



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

Controle de um Braço Robótico Articulado com 6 Graus de Liberdade

Eduardo Matias Rodrigues

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Theo Silva Lins

**Janeiro, 2022
João Monlevade–MG**

Eduardo Matias Rodrigues

Controle de um Braço Robótico Articulado com 6 Graus de Liberdade

Orientador: Theo Silva Lins

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Janeiro de 2022



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS



FOLHA DE APROVAÇÃO

Eduardo Matias Rodrigues

Controle de um braço robótico articulado com 6 graus de liberdade

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação

Aprovada em 13 de Janeiro de 2022

Membros da banca

Doutor Theo Silva Lins - Orientador(a) - Universidade Federal de Ouro Preto - DECSI
Doutor Marlon Paolo Lima - Universidade Federal de Ouro Preto - DECSI
Doutor Samuel Souza Brito - Universidade Federal de Ouro Preto - DECSI

Theo Silva Lins, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 27/01/2022



Documento assinado eletronicamente por **Theo Silva Lins, PROFESSOR DE MAGISTERIO SUPERIOR**, em 27/01/2022, às 22:39, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0272917** e o código CRC **BFD79081**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.001013/2022-15

SEI nº 0272917

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: - www.ufop.br

Este trabalho é dedicado a todos que me apoiaram e ajudaram nesta caminhada.

Agradecimentos

Agradeço aos meus pais, Cláudia e Ermelindo, que me ensinaram alguns dos valores mais importantes que sempre carregarei, por estarem presentes em todos os momentos, bons e nos difíceis. Aos meus irmãos, Gustavo e Alice, que me acompanharam e deram apoio nessa jornada. A todos os familiares, que torceram por mim, em especial aos meus tios Moisés e Eduardo (in memoriam), que sempre estiveram presentes em minha vida.

Durante esses anos, tive a oportunidade de fazer vários amigos. Fica aqui, meu agradecimento aos companheiros que me ajudaram e apoiaram nessa caminhada.

“Lembre-se de olhar para o alto, para as estrelas, e não para baixo, para os seus pés.”

— Stephen Hawking (1942 – 2018)

Resumo

O aprimoramento de métodos computacionais apoiados pelos avanços na física, matemática e engenharia permitiu o processamento de informações complexas em tempo hábil. A disseminação de dispositivos do cotidiano capazes de coletar, analisar e distribuir informações em grandes escalas através da Internet (Internet das Coisas, do inglês - *Internet of Things*) é a próxima fase para aprimoramento dos processos industriais, pessoais e acadêmicos. No campo da robótica não foi diferente. Devido aos desafios, multidisciplinaridade e aplicações práticas, diversos trabalhos têm sido desenvolvidos na área trazendo novidades a cada nova publicação. Devido aos custos elevados dos equipamentos e a rápida obsolescência dos robôs, temos cada vez mais instrumentos mecanicamente funcionais, mas com estruturas lógicas de controle defasadas em relação ao cenário atual. Neste trabalho foi desenvolvido um controlador para o manipulador robótico articulado ED-7220C com base no microcomputador BeagleBone Blue para substituir seu controlador padrão, por este não possuir tecnologias modernas para operação do equipamento. A plataforma desenvolvida permite que trabalhos futuros possam ser realizados, por se tratar de um ambiente expansível, capaz de alocar mudanças com facilidade.

Palavras-chaves: Robótica. BeagleBone Blue. Controlador. Internet das Coisas.

Abstract

The technological advance has allowed the complex information processing in due time and the integration between everyday devices and the internet (Internet of Things), providing progress in different areas of knowledge. The robotic field was not different. Due to the challenges, multidisciplinary, and practical applications, several studies have been developed in this area, bringing news to every new publication. Because of the high-cost equipment and the fast robot obsolescence, there are much more functional mechanical instruments, but controlled logical structures lagged in relation to the current scenario. In this study was developed a controller for the manipulator robotic arm ED-7220c based on the microcomputer BeagleBone Blue to replace the standard controller because this one does not have modern technology to operate the equipment. The developed board allows future studies to be done through it, since the expansive environment is able to allocate changes easily.

Key-words: Robotic. BeagleBone Blue. Controller. Internet of Things.

Lista de ilustrações

Figura 1 – Controlador ED-MK4	14
Figura 2 – ED-7220C	16
Figura 3 – Módulos	17
Figura 4 – Configuração do ED-7220C	18
Figura 5 – ED 7220C - Movimentos	18
Figura 6 – Micro switch	19
Figura 7 – Cinemática Direta	20
Figura 8 – Cinemática Inversa	23
Figura 9 – BeagleBone Blue	24
Figura 10 – Ponte-H	25
Figura 11 – Câmera - Logitech 720p	26
Figura 12 – Imagem - Espaço de trabalho	27
Figura 13 – Socket	28
Figura 14 – Pinos ED-7220C - Vista frontal	30
Figura 15 – BeagleBone Blue - Pinos	31
Figura 16 – ATmega168/328P - Pinos	31
Figura 17 – Objeto prata	35
Figura 18 – Objeto preto	36
Figura 19 – Posição de descanso	39
Figura 20 – Capturando o objeto	40
Figura 21 – Posição intermediária	41
Figura 22 – Movendo o objeto para posição definida	41
Figura 23 – Retorno a posição de descanso	42
Figura 24 – Detecção automática dos objetos	43
Figura 25 – Circuito: BeagleBone Blue	48
Figura 26 – Circuito: AtMega328p	48

Lista de tabelas

Tabela 1 – ED-MK4: Características	14
Tabela 2 – ED-7220C: Características	20
Tabela 3 – Dimensões do braço	20
Tabela 4 – Parâmetros DH	22
Tabela 5 – Limite mecânico	32
Tabela 6 – Pulsos de <i>encoder</i>	33
Tabela 7 – Posições iniciais	37
Tabela 8 – Resultados da cinemática inversa	38
Tabela 9 – Posição inicial para o ponto 1	38
Tabela 10 – Ponto 1 para ponto 2	38
Tabela 11 – Ponto 2 para ponto 3	39

Lista de abreviaturas e siglas

BBBlue *BeagleBone Blue*

DH *Denavit-Hartenberg*

DoF *Degree of Freedom*

FK *Forward Kinematic*

IK *Inverse Kinematic*

IoT *Internet of Things*

RCL *Robot Control Library*

RCPY *Robotics Cape Python*

TCP *Transmission Control Protocol*

UDP *User Datagram Protocol*

Sumário

1	INTRODUÇÃO	13
1.1	O Problema de pesquisa	13
1.2	Objetivos	14
1.3	Metodologia	15
1.4	Organização do trabalho	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	ED-7220C	17
2.2	Cinemática	19
2.2.1	Cinemática Direta	19
2.2.2	Denavit-Hartenberg	20
2.2.3	Cinemática Inversa	22
2.3	BeagleBone Blue	24
2.4	AtMega328	25
2.5	Visão Computacional	26
2.6	Interface <i>WEB</i>	27
2.7	Trabalhos relacionados	28
3	DESENVOLVIMENTO	30
3.1	Robotics Cape Python	32
3.2	ED7220-C	32
3.3	Interface <i>WEB</i>	34
3.4	Deteccção de objetos	35
4	RESULTADOS	37
4.1	Cenário de teste	37
4.2	Detectando objetos	39
5	CONCLUSÃO	44
	REFERÊNCIAS	45
	APÊNDICES	47
	APÊNDICE A – CIRCUITO ELETRÔNICO	48

1 Introdução

Braços robóticos ficam em uso por longos períodos de tempo. Entretanto, devido ao elevado custo desses equipamentos, a aquisição de robôs com tecnologias atuais por instituições de ensino e empresas de pequeno porte é inviável. Manipuladores robóticos são vendidos pelas empresas proprietárias acompanhados de diversos acessórios e dispositivos, como o controlador padrão do equipamento, que devido a falta de padronização da área é programado em uma linguagem própria, semelhante ao Assembly e pode variar entre modelos de robôs da mesma empresa, dificultando o desenvolvimento de novos projetos com o sistema.

Todos dispositivos adquiridos junto ao manipulador robótico com a empresa fabricante não possuem código fonte aberto, adicionando mais um nível de dificuldade no desenvolvimento de novos incrementos para o robô. Encontramos braços robóticos com boa estrutura mecânica (elos, motores, engrenagens e correias, por exemplo) capaz de garantir a corretude do movimento, mas cuja lógica de controle ultrapassa os 10 anos (ALVARES et al., 2018). Portanto, para poupar recursos na aquisição de novos robôs e todos seus periféricos, é necessário que haja um trabalho de *retrofitting* nesses equipamentos, permitindo que em seus sistemas sejam possíveis a adição de qualquer nova tecnologia.

Neste trabalho é descrito o processo de *retrofitting* do manipulador robótico ED-7220C, apresentado na Figura 2. Foi desenvolvido uma plataforma de controle alternativa utilizando o computador *all-in-one BeagleBone Blue* (BBBlue). O novo *hardware* e *software* cumpre o propósito de ser uma alternativa moderna e expansível para o controle do ED-7220C.

1.1 O Problema de pesquisa

O braço robótico ED-7220C, apresentado na Figura 2, possui como controlador padrão o ED-MK4, conforme apresentado na Figura 1. Toda a comunicação entre o manipulador robótico e o controlador é feita através de 8 cabos *flats*, não sendo possível a conexão com a internet ou a adição de novos componentes de *hardware*, como sensores. As características do controlador podem ser vistas com detalhes na Tabela 1.

A falta de conexão com a internet, visão computacional e a possibilidade de adição de novos sensores tornam o controlador ED-MK4 limitado quanto às possibilidades de desenvolvimento. Portanto, neste trabalho foi desenvolvido um controlador para o braço robótico ED-7220C utilizando o computador de baixo custo *all-in-one* BBBlue (INSTRUMENTS, 2018). A plataforma desenvolvida expande as funcionalidades do robô

Figura 1 – Controlador ED-MK4



Fonte: ([ED-LABORATORY, s.d.](#))

Tabela 1 – ED-MK4: Características

Componente	Descrição
Terminais de entrada	8 portas e LED
Chave de entrada	8 Portas do tipo <i>Toggle Switch</i>
Terminais de saída	8 portas e LED
Processador	Processador de 16 <i>Bits</i> , 8 <i>Bits</i> para controle dos motores, 8 <i>Bits</i> para o <i>Teach Pendant</i>
<i>Software</i>	ED-72C RoboTalk
Tensão de entrada	220V, 50/60Hz
Dimensões	385 Largura x 150 Altura 460 Espessura mm
Peso	14,3 Kg

Fonte: ([ED-LABORATORY, s.d.](#))

e o insere no cenário da *Internet of Things (IoT)*.

