Clifford.

A junta mais próxima do plano é tratada como *junta crítica*. Caso a distância entre a junta crítica e o plano seja menor que a definida em `critical`, a posição objetivo do efetuador

sofre uma alteração na direção oposta ao plano. Como um plano de Clifford π descrito conforme equação 18 leva a sua normal n como coeficientes dos elementos e_1 , e_2 e e_3 , a direção oposta ao plano pode ser computada pela inversão do sinal dos coeficientes. O novo valor da posição p , denotado ($p_{(novo)}$), é dado conforme equação 36, onde d_{crit} é a distância definida em `critical`, d_{Cliff} é a distância medida entre a junta e o plano, a qual acrescenta-se 10% como medida de segurança, e w é o vetor de sentido do deslocamento dado por $w = -n$.

$$p_{(novo)} = p_{(atual)} - (d_{crit} - 1,1d_{Cliff})w; \quad (36)$$

Caso a junta crítica esteja fora da região `critical`, porém dentro da região `warning`, então o algoritmo toma providências para amortecer o posterior movimento em direção à região onde o risco é maior. Isso é feito observando a direção de movimento atual do efetuador: caso a posição atual do efetuador esteja se distanciando do plano em relação à posição anterior, nada é executado; caso o efetuador esteja se aproximando do plano, sua coordenada mais próxima do plano é alterada para a anterior. Exemplificando, tome-se a limitação paralela ao plano xy . Caso a coordenada do objetivo do efetuador atual $p_z(atual)$ seja maior que a anterior $p_z(anterior)$, então a atual passa a ter o mesmo valor que a anterior. Esta operação é descrita na equação 37.

$$\begin{cases} p_z(novo) = p_z(atual) & \text{se } p_z(atual) < p_z(anterior) \\ p_z(novo) = p_z(anterior) & \text{se } p_z(atual) > p_z(anterior) \end{cases} \quad (37)$$

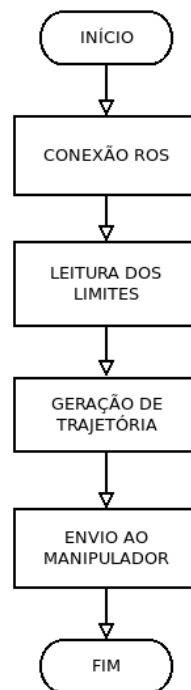
Ressalta-se que as atitudes para adaptação de trajetória estão executando alterações no destino do efetuador, não na junta crítica. Realizar movimentos na ferramenta que a distanciem de regiões de colisão tende a fazer toda a estrutura mover-se neste sentido, porém a relação entre o efetuador e a junta crítica não é linear - como toda a operação cinemática - portanto o sucesso destas alterações não pode ser garantido em todos os cenários em uma iteração. Essa incerteza de sucesso é contornada justamente ao inserir uma região de `warning` para ação preventiva, fazendo com que o algoritmo seja executado com sucesso. Ao estabelecer um ponto substituto na trajetória, a cinemática inversa é calculada novamente e a distância ao plano volta a ser mensurada para que não viole-se a região `critical`.

Independentemente da situação de alteração da trajetória, a variação dos ângulos \dot{q} sempre leva em conta o objetivo secundário de maximização da distância para os limites de junta, computado através das equações 9 e 14. Se o caso em execução leva em conta o controle da orientação, então a parcela do espaço nulo assume valor zero, restando apenas a solução

homogênea. Após a integralização e obtenção dos novos valores de q , os ganhos de cada junta no objetivo secundário é ajustado conforme algoritmo proposto no Capítulo 4. Ao concluir o cálculo de todos os valores de juntas para cada ponto da trajetória, inicia-se o processo de envio para o manipulador em tópico ROS.

A pose real do manipulador é monitorada através dos *encoders*, que fornecem informações sobre o ângulo atual das juntas através do ROS, permitindo o cálculo da cinemática direta e do erro atual. A trajetória também é monitorada por uma câmera de odometria T265, a qual publica no ROS a posição do efetuador virtual através da árvore de transformações. O nó *marker_publisher* recebe as informações e exibe os marcadores visuais da posição do manipulador no Rviz. A figura 45 ilustra os blocos principais do algoritmo no formato de organograma simplificado. Cada bloco apresentado possui seu próprio organograma, os quais estão inseridos no apêndice G de modo a manter a clareza da apresentação.

Figura 45 – Principais blocos funcionais do experimento.



Fonte: Autoria própria.

A trajetória executada pelo manipulador real é comparada aos valores calculados, acompanhado da distância até a limitação do confinamento. O experimento é registrado na forma de vídeo, disponibilizado no respectivo canal do YouTube².

² Disponível em: <https://www.youtube.com/playlist?list=PLFoGgtKo3N-byzv8hi-Sk-ONfSa1dZj7>.

6.2 RESULTADOS DOS EXPERIMENTOS

Esta seção apresenta os resultados coletados dos experimentos feitos com o Movemaster. Primeiramente é demonstrado o resultado das medições de área de trabalho realizada. Em seguida, divide-se os resultados em três subseções de estrutura semelhante, cada uma expressando os resultados de uma das trajetórias definida na seção 6.1. Ao enviar a trajetória para o manipulador, a informação fornecida pelos *encoders* permite monitorar os dados de cada junta e medir a distância mínima até os limites da área de trabalho. Destaca-se ainda que as trajetórias sem controle dos limites de junta não foi enviada ao manipulador físico.

6.2.1 Medição da Área de Trabalho

Ao executar a rotina de posicionamento do manipulador nas configurações de leitura dos limites de operação, o Matlab pôde processar as nuvens de pontos em busca dos planos que puderam ser modelados pela álgebra de Clifford. O resultado dos planos de Clifford indicados pelo algoritmo foram:

- Limitação em $+z$, descrita pelo plano:

$$\pi = 0,0522e_1 + 0,0104e_2 + 0,9986e_3 + 663,1712e$$

- Limitação em $+y$, descrita pelo plano:

$$\pi = -0,0479e_1 - 0,9968e_2 + 0,0634e_3 + 311,4408e$$

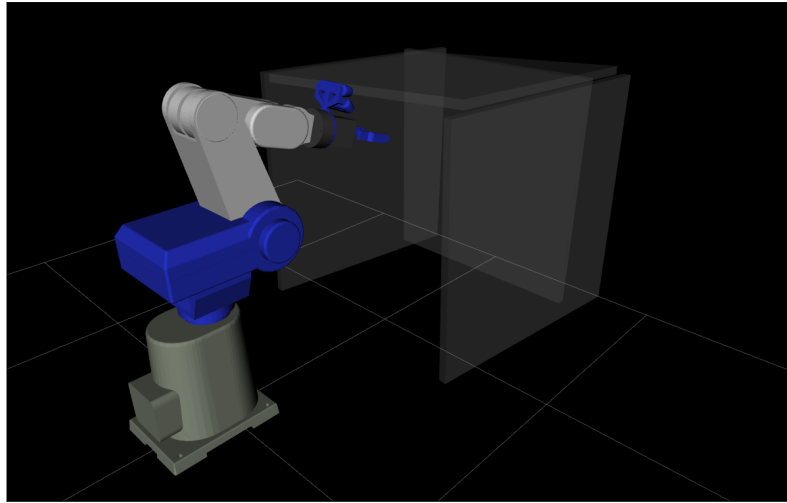
- Limitação em $-y$, descrita pelo plano:

$$\pi = 0,0577e_1 + 0,9954e_2 + 0,0763e_3 + 356,2977e$$

Ao finalizar a detecção dos planos, sua posição é publicada no ROS onde é interpretada pelo nó *marker_publisher*. Os marcadores são então representados no Rviz gerando a imagem apresentada na figura 46, onde a área de trabalho medida é vista junto do modelo URDF do manipulador.

A visualização por marcadores através do Rviz permite o monitoramento das ações do manipulador enquanto executa os comandos recebidos. A área de trabalho virtual apresentada corresponde ao modelo real de modo que é possível prosseguir com a geração de trajetória dentro de seus limites.

Figura 46 – Modelo virtual do manipulador e da área de trabalho medida.

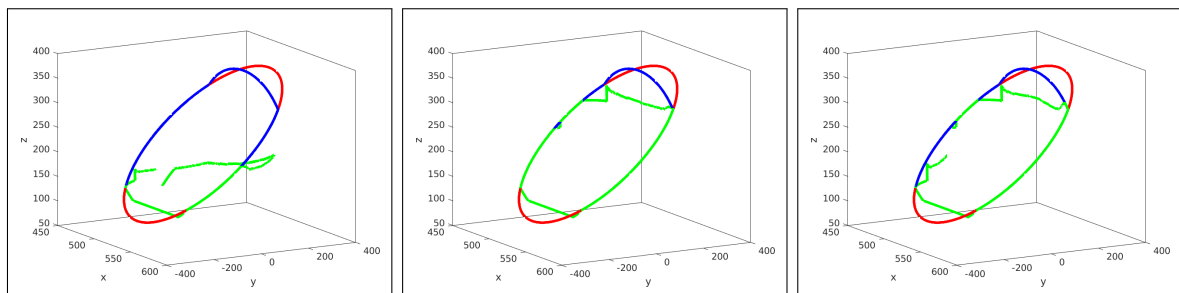


Fonte: Autoria própria.

6.2.2 Trajetória Elíptica

Nesta subseção são apresentados os resultados calculados e medidos para a trajetória elíptica proposta. A figura 47 ilustra em vermelho a trajetória desejada sem considerar as limitações da área de trabalho. O processo de *trimming* gera a trajetória em azul. A trajetória final, destacada em verde, é gerada após o cálculo da cinemática inversa e das distâncias mínimas da junta crítica. A figura divide-se em cada um dos casos de execução propostos. Nas três sub-figuras presentes na figura 47, o trajeto desejado e gerado pelo *trimming* são os mesmos, estando presentes nos três gráficos para melhor visualizar o efeito da mudança de algoritmo em verde.

Figura 47 – Trajetória elíptica processada.



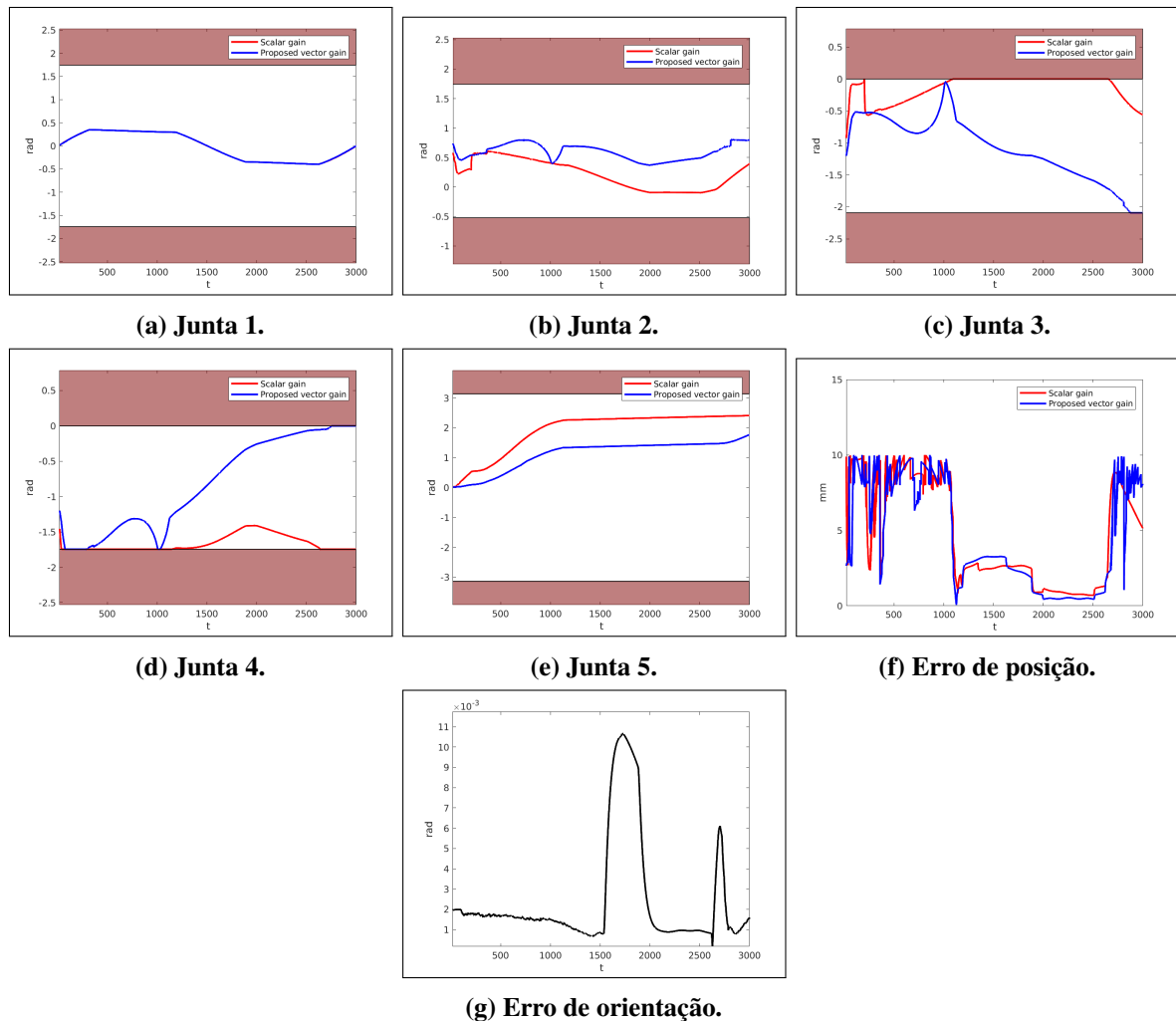
(a) Controle de orientação e posição. (b) Controle de posição e limite de juntas. (c) Controle de posição com algoritmo proposto.

Fonte: Autoria própria.

Na figura 48 observa-se a diferença dos valores computados no espaço das juntas. As sub-figuras 48(a) a 48(e) contém regiões em vermelho que representam a área fora dos limites

físicos das juntas 1 a 5, respectivamente. A sub figura 48(f) contém o erro de posição gerado pelo *drift* de integração, mantido abaixo de 10 mm pela etapa de convergência. Na sub figura 48(g) é mostrado o erro de orientação normalizado utilizando a representação via quatérnios, válido somente para a execução do primeiro experimento da trajetória. A linha vermelha representa o comportamento de cada junta e do erro no efetuador utilizando o algoritmo clássico. A linha azul representa o algoritmo proposto. É notável o desempenho do novo algoritmo na maximização da distância angular até o limite físico das juntas.

Figura 48 – Variáveis de junta e erro durante a trajetória elíptica.

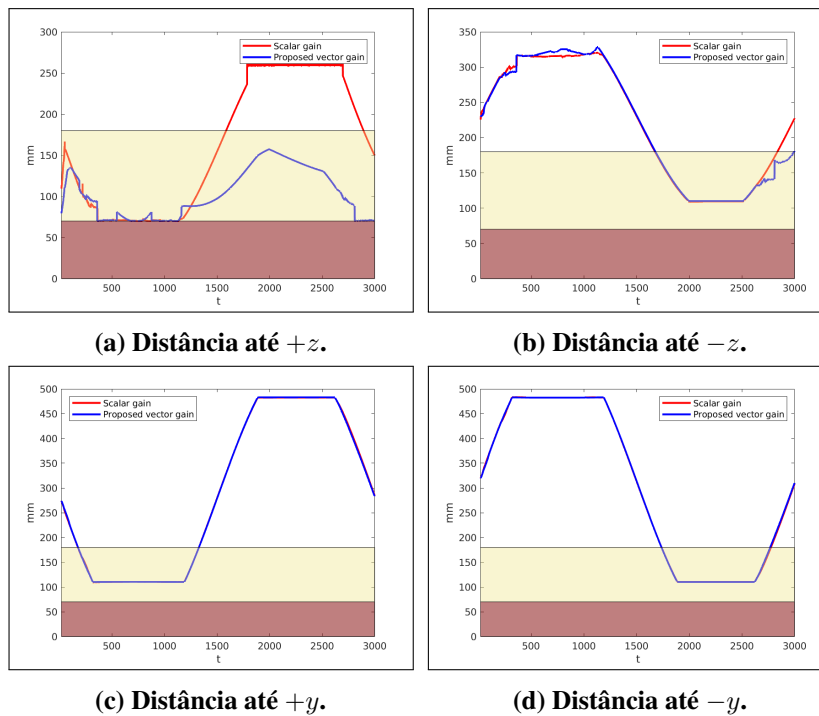


Fonte: Autoria própria.

Ao executar a trajetória com o manipulador — o qual foi feito no modelo real apenas sem o controle de orientação — foi possível medir a distância mínima até as limitações da área de trabalho. Os valores são ilustrados na figura 49, onde é possível comparar o comportamento diferenciado ao utilizar um ganho secundário escalar ou um vetor variável conforme proposto, especialmente nos limites em $\pm z$. A região em vermelho denota a distância crítica e em

amarelo a região *warning*. Nestas imagens nota-se como o manipulador manteve-se a uma distância maior que a distância crítica independente do algoritmo usado. Com o algoritmo proposto, destacado em azul, houve momentos da trajetória em que partes do manipulador mantiveram-se na divisão entre as regiões *critical* e *warning*, devido ao comportamento de toda a estrutura para maximizar o objetivo secundário, não representando uma perda de desempenho uma vez que não há colisão nestas configurações.

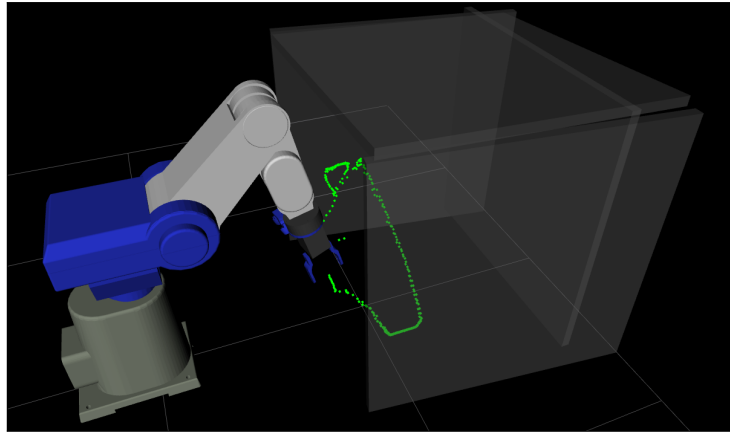
Figura 49 – Distância mínima até os planos durante a trajetória elíptica.



Fonte: Autoria própria.

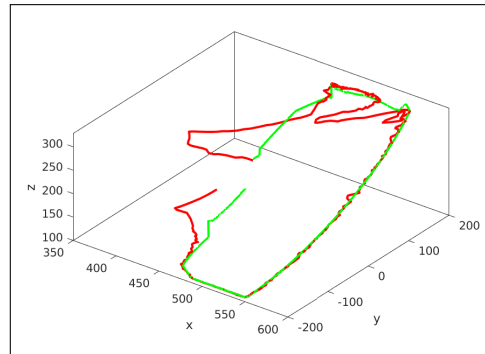
A trajetória que contempla a maximização de distância angular proposta foi enviada ao manipulador, a qual foi registrada em vídeo e em arquivo *rosbag* para análise dos dados a qualquer momento. A visualização do Rviz após o experimento encontra-se na figura 50 onde é possível visualizar nos marcadores verdes a trajetória executada do ponto de vista dos *encoders*. A mesma informação pode ser vista de forma gráfica na figura 51, onde a trajetória vista pelos *encoders* está destacada em vermelho. A trajetória verde indica a informação enviada, ou seja, a trajetória desejada. A execução dos experimentos encontra-se no formato de vídeo no já referido canal do YouTube, onde a execução pode ser melhor analisada. De qualquer forma, momentos da execução são apresentados na figura 52, onde é possível visualizar a execução da trajetória livre de colisões no protótipo real.

Figura 50 – Visualização da execução da trajetória elíptica.



