



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**LUAN GOMES MAGALHÃES LIMA**

**VIRTUALIZAÇÃO DE UM MANIPULADOR ROBÓTICO DE 5 JUNTAS ROTATIVAS**

**SOBRAL**

**2023**

LUAN GOMES MAGALHÃES LIMA

VIRTUALIZAÇÃO DE UM MANIPULADOR ROBÓTICO DE 5 JUNTAS ROTATIVAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Me. David Nascimento Coelho

SOBRAL

2023

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

M167v Magalhães Lima, Luan Gomes.  
Virtualização de um Manipulador Robótico de 5 Juntas Rotativas / Luan Gomes Magalhães Lima. – 2023.  
62 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral,  
Curso de Engenharia da Computação, Sobral, 2023.  
Orientação: Prof. Me. David Nascimento Coelho.

1. Gêmeos Digitais. 2. Indústria 4.0. 3. Manipulador Robótico. 4. Virtualização. I. Título.

CDD 621.39

---

LUAN GOMES MAGALHÃES LIMA

VIRTUALIZAÇÃO DE UM MANIPULADOR ROBÓTICO DE 5 JUNTAS ROTATIVAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 14 de Dezembro de 2023

BANCA EXAMINADORA

---

Prof. Me. David Nascimento Coelho (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Ícaro Bezerra Viana  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Reuber Régis de Melo  
Universidade Federal do Ceará (UFC)

À minha família, especialmente ao meu pai, mãe, irmã e irmão, pela capacidade de acreditar em mim e investir em mim. O apoio e dedicação deles foram cruciais para eu persistir em meus sonhos e objetivos.

## **AGRADECIMENTOS**

Gostaria de agradecer a todas as pessoas que me ajudaram a chegar até aqui, as contribuições de cada uma delas em minha vida foram essenciais para a conclusão deste trabalho.

Em especial, aos meus pais, Maria José e Francisco Waldemyr, por toda dedicação, companheirismo e apoio, os quais foram cruciais ao longo dessa jornada, motivando-me a persistir na busca dos meus objetivos. Aos meus irmãos, Emily e Ryan, pela confiança e companheirismo que compartilhamos.

Aos professores que tive ao longo do curso, em especial ao Prof. Dr. Reuber Régis, com quem tive a oportunidade de trabalhar inicialmente neste projeto, ao Prof. Me. David Nascimento pela excelente orientação e contribuições acadêmicas e pessoais, e ao Prof. Dr. Marcelo Marques pelo apoio e dedicação durante o período das monitorias. Também gostaria de agradecer ao Prof. Dr. Iális Cavalcante e ao Prof. Dr. Wendley Souza pela excelente orientação durante meu tempo de bolsa PET.

Aos professores participantes da banca examinadora, Prof. Dr. Reuber Régis e Prof. Dr. Ícaro Bezerra, por terem aceitado o convite e por toda compreensão.

Agradeço a todos os amigos e colegas com quem tive a honra de trabalhar e compartilhar experiências únicas durante o período em que participei da bolsa PET. As conversas e experiências compartilhadas contribuíram profundamente para minha carreira acadêmica e pessoal.

E, por fim, gostaria de agradecer a todas as amigas que fiz ao longo do curso por todo o apoio, companheirismo e troca de experiências.

"Concentre todos seus pensamentos na tarefa que está realizando. Os raios de sol não queimam até que sejam colocados em foco."

(Alexander Graham Bell)

## RESUMO

A Indústria 4.0 impulsiona o surgimento de diversas tecnologias e ferramentas, pautadas na automatização dos processos e na troca de informações entre os diversos dispositivos. Dentre as tecnologias vigentes, tem-se, por exemplo, a Internet das Coisas, Inteligência Artificial e Realidade Virtual. Nesse contexto, os sistemas robóticos, principalmente os manipuladores robóticos, estão sendo utilizados para desempenhar tarefas críticas que possam comprometer a segurança dos operadores humanos. Logo, esses dispositivos participam ativamente do processo industrial. Entretanto, os riscos associados à operação, custos elevados e a necessidade de intervenção humana representam os principais desafios para a inserção de sistemas robóticos. Diante disso, a virtualização emerge como uma solução promissora, visando reforçar a segurança, reduzir custos e possibilitar o estudo desses dispositivos. Sendo assim, o presente trabalho busca virtualizar um manipulador robótico didático, de 5 juntas rotativas, proporcionando um ambiente virtual interativo, por meio de uma interface gráfica, para realizar estudos e testes no manipulador em questão. A virtualização é utilizada em conjunto com os gêmeos digitais, pois estes representam o "gêmeo" virtual de algum processo industrial ou dispositivo físico. Logo, objetiva-se a criação do gêmeo digital que represente o braço robótico físico. Existem inúmeras ferramentas para a virtualização. O presente trabalho baseia-se em duas principais: o software de modelagem 3D Autodesk Inventor e o software de computação numérica e programação Matlab. Para conectar o modelo real com o modelo virtual, foi utilizado o protocolo de comunicação serial, cuja integração foi realizada por meio da plataforma Arduino. Esse tipo de protocolo se mostrou adequado, pois proporciona baixa latência, precisão e é ideal para comunicação de curta distância. Os resultados obtidos neste trabalho ressaltam a importância da virtualização para o estudo de dispositivos físicos, em especial os manipuladores robóticos, sem a necessidade de implementação física.

**Palavras-chave:** Gêmeos Digitais. Indústria 4.0. Manipulador Robótico. Virtualização.



## ABSTRACT

Industry 4.0 is driving the emergence of various technologies and tools based on automating processes and exchanging information between different devices. Current technologies include, for example, the Internet of Things, Artificial Intelligence and Virtual Reality. In this context, robotic systems, especially robotic manipulators, are being used to perform critical tasks that could compromise the safety of human operators. These devices are therefore taking an active part in the industrial process. However, the risks associated with operation, high costs and the need for human intervention represent the main challenges for the insertion of robotic systems. In view of this, virtualization has emerged as a promising solution aimed at enhancing safety, reducing costs and making it possible to study these devices. Therefore, this work aims to virtualize a didactic robotic manipulator with 5 rotary joints, providing an interactive virtual environment, through a graphic interface, to carry out studies and tests on the manipulator in question. Virtualization is used in conjunction with digital twins, as these represent the virtual "twin" of an industrial process or physical device. The aim is therefore to create a digital twin that represents the physical robotic arm. There are numerous tools for virtualization. This work is based on two main ones: the 3D modeling software Autodesk Inventor and the numerical computing and programming software Matlab. The serial communication protocol was used to connect the real model to the virtual model, which was integrated using the Arduino platform. This type of protocol proved to be suitable, as it provides low latency, precision and is ideal for short-distance communication. The results obtained in this work highlight the importance of virtualization for studying physical devices, especially robotic manipulators, without the need for physical implementation.

**Keywords:** Digital Twins. Industry 4.0. Robotic manipulator. Virtualization.

## LISTA DE FIGURAS

Figura 1 – Exemplo de um Manipulador Robótico . . . . .	21
Figura 2 – Estrutura de um Manipulador Robótico . . . . .	23
Figura 3 – Principais juntas utilizadas na indústrias . . . . .	23
Figura 4 – Vetor posição . . . . .	24
Figura 5 – Representação do vetor que relaciona a origem de {B} com {A} . . . . .	25
Figura 6 – Frames de referência para um manipulador com 3 juntas . . . . .	27
Figura 7 – Fluxograma do trabalho . . . . .	31
Figura 8 – Manipulador robótico de 5 juntas rotativas . . . . .	31
Figura 9 – Servo MG995 . . . . .	32
Figura 10 – Micro Servo MG90S . . . . .	32
Figura 11 – Arduíno UNO . . . . .	38
Figura 12 – Blocos utilizados para a comunicação serial . . . . .	38
Figura 13 – Webcam Goldentec . . . . .	39
Figura 14 – Braço robótico construído no Inventor . . . . .	41
Figura 15 – Modelo virtual no Simulink . . . . .	42
Figura 16 – Junta Rotativa . . . . .	43
Figura 17 – Blocos de entrada na junta rotativa . . . . .	43
Figura 18 – Propriedades da junta rotativa . . . . .	44
Figura 19 – Blocos de saída na junta rotativa . . . . .	44
Figura 20 – Execução do modelo no Simulink . . . . .	45
Figura 21 – Simulação do Modelo Virtual . . . . .	46
Figura 22 – Modelo de Comunicação Serial com apenas um Servo Motor . . . . .	47
Figura 23 – Protótipo da comunicação serial . . . . .	47
Figura 24 – Modelo para Comunicação Serial das 5 juntas rotativas . . . . .	48
Figura 25 – Modelo Virtual do Manipulador Robótico no Simulink . . . . .	49
Figura 26 – Interface criada no Matlab . . . . .	50
Figura 27 – Funcionamento da interface e comparação com o modelo real . . . . .	51

## LISTA DE TABELAS

Tabela 1 – Parâmetros DH de cada junta . . . . .	33
Tabela 2 – Tamanhos dos elos . . . . .	34

## LISTA DE ABREVIATURAS E SIGLAS

AR	<i>Augmented Reality</i>
CPS	<i>Cyber Physics System</i>
DH	<i>Denavit-Hartenberg</i>
GUI	<i>Graphical User Interface</i>
GUIDE	<i>Graphical User Interface Development Environment</i>
IoT	<i>Internet of Things</i>
VR	<i>Virtual Reality</i>

## LISTA DE SÍMBOLOS

${}^A P$	Descrição da posição de um ponto P em relação ao quadro {A}
${}^B P$	Descrição da posição de um ponto P em relação ao quadro {B}
${}^A_B P$	Descrição da posição de um ponto P no quadro {B} em relação ao quadro {A}
${}^A_B R$	Matriz de rotação que descreve a transformação do quadro {B} em relação ao quadro {A}
${}^A X_B$	Vetores unitários do eixo X do quadro {B} em relação ao quadro {A}
${}^A Y_B$	Vetores unitários do eixo Y do quadro {B} em relação ao quadro {A}
${}^A Z_B$	Vetores unitários do eixo Z do quadro {B} em relação ao quadro {A}
{A}	Sistema de referência ( <i>frame</i> ) A
{B}	Sistema de referência ( <i>frame</i> ) B
${}^A_B T$	Matriz de transformação do quadro {B} em relação ao quadro {A}
$\theta$	Ângulo de articulação medido sobre o eixo Z
$\alpha$	Ângulo de torção medido sobre o eixo X
$a$	Deslocamento medido sobre o eixo X
$d$	Deslocamento medido sobre o eixo Z
$V$	Tensão

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
<b>1.1</b>	<b>Objetivos</b>	<b>16</b>
<b>1.1.1</b>	<i>Objetivo Geral</i>	<b>16</b>
<b>1.1.2</b>	<i>Objetivos Específicos</i>	<b>16</b>
<b>1.2</b>	<b>Trabalhos Relacionados</b>	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>18</b>
<b>2.1</b>	<b>Indústria 4.0</b>	<b>18</b>
<b>2.2</b>	<b>Diferença entre Simulação e Virtualização</b>	<b>19</b>
<b>2.3</b>	<b>Gêmeos Digitais</b>	<b>20</b>
<b>2.4</b>	<b>Robótica</b>	<b>20</b>
<b>2.5</b>	<b>Cinemática Robótica</b>	<b>23</b>
<b>2.5.1</b>	<i>Descrição espacial</i>	<b>24</b>
<b>2.5.2</b>	<i>Cinemática Direta</i>	<b>26</b>
<b>2.5.3</b>	<i>Cinemática Inversa</i>	<b>28</b>
<b>2.6</b>	<b>Softwares de Simulação</b>	<b>28</b>
<b>2.7</b>	<b>Protocolos de Comunicação</b>	<b>29</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>31</b>
<b>3.1</b>	<b>Manipulador Robótico</b>	<b>31</b>
<b>3.2</b>	<b>Cinemática do Manipulador Robótico</b>	<b>33</b>
<b>3.3</b>	<b>Modelo Virtual</b>	<b>34</b>
<b>3.3.1</b>	<i>Inventor</i>	<b>34</b>
<b>3.3.2</b>	<i>Matlab</i>	<b>35</b>
<b>3.3.2.1</b>	<i>Integração entre softwares</i>	<b>36</b>
<b>3.3.2.2</b>	<i>Interface Gráfica</i>	<b>36</b>
<b>3.4</b>	<b>Comunicação Serial</b>	<b>37</b>
<b>3.4.1</b>	<i>Codificação</i>	<b>38</b>
<b>3.5</b>	<b>Monitoramento do Modelo Real</b>	<b>39</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>41</b>
<b>4.1</b>	<b>Modelagem 3D</b>	<b>41</b>
<b>4.2</b>	<b>Modelo Virtual</b>	<b>42</b>

4.3	<b>Simulação</b> . . . . .	45
4.4	<b>Integração entre os Modelos Real e Virtual</b> . . . . .	46
4.5	<b>Modelo Virtual Completo</b> . . . . .	49
4.6	<b>Interface Gráfica</b> . . . . .	50
4.7	<b>Funcionamento</b> . . . . .	51
5	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	52
	<b>REFERÊNCIAS</b> . . . . .	53
	<b>APÊNDICES</b> . . . . .	55
	<b>APÊNDICE A – Código para implementação da Cinemática Direta do Manipulador Robótico</b> . . . . .	55
	<b>ANEXOS</b> . . . . .	56
	<b>ANEXO A – Datasheet do Servo Motor MG995</b> . . . . .	57
	<b>ANEXO B – Datasheet do Servo Motor MG90S</b> . . . . .	60

## 1 INTRODUÇÃO

Na Indústria 4.0, termo utilizado para caracterizar a quarta revolução industrial, os diversos dispositivos e equipamentos industriais, tais como máquinas e módulos de produção, interagem entre si e trocam informações com o meio, fazendo com que todos esses dispositivos estejam interligados (WEYER *et al.*, 2015). Esse novo panorama industrial traz consigo inúmeros desafios e oportunidades, possibilitando métodos de produção mais eficientes. Dentre as novas tecnologias presentes, podem ser citadas a Internet das Coisas, a Inteligência Artificial, a Realidade Virtual e a Robótica Avançada (TAKAYAMA; PANHAM, 2022).

Nesse contexto, a automatização está presente nos diversos setores industriais, e vem acompanhada de uma utilização crescente de robôs para desempenhar tarefas críticas ou insalubres que possam comprometer a vida dos operadores humanos (TABACOW, 2017). Dentre os tipos de robôs, destacam-se os manipuladores robóticos, que não apenas realizam tarefas simples, mas também participam dos processos industriais nos quais estão inseridos.

Porém, é importante considerar as exigências e os cuidados necessários ao trabalhar com os robôs (WEYER *et al.*, 2015), visto que, muitas vezes, ainda é necessário a supervisão e controle por parte de operadores (PENHA *et al.*, 2016), e existe um grande risco de acidentes (TAKAYAMA; PANHAM, 2022). Também, o elevado custo de produção e manutenção, e a necessidade de contar com aparato técnico e tecnologia avançada, podem ser mencionados como os principais custos envolvendo essa ferramenta. Além disso, existe a limitação em responder a emergências ou alterações externas (NIKU, 2020).

Diante dessas restrições, a virtualização emerge como uma estratégia para reforçar a segurança e reduzir os custos associados à implementação de robôs, ganhando, assim, mais destaque no âmbito industrial. Takayama e Panham (2022) definem simulação virtual, ou gêmeos digitais, como um mecanismo capaz de "imitar" algum objeto ou sistema do mundo real, utilizando um ambiente virtual. A utilização da simulação virtual em conjunto com manipuladores robóticos, conforme mencionado por Penha (2016), apresenta vantagens notáveis, como a redução de custos e o aumento da produtividade.

Nesse contexto, a criação de um gêmeo digital, a partir de um modelo físico, proporciona uma compreensão didática do funcionamento de um modelo real, eliminando a necessidade de sua implementação física. Além dos benefícios econômicos diretos no setor industrial, essa abordagem possibilita a realização de estudos no campo da robótica.



