

ANO
2017



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ANDRÉ GUILHERME GEMBAROWSKI | DESENVOLVIMENTO
DE UM CONTROLADOR ABERTO PARA UM ROBÔ DIDÁTICO

DISSERTAÇÃO DE MESTRADO

**DESENVOLVIMENTO DE UM
CONTROLADOR ABERTO PARA UM
ROBÔ DIDÁTICO**

ANDRÉ GUILHERME GEMBAROWSKI

JOINVILLE, 2017

Este trabalho descreve o desenvolvimento e a análise do desempenho de um controlador aberto para um sistema robótico didático. A estrutura do controlador proposto é baseada em um sistema microcontrolado em conjunto com uma interface de operação remota. Por estar programado em um código aberto, é possível realizar consultas em qualquer parte do controlador e conhecer o algoritmo do projeto, bem como propor melhorias e outras estratégias de controle.

Orientador: Aníbal Alexandre Campos Bonilla

Joinville, 2017

ANDRÉ GUILHERME GEMBAROWSKI

**DESENVOLVIMENTO DE UM CONTROLADOR ABERTO
PARA UM ROBÔ DIDÁTICO**

Dissertação apresentada ao Curso de Mestrado Profissional em Engenharia Elétrica, no Centro de Ciências Tecnológicas, da Universidade do Estado de Santa Catarina, como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Aníbal Alexandre Campos Bonilla

**JOINVILLE – SC
2017**

Ficha catalográfica elaborada pelo(a) autor(a), com
auxílio do programa de geração automática da
Biblioteca Setorial do CCT/UEDESC

Gembarowski, André Guilherme

Desenvolvimento de um controlador aberto para um
robô didático / André Guilherme Gembarowski. -
Joinville , 2017.

121 p.

Orientador: Aníbal Alexandre Campos Bonilla
Dissertação (Mestrado) - Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas,
Programa de Pós-Graduação Profissional em Engenharia
Elétrica, Joinville, 2017.

1. Controlador aberto. 2. Braço robótico. 3.
Ensino de robótica. 4. Sistemas microcontrolados. I.
Bonilla, Aníbal Alexandre Campos. II. Universidade
do Estado de Santa Catarina. Programa de Pós-
Graduação. III. Título.

Desenvolvimento de Um Controlador Aberto para um Robô Didático

por

André Guilherme Gembarowski

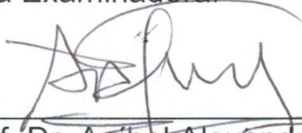
Esta dissertação foi julgada adequada para obtenção do título de

MESTRE PROFISSIONAL EM ENGENHARIA ELÉTRICA

Área de concentração em "Automação de Sistemas"
e aprovada em sua forma final pelo

CURSO DE MESTRADO PROFISSIONAL EM ENGENHARIA ELÉTRICA
CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.

Banca Examinadora:



Prof. Dr. Anibal Alexandre Campos
Bonilla
CCT/UDESC (Orientador/Presidente)



Prof. Dr. Rodrigo Trentini Preuss -
Centro Universitário - Católica SC



Profa. Dra. Mariana Santos Matos
Cavalca
CCT/UDESC

Joinville, SC, 28 de agosto de 2017.

Dedico este trabalho à minha filha Alice.

AGRADECIMENTOS

Agradeço à Universidade do Estado de Santa Catarina pela oportunidade de desenvolvimento acadêmico, bem como aos professores e demais colegas que contribuíram com a minha formação e realização deste trabalho.

Ao Serviço Nacional de Aprendizagem Industrial de Joinville, por permitir e incentivar o desenvolvimento deste trabalho.

Ao meu orientador, Professor Alexandre Campos, por me apoiar e guiar minhas pesquisas na elaboração do projeto. Sua paciência em nossas conversas foi muito importante para mim. Agradeço pelos esclarecimentos ao longo desse período de estudos.

À minha mãe Ivete e ao meu pai Almir que sempre batalharam pela minha educação. Todo esforço é minha inspiração. Muito obrigado por todos os ensinamentos e valores dados ao longo da vida.

À minha sogra Salete e ao meu sogro Idelson por todo apoio, carinho e compreensão no desenvolvimento deste trabalho.

Gostaria de agradecer também à minha irmã Flávia e ao meu irmão Leonardo que de alguma forma sempre me ajudaram e trouxeram ânimo para mim.

Agradeço especialmente à minha esposa Sacha, cuja participação foi fundamental para a realização do projeto, sempre me incentivando e auxiliando em todos os momentos.

“Tente mover o mundo – o primeiro passo
será mover a si mesmo.”

Platão

RESUMO

Este trabalho descreve o desenvolvimento e a análise do desempenho de um controlador aberto para um sistema robótico didático composto pelo robô Scorbot-ER 4pc que possui seis graus de liberdade com um efetuator do tipo garra. Assim, o controlador original foi desprezado para que o projeto de *hardware* e programação fosse descrito minuciosamente a fim de explorar o objetivo educacional do projeto, abordando teorias como a cinemática direta, inversa e diferencial. A estrutura do controlador proposto é baseada em um sistema microcontrolado que define o acionamento dos motores do robô a partir dos sinais dos *encoders*, chaves fim de curso e outros comandos enviados pelo usuário. O método de controle utilizado é o PID que tem como referência os pontos de cada etapa da trajetória realizada pelo robô. Por estar programado em um código aberto, é possível realizar consultas em qualquer parte do controlador e conhecer o algoritmo do projeto, bem como propor melhorias e outras estratégias de controle.

Palavras-chave: Controlador aberto. Braço robótico. Ensino de robótica. Sistemas microcontrolados.

ABSTRACT

This work describes the development and performance analysis of an open controller for a didactic robotic system composed by a Scorbot-ER 4pc robot with six degrees of freedom and a gripper type end-effector. Thus, the original controller is scorned so that the hardware and programming design is thoroughly described to explore the educational objective of this project, addressing theories such as forward, inverse and differential kinematics. The proposed controller structure is based on a micro-controlled system that defines the robot by motor activation from encoder signals, micro-switches, and other commands sent by user. The used control method is a PID that has the points of each trajectory step performed by the robot's as reference. As it is programmed in open source, it is possible to perform queries anywhere in the controller and to know or edit the algorithm of the project, as well as propose improvements and other control strategies.

Key-words: Open controller. Robotic arm. Robotics teaching. Micro-controlling systems.

LISTA DE ILUSTRAÇÕES

Figura 1 - Visão geral do Scorbot-ER 4pc.....	34
Figura 2 - Juntas do Scorbot-ER4pc	35
Figura 3 - Espaço de trabalho em vista lateral	36
Figura 4 - Espaço de trabalho em vista superior	36
Figura 5 - Componentes do Scorbot-ER 4 pc.....	37
Figura 6 - Detalhe do motor da junta.....	38
Figura 7 - Detalhe do <i>encoder</i>	39
Figura 8 - Sinais gerados pelos fototransistores	39
Figura 9 - Detalhe da chave fim de curso.....	40
Figura 10 - Detalhe da garra	41
Figura 11 - Esquema do conector DB-50	41
Figura 12 – Sistemas de coordenadas de corpos	44
Figura 13 - Parâmetros de <i>Denavit-Hartenberg</i>	49
Figura 14 - Sistemas de coordenadas do Scorbot-ER 4pc	50
Figura 15 – Representação dos ângulos RPY	54
Figura 16 – Braço robótico simplificado	57
Figura 17 – Detalhe do braço robótico simplificado.....	58
Figura 18 – Placa Arduino Mega 2560 Rev3.....	65
Figura 19 – Arduino Mega Protoshield	69
Figura 20 – Divisor de tensão para a chave fim de curso	69
Figura 21 – Divisor de tensão para o fototransistor.....	70
Figura 22 – Circuito integrado L298N.....	71
Figura 23 – Esquema do circuito L298N	71
Figura 24 – Módulo ponte H L298N	72
Figura 25 – Módulo <i>wireless</i> ESP8266 ESP-01	74
Figura 26 – Circuito integrado 4050	74
Figura 27 – Funcionamento da plataforma <i>Blynk</i>	75
Figura 28 – Código QR para interface de operação	76
Figura 29 – Aba inicial da interface de operação	76
Figura 30 – Aba de controle das juntas da interface	77
Figura 31 – Aba de controle do efetuator da interface	78

Figura 32 – Aba de controle PID da interface	78
Figura 33 – Aba de supervisão da interface	79
Figura 34 – Componentes do controlador.....	81
Figura 35 – Braço robótico pronto para operação	81
Figura 36 – Alimentação elétrica do controlador.....	82
Figura 37 – Sinalização do controlador	82
Figura 38 – Comparação entre os controladores.....	83
Figura 39 – Funcionamento do projeto	84
Figura 40 – Esquema do circuito ponte H.....	85
Figura 41 – Elemento auxiliar para o referenciamento	88
Figura 42 – Posição de referência do robô	89
Figura 43 – Esquema de controle PID	93
Figura 44 – Esforços do motor em relação ao sinal PWM	94
Figura 45 – Fluxograma de funcionamento dos motores.....	97
Figura 46 – Postura inicial	100
Figura 47 – Postura final.....	100
Figura 48 – Postura inicial simulada	101
Figura 49 – Postura final simulada	101
Figura 50 – Deslocamento angular no posicionamento.....	101
Figura 51 – Deslocamento angular da base	102
Figura 52 – Ação de controle na base com controle PID.....	102
Figura 61 – Deslocamento angular do ombro.....	103
Figura 62 – Ação de controle no ombro com controle PID	103
Figura 63 – Deslocamento angular do cotovelo.....	104
Figura 64 – Ação de controle com controle PID	104
Figura 57 – Perfil de suavização do movimento	105
Figura 58 – Deslocamento angular de arfagem.....	106
Figura 59 – Deslocamento angular de rolagem	106
Figura 60 – Ação de controle no motor 4.....	107
Figura 61 – Ação de controle no motor 5.....	107
Figura 62 – Abertura da garra.....	108
Figura 63 – Fechamento da garra de 100% para 75%	108
Figura 64 – Ação de controle no fechamento da garra	108
Figura 65 – Abertura da garra de 25% para 100%	109

Figura 66 – Ação de controle na abertura da garra	109
Figura 67 – Primeiro caso de aplicação de carga	110
Figura 68 – Variação angular do ombro no primeiro caso.....	110
Figura 69 – Ação de controle do ombro no primeiro caso	111
Figura 70 – Variação angular do cotovelo no primeiro caso	111
Figura 71 – Ação de controle do cotovelo no primeiro caso.....	111
Figura 72 – Variação angular da arfagem no primeiro caso.....	112
Figura 73 – Ação de controle da arfagem no primeiro caso	112
Figura 74 – Segundo caso de aplicação de carga	113
Figura 75 – Variação angular do ombro no segundo caso.....	113
Figura 76 – Ação de controle do ombro no segundo caso	113
Figura 77 – Variação angular do cotovelo no segundo caso.....	114
Figura 78 – Ação de controle do cotovelo no segundo caso.....	114
Figura 79 – Variação angular da arfagem no segundo caso.....	115
Figura 80 – Ação de controle da arfagem no segundo caso	115

LISTA DE TABELAS

Tabela 1 - Relação de trabalhos com robôs Scorbot	31
Tabela 2 - Especificações técnicas do Scorbot-ER 4pc.....	35
Tabela 3 - Relação de motores por junta	37
Tabela 4 - Cabeamento do Scorbot-ER 4pc.....	42
Tabela 5 – Parâmetros de <i>Denavit-Hartenberg</i>	51
Tabela 6 – Funções dos pinos da placa Arduino Mega 2560.....	67
Tabela 7 – Dados comparativos entre os controladores	83
Tabela 8 – Estados de funcionamento dos motores	86
Tabela 9 – Ações dos <i>encoders</i>	87
Tabela 10 – Relação dos <i>encoders</i>	96
Tabela 11 – Sintonia de parâmetros por <i>Ziegler-Nichols</i>	98
Tabela 12 – Resultados experimentais	99

LISTA DE ABREVIATURAS E SIGLAS

CC	Corrente Contínua
IDE	<i>Integrated Development Environment</i> (Ambiente de Desenvolvimento Integrado)
IP	<i>Internet Protocol</i> (Protocolo de Internet)
ISO	<i>International Organization for Standardization</i> (Organização Internacional de Normalização)
LED	<i>Light Emitting Diode</i> (Diodo Emissor de Luz)
TCP	<i>Transmission Control Protocol</i> (Protocolo de Controle de Transmissão)
PWM	<i>Pulse Width Modulation</i> (Modulação por Largura de Pulso)
UART	<i>Universal Asynchronous Receiver/Transmitter</i> (Transmissor/ Receptor Assíncrono Universal)
USB	<i>Universal Serial Bus</i> (Barramento Serial Universal)
CI	Circuito Integrado
GPIO	<i>General Purpose Input/Output</i> (Entrada e Saída de Propósito Geral)
QR	<i>Quick Response</i> (Resposta Rápida)
RGB	<i>Red-Green-Blue</i> (Vermelho-Verde-Azul)

LISTA DE SÍMBOLOS

\hat{x}_i	Eixo x do sistema de coordenadas “ i ”
\hat{y}_i	Eixo y do sistema de coordenadas “ i ”
\hat{z}_i	Eixo z do sistema de coordenadas “ i ”
${}^a p_b$	Componente escalar do sistema de coordenadas “ b ” em relação ao sistema de coordenadas “ a ”
${}^a R_b$	Orientação do sistema de coordenadas “ b ” em relação ao sistema de coordenadas “ a ”
${}^a T_b$	Matriz de transformação homogênea do sistema de coordenadas “ b ” em relação ao sistema de coordenadas “ a ”
θ_i	Ângulo de giro no eixo da junta “ i ” ($^\circ$)
a_i	Comprimento do elo ligado à junta “ i ” (mm)
d_i	Deslocamento linear da junta “ i ” (mm)
α_i	Ângulo de torção do elo ligado à junta “ i ” ($^\circ$)
φ	Ângulo de rolagem ($^\circ$)
ϑ	Ângulo de arfagem ($^\circ$)
ψ	Ângulo de guinada ($^\circ$)
v_x	Velocidade linear no eixo x do sistema de coordenadas (mm/s)
v_y	Velocidade linear no eixo y do sistema de coordenadas (mm/s)
v_z	Velocidade linear no eixo z do sistema de coordenadas (mm/s)
K_p	Valor do ganho proporcional do controlador PID
K_i	Valor do ganho integral do controlador PID
K_d	Valor do ganho derivativo do controlador PID
K_u	Valor do ganho proporcional crítico
T_u	Período de oscilação com o ganho proporcional crítico (s)

SUMÁRIO

1	INTRODUÇÃO	29
2	BRAÇO ROBÓTICO	34
2.1	MOTORES.....	37
2.2	ENCODERS.....	38
2.3	CHAVES FIM DE CURSO	40
2.4	GARRA	40
2.5	CABEAMENTO.....	41
3	MODELAGEM MATEMÁTICA	44
3.1	MATRIZ DE TRANSFORMAÇÃO HOMOGÊNEA	44
3.2	PARÂMETROS DE DENAVIT-HARTENBERG	47
3.3	CINEMÁTICA DIRETA.....	50
3.4	CINEMÁTICA INVERSA.....	55
3.5	CINEMÁTICA DIFERENCIAL	61
4	IMPLEMENTAÇÃO DO PROJETO	65
4.1	PLACA CONTROLADORA.....	65
4.2	PLACA DE INTEGRAÇÃO DE COMPONENTES.....	68
4.3	DRIVERS DOS MOTORES	70
4.4	MÓDULO DE COMUNICAÇÃO WIRELESS.....	73
4.5	INTERFACE DE OPERAÇÃO	75
4.6	PROTÓTIPO FINAL.....	79
5	PROGRAMAÇÃO DO CONTROLADOR	84
5.1	ACIONAMENTO DOS MOTORES	85
5.2	REALIMENTAÇÃO PELOS ENCODERS	86
5.3	POSIÇÃO DE REFERÊNCIA.....	87
5.4	BIBLIOTECAS AUXILIARES	91

5.4.1 Bibliotecas do módulo ESP8266	91
5.4.2 Biblioteca MatrixMath	91
5.5 FUNÇÕES AUXILIARES	92
5.5.1 Matriz de Transformação Homogênea	92
5.5.2 Cinemática Inversa	92
5.5.3 Jacobiano	92
5.5.4 Controle PID	93
5.5.5 Estimativa de tempo entre pontos	94
5.5.6 Geração de caminhos	95
5.6 MOVIMENTAÇÃO DO BRAÇO ROBÓTICO	96
6 RESULTADOS	98
6.1 SINTONIZAÇÃO DE PARÂMETROS	98
6.2 EXECUÇÃO DE TRAJETÓRIAS	99
6.3 CONTROLE DE POSICIONAMENTO	110
7 CONCLUSÃO	116
REFERÊNCIAS	119

1 INTRODUÇÃO

Os primeiros dispositivos industriais automatizados tinham a finalidade de manipular materiais perigosos para o ser humano, sendo controlados diretamente por um operador em um sistema mestre-escravo (BARRIENTOS, PEÑIN, *et al.*, 2007). O progresso industrial e a busca por sistemas de produção cada vez mais dinâmicos e eficientes impulsionaram o aperfeiçoamento desses dispositivos com a criação de sistemas de controle que substituíram a necessidade de um operador, dando origem ao conceito de robô. Dessa maneira, as aplicações de sistemas robóticos solucionaram problemas que exigiam esforços humanos excessivos, atividades repetitivas e/ou alto grau de precisão, levando ao desenvolvimento de tarefas como a soldagem, pintura, montagem, carga e descarga, inspeção militar e aplicações espaciais (SAHA, 2008).

A norma dada pela (ISO 10218, 1992) define um robô industrial como sendo “uma máquina manipuladora com vários graus de liberdade controlada automaticamente, reprogramável, multifuncional, que pode ter base fixa ou móvel para utilização em aplicações de automação industrial”.

Nesse contexto, destacam-se os braços robóticos, que são robôs manipuladores dedicados à manufatura a fim de baratear os custos na produção em série com a garantia de qualidade nos produtos, proporcionando amplos espaços de trabalho onde são programadas trajetórias para realização de suas atividades (SANTOS, 2004). O controlador desse tipo de robô deve ser projetado a fim de atender às exigências das tarefas de maneira eficaz, considerando os parâmetros e limitações dos seus dispositivos atuadores, por ser de uso frequente no desenvolvimento de tarefas que exigem precisão, repetição ou que ofereçam algum risco à saúde do homem (WALTER, 1996).

Com a tendência de redução nos custos dos componentes devido aos avanços tecnológicos, é notável que a robótica seja utilizada em campos de aplicação cada vez mais amplos. Esse fato, aliado à criação de métodos e alternativas para programação de robôs, possibilitou a inclusão da robótica em diferentes níveis de educação (CIELNIAK, BELLOTTO e DUCKETT, 2012). Logo, o acesso ao estudo prático de dispositivos robóticos representa uma opção para a validação da teoria da robótica, visto que esses sistemas simulam processos

industriais como linhas de montagem, células de manufatura, operação de ferramentas, entre outras aplicações.

Um sistema robótico apresenta um nível de complexidade que exige diversas competências para o desenvolvimento do seu projeto. Assim, um robô pode representar uma ferramenta didática interessante para verificação de conhecimentos nas áreas elétrica, mecânica e mecatrônica, por meio da análise de *hardware*, ou nas teorias de controle, programação e até mesmo matemática, com a análise do controlador. No entanto, a grande maioria dos robôs possuem sistemas proprietários que não permitem um estudo mais aprofundado de suas funcionalidades, ou seja, não há detalhes da implementação do controlador e suas interações com os demais componentes do sistema, limitando o estudo da robótica na operação da interface do robô para determinação das suas ações (SANTINI, 2009).

O desenvolvimento de sistemas robóticos em arquitetura aberta possibilita um estudo aprofundado da teoria da robótica, além de permitir a contribuição de diferentes pesquisadores aos projetos, promovendo a melhoria contínua dos trabalhos ao longo do tempo. Outra vantagem da abertura do sistema é a possibilidade de inclusão de recursos e novas tecnologias ao robô, incrementando suas ações e atualizando suas funcionalidades. Para tanto, é necessário utilizar um sistema de desenvolvimento baseado em uma plataforma computacional não proprietária, linguagem de controle padrão e possuir um barramento de comunicação que consiga integrar diversos periféricos (FORD, 1994).

O objetivo deste trabalho é criar um sistema didático para o ensino da robótica por meio da construção de um projeto simples e moderno com a estrutura de programação aberta. Dessa forma, os estudantes poderão verificar os assuntos acerca da robótica e de outras áreas do conhecimento no funcionamento do braço robótico em ambiente de laboratório, uma vez que as aulas práticas podem despertar a curiosidade e o interesse do aluno sobre o assunto e estimular a construção de ideias que se relacionem com a teoria estudada (LEITE, SILVA e VAZ, 2005). Além disso, o desenvolvimento deste controlador aberto também permite o aperfeiçoamento do projeto original por meio de modificações como a alteração das estratégias de controle ou a instalação de novos recursos, por exemplo.

A preocupação com o ensino da robótica abordando aspectos práticos mais amplos já foi explorada em trabalhos como em (AMARAL, 1985) que trata do

desenvolvimento de um laboratório dedicado para este fim, utilizando um controlador aberto. No entanto, em trabalhos como (LAGES e BRACARENSE, 2003), que tratam da adaptação de um braço robótico comercial, ou (HIRZINGER, FISCHER, *et al.*, 1999), que descrevem o desenvolvimento de um robô manipulador acadêmico, a intenção de criar um sistema robótico flexível é mais evidente devido à utilização de arquitetura aberta dedicada para fins educativos que possibilitam a integração e a ideia de colaboração entre estudantes, pesquisadores e outros projetistas.

O estudo apresentado neste trabalho baseia-se no braço robótico modelo Scrobot-ER4pc que se trata de um sistema versátil e confiável para treinamento e educação em robótica industrial (ESHED ROBOTEC, 1999). Apesar de ser um robô didático, seu controlador possui um sistema proprietário que enfatiza o estudo do sistema na sua operação prática, fato que limita o aprofundamento na teoria da robótica. Diversos trabalhos foram desenvolvidos a partir da análise de robôs modelo Scrobot, como mostrado na Tabela 1, evidenciando a importância de um controlador que possibilite a validação dos resultados teóricos obtidos.

Tabela 1 - Relação de trabalhos com robôs Scrobot

Autor(es)	Modelo do Robô	Estudo
(MIRELES, 2006)	Scrobot-ER V Plus	Análise comparativa de metodologias para o controle do robô
(CHAUDHARY, PRASAD e SUKAVANUM, 2012)	Scrobot-ER V Plus	Controle de trajetórias baseado em abordagem cinemática
(VERMA e DESHPANDE, 2011)	Scrobot-ER V Plus	Análise da posição do efetuador por modelo matemático
(DESHPANDE e GEORGE, 20012)	Scrobot-ER V Plus	Solução analítica para cinemática inversa
(NORIA, MUÑOZ e PIEDRA, 2014)	Scrobot-ER 4u	Controle PD em dois elos com compensação gravitacional
(GIERLAK, 2014)	Scrobot-ER 4pc	Controle adaptativo a partir de modelo dinâmico do robô
(PRABHU, ANAND, <i>et al.</i> , 2016)	Scrobot-ER Vu Plus	Modelagem dinâmica do braço robótico

Fonte: produção do próprio autor.

Os trabalhos citados na Tabela 1 fundamentam a análise dos resultados a partir de modelos matemáticos ou verificações teóricas por meio de recursos computacionais. Para o caso descrito neste trabalho, a validação da teoria pode ser

realizada diretamente no sistema robótico. Isso se deve ao fato de que o controlador original do robô Scrobot-ER 4pc foi substituído por um sistema em arquitetura aberta composto por dispositivos de baixo custo e fácil integração com outros periféricos, buscando uma solução mais versátil, moderna e que tornasse possível análises e modificações na programação.

No controle deste projeto, um microcontrolador realiza o processamento das tarefas a partir de sinais de entradas (*encoders*, chaves fim de curso e teclas) para acionar os motores do braço robótico com base na programação realizada. As trajetórias desenvolvidas pelo robô podem ser realizadas de maneira contínua ou entre pontos previamente programados e sua operação é realizada via protocolo TCP/IP, o que torna possível o controle do sistema por meio de dispositivos como computadores e *smartphones* que possuam o código de acesso do dispositivo, garantindo a segurança no acesso e operação do projeto. O usuário pode controlar as juntas individualmente ou simultaneamente descrevendo um movimento baseado nos eixos cartesianos dentro do espaço de trabalho. O efetuador pode ter sua orientação alterada em movimentos de arfagem e rolagem, além de possuir uma garra com controle de abertura por meio da leitura do *encoder* do motor que a aciona.

A partir do que foi exposto neste primeiro capítulo, o Capítulo 2 apresenta uma descrição do robô utilizado no projeto, bem como as características do controlador original e dos elementos necessários para sua implementação no estudo do sistema.

Em seguida, no Capítulo 3, é mostrada uma breve apresentação dos conceitos relevantes para o entendimento da criação do projeto, abordando assuntos da robótica e outros aspectos importantes para o desenvolvimento do controlador. Com base nas informações apresentadas, o trabalho demonstra o processo de modelagem matemática do braço robótico levando em consideração suas condições físicas.

Com isso, no Capítulo 4 são detalhados os elementos utilizados no desenvolvimento da solução para o acionamento e controle do sistema. Implementando os componentes do projeto, o Capítulo 5 demonstra os algoritmos de programação e outras estratégias utilizadas para a operação do robô.

O Capítulo 6 demonstra os resultados e o desempenho do controlador a partir do comportamento do braço robótico, bem como ensaios de calibração e precisão do efetador.

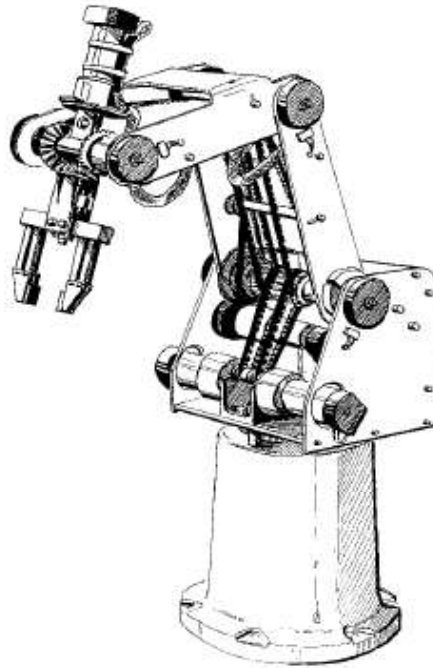
Por fim, são realizadas as conclusões sobre o projeto, com algumas abordagens, exemplos de aplicação didática do projeto e roteiros de experiências com base nas funcionalidades do controlador, além de algumas sugestões de melhorias para trabalhos futuros.

2 BRAÇO ROBÓTICO

O Scrobot-ER 4pc é um robô antropomórfico fabricado pela empresa Intelitek, projetado e desenvolvido para simular um robô industrial (ESHED ROBOTEC, 1999). A sua estrutura possui seis graus de liberdade compostos por juntas rotativas, além de um efetuator do tipo garra com movimento prismático.

O esquema mostrado na Figura 1 apresenta uma visão geral do robô, cuja estrutura permite que o usuário estude seus mecanismos internos no qual é possível observar que seu sistema de transmissão é composto por um conjunto de redutores, correias e polias.

Figura 1 - Visão geral do Scrobot-ER 4pc

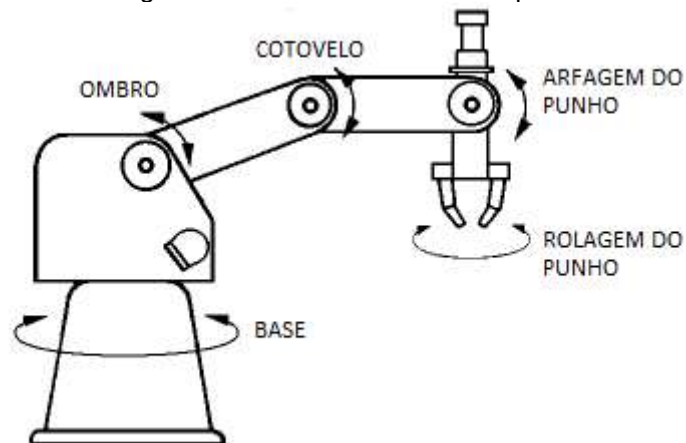


Fonte: (ESHED ROBOTEC, 1999).

