

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
DEPARTAMENTO DE ENERGIA
ENGENHARIA ELÉTRICA - HABILITAÇÃO EM ROBÓTICA E AUTOMAÇÃO
INDUSTRIAL

Bruno de Oliveira Melão

**Integração de Visão Computacional e Robótica Clássica para Manipulação
Assistiva**

Juiz de Fora

2025

Bruno de Oliveira Melão

**Integração de Visão Computacional e Robótica Clássica para Manipulação
Assistiva**

Trabalho de Conclusão de Curso apresentado
à Faculdade de Engenharia da Universidade
Federal de Juiz de Fora, como requisito para
a obtenção do grau de Engenheiro Eletricista.

Orientador: Leonardo Rocha Olivi

Juiz de Fora

2025

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Integração de Visão Computacional e Robótica Clássica para Manipulação Assistiva / Bruno de Oliveira Melão. – 2025.

71 f. : il.

Orientador: Leonardo Rocha Olivi

Dissertação – UNIVERSIDADE FEDERAL DE JUIZ DE FORA, DEPARTAMENTO DE ENERGIA. ENGENHARIA ELÉTRICA - HABILITAÇÃO EM ROBÓTICA E AUTOMAÇÃO INDUSTRIAL, 2025.

1. Visão Computacional.
 2. Assistência ao Vestir.
 3. Robótica Assistiva.
 4. Manipuladores Robóticos.
- I. Olivi, Leonardo R. II. Título.

Bruno de Oliveira Melão

Integração de Visão Computacional e Robótica Clássica para Manipulação Assistiva

Trabalho de Conclusão de Curso apresentado
à Faculdade de Engenharia da Universidade
Federal de Juiz de Fora, como requisito para
a obtenção do grau de Engenheiro Eletricista.

Aprovada em 19 de Março de 2025.

BANCA EXAMINADORA

Leonardo Rocha Olivi
- Orientador
Universidade Federal de Juiz de Fora (UFJF)

Exuperry Barros Costa
Universidade Federal de Juiz de Fora (UFJF)

Luís Henrique Lopes Lima
Universidade Federal de Juiz de Fora (UFJF)

Maria Fernanda Paulino Gomes
Universidade Estadual de Campinas (UNICAMP)



UNIVERSIDADE FEDERAL DE JUIZ DE FORA
REITORIA - FACENG - Depto. de Energia Elétrica

ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO (TCC)

CANDIDATO(A): Bruno de Oliveira Melão

ORIENTADOR(A): Prof. Dr. Leonardo Rocha Olivi

TÍTULO DO TRABALHO: "**Integração de Visão Computacional e Robótica Clássica para Manipulação Assistiva**"

BANCA EXAMINADORA:

Presidente: Leonardo Rocha Olivi / Universidade Federal de Juiz de Fora (UFJF)

Membro 1: Exuperry Barros Costa / Universidade Federal de Juiz de Fora (UFJF)

Membro 2: Luís Henrique Lopes Lima / Universidade Federal de Juiz de Fora (UFJF)

Membro 3: Maria Fernanda Paulino Gomes / Universidade Estadual de Campinas (UNICAMP)

LOCAL: LABRA - Laboratório de Automação e Robótica e Videoconferência (link <https://meet.google.com/zov-ghru-nbq>)

HORA DE ÍNICO: 17:00

Em seção publica, após exposição oral de 20 minutos do trabalho, o(a) candidato(a) foi arguido pelos membros da banca, os quais reuniram-se reservadamente, e decidiram pela:

APROVAÇÃO

REPROVAÇÃO

Para constar, a presente Ata, que aprovada por todos os presentes, vai assinada pelo orientador de TCC, pelos demais membros da banca e pelo candidato.

Juiz de Fora, 19 de março de 2025



Documento assinado eletronicamente por **Exuperry Barros Costa, Professor(a)**, em 19/03/2025, às 18:12, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Leonardo Rocha Olivi, Professor(a)**, em 19/03/2025, às 18:27, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Bruno de Oliveira Melão, Usuário Externo**, em 19/03/2025, às 18:55, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Luis Henrique Lopes Lima, Professor(a)**, em 19/03/2025, às 20:33, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade desse documento pode ser conferida no Portal do SEI-Ufjf (www2.ufjf.br/SEI) através do ícone Conferência de Documentos, informando o código verificador **2300250** e o código CRC **4A9195F6**.

Referência: Processo nº 23071.912625/2025-44

SEI nº 2300250

Documento assinado digitalmente
 MARIA FERNANDA PAULINO GOMES
Data: 20/03/2025 10:22:15-0300
Verifique em <https://validar.itigov.br>

AGRADECIMENTOS

Primeiramente, gostaria de agradecer à minha família, especialmente aos meus pais, Adilson e Rosilene, que me apoiaram durante toda a minha caminhada e não mediram esforços para que eu pudesse alcançar meus objetivos. Agradeço por terem acreditado e investido na minha educação e na do meu irmão, oferecendo apoio não apenas financeiro, mas também emocional durante toda essa jornada. Devo a eles essa conquista.

Agradeço ao meu irmão Eduardo por ter sido um exemplo para mim desde a infância. Obrigado por todos os conselhos, por todas as orientações vocacionais que me trouxeram até este momento e pela ajuda durante a graduação.

Agradeço à minha namorada, Mel, por sempre acreditar em mim, mesmo quando eu mesmo duvidava. Obrigado pelo amor, pelo carinho e pelo apoio emocional durante todo esse caminho.

Ao meu professor e orientador Leonardo Olivi, agradeço não apenas pela orientação durante a realização deste trabalho, sempre me ajudando, mas também pela excelência como professor. Seu amor pela robótica inspira.

Estendo meu agradecimento a todos os professores que contribuíram para a minha formação no Curso de Engenharia Elétrica. Muito obrigado.

Aos queridos amigos da equipe Supernova, onde pude me desenvolver não apenas tecnicamente, mas também pessoalmente, agradeço pelas risadas, pelos momentos tristes e felizes que vivemos juntos. Agradeço especialmente ao setor de Eletrônica da equipe, onde tive a oportunidade de liderar e aprender junto com os demais membros. Participar da equipe me abriu muitas portas, agradeço pela oportunidade de contribuir com o projeto.

A todos os meus amigos, obrigado por tornarem a experiência mais leve e pelo apoio. Agradeço especialmente à minha amiga Maria Fernanda, pelos conselhos, pelos desabafos, por dividir a admiração pela robótica e por contribuir também com este trabalho.

“Se eu vi mais longe, foi porque estava sobre os ombros de gigantes.”

Isaac Newton.

RESUMO

O trabalho desenvolvido propõe uma abordagem para a integração de técnicas de visão computacional com métodos de robótica clássica, visando a melhoria da qualidade de vida de pessoas com deficiência e idosos, proporcionando autonomia na tarefa de vestir. Utilizando um manipulador robótico Kinova Gen3 com sete graus de liberdade, o estudo aplica algoritmos de segmentação por cor, detecção de bordas e Transformada de Hough para identificar a posição do braço do paciente, aproximar o robô e gerar trajetórias seguras para a manipulação de vestimentas. Em ambiente simulado, com o auxílio do CoppeliaSim e de um sensor RGB-D, foram avaliadas diferentes configurações de posicionamento do braço, demonstrando a viabilidade do sistema em oferecer assistência personalizada e segura para pessoas com mobilidade reduzida. A integração dos métodos de visão computacional com o planejamento de trajetórias contribui significativamente para aumentar a autonomia dos usuários, apresentando uma alternativa promissora para a implementação em robótica assistiva.

Palavras-chave: Visão Computacional, Assistência ao Vestir, Robótica Assistiva, Manipuladores Robóticos

ABSTRACT

This work proposes an approach that integrates computer vision techniques with classical robotics methods, aiming to improve the quality of life for people with disabilities and the elderly by providing autonomy in dressing tasks. Using a Kinova Gen3 robotic manipulator with seven degrees of freedom, the study implements color segmentation, edge detection, and Hough transform algorithms to identify the patient's arm position, guide the robot approach, and generate safe trajectories for garment manipulation. In a simulated environment using CoppeliaSim and an RGB-D sensor, different arm positioning configurations were evaluated, demonstrating the system feasibility in offering personalized and secure assistance for individuals with reduced mobility. The integration of computer vision methods with trajectory planning significantly contributes to increasing user autonomy, presenting a promising alternative for assistive robotics implementation.

Keywords: Computer Vision, Dressing Assistance, Assistive Robotics, Robotic Manipulators.

LISTA DE ILUSTRAÇÕES

Figura 1 – Proposta da competição com estado inicial da simulação.	18
Figura 2 – Atividades comuns para o uso de robôs assistivos.	19
Figura 3 – Parametrização de uma reta no plano xy , representação no espaço de parâmetros $\rho\text{-}\theta$ e a subdivisão em células acumuladoras	33
Figura 4 – Configuração inicial do ambiente de simulação	34
Figura 5 – Configurações de posicionamento do braço do paciente	35
Figura 6 – Fluxograma do algoritmo de aproximação ao braço do paciente	37
Figura 7 – Exemplo de reta obtida para cálculo dos pontos objetivo	38
Figura 8 – Exemplo da imagem antes e após segmentação por cor	39
Figura 9 – Imagens obtidas nos passos da estratégia usando Transformada de Hough	40
Figura 10 – Imagens obtidas nos passos da estratégia usando Detecção de Contornos	42
Figura 11 – Dimensões e definições de <i>frames</i> do robô Kinova com 7 graus de liberdade	43
Figura 12 – Coordenadas do eixo z da garra ao longo de toda trajetória e simulação na configuração do braço inclinado 20° à esquerda e antebraço reto	46
Figura 13 – Coordenadas do eixo z da garra ao longo de toda trajetória e simulação na configuração do braço inclinado 20° à direita e antebraço 20° à esquerda	46
Figura 14 – Coordenadas do eixo z da garra ao longo de toda trajetória e simulação na configuração do braço reto e antebraço inclinado 20° à direita	47
Figura 15 – Imagens binárias da segmentação por cor do braço inclinado 20° à esquerda e antebraço reto: (a) antebraço; (b) braço	48
Figura 16 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20° à esquerda e antebraço reto usando a segmentação por cor	49
Figura 17 – Usando a segmentação por cor: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20° à esquerda e antebraço reto; (b) Trajetória seguida na simulação	49
Figura 18 – Imagens binárias da segmentação por cor do braço inclinado 20° à direita e antebraço 20° à esquerda: (a) antebraço; (b) braço	50
Figura 19 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20° à direita e antebraço 20° à esquerda usando a segmentação por cor	50
Figura 20 – Usando a segmentação por cor: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20° à direita e antebraço 20° à esquerda; (b) Trajetória seguida na simulação	50
Figura 21 – Imagens binárias da segmentação por cor do braço reto e antebraço inclinado 20° à direita: (a) antebraço; (b) braço	51

Figura 22 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita usando a segmentação por cor	51
Figura 23 – Usando a segmentação por cor: (a) Trajetória planejada em comparação com a trajetória executada com braço reto e antebraço inclinado 20º à direita; (b) Trajetória seguida na simulação	52
Figura 24 – Retas obtidas pela Transformada de Hough do braço inclinado 20º à esquerda e antebraço reto: (a) antebraço; (b) braço	53
Figura 25 – Retas finais obtidas pela Transformada de Hough do braço inclinado 20º à esquerda e antebraço reto: (a) antebraço; (b) braço	54
Figura 26 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita usando a Transformada de Hough	54
Figura 27 – Usando a Transformada de Hough: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à esquerda e antebraço reto; (b) Trajetória seguida na simulação	55
Figura 28 – Retas obtidas pela Transformada de Hough do braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço	55
Figura 29 – Retas finais obtidas pela Transformada de Hough do braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço	55
Figura 30 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20º à direita e antebraço 20º à esquerda	56
Figura 31 – Usando a Transformada de Hough: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à direita e antebraço 20º à esquerda; (b) Trajetória seguida na simulação	56
Figura 32 – Retas obtidas pela Transformada de Hough do braço reto e antebraço inclinado 20º à direita: (a) antebraço; (b) braço	57
Figura 33 – Retas finais obtidas pela Transformada de Hough do braço reto e antebraço inclinado 20º à direita: (a) antebraço; (b) braço	57
Figura 34 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita	58
Figura 35 – Usando a Transformada de Hough: (a) Trajetória planejada em comparação com a trajetória executada com braço reto e antebraço inclinado 20º à direita; (b) Trajetória seguida na simulação	58
Figura 36 – Contornos do lado esquerdo e direito detectados pelo algoritmo com braço inclinado 20º à esquerda e antebraço reto: (a) antebraço; (b) braço	60
Figura 37 – Retas finais obtidas pela detecção de contornos do braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço	60
Figura 38 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita usando a detecção de contornos	61

Figura 39 – Usando a detecção de contornos: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à esquerda e antebraço reto; (b) Trajetória seguida na simulação	61
Figura 40 – Contornos do lado esquerdo e direito detectados pelo algoritmo com braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço	62
Figura 41 – Retas finais obtidas pela detecção de contornos do braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço	62
Figura 42 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20º à direita e antebraço 20º à esquerda usando a detecção de contornos	62
Figura 43 – Usando a detecção de contornos: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à direita e antebraço 20º à esquerda; (b) Trajetória seguida na simulação	63
Figura 44 – Contornos do lado esquerdo e direito detectados pelo algoritmo com braço reto e antebraço inclinado 20º à direita: (a) antebraço; (b) braço	63
Figura 45 – Retas finais obtidas pela detecção de contornos do braço reto e antebraço inclinado 20º à direita: (a) antebraço; (b) braço	64
Figura 46 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita usando a detecção de contornos	64
Figura 47 – Usando a detecção de contornos: (a) Trajetória planejada em comparação com a trajetória executada com braço reto e antebraço inclinado 20º à direita; (b) Trajetória seguida na simulação	64

LISTA DE TABELAS

Tabela 1 – Parâmetros definidos inicialmente no algoritmo	36
Tabela 2 – Parâmetros de Denavit-Hartenberg do robô manipulador Kinova 7 DoF	43
Tabela 3 – Comparação da distância final após aproximação da garra ao braço e margem de segurança.	47
Tabela 4 – Erro entre trajetória planejada e ideal em todas configurações utilizando segmentação por cor.	52
Tabela 5 – Erro entre trajetória planejada e executada pelo manipulador robótico em todas configurações utilizando segmentação por cor.	53
Tabela 6 – Erro entre trajetória planejada e ideal em todas configurações utilizando a Transformada de Hough.	59
Tabela 7 – Erro entre trajetória planejada e executada pelo manipulador robótico em todas configurações utilizando a Transformada de Hough.	59
Tabela 8 – Erro entre trajetória planejada e ideal em todas configurações utilizando a detecção de contornos.	65
Tabela 9 – Erro entre trajetória planejada e executada pelo manipulador robótico em todas configurações utilizando a detecção de contornos.	65
Tabela 10 – Comparação entre o RMSE de cada estratégia adotada para vestir o paciente, considerando diferentes configurações de braço.	66

LISTA DE ABREVIATURAS E SIGLAS

IBGE	Instituto Brasileiro de Geografia e Estatística
PNS	Pesquisa Nacional de Saúde
OMS	Organização Mundial da Saúde
IHM	Interface Humano Máquina
WMRA	<i>Wheelchair Mounted Robotic Arm</i> (Braço Robótico montado em cadeira de rodas)
PcD	Pessoa com Deficiência
DoF	<i>Degrees of Freedom</i> (Graus de Liberdade)
CNN	<i>Convolutional Neural Network</i> (Rede Neural Convolucional)
RGB	<i>Red, Green, Blue</i> (Vermelho, Verde, Azul)
CMYK	<i>Cyan, Magenta, Yellow, Black</i> (Ciano, Magenta, Amarelo, Preto)
HSV	<i>Hue, Saturation, Value</i> (Matiz, Saturação, Valor)
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
RMSE	<i>Root Mean Squared Error</i> (Raiz do erro quadrático médio)

LISTA DE SÍMBOLOS

∂ Derivada Parcial

∇ Gradiente

SUMÁRIO

1	INTRODUÇÃO	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Robótica Assistiva	19
2.2	Robótica na Assistência ao Vestir	21
2.3	Visão Computacional	24
2.3.1	Digitalização de Imagens e Filtragem	25
2.3.2	Segmentação por cor	27
2.3.3	Detecção de bordas	30
2.3.4	Transformada de Hough	31
3	METODOLOGIA	34
3.1	Configuração da Simulação	34
3.2	Estratégias adotadas	36
3.2.1	Estratégia baseada em Segmentação por Cor	38
3.2.2	Estratégia baseada na Transformada de Hough	39
3.2.3	Estratégia baseada em Detecção de Contornos	41
3.3	Cinemática do Manipulador	42
4	RESULTADOS	45
4.1	Resultados da aproximação ao braço	45
4.2	Resultados da geração de trajetórias	47
4.2.1	Estratégia usando Segmentação por Cor	48
4.2.1.1	Braço inclinado 20° à esquerda e antebraço reto	48
4.2.1.2	Braço inclinado 20° à direita e antebraço 20° à esquerda	49
4.2.1.3	Braço reto e antebraço inclinado 20° à direita	51
4.2.1.4	Comparativo entre configurações	52
4.2.2	Estratégia usando Transformada de Hough	53
4.2.2.1	Braço inclinado 20° à esquerda e antebraço reto	53
4.2.2.2	Braço inclinado 20° à direita e antebraço 20° à esquerda	54
4.2.2.3	Braço reto e antebraço inclinado 20° à direita	57
4.2.2.4	Comparativo entre configurações	58
4.2.3	Estratégia usando Detecção de Contornos	59
4.2.3.1	Braço inclinado 20° à esquerda e antebraço reto	59
4.2.3.2	Braço inclinado 20° à direita e antebraço 20° à esquerda	60
4.2.3.3	Braço reto e antebraço inclinado 20° à direita	63

4.2.3.4	Comparativo entre configurações	65
4.3	Análise Comparativa das Estratégias	65
5	CONCLUSÕES	67
5.1	Conclusões	67
5.2	Trabalhos Futuros	68
	REFERÊNCIAS	69

1 INTRODUÇÃO

De acordo com a Pesquisa Nacional de Saúde (PNS) realizada em 2019 ([IBGE, 2021](#)), o Brasil possui aproximadamente 7,8 milhões de pessoas com deficiência física nos membros inferiores, representando 3,8% da população com 2 anos ou mais de idade. Além disso, a incidência aumenta significativamente com a idade, atingindo 14,4% entre os idosos com 60 anos ou mais. Tarefas cotidianas como se vestir, manusear objetos e se locomover podem se tornar desafiadoras, impactando a independência e a qualidade de vida desses indivíduos. Diante desse cenário, a tecnologia assistiva se torna essencial para promover a inclusão e melhorar a qualidade de vida dessas pessoas, possibilitando maior mobilidade.

Entre as diversas abordagens da tecnologia assistiva, a robótica assistiva tem se destacado como um campo promissor, como discutido em ([INTELLIGENCE, 2023](#)), voltado para o desenvolvimento de dispositivos que auxiliam indivíduos com deficiência física. Utilizando os princípios da robótica, essa área busca criar sistemas inteligentes capazes de interagir de forma segura e eficiente com os usuários, promovendo maior autonomia e qualidade de vida.

Vestir-se é uma das atividades diárias mais comuns em que pessoas com deficiência necessitam de assistência. Além disso, cuidadores relatam que auxiliar no vestuário dos pacientes é a tarefa de maior carga e a menos automatizada, conforme indicado em ([DUDGEON et al., 2008](#)) e ([MITZNER et al., 2014](#)). Nesse contexto, a introdução da robótica nessa tarefa pode melhorar significativamente a qualidade de vida de muitas pessoas, embora ainda represente um grande desafio para os robôs devido à complexidade da manipulação de roupas, materiais deformáveis e à necessidade de interação segura com os usuários.

Este trabalho foi inspirado pelo desafio proposto na competição internacional *PhyRC Challenge* voltada para a manipulação em tarefas de vestimenta assistida por robôs, promovida pela Universidade de Cornell ([EMPRISELAB, 2024](#)). Como ilustrado na Figura 1, o desafio, que utilizou o simulador *RCareWorld* ([YE et al., 2022](#)) como ferramenta de avaliação na primeira fase, incentivou o desenvolvimento de soluções inovadoras para a manipulação de roupas, enfrentando desafios como a manipulação de materiais deformáveis e a garantia de uma interação segura com os usuários. Essa iniciativa ressalta a importância do investimento em pesquisas que ampliem a autonomia e melhorem a qualidade de vida das pessoas com deficiência, servindo de estímulo para o presente estudo.