## 1.1 Objetivos

### 1.1.1 *Objetivo Geral*

Virtualizar um manipulador robótico didático, de 5 juntas rotativas, por meio de simulação computacional, possibilitando uma maior segurança para o estudo deste.

### 1.1.2 *Objetivos Específicos*

- Revisar a bibliografia existente sobre a virtualização dos manipuladores robóticos e suas aplicações;
- Construir o modelo 3D, a partir do braço robótico real;
- Implementar o modelo do manipulador em um ambiente de simulação computacional;
- Implementar a comunicação entre o modelo virtual e o modelo real.

## 1.2 Trabalhos Relacionados

No trabalho de Penha (2016), o autor menciona a simulação virtual de manipuladores robóticos como uma estratégia eficaz de reduzir custos e aumentar a produtividade em um cenário globalizado. Diante disso, ele utilizou o software SolidWorks para a construção virtual de um manipulador, e o software Matlab para realizar análises e testes de simulação. Em razão disso, observou-se que a programação *off-line* permite obter importantes informações sobre o manipulador utilizado, como as posições desejadas, a força, a trajetória, dentre outras. Como resultado, o autor concluiu que a virtualização reduz as chances de falhas e tornam o processo produtivo mais confiável.

Segundo Tabacow (2017), a simulação desempenha um papel crucial na indústria, fornecendo a capacidade de modelar sistemas robóticos em um ambiente virtual controlado, onde é possível observar antecipadamente as métricas de desempenho almejadas. Algumas ferramentas utilizadas pelo autor daquele trabalho auxiliam na implementação da virtualização, com destaque para o Matlab e a biblioteca *Robotics Toolbox* desenvolvida por Peter Corke. Essa biblioteca se destaca pela facilidade de planejamento e mapeamento de trajetórias do manipulador robótico virtual. Com isso, as simulações realizadas possibilitaram a visualização do comportamento do equipamento quando submetido à programação estabelecida.

No trabalho de Neto *et al.* (2021), a modelagem computacional de um manipulador robótico controlado por CLP foi realizada por meio do software *Rhinoceuros 3D*, permitindo trabalhar com as dimensões reais e o material constituinte do manipulador utilizado. Já o software Matlab foi útil para implementação da cinemática direta e das simulações. Os resultados apresentados naquele estudo destacam a complexidade envolvida na modelagem da cinemática e dinâmica dos manipuladores, abrangendo desde a modelagem matemática até a programação e geração de trajetórias.

Os trabalhos mencionados ressaltam a importância da virtualização na indústria, especialmente com a ascensão da Indústria 4.0 e a crescente demanda por tecnologia e eficiência. Os requisitos do mercado atual exigem produtos de alta qualidade e prazos reduzidos, o que torna essencial o uso de ferramentas que auxiliem no processo de produção industrial, visando garantir a segurança e aprimoramento desse processo. Diante disso, a importância do presente trabalho reside no fato de que a virtualização de um determinado manipulador robótico pode facilitar o estudo, aprendizado e aplicação deste, trazendo redução de riscos e custos.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os aspectos teóricos fundamentais para o desenvolvimento deste trabalho. Inicialmente, será explorada a revolução da Indústria 4.0, na qual ocorre a integração de tecnologias como a Internet das Coisas e Sistemas Ciberfísicos. Em seguida, será discutida a importância da simulação e virtualização para aprimorar processos industriais, especialmente o uso de manipuladores robóticos. Além disso, serão introduzidos conceitos de cinemática robótica, fundamentais para o controle e programação de robôs. Por fim, serão apresentadas as principais ferramentas de simulação empregadas na robótica e os principais protocolos de comunicação utilizados para a comunicação entre o modelo real e o modelo virtual, estabelecendo as bases para possíveis aplicações práticas.

### 2.1 Indústria 4.0

O modelo de produção e as técnicas de fabricação atuais estão necessitando cada vez mais de recursos inteligentes e que estejam conectados entre si. Nesse âmbito, surge o termo Indústria 4.0, o qual traz consigo uma nova revolução industrial, cujo diferencial é a inserção da Internet das Coisas (do inglês, *Internet of Things* (IoT)) e dos Sistemas Cyber Físicos (do inglês, *Cyber Physics System* (CPS)) nos meios de produção (WEYER *et al.*, 2015). Nesse novo paradigma industrial, a ideia é garantir que todos os dispositivos compartilhem informações entre si, permitindo um maior controle e gerenciamento das atividades industriais.

Além disso, esse novo panorama requer uma maior automatização e padronização dos processos e dos produtos. Diante disso, o termo IoT refere-se a um dos pilares da 4ª Revolução Industrial e vem sendo implementado nos meios industriais. Esse processo faz alusão a capacidade de conectar todos os dispositivos por meio da internet, possibilitando a troca de informações entre os dispositivos conectados em uma mesma rede (BAHGA; MADISSETTI, 2014).

Nos moldes de produção tradicional, é comum haver uma discrepância entre o mundo real e o digital. Contudo, essa diferença tem diminuído à medida que a Indústria 4.0 revoluciona o modelo de produção atual (WEYER *et al.*, 2015). A introdução de novas tecnologias no cenário industrial está impulsionando a concepção de fábricas inteligentes, nas quais a interconexão entre o mundo físico e o mundo digital é essencial. Essa integração é possibilitada pelos CPS, que unem a computação com os componentes físicos do ambiente fabril (WEYER *et al.*, 2016).

À medida que a Indústria 4.0 avança, os CPS desempenham um papel fundamental na eliminação das barreiras entre os domínios real e digital, criando um ambiente de produção integrado e inteligente. Nesse cenário, diversas aplicações têm se destacado, incluindo a incorporação de sistemas robóticos nos processos de produção, a utilização de Realidade Aumentada (do inglês, *Augmented Reality* (AR)) e Realidade Virtual (do inglês, *Virtual Reality* (VR)), bem como a análise e coleta de dados em grande escala (TAKAYAMA; PANHAM, 2022). Essa abordagem visa proporcionar uma colaboração entre os sistemas físicos e virtuais, resultando em um modelo de produção eficiente e adaptável.

## 2.2 Diferença entre Simulação e Virtualização

Com o aumento da demanda por processos industriais padronizados e automatizados, nos quais ocorre um grande fluxo de informações, torna-se essencial empregar mecanismos capazes de identificar e, quando possível, minimizar riscos e falhas potenciais no processo de fabricação industrial. Nesse contexto, surge a ideia de implementar a simulação de processos ou produtos físicos em um ambiente virtual e controlado. Essa abordagem permite identificar e abordar previamente essas falhas antes de implementar o sistema fisicamente (CARRASQUINHO, 2015). Então, nota-se a necessidade de investir cada vez mais em simuladores.

Um simulador pode ser definido como um dispositivo capaz de representar um sistema ou um processo do mundo real em um ambiente virtual, cujo objetivo é analisar o comportamento do sistema e identificar possíveis falhas que podem ser solucionadas antes da implementação do sistema físico (CARRASQUINHO, 2015). Isso serve de parâmetro para reduzir os custos dos processos industriais e garantir uma maior segurança para os operadores manuais.

No cenário da Indústria 4.0, outro termo que vem sendo bastante utilizado nas aplicações industriais é a "virtualização". Contudo, é importante destacar que, em alguns casos, a virtualização é frequentemente confundida com a simulação.

A Virtualização refere-se à capacidade de criar réplicas virtuais de ambientes ou produtos industriais, com o propósito de analisar e monitorar os processos industriais (TAKAYAMA; PANHAM, 2022). De modo geral, a virtualização é o mecanismo capaz de representar virtualmente os elementos do mundo real, buscando retratar todas as características do modelo físico.

Assim, no contexto da Indústria 4.0, é evidente a crescente importância de dispositi-

vos capazes de simular ou virtualizar processos produtivos. Isso requer uma maior automação nas indústrias e um investimento contínuo em tecnologias e mecanismos inteligentes para atender às crescentes demandas. Vale destacar, também, que essas ferramentas podem ser utilizadas para fins didáticos.

### **2.3 Gêmeos Digitais**

Uma das capacidades notáveis dos gêmeos digitais é a possibilidade de simular diferentes cenários e condições em um ambiente virtual completamente controlável. Isso permite a experimentação e otimização de sistemas sem a necessidade de implementá-los fisicamente (OCHOA *et al.*, 2021).

O termo "Simulação Virtual" pode ser igualmente empregado para descrever os gêmeos digitais. De maneira geral, essa abordagem viabiliza a criação de um "gêmeo" virtual que representa um processo ou produto do mundo real. É importante destacar que essa técnica se fundamenta em procedimentos computacionais e, em conjunto com a crescente robustez de softwares de última geração, resulta na redução dos custos operacionais nas fábricas (TAKAYAMA; PANHAM, 2022).

A implementação de gêmeos digitais promove a melhoria contínua dos processos fabris e também pode ser aplicado para fins didáticos por meio da pesquisa e desenvolvimento. Dentre as áreas de conhecimento, a robótica, por exemplo, pode utilizar os gêmeos digitais para representar os sistemas robóticos.

### **2.4 Robótica**

Os sistemas robóticos podem ser utilizados em ambientes industriais para complementar as tarefas realizadas por operadores humanos. Estes dispositivos desempenham um papel crucial em atividades críticas que demandam precisão ou envolvem riscos significativos (NIKU, 2020).

Os robôs podem ser definidos como dispositivos eletrônicos controlados por microcontroladores e projetados para desempenhar funções mais complexas, as quais demandam um certo grau de precisão e eficiência em comparação com os humanos. Sua utilização oferece inúmeras vantagens, como a capacidade de operar em ambientes insalubres e perigosos, além de proporcionar uma maior precisão nas tarefas realizadas (NIKU, 2020).

Dentre os tipos de robôs, os manipuladores robóticos são dispositivos projetados para desempenhar ações específicas, as quais estão relacionados principalmente à manipulação de objetos no espaço físico. Esses dispositivos são projetados de tal forma que possam emular um braço humano, com o intuito de pegar, mover e posicionar objetos no espaço (SIMPLICIO *et al.*, 2016). A Figura 1 apresenta um exemplo de manipulador robótico amplamente empregado em aplicações industriais, conhecido como "*The Adept 6*".

Figura 1 – Exemplo de um Manipulador Robótico



Fonte: (CRAIG, 2014).

Esses dispositivos são bastante utilizados em processos industriais, devido a sua capacidade de precisão e de automatização das tarefas. Grande parte das ferramentas de simulação robótica aplicada aos manipuladores, buscam analisar os movimentos do manipulador robótico em diversos ambientes e cenários (ŽLAJPAH, 2008).

Os principais componentes dos manipuladores robóticos são: manipulador mecânico ou braço robótico, unidade de controle, atuadores, sensores e unidade de potência (ROMANO; DUTRA, 2002). Cada um destes componentes será detalhado a seguir.

**Manipulador Mecânico:** refere-se a parte mecânica e estrutural do robô. Esse mecanismo consiste na combinação dos seguintes elementos:

- Base: localizada na parte inferior, tendo a função de sustentar o dispositivo e permanece geralmente fixa em sua posição;
- Juntas: representam as articulações do dispositivo. A quantidade de juntas está diretamente relacionada aos graus de liberdade do sistema, ou seja, quanto mais juntas, maior será o grau de liberdade<sup>1</sup> do manipulador;
- Elos: elementos que conectam as juntas do manipulador, com a finalidade de garantir a movimentação e flexibilidade do sistema. Frequentemente, esses componentes são referidos como os "braços" do manipulador;
- Efetuador: componente final do manipulador, o qual entra em contato direto com o objeto a ser manipulado. Existem vários tipos de efetadores, como garras, pinças, ventosas ou outros dispositivos projetados para agarrar ou interagir com objetos.

**Unidade de Controle:** dispositivo especializado responsável por coordenar os movimentos do manipulador, o qual recebe os comandos, processa as informações e controla algum outro dispositivo. Em geral, utiliza-se algum microcontrolador ou microprocessador para desempenhar essas funções.

**Atuadores:** dispositivos que recebem informações internas do sistema e as transformam em ações no ambiente externo. Em essência, convertem um tipo de energia em outro. Os atuadores mais comuns são aqueles que transformam energia elétrica em energia mecânica.

**Sensores:** enquanto os atuadores atuam no ambiente externo, os sensores desempenham o papel inverso, captando informações do mundo exterior e processando-as internamente. Esses dispositivos permitem inferir a posição do manipulador robótico e tomar decisões com base nas informações sensoriais.

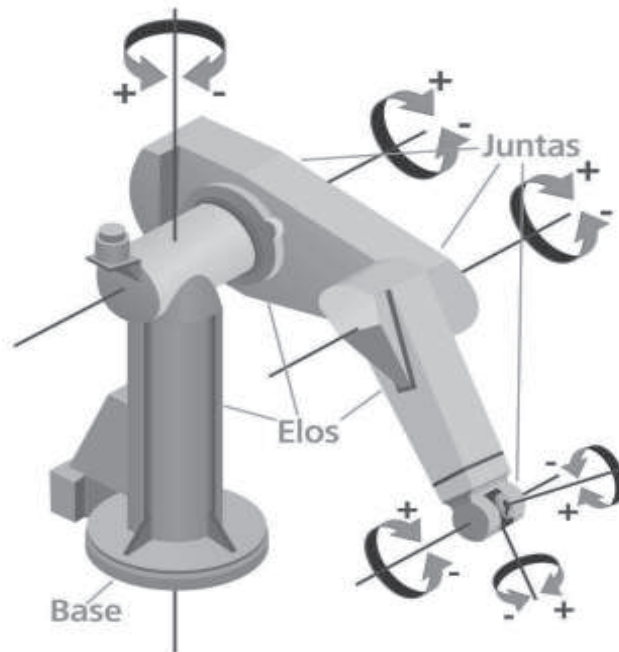
**Unidade de potência:** responsável por fornecer a energia e a potência necessárias para acionar os atuadores ou sensores e manter o sistema em funcionamento.

Para exemplificar os componentes de um manipulador robótico, a Figura 2 ilustra a base, os elos e as juntas de um manipulador. Já a Figura 3 mostra os principais tipos de juntas empregadas no setor industrial.

---

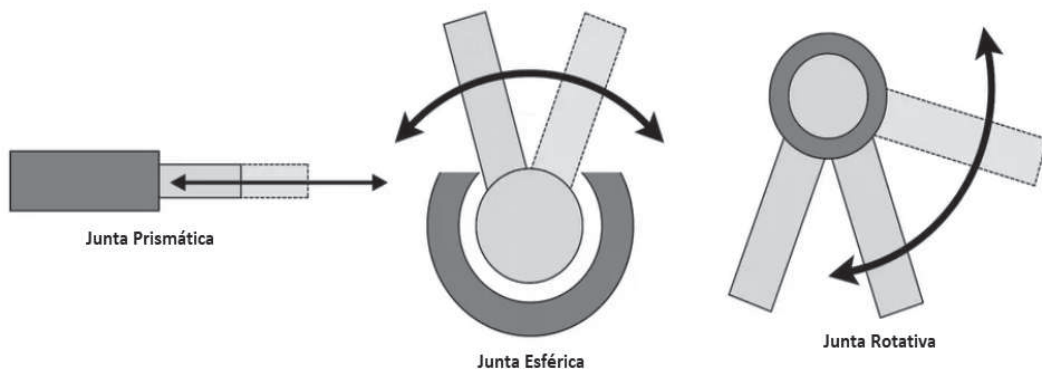
<sup>1</sup> Grau de liberdade de um manipulador representa o número de movimentos independentes que suas articulações permitem, tanto rotações quanto translações, refletindo a flexibilidade e a capacidade de movimento do sistema

Figura 2 – Estrutura de um Manipulador Robótico



Fonte: (BAYER *et al.*, 2011).