Também pode-se notar cada um dos seus eixos: base, ombro, cotovelo e os movimentos do efetuator (inclinação e giro). Todas as juntas e a garra são atuadas por motores CC com *encoders* acoplados e seus movimentos são limitados por chaves fim de curso (ESHED ROBOTEC, 1999).

A denominação de cada junta, bem como o tipo de movimentação que cada uma apresenta, está representada na Figura 2.

Figura 2 - Juntas do Scorbot-ER4pc



Fonte: (ESHED ROBOTEC, 1999).

Um resumo das especificações técnicas do braço robótico pode ser observado na Tabela 2, na qual podem ser consultados dados como suas características físicas, suas limitações e a descrição geral dos demais elementos responsáveis pelo seu funcionamento.

Tabela 2 - Especificações técnicas do Scorbot-ER 4pc

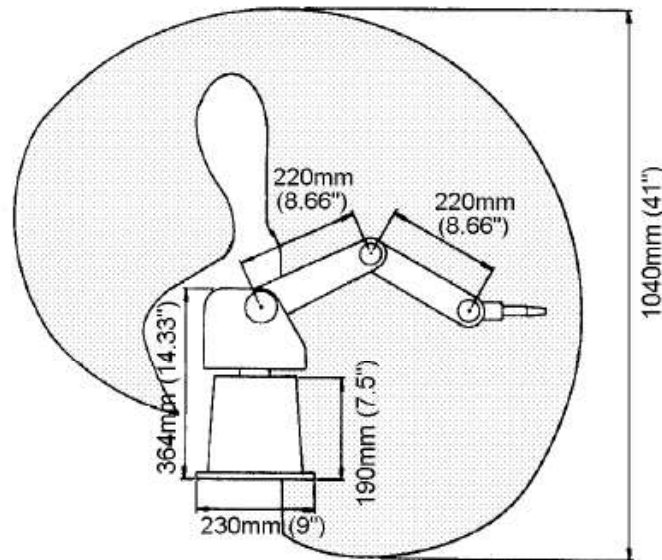
Estrutura mecânica	Vertical articulada
Quantidade de eixos	5 eixos mais um efetuator (garra)
Movimento dos eixos:	
Eixo 1: Rotação da base	310°
Eixo 2: Rotação do ombro	+130° / -35°
Eixo 3: Rotação do cotovelo	±130°
Eixo 4: Arfagem do punho	±130°
Eixo 5: Rolagem do punho	Ilimitado
Raio de operação máximo	610 mm
Efetuator	Garra com movimento paralelo
Abertura máxima da garra	75 mm
Posição de referência	Definida pelas chaves fim de curso
Realimentação	Encoder em cada eixo
Atuadores	Servomotores 12 Vcc
Capacidade dos motores (eixos 1-6)	70 W
Transmissão:	
Motores 1, 2 e 3	127,1:1
Motores 4 e 5	65,5:1
Motor 6 (garra)	19,5:1
Carga máxima	1 kg (incluindo a garra)
Repetibilidade	±0,2 mm
Peso	11,5 kg
Velocidade máxima	600 mm/s
Temperatura de operação	2° - 40°C

Fonte: (ESHED ROBOTEC, 1999).

Na Figura 3 estão esquematizadas as dimensões de cada parte do braço robótico, importantes para definição de parâmetros no projeto e para limitar as regiões que o robô pode alcançar, definindo seu espaço de trabalho. A Figura 4 representa a faixa de operação máxima do robô em vista superior.

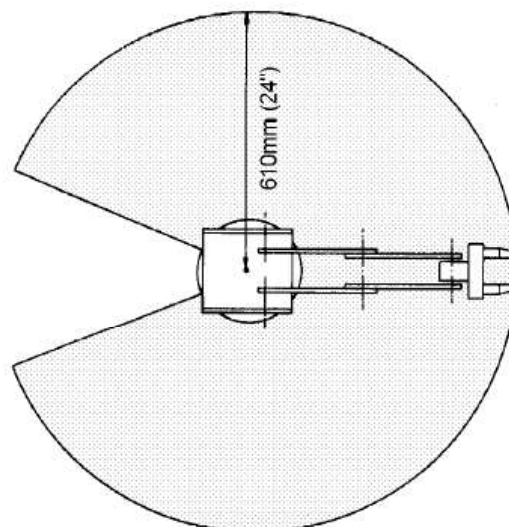
A representação dos componentes, com as respectivas denominações que serão utilizadas neste trabalho, é demonstrada na Figura 5. Os principais dispositivos são detalhados a seguir para o entendimento do funcionamento e adaptação neste projeto.

Figura 3 - Espaço de trabalho em vista lateral



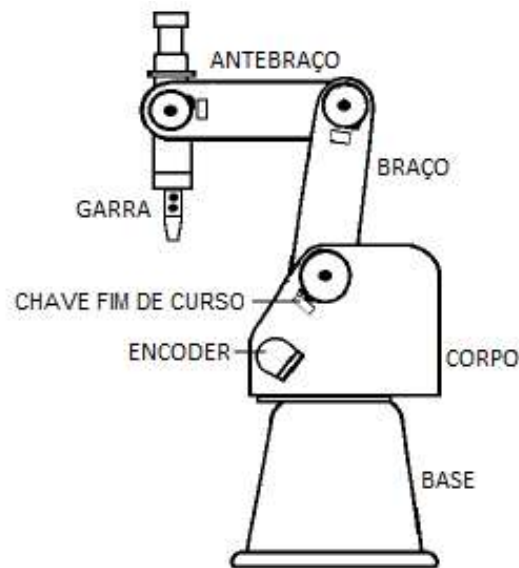
Fonte: (ESHED ROBOTEC, 1999).

Figura 4 - Espaço de trabalho em vista superior



Fonte: (ESHED ROBOTEC, 1999).

Figura 5 - Componentes do Scorbot-ER 4 pc



Fonte: (ESHED ROBOTEC, 1999).

2.1 MOTORES

A rotação das juntas do braço robótico é realizada por meio de cinco servomotores que são responsáveis pela transmissão de movimento, conforme mostra a Tabela 3, na qual é possível verificar que o funcionamento do punho depende do acionamento conjunto de dois motores. Um sexto servomotor é responsável pela movimentação da garra do robô por meio de um sistema de transmissão dedicado que realiza o movimento de abertura e fechamento do tipo prismático.

Tabela 3 - Relação de motores por junta

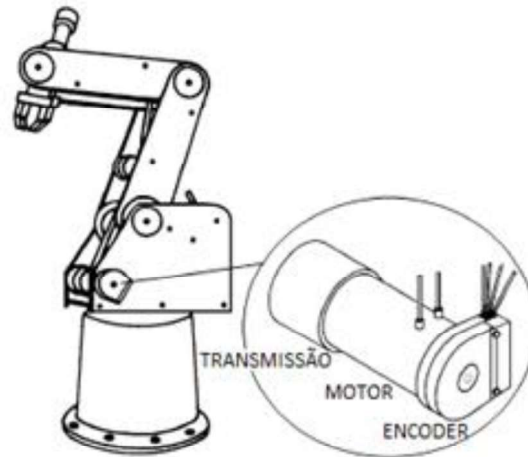
Eixo	Junta	Movimento	Motor
1	Base	Rotaciona o corpo	1
2	Ombro	Eleva e abaixa o braço	2
3	Cotovelo	Eleva e abaixa o antebraço	3
4	Articulação do punho	Eleva e abaixa o efetuador	4+5
5	Rolagem do punho	Rotaciona o efetuador	4+5

Fonte: produção do próprio autor.

A tensão nominal dos motores é 12 volts em corrente contínua e o sentido de rotação é definido pela polaridade nos seus terminais, determinando também o movimento de cada elo (ESHED ROBOTEC, 1999). A transmissão dos motores é realizada de maneira distinta para cada junta, sendo utilizado um sistema diferente

de engrenagens e/ou correias para cada situação. Cada motor possui um *encoder* correspondente para controle em malha fechada como mostra a Figura 6.

Figura 6 - Detalhe do motor da junta



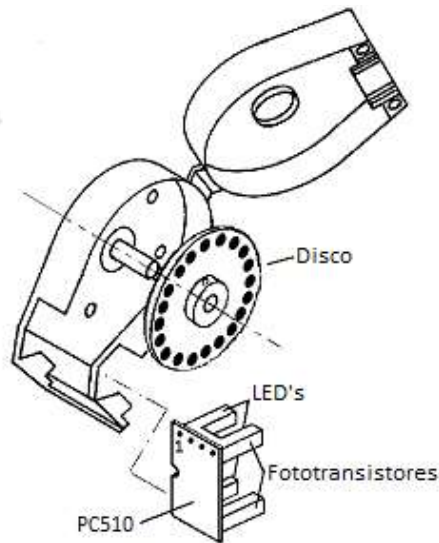
Fonte: (ESHED ROBOTEC, 1999).

2.2 ENCODERS

Para controlar os movimentos do braço robótico, cada junta possui um *encoder* óptico acoplado ao seu eixo para detectar a rotação da articulação e, conseqüentemente, a posição angular das juntas do robô. Além disso, há também um *encoder* responsável por definir o estado de abertura da garra. Assim, é possível alimentar o controlador com informações sobre a posição do efetuator e definir o estado do efetuator para um ponto desejado.

As informações enviadas pelo *encoder* se baseiam em pulsos elétricos emitidos de acordo com a rotação do motor por meio de um circuito denominado PC510 que possui um disco entre um par de LEDs e um par de fototransistores como mostra a Figura 7.

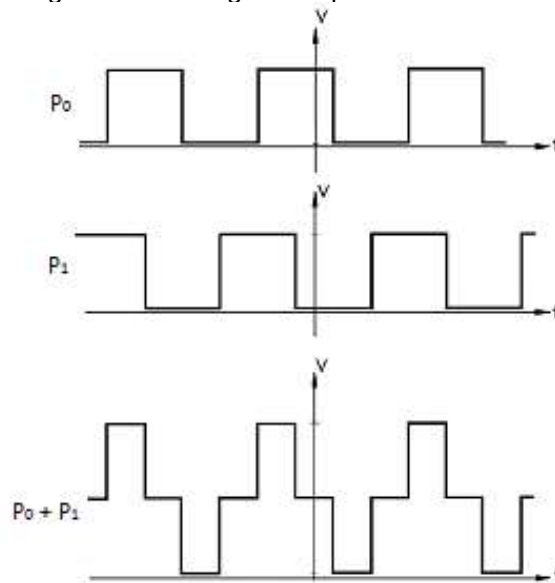
Os dados são gerados pela captação dos sinais luminosos que passam pelos orifícios do disco e são convertidos em pulsos elétricos que se alternam em níveis altos e baixos. O número de sinais é proporcional a quanto o eixo moveu-se e a sequência dos pulsos emitidos indica o sentido de rotação do motor. A partir dessas informações, o controlador define qual é o estado do robô e como executar as ações para a realização das tarefas.

Figura 7 - Detalhe do *encoder*

Fonte: (ESHED ROBOTEC, 1999).

Na Figura 8, é possível verificar o modo de operação do *encoder* a partir dos sinais gerados pelos fototransistores P0 e P1. O posicionamento de cada componente do circuito define como será a emissão do pulso elétrico, pois o bloqueio ou passagem da luz ocorre de maneira defasada entre os fototransistores de acordo com o giro do disco acoplado ao eixo do motor. Assim, os sinais gerados são comparados e a ordem dos sinais captados define qual é o sentido de rotação do motor, bem como a variação dos pulsos ao longo do tempo determina o deslocamento e a velocidade angular (ESHED ROBOTEC, 1999).

Figura 8 - Sinais gerados pelos fototransistores

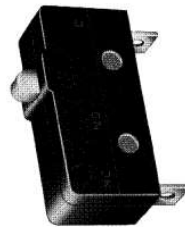


Fonte: (ESHED ROBOTEC, 1999).

2.3 CHAVES FIM DE CURSO

As chaves fim de curso definem a referência para cada junta e servem para estabelecer a posição inicial do braço robótico (ESHED ROBOTEC, 1999). Cada eixo possui um dispositivo posicionado de acordo com a calibração referencial que funciona como um interruptor capaz de determinar para o controlador a posição física de cada elo do robô. Isso ocorre por meio de um sinal elétrico que é interrompido quando a junta atinge o ângulo desejado e aciona a chave fim de curso, o que pode ser observado na Figura 9.

Figura 9 - Detalhe da chave fim de curso



Fonte: (ESHED ROBOTEC, 1999).

Durante o procedimento de localização da posição de referência, há uma ordem de ações no braço robótico no qual uma articulação move-se por vez até que atinja o ângulo adequado e interrompa o sinal no controlador por meio das chaves fim de curso, indicando que a junta está no ponto referencial. Quando a sequência de movimentos encerra, o robô está na posição de referência que permite o controlador definir o deslocamento do eixo dos motores com base nos valores realimentados pelos *encoders*. A rotina de programação do robô deve prever a realização do procedimento de referenciamento sempre que o sistema iniciar, já que é com base na posição encontrada que o braço robótico definirá o controle dos seus movimentos.

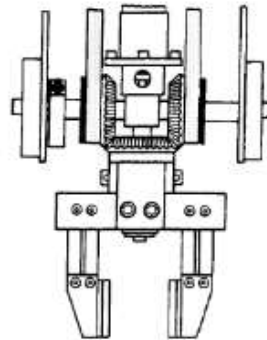
2.4 GARRA

O efetuator padrão do robô Scorbot-ER 4pc é uma garra provida de acabamentos emborrachados dedicados à manipulação de objetos como ilustra a Figura 10. No entanto, a ferramenta do braço robótico pode ser adaptada de acordo com a necessidade do operador, que pode instalar dispositivos no efetuator

inclusive com a retirada das placas de borracha da garra (ESHED ROBOTEC, 1999).

A transmissão da garra baseia-se no funcionamento de três engrenagens cônicas que formam um trem diferencial que move a articulação do punho. Quando os motores 4 e 5 são movidos em sentidos opostos, o efetuator é inclinado, realizando um movimento de arfagem. Para o caso em que motores 4 e 5 rotacionam no mesmo sentido, o efetuator se move no sentido horário e anti-horário, fazendo a ação de rolagem. Um fuso de rosca sem fim acoplado diretamente no motor 6 realiza a abertura e fechamento da garra (ESHED ROBOTEC, 1999).

Figura 10 - Detalhe da garra

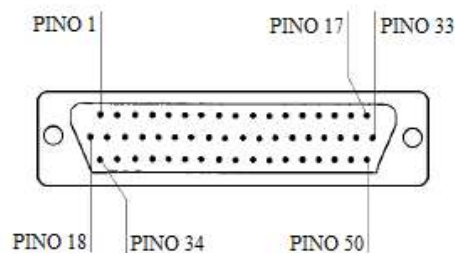


Fonte: (ESHED ROBOTEC, 1999).

2.5 CABEAMENTO

O braço robótico é conectado ao controlador por meio de um cabo localizado na base do robô que possui o terminal tipo DB-50 representado pela Figura 11. Cada pino do conector é relacionado à uma via de ligação com um componente do sistema para a transferência de informações e acionamentos.

Figura 11 - Esquema do conector DB-50



Fonte: (ESHED ROBOTEC, 1999).

Há um total de 42 vias no cabo que podem ser agrupadas em grupos de 7 vias para cada um dos 6 motores de acordo com as suas funções:

- Motor: são necessárias duas vias para alimentação elétrica de cada motor que devem ser conectadas diretamente nos respectivos terminais;
- *Encoder*: dentro do *encoder* óptico há um circuito dedicado com dois fototransistores e um diodo emissor de luz. Assim, são necessárias duas vias para enviar os sinais gerados para o controlador e uma via para alimentar o LED;
- Chave fim de curso: cada interruptor precisa de uma via para enviar um sinal para o controlador quando ele é comutado;
- Terra: uma via é usada como terra em comum com o circuito do *encoder* e a chave fim de curso.

A Tabela 4 exhibe a relação completa dos pinos e suas funções de acordo com o elemento conectado.

Tabela 4 - Cabeamento do Scorbot-ER 4pc

Elemento	Cor da via	Pino
Terminal positivo do motor 1	Branco	50
Terminal negativo do motor 1	Cinza/verde	17
Terminal positivo do motor 2	Branco	49
Terminal negativo do motor 2	Branco/verde	16
Terminal positivo do motor 3	Branco	48
Terminal negativo do motor 3	Laranja/marrom	15
Terminal positivo do motor 4	Branco	47
Terminal negativo do motor 4	Laranja/verde	14
Terminal positivo do motor 5	Branco	46
Terminal negativo do motor 5	Laranja/cinza	13
Terminal positivo do motor 6	Branco	45
Terminal negativo do motor 6	Laranja/azul	12
Terra do <i>encoder</i> 1	Branco	33
Fototransistor P1 do <i>encoder</i> 1	Branco/cinza	5
Alimentação do LED do <i>encoder</i> 1	Amarelo	11
Fototransistor P0 do <i>encoder</i> 1	Marrom	2

Terra do <i>encoder 2</i>	Branco	32
Fototransistor P1 do <i>encoder 2</i>	Branco/laranja	21
Alimentação do LED do <i>encoder 2</i>	Amarelo	27
Fototransistor P0 do <i>encoder 2</i>	Cinza	1
Terra do <i>encoder 3</i>	Branco	31
Fototransistor P1 do <i>encoder 3</i>	Marrom/azul	4
Alimentação do LED do <i>encoder 3</i>	Amarelo	10
Fototransistor P0 do <i>encoder 3</i>	Verde	36
Terra do <i>encoder 4</i>	Branco	30
Fototransistor P1 do <i>encoder 4</i>	Verde/marrom	20
Alimentação do LED do <i>encoder 4</i>	Amarelo	26
Fototransistor P0 do <i>encoder 4</i>	Laranja	35
Terra do <i>encoder 5</i>	Branco	29
Fototransistor P1 do <i>encoder 5</i>	Verde/azul	3
Alimentação do LED do <i>encoder 5</i>	Amarelo	9
Fototransistor P0 do <i>encoder 5</i>	Azul	18
Terra do <i>encoder 6</i>	Branco	28
Fototransistor P1 do <i>encoder 6</i>	Cinza/azul	19
Alimentação do LED do <i>encoder 6</i>	Amarelo	25
Fototransistor P0 do <i>encoder 6</i>	Branco/azul	34
Terra da chave fim de curso 1	Branco	33
Sinal da chave fim de curso 1	Marrom	23
Terra da chave fim de curso 2	Branco	32
Sinal da chave fim de curso 2	Cinza	7
Terra da chave fim de curso 3	Branco	31
Sinal da chave fim de curso 3	Laranja	24
Terra da chave fim de curso 4	Branco	30
Sinal da chave fim de curso 4	Verde	8
Terra da chave fim de curso 5	Branco	29
Sinal da chave fim de curso 5	Azul	6

Fonte: (ESHED ROBOTEC, 1999).

3 MODELAGEM MATEMÁTICA

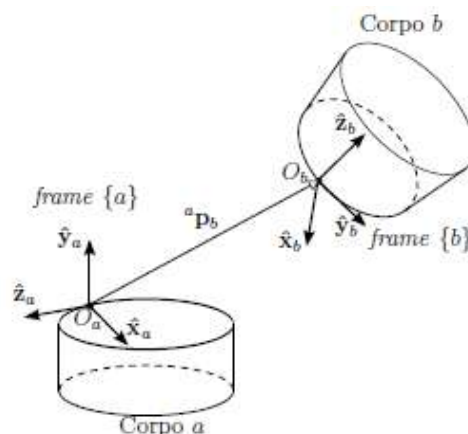
A modelagem do braço robótico abordado neste trabalho é realizada a partir do estudo da cinemática para ser aplicada a uma estratégia de controle que tem como objetivo levar o robô para um ponto desejado no espaço de acordo com sua programação. Assim, é levada em consideração a posição e a orientação das juntas do sistema, bem como suas derivadas, durante a execução das trajetórias necessárias para o desenvolvimento das tarefas programadas.

A partir da definição do ângulo de cada junta, a cinemática direta apresenta a posição e a orientação da garra. Por outro lado, também é possível estabelecer coordenadas para o efetuador e calcular os ângulos das juntas necessários para atingir o ponto por meio da cinemática inversa. A cinemática diferencial define aspectos da variação da posição dos elementos do robô no tempo, ou seja, trata da velocidade das juntas ou da garra.

3.1 MATRIZ DE TRANSFORMAÇÃO HOMOGÊNEA

A notação de matrizes por transformação homogênea combina o vetor posição e a matriz de rotação em uma única identidade matemática que define a posição e a orientação de cada junta de um robô no espaço (BARRIENTOS, PEÑIN, *et al.*, 2007).

Figura 12 – Sistemas de coordenadas de corpos



Fonte: (ROSS, SANTOS e ESPINOZA, 2011).

Para a definição de um ponto no espaço é possível fixar um sistema de coordenadas no qual a origem possui uma tríade de vetores unitários mutuamente

ortogonais (CRAIG, 1989). No caso do braço robótico estudado, a análise é realizada a partir do princípio que a postura de um corpo será sempre expressa relativamente a outro corpo, sendo estabelecido como referência um sistema de coordenadas universal imóvel e arbitrário (ROSS, SANTOS e ESPINOZA, 2011).

A Figura 12 representa dois sistemas de coordenadas denominados “*frame {a}*” e “*frame {b}*” nos quais é possível notar seus respectivos vetores que definem sua orientação, bem como as origens distanciadas pelo vetor-posição definido pela Equação (3.1) que relaciona as componentes escalares entre os pontos.

$${}^a p_b = \begin{bmatrix} {}^a p_x^b \\ {}^a p_y^b \\ {}^a p_z^b \end{bmatrix} \quad (3.1)$$

A orientação do sistema de coordenadas do “*frame {b}*” pode ser descrita expressando seus vetores unitários em relação aos do “*frame {a}*” de acordo com a Equação (3.2).

$${}^a R_b = \begin{bmatrix} {}^a \hat{x}_b & {}^a \hat{y}_b & {}^a \hat{z}_b \end{bmatrix} = \begin{bmatrix} \hat{x}_b \cdot \hat{x}_a & \hat{y}_b \cdot \hat{x}_a & \hat{z}_b \cdot \hat{x}_a \\ \hat{x}_b \cdot \hat{y}_a & \hat{y}_b \cdot \hat{y}_a & \hat{z}_b \cdot \hat{y}_a \\ \hat{x}_b \cdot \hat{z}_a & \hat{y}_b \cdot \hat{z}_a & \hat{z}_b \cdot \hat{z}_a \end{bmatrix} \quad (3.2)$$

Onde:

\hat{x}_i : Eixo x do sistema de coordenadas “*i*”

\hat{y}_i : Eixo y do sistema de coordenadas “*i*”

\hat{z}_i : Eixo z do sistema de coordenadas “*i*”

A Equação (3.3) apresenta a forma geral da matriz de transformação homogênea, na qual é descrita a origem do sistema de coordenadas do “*frame {b}*” em relação a origem do sistema de coordenadas do “*frame {a}*”, especificando as translações e rotações necessárias para atingi-la (SANTOS, 2004).

$${}^a T_b = \begin{bmatrix} {}^a R_b & {}^a p_b \\ 0^T & 1 \end{bmatrix} \quad (3.3)$$

Dessa maneira, é possível transformar qualquer vetor expresso relativamente ao “*frame {b}*” em um outro vetor relativo ao “*frame {a}*” por meio da representação homogênea da Equação (3.4).

$$\begin{bmatrix} {}^a r \\ 1 \end{bmatrix} = {}^a T_b \begin{bmatrix} {}^b r \\ 1 \end{bmatrix} \quad (3.4)$$

Com base no que foi exposto, a matriz de transformação homogênea para uma translação entre pontos em coordenadas cartesianas, sem modificar a orientação do corpo, pode ser definida pela Equação (3.5).

$$Trans(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

De maneira similar, se a posição da origem do sistema de coordenadas do corpo não se alterar, mas houver modificação de sua orientação dada por rotações em torno dos vetores unitários, uma matriz de transformação homogênea pode representar a mudança relativa aos eixos x, y e por meio da Equação (3.6), Equação (3.7) e Equação (3.8), respectivamente.

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\text{sen } \theta & 0 \\ 0 & \text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$Rot(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \text{sen } \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen } \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$$Rot(z, \theta) = \begin{bmatrix} \cos \theta & -\text{sen } \theta & 0 & 0 \\ \text{sen } \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

No caso de um braço robótico com vários graus de liberdade, é possível desenvolver as matrizes de transformação homogênea que descrevem o sistema de coordenadas de cada junta em relação à anterior e assim sucessivamente. É válido realizar a multiplicação entre as matrizes de transformação homogêneas para relacionar juntas que não são subsequentes, desde que as propriedades de pré e pós-multiplicação sejam respeitadas. Assim, com a análise partindo da base (junta 0), a Equação (3.9) define a matriz de transformação homogênea final para o caso do Scrobot-ER 4pc que considera a posição e orientação de todos os graus de liberdade do robô em relação à base.

$${}^0T_5 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 \quad (3.9)$$

Para cada matriz de transformação homogênea parcial da Equação (3.9), há uma associação com cada junta do braço robótico que pode ser relacionada com a base desde que a ordem de multiplicação esteja de acordo com a sequência das juntas do robô.

3.2 PARÂMETROS DE *DENAVIT-HARTENBERG*

Apesar do fato de que a alocação dos sistemas de coordenadas de cada junta poder ser feita de forma arbitrária, existem métodos que facilitam e tornam mais lógica a orientação de cada eixo das coordenadas do sistema. Para o estudo deste trabalho, a metodologia utilizada é a notação de *Denavit-Hartenberg* por ser uma forma simples de modelar elos e juntas de robôs a partir de qualquer configuração, independente da sua sequência e complexidade (NIKU, 2001).

A parametrização do braço robótico por esse método é realizada por meio da representação de um par junta-elo a partir de parâmetros dados pelo ângulo de giro nos eixos das juntas (θ_i), pelo comprimento dos elos (a_i), pelo deslocamento de juntas (d_i) e pelo ângulo de torção dos elos (α_i) (NAGRATH e MITTAL, 2007).

Dessa maneira, o princípio da metodologia de *Denavit-Hartenberg* estabelece que o desenvolvimento de uma matriz de transformação homogênea relaciona duas juntas consecutivas em função desses quatro parâmetros de modo que o ângulo θ define a rotação em torno do eixo z, seguido de um deslocamento d ao longo do eixo

z , um alongamento a ao longo do eixo x e uma rotação de α em torno do eixo x como pode ser resumido na Equação (3.10).

$${}^{i-1}T_i = Rot(z, \theta_i).Trans(0, 0, d_i).Trans(a_i, 0, 0).Rot(x, \alpha_i) \quad (3.10)$$

O desenvolvimento da multiplicação das matrizes representadas na Equação (3.10) resulta na matriz de transformação homogênea de duas juntas consecutivas pela representação de *Denavit-Hartenberg* como demonstra a Equação (3.11).