Visão computacional é um campo da inteligência artificial que tem como objetivo capacitar máquinas a interpretar e compreender o conteúdo de imagens e vídeos do mundo real, utilizando algoritmos que empregam técnicas de processamento digital de imagens, extração de características e aprendizado de máquina para permitir a identificação, segmentação e classificação de objetos em diferentes cenários, conforme discutido em

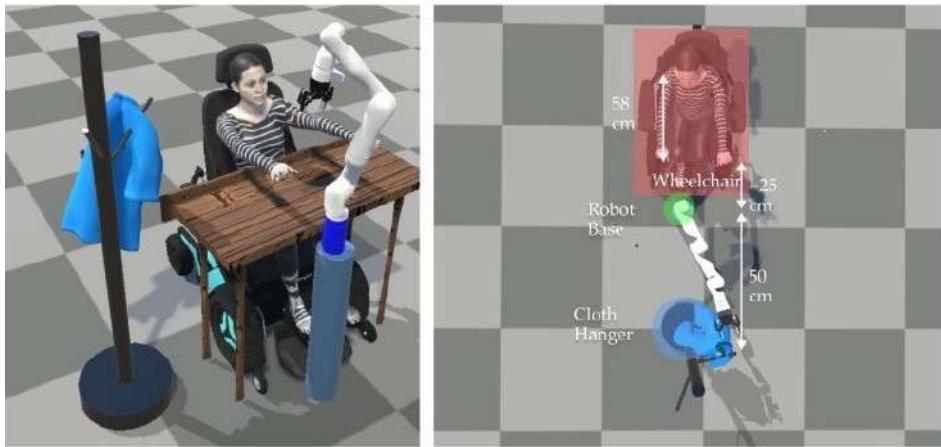


Figura 1 – Proposta da competição com estado inicial da simulação.

Fonte: ([EMPRISELAB, 2024](#))

([Szeliski, 2022](#)). Essa abordagem tem se mostrado ainda mais relevante quando integrada à robótica assistiva, onde a aplicação da visão computacional se torna fundamental para aprimorar a interação dos robôs com os pacientes em tarefas complexas, como a manipulação de roupas, contribuindo para uma assistência mais segura e eficiente.

Com o objetivo de auxiliar pessoas com deficiência, idosos e indivíduos com mobilidade reduzida em tarefas cotidianas, como o ato de se vestir, este trabalho visa utilizar algoritmos de visão computacional para gerar trajetórias seguras e precisas a serem seguidas pelo manipulador robótico, com base nos princípios da robótica clássica. Para isso, será realizada uma análise da configuração do braço do paciente, que estará estendido para receber a manga da vestimenta, permitindo o planejamento de um trajeto otimizado e seguro durante a manipulação.

Este trabalho está estruturado em cinco capítulos. O Capítulo 1 corresponde à introdução, na qual são apresentadas as justificativas para o estudo e os objetivos estabelecidos a partir do tema proposto. No Capítulo 2, é realizada a fundamentação teórica, abordando os principais conceitos e referências que sustentam a pesquisa. O Capítulo 3 descreve a metodologia do trabalho, detalhando as simulações realizadas. No Capítulo 4, são apresentados e analisados os resultados obtidos a partir dessas simulações. Por fim, no Capítulo 5, são expostas as conclusões do estudo, destacando as principais contribuições do trabalho e sugerindo possíveis melhorias para pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA

Esse Capítulo tem como objetivo fazer uma breve revisão bibliográfica dos temas abrangidos por este trabalho. Dessa forma, são apresentados trabalhos anteriores envolvendo os seguintes temas: Robótica Assistiva, Robótica na Assistência ao Vestir e Visão Computacional.

2.1 Robótica Assistiva

De acordo com dados da Organização Mundial da Saúde (OMS), estima-se que 1,3 bilhão de pessoas no mundo vivam com uma deficiência significativa ([WHO, 2023](#)). Embora a maioria delas deseje manter sua independência, as dificuldades enfrentadas nas tarefas cotidianas podem comprometer essa autonomia. A robótica de manipuladores e móvel, por meio de sua capacidade de sensoriamento, navegação e manipulação de objetos, pode viabilizar tarefas cotidianas como se alimentar, caminhar e se vestir conforme mostrado na Figura 2.



Figura 2 – Atividades comuns para o uso de robôs assistivos.

Fonte: Adaptada de ([NANAVATI; RANGANENI; CAKMAK, 2023](#))

Para viabilizar a assistência nessas tarefas, os robôs precisam interagir de maneira robusta e segura com o ambiente, integrando sensores tátteis, de movimento e câmeras para aplicação de Visão Computacional.

Nesse cenário de utilização da robótica assistiva, é possível dividi-la em duas categorias diferentes: os robôs de base fixa e os robôs de base móvel. Dessa forma, cada tipo de robô oferece assistência em tarefas específicas, como ilustrado na Figura 2, ampliando as possibilidades e permitindo até mesmo configurações híbridas, como a combinação de robôs móveis com manipuladores integrados. Dentro da área de manipuladores robóticos, ([ALLIN et al., 2010](#)) destaca três categorias principais: os manipuladores conectados a cadeiras de rodas, os braços montados em mesas, e os manipuladores que se movem autonomamente pela sala em suas próprias bases móveis.

Na década de 1990, uma pesquisa significativa em robótica assistiva resultou no desenvolvimento do ProVAR ([LOOS et al., 1999](#)), um manipulador robótico montado sobre uma mesa. Projetado para auxiliar pessoas tetraplégicas, o ProVAR aumentava a autonomia dos usuários ao permitir que, por meio de uma interface humano-máquina (IHM), eles enviassem comandos para buscar e entregar objetos. Seu controle podia ser efetuado tanto por comandos de voz quanto pelos movimentos da cabeça, demonstrando a versatilidade das tecnologias assistivas da época.

Durante essa década, também foram realizados avanços significativos em outras configurações, como a implementação de manipuladores robóticos montados em cadeiras de rodas (WMRA - *Wheelchair Mounted Robotic Arm*), como o MANUS discutido em ([VERBURG et al., 1996](#)). Esses sistemas oferecem a vantagem de acompanhar o usuário em uma variedade de ambientes — desde residências e supermercados até escritórios. Ao longo do tempo, essa configuração tem sido continuamente aprimorada, com constantes melhorias em *design* e usabilidade ([KIM et al., 2014](#)), além de sua comercialização, o que tem desempenhado um papel fundamental em atividades cotidianas e proporcionado maior autonomia às pessoas com deficiência (PcD).

De acordo com ([CHUNG; WANG; COOPER, 2013](#)), o *design* centrado no usuário envolve definir e avaliar os requisitos de projeto com a participação dos usuários finais, visando replicar a funcionalidade do braço humano. Para isso, a maioria dos manipuladores robóticos assistivos conta com sete graus de liberdade (DoF), correspondentes a sete juntas rotacionais, incluindo a garra. Cada uma dessas juntas permite a rotação de um segmento do braço, possibilitando movimentos em diversas direções e reproduzindo a mobilidade dos membros superiores. Essa ênfase nos graus de liberdade é fundamental para garantir flexibilidade e precisão nos movimentos, além de melhorar a usabilidade, facilitando que os usuários realizem tarefas de forma eficiente e em um tempo adequado.

Hoje, graças ao uso de sensores modernos e inteligência artificial, a manipulação robótica evoluiu muito. Esses avanços permitem que os robôs capturem imagens com mais qualidade e processem as informações rapidamente, ajudando-os a entender melhor o ambiente e a agir com maior precisão. Embora ainda existam desafios para que os robôs atinjam a destreza dos humanos, as novas tecnologias apontam para um futuro promissor na robótica assistiva.

O presente trabalho tem por objetivo empregar um manipulador robótico Kinova Gen3 ([KINOVA, 2024](#)), dotado de sete graus de liberdade, para replicar os movimentos de um braço humano na tarefa de vestir um paciente posicionado à sua frente, por meio do uso de técnicas de visão computacional. O manipulador será instalado sobre uma superfície de modo a garantir que o paciente se encontre dentro de seu espaço de trabalho (*workspace*).

2.2 Robótica na Assistência ao Vestir

Quando se analisa a capacidade de robôs manipuladores para auxiliar no ato de vestir um paciente, é fundamental compreender os desafios e requisitos do projeto. A segurança é um aspecto crítico dessa tarefa. Em ([BELLAMY et al., 2021](#)), é realizada uma Análise de Perigos (*Hazard Analysis*) em uma aplicação de assistência ao vestuário, destacando riscos de médio e alto impacto, tais como:

- Colisão, atrito ou tração entre o usuário e o robô devido à perda de equilíbrio ou ao emaranhamento com a vestimenta.
- Obstrução ou bloqueio da trajetória do robô pelo usuário ou por terceiros, afetando o desempenho do robô e/ou resultando em colisões.
- Natureza dinâmica das ações do usuário, incluindo movimentos súbitos, distrações, mudanças de intenção, confusão e falhas de comunicação.

Um *design* seguro exige a compreensão dos riscos envolvidos e das interações entre robô, usuário e vestimenta. A integração entre segurança e planejamento de trajetória é essencial, pois a detecção desses riscos permite que os algoritmos ajustem dinamicamente os movimentos, evitando colisões e minimizando impactos. Assim, técnicas avançadas de monitoramento e planejamento trabalham em conjunto para tornar a assistência ao vestir mais segura e eficiente.

Além dos desafios de segurança mencionados, diversas abordagens na literatura têm buscado aprimorar a robótica na assistência ao vestir por meio da integração de técnicas de cinemática e visão. Pesquisadores que investigaram essa temática concentraram seus esforços na modelagem do usuário e na definição de trajetórias seguras para os manipuladores. Por exemplo, ([GAO; CHANG; DEMIRIS, 2015](#)) desenvolveram uma metodologia focada na identificação visual da pose do usuário e na modelagem do espaço de movimento das articulações superiores, permitindo determinar os pontos ideais para posicionar as aberturas de um casaco sem mangas, contribuindo para a personalização.

Em outra vertente, ([TAMEI et al., 2011](#)) propuseram um método baseado em aprendizado por reforço, no qual o robô aprende trajetórias para vestir um manequim com uma camiseta, enfatizando as relações entre o manequim e a peça de roupa, tendo os experimentos iniciado com os braços do manequim já inseridos nas mangas. Vale ressaltar que, por se tratar de um método de aprendizado por reforço, são necessárias diversas tentativas e erros, o que pode expor o usuário a riscos e por isso foi utilizado um manequim.

De forma complementar, ([KLEE et al., 2015](#)) apresentaram uma abordagem na qual um manipulador robótico realiza a tarefa de assistência ao vestir por meio da definição de uma sequência de poses-alvo relativas ao usuário. Nesta proposta, a utilização de um

robô manipulador Baxter ([RETHINK, 2022](#)) para posicionar um chapéu em participantes demonstrou que, ao identificar quando uma meta se torna inviável, o sistema pode solicitar o reposicionamento do usuário, modelando suas restrições para determinar a melhor configuração. Essas contribuições evidenciam que, para um *design* seguro e eficaz, é imprescindível considerar tanto os aspectos cinematográficos quanto as particularidades do movimento e das limitações dos usuários na formulação de estratégias assistivas.

Além dessas abordagens, ([KAPUSTA et al., 2016](#)) destaca o uso de técnicas de percepção háptica orientada por dados para aumentar a segurança e eficácia. Esse método possibilita a detecção em tempo real das interações sutis entre o paciente e a vestimenta, permitindo que o robô identifique variações nas forças aplicadas e ajuste dinamicamente sua atuação para prevenir riscos, como colisões e enganches inesperados. Com isso, ao incorporar algoritmos de aprendizado, como modelos de Markov ocultos, esses sistemas podem antecipar os desdobramentos do movimento, reconhecendo, por exemplo, se a manga será posicionada corretamente ou se há potencial para um engasgo entre a roupa e o usuário.

Essas abordagens evidenciam a importância de integrar múltiplas fontes sensoriais para aprimorar a eficácia e segurança na assistência ao vestir. A combinação de dados visuais e hápticos permite que o sistema compreenda de forma mais completa as interações entre o robô, o paciente e a vestimenta, ajustando sua atuação de acordo com as particularidades de cada situação.

Adicionalmente, estudos recentes na área de robótica na assistência ao vestir geralmente descrevem o problema como uma tarefa de planejamento de trajetória. No entanto, oclusões severas podem ocorrer em tempo real quando o manipulador, a roupa e a pessoa estão em contato próximo, conforme discutido em ([ZHANG; CULLY; DEMIRIS, 2017](#)). O trabalho de ([LI et al., 2021](#)) exemplifica uma abordagem que trata o problema como um planejamento de trajetórias, visando aprimorar a eficiência do robô. Em vez de adotar uma estratégia excessivamente conservadora, o método proposto integra a predição do movimento humano para permitir um planejamento mais eficiente. Nesse contexto, a segurança é definida como a capacidade de evitar colisões (*collision avoidance*) ou, caso uma colisão seja inevitável, garantir um impacto seguro (*safe impact*).

Outras abordagens envolvem métodos iterativos e *online* de otimização de trajetórias, através de dados de visão e força. Em ([GAO; CHANG; DEMIRIS, 2016](#)), os autores propõem um método iterativo de otimização de trajetórias para assistência personalizada no vestir. Inicialmente, uma câmera de profundidade é empregada para modelar o espaço de movimento dos membros superiores do usuário e definir uma trajetória inicial para a ação de vestir. Durante a execução, sensores de força detectam resistências externas que indicam desconforto ou um caminho inadequado. Com base nessas medições, o algoritmo utiliza uma variante do otimizador ADAM para atualizar iterativamente os pontos da

trajetória, ajustando-os de forma a reduzir a resistência percebida. Esse processo continua até que a resistência seja minimizada, sinalizando que o caminho foi otimizado para o conforto e a segurança do usuário.

Além dos desafios já mencionados, a determinação dos *grasping points* – pontos ideais de preensão na vestimenta – constitui um problema crítico na tarefa, conforme discutido em ([ZHANG; DEMIRIS, 2020](#)). Esse desafio advém principalmente da natureza altamente deformável e variável dos tecidos, que podem assumir diferentes configurações e apresentar características visuais distintas, como variações em textura e cor. A identificação precisa desses pontos é essencial para garantir uma manipulação segura e eficaz, pois uma preensão inadequada pode resultar em erros de manipulação, colisões ou até mesmo riscos à segurança do paciente. Dessa forma, a definição e extração automática dos *grasping points* exigem a integração de técnicas avançadas de percepção, modelagem geométrica e algoritmos de aprendizado, permitindo ao sistema adaptar-se a diferentes condições e tipos de vestimentas.

O artigo ([ZHANG; DEMIRIS, 2020](#)) propõe uma abordagem inovadora para o problema dos *grasping points* em cenários de robótica na assistência ao vestir. A metodologia se baseia em uma rede neural convolucional (CNN) supervisionada que mapeia imagens de profundidade para a estimativa das coordenadas de um ponto de preensão pré-definido na vestimenta. Para minimizar a necessidade de grandes quantidades de dados reais, a estratégia combina dados sintéticos, gerados via simulação, com um conjunto limitado de dados reais, promovendo uma transferência eficaz do conhecimento adquirido em simulação para o mundo real. Além disso, o sistema inclui um módulo de cálculo da orientação de preensão a partir da nuvem de pontos da vestimenta, permitindo que o robô planeje uma trajetória de aproximação que evita colisões e garanta uma preensão correta. Essa abordagem integrada demonstra melhorias significativas na precisão da predição dos pontos de preensão e, consequentemente, na eficácia global do sistema de assistência ao vestir.

Neste trabalho, a configuração inicial do robô é simplificada para as sequências de vestir, por meio da fixação da vestimenta no efetuador final (*end-effector*), conforme realizado em ([ERICKSON et al., 2018](#)). Essa estratégia reduz a complexidade dos procedimentos iniciais ao eliminar a necessidade de um posicionamento preciso da roupa, permitindo que o foco seja direcionado ao desenvolvimento dos algoritmos de visão e planejamento de trajetórias.

A partir das abordagens discutidas, o presente trabalho busca reduzir a complexidade do processo de vestir sem comprometer a segurança do paciente. Para isso, propõe-se detectar a posição inicial do usuário por meio de uma câmera acoplada ao manipulador, dividindo a operação em duas fases: da região do punho ao cotovelo e do cotovelo ao ombro. Inicialmente, a câmera equipada com sensor de profundidade é empregada para

aproximar a vestimenta do braço, mantendo uma distância segura que previne colisões.

Nesse contexto, os algoritmos de visão computacional aliados aos dados do sensor desempenham um papel fundamental, permitindo a identificação precisa dos pontos de referência e o planejamento de trajetórias adequadas para a manipulação autônoma e segura da roupa. Essa estratégia visa mitigar os riscos de colisão e erros de manipulação identificados na literatura, garantindo um controle mais preciso da vestimenta ao longo do processo sem aumentar significativamente a complexidade do sistema.

2.3 Visão Computacional

A visão computacional é um campo da inteligência artificial que busca descrever e interpretar o mundo visual por meio de imagens digitais, permitindo que máquinas extraiam informações relevantes sobre os objetos presentes em uma cena. Isso envolve a inferência e reconstrução de propriedades dos objetos, como forma, cor, textura e profundidade. Como o próprio nome sugere, o objetivo é reproduzir computacionalmente capacidades da visão humana, incluindo reconhecimento de padrões, detecção de objetos, percepção de movimento e estimativa de profundidade. Embora essas tarefas sejam intuitivas para os seres humanos, replicá-las em computadores é um grande desafio, exigindo algoritmos avançados, modelos matemáticos sofisticados e técnicas de aprendizado de máquina para interpretar corretamente o ambiente visual.

A história da visão computacional teve seu início na década de 1970, quando as primeiras pesquisas na área começaram a ganhar destaque. Um marco importante foi o trabalho de Marvin Minsky, no MIT, que desafiou seu estudante Gerald Jay Sussman a conectar uma câmera a um computador e fazer com que a máquina descrevesse o que via. Além disso, um dos desafios centrais da época era recuperar a estrutura tridimensional a partir de imagens, possibilitando uma compreensão mais profunda das cenas capturadas. Esses esforços iniciais estabeleceram as bases para o desenvolvimento da visão computacional moderna ([Szeliski, 2022](#)).

Na década de 1980, a visão computacional avançou com algoritmos mais sofisticados, introduzindo pirâmides de imagens, detecção de contornos e fluxo óptico para análise de movimento ([Man; Vision, 1982; Horn; Schunck, 1981](#)). Nos anos 1990, essas técnicas evoluíram, permitindo reconstrução *multiview*, reconhecimento de faces, segmentação de imagens e maior integração com a computação gráfica para modelagem e renderização realistas ([Forsyth; Ponce, 2003; Szeliski, 2022](#)).

Na década de 2000, a visão computacional evoluiu com abordagens de aprendizado que possibilitaram a extração e descritores de características (*features*), aprimorando a detecção e classificação de objetos ([Viola; Jones, 2001; Lowe, 2004](#)). Já na década de 2010, a área passou por uma transformação com a adoção do *deep learning*, impulsionada

por grandes *datasets* de alta qualidade, uso intensivo de GPUs e a introdução de sensores especializados, como o *Microsoft Kinect*, que ampliaram as aplicações e a eficiência dos sistemas de análise visual (KRIZHEVSKY; SUTSKEVER; HINTON, 2012; LECUN; BENGIO; HINTON, 2015; SHOTTON et al., 2011).

A seguir, serão abordadas as principais técnicas de visão computacional e processamento digital de imagens essenciais para a realização da tarefa.

2.3.1 Digitalização de Imagens e Filtragem

No contexto da visão computacional, uma das principais etapas para a formação de imagens é a digitalização e o processamento das imagens obtidas, conforme discutido em (GONZALEZ; WOODS, 2018). Nessa etapa, uma imagem é convertida para um formato digital que possa ser interpretado por um computador. Ao final desse processo, a imagem digital pode ser vista como uma grande matriz de pixels de tamanho $M \times N$, conforme a Equação 2.1, na qual cada pixel contém um valor correspondente a uma cor específica. No caso de imagens no formato RGB, essa matriz pode ser desmembrada em três matrizes individuais (ou representada como uma matriz tridimensional), uma para cada canal de cor: vermelho, verde e azul. Essa representação digital é fundamental para as técnicas de análise e processamento de imagens que possibilitam a extração de informações e a compreensão do cenário observado.