Figura 3 – Principais juntas utilizadas na indústrias



Fonte: adaptado de Rao (2023).

## 2.5 Cinemática Robótica

A cinemática robótica é uma área essencial para o controle, geração de movimentos e programação de robôs industriais. Ela lida com o estudo dos movimentos dos componentes do robô, concentrando-se na análise das posições e trajetórias, sem considerar as forças envolvidas (CRAIG, 2014). Já quando se trata de compreender e controlar as forças, torques e acelerações que afetam o movimento do braço robótico, estamos adentrando no campo da dinâmica robótica (NIKU, 2020), a qual não será abordada neste trabalho.

No que se refere a análise da posição do efetuador no espaço, a cinemática robótica permite o equacionamento matemático para modelar os manipuladores e fornece os parâmetros



para situar estes no espaço cartesiano (CONSTANTIN *et al.*, 2015). Os equacionamentos matemáticos surgem de uma descrição espacial dos componentes do manipulador.

### 2.5.1 Descrição espacial

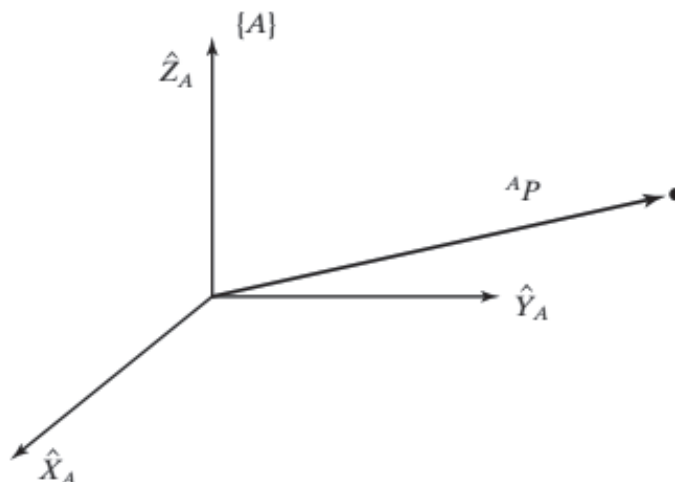
O estudo da robótica, geralmente, envolve a necessidade de localizar objetos, sejam eles peças a serem manipuladas ou ferramentas, em um espaço tridimensional. Diante disso, a descrição espacial permite detalhar diversas características de um objeto específico, como por exemplo, sua localização no espaço e sua orientação. Para isso, são criados sistemas de referência chamados *frames* ou quadros, associados a cada componente rígido do manipulador (CRAIG, 2014).

Para entender esses sistemas de referência, é fundamental determinar a localização e orientação de um objeto em relação ao *frame*. Assim que o sistema de coordenadas é estabelecido, torna-se possível determinar a posição de um objeto no espaço usando um vetor posição tridimensional, representado pela Equação abaixo:

$${}^A\mathbf{P} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (2.1)$$

A notação  ${}^A\mathbf{P}$  representa a posição do objeto em relação ao quadro  $\{A\}$ , conforme mostrado na Figura 4.

Figura 4 – Vetor posição



Fonte: (CRAIG, 2014).

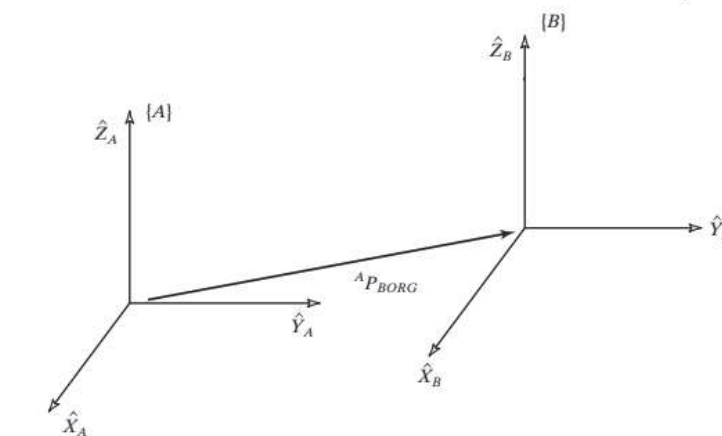
Para descrever a orientação de um objeto no espaço, é necessário estabelecer um quadro de referência para esse objeto e referenciá-lo em relação a outro quadro. Quando um objeto está localizado no quadro {B}, descrever o quadro {B} em relação ao quadro {A} é suficiente para obter a orientação do objeto. A representação da orientação é realizada por meio de uma matriz de rotação, conforme ilustrado na Equação 2.2.

$${}^A_B\mathbf{R} = \begin{bmatrix} {}^A X_B & {}^A Y_B & {}^A Z_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2.2)$$

Na Equação 2.2, observa-se a matriz de rotação que descreve a transformação das coordenadas do sistema de referência {B} em relação ao sistema de referência {A}. Essa matriz é composta por elementos  $r_{ij}$ , onde  $i$  e  $j$  representam as coordenadas dos eixos dos sistemas {A} e {B}, respectivamente. As colunas dessa matriz indicam como os eixos do quadro {B} estão orientados em relação ao quadro {A}.

A Figura 5 ilustra um elemento localizado no sistema de referência {B} em relação ao sistema de referência {A}.

Figura 5 – Representação do vetor que relaciona a origem de {B} com {A}



Fonte: adaptado de Craig (2014).

Portanto, os *frames* são sistemas de referência compostos por um vetor de posição e uma orientação, permitindo a localização precisa de um elemento no espaço tridimensional. Quando um objeto se move em relação a um referencial fixo, essa trajetória pode ser representada como uma nova configuração do referencial, permitindo uma observação do objeto sob diferentes perspectivas. Em geral, existem três tipos principais de movimentos possíveis: rotação pura,

onde o dispositivo gira em torno de um eixo fixo; translação pura, na qual o robô move-se linearmente sem rotação; e, por fim, a combinação de rotação e translação, caracterizada por um movimento que envolve tanto rotação quanto translação simultaneamente (NIKU, 2020).

Os cálculos necessários para implementar esses movimentos podem se tornar extensos e complexos. Portanto, recorre-se ao cálculo matricial para representar esses movimentos, usando matrizes de transformação homogênea. Essa ferramenta possibilita a implementação de movimentos de rotação e translação por meio de cálculos matriciais, relacionando a posição e orientação do objeto no espaço (NIKU, 2020). A matriz de transformação é representada pela Equação 2.3.

$${}^A_B\mathbf{T} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^A_B\mathbf{R} & {}^A_B\mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

A matriz  ${}^A_B\mathbf{R}$ , representa a matriz de rotação, e possui dimensão 3x3, representando as rotações do objeto em análise. O vetor  ${}^A_B\mathbf{P}$ , de dimensão 3x1, descreve as translações nas direções x, y e z, do quadro {B} em relação ao quadro {A}. Essas equações estabelecem uma relação entre dois sistemas de referência, permitindo a localização precisa de qualquer objeto no espaço. Por meio da Equação 2.4, é possível determinar a posição de um objeto no espaço ao calcular a matriz de transformação homogênea.

$$\begin{bmatrix} {}^A\mathbf{P} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_B\mathbf{R} & {}^A_B\mathbf{P} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B\mathbf{P} \\ 1 \end{bmatrix} \quad (2.4)$$

### 2.5.2 Cinemática Direta

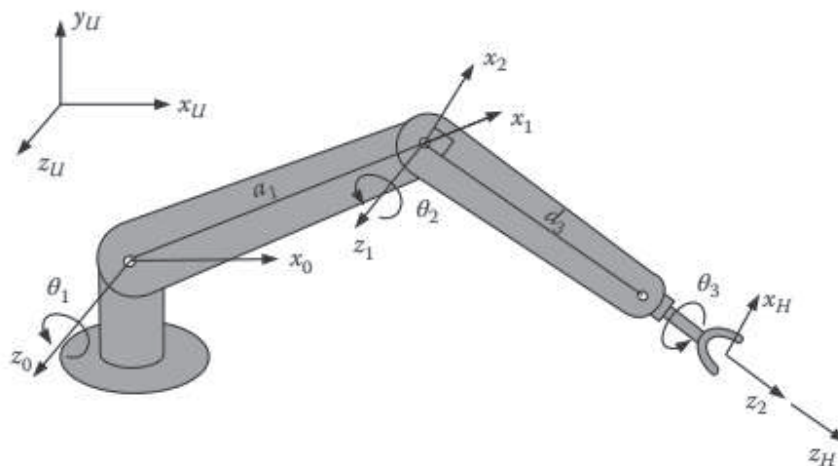
A Cinemática Direta permite encontrar a posição do efetuador de um manipulador robótico a partir das configurações espaciais das juntas que conectam cada elo. Em essência, a Cinemática Direta mapeia as posições no espaço das juntas para o espaço cartesiano, permitindo determinar a posição final do efetuador. Isso é realizado por meio do uso de matrizes de transformação homogênea e pela fixação dos sistemas de referência para cada junta (PENHA *et al.*, 2016).

A fim de obter o equacionamento matemático necessário para modelar um manipulador robótico, a técnica de *Denavit-Hartenberg* (DH) é amplamente utilizada. Essa abordagem emprega uma convenção de coordenadas para atribuir sistemas de referência aos elos e juntas de um manipulador robótico, sendo aplicável tanto na cinemática direta quanto na inversa. Com base nessa convenção, quatro parâmetros ( $a$ ,  $d$ ,  $\alpha$ ,  $\theta$ ) são definidos para descrever a geometria e a configuração do robô em cada junta (NIKU, 2020). A técnica DH segue as seguintes etapas:

1. Definição dos sistemas de coordenadas para cada junta do manipulador, seguindo a regra da mão direita, em que o sentido horário/esquerda/baixo representa angulações negativas e o sentido anti-horário/direita/cima representa angulações positivas;
2. Definir os parâmetros da matriz DH, onde "d" denota deslocamento ao longo do eixo Z, "a" representa deslocamento ao longo do eixo X, " $\theta$ " refere-se ao ângulo de articulação ao redor do eixo Z e " $\alpha$ " indica o ângulo de torção ao redor do eixo X. Esses parâmetros servem como referência;
3. Construção da matriz de transformação homogênea para cada junta, incorporando os parâmetros da matriz DH;
4. Finalmente, as matrizes homogêneas de cada junta são multiplicadas, resultando em uma única matriz de transformação que representa a cinemática direta do braço robótico.

Essa técnica permite modelar qualquer tipo de juntas e de elos dos manipuladores robóticos. A Figura 6 mostra um exemplo de modelagem de um manipulador com 3 graus de liberdade. Pode-se observar que para cada junta existe um *frame* de referência.

Figura 6 – Frames de referência para um manipulador com 3 juntas



Fonte: (NIKU, 2020).

### 2.5.3 Cinemática Inversa

A Cinemática Inversa é o processo de determinar as posições das articulações (juntas) de um manipulador com base na posição e orientação de seu efetuador final, ou seja, a partir do espaço cartesiano do efetuador encontram-se as posições das juntas (PENHA *et al.*, 2016). A Cinemática Inversa é amplamente utilizada na prática da robótica devido à sua capacidade de proporcionar um controle preciso do posicionamento e manipulação de objetos, tornando-se especialmente valiosa na geração de trajetórias para o robô.

Embora a Cinemática Inversa seja bastante utilizada, sua aplicação é mais complexa, por envolver equações não lineares. Por outro lado, a Cinemática Direta é bastante utilizada no ensino da Robótica por meio de uma abordagem mais didática e teórica, pois permite verificar a influência de variáveis de entrada (juntas) na saída do sistema (posição do efetuador). Logo, a aplicação da Cinemática Direta é suficiente para atender às necessidades deste trabalho.

## 2.6 Softwares de Simulação

Os softwares de simulação viabilizam uma representação dos sistemas robóticos e de seus ambientes, permitindo analisar o funcionamento desses sistemas sob diversas condições (ŽLAJPAH, 2008), sendo assim, ferramentas muito importantes para o estudo de manipuladores robóticos. Tais ferramentas permitem que os engenheiros projetem, testem e otimizem sistemas complexos.

A utilização de softwares de modelagem e simulação tem se tornado mais comum na indústria, permitindo simular o ambiente de produção e analisar os parâmetros necessários sem ter que implementar o manipulador robótico, fisicamente. Dentre as diversas linguagens de programação compatíveis com essa ferramenta, destacam-se o C/C++, Perl, Python, Java, LabVIEW, URBI e Matlab. Além disso, existem diferentes softwares utilizados para a modelagem desses manipuladores, como o Matlab, SolidWorks, ROS, Autodesk Inventor e CoppeliaSim (BENOTSMANE *et al.*, 2020), alguns dos quais serão detalhados a seguir.

**Autodesk Inventor:** oferece recursos avançados de modelagem paramétrica, permitindo aos usuários criar e modificar facilmente geometrias complexas. Sua funcionalidade abrangente e capacidade de oferecer um ambiente de design integrado tornam-no uma escolha popular para profissionais e empresas que trabalham com design e desenvolvimento de produtos. No entanto, é importante ressaltar que este software não está disponível gratuitamente, exigindo

um investimento inicial para sua utilização.

**SolidWorks:** software de modelagem 3D com capacidade de realizar simulações e análises de engenharia. Essa tecnologia baseia-se em três conceitos básicos: a peça, a montagem e o desenho. Uma de suas principais vantagens é a capacidade de colaboração, que possibilita que várias pessoas trabalhem no mesmo projeto simultaneamente. Isso permite o compartilhamento de informações e modificações em tempo real, tornando o processo de desenvolvimento mais eficiente e colaborativo.

**Matlab:** plataforma de computação numérica e programação desenvolvida pela MathWorks, o Matlab é amplamente utilizado em várias áreas, incluindo Engenharia, Ciência, Finanças e Pesquisa. Sua sintaxe simples e intuitiva facilita a resolução de problemas complexos através da escrita e execução de código Matlab. No contexto da robótica, esse software permite a simulação de modelos robóticos por meio do ambiente de simulação Simulink. Além disso, ele é conhecido por sua capacidade de integração com outros softwares, como o Autodesk Inventor e o SolidWorks.

**ROS:** ferramenta de código aberto amplamente utilizada na área de robótica. O ROS oferece uma extensa coleção de bibliotecas, ferramentas e drivers que facilitam o desenvolvimento e a simulação de robôs. O simulador de robôs Gazebo é frequentemente utilizado em conjunto com o ROS para a realização de simulações robóticas.

Um tópico importante, no contexto da virtualização, é a transição da simulação virtual para o mundo real, a qual requer uma comunicação entre os modelos virtuais e reais. Com isso, o estudo dos protocolos de comunicação se torna necessário.

## 2.7 Protocolos de Comunicação

A comunicação entre dispositivos desempenha um papel fundamental em uma sociedade cada vez mais conectada, onde a indústria 4.0 está se consolidando. Os protocolos de comunicação são os alicerces dessa interação, fornecendo diretrizes e regras que permitem que diferentes dispositivos compartilhem informações entre si.

Existem dois tipos principais de interfaces: as seriais e as de rede. As interfaces seriais, desde interfaces RS-232 até as interfaces USB, são frequentemente usadas para conectar dois dispositivos diretamente em uma comunicação ponto a ponto. Por outro lado, as interfaces de rede têm um escopo mais amplo, permitindo a comunicação entre vários dispositivos em uma rede. Esta conexão pode ser por fio ou sem fio (OLIVEIRA, 2017).

A comunicação serial é um dos métodos mais tradicionais, onde os dados são transmitidos sequencialmente, *bit* por *bit*, por meio de uma porta serial, conectada, geralmente, à porta USB do dispositivo. A taxa de transmissão (*baud rate*) é medida em *bits* por segundo (bps), indicando a quantidade de *bits* transmitidos ou recebidos por segundo. Outra importante tecnologia, denominada *Bluetooth*, permite comunicação sem fio em curtas distâncias, geralmente na faixa de 2.4 GHz a 2.48 GHz, com canais de 1 MHz (ANDRADE; BASSANI, 2020).