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\text{sen } \theta_i \cdot \cos \alpha_i & \text{sen } \theta_i \cdot \text{sen } \alpha_i & a_i \cdot \cos \theta_i \\ \text{sen } \theta_i & \cos \theta_i \cdot \cos \alpha_i & -\cos \theta_i \cdot \text{sen } \alpha_i & a_i \cdot \text{sen } \theta_i \\ 0 & \text{sen } \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

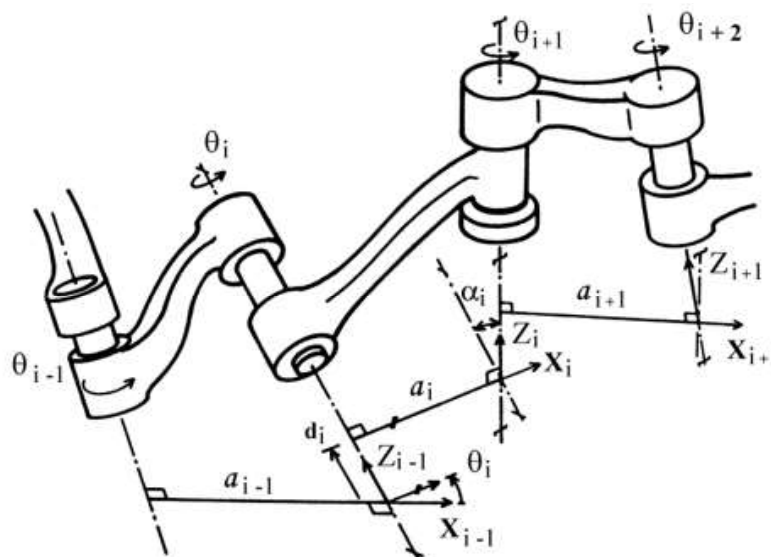
O algoritmo para obtenção da notação de *Denavit-Hartenberg* pode ser descrito da seguinte maneira (SPONG, HUTCHINSON e VIDYASAGAR, 2012):

1. Enumerar os elos iniciando com “1” e seguir a sequência até o último elo “ n ”. A base fixa do robô é numerada como elo “0”;
2. Enumerar cada junta começando com “1” para o primeiro grau de liberdade e continuar a ordem até a última junta “ n ”;
3. Verificar o eixo de cada junta. Se for rotativa, o eixo será o próprio eixo de giro. Se for prismática, o eixo segue a direção que ocorre o deslocamento;
4. Para “ i ” de “0” a “ $n-1$ ”, orientar o eixo z_i sobre o eixo da junta “ $i+1$ ”;
5. Definir a origem do sistema de coordenadas da base (S_0) em qualquer ponto do eixo z_0 . Os eixos x_0 e y_0 são orientados de forma que estabeleçam um sistema dextrogiro com o eixo z_0 ;
6. Para “ i ” de “1” a “ $n-1$ ”, situar a origem do sistema de coordenadas S_i da junta “ i ” na interseção do eixo z_i com a linha normal comum a z_{i-1} e z_i . Se ambos os eixos se cruzam, a origem de S_i se situa no ponto de corte. Se os eixos forem paralelos, a origem de S_i se situa na junta “ $i+1$ ”;
7. Orientar x_i na linha normal comum a z_{i-1} e z_i ;
8. Orientar y_i de maneira que estabeleça um sistema dextrogiro com os eixos x_i e z_i ;

9. Definir o sistema de coordenadas S_n no extremo do robô de modo que z_n coincida com a direção de z_{n-1} e x_n seja normal a z_{n-1} e z_n ;
10. Estabelecer θ_i como o ângulo de rotação em torno do eixo z_{i-1} para que x_{i-1} e x_i fiquem paralelos;
11. Definir d_i como a distância ao longo do eixo z_{i-1} entre a origem do sistema de coordenadas S_{i-1} até o ponto que x_i e x_{i-1} estejam alinhados;
12. Definir a_i como a distância ao longo do eixo x_i entre S_{i-1} e S_i ;
13. Estabelecer α_i como o ângulo de rotação em torno do eixo x_i entre S_{i-1} e S_i ;
14. Obter as matrizes de transformação homogênea parciais de acordo com a Equação (3.11);
15. Obter a matriz de transformação homogênea final de acordo com a Equação (3.9) que define a relação entre o efetuador do braço robótico com a sua base.

A Figura 13 esquematiza os parâmetros da convenção de *Denavit-Hartenberg* na qual θ_i é uma rotação do eixo x_{i-1} para o eixo x_i em torno do eixo z_{i-1} , d é um deslocamento ao longo do eixo z_{i-1} da origem O_{i-1} até o eixo x_i , a é um alongamento ao longo do eixo x_i da origem O_i até o eixo z_{i-1} e α é uma rotação do eixo z_{i-1} para o eixo z_i em torno do eixo x_i .

Figura 13 - Parâmetros de *Denavit-Hartenberg*



Fonte: (SPONG, HUTCHINSON e VIDYASAGAR, 2012).

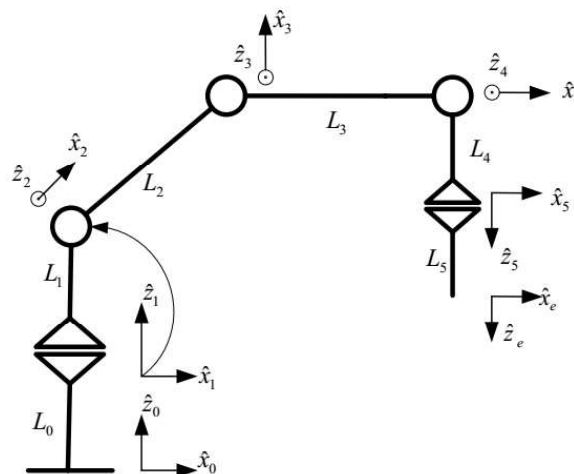
3.3 CINEMÁTICA DIRETA

O problema da cinemática direta consiste em determinar a posição e a orientação do efetuador com base nos valores para as variáveis das juntas do robô que podem ser ângulos, para o caso de juntas rotacionais, ou deslocamentos, no caso de juntas prismáticas (SPONG, HUTCHINSON e VIDYASAGAR, 2012).

O cálculo da matriz de transformação homogênea final do robô com base nos parâmetros de *Denavit-Hartenberg* é uma maneira prática de se definir a cinemática direta, pois obtém-se a posição do ponto no espaço a partir da última coluna da matriz e a sua orientação com base nas três primeiras colunas (GEMBAROWSKI e CAMPOS, 2015).

A Figura 14 mostra o esquema do robô Scrobot-ER 4pc com seus elos representados de maneira simplificada juntamente com suas juntas. Os sistemas de coordenadas para cada grau de liberdade foram estabelecidos de acordo com a notação de *Denavit-Hartenberg*, formando sistemas dextrógiros com o eixo y , omitido para melhor apresentação da disposição dos elementos.

Figura 14 - Sistemas de coordenadas do Scrobot-ER 4pc



Fonte: (TAPIA, 2010).

Com base na Figura 14 é possível analisar a configuração geral do braço robótico e definir os parâmetros de *Denavit-Hartenberg* conforme a disposição da Tabela 5 que organiza os valores de θ , d , a e α para cada uma das cinco juntas rotativas: base, ombro, cotovelo, arfagem do punho e rolagem do punho, respectivamente.

Tabela 5 – Parâmetros de *Denavit-Hartenberg*

Junta _i	θ_i	d_i	a_i	α_i
1	θ_1	$L_0 + L_1$	0	90°
2	θ_2	0	L_2	0°
3	θ_3	0	L_3	0°
4	$\theta_4 + 90^\circ$	L_4	0	90°
5	θ_5	L_5	0	0°

Fonte: produção do próprio autor.

Os valores dos elos, apresentados na Tabela 5, estão diretamente ligados ao espaço de trabalho do robô, bem como à disposição das juntas determinam a mobilidade para alcançar os pontos no espaço. Tais valores estão listados abaixo (ESHED ROBOTEC, 1999).

- $L_0 = 190\text{mm}$;
- $L_1 = 174\text{mm}$;
- $L_2 = 220\text{mm}$;
- $L_3 = 220\text{mm}$;
- $L_4 = 75\text{mm}$;
- $L_5 = 65\text{mm}$.

A partir da Equação (3.11) são definidas as matrizes de transformação homogêneas para cada par de juntas consecutivas. A primeira relação é dada pela Equação (3.12).

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 & 0 \\ 0 & -1 & 0 & L_0 + L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Por seguinte, a Equação (3.13) mostra a segunda matriz para o caso estudado.

$${}^1T_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & L_2 \cdot \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & L_2 \cdot \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

Seguindo a ordem, é apresentada a Equação (3.14) que define o terceiro par de juntas associado.

$${}^2T_3 = \begin{bmatrix} \cos \theta_3 & -\text{sen } \theta_3 & 0 & L_3 \cdot \cos \theta_3 \\ \text{sen } \theta_3 & \cos \theta_3 & 0 & L_3 \cdot \text{sen } \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

A próxima matriz é demonstrada pela Equação (3.15).

$${}^3T_4 = \begin{bmatrix} \text{sen } \theta_4 & 0 & \cos \theta_4 & 0 \\ -\cos \theta_4 & 0 & \text{sen } \theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

Por fim, a Equação (3.16) descreve a última relação que contempla a dimensão do efetuador (garra).

$${}^4T_5 = \begin{bmatrix} \cos \theta_5 & -\text{sen } \theta_5 & 0 & 0 \\ \text{sen } \theta_5 & \cos \theta_5 & 0 & 0 \\ 0 & 0 & 1 & L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

Por meio da Equação (3.9) é determinada a matriz de transformação homogênea final definida pela Equação (3.17) que define a posição e a orientação do efetuador em relação à base do braço robótico, multiplicando as matrizes resultantes de acordo com a sequência de juntas da configuração do robô.

$${}^0T_5 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

Sendo:

$$n_x = \text{sen } \theta_1 \cdot \text{sen } \theta_5 + \cos \theta_1 \cdot \cos \theta_5 \cdot \text{sen}(\theta_2 + \theta_3 + \theta_4)$$

$$n_y = \cos \theta_5 \cdot \text{sen } \theta_1 \cdot \text{sen}(\theta_2 + \theta_3 + \theta_4) - \cos \theta_1 \cdot \text{sen } \theta_5$$

$$n_z = \cos(\theta_2 + \theta_3 + \theta_4) \cdot \cos \theta_5$$

$$\begin{aligned}
s_x &= \cos \theta_5 \cdot \text{sen } \theta_1 - \cos \theta_1 \cdot \text{sen}(\theta_2 + \theta_3 + \theta_4) \cdot \text{sen } \theta_5 \\
s_y &= -\cos \theta_1 \cdot \cos \theta_5 - \text{sen } \theta_1 \cdot \text{sen}(\theta_2 + \theta_3 + \theta_4) \cdot \text{sen } \theta_5 \\
s_z &= -\cos(\theta_2 + \theta_3 + \theta_4) \cdot \text{sen } \theta_5 \\
a_x &= \cos \theta_1 \cdot \cos(\theta_2 + \theta_3 + \theta_4) \\
a_y &= \cos(\theta_2 + \theta_3 + \theta_4) \cdot \text{sen } \theta_1 \\
a_z &= -\text{sen}(\theta_2 + \theta_3 + \theta_4) \\
p_x &= \cos \theta_1 \cdot \begin{bmatrix} L_2 \cdot \cos \theta_2 + L_3 \cdot \cos(\theta_2 + \theta_3) + \\ L_5 \cdot \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix} \\
p_y &= \text{sen } \theta_1 \cdot \begin{bmatrix} L_2 \cdot \cos \theta_2 + L_3 \cdot \cos(\theta_2 + \theta_3) + \\ L_5 \cdot \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix} \\
p_z &= \begin{bmatrix} L_0 + L_1 - L_2 \cdot \text{sen } \theta_2 - L_3 \cdot \text{sen}(\theta_2 + \theta_3) + \\ -L_5 \cdot \text{sen}(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}
\end{aligned}$$

Pela verificação da Equação (3.17) é definida a posição do efetuador do robô no espaço em relação à base a partir da última coluna da matriz que apresenta as coordenadas nos eixos x , y e z representadas pelos valores de p_x , p_y e p_z . A Equação (3.18) pode ser considerada o vetor de posição da garra.

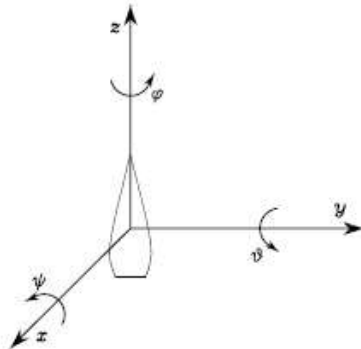
$${}^0P_5 = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (3.18)$$

De maneira análoga, a orientação do efetuador em relação à base do robô é estabelecida pela análise das três primeiras colunas da Equação (3.17) que estabelecem nove elementos como demonstrados na Equação (3.2). Assim, é possível simplificar os vetores-colunas da matriz de transformação homogênea final de maneira que esteja relacionada com uma matriz de rotação dada pela Equação (3.19).

$${}^0R_5 = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} \quad (3.19)$$

Apenas três parâmetros são suficientes para descrever a orientação do braço robótico em relação à base e, por se tratar de uma matriz 3x3, há termos redundantes apresentados. Um método empregado para simplificar a representação dada pela Equação (3.19) é a utilização dos ângulos de Euler que definem rotações em torno dos eixos do sistema de coordenadas para determinar a orientação no espaço. Há 12 possibilidades de sequência de rotações distintas, sendo a denominada RPY (*Roll-Pitch-Yaw*), que representam os movimentos de rolagem, arfagem e guinada, respectivamente, a utilizada neste trabalho. A Figura 15 demonstra a ordem dos movimentos *roll* (φ), *pitch* (ϑ) e *yaw* (ψ) que definem a orientação final do corpo.

Figura 15 – Representação dos ângulos RPY



Fonte: (SICILIANO, SCIAVICCO, *et al.*, 2009).

A representação RPY define três ângulos que representam uma sequência de três rotações em torno dos eixos fixos, determinando a orientação do efetuador do robô.

A orientação resultante da sequência RPY é realizada da seguinte forma:

1. Definir o ângulo ψ em torno do eixo x (*yaw*) de acordo com a Equação (3.6);
2. Definir o ângulo ϑ em torno do eixo y (*pitch*) de acordo com a Equação (3.7);
3. Definir o ângulo φ em torno do eixo z (*roll*) de acordo com a Equação (3.8).

A Equação (3.20) resume esse procedimento, respeitando a ordem dos termos na multiplicação matricial.

$$RPY(\varphi, \vartheta, \psi) = Rot(z, \varphi).Rot(y, \vartheta).Rot(x, \psi) \quad (3.20)$$

É possível relacionar a matriz resultante da Equação (3.20) com a matriz de rotação apresentada na Equação (3.19), sendo que os termos são dados por:

$$\begin{aligned}
 n_x &= \cos \varphi \cdot \cos \vartheta \\
 n_y &= \sin \varphi \cdot \cos \vartheta \\
 n_z &= -\sin \vartheta \\
 s_x &= \cos \varphi \cdot \sin \vartheta \cdot \sin \psi - \sin \varphi \cdot \cos \psi \\
 s_y &= \sin \varphi \cdot \sin \vartheta \cdot \sin \psi - \cos \varphi \cdot \cos \psi \\
 s_z &= \cos \vartheta \cdot \sin \psi \\
 a_x &= \cos \varphi \cdot \sin \vartheta \cdot \cos \psi - \sin \varphi \cdot \sin \psi \\
 a_y &= \sin \varphi \cdot \sin \vartheta \cdot \cos \psi - \cos \varphi \cdot \sin \psi \\
 a_z &= \cos \vartheta \cdot \cos \psi
 \end{aligned}$$

Nessas condições, é possível estabelecer algumas relações trigonométricas entre os termos da matriz que definem os valores dos ângulos φ , ϑ e ψ como demonstra a Equação (3.21), a Equação (3.22) e a Equação (3.23), respectivamente.

$$\varphi = \arctan\left(\frac{n_y}{n_x}\right) \quad (3.21)$$

$$\vartheta = \arctan\left(-\frac{n_z}{\sqrt{s_z^2 + a_z^2}}\right) \quad (3.22)$$

$$\psi = \arctan\left(\frac{s_z}{a_z}\right) \quad (3.23)$$

Determinando os valores das relações encontradas com os termos definidos a partir da matriz de transformação homogênea final dada pela Equação (3.17), os valores de φ , ϑ e ψ correspondem à orientação do efetuador em relação à base do robô.

3.4 CINEMÁTICA INVERSA

O cálculo da cinemática inversa é utilizado quando se deseja obter os ângulos das juntas a partir da posição do efetuador. Em diversas aplicações a posição e a orientação da ferramenta ou elemento final do robô devem ser controladas para que

descrevam trajetórias convenientes. Dessa forma, todas as juntas precisam se movimentar de maneira solidária para que o efetuador atinja os pontos desejados no espaço. De maneira geral, a cinemática inversa tem como finalidade determinar todas as configurações possíveis das juntas para que alcancem a posição e orientação especificada para o efetuador em relação à base do robô (NAGRATH e MITTAL, 2007).

Um dos métodos usados para determinar as relações que definem a cinemática inversa é a análise da configuração do robô com base na influência de cada junta na determinação dos parâmetros do efetuador no espaço. Sendo assim, é possível determinar as equações que regem a cinemática inversa a partir de intuições algébricas e geométricas (SICILIANO, SCIAVICCO, *et al.*, 2009).

Este trabalho utiliza o método geométrico para a definição da cinemática inversa do robô. Para isto, o braço robótico foi dividido em duas partes, considerando que as três primeiras juntas definem a posição do punho e as duas últimas juntas definem a orientação da garra a partir dos movimentos de arfagem e rolagem.

Dessa maneira, a determinação da cinemática inversa é dividida em dois subproblemas, uma vez que a solução da posição é separada da orientação (SICILIANO, SCIAVICCO, *et al.*, 2009). Portanto, é necessário descontar a dimensão do punho no cálculo da cinemática inversa da posição, como mostra a Equação (3.24), considerando que o efetuador influencia na determinação da posição do braço robótico de acordo com sua orientação em termos de z_e como ilustra a Figura 14.

$${}^0P_3 = {}^0P_5 - (L_4 + L_5)\hat{z}_e \quad (3.24)$$

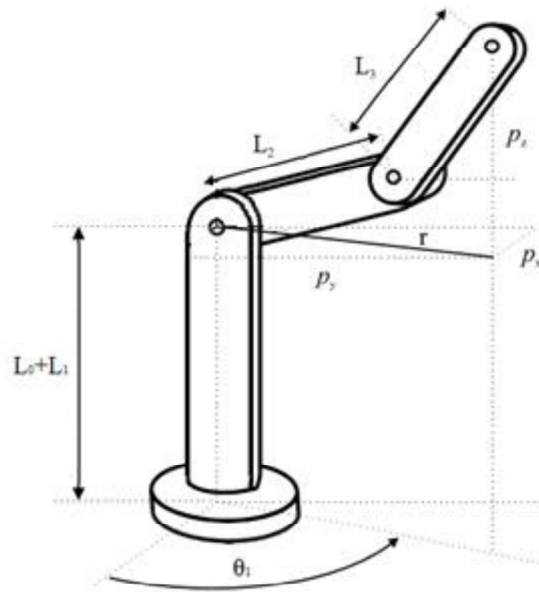
A Equação (3.24) considera que a posição desejada para o braço robótico é definida pela matriz de transformação homogênea definida pela Equação (3.17), na qual a última coluna define o vetor de posição utilizado no cálculo como demonstra a Equação (3.18).

Logo, as coordenadas de p_x , p_y e p_z usadas na determinação da cinemática inversa são definidas pela Equação (3.25) que considera a posição do punho do robô.

$${}^0P_3 = \begin{bmatrix} p_{x3} \\ p_{y3} \\ p_{z3} \end{bmatrix} \quad (3.25)$$

A Figura 16 apresenta o braço robótico simplificado representando a primeira parte do robô, responsável pela definição da posição do punho no espaço.

Figura 16 – Braço robótico simplificado



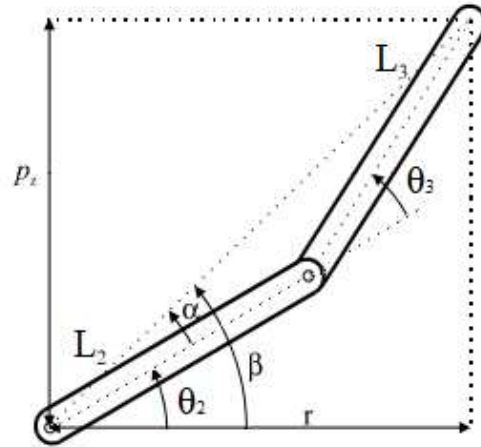
Fonte: (BARRIENTOS, PEÑIN, *et al.*, 2007).

Com base na Figura 16, é notável que o ângulo θ_1 da junta da base está diretamente relacionado com o efetuador nos eixos x e y de modo que as coordenadas p_x e p_y correspondem aos catetos de um triângulo. Logo, o ângulo da junta da base para o ponto desejado pode ser determinado pela Equação (3.26).

$$\theta_1 = \arctan\left(\frac{p_{y3}}{p_{x3}}\right) \quad (3.26)$$

A fim de facilitar a análise da geometria do robô, a Figura 17 apresenta um detalhe do braço robótico simplificado enfatizando os elos L_2 e L_3 visualizados lateralmente. Dessa maneira, o vetor r representado corresponde a linha formada pelas coordenadas p_x e p_y .

Figura 17 – Detalhe do braço robótico simplificado



Fonte: (BARRIENTOS, PEÑIN, *et al.*, 2007).

Pela lei dos cossenos é verificada a Equação (3.27) de acordo com a análise geométrica da Figura 17, na qual é considerado um triângulo cujos lados correspondem aos elos do robô e uma linha definida pelo teorema de Pitágoras a partir de p_z e r .

$$r^2 + (p_{z3} - (L_0 + L_1))^2 = L_2^2 + L_3^2 - 2L_2L_3 \cos(\pi - \theta_3) \quad (3.27)$$

Manipulando o ângulo θ_3 e expandindo r em termos de p_x e p_y se obtém a Equação (3.28).

$$p_{x3}^2 + p_{y3}^2 + (p_{z3} - (L_0 + L_1))^2 = L_2^2 + L_3^2 + 2L_2L_3 \cos \theta_3 \quad (3.28)$$

A Equação (3.28) pode ser reescrita enfatizando o ângulo θ_3 como definido na Equação (3.29).

$$\cos \theta_3 = \frac{p_{x3}^2 + p_{y3}^2 + (p_{z3} - (L_0 + L_1))^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (3.29)$$

Dessa forma, podemos isolar o ângulo θ_3 por uma relação trigonométrica a partir da definição trivial dada pela Equação (3.30).

$$\sin \theta_3 = \pm \sqrt{1 - \cos^2 \theta_3} \quad (3.30)$$

Por fim, determina-se o ângulo θ_3 por meio da Equação (3.31).

$$\theta_3 = \arctan\left(\frac{\text{sen } \theta_3}{\text{cos } \theta_3}\right) \quad (3.31)$$

Para determinar o ângulo θ_2 são calculados os ângulos auxiliares α e β mostrados na Figura 17. A Equação (3.32) define o ângulo α , determinado a partir de análise trigonométrica.

$$\alpha = \arctan\left(\frac{L_3 \text{sen } \theta_3}{L_2 + L_3 \text{cos } \theta_3}\right) \quad (3.32)$$

O ângulo β é dado pela Equação (3.33).

$$\beta = \arctan\left(\frac{(p_{z3} - (L_0 + L_1))}{\pm \sqrt{p_{x3}^2 + p_{y3}^2}}\right) \quad (3.33)$$

Finalmente, o ângulo θ_2 é definido por meio da diferença entre os ângulos auxiliares como mostrado na Equação (3.34).

$$\theta_2 = \beta - \alpha \quad (3.34)$$

Para o cálculo dos ângulos θ_4 e θ_5 é necessário obter a orientação do efetuador a partir da sua matriz de rotação em relação ao punho pelo fato desse ponto ser considerado a base da segunda parte do braço robótico (GEMBAROWSKI e CAMPOS, 2015).

A partir dos ângulos θ_1 , θ_2 e θ_3 é determinada a matriz de transformação homogênea dada pela Equação (3.35), de onde é possível definir a orientação do punho em relação à base do robô por meio das três primeiras linhas com as três primeiras colunas da matriz.

$${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3 = \begin{bmatrix} n_{x3} & s_{x3} & a_{x3} & p_{x3} \\ n_{y3} & s_{y3} & a_{y3} & p_{y3} \\ n_{z3} & s_{z3} & a_{z3} & p_{z3} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.35)$$

A matriz de rotação do punho em relação à base é representada pela Equação (3.36).

$${}^0R_3 = \begin{bmatrix} n_{x3} & s_{x3} & a_{x3} \\ n_{y3} & s_{y3} & a_{y3} \\ n_{z3} & s_{z3} & a_{z3} \end{bmatrix} \quad (3.36)$$

Onde:

$$\begin{aligned} n_{x3} &= \cos \theta_1 \cdot \cos \theta_2 \cdot \cos \theta_3 - \cos \theta_1 \cdot \sin \theta_2 \cdot \sin \theta_3 \\ n_{y3} &= \sin \theta_1 \cdot \cos \theta_2 \cdot \cos \theta_3 - \sin \theta_1 \cdot \sin \theta_2 \cdot \sin \theta_3 \\ n_{z3} &= -\sin \theta_2 \cdot \cos \theta_3 - \cos \theta_2 \cdot \sin \theta_3 \\ s_{x3} &= -\cos \theta_1 \cdot \sin \theta_2 \cdot \sin \theta_3 - \cos \theta_1 \cdot \sin \theta_2 \cdot \cos \theta_3 \\ s_{y3} &= -\sin \theta_1 \cdot \sin \theta_2 \cdot \sin \theta_3 - \sin \theta_1 \cdot \sin \theta_2 \cdot \cos \theta_3 \\ s_{z3} &= -\sin \theta_2 \cdot \sin \theta_3 - \cos \theta_2 \cdot \cos \theta_3 \\ a_{x3} &= \sin \theta_1 \\ a_{y3} &= \cos \theta_1 \\ a_{z3} &= 0 \end{aligned}$$

A matriz de rotação do efetuador em relação à base pode ser calculada conforme mostra a Equação (3.37).

$${}^0R_5 = {}^0R_1 {}^1R_2 {}^2R_3 {}^3R_4 {}^4R_5 = {}^0R_3 {}^3R_5 \quad (3.37)$$

Pelas propriedades de pré e pós multiplicação é válido obter a matriz de rotação do efetuador em relação à base a partir da relação definida pela Equação (3.38).

$${}^3R_5 = ({}^0R_3)^{-1} \cdot {}^0R_5 = \begin{bmatrix} n_{x5} & s_{x5} & a_{x5} \\ n_{y5} & s_{y5} & a_{y5} \\ n_{z5} & s_{z5} & a_{z5} \end{bmatrix} \quad (3.38)$$

Sendo:

$$\begin{aligned} n_{x5} &= \cos \theta_4 \cdot \cos \theta_5 - \sin \theta_4 \\ n_{y5} &= \sin \theta_4 \cdot \cos \theta_5 - \cos \theta_4 \\ n_{z5} &= -\sin \theta_5 \\ s_{x5} &= -\cos \theta_4 \cdot \cos \theta_5 - \sin \theta_4 \\ s_{y5} &= -\sin \theta_4 \cdot \cos \theta_5 - \cos \theta_4 \\ s_{z5} &= \sin \theta_5 \end{aligned}$$

$$a_{x5} = \cos \theta_4 \cdot \text{sen } \theta_5$$

$$a_{y5} = \text{sen } \theta_4 \cdot \text{sen } \theta_5$$

$$a_{z5} = \cos \theta_5$$

Com base na Equação (3.38), é possível realizar algumas relações trigonométricas entre os termos da matriz que definem os valores dos ângulos θ_4 e θ_5 como demonstra a Equação (3.39) e a Equação (3.40), respectivamente.

$$\theta_4 = \arctan \left(\frac{a_{y5}}{a_{x5}} \right) \quad (3.39)$$

$$\theta_5 = \arctan \left(\frac{\sqrt{a_{x5}^2 + a_{y5}^2}}{a_{z5}} \right) \quad (3.40)$$

Em resumo, o método selecionado para resolução da cinemática inversa é realizado de acordo com as seguintes etapas (SICILIANO, SCIAVICCO, *et al.*, 2009):

1. Calcular a posição do punho de acordo com a Equação (3.24);
2. Resolver a cinemática inversa para os ângulos θ_1 pela Equação (3.26), θ_2 pela Equação (3.34) e θ_3 pela Equação (3.31);
3. Determinar a orientação do punho em relação à base conforme a Equação (3.36);
4. Definir a orientação do efetuador em relação ao punho dada pela Equação (3.38);
5. Resolver a cinemática inversa da orientação para os ângulos θ_4 pela Equação (3.39) e θ_5 pela Equação (3.40).

3.5 CINEMÁTICA DIFERENCIAL

A cinemática diferencial descreve a relação das velocidades das juntas com os movimentos desenvolvidos pelo efetuador no espaço cartesiano através de uma matriz denominada Jacobiano (SICILIANO, SCIAVICCO, *et al.*, 2009). Assim, os movimentos no espaço cartesiano e no espaço de juntas podem ser mapeados de maneira que as trajetórias possam ser desenvolvidas com restrições como limites temporais ou velocidades desejadas.

Na determinação do Jacobiano é conveniente calcular a velocidade linear separadamente da velocidade angular, procedendo em duas etapas como descrito abaixo (SICILIANO, SCIAVICCO, *et al.*, 2009).