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

Com a imagem digitalizada, o próximo passo é a aplicação de técnicas de filtragem espacial, que operam diretamente sobre a matriz de pixels utilizando o conceito de vizinhança. Considerando a estrutura matricial de uma imagem, a vizinhança de um pixel corresponde aos pixels em posições próximas, tanto em termos de linhas quanto de colunas. Por exemplo, em uma 4-vizinhança, dado o pixel (x, y) , seus vizinhos são:

$$(x - 1, y), (x, y - 1), (x, y + 1), (x + 1, y) \quad (2.2)$$

Já em uma 8-vizinhança, além desses, incluem-se os pixels diagonais:

$$(x - 1, y - 1), (x, y - 1), (x + 1, y - 1), (x - 1, y + 1), (x, y + 1), (x + 1, y + 1) \quad (2.3)$$

Essa relação de vizinhança forma a base para a filtragem no domínio do espaço.

Para definir qual conjunto de pixels (vizinhança) será considerado na operação de filtragem, utiliza-se um *kernel* (também conhecido como máscara, *template* ou janela),

que é representado por uma matriz de coeficientes. Geralmente, são utilizadas matrizes quadradas de dimensão ímpar (por exemplo, 3×3 , 5×5 , 7×7 , etc.), onde cada elemento possui um valor que determina o peso do pixel correspondente na imagem.

O mecanismo de filtragem linear é implementado por meio da operação de convolução, que combina os valores dos pixels da imagem com os coeficientes do *kernel*. Considerando um *kernel* 3×3 , a resposta do filtro na posição (x, y) , denotada por $g(x, y)$, é dada pela Equação 2.4.

$$g(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 f(x + i, y + j) h(i, j) \quad (2.4)$$

Na aplicação de filtros, é comum adicionar pixels na borda da imagem para que o *kernel* possa ser aplicado a todos os pontos sem reduzir a dimensão da imagem resultante. Esse processo, chamado de *padding*, geralmente utiliza preenchimento constante (com zeros sendo a forma mais comum), mas também pode empregar técnicas como o espelhamento, a replicação dos pixels da borda ou até mesmo o preenchimento circular. Essas abordagens garantem que a aplicação do filtro seja realizada de forma consistente, preservando as dimensões originais da imagem e evitando a perda de informação nas regiões periféricas.

No processamento de imagens, a análise pode ser feita no domínio do espaço, onde a imagem é uma matriz de pixels, ou no domínio da frequência, onde é descrita como a soma de senoides de diferentes frequências e intensidades. A Transformada de Fourier permite essa conversão, sendo essencial para técnicas de filtragem. Termos como passa-baixa e passa-alta estão relacionados à manipulação de baixas e altas frequências na imagem.

No contexto da utilização de filtros, os filtros de suavização desempenham um papel crucial no processamento de imagens, atuando diretamente no domínio espacial para atenuar transições bruscas de intensidade. Esses filtros são amplamente utilizados para a remoção de ruídos, eliminando variações indesejadas que podem prejudicar a análise da imagem, e para suprimir detalhes que não sejam relevantes para a aplicação em questão. O efeito resultante é um borramento (*blur*) que facilita a extração de características mais importantes e robustas para tarefas subsequentes.

O filtro Gaussiano é um exemplo clássico de filtro de suavização, onde os coeficientes do *kernel* são definidos por uma função Gaussiana, expressa por:

$$h(s, t) = G(s, t) = K e^{-\frac{s^2+t^2}{2\sigma^2}} \quad (2.5)$$

Nesse contexto, σ representa o desvio padrão da distribuição Gaussiana e determina a extensão do desfoque aplicado, enquanto K é uma constante de normalização. Os valores dos coeficientes são maiores para os elementos próximos à origem, o que assegura que os pixels centrais tenham maior influência no processo de suavização, e diminuem

gradativamente à medida que se afastam do centro. Essa característica resulta em um efeito de borramento (*blur*) que suaviza transições bruscas de intensidade, contribuindo para a remoção de ruídos e a preservação das características essenciais da imagem.

Conforme abordado em (GONZALEZ; WOODS, 2018), os filtros passa-altas atuam no processo de aguçamento (*sharpening*), cujo objetivo é realçar detalhes e enfatizar as transições de intensidade presentes em bordas, linhas e curvas. Essa técnica fundamenta-se na aplicação de filtros derivativos, como os de Sobel, Prewitt e Laplaciano, que medem as variações de intensidade entre pixels vizinhos, resultando em imagens com maior definição e contornos mais evidentes.

2.3.2 Segmentação por cor

Outra técnica importante para o desenvolvimento deste trabalho é a segmentação por cor, pois permite a identificação e o isolamento de regiões de interesse com base nas características cromáticas das imagens. Essa capacidade de distinguir regiões homogêneas não só facilita a extração de características relevantes, mas também aprimora a precisão dos algoritmos de reconhecimento e classificação.

Inicialmente, para o estudo da segmentação por cor, é fundamental compreender os diferentes espaços de cores e suas características. As cores que são percebidas pelas pessoas resultam da interação entre fontes de luz e os diversos materiais existentes no mundo físico, o que determina as tonalidades observadas em cada situação. Nesse contexto, destacam-se dois sistemas de cores principais: os sistemas aditivos e os sistemas subtrativos.

O sistema aditivo RGB baseia-se na combinação de três cores primárias – vermelho, verde e azul – que, ao serem misturadas em diferentes intensidades, originam todas as demais cores. Cada uma dessas cores é denominada canal, de modo que a presença simultânea de todos os canais em sua intensidade máxima resulta no branco, enquanto sua ausência total gera o preto. Inspirado no funcionamento do sistema visual humano, o modelo RGB reflete a sensibilidade dos cones oculares, que são responsáveis por captar comprimentos de onda curto (azul), médio (verde) e longo (vermelho), em contraste com os bastonetes, que operam em condições de baixa luminosidade e não distinguem cores. Em termos práticos, cada pixel de uma imagem pode ser representado por uma combinação linear das cores primárias, expressa matematicamente pela Equação 2.6.

$$C = r \cdot R + g \cdot G + b \cdot B \quad (2.6)$$

Onde r, g e b são os coeficientes de mistura de cada canal. Essa representação é frequentemente expressa na forma de uma tripla de valores, comumente utilizando 8 bits por canal, variando de 0 a 255, ou por meio do sistema hexadecimal, onde os valores vão de 00 a FF (GONZALEZ; WOODS, 2018).

No modelo subtrativo CMYK, utilizado em sistemas de impressão, as cores não são produzidas através da emissão de luz, mas sim pela absorção das frequências da luz branca. Nesse processo, o que não é absorvido é refletido, determinando a cor percebida pelo olho humano, de forma que o branco corresponde à ausência de pigmento e o preto resulta da sobreposição de todos eles. Para representar as cores de maneira eficaz, são escolhidas cores primárias, sendo o sistema mais comum baseado no ciano (C), magenta (M) e amarelo (Y). A escolha do ciano, magenta e amarelo deve-se ao fato de que a combinação, tomada duas a duas, reproduz as cores do sistema RGB, evidenciando a complementaridade entre os modelos aditivo e subtrutivo.

Quando um objeto colorido é observado, sua percepção baseia-se em três atributos fundamentais: matiz (*hue*), saturação e valor. O matiz refere-se à tonalidade pura da cor, como vermelho, azul ou amarelo. A saturação representa o grau de pureza da cor, ou seja, o quanto ela está diluída pela presença de luz branca. Já o valor corresponde à intensidade luminosa, permitindo a distinção de variações de brilho na imagem. O espaço de cores HSV organiza essas características de maneira a separar a informação cromática (matiz e saturação) da intensidade luminosa, proporcionando uma descrição mais alinhada com a percepção visual humana. Essa abordagem torna-se uma ferramenta essencial para o desenvolvimento de algoritmos de processamento de imagens que necessitam identificar e segmentar cores de forma natural e eficiente.

De acordo com (SMITH, 1978), a conversão do espaço de cor RGB para o espaço HSV pode ser realizada utilizando o modelo hexagonal (*hexcone*), que separa as informações cromáticas (matiz e saturação) da intensidade luminosa (valor). Supondo que os valores de R , G e B estejam normalizados no intervalo $[0, 1]$, definem-se:

$$V = \max(R, G, B), \quad X = \min(R, G, B), \quad (2.7)$$

$$\Delta = V - X. \quad (2.8)$$

O componente valor (V) representa a luminosidade da cor. A saturação (S), que indica o grau de pureza da cor, é calculada por:

$$S = \begin{cases} 0, & \text{se } V = 0, \\ \frac{\Delta}{V}, & \text{caso contrário.} \end{cases} \quad (2.9)$$

Quando $S = 0$, a cor é acromática (variações de cinza) e o matiz (H) pode ser atribuído arbitrariamente (por convenção, $H = 0$). Para cores saturadas ($S \neq 0$), procede-se ao cálculo do matiz (H) conforme o modelo *hexcone*, seguindo os passos abaixo:

1. Cálculo dos parâmetros auxiliares:

$$r = \frac{V - R}{\Delta}, \quad g = \frac{V - G}{\Delta}, \quad b = \frac{V - B}{\Delta}. \quad (2.10)$$

2. Determinação do matiz conforme o canal dominante:

Utilizando uma abordagem baseada em parâmetros auxiliares, a matiz é definida da seguinte forma:

$$\begin{aligned} \text{Se } R = V : \quad H &= \begin{cases} 5 + b, & \text{se } G = X, \\ 1 - g, & \text{caso contrário;} \end{cases} \\ \text{Se } G = V : \quad H &= \begin{cases} 1 + r, & \text{se } B = X, \\ 3 - b, & \text{caso contrário;} \end{cases} \\ \text{Se } B = V : \quad H &= \begin{cases} 3 + g, & \text{se } R = X, \\ 5 - r, & \text{caso contrário.} \end{cases} \end{aligned}$$

3. Normalização do matiz: O valor de H é então normalizado para o intervalo $[0, 1]$ dividindo-o por 6:

$$H = \frac{H}{6}. \quad (2.11)$$

Por convenção, $H = 0$ corresponde à cor vermelha.

Este método evita o uso de funções trigonométricas, proporcionando uma conversão eficiente, especialmente em aplicações computacionais de processamento de imagens e gráficos.

Neste trabalho, a conversão das imagens para o espaço HSV é fundamental, pois permite isolar a componente de matiz, utilizada como critério de limiar para identificar o braço do usuário na simulação com a câmera devidamente posicionada. Essa abordagem torna a segmentação por cor mais robusta, já que o espaço HSV minimiza os efeitos de variações de iluminação, facilitando a extração precisa da região de interesse. Consequentemente, essa segmentação contribui para a determinação e geração de uma trajetória acurada a ser seguida pelo manipulador robótico.

Em aplicações práticas, a variação da iluminação desempenha um papel decisivo na eficácia da segmentação por cor. Alterações na intensidade, no ângulo e na temperatura da luz podem modificar significativamente a aparência das cores capturadas, influenciando tanto os valores dos canais RGB quanto as representações em espaços como o HSV. Essa variabilidade pode levar a desafios na identificação precisa de regiões de interesse, sobretudo em ambientes com iluminação mista ou não controlada, onde sombras e reflexos alteram as tonalidades reais dos objetos. Dessa forma, a adoção de estratégias de pré-processamento para normalização da iluminação e o uso de algoritmos que incorporem características invariantes à luz são fundamentais para garantir uma segmentação mais robusta e confiável em condições reais ([MAITLO et al., 2024](#)).

2.3.3 Detecção de bordas

Para ajudar na definição da trajetória que o robô deve seguir sobre o braço do paciente, utilizar métodos de detecção de bordas é uma estratégia muito útil. Essas técnicas facilitam a identificação dos contornos do braço, destacando as diferenças de intensidade entre ele e o fundo, o que torna os limites mais claros. Ao combinar essa abordagem com a segmentação por cor, conseguimos uma delimitação mais precisa da região de interesse.

As bordas em uma imagem estão associadas a pontos onde ocorrem variações abruptas na intensidade, ou seja, são regiões onde a primeira derivada da função de intensidade atinge valores extremos. Em termos práticos, isso significa que as transições bruscas de cor ou brilho, características de bordas, podem ser detectadas ao se analisar as mudanças na intensidade dos pixels. Essa abordagem fundamenta-se na ideia de que as derivadas capturam as variações locais da imagem, permitindo isolar pontos de interesse que indicam a presença de contornos.

O gradiente da imagem é a ferramenta matemática utilizada para quantificar essas variações. Definido pela Equação 2.12, o gradiente aponta na direção de subida mais íngreme da função de intensidade.

$$J(x) = \nabla I(x) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = (I_x, I_y) \quad (2.12)$$

A magnitude do gradiente, calculada na Equação 2.13, reflete a força da borda, ou seja, quanto maior essa magnitude, mais forte é a transição entre regiões, evidenciando um contorno bem definido.

$$\|\nabla I(x)\| = \sqrt{I_x^2 + I_y^2} \quad (2.13)$$

Além disso, a direção do gradiente, determinada pelo ângulo α da Equação 2.14 é perpendicular à borda, pois a maior variação de intensidade ocorre justamente na direção normal à transição de cor.

$$\alpha = \tan^{-1} \left(\frac{I_y}{I_x} \right) \quad (2.14)$$

Dessa forma, o uso de derivadas e do gradiente fornece uma base sólida para a detecção de bordas, permitindo identificar com precisão tanto a localização quanto a orientação dos contornos em uma imagem.

A partir dos conceitos apresentados anteriormente, (CANNY, 1986) propôs um método para detecção de bordas fundamentado em três objetivos principais: reduzir ao mínimo a ocorrência de falsos positivos e falsos negativos, assegurar a localização precisa dos contornos e garantir uma resposta única para cada borda. O Detector de

Canny destaca-se por sua robustez e eficiência na identificação de contornos bem definidos, tornando-o uma das técnicas mais eficazes para a detecção de bordas em imagens digitais.

Entre as diversas variantes do Detector de Canny existentes, uma das implementações mais populares adota os seguintes passos: filtrar a imagem com a derivada da Gaussiana, calcular a magnitude e a orientação do gradiente, realizar a supressão não-máxima e aplicar a limiarização com histerese.

No processo de supressão não-máxima, o objetivo é refinar as bordas mantendo apenas os pontos de máximo local. Para isso, a direção do gradiente é discretizada em 8 direções, permitindo identificar qual delas melhor se aproxima da orientação real do gradiente em cada pixel. Em seguida, compara-se a magnitude do gradiente do pixel analisado com os dos pixels vizinhos ao longo dessa direção; se algum dos vizinhos apresentar uma magnitude maior, o pixel em questão é descartado, garantindo que somente os pontos que realmente representam o máximo de uma aresta sejam mantidos.

Na etapa de limiarização com histerese, emprega-se dois valores de limiar para diferenciar os pontos da imagem com base na magnitude do gradiente. Os pixels cuja magnitude ultrapassa o limiar superior são imediatamente considerados como arestas fortes, assegurando a precisão na definição dos contornos. Por outro lado, os pixels que possuem magnitudes entre o limiar inferior e o superior são classificados como arestas fracas e só serão integrados à imagem final se estiverem conectados a alguma aresta forte. Essa estratégia permite manter apenas os contornos significativos e eliminar falsos positivos, e estudos realizados por ([CANNY, 1986](#)) demonstraram que a razão ideal entre os limiares superior e inferior deve estar entre 3:1 e 2:1, contribuindo para a eficácia e robustez do método.

2.3.4 Transformada de Hough

Na detecção de bordas em uma imagem, por exemplo, utilizando o detector de Canny, é fundamental identificar os pixels que formam linhas retas. No presente trabalho, ao detectar o contorno do braço, torna-se necessário corrigir eventuais lacunas na detecção das bordas e utilizar essas linhas para auxiliar na geração da trajetória. Para esse fim, emprega-se o método da Transformada de Hough ([HOUGH, 1962](#)).

Para compreender o conceito de espaço de parâmetros na Transformada de Hough, considere um ponto (x_i, y_i) no plano cartesiano xy e a equação de uma reta em [2.15](#).

$$y_i = ax_i + b \quad (2.15)$$

Esse ponto pode pertencer a um conjunto infinito de retas, pois, ao variar os valores de a e b , todas as equações que passam por (x_i, y_i) são satisfeitas. Reescrevendo na Equação [2.16](#) observa-se que, para um ponto fixo (x_i, y_i) , essa relação define uma única

reta no plano dos parâmetros, cujos eixos correspondem aos coeficientes a e b . Esse plano, denominado espaço de parâmetros, é essencial na Transformada de Hough, pois converte o problema de detecção de retas em imagens na identificação de interseções de linhas nesse espaço, facilitando a extração de características lineares.

$$b = -ax_i + y_i \quad (2.16)$$

Uma abordagem para detectar retas consiste em representar, no espaço de parâmetros, as retas correspondentes a todos os pontos (x_k, y_k) . Dessa forma, as retas no plano xy são definidas por pontos que representam a interseção de um grande número de retas traçadas nesse espaço. Contudo, um problema significativo surge quando a reta se aproxima da direção vertical, pois o coeficiente angular a tende a infinito, o que pode comprometer a robustez do método. Uma forma de contornar esse problema é utilizar a representação normal da reta, dada pela Equação 2.17.

$$x \cos \theta + y \sin \theta = \rho \quad (2.17)$$

Na equação 2.17, o parâmetro ρ representa a distância entre a reta e a origem do sistema de coordenadas, sendo sempre um valor positivo. Já o parâmetro θ corresponde ao ângulo que a normal à reta faz com o eixo x , variando no intervalo $[0, \pi]$. Dessa forma, a reta é identificada de maneira única no espaço de parâmetros (ρ, θ) , independentemente da sua inclinação no plano xy . Essa parametrização permite uma representação mais estável das retas, especialmente para aquelas próximas da vertical, onde a inclinação tradicional a poderia tender a valores infinitos, dificultando a análise. Além disso, a Transformada de Hough utiliza essa parametrização para converter o problema da detecção de retas em um problema de identificação de pontos de interseção no espaço de parâmetros, facilitando a extração de características lineares em imagens digitais.

Na Figura 3, é possível verificar o ponto de interseção no plano ρ - θ que corresponde à linha passando pelos pontos (x_i, y_i) e (x_j, y_j) no plano xy . Na mesma figura, é possível visualizar as células acumuladoras utilizadas no método para armazenar a contagem de votos de cada possível reta detectada. O espaço de parâmetros é discretizado em uma grade de valores de ρ e θ , onde cada célula representa uma possível configuração desses parâmetros. As células que acumulam mais votos indicam a presença de retas mais prováveis na imagem original.

A transformada de Hough é um método genérico e, por si só, não consegue distinguir o braço do paciente de outros contornos, como o da mesa abaixo do braço. Por isso, a segmentação por cor proposta anteriormente é essencial para minimizar esses problemas.

Na aplicação, a partir dos resultados obtidos pela Transformada de Hough, o objetivo é agrupar as retas detectadas de forma a identificar os contornos do braço. Com

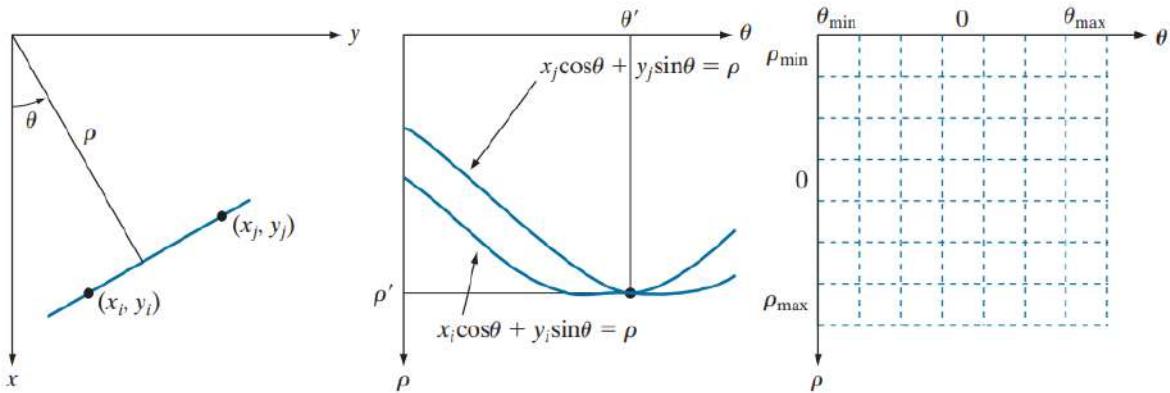


Figura 3 – Parametrização de uma reta no plano xy , representação no espaço de parâmetros ρ - θ e a subdivisão em células acumuladoras.

