

**UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

**Alexsander dos Santos Sena**

**Modelo Automatizado de Separador de Objetos**

**São Carlos**

**2024**



**Alexsander dos Santos Sena**

## **Modelo Automatizado de Separador de Objetos**

Monografia apresentada ao Curso de Engenharia Mecatrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Mecatrônico.

Orientadora: Profa. Dra. Maíra Martins da Silva

**São Carlos**

**2024**

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

S474m Sena, Alexsander dos Santos  
Modelo automatizado de separador de objetos /  
Alexsander dos Santos Sena; orientadora Maira Martins  
da Silva. São Carlos, 2024.

Monografia (Graduação em Engenharia Mecatrônica)  
-- Escola de Engenharia de São Carlos da Universidade  
de São Paulo, 2024.

1. Automatização. 2. Separador de objetos. 3.  
Manipulador robótico. 4. Garra robótica. 5.  
Microcontrolador. 6. Esteira transportadora. 7. Sensor  
de distância. 8. Modelo. I. Título.

## FOLHA DE AVALIAÇÃO

**Candidato:** Alexsander dos Santos Sena

**Título:** Modelo Automatizado de Separador de Objetos

Trabalho de Conclusão de Curso apresentado à  
Escola de Engenharia de São Carlos da  
Universidade de São Paulo  
Curso de Engenharia Mecatrônica.

### BANCA EXAMINADORA

Professora Máira Martins da Silva  
(Orientador)

Nota atribuída: 9,5 (nove, cinco)

Máira M. de Silva

(assinatura)

Professor Rodrigo Nicoletti

Nota atribuída: 9,5 (nove, cinco)

Rodrigo Nicoletti

(assinatura)

Dr. Renan Sanches Geronel

Nota atribuída: 9,5 (nove, cinco)

\_\_\_\_\_

(assinatura)

Média: 9,5 (nove, cinco)

Resultado: APROVADO

Data: 24 / 07 / 2024.

Este trabalho tem condições de ser hospedado no Portal Digital da Biblioteca da EESC

SIM  NÃO  Visto do orientador Máira M. de Silva



*Este trabalho é dedicado à minha família, que sempre apoiou e incentivou que meus irmãos e eu estudássemos para mudar nossa realidade socioeconômica. Minha mãe, Neide dos Santos Sena, que trabalhou desde jovem como empregada doméstica, e meu pai, Antônio Sérgio de Sena, que foi operário em quase toda sua vida, fizeram todos os esforços para possibilitar a realização desse sonho que vivenciei. O título de engenheiro que defenderei pertence a eles.*



## **AGRADECIMENTOS**

Aos meus pais, Neide dos Santos Sena e Antônio Sérgio de Sena, que me ajudaram nesse percurso antes e durante a graduação e, que estiveram em meus pensamentos em cada esforço que fiz para realizar meu sonho.

Ao meu irmão, Alex dos Santos Sena, que compartilhou comigo essa jornada como aluno da USP e sempre me serviu de referência desde antes do meu ingresso.

Ao meu irmão, Anderson dos Santos Sena, que custeou os livros para que eu estudasse para o vestibular e sempre me incentivou.

À professora e orientadora desse projeto, Máira Martins da Silva, que me ajudou nessa monografia, assim como ao longo de toda graduação e serve como inspiração e referência de um docente que se preocupa verdadeiramente com seus discentes.

Ao cursinho pré-vestibular FEA-USP e projeto CAPJOVEM que me abriram os horizontes do mundo acadêmico tal como possibilidade de ascensão socioeconômica através dele.



*“Se pude enxergar mais longe,  
foi porque me apoiei em ombros de gigantes.”*

*Isaac Newton*



## RESUMO

SENA, A.S. **Modelo Automatizado de Separador de Objetos**. 2024. 100 p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

A automação, presente tanto no ambiente de trabalho quanto no cotidiano doméstico, desempenha um papel vital na economia moderna ao aumentar a produtividade, reduzir custos e melhorar a qualidade de vida das pessoas. Este projeto teve como objetivo criar um modelo automatizado de separador de objetos para uso industrial, analisando cientificamente cada componente necessário para o processo de automação. As informações e os parâmetros obtidos permitiram uma integração mais precisa, segura e efetiva desses componentes. O cálculo de posição da garra, para cada ângulo dos servomotores do manipulador robótico, determinou o posicionamento da esteira transportadora e da caixa de armazenamento dos objetos. A determinação da velocidade de movimento da garra e da esteira permite uma possível otimização do processo referente à estabilidade do objeto durante a captura. O microcontrolador foi responsável pelas tomadas de decisões em cada instante do processo. A informação sobre a presença de um objeto na esteira e o tamanho desse objeto foi compartilhada com o microcontrolador por meio de sensores de distância, posicionados em lugares específicos na esteira. Para definir o percurso do manipulador robótico dependendo do objeto a ser armazenado, foi elaborada uma programação em C/C++ para o microcontrolador Arduino, visando otimizar a automatização do processo de separação. A simulação de comportamento do circuito elétrico garantiu a segurança e a estabilidade dos componentes que dependem de energia elétrica. Os dados teóricos obtidos mostraram-se compatíveis com os dados experimentais, apesar de pequenas oscilações de tensão nos circuitos elétricos devido à conexão via *plotoboard* ao invés de soldagem. O projeto funcionou adequadamente, separando de forma automatizada diferentes objetos pré-definidos.

Palavras-chave: Modelo. Automatização. Separador de objetos. Esteira transportadora. Manipulador robótico. Garra robótica. Sensor de distância. Microcontrolador. Caixa de redução. Proteção contra surtos elétricos.



## ABSTRACT

SENA, A.S. **Automated Object Separator Model**. 2024. 100 p. Monograph (Conclusion Course Paper) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2024.

Automation, present both in the workplace and in everyday domestic life, plays a vital role in the modern economy by increasing productivity, reducing costs, and improving people's quality of life. This project aimed to create an automated object separator model for industrial use, scientifically analyzing each necessary component for the automation process. The information and parameters obtained allowed for a more precise, safe, and effective integration of these components. The calculation of the claw's position for each angle of the robotic manipulator's servomotors determined the positioning of the conveyor belt and the storage box for the objects. Determining the movement speed of the claw and the conveyor belt allows for possible optimization of the process concerning the stability of the object during capture. The microcontroller was responsible for decision-making at each moment of the process. Information about the presence of an object on the conveyor belt and its size was shared with the microcontroller through distance sensors positioned at specific locations on the belt. To define the path of the robotic manipulator depending on the object to be stored, a program in C/C++ for the Arduino microcontroller was developed to optimize the automation of the separation process. The simulation of the electrical circuit's behavior ensured the safety and stability of the components that depend on electrical energy. The theoretical data obtained were compatible with the experimental data, despite small voltage fluctuations in the electrical circuits due to the connection via breadboard instead of soldering. The project functioned adequately, automatically separating different predefined objects.

**Keywords:** Model. Automation. Object separator. Conveyor belt. Robotic manipulator. Robotic claw. Distance sensor. Microcontroller. Reduction box. Protection against electrical surges.



## LISTA DE FIGURAS

Figura 1 – Desenho de corte da placa de acrílico . . . . .	30
Figura 2 – CAD da garra do manipulador robótico . . . . .	31
Figura 3 – Foto do manipulador Robótico . . . . .	31
Figura 4 – CAD da base do manipulador robótico . . . . .	32
Figura 5 – Mecanismos sobrepostos . . . . .	33
Figura 6 – Esboço do braço do manipulador robótico . . . . .	34
Figura 7 – Esquemático das relações vetoriais do braço . . . . .	34
Figura 8 – Gráfico de $\beta$ em função de $\alpha$ para $\theta$ fixado em diferentes valores . . . .	36
Figura 9 – Gráfico de $\gamma$ em função de $\theta$ para $\alpha$ fixado em diferentes valores . . . .	36
Figura 10 – Posições da garra no plano $xy$ . . . . .	38
Figura 11 – Gráfico de velocidades da garra em função do ângulo $\alpha$ para $\dot{\alpha} = 0.79$ $rad/s$ e $\dot{\theta} = 0 rad/s$ . . . . .	40
Figura 12 – Gráfico da componente horizontal da velocidade da garra em função do espaço ( $\dot{\alpha} = 0.79 rad/s$ e $\dot{\theta} = 0 rad/s$ ) . . . . .	42
Figura 13 – Gráfico da componente vertical da velocidade da garra em função do espaço ( $\dot{\alpha} = 0.79 rad/s$ e $\dot{\theta} = 0 rad/s$ ) . . . . .	42
Figura 14 – Gráfico de velocidades da garra em função do ângulo $\theta$ para $\dot{\alpha} = 0$ $rad/s$ e $\dot{\theta} = 0.79 rad/s$ . . . . .	43
Figura 15 – Gráfico da componente horizontal da velocidade da garra em função do espaço ( $\dot{\alpha} = 0 rad/s$ e $\dot{\theta} = 0.79 rad/s$ ) . . . . .	44
Figura 16 – Gráfico da componente vertical da velocidade da garra em função do espaço ( $\dot{\alpha} = 0 rad/s$ e $\dot{\theta} = 0.79 rad/s$ ) . . . . .	44
Figura 17 – Gráfico de acelerações da garra em função do ângulo $\alpha$ para $\dot{\alpha} = 0.79$ $rad/s$ e $\dot{\theta} = 0 rad/s$ . . . . .	47
Figura 18 – Gráfico de acelerações da garra em função do ângulo $\theta$ para $\dot{\alpha} = 0 rad/s$ e $\dot{\theta} = 0.79 rad/s$ . . . . .	47
Figura 19 – Animação da garra no plano $xy$ para $\dot{\alpha} = 0.79 rad/s$ e $\dot{\theta} = 0 rad/s$ . . . .	48
Figura 20 – Animação da garra no plano $xy$ para $\dot{\alpha} = 0 rad/s$ e $\dot{\theta} = 0.79 rad/s$ . . . .	48
Figura 21 – Esteira transportadora . . . . .	49
Figura 22 – Motor DC . . . . .	50
Figura 23 – Caixa de redução . . . . .	50
Figura 24 – Diagrama da caixa de redução . . . . .	51
Figura 25 – Caixa de Armazenamento . . . . .	52
Figura 26 – Circuito apenas com motor e transistor . . . . .	55
Figura 27 – Simulação do circuito apenas com motor e transistor . . . . .	55
Figura 28 – Circuito com diodo adicionado . . . . .	56

Figura 29 – Simulação do circuito com diodo adicionado . . . . .	57
Figura 30 – Circuito com diodo e capacitor adicionado . . . . .	58
Figura 31 – Simulação do circuito com diodo e capacitor adicionado . . . . .	58
Figura 32 – Módulo sensor de distância . . . . .	62
Figura 33 – Posição do sensor de distância . . . . .	63
Figura 34 – Arduino UNO . . . . .	64
Figura 35 – Pinos de conexão do Arduino UNO . . . . .	65
Figura 36 – Placa Shield v5.0 . . . . .	66
Figura 37 – Fluxograma da automatização . . . . .	69

## LISTA DE QUADROS

Quadro 1 – Especificações técnicas do Arduino UNO . . . . .	64
Quadro 2 – Conexão dos componetes com o Arduino para controle manual . . . .	66
Quadro 3 – Conexão dos componetes com o Arduino para controle automatizado .	69



## LISTA DE ABREVIATURAS E SIGLAS

2D	Duas dimensões
3D	Três dimensões
NPN	Negativo-positivo-negativo
BJT	Bipolar Junction Transistor (Transistor bipolar de junção)
FET	Field-Effect Transistor (Transistor de Efeito de Campo)
IDE	Ambiente de desenvolvimento integrado
USB	Universal Serial Bus (Barramento serial universal)
PWM	Pulse Width Modulation (Modulação por Largura de Pulso)
DC	Direct current (corrente contínua)
USP	Universidade de São Paulo
RPM	Rotação por minuto



## LISTA DE SÍMBOLOS

mm	milímetros
cm	centímetros
m	metros
s	segundos
rad	radianos
A	Amperes
mA	miliamperes
V	Volts
Hz	Hertz
MHz	megahertz
F	Farad
$\mu F$	microfarad
nF	nanofarad
H	Henry
$\mu H$	microhenry
kB	quilobytes
k	quilo
n	nano
$\Omega$	Ohms
$\Delta$	variação
$\Delta S$	variação de espaço
$\Delta t$	variação de tempo
$\Delta t_{high}$	variação de tempo em nível lógico alto
$v$	velocidade linear

$a$	aceleração linear
$v_{som}$	velocidade do som propagado no ar
$\omega$	velocidade angular
R	raio
kg	quilograma
kgf	quilograma-força
$\alpha$	ângulo alpha
$\beta$	ângulo beta
$\gamma$	ângulo gamma
$\theta$	ângulo theta
$\phi$	ângulo phi
$\dot{\alpha}$	velocidade angular alpha
$\dot{\theta}$	velocidade angular theta
$\dot{\phi}$	velocidade angular phi
$\ddot{\alpha}$	aceleração angular alpha
$\ddot{\theta}$	aceleração angular theta
$\ddot{\phi}$	aceleração angular phi
$\vec{r}$	vetor posição
$\frac{\partial}{\partial y}$	derivada parcial em relação a y
hFE	Ganho de corrente em curto-circuito

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>25</b>
<b>2</b>	<b>MECÂNICA</b>	<b>29</b>
<b>2.1</b>	<b>Manipulador Robótico</b>	<b>30</b>
2.1.1	Montagem da garra	31
2.1.2	Análise de posição	31
2.1.2.1	BASE	32
2.1.2.2	BRAÇO	33
2.1.3	Análise de velocidade	38
2.1.3.1	BASE	39
2.1.3.2	BRAÇO	39
2.1.4	Análise de aceleração	45
2.1.4.1	BASE	45
2.1.4.2	BRAÇO	46
<b>2.2</b>	<b>Esteira Transportadora</b>	<b>49</b>
2.2.1	Motor	49
2.2.2	Caixa de redução	50
<b>2.3</b>	<b>Caixa de armazenamento</b>	<b>52</b>
<b>3</b>	<b>ELETRÔNICA</b>	<b>53</b>
<b>3.1</b>	<b>Fonte de alimentação</b>	<b>53</b>
<b>3.2</b>	<b>Transistor</b>	<b>53</b>
<b>3.3</b>	<b>Motor</b>	<b>54</b>
<b>3.4</b>	<b>Diodo</b>	<b>56</b>
<b>3.5</b>	<b>Capacitor</b>	<b>57</b>
<b>3.6</b>	<b>Resistor</b>	<b>59</b>
<b>4</b>	<b>CONTROLE</b>	<b>61</b>
<b>4.1</b>	<b>Sensor de distância</b>	<b>61</b>
4.1.1	Funcionamento	61
4.1.2	Posicionamento dos sensores	62
<b>4.2</b>	<b>Microcontrolador</b>	<b>63</b>
4.2.1	Shield	65
4.2.2	Controle manual	66
4.2.3	Controle automatizado – Automatização	68
<b>5</b>	<b>CONCLUSÃO</b>	<b>77</b>

<b>REFERÊNCIAS</b> . . . . .	<b>79</b>
<b>APÊNDICES</b>	<b>81</b>
<b>APÊNDICE A – CÓDIGO NO MATLAB DA POSIÇÃO DA GARRA</b>	<b>83</b>
<b>APÊNDICE B – CÓDIGO NO MATLAB DA VELOCIDADE E ACE- LERAÇÃO DA GARRA</b> . . . . .	<b>91</b>

## 1 INTRODUÇÃO

O início da automação remonta há séculos atrás, oferecendo muitas tecnologias que mudaram a forma de vida da sociedade como conhecemos hoje. Apesar de ter sido elemento fundamental na Revolução Industrial, se engana quem acredita que foi nesse período que surgiu o conceito de automação.

Podemos elencar algumas tecnologias precursoras que se utilizavam de recursos e fenômenos da natureza para substituir tarefas repetitivas e/ou intensivas realizadas por esforço humano, como os moinhos de vento e água, por exemplo. Moinhos de vento e água, embora não sejam tecnologias eletrônicas, foram alguns dos primeiros sistemas automatizados desenvolvidos pela humanidade. Eles foram projetados para realizar tarefas como moagem de grãos ou bombeamento de água de forma contínua, sem a necessidade de intervenção humana constante, ainda no século II a.C. na Ásia Menor para o moinho de água e século II d.C. na Europa para o moinho de vento.

No final do século XVIII e início do século XIX, a Revolução Industrial marcou o surgimento das máquinas movidas a vapor, teares mecânicos, máquinas de fiar e outras inovações que substituíram muitos processos manuais e aumentaram drasticamente a produção industrial.

No século XX, houve avanços significativos na automação de tarefas específicas. Por exemplo, a linha de montagem de Henry Ford para a produção de automóveis introduziu técnicas de produção em massa, automatizando muitos segmentos do processo de fabricação.

Nas décadas de 1940 e 1950, o desenvolvimento de máquinas de controle numérico computadorizado (CNC) permitiu a automação de processos de fabricação controlados por computador. Isso possibilitou a produção de peças complexas com alta precisão, reduzindo a dependência de habilidades artesanais.

A partir da década de 1960, os robôs industriais começaram a ser introduzidos nas linhas de produção. Esses robôs eram capazes de realizar tarefas específicas, como soldagem, pintura e montagem, de forma precisa e repetitiva.

Com o avanço da tecnologia da informação, houve uma tendência crescente de integrar sistemas de automação em toda a fábrica. Isso incluiu o desenvolvimento de sistemas de controle mais sofisticados, software de gerenciamento de produção e sistemas de comunicação em rede. (Burns; Lerner; Meacham, 2000)

Na era atual a inteligência artificial, o aprendizado de máquina e a Internet das Coisas (IoT) possibilitam a criação de sistemas de automação mais inteligentes e adaptáveis.

Isso inclui robôs colaborativos, que podem trabalhar lado a lado com humanos de forma segura e sistemas de produção flexíveis que podem se adaptar rapidamente a mudanças nas demandas do mercado.

O desenvolvimento da automação é um processo contínuo, impulsionado pela busca por maior eficiência, qualidade e flexibilidade na produção industrial. Ao longo dos séculos, vimos uma evolução significativa desde as primeiras máquinas movidas a vapor até a automação avançada e inteligente dos dias atuais.

Alguns pontos-chave que podem ser ressaltados sobre sua importância no meio industrial são mostrados a seguir.

**Aumento da eficiência:** A automação permite a execução de tarefas repetitivas de forma rápida e precisa, reduzindo o tempo necessário para completar processos industriais. Isso resulta em maior produtividade e uso mais eficiente de recursos.

**Melhoria da qualidade:** A automação pode ajudar a reduzir erros humanos e variações na produção, levando a uma maior consistência e qualidade dos produtos fabricados. Sistemas automatizados podem monitorar e controlar variáveis com maior precisão do que os operadores humanos, garantindo padrões de qualidade mais elevados.

**Redução de custos:** Embora o investimento inicial em sistemas de automação possa ser significativo, a longo prazo, a automação pode reduzir os custos operacionais. Isso ocorre devido à redução do desperdício, menor necessidade de mão de obra e maior eficiência energética.

**Segurança do trabalhador:** Automatizar tarefas perigosas ou monótonas pode proteger os trabalhadores de ambientes de trabalho arriscados. Além disso, a automação pode ser usada para implementar sistemas de segurança mais robustos, como dispositivos de detecção de falhas e paradas de emergência.

**Flexibilidade e adaptabilidade:** Os sistemas automatizados podem ser reprogramados e reconfigurados com relativa facilidade para lidar com mudanças na demanda do mercado, introdução de novos produtos ou processos aprimorados. Isso proporciona às empresas maior flexibilidade para se adaptarem às condições em constante mudança.

**Competitividade no mercado global:** Empresas que adotam a automação em seus processos industriais muitas vezes conseguem produzir produtos de alta qualidade a custos mais baixos, o que as torna mais competitivas em um mercado global cada vez mais exigente.

Nesse sentido, o objetivo deste projeto é realizar a montagem de um separador de objetos automatizado aproveitando-se as tecnologias vigentes do contexto atual.

Os objetivos específicos desse projeto são:

- Compreender e explorar o funcionamento de dispositivos de sensoriamento que podem ajudar na implementação da automatização;
- Compreender e explorar os conceitos e funcionamento de circuitos e componentes elétricos como, motores DC, servomotores, resistores, capacitores, transistores, diodos assim como suas diferentes disposições dentro do circuito;
- Compreender e explorar os conceitos e funcionamento dos dispositivos mecânicos como, engrenagens, motores, redutores, esteiras e configurações entre barras (links) que formam um braço robótico;
- Compreender e explorar os conceitos e funcionamento dos microcontroladores (nesse caso, o Arduino UNO) assim como suas linguagens de programação para a comunicação entre a máquina e o ser humano;
- Projetar um modelo de baixo custo;
- Integrar as diferentes áreas da ciência (elétrica, mecânica e computação) para se obter um maior nível de autonomia do sistema, como vemos ao longo da evolução;
- Compreender e explorar os conceitos e funcionamento de um braço robótico e de uma esteira elétrica integrados em um sistema automatizado.

Um modelo de determinada tecnologia pode ser representado de forma miniaturizada em relação ao projeto real. Assim seguiu nesse caso. Não houve menção de escala visto que o objetivo deste projeto foi testar o modelo.



## 2 MECÂNICA

A mecânica desempenha um papel crucial no desenvolvimento e operação de manipuladores robóticos, comumente conhecidos como braços robóticos, e esteiras transportadoras. Esses sistemas são amplamente utilizados em diversas indústrias, desde a manufatura automotiva até a logística, devido à sua capacidade de realizar tarefas complexas com alta precisão, eficiência e repetibilidade.

A mecânica de um braço robótico envolve a análise e o *design* de seus componentes estruturais, articulações, atuadores e sistemas de transmissão de movimento. Cada um desses elementos deve ser cuidadosamente projetado para garantir que o manipulador possa operar de maneira eficiente e segura, atendendo às exigências específicas de sua aplicação. A compreensão dos princípios mecânicos subjacentes é crucial para a otimização do desempenho do robô, permitindo que ele execute tarefas de manipulação com rapidez, precisão e robustez.