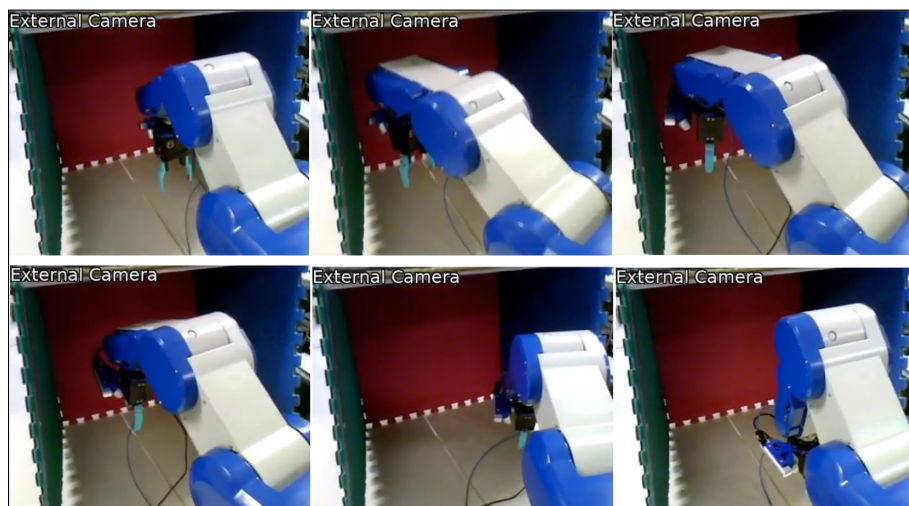
Fonte: Autoria própria.

Figura 51 – Trajetória elíptica medida pelos *encoders*, em vermelho. Trajetória desejada em verde.



Fonte: Autoria própria.

Figura 52 – Momentos da execução da trajetória elíptica no protótipo real.

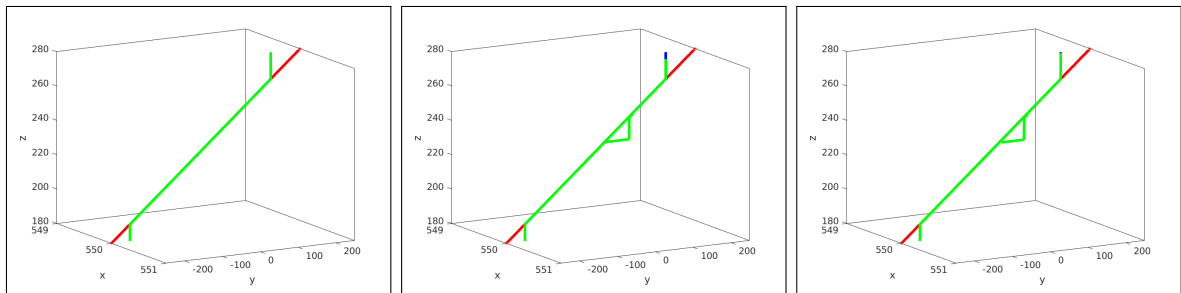


Fonte: Autoria própria.

6.2.3 Trajetória Retilínea

De modo a suprimir a repetição na explicação do conteúdo de cada imagem, os textos estão suprimidos nesta subseção. Os gráficos descritos aqui representam os resultados do experimento medidos durante a execução da trajetória linear. Para os detalhes dos dados expressos, basta consultar a subseção 6.2.2, uma vez que as figuras 47 a 52 são análogas às figuras 53 a 58. Reserva-se a seção 6.3 para a análise dos resultados.

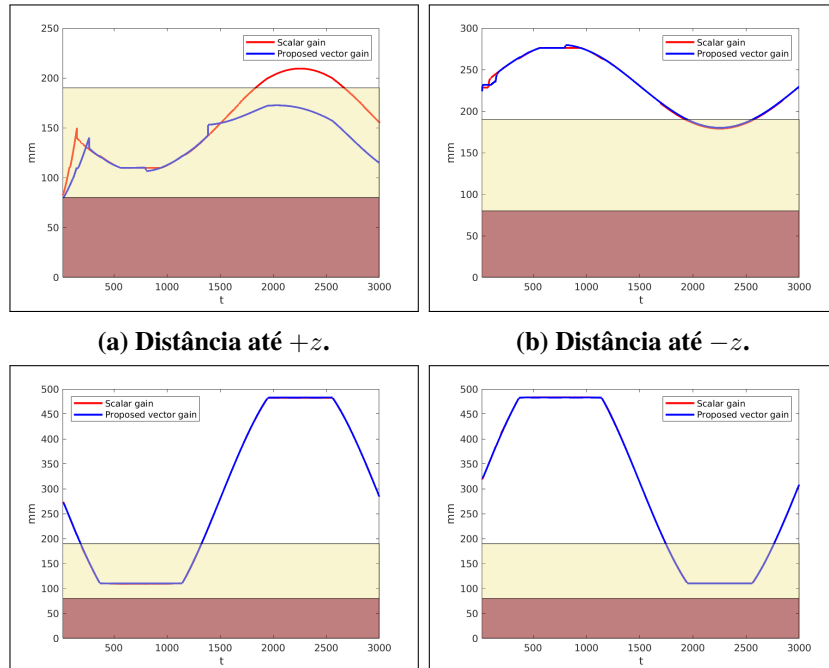
Figura 53 – Trajetória retilínea processada.



(a) Controle de orientação e posição. (b) Controle de posição e limite de juntas. (c) Controle de posição com algoritmo proposto.

Fonte: Autoria própria.

Figura 54 – Distância mínima até os planos durante a trajetória retilínea.



(a) Distância até $+z$.

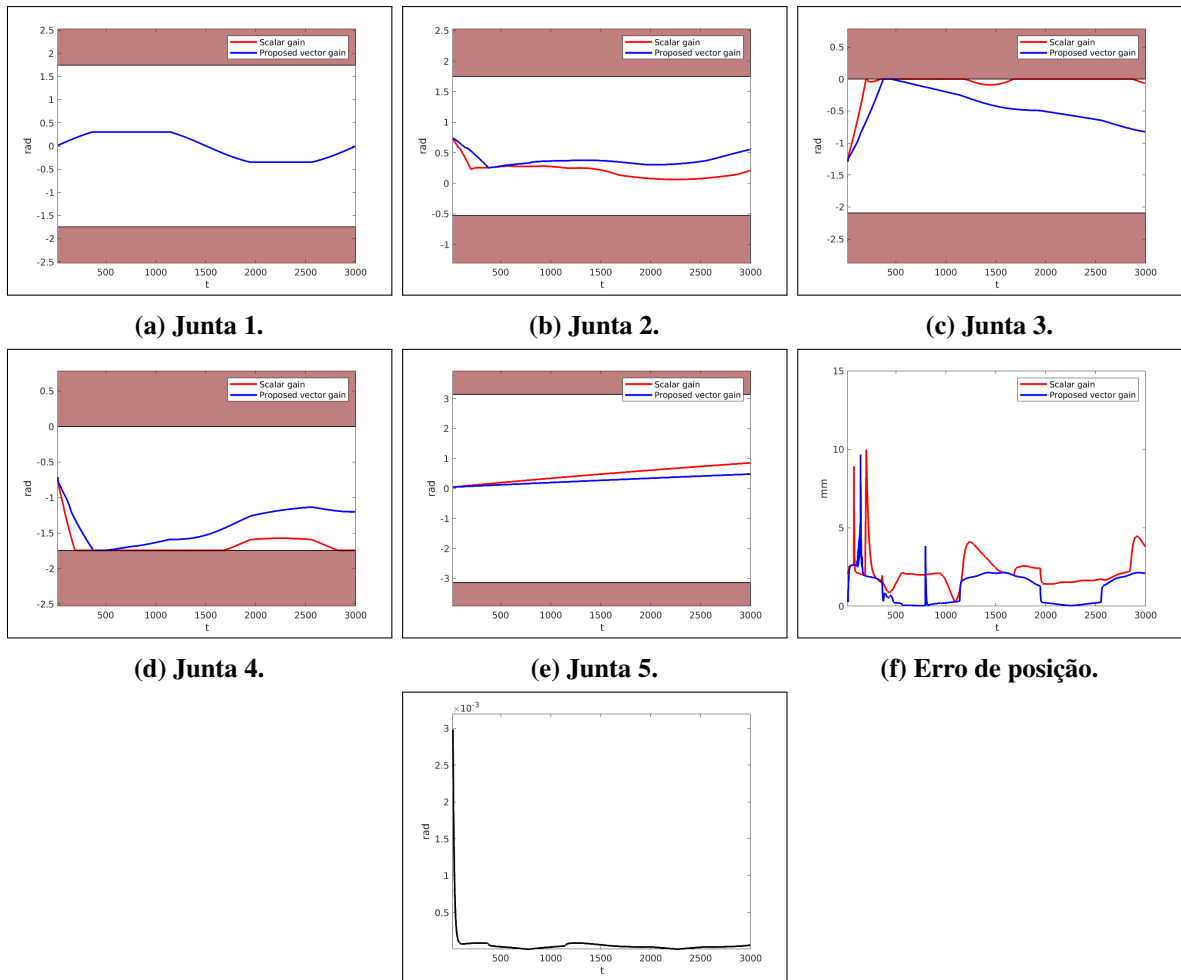
(b) Distância até $-z$.

(c) Distância até $+y$.

(d) Distância até $-y$.

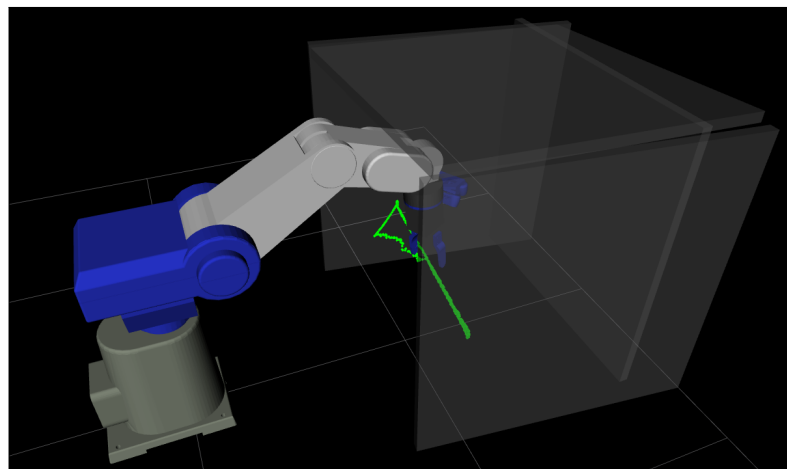
Fonte: Autoria própria.

Figura 55 – Variáveis de junta e erro durante a trajetória retilínea.



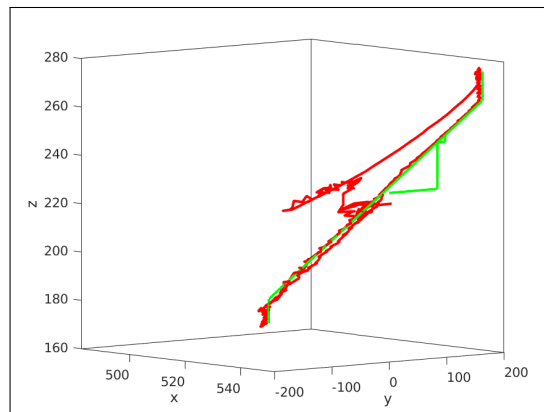
Fonte: Autoria própria.

Figura 56 – Visualização da execução da trajetória retilínea.



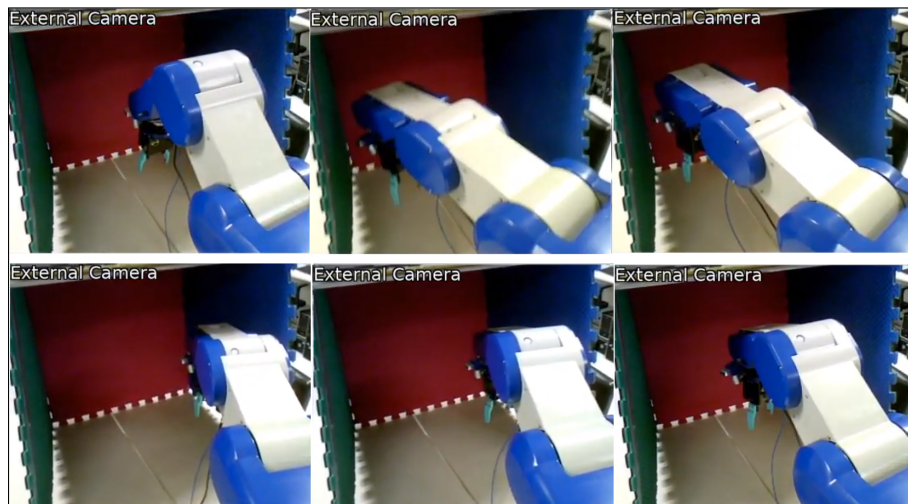
Fonte: Autoria própria.

Figura 57 – Trajetória retilínea medida pelos *encoders*, em vermelho. Trajetória desejada em verde.



Fonte: Autoria própria.

Figura 58 – Momentos da execução da trajetória em linha no protótipo real.

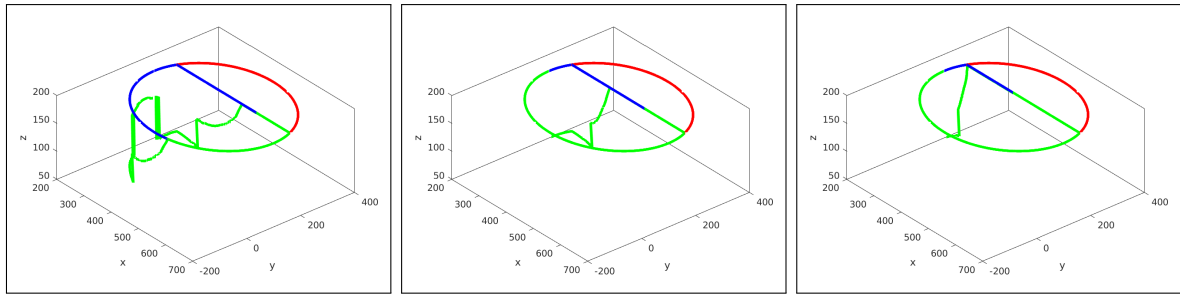


Fonte: Autoria própria.

6.2.4 Trajetória Circular

De modo a suprimir a repetição na explicação do conteúdo de cada imagem, os textos estão suprimidos nesta subseção. Os gráficos descritos aqui representam os resultados do experimento medidos durante a execução da trajetória circular. Para os detalhes dos dados expressos, basta consultar a subseção 6.2.2, uma vez que as figuras 47 a 52 são análogas às figuras 59 a 58. Reserva-se a seção 6.3 para a análise dos resultados.

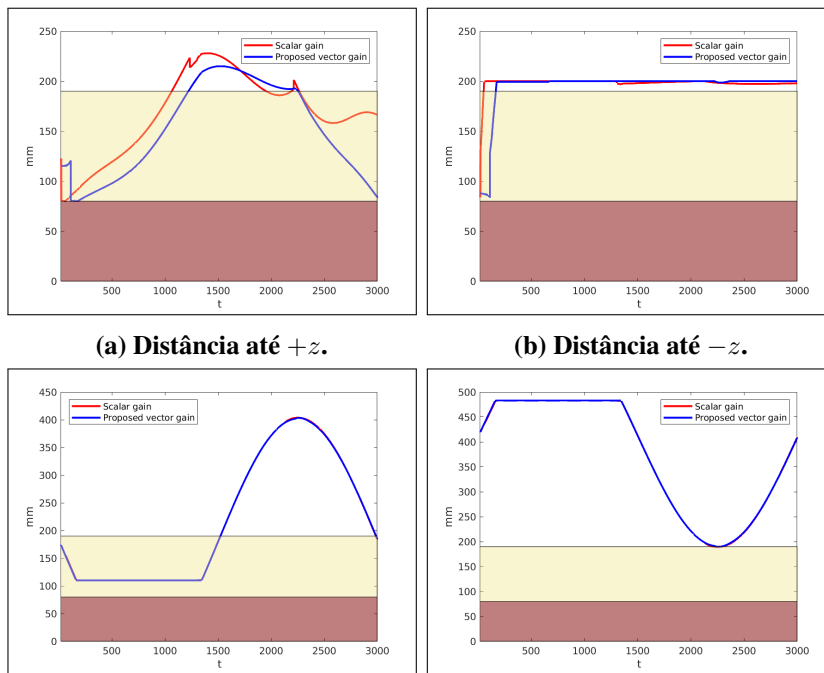
Figura 59 – Trajetória circular processada.



(a) Controle de orientação e posição. (b) Controle de posição e limite de juntas. (c) Controle de posição com algoritmo proposto.

Fonte: Autoria própria.

Figura 60 – Distância mínima até os planos durante a trajetória circular.



(a) Distância até +z.

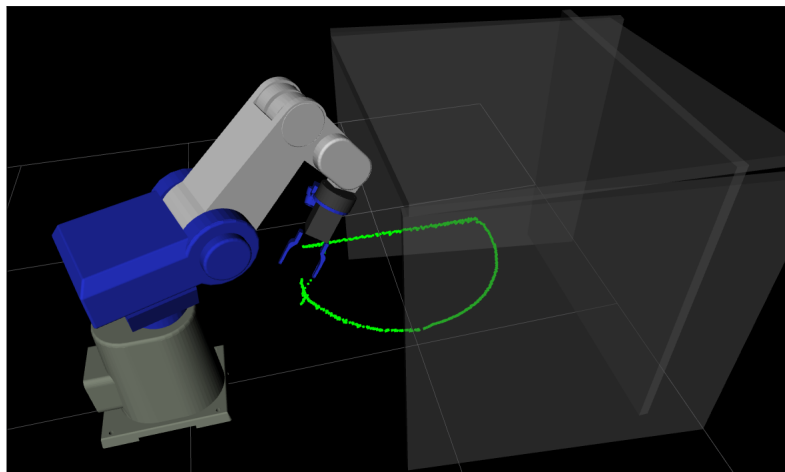
(b) Distância até -z.

(c) Distância até +y.

(d) Distância até -y.

Fonte: Autoria própria.

Figura 62 – Visualização da execução da trajetória circular.



Fonte: Autoria própria.

Figura 61 – Variáveis de junta e erro durante a trajetória circular.