Fonte: ([GONZALEZ; WOODS, 2018](#))

essas informações, determina-se a linha média entre as duas retas, permitindo que o manipulador siga uma trajetória centralizada ao longo do braço.

3 METODOLOGIA

Este Capítulo apresenta os detalhes das estratégias de visão computacional empregadas na geração de trajetórias para vestir o paciente. São descritos o ambiente de simulação proposto para a tarefa e os detalhes da cinemática do robô manipulador Kinova com 7 DoF ([KINOVA, 2024](#)).

O objetivo consiste em utilizar, a partir das imagens capturadas por um sensor RGB-D – dispositivo que capta imagens coloridas (RGB) e informações de profundidade –, métodos de processamento de imagem e visão computacional para determinar a inclinação da linha que acompanha o braço, bem como traçar o percurso da manga da vestimenta.

3.1 Configuração da Simulação

Levando em conta o desafio proposto em ([EMPRISELAB, 2024](#)), foi configurado o ambiente de simulação ilustrado na Figura 4, utilizando o simulador CoppeliaSim ([ROHMER; SINGH; FREESE, 2013](#)). Nesse ambiente, são definidos como parâmetros os ângulos das sete juntas do robô e os comandos de abertura e fechamento da garra. Além disso, o simulador fornece diversos dados, como as posições das juntas, do *end-effector*, imagens capturadas pelo sensor RGB-D e as posições do braço do paciente, os quais serão utilizados posteriormente para a comparação dos resultados.

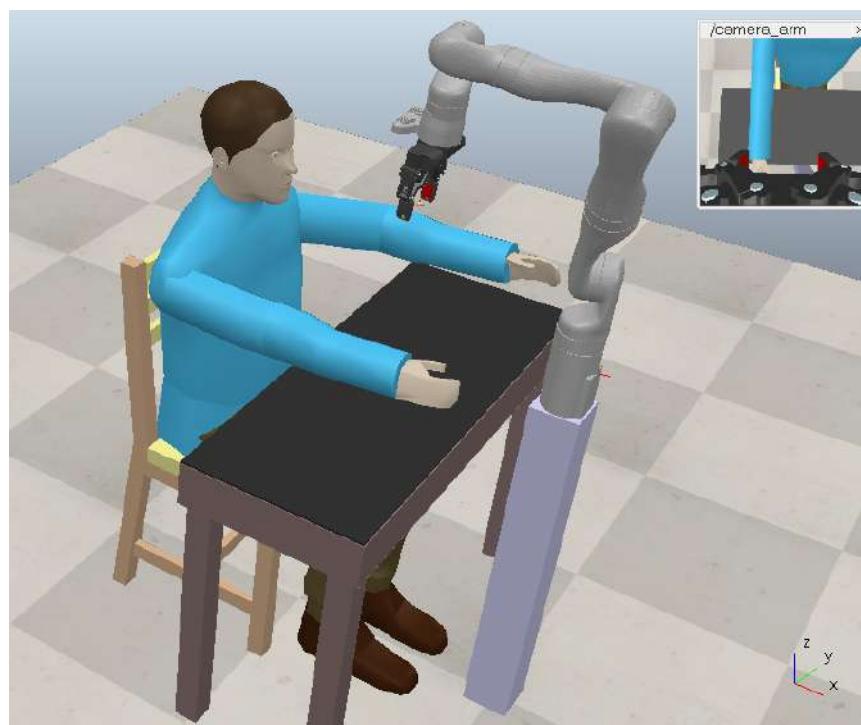


Figura 4 – Configuração inicial do ambiente de simulação.

Fonte: Autor

O posicionamento inicial do manipulador e da câmera desempenha um papel fundamental na obtenção de imagens de alta qualidade e na garantia de que o paciente esteja adequadamente posicionado no espaço de trabalho do robô. A câmera RGB-D foi estrategicamente instalada para maximizar a visibilidade do braço e da vestimenta, facilitando o processamento das informações pelos algoritmos de visão computacional. Por sua vez, o manipulador foi ajustado para proporcionar um acesso otimizado à área de interação, reduzindo movimentos desnecessários e contribuindo para a segurança do paciente.

É importante ressaltar que, embora o objetivo seja promover maior autonomia ao paciente, ainda existem diversos desafios na tarefa de vestir. Neste trabalho, adota-se uma abordagem simplificada para reduzir a complexidade da manipulação. Conforme discutido na Seção 2.2, a estratégia proposta minimiza a variabilidade no posicionamento inicial da vestimenta. Para mitigar as dificuldades inerentes à manipulação, a roupa é fixada diretamente no efetuador final do robô, conforme sugerido por (ERICKSON et al., 2018), permitindo que o algoritmo se concentre exclusivamente no planejamento da trajetória.

A vestimenta considerada neste estudo é um jaleco, uma peça de roupa aberta que facilita a tarefa ao possibilitar a definição de uma trajetória sobre o braço. Essa trajetória, uma vez traçada, poderá ser replicada para o outro braço. Na simulação, optou-se por abstrair a representação da manga do jaleco posicionado no manipulador, visto que, embora o simulador permita a simulação de objetos deformáveis, ele não fornece, de maneira nativa, opções de vestimentas.

Visando generalizar as estratégias de visão computacional, foram consideradas diversas configurações de posicionamento do braço e do antebraço do paciente. A Figura 5 apresenta as três situações de validação.



Figura 5 – Configurações de posicionamento do braço do paciente.

Fonte: Autor

3.2 Estratégias adotadas

Inicialmente, estabelece-se a conexão com o simulador CoppeliaSim por meio de uma API remota. Dessa forma, são definidos os *handlers* para cada junta e para outros objetos de interesse na cena, como a garra e o *end-effector*. Além disso, são configurados, por padrão, os parâmetros iniciais da cinemática inversa – tais como a tolerância e a taxa de aprendizado – bem como os parâmetros do algoritmo, que incluem o limiar de proximidade ao braço, a distância da trajetória do punho ao cotovelo e a distância do cotovelo ao ombro. Esses parâmetros padrão podem ser visualizados na Tabela 1.

Tabela 1 – Parâmetros definidos inicialmente no algoritmo

Parâmetro	Definição Padrão
Tolerância	10^{-4}
Taxa de Aprendizado	0,01
Limiar de proximidade	0,06 m
Distância do punho ao cotovelo	0,25 m
Distância do cotovelo ao ombro	0,16 m

Além disso, tais parâmetros foram ajustados com base em testes preliminares para garantir a precisão e a segurança do sistema.

Em seguida, para cumprir a tarefa, o manipulador se aproxima do punho do paciente utilizando o sensor de profundidade do sensor RGB-D, garantindo a segurança durante todo o processo. No entanto, para uma tarefa como a de vestir, é necessário que o manipulador se aproxime o suficiente para alinhar a manga com o braço, permitindo que ela deslize corretamente. É importante ressaltar que, durante todo o processo, o manipulador está equipado com sensores tátteis (de força) na garra, que permitem a detecção de colisões e interrompem imediatamente seu funcionamento caso ocorra algum contato inesperado.

Na Figura 6, é apresentado um fluxograma que representa o início da execução do algoritmo e o processo de aproximação. O manipulador executa um processo iterativo até que a garra esteja próxima ao punho do paciente, respeitando um limiar de segurança. Para isso, além dos dados de profundidade, é utilizada a segmentação por cor, permitindo a exclusão de objetos irrelevantes, como a própria garra do manipulador, que aparece na imagem da câmera e está muito próxima do sensor, mas não deve ser considerada na verificação de profundidade.

Após a aproximação com a coordenada z fixa, o algoritmo gera as trajetórias com base na configuração do braço do paciente. Para isso, diferentes configurações são consideradas na etapa de validação, conforme ilustrado na Figura 5. Inicialmente, o manipulador deve calcular a trajetória do punho até o cotovelo, correspondente ao movimento do antebraço. Nesse processo, os algoritmos de visão computacional devem gerar uma linha reta como resultado das estratégias discutidas nas Subseções 3.2.1, 3.2.2 e

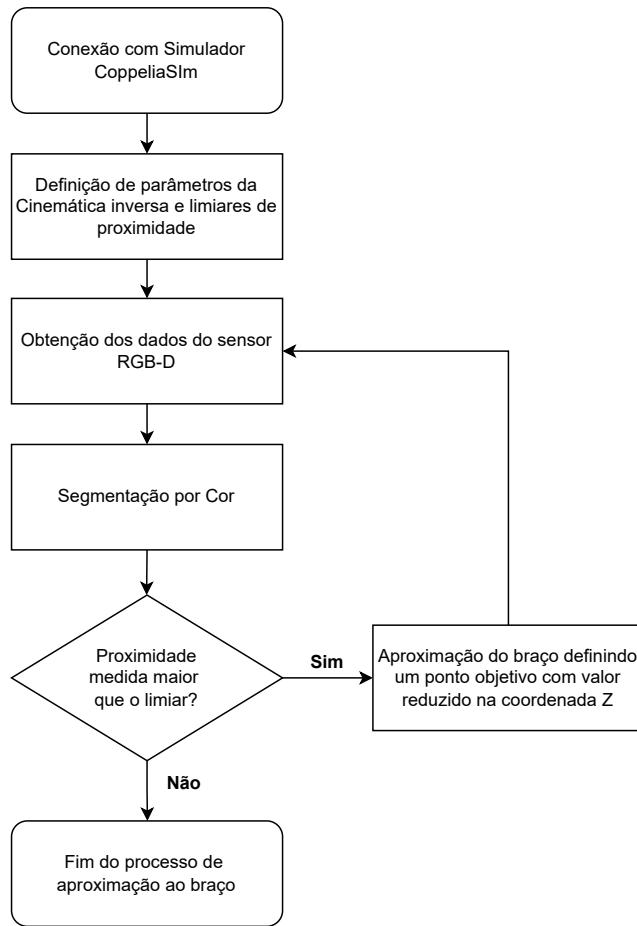


Figura 6 – Fluxograma do algoritmo de aproximação ao braço do paciente.

Fonte: Autor

3.2.3, garantindo um coeficiente linear adequado à configuração atual do braço do paciente.

Na Figura 7, é apresentado um exemplo de resultado utilizado para a determinação das trajetórias. Com base nos coeficientes da reta, é possível calcular os pontos x e y de destino a partir do ângulo θ , conforme descrito nas Equações 3.1 e 3.2. Nessa abordagem, dx representa o incremento da posição ao longo do eixo das abscissas, enquanto dy corresponde ao incremento ao longo do eixo das ordenadas. O processo continua iterativamente até atingir uma distância pré-definida, conforme estabelecido na Tabela 1.

$$x_{objetivo} = x_{atual} + dx \cdot \cos(\theta) \quad (3.1)$$

$$y_{objetivo} = y_{atual} + dy \cdot \sin(\theta) \quad (3.2)$$

Com os pontos de destino definidos, a cinemática inversa, conforme discutida na

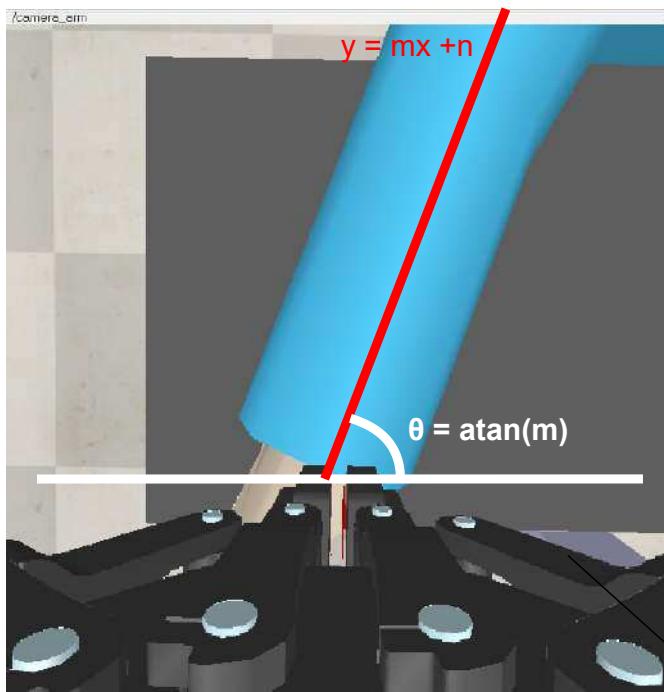


Figura 7 – Exemplo de reta obtida para cálculo dos pontos objetivo.

Fonte: Autor

Seção 3.3, é empregada para posicionar o robô até o cotovelo. Em seguida, a partir da nova imagem capturada, os passos anteriores são repetidos, finalizando a tarefa de vestir.

3.2.1 Estratégia baseada em Segmentação por Cor

Uma das estratégias adotadas para gerar a reta sobre o braço é a segmentação por cor. Nesse método, as imagens são capturadas pela câmera e, a partir daí, realiza-se a conversão do espaço de cores de RGB para HSV, de modo que a característica da cor fique concentrada em um único canal, o matiz, conforme discutido na Seção 2.3.2.

Como parâmetro para a segmentação, o algoritmo utiliza a cor do pixel central da imagem para definir uma máscara capaz de diferenciar o braço dos demais objetos. Considerando que o braço está inicialmente centralizado em relação ao manipulador e à câmera, a máscara é gerada aplicando uma tolerância de 20% em relação à cor do pixel central; pixels com valores até 20% acima ou abaixo são configurados como brancos, enquanto os demais se tornam pretos. Dessa forma, a região de interesse é destacada dos demais elementos presentes na imagem. Na Figura 8, observa-se um exemplo dessa máscara, onde a área correspondente ao braço é evidenciada. Esse procedimento permite ao algoritmo focar na extração de informações precisas dessa região, etapa fundamental para identificar a trajetória central que orientará os movimentos subsequentes do manipulador.

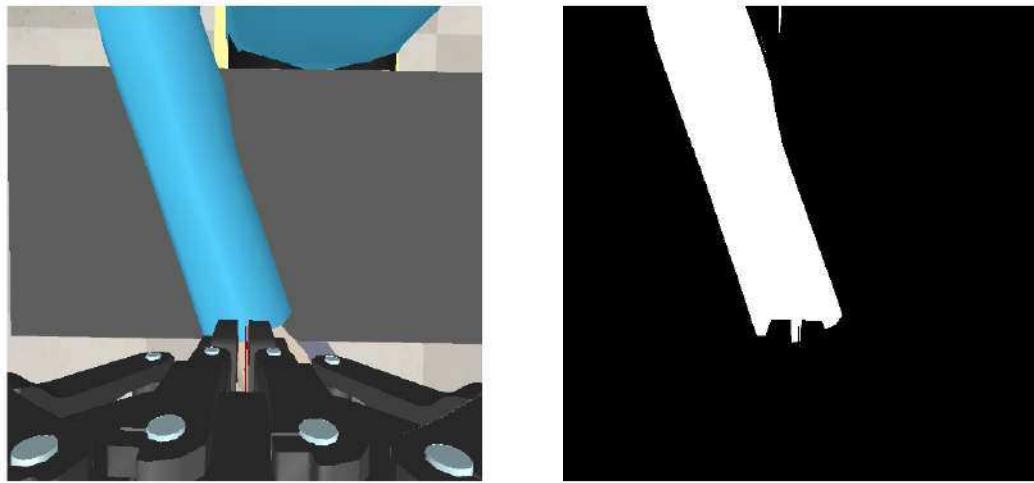


Figura 8 – Exemplo da imagem antes e após segmentação por cor.

Fonte: Autor

Após a segmentação, a máscara resultante é percorrida linha por linha. Em cada linha, são identificados os pixels com intensidade diferente de 0 (ou seja, os pixels brancos), e seus índices são armazenados em um vetor. Ao final de cada linha, calcula-se o ponto médio entre esses pixels brancos. Dessa forma, ao concluir a varredura de todas as linhas, é possível construir uma linha média composta por esses pontos médios, representando o reta central da região segmentada.

Com base nas coordenadas x e y dos pontos médios extraídos da região do braço na imagem, realiza-se um ajuste polinomial de primeiro grau para determinar a equação da reta que melhor representa a trajetória central. A partir dessa equação, obtém-se o coeficiente angular, que é utilizado para calcular o ângulo θ por meio da função arco-tangente. Esse ângulo é fundamental para orientar a trajetória a ser seguida pelo manipulador.

3.2.2 Estratégia baseada na Transformada de Hough

A estratégia que utiliza a Transformada de Hough parte da máscara obtida pela segmentação por cor, visando aprimorar os resultados obtidos apenas com esse método. A ideia é superar as limitações de generalização do modelo, permitindo identificar uma reta que se ajuste de forma mais precisa à região do braço.

Para aplicar a Transformada de Hough, é necessário realizar uma sequência de operações de processamento de imagem. Primeiramente, a imagem RGB do braço é extraída a partir da máscara de segmentação aplicada à imagem original, isolando a região de interesse. Em seguida, a imagem é convertida para escala de cinza e submetida a um filtro gaussiano, o qual suaviza os detalhes e reduz o ruído. Posteriormente, o detector

de bordas de Canny é aplicado, realçando as bordas mais significativas, e, por fim, a Transformada de Hough é utilizada por meio da função `hough` do MATLAB, retornando a matriz de transformação e os vetores dos parâmetros ρ e θ no espaço de Hough, conforme discutido na Seção 2.3.4.

A partir dos resultados obtidos, é possível identificar os picos na matriz de transformação utilizando a função `houghpeaks`, aplicando uma tolerância para definir quais valores serão considerados como picos. Com base nas interseções no espaço de Hough, os segmentos de reta são extraídos a partir dos picos detectados e dos vetores dos parâmetros ρ e θ , utilizando a função `houghlines`.

Um passo fundamental consiste em agrupar as retas que apresentam parâmetros muito próximos no espaço de Hough. Dessa forma, é possível determinar, por meio da média dos parâmetros de cada grupo, os valores finais de ρ e θ , obtendo assim a reta que melhor se ajusta ao contorno do braço. O passo a passo discutido pode ser visto na Figura 9.

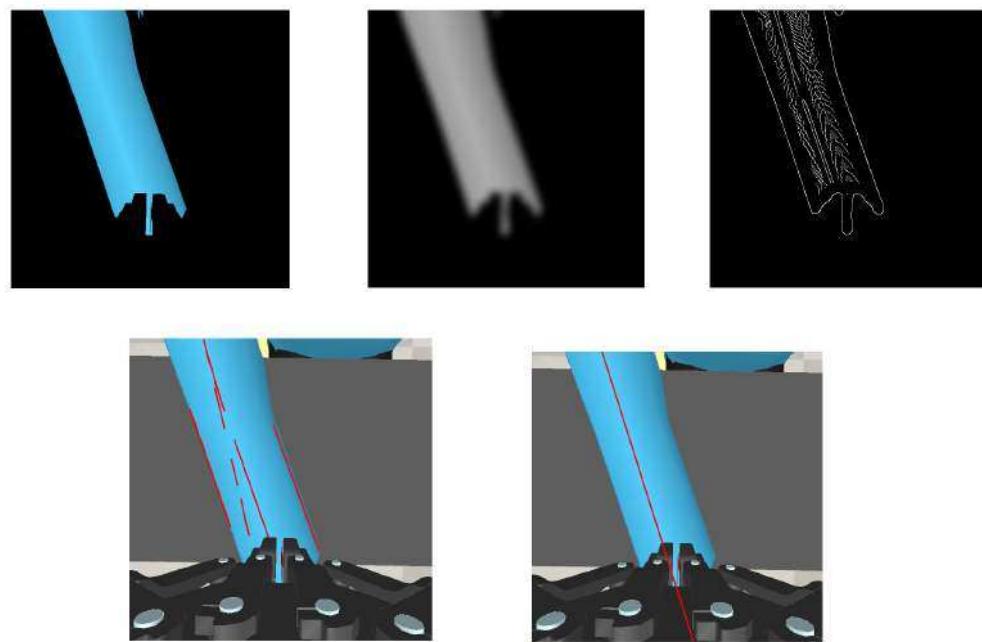


Figura 9 – Imagens obtidas nos passos da estratégia usando Transformada de Hough.

Fonte: Autor

Com os parâmetros do espaço de Hough definidos, determinam-se os valores de x a partir das coordenadas dos pontos mínimo e máximo dos segmentos de reta detectados, enquanto os valores de y são calculados conforme a Equação 3.3, obtida a partir da Equação 2.17:

$$y = \frac{\rho - x \cdot \cos(\theta)}{\sin(\theta)} \quad (3.3)$$

Com os vetores de pontos definidos, realiza-se um ajuste polinomial de primeiro grau para determinar a equação da reta. A partir deste ajuste, obtém-se o coeficiente angular por meio da função arco-tangente, sendo esse ângulo fundamental para orientar a trajetória a ser seguida pelo manipulador.