O estudo da mecânica aplicada a manipuladores robóticos abrange várias disciplinas, incluindo cinemática, dinâmica e estática. A cinemática trata da descrição dos movimentos do robô, enquanto a dinâmica se concentra nas forças e torques envolvidos nesses movimentos. A estática, por sua vez, examina o equilíbrio das forças em condições de repouso ou movimento constante. Neste capítulo, foi abordado o estudo da cinemática.

As esteiras transportadoras são componentes essenciais em linhas de produção e sistemas logísticos. Elas são projetadas para mover materiais e produtos de um ponto a outro de maneira contínua e eficiente. A mecânica dessas esteiras envolve o *design* de suas estruturas, sistemas de acionamento, roletes e superfícies de transporte. A escolha adequada dos materiais e o dimensionamento correto dos componentes mecânicos são fundamentais para garantir a durabilidade e a eficiência do sistema.

A integração de manipuladores robóticos e esteiras transportadoras em um sistema de produção automatizado oferece inúmeras vantagens. A mecânica desses dois componentes deve ser cuidadosamente coordenada para garantir uma operação harmoniosa e eficiente. Por exemplo, a precisão e a velocidade dos braços robóticos precisam ser compatíveis com o ritmo de movimentação das esteiras, permitindo a manipulação precisa de peças e produtos ao longo da linha de produção.

A sinergia entre manipuladores robóticos e esteiras transportadoras resulta em sistemas altamente eficientes e flexíveis, capazes de atender às demandas de produção de maneira ágil e adaptável. A mecânica avançada desses sistemas contribui para a melhoria contínua dos processos industriais, aumentando a produtividade, reduzindo custos e garantindo a qualidade dos produtos finais.

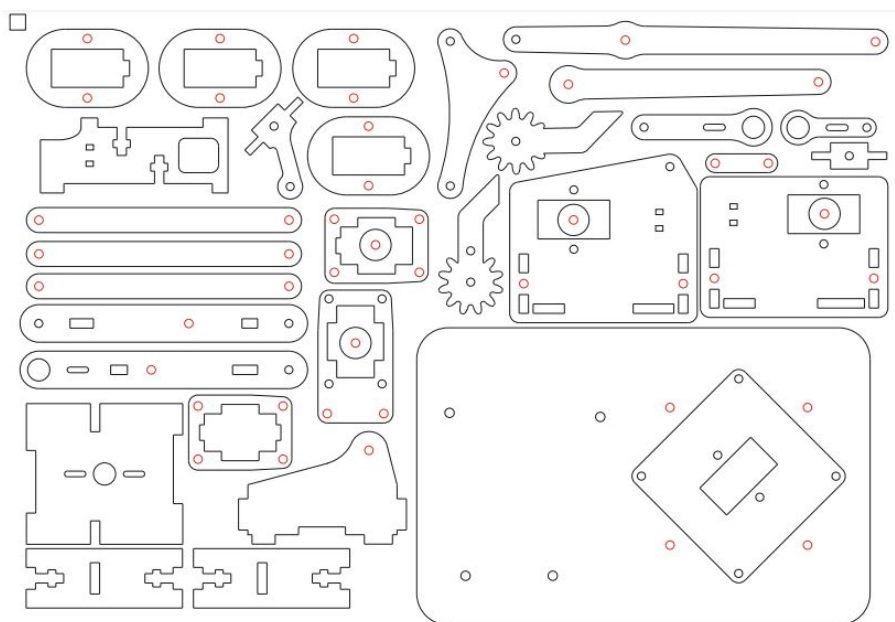
Portanto, a mecânica não só define a base estrutural e funcional de manipuladores robóticos e esteiras transportadoras, mas também influencia diretamente sua capacidade de desempenho e adaptabilidade a diferentes tarefas. O avanço contínuo na mecânica desses sistemas é essencial para o desenvolvimento de soluções robóticas mais avançadas e versáteis, capazes de atender às demandas cada vez mais complexas da indústria moderna.

## 2.1 Manipulador Robótico

Um dos componentes mais importantes — ou o mais importante — é o manipulador robótico. O manipulador robótico é um robô antropomórfico – recebe esse nome devido à semelhança da configuração de suas juntas e elos com a anatomia de um braço humano. É uma categoria de máquina amplamente usada na indústria, principalmente em processos de transposição de cargas, pinturas e soldagens. Nesta planta industrial, ele será responsável por retirar o objeto a ser separado da esteira e levá-lo até a caixa de objetos grandes ou a caixa de objetos pequenos, a depender do tamanho do objeto.

Usaremos como referência e base o projeto de manipulador robótico MeArm V0.4 da phenoptix (Gray, 2014), que consiste em peças menores de acrílico em 2D montadas entre si para compor o manipulador robótico em 3D com a ajuda de 4 servomotores. As peças menores de acrílico podem ser cortadas a laser a partir de uma placa de acrílico de 3mm de espessura, seguindo o design de corte como mostra a Figura 1.

Figura 1 – Desenho de corte da placa de acrílico

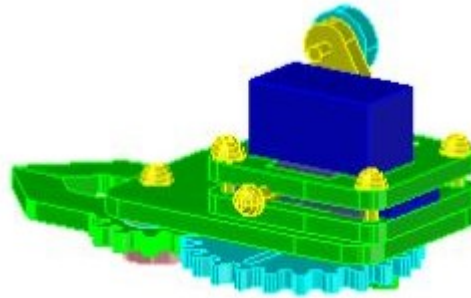


Fonte: Ben Gray (2014)

### 2.1.1 Montagem da garra

Neste projeto foi usado um manipulador robótico similar ao da Figura 1, porém com uma diferença no mecanismo de abertura e fechamento de garra, o qual será feito por um sistema de engrenagens responsáveis pela transmissão entre o servomotor e a garra, tal como na Figura 2.

Figura 2 – CAD da garra do manipulador robótico



Fonte: Saravati (2023a,p.1)

Realizada a montagem do manipulador robótico segundo as instruções do manual de montagem (Saravati, 2023a), chegou-se ao resultado mostrado na Figura 3.

Figura 3 – Foto do manipulador Robótico



Fonte: Saravati (2023b)

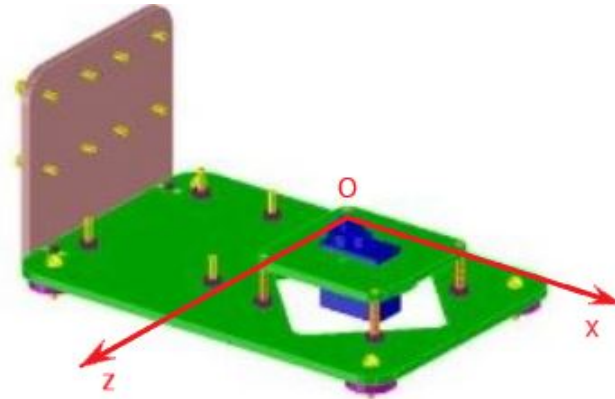
### 2.1.2 Análise de posição

O manipulador robótico possui 4 graus de liberdade, sendo eles a translação em  $x$  e  $y$  controlada pelos servomotores do braço, a rotação em torno do eixo  $y$  controlada pelo servomotor da base e mais um grau de liberdade associado à abertura e fechamento da garra no plano  $xz$  controlada pelo servomotor da garra. Apesar do manipulador robótico atuar no espaço, em 3 dimensões, foi possível analisar sua atuação em 2 planos ortogonais entre si no espaço — plano  $xy$  e plano  $xz$ . Para cada um desses planos, há partes diferentes do mecanismo que atuam.

### 2.1.2.1 BASE

A primeira, definida no plano  $xz$ , é composta pela base do manipulador robótico. Há um servomotor na base que a rotaciona em  $180^\circ$  ao longo do plano  $xz$ , Figura 4, por consequência, rotaciona a garra  $180^\circ$  ao longo do plano  $xz$ .

Figura 4 – CAD da base do manipulador robótico



Fonte: Edição do autor<sup>1</sup>

A posição da garra, em função ao mecanismo da base, é dado pelo próprio ângulo aplicado pelo servomotor da base. O servomotor da base rotaciona no eixo  $y$  e varre diferentes posições no plano  $xz$ , conforme as equações 2.1 e 2.2. Supondo a localização do servomotor como origem do sistema, pôde-se encontrar a posição da garra em função do ângulo  $\phi$ , como mostrado nas equações 2.1 e 2.2.

$$x_o = d \cdot \sin(\phi) \quad (2.1)$$

$$z_o = d \cdot \cos(\phi) \quad (2.2)$$

onde

$d$ : é a projeção da distância entre a garra e a origem do sistema no plano  $xz$ ;

$\phi$ : é o ângulo que o servomotor percorre;

$x_o$ : é a posição da garra na coordenada  $x$  em relação ao ponto de referência  $O$ , mostrado na Figura 4;

$z_o$ : é a posição da garra na coordenada  $z$  em relação ao ponto de referência  $O$ , mostrado na Figura 4.

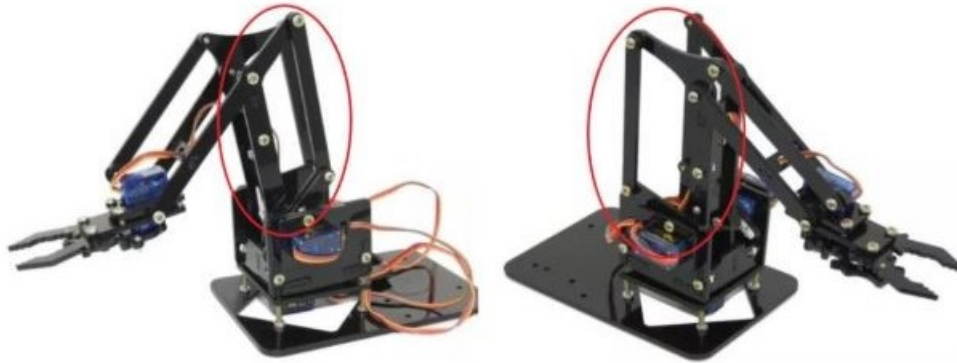
Dessa forma, sendo conhecido a projeção no plano  $xz$  da distância entre a garra e a origem do sistema, variável  $d$ , e também o ângulo  $\phi$  em que o servomotor da base encontra-se, é possível saber as coordenadas  $x_o$  e  $z_o$  do efetuador do manipulador robótico (garra).

<sup>1</sup> Edição feita a partir de imagens pertencentes ao manual de montagem (Saravati, 2023a),p.2

### 2.1.2.2 BRAÇO

A segunda parte, denominada como braço, definida no plano  $xy$ , é composta por dois mecanismos 4 barras de cadeia fechada sobrepostos entre si nesse mesmo plano, como mostra a Figura 5.

Figura 5 – Mecanismos sobrepostos



Fonte: Edição do autor<sup>2</sup>

Para facilitar a análise no plano  $xy$ , o ponto A, Figura 6, que é a posição onde os servomotores braço atuam, foi estabelecido como origem do plano. Essa mudança de referencial torna a análise mais simplificada para determinar as posições da garra.

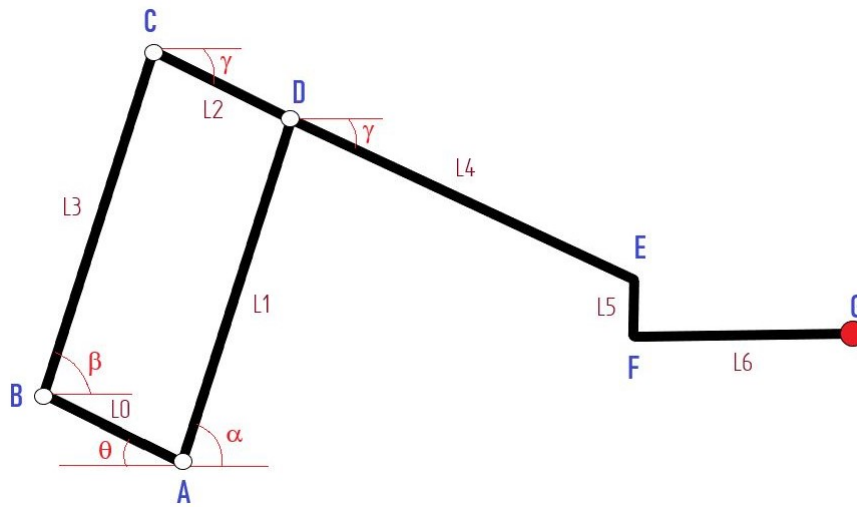
O plano  $xy$  é ortogonal ao plano  $xz$  e rotaciona no eixo  $y$  variando segundo o ângulo  $\phi$ . Logo, em uma análise em três dimensões, o plano  $xy$  pode se tornar o plano  $zy$  dependendo do ângulo  $\phi$ . Aqui, por facilidade de análise, ele foi tratado em duas dimensões como plano  $xy$ .

Os dois servomotores do braço são necessários para definir a posição da garra no plano  $xy$ . É possível analisar a sobreposição dos mecanismos como um único mecanismo 4 barras de cadeia fechada com 2 motores que se alternam na atuação. Ou seja, das 4 barras existentes no mecanismo, 2 barras específicas — barras que têm o motor acoplado — se alternam entre si como barra de atuação (ou barra motor). Enquanto uma estabelece-se momentaneamente como barra de atuação, a outra barra fica definida como ground do mecanismo, ou seja, fica fixada em uma única posição no espaço, sem sofrer alteração enquanto a barra motor movimenta-se no tempo. As outras 2 barras restantes movimentam-se em função do movimento da barra motor no plano espacial.

Foi realizado uma análise de posição do mecanismo descrito acima a partir do esboço mostrado na Figura 6, onde  $\theta$  e  $\alpha$  são ângulos aplicados pelos servomotores, logo  $\overline{AB}$  e  $\overline{AD}$  são barras de atuação (ou barras motoras). Para  $L_0 = L_2 = 3.5$  centímetros,  $L_1 = L_3 = L_4 = 8.1$  centímetros,  $L_5 = 1.2$  centímetros e  $L_6 = 5.3$  centímetros.

<sup>2</sup> Edição a partir de imagens pertencentes à BR Eletrônica (c2016)

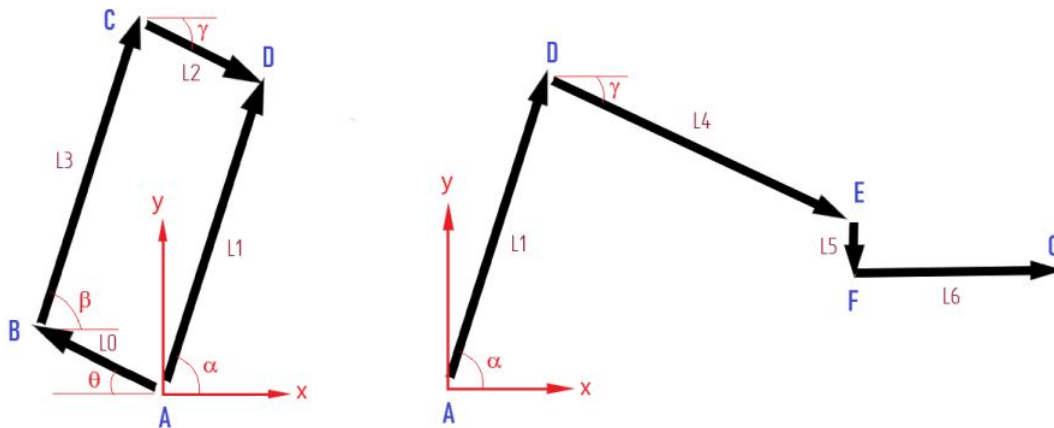
Figura 6 – Esboço do braço do manipulador robótico



Fonte: Elaborado pelo autor

Através do esboço da Figura 6 foi possível particionar o braço em dois mecanismos para realizar uma análise separada. O primeiro mecanismo composto pelas juntas A, B, C e D, de cadeia fechada, enquanto o segundo, formado pelas juntas A, D, E, F e G, de cadeia aberta. A barra  $\overline{EF}$  sempre permanece na vertical tal qual a barra  $\overline{FG}$  sempre permanece na horizontal, por atuação de um mecanismo adjacente.

Figura 7 – Esquemático das relações vetoriais do braço



Fonte: Elaborado pelo autor

A partir da relação vetorial do mecanismo composto pelas juntas A, B, C e D, foi possível determinar a equação vetorial fechada deste mecanismo, como mostra a equação 2.3.

$$\vec{r}_{AB} + \vec{r}_{BC} + \vec{r}_{CD} - \vec{r}_{AD} = 0 \quad (2.3)$$

Decompondo os vetores  $\vec{r}_{AB}$ ,  $\vec{r}_{BC}$ ,  $\vec{r}_{CD}$  e  $\vec{r}_{AD}$  nas componentes das coordenadas  $x$

e  $y$ , obtém-se:

$$\vec{r}_{AB} = \begin{cases} -L_0 \cdot \cos\theta \\ L_0 \cdot \sin\theta \end{cases} \quad (2.4)$$

$$\vec{r}_{BC} = \begin{cases} L_3 \cdot \cos\beta \\ L_3 \cdot \sin\beta \end{cases} \quad (2.5)$$

$$\vec{r}_{CD} = \begin{cases} L_2 \cdot \cos\gamma \\ -L_2 \cdot \sin\gamma \end{cases} \quad (2.6)$$

$$\vec{r}_{AD} = \begin{cases} L_1 \cdot \cos\alpha \\ L_1 \cdot \sin\alpha \end{cases} \quad (2.7)$$

Substituindo as componentes na equação 2.3 chegou-se às equações 2.8 e 2.9.

$$-L_0 \cdot \cos(\theta) + L_3 \cdot \cos(\beta) + L_2 \cdot \cos(\gamma) - L_1 \cdot \cos(\alpha) = 0 \quad (2.8)$$

$$L_0 \cdot \sin(\theta) + L_3 \cdot \sin(\beta) - L_2 \cdot \sin(\gamma) - L_1 \cdot \sin(\alpha) = 0 \quad (2.9)$$

Resolveu-se esse sistema de equações utilizando o método iterativo de Newton-Raphson através do MATLAB para obter-se  $\beta$  e  $\gamma$ . O método de Newton-Raphson é mostrado na equação 2.10.

$$\begin{Bmatrix} \beta_{k+1} \\ \gamma_{k+1} \end{Bmatrix} = \begin{Bmatrix} \beta_k \\ \gamma_k \end{Bmatrix} - \begin{bmatrix} \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ \frac{\partial f_2}{\partial \beta} & \frac{\partial f_2}{\partial \gamma} \end{bmatrix}^{-1} \cdot \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} \quad (2.10)$$

onde

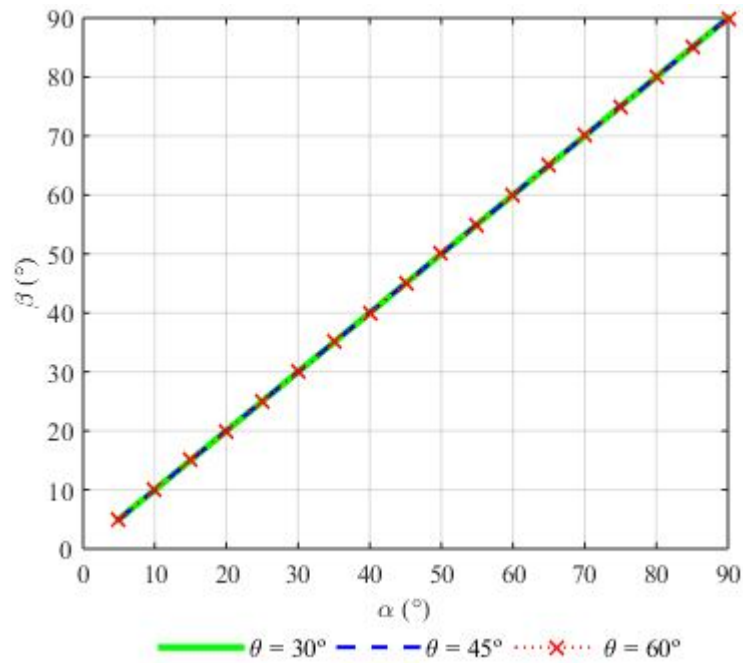
$f_1$  é igual à expressão à esquerda da equação 2.8 quando se tenta encontrar  $\beta$  e  $\gamma$  que faça  $f_1 \rightarrow 0$ ;

$f_2$  é igual à expressão à esquerda da equação 2.9 quando se tenta encontrar  $\beta$  e  $\gamma$  que faça  $f_2 \rightarrow 0$ .

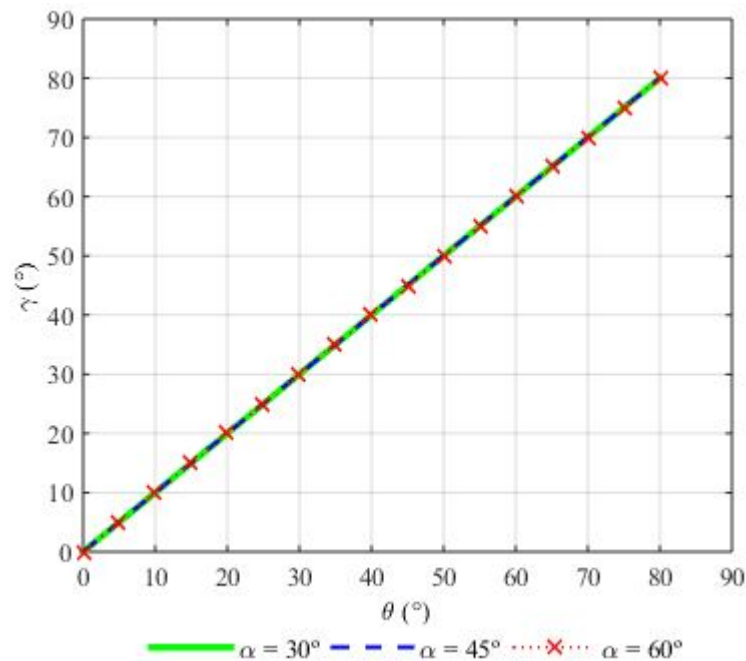
Dessa forma, o sistema de equações na forma matricial 2.10 fica como o sistema de equações na forma matricial 2.11.

$$\begin{Bmatrix} \beta_{k+1} \\ \gamma_{k+1} \end{Bmatrix} = \begin{Bmatrix} \beta_k \\ \gamma_k \end{Bmatrix} - \begin{bmatrix} -L_3 \cdot \sin\beta & -L_2 \cdot \sin\gamma \\ L_3 \cdot \cos\beta & -L_2 \cdot \cos\gamma \end{bmatrix}^{-1} \cdot \begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} \quad (2.11)$$

Colocando a equação no MATLAB, variando  $0^\circ \leq \theta \leq 80^\circ$  e  $5^\circ \leq \alpha \leq 135^\circ$  com discretização de  $5^\circ$  e erro de  $10^{-6}$ , foi possível gerar os gráficos da variação de  $\beta$  e  $\gamma$ , em função da variação de  $\alpha$  e  $\theta$ , conforme mostrado nas Figuras 8 e Figura 9.