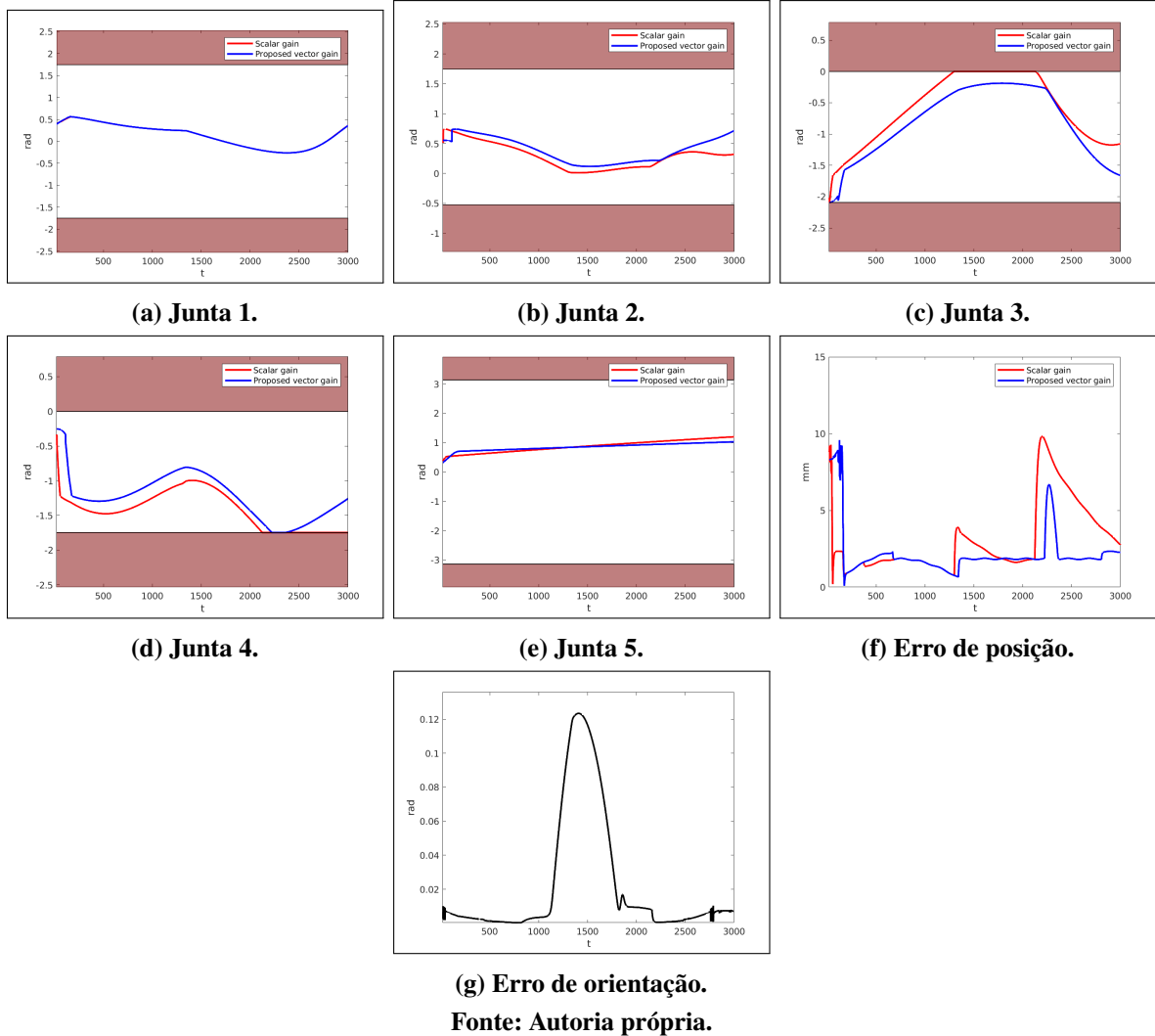
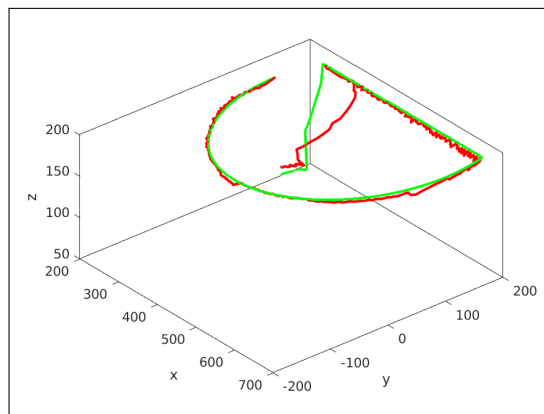
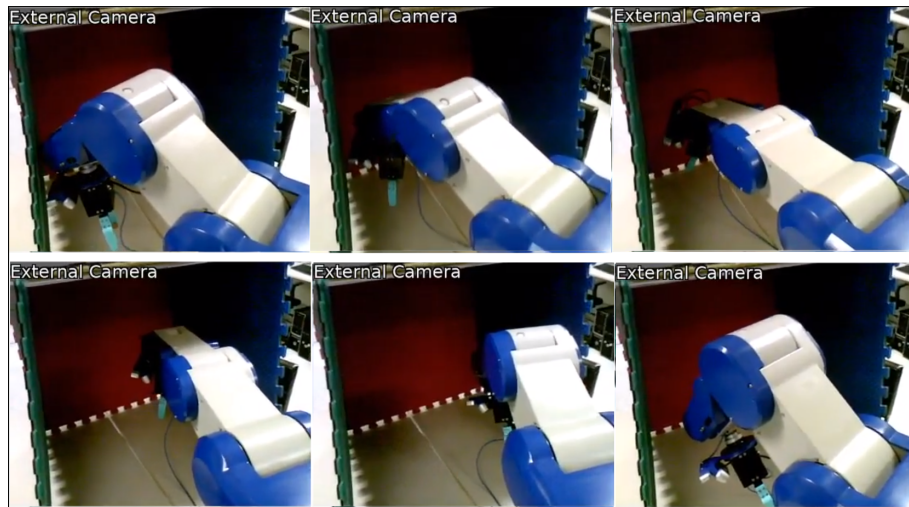


Figura 63 – Trajetória circular medida pelos *encoders*, em vermelho. Trajetória desejada em verde.



Fonte: Autoria própria.

Figura 64 – Momentos da execução da trajetória circular no protótipo real.



Fonte: Autoria própria.

6.3 ANÁLISE DOS RESULTADOS

O objetivo deste capítulo foi construir um sistema de experimentação prática e simulada para demonstrar a aplicação combinada de todos os conceitos discutidos ao longo da dissertação. O Manipulador Movemaster foi utilizado após passar pelo processo de *retrofit*, onde foi possível utilizar seus novos sistemas exteroceptivos para coletar dados do ambiente de trabalho. As informações puderam ser transmitidas através da rede ROS projetada especialmente para a comunicação com o controlador desenvolvido para substituir o originalmente instalado.

O cenário montado na bancada do manipulador permitiu simular um volume de confinamento dentro da região de possível atuação do robô. O sistema foi capaz de coletar os dados de *point cloud* relativos aos limites de confinamento do manipulador e transmiti-los via ROS, onde puderam ser interpretados e processados através da configuração das ferramentas de visão computacional fornecidas pelo Matlab.

Após a etapa de filtragem e classificação das nuvens de pontos, obteve-se novos conjuntos no formato *point cloud*, cada qual pertencente a um dos limites $\pm y$ e $+z$ da área de trabalho representando planos. Estes dados puderam ser convertidos para a representação matemática permitida pela álgebra de Clifford utilizando três pontos aleatoriamente selecionados, os quais determinam um plano unívoco. As informações de localização e orientação dos planos foram publicadas via tópicos ROS de onde puderam ser interpretadas pelo visualizador Rviz. A visualização permite validar a modelagem do ambiente e também monitorar a movimentação do manipulador modelado em arquivo URDF e do efetuador através de marcadores virtuais.

Ao estabelecer trajetórias que propositalmente violam os limites de atuação modelados, foi possível aplicar a álgebra dual-quaterniônica associada à convenção dos Helicóides Sucessivos para executar os cálculos de cinemática direta e inversa. A finalidade desta etapa foi criar um conjunto de valores para as juntas do Movemaster que produzem movimentos no efetuador seguindo a trajetória pretendida. Os cálculos dos valores para o espaço das juntas foram realizados em um *loop* de controle em malha fechada levando em conta o método da cinemática diferencial.

Ao longo do cálculo da cinemática inversa foi aplicada a propriedade da álgebra de Clifford que permite computar a distância perpendicular de pontos a planos. Foi calculada a distância de cada ponto conhecido do manipulador - suas juntas - utilizando as transformações parciais permitida pelos quatérnios duais. Ao identificar o ponto da estrutura que fornece o maior risco de colisão, ou seja, o ponto mais próximo dos planos que modelam a área de trabalho, são aplicadas medidas de correção e prevenção. Caso algum ponto do robô esteja em uma região que causa colisão, realiza-se um movimento na direção diretamente oposta ao plano. Este movimento é aplicado ao manipulador, movendo-o na direção da normal ao plano. A alteração da posição do manipulador imediatamente viola a trajetória previamente intencional, uma vez que a colisão seria inevitável. A posição da junta que causaria a colisão não pode ser controlada diretamente, uma vez que esta é determinada por consequência do controle do efetuador e a relação não é linear. A verificação de colisão é feita até que a alteração na pose do efetuador permita a continuação da trajetória modificada.

Estas ações foram executadas em três cenários: considerando a orientação do efetuador como um valor fixo para os ângulos de *pitch* e *yaw* - o valor de *roll* não pode ser controlado no Movemaster -; desconsiderando a orientação e aplicando a maximização da distância aos limites de junta como objetivo secundário conforme literatura; e aplicando a alteração proposta para variação dos ganhos de objetivo secundário.

Para o primeiro caso, onde a orientação deve seguir valores pré-determinados no decorrer de toda a trajetória, pôde ser observada a eficiência na utilização dos coeficientes dos quatérnios para determinar o erro de orientação. Este método não fornece os ângulos RPY explicitamente, porém tende a 0 quando a orientação é igual à desejada e a 1 quando é exatamente oposta, dadas as características intrínsecas da operação que relaciona dois quatérnios unitários. A utilização deste cálculo de erro é eficiente uma vez que não é relevante saber o valor da orientação ao longo da trajetória, apenas saber que ela tende ao valor desejado.

Quando foi executada a geração das trajetórias com controle de orientação, o algoritmo

realizou a adaptação que evita colisões violando a posição e mantendo a orientação do efetuador. Pode ser notado nas execuções realizadas que isso faz com que o desvio da trajetória original seja maior que nos casos onde a orientação é livre, o que é compatível com o fato de haver dois graus de liberdade a mais sendo controlados no objetivo primário. De qualquer modo, as trajetórias geradas fizeram com que as juntas saíssem dos seus limites físicos em certos pontos do percurso. De fato, isso é inevitável a partir do momento que não há redundância no manipulador para permitir multiplicidade de soluções que mantenham a trajetória desejada. Deste modo, os experimentos com orientação visaram exemplificar a aplicação dos quatérnios unitários neste cenário e validar o algoritmo para estruturas com parâmetros cinemáticos diferentes, generalizando a solução, o que foi cumprido com sucesso.

As trajetórias geradas sem o controle de orientação permitiram validar novamente a prevenção de colisão, além de tornar evidente a diferença de comportamento do manipulador entre utilizar o ganho escalar ou o vetorial proposto para o objetivo secundário. Diversas variáveis podem ser modeladas no formato de um problema de maximização e aplicadas como objetivo secundário, porém como os dados são enviados ao robô real, o objetivo escolhido foi a maximização da distância das juntas até o seu limite físico.

Ao permitir que a orientação varie livremente, dois graus de liberdade foram adicionados à redundância do manipulador. Isso faz com que para uma mesma posição existam múltiplas soluções e o algoritmo converge para a que maximiza o objetivo secundário sem afetar o primário. Ao utilizar um ganho secundário escalar e constante, observou-se que de fato o manipulador executa a trajetória com erro tendendo a zero sem cruzar os limites físicos. O algoritmo proposto envolve utilizar não um escalar, mas sim um vetor com ganhos atrelados a cada junta. Quando a cinemática inversa resulta em um valor inviável na prática para uma junta qualquer, esta é alterada para o seu valor máximo e tem o valor do ganho aumentado em 1%. Os gráficos de execução mostram que esta formulação do problema permite que as juntas fiquem mais distantes dos seus limites tirando proveito da redundância, sem comprometer o perfil de erro de posição. Como pode ser observado, essa alteração não necessariamente impede que as juntas atinjam o seu limite, porém em todos os casos a tendência é de permanecer mais distantes desta condição que usando o ganho escalar. Nota-se também que a redundância afeta a cadeia cinemática apenas nas juntas 2, 3 e 4, o que acaba por consequência afetando a junta 5, porém nada é alterado na junta 1 por estar em um ponto anterior na cadeia. Estes efeitos estão de acordo com o esperado para esta estrutura cinemática, porém teriam um comportamento diferente em outros manipuladores.

O objetivo conjunto principal das diversas trajetórias geradas foi observar o comportamento do algoritmo combinando a cinemática diferencial com a prevenção de colisões utilizando a álgebra de Clifford. Como pode ser observado na seção anterior através dos gráficos apresentados e pelos vídeos capturados durante a experimentação, a modelagem proposta é válida e foi enviada através da rede ROS implementada para o manipulador físico.

Considerou-se diversas formas de envio da trajetória ao manipulador: como os ângulos das juntas são dispostos como um vetor de dados, é possível temporizar o envio destes de forma que combinem com um tempo de execução pré-definido; também é possível enviar uma configuração, monitorar o controlador até que este chegue próximo do desejado, e então enviar a configuração seguinte; ou simplesmente enviar os dados conforme o algoritmo torna-se disponível após o processamento das leituras dos tópicos e armazenamento de dados. Como o controle real possui naturalmente um tempo de resposta, cada uma das formas de envio dos dados produziria um perfil de erro diferente no efetuador físico.

Os experimentos apresentados na seção anterior foram obtidos enviando as informações conforme o processamento dos dados permitiu, ou seja, após a leitura dos dados de monitoramento do ROS e armazenamento das informações para tratamento posterior. Essa forma de envio mostrou-se consistente com as capacidades de movimentação do robô e permitiu observar características de erro do controlador. Caso os pontos da trajetória tivessem sido enviados após o controlador atingir a última configuração enviada, o erro de posição observado seria reduzido, porém isso não permitiria observar exatamente as limitações do controlador atual.

Como pode ser observado pelos dados coletados dos *encoders*, o manipulador é capaz de seguir a trajetória desejada conforme os cálculos de cinemática inversa, porém com erros evidentes em certas configurações. Ao enviar a trajetória, algumas situações não permitiram que o controlador convergisse aos valores desejados antes do comando seguinte ser enviado, o que gerou erros como o observado no início da trajetória circular. Este tipo de erro é dado pela dificuldade de movimentação das juntas contra a força da gravidade. Outro erro encontrado é evidente durante a execução da trajetória retilínea, onde há um desvio significativo da trajetória desejada. Neste caso, isso ocorre quando as junta 2 ou 3 do Movemaster estão com seus freios acionados e em seguida desativados para assumir uma nova posição: a gravidade age na junta que está desativada por estar no seu destino e a desativação do freio faz com que esta caia e o controlador precise de tempo para agir na correção, gerando erro. Optou-se por manter essa forma de execução nos resultados apresentados como forma de demonstrar que o controlador

de fato cumpre seu propósito de manter as juntas em seu destino, porém com limitações na configuração atual, o que permite contribuições futuras como descrito no capítulo 7.

As considerações feitas ao longo desta seção podem ser melhor visualizadas nos vídeos dos experimentos, onde a movimentação torna evidente certas conclusões aqui expostas. De forma geral, a modelagem matemática do ambiente combinada à cinemática em uma notação unificada através da álgebra de Clifford, bem como a aplicação do objetivo secundário, demonstraram-se válidas em ambiente real.

6.4 CONSIDERAÇÕES

Os experimentos realizados e descritos ao longo deste capítulo objetivaram múltiplas análises. Os testes foram formulados de maneira que todos os conceitos e técnicas expostos ao longo deste dissertação pudessem ser aplicados e analisados em um cenário prático.

O foco dos experimentos está nas duas contribuições do projeto: a primeira é gerada pela aplicação da álgebra de Clifford na modelagem do sistema cinemático completo, englobando a cinemática do manipulador e aplicando de maneira inédita a abordagem de pontos e planos na descrição das limitações de sua área de operação. A execução da trajetória no protótipo físico valida novamente o correto funcionamento do controlador implementado, da rede ROS desenvolvida e dos sistema de próprio e exterocepção, não existentes no modelo antes do processo de *retrofit*.

A segunda contribuição é feita pela aplicação do algoritmo que melhora a resposta da maximização do objetivo secundário. Apesar do ajuste de ganhos realizado pelo algoritmo ser de simples implementação, a resposta ao longo da trajetória mostrou que é possível manter as juntas longe de seu limite físico sem impacto no perfil de erros.

7 CONSIDERAÇÕES FINAIS

Ao longo desta dissertação, os conceitos matemáticos foram apresentados de forma que se construiu o raciocínio levando ao entendimento das ferramentas e seu espaço de aplicação em aberto na cinemática de manipuladores. Cada capítulo descreve também resultados parciais obtidos no decorrer do projeto. Os experimentos realizados trouxeram informações relevantes quanto à viabilidade da aplicação hipotetizada, as quais são comentadas neste capítulo.

7.1 CONCLUSÕES

Ao concluir o projeto e processar os dados obtidos, é possível observar em retrospecto e concluir a relevância das etapas intermediárias que culminaram na realização do objetivo geral. O que iniciou como curiosidade em relação à elegância matemática existente na álgebra de Clifford, especialmente no que se refere aos quatérnios duais aplicados na robótica, levou ao estudo das contribuições acadêmicas existentes. No decorrer do capítulo 2, e com comentários relevantes ao longo de todo o documento, foram feitos comentários sobre os trabalhos originais e os publicados recentemente alusivos às áreas relacionadas. A observação da presença literária concludente destas técnicas de modelagem motivou o desenvolvimento da aplicação inédita na representação de confinamento de manipuladores.

De modo a obter resultados práticos que contemplam todas as imperfeições e imprevisibilidades de um modelo real, além da motivação de possuir em laboratório um robô de código aberto, passível de adição de sensoriamento e compatível com o ROS, foi realizada a reforma de dois manipuladores seriais com a modernização de suas capacidades, processo chamado de *retrofit*. Ao longo do capítulo 3, descreveu-se o *retrofit* dos manipuladores ORCA e Movemaster. O intuito de utilizar o ORCA neste projeto foi meramente didático, uma vez que suas características de robô planar permitiram realizar os testes iniciais de técnicas que estavam em desenvolvimento, o que permitiu observar o potencial de continuação do projeto. O ORCA foi direcionado a um trabalho paralelo enquanto o Movemaster tornou-se objeto focal de experimentação devido a sua robustez mecânica e maior complexidade cinemática. A ele foram adicionados sistemas de exerocepção como odometria, cor e profundidade que, junto de seu sistema de propriocepção formado por *encoders* e controles fim-de-curso, foram conectados a um controlador ROS de código aberto.

A aplicação de técnicas avançadas de cinemática de manipuladores se justifica ao contextualizar a metodologia clássica e compreender suas limitações. Este tema foi tratado ao longo do capítulo 4. Foi exposta a notação e aplicação das matrizes de transformação homogênea na solução da cinemática direta. Esta técnica é de simples compreensão e aplicação, porém já é de amplo conhecimento que possui limitações de representação e consumo computacional. Também foram descritas e comparadas duas metodologias para a parametrização de manipuladores, conhecidas como Parâmetros DH e Helicoides Sucessivos. O cálculo da cinemática direta foi aplicado aos dois manipuladores após o processo de *retrofit* para comprovar sua equivalência numérica. Concluiu-se após análise do processo de formulação dos algoritmos, além da análise das informações fornecidas em literatura, que a técnica a ser utilizada no decorrer do projeto seria a convenção dos Helicoides Sucessivos.

Ainda no capítulo 4, descreveu-se a problemática da cinemática inversa diferencial e um sistema de controle de malha fechada para sua solução. Foram explorados os conceitos da matriz Jacobiana e de sua inversão, além da redundância de soluções permitida por seu espaço de mapeamento nulo. A possibilidade de aplicação de objetivos secundários à cinemática inversa foi apresentado junto com o algoritmo de maximização das distâncias das juntas ao seu limite físico. Ao aplicar a cinemática inversa ao longo de trajetórias com o ORCA e o Movemaster, foi observada a possibilidade de realizar uma melhoria no algoritmo existente, tirando proveito da redundância na representação cinemática para propor um algoritmo de variação de ganhos discriminados por junta. O capítulo encerrou com a conclusão de que o algoritmo proposto de fato torna a execução do objetivo secundário melhor do ponto de vista de aumentar a distância até os limites de junta, o que permite maior manipulabilidade do robô. Este algoritmo trabalha de forma diferente do clássico quando a cinemática inversa converge para soluções que violam os limites de junta físicos, alterando o ganho e levando o cálculo outras soluções para o corpo do manipulador sem afetar o perfil de erro do efetuador.