O BBBlue é um computador baseado em Linux, projetado para realizar o controle de robôs. Esta plataforma de desenvolvimento dispõe de todos os recursos necessários para realizar o controle do braço robótico, além de fornecer suporte para expansão de suas funcionalidades, como conexão com a internet, entrada para sensores analógicos ou digitais.

1.2 Objetivos

O objetivo principal do presente trabalho é realizar o processo de *retrofitting* no manipulador robótico ED-7220C desenvolvendo uma plataforma de controle moderna para o equipamento. A partir da plataforma criada, deve ser possível adicionar soluções de IoT no robô. Para trabalhos futuros, novos dispositivos, como sensores de colisão, podem ser incorporados no sistema.

1.3 Metodologia

O objeto de estudo deste trabalho é o ED-7220C. Para realizar o trabalho de *retrofitting* é necessário conhecer a sua modelagem cinemática e suas métricas reais de posicionamento no espaço, aferidas em laboratório. Para o desenvolvimento da nova plataforma de controle, também foi necessário o estudo do funcionamento do computador BBBlue e todos periféricos utilizados no projeto.

O primeiro passo na execução desta pesquisa foi a realização de uma revisão bibliográfica de trabalhos relacionados ao comportamento cinemático do robô estudado e trabalhos relacionados a *retrofitting* de manipuladores robóticos. Posteriormente, foi feita uma análise da estrutura mecânica e elétrica do ED-7220C e análise das *interfaces* do BBBlue e suas possibilidades de desenvolvimento. Conhecendo o funcionamento dos equipamentos, foi realizada uma aferição dos dados reais de ângulos e pulsos de *encoder* do manipulador robótico. Em seguida, foi realizada a validação dos modelos cinemáticos estudados na literatura através de testes em laboratório e escolhido o modelo que apresentou melhor acurácia. Um modelo é dito acurado se ele sempre é capaz de converter um ponto no espaço cartesiano dentro do *workspace* do braço em ângulos de junta do manipulador.

1.4 Organização do trabalho

A disposição do trabalho está organizado da seguinte forma: Capítulo 2 faz uma revisão de toda fundamentação teórica, Capítulo 3 descreve o procedimento para construção do controlador desenvolvido, Capítulo 4 apresenta os resultados obtidos e as Considerações Finais são apresentadas no Capítulo 5.

2 Revisão bibliográfica

Para entender a complexidade do campo da robótica é necessário conhecimento de diferentes áreas, como matemática, eletrônica, computação e mecânica. O braço robótico ED-7220C, apresentado na Figura 2 e utilizado neste trabalho, é um exemplo clássico de um complexo sistema eletromecânico operado por um sistema computacional ([AMINE; NADJAH, 2017](#)).

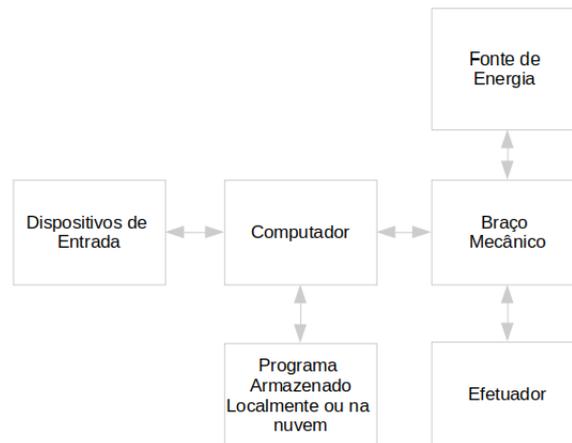
Figura 2 – ED-7220C



Fonte: ([ED-LABORATORY, s.d.](#))

Para operar um sistema robótico é necessário a comunicação entre diferentes componentes. Na Figura 3, temos a descrição de todos os elementos necessários e utilizados no projeto. Constituídos por um computador ou controlador, o programa desenvolvido para realizar o controle, fonte de energia, dispositivo de entrada e um efetuator (AMINE; NADJAH, 2017).

Figura 3 – Módulos



Fonte: elaborado pelo autor

O desafio do presente trabalho é desenvolver um sistema de controle acurado e expansível para o manipulador robótico, para que sirva de base a projetos futuros na área da IoT. A seguir serão detalhados todos os componentes utilizados nessa pesquisa.

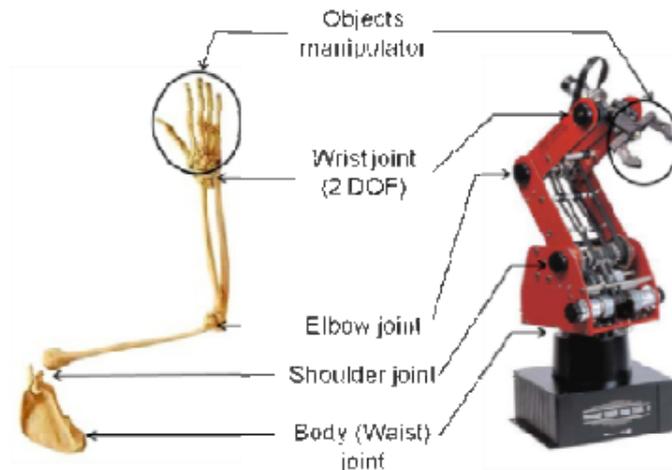
2.1 ED-7220C

Um manipulador robótico articulado é composto de elos e *links*, simulando um braço humano. Cada manipulador robótico articulado é classificado quanto aos seus Graus de Liberdade (do inglês *Degree of Freedom (DoF)*), que é o número de parâmetros independentes que definem o estado do braço.

O braço ED-7220C utilizado foi desenvolvido pela empresa coreana ED-Laboratory com fins exclusivamente didáticos. Assim como um braço humano, ele possui 6 graus de liberdade. Na Figura 4, vê-se com detalhes a correlação entre suas configurações mecânicas.

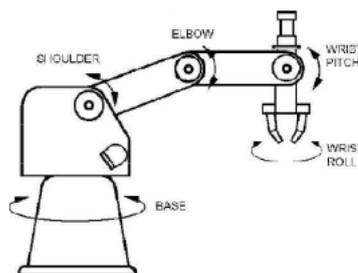
O braço é composto por base, ombro, cotovelo, pulso e um efetuator ou manipulador. O tipo do efetuator acoplado depende do propósito ao qual o equipamento é destinado, como mover objetos, soldar componentes eletrônicos ou mesmo fazer impressão 3D. Com exceção do pulso, todas as juntas possuem 1 *DoF*. O pulso possui 2 *DoF* para executar o movimento de *pitch* e *roll*, descrito na Figura 5, permitindo a garra realizar o movimento de rotação e translação.

Figura 4 – Configuração do ED-7220C



Fonte: (IQBAL; ISLAM; KHAN, 2012)

Figura 5 – ED 7220C - Movimentos



Fonte: (SENA, 2011)

O movimento do braço é realizado fazendo com que cada motor atinja uma determinada posição. A base, ombro e cotovelo são movimentadas cada uma por um motor DME38B50G-116. Para executar o movimento de *roll* e *pitch*, o pulso conta com dois motores DME38B50G-115 e a garra é movimentada pelo motor DME33B37G-171 (MANZOOR et al., 2014). Cada motor é equipado com um *encoder* óptico cujo sinal é utilizado como entrada na unidade controladora, permitindo a realização do controle em malha fechada. O braço também conta com limitadores mecânicos, necessários para definição da posição de descanso do robô ou mesmo como alerta caso haja alguma falha no controle do equipamento.

Em cada limitador há uma chave micro *switch* - Figura 6 - que informa ao controlador quando o respectivo limite mecânico foi atingido. Na Tabela 2, pode-se ver as especificações físicas do equipamento, utilizadas para definição inicial dos parâmetros do algoritmo de controle.

Figura 6 – Micro switch



Fonte: ([ELETRONICA, 2014](#))

2.2 Cinemática

Para entender como o manipulador robótico se movimenta é necessário conhecer seu comportamento cinemático. A cinemática é utilizada para descrever o movimento do robô, desconsiderando as forças que atuam sobre ele. Nessa área são modeladas as equações de transformação entre os ângulos de juntas do robô e a posição do efetuador final no espaço cartesiano. Elas são divididas em duas categorias: Cinemática Direta (do inglês *Forward Kinematic* (FK)) e Cinemática Inversa (do inglês *Inverse Kinematic* (IK)).

2.2.1 Cinemática Direta

A cinemática direta de um robô manipulador é utilizada para obter a posição do efetuador no espaço cartesiano uma vez que sabemos os ângulos de cada junta do robô, conforme visto na Figura 7. Utilizando o algoritmo *Denavit-Hartenberg* (DH) (CABRAL, ca. 2010) e sabendo as dimensões e limites mecânicos de cada *link* é possível chegar à matriz de cinemática direta de qualquer manipulador robótico serial. Uma matriz de cinemática direta nos dá a posição e orientação do efetuador final em relação a base.

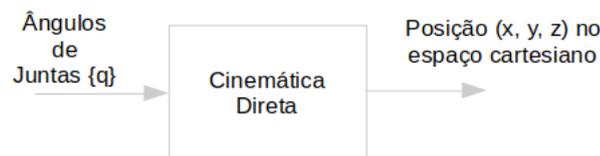
Para construção da matriz de transformação homogênea utilizando o método DH é necessário saber os ângulos mínimos e máximos do manipulador, descritos na Tabela 2, como também, as dimensões de cada *link* do robô, descritas na Tabela 3.