Figura 8 – Gráfico de  $\beta$  em função de  $\alpha$  para  $\theta$  fixado em diferentes valores

Fonte: Elaborado pelo autor

Figura 9 – Gráfico de  $\gamma$  em função de  $\theta$  para  $\alpha$  fixado em diferentes valores

Fonte: Elaborado pelo autor

Observando o gráfico da Figura 8, constatou-se que, independente dos valores de  $\theta$ ,  $\beta$  assume os mesmos valores de  $\alpha$  para o sistema dimensionado como um paralelogramo ( $L_0 = L_2$  e  $L_1 = L_3$ ). Em outras palavras, nesse mecanismo,  $\beta = \alpha$ .

O análogo acontece com  $\gamma$  e  $\theta$ , fixando qualquer valor de  $\alpha$ . Portanto,  $\gamma = \theta$  como foi possível observar no gráfico da Figura 9.

Foi escolhido  $\alpha$  a partir de  $5^\circ$  pois, por limitações mecânicas, é impossível que  $\theta = 0^\circ$  ao mesmo tempo que  $\alpha = 0^\circ$ . Como  $\theta = 0^\circ$  é muito mais usado nas aplicações desta análise, optou-se por deixar  $\alpha$  a partir de  $5^\circ$ .

Uma vez que foram encontrados  $\beta$  e  $\gamma$ , pôde-se determinar a posição da garra no plano  $x_A y_A$  a partir da equação 2.12 gerada através da relação de vetores demonstrada na Figura 7.

$$\vec{r}_G = \vec{r}_{AD} + \vec{r}_{DE} + \vec{r}_{EF} + \vec{r}_{FG} \quad (2.12)$$

De forma análoga ao mecanismo anterior, pôde-se decompor os vetores  $\vec{r}_{AD}$ ,  $\vec{r}_{DE}$ ,  $\vec{r}_{EF}$ ,  $\vec{r}_{FG}$  e  $\vec{r}_G$  em suas componentes  $x$  e  $y$ .

$$\vec{r}_{DE} = \begin{cases} L_4 \cdot \cos\gamma \\ -L_4 \cdot \sin\gamma \end{cases} \quad (2.13)$$

$$\vec{r}_{EF} = \begin{cases} 0 \\ -L_5 \end{cases} \quad (2.14)$$

$$\vec{r}_{FG} = \begin{cases} L_6 \\ 0 \end{cases} \quad (2.15)$$

Substituindo em 2.12, chegou-se nas seguintes equações:

$$x = L_1 \cdot \cos(\alpha) + L_4 \cdot \cos(\gamma) + L_6 \quad (2.16)$$

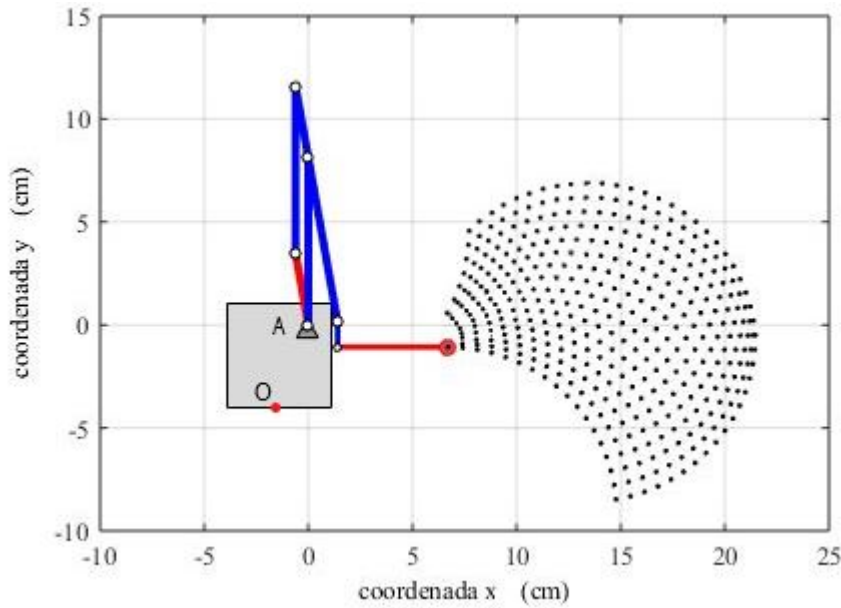
$$y = L_1 \cdot \sin(\alpha) - L_4 \cdot \sin(\gamma) - L_5 \quad (2.17)$$

Como  $\gamma = \theta$ , chegou-se nas equações 2.18 e 2.19.

$$x = L_1 \cdot \cos(\alpha) + L_4 \cdot \cos(\theta) + L_6 \quad (2.18)$$

$$y = L_1 \cdot \sin(\alpha) - L_4 \cdot \sin(\theta) - L_5 \quad (2.19)$$

Sendo  $x$  e  $y$  coordenadas com origem no ponto  $A$ , a partir das equações 2.18 e 2.19, pôde-se plotar o gráfico de posição  $x$  e  $y$  como mostra a Figura 10.

Figura 10 – Posições da garra no plano  $xy$ 

Fonte: Elaborado pelo autor

O ponto  $A$  está a um raio de 2 centímetros de distância do ponto  $O$  no plano  $xz$ . Portanto, a projeção da distância entre a garra e ponto  $O$  no plano  $xz$ , foi expresso pela variável  $d_G$ , como mostra a equação 2.20.

$$d = x + 2 \quad (2.20)$$

Dessa forma, substituindo o valor de  $d_G$  nas equações 2.1 e 2.2:

$$x_o = (x + 2) \cdot \sin(\phi) \quad (2.21)$$

$$z_o = (x + 2) \cdot \cos(\phi) \quad (2.22)$$

O ponto  $A$  também está a 4.5 centímetros de distância do ponto  $O$  na coordenada  $y$ , logo:

$$y_o = y + 4.5 \quad (2.23)$$

### 2.1.3 Análise de velocidade

O perfil de velocidade de um manipulador robótico deve ser cuidadosamente planejado para minimizar o tempo de ciclo sem sacrificar a precisão. Isto envolve a definição de trajetórias de movimento otimizadas que levam em conta as limitações dinâmicas do robô, como aceleração e desaceleração máximas, e a necessidade de evitar ressonâncias mecânicas que possam causar vibrações indesejadas. A análise de velocidade permite ajustar esses perfis para alcançar um equilíbrio ideal entre rapidez e precisão.

Para manipuladores robóticos, a análise de velocidade começa com a cinemática, podendo ser direta ou inversa. A cinemática direta envolve a determinação da posição

e orientação da extremidade do robô (ou "efetuador final") com base nas velocidades articulares, enquanto a cinemática inversa trata do cálculo das velocidades articulares necessárias para alcançar uma velocidade específica do efetuador final. Neste projeto, foi realizado uma análise de velocidades através da cinemática direta. Este processo é crucial para garantir que o robô possa se mover de maneira suave e precisa, evitando movimentos bruscos que poderiam comprometer a segurança e a precisão da operação.

### 2.1.3.1 BASE

Para encontrar a velocidade com que a garra move-se em função do movimento da base, foi necessário derivar no tempo as equações 2.21, 2.22 e 2.23, como mostrado nas equações 2.24, 2.25 e 2.26.

$$v_{x/o} = v_x \cdot \sin(\phi) + (x + 2) \cdot \dot{\phi} \cdot \cos(\phi) \quad (2.24)$$

$$v_{z/o} = v_x \cdot \cos(\phi) - (x + 2) \cdot \dot{\phi} \cdot \sin(\phi) \quad (2.25)$$

$$v_{y/o} = v_y \quad (2.26)$$

Como os servos motores nunca atuam simultaneamente, mas sim alternadamente, pôde-se separar as velocidades da garra apenas em função do movimento da base enquanto o braço fica estagnado ( $v_x = 0$  e  $v_y = 0$ ), como mostrado nas seguintes equações:

$$v_{x/o} = (x + 2) \cdot \dot{\phi} \cdot \cos(\phi) \quad (2.27)$$

$$v_{z/o} = -(x + 2) \cdot \dot{\phi} \cdot \sin(\phi) \quad (2.28)$$

$$v_{y/o} = 0 \quad (2.29)$$

onde

$$\dot{\phi} = 0.79 \frac{rad}{s}$$

$$[x] = cm$$

$$[v_{x/o}] = [v_{z/o}] = [v_{y/o}] = \frac{cm}{s}$$

Da mesma forma, para encontrar as velocidades da garra em relação à origem da base, porém com a base estagnada ( $\dot{\phi} = 0$ ) e os servomotores do braço atuando, encontrou-se:

$$v_{x/o} = v_x \cdot \sin(\phi) \quad (2.30)$$

$$v_{z/o} = v_x \cdot \cos(\phi) \quad (2.31)$$

$$v_{y/o} = v_y$$

### 2.1.3.2 BRAÇO

Para encontrar a velocidade com que a garra se move em função do movimento do braço, foi necessário derivar no tempo as equações 2.18 e 2.19, que representam as

equações de posição da garra em função do braço. Supondo que o servomotor da base permanece parado ( $\dot{\phi} = 0$ ), enquanto os servomotores do braço atuam alternadamente entre si, obtiveram-se as equações 2.32 e 2.33.

$$v_x = -\dot{\alpha} \cdot L_1 \cdot \sin(\alpha) - \dot{\theta} \cdot L_4 \cdot \sin(\theta) \quad (2.32)$$

$$v_y = \dot{\alpha} \cdot L_1 \cdot \cos(\alpha) - \dot{\theta} \cdot L_4 \cdot \cos(\theta) \quad (2.33)$$

Como os dois os servomotores do braço atuam alternadamente, para  $\dot{\theta} = 0$  obteve-se:

$$v_x = -\dot{\alpha} \cdot L_1 \cdot \sin\alpha \quad (2.34)$$

$$v_y = \dot{\alpha} \cdot L_1 \cdot \cos\alpha \quad (2.35)$$

O módulo da velocidade tangencial à trajetória pode ser expresso pela equação 2.36.

$$|\vec{v}| = \sqrt{v_x^2 + v_y^2} \quad (2.36)$$

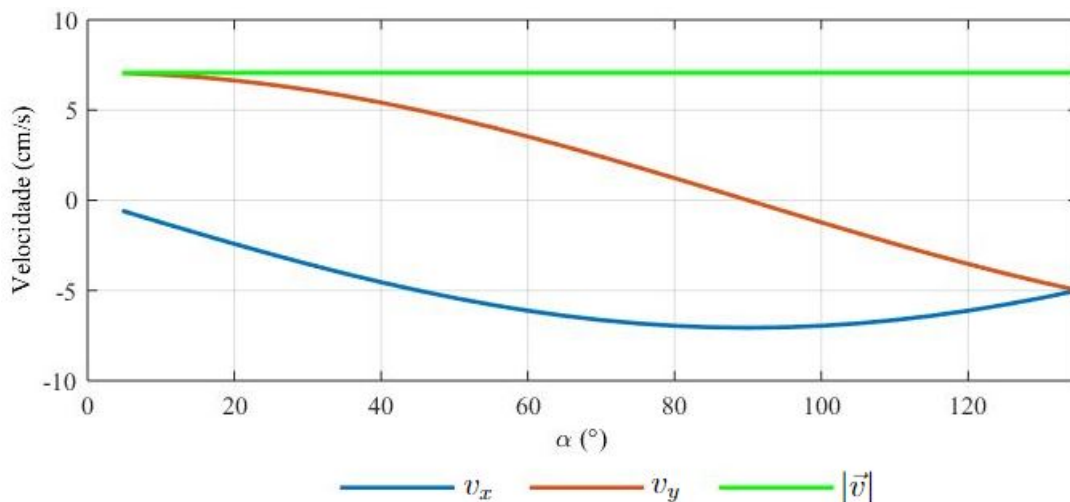
onde

$$\dot{\alpha} = 0.79 \frac{\text{rad}}{\text{s}}$$

$$[v_x] = [v_y] = [v] = \frac{\text{cm}}{\text{s}}$$

A partir das equações 2.34, 2.35 e 2.36, foi possível representar seus valores graficamente no MATLAB<sup>3</sup>, como mostra a Figura 11. Percebe-se que as velocidades independem de  $\theta$ , como denotado nas equações 2.34 e 2.35.

Figura 11 – Gráfico de velocidades da garra em função do ângulo  $\alpha$  para  $\dot{\alpha} = 0.79 \text{ rad/s}$  e  $\dot{\theta} = 0 \text{ rad/s}$



Fonte: Elaborado pelo autor

<sup>3</sup> MATLAB é um ambiente de programação e computação numérica usado por engenheiros e cientistas para analisar dados, desenvolver algoritmos e criar modelos.

Através do gráfico mostrado na Figura 11, foi possível verificar que a variação dos valores de  $v_x$  e  $v_y$  em função de  $\alpha$ , mantém  $\vec{v}$  em velocidade uniforme. Isto acontece para um mecanismo dimensionado como um paralelogramo e  $\dot{\alpha}$  em velocidade uniforme.

Os mecanismos de quatro barras dimensionados como paralelogramos são configurações de mecanismos que mantêm uma relação geométrica específica, na qual dois pares de barras opostas são iguais em comprimento e paralelas entre si. Esta configuração é conhecida por suas propriedades de transmissão de movimento e forças de maneira uniforme e previsível.

Um mecanismo de quatro barras é composto por quatro membros (ou barras) e quatro juntas (ou pivôs). Quando estas barras são dimensionadas de modo a formar um paralelogramo, as propriedades geométricas do paralelogramo conferem ao mecanismo características distintas.

**Barras Opostas Iguais e Paralelas:** Em um paralelogramo, cada par de barras opostas é igual em comprimento e paralelo entre si. Isso significa que se uma barra é movida, a barra oposta se moverá de maneira congruente, mantendo sempre a mesma orientação angular.

**Movimento de Translação Pura:** Uma das propriedades mais notáveis do mecanismo de quatro barras em forma de paralelogramo é a capacidade de produzir movimento de translação pura. Isso é especialmente útil em aplicações onde é necessário mover um componente de forma linear sem rotação.

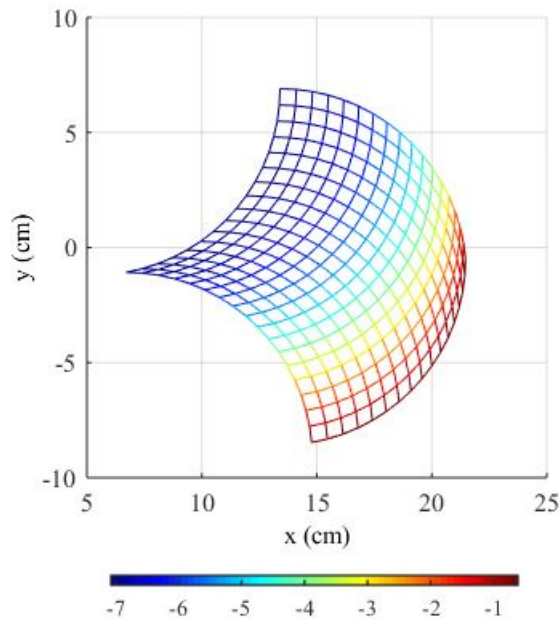
**Estabilidade e Rigidez:** Devido à simetria e à distribuição de forças em um paralelogramo, o mecanismo tende a ser estável e rígido. Isso o torna adequado para aplicações onde a precisão e a repetibilidade do movimento são cruciais.

Quando o mecanismo que possui o efetuador (garra) é fixado a um mecanismo de 4 barras dimensionado como paralelogramo, a característica de movimento de translação pura é preservada. Entretanto, como existem dois servomotores, a rotação do objeto poderia acontecer dependendo do servomotor acionado. Isso não acontece devido a um mecanismo 4 barras adicional acoplado na garra que garante que o objeto permaneça na vertical independentemente do servomotor que estiver atuando.

Esse mecanismo 4 barras adjacente, não interfere na velocidade de deslocamento da garra e exerce apenas uma função de "equilíbrio" do objeto.

Com as equações acima listadas e o uso do MATLAB, mapeou-se as velocidades  $v_x$  e  $v_y$  em cada ponto do trajeto da garra para  $0^\circ \leq \theta \leq 80^\circ$  e  $5^\circ \leq \alpha \leq 90^\circ$  no plano  $xy$ . Dessa forma, foi possível identificar a velocidade de captura do objeto na esteira, nas coordenadas  $x = 17.6$  centímetros e  $y = -0.1$  centímetros em relação ao ponto  $A$ , como mostrado nas Figuras 12 e 13.

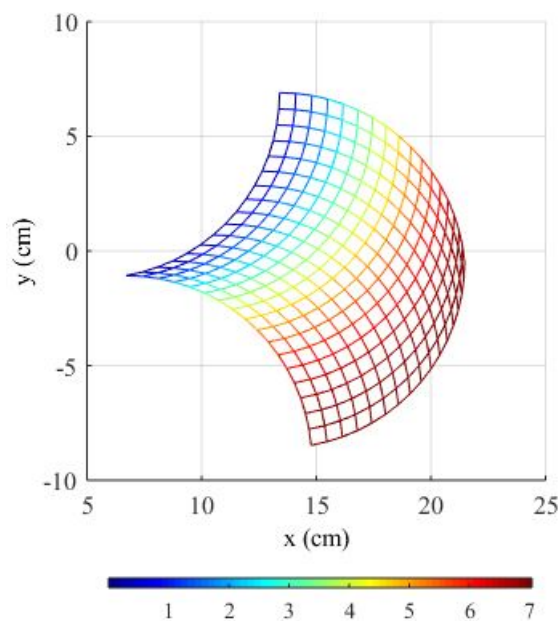
Figura 12 – Gráfico da componente horizontal da velocidade da garra em função do espaço ( $\dot{\alpha} = 0.79 \text{ rad/s}$  e  $\dot{\theta} = 0 \text{ rad/s}$ )



Fonte: Elaborado pelo autor

O gráfico mostrado na Figura 12, é uma representação em duas dimensões ( $x$  e  $y$ ) de um gráfico de três dimensões ( $x$ ,  $y$  e  $v_x$ ), onde as diferentes cores representam os diferentes valores de  $v_x$ .

Figura 13 – Gráfico da componente vertical da velocidade da garra em função do espaço ( $\dot{\alpha} = 0.79 \text{ rad/s}$  e  $\dot{\theta} = 0 \text{ rad/s}$ )



Fonte: Elaborado pelo autor

Analogamente ao gráfico mostrado na Figura 12, o gráfico mostrado na Figura 13 tem diferentes cores representando diferentes valores de  $v_y$ .

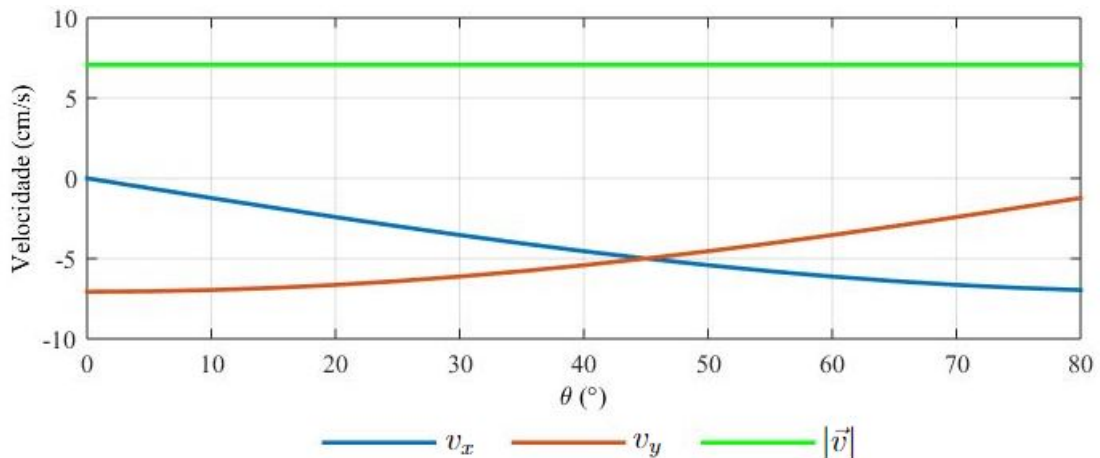
Realizando o mesmo procedimento para  $\dot{\alpha} = 0 \text{ rad/s}$  e  $\dot{\theta} = 0.79 \text{ rad/s}$ , encontrou-se as 2.37 e 2.38.

$$v_x = -\dot{\theta} \cdot L_4 \cdot \sin\theta \quad (2.37)$$

$$v_y = -\dot{\theta} \cdot L_4 \cdot \cos\theta \quad (2.38)$$

A partir das equações 2.37, 2.38 e 2.36, foi possível representar seus valores graficamente no MATLAB, como mostra a Figura 14. Percebe-se que as velocidades independem de  $\alpha$ , como denotado nas equações 2.37 e 2.38.