A contribuição para a velocidade linear pode ser definida como a derivada temporal do efetuador do braço robótico, isto é, a variação da posição 0P_n no tempo, considerando que “n” é o último grau de liberdade do robô. A Equação (3.41) expressa essa relação com a inclusão matemática das variações angulares das juntas rotativas.

$${}^0\dot{P}_n = \sum_{i=1}^n \frac{d({}^0P_n)}{d\theta_i} \frac{d\theta_i}{dt} = \sum_{i=1}^n \frac{d({}^0P_n)}{d\theta_i} \dot{\theta}_i \quad (3.41)$$

Logo, é possível concluir que a parte matriz jacobiana que define a velocidade linear do efetuador pode ser extraída da Equação (3.41), resultando na expressão mostrada na Equação (3.42).

$$J_P = \sum_{i=1}^n \frac{d({}^0P_n)}{d\theta_i} \quad (3.42)$$

Para juntas rotativas a velocidade linear deve ser calculada em relação à origem do sistema de coordenadas do efetuador. Dessa maneira, a contribuição de cada junta “i” no Jacobiano é dado pelo produto vetorial da orientação dos eixos z_i com o braço de alavanca que corresponde a distância entre a origem do *frame* do efetuador e a origem da junta correspondente. Assim, cada junta θ_i contribui com a velocidade linear do efetuador como mostra a Equação (3.43).

$$J_{P_i} = \hat{z}_{i-1} \times ({}^0P_n - {}^0P_{i-1}) \quad (3.43)$$

Considerando os ângulos RPY, a velocidade angular do efetuador pode ser resumida como demonstra a Equação (3.44).

$$\omega = \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3.44)$$

Dessa forma, para a contribuição da velocidade angular do efetuador é trivial definir a Equação (3.45), já que as variações angulares na junta “i” ocorrem no eixo “i-1” definido segundo os parâmetros de *Denavit-Hartenberg*.

$$\omega_{i-1,i} = \dot{\theta}_i \hat{z}_{i-1} \quad (3.45)$$

Então para se obter a parte da matriz jacobiana que define a velocidade angular do efetuador a partir de juntas rotativas basta desconsiderar a variação angular da junta da Equação (3.45) como é representado na Equação (3.46).

$$J_{O_i} = \hat{z}_{i-1} \quad (3.46)$$

Em resumo, a matriz jacobiana para juntas rotativas é obtida através da Equação (3.47).

$$J = \begin{bmatrix} J_{P_1} & \cdots & J_{P_n} \\ \vdots & \ddots & \vdots \\ J_{O_1} & \cdots & J_{O_n} \end{bmatrix} = \begin{bmatrix} J_{P_i} \\ J_{O_i} \end{bmatrix} = \begin{bmatrix} \hat{z}_{i-1} \times ({}^0P_n - {}^0P_{i-1}) \\ \hat{z}_{i-1} \end{bmatrix} \quad (3.47)$$

Para a determinação da velocidade da garra do braço robótico estudado neste trabalho é realizada a cinemática diferencial direta, ou seja, a partir do deslocamento das juntas se obtém o valor do deslocamento total no efetuador. A solução do Jacobiano no projeto foi definida pelo método geométrico considerando apenas as velocidades lineares nas três primeiras juntas do robô, visto que as velocidades angulares podem ser desprezadas a fim de melhorar o desempenho do controlador.

Assim, pelo fato de todas as juntas serem rotativas, os termos da matriz Jacobiana foram calculados a partir do produto vetorial da orientação dos eixos z_i das coordenadas das juntas “i”, que pode ser obtida pela terceira coluna da matriz 0R_n , com o braço de alavanca dado pela diferença da posição do efetuador em relação ao ponto onde está a junta como mostrado na Equação (3.48).

$$J = \begin{bmatrix} \hat{z}_0 \times {}^0P_5 \\ \hat{z}_1 \times ({}^0P_5 - {}^0P_1) \\ \hat{z}_2 \times ({}^0P_5 - {}^0P_2) \end{bmatrix} \quad (3.48)$$

Dessa maneira é possível determinar as velocidades lineares do efetuador no espaço cartesiano em função das velocidades das juntas como pode ser observado na Equação (3.49).

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (3.49)$$

Assim como existe a cinemática inversa em relação à cinemática direta de um robô, também é possível obter cinemática diferencial inversa para o caso em que se deseja determinar o valor das velocidades das juntas em relação à uma dada velocidade no efetuador. Para o caso do projeto proposto, o Jacobiano é uma matriz quadrada, bastando invertê-la para obter esta relação como demonstra a Equação (3.50). Se o Jacobiano não fosse uma matriz quadrada seria necessário realizar outros métodos para solução da cinemática diferencial inversa como, por exemplo, a matriz pseudo-inversa (GEMBAROWSKI e CAMPOS, 2015).

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = J^{-1} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (3.50)$$

4 IMPLEMENTAÇÃO DO PROJETO

O desenvolvimento do controlador abordado neste trabalho levou em consideração aspectos que o tornassem um projeto de baixo custo, aberto aos estudantes e pesquisadores, com fácil aplicação de novos recursos e flexibilidade para alterações na programação.

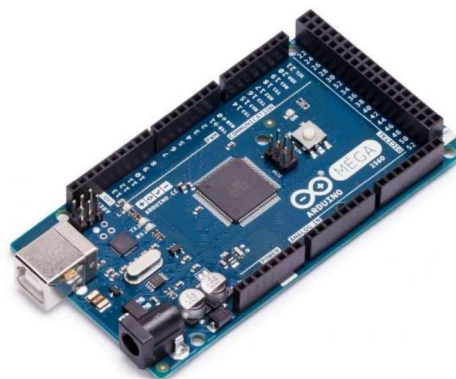
O protótipo montado possibilita que sejam realizados ajustes e testes de maneira dinâmica a partir de uma plataforma amigável, cuja construção foi montada de maneira simples a partir de módulos dedicados. Assim, a intenção de criar um projeto colaborativo é evidenciada, visto que atualizações e criação de outras versões de controlador podem ser realizadas a partir da análise deste trabalho.

Outra característica importante no projeto é a finalidade didática, já que o braço robótico pode ser utilizado em aulas práticas no ensino da robótica, bem como em experimentos em outras áreas do conhecimento como a teoria de controle.

4.1 PLACA CONTROLADORA

O controle do robô é realizado por meio da placa Arduino Mega 2560 que possui elementos que facilitam a prototipagem do projeto com base no microcontrolador ATmega2560. A placa possui 54 pinos de entrada/saída digitais, 16 pinos de entrada analógicos, um oscilador de cristal de 16 MHz, entre outros recursos (ARDUINO, 2017). A Figura 18 apresenta a visão geral da placa utilizada.

Figura 18 – Placa Arduino Mega 2560 Rev3



Fonte: (ARDUINO, 2017).

A programação da placa Arduino Mega 2560 é realizada por meio do *software* Arduino IDE, cuja transferência para o ATmega2560 ocorre através de uma interface

USB que funciona com auxílio do microcontrolador ATmega16U2, responsável pela conversão dos dados trocados com um canal serial (ARDUINO, 2017). Além disso, a interface USB ainda possibilita a alimentação da placa com correntes elétricas de até 500mA quando não há outra fonte de energia.

A alimentação externa é realizada por meio de um conector tipo *jack* que deve receber valores de tensão elétrica entre 6V e 20V, posteriormente regulados pelo CI NCP1117. É importante destacar alguns pinos dedicados à alimentação elétrica na placa Arduino Mega 2560 com níveis de terra, 3,3V e 5V, além de um pino denominado “Vin” que fornece a mesma tensão proveniente da fonte de alimentação externa.

No projeto deste trabalho, os pinos configurados como entrada recebem sinais dos *encoders* e chaves fim de curso para que, a partir de comandos do usuário, o microcontrolador defina o envio de sinais para os motores com base na programação gravada para a operação do braço robótico.

Os motores elétricos são acionados por um circuito ponte H que recebe sinais dos pinos definidos como saída do microcontrolador, determinando o sentido de rotação de cada eixo. A velocidade angular de um motor de corrente contínua é proporcional à tensão de alimentação aplicada em seus terminais. Assim, o controle de movimentação das juntas do braço robótico deste projeto é obtido por sinais PWM que definem a tensão média aplicada nos terminais de cada motor. O microcontrolador ATmega2560 possui 15 terminais digitais que podem ser usados como saídas PWM, enumerados de 2 a 13 e 44 a 46 na placa (ARDUINO, 2017).

Os *encoders* possuem dois fototransistores que atuam de maneira conjunta a fim de determinar a quantidade de pulsos emitidos, bem como a direção de rotação do disco acoplado ao eixo dos motores. Tal ação realiza a realimentação do estado das juntas para o controlador, sendo fundamental para o controle do robô.

Como os dados captados pelos *encoders* devem ser enviados de maneira instantânea para o microcontrolador, a detecção dos pulsos é realizada por meio de um recurso denominada interrupção externa, capaz de executar uma tarefa prioritária na programação. Assim, quando um dos fototransistores é acionado, o controlador realiza um incremento ou decremento na variável que armazena a quantidade de pulsos do *encoder* correspondente independente do estado de funcionamento do braço robótico. O microcontrolador ATmega2560 possui 6 terminais digitais que podem ser usados como interrupções externas, identificados

pelos pinos 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3) e 21 (interrupt 2) (ARDUINO, 2017).

A posição de referência do robô é definida com base nos sinais enviados pelas chaves fim de curso a partir da movimentação programada para iniciar cada junta. Sempre que o braço robótico for operado, é necessário que o usuário comande o robô para atingir o seu estado referencial. O programa do microcontrolador deve prever uma rotina de movimentos que precisam ser realizados de forma que, quando todos os pinos conectados às chaves fim de curso não estiverem recebendo sinais, o controlador seja referenciado e os valores iniciais dos *encoders* sejam atribuídos.

A placa Arduino Mega 2560 ainda possui 4 UARTs, cada um composto por um par de pinos com funções de receptor ou transmissor para comunicação serial. Em um canal UART é realizado a troca de informações com módulo responsável pela interface com o usuário.

A Tabela 6 relaciona a disposição dos pinos da placa utilizados com as suas respectivas funções.

Tabela 6 – Funções dos pinos da placa Arduino Mega 2560

Pino	Tipo	Função
22	Saída digital	Entrada 1 da ponte H do motor 1
23	Saída digital	Entrada 2 da ponte H do motor 1
24	Saída digital	Entrada 3 da ponte H do motor 2
25	Saída digital	Entrada 4 da ponte H do motor 2
26	Saída digital	Entrada 1 da ponte H do motor 3
27	Saída digital	Entrada 2 da ponte H do motor 3
28	Saída digital	Entrada 3 da ponte H do motor 4
29	Saída digital	Entrada 4 da ponte H do motor 4
30	Saída digital	Entrada 1 da ponte H do motor 5
31	Saída digital	Entrada 2 da ponte H do motor 5
32	Saída digital	Entrada 3 da ponte H do motor 6
33	Saída digital	Entrada 4 da ponte H do motor 6
8	Saída PWM	Habilitação A da ponte H do motor 1
9	Saída PWM	Habilitação B da ponte H do motor 2
10	Saída PWM	Habilitação A da ponte H do motor 3
11	Saída PWM	Habilitação B da ponte H do motor 4
12	Saída PWM	Habilitação A da ponte H do motor 5

13	Saída PWM	Habilitação B da ponte H do motor 6
48	Entrada digital	Sinal da chave fim de curso 1
50	Entrada digital	Sinal da chave fim de curso 2
51	Entrada digital	Sinal da chave fim de curso 3
52	Entrada digital	Sinal da chave fim de curso 4
53	Entrada digital	Sinal da chave fim de curso 5
2	Entrada digital	Sinal do fototransistor P0 do <i>encoder</i> 1
17	Entrada digital	Sinal do fototransistor P1 do <i>encoder</i> 1
3	Entrada digital	Sinal do fototransistor P0 do <i>encoder</i> 2
16	Entrada digital	Sinal do fototransistor P1 do <i>encoder</i> 2
21	Entrada digital	Sinal do fototransistor P0 do <i>encoder</i> 3
4	Entrada digital	Sinal do fototransistor P1 do <i>encoder</i> 3
20	Entrada digital	Sinal do fototransistor P0 do <i>encoder</i> 4
5	Entrada digital	Sinal do fototransistor P1 do <i>encoder</i> 4
19	Entrada digital	Sinal do fototransistor P0 do <i>encoder</i> 5
6	Entrada digital	Sinal do fototransistor P1 do <i>encoder</i> 5
18	Entrada digital	Sinal do fototransistor P0 do <i>encoder</i> 6
7	Entrada digital	Sinal do fototransistor P1 do <i>encoder</i> 6
15	Comunicação	Receptor da interface com o usuário
14	Comunicação	Transmissor da interface com o usuário
Vin	Alimentação	Fonte de alimentação externa
5V	Alimentação	Alimentação dos LED's dos <i>encoders</i> e auxílio nos sinais de entrada
3,3V	Alimentação	Alimentação do módulo de comunicação <i>wireless</i>
GND	Alimentação	Terra

Fonte: produção do próprio autor.

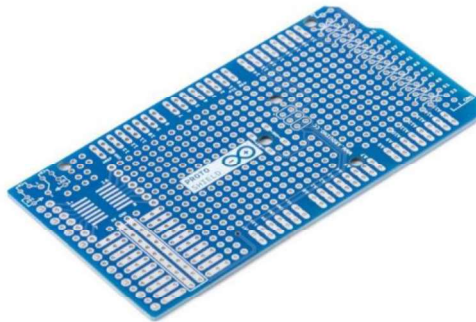
4.2 PLACA DE INTEGRAÇÃO DE COMPONENTES

O *protoshield* é uma placa ilhada desenvolvida para facilitar o projeto de circuitos personalizados através da prototipagem no padrão do Arduino Mega. Os componentes necessários para o funcionamento do projeto podem ser soldados diretamente no *protoshield* e ligados por trilhas até serem conectados diretamente nos pinos do microcontrolador, reduzindo as dimensões do circuito e evitando falhas durante o seu funcionamento. Dessa maneira é possível inserir elementos auxiliares no funcionamento do controlador por meio de barramentos que se encaixam

individualmente nos pinos da placa Arduino Mega 2560, tornando prática a substituição ou inclusão de componentes no projeto.

A Figura 19 apresenta o Arduino Mega Protoshield onde se pode notar suas características ideais para prototipagem, como as ilhas para soldagem de componentes, impressão da identificação dos pinos do microcontrolador e barramentos para alimentação.

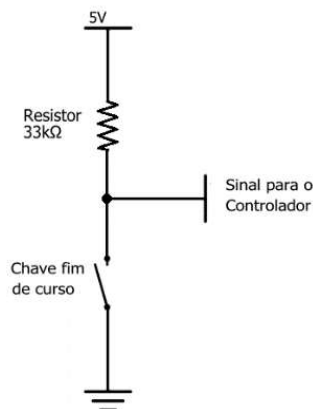
Figura 19 – Arduino Mega Protoshield



Fonte: (ARDUINO, 2017).

Neste trabalho o *protoshield* foi utilizado para conectar o cabeamento do braço robótico no controlador, tratar os sinais de entrada, acionar os LEDs dos *encoders* e ligar o módulo de comunicação *wireless* ao microcontrolador. A detecção do acionamento das chaves fim de curso é realizada através de um divisor de tensão disposto de maneira que cada pino de entrada de sinal do microcontrolador fique entre um resistor *pull-up* ligado à tensão 5V e uma chave fim de curso ligada ao terra, conforme esquematiza a Figura 20.

Figura 20 – Divisor de tensão para a chave fim de curso

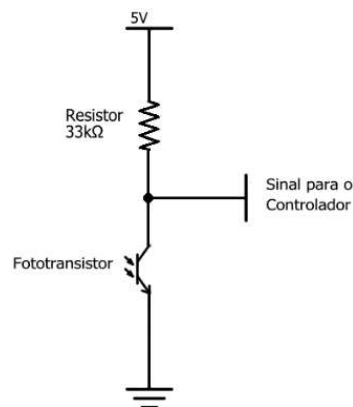


Fonte: produção do próprio autor.

O comportamento da chave fim de curso será de um resistor com resistência infinita quando estiver aberta, fazendo com que o sinal enviado para o controlador seja de nível alto. Em contrapartida, se houver o chaveamento, o pino de entrada ficará diretamente ligado ao terra, acarretando em um nível lógico baixo na entrada do microcontrolador.

O tratamento do sinal captado dos fototransistores ocorre de forma análoga. No entanto, quando o fototransistor é sensibilizado por uma intensidade luminosa proveniente do LED do *encoder* seu comportamento é de um resistor com resistência infinita, caso contrário funcionará como uma chave fechada. A Figura 21 demonstra a ligação realizada para este caso.

Figura 21 – Divisor de tensão para o fototransistor



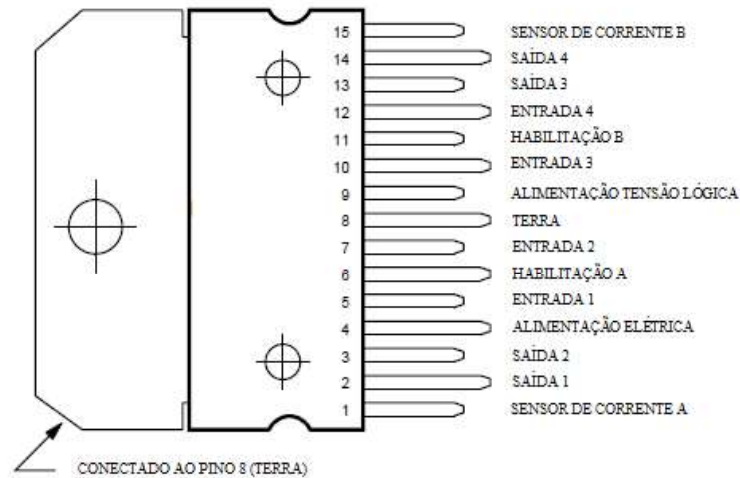
Fonte: produção do próprio autor.

4.3 DRIVERS DOS MOTORES

O acionamento dos motores do braço robótico é realizado por meio de um *driver* L298N que possui dois circuitos ponte H projetados para acionar cargas indutivas como relés, solenoides, motores de corrente contínua e motores de passo. O circuito integrado pode operar em tensões elétricas entre 5V e 35V suportando correntes de no máximo 2A por ponte H, sendo que sua tensão lógica é de 5V com uma corrente de até 36mA (STMICROELECTRONICS, 2000).

Na Figura 22 está ilustrado o circuito integrado L298N com a numeração e as funções de cada pino

Figura 22 – Circuito integrado L298N

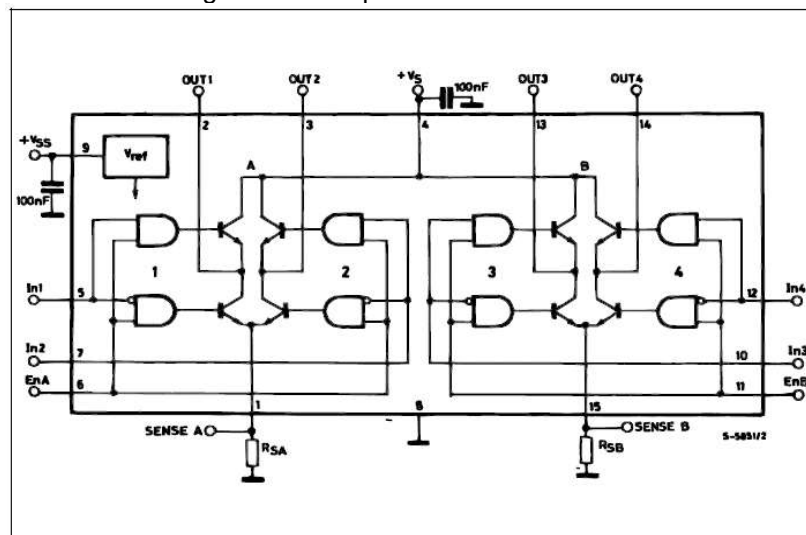


Fonte: (STMICROELECTRONICS, 2000).

A Figura 23 apresenta o esquema geral do circuito, onde é possível notar as duas pontes H denominadas A e B. Cada par de chaves opostas é identificada por um número de 1 a 4, que corresponde ao lado da ponte que é controlada. O terminal de habilitação (En) determina a ativação da ponte H e as entradas (In) definem quais chaves estarão fechadas. As saídas (OUT) fornecem a tensão elétrica de acordo com a comutação da ponte, sendo que as polaridades se alternam de acordo com a maneira que as chaves são ativadas.

Dessa maneira é possível definir o sentido de rotação de cada motor no braço robótico a partir dos sinais enviados para as entradas do CI L298N, visto que esses pinos são responsáveis por definir a polaridade nas saídas que alimentam os terminais do motor.

Figura 23 – Esquema do circuito L298N



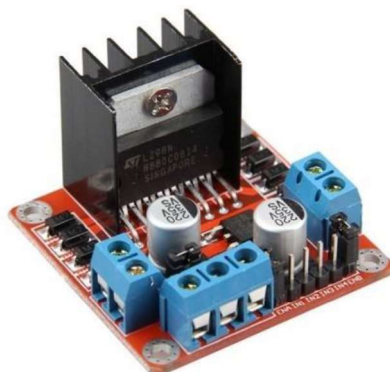
Fonte: (STMICROELECTRONICS, 2000).

Um detalhe importante é que se as chaves superiores da ponte H estiverem fechadas simultaneamente, o motor CC terá o comportamento de um gerador. Assim, caso o eixo do motor esteja em movimento, a energia produzida pela inércia será dissipada de maneira que o motor pare de maneira mais rápida. Logo, essa situação pode auxiliar a frenagem dos motores quando as juntas estiverem em uma posição desejada. Se as chaves inferiores estiverem fechadas simultaneamente, os terminais do motor estarão conectados ao terra, gerando uma tensão nula nas saídas da ponte.

Outro recurso para o controle do motor através da ponte H é o ajuste da velocidade de acordo com a frequência de comutação das chaves do circuito. Isso se deve ao fato de que as habilitações das pontes são definidas por sinais PWM que podem variar sua razão cíclica e, conseqüentemente, controlar a comutação das chaves. Com isso, o motor é alimentado com níveis altos e baixos de tensão em intervalos de tempo fixos. Na prática, a tensão média aplicada no motor será considerada o valor de alimentação dos terminais, definindo a velocidade de rotação do seu eixo.

Os pinos com recurso de PWM na placa Arduino Mega 2560 possuem uma resolução de 8 bits, o que torna possível impor a velocidade mínima e máxima do motor entre valores de 0 a 255. Assim, a tensão aplicada nos motores varia de acordo com a razão cíclica do sinal enviado, sendo que o valor mínimo representa a parada absoluta do motor, cuja razão cíclica possui valor nulo, e a velocidade máxima é definida pela tensão fornecida pela fonte de alimentação elétrica, onde o sinal PWM é proporcional ao valor de 255 e chave da ponte permanece fechada, sem gerar níveis baixos de tensão nas saídas.

Figura 24 – Módulo ponte H L298N



Fonte: (GEEKCREIT, 2017).

No projeto apresentado neste trabalho, o acionamento dos motores é realizado através de três módulos ponte H L298N, visto que cada circuito integrado é capaz de comandar dois motores do braço robótico. Assim, cada par de juntas do robô é comandado pelo controlador com auxílio de um *driver* dedicado, cuja placa é apresentada na Figura 24.

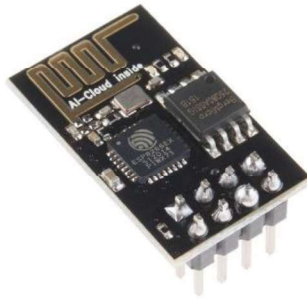
A utilização de módulos dedicados no acionamento dos motores facilita a implementação dos componentes no projeto e reduz os riscos de falha no funcionamento por se tratar de uma placa desenvolvida para este fim, possuindo componentes que otimizam a operação do circuito integrado. Os principais terminais do CI L298N estão ligados a conectores que são identificados da mesma maneira que é representada na Figura 23, onde é possível notar dois bornes tipo KRE com duas vias para cada saída das pontes H, um borne tipo KRE com três vias no qual é realizada a alimentação elétrica com base na tensão nominal dos motores, no caso 12V, e pinos para os sinais de entrada e habilitação enviados pelo microcontrolador.

4.4 MÓDULO DE COMUNICAÇÃO *WIRELESS*

Os comandos do operador enviados para o controlador proposto são realizados através de uma comunicação *wireless* em uma rede conectada a um módulo ESP8266 ESP-01. Com isso, pode-se criar uma interface para o usuário acessar o sistema proposto com flexibilidade de dispositivos na operação do braço robótico e possibilidade de ajustes ou personalizações no controle de comandos.

Entre outros recursos, o CI ESP8266 fornece conexão a redes com padrão 802.11 b/g/n funcionando como uma estação, um ponto de acesso ou ambos por meio de conexões TCP/IP. O circuito ainda possui protocolos de segurança para acesso à rede, conectores GPIO e barramento UART (ESPRESSIF, 2013).

O módulo ESP8266 modelo ESP-01 é apresentado na Figura 25, onde é possível notar a presença de 8 pinos de conexão e uma antena embutida que possibilita um alcance de aproximadamente 90 metros para a rede *wireless*. Cada pino possui uma função específica sendo um para alimentação elétrica, um para sinal de terra, um para recepção e outro para transmissão de comunicação serial, um para habilitar o funcionamento do CI, um para sinal de *reset* e dois para entrada ou saída de uso geral.

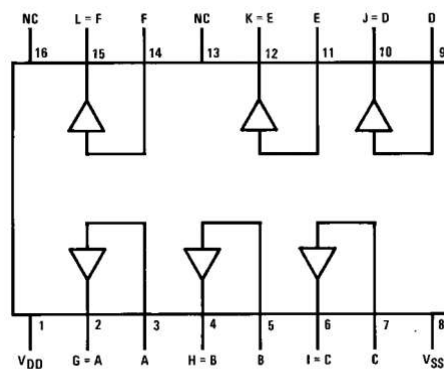
Figura 25 – Módulo *wireless* ESP8266 ESP-01

Fonte: (ESPRESSIF, 2017).

A tensão de operação do módulo ESP8266 ESP-01 é de 3,3V e a alimentação elétrica no projeto é realizada por meio de um pino da placa Arduino Mega 2560 que fornece este nível de tensão. No entanto, a comunicação serial entre o módulo e o microcontrolador deve possuir um tratamento no sinal, visto que o ATmega2560 envia sinais com uma tensão de 5V. A solução para possibilitar a comunicação entre os dois dispositivos é a aplicação de um buffer não-inversor que forneça uma tensão de 3,3V na saída a partir de um sinal 5V enviado pelo microcontrolador.

Para o tratamento do sinal foi utilizado um CI 4050, cujo esquema é mostrado na Figura 26, alimentado a partir do pino 3,3V da placa Arduino Mega 2560. Assim, quando o controlador enviar um sinal para uma das entradas do CI 4050, sua respectiva saída irá enviar um sinal com um nível adequado para o receptor do módulo ESP8266 ESP-01. A transmissão de sinal para o controlador não necessita ser tratada, pois o ATmega2560 é sensibilizado pelo nível de tensão de 3,3V.

Figura 26 – Circuito integrado 4050



Fonte: (ON SEMICONDUCTOR, 2002).

4.5 INTERFACE DE OPERAÇÃO

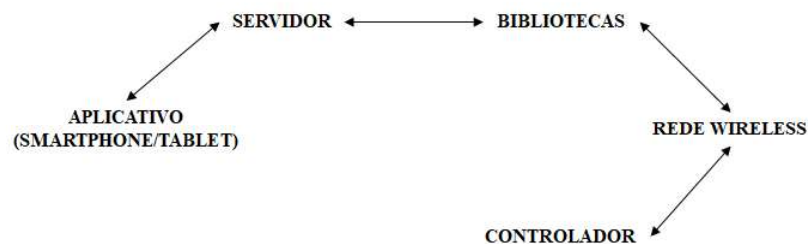
A operação do braço robótico é realizada por meio de qualquer *smartphone* ou *tablet* que tenha acesso ao controlador através do aplicativo *Blynk*, capaz de controlar diretamente a placa Arduino Mega 2560 pela Internet ou em uma rede local. Dessa forma, a interface de comando do robô é flexível quanto aos recursos, comandos e preferências do usuário.

A plataforma *Blynk* possui três componentes principais (BLYNK, 2017):

- Aplicativo: permite criar interfaces para o projeto usando ferramentas próprias;
- Servidor: realiza toda a comunicação entre o controlador e *smartphone/tablet*, podendo ser utilizado o servidor *Blynk* via Internet ou criar um serviço para acesso local;
- Bibliotecas: fornecem soluções desenvolvidas para habilitar a comunicação entre o servidor e controlador.