Depois de esclarecer as limitações presentes na computação via MTH e demonstrar as vantagens na representação de orientação utilizando quatérnios unitários, tornou-se clara a possibilidade de expandir suas propriedades algébricas para todo o cálculo da cinemática utilizando os quatérnios duais. Seu uso, já conhecido na comunidade de robótica, foi expandido neste trabalho para a descrição do ambiente do robô utilizando a totalidade da álgebra de Clifford $Cl(0,3,1)$, conhecida como grupo do movimento dos corpos rígidos, da qual os quatérnios duais são um caso específico, denotado por $Cl^+(0,3,1)$ e os quatérnios de Hamilton são outro caso

específico denotado por $C\ell(0,2,0)$. Suas propriedades e método de aplicação na cinemática foram descritos e discutidos ao longo do capítulo 5. Comprovou-se que, assim como extensivamente declarado na literatura relacionada, o uso dos quatérnios duais na cinemática é mais simples computacionalmente e não sofre com limitações matemáticas como as MTH, especialmente no quesito de orientações. Com isso, o algoritmo de cinemática direta e inversa utilizado em projeto foi definido por completo, utilizando a descrição feita por helicoides sucessivos e solucionando as equações com quatérnios duais.

O capítulo 5 seguiu com a demonstração de como a álgebra $C\ell(0,3,1)$ pode ser utilizada para representar objetos como pontos, linhas e planos, além de ser compatível com os mesmos métodos de transformação utilizadas no corpo do robô. Seu uso foi comprovado através de exemplos práticos de forma a evidenciar que estas técnicas podem ser utilizadas na descrição de ambientes que tenham características semelhantes a planos, como um confinamento no formato de cuboide, por exemplo. Como os quatérnios duais permitem o cálculo parcial das transformações da cadeia cinemática, ou seja, obtendo as informações de posição e orientação de cada uma das juntas e não somente do efetuador, foi possível utilizar as relações cunhadas por Selig (2000) para computar a distância mínima entre o corpo do robô e limites físicos do ambiente, desde que estes possam ser modelados dentro da álgebra.

Para unificar os resultados parciais obtidos durante todas as etapas do trabalho e descritas ao longo dos capítulos desta dissertação, um experimento prático foi projetado e analisado ao longo do capítulo 6. O sistema de visão instalado no Movemaster permitiu a coleta de informações do ambiente através de dados tipo *point cloud*, ou nuvem de pontos. A área de confinamento montada para atuação do manipulador é formada por elementos aproximadamente planares, os quais puderam ser medidos pela câmera D-435 e enviados via ROS para processamento no Matlab. Estes dados foram filtrados e convertidos para a notação de planos de Clifford. As informações de posição e orientação dos planos foi publicada no ROS para ser visualizada no Rviz, onde conclui-se que a coleta de poucas informações sobre o ambiente permitiu a sua modelagem completa. Após a definição do ambiente de trabalho, foi possível seguir com o cálculo de uma trajetória livre de colisões.

Para demonstrar a aplicação das relações entre pontos e planos na álgebra de Clifford, três trajetórias foram definidas de forma que sua execução propositalmente causaria colisões com os limites do ambiente. Foram definidas regiões de distância dos planos que nenhuma parte do manipulador pode cruzar por motivos de segurança. Ao executar o algoritmo, a distância de

todos os elementos do robô aos planos de Clifford foi computada ponto a ponto da trajetória e esta foi adaptada utilizando o vetor de norma dos planos para o maior distanciamento.

As trajetórias foram computadas em três cenários cada. Primeiramente foi estabelecida uma orientação constante para o efetuador: neste caso não é possível a aplicação de algoritmos de objetivo secundário e existe a violação dos limites de junta do robô físico. O objetivo deste cenário foi verificar a aplicação completa dos quatérnios duais para controlar simultaneamente posição e orientação do manipulador ao longo da trajetória. O cálculo de erro de orientação utilizado não fornece diretamente os ângulos RPY, porém estes não são necessários pois somente é relevante para o sistema de controle que o erro tenda a zero. Também torna-se desnecessária a utilização de operações custosas computacionalmente e de solução ambígua como a arco-tangente de dois argumentos, $\text{atan2}(x,y)$.

O segundo e terceiro cenário que cada trajetória foi submetida tiveram como objetivo a geração de uma trajetória livre de colisões com o controle de orientação desativado. Neste caso foi possível aplicar o objetivo secundário aproveitando-se da existência de um espaço nulo não-vazio na Jacobiana. Os dois cenários foram usados para comparar o algoritmo de maximização proposto ao longo do capítulo 4. Observou-se que a existência de redundância e multiplicidade na solução da cinemática inversa permite a melhoria do algoritmo clássico. A proposta foi a utilização de um vetor de ganhos onde cada valor está associado a uma das juntas e tem seu valor alterado ao atingir o limite físico. Os resultados coletados mostram que a aplicação do algoritmo proposto gera uma trajetória onde as juntas posicionam-se a uma distância angular maior dos limites em relação ao algoritmo clássico, sem alterar o perfil de erro do efetuador.

As trajetórias geradas foram enviadas ao manipulador através da rede ROS onde se observou, tanto através de um modelo virtual quanto do modelo real, que o ambiente está corretamente modelado e as trajetórias respeitam devidamente as limitações impostas. Foram utilizados marcadores virtuais para representar a trajetória realizada pelo efetuador para comparação com os valores enviados. É possível observar que o robô segue a trajetória desejada porém há limitações quanto a convergência do controlador. Há um atraso na reação a alterações externas bruscas como a ação da gravidade ao desativar os freios, assim como dificuldade de movimentação contra a força da gravidade. Apesar dos erros inseridos por estes pormenores, foi possível validar o correto funcionamento do sistema e das técnicas utilizadas como um todo.

Durante a análise de resultados e geração dos gráficos que demonstram a execução da trajetória, optou-se por suprimir os dados de odometria informados pela câmera T-265. Esta

câmera trabalha com algoritmos de processamento de imagem para determinar variações na sua posição. Os dados coletados em experimento foram precisos quanto à orientação, a qual é dada pela IMU presente na câmera, porém as informações de deslocamento calculadas via SLAM são infrutíferas. Há a hipótese de que a iluminação da área de experimentos ou possíveis variações bruscas causem erros no sistema da câmera, porém a conclusão é de que a câmera de odometria não é um sistema confiável para melhoria do controle de erro no efetuador. Houveram testes parciais onde a odometria seguiu as informações dos *encoders* com erro praticamente desprezível, o que valida a correta construção da árvore de transformações no ROS, porém estes casos foram a exceção e não a regra.

De forma alguma é feita a afirmação de que a exata configuração experimental tenha aplicação direta em um cenário industrial real. O intuito do projeto foi a unificação dos conceitos expostos de forma que seu uso foi comprovado. A contribuição principal desta dissertação é que a hipótese proposta é verdadeira: a álgebra de Clifford é uma ferramenta matemática eficiente para combinar a cinemática de robôs com a representação de ambientes de forma que é possível criar algoritmos claros para a computação de trajetórias e prevenção de colisões. O subproduto do presente projeto é a criação de uma bancada de trabalho com um manipulador serial de código aberto, o qual está disponível para experimentações com manipulação de objetos, visão computacional e qualquer que seja a aplicação compatível que futuros pesquisadores desejarem.

7.2 TRABALHOS FUTUROS

Projetos futuros podem ter uma abordagem direcionada às limitações encontradas neste projeto. A verificação detalhada dos motivos que causam erro na câmera de odometria deve gerar sistemas de correção realimentada da posição do robô: ao ter dados confiáveis de posição real do efetuador, é possível realizar correções em malha fechada para reduzir os erros introduzidos por variáveis espúrias.

Novos projetos de controlador podem abordar a dificuldade de convergência do sistema atual. A troca dos microcontroladores Arduino por sistemas FPGA mais robustos, assim como o encontrado nos trabalhos relacionados, pode melhorar a resposta do sistema. Além da mudança em *hardware*, um projeto onde o controle é o foco principal pode ter como alvo um controlador dinâmico. Um projeto que tenha ganhos adaptativos de forma dinâmica ou que tenha realimentação das forças executadas em cada junta tornaria o sistema todo mais robusto contra erros e com movimentações amortecidas.

Quanto ao algoritmo de adaptação da trajetória, os cenários aqui analisados separam os casos com e sem controle de orientação. Em projetos futuros pode ser viável a elaboração de um modelo híbrido, onde o controle da orientação é mantido quando possível e desativado quando este causaria uma colisão. Esta transição, bem como a transição entre as regiões *critical* e *warning* pode ser feita de forma gradual, diferente da separação brusca utilizada neste trabalho. Para abordagens relacionadas a estes casos, há espaço para aplicação de sistemas de lógica *fuzzy*.

O manipulador está totalmente apto a realizar tarefas que combinam manuseio de objetos com visão computacional. Uma proposta futura é a utilização da álgebra de Clifford para modelar a intersecção do efetuador com objetos detectados via nuvem de pontos, validando uma nova aplicação desta álgebra. Além disso, o robô permite a troca da garra por qualquer outra ferramenta, onde trabalhos futuros envolvendo novas garras com sensoriamento e utilizando outras tecnologias podem ser realizados. De forma geral, a liberdade de projeto permitida com a estrutura atual e seu código aberto tem claro potencial de aplicabilidade acadêmica, tudo ao baixo custo de um manipulador modernizado.

REFERÊNCIAS

AHN, Min Sung; CHAE, Hosik; NOH, Donghun; NAM, Hyunwoo; HONG, Dennis. Analysis and noise modeling of the intel realsense d435 for mobile robots. In: IEEE. **2019 16th International Conference on Ubiquitous Robots (UR)**. [S.l.], 2019. p. 707–711.

ALTMANN, Simon L. **Rotations, quaternions, and double groups**. [S.l.]: Courier Corporation, 2005.

ASPRAGATHOS, Nicholas A; DIMITROS, John K. A comparative study of three methods for robot kinematics. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 28, n. 2, p. 135–145, 1998.

BAYER, Jan; FAIGL, Jan. On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In: IEEE. **2019 European Conference on Mobile Robots (ECMR)**. [S.l.], 2019. p. 1–6.

BAYRO-CORROCHANO, Eduardo; FALCÓN, Luis Eduardo. Geometric algebra of points, lines, planes and spheres for computer vision and robotics. **Robotica**, Cambridge University Press, v. 23, n. 6, p. 755–770, 2005.

BAYRO-CORROCHANO, Eduardo; LASENBY, Joan. Object modelling and motion analysis using clifford algebra. In: XI'AN, CHINA. **Proceedings of Europe-China Workshop on Geometric Modeling and Invariants for Computer Vision**. [S.l.], 1995. v. 143, p. 149.

BAYRO-CORROCHANO, Eduardo; SOMMER, Gerald. Object modelling and collision avoidance using clifford algebra. In: SPRINGER. **International Conference on Computer Analysis of Images and Patterns**. [S.l.], 1995. p. 699–704.

BEKAR, Murat; YAYI, Yusuf. Dual quaternion involutions and anti-involutions. **Advances in Applied Clifford Algebras**, Springer, v. 23, n. 3, p. 577–592, 2013.

BOREN, Jonathan; COUSINS, Steve. Exponential growth of ros [ros topics]. **IEEE Robotics Automation Magazine**, v. 18, n. 1, p. 19–20, 2011.

BREUILS, Stéphane. **Algorithmic structure for geometric algebra operators and application to quadric surfaces**. 2018. Tese (Doutorado) — Université Paris-Est, 2018.

BREUILS, Stephane; TACHIBANA, Kanta; HITZER, Eckhard. New applications of clifford's geometric algebra. **Advances in Applied Clifford Algebras**, Springer, v. 32, n. 2, p. 1–39, 2022.

BUCHHOLZ, Sven; TACHIBANA, Kanta; HITZER, Eckhard. Optimal learning rates for clifford neurons. In: SPRINGER. **International conference on artificial neural networks**. [S.l.], 2007. p. 864–873.

BUSS, Samuel R; KIM, Jin-Su. Selectively damped least squares for inverse kinematics. **Journal of Graphics tools**, Taylor & Francis, v. 10, n. 3, p. 37–49, 2005.

CARDONA, Manuel; GARZÓN-CASTRO, Claudia L; GUTIÉRREZ, Sebastián. Kinematics solution of the rv-3sb robot using successive screws. In: IEEE. **2020 IEEE ANDESCON**. [S.l.], 2020. p. 1–6.

CHEN, Fei; SELVAGGIO, Mario; CALDWELL, Darwin G. Dexterous grasping by manipulability selection for mobile manipulator with visual guidance. **IEEE Transactions on Industrial Informatics**, IEEE, v. 15, n. 2, p. 1202–1210, 2018.

CHEN, Shengyong; LI, Youfu; KWOK, Ngai Ming. Active vision in robotic systems: A survey of recent developments. **The International Journal of Robotics Research**, SAGE Publications Sage UK: London, England, v. 30, n. 11, p. 1343–1377, 2011.

CHENG, Hui; GUPTA, KC. An historical note on finite rotations. 1989.

CHENG, Shili; JI, Ping. Geometric algebra approach to analyzing the singularity of six-dof parallel mechanism. **Journal of Advanced Mechanical Design, Systems, and Manufacturing**, The Japan Society of Mechanical Engineers, v. 13, n. 4, p. JAMDSM0076–JAMDSM0076, 2019.

CHEVALLIER, DP. Lie algebras, modules, dual quaternions and algebraic methods in kinematics. **Mechanism and Machine Theory**, Elsevier, v. 26, n. 6, p. 613–627, 1991.

CHIAVERINI, Stefano; EGELAND, Olav; KANESTROM, Rakel K. Weighted damped least-squares in kinematic control of robotic manipulators. **Advanced robotics**, Taylor & Francis, v. 7, n. 3, p. 201–218, 1992.

CHOI, Dong-Geol; BOK, Yunsu; KIM, Jun-Sik; SHIM, Inwook; KWEON, In So. Structure-from-motion in 3d space using 2d lidars. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 17, n. 2, p. 242, 2017.

CLEMENTS, Guy Roger. Implicit functions defined by equations with vanishing jacobian. **Transactions of the American Mathematical Society**, JSTOR, v. 14, n. 3, p. 325–342, 1913.

CLIFFORD, William Kingdon. Preliminary sketch of biquaternions. **Mathematical Papers**, v. 658, 1873.

CLIFFORD, William Kingdon. Applications of grassmann's extensive algebra. **American Journal of Mathematics**, JSTOR, v. 1, n. 4, p. 350–358, 1878.

CLIFFORD, William Kingdon. On the classification of geometric algebras. **Mathematical Papers**, Macmillan, p. 397–401, 1882.

CONWAY, John H; SMITH, Derek A. **On quaternions and octonions**. [S.l.]: CRC Press, 2003.

CRAIG, John J. **Robótica**. [S.l.]: São Paulo: Pearson Education do Brasil, 2012.

DANIILIDIS, Konstantinos. Hand-eye calibration using dual quaternions. **The International Journal of Robotics Research**, SAGE Publications, v. 18, n. 3, p. 286–298, 1999.

DEDERICK, Louis Serle. On the character of a transformation in the neighborhood of a point where its jacobian vanishes. **Transactions of the American Mathematical Society**, v. 14, n. 1, p. 143–148, 1913.

DENAVIT, Jacques; HARTENBERG, Richard S. A kinematic notation for lower pair mechanisms based on matrices. **Journal of applied mechanics**, v. 77, n. 2, p. 215–221, 1955.

DEO, Arati S; WALKER, Ian D. Overview of damped least-squares methods for inverse kinematics of robot manipulators. **Journal of Intelligent and Robotic Systems**, v. 14, p. 43, 1995.

DEWITT, Samuel J.; KETCHERSIDE, Ariel; MCQUEENY, Tim M.; DUNLOP, Joseph P.; FILBEY, Francesca M. The hyper-sentient addict: an exteroception model of addiction. **The American Journal of Drug and Alcohol Abuse**, Taylor & Francis, v. 41, n. 5, p. 374–381, 2015. PMID: 26154169. Disponível em: <https://doi.org/10.3109/00952990.2015.1049701>.

DING, Zhenjie; LI, Yuqiu; ZHANG, Zhengxing. Electric hybrid control method of assembly line robot based on plc. **Thermal Science**, v. 24, n. 3 Part A, p. 1505–1511, 2020.

DOOLEY, JR; MCCARTHY, J Michael. On the geometric analysis of optimum trajectories for cooperating robots using dual quaternion coordinates. In: IEEE. **[1993] Proceedings IEEE International Conference on Robotics and Automation**. [S.l.], 1993. p. 1031–1036.

DORAN, Chris; GULLANS, Steven R; LASENBY, Anthony; LASENBY, Joan; FITZGERALD, William. **Geometric algebra for physicists**. [S.l.]: Cambridge University Press, 2003.

DRAY, Tevian. The hodge dual operator. **Oregon State University report**, p. 1–6, 1999.

DUONG, Trung H; JAKSIC, Nebojsa I. Let's not throw away that big and bulky manipulator—revitalize it! In: **2018 ASEE Annual Conference & Exposition**. [S.l.: s.n.], 2018.

EL-LAITHY, Riyad A; HUANG, Jidong; YEH, Michael. Study on the use of microsoft kinect for robotics applications. In: IEEE. **Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium**. [S.l.], 2012. p. 1280–1288.

ELL, Todd A; SANGWINE, Stephen J. Quaternion involutions and anti-involutions. **Computers & Mathematics with Applications**, Elsevier, v. 53, n. 1, p. 137–143, 2007.

FAMILTON, Johannes C. Quaternions: A history of complex noncommutative rotation groups in theoretical physics. **arXiv preprint arXiv:1504.04885**, 2015.

FRANCHINI, Silvia; VASSALLO, Giorgio; SORBELLO, F. A brief introduction to clifford algebra. 01 2010.

FUNDA, Janez; PAUL, Richard P. A computational analysis of screw transformations in robotics. **IEEE Transactions on robotics and automation**, IEEE, v. 6, n. 3, p. 348–356, 1990.

GAN, Dongming; LIAO, Qizheng; WEI, Shimin; DAI, JS; QIAO, S. Dual quaternion-based inverse kinematics of the general spatial 7r mechanism. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 222, n. 8, p. 1593–1598, 2008.

GE, Q.J.; MCCARTHY, J.M. An algebraic formulation of configuration-space obstacles for spatial robots. In: **Proceedings., IEEE International Conference on Robotics and Automation**. [S.l.: s.n.], 1990. p. 1542–1547 vol.3.

GOUASMI, Mahmoud. Robot kinematics, using dual quaternions. **IAES International Journal of Robotics and Automation**, IAES Institute of Advanced Engineering and Science, v. 1, n. 1, p. 13, 2012.

GRUNNET-JEPSEN, Anders; HARVILLE, Michael; FULKERSON, Brian; PIRO, Daniel; BROOK, Shirit; RADFORD, Jim. **Introduction to Intel® RealSense™ Visual SLAM and the T265 Tracking Camera**. 2019. <https://dev.intelrealsense.com/docs/intel-realsensetm-visual-slam-and-the-t265-tracking-camera>.

HAMILTON, William Rowan. On a new species of imaginary quantities, connected with the theory of quaternions. **Proceedings of the Royal Irish Academy (1836-1869)**, JSTOR, v. 2, p. 424–434, 1843.

HAMILTON, William Rowan. On quaternions. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, Taylor & Francis, v. 25, n. 163, p. 10–13, 1844.

HAMILTON, Wm R. Theory of quaternions. **Proceedings of the Royal Irish Academy (1836-1869)**, JSTOR, v. 3, p. 1–16, 1844.

HAMILTON, William Rowan. **Elements of quaternions**. [S.l.]: Longmans, Green, & Company, 1866.

HART, John C; FRANCIS, George K; KAUFFMAN, Louis H. Visualizing quaternion rotation. **ACM Transactions on Graphics (TOG)**, ACM New York, NY, USA, v. 13, n. 3, p. 256–276, 1994.

HAZARIKA, Shyamanta M; DIXIT, Uday Shanker. Robotics: History, trends, and future directions. In: **Introduction to Mechanical Engineering**. [S.l.]: Springer, 2018. p. 213–239.

HIRANI, Anil N; KALYANARAMAN, Kaushik; VANDERZEE, Evan B. Delaunay hodge star. **Computer-Aided Design**, Elsevier, v. 45, n. 2, p. 540–544, 2013.

HITZER, Eckhard. Geometric operations implemented by conformal geometric algebra neural nodes. **arXiv preprint arXiv:1306.1358**, 2013.

HITZER, Eckhard; NITTA, Tohru; KUROE, Yasuaki. Applications of clifford's geometric algebra. **Advances in Applied Clifford Algebras**, Springer, v. 23, n. 2, p. 377–404, 2013.