Tabela 2 – ED-7220C: Características

Componente	Descrição
Número de juntas	5 Juntas + Garra
Construção	Braço Articulado Vertical
Quantidade de <i>Switch's</i>	5
Precisão (Posição)	$\pm 0.5mm$
Velocidade (movimento)	100mm/s max
Carga Máxima	1Kg
Atuador	Motor DC com encoder óptico
Range de Movimento	<ul style="list-style-type: none"> • Base: 310° • Ombro: $+130^\circ/-35^\circ$ • Cotovelo: $\pm 130^\circ$ • Garra (rotação): 360°
Abertura da Garra	55 mm
Dimensão	<ul style="list-style-type: none"> • Base: $220^\circ \times 180(H)mm$ • Braço: $220+220mm$
Peso	33Kg

Fonte: Elaborado pelo autor

Figura 7 – Cinemática Direta



Fonte: elaborado pelo autor

2.2.2 Denavit-Hartenberg

O algoritmo DH permite encontrar a posição relativa de um *link* do braço mecânico em relação ao *link* anterior. Na prática, podemos encontrar uma matriz de rotação e translação que descreve a posição relativa do efetuador final em relação a base.

Tabela 3 – Dimensões do braço

Elo	Base	Ombro	Cotovelo	Pulso
Apelido	L1	L2	L3	L4
Tamanho (mm)	385	220	220	155

Fonte: (IQBAL; ISLAM; KHAN, 2012)

Segundo (CABRAL, ca. 2010), a notação DH baseia-se no fato de que são necessários 4 parâmetros para definir a posição relativa de dois sistemas de coordenadas. Isso decorre do fato de um sistema de coordenadas ser definido por três retas e conhecendo dois dos eixos, o terceiro está definido pelas condições de ortogonalidade e pela regra da mão direita. A intersecção dos eixos de um sistema de coordenadas define a sua origem, portanto à partir da definição da posição relativa entre dois eixos de dois sistemas de coordenadas, pode-se descrever a posição relativa entre os dois sistemas de coordenadas.

O algoritmo DH trabalha com uma quádrupla $\{\alpha_{i-1}, a_{i-1}, d_{i-1}, \theta_{i-1}\}$, que representa ângulo de torção, tamanho do *link*, *link offset* e ângulos de juntas, respectivamente (ULLAH et al., 2014). Para qualquer manipulador robótico articulado, basta seguir os passos para encontrar os quatro parâmetros DH:

1. Eixo x_j deve interseccionar z_{j-1} ;
2. Eixo x_j deve ser perpendicular ao eixo z_{j-1} ;
3. θ : Rotacionar o eixo x_i em torno de z_i até ficar paralelo ao eixo x_{i+1} ;
4. d : Transladar o eixo de coordenadas j_i ao longo de z_i até a intersecção de z_i com x_{i+1} ;
5. a : Transladar o eixo de coordenadas j_{i+1} ao longo de x_{i+1} até a intersecção entre z_i e x_{i+1} ;
6. α : Rotacionar o eixo z_i em torno de x_{i+1} até ficar paralelo ao eixo z_{i+1} .

Após encontrar os parâmetros, deve-se aplicá-los à matriz de transformação homogênea, descrita na Equação 2.1, que nos retornará a posição relativa do *link* em relação ao *link* anterior.

$$A_{j-1}^j = Rot_z(\theta(j)) * Trans_z(d_j) * Trans_x(a_j) * Rot_x(\alpha_j) \quad (2.1)$$

A Tabela 4 lista os parâmetros DH para o robô ED-7220C.

Substituindo os parâmetros da Tabela 4 na equação de transformação homogênea 2.1 e realizando as operações matriciais, chega-se à matriz de rotação (3x3) e translação ($c1 * A, s1 * A, B$) da Equação 2.2.

$$T_{0,6} = \begin{bmatrix} c_1 * c_{2,3,4} * c_5 + s_1 * s_5 & -c_1 * c_{2,3,4} * s_5 + s_1 * c_5 & -c_1 * s_{2,3,4} & c_1 * A \\ s_1 * c_{2,3,4} * c_5 - c_1 * s_5 & -s_1 * c_{2,3,4} * s_5 - c_1 * c_5 & -s_1 * s_{2,3,4} & s_1 * A \\ -s_{2,3,4} * c_5 & s_{2,3,4} * s_5 & -c_{2,3,4} & B \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Tabela 4 – Parâmetros DH

		Juntas					
Símbolos	1	2	3	4	5	6	
α_{i-1}	0	-90°	0	0	-90°	0	
a_{i-1}	0	0	L2	L3	0	0	
d_i	L1	0	0	0	0	L4	
θ_i	θ_1	$\theta_2 - 90^\circ$	θ_3	θ_4	θ_5	0	

Fonte: (ULLAH et al., 2014)

Onde **A** e **B** são dados por:

$$A = -0.155 * s_{2,3,4} + 0.220 * c_{2,3} + 0.220 * c_2$$

$$B = 0.385 - 0.155 * c_{2,3,4} - 0.220 * s_{2,3} - 0.220 * s_2$$

e:

$$c_a = \cos(a), c_{a,b} = \cos(a + b), c_{a,b,c} = \cos(a + b + c)$$

$$s_a = \sin(a), s_{a,b} = \sin(a + b), s_{a,b,c} = \sin(a + b + c)$$

2.2.3 Cinemática Inversa

A cinemática direta é pouco útil, pois a base dos problemas envolvendo manipuladores robóticos articulados, é fazer com que a sua ferramenta de atuação, atinja uma série de pontos no espaço, seja para pegar um objeto em um ponto e deixá-lo em outro, ou soldando vários componentes eletrônicos, caso esse seja o propósito da ferramenta. A cinemática inversa, informa qual deve ser o ângulo relativo de cada *link* para que o efetuator alcance um ponto no espaço cartesiano.

A cinemática inversa ocupa em determinar as configurações de juntas para que o efetuator alcance uma posição no espaço e uma orientação, conforme visto na Figura 8.

Invertendo a matriz de transformação homogênea (Equação 2.2) através de intensos cálculos matemáticos, chega-se à solução analítica da cinemática inversa para o robô ED-7220C (ULLAH et al., 2014), que determina os corretos ângulos de juntas para *base*(θ_1), *ombro*(θ_2), *cotovelo*(θ_3) e *pulso*(θ_4) para coordenadas dentro do espaço de trabalho do equipamento. A orientação do manipulador (θ_5) é dada diretamente pela orientação do objeto a ser capturado.

Figura 8 – Cinemática Inversa



Fonte: Elaborado pelo autor

Equações de cinemática inversa do ED-7220C (ULLAH et al., 2014):

$$\theta_{base} = \arctan p_x, p_y \quad (2.3)$$

$$c_3 = \frac{(c_1 * p_x + s_1 * p_y + 0.220 * s_{2,3,4})^2 + (p_z - 0.385 + c_{2,3,4})^2 - 0.220^2 - 0.220^2}{2 * 0.220^2}$$

$$s_3 = \sqrt{1 - c_2^2}$$

$$\theta_{cotovelo} = \arctan(s_3, c_3) \quad (2.4)$$

$$c_2 = \frac{A - B}{2 * 0.220^2}$$

Onde **A** e **B** são dados por:

$$A = (c_1 * p_x + s_1 * p_y + 0.220 * s_{2,3,4}) * (c_3 * 0.220 + 0.220)$$

$$B = (p_z - 0.385 + 0.220 * c_{2,3,4}) * s_3 * 0.220$$

$$s_3 = \frac{A + B}{(c_3 * 0.220 + 0.220)^2 + s_3^2 + 0.220^2}$$

Onde **A** e **B** são dados por:

$$A = -(c_1 * p_x + s_1 * p_y + 0.220 * s_{2,3,4}) * s_3 * 0.220$$

$$B = (p_z - 0.385 + 0.220 * c_{2,3,4}) * (c_3 * 0.220 + 0.220)$$

$$\theta_{ombro} = \arctan(s_2, c_2) \quad (2.5)$$

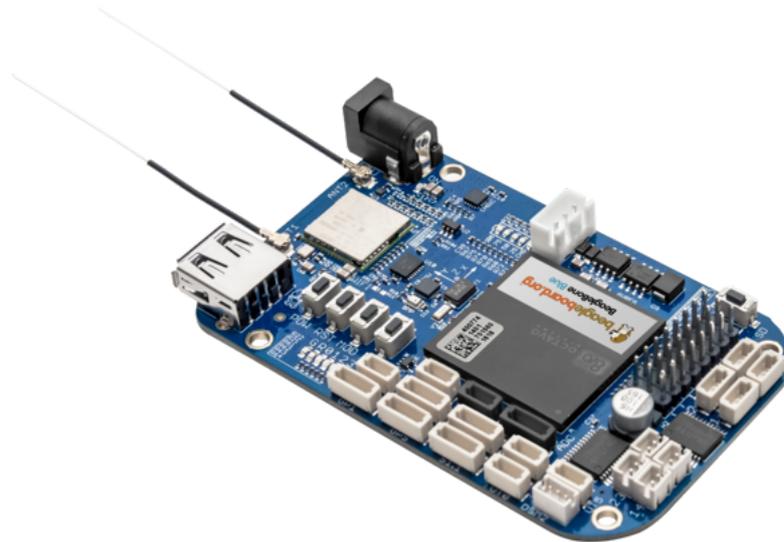
$$\theta_{pulso} = \theta_{2,3,4} - \theta_2 - \theta_3 \quad (2.6)$$

As Equações 2.3, 2.4, 2.5 e 2.6 nos dizem o ângulo de sua respectiva junta para que a garra do robô alcance um ponto no espaço. Essas equações nos fornecem a base teórica da cinemática do robô para construção do algoritmo de controle.

2.3 BeagleBone Blue

O BBBlue, apresentado na Figura 9, é um computador *open source all-in-one* baseado em Linux desenvolvido para trabalhar em projetos de robótica. Ele conta com um microprocessador Octavo OSD3358, conexões *wifi* e *bluetooth*, sensores *Inertial Measurement Unit/barometer*, reguladores de tensão para garantir segurança em sua utilização e entrada para 4 motores DC e seus *encoders*. É possível controlar o sentido de giro de cada motor e calcular o ângulo de rotação, além de 11 pinos que podem ser programados como *Input/Output*.

Figura 9 – BeagleBone Blue



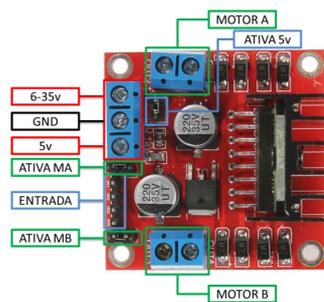
Fonte: (INSTRUMENTS, 2018)

2.4 AtMega328

O BBBlue possui 4 terminais (cada terminal é responsável por 1 motor e seu *encoder*) integrados na placa. Entretanto, são necessários 6 terminais para o controle do ED-7220C, um para cada grau de liberdade. Dada essa limitação do controlador, foi criado e integrado ao circuito um módulo composto pelo chip AtMega328 e uma ponte-H, necessários para o controle dos outros 2 motores.