3.2.3 Estratégia baseada em Detecção de Contornos

Considerando que a Transformada de Hough pode em alguns casos detectar retas provenientes de objetos indesejados, como a mesa, propõe-se uma estratégia alternativa para aprimorar os resultados. Nessa abordagem, a detecção de contornos utiliza imagens binárias — como a obtida após a segmentação por cor — para delimitar o contorno do braço e, a partir desses limites, definir o segmento de reta central.

Nessa estratégia, a imagem binária do braço, obtida a partir da máscara de segmentação por cor, é submetida à função `bwboundaries` do MATLAB. Essa função é empregada para identificar os contornos dos objetos presentes em uma imagem binária, detectando as fronteiras de cada objeto isolado. O seu retorno consiste em uma célula na qual cada elemento contém as coordenadas dos pontos que formam o contorno de um objeto detectado.

Posteriormente, o algoritmo identifica o contorno que possui o maior número de pontos dentre os contornos detectados e seleciona esse contorno para análise. Em seguida, determinam-se os valores mínimo e máximo das coordenadas horizontais dos pontos desse contorno, e calcula-se o ponto médio entre esses extremos. Com base nesse ponto médio, o contorno é dividido em duas partes: os pontos localizados à esquerda e os localizados à direita.

Na etapa seguinte, para calcular a linha central, o código percorre todos os valores únicos das coordenadas verticais (y) do contorno. Para cada valor, são extraídos os pontos correspondentes dos segmentos esquerdo e direito. Se, para um determinado y , existirem pontos em ambos os segmentos, o algoritmo calcula a média dos valores horizontais (x) desses pontos, definindo assim o ponto central para aquela linha específica. Ao repetir esse procedimento para cada valor único de y , o algoritmo constrói uma sequência de pontos médios que, quando conectados, formam a linha central do contorno. O passo a passo para a determinação da linha central a partir da imagem original pode ser visualizado na Figura 10.

Por fim, essa linha central é submetida a um ajuste polinomial de primeiro grau. A partir do coeficiente angular resultante desse ajuste, calcula-se o ângulo de inclinação, que é fundamental para orientar a trajetória a ser seguida pelo manipulador.

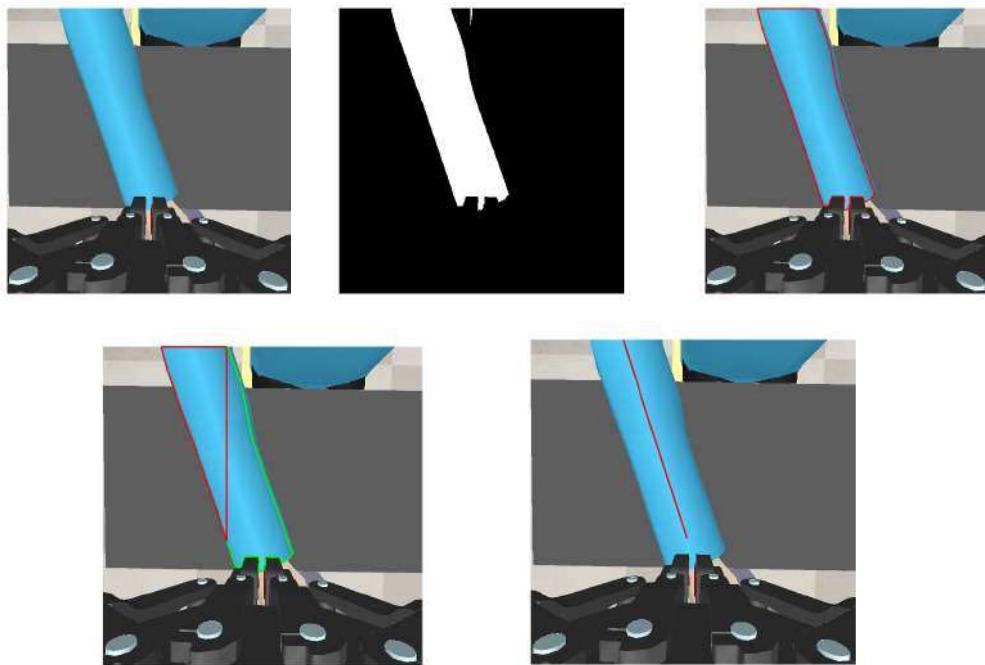


Figura 10 – Imagens obtidas nos passos da estratégia usando Detecção de Contornos.

Fonte: Autor

3.3 Cinemática do Manipulador

A partir da trajetória definida pelas estratégias adotadas na Seção 3.2, é necessário determinar os ângulos das juntas para que o manipulador robótico complete o percurso, alcançando os pontos desejados. Para tanto, é fundamental aplicar os conceitos da robótica clássica, como a cinemática direta e inversa, ao manipulador Kinova.

O manipulador Kinova 7 DoF apresenta uma estrutura composta por sete juntas rotacionais, conforme visto na Figura 11, permitindo uma ampla gama de movimentos e proporcionando elevada flexibilidade e precisão. A modelagem do manipulador envolve a definição dos parâmetros geométricos essenciais, tais como os comprimentos dos elos, os sentidos de rotação das juntas e as diferenças angulares entre elas.

A modelagem matemática do manipulador robótico é construída de modo a determinar a posição no espaço cartesiano a partir dos ângulos das juntas rotacionais. Por meio de transformações homogêneas, são estabelecidas as relações entre os referenciais (*frames*) do robô, permitindo a composição sequencial de transformações que relacionam cada junta e elo ao sistema de coordenadas global. A cinemática direta consiste em calcular a posição e a orientação do efetuador final (*end-effector*) a partir dos valores dos ângulos das juntas.

Por convenção, utiliza-se o método sistemático de Denavit-Hartenberg, descrito em (HARTENBERG; DENAVIT, 1964), para a modelagem cinemática, o qual define quatro

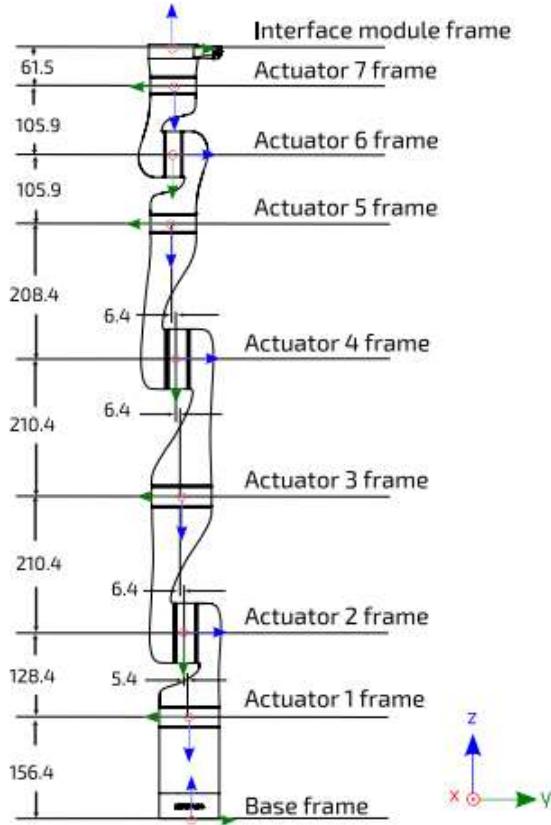


Figura 11 – Dimensões e definições de *frames* do robô Kinova com 7 graus de liberdade.

Fonte: ([KINOVA, 2022](#))

parâmetros para cada junta: d (distância ao longo do eixo z), θ (ângulo de rotação em torno do eixo z), a (deslocamento ao longo do eixo x) e α (ângulo de rotação em torno do eixo x). Com base no manual do robô ([KINOVA, 2022](#)), a Tabela 2 apresenta os valores desses quatro parâmetros para a transição entre os *frames* do robô, sendo θ_i^* os ângulos variáveis das juntas.

Tabela 2 – Parâmetros de Denavit-Hartenberg do robô manipulador Kinova 7 DoF

Frame	$\alpha_i(\text{rad})$	$a_i(\text{m})$	$d_i(\text{m})$	$\theta_i(\text{rad})$
0	π	0,0	0,0	0
1	$\pi/2$	0,0	$-(0,1564+0,1284)$	θ_1^*
2	$\pi/2$	0,0	$-(0,0054+0,0064)$	$\theta_2^* + \pi$
3	$\pi/2$	0,0	$-(0,2104+0,2104)$	$\theta_3^* + \pi$
4	$\pi/2$	0,0	$-(0,0064+0,0064)$	$\theta_4^* + \pi$
5	$\pi/2$	0,0	$-(0,2084+0,1059)$	$\theta_5^* + \pi$
6	$\pi/2$	0,0	0,0	$\theta_6^* + \pi$
7	π	0,0	$-(0,1059+0,0615)$	$\theta_7^* + \pi$

Para definir a cinemática inversa, isto é, determinar os ângulos das juntas rotacionais a partir da posição e da orientação desejadas no espaço cartesiano, é necessário empregar

métodos de otimização, como o gradiente descendente. Neste contexto, a função objetivo é constituída pelo erro quadrático médio entre a posição e a orientação do efetuador final e as posições e orientações desejadas. Por esse motivo, a Tabela 1 apresenta os valores da taxa de aprendizado e da tolerância adotados. A taxa de aprendizado mostra o tamanho dos ajustes feitos a cada iteração e determina a rapidez com que o algoritmo se aproxima da solução ideal. Como o método funciona de forma iterativa, é importante definir uma tolerância adequada para que o algoritmo saiba quando os ângulos das juntas estão suficientemente próximos dos valores que minimizam o erro.

Por fim, os ângulos das juntas calculados pelo método do gradiente descendente são enviados para o simulador CoppeliaSim. Dessa forma, o manipulador segue a trajetória definida, ajustando os ângulos de forma sequencial para que, a cada ponto, o robô se posicione corretamente e complete o percurso com precisão.

4 RESULTADOS

Neste Capítulo, são apresentados os resultados obtidos no desenvolvimento do trabalho. São detalhados os resultados referentes à aproximação do braço e à geração de trajetórias, com base nas estratégias descritas na Seção 3.2. São expostos os resultados das simulações, bem como a adequação das trajetórias geradas em comparação à trajetória ideal, além de um comparativo entre as estratégias adotadas para identificar a mais eficaz.

Para generalizar o método para diferentes configurações, foram testadas três situações em todos os resultados, conforme ilustrado na Figura 5: na primeira imagem, a seção do ombro ao cotovelo (braço) forma um ângulo de 20º para a esquerda em relação à posição reta, enquanto a seção do cotovelo ao punho (antebraço) permanece alinhada; na segunda imagem, o braço está a 20º para a direita e o antebraço a 20º para a esquerda; e, na última, o braço permanece reto e o antebraço está a 20º para a direita.

4.1 Resultados da aproximação ao braço

Nesta etapa, o manipulador deve se aproximar adequadamente do braço, pois essa proximidade é fundamental para a tarefa de vestir, sem comprometer os fatores de segurança. Conforme apresentado na Tabela 1, o algoritmo define um limiar de 6 cm entre a garra, que segura a manga da roupa, e o braço, utilizando um raio de segurança de 5 cm para a comparação. É importante ressaltar que esse limiar já considera a distância medida pelo sensor RGB-D do *end-effector*, de modo que os 6 cm representam a distância efetiva entre a garra e o braço.

A configuração inicial do braço do paciente é a apresentada na primeira imagem da Figura 5, com o braço inclinado 20º à esquerda e o antebraço reto. A partir de uma posição em z elevada do manipulador robótico sobre o braço, conforme ilustrado na Figura 12, pode-se observar as coordenadas do eixo z do *end-effector* ao longo de toda trajetória (linha azul) e uma reta, em vermelho, representando o raio de segurança.

Na Figura 12 é possível verificar também a proximidade para a tarefa de vestir de forma visual pela imagem do simulador com a configuração do braço e antebraço utilizada inicialmente. A Tabela 3 apresenta a distância final do *end-effector* em relação ao braço, bem como a margem de segurança (expressa em porcentagem) acima deste, considerando que o braço do paciente está posicionado no plano z = 0,92 m.

Em seguida, utilizando a configuração exibida na segunda imagem da Figura 5, que apresenta o braço inclinado em 20º para a direita e o antebraço em 20º para a esquerda, e partindo de uma posição elevada, o objetivo é validar a estratégia de aproximação para diferentes configurações. Na Figura 13, as coordenadas do eixo z do *end-effector* são exibidas ao longo de toda a trajetória (linha azul), juntamente com uma reta vermelha

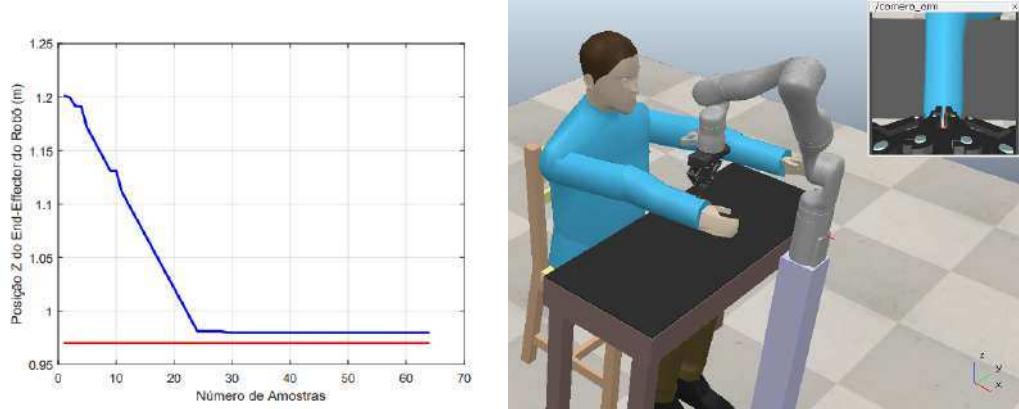


Figura 12 – Coordenadas do eixo z da garra ao longo de toda trajetória e simulação na configuração do braço inclinado 20° à esquerda e antebraço reto.

Fonte: Autor

que representa o raio de segurança. A distância final ao braço é apresentada em 3 e a aproximação pode ser verificada de forma visual pela imagem do simulador na Figura 13.

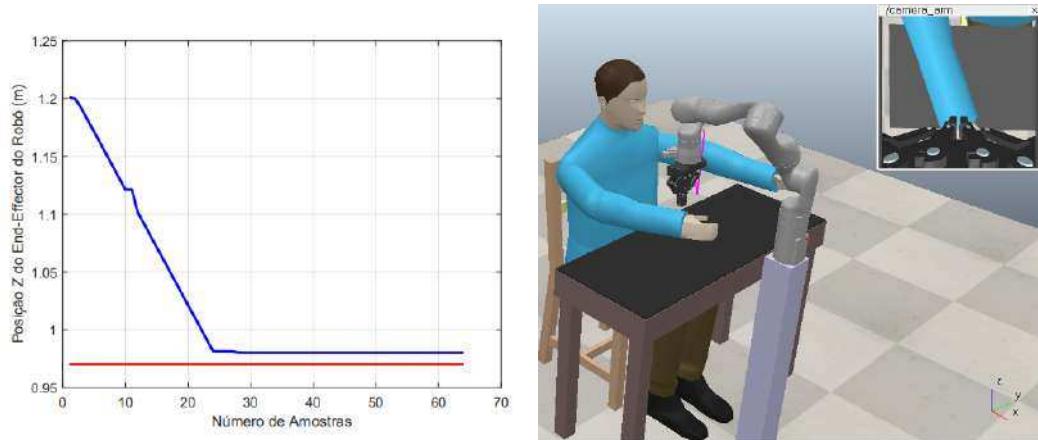


Figura 13 – Coordenadas do eixo z da garra ao longo de toda trajetória e simulação na configuração do braço inclinado 20° à direita e antebraço 20° à esquerda.

Fonte: Autor

Por fim, os dados referentes à posição no eixo z foram coletados na última configuração apresentada na Figura 5, na qual o braço se mantém reto e o antebraço está inclinado em 20° para a direita. Na Figura 14, a linha azul mostra as coordenadas do eixo z durante toda a tarefa, enquanto a linha vermelha indica o raio de segurança. Além disso, o resultado na simulação pode ser visto na segunda imagem da Figura 14 e a distância final na Tabela 3.

Os dados da Tabela 3 demonstram a consistência dos resultados, mesmo diante de variações, pois todas as três configurações testadas apresentaram uma distância final média de 5,94 cm entre a garra e o braço, com uma margem de segurança média de 6,45%. Esses

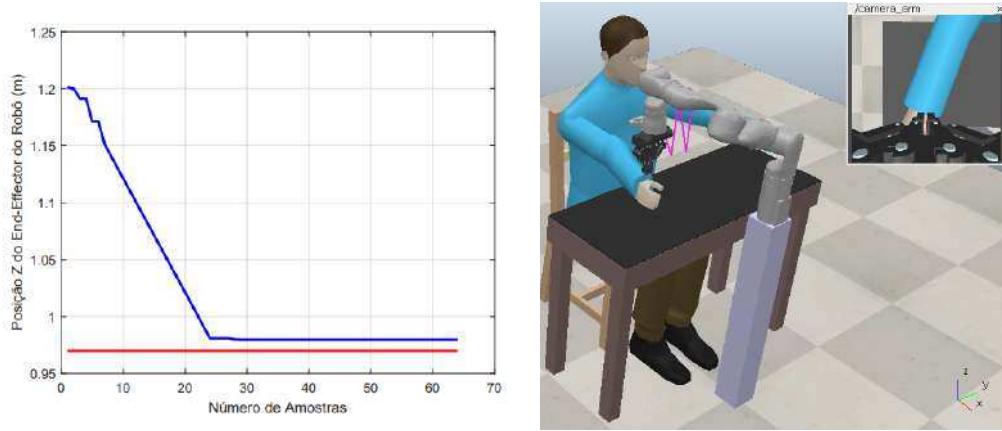


Figura 14 – Coordenadas do eixo z da garra ao longo de toda trajetória e simulação na configuração do braço reto e antebraço inclinado 20º à direita.

Fonte: Autor

Tabela 3 – Comparação da distância final após aproximação da garra ao braço e margem de segurança.

Configuração	Distância final da garra ao braço (cm)	Porcentagem acima do braço
Braço inclinado 20º à esquerda e antebraço reto	5,9935	6,51 %
Braço inclinado 20º à direita e antebraço 20º à esquerda	5,8527	6,36 %
Braço reto e antebraço inclinado 20º à direita	5,9739	6,48 %

resultados indicam que o algoritmo de aproximação, baseado em dados do sensor RGB-D, assegura a proximidade ideal para a tarefa de vestir sem comprometer os parâmetros de segurança.

4.2 Resultados da geração de trajetórias

Para validar cada uma das estratégias quanto à adequação ao braço, nesta etapa são apresentadas as imagens resultantes dos algoritmos de visão computacional. Essa abordagem permite identificar possíveis causas de divergência entre a reta resultante e a trajetória ideal.

Além disso, para comparar as estratégias, é necessário avaliar a resposta no plano xy tanto da trajetória ideal quanto da planejada pelo algoritmo. Para isso, utilizam-se três pontos *dummy* no simulador, localizados no punho, cotovelo e ombro. Esses pontos possibilitam que o algoritmo receba as coordenadas e trace uma reta entre eles para futura comparação com a trajetória planejada.

Por fim, é importante verificar a trajetória efetiva executada pelo manipulador,

obtida por meio da cinemática inversa, algoritmo de robótica clássica discutido na Seção 3.3.

4.2.1 Estratégia usando Segmentação por Cor

As análises dos resultados de segmentação serão feitas individualmente para cada configuração do braço do paciente. Ao final, na Tabela 4 são apresentados os valores da raiz do erro quadrático médio (RMSE, do inglês *root mean squared error*) e de outras métricas utilizadas para comparar as trajetórias planejada e ideal em todas configurações. Ademais, as métricas de erro entre a trajetória planejada e a executada pela cinemática inversa das 3 configurações estão dispostas na Tabela 5.

4.2.1.1 Braço inclinado 20º à esquerda e antebraço reto

Inicialmente, com o braço do paciente inclinado em 20º para a esquerda e o antebraço reto, as duas imagens binárias resultantes da segmentação são apresentadas na Figura 15.



Figura 15 – Imagens binárias da segmentação por cor do braço inclinado 20º à esquerda e antebraço reto: (a) antebraço; (b) braço.

Fonte: Autor

Na Figura 16 observa-se a trajetória planejada (em vermelho) e a trajetória ideal (em azul), definida a partir dos pontos *dummy* no simulador, para a mesma configuração. Nota-se que, neste caso, a trajetória planejada difere significativamente da ideal na região correspondente ao braço, uma vez que o algoritmo também considera o corpo do paciente durante a segmentação.