HODGE, William Vallance Douglas; PEDOE, Daniel. **Methods of algebraic geometry**. [S.l.]: CUP Archive, 1947. v. 1.

HUANG, Shih-Jer; LIU, Hou-Der; HUANG, CK; TING, Chun-Chien; CHEN, Chun-Hoa. Hybrid force/position control of master and slave robotic systems. **Journal of the Chinese Institute of Engineers**, Taylor & Francis, v. 43, n. 8, p. 713–724, 2020.

HUANG, Shih-Jer; WU, Shian-Shin. Vision-based robotic motion control for non-autonomous environment. **Journal of Intelligent and Robotic Systems**, Springer, v. 54, n. 5, p. 733–754, 2009.

HUO, Xinming; SUN, Tao; SONG, Yimin; QI, Yang; WANG, Panfeng. An analytical approach to determine motions/constraints of serial kinematic chains based on clifford algebra. **Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science**, SAGE Publications Sage UK: London, England, v. 231, n. 7, p. 1324–1338, 2017.

IFR - International Federation of Robotics. **Robot Sales Rise Again**. 2021. Site da IFR. Disponível em: <https://ifr.org/ifr-press-releases/news/robot-sales-rise-again>.

INFINEON. **BTS 7960 - High Current PN Half Bridge**. [S.l.], 2004. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/152657/INFINEON/BTS7960.html>.

Intel Corporation. **Depth Camera D435**. 2022. Site da Intel Realsense. Disponível em: <https://www.intelrealsense.com/depth-camera-d435/>.

Intel Corporation. **Tracking Camera T265**. 2022. Site da Intel Realsense. Disponível em: <https://www.intelrealsense.com/tracking-camera-t265/>.

JAKSIC, Nebojsa I; DUONG, Trung H. Teaching an old robot new tricks. In: **2018 ASEE Zone IV Conference**. [S.l.: s.n.], 2018.

JAVOID, Mohd; HALEEM, Abid; VAISH, Abhishek; VAISHYA, Raju; IYENGAR, Karthikeyan P. Robotics applications in covid-19: A review. **Journal of Industrial Integration and Management**, World Scientific, v. 5, n. 04, p. 441–451, 2020.

JESUS, Raphael CO; MOLINA, Lucas; CARVALHO, Elyson AN; FREIRE, Eduardo O. Singularity-free inverse kinematics with joint prioritization for manipulators. **Journal of Control, Automation and Electrical Systems**, Springer, p. 1–10, 2022.

JOHANSEN, Tor-Aleksander; SÁNCHEZ, José J; KRISTIANSEN, Raymond. Dual quaternion control: a review of recent results within motion control. *Nonlinear Studies*, 2019.

KENWRIGHT, Ben. A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3d character hierarchies. Citeseer, 2012.

KENWRIGTH, B. Inverse kinematics with dual-quaternions, exponential-maps, and joint limits. **International Journal on Advances in Intelligent Systems**, Citeseer, v. 6, n. 1, 2013.

KUNG, Ying-Shieh; CHEN, Chia-Sheng. Fpga-realization of a motion control ic for robot manipulator. **Robot Manipulators**, IntechOpen, 2008.

LASENBY, Joan; FITZGERALD, William J; LASENBY, Anthony N; DORAN, CJL. New geometric methods for computer vision: An application to structure and motion estimation. **International Journal of Computer Vision**, Springer, v. 26, n. 3, p. 191–213, 1998.

LI, Yong-Bin; LI, Tie-Jun; ZHU, Hui-Fang; YANG, Dong; CHEN, Ya-Jun; ZHANG, Wen-Ming; LIU, Jian-Ming; ZHANG, Tian. Comparative analysis of the kinematics solution based on the dh method and screw theory. **Mathematical Problems in Engineering**, Hindawi, v. 2021, 2021.

LINDHAGEN, Elin; NYGREN, Peter; LARSSON, Rolf. The fluorometric microculture cytotoxicity assay. **Nature protocols**, Nature Publishing Group, v. 3, n. 8, p. 1364–1369, 2008.

LIU, Diyi; ARAI, Shogo; XU, Yajun; TOKUDA, Fuyuki; KOSUGE, Kazuhiro. 6d pose estimation of occlusion-free objects for robotic bin-picking using ppf-meam with 2d images (occlusion-free ppf-meam). **IEEE Access**, IEEE, v. 9, p. 50857–50871, 2021.

LYMAN, Gregory; WILCOX, Jeffery; SANFORD, Rowdy. Robotics retrofit: Renovating outdated robotics platforms to meet current curriculum requirements driven by industry demand. In: **2019 CIEC**. [S.l.: s.n.], 2019.

LYNCH, Kevin M; PARK, Frank C. **Modern Robotics**. [S.l.]: Cambridge University Press, 2017.

MA, Jiayi; CHEN, Qiaohong; YAO, Huijing; CHAI, Xinxue; LI, Qinchuan. Singularity analysis of the 3/6 stewart parallel manipulator using geometric algebra. **Mathematical Methods in the Applied Sciences**, Wiley Online Library, v. 41, n. 6, p. 2494–2506, 2018.

MedWOW. **Manufacturer Specifications - ORCA, Beckman Coulter**. 2022. Plataforma online de equipamentos médicos. Disponível em: <http://www.medwow.com/med/chemistry-analyzers-automated/beckman-coulter/orca/43900.model-spec>.

Mitsubishi Electric Corporation. **Industrial Micro-Robot System Model RV-M2 Instruction Manual**. [S.l.], 1990.

MOHAMED, Mamdouh S; HIRANI, Anil N; SAMTANEY, Ravi. Comparison of discrete hodge star operators for surfaces. **Computer-Aided Design**, Elsevier, v. 78, p. 118–125, 2016.

MONTGOMERY-SMITH, Stephen; SHY, Cecil. An introduction to using dual quaternions to study kinematics. **arXiv preprint arXiv:2203.13653**, 2022.

MONTOYA, Francisco G; BAÑOS, Raúl; ALCAYDE, Alfredo; ARRABAL-CAMPOS, Francisco Manuel; ROLDÁN-PÉREZ, Javier. Vector geometric algebra in power systems: An updated formulation of apparent power under non-sinusoidal conditions. **Mathematics**, Multidisciplinary Digital Publishing Institute, v. 9, n. 11, p. 1295, 2021.

MONTOYA, Francisco G; EID, Ahmad H. Formulating the geometric foundation of clarke, park, and fbd transformations by means of clifford's geometric algebra. **Mathematical Methods in the Applied Sciences**, Wiley Online Library, 2021.

NAKAMURA, Yoshihiko; HANAFUSA, Hideo. Inverse kinematic solutions with singularity robustness for robot manipulator control. 1986.

OLIVEIRA, André Schneider de. **Análise cinemática via quatérnios duais aplicada a um sistema veículo-manipulador subaquático**. 2011. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2011.

Open Robotics. **ROS - Robot Operating System**. 2021. <https://www.ros.org/>.

OSTANIN, Mikhail; MIKHEL, Stanislav; EVLAMPIEV, Alexey; SKVORTSOVA, Valeria; KLIMCHIK, Alexandr. Human-robot interaction for robotic manipulator programming in mixed reality. In: IEEE. **2020 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2020. p. 2805–2811.

PARK, Chanhun; PARK, Kyoungtaik. Design and kinematics analysis of dual arm robot manipulator for precision assembly. In: IEEE. **2008 6th IEEE International Conference on Industrial Informatics**. [S.l.], 2008. p. 430–435.

PEREZ, Alba; MCCARTHY, J. M. Dual Quaternion Synthesis of Constrained Robotic Systems . **Journal of Mechanical Design**, v. 126, n. 3, p. 425–435, 10 2003. ISSN 1050-0472. Disponível em: <https://doi.org/10.1115/1.1737378>.

RADAVELLI, Luiz Alberto; PIERI, Edson Roberto De; MARTINS, Daniel; SIMONI, Roberto. Points, lines, screws and planes in dual quaternions kinematics. In: **Advances in robot kinematics**. [S.l.]: Springer, 2014. p. 285–293.

RADAVELLI, Luiz Alberto; PIERI, Edson Roberto De; MARTINS, Daniel; SIMONI, Roberto. A screw dual quaternion operator for serial robot kinematics. **Proceedings of PACAM XIV, Santiago**, p. 27–38, 2014.

RAKPRAYOON, Panjawee; RUCHANURUCKS, Miti; COUNDOUL, Ada. Kinect-based obstacle detection for manipulator. In: IEEE. **2011 IEEE/SICE International Symposium on System Integration (SII)**. [S.l.], 2011. p. 68–73.

RIBEIRO, Luiz; GUENTHER, Raul; MARTINS, Daniel. Screw-based relative jacobian for manipulators cooperating in a task. In: ABCM. **ABCM Symposium Series in Mechatronics**. [S.l.], 2008. v. 3, p. 276–285.

ROONEY, Joe. On the three types of complex number and planar transformations. **Environment and Planning B: Planning and Design**, SAGE Publications Sage UK: London, England, v. 5, n. 1, p. 89–99, 1978.

RUIZ, Elvis; ACUÑA, Raúl; CERTAD, Novel; TERRONES, Angel; CABRERA, María Eugenia. Development of a control platform for the mobile robot roomba using ros and a kinect sensor. In: **2013 Latin American Robotics Symposium and Competition**. [S.l.: s.n.], 2013. p. 55–60.

SANGWINE, Stephen J; HITZER, Eckhard. Clifford multivector toolbox (for matlab). **Advances in Applied Clifford Algebras**, Springer, v. 27, n. 1, p. 539–558, 2017.

SANGWINE, Stephen J; HITZER, Eckhard. **Clifford Multivector Toolbox**. 2022. Sourceforge. Disponível em: <https://clifford-multivector-toolbox.sourceforge.io/>.

SCIAVICCO, Lorenzo; SICILIANO, Bruno. **Modelling and control of robot manipulators**. [S.l.]: Springer Science & Business Media, 2012.

SELIG, Jon M. Clifford algebra of points, lines and planes. **Robotica**, v. 18, p. 545 – 556, 2000.

SELIG, Jon M. **Geometric fundamentals of robotics**. [S.l.]: Springer Science & Business Media, 2005.

SPONG, Mark W; HUTCHINSON, Seth; VIDYASAGAR, Mathukumalli. **Robot modeling and control**. [S.l.]: John Wiley & Sons, 2020.

ST MICROELETRONICS. **Automotive fully integrated H-bridge motor driver**. [S.l.], 2017. Disponível em: <https://www.st.com/en/automotive-analog-and-power/vnh3sp30-e.html>.

TADIC, Vladimir; ODRY, Akos; KECSKES, Istvan; BURKUS, Ervin; KIRALY, Zoltan; ODRY, Peter. Application of intel realsense cameras for depth image generation in robotics. **WSEAS Transac. Comput**, v. 18, p. 2224–2872, 2019.

TARHASAARI, Timo; KETTUNEN, Lauri; BOSSAVIT, Alain. Some realizations of a discrete hodge operator: a reinterpretation of finite element techniques [for em field analysis]. **IEEE Transactions on magnetics**, IEEE, v. 35, n. 3, p. 1494–1497, 1999.

The MathWorks, Inc. **Computer Vision Toolbox**. 2022. <https://www.mathworks.com/help/vision/index.html>.

THIRUVENGADAM, Sudharsan; MILLER, Karol. A geometric algebra based higher dimensional approximation method for statics and kinematics of robotic manipulators. **Advances in Applied Clifford Algebras**, Springer, v. 30, n. 1, p. 1–43, 2020.

THIRUVENGADAM, Sudharsan; TAN, Jei Shian; MILLER, Karol. A generalised quaternion and clifford algebra based mathematical methodology to effect multi-stage reassembling transformations in parallel robots. **Advances in Applied Clifford Algebras**, Springer, v. 31, n. 3, p. 1–69, 2021.

TSAI, Lung-Wen. **Robot analysis: the mechanics of serial and parallel manipulators**. [S.l.]: John Wiley & Sons, 1999.

WALUŚ, Przemysław. **Modelo virtual do Mitsubishi MOVEMASTER RV-M1**. 2018. Repositório virtual do Grabcad. Disponível em: <https://grabcad.com/library/mitsubishi-movemaster-rv-m1-1>.

WANG, Xingyu; WANG, Anna; WANG, Dazhi; LIU, Zhen; QI, Yufei. A novel trajectory tracking control with modified supertwisting sliding mode for human-robot cooperation manipulator in assembly line. **Journal of Sensors**, Hindawi, v. 2022, 2022.

WONG, Ching-Chang; FENG, Hsuan Ming; LAI, Yu-Cheng; CHEN, Hsiang-Yun. Manipulator system designs for drawing random objects through point cloud posture estimation. **Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture**, SAGE Publications Sage UK: London, England, p. 0954405420979461, 2021.

YANG, Guang-Zhong; BELLINGHAM, Jim; DUPONT, Pierre E; FISCHER, Peer; FLORIDI, Luciano; FULL, Robert; JACOBSTEIN, Neil; KUMAR, Vijay; MCNUTT, Marcia; MERRIFIELD, Robert et al. The grand challenges of science robotics. **Science robotics**, American Association for the Advancement of Science, v. 3, n. 14, p. eaar7650, 2018.

ZAPLANA, Isiah; HADFIELD, Hugo; LASENBY, Joan. **Singularities of serial robots: Identification and distance computation using geometric algebra**. arXiv, 2021. Disponível em: <https://arxiv.org/abs/2109.12470>.

ÖZGÜR, Erol; MEZOUAR, Youcef. Kinematic modeling and control of a robot arm using unit dual quaternions. **Robotics and Autonomous Systems**, v. 77, p. 66–73, 2016. ISSN 0921-8890. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0921889015301184>.

APÊNDICES

APÊNDICE A – MATRIZES DE TRANSFORMAÇÃO HOMOGÊNEA

As Matrizes de Transformação Homogênea (MTH) são uma ferramenta para cálculos em cinemática amplamente conhecida e analisada em diversos trabalhos de referência na área, como em Sciavicco e Siciliano (2012), Tsai (1999) e Selig (2005), devido à sua simplicidade de compreensão e implementação. As relações e notações descritas ao longo deste apêndice são dadas de acordo com estas referências bibliográficas. Para descrever a estrutura e função de uma MTH, primeiramente é necessário definir o que é um ponto no sentido de ser um elemento geométrico e o que são as operações de rotação e translação.

Usando a representação conforme Sciavicco e Siciliano (2012), um ponto no espaço é o objeto geométrico mais elementar, possuindo três graus de liberdade em um sistema tridimensional, sendo definido por um vetor de três elementos $p = [x, y, z]$. Com um ponto é possível realizar a operação de translação no espaço, a qual consiste em realizar um deslocamento na posição deste ponto simplesmente pela adição de um vetor de translação também contendo três elementos, definido como $d = [x, y, z]$. Para realizar um movimento de translação, basta realizar a adição entre o vetor representando o ponto e o vetor representando a translação, conforme equação 38.

$$p' = p + d \quad (38)$$

A operação de rotação é o movimento de um elemento de forma angular a um eixo de referência. Esta operação é definida levando em conta a projeção do objeto nos eixos de referência em relação à origem de seu sistema, considerando a distância l da origem. Esta operação é representada pela multiplicação do ponto inicial pela representação da rotação, dada conforme equações 39 e 40.

$$p = \begin{cases} p_x = l \cos(\theta) \\ p_y = l \sin(\theta) \end{cases} \quad (39)$$

$$p' = Rp \quad (40)$$

Para generalizar a notação de uma matriz de rotação R que descreve o movimento em torno de um eixo, partindo do *frame* de referência i até o *frame* de referência $i + 1$, utiliza-se as matrizes descritas nas equações 41, 42 e 43, as quais descrevem uma rotação de magnitude θ em torno dos eixos x , y e z , respectivamente.

$$R_x(\theta)_{(i+1)}^i \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\text{sen}(\theta) \\ 0 & \text{sen}(\theta) & \cos(\theta) \end{bmatrix} \quad (41)$$

$$R_y(\theta)_{(i+1)}^i \begin{bmatrix} \cos(\theta) & 0 & \text{sen}(\theta) \\ 0 & 1 & 0 \\ -\text{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (42)$$

$$R_z(\theta)_{(i+1)}^i \begin{bmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (43)$$

Como explicado por Tsai (1999), para representar um corpo rígido de forma genérica são necessários seis parâmetros. O espaço tridimensional possui seis graus de liberdade, sendo três para a representação de posição e três para a orientação de um objeto. Dessa forma, um objeto em movimento no espaço não realiza transformações elementares mas sim uma composição de transformações consecutivas envolvendo rotações e translações simultaneamente. Para combinar o efeito de n consecutivas rotações no espaço, basta multiplicar as matrizes de rotação, conforme equação 44.

$$R(\theta)_n^o = R(\theta)_1^o R(\theta)_2^1 R(\theta)_3^2 \dots R(\theta)_n^{n-1} \quad (44)$$

De maneira semelhante, a combinação consecutiva de n translações no espaço pode ser descrita através da soma do efeito das translações elementares, conforme equação 45.

$$d_n^o = d_1 + d_2 + d_3 + \dots + d_n \quad (45)$$

Como o deslocamento generalizado de um corpo rígido é composto de rotações e translações simultâneas, um deslocamento acoplado pode ser descrito como:

$$p' = R(\theta)_n^o p + d_n^o \quad (46)$$

Como a equação 46 não descreve as transformações de uma maneira compacta, é possível introduzir a aplicação das transformações homogêneas. Conforme Tsai (1999), a coordenada homogênea generaliza a representação de um vetor tridimensional inserindo um quarto elemento de escala não-nulo ρ , de forma que:

$$p = [\rho p_x, \rho p_y, \rho p_z, \rho]^T \quad (47)$$

Para a aplicação específica na cinemática, define-se que $\rho = 1$, uma vez que a representação matemática não deve introduzir escalas quando representando modelos reais. Dessa forma, a representação de um ponto no espaço em coordenadas homogêneas é dada conforme a equação 48.

$$p = [p_x, p_y, p_z, 1]^T \quad (48)$$

A representação no espaço homogêneo permite, então, a descrição da Matriz de Transformação Homogênea. A MTH é uma matriz 4×4 que contém a representação de elementos de rotação e translação simultaneamente, além de um vetor de perspectiva e uma escala. Esta matriz é, portanto, composta por submatrizes sendo uma matriz $R_{(3 \times 3)}$ para a rotação de um corpo rígido, um vetor $d_{(3 \times 1)}$ para representar a translação, um vetor $\gamma_{(1 \times 3)}$ para descrever a perspectiva e um escalar $\rho_{(1 \times 1)}$ que descreve a escala de transformação. Sciavicco e Siciliano (2012) também esclarecem que para aplicações em robótica, não há nenhuma distorção de perspectiva entre as transformações, bem como a escala do objeto não é alterada, portanto $\gamma = 0$ e $\rho = 1$. A estrutura da MTH é dada conforme equação 49.

$$H_{(i+1)}^i \begin{bmatrix} R_{(i+n)(3 \times 3)}^i & \vdots & d_{(i+n)(3 \times 1)}^i \\ \dots & \ddots & \dots \\ 0_{(1 \times 3)} & \vdots & 1_{(1 \times 1)} \end{bmatrix} \quad (49)$$

Utilizando a representação de um ponto em coordenadas homogêneas conforme equação 48, uma transformação homogênea é descrita conforme equação 50.

$$p' = Hp \quad (50)$$

E a concatenação de n transformações é dada conforme equação 51.

$$p^{(i+n)} = H_{(i+1)}^i H_{(i+2)}^{(i+1)} H_{(i+3)}^{(i+2)} \dots H_{(i+n)}^{(i+n-1)} p^i \quad (51)$$

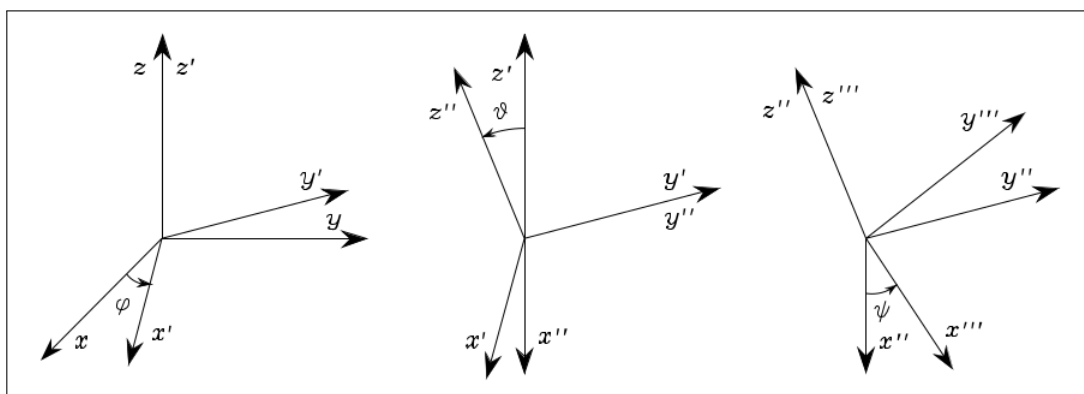
A utilização das MTH para a concatenação de transformações é a solução clássica comumente empregada. Apesar de sua simplicidade de compreensão e implementação, ela traz problemas de otimização computacional decorrentes da quantidade de operações necessárias para a realização dos cálculos, além do fato de utilizar 16 elementos para descrever estruturas que

podem ter até 6 graus de liberdade. Esta metodologia também introduz ao sistema singularidades de representação como a conhecida *Gimbal's Lock*, a qual degenera momentaneamente os cálculos de cinemática. Todos estes conceitos são discutidos em detalhes na tese de Oliveira (2011), e a existência de tais problemas motivam a introdução de ferramentas matemáticas mais avançadas e apropriadas para a cinemática, onde existem aplicações diretas da Álgebra de Clifford. Estes conceitos são abordados no apêndice F.