A Figura 27 demonstra um esquema geral que resume o funcionamento da plataforma *Blynk* no projeto.

Figura 27 – Funcionamento da plataforma *Blynk*



Fonte: produção do próprio autor.

O dispositivo (*smartphone* ou *tablet*) que possui o aplicativo *Blynk* instalado só terá permissão de acesso ao controlador caso possua permissão por meio de um código de segurança alfanumérico (*token*). Neste trabalho, o projeto possui um recurso de compartilhamento onde qualquer dispositivo pode cadastrar as interfaces de controle do braço robótico por meio de um código QR, permitindo que o usuário utilize os comandos apresentados sem poder realizar edições.

A interface de operação do projeto pode ser obtida a partir do escaneamento do código QR da Figura 28 pelo aplicativo *Blynk* por qualquer usuário que possua a

plataforma instalada em seu *smartphone* ou *tablet*. Através da tela do seu dispositivo, o operador poderá referenciar as juntas do braço robótico, controlar a garra do robô nos eixos cartesianos, movimentar as juntas individualmente, gravar pontos no espaço, executar trajetórias programadas e acompanhar o vídeo do robô em tempo real pela mesma interface. Para isso é necessário que o controlador esteja conectado à Internet ou em uma rede *wireless* local, bem como o dispositivo do operador.

Pela interface completa o usuário terá acesso a todos os recursos citados anteriormente através de abas que organizam os comandos por categorias.

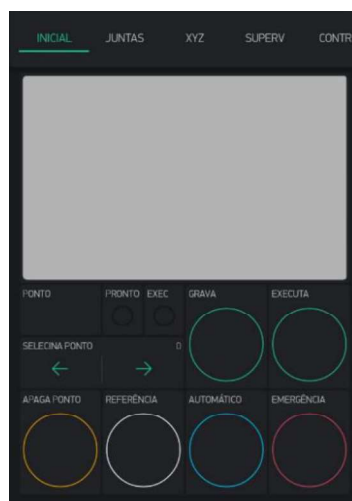
Figura 28 – Código QR para interface de operação



Fonte: produção do próprio autor.

A tela inicial da interface de operação acessada pelo código QR acima está ilustrada na Figura 29.

Figura 29 – Aba inicial da interface de operação



Fonte: produção do próprio autor.

O usuário do sistema poderá visualizar o braço robótico em tempo real nas três primeiras abas da interface através de uma janela de vídeo. Se preferir, o

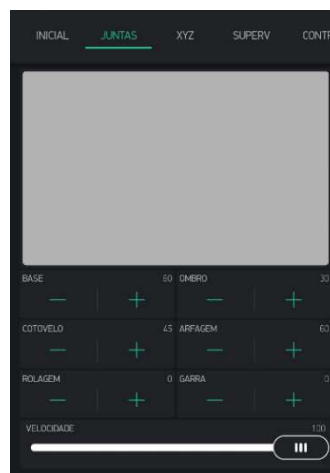
operador poderá interromper ou retomar a visualização em tempo real através de toques nessa janela.

Na aba “INICIAL” o operador poderá verificar o estado do robô por meio dos sinalizadores “PRONTO”, que indica que o sistema está pronto para operar após o referenciamento realizado pelo botão “REFERÊNCIA”, e “EXEC”, que sinaliza a execução de uma tarefa. Através do botão “GRAVA” é possível armazenar um ponto dado pela postura atual do robô. Além disso, o usuário poderá navegar entre os pontos salvos por meio dos botões de “SELECIONA PONTO”, onde o ponto selecionado é mostrado no indicador “PONTO”. Se desejar, o operador poderá apagar o ponto selecionado através do botão “APAGA PONTO”.

Na aba “JUNTAS” é realizado o controle individual das juntas, onde os botões “BASE”, “OMBRO”, “COTOVELO”, “ARFAGEM” e “ROLAGEM” posicionam as juntas de mesmo nome do braço robótico entre os ângulos mínimo e máximo de acordo com as suas limitações físicas com uma resolução de 1°. Os botões de “GARRA”, por sua vez, determina o valor de abertura da garra em porcentagem. É possível, ainda, definir a porcentagem da velocidade de movimentação através do botão deslizante “VELOCIDADE”.

A Figura 30 ilustra a tela de controle das juntas da interface de operação.

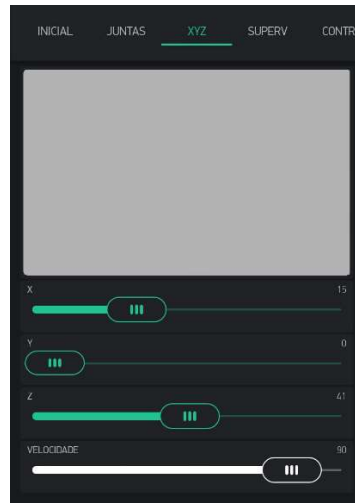
Figura 30 – Aba de controle das juntas da interface



Fonte: produção do próprio autor.

No caso da aba “XYZ”, mostrada na Figura 31, é possível realizar o controle do posicionamento da garra nos eixos cartesianos pelos botões deslizantes “X”, “Y” e “Z”, além de definir a porcentagem da sua velocidade de movimentação por meio de “VELOCIDADE”.

Figura 31 – Aba de controle do efetuator da interface



Fonte: produção do próprio autor.

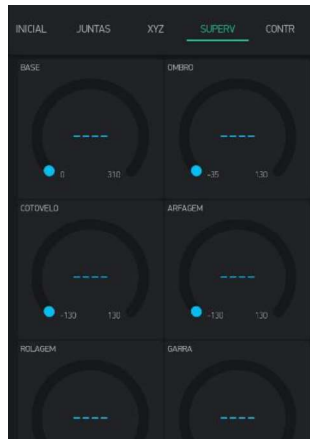
A aba “CONTR”, mostrada na Figura 32, oferece a possibilidade de o operador selecionar as estratégias de controle por meio dos botões “P”, “PI”, “PD” e “PID”, indicadas pelos sinalizadores de mesmo nome. Ainda é possível realizar modificações nos parâmetros de controle do braço robótico através da variação manual dos botões “PARÂMETRO” e “AJUSTE” que variam os valores indicados em “P”, “I” e “D” que correspondem ao valor dos ganhos das componentes proporcional, integral e derivada, respectivamente. Os valores iniciais destas variáveis foram definidos por um ajuste pelo método de *Ziegler-Nichols*, mas o projeto apresenta este recurso de modificação para fins de experimentação e análise didática, permitindo que o usuário proponha ganhos diferentes para cada situação.

Figura 32 – Aba de controle PID da interface



Fonte: produção do próprio autor.

Figura 33 – Aba de supervisão da interface



Fonte: produção do próprio autor.

Por fim, a aba “SUPERV”, demonstrada na Figura 33, tem a função de verificar o ângulo atual de cada junta do braço robótico.

Para que o aplicativo possa trocar informações com o controlador, é necessário que haja um servidor a fim de realizar a comunicações entre os dispositivos, bem como bibliotecas específicas que garantem um bom funcionamento do projeto a partir de soluções desenvolvidas para os módulos de comunicação e controle utilizados (BLYNK, 2017).

A plataforma *Blynk* oferece um servidor que pode ser acessado via Internet, realizando o acesso ao controlador a partir da autenticação do código de segurança (*token*) gerado para o projeto. Há ainda a possibilidade de criação de um servidor próprio a partir de um computador ou notebook, por exemplo, dispensando a necessidade de acesso à Internet e utilizando uma rede local, mais segura, rápida e estável.

As bibliotecas fornecem um conjunto de comandos e funções que realizam ações a partir de um código já desenvolvido, simplificando o desenvolvimento da programação. No caso das bibliotecas *Blynk* utilizadas neste trabalho, existem soluções que permitem a comunicação entre o aplicativo instalado no *smartphone* ou *tablet* e a placa Arduino Mega 2560 por meio do módulo ESP8266 ESP-01, entre outros recursos para troca de dados.

4.6 PROTÓTIPO FINAL

O protótipo do controlador proposto foi montado com base nos componentes apresentados anteriormente, conforme mostra a Figura 34, no qual a disposição das

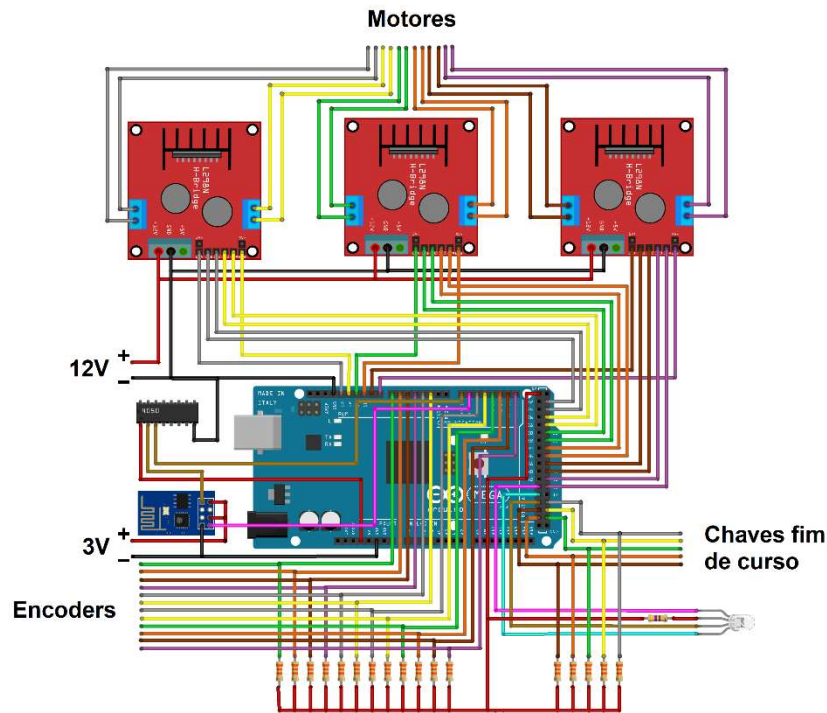
conexões e a relação entre os elementos são definidas de acordo com as suas funcionalidades. Para auxiliar na montagem do projeto foi utilizado o *protoshield*, que não está representado no esquema para melhorar a visualização das ligações. O projeto foi acomodado em uma caixa plástica para proteger os dispositivos e outros componentes auxiliares foram utilizados no protótipo a fim de garantir o bom funcionamento e facilitar sua operação.

Na alimentação do projeto foi utilizado um conector *jack* P4 como entrada da tensão 12V, onde o controle de acionamento do protótipo é realizado por meio de uma chave que define o estado de funcionamento. Para garantir a integridade dos dispositivos, um fusível 3A foi instalado como segurança contra eventuais sobrecargas.

Os *drivers* são componentes que aquecem durante a operação do robô, sendo que seus módulos possuem dissipadores. Além disso, a fonte de tensão 3,3V da placa Arduino Mega 2560 não suporta a alimentação do módulo ESP8266 ESP-01, sendo necessário o uso de reguladores para rebaixar a tensão da fonte principal, processo que também aquece o ambiente interno da caixa. Assim, foram instalados dois *coolers* para ventilação e dissipação do calor, mantendo o protótipo em uma faixa de temperatura adequada para seu funcionamento. Um LED RGB foi instalado junto ao controlador a fim de indicar o estado de operação do braço robótico, sendo que a luz vermelha indica que o controlador está energizado, mas ainda não foi conectado à rede *wireless*, a iluminação verde sinaliza que o robô está pronto para ser operado e a luz azul significa que o braço robótico está no modo de operação automático.

Os condutores ilustrados na Figura 34 fazem parte do cabeamento conectado no conector DB-50 conforme a Tabela 4. É possível notar que a placa Arduino Mega 2560 é responsável pelo controle de todos os processos através do recebimento de sinais de entrada enviados pelo braço robótico, por meio dos *encoders* e chaves fim de curso, e pelo operador, a partir dos comandos da interface conectada ao módulo ESP8266 ESP-01, responsável pela comunicação externa do controlador. Com base no programa do microcontrolador, são transmitidos sinais para os *drivers* representados pelos módulos L298N que acionam os motores do robô alimentados por uma fonte externa de 12V.

Figura 34 – Componentes do controlador



Fonte: produção do próprio autor.

A iluminação do LED RGB é controlada diretamente pela placa Arduino Mega 2560 de acordo com a programação gravada. Assim, após a conexão do módulo ESP8266 ESP-01 com a rede *wireless*, a luz vermelha apaga e o LED fica verde, sinalizando que o braço robótico está pronto para ser operado como mostra a Figura 35.

Figura 35 – Braço robótico pronto para operação



Fonte: produção do próprio autor.

Com a movimentação do robô, é possível gravar os pontos desejados para que sejam realizadas tarefas definidas. Assim, após salvar as ações desejadas para o braço robótico, o usuário pode deixar o controlador em modo automático para execução das tarefas de maneira sequencial. Nesse caso, a luz verde se apaga e o LED fica azul.

A **Erro! Fonte de referência não encontrada.** apresenta a lateral do controlador por onde é realizada a alimentação elétrica do protótipo que pode ser chaveada para definir o estado de acionamento do controlador. O porta-fusível fica disposto logo abaixo e pode ser acessado facilmente caso haja necessidade de troca. Outro componente verificado é o *cooler* utilizado para realizar a ventilação da caixa onde está acomodado o projeto, resfriando os elementos internos da solução.

Figura 36 – Alimentação elétrica do controlador



Fonte: produção do próprio autor.

O LED RGB responsável por sinalizar os estados de operação do controlador é mostrado em detalhe na Figura 37. Também é apresentado o *cooler* usado na exaustão do ar interno do protótipo, otimizando o processo de dissipação do calor nos dispositivos e garantindo a integridade dos componentes.

Figura 37 – Sinalização do controlador



Fonte: produção do próprio autor.

Para efeito de análise do projeto apresentado, a Figura 38 compara a solução proposta com o controlador original do braço robótico. É evidente a diferença entre as dimensões dos controladores, bem como o peso, cujos dados são apresentados na Tabela 7.

Tabela 7 – Dados comparativos entre os controladores

Controlador	Peso	Comprimento	Largura	Altura
Original	9,0kg	31cm	23cm	11cm
Protótipo	0,5kg	19cm	11cm	5cm

Fonte: produção do próprio autor.

Outro fato interessante é a comparação entre as interfaces com o usuário de cada sistema, na qual o controlador original utiliza um dispositivo dedicado, mostrado à frente na Figura 38, e o projeto proposto pode adaptar como interface de operação qualquer smartphone ou tablet que possua o aplicativo Blynk e o código de acesso ao dispositivo.

Figura 38 – Comparação entre os controladores



Fonte: produção do próprio autor.

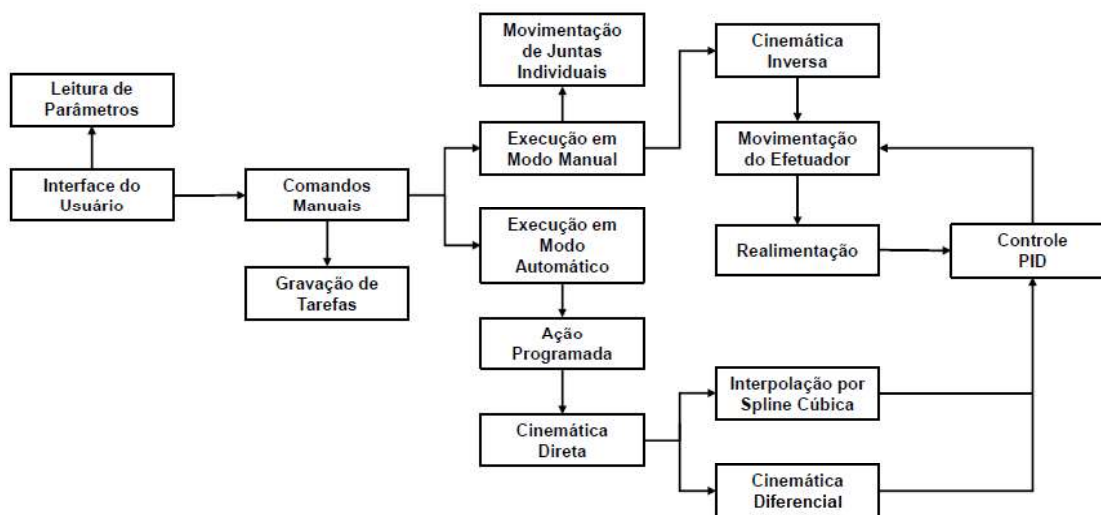
Quanto às funcionalidades, os dois controladores possuem recursos semelhantes, com a exceção de que o projeto proposto não apresenta conexão com dispositivos externos de entrada e saída. A versatilidade na operação e a possibilidade de acesso remoto ao sistema são as vantagens que mais se destacam no projeto proposto.

5 PROGRAMAÇÃO DO CONTROLADOR

A programação do controlador foi realizada através do *software* Arduino IDE, também referido a partir da sigla em inglês como Ambiente de Desenvolvimento Integrado, no qual é possível programar as ações e rotinas do projeto com uma linguagem de alto nível baseada em C/C++.

O funcionamento do projeto proposto pode ser resumido através de um esquema que representa de uma maneira geral as funções e o fluxo do processo como mostra a Figura 39.

Figura 39 – Funcionamento do projeto



Fonte: produção do próprio autor.

O programa da solução proposta pode ser acessado pela *Internet* através do endereço <https://goo.gl/MUSJqb>, que possui o código do projeto aberto, onde se pode verificar a aplicação de todas as funções e algoritmos integrados entre os componentes do projeto e o *software* de programação junto às bibliotecas auxiliares.

A partir do que foi exposto, o programa foi desenvolvido a fim de executar as tarefas definidas a partir de comandos específicos dados pelo usuário do controlador ou pela realimentação do sistema.

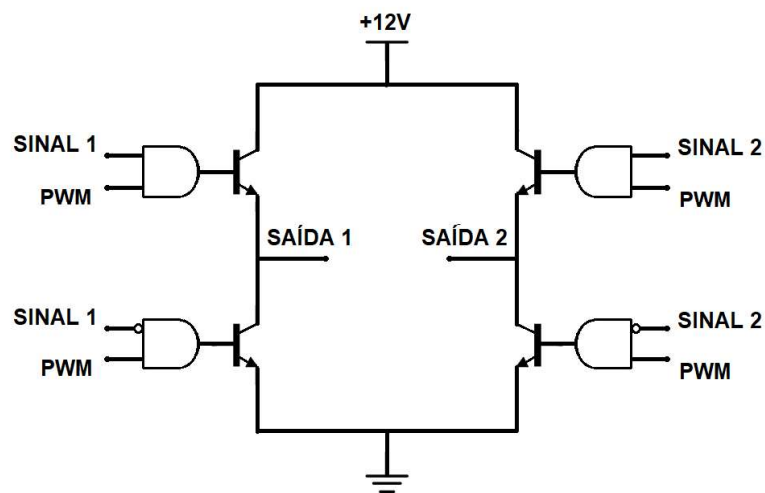
A configuração de cada componente da solução, a integração entre os elementos de comando e acionamento dos atuadores, bem como as estratégias desenvolvidas a fim de adaptar as rotinas de processo no Arduino IDE para cada uma dessas ações são detalhadas nos tópicos a seguir.

5.1 ACIONAMENTO DOS MOTORES

O controle dos motores de corrente contínua de cada junta do braço robótico se baseia no funcionamento das pontes H dos módulos L298N como pode ser visualizado na Figura 40 que representa o detalhamento do circuito de acionamento de cada motor a partir de sinais enviados pelo controlador.

O esquema abaixo demonstra que o *driver* é ativado a partir de dois sinais digitais e um sinal PWM que são responsáveis pelo chaveamento dos transistores que definem o fluxo da corrente elétrica no circuito.

Figura 40 – Esquema do circuito ponte H



Fonte: produção do próprio autor.

O estado de cada sinal digital determina quais chaves da ponte H se fecharão a partir do sinal de ativação dado pelo sinal PWM. É importante notar na Figura 40 que os transistores superiores e inferiores de um mesmo lado nunca serão acionados simultaneamente de modo a evitar um curto-circuito.

O valor da razão cíclica do sinal PWM define a frequência de comutação das chaves o que, na prática, definirá a velocidade da junta pelo controle do valor de tensão média aplicada. Um detalhe que deve ser levado em consideração é que o torque necessário para rotacionar cada junta está relacionado com uma tensão mínima para que cada motor realize a movimentação desejada. Assim, a proporção de valores utilizados na razão cíclica, que pode variar em uma faixa de 0 a 255, em relação a parcela de tensão aplicada no motor deve ser considerada em cada caso de modo que defina um valor inicial suficiente para que o deslocamento da junta ocorra.

Assim, na programação do controlador, cada porta da placa Arduino Mega 2560 deve ser configurada conforme definido na Tabela 6 para que seja possível realizar as funções de acionamento das juntas do robô. Os terminais de cada motor são conectados às saídas representadas na Figura 34, sendo possível se obter quatro estados de funcionamento baseados na tabela verdade mostrada na Tabela 8.

Tabela 8 – Estados de funcionamento dos motores

Sinal 1	Sinal 2	Função
0	0	Neutro
0	1	Anti-horário
1	0	Horário
1	1	Freio

Fonte: produção do próprio autor.

Dessa maneira o programa possui três funções para ativação de estados dos motores:

1. Acionamento: realizado a partir da identificação da junta desejada, o sentido de rotação dado pelo valor dos sinais digitais e a velocidade desejada proporcional ao sinal PWM aplicado;
2. Freio: função que para o motor da junta desejada por meio do envio simultâneo de todos os sinais em nível alto para o *driver*;
3. Neutro: conecta os terminais do motor ao terra, de modo que não gere corrente reversa no circuito caso suas juntas se movam por agentes externos. Tal função é útil caso se deseje movimentar o braço robótico manualmente, sem a ação dos *drivers*.

5.2 REALIMENTAÇÃO PELOS ENCODERS

Uma função indispensável para o controle do braço robótico se dá pela realimentação dos *encoders* acoplados em cada motor que são responsáveis por indicar a posição angular de cada junta e, conseqüentemente, a posição do efetuador. Além disso, existe um *encoder* dedicado a informar o grau de abertura da garra do robô.

A solução definida para captar os dados de movimentação de forma precisa ao mesmo tempo que outras funções são executadas se dá pelo uso do recurso de

interrupções da placa Arduino Mega 2560. Dessa maneira, é possível priorizar as variações de sinais nos *encoders* para que os pulsos que determinam o valor da velocidade e do deslocamento angular sejam contabilizados sem prejudicar outras tarefas, visto que após a ação de realimentação do sistema o programa retorna imediatamente para o mesmo processo que foi interrompido.

Para a solução foram definidas as conexões dos fototransistores conforme a Tabela 6, onde P1 é responsável por realizar as interrupções no programa a partir de disparos correspondentes a mudança de estado, tanto para borda de subida quanto de descida. O fototransistor P0 tem a função de definir qual o sentido de rotação do motor a partir da comparação do seu estado com P1. Desse modo, o contador do *encoder* no programa receberá um incremento ou decremento em seu valor conforme detalha a Tabela 9.

Tabela 9 – Ações dos *encoders*

Fototransistor P1	Fototransistor P0	Ação do <i>Encoder</i>
0	0	Incremento
0	1	Decremento
1	0	Decremento
1	1	Incremento

Fonte: produção do próprio autor.

5.3 POSIÇÃO DE REFERÊNCIA

Sempre que o braço robótico é iniciado, a primeira função que deve ser executada é a de referência das juntas para que os *encoders* sejam calibrados com seus valores iniciais. Assim, nenhuma outra ação pode ser realizada antes que o controlador defina o estado e a posição atual do robô a partir do seu referenciamento.

A função que executa a movimentação para a posição de referência ocorre a partir de um padrão configurado no programa onde cada parte do robô é deslocada até atingir o ponto inicial com base no acionamento das chaves fim de curso e mudança de valores nos contadores dos *encoders*.

Assim, é necessário identificar exatamente quais são os movimentos e sentidos de rotação das juntas necessários para atingir os ângulos de referência de modo que se obtenha uma precisão adequada no posicionamento dos elos no espaço e atribuição dos valores iniciais para os *encoders*. Para isso existe um

elemento projetado especialmente para acionar as chaves de fim de curso que possui um ressalto com a finalidade de se comparar o sinal de referência emitido com o sentido que o motor estava no momento do chaveamento.

O elemento citado é mostrado na Figura 41 e se encontra nas juntas que realizam movimentos verticais em relação ao robô, ou seja, no ombro, no cotovelo e na junta responsável pelo movimento de arfagem do efetuador. Para o posicionamento da base e da garra no movimento de rolagem não é necessário usar tal recurso.

Figura 41 – Elemento auxiliar para o referenciamento



Fonte: produção do próprio autor.

A lógica do posicionamento de referência foi determinada levando em consideração que o braço robótico pode colidir com outros objetos ou entre seus próprios elos durante o processo de movimentação até as chaves fim de curso. Dessa maneira, os movimentos verticais são realizados sendo que primeiramente ocorre o deslocamento do ombro, em seguida o cotovelo e a arfagem do punho. Após isso, a base será deslocada até o ponto de referência seguida do movimento de rolagem, deixando o efetuador na posição inicial, para que a garra seja totalmente aberta. Por último, as juntas responsáveis pelos movimentos verticais são novamente verificadas para garantir que não houve nenhum deslocamento indesejado durante o processo, garantindo precisão no posicionamento inicial do robô.

A Figura 42 ilustra o braço robótico na sua posição inicial definida após o referenciamento das juntas.

Figura 42 – Posição de referência do robô



Fonte: produção do próprio autor.

Na programação, o algoritmo que determina a sequência de movimentos necessários para ativar as chaves fim de curso e definir a posição de referência do robô são detalhados a seguir de acordo com cada junta.

- Base: o motor 1 é acionado em sentido horário até que a chave fim de curso seja ativada, parando a movimentação. Caso o *encoder* pare de variar, significa que a junta chegou no seu limite, sendo necessário inverter a rotação para atingir a chave fim de curso;
- Ombro: Se a chave fim de curso estiver ativada, o motor 2 inicia em sentido horário até que ela seja desativada para, em seguida, ser invertida a rotação e reativá-la, encerrando a movimentação. Caso a chave não estiver acionada no início do processo, o motor 2 se moverá em sentido anti-horário até que ela seja ativada. Se o *encoder* parar de variar, significa que a junta chegou no seu limite, sendo necessário inverter a rotação para atingir a chave fim de curso;

- Cotovelo: Caso a chave fim de curso estiver ativada, o motor 3 inicia em sentido horário até que ela seja desativada para imediatamente ser reativada com a inversão no sentido do motor, encerrando a movimentação. Se a chave não estiver acionada no início do processo, o motor 3 se moverá em sentido anti-horário até que ela seja ativada. Caso o *encoder* parar de variar, significa que a junta chegou no seu limite, sendo necessário inverter a rotação para atingir a chave fim de curso e manter a movimentação até que ela seja desativada e, em seguida, ser invertida a rotação novamente e reativá-la.;
- Arfagem do punho: Se a chave fim de curso estiver ativada, os motores 4 e 5 iniciam em sentido horário até que ela seja desativada para imediatamente ser reativada com a inversão dos motores, encerrando a movimentação. Caso a chave não estiver acionada no início do processo, os motores 4 e 5 se moverão em sentido anti-horário até que ela seja ativada. Caso os *encoders* pararem de variar, significa que a junta chegou no seu limite, sendo necessário inverter a rotação dos motores para atingir a chave fim de curso e manter a movimentação até que ela seja desativada e, em seguida, novamente reativá-la a partir da inversão no sentido dos motores;
- Rolagem do punho: enquanto a chave fim de curso não ser acionada, o motor 3 se moverá em sentido horário, ao mesmo tempo que o motor 4 se movimenta em sentido anti-horário. Após a ativação da chave, os motores invertem a rotação e se movem durante 150 milissegundos e param a fim de alinhar o efetuador;
- Garra: o motor 5 será ativado em sentido horário até seu *encoder* pare de variar, indicando que a garra foi totalmente aberta.

Após o processo de definição do posicionamento de referência, são atribuídos os valores absolutos que os contadores dos *encoders* iniciarão com base nos ângulos referentes a posição definida para configuração dos parâmetros de *Denavit-Hartenberg*. Além disso, é determinada a posição da garra no espaço cartesiano a partir do cálculo da cinemática inversa do braço robótico.

5.4 BIBLIOTECAS AUXILIARES

Os recursos para programação no *software* Arduino IDE podem ser expandidos através do uso de bibliotecas que fornecem funcionalidades extras que podem ser incluídas de acordo com as necessidades do programador (ARDUINO, 2017).