O AtMega328 é um microcontrolador que combina 23 pinos *Input/Output* de propósito geral, interrupções internas e externas, serial programável USART e conversor Analógico/Digital de 6 canais de 10 bits (INCORPORATED, 2020). Integrando uma ponte-H, vista na Figura 10, é possível controlar o sentido de giro e velocidade de até 2 motores de corrente contínua.

Figura 10 – Ponte-H



Fonte: (FILIFEFLOP, 2013)

Apenas o pulso do ED-7220C possui dois motores, portanto, ao módulo adjunto (AtMega328+Ponte-H) foi delegada a responsabilidade de executar os movimentos de *pitch* e *roll*. O BBBlue, unidade central de controle, deve efetuar todos os cálculos e tratamento de dados e informar ao AtMega328 apenas o tipo de movimento a ser realizado pelo pulso e o ângulo de rotação dos motores.

2.5 Visão Computacional

O algoritmo de controle do manipulador robótico trabalha recebendo os pontos nos quais os objetos estão localizados, executa os cálculos e movimenta os motores para que o efetuador atinja todas as posições pré-determinadas. Portanto, é necessário que haja a identificação prévia das coordenadas onde cada objeto está localizado. Para aprimorar a autonomia do robô, uma câmera tira uma fotografia de todo espaço de trabalho do manipulador robótico e um algoritmo de detecção de objetos identifica as posições de cada objeto na imagem.

Segundo (PETERS, 2017), o principal objetivo da visão computacional é fornecer ao computador a capacidade de construir e interpretar cenas que para os seres humanos são naturais, ou seja, extrair informações úteis sobre as estruturas e padrões a partir de imagens capturadas por câmeras.

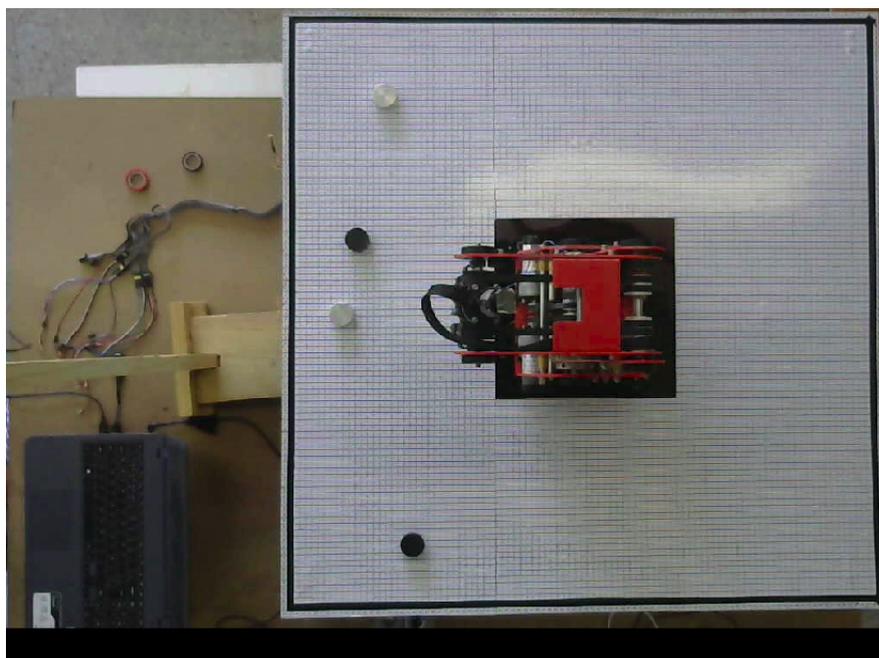
Na plataforma de controle desenvolvida, o módulo de detecção de objetos tem a responsabilidade de capturar a imagem (nos testes realizados foi utilizada uma câmera logitech de 720 *pixels*, semelhante à apresentada na Figura 11) em duas dimensões no espaço de trabalho do ED-7220C e realizar a detecção dos objetos e suas respectivas posições. A terceira dimensão (coordenada Z) foi fixada medindo-se a altura da plataforma de testes montada para o braço robótico. A Figura 12 é um exemplo do tipo de fotografia tirada pelo BBBlue no ambiente de trabalho do robô, imagem passará por tratamentos pelo módulo de visão computacional e o resultado são os pontos a serem atingidos pelo manipulador robótico.

Figura 11 – Câmera - Logitech 720p



Fonte: Elaborado pelo autor

Figura 12 – Imagem - Espaço de trabalho



Fonte: Elaborado pelo autor

2.6 Interface *WEB*

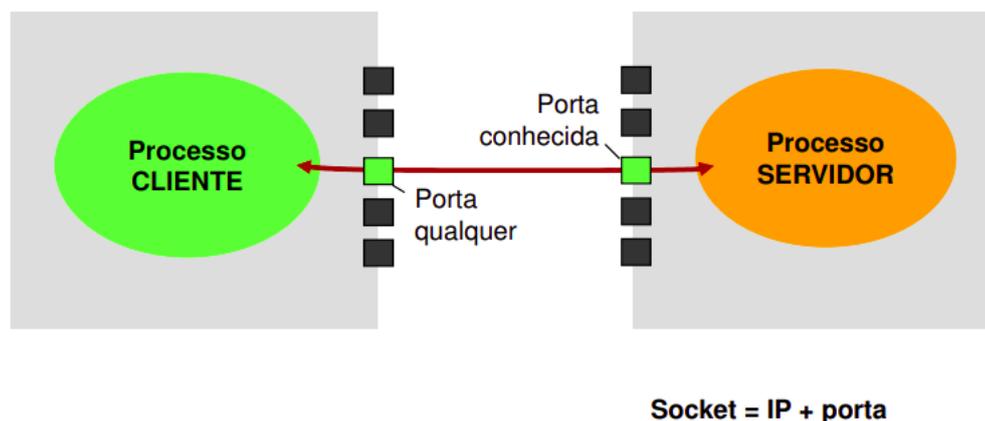
O início dos anos 2000 marcou a transição entre o conteúdo estático disponibilizado na *WEB* para um conteúdo dinâmico com o qual as pessoas podem interagir (ABRIATA et al., 2018) de forma ativa e dinâmica, marcando o *tipping point* para os sistemas desenvolvidos. Hoje, desenvolver um projeto de automação que não possui conexão com a internet, base de dados, *web services* e computação de alta performance, restringe a abrangência do sistema (STEIN, 2008), tornando-o defasado rapidamente por não ser capaz de acomodar a infinidade de aplicações que são desenvolvidas todos os dias por pesquisadores, estudantes e empresas ao redor do mundo. Dito isso, é imperativo que o sistema desenvolvido fosse capaz de se comunicar de forma remota através da internet, tornando-o escalável para novas funcionalidades.

Neste projeto foi desenvolvido um canal de comunicação com a internet utilizando *sockets*, que é capaz de receber comandos pré-determinados para realização de movimentos do ED-7220C. Um *Web Socket* é definido como uma tecnologia que habilita *hosts* remotos a realizarem comunicação *full-duplex* (LIU, 2012).

Os *sockets* *User Datagram Protocol* (UDP) e *Transmission Control Protocol* (TCP) são *interfaces* providas pelos respectivos protocolos da camada de transporte para comunicação interprocessos de diferentes *hosts* (TACLA, 2015). Neste projeto foi optado pelo protocolo TCP para comunicação entre o controlador (lado servidor) e o cliente (aplicação

que executa comandos no braço robótico), pela confiabilidade na entrega dos dados que este protocolo implementa. A figura 13 ilustra o funcionamento de uma comunicação via *socket*. O processo cliente deve comunicar-se com o sistema de controle, enviando comandos interpretáveis pelo controlador.

Figura 13 – Socket



Fonte: (TACLA, 2015)

2.7 Trabalhos relacionados

No artigo *Modeling and Computed Torque Control of a 6 Degree of Freedom Robotic Arm* (ULLAH et al., 2014), os autores apresentam os modelos de cinemática direta e inversa para o ED-7220C e uma técnica de controle não linear implementada no MATLAB/Simulink.

No trabalho *An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation* (IQBAL; ISLAM; KHAN, 2012), foi projetado uma nova plataforma de desenvolvimento para o ED-7220C centrada no controlador embarcado de 16-bit DSPIC33F, capaz de realizar conexões com dispositivos periféricos através de suas portas analógicas e digitais de *Input* e *Output*. No trabalho, uma câmera *on-board* permite o desenvolvimento de aplicações autônomas. Um sensor de força foi montado no efetuador final do robô para diferenciar objetos duros e macios.

Na pesquisa *Exploring the Training Potential of Recent Virtual Robotic Platforms: A Comprehensive Review* (AJWAD; ISLAM; IQBAL, 2017) são apresentadas plataformas virtuais para simulação de manipuladores robóticos articulados, que diminuem a necessidade de ter equipamentos laboratoriais caros e sofisticados. A virtualização do sistema também permite o treinamento de instrutores, pesquisadores, alunos, profissionais e hobistas. Nesse artigo são apresentados pontos relevantes sobre a simulação de sistemas robóticos

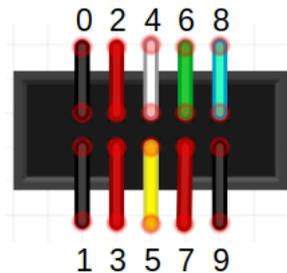
que podem vir a ser inspiração para futuros trabalhos com o BBBlue e o projeto aqui desenvolvido.

Neste trabalho optou-se pelo BBBlue como unidade principal de controle pelo suporte que o seu *hardware* fornece. O algoritmo de controle, desenvolvido em Python, foi criado não apenas para atender aos requisitos de funcionamento do ED-7220C, mas também para atender futuros incrementos. Dessa forma, foi criada uma plataforma em que qualquer tipo de tecnologia, pode ser inserida no sistema.