APÊNDICE B – REPRESENTAÇÃO MÍNIMA DE ORIENTAÇÃO

Uma maneira de representar a orientação de um corpo rígido no espaço cartesiano é através do uso dos ângulos de Euler (SPONG et al., 2020). Esta notação define um conjunto de três ângulos independentes que estão relacionados aos três eixos de rotação. Como ilustrado na figura 65, os ângulos de Euler fornecem a orientação de um corpo rígido em relação a um sistema de referência, porém a combinação das rotações pode ser feita de diversas formas diferentes e de forma sucessiva. Para exemplificar uma das combinações de ângulos de Euler, é apresentada a rotação composta pelos ângulos de Euler **ZYZ**, ou seja, uma rotação em torno do eixo Z , seguida por uma rotação em Y , seguida por outra rotação em Z .

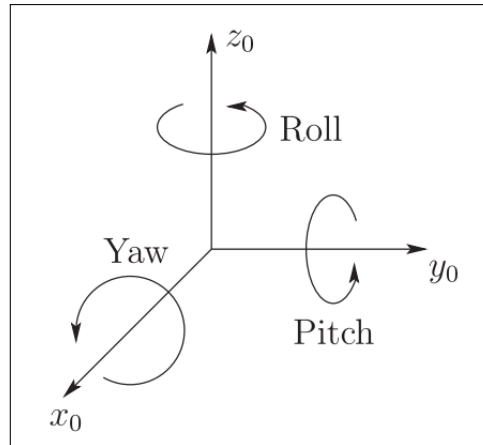
Figura 65 – Rotação composta pelos ângulos de Euler ZYZ.



Fonte: Sciavicco e Siciliano (2012)

A rotação composta de ângulos de Euler denominada **ZYX** tem aplicação comum em robótica e sistemas aeronáuticos, recebendo a nomenclatura de *ângulos Roll-Pitch-Yaw*, ou simplesmente RPY (SCIATICCO; SICILIANO, 2012). A figura 66 exemplifica a rotação via ângulos RPY. Os autores salientam que as rotações definidas pelos ângulos de Euler são realizadas de forma independente: a segunda rotação é realizada sobre o sistema resultante da primeira, assim como a terceira é feita sobre o sistema resultante da segunda.

Nos cálculos de cinemática envolvendo transformações descritas através das MTH, é possível extrair os ângulos RPY a partir da matriz de rotação R . Considerando uma matriz de rotação genérica como na equação 52, os ângulos Roll (φ), Pitch (ϑ) e Yaw (ψ) podem ser extraídos conforme equações 53, 54 e 55. Esta representação fornece o valor de ϑ no intervalo $(-\pi/2, \pi/2)$.

Figura 66 – Representação dos ângulos Roll-Pitch-Yaw.

Fonte: Spong et al. (2020)

$$R = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (52)$$

$$\varphi = \text{atan2}(a_{21}, a_{11}) \quad (53)$$

$$\vartheta = \text{atan2}(-a_{31}, \sqrt{a_{32}^2 + a_{33}^2}) \quad (54)$$

$$\psi = \text{atan2}(a_{32}, a_{33}) \quad (55)$$

A extração dos ângulos RPY a partir da matriz de rotação exige a aplicação da função $\text{atan2}(y, x)$. Esta função possui descontinuidades matemáticas que introduzem problemas numéricos e soluções não-válidas. Estes problemas são contornados com o uso da representação por quatérnios unitários, que é discutida no apêndice F.

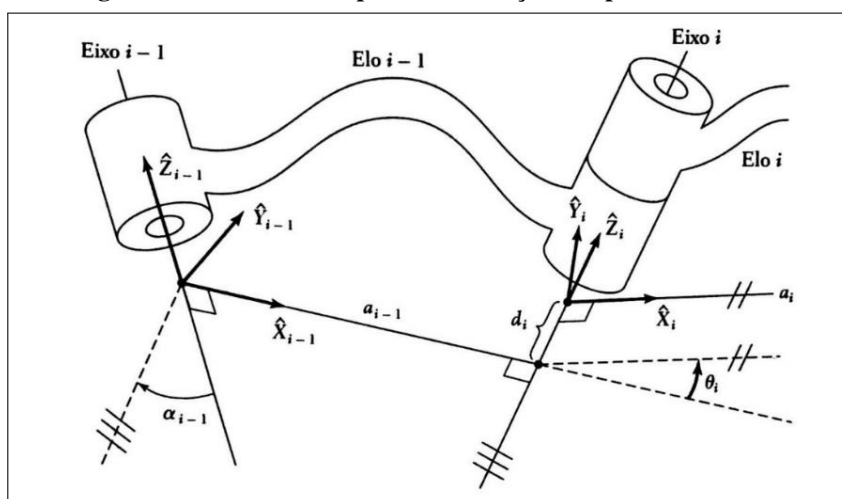
Uma métrica relevante na cinemática é a diferença entre duas orientações, a qual também pode ser compreendida como o erro existente entre a orientação atual e a desejada. Para computar a diferença e_O entre a orientação atual ϕ e a orientação desejada ϕ_d , ambas dadas em ângulos de Euler, basta realizar a subtração dos dois vetores conforme equação 56 (SCIATICCO; SICILIANO, 2012).

$$e_O = \phi_d - \phi \quad (56)$$

APÊNDICE C – PARÂMETROS DENAVIT-HARTENBERG

Denavit e Hartenberg (1955) desenvolveram um método para descrição de estruturas robóticas baseado em matrizes. Em seu trabalho, é apresentada uma notação que envolve a atribuição de quatro parâmetros para cada par junta-elo, a qual pode ser aplicada em qualquer estrutura de corpos rígidos. Segundo os próprios autores, a notação através dos então denominados parâmetros DH torna possível descrever completamente manipuladores robóticos, sendo uma ferramenta confiável para procedimentos analíticos. Para melhor visualização, Craig (2012) apresenta através da Figura 67, um eixo torcido com sua forma exagerada para exemplificar os quatro parâmetros DH de forma visual, também fornecendo uma explicação sobre cada variável descrita.

Figura 67 – Eixo torcido para visualização dos parâmetros DH.



Fonte: Craig (2012).

Na Figura 67, observa-se que são estabelecidos sistemas cartesianos de referência $(X, Y, Z)_{i-1}$ e $(X, Y, Z)_i$ de maneira a estabelecer orientações relativas entre os eixos inicial e final do elo i . Logo, definem-se os parâmetros α_i , a_i , θ_i e d_i :

- α_i representa o ângulo formado entre Z_{i-1} e Z_i em torno do eixo X_i ;
- a_i representa a distância entre Z_{i-1} e Z_i ao longo do eixo X_i ;
- θ_i representa o ângulo formado entre X_{i-1} e X_i em torno do eixo Z_i ;
- d_i representa a distância entre X_{i-1} e X_i em torno do eixo Z_i ;

Os parâmetros DH são definidos seguindo uma ordem de transformação convencional. Tal ordem visa transferir a origem de um sistema cartesiano para o seguinte, rotacionando e transladando seu *frame* de referência, de modo que o mesmo coincida com o *frame* da junta seguinte. A ordem de aplicação dos parâmetros DH deve ser: primeiramente uma rotação em α , seguida de uma translação em a , uma rotação em θ e uma translação em d .

Como estes parâmetros devem ser definidos para cada par junta-elo, constrói-se uma tabela com as informações para cada grau de liberdade da estrutura robótica. A notação funciona também para juntas mais complexas como, por exemplo, uma junta esférica com 3 graus de liberdade. Para a descrição cinemática de uma estrutura assim, pode se considerar que uma junta com n graus de liberdade é o equivalente a n juntas de 1 grau de liberdade conectadas por $n - 1$ elos de comprimento zero (KENWRIGHT, 2012).

A transformação do sistemas de coordenadas de uma junta na seguinte usando a convenção Denavit-Hartenberg é feita em uma ordem específica, conforme equação 57, onde é combinada uma sequência de rotações e translações puras conforme definidas nas equações 38, 41, 42 e 43, presentes no apêndice A. A equação 57 representa a transformação de cada junta, portanto a concatenação de transformações a partir de um referencial na base do manipulador resulta na posição do efetuador.

$$A_n^{n-1} = R_z(\theta_n)T_z(d_n)T_x(a_n)R_x(\alpha_n) \quad (57)$$

De acordo com a notação de Craig (2012), a MTH entre dois sistemas de coordenadas pode ser escrita de forma generalizada em termos dos parâmetros DH como na equação 58.

$$H_n^{n-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_{i-1} \\ \sin(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\cos(\alpha_i) & -\sin(\alpha_i) & -\sin(\alpha_i)d_i \\ \sin(\theta_i)\sin(\alpha_i) & \cos(\theta_i)\sin(\alpha_i) & \cos(\alpha_i) & -\cos(\alpha_i)d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (58)$$

Para um manipulador de n graus de liberdade, a MTH de toda a cadeia cinemática pode ser expressa conforme equação 59, sendo q_n a variável de junta. Como neste projeto lida-se apenas com manipuladores compostos por juntas rotacionais, $q = \theta$ para todas as juntas. Esta notação deixa explícito a possibilidade de trabalhar com manipuladores de juntas prismáticas onde a variável é d .

$$H_n^0 = A_1^0(q_1)A_2^1(q_2) \dots A_n^{n-1}(q_n) \quad (59)$$

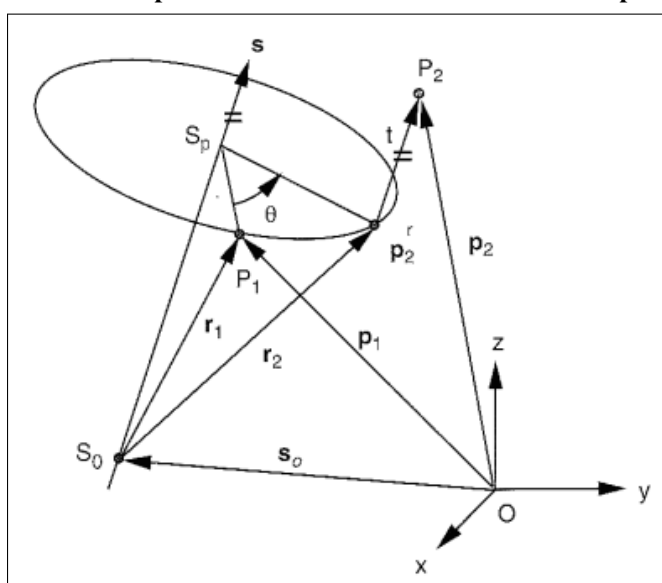
A matriz H resultante da operação de transformação pode ser aplicada na equação 50 do apêndice A para realizar a transformação de um ponto p no *frame* de referência, normalmente na coordenada da base do manipulador, até o efetuador. O resultado é o ponto transformado p' no formato de coordenadas homogêneas que descreve a posição do efetuador em relação à base. Esta informação junto da submatriz R encontrada conforme equação 49 do apêndice A fornece a descrição completa de posição e orientação, ou *pose*, do efetuador em relação ao *frame* de referência. Esta é, portanto, a solução da cinemática direta utilizando os parâmetros DH e Matrizes de Transformação Homogênea.

APÊNDICE D – HELICOIDES SUCESSIVOS

O método dos Helicoides Sucessivos para a representação matemática de uma cadeia cinemática é uma convenção alternativa aos clássicos parâmetros de Denavit-Hartenberg. Para o emprego deste método utiliza-se um elemento geométrico chamado *helicoides*, sendo este capaz de representar uma rotação e uma translação na forma análoga a um parafuso entre quaisquer dois sistemas de coordenadas. Os detalhes desta convenção são discutidos a fundo no livro de Tsai (1999).

Um deslocamento helicoidal pode ser descrito como um deslocamento de uma pose a outra ao aplicar uma rotação e uma translação ao longo de um eixo. Um exemplo de um deslocamento helicoidal é apresentado na figura 68, onde é descrito o deslocamento do ponto P_1 até o ponto P_2 . A transformação é feita através de uma rotação θ e uma translação τ ao longo do eixo s . Uma vez que estes parâmetros definem o deslocamento de um corpo rígido, pode-se desenvolver uma notação para a transformação serial de *frames* de referência.

Figura 68 – Exemplo de deslocamento helicoidal de um corpo rígido.



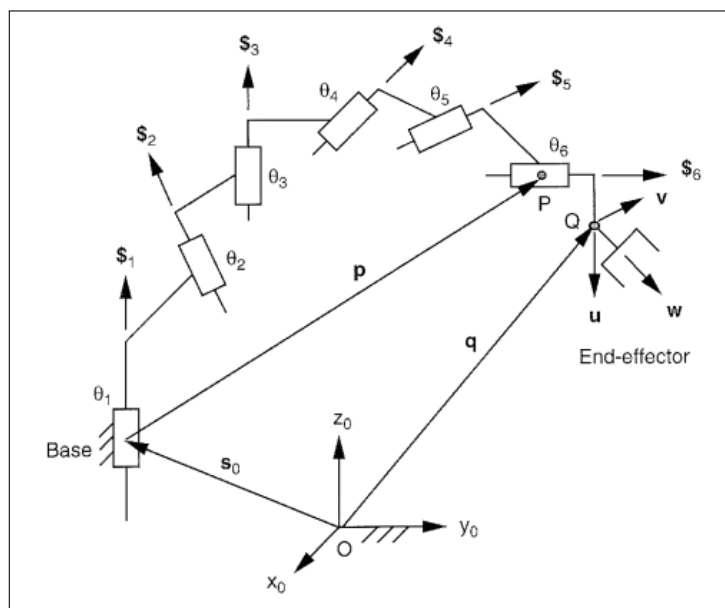
Fonte: Tsai (1999)

Para aplicar a convenção dos Helicoides Sucessivos, primeiramente é necessário selecionar uma pose de referência para o manipulador, onde todas as variáveis são conhecidas, preferencialmente onde os ângulos das juntas são zero. Conhecendo a posição espacial de cada junta e do efetuador, bem como definindo a direção do eixo de rotação de cada junta nesta configuração, basta inserir um elemento helicoidal em cada grau de liberdade, sendo:

- O vetor de direção do helicóide, ou seja, a direção do eixo de cada junta no formato $s_i = (s_x, s_y, s_z)$;
- O vetor de posição do eixo do helicóide, ou seja, a posição espacial da junta no formato $s_{0i} = (s_{0x}, s_{0y}, s_{0z})$;

A representação através de Helicóides Sucessivos é exemplificada na figura 69, onde cada junta do manipulador possui um deslocamento helicoidal $\$i$ associado. Conforme descrito pelo autor Tsai (1999), $\$i$ deve ser considerado como o resultado do deslocamento do *frame* $\$i-1$ através de seus parâmetros helicoidais. Ao aplicar a transformação *sucessiva* entre os helicóides de cada junta, obtém-se o *frame* de referência do efetuador, descrito por u, v, w e q , ainda na figura 69.

Figura 69 – Exemplo de deslocamento helicoidal em um manipulador.



Fonte: Tsai (1999)

Diferente da convenção de Denavit-Hartenberg, onde a transformação é realizada a partir do ponto p que representa a origem do sistema ou a base do manipulador, no método dos Helicóides Sucessivos a transformação é aplicada ao ponto p que descreve a posição do manipulador na configuração inicial adotada para a determinação dos helicóides.

Devido à diferença na referência da transformação, a matriz de transformação que considera rotações e translações descrita na equação 58 não se aplica. A estrutura de H continua igual à descrita na equação 49, porém os elementos R e d assumem a forma descrita nas equações 61 e 62, conhecida como Fórmula de Rotações Finitas de Rodrigues (CHENG; GUPTA, 1989).

Para simplificação de notação no documento, os elementos de R na equação 62 estão escritos em referência ao mapeamento apresentado na equação 60. De forma semelhante, na equação 61 entende-se que $c(\theta)$ e $s(\theta)$ são abreviações para $\cos(\theta)$ e $\sin(\theta)$, respectivamente.

$$R = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (60)$$

$$R = \begin{bmatrix} (s_x^2 - 1)(1 - c(\theta)) + 1 & s_x s_y (1 - c(\theta)) - s_z s(\theta) & s_x s_z (1 - c(\theta)) - s_y s(\theta) \\ s_y s_x (1 - c(\theta)) - s_x s(\theta) & (s_y^2 - 1)(1 - c(\theta)) + 1 & s_y s_z (1 - c(\theta)) - s_x s(\theta) \\ s_z s_x (1 - c(\theta)) - s_y s(\theta) & s_z s_y (1 - c(\theta)) - s_x s(\theta) & (s_z^2 - 1)(1 - c(\theta)) + 1 \end{bmatrix} \quad (61)$$

$$d = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} t s_x - s_{0x} (a_{11} - 1) - s_{0y} a_{12} - s_{0z} a_{13} \\ t s_y - s_{0x} a_{21} - s_{0y} (a_{22} - 1) - s_{0z} a_{23} \\ t s_z - s_{0x} a_{31} - s_{0y} a_{32} - s_{0z} (a_{33} - 1) \end{bmatrix} \quad (62)$$

APÊNDICE E – MATRIZ JACOBIANA NA CINEMÁTICA INVERSA DIFERENCIAL

A cinemática inversa diferencial tem como propósito estabelecer uma relação entre as velocidades de junta do manipulador e as velocidades linear e angular do efetuador. Isso significa que deve ser definida uma equação que relacione a derivada das variáveis de junta \dot{q} às derivadas de posição e orientação do efetuador \dot{p}_e e ω_e , conforme representado nas equações 63 e 64, onde as matrizes J_P e J_O são as matrizes Jacobianas ($3 \times n$) de posição e orientação, respectivamente.

$$\dot{p}_e = J_P(q)\dot{q} \quad (63)$$

$$\omega_e = J_O(q)\dot{q} \quad (64)$$

A definição de velocidade de um efetuador em função das variáveis de junta pode ser reescrita conforme equação 65, onde v_e é o vetor de velocidade linear e angular do efetuador e $J(q)$ é a matriz Jacobiana.

$$v_e = J(q)\dot{q} \quad (65)$$

Rearranjando os termos da equação 65, obtém-se a equação 66, onde é possível observar que a derivada das variáveis de junta pode ser obtida através da inversão da matriz Jacobiana, a qual será definida em seguida. Esta inversão é um dos problemas em aberto na área de cinemática uma vez que esta nem sempre é possível e também pode fornecer mais de uma possível solução.

$$\dot{q} = J^{-1}(q)v_e \quad (66)$$

Para definir os algoritmos que fornecem soluções para a equação 66, primeiramente é necessário descrever a computação da matriz Jacobiana. Existem duas definições principais, chamadas *Jacobiana Geométrica* e *Jacobiana Analítica*. Ambas são computadas de maneira diferente e tem significados distintos quanto à interpretação da parcela de orientação. A Jacobiana Geométrica é utilizada quando a pose do efetuador é descrita em relação ao *frame* da base, o que permite calcular cada coluna da matriz utilizando relações geométricas conforme equação 67.

$$J(q)_i = \begin{bmatrix} J_P(q)_i \\ J_O(q)_i \end{bmatrix} = \begin{cases} \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{para juntas prismáticas} \\ \begin{bmatrix} z_{i-1} \times (p_e - p_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{para juntas rotacionais} \end{cases} \quad (67)$$

Uma característica da equação 67 é que ela é dada apenas em função de variáveis já conhecidas do manipulador: p_e é a posição do efetuador em relação à base, p_{i-1} e z_{i-1} são a posição e o eixo de rotação da junta anterior à qual se está calculando a coluna da Jacobiana, disponíveis ao extrair as submatrizes da equação de cinemática direta, conforme equação 49.