Algumas bibliotecas são instaladas junto ao pacote do *software* Arduino IDE, mas é possível realizar a inclusão de ferramentas independentes desenvolvidas por outros programadores ou até mesmo criar suas próprias bibliotecas.

5.4.1 Bibliotecas do módulo ESP8266

As bibliotecas do módulo ESP8266 são a *ESP8266_Lib* e a *BlynkSimpleShieldEsp8266* e têm como finalidade integrar a placa Arduino Mega 2560 com o módulo ESP8266 ESP-01 na programação através de funções específicas que realizam a autenticação com a rede *wireless* e a comunicação com o aplicativo *Blynk* utilizado na interface com o usuário.

Uma função importante dessas bibliotecas é o uso do *BlinkTimer* para executar ações periódicas durante o funcionamento do controlador, sendo um recurso que possui melhorias de integração com a biblioteca *SimpleTimer*, visto que o aplicativo *Blynk* exige chamadas da função principal com frequência para processar os comandos de entrada e executar tarefas de conexão (BLYNK, 2017).

A biblioteca *SimpleTimer* é responsável por realizar aplicações sem interrupções com uma resolução de 1 milissegundo, evitando problemas potenciais com variáveis globais compartilhadas entre a rotina de serviço de interrupção e o programa principal (ROMANI, 2017).

5.4.2 Biblioteca MatrixMath

A utilização de matrizes no estudo da robótica é muito importante na organização dos parâmetros dos sistemas e realização dos cálculos como nas cinemáticas direta, inversa e diferencial.

Na programação do controlador, a biblioteca *MatrixMath* é utilizada para realizar as manipulações e cálculos envolvendo matrizes. Para isso, a biblioteca

utiliza matrizes armazenadas em vetores principais que devem ser declaradas nas funções junto com os valores das suas dimensões (MATLACK, 2017).

5.5 FUNÇÕES AUXILIARES

Algumas funções elaboradas na programação realizam tarefas importantes para o funcionamento do projeto definindo o valor de variáveis através cálculos e estratégias específicas, bem como realizando ações que auxiliam no fluxo dos processos.

Nos tópicos abaixo as lógicas dessas funções são explicadas de maneira simplificada.

5.5.1 Matriz de Transformação Homogênea

Com base na Equação (3.11), essa função tem como finalidade auxiliar no processo de cálculo e definição da cinemática direta e da matriz jacobiana a partir da informação dos dados de ângulo de giro no eixo da junta, do comprimento do elo, do deslocamento de junta e do ângulo de torção do elo, com base nos parâmetros de *Denavit-Hartenberg*.

5.5.2 Cinemática Inversa

A referência de um ponto no espaço ou a determinação do posicionamento do efetuador são realizadas por meio de uma função que define os ângulos da base, do ombro e do cotovelo através do cálculo da cinemática inversa do robô por meio da Equação (3.26), Equação (3.34) e Equação (3.31), respectivamente, a partir da informação do valor de cada coordenada do ponto desejado.

5.5.3 Jacobiano

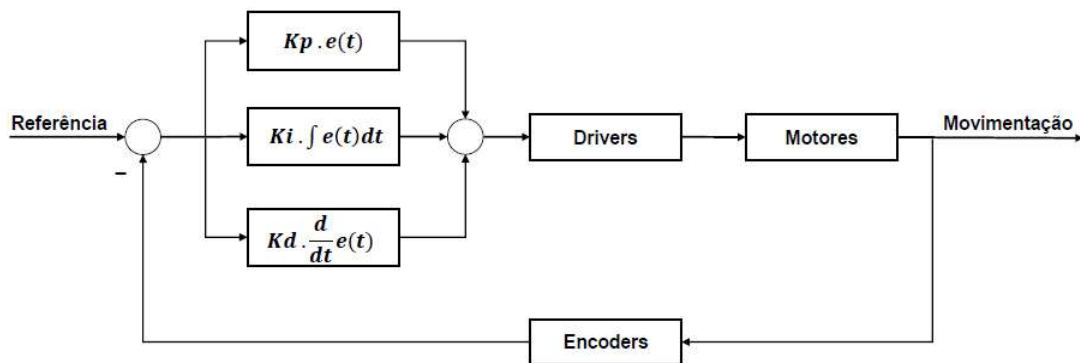
A função Jacobiano é utilizada para definir a velocidade angular das juntas a partir da velocidade do efetuador em relação às coordenadas cartesianas, sendo útil para impor os valores das velocidades inicial e final das juntas para um determinado deslocamento. Primeiramente, é montada a matriz jacobiana dada pela Equação

(3.48) por meio da informação do ponto desejada. Então, para a definição das velocidades angulares é utilizada a Equação (3.50).

5.5.4 Controle PID

Durante o deslocamento das juntas é conveniente variar a velocidade de maneira a atingir um determinado ponto no espaço com mais precisão, sem oscilações ao redor da referência e com amortecimento ao final do movimento. A variação da velocidade dos motores é realizada através de uma função de controle PID que pode ser resumida através do diagrama apresentado na Figura 43.

Figura 43 – Esquema de controle PID



Fonte: produção do próprio autor.

No diagrama, a referência é dada pelo ângulo desejado para cada junta do braço robótico, isto é, os valores que os contadores dos *encoders* devem alcançar para atingir o objetivo. Para isso, a posição angular de cada motor é lida para definir o erro que deve ser corrigido pelo controlador, representado em função do tempo no diagrama por $e(t)$. A periodicidade do controlador PID é de 50 milissegundos, fazendo com que a ação de controle ocorra em intervalos discretos.

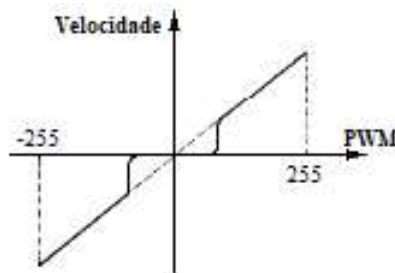
De acordo com a diferença entre a referência e o valor da posição atual dado pelos *encoders*, há um comportamento definido pelos parâmetros de controle como explicado a seguir (ESHED ROBOTEC, 1999).

- Proporcional: ajustado pelo parâmetro K_p que é multiplicado diretamente pelo erro, essa ação de controle permite uma reação rápida e efetiva do robô durante os movimentos, sendo responsável pela repetibilidade do sistema;

- Integral: adaptado através do parâmetro K_i que é multiplicado pelo erro acumulado ao longo do processo, auxilia o parâmetro proporcional a eliminar o erro de estado estacionário;
- Derivativo: regulado pelo parâmetro K_d que é multiplicado pela variação do erro em cada iteração do processo, reduz o erro de velocidade e fornece a atenuação de oscilações necessária para o atuador.

O controle PID atua na determinação da velocidade angular das juntas por meio da variação do sinal PWM enviado para os *drivers*. No entanto, existe uma tensão mínima que o motor necessita ser alimentado para realizar a movimentação, o que acarreta em limitações que devem ser consideradas na programação, analisando os esforços na atuação dos motores para cada uma das juntas como esquematizado na Figura 44. A faixa de valores em que o motor não suporta deslocar a junta deve ser desprezada, exigindo um valor mínimo de sinal PWM de acordo com as características de cada junta. Assim, para cada motor é definido um valor inicial de sinal PWM.

Figura 44 – Esforços do motor em relação ao sinal PWM



Fonte: produção do próprio autor.

5.5.5 Estimativa de tempo entre pontos

Na movimentação do braço robótico entre duas posições no espaço, um caminho é definido a partir de uma interpolação que considera a função entre dois pontos cartesianos e o tempo necessário para que o efetuador se desloque entre esses dois pontos no espaço.

Para estimar o tempo que será utilizado na definição do caminho do braço robótico, uma função é chamada levando em consideração a posição atual e o ponto que se deseja alcançar, que define a distância entre os pontos pela Equação (5.1).

$$\text{Distância} = \sqrt{(x_f - x_i)^2 + (y_f - y_i)^2 + (z_f - z_i)^2} \quad (5.1)$$

Assim, utilizando a velocidade média do efetuador, é possível definir o tempo necessário para alcançar o objetivo da movimentação por meio da Equação (5.2).

$$\text{Tempo} = \frac{\text{Distância}}{\text{Velocidade Média}} \quad (5.2)$$

5.5.6 Geração de caminhos

Os caminhos são definidos a partir da variação das referências dos ângulos nas juntas em intervalos discretos com base em uma interpolação denominada *spline* cúbica, onde a geração dos caminhos é determinada através de uma função que relaciona os ângulos de cada junta entre os dois pontos com o instante de tempo do deslocamento como mostra a Equação (5.3).

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 \\ a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 \\ a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3 \end{bmatrix} \quad (5.3)$$

Esse método determina deslocamentos suaves do efetuador de modo que todas as juntas iniciam e terminam os movimentos simultaneamente, o que leva as juntas acelerarem e desacelerarem de maneira conveniente (GEMBAROWSKI e CAMPOS, 2015).

Dessa maneira, a cada ação de controle, os ângulos de referência são atualizados a partir do cálculo da interpolação com valor do instante de tempo correspondente a iteração do controlador PID, que ocorre a cada 50 milissegundos.

Assim, os valores de tempo inicial e final são relacionados como mostra a Equação (5.4).

$$S = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_i & 3t_i^2 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \quad (5.4)$$

Por fim, os coeficientes da Equação (5.3) são definidos a partir da Equação (5.5).

$$\begin{bmatrix} a_{10} & a_{20} & a_{30} \\ a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} = S^{-1} \begin{bmatrix} \theta_{1i} & \theta_{2i} & \theta_{3i} \\ \theta_{1f} & \theta_{2f} & \theta_{3f} \\ \dot{\theta}_{1i} & \dot{\theta}_{2i} & \dot{\theta}_{3i} \\ \dot{\theta}_{1f} & \dot{\theta}_{2f} & \dot{\theta}_{3f} \end{bmatrix} \quad (5.5)$$

5.6 MOVIMENTAÇÃO DO BRAÇO ROBÓTICO

As movimentações das juntas são realizadas a partir da comparação dos valores das posições angulares desejadas com o estado atual do braço robótico. Assim, uma função é responsável por integrar todas as ações apresentadas no diagrama da Figura 39 a partir da informação do valor do ângulo para cada junta.

A resolução da posição angular das juntas é definida de acordo com a quantidade de pulsos que definem o valor dos contadores dos *encoders* acoplados nos motores para deslocar um grau no ângulo de cada junta. Para o caso da garra, a resolução é definida pela porcentagem de abertura.

A Tabela 10 apresenta as relações dos valores dos *encoders* para cada motor, obtidas através de medições e observações experimentais a partir da contagem de pulsos elétricos enviados para que as juntas atingissem seus limites.

Tabela 10 – Relação dos *encoders*

Motor	Junta	Relação
1	Base	70 pulsos
2	Ombro	55 pulsos
3	Cotovelo	55 pulsos
4	Punho	12 pulsos
5	Punho	12 pulsos
6	Garra	23 pulsos

Fonte: produção do próprio autor.

Dessa maneira é possível definir facilmente a referência que os contadores dos *encoders* devem atingir para alcançar a posição angular desejada ou a porcentagem de abertura da garra.

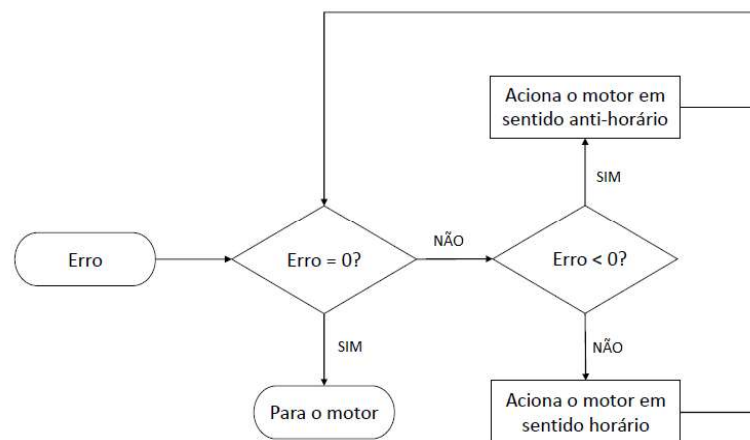
Se o valor da referência for maior que o valor da posição atual, o motor girará no sentido horário. Caso contrário, o motor será acionado no sentido anti-horário. O programa realiza a movimentação com auxílio da função de controle PID até atingir a posição desejada e ativar o freio. Havendo variação do valor do *encoder*, o controlador atuará os motores novamente a fim de manter a referência, visto que os

sinais dos *encoders* são tratados com prioridade na programação por meio das funções de interrupção.

Dessa maneira, o braço robótico realiza os movimentos a partir dos comandos do usuário dentro dos limites impostos pela programação do controlador, mesmo com interferências externas como aplicação de cargas.

O fluxograma apresentado na Figura 45 resume a lógica de funcionamento dos motores para movimentar o braço robótico.

Figura 45 – Fluxograma de funcionamento dos motores



Fonte: produção do próprio autor.

6 RESULTADOS

Os resultados deste trabalho foram definidos a partir da aquisição de dados diretamente dos *encoders* do robô para posterior comparação com a situação ideal de comportamento.

Assim, foi possível verificar a eficiência do controlador com base na programação e estratégias adotadas, além de se poder observar parâmetros como a repetibilidade e a precisão do braço robótico por meio da organização das informações obtidas e análise da teoria com o funcionamento do projeto.

6.1 SINTONIZAÇÃO DE PARÂMETROS

Para o ajuste do controlador deste projeto, é possível realizar modificações nos parâmetros proporcional, integral e derivativo conforme se julgar melhor. Existem diversas maneiras de sintonizar esses valores, mas para os resultados demonstrados neste trabalho foi usado o método de *Ziegler-Nichols*.

A sintonia dos parâmetros PID pelo método *Ziegler-Nichols* é realizada por meio de regras práticas com base na resposta em frequência ou na resposta ao degrau do processo. O método é muito utilizado tanto na sua forma original quanto com algumas modificações a partir de características da dinâmica do processo que definem os parâmetros do controlador em termos de simples fórmulas como mostra a Tabela 11 (ÅSTRÖM e HÄGGLUND, 2006).

Tabela 11 – Sintonia de parâmetros por *Ziegler-Nichols*

Controlador	K_p	K_i	K_d
P	$0,50. K_u$	0	0
PI	$0,45. K_u$	$0,54. K_u/T_u$	0
PD	$0,60. K_u$	0	$0,075 K_u. T_u$
PID	$0,60. K_u$	$1,2. K_u/T_u$	$0,075 K_u. T_u$

Fonte: (ÅSTRÖM e HÄGGLUND, 2006).

Sendo K_u o ganho crítico dado pelo valor máximo de K_P sem a influência de K_i e K_d de modo que o sistema possua uma resposta oscilatória sustentada e T_u é o período da oscilação crítica.

Os critérios de estabilização dado pelo método de *Ziegler-Nichols* são empíricos e dão resultados aceitáveis, porém foram realizados ajustes finos nos valores encontrados no processo. A análise do controlador proposto foi realizada a

partir da reação de cada uma das juntas do braço robótico às variações dos parâmetros a partir dos procedimentos detalhados a seguir.

1. Os parâmetros integral e derivativo foram desconsiderados, anulando os valores de K_i e K_d ;
2. Impondo o controlador como um proporcional simples, o valor de K_p foi aumentado até atingir K_u que corresponde ao valor crítico de K_p onde o braço robótico começa a oscilar de maneira sustentada;
3. Foi realizada a aquisição dos dados de resposta do robô de maneira que foi possível obter o período de oscilações T_u para o ganho K_u .

Os procedimentos citados costumam ser utilizados na sintonização de parâmetros em processos contínuos. No entanto, neste trabalho foi adotado esse método por apresentar resultados satisfatórios a partir da análise do comportamento do braço robótico e uma adaptação para o caso controle discreto.

Dessa maneira a sintonia dos parâmetros foi feita de maneira experimental, onde os ajustes foram realizados de acordo com os resultados e o comportamento do braço robótico. Os valores de K_u e T_u para o motor de cada uma das juntas estão demonstrados na Tabela 12.

Tabela 12 – Resultados experimentais

Motor	K_u	T_u
1	17,2	0,17
2	15,8	0,12
3	23,4	0,26
4	3,7	0,16
5	3,1	0,27
6	35,1	0,10

Fonte: produção do próprio autor.

6.2 EXECUÇÃO DE TRAJETÓRIAS

A referência dos ângulos que cada junta do braço robótico deve atingir em um dado instante de tempo é definido por meio da interpolação por *spline* cúbica que gera um caminho com o deslocamento angular suave com base no controle da velocidade dos motores de modo que todas juntas que determinam a posição do efetuador iniciem e terminem o movimento nos mesmos instantes de tempo. Outra característica importante é que o uso da interpolação por *spline* cúbicas na

determinação do caminho reduz consideravelmente as vibrações durante a movimentação do robô.

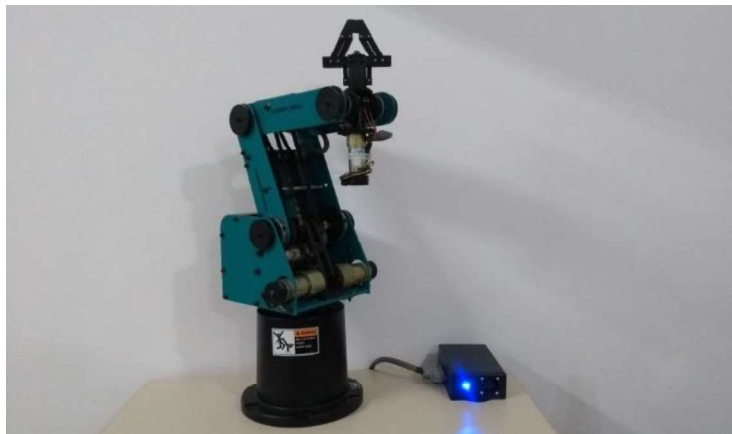
Para verificação dos resultados na execução das trajetórias, foi definido um caminho que o robô executa conforme é apresentado na Figura 46 e na Figura 47, que ilustram a postura inicial e final do braço robótico no caminho desenvolvido.

Figura 46 – Postura inicial



Fonte: produção do próprio autor.

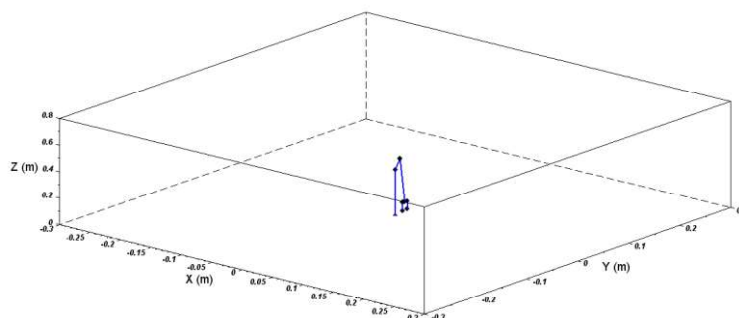
Figura 47 – Postura final



Fonte: produção do próprio autor.

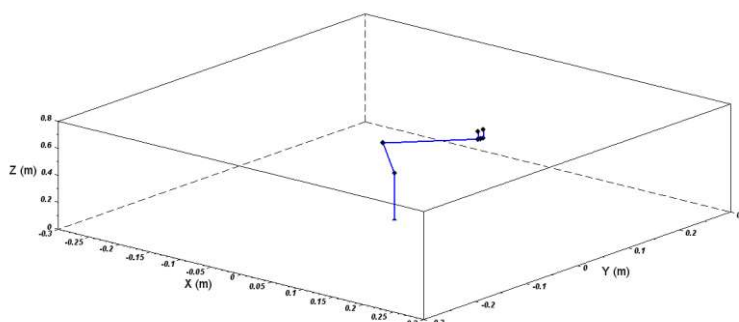
Para comparação dos resultados, a movimentação do braço robótico foi simulada com auxílio do *software* Scilab como demonstra a Figura 48 e a Figura 49.

Figura 48 – Postura inicial simulada



Fonte: produção do próprio autor.

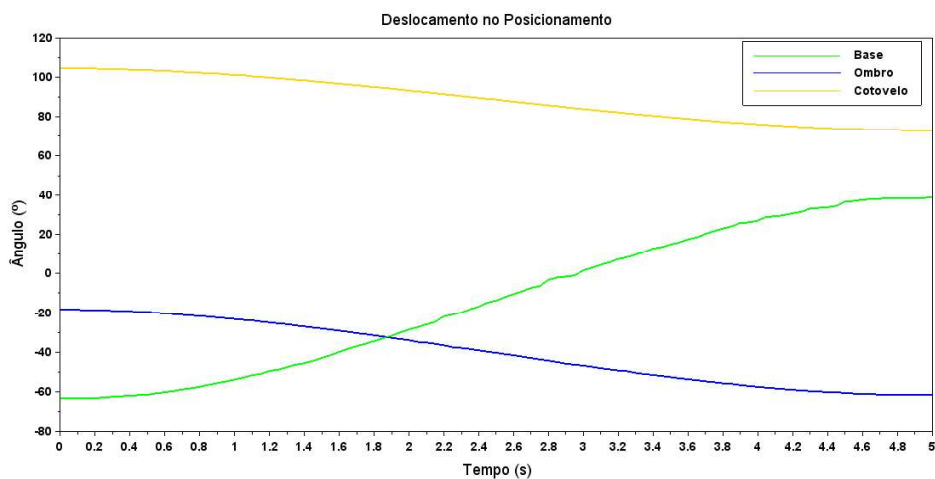
Figura 49 – Postura final simulada



Fonte: produção do próprio autor.

Na Figura 50 são demonstrados graficamente os resultados da execução do caminho com auxílio do controle PID nas três primeiras juntas, responsáveis pelo posicionamento do braço robótico. É importante notar que os movimentos das juntas iniciam e terminam no mesmo instante de tempo.

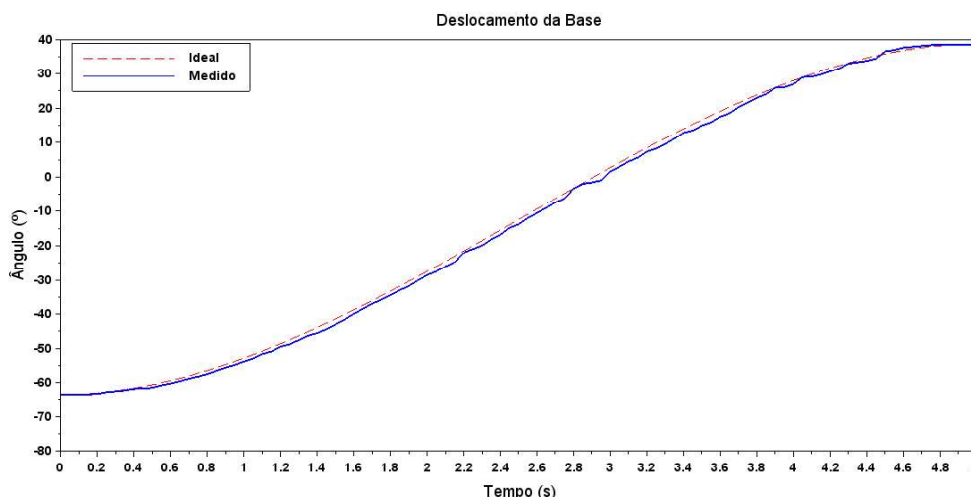
Figura 50 – Deslocamento angular no posicionamento



Fonte: produção do próprio autor.

A Figura 51 detalha a movimentação da base na realização do caminho em relação à referência dada pela interpolação por *spline* cúbica.

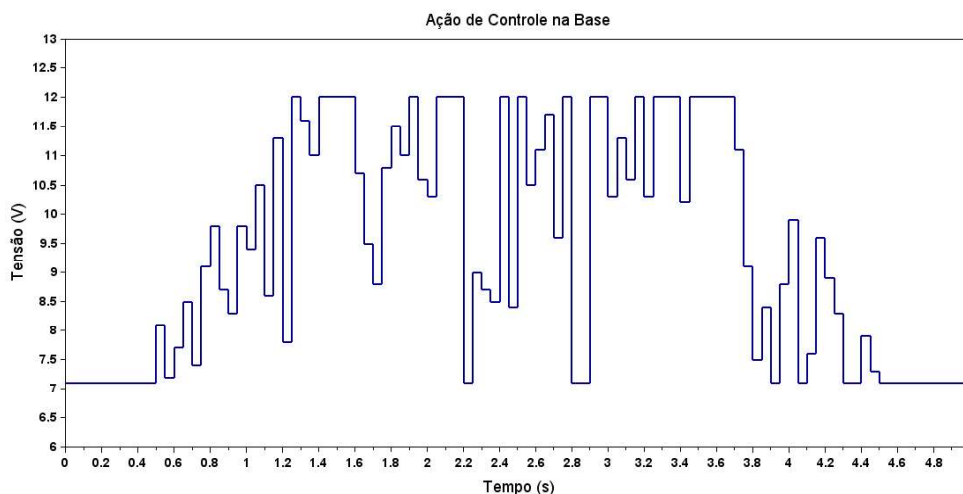
Figura 51 – Deslocamento angular da base



Fonte: produção do próprio autor.

A ação de controle dada pela tensão no motor da junta da base no intervalo apresentado é ilustrada na Figura 52.

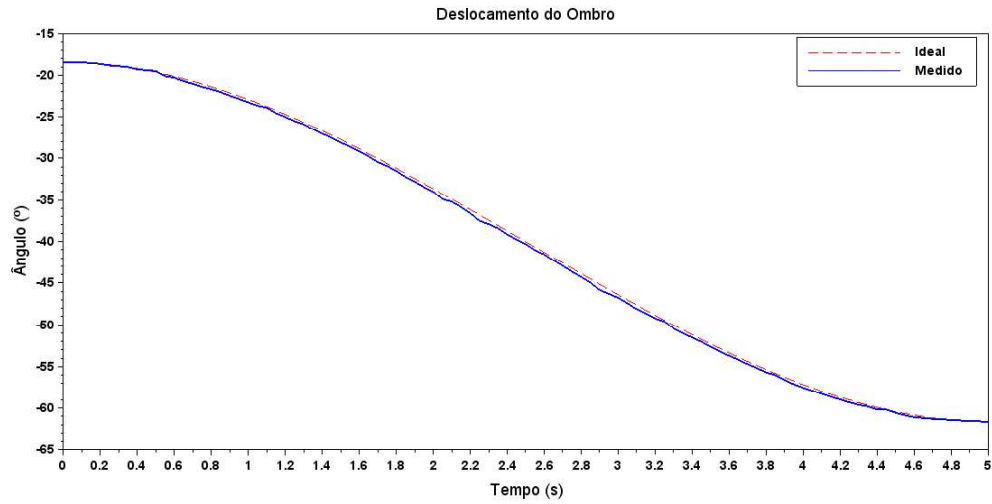
Figura 52 – Ação de controle na base com controle PID



Fonte: produção do próprio autor.

Para o caso do deslocamento do ombro na execução da trajetória em relação à referência, o gráfico é mostra na Figura 53.

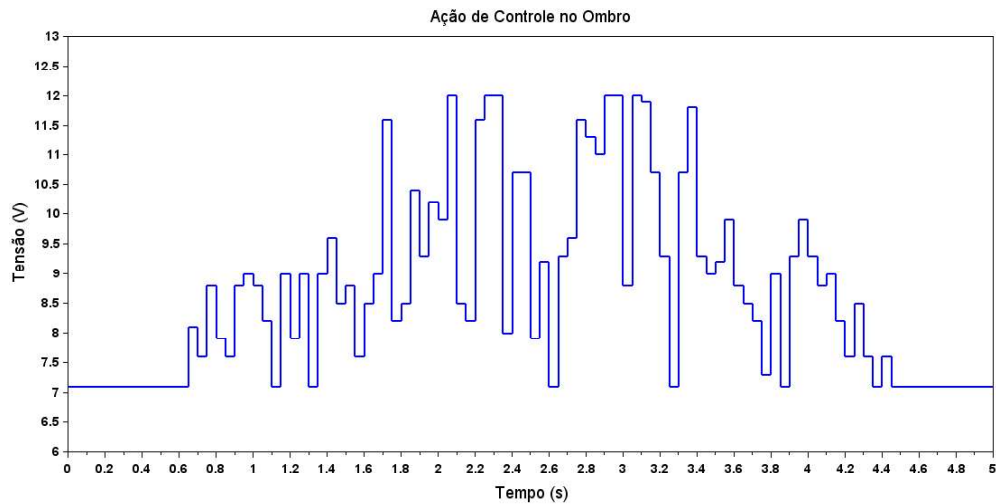
Figura 53 – Deslocamento angular do ombro



Fonte: produção do próprio autor.