3 Desenvolvimento

Para construir o projeto do controlador com as equações de cinemática inversa 2.3, 2.4, 2.5 e 2.6 e a unidade de processamento, o BBBlue, foi necessário conhecer o funcionamento do articulador robótico a nível eletrônico. Para cada motor, há um cabo *flat* com 10 portas, utilizadas para sua alimentação e a do *encoder*, controle e sentido de rotação, leitura dos pulsos de *encoder*, que permite saber o sentido de giro atual e por fim o terminal do *micro switch*, que informa quando o *link* atinge o limite mecânico. Na Figura 14 é descrito todos os pinos de controle.

Figura 14 – Pinos ED-7220C - Vista frontal



Fonte: elaborado pelo autor

- 0: Alimentação do motor, terminal negativo;
- 1: Alimentação do motor, terminal negativo;
- 2: Alimentação do motor, terminal positivo;
- 3: Alimentação do motor, terminal positivo;
- 4: Terra do chassi;
- 5: *Micro switch*;
- 6: Canal B do *encoder*;
- 7: Alimentação do *encoder*, terminal positivo;
- 8: Canal A do *encoder*;
- 9: Alimentação do *encoder*, terminal negativo.

3.1 Robotics Cape Python

A linguagem de programação Python foi usada para projetar a lógica e realizar o controle do equipamento, por se tratar de uma linguagem moderna, ideal não apenas para automatização de processos repetitivos, mas também é amplamente utilizada em aplicações *WEB*, em ciência de dados, administração de sistemas e interfaces gráficas, entre tantas outras (FOUNDATION, s.d.).

O BBBlue por ser um *hardware* dedicado ao controle de robôs, conta com inúmeras bibliotecas que fornecem uma camada de abstração para o programador. Neste projeto, foi optado pela biblioteca *Robotics Cape Python* (RCPY), que vincula funcionalidades da biblioteca *Robot Control Library* (RCL). A RCL é uma biblioteca de baixo nível escrita em C, projetada para ser a interface para o *hardware* do BBBlue, contém extensas bibliotecas matemáticas para o controle de tempo discreto em malha fechada (OLIVEIRA, 2019).

3.2 ED7220-C

Na Tabela 2 é descrito o ângulo de movimento de cada *link* do robô, entretanto, estes são valores ideais, descritos pelo manual do equipamento que não se verificaram exatos na prática. Após extensos testes com o braço chegou-se aos ângulos reais de juntas apresentados na Tabela 5, que foram usados no algoritmo de controle. Para simplificação e melhor entendimento, os ângulos foram mapeados de zero grau, até o máximo atingido pela junta.

Tabela 5 – Limite mecânico

Articulação	Range de movimento
Base	0° - 310°
Ombro	0° - 50°
Cotovelo	0° - 160°
Pulso (<i>pitch</i>)	0° - 260°
Pulso (<i>roll</i>)	0° - 360°

Fonte: Elaborado pelo autor

Cada motor possui um *encoder* acoplado, para aferir os *ranges* reais de movimento de cada elo do braço robótico, foi realizado a contagem de pulsos de *encoder* para cada articulação lendo a amplitude máxima que um *link* pode alcançar. Para fins de simplificação, os pulsos de *encoder* foram mapeados de zero até a quantidade máxima medida. Na Tabela 6 estão descritos os resultados encontrados.

Tabela 6 – Pulsos de *encoder*

Articulação	<i>Range</i> de movimento
Base	14200
Ombro	2000
Cotovelo	6000
Pulso (<i>pitch</i>)	260
Pulso (<i>roll</i>)	400

Fonte: Elaborado pelo autor

Com as equações de cinemática inversa 2.3, 2.4, 2.5 e 2.6, e os dados das Tabelas 5 e 6, após extensos testes no laboratório, foi construído o algoritmo que posiciona o efetuator final do robô na posição desejada no espaço cartesiano. O código de posicionamento do braço robótico foi dividido entre o BBBlue (responsável pelo movimento da base, ombro, cotovelo e garra) e o Atmega328p, responsável pelos movimentos de *pitch* e *roll* do braço robótico, ambos os códigos estão disponíveis no repositório (RODRIGUES, 2022). O módulo de comunicação entre os dois controladores também foi desenvolvido e disponibilizado no mesmo repositório.

3.3 Interface WEB

Antes da construção do módulo de conexão com a internet, foi necessário conectar a placa BBBlue à internet. A BBBlue possui uma placa *Wifi* integrada, sendo necessário realizar a conexão à rede desejada:

Algoritmo 3.1 – Conexão Wifi (KRIDNER, 2020)

```

root@beaglebone:~# connmanctl
connmanctl> enable wifi
Enabled wifi
connmanctl> tether wifi disable
Disabled tethering for wifi
connmanctl> scan wifi
Scan completed for wifi
connmanctl> services
    wifi_ec1127bffa51_776865656c696573322e34_managed_psk
    wifi_ec1127bffa51_41545435333633_managed_psk
    wifi_ec1127bffa51_3257495245343037_managed_psk
    wifi_ec1127bffa51_41545438664848686669_managed_psk
connmanctl> agent on
Agent registered
connmanctl>
connmanctl> connect
    ↪ wifi_ec1127bffa51_776865656c696573322e34_managed_psk
Agent RequestInput
    ↪ wifi_ec1127bffa51_776865656c696573322e34_managed_psk
Passphrase = [ Type=psk, Requirement=mandatory, Alternates=[
    ↪ WPS ] ]
WPS = [ Type=wpspin, Requirement=alternate ]
Passphrase? your_password
connmanctl>
Connected
    ↪ wifi_ec1127bffa51_776865656c696573322e34_managed_psk
connmanctl> quit

```

Uma vez conectada à internet, desenvolveu-se o algoritmo que recebe o comando na porta 5020 e o executa. Todo comando recebido deve ter o seguinte formato:

$(x, y, z); \hat{\text{ângulo}} - \text{roll}; \text{garra}(\text{catch}, \text{release}, \text{nothing})$

O ângulo de rotação do pulso (*roll*) depende exclusivamente da orientação do objeto a ser capturado ou do local onde o objeto será depositado. A garra possui três comandos primordiais:

- *catch*: agarra o objeto;
- *release*: solta o objeto;
- *nothing*: não faz nada.

A quantidade de pulsos de *encoder* que a garra deve atingir depende do tamanho do objeto a ser capturado, esse valor deve ser informado pelo usuário para todo tipo de objeto a ser trabalhado, finalmente, o comando *q0* faz com que o robô retorne a posição de descanso. O algoritmo desenvolvido para conexão do sistema à internet se encontra disponível no repositório (RODRIGUES, 2022).

3.4 Detecção de objetos

Para que a câmera possa registrar com totalidade o espaço de trabalho do ED-7220C foi necessário a montagem de um suporte que a mantenha acima do robô, permitindo-a capturar uma imagem como a registrada na Figura 12.

O algoritmo de detecção de objetos usado no projeto foi desenvolvido pelo Laboratório IMobilis da UFOP (Laboratório Imobilis Computação Móvel - DECOM - UFOP, Atualizado em 2020). Este algoritmo realiza a captura automática da imagem através da câmera conectada ao BBBlue e identifica as posições *x* e *y* de cada peça pré-configurada (o eixo *z* é dado pelo tamanho da plataforma em volta do braço robótico).

Além da posição, é identificada a cor de cada peça, para isso deve ser informado ao algoritmo modelos de objetos que devem ser detectados e sua respectiva paleta de cor. Neste trabalho foram calibradas as cores preta e prata, observadas na Figura 12. Os objetos procurados pelo algoritmo devem ser informados previamente recortando-os da imagem capturada pela câmera (resolução e iluminação devem ser mantidas) e inserindo-o como entrada no programa. As figuras 17 e 18 mostram os objetos utilizados nos testes.

Figura 17 – Objeto prata



Fonte: elaborado pelo autor

Figura 18 – Objeto preto



Fonte: elaborado pelo autor

4 Resultados

A partir do modelo cinemático do ED-7220C foi desenvolvido uma plataforma de controle para o manipulador. Neste capítulo são apresentados alguns cenários de testes efetuados no robô. A cinemática inversa nos informa os ângulos de junta corretos para alcançar um ponto no espaço de trabalho do equipamento. Sendo assim, para qualquer ponto dentro do espaço operacional o sistema sempre posiciona cada motor em sua respectiva posição para alcançar a peça desejada.

4.1 Cenário de teste

Para operação do sistema, deve ser informado ao algoritmo a posição do objeto a ser capturado. Caso essa posição esteja dentro do espaço de trabalho do braço, o método que calcula a cinemática inversa é chamado, os ângulos de juntas são convertidos em pulsos de *encoder* com as informações das Tabelas 5 e 6, finalmente, o método responsável pelo movimento efetivo dos motores é chamado e cada motor é acionado até que a quantidade de pulsos de *encoder* necessária seja atingida.

Inicialmente devemos definir os ângulos de descanso do manipulador, esses ângulos, dependem de seus limitadores mecânicos e da posição de cada micro *switch*, que pode ser ajustada manualmente. Na Tabela 7 é descrita a posição inicial de cada junta utilizada nos testes. As posições iniciais são a referência do equipamento e armazenadas em constantes no código fonte.

Tabela 7 – Posições iniciais

Métrica	Base	Ombro	Cotovelo	<i>Pitch</i>	<i>Roll</i>
Ângulo	180°	-80°	75°	0°	0°

Fonte: Elaborado pelo autor

As sequências de Figuras 19, 20, 21, 22 e 23 ilustram um teste realizado no laboratório. Neste teste, o objeto a ser capturado foi posicionado no ponto 1 $(-0.2, -0.2, 0.22)$ com uma orientação de 90° , ele deve ser levado a uma posição intermediária 2 $(-0.25, -0.2, 0.3)$ e em seguida depositado em sua posição final 3 $(-0.22, 0.22, 0.22)$. A quantidade de pontos seguidos, dentro de seu espaço que o robô pode percorrer, é infinita. Recebido os pontos, a cinemática inversa calcula os ângulos de juntas com base na posição anterior do robô. Resultados vistos na Tabela 8.