Quando a pose do efetuador é especificada em termos de uma representação mínima como nas equações de 53 a 55, naturalmente surge a possibilidade de derivar a equação cinemática em relação às variáveis de junta para obter a matriz Jacobiana. Esta abordagem resulta na Jacobiana Analítica, a qual é representada conforme equação 68, onde $k(q)$ é a equação da cinemática direta do efetuador.

$$J_A(q) = \frac{\partial k(q)}{\partial q} \quad (68)$$

Para computar a Jacobiana Analítica é necessário realizar a diferenciação das parcelas de translação e rotação individualmente. Para encontrar a velocidade linear do efetuador, aplica-se a derivada da equação de posição em relação às variáveis de junta. Para a parcela representando a velocidade rotacional, a qual neste caso é dada em termos de representação mínima, ou seja, ϕ_e , não há uma função direta que permita realizar a diferenciação. Para computar esta parcela é necessário derivar a matriz de rotação R da equação cinemática, o que pode não ser trivial.

Ainda conforme Sciavicco e Siciliano (2012), é possível estabelecer relações entre as matrizes Jacobiana Analítica e Geométrica, além de que em seu trabalho os algoritmos de cinemática diferencial são descritos em termos da abordagem analítica. Para o projeto descrito nesta dissertação não são utilizadas variações de orientação ao longo da trajetória para manter em evidência os resultados de interesse, sendo normalmente estabelecido um valor de orientação fixo em relação ao *frame* da base, como é descrito ao longo do Capítulo 6. Também é demonstrado que os algoritmos funcionam da mesma forma aplicando a Jacobiana Geométrica em todos os casos, portanto os pormenores da Jacobiana Analítica não são abordados, estando estes disponíveis na respectiva referência bibliográfica.

O objetivo de computar a Jacobiana é utilizar a sua inversa para obter a velocidade das juntas, conforme apresentado na equação 66. Esta operação é definida de maneira simples porém representa uma área de interesse acadêmico devido à existência das chamadas *singularidades*.

A Jacobiana representa um mapeamento linear entre o espaço das juntas e o espaço operacional. Em situações onde o *rank* de J é deficiente, ou seja, há colunas na matriz que são linearmente dependentes, ocorre uma singularidade. É de interesse de qualquer aplicação com manipuladores implementar métodos que evitem singularidades pelas seguintes razões:

- Na configuração de singularidade, a mobilidade da estrutura é reduzida. Isso significa que é possível que existam movimentos realizados no espaço das juntas que não realizam nenhuma alteração no espaço operacional. Também significa que não é possível realizar movimentos arbitrários com o efetuador;
- Em uma singularidade ou nos arredores da mesma, é possível que se exija uma grande movimentação nas juntas intermediárias da cadeia cinemática para realizar um pequeno movimento do efetuador;
- Nos arredores de uma singularidade, é possível que existam infinitas soluções para a cinemática inversa.

O método de conhecimento comum para calcular matriz inversa de J , cuja notação é J^{-1} , sem utilizar longos sistemas de equações é a inversão pela matriz adjunta, descrito na equação 69.

$$J^{-1} = \frac{1}{\det(J)} \text{adj}(J) \quad (69)$$

Este método exige em uma de suas etapas o cálculo do determinante de J , o qual é aplicado no denominador, trazendo duas limitações:

- Caso $\det(J) = 0$, o cálculo torna-se indefinido e não há uma solução possível, causando uma singularidade.
- O determinante pode ser calculado apenas para matrizes quadradas. A exemplo: caso o espaço operacional seja definido em três variáveis de posição e três de orientação, o manipulador deve ter exatamente seis graus de liberdade.

Métodos alternativos para a inversão da matriz Jacobiana são extensamente investigados na literatura relacionada. Buss e Kim (2005) descrevem e comparam as principais abordagens, sendo as duas de maior relevância para este projeto a Jacobiana Pseudo-inversa e os Mínimos Quadrados Amortecidos (DLS, do inglês *Damped Least Squares*).

A Jacobiana pseudoinversa, de notação J^\dagger , permite a inversão de matrizes de qualquer dimensão e tem a propriedade de minimizar localmente a norma das velocidades das juntas. A computação é realizada em função da matriz transposta, conforme equação 70.

$$J^{-1} \equiv J^\dagger = J^T (J J^T)^{-1} \quad (70)$$

A inversão utilizando J^\dagger permite a realização do cálculo porém não evita que o manipulador caia em uma configuração de mapeamento singular. Uma técnica que introduz robustez ao cálculo na vizinhança de singularidades é o DLS, o qual estende a pseudoinversa com um fator de amortecimento λ . O valor do fator de amortecimento pode ser determinado de diversas maneiras, sendo constante ou variável, como demonstrado nas análises feitas por Nakamura e Hanafusa (1986), Chiaverini et al. (1992) e Deo e Walker (1995), além de poder ser definida de forma empírica como descrito no trabalho de Buss e Kim (2005) e usado recentemente em Jesus et al. (2022). A inversão pelo método DLS é apresentada na equação 71, onde I é a matriz identidade que permite o ajuste de dimensões no cálculo.

$$J^{-1} \equiv J^* = J^T (J J^T + \lambda^2 I)^{-1} \quad (71)$$

No decorrer deste documento, a notação utilizada ao referir-se à inversão da Jacobiana é J^\dagger , subentendendo-se que a adição do elemento de amortecimento não é obrigatória.

A equação 66 fornece \dot{q} , que é a derivada das variáveis de junta. Na prática, deseja-se computar as variáveis de junta em si para formar a trajetória no espaço das juntas do manipulador. Dessa forma é necessário realizar a integração da equação. Como os algoritmos são executados em ambiente digital, utiliza-se o método de integração discreta de Euler, o qual fornece os valores de q no instante (ou iteração) seguinte, conforme equação 72.

$$q(t_{k+1}) = q(t_k) + \dot{q}(t_k) \Delta t \quad (72)$$

Todas as variáveis necessárias para computar a equação 72 estão disponíveis: $q(t_k)$ refere-se aos valores das variáveis de junta no momento atual, $\dot{q}(t_k)$ é a solução da equação 66 para cinemática inversa e Δt é o intervalo de integração. Cabe salientar que, conforme explanado

no decorrer do capítulo 4, o intervalo de integração aplica-se em situações onde de fato se está calculando velocidades reais atreladas ao tempo de execução da trajetória. Esta metodologia também pode ser aplicada em métodos iterativos.

APÊNDICE F – CINEMÁTICA ATRAVÉS DA ÁLGEBRA DE CLIFFORD

O desenvolvimento da álgebra de Clifford, cuja autoria é do matemático William Kingdon Clifford, iniciou tendo como base os trabalhos de Sir William R. Hamilton. O trabalho de Hamilton abordou a generalização dos números complexos para operações algébricas em três dimensões, o que deu origem aos quatérnios. Quatérnios são elementos adequados para a descrição de transformações envolvendo rotação de objetos no espaço tridimensional (HART et al., 1994).

A descoberta dos quatérnios foi feita por Hamilton em 1843, sendo publicada no ano seguinte nos primeiros trabalhos denominados *On a new Species of Imaginary Quantities Connected with a theory of Quaternions* (HAMILTON, 1843) e *On Quaternions* (HAMILTON, 1844a), sendo que o trabalho completo sobre a teoria foi publicado em 1844 no artigo *Theory of quaternions* (HAMILTON, 1844b). Posteriormente, em 1866, Hamilton publicou um livro completo sobre o tema chamado *Elements of Quaternions* (HAMILTON, 1866).

Em 1873, William Kingdon Clifford emprega o termo biquatérnios ao descrever um elemento no formato $q + wr$, onde q e r são quatérnios definidos conforme os trabalhos de Hamilton, em seu trabalho considerado um clássico denominado *Preliminary Sketch of Biquaternions* (CLIFFORD, 1873). O termo ω empregado por Clifford recebeu diferentes definições em seus trabalhos, tornando seu significado ambíguo. Inicialmente o elemento tem a propriedade de que seu quadrado é nulo, $\omega^2 = 0$, o que o torna uma entidade algébrica abstrata. No mesmo trabalho, Clifford também usa a definição $\omega^2 = 1$ em um outro contexto. Em outros trabalhos de Clifford, como em *Applications of Grassmann's Extensive Algebra* (CLIFFORD, 1878) e *On the Classification of Geometric Algebras* (CLIFFORD, 1882), a propriedade do quadrado de ω é definida como $\omega^2 = \pm 1$. Estas definições permaneceram ambíguas até o ano de 1978, onde Rooney define em *On the three types of complex number and planar transformations* (ROONEY, 1978) as interpretações no contexto algébrico da cinemática.

Segundo Rooney, a propriedade do quadrado do operador ω define o elemento em três categorias, sendo estas:

1. Número complexo: representado no formato $a + ib$, sendo que $i^2 = -1$;
2. Número dual: representado no formato $a + eb$, sendo que $e^2 = 0$;
3. Número duplo: representado no formato $a + jb$, sendo que $j^2 = 1$;

Cada um dos tipos de número complexo possui suas próprias características, e ao definir o par (a,b) como dois quatérnios clássicos de Hamilton, por exemplo r e s , obtém-se três categorias de elementos algébricos definidos conforme nomenclatura apresentada nos trabalhos de Altmann (2005), Selig (2005) e Conway e Smith (2003):

1. Número hipercomplexo ou Octônio: representado no formato $r + is$, sendo que $i^2 = -1$;
2. Quatérnio dual: representado no formato $r + es$, sendo que $e^2 = 0$;
3. Quatérnio duplo: representado no formato $r + js$, sendo que $j^2 = 1$;

A álgebra de Clifford, também denominada Álgebra Geométrica (OLIVEIRA, 2011), é associativa e gerada por elementos base e_1, e_2, \dots, e_n . A notação para caracterizar uma álgebra de Clifford é dada conforme $Cl(p,q,r)$, onde p, q e r são o número de bases geradoras cujo quadrado é igual a 1, -1 e 0, respectivamente. Exemplificando, uma álgebra denotada como $Cl(0,3,1)$ possui 3 bases geradoras e_1, e_2 e e_3 cujo quadrado é -1 e uma base geradora e cujo quadrado é 0. Esta é uma álgebra de grau 4, uma vez que é composta por 4 bases geradoras e possui, portanto, $2^4 = 16$ elementos. Os elementos geradores da álgebra $Cl(0,3,1)$ são:

$$\left(\begin{array}{ll} 1 & \text{grau 0} \\ e_1, e_2, e_3, e & \text{grau 1} \\ e_1e_2, e_1e_3, e_2e_3, e_1e, e_2e, e_3e & \text{grau 2} \\ e_1e_2e_3, e_1e_2e, e_1e_3e, e_2e_3e & \text{grau 3} \\ e_1e_2e_3e & \text{grau 4} \end{array} \right)$$

As álgebras de Clifford podem ser divididas em subálgebras par e ímpar, apenas separando as bases formadas pela quantidade de elementos geradores utilizados. Uma subálgebra de Clifford par é representada por $Cl^+(p,q,r)$ e uma ímpar por $Cl^-(p,q,r)$. Exemplificando, a álgebra $Cl(0,3,1)$ pode ser dividida em subálgebras pares e ímpares conforme:

$$Cl^+(0,3,1) = \left(\begin{array}{ll} 1 & \text{grau 0} \\ e_1e_2, e_1e_3, e_2e_3, e_1e, e_2e, e_3e & \text{grau 2} \\ e_1e_2e_3e & \text{grau 4} \end{array} \right)$$

$$Cl^-(0,3,1) = \left(\begin{array}{ll} e_1, e_2, e_3, e & \text{grau 1} \\ e_1e_2e_3, e_1e_2e, e_1e_3e, e_2e_3e & \text{grau 3} \end{array} \right)$$

Faz-se necessária a desambiguação dos termos referentes a cada parcela da álgebra. Conforme será discutido ao longo da seção F, a cinemática é descrita em sua totalidade apenas utilizando os elementos pares deste caso especial. O termo usado para se referenciar aos 16 elementos da álgebra $Cl(0,3,1)$ é *Grupo do Movimento de Corpos Rígidos* (SELIG, 2000), e o termo utilizado para referir-se aos oito elementos de grau par em $Cl^+(0,3,1)$ é *álgebra Dual-Quaterniônica* e cada elemento é um *quatérnio dual* (ROONEY, 1978).

Neste contexto, existem álgebras chamadas isomórficas, uma vez que contém igual quantidade de termos e implicam em relações aritméticas semelhantes. Uma álgebra de Clifford qualquer é isomórfica a uma subálgebra par de grau superior, ou seja:

$$Cl(p,q,r) = Cl^+(p,q+1,r)$$

As relações associativas das bases geradoras definem as propriedades de uma determinada álgebra de Clifford com base em três axiomas (OLIVEIRA, 2011), sendo:

1. O produto dos elementos de base da álgebra de Clifford é a justaposição dos mesmos, resultando em um novo termo da álgebra:

$$e_i \cdot e_j = e_i e_j \quad (73)$$

2. O produto dos elementos das bases geradoras é anti-comutativo:

$$e_i e_j = -e_j e_i \quad (74)$$

3. O conjugado de um elemento é dado pelo negativo de suas bases geradoras e invertendo a ordem:

$$(e_i e_j)^* = (e_j^*)(e_i^*) = (-e_j)(-e_i) = -e_i e_j \quad (75)$$

Com as definições que geram todos os elementos e propriedades das álgebras de Clifford, serão descritas as propriedades dos quatérnios de Hamilton. Esta abordagem tem o intuito de relacionar os quatérnios a elementos pertencentes a uma álgebra de Clifford, o que permite a aplicação de suas propriedades de forma análoga.

QUATÉRNIOS

Para compreender a relação da álgebra de Clifford com as equações de cinemática, primeiro deve-se expor a sua relação com a teoria dos quatérnios. FAMILTON (2015) realizou uma revisão dos pormenores do problema que Hamilton tentou resolver durante 10 anos, ou seja, obter uma forma de descrever pontos em 3 dimensões da mesma forma que é possível descrever pontos em 2 dimensões através de números complexos na forma retangular $(a + bi)$. Na primeira publicação alusiva aos quatérnios de Hamilton em 1844, o autor descreve sua conclusão:

...E aqui caiu em mim a noção de que devemos admitir, em algum sentido, uma quarta dimensão de espaço para cálculo com tríades; ou transferindo o paradoxo para a álgebra, devemos admitir um terceiro imaginário distinto de símbolo k a não ser confundido nem com i nem com j mas igual ao produto do primeiro como multiplicador e o segundo como multiplicando; portanto, fui conduzido a introduzir os quatérnios, de tal forma que $q = a + bi + cj + dk$, ou (a, b, c, d) . (HAMILTON, 1843, 492-493)

Conforme descrito na tese de FAMILTON (2015), apesar de Hamilton ter criado uma relação entre os quatérnios e rotações espaciais, ele nunca pôde aplicar os quatérnios dentro do contexto de sua época. A álgebra quaterniônica não voltou a ser aplicada no meio científico até o final do século XX quando estes surgiram como uma alternativa mais rápida computacionalmente do que as já conhecidas matrizes para realizar rotações em animações gráficas.

Hamilton define, então, os quatérnios como entidades com três componentes complexas que assumem interpretação real em rotação de sistemas. As três componentes i , j e k são interpretadas como versores mutuamente ortogonais e satisfazem as seguintes condições:

$$i^2 = j^2 = k^2 = ijk$$

$$jk = i = -kj$$

$$ij = k = -ji$$

$$ki = j = -ik$$

Dado um quatérnio no formato $q = a + bi + cj + dk$ (não confundir a notação q de um quatérnio com a notação idêntica utilizada para as variáveis de junta de um manipulador), e partindo de suas definições fundamentais, derivam-se suas propriedades aritméticas. Os axiomas que envolvem as aplicações em rotações espaciais são descritos conforme:

1. **Multiplicação:** A multiplicação de dois quatérnios, $q_1 = a_1 + b_1i + c_1j + d_1k$ e $q_2 = a_2 + b_2i + c_2j + d_2k$ é dada pelo produto polinomial de seus termos. Devido à propriedade anti-comutativa dos elementos de um quatérnio, $q_1q_2 \neq q_2q_1$.

$$\begin{aligned} q_1q_2 = & (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) \\ & + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)i \\ & + (a_1c_2 - b_1d_2 + c_1a_2 + d_1b_2)j \\ & + (a_1d_2 + b_1c_2 - c_1b_2 + d_1a_2)k \end{aligned} \quad (76)$$

Uma representação simplificada da multiplicação pode ser obtida rearranjando os elementos da equação 76, dada conforme equação 77, onde w_i e v_i são as parcelas escalar e complexa de q_i , respectivamente;

$$q_1q_2 = [w_1w_2 - v_1 \cdot v_2, w_1v_2 + w_2v_1 + (v_1 \times v_2)] \quad (77)$$

2. **Conjugado:** o conjugado de um quatérnio é dado pela inversão de sinal de suas parcelas complexas.

$$q_1^* = a - bi - cj - dk \quad (78)$$

Além disso, $(q_1q_2)^* = q_2^*q_1^*$;

3. **Norma:** a norma de um quatérnio $\|q\|$ é dada pela soma dos quadrados de todos os coeficientes.

$$\|q\| = a^2 + b^2 + c^2 + d^2 \quad (79)$$

Da equação 79 também pode-se extrair a relação $\|q\| = q^*q = qq^*$;

4. **Módulo:** o módulo de um quatérnio é dado pela extração da raiz quadrada de sua norma;

$$|q| = \sqrt{\|q\|} \quad (80)$$

5. **Inverso:** o inverso de um quatérnio é dado pela divisão do conjugado pela norma;

$$q^{-1} = \frac{q^*}{\|q\|} \quad (81)$$

6. Divisão: a divisão de q_1 por q_2 é feita através da multiplicação de q_1 pelo inverso de q_2 ;

$$\frac{q_1}{q_2} = q_1 q_2^{-1} \quad (82)$$

7. Quatérnio Puro: um quatérnio puro é aquele cuja componente real é nula;

$$q = bi + cj + dk \quad (83)$$

8. Quatérnio Unitário: um quatérnio é unitário quando:

$$\|q\| = 1 \quad (84)$$

A aplicação de quatérnios unitários leva à simplificação da formulação de seu inverso, uma vez que neste caso ela é equivalente ao seu conjugado. Portanto, para o caso em que q é um quatérnio unitário, a equação 85 é verdadeira.

$$q^{-1} = q^* \quad (85)$$

Como discutido anteriormente, o trabalho de Hamilton envolvendo quatérnios foi um dos motivadores para o estudo que levou Clifford a desenvolver sua álgebra. É possível definir uma álgebra de Clifford que implique nas mesmas propriedades aritméticas que os quatérnios. A álgebra $C\ell(0,2,0)$, a qual possui dois elementos e_1 e e_2 cujo quadrado é igual a -1 é isomórfica aos quatérnios de Hamilton. Dessa forma, é possível interpretar um quatérnio como um caso específico da álgebra de Clifford (OLIVEIRA, 2011).