A trajetória efetiva do manipulador Kinova, obtida a partir da trajetória planejada (em vermelho) sob os parâmetros de tolerância e taxa de aprendizado apresentados na Tabela 1, é exibida em azul na Figura 17. Ademais, o rastro magenta no simulador evidencia a trajetória percorrida.

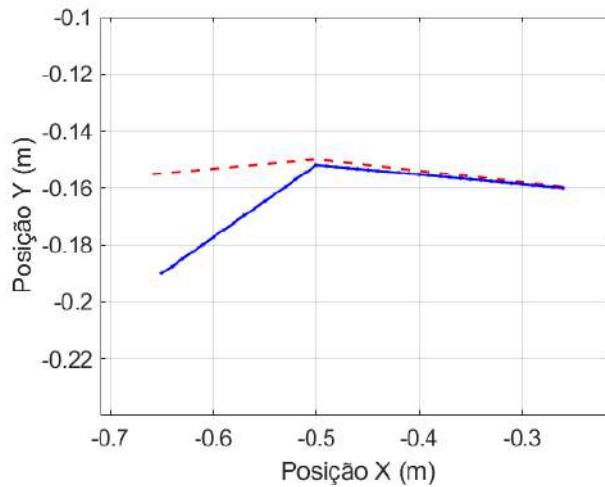


Figura 16 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20º à esquerda e antebraço reto usando a segmentação por cor.

Fonte: Autor

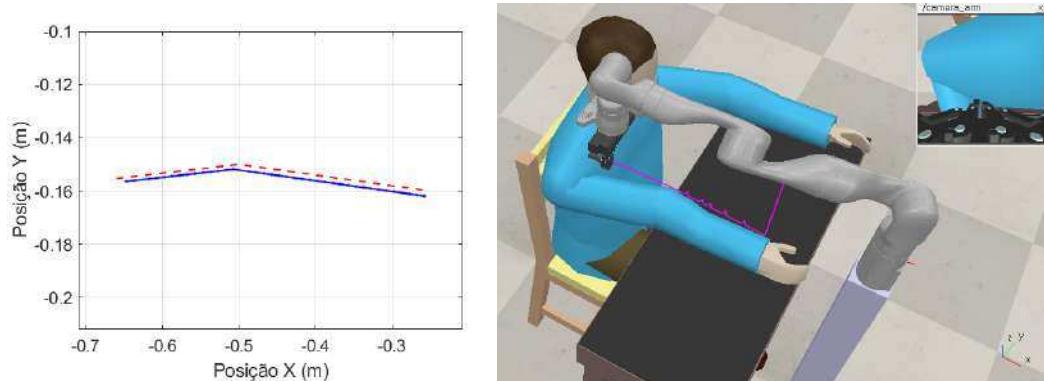


Figura 17 – Usando a segmentação por cor: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à esquerda e antebraço reto; (b) Trajetória seguida na simulação.

Fonte: Autor

4.2.1.2 Braço inclinado 20º à direita e antebraço 20º à esquerda

Com o braço inclinado em 20º para a direita e o antebraço em 20º para a esquerda, as imagens binárias provenientes da segmentação por cor estão ilustradas na Figura 18. Em seguida, a Figura 19 apresenta, em vermelho, a trajetória planejada, comparada à trajetória ideal, em azul.

Por fim, a Figura 20 apresenta a trajetória efetivamente executada pelo manipulador Kinova, obtida por meio da cinemática inversa com os parâmetros de tolerância e taxa de aprendizado estabelecidos na Tabela 1. Nesta representação, a trajetória executada é exibida em azul, enquanto, na outra imagem, o rastro magenta indica o caminho seguido no simulador.

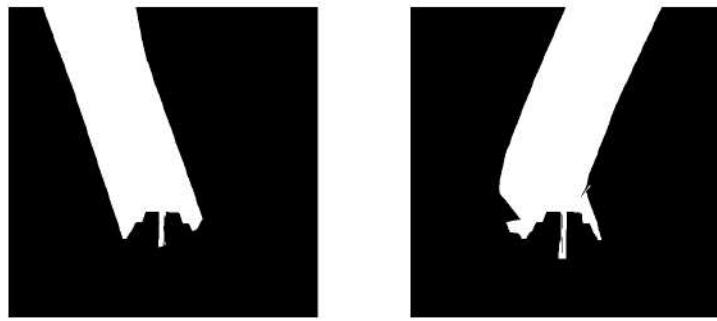


Figura 18 – Imagens binárias da segmentação por cor do braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço.

Fonte: Autor

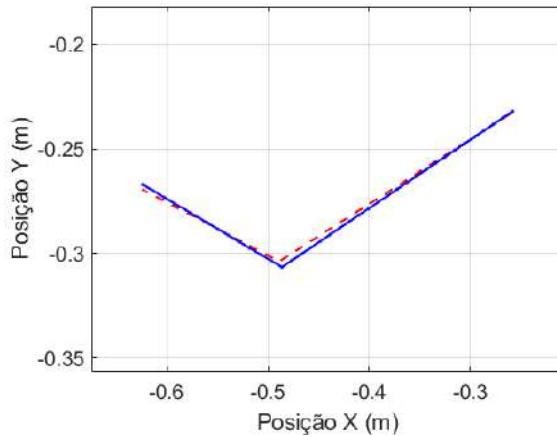


Figura 19 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20º à direita e antebraço 20º à esquerda usando a segmentação por cor.

Fonte: Autor

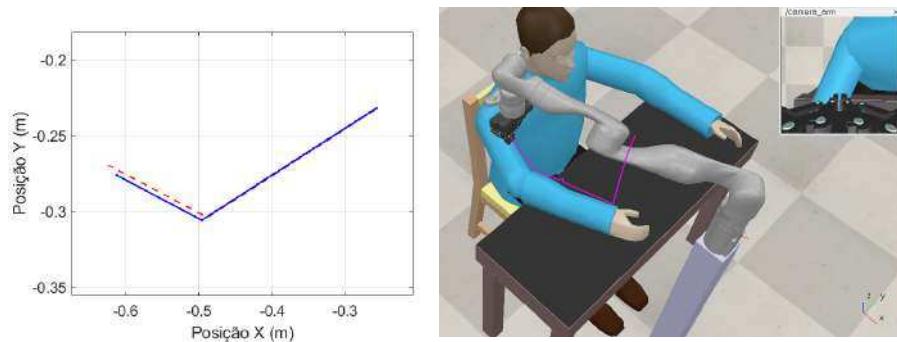


Figura 20 – Usando a segmentação por cor: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à direita e antebraço 20º à esquerda; (b) Trajetória seguida na simulação.

Fonte: Autor

4.2.1.3 Braço reto e antebraço inclinado 20° à direita

Por último, com o braço reto e o antebraço inclinado em 20° para a direita, as imagens binárias da segmentação por cor são apresentadas na Figura 21. Em seguida, a Figura 22 exibe, em vermelho, a trajetória planejada em relação trajetória ideal em azul.



Figura 21 – Imagens binárias da segmentação por cor do braço reto e antebraço inclinado 20° à direita: (a) antebraço; (b) braço.

Fonte: Autor

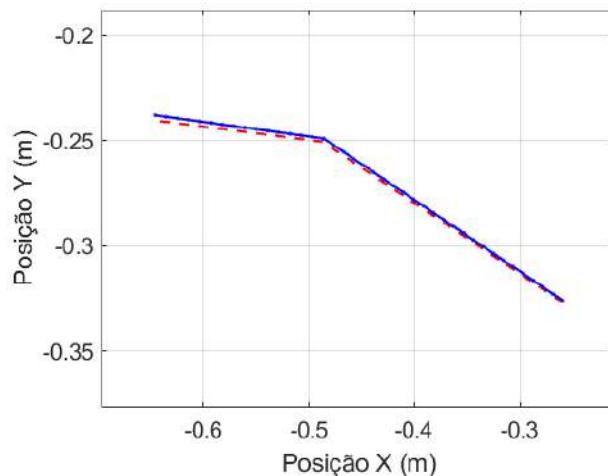


Figura 22 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20° à direita usando a segmentação por cor.

Fonte: Autor

Por fim, a Figura 23 demonstra a trajetória efetivamente executada pelo manipulador Kinova, obtida por meio da cinemática inversa com os parâmetros de tolerância e taxa de aprendizado especificados na Tabela 1. Nesta representação, a trajetória executada é ilustrada em azul, enquanto, na outra imagem, o caminho percorrido no simulador pode ser visto pelo rastro magenta.

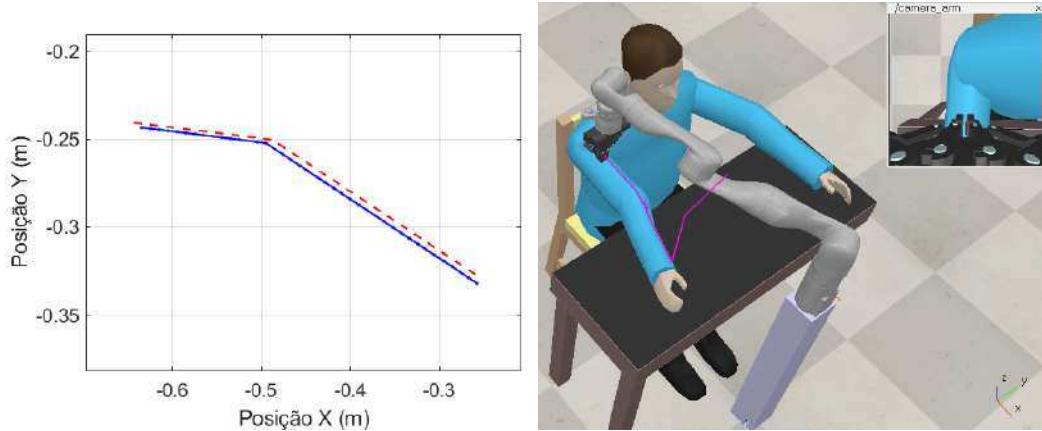


Figura 23 – Usando a segmentação por cor: (a) Trajetória planejada em comparação com a trajetória executada com braço reto e antebraço inclinado 20° à direita; (b) Trajetória seguida na simulação.

Fonte: Autor

4.2.1.4 Comparativo entre configurações

Analizando a Tabela 4, que apresenta as métricas de erro empregadas na comparação entre as trajetórias ideais e planejadas para três configurações distintas, é possível identificar algumas características da estratégia proposta.

Na configuração com o braço inclinado em 20° para a esquerda e o antebraço reto, observa-se que, apesar do RMSE ser relativamente baixo, o desvio padrão se mostra elevado. Esse comportamento ocorre porque a trajetória do antebraço se ajusta adequadamente à trajetória ideal, enquanto a região correspondente ao braço apresenta um aumento do erro decorrente da imagem obtida pela segmentação por cor.

Tabela 4 – Erro entre trajetória planejada e ideal em todas configurações utilizando segmentação por cor.

Configuração	RMSE	Média do erro	Desvio Padrão do erro
Braço inclinado 20° à esquerda e antebraço reto	0,1414 cm	0,91 cm	1,13 cm
Braço inclinado 20° à direita e antebraço 20° à esquerda	0,0520 cm	0,4143 cm	0,3180 cm
Braço reto e antebraço inclinado 20° à direita	0,0447 cm	0,3291 cm	0,2991 cm

Como resultado do algoritmo de cinemática inversa, a Tabela 5 apresenta os erros entre a trajetória que foi planejada e a trajetória seguida pela manipulador robótico.

Após os resultados apresentados, optou-se por investigar estratégias alternativas, com o objetivo de aproximar mais fielmente a reta ideal e corrigir possíveis erros, como os decorrentes da segmentação por cor evidenciados na Figura 16.

Tabela 5 – Erro entre trajetória planejada e executada pelo manipulador robótico em todas configurações utilizando segmentação por cor.

Configuração	RMSE	Média do Erro	Desvio Padrão do erro
Braço inclinado 20º à esquerda e antebraço reto	0,1414 cm	1,25 cm	0,26 cm
Braço inclinado 20º à direita e antebraço 20º à esquerda	0,1389 cm	1,3355 cm	0,3855 cm
Braço reto e antebraço inclinado 20º à direita	0,1396 cm	1,3657 cm	0,2962 cm

4.2.2 Estratégia usando Transformada de Hough

Assim como realizado na Subseção 4.2.1, a análise dos resultados obtidos com a estratégia baseada na Transformada de Hough será conduzida individualmente para cada configuração do braço do paciente.

4.2.2.1 Braço inclinado 20º à esquerda e antebraço reto

Em primeira análise, considerando o braço inclinado em 20º para a esquerda e o antebraço reto, as duas imagens que evidenciam as retas obtidas por meio da Transformada de Hough são apresentadas na Figura 24.

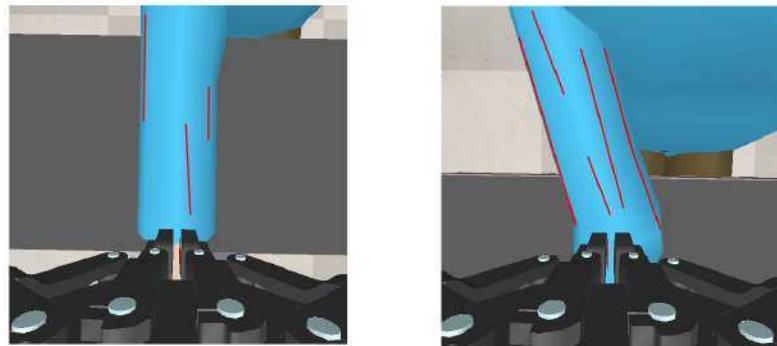


Figura 24 – Retas obtidas pela Transformada de Hough do braço inclinado 20º à esquerda e antebraço reto: (a) antebraço; (b) braço.

Fonte: Autor

Com base nas retas identificadas, foram definidas as retas finais pela média dos parâmetros ρ e θ das retas, conforme ilustrado na Figura 25.

Dessa forma, é possível comparar a trajetória planejada, em vermelho com a trajetória ideal na Figura 26, a fim de avaliar a efetividade da estratégia. As métricas de erro entre a trajetória ideal e a planejada estão dispostas na Tabela 6.

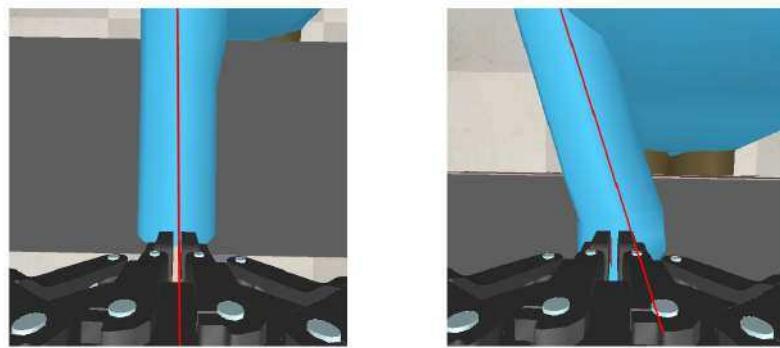


Figura 25 – Retas finais obtidas pela Transformada de Hough do braço inclinado 20º à esquerda e antebraço reto: (a) antebraço; (b) braço.

Fonte: Autor

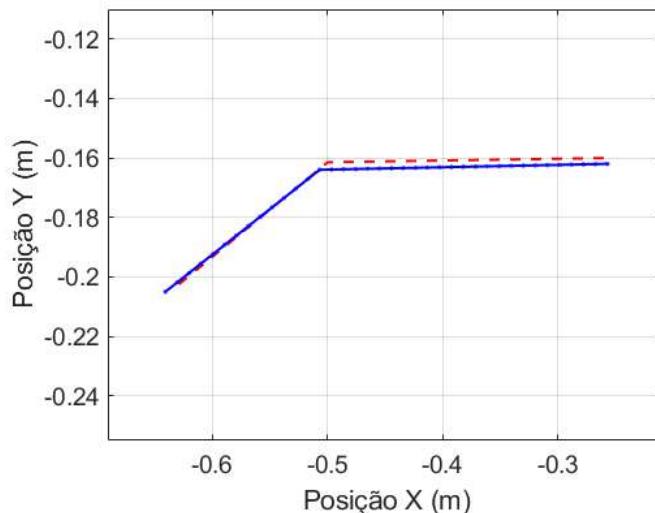


Figura 26 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita usando a Transformada de Hough.

Fonte: Autor

Assim, na Figura 27 é possível comparar a trajetória planejada (em vermelho) com a efetivamente executada (em azul) pelo manipulador robótico. As métricas de erro que evidenciam essa comparação encontram-se na Tabela 7.

4.2.2.2 Braço inclinado 20º à direita e antebraço 20º à esquerda

Em uma segunda análise, o paciente foi posicionado de modo que o braço se encontra inclinado em 20º para a direita e o antebraço em 20º para a esquerda. As retas identificadas por meio da Transformada de Hough estão ilustradas na Figura 28. A partir dessas retas, as linhas finais do braço e do antebraço foram determinadas pela média, conforme apresentado na Figura 29.

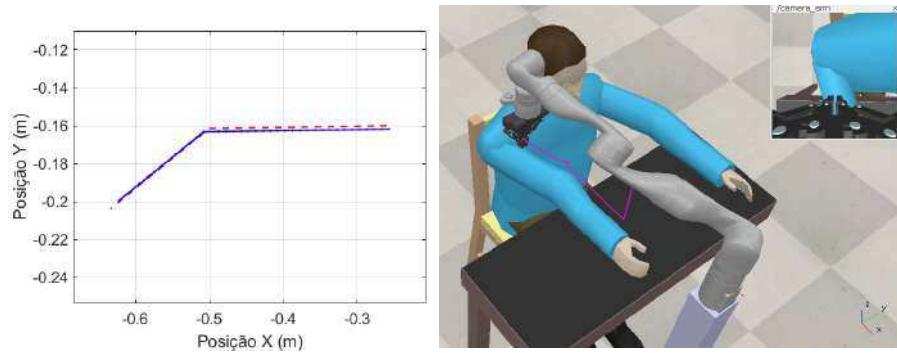


Figura 27 – Usando a Transformada de Hough: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à esquerda e antebraço reto; (b) Trajetória seguida na simulação

Fonte: Autor

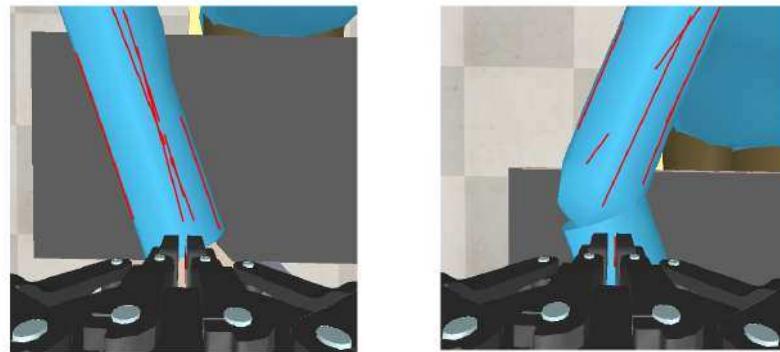


Figura 28 – Retas obtidas pela Transformada de Hough do braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço.

Fonte: Autor

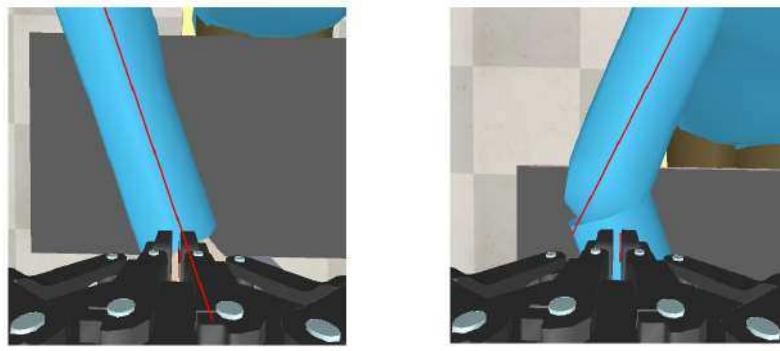


Figura 29 – Retas finais obtidas pela Transformada de Hough do braço inclinado 20º à direita e antebraço 20º à esquerda: (a) antebraço; (b) braço.

Fonte: Autor

A partir do coeficiente angular das retas, foi determinada a trajetória planejada (em vermelho), que é comparada com a trajetória ideal (em azul) na Figura 30. Neste caso, é possível identificar um desvio angular em relação à trajetória ideal, decorrente das características das retas obtidas. O erro entre as trajetórias está apresentado na Tabela 6.