Com base na posição a ser alcançada por cada junta, o algoritmo calcula o deslocamento de cada *link* partindo da posição anterior para que o efetuidor alcance o local

Tabela 8 – Resultados da cinemática inversa

Posição no espaço	Base	Ombro	Cotovelo	<i>Pitch</i>	<i>Roll</i>
Inicial	180°	-80°	75°	0°	0°
1	-135°	-49.97°	104.45°	-59.47°	90°
2	-141.34°	-57.15°	88.79°	-36.64°	0°
3	135°	-45.37°	94.81°	-54.44°	0°

Fonte: Elaborado pelo autor

desejado. As Tabelas 9, 10 e 11 descrevem qual movimento deve ser realizado por cada motor. Esses números são obtidos através de equações fechadas e sempre se repetem para a realização do movimento entre os mesmos pontos. Após deixar o objeto no ponto 3 (ação ilustrada na Figura 22) o equipamento retorna a posição de descanso. Para retornar a posição de descanso não é necessário realizar os cálculos de cinemática inversa, uma vez que essa posição é marcada pelos micros *switches*.

Tabela 9 – Posição inicial para o ponto 1

Junta	Posição de partida	Posição final	Deslocamento angular	Pulsos de encoder
base	180°	225°	45°	2061.29
ombro	-80°	-49.97°	30.03°	1201.16
cotovelo	75°	104.45°	59.47°	2230.29
<i>pitch</i>	0°	-59.47°	0°	0
<i>roll</i>	0°	90°	90°	100

Fonte: Elaborado pelo autor

Tabela 10 – Ponto 1 para ponto 2

Junta	Posição de partida	Posição final	Deslocamento angular	Pulsos de encoder
base	225°	218.66°	-6.34°	-290.42
ombro	-49.97°	-57.15°	-7.18°	-287.05
cotovelo	104.45°	88.79°	-22.84°	-856.32
<i>pitch</i>	-59.47°	-36.64°	-59.47°	-59.47
<i>roll</i>	90°	90°	0°	0

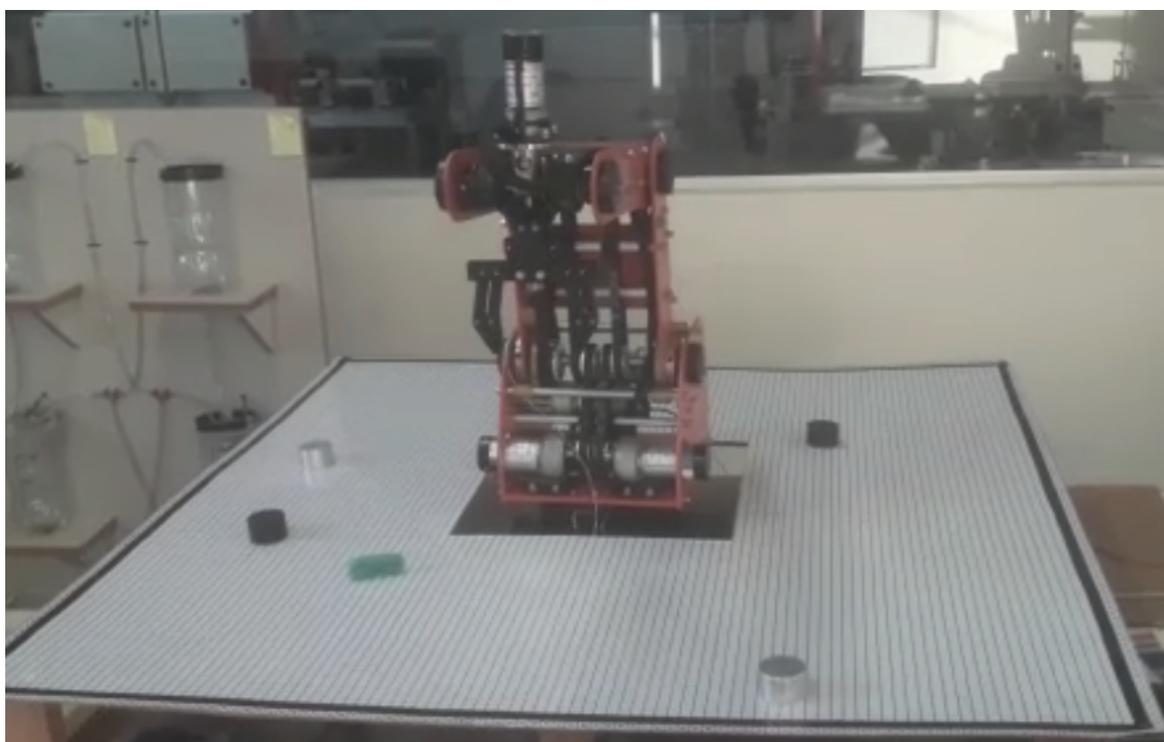
Fonte: Elaborado pelo autor

Tabela 11 – Ponto 2 para ponto 3

Junta	Posição de partida	Posição final	Deslocamento angular	Pulsos de encoder
base	218.66°	135°	-83.66°	-3832.16
ombro	-57.15°	-45.37°	11.78°	471.27
cotovelo	88.79°	94.81°	17.80°	667.58
<i>pitch</i>	-36.64°	-54.44°	-36.64°	-36.64
<i>roll</i>	90°	90°	0°	0

Fonte: Elaborado pelo autor

Figura 19 – Posição de descanso

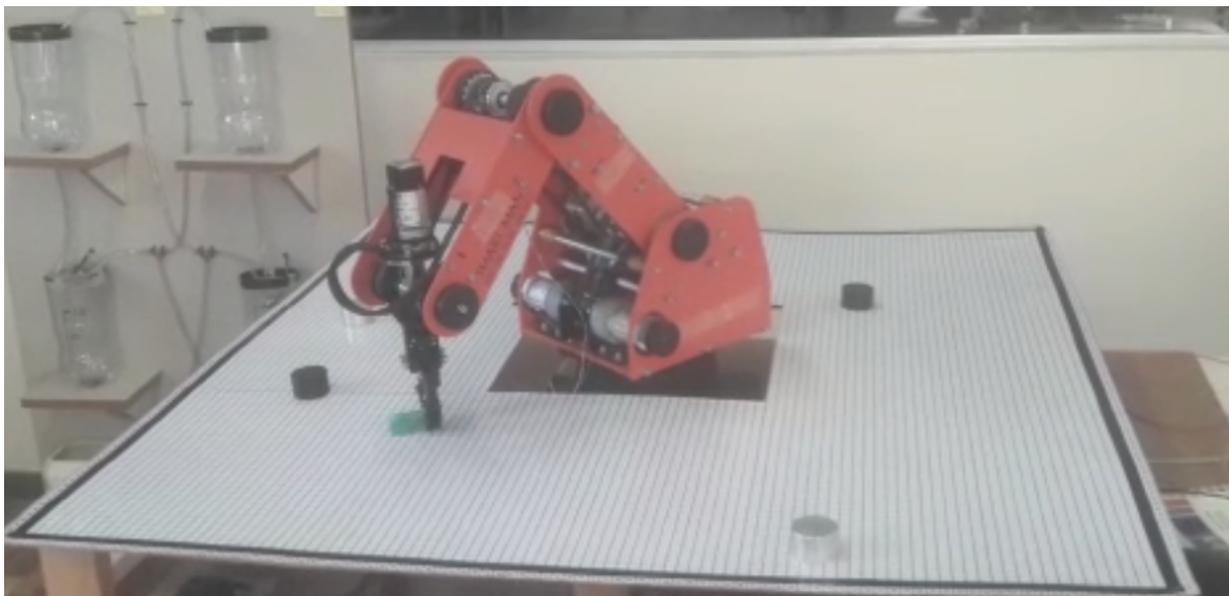


Fonte: elaborado pelo autor

4.2 Detectando objetos

Usando os resultados do algoritmo de detecção de objetos ([Laboratório Imobilis Computação Móvel - DECOM - UFOP, Atualizado em 2020](#)) como entrada para o módulo de cinemática inversa foi possível automatizar o processo de identificação dos pontos no espaço de trabalho do robô. Na Figura 24 podemos visualizar os pontos $(-0.4, 0.13, 0.22)$, $(-0.3, -0.4, 0.22)$ e $(-0.42, 0, 0.22)$ identificados pelo módulo de visão computacional. Nota-se que nesta simulação, uma peça da cor prata não foi detectada pelo algoritmo em decorrências das condições de iluminação do laboratório no momento do teste. Quando as condições de iluminação das peças divergem daquelas usadas para calibrar o algoritmo de