A tecnologia *Wi-Fi*, permite a conexão entre dispositivos em redes locais e sem fio. Essa tecnologia, juntamente com *Bluetooth*, está se tornando cada vez mais comuns e muitos dispositivos possuem essas interfaces já embutidas (OLIVEIRA, 2017).

A escolha entre protocolos de comunicação com fio ou sem fio é uma decisão fundamental durante o escopo do projeto. Protocolos com fio, como a comunicação serial, são ideais para comunicação de curta distância e ambientes onde a precisão e o tempo de resposta são essenciais. Já os protocolos de comunicação sem fio enfrentam alguns problemas, como interferência e ruídos. Além disso, possuem uma latência variável, o que pode impactar no tempo de resposta do sistema. Portanto, esses fatores devem ser levados em consideração durante a escolha do protocolo de comunicação.

### 3 MATERIAIS E MÉTODOS

O presente capítulo descreve os materiais e métodos utilizados para o desenvolvimento deste trabalho. As etapas abordadas seguem o fluxograma abaixo.

Figura 7 – Fluxograma do trabalho

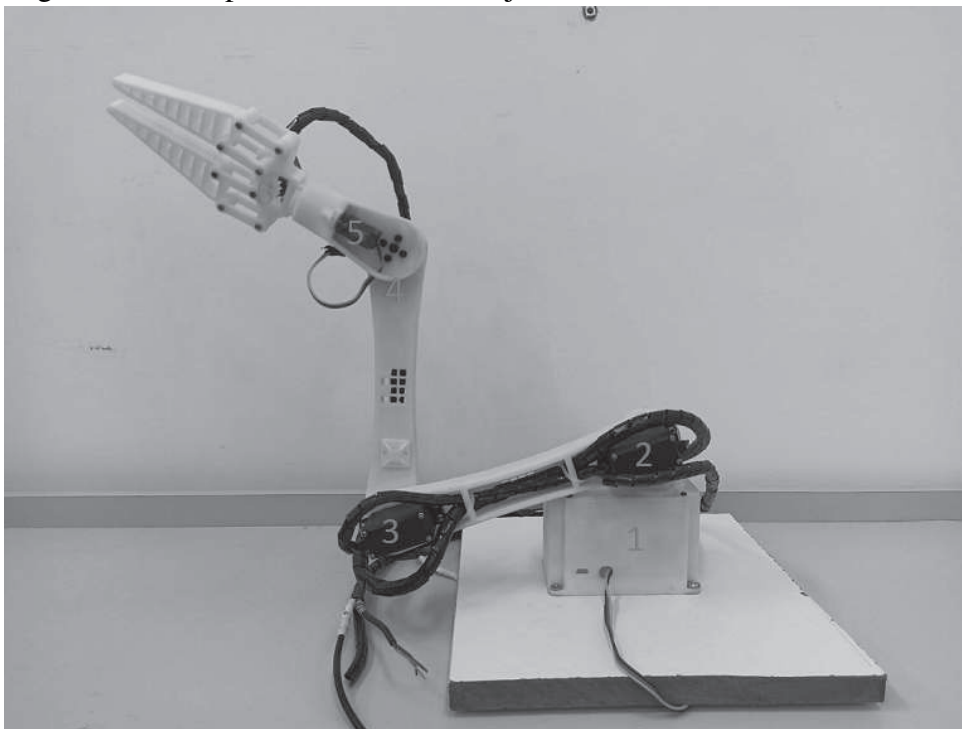


Fonte: elaborado pelo autor.

#### 3.1 Manipulador Robótico

A Figura 8 representa o braço robótico utilizado para estudos e testes, o qual foi desenvolvido por um projeto de Robótica do IFCE. O projeto proposto busca a disseminação e o desenvolvimento do estudo da robótica. Diante disso, o manipulador foi fornecido à UFC para realização de estudos e testes pelos alunos. O modelo das peças mecânicas deste manipulador pode ser obtido gratuitamente <sup>2</sup>.

Figura 8 – Manipulador robótico de 5 juntas rotativas



Fonte: elaborado pelo autor.

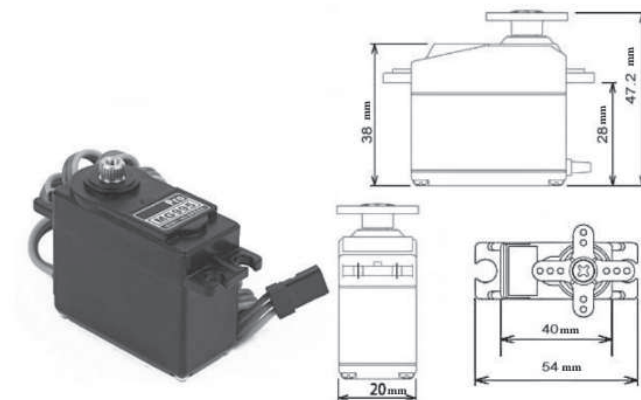
<sup>2</sup> <https://robotica.ifce.edu.br>



Este manipulador é composto por 5 juntas rotativas e o efetuator final, numerados de 1 a 6, como representado na Figura 8. Mediante compreensão da relação entre as juntas e os graus de liberdade, determina-se que o modelo em questão possui um total de 5 graus de liberdade. Esses graus de liberdade definem o plano de operação do manipulador (CRAIG, 2014).

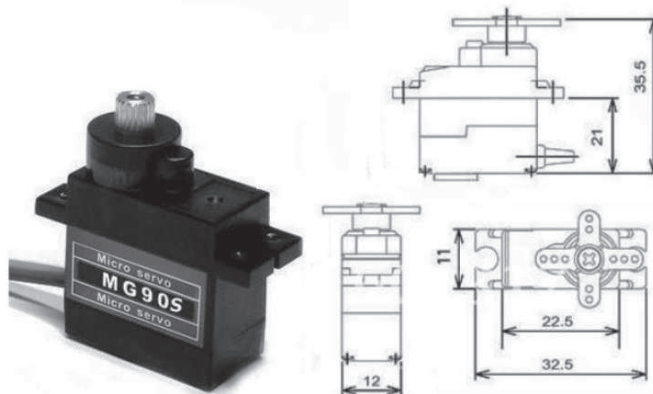
As juntas 1, 2, 3 e 4 possuem um servo motor MG995, conforme mostra a Figura 9, com tensão de operação entre 4.8 V e 7.2 V, e um torque de parada entre 8.5 kgf.cm e 10 kgf.cm. O ângulo de rotação desse dispositivo é de 120 graus (+- 60 graus em torno do centro). Em contrapartida, a junta 5 e o efetuator final (6) utilizam servos motores MG90S, como ilustrado na Figura 10, com tensão de operação entre 4.8 V e 6.0 V, e um torque de parada entre 1.8 kgf.cm e 2.2 kgf.cm. O dispositivo pode girar aproximadamente 180 graus (90 graus em cada direção). Os anexos A e B mostram os *datasheets* dos servos motores MG995 e MG90S, respectivamente.

Figura 9 – Servo MG995



Fonte: *datasheet* do Servo MG995 - Anexo A.

Figura 10 – Micro Servo MG90S



Fonte: *datasheet* do Micro Servo MG90S - Anexo B.

No modelo virtual, não foi implementada a capacidade de movimentação do efetuador final. Mesmo o manipulador possuindo um servo motor acoplado, esta escolha de projeto foi feita devido ao escopo do presente trabalho, que se concentra principalmente na cinemática do modelo apresentado. A determinação da posição final da garra é crucial para fins didáticos e industriais, possuindo diversas aplicações que envolvem a manipulação de objetos, desde a montagem até o posicionamento. Logo, o movimento de abrir e fechar da garra não impacta no objetivo deste trabalho.

### 3.2 Cinemática do Manipulador Robótico

Para implementar a Cinemática do manipulador foram utilizadas técnicas algébricas que envolvem transformações matriciais, com base na técnica de DH. Nesse processo, levando em consideração as restrições físicas impostas pelo modelo real, cada junta foi modelada matematicamente com base em sistemas de referência apropriados, um para cada junta rotativa. Diante disso, a partir da modelagem física do manipulador robótico didático, foram extraídos os parâmetros para cada junta rotativa, utilizando a técnica DH (SOARES, 2023). Tais parâmetros podem ser vistos na Tabela 1.

Tabela 1 – Parâmetros DH de cada junta

Junta (i)	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	0	+90°	L1	$\theta_1$
2	L2	+180°	0	$\theta_2$
3	L3	-180°	0	$\theta_3$
4	0	+90°	0	$\theta_4 + 90$
5	0	0	L4+L5	$\theta_5$

Fonte: adaptado de Soares (2023).

Cada parâmetro da Tabela 1 refere-se aos elementos da matriz DH, onde "i" representa cada junta do braço robótico. Os valores de cada elo do manipulador real são referenciados pelos elementos L1, L2, L3, L4 e L5, os quais foram obtidos a partir de medições feitas no braço físico. A Tabela 2 mostra os valores de cada elo, medidos em centímetros (cm).

A matriz DH permite a implementação das equações da cinemática, tanto a direta, quanto a inversa. Essa matriz permite calcular a posição do efetuador final no plano tridimensional com base nas posições e orientações de cada junta. Levando em consideração o escopo deste trabalho, não será implementada a cinemática inversa do manipulador.

Tabela 2 – Tamanhos dos elos

Elo	Tamanho (cm)
1	10,0
2	18,0
3	18,0
4	6,0
5	12,0

Fonte: elaborado pelo autor.

### 3.3 Modelo Virtual

A etapa de virtualização do manipulador real envolve a criação do seu gêmeo digital, exigindo a modelagem do sistema físico em um ambiente computacional para reproduzir as características e restrições do modelo real. Nesse contexto, o software Autodesk Inventor é utilizado como ferramenta de modelagem 3D. Com ele, é possível construir o desenho do braço robótico, por meio da montagem de todas as peças do modelo virtual. Vale ressaltar que a montagem do modelo 3D leva em consideração todas as restrições e características físicas do modelo real.

Os modelos 3D de cada uma das peças do manipulador foram disponibilizados pelo professor Izamaro de Araújo, membro do projeto de robótica do IFCE, o qual atua no campus do IFCE em Jaguaribe.

Além da modelagem, a simulação e os testes computacionais são etapas da Virtualização. Para essa finalidade, utiliza-se o software Matlab, especialmente o ambiente Simulink, a qual se integra com o Matlab. A capacidade de conexão e interatividade entre o Autodesk Inventor e o Matlab/Simulink é fundamental para o desenvolvimento do gêmeo digital.

#### 3.3.1 *Inventor*

O Inventor, desenvolvido pela Autodesk, é um dos softwares de modelagem amplamente utilizados na área de Engenharia e Modelagem devido à sua capacidade de processamento robusta e ampla gama de funcionalidades oferecidas. Além disso, o software possui compatibilidade com diversos sistemas operacionais, incluindo Windows, Linux e MacOS (GOUVEIA; ANDRADE, 2023).

A escolha do inventor como ferramenta de modelagem para este trabalho se baseou em vários fatores, como sua capacidade avançada de criação de modelos precisos e detalhados, e o consumo reduzido de recursos de processamento. Sua interface intuitiva e ferramentas de

modelagem para corpos rígidos o tornam ideal para a modelagem de manipuladores robóticos.

O software em questão também oferece compatibilidade direta com o Matlab, graças ao *plug-in Simscape Multibody Link*. Este *plug-in* possibilita a exportação de modelos criados no Inventor para o *Simscape Multibody*, um ambiente de simulação do Simulink utilizado para modelagem e simulação de sistemas físicos e dinâmicos. A instalação e habilitação deste *plug-in* permite a integração entre os softwares, desde que as versões sejam compatíveis. Importante destacar que a partir da versão 2022 do Inventor, essa funcionalidade não está mais disponível, sendo a versão do software utilizada neste trabalho a Professional 2021.

A integração entre o Autodesk Inventor e o Simulink ocorre de forma direta. Após a criação do modelo no Inventor, a exportação para o Simulink é realizada por meio de uma janela dedicada, ativada durante o processo de exportação. O *plug-in Simscape Multibody Link* transfere informações sobre os componentes mecânicos do modelo, incluindo geometria, restrições, articulações e massas. Essas informações são transmitidas em um formato de arquivo XML, que serve como base para a construção de um modelo de simulação no Simulink. Esse modelo resultante representa o sistema mecânico projetado no Inventor e permite análises e simulações de seu comportamento dinâmico.

Esta integração é uma ferramenta valiosa para a virtualização e criação do gêmeo digital do manipulador robótico. Esta combina recursos avançados de modelagem 3D com a capacidade de realizar simulações e análises dinâmicas complexas, sem a necessidade de trabalhar com o modelo real, tornando-a essencial para a criação e validação de projetos de sistemas mecânicos.

### 3.3.2 *Matlab*

O Matlab, software desenvolvido pela MathWorks, oferece um ambiente de programação e computação numérica utilizado em diversas áreas, desde Engenharia, Matemática até Ciência da Computação. A grande vantagem de utilizar esse software é sua versatilidade e capacidade de trabalhar com diversas aplicações, sem contar a ampla gama de bibliotecas disponíveis.

Dentro do Matlab, várias *Toolboxes* estão disponíveis para simplificar o desenvolvimento e a análise de diversas aplicações e sistemas. A seguir são listadas algumas das *Toolboxes* utilizadas.

- ***Robotics Toolbox***: oferece funcionalidades específicas para modelagem e controle de

robôs e manipuladores. Esta biblioteca é bastante útil para verificar a cinemática direta do manipulador robótico.

- **Simulink**: ambiente de diagrama de blocos utilizado para simulação e design em tempo real.
- **Simscape**: permite criar modelos de sistemas físicos no ambiente Simulink. Com essa ferramenta, é possível construir modelos de componentes físicos e inserir interações com os modelos. Tudo é feito por meio de blocos que representam as partes físicas do modelo.
- **Simscape Multibody**: fornece um ambiente de simulação multicorpo para sistemas mecânicos 3D, o qual pode ser utilizado para modelagem de manipuladores robóticos.

### 3.3.2.1 Integração entre softwares

Ao integrar o modelo do manipulador robótico, gerado no Autodesk Inventor, ao Simulink, permite-se a execução de simulações em tempo real, além de permitir o controle e análise do comportamento dinâmico do manipulador em resposta a diferentes entradas e condições. Para importar o modelo do Inventor é necessário utilizar a função *smimport* do Matlab, a qual irá criar um modelo virtual do manipulador robótico, em formato SLX. A partir disso, diversas modificações e simulações podem ser feitas no Simulink.

A interação entre os diversos componentes, no Simulink, podem ser feitas por meio de uma interface gráfica. A vantagem de utilizar essa ferramenta é a usabilidade e acessibilidade para os usuários.

### 3.3.2.2 Interface Gráfica

No contexto da virtualização, a criação de uma Interface Gráfica de Utilizador (do inglês, *Graphical User Interface* (GUI)) permite a interação intuitiva entre os usuários e os sistemas computacionais. O Matlab por sua vez, oferece várias abordagens para o desenvolvimento de GUIs. Abaixo estão algumas dessas possibilidades:

- **Graphical User Interface Development Environment (GUIDE)**: essa ferramenta permite que os desenvolvedores criem interfaces de forma interativa, arrastando e soltando alguns elementos como botões, caixas de texto, gráficos e controles deslizantes diretamente na interface. O GUIDE gera automaticamente o código Matlab correspondente a GUI criada, facilitando a personalização e a extensão da interface;
- **App Designer**: ferramenta mais recente em comparação com o GUIDE. Apresenta uma

abordagem orientada a objetos que permite aos desenvolvedores criar aplicativos interativos no MATLAB de maneira visual e estruturada. Esses aplicativos consistem em pequenas interfaces que se integram com a interface principal. O *App Designer* oferece uma ampla gama de componentes predefinidos, incluindo botões, tabelas, gráficos e áreas de texto. A interface é criada por meio de funções que respondem a eventos gerados pelos componentes da GUI;

- **Programação direta:** permite o controle total do design e funcionalidade da interface, bem como a capacidade de personalização avançada. Utilizando esta estratégia, é possível criar GUIs do zero ou modificar as GUIs geradas pelo GUIDE, ou pelo *App Designer*. A grande desvantagem dessa ferramenta é a codificação extensa que exige um domínio maior da programação. No entanto, essa abordagem é bastante útil para projetos mais complexos que exigem uma personalização extensa.