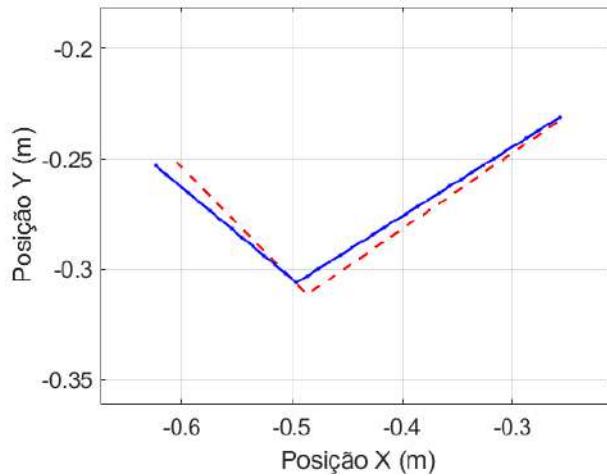


Figura 30 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20º à direita e antebraço 20º à esquerda.

Fonte: Autor

Em seguida, na Figura 31 pode-se observar a sobreposição da trajetória planejada (em vermelho) com a trajetória efetivamente executada (em azul) pelo manipulador robótico. As métricas de erro que evidenciam essa correspondência estão apresentados na Tabela 7.

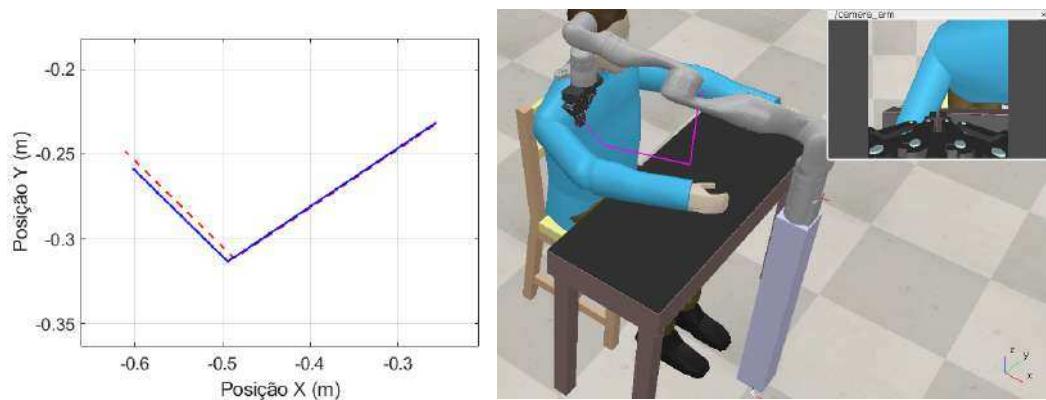


Figura 31 – Usando a Transformada de Hough: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à direita e antebraço 20º à esquerda; (b) Trajetória seguida na simulação.

Fonte: Autor

4.2.2.3 Braço reto e antebraço inclinado 20° à direita

Em terceira análise, o paciente foi posicionado de forma que o braço permanecesse reto e o antebraço ficasse inclinado em 20° para a direita. Nesse caso, as retas identificadas pela Transformada de Hough estão dispostas na Figura 32, enquanto as retas finais, obtidas pela média, estão apresentadas na Figura 33.

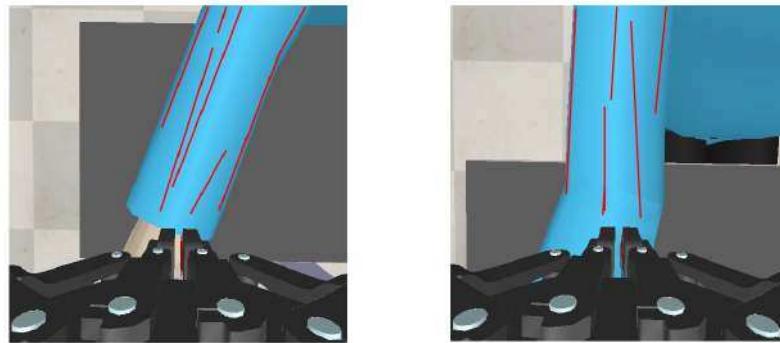


Figura 32 – Retas obtidas pela Transformada de Hough do braço reto e antebraço inclinado 20° à direita: (a) antebraço; (b) braço.

Fonte: Autor

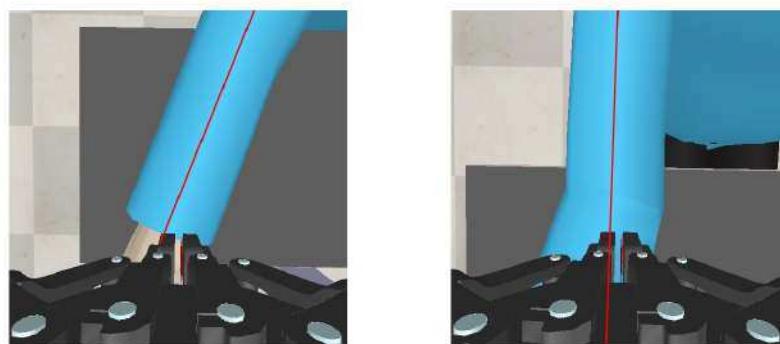


Figura 33 – Retas finais obtidas pela Transformada de Hough do braço reto e antebraço inclinado 20° à direita: (a) antebraço; (b) braço.

Fonte: Autor

Portanto, com as retas finais é possível obter a trajetória planejada (em vermelho) e compará-la com a trajetória ideal (em azul), conforme visto na Figura 34. O erro entre essas trajetórias está disposto na Tabela 6.

Por fim, concluindo a análise para as diferentes configurações desta estratégia, na Figura 35 é apresentada a trajetória efetivamente executada pelo manipulador (em azul) em comparação com a trajetória planejada (em vermelho), bem como o rastro (em

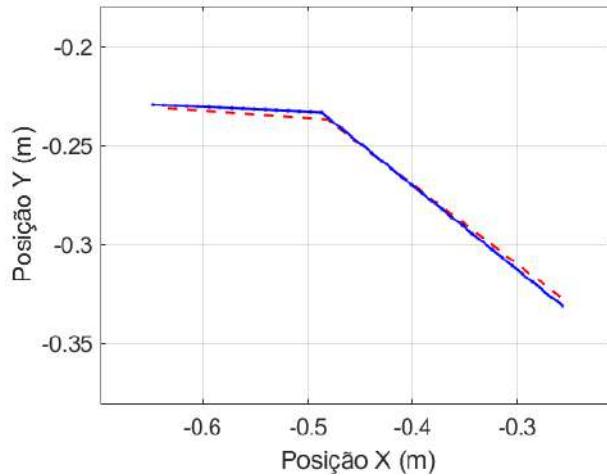


Figura 34 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita.

Fonte: Autor

magenta) deixado pelo manipulador no simulador. Os erros entre as trajetórias efetiva e planejada estão dispostos na Tabela 7.

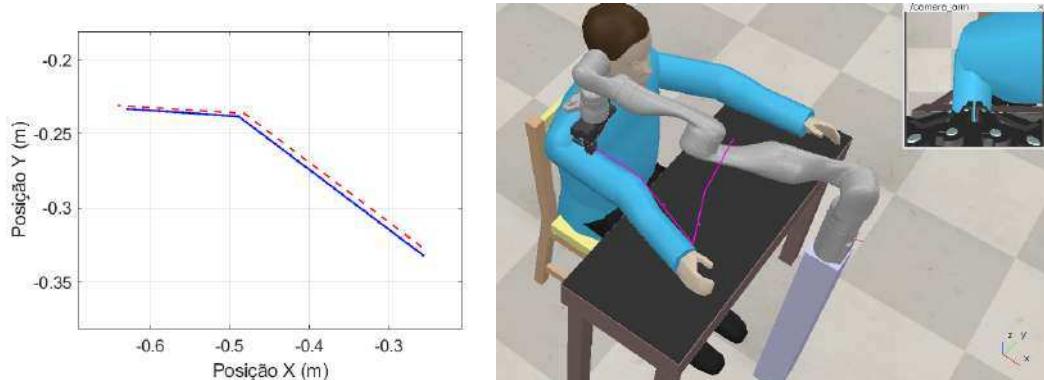


Figura 35 – Usando a Transformada de Hough: (a) Trajetória planejada em comparação com a trajetória executada com braço reto e antebraço inclinado 20º à direita; (b) Trajetória seguida na simulação.

Fonte: Autor

4.2.2.4 Comparativo entre configurações

Ao analisar a Tabela 6, que apresenta as métricas de erro utilizadas para comparar as trajetórias ideal e planejada nas três configurações distintas, observa-se uma melhoria na aproximação à trajetória ideal com a aplicação desta estratégia. Entretanto, o maior erro em relação ao ideal foi registrado na configuração em que o braço está inclinado em 20º para a direita e o antebraço em 20º para a esquerda, como mencionado anteriormente. Nesse caso, constatou-se um desvio angular maior.

Tabela 6 – Erro entre trajetória planejada e ideal em todas configurações utilizando a Transformada de Hough.

Configuração	RMSE	Média do erro	Desvio Padrão do erro
Braço inclinado 20º à esquerda e antebraço reto	0,0424 cm	0,3711 cm	0,1966 cm
Braço inclinado 20º à direita e antebraço 20º à esquerda	0,0707 cm	0,6299 cm	0,3321 cm
Braço reto e antebraço inclinado 20º à direita	0,0387 cm	0,3662 cm	0,1398 cm

Como resultado da aplicação do algoritmo de cinemática inversa, a Tabela 7 exibe os erros entre a trajetória planejada e a trajetória efetivamente seguida pelo manipulador robótico.

Tabela 7 – Erro entre trajetória planejada e executada pelo manipulador robótico em todas configurações utilizando a Transformada de Hough.

Configuração	RMSE	Média do Erro	Desvio Padrão do erro
Braço inclinado 20º à esquerda e antebraço reto	0,1360 cm	1,3049 cm	0,3869 cm
Braço inclinado 20º à direita e antebraço 20º à esquerda	0,1308 cm	1,2856 cm	0,2408 cm
Braço reto e antebraço inclinado 20º à direita	0,1539 cm	1,4849 cm	0,4139 cm

Assim, com base nos resultados obtidos, foi implementada uma estratégia alternativa para corrigir problemas associados ao uso da transformada de Hough, como desvios angulares superiores aos ideais.

4.2.3 Estratégia usando Detecção de Contornos

Conforme a análise das subseções anteriores, a análise dos resultados obtidos através da abordagem de detecção de contornos será realizada individualmente para cada configuração do braço do paciente.

4.2.3.1 Braço inclinado 20º à esquerda e antebraço reto

Inicialmente, para o caso em que o braço está inclinado 20º para a esquerda e o antebraço se encontra reto, a Figura 36 apresenta duas imagens evidenciando os contornos extraídos pelo algoritmo.

A partir dos contornos identificados, é possível identificar as retas finais na Figura 37.

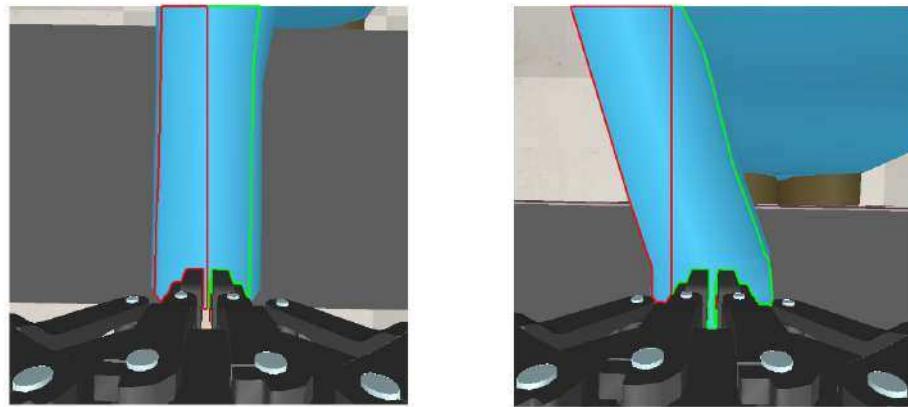


Figura 36 – Contornos do lado esquerdo e direito detectados pelo algoritmo com braço inclinado 20° à esquerda e antebraço reto: (a) antebraço; (b) braço.

Fonte: Autor

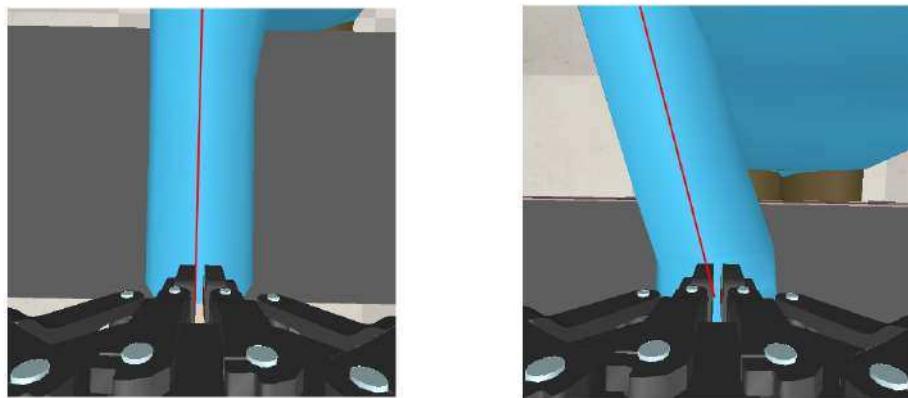


Figura 37 – Retas finais obtidas pela detecção de contornos do braço inclinado 20° à direita e antebraço 20° à esquerda: (a) antebraço; (b) braço.

Fonte: Autor

Dessa forma, foi possível comparar a trajetória planejada (em vermelho) com a trajetória ideal (em azul), como mostrado na Figura 38, permitindo avaliar o desempenho da abordagem. Os erros entre essas trajetórias estão detalhados na Tabela 8.

A Figura 39 mostra a comparação entre a trajetória planejada (em vermelho) e a executada (em azul) pelo manipulador robótico, sendo que as respectivas métricas de erro estão apresentadas na Tabela 9.

4.2.3.2 Braço inclinado 20° à direita e antebraço 20° à esquerda

Na segunda configuração, o braço do paciente foi posicionado com uma inclinação de 20° à direita, enquanto o antebraço inclinou-se 20° à esquerda. A Figura 40 apresenta

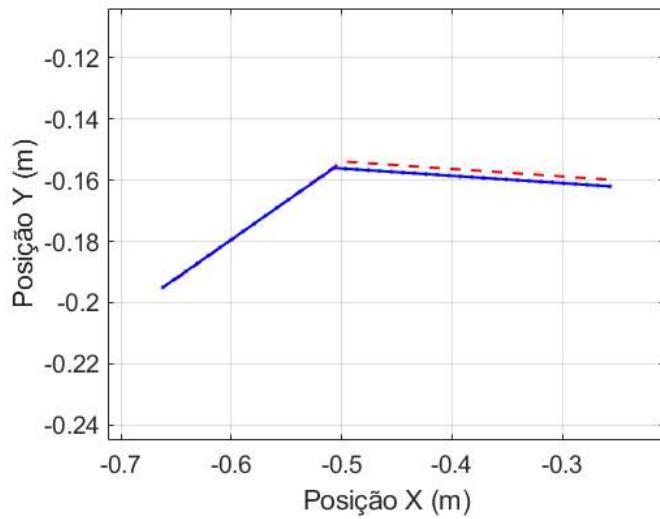


Figura 38 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20º à direita usando a detecção de contornos.

Fonte: Autor

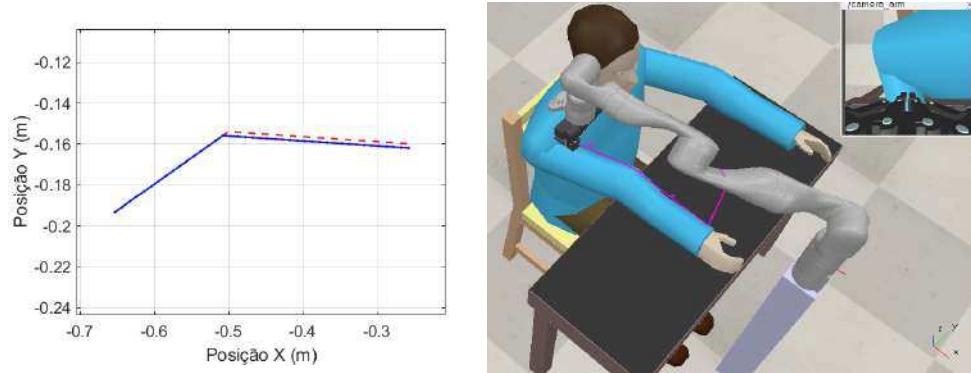


Figura 39 – Usando a detecção de contorno: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à esquerda e antebraço reto; (b) Trajetória seguida na simulação

Fonte: Autor

os contornos detectados pelo algoritmo, e a Figura 41 exibe as retas finais.

Utilizando os coeficientes angulares das linhas extraídas, foi gerada a trajetória planejada (em vermelho), posteriormente comparada à trajetória ideal (em azul) na Figura 42. A Tabela 8 apresenta os erros identificados.

A Figura 43 ilustra a sobreposição da trajetória planejada (em vermelho) com a executada (em azul) pelo manipulador, cujas métricas de erro estão dispostas na Tabela 9.

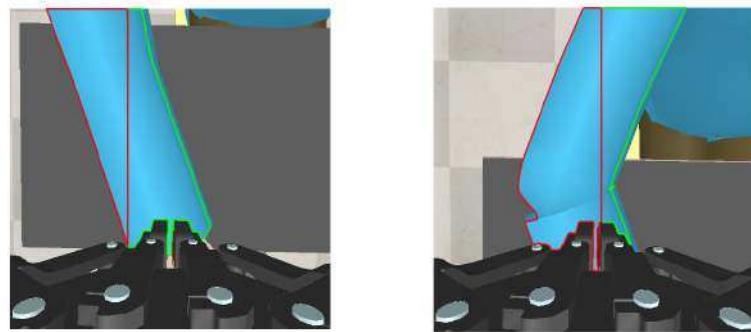


Figura 40 – Contornos do lado esquerdo e direito detectados pelo algoritmo com braço inclinado 20° à direita e antebraço 20° à esquerda: (a) antebraço; (b) braço.

Fonte: Autor

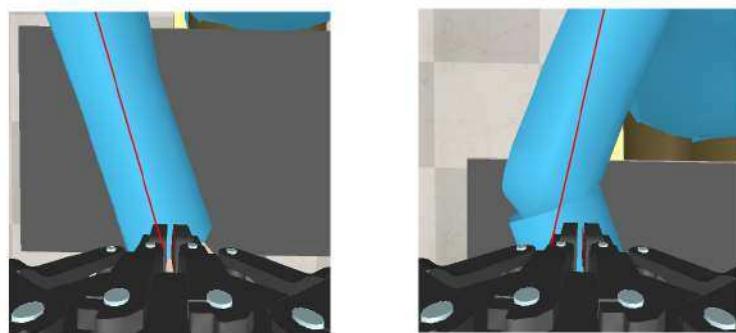


Figura 41 – Retas finais obtidas pela detecção de contornos do braço inclinado 20° à direita e antebraço 20° à esquerda: (a) antebraço; (b) braço.

Fonte: Autor

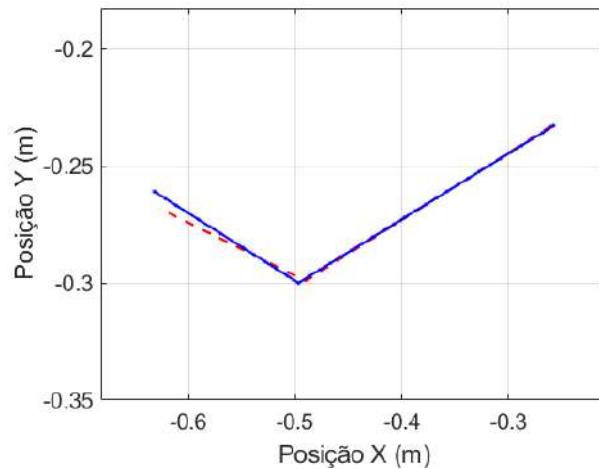


Figura 42 – Trajetória planejada em comparação com a trajetória ideal com braço inclinado 20° à direita e antebraço 20° à esquerda usando a detecção de contornos.