A variação de tensão no motor dado pelo sinal de controle no intervalo apresentado é mostrada na Figura 54.

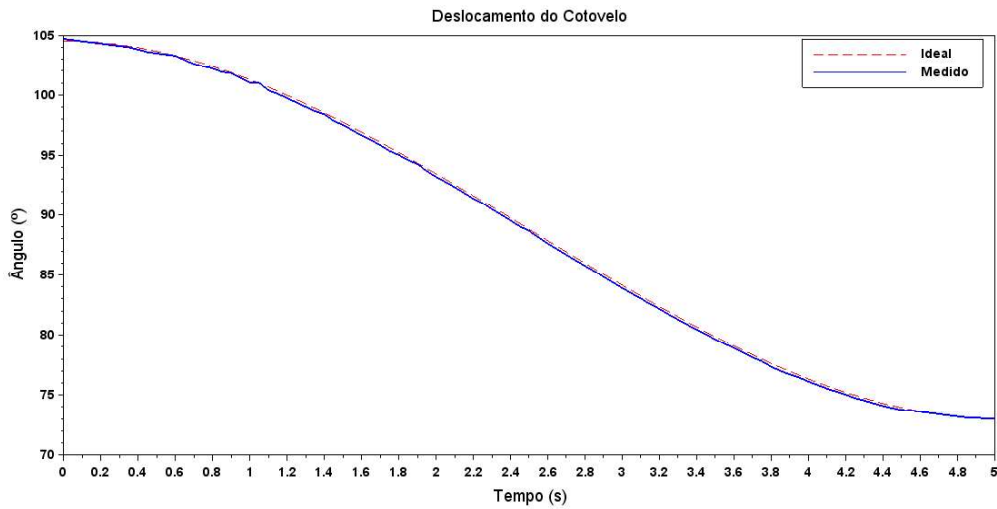
Figura 54 – Ação de controle no ombro com controle PID



Fonte: produção do próprio autor.

A Figura 55 mostra a movimentação do cotovelo na realização do caminho em relação à referência.

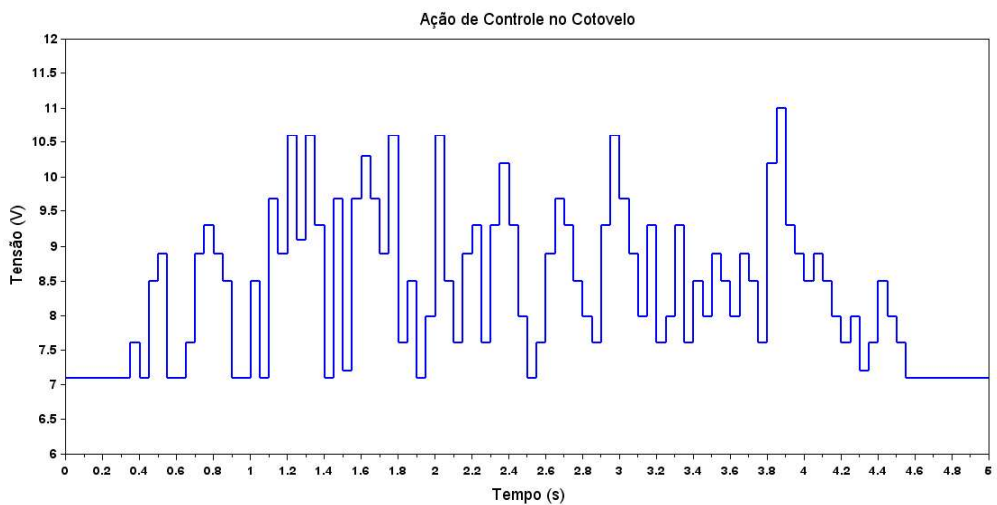
Figura 55 – Deslocamento angular do cotovelo



Fonte: produção do próprio autor.

A ação de controle no motor do cotovelo no intervalo medido é ilustrada na Figura 56.

Figura 56 – Ação de controle com controle PID



Fonte: produção do próprio autor.

A partir dos gráficos que demonstram o comportamento do braço robótico no posicionamento do efetuador no espaço, pode-se afirmar que a estratégia de execução de trajetórias por meio da interpolação por *spline* cúbica satisfaz a condição de deslocamento suave das juntas.

Para os casos apresentados, o tempo de deslocamento é controlado por meio da variação da velocidade das juntas de acordo com a referência dada pela interpolação calculada a cada iteração do controlador. Isso se deve ao fato de que a

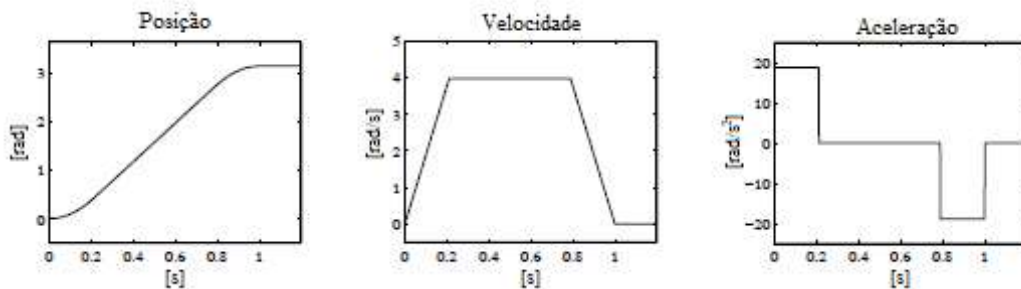
ação de controle define as tensões aplicadas nos motores das juntas, determinando a velocidade com base no erro calculado pelo controlador.

Para otimizar a suavização do movimento, o controle do projeto impõe que as juntas responsáveis pelo posicionamento do efetuador iniciem e encerrem seus movimentos no mesmo instante de tempo. Tal condição apresentou resultados satisfatórios, como pode ser observado nos gráficos.

As ações de controle ilustram os esforços do controlador para manter as juntas na referência imposta, variando a velocidade dos motores conforme as necessidades ao longo do deslocamento. É importante notar que, apesar da estratégia de controle não contemplar as influências gravitacionais, a velocidade apresentada lembra um formato trapezoidal devido às saturações impostas com base nos limites mínimo e máximo de tensão aplicada nos motores.

Devido às simplificações matemáticas utilizadas no projeto, os movimentos de arfagem e rolagem não possuem controle por interpolação dos pontos no tempo, mas a suavização do deslocamento angular é realizada através da aceleração e desaceleração controlada dos motores, a fim de que a velocidade possuam um perfil trapezoidal, assim como propõem (SICILIANO, SCIAVICCO, *et al.*, 2009), cujo perfil da posição, velocidade e aceleração para esses casos é ilustrado na Figura 57.

Figura 57 – Perfil de suavização do movimento



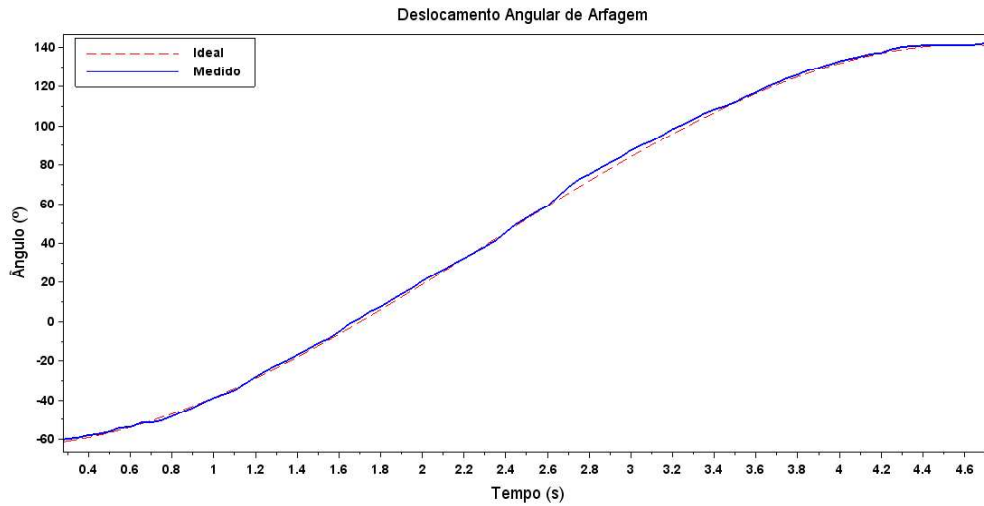
Fonte: (SICILIANO, SCIAVICCO, *et al.*, 2009).

Outro fato que se pode verificar é que o deslocamento angular para esta estratégia possui um perfil semelhante à interpolação por *spline* cúbica, mas o caminho não é limitado no tempo. Dessa forma, os movimentos de orientação da garra ocorrem livremente de acordo com as referências impostas.

A Figura 58 demonstra o deslocamento angular do movimento de arfagem dada pela média dos dados dos *encoders* 4 e 5, visto que os motores 4 e 5 atuam

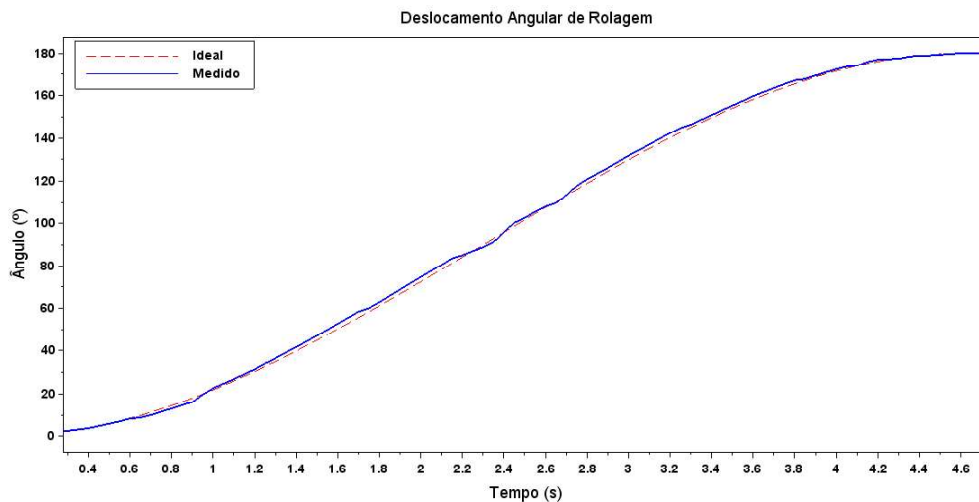
de maneira colaborativa no mesmo sentido de rotação. No caso da rolagem, os motores atuam em sentidos opostos e o gráfico dessa movimentação pode ser visualizado na Figura 59.

Figura 58 – Deslocamento angular de arfagem



Fonte: produção do próprio autor.

Figura 59 – Deslocamento angular de rolagem

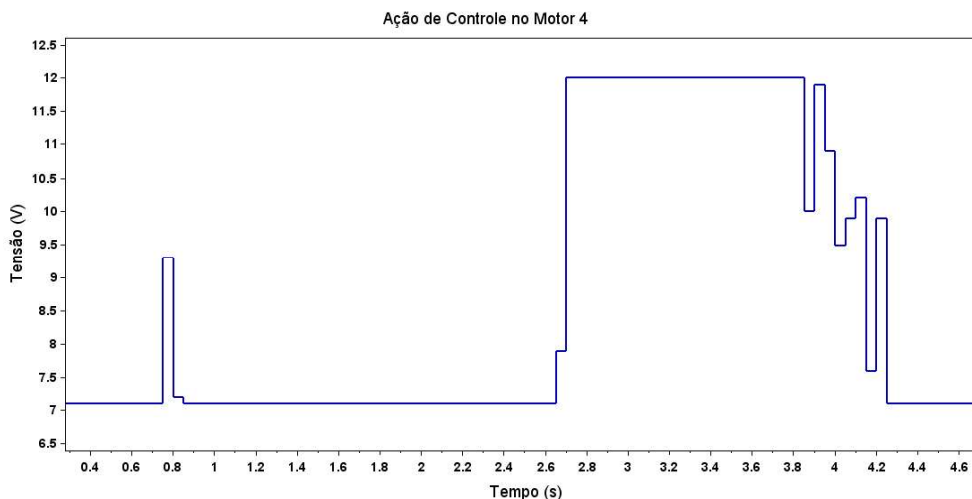


Fonte: produção do próprio autor.

Como os movimentos de arfagem e rolagem da garra ocorrem simultaneamente pelo fato de compartilharem os mesmos motores, as ações de controle para estes casos devem ser analisadas de maneira conjunta, onde os esforços para controle da velocidade de um motor ocorre de maneira colaborativa

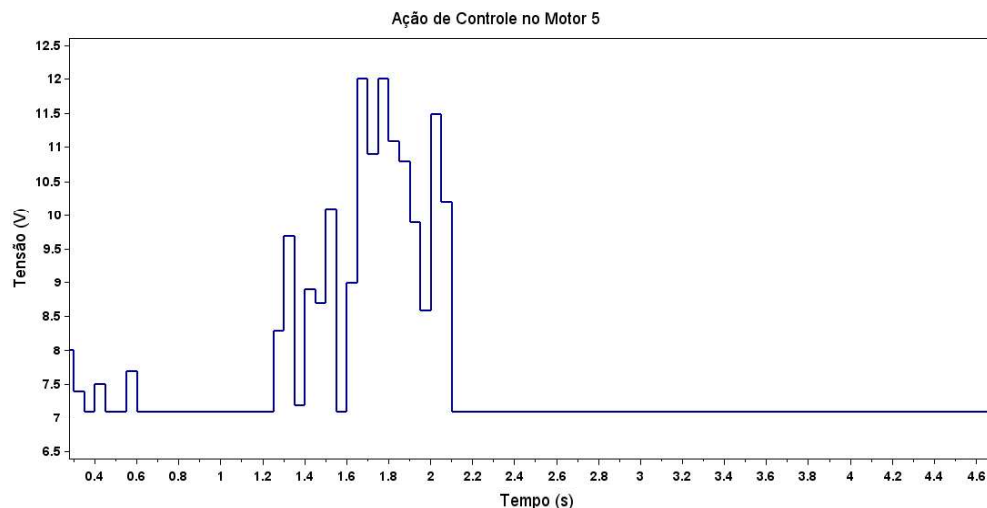
em relação ao outro, como é possível observar na Figura 60 que ilustra a tensão aplicada no motor 4 e na Figura 61 que mostra a tensão no motor 5 no intervalo de deslocamento.

Figura 60 – Ação de controle no motor 4



Fonte: produção do próprio autor.

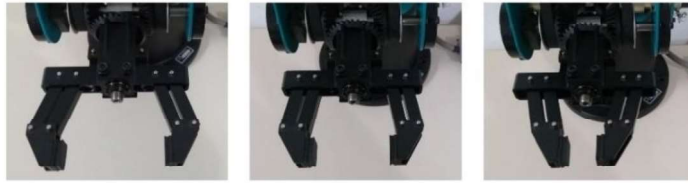
Figura 61 – Ação de controle no motor 5



Fonte: produção do próprio autor.

Para obter o perfil de deslocamento linear da garra, foram realizados movimentos que correspondem a uma porcentagem de abertura. A Figura 62 apresenta uma sequência de imagens em que a garra está 100%, 75% e 25% aberta, respectivamente.

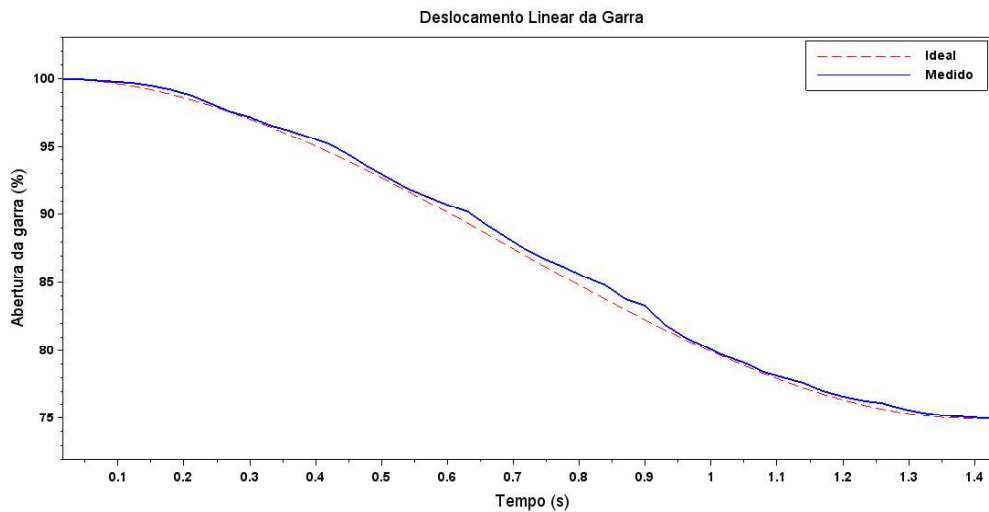
Figura 62 – Abertura da garra



Fonte: produção do próprio autor.

A mesma estratégia dos movimentos de arfagem e rolagem foram usados para a garra, ou seja, há um controle da velocidade durante o movimento. A Figura 63 mostra o comportamento da garra no fechamento da garra de uma abertura de 100% para 75%.

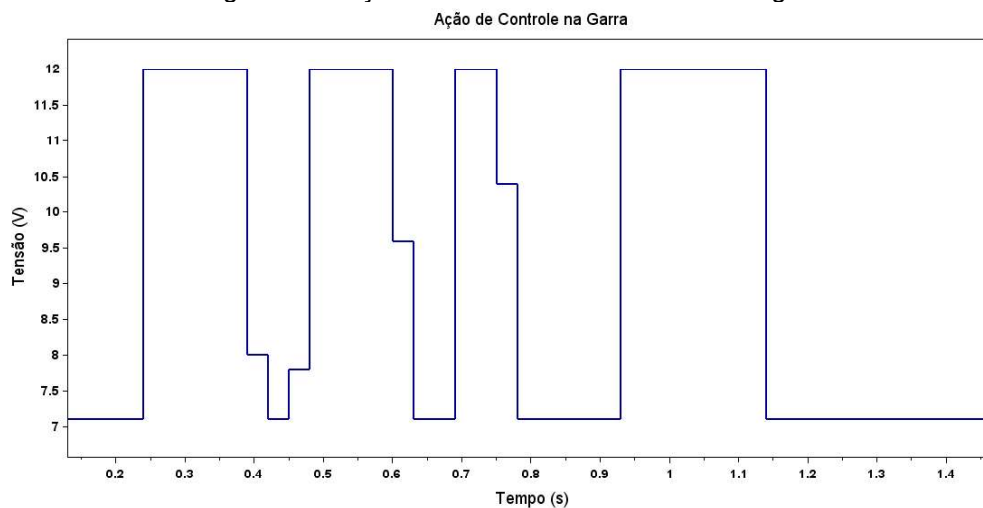
Figura 63 – Fechamento da garra de 100% para 75%



Fonte: produção do próprio autor.

A ação de controle ilustrada na Figura 64 evidencia mudanças bruscas na velocidade do motor da garra a fim de manter a referência imposta.

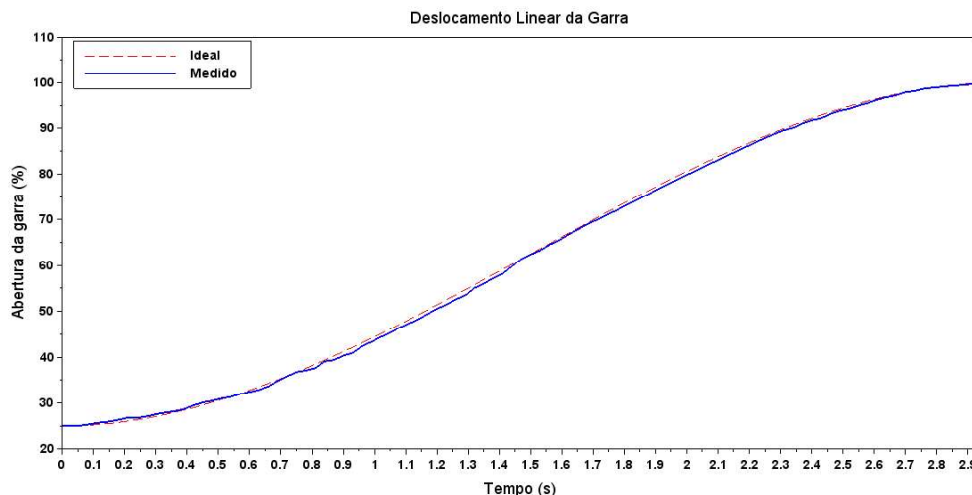
Figura 64 – Ação de controle no fechamento da garra



Fonte: produção do próprio autor.

A Figura 65 mostra o comportamento da garra na abertura da garra de 25% para 100%.

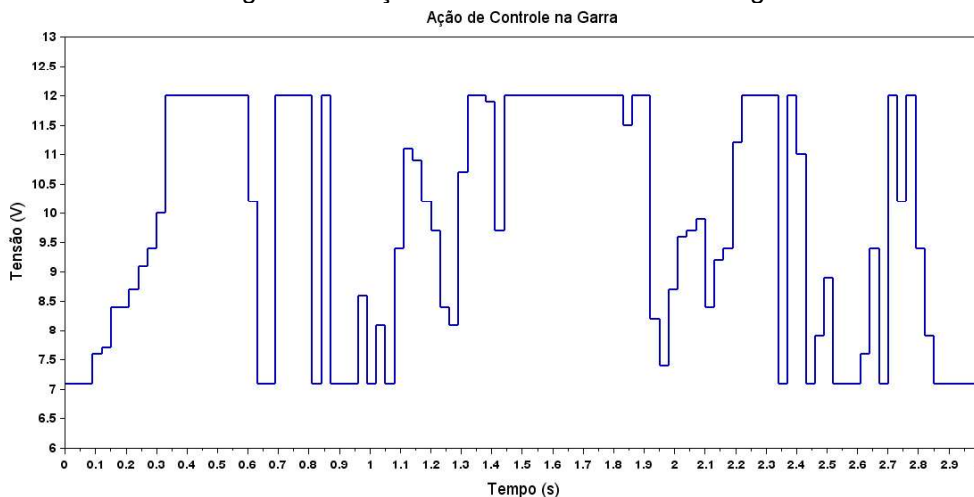
Figura 65 – Abertura da garra de 25% para 100%



Fonte: produção do próprio autor.

A Figura 66 mostra a ação de controle durante o processo de abertura da garra.

Figura 66 – Ação de controle na abertura da garra



Fonte: produção do próprio autor.

Os resultados apresentados mostram que o controlador proposto atende às necessidades de execução de trajetórias do projeto, apresentando movimentos suaves na geração dos caminhos com base no controle da velocidade dos motores ao longo do deslocamento das juntas.

6.3 CONTROLE DE POSICIONAMENTO

Os ensaios para obtenção dos resultados no controle do posicionamento do braço robótico foram realizados por meio da aplicação de carga em uma posição até que houvesse alguma vibração ou esforço visível do robô para manter o efetuator no ponto desejado.

No primeiro caso, foi aplicado uma carga de 3 kg na garra do braço robótico como mostra a Figura 67.

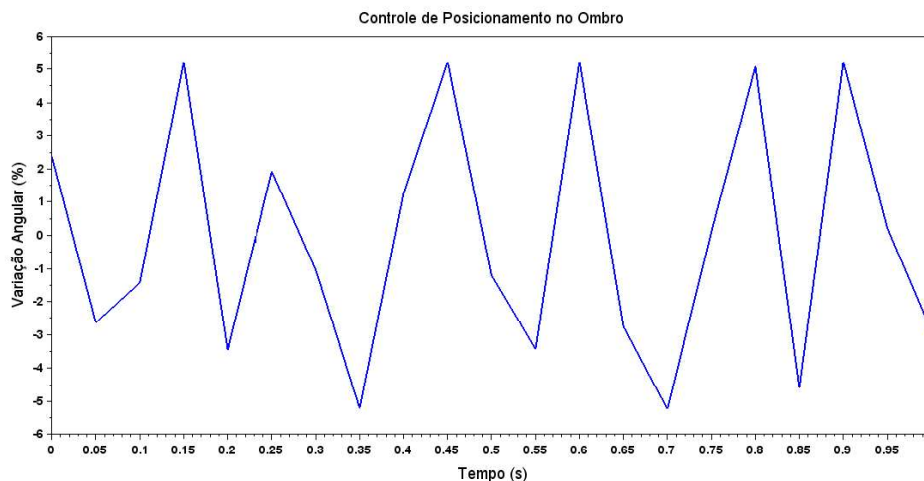
Figura 67 – Primeiro caso de aplicação de carga



Fonte: produção do próprio autor.

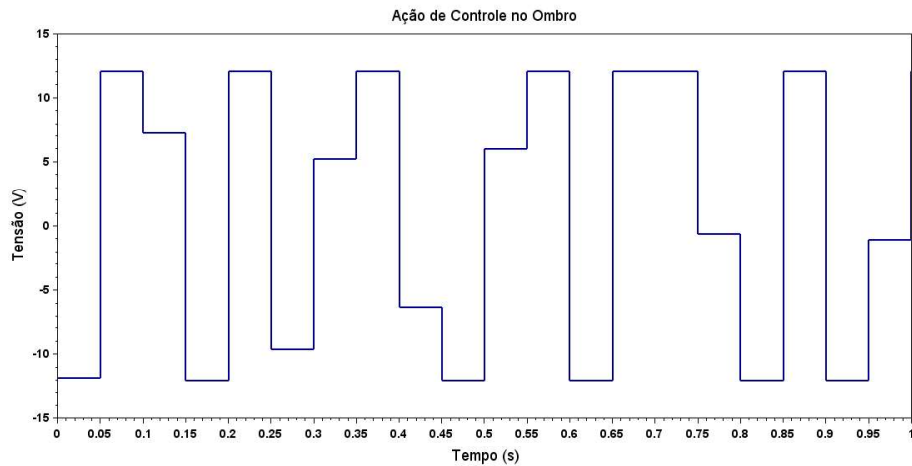
Os resultados apresentados a seguir mostram a porcentagem de variação angular nas juntas que possuem movimentos verticais, ou seja, ombro, cotovelo e o movimento de arfagem. Essa análise permite que seja observada as ações de controle de posicionamento a partir da influência gravitacional de uma carga externa no efetuator. A variação angular no ombro é observada na Figura 68 e o esforço de controle no posicionamento é verificado na Figura 69.

Figura 68 – Variação angular do ombro no primeiro caso



Fonte: produção do próprio autor.

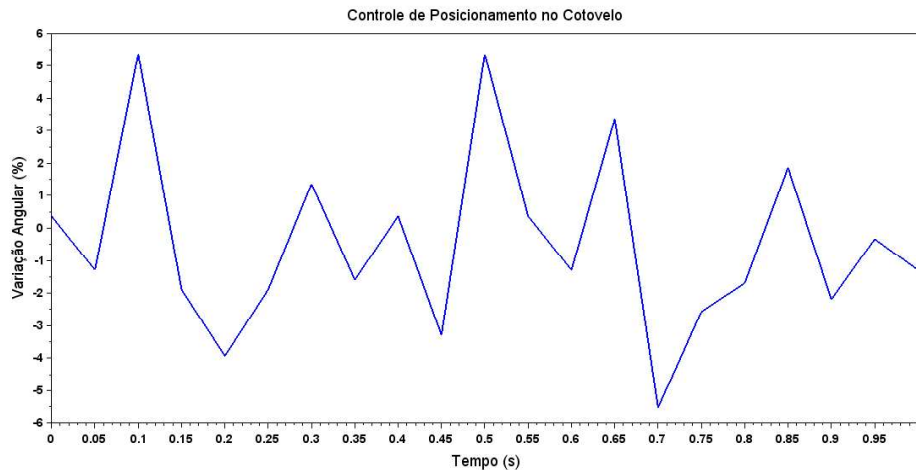
Figura 69 – Ação de controle do ombro no primeiro caso



Fonte: produção do próprio autor.