No escopo do presente trabalho, optou-se pela utilização do GUIDE como a ferramenta primária para o desenvolvimento da interface gráfica. A escolha dessa ferramenta se deu pela acessibilidade, facilidade de uso e por esta se destacar pela eficiência e praticidade na criação de interfaces simples.

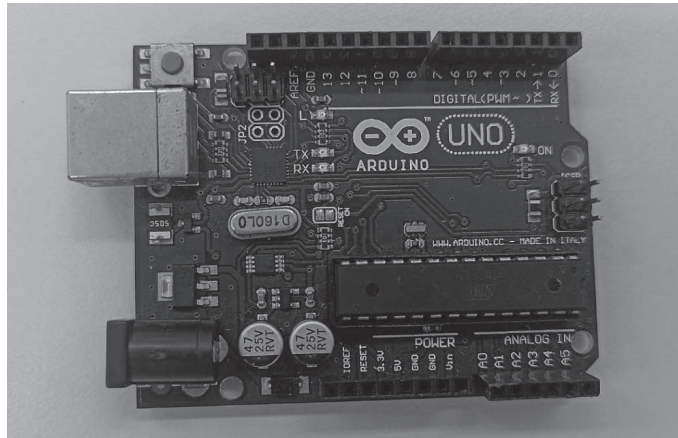
### 3.4 Comunicação Serial

A comunicação entre o modelo real e o modelo virtual, é uma etapa importante da virtualização, pois permite validar e comprovar no modelo real os testes e simulações feitos no ambiente computacional. Essa comunicação pode ser feita utilizando diversos protocolos de comunicação, entre eles, a comunicação via rede, a comunicação bluetooth, a comunicação por USB, dentre outros.

No presente trabalho, utiliza-se a Comunicação Serial. Essa escolha se deu, principalmente, devido à sua simplicidade e facilidade de implementação, em comparação com outros protocolos mais complexos, como o Ethernet, usado na Comunicação via Rede. Outros pontos, envolvem o baixo custo e compatibilidade, pois a maioria dos componentes eletrônicos possuem portabilidade necessária para receber e enviar dados via porta serial e também a baixa latência, isso é ideal em sistemas de simulação em tempo real.

A comunicação se deu através da plataforma Arduino, o qual é ilustrado na Figura 11. Este dispositivo é responsável por receber e enviar dados do modelo real para o virtual e vice-versa, por meio da porta serial.

Figura 11 – Arduíno UNO



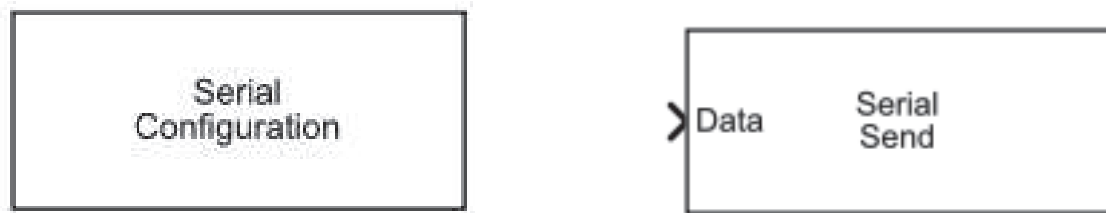
Fonte: elaborado pelo autor.

### 3.4.1 Codificação

A implementação da comunicação serial do modelo virtual, construído no Simulink, é realizada com o auxílio da biblioteca *Instrument Control Toolbox*. Com essa biblioteca é possível gerar dados no Matlab/Simulink para enviar a um instrumento, neste caso se utiliza o Arduino, ou ler dados no Matlab/Simulink a partir de algum instrumento, os quais podem ser utilizados para análise e visualização. Logo, essa ferramenta permite a configuração dos parâmetros necessários para estabelecer a comunicação entre os dispositivos. No ambiente do Simulink, utiliza-se dois blocos: o *Serial Configuration* e o *Serial Send*, mostrados na Figura 12, item (a) e (b), respectivamente.

O bloco *Serial Configuration* é responsável por configurar a porta serial selecionada para a comunicação. Isso inclui a definição da taxa de transmissão de dados, os bits de parada e outros parâmetros necessários para a configuração inicial da comunicação. Já o bloco *Serial Send* é responsável por transmitir os dados do modelo virtual para a porta serial, permitindo a troca de informações entre o modelo no Simulink e o modelo real.

Figura 12 – Blocos utilizados para a comunicação serial



(a) Bloco Serial Configuration

(b) Bloco Serial Send

Fonte: elaborado pelo autor.

O código desenvolvido para lidar com os dados relacionados às posições de cada junta do manipulador foi implementado na plataforma Arduino. Seu objetivo é controlar os cinco servomotores acoplados a cada junta do manipulador robótico. A comunicação é estabelecida de maneira bidirecional com o modelo virtual no Simulink por meio da porta serial.

A conexão entre o Simulink e o Arduino é estabelecida quando o modelo no Simulink entra em execução. Em seguida, os valores de cada junta do manipulador são enviados, em tempo real, para a porta serial. Além disso, o Arduino monitora essa porta serial constantemente, lendo os valores e controlando a posição do servo motor correspondente.

### 3.5 Monitoramento do Modelo Real

Para analisar, simultaneamente, a simulação e a integração entre o modelo real e o modelo virtual construído no Simulink, utiliza-se um dispositivo de captura de vídeo. Esta implementação envolve a utilização de uma câmera ou webcam como uma interface de comunicação em tempo real entre esses dois dispositivos, permitindo monitorar e controlar o braço físico.

A webcam utilizada para testes foi o modelo HD GT da Goldentec, conforme mostra a Figura 13.

Figura 13 – Webcam Goldentec



Fonte: elaborado pelo autor.

A webcam, conectada ao computador que executa o Simulink, captura continuamente a imagem do manipulador físico em tempo real. Para implementar essa funcionalidade é necessário utilizar a biblioteca *Image Acquisition Toolbox*, a qual permite detectar e configurar um hardware de aquisição de imagens, as quais podem ser processadas no Matlab ou Simulink. Essa ferramenta permite utilizar uma webcam como dispositivo de captura de vídeo.



As imagens são adquiridas em tempo real e transmitidas através da interface GUIDE implementada no Matlab, isso permite a sincronização em tempo real entre os modelos virtual e real. A comunicação via webcam viabiliza a condução de experimentos práticos com o manipulador robótico real, pois é possível observar os movimentos do braço físico através da interface. Essa integração proporciona uma plataforma para pesquisa, desenvolvimento e experimentação com manipuladores robóticos.

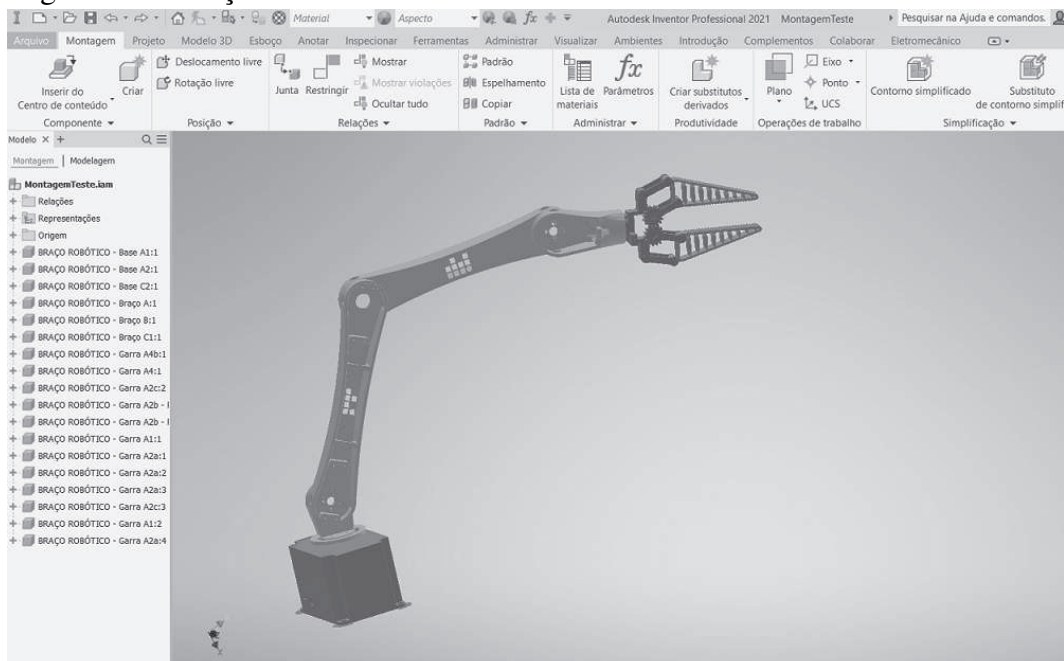
## 4 RESULTADOS E DISCUSSÕES

Neste capítulo, serão apresentados os resultados obtidos ao longo deste trabalho, destacando cada etapa da virtualização do manipulador robótico. Iniciando pela modelagem 3D realizada no software Autodesk Inventor, seguindo pela implementação do modelo virtual no Simulink e, por fim, a comunicação serial entre o modelo virtual e o modelo real. Além disso, serão mostrados os detalhes da interface gráfica criada. Cada parte desse processo será detalhada nas próximas seções.

### 4.1 Modelagem 3D

A criação do gêmeo digital inicia-se com a elaboração do modelo virtual do manipulador robótico. Utilizando o software Autodesk Inventor, elabora-se o desenho do modelo físico. Cada peça individual que compõe o manipulador foi fornecida pelo autor e professor Izamaro de Araújo, membro da equipe de Robótica do IFCE. Posteriormente, essas peças são unidas para formar o modelo final do braço robótico. A Figura 14 exibe a interface do software e o desenho completo do braço robótico.

Figura 14 – Braço robótico construído no Inventor



Fonte: elaborado pelo autor.

Durante a construção do desenho, deve-se levar em consideração as restrições das juntas rotativas e o ângulo máximo permitido para cada junta, visto que tais considerações se

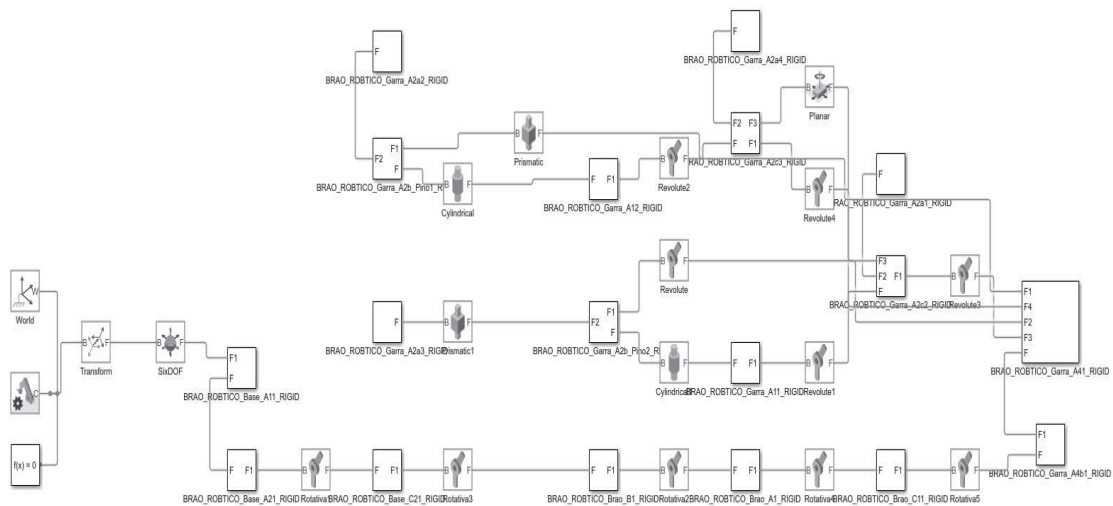
aplicam ao manipulador real. Além disso, o software Inventor oferece a capacidade de realizar testes e análises computacionais, permitindo a verificação das limitações angulares de cada junta rotativa. Essa ferramenta possibilita o refinamento do modelo virtual, a fim de garantir uma correspondência com o modelo real.

Após o ajuste do desenho no Inventor, utiliza-se o *plug-in Simscape Multibody Link* para exportar o modelo para o ambiente Matlab/Simulink. O arquivo resultante está no formato XML e incorpora todas as características definidas no desenho.

## 4.2 Modelo Virtual

Para trabalhar com o modelo virtual no Simulink utiliza-se o comando `'smimport('nomearquivo.xml')` no prompt de comando do Matlab. Esse comando gera um arquivo no formato SLX, que representa o modelo virtual do braço robótico em diagrama de blocos. Esse diagrama pode ser posteriormente modificado e adaptado conforme as necessidades específicas do projeto. A Figura 15 apresenta o modelo do manipulador no Simulink.

Figura 15 – Modelo virtual no Simulink



Fonte: elaborado pelo autor.

Cada bloco mostrado na Figura 15 refere-se a algum parâmetro ou parte física do braço robótico. Alguns blocos que não têm influência na análise computacional são excluídos do modelo, enquanto outros, especialmente os blocos que representam as juntas rotativas, merecem atenção especial, pois são responsáveis por inserir o movimento do braço.

O bloco relacionado à junta rotativa, conforme mostrado na Figura 16, permite a implementação da cinemática direta. Isso é feito inserindo os valores angulares em cada

junta do manipulador. Neste processo, modificam-se dois parâmetros presentes na configuração "Actuation" da junta rotativa. O primeiro deles, denominado "Motion", é modificado para aceitar os valores de entrada, assegurando, assim, a aplicação de movimento em cada junta por meio da inserção de valores angulares. O segundo parâmetro, denominado "Torque", é configurado para ser calculado automaticamente.

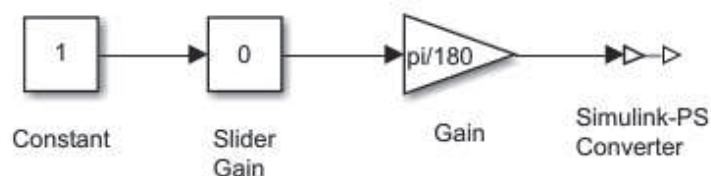
Figura 16 – Junta Rotativa



Fonte: elaborado pelo autor.

Para inserir movimentos nas juntas, são empregados quatro blocos em sequência, conforme ilustrado na Figura 17. O primeiro bloco, denominado *Constant*, é uma constante que gera um sinal de entrada para o bloco *Slider Gain*. Este último é responsável por aplicar um ganho escalonado, que pode ser ajustado durante a simulação, e está relacionado às limitações angulares de cada junta. O bloco *Gain* desempenha a função de converter a entrada, originalmente em radianos, para graus, representando o movimento físico. Por fim, o bloco *Simulink-PS Converter* realiza a conversão de um sinal de entrada do Simulink em um sinal físico, tornando-o compatível com o ambiente de simulação Simscape.

Figura 17 – Blocos de entrada na junta rotativa



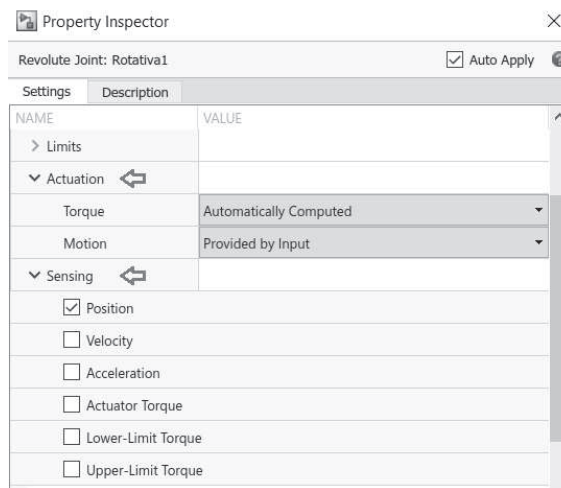
Fonte: elaborado pelo autor.