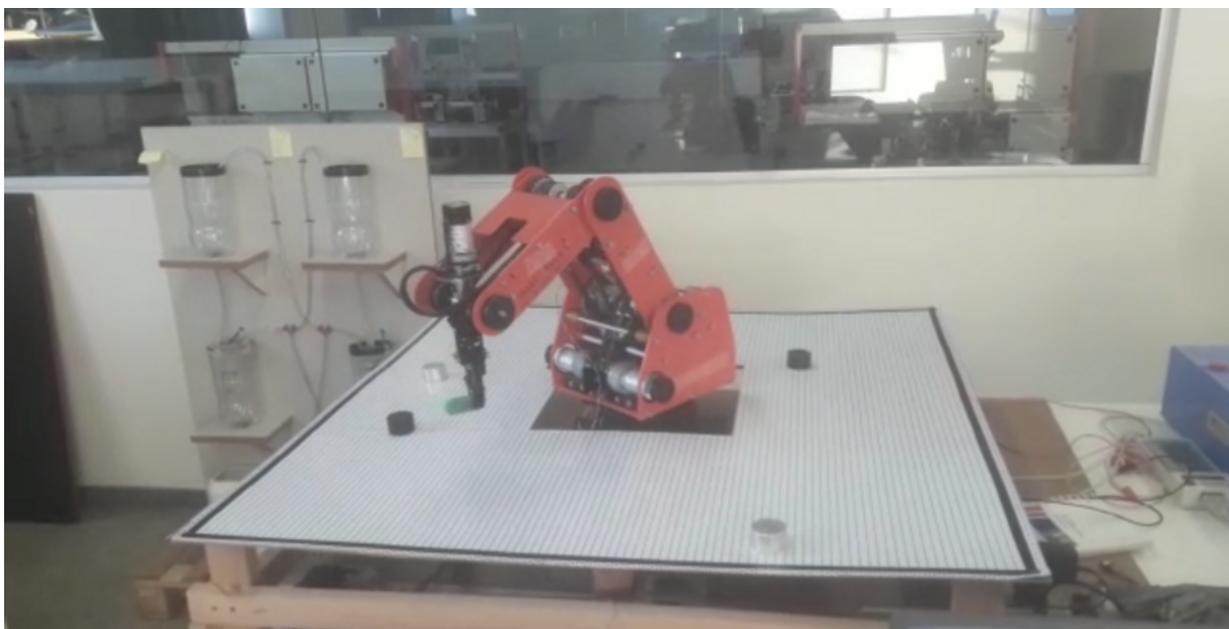
Figura 20 – Capturando o objeto



Fonte: elaborado pelo autor

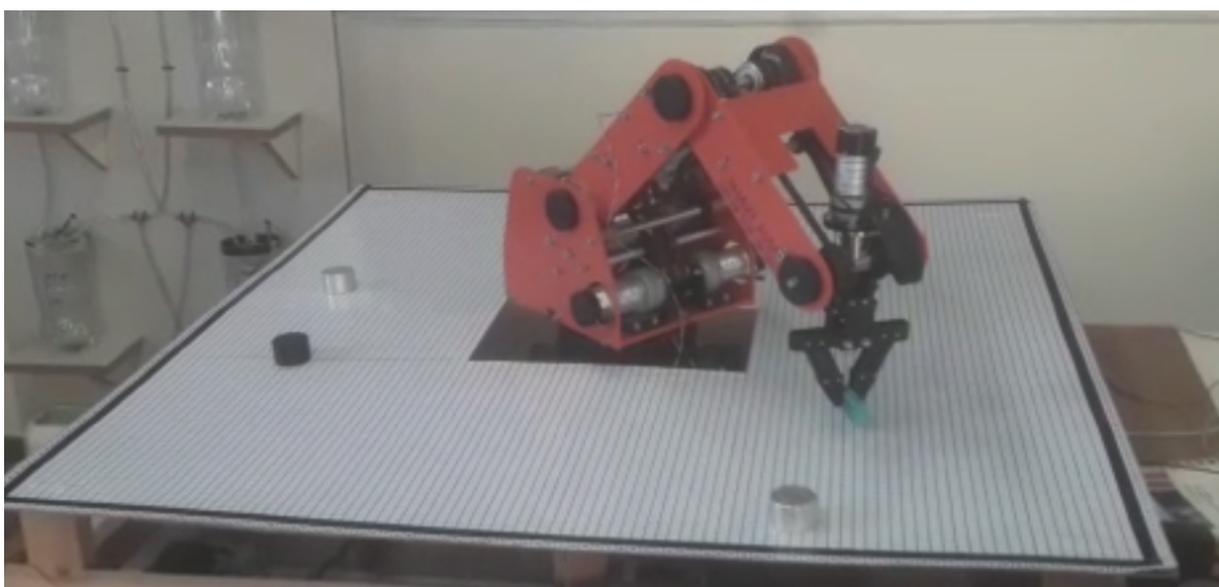
detecção, vistas nas Figuras 17 e 18, uma nova calibração deve ser feita trocando as figuras usadas como entrada no algoritmo para figuras com condições próximas da iluminação atual.

Figura 21 – Posição intermediária



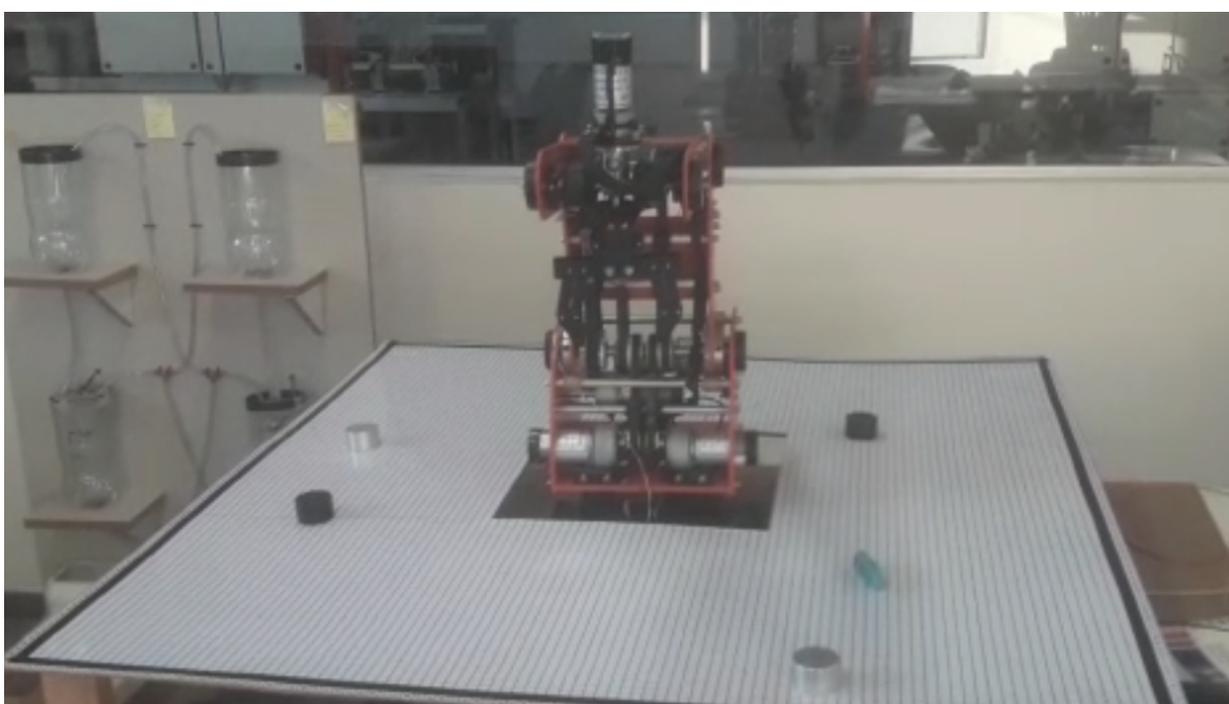
Fonte: elaborado pelo autor

Figura 22 – Movendo o objeto para posição definida



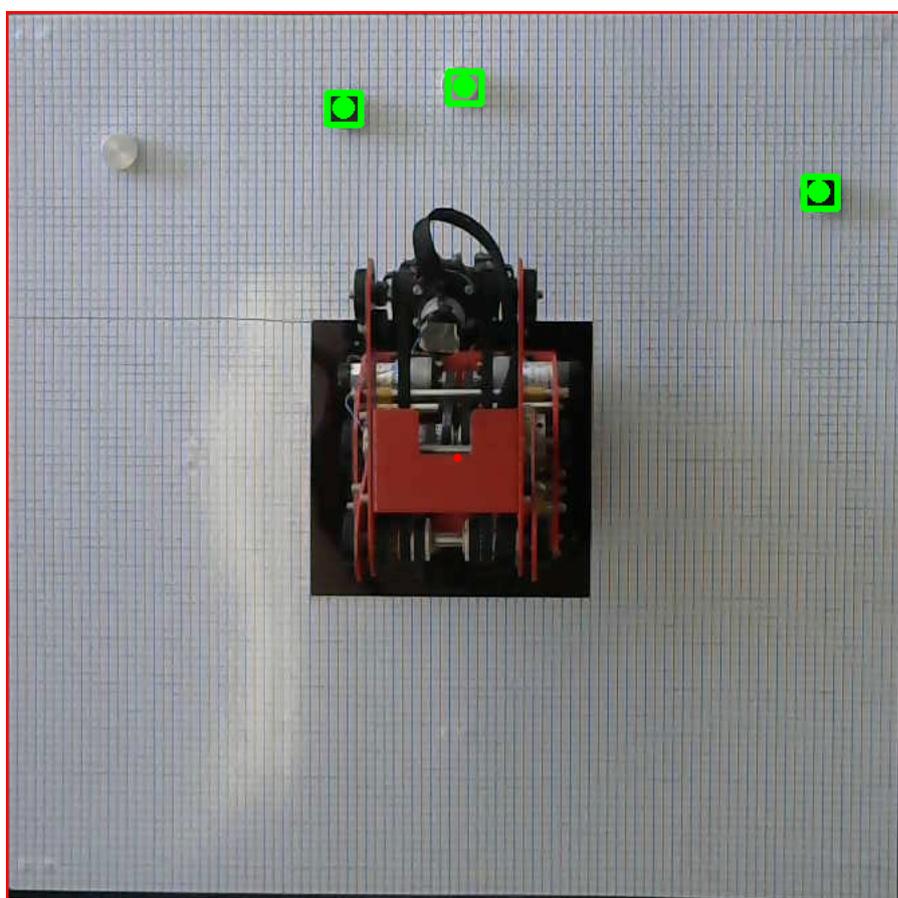
Fonte: elaborado pelo autor

Figura 23 – Retorno a posição de descanso



Fonte: elaborado pelo autor

Figura 24 – Detecção automática dos objetos



Fonte: Elaborado pelo autor

5 Conclusão

No desenvolvimento do presente trabalho, foi projetado um sistema de controle efetivo para o manipulador ED-7220C, amplamente utilizado no aprendizado de robótica em instituições de ensino. A construção de todo algoritmo de controle foi baseado na teoria cinemática de manipuladores robóticos articulados, ao mesmo tempo em que foram adicionados novos recursos tecnológicos, como conexão com a Internet e detecção automática de objetos a partir de imagens.

O controlador projetado é capaz de substituir o controlador original ED-MK4 por não apenas executar as mesmas funcionalidades que esse, mas também expandir as possibilidades de desenvolvimento do robô para o âmbito da IoT.

No projeto foi realizado o trabalho de *retrofitting* sob o braço robótico, fornecendo uma plataforma moderna e expansível para o equipamento cuja estrutura de controle padrão está defasada no atual cenário tecnológico, poupando recursos na aquisição de um novo braço robótica pela falta de possibilidades de seu *hardware* de controle.

Para trabalhos futuros, um próximo passo poderia ser a criação de interfaces gráficas de alto nível para os usuários, possibilitando toda a interação com o robô através de um ambiente amigável no qual uma pessoa sem *expertise* em computação ou área correlata poderia operar o sistema. Uma segunda orientação para projetos, é a inserção de novos sensores no BBBlue para aprimorar o funcionamento do robô, por exemplo, inclusão de sensores de distância e algoritmos de detecção de colisões, para evitar possíveis obstáculos.