A partir das bases geradoras da álgebra $C\ell(0,2,0)$, é possível gerar conjunto de elementos $(1, e_1, e_2, e_1 e_2)$ que formam as relações associativas desta álgebra. As relações são descritas na Tabela 9.

Tabela 9 – Tabela de Multiplicação dos elementos de um quatérnio.

	1	i	j	k
1	1	i	j	k
i	i	-1	k	$-j$
j	j	$-k$	-1	i
k	k	j	$-i$	-1

Fonte: Autoria própria.

Considerando um elemento da álgebra $Cl(0,2,0)$ como $q = a_1 + a_2e_1 + a_3e_2 + a_4e_1e_2$ e realizando a multiplicação polinomial de dois destes elementos q_1 e q_2 , define-se a relação de multiplicação conforme equação 86.

$$\begin{aligned}
 q_1q_2 = & (a_1b_1 - a_2b_2 - a_3b_3 - a_4b_4) \\
 & + (a_1b_2 + a_2b_1 + a_3b_4 - a_4b_3)e_1 \\
 & + (a_1b_3 - a_2b_4 + a_3b_1 + a_4b_2)e_2 \\
 & + (a_1b_4 + a_2b_3 - a_3b_2 + a_4b_1)e_1e_2
 \end{aligned} \tag{86}$$

Ao comparar a equação 76 com a equação 86, torna-se evidente a relação entre a álgebra de Clifford e os quatérnios de Hamilton quando considera-se as seguintes equivalências:

$$i = e_1$$

$$j = e_2$$

$$k = e_1e_2$$

A equivalência entre álgebras permite concluir que a $Cl(0,2,0)$ possui características capazes de representar rotações. Para aplicar operações de transformação é necessário, portanto, definir pontos no espaço usando os elementos dessa álgebra.

Conforme Selig (2000), um ponto espacial é definido dentro da álgebra $Cl(0,2,0)$ através da projeção do vetor $p = (p_x, p_y, p_z)$, onde p_x , p_y e p_z são as coordenadas cartesianas de p no espaço tridimensional, sendo:

$$p = p_xe_1 + p_ye_2 + p_z e_1e_2 \tag{87}$$

A aplicação de rotações por quatérnios é discutida de forma detalhada no trabalho de Selig (2000), onde demonstra-se que um vetor de posição arbitrário p pode ser rotacionado em torno de um vetor unitário \mathbf{n} definido por um quatérnio unitário g em um ângulo duplo 2ϕ . Dessa forma, define-se um quatérnio de rotação utilizando a representação de meio ângulo $\frac{\phi}{2}$. Um quatérnio que aplica uma rotação de ϕ em torno de um vetor unitário \mathbf{n} é definido como:

$$g = \cos\left(\frac{\phi}{2}\right) + \mathbf{n} \operatorname{sen}\left(\frac{\phi}{2}\right) [e_1 + e_2 + e_1e_2] \tag{88}$$

Um quatérnio de rotação é aplicado a um vetor que representa a posição de um ponto realizando a pré-multiplicação pelo quatérnio de rotação e a pós-multiplicação pelo seu complexo conjugado.

$$p' = gpg^* \quad (89)$$

A representação mínima de orientação de corpos rígidos através dos ângulos RPY é apresentada no anexo B, onde os valores são computados a partir da matriz de rotação e utilizando relações com a função $\text{atan2}(x,y)$. Os quatérnios permitem uma solução para o problema de descontinuidade e indefinição inerente deste método. Os ângulos Roll (φ), Pitch (ϑ) e Yaw (ψ) de um quatérnio unitário escrito na forma da equação 88, são obtidos utilizando as relações:

$$[\varphi, \vartheta, \psi] = [2 \text{sen}^{-1}(\mathbf{n}_1), 2 \text{sen}^{-1}(\mathbf{n}_2), 2 \text{sen}^{-1}(\mathbf{n}_3)] \quad (90)$$

O erro de orientação pode ser computado utilizando os ângulos RPY da mesma forma que o realizado com a representação por matrizes de transformação homogênea, porém há outra relação onde não é necessário computar o valor dos ângulos. A diferença de orientação entre o quatérnio desejado q_d e o quatérnio atual q é dada conforme equação 91.

$$\Delta q = q_d q^{-1} \quad (91)$$

Caso a orientação desejada seja idêntica à atual, então $\Delta q = 1$, indicando que os elementos e_1 , e_2 e $e_1 e_2$ tem seus respectivos coeficientes A_1 , A_2 e A_3 nulos. É suficiente, portanto, definir o erro de orientação entre dois quatérnios conforme equação 92.

$$e_O = [A_1, A_2, A_3] \quad (92)$$

Um quatérnio pode facilmente representar rotações espaciais, porém não possui elementos para fornecer graus de liberdade suficientes para uma representação simultânea de rotações e translações. Dessa forma, na continuação do trabalho de Clifford, os quatérnios são expandidos para elementos de outra álgebra contendo mais elementos. Estes elementos são denominados biquatérnios e possuem um caso específico conhecido como quatérnios duais (ROONEY, 1978).

QUATÉRNIOS DUAIS

Um biquatérnio é definido por Clifford (1873) como a associação de dois quatérnios de Hamilton através do operador ω . Neste apêndice primeiramente apresenta-se os quatérnios duais de acordo com a notação que não leva as bases geradoras e_i da álgebra de Clifford, usada em trabalhos como o de Kenwright (2012).

$$Q = q_1 + q_2\omega \quad (93)$$

Apesar das várias definições de ω dadas por Clifford em seus trabalhos, conforme discutido anteriormente, o trabalho de Rooney (1978) contém a desambiguação do termo onde é definido que quando $\omega^2 = 0$ o biquatérnio Q é chamado de quatérnio dual. Neste caso, ω é representado como ε e, portanto, um quatérnio dual é descrito como:

$$Q = q_1 + q_2\varepsilon \mid \varepsilon^2 = 0 \quad (94)$$

Este elemento é capaz de representar simultaneamente rotações e translações espaciais de maneira semelhante a como um quatérnio unitário é capaz de representar rotações. Na equação 95, h é o quatérnio dual que realiza a translação e rotação do elemento ξ .

$$\xi' = h\xi h^* \quad (95)$$

O operador h , o qual representa a operação de transformação espacial, é definido conforme equação 96, onde g é o quatérnio unitário de rotação e t é um quatérnio puro representando o vetor de translação.

$$h = g + \frac{1}{2}tg\varepsilon \quad (96)$$

Ainda conforme Selig (2000), a condição de que o elemento de transformação deve ser unitário culmina na propriedade de que a multiplicação de h pelo seu conjugado também deve ser unitária, conforme equação 97.

$$hh^* = 1 \quad (97)$$

Por fim, da mesma maneira que um quatérnio de Hamilton é representado completamente pela álgebra $\mathcal{Cl}(0,2,0)$, um quatérnio dual é completamente descrito pela álgebra

$Cl(0,3,1)^+$ (SELIG, 2000). A notação utilizada para comparar um quatérnio dual com a álgebra de Clifford é dada termo a termo conforme:

$$\begin{aligned} i &= e_2e_3 & i\varepsilon &= e_1e \\ j &= e_3e_1 & j\varepsilon &= e_2e \\ k &= e_1e_2 & k\varepsilon &= e_3e \\ \varepsilon &= -e_1e_2e_3e \end{aligned}$$

Através desta notação, reescreve-se o quatérnio de Hamilton como o elemento de transformação rotacional:

$$g = \cos\left(\frac{\phi}{2}\right) + \mathbf{n} \operatorname{sen}\left(\frac{\phi}{2}\right) [e_2e_3 + e_3e_1 + e_1e_2] \quad (98)$$

Um ponto no espaço, bem como um elemento indicando componentes translacionais de uma transformação denotado como t , pode ser escrito na álgebra de Clifford como:

$$t = t_x e_1 + t_y e_2 + t_z e_3 \quad (99)$$

Isso permite criar o elemento de transformação dual-quatérniônica, isto é, rotação e translação simultâneas dentro da álgebra de Clifford, de forma idêntica à equação 96, apenas indicando a relação entre ε e e , sendo:

$$h = g + \frac{1}{2}tge \quad (100)$$

Para aplicar a transformação a elementos cinemáticos, primeiro é necessário definir a operação de conjugação de um quatérnio dual. De acordo com Ell e Sangwine (2007), o conjugado de um quatérnio unitário é definido de forma unívoca seguindo a equação 78, porém, conforme Bekar e Yayı (2013), existem três maneiras diferentes de definir o conjugado de um quatérnio dual.

Ao que tange a aplicação em cinemática, o conjugado dual-quatérniônico é aplicado ao realizar a operação de transformação descrita na equação 95, onde o elemento rígido ξ sofre a rotação e translação descrita por h . Dessa forma, conforme exposto por Tsai (1999), pode-se computar o conjugado de um quatérnio dual de transformação conforme a equação 101.

$$h^* = g^* + \frac{1}{2}g^*te \quad (101)$$

Com estas definições, para representar um corpo rígido no espaço ξ utilizando a notação de quatérnio duais, basta descrevê-lo utilizando uma variável da álgebra de Clifford $C\ell(0,3,1)$ contendo a informação de orientação g_ξ e posição t_ξ .

$$\xi = g_\xi + \frac{1}{2}t_\xi g_\xi e \quad (102)$$

Para tornar completa a aplicação dos quatérnios duais na cinemática direta, é necessário definir a transformação consecutiva entre os elementos de uma cadeia cinemática. De forma análoga à multiplicação concatenada realizada com matrizes de transformação homogênea, descrita na equação 51, a transformação sucessiva de quatérnios duais é dada em uma organização no formato de camadas (OLIVEIRA, 2011). A equação 103 descreve a transformação acoplada do elemento ξ ao longo de uma cadeia cinemática de n graus de liberdade. A transformação é feita respeitando a definição da equação 95, onde as transformações nas camadas próximas a ξ são as primeiras a ser realizadas e as mais externas são as últimas.

$$\xi' = (h_n \dots (h_2 (h_1 (\xi) h_1^*) h_2^*) \dots h_n^*) \quad (103)$$

A transformação acoplada que resulta no elemento transformado ξ' é a solução da cinemática direta através de quatérnios duais. Da mesma forma que para a solução através das MTH, é necessário definir um sistema de parâmetros para construir um quatérnio dual de transformação. Isso significa que é possível aplicar tanto a convenção Denavit-Hartenberg ou os Helicóides Sucessivos para resolver a cinemática através de quatérnios duais.

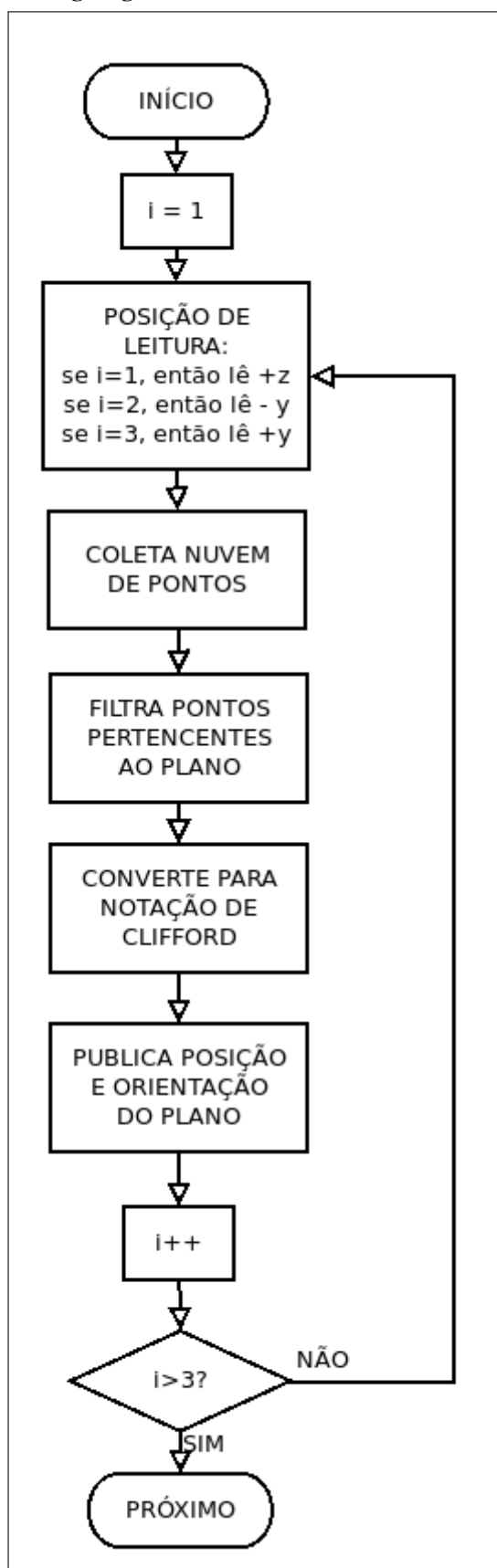
APÊNDICE G – DETALHAMENTO DOS BLOCOS DO ORGANOGRAMA DO PROJETO EXPERIMENTAL

Figura 70 – Organograma do bloco "CONEXÃO ROS".



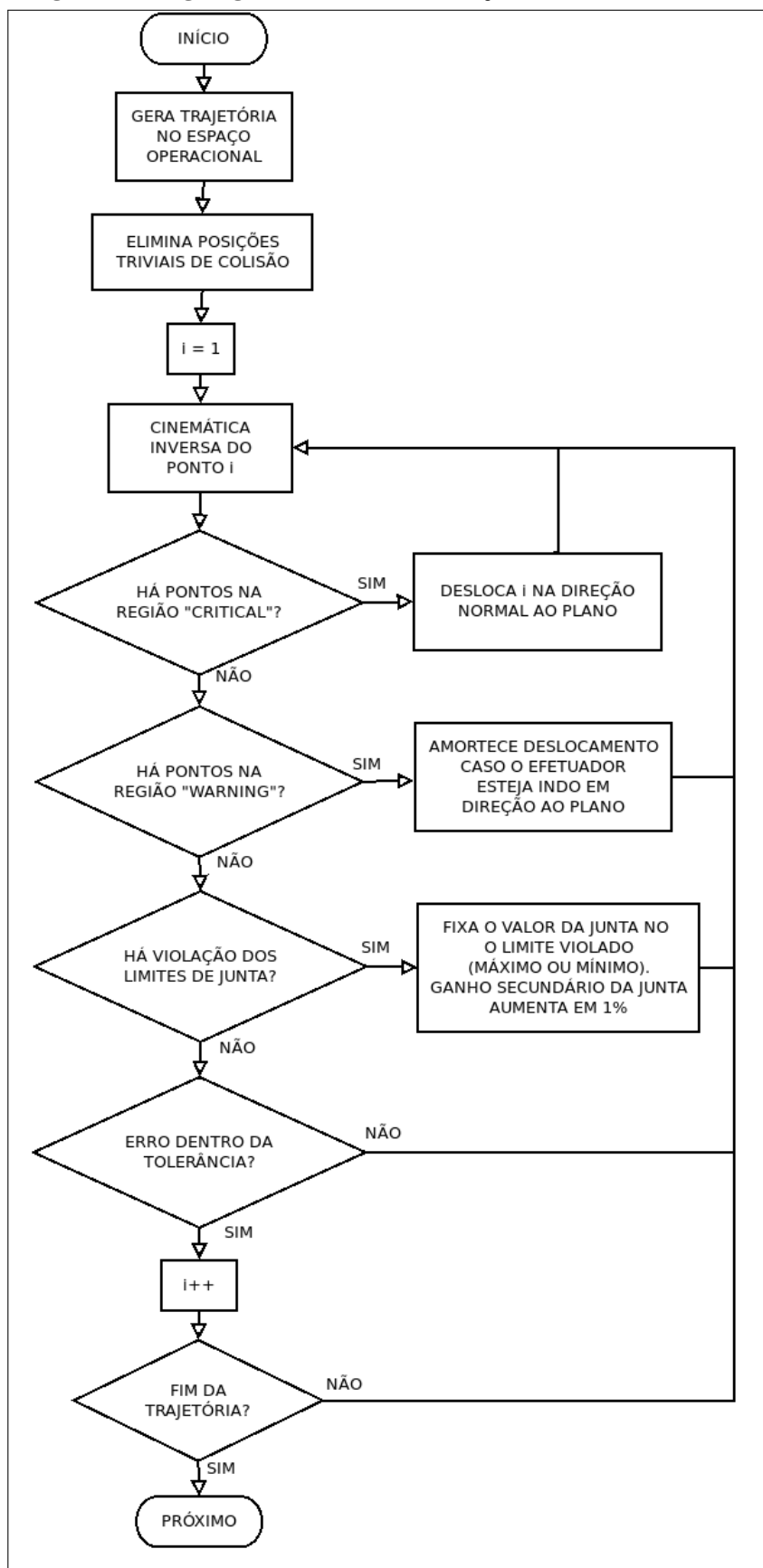
Fonte: Autoria própria.

Figura 71 – Organograma do bloco "LEITURA DOS LIMITES".



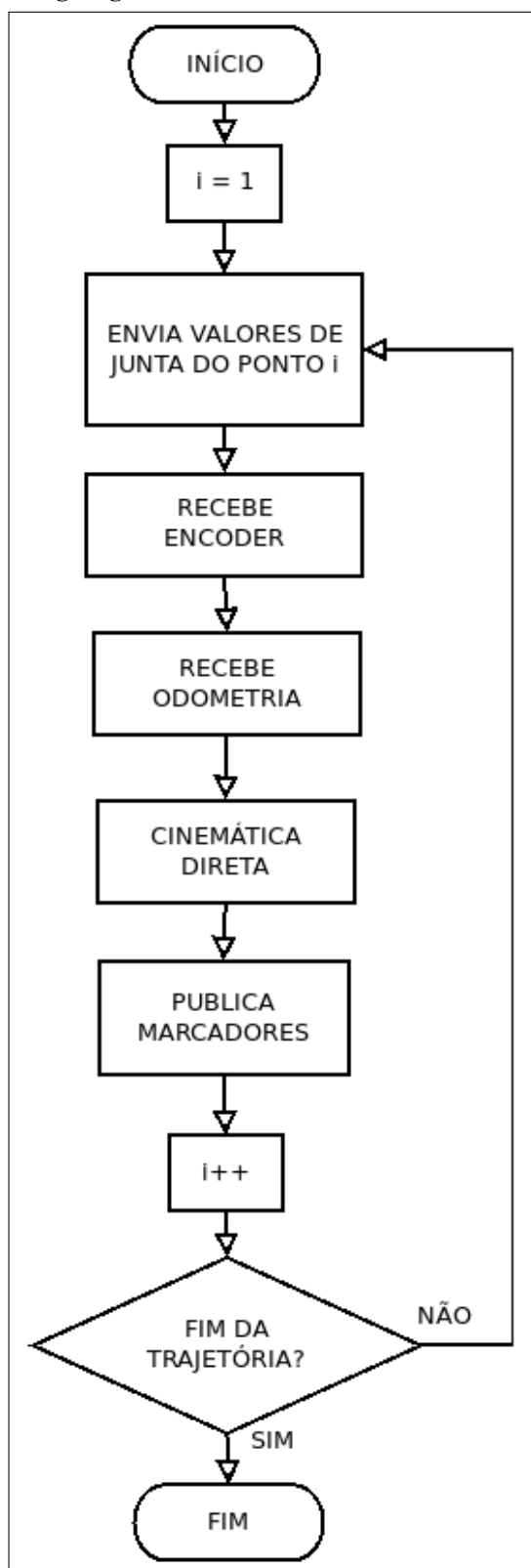
Fonte: Autoria própria.

Figura 72 – Organograma do bloco "GERAÇÃO DE TRAJETÓRIA".



Fonte: Autoria própria.

Figura 73 – Organograma do bloco "ENVIO AO MANIPULADOR".



Fonte: Autoria própria.