Essa sequência de blocos permite a inserção de movimentos em cada junta por meio de alterações realizadas no bloco *Slider Gain* do modelo virtual. Os movimentos podem ser

ajustados em tempo real durante a simulação.

Além disso, no bloco relacionado à junta, é possível incluir uma saída que exibe a posição da respectiva junta. Isso é alcançado ajustando o parâmetro "*Sensing*" e ativando a opção para mostrar a posição angular da junta. A Figura 18 mostra os parâmetros "*Actuation*" e "*Sensing*" de uma das juntas rotativas do modelo virtual.

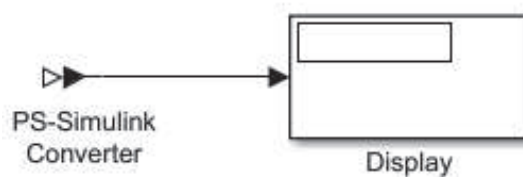
Figura 18 – Propriedades da junta rotativa



Fonte: elaborado pelo autor.

Para exibir as posições de cada junta, são necessários dois blocos em série: o *PS-Simulink-Converter* responsável por converter um sinal físico de entrada em um sinal de saída Simulink, e o bloco *display* que exibe o valor do ângulo da junta. A Figura 19 ilustra esses blocos.

Figura 19 – Blocos de saída na junta rotativa



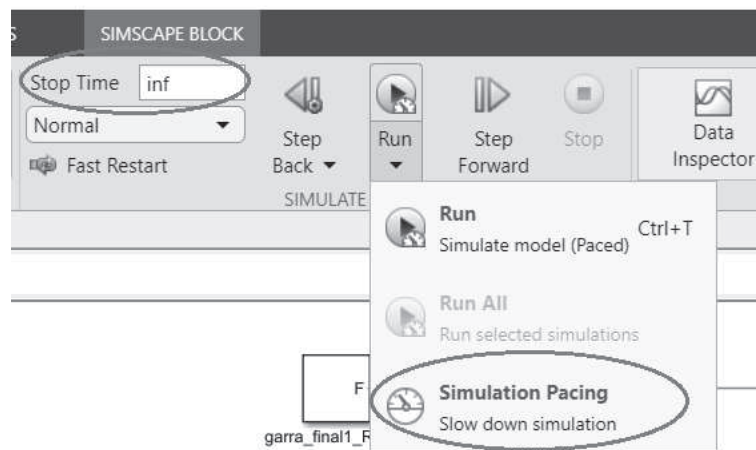
Fonte: elaborado pelo autor.

Por meio dessas adaptações, torna-se viável uma análise inicial da simulação do modelo para avaliar o seu desempenho. Com base nessa avaliação inicial, é possível realizar ajustes no modelo, incluindo a definição das limitações angulares de cada junta, a fim de assegurar uma correspondência com o modelo real.

### 4.3 Simulação

A simulação é realizada diretamente no ambiente do Simulink, para isso, são necessários ajustes em dois parâmetros. O primeiro parâmetro a ser configurado é a ativação do "*Simulation pacing*" que permite a alteração em tempo real dos parâmetros de entrada. Esses parâmetros de entrada correspondem aos ângulos das juntas durante a simulação, possibilitando, assim, a análise dos resultados e os movimentos do braço robótico. O segundo parâmetro a ser ajustado é o "*Stop Time*" o qual deve ser definido como "inf". Isso garante que a simulação seja conduzida por um período de tempo indeterminado e só seja encerrada quando for cancelada de forma deliberada. Esses parâmetros estão relacionados aos ajustes de configuração da simulação e são realizados diretamente na interface do Simulink, conforme mostra a Figura 20.

Figura 20 – Execução do modelo no Simulink

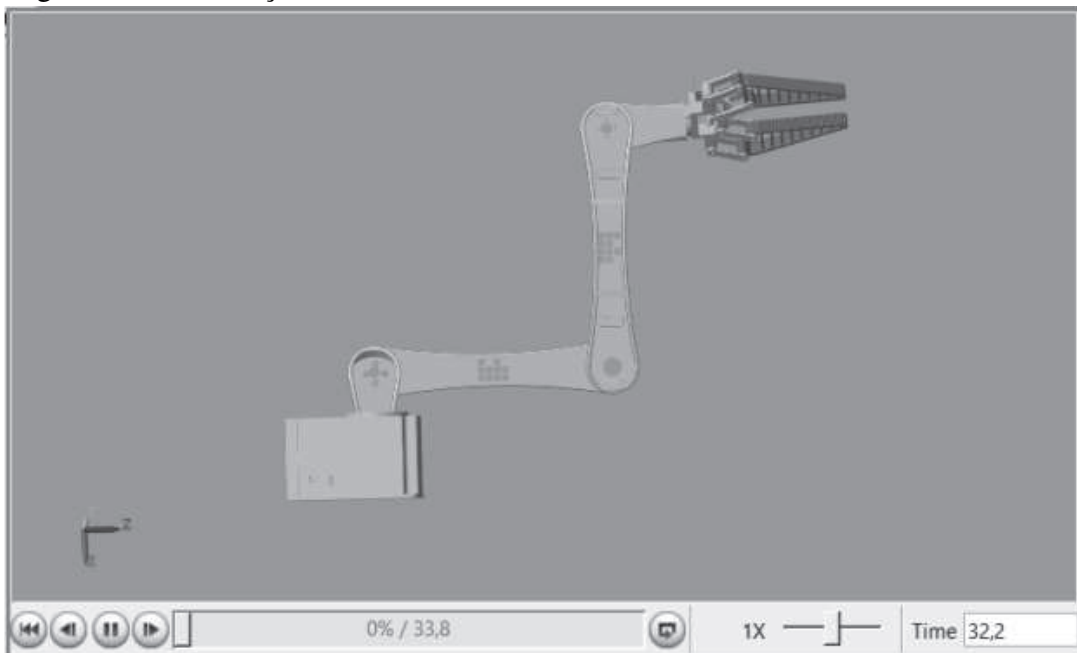


Fonte: elaborado pelo autor.

Após a conclusão dessas configurações, o modelo é executado, possibilitando a análise do funcionamento e dos movimentos do manipulador. O resultado dessa simulação pode ser visualizado na Figura 21, oferecendo uma representação gráfica da simulação em tempo real.

Para adicionar movimentos ao modelo, basta ajustar os valores de cada junta nos blocos *Slider Gain*. Esses valores correspondem aos parâmetros de entrada do modelo. A partir dessas modificações, é viável realizar uma análise em tempo real das ações executadas pelo manipulador virtual. Isso proporciona a capacidade de visualizar e compreender o comportamento do sistema conforme as variáveis de entrada são modificadas.

Figura 21 – Simulação do Modelo Virtual



Fonte: elaborado pelo autor.

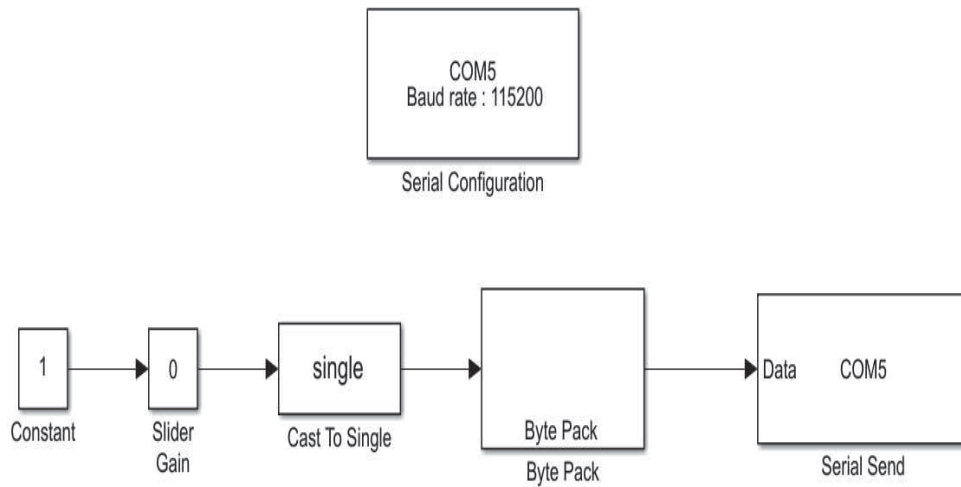
#### 4.4 Integração entre os Modelos Real e Virtual

A integração entre o modelo virtual e o modelo real foi estabelecida por meio da comunicação serial. Para realizar essa comunicação, utiliza-se dois blocos da biblioteca *Instrument Control Toolbox* no Simulink: o *Serial Configuration* e o *Serial Send*. Esses componentes permitem a conexão entre o ambiente virtual e o braço físico.

Além disso, para construir o modelo que envia os dados para o Arduino, utiliza-se outros dois blocos em conjunto com o *Serial Send*. O primeiro bloco, chamado *Cast To Single*, tem a função de garantir a uniformidade dos dados no modelo, assegurando que os dados enviados para o Arduino sejam do tipo ponto flutuante. O segundo bloco, denominado *Byte Pack*, é responsável por empacotar os dados em um único vetor de saída e transmiti-los pela porta serial. As entradas de cada uma das juntas rotativas, representadas pelos blocos *Constant* e *Slider Gain*, do manipulador robótico também estão conectadas a esse conjunto de blocos.

Esses blocos garantem que os dados tratados no modelo sejam uniformes. Além disso, eles levam em consideração o fato dos dados serem transmitidos sequencialmente pela porta serial, ou seja, *byte por byte*. A partir disso, realiza-se testes empregando um único servo motor, representando uma das juntas do manipulador. Nessa análise, foi utilizado um Micro Servo SG90 com ângulo de rotação de 180 graus. A representação do modelo é apresentada na Figura 22.

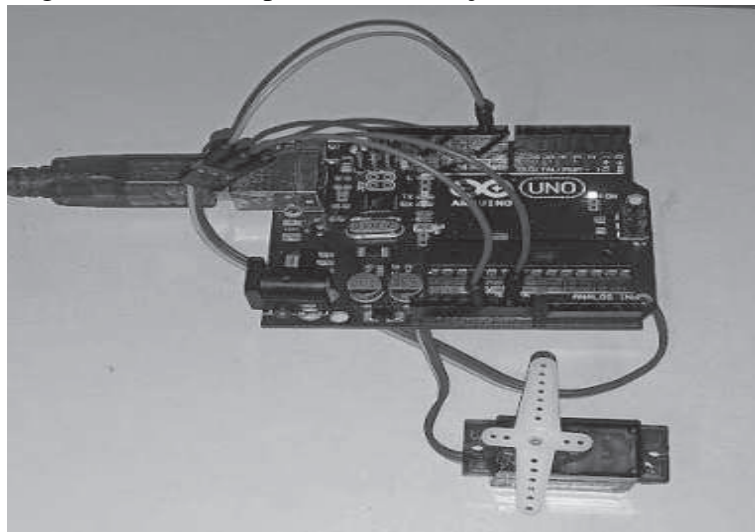
Figura 22 – Modelo de Comunicação Serial com apenas um Servo Motor



Fonte: elaborado pelo autor.

O protótipo real, utilizado nos testes, está ilustrado na Figura 23. A alimentação do servo motor e o sinal de controle foram conectados diretamente ao Arduino.

Figura 23 – Protótipo da comunicação serial



Fonte: elaborado pelo autor.

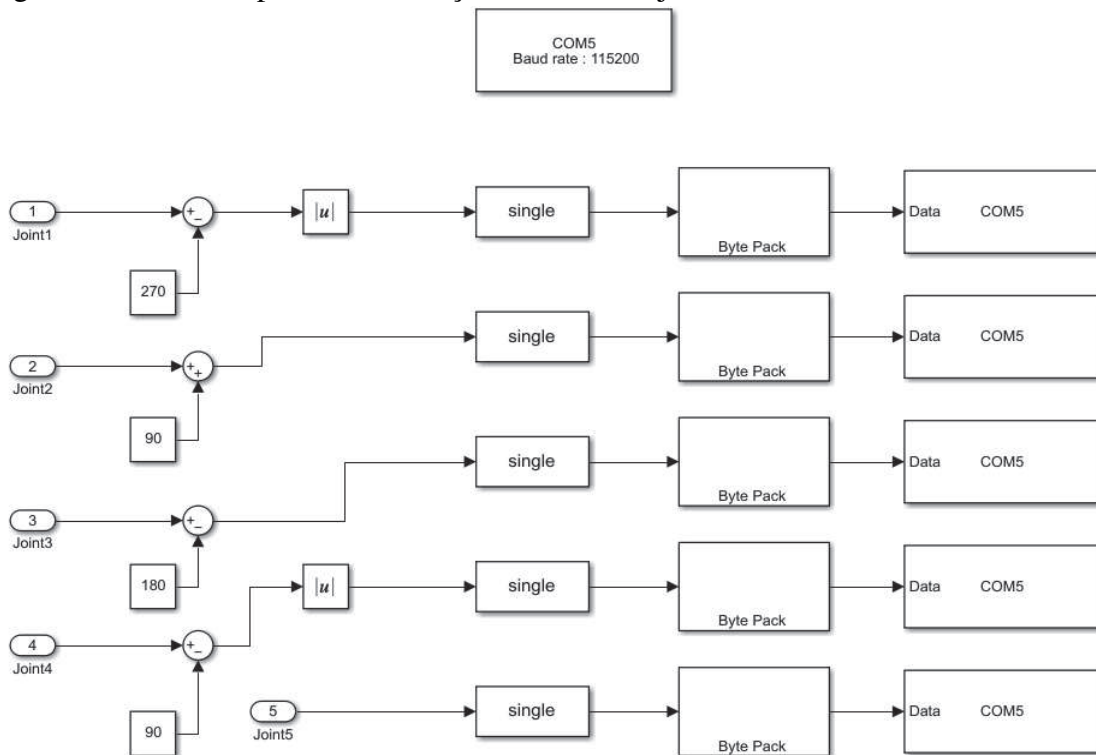
O código desenvolvido em Arduino permite a integração entre os modelos real e virtual. Utiliza-se a biblioteca "Servo.h" para criar um objeto "Servo", vinculado a junta rotativa representada pelo servo motor real. Além disso, emprega-se uma estrutura denominada "BtoF" para converter os valores angulares das juntas em *bytes*, e vice-versa. As configurações da comunicação serial e do servo motor são definidas na função "setup". No loop principal, os valores recebidos do modelo no Simulink são tratados e convertidos em movimentos no servo motor.



Além disso, funções auxiliares, como "readFromMatlab" e "writeToMatlab", também foram implementadas e servem para ler ou escrever valores vindos do modelo virtual construído no Simulink.

Essas considerações visam assegurar a consistência dos dados tratados no modelo, tendo em mente que esses dados são transmitidos sequencialmente através da porta serial, *byte* por *byte*. Levando isso em consideração e expandindo o modelo apresentado para apenas uma junta rotativa, podemos aplicá-lo às cinco juntas rotativas do manipulador, como ilustrado na Figura 24.

Figura 24 – Modelo para Comunicação Serial das 5 juntas rotativas



Fonte: elaborado pelo autor.