Figura 14 – Gráfico de velocidades da garra em função do ângulo  $\theta$  para  $\dot{\alpha} = 0 \text{ rad/s}$  e  $\dot{\theta} = 0.79 \text{ rad/s}$



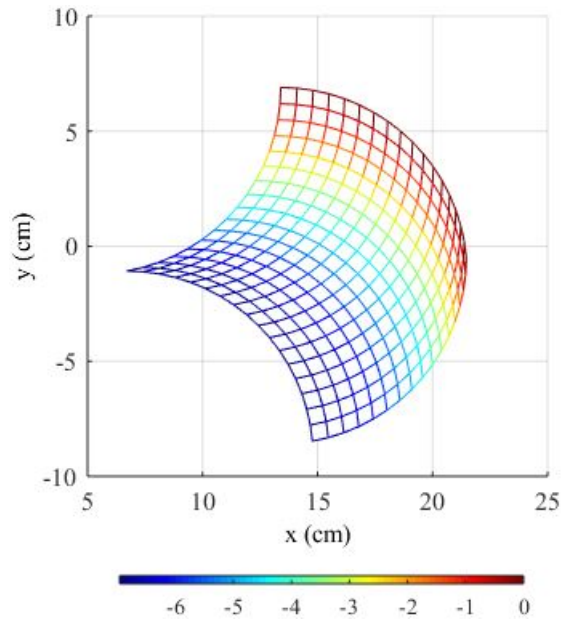
Fonte: Elaborado pelo autor

Através do gráfico da Figura 14, foi possível verificar que mesmo que  $v_x$  e  $v_y$  variem seus valores em função de  $\theta$ , sua composição  $\vec{v}$  mantém velocidade uniforme. Isto acontece para o mecanismo dimensionado como um paralelogramo, onde  $\dot{\theta}$  mantém-se em velocidade uniforme.

As velocidades  $v_x$  e  $v_y$  podem mudar de positivo para negativo dependendo se a trajetória for no sentido do ponto  $A$  ou no sentido oposto. Independentemente do sentido do percurso, o módulo de  $v_x$  e  $v_y$  permanecem os mesmos.

A partir dessas informações, foi possível mapear as velocidades  $v_x$  e  $v_y$  em cada ponto do trajeto da garra para  $0^\circ \leq \theta \leq 80^\circ$  e  $5^\circ \leq \alpha \leq 90^\circ$  no plano  $xy$ . Dessa forma, foi possível identificar a velocidade de captura do objeto na esteira, nas coordenadas  $x = 17.6$  centímetros e  $y = -0.1$  centímetros, em relação ao ponto  $A$ , como mostrado nos gráficos das Figuras 15 e 16.

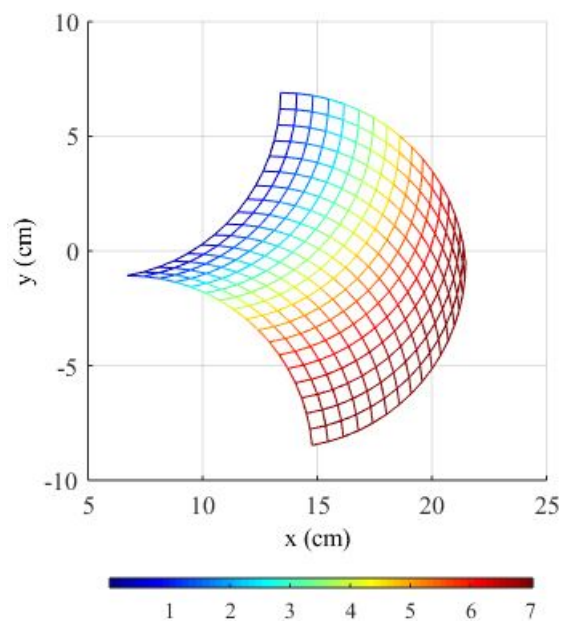
Figura 15 – Gráfico da componente horizontal da velocidade da garra em função do espaço ( $\dot{\alpha} = 0 \text{ rad/s}$  e  $\dot{\theta} = 0.79 \text{ rad/s}$ )



Fonte: Elaborado pelo autor

Analogamente ao gráfico da Figura 12, o gráfico da Figura 15 e o gráfico da Figura 16 têm diferentes cores representando diferentes valores de  $v_x$  e  $v_y$ .

Figura 16 – Gráfico da componente horizontal da velocidade da garra em função do espaço ( $\dot{\alpha} = 0 \text{ rad/s}$  e  $\dot{\theta} = 0.79 \text{ rad/s}$ )



Fonte: Elaborado pelo autor

### 2.1.4 Análise de aceleração

Os perfis de aceleração devem ser planejados para evitar mudanças bruscas, conhecidas como "jerk" (taxa de variação da aceleração), que podem causar choques mecânicos. Perfis de aceleração suave ajudam a distribuir as forças de maneira mais uniforme, reduzindo o impacto nos componentes mecânicos e melhorando a precisão dos movimentos. A otimização desses perfis é essencial para garantir que o robô possa operar de maneira eficiente e segura, mesmo em aplicações de alta velocidade.

#### 2.1.4.1 BASE

Para encontrar a aceleração da garra em função do movimento da base, derivou-se no tempo as equações 2.24, 2.25 e 2.26, chegando nas seguintes equações:

$$a_{x/o} = a_x \cdot \sin(\phi) + 2 \cdot v_x \cdot \dot{\phi} \cdot \cos(\phi) + (x + 2) \cdot (\ddot{\phi} \cdot \cos(\phi) - \dot{\phi}^2 \cdot \sin(\phi)) \quad (2.39)$$

$$a_{z/o} = a_x \cdot \cos(\phi) - 2 \cdot v_x \cdot \dot{\phi} \cdot \sin(\phi) - (x + 2) \cdot (\ddot{\phi} \cdot \sin(\phi) + \dot{\phi}^2 \cdot \cos(\phi)) \quad (2.40)$$

$$a_{y/o} = a_y \quad (2.41)$$

Todos os servomotores atuam com velocidade angular uniforme,  $\ddot{\phi} = 0$ . Logo, as equações acima ficam:

$$a_{x/o} = a_x \cdot \sin(\phi) + 2 \cdot v_x \cdot \dot{\phi} \cdot \cos(\phi) - (x + 2) \cdot \dot{\phi}^2 \cdot \sin(\phi) \quad (2.42)$$

$$a_{z/o} = a_x \cdot \cos(\phi) - 2 \cdot v_x \cdot \dot{\phi} \cdot \sin(\phi) - (x + 2) \cdot \dot{\phi}^2 \cdot \cos(\phi) \quad (2.43)$$

$$a_{y/o} = a_y$$

Os servomotores nunca atuam simultaneamente, mas sim alternadamente. Logo, pôde-se separar as acelerações da garra apenas em função do movimento da base enquanto o braço fica estagnado ( $v_x = v_y = 0$  e  $a_x = a_y = 0$ ):

$$a_{x/o} = -(x + 2) \cdot \dot{\phi}^2 \cdot \sin(\phi) \quad (2.44)$$

$$a_{z/o} = -(x + 2) \cdot \dot{\phi}^2 \cdot \cos(\phi) \quad (2.45)$$

$$a_{y/o} = 0 \quad (2.46)$$

onde

$$\dot{\phi} = 0.79 \frac{\text{rad}}{\text{s}}$$

$$[a_{x/o}] = [a_{z/o}] = [a_{y/o}] = \frac{\text{cm}}{\text{s}^2}$$

Da mesma forma, para encontrar as acelerações da garra em relação à origem da base, com a base estagnada ( $\dot{\phi} = 0$  e  $\ddot{\phi} = 0$ ) e os servomotores do braço atuando,

encontrou-se:

$$a_{x/o} = a_x \cdot \sin(\phi) \quad (2.47)$$

$$a_{z/o} = a_x \cdot \cos(\phi) \quad (2.48)$$

$$a_{y/o} = a_y$$

#### 2.1.4.2 BRAÇO

Para encontrar a aceleração com que a garra se move em função do movimento do braço, foi necessário derivar no tempo as equações 2.32 e 2.33, que representam as equações de velocidade da garra em função do braço. Supondo que o servomotor da base permaneça parado ( $\dot{\phi} = 0$  e  $\ddot{\phi} = 0$ ), enquanto os servomotores do braço atuam alternadamente entre si, obtiveram-se as equações 2.49 e 2.50:

$$a_x = -\ddot{\alpha} \cdot L_1 \cdot \sin(\alpha) - \dot{\alpha}^2 \cdot L_1 \cdot \cos(\alpha) - \ddot{\theta} \cdot L_4 \cdot \sin(\theta) - \dot{\theta}^2 \cdot L_4 \cdot \cos(\theta) \quad (2.49)$$

$$a_y = \ddot{\alpha} \cdot L_1 \cdot \cos(\alpha) - \dot{\alpha}^2 \cdot L_1 \cdot \sin(\alpha) - \ddot{\theta} \cdot L_4 \cdot \cos(\theta) + \dot{\theta}^2 \cdot L_4 \cdot \sin(\theta) \quad (2.50)$$

Os servomotores atuam com velocidade angular uniforme,  $\ddot{\alpha} = 0$  e  $\ddot{\theta} = 0$ . Logo, as equações acima ficam:

$$a_x = -\dot{\alpha}^2 \cdot L_1 \cdot \cos(\alpha) - \dot{\theta}^2 \cdot L_4 \cdot \cos(\theta) \quad (2.51)$$

$$a_y = -\dot{\alpha}^2 \cdot L_1 \cdot \sin(\alpha) + \dot{\theta}^2 \cdot L_4 \cdot \sin(\theta) \quad (2.52)$$

Como os dois os servomotores do braço atuam alternadamente, para  $\dot{\theta} = 0 \text{ rad/s}$  e  $\dot{\alpha} = 0.8726 \text{ rad/s}$  obteve-se:

$$a_x = -\dot{\alpha}^2 \cdot L_1 \cdot \cos(\alpha) \quad (2.53)$$

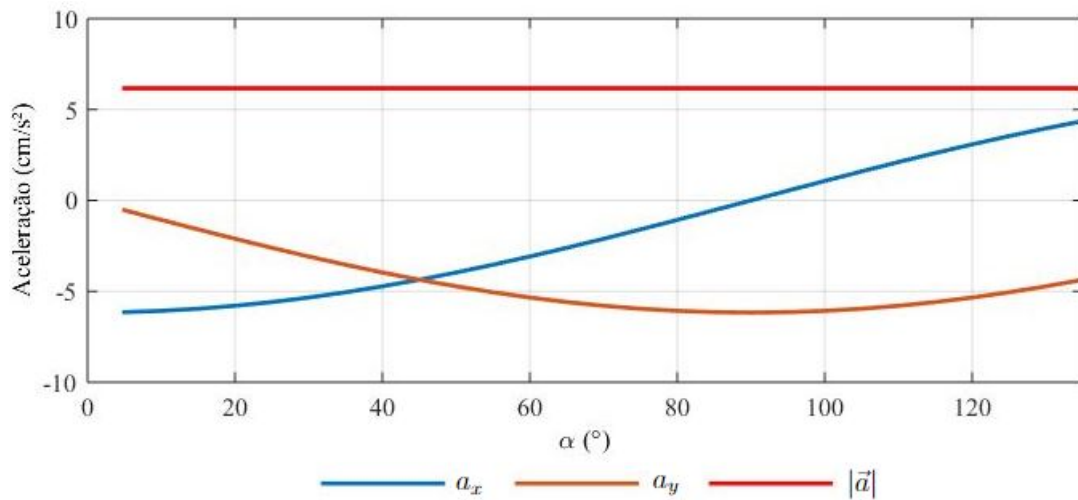
$$a_y = -\dot{\alpha}^2 \cdot L_1 \cdot \sin(\alpha) \quad (2.54)$$

Para encontrar o módulo da aceleração radial à trajetória, usou-se a seguinte equação:

$$|\vec{a}| = \sqrt{a_x^2 + a_y^2} \quad (2.55)$$

A partir das equações 2.53, 2.54 e 2.55, foi possível representar os valores de  $a_x$ ,  $a_y$  e  $|a|$  graficamente no MATLAB, como mostra o gráfico da Figura 17. Percebe-se que as acelerações independem de  $\theta$ , como denotado nas equações 2.53 e 2.54.

Figura 17 – Gráfico de acelerações da garra em função do ângulo  $\alpha$  para  $\dot{\alpha} = 0.79 \text{ rad/s}$  e  $\dot{\theta} = 0 \text{ rad/s}$



Fonte: Elaborado pelo autor

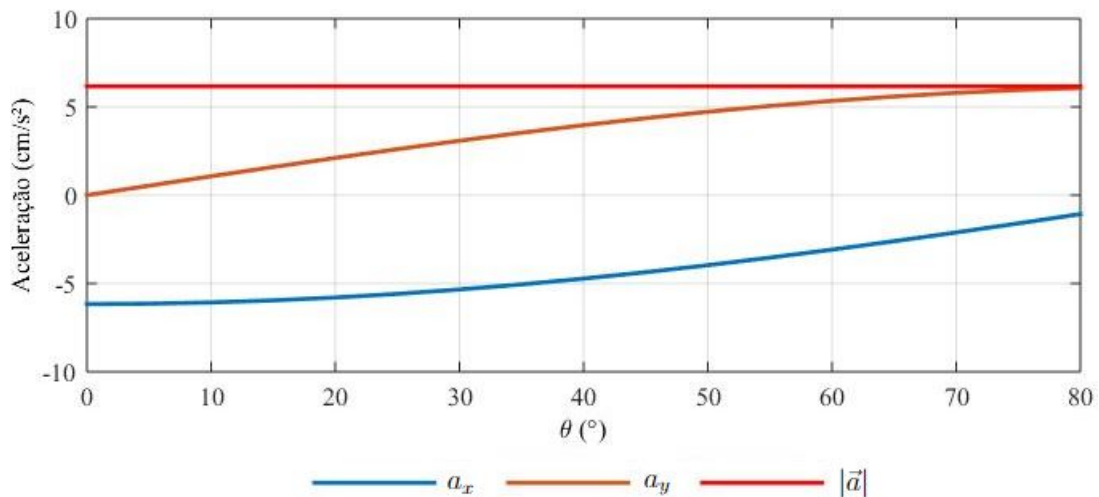
Realizando o mesmo procedimento para  $\dot{\alpha} = 0 \text{ rad/s}$  e  $\dot{\theta} = 0.79$ , encontrou-se as equações 2.56 e 2.57.

$$a_x = -\dot{\theta}^2 \cdot L_4 \cdot \cos(\theta) \quad (2.56)$$

$$a_y = \dot{\theta}^2 \cdot L_4 \cdot \sin(\theta) \quad (2.57)$$

A partir das equações 2.56, 2.57 e 2.55, foi possível representar seus valores graficamente no MATLAB, como mostra a Figura 18. Percebe-se que as acelerações independem de  $\alpha$ , como denotado nas equações 2.56 e 2.57.

Figura 18 – Gráfico de acelerações da garra em função do ângulo  $\theta$  para  $\dot{\alpha} = 0 \text{ rad/s}$  e  $\dot{\theta} = 0.79 \text{ rad/s}$



Fonte: Elaborado pelo autor

Através do gráfico mostrado na na Figura 17 e na Figura 18, foi possível verificar que mesmo que  $a_x$  e  $a_y$  variem seus valores em função de  $\alpha$  ou em função de  $\theta$ , seu módulo,  $|\vec{a}|$ , mantém aceleração radial uniforme. Essa aceleração age na direção radial da trajetória, de forma centrípeta.

Obtida as velocidades e acelerações da garra em função do braço, pôde-se efetuar uma animação da atuação da velocidade e da aceleração durante as trajetórias do braço, como mostrado na Figura 19 e na Figura 20.

Figura 19 – Animação da garra no plano  $xy$  para  $\dot{\alpha} = 0.79 \text{ rad/s}$  e  $\dot{\theta} = 0 \text{ rad/s}$

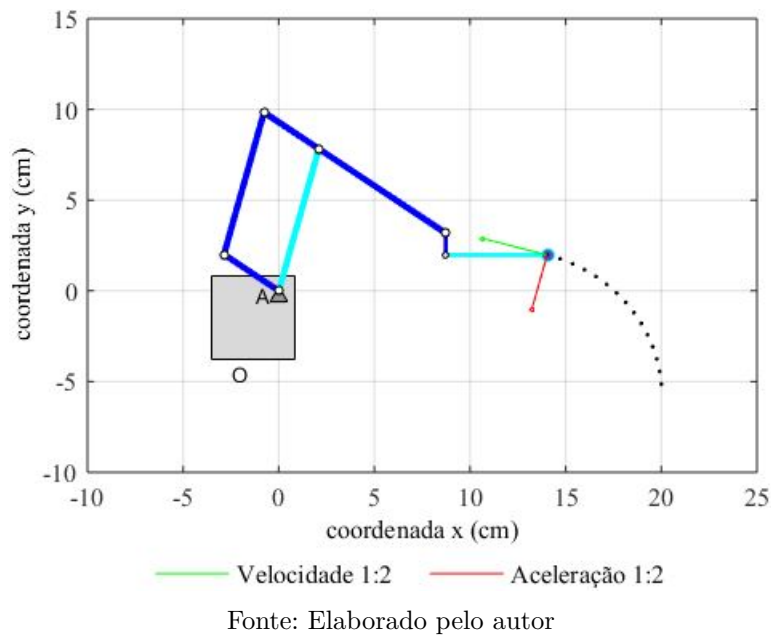
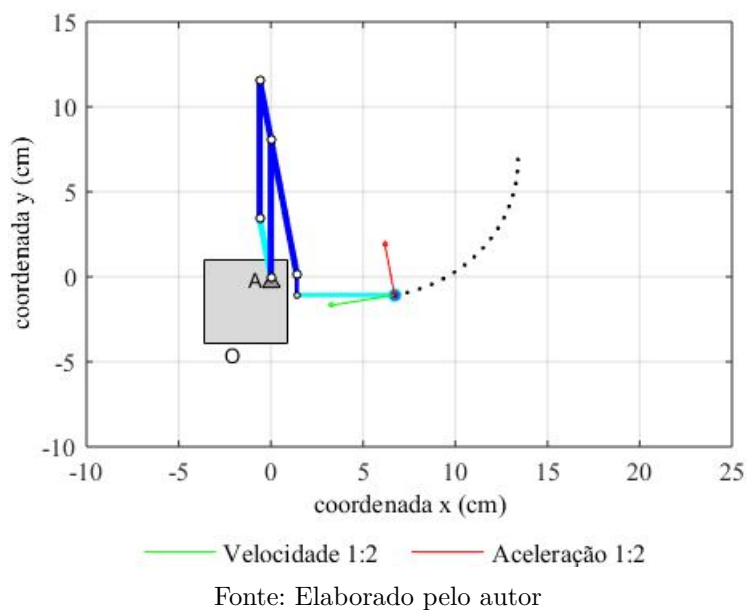


Figura 20 – Animação da garra no plano  $xy$  para  $\dot{\alpha} = 0 \text{ rad/s}$  e  $\dot{\theta} = 0.79 \text{ rad/s}$



## 2.2 Esteira Transportadora

A esteira transportadora tem papel fundamental no processo industrial no mundo e na história. Muito utilizada a partir da revolução industrial em meados do século XVIII, foi a máquina que conferiu viabilidade à implementação do fordismo nas indústrias no início do século XX, sendo muito usada até os tempos atuais. Nesta planta em específico, ela foi responsável por levar o objeto a ser separado até o manipulador robótico.

A esteira transportadora é composta a partir de pelo menos duas roldanas cilíndricas nas extremidades, uma correia conectando-as e um motor conectado a uma dessas roldanas para transferir movimento ao sistema. O mecanismo transforma movimento angular do motor em movimento linear para a esteira.

A partir deste conceito, a esteira deste projeto foi feita com roldanas de material plástico e as correias de material de borracha, como mostra a Figura 21.

Figura 21 – Esteira transportadora

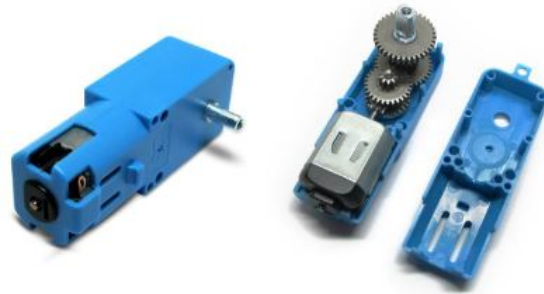


Fonte: Elaborado pelo autor

### 2.2.1 Motor

O motor escolhido para atuar transferindo movimento angular para a esteira, foi um motor elétrico de *corrente contínua* (DC) que atua de 3 a 6 *volts* de tensão. Esse motor tem uma caixa redutora interna, com redução de 90:1 que produz uma velocidade angular de 60 *RPM* (6.28 *rad/s*) e torque de 4 *kgf · cm* a 3 *volts*, mostrado na Figura 22.

Figura 22 – Motor DC



Fonte: Robocore (c2008)

Calculou-se através da equação 2.58, a velocidade linear que a esteira atingiria se fosse ligada diretamente ao motor, para  $R_E = 2.5 \text{ cm}$  e  $\omega_E = -6.28 \text{ rad/s}$ .

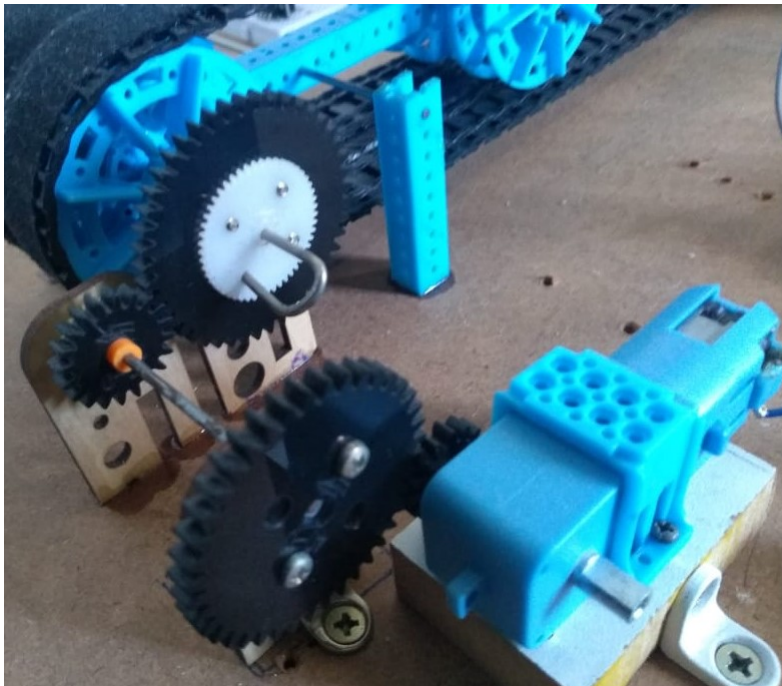
$$v = \omega \cdot R \quad (2.58)$$

Encontrou-se  $v_E = -15.7 \text{ cm/s}$  o que é um valor muito elevado para a captura da garra pelo manipulador robótico utilizado neste projeto.