Referências

- ABRIATA, L. A. et al. Augmenting research, education, and outreach with client-side web programming. *Trends in biotechnology*, Elsevier, v. 36, n. 5, p. 473–476, 2018. Citado na página 27.
- AJWAD, S.; ISLAM, M. U.; IQBAL, J. Exploring the training potential of recent virtual robotic platforms: A comprehensive review. *Journal of Advancements in Robotics*, v. 4, p. 9–25, 08 2017. Citado na página 28.
- ALVARES, A. et al. Retrofitting of the irb6-s2 robotic manipulator using computer numerical control- based controllers. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. 40, 03 2018. Citado na página 13.
- AMINE, B. M. E.; NADJAH, K. A. *Design Of An Active Disturbance Rejection Control For The Ed-7220c Robot Arm*. Dissertação (Mestrado) — Université M’hamed Bougara - Bouverdes, 2017. Citado 2 vezes nas páginas 16 e 17.
- ARDUINO. *ATmega168/328P-Arduino Pin Mapping*. [S.l.], s.d. Disponível em: <<https://www.arduino.cc/en/Hacking/PinMapping168>>. Acesso em: 10 Jul. 2021. Citado na página 31.
- CABRAL, E. L. L. *Cinemática direta de robôs manipuladores*. [S.l.], ca. 2010. Disponível em: <<http://sites.poli.usp.br/p/eduardo.cabral/CinematicaDireta.pdf>>. Acesso em: 01 jun. 2021. Citado 2 vezes nas páginas 19 e 21.
- ED-LABORATORY. *ED-7220 ARM Robot Trainer Specification*. [S.l.], s.d. Citado 2 vezes nas páginas 14 e 16.
- ELETRONICA, B. da. *Micro switch*. [S.l.], 2014. Disponível em: <<https://www.baudaeletronica.com.br/chave-micro-switch-kw-11-3z-5a-3-terminais-1.html>>. Acesso em: 02 jun. 2021. Citado na página 19.
- FILIPEFLOP. *Motor dc com driver ponte h l298n*. [S.l.], 2013. Disponível em: <<https://www.filipeflop.com/blog/motor-dc-arduino-ponte-h-l298n/>>. Acesso em: 10 Jul. 2021. Citado na página 25.
- FOUNDATION, P. S. *Python*. [S.l.], s.d. Disponível em: <<https://www.python.org/>>. Acesso em: 10 jul. 2021. Citado na página 32.
- INCORPORATED, M. T. *ATmega - megaAVR® Data Sheet*. [S.l.], 2020. Disponível em: <<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf>>. Acesso em: 10 Jul. 2021. Citado na página 25.
- INSTRUMENTS, T. *Beagle bone blue*. [S.l.], 2018. Disponível em: <<https://beagleboard.org/bluef>>. Acesso em: 28 fev. 2021. Citado 2 vezes nas páginas 13 e 24.
- IQBAL, J.; ISLAM, M. U.; KHAN, H. Modeling and analysis of a 6 dof robotic arm manipulator. *Canadian Journal on Electrical and Electronics Engineering*, v. 3, p. 300–306, 01 2012. Citado 3 vezes nas páginas 18, 20 e 28.

KRIDNER, J. *Networking WiFi*. [S.l.], 2020. Citado na página 34.

Laboratório Imobilis Computação Móvel - DECOM - UFOP. *Algoritmo de captura de imagem - OpenCV*. Atualizado em 2020. Citado 2 vezes nas páginas 35 e 39.

LIU, X. S. Q. Research of web real-time communication based on web socket. *International Journal of Communications, Network and System Sciences*, v. 5, n. 12, p. 5, 2012. Citado na página 27.

MANZOOR, S. et al. An open-source multi-dof articulated robotic educational platform for autonomous object manipulation. *Robotics and Computer-Integrated Manufacturing*, v. 30, p. 351–362, 06 2014. Citado na página 18.

MATHWORKS. *A Communicating with EEPROM Using BeagleBone Blue Hardware*. [S.l.], 2021. Disponível em: <<https://www.mathworks.com/help/supportpkg/beagleboneblue/ref/communicating-with-EEPROM-using-beaglebone-blue-hardware.html>>. Acesso em: 30 Set. 2021. Citado na página 31.

OLIVEIRA, M. de. *Robot Control Library*. [S.l.], 2019. Disponível em: <<https://github.com/mcdeoliveira/rcpy>>. Acesso em: 10 jul. 2021. Citado na página 32.

PETERS, J. *Foundations of Computer Vision*. [S.l.: s.n.], 2017. v. 124. ISBN 978-3-319-30260-7, ISBN 978-3-319-30262-1 (eBook). Citado na página 26.

RODRIGUES, E. *Controle de um Braço Robótico Articulado com 6 Graus de Liberdade*. [S.l.]: GitHub, 2022. <<https://github.com/eduardoComp/tcc.git>>. Citado 2 vezes nas páginas 33 e 35.

SENA, H. *Controle robótico referenciado por sistema de visão computacional utilizando o Kinect*. 2011. Monografia (Engenharia de Controle e Automação), FAG (Faculdade Assis Gurgacz), Cascavel, Brazil. Citado na página 18.

STEIN, L. D. Towards a cyberinfrastructure for the biological sciences: progress, visions and challenges. *Nature Reviews Genetics*, v. 9, n. 678–688, 2008. Citado na página 27.

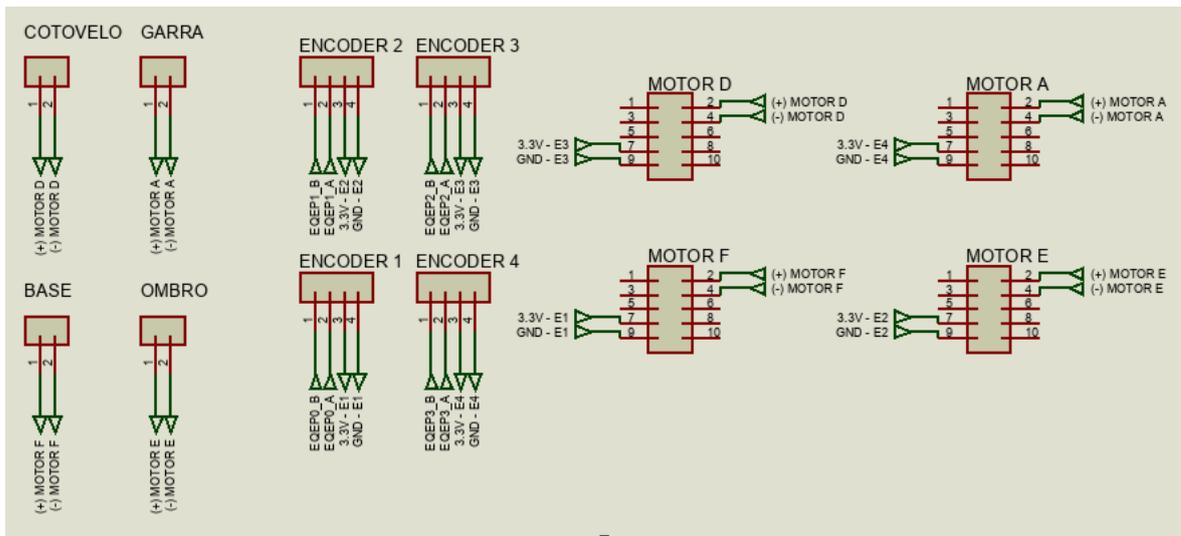
TACLA, C. A. *SOCKETS UDP, TCP E MULTICAST*. 2015. Acesso em: 20 Jun. 2021. Disponível em: <<https://www.inf.ufsc.br/~bosco.sobral/ensino/ine5645/2014-1/0050-Sockets.pdf>>. Citado 2 vezes nas páginas 27 e 28.

ULLAH, M. et al. Modeling and computed torque control of a 6 degree of freedom robotic arm. In: . [S.l.: s.n.], 2014. p. 133–138. ISBN 978-1-4799-5132-1. Citado 4 vezes nas páginas 21, 22, 23 e 28.

Apêndices

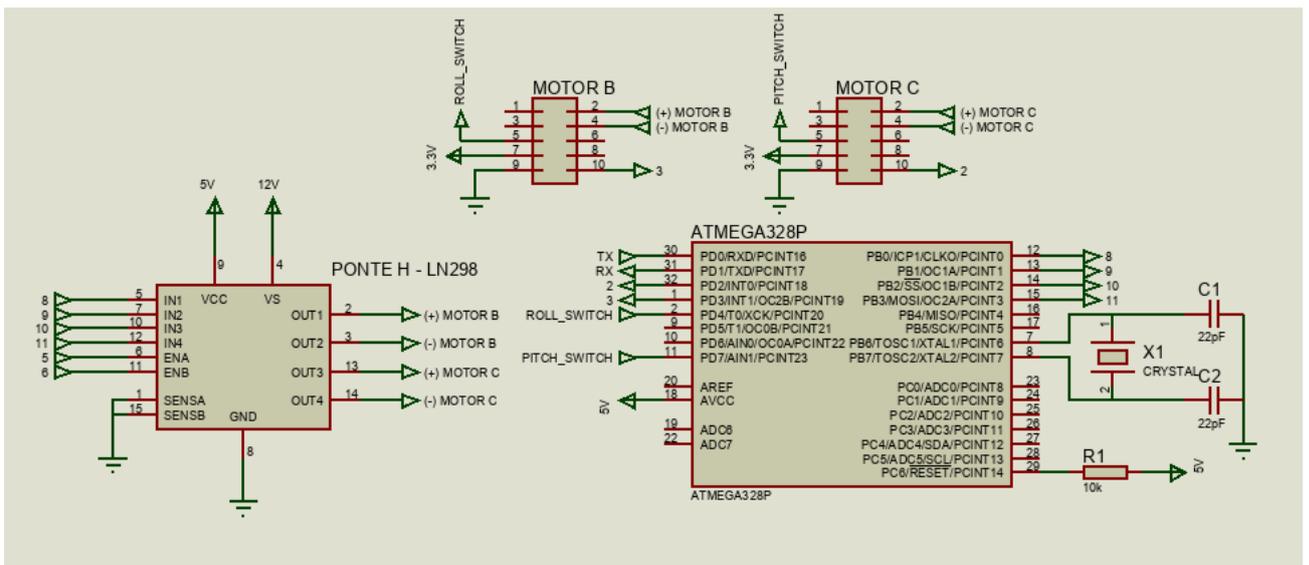
APÊNDICE A – Circuito eletrônico

Figura 25 – Circuito: BeagleBone Blue



Fonte: elaborado pelo autor

Figura 26 – Circuito: AtMega328p



Fonte: elaborado pelo autor