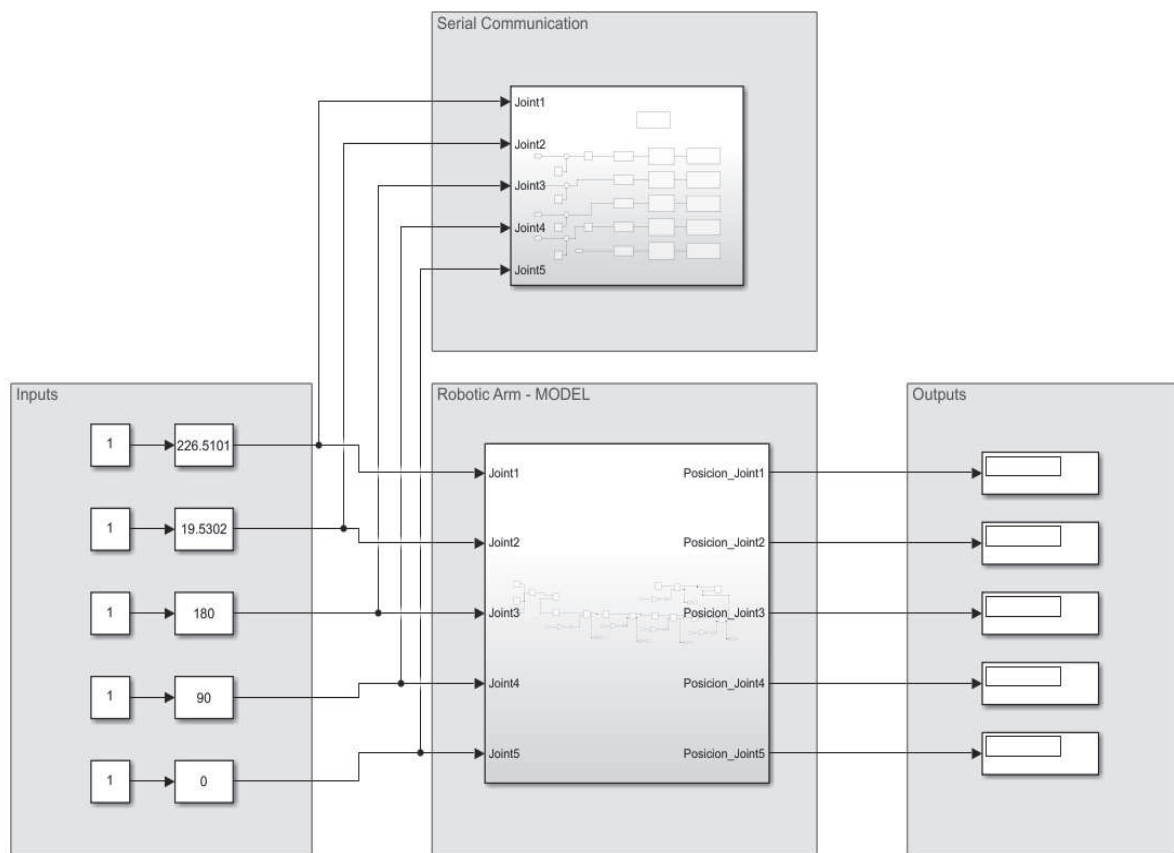
Além das modificações realizadas no modelo do Simulink, houve adaptações no código originalmente empregado para controlar um único servo motor. Para distinguir e identificar a junta apropriada, foram incorporados cabeçalhos em cada bloco *Serial Send*, que são posteriormente reconhecidos e direcionam o controle do servo motor correspondente. Esses dados são processados no código Arduino, disponível no repositório do GitHub<sup>3</sup>.

<sup>3</sup> <https://github.com/LuanGomes12/Virtualizacao-de-um-Manipulador-Robotico-de-5-Juntas-Rotativas>

#### 4.5 Modelo Virtual Completo

A partir dos ajustes feitos no modelo do Simulink e da integração da comunicação serial ao modelo, é possível realizar uma análise mais completa do funcionamento do sistema. O modelo virtual do manipulador é apresentado na Figura 25, onde cada bloco em azul representa um subsistema no Simulink. A organização em subsistemas foi adotada para proporcionar uma estrutura mais clara e legível ao modelo virtual.

Figura 25 – Modelo Virtual do Manipulador Robótico no Simulink



Fonte: elaborado pelo autor.

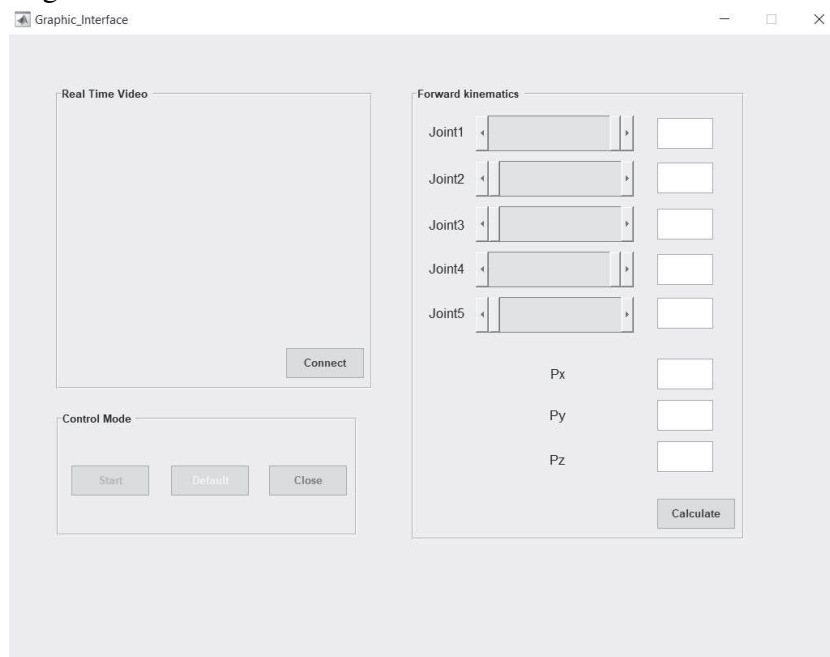
Cada um dos subsistemas desempenha um papel distinto no funcionamento do manipulador robótico. O subsistema *Serial Communication* possui os blocos encarregados de gerenciar a comunicação serial entre o modelo virtual e o modelo real. O subsistema *Robotic Arm - MODEL* representa o modelo físico do manipulador, incluindo todas as suas juntas e características físicas.

Além disso, existem os subsistemas *Inputs* e *Outputs*, que tratam, respectivamente, das entradas e saídas do modelo virtual. Cada um desses subsistemas é projetado de forma especializada para realizar funções específicas no contexto da simulação do manipulador.

## 4.6 Interface Gráfica

Para garantir maior interatividade e acessibilidade do usuário com o modelo virtual, foi criada uma interface gráfica utilizando a ferramenta GUIDE. Essa ferramenta utiliza princípios da programação orientada a objetos, um paradigma típico do Matlab, onde cada botão ou janela da interface pode ser modelado como um objeto. Para a programação de cada botão, utilizam-se funções de *callback* que representam a execução de ações quando os botões são pressionados. A Figura 26 ilustra a interface criada.

Figura 26 – Interface criada no Matlab



Fonte: elaborado pelo autor.

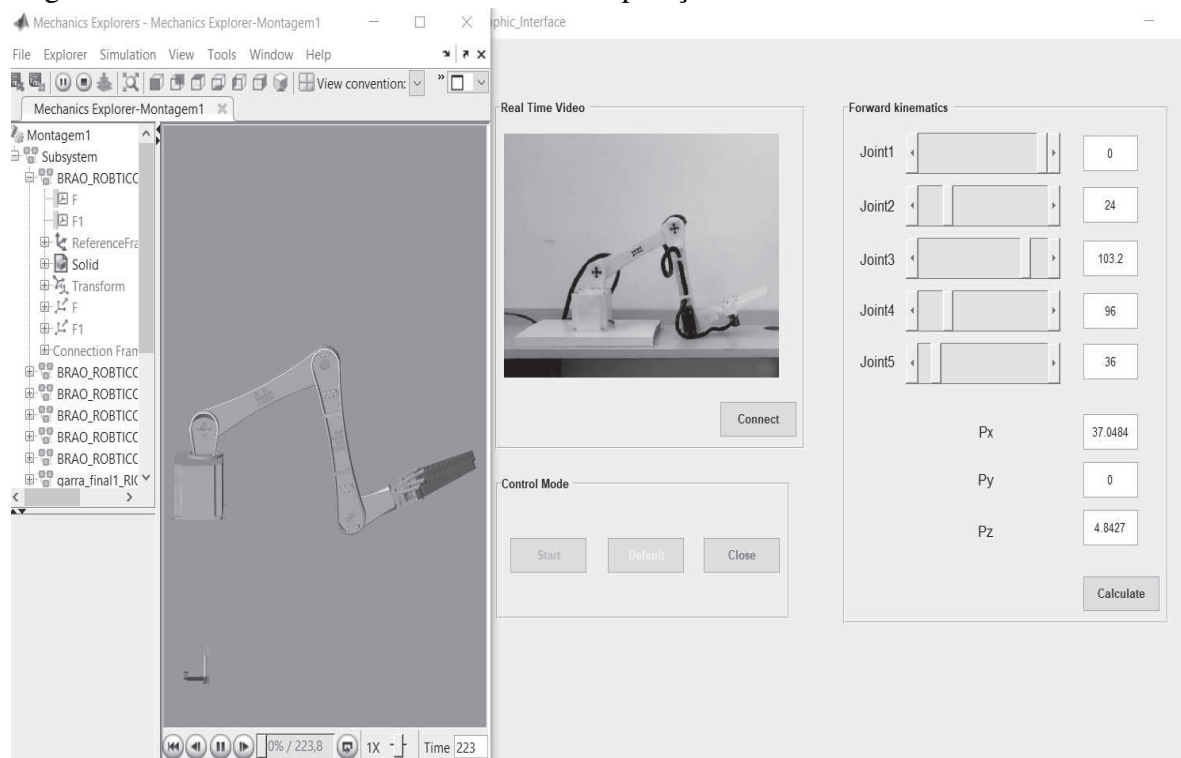
A interface é dividida em janelas internas, cada uma representando uma funcionalidade específica. A janela interna *Real Time Video* exibe o vídeo em tempo real do modelo físico por meio de uma webcam. Para ativá-la, basta pressionar o botão *Connect*. A janela interna *Control Mode* contém três botões: *Start*, *Default* e *Close*. O botão *Start* inicia o modelo virtual, permitindo modificar os valores de cada junta ao mover os controles deslizantes (*Slider Gains*) na janela interna *Forward kinematics*. O botão *Default* restaura todas as juntas do manipulador virtual à sua posição padrão, enquanto o botão *Close* encerra o modelo e fecha a interface gráfica.

Por fim, a janela interna *Forward kinematics* é responsável por implementar a Cinemática Direta do manipulador, mostrando as posições x, y e z do braço robótico. Para isso, basta pressionar o botão *Calculate*. Detalhes sobre a função de *callback* utilizada para essa funcionalidade estão disponíveis no Apêndice A.

## 4.7 Funcionamento

Após a integração de todas as ferramentas para a criação do gêmeo digital do manipulador robótico, a interface gráfica desenvolvida possibilita a análise do funcionamento do modelo virtual, a geração de movimentos em ambos os modelos e a comparação em tempo real. A Figura 27 ilustra o funcionamento do modelo virtual por meio da interface gráfica.

Figura 27 – Funcionamento da interface e comparação com o modelo real



Fonte: elaborado pelo autor.

Como teste de validação da cinemática direta, utilizou-se a biblioteca *Robotics Toolbox*, definindo valores das posições angulares de cada junta e comparando as posições X, Y e Z do efetuador no modelo virtual e na interface da biblioteca. Além disso, utilizou-se o modelo virtual no Matlab/Simulink, por meio da interface gráfica. Para isso, atribuíram-se valores mensuráveis a cada uma das juntas, como  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$  e  $360^\circ$ , e analisou-se a posição do efetuador. Vale ressaltar que os eixos mostrados no modelo virtual não correspondem aos eixos que indicam as posições X, Y e Z do efetuador na interface. Como o circuito de potência não foi implementado, não há sincronismo entre o modelo real e virtual. Ambos os modelos são apresentados na Figura 27 para ilustrar o funcionamento da interface e do modelo virtual.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, propôs-se a virtualização de um manipulador robótico didático de 5 juntas rotativas. Inicialmente, realizou-se um levantamento bibliográfico para identificar os softwares adequados à virtualização de manipuladores robóticos. Utilizando o software Autodesk Inventor, construiu-se o modelo 3D a partir do braço robótico real. Com base nesse modelo, empregou-se o software Matlab, especialmente o ambiente de simulação Simulink, para criar o modelo virtual do braço físico em um ambiente computacional.

A comunicação entre o modelo virtual e o modelo real foi estabelecida por meio do protocolo de comunicação serial, o qual demonstrou eficiência para conectar dispositivos em curtas distâncias, apresentando baixa latência. Além disso, desenvolveu-se uma interface gráfica no Matlab para visualizar simultaneamente os modelos real e virtual. Os resultados obtidos indicam que os objetivos deste trabalho foram alcançados, uma vez que o gêmeo digital do manipulador didático foi construído.

No que se refere a trabalhos futuros, algumas possibilidades são: realizar experimentações para comparar os movimentos no modelo virtual e real, validando a precisão da simulação; implementar o circuito de potência do manipulador robótico utilizado, buscando gerar movimentos simultâneos entre os dois modelos; aplicar a cinemática inversa ao modelo virtual, explorando novas possibilidades de controle e programação do manipulador; e programar movimentos virtuais para análise no braço robótico real.

As sugestões para trabalhos futuros visam aprimorar a compreensão da aplicabilidade de manipuladores robóticos no ambiente industrial e a utilização desses dispositivos para estudos na área de robótica.

## REFERÊNCIAS

- ANDRADE, K. de O.; BASSANI, T. Comunicação serial para um simulador de imersão de carrinho de rolimã, comparando json, xml e texto. **Revista Eletrônica de Iniciação Científica em Computação**, v. 18, n. 4, 2020.
- BAHGA, A.; MADISETTI, V. **Internet of Things: A Hands-On Approach**. [S.l.: s.n.], 2014.
- BAYER, F. M.; ECKHARDT, M.; MACHADO, R. Automação de sistemas. **Santa Maria-RS: Rede E-tec Brasil**, 2011.
- BENOTSMANE, R.; DUDAS, L.; KOVACS, G. Simulation and trajectory optimization of collaborating robots by application of solidworks and matlab software in industry 4.0. **Academic Journal of Manufacturing Engineering**, v. 18, n. 4, 2020.
- CARRASQUINHO, F. C. **Ferramenta de Simulação para Robô Industrial**. 2015. 63 f. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) — Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, 2015.
- CONSTANTIN, D. *et al.* Forward kinematic analysis of an industrial robot. In: **Proceedings of the International Conference on Mechanical Engineering (ME 2015)**. [S.l.: s.n.], 2015. p. 90–95.
- CRAIG, J. J. **Introduction to robotics: Mechanics and Control**. 3. ed. Harlow, Essex: Pearson, 2014.
- GOUVEIA, F.; ANDRADE, T. Indústria 4.0: O uso da modelagem 3d na prototipação industrial. 2023.
- NETO, L. J. D. B. *et al.* Desenvolvimento, modelagem matemática e simulação de um braço robótico acionado por clp. **Revista Eletrônica de Engenharia Elétrica e Engenharia Mecânica**, v. 3, n. 2, p. 105–115, 2021.
- NIKU, S. B. **Introduction to robotics: Analysis, Control, Applications**. 3. ed. Hoboken, USA: John Wiley & Sons, 2020.
- OCHOA, V.; PEIXOTO, J.; BORIN, A. Autogemini-gêmeos digitais na indústria 4.0. **Salão Integrado de Ensino, Pesquisa e Extensão da Uergs (SIEPEX)**, v. 1, n. 10, 2021.
- OLIVEIRA, S. de. **Internet das coisas com ESP8266, Arduino e Raspberry PI**. [S.l.]: Novatec Editora, 2017.
- PENHA, B. S. **Modelagem e simulação de um robô com 6 graus de liberdade utilizando Matlab e Simulink**. Trabalho de Conclusão de Curso — Universidade Tecnológica do Paraná, 2016.
- PENHA, B. S.; QUEIROZ, M. E.; ROSSINI, F. L. Modelagem e análise da cinemática direta e inversa de manipulador robótico com cinco juntas rotativas. 2016.
- RAO, R. **What are manipulator robots? Understanding their Design, Types, and Applications**. 2023. WEVOLVER. Disponível em: <<https://www.wevolver.com/article/what-are-manipulator-robots-overview-of-types-and-applications>>.