### 2.2.2 Caixa de redução

Para que a esteira atingi-se uma velocidade adequada de funcionamento sem perder o torque necessário para transportar os objetos, foi necessário a construção de uma caixa de redução externa entre o motor e a esteira, como mostra a Figura 23.

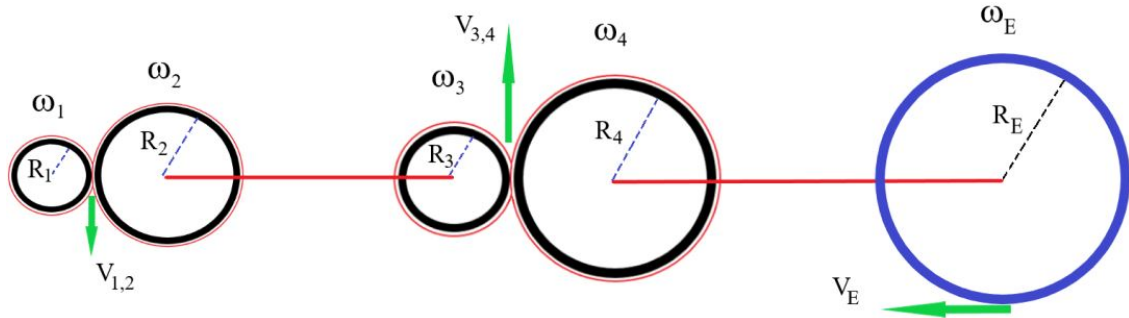
Figura 23 – Caixa de redução



Fonte: Elaborado pelo autor

Foi possível representar esse sistema de engrenagens a partir de um diagrama mostrado na Figura 24, onde a engrenagem 1 foi acoplada diretamente ao motor, portanto esta engrenagem tem mesma velocidade angular do motor ( $\omega_1 = 6.28 \text{ rad/s}$ )

Figura 24 – Diagrama da caixa de redução



Fonte: Elaborado pelo autor

Pela figura 24 foi possível constatar que  $v_1 = v_2$ . Substituindo através da equação 2.58, encontrou-se:

$$\omega_1 \cdot R_1 = \omega_2 \cdot R_2 \Rightarrow \omega_2 = \omega_1 \cdot \frac{R_1}{R_2} \quad (2.59)$$

Pelo diagrama, também foi possível averiguar que  $\omega_2 = \omega_3$ . Substituindo em 2.59, chegou-se na seguinte equação:

$$\omega_3 = \omega_1 \cdot \frac{R_1}{R_2} \quad (2.60)$$

Como  $v_3 = v_4$

$$\omega_3 \cdot R_3 = \omega_4 \cdot R_4 \Rightarrow \omega_4 = \omega_3 \cdot \frac{R_3}{R_4} \quad (2.61)$$

Substituindo 2.60 em 2.61:

$$\omega_4 = \omega_1 \cdot \frac{R_1}{R_2} \cdot \frac{R_3}{R_4} \quad (2.62)$$

Novamente através do diagrama, verifica-se que  $\omega_4 = \omega_E$ . Portanto, substituindo essa variável:

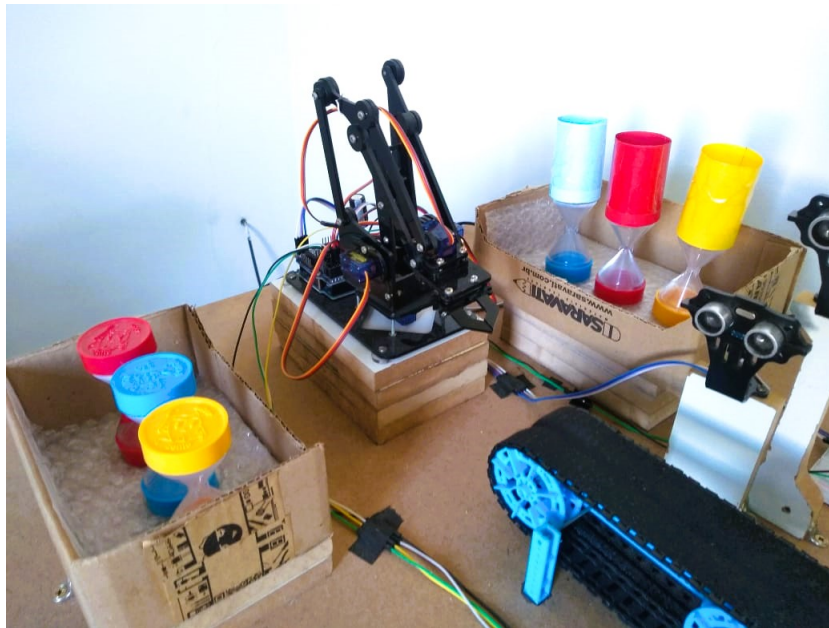
$$\omega_E = \omega_1 \cdot \frac{R_1}{R_2} \cdot \frac{R_3}{R_4} \quad (2.63)$$

Para  $R_1 = 0.75 \text{ cm}$ ,  $R_2 = 2 \text{ cm}$ ,  $R_3 = 1 \text{ cm}$ ,  $R_4 = 2.25 \text{ cm}$  e  $\omega_1 = 6.28 \text{ rad/s}$ , chegou-se a  $\omega_E = 1.05 \text{ rad/s}$  e, usando a equação 2.58 com  $R_E = 2.5 \text{ cm}$ , chegou-se a  $v_E = 2.62 \text{ cm/s}$ , o que é uma velocidade adequada para o manipulador robótico capturar o objeto da esteira.

## 2.3 Caixa de armazenamento

As caixas de armazenamento é o local onde o manipulador robótico coloca os objetos após capturá-los da esteira transportadora. Neste projeto, foi colocada uma caixa de armazenamento para cada tipo de objeto (grande ou pequeno). A caixa de armazenamento para objeto grande foi colocada à esquerda do manipulador robótico, enquanto a caixa de armazenamento para objeto pequeno foi colocada à direita do manipulador robótico, como mostra a Figura 25.

Figura 25 – Caixa de Armazenamento



Fonte: Elaborado pelo autor

## 3 ELETRÔNICA

Como foi visto no capítulo 2, tanto o manipulador robótico quanto a esteira transportadora funcionam por meio de motores elétricos. O manipulador robótico usa 4 servomotores elétricos, enquanto a esteira usa um motor de corrente contínua. No capítulo 4, foi abordado componentes como microcontrolador e sensores de distância, que também precisam de energia elétrica para funcionarem. Neste sentido, este capítulo aborda os circuitos elétricos dos componentes usados, para garantir um funcionamento estável, seguro e efetivo destes no cumprimento de suas funções.

### 3.1 Fonte de alimentação

O circuito do motor elétrico usado na esteira transportadora, tem uma tensão de entrada de 5 volts e o motor trabalha com uma corrente de *stall* de 0.5 amperes a 3 volts de tensão (Robocore, c2008).

Já a corrente de *stall* do micro servomotor SG90 é aproximadamente 600 miliampères (Luxosparts, 2017). Este valor pode variar ligeiramente dependendo das condições de operação e da tensão de alimentação, mas é uma estimativa aceitável para este micro servomotor SG90. Como o manipulador robótico funciona com 4 micro servomotores SG90, é necessária uma alimentação de 5 volts de tensão e no mínimo 2.4 amperes para o funcionamento adequado deste.

A corrente nominal dos sensores de distância ultrassônicos, é de 15 miliampères para cada um deles (Cytron Technologies Sdn. Bhd, 2013). Logo, os dois sensores ultrassônicos usam no máximo 30 miliampères de corrente.

Desta forma, somando as correntes destes componentes e acrescentando um percentual de 50% para uma margem de segurança devido a componentes secundários, foi estimado o uso de uma fonte de alimentação de 5 volts e 5 amperes para o funcionamento adequado dos circuitos elétricos.

### 3.2 Transistor

O transistor é um dispositivo semicondutor usado para amplificação, comutação e modulação de sinais elétricos. Ele é um componente essencial em circuitos eletrônicos e pode ser encontrado em praticamente todos os dispositivos eletrônicos modernos. Existem vários tipos de transistores, mas os mais comuns são o *transistor bipolar de junção* (BJT) e o *transistor de efeito de campo* (FET). No transistor bipolar de junção, quando uma pequena corrente é injetada na base, ela permite que uma corrente maior flua do emissor

para o coletor. A corrente de base controla a corrente entre o emissor e o coletor, permitindo a amplificação do sinal.

Para esse projeto, o transistor serve exclusivamente como chave para permitir ou não a passagem de corrente para o motor. Em nível lógico baixo (0 volts), o transistor não permite a passagem de corrente entre o emissor e o coletor, desabilitando a conexão entre o polo negativo da fonte com o polo negativo do motor. Em nível lógico alto (a partir de 0.7 volts), o transistor permite a passagem de corrente entre o emissor e o coletor, habilitando a conexão entre o polo negativo da fonte com o polo negativo do motor. Neste projeto foi usado o transistor TIP122 que é um modelo bipolar de junção NPN (onde 'NPN' indica a sequência de camadas de semicondutores tipo n, tipo p, e tipo n).

### 3.3 Motor

O motor de corrente contínua é um dispositivo eletromecânico que converte energia elétrica em energia mecânica através do uso de corrente contínua. Alguns de seus componentes são:

**Estator:** A parte fixa do motor que cria um campo magnético estático. Pode ser composto de ímãs permanentes ou enrolamentos de campo.

**Rotor (ou armadura):** A parte rotativa do motor onde a corrente elétrica é induzida para produzir movimento. Geralmente consiste em um núcleo de ferro com enrolamentos de fio condutor.

**Comutador:** Um interruptor rotativo que inverte a direção da corrente nos enrolamentos do rotor, garantindo que o torque gerado tenha uma direção constante.

**Escovas:** Feitas de material condutor (como grafite ou metal), as escovas mantêm contato com o comutador para transmitir corrente elétrica para os enrolamentos do rotor.

Quando a corrente contínua é aplicada ao motor, ela flui através das escovas e entra no comutador. A corrente que flui pelos enrolamentos do rotor cria um campo magnético ao redor dele. O campo magnético do rotor interage com o campo magnético do estator. Esta interação gera uma força (conhecida como força de Lorentz<sup>1</sup>) que faz o rotor girar. À medida que o rotor gira, o comutador troca a direção da corrente nos enrolamentos do rotor. Isso mantém a direção do campo magnético do rotor oposta ao do estator, garantindo que o torque (força de rotação) continue a girar o rotor na mesma direção. O processo de comutação ocorre continuamente enquanto a corrente é aplicada, permitindo que o rotor continue girando e produzindo movimento mecânico.

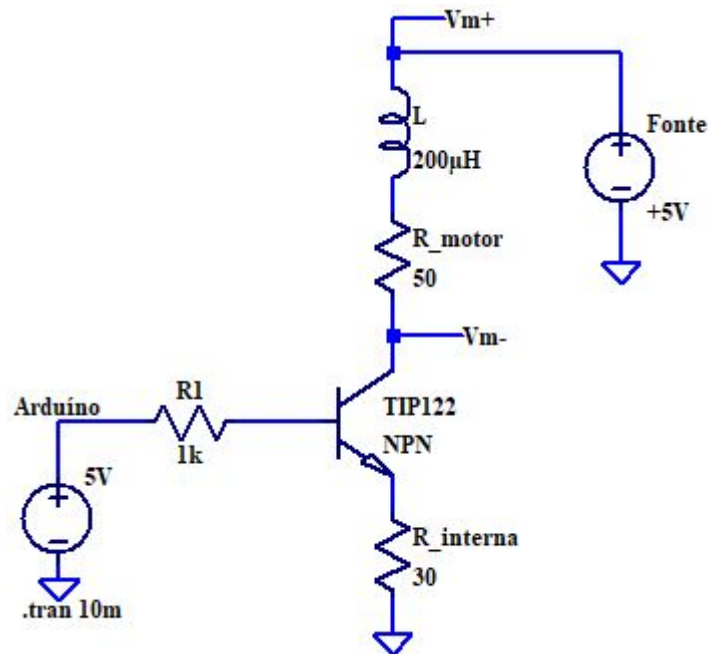
Quando a corrente é interrompida, o campo magnético colapsa rapidamente, gerando uma tensão de retorno ou "*flyback*", que pode ser bastante alta e danificar outros

---

<sup>1</sup> Físico holandês que trabalhou principalmente com eletromagnetismo, chegando a ganhar o prêmio Nobel em 1902 por seu trabalho sobre radiações eletromagnéticas.

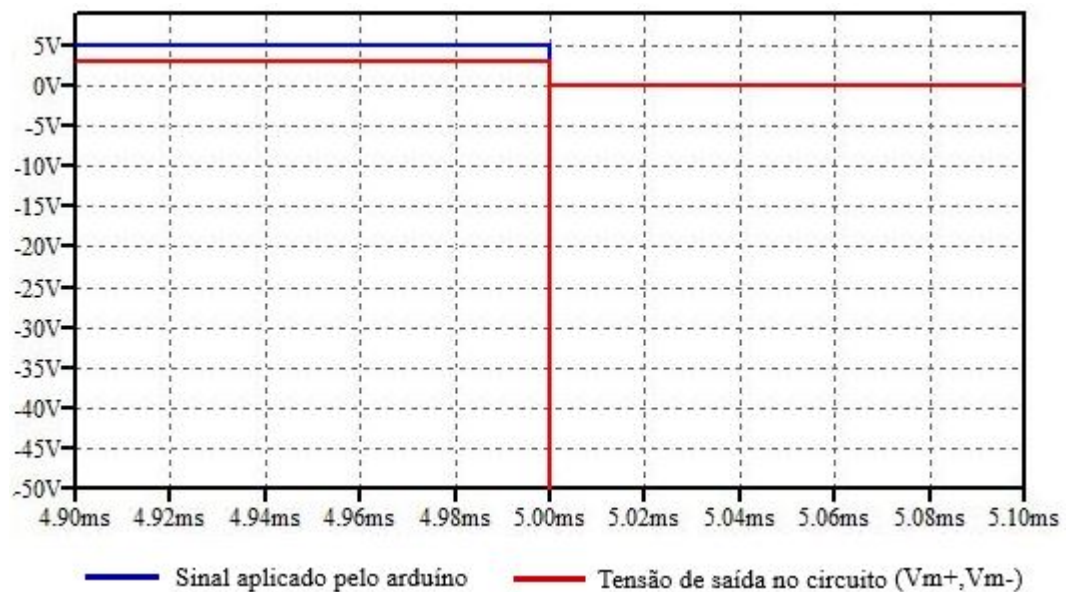
componentes do circuito, como transistores ou controladores, como mostrado na Figura 27. O esquema do circuito é mostrado na Figura 26.

Figura 26 – Circuito apenas com motor e transistor



Fonte: Elaborado pelo autor

Figura 27 – Simulação do circuito apenas com motor e transistor



Fonte: Elaborado pelo autor

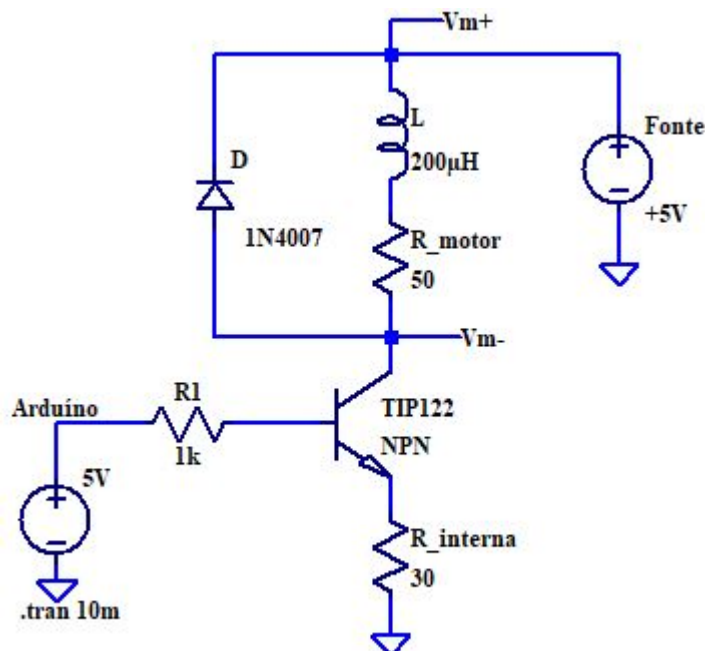
### 3.4 Diodo

Os diodos são componentes eletrônicos semicondutores que permitem a passagem de corrente elétrica em uma única direção, funcionando como uma espécie de válvula unidirecional. Eles são fundamentais em muitos circuitos eletrônicos devido às suas propriedades de retificação, modulação e proteção principalmente.

Um diodo de silício foi colocado em paralelo com um motor DC, como mostrado na Figura 28, para proteger o circuito de picos de tensão gerados pelo motor, especialmente quando ele é desligado repentinamente.

O diodo é conectado em paralelo com o motor em polaridade inversa em relação à tensão de alimentação do motor. Durante o funcionamento normal, a corrente flui através do motor e o diodo fica inversamente polarizado e não conduz. Quando há interrupção na corrente, o motor desliga e a corrente induzida pelo colapso do campo magnético gera uma tensão negativa (tensão de *flyback*). Essa tensão polariza o diodo diretamente, que fornece um caminho para a corrente induzida, permitindo que ela circule de volta através do motor e do diodo, dissipando a energia de forma segura. Foi possível simular esse comportamento através do software LTspice<sup>2</sup> para o circuito em questão, como mostrado na Figura 29.

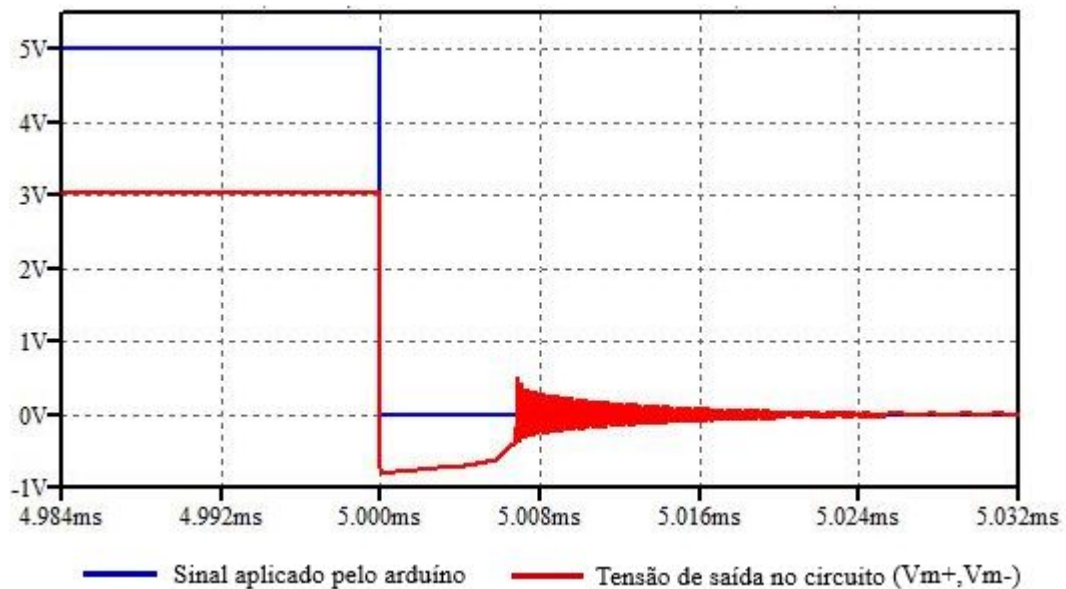
Figura 28 – Circuito com diodo adicionado



Fonte: Elaborado pelo autor

<sup>2</sup> O LTspice é um *software* de simulação que apresenta gráficos do comportamento de circuitos elétricos. Foi desenvolvido pela Linear Technology Corporation, pertencente desde 2017 à Analog Devices.

Figura 29 – Simulação do circuito com diodo adicionado



Fonte: Elaborado pelo autor

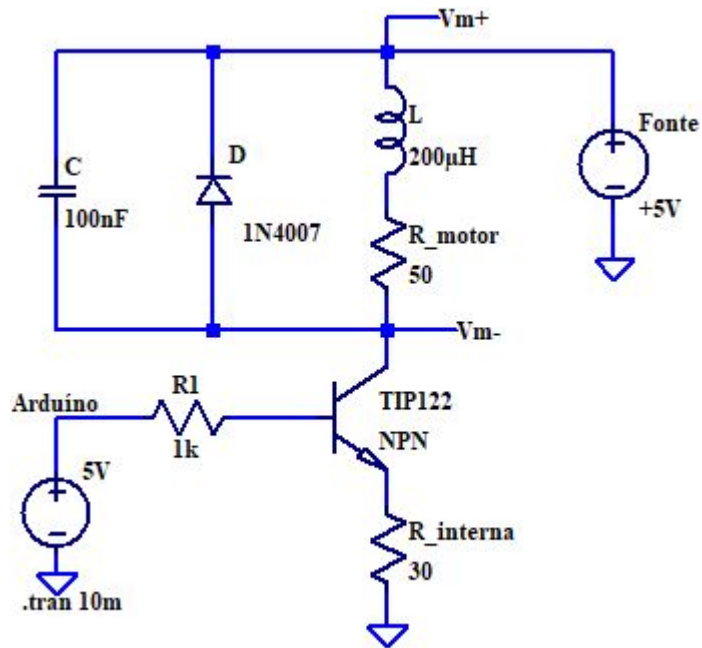
### 3.5 Capacitor

Capacitores são componentes eletrônicos passivos que armazenam energia elétrica no campo elétrico entre um par de placas condutoras separadas por um material isolante (dielétrico). São largamente usados em circuitos elétricos devido às características de filtragem de sinais e armazenamento de energia que estes apresentam.