Fonte: Autor

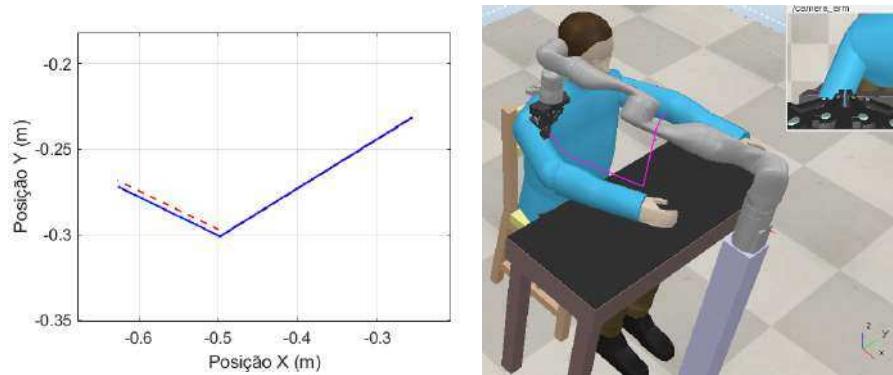


Figura 43 – Usando a detecção de contornos: (a) Trajetória planejada em comparação com a trajetória executada com braço inclinado 20º à direita e antebraço 20º à esquerda; (b) Trajetória seguida na simulação

Fonte: Autor

4.2.3.3 Braço reto e antebraço inclinado 20º à direita

Na terceira configuração, o braço do paciente permaneceu reto enquanto o antebraço se inclinava 20º para a direita. A Figura 44 exibe os contornos detectados, e a Figura 45 mostra a reta final.

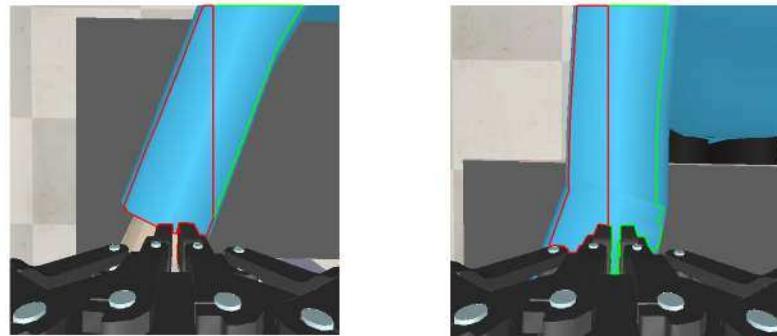


Figura 44 – Contornos do lado esquerdo e direito detectados pelo algoritmo com braço reto e antebraço inclinado 20º à direita: (a) antebraço; (b) braço.

Fonte: Autor

A trajetória planejada (em vermelho) foi então determinada e comparada à trajetória ideal (em azul), conforme representado na Figura 46, com os erros respectivos descritos na Tabela 8.

Para finalizar a análise, a Figura 47 apresenta a trajetória executada pelo manipulador (em azul) sobreposta à planejada, junto com o rastro (em magenta) registrado na simulação. Os erros entre a trajetória efetiva e a planejada estão listados na Tabela 9.

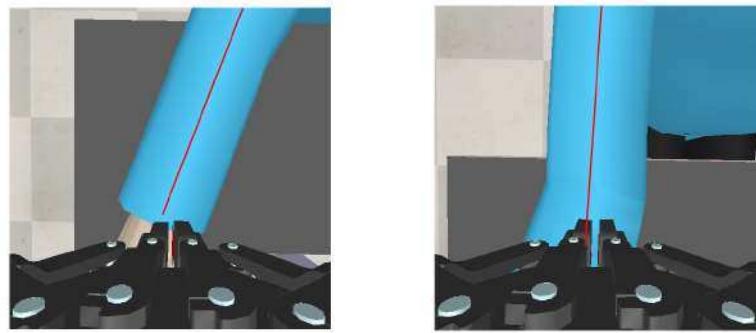


Figura 45 – Retas finais obtidas pela detecção de contornos do braço reto e antebraço inclinado 20° à direita: (a) antebraço; (b) braço.

Fonte: Autor

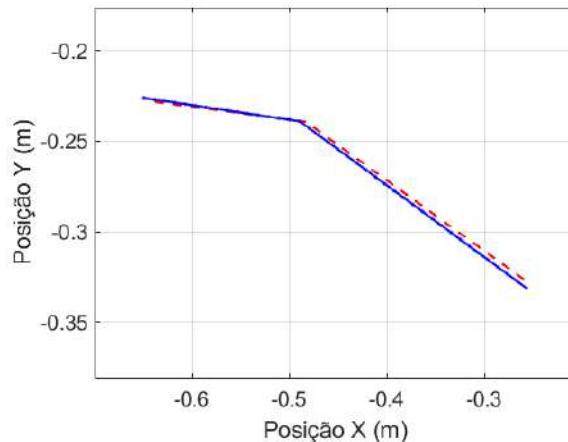


Figura 46 – Trajetória planejada em comparação com a trajetória ideal com braço reto e antebraço inclinado 20° à direita usando a detecção de contornos.

Fonte: Autor

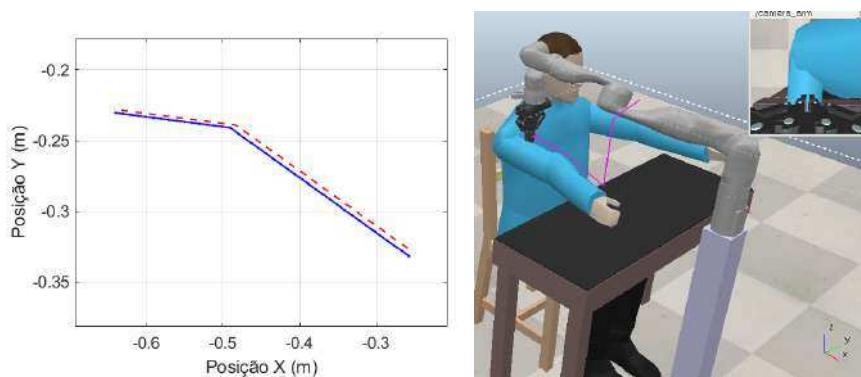


Figura 47 – Usando a detecção de contornos: (a) Trajetória planejada em comparação com a trajetória executada com braço reto e antebraço inclinado 20° à direita; (b) Trajetória seguida na simulação

Fonte: Autor

4.2.3.4 Comparativo entre configurações

A Tabela 8 apresenta as métricas de erro obtidas ao comparar a trajetória ideal com a planejada nas diferentes configurações, revelando uma aproximação mais fiel à trajetória ideal com o uso desta estratégia.

Tabela 8 – Erro entre trajetória planejada e ideal em todas configurações utilizando a detecção de contornos.

Configuração	RMSE	Média do erro	Desvio Padrão do erro
Braço inclinado 20º à esquerda e antebraço reto	0,0412 cm	0,3629 cm	0,1902 cm
Braço inclinado 20º à direita e antebraço 20º à esquerda	0,0412 cm	0,3359 cm	0,2322 cm
Braço reto e antebraço inclinado 20º à direita	0,0387 cm	0,3660 cm	0,1472 cm

Com a aplicação do algoritmo de cinemática inversa, a Tabela 9 fornece os erros resultantes da comparação entre a trajetória planejada e a efetivamente seguida pelo manipulador robótico.

Tabela 9 – Erro entre trajetória planejada e executada pelo manipulador robótico em todas configurações utilizando a detecção de contornos.

Configuração	RMSE	Média do Erro	Desvio Padrão do erro
Braço inclinado 20º à esquerda e antebraço reto	0,0300 cm	0,3629 cm	0,1902 cm
Braço inclinado 20º à direita e antebraço 20º à esquerda	0,0346 cm	0,3208 cm	0,1121 cm
Braço reto e antebraço inclinado 20º à direita	0,0510 cm	0,4976 cm	0,1103 cm

Com base nos dados apresentados nas tabelas, observa-se um erro reduzido e uma boa aderência à trajetória ideal ao aplicar essa estratégia. Assim, é interessante realizar uma análise comparativa para identificar a abordagem mais eficaz na geração das trajetórias.

4.3 Análise Comparativa das Estratégias

Para avaliar o desempenho de cada estratégia em função das distintas configurações do braço do paciente, foi elaborada a Tabela 10, na qual cada coluna apresenta o erro quadrático médio entre a trajetória planejada e a trajetória ideal.

Dessa forma, considerando o erro quadrático médio apresentado na Tabela 10, verifica-se que a estratégia baseada em detecção de contornos é a mais assertiva e apresenta

Tabela 10 – Comparaçāo entre o RMSE de cada estratégia adotada para vestir o paciente, considerando diferentes configurações de braço.

Configuração	Segmentação por Cor	Transformada de Hough	Detecção de Contornos
Braço inclinado 20º à esquerda e antebraço reto	0,1414 cm	0,0424 cm	0,0412 cm
Braço inclinado 20º à direita e antebraço 20º à esquerda	0,0520 cm	0,0707 cm	0,0412 cm
Braço reto e antebraço inclinado 20º à direita	0,0447 cm	0,0387 cm	0,0387 cm

maior acurácia. Entretanto, é importante destacar que as demais abordagens também demonstraram desempenho notável, visto que, embora apresentem erros um pouco superiores, na maioria das vezes elas se revelam como alternativas viáveis.

5 CONCLUSÕES

5.1 Conclusões

Neste trabalho, o objetivo central foi desenvolver uma solução assistiva baseada em algoritmos de visão computacional para o planejamento de trajetórias seguras e precisas na manipulação de vestimentas, visando promover maior autonomia e qualidade de vida para pessoas com deficiência, idosos e indivíduos com mobilidade reduzida. Para atingir esse objetivo, adotou-se uma abordagem que combinou os fundamentos da robótica clássica com técnicas avançadas de processamento digital de imagens, permitindo a análise da configuração do braço do paciente e o planejamento de trajetórias otimizadas.

Com base na fundamentação teórica e na análise da tarefa e do cenário, foram adotadas três estratégias baseadas em visão computacional, conforme visto no Capítulo 3: segmentação por cor, transformada de Hough e detecção de contornos. Essas abordagens tiveram como objetivo identificar e descrever as retas mais adequadas referentes ao braço e ao antebraço, a fim de viabilizar a vestimenta da manga pelo robô. Ademais, por meio das imagens capturadas pela câmera com sensor de profundidade (sensor RGB-D), instalada no manipulador robótico, foi possível detectar as diferentes configurações do braço e, consequentemente, adaptar a trajetória planejada à sua posição real.

Um dos objetivos do algoritmo foi se aproximar do braço de forma segura, o que é essencial na tarefa de vestir. Com o método proposto, foi possível verificar os resultados de distância final da garra do manipulador robótico em relação ao braço na Seção 4.1. Mesmo em diferentes configurações, o limiar de segurança foi mantido, com as seguintes distâncias finais: 5,9935 cm para o braço inclinado 20° à esquerda com antebraço reto; 5,8527 cm para o braço inclinado 20° à direita com antebraço 20° à esquerda; e 5,9739 cm para o braço reto com antebraço inclinado 20° à direita.

Em relação às estratégias para geração de trajetórias, a Tabela 10 evidencia que os baixos valores de RMSE validam a eficácia das três abordagens para a tarefa de vestir o paciente. Entretanto, em alguns casos, como na segmentação por cor, foi observado um desvio angular fora do esperado, conforme demonstrado pelos resultados da Subseção 4.2.1 na primeira configuração.

Considerando a Tabela 10, utilizando a melhoria percentual entre os RMSEs, observa-se que, na primeira configuração, a estratégia de detecção de contornos apresentou uma acurácia 91,5% superior à segmentação por cor e 5,56% maior do que a abordagem baseada na Transformada de Hough. Na segunda configuração, a detecção de contornos demonstrou uma melhoria de 37,04% em relação à segmentação por cor e 66% em comparação à Transformada de Hough. Por fim, na terceira configuração, essa estratégia obteve um ganho de 25% de acurácia em relação à segmentação por cor, mantendo o mesmo

desempenho da Transformada de Hough. Portanto, a abordagem baseada na detecção de contornos mostrou, de maneira geral, maior acurácia em relação à trajetória ideal.

5.2 Trabalhos Futuros

Como trabalho futuro, e considerando os bons resultados obtidos na simulação, propõe-se a implementação da geração de trajetórias em um robô manipulador Kinova 7 DoF em ambiente físico, utilizando inicialmente um manequim para validar os resultados e priorizar a segurança do paciente. Nessa etapa, será fundamental investigar como o algoritmo se comporta com imagens reais e identificar as adaptações necessárias para uma transferência efetiva da simulação para o mundo real (*sim-to-real*), especialmente diante de desafios como oclusão e variações na iluminação.

Além disso, é importante testar o algoritmo com a manga da vestimenta para avaliar seu comportamento, pois, conforme discutido no Capítulo 2, alguns problemas podem surgir na aplicação real. Ademais, seria relevante investigar os pontos de preensão (*grasping points*) da vestimenta para aumentar a autonomia do sistema, eliminando a necessidade de posicioná-la inicialmente na garra. Essa simplificação foi adotada no presente trabalho devido às limitações da simulação e ao objetivo específico do algoritmo.

Outras pesquisas podem ser feitas a respeito da utilização de técnicas de *deep learning* para tratar, em tempo real, as variações na posição e conformação do braço durante o processo de vestimenta.

REFERÊNCIAS

- ALLIN, S. et al. Recent trends in the development and evaluation of assistive robotic manipulation devices. *Physical Medicine and Rehabilitation Clinics*, Elsevier, v. 21, n. 1, p. 59–77, 2010.
- BELLAMY, D. D. et al. Safety assessment review of a dressing assistance robot. *Frontiers in Robotics and AI*, Frontiers Media SA, v. 8, p. 667316, 2021.
- CANNY, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, n. 6, p. 679–698, 1986.
- CHUNG, C.-S.; WANG, H.; COOPER, R. A. Functional assessment and performance evaluation for assistive robotic manipulators: Literature review. *The journal of spinal cord medicine*, Taylor & Francis, v. 36, n. 4, p. 273–289, 2013.
- DUDGEON, B. J. et al. Managing activity difficulties at home: a survey of medicare beneficiaries. *Archives of physical medicine and rehabilitation*, Elsevier, v. 89, n. 7, p. 1256–1261, 2008.
- EMPRISELAB, C. *PhyRC Challenge*. 2024. Disponível em <<https://emprise.cs.cornell.edu/rcareworld/challenge/>> Acessado em 09 fev. 2025.
- ERICKSON, Z. et al. Tracking human pose during robot-assisted dressing using single-axis capacitive proximity sensing. *IEEE Robotics and Automation Letters*, IEEE, v. 3, n. 3, p. 2245–2252, 2018.
- FORSYTH, D. A.; PONCE, J. A modern approach. *Computer vision: a modern approach*, Prentice-Hall, v. 17, p. 21–48, 2003.
- GAO, Y.; CHANG, H. J.; DEMIRIS, Y. User modelling for personalised dressing assistance by humanoid robots. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2015. p. 1840–1845.
- GAO, Y.; CHANG, H. J.; DEMIRIS, Y. Iterative path optimisation for personalised dressing assistance using vision and force information. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2016. p. 4398–4403.
- GONZALEZ, R.; WOODS, R. *Digital Image Processing*. Pearson, 2018. ISBN 9780133356724. Disponível em: <<https://books.google.com.br/books?id=0F05vgAACAAJ>>.
- HARTENBERG, R. S.; DENAVIT, J. *Kinematic Synthesis of Linkages*. New York: McGraw-Hill, 1964.
- HORN, B. K.; SCHUNCK, B. G. Determining optical flow. *artificial intelligence*. 1981.
- HOUGH, P. V. *Method and means for recognizing complex patterns*. [S.l.]: Google Patents, 1962. US Patent 3,069,654.
- IBGE. *Pesquisa nacional de saúde : 2019 : ciclos de vida : Brasil / IBGE, Coordenação de Trabalho e Rendimento, [Ministério da Saúde]*. 2021. Disponível em <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv101846.pdf>> Acessado em 08 fev. 2025.

- INTELLIGENCE, M. *Tamanho do mercado de robótica assistiva e análise de ações – Tendências e previsões de crescimento (2024 – 2029)*. 2023. Disponível em <<https://www.mordorintelligence.com/pt/industry-reports/assistive-robotics-market>> Acessado em 08 fev. 2025.
- KAPUSTA, A. et al. Data-driven haptic perception for robot-assisted dressing. In: IEEE. *2016 25th IEEE international symposium on robot and human interactive communication (RO-MAN)*. [S.l.], 2016. p. 451–458.
- KIM, D.-J. et al. System design and implementation of ucf-manus—an intelligent assistive robotic manipulator. *IEEE/ASME Transactions on Mechatronics*, v. 19, n. 1, 2014.
- KINOVA. *Kinova Gen3 Ultra lightweight robot User Guide*. 2022. Disponível em <<https://www.kinovarobotics.com/uploads/User-Guide-Gen3-R07.pdf>> Acessado em 27 fev. 2025.
- KINOVA. *Kinova Gen3 Robots 7 DoF*. 2024. Disponível em <<https://www.kinovarobotics.com/product/gen3-robots>> Acessado em 23 fev. 2025.
- KLEE, S. D. et al. Personalized assistance for dressing users. In: SPRINGER. *Social Robotics: 7th International Conference, ICSR 2015, Paris, France, October 26-30, 2015, Proceedings* 7. [S.l.], 2015. p. 359–369.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.
- LI, S. et al. Provably safe and efficient motion planning with uncertain human dynamics. *Robotics: Science and Systems Foundation*, 2021.
- LOOS, H. M. Van der et al. Provar assistive robot system architecture. In: IEEE. *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*. [S.l.], 1999. v. 1, p. 741–746.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Springer, v. 60, p. 91–110, 2004.
- MAITLO, N. et al. *Color Recognition in Challenging Lighting Environments: CNN Approach*. 2024. Disponível em: <<https://arxiv.org/abs/2402.04762>>.
- MAN, D.; VISION, A. A computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company, San Francisco*, v. 1, p. 1, 1982.
- MITZNER, T. L. et al. Identifying the potential for robotics to assist older adults in different living environments. *International journal of social robotics*, Springer, v. 6, p. 213–227, 2014.
- NANAVATI, A.; RANGANENI, V.; CAKMAK, M. Physically assistive robots: A systematic review of mobile and manipulator robots that physically assist people with disabilities. *Annual Review of Control, Robotics, and Autonomous Systems*, Annual Reviews, v. 7, 2023.

RETHINK. *Rethink Robotics Baxter User Guides*. 2022. Disponível em <<https://support.rethinkrobotics.com/support/solutions/articles/80000976917-user-guides>> Acessado em 23 fev. 2025.

ROHMER, E.; SINGH, S. P.; FREESE, M. Vrep: A versatile and scalable robot simulation framework. In: IEEE. *2013 IEEE/RSJ international conference on intelligent robots and systems*. [S.l.], 2013. p. 1321–1326.

SHOTTON, J. et al. Real-time human pose recognition in parts from single depth images. In: IEEE. *CVPR 2011*. [S.l.], 2011. p. 1297–1304.

SMITH, A. R. Color gamut transform pairs. *SIGGRAPH Comput. Graph.*, Association for Computing Machinery, New York, NY, USA, n. 3, p. 12–19, ago. 1978. ISSN 0097-8930. Disponível em: <<https://doi.org/10.1145/965139.807361>>.

SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Nature, 2022.

TAMEI, T. et al. Reinforcement learning of clothing assistance with a dual-arm robot. In: IEEE. *2011 11th IEEE-RAS International Conference on Humanoid Robots*. [S.l.], 2011. p. 733–738.

VERBURG, G. et al. Manus: The evolution of an assistive technology. *Technology and Disability*, IOS Press, v. 5, n. 2, p. 217–228, 1996.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. [S.l.], 2001. v. 1, p. I–I.

WHO, W. H. O. *Disability*. 2023. Disponível em <<https://www.who.int/news-room/fact-sheets/detail/disability-and-health>> Acessado em 12 fev. 2025.

YE, R. et al. Rc care world: A human-centric simulation world for caregiving robots. In: IEEE. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2022. p. 33–40.

ZHANG, F.; CULLY, A.; DEMIRIS, Y. Personalized robot-assisted dressing using user modeling in latent spaces. In: IEEE. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2017. p. 3603–3610.

ZHANG, F.; DEMIRIS, Y. Learning grasping points for garment manipulation in robot-assisted dressing. In: IEEE. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.], 2020. p. 9114–9120.