ROMANO, V.; DUTRA, M. Introdução a robótica industrial. **Robótica Industrial: Aplicação na Indústria de Manufatura e de Processo**, Edgard Blücher, São Paulo, p. 1–19, 2002.

SIMPLICIO, P. V. G.; LIMA, B. R.; JUNKES, J. A. Manipuladores robóticos industriais. **Caderno de Graduação-Ciências Exatas e Tecnológicas-UNIT-SERGIPE**, v. 3, n. 3, p. 85–94, 2016.

SOARES, M. **Braco\_Robotico**. 2023. Disponível em: <[https://github.com/matheussoares/Braco\\_Robotico](https://github.com/matheussoares/Braco_Robotico)>. Acesso em: 18 out. 2023.

TABACOW, R. P. Utilização do matlab como ferramenta de simulação da movimentação dos eixos de um robô. **TAS Journal**, v. 1, n. 1, p. 9–19, 2017.

TAKAYAMA, A.; PANHAM, A. M. Indústria 4.0: Desafios e oportunidades para a indústria brasileira. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, v. 8, n. 5, p. 2675–3375, 2022.

WEYER, S.; MEYER, T.; OHMER, M.; GORECKY, D.; ZÜHLKE, D. Future modeling and simulation of cps-based factories: an example from the automotive industry. **Ifac-Papersonline**, Elsevier, v. 49, n. 31, p. 97–102, 2016.

WEYER, S.; SCHMITT, M.; OHMER, M.; GORECKY, D. Towards industry 4.0 - standardization as the crucial challenge for highly modular, multi-vendor production systems. **IFAC-PapersOnLine**, v. 48, n. 3, p. 579–584, 2015.

ŽLAJPAH, L. Simulation in robotics. **Mathematics and Computers in Simulation**, Elsevier, v. 79, n. 4, p. 879–897, 2008.

## APÊNDICE A – CÓDIGO PARA IMPLEMENTAÇÃO DA CINEMÁTICA DIRETA DO MANIPULADOR ROBÓTICO

A função abaixo calcula as posições x, y e z no espaço tridimensional a partir da orientação e posição de cada uma das juntas do manipulador robótico.

Código-fonte 1 – Função que calcula as posição Px, Py e Pz do Manipulador Robótico

```

1 function pushbutton4_Callback(hObject, eventdata, handles)
2 % Junta 1
3 theta_1 = get(handles.slider1, 'value');
4 theta_1 = abs(theta_1 - 270);
5 % Junta 2
6 theta_2 = get(handles.slider2, 'value');
7 theta_2 = theta_2 + 90;
8 % Junta 3
9 theta_3 = get(handles.slider3, 'value');
10 theta_3 = theta_3 - 180;
11 % Junta 4
12 theta_4 = get(handles.slider4, 'value');
13 theta_4 = abs(theta_4 - 90);
14 % Junta 5
15 theta_5 = get(handles.slider5, 'value');
16
17 % Px
18 px = 18*cos((pi*theta_1)/180)*cos((pi*theta_2)/180) - 18*cos((pi
    *(theta_4 + 90))/180)*(cos((pi*theta_1)/180)*cos((pi*theta_2)
    /180)*sin((pi*theta_3)/180) - cos((pi*theta_1)/180)*cos((pi*
    theta_3)/180)*sin((pi*theta_2)/180)) + 18*sin((pi*(theta_4 +
    90))/180)*(cos((pi*theta_1)/180)*sin((pi*theta_2)/180)*sin((
    pi*theta_3)/180) + cos((pi*theta_1)/180)*cos((pi*theta_2)
    /180)*cos((pi*theta_3)/180)) + 18*cos((pi*theta_1)/180)*sin((
    pi*theta_2)/180)*sin((pi*theta_3)/180) + 18*cos((pi*theta_1)
    /180)*cos((pi*theta_2)/180)*cos((pi*theta_3)/180);
19 set(handles.edit8, 'string', num2str(px));

```



```
20
21 % Py
22 py = 18*cos((pi*theta_2)/180)*sin((pi*theta_1)/180) - 18*cos((pi
    *(theta_4 + 90))/180)*(cos((pi*theta_2)/180)*sin((pi*theta_1)
    /180)*sin((pi*theta_3)/180) - cos((pi*theta_3)/180)*sin((pi*
    theta_1)/180)*sin((pi*theta_2)/180)) + 18*sin((pi*(theta_4 +
    90))/180)*(cos((pi*theta_2)/180)*cos((pi*theta_3)/180)*sin((
    pi*theta_1)/180) + sin((pi*theta_1)/180)*sin((pi*theta_2)
    /180)*sin((pi*theta_3)/180)) + 18*cos((pi*theta_2)/180)*cos((
    pi*theta_3)/180)*sin((pi*theta_1)/180) + 18*sin((pi*theta_1)
    /180)*sin((pi*theta_2)/180)*sin((pi*theta_3)/180);
23 set(handles.edit9, 'string', num2str(py));
24
25 % Pz
26 pz = 18*sin((pi*theta_2)/180) - 18*cos((pi*theta_2)/180)*sin((pi
    *theta_3)/180) + 18*cos((pi*theta_3)/180)*sin((pi*theta_2)
    /180) - 18*cos((pi*(theta_4 + 90))/180)*(cos((pi*theta_2)
    /180)*cos((pi*theta_3)/180) + sin((pi*theta_2)/180)*sin((pi*
    theta_3)/180)) - 18*sin((pi*(theta_4 + 90))/180)*(cos((pi*
    theta_2)/180)*sin((pi*theta_3)/180) - cos((pi*theta_3)/180)*
    sin((pi*theta_2)/180)) + 10;
27 set(handles.edit10, 'string', num2str(pz));
```

## **ANEXO A – DATASHEET DO SERVO MOTOR MG995**

O presente anexo contém especificações técnicas abrangentes, sobre o servo motor MG995, incluindo faixa de tensão de operação, torque de parada, ângulo de rotação e outras características essenciais. Este dispositivo é projetado para funcionar em sistemas de alta velocidade. Possui compatibilidade com a maioria dos receptores, incluindo Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum e Hitec.

# 31150-MP

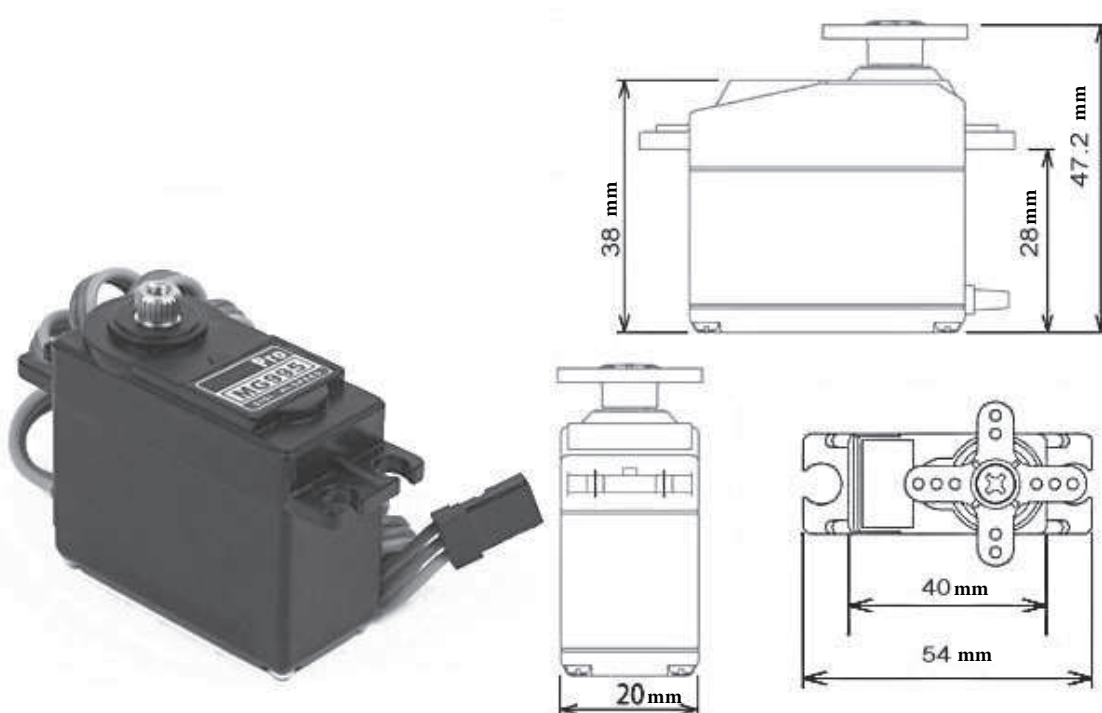
## MG995 High Speed Servo Actuator

The unit comes complete with color coded 30cm wire leads with a 3 X 1 pin 0.1" Pitch type female header connector that matches most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-speed servo actuator is not code dependant; You can use any servo code, hardware or library to control them. The MG995 Actuator includes arms and hardware to get started.

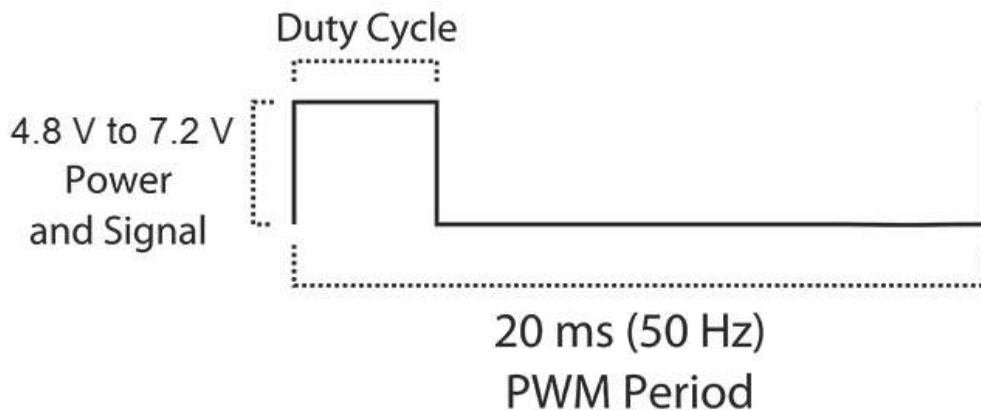
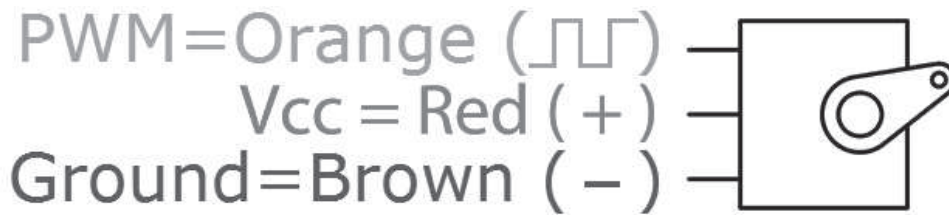
### Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 8.5 kgf·cm (4.8 V ), 10 kgf·cm (6 V)
- Rotation Angle: 120deg. (+- 60 from center)
- Operating speed: 0.2 s/60° (4.8 V), 0.16 s/60° (6 V)
- Operating voltage: 4.8 V to 7.2 V
- Dead band width: 5  $\mu$ s
- Stable and shock proof double ball bearing design
- Metal Gears for longer life
- Temperature range: 0 °C – 55 °C



# 31150-MP

## MG995 High Speed Servo Actuator



Information obtained from or supplied by mpja.com or Marlin P. Jones and Associates inc. is supplied as a service to our customers and accuracy is not guaranteed nor is it definitive of any particular part or manufacturer. Use of information and suitability for any application is at users own discretion and user assumes all risk.



**MARLIN P. JONES & ASSOC., INC.**

P.O. Box 530400 Lake Park, FL 33403

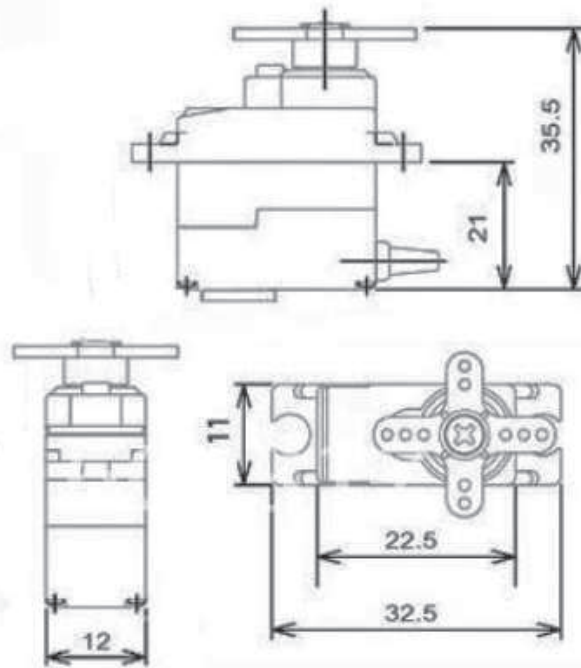
800-652-6733 FAX 561-844-8764

WWW.MPJA.COM

## **ANEXO B – DATASHEET DO SERVO MOTOR MG90S**

O presente anexo contém especificações técnicas abrangentes, sobre o servo motor MG90S, incluindo faixa de tensão de operação, torque de parada, ângulo de rotação e outras características essenciais. O dispositivo se destaca por ser um componente compacto e leve, projetado para aplicações que requerem potência de saída entre 4.8 V e 6 V. É equipado com engrenagens metálicas para aumentar a resistência e durabilidade, tornando-o ideal para uso em Helicópteros, Quadricópteros e Robôs.

## MG90S Metal Gear Servo




### MG90S servo, Metal gear with one bearing

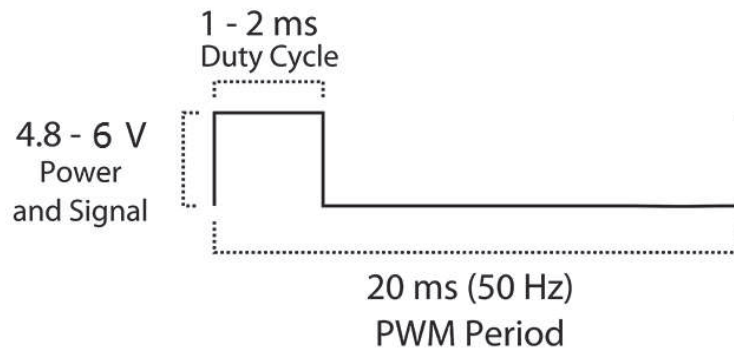
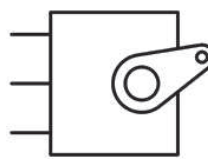
Tiny and lightweight with high output power, this tiny servo is perfect for RC Airplane, Helicopter, Quadcopter or Robot. This servo has *metal gears* for added strength and durability.

Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

### Specifications

- Weight: 13.4 g
- Dimension: 22.5 x 12 x 35.5 mm approx.
- Stall torque: 1.8 kgf·cm (4.8V ), 2.2 kgf·cm (6 V)
- Operating speed: 0.1 s/60 degree (4.8 V), 0.08 s/60 degree (6 V)
- Operating voltage: 4.8 V - 6.0 V
- Dead band width: 5  $\mu$ s

PWM=Orange (  )  
Vcc = Red ( + )  
Ground=Brown ( - )



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.