Um capacitor de poliéster de 100nF e tensão máxima de 63 volts foi colocado em paralelo com o motor, Figura 30, para reduzir ruídos elétricos e interferências eletromagnéticas causados pelo motor. Motores DC, especialmente com escovas, geram ruídos elétricos devido à comutação rápida das escovas e ao contato intermitente com o comutador. Esses ruídos podem causar interferência em outros componentes eletrônicos. O capacitor ajuda a suavizar esses picos de ruído ao fornecer um caminho de baixa impedância para as altas frequências geradas durante a comutação. O capacitor age como um filtro passa-baixa, permitindo que as altas frequências (ruídos) passem através dele, enquanto as frequências mais baixas (a corrente contínua principal do motor) são bloqueadas. Quando os picos de ruído são gerados, o capacitor fornece um caminho alternativo de baixa impedância para esses picos, evitando que se propaguem para o restante do circuito.

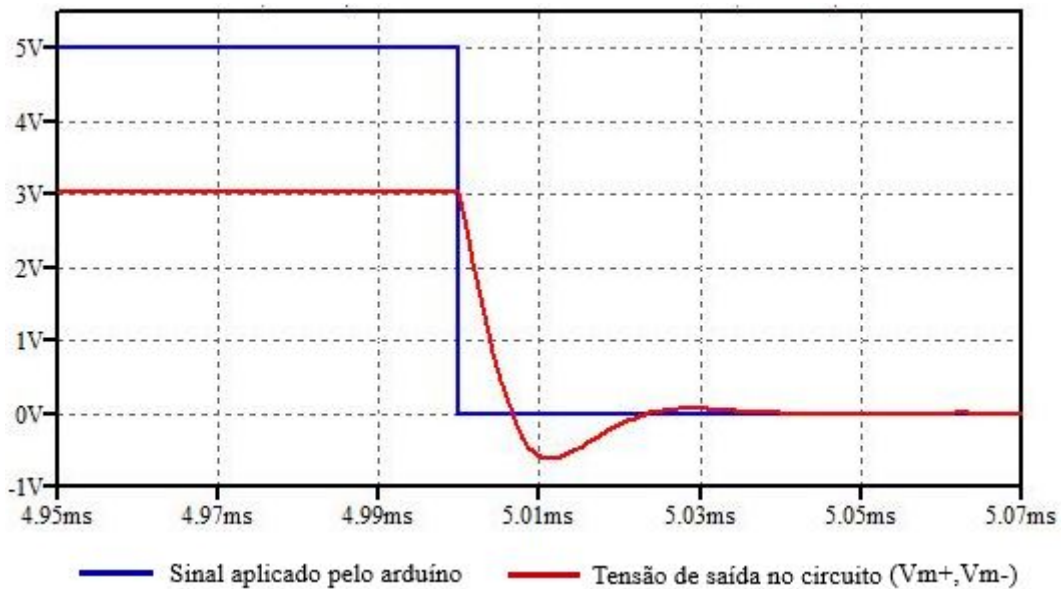
O capacitor também ajuda a estabilizar a tensão de alimentação, absorvendo os picos momentâneos de tensão e fornecendo corrente durante breves quedas de tensão. O capacitor pode armazenar energia temporariamente e liberá-la quando necessário, ajudando a manter a tensão de alimentação do motor estável. Foi possível simular esse comportamento através do software LTspice para o circuito, como mostrado na Figura 31.

Figura 30 – Circuito com diodo e capacitor adicionado



Fonte: Elaborado pelo autor

Figura 31 – Simulação do circuito com diodo e capacitor adicionado



Fonte: Elaborado pelo autor

Foi colocado um capacitor eletrolítico de  $1000\mu\text{F}$  em paralelo com os servomotores do manipulador robótico, seguindo o mesmo propósito do motor DC.

### 3.6 Resistor

O resistor limita a corrente absorvida pelo transistor para não ultrapassar a corrente máxima de funcionamento e queimar este dispositivo. O ganho de corrente de um transistor ( $h_{FE}$ ) pode variar com a temperatura e entre diferentes unidades do mesmo modelo de transistor. O resistor de base ajuda a controlar a corrente da base, garantindo que a corrente do coletor seja a desejada, independentemente dessas variações.



## 4 CONTROLE

O controle é uma disciplina essencial no desenvolvimento e operação de sistemas automatizados, incluindo manipuladores robóticos, esteiras transportadoras e sensores de distância. Em um contexto industrial, o controle adequado desses elementos é vital para garantir a eficiência, precisão e segurança das operações. Este capítulo aborda o controle usando esses componentes através da programação, destacando suas funções dentro desse projeto.

O controle preciso dos movimentos de um manipulador robótico é crucial para garantir que ele execute suas tarefas de maneira eficiente e segura. Isso envolve a utilização de algoritmos avançados que permitem o ajuste fino das posições, velocidades e acelerações das articulações do robô, garantindo movimentos suaves e precisos.

As esteiras transportadoras são componentes essenciais em linhas de produção e sistemas logísticos, responsáveis pelo transporte contínuo de materiais e produtos. O controle eficaz das esteiras transportadoras é crucial para garantir um fluxo constante e eficiente de materiais, minimizando paradas e interrupções no processo produtivo. Isso envolve a regulação da velocidade, aceleração e posicionamento da esteira, além de garantir a sincronização com outros equipamentos na linha de produção.

### 4.1 Sensor de distância

Sensores são responsáveis por identificar, converter e transmitir informações. O ser humano tem diversos sistemas sensoriais no seu corpo que os ajudam a compreender e interagir com o ambiente em que está. Esses inúmeros sistemas sensoriais do ser humano está diretamente ligado à sua evolução como espécie, assim como outras espécies que existem no mundo. Na automação, os sistemas de sensoriamento não são diferentes. Eles são fundamentais para coletar e converter dados do mundo físico ao mundo digital e auxiliar nas diferentes tomadas de decisões. Neste projeto, foi usado um sensor de distância que é capaz de entregar dados em tempo real e é muito utilizado em processos de automação. Ele é o responsável por verificar se há objetos na esteira e, caso haja, transmitir um sinal ao microcontrolador para iniciar uma rotina de programação específica para cada caso.

#### 4.1.1 Funcionamento

O módulo sensor de distância HC-SR04, é um dispositivo eletrônico capaz de medir a distância de um obstáculo usando ondas ultrassônicas que atingem o obstáculo e refletem de volta ao sensor de distância. Para que isso seja possível, o sensor de distância conta com duas partes que exercem funções diferentes na aplicação. A primeira, chamada de

transmissor, é responsável por enviar o pulso de onda por um determinado período. Já a segunda parte, chamada de receptor, é responsável por captar a onda refletida de volta ao módulo sensor de distância. Esse módulo consegue medir distâncias de 2 *cm* a 400 *cm* com erro de 3 *mm*.

Para começar uma medida, o pino *trigger* do módulo deve receber um pulso alto, ou seja, 5V do microcontrolador por pelo menos 10us. Isso inicia o sensor, o qual envia 8 ciclos de sinal ultrassônico a 40kHz pelo transmissor e espera pelo mesmo sinal refletido no receptor. Assim que o sinal sai do transmissor, o pino *echo* fica em nível lógico alto, ou seja, 5V e, quando esse sinal é detectado pelo receptor, o pino *echo* volta para nível lógico zero, 0V. Para obter a distância desse obstáculo, bastou medir o tempo que o pino *echo* fica com nível lógico alto, usando a fórmula de velocidade em função da distância e tempo, como mostra a equação 4.1.

$$v_{som} = \frac{2 \cdot \Delta S}{\Delta t_{high}} \Rightarrow \Delta S = \frac{v_{som} \cdot \Delta t_{high}}{2} \quad (4.1)$$

onde

$v_{som}$ : é a velocidade do som propagado no ar, que é constante a  $340 \frac{m}{s}$

$\Delta t_{high}$ : é o intervalo de tempo que o *echo* fica em nível lógico alto

$\Delta S$ : é a distância entre o módulo sensor de distância e o obstáculo

Observa-se que a distância percorrida pela onda no percurso de sair do *trigger*, reverberar no obstáculo e retornar ao *echo*, foi o dobro da distância entre o módulo e o obstáculo. Por esse motivo, foi necessário ajustar a equação 4.1 para se obter a distância entre o módulo e o obstáculo diretamente. A Figura 32 mostra uma foto do dispositivo HC-SR04.

Figura 32 – Módulo sensor de distância



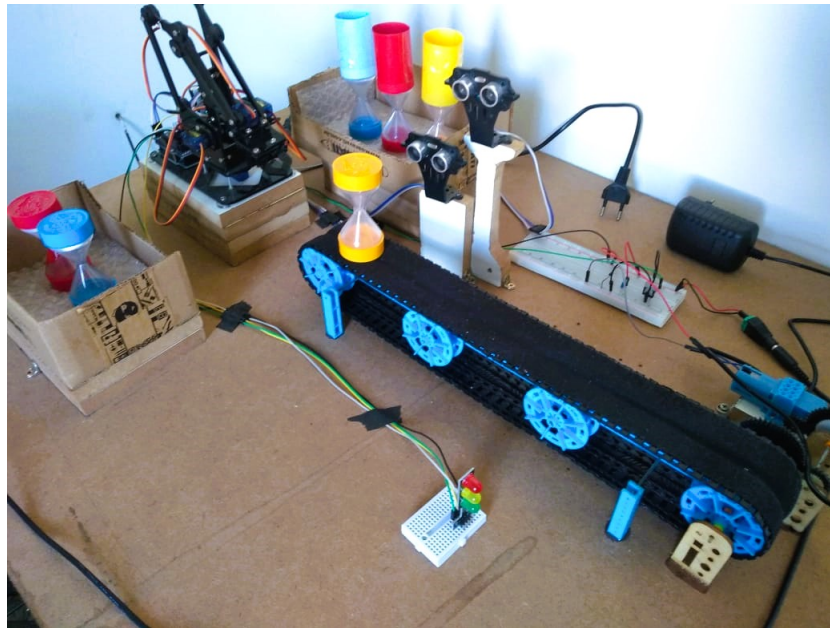
Fonte: Cytron Technologies (2013)

#### 4.1.2 Posicionamento dos sensores

Os sensores de distância foram posicionados próximo ao final de curso da esteira, como mostra a Figura 33. Para objetos pequenos, o sensor de distância foi posicionado a

6cm e para objetos grandes, o sensor de distância foi posicionado a 10cm do final de curso da esteira. Essa localização permite o melhor desempenho entre os diferentes componentes da planta industrial, segundo o código implementado no microcontrolador. Isso porque o programa entra em uma rotina de automação assim que é identifica o objeto e, se a rotina durar menos tempo, o programa tem mais tempo para executar outras tarefas previstas na otimização do processo.

Figura 33 – Posição do sensor de distância



Fonte: Elaborado pelo autor

Ainda, os sensores foram posicionados conforme a altura dos objetos que passam na esteira. A altura do sensor para objetos grandes não permite que este capte objetos pequenos. Este sensor também deve estar posicionado antes do sensor para objetos pequenos, para que a rotina programada para objetos grandes possa começar antes e, por consequência, desativar a rotina programada para objetos pequenos, em um dado intervalo de tempo. Isto é necessário, visto que o sensor para objetos pequenos, sempre capta o objeto grande, a menos que esteja desativado.

Ambos os sensores estão posicionados a uma distancia entre 5 e 10 cm da esteira, que é uma distância adequada para as dimensões dos objetos a serem separados.

## 4.2 Microcontrolador

O microcontrolador escolhido neste projeto foi o Arduino UNO, Figura 34, que é um microcontrolador popular de baixo custo, podendo ser utilizado em diversas aplicações, desde *internet das coisas* (IoT) até aplicações de automatização industrial. Este microcontrolador conta com um ambiente de desenvolvimento integrado (IDE) próprio

com editor de códigos na linguagem c/c++ e transfere dados compilados entre a IDE e o microcontrolador via conexão USB. O Quadro 1 mostra as especificações do Arduino UNO.

Quadro 1 – Especificações técnicas do Arduino UNO

Microcontrolador	ATmega328P
Tensão de operação	5V
Tensão de entrada (recomendada)	7-12V
Tensão de entrada (limite)	6-20V
Pinos digitais I/O	14 (6 podem ser usados como saídas PWM).
Pinos de entrada analógica	6
Corrente DC por pino I/O	20 mA
Corrente DC para o pino 3.3V	50 mA
Memória flash	32 kB (ATmega328P) dos quais 0.5 kB são usados pelo bootloader.
SRAM	2 kB (ATmega328P)
EEPROM	1 kB (ATmega328P)
Frequência de clock	16 MHz
Interface de programação	USB

Fonte: Arduino S.R.L. 2024

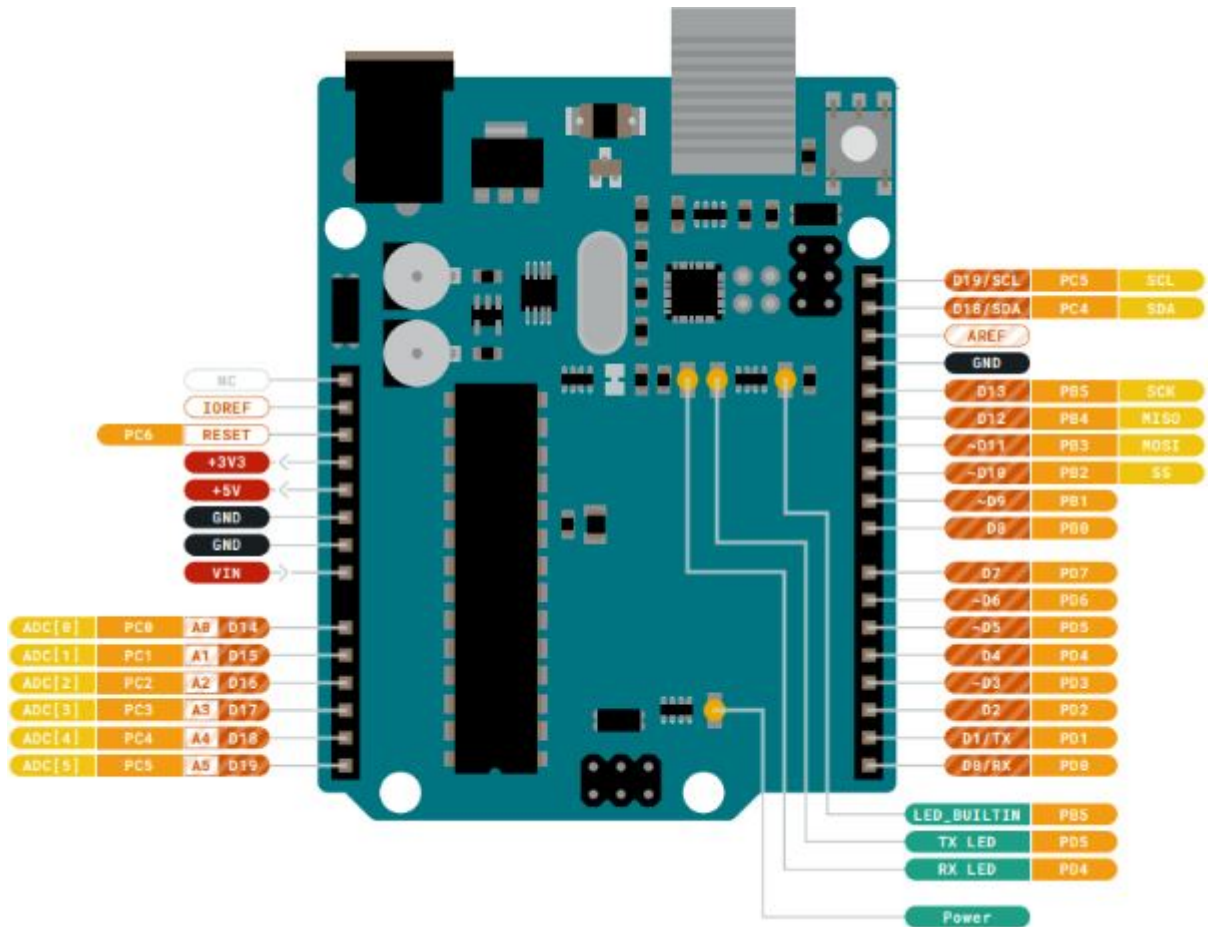
Figura 34 – Arduino UNO



Fonte: Arduino S.R.L. (2024, p.1)

Sua popularidade é sustentada pela combinação de facilidade de uso, ampla documentação e suporte comunitário. A Figura 35 mostra o diagrama de conexões dos pinos do Arduino UNO. Nele foi possível verificar que as entradas analógicas também podem ser usadas como saídas digitais, dependendo do código implementado. Isto foi útil no projeto, uma vez que existem muitos componentes utilizados e poucas portas de saída no Arduino UNO.

Figura 35 – Pinos de conexão do Arduino UNO



Fonte: Arduino S.R.L. (2024, p.9)

#### 4.2.1 Shield

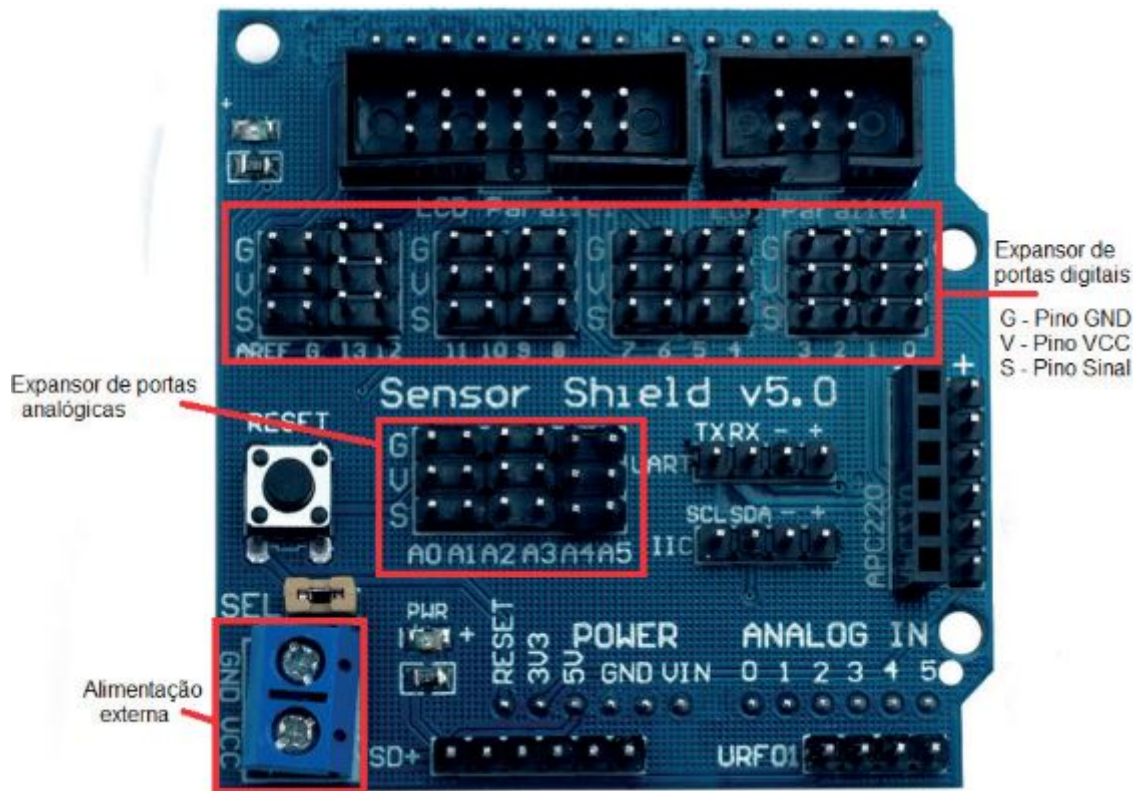
A Shield V5.0 é uma placa adicional frequentemente utilizada em conjunto com plataformas de prototipagem eletrônica, por exemplo.

De maneira geral, a Shield V5.0 apresenta compatibilidade com diversas versões de placas Arduino, como Uno, Mega e Leonardo. Sua principal vantagem é a facilidade de uso, sendo projetada para se conectar diretamente às portas da placa base sem a necessidade de soldagem. Além disso, ela expande as funcionalidades da placa base, permitindo a integração de módulos adicionais como controle de motores, comunicação sem fio e sensores.

Pode ser utilizada para o controle de motores DC, motores de passo e servomotores. Tem uma arquitetura que facilita a conexão de servomotores e potenciômetros para controle de manipuladores robóticos, como o Arduino, como mostra a Figura 36.

Proporciona uma série de conectores para sensores e módulos, facilitando a conexão de múltiplos sensores como os de temperatura, umidade, infravermelho e ultrassom — tendo inclusive uma área exclusiva para conexão desse sensor. É compatível com uma ampla gama de sensores, tornando-se uma ferramenta versátil para diversos projetos.

Figura 36 – Placa Shield v5.0



Fonte: Secretaria da Educação do Estado do Paraná (2021, p.6)

#### 4.2.2 Controle manual

Para que a automatização do processo fosse bem sucedida, foi necessário primeiro encontrar os ângulos que os servomotores do manipulador robótico teriam que implementar para levar os objetos da esteira transportadora até as caixas de recolhimento. Para isto, foram necessários a utilização 4 potenciômetros de 10 k $\Omega$  conectados às portas do Arduino, como mostrado no Quadro 2.

Quadro 2 – Conexão dos componentes com o Arduino para controle manual

Pino de sinal do componente	Porta
Potenciômetro que controla a base	A0
Potenciômetro que controla o braço	A1
Potenciômetro que controla o antebraço	A2
Potenciômetro que controla a garra	A3
Servomotor que controla a base	D2
Servomotor que controla o braço	D3
Servomotor que controla o antebraço	D4
Servomotor que controla a garra	D5

Fonte: Elaborado pelo autor

A partir dessa configuração de *hardware*, foi implementado o código de *software* para capturar os ângulos de cada servomotor no trajeto.