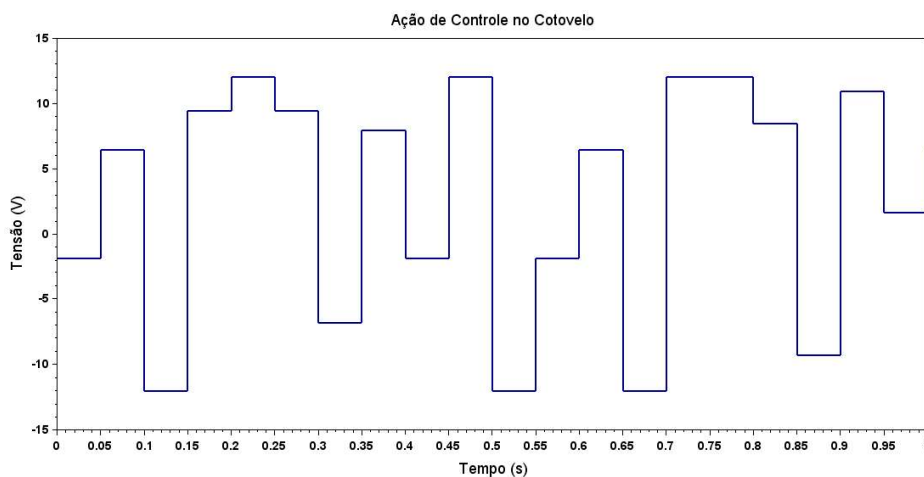
No caso do cotovelo, a Figura 70 mostra a porcentagem de variação angular na junta e a Figura 71 apresenta a ação de controle.

Figura 70 – Variação angular do cotovelo no primeiro caso



Fonte: produção do próprio autor.

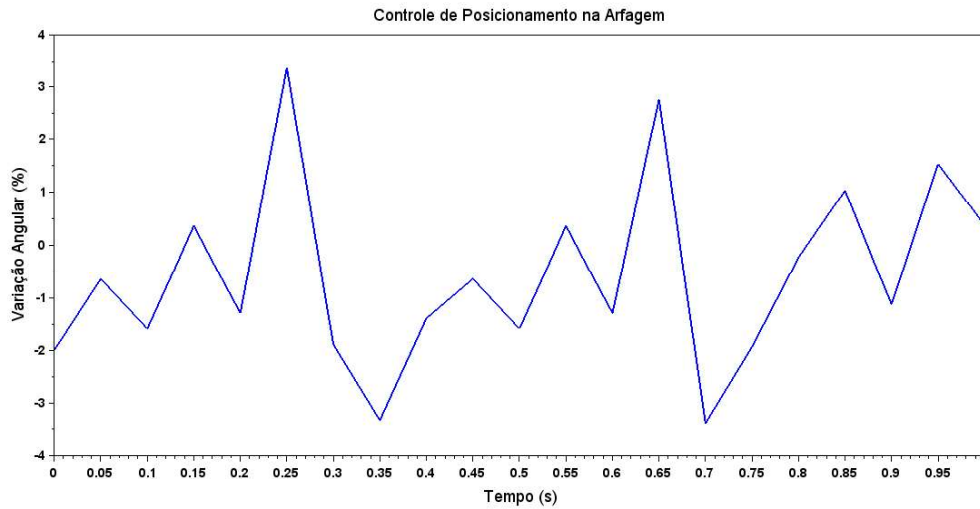
Figura 71 – Ação de controle do cotovelo no primeiro caso



Fonte: produção do próprio autor.

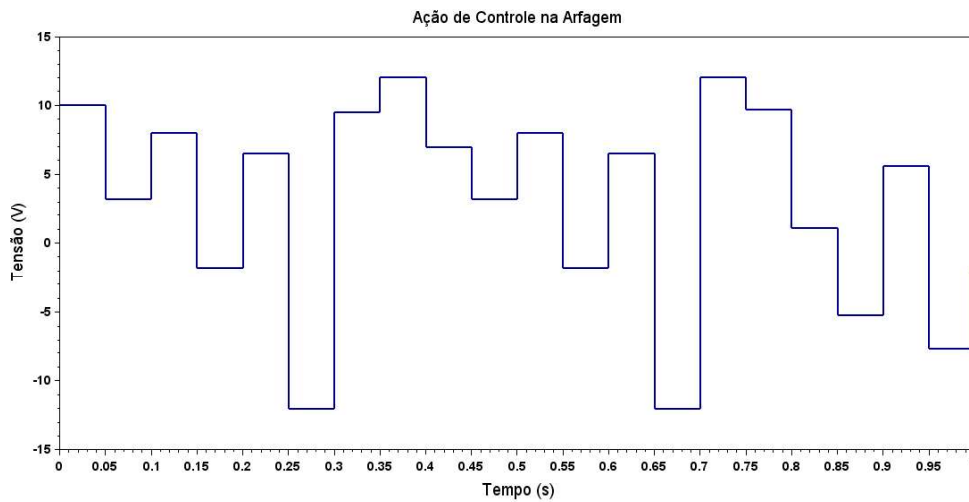
A porcentagem de variação angular no movimento de arfagem é mostrada na Figura 72 e a ação de controle no posicionamento é apresentada na Figura 73.

Figura 72 – Variação angular da arfagem no primeiro caso



Fonte: produção do próprio autor.

Figura 73 – Ação de controle da arfagem no primeiro caso



Fonte: produção do próprio autor.

No segundo caso, a carga aplicada também foi de 3 kg, mas o cotovelo foi elevado em relação ao primeiro caso como mostra a Figura 74.

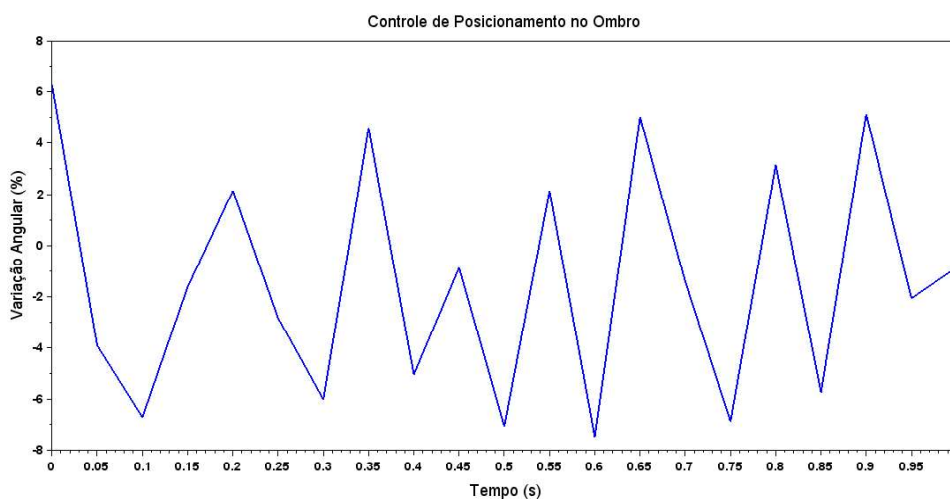
Figura 74 – Segundo caso de aplicação de carga



Fonte: produção do próprio autor.

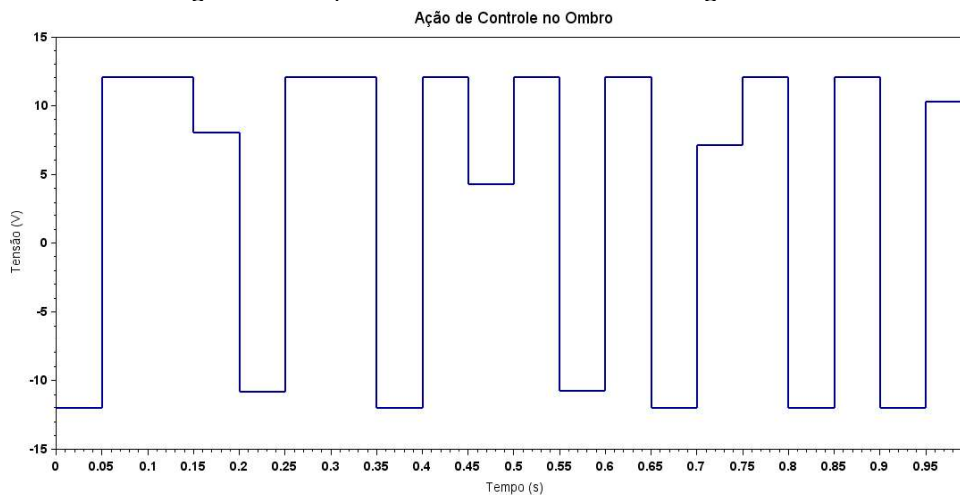
A variação angular no ombro é observada na Figura 75 e a ação de controle no posicionamento é verificado na Figura 76.

Figura 75 – Variação angular do ombro no segundo caso



Fonte: produção do próprio autor.

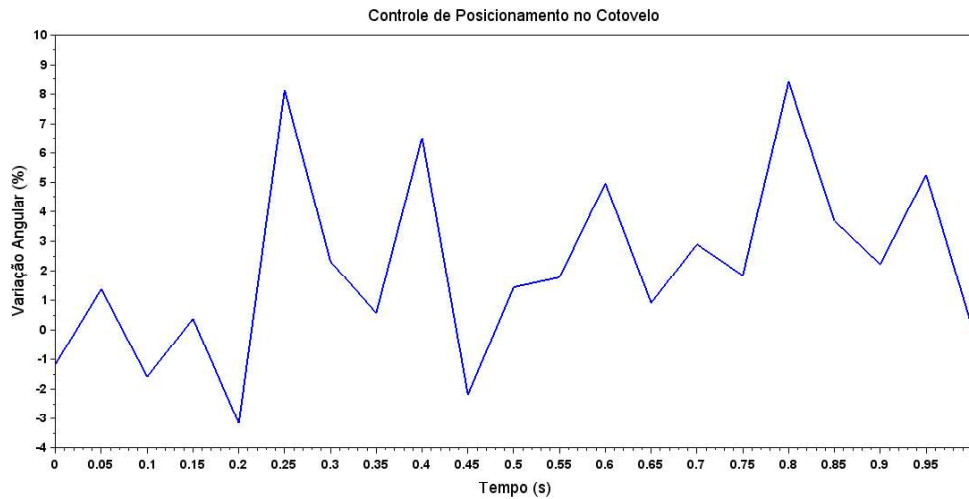
Figura 76 – Ação de controle do ombro no segundo caso



Fonte: produção do próprio autor.

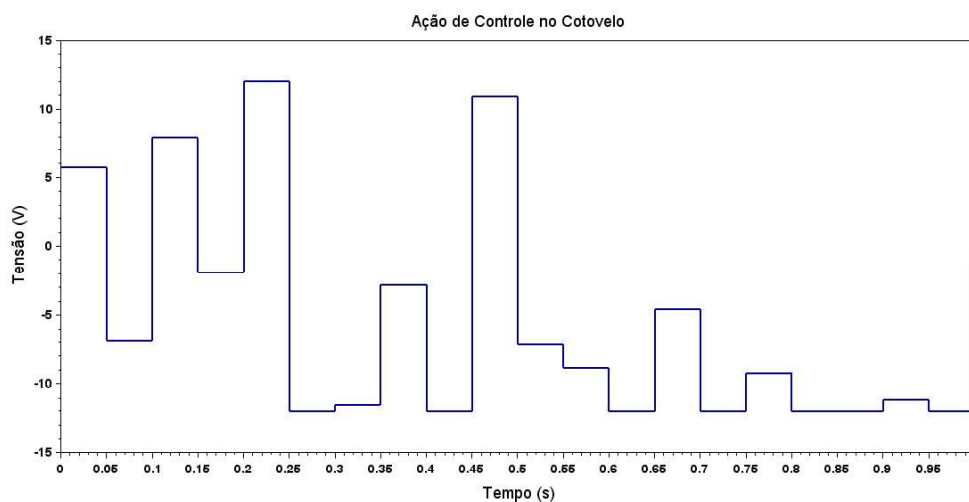
No caso do cotovelo, a Figura 77 mostra a porcentagem de variação angular na junta e a Figura 78 apresenta o esforço de controle.

Figura 77 – Variação angular do cotovelo no segundo caso



Fonte: produção do próprio autor.

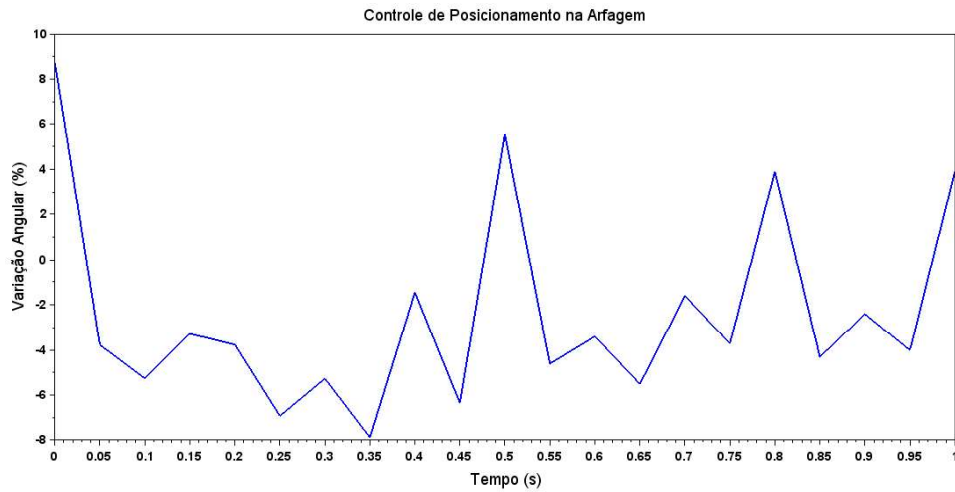
Figura 78 – Ação de controle do cotovelo no segundo caso



Fonte: produção do próprio autor.

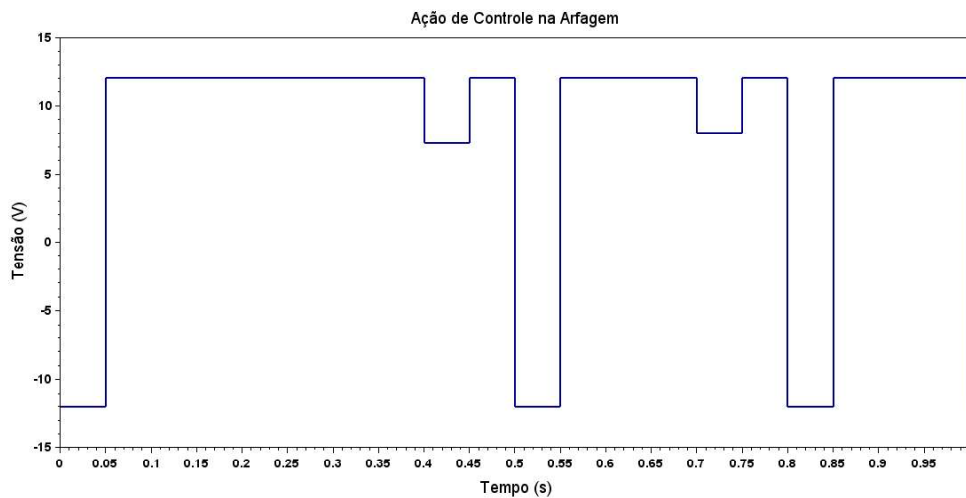
A porcentagem de variação angular no movimento de arfagem é mostrada na Figura 79 e a ação de controle no posicionamento é apresentado na Figura 80.

Figura 79 – Variação angular da arfagem no segundo caso



Fonte: produção do próprio autor.

Figura 80 – Ação de controle da arfagem no segundo caso



Fonte: produção do próprio autor.

Os resultados apresentados têm como finalidade demonstrar a eficiência do controlador quanto ao posicionamento em caso de variação em relação à uma referência, detectada pelos *encoders*, que neste caso foi definida por um ponto fixo no espaço, onde o braço robótico tende a manter sua postura mesmo com uma influência externa. É importante ressaltar que a carga utilizada nesse ensaio foi de 3 kg, mas a carga máxima recomendada pelo fabricante, incluindo a garra, é de 1 kg. Logo, é possível afirmar que os resultados apresentados foram bastante satisfatórios.

7 CONCLUSÃO

O controlador proposto apresentou resultados satisfatórios no seu funcionamento utilizando elementos de baixo custo e um código aberto na sua programação, cumprindo o objetivo principal do projeto. Assim, o protótipo desenvolvido pode ser utilizado como objeto de estudo em diversas áreas do conhecimento. Para elucidar as aplicações práticas deste projeto no ensino da robótica, alguns roteiros didáticos podem ser desenvolvidos com o controlador proposto.

O uso do braço robótico na execução de tarefas programadas remete a aspectos comuns no ambiente industrial, sobretudo em tarefas de manipulação, que necessitam de uma sequência de ações programadas pelo operador de forma que um processo seja realizado. Assim, as noções básicas de programação de sistemas robóticos podem ser desenvolvidas através de roteiros que envolvam uma situação que precise ser resolvida com a utilização do robô.

Um exemplo de atividade didática é a manipulação de objetos que precisam ser organizados com base em condições específicas, bem como restrições impostas no roteiro proposto como ações de espera ou comandos do usuário. O estudante pode utilizar a interface de operação para gravar os pontos de interesse de maneira que sejam realizados os caminhos adequados do efetuador e definir as ações realizadas pelo braço robótico.

Com a utilização de um controlador aberto é possível evidenciar os assuntos que são abordadas no estudo da robótica como as cinemáticas direta, inversa e diferencial, de modo que haja uma relação direta entre as análises matemáticas estabelecidas e a aplicação no sistema levando em consideração aspectos reais como as limitações físicas do robô.

Uma proposta de roteiro didático é a realização de exercícios para determinação das equações elementares do robô a partir de características básicas e informações dadas pelo catálogo do fabricante do braço robótico para validação dos cálculos no sistema real em seguida.

A verificação da teoria em um ambiente experimental, no qual é possível realizar a análise prática do sistema em funcionamento, tende a expor o estudante em uma situação que facilita o entendimento e estimula a construção de relações entre os assuntos abordados em aula (LEITE, SILVA e VAZ, 2005).

O controle utilizado no projeto é do tipo PID, no qual os valores padrões são definidos sempre que o controlador é iniciado. No entanto, a possibilidade de modificação dos valores dos parâmetros proporcional, integral e derivativo através da interface de operação permite que atividades didáticas possam ser realizadas com base em ensaios no controlador a partir de alterações realizadas diretamente pelo usuário para que seja analisado o comportamento do braço robótico com tais modificações.

Um exemplo de roteiro para aplicação em uma aula prática é o próprio procedimento de sintonia de parâmetros utilizado neste trabalho, no qual o método de *Ziegler-Nichols* foi adaptado por meio da análise do comportamento do robô. No entanto, outras formas de regulação do controlador podem ser utilizadas, bem como é possível omitir um ou mais parâmetros a fim de abordar outras estratégias de controle.

Uma possibilidade de avaliação da eficiência do controlador como ferramenta didática pode ser realizado por meio de estudos e aplicações práticas como realizado em (TEIXEIRA, 2017) que aborda aspectos do processo ensino-aprendizagem de robótica industrial. Dessa maneira, o projeto pode ser adaptado para uso em sala de aula para estudos como as teorias cinemáticas ou pela verificação de estratégias e implementações que proporcionam análises aprofundadas do controlador. Neste contexto, o trabalho realizado contribuiu com a possibilidade do braço robótico ser utilizado novamente nos laboratórios da universidade, visto que o controlador original encontra-se danificado e o custo elevado na manutenção impedia seu uso.

O protótipo também pode ser usado em estudos que visam o aprimoramento dos recursos desenvolvidos, bem como a adaptação de funcionalidades adicionais ou a aplicação de estratégias de controle alternativos para o processo. Nesse contexto, é importante salientar o objetivo colaborativo do projeto a partir do qual pretende-se incentivar estudantes e pesquisadores a aperfeiçoarem o protótipo proposto e expandirem as técnicas abordadas neste trabalho de modo que hajam aplicações efetivas no uso do braço robótico, principalmente no ensino da robótica e controle de processos.

Algumas sugestões para trabalhos futuros são citadas a seguir:

- Os componentes do projeto são dispostos em módulos, de maneira independente, conectados por condutores responsáveis por emitir os

sinais de controle e potência, sem a utilização de uma placa dedicada para este fim. Dessa forma, existem desvantagens em relação ao risco de falhas por eventuais problemas relacionados com cabeamento como má conexão e interferências eletromagnéticas. A utilização de uma placa de circuito impresso reduziria as dimensões do controlador e otimizaria o uso dos componentes da solução, além de reduzir significativamente os riscos quanto às falhas citadas;

- O projeto não aborda as equações dinâmicas do sistema robótico, o que representa uma boa oportunidade de estudo e implementação no braço robótico tratado neste trabalho;
- Não foram incluídos dispositivos de entrada ou de saída no controlador para interação do sistema com elementos externos como botões, sensores e relés. Seria interessante incluir estes recursos ao projeto visto que possibilita a troca de informações com o ambiente de forma que as funcionalidades do robô poderiam ser expandidas ou até mesmo as aplicações didáticas ficariam mais atrativas;
- Apesar de haver o recurso de interface de operação do controlador por meio de dispositivos como *smartphones* ou *tablets*, a utilização de um dispositivo local que possibilite a interação do usuário com o controlador sem a necessidade de uma rede de comunicação externa seria uma alternativa em caso de impossibilidade do uso da rede *wireless*.

Por fim, o uso do controlador em arquitetura aberta possibilita o compartilhamento de informações que incentivam o estudo de alternativas de sistemas robóticos semelhantes ao abordado, bem como a aplicação de melhorias e trabalhos colaborativos ao projeto desenvolvido.

REFERÊNCIAS

- AMARAL, P. S. **Lider - Conceção e Implementação de um Laboratório Integrado para Desenvolvimento e Ensino da Robótica**. Universidade de São Paulo. São Paulo. 1985.
- ARDUINO. Arduino - Libraries. **Arduino Official Site**, 2017. Disponível em: <<https://www.arduino.cc/en/reference/libraries>>. Acesso em: 23 Junho 2017.
- ARDUINO. Arduino Mega 2560 Rev3. **Arduino Official Site**, 2017. Disponível em: <<http://www.arduino.cc>>. Acesso em: 23 Junho 2017.
- ÅSTRÖM, K. J.; HÄGGLUND, T. **Advanced PID Control**. Durham: ISA-The Instrumentation, Systems, and Automation Society, 2006.
- BARRIENTOS, A. et al. **Fundamentos da Robótica**. 2ª. ed. Madrid: McGraw-Hil, 2007.
- BLYNK. docs.blynk.cc, 2017. Disponível em: <<http://www.blynk.cc>>. Acesso em: 28 Junho 2017.
- BLYNK. docs.blynk.cc. **Blynk Official Site**, 2017. Disponível em: <<http://www.blynk.cc>>. Acesso em: 28 Junho 2017.
- CHAUDHARY, H.; PRASAD, R.; SUKAVANUM, N. Trajectory Tracking Control of Scrobot-ER V plus Robot Manipulator Based on Kinematical Approach. **International Journal of Engineering Science and Technology**, 2012. 1174-1182.
- CIELNIAK, G.; BELLOTTO, N.; DUCKETT, T. Integrating mobile robotics and vision with undergraduate computer science. **IEEE Transactions on Education**, 2012. 48-53.
- CRAIG, J. J. **Introduction to Robotics: mechanisms and control**. 2ª. ed. Massachusetts: Addison-Wesley, 1989.
- DESHPANDE, V. A.; GEORGE, P. M. Analytical Solution for Inverse Kinematics of Scrobot-ER Vplus Robot. **International Journal of Emerging Technology and Advanced Engineering**, 2012. 478-481.
- ESHED ROBOTEC. **Scrobot-ER 4pc User's Manual**. Catalog #100118 Rev.A. ed. [S.l.]: [s.n.], 1999.
- ESPRESSIF. **Espressif Smart Connectivity Platform: ESP8266**. Xangai: [s.n.], 2013.
- ESPRESSIF. ESP8266EX Overview, 2017. Disponível em: <<http://www.espressif.com/>>. Acesso em: 25 Junho 2017.
- FORD, W. What is an open architecture robot controller? **IEEE International Symposium on Intelligent Control**, Columbus, 1994. 27-32.

GEEKCREIT. L298N Dual H-Bridge Motor Driver Board, 2017. Disponível em: <<http://www.geekcreit.com/>>. Acesso em: 25 Junho 2017.

GEMBAROWSKI, A. G.; CAMPOS, A. A. Análise e comparação da geração de trajetórias lineares e cúbicas em um sistema robótico. **E-tech: Tecnologias para Competitividade Industrial**, Florianópolis, v. 8, p. 111-126, 2015.

GIERLAK, P. Adaptive Control of the Scorbot-ER 4pc Manipulator. **RUTMech**, 2014. 501-516.

HIRZINGER, G. et al. Advances in Robotics: The DLR Experience. **The International Journal of Robotic Research**, 1999. 1064-1087.

LAGES, W. F.; BRACARENSE, A. Q. Robot Retrofitting: a perspective to small and medium size enterprises. **Austrian Brazilian Automation Day**, São Bernardo do Campo, 2003. 1-11.

LEITE, A. C. S.; SILVA, P. A. B.; VAZ, A. C. R. A importância das aulas práticas para alunos jovens e adultos: uma abordagem investigativa sobre a percepção dos alunos do PROEF II. **Ensaio Pesquisa em Educação em Ciências**, Belo Horizonte, 2005. 3.

MATLACK, C. Arduino Playground - MatrixMath. **Arduino Official Site**, 2017. Disponível em: <<https://playground.arduino.cc/Code/MatrixMath>>. Acesso em: 23 Junho 2017.

MIRELES, J. R. M. **Controladores de Robots Rígidos: Un Análisis Comparativo entre las Metodologías de Control Clásico, Adaptable y Robusto Basadas en el Método de Lyapunov**. Centro Nacional de Investigación y Desarrollo Tecnológico. Cuernavaca. 2006.

NAGRATH, I. J.; MITTAL, R. K. **Robotics and Control**. 6ª. ed. Nova Deli: Tata McGraw-Hill, 2007.

NIKU, S. B. **Introduction to Robotics: Analysis, systems, applications**. Nova Jersey: Prentice Hall, 2001.

NORIA, R. L.; MUÑOZ, F. G. M.; PIEDRA, S. M. D. L. Implementación de un control de regulación PD + G para la segunda y tercera articulación del robot SCORBOT-ER 4u. **Congreso Mexicano de Robótica**, Mazatlán, 2014.

ON SEMICONDUCTOR. **CD4049UBC/CD4050BC**: Hex Inverting Buffer/Hex Non-Inverting Buffer. Colorado: Literature Distribution Center for ON Semiconductor, 2002.

PRABHU, N. et al. Dynamic Modeling of Scorbot-ER Vu Plus Industrial Robot Manipulator using LabVIEW. **Indian Journal of Science and Technology**, 2016. 1-10.

ROMANI, M. Arduino Playground - SimpleTimer Library. **Arduino Official Site**, 2017. Disponível em: <<https://playground.arduino.cc/Code/SimpleTimer>>. Acesso em: 23 Junho 2017.

ROSS, A. F.; SANTOS, A. H.; ESPINOZA, R. V. **Projeto e implementação de um sistema de geração de trajetórias para o robô Handler da UTFPR**. Universidade Tecnológica Federal do Paraná. Curitiba, p. 117. 2011.

SAHA, S. K. **Introducción a la Robótica**. Noida: McGraw-Hill, 2008.

SANTINI, D. C. **Arquitetura Aberta para Controle de Robôs Manipuladores**. Universidade Federal do Rio Grande do Sul. Porto Alegre. 2009.

SANTOS, V. M. F. **Robotica Industrial**. Aveiro: Departamento de Engenharia Mecânica da Universidade de Aveiro, 2004.

SICILIANO, B. et al. **Robotics: Modelling, planning and control**. Londres: Springer, 2009.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. **Robot Modeling and Control**. 2ª. ed. Nova Iorque: John Wiley & Sons, 2012.

STMICROELECTRONICS. **L298: Dual Full-Bridge Driver**. Agrate Brianza: [s.n.], 2000.

TAPIA, E. J. S. **Robot Control and Programming: Class Notes**. Universidade de Navarra. Navarra. 2010.

TEIXEIRA, J. V. **Impacto de Recursos Computacionais no Processo Ensino-Aprendizagem de Robótica Industrial**. Universidade do Estado de Santa Catarina. Joinville. 2017.

VERMA, A.; DESHPANDE, V. A. End-effector Position Analysis of SCORBOT-ER Vplus Robot. **International Journal of Smart Home**, 2011. 1-6.

WALTER, J. **Rapid Learning in Robotics**. Göttingen: Cuvillier Verlag, 1996.