## Código para controle manual

```
#include <Servo.h>
#define potBase A0
#define potBraco A1
#define potAnteBraco A2
#define potGarra A3

Servo servoBase, servoBraco, servoAnteBraco, servoGarra;
int anguloBase, anguloBraco, anguloAnteBraco, anguloGarra;

void setup() {
  servoBase.attach(2);
  servoBraco.attach(3);
  servoAnteBraco.attach(4);
  servoGarra.attach(5);
  Serial.begin(9600);
}

void loop() {
  anguloBase = map(analogRead(potBase), 0, 1023, 0, 180);
  anguloBraco = map(analogRead(potBraco), 0, 1023, 15, 140);
  anguloAnteBraco = map(analogRead(potAnteBraco), 0, 1023, 0,
    160);
  anguloGarra = map(analogRead(potGarra), 0, 1023, 9, 60);
  servoBase.write(anguloBase);
  servoBraco.write(anguloBraco);
  servoAnteBraco.write(anguloAnteBraco);
  servoGarra.write(anguloGarra);
  ImprimePosicao();
}

void ImprimePosicao(){
  Serial.print("posicao da Base: ");
  Serial.println(servoBase.read());
  Serial.print("posicao do Braco: ");
  Serial.println(servoBraco.read());
  Serial.print("posicao do AnteBraco: ");
  Serial.println(servoAnteBraco.read());
  Serial.print("posicao da Garra: ");
  Serial.println(servoGarra.read());
  Serial.println("-----");
  delay(100);
}
```

Através do controle manual do manipulador robótico, foi possível registrar os ângulos dos servomotores no trajeto percorrido da esteira transportadora até a caixa de recolhimento, como mostra a Tabela 1.

Tabela 1 – Posições do manipulador robótico registradas com controle manual

Posição	Base	Braço	Antebraço	Garra
Inicial	92°	45°	110°	9°
Leva a garra até o objeto	92°	92°	125°	32°
Captura o objeto	92°	92°	125°	9°
Levanta o objeto	92°	20°	160°	9°
Gira a base	0°	20°	160°	9°
Coloca o objeto na caixa	0°	100°	128°	36°
Manobra a garra	0°	100°	110°	36°
Retrai o braço	0°	45°	110°	36°
Volta ao início	92°	45°	110°	9°

Fonte: Elaborado pelo autor

Como pôde ser visto na Tabela 1, o percurso da garra entre a retirada do objeto da esteira, sua alocação na caixa de armazenamento e retorno à posição inicial, envolve algumas manobras do braço. Por esse motivo se torna mais simples gravar esse percurso a partir de um controle manual do manipulador robótico. Também seria possível encontrar essas posições através dos gráficos e informações adquiridas no capítulo 2, onde foi possível encontrar a posição da garra para qualquer combinação de graus dos servomotores. Entretanto, por facilidade, o controle manual do manipulador robótico torna-se um recurso mais efetivo.

#### 4.2.3 Controle automatizado – Automatização

O controle automatizado do manipulador robótico gera um desempenho de trabalho muito maior que o controle manual. Os objetos são separados com maior agilidade, há uma precisão maior e não exige nenhuma dependência ou intervenção humana. Em outras palavras, a eficiência aumenta consideravelmente.

Para que a automatização do sistema seja possível, foi fundamental a integração dos sensores de distância ao *hardware*. A conexão dos componentes com o Arduino pode ser vista no Quadro 3.

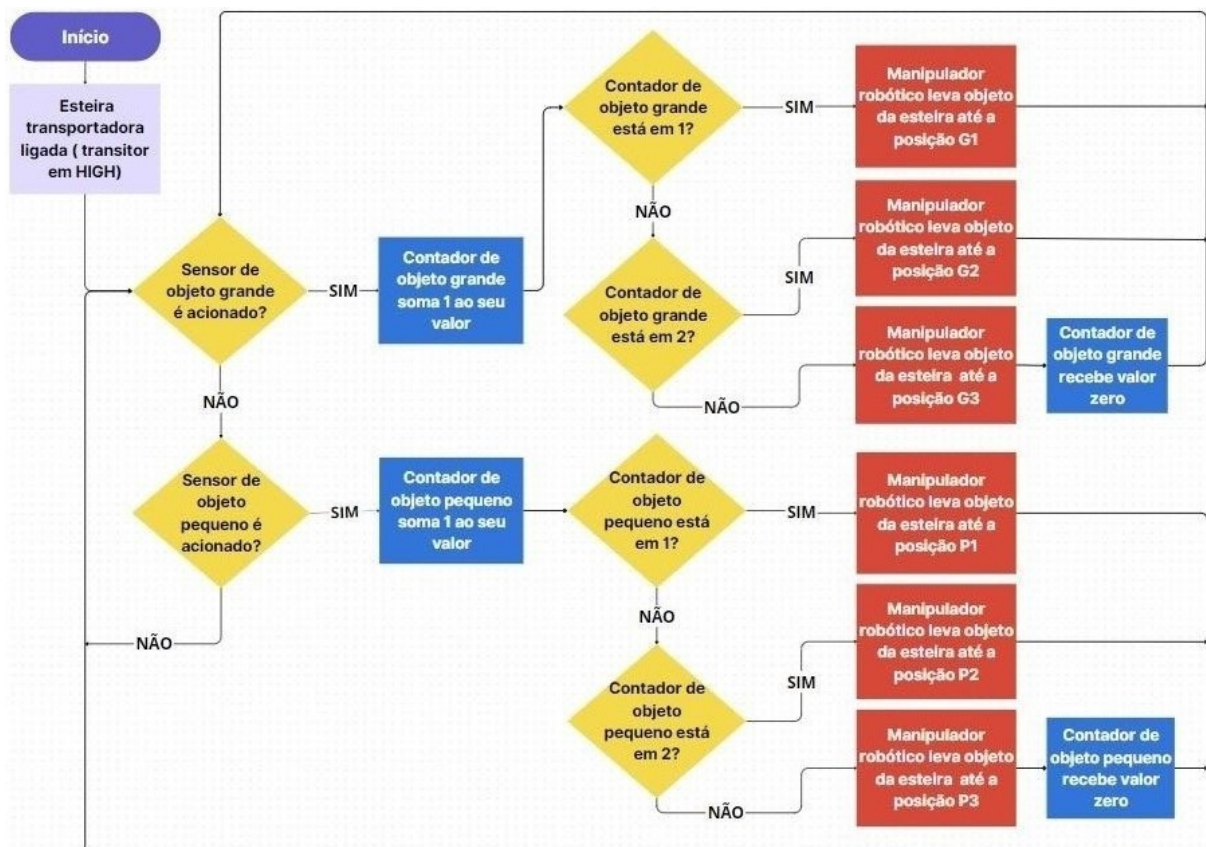
Com os componentes conectados ao Arduino, implementou-se o código de programação com base no fluxograma mostrado na Figura 37.

Quadro 3 – Conexão dos componentes com o Arduino para controle automatizado

Pino de sinal do componente	Porta
Luz verde	D2
Luz vermelha	D3
Servomotor que controla a base	D4
Servomotor que controla o braço	D5
Servomotor que controla o antebraço	D6
Servomotor que controla a garra	D7
Trigger (sensor de objetos pequenos)	D8
Echo (sensor de objetos pequenos)	D9
Transistor	D10
Luz amarela	D11
Trigger (sensor de objetos grandes)	D14 (A0)
Echo (sensor de objetos grandes)	D15 (A1)

Fonte: Elaborado pelo autor

Figura 37 – Fluxograma da automatização



Fonte: Elaborado pelo autor

### Código de automatização

```

#include <Servo.h>
#include <HCSR04.h>
  
```

```
#define transistor 10
#define luz_verde 2
#define luz_vermelha 3
#define luz_amarela 11

Servo servoBase, servoBraco, servoAnteBraco, servoGarra;
UltraSonicDistanceSensor sensor_1 (A0, A1);
// sensor_1: sensor que capta objetos grandes (trigger, eccho)
UltraSonicDistanceSensor sensor_2 (8, 9);
// sensor_2: sensor que capta objetos pequenos (trigger, eccho)

int tmp = 20;
// constante de tempo utilizada para deixar o movimento dos
// bracos roboticos mais suave. Tambem define a velocidade dos
// servomotores como 0.8726 rad/s
float distancia_1;
// variavel que recebe a distancia do sensor 1 (sensor grande)
float distancia_2;
// variavel que recebe a distancia do sensor 2 (sensor pequeno)
float dist_minima = 12.00;
// constante da distancia minima para captar o objeto
int chegada_1 = 2600;
// constante de tempo que conta quando o objeto sai do sensor_1 e
// chega no final da esteira
int chegada_2 = 1200;
// constante de tempo que conta quando o objeto sai do sensor_2 e
// chega no final da esteira
int contador_obj_grande = 0;
// contador que vai de 0 a 3
int contador_obj_pequeno = 0;
// contador que vai de 0 a 3

void setup() {
    // Configurando pinos analogicos A0 e A1 para funcionarem como
    // pinos digitais para o sensor de distancia de objetos grandes
    pinMode(A0, OUTPUT);
    pinMode(A1, INPUT);
    pinMode(transistor, OUTPUT);
    digitalWrite(transistor, HIGH);

    // Configurando pinos dos servomotores no arduino
```

```
servoBase.attach(4);
servoBraco.attach(5);
servoAnteBraco.attach(6);
servoGarra.attach(7);

//Posicao inicial:
servoBase.write(92);
servoBraco.write(45);
servoAnteBraco.write(110);
servoGarra.write(9);

// Configurando semaforo para iniciar com luz verde
pinMode(luz_verde, OUTPUT);
pinMode(luz_vermelha, OUTPUT);
pinMode(luz_amarela, OUTPUT);
digitalWrite(luz_verde, HIGH);
digitalWrite(luz_vermelha, LOW);
digitalWrite(luz_amarela, LOW);
}

void loop() {
  // Variavel distancia recebendo as medidas do sensor de
  distancia
  distancia_1 = sensor_1.measureDistanceCm();
  distancia_2 = sensor_2.measureDistanceCm();

  // Verificando se ha objeto grande na esteira passando pelos
  sensores
  if((distancia_1 < dist_minima) && (distancia_1 > 0)){
    contador_obj_grande++;
    // contador de objetos grandes soma 1 ao seu valor atual
    switch(contador_obj_grande){
      case 1:
        obj_grande_1();
        break;
      case 2:
        obj_grande_2();
        break;
      default:
        obj_grande_3();
        contador_obj_grande = 0;
        break;
    }
  }
```

```
}  
// && (distancia_1 > 0) necessario pois se nao houver  
  obstaculos no limite de distancia que o sensor capta, 4  
  metros, o sensor por padrao da biblioteca utilizada, retorna  
  -1  
else if((distancia_2 < dist_minima) && (distancia_2 > 0)){  
  contador_obj_pequeno++;  
  switch(contador_obj_pequeno){  
    case 1:  
      obj_pequeno_1();  
      break;  
    case 2:  
      obj_pequeno_2();  
      break;  
    default:  
      obj_pequeno_3();  
      contador_obj_pequeno = 0;  
      break;  
  }  
}  
// Se nao, so continua rodando a esteira  
else{  
  digitalWrite(transistor, HIGH);  
}  
}
```

Como pôde ser visto acima, existem 6 funções no código desse programa (3 para o objeto grande e 3 para o objeto pequeno). Cada uma destas funções representam um posicionamento diferente na caixa de armazenamento para os objetos serem alocados. Como a caixa de armazenamento recebe no máximo 3 objetos de cada tipo (grandes ou pequenos), foi necessário criar 3 posições diferentes para a alocação desses objetos na caixa. Desta forma, as funções têm códigos similares com diferenciação apenas nos ângulos que os servomotores imprimiram para obter posições diferentes no espaço. Para evitar repetições e deixar este relatório mais conciso, optou-se por mostrar apenas uma destas funções, que foi implementada logo após o código da parte principal.

### Código da função de posição do objeto

```
void obj_pequeno_1(){  
  // Ascende luz vermelha:  
  digitalWrite(luz_verde, LOW);  
  digitalWrite(luz_vermelha, HIGH);
```

---

```
//Espera a chegada do objeto ate o alcance do manipulador
    robotico
delay(chegada_2);

// Abre a garra:
    servoGarra.write(32);

// Levanta o antebraço suficientemente:
for (int i = 110; i < 125; i++){
    servoAnteBraco.write(i);
    delay(tmp);
}
// Leva o braço ate o objeto:
for (int i = 45; i < 92; i++){
    servoBraco.write(i);
    delay(tmp);
}
// Fecha a Garra e captura a objeto:
for (int i = 32; i > 9; i--){
    servoGarra.write(i);
    delay(30);
}
// Levanta objeto da esteira:
for (int i = 125; i < 160; i++){
    servoAnteBraco.write(i);
    delay(tmp);
}
// Puxa o braço para perto (diminuindo o momento de inercia
    angular):
for (int i = 92; i > 20; i--){
    servoBraco.write(i);
    delay(tmp);
}
// Ascende luz verde:
digitalWrite(luz_verde, HIGH);
digitalWrite(luz_vermelha, LOW);

// Gira a base na posição da caixa:
for (int i = 92; i > 0; i--){
    servoBase.write(i);
    delay(tmp);
}
```

```
}
// Posiciona braco para colocacao do objeto na caixa:
for (int i = 20; i < 100; i++){
    servoBraco.write(i);
    delay(tmp);
}
// Posiciona antebraço para colocacao do objeto na caixa:
for (int i = 160; i > 128; i--){
    servoAnteBraco.write(i);
    delay(tmp);
}
// Abre a garra deixando o objeto na caixa:
for (int i = 9; i < 36; i++){
    servoGarra.write(i);
    delay(tmp);
}
// Abaixa o antebraço para não colidir com o objeto na volta
para a posicao inicial:
for (int i = 128; i > 110; i--){
    servoAnteBraco.write(i);
    delay(tmp);
}
// Recolhe braco para a posicao inicial:
for (int i = 100; i > 45; i--){
    servoBraco.write(i);
    delay(tmp);
}
// Fecha a garra para a posicao inicial:
for (int i = 36; i > 9; i--){
    servoGarra.write(i);
    delay(10);
}
// Gira a base para posicao inicial:
for (int i = 0; i < 92; i++){
    servoBase.write(i);
    delay(tmp);
}
}
```

Como foi possível observar, a variável *tmp*, definida no código principal e usada nos códigos das funções, define um *delay* de 20 milissegundos (0.02 segundos) para cada grau percorrido. Dessa forma, foi possível encontrar a velocidade aplicado pelos servomotores,

---

$\dot{\alpha}$  e  $\dot{\theta}$ , com uma regra de três simples, como mostra a equação a seguir:

$$\pi \text{ rad} = 180^\circ \quad (4.2)$$

$$x = 1^\circ \quad (4.3)$$

Logo

$$x \cdot 180^\circ = 1^\circ \cdot \pi \text{ rad} \Rightarrow x = \frac{1^\circ \cdot \pi \text{ rad}}{180^\circ} \Rightarrow x = 0.0174 \text{ rad} \quad (4.4)$$

A Velocidade do servomotor SG90 é de  $8.73 \text{ rad/s}$ . Logo, para percorrer  $0.0174 \text{ rad}$  ele demora  $0,002$  segundos. Para encontrar a velocidade média que o servomotor atua, usou-se a seguinte equação:

$$\frac{0.0174 \text{ rad}}{(0.02 + 0,002) \text{ s}} = 0.79 \text{ rad/s} \quad (4.5)$$

Portanto a velocidade média é de  $0.79 \text{ rad/s}$ .



## 5 CONCLUSÃO

Os capítulos anteriores mostraram resultados satisfatórios para os objetivos deste projeto. Foi possível realizar a separação de dois objetos que se diferenciavam por suas alturas, de forma totalmente automatizada, sem nenhuma dependência ou interferência humana. Ou seja, neste projeto foi realizado a automatização com sucesso de um processo que exigia tomadas de decisões a partir de um critério definido previamente

Devido ao baixo custo do projeto, existiram algumas imprecisões na *performance* deste, por exemplo, a velocidade atingida pela esteira que depende diretamente da tensão aplicada ao motor DC. Uma vez que as conexões elétricas são realizadas por fios encaixados em *protoboards*, a conexão nem sempre é perfeita implicando em alguns mal contatos que fazem a tensão de alimentação do motor DC oscilar quando há movimentações significativas na plataforma de encaixe dos componentes. Isto resulta em oscilações na velocidade linear exercida pela esteira que pode comprometer a captura do objeto pela garra no tempo estimado e, conseqüentemente, não conseguir capturar o objeto. Uma alternativa de resolução deste problema seria uma conexão destes componentes elétricos através da solda ao invés da *protoboard*.

Projetar um modelo de planta industrial e montá-lo fisicamente permite denotar diferentes dificuldades, porém com mesmo grau de complexidade. Neste projeto, houve a necessidade de compreender o modelo físico para compreender sua parte teórica, assim com também o inverso. Um modelo miniaturizado de uma planta industrial pôde descrever de forma satisfatória como seria o comportamento de uma planta industrial em escala real. Desta forma, pôde-se concluir que um modelo pode fornecer informações e dados importantes a serem observados em um projeto. Obviamente, respeitando a diferença de escalas, um modelo representa o comportamento real do objeto de estudo.

Neste projeto foram utilizados diversos conceitos de muitas áreas distintas mas que se reúnem no estudo da automatização. Pode-se citar o estudo da cinemática no manipulador robótico e na esteira na área da mecânica, assim como a programação no microcontrolador na área da computação e o estudo de circuitos eletrônicos para fornecer energia de forma segura a estes componentes, na área da elétrica. Ou seja, o estudo de automatização de sistemas depende diretamente do estudo outras áreas da ciência para ser realizado.

A manufatura do projeto pode ser vista através do vídeo na plataforma do YouTube acessando <https://youtu.be/bomavxY2Deo>.

Este projeto demonstra o marco histórico atual que a espécie humana vive, onde foi possível compreender algumas das leis e fenômenos que regem a natureza e utilizá-los

em favorecimento dessa espécie. A realização de trabalho, sem a necessidade de esforço humano, denota um avanço evolutivo que reflete diretamente na qualidade de vida da sociedade.

## REFERÊNCIAS

ARDUINO S.R.L. **Arduíno UNO**: datasheet. Monza, Itália, 2024. 1,9 p. Disponível em: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>. Acesso em: 08 jun 2024.

BR ELETRÔNICA. **Braço robótico**. c2016. Disponível em: <https://www.breletronica.com.br/buscar?q=bra%C3%A7o+rob%C3%B3tico>. Acesso em: 07 jun 2024.

BURNS, E. M.; LERNER, R. E.; MEACHAM, S. **História da civilização ocidental: do homem das cavernas às naves espaciais**. [S.l.: s.n.]: Globo, 2000.

CYTRON TECHNOLOGIES SDN. BHD. **Manual de uso do HC-SR04**. 2013. 1,6 p. Disponível em: <https://cdn.awsli.com.br/945/945993/arquivos/HCSR04.pdf>. Acesso em: 08 jun 2024.

GRAY, B. **MeArm V0.4 - Braço robótico de bolso**. 2014. Licença: Creative Commons, 11 jun. 2014. Disponível em: <https://www.instructables.com/Pocket-Sized-Robot-Arm-meArm-V04/>. Acesso em: 1 ago. 2023.

LUXOSPARTS. **Micro servo motor**: datasheet. Malmo, Suécia, 2017. 1 p. Disponível em: [https://www.kjell.com/globalassets/mediaassets/701916\\_87897\\_datasheet\\_en.pdf?ref=4287817A7A](https://www.kjell.com/globalassets/mediaassets/701916_87897_datasheet_en.pdf?ref=4287817A7A). Acesso em: 08 jun 2024.

NORTON, R. L.; HAN. **Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines**. [S.l.: s.n.]: McGraw-Hill New York, 2008. v. 4.

ROBOCORE. **Motor DC**. c2008. Disponível em: <https://www.robocore.net/motor-motoredutor/motor-dc-3-6v-com-caixa-de-reducao-de-aco>. Acesso em: 08 jun 2024.

SARAVATI. **Braço Robótico**: Manual de montagem. 2023. Disponível em: <https://drive.google.com/file/d/18bnIST5AHhpAxmiQllfTLz5viHrOjE2h/view>. Acesso em: 06 jun 2024.

SARAVATI. **Braço robótico**. 2023. Disponível em: <https://www.saravati.com.br/braco-robotico-ps-preto-4-micro-servos-para-montar.html>. Acesso em: 06 jun 2024.

SECRETARIA DA EDUCAÇÃO DO ESTADO DO PARANÁ. **Robótica**: Controlando servos motores. Curitiba, 2021. 6 p. Disponível em: [https://aluno.escoladigital.pr.gov.br/sites/alunos/arquivos\\_restritos/files/documento/2021-09/aula18\\_controlando\\_servos\\_motores\\_m2.pdf](https://aluno.escoladigital.pr.gov.br/sites/alunos/arquivos_restritos/files/documento/2021-09/aula18_controlando_servos_motores_m2.pdf). Acesso em: 08 jun 2024.



## APÊNDICES



## APÊNDICE A – CÓDIGO NO MATLAB DA POSIÇÃO DA GARRA

A posição da garra,  $\beta$  e  $\gamma$  em função de  $5^\circ < \alpha < 90^\circ$  e  $0^\circ < \theta < 80^\circ$  está descrita no código de MATLAB abaixo:

```

% ----- Mecanismo principal -----
clc;
clear;

L0 = 3.5;
L1 = 8.1;
L2 = 3.5;
L3 = 8.1;

alfa = 5*pi/180:pi/36:90*pi/180;
theta = 0*pi/180:pi/36:80*pi/180;

erro = 10^(-6);
Nmax = 10;
X = zeros(2,1);
beta = zeros(length(theta),length(alfa));
gama = zeros(length(theta),length(alfa));

for i = 1:length(theta)
    X0 = [pi/4;
         pi/4];

    for j = 1:length(alfa)
        for k = 1:Nmax
            beta(i,j) = X0(1,1);
            gama(i,j) = X0(2,1);

            f = [-L0*cos(theta(i)) + L3*cos(beta(i,j)) + L2*cos(
                gama(i,j)) - L1*cos(alfa(j));
                L0*sin(theta(i)) + L3*sin(beta(i,j)) - L2*sin(
                gama(i,j)) - L1*sin(alfa(j))];

            J = [-L3*sin(beta(i,j)), -L2*sin(gama(i,j));
                L3*cos(beta(i,j)), -L2*cos(gama(i,j))];

            X = X0 - J\f;
        end
    end
end

```

```
        if((abs(X(1,1) - X0(1,1)) < erro) && (abs(X(2,1) - X0
            (2,1)) < erro))
            break;
        end

        X0 = X;
    end
end
end
betaGraus = round(beta*180/pi)
gamaGraus = round(gama*180/pi)

figure(1)
subplot(2,2,1)
plot(alfa*180/pi,betaGraus(1,:), 'r-')
title('\theta = 0 graus')
xlabel('\alpha')
ylabel('\beta')
axis([0 90 0 90])
grid on

subplot(2,2,2)
plot(alfa*180/pi,betaGraus(7,:), 'r-')
title('\theta = 30 graus')
xlabel('\alpha')
ylabel('\beta')
axis([0 90 0 90])
grid on

subplot(2,2,3)
plot(alfa*180/pi,betaGraus(13,:), 'r-')
title('\theta = 60 graus')
xlabel('\alpha')
ylabel('\beta')
axis([0 90 0 90])
grid on

subplot(2,2,4)
plot(alfa*180/pi,betaGraus(17,:), 'r-')
title('\theta = 80 graus')
xlabel('\alpha')
```

```
ylabel('\beta')
axis([0 90 0 90])
grid on

figure(2)
subplot(2,2,1)
plot(theta*180/pi,gamaGraus(:,1),'b-')
title('\alpha = 5 graus')
xlabel('\theta')
ylabel('\gamma')
grid on

subplot(2,2,2)
plot(theta*180/pi,gamaGraus(:,6),'b-')
title('\alpha = 30 graus')
xlabel('\theta')
ylabel('\gamma')
grid on

subplot(2,2,3)
plot(theta*180/pi,gamaGraus(:,12),'b-')
title('\alpha = 60 graus')
xlabel('\theta')
ylabel('\gamma')
grid on

subplot(2,2,4)
plot(theta*180/pi,gamaGraus(:,18),'b-')
title('\alpha = 90 graus')
xlabel('\theta')
ylabel('\gamma')
grid on
% ----- Mecanismo secundario e Animacao -----
L4 = 8.1;
L5 = 1.2;
L6 = 5.3;

for i = 1:length(theta)
    for j = 1:length(alfa)

        xB = -L0*cos(theta(i));
        yB = L0*sin(theta(i));
```

```

xC = L3*cos(beta(i,j));
yC = L3*sin(beta(i,j));
xD = L1*cos(alfa(j));
yD = L1*sin(alfa(j));
xE = L4*cos(gama(i,j));
yE = -L4*sin(gama(i,j));
xF = 0;
yF = -1.2;

xG = L1*cos(beta) + L4*cos(gama) + L6*ones(17,18); % beta
      esta no lugar de alfa ja que sao iguais
yG = L1*sin(beta) - L4*sin(gama) - L5*ones(17,18); % beta
      = alfa com dimenssao matricial 9x9

figure(3)
plot(-1.35,-1.5,'ks','markersize',70,'markerfacecolor',
      ,[0.85 0.85 0.85])
hold on
plot([0 xB],[0 yB],'r-','linewidth',3)
plot([xB xB+xC],[yB yB+yC],'b-','linewidth',3)
plot([xB+xC xD],[yB+yC yD],'b-','linewidth',3)
plot([0 xD],[0 yD],'b-','linewidth',3)
plot([xD xD+xE],[yD yD+yE],'b-','linewidth',3)
plot([xD+xE xD+xE+xF],[yD+yE yD+yE+yF],'b-','linewidth',
      ,2)
plot([xD+xE+xF xG(i,j)],[yD+yE+yF yG(i,j)],'r-','
      linewidth',2)
plot(0,-0.3,'k^','markersize',7,'markerfacecolor',[0.6
      0.6 0.6])
plot(0,0,'ko','markersize',4,'markerfacecolor','w')
plot(xB,yB,'ko','markersize',4,'markerfacecolor','w')
plot(xB+xC,yB+yC,'ko','markersize',4,'markerfacecolor','w
      ')
plot(xD,yD,'ko','markersize',4,'markerfacecolor','w')
plot(xD+xE,yD+yE,'ko','markersize',4,'markerfacecolor','w
      ')
plot(xD+xE+xF,yD+yE+yF,'ko','markersize',3,'
      markerfacecolor','w')
plot(xG(i,j),yG(i,j),'ro','markersize',6,'markerfacecolor
     ',[0.8 0.3 0.3])
plot(xG(1:i,1:j),yG(1:i,1:j),'k.')
```

```
hold off
```

```

axis([-10 25 -10 25])
axis square
xlabel('coordenada x_A (cm)')
ylabel('coordenada y_A (cm)')
title('Posicoes da garra do Braco Robotico')
grid on
pause(0.1)
end
end

```

Já para encontrar a posição da garra,  $\beta$  e  $\gamma$  em função de  $90^\circ < \alpha < 145^\circ$  e  $0^\circ < \theta < 35^\circ$  pode-se incluir o código abaixo:

```

L0 = 3.5;
L1 = 8.1;
L2 = 3.5;
L3 = 8.1;

alfa = 90*pi/180:pi/36:135*pi/180;
theta = 0*pi/180:pi/36:35*pi/180;

beta = zeros(length(theta),length(alfa));
gama = zeros(length(theta),length(alfa));

for i = 1:length(theta)
    X0 = [pi/4;
          pi/4];

    for j = 1:length(alfa)
        for k = 1:Nmax
            beta(i,j) = X0(1,1);
            gama(i,j) = X0(2,1);

            f = [-L0*cos(theta(i)) + L3*cos(beta(i,j)) + L2*cos(
                gama(i,j)) - L1*cos(alfa(j));
                L0*sin(theta(i)) + L3*sin(beta(i,j)) - L2*sin(
                gama(i,j)) - L1*sin(alfa(j))];

            J = [-L3*sin(beta(i,j)), -L2*sin(gama(i,j));
                L3*cos(beta(i,j)), -L2*cos(gama(i,j))];

```

```
X = X0 - J\f;

if((abs(X(1,1) - X0(1,1)) < erro) && (abs(X(2,1) - X0
    (2,1)) < erro))
    break;
end

X0 = X;
end
end
end

L4 = 8.1;
L5 = 1.2;
L6 = 5.3;

for i = 1:length(theta)
    for j = 1:length(alfa)

        xB = -L0*cos(theta(i));
        yB = L0*sin(theta(i));
        xC = L3*cos(beta(i,j));
        yC = L3*sin(beta(i,j));
        xD = L1*cos(alfa(j));
        yD = L1*sin(alfa(j));
        xE = L4*cos(gama(i,j));
        yE = -L4*sin(gama(i,j));
        xF = 0;
        yF = -1.2;

        XG = L1*cos(beta) + L4*cos(gama) + L6*ones(8,10);
        YG = L1*sin(beta) - L4*sin(gama) - L5*ones(8,10);

        figure(4)
        plot(-1.25,-1.5,'ks','markersize',70,'markerfacecolor',
            ,[0.85 0.85 0.85])
        hold on
        plot([0 xB],[0 yB],'r-','linewidth',3)
        plot([xB xB+xC],[yB yB+yC],'b-','linewidth',3)
        plot([xB+xC xD],[yB+yC yD],'b-','linewidth',3)
        plot([0 xD],[0 yD],'b-','linewidth',3)
        plot([xD xD+xE],[yD yD+yE],'b-','linewidth',3)
```

---

```

plot([xD+xE xD+xE+xF],[yD+yE yD+yE+yF], 'b-', 'linewidth',
     ,2)
plot([xD+xE+xF XG(i,j)],[yD+yE+yF YG(i,j)], 'r-', '
     linewidth',2)
plot(0,-0.3, 'k^', 'markersize',7, 'markerfacecolor',[0.6
     0.6 0.6])
plot(0,0, 'ko', 'markersize',4, 'markerfacecolor','w')
plot(xB,yB, 'ko', 'markersize',4, 'markerfacecolor','w')
plot(xB+xC,yB+yC, 'ko', 'markersize',4, 'markerfacecolor','w
     ')
plot(xD,yD, 'ko', 'markersize',4, 'markerfacecolor','w')
plot(xD+xE,yD+yE, 'ko', 'markersize',4, 'markerfacecolor','w
     ')
plot(xD+xE+xF,yD+yE+yF, 'ko', 'markersize',3, '
     markerfacecolor','w')
plot(XG(i,j),YG(i,j), 'ro', 'markersize',6, 'markerfacecolor
     ',[0.8 0.3 0.3])
plot(XG(1:i,1:j),YG(1:i,1:j), 'k.')
hold off
axis([-10 25 -10 25])
axis square
xlabel('coordenada x_A (cm)')
ylabel('coordenada y_A (cm)')
title('Posicoes da garra do Braco Robotico')
grid on
pause(0.1)
end
end

```



## APÊNDICE B – CÓDIGO NO MATLAB DA VELOCIDADE E ACELERAÇÃO DA GARRA

A velocidade e aceleração da garra com  $\dot{\phi} = 0$ ,  $\ddot{\phi} = 0$ ,  $\dot{\alpha} = 0.8726$ ,  $\ddot{\alpha} = 0$ ,  $\dot{\theta} = 0$  e  $\ddot{\theta} = 0$  está descrita no código de MATLAB abaixo:

```

clc;
clear;

L0 = 3.5;
L1 = 8.1;
L2 = 3.5;
L3 = 8.1;
L4 = 8.1;
L5 = 1.2;
L6 = 5.3;

alfa = 5*pi/180:pi/36:90*pi/180;
theta = 0*pi/180:pi/36:80*pi/180;

V_alfa = 0.8726;
V_theta = 0;

[Alfa,Theta] = meshgrid(alfa,theta);

Vx = -V_alfa*L1*sin(Alfa);
Vy = V_alfa*L1*cos(Alfa);

V = sqrt(Vx.^2 + Vy.^2);

figure(1)
subplot(2,2,1)
plot(alfa*180/pi,Vx,'linewidth',2)
title('Velocidade na coordenada x_A da Garra')
xlabel('\alpha (graus)'),ylabel('Velocidade em x (cm/s)')
axis([0 150 -8 0])
axis square
grid on

subplot(2,2,2)
plot(alfa*180/pi,Vy,'linewidth',2)

```

```
title('Velocidade na coordenada y_A da Garra')
xlabel('\alpha (graus)'),ylabel('Velocidade em y (cm/s)')
axis([0 150 -5 7])
axis square
grid on

subplot(2,2,3)
plot(alfa*180/pi,V,'g-', 'linewidth',2)
title('Velocidade da Garra')
xlabel('\alpha (graus)'),ylabel('Velocidade (cm/s)')
axis([0 135 0 8])
axis square
grid on

    ax = -(V_alfa^2)*L1*cos(Alfa);
    ay = -(V_alfa^2)*L1*sin(Alfa);

    a = sqrt(ax.^2 + ay.^2);

figure(2)
subplot(2,2,1)
plot(alfa*180/pi,ax,'linewidth',2)
title('Aceleracao na coordenada x_A da Garra')
xlabel('\alpha (graus)'),ylabel('Aceleracao em x (cm/s^2)')
axis([0 150 -7 5])
axis square
grid on

subplot(2,2,2)
plot(alfa*180/pi,ay,'linewidth',2)
title('Aceleracao na coordenada y_A da Garra')
xlabel('\alpha (graus)'),ylabel('Aceleracao em y (cm/s^2)')
axis([0 150 -8 0])
axis square
grid on

subplot(2,2,3)
plot(alfa*180/pi,a,'r-', 'linewidth',2)
title('Aceleracao da Garra')
xlabel('\alpha (graus)'),ylabel('Aceleracao (cm/s^2)')
axis([0 135 0 8])
axis square
```

```
grid on
```

```
for i = 1:length(theta)
    for j = 1:length(alfa)

        xB = -L0*cos(theta(i));
        yB = L0*sin(theta(i));
        xC = L3*cos(Alfa(i,j));
        yC = L3*sin(Alfa(i,j));
        xD = L1*cos(alfa(j));
        yD = L1*sin(alfa(j));
        xE = L4*cos(Theta(i,j));
        yE = -L4*sin(Theta(i,j));
        xF = 0;
        yF = -1.2;

        xG = L1*cos(Alfa) + L4*cos(Theta) + L6*ones(17,18);
        yG = L1*sin(Alfa) - L4*sin(Theta) - L5*ones(17,18);

        figure(3)
        plot(-1.35,-1.5,'ks','markersize',70,'markerfacecolor',
            ,[0.85 0.85 0.85])
        hold on
        plot([0 xB],[0 yB],'b-','linewidth',3)
        plot([xB xB+xC],[yB yB+yC],'b-','linewidth',3)
        plot([xB+xC xD],[yB+yC yD],'b-','linewidth',3)
        plot([0 xD],[0 yD],'c-','linewidth',3)
        plot([xD xD+xE],[yD yD+yE],'b-','linewidth',3)
        plot([xD+xE xD+xE+xF],[yD+yE yD+yE+yF],'b-','linewidth',
            ,2)
        plot([xD+xE+xF xG(i,j)],[yD+yE+yF yG(i,j)'],'c-','
            linewidth',2)
        plot(0,-0.3,'k^','markersize',7,'markerfacecolor',[0.6
            0.6 0.6])
        plot(0,0,'ko','markersize',4,'markerfacecolor','w')
        plot(xB,yB,'ko','markersize',4,'markerfacecolor','w')
        plot(xB+xC,yB+yC,'ko','markersize',4,'markerfacecolor','w'
            ')
        plot(xD,yD,'ko','markersize',4,'markerfacecolor','w')
        plot(xD+xE,yD+yE,'ko','markersize',4,'markerfacecolor','w'
            ')
    end
end
```

```

plot(xD+xE+xF,yD+yE+yF,'ko','markersize',3,'
      markerfacecolor','w')
plot(xG(i,j),yG(i,j),'co','markersize',6,'markerfacecolor
     ',[0.3 0.3 0.8])
plot(xG(i,1:j),yG(i,1:j),'k.')
pVel = plot([xG(i,j) xG(i,j)+Vx(i,j)/2],[yG(i,j) yG(i,j)+
      Vy(i,j)/2],'g-');
plot(xG(i,j)+Vx(i,j)/2,yG(i,j)+Vy(i,j)/2,'g<','markersize
      ',2)
pAce =plot([xG(i,j) xG(i,j)+ax(i,j)/2],[yG(i,j) yG(i,j)+
      ay(i,j)/2],'r-');
plot(xG(i,j)+ax(i,j)/2,yG(i,j)+ay(i,j)/2,'r<','markersize
      ',2)
hold off
text(-1.2,-0.1,'A')
text(-2.5,-4.5,'0')
legend([pVel pAce],{'Velocidade 1:2','Aceleracao 1:2'})
axis([-10 25 -10 25])
axis square
xlabel('coordenada x (cm)')
ylabel('coordenada y (cm)')
title('Posicoes da garra do Braco Robotico')
grid on
pause(0.07)
end
end

figure(4)
mesh(xG,yG,Vx)
title('Velocidade na coordenada x da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Velocidade em x (cm/s)'
)
axis square
colormap jet

figure(5)
mesh(xG,yG,Vy)
title('Velocidade na coordenada y da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Velocidade em y (cm/s)'
)
axis square
colormap jet

```

```
figure(6)
mesh(xG,yG,V)
title('Velocidade da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Velocidade (cm/s)')
axis square
colormap jet

figure(7)
mesh(xG,yG,ax)
title('Aceleracao na coordenada x da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Aceleracao em x (cm/s
^2)')
axis square
colormap jet

figure(8)
mesh(xG,yG,ay)
title('Aceleracao na coordenada y da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Aceleracao em y (cm/s
^2)')
axis square
colormap jet

figure(9)
mesh(xG,yG,a)
title('Aceleracao da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Aceleracao (cm/s^2)')
axis square
colormap jet
```

A velocidade e aceleração da garra com  $\dot{\phi} = 0$ ,  $\ddot{\phi} = 0$ ,  $\dot{\alpha} = 0$ ,  $\ddot{\alpha} = 0$ ,  $\dot{\theta} = 0.8726$  e  $\ddot{\theta} = 0$  está descrita no código de MATLAB abaixo:

```
clc;
clear;

L0 = 3.5;
L1 = 8.1;
L2 = 3.5;
L3 = 8.1;
```

```
L4 = 8.1;
L5 = 1.2;
L6 = 5.3;

alfa = 5*pi/180:pi/36:90*pi/180;
theta = 0*pi/180:pi/36:80*pi/180;

V_alfa = 0;
V_theta = 0.8726;

[Alfa,Theta] = meshgrid(alfa,theta);

    Vx = - V_theta*L4*sin(Theta);
    Vy = - V_theta*L4*cos(Theta);
    V = sqrt(Vx.^2 + Vy.^2);

figure(1)
subplot(2,2,1)
plot(theta*180/pi,Vx,'linewidth',2)
title('Velocidade na coordenada x_A da Garra')
xlabel('\theta (graus)'),ylabel('Velocidade em x (cm/s)')
axis([0 90 -8 0])
axis square
grid on

subplot(2,2,2)
plot(theta*180/pi,Vy,'linewidth',2)
title('Velocidade na coordenada y_A da Garra')
xlabel('\theta (graus)'),ylabel('Velocidade em y (cm/s)')
axis([0 90 -8 0])
axis square
grid on

subplot(2,2,3)
plot(theta*180/pi,V,'g-','linewidth',2)
title('Velocidade da Garra')
xlabel('\theta (graus)'),ylabel('Velocidade (cm/s)')
axis([0 90 0 8])
axis square
grid on

    ax = -(V_theta^2)*L4*cos(Theta);
```

```
ay = (V_theta^2)*L4*sin(Theta);
a = sqrt(ax.^2 + ay.^2);

figure(2)
subplot(2,2,1)
plot(theta*180/pi,ax,'linewidth',2)
title('Aceleracao na coordenada x_A da Garra')
xlabel('\theta (graus)'),ylabel('Aceleracao em x (cm/s^2)')
axis([0 90 -8 0])
axis square
grid on

subplot(2,2,2)
plot(theta*180/pi,ay,'linewidth',2)
title('Aceleracao na coordenada y_A da Garra')
xlabel('\theta (graus)'),ylabel('Aceleracao em y (cm/s^2)')
axis([0 90 0 8])
axis square
grid on

subplot(2,2,3)
plot(theta*180/pi,a,'r-','linewidth',2)
title('Aceleracao da Garra')
xlabel('\theta (graus)'),ylabel('Aceleracao (cm/s^2)')
axis([0 90 0 8])
axis square
grid on

for i = 1:length(alfa)
    for j = 1:length(theta)

        xB = -L0*cos(theta(j));
        yB = L0*sin(theta(j));
        xC = L3*cos(Alfa(j,i));
        yC = L3*sin(Alfa(j,i));
        xD = L1*cos(alfa(i));
        yD = L1*sin(alfa(i));
        xE = L4*cos(Theta(j,i));
        yE = -L4*sin(Theta(j,i));
        xF = 0;
        yF = -1.2;
```

```

xG = L1*cos(Alfa) + L4*cos(Theta) + L6*ones(17,18);
yG = L1*sin(Alfa) - L4*sin(Theta) - L5*ones(17,18);

figure(3)
plot(-1.35,-1.5,'ks','markersize',70,'markerfacecolor',
     ,[0.85 0.85 0.85])
hold on
plot([0 xB],[0 yB],'c-','linewidth',3)
plot([xB xB+xC],[yB yB+yC],'b-','linewidth',3)
plot([xB+xC xD],[yB+yC yD],'b-','linewidth',3)
plot([0 xD],[0 yD],'b-','linewidth',3)
plot([xD xD+xE],[yD yD+yE],'b-','linewidth',3)
plot([xD+xE xD+xE+xF],[yD+yE yD+yE+yF],'b-','linewidth',
     ,2)
plot([xD+xE+xF xG(j,i)],[yD+yE+yF yG(j,i)'],'c-','
     linewidth',2)
plot(0,-0.3,'k^','markersize',7,'markerfacecolor',[0.6
     0.6 0.6])
plot(0,0,'ko','markersize',4,'markerfacecolor','w')
plot(xB,yB,'ko','markersize',4,'markerfacecolor','w')
plot(xB+xC,yB+yC,'ko','markersize',4,'markerfacecolor','w
     ')
plot(xD,yD,'ko','markersize',4,'markerfacecolor','w')
plot(xD+xE,yD+yE,'ko','markersize',4,'markerfacecolor','w
     ')
plot(xD+xE+xF,yD+yE+yF,'ko','markersize',3,'
     markerfacecolor','w')
plot(xG(j,i),yG(j,i),'co','markersize',6,'markerfacecolor
    ',[0.3 0.3 0.8])
plot(xG(1:j,i),yG(1:j,i),'k.')
pVel = plot([xG(j,i) xG(j,i)+Vx(j,i)/2],[yG(j,i) yG(j,i)+
     Vy(j,i)/2],'g-');
plot(xG(j,i)+Vx(j,i)/2,yG(j,i)+Vy(j,i)/2,'g<','markersize
     ',2)
pAce =plot([xG(j,i) xG(j,i)+ax(j,i)/2],[yG(j,i) yG(j,i)+
     ay(j,i)/2],'r-');
plot(xG(j,i)+ax(j,i)/2,yG(j,i)+ay(j,i)/2,'r^','markersize
     ',2)
hold off
text(-1.2,-0.1,'A')
text(-2.5,-4.5,'0')

```

```
        legend([pVel pAce],{'Velocidade 1:2','Aceleracao 1:2'})
        axis([-10 25 -10 25])
        axis square
        xlabel('coordenada x (cm)')
        ylabel('coordenada y (cm)')
        title('Posicoes da garra do Braco Robotico')
        grid on
        pause(0.07)
    end
end

figure(4)
mesh(xG,yG,Vx)
title('Velocidade na coordenada x da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Velocidade em x (cm/s)')
)
axis square
colormap jet

figure(5)
mesh(xG,yG,Vy)
title('Velocidade na coordenada y da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Velocidade em y (cm/s)')
)
axis square
colormap jet

figure(6)
mesh(xG,yG,V)
title('Velocidade da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Velocidade (cm/s)')
axis square
colormap jet

figure(7)
mesh(xG,yG,ax)
title('Aceleracao na coordenada x da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Aceleracao em x (cm/s
^2)')
axis square
colormap jet
```

```
figure(8)
mesh(xG,yG,ay)
title('Aceleracao na coordenada y da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Aceleracao em y (cm/s
    ^2)')
axis square
colormap jet
```

```
figure(9)
mesh(xG,yG,a)
title('Aceleracao da Garra')
xlabel('x (cm)'),ylabel('y (cm)'),zlabel('Aceleracao (cm/s^2)')
axis square
colormap jet
```